

**2. Auflage**

**Adrian Specker**

**Ein methodischer  
Leitfaden zur  
Projektentwicklung**

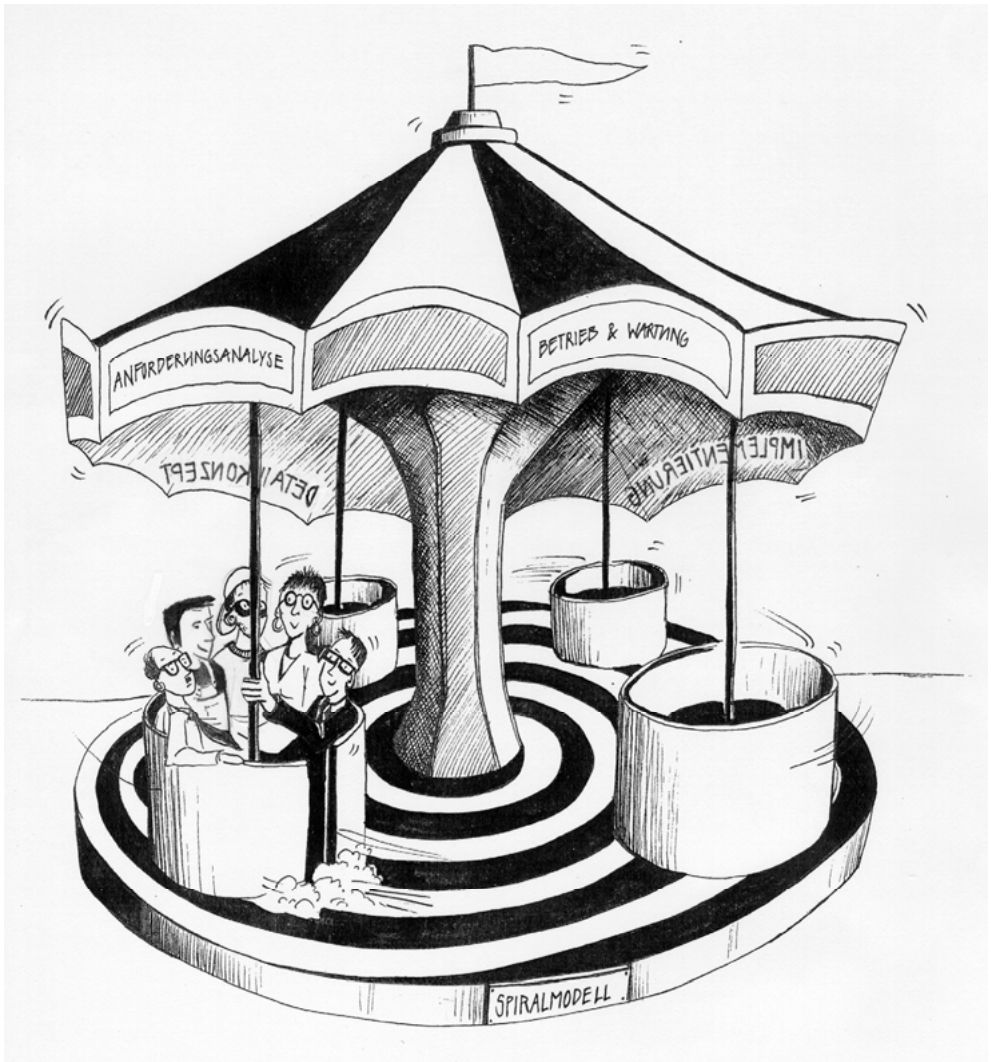
**Modellierung von**

**Informationssystemen**

**W**

**IRTSCHAFTSINFORMATIK**

**v/d/f**



Adrian Specker

# **Modellierung von Informations- systemen**

**Ein methodischer Leitfaden  
zur Projektabwicklung**

**2. Auflage**

**v/d/f**

**vdf Hochschulverlag AG an der ETH Zürich**

---

**Bibliografische Information Der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

---

Illustrator: Christian Frehner

1. Auflage 2001

**2. überarbeitete und erweiterte Auflage 2005**

© vdf Hochschulverlag AG an der ETH Zürich

This work is licensed under  
a creative commons license.



ISBN 978-3-7281-2984-0 (Printausgabe)

ISBN 978-3-7281-3713-5 (E-Book)

DOI 10.3218/3713-5

---

[www.vdf.ethz.ch](http://www.vdf.ethz.ch)  
[verlag@vdf.ethz.ch](mailto:verlag@vdf.ethz.ch)



## **Meiner Familie**



## **Vorwort**

Dieses Buch zum Thema Modellierung und Projektabwicklung von Informationssystemen hat mehrere Wurzeln. Es ist zum einen das Produkt eines Lehrauftrages für Wirtschaftsinformatik, welchen ich seit Wintersemester 1998/1999 an der Eidgenössischen Technischen Hochschule inne habe. Es ist sodann aus einem von der KTI (Kommission für Technologie und Innovation der Schweizerischen Eidgenossenschaft) geförderten Projekt hervorgegangen. Dieses Projekt hatte zum Ziel, eine Methodik für die prozessorientierte Einführung von Informationssystemen zu entwickeln.

Im Rahmen der Lehrtätigkeit und der damit verbundenen wissenschaftlichen Tätigkeit entstand der im Mittelpunkt dieses Buches stehende Ansatz zur Modellierung von betrieblichen Informationssystemen. Die Zielsetzung dieses Ansatzes besteht darin, Studenten und Praktikern der Wirtschaftsinformatik einen methodischen Rahmen zum tieferen Verständnis von prozess-, objekt-, funktions-, aufgaben- und technikorientierten Methoden zu geben.

Weiter sind sodann die beruflichen Erfahrungen des Autors im Bereich der Projektabwicklung von Informationssystemen, insbesondere ERP-Systemen (Enterprise Resource Planning), eingeflossen. Diese Erfahrungen stammen zum einen aus der Tätigkeit als Unternehmensberater auf dem Gebiet der Informationstechnologie (IT), zum anderen aber auch aus der Erfahrung als Systemintegrator einer innovativen ERP-Lösung für Klein- und Mittelbetriebe. Das vorliegende Buch ist aber auch eine Reaktion auf die Unzufriedenheit des Autors über die diffusen und häufig unübersichtlichen Modellierungsmethoden. Diese Unzufriedenheit mit den vorhandenen Methodiken hat dazu geführt, das erwähnte Rahmenwerk zur Einordnung der unterschiedlichen Modellierungsansätze zu entwickeln.

Ich möchte an dieser Stelle allen meinen Kollegen danken, die mit unzähligen Diskussionen dazu beigetragen haben, das hier vorliegende Buch in dieser Form zu erstellen. Allen voran danke ich Herrn Prof. Paul Schönsleben für sein Vertrauen und den Freiraum, der es mir ermöglicht hat, Wissenschaft und Praxis in einer optimalen Weise zu kombinieren. Christian Frehner danke ich für seine treffenden Illustrationen. Zu Dank verpflichtet bin ich sodann Robert Alard, Jürg Ammann, Dieter Ackermann, Ursula Balsiger, Markus Bärtschi, Kurt Bauknecht, Mario Becker, Wilhelm Bögershausen, Gerhard Burger, Roger Cruz, Martin de Quervain, Dunja Furrer, Sirpa Goeggel, Ingo Hartel, Ralph Hieber, Bernd Knappmann, Markus Kühni, Isabel Kühnlein Specker, Nicola Manecke, Norbert Marx, Fritz Müller, Jörg Nienhaus, Karsten Schierholt, Matthias Schnetzler, Andreas Sennheiser, Ernst Specker, Jürg Treichler, Prof. Eberhard Ulich, Alexander Verbeck, Toni Wäfler, Christine Wetli und Prof. Carl August Zehnder.

Sodann haben eine ganze Reihe von Unternehmen dazu beigetragen, dass das vorliegende Buch zustande kam. Es sind dies in alphabetischer Reihenfolge: Ascom, Audatex, Credit Suisse, Härtereier Gerster, IBM Schweiz, Interelectric, Möbelfabrik Fraubrunnen, Miracle, SAP Schweiz, seetal swiss, Sennheiser electronic, Swiss RE, Trilab und WDV. Auch diesen Unternehmen sei an dieser Stelle für ihren Beitrag herzlich gedankt.

## **Vorwort zur 2. Auflage**

Der Erfolg der ersten Auflage hat gezeigt, dass es einem Bedürfnis entsprach, eine Übersicht der gängigen Modellierungsmethoden zu erhalten. Die zweite Auflage hat zum Ziel, das Thema der Projektabwicklung umfassend zu erweitern und neu zu strukturieren, um dem Anspruch eines übersichtlichen Lehrbuches und Praxisleitfadens zu genügen.

Im Zusammenhang mit den Arbeiten zur zweiten Auflage bin ich Robert Franz, Ties-Christian Gerdes, Stefan Grossen, Klaus Höling, Alain Homberger, Nikolai Iliev, Christian Josephy, Martin Jufer, Susanne Keel, Dagmar Müller, Emil Neff, Andreas Nobs, Daniel Rom, Prof. Jörg Sennheiser, Jürgen von Sachs, Christoph Waschbüsch, Prof. Theo Wehner, Frank Weissberg und Simone Zimmermann zu grossem Dank verpflichtet.

Der Autor freut sich über Anmerkungen, Fragen und Diskussionspunkte zu diesem Buch oder zur behandelten Thematik an folgende Adresse:

adrian.specker@ethz.ch.

Zürich, den 23. September 2004

# Inhaltsverzeichnis

<b>1</b>	<b>INHALTSÜBERBLICK .....</b>	<b>17</b>
1.1	PROBLEME BEI DER MODELLIERUNG VON INFORMATIONSSYSTEMEN .....	18
1.2	ZIELSETZUNGEN DIESES BUCHES .....	19
 <b>TEIL 1: MODELLIERUNG</b>		
<b>2</b>	<b>MODELLIERUNG VON INFORMATIONSSYSTEMEN .....</b>	<b>25</b>
2.1	SYSTEME .....	26
2.2	ANALYSE VON KOMPLEXEN SYSTEMEN.....	30
2.3	ASPEKTORIENTIERTE DARSTELLUNGSTECHNIKEN .....	38
2.4	ASPEKTORIENTIERTE MODELLIERUNGSMETHODEN .....	41
2.5	MORPHOLOGIE DER MODELLIERUNGSMETHODEN.....	46
2.6	TEAMORIENTIERTE INTEGRATION DER MODELLE.....	48
2.7	LITERATUR ZUR MODELLIERUNG VON SYSTEMEN .....	49
<b>3</b>	<b>MODELLIERUNG DER PROZESSE .....</b>	<b>53</b>
3.1	PROZESSORIENTIERTE MODELLIERUNG.....	54
3.2	PROZESSORIENTIERTE METHODEN IM ÜBERBLICK.....	57
3.3	PROZESSMODELL.....	58
3.4	IDENTIFIKATION UND SYSTEMABGRENZUNG DER PROZESSE.....	62
3.5	SPEZIFIKATION VON PROZESSEN: WERTSCHÖPFUNGSKETTEN .....	64
3.6	FLUSSDIAGRAMM.....	68
3.7	SEQUENZDIAGRAMM .....	69
3.8	STELLENORIENTIERTES ABLAUFDIAGRAMM .....	70
3.9	INTEGRIERTE PROZESSORIENTIERTE METHODE: ARIS .....	74
3.10	ÜBUNG: PROZESSORIENTIERTE MODELLIERUNG .....	75
3.11	LITERATUR ZUR PROZESSORIENTIERTEN MODELLIERUNG .....	76

<b>4</b>	<b>MODELLIERUNG DER FUNKTIONEN .....</b>	<b>79</b>
4.1	FUNKTIONSORIENTIERTE MODELLIERUNG.....	80
4.2	FUNKTIONSORIENTIERTE METHODEN IM ÜBERBLICK .....	81
4.3	FUNKTIONSMODELL.....	82
4.4	IDENTIFIKATION UND SYSTEMABGRENZUNG DER FUNKTIONEN .....	83
4.5	SPEZIFIKATION VON FUNKTIONEN .....	86
4.6	FUNKTIONEN-BLOCKDIAGRAMM.....	87
4.7	DATENFLUSSDIAGRAMM.....	88
4.8	USE-CASE-DIAGRAMM .....	90
4.9	INTEGRIERTE FUNKTIONALE METHODE: STRUKTURIERTE ANALYSE .....	92
4.10	LITERATUR ZUR FUNKTIONSORIENTIERTEN MODELLIERUNG .....	93
<b>5</b>	<b>MODELLIERUNG DER OBJEKTE .....</b>	<b>95</b>
5.1	OBJEKTORIENTIERTE MODELLIERUNG .....	96
5.2	OBJEKTORIENTIERTE METHODEN IM ÜBERBLICK .....	99
5.3	OBJEKTMODELL .....	100
5.4	IDENTIFIKATION UND SYSTEMABGRENZUNG DER OBJEKTE .....	104
5.5	SPEZIFIKATION VON OBJEKTEN.....	105
5.6	ZUSTANDSÜBERGANGSDIAGRAMM.....	107
5.7	CLASS-RESPONSIBILITIES-COLLABORATORS-DIAGRAMM (CRC) .....	108
5.8	KOLLABORATIONS-DIAGRAMM .....	109
5.9	INTEGRIERTE OBJEKTORIENTIERTE METHODE: UML UND RUP .....	110
5.10	INTEGRIERTE OBJEKTORIENTIERTE METHODE: RAW .....	111
5.11	ÜBUNG: OBJEKTORIENTIERTE MODELLIERUNG .....	112
5.12	LITERATUR ZUR OBJEKTORIENTIERTEN MODELLIERUNG .....	113
<b>6</b>	<b>MODELLIERUNG DER AUFGABEN .....</b>	<b>115</b>
6.1	AUFGABENORIENTIERTE MODELLIERUNG.....	116
6.2	AUFGABENORIENTIERTE METHODEN IM ÜBERBLICK .....	117
6.3	ORGANISATIONSMODELL .....	118
6.4	IDENTIFIKATION UND SYSTEMABGRENZUNG DER AUFGABEN .....	119
6.5	SPEZIFIKATION VON AUFGABEN.....	120
6.6	STELLENORIENTIERTER INFORMATIONSFLUSS.....	122
6.7	STELLENFUNKTIONENDIAGRAMM.....	124
6.8	ARBEITSOBJEKTDIAGRAMM .....	126
6.9	INTEGRIERTE AUFGABENORIENTIERTE METHODE: MTO-ANALYSE .....	127
6.10	ASPEKTORIENTIERTE ORGANISATIONSFORMEN.....	129
6.11	LITERATUR ZUR AUFGABENORIENTIERTEN MODELLIERUNG .....	133
<b>7</b>	<b>MODELLIERUNG DER TECHNIK .....</b>	<b>137</b>
7.1	TECHNIKORIENTIERTE MODELLIERUNG.....	138
7.2	TECHNIKORIENTIERTE METHODEN IM ÜBERBLICK.....	139
7.3	SYSTEMARCHITEKTURMODELL .....	140
7.4	IDENTIFIKATION UND SYSTEMABGRENZUNG DER INFORMATIONSTECHNIK .....	143
7.5	SPEZIFIKATION EINES TECHNIKSYSTEMS .....	144

7.6	SYSTEMSCHNITTSTELLENDIAGRAMM .....	145
7.7	SYSTEMFUNKTIONENDIAGRAMM .....	146
7.8	DATENOBJEKTDIAGRAMM .....	147
7.9	SYSTEMNUTZUNGSDIAGRAMM .....	148
7.10	PROZESSTECHNOLOGIEDIAGRAMM.....	149
7.11	FUNKTIONSUNTERSTÜTZUNGSDIAGRAMM .....	150
7.12	OBJEKTZUGRIFFSDIAGRAMM.....	151
7.13	TECHNIKEINSATZDIAGRAMM.....	152
7.14	ASPEKTORIENTIERTE INFORMATIONSSYSTEME.....	153
7.15	VOLLSTÄNDIGES MODELL .....	159
7.16	LITERATUR ZUR TECHNIKORIENTIERTEN MODELLIERUNG .....	160

## **TEIL 2: PROJEKTABWICKLUNG**

8	VORGEHENSMODELLE DER PROJEKTABWICKLUNG .....	165
8.1	PROBLEME UND FEHLENDER ERFOLG VON IT-PROJEKTEN .....	166
8.2	METHODEN DER PROJEKTABWICKLUNG .....	169
8.3	DAS LEBENSPHASENMODELL GEMÄß SYSTEMS ENGINEERING .....	170
8.4	SEQUENZIELLES VORGEHENSMODELL: WASSERFALLMODELL .....	172
8.5	ZYKLISCH-ITERATIVES VORGEHENSMODELL: SPIRALMODELL .....	181
8.6	KOMBINATION AUS SEQUENZIELLEM UND ZYKLISCHEM VORGEHEN.....	186
8.7	EINSATZ VON MODELLIERUNGSMETHODEN .....	189
8.8	DER WIEDERKEHRENDE PROBLEMLÖSUNGSZYKLUS .....	190
8.9	LITERATUR ZU DEN VORGEHENSMODELLEN DER PROJEKTABWICKLUNG.....	198
9	PROJEKTANTRAG .....	203
9.1	INHALT UND VORGEHEN .....	204
9.2	PROBLEMANALYSE .....	206
9.3	PROJEKTZIELSETZUNG .....	207
9.4	PROJEKTVARIANTEN.....	209
9.5	WIRTSCHAFTLICHE UND STRATEGISCHE PROJEKTBEWERTUNG .....	211
9.6	PROJEKTANTRAG.....	219
9.7	PROJEKTENTSCHEIDUNG .....	220
9.8	LITERATUR ZUM PROJEKTANTRAG .....	221
10	GROBKONZEPT .....	223
10.1	INHALT UND VORGEHEN .....	224
10.2	GESAMTSYSTEM- UND UMFELDDANALYSE .....	227
10.3	HAUPTZIELSETZUNGEN.....	235
10.4	GROBKONZEPTVARIANTEN .....	236
10.5	BEWERTUNG DER GROBKONZEPTVARIANTEN .....	245
10.6	GROBKONZEPTBERICHT .....	247
10.7	GROBKONZEPTENTSCHEIDUNG.....	248
10.8	LITERATUR ZUM GROBKONZEPT .....	249



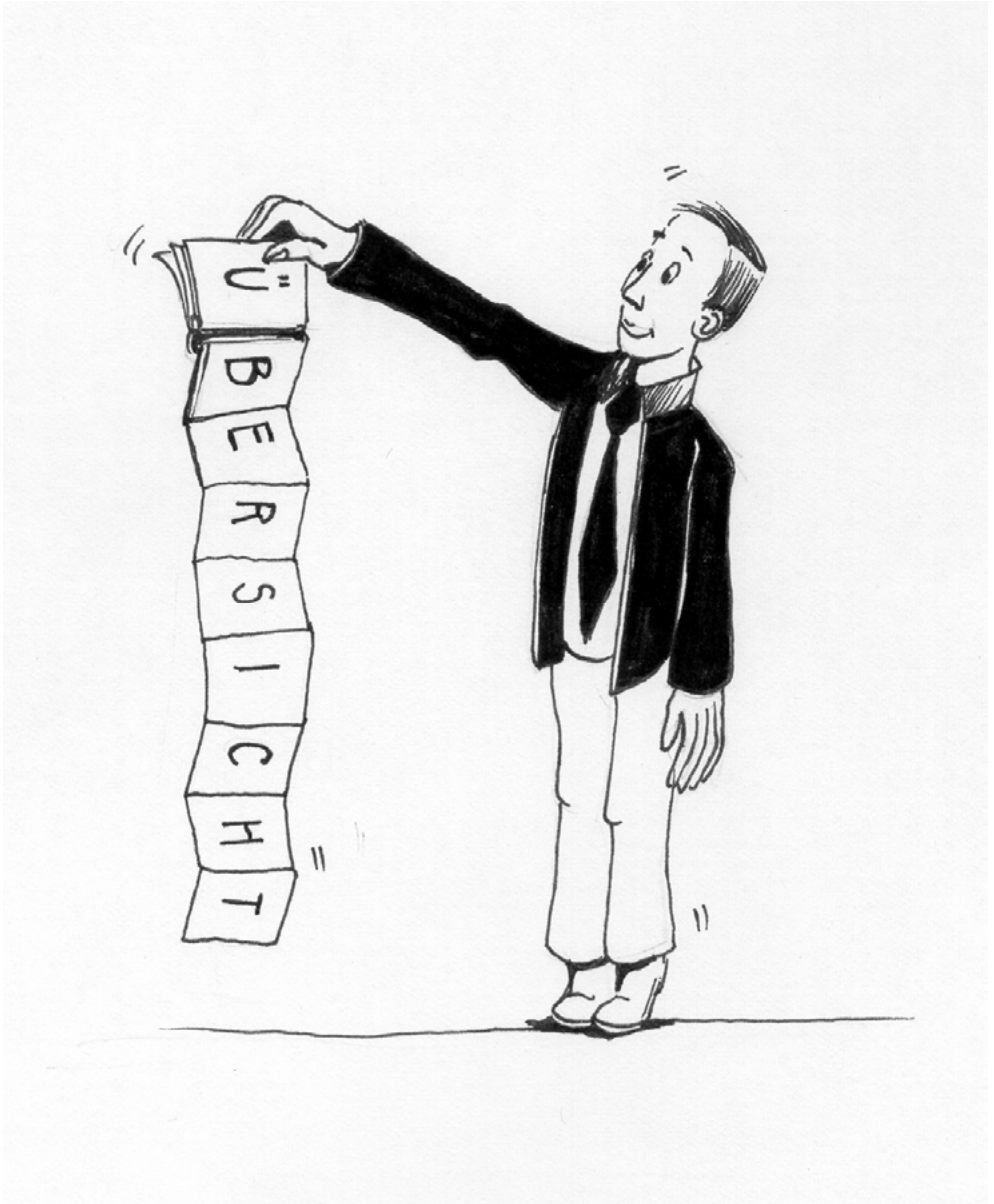
<b>11</b>	<b>KONZEPT .....</b>	<b>251</b>
11.1	INHALT UND VORGEHEN .....	252
11.2	SITUATIONSANALYSE .....	255
11.3	SYSTEMZIELE .....	261
11.4	KONZEPTION UND SYSTEMSPEZIFIKATIONEN .....	262
11.5	BEWERTUNG DER KONZEPTVARIANTEN .....	275
11.6	KONZEPTBERICHT .....	276
11.7	KONZEPTENTSCHEIDUNG .....	277
11.8	LITERATUR ZUM KONZEPT .....	278
<b>12</b>	<b>PFLICHTENHEFT UND EVALUATION .....</b>	<b>281</b>
12.1	INHALT UND VORGEHEN .....	282
12.2	PFLICHTENHEFT .....	284
12.3	ZIELSETZUNGEN UND KRITERIENKATALOG .....	290
12.4	AUSSCHREIBUNG .....	293
12.5	BEWERTUNG DER ANGEBOTE .....	298
12.6	EVALUATIONSBERICHT .....	301
12.7	EVALUATIONSENTSCHEIDUNG .....	302
12.8	LITERATUR ZU PFLICHTENHEFT UND EVALUATION .....	303
<b>13</b>	<b>VERIFIKATION UND VERTRAG .....</b>	<b>305</b>
13.1	INHALT UND VORGEHEN .....	306
13.2	VERIFIKATION UND ANALYSE DER VERTRAGSSITUATION .....	307
13.3	VERTRAGSZIELE .....	309
13.4	VERTRAGSKONZEPTE .....	310
13.5	BEWERTUNG DER VERTRAGSKONZEPTE .....	316
13.6	VERTRÄGE .....	317
13.7	VERTRAGSENTSCHEIDUNG .....	318
13.8	LITERATUR ZU VERIFIKATION UND VERTRAG .....	319
<b>14</b>	<b>DETAILLKONZEPT .....</b>	<b>321</b>
14.1	INHALT UND VORGEHEN .....	322
14.2	DETAILANALYSEN .....	325
14.3	ZIELSETZUNGEN DETAILLKONZEPT .....	327
14.4	DETAILLKONZEPTE .....	328
14.5	BEWERTUNG DER DETAILLKONZEPTE .....	335
14.6	DETAILLKONZEPTBERICHT .....	336
14.7	DETAILLKONZEPTENTSCHEIDUNG .....	337
14.8	LITERATUR ZUM DETAILLKONZEPT .....	338
<b>15</b>	<b>DESIGN UND IMPLEMENTATION .....</b>	<b>341</b>
15.1	INHALT UND VORGEHEN .....	342
15.2	SYSTEMANALYSE .....	345
15.3	SYSTEMDESIGN .....	346
15.4	IMPLEMENTATION UND KONFIGURATION .....	349

15.5	TEST .....	352
15.6	SYSTEMDOKUMENTATION .....	353
15.7	TESTABSCHLUSSENTSCHEIDUNG .....	354
15.8	LITERATUR ZU DESIGN UND IMPLEMENTATION .....	355
<b>16</b>	<b>ABNAHME UND INBETRIEBNAHME .....</b>	<b>357</b>
16.1	INHALT UND VORGEHEN .....	358
16.2	BEREITSCHAFTSANALYSE .....	359
16.3	SCHULUNG .....	360
16.4	EINFÜHRUNGSVORBEREITUNG UND INSTALLATION .....	361
16.5	ABNAHMEPRÜFUNG UND -BEWERTUNG .....	366
16.6	ABNAHMEENTSCHEIDUNG .....	369
16.7	PRODUKTIVSTART UND INBETRIEBNAHME .....	370
16.8	LITERATUR ZU ABNAHME UND INBETRIEBNAHME .....	371
<b>17</b>	<b>BETRIEB UND UNTERHALT .....</b>	<b>373</b>
17.1	BETRIEB UND UNTERHALT .....	374
17.2	PROZESSMODELL BETRIEB UND UNTERHALT .....	375
17.3	FUNKTIONSMODELL BETRIEB UND UNTERHALT .....	376
17.4	IT-PRODUKTE .....	377
17.5	ORGANISATION BETRIEB UND UNTERHALT .....	378
17.6	MORPHOLOGISCHE MATRIX FÜR BETRIEB UND UNTERHALT .....	379
17.7	BETRIEBS- UND UNTERHALTSHANDBUCH .....	380
17.8	LITERATUR ZU BETRIEB UND UNTERHALT .....	381
<b>18</b>	<b>PROJEKTMANAGEMENT .....</b>	<b>383</b>
18.1	INHALT UND VORGEHEN DES PROJEKTMANAGEMENTS .....	384
18.2	PROJEKT .....	385
18.3	PROJEKTMANAGEMENT .....	387
18.4	PROJEKTMANAGEMENTFUNKTIONEN .....	391
18.5	INHALTSMANAGEMENT .....	392
18.6	TERMINMANAGEMENT .....	395
18.7	RESSOURCENMANAGEMENT .....	396
18.8	AUFTRAGSMANAGEMENT .....	397
18.9	KOMMUNIKATIONSMANAGEMENT .....	398
18.10	RISIKOMANAGEMENT .....	399
18.11	QUALITÄTSMANAGEMENT .....	400
18.12	PROJEKTMANAGEMENTPROZESS .....	401
18.13	PROJEKTORGANISATION UND -ROLLEN .....	407
18.14	ERGEBNISÜBERSICHT PROJEKTMANAGEMENT .....	410
18.15	LITERATUR ZUM PROJEKTMANAGEMENT .....	411
<b>19</b>	<b>LITERATURVERZEICHNIS .....</b>	<b>413</b>
<b>20</b>	<b>STICHWORTVERZEICHNIS .....</b>	<b>427</b>

# Abkürzungsverzeichnis

a.a.O.	am angegebenen Ort (bereits oben zitiert)
BDE	Betriebsdatenerfassung
CAD	Computer Aided Design
CASE	Computer Aided Software Engineering
CNC	Computer Numerical Control
CPU	Central Processing Unit
CRC	Class-Responsibilities-Collaborators
CRM	Customer Relationship Management System
DB	Datenbank
DBMS	Data Base Management System
DFD	Datenflussdiagramm
DIN	Deutsche Industrie Norm
EPK	Ereignis Prozess Kette
ERP	Enterprise Resource Planning System
GU	Generalunternehmung
HW	Hardware
ISO	International Standards Organization
IT	Information Technology
ITIL	IT Infrastructure Library
KMU	Kleine und mittlere Unternehmen
LAN	Local Area Network
MAWI	Materialwirtschaft
MTO	Mensch Technik Organisation
NPV	Net Present Value
PMI	Project Management Institute
QS	Qualitätssicherung
RAW	Requirements Analysis Workshop
ROI	Return on Investment
RUP	Rational Unified Process
SCM	Supply Chain Management
SLA	Service Level Agreement
SQL	Structured Query Language
SW	Software
SWOT	Strength Weakness Opportunities Threats
UML	Unified Modelling Language
XML	eXtenden Markup Language





# 1

## Inhaltsüberblick

### Kapitelinhalt

- Über die Schwierigkeit, Informationssysteme zu modellieren
- Zielsetzungen des Buches
- Inhaltsüberblick
- Was dieses Buch nicht behandelt

Es ist unverkennbar, dass im Bereich der Modellierung von Informationssystemen in der Praxis große Unsicherheiten bestehen. So ist man sich sowohl in der Praxis als auch in der Lehre nicht einig, welches die am besten geeigneten Modellierungsmethoden sind, und dies, obwohl eine nicht geringe Anzahl an Modellierungsmethoden besteht.

Vielleicht ist auch gerade aus dieser Uneinigkeit eine große Anzahl von konkurrierenden Methoden entstanden. In Konkurrenz stehen prozessorientierte Ansätze, funktionsorientierte Techniken sowie die Methoden der objektorientierten Modellierung. Trotzdem muss festgestellt werden: Für eine ganze Berufsgruppe und insbesondere für Wirtschaftsinformatiker besteht nach wie vor keine methodische Sicherheit in ihrem Berufsalltag. Dieser Umstand verunsichert verantwortliche Projektleiter, Manager und nicht zuletzt die Anwender, welche diese methodischen Unzulänglichkeiten ebenfalls spüren. Damit stellt sich die Frage nach einem methodischen Grundgerüst. Im vorliegenden Buch wird ein integrierender Rahmen von bereits bestehenden Modellierungsmethoden präsentiert. Es werden bestehende Methoden und etablierte Begriffe integriert und neu beleuchtet.

## 1.1 Probleme bei der Modellierung von Informationssystemen

Erfahrungsgemäß stellt die Spezifikation, Auswahl und Einführung neuer Informationssysteme die involvierten Projektleiter, Berater und Anwender seit Jahren vor dieselben Schwierigkeiten:

- Wie erstellt man Spezifikationen für Informationssysteme?
- Welche Modellierungsmethoden sollen eingesetzt werden?
- Welcher Detaillierungsgrad ist phasenadäquat?
- Ist ein unternehmensweites Datenmodell zu erstellen?
- Welche Modellierungstechniken sind gerade “State of the Art”?
- Wie ist eine Evaluation vorzunehmen?
- usw.

Bei all diesen Fragen stellt man bei Wirtschaftsinformatikern eine recht hohe Unsicherheit fest. Dies ist erstaunlich, wird doch seit mehreren Jahrzehnten genau an diesem Problem gearbeitet und wurden im Verlauf dieser Zeit bereits unzählige Methoden propagiert.

Andererseits ist man aber bei einem integrierten Informationssystem tatsächlich mit einer kaum fassbaren Komplexität konfrontiert – und dies bereitet zu Recht einiges an Schwierigkeiten. Die Verunsicherung im Bereich der Methoden rührt indessen auch daher, dass regelmäßig “ultimativ” neue Methoden propagiert werden – etwa Methoden der “Objektorientierung” oder Methoden der “Geschäftsprozessgestaltung”. Diese Methoden haben in vielen Anwendungsgebieten indessen aus Sicht der angesprochenen Berufsgruppen oftmals nicht das gehalten, was sie versprochen haben.

Es ist sodann eine traurige Tatsache, dass Informatikprojekte regelmäßig die gesteckten Zielsetzungen und Erwartungen nicht im erwarteten Ausmaß erfüllen. Dies kann vielfältige Ursachen haben, und einseitige Schuldzuweisungen bringen in der Regel wenig.

So mag es aus Sicht der Softwareanbieter stimmen, dass die Erwartungen an Informationssysteme häufig unrealistisch hoch sind. Es mag auch sein, dass die Sichtweise der Anwender zutrifft, wonach Informationssysteme tatsächlich zu schwerfällig und zu träge sind und das Geschäft daher zeitweise eher behindern als unterstützen. Sicher aber trifft die Tatsache zu, dass Informationssysteme eine hohe systembedingte Komplexität aufweisen. Dies ist weder die Schuld der Anwender noch der Softwareanbieter – vielmehr sind betriebliche Informationssysteme in ihrer Natur komplex. Und diese Komplexität gilt es in den Griff zu bekommen.



## **1.2 Zielsetzungen dieses Buches**

### **1.2.1 An wen richtet sich dieses Buch?**

Dieses Buch richtet sich primär an Projektleiter und Projektmitarbeiter auf dem Gebiet der Informationstechnologie (IT), Wirtschaftsinformatiker, Berater sowie an Studenten der Wirtschaftsinformatik und Informatik. Es richtet sich generell an jene Personen, die vor der Aufgabe stehen, betriebliche Informationssysteme zu modellieren und einzuführen. Der Leser bzw. die Leserin mag vor der Aufgabe stehen, ein Pflichtenheft für eine integrierte Standardsoftware eines Industriebetriebes zu erstellen oder Spezifikationen für eine Bankapplikation formulieren zu müssen. Und häufig ist es mit der Spezifikation nicht getan, das Projekt muss erfolgreich abgewickelt werden.

### **1.2.2 Allgemeine Zielsetzungen**

Das vorliegende Buch verfolgt im Wesentlichen folgende Zielsetzungen:

- Es soll ein tieferes Verständnis für Informationssysteme entwickelt und aufgezeigt werden, wie Informationssysteme methodisch fassbar und schrittweise modelliert werden können,
- Es soll weiter ein Verständnis für alternative Vorgehensmodelle zur Projektabwicklung entwickelt werden.

Beiden Zielsetzungen liegt die These zugrunde, dass die Komplexität von Informationssystemen durch deren Mehrdimensionalität (Prozesse, Objekte, Funktionen, Aufgaben und Technik) zustande kommt. Es ist daher das Hauptziel dieses Buches, diese Dimensionen zu erläutern und die zugrunde liegenden Methodenmodelle aufzuzeigen.

### **1.2.3 Ziele hinsichtlich der Modellierung von Informationssystemen**

Das erste Hauptziel besteht darin, Informationssysteme methodisch fassbar zu machen. Dies beinhaltet den Punkt, bestehende und historisch gewachsene Modellierungsansätze in einen Gesamtrahmen zu stellen und methodisch einzuordnen. Dies soll es dem Leser erlauben, bereits bekannte Methoden der Prozess-, Funktions-, Objekt- und Aufgabenmodellierung in einen Gesamtkontext zu stellen, diese in der Praxis selbständig anzuwenden und ein Verständnis für die Zusammenhänge der Modellierungsmethoden zu entwickeln. Es werden dabei 25 verschiedene Modellierungsmethoden erläutert und damit ein breites methodisches Wissen vermittelt. Darüber hinaus werden die gängigen "integrierten Methoden" wie ARIS, UML, die strukturierte Analyse und die MTO-Analyse in den geschaffenen Gesamtrahmen eingeordnet und positioniert.

#### 1.2.4 Ziele hinsichtlich der Projektabwicklung

In einem zweiten Teil des Buches wird das Ziel verfolgt, ein Verständnis für die Vorgehensmodelle zur Gestaltung von Informationssystemen zu vermitteln. Es werden die zwei alternativen Vorgehensmodelle der Projektabwicklung erläutert: das Wasserfall- und das Spiralmodell. Aufbauend auf diesen beiden Alternativen können projektspezifische “Mischformen” dieser beiden Modelle angewendet werden.

Das Wasserfallmodell postuliert entsprechend dem Vorgehen des Systems Engineering ein sequenzielles Durchlaufen der Gestaltungsphasen – vom Grobkonzept über die Detailstudie bis zur Einführung. In jeder der sequenziell durchschrittenen Phasen wird gewissermaßen nur eine Tätigkeit (also z.B. Analyse oder Programmierung) ausgeführt, diese dafür sehr umfassend und vollständig.

Das Spiralmodell verfolgt eine zyklisch-iterative Entwicklung der Projektentwicklung. In jeder Phase des Projektes erfolgt ein Zyklus der Entwicklung des Gesamtsystems. Das System wächst ausgehend von einem Kern schrittweise oder schalenförmig an.

Jede Tätigkeit wird im zweiten Teil des Buches ausführlich beschrieben und angegeben, welche Modellierungsmethoden gemäß der propagierten Modellierungsmethodik je Phase zur Anwendung gelangen können. Es ist aber nie beabsichtigt, einen absolut starren Vorgehensplan zu propagieren, dies wäre unrealistisch und würde der Praxis nicht gerecht. Vielmehr soll die Erkenntnis vermittelt werden, dass die kontinuierliche Integration und Erweiterung der unterschiedlichen Sichtweisen (Prozess, Objekt, Funktion, Aufgabe und Technik) anzustreben ist.

#### 1.2.5 Schwerpunkt der Modellierungsbeispiele

Die in diesem Buch verwendeten Modellierungsbeispiele behandeln die Modellierungsebenen vom Grobkonzept bis zum Design im gleichen Detaillierungsgrad. Es war also nicht das Ziel, beispielsweise eine sehr detaillierte Anleitung der Unified Modeling Language (UML) zu geben. Wir sind indessen überzeugt, dass die behandelten Modellierungstechniken und Systematiken auch Analytikern und Programmierern zu einem tieferen Verständnis der Systemzusammenhänge in den Design- und Implementationszyklen dienen können.

Die präsentierten Modellierungstechniken und -ansätze lassen sich nämlich als eine ganz generelle Methodik zur Analyse von komplexen Systemen verstehen. In diesem Sinne können viele der präsentierten Modelle als generelle Systemanalysemethode verstanden und einem breiteren Anwendungsfeld zugerechnet werden.

### 1.2.6 Nicht primäre Ziele dieses Buches

Dieses Buch positioniert sich eher auf der operativen Ebene der Modellierung und Projektabwicklung als auf der strategischen Ebene. Dementsprechend müssen gewisse Themen, die ebenfalls zum effizienten und korrekten Einsatz von Informationssystemen gehören, unbehandelt bleiben. Dies betrifft Themen wie:

- Inhalte einer Informatikstrategie
- Detaillierte Auslegung der Hardware- und Kommunikationsinfrastruktur
- Budgetprozess der Informatik
- Controlling von Projektportfolios
- etc.

Zur Vertiefung des Themas, wie die Ableitung der Informatikstrategie aus der Unternehmensstrategie erfolgt, wird auf die entsprechende Literatur verwiesen. Dieses Thema des “Business/IT-Links” hat in den letzten Jahren eine große Bedeutung erlangt, und entsprechende Literaturangaben finden sich jeweils am Schluss der Kapitel.

Ganz verzichtet wird ebenfalls auf die Ausgestaltung der Hardware und die Konzeption der Kommunikationsarchitektur. Auch diese Punkte sind in jedem Projekt wichtige Erfolgsfaktoren, müssen aber an dieser Stelle ebenfalls unbehandelt bleiben.



# Teil 1

## Modellierung

Der erste Teil dieses Buches beschäftigt sich mit der Modellierung und der Spezifikation von Informationssystemen. Mithin geht es um die Fragestellung, wie ein komplexes Informationssystem gedanklich fassbar gemacht werden kann.

Vergleicht man die Gestaltung von Informationssystemen mit dem Bau eines Hauses, dann soll dieser erste Teil aufzeigen, welche Pläne von Häusern angefertigt werden können, z.B. Grundrisse, Skizzen, Leitungspläne und Umgebungspläne, und wie diese untereinander zusammenhängen.



# 2

## Modellierung von Informationssystemen

### Kapitelinhalt

- Grundkenntnisse in Bezug auf den Systembegriff
- Hierarchische und aspektbezogene Analysen von Systemen
- Modellierung der Prozess-, Funktions-, Objekt- & Aufgabensicht
- Übersichtsmatrix der Modellierungsmethoden

Dieses Kapitel legt dar, dass Informationssysteme als komplexe und mehrdimensionale Systeme analysiert und verstanden werden können. Es wird zunächst aufgezeigt, über welche Merkmale ein System verfügt und dass sich ein System primär durch die a) hierarchische und b) aspektweise Betrachtung analysieren und verstehen lässt. Weiter wird darauf eingegangen, dass sich in der Vergangenheit Modellierungsmethoden entwickelt haben, welche jeweils exakt einen Systemaspekt näher darstellen: Prozess-, Funktions-, Objekt- und Organisationsmodelle.

Zusätzlich haben sich Darstellungstechniken entwickelt, welche jeweils zwei Systemaspekte in ihrer Kombination darstellen. Es sind dies z.B. Flussdiagramme (Prozess/Funktion), Stellenorientierte Ablaufdiagramme (Prozess/Aufgabe) oder Use-Cases (Funktion/Aufgabe). Mit Hilfe einer Modellierungsmatrix mit 16 Feldern werden die verbreiteten Modellierungsmethoden positioniert. Die Methoden der Wirtschaftsinformatik lassen sich dadurch systemtheoretisch in einen vereinheitlichten Rahmen integrieren.



## 2.1 Systeme

### 2.1.1 Systembegriff

Beer, einer der Begründer der Systemtheorie, definiert Systeme wie folgt: “Das Wort System steht ... für Konnektivität. Wir meinen damit jede Ansammlung miteinander in Beziehung stehender Teile (...). Was wir als System definieren, ist deshalb ein System, weil es miteinander in Beziehung stehende Teile umfasst und in gewisser Hinsicht ein (...) Ganzes bildet” (Beer 1962, S. 24).

Bei der Analyse und Gestaltung von größeren Systemen (Unternehmen, Informationssysteme, etc.) steht man vor dem Problem, dass die Komplexität des Gesamtsystems von einer einzelnen Person gedanklich oftmals nicht mehr in seiner Ganzheit fassbar ist. Im Rahmen von Projekten stellt man deshalb häufig fest, wie sehr Kenntnisse von betrieblichen Systemen auf verschiedene “Köpfe” verteilt sind. Es werden sodann meist einzelne Aspekte eines Systems vereinfachend in den Vordergrund gestellt. Von Weizsäcker drückt dies folgendermaßen aus:

“Einige Dinge hängen enger zusammen, alles hängt mit allem zusammen. Da wir aber den Zusammenhang von allem mit allem nicht zu denken, sondern nur zu behaupten vermögen, denken wir Einzelnes. Unsere Schwäche ist, dass wir dann das Einzelne isoliert denken und nicht in seinem echten Zusammenhang. Dann stoßen wir darauf, dass das doch falsch ist – und dies ist das philosophische Grunderlebnis. Und dann versucht man es besser zu machen, bis man entdeckt, dass man es immer noch falsch gemacht hat” (von Weizsäcker 1989, S. 98).

Im Rahmen von Systemtheorien, wie beispielsweise dem Systems Engineering (Haberfellner 2002) und der Systems Dynamics (Sterman 2000), wird aus diesem Grunde versucht, komplexe Systeme durch geeignete funktionale und strukturelle Abstraktionen besser fassbar und gestaltbar zu machen. In gewisser Weise stützen sich viele Erkenntnisse auf grundlegende Prinzipien und Erkenntnisse bezüglich der Denkweise des Menschen ab (Kognition).

Systeme sind aufgrund der oben stehenden Merkmale zusammenfassend primär durch folgende Merkmale gekennzeichnet:

- **Systeme bilden eine Einheit (z.B. ERP-System)**
- **Systeme sind in sich gegliedert (z.B. “Personal”, “Einkauf”)**
- **Systeme lassen sich nicht als Ganzes fassen**
- **Elemente stehen in Beziehung zueinander**

### 2.1.2 Komplexität

Es wurde im letzten Punkt angesprochen: Systeme allgemein – und im speziellen Unternehmens- bzw. Informationssysteme – sind häufig sehr komplex. Aber wodurch wird diese Komplexität hauptsächlich bedingt?

Wie bereits einleitend angedeutet, wird die Komplexität durch “Zusammenhänge”, d.h. durch das Zusammenspiel von unterschiedlichen Systemelementen bedingt. Man könnte vielleicht folgende Analogie anführen: Ein einfaches Pendel ist für sich alleine noch nicht komplex. Wenn man aber zwei oder drei unabhängige Pendel durch Federn lose koppelt, dann entstehen plötzlich mehrdimensionale und daher sehr komplexe Bewegungen. Informationssysteme sind nun in diesem Sinne ebenfalls mehrdimensional: Prozesse, Funktionen, Objekte und Aufgaben sowie viele weitere Faktoren sind in gewissem Sinne ebenfalls gekoppelt und hängen eng zusammen. Faktoren, die zur Komplexität beitragen, sind:

- Mehrere Dimensionen (Prozesse, Funktionen, Objekte, ...)
- Sozio-technisches System (Interaktion, Qualifikation, ...)
- Grosse Anzahl von Funktionen und Objekten
- Hohe Anzahl von Systembenutzern
- Unterschiedliche organisatorische Bereiche
- Unterschiedlichste Informationsbedürfnisse
- Zeitlicher Wandel des Systems (neue Bedürfnisse)

Informationssysteme sind sozio-technische Systeme, d.h. sie enthalten ein soziales und ein technisches Teilsystem. Diese durchdringen sich gegenseitig, und die Akzeptanz und die Motivation der Anwender wird durch diese Abhängigkeit beeinflusst. So werden unter Umständen die zur Verfügung stehenden Systemfunktionen überhaupt nicht oder falsch genutzt bzw. sogar zwangsläufig “überlistet”.

Informationssysteme decken überdies eine sehr große Anzahl von Funktionen ab, gelegentlich sogar mehr Funktionen als einem lieb ist. Auch die Zahl der verschiedenen Datenelemente und Benutzer ist sehr groß. Ein besonderes Problem ist sodann der zeitliche Wandel: Die Anforderungen ändern sich kontinuierlich, und zeitverschoben befinden sich die Systeme daher in einem stetigen Wandel.

Auch die Schnittstelle zwischen Mensch und Rechner – das User Interface – trägt zur Komplexität bei. Verschiedene Benutzer haben unterschiedlichste Informationsbedürfnisse. Insbesondere können aber auch scheinbar ähnliche Organisationsbereiche je eigene Bedürfnisse aufweisen.

### 2.1.3 Komplexität formal begreifbar machen

Dies führt uns zu einer wichtigen Zielsetzung und Frage in Bezug auf die Modellierung: Weshalb wird eigentlich modelliert? Das Hauptziel von Modellierungsmethoden besteht darin, die erwähnte Komplexität eines Informationssystems begreifbarer zu machen. Überdies muss die Komplexität durch Analysen und Gestaltungskonzepte von der informellen Ebene in eine formale Sprache transformiert werden.

Das Wissen in einem Unternehmen über die zugrunde liegenden Prozesse, Objekte, etc. steckt häufig einzig in den "Köpfen" der Mitarbeiter. Bei der Einführung von Informationssystemen soll eine formale Sprache dazu dienen, alle beteiligten Personen auf einen ähnlichen Stand zu bringen. Erst damit kann eine Diskussionsbasis geschaffen werden.

Andererseits müssen häufig Anforderungen auch absolut verbindlich festgelegt werden, unter Umständen sogar als Bestandteil von Verträgen mit Softwarehäusern. Denn nur so lassen sich rechtlich klar umrissene Angebote einholen. Aber auch die beteiligten Stellen innerhalb der Firma sollen erfahren, in welchem Umfang Ihre Bedürfnisse abgedeckt werden.

Neben rein verbalen Beschreibungen sind somit gerade *grafische Darstellungen und Modellierungstechniken* von großer Bedeutung: Sie dienen dazu, die Realität oder die Konzepte gewissermaßen abzubilden und in eine vom Menschen leichter fassbare Form zu bringen.

Ideen und Entwürfe sind in der modellhaften Form besser diskutierbar und analysierbar. Dies spielt insbesondere in Projektteams eine große Rolle. Komplexe Informatikprojekte können nicht von einzelnen Personen vollumfänglich abgewickelt werden. Man ist darauf angewiesen, dass funktionierende Teams über eine funktionierende Sprache verfügen.

Wie weiter unten gezeigt wird, war in der Vergangenheit gerade dies häufig ein Problem von Informatikprojekten. In jeder Phase gelangen unterschiedliche Fachbereiche (Berater, Analytiker, Programmierer, etc.) zum Einsatz, und alle verwenden ihre eigenen Darstellungstechniken.

Auf die Modellierungsmethoden der Systems Dynamics (Sterman 2000), welche ihre Stärke darin haben, Rückkoppelungs- und Verstärkungsmechanismen unterschiedlicher Systemklassen zu erkennen und aufzuzeigen, werden wir an dieser Stelle nicht eingehen. Ihr Anwendungsgebiet liegt bis heute eher bei logistischen Fragestellungen oder auf der strategischen Ebene als in der Wirtschaftsinformatik.

### 2.1.4 Sensibilisierung der Modellierungsproblematik

Zur Sensibilisierung in Bezug auf die Modellierungsproblematik sollte idealerweise eine kleine Übung durchgeführt werden. Im unten stehenden Kasten ist eine Beschreibung eines Fahrradherstellers enthalten:

Ein Fahrradunternehmen mit den Abteilungen Verkauf, Fertigung und Service vertreibt drei Produkte: Standardfahrräder, konfigurierte Fahrräder und Spezialanfertigungen. Infolge des starken Wachstums wird der Kauf einer Standardsoftware zur Unterstützung der administrativen Tätigkeiten in Betracht gezogen (Kundendaten, Angebote, Aufträge, Vor- und Nachkalkulation, Lieferpapiere, Rechnungen, Einkauf etc.).

Sowohl die Verkaufs- als auch die Serviceaufträge sollen künftig informatikgestützt abgewickelt werden. Insbesondere die Kalkulation der konfigurierten Produkte auf Basis von Standardkomponenten muss zukünftig besser unterstützt werden. Aber auch die Fertigung von Spezialteilen in den Abteilungen Fertigung und Service sollen in ihrer Kalkulation der Material- und Fertigungskosten entlastet werden.

Das Komponentenlager soll zukünftig ebenfalls informatikgestützt betrieben werden. Alle Abteilungen erhalten die Möglichkeit, Materialpositionen zu überprüfen und bei Bedarf zu reservieren und abzubuchen.

Die Buchhaltung des Fahrradgeschäftes verbleibt weiterhin beim Treuhänder – abgesehen von der Debitoren- und der Kreditorenbuchhaltung (Geschäftsführer). Die Adressen der Kunden sollen sowohl der Buchhaltung als auch dem Vertrieb zur Verfügung stehen.

Versuchen Sie, dieses System zu modellieren bzw. grafisch zu strukturieren. Nehmen Sie an, Sie müssten dieses betriebliche System visualisieren und allenfalls einem möglichen Softwarelieferanten schildern. Es ist an dieser Stelle anzufügen, dass die abgegebene verbale Beschreibung natürlich bereits selber einer Modellierung entspricht. In der Regel liegt ja eine solche Analyse noch nicht vor, sondern muss erst erstellt werden. Beantworten Sie zunächst die beiden unten stehenden Fragen.

*Frage 1: Ist das beschriebene System komplex und wenn ja, weshalb?*

*Frage 2: Wie lässt sich diese Komplexität formal beschreiben?*

Die folgenden Abschnitte werden Ihnen jetzt aufzeigen, welche Ansätze zur Systemspezifikation existieren und wie das System beschrieben werden kann.

## 2.2 Analyse von komplexen Systemen

### 2.2.1 Systemabgrenzung

Zunächst werden wir uns mit einigen allgemeinen Ansätzen des Systems Engineerings beschäftigen. Als erstes Merkmal von Systemen wurde oben vermerkt: *“Systeme bilden eine Einheit”*. *“Offene Systeme”* – im Gegensatz zu geschlossenen Systemen – haben Beziehungen mit ihren Umssystemen. Die Nahtstelle des Systems mit den Umsystemen heißt *Systemgrenze* (s. Abb. 1). Als Umfeld bezeichnet man gelegentlich jenen Teil der Umsysteme, welche von besonderer Relevanz sind und daher näher betrachtet werden sollten. In einem ersten Schritt muss man sich in Projekten entscheiden, welche Elemente zum System (Eingriffssystem) gehören sollen bzw. welche Elemente im Rahmen der Umfeldanalyse betrachtet werden sollten. In Projekten lassen sich gewisse Bereiche ausgrenzen bzw. durch Schnittstellen vom System abgrenzen.

Oft erweist es sich als sinnvoll, das Systemumfeld zunächst eher weit zu fassen, um nicht in die Gefahr zu laufen, lediglich isolierte Lösungsansätze zu verfolgen. Erfahrungsgemäß kann sich im Verlauf eines Projektes eine einmal definierte Systemgrenze durchaus auch verschieben. Ein allgemeines Prinzip der Modellierung von offenen Systemen besteht darin, die Systemgrenzen so zu legen, dass die Systeme möglichst geschlossen sind und sich *relativ isolierte Systeme* ergeben, welche nur über eine Anzahl klar definierter Ein- und Ausgänge verfügen.

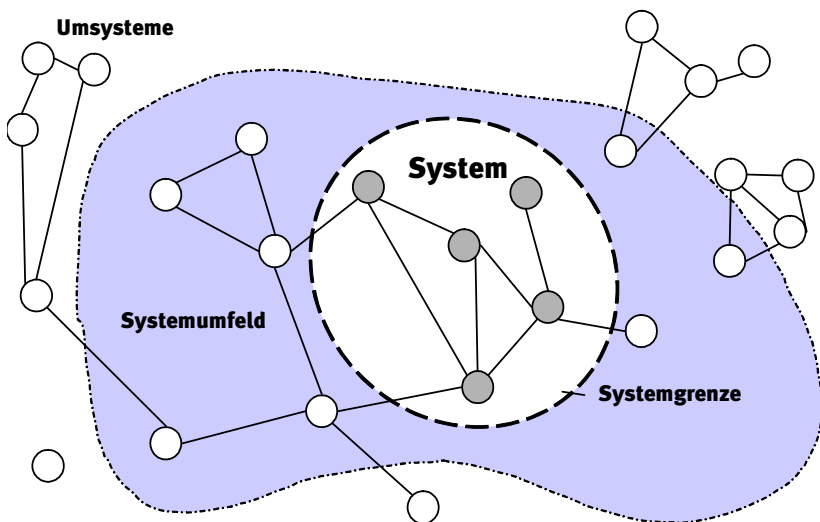


Abb. 1: Beispielhafte Systemabgrenzung

### 2.2.2 Black-box-Methode

Von der Black-box-Methode zur Systembeschreibung spricht man, wenn man von der inneren Struktur und den Eigenschaften des Systems abstrahiert und ausschließlich die Beziehungen des Systems mit seinem Umfeld bzw. den Umsystemen analysiert. Der Begriff “Black-box” kommt daher, dass bei dieser Analyse bewusst darauf verzichtet wird, Licht ins Innere der Systemstruktur und der allenfalls komplexen Mechanismen zu bringen. Mit Hilfe der Black-box-Methode lassen sich gerade auch sehr komplexe Systeme in ihrer Wirkung gut beschreiben. So ließe sich z.B. eine Quarzuhr auf relativ einfache Art und Weise dadurch beschreiben, dass die Zeit abgelesen, die Uhrzeit eingestellt sowie die Batterie ausgewechselt werden kann. Die allenfalls sehr komplizierte innere technische Funktionsweise der Quarzuhr bleibt bei dieser Betrachtung allerdings verborgen.

Die Black-box-Methode zwingt zu einer umfassenden Analyse aller Kontextbeziehungen eines Systems und damit zu einer wirklich vollständigen Analyse der Aussenbeziehungen. Es zwingt zu einer übersichtsmässigen “Vogelperspektive”, und man konzentriert sich auf die Wirkungen des Systems und verhindert ein vorschnelles Eintauchen in die Details.

Die Black-box-Methode wird meistens in einem ersten Schritt angewendet und dient dazu, die Grundlagen für die weiteren Schritte zu liefern. Der schwarze Kasten kann später “geöffnet” werden – jeder Input und Output muss aber in der inneren Struktur irgendwie behandelt werden.

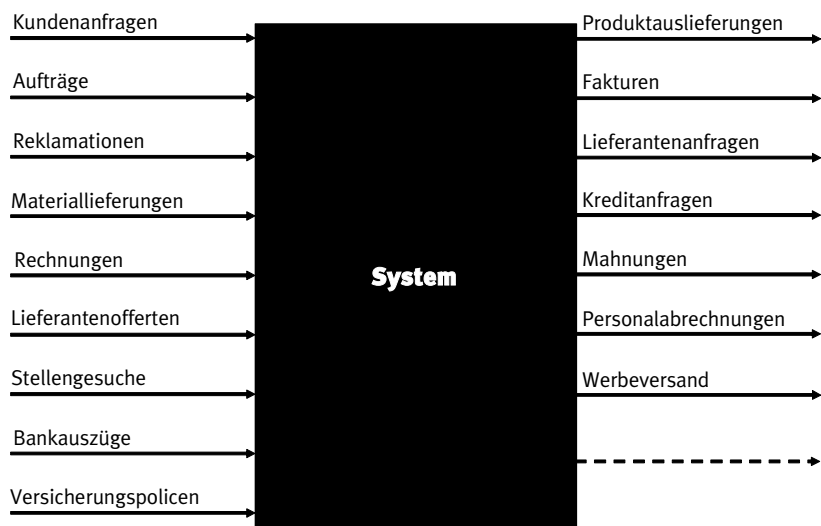


Abb. 2: Black-box-Betrachtung

### 2.2.3 Analyse von Systemhierarchien

Als zweites Merkmal von Systemen wurde angeführt: *“Systeme sind in sich gegliedert”*. Dies bedeutet, dass Teile eines Systems selber wiederum Systeme darstellen können und somit *Systemhierarchien entstehen* – man spricht von Über- und Untersystemen bzw. Systemebenen. Ein Unternehmenssystem kann z.B. in seine Unternehmensbereiche unterteilt werden, ein Staat in Länder/Kantone oder ein Körper in seine Körperteile. Es muss als grundlegende Eigenschaft betrachtet werden, dass sich Systeme hierarchisch analysieren und darstellen lassen. Da diese Untersysteme selber wieder Systeme darstellen, greift man bei der Bildung von Systemhierarchien auf die oben genannten Grundsätze der Systemabgrenzung zurück: Es werden relativ isolierte und gut abgrenzbare Untersystemeinheiten gebildet (s. Abb. 3).

*Dabei sind zwei Hierarchieformen zu unterscheiden:* Bestandteil- und Vererbungshierarchien. Die Bestandteilhierarchie besagt, dass ein Element die darunterliegenden Elemente als Bestandteile einschließt (*“Auto besteht aus Motor und Getriebe”*). Eine Vererbungshierarchie (Spezialisierung) drückt aus, dass ein Element seine Eigenschaften an die darunterliegenden Elemente weitergibt und diese Spezialformen darstellen (*“Auto ist ein spezielles Fahrzeug”*). In der vorliegenden Abbildung sind am Beispiel einer Funktionshierarchie sowohl die Bestandteilhierarchien (Linien) als auch Vererbungshierarchien (Pfeile in Richtung *“Vorfahr”*) dargestellt.

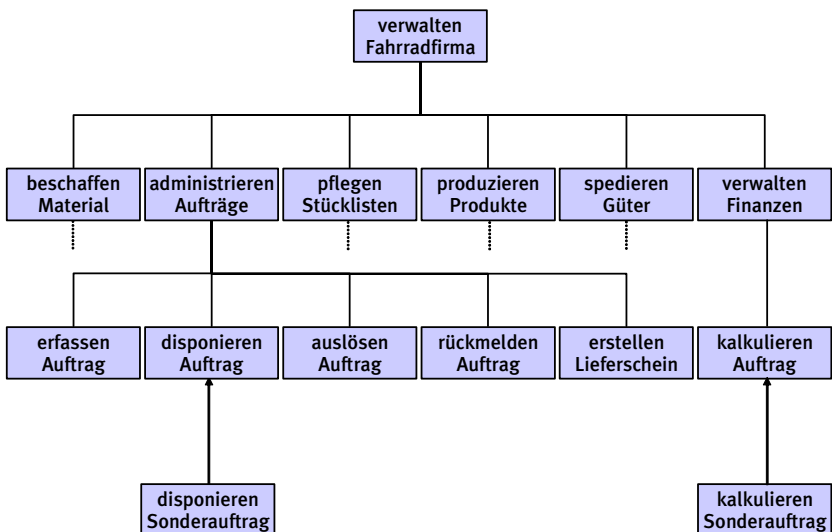


Abb. 3: Beispiel einer Funktionshierarchie



## Aspektweise Analyse des Systems und seiner Operationen

Als drittes Merkmal von Systemen wurde vermerkt, dass sich Systeme *nicht als Ganzes fassen lassen*. Komplexe Systeme lassen sich daher meist nur *aspektweise* modellieren und begreifbar machen. Man untersucht das System mit anderen Worten unter einem bestimmten Gesichtspunkt (Informationsprozess, Organisationssicht, Blutversorgung im Körper etc.). Die daraus gewonnen Systeme heißen “Teilsysteme”.

Nach welchen Aspekten können nun Informationssysteme analysiert und als Teilsysteme dargestellt werden? Zur Beantwortung dieser Frage lohnt sich ein Blick auf die Elemente eines sozio-technischen Systems. Ein sozio-technisches System kann in stark abstrahierter Weise als eine große Anzahl von in raum-zeitlicher Verbindung stehenden Operationen verstanden werden (s. Abb. 4). Diese Operationen werden in einer zeitlichen Abfolge abgewickelt und beinhalten unterschiedliche Objekte (Person, Lieferant, ...). Operationen werden von definierten Aufgaben wahrgenommen bzw. können ähnlichen Funktionen zugeordnet werden.

In sozio-technischen Systemen stellen demnach “Tätigkeiten” – vollzogen durch Handlungen und **Operationen** – das zentrale Kriterium des Systems dar (Leont’ev 1977). Unter dem Begriff “Operationen” verstehen wir in der Folge die in einem gegebenen Analysestadium tiefstliegenden “atomaren Handlungseinheiten”, wie beispielsweise “erfassen Stunden” oder “zahlen Person”. Selbstverständlich könnten auch diese Operationen später bei Bedarf nochmals weiter unterteilt werden.

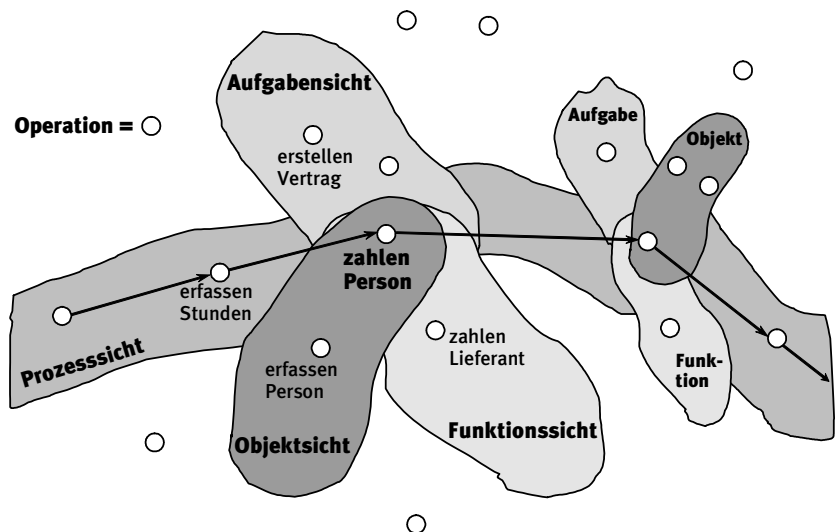


Abb. 4: Aspektweise Gliederung der Operationen eines Systems

Zentral ist nun der Umstand, dass sich diese Gesamtheit der Operationen nach unterschiedlichen *Gesichtspunkten strukturieren lässt*.

Sie lassen sich stets wie folgt gruppieren:

<b>Prozess der Operationenabfolge</b>	erfassen Stunden → zahlen Person
<b>Verwandte Elementarfunktionen</b>	zahlen Lieferant, zahlen Person
<b>Analoge Bearbeitungsobjekte</b>	erfassen Person, zahlen Person, ...
<b>Zusammengehörende Aufgaben</b>	erstellen Vertrag, zahlen Person, ...

Das bedeutet nichts anderes, als dass wir uns genau einen Aspekt herausgreifen und die Gesamtheit der Operationen entsprechend diesem Aspekt neu gruppieren und strukturieren. Wir könnten uns also beispielsweise entscheiden, die Prozesse, die Funktionen, die Objekte oder aber die Stellen/Aufgaben näher zu betrachten und detaillierter zu analysieren. Die funktionale Sicht darf nicht mit der funktionalen Organisation verwechselt werden – die Organisationssicht entspricht der Aufgabensicht!

Dementsprechend entstehen auf Basis derselben Operationenmenge vier unterschiedliche, aspektorientierte Teilsystemmodelle (s. Abb. 5). Man spricht von Prozess-, Funktions-, Objekt- und Organisationsmodellen.

Damit verbunden ist die Erkenntnis, dass die Modelle nicht voneinander unabhängig sind. Sie lassen sich voneinander ableiten und durch Vergleich in Bezug auf die Konsistenz und Vollständigkeit überprüfen.

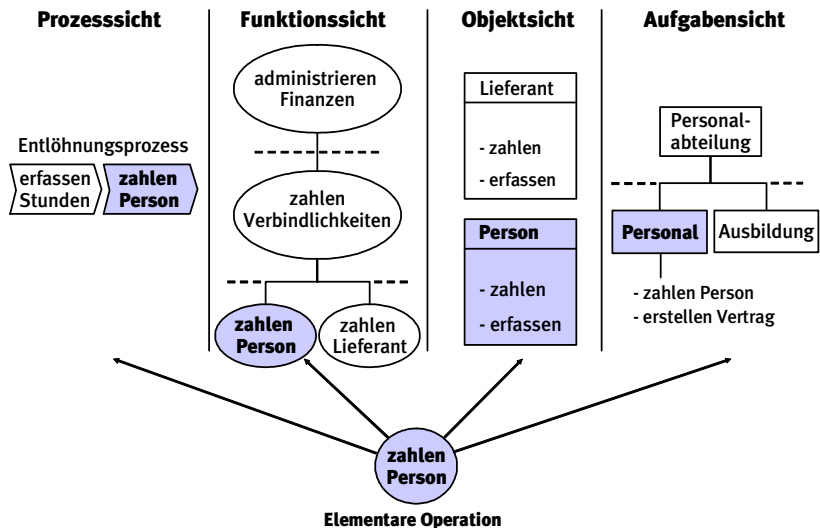


Abb. 5: Aspektweise Modellierung und Darstellung des Systems

### 2.2.5 Formale Definitionen der vier Aspektsysteme und ihrer Elemente

Aufgrund der obigen Definition des Begriffs “Operation” sowie ausgehend vom Ansatz der aspektweisen Betrachtung eines sozio-technischen Systems, ergeben sich folgende Definitionen als Ausgangsbasis für die weiteren Ausführungen:

#### Aspekt der Prozesse

*Definition: Die Prozesssicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der sachlogischen und zeitlichen Abfolge der Operationen.*

*Definition: Prozesse sind zusammengehörige und in sich relativ geschlossene Einheiten von Operationenabfolgen der unterschiedlichen Systemniveaus.*

#### Aspekt der Funktionen

*Definition: Die Funktionssicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der Verwandtschaft und Ähnlichkeit der Elementarfunktionen.*

*Definition: Funktionen sind zusammengehörige und in sich relativ geschlossene Einheiten von verwandten und ähnlichen Elementarfunktionen der unterschiedlichen Systemniveaus.*

#### Aspekt der Objekte

*Definition: Die Objektsicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der ihm zugrunde liegenden Bearbeitungselemente.*

*Definition: Objekte sind zusammengehörige und in sich relativ geschlossene Einheiten von Bearbeitungselementen der unterschiedlichen Systemniveaus.*

#### Aspekt der Aufgabe / Stellen

*Definition: Die Aufgabensicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der stellen- bzw. personenbezogenen Zuordnung der Operationen.*

*Definition: Aufgaben sind zusammengehörige und in sich relativ geschlossene Einheiten von Stellen zugeordneten Operationen der unterschiedlichen Systemniveaus.*

Man beachte, dass diese Definitionen über die “Operationen” definiert sind und sich aus einer “Bottom-up-Betrachtung” ergeben. “Bottom up” heißt, dass von Elementen der untersten Systemebene ausgegangen wird. Dies steht im Gegensatz zur Möglichkeit, Prozesse, Funktionen, Objekte und Stellen eigenständig und voneinander losgelöst zu definieren.

### 2.2.6 Analyse der Beziehungen

Als viertes Merkmal von komplexen Systemen wurde die Eigenschaft angeführt, dass Systemelemente untereinander *in Beziehung stehen*. Die Analyse der Beziehungen zwischen den Systemelementen nimmt bei der Modellierung von Informationssystemen *die* zentrale Stellung ein. Das ist nicht weiter verwunderlich, begründen doch – wie oben erwähnt – gerade erst die mannigfachen Beziehungen die Komplexität des Systems. Die Analyse der Beziehungen zwischen den Systemelementen führt zur *Struktur* bzw. dem *Ordnungsprinzip* des Systems.

Beziehungen können in verschiedenster Ausprägung und Form bestehen. So lassen sich gemäß der vorhergehenden Abbildung beispielsweise Beziehungen zwischen Funktionen im Sinne eines Funktionsbaumes aufzeigen. Ein weiteres Beispiel wären Objektmodelle, die Beziehungen zwischen Geschäftsobjekten aufzeigen. In der nachfolgenden Darstellung ist am Beispiel des Fahrradgeschäftes illustriert, in welcher Weise Objekte, wie beispielsweise das “Kundenobjekt”, mit anderen Objekten in Beziehung stehen können (s. Abb. 6). Ein weiteres Beispiel stellen Organigramme dar, die die expliziten Unterstellungsbeziehungen zwischen organisatorischen Aufgaben darstellen.

Alle diese Darstellungen stellen Beziehungen zwischen Elementen des gleichen Aspektes dar (Funktionen zu Funktionen, Objekte zu Objekte, Organisationseinheit zu Organisationseinheit).

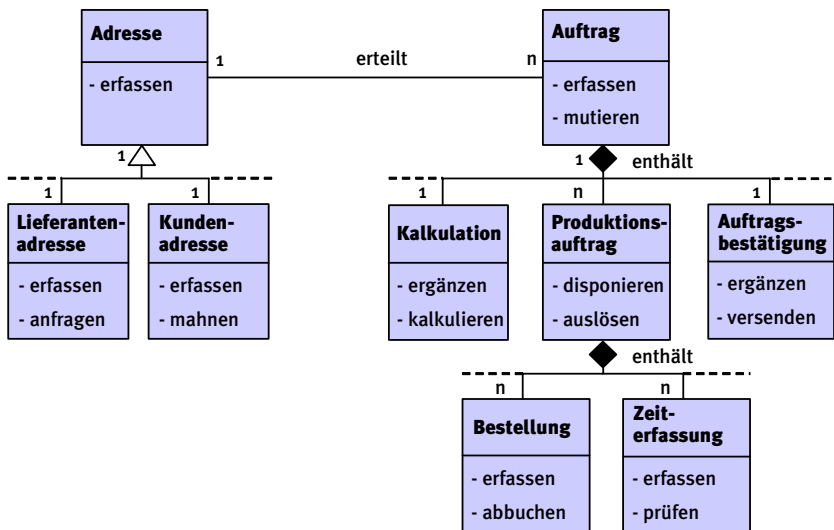


Abb. 6: Beispielhafte Beziehungen zwischen Systemelementen

### 2.2.7 Beziehungen zwischen unterschiedlichen Systemdimensionen

Die Problematik des oben angefügten Übungsbeispiels (Abschnitt 2.1) verdeutlicht nun aber, dass die Systemkomplexität gerade auch durch Beziehungen der verschiedenen Teilaspekte zustande kommen kann. Schwierigkeiten bereitet die Modellierung im oben genannten Beispiel der Fahrradfirma nämlich deshalb, weil die vier “Dimensionen” auf vielfältige Weise miteinander verknüpft sind:

So gibt es mehrere Prozesse, die *dieselben* Funktionen verwenden: z.B. kann die Funktion “kalkulieren Spezialteile” sowohl im Service- als auch im Auftragsabwicklungsprozess vorkommen. Andererseits können auch gleiche Objekte in verschiedenen Funktionen vorkommen: die Kundenadressen kommen sowohl in der Debitorenbuchhaltung als auch in der Vertriebsfunktion vor.

Demnach muss der Analyse von Beziehungen gleicher Systemelemente auch eine Analyse zwischen den verschiedenen Aspektelelementen folgen. Es sind nun nämlich genau diese *vierdimensionalen* Zusammenhänge, die die Komplexität von Informationssystemen begründen. Im Prinzip könnte man die vorhandenen Beziehungen mathematisch durch “Tupel in einem vierdimensionalen Raum” beschreiben (s. Abb. 7):

(Serviceprozess, erfassen Auftrag, Kundenobjekt, Vertrieb)

(Verkaufsprozess, erfassen Auftrag, Kundenobjekt, Vertrieb), etc.

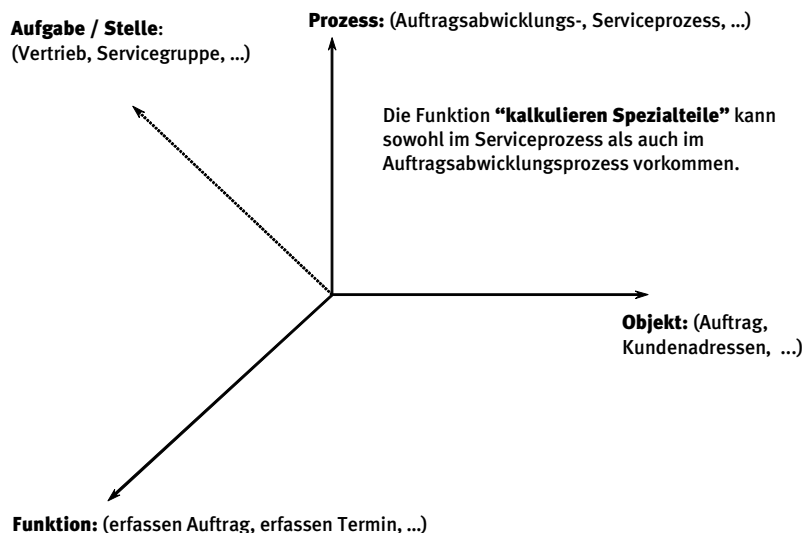


Abb. 7: Vier Dimensionen eines Informationssystems

## 2.3 Aspektorientierte Darstellungstechniken

### 2.3.1 Darstellung der Beziehungen zwischen zwei Aspekten

Leider gelingt es nicht, alle vier genannten Dimensionen in ihrem Zusammenhang vollumfänglich auf einem zweidimensionalen Blatt Papier in grafischer Form festzuhalten. Dies wurde bereits im obigen Übungsbeispiel zum Fahrradhersteller klar. Allerdings können in einer grafischen Darstellung zwei Aspekte in ihrer Verknüpfung dargestellt werden. Zur Illustration soll das Beispiel eines Funktions-/ Aufgabendiagrammes erläutert werden. Die nun folgende Darstellung zeigt die Funktionen, und als zweiter Aspekt werden Aufgaben als Verbindung zu den Funktionen eingezeichnet (s. Abb. 8). In diesem Sinn zeigt das Use-Case-Diagramm die Aspekte mit ihren Beziehungen: Funktionen und Aufgaben. Allerdings geht in dieser Darstellung die hierarchische Tiefe der Funktionen verloren, man beschränkt sich auf eine gewisse Ebene der Funktionshierarchie.

Fazit: Die Komplexität des Systems und die Vielfalt der Beziehungen ermöglicht es in der Regel nicht, das gesamte Informationssystem in einer einzigen Grafik darzustellen. Damit bleibt meist nur übrig, sich auf Teilmodelle zu konzentrieren und Systeme durch eine Anzahl von Diagrammen darzustellen.

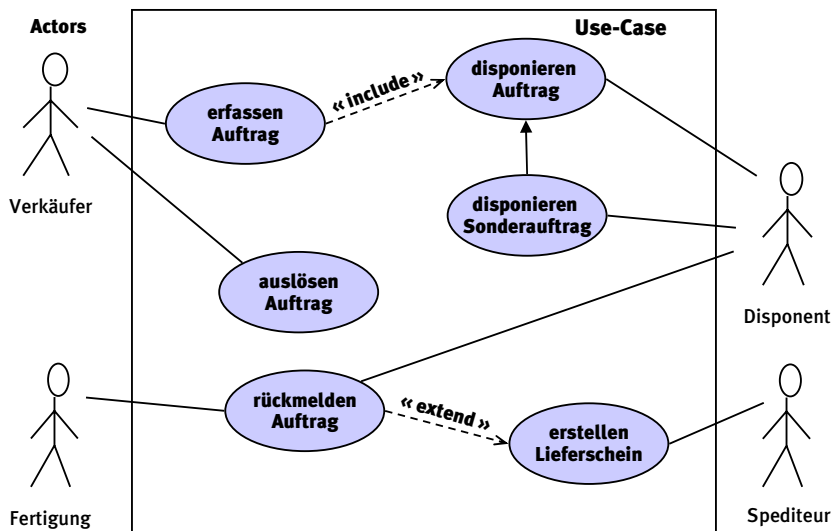


Abb. 8: Use-Case-Diagramm

### 2.3.2 Matrix der Aspektdarstellungen und ihrer Beziehungen

Es lassen sich, wie oben gezeigt, jeweils mindestens zwei Dimensionen mit ihren Beziehungen darstellen (beispielsweise eine Gruppe von Funktionen mit ihren Aufgaben). Geht man nun davon aus, dass sich im Sinne einer allgemeineren Regel zwei Dimensionen kombiniert darstellen lassen, dann ergeben sich entsprechend einer Matrix mit 16 Feldern auch 16 verschiedene Kombinationen von Darstellungen (s. Abb. 9).

Auf der Diagonale finden wir zunächst die “reinen” Darstellungen, die – wie oben dargestellt – Elemente desselben Aspekts miteinander verknüpfen (Prozesse mit Prozessen, Objekte mit Objekten, etc.). Dies entspricht den oben erwähnten vier Systemmodellen.

Die anderen Felder der Matrix kombinieren jeweils zwei Aspekte bzw. Systemdimensionen und unterscheiden zwischen einer primären Sicht bzw. sekundären Sicht. Das im letzten Abschnitt erläuterte “Use-Case-Diagramm” kombiniert dementsprechend die Funktionen als primären Aspekt mit den Aufgaben als sekundärem Aspekt. Es ist in der unten stehenden Matrix entsprechend positioniert und der Funktionssicht zugeordnet (s. Abb. 9).

Die 16 Darstellungen sind allerdings nicht vollständig voneinander unabhängig. Vielmehr sind die Darstellungen an der Diagonalen gespiegelt und zeigen enge Bezüge.

<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht	Prozessmodell			
Funktionssicht		Funktionsmodell		
Objektsicht			Objektmodell	
Aufgabensicht		Use-Case-Diagramm		Organisationsmodell

Abb. 9: Matrix der Beziehungen zwischen den Aspekten

### 2.3.3 Modellierungsmatrix zur Einordnung von Modellierungsmethoden

Auf dieser Basis und in Erweiterung des obigen Beispiels können wir uns nun fragen, inwiefern sich bestehende Diagramme (z.B. Datenflussdiagramme, Use-Cases, etc.) den Feldern dieser zweidimensionalen Modellierungsmatrix zuordnen lassen.

Es gelingt, die verbreitetsten und gebräuchlichsten Modellierungsdiagramme den "weißen Feldern" der oben postulierten Modellierungsmatrix zuzuordnen (s. Abb. 10). Dies betrifft die Darstellungen der Prozesssicht, der Funktionssicht und der Objektsicht. Bei der Aufgabensicht gibt es zwei kleinere Lücken, die trotz einer Literaturrecherche und Expertenbefragungen nicht geschlossen werden konnten. Es waren dies das "Stellenfunktionsdiagramm" und das "Arbeitsobjektdiagramm".

Damit zeigt das Postulat, dass in einem Diagramm jeweils Beziehungen von zwei Systemaspekten abgebildet werden, erstaunlich gute Resultate: Den 16 Feldern der Modellierungsmatrix können die wesentlichen Darstellungstechniken der Prozess-, Funktions-, Objekt- und Aufgabenmodellierung exakt zugeordnet werden. Dieses Matrix-Modell ist daher in der Lage, den bestehenden aspektorientierten Modellierungstechniken einen vereinheitlichten Rahmen zu geben und die tieferen Zusammenhänge der unterschiedlichen und historisch gewachsenen Modellierungsmethoden aufzuzeigen. Es wird damit auch ersichtlich, dass zur formalen, grafischen Systemspezifikation diese 16 Diagrammtypen notwendig sind.

<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht	Prozessmodell	Funktionen-Blockdiagramm	Zustandsübergangsdiagramm	Stellenorientierter Informationsfluss
Funktionssicht	Flussdiagramm	Funktionsmodell	Class-Responsibilities-Collaborators	Stellenfunktionen-diagramm
Objektsicht	Sequenzdiagramm	Datenflussdiagramm	Objektmodell	Arbeitsobjekt-diagramm
Aufgabensicht	Stellenorientiertes Ablaufdiagramm	Use-Case-Diagramm	Kollaborationsdiagramm	Organisationsmodell

Abb. 10: Modellierungsmatrix: Prozess, Funktion, Objekt, Aufgabe



## 2.4 Aspektorientierte Modellierungsmethoden

Im Verlauf der Jahre sind verschiedene Modellierungstechniken entstanden, die meist einen Aspekt *übermäßig stark* in den Vordergrund gestellt haben.

So entstanden *prozessorientierte*, *funktionsorientierte*, *objektorientierte* und *aufgabenorientierte* Methoden. Diese lassen sich gemäß der oben stehenden Matrix in den Kolonnen klar positionieren. Im Verlauf der letzten Jahrzehnte gab es sodann einen Wandel des Hauptaspektes der Modellierung (s. Abb. 11).

In den 70er-Jahren entstanden primär prozessorientierte Methoden – insbesondere das Flow Chart. Dies ergab sich aus der Programmierung mit Hilfe von Subroutine und von if-then bzw. goto Anweisungen.

In den 80er-Jahren ergab sich im Gleichschritt mit der Entstehung von prozeduralen und modularorientierten Sprachen (Pascal, C) ein Fokus hin zu funktionsorientierten Betrachtungsweisen (Datenflussdiagramme).

In einem dritten Entwicklungsschritt verbreiteten sich insbesondere in den 90er-Jahren die objektorientierten Ansätze, mit Sprachen wie C++, etc. Treibende Kraft waren also meist die Programmiersprachen.

Etwas losgelöst von diesen Methoden entwickelten sich die aufgabenorientierten Ansätze.

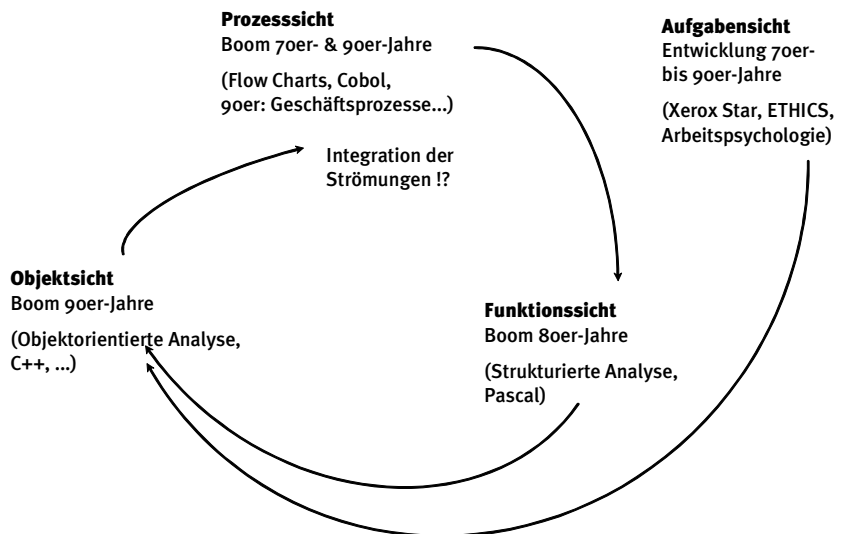


Abb. 11: Wandel des Aspektfokus im Verlauf der Zeit



### 2.4.2 Funktionsorientierte Modellierungsmethoden

Unter “funktionsorientierten Methoden” werden wir im Folgenden jene Modelle verstehen, die den funktionsorientierten Aspekt als Hauptfokus haben und von den Funktionen ausgehen (s. Abb. 13).

Viele Unternehmen sind auf Basis funktionaler Kriterien organisiert. Dies führt gelegentlich zu Verwechslungen zwischen der Funktions- und der weiter unten angeführten Aufgabensicht. Es wäre aber ein nicht zulässiger Umkehrschluss, die beiden Aspekte nur deswegen gleichzusetzen, weil es Unternehmen gibt, die ihre Aufgaben- bzw. Organisationsstruktur aufgrund funktionaler Gesichtspunkte definieren.

Das Hauptmodell der Funktionssicht ist das Funktionsmodell. Dieses lässt sich unabhängig davon, ob das Unternehmen funktions-, prozess- oder aufgabenorientiert organisiert ist, erstellen. In analoger Weise ergeben sich die weiteren funktionalen Darstellungen durch die Ergänzung mit den Aspekten der anderen Sichtweisen (s. Kapitel 5):

- Den Funktionen zugeordnete Prozesse: Funktionen-Blockdiagramm
- Den Funktionen zugeordnete Objekte: Datenflussdiagramm
- Den Funktionen zugeordnete Aufgaben: Use-Case-Diagramm (s. Abschnitt 2.3)

		Hauptfokus ↓		
primär sekundär	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
	Prozesssicht	Funktionen-Blockdiagramm		
Funktionssicht	Funktionssicht	Funktionsmodell		
	Objektsicht	Datenfluss-Diagramm		
Aufgabensicht	Aufgabensicht	Use-Case-Diagramm		

Abb. 13: Funktionsorientierte Modellierungsmethoden

### 2.4.3 Objektorientierte Modellierungsmethoden

Unter “objektorientierten Methoden” verstehen wir jene Modelle, die den objektorientierten Aspekt als Hauptfokus haben und bei ihren Modellen von den Objekten- und Objekthierarchien ausgehen (s. Abb. 14).

Während die funktionale Sicht Operationen nach der “Ähnlichkeit der Bearbeitung” strukturiert, fasst die objektorientierte Sichtweise diese nach dem “*bearbeiteten Objekt*” zusammen. Ganz natürlich entsteht dadurch auch die objektorientierte Sichtweise im programmiertechnischen Sinn (Zusammengehörigkeit von Objekt und Methode). Dies steht im Gegensatz zur lange Zeit vorherrschenden, eher “datenorientierten Sichtweise”.

Im Zentrum der objektorientierten Modellierung steht das Objektmodell.

Dementsprechend ergänzen die weiteren Darstellungsmethoden der objektorientierten Modellierung diese Objekte mit Beziehungen zu den verbleibenden Aspekten. Es ergeben sich folgende Darstellungen:

- Objekten zugeordnete Prozesse: Zustandsübergangsdiagramm
- Den Objekten zugeordnete Funktionen: Class-Responsibilities-Collaborators (CRC).
- Den Objekten zugeordnete Aufgaben: Kollaborationsdiagramm

<div>primär</div> <div>sekundär</div>		<div>Hauptfokus</div> <div></div>		
		Prozesssicht	Funktionssicht	Objektsicht
Prozesssicht			Zustandsübergangsdiagramm	
Funktionssicht			Class-Responsibilities-Collaborators	
Objektsicht			Objektmodell	
Aufgabensicht			Kollaborationsdiagramm	

Abb. 14: Objektorientierte Modellierungsmethoden

#### 2.4.4 Aufgabenorientierte Modellierung

Unter “aufgabenorientierten Methoden” verstehen wir jene Diagramme, die den Aufgabenaspekt zum Hauptfokus haben und die Modelle vom Organisationsmodell ableiten (s. Abb. 15). Die Aufgabensicht strukturiert die Operationen nach ihrer Stellenzugehörigkeit innerhalb des betrachteten Systems.

Im Mittelpunkt der aufgabenorientierten Modellierung steht das Organisationsmodell (allgemeiner wäre der Begriff “Aufgabenmodell”). Das Organisationsmodell kann zwar Ähnlichkeiten mit dem Funktionsmodell aufweisen oder in anderen Fällen auch Ähnlichkeiten mit dem Prozessmodell besitzen. Es sei aber nochmals festgehalten, dass es nicht mit diesen Modellen verwechselt werden darf, da das Funktionsmodell das System unabhängig von der bestehenden Organisation betrachtet. So kann es auch durchaus sein, dass verschiedene Stellen genau dieselben Funktionen ausüben. Es bestehen weiter folgende Modelle:

- Alle mit einer Aufgabe verbundenen Prozesse: Stellenorientierter Informationsfluss
- Einer Stelle bzw. Aufgabe zugeordnete Funktionen: Stellenfunktionsdiagramm.
- Einer Aufgabe zugeordnete Objekte: Arbeitsobjektdiagramm

<div style="display: flex; align-items: center; justify-content: center;"> <div style="transform: rotate(-45deg);">primär sekundär</div> </div>					
		Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht					Stellenorientierter Informationsfluss
Funktionssicht					Stellenfunktionsdiagramm
Objektsicht					Arbeitsobjektdiagramm
Aufgabensicht					Organisationsmodell

Hauptfokus  
↓

Abb. 15: Aufgabenorientierte Modellierungsmethoden

## 2.5 Morphologie der Modellierungsmethoden

In den vorangegangenen Abschnitten haben wir uns gefragt, wie sich die bekannten und bewährten Modellierungsmethoden in der postulierten Modellierungsmatrix positionieren lassen. Selbstverständlich könnte man auch umgekehrt fragen: Was sollte ein bestimmtes Feld der Matrix darstellen? Bei dieser Fragestellung hilft uns die unten stehende Morphologie (s. Abb. 16). In den Feldern der Matrix ist angegeben, welche Elemente ein entsprechender Diagrammtyp tatsächlich darzustellen hat.

Illustrieren wir dies am Beispiel des Use-Case-Diagrammes (schattiertes Feld). Es besagt: “Den direkten Nachfolgern einer Funktion Aufgaben zuordnen” (was unter “direkten Nachfolgern” genau zu verstehen ist, wird im nächsten Abschnitt ausgeführt). Der Grundgedanke ist indessen der Folgende: Es sollen in einem ersten Schritt (“primär”) Funktionen eingezeichnet und diesen anschließend (“sekundär”) Aufgaben zugeordnet werden.

Man wird also beispielsweise in der Funktionssicht immer **primär eine Auswahl von Funktionen** treffen. Allerdings stellt sich dann sofort die Frage, welche Funktionen zu wählen sind. Ist man hier frei? Oder gibt es ein bestimmtes methodisches Vorgehen, welches die Auswahl “rezepthaft vorgibt”? In der Matrix wird als generelle Regel vermerkt, dass man jeweils von “direkten Nachfolgern” auszugehen hat. Was dies bedeutet, ist, wie bereits erwähnt, Gegenstand des nächsten Abschnitts.

<div> <div>primär</div> <div>sekundär</div> </div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht	Den Prozessen Prozesse zuordnen	Direkten Nachfolgern einer Funktion Prozesse zuordnen	Direkten Nachfolgern eines Objektes Prozesse zuordnen	Direkten Nachfolgern einer Stelle Prozesse zuordnen
Funktionssicht	Direkten Nachfolgern eines Prozesses Funktionen zuordnen	Den Funktionen Funktionen zuordnen	Direkten Nachfolgern eines Objektes Funktionen zuordnen	Direkten Nachfolgern einer Stelle Funktionen zuordnen
Objektsicht	Direkten Nachfolgern eines Prozesses Objekte zuordnen	Direkten Nachfolgern einer Funktion Objekte zuordnen	Den Objekten Objekte zuordnen	Direkten Nachfolgern einer Stelle Objekte zuordnen
Aufgabensicht	Direkten Nachfolgern eines Prozesses Aufgaben zuordnen	Direkten Nachfolgern einer Funktion Aufgaben zuordnen	Direkten Nachfolgern eines Objektes Aufgaben zuordnen	Den Aufgaben Aufgaben zuordnen

Abb. 16: Morphologie der Beziehungsdarstellungen

### 2.5.1 Auswahl der “direkten Nachfolger” in der primären Sicht

Greifen wir zur Illustration des Begriffes der “direkten Nachfolger” das bereits bekannte Beispiel des Use-Case-Diagrammes auf (s. Abschnitt 2.3.1). Hier könnten zunächst alle Funktionen der obersten Hierarchieebene des Funktionsbaumes ausgewählt werden, d.h. gemäß der unten stehenden Darstellung jene Elemente, die sich als *direkte Nachfolger* (engl.: “direct descendants”) der Funktion “verwalten Fahrradfirma” befinden (s. Abb. 17). Erst anschließend werden die Aufgaben mit ihren Beziehungen zu den Hauptfunktionen eingezeichnet.

In einem weiteren Schritt wird nun gemäß der unten stehenden Abbildung eine der Hauptfunktionen ausgewählt (z.B. “administrieren Aufträge”) und die direkt darunterliegenden Funktionen des Baumes (schattiert) in einem Use-Case-Diagramm dargestellt (s. Abb. 8).

Man wählt also nicht eine beliebige Auswahl von Elementen des primären Aspektes aus, sondern durchschreitet die Baumstruktur jeweils methodisch “top-down”. Auf diese Weise erhält man in sich geschachtelte und zusammenhängende Diagramme.

Dieses Vorgehen kann man als generelle Leitlinie verstehen: Man greife die “direkten Nachfolger” eines Elementes aus dem Baum heraus und stelle die ausgewählten Elemente als primären Aspekt dar. Anschließend werden die sekundär zu betrachtenden Elemente eingezeichnet. “Nachfolger” bedeutet also nicht, dass sich die Elemente untereinander folgen!

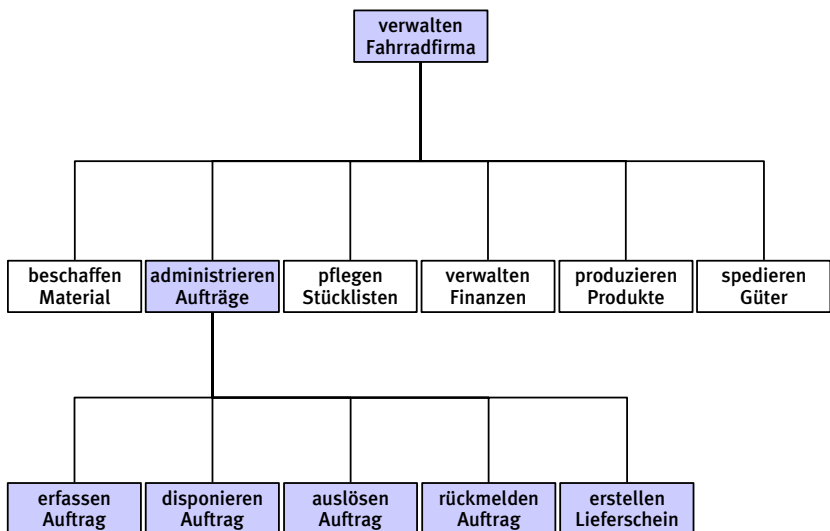


Abb. 17: Der primäre Aspekt betrachtet “direkte Nachfolger”

## 2.6 Teamorientierte Integration der Modelle

Im Rahmen dieses ersten Teils wurde auf den Umstand hingewiesen, dass die Modelle der Matrix untereinander verknüpft sind. Außerdem wurde erwähnt, dass die Diagramme in gewisser Weise an der Diagonale gespiegelt sind.

Zum anderen gibt es aber auch diverse Abhängigkeiten zwischen den Diagrammen innerhalb einer Sicht, egal ob dies die primäre oder sekundäre Sicht betrifft. Beispiel: Das Organisationsmodell enthält eine gewisse Anzahl von Geschäftseinheiten und Abteilungen. Es sind genau diese Stellen, die in den der Aufgabensicht zugeordneten Diagrammen vorkommen dürfen und müssen, sei dies nun im "stellenorientierten Ablaufdiagramm", dem "stellenorientierten Informationsfluss", etc.

Dieser Umstand eröffnet die Möglichkeit, verschiedene Diagramme auf ihre gegenseitige Konsistenz hin zu prüfen und bei Inkonsistenzen laufend Anpassungen vorzunehmen und die Modelle zu verfeinern. Es macht daher wenig Sinn, einen starren Vorgehensplan zum Erarbeiten der Modelle vorzugeben. In der vorangegangenen Darstellung wird einzig ein exemplarischer Vorgehensplan aufgezeigt. Von Bedeutung ist aber eher der konsequente Abgleichungs- und Integrationsprozess.

Ein solcher Integrationsprozess muss insbesondere auch von Teams wahrgenommen werden. Bei größeren Projekten ist es fast unumgänglich, dass verschiedene Teammitglieder an unterschiedlichen Modellen und Diagrammen arbeiten. Der Integrationsprozess wird zu einem wichtigen Faktor eines Projektes. Er kann zu einer gemeinsamen Sprache und einem Teamverständnis führen. Die verschiedenen Ansprüche und Erwartungen lassen sich in diesem Abgleichprozess kommunizieren.

Dies setzt allerdings voraus, dass die unterschiedlichen Personengruppen bereit sind, die Modelle und Vorstellungen der jeweils anderen Sichtweisen offen zu diskutieren. Hier verbirgt sich in der Regel viel Konfliktstoff. Ganz besonders im Grobkonzept werden die unterschiedlichen Blickwinkel auf das komplexe Unternehmenssystem aufeinander prallen.

Jede Gruppe möchte ihr Teilsystem optimieren: Die Techniksicht sucht nach einer möglichst einheitlichen Systemplattform, welche sich analog für das gesamte Unternehmen einsetzen lässt, die Aufgabensicht sucht nach möglichst autonomen Arbeitssystemen und die Prozesssicht ist beispielsweise bestrebt, die Kundenwünsche möglichst schnell und kostengünstig zu befriedigen.



## 2.7 Literatur zur Modellierung von Systemen

- Anderson, J.R. (2001): Kognitive Psychologie: Eine Einführung. (3. Aufl.). Heidelberg: Spektrum der Wissenschaft Verlagsgesellschaft.
- Beer, S. (1962): Kybernetik und Management. Frankfurt am Main: S. Fischer.
- Bertin, J. (1982): Graphische Darstellungen und die graphische Weiterverarbeitung der Information. Berlin: Walter de Gruyter.
- Chestnut, H. (1966): Systems Engineering Tools. New York: John Wiley & Sons.
- Daenzer, W. (Hrsg.) (1976): Systems Engineering: Leitfaden zur methodischen Durchführung umfangreicher Planungsvorhaben. Zürich: Verlag Industrielle Organisation.
- Grün, J., (1995): Führen durch visuelles Management. In: io Management Zeitschrift, 64/10. Zürich: Verlag Industrielle Organisation. 23–27.
- Haberfellner, R., Nagel, P. et al. (2002): Systems Engineering: Methodik und Praxis, 11. Aufl. (Daenzer, W., Huber, F.; Hrsg.). Zürich: Verlag Industrielle Organisation.
- Hall, A. (1978): A Methodology for Systems Engineering. Princeton: Van Nostrand.
- Leont'ev, A.N. (1977): Tätigkeit, Bewusstsein, Persönlichkeit. Stuttgart: Klett.
- Mertens, P., Bodendorf, F. et al. (2004): Grundzüge der Wirtschaftsinformatik. 8. Auflage. Berlin: Springer.
- Müller, R. (1983): Zur Geschichte des Modelldenkens und des Modellbegriffs. In: Modelle – Konstruktion der Wirklichkeit. (Stachowiak, H.; Hrsg.). München: Wilhelm Fink.
- Scheer, A.-W. (1998): Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. Berlin: Springer
- Schönsleben, P. (2000): Integrales Logistikmanagement: Planung und Steuerung von umfassenden Geschäftsabläufen. Berlin: Springer.
- Schönsleben, P. (2001): Integrales Informationsmanagement: Informationssysteme für Geschäftsprozesse – Management, Modellierung, Lebenszyklus und Technologie. Berlin, Heidelberg, New York: Springer.

- Schotten, M. (1998): Aachener PPS-Modell. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 9–28.
- Specker, A. (1993): Die Informatikwelt wird zum sozialen Gefüge: Vom Personal Computing zum Interpersonal Computing. In: Neue Zürcher Zeitung, Nr. 225, ORBIT-Beilage.
- Specker, A. (1997): Kognitives Software Engineering: Ein schema- und scriptbasierter Ansatz. Zürich: vdf Hochschulverlag.
- Sterman, J.D. (2000): Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill.
- von Weizsäcker, C.F. (1991). Wahrnehmung des Denkens. In: Erfahrung des Denkens – Wahrnehmung des Ganzen: Carl Friedrich von Weizsäcker als Physiker und Philosoph (Ackermann, P., Eisenberg, W.; Hrsg). Berlin: Akademie-Verlag.
- Wiener, N. (1948, dt. 1963): Kybernetik. Regelung und Nachrichtenübertragung in Lebewesen und Maschine. Düsseldorf: ECON.
- Zuboff, S. (1988): In the Age of the Smart Machine: The Future of Work and Power. New York: Basic Books.





# 3

## Modellierung der Prozesse

### **Kapitelinhalt**

- Prozesse als sachlogische und zeitliche Abfolge von Operationen
- Prozessmodelle, Prozesshierarchien und Prozess-Segmentierung
- Identifikation, Spezifikation und Messgrößen von Prozessen
- Modellierungsmethoden und Diagramme der Prozesssicht

Wir werden uns in diesem Kapitel der prozessorientierten Sicht zuwenden. Die Prozesssicht untersucht die Operationen eines sozio-technischen Systems nach ihrer zeitlichen und sachlogischen Abfolge.

Ein erster Themenkreis betrifft die Frage, was unter einem Prozess exakt zu verstehen ist. Es wird dargelegt, dass Prozesse hierarchische Strukturen aufweisen. Einerseits können Prozesse im Sinne der Ähnlichkeit segmentiert werden. Andererseits können Prozesse in eine hierarchisch-sequenzielle Struktur gebracht werden. Dies bedeutet, dass Prozesse durch ihre hierarchisch tiefer liegenden Subprozesse vollzogen werden.

In einem zweiten Themenkreis werden wir der Frage nachgehen, wie man die Hauptprozesse eines Unternehmens identifiziert und in welcher Weise diese spezifiziert werden können.

Im dritten Teil werden wir die verbreiteten Modellierungsmethoden zur Prozessbeschreibung näher beleuchten: Wertschöpfungsketten, Flow Charts, etc.

### 3.1 Prozessorientierte Modellierung

#### 3.1.1 Prozesssicht

*Definition: Die Prozesssicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der sachlogischen und zeitlichen Abfolge der Operationen.*

*Definition: Prozesse bezeichnen zusammengehörige und in sich relativ geschlossene Einheiten von Operationenabfolgen der unterschiedlichen Systemniveaus.*

Die nachfolgende Darstellung illustriert diese Definition: ein “Beschaffungsprozess” setze sich aus der Abfolge “Bestellungsprozess”, “Wareneingangsprozess” und “Zahlungsprozess” zusammen (s. Abb. 18). Jeder Teilprozess stellt wiederum selbst eine Abfolge von Subprozessen bzw. Operationen dar. “A process is thus a specific ordering of work activities across time and place (...): a structure for action” (Davenport 1993, S. 5).

Ein Prozess integriert damit auch Operationen, welche in der funktionalen Sichtweise nicht zusammengefasst werden würden, da sie funktional nicht ähnlich sind (s. Kapitel 4). Hier liegt auch der große Unterschied zwischen einem Prozess “Beschaffung” und einer sehr ähnlich lautenden Hauptfunktion “beschaffen Material”. Es ist also ein Merkmal von Prozessen, dass Funktionsbäume horizontal “durchquert” werden.

#### Operativer Beschaffungsprozess

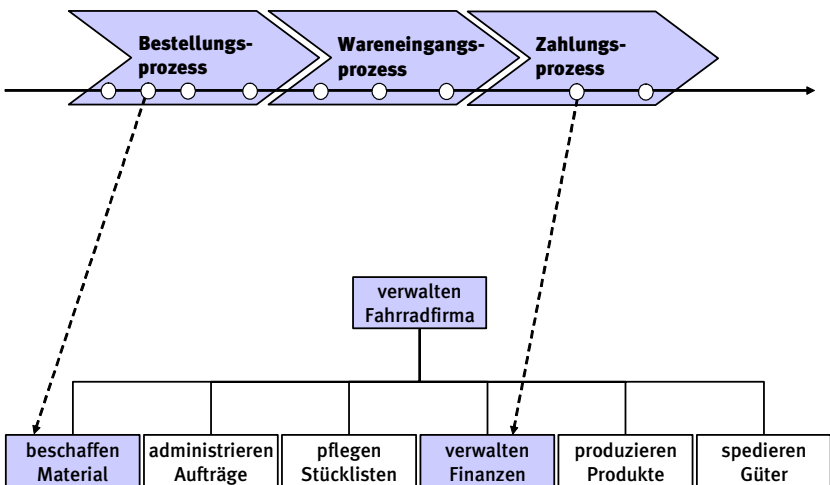


Abb. 18: Prozesssicht als sachlogische Abfolge von Operationen



### 3.1.2 Systeme sind ein komplexes Operationennetzwerk

Prozesse verlaufen nun in der Realität in aller Regel nicht dermaßen linear ab, wie es das vorhergehende Beispiel suggeriert. Vielmehr sollte ein sozio-technisches System als ein komplexes Aktivitäten- und “Operationennetzwerk” verstanden werden, bei welchem zwischen einzelnen Operationen und Prozessen unzählige Beziehungen bestehen. Die nachfolgende Darstellung soll diese Sichtweise illustrieren (s. Abb. 19).

Prozesse eines sozio-technischen Systems laufen in der Regel zudem nie exakt gleich ab. Sie können demzufolge selten bis ins letzte Detail formalisiert werden. Wie können nun Prozesse unter dieser Voraussetzung aber einer Analyse und Modellierung zugeführt werden? Wie soll es gelingen, Prozesse formal darzustellen und abzubilden?

Eine klare Modellierung von Prozessen kann wiederum nur gelingen, wenn man die Vielfalt der Prozesse nach ihren strukturellen Ähnlichkeiten untersucht. Trotz individueller Eigenheiten eines jeden Prozesses werden gewisse Operationenabfolgen und Teilprozesse durchaus gewisse schematisierbare und wiederkehrende Strukturen aufweisen. Wenn in der Folge also von “Prozessen” oder “Geschäftsprozessen” gesprochen wird, dann geht man von solcherart schematisierten Betrachtungen aus. Es entspricht in der Tat der natürlichen Kognition des Menschen, Handlungsabläufe trotz individueller Eigenschaften mental als einheitlichen Prozess wahrzunehmen (Schank & Abelson 1977).

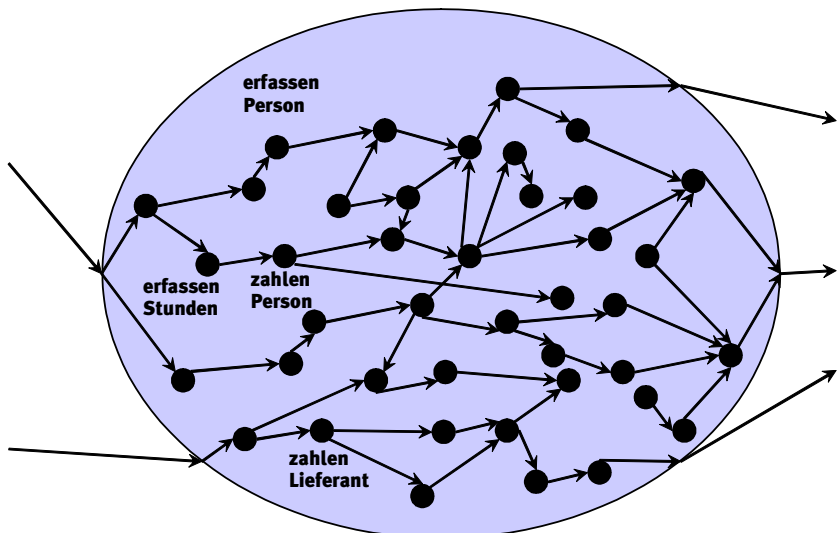


Abb. 19: System als Netzwerk von Operationen in Raum und Zeit



### 3.2

## Prozessorientierte Methoden im Überblick

Im Sinne eines Überblickes seien an dieser Stelle nochmals die verschiedenen Methoden angeführt, die in weitergehender Weise der prozessorientierten Spezifikation zugeordnet werden können. Wir grenzen zunächst vier Prozessmodelle ab, welche gemäß unserer Definition der prozessorientierten Modellierung zuzuordnen sind, insofern als sie alle den Prozess als primären Aspekt darstellen. In der unten stehenden Matrix sind sie in der Vertikale schattiert unterlegt (s. Abb. 20).

Sodann gibt es eine Reihe von Modellierungsmethoden, die die Prozesse zwar ebenfalls beinhalten, allerdings den Prozess lediglich als sekundären Aspekt aufnehmen. Gemeinsam spannen diese Methoden den Raum der prozessorientierten Darstellungen auf. Dies bedeutet auch, dass all diese Modelle in konsistenter Weise voneinander abhängen müssen.

Im diesem Kapitel werden nur die schattierten Methoden behandelt, also die prozessorientierten Methoden. Die weiteren Methoden (Prozess als sekundärer Aspekt) werden in den nachfolgenden Kapiteln behandelt.

Im Zentrum der prozessorientierten Analyse steht zunächst die Erarbeitung eines Prozessmodells, d.h. die "reine" Prozessbetrachtung. Dieser Analyse wird im nächsten Abschnitt viel Raum gegeben. Anschließend werden wir der Reihe nach auf die anderen Methoden eingehen.

		Hauptfokus ↓		
primär \ sekundär	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht	Prozessmodell	Funktionen-Blockdiagramm	Zustandsübergangsdiagramm	Stellenorientierter Informationsfluss
Funktionssicht	Flussdiagramm			
Objektsicht	Sequenzdiagramm			
Aufgabensicht	Stellenorientiertes Ablaufdiagramm			

Abb. 20: Primäre und sekundäre Modellierung der Prozesse

### 3.3 Prozessmodell

*Definition: Prozessmodelle stellen die hierarchischen Strukturen und Beziehungen der Prozesse eines Systems dar.*

Prozesse weisen entsprechend den allgemeinen Eigenschaften von Teilsystemen ebenfalls hierarchische Strukturen auf. Auf einer ersten Ebene könnte das System als ein einziger “Gesamtsystemprozess” betrachtet werden (s. Abb. 21).

Dieser Gesamtsystemprozess wird durch die darunterliegenden Hauptprozesse vollzogen. Auf der zweiten Ebene können daher “Hauptprozesse” (engl.: “Major processes”, Davenport 1993) identifiziert werden, analog zu “Hauptfunktionen”. Diese werden alternativ häufig auch als so genannte “Geschäftsprozesse” bezeichnet.

Ausgehend von den Hauptprozessen können diese weiter auf ihre Hierarchie der Teilprozesse hin analysiert und stufenweise weiter detailliert werden – bis hin zu den “atomaren” Prozessen bzw. Operationen.

Fazit: Ein sozio-technisches System lässt sich in der Prozesssicht als eine Anzahl von “relativ isolierten Prozessen” beschreiben. Diese Prozesse bilden die Grundlage für alle weiteren Betrachtungen der Prozesssicht.

Das Prozessmodell wird häufig auch in Form einer so genannten “Prozesslandkarte” dargestellt. Diese beschränkt sich auf die Visualisierung der Geschäftsprozessebene.

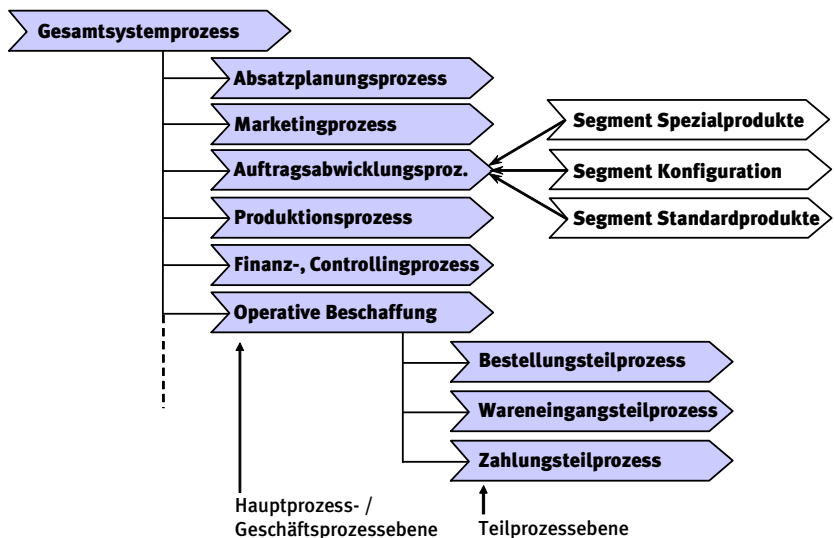


Abb. 21: Struktur eines Prozessmodells

### 3.3.1 Bestandteilhierarchie: Hierarchisch-sequenzielle Bearbeitungsstruktur

Betrachten wir eine solche Prozesshierarchie am Beispiel des Beschaffungsprozesses. Wie die unten stehende Darstellung aufzeigt, lässt sich der “Beschaffungsprozess” auf der zweiten Ebene in drei Teilprozesse unterteilen und gewissermaßen durch diese vollziehen. Prozesse werden mit anderen Worten “hierarchisch-sequenziell” durchlaufen (s. Abb. 22).

Der Teilprozess “Bestellungsprozess” wird sodann wiederum durch die vier darunterliegenden Subprozesse vollzogen. Das bedeutet, dass zur Bearbeitung eines Beschaffungsprozesses zunächst der Bestellprozess abgehandelt wird. Hierzu müssen aber erneut dessen vier Subprozesse beendet bzw. die relevanten Teilziele erreicht werden. Erst danach erfolgt die Bearbeitung des Wareneingangsprozesses.

Diese Art der Hierarchie unterscheidet sich in struktureller Hinsicht zwar nicht wesentlich von den Funktions-, Objekt- und Aufgabenhierarchien (s. u.) – bei allen Aspekten lassen sich Bestandteilhierarchien ausmachen.

In einem Punkt unterscheidet sich die Prozesshierarchie fundamental von den anderen Aspekten: sie wird als einzige Hierarchie in einer (relativ) vorgegebenen Abfolge, eben hierarchisch-sequenziell, durchlaufen.

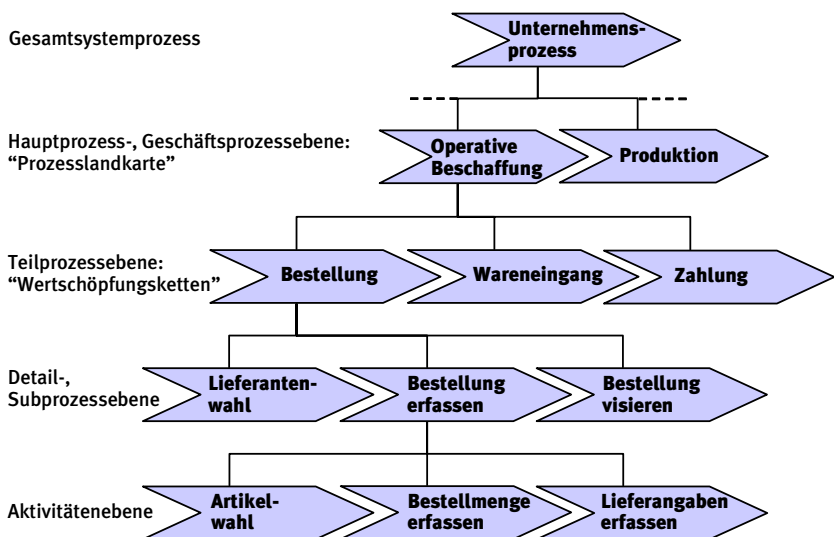


Abb. 22: Hierarchisch-sequenzielle Prozesshierarchie

### 3.3.2 Vererbungshierarchie von Prozessen: Segmentierung

Bei der prozessorientierten Modellierung spielt ein zweiter Begriff eine wichtige Rolle: der Begriff der “Segmentierung”. Dieser drückt aus, dass gewisse Hauptprozesse über ähnliche Grundstrukturen verfügen (z.B. Eigenschaften der Auftragsabwicklung), trotzdem aber unterschieden werden müssen. Dies entspricht der oben erwähnten Vererbungs- bzw. Spezialisierungshierarchie. In der Praxis kommt der Segmentierung von Auftragsprozessen eine wichtige Rolle zu. Mögliche Segmentierungsansätze bestehen nach folgenden Gesichtspunkten:

- Menge bzw. Wiederholfrequenz (ABC-Analyse)
- Kundengruppen: z.B. institutionelle und private Investoren
- Eingangskanal des Kunden (Internet, Schalter)
- Region (vielleicht Sprache)
- Produkten (Spezialfahrräder, Konfiguration, Standard, etc.)
- Produktklasse (Tische, Stühle, etc.)

Nachfolgend ist das Beispiel einer Firma dargestellt, welche ihren Auftragsabwicklungsprozess nach der Wiederholfrequenz der Produkte segmentiert hat (s. Abb. 23). Sie unterscheidet drei zentrale Auftragsprozesse, welche aufgrund ihrer Häufigkeit unterschiedlich gefertigt werden.

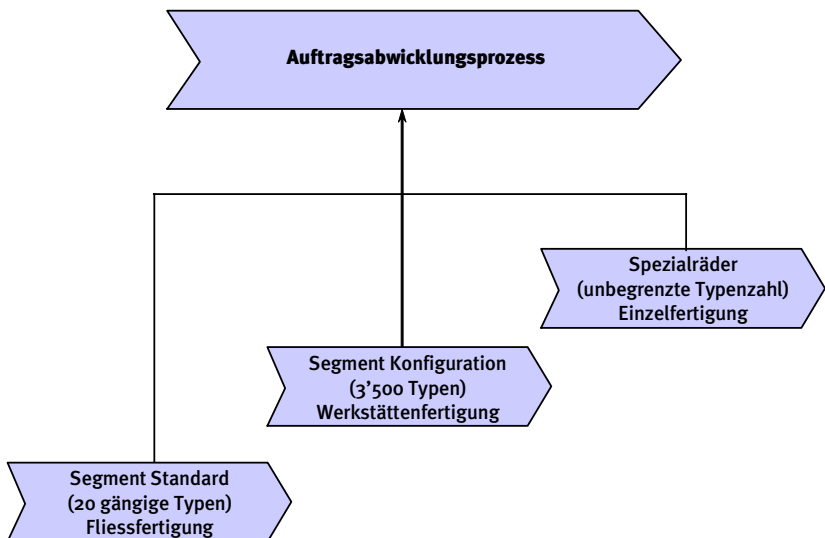


Abb. 23: Segmentierung (Spezialisierung) von Prozessen

### 3.3.3 Prozesslandkarte

Es ist ein Merkmal von vielen Prozessmodellen, dass sie über recht viele Hauptprozesse verfügen ("flache Hierarchie"). Da der Umfang eines Prozessmodells größere Ausmaße annehmen kann – bereits bei den Hauptprozessen muss man von rund 10 bis 20 Prozessen ausgehen – lässt sich dieses häufig nicht als Ganzes mit allen Hierarchieebenen abbilden. Man wird daher den Weg gehen, verschiedene Modellierungsebenen separat zu betrachten und mit einer so genannten "Prozesslandkarte" zu beginnen (s. Abb. 24).

Häufig wird auch eine weitere Gruppierung der Hauptprozesse vorgenommen, wobei man hier vielfach mit einer Gruppe "Kundenprozesse" beginnt. In einem zweiten Schritt kann man der Frage nachgehen, welche Hauptprozesse ihrerseits die "Kundenprozesse" unterstützen. Beispielsweise könnte der Prozess der "Strategischen Beschaffung" unabhängig vom konkreten Auftragsprozess wahrgenommen und als solcher definiert werden. Sodann gilt es auch die vierte Gruppe zu beachten – die Management- und Supportprozesse. Diese haben einen systemerhaltenden Charakter: Entlohnung des Personals, Sicherstellung der Informatik, Strategie, etc. Letztlich muss auch noch geklärt werden, welche Prozesse in der Systembetrachtung mit eingeschlossen werden. Hauptprozesse lassen sich damit in ihrer Abhängigkeit in einer so genannten "Prozesslandkarte" mit ihren Beziehungen untereinander darstellen.

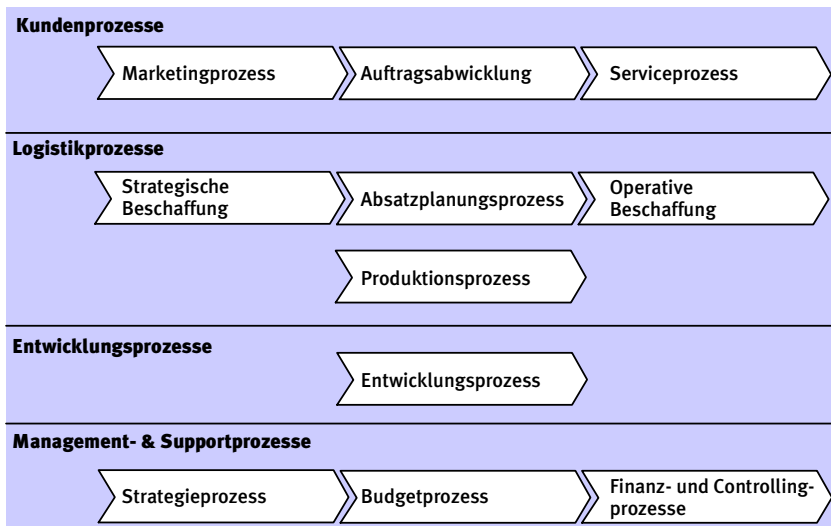


Abb. 24: Die Prozesslandkarte als oberste Ebene des Prozessmodells

### 3.4 Identifikation und Systemabgrenzung der Prozesse

In der Praxis stellt sich zunächst meist die konkrete Frage, wie man zu den Hauptprozessen eines gegebenen Systems gelangt bzw. diese identifiziert. Leider kann man nur in den wenigsten Fällen auf ein existierendes Prozessmodell der zu analysierenden Firma zurückgreifen, da nach wie vor nur recht wenige Unternehmen über ein solches verfügen.

Gesucht sind zunächst diejenigen Hauptprozesse, welche das gegebene System im Sinne von "relativ isolierten Prozesssystemen" strukturieren und demnach *mit möglichst wenigen Schnittstellen* auskommen. In aller Regel muss diese Analyse infolge der großen Menge der Operationen eher Top-down erfolgen. Die Operationsabfolgen der Hauptprozesse "durchqueren" das System und enden erst an den Systemgrenzen – mit anderen Worten, bei externen bzw. internen "Kunden".

Daher kann man sich beispielsweise der Technik der Systemabgrenzung (Black-box) bedienen und auf das Kontextdiagramm mit einer vollständigen Liste der In- und Outputs des Systems zurückgreifen (s. Abb. 25). Jede Beziehung zu den Umsystemen (z.B. zu den Kunden, Lieferanten) muss nämlich zwingend durch einen Geschäftsprozess "abgefangen" werden. Einer Kundenanfrage liegt demnach ein anderer Prozess zugrunde als einem Stellengesuch. In einem weiteren Schritt können die Prozesse zusammengefasst und bereinigt werden. Daraus resultieren die Geschäftsprozesse bzw. Hauptprozesse.

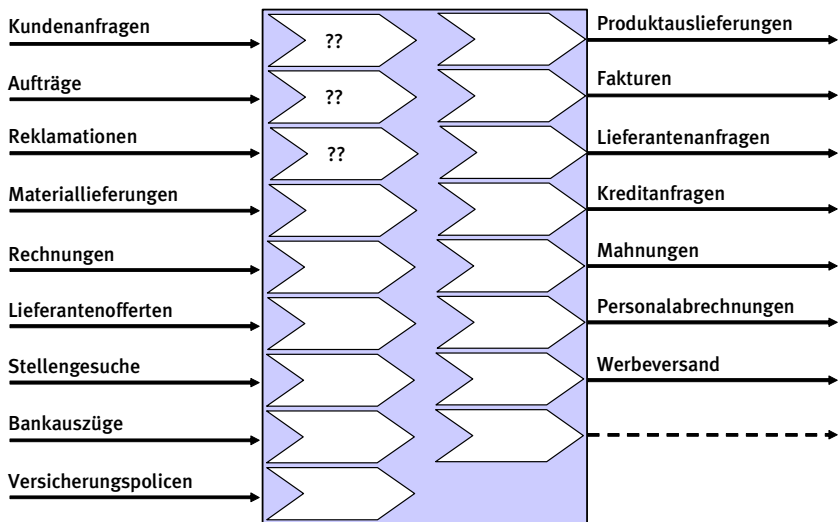


Abb. 25: Kontextbetrachtung zur Identifikation von Hauptprozessen

### 3.4.1 Systemabgrenzung

Sind die Geschäftsprozesse gemäß obigem Vorgehen bezeichnet, so werden sie im Rahmen der prozessorientierten Modellierung näher beschrieben bzw. klar spezifiziert werden. Zur Spezifikation eines Prozesses gehört es zunächst, ihn von seinen "Umsystemen" im Netzwerk abzugrenzen. Dies bedeutet, dass die Beziehungen zu anderen Prozessen aufzuzeigen sind. Prozesse werden definitionsgemäss dahingehend strukturiert, dass sie nur über möglichst wenige Schnittstellen verfügen. Sie lassen sich daher als Teilsysteme im Sinne einer Black-box verstehen und mit klar definierten In- und Outputs beschreiben. In einem ersten Schritt gehört es daher zur Spezifikation eines Prozesses, die relevanten In- und Outputs aufzulisten (s. Abb. 26).

Eine weitere Form der In- und Outputs sind sodann die Beziehungen zu Prozessen, die angestoßen oder von denen Leistungen bezogen werden können. Häufig stellt sich in der Praxis die Frage, ob In- und Outputs, welche innerhalb des Prozesses generiert werden, ebenfalls erfasst werden sollen. Im Sinne der Systembetrachtung müssen aber je nach Systemebene nur diejenigen In- und Outputs angeführt werden, die über Schnittstellen zu anderen Prozessen verfügen. Zur Modellierung eines Prozesses gehört weiter, dass die Zielsetzungen und Messgrößen an den Prozess klar definiert werden. Messgrößen betreffen den Output des Prozesses oder vielleicht das Verhältnis von Input zum Output.

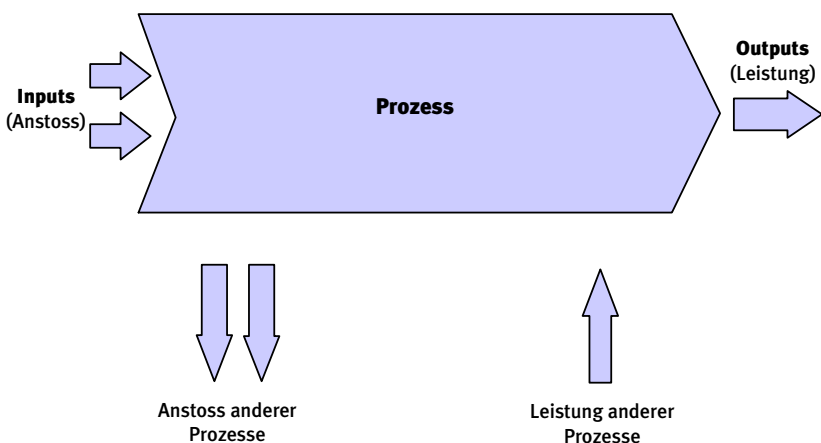


Abb. 26: Prozessspezifikation als relativ geschlossenes System

### 3.5 Spezifikation von Prozessen: Wertschöpfungsketten

Wie im letzten Abschnitt ausgeführt, lassen sich Prozesse hierarchisch in Teilprozesse gliedern. Diese können gemäß der Notation der “Wertschöpfungskette” durch aufeinanderfolgende Pfeile dargestellt werden (s. Abb. 27).

Die einzelnen Elemente der Wertschöpfungskette sind Prozesse (und nicht Funktionen). Auf dieser Ebene bestehen meist keine Parallelitäten, bzw. man verzichtet der Einfachheit halber auf ihre Darstellung. Selbstverständlich könnte diese lineare Form auch um Parallelitäten, Regelbedingungen und Beziehungen erweitert werden.

Bei der Detaillierung eines Prozesses mit seinen Teilprozessen ist man mit der Frage konfrontiert, wie man die Teilprozesse identifiziert. Ein Ansatz zur Lösung dieses Problems besteht darin, nach den Teilzielen bei der Abwicklung des Geschäftsprozesses zu fragen. Teilziele könnten beispielsweise sein: Die Bestellung ist erfolgt, Mahnung durchgeführt, Ware geprüft, etc.

Ein häufiger Fehler bei der Erstellung eines Wertschöpfungsketten-Diagrammes besteht darin, dass die Hauptprozesse nicht wirklich “end-to-end” modelliert werden. Ein Beschaffungsprozess ist beispielsweise nicht mit dem Wareneingang abgeschlossen. Erst wenn auch die Zahlung erfolgt ist, kann der Prozess in aller Regel als beendet gelten. Deshalb gehört ein entsprechender Teilprozess an das Ende dieser Prozesskette.

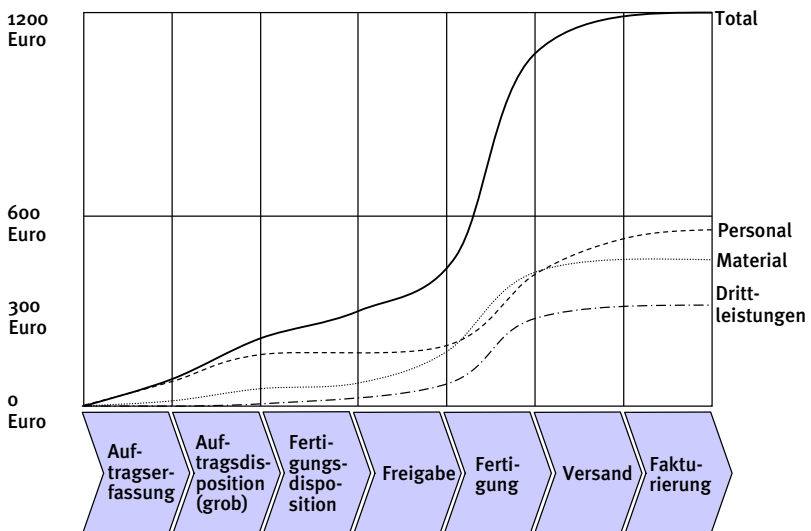


Abb. 27: Analyse der Wertschöpfungskette



### 3.5.1 Prozessregeln

Wie oben erwähnt, verläuft in der Realität kein Prozess absolut gleich einem zweiten. Es wäre daher geradezu unmöglich, jeden auftretenden Prozessfall bis in alle Einzelheiten zu modellieren. Die Charakterisierung von Prozessen und ihrer Vielfalt erfolgt daher häufig aufgrund der expliziten Angabe von Prozessregeln. Sie dienen der Präzisierung und Fallunterscheidung. Regeln eignen sich z.B. zur Darstellung von:

- Alternativen (wenn Prozess A oder B infrage kommt)
- Ausnahmen (falls Typ A, dann überspringe den Prozessschritt)
- Geschäftsregeln (falls Mahnstufe 3, dann "Einschreiben")

Die Modellierung des Regelwissens ist auch Gegenstand der Gestaltung von Expertensystemen. Es hat sich aber dabei gezeigt, dass die Komplexität des Regelwissens ausgesprochen hoch sein kann und sich die Gesamtkomplexität des Systems trotz großer formaler Anstrengungen nicht fassen lässt.

An dieser Stelle wird daher dafür plädiert, die Angabe von Regeln gewissermaßen zur informellen Information zu verwenden und in einem phasengerechten Ausmaß zu erfassen.

Regeln verknüpfen häufig verschiedene Aspekte. Beispiel: Im Fall eines Kunden der Kundengruppe A (Objekt) beträgt der Rabatt auf dem Handelssortiment fünf Prozent und auf der Eigenproduktion 10 Prozent. Der Kunde erhält in diesem Fall keine Auftragsbestätigung (Prozess) und bezahlt mit 30 Tagen Zahlungsfrist.

Solche Regeln können Informationssysteme beliebig kompliziert werden lassen. Es wird daher empfohlen, sich auf die 80:20 %-Regel abzustützen (Pareto-Prinzip) und nur Regeln zu modellieren, die wirklich geschäftskritisch sind.

Mit anderen Worten wird man sich im Verlauf der Spezifikation insbesondere auf Kundenprozesse und den Auftragsabwicklungsprozess konzentrieren und dort diejenigen Regeln erfassen, welche jeden Tag unzählige Male zur Anwendung gelangen.

### 3.5.2 Messgrößen

Die “end-to-end”-Durchgängigkeit von Geschäftsprozessen – vom Kunden zum Kunden – bringt bei genauer Analyse eine Vielzahl von Informationen und Vorteile bei der Beurteilung der Unternehmensleistung. So lassen beispielsweise neben der Wiederholungshäufigkeit eines Prozesses auch Informationen über die mittlere Gesamtdurchlaufzeit eines Geschäftsprozesses Rückschlüsse auf die Zeiteffizienz des Unternehmens zu. Prozess-Kenngrößen überwinden die isolierte Beurteilung von einzelnen funktionalen Unternehmenseinheiten und stellen die integrierte *Gesamtleistung* des Unternehmens in den Mittelpunkt.

#### Strategische Vorgaben und Überprüfung der Prozessmessgrößen

Bei der Analyse und Gestaltung von Prozessen werden idealerweise sogleich auch Messgrößen und Kennzahlen für die Hauptprozesse definiert und mit der Strategie in Einklang gebracht.

Messgrößen von Geschäftsprozessen dienen Unternehmen somit einerseits zur Überprüfung der Strategie oder aber andererseits als strategische Zielvorgaben bei einer nötigen Neugestaltung des sozio-technischen Systems. In diesem Sinne bilden die Messgrößen einen Brückenschlag zwischen Strategie und operativer Systemgestaltung.

Welches sind die strategischen Zielsetzungen, die mit der Neugestaltung eines Prozesses erreicht werden sollen? Auf Prozesse können üblicherweise unzählige Kenngrößen und Ziele angewendet werden.

#### Kritische Erfolgsfaktoren

Die inhärente Kundenorientierung von Geschäftsprozessen ermöglicht die Ableitung der relevanten Ziele und Kenngrößen aufgrund von so genannten “kritischen Erfolgsfaktoren”. Kritische Erfolgsfaktoren sind jene Größen, die im Markt und bei den Kunden als besonders wichtig angesehen werden und den Erfolg des “Geschäftsprozesses” bedingen. Diese Größen können für alle zu betrachtenden Prozesse erhoben werden, z.B.:

- Leistung / Qualität / Durchlaufzeit
- Kosten
- Flexibilität

Je Geschäftsprozess sind daher die wirklich kritischen Größen zu definieren. So wird sichergestellt, dass diese später als erfolgreich beurteilt werden.

### 3.5.3 Zusammenfassende Charakterisierung von Prozessen

Betrachten wir nun, wie die oben angeführten Merkmale und Eigenschaften zur effizienten Charakterisierung eines Prozesses herbeigezogen werden könnten. Es handelt sich im nachfolgenden Beispiel um die Beschreibung des Auftragsgewinnungsprozesses eines Softwarehauses (s. Abb. 28).

Jeder Prozess kann z.B. der Übersichtlichkeit halber durch eine einzige A4-Seite beschrieben werden. In der Mitte des Blattes ist der Prozess mit seiner Zielsetzung dargestellt. Auf den Geschäftsprozess wirken die Inputs von links und die Outputs nach rechts. Im oberen Teil der Darstellung wird vermerkt, welche Kenngrößen als Messgrößen dienen. Der mittlere Teil enthält die Beschreibung des Inhaltes des Geschäftsprozesses. Der unten stehende Teil enthält Hilfsmittel und bezeichnet den oder die entsprechenden Prozessowner. Selbstverständlich kann man sich auch eine veränderte und erweiterte Struktur zur Beschreibung von Prozessen als Schablone festlegen. Soviel vorerst zur Beschreibung des Prozessmodells.

Wir werden uns nun dem weiteren Aspektdarstellungen von Geschäftsprozessen zuwenden und drei Methoden zur Modellierung von Prozessen näher ansehen: das Flussdiagramm (Flow Chart), das Sequenzdiagramm und das stellenorientierte Ablaufdiagramm. Abschließend wird eine integrierte prozessorientierte Methode vorgestellt (ARIS).

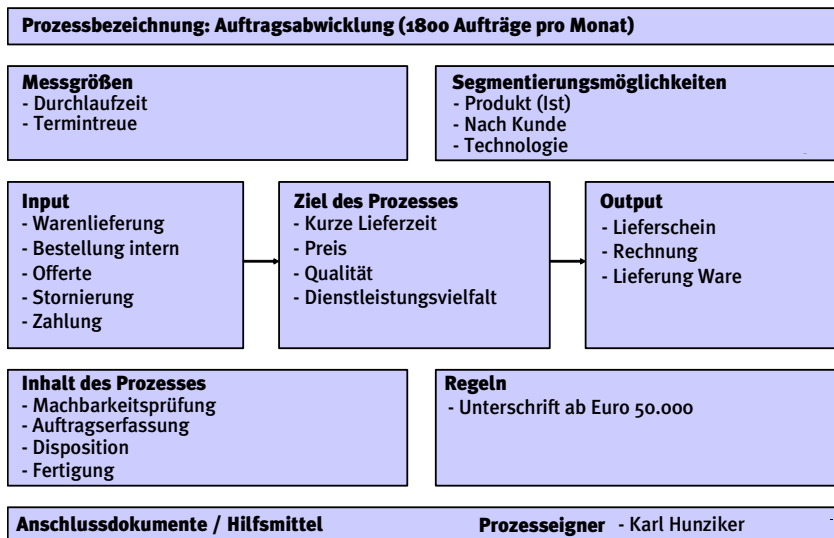


Abb. 28: Charakterisierung eines Prozesses

### 3.6 Flussdiagramm

*Definition: Flussdiagramme stellen primär einen Prozess dar und sekundär die diesem Prozess zugrunde liegenden Funktionen.*

Klassische Flussdiagramme zeigen in den Rechtecken *Funktionen* (Scheer 2001). Sie bieten die Möglichkeit, *funktionale Anforderungen* an Informationssysteme zu erheben, da sie Prozesse mit Funktionen verknüpfen (s. Abb. 29). Flussdiagramme enthalten gemäß DIN- (66001) bzw. ISO-Norm neben dem Symbol für “Funktionen” zusätzliche Symbole, z.B. für “Entscheidungen” (Rhombus) und für “Nahtstellen” zu anderen Prozessen (Kreise) (Chapin 1970). Mit Hilfe des Rhombus können Prozessalternativen durch ein einzelnes Diagramm abgebildet werden. Nicht systemgestützte Tätigkeiten können speziell markiert werden.

Die dargestellten Funktionen sind entweder “höhere Funktionen” des Funktionsmodells oder Elementarfunktionen (s. Abschn. 2.2.4). Bei Elementarfunktionen verschwindet interessanterweise der Unterschied zu Prozessen, da nun eine Operationenfolge dargestellt wird. Auf der untersten Granularitätsebene der Systemanalyse fallen demnach Prozess-, Fluss- und Sequenzdiagramme vollständig zusammen. Die Elemente zeigen sowohl Operationen und Elementarfunktionen als auch Objekte. Um Missverständnissen vorzubeugen, sollte man Funktionen und Prozesse mit verschiedenen Symbolen versehen.

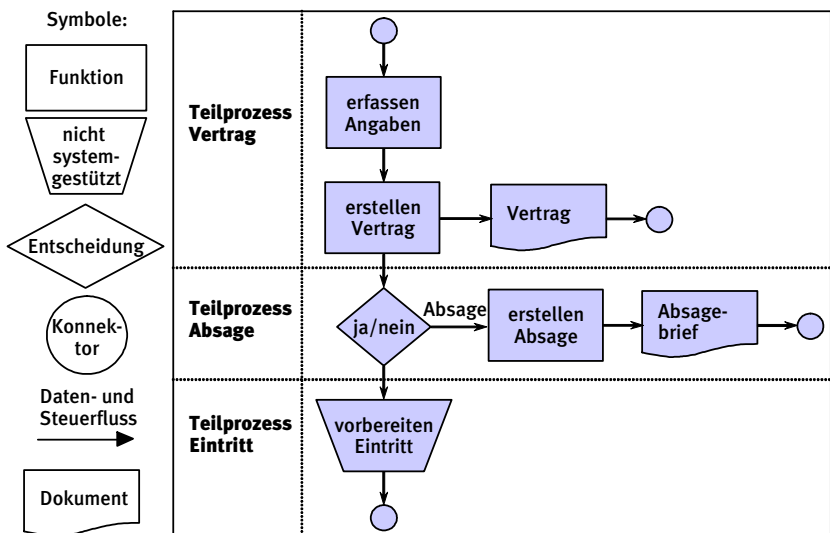


Abb. 29: Flussdiagramm

### 3.7 Sequenzdiagramm

*Definition: Sequenzdiagramme stellen primär einen Prozess dar und sekundär die von diesem Prozess tangierten Objekte.*

Diese Darstellung zeigt das Sequenzdiagramm einer Auftragsabwicklung: auf vertikalen Linien sind Objekte vermerkt, die im Verlauf des Prozesses eröffnet werden oder schrittweise Veränderungen erfahren (s. Abb. 30).

Objekte spielen in sozio-technischen Systemen eine zentrale Rolle. Leider werden Sequenzdiagramme trotzdem nur sehr beschränkt zur Systemspezifikation verwendet. Die in der Literatur anzutreffenden Beispiele richten sich meist an implementierungsnahe Phasen (z.B. Burkhard 1997). Sequenzdiagramme könnten aber auch in den frühen Phasen sinnvoll vermehrt genutzt werden. So ließe sich beispielsweise im Rahmen einer Evaluation prüfen, ob Auftragsstrukturen (Objekte) korrekt abgebildet werden können und ob auf den Auftragsobjekten die notwendigen "Transformationen" tatsächlich möglich sind. So werden im unten stehenden Beispiel aus dem Kundenauftrag und den Vertriebsartikeln automatisch zwei Produktionsaufträge generiert werden müssen. Diese Transformation könnte dann Schwierigkeiten bereiten, wenn solche Objekttransformation nach sehr unternehmensspezifischen Regeln erfolgen müssen.

Anstelle des hier dargestellten Geschäftsprozesses könnte man auch lediglich einen ausgewählten Teilprozess mit Subobjekten darstellen. Bei Diagrammen lässt sich also entweder die primäre, die sekundäre oder beide Dimensionen gleichzeitig hierarchisch vertiefen.

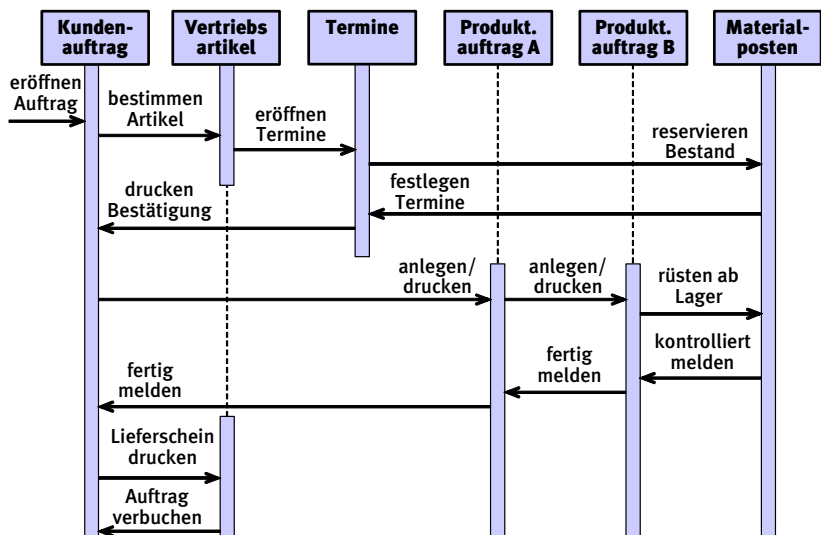


Abb. 30: Sequenzdiagramm

### 3.8 Stellenorientiertes Ablaufdiagramm

*Definition: Stellenorientierte Ablaufdiagramme stellen primär einen Prozess dar und sekundär die von diesem Prozess tangierten Stellen.*

Beim stellenorientierten Ablaufdiagramm werden in den vertikalen Bahnen (engl.: "Swim lanes") Stellen bzw. Aufgaben eingezeichnet (s. Abb. 31). Diese werden von Teilprozessen tangiert. Die Zeitachse ist von links nach rechts und von oben nach unten gerichtet.

Die Parallelitäten zum Flussdiagramm und zum Sequenzdiagramm werden offensichtlich. Bei allen Darstellungen werden in der Horizontalen die Elemente des sekundären Aspekts (Funktion, Objekt bzw. hier Stelle) dargestellt. Für einen ausgewählten Prozess wird also abgebildet, in welcher Weise er die jeweils sekundären Aspekte tangiert.

Die Bedeutung des stellenorientierten Ablaufdiagramms bei der Modellierung von Informationssystemen liegt nun darin, dass es die Aufbau- und die Ablauforganisation verknüpft und integriert darstellt.

Im Gegensatz zum Prozessmodell oder der Prozesslandkarte weist es allerdings bereits einen erheblich verfeinerten Detaillierungsgrad auf und wird aus diesem Grund häufig erst im Detailkonzept eingesetzt. Selbstverständlich könnte man eine relativ ähnliche Darstellungsform auf einer höheren Systemebene vornehmen und lediglich Teilprozesse eintragen.

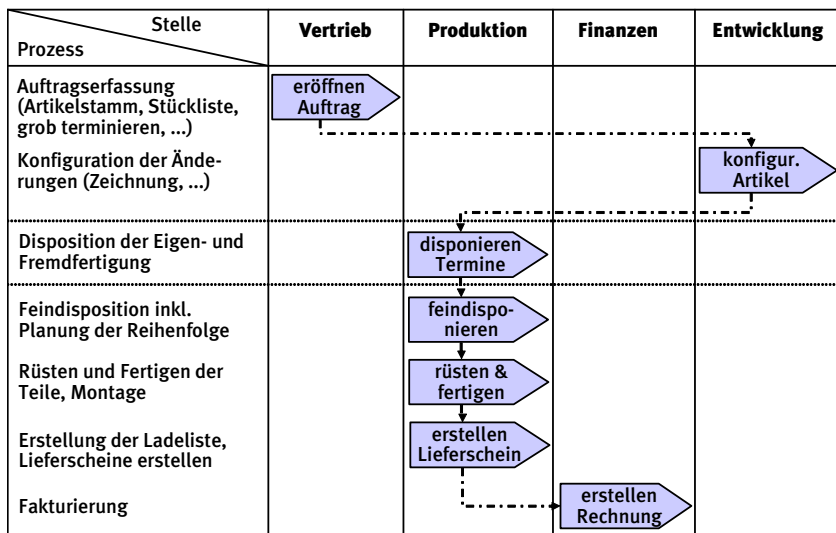


Abb. 31: Stellenorientiertes Ablaufdiagramm

### 3.8.1 Historischer Rückblick

Das stellenorientierte Ablaufdiagramm lässt sich zumindest auf den Beginn des letzten Jahrhunderts zurückverfolgen (Grull 1928). Es ist davon auszugehen, dass von dieser Darstellung gerade auch zu den Zeiten des so genannten "Scientific Management", welches später auch unter dem Begriff des "Taylorismus" bekannt wurde, solche Methoden entwickelt und angewendet wurden.

Die nachfolgende Reproduktion zeigt den Ausschnitt eines stellenorientierten Ablaufdiagramms aus dem Jahre 1911, bezeichnet als den "Gang der Werkstattsaufträge" (s. Abb. 32).

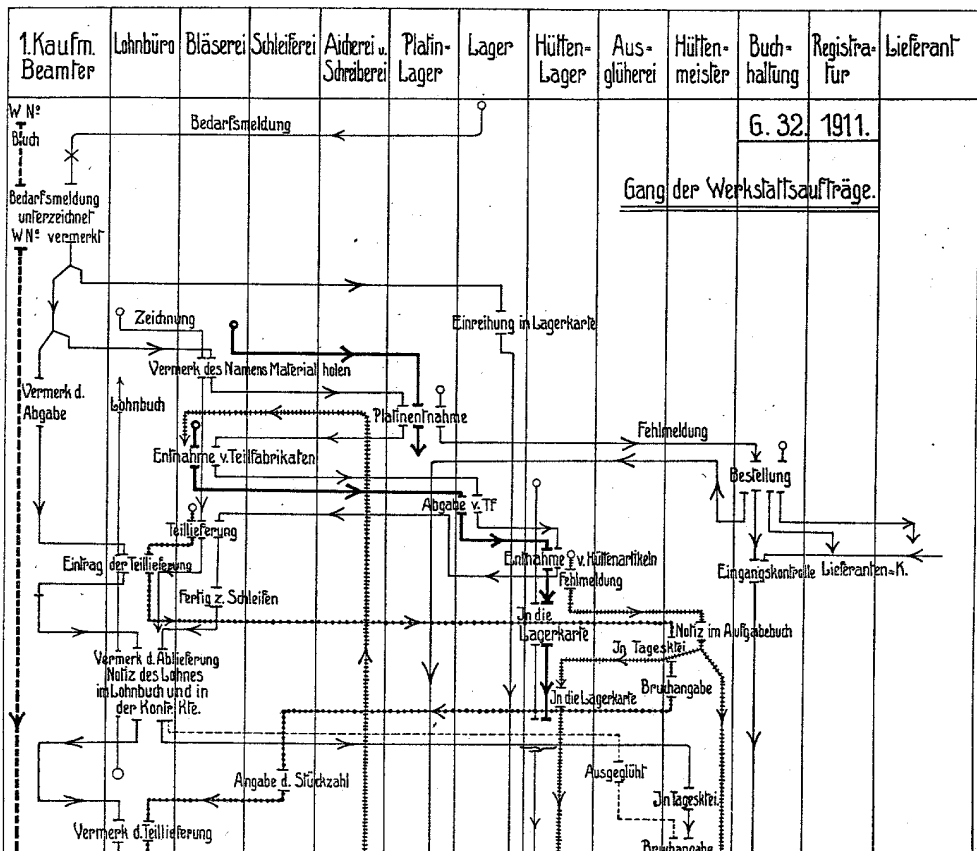


Abb. 32: Stellenorientiertes Ablaufdiagramm (Grull 1928)

### 3.8.2 Erweiterte Formen des stellenorientierten Ablaufdiagramms

Die Art der Darstellung des stellenorientierten Ablaufdiagramms kann in der Literatur variieren und weitere Aspekte einbeziehen. In der nachfolgenden Darstellung sind z.B. anstelle von Subprozessen in Form von Rechtecken "Funktionen" vermerkt (s. Abb. 33).

Gelegentlich werden zusätzlich "Objekte" in einer eigenen Spalte rechts außen eingetragen. Alternativ könnte auch das unterstützende Informationssystem dargestellt werden. Solche Darstellungen bieten einen *guten Überblick* über einen Prozess. Es werden mehrere Aspekte in ihrem Zusammenhang dargestellt. Es besteht allerdings die Gefahr, dass "zuviel" in eine Abbildung "hineingepackt" wird und nicht mehr klar ist, welche Elemente analytisch sauber modelliert sind oder ob gewisse Elemente mehr "informeller" Natur sind.

Allerdings geht durch diese Art der Darstellung die formale Klarheit verloren. Die Abgrenzung zwischen Funktionen und Prozessen ist wie oben erwähnt schwierig und wird daher auch meist unsauber vorgenommen.

Jede Funktion kann ja bei vertiefter Analyse und Detaillierung später wieder als Prozess aufgefasst werden. Es ist daher vielleicht eher empfehlenswert, auf den Gebrauch von rechteckigen Funktionssymbolen zunächst zu verzichten und alle Elemente als Subprozesse in Pfeilform darzustellen.

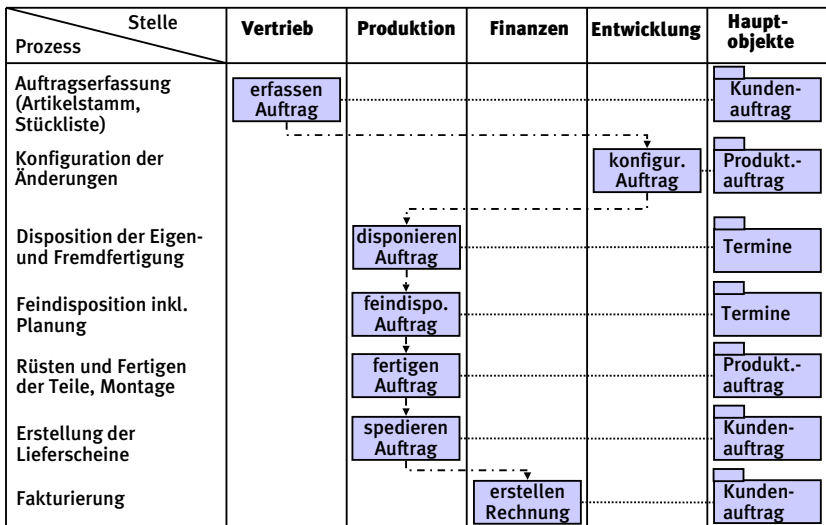


Abb. 33: Erweitertes stellenorientiertes Ablaufdiagramm



### 3.8.3 Darstellung von Auftragskaskaden

Eine abgewandelte Form zur Darstellung von Prozessen zeigt die unten stehende Darstellung (s. Abb. 34). In dieser Form werden die am Prozess beteiligten Stellen in der Vertikalen und in der Horizontalen die Teilprozesse in Form von Wertschöpfungspfeilen eingezeichnet (Zeitachse). Das stellenorientierte Ablaufdiagramm ist damit um 90 Grad gedreht und beide Darstellungen können im Grunde dieselben Informationen enthalten.

Die vorliegende Form der Darstellung findet idealerweise dann Verwendung, wenn zwischen den Stellen Kunden-Lieferanten-Beziehungen im Sinne von Auftragsverhältnissen bestehen (Tipotsch 1997; Schönsleben 2000). Der Vorteil der Notation liegt darin, dass sich mit dem Symbol der Wertschöpfungspfeile gut ausdrücken lässt, über welche Zeitdauer die Teilprozesse (z.B. Aufträge) der jeweiligen Stellen "aktiviert" sind. Beim stellenorientierten Ablaufdiagramm liegt die Betonung eher auf der inhaltlichen Detaillierung der Prozesse/Teilprozesse und weniger auf dem zeitlichen Aspekt.

Ein weiterer Vorteil dieser Darstellung ist darin zu sehen, dass die Bestrebung vieler Unternehmen unterstützt wird, klare Kunden-Lieferanten-Verhältnisse zu schaffen und Schnittstellen in Auftragsprozessen zu eliminieren. Die Wertschöpfungspfeile und die Teilprozesse enden für die Stellen erst dann, wenn die Lieferung tatsächlich erfolgt ist.

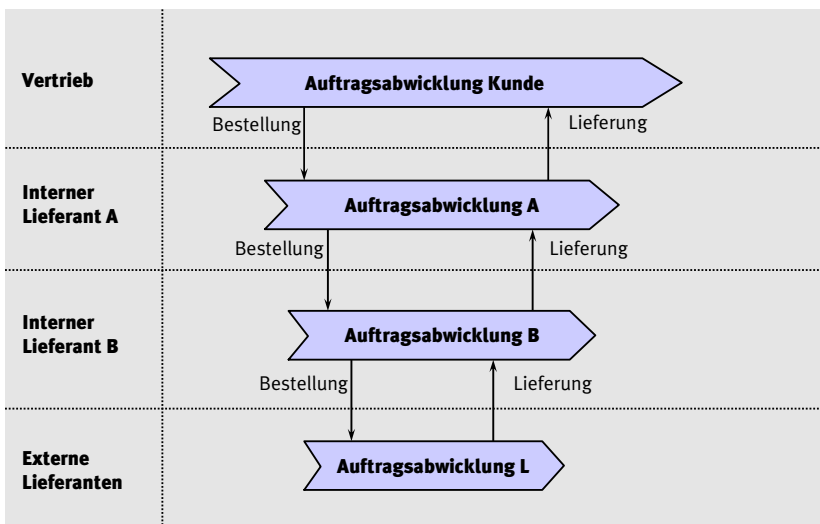


Abb. 34: Auftragskaskade

### 3.9 Integrierte prozessorientierte Methode: ARIS

Nachfolgend wird am Beispiel “ARIS” eine integrierte prozessorientierte Methode vorgestellt. ARIS ist ein Methodenset, welches primär der Prozesssicht und sekundär der Funktionssicht zuzuordnen ist (s. Abb. 35). Der Grund für die Zuordnung liegt im methodischen Schwerpunkt auf den so genannten erweiterten ereignisgesteuerten Prozessketten (eEPK). Diese zeigen die Verknüpfung der Prozesse mit den Funktionen und entsprechen in abgewandelter Weise dem Flussdiagramm (Scheer 2001).

Die unten stehende Abbildung zeigt, dass neben den Prozessketten weitere “Diagramme” in der einen oder anderen Form enthalten sind. Der ARIS-Ansatz hat den Vorteil, dass er bereits mehrere Aspekte berücksichtigt und diese in entsprechender Weise zu kombinieren versucht. Allerdings geht ARIS im Grundansatz von einer datenorientierten und nicht von einer objektorientierten Betrachtungsweise aus – auch wenn neu gewisse Ergänzungen im Bereich UML bestehen (s. Kapitel 5). Außerdem wird die Aufgabensicht vernachlässigt.

In den letzten Jahren wurde bei ARIS die Strukturierung der Prozesse im hierarchischen Sinne stark verbessert und ein hierarchisches Prozessmodell hinzugefügt. Damit wird wurde entsprechend dem “Prozessfokus” nun auch das Hauptmodell der Prozesssicht hinzugefügt. Weitere integrierte Methoden mit einem Schwergewicht auf dem Prozess sind: Medils (Schönsleben 2000), Business Engineering (Österle 1995).

↓ **Hauptfokus**

<div>primär</div> <div>sekundär</div>	<b>Prozesssicht</b>	<b>Funktionssicht</b>	<b>Objektsicht</b>	<b>Aufgabensicht</b>
<b>Prozesssicht</b>	<b>Prozessmodell</b>	Informationsflussdiagramm		
<b>Funktionssicht</b>	Flussdiagramm: Ereignisgesteuerte Prozesskette (EPK)	<b>Funktionsmodell</b>		
<b>Objektsicht</b>			<b>Objektmodell (Entity-Relationship &amp; UML)</b>	
<b>Aufgabensicht</b>		Funktionszuordnungsdiagramm		<b>Organisationsmodell</b>

Abb. 35: Integrierte prozessorientierte Methode – ARIS

**3.10****Übung: Prozessorientierte Modellierung**

Wiederum wollen wir eine kurze Case Study zur Sensibilisierung durchführen. Auch dieser Case betrifft den bereits bekannten Fahrradhersteller (s. Abschnitt 2.1.4).

Versuchen Sie, aufgrund der oben stehenden Beschreibung (s. Abb. 1) des Fahrradgeschäftes eine Prozesslandkarte zu erstellen. Eine Lösung ist unten angegeben (s. Abb. 36).

Wie Sie sehen, werden in dieser Lösung drei Prozessgruppen unterschieden. Der Prozess des Finanzcontrollings wurde zum Treuhänder ausgelagert – man spricht in diesem Zusammenhang auch von Outsourcing eines Prozesses. Der Kundenprozess wurde in drei bzw. vier Auftragsprozesse segmentiert.

Versuchen Sie sodann zu eruieren, welche Prozesse Funktionselemente enthalten, die dem Funktionsbereich “verwalten Finanzen” zugeordnet werden könnten.

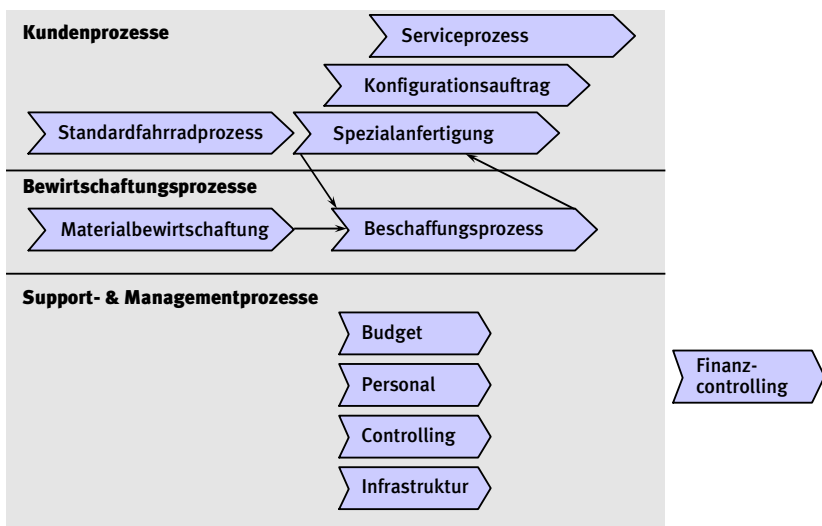
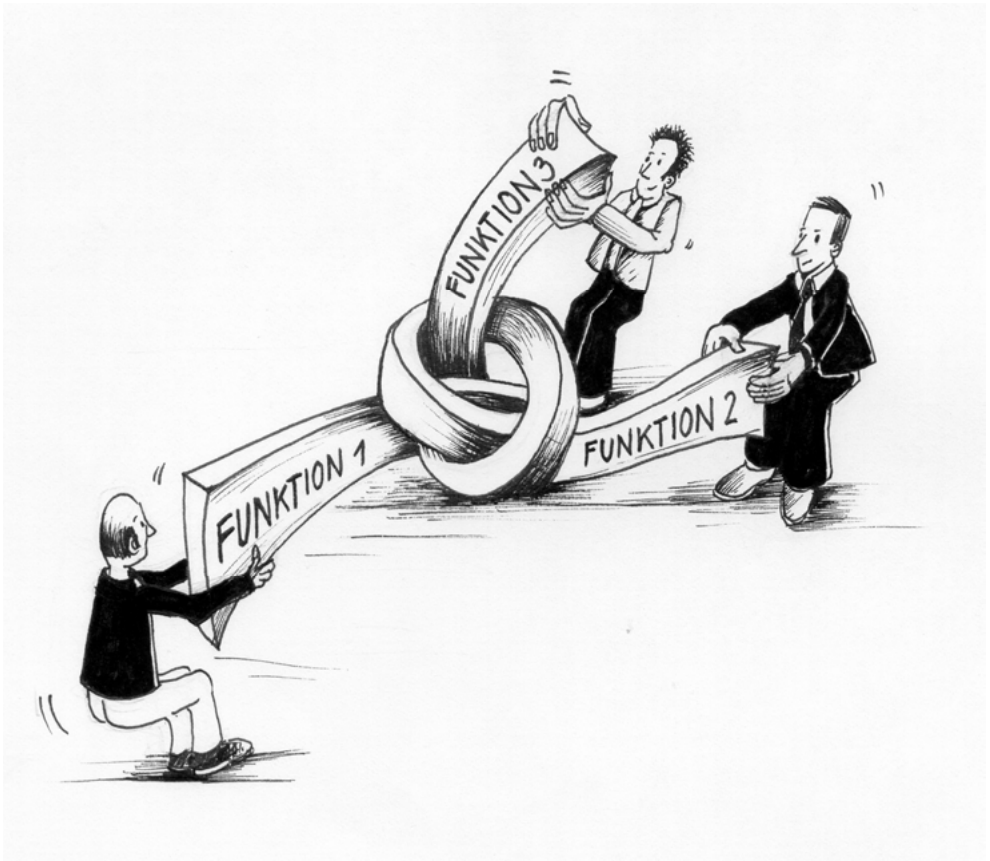


Abb. 36: Beispiel-Prozesslandkarte des Fahrradherstellers

### **3.11 Literatur zur prozessorientierten Modellierung**

- Burkhard, R. (1997): UML – Unified Modeling Language: Objektorientierte Modellierung für die Praxis. Bonn: Addison Wesley Longman.
- Chapin, N. (1970): Flowcharting with the ANSI Standard: A Tutorial. ACM Computing Surveys, Vol. 2, No. 2. New York: ACM Publications.
- Davenport, T. (1993): Process Innovation: Reengineering Work through Information Technology. Boston, MA: Harvard Business School Press.
- Grull, W. (1928). Die Organisation von Fabrikbetrieben. Leipzig: Glöckner.
- Hess, T., Brecht, L. (1996): State of the Art des Business Process Redesign: Darstellung und Vergleich bestehender Methoden. 2. Auflage. Wiesbaden: Gabler.
- Johannson, H., McHugh, P. et al. (1996): Business Process Reengineering: Break Point Strategies for Market Dominance. Chichester, West Sussex: John Wiley & Sons.
- Osterloh, M., Frost, J. (2003): Prozessmanagement als Kernkompetenz: wie Sie Business Reengineering strategisch nutzen können. 3. Auflage. Wiesbaden: Gabler.
- Österle, H. (1995): Business Engineering: Prozess- und Systementwicklung. Berlin: Springer.
- Schank, R.C. & Abelson, R.P. (1977): Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures. Hillsdale, NJ: Lawrence Erlbaum.
- Scheer, A.W. (2001): ARIS: Modellierungsmethoden, Metamodelle, Anwendungen. Berlin: Springer.
- Schönsleben, P. (2000): Integrales Logistikmanagement: Planung und Steuerung von umfassenden Geschäftsabläufen. Berlin: Springer.
- Tipotsch, C. (1997): Business Modelling: Vorgehensmethodik und Gestaltungsmodelle. Graz: Technische Universität.
- Yourdon, E. (1989): Modern Structured Analysis. Englewood Cliffs, NJ: Prentice Hall.





# 4

## Modellierung der Funktionen

### Kapitelinhalt

- Funktionen als Zusammenfassung “ähnlicher Transformationen”
- Funktionsmodelle, Funktionshierarchien und Vererbung
- Identifikation und Spezifikation von Funktionen
- Modellierungsmethoden und Diagramme der Funktionssicht

Dieses Kapitel beschäftigt sich mit der funktionalen Sichtweise – also mit der Modellierung eines Informationssystems aus der funktionsorientierten Sichtweise – man könnte auch sagen, wir setzen uns eine funktionale Brille auf.

Nach einer allgemeinen Einführung behandeln die nachfolgenden Abschnitte die funktionalen Methoden im Detail. Bei der Darstellung der funktionsorientierten Modelle wird wieder demselben Schema wie bei der prozessorientierten Modellierung gefolgt. Zunächst wird auf das “reine Funktionsmodell” und die nähere Spezifikation von Funktionen eingegangen.

Zum Schluss wird auf eine integrierte funktionale Methode, die “strukturierte Analyse”, eingegangen. Bei der strukturierten Analyse handelt es sich um eine Modellierungsmethode, welche sich in den 80er-Jahren in gewisser Weise zu einem Modellierungsstandard entwickelt hat. Diese Methode wird denn auch durch viele Case Tools unterstützt.

## 4.1 Funktionsorientierte Modellierung

### 4.1.1 Funktionssicht

*Definition: Die Funktionssicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der Verwandtschaft und Ähnlichkeit der Elementarfunktionen.*

*Definition: Funktionen sind zusammengehörige und in sich relativ geschlossene Einheiten von verwandten und ähnlichen Elementarfunktionen der unterschiedlichen Systemniveaus.*

Sozio-technische Funktionen können wie mathematische Funktionen bzw. Transformationen aufgefasst werden, die einem Argument  $x$  einen Wert  $f(x)$  zuordnen. “Functions transform inputs into outputs” (Yourdon 1989, S. 67). Die Funktion “zahlen (Objekt)” kann auf das Objekt “Lieferant” bzw. auf “Person” wirken. Funktionen wie “zahlen” und “mahnen” können nun im Sinne der Bearbeitung als “funktional verwandt” bezeichnet werden. Bringt man solcherart verwandte Funktionen in eine hierarchische Struktur, dann entsteht ein Funktionsbaum bzw. Funktionsmodell. Es entsteht eine zweite Sichtweise, die sich von der Prozesssicht offensichtlich unterscheidet – Prozesse werden nach der Abfolge und nicht nach der Ähnlichkeit gruppiert (s. Abb. 37). Problematisch ist allerdings, dass der Begriff “verwandt” *offen* lässt, was dies exakt bedeutet. Es entstehen daher auch verschieden strukturierte Funktionsmodelle.

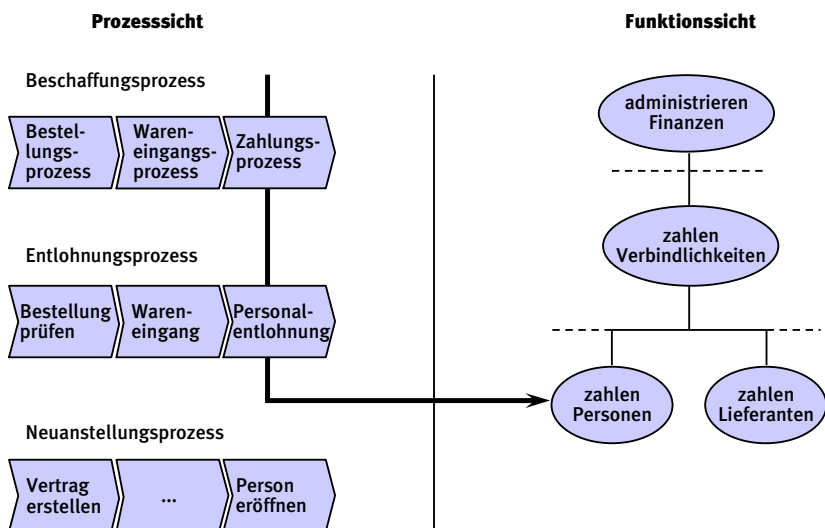


Abb. 37: Prozess- versus Funktionssicht



## 4.2

**Funktionsorientierte Methoden im Überblick**

Funktionsorientierte Methoden stellen demnach “Elementarfunktionen” und darauf aufbauend aggregierte Funktionsgruppen des Systems in den Mittelpunkt. Sie gehen von diesen Funktionsgruppen aus und betrachten Beziehungen zu den übrigen Aspekten. Entsprechend den prozessorientierten Methoden gibt es Methoden, die Funktionen primär betrachten und solche, welche den sekundären Aspekt untersuchen (s. Abb. 38).

Im Zentrum der funktionsorientierten Modellierung steht das Funktionsmodell. Dieses ordnet die Gesamtheit der Elementarfunktionen in eine hierarchische Struktur mit unterschiedlichen Ebenen ein und zeigt die gegenseitigen hierarchischen Beziehungen zwischen den Funktionen. Das Funktionen-Blockdiagramm zeigt – aufbauend auf den identifizierten Funktionen – Prozess-Beziehungen zwischen diesen Funktionen auf.

Das Datenflussdiagramm zeigt Beziehungen zwischen den Funktionen und den Datenobjekten sowie zwischen den Funktionen untereinander.

Das Use-Case-Diagramm bildet sodann die Beziehungen zwischen den Funktionen und den Aufgaben ab. Es vermerkt, welche Funktionen von welchen Aufgaben benutzt (“use”) werden.


		<div style="text-align: center;"> <b>Hauptfokus</b>   </div>		
<b>primär</b>	<b>Prozesssicht</b>	<b>Funktionssicht</b>	<b>Objektsicht</b>	<b>Aufgabensicht</b>
<b>sekundär</b>				
<b>Prozesssicht</b>		Funktionen-Blockdiagramm		
<b>Funktionssicht</b>	Flussdiagramm	<b>Funktionsmodell</b>	Class-Responsibilities-Collaborators	Stellen-funktionen-diagramm
<b>Objektsicht</b>		Datenfluss-Diagramm		
<b>Aufgabensicht</b>		Use-Case-Diagramm		

Abb. 38: Primäre und sekundäre Modellierung der Funktionen

### 4.3 Funktionsmodell

*Definition: Funktionsmodelle stellen die hierarchischen Strukturen und Beziehungen von Funktionen eines Systems dar.*

Auch für Funktionsmodelle ist der Umstand wichtig, dass sich Funktionen hierarchisch strukturieren lassen (HIPO 1974). Dies bedeutet beispielsweise, dass der Funktion “administrieren Aufträge” die Subfunktionen “erfassen Auftrag”, “terminieren Auftrag”, etc. zugeordnet werden können (s. Abb. 39). Gut ersichtlich wird diese Hierarchie auch in “Menüebäumen” von funktional programmierten Informationssystemen, z.B. dem Auftragswesen: Aufträge anlegen, Aufträge erfassen etc. Anders als bei der Prozesshierarchie werden die Subfunktionen einer Hierarchieebene allerdings nicht einfach sequenziell durchlaufen, sondern von verschiedenen Prozessen genutzt.

Häufig werden Anforderungen an IT-Systemen nach dem funktionalen Gesichtspunkt aufgebaut. Beispiel: Die Funktionsliste “einkaufen” besteht aus den Subfunktionen: “auswählen Lieferanten”, “tätigen Bestellungen”, “buchen Wareneingänge”. Durch den Einsatz von Prozessmodellen und dem Rückgang funktional programmierter Systeme hat die Bedeutung von Funktionsmodellen leicht an Bedeutung verloren und es werden vermehrt auch Prozessmodelle verwendet. Das Konstrukt der Funktion ist, wie bereits mehrfach erwähnt, problematisch, da es in Gefahr läuft, mit Teilprozessen vermischt zu werden.

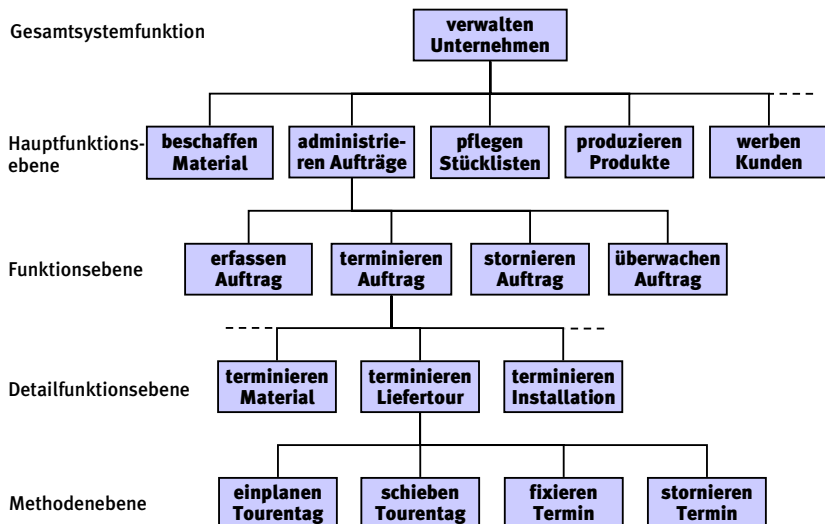


Abb. 39: Funktionsmodell

#### 4.4 Identifikation und Systemabgrenzung der Funktionen

Was für Prozesse gilt, gilt auch für Funktionen: Funktionen sind nicht “per se” bekannt und müssen identifiziert werden. Es ist sodann auch zu klären, welche Funktionen innerhalb der Systemgrenze liegen. Es kann beispielsweise sein, dass im Rahmen eines Projektes zwar der Auftragsabwicklungsprozess betrachtet wird, die Funktion “administrieren Finanzen” aber ausgenommen bleibt. Mit anderen Worten: Neben der Systemabgrenzung hinsichtlich der Prozesse (z.B. “Ausschluss des Serviceprozesses”) ist u.U. eine Abgrenzung hinsichtlich der Funktionen vorzunehmen. Eine funktionale Systemabgrenzung erfolgt idealerweise mit dem Funktionen-Blockdiagramm (s. u.).

Die funktionsorientierten Modelle sind im Grunde nicht besonders komplex oder schwer verständlich. Kritisch ist eher die methodische Frage, wie man aufgrund eines gegebenen Systems zu den “korrekten Funktionen” kommt. Im Prinzip gibt es zwei Ansätze: einen Top-down-Ansatz und einen Bottom-up-Ansatz (Yourdon 1989, S. 360). Zusätzlich lassen sich Funktionen aus Prozessen oder anderen Modellen ableiten (s.u.).

Probleme kann allerdings die Abgrenzung einer Funktion zu einem Teilprozess bereiten. Auf der untersten Analyseebene (Operationen) entsprechen sich Funktion und Teilprozess. Würde die Funktion nochmals näher detailliert und analysiert, so würde diese ebenfalls wieder durch einen Prozess vollzogen. Mit der vertieften Analyse einer elementaren Funktion mutiert diese aber gewissermaßen neu zu einem Prozess.

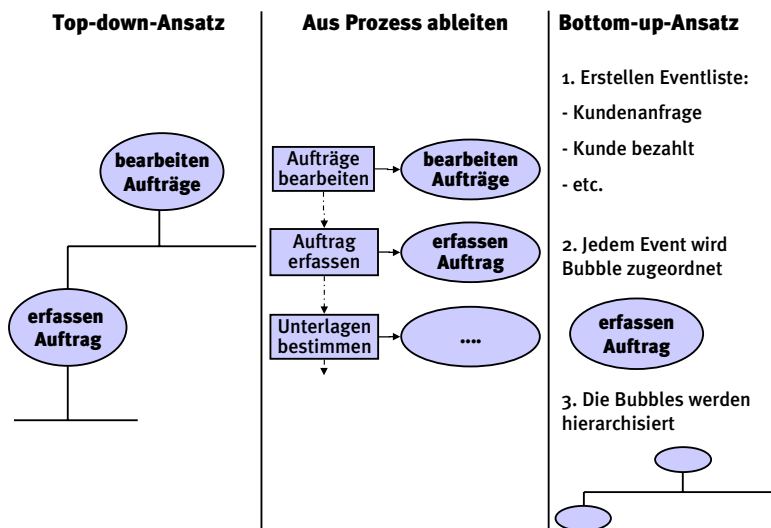


Abb. 40: Möglichkeiten zur Identifikation von Funktionen

#### 4.4.1 Top-down versus Bottom-up

Im Top-down-Ansatz versucht man, ausgehend von einer Top-Funktion, die darunterliegenden Funktionen analytisch und gedanklich abzuleiten. Z.B. nimmt man die Top-Funktion "bearbeiten Aufträge" und sucht daraus die darunterliegenden Funktionen abzuleiten (s. Abb. 40).

Diese Vorgehensmethode mag zwar einleuchten, die Problematik besteht aber darin, dass durch abstraktes und rein analytisches Zusammenfassen unter Umständen die Nähe zum Denken der Benutzer verlorengeht. Während DeMarco auch ein hierarchisches "Top-down"-Vorgehen zur Ermittlung der Gesamtheit aller Datenflussdiagramme vorsieht, schlägt Yourdon ein anderes Vorgehen vor. Yourdon kritisiert nämlich, dass der "Top-down"-Ansatz von DeMarco bei sechs Analytikern zu sechs verschiedenen Modellen führe (Yourdon 1989, S. 360).

Beim Bottom-up-Ansatz geht man umgekehrt vor. Man bestimmt zunächst, welche Ereignisse (engl.: "Events") in einem System überhaupt auftreten können. Auf dieser Basis erstellt man eine so genannte "Event-Liste". Events entsprechen unseren Operationen und können sein: Kunde bestellt, Material wird ausgelagert, etc. Hat man die "Event-Liste" erst einmal erstellt, dann wird jedem Event eine eigene Elementarfunktion, ein einzelnes "Bubble", zugewiesen. Darauf beginnt man gewissermaßen mit einem Bereinigungsprozess. Dieser ordnet, sortiert und fasst die Menge der Elementar-Bubbles (Elementarfunktionen) nach der Verwandtschaft zusammen (Yourdon 1989, S. 371).

Die so entstandene Gesamtheit von Datenflussdiagrammen wird nun anschließend einem Reorganisationsprozess und einem Bereinigungsprozess unterzogen. Hierzu werden aufeinander bezogene Funktionen zu Aggregaten gruppiert. Yourdon bemerkt dabei, dass ohne eine solche Aggregation das Modell für den Benutzer nicht verständlich ist: "However, it should be evident that this model cannot be presented to the user for verification. Why not? Primarily because it is too complicated" (Yourdon 1989, S. 370). Wir werden im zweiten Teil des Buches sehen, dass es noch weitere Möglichkeiten gibt, zu den Funktionen zu gelangen. Insbesondere besteht die Möglichkeit, die Funktionen von jenen Modellen zu extrahieren, welche die funktionale Sicht sekundär beleuchten. Man wird also ausgehend vom Prozess (Flussdiagramm), Class-Responsibilities-Collaborators und dem Stellenfunktionendiagramm zu den Funktionen gelangen. Insbesondere das Stellenfunktionendiagramm (s. u.) erscheint geeignet, ausgehend von den tatsächlichen Aufgaben einer Organisationseinheit die relevanten Funktionen abzuleiten und diese anschließend zu strukturieren.

#### 4.4.2 Systemabgrenzung

Auch im Rahmen der funktionsorientierten Modellierung werden Systeme abgegrenzt – man spricht hier von Kontextdiagrammen (Yourdon 1989). Ein Kontextdiagramm enthält demnach eine Funktion (s. Abb. 41).

Diese Funktion wird als so genanntes “Bubble” dargestellt und ist mit einer Bezeichnung versehen, die ausdrückt, dass es sich um eine Funktion handelt. Idealerweise beginnen alle Funktionsbezeichnungen mit einem Verb. Die Rechtecke stellen so genannte “Terminatoren” dar. Es sind dies die Umsysteme des Systems. Die Pfeile zwischen der Funktion und den Terminatoren geben die Richtung an und müssen zwingend beschriftet werden.

Es gilt die Regel (zumindest in der strukturierten Analyse – s. u.), dass zwischen den Terminatoren keine Pfeile gezeichnet werden dürfen. Terminatoren dürfen aber – falls es der Übersichtlichkeit dient – mehrfach vorkommen.

Das Kontextdiagramm entspricht damit exakt einer Systemabgrenzung gemäß Systems Engineering. Es zeigt auf, welche Beziehungen die abzugrenzende Funktion (in unserem Beispiel “administrieren Aufträge”) mit seinen Umsystemen hat.

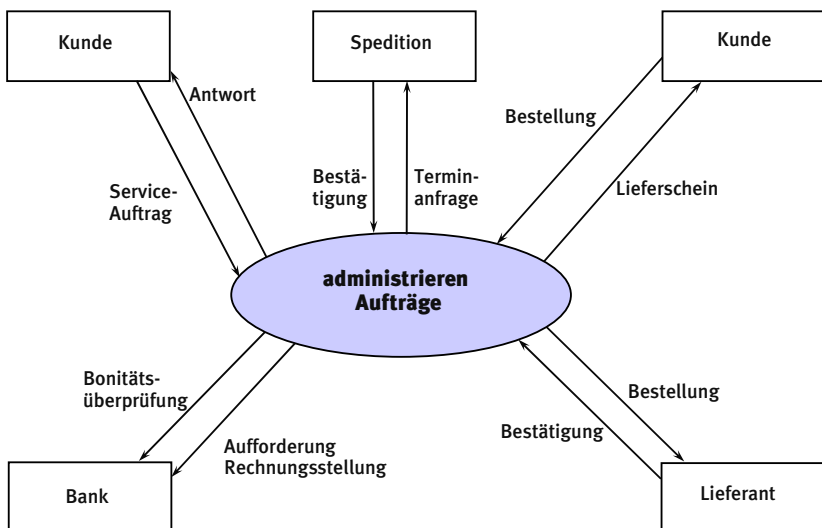


Abb. 41: Kontextdiagramm einer Funktion

## 4.5 Spezifikation von Funktionen

Sind die Funktionen erst einmal identifiziert, dann müssen auch sie näher spezifiziert werden. Das Vorgehen zur Spezifikation von Funktionen ist analog dem Vorgehen zur Spezifikation von Prozessen.

### Abgrenzung und Beziehungen zu Umsystemen

In einem ersten Schritt ist festzuhalten, zu welchen anderen Funktionen bzw. Umsystemen Beziehungen bestehen (s. Systemabgrenzung).

### Inhalte einer Funktion und enthaltene Subfunktionen

Sodann ist die "innere Struktur" einer Funktion zu beschreiben. Die in einer Funktion enthaltenen Subfunktionen lassen sich aus dem Funktionsmodell ableiten. Sie können nun entweder durch strukturierten Text im Sinne einer tabellarischen Aufzählung beschrieben oder durch einen Funktionsbaum visualisiert werden (s. Funktionsmodell).

Elementare Funktionen können durch "strukturierten Text" beschrieben werden. Diese Form der Spezifikation wird auch "minispec" für "miniature specification" bezeichnet. Beispiel:

Die Fertigungsaufträge müssen in eine hierarchische Form gebracht werden können. Dies bedeutet, dass ein Fertigungsauftrag mehrere Subaufträge enthalten kann und diese unabhängig durch die Fertigung laufen können. Ein Oberauftrag darf keinen Status enthalten, der weiter ist als einer seiner Unteraufträge ...

### Mengen, Häufigkeiten und Regeln

Neben funktionsbezogenen Angabe von Mengen und Häufigkeiten können zur Spezifikation von Funktionen ebenfalls Regeln der Funktion beschrieben. Dies umfasst beispielsweise Pre-/Post-Konditionen sowie etwaige Entscheidungstabellen (Yourdon 1989, S. 203).

### Beziehungen zu anderen Aspektsystemen

Zur Spezifikation einer Funktion gehört sodann, dass ihre Beziehungen zu den übrigen Aspektsystemen aufgezeigt werden. Dies sind die Beziehungen zu den Prozessen, Objekten und Aufgaben. Diese Beziehungen werden nun aber gerade durch die entsprechenden Diagramme spezifiziert. Damit besteht ein ganz wesentlicher Teil der Spezifikation aus den nachfolgend zu beschreibenden Systemmodellen.

## 4.6 Funktionen-Blockdiagramm

*Definition: Funktionen-Blockdiagramme stellen primär Funktionen dar und sekundär die diese Funktionen tangierenden Prozesse.*

Das Funktionen-Blockdiagramm hat im Rahmen der Modellierung von Informationssystemen eine große Verbreitung erlangt. Die (primären) Funktionen werden in Form von Funktionsrechtecken dargestellt (z.B. erfassen Auftrag, disponieren Auftrag) (s. Abb. 42). Es ist darauf zu achten, dass die Rechtecke *Funktionen* bezeichnen und *nicht* organisatorische Stellen, wie häufig unklar abgegrenzt wird.

Als zweiter Aspekt werden in einem nächsten Schritt die Prozesse eingetragen. Dies erfolgt durch Angabe der Verbindungen zwischen den Funktionen. Diese Verbindungen werden häufig je Prozess mit einer unterschiedlicher Strichart symbolisiert. Ergänzend kann der Informationsfluss beschriftet oder sogar explizit nummeriert werden.

In gewissen Fällen kann auch dieses Diagramm um eine dritte Dimension ergänzt werden, indem die organisatorischen Verantwortlichkeiten eingezeichnet oder Datenspeicher eingezeichnet werden. Man könnte also beispielsweise zusätzlich vermerken, welche der hier vorliegenden Funktionen auf welche Datenquellen zugreifen. Funktionen-Blockdiagramme werden häufig auch für Systemabgrenzungen verwendet. Man kann so visualisieren, welche Funktionen ein Informationssystem abdecken sollte.

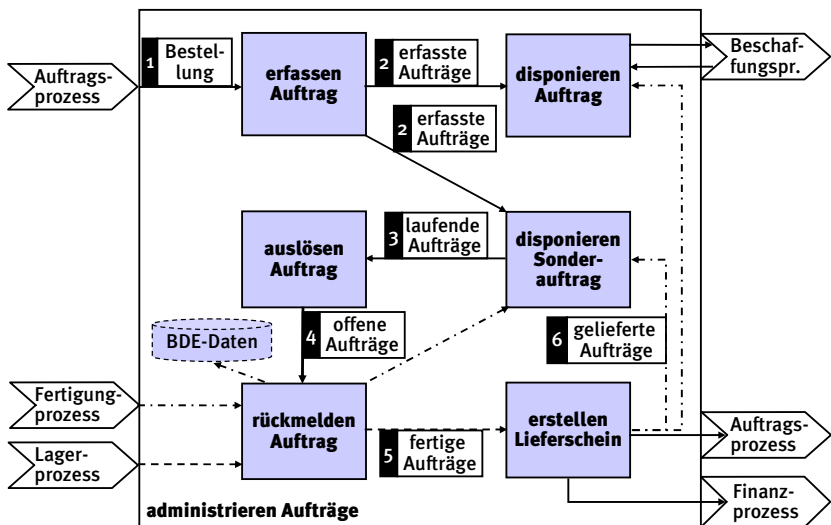


Abb. 42: Funktionen-Blockdiagramm

## 4.7 Datenflussdiagramm

*Definition: Datenflussdiagramme (DFD) stellen primär Funktionen dar und sekundär die sie tangierenden Objekte sowie gegenseitige Informationsbeziehungen.*

“Keep in mind that the DFD is just one of the modeling tools available to the systems analyst and that it provides only one view of a system – the function-oriented view” (Yourdon 1989, S. 140).

In der unten stehenden Grafik ist ein typisches Datenflussdiagramm dargestellt: Die Kreise (engl.: “Bubbles”) stehen für Funktionen (s. Abb. 43). Die Pfeile visualisieren Informationsbeziehungen zwischen den Funktionen und von den Funktionen zu den Datenspeichern (geöffnete Rechtecke). In der vorliegenden Darstellung greift z.B. die Funktion “erfassen Auftrag” auf den Datenspeicher “Kundenauftrag” zu. Die Funktion muss zur Erledigung ihrer Aufgabe den Datenspeicher schreiben und lesen. Datenspeicher werden in der Regel erst dann eingezeichnet, wenn in einer Grafik mehrere “Bubbles” auf denselben Datenspeicher zugreifen. Unter Umständen werden auch Beziehungen zu den Umsystemen durch so genannte Terminatoren dargestellt, wie dies bereits bei der Systemabgrenzung mit Hilfe des Kontextdiagramms zur Sprache gekommen ist. Im Gegensatz zum Flussdiagramm werden nicht die zu einem Prozess gehörigen Funktionen gezeichnet, sondern nur die direkten Nachfolger einer übergeordneten Funktion (s. nächster Abschnitt).

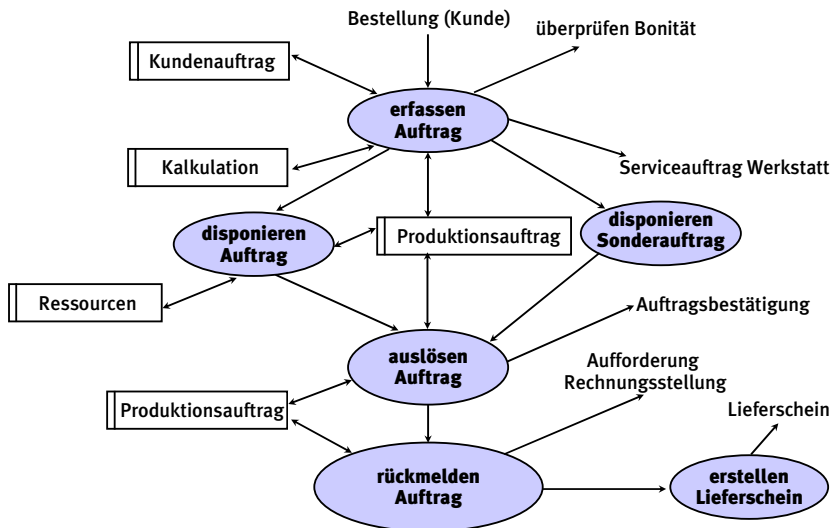


Abb. 43: Datenflussdiagramm



### Hierarchische Gliederung der Datenflussdiagramme

Von Bedeutung ist nun der Umstand, dass einzelne Funktionen ("Bubbles") auf einer tieferen Ebene wiederum selber als Datenflussdiagramm dargestellt und beschrieben werden können. Damit entsteht eine baumförmige Hierarchie von Datenflussdiagrammen, die zueinander in Beziehung stehen. Im Prinzip gilt dasselbe Vorgehen natürlich für alle Diagramme, begonnen bei den Prozessmodellen, für das bereits erwähnte Funktionen-Blockdiagramm sowie für alle nachfolgenden Diagramme.

Wenn man das Funktionsmodell erstellt hat, welches die eigentliche Grundlage bildet, dann ist auch die Basis für die Erstellung der Datenflussdiagramme erfolgt. Die Topfunktion im Kontextdiagramm entspricht nämlich der Wurzel der Hierarchie. Das erste Datenflussdiagramm (auch als Ebene 0 bezeichnet) enthält sodann alle Funktionen auf der direkt darunter liegenden Ebene.

Datenflussdiagramme sind daher nicht fundamental von Funktionsmodellen verschieden. Sie zeigen aber ergänzend die Beziehungen zwischen den Funktionen und die betroffenen Objekte, jeweils bezogen auf eine einzelne Hierarchiestufe. Eine wichtige Eigenschaft von Datenflussdiagrammen ist die Möglichkeit von Konsistenzchecks. So kann man die Anzahl der In- und Outputs eines "Bubble" auf der oberen Stufe zählen und auf der unteren Ebene überprüfen, ob die gleiche Anzahl an Informationsbeziehungen wieder erscheint.

Gelegentlich wird argumentiert, dass das Datenflussdiagramm auch die *Prozesse* zeigt. Dies stimmt nur bedingt. Es zeigt nämlich die Summe aller Prozesse, da alle Informationsbeziehung bzw. In- und Outputs der Funktionen eingezeichnet werden müssen, unabhängig davon, um welchen Prozess es sich handelt. Würde man nur die Informationsbeziehungen eines Prozesses einzeichnen, dann mutierte das Datenflussdiagramm zu einem Flussdiagramm, allerdings angereichert um die Datenspeicher.

### Problem des Komplettheitsanspruches

Kommen wir nun noch zu einigen Problemen bzw. zu den Schlussfolgerungen der Modellierung von Datenflussdiagrammen: Das vielleicht größte Problem liegt im Komplettheitsanspruch dieses Modellierungsansatzes. Durch die baumartige Struktur entstehen sehr schnell sehr viele Datenflussdiagramme. Dies kann dazu führen, dass man den Überblick verliert. Damit ist aber auch die Gefahr verbunden, dass die Systemspezifikation durch die Benutzer nicht mehr wirklich verifiziert wird.

## 4.8 Use-Case-Diagramm

*Definition: Use-Case-Diagramme stellen primär Funktionen dar und sekundär die diese Funktionen wahrnehmenden Stellen sowie ihre zur Aufgabenerfüllung notwendigen Beziehungen.*

Use-Cases legen die Abhängigkeiten zwischen den Funktionen ("they capture a system's functional requirements", Jacobson 1992) und den Aufgaben ("Actors") dar (s. Abb. 44). Die Stärke des Use-Case-Diagrammes liegt darin, dass Funktionen bezogen auf die Anwender dargestellt werden und eine sehr persönliche Sichtweise ermöglichen.

Use-Cases haben sich im objektorientierten Umfeld etabliert und ergänzen dort die funktionale Sichtweise. Sie werden zur Darstellung funktionaler Anforderungen aus Sicht der Anwender ("Actors") verwendet: Ein Use-Case "is a way to use the system. Users interact with a system by interacting with its use cases. Taken together, a system's use cases represent everything users can do with it" (Jacobson 1995, S. 310). Leider ist in der Praxis meist unklar, ob es sich bei Use-Cases tatsächlich um Funktionen handelt oder nicht viel eher um Prozesse. Einschränkend erscheint außerdem der Begriff "Use-Case". Es gibt keinen Grund, in einem frühen Analysestadium lediglich jene Funktionen im Use-Case-Diagramm darzustellen, welche später auch als Systemfunktionalität umgesetzt werden sollten. Dies legt aber der Begriff "use" nahe. Man würde die Mächtigkeit dieser Darstellungsform einschränken.

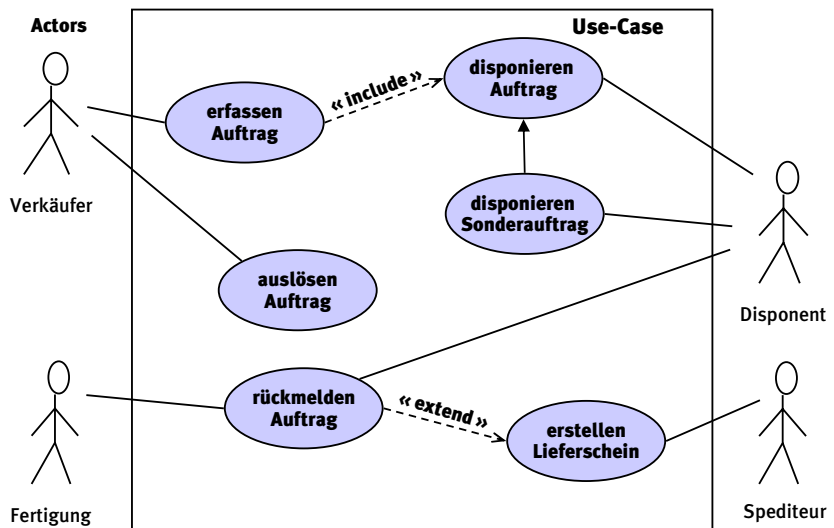


Abb. 44: Use-Case-Diagramm

#### 4.8.1 Beziehungen zwischen Use-Cases

Use-Cases (Funktionen) können außer der standardmäßigen “Assoziations-Beziehung” auch durch so genannte “extend-Beziehungen” bzw. “include-Beziehungen” untereinander in Verbindung stehen (s. Abb. 45). Die “extend-Beziehung” bedeutet, dass ein Use-Case die Funktionalität eines anderen Use-Cases nur punktuell erweitert. Die “include-Beziehung” bedeutet, dass ein gewisser Use-Case die Funktionalität eines anderen Use-Cases voll mit einschließt. Mit der “Generalisierungs-Beziehung” erhält man die Möglichkeit, Use-Cases (Funktionen) in eine Vererbungshierarchie hineinzubringen und diese Top-down zu strukturieren. Es bedeutet, dass z.B. die Disposition von Sonderaufträgen eine Erweiterung zur Disposition von normalen Aufträgen darstellt. Das gleiche Prinzip wurde bereits bei den Prozessen (Segmentierung) diskutiert und wird uns später auch bei der Objektorientierung beschäftigen. In diesem Sinne findet sich das Grundprinzip der Vererbung in allen Sichten wieder. Use-Cases (Funktionen) können dadurch ganz entsprechend den allgemeinen Systemeigenschaften hierarchisch geschachtelt werden.

Auf Basis des Verständnisses, dass Use-Cases Funktionen darstellen, erscheint es sehr unglücklich, wenn Use-Cases zur Modellierung von Prozessen hinzugezogen werden (Jacobson 1995, S. 208). Zwar könnten im Prinzip Elementarfunktionen als Prozesse betrachtet werden. Im Sinne eines langfristig konsistenten Modells sollte aber eine klarere Abgrenzung zwischen der Prozess- und der Funktionssicht vorgenommen werden.

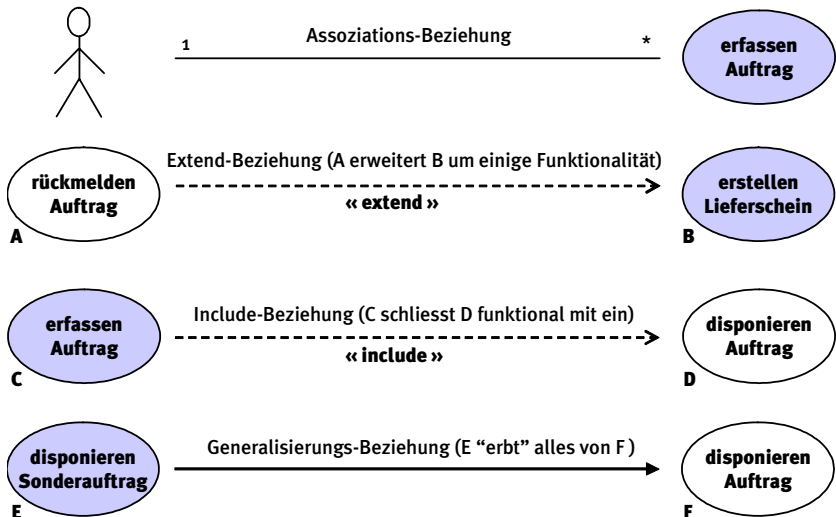


Abb. 45: Beziehungen zwischen Use-Cases

## 4.9 Integrierte funktionale Methode: Strukturierte Analyse

Die Methode der strukturierten Analyse und Design (SA/SD) entstand Ende der 70er-Jahre insbesondere durch Yourdon und Constantine (1975), DeMarco (1978), IDEF (1981) SADT von Ross (1985), Gane & Sarson (1975), Yourdon (1987). Ausgangspunkt waren die Ansätze der funktionsorientierten Programmierung (HIPO 1974). In den 80er-Jahren hat sich die Methode zu einem De-facto-Standard im Bereich der Systemanalyse entwickelt, aber in den 90er-Jahren etwas an Terrain eingebüsst, da die objektorientierten Methoden Überhand gewonnen haben. Sie betont gemäß der unten stehenden Matrix primär den funktionalen Gesichtspunkt (s. Abb. 46). Im Mittelpunkt steht eine Erweiterung des Funktionsmodells durch die Datenflussdiagramme.

Zusätzlich wurde diese primär funktionale Betrachtungsweise durch das Entity Relationship Model ergänzt, um eine "ausgewogenere" Analysetechnik zu erhalten. Lehrbücher der strukturierten Analyse zeigen überdies noch Diagrammtypen wie Zustandsübergangsdiagramme und Flussdiagramme, welche den Methodenbaukasten nochmals erweitern.

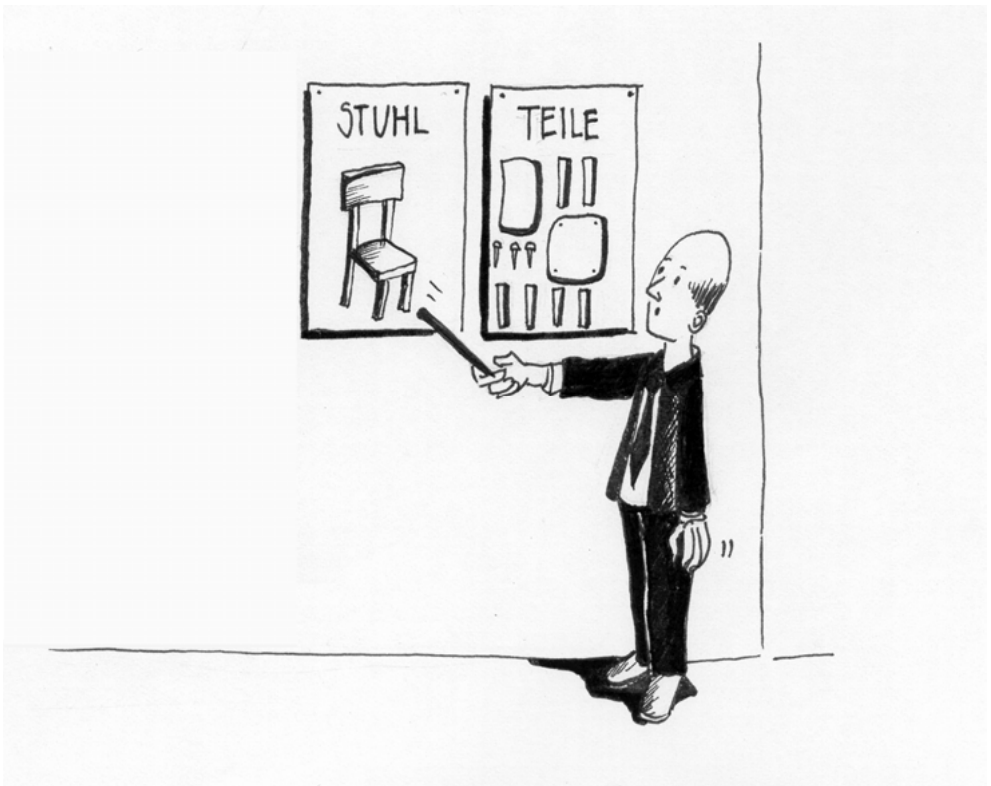
Fazit: Die strukturierte Analyse deckt aspektweise Darstellungen aus fast allen Dimensionen ab – abgesehen von der aufgabenorientierten Betrachtungsweise. Außerdem ist anzumerken, dass häufig auch das Datenflussdiagramm "missbraucht" wird, um Prozesse darzustellen.

		Hauptfokus ↓		
primär sekundär	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
	Prozesssicht		Zustands- übergangs- diagramm	
Funktionssicht	Flussdiagramm	Funktionsmodell		
Objektsicht		Datenfluss- Diagramm	Objektmodell (Entity-Relation- ship)	
Aufgabensicht				

Abb. 46: Methodenkasten der Strukturierten Analyse

**4.10 Literatur zur funktionsorientierten Modellierung**

- DeMarco, T. (1978): Structured Analysis and System Specification. Englewood Cliffs, NJ: Prentice Hall.
- Gane, C., Sarson, T. (1975): Structured Systems Analysis: Tools and Techniques, Englewood Cliffs, N.J.: Prentice Hall.
- HIPO 1974: HIPO – A Design Aid and Documentation Technique. White Planes, N.Y.: IBM.
- IDEF (1981): Standard Users Manual for the ICAM Function Modeling Method – IDEF0, IEEE draft standard, P1320-1-1-1997. Piscataway, N.J.: IEEE Publications.
- Jacobson I. (1992): Object-Oriented Software Engineering: A Use Case Driven Approach. Reading, MA: Addison-Wesley.
- Jacobson, I., Ericsson, I. et al. (1992): The Object Advantage: Business Process Reengineering with Object Technology. Reading, MA: Addison-Wesley.
- Jacobson I. (2001): The Object Advantage: Business Process Reengineering with Object Technology. Reading, MA: Addison-Wesley.
- Jacobson I. (1995): The Use-Case Construct in Object-Oriented Software Engineering. In: Scenario-Based Design – Envisioning Work and Technology in System Development. (Carroll, J.; ed.). New York: John Wiley & Sons. 309–336.
- Ross, D. (1985): Applications and extensions of SADT. Piscataway, N.J.: IEEE Publications.
- Schotten, M. (1998): Funktionen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 144–218.
- Yourdon, E., Constantine, L. (1975): Structured Design: Fundamentals for a Discipline of Computer Program and System Design. New York: Yourdon Press.
- Yourdon, E. (1989): Modern Structured Analysis. Englewood Cliffs, NJ: Prentice Hall.



# 5

## Modellierung der Objekte

### **Kapitelinhalt**

- **Objekte als natürliche Sichtweise von Bearbeitungselementen**
- **Objektmodelle, Bestandteil- und Vererbungshierarchien**
- **Identifikation und Spezifikation von Objekten**
- **Modellierungsmethoden und Diagramme der Objektsicht**

Dieses Kapitel behandelt die objektorientierte Sichtweise. Ein erster Teil betrifft die Grundgedanken der objektorientierten Modellierung. Ein weiterer Themenkreis stellt die Frage, was unter dem Begriff “Objektorientierung” zu verstehen ist. Es wird dargelegt, dass die objektorientierte Sichtweise darin besteht, die Bearbeitungsobjekte des betrieblichen Umfeldes in den Mittelpunkt zu stellen.

Sodann wird ausgeführt, dass Objekte und Klassen über hierarchische Strukturen verfügen und dass die objektorientierte Modellierung sich primär dieser Hierarchien bedient.

Auch dieses Kapitel folgt damit der Struktur der vorangegangenen Kapitel: Zunächst werden die reinen Modelle, also die Objektmodelle, dargestellt, dann wird auf die Charakterisierung von Objekten eingegangen. Danach wird dargelegt, dass es genau diese Objekte sind, an welchen die Tätigkeiten in einem Betrieb ausgeübt werden. Mit anderen Worten, ein Prozess ist nichts anderes als eine Abfolge von Bearbeitungsschritten an konkreten Objekten.

## 5.1 Objektorientierte Modellierung

### 5.1.1 Rekapitulation Funktions- versus Prozesssicht

Bevor wir auf den Begriff der Objektorientierung näher eingehen, werden nochmals kurz die vorangegangenen Abschnitte rekapituliert. Wir haben gesehen, dass sich die Funktions- und die Prozesssicht in der Art der Strukturierung von Operationen unterscheiden (s. Abb. 47).

Wir haben dargelegt, dass die beiden Sichten die Operationen einer Grundgesamtheit nach einem anderen Kriterium gliedern. Die Prozesssicht gliedert die Operationen des Systems nach der sachlogischen und der zeitlichen Abfolge – die Funktionssicht gliedert sie gewissermaßen nach der funktionalen Bearbeitungsähnlichkeit.

Man könnte auch sagen, die Prozesssicht stehe “senkrecht” zur Funktionssicht. Dies wird am besten klar, wenn man sich vor Augen hält, dass Prozesse gerade Funktionsbäume horizontal durchqueren. So können ganz unterschiedliche Prozesse dieselben Hauptfunktionen durchschreiten. Besonders evident wird dies, wenn man funktionsorientiert organisierte Unternehmen betrachtet.

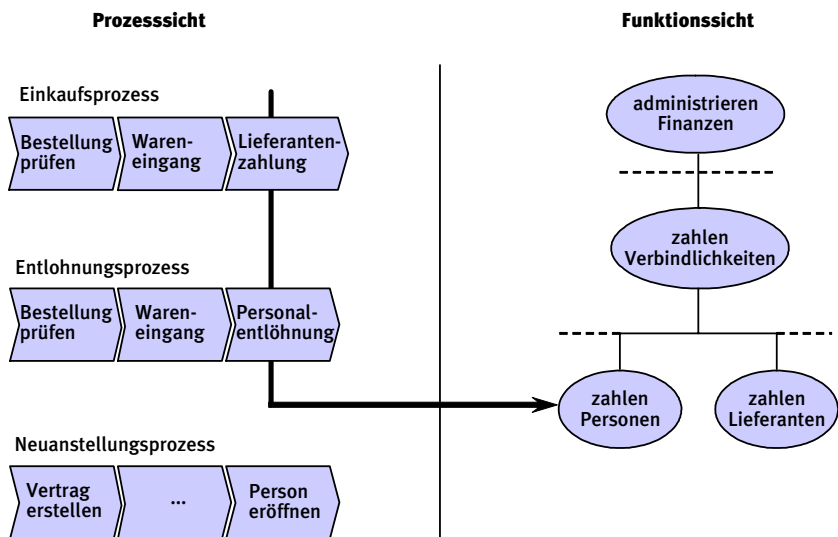


Abb. 47: Prozess- versus Funktionssicht



### 5.1.2 Objektsicht

*Definition: Die Objektsicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der ihm zugrunde liegenden Bearbeitungselemente.*

*Definition: Objekte sind zusammengehörige und in sich relativ geschlossene Einheiten von Bearbeitungselementen der unterschiedlichen System-niveaus.*

Auch die Objektsicht nimmt nun also wieder einmal mehr dieselbe Grundgesamtheit an Operationen auf und gliedert sie nach den Bearbeitungseinheiten bzw. Objekten (s. Abb. 48).

Interessant ist der Umstand, dass *Objekten damit natürlicherweise Elementarfunktionen zugewiesen werden*. Der objektorientierte Ansatz unterscheidet sich dadurch vom “eingeschränkten datenorientierten Ansatz”, der die Bearbeitungseinheiten als Datenobjekte ohne eigene Funktionalität aufgefasst und diesen nur Attribute “zugewiesen hat”.

Im unten stehenden Beispiel (s. Abb. 48) werden daher bei den Bearbeitungseinheiten “Person” bzw. “Lieferant” die ihnen zugehörigen Funktionen (Methoden) angemerkt. Im Falle des Objektes “Person” wären dies die Funktionen “zahlen” und “eröffnen”.

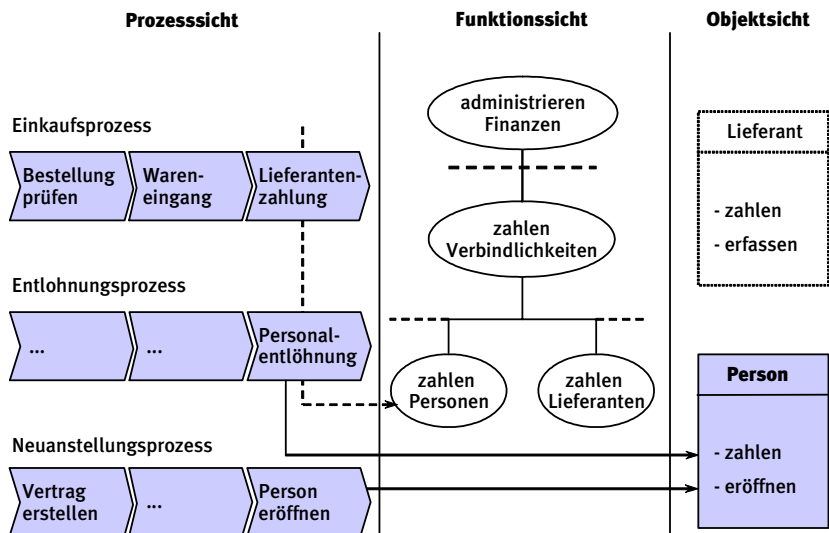


Abb. 48: Funktions-, Prozess- und Objektsicht

### 5.1.3 Objektorientierung – eine natürliche Sichtweise

Fazit: Wir haben in der so genannten Objektsicht ein weiteres Gliederungskriterium unserer Aktivitäten gefunden. Nach der funktionalen Ähnlichkeitsgruppierung und der Prozesssicht stellen wir nun fest: Es gibt eine weitere scheinbar einfache und natürliche Sichtweise.

“Its primary new features are summarized in two key words, encapsulation and inheritance. Encapsulation means that the consumer no longer applies operators to operands (...)” (Cox 1991, S. 12). Cox verweist also zunächst darauf, dass die Objekte “ihre” Funktionalität kapseln. In einem zweiten Punkt verweist er auf die Eigenschaft, dass Objekte ebenfalls über eine “Vererbungsstruktur” verfügen, was im herkömmlichen datenorientierten Ansatz nicht ersichtlich war.

Die Welt, wie wir sie wahrnehmen, besteht aus Objekten: Stühlen, Fenstern, Personen, Büchern, Maschinen, etc. Es liegt in der Natur unseres Denkens, dass wir versuchen, unsere Umwelt zu strukturieren und zu klassifizieren. Wir gebrauchen zu dieser Strukturierung mindestens drei verschiedene Ansätze (Coad et al. 1990):

- Gewohnheitsmäßige Einteilung der Welt in Objekte (z.B. Stuhl)
- Unterscheidung von Objekten und deren Bestandteilen (z.B. Lehne, Beine)
- Vererbungshierarchie von Objekten (Stuhl erbt Eigenschaften von Möbel)

Der Konsens bei Objekten rührt vielleicht daher, dass Objekte eben ganz natürliche Gegenstände und Konstrukte unserer Alltagswelt sind und wir diese bereits tagtäglich in der üblichen Kommunikation benennen. Wir sprechen von den Kunden, von den Aufträgen, etc.

Bei den Prozessen haben wir Mühe, uns über bestimmte Namens- bzw. Ordnungskriterien zu einigen. Ein Geschäftsprozess “Operativer Einkauf” ist weniger klar als ein Objekt “Kunde”.

Es wird gelegentlich argumentiert: “Die einzigen Personen, welche nicht objektorientiert denken, sind die Informatiker”. Die Objektsicht ist in gewisser Weise unabhängig von den beiden anderen Sichten: Ein Objekt “Kunde” kann in verschiedenen Prozessen (beispielsweise im Verkaufs- und im Serviceprozess) vorkommen – aber auch in unterschiedlichen Funktionen (z.B. in der Funktion “administrieren Finanzen” und in der Funktion “verkaufen Produkte”).

## 5.2 Objektorientierte Methoden im Überblick

Objektorientierte Methoden stellen wie erwähnt die “Bearbeitungseinheiten” eines sozio-technischen Systems in den Mittelpunkt. Sie gehen von den Objekten aus und betrachten Beziehungen zu den übrigen Aspekten. In Entsprechung zu den anderen Methoden gibt es auch hier Methoden, die die Objekte primär betrachten (schattiert) und solche, welche Objekte als sekundären Aspekt untersuchen (s. Abb. 49).

Im Zentrum der objektorientierten Modellierung steht das Objektmodell. Es ordnet die Gesamtheit der Objekte hierarchisch und zeigt damit die gegenseitigen Beziehungen zwischen den Objekten auf.

Das Zustandsübergangsdiagramm zeigt sodann Prozess-Beziehungen zwischen Objekten auf.

Das Class-Responsibilities-Collaborators Diagramm zeigt Beziehungen – diesmal aber zwischen den Objekten und den Funktionen.

Das Kollaborationsdiagramm bildet die Beziehungen zwischen den Objekten und den Aufgaben ab. Es vermerkt, welche Objekte von welchen Aufgaben tangiert werden und wie die Objekte zu diesem Zweck untereinander kollaborieren.

<div><div>primär</div><div>sekundär</div></div>		Hauptfokus ↓		
		Prozesssicht	Funktionssicht	Objektsicht
Prozesssicht			Zustands- übergangs- diagramm	
Funktionssicht			Class- Responsibilities- Collaborators	
Objektsicht	Sequenz- diagramm	Datenfluss- Diagramm	Objektmodell	Arbeitsobjekt- diagramm
Aufgabensicht			Kollaborations- diagramm	

Abb. 49: Primäre und sekundäre Modellierung der Objekte

### 5.3 Objektmodell

*Definition: Objektmodelle stellen die hierarchischen Strukturen sowie die Beziehungen zwischen den Objekten eines Systems dar.*

Gemäß unserer Definition zeigen Objektmodelle sowohl primär als auch sekundär Objekte auf. Man wird also eine Auswahl von Objekten treffen und ihnen sodann ihre verbundenen Objekte zuweisen. Denn auch Objekte können sehr wohl "enge" Beziehungen zueinander unterhalten. Das Objekt "Adresse" hat beispielsweise eine Beziehung zum Objekt "Auftrag" (s. Abb. 50). Die unten stehende Notation zur Darstellung eines Objektes enthält im oberen Teil die Bezeichnung des Objektes und im unteren die entsprechenden Elementarfunktionen (Methoden). Leider ermöglichen solche Objektmodelle den Dialog mit Benutzern nur schlecht.

Objekte können einwertige oder mehrwertige Beziehungen aufweisen. Wenn ein Kunde mehrere Aufträge erteilt, dann besteht eine mehrwertige Beziehung zu den Aufträgen. Man spricht in diesem Zusammenhang von Kardinalitäten und von einer "1 zu m"-Beziehung. Man unterscheidet die Beziehungen: "1 zu 1", "1 zu m" und "m zu m" (wobei m den Wert "∅" annehmen kann). Die unten stehende Darstellung zeigt eine "einfache Beziehung" zwischen der Adresse und dem Auftrag. Auf die weiteren Beziehungen, die hierarchischer Art und mit zusätzlichen Symbolen versehen sind, wird im nächsten Abschnitt eingegangen.

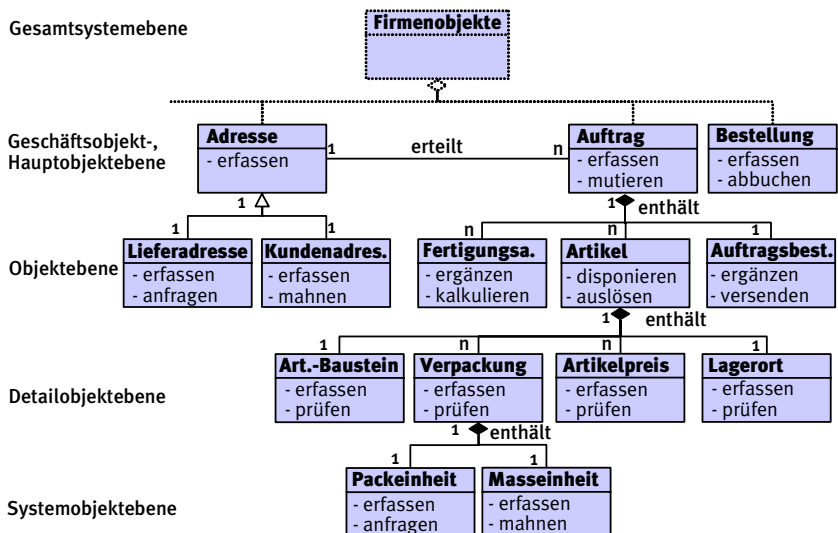


Abb. 50: Hierarchien und Beziehungen der Objekte im Objektmodell

### 5.3.1 Bestandteilhierarchien

Nach den “einfachen Beziehungen” kommen wir zu den Hierarchien. In gewisser Weise sind Hierarchien – wie oben gesehen – speziell ausgeprägte Arten von Beziehungen. Objekte können also – ganz analog wie die Funktionen und Prozesse – hierarchisch geordnet werden.

Betrachten wir dies zunächst am Beispiel des Objektes “Auftrag für Fahrrad” (s. Abb. 51). Ein Fahrrad kann sich ganz offensichtlich aus mehreren Bestandteilen zusammensetzen – Kinder und Väter probieren das gerne selbst aus. Eine solche Hierarchie wird Bestandteilhierarchie – in der Unified Modeling Language (UML) “Komposition” genannt (UML 2003).

Bestandteilhierarchien können sodann zwei Ausprägungen haben. Nehmen wir an, in der unten stehenden Abbildung handle es sich um Objekte, die im Informationssystem abgebildet seien: Der schwarz ausgefüllte Rhombus besagt, dass die Lebensdauer des Objektes “Auftrag für Fahrrad” der Lebensdauer seiner Teilaufträge entspricht. Wenn der Auftrag gelöscht wird, dann werden auch die Positionen (nicht die Teile) gelöscht.

Auf der anderen Seite enthält eine Adresse eine Ortschaft. Die “Ortschaft” darf nicht gelöscht werden, wenn das Objekt “Kundenadresse” gelöscht wird. Ortschaften bestehen unabhängig von Adressen. Dieser Umstand wird durch die schwache Beziehung, einen unausgefüllten Rhombus, dargestellt. Sie wird in der UML-Sprache als “Aggregation” bezeichnet.

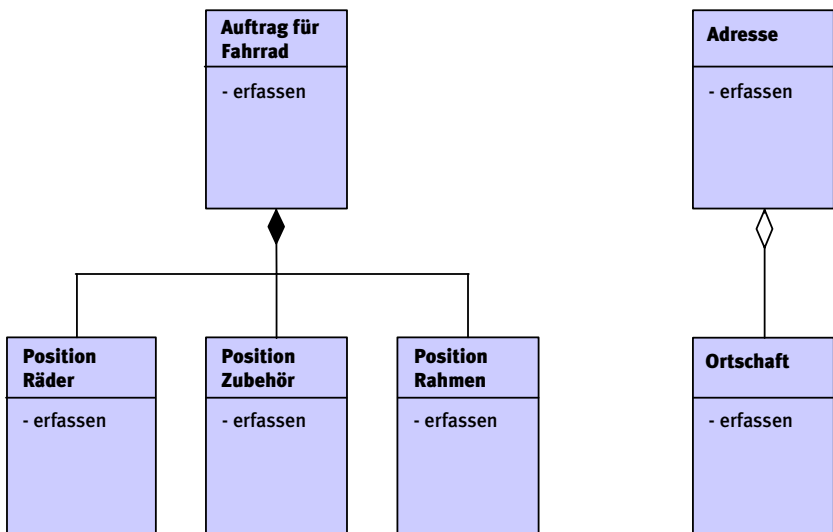


Abb. 51: Zwei Formen von Bestandteil-Hierarchien

### 5.3.2 Vererbungshierarchien und Klassendiagramme

Eine zweite, anders gelagerte Hierarchie betrifft jene der Vererbung.<sup>1</sup> Objekte können von anderen Objekten abgeleitet werden (s. Abb. 52). Ein abgeleitetes Objekt übernimmt dann in der Regel alle Eigenschaften des Mutterobjektes und erhält noch einige spezifische neue Eigenschaften hinzu. Mit anderen Worten: In unserem Beispiel erhält das Objekt "Lieferant" die Eigenschaften des Objektes "Firma", hat aber noch einige zusätzliche Eigenschaften. Beispielsweise wird beim "Lieferanten" z.B. eine Lieferantenummer als zusätzliches Attribut notwendig sein.

Die Grafik zeigt dagegen nicht die so genannten Mehrfachvererbungen. Dies bedeutet, dass ein Objekt (Klasse) von mehreren anderen Objekten gewisse Eigenschaften erben kann (zur genaueren Abgrenzung zwischen Klassen und Objekten s. weiter unten). Ein Objekt "Adressetikette" kann beispielsweise Eigenschaften sowohl von den "Etiketten-Objekten" als auch von den "Adressen-Objekten" erben.

Wir schließen hier die Ausführungen zum Objektmodell ab und verweisen auf die Literatur (z.B. Schönsleben 2001).

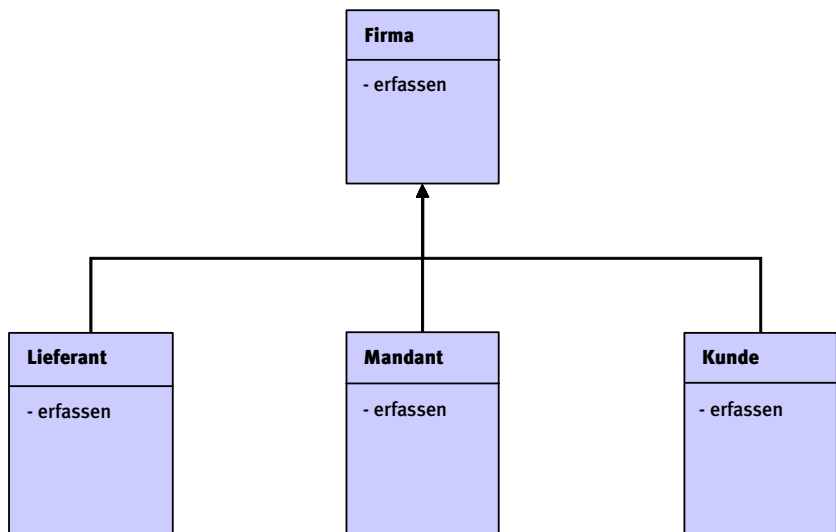


Abb. 52: Vererbungsbeziehungen

<sup>1</sup> Besser würde der Begriff "Spezialisierung" verwendet, da keine "Vater-Sohn"-Beziehung besteht. Vielmehr muss von einer "is-a"-Beziehung gesprochen werden.



## 5.4 Identifikation und Systemabgrenzung der Objekte

Bei Prozessen und bei Funktionen müssen auch bei den Objekten die Fragen beantwortet werden, wie diese a) identifiziert werden und b) welche Objekte innerhalb des betrachteten Systems liegen.

Wenden wir uns zunächst der Frage zu, wie die Objekte eines gegebenen Systems identifiziert werden können. Primär könnten die Objekte aus den Operationen abgeleitet werden, da diese ja gerade als ihre “Bearbeitungseinheiten” definiert wurden. In den seltensten Fällen liegt aber die Gesamtmenge der Operationen für diese Bottom-up-Analyse vor.

In Analogie zur funktionalen und prozessorientierten Analyse kann natürlich auch Top-down vorgegangen werden, indem ausgehend von komplexen Objekten, wie beispielsweise Kundenauftragsobjekten, Produktionsaufträgen, Adressobjekten, Vertriebsprodukten, etc. die hierarchisch tiefer liegenden Objekte abgeleitet werden.

Es ist sodann möglich, die Objekte aufgrund derjenigen Modelle abzuleiten, welche Objekte als sekundären Aspekt betrachten – z.B. ausgehend von den bereits betrachteten Sequenzdiagrammen, Datenflussdiagrammen und den noch zu betrachteten Arbeitsobjektdiagrammen. Greifen wir das Beispiel der Datenflussdiagramme auf. Dort werden alle “Datenspeicher” des Systems angeführt. Damit ist bereits ein initialer Datenkatalog erstellt.

Die Objekte zu finden wird erschwert durch ihre große Anzahl – in einem Industrieunternehmen kann es gut und gerne über 100 verschiedene Klassen geben. Bei den Geschäftsprozessen liegt die Anzahl mit zwischen 10 und 20 Elementen erheblich tiefer und erscheint leichter bewältigbar. Allerdings lässt sich auch diese Aussage relativieren, falls man nur die komplexeren Geschäftsobjekte betrachtet, von denen es vielleicht je nach Klassenhierarchiestruktur zwischen 20 bis 30 gibt.

Glücklicherweise kann also auch bei Objekten zwischen relevanten Geschäftsobjekten und eher untergeordneten Objekten unterschieden werden. Geschäftsobjekte wären Objekte wie “Auftrag, Kunde, Bestellung”, etc. Zu den Systemobjekten könnten zählen: Lieferkonditionen, Währungsobjekte, Mengeneinheitsobjekte, Lohnarten, etc.



## 5.5 Spezifikation von Objekten

In gewisser Weise geht es bei der Beschreibung und der Spezifikation von Systemen immer um dieselbe Fragestellung: Systeme – egal ob dies nun Prozesse, Funktionen oder ein Objekte sind – besitzen eine innere Struktur, die beschrieben bzw. spezifiziert werden muss – bei Objekten sind dies die Attribute und die Methoden (Funktionen). Außerdem müssen die Beziehungen nach außen dargelegt werden. Ein Aspektsystem – sei dies nun wiederum ein Prozess, eine Funktion oder ein Objekt bzw. eine Stelle – hat nämlich immer auch Beziehungen zu seinen Umsystemen.

Bis jetzt haben wir den Begriff “Objekt” etwas unscharf verwendet. In der objektorientierten Sprache unterscheidet man zwischen dem Begriff der “Klasse” und den “Objekten”. Nach dieser Terminologie stellen konkrete Objekte “Instanzen” einer allgemeinen Klasse dar.

Eine Klasse beschreibt eine Menge von Objekten mit gleichen Eigenschaften (s. Abb. 53). Spricht man von einem ganz bestimmten Exemplar einer Klasse, also beispielsweise von “Hans Müller”, geboren am 3. Februar 1969, dann spricht man von einer “Instanz” der Klasse “Kunde” bzw. vom Objekt “Müller”.

Die nachfolgende Darstellung zeigt eine Klasse in der Notation der Unified Modeling Language (UML 2003). Neben der Klassenbezeichnung werden deren beschreibende Merkmale – (Daten-)“Attribute” – angeführt. Außerdem werden die Elementarfunktionen – “Methoden” – angegeben.

Klasse	<b>Katze</b>	Objekt	<b><u>Mikesch: Katze</u></b>
Attribute	Kennung: Integer Name: Text Strasse: Text Nummer: Integer Ort: Text Postleitzahl: Integer	Attribute	o6748 Mikesch Hundsweg 3 Hausen 45679
Methoden	- füttern () - baden () - steicheln () - singen () - ...	Methoden	- füttern - baden - streicheln - singen - ...

Abb. 53: Unterscheidung von “Klasse” und dem instantiierten “Objekt”

### 5.5.1 Weitergehende Spezifikation von Objekten

Sind die Objekte erst einmal identifiziert, dann müssen sie näher spezifiziert werden. Das Vorgehen zur Spezifikation von Objekten ist analog dem bereits geschilderten Vorgehen.

#### **Abgrenzung und Beziehungen zu Umsystemen**

In einem ersten Schritt ist festzuhalten, zu welchen anderen Objekten bzw. Umsystemen Beziehungen bestehen. Diese Information lässt sich einerseits dem Objektmodell entnehmen, andererseits ergeben sich hierzu auch Angaben aus einer allenfalls vorhandenen Systemabgrenzung.

#### **Inhalte eines Objektes und aggregierte Subobjekte**

Sodann ist die "innere Struktur" eines Objektes zu beschreiben. Die in einem Objekt enthaltenen bzw. aggregierten Subobjekte lassen sich aus dem Objektmodell ableiten (s. Objektmodell). Zudem werden die inneren Strukturen durch die oben erwähnten Attribute (die selber wieder Objekte darstellen können) und die Methoden (Elementarfunktionen) des Objektes beschrieben.

#### **Mengen, Häufigkeiten und Regeln**

Neben objektbezogenen Angaben von Mengen und Häufigkeiten könnten zur Spezifikation von Objekten nun ebenfalls Regeln hinsichtlich des Objektes beschrieben werden.

#### **Beziehungen zu anderen Aspektsystemen**

Zur Spezifikation eines Objektes gehört, seine Beziehungen zu den übrigen Aspektsystemen aufzuzeigen. Dies sind die Beziehungen zu den Prozessen, Funktionen und Aufgaben. Diese Beziehungen werden durch die entsprechenden Diagramme, dem Zustandsübergangs-, Class-Responsibilities-Collaborators- und dem Kollaborationsdiagramm spezifiziert.

Damit besteht auch in der objektorientierten Spezifikation ein ganz wesentlicher Teil aus den weiter unten beschriebenen Systemdiagrammen.

## 5.6 Zustandsübergangsdiagramm

*Definition: Zustandsübergangsdiagramme stellen primär Objekte dar und sekundär die Prozesse, welche diese Objekte tangieren.*

Zustandsübergangsdiagramme (engl. “State Transition Diagrams”; State Diagram) stellen ausgewählte Objekte des Objektmodells dar (z.B. Subobjekte eines Auftrages). Die in der Literatur angeführten Beispiele beschränken sich meist auf Darstellungen aus der Designphase. Die Notation abgerundeter Rechtecke für Zustände entspricht UML (UML 2003).

Die nachfolgende Abbildung zeigt am Beispiel von verschiedenen Zuständen eines Auftragsobjektes, wie und in welcher Reihenfolge das Auftragsobjekt durch Teilprozesse tangiert wird. Zur besseren Visualisierung der Auftragszustände ist aus didaktischen Gründen zusätzlich illustriert, welche Subobjekte in jedem Zustand an das Auftragsobjekt angefügt werden.

Gewisse Subobjekte des Auftragsobjektes entstehen erst im Verlauf des Prozesses, und bei anderen verändert sich nur deren Zustand im Verlauf des Prozesses. Neue Subobjekte und Zustände können entstehen, wenn der Auftragsabwicklungsprozess einen neuen Status erhält, oder aber, wenn einem Auftragsobjekt ganz einfach ein neues Objekt zugefügt wird.

Zustandsübergangsdiagramme können in dieser Form also durchaus auch zur Spezifikation von Benutzeranforderungen aus Anwendersicht oder zur Dokumentation von objektorientierten Anwendungen dienen.

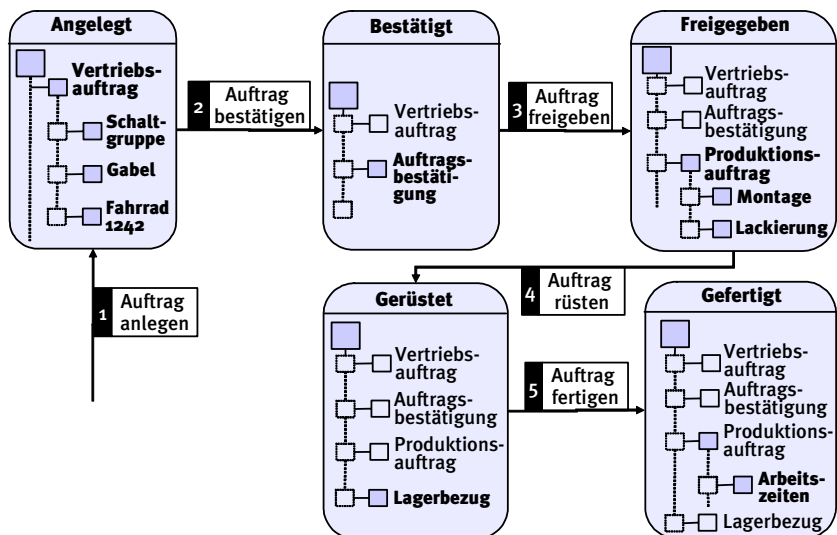


Abb. 54: Zustandsübergangsdiagramm

## 5.7 Class-Responsibilities-Collaborators-Diagramm (CRC)

*Definition: Class-Responsibilities-Collaborators-Diagramme stellen primär Objekte dar und sekundär jene Funktionen, für welche diese “verantwortlich” sind.*

Ausgehend von einer Anzahl Objekte (Klassen) wird in einem CRC-Diagramm angegeben, für welche Funktionen diese Objekte “hauptverantwortlich” zeichnen (Beck & Cunningham 1989). CRC-Diagramme ermöglichen es, den Objekten des Objektmodells die “hierarchisch höheren” Funktionen des Funktionsmodells zuzuweisen (s. Abb. 55). Höhere Funktionen tangieren meistens mehrere Geschäftsobjekte und lassen sich daher auch nicht 1:1 einem Objekt im Sinne einer Methode zuordnen. Es muss neben der “Hauptverantwortung” zusätzlich vermerkt werden, welche Objekte zur Erfüllung dieser Verantwortung “kollaborieren”.

Ein CRC-Diagramm in der hier vorliegenden Notation zeigt also explizit auf, welche Objekte zur Erfüllung einer bestimmten Funktion notwendig sind. Es wird angegeben, welche Kollaboratoren für welche Funktionen bestehen. Entsprechend lassen sich nun ausgewählten Objekten höhere Funktionen zuordnen und ergänzend die zur Erfüllung der Funktion erforderlichen Beziehungen zwischen den Objekten darstellen (die Linien sind zum Text redundant). Somit ist man in der Lage, bereits in den frühen Projektphasen die Funktionen und die Objekte einander zuzuordnen.

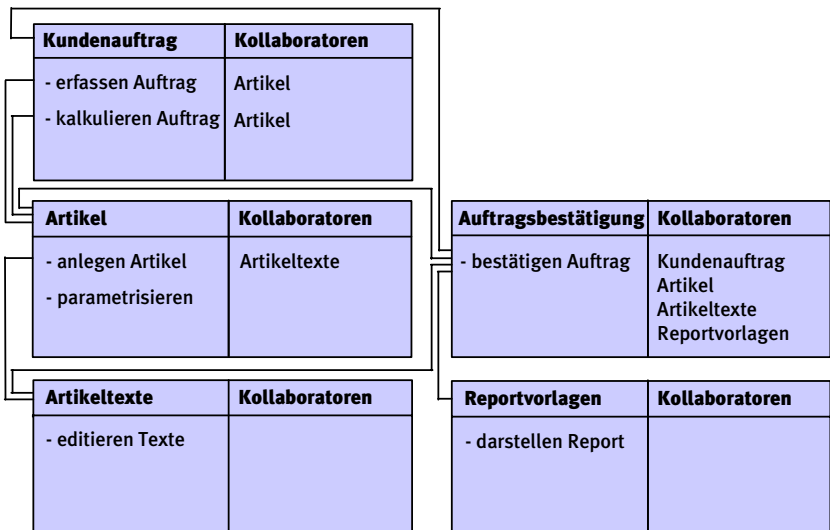


Abb. 55: Class-Responsibilities-Collaborators-Diagramm

## 5.8 Kollaborationsdiagramm

*Definition: Kollaborationsdiagramme (engl.: “Collaboration Diagrams”) stellen primär Objekte dar und sekundär die Beziehungen dieser Objekte zu den Stellen (Aktoren) und ihr gegenseitiges Zusammenspiel.*

Das Kollaborationsdiagramm stellt demnach eine Anzahl von Objekten einer ausgewählten Hierarchieebene in den Vordergrund und weist ihnen jene Aufgaben/Stellen zu, die auf diese Objekte zugreifen bzw. diese manipulieren (s. Abb. 56). Die Notation entspricht wieder UML (UML 2003).

In gewisser Weise entspricht dieses Diagramm der objektorientierten Version des Use-Case-Diagramms. Beim funktionsorientierten Use-Case wird das Zusammenspiel von Funktionen in Bezug zu Aktoren dargestellt. Hier wird Zusammenspiel von Objekten aufgezeigt. In der Literatur ist das Kollaborationsdiagramm allerdings nicht immer ganz klar definiert. Es wird meist auch einfach dazu benutzt, das Zusammenspiel einer Menge von beliebigen Objekten aufzuzeigen. In diesem Sinne ist der Übergang zu anderen Diagrammen oft unscharf (Oestereich 2001).

Gemäß unserer etwas präziseren, aber dadurch auch engeren Definition zeigt das Kollaborationsdiagramm ausgehend von zusammengehörigen Objekten weiter auf, welche unterschiedlichen Stellen dieselben Objekte bearbeiten. Dies kann Rückschlüsse auf die Komplexität und Interaktionsfelder innerhalb des Systems erlauben.

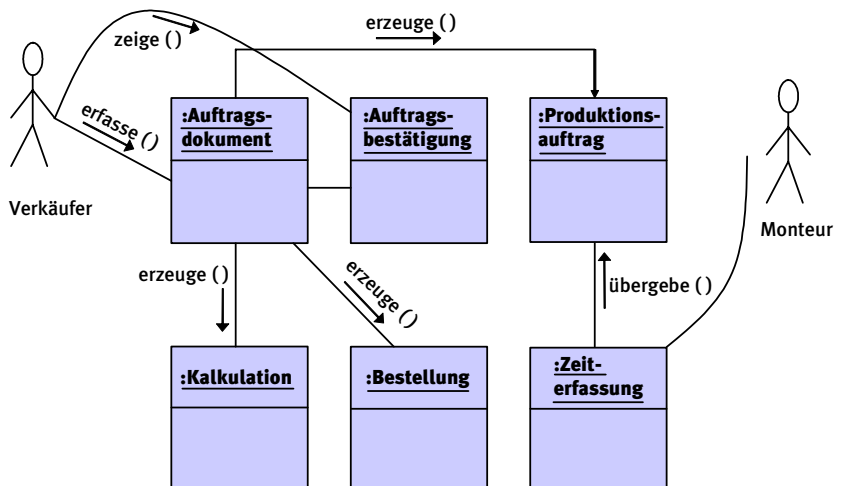


Abb. 56: Kollaborationsdiagramm

## 5.9 Integrierte objektorientierte Methode: UML und RUP

Die Unified Modeling Language (UML) ist aus einer Synthese verschiedener objektorientierter Methoden hervorgegangen. Vereinigt wurden die Methoden von Booch (1991), Rumbaugh (1991) und Jacobson (1992) zur UML (Booch et al. 1999). In der Folge wurde von der Firma Rational Inc., der die drei Autoren angehören, der Softwareentwicklungsprozess "Rational Unified Software Development Process" ("RUP") propagiert.

Obwohl die Methode "unified" genannt wird, liegt ihr Fokus auf dem objektorientierten Aspekt (s. Abb. 57). So sind gemäß dieser Methode nur die Use-Cases und das stellenorientierte Ablaufdiagramm, welches in der UML "Tätigkeitsdiagramm" genannt wird, anderen Sichten zuzuordnen.

Damit eignet sich der RUP nur beschränkt zur umfassenden sozio-technischen Modellierung von Informationssystemen und hat sein Schwergewicht immer noch auf dem Design und der Implementation. Die Modellierung der Prozesssicht und insbesondere die Modellierung der Aufgabensicht kommen zu kurz. Die Verwendung der Tätigkeitsdiagramme und der Use-Case-Diagramme kann das Manko im Bereich der Prozessmodelle und im Bereich der Aufgabenmodellierung nicht wettmachen. Es ist daher zu empfehlen, beim Einsatz von UML frühe Phasen durch weitere Methoden anzureichern. Der Einsatzzweck von UML ist zwangsläufig auf den Bereich implementierungsnaher Designphasen eingeschränkt.

		Hauptfokus ↓		
primär ↙ ↘ sekundär	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht			Zustands- übergangs- diagramm	
Funktionssicht				
Objektsicht	Sequenz- diagramm		Objektmodell	
Aufgabensicht	Stellenorientiertes Ablaufdiagramm (Tätigkeitsdiagr.)	Use-Case- Diagramm	Kollaborations- diagramm	

Abb. 57: Diagramme der Unified Modeling Language

## 5.10 Integrierte objektorientierte Methode: RAW

Eine zweite integrierte Methode, die auf Basis von SOMA (Graham 2001) entwickelt wurde, ist der “Requirements Analysis Workshop” (RAW). Auch diese Methode bedient sich eines ganz bestimmten Methodensets und navigiert gewissermaßen durch die Matrix. Der RAW gibt aber klare Leitlinien, sich durch die verschiedenen Modelle zu bewegen und zeigt, wie sich diese über die Zeit entwickeln (s. Abb. 58).

Begonnen wird mit dem Prozess (Process Use-Case). Anschließend werden die für diesen Prozess relevanten Funktionen (Use-Cases) definiert. Diese werden sodann mit Hilfe eines Flussdiagramms (Task Objektmodelle) funktional weiter detailliert, wobei pro Task (Use-Case) die hierfür notwendigen Sub-Funktionen erhoben werden. Aus diesem funktionalen Modell werden nachfolgend Objekte abgeleitet (Objektmodell).

In einem nächsten Schritt werden diesen Objekten mit Hilfe von CRC-Karten die Verantwortlichkeiten (Funktionen je Objekt) zugeordnet. Zum Schluss wird der Prozess durch Sequenzdiagramme (Interaktionsdiagramme) nochmals abgebildet – nunmehr auf Basis der gefundenen Objekte. Mit diesen sechs Modellen werden also alle Sichten in gewisser Weise abgedeckt: Prozess-, Funktions-, Objekt- und sekundär die Aufgabensicht. Diese Methode hat ein solides Fundament, muss aber Lücken im Bereich der sozio-technischen Modellierung hinnehmen.

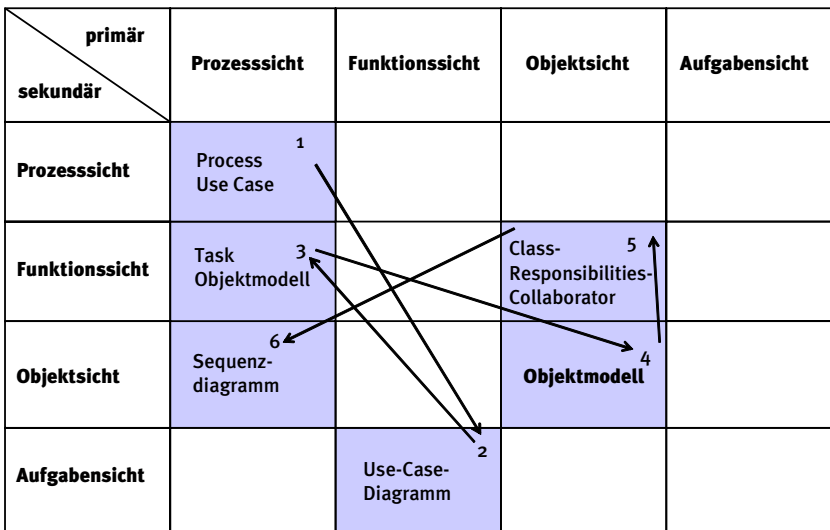


Abb. 58: Methoden und Vorgehen des RAW

### 5.11 Übung: Objektorientierte Modellierung

Auch an dieser Stelle soll eine kurze Case Study durchgeführt werden. Der Case betrifft den bereits eingeführten Fahrradhersteller.

Auf der Darstellung sehen Sie die bekannte Prozessabfolge mit 14 Operationen, hinzugefügt wurden noch Operationen eines beispielhaften Serviceprozesses (s. Abb. 59).

Versuchen Sie nun bitte, aufgrund dieser beiden Angaben die relevanten Objekte zu extrahieren und allenfalls sogar ein Objektmodell abzuleiten.

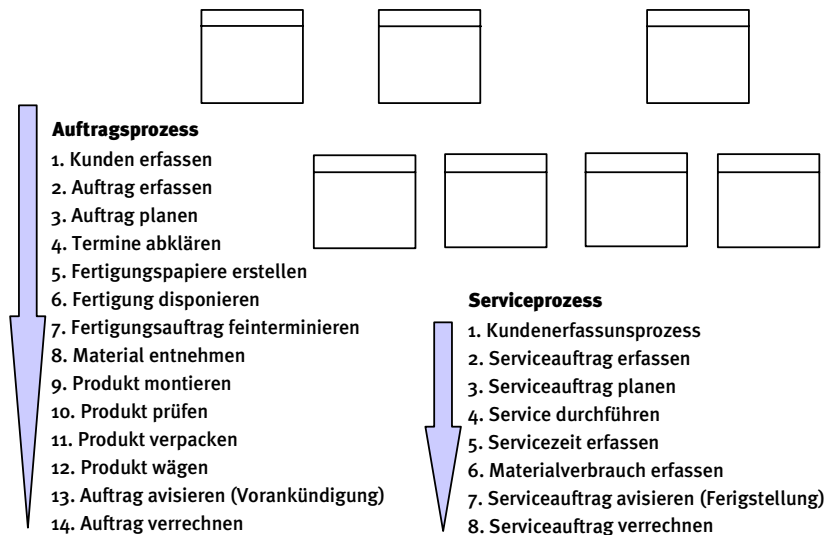
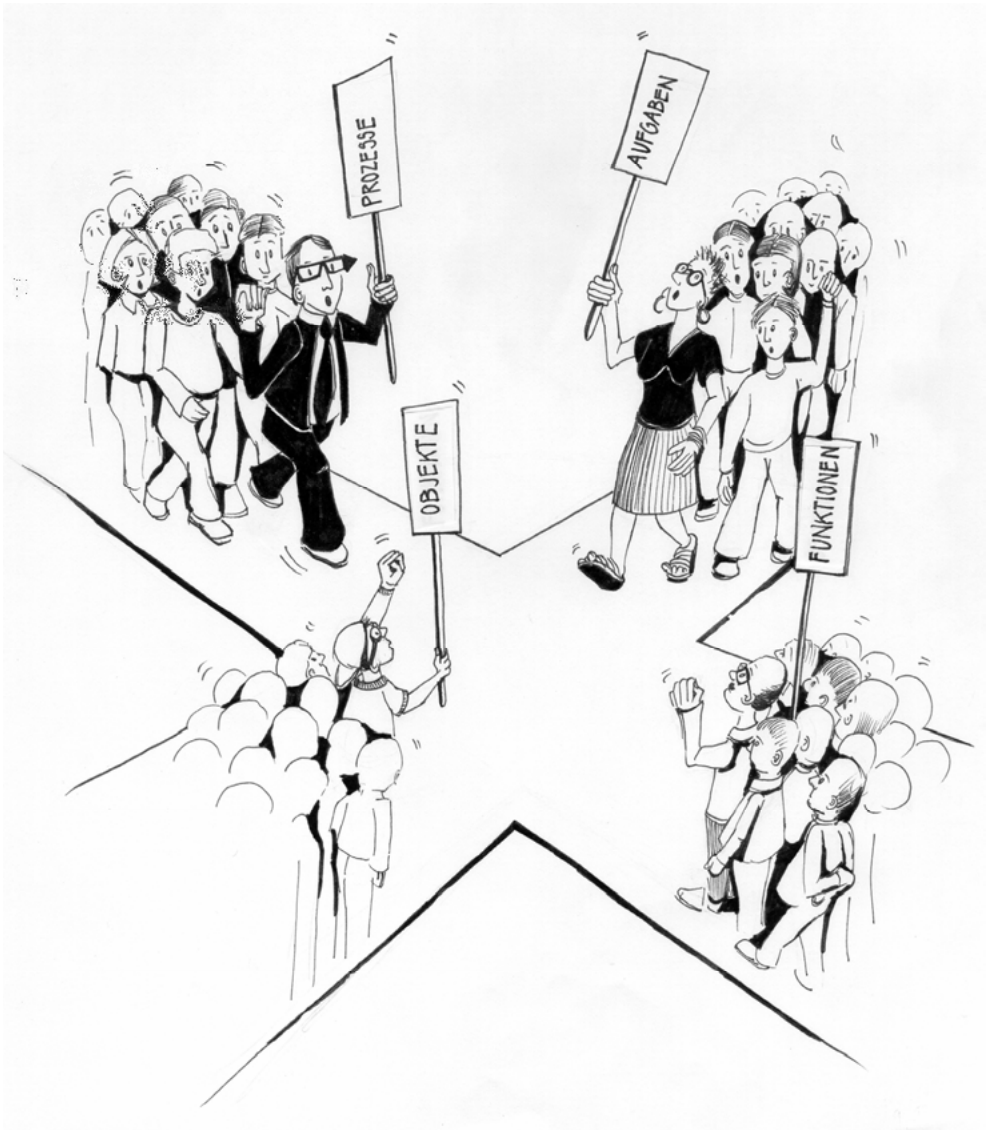


Abb. 59: Übung zur objektorientierten Modellierung



## 5.12 Literatur zur objektorientierten Modellierung

- Beck K., Cunningham W. (1989). A Laboratory for Teaching Object-Oriented Thinking. ACM SIGPLAN Notices vol. 24 (10).
- Booch, G. (1991): Object oriented design with applications. Redwood City, CA: Benjamin/Cummings.
- Booch, G., Rumbaugh, J., Jacobson, I. (1999): The Unified Modeling Language User Guide. Reading, MA: Addison Wesley Longman.
- Coad, P., Yourdon, E. (1991, 2<sup>nd</sup> ed.): Object-Oriented Analysis. Englewood Cliffs, New Jersey: Yourdon Press/Prentice Hall.
- Cox, B., Novobilski, A. (1991, 2<sup>nd</sup> ed.): Object-oriented programming: an evolutionary approach. Reading, MA: Addison-Wesley.
- Graham, I. (2001): Object Oriented Methods. Reading, MA: Addison-Wesley.
- Jacobson I. (1992): Object-Oriented Software Engineering: A Use Case Driven Approach. Reading, MA: Addison-Wesley.
- Jacobson I., Booch, G., Rumbaugh J. (1999): The Unified Software Development Process. Reading, MA: Addison Wesley Longman.
- Oestereich, B. (2001): Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified Modeling Language, 5. völlig überarbeitete Auflage. München: Oldenbourg.
- Rumbaugh, J. (1991): Object-oriented Modeling and Design. Englewood Cliffs, NJ: Prentice Hall.
- Taylor, D. A. (1990): Object Oriented Technology: A Manager's Guide. Reading, MA: Addison-Wesley.
- Schönsleben, P. (2001): Integrales Informationsmanagement: Informationssysteme für Geschäftsprozesse – Management, Modellierung, Lebenszyklus und Technologie. Berlin, Heidelberg, New York: Springer.
- UML (2003): OMG Unified Modeling Language Specification. Needham, MA: Object Management Group.
- White, I. (1994): Using the Booch Method: A Rational Approach. Redwood City, CA: Benjamin/Cummings.



# 6

## Modellierung der Aufgaben

### Kapitelinhalt

- Aufgaben als personenbezogene Zuweisung von Operationen
- Organisationsmodelle
- Spezifikation von Aufgaben und Stellenbeschreibungen
- Modellierungsmethoden und Diagramme der Aufgabensicht

Dieses Kapitel behandelt die Modellierung aus Sicht der Aufgaben bzw. Organisation. Zunächst wird ausgeführt, was unter dem Begriff der Aufgabe zu verstehen ist und wie sich die Modellierung der Aufgaben bezüglich der Modellierung von Prozessen und Funktionen abgrenzt.

Die aufgabenorientierte Sichtweise besteht darin, die Aufgaben in den Mittelpunkt zu stellen, die ihrerseits von einzelnen oder mehreren Personen wahrgenommen werden.

Entsprechend den anderen Sichtweisen können Aufgaben auch in eine hierarchische Form gebracht werden. Dies geschieht üblicherweise in Form eines Organigramms, welches in aller Regel Mehrpersonenaufgaben darstellt. Dieses aufgabenorientierte Kapitel folgt somit der Struktur der vorangegangenen Kapitel: Zunächst wird das Organigramm dargestellt und es wird auf die Charakterisierung bzw. Spezifikation von Aufgaben eingegangen. In einem zweiten Block werden wir uns den aufgabenorientierten Modellierungsmethoden zuwenden und zwei neue Diagramme postulieren, die sich aus Sicht der Vollständigkeit unseres Modells ergeben.

## 6.1 Aufgabenorientierte Modellierung

### 6.1.1 Aufgabensicht

*Definition: Die Aufgabensicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der stellen- bzw. personenbezogenen Zuordnung der Operationen.*

*Definition: Aufgaben bezeichnen zusammengehörige und in sich relativ geschlossene Einheiten von "Stellen zugeordneten Operationen" der unterschiedlichen Systemniveaus.*

Stellen und Aufgaben sind an die Personen des sozio-technischen Systems gebunden und können sowohl als Einzelpersonen- als auch als Mehrpersonenaufgaben definiert sein. Die Modellierung der Aufgabensicht ist nicht mit der Funktions- bzw. Prozesssicht zu verwechseln (s. Abb. 6o). Dies zeigt der Umstand, dass eine Stelle häufig mehrere Funktionen wahrnehmen muss, die ganz unterschiedlichen Hauptfunktionen zugehören.

Es ist im Gegenteil das Anliegen "arbeitsorientierter" Ansätze (Ulich 2001), Aufgaben derart zu gestalten, dass die Arbeitsbedingungen motivierend und persönlichkeitsfördernd sind. Weder die einseitige Ausrichtung der Aufgaben an Funktionen im Sinne einer funktionalen Arbeitsteilung noch die Ausrichtung an Prozessen erfüllen dieses Kriterium.

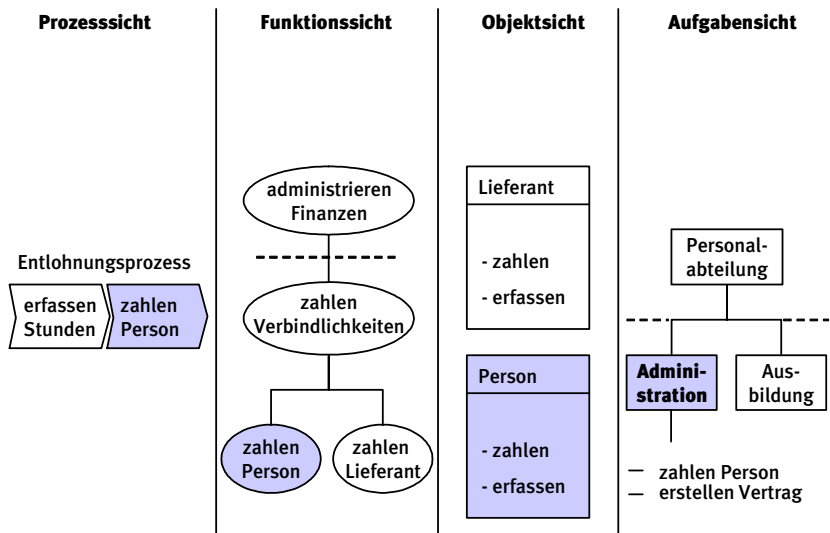


Abb. 6o: Aufgaben als viertes Gliederungskriterium von Operationen

## 6.2 Aufgabenorientierte Methoden im Überblick

Historisch gesehen wurden Aufgaben zuerst durch F.B. Gilbreth im Jahre 1911 und F.W. Taylor (1913) sowie durch weitere Vertreter des Scientific Managements analysiert. Die Verfahren waren auf die Beschreibung der motorischen Tätigkeiten ausgerichtet. Vollständigere Ansätze zur Analyse und Spezifikation von Aufgaben entstanden erst in den 50er- und 60er-Jahren (z.B. Trist & Bamforth 1951). Aufgabenorientierte Methoden stellen demnach die “Aufgaben” eines Systems in den Mittelpunkt.

Im Zentrum der aufgabenorientierten Modellierung steht das “Aufgabenmodell” – hier allerdings Organisationsmodell genannt (s. Abb. 61). Dieses ordnet die Gesamtheit der Aufgaben in eine hierarchische Struktur. Es stellt damit die gegenseitigen Beziehungen zwischen den Aufgaben dar. Der stellenorientierte Informationsfluss zeigt sodann Prozess-Beziehungen zwischen Aufgaben auf.

Das Stellenfunktionendiagramm widmet sich den Beziehungen der Aufgaben und Funktionen. Es konnte – anders als die übrigen 14 Diagramme – trotz Literaturrecherche nicht eruiert werden und wird daher neu postuliert. Dasselbe gilt für das Arbeitsobjektdiagramm, welches sich ebenfalls aus der Morphologie ergibt, aber auch nicht aufgefunden werden konnte. Es zeigt primär Aufgaben und sekundär die diesen Aufgaben zugrunde liegenden Objekte.

<div> <div>primär</div> <div>sekundär</div> </div>					
		Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht
Prozesssicht					Stellenorientierter Informationsfluss
Funktionssicht					Stellenfunktionendiagramm
Objektsicht					Arbeitsobjektdiagramm
Aufgabensicht		Stellenorientiertes Ablaufdiagramm	Use-Case-Diagramm	Kollaborationsdiagramm	Organisationsmodell

Abb. 61: Primäre und sekundäre Modellierung der Aufgaben

### 6.3 Organisationsmodell

*Definition: Organisationsmodelle stellen die hierarchischen Strukturen und Beziehungen zwischen den Aufgaben bzw. Stellen des Systems dar.*

Wenn wir im Folgenden von Organisationsmodellen sprechen, so entsprechen diese nicht unbedingt den verabschiedeten Organigrammen. Es ist nämlich zunehmend der Fall, dass Organigramme nicht die effektiven Aufgaben aufzeigen. Durch den rasanten Wandel entstehen zusätzlich parallele Projektorganisationsstrukturen, die selbstverständlich ebenfalls im Sinne von Stellen und Aufgaben zu betrachten sind.

Im Prinzip ließen sich auch Aufgaben einer “Vererbung” unterziehen. Die Aufgabe einer Operationsschwester bzw. des Intensivstationspflegers liesse sich beispielsweise als spezielle Form einer Schwester bzw. eines Pflegers definieren. Es scheint aber nicht üblich zu sein, Organigramme mit Vererbungsbeziehungen zu versehen.

Aufgaben werden in aller Regel hierarchisch analysiert und können beispielsweise entsprechend der unten stehenden Darstellung auf vier Stufen betrachtet werden (s. Abb. 62) (Strohm et al. 1997).

Das Organisationsmodell darf, wie bereits früher erwähnt, nicht mit dem Funktionsmodell verwechselt werden – auch wenn es in vielen, funktional organisierten Unternehmen diesem ähnliche Grundstrukturen aufweist.

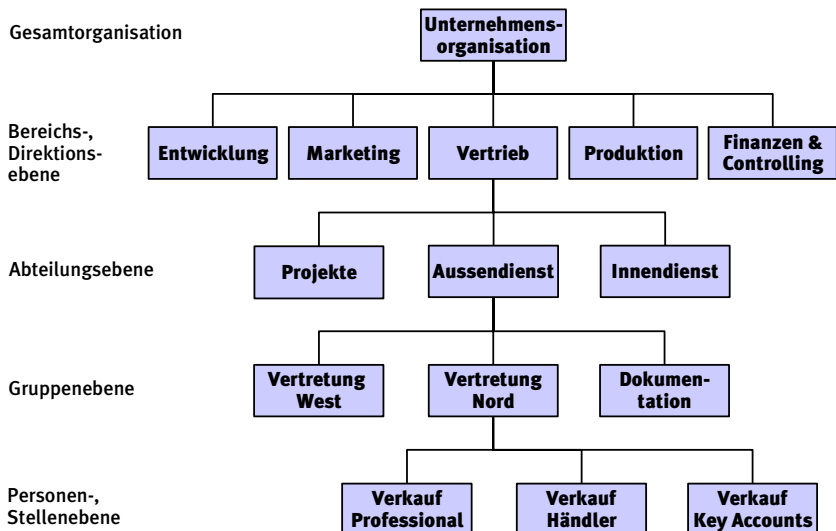


Abb. 62: Organisationsmodell

## 6.4 Identifikation und Systemabgrenzung der Aufgaben

Als viertes Element des Systems müssen nun auch “Aufgaben” identifiziert und spezifiziert werden – sodann muss für diese Aufgaben geklärt werden, welche von ihnen zum analysierten System gehören bzw. davon ausgeschlossen sind. Dies entspricht in vollständiger Analogie der Identifikationsaufgabe im Bereich der übrigen Aspektsysteme.

Man ginge also auch in diesem Fall von den Operationen aus, wiederum mit derselben Relativierung, dass ein Bottom-up-Ansatz infolge der großen Anzahl an Operationen als nicht praktikabel erscheint.

Zum zweiten gibt es den Ansatz, Aufgaben ausgehend von den Personen bzw. aufgrund der bestehenden Organisationsstrukturen festzulegen. Dies führt in aller Regel rascher zum Ziel, allerdings mit der bereits erwähnten Einschränkung, dass ein verabschiedetes Organigramm nicht zwingend den gelebten Aufgaben entspricht und ein zunehmender Anteil an Aufgaben projektbezogen erfolgt. Gerade diese Projekte sind weniger dokumentiert.

Andererseits besteht neben diesen Ansätzen wiederum die Möglichkeit, die Aufgaben aus anderen Sichtweisen abzuleiten. So könnte beispielsweise aufgrund des stellenorientierten Ablaufdiagrammes klar werden, welche Stellen und Aufgaben ein Auftragsabwicklungsprozess beinhaltet. Gleiches gilt für den Use-Case oder das Kollaborationsdiagramm. Alle diese Darstellungen enthalten als sekundären Aspekt Stellen bzw. Aufgaben und müssen im Organigramm abgeglichen werden.

Wenn die Stellen bzw. Aufgaben auf den unterschiedlichen Niveaus erst einmal definiert sind, gilt es festzuhalten, welche dieser Stellen einen Teil des Projektes bilden bzw. in diesem Sinne innerhalb der Systemgrenzen zu liegen kommen.

## 6.5 Spezifikation von Aufgaben

### 6.5.1 Stellenbeschreibung

Der Bereich der Methoden zur Analyse und Spezifikation von Aufgaben hat sich in den vergangenen Jahrzehnten stark entwickelt. Insbesondere die Forschungsgebiete der Arbeitspsychologie und Organisationspsychologie haben diverse Instrumentarien und Methodiken entwickelt, mit welchen Aufgaben analysiert, beschrieben und gestaltet werden können.

Infolge der hierarchischen Struktur der Aufgaben haben sich konsequenterweise Methoden entwickelt, die unterschiedlichen Systemebenen zuzuordnen sind: Techniken zur Analyse von personenbezogenen Aufgaben oder Techniken zur Analyse von Gruppen.

Entsprechend dem hierarchischen Ansatz wird nun eine Reihe von ergänzenden Aufgaben zu einer personenbezogenen Stelle zusammengefasst. Eine Stelle wird in aller Regel durch eine so genannte "Stellenbeschreibung" spezifiziert. Gemäß den allgemeinen Systemmerkmalen wird eine Stellenbeschreibung nach dem Muster des "Stelleninhalts" und der "Beziehungen" zu anderen Stellen beschrieben (s. Abb. 63).

#### **Stelleninhalt**

- Stellenbezeichnung
- Hauptziel und Aufgabenerfüllungsziele der Stelle
- Stellenfunktionen
- Kompetenzen des Stelleninhabers
- Formale Anforderungen an Stelleninhaber

#### **Abgrenzungen und Beziehungen zu anderen Stellen**

- Instanz, der die Stelle unterstellt ist
- Direkt unterstellte Stellen
- Abgrenzung der Verantwortlichkeiten
- Informationsrechte und -pflichten
- Zusammenarbeit mit anderen Stellen

#### **Bedingungen**

- Arbeitsbedingungen
- Budget der Stelle
- Arbeitsort, Arbeitszeit, Urlaub, Vergütungen, Sozialleistungen

Abb. 63: Stellenbeschreibung (in Anlehnung an: Spisak 1999, S. 31)



### 6.5.2 Psychologische Tätigkeitsanalysen

Die psychologische Tätigkeitsanalyse untersucht und beschreibt Aufgaben von Personen, um die Gesetzmäßigkeiten der psychischen Tätigkeitsregulation und deren Probleme bzw. Charakteristiken zu erfassen (Matern 1983, S. 78). Es geht insbesondere um Fragen der zeitlichen Belastung, der Wiederholungshäufigkeit, der Abfolgeform und der verfügbaren Zeit von personenunabhängigen Tätigkeitselementen einer Aufgabe (Ulich 2001).

Die in diesem Sinn verstandene psychologische Tätigkeitsanalyse orientiert sich somit stark an den beobachtbaren, motorischen Tätigkeiten. Andere Verfahren zur Analyse und Bewertung von Tätigkeiten, wie z.B. das VERA-Verfahren (Volpert et al. 1983), beziehen vermehrt Anforderungen an die Denkprozesse mit ein. Ihr Ziel liegt in der Bewertung der Tätigkeit hinsichtlich Faktoren wie beispielsweise der Persönlichkeitsförderlichkeit und Schädigungsfreiheit.

### 6.5.3 Spezifikation der Aufgaben von Arbeitsgruppen

Die eher Personen-bezogenen Modelle entstammen der so genannten Handlungsregulationstheorie, die Gestaltungsmerkmale auf der Ebene von Gruppen entstammen der Theorie über sozio-technische Systeme (Primär- und Sekundäraufgaben).

Von besonderem Interesse bei der Analyse von Aufgaben sind Fragen nach Aspekten wie der Selbstregulation, der relativen Unabhängigkeit von Organisationseinheiten, der Polyvalenz von Arbeitstätigen, etc. Zudem stehen Fragen wie die Folgenden im Vordergrund:

- sequenzielle Vollständigkeit der Aufgabe (Prozessabdeckungsgrad)
- Anforderungskomplexität (Planung, Arbeitszeit, Ausbildung, Zusammensetzung, Koordination)
- Regulation auf der Ebene einer Gruppe: Ausmaß & Häufigkeit, gemeinsame Beurteilung, Planung, Lösungsprozesse und Entscheidungen.

## 6.6 Stellenorientierter Informationsfluss

*Definition: Stellenorientierte Informationsflüsse stellen primär Stellen dar und sekundär die diese Stellen tangierenden Prozesse.*

Aus aufgabenbezogener Sicht interessiert zunächst, welche Art von Schnittstellen durch die verschiedenen Prozesse in und zwischen den Stellen entstehen (s. Abb. 64). Da hierfür keine einheitliche Terminologie auszumachen war, wird im Folgenden der Begriff "stellenorientierter Informationsfluss" verwendet.

Die Informationsbeziehungen zwischen den Stellen kommen aufgrund von Prozessen zustande – von formellen oder aber von informellen Prozessen. Dieses Diagramm wird verwendet, um die Prozessbeziehungen zwischen den Stellen darzustellen. Man erhält ein klares Bild, welche Informationsbeziehungen zwischen den Organisationseinheiten bestehen.

Gelegentlich wird man zusätzlich darstellen, in welcher Sequenz die Stellen im Prozessablauf tangiert werden. Dann wird die Sequenz der Informationsbeziehungen durch die Bezeichnung der Pfeile dargestellt.

Als eingeschränkte Sicht kann man sich auch nur lediglich einen einzigen Hauptprozess herausgreifen (z.B. den Auftragsprozess). Werden allerdings die Beziehungen mehrerer Prozesse gemeinsam dargestellt, so könnte die Nummerierung die Prozessnummer mit beinhalten (s. u.).

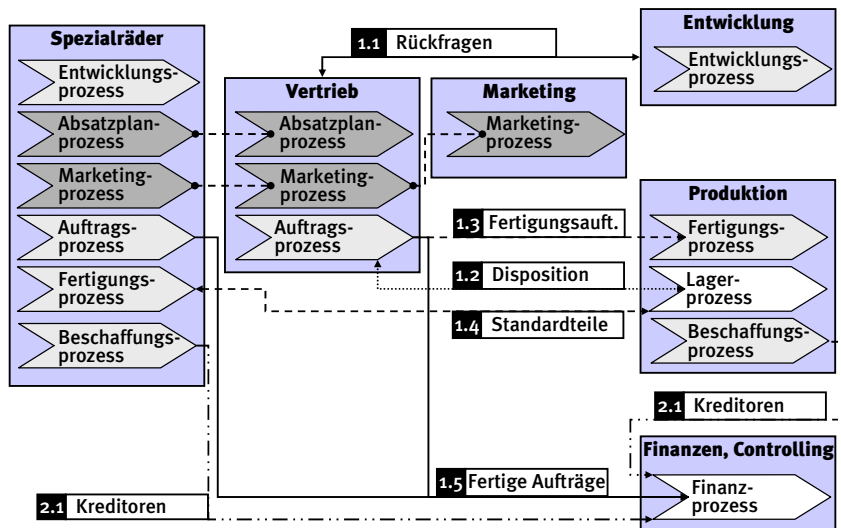


Abb. 64: Stellenorientierter Informationsfluss

### 6.6.1 Historischer Rückblick

Das Diagramm des stellenorientierten Informationsflusses lässt sich mindestens bis auf den Beginn des letzten Jahrhunderts zurückverfolgen (s. Abb. 65). Aller Wahrscheinlichkeit nach wurde dieses Diagramm in den Zeiten des "Scientific Management" entwickelt.

Die nachfolgende Abbildung zeigt einen stellenorientierten Informationsfluss aus dem Jahre 1926 in einer etwas anderen Darstellungsweise als oben.

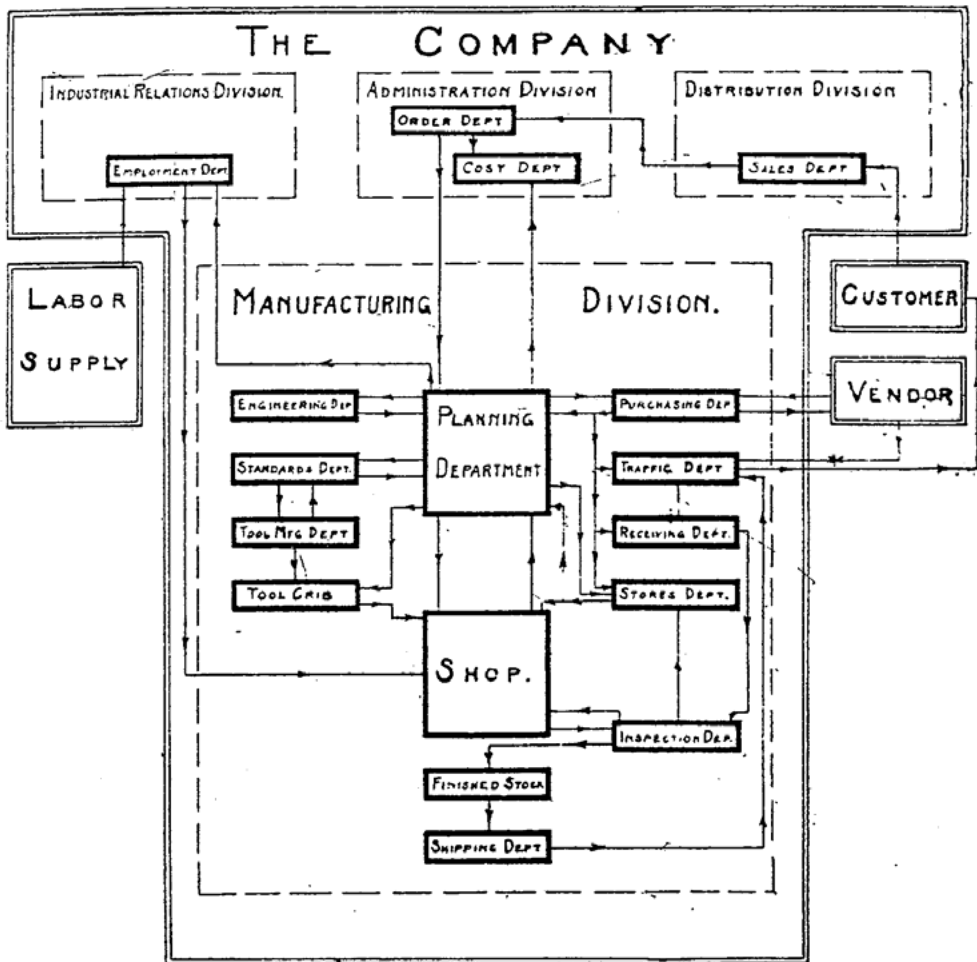


Abb. 65: Stellenorientierter Informationsfluss (Atkins 1926)

## 6.7 Stellenfunktionendiagramm

*Definition: Stellenfunktionendiagramme stellen primär Stellen dar und sekundär die durch diese Stellen wahrgenommen Funktionen.*

Das Stellenfunktionendiagramm zeigt eine Anzahl von Stellen und stellt dar, welche Funktionen gemeinsam bzw. einzeln wahrgenommen werden. In gewisser Weise entspricht die hier postulierte Darstellung der Spiegelung der Use-Cases. Beim Use-Case-Diagramm geht man von den Funktionen aus und stellt die Frage, welche Akteure (Aufgaben) davon betroffen sind.

Wie bereits erwähnt, scheint das Stellenfunktionendiagramm nur sehr selten eingesetzt zu werden. Zumindest in der Literatur konnte es vom Autor nicht gefunden werden. In den meisten Fällen werden lediglich Funktionslisten erstellt, welche nach den Organisationseinheiten geordnet sind.

Der Vorteil einer grafischen Darstellung gegenüber einer einfachen Liste kann darin liegen, dass Funktionen hervorgehoben werden, welche von mehreren Stellen *gemeinsam wahrgenommen* werden (s. Abb. 66). Dies führt zu einem besseren Verständnis für die Systemzusammenhänge.

In ähnlicher Weise könnte man unten anstelle von Funktionen auch Prozesse darstellen. Allerdings würde es sich dann um eine spezifische Darstellungsform des "Stellenorientierten Informationsflusses" handeln und nicht mehr um ein "Stellenfunktionendiagramm".

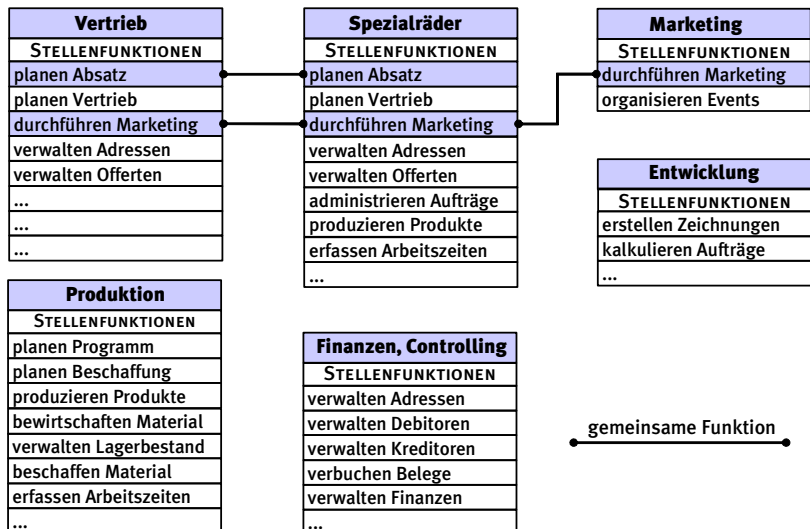


Abb. 66: Stellenfunktionendiagramm

### 6.7.1 Stellenfunktionen-Matrix

Die Beziehungen zwischen den Aufgaben/Stellen und den Funktionen müssen oftmals noch näher spezifiziert werden. Anstelle einer grafischen Darstellung kann selbstverständlich auch eine entsprechende Matrixdarstellung erstellt werden (s. Abb. 67).

Die vorliegende Form einer Matrix von zwei Aspekten kann für jeden der beschriebenen Diagrammtypen angewendet werden. Wir haben daher nicht bei jedem Diagramm auf diesen Umstand hingewiesen. Eine solche Matrix-Darstellung eignet sich insbesondere dann, wenn die Beziehungen zwischen den zwei Sichten, also z.B. zwischen den Stellen und den Funktionen in sehr detaillierten, und vollständiger Art und Weise aufgenommen bzw. katalogisiert werden müssen.

Anstelle der oben beschriebenen grafischen Darstellung werden meistens Zusatzinformationen hinsichtlich der Kompetenzen und Verantwortlichkeiten in der Tabelle angegeben (Spisak 1999, S. 32).

	Service & Spezialräder	Vertrieb	Marketing
planen Absatz	verantwortlich	verantwortlich	beratend
planen Vertrieb			
durchführen Marketing	entscheidend	entscheidend	verantwortlich
verwalten Adressen	durchführend	durchführend & verantwortlich	durchführend
verwalten Angebote			
administrieren Aufträge			
...			

Abb. 67: Matrix-Darstellung der Beziehung Stelle – Funktion

## 6.8 Arbeitsobjektdiagramm

*Definition: Arbeitsobjektdiagramme stellen primär Stellen dar sowie sekundär die von diesen Stellen bearbeiteten Objekte.*

In gewisser Weise ist das Arbeitsobjektdiagramm dem Datenflussdiagramm (DFD) ähnlich (s. o.). Das DFD ordnet die Objekte den Funktionen zu. Die einzelnen Kästchen zeigen nun aber Aufgaben bzw. Stellen (Rechtecke) – und dürfen gerade nicht mit den Funktionen verwechselt werden (Abb. 68).

Trotz Literaturrecherche und Expertenbefragung ließ sich dieses Diagramm wie erwähnt nicht finden. Es soll daher ganz einfach postuliert werden. Vergewenwärtigen wir uns nochmals die Fragestellung: “Gesucht sind die den Aufgaben zugeordneten Arbeitsobjekte”. Die Notation einer Karteikarte steht für die Objekte. Auch dieser Diagrammtyp ließe sich ganz entsprechend dem Datenflussdiagramm (DFD) hierarchisch gliedern.

Aus der Fülle der vorhandenen Arbeitsobjekte muss in der Praxis allenfalls der Ansatz verfolgt werden, auf einer Ebene primär jene Arbeitsobjekte zu verzeichnen, die von mehreren Aufgaben/Stellen verwendet werden.

Andererseits wird man sich in der Praxis anstelle eines Diagrammes vermutlich in vielen Fällen darauf beschränken, eine tabellarische Auflistung der Arbeitsobjekte zu erstellen. Eine solche lässt sich auch sortieren, z.B. nach der verantwortlichen Organisationseinheit, der Menge etc.

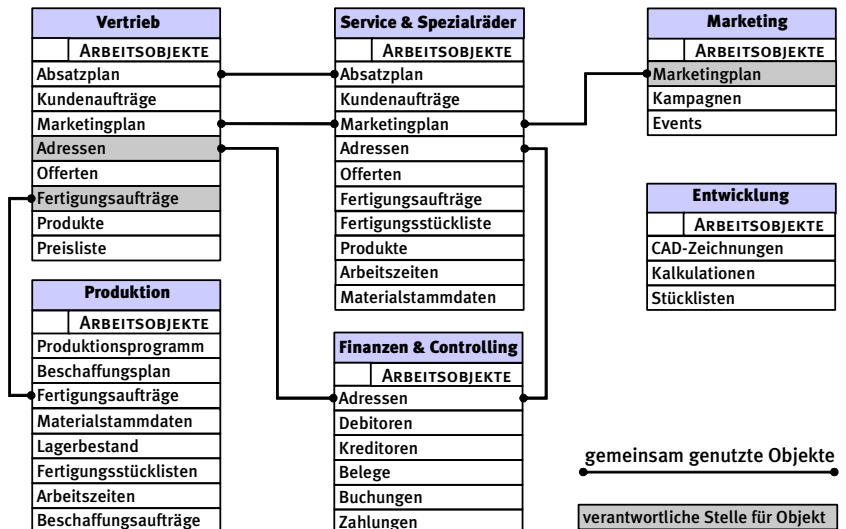


Abb. 68: Arbeitsobjektdiagramm

## 6.9 Integrierte aufgabenorientierte Methode: MTO-Analyse

In der Vergangenheit wurden nur wenige kombinierte sozio-technische Systemanalysen durchgeführt (Strohm et al. 1997). Betrachtet man die methodischen Lücken hinsichtlich der “Aufgabensicht” in den bereits behandelten kombinierten Methoden, dann erstaunt dies nicht.

Emery unterscheidet neun Schritte einer sozio-technischen Analyse (s. Abb. 69) (Emery 1967). Strohm und Ulich haben sodann, aufbauend auf Emery und angesichts des Bedarfes an kombinierten Analyseverfahren, ein Analyse- und Bewertungsinstrumentarium entwickelt (MTO-Analyse), welches mehrere aufgabenorientierte Methoden integriert. Dieses wird in der nachfolgenden Tabelle dem Ansatz von Emery gegenübergestellt (Strohm et al. 1997).

<b>Emery 1967</b>	<b>Strohm &amp; Ulich 1997</b>
Grobanalyse des Produktionssystems (1)	Analyse auf der Ebene des Unternehmens (1)
Beschreibung des Produktionsprozesses nach Input, Transformationen und Output (2)	Analyse von Auftragsdurchläufen (2)
Ermittlung der Hauptschwankungen im Produktionsprozess (3) Analyse des sozialen Systems einschliesslich der Bedürfnisse der Mitarbeiter (4)	Analyse von Arbeitssystemen inkl. Schwankungen und Störungen (3) Analyse von Arbeitsgruppen (4) Bedingungsbezogene Analyse von Schlüssel-tätigkeiten (5)
Analyse der Rollenwahrnehmung der Mitarbeiter (5)	Personenbezogene Arbeitsanalysen (6)
Einfluss des Instandhaltungssystems (6) Einfluss des Zuliefer- u. Abnehmersystems (7) Einfluss des Umweltsystems (8)	
	Analyse der soziotechnischen Geschichte (7)
Erarbeitung von Gestaltungsvorschlägen (9)	

Abb. 69: Gegenüberstellung aufgabenorientierter Ansätze<sup>2</sup>

<sup>2</sup> Für genauere Erläuterungen wird auf die zitierte Literatur verwiesen.

Hinzugekommen bei Strohm & Ulich sind im Wesentlichen die bedingungsbezogene Analyse der Schlüsseltätigkeiten (5) sowie eine Analyse der soziotechnischen Geschichte des Unternehmens (7). Im Übrigen basiert das MTO-Verfahren auf einer Anzahl neuerer Analyseinstrumente. Das MTO-Analyseverfahren ist somit als ausgewogenes und abgestimmtes "Set" einer Anzahl von geeigneten Methoden und damit als Leitfaden zur umfassenden aufgabenorientierten Analyse soziotechnischer Systeme zu verstehen.

In der unten stehenden Matrix ist wiederum vermerkt, welche Bereiche an Modellen die kombinierte Methode abdeckt (s. Abb. 70). Es ist allerdings zu bemerken, dass weder das Stellenfunktionen- noch das Arbeitsobjekt-diagramm in der zuvor beschriebenen Form angewendet werden.

Die Methode geht vom Organisationsmodell aus und untersucht diesem entsprechend in einem Top-down-Approach die Aufgaben der verschiedenen Ebenen (Unternehmen, Gruppe, Arbeitssystem, Tätigkeit).

Die MTO-Analyse integriert sodann eine Reihe von arbeitspsychologischen Methoden wie KABA (Dunckel et al. 1993) und VERA (Volpert et al. 1983), die die Aufgaben und Aufgabenbedingungen detailliert beleuchten.

					Hauptfokus ↓
primär sekundär	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht	
	Prozesssicht				Stellen-orientierter Informationsfluss
	Funktionssicht				Stellen-funktionen-diagramm
	Objektsicht				Arbeitsobjekte
	Aufgabensicht	Stellen-orientiertes Ablaufdiagramm			Organisationsmodell

Abb. 70: Analysebereiche und Diagramme der MTO- Analyse



## 6.10 Aspektorientierte Organisationsformen

Interessant ist, dass sich im Verlauf der vergangenen Jahre Organisationsformen entwickelt haben, welche sich an den dargestellten Sichten orientieren. Jede Organisationsform legt einen Fokus auf die eine oder andere Sichtweise. So entstanden prozess-, funktions-, objekt-, und aufgabenorientierte Organisationsformen:

- prozessorientierte Organisationsformen
- funktionale Organisationsformen
- objektorientierte Organisationsformen
- aufgabenorientierte Organisationsformen

Die funktionale Organisationsform entspricht der klassischen, funktional arbeitsteiligen Organisation. Die prozessorientierte Organisation entspricht einem neueren Ansatz, der versucht, die funktionalen Hürden zu überwinden und ganzheitliche Prozesse organisatorisch zusammenzufassen.

Der objektorientierte Ansatz entspricht am ehesten einer divisionalen, produktbezogenen Struktur, wie sie meist in größeren Unternehmen und Konzernen anzutreffen ist. Allerdings wird dann auf den tieferen Ebenen meist wieder auf die funktionale Organisation gewechselt.

Die aufgabenorientierte Organisation ist bestrebt, die Organisation nicht einfach an einer anderen Sicht (Prozess, Funktion, Objekt) auszurichten. Vielmehr stellt sie das Primat der Motivation und der Gesamtoptimierung in den Vordergrund. Sie geht davon aus, dass keine Patentrezepte bestehen und die Organisation eine eigene Sichtweise darstellt.

Man könnte sich an dieser Stelle fragen, weshalb sich die Organisationsformen nach jeweils einem Aspekt des Systems ausgerichtet haben. Eine Begründung mag darin liegen, dass die Ausrichtung der Organisation anhand der Prozesse, Funktionen oder Objekte zu einer gewissen Komplexitätsreduktion führt. Einzig die "eigenständige" Organisationsgestaltung im Sinne der Optimierung der Aufgaben führt gewissermaßen zu einer weiteren "Dimension" des Systems.

Im Rahmen von Informatikprojekten sind die organisatorischen Spielräume allerdings häufig leider eher klein bzw. sie beschränken sich auf organisatorische Feinanpassungen. Häufig werden bestehende Organisationsstrukturen als etwas Unveränderliches betrachtet und es braucht einen Anstoß von externen Mitarbeitern, damit neue Organisationsformen umgesetzt werden.

### 6.10.1 Funktionsorientierte Organisationsformen

Mit dem “Taylorismus” (Taylor 1913) verbreitete sich zu Beginn dieses Jahrhunderts die so genannte funktionale Organisationsform in den meisten Unternehmen. Diese zeichnet sich primär durch funktionale Arbeitsteilung aus: Jeder Mitarbeiter erhält eine sehr eingeschränkte Funktion zugewiesen und soll diese nach kurzer Zeit effizient beherrschen. Damit wird eine hohe Produktivität bei minimalen Qualifikationsanforderungen und gesamthaft niedrigen Lohnkosten angestrebt.

Viele Unternehmen sind noch heute nach diesem funktionalen, tayloristischen Prinzip organisiert. Das kann sich darin ausdrücken, dass eine Firma über organisatorisch klar getrennte Geschäftsbereiche für Vertrieb, Entwicklung, Fertigung, etc. verfügt – wie auf der Grafik dargestellt (s. Abb. 71). Gelegentlich spricht man in diesem Zusammenhang auch von “funktionalen Silos”. Häufig ist auch die Fertigung stark funktional organisiert: z.B. mit einer Rüsterei, Dreherei, Fräselei, Montage, etc. Es ist nun eine eklatante Schwäche dieser Organisationsform, dass *kundenorientierte* Aufträge diese “Silos” quer durchlaufen.

Eine kundenspezifische Produktion erfordert aber eine enge Koordination und Kooperation aller am Auftrag beteiligten Funktionen. Die organisatorischen Barrieren der einzelnen “Silos” behindern nun genau die Durchgängigkeit und somit die flexible Abwicklung des Kundenauftrages. Gelegentlich müssen operative Probleme auf höchster Ebene gelöst werden.

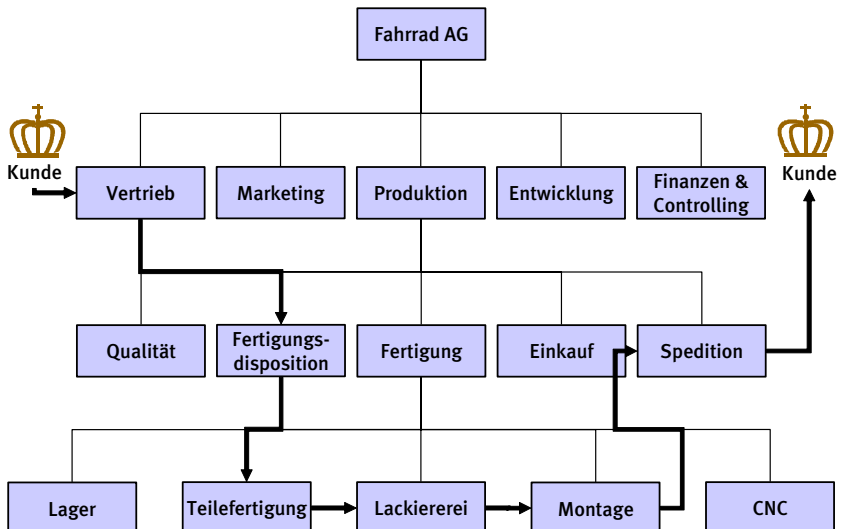


Abb. 71: Geschäftsprozesse durchqueren funktionale Organisationen

### 6.10.2 Prozessorientierte Organisationsformen

In den Unternehmen hat sich zunehmend die Einsicht durchgesetzt, dass die Zeit für organisatorische Veränderungen reif ist.

Als Lösungsansatz bietet es sich nun an, Aufträge über Organisationsstrukturen abzuwickeln, die mehrere Funktionen eines Geschäftsprozesses integrierter und damit "vollständiger" behandeln. Der Begriff "vollständig" drückt aus, dass sich die Betriebsorganisation an den umfassenden Geschäftsprozessen und nicht an den Funktionen orientiert.

In diesem Sinne spricht man auch von einem "90 degree shift" der Organisation. Von eher funktional "vertikalen" Strukturen geht man zu Strukturen über, die sich am horizontal dargestellten Geschäftsprozess orientieren. Aber auch hier ist Vorsicht angebracht: Eine unbesehene Übernahme und Abbildung von Geschäftsprozessen auf die Organisationsstrukturen kann genauso wie die funktionale Denkweise zu unzumutbaren Lösungen führen.

Eine horizontale Einheit integriert nun mehrere Funktionen, die früher organisatorisch getrennt waren. Sie verfügt beispielsweise über eigene Vertriebs-, Fertigungs- und Finanzkompetenzen. Auch die Entwicklung rückt viel näher zum Vertrieb und zur Fertigung – die Kommunikationswege haben sich verkürzt. Dies hat zur Folge, dass spezifische Kundenanforderungen rasch und flexibel abgearbeitet werden können; mithin verflachen sich auch die Organisationsstrukturen.

In gewisser Weise entspricht diese eher prozessorientierte Organisation einem Rückgriff auf traditionelle handwerkliche Organisationsformen. Eine prozessorientierte Einheit ist wieder verantwortlich für den Verkauf, die Fertigung und den Einkauf. Bei handwerklichen Betrieben, wie bei einem Bäcker oder einem Schreiner, war dies schon immer der Fall. Eine ideale Form der Reorganisation des Unternehmens besteht häufig in der Bildung von "Segmenten". Dies wurde bereits weiter oben dargelegt.

Es sei an dieser Stelle aber zusätzlich angemerkt, dass die Segmentierung nicht nach der Komplexität erfolgen sollte. Ansonsten erzeugt man Unternehmensbereiche, die qualifizierende Aufgaben haben, wogegen sich andere Bereiche lediglich mit Routinearbeiten befassen müssen.

### 6.10.3 Objektorientierte Organisationsformen

Die objektorientierte Organisationsform nimmt die Objekte des Unternehmens als Strukturierungsansatz. Hier könnten insbesondere unterschiedliche Produkte zum Gegenstand der Strukturierung genommen werden, wie es beispielsweise bei divisionalen Unternehmen der Fall ist. Auch Vertriebsorganisationen sind häufig nach diesem Ansatz organisiert.

#### 6.10.4 Aufgabenorientierte Organisationsformen

Unter einer aufgabenorientierten Organisationsform versteht man Organisationsformen, welche nach dem Primat arbeitsorganisatorischer Grundsätze strukturiert sind. Negativ formuliert bedeutet dies, dass diese nicht einfach unbesehen Funktions-, Prozess- oder Objektstrukturen übernehmen und diese auf die Organisation abbilden. Vielmehr sollten verschiedenartige Gestaltungsansätze geprüft werden. Die Arbeitspsychologie schlägt vor, folgende Gestaltungsmerkmale für Aufgaben zu prüfen (Ulich 2001):

<i>Organisationsebene</i>	<i>Strukturprinzip</i>
Unternehmen	Dezentralisierung
Organisationseinheit	Funktionale Integration
Gruppe	Selbstregulation
Individuum	Qualifizierte Tätigkeit

Die Bedeutung von motivierten und qualifizierten Mitarbeitern kann nicht genug hervorgehoben werden. Auch die Betriebswirtschaft erkennt mit ihren neueren Ansätzen – beispielsweise dem Balanced Scorecard (Kaplan 1997) – die Entwicklung der Mitarbeiter und deren Fachwissen endlich als Hauptvoraussetzung für den langfristigen Erfolg eines Unternehmens an.

Es wird postuliert, dass das Lernen als Grundlage für die Geschäftsprozesse steht, diese sich wiederum auf die Kunden auswirken und erst am Ende die finanziellen Auswirkungen im Unternehmen zu erkennen sind.

Wie sehr die Qualifikation der Mitarbeiter die Qualität der Prozesse beeinflusst, wird deutlich, wenn man die Korrelation zwischen der Beurteilung der Aufgabensicht (s. o.) mit der Evaluation von Prozesskriterien (Termin-treue, Durchlaufzeit, Flexibilität, etc.) betrachtet. Es wurde klar: je höher die Aufgabenorientierung, desto bessere Prozessparameter erreicht das System (Ulich 2001).

Wenn also motivierte und lernende Mitarbeiter sowohl aus Sicht der Betriebswirtschaft als auch der Arbeitspsychologie ein Hauptmerkmal für den Erfolg eines Unternehmens in der Zukunft bilden, dann ist die Frage berechtigt, wodurch diese Motivation erreicht und bewirkt werden kann.

Auf das Prinzip der Dezentralisierung auf Unternehmensebene wird an dieser Stelle nicht näher eingegangen, da dieser Bereich eingehender im Bereich der Geschäftsprozessgestaltung und -segmentierung behandelt wird (s. u.).

## 6.11 Literatur zur aufgabenorientierten Modellierung

- Atkins, P. (1926). *Factory Management*. New York: Prentice-Hall.
- Dunckel, H., Volpert, W., Zölch, M. et al. (1993): Kontrastive Aufgabenanalyse im Büro. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 7. Zürich/Stuttgart: Verlag der Fachvereine/Teubner.
- Dunckel, H. (Hrsg.) (1999): Handbuch psychologischer Arbeitsanalyseverfahren. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 14. Zürich: vdf Hochschulverlag.
- Emery, F.E. (1967): Analytical Model for Socio-Technical Systems. In: *The Emergence of a New Paradigm of Work* (Emery, F.E.; Hrsg.); Canberra: Australian National University. 95–106.
- Frei, F. (1981): Psychologische Arbeitsanalyse – eine Einführung zum Thema. In: *Beiträge zur psychologischen Tätigkeitsanalyse*; (Frei, F., Ulich, E.; Hrsg.); Schriften zur Arbeitspsychologie, Band 31. Bern: Huber. 11–36.
- Grote, G. (1991): Effekte der Nutzung eines Bürokommunikationssystems auf Arbeitsprozesse und -strukturen. In: *German Chapter of the ACM, Band 34: Computergestützte Gruppenarbeit (CSCW)*. (Friedrich, J., Rödiger, K.H.; Hrsg.); Stuttgart: Teubner. 221–249.
- Hacker, W. (1978): Allgemeine Arbeits- und Ingenieurpsychologie: Psychische Struktur und Regulation von Arbeitstätigkeiten; Schriften zur Arbeitspsychologie, Band 20. Bern: Hans Huber.
- Matern, B. (1983): Psychologische Arbeitsanalyse. Spezielle Arbeits- und Ingenieurpsychologie (Hacker, W.; Hrsg.), Lehrtext 3. Berlin: Deutscher Verlag der Wissenschaften.
- Oesterreich, R. (1984): Zur Analyse von Planungs- und Denkprozessen in der Produktion. Das Arbeitsanalyseinstrument VERA. In: *Diagnostica* 30. 216–234.
- Roos, E. (1992). Informationsmodellierung für PPS-Systeme: Ein Konzept zur aufgabenorientierten Systementwicklung. Berlin: Springer.
- Spisak, M. (1999). Organisation als Führungsaufgabe. In: *Handbuch angewandte Psychologie für Führungskräfte* (Steiger, Th.; Lippmann, E.; Hrsg.). Führungskompetenz und Führungswissen, Band 2. Berlin: Springer.

- Strohm, O., Ulich, E. (Hrsg.) (1997): Unternehmen arbeitspsychologisch bewerten: ein Mehr-Ebenen-Ansatz unter besonderer Berücksichtigung von Mensch, Technik und Organisation (Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 10. Zürich: vdf Hochschulverlag.
- Taylor, F.W. (1913): Die Grundsätze wissenschaftlicher Betriebsführung. Berlin: Oldenburg.
- Trist, E.L. & Bamforth, K.W. (1951): Some social and psychological consequences of the longwall method of coal getting. In: Human Relations 4. 3–38.
- Ulich, E. (2001): Arbeitspsychologie, 5. Aufl. Zürich/Stuttgart: vdf Hochschulverlag/Schäffer-Poeschel.
- Volpert W., Oesterreich, R. et al. (1983): Verfahren zur Ermittlung von Regulationserfordernissen in der Arbeitstätigkeit (VERA). Analyse von Planungs- und Denkprozessen in der industriellen Produktion. Köln: Verlag TÜV Rheinland.
- Weber, W.G. (1994): Psychologische Analyse und Bewertung computergestützter Facharbeit. Berlin: Quintessenz.
- Weber, W.G. (1996): Analyse von Gruppenarbeit in der Produktion: Versuch einer Integration von Konzepten zur kollektiven Handlungsregulation in soziotechnischen Systemen. Habilitationsschrift. Zürich: Eidgenössische Technische Hochschule.







# 7

## Modellierung der Technik

### **Kapitelinhalt**

- Techniksysteme als Elemente zur Unterstützung & Automation
- Systemarchitekturmodelle, Hardware- und Netzwerkmodelle
- Systemabgrenzung und Spezifikation von Techniksysteme
- Modellierungsmethoden und Diagramme der Techniksicht

Das nachfolgende Kapitel behandelt die Modellierung der technikbezogenen Aspekte eines Informationssystems. Zunächst wird ausgeführt, was unter dem Begriff eines Techniksystems zu verstehen ist und wie sich die Modellierung der Technik von den anderen Aspekten abgrenzt.

Es wird dargelegt, dass die technikorientierte Sichtweise darin besteht, das Techniksystem in den Mittelpunkt zu stellen. Entsprechend den anderen Sichten können auch Techniksysteme in eine hierarchische Form gebracht werden. Dies geschieht in Form der Systemarchitektur.

Auch dieses Kapitel folgt der Struktur der vorangegangenen Kapitel: Zunächst wird das Systemarchitekturmodell beschrieben und dann auf die Spezifikation von Techniksystemen eingegangen.

Wichtig ist hier der Übergang von einer allgemeinen Systemsichtweise zur technikbezogenen Sicht. Daraus ergibt sich eine Erweiterung der Modellierungsmatrix auf insgesamt 25 Diagramme.

## 7.1 Technikorientierte Modellierung

### 7.1.1 Techniksicht

Gewisse Operationen, Prozesse, Funktionen und Aufgaben eines sozio-technischen Systems werden technisch mit einem Informationssystem unterstützt. In Erweiterung des Matrixmodells muss daher eine Techniksicht hinzukommen (s. Abb. 72).

Auch diese Sicht auf das System ist wiederum ein eigenständiger Gesichtspunkt und gruppiert die Gesamtmenge der Operationen nach dem sie unterstützenden Informationssystem. Es entsteht dadurch ein weiteres Systemmodell – das Systemarchitekturmodell.

Wie wir weiter unten sehen werden, darf dieses anwendungsbezogene Systemarchitekturmodell nicht mit einer reinen technischen Systemarchitektur verwechselt werden.

*Definition: Die Techniksicht betrachtet ein sozio-technisches System unter dem Gesichtspunkt der die Operationen unterstützenden Techniksyste.*

*Definition: Techniksyste bezeichnen zusammengehörige und in sich relativ geschlossene technische Einheiten zur Unterstützung oder Automation von Operationen der unterschiedlichen Niveaus.*

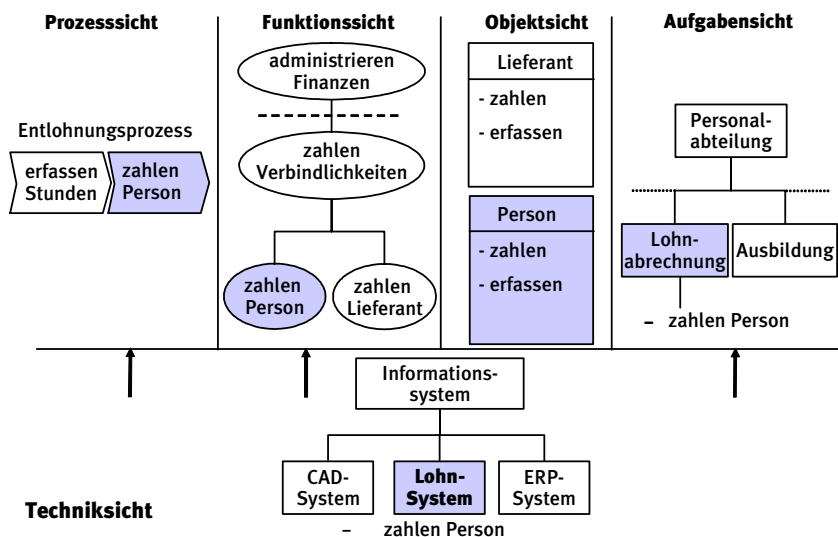


Abb. 72: Techniksicht

## 7.2

**Technikorientierte Methoden im Überblick**

Mit der Techniksicht kommt nun eine neue Dimension zu unserem Matrixmodell dazu. Ausgehend vom Technikmodell, welches die Beziehungen der einzelnen Technikkomponenten zueinander aufzeigt, müssen Beziehungen zu anderen Aspekten aufgezeigt und untersucht werden. Damit erhält auch die bestehende Matrix gemäß der unten stehenden Darstellung eine weitere Dimension (s. Abb. 73). Wiederum lassen sich eine primäre und eine sekundäre Sicht unterscheiden.

Allerdings bestehen im Bereich der Technikmodellierung nicht gleichermaßen integrierte Modellierungsansätze wie dies bei der Prozess-, Funktions-, Objekt- und Aufgabenmodellierung der Fall ist. Trotzdem finden sich viele der nachfolgenden Diagramme in der einen oder anderen Form in bestehenden Ansätzen wieder.

Dies betrifft insbesondere das Systemschnittstellen-Diagramm und das Prozesstechnologie-Diagramm. Wie wir nachfolgend ausführen werden, gibt es aber auch für weitere Diagrammtypen dieser Morphologie ebenfalls gut begründbare Anwendungsfälle. Viele Bezeichnungen mussten mangels vorhandener Begriffe selbst gewählt werden.

<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht	Techniksicht
Prozesssicht					System-schnittstellen-diagramm
Funktionssicht					System-funktionen-diagramm
Objektsicht					Datenobjekt-diagramm
Aufgabensicht					System-nutzungs-diagramm
Techniksicht	Prozess-technologie-diagramm	Funktionsunterstützungs-diagramm	Objekt-zugriffs-diagramm	Technik-einsatz-diagramm	<b>System-architektur-modell</b>

Abb. 73: Primäre und sekundäre Modellierung der Technik

### 7.3 Systemarchitekturmodell

*Definition: Systemarchitekturmodelle stellen die Strukturen und die Beziehungen zwischen den verschiedenen Techniksyste men dar.*

Bei der Techniksicht steht das Systemarchitekturmodell am Anfang aller Betrachtungen (s. Abb. 74). In der unten stehenden Abbildung ist beispielhaft die Hierarchie des Informationssystems eines Industriebetriebes dargestellt. Häufig wird dieses Modell auch als “Anwendungsarchitektur” oder “Applikationsarchitektur” bezeichnet.

Das Systemarchitekturmodell zeigt auf der Anwendungssystemebene die Hauptanwendungen (engl.: “Solutions”). Im gezeigten Beispiel gibt es für das CAD-System zwei grundlegend unterschiedliche Installationen, obwohl es sich um das gleiche Grundprodukt handelt. Sie müssen aber wie eigenständige Anwendungen konzipiert und behandelt werden.

Auf der Modul- und Subsystemebene finden sich im Beispiel einerseits die Module des ERP-Systems. Das CRM-Modul CRM ist als Subsystem für zwei verschiedene Unternehmensbereiche eigenständig implementiert (falls die Softwarelösung dies erlaubt).

Wichtig ist also nicht nur die Überlegung, welche Softwareanwendungen zum Einsatz gelangen, sondern welche autonomen Installationsbereiche und Konfigurationen implementiert sind oder konzeptionell angedacht werden. Diese Autonomie kann auf verschiedenen Ebenen bestehen.

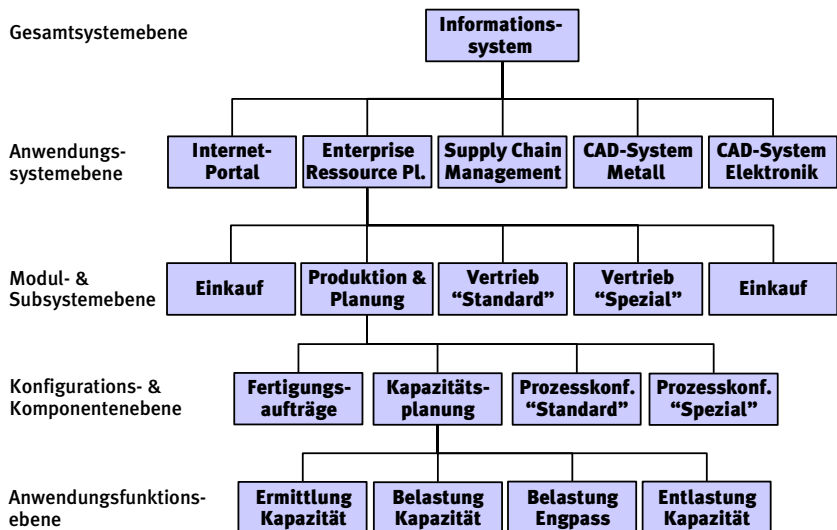


Abb. 74: Systemarchitekturmodell im Sinne der Anwendungsarchitektur

### 7.3.1 Technische Systemarchitektur

Ein Informationssystem verfügt neben einer Anwendungsarchitektur über eine “Technische Systemarchitektur” im Sinne von Systemplattformen (s. Abb. 75). Die unten stehende Darstellung zeigt eine Mehrebenen-Architektur (engl.: “Multi-tier”), wie sie in vernetzten Umgebungen Verbreitung erlangt hat. Wir möchten an dieser Stelle nicht auf Details eingehen, da dies zu technisch wäre und den Rahmen sprengen würde. Es genügt uns der Hinweis auf die wichtigsten Architekturtypen: *Host-, Client/Server-* und *Multi-Tier-Architekturen*. Diese Architekturtypen unterscheiden sich in Bezug auf die Zuteilung der Programmlogik auf die Systemebenen. Die Host-Architektur überträgt die Programmlogik ausschließlich dem Host; dem Terminal bleibt nur noch die Darstellung der Ergebnisse; die Client-Server-Architektur verteilt die Anwendungslogik auf den Server und den Client. Die Multi-Tier-Architektur verteilt die Logik auf mehr als zwei Ebenen, was aber auch zu höherer Komplexität führen kann.

Mit der Einführung des Aspektes der technischen Architektur ergibt sich im Prinzip auch eine neue Dimension in unserem Modellierungsansatz und man könnte die Modellierungsmatrix um diesen Aspekt ergänzen. Der praktische Nutzen einer solchen Erweiterung stellt sich aber als zu gering dar, da nur wenige ergänzende Modelle von praktischer Hilfe wären.

Dies mindert allerdings nicht die Bedeutung der technischen Systemarchitektur bei deren Gestaltung im Rahmen des Designs und Implementation.

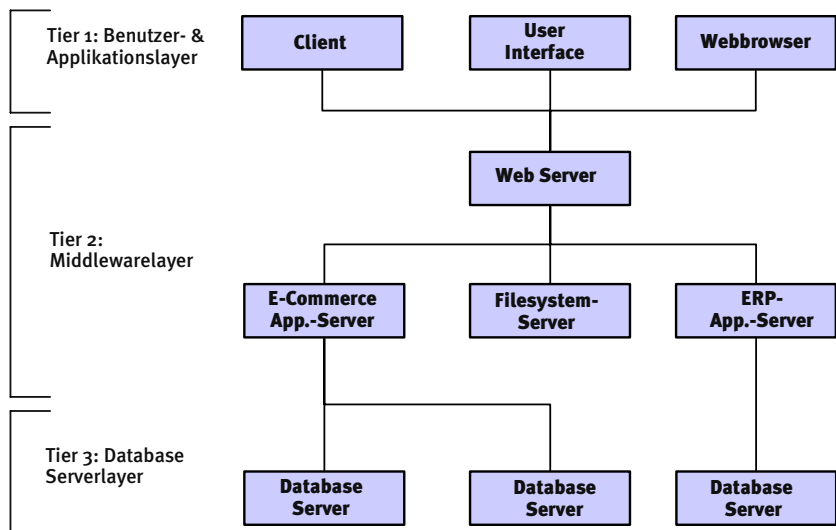


Abb. 75: Modell der technischen Systemarchitektur

### 7.3.2 Physisches Hardware- und Netzwerkmodell

Das physische Hardware- und Netzwerkmodell stellt ein weiteres Technikmodell dar (s. Abb. 76). Es kann nämlich durchaus sein, dass auf gewissen Hardwarekomponenten (z.B. Arbeitsstationen oder Serversystemen) verschiedene Applikationsserver, Webserver oder Informationssysteme zum Einsatz kommen.

Es würde an dieser Stelle aber wie bei der oben stehenden technischen Systemarchitektur den Rahmen sprengen bzw. das Ziel verfehlen, diese zusätzliche Sicht wiederum vollständig mit all ihren Abhängigkeiten und Beziehungen zu behandeln. Es ist aber natürlich ein möglicher Ansatz, bei einer gegebenen Problemstellung im technischen Umfeld des IT-Betriebs, auch solche weiterführenden Darstellungen zu entwickeln. Sie würden die Beziehungen zwischen diesen technischen Architekturen und den Anwendungssystemen aufzeigen.

Wir werden uns an dieser Stelle aber auf die beispielhafte Modellierung der Hardware- und Netzwerkarchitektur beschränken. Diese Darstellung wird im Rahmen von Systemspezifikationen und Pflichtenheften recht häufig zum Einsatz kommen.

Die unten stehende Abbildung ist nur eine mögliche Art der Visualisierung, welche sich an der Kommunikationsarchitektur orientiert. Gleichermaßen könnte selbstverständlich ein formales hierarchisches Modell im Sinne einer "Baumstruktur" erstellt werden.

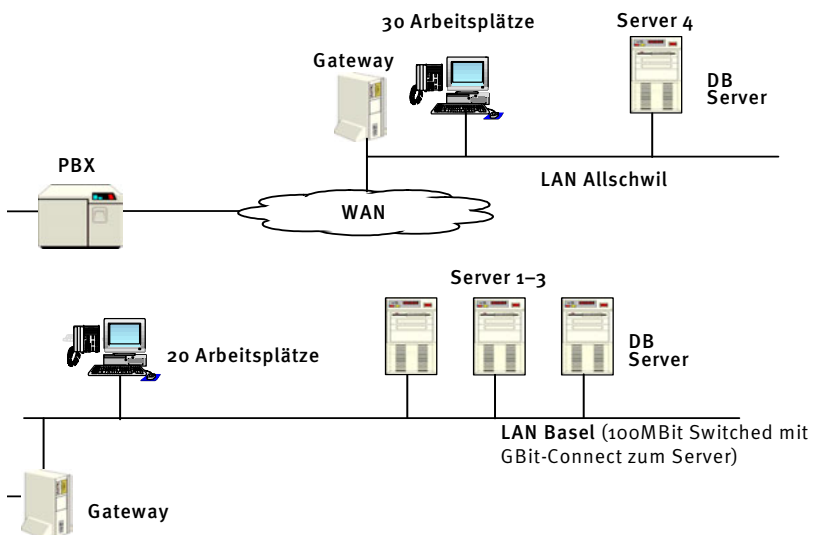


Abb. 76: Physisches Hardware- und Netzwerkmodell

## 7.4 Identifikation und Systemabgrenzung der Informationstechnik

Wie bei den anderen Systemaspekten muss nun auch im Bereich der Technik zunächst die Frage beantwortet werden, wie sich Techniksyste-  
me (Anwendungssysteme, Subsysteme etc.) identifizieren lassen.

Die Identifikation der Techniksyste-  
me kann in der Regel auf relativ einfache Art und Weise Top-down erfolgen. Zumindest bei einer Ist-Analyse ist ja bekannt, welche Anwendungssysteme im Einsatz sind. Auch hier bieten natürlich gewisse weiterführende Möglichkeiten, ausgehend von bereits erstellten Modellen, Rückschlüsse auf die vorhandenen Techniksyste-  
me und Subsysteme zu ziehen.

Die Systemabgrenzung der Informationstechnik ist eine rein technische Abgrenzung. So könnte beispielsweise in einem Projekt definiert werden, dass weder CAD- noch Betriebsdatensysteme betrachtet werden (s. Abb. 77).

Es wird ersichtlich, dass die Systemabgrenzung neben Prozessen, Funktionen, Objekten und Aufgaben unter Umständen für eine weitere Dimension spezifiziert werden muss. Fallweise kann es zwar gelingen, diese Sichten in geeigneter Weise zu einer allgemeinen Systemabgrenzung zu kombinieren. Dies aber nur dann, wenn die verschiedenen Aspekte in gewissem Sinne eng gekoppelt sind – also beispielsweise das CAD-System nur in einem definierten Prozess bzw. in einer Abteilung (Aufgabe) eingesetzt wird.

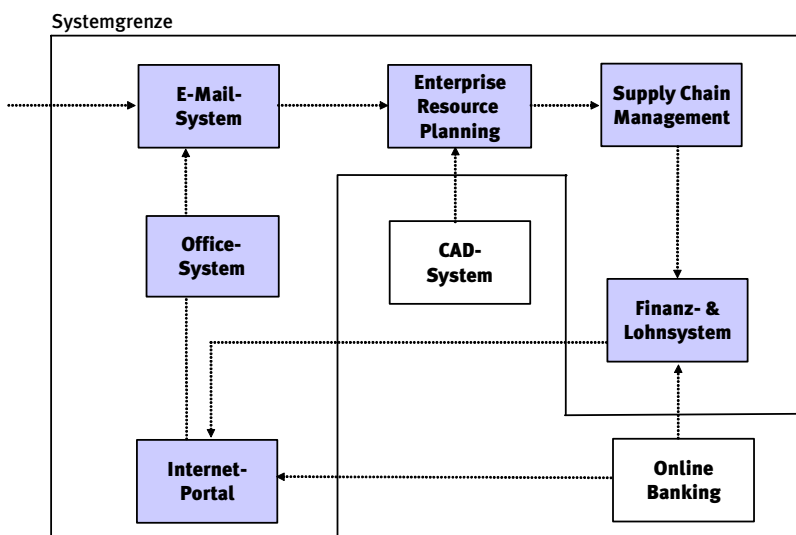


Abb. 77: Beispielhafte Systemabgrenzung der Informationstechnik

## 7.5 Spezifikation eines Techniksystems

Die Spezifikation eines technischen Systems erfolgt ebenfalls nach dem bereits mehrfach aufgezeigten Muster.

### **Abgrenzung und Beziehungen zu Umsystemen**

Es muss definiert werden, welche Beziehungen ein gegebenes System mit den übrigen Techniksyste men hat. Diese Informationen ergeben sich aus der Systemabgrenzung.

### **Inhalte eines Techniksyste ms und Subsysteme**

Sodann ist die "innere Struktur" eines Techniksyste ms zu beschreiben. Die in einem Techniksyste m enthaltenen bzw. aggregierten Subsysteme lassen sich aus dem Systemarchitekturmodell ableiten.

Von besonderer Bedeutung ist selbstverständlich auch hier, welche Funktionen und welche Objekte ein bestimmtes technisches System unterstützt beziehungsweise abdeckt.

### **Mengen, Häufigkeiten und Regeln**

Im Hinblick auf die Spezifikation eines technischen Systems sind oftmals Angaben von Mengen und Häufigkeiten ganz entscheidend.

### **Beziehungen zu anderen Aspektsyste men**

Außerdem wird zu eruieren sein, in welchen Beziehungen das System zu den übrigen Aspekten steht: Prozesse, Aufgaben, etc.

Nimmt man alle diese Punkte zusammen, dann ergibt sich wiederum eine Systemspezifikation im bereits bekannten Sinn. Denn jedes Techniksyste m wird selbst wieder (wie bereits das sozio-technische System) über mehrere Dimensionen verfügen. Es verfügt allerdings nicht über eine Aufgabensicht, da es ja keine personenbezogenen Stellen bzw. Aufgaben im eigentlichen Sinne beinhaltet.



## 7.6 Systemschnittstellendiagramm

*Definition: Systemschnittstellendiagramme stellen primär Techniksysteme dar sowie sekundär die diese Techniksysteme tangierenden Prozesse in Form von Schnittstellen.*

Die Schnittstellen zwischen Subsystemen können stets einem Prozess zugeordnet sein, weshalb sich dieses Diagramm im Bereich der Technik- und Prozesssicht positioniert (s. Abb. 78).

Die Analyse von Systemschnittstellen verdient in jedem Informatikprojekt viel Aufmerksamkeit. Schnittstellen sind eine häufige Ursache von Problemen in Projekten. Sie erfordern daher eine exakte Koordination und erhöhte Aufmerksamkeit des Projektleiters. Insbesondere sollte eine optimale Kommunikation zwischen unterschiedlichen Systemlieferanten sichergestellt werden, um nachträgliche Schuldzuweisungen zu vermeiden.

Das vorliegende Diagramm eignet sich gut, um das Verständnis der Systemzusammenhänge und Prozessabläufe eines komplexen Systemumfeldes zu visualisieren. Ergänzend werden daher gelegentlich die Informationsflüsse mit Nummern versehen, welche die Abfolge der zum Zuge kommenden Schnittstellen visualisiert.

Eventuell gemeinsam genutzte Datenbestände können diese Darstellung um eine weitere Sicht ergänzen.

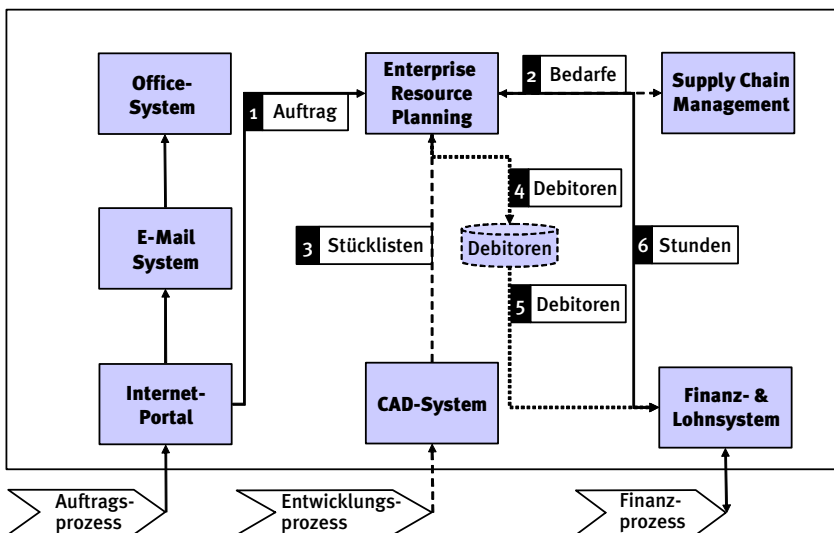


Abb. 78: Systemschnittstellendiagramm

## 7.7 Systemfunktionendiagramm

*Definition: Systemfunktionendiagramme stellen primär Techniksysteme (Subsysteme) dar und sekundär die durch diese Subsysteme abgedeckten Funktionen.*

Bei der Gestaltung von Informationssystemen wird es immer darum gehen zu entscheiden, welche Funktionen von welchem Informationssystem unterstützt oder abgedeckt werden sollen. Im Gegensatz zum Stellenfunktionendiagramm zeigt dieses Diagramm also nur jene Funktionen, welche auch von Technik unterstützt werden.

Es wird in der unten stehenden Darstellung ersichtlich, dass die Ausführung der Funktion "verwalten Adressen" die Koordination von mindestens vier unterschiedlichen Systemen bzw. Subsystemen erfordert (s. Abb. 79). Dies könnte Rückschlüsse auf die Komplexität einer gegebenen Systemarchitektur zulassen.

Das unten stehende Diagramm könnte auch auf andere Weise grafisch visualisiert werden (z.B. mit Funktionsbubbles). Hier wurde eine methodisch allenfalls weniger saubere, dafür praxisnähere Methode vorgeschlagen. Für das Systemfunktionendiagramm konnte in der Literatur kein verbindliches Beispiel gefunden werden. Dies deutet darauf hin, dass in der Praxis für ein solches Diagramm wenig Notwendigkeit besteht. Allerdings liegt eher die Vermutung nahe, dass in den meisten Fällen aufgrund der hohen Funktionszahl die Form einer einfachen Liste in Textform gewählt wird.

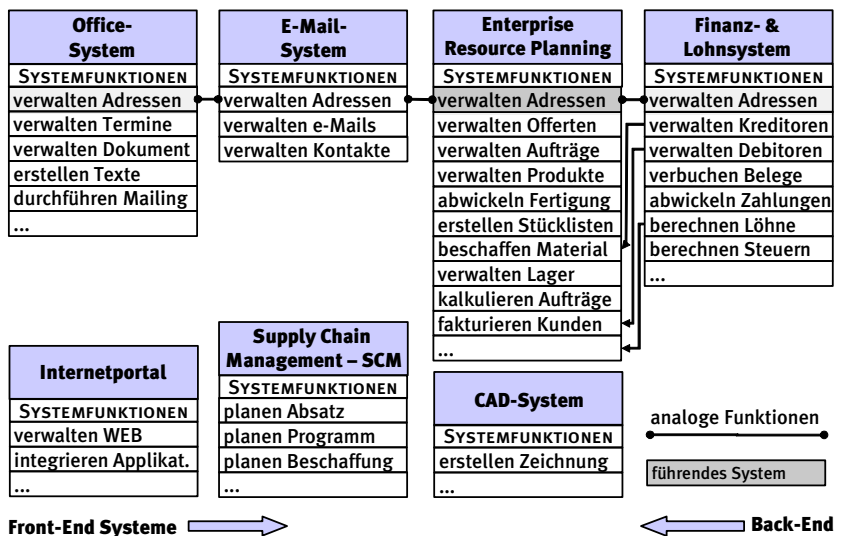


Abb. 79: Systemfunktionendiagramm

## 7.8 Datenobjektdiagramm

*Definition: Datenobjektdiagramme stellen primär Techniksysteeme dar so wie sekundär die von diesen Techniksysteemen genutzten Objekte.*

In einem heterogenen Informatikumfeld ist es von ganz besonderer Bedeutung zu wissen, welche Systeme die "Datenhoheit" bzw. den Zugriff auf die vorhandenen Datenobjekte besitzen.

Das Datenobjektdiagramm stellt damit dar, über welche Datenbestände eine gegebene Systemarchitektur verfügt bzw. verfügen wird (s. Abb. 80). Zusätzlich kann vermerkt werden, welches die gemeinsam verwalteten Datenobjekte eines jeden Subsystems sind. In der unten stehenden Abbildung wird beispielsweise ersichtlich, dass die kritischen Datenobjekte die Kundenadressen sind. Diese werden von mehreren Subsystemen mitverwendet.

Diese Darstellung kann nützlich sein, wenn es darum geht, die relevanten Datenobjekte im Zuge einer Datenübernahme zu identifizieren. Dies können beispielsweise Adressen, Produktionsaufträge, Material, lagerhaltige Teile, etc. sein. Unter Umständen wird man auf diese Weise auch erfahren, in welchen Systemen redundante Datenobjekte vorhanden sind und ob diese weiter redundant gepflegt werden sollen.

Dieses Diagramm könnte auch in der Notation des unten stehenden Objektzugriffdiagramms dargestellt werden.

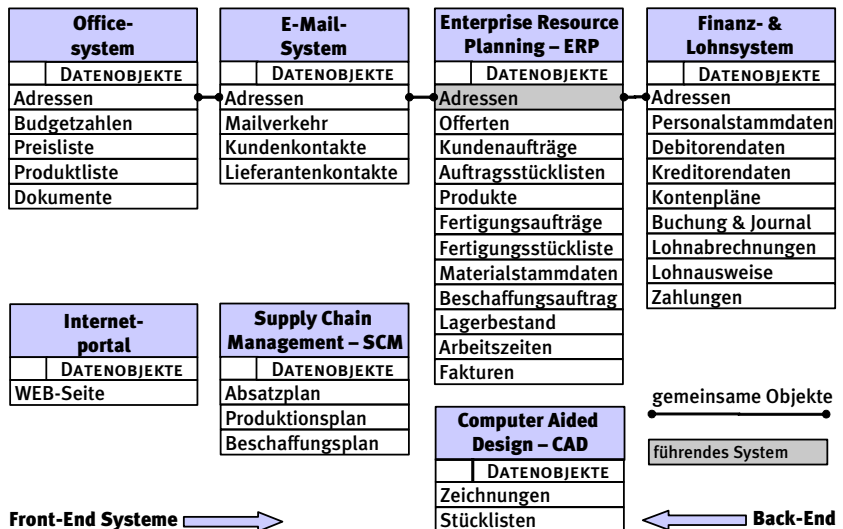


Abb. 80: Datenobjektdiagramm zur Visualisierung der Datenbestände

## 7.9 Systemnutzungsdiagramm

*Definition: Systemnutzungsdiagramme stellen primär Techniksyste-  
me dar sowie sekundär die von diesen Techniksyste-  
men unterstützten Systembe-  
nutzer (Stellen).*

Das Systemnutzungsdiagramm zeigt ausgehend von einer gegebenen Systemarchitektur auf, welche Systeme und Subsysteme von welchen Organisationseinheiten bzw. von welchen Aufgaben genutzt werden (s. Abb. 81). Primär ist hier die Sicht auf die Technik, im zweiten Schritt werden die zugehörigen Aufgaben dazu vermerkt.

In der unten stehenden Abbildung wurden die relevanten Subsysteme der Systemarchitektur aufgezeichnet. Anschließend wurde untersucht, welche Inseln bzw. Organisationseinheiten des Unternehmens auf die jeweiligen Systeme zugreifen.

Dieses Diagramm eignet sich gut, um sich eine anwendungsbezogene Gesamtübersicht des Informatikeinsatzes in einem Unternehmen zu verschaffen. Es kann einerseits der Analyse dienen oder auch im Rahmen einer Neukonzeption beispielsweise im Grobkonzept eingesetzt werden.

Das Diagramm muss selbstverständlich nicht zwingend das gesamte Informationssystem darstellen. Fallweise wird man es auch verwenden können, um z.B. die Nutzung der ERP Module (welche hier nicht dargestellt sind) auf die Nutzung durch die Unternehmensbereiche darzustellen.

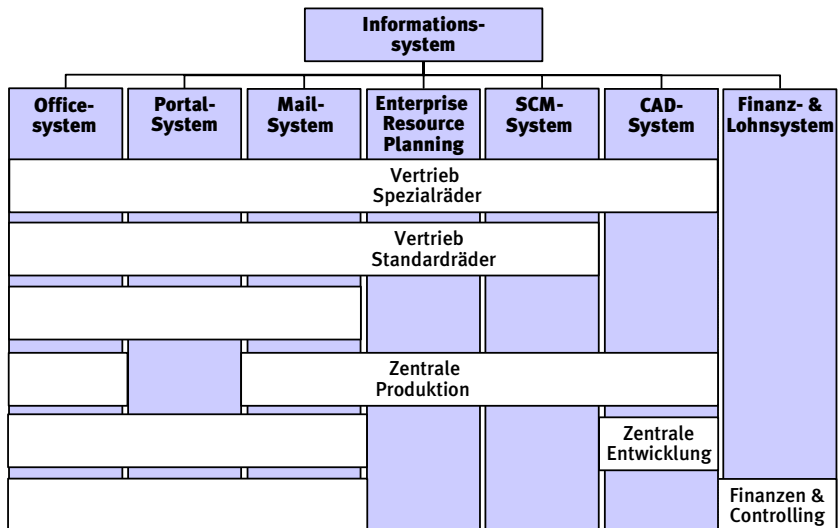


Abb. 81: Systemnutzungsdiagramm

## 7.10 Prozesstechnologiediagramm

*Definition: Prozesstechnologiediagramme stellen primär Prozesse dar sowie sekundär die diese Prozesse unterstützenden Techniksyste.*

Das Prozessunterstützungsdiagramm bietet sich als geeignete Analysehilfe zur Darstellung der in einem Prozess zum Einsatz kommenden Technologien an (s. Abb. 82).

Die hier gewählte Visualisierung erfolgt in recht anschaulicher Form, indem die bestehenden Systeme grafisch bzw. symbolisch dargestellt werden. Es wurde ausgehend von einer Wertschöpfungskette illustriert, welche technischen Systeme im Verlauf der Prozessabwicklung Anwendung finden.

Diese Darstellung gibt ein gutes Bild der zum Einsatz gelangenden Technologien. Im Bereich automatisierter Produktionssysteme werden auch Anlagenlayouts zur Visualisierung des Prozessablaufes herangezogen. Im Prinzip entspricht dies von der Information her gesehen ebenfalls einem Prozesstechnologiediagramm – ergänzt durch layoutmässige Angaben.

Alternativ zu dieser Darstellung könnte man auch auf eine analoge Darstellungsform wie beim “stellenorientierten Ablaufdiagramm” setzen. Man würde in diesem Fall in der Horizontalen Techniksyste eintragen und den schrittweisen Ablauf des Prozesses darstellen. In komplexeren Fällen wäre diese Form des “technikorientierten Ablaufdiagramms” zu wählen.

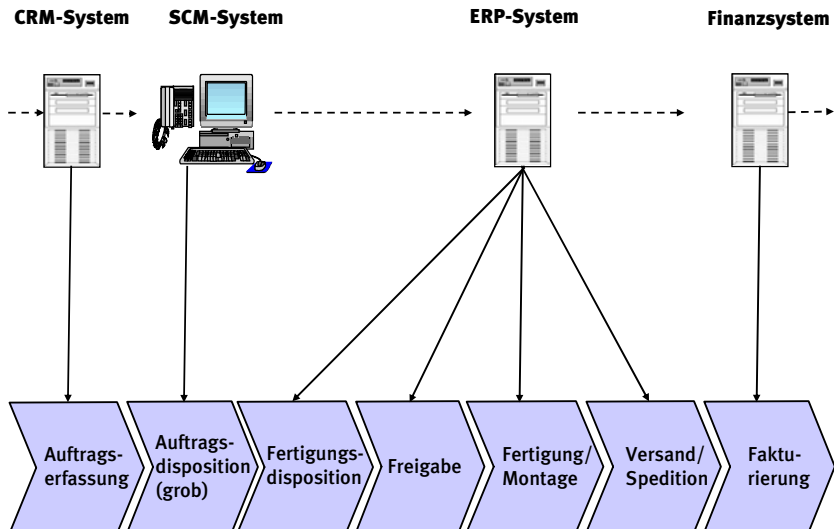


Abb. 82: Prozesstechnologiediagramm

## 7.11 Funktionsunterstützungsdiagramm

*Definition: Funktionsunterstützungsdiagramme stellen primär Funktionen dar sowie sekundär die diese Funktionen unterstützenden Techniksyste-me.*

Man wird beispielsweise klären müssen, welche Systeme den Funktionsbereich “verwalten Aufträge” zu unterstützen haben. Dieser Funktion mögen in der Funktionshierarchie die sechs unten dargestellten Subfunktionen zugeordnet sein (s. Abb. 83).

In einem zweiten Schritt zeichnet man nun ein, welche Systeme der vorliegenden Systemarchitektur die entsprechenden Funktionen unterstützen. Im Gegensatz zum oben beschriebenen “Systemfunktionendiagramm” geht dieses Diagramm von den Funktionen aus und ordnet diesen Techniksубsysteme zu.

Eine solche Darstellung kann einerseits die Systemanwender interessieren, wenn sie Klarheit darüber haben möchten, welche Systemkomponenten funktional Anwendung finden. Andererseits wird ein solches Diagramm insbesondere auch dem Informatiker dienlich sein, um eine Systemplanung vorzunehmen. Dies gilt selbstverständlich insbesondere dann, wenn funktional orientiert implementiert wird.

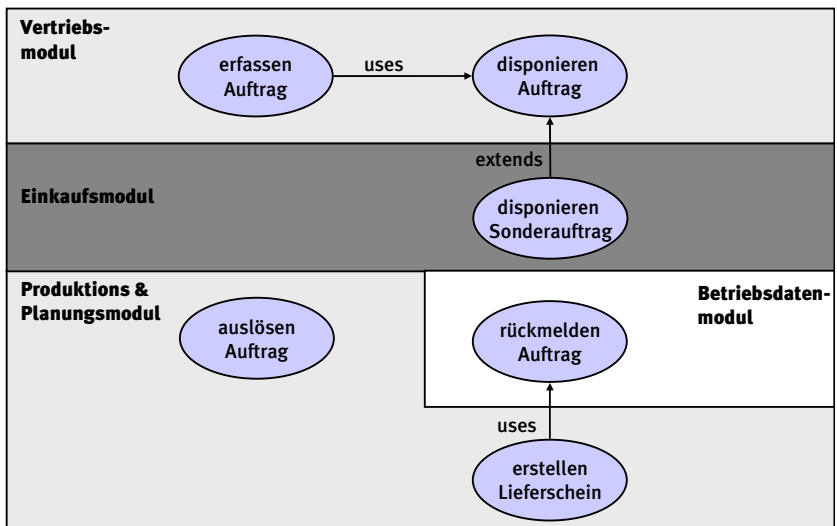


Abb. 83: Funktionsunterstützungsdiagramm

## 7.12 Objektzugriffsdiagramm

*Definition: Objektzugriffsdiagramme stellen primär Objekte dar sowie sekundär die auf diese Objekte zugreifende Techniksysteme.*

Das Objektzugriffsdiagramm zeigt auf, auf welche Datenobjekte von welchen Anwendungssystemen aus zugegriffen wird (s. Abb. 84). Ein solcher Zugriff kann entweder darin bestehen, dass Systeme Datenobjekte lediglich “lesen” oder darin, dass diese Objekte auch “geändert” bzw. “geschrieben” werden können. Selbstverständlich ließe sich auch diese Information visualisieren.

In der unten stehenden Abbildung ist dargestellt, dass das E-Mail-System auf Datenobjekte nur lesend zugreift. Gleiches gilt für das Finanz- & Lohnsystem, welches im Beispiel die Datenobjekte hinunterlädt und anschließend autonom bearbeitet.

Ein solches Diagramm eignet sich auch dazu, die Komplexität einer Systemumgebung zu visualisieren. Außerdem können die technischen Systemelemente durchaus auch Subsysteme eines integrierten ERP-Systems darstellen. In diesem Fall zeigt es Abhängigkeiten von verschiedenen Modulen auf.

Der Unterschied zum oben beschriebenen “Datenobjektdiagramm” liegt darin, dass das vorliegende Diagramm von der Objekthierarchie ausgeht und den gegebenen Objekten Subsysteme zuweist.

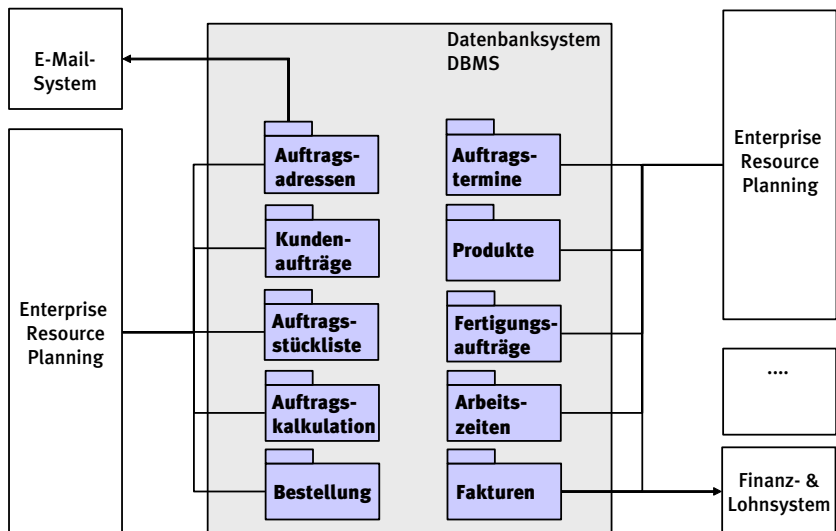


Abb. 84: Objektzugriffsdiagramm

### 7.13 Technikeinsatzdiagramm

*Definition: Technikeinsatzdiagramme stellen primär Stellen dar und sekundär die diese Stellen unterstützenden Techniksysteme.*

Dieses Diagramm veranschaulicht ausgehend von ausgewählten Organisationseinheiten, welche Techniksysteme in diesen zur Anwendung gelangen (s. Abb. 85). Das Systemnutzungsdiagramm stellt demgegenüber dar, welche Organisationseinheiten ein gegebenes System benutzen.

Das Technikeinsatzdiagramm zeigt also auf, welche Stellen innerhalb eines Arbeitssystems welche Technologien einsetzen bzw. diese teilen. Auch dieses Diagramm lässt sich, wie alle anderen, hierarchisch abstufen.

Die nachfolgende Abbildung zeigt, ausgehend von den Unternehmensbereichen, die von ihnen genutzten Informationssysteme. Die Terminologie "Technikeinsatzdiagramm" wurde gewählt, da dieses Diagramm aufzeigt, welche Systeme von verschiedenen Organisationseinheiten gemeinsam genutzt werden. Es lässt auch Rückschlüsse auf die Beurteilung der technischen Autonomie zu.

Das Diagramm entspricht der Spiegelung des Systemnutzungsdiagramms. Es geht als primäre Sicht von der Organisation aus und stellt alle von ihr genutzten Systeme dar. Das Systemnutzungsdiagramm geht von einem Informationssystem oder von Modulen aus.

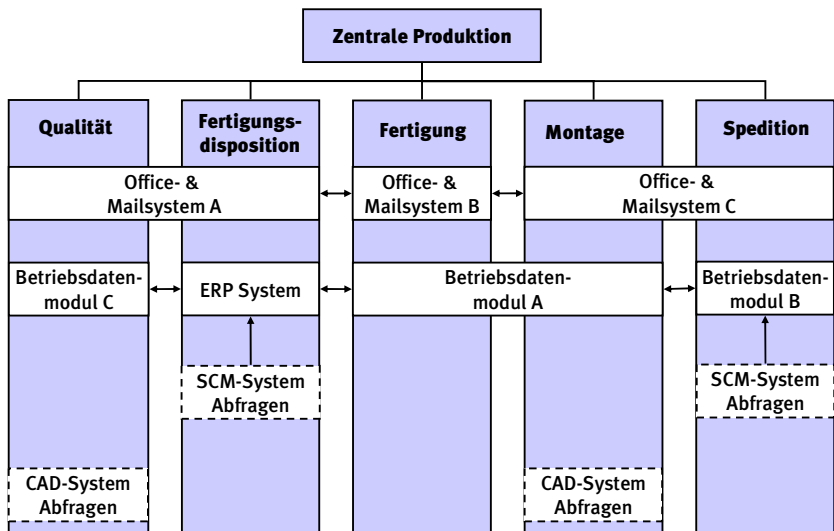


Abb. 85: Technikeinsatzdiagramm



## 7.14 Aspektorientierte Informationssysteme

### 7.14.1 Auswirkungen der Sichten auf die Informationstechnologie

In den vorangegangenen Kapiteln wurde jeweils am Ende der Kapitel auf eine integrierte Methode des entsprechenden Aspektes eingegangen. Man könnte sich nun auch bei der technikorientierten Modellierung fragen, ob es entsprechende integrierte Methoden gibt.

Viel interessanter erscheint allerdings die Feststellung, dass im Prinzip jedes technische Informationssystem selber gerade eine Integration der entsprechenden Aspekte darstellt. In jedem Informationssystem müssen in der einen oder anderen Weise Prozesse vollzogen, Funktionen ausgeführt, Objekte bearbeitet und Aufgaben abgedeckt werden. Es stellt sich nur die Frage, wie und auf welcher Komfortstufe dies geschieht.

Interessanterweise haben sich in der Vergangenheit gerade Systeme entwickelt, die – ganz den unterschiedlichen Methodenfoki entsprechend – ebenfalls einen Fokus auf die eine oder andere Sichtweise legen. So entstanden prozess-, funktions-, objekt-, und aufgabenorientierte Informationssysteme (s. Abb. 86).

IT-System Sicht	Informationssystemtyp	Programmstruktur	Datenstruktur	User-Interface Paradigma
<b>Prozesssicht</b>	Prozess-orientiertes Informationssystem	Strukturierte Sprachen	Filestrukturen	Job-Auswahl / Workflow
<b>Funktionssicht</b>	Funktions-orientiertes Informationssystem	Modulare Sprachen	Datenbanken	Funktionsauswahl / Zeichenorientiert
<b>Objektsicht</b>	Objekt-orientiertes Informationssystem	Objektorientierte Sprachen	Objekte	Objektauswahl / Grafikorientiert
<b>Aufgabensicht</b>	Aufgaben-orientiertes Informationssystem	Kontext-orientierte Sprachen	Objekte	Aufgabenauswahl / Arbeitsplatz-metapher

Abb. 86: Generationen der Informationstechnologie

### 7.14.2 Prozessorientierte Informationssysteme

In den 70er-Jahren wurde mehrheitlich der Prozessaspekt betont. Die Programmzeilen wurden derart zusammengefasst, dass sie integral den Prozess abbilden konnten. Dies führte zu komplexen und kaum mehr durchschaubaren Programmstrukturen mit vielen “goto”-Sprüngen innerhalb des Programms. Die Programmstrukturen waren gewissermaßen ein Abbild der Flow Charts. In den “prozessorientierten 70er-Jahren” wurde für jeden Prozess ein eigenes, vom Programm getrenntes, Datenfile angelegt. Jeder Prozess konnte beispielsweise seine eigenen Daten für “Kunden” anlegen, es entstanden damit aber auch große Redundanzen. Die Daten waren den Prozessen unterworfen.

In den 90er-Jahren gab es ein “revival” der Prozesse. Dementsprechend entstanden auch wieder prozessorientierte Systeme, diesmal aber mit interaktiver Steuerung im Sinne von Workflows (s. Abb. 87). Die Navigation durch diese Systeme erfolgt entlang eines Prozesses. Im Verlauf eines Prozesses werden verschiedene Zustände durchlaufen. Ein Beispiel: Mit jedem Status eines Kundenauftrages werden spezielle Funktionen auf das Auftragsobjekt angewendet: “vorkalkulieren” des Auftrages, “disponieren” des Auftrages, “fakturieren” des Auftrages, etc.

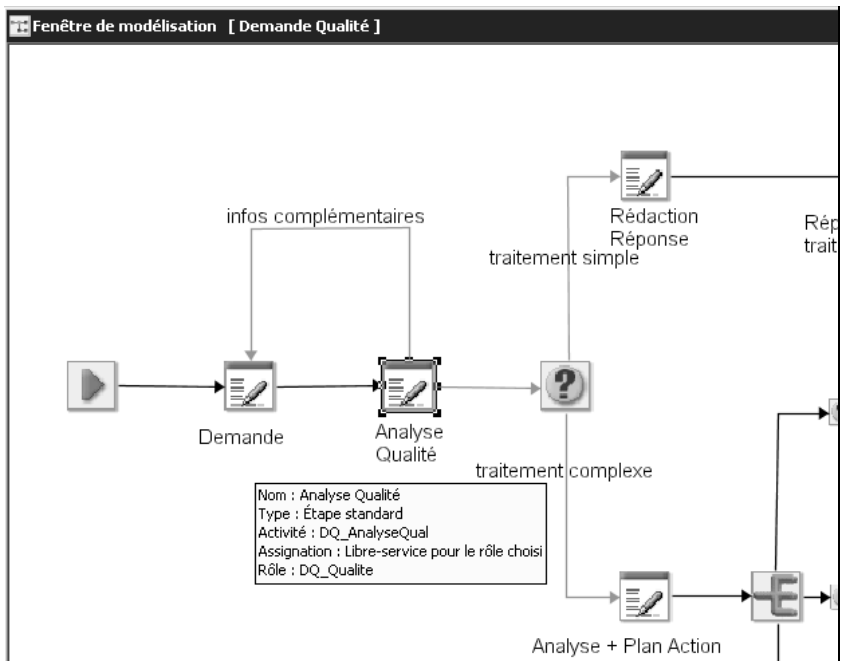


Abb. 87: Workflowsystem “W4-Studio” der Firma W4

### 7.14.3 Funktionsorientierte Informationssysteme

In den 80er-Jahren erkannte man, dass Job-Programme kaum mehr wartbar sind. Es wurde versucht, die Codezeilen nach funktionalen Gesichtspunkten zusammenzufassen. Es entstanden funktionale Module wie beispielsweise ein Modul "Rechnungswesen".

Die meisten Standardlösungen sind aus "historischen Gründen" nicht darauf ausgerichtet, integriertes Arbeiten von Generalisten zu unterstützen. Funktionale Systeme wurden dazu konzipiert, eher funktional arbeitsteilige Unternehmen zu unterstützen. Dieser herkömmliche Ansatz schlägt sich in einer funktionalen Menüstruktur und in Modulwechseln bei der Abarbeitung eines Prozesses nieder. Einzelne Funktionsmodule für Vertrieb, Logistik, Einkauf, etc. werden dafür meist mit großer Funktionalität abgedeckt. Dies kann indessen die Bedienung komplizieren und die Benutzer dazu zwingen, einen großen Aufwand an Schulung zu betreiben, um auch nur die Grundzüge des Systems kennen zu lernen. Man könnte diese Systeme als "funktional-orientierte Informationssysteme" bezeichnen. Diese Vorgehensweise sei nochmals mittels Beispielmasken illustriert (s. Abb. 88). Beim System R/3 von SAP handelt es sich um ein typisch funktionales System. Erst wenn die richtige funktionale Maske gefunden ist, wählt man ein Objekt und startet die Aktivität. Ein funktionales System verfügt also typischerweise über eine Menüstruktur, die die Funktionsmodule widerspiegelt: Finanzwesen, Logistik, Personalwesen, etc.

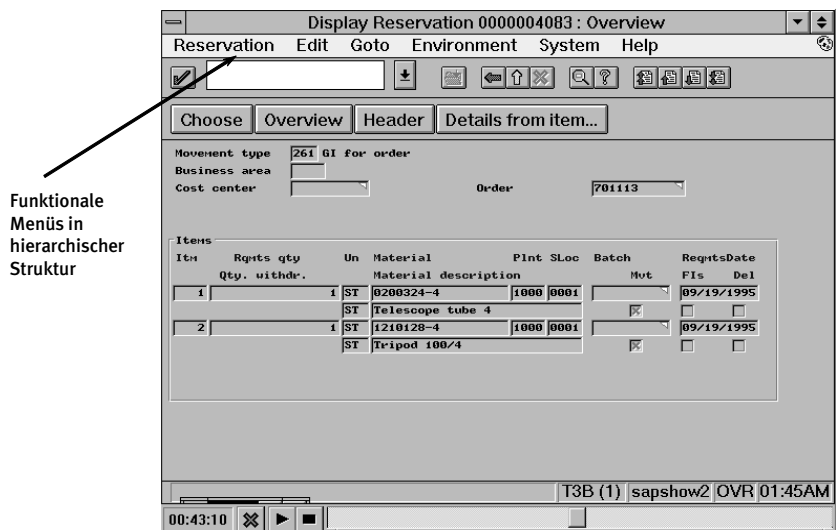


Abb. 88: Funktionale Navigation am Beispiel "R/3 - System" von SAP

#### 7.14.4 Objektorientierte Informationssysteme

Traditionelle Softwaresysteme bestehen aus einem großen monolithischen Programmblock, der die Mehrzahl der Funktionen beinhaltet. Derart strukturierte Systeme wuchsen in der Vergangenheit zu Programmkomplexen an, die bei Änderungen immer unüberschaubarer wurden, worunter auch die Qualität litt. Ein objektorientiertes Softwaresystem besteht im Gegensatz zu den herkömmlichen, auf Funktionen ausgerichteten Systemen aus einer Kollektion von kleinen Codeeinheiten, Objekte genannt, die miteinander kommunizieren. Im objektorientierten Sinn stellt ein "Programm" eine sinnvolle Kollektion von Objekten oder Codeeinheiten dar.<sup>3</sup>

Stehen bei der strukturierten Analyse die Funktionen im Zentrum der Aufmerksamkeit (z.B. Vertrieb, Materialverwaltung, ...), so sind es bei der objektorientierten Analyse die Objekte, auf denen die genannten Funktionen ausgeführt werden (z.B. Kunden, Teile, ...). Dies ist insbesondere bei Änderungen und Ergänzungen des Systems im Funktionsumfang ein Vorteil.

Seit den 60er-Jahren ist man im Begriff, die Codezeilen nach der objektorientierten Sichtweise zu bündeln (die Idee entstand allerdings schon in den 60er-Jahren). Die objektorientierten Programmstrukturen entsprechen damit auch 1:1 den Objektstrukturen. Wenn also 20 Objekte identifiziert werden, dann wird das objektorientierte Informationssystem, vereinfacht gesprochen, auch über 20 kleine Programmbausteine verfügen. Jedes Objekt wird als solches mit seinen Attributen und seinen Methoden beschrieben. Die Methoden entsprechen gerade diesen kleinsten Codeeinheiten.

In den 60er-Jahren wurde die Kapselung der Daten nochmals weitergetrieben. Der Zugriff auf die Objekte ist, wie bereits erwähnt wurde, nur mit "Erlaubnis" des betreffenden Objektes möglich. Im objektorientierten Sinn ist also der direkte Zugriff auf eine Datenbank nicht erlaubt – die Trennung zwischen Daten und Funktionen wird aufgehoben. Wenn es noch eine Datenbank gibt, dann holt sich das Objekt die Daten gewissermaßen selber aus der Datenbank und übergibt die entsprechenden Attribute oder Werte auf Anfrage hin anderen Objekten. Die Daten sind damit unter vollständiger Kontrolle der Objekte ("Kapselung der Daten").

---

<sup>3</sup> "Object-oriented programming is not so much a coding technique as it is a *code packaging technique*, a way for code suppliers to encapsulate functionality for delivery to consumers" (Brad Cox 1991, S. 13).

### Objektorientierte Navigation

Ein objektorientiertes System geht, wie erwähnt, vom Objekt aus. Die unten stehende Darstellung zeigt einen Ausschnitt aus einem Text an (s. Abb. 89). Der selektierte Text entspricht einem "Textobjekt". Ausgehend vom Objekt hat man nun die Möglichkeit, aus den auf diesem Objekt beruhenden Funktionen eine spezifische Funktion auszuwählen. Diese Art der Interaktion wird auch als "object first, action second" bezeichnet. Diese objektorientierten Interaktionstechniken wurden durch Apple populär gemacht (allerdings nicht erfunden).

Worin liegt nun für den Anwender der Unterschied zwischen einem prozess- einem funktions- und einem objektorientierten Programm? Machen wir hierzu ein Beispiel aus dem Personalbereich. Nehmen wir an, ein Benutzer eines Informationssystems soll den Lohn für "Hans Müller" auszahlen. In den 70er-Jahren erfolgte eine solche Auszahlung vielleicht nur einmal im Monat in einem so genannten Batch-Lauf. Man musste also warten, bis der Job gestartet wurde und gab an, dass man auch den Lohn für den Hans Müller ausgezahlt haben wolle – hierfür wurde eine Lochkarte mit den Daten des Hans Müllers eingeschoben. Im funktionsorientierten Ansatz geht es schon bedeutend interaktiver: Hier ist der Ausgangspunkt die Funktion. Man wird also über ein hierarchisches Menü beispielsweise im Modul "Finanzen" den entsprechenden Menüpunkt "Lohnzahlung" wählen. Dieses Funktionsmodul ist bestimmt, eine Auszahlung vorzunehmen. Das Finanzmodul wird unter Umständen noch auf das Personalmodul zugreifen oder umgekehrt. Im objektorientierten Ansatz steht das Objekt im Vordergrund. Der Benutzer wird sich zunächst das Objekt "Müller Hans" suchen. Anschließend wird er oder sie die Funktion "zahlen" auf diesem Objekt ausführen.

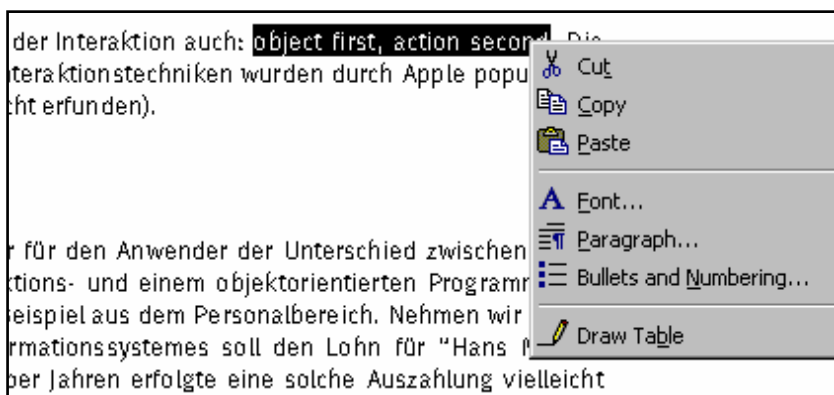


Abb. 89: Objektorientierte Navigation (Microsoft Word)

### 7.14.5 Aufgabenorientierte Informationssysteme

Im Gegensatz zu den zuvor besprochenen Informationssystemen stellen “aufgabenorientierte Informationssysteme” die Aufgabe in den Vordergrund. Die primäre Navigation erfolgt entsprechend der konkret zu vollziehenden Aufgabe. Dies bedeutet nichts anderes, als dass für jede Aufgabe bzw. Rolle im Unternehmen und für jede “Teilaufgabe” jeweils eine entsprechende Sichtweise auf das Informationssystem vorhanden ist und entsprechende “Bildschirme” angeordnet werden (s. Abb. 90). Dies entspricht dem Postulat der “Arbeitsplatzmetapher”. Die für eine Aufgabe notwendigen Arbeitsobjekte werden auf dem Arbeitsplatz bereitgestellt und die Aufgabe erlaubt es, die entsprechenden Operationen auszuführen. Die grundlegenden Elemente sind daher ebenfalls Objekte, nun aber im entsprechenden Kontext angeordnet. Dies entspricht dem Konzept von System EVO, einer integrierten ERP-Lösung für KMUs (s. Abb. 90).

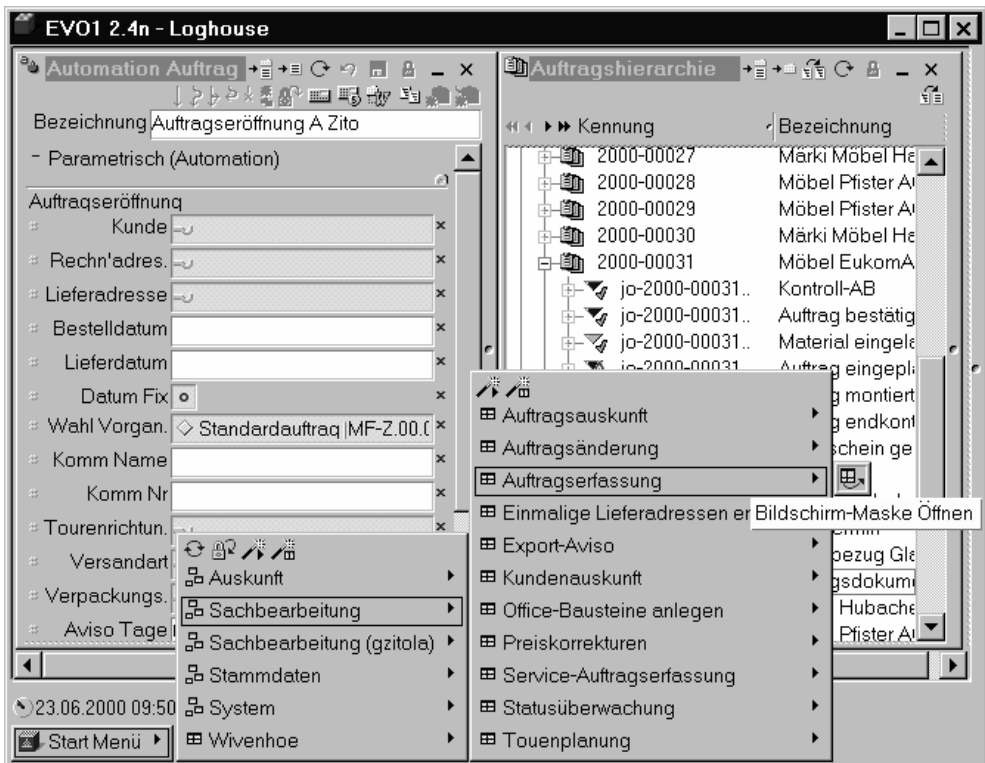


Abb. 90: Aufgabenorientierte Navigation (EVO logix, TRILAB, Schweiz)

## 7.15 Vollständiges Modell

Aufbauend auf den oben beschriebenen Methoden kann nun die vollständige Systemmodellierungsmatrix dargestellt werden (s. Abb. 91). In Ergänzung zur oben stehenden Matrix, die das System nur logisch, d.h. technikunabhängig, betrachtet, bildet das vollständige Matrixmodell auch den Bezug der Systemelemente zur Technik mit ab.

Diese erweiterte Matrix bildet den zentralen Punkt dieses Buches. Sie bildet die Grundlage zu einer "vollständigen sozio-technischen Analyse" von Informationssystemen.

In den nachfolgenden Kapiteln wird gezeigt, dass gerade auch die sozio-technischen Diagramme, die die Beziehungen zwischen der Prozess-, Aufgaben- und Techniksicht näher untersuchen, von großer Bedeutung für die Konzeption technikgestützter Systeme sind. Dies betrifft jene Diagramme, die sich im Schnittpunkt der Prozess- bzw. Aufgabensicht und der Techniksicht befinden.

Wie oben erwähnt, wird die Techniksicht auf die Systemarchitektur beschränkt und beinhaltet keinesfalls die Hardwaresicht. Dies würde nochmals eine weitere Systemdimension eröffnen. Im Bereich der Hardwareplanung könnten solche Diagramme einen gewissen Nutzen bringen.

<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht	Techniksicht
Prozesssicht	<b>Prozessmodell</b>	Funktionen-Blockdiagramm	Zustandsübergangsdiagramm	Stellenorientierter Informationsfluss	Systemschnittstellendiagramm
Funktionssicht	Flussdiagramm	<b>Funktionsmodell</b>	Class-Responsibilities-Collaborators	Stellenfunktionen-diagramm	Systemfunktionen-diagramm
Objektsicht	Sequenzdiagramm	Datenflussdiagramm	<b>Objektmodell</b>	Arbeitsobjekt-diagramm	Datenobjekt-diagramm
Aufgabensicht	Stellenorientiertes Ablaufdiagramm	Use-Case-Diagramm	Kollaborationsdiagramm	<b>Organisationsmodell</b>	Systemnutzungsdiagramm
Techniksicht	Prozess-technologie-diagramm	Funktionsunterstützungsdiagramm	Objektzugriffsdiagramm	Technikeinsatzdiagramm	<b>Systemarchitekturmodell</b>

Abb. 91: Vollständiges Matrixmodell

**7.16****Literatur zur technikorientierten Modellierung**

- Bauknecht, K., Zehnder, C.A. (1997): Grundlagen für den Informatikeinsatz. Stuttgart: Teubner.
- Cox, B., Novobilski, A. (1991): Object-oriented Programming: An Evolutionary Approach. Reading, MA: Addison-Wesley.
- Heinrich, L.J. (1996): Systemplanung: Planung und Realisierung von Informatik-Projekten. München: Oldenbourg.
- Jacob, O. (1989). Aufgabenintegrierte Büroinformationssysteme: Allgemeines Datenmodell und Probleme der Realisierung. Heidelberg: Physica.
- Mertens, P. (2000). Integrierte Informationsverarbeitung 1: Administrations- und Dispositionssysteme in der Industrie, 12. überarbeitete Auflage. Wiesbaden: Gabler.
- Mertens, P., Bodendorf, F. et al. (2004). Grundzüge der Wirtschaftsinformatik, 8. Auflage. Berlin: Springer.
- Österle, H. (Hrsg.) (1990): Integrierte Standardsoftware: Entscheidungshilfen für den Einsatz von Softwarepaketen. Band 2: Auswahl, Einführung und Betrieb von Standardsoftware. München: Angewandte Informations Technik – AIT.
- Scheer, A.-W. (1990). Wirtschaftsinformatik: Informationssysteme im Industriebetrieb. Berlin: Springer.
- Schmager, B. (1989). Betriebsdatenerfassung und -verbreitung als Instrument der betrieblichen Informationsgewinnung: Ein Ansatz zur Reduzierung der Distanzen bei der BDEV unter Berücksichtigung ziel- und benutzerorientierter Aspekte. Hamburg: Technische Universität Hamburg-Harburg.
- Stahlknecht, P. (2002). Einführung in die Wirtschaftsinformatik, 10. überarbeitete Auflage. Berlin: Springer.
- Störmer, W., Droge, M. (1997). Personalzeiten und Betriebsdaten: Konzepte, Lösungen und Erfahrungen aus der Praxis. München: Hanser.
- Zehnder, C.A. (2002): Informatik-Projektentwicklung: Eine Einführung für Informatikstudenten und Praktiker, 4. Aufl. Zürich: vdf Hochschulverlag.







# Teil 2

## Projektentwicklung

Der zweite Teil des Buches beschäftigt sich mit der Projektentwicklung und den Vorgehensmodellen zur IT-Systemgestaltung. Zieht man wieder den Vergleich des Hausbaus heran, dann ging es im ersten Teil um die Pläne des Hauses und nun um die Tätigkeiten und das Vorgehen, wie man von der ersten Idee zum schlüsselfertigen Haus gelangt, selbstverständlich unter Zuhilfenahme dieser Pläne. Der Vergleich ist nicht zufällig: Auch beim Bau eines Hauses hat man es mit einem komplexen System zu tun und die Gestaltung kann nur mit einem strukturierten Vorgehensmodell bewältigt werden.

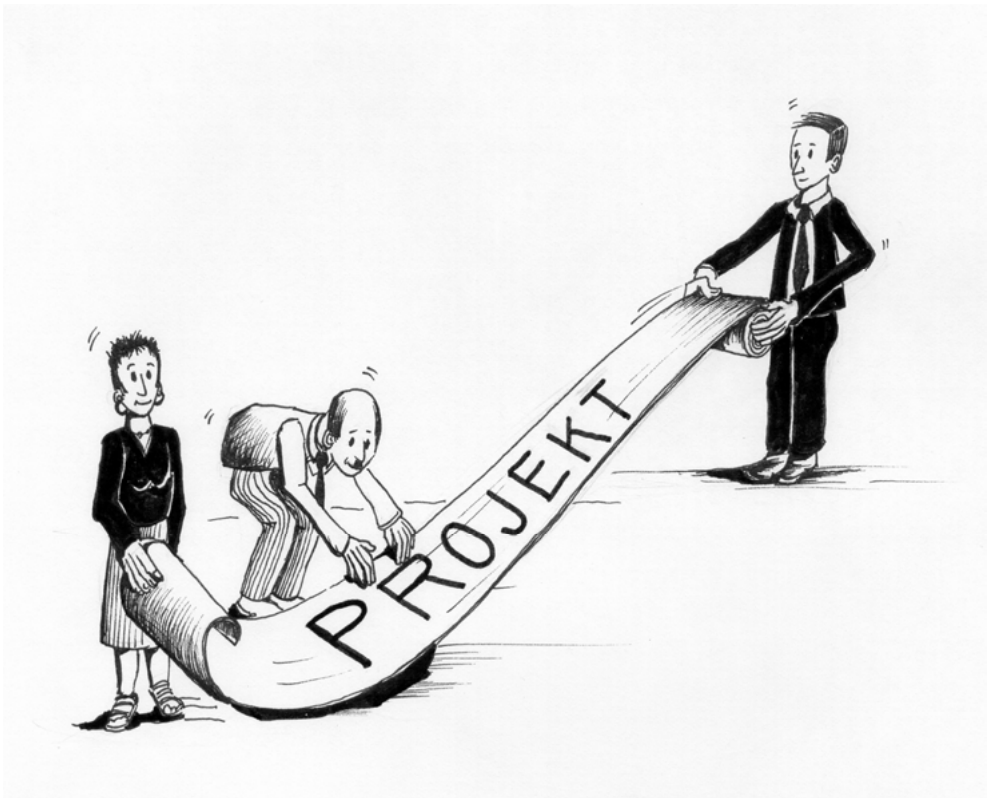
### **Vorgehensmodelle mit klar definierten Phasen**

In den vergangenen Jahren haben sich diverse Vorgehensmodelle für IT-Projekte entwickelt. Die gemeinsame Basis aller Vorgehensmodelle besteht darin, dass das Vorgehen a) in klar definierte Phasen unterteilt wird und b) in jeder Phase ein klares und vorgängig definiertes Ziel erreicht werden muss.

Teil 2 dieses Buches beschreibt zunächst die unterschiedlichen Vorgehensmodelle und anschließend die auszuführenden Tätigkeiten in je einem eigenen Kapitel (Projektantrag, Grobkonzeption, Konzept, Verträge, Detailkonzepte, Design, Implementation etc.). Die Tätigkeiten kommen in allen Vorgehensmodellen vor. Je nach Vorgehensmodell werden sie aber auf unterschiedlich umfangreiche Art und Weise in den Projektphasen ausgeführt. Die Kapitelstruktur darf daher auch keinesfalls ausschliesslich als Anleitung zum Wasserfallmodell aufgefasst werden.

### **Projektmanagement**

Unabhängig vom gewählten Vorgehensmodell bildet ein gutes Projektmanagement ein unabdingbarer Erfolgsfaktor jedes Projektes. Den Grundzügen des Projektmanagements ist daher ein eigenes, zusätzliches Kapitel gewidmet.



# 8

## Vorgehensmodelle der Projektabwicklung

### Kapitelinhalt

- Probleme und Schwierigkeiten von IT-Projekten
- Vorgehensmodelle: Wasserfall- & Spiralmodell
- Der mehrfach wiederkehrende Problemlösungszyklus
- Kombinationsmöglichkeiten und Tailoring des Vorgehensmodells

Vorgehensmodelle zur Projektabwicklung können als in Phasen strukturierte Prozesse verstanden werden, die mittels eines methodischen Vorgehens Informationssysteme gestalten sollen. Die Gestaltung von Systemen kann nach zwei Grundvorgehen erfolgen:

- a) Man plant zuerst das Gesamtvorhaben mit allen Details und implementiert dann nach diesem Plan (Wasserfallmodell);
- b) Man beginnt ohne eine detaillierte Gesamtplanung relativ rasch mit der Implementation eines Teilbereichs und ergänzt das System fortlaufend (Spiralmodell).

Das Wasserfallmodell postuliert ein sequenzielles Vertiefen der Analyse und Konzepttätigkeiten. Die Analysen werden möglichst umfassend vorgenommen. Erst nach dem Abschluss aller Analysen wird mit der Implementation begonnen. Das Spiralmodell postuliert, dass das System in jeder Phase ein Stück weit entwickelt und nach jeder Phase den Benutzern übergeben wird.

## 8.1 Probleme und fehlender Erfolg von IT-Projekten

Die Komplexität von IT-Projekten sollte unter keinen Umständen unterschätzt werden. Dies zeigen die große Anzahl gescheiterter oder stark verzögerter Projekte und die in den vorangegangenen Kapiteln angeführte Mehrdimensionalität von sozio-technischen Systemen. In der Schweiz sind unter anderem in letzter Zeit folgende Beispiele bekannt geworden. Vermutlich erfährt man aber viele erfolglose Projekte gar nicht erst:

- Elektronische Börse der Schweiz benötigte mehrere Anläufe.
- Im Dezember 1998 wurde publik, dass das 250 Mio. Franken teure Computersystem für die Produktion des Kleinwagens Smart mit Abstürzen für massive Störungen sorgte. Der verantwortliche Implementationspartner setzte 150 Personen zur Lösung der selbstverursachten Probleme ein.
- Die Verzögerung des deutschen Autobahn-Maut-Systems (Toll Collect) führte im Jahr 2004 zu einer landesweiten Diskussion.

Zum Abschluss sei noch eine mittlerweile etwas ältere, aber bereits klassische Studie aus den USA angeführt (s. Abb. 92). Es wurde untersucht, wie erfolgreich Software-Projekte in der Verwaltung waren. Das Resultat war erschreckend: Lediglich fünf Prozent der übergebenen IT-Lösungen wurden nach Lieferung tatsächlich gebraucht. Diese Erkenntnisse bestärkten die Notwendigkeit einer methodisch sauberen Projektabwicklung.

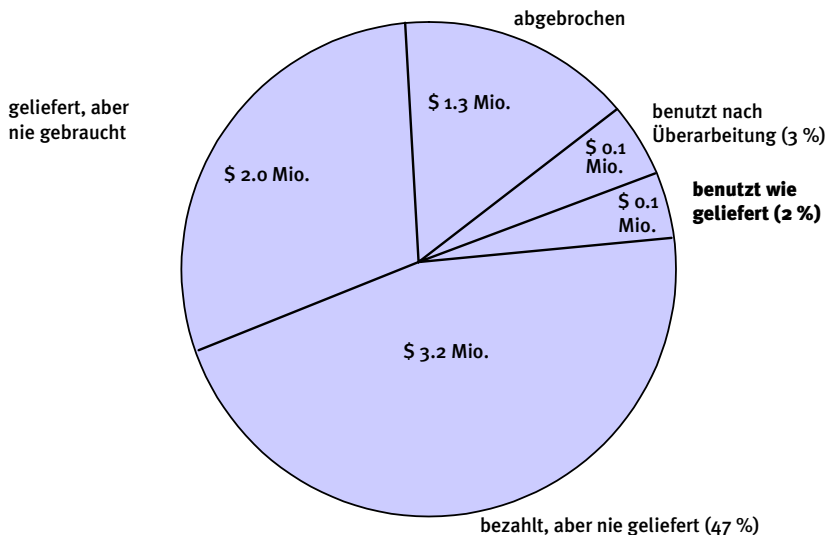


Abb. 92: IT-Projekte in der amerikanischen Verwaltung (Cox 1991)

### 8.1.1 Zeit- und Kostenüberschreitungen in Projekten

Zugegebenermaßen ist diese Studie für heutige Verhältnisse kaum mehr repräsentativ. Etwas neuere Studien der Standish Group kommen allerdings immer noch zu recht drastischen Ergebnissen (s. Abb. 93).

Interessant an den Untersuchungen der Standish Group ist die Betrachtung der Kosten- und Zeitüberschreitung. Im Durchschnitt der betrachteten Projekte wurde eine Zeit- und Kostenüberschreitung von 200 Prozent festgestellt (Standish 2003).

Die angeführten Beispiele lassen den Schluss zu, dass mit Informatikprojekten immer einiges schief gehen kann (Inhalt, Zeit, Kosten). Ein Grund für diese Probleme mag darin liegen, dass sich die Softwareentwicklung nach wie vor in einer vorindustriellen Phase – etwa entsprechend einer mittelalterlichen Manufaktur – befindet.

So ist der Aufwand für die Individualentwicklung einer Softwarelösung meist unverhältnismäßig hoch. Die Systeme werden immer überwiegend von Grund auf neu programmiert. Es gibt nur wenige Komponenten, wie das bei der industriellen Fertigung schon seit langem der Fall ist.

Kein Automobilhersteller hat eine gleichermaßen hohe "Fertigungstiefe" und stellt beinahe alle Komponenten selber her – die größte Anzahl der verwendeten Teile werden zugekauft.

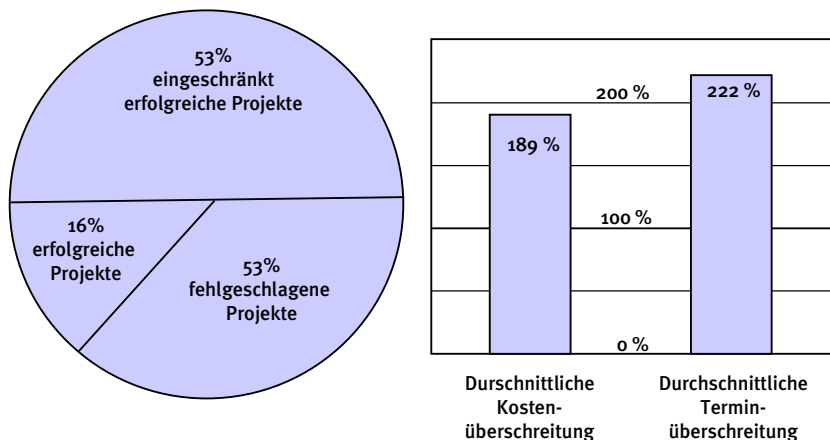


Abb. 93: Erfolg von IT-Projekten (Standish Group, 2003)

Aufgrund dieser Ausführungen wird klar, dass der Einsatz von Informationssystemen oftmals mit erheblichen Schwierigkeiten verbunden ist. Die Problemgruppen lassen sich dabei wie folgt identifizieren:

#### **Technisch unzulängliche Systeme**

Viele der Hauptkritikpunkte an Informationssystemen betreffen gemäß Untersuchungen deren technische Unzulänglichkeit wie *unzureichende Funktionalität, geringe Flexibilität bzw. geringe Benutzungsfreundlichkeit* (Strohm 1995, S. 102). Eigentlich ein vernichtendes Urteil: Die eingesetzten Informationssysteme sind weder in der Funktionalität ausreichend noch einfach zu nutzen.

#### **Lange Einführungsdauer**

Die Probleme in der Handhabung von Informationssystemen drücken sich z.B. in sehr langen Einführungszeiten von ERP-Systemen aus: Strohm gibt eine mittlere Projektdauer von 2.3 Jahren an (a.a.O.). Martin gelangt mit 4.3 Jahren zu noch viel drastischeren Ergebnissen (Martin 1993, S. 177). Beide Untersuchungen stellen außerdem fest, dass große Unterschiede bei der Einführungsdauer bestehen. Die Schwankungen reichen dabei von einem halben Jahr bis zu zehn Jahren. Eine Ursache dieser Schwankungen liegt im gewählten System selbst. Aufgrund der Ergebnisse genannter Untersuchungen kann sehr wohl von einem Einfluss der Informationssysteme auf den Einführungserfolg ausgegangen werden (Martin 1993, S. 208). Neuere diesbezügliche Studien sind dem Autor leider nicht bekannt.

#### **Mangelhafte Zielerreichung**

Strohm (a.a.O.) führt aus, dass die Zielerreichung beim Einsatz von Informationssystemen als äußerst niedrig einzustufen sei. So hätten beispielsweise bei ERP-Einführungen nur 29 % der Betriebe eine Steigerung der Termintreue erzielt, obwohl hauptsächlich diese durch die Betriebe angestrebt worden sei. Hierzu sei allerdings bemerkt, dass die Erwartungen oft zu hoch gesteckt sind. Falsche Versprechungen von Systemanbietern mögen dazu beigetragen haben.

#### **Wirtschaftlichkeit von Informationssystemen**

Soweit ersichtlich, fehlt es an verlässlichen Untersuchungen zur Wirtschaftlichkeit von ERP-Software. Martin resümiert, aufgrund der genannten Nachteile müsse die Wirtschaftlichkeit einer Reihe der untersuchten ERP-Einführungen jedenfalls bezweifelt werden (a.a.O., S. 180). In Zusammenhang mit Investitionen in die Technik postulieren andere Autoren deshalb eher Zurückhaltung: "Dass lokale Selbstregulation darüber hinaus u.U. erheblich effizienter und kostengünstiger sein kann als ein technisch ausgeklügeltes PPS-System (...)" (Ulich 2001).



## 8.2 Methoden der Projektabwicklung

Die im obigen Abschnitt aufgezeigten Probleme können aufgrund der gewonnenen Erfahrungen nur *mit methodischen Mitteln* angegangen werden.<sup>4</sup> Damit besteht Hoffnung, dass Projekte erfolgreicher verlaufen.

Methoden des Engineerings komplexer Informationssysteme entstanden erstmals in den 60er-Jahren in Zusammenhang mit der Diskussion über die so genannte Softwarekrise (Sommerville 1989, S. 3). Die damaligen Techniken und Methoden befriedigten nicht, um den stark gestiegenen informationstechnischen Möglichkeiten und der daraus resultierenden Komplexität Rechnung zu tragen.

Viele Techniken des Engineerings von Informationssystemen unterliegen im Prinzip den genau gleichen Kriterien wie das Systems Engineering (Schönsleben 2001). Dies gilt insbesondere für das Prinzip der phasenweisen Systemgestaltung.

Die Gesamtheit aller Phasen bezeichnet man üblicherweise als Software-Lebenszyklus bzw. "Software Lifecycle". Bei der phasenweisen Systemgestaltung geht man von inhaltlich wie auch zeitlich klar abgegrenzten Projektphasen aus. Das entspricht dem Postulat "vom Groben zum Detail" des Systems Engineering sowie dem Grundsatz, in sich abgegrenzte und klar überschaubare Projekteinheiten zu schaffen (Hall 1962, Chestnut 1966, Daenzer 1976).

Die in den 60er-Jahren entwickelten Methoden der Systemgestaltung wurden in den Grundsätzen ins Engineering von Informationssystemen übernommen. Sommerville (1989, S. 6) umschreibt dies folgendermaßen: "(...) thus a model of the software development process which was derived from other engineering activities was suggested".

Da die Grundlagen des Engineerings von Informationssystemen auf den Grundlagen des Systems Engineering basieren, sollen nachfolgend dessen wesentlichste Grundlagen kurz zusammengefasst dargelegt werden. Für ein vertieftes Studium des Systems Engineerings wird auf die entsprechende Literatur verwiesen (Haberfellner, 2002).

---

<sup>4</sup> Eine große Anzahl gescheiterter oder verzögerter Software-Projekte führte zu dieser Erkenntnis (s. o.).

### 8.3 Das Lebensphasenmodell gemäß Systems Engineering

Die Methodik des Systems Engineering ist eine allgemeine Wegleitung zur zielgerichteten Gestaltung technischer Systeme. Dies bedeutet, dass sie nicht speziell auf IT-Systeme ausgerichtet ist. In Teil 1 wurden bereits einige Grundlagen des Systems Engineerings wie der “Systembegriff” und die wesentlichsten Merkmale von Systemen dargelegt.

Das Systems Engineering bietet nun neben diesem “Systemdenken” auch ein allgemeines Vorgehens- bzw. “Lebensphasenmodell”, welches bezweckt, den Prozess der Systemgestaltung in klare Phasen zu unterteilen (s. Abb. 94). Der Vorteil von Phasenmodellen besteht im Zwang zur Erarbeitung von Zwischenergebnissen, zum Einbezug des Auftraggebers und zur Staffelung der Kosten und Termine. Mitunter soll mehr Sicherheit in der Projektentwicklung entstehen.

In jeder Phase wird das System detaillierter gestaltet. Im Grobkonzept wird das Gesamtsystem betrachtet und es werden mit vertretbarem Aufwand Abklärungen getroffen. In der Hauptstudie wird ein Gesamtkonzept entwickelt und die Detailstudien behandeln Untersysteme. In den Realisierungsphasen wird das System gebaut und eingeführt. Das Lebensphasenmodell spiegelt damit den zentralen Grundgedanken “vom Groben zum Detail” wieder, indem es zunächst einen generellen Lösungsrahmen festlegt und diesen schrittweise einengt. Dies entspricht einer stufenweisen Vertiefung der Analyse der Systemhierarchie in jeder Lebensphase.

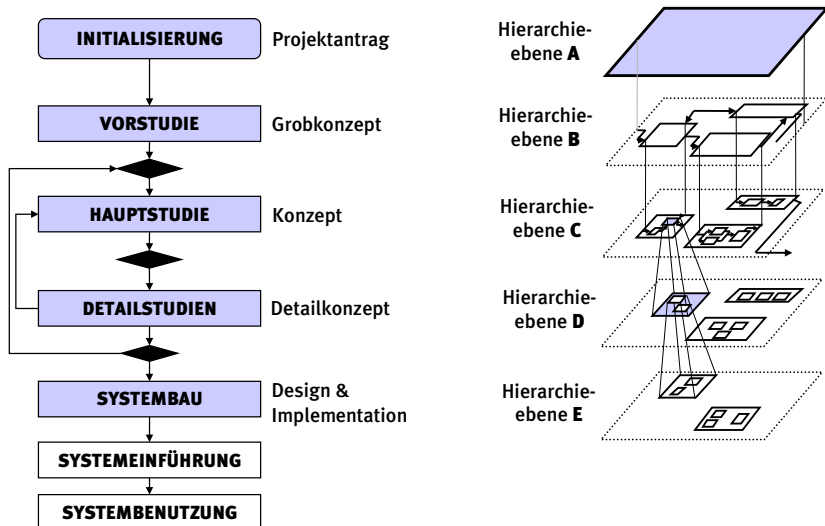


Abb. 94: Lebensphasenmodell des Systems Engineering

### 8.3.1 “Vom Groben zum Detail”

Die nachfolgende Darstellung soll dieses Prinzip der schrittweisen Einarbeitung der Lösung, ausgehend von einem generellen Lösungsrahmen bis zur fertigen Lösung, nochmals illustrieren. Nehmen wir das Beispiel des Customer Relationship Management Systems (CRM-System) zur Pflege der Kundenkontakte, Kundenmailings, Vertriebssteuerung etc. Dieses Beispiel wird uns in der Folge stets begleiten (s. Abb. 95).

In der Phase des Projektantrags werden verschiedene Projektvarianten geprüft werden – beispielsweise von der begrenzten Schnelllösung eines Servicesystems bis hin zum integrierten Gesamtsystem. Es wird nun eine Projektwahl getroffen und ein entsprechender Projektantrag erstellt.

Wenn es sich um ein größeres Projekt handelt, dann folgt in einem nächsten Schritt die Erstellung eines Grobkonzeptes. Dieses prüft die best geeignete Variante für das Unternehmen unter Berücksichtigung der Nutzenpotenziale und der Wirtschaftlichkeit. Es resultiert daraus z.B. die Grobkonzeptvariante (“CRM Light”). Aufbauend auf diesem Lösungsprinzip wird das System sukzessive weiter konzipiert und spezifiziert. Aus der Summe einzelner Konzeptanforderungen je Geschäftsprozess geht hervor, welches die beste Gesamtkonzeption darstellt. Aufbauend auf einer eventuellen Evaluation wird das System im Detailkonzept weiter spezifiziert und die Lösung eingeeengt. Allerdings werden wiederum pro Geschäftsprozess je einzelne Detailkonzepte erstellt werden müssen.

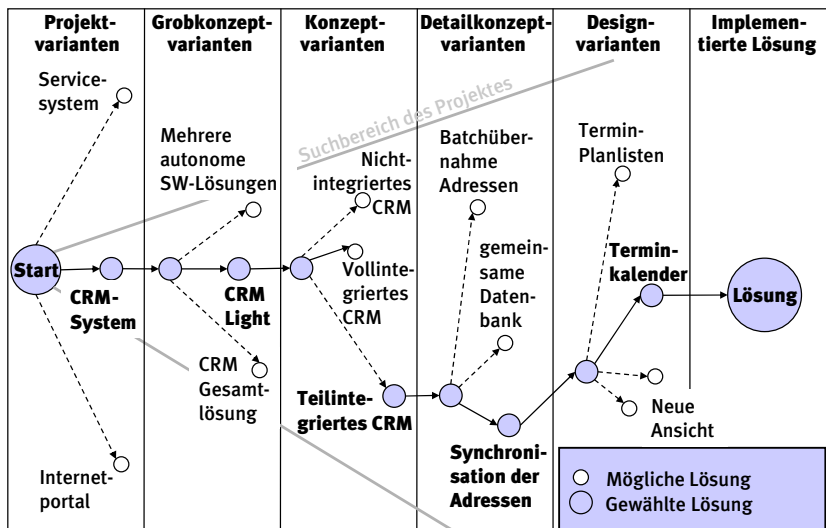


Abb. 95: Stufenweise Festlegung der Lösung

## 8.4 Sequenzielles Vorgehensmodell: Wasserfallmodell

### 8.4.1 Vorgehen gemäß Systems Engineering

Gehen wir zunächst auf das klassische Wasserfallmodell des Software Engineerings ein, welches eine große Verbreitung erlangt hat. Dieses unterliegt im Prinzip denselben Kriterien wie das Systems Engineering. Dies gilt insbesondere für das Prinzip der sequenziellen Systemgestaltung nach dem Grundsatz “vom Groben zum Detail”. Man geht von inhaltlich wie auch zeitlich klar abgegrenzten Phasen aus, die in Bezug auf den Inhalt und die wahrgenommenen Tätigkeiten streng sequenziell und vollständig durchschritten werden. “Vollständig” heißt, dass z.B. das “Grobkonzept” nur in einer Phase erfolgt, dafür umfassend und bezogen auf das gesamte denkbare System. Die Ergebnisse der Phase werden in der nächsten Phase weiter konkretisiert (s. Abb. 96). Die erste Darstellung dieses Vorgehens stammt von Royce um 1970 (Boehm 1986, S. 30). Die exakten Phasen können sich von Projekt zu Projekt und je nach dem Fokus auf Individualentwicklung oder Standardsoftwareeinführung noch leicht unterscheiden. Dies ändert aber nicht grundsätzlich an der Philosophie des Vorgehensansatzes.

Das begleitende Projektmanagement gehorcht den allgemeinen Regeln des Projektmanagements. Die Projektantragsphase ist gewissermaßen sowohl Bestandteil der Systemgestaltung als auch des Projektmanagements.

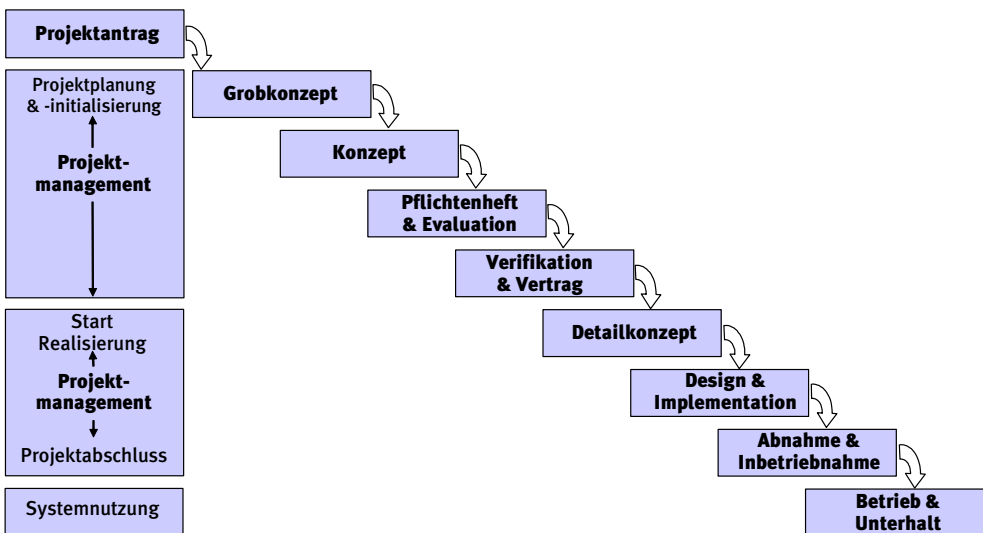


Abb. 96: Sequenzielle Abfolge von Tätigkeiten im Wasserfallmodell

### 8.4.2 Grundsätze und Postulate des Wasserfallmodells

Das Wasserfallmodell ist durch einige Grundsätze geprägt, welche nachfolgend angeführt sind:

- **Phasenkonzept:** Es werden in sich abgegrenzte, klar überschaubare Phasen mit definierten Meilensteinen geschaffen.
- **Vom Groben zum Detail:** Das System wird vom Grobkonzept bis zur Detailstudie stufenweise verfeinert und sukzessive detailliert.
- **Sequenz:** Tätigkeiten werden in jeder Phase sequenziell ausgeführt: Grobkonzept, Detailkonzept, Programmierung, Einführung etc. stellen für sich je eine eigene Phase dar.
- **Vollständigkeit:** Die Phasentätigkeiten werden dafür sehr umfassend vorgenommen und erstrecken sich über den vollständigen Eingriffsbereich (z.B. Konzeption aller Systembereiche).
- **Meilensteine:** Das Erreichen von Meilensteinen wird überprüft und am Ende jeder Phase aufgrund neuer Erkenntnisse durch den Auftraggeber entschieden, ob das Projekt weitergeführt wird.
- **Qualitätsprüfung:** Die Ergebnisse jeder Phase werden in den nachfolgenden Phasen nochmals verifiziert und nachvollzogen. Am Ende jeder Phase erfolgt eine Qualitätskontrolle.
- **Fehlerreduktion:** Der Systembau (Implementation) beginnt erst, nachdem umfassende Studien erfolgt sind. Damit sollen Fehler vor der Einführung erkannt und Kosten minimiert werden.
- **Permanenter Aufbau:** schrittweise Verbesserung des Modells
- **Problemlösungszyklus:** In jeder Phase gelangt der Problemlösungszyklus zur Anwendung.

Im Wasserfallmodell kommen unter Umständen einige Aspekte zu kurz. Wir werden weiter unten nochmals ausführlich auf diese Nachteile und mögliche Gegenmassnahmen zu sprechen kommen. Zu kurz kommen im Wasserfallmodell namentlich:

- Gewinnung von Wissen beim Bearbeiten realer Aufgaben.
- Bestandteil eines Organisationsentwicklungsprozesses, da keine evolutionäre Entwicklung des Systems erfolgt.

### 8.4.3 Phasenweise Erweiterung der Betrachtungstiefe

Im ersten Teil wurde ausgeführt, dass sich Systeme sowie Aspektsysteme (Prozesse, Funktionen etc.) hierarchisch dekomponieren lassen. Der Grundsatz des Wasserfallmodells “vom Groben zum Detail” bedeutet nichts anderes, als dass die Systemhierarchie von Phase zu Phase stufenweise tiefer analysiert und gestaltet wird (s. Abb. 97). Der Grundsatz der Vollständigkeit bedeutet, dass eine Hierarchieebene in jeder Phase in der vollen Breite betrachtet wird; das Konzept deckt z.B. alle Prozesse ab und im Detailkonzept werden alle vorhandenen Prozesse nochmals vertieft analysiert. Es erfolgt keine Fokussierung auf das Wesentliche.

Dem Modell liegt der Hauptgedanke zugrunde, dass Vor-, Haupt- und Detailstudien, welche sich über das gesamte System bzw. den gesamten Eingriffsbereich in der vollen Breite erstrecken, mögliche konzeptionelle Fehler früh entdecken lassen und kostenmäßig geringe Auswirkungen auf das Gesamtprojekt haben. Daher sollte man idealerweise mit der Implementation möglichst lange abwarten und erst dann damit beginnen, wenn alles Konzeptionelle klar ist.

Umgekehrt würde sich, wird derselbe Fehler erst während der Einführungsphase bemerkt, dessen Beseitigung auf alle anderen Bereiche (Spezifikation, Programm, Handbücher, etc.) auswirken. Gemäß diesem Denkansatz steigt der Beseitigungsaufwand exponentiell mit dem Phasenfortschritt an.

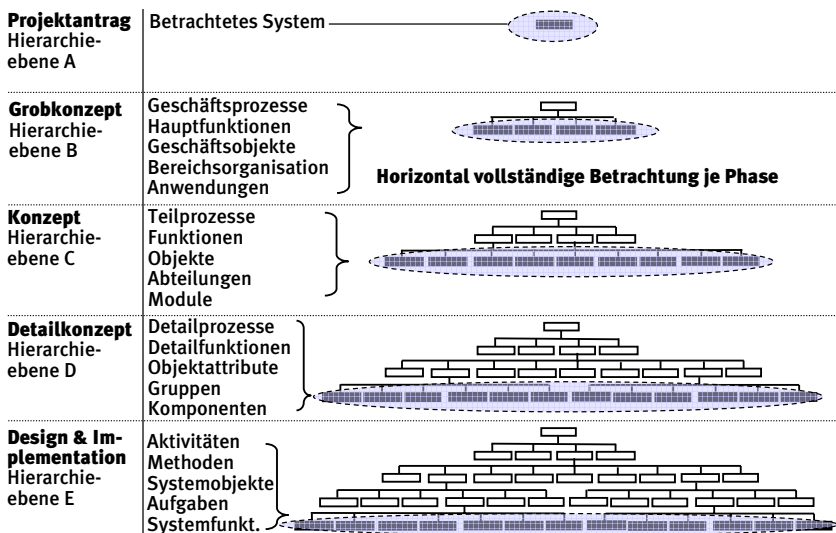


Abb. 97: Vertiefung der vollständigen “horizontalen” Betrachtung je Phase

### 8.4.4 Übersicht der Gestaltungstätigkeiten

Nachfolgend wird eine kurze Übersicht der wichtigsten Tätigkeiten und Ergebnisse je Phase des Wasserfallmodells gegeben. Dieselben Gestaltungstätigkeiten werden uns später in den weiteren Vorgehensmodellen ebenfalls begegnen, allerdings werden dort mehrere Tätigkeiten in einer Phase zusammengefasst sein.

#### Projektantrag

Der Projektantrag eines Informatikvorhabens prüft, welches Ziel das Projekt verfolgt, welche Anspruchsgruppen betroffen sind und ob tatsächlich das richtige Problem angegangen wird. Dies wird in einem Projektantrag und einer Wirtschaftlichkeitsbetrachtung (engl.: "Business Case") dokumentiert.

#### Grobkonzept

Im Rahmen des Grobkonzeptes wird ein erstes grobes Lösungskonzept erstellt, grundlegende sozio-technische Varianten der Systemkonzeption werden analysiert, konzipiert und bewertet.

Die Analysetiefe im Grobkonzept ist auf die oberste Analysestufe beschränkt (s. Abb. 98). Natürlich wird sich die Betrachtungstiefe nicht immer streng formal an diese Systematik halten bzw. die Betrachtung kann sich im Detaillierungsgrad stellenweise bereits auf Konzeptstufe bewegen.

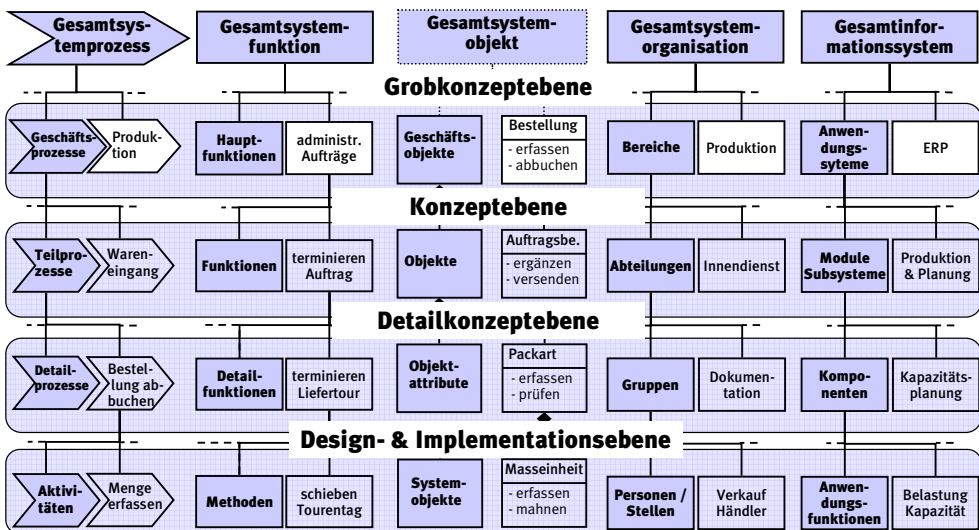


Abb. 98: Analyse- und Gestaltungsebenen

**Konzept**

Nach der Wahl eines Lösungsprinzips wird im Sinne einer Hauptstudie ein umfassendes Konzept erstellt, welches die Prozesse, Funktionen, Objekte und Aufgaben und die Systemarchitektur beschreibt. Man wird die Anforderungen möglichst logisch, d.h. unabhängig von einer gegebenen Softwarelösung, beschreiben.

**Pflichtenheft und Evaluation**

Die meisten Softwareprojekte werden unter Einbeziehung von externen Lieferanten durchgeführt. In diesen Fällen muss in einem Pflichtenheft klar spezifiziert werden, welche Leistungen vom Lieferanten zu welchen Kosten erbracht werden sollen.

**Verifikation und Vertrag**

Die Verifikations- und Vertragsphase schließt an die Evaluationsphase an. Zunächst wird versucht, gewisse in der Evaluation vielleicht noch offene Punkte vor dem Eintreten in die Vertragsphase zu klären. Die anschließende Ausarbeitung des Vertrages umfasst alle relevanten Aspekte des Angebots sowie die Modalitäten der Projektentwicklung. Bevor mit der Entwicklung des Projektes begonnen werden kann, müssen die Modalitäten der Projektentwicklung geklärt werden. So gilt es, die terminlichen, aufwandsbezogenen und sachlichen Aspekte detailliert zu planen und die entsprechenden Verantwortlichkeiten festzulegen.

**Detaillkonzept**

Das nachfolgende Detaillkonzept beinhaltet die Erstellung einer detaillierten Spezifikation aller Datenelemente und Funktionen auf einer detaillierten Ebene. Die Detailkonzeption erfolgt bezogen auf das gewählte technische System und hängt von dessen Beschränkungen ab. Hier gelangen die Teilprojektteams zum Einsatz, und das fachbereichsspezifische Wissen muss vom Kunden zum Anbieter transferiert werden.

**Design und Implementation**

In der Design- und Implementationsphase wird die eigentliche Konfiguration von Standardsoftwarekomponenten oder allenfalls die notwendige Anpassung bzw. die Programmierung vorgenommen. Dies beinhaltet neben der Codierung die Integration der einzelnen Module und den Test des Systems.

**Abnahme und Inbetriebnahme**

Spätestens in der Inbetriebnahmephase müssen die Mitarbeiter geschult werden. Sodann werden die echten Daten von etwa abgelösten Systemen übernommen und es erfolgt der Produktivstart.



### 8.4.5 Vor- und Nachteile des Wasserfallmodells

Die Problematik des Wasserfallmodells liegt darin, dass die Anforderungsanalyse und das Design sehr lange nur auf dem Papier bestehen. Es hängt damit wesentlich von der Vorstellungskraft der zukünftigen Benutzer ab, wie gut diese Informationen geistig in ein reales Produkt umgesetzt werden können.

Besonders offensichtlich wird dieses Problem in den realisierungsnahen Phasen, anlässlich welcher beispielsweise Datenbankstrukturen festgelegt werden. Benutzer verstehen die Beschreibungssprache für Datenbanken oft nur punktuell. Es kann deshalb sein, dass die Entwicklung eines Prototyps günstiger ist als die Behebung von Fehlern und Missverständnissen im Wasserfallmodell.

Außerdem können sich die Ziele und Anforderungen im Verlauf des Projektes ändern. Dementsprechend ist am Wasserfallmodell vermehrt Kritik geübt worden: "In der Vergangenheit hat es sich aber mehrheitlich gezeigt, dass Software Engineering nicht mit der genau gleichen Strenge der Lebensphasen, besonders während der Konzeption und Systemerarbeitung, betrieben werden kann, wie dies die Theorie des Systems Engineering gerne vorschreibt" (Schönsleben 2001).

Am Zentrum für Unternehmenswissenschaften der ETH (BWI) wurde der Software-Entwicklungsprozess einer Bank analysiert. Zunächst wurden im Rahmen einer "Kritikphase" die Hauptschwächen des bestehenden Prozesses aufgenommen. Es ergab sich, dass sich die Methodik dieser Bank weitgehend am Wasserfallmodell orientierte. Die in die Projekte involvierten Mitarbeiter schilderten folgende Hauptprobleme:

- Schlechte Zusammenarbeit der Informatik- und Fachabteilungen
- Es fehlt an einer gemeinsamen Sprache
- Es fehlt an einer gemeinsamen und durchgängigen Modellierungsmethode
- Der Know-how-Verlust an den Phasengrenzen ist hoch

Wo können nun die Gründe hierfür liegen?

#### 8.4.6 “Team Breaks” durch funktionale Arbeitsteilung

Ein Hauptgrund liegt im sequenziellen Durchschreiten des Prozesses durch eine funktional arbeitsteilige Organisation. Man vergegenwärtige sich folgende beispielhafte Situation (s. Abb. 99):

Die Fachabteilung erstellt ein Grobkonzept, was vielleicht sechs Monate dauert. Anschließend beginnt die Organisationsabteilung mit einem Konzept – dies dauert vielleicht nochmals sechs Monate. Dann gibt es eine Unterbrechung von vier Monaten, bis das weitere Geld bewilligt ist. Nach eineinhalb Jahren kommt erstmals die Informatikabteilung mit ihrem Detailkonzept zum Einsatz. Sie beginnt nochmals von vorn, da die bestehenden Dokumente vermutlich zu lückenhaft und nicht mehr aktuell sind. Ist dieses Dokument fertig, wird es aber von Seiten der Fachabteilung kaum zu verstehen sein. Die Programmierer beginnen trotzdem mit der Arbeit. Es sind nun bereits zwei Jahre verstrichen. In diesen zwei Jahren wurden drei Konzepte geschrieben, es waren fünf unterschiedliche Teams verantwortlich, wovon allerdings jedes eine andere Sprache spricht. Fazit: Die Vorstellungskraft der Endbenutzer reichte nicht aus, das entstehende Produkt voraus zu denken. Diese Beschreibung gibt eine Idee, weshalb Software-Projekte häufig scheitern.

Dieser Fokuswandel und ständig ändernde Teams verhindert eine effiziente Kommunikation zwischen den Beteiligten. Man spricht an dieser Stelle auch von “Team Breaks”.

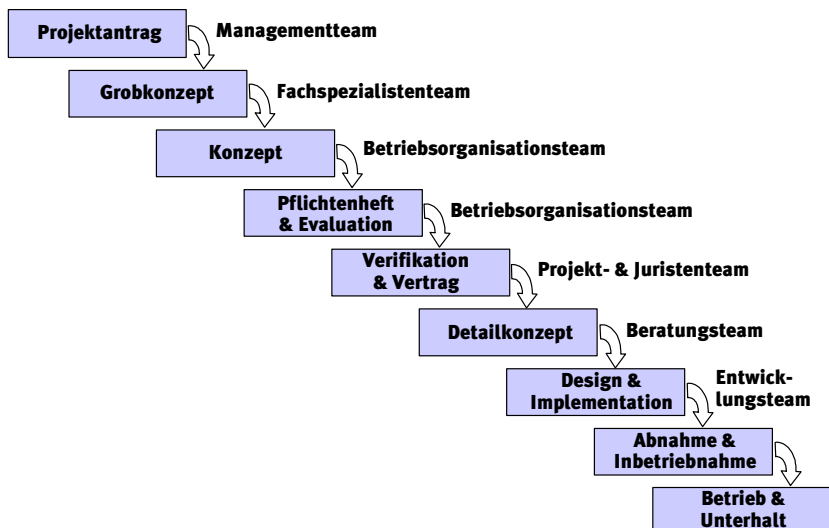
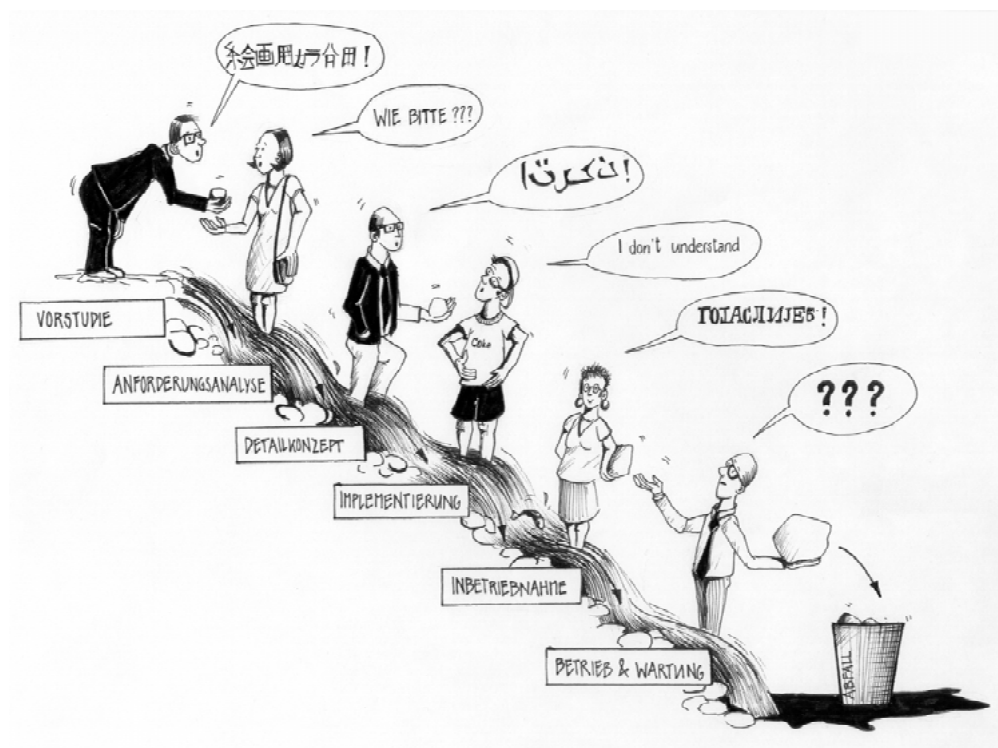


Abb. 99: Funktionale Arbeitsteilung verursacht “Team Breaks”



### 8.4.7 “Model Breaks” im Wasserfallmodell

Infolge der sich ändernden Teams verwendet jedes Team auch eigene Modellierungstechniken. Es lässt sich die These vertreten, dass jede funktionale Phase im Wasserfallmodell schwerpunktmäßig eine eigene Sichtweise vertritt (s. Abb. 100). Dadurch entstehen neben den beschriebenen Team Breaks auch so genannte “Model Breaks”, d.h. Brüche in den Modellierungstechniken und -sichtweisen. In gewissem Sinne lässt sich dies auch mit den unterschiedlichen Sichten und einer funktionalen Arbeitsteilung der Projektbeteiligten in Beziehung setzen.

Zunächst wird im Rahmen des Grobkonzeptes ein grobes sozio-technisches Lösungskonzept erstellt – dieses berücksichtigt die Aufgaben des Unternehmens. In einem zweiten Schritt werden die Prozesse spezifiziert – zumindest in jüngerer Zeit wurde diese Tätigkeit von Beratern häufig prozessorientiert durchgeführt.

Das Detailkonzept analysiert die tatsächlichen Prozesse in einem größeren Detaillierungsgrad – mit anderen Worten: Es werden die benötigten Funktionen explizit aufgenommen. Wenn die (Anpassungs-) Programmierung objektorientiert erfolgt, dann widmet sich diese Phase primär der objektorientierten Sichtweise. Die Inbetriebnahme und die Schulung müssen in einem nächsten Schritt auf die tatsächlichen Stellen und Arbeitsplätze hin ausgerichtet erfolgen. Erst im Betrieb kommt das System letztlich mit all seinen Aspekten zum Tragen.

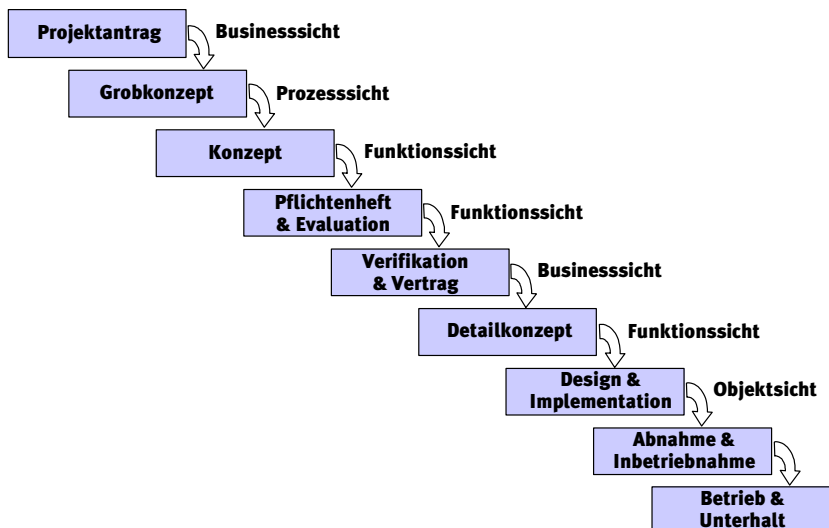


Abb. 100: Primäre Sichtweisen und “Model Breaks”

## 8.5 Zyklisch-iteratives Vorgehensmodell: Spiralmodell

In Abgrenzung zum Wasserfallmodell gibt es ein zweites, diametral unterschiedliches Basis-Vorgehensmodell: das zyklisch-iterative Spiralmodell. Das Wasserfallmodell geht davon aus, dass in jeder Phase nur eine Tätigkeit ausgeführt wird (z.B. Grobkonzeptionstätigkeit, Implementation), diese dafür möglichst umfassend und vollständig. Ein benutzbares System steht erst am Ende des Projektes bereit – dafür mit voller Funktionalität.

Im Gegensatz hierzu postuliert das Spiralmodell die Integration aller Tätigkeiten in jeder Projektphase, diesmal in zyklischer inhaltlicher Erweiterung der Ergebnisse aus früheren Phasen. Nach jeder Phase steht ein benutzbares System bereit – die Funktionalität wird von Phase zu Phase erweitert. Dies entspricht einer “Inkremental-Entwicklung” und erzeugt eine Folge von sukzessiv erweiterten Systemen (Boehm 1986).

Der wesentliche Unterschied zum Wasserfall liegt darin, dass in jeder Phase “alle” Tätigkeiten: Grobkonzept, Konzept, etc. wiederkehrend wahrgenommen werden (s. Abb. 101). Diese Tätigkeiten stellen in diesem Modell also keine eigene Phase dar, sondern Aktivitäten innerhalb der Phase! Das Spiralmodell unterscheidet sich damit vom Wasserfallmodell in der Zuordnung der Tätigkeiten zu den Phasen, nicht aber in der Art und dem Inhalt der wahrgenommenen Tätigkeiten. Die Tätigkeiten werden im Spiralmodell nicht mit derselben Breite wie im Wasserfallmodell durchgeführt, sondern je Phase erweitert (s. nächster Abschnitt).

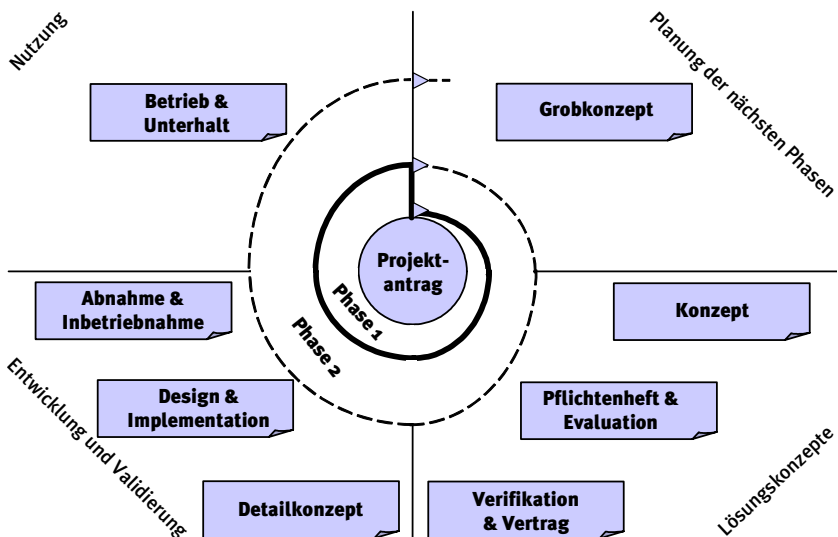


Abb. 101: Zyklische Integration aller Gestaltungstätigkeiten in jeder Phase

### 8.5.1 Vergleich der Projektphasen und Terminpläne

Einen guten Eindruck der Unterschiede zwischen den Vorgehensmodellen bietet der Vergleich der Terminpläne (s. Abb. 122). Es wird sichtbar, dass im Wasserfallmodell je Phase nur eine Tätigkeit durchgeführt wird und die Phasenbezeichnung diese Tätigkeit ausdrückt.

Das Spiralmodell integriert die Tätigkeiten und die Phase bezeichnet das zu erzielende Resultat, welches in jedem Zyklus den Benutzern übergeben wird. Durch das zyklische Durchlaufen der Systemgestaltungstätigkeiten ergibt sich im Spiralmodell eine Verringerung des Umfanges einer Tätigkeit je Phase. Hingegen entwickelt sich jede Tätigkeit kontinuierlich über mehrere Zyklen weiter.

Würde also z.B. ein Internetbrowser entwickelt, dann könnte in einer ersten Phase ein erster Browser programmiert werden und dieser in den nächsten Phasen um E-Mail und weitere wesentliche Funktionen erweitert werden.

Im Wasserfallmodell müssten alle für das abgegrenzte System wesentlichen Funktionen von Beginn weg konzipiert und bereits in einem ersten Release ausgeliefert werden.

Dieses Vorgehen lässt sich auf Standard- als auch auf Individualsoftware-Projekte anwenden. Interessant ist die Fragestellung, was es bedeutet, die Vertragsphase oder die Evaluationsphase mehrfach zu durchlaufen.

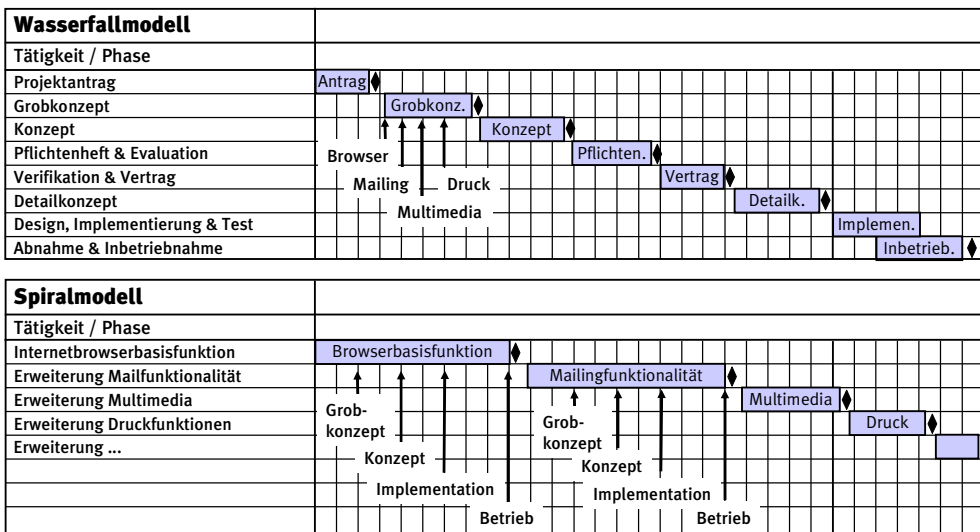


Abb. 102: Vergleich der Projektterminpläne von Wasserfall- und Spiralmodell

### 8.5.2 Vertikale Systembetrachtung im Spiralmodell

Während das Wasserfallmodell unter dem Motto “vom Groben zum Detail” steht, kann das Spiralmodell durch den Leitsatz “vom Wichtigen zum weniger Wichtigen” charakterisiert werden.

Die unten stehende Abbildung zeigt, dass im Rahmen des Spiralmodells in jeder Phase alle Gestaltungstätigkeiten integriert werden und bereits in der ersten Phase auch eine sehr detaillierte Betrachtung in einem ausgewählten Bereich auf unterster Systemebene erfolgt (s. Abb. 103). Dieser Bereich wird in den folgenden Phasen sukzessive erweitert bzw. ausgedehnt.

Ein Hauptvorteil dieses Vorgehens liegt in der raschen Benutzung und den verlässlichen Beurteilungen durch die Benutzer. Für die Endbenutzer ist dies anhand eines greifbaren Systems möglich. Modelle auf Papier stellen keine vollständige Integration dar.

Die zentrale Forderung der IT-Projektentwicklungsmethodik liegt somit darin, die verschiedenen Aspektsysteme (Prozesse, Objekte, Funktionen, Aufgaben und Technik) durch gezielte Maßnahmen in ihrer Integration zu fördern und das System möglichst rasch als Ganzes erfahrbar zu machen.

Das Spiralmodell läuft auf eine stärkere Partizipation der Benutzer hinaus, insofern die Benutzer viel schneller mit dem System konfrontiert sind und die Entwicklung nach der ersten Nutzungsphase maßgeblich bestimmen.

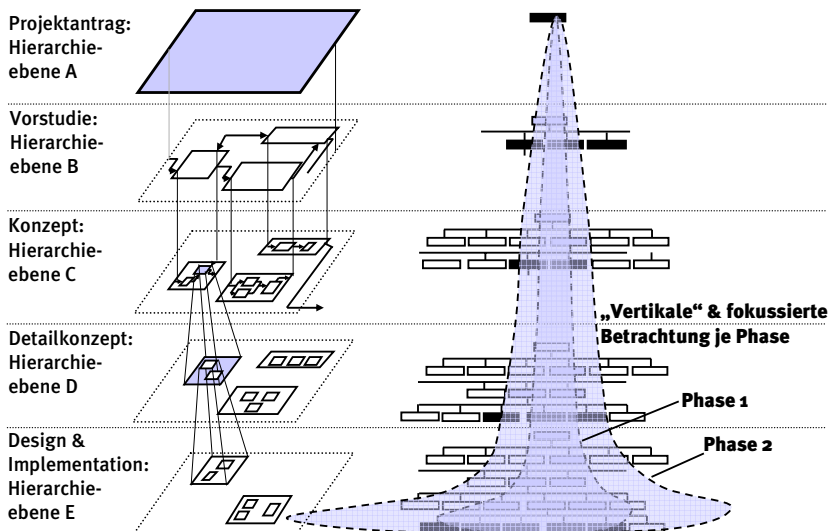


Abb. 103: Ausweitung der “vertikalen” Betrachtung je Zyklus im Spiralmodell

### 8.5.3 Einbezug der Endbenutzer mit partizipativem Ansatz

Gemäß Erkenntnissen aus der Software Ergonomie sind unter anderem folgenden zusätzlichen Vorteile bei partizipativen Ansätzen, wie es das Spiralmodell darstellt, zu erwarten (Rauterberg et al. 1994):

- Bessere Aufgabenangemessenheit der Software
- Höhere Akzeptanz, Motivation, Identifikation und Einsicht
- Geringere Kosten infolge Korrekturen
- Feedback an den Entwickler.

Von besonderer Bedeutung ist die zyklische Optimierung des Softwareproduktes, wie dies im Spiralmodell postuliert wird. Das Spiralmodell geht ja davon aus, dass mit jedem Inkrementalfortschritt eine Testphase durch den Benutzer erfolgt und anschließend neue Anforderungen an das System durch die Benutzer formuliert werden. Damit ist beim Spiralmodell die Einbeziehung des Benutzers automatisch gegeben. So haben Gould & Lewis (1985) drei Prinzipien für die Gestaltung von Benutzungsoberflächen vorgeschlagen: Früher Fokus auf den Benutzer und dessen Tätigkeit, empirische Überprüfung und iteratives Design.<sup>5</sup>

Es existiert eine Vielzahl von Publikationen bezüglich benutzerorientierter bzw. partizipativer Methoden ((Mumford & Weir 1979, Floyd 1984, Norman 1986, Greenbaum & Kyng 1991, Rauterberg et al. 1994, Wehner et al. 1996)).

Obwohl viele wissenschaftliche Arbeiten im Bereich der Softwareergonomie durchgeführt wurden und werden, gibt es nur wenige, die kritische Erfolgsfaktoren in der Praxis erhoben haben. In einer diesbezüglichen Studie, welche 14 erfolgreiche und 14 nicht erfolgreiche Softwareentwicklungsprojekte hinsichtlich der Einbeziehung von Benutzern untersucht hat, stellte sich eine klare Korrelation zur Stärke der Benutzereinbeziehung heraus (Keil & Carmel 1995). Insbesondere konnte aufgezeigt werden, dass mit steigender Anzahl von so genannten "customer developer links" die Projekterfolgchancen steigen.

Als "customer developer links" werden unter anderem erwähnt: "user group", "requirements prototyping", "user interface prototyping", "interviews", "support line", "testing", etc.

Der Trend zum Spiralmodell hat sich nicht zuletzt auch aufgrund gewisser Fortschritte in der Programmiertechnik entwickelt.

---

<sup>5</sup> s. auch Bellotti (1988).



### 8.5.4 Vor- und Nachteile des Spiralmodells

#### Vorteile

Der Hauptvorteil des Spiralmodells liegt darin, dass dieses Vorgehen verlässliche und frühe Beurteilungen durch die effektiven Benutzer ermöglicht. Außerdem müssen wichtige Funktionalitäten zwangsläufig berücksichtigt werden. Dies zwingt die Entwickler ebenfalls dazu, sich auf die für den Kunden nutzbringenden Funktionalitäten zu konzentrieren und nicht das technisch Mögliche auszuschöpfen. Da im Rahmen des Spiralmodells in jeder Phase alle Gestaltungstätigkeiten integriert werden müssen, sind automatisch auch integrierte Teams mit vielfältigen Fähigkeitsprofilen am Werk. Teambreaks und Model Breaks können so nicht entstehen. Dies lässt sich gut mit der Theorie der Aspektsysteme erklären (s. Teil 1). Es ist daher anzustreben, die verschiedenen Sichten möglichst rasch mental zu integrieren.

Es vergehen zudem nicht "Jahre", in denen nur Papier produziert wird, welches zudem fehlinterpretiert werden kann. Missverständnisse zwischen den Beteiligten werden sich am konkreten Produkt rasch zeigen. Dies fördert die gemeinsame Sprache. Außerdem können sich die Ziele und Anforderungen im Verlauf des Projektes ändern.

#### Nachteile

Das Spiralmodell postuliert, dass möglichst bald mit der Implementation wichtigster Funktionen begonnen werden soll. Dies stellt natürlich auch eine große Gefahr dar, insofern unter Umständen die technische Lösung in eine ganz falsche Richtung verlaufen kann. So wird unter Umständen aufgrund der eingeschränkten Betrachtung fälschlicherweise postuliert, eine einfache Individualprogrammierung reiche aus. Mit der Zeit wächst das System mehr und mehr und man stellt fest, dass bei umfassenden Vorabklärungen klar gewesen wäre, dass der Lösungsumfang eine Eigenentwicklung übersteigt.

Das Spiralmodell kann unter Umständen auch als "Alibi" für eine rasche Technologieentscheidung herhalten, nach dem Motto: Wir beschaffen uns eine Entwicklungsumgebung oder ein Tool und dann schauen wir weiter. Auch das Spiralmodell erfordert die genaue Analyse und Erhebung von Benutzerbedürfnissen und ist in gewisser Weise nicht weniger anspruchsvoll als das Wasserfallmodell.

In vielen Projekten stellt man aber fest, dass der Kunde ja gerade für einen relativ großen Funktionsumfang bereits vor seiner Entscheidung für einen spezifischen Lieferanten ein Angebot haben möchte. Damit man verbindliche Angebote einholen kann, müssen bereits weite Teile in den Grundzügen spezifiziert sein.

## 8.6 Kombination aus sequenziellem und zyklischem Vorgehen

Die oben stehenden Ausführungen könnten nun den Eindruck erwecken, das Wasserfallmodell sei gemäß heutigem Stand der Erkenntnisse absolut und in jedem Falle zu vermeiden. Demgegenüber sei dem Spiralmodell immer der Vorzug zu geben. Tatsächlich darf die Sache nicht dermaßen polarisiert betrachtet werden. Wie so oft, wird auch hier die Wahrheit projektbezogen in der Mitte liegen.

Die beiden Modelle stellen exemplarische Extremsichten dar. Beim Wasserfallmodell geht man davon aus, dass das gesamte System mit allen seinen Details bereits vor dem Beginn der Implementation und Programmierung festgelegt sein muss. Viele gescheiterte Projekte zeugen von den Problemen dieses Ansatzes.

Beim Spiralmodell postuliert man den sofortigen Beginn der Implementation und Einführung bereits in der ersten Projektphase. Auch hier hat die Realität bereits zu oft gezeigt, dass ein solcher Anspruch nicht durchführbar ist oder aber auch bei diesem Vorgehen Überraschungen hinsichtlich der Dimensionierung eines Projektes auftreten können. Unter Umständen wird man daher häufig ein etwas umfangreicheres Grobkonzept im Sinne des Wasserfallmodells durchführen und erst danach zyklisch-iterativ vorgehen.

Demnach sind in den meisten Fällen weder das Spiral- noch das Wasserfallmodell in seiner abstrakten Reinheit durchführbar. Unter Umständen sind ja in gewissen Bereichen des Unternehmens schnell erste Resultate gefordert. Es gilt daher, die Vor- und Nachteile dieser Ansätze zu kennen und den geeigneten und situationsgerechten Mittelweg zu finden. Selbstverständlich liegt die Empfehlung nahe, möglichst bald iterativ zu entwickeln und daher alles in Gang zu setzen, abgrenzbare Projekteinheiten zu schaffen und diese kurzfristig und mit kleineren Geldbeträgen umzusetzen.

In den nachfolgenden Kapiteln wird auf die Gestaltungstätigkeiten (Grobkonzept, Konzept, ...) detailliert eingegangen. Die Beschreibung dieser Gestaltungstätigkeiten nimmt dabei keine besondere Rücksicht darauf, ob die Phasen nun linear oder zyklisch iterativ durchlaufen werden. Wie gerade ausgeführt, ist diese Mischung ja auch situationsbezogen vorzunehmen. Wichtig ist zunächst, die Elemente und Inhalte einer bestimmten Tätigkeit kennen zu lernen und anschliessend in der projektspezifischen Breite und Tiefe auszuführen. Hinzu kommt, dass sich die Phasen des Vorgehensmodelles auch überlappen können. Gewisse Gestaltungstätigkeiten können u.U. und je nach Projekt auch weggelassen werden (z.B. Vertrag, Evaluation).

### 8.6.1 Beispiel: Rational Unified Process – RUP

Es haben sich im Verlauf der letzten Jahre diverse Vorgehensmodelle entwickelt. Sie entsprechen in jedem Fall einer Mischung zwischen Wasserfall- und Spiralmodell (Hermes, V-Modell, RUP, Extreme Programming, ...).

Die Firma Rational (eine IBM-Tochter) hat beispielsweise ihre UML-Sprache durch den Rational Unified Process (RUP) ergänzt. Sie hat sich das hohe Ziel gesetzt, den Systementwicklungsprozess zu vereinheitlichen. Jedes Projekt wird in vier Phasen aufgeteilt (Einstieg, Ausarbeitung, Konstruktion, Überleitung). Entsprechend dem Spiralmodell wird die Integration der Tätigkeiten je Phase postuliert (s. Abb. 104). Die Fläche zeigt die approximative Kapazitätsbelastung je Tätigkeit in den Phasen.

Gut ersichtlich ist der Versuch eines Mixes aus Wasserfall- und Spiralmodell: In den ersten beiden Phasen überwiegen die Analysetätigkeiten – der Einsatz bei den Benutzern wird nur zaghaf vorgenommen. In den weiteren Phasen erfolgt die vollständige Implementation.

Etwas problematisch erscheint bei diesem Vorgehensmodell einerseits die starre Vorgabe der postulierten vier Phasen. Praxisgerechter wäre vermutlich das Postulat eines situativen Mixes aus Wasserfall- und Spiralmodell. Viele Firmen möchten und können sich diesbezüglich nicht einschränken. Eine Einschränkung auf ein Phasenmodell trägt der Vielfalt möglicher Informatikprojekte zu wenig Rechnung und führt zu geringer Akzeptanz.

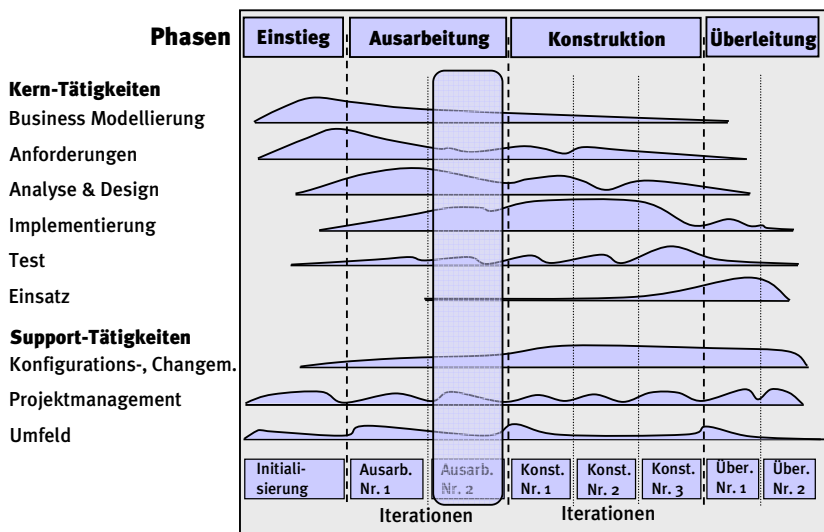
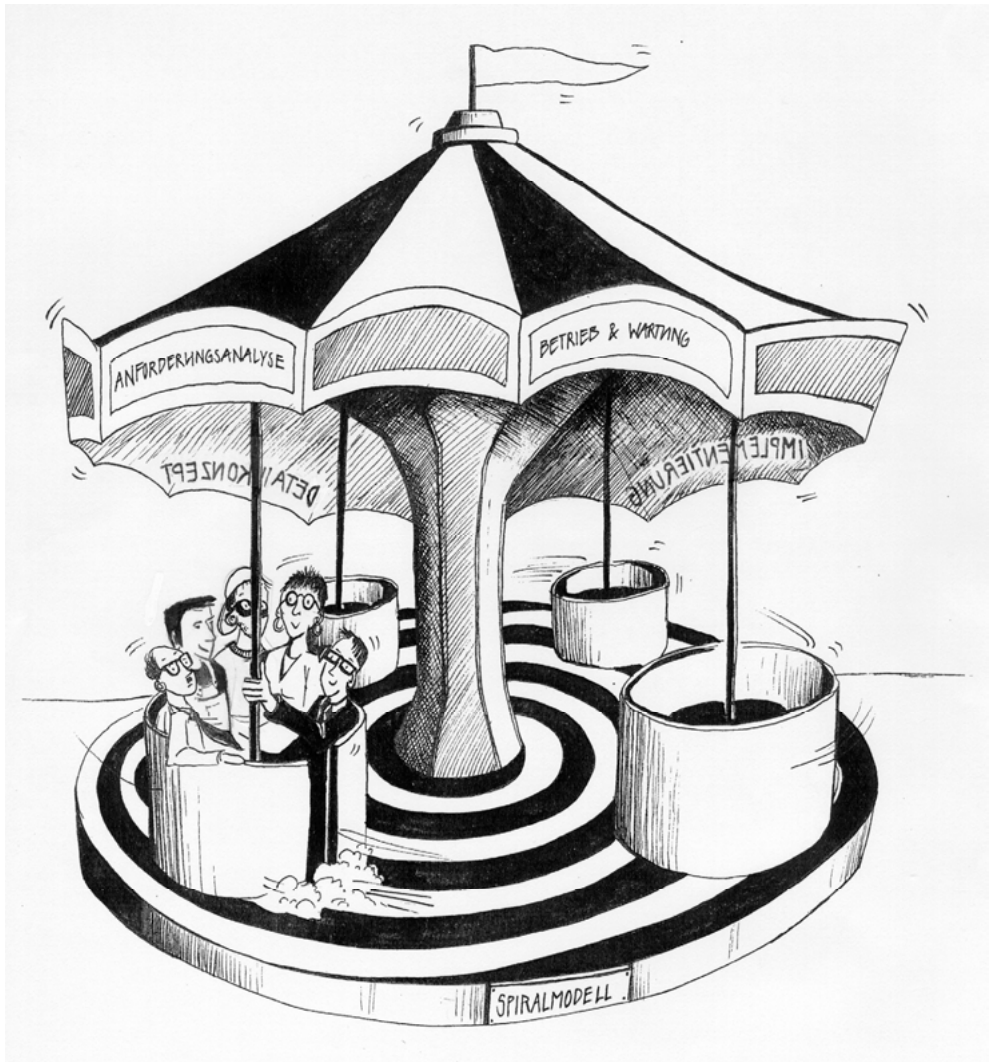


Abb. 104: Phasenmodell des Rational Unified Process (RUP)



## 8.7 Einsatz von Modellierungsmethoden

Wie in Teil 1 aufgezeigt, wurden in der Vergangenheit mehrere integrierte Methoden entwickelt: prozessorientierte Methoden (z.B. ARIS), funktionale Methoden (z.B. strukturierte Analyse), objektorientierte Methoden (z.B. UML / RUP) und aufgabenorientierte Methoden (z.B. MTO-Analyse).

Jede dieser Methoden kann zwar mit gewissem Recht von sich behaupten, die Prozess-, Funktions-, Objekt- und Aufgabenaspekte zu berücksichtigen. Trotzdem wird bei jeder der genannten Methoden der Hauptfokus auf eine Sicht gelegt. Dies hat häufig zu einem Methodenstreit geführt – Berater setzen auf Prozesse und die Informatiker auf Objekte. Bekannt ist dieses Phänomen auch von politischen Parteien, die genau einen Aspekt des komplexen Gesellschaftssystems in den Mittelpunkt rücken (z.B. Wirtschaft, Nationales, Ethik, Soziales, etc.).

Vor diesem Hintergrund muss bei der Projektabwicklung – egal ob Wasserfall- oder Spiralmodell – stets ein ausgewogener Methodenmix angestrebt werden, mit dem Ziel, die *Integration der verschiedenen Sichten durch entsprechende Darstellungen* zu fördern.

Ganz wichtig erscheint die Forderung, die verschiedenen Teilmodelle der Matrix einem stetigen Abgleichungsprozess zu unterziehen. Sinnbildlich könnte man sich vorstellen, dass die verschiedenen Projektbeteiligten “ihre Modelle” gegenseitig am “runden Tisch” vorstellen und abgleichen.

Wenn in den nachfolgenden Kapiteln Modellierungsbeispiele angeführt werden, so sollen diese Modelle keine starre Vorgehensmethodik vorgeben.

Der Schritt und die Inhalte z.B. eines Grobkonzeptes werden mehr oder weniger dieselben bleiben, auch wenn in den verwendeten Modellierungsmethoden eigene Kreativität angewandt wurde oder auf ein kommerziell positioniertes Methodenset wie ARIS oder RUP gesetzt wird.

Die Wahl des Methodensets spielt eine viel untergeordnetere Rolle, wenn ein klares Verständnis dafür vorhanden ist, was mit den Modellen machbar ist und wo möglicherweise Lücken bestehen. Die größte Gefahr besteht darin, dass man in einem frühen Stadium eines Projektes mit einer zu detaillierten Sichtweise operiert und insbesondere die organisationsrelevanten Punkte wie die möglichen Autonomiespielräume unterschiedlicher Organisationseinheiten vergisst.

Auf die Akzeptanz eines Systems durch unterschiedliche Stellen und Organisationseinheiten wird es im Projekt ankommen.

## 8.8 Der wiederkehrende Problemlösungszyklus

Der Problemlösungszyklus ist ein Vorgehen, welches ausgehend von einem Problem durch klar strukturierte Schritte zur optimalen Lösung führen soll. Innerhalb jeder Phase der Systemgestaltung kommt der *Problemlösungszyklus* einmal, im Spiralmodell auch mehrfach wiederkehrend zur Anwendung. Im Mittelpunkt steht der Gedanke, dass die Lösungsfindung niemals aus “dem hohlen Bauch” heraus erfolgen darf. Es wird gefordert, in jeder Phase zunächst mit einer Situationsanalyse zu beginnen, um ein Problemverständnis zu erarbeiten und darauf aufbauend messbare Ziele festzulegen. Erst dann darf mit der “Synthese” von Konzepten begonnen werden. In der Konzeptanalyse werden die Mängel aufgedeckt und zyklisch verbessert. Es wird nicht vorschnell eine Lösung favorisiert, sondern es werden Varianten geprüft. Modellierungstechniken kommen doppelt zur Anwendung. In der Situationsanalyse dienen sie als *Abbild* der Ist-Situation, in der Konzeption dienen Soll-Modelle als *Lösungskonzept*. Die formelle Auswahl der besten Lösungsvariante schließt den Prozess ab. Das Vorgehen wird in den folgenden Kapiteln klar werden, da diese stets entsprechend diesem Zyklus aufgebaut sein werden (s. Abb. 105).

Man könnte argumentieren, dieses Vorgehen folge lediglich dem gesunden Menschenverstand – dies trifft auch zu. Andererseits lässt sich beobachten, dass in der Praxis oft Lösungen erarbeitet werden, ohne dass die Situation sauber geklärt ist oder Zielsetzungen explizit feststehen.

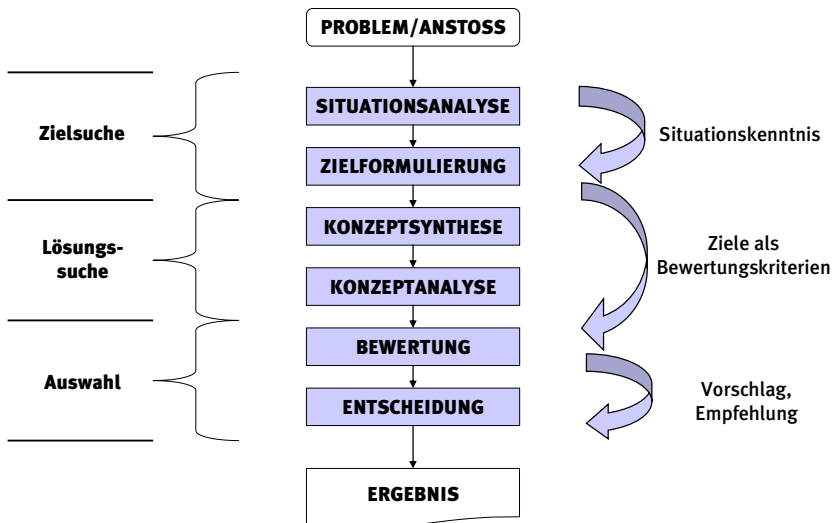


Abb. 105: Problemlösungszyklus des Systems Engineering

### 8.8.1 Situationsanalyse

**Zweck:** Der erste Schritt des Problemlösungszyklus beinhaltet eine saubere Analyse der Situation mit dem Zweck, den Eingriffsbereich abzustecken, die Kenntnisse zu vertiefen und eine Informationsbasis zu schaffen.

**Vorgehen:** Ein gutes Problemverständnis erfordert neben der exakten Analyse der Aufgabe zunächst eine genaue Beschreibung des Systems. Im Rahmen der Situationsanalyse wird daher eine *Systemabgrenzung* vorgenommen und der Eingriffsbereich festgelegt. Danach wird der Ist-Zustand aufgenommen. Es ist nicht sinnvoll, im Rahmen der Situationsanalyse des Grobkonzepts bereits alle erdenklichen Daten für die spätere Konzeptphase zu erheben. Der Detaillierungsgrad der Situationsanalyse richtet sich daher nach der jeweiligen Lebensphase. Zu groß ist die Gefahr, unnötige Erhebungen durchzuführen. Eine rationelle Informationsbeschaffung verlangt eine vorausschauende Abklärung. Dabei werden auch die *Stärken und Schwächen* ermittelt und deren Ursachen ergründet. Im Rahmen der Situationsanalyse sollte weiter eine Zukunftsanalyse durchgeführt werden, welche zusätzliche Erkenntnisse über die Entwicklungen des Umfelds ergibt (s. Abb. 106).

**Techniken:** Als Techniken zur Informationsbeschaffung und -aufbereitung können verschiedene Befragungs- und Beobachtungstechniken und Auswertungen bestehender Unterlagen zum Einsatz kommen. Die Darstellung erfolgt mit Hilfe der Modellierungstechniken.

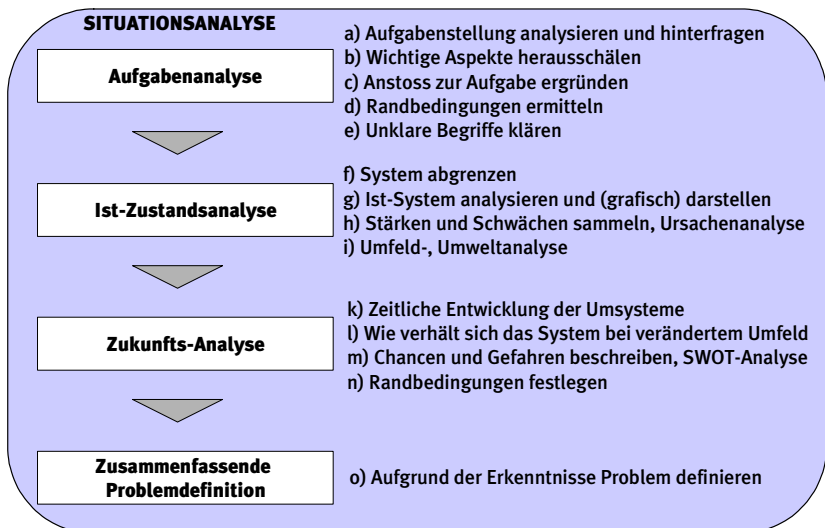


Abb. 106: Schritte der Situationsanalyse





### 8.8.2 Zielformulierung

**Zweck:** Die Zielformulierung hat den Zweck, Zielvorstellungen, die sich teilweise auch schon in der Situationsanalyse ergeben haben, zu bereinigen, systematisch zu strukturieren, auf Vollständigkeit zu überprüfen, zu ergänzen und schließlich in verbindlicher Form festzuhalten. Eine Lösungssuche ist erst sinnvoll, wenn man wirklich weiß, was man will.

**Vorgehen:** Die Zielformulierung beginnt mit dem Sammeln von Zielideen, der Gruppierung und Vollständigkeitsprüfung. In einem nächsten Schritt werden Unterziele abgeleitet, die relative Bedeutung festgelegt und die Ziele genehmigt. Folgende Prinzipien gelten für die Zielformulierung:

- Lösungsneutralität (Ziele lösungsneutral formulieren)
- Trennung von Muss- und Optimierungszielen
- Vollständigkeit der Ziele (Ergänzung von Zielen)
- Operationalität (Ziele sollten mit Eigenschaften messbar sein)

**Techniken:** Die wichtigste Technik zur Strukturierung der Ziele ist der Zielbaum, der die Gesamtheit der Ziele in eine hierarchische Form bringt (Haberfellner 2002) (s. Abb. 108). Klar zu unterscheiden sind System- und Vorgehensziele. Systemziele beziehen sich auf die Anforderungen des zukünftigen Systems ("Möglichkeit von Massenmailings"), Vorgehensziele beziehen sich auf das Projekt (System in 3 Monaten fertig gestellt).

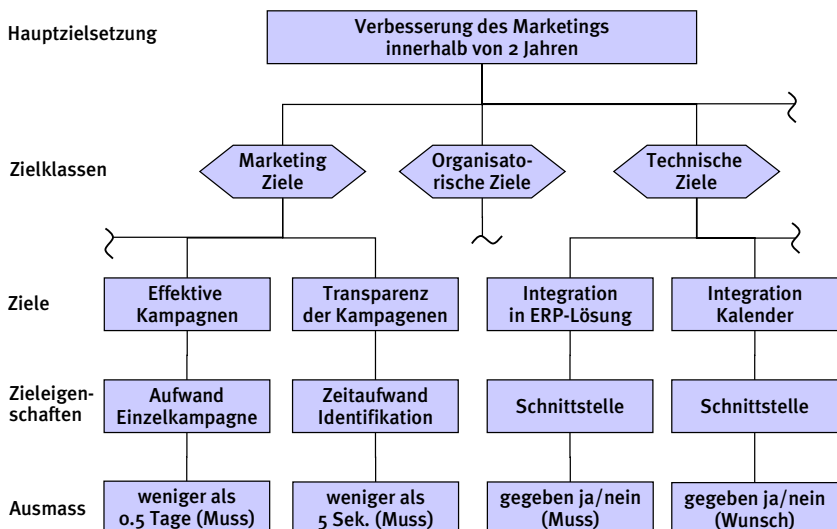


Abb. 108: Zielbaum

### 8.8.3 Lösungssuche: Konzeptsynthese & -analyse

**Zweck:** Der Zweck der Lösungssuche besteht darin, die bestgeeigneten Lösungen zu finden. Dies bedeutet meist, dass man sich nicht mit der erstbesten Lösung zufrieden gibt, sondern auf kreative Art und Weise ein Spektrum von ausreichend dokumentierten Lösungen erarbeitet. Es bleibt die Aufgabe der Bewertung, die beste Lösung auszuwählen.

**Vorgehen:** Die Lösungssuche stellt einen Prozess dar, welcher sich ebenfalls nach dem Motto "vom Groben zum Detail" gestaltet. Man wird also zunächst mit Grobvarianten beginnen und dann die Lösungskonzepte fortlaufend konkreter gestalten. Im Rahmen der Konzeptsynthese sollen verschiedenartige Lösungstypen erarbeitet werden. Ein Ansatz besteht darin, je Hauptziel eine bestgeeignete Lösung zu konstruieren. In der Konzeptanalyse werden Lösungen ausgeschieden, welche eindeutig schlecht sind bzw. Mussziele nicht erfüllen. In diesem Wechselspiel werden Lösungen fortlaufend konkreter dargestellt, analysiert und verbessert (s. Abb. 109).

**Techniken:** Man unterscheidet systematische und kreative Techniken zur Lösungssuche. Als systematische Kreativitätstechnik werden wir weiter unten die Methode der Morphologie kennen lernen. Systematische Techniken bedienen sich meist der Technik des Analogieschlusses.

Als kreative Techniken sind zu nennen: Brainstorming, Brainwriting etc. Im nächsten Abschnitt wird die Methode der Zukunftswerkstatt beschrieben.

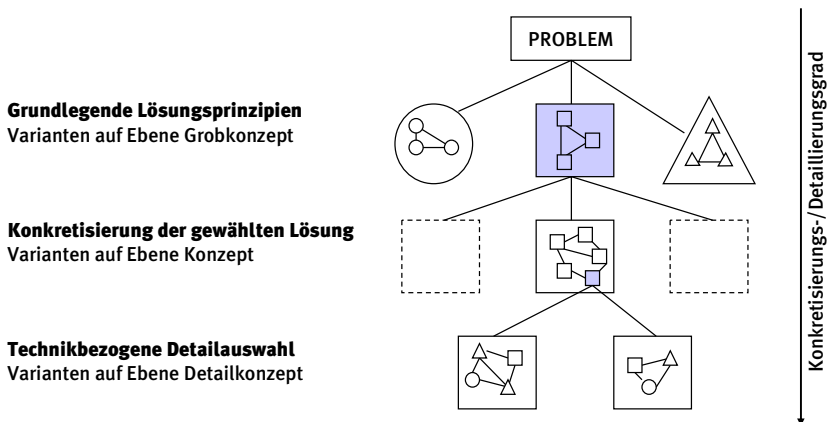


Abb. 109: Stufenweise Konkretisierung der Lösung: "Vom Groben zum Detail"

### Lösungssuche mit Hilfe von Kreativitätstechniken

Eine besonders geeignete Workshop-Technik auch für Informatikprojekte, weil partizipativ und gleichermaßen effizient, stellt die so genannte Zukunftswerkstatt dar (Jungk 1994; Kuhnt et. al. 2000). Die Werkstattarbeit spielt sich in drei aufeinander folgenden Phasen ab: der Kritikphase, der Phantasie- sowie der Verwirklichungsphase (s. Abb. 110).

*Kritikphase:* Es wird Kritik am System geäußert und in Stichworten auf einen großen Papierbogen geschrieben (Metaplantechnik). Im Anschluss an die Auflistung erfolgt die Auswahl der wichtigsten Punkte.

*Phantasiephase:* In einer darauf folgenden Brainstormingphase werden innovative Lösungen, Vorschläge und Ideen zur Verbesserung des Systems gesammelt und es wird wiederum das Wichtigste ausgewählt.

*Verwirklichungsphase:* In der letzten Phase erfolgt eine kritische Prüfung der Phantasien. Es werden also die konkreten Realisierungsmöglichkeiten abgeschätzt und entsprechende Verwirklichungsschritte formuliert.

Überdies wird festgehalten, welche der erarbeiteten Vorschläge und Maßnahmen aufbauorganisatorischer, ablauforganisatorischer, informationstechnischer bzw. personeller Natur sind. Damit können beispielsweise verfehlte Ansprüche an die Technik bereits zu einem frühen Zeitpunkt relativiert oder anderweitige Maßnahmen ergriffen werden.

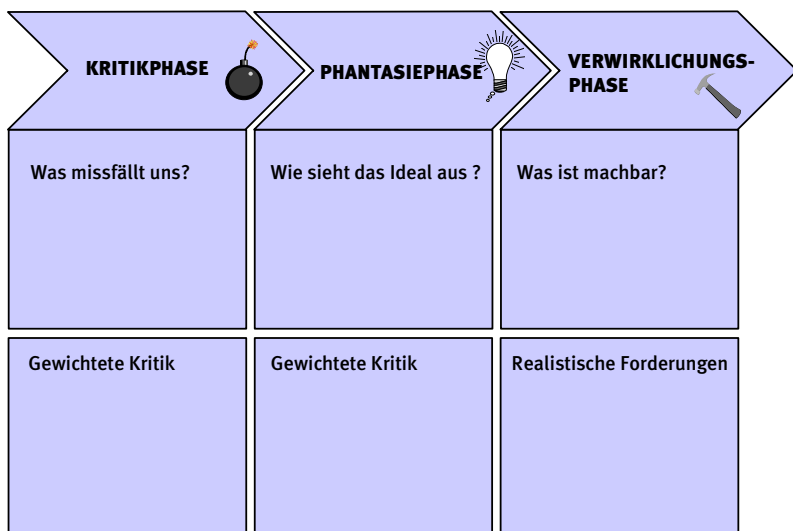


Abb. 110: Phasen einer Zukunftswerkstatt

### 8.8.4 Auswahl: Bewertung & Entscheidung

**Zweck:** Der Auswahlprozess setzt sich aus einem Bewertungs- und Entscheidungsschritt zusammen. Der Bewertungsschritt hat die Aufgabe, die Entscheidung zwischen mehreren machbaren Lösungsvarianten vorzubereiten. Er bezweckt also keine Überprüfung der Machbarkeit, denn dies war bereits Aufgabe der Konzeptanalyse. Die Unterscheidung von Bewertungs- und Entscheidungsschritt beabsichtigt, dem Entscheidungsgremium auf Basis einer guten und nachvollziehbaren Bewertung die abschließende Entscheidung zu überlassen. Es wird verhindert, dass die Entscheidung vorweggenommen und nur eine Variante vorgeschlagen wird.

**Vorgehen:** Mit diesem Vorgehen soll die Gefahr der Unbedachtheit, Voreingenommenheit oder Willkür reduziert werden. Eine Entscheidungssituation liegt vor, wenn mehrere gangbare Handlungsalternativen zur Verfügung stehen. Die Wahl des Entscheidungsverfahrens ist abhängig von der Bedeutung, dem Gremium und den Alternativen.

**Techniken:** Als Techniken zur Bewertung von Lösungen gelangen die Argumentenbilanz, Wirtschaftlichkeitsverfahren, Portfoliotechniken oder Nutzwertanalysen zu Anwendung. Alle Verfahren werden im zweiten Teil behandelt. Die unten dargestellte Nutzwertanalyse beruht auf der strukturierten Gewichtung und Punktevergabe aller Beurteilungskriterien zur Errechnung eines so genannten Nutzwertes jeder Lösung (s. Abb. 111).

	Gewicht in % G1, G2, G3			GA	Lösung A		Lösung B	
					Pkt.	Pkt.*GA	Pkt.	Pkt.*GA
<b>Funktionale Abdeckung</b>	<b>35</b>							
Kontakte & Termine	30	✓		5.25	4	21		
Adressen verwalten	50	✓		5.25	4	21		
...	50							
Kampagnen	30							
Marketingplan	25			2.6	5	13		
Zielgruppen festl.	50			5.2	6	31		
...	25			2.6	3	8		
Flexibilität	50			7	6	42		
...	50			7	7	49		
<b>Flexibilität, Konfigurat.</b>	<b>20</b>							
Modulararchitektur								
Mandantenfähig								
Subsystemfähig								
SW-Konfiguration								
Parametrisierbar								
Flexibilität								
<b>Technische Konzeption</b>	<b>20</b>	100		20	8	160		
<b>Einführung &amp; Vorgehen</b>	<b>5</b>	100		5	3	15		
<b>Anbieter</b>	<b>20</b>	100		20	6	120		
<b>Gesamtpunktzahl</b>	<b>100</b>			<b>100</b>		<b>574</b>		

Abb. 111: Nutzwertanalyse zur Bewertung

### 8.8.5 Wichtigste Ergebnisse je Tätigkeit in IT-Projekten

Die nachfolgende Darstellung zeigt einen Gesamtüberblick aller wichtigen Ergebnisse von IT-Projekten. Die Ergebnisse werden dabei in der Vertikalen schrittweise vom Projektantrag bis zum fertigen System konkretisiert.

Bereich Tätigkeit	Sozio-technisches System	Einführung & Ausbildung	Betrieb & Unterhalt	Projektmanagement & Controlling
<b>Projektantrag</b>	Problem- & Aufgabenanalyse; Projekt Mission Statement; Projektvarianten & Lösungsideen; Business Case & Projektantrag.	Projektmriss Einführung & Ausbildung.	Projektmriss Betrieb & Unterhalt.	Projektplanung; Projektauftrag; Projektsentscheidung.
<b>Grobkonzept</b>	Gesamtsystem- & Umfeldanalyse; Hauptzielsetzungen; Lösungsprinzipien, Grobkonzepte; Grobkonzeptbericht.	Grobkonzept Einführung & Ausbildung	Grobkonzept für Betrieb, Unterhalt & Sicherheit.	Projektstartplanung; Projektmanagement & Projektkontrolling; Grobkonzeptentscheid.
<b>Konzept</b>	Systemanalyse; Systemziele; Konzept, Systemspezifikationen; Konzeptbericht.	Einführungs- & Ausbildungskonzept; Einführungsprioritäten.	Betriebs- & Unterhalts- & Sicherheitskonzept.	Projekphasenplanung; Projektmanagement & Projektkontrolling; Konzeptentscheidung.
<b>Pflichtenheft &amp; Evaluation</b>	Pflichtenheft; Kriterienkatalog & -gewichtung; Lösungsanbieter-, ausschreibung; Evaluationsbericht.	Anforderungen Einführung & Ausbildung	Anforderungen Betrieb, Unterhalt & Sicherheit.	Projekphasenplanung; Projektmanagement & Projektkontrolling; Evaluationsentscheid.
<b>Verifikation &amp; Vertrag</b>	Verifikations- & Vertragsanalyse; Vertragsziele; Vertragsvarianten & -konzept; Software-Lizenzvertrag.	Projekt- und Dienstleistungsvertrag	Verifikationsprototyp; Wartungs-, Update- & Supportvertrag	Projekphasenplanung; Projektmanagement & Projektkontrolling; Vertragsentscheidung.
<b>Detailkonzept</b>	Detailanalysen; Detailziele; Detailkonzepte, -spezifikationen; Detailkonzeptbericht.	Detailkonzept Einführung & Ausbildung; Migrationsverfahren.	Prototypenbetrieb; Detailplanung Betrieb & Unterhalt; Projekthinfrastuktur.	Realisierungsplanung Projektmanagement & Projektkontrolling; Detailkonzeptentscheid.
<b>Design &amp; Implementation</b>	Design-, Implementationsanalyse; Design- & Implementationsziele; Design, Implementation & Test; Systemdokumentation.	Ausbildungsplanung; Einführungsplanung; Migrationsdesign; Migrationstests.	Handbücher Betrieb; Sicherheitsregelung; Betrieb Infrastruktur; Erfasste Stammdaten.	Projektphasenplanung; Projektmanagement & Projektkontrolling; Testabschlusentscheid.
<b>Abnahme &amp; Inbetriebnahme</b>	Bereitschaftsanalyse; Abnahmeziele & -konzepte; Ausbildung; Abnahmeprüfung; Produktivstart & Einführung.	Anwendungshandbuch; Organisationshandbuch; Ausbildung; Migration.	Betriebshandbuch; Supporthandbuch; Aktivierung & Betrieb; Mängelliste.	Projekphasenplanung; Projektmanagement & Projektkontrolling; Einführungsentscheid.

Abb. 112: Ergebnisübersicht je Gestaltungstätigkeit

## 8.9 Literatur zu den Vorgehensmodellen der Projektentwicklung

- Bainbridge, L., Sanderson, P. (1995): Verbal Protocol Analysis. In: Evaluation of Human Work: A Practical Ergonomics Methodology (Wilson, J.R., Corlett, E.N.; Hrsg.), 2<sup>nd</sup> edition. London: Taylor & Francis Ltd. 169–201.
- Bauknecht, K., Zehnder, C.A. (1997): Grundlagen für den Informatikeinsatz. Stuttgart: Teubner.
- Becker, M., Haberfellner, R., Liebetrau, G. (2000): EDV-Wissen für Anwender: ein Informatik-Handbuch für die Praxis (11. Aufl.). Zürich: Industrielle Organisation.
- Bellotti, V. (1988): Implications of Current Design Practice for the Use of HCI Techniques. In: People and Computers IV (Jones, D.M., Winder, R.; Hrsg.). Cambridge: Cambridge University Press. 13–34.
- Boehm, B.W. (1986): Wirtschaftliche Software-Produktion. Wiesbaden, Forkel.
- Chestnut, H. (1966): Systems Engineering Tools. New York: John Wiley & Sons.
- Cox, B., Novobilski, A. (1991): Object-oriented Programming: An Evolutionary Approach. Reading, MA: Addison-Wesley.
- Daenzer, W. (Hrsg.) (1976): Systems Engineering: Leitfaden zur methodischen Durchführung umfangreicher Planungsvorhaben. Zürich: Verlag Industrielle Organisation.
- Ericsson, K.A., Simon, H.A. (1984): Protocol Analysis: Verbal Reports as Data. Cambridge, MA: MIT Press.
- Floyd, C. (1984): A Systematic Look at Prototyping. In: Approaches to Prototyping. (Budde, R. et al.; Hrsg.). Heidelberg: Springer.
- Glinz, M. (1999). Eine geführte Tour durch die Landschaft der Software-Prozesse und -Prozessverbesserung. In: Informatik/Informatique Nr. 6, Dez. 1999.
- Gould, J.D., Lewis, C. (1984): Designing for Usability – Key Principles and What Designers Think. In: Proceedings of the ACM, Human Computer Interaction. 50–53.
- Greenbaum, J., Kyng, M. et al. (1991): Design at Work: Cooperative Design of Computer Systems. Hillsdale, NJ: Lawrence Erlbaum.

- Hacker, W. (1992): Expertenkönnen: Erkennen und Vermitteln. Göttingen: Verlag für Angewandte Psychologie.
- Hall, A. (1978): A Methodology for Systems Engineering. Princeton: Van Nostrand.
- Heinrich, L.J. (1997): Management von Informatik-Projekten. München: Oldenbourg.
- Johnson, P. (1992): Human Computer Interaction: Psychology, Task Analysis and Software Engineering. Cambridge, MA: Harvard University Press.
- Jungk, R. (1994): Zukunftswerkstätten: mit Phantasie gegen Routine und Resignation. München: Heyne.
- Keil, M., Carmel, E. (1995): Customer-Developer Links in Software Development. In: Communications of the ACM, 38(5). 33–44.
- Kuhnt, B., Müllert, N.R. (2000). Moderationsfibel Zukunftswerkstätten: verstehen, anleiten, einsetzen; das Praxisbuch zur Sozialen Problemlösungsmethode Zukunftswerkstatt (3. Aufl.). Münster: Ökoptopia.
- Martin, R. (1993): Einflussfaktoren auf Akzeptanz und Einführungsumfang von Produktionsplanung und -steuerung (PPS). Frankfurt: Peter Lang.
- Mayring, P. (2002): Qualitative Inhaltsanalyse: Grundlagen und Techniken. Weinheim: Deutscher Studien Verlag.
- Miles, M.B., Hubermann, M. (1994): Qualitative Data Analysis: An expanded sourcebook. Thousand Oaks: SAGE Publ.
- Mumford, E., Weir, M. (1979). Computer systems in work design – the ETH-ICS method: Effective Technical and Human Implementation of Computer Systems. New York: John Wiley & Sons.
- Norman, D., Draper, S. (1986): User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum.
- Rauterberg, M., Spinass, P. et al. (1994): Benutzerorientierte Software-Entwicklung: Konzepte, Methoden und Vorgehen zur Benutzerbeteiligung. Schriftenreihe Mensch – Technik – Organisation (Ulrich, E.; Hrsg.), Band 3. Zürich/Stuttgart: Verlag der Fachvereine/Teubner.

- Schönsleben, P. (2001): Integrales Informationsmanagement: Informationssysteme für Geschäftsprozesse – Management, Modellierung, Lebenszyklus und Technologie. Berlin, Heidelberg, New York: Springer.
- Sommerville, I. (1989): Software Engineering. – 3<sup>rd</sup> ed. Reading, MA: Addison-Wesley.
- Standish Group (2003): Chaos Report. West Yarmouth, Ma.
- Strohm, O. (1990): Arbeitsorganisation, Methodik und Benutzerorientierung bei der Softwareentwicklung. Projektberichte Nr. 2 zum Forschungsprojekt Benutzer-orientierte Softwareentwicklung und Schnittstellengestaltung (P. Spinass, M. Rauterberg et al.; Hrsg.). Zürich: ETH, Institut für Arbeitspsychologie.
- Strohm, O. (1995): Produktionsplanung und -steuerung im Industriebetrieb aus arbeitspsychologischer Sicht: Arbeits- versus Technik-orientierte Gestaltungskonzepte; Dissertation Universität Bern.
- Ulich, E. (2001): Arbeitspsychologie, 5. Aufl. Zürich/Stuttgart: vdf Hochschulverlag/Schäffer-Poeschel.
- Wehner, T., Clases, C., Endres, E. (1996): Situiertes Lernen und kooperatives Handeln in Praxisgemeinschaften. In: Zwischenbetriebliche Kooperation: Die Gestaltung von Lieferbeziehungen (Endres, E., Wehner, T.; Hrsg.). Weinheim: Psychologie Verlags Union. 71–85.
- Zehnder, C.A. (2002): Informatik-Projektentwicklung: Eine Einführung für Informatikstudenten und Praktiker, 4. Aufl. Zürich: vdf Hochschulverlag.







## Projektantrag

### Kapitelinhalt

- Analyse und Klärung der Projektidee und Problemstellung
- Definition der Projektzielsetzung und der unternehmerischen Ziele
- Festlegung des Projektinhalts und des angemessenen Umfangs
- Dokumentation Projektantrag mit Wirtschaftlichkeitsbetrachtung

Am Anfang eines Projektes steht ein Problem oder eine Projektidee. Nach der Identifikation derselben gilt es, das Problem vertieft abzuklären, die Zielsetzungen zu definieren, den Inhalt und die Projektgrenzen abzustechen sowie die Ergebnisse in einem sauberen Projektantrag festzuhalten. Dies ist notwendig, um das Projekt gegenüber dem Management zu begründen und die Freigabe zu erwirken.

Der Umfang eines Projektantrags kann je nach Thematik und Projektgröße von einer Seite bis zu einem umfangreichen Dokument mit einer integrierten Wirtschaftlichkeitsbetrachtung (engl.: “Business Case”) reichen. Im Projektantrag muss es darum gehen, aus einem unternehmerischen Blickwinkel abzuklären, welches der Projektfokus sein sollte und zu welchen Kosten welcher Nutzen erzielt werden kann.

In diesem und den nachfolgenden Kapiteln werden wir den Projektprozess am Beispiel eines Customer-Relationship-Management-Systems (CRM-System) zur Optimierung des Vertriebs einer Fahrradfirma beschreiben. In Ergänzung dazu wird gelegentlich auf ein Projekt aus einem “artfremden” Bereich verwiesen, welches die Transportanlage eines Kurortes betrifft.

## 9.1 Inhalt und Vorgehen

### 9.1.1 Inhalt

Ein "Projektantrag" ist ein gut strukturierter Investitionsantrag zur Beurteilung eines Projektes durch das Management. Bei Projekten darf es nicht nur darum gehen, Geld für ein Projekt zu beantragen. Vielmehr müssen bereits in der Antragsphase *strategische und wirtschaftliche* Überlegungen vorgenommen werden – es muss ein so genannter "Business Case" miterstellt werden. Echte Bedürfnisse des Unternehmens sollen erkannt, der Rahmen abgesteckt, Lösungen vorgeschlagen, die Konsequenzen des Handelns aufgezeigt und eine Problemstellung formuliert werden. Ferner soll der Projektantrag bereits explizite Angaben zu den notwendigen Investitionen sowie zur Wirtschaftlichkeit enthalten. Die Bearbeitung des Projektantrages sollte noch keine nennenswerten Kosten verursachen und wird meist durch vorhandene Budgetmittel finanziert.

### 9.1.2 Vorgehen

Die Erstellung des Projektantrages läuft gemäß dem Problemlösungszyklus des Systems Engineering ab (s. Abb. 113). Zunächst muss die Problemstellung analysiert werden. Anschließend erfolgt die Definition der Projektzielsetzung. Entsprechend dem Denken in Varianten müssen danach unterschiedliche Projektvarianten erarbeitet werden. Die beste Projektvariante wird detailliert und in Form eines Projektantrages dokumentiert.

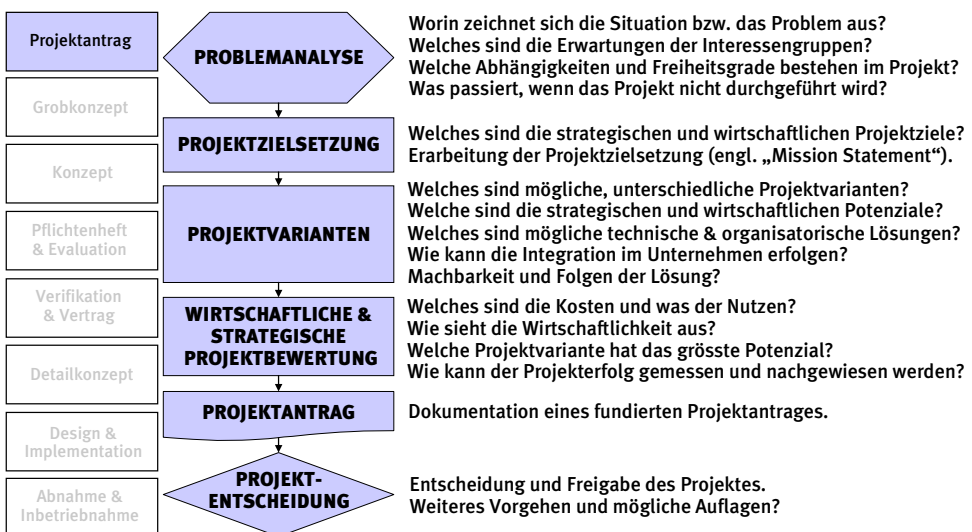


Abb. 113: Vorgehensschritte der Projektantragstätigkeit

### 9.1.3 Ebene der Systembetrachtung im Projektantrag

In diesem und in den nachfolgenden Kapiteln soll jeweils aufgezeigt werden, auf welcher Systemebene, bezogen auf den Fortschritt im Projekt, analysiert und gestaltet werden soll.

So liegt es beispielsweise auf der Hand, dass im Rahmen der Projektinitialisierung noch keine Details des Systems analysiert werden sollten. Mit zunehmender Konkretisierung im Rahmen des Grobkonzeptes, des Konzepts und den weiteren Entwicklungsstufen werden tiefere Systemebenen betrachtet und konzipiert werden müssen (s. Abb. 114).

Die Betrachtungsebene des Projektantrages liegt dementsprechend auf der allerhöchsten Systemebene. Es wird das Gesamtsystem betrachtet und es gilt zudem, auf die relevanten Umsysteme näher einzugehen, um das Projekt von seinen Umsystemen abzugrenzen.

Auf eine formale Systemanalyse sollte zu diesem Zeitpunkt meist noch verzichtet werden. Eine strukturierte Erhebung einer Geschäftsprozesslandkarte würde zu weit führen. Allenfalls bestehen erste grobe Vorstellungen, welche Geschäftsprozesse das Projekt abdecken sollte.

Im Rahmen des Projektantrags sollen primär alternative und unterschiedliche Projektvarianten gesucht und anschließend das beste Projekt in Form eines Projektantrags formuliert werden. Es geht aber noch nicht darum, konkrete Lösungen zu entwickeln.

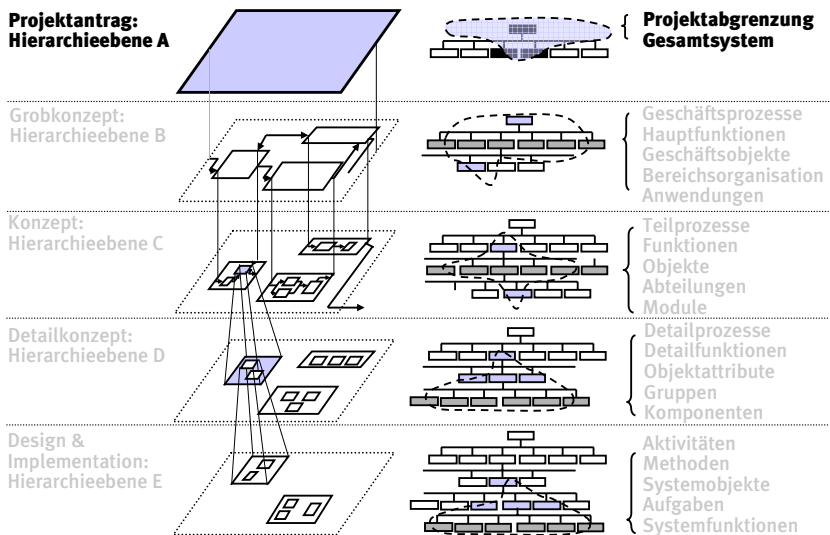


Abb. 114: Betrachtungstiefe des Systems im Projektantrag

## 9.2 Problemanalyse



Ganz am Anfang eines neuen Projektes steht meist ein Problem, eine Idee oder gelegentlich gar eine Produktlösung. Solche Ideen sollten zunächst vertieft auf die Ausgangslage und ihre Beweggründe hin analysiert werden. Es muss hinterfragt werden, welche verborgenen Bedürfnisse damit eigentlich abgedeckt werden sollen und ob tatsächlich auch die richtige Aufgabe gestellt wird. Die dahinter steckenden Absichten und die Erwartungen an die Lösung sind zu konkretisieren. Gemäß Abschnitt 8.8.1 besteht die Problem- und Aufgabenanalyse aus den folgenden Schritten:

- a) Aufgabenstellung analysieren und hinterfragen
- b) Wichtige Aspekte herauschälen (Schwierigkeiten, Missstände)
- c) Anstoß zur Aufgabe ergründen
- d) Randbedingungen und Freiheitsgrade ermitteln
- e) Unklare Begriffe klären

Im Rahmen einer Aufgabenanalyse ist weiter zu untersuchen, wie die späteren Auftraggeber das bestehende Problem beurteilen und welches die exakten Erwartungen an eine Problemlösung sind.

Eine besondere Aufmerksamkeit sollte also den möglichen Beteiligten und Betroffenen gelten. Im Rahmen der Aufgabenanalyse sind diese zu befragen, um eine ganzheitliche Sicht auf die Problemstellung zu finden.

Der Aufgabenanalyse kommt im Projektantrag eine ganz zentrale Bedeutung zu. Es muss nämlich auch überprüft werden, ob die identifizierten Probleme allenfalls lediglich Symptome einer sehr viel umfassenderen Problematik sind. Wäre dies der Fall, dann müsste man konsequenterweise das Projekt weiter fassen als ursprünglich beabsichtigt. Es macht in der Regel keinen Sinn, lediglich Symptome zu bekämpfen, anstatt die Probleme an der Wurzel zu beheben.

Im Rahmen eines Projektantrags wird man sich mit der geschäftlichen Situation, der Organisation und den vorhandenen bzw. bereits eingeplanten Anwendungen vertraut machen müssen. Abgrenzungen zu anderen Projekten bzw. Anwendungen sollten ebenfalls erkannt und Doppelspurigkeiten vermieden werden.

Sodann sind bereits früh Vorstellungen zu entwickeln, welches die Systemgrenzen darstellen. Auch bestehende Lösungsansätze werden zur Kenntnis genommen, dürfen aber bei der späteren Lösungssuche nicht einschränkend wirken. Es ist zu klären, was passiert, wenn das Projekt nicht durchgeführt würde. Abschließend wird man eine zusammenfassende Problemstellung formulieren können.

### 9.3 Projektzielsetzung



Jedes Projekt muss einer spezifischen Hauptzielsetzung gerecht werden, welche es aufgrund der Aufgabenanalyse abzuleiten und zu definieren gilt. In einem nächsten Schritt wird daher eine klare Projektzielsetzung – ein so genanntes “Mission Statement” – verfasst. Insbesondere bei Produktideen oder Schlagworten im Stile von “CRM”, “SCM” etc. muss präzisiert werden, weil “Schlagworte” bei verschiedenen Personen ganz unterschiedliche Erwartungen wecken.

Die Projektzielsetzung verfolgt den Zweck, den Inhalt und die Zielsetzung des Projektes in einfacher und allgemein verständlicher Form zu formulieren und einen Bezug zu den Unternehmenszielen herzustellen. Es kann bereits gewisse zeitliche und quantitative Angaben zur Projektdauer und den Projektkosten enthalten.

Es folgt ein Beispiel für eine Projektzielsetzung eines Marketing-Systems (s. Abb. 115), das uns wie erwähnt in den nachfolgenden Abschnitten und Kapiteln stets weiter begleiten und als roter Faden der Projektabwicklung dienen wird.

Die Projektzielsetzung kann sich im Verlauf eines Projektes verändern und konkretisiert werden. Es sollte die aktuelle Projektzielsetzung widerspiegeln. Insbesondere am Ende des Projektantrags bzw. nach der Wahl einer Projektvariante wird man es anpassen müssen.

#### PROJEKTZIELSETZUNG FÜR MARKETING-PROJEKT

Das Marketing-Projekt verfolgt das Ziel, die Marketingabteilung, den Vertrieb & den Service hinsichtlich Effizienz und Qualität zukünftig besser zu unterstützen.

Es bezweckt die Professionalisierung des gesamten Marketings mit Hilfe eines geeigneten Informationssystems und angepassten Geschäftsprozessen:

- a) **Transparente Kundeninformationen** (Adressen, Businessangaben)
- b) **Dokumentierte Kundenbeziehungen** (Besuche, Angebote, Mails, Support, Events, Servicefälle, etc.) mit einfacher und übersichtlicher Verwaltung.
- c) **Effiziente Kundenprozesse (Mailings, Events, Vertrieb & Service)**, welche qualitativ hochstehend gestaltet sind.

Das Marketing System ist in den nächsten **1 bis 1 1/2 Jahren** in die neue Systemarchitektur zu integrieren und ist mit einem maximalen Budget von **Euro 500.000** zu versehen.

Abb. 115: Beispiel einer Projektzielsetzung

### 9.3.1 Vervollständigung der Zielsetzungen

Ein weiterer Schritt des Projekt-Zielsetzungsprozesses besteht darin, die Vollständigkeit der Zielsetzungen zu überprüfen und allenfalls fehlende Projektziele zu ergänzen. Zusätzlich zu der auf hoher Ebene abgehaltenen Projektzielsetzung können daher auch gewisse weiterführende Zielsetzungen an das Projekt definiert werden. Je klarer und vollständiger die Zielsetzung an das Projekt formuliert ist, desto besser wird auch kommuniziert, welche Ziele die freizugebenden Mittel verfolgen werden.

Zur Überprüfung der geforderten Vollständigkeit lassen sich unter anderem gewisse allgemeine unternehmerische Trends heranziehen: Viele Unternehmen sehen sich heute z.B. mit kürzeren Lieferfristen oder Reaktionszeiten konfrontiert. Dadurch haben sich die Anforderungen an die *Transparenz der Prozesse* stark erhöht. Sodann müssen Prozesse eine hohe *Flexibilität* (z.B. Kundenvarianten, Erweiterungen, Prozesse) aufweisen. In Zeiten eines raschen Wandels ist überdies die *Anpassungsfähigkeit der IT-Systeme* an grundlegend neue Prozesse von hoher Bedeutung.

Neben Investitionen in Technik werden aber auch neue *Organisationskonzepte* umgesetzt. Es wird beispielsweise zunehmend auf die Kooperation mit Lieferanten und Kunden gesetzt. Daraus ergeben sich wiederum neue Anforderungen an die *Kommunikation mit Umsystemen*. Zusammengefasst ergeben sich folgende Zielsetzungen an IT-Systeme, welche sich zur Vervollständigung der Zielsetzung heranziehen lassen (s. Abb. 116):

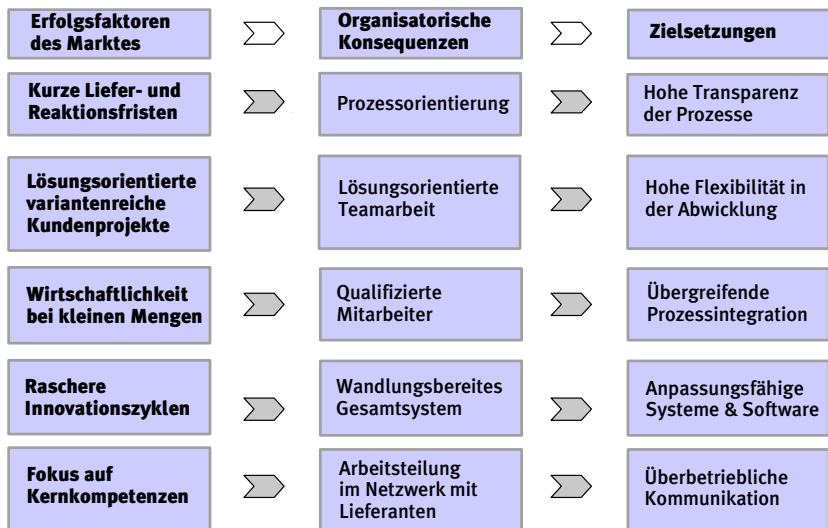


Abb. 116: Unternehmerische Zielsetzungen als Ansatz zur Vervollständigung



## 9.4 Projektvarianten



In der Phase Projektantrag sollte man sich nicht vorschnell auf eine spezifische Projektvariante festlegen. Es sind in jedem Fall unterschiedliche Projektvarianten von unterschiedlichen Umfang und Schwerpunkt in Betracht zu ziehen. Es sind dabei auch kreative Projektideen prüfenswert. Erst anschließend ist die beste Variante auszuwählen.

Mit Hilfe der IT lassen sich sowohl Prozess- als auch Produktinnovationen realisieren (s. Abb. 117) (Pleschak 1996, S. 91). Je innovativer die Prozesse gestaltet werden, desto weiter rechts kommt das Projekt im Portfolio zu liegen. In einer Buchhandlung könnte man den Hauptprozess z.B. mit Hilfe der IT rationalisieren ("Prozessoptimierung"). Wenn der Prozess zusätzlich via Internet abgewickelt würde, dann stellt dies eine Prozessinnovation dar und kommt im Feld "Servicemodelle" zu liegen, da dies den Servicegrad erhöht (Transparenz, Geschwindigkeit). Wird ein elektronisches Buch hingegen über den herkömmlichen Handel vertrieben, dann entspricht dies einem neuen "Produktmodell" ohne Veränderung des Geschäftsprozesses. Wettbewerbsvorteile werden sich meist in der Kombination aus Prozess- und Produktinnovation erzielen lassen, also dort, wo neue "Businessmodelle" entstehen. So könnten z.B. elektronische Bücher via Internet digital verkauft oder "virtuell vorgelesen" werden. Dies soll aufzeigen, dass in der Antragsphase breit gesucht und unterschiedliche Projektvarianten angedacht werden sollten.

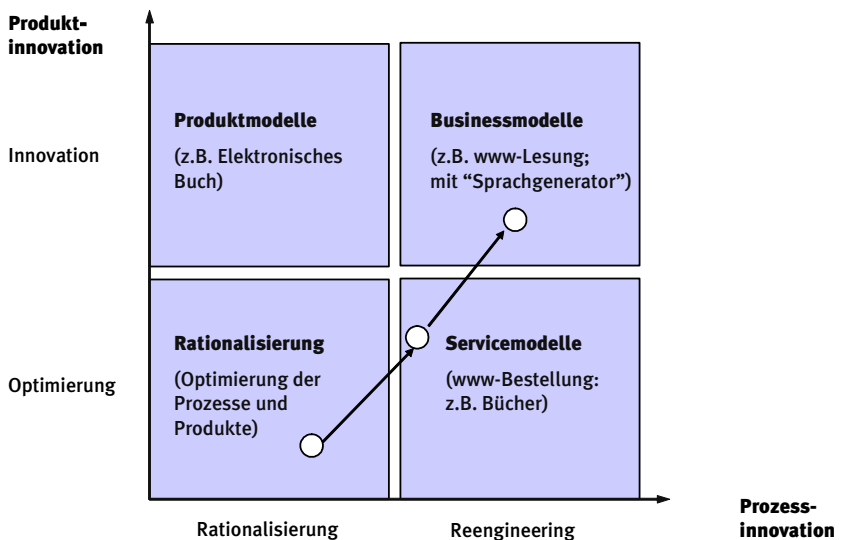


Abb. 117: Strategische Positionierung der Informatik

### 9.4.1 Vorauswahl einer Projektvariante

Nicht in allen Fällen wird man ein neues Businessmodell propagieren wollen. In unserem Beispielprojekt könnten u.U. drei unterschiedliche Projekte zur Diskussion stehen (s. Abb. 118):

- a) Marketing-Internetportal zur Optimierung der Kommunikation und des Internetauftritts gegenüber den Kunden,
- b) Marketing-Servicesystem zur Optimierung der After Sales Kundenbindung auf Basis von Service- und Kundendaten,
- c) CRM-System zur Optimierung des Direktmarketings, der Kundenkontakte und des Außendienstes.

Mit jeder Projektvariante ist ein Lösungsansatz verbunden. Für diese lasse sich nun bereits zu einem frühen Zeitpunkt mit einer so genannten *Argumentenbilanz* überlegen, welche Variante den besten Wert für das Unternehmen erzielt und damit die besten Chancen für eine Freigabe bzw. Realisierung hat. Selbstverständlich können alle drei Varianten auch aufwändiger je mit Hilfe einer Wirtschaftlichkeits- und Portfoliobetrachtung bewertet werden (s. u.).

Chancen für die Bewilligung eines Projektes wird man in jedem Fall dann haben, wenn sich das Projekt strategisch und wirtschaftlich gut positioniert. Mit Hilfe einer einfachen Argumentenbilanz lässt sich unserem Beispiel feststellen, dass die Projektvariante 3 weiterverfolgt werden sollte.

Varianten	Projektvariante 1 Internetportal	Projektvariante 2 Servicesystem	Projektvariante 3 CRM-System
<b>Merkmale</b>			
Vorteile	Besserer Auftritt; Gute Konfiguration; Rasche Umsetzung.	Verbesserter Service; Präventiver Service; Weniger Reklamation; Kleineres Servicelager; Geringe Kosten.	Erhöhter Umsatz; Höhere Transparenz; Raschere Aktionen; Besserer Marktauftritt; Optimierung Verkauf; Auskunftsbereitschaft.
Nachteile	Nur Auskunftssystem; Geringer Mehrumsatz; Kleine Zielgruppe; Ähnliche Projekte.	Gleicher Umsatz; Beschränkte Wirkung; Nur bestehende Kunden.	Implementationskosten; Lange Projektzeit; Schwierige Integration; Unklare Abgrenzung.
Schlussfolgerung	Rang 3	Rang 2	Rang 1

Abb. 118: Auswahl der Projektvariante mit Hilfe einer Argumentenbilanz

## 9.5 Wirtschaftliche und strategische Projektbewertung



Der englische Begriff “Business Case” betont, dass ein Projekt nicht Selbstzweck sein darf, sondern jede Investition eine klare wirtschaftliche bzw. unternehmerische Begründung erfordert. Nachdem die optimale Projektvariante skizziert und ausgewählt wurde – in unserem Beispiel hätte man das CRM-Projekt gewählt – so ist dieses Projekt näher auszuformulieren und in Bezug auf die Wirtschaftlichkeit zu überprüfen. Nachfolgend wird der Frage nachgegangen, wie die Kosten, der Nutzen und die Wirtschaftlichkeit von IT-Investitionen ermittelt werden können.

Wirtschaftlichkeit heißt, dass man Kosten dem Nutzen gegenüberstellt und der Nutzen die Kosten übersteigt. Es bestehen unterschiedliche Verfahren zur Beurteilung der Wirtschaftlichkeit (s. Abb. 119). Die Wirtschaftlichkeitsverfahren stützen sich auf *quantifizierbare* Größen ab, wie z.B. auf direkte Einsparungen.

In Ergänzung dazu müssen auch *nicht quantifizierbare* Größen wie strategische Vorteile und Risiken zur Beurteilung herangezogen werden. In der Regel werden dazu so genannte Portfoliomethoden verwendet (s. u.). Die Wirtschaftlichkeitsermittlung wird in der Projektantragsphase noch sehr ungenau sein. Die Wirtschaftlichkeit wird daher während dem Grobkonzept nochmals überprüft und mit fortschreitendem Projekt laufend exakter ausfallen. Das dargestellte Prinzip bleibt indessen stets dasselbe und wird aus didaktischen Gründen bereits hier ausführlich behandelt.

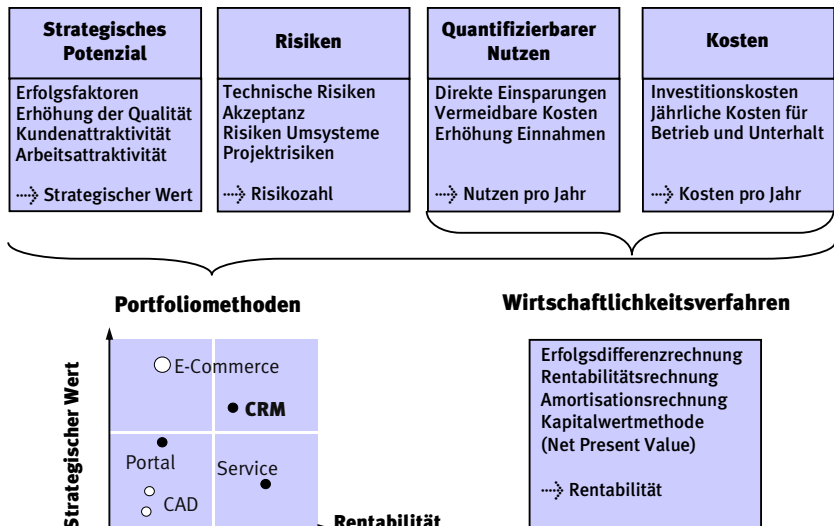


Abb. 119: Übersicht der Methoden zur Projektbewertung

### 9.5.1 Kostenkalkulation

An erster Stelle der Betrachtungen zur Wirtschaftlichkeit steht die Ermittlung der anfallenden Investitionskosten sowie der jährlich wiederkehrenden Betriebskosten (s. Abb. 120). In unserem Beispiel beschränken wir uns auf eine Projektvariante. Die Schätzung der Kosten stellt im Gegensatz zur Nutzenermittlung meist noch den einfacheren Teil dar. Gelegentlich werden Investitions- und Betriebskosten allerdings stark unterschätzt. Diesen Schluss könnten zumindest die oben zitierten Studien zu den Kostenüberschreitungen in Projekten zulassen. Eine Wirtschaftlichkeitsbetrachtung wird typischerweise über mehrere Jahre vorgenommen, z.B. über einen Zeitraum von 5 bis 7 Jahren. Investitionen rechtfertigen sich ja in der Regel auch erst über einen längeren Zeitraum.

#### Investitionskosten

Das unten stehende Kalkulationsbeispiel zeigt typische Investitionskomponenten von IT-Projekten auf. Nur selten werden in korrekter Weise auch die internen Projektkosten der Mitarbeiteraufwände miteinkalkuliert.

#### Betriebskosten

Unter den Betriebskosten müssen neben den extern anfallenden Kosten zwingend auch die Kosten für den internen Betrieb & Unterhalt miteinbezogen werden. Gelegentlich werden hierfür ja neue Mitarbeiter eingestellt. In den Kosten werden keine Abschreibungen berücksichtigt, da dies für die Kapitalwertmethode später nicht erforderlich sein wird.

	Jahr 1	Jahr 2	Jahr 3	Jahr 4	Jahr 5
<b>Investitionskosten</b>					
Projektierungskosten (Vorstudie)	30				
<b>Hardwarekosten</b>					
Server & Arbeitsstationen	10				
Netzwerk, Hubs	5				
Hilfsgeräte (USB, Drucker)	5				
<b>Softwarekosten</b>					
Anwendungssoftware	60				
Systemsoftware	10				
Datenbanksoftware	10				
<b>Realisierungskosten</b>					
Externe Projekt-Dienstleistungen	120				
Programmierung, Tests	20				
Datenübernahme	10				
Dokumentation, Schulung	20				
<b>Betriebskosten</b>					
Update-, Upgradekosten		3	3	3	3
Wartung, Support, Hotline		9	9	9	9
Dienstleistung, Schulung		7	7	7	7
Betriebskosten (int. /ext.)		3	3	3	3
Telekomm., Raum-, Energie, etc.		3	3	3	3
<b>Total</b>	<b>300</b>	<b>25</b>	<b>25</b>	<b>25</b>	<b>25</b>

Abb. 120: Beispielhaftes Schema zur Kostenkalkulation

### 9.5.2 Nutzenkalkulation

In einem zweiten, anspruchsvolleren Schritt muss nun der Nutzen des Projektes abgeschätzt werden. Man unterscheidet dabei typischerweise zwischen dem *quantifizierbaren Nutzen* (z.B. geringere Wartungskosten) und dem *nicht quantifizierbaren Nutzen* (z.B. Arbeitsplatzattraktivität). Der nichtquantifizierbare Nutzen ist Gegenstand der späteren Portfolio Betrachtung. Das nachfolgende Kalkulationsschema erlaubt, den quantifizierbaren Nutzen von IT-Vorhaben abzuschätzen. Zur vereinfachten Schätzung werden jeweils einen Minimum, ein Maximum und der erwartete Mittelwert geschätzt (s. Abb. 121).

Falls bestehende Kosten eingespart werden, dann spricht man von *direkten Einsparungen*. Unter den *vermeidbaren Kosten* fasst man jene Kosten zusammen, welche heute noch nicht anfallen, aber in Zukunft zusätzlich anfallen würden, würde man das System nicht in Betrieb nehmen (Opportunitätskosten). Ein Nutzen entsteht, da das System diese nun vermeidet.

Von einer *Erhöhung der Einnahmen* spricht man immer dann, wenn durch das System in der Firma ein zusätzlicher Gewinn anfällt.

Bei Ersatzinvestitionen wie z.B. ERP-Systemablösungen präsentieren sich solche Berechnungen meist sehr schwierig. Man müsste als Vergleich den Betrieb ohne jegliche Informatikmittel heranziehen. Dies ist aber kaum praktikabel. Deshalb werden eher strategische Begründungen gesucht.

	Jahr 1			Jahr 2			Jahr 3			Jahr 4			Jahr 5		
<b>Einsparungen</b>	Min	Mit	Max	Min	Mit	Max	Min	Mit	Max	Min	Mit	Max	Min	Mit	Max
<b>Direkte Einsparungen</b>															
Erhöhte Produktivität	10	30	50	15	30	55	20	30	40	10	30	50	10	30	50
Reduktion Überzeiten	0	15	30	5	15	25	0	15	30	0	15	30	0	15	30
Weniger Materialkost.	5	5	5	4	5	6	5	5	5	5	5	5	5	5	5
Geringere Mietkosten	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Geringere Wartung	5	5	5	4	5	6	5	5	5	5	5	5	5	5	5
Geringere Lagerkosten	0	5	10	0	5	10	0	5	10	0	5	10	0	5	10
Geringere Fehlerkosten	0	10	20	0	10	20	0	10	20	0	10	20	0	10	20
Geringere Ausfallkost.	0	5	10	0	5	10	0	5	10	0	5	10	0	5	10
Geringere Zinskosten	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
etc.															
<b>Vermeidbare Kosten</b>															
Vermeidung Unterhalt	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
Vermeidung Ausfälle	3	5	7	3	5	7	3	5	7	3	5	7	3	5	7
<b>Erhöhung Erträge</b>															
Höhere Erträge	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
Höhere Zinserträge	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Geringeres Delkredere	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4
<b>Gesamtnutzen</b>		<b>100</b>			<b>100</b>			<b>100</b>			<b>100</b>			<b>100</b>	

Abb. 121: Beispielhaftes Schema zur Nutzenkalkulation

### 9.5.3 Wirtschaftlichkeitsverfahren

Basierend auf der Kosten- und Nutzenkalkulation kann mit Hilfe von Wirtschaftlichkeitsverfahren die Rentabilität berechnet werden. Das elementarste Verfahren ist die Differenzrechnung:

$$\text{Erfolg} = \text{Nutzen} - \text{Kosten}.$$

#### Rentabilitätsrechnung (Return on Investment)

Ein zweites Verfahren berechnet die durchschnittliche Rendite der Investition. Man könnte den ROI auch unter Berücksichtigung von Abschreibungen jährlich auf dem aktuellen Investitionsrestwert berechnen, müsste dann aber auch in den Kosten "Abschreibungen" ausweisen. Der ROI entspricht der Verzinsung des Kapitals:

$$\text{ROI} = \text{Erfolg} / \text{Investition} * 100.$$

#### Amortisationsrechnung (Pay-back-Dauer)

Die Amortisationsrechnung berechnet die Dauer in Jahren, in welcher sich die Investition amortisiert. Dies entspricht einer Break-Even-Analyse von Kosten und Nutzen (s. Abb. 122). Die Dauer berechnet sich:

$$\text{Pay-back-Dauer} = \text{Investition} / (\text{Nutzen} - \text{Kosten})$$

#### Kapitalwertmethoden (Net Present Value, NPV)

Im Gegensatz zu den "statischen Verfahren" berücksichtigen "dynamische Verfahren" den Zeitpunkt des Geldflusses. Je weiter ein Kosten- oder Nutzelement in der Zukunft liegt, desto weniger ist es heute wert. Die Kapitalwertmethode diskontiert diese Elemente mit einem Zinssatz (z.B. 8%).

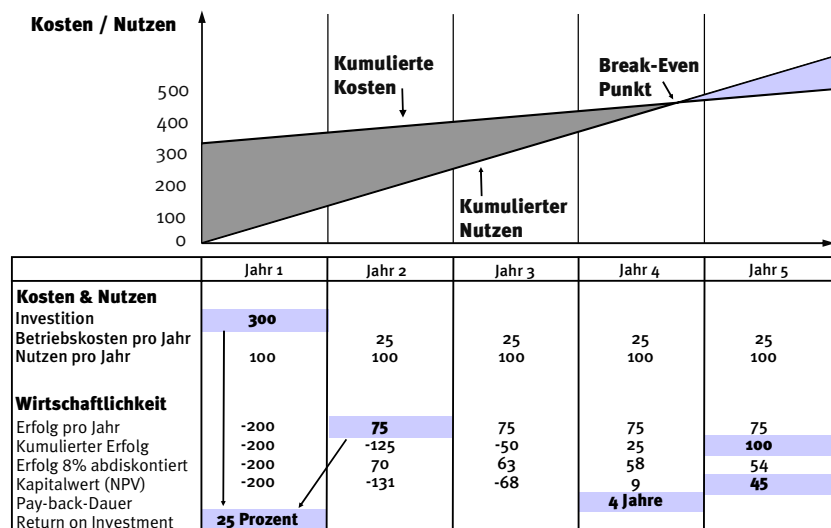


Abb. 122: Beispielhafte Wirtschaftlichkeitsbetrachtung

#### 9.5.4 Strategisches Potenzial

Es wird leider nicht immer gelingen, den Nutzen von IT-Projekten quantitativ im Sinne der obigen Wirtschaftlichkeitsverfahren zu untermauern. Auch in wissenschaftlichen Studien konnte nachgewiesen werden, dass sich der Aufwand und Nutzen betrieblicher Informationssysteme häufig etwa die Waage halten (Brynjolfsson 1993).

Zunehmend werden daher bei IT-Projekten die strategischen Faktoren und Vorteile mitberücksichtigt (Wettbewerbsvorteile, Marktchancen etc.). In letzter Zeit wird allerdings der strategische Nutzen von Informationssystemen etwas in Frage gestellt. Insbesondere wird teilweise bestritten, mit Hilfe der Informationstechnologie seien Vorteile gegenüber der Konkurrenz zu erzielen (Carr 2003).

Wenn wir daher nachfolgend von “strategischen Faktoren” sprechen, so fassen wir darunter ganz generell all jene Argumente zusammen, welche *nicht quantifizierbar* und damit im Rahmen der Wirtschaftlichkeitsverfahren bisher unberücksichtigt geblieben sind. Es müssen dies also nicht nur “unternehmensstrategische” Faktoren sein.

Die Berücksichtigung der nicht quantifizierbaren Faktoren erfolgt wie erwähnt meistens mit so genannten Portfoliomethoden. Das Konzept des Portfoliomanagements wurde ursprünglich durch das Marktattraktivität-Wettbewerbsvorteil-Portfolio bekannt (Pleschak et al. 1996, S. 66). Die klassische Vierfeldermatrix stammt aus den 60er-Jahren und gehört heute zu den Standardtechniken der strategischen Führung.

Bedingt durch den großen Erfolg strategischer Portfolios und deren Management-gerechter Verdichtung der Information wurden in der Folge weitere Portfolios für das IT-Management entwickelt.

Unternehmen haben ja in aller Regel eine größere Anzahl von mehr oder weniger wichtigen IT-Projekten ganz unterschiedlicher Größe und Ausrichtung. Die Summe aller Projekte und Anwendungen bezeichnet man als das IT-Portfolio des Unternehmens (s. Abb. 123) .

Im Rahmen der Budgetprozesse werden nun neue Projektanträge ebenfalls durch Portfoliotechniken bewertet und selektioniert.

Wir werden nachfolgend das so genannte Potenzialportfolio betrachten, welches auch dazu dienen wird, den Technologieeinsatz und die Projektrisiken abschätzen. Es zeigt auf der vertikalen Achse den strategischen und auf der horizontalen Achse den quantifizierbaren Nutzen auf.

### 9.5.5 Zusammenfassende Projektbewertung im Portfolio

Das unten dargestellte Portfolio zeigt auf, wie sich die Software-Projekte (Kreise) und -Anwendungen (Dreiecke) eines Unternehmens hinsichtlich der strategischen Bedeutung bzw. der Wirtschaftlichkeit positionieren (s. Abb. 123). Im Beispiel sind die drei möglichen Projektvarianten ersichtlich, wobei das CRM-Projekt den besten Gesamtnutzen erzielt.

Die vertikale Achse zeigt den strategischen Wert und die horizontale Achse das Ergebnis des gewählten Wirtschaftlichkeitsverfahrens. Auf der horizontalen Achse wird das oben ermittelte Rationalisierungspotenzial dargestellt.

Die Vertikale zeigt demgegenüber alle nicht quantifizierbaren Faktoren. Dies entspricht dem relativen strategischen Potenzial der einzelnen Projekte und Anwendungen. Es sind dies z.B. alle Faktoren, welche die Erfolgsfaktoren des Unternehmens unterstützen, z.B. Auskunftsbereitschaft, Transparenz, Qualität, Kundenorientierung, Durchlaufzeiten, Verfügbarkeit von Informationen, rasche Entscheidungsfindung.

Die Positionierung dieser nicht quantifizierbaren Größen erfolgt idealerweise mit Hilfe einer gewichteten Punktevergabe (Nutzwertanalyse). Ein Projekt erhält damit einen höheren strategischen Wert, je besser es in Summe die strategischen Faktoren unterstützt. Die E-Commerce-Anwendung scheint also die nicht quantifizierbaren Faktoren besser zu unterstützen als das CRM-Projekt.

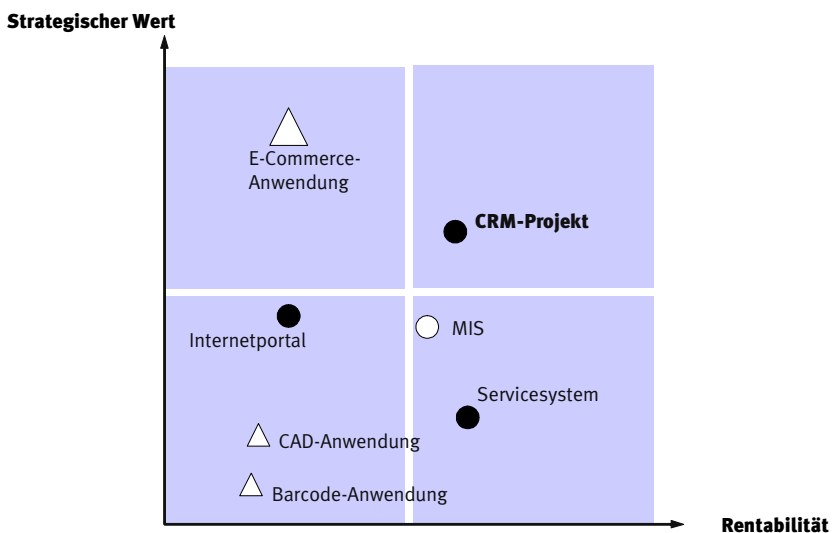


Abb. 123: Potenzial-Portfolio der Projekte und Anwendungen



### 9.5.6 Beurteilung der Projektrisiken

In einem nächsten Schritt sollte in ähnlicher Weise wie beim strategischen Wert eine Abschätzung des Risikowertes vorgenommen werden, d.h. eine Nutzwertanalyse betreffend der Risikofaktoren (s. Abschn. 18.10). Es betrifft dies die Projektrisiken, Akzeptanzrisiken, Risiken im Bereich der Um Systeme (Kunden, Lieferanten etc.) sowie Risiken des Einsatzes neuer Technologien. Ergänzend sollte klar definiert werden, wie der Projekterfolg nach Projektabschluss gemessen werden kann. Hierfür eignet sich eine Liste der Messkriterien.

Zur Beurteilung der Risiken der Informationstechnologie können die Grundkonzepte des Technologiemanagements und der Technologielebenszyklen angewendet werden (Saad et al. 1991, S. 69). Es werden in der Regel vier Stadien im Lebenszyklus von Technologien definiert.

#### **Basistechnologie**

Basistechnologien bezeichnen allgemein verwendete Technologien, die sich bereits in der Reifephase befinden. Ihr Einsatz ist mit eher kleinen unternehmerischen Risiken behaftet. Allerdings ist auch klar, dass alle Mitbewerber die Basistechnologien beherrschen und damit keine Möglichkeit vorhanden ist, wirklich relevante Marktvorteile zu erzielen. Beispiele von Basistechnologien wären zurzeit z.B. SQL-Datenbanken und Client-Server-Konzepte.

#### **Schlüsseltechnologie**

Schlüsseltechnologien stehen an der Pforte zur Basistechnologie, befinden sich aber noch in einer starken Entwicklungsphase. Sie haben infolge ihres starken Wachstums einen überragenden Einfluss auf die Wettbewerbsfähigkeit. Sie bieten darüber hinaus ein großes Potenzial für innovative Entwicklungen sowohl im Prozess- als auch im Produktbereich. Beispiele von Schlüsseltechnologien könnten zurzeit sein: Internettechnologien wie XML und webbasierte Scriptsprachen.

#### **Schrittmachertechnologie**

Schrittmachertechnologien befinden sich noch in einem frühen Entwicklungsstadium. Es zeichnen sich aber bereits auch erste konkrete Einsatzgebiete ab. Selbstverständlich können die Potenziale zur Differenzierung gegenüber den Mitbewerbern sehr groß sein. Dies ist allerdings noch mit gewissen Unsicherheiten behaftet. Beispiele könnten sein: XML-Datenbanken und kabellose Hochgeschwindigkeitsnetzwerke.

#### **Zukunftstechnologie**

Zukunftstechnologien befinden sich im Forschungsstadium an Universitäten oder in Grundlagenforschungslabors von Unternehmen. Sie werden daher für Projekte im betrieblichen Umfeld noch keine Rolle spielen.

### Risikoanalyse der eingesetzten Technologie

Lässt sich dieses Lebensphasenkonzept nun aber tatsächlich auch sinnvoll anwenden – oder bleibt es ein theoretisches Konstrukt? Der praxisrelevante Gedanke liegt nun darin, dass sich das Risiko der Projekte von der Basistechnologie hin zu Zukunftstechnologien fortlaufend steigert. Man geht also davon aus, dass sich Basistechnologien dadurch auszeichnen, dass sie weitgehend beherrscht werden. Schlüsseltechnologien sind demgegenüber bereits mit viel größeren Unsicherheiten und Stolpersteinen verbunden. Ein hohes Risiko und handfeste Überraschungen wird man beim Einsatz von Schrittmachertechnologien erwarten müssen. Solche Projekte werden sich zeitlich auch nicht mehr klar planen lassen. Der erhöhten Technologieunsicherheit steht nun aber – und dies ist der zentrale Punkt – auch ein viel größeres Chancenpotenzial gegenüber.

Unter diesen Voraussetzungen ergibt sich eine Zuordnung der Potenziale zu den Lebensphasen (s. Abb. 124). Man wird genau dort kalkulierte Risiken eingehen, wo ein hohes unternehmerisches Potenzial vorhanden ist. In unserem Beispiel wird man also keine Schlüsseltechnologien für die Ablösung eines ERP-Systems einsetzen – oder aber nur für den Fall, dass damit gleichzeitig auch die E-Commerce-Anwendung mit abgedeckt werden könnte. Die Servicelösung verspricht ein recht hohes Potenzial – womit gewisse Risiken neuer Technologien in Kauf genommen werden.

Die Beurteilung der weiteren Risiken wurde nicht weiter ausgeführt. Sie wird im Kapitel Projektmanagement weiter ausgeführt (s. Abschn. 18.10).

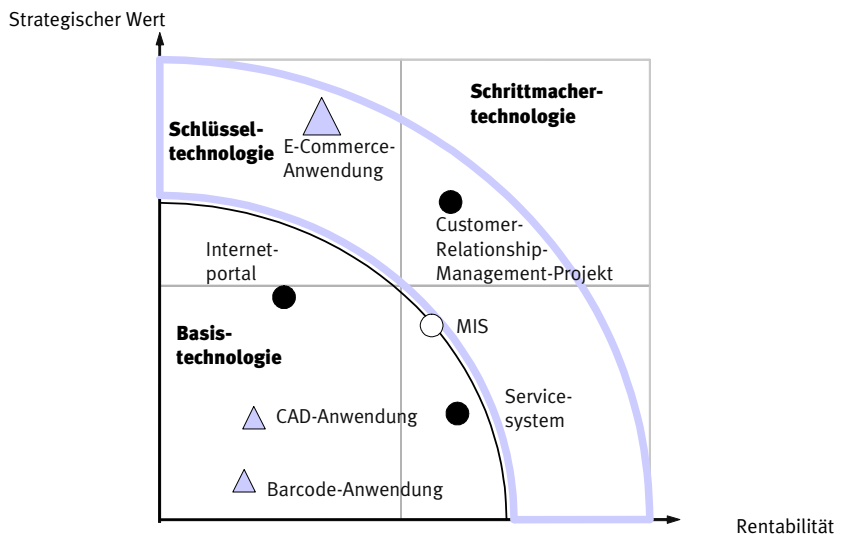


Abb. 124: Risiko- und Chancenabwägung beim Technologieeinsatz

## 9.6

## Projektantrag



Die Struktur und der Inhalt eines Projektantrages richten sich nach den Arbeitsschritten des Problemlösungszyklus (Analyse, Ziele, Konzept, ...). Der Umfang wird projektabhängig festgelegt werden. Bei kleineren Projekten wird man eher vom einseitigen "Projektantrag" sprechen, bei größeren Projekten wird neben dem formellen Projektantrag zusätzlich eine umfassende Wirtschaftlichkeits- und Risikobetrachtung (Business Case) erstellt. In der Struktur werden sich diese beiden Ausprägungen aber nicht wesentlich unterscheiden, einzig in der Länge der Ausführungen.

Der Inhalt eines Projektantrags fasst die Ergebnisse der Initialisierung zusammen und kann wie folgt strukturiert werden:

**I ZUSAMMENFASSUNG****II AUSGANGSLAGE UND PROBLEMANALYSE**

Analyse Ausgangssituation, Schwächen, Problemstellung, Abgrenzung und Abhängigkeiten des Projektes.

**III PROJEKTZIELSETZUNG****IV PROJEKTVARIANTEN**

- **Projektvarianten und Wahl der beantragten Variante**
- **Darstellung des Projektes**

Optimierte Geschäftsprozesse, realisierte Funktionen, zugrunde liegende Datenbestände, organisatorische & technische Aspekte.

**V WIRTSCHAFTLICHKEIT, STRATEGISCHE POTENZIALE & RISIKEN**

- **Kosten-, Nutzen, Wirtschaftlichkeitsbetrachtung**

Einmalige Kosten, wiederkehrende jährliche Kosten, Ressourcenbedarf, quantifizierbarer und nicht-quantifizierbarer Nutzen, Wirtschaftlichkeitszahlen wie Kapitalwert und Pay-back-Dauer etc.

- **Strategischer Nutzen aus Unternehmenssicht**

- **Identifikation und Beurteilung der Risiken**

Organisatorische Risiken, Ressourcenverfügbarkeit, Technische Risiken, Risiken der Benutzerakzeptanz, wirtschaftliche Risiken, Messbarkeit des Projekterfolgs.

- **Konsequenzen bei der Nichtrealisierung des Projektes**

**VI PLANUNG DES PROJEKTES**

- **Etappiierung und Terminierung**
- **Vorschlag zur Projektorganisation**

**VII FORMELLER PROJEKTANTRAG**

## 9.7 Projektentscheidung



Man wird sich nach dem Projektantrag als Ergebnis im Klaren sein, welchen Inhalt und welches Ziel das Projekt verfolgen sollte. Unter Umständen wird man dem Management auch mehrere Projektvarianten mit einer entsprechenden strategischen und wirtschaftlichen Bewertung vorschlagen und die Entscheidung offen halten. In unserem Beispiel hätte man sich entschieden, den Projektantrag für ein umfassendes Marketingsystem (CRM-System) zu erstellen. Die Varianten eines neuen Internetportals bzw. Servicesystems wurden verworfen.

*Analogie: Im Kurort hätte man sich in dieser Phase entschieden, das Projekt "Transportanlage" weiter zu verfolgen. Es ist nach Abschluss des Projektantrag aber noch nicht entschieden, ob ein Sessellift oder besser eine Luftseilbahn gebaut werden sollte. Die Klärung des bestgeeigneten Grobkonzeptes wird erst Bestandteil des nächsten Schrittes sein.*

Als Resultate des Projektantrags fallen gemäß nachfolgender Abbildung Ergebnisse in jeder Phase des durchlaufenen Problemlösungszyklus an. Die wichtigsten Ergebnisse sind tabellarisch zusammengefasst.

Bereich Schritt	Sozio-technisches System	Einführung & Ausbildung	Betrieb & Unterhalt	Projektmanagement & Controlling
<b>Problem-analyse</b>	Problemanalyse; Ermittlung Aufgabenanstoß; Analyse der Ausgangssituation; Analyse der Interessengruppen; Bestimmung der Freiheitsgrade; Klärung Projektabhängigkeiten; Projektabgrenzung; Etwaige Vorleistungen; Konsequenzen Nichtrealisierung; Geklärte Problemstellung.			
<b>Projektzielsetzung</b>	Projektzielsetzung; Auflistung der Projektziele.	Projektziele der Einführung.	Projektziele von Betrieb & Unterhalt.	
<b>Projektvarianten</b>	Projektumfang & -varianten; Analyse der Projektpotenziale; Positionierung Projektvarianten; Identifikation ähnliche Lösungen; Machbarkeitsüberlegungen; Risikobetrachtungen; Folgen der Neulösung.	Varianten bezüglich Einführung & Ausbildung.	Varianten bezüglich Betrieb & Unterhalt.	Projektplanung.
<b>Wirtschaftliche &amp; strategische Projektbewertung</b>	Bewertung Potenziale & Risiken; Bewertung Wirtschaftlichkeit; Messkriterien für Projekterfolg; Auswahl beste Projektvariante.			
<b>Projektantrag</b>	Formulierter Projektantrag.			
<b>Projektentscheidung</b>				Projektentscheidung; Projektauftrag mit allfälligen Auflagen.

Abb. 125: Ergebnisübersicht der Projektantragstätigkeit

## 9.8 Literatur zum Projektantrag

- Brynjolfsson, E. (1993): The Productivity Paradox of Information Technology: Review and Assessment. *Communications of the ACM*, Dec.
- Carr, N.G. (2003): IT Doesn't Matter. In: *Harvard Business Review*, May 2003.
- Hammer, M., Champy, J. (1993): *Reengineering the Corporation*. New York: Harper Business.
- Hebeisen, W. (1999): *Taylor und der Taylorismus*. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 23. Zürich: vdf Hochschulverlag.
- Jantzen-Homp, D. (2000): *Projektportfolio-Management: Multiprojektarbeit im Unternehmenswandel*. Wiesbaden: Gabler.
- Pleschak, F., Sabisch, H. (1996): *Innovationsmanagement*. Stuttgart: Schäffer-Poeschel.
- Saad, K.N. et al. (1991): *Management der F&E-Strategie*. Wiesbaden: Gabler.
- Specker, A. (2000): Unterstützung kritischer Erfolgsfaktoren durch ERP-Systeme: Resultate einer Expertenbefragung anlässlich der Zürcher PPS-Tage 2000. [www.tagungen.bwi.ethz.ch/rueckblickpps](http://www.tagungen.bwi.ethz.ch/rueckblickpps).
- Taylor, F.W. (1913): *Die Grundsätze wissenschaftlicher Betriebsführung*. Berlin: Oldenburg.
- Tschirky, H., Koruna, S. (Hrsg.) (1998): *Technologie-Management: Idee und Praxis*. Zürich: Verlag Industrielle Organisation.
- Ulich, E. (2001): *Arbeitspsychologie*, 5. Aufl. Zürich/Stuttgart: vdf Hochschulverlag/Schäffer-Poeschel.
- Zuboff, S. (1988): *In the Age of the Smart Machine: The Future of Work and Power*. New York: Basic Books.



# 10

## Grobkonzept

### Kapitelinhalt

- Gesamtsystemanalyse, Systemabgrenzung & Umfeldanalyse
- Festlegung der Hauptzielsetzungen
- Integrierte organisatorische und technische Grobkonzeptvarianten
- Auswahl und Machbarkeitsnachweis der besten Lösung

Nachdem das Projekt aufgrund eines gut dokumentierten Projektantrages freigegeben wurde, wird im Rahmen des Grobkonzeptes geklärt, welches das beste Lösungsprinzip zur Erfüllung der Projektzielsetzung darstellt. Dies entspricht einer Vorstudie, in welcher mit 20 Prozent des Gesamtprojektaufwandes ein Grobkonzept skizziert werden soll, welches die zukünftige Lösung bereits zu rund 80 Prozent festlegt. Die in Betracht gezogenen Grobkonzeptvarianten sollten möglichst breit gefächert sein.

In unserem CRM-Beispiel wäre zunächst zu klären, welche Geschäftsprozesse durch das neue IT-System abzudecken sind. Weiter wird man prüfen, ob jeweils eigenständige CRM-Lösungen je Geschäftsbereich vorteilhafter wären als eine gemeinsame Standardlösung für das ganze Unternehmen. Im Rahmen des Grobkonzeptes ist weiter zu verifizieren, ob die Machbarkeit des Projektes gegeben ist und welche organisatorischen, technischen und wirtschaftlichen Konsequenzen damit verbunden sind.

Es sei an dieser Stelle nochmals betont, dass die Kapitelstruktur in Teil 2 dieses Buches (Grobkonzept, Konzept, ...) kein Vorgehensmodell favorisiert: Im Wasserfallmodell fallen die Kapitel mit Phasen zusammen; im Spiralmodell werden alle Kapitel je Phase einmal verkürzt durchlaufen.

## 10.1 Inhalt und Vorgehen

### 10.1.1 Inhalt

Zu Beginn des Grobkonzepts wird die Problemstellung des Projektantrags nochmals überprüft und vertieft geklärt, ob tatsächlich auch das richtige Problem angegangen wird. Das Grobkonzept erhebt den Anspruch, das vorliegende System auch erstmals analytisch sauber abzuklären, die Umssysteme zu identifizieren und deren Einfluss abzuschätzen. Anschließend sind die Grobkonzeptvarianten mit einem vertretbaren Aufwand (80/20-Regel) zu erarbeiten. Die grundsätzliche Machbarkeit sollte ebenfalls geklärt werden, noch bevor größere Investitionen getätigt werden.

### 10.1.2 Vorgehen

Das Grobkonzept beginnt mit einer sauberen Analyse des Gesamtsystems. Danach wird die vorhandene Projektzielsetzung durch Hauptzielsetzungen ergänzt und erweitert. Der zentrale Schritt des Grobkonzeptes besteht darin, alternative organisatorische und technische Grobkonzeptvarianten zu erarbeiten und damit die sozio-technisch beste Lösung zu suchen. Ein verbreiteter Fehler in Projekten liegt darin, dass vorschnell eine Lösung feststeht und vorhandene Lösungsspielräume nicht erkannt werden. Die angestrebte Variantenvielfalt ist im Bereich der organisatorischen und technischen Lösungen zu suchen. Außerdem ergeben sich unterschiedliche Varianten im Lösungsumfang, so im Abdeckungsgrad der Lösung.

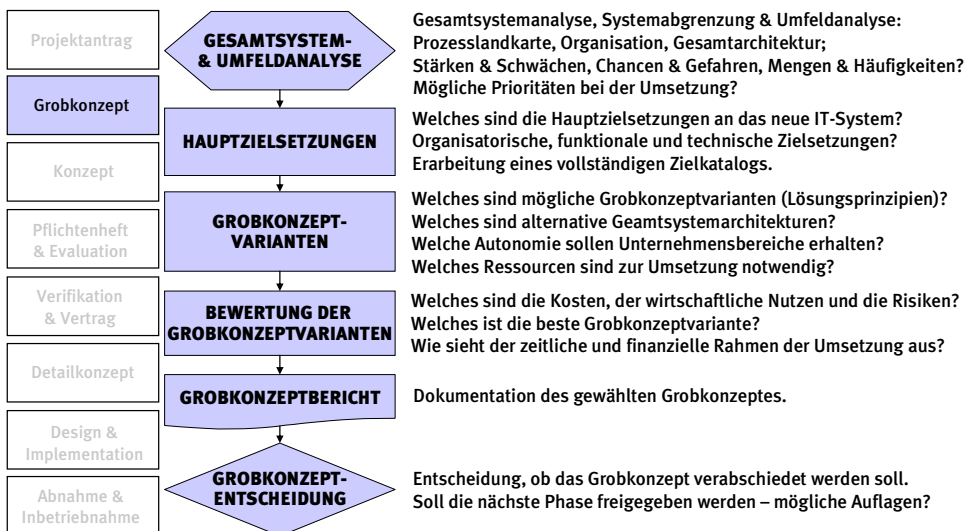


Abb. 126: Vorgehensschritte der Grobkonzepttätigkeit



### 10.1.3 Ebene der Systembetrachtung im Grobkonzept

Wie oben erwähnt, erhebt das Grobkonzept in Erweiterung zum Projektantrag den Anspruch, erstmals wirklich analytische Abklärungen durchzuführen (s. Abb. 127).

Analytisch sauber heißt in diesem Zusammenhang, dass das zu betrachtende Gesamtsystem methodisch sauber modelliert wird und Geschäftsprozesse, Hauptfunktionen, Geschäftsobjekte, die Organisation und die Systemtechnik erhoben, benannt und abgegrenzt werden. Es ist ein primäres Ziel des Grobkonzeptes, eine gute Systemübersicht zu erarbeiten und die Beziehungen und Systemgrenze zu den Umsystemen bereits klar festzulegen. Denn von dieser Grenze hängen die Projektkosten oftmals stark ab. Die Geschäftsprozesse müssen zu diesem Zweck selbstverständlich noch nicht vertieft auf ihre Teilprozesse hin analysiert werden. Dies wird erst Bestandteil des ausführlicheren Konzeptes sein, denn der Analyseaufwand steigt mit jeder Betrachtungsebene exponentiell an. Allenfalls wird man die zentral wichtigsten Geschäftsprozesse bereits im Grobkonzept etwas vertieft betrachten, um zusätzlich an Sicherheit zu gewinnen.

Entscheidend ist in dieser Situation, dass man eine wirklich integrierte Sichtweise des Systems erarbeitet, in dem kein Aspekt vollständig ausgeklammert wird und neben der Prozesssicht Vorstellungen bezüglich der Funktionalität, der relevanten Geschäftsobjekte, der organisatorischen Situation und des technischen Systemumfelds erhält.

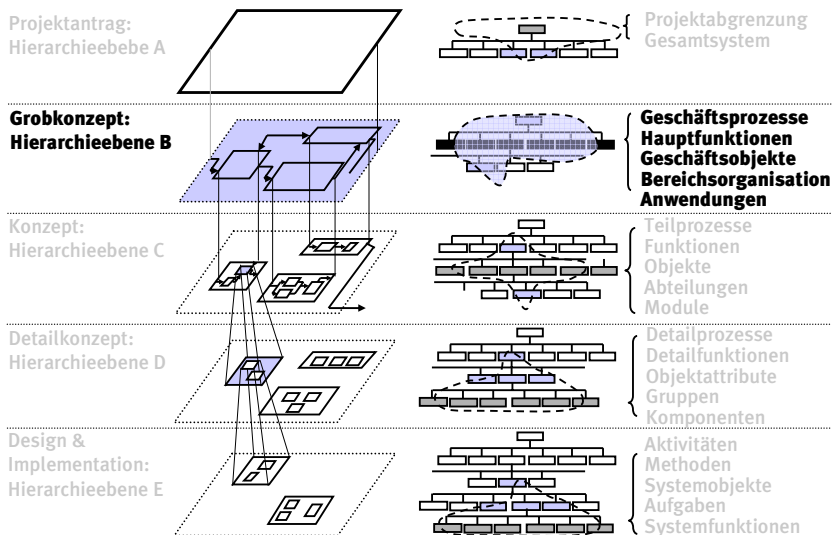


Abb. 127: Betrachtungstiefe des Systems im Grobkonzept

### 10.1.4 Verwendete Modellierungsmethoden

Wir wollen uns der Frage zuwenden, welche Modellierungsmethoden im Rahmen des Grobkonzeptes zur Anwendung gelangen. Die Methoden wurden in Teil 1 dieses Buches bereits ausführlich behandelt. Um Redundanzen zu vermeiden, wird in diesem Teil mehrheitlich auf die Integrations- und Querbezüge zwischen den Darstellungstechniken eingegangen.

Die hier präsentierte Methodenauswahl entspricht außerdem einem Normalvorgehen; im Einzelfall muss dies weder zwingend noch vollständig sein – wichtig sind das Verständnis und die korrekte Integration im konkreten Fall. Der Kreativität des Einsatzes von Modellierungsmethoden und Techniken soll kein Einhalt geboten werden. Betrachtet man das beispielhafte Methoden-Portfolio des Grobkonzeptes (s. Abb. 128), dann fällt auf, dass die *Prozess- und die Aufgabensicht* sowie die *Techniksicht bei der Erarbeitung des Grobkonzeptes betont werden*. Diese Betonung entspricht der Zielsetzung, die Aufbau- und Ablauforganisation (Prozesssicht) in Verbindung mit der Technik zu gestalten. Besonders wichtig sind daher die Diagramme im Schnittpunkt dieser Sichten.

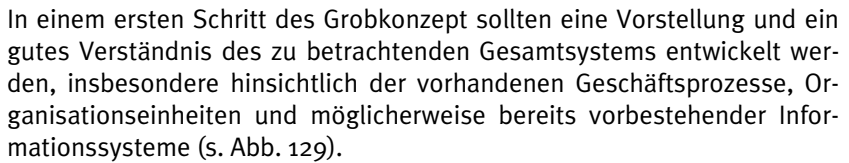
Diese Modelle werden stets zweimal eingesetzt. Einerseits im Rahmen der Ist-Analyse und zweitens zur Dokumentation des Grobkonzeptes. Dabei ist zu beachten, dass die Ist-Analyse meistens technikunabhängig erfolgt, die Konzeption indessen darauf Bezug nimmt, welche Funktionen und Prozesse technikgestützt sind.

<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht	Techniksicht
Prozesssicht	Prozessmodell Prozess- landkarte			Stellenorien- tierter Informa- tionsfluss	System- schnittstellen- diagramm
Funktionssicht		Haupt- funktionen			
Objektsicht			Geschäfts- objekte		
Aufgabensicht				Organisations- modell	
Techniksicht				Technikeinsatz- diagramm	System- architektur / Anwendungen

Abb. 128: Wichtigste Modellierungsmethoden im Grobkonzept

### 10.2.1 Systemabgrenzung und Umfeldanalyse

### 10.2.1 Systemabgrenzung und Umfeldanalyse



Auf Basis dieser Systembetrachtung kann die so genannte Systemabgrenzung vorgenommen werden, welche die Grenzen des zu Eingriffsbereiches im Projekt festzulegen hat.

Wie im ersten Teil des Buches aufgezeigt wurde, ließe sich eine solche Systemabgrenzung grundsätzlich für verschiedene Systemaspekte vornehmen. So könnte die Systemgrenze aufgrund der tangierten Geschäftsprozesse gezogen werden oder die Grenze wird bezogen auf die betrachteten Organisationseinheiten festgelegt.

Die saubere Systemabgrenzung mit der Identifikation der Umsysteme und Visualisierung der Beziehungen zu den Umsystemen entspricht dem Kontextmodell des Projektes. Dieses wird später dazu dienen, die Vollständigkeit der Prozesse zu hinterfragen. Jede Beziehung zu einem Umsystem muss ja eindeutig durch einen Geschäftsprozess "aufgefangen" werden.

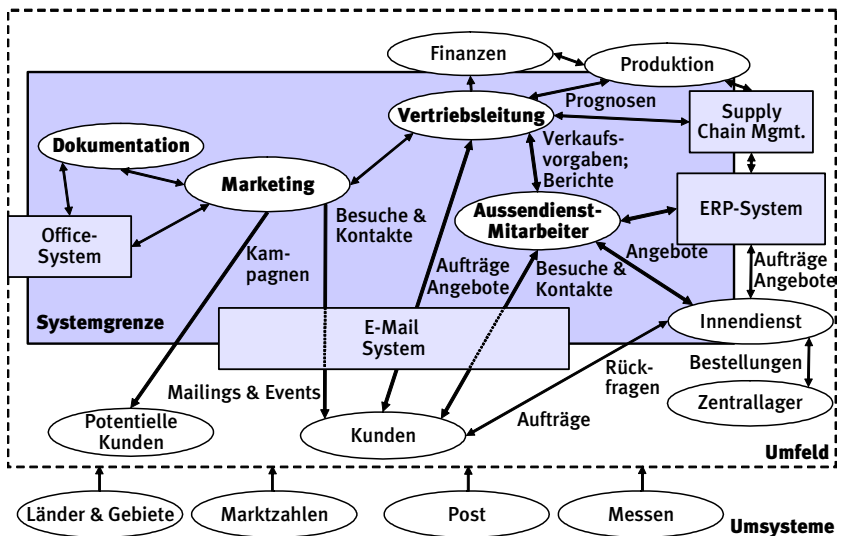


Abb. 129: Systemabgrenzung und Umfeldanalyse

### 10.2.2 Ist-Prozesslandkarte

Im Anschluss an die Erarbeitung der Systemabgrenzung und Umfeldanalyse sollte eine Prozessbetrachtung auf hoher Systemebene vorgenommen werden. Dies erfolgt mit Hilfe der Prozesslandkarte, welche alle Geschäftsprozesse des Systems identifiziert. Die unten stehende Darstellung zeigt eine beispielhafte Prozesslandkarte für den Bereich Marketing und Vertrieb unseres CRM-Beispiels der Fahrradfirma (s. Abb. 130). Diese Prozesslandkarte wird im weiteren Projektverlauf als Ausgangspunkt für diverse Analysen dienen.

In der Praxis stellt man leider häufig fest, dass den Unternehmen – trotz ISO- und weiterer Qualitätsanstrengungen – die Geschäftsprozesse nicht bekannt sind oder nicht in der gewünschten Form vorliegen. Man trifft daher auch nur in wenigen Fällen auf ein breites Verständnis, was unter dem Begriff “Prozess” bzw. “Geschäftsprozess” zu verstehen ist.

Die Erfahrung zeigt, dass sich die Geschäftsprozesse verschiedener Unternehmen auf dieser obersten Ebene nicht sehr stark voneinander unterscheiden. Die Prozessübersicht gibt auch bereits implizite Hinweise auf die Hauptfunktionen und die Geschäftsobjekte.

Jedem Geschäftsprozess liegt in aller Regel ein zentrales Objekt – ein “Geschäftsobjekt” – und eine “Hauptfunktion” zugrunde (z.B. dem Eventprozess “Event”). Später im Konzept lassen sich in analoger Weise die Objekte aufgrund der Teilprozesse ableiten.

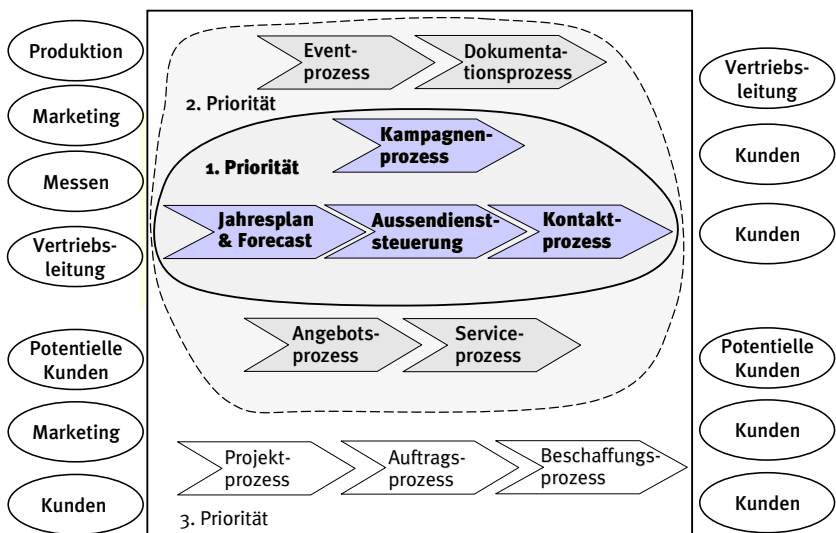


Abb. 130: Beispielhafte Prozesslandkarte

### 10.2.3 Ist-Organisation

Der nächste Schritt des Grobkonzeptes besteht in der Analyse des bestehenden Organisationsmodells (s. Abb. 131). Im unten stehenden Organisationsmodell (Organigramm) ist ergänzend vermerkt, wie viele Personen einer bestimmten Stelle angehören (Zahl rechts oben).

Das Organisationsmodell eines Unternehmens liegt, im Gegensatz zum Prozessmodell, meist schon vor. Für die Aufgabenorganisation im Unternehmen ist in der Regel ebenfalls ein gutes Verständnis vorhanden. Man kann sich daher meist recht rasch einen Überblick über die diesbezügliche Situation im Unternehmen verschaffen. Zunehmend trifft man allerdings auf den Umstand, dass das Organigramm entweder nicht mehr aktuell ist bzw. der tatsächlichen Aufgabenverteilung innerhalb des Unternehmens nicht mehr entspricht. Dies ist ein Hinweis auf die Schnelllebigkeit der Systemstrukturen und rechtfertigt durchaus kritische Fragen hinsichtlich der Auswirkungen solch stetiger Änderungen auf die Ist-Struktur.

Das Organigramm lässt eine erste Beurteilung der Aufgabenteilung im Unternehmen zu. Im Vordergrund steht zunächst die Frage, ob es sich um eine funktionale oder divisionale Organisation handelt. Im vorliegenden Beispiel ist eine gemischte Struktur gegeben. Der Bereich Spezialfahräder bildet eine eigene "Division", während der Bereich der Standardräder funktional organisiert ist. Diese Betrachtung wird eine wichtige Rolle spielen, wenn es später um die Grobkonzeption der Systemarchitektur geht.

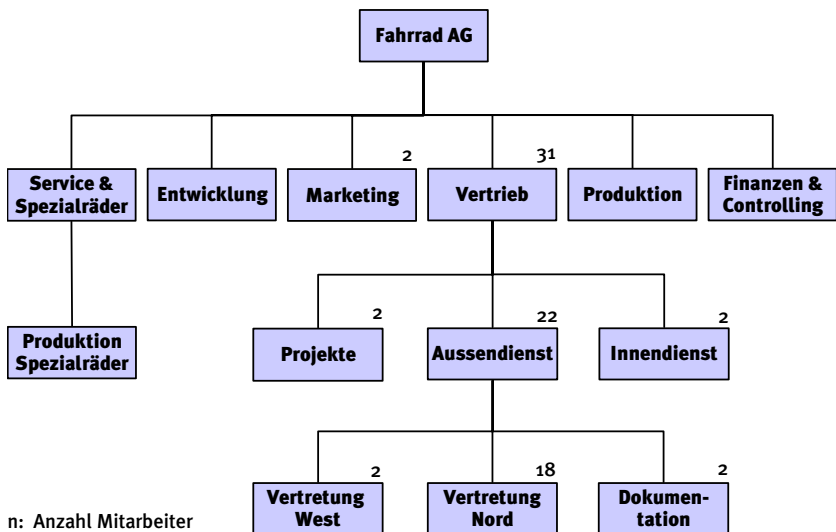


Abb. 131: Organisationsmodell

#### 10.2.4 Stellenorientierter Ist-Informationsfluss

Aufbauend auf dem Organisationsmodell und dem bereits erarbeiteten Prozessmodell könnte nun fallweise in einem weiteren Schritt das Diagramm des “stellenorientierten Informationsflusses” (s. Abschnitt 6.6) erarbeitet werden (s. Abb. 132). Es verknüpft die Organisation mit den Geschäftsprozessen. Man erhält einen Überblick über die Kommunikationsbeziehungen im Unternehmen bzw. im betrachteten System.

Erfahrungsgemäß entsteht eine Vielzahl von Problemen in einem Unternehmen an den innerbetrieblichen Organisationsgrenzen, d.h. dort, wo ein Prozess eine organisatorische Einheit verlässt und von einer anderen Stelle aufgenommen wird. Hier wird regelmäßig ein Informationsverlust zu erwarten sein. Zusätzlich ist damit zu rechnen, dass der Prozess nicht unmittelbar weiterbearbeitet wird oder Aufträge “liegen bleiben”.

Solche Probleme benennt man auch “Schwankungen und Störungen” (s. Abb. 132). Es hat sich als aufschlussreich erwiesen, die Schwankungen und Störungen einer Organisation den Schnittstellen zwischen Organisationseinheiten zuzuweisen. Dies entspricht nun nichts anderem als der Darstellung des “stellenorientierten Informationsflusses”. Er zeigt primär die Aufgaben und sekundär die Prozesse. Das Diagramm eignet sich aus diesem Grunde ideal zur Lokalisierung von Problemen im Auftragsdurchlauf. Es könnte weiter vermerkt werden, wie häufig die einzelnen Schwankungen und Störungen vorkommen (täglich, wöchentlich, etc.).

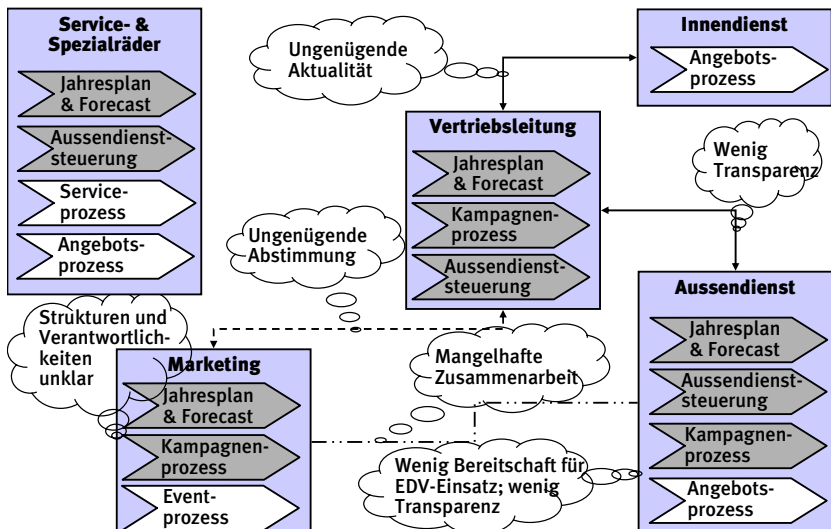


Abb. 132: Schwankungen und Störungen im Bereich Marketing und Vertrieb

### 10.2.5 Ist-Systemarchitektur

Zur vollständigen Situationsanalyse im Rahmen des Grobkonzeptes gehört die Analyse des bestehenden Technikeinsatzes – insbesondere der vorhandenen Informationstechnik und Anwendungssysteme. Zunehmend werden in Informatikprojekten ja auch bestehende Systeme abgelöst oder ergänzt. Daher ist es unabdingbar, auch wirklich genaue Kenntnis vom erreichten Technikumfang zu haben. Es ist bei dieser Analyse zu beachten, dass gelegentlich Systeme nur unvollständig genutzt werden und die Analyse des Nutzungsgrads zusätzlichen Aufschluss geben könnte.

Gewisse Modelle wurden bereits zuvor erarbeitet und können nun für die nachfolgenden Diagramme der Techniksicht als Grundlage dienen. Zunächst wird auch in der Techniksicht das Grundmodell, das heißt das bestehende Systemarchitekturmodell, die Voraussetzungen für die weiteren Analysen und Konzepte schaffen. Es zeigt auf, wie die Gesamtsystemarchitektur und die Subsysteme strukturiert sind (s. Abschnitt 7.3).

Unter Umständen werden in einem zweiten Schritt dann auch die bestehenden Systemschnittstellen betrachtet werden müssen – insbesondere dann, wenn die sozio-technische Konzeption eine Vielzahl von Schnittstellen erfordert (s. Abb. 133). Im Beispiel bestehen nur wenige Schnittstellen zwischen den einzelnen Anwendungssystemen. Das zentrale ERP-System hat als integrierte Lösung lediglich eine Schnittstelle zum Lohnsystem.

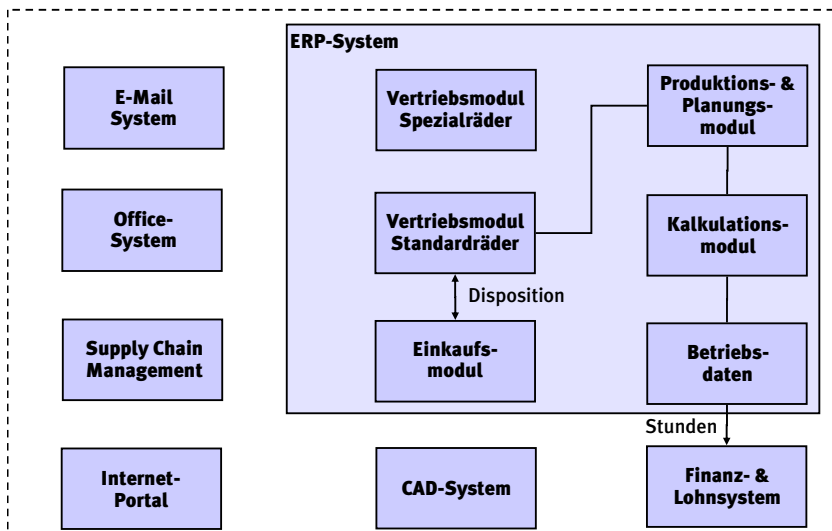


Abb. 133: Analyse der bestehenden Anwendungssysteme und -schnittstellen

### 10.2.6 Analyse der Informatikgeschichte

Mit dem Einsatz von Informationssystemen, seien dies nun CRM-, CAD-, BDE- oder ERP-Systeme (Enterprise Resource Planning), sind auch vielfältige emotionale Aspekte verbunden. Sowohl Mitarbeiter als auch Projektverantwortliche sind unter Umständen mit den bestehenden Systemen “emotional” verbunden, da sie viel Arbeitszeit in deren Einsatz investiert haben. Andererseits können, bedingt durch die Komplexität von Informatikprojekten, auch sehr negative Erfahrungen mit solchen Projekten verbunden sein.

Daher empfiehlt es sich, über die “sozio-technische Geschichte” des Unternehmens, bezogen auf den bestehenden oder den früheren Informatikeinsatz, etwas in Erfahrung zu bringen. Es kann nämlich erwartet werden, dass sich gewisse Erfahrungen (Budgetüberschreitungen, Terminverzögerungen, Stress, Kündigungen, etc.) als Verhaltensmuster – sowohl positiver wie auch negativer Natur – auf das Projekt auswirken könnten.

Häufig ist auch zu beobachten, dass im Zuge einer Systemablösung neue, allenfalls jüngere Mitarbeiter verantwortliche Funktionen im Zusammenhang mit der Informatik einnehmen, Funktionen, die nun altgedienten Mitarbeitern abhanden kommen. Diese werden gewissermaßen in die bestehende Informatik zurückversetzt und bringen Unruhe in den Betrieb.

### 10.2.7 Mengen und Häufigkeiten – Schätzung der Zukunftsentwicklung

Ein wichtiger Punkt eines Grobkonzeptes besteht in der Erarbeitung eines Mengengerüsts. Dieses gibt einerseits erste Hinweise auf die Geschäftsobjekte des Systems und ermöglicht andererseits ein besseres Verständnis des Geschäfts. In unserem Beispiel:

#### **Außendienst**

1200 Außendienstbesuche; 2 600 000 km Spesenkilometer

#### **Marketing**

16 Mailings mit durchschnittlich 1300 Adressen

#### **Organisation**

20 Mitarbeiter, davon 12 Außendienstmitarbeiter und 1 Systemsupport

#### **Geschäftsobjekte**

4000 aktive Adressen; 12 000 inaktive Adressen; 3000 Kunden

#### **Informationstechnik**

34 Bildschirmarbeitsplätze, davon 12 Laptops



### 10.2.8 Stärken- und Schwächenanalyse

Aufgrund der vorgenommenen Situationsanalysen wie beispielsweise auch der obigen Analyse der “Schwankungen und Störungen” ergeben sich meist eine Vielzahl von Stärken und Schwächen im betrachteten System. Diese Stärken und Schwächen werden nun zusammengefasst, bewertet und tabellarisch aufgelistet (s. Abb. 135).

Es hat sich als sinnvoll erwiesen, die Schwächen zusätzlich einer Ursachen-/Wirkungsanalyse zu unterziehen. Oftmals ist ja eine einzige Schwachstelle für eine Vielzahl anderer Schwächen ursächlich verantwortlich. So könnte beispielsweise eine “fehlende Marketingleitung” als Grund für die mangelhafte Marketingplanung, ungenügende Events, unkoordinierte Marketingaktionen verantwortlich sein. In einem solchen Fall nützt auch die Beschaffung des besten Informationssystems nichts.

Interessante Ergebnisse liefert auch die Klassifikation nach Prozess-, Organisations-, Qualifikations- und Technikschwächen. Eine solche Zuordnung hat den Vorteil, dass ersichtlich wird, welches die Ursachen der bestehenden Unzulänglichkeiten sind.

Unter Umständen wird man durch diese Klassifikation ebenfalls erkennen, welche Schwächen sich nicht durch die Einführung eines neuen Informationssystems beheben lassen werden. Vielfach liegen gravierende Schwächen wie erwähnt in der Aufbauorganisation oder dem ungenügenden Ausbildungsstand und der Qualifikation der Mitarbeiter.

Faktor	--	-	o	+	++	Begründung
Marketing-unterlagen					X	Die Marketingunterlagen sind optisch und inhaltlich ansprechend und neueren Datums.
Marktpräsenz					X	Das Unternehmen ist im Fahrradmarkt seit langem gut etabliert.
Vertrieb				X		Der Vertrieb hat ein gutes Know-how und gute persönliche Beziehungen zu Kunden.
Fehlende Marketingleitung	X					Zahlreiche Probleme bezüglich Absatzmengen, Schnittstellen, Konditionen, Marktaufreten
Zuwenig Events	X					Kompetent, teuer, teilweise arrogant
Kontakte		X				Kundenkontakte sind nicht dokumentiert
Kommunikation		X				Geringer Kontakt zwischen Verkauf und Marketing.
Direktmarketing		X				Ungenügende Präsenz auf dem Markt durch Direktmarketing.
Unkoordinierte Aktionen	X					Jeder Bereich macht, was er will.

Abb. 134: Stärken- und Schwächenanalyse

### 10.2.9 Chancen- und Gefahrenanalyse

Bevor mit dem Zielfindungs- und Lösungsfindungsprozess begonnen wird, sollten die für das zu betrachtende System relevanten Zukunftsprognosen analysiert werden. Die zu erarbeitende Lösung soll ja für die Zukunft gebaut werden. Man wird sich über den Markt, die Konkurrenz, die Veränderungen in der Technologie und der Volkswirtschaft Gedanken machen müssen.

Aus den Veränderungen des Umfeldes und des Marktes ergeben sich für das bestehende System sodann Chancen und Gefahren für die Zukunft. Chancen und Gefahren sind also stets Veränderungen im Umfeld und *nicht Veränderungen im Unternehmen*.

Diese Veränderungen sind unter dem Gesichtspunkt der vorhandenen Stärken und Schwächen des Unternehmens zu analysieren (SWOT-Analyse: strengths, weaknesses, opportunities, threats). Chancen könnten sich je nach Unternehmen potentiell mit Stärken decken oder aber Schwächen mit Gefahren. Im Rahmen des Grobkonzeptes muss auf diese Fragestellung ein spezielles Augenmerk gerichtet werden (s. Abb. 135).

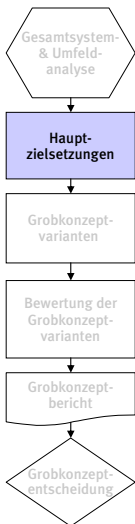
Es könnte beispielsweise eine Gefahr oder aber auch eine Chance darstellen, dass sich in den neuen europäischen Mitgliedstaaten neue Märkte eröffnen. Im Beispielunternehmen wird diese Entwicklung als Chance betrachtet. Das neue System sollte Chancen unterstützen und Gefahren abwenden.

Stärken	Chancen
Gute Marketingunterlagen	Kaufkraftsteigerung in neuen EU-Mitgliedsstaaten
Gute Marktpräsenz und starke Absatzorganisation	Erhöhung der Umsatzerwartung in den bestehenden Märkten
Starke Vertriebsorganisation	...
Schwächen	Gefahren
Fehlende Marketingleitung	Professionalisierung der ausländischen Konkurrenz auf dem Markt.
Zuwenige Events	Zunehmende Popularität von Messen und Grossanlässen bei Zielgruppe
Ungenügendes Kontaktmanagement	Informationsflut wird stetig zunehmen. Jeder persönliche Kontakt wird wichtig.
Schwaches Direktmarketing	...

Abb. 135: Chancen- und Gefahrenanalyse

## 10.3

## Hauptzielsetzungen



Die Zielsetzungen an das zu gestaltende System lassen sich primär aus der Situationsanalyse und insbesondere aus den Schwächen und Gefahren ableiten. Die Erstellung des Zielkataloges dient der Konsolidierung und Vervollständigung dieser Erkenntnisse (s. Abb. 136). Jedes Ziel im Zielkatalog muss mit einer Zieleigenschaft und dem Ausmaß (Messgröße) versehen werden, welche das Messkriterium festlegt.

Für die spätere Auswahl der besten Lösung ist die Unterscheidung zwischen Muss- (M) und Wunschzielen (W) methodisch von großer Bedeutung. Mussziele geben dabei ein hartes Selektionskriterium vor. Erfüllt eine Lösung dieses Kriterium nicht, dann fällt diese außer Betracht. Optimierungsziele beinhalten keine derart scharfen K.O.-Kriterien. Sie müssen einfach möglichst gut erfüllt werden. Es ist zu empfehlen, die Hauptzielsetzungen vom Auftraggeber nochmals formell absegnen zu lassen.

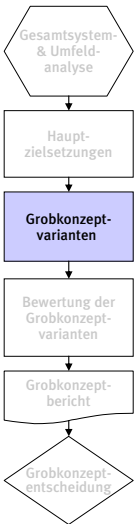
Im Rahmen der Zielsetzung sollten weiter so genannte Systemziele und Vorgehensziele wirklich klar unterschieden werden. Vorgehensziele beziehen sich auf den Prozess der Projektabwicklung und könnten z.B. heißen: kurze Dauer des Projektes, effizienter Personaleinsatz und kostengünstige Entwicklung. Die Systemziele beziehen sich auf das zu gestaltende System, wie z.B.: Effizienz des Marketings, geringe Wartungskosten, Benutzungsfreundlichkeit, Autonomie der Abteilungen.

Zielklassen	Ziele	Zieleigenschaften	Ausmass	Zielart/ Priorität	Zeitpunkt
Marketing-Ziele	Effektive Kampagnen	Aufwand für Einzelkampagne	<= 0.5 Tage	Mussziel Prio. 1	nach Einführung
	Transparenz der Kampagnen	Zeitaufwand Identifikation	<= 5 Sek.	Mussziel Prio. 3	pro Jahr
	...	...	...	...	...
Organisations-Ziele	Dezentraler Aussendienst	Anzahl Bereiche	>= 5 Bereiche	Mussziel Prio. 2	Betrieb nach 1 Jahr
	...	...	...	...	...
Ressourcen-Ziele	Betrieb mit geringem Zusatzpersonal	Personal Betrieb	Stellenprozentage	Prio. 5	pro Jahr
	...	...	...	...	...
Technische Ziele	Integration in ERP-Lösung	Schnittstelle zu XYZ	gegeben	Mussziel Prio. 4	nach Einführung
	Kalender	Schnittstelle		Prio. 6	...

Abb. 136: Zielkatalog

## 10.4 Grobkonzeptvarianten

### 10.4.1 Sozio-technischer Systemansatz



Nach der Gesamtsystem- und Umfeldanalyse sowie der Erarbeitung der Zielsetzungen beginnt die kreative Phase der Grobkonzeption neuer Lösungen. Die Konzeption von technikgestützten Arbeitssystemen war bereits zu Beginn des letzten Jahrhunderts Gegenstand der Forschung. Man erkannte allerdings erst spät, dass der Einsatz neuer und leistungsfähiger technischer Hilfsmittel in vielleicht bis dahin recht gut funktionierenden Organisationseinheiten zu neuen Problemen führen kann.

Von besonderer Bedeutung war die Erkenntnis, dass eine Organisationseinheit zur Erbringung ihrer Leistung neben guter Technik und effizienter Prozesse auch über eine gut funktionierende Organisation (soziales Teilsystem) verfügen muss. Diesbezügliche Studien im englischen Kohlebergbau führten zu Beginn der 50er-Jahre zum Begriff des sozio-technischen Systems (Trist und Bamforth 1951). Trist und Bamforth konnten zeigen, dass Taylors Ansatz, die Leistung des einzelnen Individuums durch technische und arbeitsorganisatorische Mittel zu optimieren, im genannten Beispiel der Kohleförderung für die Gesamtleistung der Gruppe nicht optimal war, die "Optimierung einzelner Elemente" also kein Gesamtoptimum ergeben muss.

Ein wesentliches Fazit des sozio-technischen Ansatzes betrifft damit die Organisation von Tätigkeiten in Gruppen, wonach eine gut "funktionierende" und kommunizierende Gruppe wesentlich bessere Leistungen erbringt als eine nicht funktionierende Gruppe mit demotivierten Mitarbeitern, die aber mit neuesten Technologien ausgestattet sind.<sup>7</sup> Gemäß dem sozio-technischen Systemansatz muss also die Konzeption des technischen Systems gemeinsam mit der organisatorischen Konzeption und eventuellen Qualifikationsmaßnahmen erfolgen (Ulich 2001). Diese Prinzipien zur sozio-technischen Systemgestaltung bilden eine auch für Informationssysteme gültige Gestaltungsrichtlinie.

Wir werden uns nun mit der Frage beschäftigen, wie die oben beschriebenen Spezifikationen zu einem möglichen ganzheitlichen Konzept integriert werden können. Es erscheint an dieser Stelle wichtig zu erwähnen, dass es aus unserer Sicht nicht die alleinig richtige Methode geben kann und soll. Vielmehr ist essentiell, dass man im Verlaufe eines Projektes alle Aspekte beleuchtet und diesen je nach Bedeutung und Umfang im Projekt den richtigen Stellenwert verleiht.

<sup>7</sup> s. u.a. Zuboff (1988), Nohria & Chalykoff (1990) gem. Davenport (1993, S. 98).

### 10.4.2 Möglichst breite sozio-technische Lösungssuche

Es ist von großer Bedeutung in dieser frühen Phase der Systemgestaltung, dass man sich nicht in Gefahr begibt, verfrüht irrelevante Detailfragen und technischen Details zu klären (s. Abb. 137).

Vielmehr sollte ganz zu Beginn der Lösungssuche ein möglichst breiter Fächer an fundamental unterschiedlichen organisatorischen und technischen Grobkonzeptvarianten erarbeitet werden. Es kann zwar sein, dass tatsächlich die nahe liegende Lösung auch die beste Lösung ist – häufig wird dies aber gerade nicht der Fall sein. Eine gute Strategie besteht darin, je Hauptziel eine optimale und eigenständige Lösung zu erarbeiten. Man wird auf diese Weise unterschiedliche Lösungen finden. Erst anschließend versucht man, diese Lösungen zu vereinen und das Beste von jedem Lösungsansatz zu nehmen. Es ist dies das Prinzip des aufgeschobenen Urteils.

*Im Beispiel des Kurortes wäre nun zu prüfen, ob mehrere Sessellifte, eine große Luftseilbahn oder eine Standseilbahn ideal wären.*

Bei Informationssystemen bietet es sich auch an, zunächst zu überlegen, ob bereits verfügbare Systeme ausgebaut werden können. Alternativ zum Ausbau bestehender System müsste auch ein dediziertes Neusystem geprüft werden. Die Variantensuche muss sich ergänzend auf neue organisatorische Lösungsvarianten ausrichten.

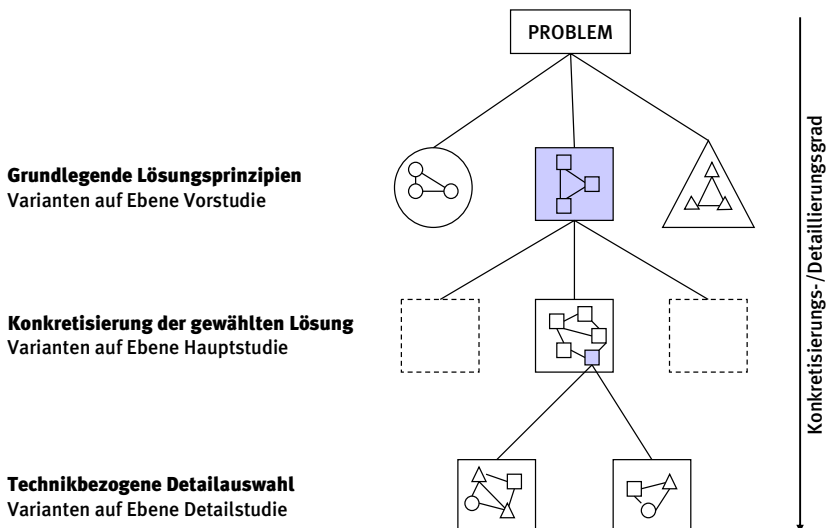


Abb. 137: Suche nach unterschiedlichen Grobkonzeptvarianten

### 10.4.3 Ideensuche mit Hilfe von Brainstorming

Das Grobkonzept kann aber beispielsweise auch ausgehend von Phantasien und Ideen eingeleitet werden, die im Rahmen von partizipativen “Zukunftswerkstätten” entstanden sind. Ein bewährter Ansatz besteht darin, die Ideen wie folgt zu klassifizieren:

- Ideen zur verbesserten Prozessgestaltung (Prozesse)
- Ideen zur Aufbauorganisation (Aufgaben)
- Ideen betreffend Qualifikationsmassnahmen (Mensch)
- Ideen zur Informationstechnik (Technik)

Die Phantasien und Grobkonzeptvorschläge einer Zukunftswerkstatt können ebenfalls von unterschiedlichen Personen bewertet und gewichtet werden (s. Abb. 138). Solcherart entstandene Ideen bilden einen gewissen Fundus zur Gestaltung des Grobkonzeptes.

Bei der Anwendung von Zukunftswerkstätten fällt auf, dass auch in Informatikprojekten das größte Veränderungspotenzial häufig nicht unbedingt in der Technik gesehen wird, sondern in rein organisatorischen bzw. Qualifikationsmassnahmen. Obwohl sich das eine vom anderen nicht scharf trennen lässt, ist diese Feststellung zumindest ein Zeichen, dass der Glaube an die Informationstechnologie nicht in den Himmel wachsen sollte. Es bestätigt noch einmal die Regel, dass Organisation vor Technik kommt, sowohl von der Bedeutung her gesehen als auch zeitlich im Projektablauf.

Gebiet	Gewichtung	Idee
Mensch	••••	Qualifizierung in Marketing
	••	Schulung
Technik	••	Innovatives Informationssystem
	•	Kausale Zusammenhänge abbilden
	•	Prozesssicherheit
Organisation	••••	Stelle für Marketingsupport
	••	Kompetenzen für Außendienst
	•••	Bessere Lagerstrategie
Prozess	•••••	Direkte Selbstbestellung Vertrieb
	••••	Effiziente Außendienst-Steuerung
	••	Nur Rahmenaufträge
	•	Produktionszeit = Lieferzeit
	•	Kundenauftragsbezogene Produktion

Abb. 138: Kategorisierte und klassifizierte Schwächen

#### 10.4.4 Grobkonzeption der Soll-Prozesslandkarte

Im Rahmen der Grobkonzeption ist an erster Stelle zu entscheiden, welche Geschäftsprozesse durch das künftige Informationssystem abgedeckt bzw. optimiert werden sollen. Man beginnt daher meist damit, die Prozesslandkarte als Ganzes zu betrachten und die Möglichkeiten der Neugestaltung aufgrund von Unternehmensstrategien zu prüfen (Österle 1995, Fleisch 2001). Es betrifft primär die Frage, welche Geschäftsprozesse zu optimieren, neu zu segmentieren, neu zu schaffen sind oder unter Umständen ausgelagert oder ganz eliminiert werden müssten. Dies erfolgt in Abstimmung mit der bestehenden und der geplanten Aufbauorganisation.

Das Grobkonzept der Prozessunterstützung in unserem Beispielprojekt sieht z.B. vor, dass die Jahres- und Kampagnenplanung bzw. der Eventprozess durch den Vertrieb und das Marketing *gemeinsam* ausgeführt und wahrgenommen werden soll (s. Abb. 139).

Entscheidend im Hinblick auf die spätere Implementation kann an dieser Stelle weiter eine Beurteilung sein, wie ähnlich sich recht analoge, indessen nicht gemeinsame Prozesse unterschiedlicher Stellen sind (grau). Die Fragestellung könnte also lauten: Wie ähnlich ist die Außendienststeuerung für den Bereich Spezialräder zur Außendienststeuerung der Standardräder. Benötigen die beiden Bereiche je eine eigenständige Implementation? Diese Prozesslandkarte liefert sodann erste Anhaltspunkte bezüglich der Funktionalitäten des künftigen Informationssystems.

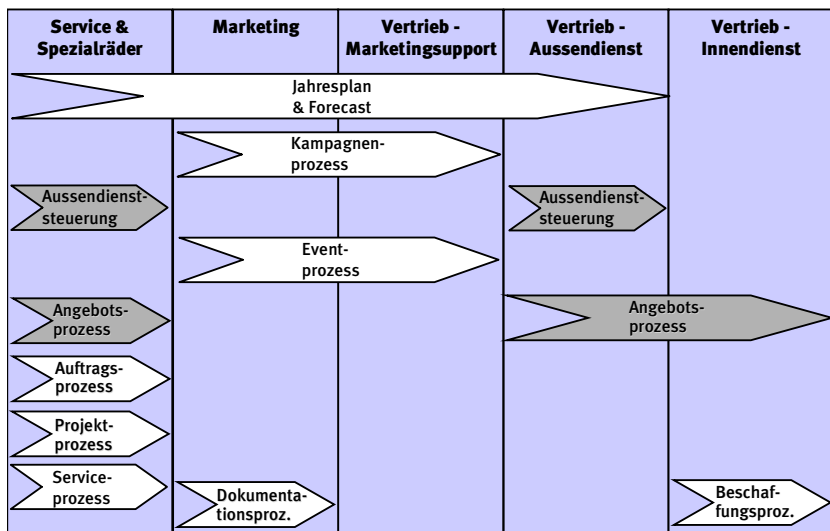


Abb. 139: Prozesslandkarte der CRM-unterstützten Marketingprozesse

### 10.4.5 Grobkonzeption der Soll-Organisation

Im Rahmen eines Grobkonzepts wird in Verbindung mit den Prozessen auch geklärt werden müssen, welche künftige Soll-Organisation durch das IT-System vorgesehen und unterstützt werden sollte. Eine eventuelle organisatorische Neukonzeption sollte primär aufgrund von Kriterien der Motivation erfolgen (Ulich 2001). Es wurde erkannt, dass relativ autonome Organisationseinheiten und der klare Bezug einer Organisationseinheit zu einem definierten Endprodukt die Motivation der Mitarbeiter zu erhöhen imstande ist.

In vielen IT-Projekten wird sich die Organisation auf Unternehmensebene allerdings nicht grundlegend ändern. Häufig wird die Optimierung einzelner Aufgaben im Vordergrund stehen. Anders verhält es sich allenfalls bei strategischen Gesamtprojekten, in denen eine gesamte Reorganisation in Verbindung mit einem neuen Informationssystem erforderlich wird.

Im unserem begrenzten CRM-Projektbeispiel sei erkannt worden, dass der Vertrieb über einen neue Abteilung "Marketingsupport" verfügen sollte; die Analyse der Schwankungen und Störungen habe dies aufgezeigt. Es hat daher direkte Veränderungen der Marketingorganisation zur Folge (s. Abb. 140). Diese aufbauorganisatorische Korrektur soll die Koordination von Kampagnen verbessern und den Vertriebsleiter entlasten. Bei der Neukonzeption der Marketingprozesse ist daher davon auszugehen, dass der Vertrieb über einen Marketingsupport verfügt.

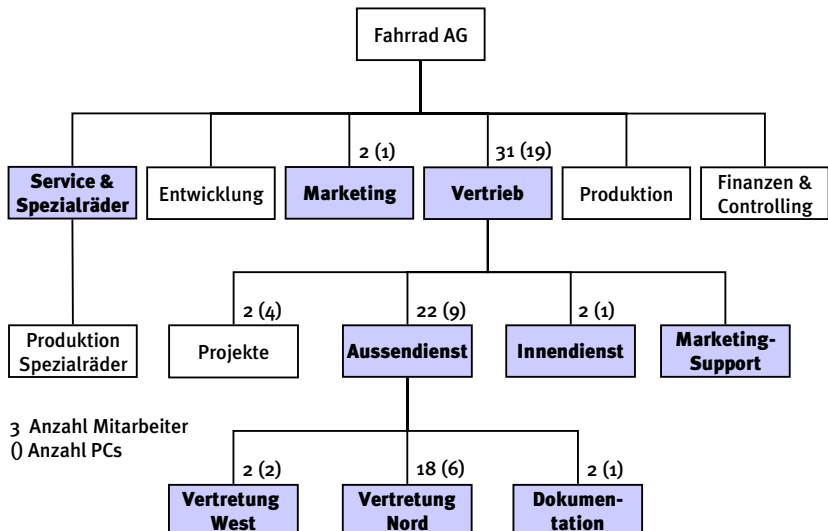


Abb. 140: Zukünftige Soll-Organisation



### 10.4.6 Grobkonzept der Soll-Systemarchitektur

In den vorangegangenen Abschnitten wurden Gestaltungsmöglichkeiten hinsichtlich innovativer Prozess- und Organisationskonzepte angesprochen. Aufbauend auf diesen organisatorischen Rahmenbedingungen wird man sich nun auch Gedanken zur Ausgestaltung des Informationssystems auf einer hohen Ebene der Systemarchitektur machen.

Im Grobkonzept muss man sich primär Überlegungen zur Gesamtarchitektur auf Stufe der Anwendungen machen. Dies betrifft beispielsweise die Frage, ob ein System zur Abdeckung des oben stehenden Prozessentwurfes, der angestrebten Organisation und der erforderlichen Systemfunktionalität ausreicht, oder ob allenfalls mehrere eigenständige Neusysteme notwendig wären (s. Abb. 141).

Im unten stehenden Beispiel werden die Varianten geprüft, zwei separate CRM-Systeme für die unterschiedlichen Unternehmensbereiche einzuführen. Nur falls die Unterschiede sehr groß sind oder die Bereiche zu eigenständig, müssen mehrere autonome Anwendungen ins Auge gefasst werden. Allenfalls kann dann immer noch dieselbe Software eingesetzt werden – mit verschiedenen Installationen.

Gelegentlich wird man diese Entscheidung auch noch offen lassen, um im Rahmen einer späteren Evaluation die unterschiedlichen Optionen zu prüfen.

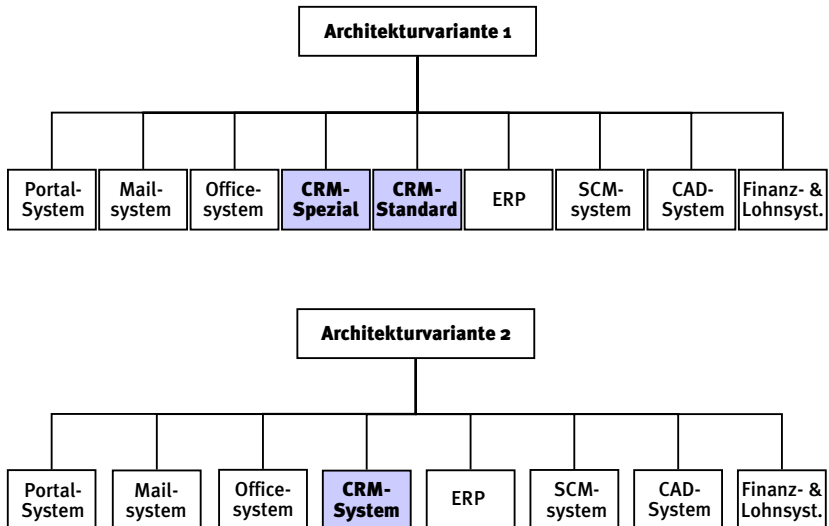


Abb. 141: Varianten der CRM-Architektur

#### 10.4.7 Integrierte Gesamtbetrachtung von Organisation und Systemarchitektur

Die vorangegangenen Konzeptvarianten für die Prozesse, die Organisation und die Technik bilden die Basis zur Neugestaltung eines integrierten sozio-technischen Gesamtkonzeptes, welches die organisatorischen und technischen Zielsetzungen abdeckt. Unumgänglich sind gewisse Kenntnisse über den Stand und die Möglichkeiten der Informationstechnologie (z.B. Produkte). Unter diesen gegebenen Rahmenbedingungen sind die verschiedenen innovativen Möglichkeiten und Ansätze von

- Grobkonzept der Prozesslandkarte
- Grobkonzept der Aufbauorganisation
- Grobkonzept der Systemarchitektur

zu prüfen und diese zu integrierten Gesamtsystemvarianten zu verschmelzen. Fallweise müssen auch die Hauptfunktionen und Geschäftsobjekte mitbetrachtet werden, insofern sie nicht in der Prozess- und Systemarchitekturbetrachtung mit enthalten sind.

Einen guten Überblick zur Veranschaulichung des Grobkonzeptes bietet das nachfolgende angeführte Technikeinsatzdiagramm (s. Abb. 142). Es veranschaulicht, welche Organisationseinheiten durch welche Techniksyste- me zukünftig unterstützt und allenfalls auch gemeinsam genutzt werden.

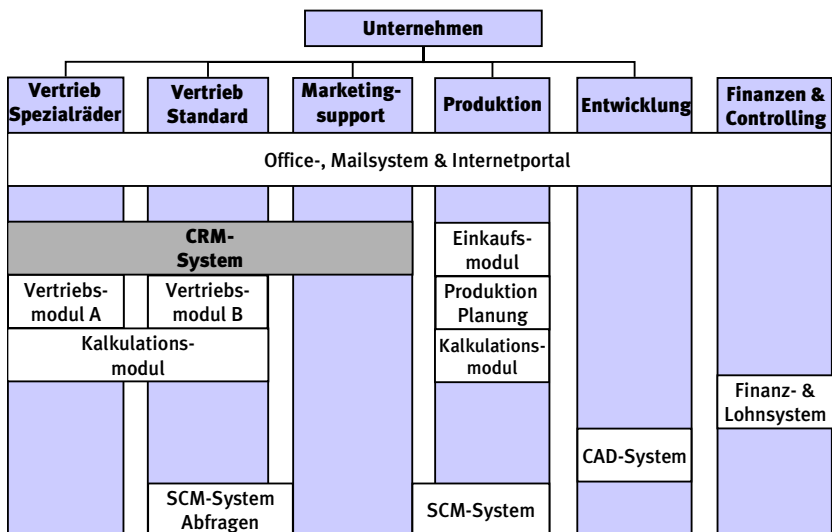


Abb. 142: Grobkonzept Technikeinsatz (Technikeinsatzdiagramm)

#### 10.4.8 Grobkonzeption Einführung und Schulung

Ein Teil der Grobkonzeption sollte sich bereits mit der Einführung und Schulung der Benutzer auseinandersetzen.

Die Grobkonzeption der Einführung beinhaltet eine mögliche Priorisierung der einzuführenden Funktionen und gibt eine praxisbezogene Überprüfung der gesetzten Hauptzielsetzungen.

Die Grobkonzeption der Schulung hilft dem Projektleiter, auch diesen Bereich frühzeitig anzusprechen und das Management der Anwendungsbereiche sukzessive darauf vorzubereiten, dass eine entsprechende Schulung notwendig sein wird.

Die Aufwändungen für die Schulung sollte seriös geschätzt werden und in die weiteren Überlegungen einfließen.

#### 10.4.9 Grobkonzeption Betrieb und Unterhalt

Eine häufige Schwäche bei der Grobkonzeption besteht darin, dass die Betriebs- und Unterhaltsbelange nicht oder nicht genügend berücksichtigt werden. Das Grobkonzept sollte bereits erste Vorstellungen entwickeln, wer für den späteren Betrieb verantwortlich ist und welche Stelle welche Rollen einnimmt (s. Kapitel 17).

Häufig ergeben sich durch den Einsatz neuer Technologien auch neue Stellenanforderungen. Nur so können auch die organisatorischen Rahmenbedingungen geschaffen und die Kosten ermittelt werden.

Die grundsätzliche Rollenaufteilung sollte eigentlich aus der Informatikstrategie oder der Beschreibung der Informatikorganisation hervorgehen. Trotzdem können die Vorstellungen über den späteren Betrieb insbesondere zwischen der zentralen Informatik und den Anwendungsbereichen auseinander gehen.

So wird die Informatikabteilung allenfalls davon ausgehen, dass der Anwendungssupport und die Betreuung der Anwender bei kleineren Änderungen durch die Anwendungsbereiche selbst übernommen werden wird. Diese stellen sich aber auf den Standpunkt, dass dies gerade Sache der zentralen Informatik wäre.

Solche Kontroversen sind zu verhindern, wenn sie bereits zum Gegenstand des Grobkonzeptes gemacht werden und die Diskussion frühzeitig initialisiert ist. Wird die Diskussion erst nach der Abnahme geführt, ist es eindeutig zu spät.

#### 10.4.10 Morphologische Matrix für das Grobkonzept

Eine leistungsfähige Methode zur systematischen Variantenbildung stellt die Technik der morphologischen Matrix dar (s. Abb. 143). Gemäß der Methodik der Morphologie werden alle möglichen Lösungsausprägungen eines Lösungsparameters in eine Tabelle eingetragen (Haberfellner 2002). In der Vertikalen befinden sich die Parameter und in der Horizontalen die zu jedem Parameter denkbaren Ausprägungen. Im Rahmen der vorangegangenen Abschnitte wurden bereits die möglichen Lösungsausprägungen für die Parameter Prozesslandkarte und Systemarchitektur angeführt.

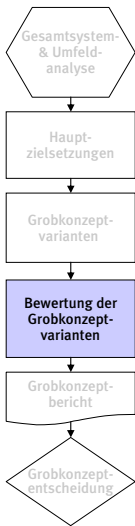
Auf der Basis dieser morphologischen Matrix können methodisch alle möglichen Grobkonzeptvarianten erarbeitet werden (s. Abb. 143). Jede Grobkonzeptvariante besteht im angegebenen Beispiel aus einem Variantenpfad. Firmenspezifische Parameter müssen selbstverständlich ergänzt werden. Nicht jede dieser Kombinationen macht Sinn. Es wird daher auch nicht notwendig sein, alle Varianten methodisch zu prüfen. Vielmehr wird man einige mögliche und insbesondere sinnvolle Kombinationen näher betrachten und die Zahl der in Betracht zu ziehenden Grobkonzepte auf drei bis fünf beschränken.

Der nachfolgende Pfad zeigt die Grobkonzeptvariante "CRM-Light", welche nur eine definierte Auswahl der Vertriebsprozesse abdeckt und davon ausgeht, dass eine gemeinsame Neuanwendung für alle Bereiche installiert wird.

<b>Ausprägung</b> <b>Parameter</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>
<b>Grobkonzept der Prozesslandkarte</b>	Abdeckung aller Prozesse	<b>Ausgewählte Prozesse</b>	Abdeckung von Teilprozessen	
<b>Systemarchitektur</b>	Einsatz mehrerer Neuanwendungen	<b>Einsatz einer Neuanwendung</b>	Erweiterung einer bestehenden SW-Anwendung	Nutzung bereits vorhandener SW-Funktionalität
<b>Organisatorische Autonomie</b>	<b>Eine Anwendung für alle Bereiche</b>	Gleiche Lösung, autonomer Einsatz	SW-Autonomie mit Austausch	Volle Autonomie je Bereich
<b>Softwaretyp</b>	<b>Standardsoftware</b>	Standardsoftware mit moderaten Anpassungen	Standardsoftware mit individuellen Erweiterungen	Individualsoftware
<b>Integrationsgrad</b>	Volle Integration mit Drittlösungen	<b>Abgleich mehrerer Datenobjekte</b>	Abgleich weniger Datenobjekte	Keine Integration
....	...	...	...	...

Abb. 143: Beispielhafte Morphologie zur Variantenbildung im Grobkonzept

## 10.5

**Bewertung der Grobkonzeptvarianten**

Es wurde bereits mehrfach erwähnt, dass im Rahmen der Grobkonzeption mehrere unterschiedliche Grobkonzeptvarianten erarbeitet werden sollten. Sie sollten idealerweise die volle Spannweite der möglichen Lösungsansätze hinsichtlich des Umfanges und der Systemintegration aufzeigen. In unserem Beispiel stehen die unten beschriebenen Varianten zur Auswahl (s. Abb. 144). Jede Variante ist machbar und hat ihre Vor- und Nachteile. Die erste Variante hätte den Vorteil, dass jeder Unternehmensbereich in der Beschaffung autonom agieren könnte. Die beiden anderen Varianten sehen eine unternehmensweite einheitliche Lösung vor.

Die drei Grobkonzeptvarianten werden nun in einem nächsten Schritt einer formellen Bewertung unterzogen. Die Methodik der Bewertung entspricht den bereits in Kapitel "Projektantrag" aufgezeigten Verfahren. Die Beurteilung umfasst also neben der Betrachtung der Wirtschaftlichkeit auch die strategischen Faktoren und die Risikoanalyse. Die Variante mehrerer autonomer SW-Lösungen für die von den Prozessen her doch sehr unterschiedlichen Unternehmensbereiche wurde verworfen. Es besteht natürlich nach wie vor die Möglichkeit, eine autonome Modulimplementierung im Rahmen der Konzeptphase zu prüfen.

*Beispiel: In unserem Kurortbeispiel wäre nun klar, dass die Luftseilbahn die beste Variante darstellt und die Grobkonzeptvarianten einer Zahnradbahn und eines Sesselliftes infolge der Topologie zu verwerfen sind.*

<b>Varianten Merkmale</b>	<b>Mehrere autonome SW-Lösungen</b>	<b>CRM-Light-Lösung</b>	<b>CRM-Gesamtlösung</b>
Beschreibung der Lösung	Die Bereiche Spezialfahrräder & Standardfahrräder erhalten eine eigene CRM-Lösung, welche optimal auf sie zugeschnitten ist.	Die Bereiche erhalten gemeinsame Lösung für Kampagnenprozess; Jahresplanung, Forecast; Aussendienst und Kontaktprozess.	Diese Lösung würde neben den Marketingprozessen auch die Angebots-, Auftrags- & Beschaffungsprozesse mitabdecken.
Vorteile	Hohe Autonomie der Bereiche – technisch-organisat. Konvergenz	Einheitliche Software ermöglicht Koordination; vernünftiger Umfang.	Gesamtlösung mit Schnittstellen reduziert Doppelspurigkeiten
Nachteile	Lösungsvielfalt ist unter Umständen nicht notwendig.	Angebots- und Auftragsprozesse sind entkoppelt.	Viele Schnittstellen und hohes Implementationsrisiko.
Investitionssumme	500	350	700
Betriebskosten p.J.	200	140	200
Kapitalwert (NPV)	140	140	60
Strategischer Wert	62	80	86
Risikozahl	22	35	45

Abb. 144: Bewertung der Grobkonzeptvarianten

### Bewertung der technisch-organisatorischen Konvergenz

Das gewählte Grobkonzept sollte weiter gewährleisten, dass eine unternehmensweite Akzeptanz in allen betroffenen Organisationsbereichen entstehen kann. Akzeptanzprobleme einzelner Unternehmensbereiche werden sich erfahrungsgemäß dann ergeben, wenn die Lösung für ausgewählte Bereiche optimiert ist und andere Bereiche vernachlässigt werden. Ein wichtiger Aspekt bei der Bewertung der Grobkonzeptvarianten liegt daher in der Bewertung der „technisch-organisatorischen Konvergenz“ (Ulich 2001). Die Konvergenz von Technik und Organisation lässt sich anhand des Vergleichs von Organigramm und Systemarchitektur beurteilen (s. Abb. 145). Vereinfacht ausgedrückt besagt dies, dass eine stark dezentrale, divisionale Organisation auch über eine dezentrale Systemarchitektur mit autonomen Anwendungssystemen verfügen sollte. Im abgebildeten Beispiel müssten die Divisionen je über eine eigene CRM-Anwendung verfügen. Häufig werden unterschiedliche Divisionen auch unterschiedliche Anforderungen aufweisen (z.B. Projekt- vs. Massengeschäft). Außerdem sind sie gewohnt, eigenverantwortlich zu handeln.

Die Vorteile der Autonomie sind mit den Nachteilen heterogener Systeme abzuwägen. Es könnten dies Nachteile bei der Beschaffung der Softwarelizenzen sein, Nachteile bei der Konsolidierung der Zahlen auf Geschäftsleitungsebene oder Nachteile beim Unterhalt der Systeme etc. Als Mittelweg könnte sich in gewissen Unternehmen eine autonome Projektierung und Konfiguration für die autonomen Unternehmensbereiche anbieten.

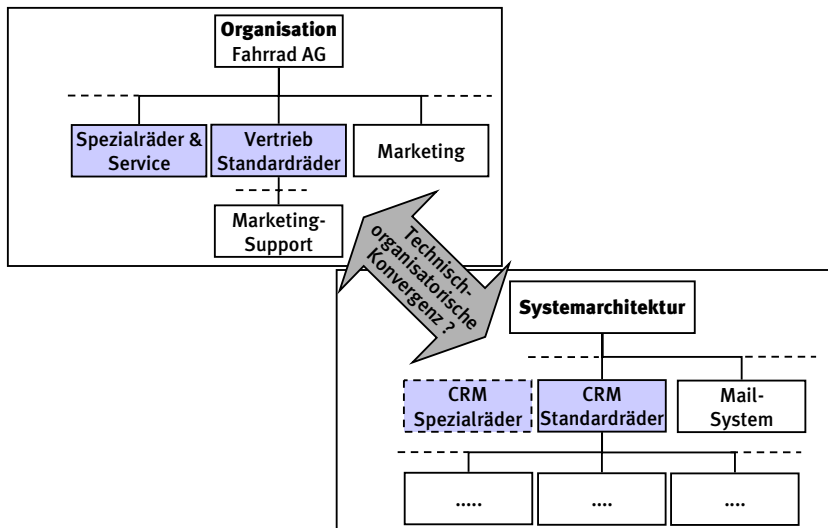
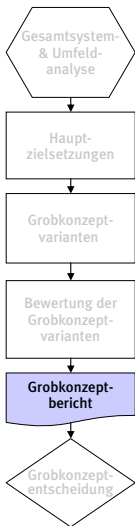


Abb. 145: *Technisch-organisatorische Konvergenz*

## 10.6

## Grobkonzeptbericht



Nach der Auswahl der besten Grobkonzeptvariante kann die gewählte Lösung näher in Form eines Grobkonzeptberichtes dokumentiert werden. Im Spiralmodell wird dieser Bericht kürzer ausfallen und allenfalls mit dem Konzept- und Detailkonzeptbericht verschmolzen werden (s. u.).

**I ZUSAMMENFASSUNG****II GROBKONZEPTION**

- **Ausgangslage, Problemstellung und Projektzielsetzung**

- **Gesamtsystem- und Umfeldanalyse**

Geschäftsprozesse, Organisation, Systemarchitekturen, Schnittstellen, Stärken und Schwächen, Chancen und Gefahren.

- **Hauptzielsetzungen**

- **Grobkonzeptvarianten**

Unterschiedliche Grobkonzeptvarianten bezüglich Prozessgestaltung, Funktionalität, Daten, Organisation, Systemarchitektur.

- **Bewertung und Auswahl**

**III BESCHREIBUNG DER GEWÄHLTEN GROBKONZEPTVARIANTE**

- **Sozio-technisches Grobkonzept**

Geschäftsprozesse, Systemfunktionen, Datenorganisation, Organisation, Systemarchitektur, Qualifikation.

- **Technologie und Machbarkeitsbegründung**

Eingesetzte Informationstechnologie; Begründung durch Success-Stories von ähnlichen Konzepten, Hardware, Software, Netzwerke

- **Grobkonzept Betrieb und Unterhalt**

Betriebsorganisation, Support, Unterhalt, Erweiterungen.

- **Grobkonzept Einführung und Schulung**

Einführungsplan, Roll-out-Ansatz, Schulungskonzept und -inhalte.

- **Wirtschaftlichkeitsbetrachtung des Grobkonzeptes**

Einmalige Kosten, wiederkehrende jährliche Kosten, Ressourcenbedarf, quantifizierbarer Nutzen, Wirtschaftlichkeitszahlen wie ROI, Pay-back oder ähnliche Angabe.

- **Risikoanalyse des Grobkonzeptes**

Organisatorische Risiken, Ressourcenverfügbarkeit, Technische Risiken, Risiken der Benutzerakzeptanz, wirtschaftliche Risiken.

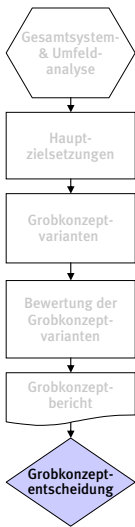
- **Terminierung und Etappierung des weiteren Projektverlaufes**

Meilensteine des Projektes, Zwischenetappen und Zeitplanung.

- **Projektmanagement und Projektorganisation**

Projektorganisation und -funktionen, Controlling, etc.

## 10.7 Grobkonzeptentscheidung



Die nachfolgende Ergebnisübersicht zeigt, dass im Rahmen der Systementwicklung Ergebnisse erarbeitet werden, welche der Situationsanalyse der Zielsetzung, der Lösungssuche bzw. der Bewertung zuzuordnen sind (s. Abb. 146). Es ist aus dieser Übersicht erkennbar, dass auch ein Einführungs- und Betriebsgrobkonzept zu erarbeiten ist. Diese Ergebnisse sind unabhängig davon, ob das Grobkonzept im Rahmen eines Wasserfall- oder Spiralmodells erarbeitet wurden. Wir sprechen daher auch bewusst von Grobkonzepttätigkeit – und nicht von einer Grobkonzeptphase. Als Hautergebnis steht nach einem Grobkonzept fest, welches Lösungsprinzip weiter verfolgt werden soll. In unserem CRM-Beispielprojekt schnitt die Variante “CRM-Light” als beste Lösung ab. Sie sieht eine unternehmensweite, gemeinsame SW-Lösung vor und deckt primär den Kampagnen- und den Außendienstprozess ab. Funktionalitäten zur Projekt- & Angebotserstellung sollen aus dem Projekt ausgeklammert werden.

Im nächsten Schritt der Konzeption wird es darum gehen, die Prozesse des Kampagnenmanagements und der Außendienststeuerung näher zu analysieren und das System mit seinen Zusammenhängen zu konzipieren.

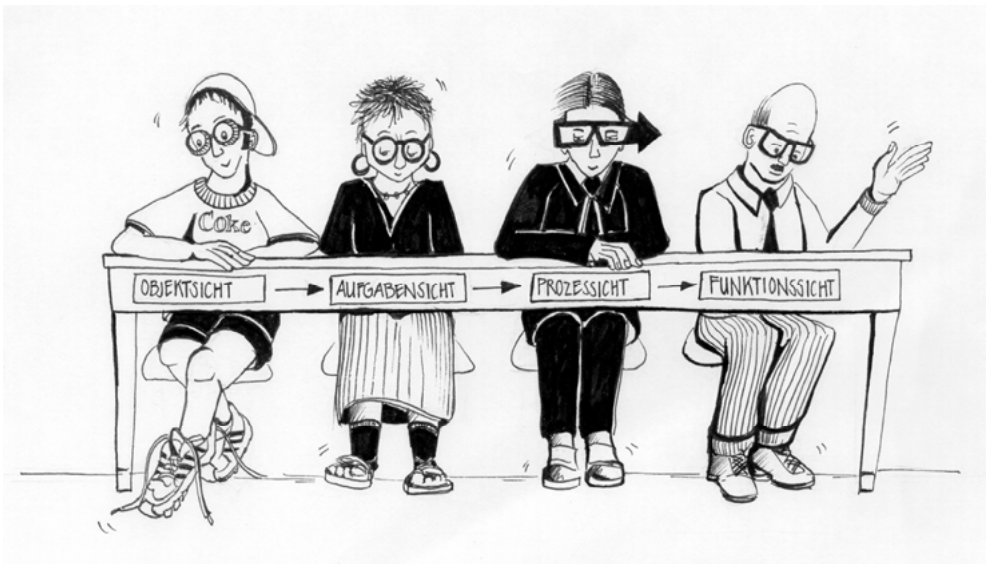
Bereich Schritt	Sozio-technisches System	Einführung & Ausbildung	Betrieb & Unterhalt	Projektmanagement & Controlling
<b>Gesamtsystem- &amp; Umfeldanalyse</b>	Gesamtsystem- & Umfeldanalyse; Systemabgrenzung; Prozesslandkarte; Geschäftsobjekte; Organisation; Bestehende Systemanwendungen; Vorhandene Systemschnittstellen; Stärken/Schwächen- & Chancen/Gefahren-Analyse; Mengen & Häufigkeiten; Zukunftsanalyse.		Grobanalyse von Betrieb & Unterhalt.	Projektstart; Projektplanung; Projektcontrolling.
<b>Hauptzielsetzungen</b>	Hauptzielsetzungen; Prioritäten; Zielkatalog.	Einführungs- & Ausbildungsziele.	Hauptziele an Betrieb & Unterhalt; Sicherheitsaspekte.	
<b>Grobkonzeptvarianten</b>	Grobkonzepte Prozesslandkarte; Grobkonzepte Funktionalität; Grobkonzepte Geschäftsobjekte; Grobkonzepte Organisation; Systemanwendungen; Systemschnittstellen; Risikokatalog; Machbarkeitsbeurteilung; Wirtschaftlichkeit der Varianten.	Grobkonzepte für Einführung & Ausbildung.	Grobkonzepte für Betrieb & Unterhalt.	
<b>Bewertung der Grobkonzeptvar.</b>	Bewertung Grobkonzeptvarianten; Kosten, Nutzen, Wirtschaftlichkeit.			
<b>Grobkonzeptbericht</b>	Grobkonzeptbericht.			
<b>Grobkonzeptentscheidung</b>				Entscheidung Grobkonzept.

Abb. 146 Ergebnisübersicht der Grobkonzepttätigkeit



## 10.8 Literatur zum Grobkonzept

- Blessing, D. (2002): Wissensmanagement in Beratungsunternehmen. Nordestedt: BoD.
- Daenzer, W. (Hrsg.) (1976): Systems Engineering: Leitfaden zur methodischen Durchführung umfangreicher Planungsvorhaben. Zürich: Verlag Industrielle Organisation.
- Davenport, T. (1993): Process Innovation: Reengineering Work through Information Technology. Boston, MA: Harvard Business School Press.
- Fleisch, E. (2001): Das Netzwerkunternehmen: Strategien und Prozesse zur Steigerung der Wettbewerbsfähigkeit in der "Networked economy". Berlin: Springer.
- Haberfellner, R., Nagel, P. et al. (2002): Systems Engineering: Methodik und Praxis, 11. Aufl. (Daenzer, W., Huber, F.; Hrsg.). Zürich: Verlag Industrielle Organisation.
- Hafen, U., Künzler, C. et al. (1999): Erfolgreich restrukturieren in KMU: Werkzeuge und Beispiele für eine nachhaltige Veränderung. Zürich: vdf Hochschulverlag.
- Kaplan, R., Norton, D.P. (1997). Balanced Scorecard: Strategien erfolgreich umsetzen. Stuttgart: Schäffer-Poeschel.
- Österle, H. (1995): Business Engineering: Prozess- und Systementwicklung. Berlin: Springer.
- Strohm, O., Ulich, E. (Hrsg.) (1997): Unternehmen arbeitspsychologisch bewerten: ein Mehr-Ebenen-Ansatz unter besonderer Berücksichtigung von Mensch, Technik und Organisation (Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 10. Zürich: vdf Hochschulverlag.
- Trist, E.L., Bamforth, K.W. (1951): Some social and psychological consequences of the longwall method of coal getting. In: Human Relations 4; 3–38.
- Ulich, E. (1978): Über das Prinzip der differentiellen Arbeitsgestaltung. In: io Management Zeitschrift, 47/12. 566–568.
- Ulich, E. (2001): Arbeitspsychologie, 5. Aufl. Zürich/Stuttgart: vdf Hochschulverlag/Schäffer-Poeschel.
- Zuboff, S. (1988): In the Age of the Smart Machine: The Future of Work and Power. New York: Basic Books.



# 11

## Konzept

### Kapitelinhalt

- Analyse Prozesse, Funktionen, Objekte, Organisation & Technik
- Zielsetzungen und Neukonzeption der Geschäftsprozesse
- Spezifikation der erforderlichen Funktionalitäten
- Erarbeitung der Schnittstellenkonzepte

Mit der Wahl und der Dokumentation der optimalen Grobkonzeptvariante wurde der Lösungsrahmen des projektierten Informationssystems erst “skizzenhaft” festgelegt. In dem nun folgenden Konzept wird diese Lösungsskizze weiter konkretisiert und für alle Systemelemente (Geschäftsprozesse, Funktionen etc.) eine saubere Ausarbeitung vorgenommen.

*Kurortbeispiel: Im Beispiel des Transportsystems war nach dem Grobkonzept klar, dass eine Luftseilbahn gebaut würde. Im Konzept werden nun die Pläne und Layouts je für Parkplätze, Tal-, Mittel- und für die Bergstation erarbeitet. Die Projektierung muss selbstverständlich ein zusammenhängendes und funktionierendes Konzept ergeben.*

Die Konzeption und die Spezifikation der Systemanforderungen eines Informationssystems konkretisieren demgegenüber die einzelnen Geschäftsprozesse, die Objekte und die Module der Systemarchitektur. Bei einem Informationssystem ergänzt das Konzept schwerpunktmäßig den funktionalen Aspekt, insofern je Teilbereich alle funktionalen Anforderungen analysiert, konzipiert und aufgelistet werden. Dieses Vorgehen ist unabhängig davon, ob ein Wasserfall- oder Spiralmodell gewählt wird.

## 11.1 Inhalt und Vorgehen

### 11.1.1 Inhalt

Das Konzept kann zunächst als Konkretisierung und Ausarbeitung des Grobkonzeptes aufgefasst werden. Es wird vertieft geklärt, in welcher Weise die bestehenden Geschäftsprozesse auf Basis der technologischen Möglichkeiten anzupassen und zu optimieren wären. Jeder Geschäftsprozess wird sauber konzipiert. Außerdem sind auch die Zusammenhänge der verschiedenen Systemelemente darzulegen.

Eine Schwierigkeit der Konzeptions- und Spezifikationstätigkeit liegt häufig darin, den angemessenen Konkretisierungsgrad zu treffen. Einerseits sollte kein übermäßiger Aufwand betrieben werden, andererseits dürfen die Spezifikationen nicht zu oberflächlich abgefasst sein.

### 11.1.2 Vorgehen

Das Vorgehen der Konzeptionstätigkeit richtet sich nach dem Problemlösungszyklus aus. Dies bedeutet, dass auch im Konzept mit einer Situationsanalyse begonnen werden muss, allerdings auf einer tieferen Systemebene als im Grobkonzept. Die Zielsetzungen des Grobkonzeptes werden ebenfalls konkretisiert und beziehen auf die einzelnen Geschäftsprozesse und Teilgebiete. Die nachfolgende Konzeption, Bewertung und Dokumentation erfolgt gemäß dem Ablauf entsprechend Abb. 147.

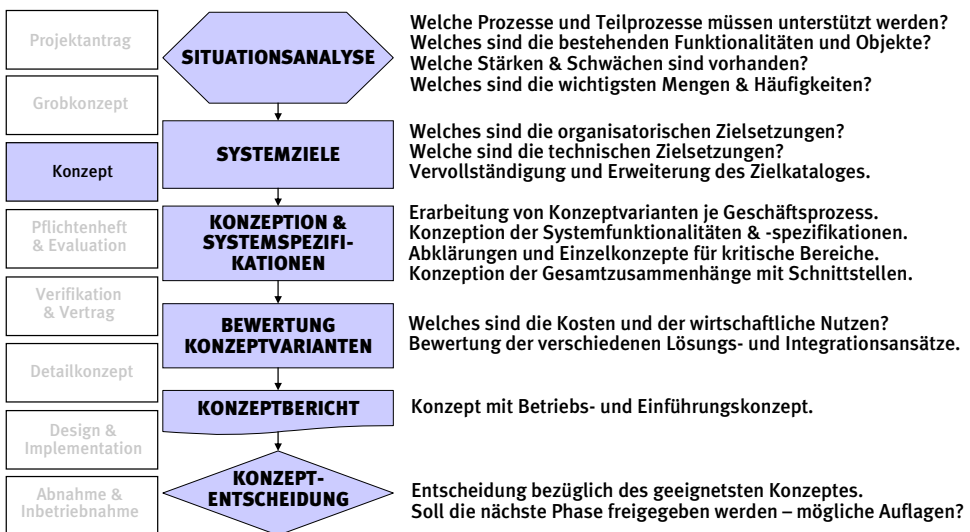


Abb. 147: Ergebnisübersicht der Konzepttätigkeit

### 11.1.3 Ebene der Systembetrachtung im Konzept

Die Konzept- & Spezifikationstätigkeit besteht wie erwähnt darin, das Grobkonzept zu verfeinern, auszubauen und die relevanten Prozesse, Funktionen, Objekte, Aufgaben und Techniksysteme näher zu spezifizieren. Daher kommt im Konzept die Betrachtungsebene für alle Systemsichten eine Stufe tiefer zu liegen als im Grobkonzept (s. Abb. 148). Als Ausgangspunkt dienen die Dokumente des Grobkonzeptes (s. Abb. 112).

Dies betrifft zunächst die Geschäftsprozesse, welche im Konzept auf ihre Teilprozesse hin analysiert und als Wertschöpfungsketten dargestellt werden. Auf Basis dieser Wertschöpfungsketten lassen sich die funktionalen Anforderungen konkretisieren und ebenfalls vertieft spezifizieren.

Auch die Geschäftsobjekte werden im Konzept vertieft betrachtet. Es wird in der Regel erst im Konzept ein strukturierter Objekt-/Datenkatalog erstellt, welcher mit den Mengen und Häufigkeiten jedes Objektes zu versehen ist. Hier ist zu beachten, dass die Objekte nicht zu detailliert betrachtet werden. Man sollte nicht Gefahr laufen, ein umfassendes Objektmodell erstellen zu wollen, denn dies wäre in diesem Stadium der Systemgestaltung eindeutig verfrüht. Weiter werden die Organisationsstrukturen vertieft analysiert und allenfalls die Struktur der Abteilungen betrachtet. Neben der Betrachtung und der Detaillierung der einzelnen Geschäftsprozesse und Funktionen müssen auch die Zusammenhänge zwischen den Geschäftsprozessen gestaltet werden.

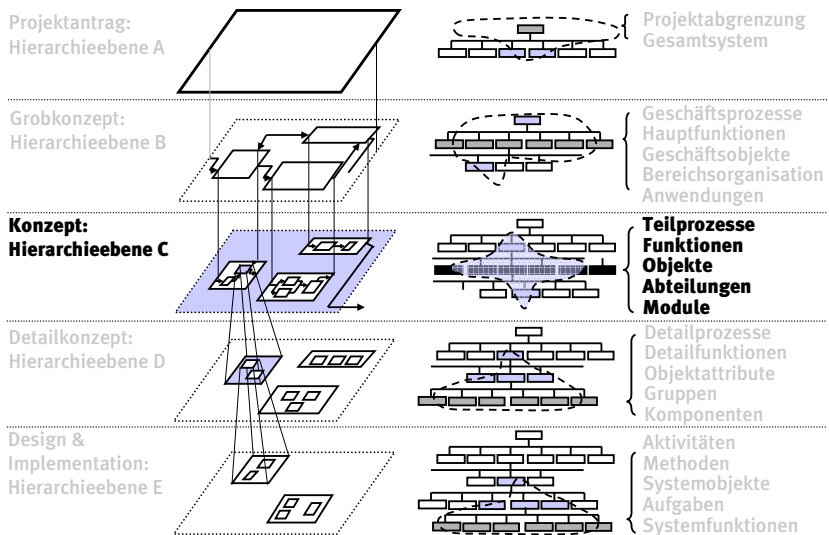


Abb. 148: Betrachtungstiefe des Systems im Konzept

### 11.1.4 Verwendete Modellierungsmethoden

Ausgehend von den im Grobkonzept erarbeiteten Modellen und Diagrammen kommen im Konzept gegenüber dem Grobkonzept neue und ergänzende Modellierungsmethoden hinzu. Die unten stehende Matrix wird mit ausgebaut und sukzessive komplettiert (s. Abb. 149).

Die bereits erarbeiteten Modelle dienen indessen als Grundlage zum weiteren Auf- und Ausbau des vorgesehenen Methodensets. Ein möglicher Vorgehenspfad zur Erarbeitung der Systemanforderungen kann der Abbildung entnommen werden. Wiederum darf diese Anleitung allerdings nicht als "best way," im Sinne eines starren Vorgehens, verstanden werden. Die unten stehende Matrix soll zunächst verdeutlichen, dass das im Grobkonzept erarbeitete Prozessmodell auch im Konzept zum weiteren Ausgangspunkt genommen werden kann.

Ein Schwerpunkt des Konzeptes wird wie erwähnt die vertiefte Analyse des Prozessmodells mit Hilfe von Wertschöpfungsketten darstellen. Der Fokus liegt also im Konzept nicht mehr auf den Gesamtprozessen, sondern auf der Identifikation der Teilprozesse.

Wesentliche Ergänzungen werden im Bereich der *Funktionssicht* vorgenommen. Die Funktionen lassen sich meist direkt aus einer Analyse der Teilprozesse ableiten. Ein weiterer Hinweis auf die Funktionen ergibt sich aus der Analyse der Stellenfunktionen. Ein wichtiges Resultat des Konzeptes stellt die Liste der gewünschten und abzudeckenden Funktionen dar.

<b>primär</b> <b>sekundär</b>	<b>Prozesssicht</b>	<b>Funktionssicht</b>	<b>Objektsicht</b>	<b>Aufgabensicht</b>	<b>Techniksicht</b>
<b>Prozesssicht</b>	<b>Prozessmodell</b> <b>Wertschöpfungsketten</b>			Stellenorientierter Informationsfluss	System-schnittstellendiagramm
<b>Funktionssicht</b>	Prozess-funktionen ↓	<b>Funktionsmodell / Funktionsliste</b>		Stellenfunktionen	
<b>Objektsicht</b>	Prozess-objekte →		<b>Liste der Objekte</b>		Datenobjekt-diagramm
<b>Aufgabensicht</b>				<b>Organisationsmodell</b>	
<b>Techniksicht</b>				Technikeinsatz-diagramm	<b>Systemarchitektur / Module</b>

Abb. 149: Methoden zur Modellierung des Konzeptes

## 11.2 Situationsanalyse

### 11.2.1 Analyse der Geschäftsprozesse



Wie bereits das Grobkonzept, so beginnt auch das Konzept mit einer sauberen Situationsanalyse. Es hat sich bewährt, ebenfalls mit der Betrachtung der Geschäftsprozesse zu beginnen und diese auf Stufe Gesamtprozess nochmals vertieft zu analysieren (s. Abb. 150). Hierfür wird die bestehende Geschäftsprozesslandkarte zur Ausgangsbasis genommen.

Die Konkretisierung auf Stufe Gesamtprozess bedeutet, dass z.B. je Geschäftsprozess eine exakte Definition erfolgt und Elemente wie die kritischen Erfolgsfaktoren, Segmentierungsmöglichkeiten etc. strukturiert erhoben werden. Der Vorteil der prozessorientierten Betrachtung liegt auch hier darin, dass man sich von den bestehenden organisatorischen Grenzen löst und die Leistungserbringung in den Mittelpunkt stellt. Für jeden Prozess sollten Überlegungen zu folgenden Punkten angestellt werden:

- Welches sind die Zielsetzungen des Prozesses?
- Systemabgrenzung des Prozesses (Inputs & Outputs) (Abschnitt 3.5.1)
- Messgrößen eines Prozesses? (Abschnitt 3.5.4)
- Segmentierungsgrößen? (Abschnitt 3.3.2)
- Bewertung der Flexibilität und der Qualität

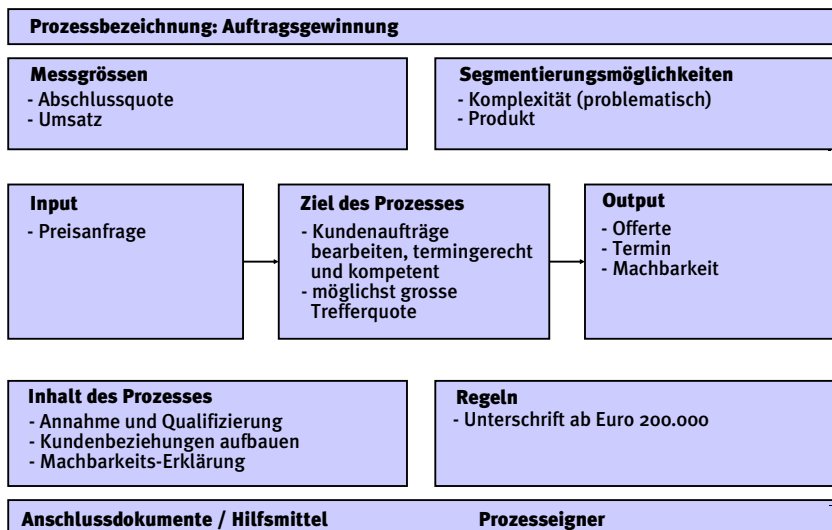


Abb. 150: Übersichtsmäßige Charakterisierung eines Geschäftsprozesses

### 11.2.2 Analyse der Teilprozesse je Geschäftsprozess

Nachdem die Konkretisierung der Geschäftsprozesslandkarte und die präzise Definition der Geschäftsprozesse vorgenommen wurde, wird in einem nächsten Schritt jeder betroffene Geschäftsprozess "eine Ebene" vertieft analysiert werden. Dies entspricht der unten dargestellten Detaillierung der einzelnen Geschäftsprozesse im Sinne von Wertschöpfungsketten bzw. der Analyse und Bestimmung der Teilprozesse (s. Abb. 151).

Ausgehend von der Analyse der Teilprozesse können diese auch einzeln näher charakterisiert und spezifiziert werden. So lassen sich auf Basis der erhobenen Teilprozesse die ihnen eigenen Funktionen sowie die wichtigsten Objekte je Teilprozesse ableiten. Die Darstellung der Einzelfunktionen und Objekte muss selbstverständlich nicht zwingendermassen gemäß der unten stehenden Darstellung erfolgen. Die Erfassung und Darstellung kann durchaus in textueller Form oder in Form einer Tabelle vorgenommen werden, welche entsprechend den Teilprozessen gegliedert ist.

Wird das Projekt gemäß dem Vorgehensmodell des Wasserfallmodells abgewickelt, so muss diese vertiefte Analyse der Teilprozesse sehr flächendeckend erfolgen. Dies kann einen nicht unerheblichen Aufwand bedeuten, da für jeden Geschäftsprozess im Rahmen von Workshops vertiefte Analysen durchzuführen sind. Im Spiralmodell erfolgt diese Analyse lediglich beschränkt auf die wichtigsten Hauptprozesse gemäß Auswahl und Prioritätensetzung.

Planung Kampagne	Definition Zielgruppe	Adressen generieren	Text, Druck & Mailing	Versand & Follow-up
Marketing	Marketing & Vertrieb	Marketing	Marketing	Vertrieb & Aussendienst
FUNKTIONEN	FUNKTIONEN	FUNKTIONEN	FUNKTIONEN	FUNKTIONEN
erstellen Marketingplan; planen Kampagne; erfassen Budget.	definieren Zielgruppe; selektieren Adressen; ...	generieren Adressen; exportieren Resultat.	erzeugen Briefe ...	...
OBJEKTE	OBJEKTE	OBJEKTE	OBJEKTE	OBJEKTE
Marketingplan; Kampagne; Budget.	Zielgruppe; Kunden; Inaktive Adressen Kundengruppen; Interessengruppe.	Adressen.	Textdokumente; E-Mail; Flyer.	...

Abb. 151: Wertschöpfungskette der Marketingkampagne



### 11.2.3 Analyse der Wertschöpfungskette

Die Technik der Wertschöpfungskette eignet sich – daher auch ihr Name – zur vertieften Analyse der Wertschöpfung je Teilprozess eines Geschäftsprozesses. Die Analyse der Wertschöpfungskette ergibt eine gute Übersicht der möglichen Ansatzpunkte zur Optimierung (s. Kapitel 3).

Vor einer Neugestaltung eines Prozesses können fallweise auch Analysen hinsichtlich der Bearbeitungs- und Durchlaufzeiten je Teilprozess vorgenommen werden. Die nachfolgende Grafik zeigt die Analyse dieses Verhältnisses von Durchlauf- und Bearbeitungszeit (s. Abb. 152). Das Verhältnis gibt an, wie sich die effektive Bearbeitungszeit in einem Teilprozess zur Durchlaufzeit verhält. Je größer dieses Verhältnis, desto länger bleibt das Bearbeitungsobjekt unbearbeitet liegen. Ob eine solche Liegezeit berechtigt ist oder nicht, können selbstverständlich erst weitergehende Abklärungen ergeben. Die Erstellung eines solchen Diagramms ist relativ aufwändig. Man wird daher eine solche Analyse primär dort vornehmen, wo tatsächlich eine zu optimierende Potenzial zu erwarten ist.

Durch diese Analyse entstehen erste Hinweise, wo und in welcher Form man die Prozesse beschleunigen könnte. Es wird auch bereits ein Gefühl dafür entstehen, wie groß die Reduktion der Durchlaufzeit sein könnte. In ähnlicher Weise sollten entsprechende Analysen und Benchmarks von Prozessen auch im überbetrieblichen Bereich der Supply Chains erhoben werden. Für weitere Angaben zu diesen Analysen sei auf die Literatur verwiesen (Sennheiser 2004).

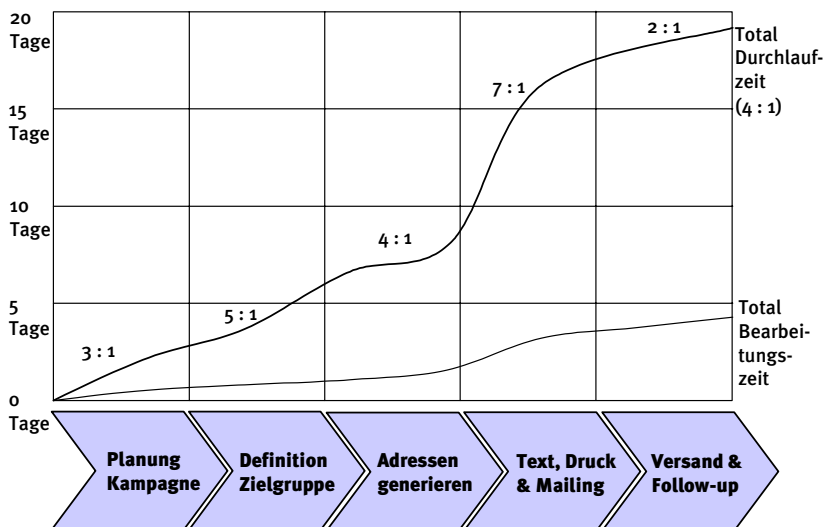


Abb. 152: Analyse der Durchlauf- und Bearbeitungszeiten

### 11.2.4 Analyse der Funktionen

Eine zentrale Erweiterung des Konzepts gegenüber dem Grobkonzept liegt sodann in der Analyse, Spezifikation und Priorisierung der durch das IT-System zu unterstützenden Funktionalitäten. Der funktionale Aspekt wird im Grobkonzept meist nur eingeschränkt und skizzenhaft beleuchtet. Hinweise auf die Funktionalität ergeben sich dort meist implizit aus der Definition der zu optimierenden Geschäftsprozesse.

Das Konzept hat eine umfassende Liste der Funktionen zu erarbeiten. Erste Hinweise auf die funktionalen Anforderungen haben sich aus der Analyse der Wertschöpfungskette ergeben. Weitere Hinweise auf die einzelnen Funktionen des betrachteten Systems lassen sich aus der Analyse der Stellenfunktionen ableiten (s. Abb. 153). Hierfür wurde das Stellenfunktionsendiagramm eingeführt. Unter Umständen wird man auch eine Kombination wählen, insofern die Funktionen je Organisationseinheit nach den Prozessen gegliedert werden.

Wie aus der unten stehenden Abbildung ersichtlich ist, können sich Redundanzen ergeben, falls verschiedene Stellen über exakt dieselben Funktionen verfügen. Diese Redundanzen verschwinden erst dann, wenn ein Funktionsmodell erstellt wird, welches die analogen Funktionen zusammenfasst und in einer Baumstruktur gruppiert. In der Praxis wird häufig die Form einer Auflistung der Einzelfunktionen gewählt, wobei das Gliederungskriterium der Prozess, die Stelle oder die Funktionsgruppe ist.

Marketing	Vertriebsleitung	Aussendienst	Innendienst
STELLENFUNKTIONEN	STELLENFUNKTIONEN	STELLENFUNKTIONEN	STELLENFUNKTIONEN
<b>Kampagnenprozess</b> erstellen Marketingp.; planen Kampagne; definieren Zielgruppe; erfassen Budget.	<b>Kampagnenprozess</b> planen Absatz; planen Vertrieb; durchführen Marketing.	<b>Kampagnenprozess</b> planen Absatz; planen Vertrieb; durchführen Marketing.	<b>Kampagnenprozess</b> planen Programm; planen Beschaffung.
<b>Eventprozess</b> planen Events; verwalten Anmeldung; reservieren Räume; terminieren Event; drucken Unterlagen.	<b>Aussendienst</b> planen Absatz; planen Vertrieb.	<b>Aussendienst</b> verwalten Kontakte; verwalten Offerten; administr. Aufträge; verwalten Produkte; fakturieren Kunden.	
	<b>Jahresplan, Forecast</b> planen Jahresumsatz; planen Prod.-umsatz.		
	<b>verwalten Adressen;</b> verwalten Kontakte; verwalten Offerten; administr. Aufträge; verwalten Produkte; fakturieren Kunden.		

Abb. 153: Stellenfunktionen

### 11.2.5 Analyse der Objekte

Im Konzept sollten nun auch Analysen die durch das System abzudeckenden Objekte betreffend vorgenommen werden. Ein erster Ansatz zur Erfassung der Objekte ergibt sich durch die nähere Betrachtung der Wertschöpfungskette. Ergänzend dazu kann eine Analyse aufgrund des Diagramms der Arbeitsobjekte je Organisationseinheit erfolgen (s. Abb. 154).

Stehen die wesentlichen Objekte aufgrund dieser Analyse fest, so können als nächstes die Mengen und Häufigkeiten festgehalten werden. Das wird eine Aussage über die notwendigen Systemressourcen zulassen. Auch wird bei der Ablösung bestehender Informationssysteme verlangt, dass die bestehenden Daten in das neue System übernommen werden können.

Es empfiehlt sich allenfalls, bereits im Grobkonzept mit einer Katalogisierung bestehender Datenbestände zu beginnen und auch die Qualität des Datenkatalogs im Konzept zu erweitern. Man wird also im Verlauf der Spezifikation der Systemanforderungen festhalten müssen, welche Datenbestände auf welche Subsysteme zu migrieren sind.

Eine weitere häufige Anforderung betrifft die Konsolidierung von Daten. So sind beispielsweise Adressdaten an verschiedenen Stellen historisch gewachsen, die man nun im Rahmen des Projektes zentral führen möchte. Solch gewachsene Strukturen komplizieren Projekte und bedürfen einer genauen Überprüfung.

Marketing	Vertrieb Standard	Aussendienst	Innendienst
ARBEITSOBJEKTE	ARBEITSOBJEKTE	ARBEITSOBJEKTE	ARBEITSOBJEKTE
Marketingplan	Absatzplan	...	Adressen
Kampagnen	Kundenaufträge	...	Debitoren
Events	Marketingplan	...	Kreditoren
	Adressen	...	Belege
	Offerten	...	Buchungen
	Fertigungsaufträge	...	Zahlungen
	Produkte	...	
	Preisliste	...	

Spezialräder
ARBEITSOBJEKTE
Absatzplan
Kundenaufträge
Marketingplan
Adressen
Offerten
Fertigungsaufträge
Fertigungsstückliste

Abb. 154: Erhebung der Arbeitsobjekte je Organisationseinheit

### 11.2.6 Analyse der Informationsflüsse

Nachdem in den vorangehenden Analysen der Fokus stets auf einzelne Geschäftsprozesse ausgerichtet war, gilt es nun, die Analysetätigkeit auf die Systemzusammenhänge und Beziehungen zu verlegen. Einen guten Überblick über die Zusammenhänge im Gesamtsystem ergibt sich aus der Darstellung des Stellenorientierten Informationsflusses (s. Abb. 155).

Im Gegensatz zum Grobkonzept werden die Informationsflüsse zwischen den Stellen/Organisationseinheiten vertieft analysiert, präzisiert und allenfalls einzeln im Diagramm beschriftet. Die explizite Darstellung der Prozesspfeile in den Organisationskästchen – wie in der unten stehenden Abbildung jeweils vorgenommen – ist nicht zwingend notwendig. Die Prozesse können auch den Beschriftungen entnommen werden. Es geht darum, die Beziehungen zwischen den einzelnen Organisationseinheiten zu analysieren und ein nochmals tieferes Verständnis für das Gesamtsystem zu entwickeln.

In gewissen Fällen wird diese Darstellung auch den Materialfluss beinhalten und explizit vom Informationsfluss unterscheiden. Dies ist primär in Projekten zur Optimierung der Supply Chain von Interesse.

Aus dem stellenorientierten Informationsfluss entsteht meist eine relativ umfangreiche Darstellung, welche detailliert die Beziehungen und Prozesse der verschiedenen Organisationseinheiten aufzeigt und dokumentiert. Selbstverständlich besteht die Möglichkeit, je Prozessgruppe ein separates Diagramm zu erstellen, um die Darstellung zu vereinfachen.

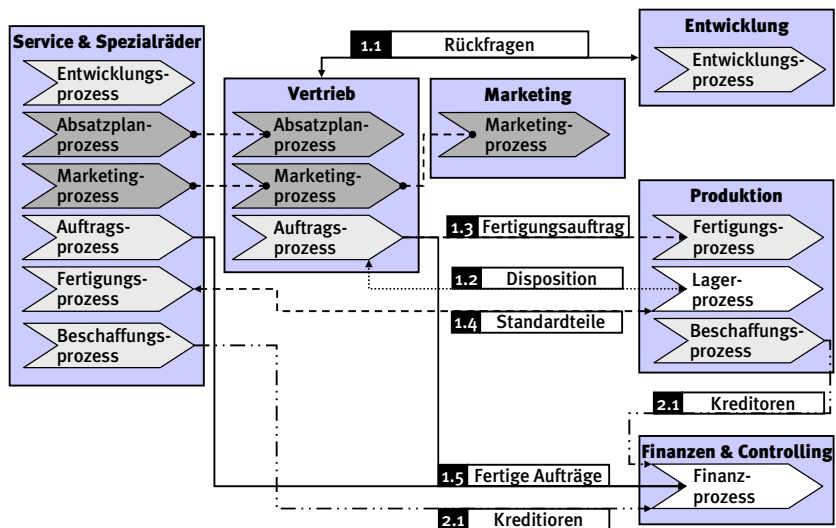
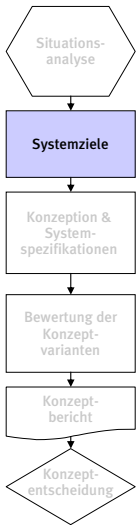


Abb. 155: Visualisierung der Gesamtsystemzusammenhänge

## 11.3

## Systemziele



Nach der Situationsanalyse und einer möglichen Stärken/Schwächenanalyse müssen auch im Konzept die Zielsetzungen festgelegt werden. Dieser Zielsetzungsprozess kann allerdings als Anpassung und Verfeinerung der Ziele des Grobkonzeptes verstanden werden. Die Gestalt der Zielformulierung bleibt dieselbe wie im Grobkonzept.

Der einfachste Ansatz zur Erweiterung besteht darin, jeweils für jeden Geschäftsprozess eine Reihe von Einzelzielen abzuleiten. Diese geschäftsprozessbezogenen Zielsetzungen werden anschließend zu einem umfassenden Zielkatalog zusammengefasst (s. Abb. 156).

Allenfalls ergeben sich auch gewisse gemeinsame und übergreifende Zielsetzungen, welche in jedem Geschäftsprozess wiederkehrend auftauchen. Es könnten die Zielsetzungen hinsichtlich der Ergonomie, der Technik oder der Organisation sein. Diese lassen sich selbstverständlich im Zielkatalog aggregieren und zusammenfassen.

In unserem CRM-Beispiel werden gemäß der unten stehenden Tabelle einzelne Zielsetzungen für den Geschäftsprozess "Kampagne" und Zielsetzungen für den Geschäftsprozess "Außendienst" abgeleitet. Allenfalls wird man sich separate Zielsetzungen für das Zusammenspiel der solcherart einzeln betrachteten Geschäftsprozesse festlegen. Die nachfolgende Konzeption wird sich, sofern sie sich auf mehrere Geschäftsprozesse bezieht, ebenfalls separat für jeden Geschäftsprozess erstellt werden.

Zielklassen	Ziele	Zieleigenschaften	Ausmass	Zielart/ Priorität	Zeitpunkt
Funktionsziele Kampagne	Effektive Kampagnen	Aufwand für Einzelkampagne	<= 0.5 Tage	Mussziel Prio. 1	nach Einführung
	Ungeplante Kampagnen	Zeitaufwand Identifikation	<= 5 Sek.	Mussziel Prio. 3	pro Jahr
	Erfolgskriterien festhalten	Attribute gegeben	...	...	...
	Marketingplan im System		...	...	...
	Freie Wahl der Medien				
Funktionsziele Außendienst	Bessere Abstimmung	Personal Betrieb	Stellenprozente	Prio. 5	pro Jahr
	...	...	...	...	...
Technische Ziele	Integration in ERP-Lösung	Schnittstelle zu XYZ	gegeben	Mussziel Prio. 4	nach Einführung
	Kalender	Schnittstelle		Prio. 6	...

Abb. 156: Ergänzter Zielkatalog

## 11.4 Konzeption und Systemspezifikationen

### 11.4.1 Konzept der Geschäftsprozesse



Gemäß dem Problemlösungsprozess des Systems Engineering erfolgt im Anschluss an die Festlegung der Ziele die eigentliche Konzeptionstätigkeit. Die Neukonzeption sollte mit einer Überprüfung der im Grobkonzept definierten Geschäftsprozesse auf hoher Ebene beginnen. Es ist die Frage nach den Möglichkeiten einer Neusegmentierung der Prozesse zu stellen, der Abgrenzung der Prozesse und Verantwortlichkeiten etc.

Anschließend wird eine Überprüfung und Neukonzeption der Geschäftsprozesse auf Stufe der Teilprozesse erfolgen müssen. Während im Grobkonzept der Fokus auf der Ebene der Geschäftsprozesse lag, wird man sich im Konzept eine Ebene tiefer mit den Teilprozessen auseinandersetzen. Die unten stehende Grafik stellt dar, welche Ansätze methodisch zu prüfen sind, um Teilprozesse bzw. Aktivitäten im Hinblick auf die genannten Punkte zu optimieren (s. Abb. 157).

Primär wird man bestrebt sein, durch das Informationssystem Verbesserungen hinsichtlich der Durchlaufzeiten, der Kosten- und der Qualitätsaspekte eines Geschäftsprozesses zu erreichen.

In Ergänzung zur Konzeption einzelner Geschäftsprozesse wird das Augenmerk auch wieder auf das Zusammenspiel und die Beziehungen zwischen den Geschäftsprozessen zu richten sein.

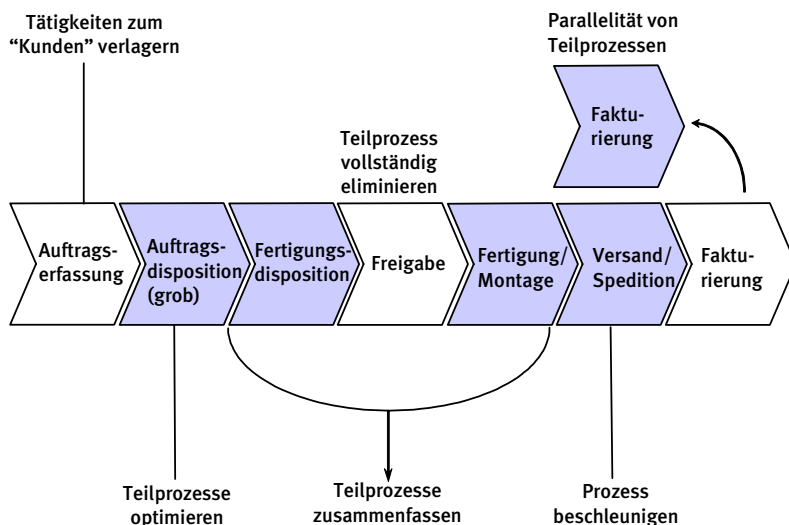


Abb. 157: Möglichkeiten zur Optimierung von Wertschöpfungsketten

### Business Process Reengineering

Die Neukonzeption von Geschäftsprozessen (engl.: “Business Process Reengineering”) sollte unter keinen Umständen dahingehend verstanden werden, dass Prozesse im Sinne eines Qualitätshandbuches lediglich streng formalisiert festzulegen wären und mit harter Hand durchzusetzen sind. Dieses weit verbreitet anzutreffende Missverständnis entspricht der Fehleinschätzung, dort “für Ordnung” sorgen zu müssen, wo eigentlich Flexibilität angesagt wäre. Prozessorientierung bedeutet das Gegenteil: Funktionen sollen organisatorisch zusammengefasst werden, damit Kundenaufträge in Teams effizienter bewältigt werden können.

Auch Hammer & Champy postulierten nichts anderes: “Das grundlegendste gemeinsame Merkmal aller radikal neugestalteten Unternehmensprozesse ist die Tatsache, dass es kein Fließband mehr gibt, d.h. viele ehemals voneinander getrennte Positionen und Aufgaben werden integriert und zusammengefasst. Wir beobachteten dies bei IBM Credit, wo eine einzige Person, der so genannte Deal Structurer, an die Stelle etlicher Spezialisten wie Bonitätsprüfer und Preisgestalter getreten ist” (Hammer & Champy 1993). Die Vorteile dieser Organisationsprinzipien zeigten bereits Studien aus den 60er-Jahren (Ulich 2001). Im Einzelnen bestehen folgende Möglichkeiten zur Prüfung:

- Optimierung von Teilprozessen und Aktivitäten (z.B. Einsatz eines Variantenkonfigurators),
- Parallele Durchführung von bisher sequenziellen Teilprozessen (z.B. können bisher nur einmal vorhandene Dokumente gescannt und mehreren Personen gleichzeitig elektronisch zur Verfügung gestellt werden),
- Beschleunigung des Ablaufs an den Randstellen der Teilprozesse (z.B. Kommunikationswege durch E-Mail beschleunigen),
- Elimination von Teilprozessen (z.B. Verzicht auf den Druck eines Manuals und Zurverfügungstellung per Internet),
- Integration eines gesonderten Teilprozesses in einen vor- oder nachgelagerten Teilprozess (auf diesen Punkt wird im nächsten Abschnitt der Aufgabengestaltung vertieft eingegangen),
- Überwälzung von Teilprozessen an die Kunden oder Lieferanten. Eine interessante Möglichkeit besteht heute darin, mit Hilfe der Informationstechnologie Sachbearbeitungstätigkeiten auszugliedern: Wenn ein Kunde eines Pizzageschäftes per Internet die Bestellung erfasst, dann übernimmt er “unwissentlich” die Bestellung, die Erstellung des Lieferscheines und die Rechnungsstellung. Gedankt wird ihm hierfür nicht – es wird den Kunden zurzeit als technologischer Fortschritt verkauft.

### 11.4.2 Konzept der Beziehungen zwischen den Geschäftsprozessen

Das oben stehende Reengineering hat Methoden zur Neugestaltung und Optimierung der einzelner Geschäftsprozesse und ihrer Teilprozesse aufgezeigt. Im Wechselspiel mit der Neugestaltung und Optimierung einzelner Geschäftsprozesse sind wiederum auch die Beziehungen zwischen den Geschäftsprozessen optimal auszulegen (s. Abb. 158). Diese Vorstellung basiert auf dem fundamentalen Grundverständnis von Systemen, bei dem diese einerseits über Einzelelemente verfügen und andererseits von den Beziehungen zwischen den Elementen leben. Es sollte bei der Konzeption auf ein zusammenhängendes Ganzes im Sinne eines Gesamtoptimums geachtet werden (Schnetzler 2003).

*Kurortanalogie: Im Konzept der Luftseilbahn müssen neben Plänen für die Systemelemente "Parkplätze" und "Bergstation" auch Betrachtungen des Zusammenspiels bedacht werden: Die Lage und das Konzept der Talstation steht in Verbindung mit der Anbindung an den öffentlichen Verkehr. Das Konzept der Bergstation hat unter Berücksichtigung der Kapazitäten der Tal- und Mittelstation sowie der Pisten zu erfolgen.*

Man wird sich also neben der Detaillierung der Prozesse ein Bild über das zukünftige Zusammenspiel der Subsysteme machen. In der Informatik werden allerdings nicht Personen, sondern Informationen befördert. Auch Informationen sollten sich in Analogie zur Bergbahn nicht stauen, Wartezeiten beinhalten, sondern zielgerichtet an den Bestimmungsort gelangen.

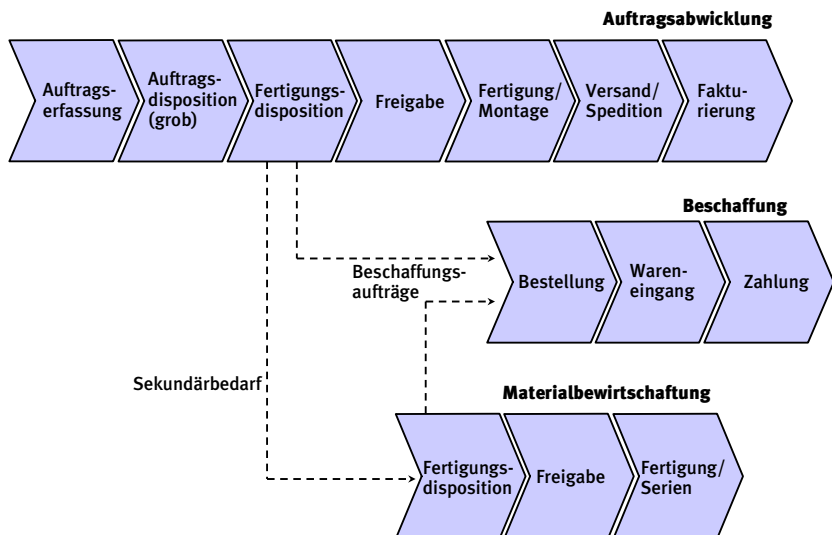


Abb. 158: Konzeption der Beziehungen von Geschäftsprozessen



### 11.4.3 Konzept und Spezifikation des Funktionsmodells

Ausgehend von den nunmehr erarbeiteten Modellen der Funktionssicht (Funktionen der Wertschöpfungsketten, Stellenfunktionen etc.) lässt sich in einem nächsten Schritt ein umfassendes Funktionsmodell erarbeiten (s. Abb. 159). Unter Umständen kann es an dieser Stelle auch hilfreich sein, hierfür auf bereits vorhandene Funktionskataloge zurückzugreifen, wie sie allenfalls bei anderen Unternehmen, Herstellern oder von Beratungshäusern bestehen.

Das Funktionsmodell stellt ein wirklich wichtiges Resultat des Konzepts dar. Funktionen werden meist noch relativ unabhängig von einer Systemarchitektur dargestellt, da die Softwarelösung ja in vielen Fällen noch nicht feststeht. Das Funktionsmodell müsste damit streng genommen auch keine Mensch-Maschine-Funktionsteilung aufzeigen (s. u.). Es kann grundsätzlich noch alle erforderlichen Funktionen des zukünftigen sozio-technischen Systems enthalten.

Infolge der regelmäßig hohen Anzahl einzelner Funktionen wird das Funktionsmodell meist eher textuell oder in Form einer Checkliste erstellt werden müssen. Eine grafische Darstellung wird, wie unten abgebildet, meist die Ausnahme bleiben müssen.

Die textuelle Form erlaubt weiter eine detailliertere Unterteilung der Funktionen sowie ergänzende Angaben, wie Prioritäten, Releaseplanung, etc.

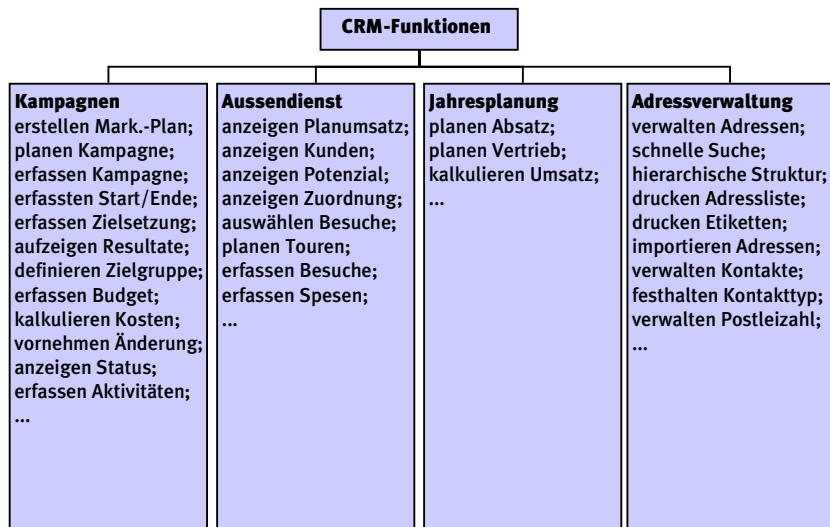


Abb. 159: Funktionsmodell

### **Konzept der Mensch-Maschine-Funktionsteilung**

Mit dem Einsatz von Informationssystemen stellt sich immer auch die Frage, welche Tätigkeiten automatisiert bzw. teilweise automatisiert werden sollen. Häufig wird mit dem Wort "Informatik" die Phantasie verbunden, in Zukunft müsse generell alles komplett "automatisch" ablaufen.

Diese Vorstellung kann zu Konzepten und Anforderungen führen, die sich gelegentlich in keiner Weise mehr begründen lassen – weder mit sachlichen noch mit wirtschaftlichen Argumenten. Als erfolgversprechendere Alternative hat sich in diesen Fällen erwiesen, Informationssysteme zunächst als Unterstützungssystem für qualifizierte Mitarbeiter zu verstehen.

Beispiel: Es könnte im Verlauf des Konzeptes der Wunsch aufkommen, die Spedition mitsamt der Tourenplanung der firmeneigenen Lastwagen dem Informationssystem zu übertragen – obwohl vielleicht nur ein einziger Lastwagen zu planen ist. Im Rahmen des Detailkonzeptes stellt es sich weiter heraus, dass dermaßen viele Restriktionen und Optimierungsmöglichkeiten bei der Tourenplanung vorhanden sind, dass die Programmierung des Systems nicht finanzierbar ist.

Der viel erfolgversprechendere Ansatz besteht daher darin, dem erfahrenen Spediteur "einfach" alle zur Tourenoptimierung relevanten Informationen zur Verfügung zu stellen. Durch einfache Operationen wird der Spediteur in die Lage versetzt, eine Lieferung rasch und einfach zusammenzustellen.

Ein solcher Ansatz stellt sich häufig als kostengünstiger und flexibler heraus als die komplexe Programmierung eines ausgefeilten Expertensystems. Insbesondere kleinere Betriebe werden mit einem solchen Vorgehen ihre Bedürfnisse abdecken müssen.

Zu einer genauen Analyse der tatsächlichen Vorteile ist an dieser Stelle zu raten. Eine weitere Möglichkeit würde sodann auch darin bestehen, dass die automatisch erzeugten Planungsoptimierungen nur im Sinne eines Vorschlages erstellt werden und durch den Anwender einfach und transparent angepasst werden kann. Ganz generell ist mit dem Aufkommen von so genannten Supply Chain Tools der Glaube an die Automatisierbarkeit von komplexen betrieblichen Entscheidungen mit Hilfe des Operations Research wieder neu aufgekeimt. Leider ist der Vorteil der Verfahren des Operations Research in gewissen Bereichen ohne die Einbeziehung entsprechender Experten nur sehr schwierig abschätzbar. Die Beurteilung des Einsatzes numerischer Verfahren zur Planung erfordert eine genaue Kenntnis, und einfache Rezepte, welche abschätzen lassen, wo ein Einsatz sinnvoll wäre, existieren in Folge der Diversifizität nicht.

#### 11.4.4 Konzept der Organisation

##### Konzept der funktionalen Integration von Organisationseinheiten

Die Gestaltung der Organisation betreffend gilt es im Konzept zu prüfen, inwiefern benachbarte Teilprozesse eines Geschäftsprozesses integrierbar sind. Dieses Vorgehen wird als “funktionale Integration” bezeichnet.

Die unten stehende Darstellung soll dies illustrieren (s. Abb. 160). Während die alte Organisation über vier separate organisatorische Stellen (A, B, C, D) zur Bearbeitung von vier Teilprozessen verfügt hat, werden nun diese vier Teilprozesse funktional in eine einzige Organisationseinheit integriert. Insbesondere ist hier jeweils zu prüfen, ob die Verlagerung von Teilprozessen von einer Abteilung in eine andere Sinn macht.

Damit gewinnt zumindest dieser Prozessabschnitt an Schub in Richtung Flexibilisierung. In der Praxis ist häufig der Ansatz prüfenswert, Planungs- und Dispositionsaufgaben in die produzierenden Organisationseinheiten zu integrieren. Dies ist eine Strategie, welche viele Unternehmen in den letzten Jahren verfolgt haben.

Die Produktionsplanung und -steuerung wurde in die Fertigung zurückverlagert – der Taylorismus wurde damit teilweise beseitigt. Funktionale Integration heißt immer auch ein Stück weit Qualifizierung und “Job Enlargement” der Mitarbeiter – leider vielleicht ein Grund für gewisse Unternehmen, aus Kostengründen nichts in diese Richtung zu unternehmen.

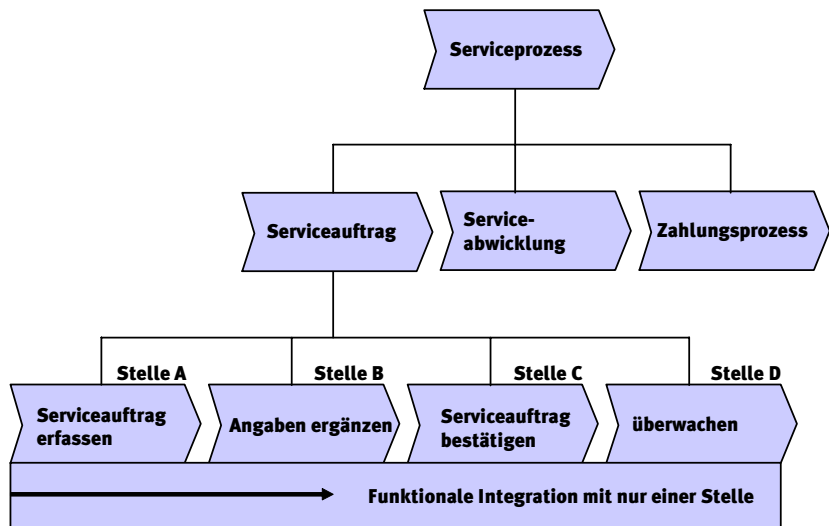


Abb. 160: Strategie der funktionalen Integration

### Konzept der Aufgabengestaltung

Im Grobkonzept wurde allenfalls die Schaffung eines neuen organisatorischen Bereichs im Sinne der Dezentralisierung angedacht. Im Konzept wird allenfalls eine Systemebene tiefer geprüft, ob neue Tätigkeiten geschaffen werden.

Ein zentraler Aspekt guter Aufgabengestaltung ist aus Sicht der Arbeitspsychologie das Prinzip der “Aufgabenorientierung”. Aufgabenorientierung bezeichnet einen Zustand des Interesses und des Engagements (Ulich 2001). Dieser Zustand kann durch verschiedene Merkmale der Aufgabengestaltung begünstigt werden, wie:

- Ganzheitlichkeit der einer Person übertragenen Aufgabe
- Anforderungsvielfalt der Aufgaben
- Möglichkeiten der sozialen Interaktion
- Autonomie
- Lern- und Entwicklungsmöglichkeiten
- Zeitelastizität und stressfreie Regulierbarkeit
- Sinnhaftigkeit der Aufgabe.

Unter den erwähnten Merkmalen kommt dabei der “Ganzheitlichkeit” eine hervorragende Rolle zu. Ganzheitlichkeit bedeutet, dass Aufgaben einen planenden, ausführenden und kontrollierenden Anteil beinhalten. Damit wird eine Rückmeldung innerhalb der Arbeitstätigkeit ermöglicht. Der Begriff der Ganzheitlichkeit wird oft auch mit dem Begriff der “vollständigen Aufgabe” oder “vollständigen Handlung” umschrieben. Merkmale der Ganzheitlichkeit oder vollständiger Aufgaben sind (Ulich 2001):

- Das selbständige Setzen von Zielen, die in übergeordnete Ziele eingebettet werden können;
- selbständige Handlungsvorbereitungen im Sinne der Wahrnehmung von Planungsfunktionen;
- Auswahl der Mittel einschließlich der erforderlichen Interaktionen zur adäquaten Zielerreichung;
- Ausführungsfunktionen mit Ablauffeedback zur notwendigen Handlungskorrektur;
- Kontrolle mit Resultatfeedback und der Möglichkeit, Ergebnisse der eigenen Handlungen auf Übereinstimmung mit den gesetzten Zielen zu überprüfen.

### 11.4.5 Konzept der Systemarchitektur

Mit den oben stehenden Prozess-, Funktions-, Objekt- und Organisationskonzepten wurde die Basis gelegt, um eine präzisierte Gestaltung der Systemarchitektur und der Ausprägung der einzelnen Systemmodule vorzunehmen.

An erster Stelle ist festzulegen, inwieweit das künftige Informationssystem über mehrere autonome Subsysteme desselben Modultyps verfügen müsste.

Man wird sich also beispielsweise für ein gemeinsames CRM-System für alle Bereiche entschieden haben, aber für die beiden autonomen Geschäftsbereiche je zwei autonome Planungssubsysteme, d.h. je eine eigenständige Konfiguration, aufsetzen. Damit hat jeder Bereich die Möglichkeit, Änderungen an den Modulen selbständig zu vollziehen (s. Abb. 161).

Die zweite Systemlösung, für unterschiedliche Geschäftsbereiche jeweils ein eigenes CRM-System vorzusehen, wurde im Grobkonzept bereits verworfen, da die Bedürfnisse einerseits doch recht ähnlich sind und andererseits bei der Implementation durchaus Synergien zum Zuge kommen können. Außerdem wäre die Koordination innerhalb des Unternehmens bei unterschiedlichen Systemen kaum mehr realisierbar gewesen und die Auskunftsbereitschaft gegenüber Kunden schwierig zu realisieren.

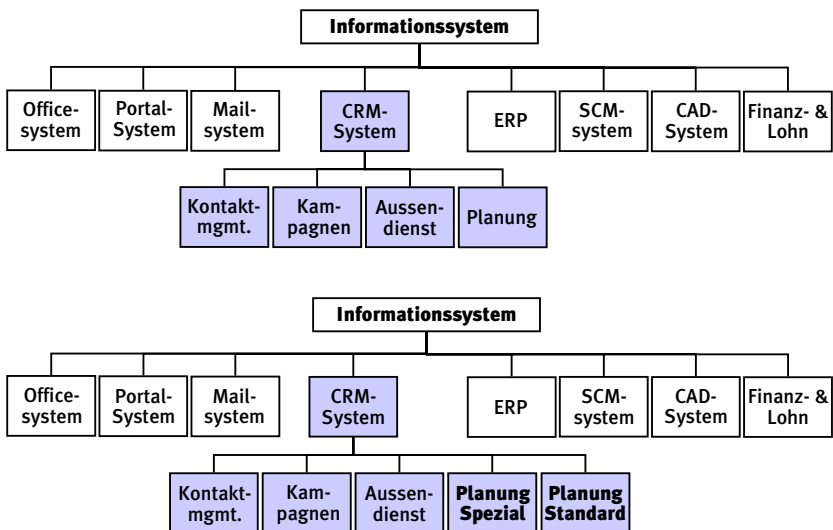


Abb. 161: Systemarchitekturvarianten für das CRM-System

#### 11.4.6 Konzept der Systembeziehungen und -schnittstellen

Das Wechselspiel zwischen einer Einzelelementebetrachtung und der Betrachtung des Gesamtsystems mit den Schnittstellen und Beziehungen sollte auch beim technischen System zur Anwendung gelangen.

Der Konzeption der Schnittstellen zwischen den Subsystemen kommt bei der Konzeption des technischen Systems eine erstrangige Bedeutung zu (s. Abb. 162). Jede Schnittstelle zwischen zwei Systemen bedeutet auch eine Schwachstelle.

Trotzdem gehört es zum Verständnis des Gesamtsystems, die Subsystem-Schnittstellen in Verbindung mit den relevanten Prozessen genau zu konzipieren.

Dabei ist zu beachten, dass im Grunde genommen nicht nur technische Schnittstellen betrachtet werden müssten, sondern auch Modulschnittstellen zwischen entsprechenden Subsystemen eines größeren modularen Systems.

In Ergänzung zu den Schnittstellen kann in diesem Systemschnittstellendiagramm auch angegeben werden, welche Systeme auf eine allenfalls zentrale Datenbank gemeinsam zugreifen.

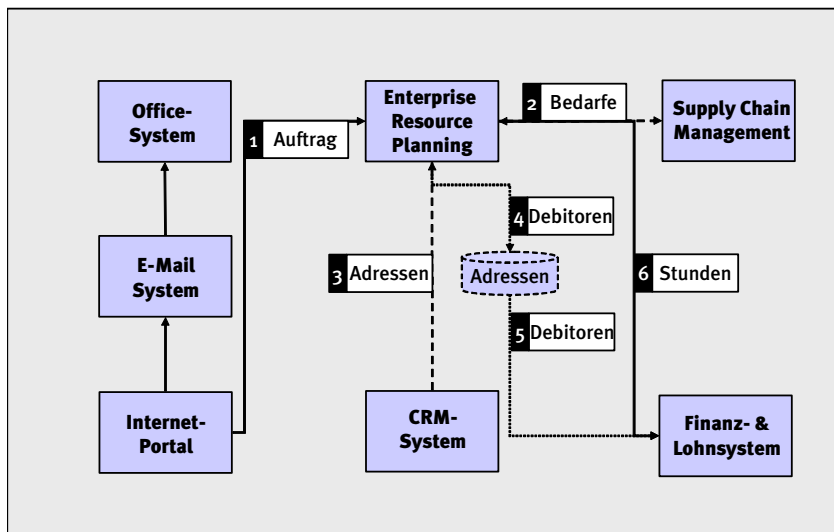


Abb. 162: Konzept der Systemschnittstellen

#### 11.4.7 Konzeption der Objekte und Objektzugriffe

Beziehungen zwischen den unterschiedlichen Anwendungssystemen entstehen häufig auf Basis von gemeinsam genutzten Datenobjekten dieser Systeme. Die Integration und der Zugriff auf mehrfach genutzte Objekte bzw. die Synchronisierung dieser Objekte ist in jedem Informatikprojekt eine wichtige Fragestellung (s. Abb. 163).

Dies soll aber nicht bedeuten, dass im Rahmen des Konzepts bereits Objektmodelle mit ihren Beziehungen erstellt werden sollten. Die Erstellung wirklich konsistenter Objektmodelle sollte in der Regel erst im Rahmen der Design- und Implementationstätigkeit erfolgen. Der frühe Fokus auf ein Objektmodell hat in der Vergangenheit zu hohem Projektaufwand geführt, und der Nutzen war meist von zweifelhaftem Wert.

Wenn man im Rahmen des Konzeptes also von der Konzeption der Objekte spricht, so sollte im Vordergrund stehen, die Systemintegration und Strukturen der wichtigsten Objekte zu überlegen und entsprechende Varianten zu erarbeiten. Im Zentrum der Überlegungen wird stehen, welche Daten im neuen Anwendungssystem isoliert bereitstehen und bei welchen Objekten allenfalls eine Synchronisation oder gar eine vollständige Integration in vorbestehende Datenbestände notwendig wäre.

Es ist weiter zu empfehlen, für die zentral wichtigsten Objekte (z.B. Auftrag, Kampagne im Beispiel) die zukünftige hierarchische Objektstruktur zu modellieren.

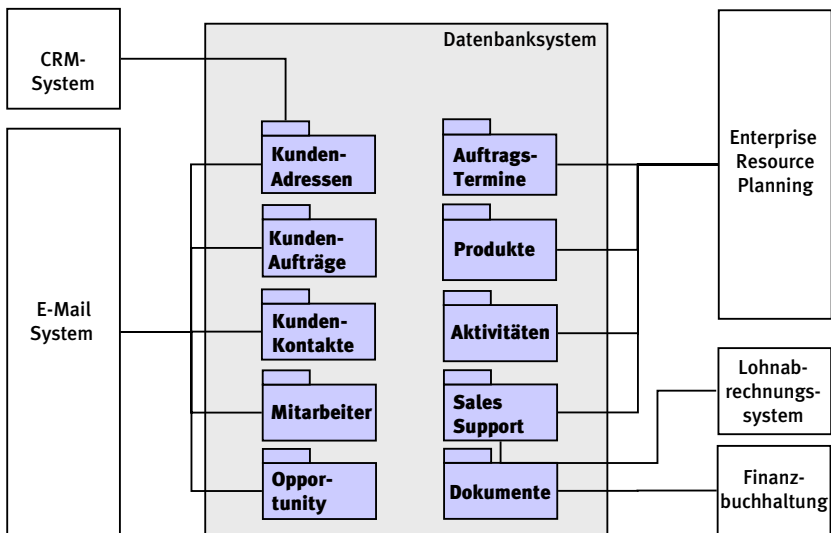


Abb. 163: Objektzugriffsdiagramm

### 11.4.8 Konzept der Hardware- und Kommunikationsarchitektur

Die technische Systemarchitektur sollte nicht in allen Einzelheiten fix festgelegt werden. In vielen Fällen empfiehlt es sich, die Konzeption der technischen Belange den potentiellen Anbietern zu überlassen und diesen keine unnötig einschränkenden Rahmenbedingungen vorzugeben.

Fragen der "Autonomie" betreffen allerdings das sozio-technische Konzept und dürfen dem Anbieter gerade nicht überlassen werden. Hier gilt es daher, den richtigen Grad zwischen zwingenden Anforderungen und Offenheit zu finden. In einem letzten Punkt der Spezifikationen sollte man sich daher gewisse Gedanken zur möglichen Hardware- und Kommunikationsarchitektur machen (s. Abb. 164). Es empfiehlt sich aber, wie bereits oben erwähnt, nicht allzu viel Zeit in diese Tätigkeit zu investieren.

Oftmals ist dies ja gerade die Aufgabe des Systemlieferanten, eine adäquate Hardwarearchitektur vorzuschlagen, welche auf die angebotene Softwarearchitektur abgestimmt ist. Die Dimensionierung der Hardwarekomponenten hängt jedenfalls stark vom Software- und Datenhaltungskonzept ab. Selbstverständlich müssen die grundlegenden Anforderungen bekannt gegeben werden, wie z.B.:

- Anzahl der bestehenden Arbeitsplatzrechner (PCs)
- Anzahl der Serversysteme
- Beziehungen zu den Informationssystemen

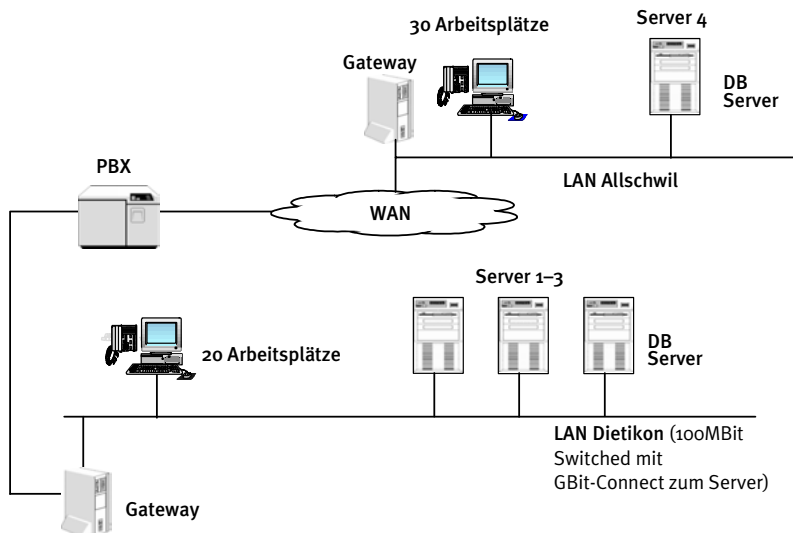


Abb. 164: Technisches Konzept im Sinne eines Vorschlags



#### 11.4.9 Einführungs- und Ausbildungskonzept

Auch im Konzeptionsschritt (um nicht von einer Phase zu sprechen) sollte ein bestimmter Anteil der Projektarbeit für das Einführungs- und Ausbildungskonzept reserviert werden (s. Abb. 165). Man wird sich also gegenüber dem Grobkonzept vertiefte Überlegungen machen müssen, in welcher Art und Weise die Einführung erfolgen sollte. Dies betrifft zunächst die Prioritäten der Einführung. Andererseits muss man sich zusätzliche Überlegungen machen, welche Benutzer wann und wie geschult werden müssten.

Eine besondere Beachtung verdient sodann die Konzeption einer etwaigen Datenübernahme (Datenmigration). Es sollte z.B. beantwortet werden, welche Datenobjekte zwingend übernommen werden sollten. In der Regel betrifft dies die so genannten Stammdaten (Adressen, Produkte, Stücklisten etc.). Die Bewegungsdaten (laufende Aufträge, Einkaufsaufträge etc.) werden in den allermeisten Fällen nicht übernommen werden müssen.

#### 11.4.10 Betriebs- und Unterhaltskonzept

In diesem Schritt spielen nun Sicherheits- und Datenschutzbetrachtungen eine erhebliche Rolle. Hohe Sicherheitsanforderungen – insbesondere was die Hochverfügbarkeit anbelangt – werden sich auf der Kostenseite erheblich auswirken. Da das Konzept die Grundlage für das Pflichtenheft bilden wird, müssen diese Aspekte bereits klar dargelegt werden.

Projekt: xy	Terminplanung																									
Tätigkeit / Phase																										
Pflichtenheft & Evaluation	■	■																								
Verifikation & Vertrag			■	■																						
Realisierungsvorbereitung				■	■																					
Detaillkonzept					■	■	■	■	■	■																
Design, Implementierung & Test											■	■	■	■	■	■										
Abnahme & Inbetriebnahme																								■		
Betrieb und Unterhalt																								■	■	■
■ Soll	Jan	Mrz	Mai	Jul	Sep	Nov	Jan	Mrz	Mai	Jul	Sep	Nov	Jan	Mrz	Mai	Jul	Sep	Nov	Jan	Mrz	Mai	Jul	Sep	Nov	Jan	Mrz
	05						06						06						06						06	

Abb. 165: Terminplan eines möglichen Einführungsplanes

#### 11.4.11 Morphologische Matrix für das Konzept

Entsprechend dem Grobkonzept kann auch im Rahmen des Konzeptes die Technik der Morphologie eingesetzt werden. Die nachfolgende Morphologie zeigt uns typische Parameter und Ausprägungen des Konzeptes, auf denen sich mögliche Konzeptvarianten erarbeiten lassen (s. Abb. 166).

In der Abbildung ist die favorisierte Lösungsvariante "Teilintegriertes CRM" eingezeichnet. Diese sieht eine bereichsspezifische Konfiguration der Module vor, welche die wichtigsten Daten mit den bereits vorhandenen IT-Systemen abgleicht. Es wäre auch denkbar, dass je Bereich ein so genannter Mandant installiert würde. Hierfür muss die Software "mandantenfähig" sein, was bedeutet, dass diese auf einer gemeinsamen Basisinstallation mehrere fast "unabhängige Firmen" betreiben kann.

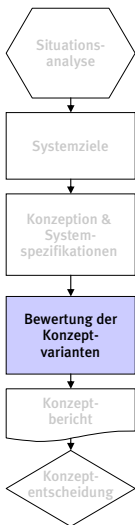
Die Gesamtheit der Einzelkonzepte ergeben zusammen eine oder mehrere Gesamtkonzeptvarianten. Es empfiehlt sich, auch in dieser Entwicklungsstufe nochmals verschiedene Gesamtkonzeptvarianten zu prüfen.

Im Vordergrund stehen oftmals die konzeptionellen Anforderungen an die Systemintegration und die Durchgängigkeit der erhobenen Geschäftsprozesse. Hier können meist unterschiedliche "Effizienzstufen" angestrebt werden. Das Grobkonzept fokussierte sich auf den angestrebten Lösungsumfang und eine grobe Integration. Das Konzept legt die Schnittstellen bereits ausgereifter dar.

<b>Ausprägung</b> <b>Parameter</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>
<b>System-architektur</b>	<b>Einsatz einer neuen Software</b>	Module von bestehendem Softwarepaket	Nutzung bestehender Module	Wird aufgrund Evaluation entschieden
<b>Konfigurations-autonomie</b>	Einheitliche Module für alle Bereiche	Einheitliche Konfiguration	<b>Autonome Konfiguration je Bereich</b>	Eigener Mandant/eigenes System je Bereich
<b>Softwaretyp</b>	<b>Standardsoftware</b>	Standardsoftware mit moderaten Anpassungen	Standardsoftware mit individuellen Erweiterungen	Individualsoftware
<b>Daten-integration</b>	System verfügt über isolierte Daten	<b>Daten werden mit Drittlösungen abgeglichen</b>	Daten werden synchronisiert und abgeglichen	Datenbankzugriff auf gemeinsame Datenbank
<b>System-integration</b>	<b>Keine funktionalen Schnittstellen</b>			....
...	...			....

Abb. 166: Beispielhafte Morphologie auf Ebene der Konzeption

## 11.5 Bewertung der Konzeptvarianten



Ganz entsprechend dem Problemlösungszyklus erfolgt im nächsten Schritt eine notwendige Bewertung von Lösungsvarianten. Im Konzept werden einerseits diverse Variantenentscheidungen hinsichtlich der Optimierung einzelner Geschäftsprozesse gefällt werden müssen. Es könnte also eine isolierte Bewertung für die unterschiedlichen Optimierungsvarianten des Kampagnenprozesses und eine Bewertung der Varianten des Aussen-dienstprozesses erfolgen.

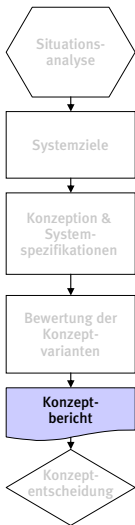
Zusätzlich ergeben sich im Konzept verfeinerte Gesamtkonzeptvarianten. Diese können z.B. den Integrationsgrad in die Systemlandschaft betreffen (s. Abb. 167). Außerdem könnten sich im Konzept auch neue Gesamtvarianten hinsichtlich der funktionalen Abdeckung bzw. des Projektumfangs ergeben. Die Grobkonzeption sollte sich zwar nicht grundsätzlich ändern, es sollte aber nicht verboten sein, auf Entscheidungen zurückzukommen.

In unserem Beispiel stehen drei Konzeptvarianten zur Wahl, welche einen unterschiedlichen Integrationsgrad in die bestehende Systemlandschaft aufweisen. Eine erste Variante betrifft die Teilintegration mit einem Abgleich der wichtigen Datenobjekte. Die zweite Variante sieht die vollständige Integration in das bestehende ERP-System vor. Die dritte Variante ist ein losgelöstes IT-System. In unserer Bewertung schneidet die Teilintegration am besten ab. Unter Umständen wird man sich aber erst im Rahmen der Evaluation definitiv für oder gegen eine dieser Varianten entscheiden.

<b>Varianten</b> <b>Merkmale</b>	<b>Teilintegriertes CRM</b>	<b>Vollintegriertes CRM</b>	<b>Nichtintegriertes CRM</b>
Beschreibung der Lösung	Das CRM-System weist eine Integration der wichtigsten Daten mit dem ERP-System auf (Adressen, Umsätze, etc.).	Das CRM-System ist voll in das bestehende ERP-System integriert. Es baut auf derselben Datenbank auf und muss vom selben Hersteller stammen.	Diese Lösung sieht ein komplett losgelöstes System vor. Es pflegt eigene Adressen und erhält keine Angaben aus dem ERP (z.B. Umsatz).
Vorteile	Relativ einfache Implementation; diverse Anbieter.	Optimale Integration.	Einfache Implementation.
Nachteile	Keinen Zugriff auf die Aufträge. Schnittstelle muss erstellt werden.	Kompromisse bei der Lösung. Hohe Kosten & Komplexität.	Viele Schnittstellen und hohes Implementationsrisiko.
Investitionssumme	350	400	250
Betriebskosten p.J.	150	200	100
Kapitalwert (NPV)	140	100	60
Strategischer Wert	62	65	55
Risikozahl	22	35	18

Abb. 167: Bewertung unterschiedlicher Konzeptvarianten zur Integration

## 11.6

**Konzeptbericht**

Im Gegensatz zum Grobkonzept steht im Konzeptbericht nicht die Variantenwahl im Vordergrund der Dokumentation. Im Zentrum steht hier die Beschreibung der Prozesse, der Funktionalitäten und der Integration.

**I ZUSAMMENFASSUNG****II SITUATIONSANALYSE**

- Systemelemente & Systemabgrenzung
- Organisationsmodell
- Stellenorientierter Informationsfluss, Materialfluss
- Ist-Geschäftsprozesse  
Wertschöpfungsketten mit Teilprozessen, Funktionen, Objekten.
- Ist-Systemarchitektur  
Technikeinsatz, Systemschnittstellen, HW-, Netzwerkarchitektur.

- Mengen & Häufigkeiten
- Stärken & Schwächen, Chancen & Gefahren
- Zukunftsanalyse

**III ZIELFORMULIERUNG**

- Strategische Ziele, Systemziele, Vorgehensziele

**IV KONZEPT**

- Beschreibung der Konzeptvarianten
- Stellenorientierter Informationsfluss  
Änderungen und Ergänzungen gegenüber Grobkonzept.
- Konzept Geschäftsprozess 1  
Gestaltung der Wertschöpfungskette, Spezifikation der Funktionen, Organisationsänderungen, Objektstrukturen, ...
- Konzept Geschäftsprozess 2 bis n
- Systemarchitektur  
Technikeinsatz, Systemschnittstellen, Datenobjektdiagramm.

- Betriebs- und Unterhaltskonzept

**V WIRTSCHAFTLICHKEITSVORSCHAU**

- Kosten, Nutzen, Risiken, Wirtschaftlichkeit

**VI WEITERES VORGEHEN**

- Sofortmassnahmen
- Terminierung und Etappierung des Projektes
- Einführungs-, Migrations- und Ausbildungskonzept
- Projektmanagement und Projektorganisation

## 11.7 Konzeptentscheidung



Das ursprüngliche Grobkonzept hat ergeben, dass die “CRM-Light”-Variante die optimale Lösung darstellt. Im Rahmen des Konzeptes wurden die gewählten Geschäftsprozesse (Kampagnenprozess, Außendienstprozess, etc.) näher analysiert und die notwendigen Funktionalitäten spezifiziert. Das Gesamtkonzept wurde hinsichtlich der Integration verfeinert.

Zusätzlich wurde konzipiert, dass zumindest eine Teilintegration mit dem bestehenden Systemumfeld angestrebt werden muss. Dies wurde im Grobkonzept noch nicht klar spezifiziert bzw. weitestgehend noch offen gelassen. Außerdem wurde nun klar, dass den spezifischen Gegebenheiten des Unternehmensbereiches für Spezialfahräder dahingehend Rechnung getragen werden kann, indem für diesen Bereich eine eigenständige Systemkonfiguration vorgenommen wird.

Es bestehen auch bereits klarere Vorstellungen zur geplanten Systemeinführung. Zusätzlich wurde ein Betriebs- und Unterhaltskonzept erarbeitet, welches die Verantwortlichkeiten für den Systemunterhalt festlegt (s. Kapitel 17).

Bereich Schritt	Sozio-technisches System	Einführung & Ausbildung	Betrieb & Unterhalt	Projektmanagement & Controlling
<b>Systemanalyse</b>	Analyse der Informationsflüsse; Analyse Wertschöpfungsketten; Analyse Prozessbeziehungen; Funktionen Wertschöpfungskette; Stellenfunktionen; Objektlisten; Organisationsmodelle; Analyse Systemarchitektur; Analyse Systemschnittstellen; Mengen & Häufigkeiten; Prioritätensetzung.	Analyse des Ausbildungsbedarfes.	Analyse von Betrieb & Unterhalt.	Projektneuplanung; Projektmanagement, Projektcontrolling.
<b>Systemziele</b>	Verfeinerung der Hauptziele.	Einführungs- & Ausbildungsziele.	Betriebs- & Unterhaltsziele.	
<b>Konzeption &amp; System-spezifikationen</b>	Konzept Wertschöpfungsketten; Konzept Prozessbeziehungen; Organisatorisches Konzept; Funktionsmodell / -katalog; Konzeption der Datenzugriffe; Konzeption des Technikeinsatzes; Systemschnittstellenkonzept; Gesamtsystemarchitektur; Wirtschaftlichkeit & Risikoliste.	Einführungs- & Ausbildungskonzept; Prioritäten der Einführung; Ausbildungskonzept; Datenübernahme-konzept.	Betriebs- & Unterhaltskonzept; Datenschutz- & Sicherheit; Betriebskosten.	
<b>Bewertung der Konzeptvarianten</b>	Bewertung der Konzeptvarianten; Bewertung Wirtschaftlichkeit.			
<b>Konzeptbericht</b>	Dokumentation Konzept & Systemspezifikationen.			
<b>Konzept-entscheidung</b>				Entscheidung Konzept.

Abb. 168: Ergebnisübersicht der Konzepttätigkeit

## 11.8 Literatur zum Konzept

- Alard, R. (2002): Internetbasiertes Beschaffungsmanagement direkter Güter: Konzept zur Gestaltung der Beschaffung durch Nutzung internetbasierter Technologien; Dissertation ETH Zürich.
- DeMarco, T. (1978): Structured Analysis and System Specification. Englewood Cliffs, NJ: Prentice Hall.
- Gause, D., Weinberg, G. (1993): Software Requirements: Anforderungen erkennen, verstehen und erfüllen. München: Carl Hanser.
- Macaulay, L. (1988): Requirements Engineering. Berlin: Springer.
- Martin, C.F. (1988): User-Centered Requirements Analysis. Englewood Cliffs, NJ: Prentice Hall.
- Pohl, K. (1996): Process-Centered Requirements Engineering. New York: John Wiley & Sons.
- Rosson, M.B., Maass, S., Kellogg, W.A. (1988): The Designer as User: Building Requirements for Design Tools from Design Practice. In: Communications of the ACM, 31(11). 1288–1298.
- Schnetzler M., Sennheiser, A. (2003): Identification of Improvement Strategies in Production Networks. In: Building the Knowledge Economy – Issues, Applications, Case Studies (Cunningham, P. et. al.; ed.). Amsterdam: IOS Press. 306–312.
- Sennheiser, A. (2004): Determinant based selection of Benchmarking Partners and Logistics Performance Indicators; Dissertation ETH Zürich.
- Yourdon, E. (1989): Modern Structured Analysis. Englewood Cliffs, NJ: Prentice Hall.







# 12

## **Pflichtenheft und Evaluation**

### **Kapitelinhalt**

- Inhalte und Ausprägungen von Pflichtenheften
- Definition des Kriterienrasters mit Gewichtung
- Identifikation der Anbieter und Verfahren zur Ausschreibung
- Nutzwertanalyse und Evaluationsverfahren zur Bewertung

Softwareprojekte werden häufig unter Einbeziehung von externen Lieferanten durchgeführt. Diese zeichnen in diesen Fällen verantwortlich für die Lieferung der Software, der notwendigen Hardware und sie erbringen diverse Dienstleistungen in Zusammenhang mit der Einführung.

Falls ein externer Lieferant einbezogen wird, muss idealerweise sehr klar festgelegt werden, welche Leistungen zu welchen Kosten erbracht werden sollten. Das Auswahlverfahren stellt einen mehrstufigen Prozess dar, welcher die infrage kommenden Anbieter sukzessive einengt, bis man zur bestgeeigneten Lösung gelangt.

Als Basis für das Auswahlverfahren dient ein Pflichtenheft (Lastenheft, engl.: “Request for Proposal – RFP”), welches die Anforderungen in Form von Spezifikationen nochmals sauber aufbereitet und in eine knize Form bringt. Das Pflichtenheft wird gegenüber dem Konzept um zusätzliche Punkte ergänzt, welche das Evaluationsverfahren erleichtern. Dies betrifft primär Fragestellungen im Zusammenhang mit dem Anbieter und betreffend Referenzinstallation, Kosten etc.

## 12.1 Inhalt und Vorgehen

### 12.1.1 Inhalt

Im Zentrum steht die mehrstufige und methodische Auswahl eines Anbieters mit Hilfe einer formellen Bewertung des Kosten-/Nutzenverhältnisses. Diese Auswahl kann nur auf Basis gut strukturierter Angebote erfolgen. Nach der vorangehenden Konzeption und Spezifikation des Systems sind alle (funktionalen und technischen) Anforderungen an das System in einer genügenden Tiefe bekannt, um diese in diesem Schritt in einem Pflichtenheft (synonym: Lastenheft; engl. "Request for Proposal – RFP") zusammenzufassen und an Anbieter zur Angebotsabgabe abgegeben.

### 12.1.2 Vorgehen

Auch diese Phase (oder Tätigkeitsschritt) wird idealerweise entsprechend dem bekannten Problemlösungsprozess abgewickelt. Zunächst werden die im Rahmen des Konzeptes erarbeiteten Anforderungen analysiert und in einem Pflichtenheft zusammengefasst. Danach werden die Zielsetzungen und Kriterien zur Auswahl festlegt (s. Abb. 169).

Im Rahmen der nachfolgenden Ausschreibung wird der Markt der möglichen Lösungsanbieter analysiert und die geeigneten Anbieter werden angeschrieben. Die strukturierte Bewertung mit Hilfe einer Nutzwertanalyse schließt diesen Schritt ab.

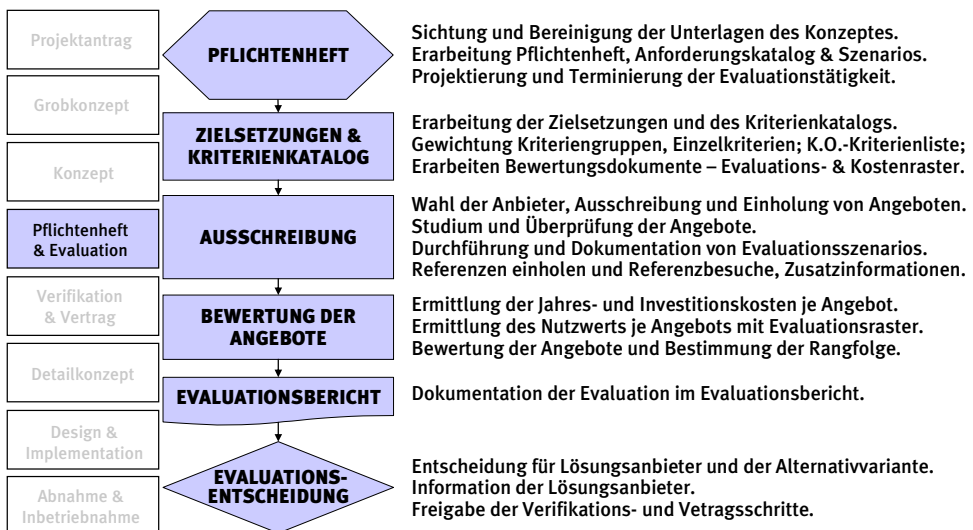


Abb. 169: Vorgehensschritte der Pflichtenheft- und Evaluationstätigkeit

### 12.1.3 Mehrstufiges Auswahlverfahren

Die Bewertung von Angeboten und die Auswahl des Lieferanten erfolgt mit Hilfe eines mehrstufigen Auswahlverfahren (s. Abb. 170). Mit Hilfe eines "Vorfilters" werden die auf dem Markt verfügbaren und potentiellen Lösungen einer ersten Sichtung unterzogen. Die allenfalls sehr große Zahl der in Betracht kommenden Anbieter muss aufgrund einiger weniger Filterkriterien (z.B. Sprache, Modulangebot, ...) auf eine begrenzte Anzahl reduziert werden (Longlist). Bei dieser Wahl sollte auch darauf geachtet werden, dass auch innovative Lösungen mitberücksichtigt werden.

Die Anbieter auf der Longlist nehmen an der Evaluation teil und werden als nächstes mit einem Grobpflichtenheft bedient. Je nach Anzahl der selektierten Anbieter könnte auch bereits das vollständige Pflichtenheft an die Anbieter abgegeben werden. Dies erhöht allerdings für die Anbieter den Aufwand zur Angebotserstellung. Auch die Bewertung der Angebote im Rahmen des Grobfilters sollte noch mit einem begrenzten Aufwand erfolgen und auf Basis von wenigen Musskriterien (Referenzen etc.) erfolgen.

Im Anschluss an diese Grobselektion erfolgt im Rahmen eines Feinfilters die eigentliche Evaluation der nunmehr verbliebenen Lösungsanbieter. Sie erfüllen alle die Musskriterien, und die anschließende Auswahl erfolgt auf Basis eines Pflichtenheft und sehr detaillierter Angebote. Als Instrument dient die Nutzwertanalyse mit den definierten Kriterien.

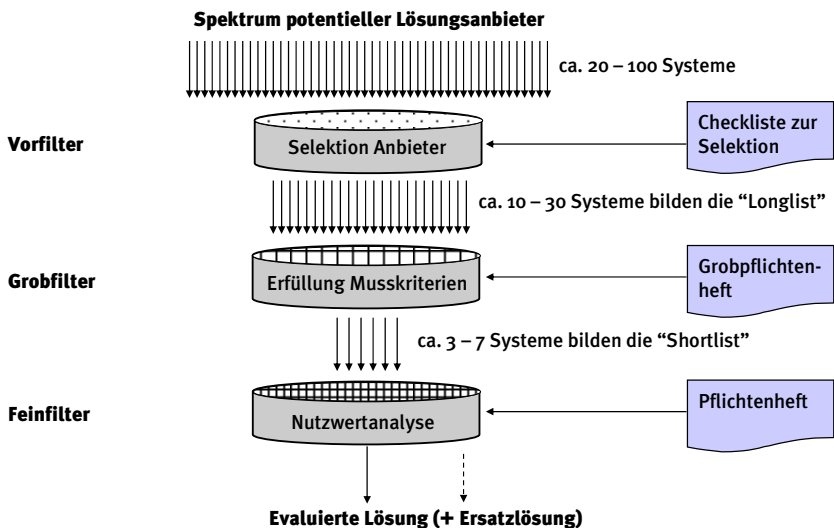
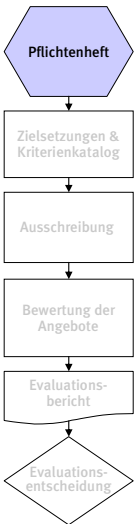


Abb. 170: Ablauf einer Evaluation

## 12.2 Pflichtenheft



Die dem Pflichtenheft zugrunde liegenden Systemmodelle entsprechen, wie bereits oben erwähnt, exakt den in der Konzept- und Spezifikationstätigkeit entwickelten Diagrammen und Beschreibungen. Die Grundlagen bestehen also und man nimmt allenfalls nur gewisse Anpassungen vor. So ist man unter Umständen nicht gewillt, vertrauliche Details des Unternehmens zu veröffentlichen. Auch können gewisse Analysen (z.B. Stärken- bzw. Schwächenanalysen) einer erhöhten Vertraulichkeit unterliegen.

Andererseits müssen wiederum andere Punkte speziell für das Pflichtenheft ergänzt werden. So sollte jedes Pflichtenheft eine konzise und illustrative Übersichtsdarstellung der Firma enthalten. Diese wurde unter Umständen noch nicht in geeigneter Weise erstellt und muss nun ergänzt werden.

Der Detaillierungsgrad von Pflichtenheften kann recht stark variieren. Noch vor wenigen Jahren war es üblich, dass Unternehmen – ganz entsprechend der Philosophie des Wasserfallmodells – äußerst umfangreiche Spezifikationen und Pflichtenhefte erstellt haben. Die Anforderungen an das neue System wurden also zumeist recht detailliert erhoben und auch sehr umfassend dokumentiert. Allerdings erfolgte diese Modellierung in den überwiegenden Fällen aufgrund einer überwiegend funktionalen Vorstellung, d.h. mit entsprechend funktional aufgebauten Checklisten. Die Analyse und Dokumentation der Prozesssicht und weiterer Sichten kam indessen meistens zu kurz.

Grundsätzlich muss den Anbietern ein Verständnis für die Geschäftsprozesse mit den Teilprozessen verschafft werden, andererseits müssen auch die funktionalen Anforderungen klar dargelegt werden. Drittens muss auch klar hervorgehen, welche Struktur und Komplexität die Geschäftsobjekte haben (einstufig, mehrstufig, variantenreich, etc.)

Die Ausführungen in den vorangegangenen Kapiteln haben gezeigt, dass IT-Lösungen in Zukunft teilweise neuen Anforderungen zu genügen haben. Ansonsten laufen die Unternehmen stark Gefahr, dass ihr Informationssystem in einem dynamischen Umfeld zum Hemmschuh bei der Unterstützung der wesentlichen Erfolgsfaktoren wird.

So wurde in Abschnitt 9.3.1 ausgeführt, dass bei der Konzeption von Informationssystemen neben der reinen Abdeckung der betriebspezifischen Spezifikationen zunehmend die Frage der Flexibilität, Anpassungsfähigkeit, Eignung zur Teamarbeit und der Möglichkeiten zur flexiblen Kommunikation geprüft und die Zielsetzungen des Unternehmens berücksichtigt werden. Die beste Modellierung und Einführung eines Systems ist nutzlos, wenn diese Faktoren nicht erfüllt sind.

### Funktionskatalog im Pflichtenheft

Wie Eingangs des Kapitels erwähnt, basieren die im Pflichtenheft enthaltenen Systemanforderungen auf den Resultaten des Konzepts. In vielen Fällen werden die funktionalen Anforderungen zur Grundlage genommen und in Form eines Funktionskatalogs konkretisiert (s. Abb. 171).

Dieses Beispiel zeigt eine pragmatische Vorgehensweise, indem für jeden Teilprozess eines Geschäftsprozesses die entsprechenden Funktionalitäten bzw. funktionalen Anforderungen in einer tabellarischen Hierarchie aufgelistet sind.

Es kann weiter angegeben werden, mit welcher Priorität die entsprechende Funktion abzudecken ist. Damit wird auch gleich geklärt, welche Funktionen im Rahmen der Evaluation als Muss-Kriterien gehandhabt werden und die Nicht-Erfüllung mit hoher Wahrscheinlichkeit zum Ausschluss der Lösung aus dem Evaluationsprozess führt.

Idealerweise wird eine solche Tabelle den Anbietern elektronisch abgegeben, damit diese ihre Antworten direkt eintragen können. Hierfür müssten dann weitere Kolonnen angeführt werden, welche zur Beantwortung der Funktionalität dienen.

Das unten stehende Beispiel stellt nur einen kleinen Ausschnitt dar. Es kommen weitere Funktionalitäten und zusätzliche Fragen zum Anbieter hinzu, wie dies im Raster weiter unten angegeben wird.

Erforderliche CRM Funktionen	Priorität der Funktion				
	Priorität 1 (Muss)				
	Priorität 2				
	Priorität 3				
	Zukunft				
	1	2	3	4	Bemerkungen
<b>Einzelfunktionen</b>					
<b>Kundenprofil</b>					
• Freie Definition von Kundenprofilen	x				
• Kundenprofil am Bildschirm und als Ausdruck verfügbar		x			
• Umsatzanzeige nach Produktgruppen	x				
• Wichtigste Aufträge je Kunde			x		
• Wichtigste Angebote je Kunde	x				
• Sales Pipeline	x				
• Sicht auf die Produkte wechseln	x				
<b>Unterlagen</b>					
• Zeige alle Unterlagen, welche der Kunde bereits erhalten hat	x				
• Marketing Enzyklopädie (alle Unterlagen des Marketings)	x				
<b>Kundenbesuch</b>					
• On- und offline-Zugriff auf die Daten des Kunden	x				

Abb. 171: Funktionale Beschreibung im Pflichtenheft

### 12.2.1 Kritik am Pflichtenheft

Infolge des zunehmenden Zeitdruckes im Geschäftsleben ("Time-to-market") wird von Systemanbietern immer häufiger verlangt, Angebote zu erstellen, ohne dass die Anforderungen an die Lösung vorgängig klar spezifiziert wurden. Damit sei erwähnt, dass viele Unternehmen nicht mehr gewillt sind, umfangreiche Pflichtenhefte zu erstellen. Betrachten wir uns die gängigsten Kritikpunkte. Es wird unter anderem argumentiert:

- Pflichtenhefte sind zu aufwändig und zu teuer bzw. die Zeit reicht zur Erstellung eines Pflichtenheftes nicht mehr aus;
- Man wählt besser einen bekannten Anbieter, der sich in ähnlichen Projekten bereits bewährt hat;
- Es gibt genügend schlüsselfertige Lösungen.

Eine Vorgehensweise ohne Pflichtenheft birgt aber nicht nur für die Softwareanbieter hohe Risiken in sich. Auch die Gefahr eines Projektmiss Erfolgs erhöht sich drastisch. Betrachten wir daher nachfolgend die oben genannten Kritikpunkte und fragen uns, wie darauf zu antworten ist.

#### **Gescheiterte Projekte auch mit bekannten Standardlösungen**

Die in Kapitel 8 angeführten Beispiele von gescheiterten Projekten haben aufgezeigt, dass Informatikprojekte mit hohen Risiken behaftet sind. Ohne ein klares Pflichtenheft hat man nun aber im Fall auftretender Probleme keine klare Grundlage zur Beurteilung anstehender Meinungsdivergenzen. Es stellt sich daher die Frage, ob mit dem Verzicht auf ein Pflichtenheft nicht am falschen Ort gespart wird. Die Kosten eines Pflichtenheftes betragen im Vergleich zu den Gesamtkosten nämlich nur rund 10 Prozent und sind damit vergleichsweise recht gering.

Es gibt sodann auch genügend Beispiele von bekannten Softwareanbietern mit schlüsselfertigen Lösungen, die zeigen, dass die Wahl eines etablierten Anbieters alleine noch keine Garantie für ein erfolgreiches Projekt darstellt. Ein diesbezügliches Beispiel betrifft den Marktdominator im Bereich ERP-Systeme. Diese Firma wurde in den USA von einem seiner Kunden auf rund 500 Mio. Dollar verklagt. Der Kunde behauptete gemäß entsprechenden Zeitungsberichten, dass das Unternehmen in Konkurs geraten sei, weil das System nie richtig hätte eingeführt werden können.

Auf wessen Seite die Schuld in diesem Fall lag, ist kaum zu beantworten. Sicher muss die Projektleitung sowohl auf Kunden- wie auf Lieferantenseite einen maßgebenden Anteil an einem derartigen Misserfolg getragen haben. Es ist aber zumindest ein Indiz dafür, dass auch mit etablierten und erfolgreichen Anbietern Projekte scheitern können.

### 12.2.2 Pflichtenheft im Wasserfall- und Spiralmodell

Wie wirkt sich die Wahl des zugrunde gelegten Vorgehensmodells auf die Form des Pflichtenheftes aus? Im Wasserfallmodell wird jede Projektstätigkeit (z.B. Spezifikation, Pflichtenhefterstellung) nur einmal vorgenommen, dafür um so “vollständiger” (s. Kapitel 8). Demzufolge werden beim Wasserfallmodell sehr umfassende Pflichtenhefte erstellt. Beim Spiralmodell beschränkt sich das Pflichtenheft zunächst auf die Kernanforderungen und wird sodann gewissermaßen zyklisch iterativ erweitert.

Ein relativ vollständiges Pflichtenheft gemäß Wasserfallmodell wird immer dann unvermeidbar sein, wenn die Anforderungen an das zu beschaffende Informationssystem eine umfassende Funktionalität abdecken sollen (z.B. Gesamtablösung). Es gibt viele Unternehmen, die in einem einzigen Schritt diverse Anwendungen gemeinsam ablösen möchten, z.B. das Vertriebs-, Produktions-, Finanz- und das Personalsystem. Bei derart umfassenden Projekten muss im Rahmen der Evaluation geklärt werden, ob die Anforderungen durch den Systemlieferanten in vollem Umfang abgedeckt werden können. Dies erfordert zwangsläufig ein relativ umfangreiches Pflichtenheft, wenn man nicht unnötige Risiken eingehen möchte. So sollten insbesondere die Gesamtkosten für das Projekt geklärt sein.

Die zyklische Vorgehensweise gemäß dem Spiralmodell bietet sich andererseits dann an, wenn die Systemlieferanten klar feststehen und sich das Projekt dafür eignet, stufenweise in Betrieb zu gehen. Man wird aber in den meisten Fällen eine Kombination zwischen dem Wasserfall- und dem Spiralmodell anwenden – und dies auch bei integrierter Standardsoftware. Es wird daher an dieser Stelle nochmals darauf hingewiesen, dass sich das Spiralmodell und das Wasserfallmodell keinesfalls ausschließen, sondern vielmehr ergänzen. In den meisten Projekten wird man eine Mischform der beiden Vorgehensmodelle anwenden müssen.

Denn auch im oben erwähnten Fall einer Gesamtablösung ist es möglich, zunächst nur die wesentlichsten, geschäftskritischen Anforderungen umfassend und detailliert zu spezifizieren. Jene Bereiche, die nicht gleichermaßen geschäftskritisch sind (z.B. Einkauf, Personal), werden nur summarisch spezifiziert.

Bei den relevanten Anforderungen geht man demnach nach dem Wasserfallmodell vor, bei den sekundären Anforderungen nach dem Spiralmodell. Eine zeitgemäße Evaluation wird sich daher vermehrt auf die kritischen Hauptprozesse rund um die Auftragsabwicklung fokussieren. Im Rahmen von Szenarien kann zudem überprüft werden, inwiefern sich unterschiedliche Handlungsabläufe im genannten System vollziehen lassen und dadurch das Kriterium Flexibilität erfüllen.

### 12.2.3 Unterschiede im Pflichtenheft für Individual- und Standardsoftware

In welcher Weise unterscheidet sich die Form eines Pflichtenheftes für Standardsoftware von derjenigen einer Individualsoftware? Die Erstellung von Pflichtenheften zur Einführung von Standardsoftware hat zunächst den großen Vorteil, dass häufig auf bereits vorhandene Vorlagen bzw. auf vorgefertigte Checklisten zurückgegriffen werden kann.

Dies ist übrigens mit ein Grund, weshalb die oben erwähnte Feststellung, es gäbe schon genügend schlüsselfertige Lösungen, die das Pflichtenheft erübrigen, nicht wirklich überzeugt. Denn gerade bei schlüsselfertigen Lösungen bietet sich explizit die Möglichkeit, auf bestehende Funktionslisten als Vorlage zurückzugreifen und einfach weiter darauf aufzubauen. Mit diesem Vorgehen erhält man einerseits ein verbindliches Angebot für einen klar definierten Funktionsumfang, andererseits hält sich der Aufwand relativ gesehen in Grenzen.

Die Vorteile von Standardsoftwareprojekten sind:

- “man kann schon etwas sehen”,
- es gibt bestehende Funktionskataloge,
- es gibt bereits Referenzinstallationen,
- man kann sich auf die Unterschiede verschiedener Lösungen konzentrieren und die Zahl der Anbieter dadurch reduzieren.

Bei Individualsoftware betritt man indessen häufig applikatorisches Neuland. Dies erschwert das Vorgehen, da weniger gut abschätzbar ist, welches die wirklich kritischen Punkte sein dürften. Auf der anderen Seite gibt es auch einige Vorteile zu verzeichnen:

- die Lösungen sind meist weniger umfassend und man konzentriert sich natürlicherweise auf die Kernanforderungen,
- die Ziele werden zwingenderweise klar formuliert.

Von der Form des Pflichtenheftes her gibt es aber keine generellen Unterschiede. Im Grunde genommen wird in beiden Fällen auf die beschriebenen Modelle zurückgegriffen.



#### 12.2.4 Pflichtenheft als konzeptioneller “Vorvertrag”

Im Pflichtenheft müssen die spezifizierten Prozesse, Funktionen, Objekte und Aufgaben nicht nur nach Außen an die Anbieter kommuniziert werden, sondern sollten auch gegenüber den internen Stellen als eine verbindliche Richtschnur der Anforderungen genommen werden können. Selbstverständlich stellt auch das Konzept bereits eine Grundlage dar.

Damit stellt das Pflichtenheft sowohl einen “Vorvertrag” nach Innen als auch allenfalls einen “Vertragsbestandteil” nach Außen dar (Becker 2000) (s. Abb. 172). In der Regel wird allerdings erst das Angebot zum expliziten Vertragsbestandteil, da häufig nicht alle Pflichtenheftelemente vom Anbieter vollständig erfüllt werden können.

Das Pflichtenheft hat aber auch eine firmeninterne Funktion. Die diversen Anwendergruppen wollen nämlich Klarheit darüber haben, welchen Nutzen sie aus einem Informatikprojekt ziehen können. Denn in der Regel erfordert ein solches Projekt hohe Investitionen, die eine breite Unterstützung in der Geschäftsleitung genießen müssen. Die Verantwortlichen aus dem Vertrieb, der Produktion, etc. wollen also in der Regel ganz klare Informationen, was der entsprechende Verantwortungsbereich von einem neuen Informationssystem erwarten darf. Ein Projektleiter bzw. eine Projektleiterin hat demnach auch die Aufgabe, diesen Anspruchsgruppen klar aufzuzeigen, wo die Verbesserungspotenziale des neuen Systems liegen.

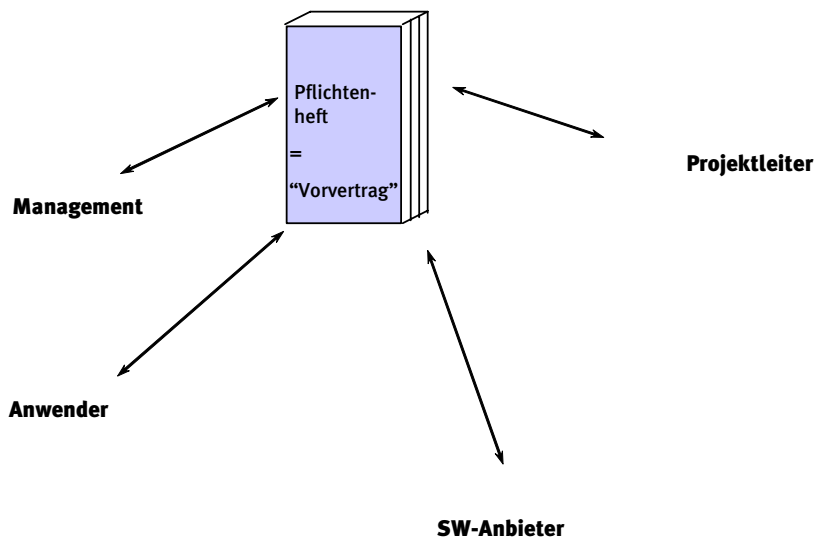
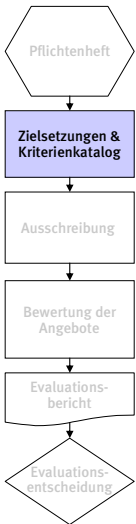


Abb. 172: Pflichtenheft als “Vorvertrag”

## 12.3 Zielsetzungen und Kriterienkatalog

### 12.3.1 Zielsetzungen



Das Grundproblem bei gescheiterten Projekten dürfte sehr häufig gerade darin liegen, dass sich die Unternehmen trotz vorhandenen Pflichtenhefts nie wirklich bewusst waren, was an ihrem Unternehmenssystem tatsächlich besonders ist bzw. welches für das Unternehmen die wirklich kritischen Anforderungen sind. Es gelingt in diesen Fällen daher auch nicht, die Spezialitäten und Besonderheiten gegenüber dem Lieferanten zu kommunizieren und im Rahmen der Evaluation ein besonderes Augenmerk zu richten. Es ist nämlich zu erwarten, dass gerade die Besonderheiten durch eine allenfalls vorhandene Standardlösung noch nicht abgedeckt sind. Sonst wäre es per Definition keine Besonderheit.

Hierfür gibt es gleich mehrere Gründe. Häufig besteht in kleineren und mittleren Unternehmen nicht das organisatorische Fachwissen, um solcherart komplexe Informatikprojekte durchzuführen. In der Regel ist es in kleineren Unternehmen nicht möglich, eine qualifizierte Person mit dem notwendigen Fachwissen für ein solches Projekt freizustellen. Es mangelt oft auch an der notwendigen Distanz, um selbst zu beurteilen:

- welche Schwächen die Geschäftsprozesse aufweisen,
- was bei einer neuen Lösung die wichtigsten Ziele wären.

Es erfordert nämlich eine umfangreiche Kompetenz hinsichtlich der am Markt verfügbaren Informationstechnologien, um zu beurteilen, was mit deren Hilfe tatsächlich gelöst werden sollte.

Der Auslöser für Informatikprojekte besteht gelegentlich einzig darin, bestehende, vielleicht veraltete, Informatiklösungen zu ersetzen. Gerade dies greift aber unter Umständen zu kurz. Pflichtenhefte werden also den an sie gestellten Anspruch nicht erfüllen, wenn nicht auch die vorangehenden Projektschritte sauber wahrgenommen werden. Insbesondere muss die Phase des "Projektantrags" die wesentlichen Ziele definieren. Das anschließende sozio-technische Konzept hat diese weiter zu festigen.

Spätestens bei der Erstellung eines Pflichtenheftes muss sich ein Unternehmen Klarheit verschaffen, welches die Zielsetzungen sein sollten, z.B.:

- welches die wichtigsten Prozesse sind,
- welches die organisatorischen und technischen Schwächen sind,
- ob diese Schwächen mit Informatik gelöst werden können,
- welches die resultierenden Hauptziele des Projekts sind.

### 12.3.2 Kriterienkatalog

Der Kriterienkatalog bildet im Rahmen einer Evaluation die Grundlage für transparente Auswahl im Sinne einer Nutzwertanalyse. Er hält fest, welche Elemente der Angebote bewertet werden sollen. Selbstverständlich sind nicht alle Elemente einer Lösung gleich bedeutend. Daher werden die einzelnen Kriteriengruppen ebenfalls unterschiedlich gewichtet (s. Abb. 173).

Wurde im Vorfeld bereits eine saubere Zielhierarchie erstellt, dann wird es nun auch leicht fallen, diese in einen Kriterienkatalog einzubringen. Die nachfolgend angeführten Kriteriengruppen sind für eine Evaluation also meistens von Bedeutung. Diese Kriteriengruppen stellen nur mögliche Vorschläge dar. In jedem Projekt wird selbstverständlich eine diesbezügliche Struktur erarbeitet werden müssen.

#### Funktionale Abdeckung

Das vielleicht bedeutendste Kriterium bei der Wahl einer Lösung besteht darin, dass die Lösung eine adäquate Funktionalität aufweist. Was nützt eine noch so gute Technik, wenn die Funktionalität nicht gegeben ist?

Gemäß unserem Verständnis bedeutet dies, dass die relevanten Prozesse, Funktionen, Objekte und Aufgaben durch das System in optimaler Weise abgebildet werden können. Diese funktionalen Kriterien werden meist in einem Kriterienblock zusammengefasst, welcher auch regelmäßig das höchste Gewicht erhält. Die Summe aller Gewichte ergibt 100 Prozent.



Abb. 173: Beispielhafter Kriterienkatalog

### **Flexibilität und Konfiguration**

Ein zweiter Kriterienblock sollte die Möglichkeiten Flexibilität und Konfiguration abdecken. Je besser und einfacher ein Informationssystem an neue Situationen angepasst werden kann, desto besser erscheint seine Akzeptanz im Unternehmen und in der Zukunft garantiert. Das System darf also nicht nur für den gegebenen Zeitpunkt ausgerichtet sein, denn bereits nach Ende des Projektes sind neue und geänderte Anforderungen entstanden. Dieser Punkt beinhaltet daher die Beurteilung der Möglichkeiten, neue Reports, neue Felder und Geschäftsprozesse rasch und einfach zu implementieren.

Aber auch die angemessene Basiskonfiguration bildet ein wichtiges Kriterium in vielen Unternehmen. Allenfalls sollen ja – wie in unserem CRM-Beispiel angedacht – relativ autonome Geschäftsbereiche zukünftig über eine autonome Systemkonfiguration verfügen können. Idealerweise sind Systeme modular aufgebaut.

### **Technische Konzeption**

In einem weiteren Kriterienblock können sodann alle zusätzlichen und verbleibenden technischen Belange zusammengefasst werden. Dies betrifft zunächst die Unterstützung der geforderten Betriebssysteme und Hardwareplattformen, eine nötige Beurteilung der gewählten Programmiersprache des Anbieters, die Abdeckung der Anforderungen an die Systemsicherheit wie Passwortschutz, Möglichkeiten der Zugriffsbeschränkung auf Module, Objekte und Einzelfelder sowie die Rechtevergabe im Sinne des Lesens und Schreibens von Datenfeldern.

Das Softwareprodukt soll ergonomisch sein, eine gute Performance aufweisen und eine optimale Benutzungsoberfläche haben: Dazu gehören einfache Erlernbarkeit für Anfänger und einfache Bedienbarkeit für geübte Benutzer.

### **Einführung und Vorgehen**

Ein eigenständiger Kriterienblock sollte zu den Zielsetzungen der Einführung angelegt werden. Es betrifft dies zunächst die Beurteilung der allgemeinen Einführungsmethodik, aber auch die Abstimmung des Einführungsvorgehens mit den Anforderungen des Unternehmens. Entspricht beispielsweise der Vorschlag zur Erarbeitung des Detailkonzeptes den gewünschten Anforderungen?

### **Anbieter und Vertrag**

Der Anbieter spielt eine nicht zu unterschätzende Rolle bei der Wahl eines Systems. Es betrifft dies primär die Kompetenz und Erfahrung im genannten Gebiet. Der Anbieter sollte aber auch eine Kontinuität und eine optimale Wartung bzw. Weiterpflege des Systems garantieren können.

## 12.4 Ausschreibung

### 12.4.1 Selektion der Anbieter durch Vorfilter



Der Markt der Anbieter von Softwarelösungen ist in aller Regel sehr unübersichtlich. Oftmals handelt es sich um recht junge und zersplitterte Märkte. Allein im deutschsprachigen Raum bieten beispielsweise im Marktsegment der ERP-Systeme mehrere hundert Firmen Softwarelösungen an. Zum einen gibt es branchenbezogene Programme, andere Lösungen bieten sich branchenübergreifend an.

In anderen Anwendungsgebieten (z.B. Versicherungssoftware, Bankenlösungen) ist das Angebot an Standardlösungen indessen wesentlich geringer. Der Grund hierfür mag darin liegen, dass es sich diese Firmen noch bis vor kurzem leisten konnten, selber zu entwickeln.

Diese Zeit ist beispielsweise in der Maschinenindustrie schon länger zu Ende. Im Dienstleistungsbereich zeichnet sich eine ähnliche Bewegung ab, so bei Banken und Versicherungen.

#### Lieferantengruppierungen

Im Bereich der Systemanbieter lassen sich etwa folgende Gruppen bilden:

- Hardwarehersteller (heute häufig als Generalunternehmer tätig (z.B. IBM, HP, Siemens))
- Softwarehäuser (z.B. SAP, Oracle, Trilab)
- Systemhäuser (z.B. CAP Gemini, Andersen, CSC Ploenzke)
- Fachhändler (Business-Partner der Hardwarehersteller wie IBM Partner)
- Softwareentwicklungshäuser
- Outsourcing-Partner (z.B. EDS)

Unter Umständen macht es Sinn, bei der Auswahl potentieller Lieferanten auf Anbieter ganz unterschiedlicher Art zuzugehen und Angebote einzuholen. Man erhält dadurch einen unterschiedlichen Zugang zu ganz diversen Lösungsansätzen.

Die Wahl des Lösungsanbieters könnte auch davon abhängen, in welchem Lebenszyklus sich die zu evaluierende Technologie befindet.

### 12.4.2 Identifikation potentieller zusätzlicher Lösungsanbieter

Im Konzept wurde in unserem Beispiel davon ausgegangen, dass eine integrierte CRM-Lösung klar zu favorisieren wäre – allenfalls ergänzt durch gewisse autonome Subsysteme.

Es wurde aber noch nicht absolut klar festgelegt, wie die künftige Systemarchitektur auszusehen hätte. Dies wurde bewusst offen gelassen. Es bietet sich nämlich in den vielen Projekten an, die Varianten der Systemarchitektur bis zur Evaluation aufzuschieben und im Rahmen der Ausschreibung die optimale Variante zu selektionieren.

Konkret bedeutet dies folgendes: Es wäre in unserem Beispiel zu prüfen, ob die CRM-Funktionalität auf Basis eines der bereits bestehenden Systeme ergänzt werden kann. Hierfür bieten sich grundsätzlich mehrere Systeme zur Prüfung an. Neben dem nahe liegenden ERP-System (diese verfügen z.B. meistens über ein "CRM-Modul") kann auch eine Erweiterung des Mail-Systems, des Portals oder des Office-Systems in Betracht gezogen werden. Selbstverständlich müssten auch diese Varianten die jeweiligen Integrationsanforderungen und -wünsche abdecken (z.B. hinsichtlich der Adressintegration) (s. Abb. 174).

Anstatt dass man also lediglich auf die ERP-Erweiterungsvariante oder die komplette Neulösung setzt, kommen weitere zusätzliche und sogar vorbestehende Lieferanten in Betracht, welche ebenfalls mit einem Pflichtenheft zu bedienen wären.

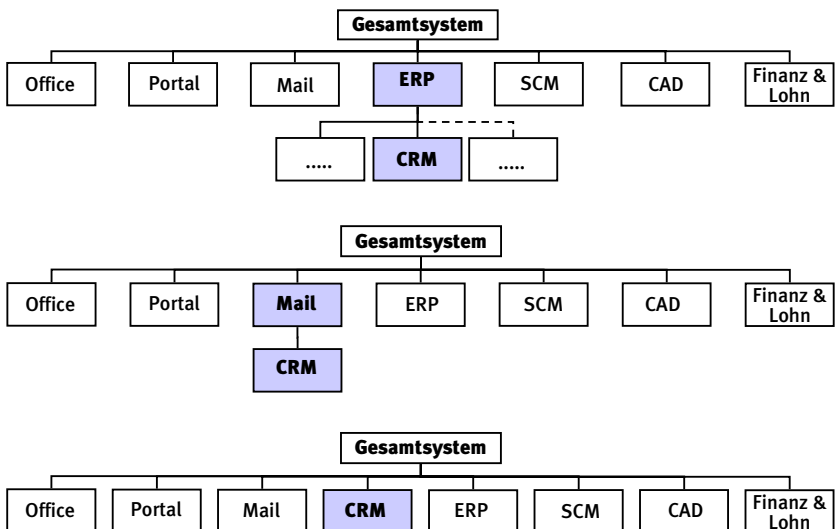


Abb. 174: Integrationsvarianten zur Erweiterung des Auswahlprozesses

### 12.4.3 Ausschreibung Grobfilter

Die Anzahl der im Vorfilter selektierten Anbieter (engl.: “Longlist”) ist häufig hoch und kann sich durchaus in der Größenordnung von 20 bis 30 Anbieter bewegen. Es ist in diesem Fall notwendig, nach dem Vorfilter mit Hilfe eines so genannten “Grobfilters” die Zahl Anbieter auf ein vernünftiges Maß zu reduzieren. Der hierfür notwendige Aufwand sollte sich sowohl auf Seite des evaluierenden Unternehmens als auch auf Seite der Anbieter noch möglichst gering halten.

Ein geeignetes Vorgehen besteht darin, zunächst ein konzises Grobpflichtenheft in Form einer kurzen Checkliste zu erstellen, welches ausschließlich die Hauptkriterien und Musskriterien enthält. Dies ermöglicht die sehr effiziente Reduktion der in Frage kommenden Anbieter, abhängig von der Erfüllung der Musskriterien (vorhandene Funktionsmodule, Referenzen, Standort des Anbieters etc.). Selbstverständlich könnte man auch direkt allen Anbietern ein umfangreiches Pflichtenheft zustellen. Allerdings würde dies nichts am Umstand ändern, dass mit Hilfe eines Grobfilters eine Vorauswahl zu erfolgen hat.

Das Grobfilterverfahren birgt ein Risiko in sich. Die “ehrlichen” Anbieter könnten unter Umständen auf der Strecke bleiben, da eine vereinfachte Checkliste (z.B. Lagerverwaltung ja/nein) großen Interpretationsspielraum zulässt und gewiefte Anbieter diesen Spielraum zu ihren Gunsten ausnutzen. Das Raster eines Grobpflichtenheftes kann wie folgt aussehen (s. Abb. 175):

- |  |  |
|--|--|
| <b>I Situationsanalyse</b>                     | Kennzahlen der eigenen Firma, Organisation, Produkte, Größe.       |
| <b>II Anwendungsgebiete</b>                    | Checkliste zu den Mussfunktionen und Mussanwendungen.              |
| <b>III Allgemeine technische Anforderungen</b> | Hardware, etwaige Schnittstellen.                                  |
| <b>IV Technische Voraussetzungen</b>           | Betriebssysteme, Datenbanksysteme, Hardwareerfordernisse.          |
| <b>V Kostenrahmen</b>                          | Schätzungen des Anbieters zu den wahrscheinlichen Kosten.          |
| <b>VI Fragen zum Systemanbieter</b>            | Firmenprofil, Anzahl Installationen, Partner, Implementationszeit. |

Abb. 175: Mögliches Raster eines Pflichtenheftes für die Grobevaluation

#### 12.4.4 Ausschreibung Feinfilter

Nach erfolgter Grobauswahl kann nun die umfangreiche Ausschreibung mit Hilfe eines detaillierteren Pflichtenheftes erfolgen. Dieses verlangt von den Anbietern bereits mehr Aufwand zur Angebotserstellung. Die Anbieter sehen in diesem Stadium aber auch bereits größere Chancen, einen Kunden zu gewinnen. Sie sind daher auch bereit, mehr in die Beantwortung des Pflichtenheftes und damit in die Akquisition des Kunden zu investieren. Das Raster eines Pflichtenheftes muss sich nicht grundsätzlich vom Grobpflichtenheft unterscheiden. Vielmehr werden alle Gebiete weiter ausgebaut. Idealerweise gibt man dem Anbieter bereits in der Ausschreibung ein Vertragsentwurf ab, zu welchem sich der Anbieter verpflichten muss. Damit verhindert man, dass die Anbieter mit eigenen Geschäftsbedingungen aufkreuzen, welche nicht im Sinne des Kunden liegen. Ein Raster kann typischerweise wie folgt aussehen (s. Abb. 176):

##### **I Situationsanalyse**

Hersteller, Systemabgrenzung, Ist-Zustand, bestehende Informatik, Aufgaben- und Organisationsmodelle.

##### **II Explizite Zielsetzungen**

Generelle Zielsetzungen, Zielsetzungen bezüglich Flexibilität.

##### **III Anwendungsgebiete**

Prozessmodelle (inkl. Besonderheiten des Unternehmens).

Systemfunktionen (Funktionsmodelle mit Checklisten).

Wichtigste Datenobjekte (Mengen und Häufigkeiten).

##### **IV Allgemeine Funktionen**

Benutzungsoberfläche, Sicherheit, Sprachfähigkeit, Numerierung, etc.

##### **V Technisches Lösungskonzept**

Hardware, Netzwerk, Datenbank, Schnittstellen, Antwortzeiten, etc.

##### **VI Prioritäten und Einführungskonzept**

Zeitplan, Datenübernahme (wichtig), etc.

##### **VII Aufbau des Angebots**

Inhaltsstruktur, Raster, verbindliche Antworten verlangen, Geheimhaltung.

##### **VIII Fragen zum Systemanbieter**

Merkmale des Anbieters, Referenzen, Zusage zu den abgegebenen Vertragsvorlagen.

Abb. 176: Mögliches Raster eines Pflichtenheftes für die Feinevaluation



### 12.4.5 Präsentationen, Szenarien und Referenzbesuche

#### Szenariobasierte Evaluation

Eine gute Möglichkeit zur Überprüfung des Systemumfanges besteht darin, im Rahmen von Präsentation der Software beispielhafte Prozesse gewissermaßen “End-to-end” im Sinne von Szenarien durchzuspielen. Der Anbieter sollte also nicht nur die übliche Marketingpräsentation darbieten müssen. Szenarien lassen die Überprüfung der Modulintegration zu und zeigen darüber hinaus, inwieweit die wichtigsten Funktionen abgedeckt sind.

Besonders negativ muss bewertet werden, wenn Prozesse nur fix ablaufen und keine Handlungsspielräume bestehen. Und hier liegt ein erster Schwachpunkt vieler Informationssysteme: Sie gehen fälschlicherweise davon aus, dass es nur einen idealtypischen Prozess gibt, welchen es nur im System zu implementieren gilt. Gelegentlich wird mit Hilfe von IT-Lösungen sogar das Ziel verfolgt, “Ordnung” in die komplexer gewordenen Betriebsabläufe zu bringen. Das starre Vorschreiben von betrieblichen Abläufen in einem Informationssystem führt unweigerlich dazu, dass die Benutzer bekanntlich versuchen, das Systemhindernis zu “überlisten”. Dazu gehört auch das häufig anzutreffende Phänomen, dass gewisse Daten nur deshalb erfasst werden, weil es das System “so will”.

In Zukunft müssen vielfältige unternehmensspezifische Anpassungen rasch und einfach vorgenommen werden, und zwar aufbauend auf flexiblen Grundstrukturen. Vorteilhafterweise sollte es den Unternehmen oder einzelnen Abteilungen möglich sein, Modifikationen selbständig und trotzdem releasefähig vorzunehmen. Aber auch Zusatzfelder, firmenspezifische Belege und Dokumente müssen durch das Unternehmen zukünftig selbst definiert werden können. Auch dieser Punkt kann im Rahmen von Szenarien effizient überprüft werden.

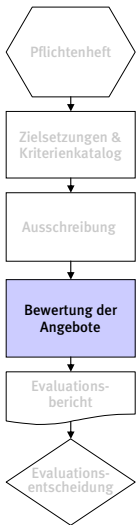
#### Referenzbesuche

Einen guten Eindruck über die Vor- und Nachteile eines Systems bilden die zwingend notwendigen Referenzbesuche. Diese lassen Fragen nach dem System im Produktiveinsatz zu und ermöglichen Fragen betreffend der Anpassungsfähigkeit und Flexibilität im Betrieb. Zusätzlich können wichtige Tipps im Umgang mit dem Lieferanten in Erfahrung gebracht werden.

Referenzbesuche erfolgen idealerweise ohne Begleitung des Anbieters, denn dies ermöglicht eine bessere Möglichkeit zu kritischen Fragen. Die Anbieter werden es verstehen und es ist letztlich ja auch im Sinne der Begrenzung des Angebotsaufwands des Anbieters. Über die Referenzbesuche sollte ebenfalls Protokoll geführt werden; das Resultat der Beurteilung hat später in die Nutzwertanalyse einzufließen.

## 12.5 Bewertung der Angebote

### 12.5.1 Nutzwertanalyse zur Bewertung der Angebote



Nachdem die Angebote der verschiedenen Anbieter eingegangen sind, müssen sie im nächsten Schritt auf ihre Eignung hin untersucht werden.

Wie in jedem Bewertungsschritt, nimmt man auch im Rahmen der Evaluation der Angebote eine Nutzwertanalyse vor (s. Abb. 177; Abb. 111). Die Nutzwertanalyse stellt den errechneten Nutzen einer Lösung den zu erwartenden Kosten gegenüber (z.B. SVD 1985, Haberfellner et al. 2002, Zehnder 2002).

Hierfür wird der oben erstellte Kriterienkatalog und die Gewichtung als Grundlage genommen. In vielen Fällen wird der Kriterienkatalog auch erst nach dem Eintreffen der Angebote zur Bewertung erstellt. Dies hat den Nachteil, dass man unter Umständen auf Kriterien trifft, welche man gerne beurteilt hätte, die bei der Ausschreibung aber nicht berücksichtigt wurden. Andererseits wird man einen früher erstellen Kriterienkatalog realistischere bei der Beurteilung ebenfalls nochmals modifizieren. Dies ist dann legitim, wenn es zu einer besseren Entscheidungsfindung führt. Nicht legitim wäre es, wenn man die Kriterien lediglich daher anpasst, weil die favorisierte Lösung nicht den gewünschten Wert erreicht. In diesem Fall dürfte man sich höchstens fragen, weshalb man eine bestimmte Lösung favorisiert und ob die Gewichtung diese Wünsche allenfalls nicht genügend berücksichtigt hat.

Gewicht in %	G1	G2	G3	GA	Athena AG			Neptun AG			Zeus GmbH		
<b>Funktionale Abdeckung</b>	<b>35</b>				Pkt.	Pkt.*GA		Pkt.	Pkt.*GA		Pkt.	Pkt.*GA	
Kontakte & Termine		30	50	5.25	4	21		6	31.5		6	31.5	
Adressen verwalten		50	50	5.25	4	21		8	42		7	37	
Kontakte verwalten		30											
Kampagnen			25	2.63	5	13	53%	8	21	72%	7	18.5	63%
Marketingplan			50	5.25	6	31		10	52.5		6	31	
Zielgruppe festlegen			25	2.62	3	8		6	15.75		5	13	
Budget erfassen			40										
Vertriebsplanung			50	7	6	42		7	49		7	49	
Aussendienstplanung			50	7	7	49		6	42		6	42	
Besuchsberichte													
<b>Flexibilität, Konfigurat.</b>	<b>20</b>												
Modulararchitektur		50						8	32		2	8	
Mandantenfähigkeit		40	4	4	16			8	48		3	18	
Subsystemfähigkeit		60	6	3	18	47%							35%
SW-Konfiguration		50											
Parametrisierbarkeit		50	5	6	30			6	30		5	25	
Flexibilität		50	5	6	30			6	30		4	20	
<b>Technische Konzeption</b>	<b>20</b>	100		20	8	160	80%	7	140	70%	5	100	50%
<b>Einführung &amp; Vorgehen</b>	<b>5</b>	100		5	3	15	30%	7	35	70%	4	20	40%
<b>Anbieter</b>	<b>20</b>	100		20	6	120	60%	7	140	70%	5	100	50%
<b>Gesamtpunktzahl</b>		100		100		<b>574</b>			<b>709</b>			<b>513</b>	
<b>Jahreskosten (Betrieb &amp; Abschreibung)</b>						<b>110</b>			<b>125</b>			<b>285</b>	
<b>Nutzwert (Punkte je Kosteneinheit)</b>						<b>5.22</b>			<b>5.67</b>			<b>1.81</b>	
<b>Rang</b>						<b>2</b>			<b>1</b>			<b>3</b>	

Abb. 177: Nutzwertanalyse auf Basis des Kriterienkatalogs

### 12.5.2 Kosten-/Nutzenverhältnis der Angebote

Nachdem der Nutzen mit Hilfe der Nutzwertanalyse ermittelt wurde, müssen in einem nächsten Schritt für jede Lösung die Kosten ermittelt werden. Idealerweise werden hierfür die *Gesamtjahreskosten* der jeweiligen Angebote herangezogen. Man wird also die Gesamtjahreskosten je Variante aufgrund der vorliegenden Angebote grob abschätzen. Die Gesamtjahreskosten errechnen sich aus der Summe der jährlichen Betriebskosten und der notwendigen Abschreibung der Investition.

Nachdem für jedes Angebot ein Wert feststeht, welcher die Gesamtjahreskosten der Lösung repräsentiert, kann in einem nächsten Schritt das Verhältnis zwischen dem Nutzen und den Gesamtjahreskosten berechnet und grafisch dargestellt werden (s. Abb. 178). Dieses Verhältnis steht für die Güte eines Angebots. Je höher der Nutzwert auf der Vertikalen und je kleiner die Kosten auf der Horizontalen, desto besser ist dieses Verhältnis. Es drückt sich in der Steigung der Gerade aus – je steiler die Gerade, desto besser ist das Angebot.

Mit Hilfe einer Sensitivitätsanalyse lässt sich die Stabilität des Resultats überprüfen. Zunächst werden die Punktevergaben und Prozentgewichte leicht verändert und anschließend wird analysiert, wie sensitiv die erreichte Punktzahl auf die Veränderungen reagiert. Wenn sich die Rangfolge der Anbieter verändert, dann ist dies ein Hinweis dafür, dass die Wahl nicht eindeutig klar zu sein scheint.

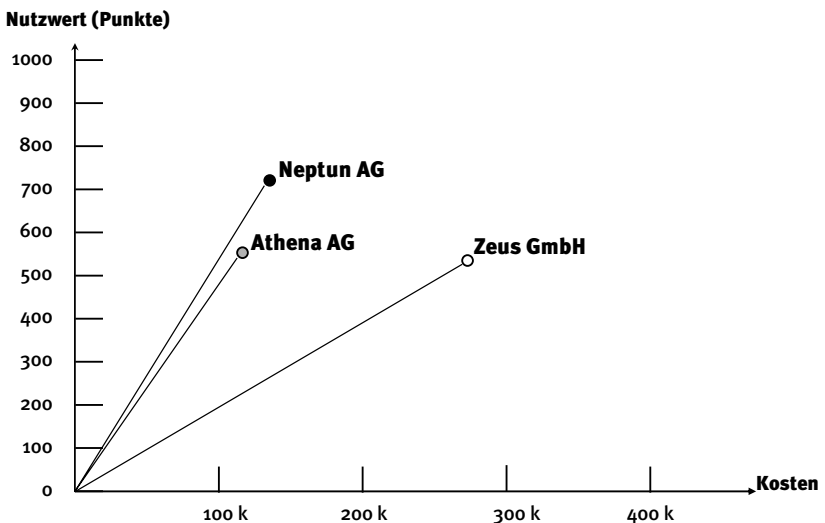


Abb. 178: Darstellung des Nutzwertes (Verhältnis von Nutzen zu Kosten)

### 12.5.3 Stärken- und Schwächenprofil der Angebote

In Ergänzung des Kosten-/Nutzenverhältnis zur Berechnung der Rangfolge der Angebote wird in der Regel ein Profil bezüglich der Stärken und Schwächen jedes Angebots erstellt (s. Abb. 179).

Dieses Profil beschränkt sich lediglich auf den Nutzwert und weist keine Kosten aus. Die Struktur des Profils orientiert sich an den Kriteriengruppen des Kriterienkatalogs, welche sich in der Vertikalen widerspiegeln. Das Diagramm zeigt auf, dass die Lösung von Neptun bei der Bewertung relativ konstant am besten abschneidet. Lediglich in der technischen Konzeption liegt wurde die Lösung von Athena höher bewertet.

Selbstverständlich könnte bei dieser Darstellung eine noch feinere Unterteilung gewählt und eine weitere Betrachtungsebene verglichen werden. Keinen Sinn macht es in der Regel, wenn jedes Einzelkriterium dargestellt wird. Dazu könnte man ja direkt auf die Evaluationsmatrix zurückgreifen.

In Ergänzung zu dieser Darstellung sollte eine verbale Auflistung der jeweiligen Hauptstärken und Hauptschwächen je Anbieter erfolgen. Dies erleichtert die Begründung, weshalb ein bestimmter Anbieter die gewünschte Bewertung verpasst hat und liefert das notwendige Argumentarium bei allenfalls unangenehmen Absagen. Die Anbieter investieren teilweise recht hohen Aufwand in eine Evaluation und sind natürlich interessiert, über die Gründe des Ausscheidens zu erfahren.

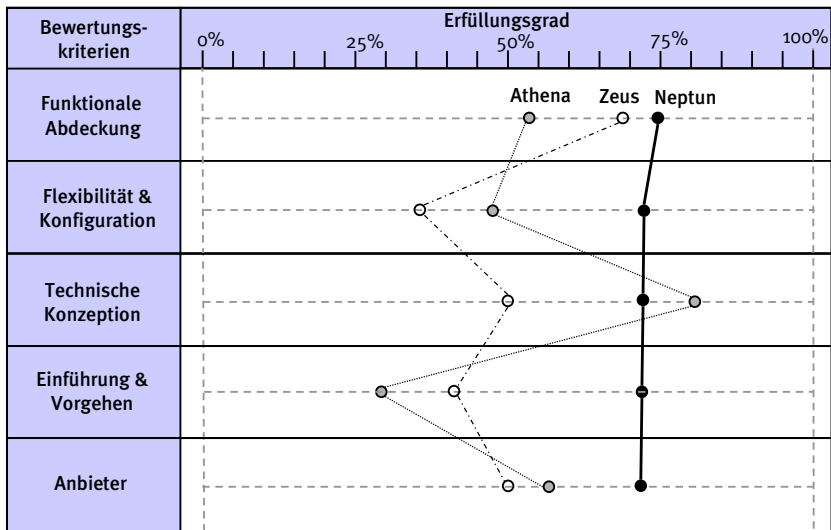
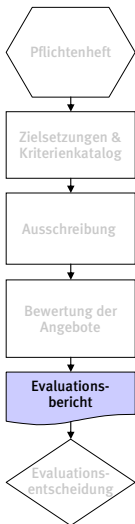


Abb. 179: Stärken-/Schwächenprofil der Angebote

## 12.6

## Evaluationsbericht



Die Struktur eines Evaluationsberichtes orientiert sich idealerweise einmal mehr am Problemlösungszyklus und umfasst die Beschreibung der Ausgangssituation, der Ziele, der alternativen Angebote und die Auswahl.

**I ZUSAMMENFASSUNG****II AUSGANGSLAGE**

- **Ausgangslage und Terminplan der Evaluation**
- **Evaluationsteam**

**III EINGEGANGENE ANGEBOTE**

- **Übersicht über die eingegangenen Angebote**

**IV EVALUATION**

- **Evaluationsverfahren (Beschreibung Grob- und Feinfilter)**
- **Erfüllung Grobfilter (Musskriterien)**

Darstellung der Erfüllung der Musskriterien der Angebote.

- **Bewertung Feinfilter**

Darstellung des Kriterienkataloges und der Gewichtung.

- **Kostenraster**

Ermittlung der einmaligen, wiederkehrenden und Jahreskosten der eingegangenen Angebote; Nötigenfalls Normierung der Kosten bei unterschiedlichen Leistungsumfang.

**V ERGEBNISSE UND BEURTEILUNG**

- **Kosten-/Leistungsvergleich**

Darstellung einer Portfolio-Nutzwertanalyse (s. o.); Rangfolge der Angebote im Sinne eines Kosten-/Nutzenvergleiches. Darstellung des Angebotsprofiles (s. o.).

- **Stärken und Schwächen**

Tabellarische Aufstellung der Stärken/Schwächen der Angebote.

**VI RISIKOANALYSE**

- **Risiken**

Tabellarische Aufstellung der Risiken mit Maßnahmen.

**VII EMPFEHLUNGEN DES EVALUATIONSTEAMS**

- **Empfehlungen**

Empfehlung und kurze Argumentation.

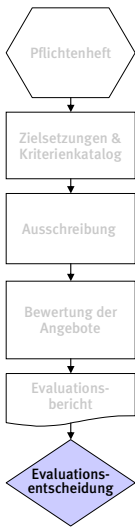
**VIII WEITERES VORGEHEN**

- **Planung**

Inhalts- und Terminplanung der nächsten Schritte.

**IX ANHANG**

## 12.7 Evaluationsentscheidung



Die definitive Evaluationsentscheidung wird in aller Regel durch den Projektausschuss gefällt. Das mit der Evaluation beauftragte Evaluationsteam bereitet die Evaluation vor und nimmt eine Punktebewertung der Angebote und Einzelkriterien vor.

Die Gewichtung der Kriteriengruppen kann aber wiederum durchaus dem Projektausschuss überlassen werden. Eine interessante Methode sowohl bei der Gewichtung als übrigens auch bei der Punktevergabe besteht darin, alle beteiligten Personen separat eine Bewertung vornehmen zu lassen und damit das Verfahren in gewissem Sinne auch demokratisch zu gestalten. Die Beteiligten werden im Unternehmen je eigene Sichtweisen einbringen. Diese unterschiedlichen Sichtweisen gilt es letztlich abzustimmen und ein Kompromiss zu finden.

Bei einem solchen Vorgehen ist sichergestellt, dass nicht die dominante Person des Gremiums alle Weichen im Alleingang stellt. Die abschließende Entscheidung zugunsten des einen oder anderen Anbieters wird auf Basis der vollständigen Evaluation vollzogen werden.

Bereich Schritt	<b>Sozio-technisches System</b>	<b>Einführung &amp; Ausbildung</b>	<b>Betrieb &amp; Unterhalt</b>	<b>Projektmanagement &amp; Controlling</b>
<b>Pflichtenheft</b>	Dokumentierte Zielsetzungen; Dokumentierte Anforderungen; Grobpflichtenheft; Vollständiges Pflichtenheft; Inhaltsraster für Offerten.	Anforderungen an Einführungszeitraum und Abfolge.	Anforderungen an Betrieb und Unterhalt.	Projektplanung; Planung Evaluation.
<b>Zielsetzungen &amp; Kriterienkatalog</b>	Gewichteter Kriterienkatalog.	Kriterien Einführung & Schulung.	Kriterien Betrieb & Unterhalt.	
<b>Ausschreibung</b>	Marktanalyse Anbieter; Checkliste für Selektion; Selektion Grobfilter (Longlist); Selektion Feinfilter (Shortlist); Dokumentierte Evaluationsszenarien; Eingeholte Referenzauskünfte.			
<b>Bewertung der Angebote</b>	Bereinigte Angebote; Evaluationsbewertung.	Bewertung Einführung & Schulung.	Bewertung Betrieb & Unterhalt.	Bewertung Projektmanagement.
<b>Evaluationsbericht</b>	Evaluationsbericht.			
<b>Evaluationsentscheidung</b>				Evaluations- entscheidung.

Abb. 180: Ergebnisse der Pflichtenheft- und Evaluationstätigkeit

## 12.8 Literatur zu Pflichtenheft und Evaluation

- Armbruster, S., Springer, J., Luczak, H. (1998): Bewertung von PPS-Systemen hinsichtlich der Erfüllung von Benutzeranforderungen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 327–375.
- Becker, M., Haberfellner, R., Liebetrau, G. (2000): EDV-Wissen für Anwender: ein Informatik-Handbuch für die Praxis (11. Aufl.). Zürich: Industrielle Organisation.
- Brynjolfsson, E. (1993): The Productivity Paradox of Information Technology: Review and Assessment. In: Communications of the ACM, Dec.
- Daenzer, W., Huber, F. (Hrsg.) (1997): Systems Engineering: Methodik und Praxis, 9. Aufl. Zürich: Verlag Industrielle Organisation.
- Grupp, B. (1994). Standard-Software richtig auswählen und einführen: Mit System zur kostengünstigen und umfassenden DV-Lösung. Wuppertal: TAW-Verlag.
- Grupp, B. (1995). EDV-Pflichtenheft zur Hardware- und Softwareauswahl: Praktische Anleitung. Köln: TÜV Rheinland.
- Holz auf der Heide, B. (1993): Welche software-ergonomischen Evaluationsverfahren können was leisten? In: German Chapter of the ACM, Band 39: Software-Ergonomie '93. (Rödiger, K.H.; Hrsg.); Stuttgart: Teubner. 157–171.
- Kaiser, H., Paegert, C., Schotten, M. (1998): Auswahl von PPS-Systemen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 292–326.
- Lechner, S. (1993). Betriebstypologische Auswahl von Standardsoftware zur Produktions-Planung und Steuerung (PPS). Fortschrittsberichte VDI Reihe 20, Nr. 89. Düsseldorf: VDI Verlag.
- Oppermann, R., Murchner, B., Reiterer, H. & Koch, M. (1992): Software-ergonomische Evaluation. Der Leitfaden EVADIS II. Berlin: DeGruyter.
- SVD (1985): Evaluation von Informatiklösungen: Verfahren – Methoden – Beispiele. Schweizerische Vereinigung für Datenverarbeitung (Hrsg.). Bern/Stuttgart: Haupt.
- Zehnder, C.A. (2002): Informatik-Projektentwicklung: Eine Einführung für Informatikstudenten und Praktiker, 4. Aufl. Zürich: vdf Hochschulverlag.





# 13

## Verifikation und Vertrag

### Kapitelinhalt

- Bestimmung des Verifikationsbedarfs und Analyse der Verträge
- Ziele an Hardware-, Software-, Dienstleistungs- & Supportverträge
- Projektplanung und Konzeption des gesamten Vertragswerkes
- Unterzeichnung der Verträge

Den nächsten Schritt im Anschluss an die Auswahl des Lösungsanbieters stellt die Verifikations- und Vertragsphase dar. Es hat als sinnvoll erwiesen, noch vor der definitiven Vertragsunterzeichnung einen Verifikations-schritt vorzusehen.

Auch ein bereits evaluierter Anbieter wird also im Rahmen einer Verifikation gewisse noch offene Fragen über sich ergehen lassen müssen. Die Verifikation kann auch darin bestehen, einen bezahlten Prototypen erstellen zu lassen, um heikle Punkte zu überprüfen. Dies bedeutet, dass dem gemäß Evaluation zweitplazierten Lösungsanbieter noch keine definitive Absage erteilt werden sollte. Im Rahmen der Verifikation kann gleichzeitig mit der Vertragskonzeption und -ausgestaltung begonnen sowie die Projektplanung für die spätere Umsetzung an die Hand genommen werden. Diese Aktivitäten dienen letztlich ebenfalls der Verifikation.

Ein IT-Projekt beinhaltet in der Regel nicht nur einen Vertrag, sondern ein ganzes Vertragswerk für Hardware, Software und Dienstleistungen.

## 13.1 Inhalt und Vorgehen

### 13.1.1 Inhalt

Der Schwerpunkt des Verifikations- und Vertragsschrittes liegt in der Konzeption und formellen Absicherung des Projektes und der späteren Betriebs- und Unterhaltsphase. Nach der Evaluation ist der Anbieter zwar gewählt, trotzdem besteht aber häufig noch ein Zusatzbedarf, gewisse kritische Punkte zu verifizieren, bevor der Vertrag unterzeichnet wird.

Nachdem diese Punkte geklärt sind, sollten im Sinne des Denkens in Varianten unterschiedliche Vertragskonzepte in Betracht gezogen werden. Es stellt sich analog wie beim Hausbau beispielsweise die Frage, ob ein Fixpreisvertrag oder ein Vertrag mit Verrechnung nach Aufwand bevorzugt werden sollte.

### 13.1.2 Vorgehen

Zu Beginn sollte zunächst eine Standortbestimmung im Sinne einer Situationsanalyse vorgenommen werden. Diese besteht darin, die möglichen Risiken des gewählten Anbieters zu identifizieren. Allenfalls vom Anbieter vorgelegte Verträge sollten auf ihren Inhalt hin analysiert werden. Sodann gilt es, die vorhandenen vertragsrelevanten Unterlagen wie Pflichtenheft, Angebot und Briefverkehr zu konsolidieren. Ein zweiter Teil der Analyse besteht in der Vorbereitung von Entscheidungen hinsichtlich der Konzeption der Dienstleistungen und Wartungsleistungen (s. Abb. 181).

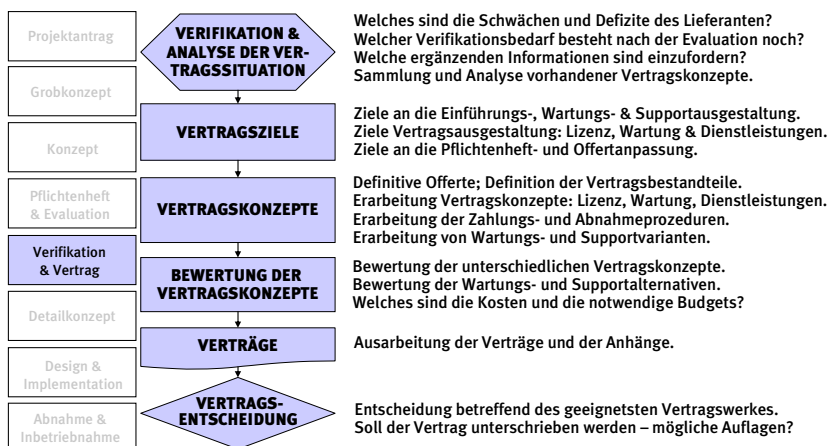


Abb. 181: Vorgehensschritte der Verifikations- und Vertragstätigkeit

## 13.2 Verifikation und Analyse der Vertragssituation

### 13.2.1 Verifikation



Im Rahmen der Evaluation ist das Evaluationsteam zeitlich und inhaltlich meist stark beansprucht. Das Evaluationsteam ist daher häufig nicht wirklich in der Lage, alle wichtigen Fragestellungen in der tatsächlich wünschenswerten Tiefe abzuklären. Wenn also die Entscheidung intern zugunsten eines Anbieters gefallen ist, so sollte man dem zweitplazierten Anbieter bis zur definitiven Vertragsunterzeichnung keinesfalls eine klare Absage erteilen. Dies würde dem erstplazierten Anbieter eine bessere Verhandlungsposition im Rahmen der Vertragsverhandlungen bieten.

Es hat sich daher als ideal erwiesen, nach abgeschlossener Evaluation einen expliziten Verifikationsschritt einzuschalten. In diesem Schritt sollen risikoreiche Punkte identifiziert und nochmals klar hinterfragt werden. Die Verifikation kann auch darin bestehen, dass man einen allenfalls bezahlten Verifikationsprototyp erstellen lässt. Im Beispiel wurde ein Prototyp für ein CRM-Portalsystem erstellt und anhand dieses Verifikationsprototyps konnte Klarheit hinsichtlich der Möglichkeiten des Systems und des Lieferanten gewonnen werden (s. Abb. 182).

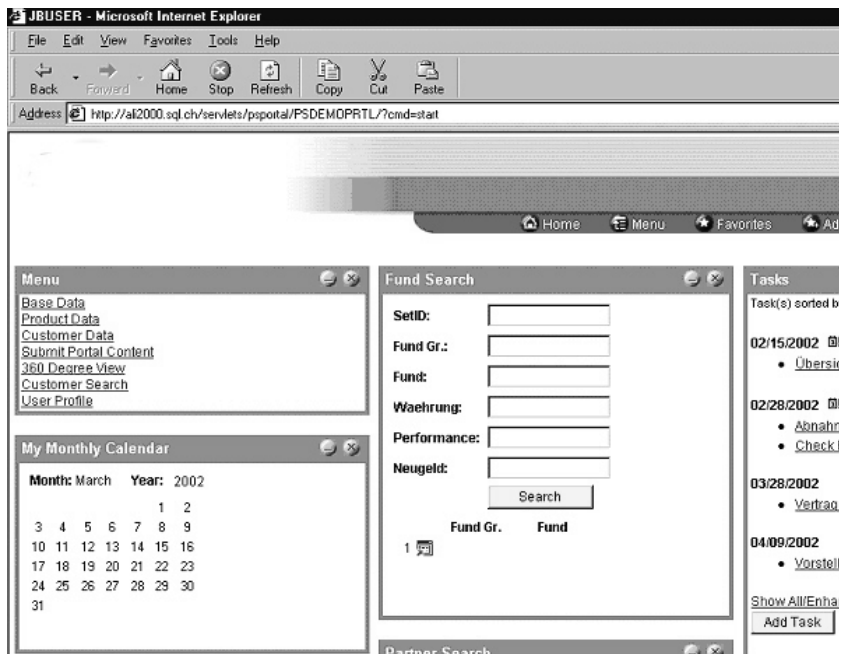


Abb. 182: Ausschnitt aus einem CRM-Verifikationsprototyp

### 13.2.2 Analyse der Vertragssituation

#### Klarheit der Spezifikationen

Das Vertragskonzept wird davon abhängen, in welchem Detaillierungsgrad die Spezifikationen im Pflichtenheft dargelegt wurden. Besteht eine klare Spezifikation der Anforderungen, so wird der Anbieter in der Lage sein, einen Vertrag mit klarer Preisangabe zu unterbreiten.

In gewissen Projekten liegt allerdings keine klare Spezifikation vor. In diesen Fällen wird man vorsehen müssen, zunächst eine Spezifikationsphase einzuschalten und den Vertrag für die Umsetzung erst anschließend zu definieren. Vorsicht ist geboten, wenn klare Spezifikationen vorliegen, der Anbieter aber trotzdem auf einer Detaillierung besteht. Dies könnte ein Hinweis sein, dass dieser die geforderten Funktionalitäten nicht abzudecken vermag oder zu wenig Erfahrung mitbringt.

#### Analyse der Risiken

Bevor man an die explizite Vertragsausarbeitung gehen kann, sollte man sich nochmals sehr bewusst mit der Situation des Anbieters und im Unternehmen auseinandersetzen. Es ist zu überlegen, welche Risiken vorhanden sind und vertraglich abzusichern wären. Insbesondere die Risiken können sich von Projekt zu Projekt stark unterscheiden. Sie sind abhängig von der Anbieterkonstellation, der gewählten Softwarelösung, deren Reifegrad und einer Reihe weiterer Faktoren.

#### Bereinigung von Pflichtenheft und Offerte

In der Regel werden das Pflichtenheft, mit den darin enthaltenen Spezifikationen, und das Angebot einen impliziten Bestandteil des Vertrags bilden. Im Vertrag wird auf die Erfüllung des Pflichtenheftes verwiesen. Häufig ist allerdings unklar, welche Teile nun Gültigkeit erlangen, da im Rahmen der Evaluation Zusatzgespräche geführt und Änderungen diskutiert wurden. Es gilt daher, diese Diskrepanzen zu bereinigen und das Pflichtenheft sowie die Offerte in eine absolut verbindliche Form zu bringen.

#### Analyse der vorhandenen Verträge und der Vertragssituation

Falls Vertragsentwürfe seitens des Anbieters vorliegen, so werden diese nun einer sauberen Analyse unterzogen. Ein besonders Augenmerk sollte Allgemeinen Geschäftsbedingungen (AGB) gegeben werden. In der Regel diktiert der wirtschaftlich stärkere Partner dem schwächeren Partner die allgemeinen Bedingungen. Der Begriff "Allgemeine Geschäftsbedingungen" versucht auch zu suggerieren, dass diese Bedingungen unabänderlich wären und nicht modifiziert werden könnten. Dem ist entgegenzuhalten, dass die Vertragspartner in Tat und Wahrheit frei sind, ihre Verträge zu definieren. Lediglich gesetzliche Auflagen dürfen nicht verletzt werden.

## 13.3

**Vertragsziele**

Ganz entsprechend dem Problemlösungszyklus wird man im Anschluss an die Analyse der Ausgangssituation auch im Rahmen des Vertrags als nächstes die vertraglichen Anforderungen für den Projekt- und Unterhaltsvertrag festlegen.

Die vertraglichen Ziele betreffen Punkte zur Absicherung des Projekterfolges und Ziele an eine reibungslose und effiziente Wartung nach der Systemeinführung. Der Vertrag dient außerdem dazu, alle im Rahmen der Evaluation entdeckten Stärken und Schwächen nochmals konkret abzusichern und die etwaigen Zusicherungen des Lieferanten zu dokumentieren.

**Vertragsziele zur Projektabsicherung (Software & Dienstleistung)**

Die primären Ziele an den Projektvertrag betreffen zunächst die Sicherstellung der Termine, die Sicherstellung der geforderten Funktionalitäten und der Sicherstellung des vorgesehenen Kostenumfanges.

Die Ziele haben primär die im Rahmen der Analysephase aufgedeckten Risiken aufzunehmen. Dies könnten Ziele im Hinblick auf die Einbindung der eigenen Mitarbeiter, deren zeitliche Belastung oder aber auch um die Einbindung von fähigen Mitarbeitern auf Seiten des Lieferanten gehen.

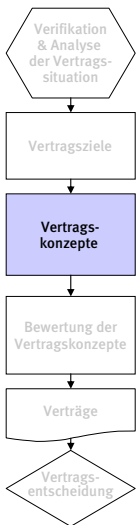
Eine besonders wichtige Zielsetzung betrifft die Fragestellung, wie mit einem größeren Anbieterkollektiv umzugehen wäre: Soll die Gesamtverantwortung allenfalls im Sinne eines Generalunternehmervertrags abgesichert werden? Selbstverständlich wäre es für eine solche Forderung hilfreich, wenn diese Fragestellung bereits im Rahmen der Ausschreibung geklärt wäre und im Angebot bereits enthalten ist. Es kann aber immer sein, dass ein Anbieter von sich aus ein ganzes Kollektiv mit einbringt, obwohl dies nicht entsprechend vorgesehen war.

**Vertragsziele zu Support und Wartung**

Während die obigen Punkte Zielsetzungen im Zusammenhang mit der Projektabwicklung und mit Softwarelizenzen bzw. -kauf festhalten, muss gleichbedeutend mit diesem Vertrag auch bereits über die Phase des Supports und der Wartung bzw. der Updates nachgedacht werden. Vielfach sind die Wartungsgebühren derart hoch, dass sie kostenmäßig mindestens gleichbedeutend den Softwarelizenzen sind.

Man wird also Ziele festlegen wollen, wie lange der Support sicherzustellen ist und wie mit Problemen im Umgang mit der Applikation verfahren werden sollte. Es betrifft dies insbesondere die erwarteten Reaktionsfristen seitens des Anbieters.

### 13.4 Vertragskonzepte



Im idealen Fall werden die Vertragsentwürfe den Anbietern bereits im Rahmen der Ausschreibung abgegeben und damit dem Lieferanten vorge-schrieben (s. Kapitel 12). Die Anbieter werden vorgegebene Verträge näm-lich am ehesten in der Angebotsphase akzeptieren, da sie ja das Projekt gerne gewinnen und daher zu größeren Konzession bereit sind. Bereits nach erfolgter Evaluation hat sich ihre Stellung verbessert und die Wahl-freiheit des Kunden ist eingeschränkt. Größere Unternehmen (Behörden, Konzerne) verfügen sogar häufig über eigene Informatikverträge und sind nicht gewillt, die Verträge der Anbieter zu übernehmen. Kleinere Unter-nehmen haben diese Möglichkeit und Stärke nicht. Sie können allenfalls auf ausgewogene Standardverträge zurückgreifen, wie diese teilweise von nationalen Informatikvereinigungen abgegeben werden. Der Kunde muss sich zunächst überlegen, ob er das Projekt mit einem Generalunternehmer abwickelt oder ob er Einzelverträge vorzieht.

#### Konzept des Generalunternehmervertrags

Die Beschaffung eines integrierten Informationssystems umfasst eine Vielzahl unterschiedlicher Komponenten, wie Betriebssysteme, Datenban-ken, SW, HW und Netzwerk. Um sicherzustellen, dass sich diese Kompo-nenten zu einem funktionierenden Ganzen zusammenfügen, wird der Auf-trag häufig einem so genannten Generalunternehmer übertragen. Es soll primär gesichert werden, dass der Kunde nicht mit den Subunternehmern einzeln verhandeln muss und sich die Subunternehmer bei Problemen die Schuld gegenseitig in die Schuhe schieben. Der Generalunternehmer trägt dann die Verantwortung und ist die primäre Ansprechperson. Der Gene-ralunternehmer wird in einem Generalunternehmervertrag verpflichtet.

Ein Generalunternehmervertrag besteht aus Einzelverträgen, welche die genannten Bereiche der Software-Lizenzrechte, Dienstleistungen etc. re-geln. Im Unterschied zu isolierten Einzelverträgen werden sie durch den Generalunternehmervertrag zusammengehalten. Generalunternehmerver-träge richten sich am Vertragstyp des Werkvertrages aus (s. u.).

#### Konzept der Einzelverträge

Als Alternative zum Generalunternehmervertrag kann man sich dazu ent-schließen, mit den beteiligten Lieferanten so genannte "Einzelverträge" abzuschließen. Man wird also mit dem Softwarelieferanten den Software-Lizenzvertrag und dem Beratungshaus einen separaten Dienstleistungs-vertrag abschließen. Dem höheren Risiko dieser Variante stehen allenfalls auch geringe Kosten gegenüber. Dies mag daher dort sinnvoll sein, wo die eingesetzte Technologie bekannt ist. Der vom Generalunternehmer übli-cherweise eingerechnete Risikozuschlag entfällt bzw. wird vom Kunden selbst getragen.

### 13.4.1 Konzept des Software Lizenzvertrags

Bestandteil jedes Vertragskonzeptes, unabhängig davon, ob im Rahmen eines Generalunternehmer- oder Einzelvertrags, ist der Software-Lizenzvertrag. Die umgangssprachliche Meinung, dass Software gekauft würde, ist in diesem Zusammenhang eher irrtümlich. Software wird in den meisten Fällen nicht gekauft, sondern lizenziert. Es wird daher auch kein Software-Kaufvertrag abgeschlossen. Software wird lediglich lizenziert und es bedarf eines Lizenzvertrags. Wenn man also davon spricht, man kaufe "Microsoft Word", so ist dies im eigentlichen Sinn nicht korrekt. Man erkaufte sich das immaterielle Recht, den von Microsoft zur Verfügung gestellten Programmcode in einem streng definierten Rahmen zu nutzen. Gekauft wird bestenfalls die Schachtel, in der die Software verpackt ist. In vielen Fällen erhalten die Kunden vom Lizenznehmer auch keine Informationen zum eigentlichen Programmcode und dürfen Lizenzen gelegentlich auch nicht weiterverkaufen.

#### Lizenzierungsmodelle

In Zusammenhang mit der Software-Lizenz spricht man von den Lizenzierungsmodellen. Diese bestimmen die Anzahl der Benutzer und die damit verbundene Höhe der Lizenzkosten. In der Regel wird zwischen den Modellen Named User, Concurrent User und Site Licence unterschieden.

Das Modell Named User berechnet für jeden namentlich genannten Benutzer eine Lizenzgebühr. Diese ist unabhängig davon, wie häufig der Benutzer mit dem System arbeitet und dieses nutzt. Das Modell Concurrent User berechnet für eine bestimmte Anzahl von gleichzeitigen Benutzern die Lizenzgebühr. Es wird offen gelassen, wie viele Benutzer namentlich auf dem System angelegt werden und dieses sporadisch benutzen. Entscheidend ist nur, dass nicht mehr Benutzer als vorgesehen gleichzeitig im System eingeloggt sind. Das Modell einer Site Licence ermöglicht es einem Unternehmen, beliebig viele Benutzer aufzuschalten. Es wird in der Regel eine Gebühr entrichtet, welche von der Unternehmensgröße abhängig gemacht wird.

#### Softwarekauf

Ein eigentlicher Software-Kaufvertrag wird nur dann zur Anwendung gelangen, wenn eine bestehende und bereits programmierte Softwarelösung vollständig und mit allen Urheberrechten bzw. Verwertungsrechten an den Käufer übergeht.

Dies wird selten der Fall sein. Der Kaufvertrag wird damit primär im Bereich der Hardware- und Netzwerkkomponenten zur Anwendung gelangen.

### 13.4.2 Konzept des Dienstleistungsvertrags

Die Lizenzierung einer Software reicht nicht aus, um ein funktionierendes System zu erhalten. Es bedarf in den meisten Fällen einer umfassenden Implementation, Konfiguration und Modifikation der einzelnen Softwarekomponenten. Sie sind auf den Betrieb abzustimmen und zu einem funktionierenden Ganzen zusammenzufügen.

Diese Form der Dienstleistung ist nicht im Software Lizenzvertrag enthalten, sondern erfordert einen separaten Dienstleistungsvertrag (Projektvertrag). Der Titel "Dienstleistungsvertrag" bedeutet allerdings noch nicht automatisch, dass es sich bei diesem Vertrag im rechtlichen Sinn um einen "Auftrag" oder um einen so genannten "Werkvertrag" handelt.

Das Recht der meisten Länder unterscheidet zwischen diesen beiden Vertragsformen. Inwieweit es sich bei der Beauftragung eines Lieferanten um die eine oder andere (bzw. um eine Kombination) dieser Rechtsformen handelt, ergibt sich erst aus dem Vertragsinhalt und -text. Der alleinige Titel Werkvertrag oder Auftrag reicht zur Qualifizierung nicht aus. Die Wahl der Vertragsform kann aber große Auswirkungen im Falle eines Projektmisserfolges zeitigen.

#### Konzept des Werkvertrags

Bei einem "Werkvertrag" schuldet der Auftragnehmer eine klar definierte Leistung (ein Werk). Im Gegensatz zu einem Auftrag wird also nicht lediglich die sorgfältig durchgeführte Arbeitsleistung eingekauft, sondern ein klar spezifiziertes Endprodukt (welches daher auch vorgängig spezifiziert sein muss). Der Lieferant schuldet den Erfolg im Sinne eines erfolgreich funktionierenden Arbeitsergebnisses (erfolgreiches Werk).

Der Auftraggeber wird daher in vielen Fällen bestrebt sein, mit dem Lieferanten einen Werkvertrag abzuschließen, damit er auch tatsächlich eine funktionierende Lösung erhält. Den Aufwand für dieses Endprodukt kann man immer noch nach Aufwand entschädigen, allerdings ändert dies nichts an der Tatsache, dass ein Werkvertrag besteht.

#### Konzept des Auftrags

Bei einem Auftrag schuldet der Lieferant – erstaunlicherweise – "nur" eine sorgfältige Arbeitsleistung, nicht aber ein erfolgreiches und funktionierendes Werk. Dies ist ein fundamentaler Unterschied zum Werkvertrag und mag zunächst erstaunen, da ja eigentlich landläufig auch bei einem Auftrag ein Erfolg erwartet wird. Auch ein Auftrag kann nach Aufwand abgerechnet werden. Der Lieferant wird daher in der Regel bestrebt sein, seine Leistungen eher in Form eines "Auftrags" zu erbringen, da er damit nur sorgfältige Arbeit schuldet, nicht aber, dass das Ganze am Ende funktioniert.



### **Inhalte des Dienstleistungsvertrags**

Der Dienstleistungsvertrag regelt alle Punkte zur Abwicklung eines Projektes und umfasst u.a. typischerweise die folgenden Punkte:

*Präambel:* Die Präambel umschreibt einleitend die Grundlagen und Absichten der Vertragsparteien.

*Gegenstand des Vertrags:* Der Gegenstand beinhaltet eine genaue Beschreibung, welche Leistungen zu welchem Preis erbracht werden.

*Projektänderungen:* Eine Schwierigkeit ist der Umgang mit Änderungen. Dieser Punkt beschreibt das Verfahren gemäß Projektmanagement.

*Projektorganisation:* Bei der Projektorganisation ist darauf zu achten, dass die Kunden- und die Lieferantenseite möglichst klar getrennt werden.

*Verpflichtungen des Lieferanten:* Die Verpflichtungen des Lieferanten und dessen Fähigkeiten werden in diesem Punkt explizit genannt. Es wird klar, ob es sich um einen Werkvertrag oder um einen Auftrag handelt.

*Verpflichtung und Mitwirkung des Kunden:* Den Kunden trifft meist eine Verpflichtung zur Mitarbeit, welche umfangmäßig definiert sein sollte.

*Abnahme:* Das Prozedere der Abnahme muss im Vertrag klar definiert werden (s. Kapitel 13).

*Rechte an den Arbeitsresultaten:* Die Rechte an den Arbeitsresultaten und am Programmcode sind festzuhalten und zu dokumentieren.

*Eigentums- und Schutzrechte:* Selbst wenn Software für den Kunden erstellt wird, muss vorgängig geklärt werden, wer die Urheberrechte erhält.

*Geheimhaltung:* Üblicherweise werden klare Geheimhaltungsklauseln vorgesehen, welche dem Lieferanten eine Schweigepflicht auferlegen.

*Vergütung und Zahlungsbedingungen:* Die Vergütungen und Zahlungsbedingungen werden festgehalten. Üblicherweise wird ein bestimmter Anteil bis zur Abnahme zurückbehalten. Die Zahlung kann nach Fortschritt bzw. nach effektiven Ergebnissen erfolgen.

*Garantie und Gewährleistung:* Es ist eine Garantie- und Gewährleistungszeit vorzusehen, während welcher Fehler kostenfrei behoben werden.

*Haftung und Verzug:* Haftungsfragen (inkl. Schadloshaltung) oder Fragestellungen bei Eintritt von Gläubiger- oder Schuldnerverzug sind zu regeln.

*Zeitplanung und Meilensteine:* Wichtige Termine und Projektmeilensteine sollten ebenfalls festgehalten werden.

*Anwendbares Recht und Gerichtsstand:* Eine Regelung dieser Punkte ist vor allem bei Verträgen im internationalen Verhältnis unumgänglich.

### 13.4.3 Konzept des Hardwarevertrags

Die Beschaffung von Hardware- und Netzwerkkomponenten unterliegt in der Regel einem klassischen Kaufvertrag. Es ist dies ein Vertragstypus, mit welchem die Unternehmen in den allermeisten Fällen am wenigsten Probleme haben, da sie von Seiten des Einkaufs bereits über eine große Erfahrung verfügen. Bei einem Generalunternehmervertrag muss die Rückgabe selbstverständlich gesichert sein und der Generalunternehmer hat entsprechende Garantien abzugeben.

#### Hardwaremietvertrag

Gelegentlich stellt sich die Frage, inwiefern die Hardware besser gemietet wird und ob diese Variante nicht kostengünstiger wäre. Es kann dies insbesondere dann von Vorteil sein, wenn das Projekt über eine lange Laufzeit verfügt und zu erwarten ist, dass die Hardware am Ende der Projektlaufzeit bereits veraltet ist. Selbstverständlich stellt sich diese Frage auch nur dann, wenn der Anbieter den Kauf für das Entwicklungssystem zur Bedingung macht und dieses nicht selber zur Verfügung stellt. Das Leasing kommt primär aus steuerlichen oder Liquiditätsgründen in Betracht.

### 13.4.4 Konzept des Support- & Wartungsvertrags

Die Wartung stellt wie erwähnt einen nicht unbedeutenden Kostenfaktor dar. Die Wartungsgebühren für die Softwarewartung, welche regelmäßige Updates bei Produktneuerungen enthält, bewegen sich in der Größenordnung von jährlich 15 Prozent. Es ist zu bedenken, dass bei einem Verzicht auf eine Softwarewartung der Releasewechsel mindestens so teuer zu stehen kommt wie bei einer Wartung mit entsprechenden Updates.

Beim Support und Wartungsvertrag handelt es sich um einen freien Vertragstyp, welcher im Gesetz nicht geregelt wird. Im Gegensatz zum oben genannten Werkvertrag gibt es damit keine gesetzlich zwingenden Grundlagen, welche einzuhalten wären. Der Supportvertrag kann damit von den Vertragsparteien vollständig frei ausgestaltet werden und die Zielsetzungen des Kunden bzw. Lieferanten abdecken.

Viele Softwareanbieter stufen ihre Support- und Wartungsverträge nach unterschiedlichen Servicegraden ab (Service Levels). Es betrifft dies einerseits die Verfügbarkeit der Hotline oder die Reaktionsfrist bei Störungen.

Damit können unterschiedliche Vertragskonzepte erarbeitet werden, z.B.:

- A) Regelmäßige Softwareupdates und zu Bürozeiten eine Hotline
- B) Wartungsleistungen nur nach Aufwand und auf expliziten Abruf
- C) Gewährleistung eines hochverfügbaren Systems mit Service Level

### 13.4.5 Morphologische Matrix für den Vertrag

Im Vertragswesen eröffnen sich gemäß den oben stehenden Ausführungen wiederum eine Vielzahl von alternativen Lösungsausprägungen und Gestaltungsspielräumen. Diese gilt es bewusst auszuloten. Das Unternehmen wird eine kombinierte Wahl treffen müssen, welche seine Bedürfnisse optimal abdeckt und insbesondere von den Kosten her zu vertreten ist (s. Abb. 183).

An erster Stelle steht eine Entscheidung bezüglich des Parameters "Gesamtvertragskonzept" an. Hier eröffnen sich die unterschiedlichen Ausprägungen eines Generalunternehmervertrags oder die Kombination von Einzelverträgen.

Sodann bestehen weiter die Parameter Hardware-, Software, Dienstleistungs- und Supportverträge, deren Ausprägungen in der unten stehenden Tabelle dargestellt sind.

Wiederum ergibt sich eine Gesamtvertragskonzeption aus einem Pfad durch diese Matrix. Man wird allerdings in den wenigsten Fällen derart strukturiert vorgehen, und uns dient die Darstellung eher zum Gedankenanstoss, damit nicht vorschnell vordergründige Lösungen bevorzugt werden sollten.

<b>Ausprägung Parameter</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>
<b>Gesamtvertragskonzept</b>	<b>Generalunternehmervertrag</b>	Unabhängige Einzelverträge	...	...
<b>Hardwarevertrag</b>	<b>Kaufvertrag</b>	Mietvertrag	Leasingvertrag	...
<b>Software- &amp; Lizenzvertrag</b>	<b>Software-Lizenzvertrag</b>	Software-Werkvertrag	Application Service Provider Vertrag	Softwarekauf (Kauf des Sourcecodes)
<b>Dienstleistungsvertrag</b>	Dienstleistungsauftrag nach Aufwand	<b>Dienstleistungs-Werkvertrag mit Fixpreis</b>	Dienstleistungs-Werkvertrag nach Aufwand	Auftrag
<b>Support- &amp; Wartungsvertrag</b>	<b>Wartungs-Auftrag nach Aufwand</b>	Wartungsvertrag mit Service Levels	Outsourcing-Vertrag	....
<b>Update- &amp; Upgradevertrag</b>	Update- & Upgradevertrag	<b>Updates &amp; Upgrades einzeln kaufen</b>	keine Updates & Upgrades	...

Abb. 183: Beispielhafte Morphologie im Vertragswesen

### 13.5 Bewertung der Vertragskonzepte



Die Bewertung alternativer Vertragskonzepte wird vermutlich nur in den wenigsten Fällen aufgrund einer formalen Nutzwertanalyse erfolgen. Häufig werden die Vertragskonzepte (z.B. Kaufvertrag, Dienstleistungsvertrag, GU-Vertrag etc.) allenfalls mit Hilfe einer einfachen Argumentenbilanz gegenübergestellt und dem Projektausschuss in dieser Form vorgelegt.

Die unten stehenden Vertragsvarianten stellen beispielhaft drei Möglichkeiten für den Projektvertrag bzw. Generalunternehmervertrag dar (s. Abb. 184).

Es sei an dieser Stelle nochmals auf die Vorteile der verschiedenen Vertragstypen verwiesen. Diese Bewertung muss im Einzelfall nicht zwingend gleich sein und bedarf selbstverständlich der Anpassung.

Realistisch ist allenfalls der Umstand, dass der Generalunternehmervertrag den scheinbar höchsten Preis aufweist. Allerdings ist dies dahingehend zu relativieren, dass bei den anderen Varianten die Streuung viel größer sein wird und der vom Generalunternehmer angebotene Preis durchaus übertroffen werden kann.

Daher macht es Sinn, auch bei der Beurteilung der Verträge nicht nur auf die Kosten zu achten, sondern die Risiken bewusst in die Vertragsbewertung mit einzubeziehen.

<b>Varianten</b> <b>Merkmale</b>	<b>Generalunternehmer</b>	<b>Einzelne Werkverträge</b>	<b>Einzelne Aufträge</b>
Beschreibung der Vertragslösung	Der GU realisiert das Projekt zu einem definierten Fixpreis. Leistungen werden trotzdem idealerweise nach Aufwand abgerechnet.	Das CRM-System ist voll in das bestehende ERP-System integriert. Es baut auf derselben Datenbank auf und muss vom selben Hersteller stammen.	Diese Lösung sieht ein komplett losgelöstes System vor. Es pflegt eigene Adressen und erhält keine Angaben aus dem ERP (z.B. Umsatz).
Vorteile	Ein Ansprechpartner; klare Verantwortung; klare Preisobergrenze.	Projektfumfang flexibel anpassbar; Lieferanten können alle selbst bestimmt werden.	Geringer Aufwand zur Definition Resultat, evolutionärer Ansatz, wenn "Werk" unklar.
Nachteile	Risikozuschlag; tendenziell höhere Kosten; Gefahr, dass nur Minimum realisiert wird; Diskussionen um den Funktionsumfang.	Ansprechpartner bei Problemen unklar; Schuldzuweisungen zwischen Lieferanten müssen selbst behoben werden.	Kein Erfolg garantiert bzw. geschuldet. Der Lieferant trifft wenig Verantwortung für das Projekt.
Erwarteter Preis	Fixpreis: 400	350	200 – 400
Risikobeurteilung	geringes Risiko	mittleres Risiko	hohes Risiko

Abb. 184: Beispielhafte Argumentenbilanz für Projektvertrag des CRM-Systems

## 13.6

## Verträge



Die Ausarbeitung der Verträge kann erfahrungsgemäß recht lange dauern. Häufig wird nach der Wahl eines Vertragstypus noch längere Zeit um einzelne vertragliche Details verhandelt.

Leider dauert diese Phase der Vertragserstellung in den meisten Fällen derart lange, dass mit dem Projekt bereits begonnen werden muss und sich während einer bestimmten Zeit die Vertragsparteien in einem vertragslosen Zustand bewegen. Dies kann selbstverständlich wieder für die eine oder andere Vertragspartei von Vorteil sein. Es ist zu empfehlen, keinesfalls vor Vertragsunterzeichnung mit den Arbeiten zu beginnen.

Nachfolgend die typische Struktur eines Projektvertrags:

### **I VERTRAGSGEGENSTAND**

- **Präambel**
- **Ausgangslage**
- **Ziele**

### **II ART UND UMFANG DER LEISTUNGEN**

- **Gegenstand des Vertrags**
- **Projektänderungen**

### **III PROJEKTORGANISATION**

- **Projektorganisation**
- **Verpflichtungen des Lieferanten**
- **Verpflichtung und Mitwirkung des Kunden**
- **Erfüllungsort und Unterauftragnehmer**

### **IV VORGEHEN BEI DER VERTRAGSABWICKLUNG**

### **V ABNAHME VON VERTRAGSLEISTUNGEN**

### **VI RECHTE AM ARBEITSRESULTAT**

- **Eigentums- und Schutzrechte**
- **Geheimhaltung**

### **VII VERGÜTUNG UND ZAHLUNGSBEDINGUNGEN**

### **VIII GARANTIE UND GEWÄHRLEISTUNG**

### **IX HAFTUNG UND VERZUG**

### **X VERTRAGSDAUER UND BEENDIGUNG**

### **XI SCHLUSSBESTIMMUNGEN**

- **Abwerbung von Mitarbeitern**
- **Anwendbares Recht und Gerichtsstand**

### 13.7 Vertragsentscheidung



Nach der Ausarbeitung der Verträge und deren anschließender Dokumentation als Vertragswerk muss eine formelle Entscheidung durch den Auftraggeber bzw. Projektausschuss gefällt werden. Es wird entweder entschieden, die Verträge in der vorliegenden Form zu unterzeichnen oder es müssen nochmals geänderte Varianten bzw. Anpassungen in Betracht gezogen werden.

Da sich auch dieser Prozess häufig nochmals in die Länge zieht, da neben dem Management bei jeder Änderung allenfalls auch die Rechtsdienste sich nochmals damit befassen, wird gelegentlich auf einen so genannten “Letter of Intent” zurückgegriffen. Dieser beinhaltet die grundsätzliche Willens- bzw. Absichtsäußerung, dass das Projekt mit dem Lieferanten durchgeführt werden soll.

Dies ermöglicht insbesondere dem Projektleiter auf Seite des Anbieters, seine eigenen Aufwände gegenüber dem Management zu rechtfertigen. Der Vorschlag für einen solchen “Letter of Intent” wird daher auch von Anbieterseite an den Kunden herangetragen werden.

Bereich Schritt	Sozio-technisches System	Einführung & Ausbildung	Betrieb & Unterhalt	Projektmanagement & Controlling
<b>Verifikation &amp; Analyse der Vertragssituation</b>	Analyse der offenen Punkte; Definierte Verifikationsthemen; Etwaiger Verifikationsprototyp; Berurteilung des Prototyps; Vertragsanalyse Software-Lizenzvertrag; Risikoanalyse; Stärken & Schwächen.	Analyse Dienstleistungs- vertrag.	Analyse Wartungs-, Support- & Update-Vertrag.	Projektplanung Vertrags/Verifikation; Projektcontrolling.
<b>Vertragsziele</b>	Ziele Lizenzvertrag.	Ziele Dienstleistungs- vertrag.	Ziele Wartungs-, Support- & Update- vertrag.	
<b>Vertragskonzepte</b>	Konzept Software-Lizenzvertrag.	Konzept Dienstleistungs- vertrag.	Konzept Wartungs-, Support- & Updatevertrag.	Projektplanung Realisierung als Vertragsbestandteil.
<b>Bewertung der Vertragskonzepte</b>	Bewertung der SW-Lizenzvertragskonzepte.	Bewertung Dienstleistungs- vertragskonzepte	Bewertung Wartungs- vertragskonzepte.	
<b>Verträge</b>	Vertrag Software-Lizenzvertrag; eventuell SW-Kaufvertrag.	Dienstleistungs- vertrag.	Wartungs-, Support- & Updatevertrag.	
<b>Vertragsentscheidung</b>				Vertragsentscheid.; Freigabe Realisierung.

Abb. 185: Ergebnisübersicht der Verifikations- und Vertragstätigkeit

### **13.8 Literatur zu Verifikation und Vertrag**

- SVD (1985): Evaluation von Informatiklösungen: Verfahren – Methoden – Beispiele. Schweizerische Vereinigung für Datenverarbeitung (Hrsg.). Bern/Stuttgart: Haupt.
- Bernath, F.A. (1995): Rechtsverhältnisse im Informatik-Bereich. In: EDV-Wissen für Anwender: ein Informatik-Handbuch für die Praxis (Becker, M., Habermellner, R., Liebetrau, G.) (11. Aufl.). Zürich: Industrielle Organisation.
- Montani, S.; Kummer, F. (2002): Informatikrecht in der Praxis: Recht und Praxis rund um den Einsatz von Informatik- und Kommunikationsmitteln. Zürich: WEKA Verlag AG.
- Neff, E.F. (2003): Der Projektvertrag als Schlüssel für ein erfolgreiches Informatik-Projekt. Zürich: Vorlesungsunterlage ETH Zürich.
- Straub, W. (2004): Informatikrecht: Einführung in Softwareschutz, Projektverträge und Haftung. Zürich/Bern: vdf Hochschulverlag/Stämpfli.

VOM GROBEN ...



... INS DETAIL !





# 14

## Detailkonzept

### Kapitelinhalt

- Detailanalyse Prozesse, Funktionen, Objekte und Aufgaben
- Detailkonzeption der Prozesse mit Prozessstatus & Funktionen
- Detailkonzeption der Geschäftsobjekte und Layout von Reports
- Detailkonzeption der Systemschnittstellen

Das Detailkonzept hat zwei Zielsetzungen.

A) Es hat zunächst die gegenüber dem Konzept abermals detailliertere und verfeinerte Beschreibung des Informationssystems zum Ziel. Das Detailkonzept soll die Systemeigenschaften so exakt festhalten, dass der anschließenden Implementation keine wesentlichen Spielräume hinsichtlich der Funktionalität mehr erlaubt sind. Lediglich die technische Ausgestaltung – das Design – sollte nach dem Detailkonzept noch vorgenommen werden – funktional ist das System mit dem Detailkonzept festgelegt.

B) Zweitens erfolgt die Detailspezifikation bezogen auf eine benannte Softwareplattform. Während also im Konzept (und im Pflichtenheft) die Spezifikationen meist bewusst vollständig unabhängig von einer spezifischen Softwarelösung dargelegt wurden, wird man sich nun überlegen müssen, wie sich die gewünschten Funktionalitäten auf Basis der gewählten Softwareplattform optimal integrieren lassen.

## 14.1 Inhalt und Vorgehen

### 14.1.1 Inhalt

Primäres Ziel der Detailkonzeption ist die abermals vertiefte Analyse und Konzeption aller Systemelemente und Beziehungen des Informationssystems. Dies umfasst zunächst eine Analyse der Geschäftsprozesse unter Definition der zu implementierenden Prozessstatus. Weiter hat eine detaillierte Beschreibung der Objektattribute zu erfolgen (z.B. Attribute des Kundenstamms). Es erfolgt eine Beschreibung der wichtigsten Reports (Rechnungen, Statistiken, etc.) und die Beschreibung der Schnittstellenanforderungen (z.B. CAD, BDE, Finanzbuchhaltung). Sodann müssen die Prioritäten der Implementation überprüft und festgelegt werden, denn häufig haben sich im Unternehmen während dem Projekt neue und zusätzliche Erkenntnisse ergeben.

### 14.1.2 Vorgehen

Die Detailkonzeption beginnt mit einer Analyse, welche erneut in einem höheren Detaillierungsgrad als im Konzept ausfällt (s. Abb. 186). Die nachfolgenden Zielsetzungen und Detailkonzepte werden in der Regel für jeden zu betrachtenden Geschäftsprozess oder Teilbereich isoliert erarbeitet. Vielfach wird für jeden Teilbereich ein eigenes Teilprojekt und ein Teilprojektteam gebildet. Der Detailkonzeptbericht entspricht der Zusammenfassung und Integration der verschiedenen Detailkonzepte.

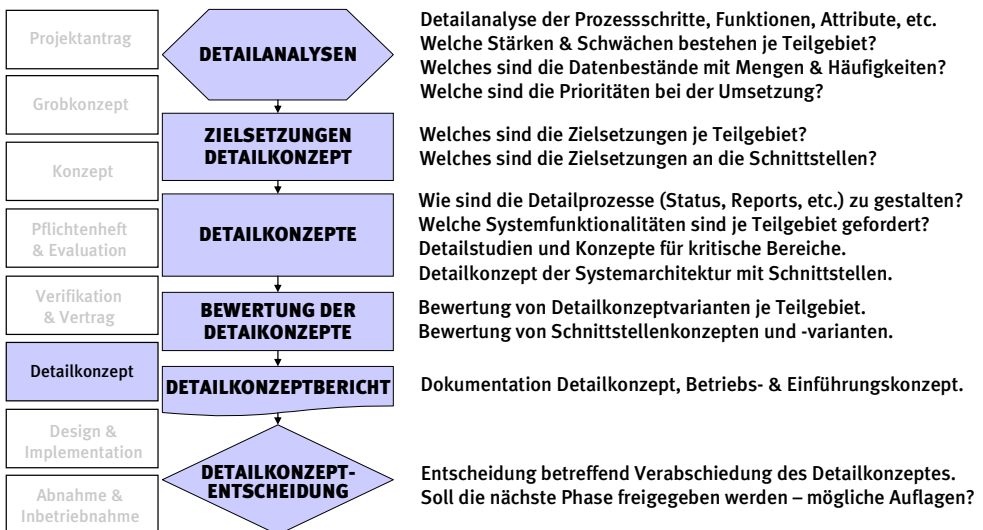


Abb. 186: Vorgehensschritte bei der Detailkonzepttätigkeit

### 14.1.3 Ebene der Systembetrachtung im Detailkonzept

Im Rahmen des Detailkonzeptes hat entsprechend der unten stehenden Darstellung eine nochmalige Detaillierung des Konzeptes auf einer neuerlich tieferen Systemebene zu erfolgen (s. Abb. 187). Diese Detailanalysen und -konzepte haben sich wiederum auf alle Sichten des Informationssystems zu beziehen. Als Ausgangspunkt dienen die Dokumente des Konzeptes (s. Abb. 112).

In den meisten Fällen kann allerdings nicht sofort mit der Detailkonzeption begonnen werden. Wenn im Rahmen der Evaluation eine Standardsoftware evaluiert wurde, so können die vordefinierten Geschäftsprozesse des Systems einen nicht unerheblichen Einfluss auf die Gestaltung der Prozesse auf Konzeptebene haben. Man wird daher gewissermaßen nochmals im Konzept einsteigen müssen und erst nach den hier notwendigen Anpassungen zur eigentlichen Detaillierung übergehen.

Das Detailkonzept beinhaltet zunächst die genaue Beschreibung jedes einzelnen Teilprozesses und der damit verbundenen Informationen zur späteren Konfiguration des Systems. Es wird also gewissermaßen eine Lupe genommen und mit dieser jeder einzelne Teilprozess nochmals auf seine Detailschritte hin analysiert.

Diese Detailschritte liefern ihrerseits wiederum Hinweise auf die Detailfunktionen. Hinsichtlich der Objekte erfolgt primär eine Analyse mit Blick auf ihre zu implementierenden Attribute.

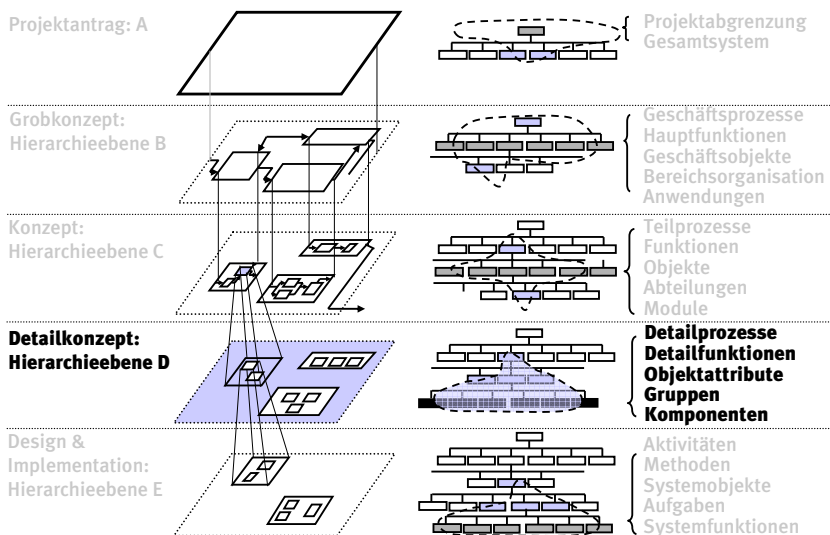


Abb. 187: Betrachtungstiefe des Systems im Detailkonzept

#### 14.1.4 Verwendete Modellierungsmethoden

Das Detailkonzept stellt wie erwähnt einen Übergang von produktunabhängigen Modellen zu den produktbezogenen Modellen dar. Es ist in diesem Stadium klar, welches System die Zielplattform des Informationssystems darstellt und über welche Module dieses verfügt. Zumindest ist dies dann der Fall, wenn eine Standardsoftware evaluiert wurde. Dadurch ergeben sich gegenüber dem Schritt der Anforderungsspezifikationen neue Rahmenbedingungen.

In vielen Fällen muss die angestrebte aufbau- und insbesondere ablauforganisatorische Lösung an die Möglichkeiten des technischen Informationssystems angepasst werden. Die folgende Matrix drückt dies dadurch aus, dass die bereits erarbeiteten Diagramme, wie das Prozessmodell und das Funktionsmodell, einer größeren Überarbeitung zu unterziehen sind (s. Abb. 188).

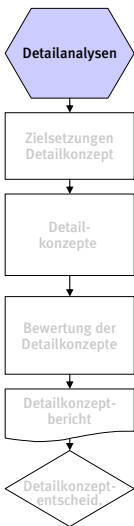
Häufig wird kritisiert, "dass das Unternehmen sich dem System anpassen musste, statt umgekehrt". Ganz bestimmt sind bei allen Systemeinführungen vielfältige Anpassungen zu erwarten und zu bewältigen. Heikel wird es dann, wenn es um geschäftskritische Abläufe geht und dadurch die Möglichkeiten eines Unternehmens, innovative oder kostensparende Ansätze zu verwirklichen, verhindert werden.

<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht	Techniksicht
Prozesssicht	Prozessmodell Detailprozesse				System- schnittstellen- diagramm
Funktionssicht	Flussdiagramm	Funktions- modell/Detail- funktionen			System- funktionen- diagramm
Objektsicht			Objektmodell Detailobjekte		Datenobjekt- diagramm & Attribute
Aufgabensicht	Stellenorien- tiertes Ablauf- diagramm	Use-Case- Diagramm		Organisations- modell	
Techniksicht					System- architektur / Komponenten

Abb. 188: Beispielhafte Modellierungstechniken des Detailkonzeptes

## 14.2 Detailanalysen

### 14.2.1 Detailanalyse der Prozesse



Wie eingangs erwähnt sollte zunächst eine Überprüfung der Geschäftsprozesslandkarte und der Wertschöpfungsketten vorgenommen werden, diesmal auf der Grundlage der Systemmöglichkeiten des allenfalls evaluierten Systems.

In einem nächsten Schritt eignet sich das stellenorientierte Ablaufdiagramm in idealer Weise, um Prozesse detaillierter zu analysieren und transparent zu machen. Der Vorteil der Darstellung des stellenorientierten Ablaufdiagrammes liegt primär darin, dass es die Aufbau- und die Ablauforganisation in ihrer optimalen Verknüpfung klar aufzeigt (s. Abb. 189).

Entsprechend der Ebene des Detailkonzeptes wird man damit faktisch für jeden einzelnen Teilprozess ein eigenes Ablaufdiagramm erstellen müssen. Hier stellt sich selbstverständlich die Frage nach dem Aufwand zur Erfassung der Ist-Prozesse. Unter Umständen wird man sich gleich auf die Gestaltung der Soll-Prozesse konzentrieren.

Wie in den vorangegangenen Kapiteln aufgezeigt, lässt sich dieses Diagramm optimal durch zusätzliche Dimensionen "anreichern", z.B. in einer Spalte rechts außen. Oder aber es werden Zwischenresultate oder Ergebnisdokumente direkt in den Prozessablauf eingezeichnet.

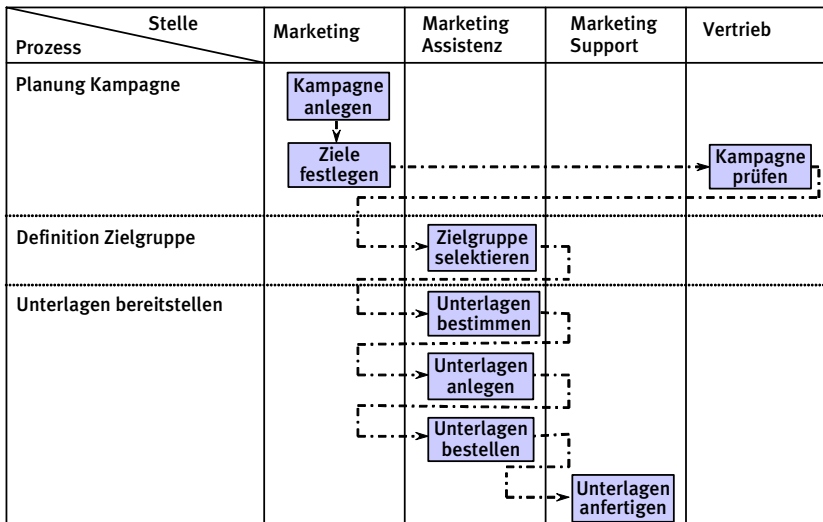


Abb. 189: Stellenorientiertes Ablaufdiagramm

### 14.2.2 Detailanalyse der Funktionen

Ausgehend von den zu diesem Zeitpunkt verfügbaren Analysen wird in einem nächsten Schritt des Detailkonzeptes auch die funktionale Sichtweise nochmals erweitert werden müssen. Besonders wichtig ist der verbindliche Abgleich der Funktionsliste des Pflichtenheftes mit den Möglichkeiten des Lieferanten. Über die abzudeckenden Funktionen und Detailfunktionen sollte genauestens Buch geführt werden, denn es wurde ja im Rahmen der Offerte und des Vertrags eine gewisse Funktionalität zugesichert. Diese Zusicherung sollte nicht über Bord geworfen werden, sondern im Rahmen der Detailanalyse als Basis für die Funktionslisten erhalten.

Zur Erfassung der detaillierten Funktionen eignen sich selbstverständlich alle Methoden der Funktionssicht. In der Praxis wird häufig die Ableitung aus den oben genannten Flussdiagrammen oder Stellenfunktionendiagrammen dienen (s. Abb. 190). Allerdings wird es hier unumgänglich sein, die Vielzahl der Funktionen in textueller Form, beispielsweise in einer Tabellenkalkulation aufzulisten, zu priorisieren etc.

Bei Individualsoftwareprojekten wird sich der Einsatz von Use-Cases empfehlen. Ein Vorteil der Use-Cases liegt darin, dass auch diese intuitiv sehr verständlich sind. Die Anwender finden sich direkt darin wieder. Gemäß unserer Definition verknüpfen Use-Cases ja gerade die Aufgaben- mit der Funktionssicht. Die Use-Cases können auch direkt von einem Stellenorientierten Ablaufdiagramm abgeleitet werden (s. Abb. 190).

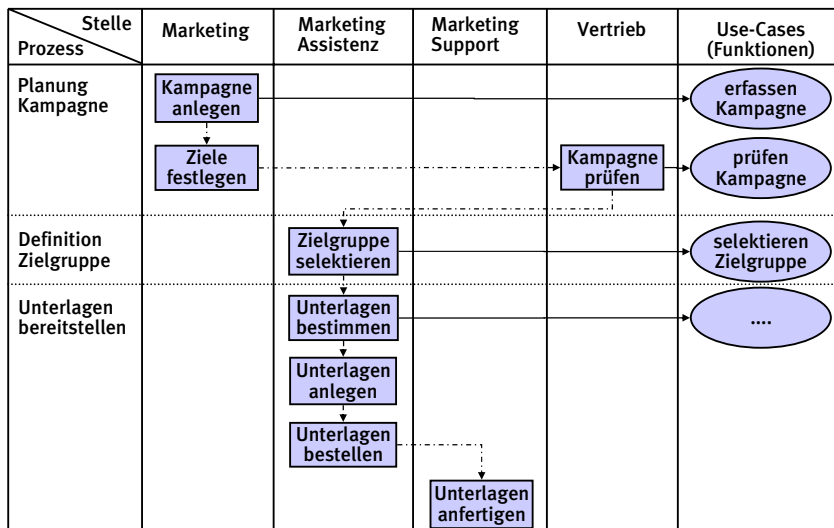


Abb. 190: Ableitung der Use-Cases (Funktionen)

14.3 Zielsetzungen Detailkonzept



Die Zielsetzungen im Detailkonzept entsprechen einer neuerlichen Verfeinerung der Zielsetzungen aus dem Konzept bzw. Pflichtenheft. Man wird diese Zielsetzungen idealerweise zu übernehmen versuchen und anschließend fortlaufend ergänzen und verfeinern (s. Abb. 191).

Häufig findet im Stadium des Detailkonzeptes der erste wirklich vertiefte Kontakt zwischen dem Systemanbieter und dem Unternehmen statt. Jetzt stellt sich heraus, ob die Versprechen der Verkäufer im Angebots- und Vertragsstadium von den für die Einführung verantwortlichen Technikern auch eingehalten werden können.

Wenn in den vorangegangenen Schritten unsauber gearbeitet wurde, dann können Unternehmen nun erstmals vor gewisse unangenehme Tatsachen gestellt werden. Durch die Übernahme der Zielsetzungen wird aber nochmals klar dokumentiert, dass das Projekt auch tatsächlich so umgesetzt werden will, wie es ausgeschrieben wurde.

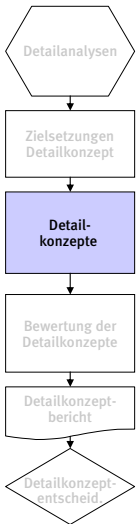
Es ist in diesem Zusammenhang besonders wichtig, dass Anwender möglichst rasch die konkreten Möglichkeiten und die Arbeitsweise des Systems kennen lernen. Vielfach sind nicht nur die Systemanbieter mit der Komplexität und Vielfalt des Unternehmens konfrontiert. Auch die Kundenseite ist gleichermaßen von der Mächtigkeit eines neuen Informationssystems betroffen.

Zielklassen	Ziele	Zieleigenschaften	Ausmass	Zielart/ Priorität	Zeitpunkt
Detailziele Kampagnen- prozess	Effektive Kampagnen	Aufwand für Einzelkampagne	<= 0.5 Tage	Mussziel Prio. 1	nach Einführung
	Wiederkehrende Kampagnen	binär	ja / nein	Mussziel Prio. 3	nach Einführung
	Kopierfunktion Kampagnen	binär	ja / nein	Mussziel Prio. 4	nach Einführung
	Auswertungen zu Kampagnen	Zeitaufwand Auswertung	<= 5 Sek.	Mussziel Prio. 2	nach Einführung
	Eingeschränkte Rechtevergabe	Anzahl Ebenen Rechtevergabe	2 Ebenen	Wunsch Prio. 5	nach Einführung
	Monatsberichte Kampagnen	Anzahl versch. Berichte	3 versch. Berichte	Mussziel Prio. 3	nach Einführung
	Marketing Verkaufstendenz	Qualität der Darstellung	Text / Grafik	Mussziel Prio. 1	nach Einführung
	Einsatz Portal-Navigator	Grad der Integration	teilw. / voll.	Wunsch Prio. 5	nach Einführung

Abb. 191: Beispielhafter Ausschnitt aus dem Zielkatalog je Detailkonzept

## 14.4 Detailkonzepte

### 14.4.1 Detailkonzept der zu implementierenden Geschäftsprozesse



Während die vorangegangenen Analysen allenfalls nochmals unabhängig von der konkreten Softwarelösung erstellt werden können, muss die nun die folgende Detailkonzeption der Prozesse klar bezogen auf die gewählte Software(plattform) erfolgen. Dies trifft nur dann nicht zu, wenn es sich um eine Individualsoftwarelösung handelt.

Zunächst werden die Prozesse in eine verbindliche und systembezogene Form gebracht werden müssen. So gilt es beispielsweise definitiv festzuhalten, welche Prozesse zu implementieren und zu segmentieren sind (s. Abb. 192). In unserem Beispiel sollen fünf Kampagnenprozesse angelegt werden, welche über jeweilige eigene Prozessschritte verfügen. Weiter sollte auch detailliert festgehalten werden, welche Aktionen zu erfolgen haben, wenn ein bestimmter Prozessstatus erreicht ist. So kann die Erreichung des Status "Kampagne angelegt" erfordern, dass eine Mail an die Vertriebspersonen versandt werden sollte. Die bereits vorhandenen Spezifikationen werden also nochmals weiter präzisiert und sukzessive vertieft.

Eine scharfe Definition des adäquaten Detaillierungsgrads einer Detailkonzeption lässt sich nicht klar angeben. Gelegentlich wird man nämlich auch direkt dazu übergehen, Ansätze des Detailkonzeptes im Rahmen einer prototypischen Implementation vorzunehmen.

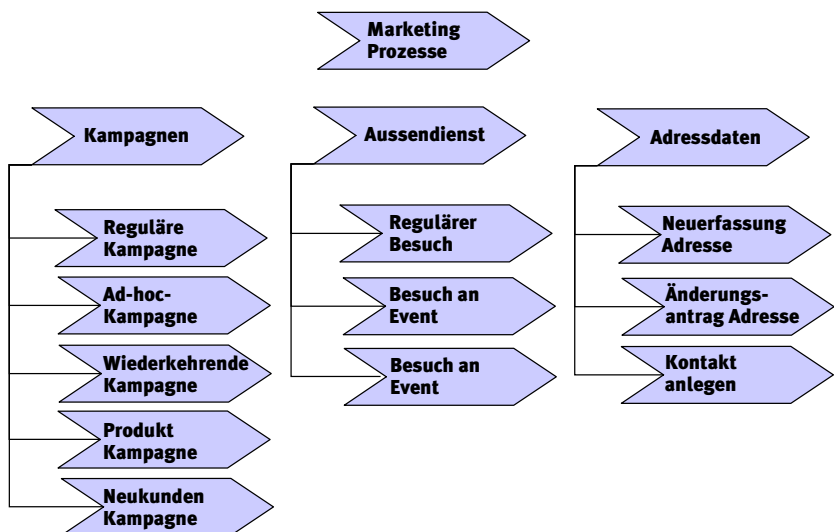


Abb. 192: Detailliertes Prozessmodell und Segmentierung einzelner Prozesse



### Detailkonzept der Teilprozesse

Die Detailkonzeption von Prozessen wird in vielen Fällen nicht vollständig frei gestaltbar sein. Bei Projekten auf Basis von Standardsoftware bestehen seitens der Lösung häufig die erwähnten Restriktionen. Nur wenige Standardsoftwaresysteme lassen eine vollständig freie Konfiguration der Prozesse zu, meist wird auf Basis einer Vorlage (engl.: "Template") vorgegangen, welche angepasst werden muss. Damit beginnen aber oftmals auch die Detailkonzeptvarianten und die Entscheidung, ob der bereits implementierte Prozess tatsächlich als Basis ausreicht und die Bedürfnisse der Benutzer abzudecken in der Lage ist, oder ob Änderungen an der Software vorzusehen sind.

Unabhängig von dieser Diskussion ist im Detailkonzept der Ansatz verbreitet, Prozesse durch Flussdiagramme zu konkretisieren. Durch die Konnektoren kann im Flussdiagramm visualisiert werden, welche Beziehungen zu anderen Geschäftsprozessen bestehen. Immer wieder entstehen Probleme, da im Detailkonzept unterschiedliche Teams mit Teilprojekten betraut werden und die integrale Sicht verloren geht.

Das Flussdiagramm eignet sich nun auch ausgezeichnet, um den Übergang zur detaillierten Funktionssicht einzuleiten (s. Abb. 193). Die erfolgreiche Abwicklung von Detailprozessen mit Hilfe der IT wird sich ja gerade daran messen, ob die notwendigen Detailfunktionalitäten vorhanden sind. Was liegt also näher, die benötigten Systemfunktionen wiederum direkt aus diesen Prozessen abzuleiten?

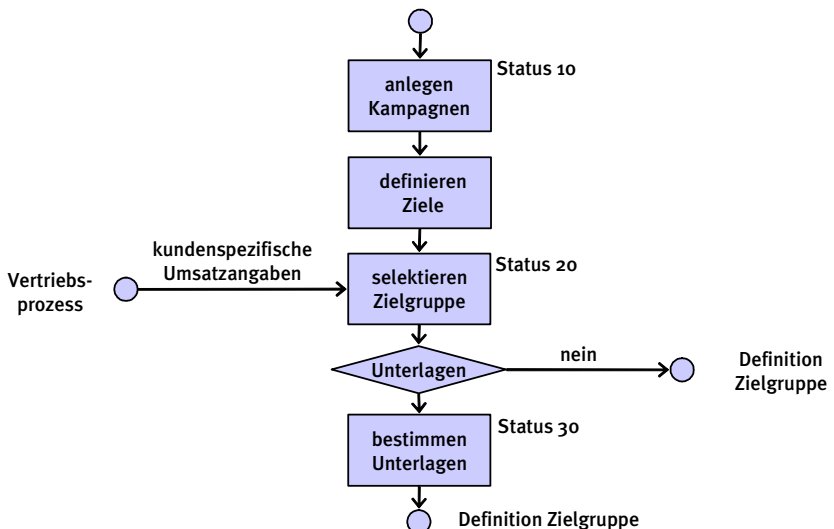


Abb. 193: Flussdiagramm eines Teilprozesses

### 14.4.2 Detailkonzept der Funktionen

In aller Regel wird im Rahmen der Detailkonzeption auch die im Konzept definierte Funktionalität nochmals überarbeitet und bereinigt (s. Abb. 194). Dies erfolgt häufig unter Mitarbeit eines größeren Personenkreises als im Spezifikationsprozess.

Es ist daher durchaus möglich, dass nochmals neue Funktionalitäten zum Vorschein kommen oder aber auch nicht tatsächlich benötigte Funktionen eliminiert werden können. Dies trifft insbesondere dann zu, wenn sich die Prioritäten bei der Realisierung geändert haben und allenfalls bestimmte Funktionsbereiche entfallen.

Außerdem neigt man vor einer Evaluation häufig dazu, die geforderte Funktionalität künstlich hochzuschrauben, um eher auf der sicheren Seite zu stehen, was den Funktionsumfang des Systems anbelangt. Man wird aber im Rahmen des Detailkonzeptes die tatsächlich abnahmerelevanten Punkte explizit angeben müssen. Hierfür greift man am besten wie bei der Zielsetzung auf die im Rahmen der Ausschreibung erstellte Funktionsliste zurück und präzisiert bzw. ergänzt diese fortlaufend.

Auch dieser Prozess funktionaler Anpassungen hat unter dem strukturierten Änderungsmanagement zu erfolgen, wie es im Kapitel Projektmanagement behandelt wird. Es sind mit dem Anbieter stets klare Voraussetzungen zu schaffen, welche Funktionen wegfallen und welche neu hinzukommen.

Firma XY	Funktionalität				
	Priorität 1 (Muss)				
	Priorität 2				
	Geringe Priorität				
	Zukunft				
	1	2	3	4	Bemerkungen
Einzelfunktionen					
Kundenprofil					
• Freie Definition von Kundenprofilen	x				
• Kundenprofil am Bildschirm und als Ausdruck verfügbar		x			
• Umsatzanzeige nach Produktgruppen	x				
• Wichtigste Aufträge je Kunde			x		
• Wichtigste Angebote je Kunde	x				
• Sales Pipeline	x				
• Sicht auf die Produkte wechseln		x			
Unterlagen					
• Zeige alle Unterlagen, welche der Kunde bereits erhalten hat	x				
• Marketing Enzyklopädie (alle Unterlagen des Marketings)	x				
Kundenbesuch					
• On- und offline-Zugriff auf die Daten des Kunden		x			

Abb. 194: Detailkonzeption der Funktionen

### 14.4.3 Detailkonzeption der Objekte

Im Detailkonzept müssen durch die Projektteams weiter die notwendigen Attribute aller Geschäftsobjekte festgelegt werden (s. Abb. 195). Es muss beispielsweise detailliert festgehalten werden, welche Datenfelder und Attribute der Adressstamm, die Aufträge bzw. die Produkte enthalten sollen. In dieser Entwicklungsphase wird man sich nicht mehr auf eine mehr oder weniger unverbindliche Auflistung abstützen können. Projektleiter müssen daher bestrebt sein, diese Angaben möglichst verbindlich und termingerecht zu erhalten.

Im Unternehmen sind meist eine Vielzahl von Druckerzeugnissen vorhanden (Auftragsbestätigungen, Lieferscheine, Fortschrittslisten, Inventurlisten, Umsatzstatistiken, etc.). Die Benutzer werden letztlich mit dem Projekt nur zufrieden gestellt sein, wenn die Druckvorlagen die korrekte Form aufweisen, und das Management wünscht die Statistiken in der gewohnten Form. Im Informatikjargon spricht man an dieser Stelle meistens von "Reports", weil diese Darstellungen meist nicht "reine" Objekte darstellen, sondern aus einer Kombination von mehreren Objekten bestehen. Hinweise auf Objektattribute ergeben sich daher auch aus diesen Reports.

Wie bereits angeführt, beansprucht die Erstellung von Reports einen hohen Anteil des gesamten Implementationsaufwandes. Es ist daher absolut empfehlenswert, möglichst früh einen guten Überblick aller notwendigen Reports zu erarbeiten und die einzelnen Reports genau zu definieren.

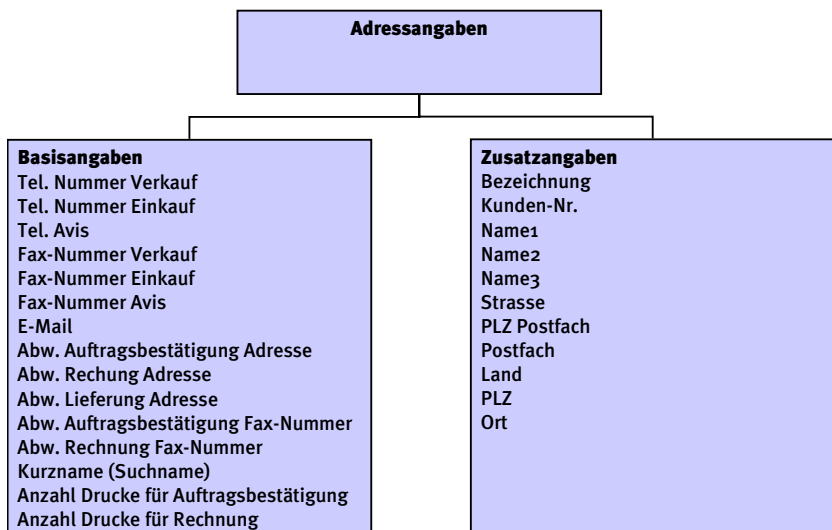


Abb. 195: Attribute eines Objektes

#### 14.4.4 Detailkonzeption der Organisation und Aufgaben

Es lohnt sich auch im Detailkonzept, die bestehende Aufgabenteilung einer etwaigen Reorganisation zu unterziehen. Falls dies nicht schon im Konzept definiert wurde, müssen spätestens in diesem Stadium Überlegungen angestellt werden, welche organisatorischen Veränderungen vorzusehen sind. Der heutige Trend in industrialisierten Ländern nach kundenspezifischen Lösungen kommt der Forderung der Arbeitspsychologie nach qualifizierter Arbeit entgegen. Damit stehen die Forderungen der Betriebswirtschaft nach qualifizierten Arbeitskräften nicht mehr in Kontrast zu den Postulaten nach humaner Arbeit. Dies ist als eine erfreuliche Tatsache zu werten.

Studien zeigen, dass mit steigendem Anpassungsaufwand die Zufriedenheit mit der Lösung sinkt (Rieder 1988). Andere Studien kommen Gegenteilig zum Schluss, dass Anpassungen in einem sinnvollen Umfang die Akzeptanz durchaus erhöhen. Gemäß dieser zweiten Studie ist ein Verhältnis der Anpassungs- zu den Gesamtkosten von 15 bis 40 % optimal (Agiplan 1993).

Die Gestaltung der Organisation im Rahmen der Detailkonzeption betrifft meist die Gestaltung von Aufgaben auf der Ebene der Mitarbeiter. Ein damit verbundener Punkt betrifft die Erkenntnis, dass es den "one best way" der Aufgabengestaltung nicht gibt. Ulich formuliert dies wie folgt: "Einer der grundlegenden Irrtümer traditioneller Arbeitsgestaltung besteht offenbar gerade in der Annahme, dass es für jede Arbeitstätigkeit einen "best way" gebe, den es nur herauszufinden und den beschäftigten Personen zu vermitteln gelte" (Ulich 1978, S. 567).

Es wird also insbesondere für unterschiedliche Qualifikationsstufen der Mitarbeiter auch entsprechende unterschiedliche Arbeitsweisen geben können. Den Mitarbeitern sollte daher ermöglicht werden, sich im Zuge der laufenden Höherqualifizierung auch neue geänderte und zusätzliche Arbeitsweisen anzueignen. Es können drei Stufen unterschieden werden:

- flexible Arbeitssysteme
- differentielle Arbeitssysteme
- dynamische Arbeitssysteme.

Es wird postuliert, dass Arbeitssysteme so flexibel zu gestalten sind, dass individuelle Arbeitsweisen realisiert werden können. Die differentielle Arbeitsgestaltung bezieht sich auf das Angebot verschiedener Systeme. Die dynamische Arbeitsgestaltung bedeutet die Möglichkeit für die Mitarbeiter, ihre IT-Systeme selbst anzupassen (a.a.O.). Dies verlangt nach einer dezentralen Kontrolle der Benutzer über "ihre Informatik".

#### 14.4.5 Detailkonzeption der System- und Modularchitektur

Im Detailkonzept wird man sich meist zusammen mit den Unternehmensbereichen und dem Lieferanten über die definitive Struktur der Systemarchitektur einigen müssen. Dies betrifft zunächst die Fragestellung, ob und welche Module vorzusehen sind und für welche Unternehmensbereiche eigenständige Modulimplementationen vorzunehmen sind (s. Abb. 196).

Aufbauend auf der bereits im Konzept definierten Systemarchitektur wird man deshalb im nächsten Schritt die relevanten Schnittstellen mit Hilfe des Systemschnittstellendiagramms (s. Teil 1) spezifizieren.

Wie bereits oben dargelegt, muss der exakten Konzeption der Schnittstellen in allen Informatikprojekten stets eine sehr hohe Aufmerksamkeit zukommen. Denn leider beginnen die wenigsten Informatikprojekte absolut auf der grünen Wiese. Daher muss zumindest über die wesentlichen Schnittstellen Klarheit bestehen. Ganz besonders trifft dies zu, wenn es darum geht, Schnittstellen zu bestehenden Systemen zu schaffen.

Allenfalls drängen sich im Detailkonzept zusätzliche Darstellungen auf, welche die Beziehungen zwischen den Subsystemen detailliert aufzeigen. Hierzu eignen sich sowohl die im Rahmen des Konzeptes dargestellten Objektzugriffdiagramme als auch ein verfeinertes Systemschnittstellendiagramm.

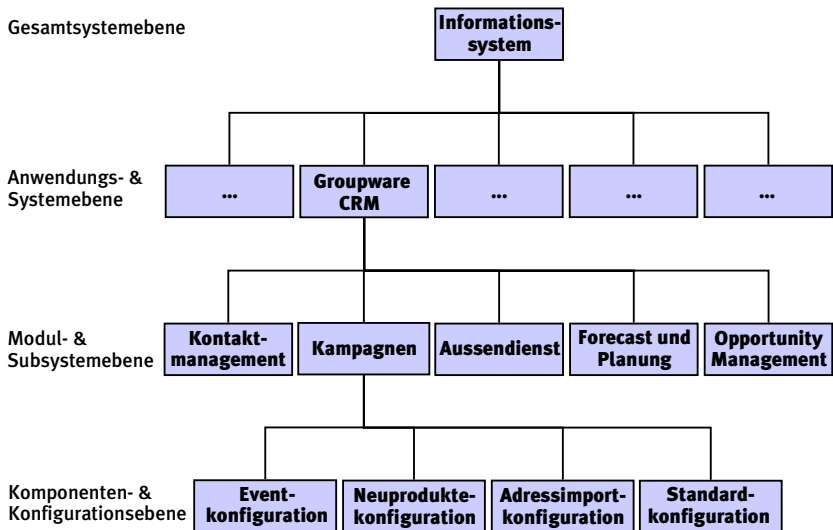


Abb. 196: Architektur im Detailkonzept

#### 14.4.6 Morphologische Matrix für das Detailkonzept

Die nachfolgend angeführte morphologische Matrix der Detailkonzeption fasst nochmals zusammen, welche Parameter und möglichen Ausprägungen in Betracht gezogen werden müssen (s. Abb. 197).

Eine erste Betrachtung betrifft die Gestaltung der Detailprozesse. Es wurde angemerkt, dass viele Entscheidungen in Bezug auf den Anpassungsgrad der Prozessschritte zu erfolgen haben.

Mitunter geht es eng verbunden um die zweite Frage, ob bei Standardsoftware die bestehende SW-Funktionalität genutzt wird oder in welchem Umfang notwendige Anpassungen vorzunehmen wären. Die schließt die Anpassung der Prozessschritte der Standardsoftware mit ein.

Weiter ist das Ausmaß der funktionellen Unterstützung zu definieren und die Mensch-Maschine-Funktionsteilung festzulegen. Sodann stellen sich häufig Fragen betreffend der Integration der Datenobjekte und deren Abgleich bei komplexeren Systemarchitekturen.

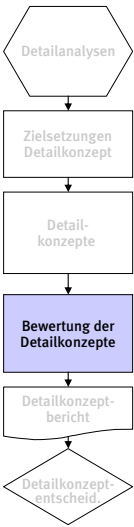
Von hoher Bedeutung ist der organisatorische Aspekt. Auf Stufe des Detailkonzeptes wird zu entscheiden sein, in welchem Ausmaß bestehende Tätigkeiten von Mitarbeitern anzupassen bzw. zu redefinieren sind.

Auch der Betrachtung der Systemschnittstellen kommt eine wichtige Rolle zu. Hier wären die möglichen Ausprägungen des konkreten Projektes einzutragen.

<b>Ausprägung</b> <b>Parameter</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>
<b>Detailkonzept Prozess</b>	Keine Anpassung des Prozesses	<b>Prozessschritte werden teilweise angepasst</b>	Prozess wird komplett neu gestaltet	...
<b>Detailkonzept Software</b>	<b>Bestehende Funktionalität wird genutzt</b>	SW-Funktionen werden teilweise angepasst	SW-Funktionen umfassend verändert	....
<b>Detailkonzept Funktionalität</b>	Keine funktionelle Unterstützung	<b>Begrenzte funktionelle Unterstützung</b>	Durchgängige Unterstützung aller Funktionen	...
<b>Detailkonzept Datenobjekte</b>	Lokal verfügbare Datenobjekte	<b>Daten werden periodisch abgeglichen</b>	Daten werden synchronisiert und abgeglichen	Datenbankzugriff auf gemeinsame Datenbank
<b>Detailkonzept Aufgaben</b>	Keine Anpassung der Aufgaben (Stellenprofile)	<b>Aufgaben (Stellenprofile) teilw. angepasst</b>	Aufgaben werden komplett neu gestaltet	...
<b>Detailkonzept Schnittstellen</b>	Keine Schnittstelle zu anderen Prozessen	<b>Teilweise Schnittstellen</b>	Diverse Schnittstellen	...

Abb. 197: Beispielhafte Morphologie im Detailkonzept

14.5 Bewertung der Detailkonzepte



Die Bewertung von Detailkonzepten erfolgt ähnlich wie beim Konzept meist nicht auf der Ebene von Gesamtsystemvarianten. Jeder Teilbereich oder jedes einzelne Detailkonzept eines Geschäftsprozesses oder Teilprozesses müsste grundsätzlich einer eigenen Bewertung unterzogen werden müssen. Dies führt zu einer hohen Anzahl an Einzelbewertungen. Selbstverständlich wird man daher auch hier formelle und methodisch saubere Bewertungsverfahren nur in den wichtigsten Fällen anwenden können und sich in anderen Fällen auf einfache Argumentenbilanzen beschränken.

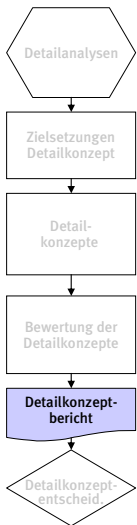
Im unten stehenden Beispiel werden z.B. die unterschiedlichen Integrationsvarianten formell bewertet, da es sich bei der Integration um eine wichtige Entscheidung handelt. Es geht um die Frage, in welcher Weise das CRM-System an das ERP-System angebunden werden soll (s. Abb. 198).

Im Detailkonzept kann es durchaus nochmals zu kritischen und richtungsweisenden Situationen kommen, welche klare Entscheidungen erfordern. Es drängen sich auch oft nochmals Entscheidungen auf, welche mit nicht unerheblichen Kostenfolgen verbunden sein können. Dies ist in diesem Projektstadium natürlich nicht ideal, da die Kosten im Konzept und im Rahmen der Evaluation eigentlich hätten abschliessend geklärt werden sollen. Leider ist dies aber realistisch und tritt immer wieder auf. Für solche Fälle sollten präventive Budgetmittel vorgesehen werden.

<b>Varianten Merkmale</b>	<b>Synchronisation Adressen</b>	<b>Gemeinsame Datenbank</b>	<b>Batchübernahme der Adressen</b>
Beschreibung der Variante	Mutationen dürfen im ERP und im CRM vorgenommen werden. Geänderte Daten werden beim Speichern beidseitig synchronisiert.	Das CRM-System ist voll in das bestehende ERP-System integriert. Es baut auf derselben Datenbank auf und muss vom selben Hersteller stammen.	Diese Lösung sieht eine einseitige Änderung der Adressen im ERP vor. Geänderte Angaben werden über Nacht synchronisiert.
Vorteile	Stets aktuelle Angaben; flexible Handhabung beider Systeme.	Vollständige Integration der Adressdaten. Optimale Aktualität und keine Redundanzen.	Geringer Aufwand zur Umsetzung, klarer und einfacher Prozess.
Nachteile	Eingriff auf beiden Seiten notwendig, auch im ERP ist eine Anpassung notwendig.	Hoher Realisierungsaufwand, hohes technisches Risiko.	Gefahr, dass Änderungen im CRM überschrieben werden. Daten sind nicht immer aktuell verfügbar.
Erwartete Kosten	120 000	250 000	40 000
Risikobeurteilung	Geringes Risiko	Hohes Risiko	Geringes Risiko

Abb. 198: Bewertung einzelner Detailkonzeptvarianten

## 14.6

**Detailkonzeptbericht**

Das Detailkonzept besteht in der Regel aus einer Summe von einzelnen Detailkonzepten der verschiedenen Teilprojekte & Teilgebiete, wie sie im Rahmen der Projektorganisation definiert wurden. Die Detailkonzepte einzelner Teams werden zusammengenommen und in einem integrierten Dokument dargestellt:

**I ZUSAMMENFASSUNG****II AUSGANGSLAGE****III DETAILKONZEPT STAMMDATEN****IV DETAILKONZEPT TEILGEBIET 1**

- **Ausgangslage & bereinigte Zielsetzungen**  
Konzeptionelle Grundlage, Stärken & Schwächen.
- **Prozesse**  
Flussdiagramme, Stellenorientierte Ablaufdiagramme.
- **Funktionen**  
Bereinigte und erweiterte Funktionslisten (aus Konzept), Definition, was in welcher Phase abgedeckt werden soll, detaillierte Use-Cases.
- **Wichtigste (Daten-)Objekte und ihre Attribute**
- **Gestaltung von Druckerzeugnissen (Reports)**
- **Mengen & Häufigkeiten**

**V DETAILKONZEPTE TEILGEBIET 2 – n****VI DETAILKONZEPT SYSTEMARCHITEKTUR**

- **Systemschnittstellen**  
Änderungen und Ergänzungen gegenüber Grobkonzept.
- **Systemarchitektur**  
Technikeinsatz, Systemschnittstellen, Datenobjektdiagramm.
- **Detailkonzept Betrieb und Unterhalt**

**VII DETAILKONZEPT EINFÜHRUNG UND AUSBILDUNG**

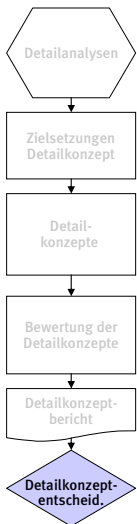
- **Einführungsplan, Ausbildungsplan, Datenübernahme (Migration)**

**VIII DETAILKONZEPT BETRIEB UND UNTERHALT****IX WEITERES VORGEHEN**

- **Terminierung und Etappierung des Projektes**
- **Einführungs-, Migrations- und Ausbildungskonzept**
- **Projektmanagement und Projektorganisation**  
Projektorganisation und -funktionen, Controlling, etc.



## 14.7 Detailkonzeptentscheidung



Die Detailkonzeptentscheidung entspricht einer Abnahme des Detailkonzeptes; dies stellt in jedem Projekt einen sehr wichtigen Meilenstein dar. Mit dieser Entscheidung wird der Umfang des künftigen Informationssystems in allen detaillierten Belangen festgelegt. Ein Projekt droht seine Vorgaben zu überschreiten, wenn fortlaufend neue Anforderungen auftauchen. Man sollte daher ab diesem Zeitpunkt die Anforderungen "einfrieren" und Anpassungen nur noch unter einem strukturierten Änderungsmanagement erlauben (s. Kapitel 18).

Aufgrund des Konzeptes bestand noch ein gewisser Freiraum. Es muss beispielsweise nicht feststehen, wie viele Arten von Kampagnenprozessen implementiert werden sollen. Es wird im Konzept lediglich verlangt, dass unterschiedliche Kampagnen angelegt werden können.

Nun gilt es aber beispielsweise definitiv zu verabschieden, welche Prozesse abschließend zu implementieren sind. Es steht z.B. weiter fest, welche Angaben im Kundenstamm enthalten und welche Status ein Prozess durchlaufen sollte. Das Detailkonzept hält außerdem fest, wo Anpassungsprogrammierungen vorzunehmen sind.

Bereich Schritt	Sozio-technisches System	Einführung & Ausbildung	Betrieb & Unterhalt	Projektmanagement & Controlling
<b>Detailanalysen</b>	Detailanalysen Prozessschritte; Detailanalysen Funktionalitäten; Detailanalyse Stammdaten; Detailanalysen Schnittstellen; Prioritäten festlegen; Analyse Technik & Peripherieanforderungen.	Analyse der Qualifikationen.	Detailanalyse von Betrieb & Unterhalt; Detailkonzeption Sicherheit.	Projektabgrenzung; Realisierungsplan; Projektplan; Verantwortlichkeiten; Projekthandbuch; Projektmanagement; Projektcontrolling.
<b>Zielsetzungen Detailkonzept</b>	Zielsetzungen Teilgebiete; Zielsetzungen Prototyping.	Einführungs- & Ausbildungsziele.	Ziele an den Betrieb & Unterhalt.	
<b>Detailkonzepte</b>	Detailkonzepte & -spezifikation; Systemprozesse & Status; Detailfunktionalitäten & funktionale Gaps identifizieren; Detailkonzept Stammdaten; Report-Layouts; Interfaces; Konzeption etwaiger Prototyp; Schnittstellenkonzepte; Systemarchitektur, -auslegung.	Datenübernahmeverfahren (Migration); Ausbildungskonzept; Datenübernahme; Geschulte Key User; Geschulte Techniker; Detailkonzept Einführung.	Detailkonzeption Betrieb & Unterhalt; Detailkonzept Sicherheit; Installation/Betrieb Projektinfrastruktur.	
<b>Bewertung der Detailkonzepte</b>	Bewertung von alternativen Detailkonzeptvarianten.			
<b>Detailkonzeptbericht</b>	Detailkonzeptbericht.			
<b>Detailkonzeptentscheidung</b>				Verabschiedung Detailkonzept.

Abb. 199: Ergebnisübersicht der Detailkonzepttätigkeit

## **14.8 Literatur zum Detailkonzept**

- Ehn, P., Kyng, M. (1991): Cardboard Computers: Mocking-it-up or Hands-on the Future. In: Design at Work: Cooperative Design of Computer Systems; (Greenbaum, J., Kyng, M.; Hrsg.). Hillsdale, NJ: Lawrence Erlbaum. 169–196.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999): The Unified Software Development Process. Reading, MA: Addison Wesley Longman.
- Rosson, M.B., Carroll, J. (1995): Narrowing the Specification-Implementation Gap in Scenario-Based Design. In: Scenario-Based Design – Envisioning Work and Technology in System Development. (Carroll, J.; ed.). New York: John Wiley & Sons. 247–278.





# 15

## Design und Implementation

### Kapitelinhalt

- Konsistente Systemanalyse & Systemdesign auf tiefster Ebene
- Implementation Objekte, Funktionen, Datenbank & Oberflächen
- Konfiguration und Test des Systems
- Systemdokumentation

Im Design- und Implementationsschritt wird das Informationssystem zunächst nochmals einer sehr strukturierten Analyse auf tiefster Systemebene unterzogen. Für diese Tätigkeit wird in der Informatik (engl.: “Computer Science”) der Begriff “Systemanalyse” (engl.: “Systems Analysis”) verwendet. Gemäß dem allgemeinen Systems Engineering stellen aber selbstverständlich bereits alle Analysen im Grobkonzept, Konzept und Detailkonzept ebenfalls “Systemanalysen” dar. Diese unterschiedliche Begrifflichkeit könnte zu Missverständnissen in IT-Projekten führen.

Im Anschluss an die “implementationsnahe” Analyse wird das Design erstellt. Die Design­tätigkeit widmet sich dem konsistenten und vollständigen Entwurf (engl.: “Design”) aller Software- und Datenbankkomponenten sowie der damit verbundenen Benutzungsoberflächen. Dies entspricht der “Konzeption” auf tiefster Systemebene. In Standardsoftwareprojekten wird diese Analyse- und Design­­tätigkeit nur punktuell erfolgen. Nach dem Design wird Software implementiert, konfiguriert und getestet.

## 15.1 Inhalt und Vorgehen

### 15.1.1 Inhalt

Die Systemanalyse entspricht einer implementationsnahen Analyse und das Design bezeichnet die konzeptionelle Gestaltung der Systemarchitektur sowie der Software-, Datenbank- und Benutzungsoberflächenkomponenten. Im Rahmen dieser Tätigkeit muss neben betrieblichen Anforderungen schwergewichtig auch den technischen Rahmenbedingungen Rechnung getragen werden. Es gilt, die Möglichkeiten der Programmiersprache mit der Performance und den betrieblichen Anforderungen abzustimmen.

### 15.1.2 Vorgehen

Der Design- & Implementationsschritt beginnt mit einer sehr strukturierten und konsistenten Systemanalyse auf tiefster Ebene. Während das Detailkonzept zum Ziel hatte, die Anforderungen aus betrieblicher Sicht zu beschreiben, so reicht dieser Detaillierungsgrad als Vorgabe für die nachfolgende Implementation noch nicht aus (s. Abb. 200).

Die Implementation und Konfiguration beinhaltet anschließend den eigentlichen "Systembau". Nach der Implementation erfolgen eine Reihe von Tests, welche so lange fortgeführt werden, bis das System die gewünschten Resultate erbringt. Weiterer Bestandteil der Design- und Implementationstätigkeit ist eine lückenlose Systemdokumentation.

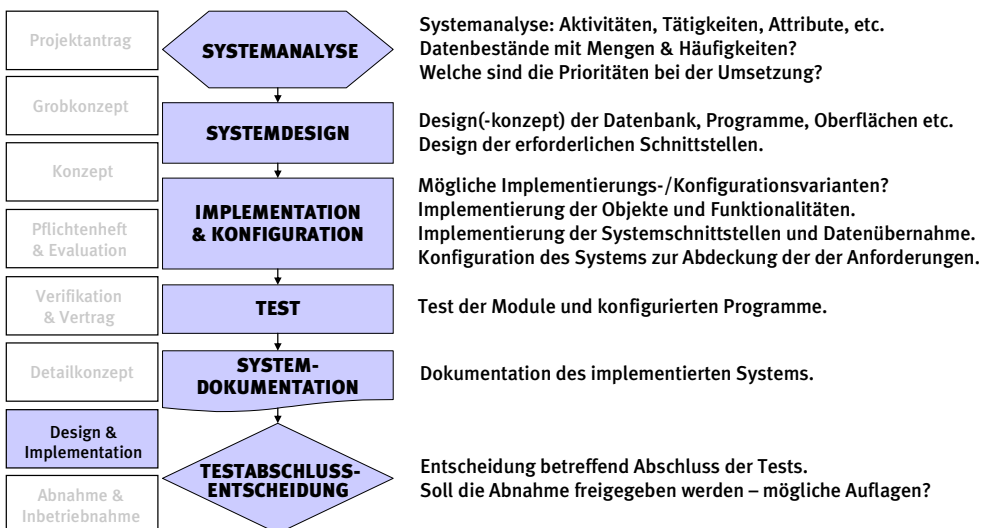


Abb. 200: Vorgehensschritte der Design- und Implementationstätigkeit

### 15.1.3 Ebene der Systembetrachtung in Design und Implementation

Nachdem im Rahmen der Detailkonzeption bereits die “Details” des Systems festgelegt wurden, könnte man zu Recht fragen, welche Betrachtungsebene noch detaillierter sein könnte als das “Detailkonzept”. Die Antwort besteht darin, dass das “Detailkonzept” das System primär aus Sicht der Anwender wiedergibt und für diese bereits eine sehr detaillierte Detailspezifikation darstellen mag. Was aber aus Anwendersicht detailliert sein kann, ist für die Programmierer noch zu grob formuliert.

Dies gilt analog bei anderen technischen Systemen. Die detaillierte Spezifikation einer Kaffeemaschine aus Anwendersicht beschreibt noch nicht, in welcher Form die Schläuche im Innern der Maschine verlaufen sollten. Diese technischen Spezifika interessieren den Anwender in der Regel nicht, oder nur dann, wenn sie Auswirkungen in Bezug auf die Benutzung haben.

Die Betrachtungsebene im Rahmen des Designs und der Implementation fokussiert nun also auf die systemtechnische Ebene des Systems. Als Ausgangspunkt dienen die Dokumente des Detailkonzeptes (s. Abb. 201). Ein gewisser Teil (z.B. Geschäftsobjekte, Attribute und Funktionen) mag für die Benutzer zwar erkennbar sein, andere Elemente sind es nicht und bleiben verborgen. Diese systemtechnische Ebene fokussiert z.B. auf die zu implementierenden Methoden, auf Systemobjekte, welche dem Informationssystem als Ganzem zur Funktionsfähigkeit verhelfen.

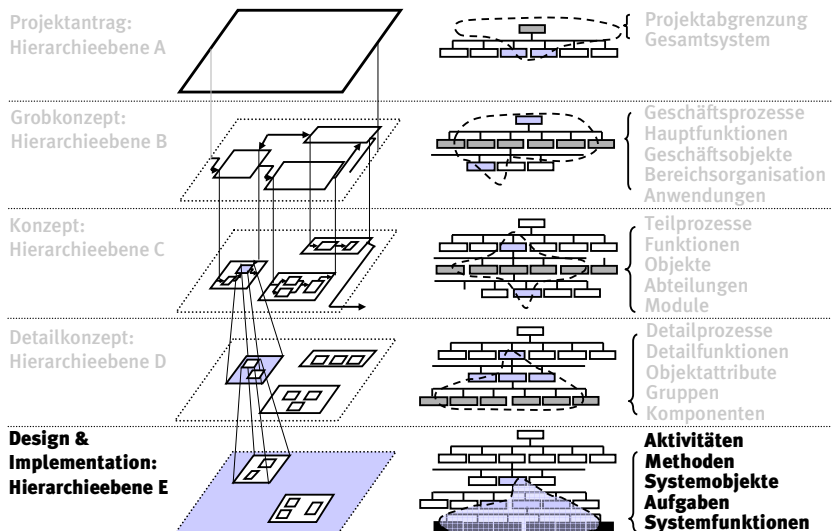


Abb. 201: Betrachtungstiefe des Systems in Design und Implementation

### 15.1.4 Verwendete Modellierungsmethoden

Die angewendeten Modellierungsmethoden im Rahmen der Design- und Implementationstätigkeit hängen zunächst davon ab, ob es sich um eine Individualsoftware- oder um Standardsoftwareentwicklung handelt. Bei Individualsoftwareprojekten werden die Analysen und Konzepte mit Hilfe einer Auswahl der unten dargestellten Diagramme erfolgen. Bei Standardsoftwareprojekten bevorzugt man häufig Prototypen und modelliert allenfalls nur noch punktuell ergänzende Funktionalitäten oder Schnittstellenkonzepte.

Aus der unten stehenden Matrix wird ersichtlich, dass die verwendeten Modelle primär im Bereich der Objekt- und der Funktionssicht anzusiedeln sind (s. Abb. 202). Während in den 80er-Jahren funktionsorientierte Modellierungstechniken im Vordergrund standen, werden heute fast ausschließlich objektorientierte Methoden verwendet. Damit wird das Objektmodell zum zentralen Modell dieses Gestaltungsschrittes. Zusätzlich kommen die technischen Systemarchitekturmodelle zum Einsatz, welche die Systemebenen und -schnittstellen visualisieren.

In den wenigsten Fällen werden alle unten schraffierten Modelle gemeinsam zum Zuge kommen. Trotzdem ist erkennbar, dass durch die zunehmende Stärke des Formalisierungsanspruchs mit der nahenden Implementation ebenfalls das Bedürfnis nach entsprechenden Modellen steigt. Viele Diagramme werden aber wie erwähnt nur sehr punktuell eingesetzt.

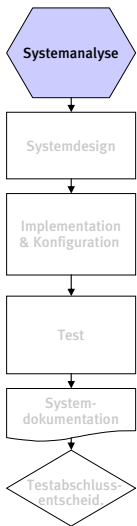
<div>primär</div> <div>sekundär</div>	Prozesssicht	Funktionssicht	Objektsicht	Aufgabensicht	Techniksicht
Prozesssicht	Prozessmodell	Funktionen-Block-diagramm	Zustands-übergangs-diagramm		System-schnittstellen-diagramm
Funktionssicht	Fluss-diagramm	Funktions-modell			
Objektsicht	Sequenz-diagramm		Objektmodell		Datenobjekt-diagramm
Aufgabensicht		Use-Case-Diagramm	Kollaborations-diagramm	Organisations-modell	
Techniksicht			Objekt-zugriffs-diagramm		System-architektur-modell

Abb. 202: Beispielhaftes Modellierungstechniken in Design & Implementation



## 15.2

## Systemanalyse



In der Sprache der Informatik (engl.: “Computer Science”) wird im Unterschied zur Wirtschaftsinformatik unter “Systemanalyse” ausschließlich die Form der implementationsnahen Analyse auf tiefster Systemebene verstanden. Es wird daher auch oft im gleichen Atemzug von “Analysis & Design” gesprochen. Vielleicht drückt diese gemeinsame Nennung von Analyse und Design auch aus, dass sich die implementationsnahe Analysetätigkeit nicht mehr in derselben Strenge vom konzeptionellen Designschritt abgrenzen lässt, wie dies in den vorangegangenen Gestaltungsschritten der Fall war. Die im Rahmen der Systemanalyse erstellten Modelle gehen meist fließend in die Designmodelle über. Daher wird in diesem Kapitel auch nicht die gleichermaßen scharfe Abgrenzung vorgenommen und es werden im nachfolgenden Abschnitt diverse Aspekte zum Design behandelt. Ausgangspunkt für die implementationsnahe Systemanalyse ist das zuvor erstellte Detailkonzept, welches die Systemelemente in Bezug auf den Umfang und Funktionalität umfassend festgelegt hat. Die nun vorgenommene Analyse muss diese detaillierten Vorgaben bezüglich Vollständigkeit und Konsistenz überprüfen.

Bei Individualsoftwareprojekten werden meist die im Detailkonzept erarbeiteten Use-Cases nochmals weiter detailliert und strukturierter dargestellt (s. Abb. 203). Allenfalls werden Vererbungsprinzipien von Use-Cases und weitere Beziehungstypen detailliert. Im Zentrum steht die strukturierte und konsistente Analyse des Objektmodells mit allen seinen Beziehungen.

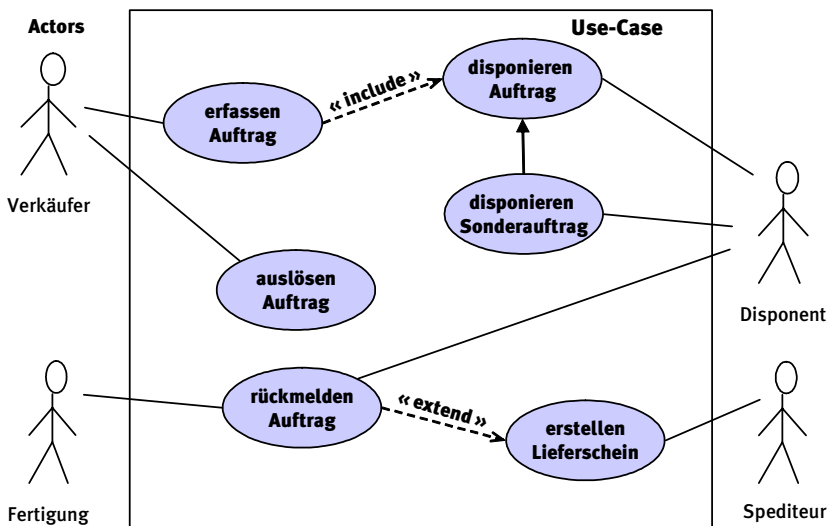
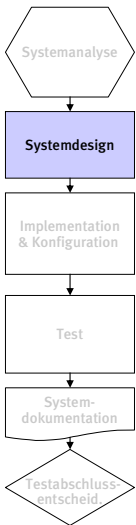


Abb. 203: Use-Cases zur formalen Analyse funktionaler Anforderungen

## 15.3 Systemdesign



### Design der technischen Systemarchitektur

Im Rahmen des Designs muss zunächst eine technische Systemarchitektur definiert werden. Es ist früh zu entscheiden, ob das Projekt mit Hilfe einer Host-, Client/Server- oder einer Multi-Tier-Architektur realisiert werden sollte. Die Wahl der technischen Architektur hat unternehmensbezogene Plattformstandards, Performance-, Wartungskosten- und Weiterentwicklungsansprüche zu berücksichtigen. Die Architektur muss frühzeitig festgelegt werden, da sie das Design der Anwendung stark beeinflusst. So wird in Abhängigkeit der Architektur entschieden, auf welcher Plattfor-mebene die Programmelemente zu implementieren sind – auf der Datenbank-, Anwendungs- oder Clientebene. Idealerweise könnte die Logik unabhängig von der Architektur programmiert werden, doch dies ist meist nicht möglich, da die Ebenen unterschiedliche Sprachen aufweisen.

### Design der Anwendungsarchitektur und der Programmstruktur

Im Rahmen des Systemdesigns ist weiter die definitive Anwendungsarchitektur festzulegen. Es betrifft dies den Entwurf und die Ausgestaltung der Subsysteme, Softwaremodule, Komponenten bzw. die Zuweisung der Einzelfunktionalitäten zu den jeweiligen Systemelementen. Das Design der Komponenten und Programmstrukturen richtet sich in objektorientierten Softwarelösungen primär nach dem Design der Objektklassen (s. u.). Das Objektdesign stellt daher bei objektorientierten Anwendungen gleichzeitig auch das Programmdesign dar.

### Design der Datenbank

Die Speicherung (Persistenz) von Objekten muss im Rahmen des Designs explizit definiert werden, da die Programmiersprachen die Objekte in der Regel nicht “automatisch” in eine Datenbank abspeichern. Mit Hilfe einer speziellen Datenbanksprache muss daher bei der Implementation definiert werden, welche Objekte abzuspeichern sind. Meist erfolgt dies auf Basis einer relationalen Datenbank. Beim Einsatz einer relationalen Datenbank muss in Ergänzung zum objektorientierten Modell zusätzlich ein relationales Modell erstellt werden, ein “Entity-Relationship-Modell”. Die Abbildung (engl.: “Mapping”) des Objektmodells auf das relationale Modell stellt einen zusätzlichen Modellierungsaufwand dar.

### Design der Benutzungsoberflächen (User Interface Design)

Im Rahmen des Systemdesigns sind auch Gestaltungsentscheidungen hinsichtlich der Darstellung der Anwendungsobjekte im Sinne der Benutzungsoberflächen vorzunehmen. Dies beinhaltet die Gestaltung von Bildschirmlayouts, Interaktionselementen, Eingabefeldern, Auswahl-listen, aber auch die definitiven Entwürfe von Druckvorlagen und Listen.

### 15.3.1 Design der Objekte

Während in den vorangegangenen Gestaltungsschritten (Grobkonzept, Konzept, Detailkonzept) die Objektsicht nur sehr eingeschränkt betrachtet wurde, wird im Rahmen des Systemdesigns der Schwerpunkt auf die Objektsicht verlegt. Bei Standardsoftwarelösungen trifft dies allenfalls nicht gleichermaßen wie bei Individuallösungen zu.

Der Gestaltung des Objektmodelles kommt eine wichtige Doppelfunktion zu. Das Objektmodell muss zunächst den Ansprüchen der Anwender hinsichtlich der Geschäftsobjekte genügen. Es stellt aber gleichzeitig auch die Grundlage der Programmstruktur dar. Der Programmcode einer objektorientierten Anwendung wird ja gemäss diesen Objekten strukturiert. Dies unterstreicht die Bedeutung des Objektdesigns. Über die Methoden des optimalen Designs gibt es daher eine umfangreiche Literatur.

Der Weg zum optimalen Objektdesign wird nicht geradlinig verlaufen. Ein erster Ansatz des Objektmodells ergibt sich z.B. aus der Analyse der Use-Cases. Mit Hilfe einer Textanalyse können aus den Substantiven der Use-Cases die Kandidaten für Klassen und aus den Verben die Kandidaten für Methoden abgeleitet werden. Damit entsteht ein erster Entwurf eines Objektmodells, welches anschließend in Bezug auf die Beziehungen zwischen den Klassen analysiert wird (s. Abb. 204). Die weitere Verfeinerung dieses "Domain Object Models" führt zu einem ausgereiften Klassendiagramm mit allen Methoden, Attributen und Beziehungen.

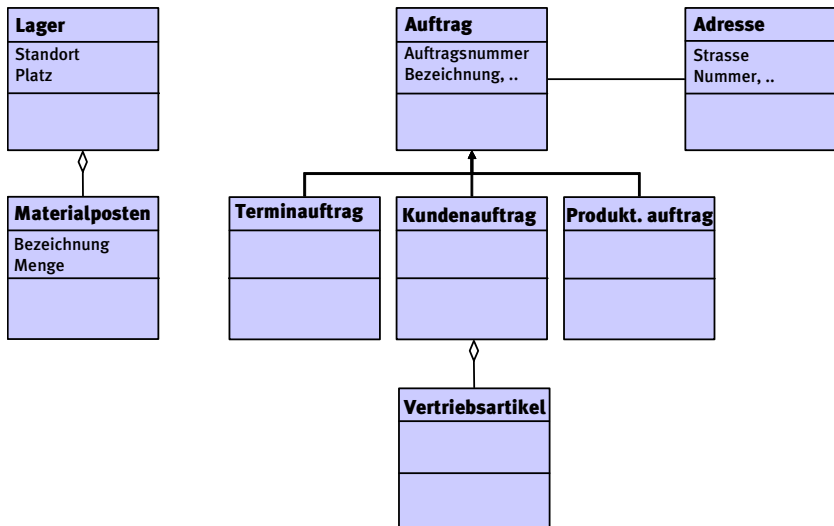


Abb. 204: Design des Objektmodells

### 15.3.2 Design der Prozesse

Im Anschluss an das Design der Objekte können in Verbindung mit den abzudeckenden Prozessen Sequenzdiagramme (s. Abb. 205) und Zustandsübergangsdiagramme erstellt werden. Diese helfen zur Überprüfung des Designs und zur Aufdeckung noch unentdeckter Inkonsistenzen. Sequenzdiagramme wurden in der Vergangenheit eher vernachlässigt. Dabei mag mitgespielt haben, dass sich objektorientierte Techniken erst in den letzten Jahren wirklich etabliert haben. Durch die zunehmende Bedeutung der objektorientierten Ansätze im Bereich unternehmensweiter Softwarelösungen könnten Sequenzdiagramme unter Umständen auch bereits in der Spezifikationsphase an zusätzlichem Stellenwert gewinnen.

In der unten stehenden Grafik greifen wir nochmals auf das Beispiel eines Auftragsabwicklungsprozesses zurück. Der Kundenauftrag (Fahrrad) habe die Eigenschaft, dass sich die Vertriebsauftragsstruktur von der Produktionsauftragsstruktur klar unterscheidet. Vertriebsaufträge müssen jeweils für die Produktion in zwei separat zu bearbeitende Produktionsaufträge A und B überführt werden. Demnach sollen alle farblackierten Teile in einem Produktionsauftrag A und alle unlackierten Teile in einem zweiten Produktionsauftrag B gefertigt werden.

Diese Transformationsanforderung an das Informationssystem ist nicht trivial und die entsprechenden Regeln müssen daher explizit spezifiziert werden. Die im Sequenzdiagramm dargestellten Objekte stellen Auftragsobjekte dar und können dem Anwendungskontext entnommen werden.

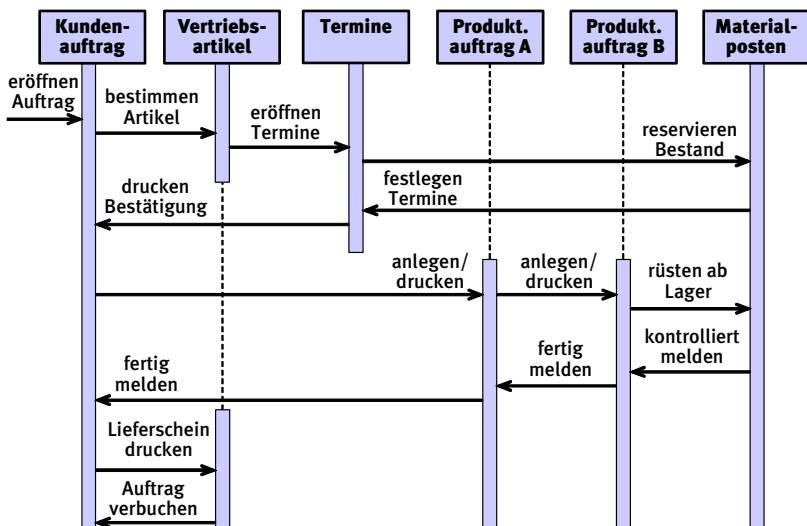
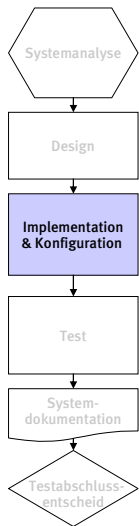


Abb. 205: Sequenzdiagramme zur Konsistenzprüfung

## 15.4 Implementation und Konfiguration

### 15.4.1 Implementation



Nach den diversen analytischen und konzeptionellen Vorarbeiten erfolgt im Rahmen des Implementationsschrittes der eigentliche “Systembau”. Die Prozesse, Funktionalitäten, Objekte und Benutzerrollen müssen mit Hilfe einer Programmiersprache “in Sprachform modelliert” werden. Bei objektorientierten Programmiersprachen wird jede Klasse mit ihren Attributen formuliert und die ihr zugeordneten Funktionen (Methoden) werden programmiert. Objektorientierte Programmiersprachen unterstützen dabei die oben dargestellten Modellierungskonzepte wie Aggregations-, Vererbungs- und Beziehungsmechanismen.

Die Programmerstellung erfolgt in einer von der Programmiersprache extrem klar vorgegebenen Syntax. Während bereits im Design ein recht hoher Anspruch an die Konsistenz und Exaktheit bestand, so wird das Softwareprogramm keinerlei Inkonsistenzen mehr tolerieren. Nachdem die Systemelemente in einem “Quellprogramm” (engl.: “Sourcecode”) beschrieben wurden, wird der Source mit Hilfe eines Übersetzers (engl.: “Compilers”) in eine “Maschinensprache” (engl.: “Binary Code”) übersetzt. Letztlich versteht ein Computerprozessor ja nicht Text, sondern nur ein binäres Programm. Der Compiler überprüft beim Übersetzungsvorgang das Programm auf Inkonsistenzen und zeigt diese an.

Die Softwareentwicklung ist einem schrittweisen Verfeinerungsprozess unterworfen (stepwise refinement) (Wirth 1971). Auf Basis einer ersten Implementation und Fehlerrückmeldungen werden in rascher Abfolge neue Versionen erzeugt, überprüft und verbessert. Dies führt zum Problem einer adäquaten Versionsverwaltung. In größeren Entwicklungsteams sollte der aktuelle Code möglichst breit gestreut werden, man steht hier aber im Konflikt zwischen der Stabilität und der Aktualität der Versionen.

Neben der Programmierung der Anwendungsfunktionalitäten müssen auch in gleicher Weise Schnittstellen zu etwaigen Drittsystemen implementiert werden. Weiter sind unter Umständen zusätzliche Prozeduren zur Datenübernahme (Migration) aus bestehenden Systemen bereitzustellen.

#### Implementation der Datenbank

Neben der Implementation der Anwendungslogik muss wie erwähnt auch die sichere Abspeicherung von Daten in der Datenbank sichergestellt werden (Persistenz). Dies bedeutet, dass eine Datenbank installiert und in der Datenbank die entsprechenden Strukturen angelegt werden müssen (Tabellen, Attribute, Zugriffsrechte etc.). Auf diese kann das Anwendungsprogramm anschließend zurückgreifen. Diese Definition erfolgt mit Hilfe einer datenbankeigenen Datendefinitionssprache “DDL”.

### 15.4.2 Konfiguration

Bei Standardsoftwareprojekten wird meist nur ein sehr geringer Anteil der Software neu programmiert. Die Systemeigenschaften werden in diesen Fällen durch eine so genannte “Konfiguration” auf die Prozesse des Unternehmens angepasst. Standardsoftwarepakete sehen also bereits gewisse “Schalter und Anpassungsmechanismen” zur Konfiguration vor.

Die Konfiguration von Standardsoftware besteht aus der Einstellung von Systemparametern im Hinblick auf die effektiven betrieblichen Anforderungen. Die Mehrzahl der auf dem Markt erhältlichen ERP-Systeme hat weit über 500 Systemparameter, die Art und Aussehen der Anwendung in gewissen Grenzen steuern können. Es können dies beispielsweise sein: Art und Anzahl der Prozessstatus, Merkmale der Benutzungsoberflächen, (z.B. Warnverfahren) und diverse Merkmale von Stammdaten.

In ähnlicher Weise werden Reports für die unterschiedlichen Prozesse konfiguriert werden. Auch hier kann bei Standardsoftwareprojekten meist auf Beispielreports zurückgegriffen werden. Wie oben erwähnt, macht diese Arbeit einen nicht unbedeutenden Projektanteil aus. Außerdem werden im Rahmen der Konfiguration oft auch bereits erste Arbeitsplätze zum Aufbau von Datenstämmen in Betrieb genommen werden. Dies bedingt allerdings eine Schulung der betreffenden Personen, die eventuell schon Daten erfassen sollten.

Natürlich setzt dieses Vorgehen im Rahmen der Konfiguration voraus, dass die notwendigen Hardware-, Kommunikations- und Softwarevoraussetzungen geschaffen wurden, zumindest bei einigen Arbeitsplätzen. Die Projektleiter und Systemverantwortlichen werden dadurch auch in die Lage versetzt, bereits auch erste eigene Schritte am System zu unternehmen. Während des Stadiums des initialen Prototyps sollten nach Möglichkeit auch schon erste Tests einer nötigen Stammdatenübernahme durchgeführt werden. Es bleibt dann nämlich noch genügend Zeit zum Bereinigen der Daten bzw. zu einer notwendigen Anpassung des Konzeptes.

Im Rahmen einer CRM-Einführung wären z.B. im Anschluss an die Konfiguration folgende Arbeiten vollzogen:

- Kampagnenprozessarten sind gemäß den Anforderungen angelegt
- Wichtigste Reports sind definiert und angelegt
- Verfügbare Marketingunterlagen sind weitgehend erfasst
- Zugriffsrechte definiert
- Ein erster vollständiger Kampagnenprozess lässt sich mit den wichtigsten Eigenschaften abwickeln

### 15.4.3 Morphologische Matrix für Design und Implementation

Die Morphologie in Design und Implementation bezieht sich jeweils auf einzelne Systemelemente, Einzelfunktionen oder einzelne Objekte. Sie ist nicht geeignet, Gesamtsystemvarianten zu erzeugen.

Diese Morphologie beschränkt sich auf Designausprägungen der untersten Systemebene. Die Vielzahl der Parameter zeigt auf, dass im Rahmen des Designs nach wie vor eine ganze Reihe von einzelnen Entscheidungen zu treffen sind. Die Reihenfolge der Parameter ist daher auch nicht von besonderer Aussagekraft (s. Abb. 206).

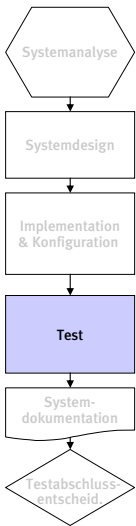
Viele der Parameter sind fast rein technischer Natur und mit beschränkter Auswirkung auf den Benutzer. So muss z.B. die Schicht der Funktionalität, die Art der Datenspeicherung, die Art des Reportings und die Anpassung der Software die Benutzer nicht unmittelbar betreffen.

Im Rahmen des Designs werden allenfalls gewisse Prioritäten nochmals überprüft. Weiter ist zu entscheiden, ob die Konfiguration auf Basis einer vorbestehenden Funktion erfolgen kann und wenn nein, ob die Anpassung releasefähig ist.

<b>Ausprägung</b> <b>Parameter</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>
<b>Priorität der Funktionalität</b>	Hohe Priorität, für Release zentral	<b>Normale Priorität</b>	Geringe Priorität, nächster Release	Verschieben auf Weiteres
<b>Konfigurations-element</b>	<b>Vorbestehende Funktionalität</b>	SW-Änderung releasefähig	SW-Änderung nicht releasefähig	Verzicht auf Funktionalität
<b>Schicht der Funktionalität</b>	<b>Datenbank Stored Procedure</b>	Applikations-funktionalität	Script in Laufzeit-umgebung	Drittlösung (z.B. Excel)
<b>Komplexität des Elementes</b>	<b>Hierarchisches Objekt</b>	Normales Objekt	Attribut eines Objektes	Einfaches Textfeld
<b>Datenherkunft</b>	<b>Applikation</b>	andere Lösung synchronisiert	Manuell abgeglichen	...
<b>Datenänderungs-frequenz</b>	<b>Konfigurations-daten</b>	Stammdaten	Bewegungs-daten	...
<b>Daten-speicherung</b>	<b>Datenbank-element</b>	Fileelement (z.B. Logfile)	Registry	Keine Speicherung
<b>Navigation</b>	Menü-navigation	<b>Dropdown-Menü</b>	Auswahlliste, Option	Keine Vorgabe
<b>Reportingtools</b>	Internes Reporting	<b>Angebundenes Reporting Tool</b>	Reporting mit Tabellenkalkulat.	Kein Reporting
<b>Verbreitung in Applikation</b>	Lokale Wirkung nur in einer GUI	<b>In mehreren GUI sichtbar</b>	In mehreren GUIs mehrere Stellen	...

Abb. 206: Beispielhafte morphologische Matrix im Designprozess

## 15.5 Test



Während der Entwicklung werden die entwickelten Teile fortlaufenden Tests unterzogen. Eine geordnete Entwicklung hat sich außerdem an ein Testkonzept zu halten, welches zwischen *Modul-, Integrations- und Systemtests* unterscheidet. Modultests testen die Eigenschaften von Modulen, Integrationstests überprüfen das Zusammenspiel der Module und der Systemtest überprüft den gesamten Systemumfang.

Idealerweise wird für jede Teststufe ein eigenes System aufgesetzt. Die Modultests werden direkt auf dem *Entwicklungssystem* durchgeführt. Für die Integrationstests wird ein spezielles *Testsystem* bereitgestellt, welches mit dem späteren *Produkktivsystem* identisch sein sollte. Alternativ wird gelegentlich auf dem Entwicklungssystem parallel ein “Testmandant” eingerichtet, auf welchem die Integrationstests durchgeführt werden können. Alle Tests sind idealerweise auf dem Produkktivsystem zu wiederholen, um absolut sicherzustellen, dass das System funktionsfähig ist.

Tests und Prüfungen haben sich stets an einer klaren Testvorgabe zu orientieren. Das deutsche “V-Modell” zeigt dies gut auf (s. Abb. 207). Es verknüpft das herkömmliche Wasserfallmodell mit Integrationsstufen auf allen Ebenen und entsprechenden Tests (V-Modell 1997).

Es zeigt auf, dass sich die Abnahme am Konzept und am Pflichtenheft messen muss, Systemtests am Detailkonzept und Integrationstests am jeweiligen Design.

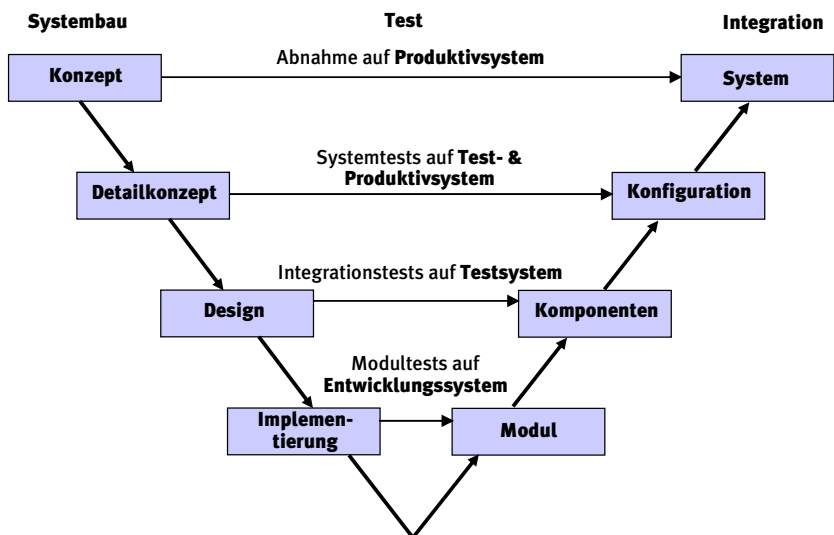
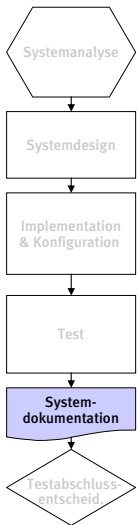


Abb. 207: V-Modell



## 15.6

## Systemdokumentation



Die Dokumentation eines IT-Systems hat auf unterschiedlichen Ebenen zu erfolgen. Zunächst ist für das Unternehmen das **Organisationshandbuch** zu erstellen oder nachzuführen. Es regelt die organisatorischen Belange der Geschäftsprozesse und wird idealerweise in Abstimmung mit einer schon bestehenden Qualitätssicherungsdokumentation erstellt.

Die so genannte **Anwendungsdokumentation** bietet den Benutzern eine Hilfestellung und ein Nachschlagewerk für die Arbeit mit dem System. Die Anwendungsdokumentation ist in enger Abstimmung mit der heute geforderten Online-Hilfe zu erstellen.

Drittens muss eine **Technische Systemdokumentation** erstellt werden, welche das System mit seinen Programm- und Objektstrukturen in allen Details dokumentiert. Es stellt sicher, dass die Weiterentwicklung zukünftigen Entwicklern übergeben werden kann.

Diese Unterteilung beruht einmal mehr auf dem Prinzip der unterschiedlichen Systemebenen. Das Konzept legte die Rahmenbedingungen fest und bildet daher auch die Grundlage für das Organisationshandbuch.

Das Detailkonzept beschreibt die Prozessabläufe mit allen Prozessstatus und bildet die Grundlage für die Erstellung des Anwendungshandbuchs. Die Design- und Implementationstätigkeit führt direkt zur technischen Systemdokumentation.

Eine Anwendungsdokumentation kann sich typischerweise etwa wie folgt gestalten:

#### I ALLGEMEINES

- Start und Beendigung der Anwendung
- Allgemeiner Bildschirmaufbau und Benutzungsführung
- Allgemeines zu Druckerzeugnissen und Reports

#### II ÜBERSICHT DER ANWENDUNGSGBIETE

- Beschreibung der Anwendungsgebiete und Geschäftsprozesse
- Schnittstellen zwischen den Anwendungsgebieten

#### III DETAILLIERTE BESCHREIBUNG DER ANWENDUNGSGBIETE 1-N

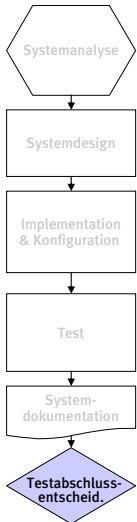
- Funktionsübersicht Anwendungsgebiet
- Beschreibung des Prozessabläufe und der Bildschirmmasken

#### IV EXPORTFUNKTIONEN

#### V KONFIGURATION DER ANWENDUNG

#### VI SCHNITTSTELLEN ZU DRITTANWENDUNGEN

## 15.7 Testabschlussentscheidung



Die Entscheidung hinsichtlich des Abschlusses der Tests entspricht gewissermaßen auch die Entscheidung zum Abschluss der Entwicklungsarbeiten. Bei der Zusammenarbeit mit einem Lieferanten wird der Lieferant diese Entscheidung autonom zu fällen haben und damit gleichzeitig dem Kunden die Bereitschaft des Systems zur Abnahme melden.

Die Entscheidung zur Freigabe der Abnahmeprüfung ist damit für den Kunden noch von keiner weiteren Tragweite, da er immer die Möglichkeit hat, bei einer erfolglosen Abnahmeprüfung die Wiederaufnahme der Entwicklung und entsprechender Tests zu veranlassen.

Allerdings ist die Entscheidung zum Testabschluss selbstverständlich aus Lieferantensicht eine ganz massgebliche. In der Regel stehen nicht beliebig viele Abnahmeversuche zur Verfügung. Auch wenn diese Einschränkung kein Problem darstellen würde, so ist unter Umständen mit einer schlechten Presse oder finanziellen Folgen zu rechnen, wenn das System die Abnahme nicht besteht.

Bereich Schritt	<b>Sozio-technisches System</b>	<b>Einführung &amp; Ausbildung</b>	<b>Betrieb &amp; Unterhalt</b>	<b>Projektmanagement &amp; Controlling</b>
<b>Systemanalyse</b>	Systemanalyse Prozesse; Systemanalyse Funktionen; Systemanalyse Objekte; Systemanalyse Aufgaben; Systemanalyse Technik.	Feinanalyse Ausbildungsbedarf.	Betrieb der Entwicklungs- & Testumgebung.	Projektplanung; Projektcontrolling; QS-Prüfplan.
<b>Systemdesign</b>	Oberflächendesign; -prototyp; Funktionale Designvorschläge; Datenbankdesign; Konfigurationsdesign; Schnittstellendesign; Systemtechnisches Design.	Design der Datenübernahme; Ausbildungsdesign.		
<b>Implementation &amp; Konfiguration</b>	Implementierte Funktionalität; Konfiguriertes System; Vorerfasste Stammdaten; Testplanung; Modul- & Systemtests; Integrationstests; Volumen- und Stresstests; Testprozeduren, -protokolle; Benutzerprofile und -rollen.	Migrationsprozeduren; Systemintegrationsplan; Schulungsunterlagen; Ausbildungsplanung; Ausbildungslokalitäten.	Schulungssystem; Datenschutz- & -sicherheitsrichtlinie.	Änderungsanträge; Änderungsstatusliste; Fehler-, Mängelliste.
<b>Test</b>	Bewertung Implementation & Tests.			
<b>Systemdokumentation</b>	Organisationshandbuch; Anwendungsdokumentation; Technische Systemdokument.		Betriebshandbuch; Supporthandbuch.	
<b>Testabschlussentscheidung</b>				Testabschlussentscheidung.

Abb. 208: Ergebnisübersicht der Design- und Implementationstätigkeit

## 15.8 Literatur zu Design und Implementation

- Agiplan (1993): PPS erfolgreich einführen und nutzen, Praxis-Studie. Mülheim/Ruhr.
- Bauknecht, K. (Hrsg.) (1992): Informatik-Anwendungsentwicklung – Praxiserfahrungen mit CASE: Probleme, Lösungen und Erfahrungen bei Einführung und Einsatz von CASE. Stuttgart: Teubner.
- Ehn, P., Kyng, M. (1991): Cardboard Computers: Mocking-it-up or Hands-on the Future. In: Design at Work: Cooperative Design of Computer Systems; (Greenbaum, J., Kyng, M.; Hrsg.). Hillsdale, NJ: Lawrence Erlbaum. 169–196.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999): The Unified Software Development Process. Reading, MA: Addison Wesley Longman.
- Kay, A. (1990): User Interface: A Personal View. In: The Art of Human-Computer Interface Design. (Laurel, B.; Hrsg.); Reading, MA: Addison-Wesley. 191–207
- Kyng, M. (1995): Creating Contexts for Design. In: Scenario-Based Design – Envisioning Work and Technology in System Development. (Carroll, J.; ed.). New York: John Wiley & Sons. 85–107.
- Norman, D., Draper, S. (1986): User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum.
- Rieder, B. (1988): Die Gestaltung des Implementierungsprozesses bei der Einführung von integrierter Standardsoftware. Dissertation, Regensburg.
- V-Modell (1997): Entwicklungsstandard für IT-Systeme des Bundes. Allgemeiner Umdruck Nr. 250. Berlin.
- Wirth, N. (1971): Program Development by Stepwise Refinement. In: Communications of the ACM, Vol. 14, No. 4. New York: ACM Publications.
- Zehnder, C.A. (1998): Informationssysteme und Datenbanken, 6. Aufl. Zürich: vdf Hochschulverlag.



# 16

## **Abnahme und Inbetriebnahme**

### **Kapitelinhalt**

- Überprüfung der effektiven Bereitschaft zur Systemumstellung
- Aufsetzen des Produkktivsystems und Schulung der Benutzer
- Vornahme einer Abnahmeprüfung; Protokollierung der Ergebnisse
- Produktivstart und rascher Support erster Probleme

Die Abnahme und die nachfolgende Inbetriebnahme des Informationssystems bedeuten den Abschluss der Projektarbeiten und damit bei erfolgreicher Abnahme die Übergabe des Systems an den Kunden sowie die Übergabe der Verantwortung für das System vom Lieferanten an den Kunden. Gelegentlich wird diese Tätigkeit auch als “Einführung” bezeichnet. Dieser Begriff könnte allerdings missverstanden werden, insofern er auch die vorhergehenden Tätigkeiten wie die Detailspezifikation mit umfassen könnte.

Spätestens in diesem Zeitpunkt ist die Schulung der Benutzer anzugehen. Hier stellt sich häufig die Frage nach dem richtigen Zeitpunkt. Werden die Benutzer zu früh geschult, können sie das Wissen nicht anwenden, werden sie zu spät geschult, ist die Akzeptanz gefährdet.

Es wird anschliessend die Frage beleuchtet, wie die Umstellung auf ein neues System erfolgen kann. Man spricht in diesem Zusammenhang von so genannten “Roll-out-Konzepten”. Unterschieden werden zunächst die Komplettumstellung und die gestaffelte Einführung.

## 16.1 Inhalt und Vorgehen

### 16.1.1 Inhalt

Im Rahmen dieses Schrittes wird konzeptionell höchstens noch ein Feinschliff vorgenommen und beispielsweise geklärt, an welchen Tagen die Umstellung erfolgt, wie die Verantwortlichkeiten liegen, welche Sicherheitsmassnahmen einzuleiten sind etc. Die grundlegenden Einführungs- bzw. Roll-out-Konzepte wurden selbstverständlich bereits im Rahmen der Konzepterstellung geprüft und konzipiert. Aus didaktischen Gründen erfolgt die Darstellung unterschiedlicher Einführungskonzepte (gestaffelte Einführung, Komplettumstellung etc.) aber erst in diesem Kapitel.

### 16.1.2 Vorgehen

Auch die Abnahme- und Inbetriebnahmetätigkeit beginnt idealerweise mit einer Analysephase – in diesem Fall mit einer Bereitschaftsanalyse, welche feststellt, ob das System grundsätzlich für die Abnahme und Inbetriebnahme bereit erscheint (s. Abb. 209).

Die Ausbildung der Benutzer kann dann flächendeckend beginnen, wenn klar ist, ob und zu welchem Zeitpunkt der Produktivstart erfolgen wird.

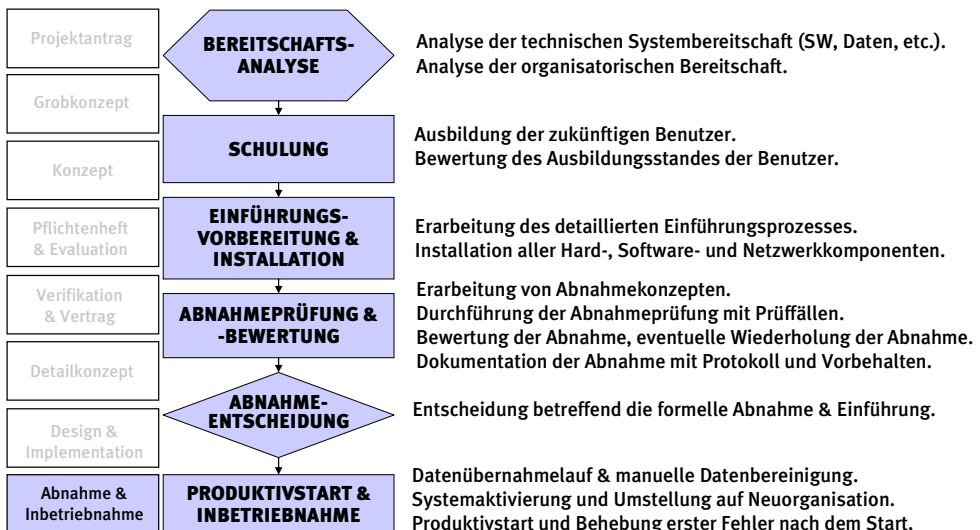
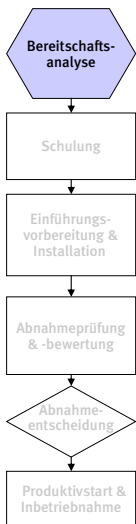


Abb. 209: Vorgehensschritte der Abnahme- und Einführungstätigkeit

## 16.2

**Bereitschaftsanalyse**

Nach der Tätigkeit der Implementation und des Tests kommt die kritische Phase des Übergangs in den so genannten Echtbetrieb. Der Zeitpunkt der Inbetriebnahme ist nah und es sollte im Rahmen einer so genannten Bereitschaftsanalyse (engl.: “Readiness”) sauber geklärt werden, inwieweit das System bereit steht und wo noch offene Punkte bestehen.

Ein möglicher erster Test im Hinblick auf dieses Ereignis ist ein Probelauf der Datenübernahme von Bewegungsdaten. Es hat weiter eine Analyse der von den Benutzern zu erfassenden Stammdaten zu erfolgen. Da das System in dieser Projektphase zur Datenerfassung zur Verfügung steht und größere Konfigurationsänderungen nicht mehr zu erwarten sind, ist dies meist auch problemlos machbar.

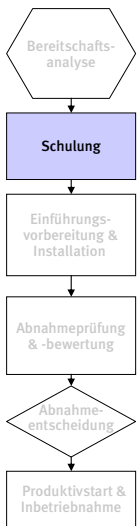
Es genügt aber häufig nicht, nur Stammdaten zu übernehmen. Vielmehr müssen auch so genannte Bewegungsdaten übernommen werden. Nach dem Aufbau der Stammdaten müssen bei der Systemeinführung beispielsweise die effektiven Lagerbestände mit den wahren Beständen übereinstimmen. Die Übernahme der Bewegungsdaten ist meist um ein Vielfaches komplizierter als die Stammdatenübernahme. Man sollte daher nur die absolut notwendigsten Daten übernehmen.

Ein erster Probelauf dieser Bewegungsdaten erfolgt in der Regel rund einen Monat vor dem Produktivstart. Und nun beginnt nochmals eine Testphase, diesmal aber mit den echten Daten im Sinne einer Simulation. Eventuell müssen aufgrund dieser Erfahrungen nämlich nochmals letzte Korrekturen vorgenommen werden, insbesondere was die Übernahme der Bewegungsdaten anbelangt.

Häufig erfolgt dann rund eine Woche vor Produktivstart die definitive Datenübernahme. Dies bedeutet aber, dass bis zum definitiven Produktivstart das abzulösende und das neue System parallel weitergeführt werden, da die definitive Datenübernahme ja bereits erfolgt ist.

Erst am Tag des Produktivstarts, meistens an einem Wochenende oder in der Urlaubszeit, erfolgt dann die eigentliche Inbetriebnahme oder Umstellung. Nach der Inbetriebnahme muss man sich auf eine Periode allgemeiner Unsicherheit gefasst machen, sowohl auf der Technik- als auf der Anwendungsseite. Man könnte auch von einer Tuningphase sprechen. Erfahrungsgemäß treten in der ersten Nutzungsphase viele Fragen und Bedürfnisse auf. Es gilt, diese für die Anwender besonders schwierige Zeit möglichst günstig zu überbrücken. Auch für diese Tuningphase sollte die Bereitschaft überprüft werden.

### 16.3 Schulung



Spätestens im Rahmen der Abnahme und der Inbetriebnahme muss eine intensive Schulung der Benutzer auf dem System erfolgen. Der richtige Zeitpunkt der Schulung ist häufig schwierig zu bestimmen. Die Schulung sollte nicht zu früh angesetzt werden, da sonst das erlernte Wissen wieder verloren geht. Ein wesentliches Ziel jeder Schulung muss nämlich sein, das Wissen sofort und direkt in der Praxis umzusetzen. Wie wir im nachfolgenden Kapitel sehen werden, kommt der Schulung eine absolut zentrale Rolle für den Erfolg des Projektes zu.

Eine Besonderheit von IT-Einführungen besteht darin, dass sich solche betrieblichen Standardsoftwarelösungen ganz direkt und fundamental auf die Arbeit vieler Mitarbeiter auswirken. Eine solche Veränderung in der täglichen Arbeit provoziert Widerstände und Konflikte der Beteiligten.

In einer Studie in hundert mittelständischen deutschen Industrieunternehmen wurden die Erfolgsfaktoren bei der Einführung von ERP-Software detailliert untersucht. Betrachtet wurde u.a., wie sich die Art der Schulung bzw. die Anzahl der Schulungstage auf die Zufriedenheit der Benutzer und auf die generelle Systemakzeptanz auswirkt. So scheint gerade die Schulung über den Erfolg und über die Akzeptanz einer Systemlösung zu entscheiden. Die Akzeptanz nimmt mit steigender Anzahl an Schulungstagen kontinuierlich zu und scheint keine Obergrenze zu kennen (Martin 1993, S. 102). Wichtig ist in diesem Zusammenhang die bereichsübergreifende Schulung. Es sollten also insbesondere auch Zusammenhänge vermittelt werden, die Querbezüge ermöglichen.

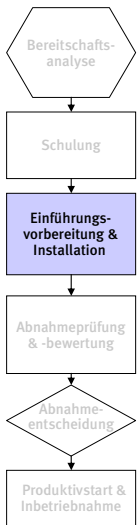
Darüber hinaus erscheint, dass mit Hilfe von Schulungsanstrengungen auch entsprechende Systemdefizite korrigiert werden können. Es hat sich gezeigt, dass für eine hohe Zufriedenheit im Durchschnitt ein Schulungspensum von ca. zwölf Tagen erforderlich ist.

Diesem Ergebnis muss eine Befragung der Systemanbieter entgegengestellt werden. Letztere sind in Diskrepanz zum vorherigen Ergebnis der Ansicht, ein Schulungsumfang von ca. 5 bis 9 Tagen reiche aus. Die Bedeutung der Schulung wird demnach von Seiten der Anbieter häufig nicht erkannt oder bewusst heruntergespielt. Fatalerweise deckt sich diese Strategie aus Kostengründen mit den Zielen der anwendenden Unternehmen. Diese sind nämlich nicht an hohen Kosten für die Einführung interessiert. Gerade bei der Schulung sollte nun aber gemäß obigen Ergebnissen nicht gespart werden. So zeigt es sich, dass auch suboptimale Systeme bei einer genügenden Anzahl von Schulungstagen eine recht gute Akzeptanz erreichen können.



## 16.4 Einführungsvorbereitung und Installation

### 16.4.1 Komplettumstellung versus gestaffelte Einführung



Der nächste Abschnitt behandelt die Frage, wie die Einführung des Systems im Einzelnen erfolgen sollte. Das Konzept hierfür wird selbstverständlich bereits im Rahmen der Projektierung erstellt. Aus Gründen des thematischen Kapitelaufbaus werden die Einführungsvarianten erst an dieser Stelle vertieft behandelt.

Betriebliche Standardsoftwarepakete umfassen in aller Regel diverse Module. Daher stellt sich die Frage, ob nun alle Module gleichzeitig abgelöst werden sollten, oder ob man in Betracht ziehen soll, die Umstellung sukzessive Modul für Modul vorzunehmen.

Eine Expertenbefragung des Forschungsinstitutes für Rationalisierung in Aachen (FIR) kam zum Schluss, dass die Hauptvorteile einer Komplettumstellung darin liegen, dass keine temporären Schnittstellen entstehen (s. Abb. 210) (Schmitz 1998). Es kann nämlich relativ kostenaufwändig sein, wenn lediglich für eine Übergangsfrist Schnittstellen programmiert werden müssen. Aber auch der Vorteil der kürzeren Projektdauer spricht im Urteil der Experten für eine Komplettumstellung.

Nachteilig für eine Komplettumstellung sind dabei mögliche personelle Engpässe (erwähnt von 71 Prozent der Experten) und das insgesamt höhere Risiko dieses Vorgehens.

#### Vorteile Komplettumstellung

#### Nachteile Komplettumstellung

<b>Keine temporären Schnittstellen (85%)</b>	<b>Personelle Engpässe (71%)</b>
Kürzere Projektdauer (69%)	Zu hohes Risiko (46%)
Höhere Transparenz des Projektes (59%)	Stabilität Auftragsabwicklung leidet (34%)
Geringere Kosten	Hohe Belastung der Mitarbeiter
Schnellerer Erfolg	Hohe Anforderung an Projektmanagement
Hohe Akzeptanz der Benutzer	
Keine Parallelorganisation	
Effektiveres Arbeiten	
Keine Doppelerfassung der Daten	

Abb. 210: Komplettumstellung versus gestaffelte Einführung (Schmitz 1998)

### 16.4.2 Stichtagsumstellung versus Parallellauf-Phase

Im Zusammenhang mit der Umstellung von System A auf System B stellt sich noch eine zweite Frage: Soll die Umstellung schlagartig erfolgen, d.h. per Stichtag, oder soll es eine gewisse Zeitdauer geben, in welcher beide Systeme im produktiven Betrieb nebeneinander laufen?

Dies ist ganz offensichtlich eine andere Fragestellung als jene nach der Komplettumstellung. Bei der Stichtagsumstellung geht es um die Frage, ob beispielsweise das Modul "Vertriebsabwicklung" gleichzeitig auf dem alten System und dem neuen System zur Verfügung steht, das alte System gewissermaßen als Backup-System, falls einiges schief laufen sollte. Es ist dies eine Frage des Migrationsverfahrens. Die Anwender haben damit die Wahl, auf welchem System sie die Aufträge abwickeln wollen. Auch hierüber wurden die Experten befragt (s. Abb. 211) (Schmitz 1998). Die Vorteile einer Stichtagsumstellung überwiegen klar. Insbesondere wird der Vorteil einer einheitlichen Arbeitsweise genannt.

Des Weiteren sehen die Experten den Vorteil der kürzeren Projektdauer und einer höheren Motivation der betroffenen Mitarbeiter. Als Nachteil wird erwähnt, dass bei einer Stichtagsumstellung in der ersten Zeit mit einer wenig stabilen Auftragsabwicklung zu rechnen ist. Das Hauptproblem bei einem Parallellauf ist zusätzlich, dass beispielsweise die Bewegungsdaten in gewissen Fällen nur stimmen werden, wenn wirklich beide Systeme stets mitgeführt werden.

#### Vorteile Stichtagsumstellung

<b>Einheitliche Arbeitsweise</b>	<b>(85%)</b>
Kürzere Projektdauer	(85%)
Günstig bei kleinen Betrieben	(69%)
Höhere Motivation	(54%)
Geringere Belastung	(46%)
Grössere Transparenz	
Keine Parallelorganisation	
Zwang zum Neuen	
Beteiligung der GL höher	

#### Nachteile Stichtagsumstellung

<b>Wenig stabile Auftragsabwicklung</b>	<b>(31%)</b>
Keine Sicherheitslösung	
Sehr hoher Aufwand der Umstellung	
Hohe Belastung nach dem Stichtag	
Kein kontinuierlicher Vertrauensaufbau	
Hohes Risiko	
Hoher Testaufwand	

Abb. 211: Stichtagsumstellung versus Parallellauf-Phase (Schmitz 1998)

### 16.4.3 Einführungsstrategien in der Praxis aus Sicht der Experten

Schauen wir uns an, was dies aus Sicht der befragten Experten zusammenfassend für die Projektierung einer ERP-Ablösung heißt (s. Abb. 212). Im Grunde werden zwei Verfahren fast gleichwertig bewertet.

Im Urteil der Experten steht die modulweise Ablösung zum Stichtag an oberster Stelle (Schmitz 1998). Sie wird von 40% der befragten Experten bevorzugt. Ein Parallelbetrieb wird als nicht sinnvoll erachtet.

An zweiter Stelle wird das Vorgehen favorisiert, wiederum per Stichtag die gesamte Umstellung aller Module im Sinne eines “Big bang” vorzunehmen.

Nur wenige Experten plädieren für eine wirklich gestaffelte Vorgehensweise, aber nicht nach Modulen, sondern nach Produktionsbereichen. Die Gesamtheit der Module soll also beispielsweise für die Vormontage bzw. die Montage eingeführt werden.

Zusammenfassend ist der Umstand interessant, dass die überwiegende Mehrheit für die Stichtagsumstellung plädiert; nur eine sehr kleine Minderheit kann gemäß dieser Resultate für einen Parallelbetrieb votiert haben.

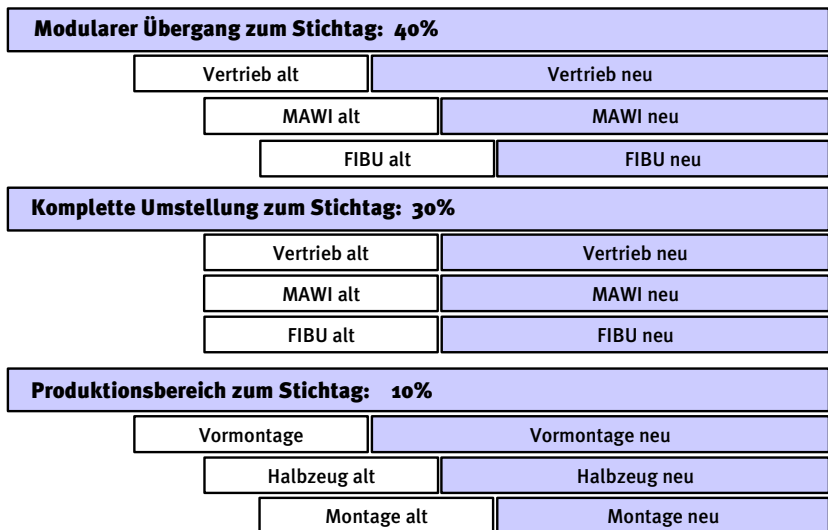


Abb. 212: Roll-Out Konzepte gemäß Expertenbefragung (Schmitz 1998)

#### 16.4.4 Installation

Als integraler Bestandteil der Einführungsvorbereitung muss nun auch die flächendeckende Installation der für den Betrieb notwendigen Hard-, Software- und Netzwerkkomponenten vorgenommen werden. Damit können in der nachfolgenden Abnahme neben den funktionalen Tests auch Belastungstests durchgeführt werden. Es ist das Ziel, dass das System für die Abnahme bereitsteht.

Der Zeitpunkt der flächendeckenden Installation kann von Projekt zu Projekt unterschiedlich sein und wird unter Umständen auch erst nach einer erfolgreichen Abnahme vollzogen. Auf der anderen Seite ist die Installation der Server- und Anwendungsfunktionalitäten bereits im Rahmen der Systemtests erfolgt. Bei größeren Systementwicklungen und -installationen unterscheidet man mehrere Systemstufen zwischen dem Entwicklungssystem und dem so genannten Produktivsystem. Bei der Installation ist nämlich darauf zu achten, dass nach der Installation eines Softwarestandes auf dem Produktivsystem neue Versionen nur in absolut geordneter und ausgetesteter Art und Weise auf das Produktivsystem überspielt werden.

Man steht auch gelegentlich vor dem Dilemma, dass eine zu kurze Abfolge neuer Releasestände und Patches wohl eine Verbesserung des Systems zur Folge haben, andererseits gewisse unerwünschte und nicht vollständig ausgetestete Effekte zeitigen könnte.

##### **Installation der Netzwerk- und Serverkomponenten**

Die Installation erfolgt technisch gesehen "Bottom-up". Das heißt, es muss mit den technisch tiefsten Komponenten begonnen werden, da diese die Voraussetzung für die nachfolgenden Schichten wie Betriebssystem, Middleware und Anwendungssoftware bilden.

##### **Installation der Clients und Peripheriegeräte**

Die Installation der Produktivarbeitsplätze wird sich nach dem gewählten Einführungsprinzip ausrichten.

Für die Verteilung etwaiger Softwarekomponenten auf den Arbeitsplätzen können u.U. Werkzeuge zum Einsatz kommen, welche die Installation automatisieren und damit vereinfachen.

Die Installation von Web-Anwendungen ist aus diesem Grunde viel einfacher, da man keine Verteilung der Software auf dem Client vornehmen muss. Die Software wird bei jedem Start der Anwendung neu auf die Arbeitsplatzrechner geladen.

#### 16.4.5 Datenübernahme

Datenübernahmen stellen in jedem Projekt besonders heikle und sensitive Punkte dar. Auf der einen Seite sind die Anwender ganz direkt betroffen, wenn bestehende Daten nicht übernommen werden können, weil eine vollständige Neuerfassung der Daten in aller Regel viel Aufwand bedeutet.

Andererseits dürfen die Vorteile von automatischen Datenübernahmen nicht überschätzt werden. In den meisten Fällen ist nämlich trotz so genannter automatischer Datenübernahme eine größere Nacharbeit der Daten notwendig. Die Datenstrukturen sind häufig derart verschieden, dass eine gewisse Überarbeitung zwingend ist. Man entscheidet sich daher manchmal dafür, die bestehenden Datenbestände zu bereinigen. Gelegentlich entsteht auf Seiten der Anwender der Eindruck, die vollständige Neuerfassung wäre fast einfacher gewesen als die Bereinigung der Altdaten. Dadurch hätte man aber einen doppelten Aufwand erzeugt, da natürlich auch die automatische Datenübernahme einen bereits nicht geringen Programmieraufwand bedeutet.

Es lohnt sich daher in allen Fällen, sehr detaillierte Überlegungen anzustellen, welche Datenbestände mit Hilfe von welchen Übernahmeprozeduren zu übernehmen sind. Am besten wird diese Aufgabe einem eigenen Teilprojekt übergeben. Damit sind die Verantwortlichkeiten von Beginn an klar und die Aufgaben müssen mit den Benutzern abgesprochen werden.

Es ist außerdem sehr ratsam, die eigentliche Datenübernahme mit einem umfassenden Probelauf anzugehen. Idealerweise erfolgt dieser Probelauf mindestens einen Monat vor Beginn des eigentlichen Produktivstartes. Jede Verzögerung in diesem Stadium kann sich nämlich sonst auf den erfolgreichen Produktivstarttermin auswirken.

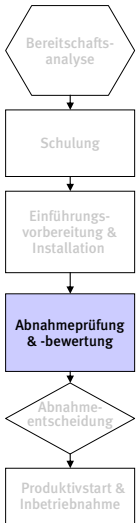
##### **Definitive Datenübernahme**

In den meisten Fällen hat die definitive Datenübernahme der Bewegungsdaten rund eine Woche vor dem Produktivstart zu erfolgen. Damit bleibt genügend Zeit, die allenfalls noch notwendigen Bereinigungen tatsächlich vorzunehmen. Zudem müssen nicht zwei äquivalente Datenbestände über längere Zeit parallel geführt werden. Ist nämlich die definitive Datenübernahme erfolgt, dann müssen auf zwei Systemen Daten gepflegt und aktualisiert werden.

Außerdem muss von Interesse sein, von welcher Qualität die Daten sind und wo mögliche strukturelle Unzulänglichkeiten in den Datenbeständen vorliegen. Dies kann auch dazu dienen, die etwa anstehende Datenübernahme ins rechte Licht zu rücken und zu prüfen, wo es sinnvoll ist, bestehende Datenbestände grundlegend zu überarbeiten.

## 16.5 Abnahmeprüfung und -bewertung

### 16.5.1 Abnahmeprüfung



Vor dem Produktivstart wird mit einer formellen Abnahmeprüfung (Validation, Inspection) beurteilt, ob das Informationssystem den Anforderungen genügt. Grundlage für die Abnahme bilden entsprechende Abnahmeszenarien, welche auf dem ursprünglichen Anforderungskatalog basieren und jede Funktion überprüfen. Zum Tragen kommen hier auch Belastungstests oder zusätzliche Testverfahren, wie sie bereits im letzten Kapitel angeführt wurden.

Für die Systemlieferanten oder Projektverantwortlichen ist das Projekt erst dann erfüllt, wenn aufgrund einer so genannten Abnahmeprüfung eine definitive Abnahme erfolgt ist.

Wie oben bereits erwähnt, ist eine klare Unterscheidung zwischen Tests und Abnahmeprüfung vorzunehmen. Tests stehen in der Verantwortung des Lieferanten. Selbstverständlich sollte die Kundenseite bei gewissen Tests unterstützend mitwirken. Die Verantwortung für die Tests liegt aber auf Seite des Lieferanten.

Anders verhält es sich bei der Abnahme. Hier liegt die Verantwortung auf der Seite des Kunden. Der Kunde wird in diesem Fall Prüfscenarien erstellen, welche selbstverständlich auf den definierten Spezifikationen und dem Detailkonzept beruhen müssen. Im Rahmen der Prüfung sollten nicht nur Funktionalitäten geprüft, sondern auch Belastungstests durchgeführt bzw. absichtlich Fehleingaben produziert werden.

Abnahmen führen in der Praxis häufig zu Meinungsdivergenzen zwischen Kunden und Lieferanten. Meistens wurde nämlich im Vorfeld des Projektes nicht präzise definiert, welche Punkte für die Abnahme tatsächlich maßgeblich zu erfüllen sind. Es wurde im Vertrag einfach summarisch von einer Abnahme gesprochen.

Der Kunde stellt sich nun auf den Standpunkt, alle denkbaren Anforderungen und Wünsche müssten für die erfolgreiche Abnahme abgedeckt sein. Diese Vorstellung ist aber wenig praxisgerecht, da sich immer wieder Anforderungen finden lassen, die im Pflichtenheft in irgendeiner Weise begründet sind. Damit hätte es der Kunde in der Hand, die effektive Abnahme fast endlos zu verzögern.

Es ist daher entscheidend, bereits im Vorfeld ganz klar zu definieren, wie viele offene Punkte für eine erfolgreiche Abnahme bestehen dürfen. Hierzu müssen Anforderungen in zentrale, wichtige und sekundäre klassifiziert werden. Zu jeder dieser drei Klassen ist anzugeben, wie viele offene Punkte eine erfolgreiche Abnahme verhindern.

### 16.5.2 Bewertung der Abnahmeprüfung

Das Ergebnis der Abnahmeprüfung oder jedes Abnahmeversuches wird sauber protokolliert und seitens des Lieferanten und des Kunden mit Ort und Datum unterschrieben. Vorbehalte gegenüber einer vollständigen Abnahme werden ebenfalls schriftlich festgehalten. Zusätzlich muss erwähnt werden, um welchen Vertragsgegenstand es sich handelt und ob eine Gesamt- oder eine Teilabnahme erfolgt ist. Jede Abnahmeprüfung wird also einer formellen Bewertung unterzogen.

Das Prozedere der Abnahmeprüfung muss bereits im Rahmen der Vertragsverhandlungen klar festgelegt werden. Aufgrund der Folgen einer Abnahme hat der Lieferant das Interesse, die Abnahmebestätigung möglichst rasch zu erhalten. Der Kunde wird indessen u.U. versuchen, die Abnahme möglichst lange hinauszuzögern. Daher müssen von Beginn an die Regeln für eine “erfolgreiche Abnahme”, für eine “bedingte Abnahme mit gewissen Mängeln” und für eine “nicht erfolgreiche Abnahme” definiert werden.

Die Regeln basieren auf den bereits in Kapitel “Verifikation und Vertrag” angeführten Fehlerkategorien (“Severity-Stufen”). Diese klassifizieren einen festgestellten Mangel nach der Einschränkung der Nutzung.

In gewissen Fällen verlagert man die Abnahme auf die Zeit nach dem Produktivstart. Dies birgt allerdings die Gefahr, dass bei einer wiederholt scheiternden Abnahme die Möglichkeit zur Rückabwicklung des Projektes massiv eingeschränkt wird. Außerdem sehen gewisse Vertragskonzepte vor, dass mit der Inbetriebnahme des Produktsystems automatisch eine Abnahme erfolgt sei. Vor einem solchen Schritt muss also der Vertrag nochmals gut studiert werden.

Die Abnahme hat rechtliche Folgen betreffend den Beginn einer Garantie- bzw. Gewährleistungsfrist. Häufig sind auch Zahlungsmodalitäten an eine erfolgreiche Abnahme geknüpft – z.B. erfolgen Zahlungen für die Software-Lizenzen der Benutzer idealerweise erst nach erfolgter Abnahme.

Nach der formellen Abnahme im Sinne einer unterzeichneten Abnahmebestätigung beginnt in den meisten Fällen eine so genannte Garantie- bzw. Gewährleistungsphase. Diese stellt sicher, dass nach der Abnahme entdeckte Fehler in der Regel während eines Jahres durch den Lieferanten auf dessen eigene Kosten behoben werden. Die Gewährleistung bezieht sich allerdings nur auf Fehler und gilt keinesfalls für einen funktional ungenügenden Umfang. Ein ungenügender Funktionsumfang hätte im Rahmen des Abnahmeprozederes aufgedeckt oder erkannt werden müssen.

### 16.5.3 Morphologische Matrix für die Inbetriebnahme

Im Schritt der Abnahme und Inbetriebnahme haben sich wieder eine ganze Reihe von konzeptionellen Entscheidungen gestellt. Die Zusammenfassung dieser einzelnen Entscheidungen könnte wiederum als Gesamtvariante zur Inbetriebnahme bezeichnet werden.

Entsprechend den vorangegangenen Kapiteln soll daher auch an dieser Stelle der Lösungsraum im Sinne einer morphologischen Matrix aufgespannt werden (s. Abb. 213).

An erster Stelle steht eine Entscheidung hinsichtlich des Umfangs der Inbetriebnahme, mit den Ausprägungen eines absoluten Big Bangs bis hin zur begrenzten Piloteneinführung in einer ausgewählten Organisationseinheit und einem begrenzten Funktionsumfang. Weiter stellten sich Fragen hinsichtlich des Einsatzes temporärer Systeme und Schnittstellen.

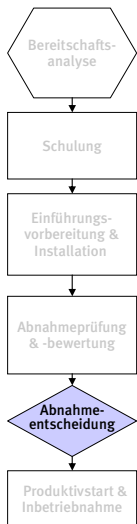
Es muss sodann entschieden werden, welche Daten etwa aus einem bestehenden System in das neue System zu übernehmen sind und welche Daten in diesem Zug einer Datenbereinigung zu unterziehen wären.

<b>Ausprägung</b> <b>Parameter</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>	<b>Ausprägung</b>
<b>Umfang Inbetriebnahme</b>	<b>Big Bang – alle Module, Bereiche</b>	Alle Bereiche, nur gewisse Module	Alle Module, nur gewisse Bereiche	Gewisse Bereiche Gewisse Module
<b>Temporäres Backup-System</b>	<b>Stichtagsumstellung</b>	Parallelbetrieb Alt- & Neusystem	Altsystem als Notlösung bereit	Kein Altsystem vorhanden
<b>Temporäre Schnittstellen</b>	<b>Keine temp. Schnittstelle</b>	Gewisse temp. Schnittstellen	Umfangreiche Schnittstellen	...
<b>Datenmigration</b>	<b>Stammdaten</b>	Stammdaten & Bewegungsdaten	Nur Bewegungsdaten	Keine Datenübernahme
<b>Datenbereinigung</b>	<b>Manuelle Bereinigung</b>	Automatische Bereinigung	Keine Bereinigung	...
<b>Abnahme</b>	Abnahme vor Produktivstart	<b>Teilabnahme vor Produktivstart</b>	Endabnahme nach Produktivstart	...
<b>Produktivstart</b>	keine Prozedur notwendig	<b>automatische Migration</b>	manuelle Endbereinigung	Keine Migration
..	..	..	...	...

Abb. 213: Morphologie zur Inbetriebnahme (Rollout)



## 16.6

**Abnahmeentscheidung**

Die Abnahmeentscheidung ist vielleicht die wichtigste Entscheidung im Rahmen des Projektes, denn sie wird im Gegensatz zu anderen Entscheidungen nicht mehr reversibel oder korrigierbar sein. Wenn der Kunde dem (internen oder externen) Lieferanten formell bestätigt, dass das Informationssystem seinen Anforderungen genügt, so ist der Projektauftrag erfüllt.

Diese Entscheidung wird daher auch nie vorschnell gefällt, sondern basiert auf gesicherten Entscheidungsgrundlagen. Oftmals erfolgt nur eine bedingte Abnahme, welche die Vorbehalte zur Abnahme anmerkt.

Bestandteil der Abnahmeentscheidung ist das Abnahmeprotokoll, welches darlegt, auf welcher Basis der Antrag auf die Abnahme erfolgt. Die entsprechende Grundlage kann wie folgt strukturiert werden:

**I ZUSAMMENFASSUNG****II ÜBERSICHT DER ABNAHMEGEBIETE****III PRÜFPROZEDUREN FÜR ANWENDUNGSGEBIETE 1 – N**

- Ausgangslage und Datenbasis für die Abnahme  
Konzeptionelle Grundlage, Stärken & Schwächen.
- Prüffälle und Prüfprozeduren als Checklisten  
Exakte Beschreibung der vorzunehmenden Prüfprozeduren.
- Zu prüfende Funktionalitäten  
Funktionsliste aller zu überprüfenden Funktionalitäten.
- Zu prüfende Objekte und Attribute
- Zu prüfende Druckerzeugnisse und Reports
- Vorzunehmende Stress- und Belastungstests

**IV ABNAHMEPROTOKOLL**

- Protokollierung der vorgefundenen Datenbasis
- Protokollierung der Anwendungsgebiete entsprechend Checkliste  
Unterteilung der vorgefundenen Probleme nach Problemklassen.

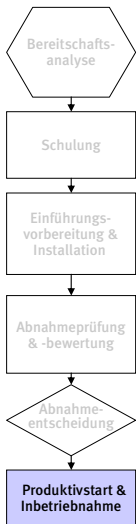
**V ABNAHMEBEWERTUNG**

- Zusammenfassung der Probleme nach Problemklassen
- Entscheidung bezüglich Abnahme bzw. Nicht-Abnahme

**V ABNAHMEVORBEHALTE****VI FORMELLE ABNAHMEBESTÄTIGUNG**

- Abnahme unter Verweis auf etwaige Vorbehalte
- Ort und Datum
- Unterschrift der Verantwortlichen

## 16.7

**Produktivstart und Inbetriebnahme**

Im Anschluss an die Abnahmeentscheidung erfolgen der effektive Produktivstart und die Inbetriebnahme des neuen Informationssystems (engl.: “Go live”). Gelegentlich erfolgt der Produktivstart auch vor der Abnahme. In diesen Fällen müsste allerdings geprüft werden, dass im Vertrag der Produktivstart nicht implizit als stillschweigende Abnahme gilt. Aus Sicht des Anbieters ist damit die Entscheidung zur Nutzung gefallen.

Der Schritt des Produktivstarts und der Inbetriebnahme wurde entsprechend den oben stehenden Ausführungen sehr minutiös geplant. Ein etwaiges Notfallszenario kann zusätzliche Sicherheit bieten, damit auch bei unerwarteten Ereignissen kein Systemunterbruch eintritt. Die detaillierten Inbetriebnahmeroutinen haben insbesondere auch festgehalten, wie und in welcher Abfolge die Datenmigration zu erfolgen hat. Bei manuellen Nachbesserungen an gewissen Daten nach dem Produktivstart muss auch hierfür nochmals genügend Zeit eingeplant werden.

Alle Vorkommnisse im Zusammenhang mit der Einführung werden in einem “Einführungsprotokoll” vermerkt.

Bereich Schritt	<b>Sozio-technisches System</b>	<b>Einführung &amp; Ausbildung</b>	<b>Betrieb &amp; Unterhalt</b>	<b>Projektmanagement &amp; Controlling</b>
<b>Bereitschafts-analyse</b>	Produktivstart Checks; Organisatorische Bereitschaft; Technische Bereitschaft; Letzte Datenübernahmetests; Identifikation von Nacharbeiten nach Produktivstart.	Bereitschaft Ausbildungsstand; Produkktivsystem installiert; Notfallszenarios; Schulungsunterlagen; Information über Schulung.	Bereitschaft Betrieb & Unterhalt.	Feinplanung Inbetriebnahme; Projektmanagement; Projektcontrolling; Information über Einführungszeitpunkt; Einführungsentscheid.
<b>Schulung</b>		Ausgebildete Benutzer.		
<b>Einführungs-vorbereitung &amp; Installation</b>	Vorbereitung aller Einführungsaktivitäten.	Installiertes Produktivsystem; Übernommene Stammdaten; Letzte Mängel behoben; Organisatorische Vorkehrungen für Umstellung.	Betrieb des Produktivsystems; Aktivierung Unterhaltsbetrieb.	
<b>Abnahmeprüfung &amp; -bewertung</b>	Abnahmeprotokoll; Klassifizierte Mängelliste.			
<b>Abnahme-entscheidung</b>				Abnahme-entscheidung.
<b>Produktivstart &amp; Inbetriebnahme</b>	Systemaktivierung; Umgestellte Organisation.	Übernommene Bewegungsdaten.	Support & Tuning; Start Wartung.	Einführungsentscheid Abschlussbericht.

Abb. 214: Ergebnisübersicht der Abnahme- und Inbetriebnahmetätigkeit

## **16.8      Literatur zu Abnahme und Inbetriebnahme**

Schmitz, R. (1998): Einführung von PPS-Systemen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 327–375.



# 17

## Betrieb und Unterhalt

### Kapitelinhalt

- Definition der Betriebs- und Unterhaltsziele
- Analyse und Konzept der Betriebs- und Unterhaltsprozesse
- Organisation und Verantwortlichkeiten für Betrieb und Unterhalt
- Dokumentation

Welche Faktoren sind für einen erfolgreichen Betrieb und Unterhalt der projektierten Lösung Anwendung verantwortlich?

Wie in den früheren Kapitel wiederholt angeführt wurde, gehört es zu jeder Lebensphase eines IT-Projektes, dass man sich weitergehende Überlegungen macht, wie der spätere Betrieb und Unterhalt auszusehen hat und insbesondere welche Stellen welche Verantwortlichkeiten übernehmen müssen. Es kann sein, dass ganz unterschiedliche Vorstellungen über den Betrieb der Lösung herrschen.

Es ist daran zu erinnern, dass das beste Techniksystem nichts nützt, wenn nicht auch der Betrieb und Unterhalt der Lösung sichergestellt sind.

Auch hier werden wir uns wieder auf die bewährte Analyse des Gesamtsystems “Betrieb und Unterhalt” konzentrieren und die Betriebsprozesse, Betriebsfunktionen, Betriebsobjekte (IT-Services) und die Betriebsorganisation beleuchten.

## 17.1 **Betrieb und Unterhalt**

Dem Betrieb und Unterhalt einer Informatiklösung kommt neben der operativen Bedeutung auch kostenmäßig ein hoher Stellenwert zu. Die kumulierten Kosten für Betrieb und Unterhalt einer Informatiklösung übersteigen die gesamten Projekt- und Implementationskosten in der Regel um ein Mehrfaches in der Größenordnung von ungefähr 80/20 (Balzert 1996). Vor diesem Hintergrund wird die Bedeutung des Betriebs- und Unterhaltskonzeptes auch aus wirtschaftlicher Sicht klar.

Durch den stetig gestiegenen Informatik-Kostenblock und den in vielen Bereichen äußerst geschäftskritischen Softwareanwendungen kommt der Informatik mittlerweile generell ein sehr hoher Stellenwert im Unternehmen zu. Mit dieser Einsicht verbunden ist die Professionalisierung des IT-Managements in den vergangenen Jahren.

Ein professionelles IT-Managementsystem definiert zunächst ein Set von Standardprozessen, welche zum erfolgreichen Management bzw. zum Betrieb und Unterhalt der IT notwendig sind.

Neben einem erfolgreichen Betrieb und Unterhalt sind eine Reihe von Voraussetzungen zu berücksichtigen, welche die übergeordnete Eingliederung der Informatiklösung beabsichtigen.

### **Strategische Ausrichtung der IT**

Der Betrieb und Unterhalt der IT beschränkt sich nicht auf den operativen Betrieb. Die Informatik eines Unternehmens muss in ihrer Gesamtheit mit Hilfe einer "strategischen Informatikplanung" ausgerichtet werden. Diese dient dazu, die IT mit den Unternehmenszielen in Übereinstimmung zu bringen. Die strategische Ausrichtung der IT definiert weiter ein Set von wichtigen technischen Systemarchitekturen im Sinne der Plattformen.

### **Portfoliomanagement der IT**

Die effektive Gesamtentwicklung der Informatik wird durch das IT-Portfoliomanagement sichergestellt. Dieses garantiert, dass das Portfolio der Informatikvorhaben, -projekte und -anwendungen zielgerichtet und konsequent gepflegt wird und Investitionsentscheidungen basierend auf fundierten Grundlagen gefällt werden. Daher nehmen die im Kapitel "Projektantrag" erwähnten Projektportfolios eine wichtige Rolle ein. Sie verschaffen einen Überblick über die wichtigsten Projekte.

### **Risikomanagement**

Das Risikomanagement und die IT-Sicherheit stellen die Standardprozesse zur stabilen Geschäftsabwicklung und Disaster Recovery sicher. Es soll verhindert werden, dass die Leistungserbringung des Unternehmens durch unangenehme "Überraschungen" der IT behindert wird.

## 17.2 Prozessmodell Betrieb und Unterhalt

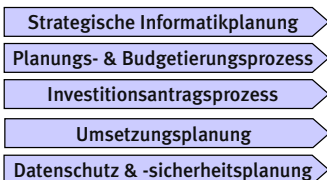
Verbunden mit der oben erwähnten Professionalisierung des IT-Managements entstanden in den vergangenen Jahren verschiedene Prozessmodelle zum IT-Management (z.B. "ITIL – IT-Infrastructure Library"). IT-Prozessmodelle verfolgen das Ziel, eine geordnete Führung des Betriebs und optimale Abläufe zum Betrieb und Unterhalt von Anwendungen sicherzustellen. Sie beschreiben das IT-Betriebs- und Unterhaltssystem unter dem prozessorientierten Aspekt. Durch das Prozessmanagement soll die laufende Qualität sichergestellt und permanent verbessert werden (Vogt 2000). Die schweizerische Bundesverwaltung hat ihre IT-Organisation aus diesem Grund vollständig auf eine prozessorientierte Sichtweise ausgerichtet (NOVE-IT 2003).

Die nachfolgende Abbildung (s. Abb. 215) zeigt ein Beispiel, wie eine IT-Prozesslandkarte dargestellt werden kann. Auf der linken Seite sind zwei Gruppen angeführt, welche die Führung des IT-Managements betreffen. Auf der rechten Seite sind die operationellen Betriebs-, Unterhalts- und Supportprozesse vermerkt. In Zusammenhang mit der Erstellung eines Betriebs- und Unterhaltskonzepts wäre im nächsten Schritt zu definieren, wie die neue Anwendung hinsichtlich der Betriebs- und Supportprozesse im Detail einzeln abzuwickeln wären. Das "Stellenorientierte Ablaufdiagramm" könnte beispielsweise visualisieren, wie der Supportprozess zu verlaufen hat, indem die Verantwortlichkeiten für First- und Secondlevel dargestellt werden.

### Führungsprozesse



### Planungsprozesse



### Betriebsprozesse



### Supportprozesse

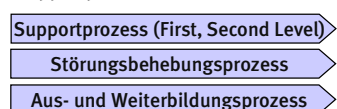


Abb. 215: Betriebs- und Unterhaltsprozesse

### 17.3 Funktionsmodell Betrieb und Unterhalt

Das nachfolgende Funktionsmodell zeigt einen möglichen Ansatz, wie das System des Betriebs und Unterhalts aus funktionaler Sicht dargestellt werden könnte (s. Abb. 216). In der Regel wird man sich allerdings alternativ für ein Prozessmodell oder ein Funktionsmodell entscheiden. In kleineren Unternehmen wird vermutlich häufig die funktionsorientierte Darstellung gewählt werden, da der Aufwand für eine Prozessdokumentation als zu hoch erscheint. Größere Unternehmen bevorzugen zunehmend die prozessorientierte Dokumentationen.

Das unten dargestellte Funktionsmodell des Betriebs und Unterhalts sieht eine Trennung der benutzungsnahen und systemnahen Funktionen vor. Diese werden durch Führungs- und Basisfunktionen ergänzt. Im Rahmen eines zu erstellenden Betriebs- und Unterhaltskonzepts wäre nun beispielsweise zusätzlich ein "Stellenfunktionendiagramm" zu erstellen, welches die Funktionen den IT-Organisationseinheiten mit den jeweiligen Verantwortlichkeiten zuweist. Unter Umständen müssten die Betriebs- und Unterhaltsfunktionen weiter detailliert aufgelistet werden, um primär die Abgrenzung der Verantwortlichkeiten im Unternehmen zwischen der zentralen IT-Organisation und den dezentralen anwendenden Organisationseinheiten herauszustellen.

Auf die Problematik dieser Abgrenzung wird weiter unten im Abschnitt zur IT-Organisation näher eingegangen.

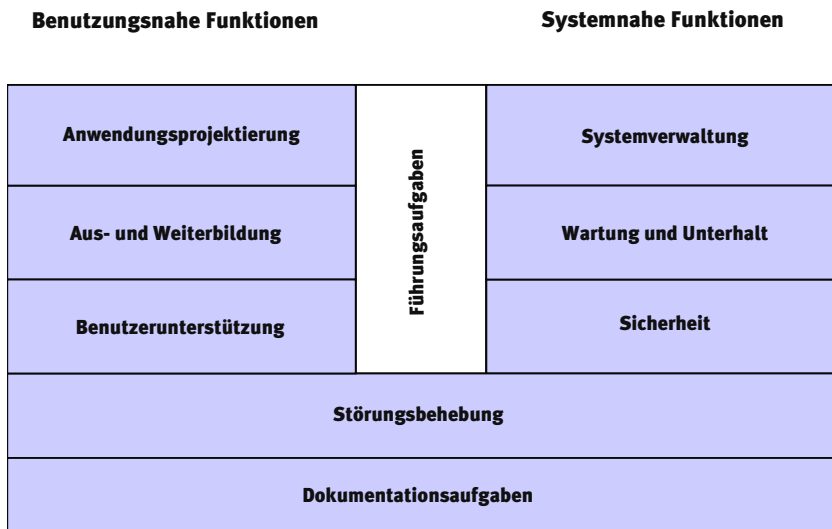


Abb. 216: Funktionsmodell Betrieb und Unterhalt



## 17.4 IT-Produkte

Entsprechend dem in Teil 1 propagierten Systemansatz stellt sich nach der Prozess- und Funktionsbetrachtung in einem nächsten Schritt die Frage, welches die Bearbeitungsobjekte des IT-Betriebs- und Unterhaltssystems wären. Bei dieser Frage interessieren uns zunächst die IT-Dienstleistungen (Services) und deren Weiterverrechnung im Unternehmen (s. Abb. 217).

Im Rahmen des zu erstellenden Betriebs- und Unterhaltskonzeptes für eine neue Anwendung müssen zunehmend auch Fragestellungen im Bereich der Verrechnung von IT-Services mitbehandelt werden. Denn mit zunehmenden IT-Kosten erlangt auch die Verrechnung dieser Services gegenüber den internen Kunden an Bedeutung. Die Kosten eines größeren Projektes sowie des Betriebs und Unterhalts müssen in irgendeiner Form im Unternehmen transparent verteilt werden. Dies bedingt, dass eine Kostentransparenz mit Hilfe einer Kostenträgerrechnung hergestellt wird. Auf der anderen Seite verlangen Kunden zunehmend eine Transparenz der bezogenen IT-Services.

Es bestehen mitunter ganz unterschiedliche Ansätze, wie sich Services im Unternehmen verrechnen lassen. Das Spektrum reicht von der "pauschalen Verrechnung" bis hin zur "nutzungsbasierten Abrechnung" auf Basis von Messgrößen. Hierfür werden so genannte "Service Level Agreements" (SLA) vereinbart. Diese halten fest, welche Services in welcher Qualität und in welcher Verfügbarkeit bereitgestellt werden müssen.

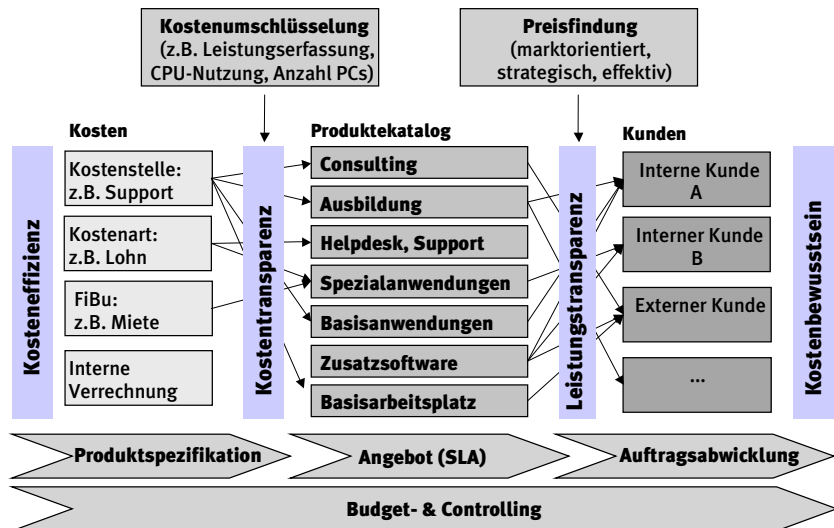


Abb. 217: Kosten- und Leistungstransparenz

## 17.5 Organisation Betrieb und Unterhalt

Im Rahmen des Betriebs- und Unterhaltskonzeptes für eine neue Anwendung muss frühzeitig geklärt werden, wie die Verantwortung für den Betrieb und Unterhalt organisatorisch gestaltet werden soll und ob im Unternehmen diesbezügliche Änderungen vorteilhaft wären. Es ist zu klären, mit welcher IT-Organisation die wachsenden Aufgaben optimal abgedeckt werden können (s. Abb. 218).

Es lassen sich drei Formen der Organisation unterscheiden: Bei der *“zentralen Lösung”* übernimmt eine zentrale IT-Organisation alle Betriebs- und Unterhaltsfunktionen. Die *“dezentrale Lösung”* delegiert die gesamte Verantwortung an die dezentralen Anwendungsbereiche. Bei der *“koordinierten Dezentralisation”* wird eine Kombination aus zentraler und dezentraler Informatik definiert, wobei die Aufgabenteilung funktionsbezogen abzugrenzen ist. Die zentrale Informatik erbringt bei dieser dritten Organisationsform in der Regel gewisse zentrale Basisdienstleistungen und behebt technische anspruchsvollere Probleme (Second Level Support). Bei den Anwendungsbereichen bleibt häufig der Erstsupport für die Anwender in Zusammenhang mit Fragestellungen der Anwendungsbedienung.

Bei der Wahl der IT-Organisation muss darauf geachtet werden, dass diese in Übereinstimmung mit der bestehenden Bereichsautonomie steht. Probleme mit zentralen IT-Organisationen entstehen häufig dann, wenn autonome Bereiche von zentralen Informatik-Abteilungen abhängig sind.

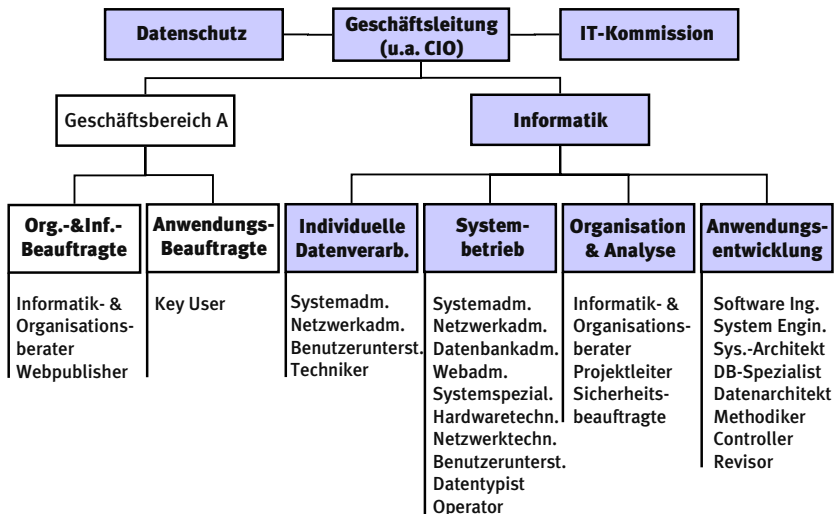


Abb. 218: Beispielhafte Aufbauorganisation der IT

## 17.6

**Morphologische Matrix für Betrieb und Unterhalt**

Die Ausgestaltung von Betrieb und Unterhalt einer Informatikanwendung erfolgt entsprechend den oben genannten Parametern (s. Abb. 219). Es ist dabei zu berücksichtigen, dass die Betriebsorganisation in den meisten Fällen nicht nur auf die neue Anwendung ausgerichtet werden kann. In der Regel bestehen diverse IT-Anwendungen in einem Unternehmen, welche es gemeinsam optimiert zu unterhalten gilt. Allenfalls wird man bei der Ablösung einer integrierten Gesamtanwendung die umfassende Reorganisation der IT-Verantwortlichkeiten in Betracht ziehen (z.B. ERP-System).

An erster Stelle steht die Frage, wie die Verantwortung für den zukünftigen Betrieb und Unterhalt der Applikation ausgestaltet werden sollen. Wird die ganze Verantwortung zentral wahrgenommen, oder sind die dezentralen Anwendungsbereiche beim Betrieb und Unterhalt mitbeteiligt?

Eine zentrale Informatikorganisation kann in verschiedenen Autonomiestufen geführt werden: als Aufwandskonto, Cost-Center, Profit-Center oder gar im Sinne eines internen Service Providers im Unternehmen.

Es ist weiter zu entscheiden, ob und wie die neue Anwendung gegenüber den Leistungsbeziehern zu verrechnen ist. Wenn eine interne Verrechnung erfolgen soll, so müssen auch die Anwendungskosten erhoben werden. In diesem Fall wäre auch zu klären, welche IT-Services zu definieren sind und welchen Umfang und Struktur der IT-Servicekatalog annehmen sollte.

<b>Ausprägung</b> <b>Parameter</b>	<b>Ausprägung 1</b>	<b>Ausprägung 2</b>	<b>Ausprägung 3</b>	<b>Ausprägung ...</b>
<b>Organisatorische Einbindung</b>	<b>Zentrale Informatik</b>	Dezentrale Informatikstellen	Koordinierte Dezentralisierung	Outgesourcte Informatik
<b>IT-Organisation</b>	Aufwandskonto	<b>Cost-Center</b>	Profit-Center	Enterprise Service Provider
<b>Verrechnung der Leistungen</b>	<b>Keine Verrechnung</b>	High-level-Verrechnung	Servicebasiert	Nutzungsbasiert; SLA-basiert
<b>Kosten-transparenz</b>	<b>Kostenstellen</b>	Kostenarten	Kostenträgerrechnung (Services)	Activity Based Costing
<b>Leistungs-transparenz</b>	<b>Keine definierten Services</b>	Services definiert	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

Abb. 219: Beispielhafte morphologische Matrix Betrieb und Unterhalt

## 17.7 Betriebs- und Unterhaltshandbuch

Das Betriebs- und Unterhaltshandbuch wird stark unternehmensspezifisch ausfallen. Das nachfolgende Beispiel ist nur eine mögliche Struktur. Es nimmt neben den betrieblichen Aspekten auch Aspekte der so genannten IT-Governance auf, welche aus unternehmerischer Sicht den wirtschaftlichen und sicheren Informatikbetrieb sicherzustellen hat.

### **I ZUSAMMENFASSUNG**

### **II ÜBERSICHT BETRIEBS- UND UNTERHALT**

- Zielsetzungen an den Betrieb & Unterhalt
- Aufgaben und Gesamtorganisation

### **III STRATEGISCHE AUSRICHTUNG DER INFORMATIK**

- Strategische Informatikplanung
- Unterstützung der Unternehmensziele durch die Informatik
- Projekt- und Anwendungsportfolio
- Überblick über die wichtigsten Projekte

### **IV SYSTEMARCHITEKTUR**

### **V BETRIEBSPROZESSE UND BETRIEBSFUNKTIONEN**

### **VI SERVICEKATALOG UND SERVICEVEREINBARUNGEN**

### **VII ZENTRALE UND DEZENTRALE BETRIEBSORGANISATION**

### **VIII SICHERHEIT UND DATENSCHUTZ**

### **IX ÜBERWACHUNG BETRIEB**

- Performance Monitoring
- Transparenz der Gesamtinformatik
- Kosteneffizienz
- Leistungstransparenz

### **X RISIKOMANAGEMENT**

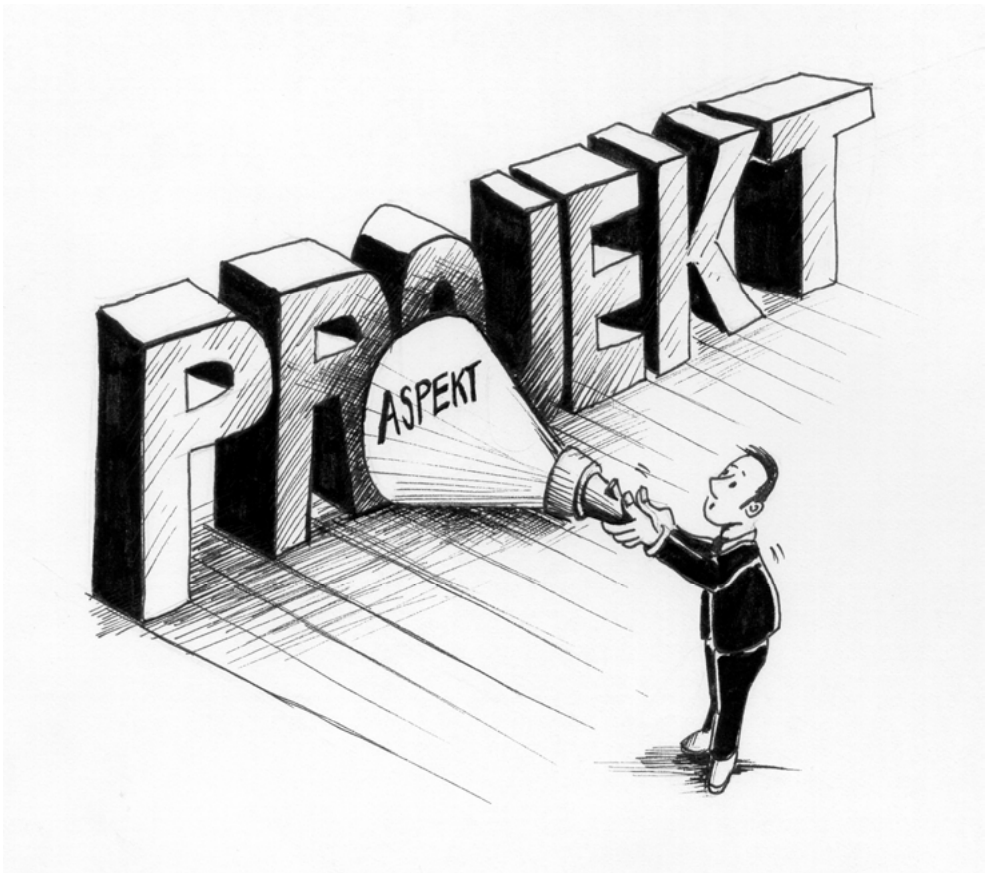
- Risikomanagement
- IT-Sicherheit
- Stabile Prozesse und Disaster Recovery

### **XI ANHÄNGE**

- Systemarchitektur
- Applikationsverzeichnis
- Projektmanagementleitfaden
- etc.

## 17.8 Literatur zu Betrieb und Unterhalt

- Balzert, H. (1996): Lehrbuch der Software-Technik: Software Entwicklung. Heidelberg/Berlin: Spektrum Akademischer Verlag.
- Suter, A. (2002): Informatik-Kostenverrechnung als wesentliches Element der IT-Governance: Aufbau eines Systems für die prozessorientierte interne Verrechnung der Informatikkosten. In: Der Schweizer Treuhänder, 3/02.
- Schönsleben, P, Specker, A. et al. (1999): Unternehmerische Behandlung der Softwareproduktion – mehr als den Softwareentwicklungsprozess im Griff haben. In: io Management Zeitschrift, 6/99. 90–96.
- Sturm, R., Morris, W., Jander, M. (2000): Foundations of Service Level Management. Indiana: SAMS.
- Vogt, W. (2000): Nutzen ohne Frust: IT Service kundenorientiert planen und steuern. Basel: Perseo.
- Hermes (2003): Hermes: Führen und Abwickeln von Projekten der Informations- und Kommunikationstechnik (IKT); Handbuch Systementwicklung. Bern: Informatikstrategieorgan Bund ISB.
- IT Governance Institute (2000): Information Systems Audit and Control Foundation (ISACF): COBIT. 3 Aufl.
- Dunckel, H., Volpert, W., Zölch, M. et al. (1993): Kontrastive Aufgabenanalyse im Büro. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 7. Zürich/Stuttgart: Verlag der Fachvereine/Teubner.
- NOVE-IT (2003): Informatikprozesse der Bundesverwaltung: Definierte Abläufe in der Informatik. Bern: Informatikstrategieorgan Bund ISB.



# 18

## Projektmanagement

### Kapitelinhalt

- Definition des Projektmanagements und der Projektorganisation
- Planung des Projektes: Inhalts-, Termin- und Ressourcenplanung
- Laufendes Projektmanagement: Aufträge, Änderungen, etc.
- Periodisches Projektcontrolling (Status des Projektes)

Dieses Kapitel geht der Frage nach, was gutes Projektmanagement kennzeichnet und in welcher Form es die Phasen von Informatikprojekten zu begleiten hat. Die allgemeinen Grundlagen des Projektmanagements gelten unabhängig vom gewählten Vorgehensmodell.

Es wird aufgezeigt, dass das Projekt und das Projektmanagement idealerweise selbst als System aufgefasst und behandelt werden kann. Dieses System ist nicht identisch mit dem zu gestaltenden Informationssystem. Elemente des Projektsystems sind: Projektorganisation, Projektmanagementprozess, Projektaufgaben, etc. Wie jedes System verfügt dieses "Projektmanagementsystem" also über Prozesse, Projektmanagementfunktionen, Objekte und eine Projektmanagementorganisation.

Es wird aufgezeigt, welche Faktoren für erfolgreiche IT-Projekte verantwortlich sind. Studien belegen, dass es sich hierbei mehrheitlich um Faktoren handelt, welche dem allgemeinen Projektmanagement zugeordnet werden können und nichts Informatik-spezifisches in sich tragen.

## 18.1 Inhalt und Vorgehen des Projektmanagements

Der Inhalt des Projektmanagements besteht darin, ein Projekt zu planen, abzuwickeln und zu überwachen. Dabei sollten der Projektprozess und der Projektmanagementprozess klar unterschieden werden. Der Projektprozess entspricht dem oben beschriebenen Vorgehensmodell mit seinen Projektphasen gemäß Wasserfall- oder Spiralmodell oder einer Kombination dieser beiden Vorgehensmodelle. Der Projektmanagementprozess übernimmt eine übergelagerte Projektführungsfunktion. Im Vordergrund stehen die Projektplanung, die Projektmanagementausführung sowie die Projektüberwachung (s. Abb. 220).

Der Ausgangspunkt des Projektmanagements besteht in der Planung des Projektes in Bezug auf Inhalte, Termine und Ressourcen. Hier wird der Grundstein für die spätere Projektmanagementausführung und das Controlling gelegt. Neben Inhalt, Terminen und Kosten muss z.B. auch geplant werden, in welcher Weise Projektänderungen gemanagt werden.

“Projektmanagementausführung” bedeutet hier nicht etwa die inhaltliche Ausführung wie beispielsweise die “Spezifikation” oder “Programmierung”. Es umfasst die Ausführung von Projektmanagementfunktionen.

Das Projektcontrolling ist ein regelmäßiges Projektstatuscontrolling, welches periodisch und am Ende jeder Phase erfolgt. Das Controlling liefert den Status des Projektes hinsichtlich seines Fortschritts. Im Rahmen der Projektmanagementausführung erfolgt das fortlaufende Controlling.

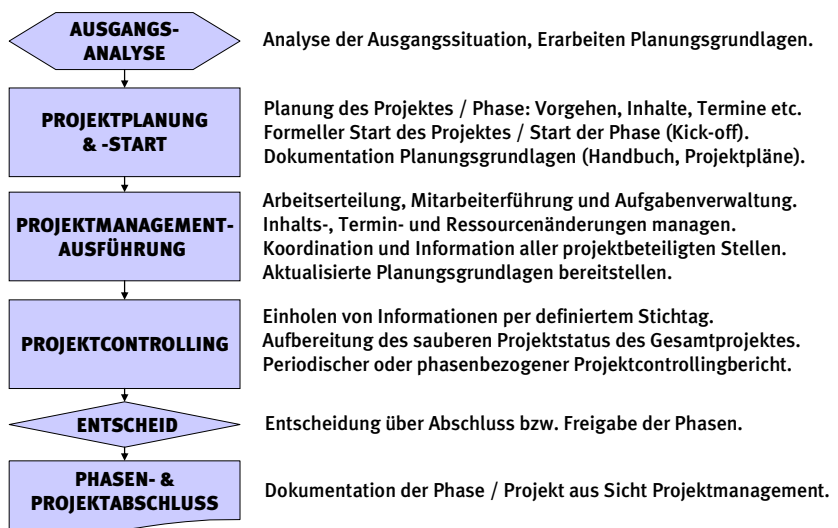


Abb. 220: Vorgehensschritte und Projektmanagementprozess



## 18.2 Projekt

### 18.2.1 Projektdefinition

Projekte werden als zeitlich begrenzte Aufgaben definiert, welche die Lösung eines gegebenen Problems anstreben. Ein Projekt ist durch die Zielerreichung abgeschlossen. Es lässt sich zeitlich abgrenzen und beinhaltet eine gewisse Einmaligkeit.

Projekte grenzt man von “Aufgaben” dahingehend ab, als sie einen größeren Komplexitätsgrad aufweisen, welcher spezifische Instrumente zur Bewältigung erforderlich macht (Systems Engineering & Projektmanagementtechniken).

Die Komplexität von Projekten wird durch das Zusammenspiel diverser Faktoren erzeugt. So gilt es Inhalte, Termine und Ressourcen aufeinander abzustimmen und zu koordinieren. Überdies ist bei einem Projekt in der Regel vorgängig meist nicht wirklich klar, wie die zu erstellende Lösung aussehen wird. Projekte können weiter durch gewisse unvorhersehbare Ereignisse tangiert werden, welche es häufig unter Zeitdruck zu bewältigen gilt. Besonders schwierig ist in diesem Zusammenhang der Umgang mit Änderungen an den Projektzielen und Projekthinhalten (s. u.).

Wie bereits im Systems Engineering angedeutet, werden komplexe Probleme idealerweise methodisch angegangen. Dies sollte daher auch für das Projektmanagement gelten. Für das Projektmanagement gibt es mittlerweile eine ganze Reihe von bewährten methodischen Instrumenten.

Es hilft also auch hier, auf die zahlreichen bestehenden Erfahrungen zurückzugreifen. Im Gegensatz zu vielen allenfalls technischen Unwägbarkeiten eines Projektes lassen sich die Grundzüge des Projektmanagements erlernen und strukturiert behandeln.

Nachfolgend nochmals zusammenfassend die wesentlichen Merkmale von Projekten:

- Einmaligkeit (keine Routinetätigkeit)
- Zeitliche Begrenzung
- Fokussierung auf ein definiertes Ziel
- Beteiligung verschiedener Organisationseinheiten
- Konkurrenz bei Ressourcen
- Komplexität, welche die Bildung von Teilaufgaben erfordert
- Notwendigkeit zur methodischen Behandlung der Komplexität

### 18.2.2 Projektkategorien

Das Projektmanagement sollte in der Lage sein, ganz unterschiedliche Projekte in Bezug auf deren Dimensionierung und Inhalt abzudecken. Um den Unterschieden gerecht zu werden ist es lohnenswert, klar abgegrenzte Projektkategorien zu definieren. In diesem Zusammenhang kann dann für jede Projektkategorie ein ganz spezifisch definiertes Set von Projektmanagementfunktionen und -prozessen zur Unterstützung vorgegeben werden. Typischerweise werden rund drei Kategorien unterschieden:

**Kategorie A:**            **sehr komplexe Projekte**

**Kategorie B:**           **komplexe Projekte**

**Kategorie C:**           **einfache Projekte**

Die Zuordnung eines Projektes zu einer Projektkategorie kann von einer ganzen Reihe von (Komplexitäts-)Faktoren abhängig gemacht werden. Es können dies sein: z.B. Projektumfang (Kosten), organisatorische Komplexität, Komplexität des Kundenproblems, bestehendes Know-how, Technologiekomplexität, kulturelle Aspekte (Sprachen), Standorte, etc.

Die Einteilung in Projektkategorien verfolgt das Ziel, dass mittel und wenig komplexe Projekte nicht durch unnötige administrative Vorschriften überfrachtet werden.

Allerdings gilt es zu bedenken, dass auch bei kleinen und wenig komplexen Projekten ebenfalls alle Projektmanagementfunktionen wahrzunehmen wären. Denn auch hier gibt es Aufträge, unerledigte Aufgaben, Termine, Ressourcen, Risiken etc.

Man könnte zusätzlich argumentieren, dass Kleinprojekte inhaltlich automatisch weniger umfassend seien und daher bezüglich der Ausführung der Projektfunktionen auch keine Abstriche gemacht werden müssen. So wäre z.B. die Liste der Projektrisiken kürzer, die Zahl der Änderungsanträge geringer etc.

Es ändert sich primär etwas im Bereich der Institutionalisierung gewisser Aufgaben. So wird bei einem komplexen Großprojekt allenfalls ein eigens dafür eingerichtetes Projektoffice installiert, welches gewisse Aufgaben wahrnimmt, die der Projektleiter oder die Projektleiterin implizit wahrnehmen würde.

Gleiches gilt für das Controlling. In einem großen, komplexen Projekt wird unter Umständen ein externes Projektcontrolling installiert. Bei wenig komplexen Projekten wird man eine solche Dienstleistung sicher nicht beanspruchen, sondern im bestehenden Team verankern.

### 18.3 Projektmanagement

Die Führungslehre betrachtet die Führung von Projekten als besonders schwierig. Dies rührt daher, dass mit dem Grad des Fortschrittes die Aufgaben und die Führungsanforderungen ständig wechseln. Zusätzlich ist wie erwähnt ein Projekt mit Aufgaben konfrontiert, welche eine Einmaligkeit aufweisen und sich oft auf ungeahnte Art und Weise auf das System und die Umsysteme auswirken.

Das Projektmanagement begleitet ein Projekt von dessen Beginn bis zu seinem Abschluss (s. Abb. 221). Dies gilt für das Spiralmodell wie auch gleichermaßen für das Wasserfallmodell. Auch die Phasen des Spiralmodells müssen sauber gemanagt werden.

Die Phase "Projektantrag" wird gelegentlich auch als zum Projektmanagementprozess zugehörig betrachtet. Als Resultat der Tätigkeit (Phase) des "Projektantrags" entsteht ein formeller Projektantrag, in welchem die Grundzüge des Systems (Eigenschaften, Nutzen, ...) und die Grundzüge des Vorgehens (Phasen, Meilensteine etc.) dargelegt sind.

Erst nach einer offiziellen Projektfreigabe können der Projektstart und die Installation eines institutionalisierten Projektmanagements erfolgen. Während der Tätigkeit des "Projektantrags" besteht ja auch noch kein eigentliches Projekt, sondern lediglich eine begrenzte Aufgabe zur Erstellung des Projektantrags.

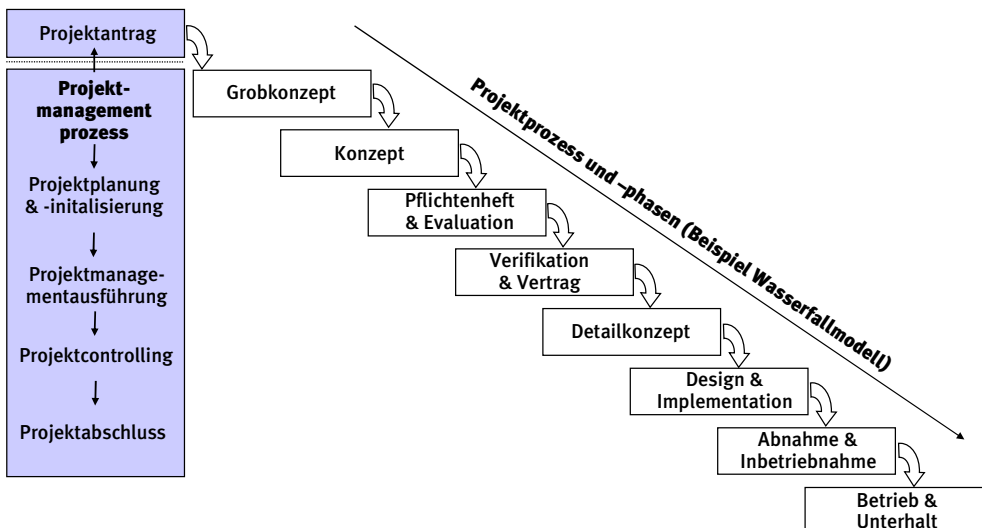


Abb. 221: Projektprozess und Projektmanagementprozess

### 18.3.1 Systemisches Projektmanagement

Auch Projekte und ihr Projektmanagement können als “System” verstanden und aufgefasst werden. Das Projektmanagementsystem darf aber nicht mit dem Gestaltungssystem gleichgesetzt werden. Vielmehr überlappen und durchdringen sich diese beiden Systeme (s. Abb. 222). Man spricht daher auch häufig von “systemischem Projektmanagement”. Das Projekt verfügt über einen Vorgehens- & Projektmanagementprozess, Projektmanagementfunktionen, Objekte und eine Projektorganisation.

Das systemische Projektmanagement empfiehlt, “Projektelemente” zu identifizieren und deren Beziehungen zu analysieren, in Analogie zu der Analyse der Systemelemente und -beziehungen. Weiter soll das Projektumfeld einer so genannten Projektumfeldanalyse unterzogen werden – ebenfalls in analoger Weise zur Analyse des Systemumfelds. Beispielsweise zählen die Fachvorgesetzten des Projektleiters zwingend zum Projektumfeld. Sie müssen aber nicht zwingend zum Systemumfeld zählen.

Auch das Projektsystem kennt wie erwähnt Prozesse (Vorgehensprozesse und Projektmanagementprozesse), Projektmanagementfunktionen, Projektobjekte, Projektaufgaben und Projektmanagementtools, welche nicht mit dem Gestaltungssystem identisch sind. Es beinhaltet die Projektmitarbeiter und deren Rollen und Beziehungen (Machtverhältnisse, Konflikte). Die Analyse dieses Projektsystems deckt Ziele auf und erfasst die “Kultur” des Projektes. Die einzelnen Personen und Rollen innerhalb des Projektsystems haben je eigene Ziele oder Erwartungen.

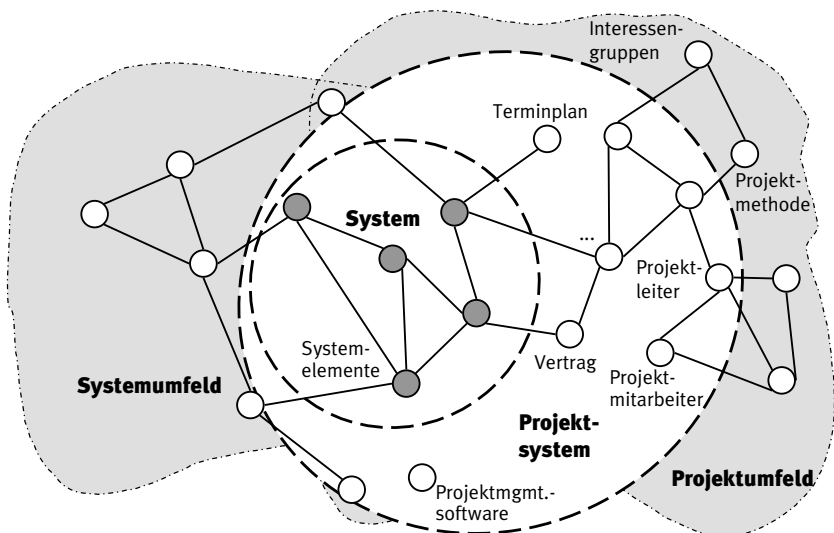


Abb. 222: Analyse des Projektsystems und des Projektumfeldes

### 18.3.2 Erfolgsfaktoren des Projektmanagements

Wir wollen uns nun der Frage zuwenden, welche Faktoren den Erfolg einer Systemeinführung zu beeinflussen vermögen. Interessanterweise verzeichnen Studien, welche Erfolgsfaktoren von IT-Projekten untersucht haben, über weite Strecken Erfolgsfaktoren, die dem allgemeinen Projektmanagement angehören (z.B. Martin 1993). Neben diesen allgemeinen Faktoren treten bestimmte Punkte ganz besonders hervor. Es sind dies die Aspekte "Schulung" und "Handlungsautonomie". Auf diese Aspekte wurde in den oben stehenden Abschnitten bereits vertieft eingegangen. Es versteht sich daher von selbst, dass die korrekte Anwendung der in den vorangegangenen Kapiteln erwähnten Methoden den Einführungserfolg ebenfalls begünstigt. So zeigt es sich ganz besonders, dass eine saubere Evaluation einen maßgeblichen Beitrag zu leisten vermag (a.a.O).

In der Erkenntnis, dass das allgemeine Projektmanagement für den Einführungserfolg von Informationssystemen ganz besonders maßgebend ist, wird nachfolgend eine kurze Zusammenfassung der allgemeinen Erfolgsfaktoren des Projektmanagements präsentiert. Doch welches sind nun die ganz allgemeinen Erfolgsfaktoren von Projekten? Dieser Frage wurde bereits in unzähligen Forschungsvorhaben nachgegangen. Wir stützen unsere Angaben daher nachfolgend auf eine so genannte Metastudie ab (Gemünden 1998). Die Metastudie hat nicht nochmals selber Erfolgsfaktoren von Projekten analysiert. Vielmehr wurden die Ergebnisse von 44 bereits vorliegenden Studien zum Thema Projekterfolg miteinander verglichen und die angegebenen Erfolgsfaktoren verdichtet. Jede der genannten Studien hatte eigene Erfolgsfaktoren extrahiert, und die Metastudie hatte zum Ziel, diese Studien zusammenzufassen. Insgesamt liegen dieser Metastudie somit rund 5 000 Projekte zugrunde.

Betrachten wir die unten stehenden 10 Punkte des Projektmanagements (s. Abb. 223). Ein erster Punkt betrifft die Projektleitung, die sich nicht scheuen darf, Probleme anzusprechen und neben Fachwissen auch über genügend Befugnisse verfügen muss. Die Zielsetzungs- und Planungsfaktoren zeigen auf, dass eine saubere Planung zwar notwendig ist, diese aber nicht zu weit getrieben werden darf. Gleiches gilt für den Instrumenteneinsatz, wie z.B. Projektplanungstools. Auch diese machen streng genommen nur dann Sinn, wenn sie nicht zum Selbstzweck werden. Von ganz wesentlicher Bedeutung sind die Kommunikation und der partizipative Einbezug der Mitarbeiter und des Managements.

Hinzu kommt speziell bei Informatikprojekten die überaus hohe Bedeutung der Schulung der Benutzer. Ihr Gewicht kann nicht genug betont werden, da jeder zusätzliche Schulungstag die Akzeptanz erhöht.

Es besteht sogar die Möglichkeit, Unzulänglichkeiten eines Systems durch gezielte Schulungsmaßnahmen direkt zu korrigieren. Dies ist insbesondere dann von Bedeutung, wenn, aus welchen Gründen auch immer, ein nichtideales System evaluiert bzw. eingeführt werden musste.

#### **Projektleiter**

- Ausreichende Entscheidungs- und Weisungsbefugnisse
- Gesamtverantwortung liegt beim Projektleiter (Ressourcengewalt)
- Probleme frühzeitig erkennen und ansprechen
- Fachwissen und Überblick über Gesamtsystematik

#### **Projektteam**

- Projektteam mit den geeignetsten Mitarbeitern
- Monatliche Projektteamsitzungen

#### **Steuerungsgremium**

- Einsatz eines Steuerungsgremiums

#### **Projektziele**

- Vereinbarung von fordernden, realistischen und messbaren Zielen
- Vermeidung von Zieländerungen

#### **Terminziele**

- Klare Terminziele setzen

#### **Planung**

- Keine zu detaillierte Planung, dafür rollierende Erweiterungen
- Team sollte nicht durch weitere Projekte absorbiert werden
- Aktive und nicht reaktive Steuerung

#### **Instrumente**

- Angepasster Instrumenteneinsatz
- Projektstruktur-, Termin- und Kostenplan stehen im Zentrum
- Pläne müssen aktualisiert werden

#### **Partizipation**

- Beteiligung des Projektteams an den Projektentscheidungen
- Umsetzung eines kooperativen Projektführungsstils

#### **Kommunikation**

- Planung und Implementierung eines Informationswesens
- Information über die Gründe und Ziele der Einführung
- Information über die gesamte Vorgehensweise bei der Einführung
- Informieren, welche Maßnahmen die Betroffenen erwarten
- Verständliche Informationen
- Förderung informeller Kommunikation

#### **Einbeziehung des Managements**

- Geschäftsleitung muss einbezogen werden
- Engagement durch Projektmarketing fördern
- Keine direkte operative Rolle des Managements

Abb. 223: Schlüsselfaktoren des Projekterfolgs (Gemünden 1998)

## 18.4 Projektmanagementfunktionen

Das Projektmanagement hat eine Reihe von Projektmanagementfunktionen abzudecken (auf den Prozess werden wir später zu sprechen kommen). Es erweist sich als sinnvoll, alle wahrzunehmenden Funktionen in einem entsprechenden Funktionsmodell zu gruppieren (s. Abb. 224). Im Zentrum stehen die Hauptfunktionen des Inhalts-, Termin- & Ressourcenmanagements mit ihren Subfunktionen. Sie werden durch vier weitere Hauptfunktionen umrahmt. Speziell hinzugefügt wurde das Auftragsmanagement, da auch in Projekten Aufträge und zu erledigende Punkte eine zentral wichtige Rolle spielen. In anderen funktionalen PM-Modellen werden häufig gerade diese Funktionen vernachlässigt.

Ein breit abgestütztes und allseits akzeptiertes PM-Funktionsmodell existiert nicht. Viele Firmen haben eigene Projektmanagementmethoden definiert. Internationale Unternehmen setzen teilweise auf das Modell des Project Management Institute (PMI 2000). Dieses geht von 9 Funktionsgruppen aus und kann in den vielen Ländern zertifiziert werden. Diesem Vorteil steht der Nachteil gegenüber, dass es sich eher an Projektmanagementspezialisten richtet und weniger an die Projektmitarbeiter.

In der Schweiz gelangt in der Verwaltung häufig die Projektmethodik "Hermes" zur Anwendung, welches von vier Submodellen ausgeht: Qualitätssicherung, Risikomanagement, Konfigurationsmanagement und Projektmarketing (Hermes 2003).

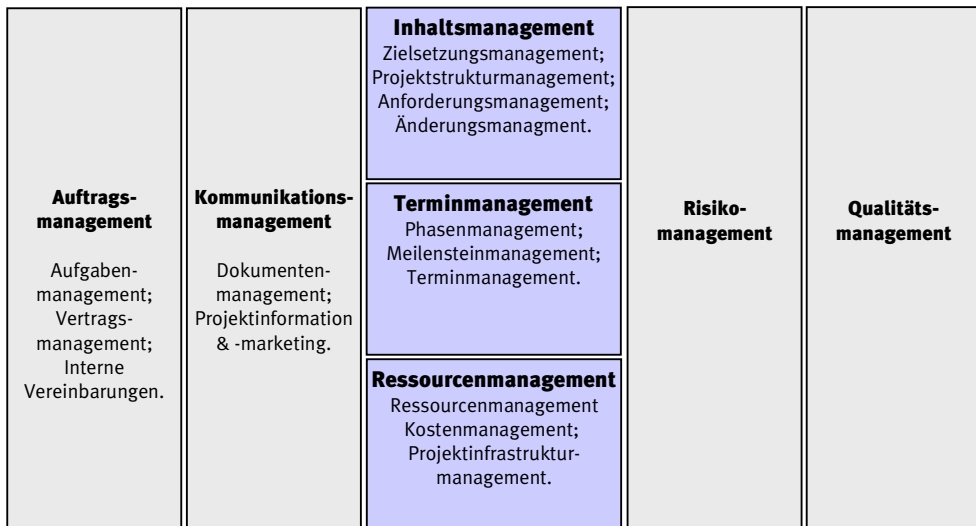


Abb. 224: Funktionsmodell des Projektmanagements

## 18.5 Inhaltsmanagement

### Zielsetzungsmanagement

Die vielleicht wichtigste Funktion des Projektmanagements besteht in der Sicherstellung, dass die angestrebten Zielsetzungen auch tatsächlich inhaltlich erreicht werden. Hierfür müssen die Ziele aber zunächst klar dargelegt und im Projektplan bzw. -handbuch festgehalten werden.

In der Praxis wird oftmals gerade dieser Zielsetzungsprozess stark vernachlässigt. Eine gemeinsame Zielerreichung eines größeren Teams kann indessen nur dann in effizienter Art erreicht werden, wenn alle Projektmitarbeiter exakt dieselben Ziele vor Augen haben.

### Projektstrukturmanagement

Der Inhalt eines Projektes konkretisiert sich in der so genannten Projektstruktur. Projektstrukturpläne definieren auf hierarchische Weise Teilprojekte, Arbeitspakete und die Ergebnisse des Projektes (s. Abb. 225).

Projektstrukturdarstellungen visualisieren u.U. auch inhaltliche Abhängigkeiten. Sie können prozess-, funktions-, objekt- oder aufgabenbezogen erstellt werden. Die Strukturelemente sollten bereits früh mit klaren Budgetwerten geplant und bewertet werden. Die Projektstruktur und die Arbeitspakete bilden den Ausgangspunkt, um im Rahmen des Terminmanagements die zeitlich-logische Abfolge bzw. den Projektablauf und die Terminplanung zu definieren.

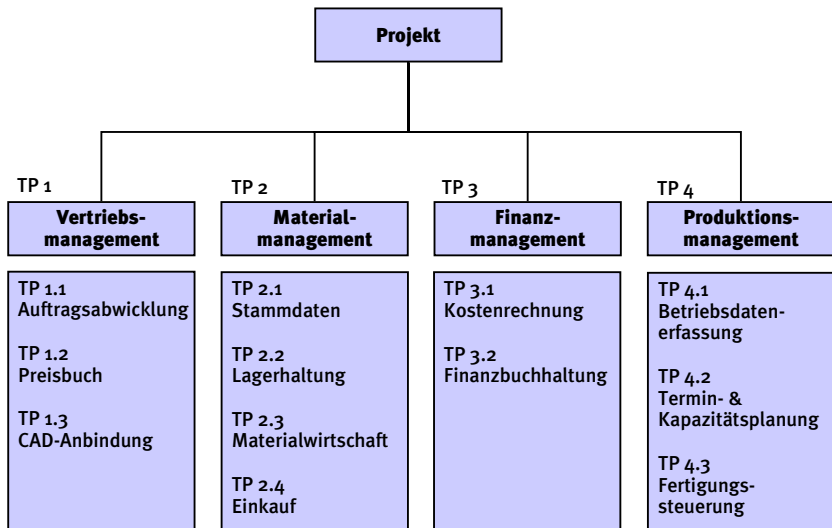


Abb. 225: Projektstrukturplan



### Beispiel: Projektstrukturmanagement von ERP-Projekten

Im Rahmen des Projektstrukturmanagements stellt sich zunächst die Frage, welche Teilprojekte zu bilden sind und in welcher Zahl bzw. thematischen Gruppierung dies zu erfolgen hat. Sodann ist zu klären, wie die Verantwortung für Teilprojekte zugeteilt werden soll.

Im Rahmen einer Expertenbefragung wurde anhand von ERP-Projekten untersucht, welche Sachbereiche sich idealerweise in Teilprojekte abgrenzen lassen oder aber zusammengefasst werden sollten (Schmitz 1998). Das Ergebnis der Befragung zeigt die unten stehende Grafik (s. Abb. 226).

Die angegebenen Prozentzahlen weisen aus, wie viel Experten für die Zusammenfassung der entsprechenden Gebiete plädiert haben. Es ist erkennbar, dass ERP-Projekte gemäß Expertenmeinung im Durchschnitt in vier Teilprojekte unterteilt werden sollten. Eine Mehrheit der Experten spricht sich sodann für die Zusammenfassung von Lagerhaltung, Einkauf und Materialwirtschaft aus.

Einen zweiten Bereich könnte man dementsprechend Produktionsmanagement nennen. Einen dritten, ganz offensichtlich recht eigenständigen Bereich bildet der Vertrieb, wobei die Bereiche Finanzen- und Controlling wieder zu einem Teilprojekt zusammengefasst werden können.

Anzumerken bleibt, und dies zeigen auch die vorliegenden Expertenzahlen, dass keine absolut klaren und übereinstimmenden Trennungen zwischen diesen Bereichen auszumachen sind.

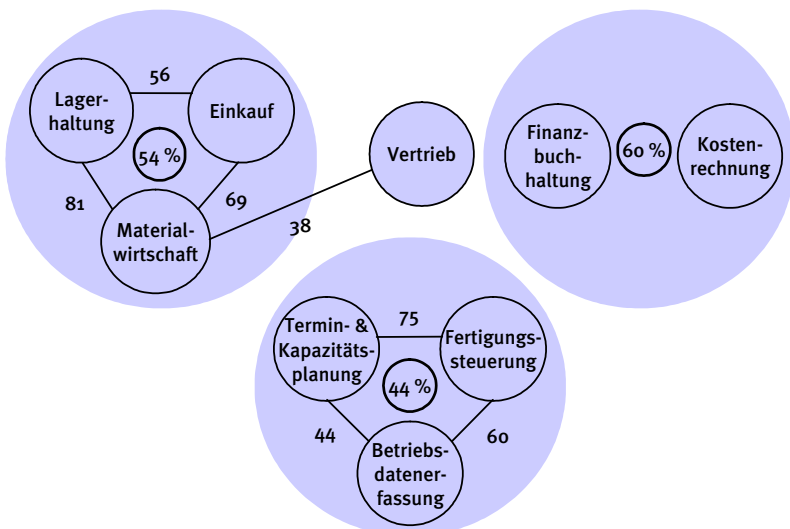


Abb. 226: ERP-Teilprojektbildung aus Sicht von Experten (Schmitz 1998)

### Anforderungsmanagement

Das so genannte Anforderungsmanagement ist verantwortlich für die klar strukturierte Dokumentation und Fortschreibung der Systemanforderungen (s. Kapitel 11). Häufig bestehen mehrere ähnliche und überlappende Darstellungen des geforderten Funktionsumfangs (Pflichtenheft, Angebote, Prospekte etc.). Was genau gilt, ist aber nicht in jedem Fall klar. Die Anforderungen werden daher idealerweise in funktionaler Listenform gegliedert und verbindlich aufgeführt. Nur so kann sichergestellt werden, dass Anforderungen und deren Erfüllung tatsächlich überprüft werden können. Eine besondere Problematik besteht darin, dass sich diese Anforderungen oftmals schon im Verlauf der Pflichtenheft- und Angebotsphase ändern.

### Änderungsmanagement

Das Änderungsmanagement (Change Request, Konfigurationsmanagement) stellt einen strukturierten Prozess für Änderungen von Systemanforderungen dar. In jedem Projekt kommt es zu diversen Änderungswünschen seitens des Kunden bzw. des Lieferanten, welche es unter Kontrolle zu halten gilt (Mehr- oder Minderwünsche). Änderungen gegenüber den ursprünglichen Anforderungen sollten in der Regel nur aufgrund von formellen Meldungen erfolgen und werden in einer Liste geführt (s. Abb. 227). Die Analyse von Änderungsanträgen muss die technischen, ressourcen-, termin- und organisationsbezogenen Auswirkungen auf das Projekt beinhalten. Änderungen mit Kostenfolge haben meist vertragliche Anpassungen zur Folge und bedürfen der Schriftform (s. Vertragsmanagement).

Nr	TP	Antrag durch	Erfass. Datum	Inhalt der Änderung; Auswirkungen; Auswirkungen bei Nichtrealisierung.	Wunschdatum	Priorität	Status	Ergebnis der Analyse; Erforderliche Aktivitäten & Personen	Begutachter	Plandatum	Aufwand & Kosten
1	3.1	Kocher	11.3.06	Kundenhierarchie mit 3 Ebenen, Übersicht	11.4.06	2	akzeptiert	Stellt keine Probleme dar.	Petri 3.4.06	11.6.06	5 MT
2	1.0	Leder	12.3.06	Interface analog Notes im Kalender	15.5.06	1	offen	grössere Änderung.	Heer 15.4.06	15.9.06	25 MT
3	2.1	Kohli	12.3.06	Schnittstelle zu Palm	15.5.06	3	abgelehnt	Stellt keine Priorität dar.	Heer 3.4.06	später	20 MT
4	3.2	Kocher	18.3.06	Adressen mit Umsatz täglich aktualisieren	15.5.06	2	akzeptiert	Im Batch machbar.	Petri 3.4.06	15.7.06	5 MT
5	3.3	Kocher	21.3.06	Visuelle Darstellung der Kampagnenstati.	15.6.06	2	akzeptiert	Stellt keine Probleme dar.	Petri 3.4.06	15.7.06	3 MT
6	3.3	Kocher	26.3.06	Kampagnen mit Zielen kopieren	15.7.06	1	akzeptiert	Stellt keine Probleme dar.	Petri 3.4.06	16.7.06	6 MT

Abb. 227: Beispielformular für Änderungsanträge (Change Request Form)

## 18.6 Terminmanagement

### Phasenmanagement

Das Phasenmanagement beinhaltet die Phasenplanung und Überwachung eines Projektes. Es definiert und plant demnach das übergeordnete Vorgehensmodell mit den entsprechenden Phasen. Diese Planung erfolgt in enger Abstimmung mit dem Projektstrukturplan und den dort definierten Arbeitspaketen.

### Meilensteinmanagement

Ein wesentlicher Punkt des Terminmanagements besteht darin, Meilensteine zu definieren und diese explizit aufeinander abzustimmen. Es hat sich gezeigt, dass eine allzu akribische Terminplanung eines Projektes nicht den erhofften Mehrwert bringt (Gemünden 1998). Allerdings hat es sich bewährt, klare und gut überprüfbare Meilensteine zu setzen und diese Meilensteine in nicht zu großen zeitlichen Abständen festzulegen.

### Terminmanagement

Nachdem das Vorgehensmodell mit seinen Phasen und Meilensteinen definiert wurde, kann in einem nächsten Schritt die zeitliche Planung des Projektes auf Basis von einzelnen Arbeitspaketen vorgenommen werden. Es sind hierzu Zeitbudgets zu definieren und die verschiedenen Teilprojekte und Arbeitspakete in die terminliche Abstimmung zu bringen (s. Abb. 228). Als Ergebnis entsteht der laufend fortgeführte Projektterminplan.

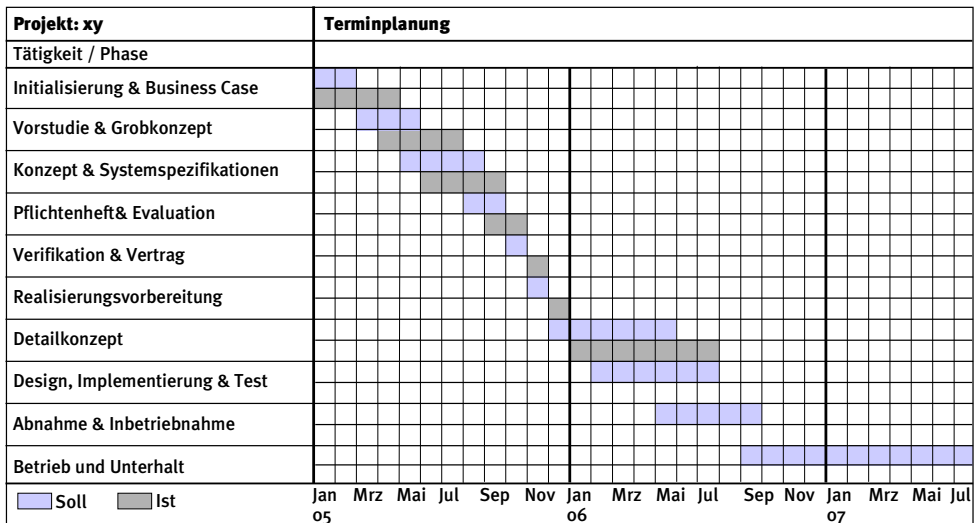


Abb. 228: Beispiel eines Terminplans mit Soll-/Ist-Vergleich

## 18.7 Ressourcenmanagement

### Ressourcenmanagement

In Abstimmung mit der Inhalts- und Terminplanung müssen im Rahmen der Projektplanung die Ressourcen für das Gesamtprojekt sowie für die Teilprojekte geplant und überwacht werden.

Diese Aufgabe umfasst z.B. auch die Planung, Beschaffung und Führung der Mitarbeiter des Projektes. Im gewissen Modellen besteht hierfür eine Funktion "Human Resources".

### Kostenmanagement

Der oben entstandene Terminplan führt in Abstimmung mit den Budgets der einzelnen Arbeitspakete zu einer Plankostenkurve des Projektes. Diese zeigt den geplanten zeitlichen Kostenverlauf auf (daher Plankosten). Die laufenden Kosten des Projektes sind selbstverständlich ständig zu erheben, zu überwachen und auf ihre Rechtfertigung hin zu überprüfen.

Die Kosten für die Einführung von Informationssystemen, z.B. bei ERP-Systemen, betragen im Vergleich zu den reinen Softwarekosten häufig ein Mehrfaches, typischerweise zwischen einem Faktor 1.5 und einem Faktor 5 liegend. Es lohnt sich daher auch immer die Frage, welches im Rahmen des Projektes die größten Ressourcenverbraucher sind. Eine Studie der SAP hat diesen Punkt für ERP-Systeme genauer untersucht (s. Abb. 229) (SAP European Support Group 1998). Den größten Aufwand erzeugen interessanterweise Reports und die Gestaltung von Druckvorlagen.

### Projektinfrastrukturmanagement

Unter das Ressourcenmanagement fällt auch das Management projektbezogener Infrastrukturressourcen. Es kann dies bedeuten, eine genügende Anzahl von Räumen und Arbeitsplätzen für die Teams bereitzustellen.

Erstellung von Reports und Druckvorlagen	(22%)
Schulung der Endanwender	(17%)
Projektmanagement, Diverses	(16%)
Erstellung von Schnittstellen / Datenübernahme	(11%)
Systemkonfiguration	(10%)
Fehlerbehebung	( 6%)
Vergabe von Berechtigungen	( 5%)
Übergang von Test in Produktivphase	( 4%)
Gestaltung der Organisationsstrukturen	( 3%)
Installation R/3	( 2%)
Entscheidungsfindung	( 1%)

Abb. 229: Ressourcenverzehr der Implementation eines SAP-Projektes

## 18.8 Auftragsmanagement

### Aufgabenmanagement

Die vielleicht elementarste Projektmanagementfunktion besteht in der Definition und Übertragung von Aufgaben und dem Management der zu erledigenden Punkte. Die Aufgabenliste sollte von den Projektbeteiligten stets einsehbar sein und nach diversen Kriterien sortiert werden können (z.B. Status, Teilprojekt). In der Projektmanagementliteratur wird die Funktion des Auftragsmanagements leider oftmals nicht explizit angeführt.

### Vertragsmanagement

Das Vertragsmanagement kümmert sich um die Behandlung aller Vertragsaspekte im Rahmen des Projektes. Dies beinhaltet neben erstmaligen Vertragsabschlüssen die Nachführung von Verträgen bei inhaltlichen Änderungswünschen (engl.: "Claims") seitens des Kunden oder Lieferanten.

### Interne Projektvereinbarungen / Contract Management

Der Gedanke des "Vertrags" kann auf interne Teams/Mitarbeiter übertragen werden: Man verpflichtet diese mit internen Projektvereinbarungen (eng.: "Contracts") auf die zu erfüllenden Ziele (s. Abb. 230). Eine Vereinbarung enthält Leistungsumfang, Termine & Ressourcen. Schriftliche Vereinbarungen stärken das Bewusstsein für die Bedeutung einer übertragenen Aufgabe oder eines wichtigen Meilensteins. Sie zwingen die Mitarbeiter zudem, ihre Arbeitskapazität mit Fachvorgesetzten zu diskutieren und abzustimmen.

#### Projekt: Customer Relationship Management

Teilprojekt / Projektphase: Detailkonzeption

Auftraggeber: K. Kundert

Auftragnehmer: L. Landau

#### Gegenstand der Vereinbarung (Spezifikation der zu liefernden Endprodukte)

Erarbeitung einer detaillierten Spezifikation der Adressverwaltung

Beschreibung der notwendigen Managementinformationen

Grafiken zur Darstellung von Umsatz- und Deckungsbeitragsinformationen

Aktivitäten und Ergebnisse :	Auftragnehmer	Aufwand	Termin
Erstellung und Review	Landau	6 PT	31.07.05
Dokumentation	Landau	16 PT	31.09.05
Weitere Reviews, Interne Verabschiedung	Landau / Haas	2 PT	31.12.05
Gesamtaufwand in PT		24 PT	
<b>Abnahme der spezifizierten Endprodukte :</b>			
Zwischenreview:			31.08.05
Abnahmedatum:			31.12.05
Nachbesserung:			01.02.06

Abb. 230: Beispiel einer internen Projektvereinbarung (Contract)

## 18.9 Kommunikationsmanagement

### Dokumentenmanagement

Das Dokumentenmanagement hat sicherzustellen, dass im Projekt stets Klarheit herrscht, wo wichtige Dokumente abgelegt sind und welche Dokumente verbindliche Gültigkeit besitzen. Eine erste Maßnahme besteht darin, alle Dokumente zu versionieren und ihren Ablageort zu definieren (Versionsnummer, Datum, Name, Status, Ablageort). Was für Dokumente gilt, sollte während der Implementationsphase gleichermaßen auch für Softwaremodule und Softwarereleases gelten.

### Projektinformation & -marketing

Unter dem Begriff “Projektinformation & -marketing” werden all jene Maßnahmen zusammengefasst, welche darauf abzielen, Projektbeteiligte und Interessengruppen stets angemessen zu informieren und auf dem Laufenden zu halten (s. Abb. 231).

Durch klare und regelmäßige Informationen wird ein großer Beitrag zum Projekterfolg geleistet. Nur so kann eine breite Unterstützung und Akzeptanz für ein Projekt realisiert werden. Neben den formellen Informationen (Protokolle, Projektstatusreport etc.) sind auch informelle Projektberichte in Firmenzeitschriften, Projektnewsletter etc. in Betracht zu ziehen.

Zielgruppe	Informationsbedarf	Informationsziel	Periodizität	Kommunikationsmedium
<b>Auftraggeber</b>	Konsolidierter Projektstatus	Neuster Stand Projektziele	1 x monatlich	Projektstatusbericht Kurzpräsentation
<b>Projekt-ausschuss</b>	Konsolidierter Status Entscheidungsgrundlagen	aktuelle Planung	1 x monatlich	Projektstatusbericht, PA – Sitzungen
<b>Controller</b>	Projektstatus	aktueller Status	1 x monatlich	Tabellen und Berichte
<b>Kernteam</b>	Vollständige, detaillierte Dokumentation für das Teilprojekt Arbeitspläne	Neuster Stand Projektziele	Wöchentlich Montag	Zweiwöchentliches Reviewmeeting
<b>Teilprojektteam</b>	Dokumentation für das Teilprojekt Arbeitspläne	aktuelle Planung	Gemäss Plan	Reviewmeeting
<b>Partner</b>	Informationen betreffend Schnittstellen	aktuelle Planung	Gemäss Plan	Bericht
<b>Betroffene</b>	Informationen betreffend Auswirkungen auf Arbeit	aktuelle Erkenntnisse	Alle 2 Monate	Infomeeting, Hauszeitschrift
...				

Abb. 231: Beispiel eines Projektinformationsplans

## 18.10 Risikomanagement

Jedes Projekt hat seine besonderen Risiken, welche durch das Projektmanagement überwacht werden müssen (s. Abb. 232). Zur Absicherung des Projekterfolges überprüft man im Rahmen des Risk Managements, wie es um diese Risiken steht und welche Maßnahmen die Risiken vermeiden, vermindern oder auf andere Stellen überwälzen können (Versicherung).

Wichtig ist, dass im Risikokatalog nicht Risiken angeführt werden, welche jedem Projekt zu eigen sind, wie z.B. "Risiko der Kostenüberschreitung", "Risiko Terminverzug". Vielmehr sollten projektspezifische Risiken erhoben werden: z.B. "Risiko, dass Schnittstellenbeschreibung von Lieferant nicht innerhalb Frist vorliegt". Dies bedingt, dass man zunächst die projektrelevanten Risiken identifiziert und dieses dann bezüglich ihrer Eintretenswahrscheinlichkeit bzw. ihrer potentiellen finanziellen Auswirkung – des Schadens – einstuft. Besonders schwierig ist die Abschätzung von Eintretenswahrscheinlichkeit und Schadensauswirkung. Alternativ könnte man sich daher zur Bewertung auch die hypothetische Frage stellen, was man zur Elimination des Risikos zu zahlen bereit wäre. In einem dritten Schritt werden Maßnahmen definiert, welche das Risiko entweder ganz eliminieren, vermindern oder überwälzen. Die Maßnahmen werden terminiert und mit Verantwortlichkeiten versehen. Aus Wahrscheinlichkeit und Kostenfolge errechnet sich für verbleibende Risiken ein Risikowert. Die Summe der Einzelrisiken ergibt den Gesamtprojektrisikowert, welcher sich über die Zeitachse verfolgen lässt.

Risiko Nr.	Risiko erfasst am:	Eintretenswahrscheinlichkeit	Kostenfolge (Euro)	Risikowert (Euro)	Risikobeschreibung	Beschreibung Auswirkung	Massnahmen Minderung, Beseitigung	Zuständigkeiten	Termin für Massnahmen Erledigung
1	12.4.06	30%	150'	45'	Schnittstelle entspricht nicht den Erwartungen	Datenbankintegration als teure Ersatzlösung	Testlauf in 1 Monat gefordert	Voert	...
2	12.4.06	20%	100'	20'	Neuerliche Reorganisation des Vertriebs	Vertriebsstrukturen anzupassen	Information über Auswirkung	Kienast	...
3	12.4.06	5%	400'	20'	Vertrieb wird outgesourct	Einführung allenfalls obsolet	keine Massnahme	Hofmann	...

Abb. 232: Beispiel eines Risikokatalogs

## 18.11 Qualitätsmanagement

Die Qualität in Projekten muss geplant und überwacht werden. Wenn man von Qualität spricht, so sollten zwei Qualitätsaspekte unterschieden werden: Produktqualität und Prozessqualität. Die Produktqualität bezieht sich auf das zu entwickelnde System, d.h. auf die Qualität des Softwareproduktes. Dies entspricht der Qualitätsauffassung im Sinne der Produktqualität zur optimalen Erfüllung der Kundenerwartung ("Qualität des Weins").

Unter Prozessqualität versteht man demgegenüber die Qualität des Projektprozesses (Qualität der Projektabwicklung, Qualität des Projektmanagements etc.). Diese Auffassung entspricht der Philosophie des "Total Quality Managements" bzw. der "Qualität der Schläuche".

Im Rahmen von Projekten sollte neben der inhaltlichen Qualität auch die Qualität des Projektmanagements geplant, überprüft und fortlaufend verbessert werden. Zu diesem Zweck werden frühzeitig Maßnahmen wie ein Qualitätssicherungsplan, Prüfpläne, Reviews, Assessments definiert. Es empfiehlt sich, das Projektmanagement regelmäßig anhand kritischer Erfolgsfaktoren zu überprüfen und ein Rating anzugeben (s. Abb. 233).

Qualitätsberichte rapportieren über die Ergebnisse dieser Reviews und beurteilen, ob überhaupt ein Änderungsverfahren besteht oder ob monatliche Sitzungen stattfinden. Der Controllingbericht rapportiert indessen inhaltlich, z.B. über die Art und Anzahl der Änderungen.

### Qualität Projektleitung und Projektmitarbeiter



**75 %**

#### **Fokussiertes, engagiertes Team mit geeignetsten Mitarbeitern (95%)**

Das Team ist sehr gut besetzt.

#### **Team sollte nicht durch weitere Projekte absorbiert werden (70%)**

Weitere Projekte sind die Regel.

#### **Beteiligung des Projektteams an den Projektentscheidungen (70%)**

Der Faktor Zeit steht gelegentlich im Vordergrund. Entscheidungen werden daher nicht immer vollständig breit abgestützt und relativ adhoc gefällt.

#### **Monatliche Projektteamsitzungen (85%)**

Die Projektteamsitzung wurde nicht immer konsequent eingehalten. Es wurde aus Fehlern gelernt und Koordinationssitzungen haben diesen Mangel behoben.

#### **Einbeziehung der Endbenutzer (65%)**

Die Dokumente könnten teilweise noch besser auf die Benutzer ausgerichtet sein. Eine Einbeziehung ist nicht in allen Belangen gegeben.

Abb. 233: Auszug aus einem Qualitätsreview (Punkt Projektleitung)



## 18.12 Projektmanagementprozess

Ganz entsprechend den allgemeinen Systemmerkmalen verfügt auch das Projektmanagement über einen oder mehrere Projektmanagementprozesse. Der zentrale Projektmanagementprozess beschreibt, in welcher Folge die Projektmanagementtätigkeiten durchgeführt werden (s. Abb. 234). Der Prozess lässt sich typischerweise in vier Teilprozesse unterteilen. Die unten stehende Darstellung zeigt, dass jeder Teilprozess die oben beschriebenen Projektmanagementfunktionen mehr oder weniger stark integriert.

Ein Missverständnis könnte in Bezug auf die Positionierung und Eigenschaft des Projektcontrollings entstehen. Die Darstellung sollte nicht den Eindruck entstehen lassen, dass das Projektcontrolling lediglich einmal kurz vor dem Projekt- oder Phasenabschluss wahrzunehmen wäre. Ein explizites, standardmäßiges Projektcontrolling im Sinne eines "Statuscontrollings" muss mehrfach wiederkehrend gemäß einem vorgängig festgelegten Controllingterminplan (periodisch und/oder phasenbezogen) vorgenommen werden.

Zusätzlich wird natürlich auch im Rahmen der Projektmanagementausführung durch die Projektleitung bzw. das Projektcontrolling ein ständiges und laufendes Controlling erfolgen müssen. Diese steht hier in klarer Abgrenzung zum stichtagsbezogenen Statuscontrolling.

Die einzelnen Teilprozessschritte werden in den nächsten Abschnitten einzeln ausgeführt.

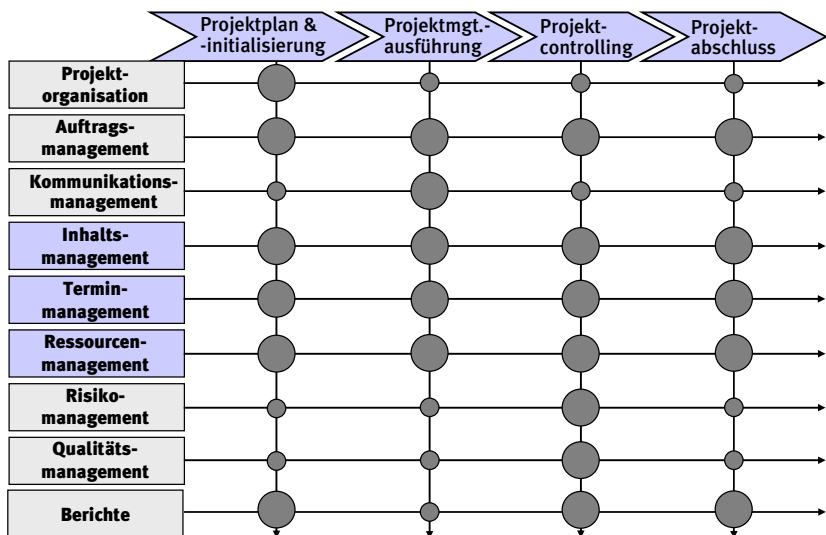


Abb. 234: Projektmanagement-Prozess mit Schwerpunkten

### 18.12.1 Projektplan und -initialisierung

Jedes Projekt bedarf zunächst einer sauberen Planung und eines guten Starts. Zu Beginn muss geklärt werden, wer die Verantwortung für die Projektrealisierung zu tragen hat. Die Prozeduren wie Berichtswesen, Dokumentationspflichten, etc. müssen zusätzlich definiert werden (s. Abb. 235).

Im Rahmen der Projektplanung müssen die aufbau- und ablauforganisatorischen Voraussetzungen in Bezug auf das Projektmanagement nochmals überprüft werden. Es gilt, die Projektteams und die Verantwortlichkeiten sowohl auf Seiten der Anwenderfirma als auch auf Seiten des (internen bzw. externen) Lieferanten klar zu definieren.

Die Prozeduren wie Berichtswesen, Dokumentationspflichten, etc. müssen zusätzlich definiert werden (s. Abb. 235). In der Regel erfolgt die Dokumentation der Projektteamsitzungen und Projektausschusssitzungen durch die Kundenfirma, die Dokumentation von Designworkshops zur Systemspezifikation wird demgegenüber vom Systemanbieter wahrgenommen.

- Projektorganisation und Projektvereinbarungen definieren und verabschieden (Projektausschuss und -team)
- Dokumentation und Protokollierung festlegen (Welche Sitzungen werden von wem protokolliert)
- Auftragsverwaltung institutionalisieren (wer führt diese Liste)
- Projektstrukturplan, Ergebnisplanung und Änderungsverfahren
- Terminplanung und Meilensteine festsetzen
- Budget- und Kostenplanung
- Risiko- und Qualitätsmanagement definieren
- Terminplanung Projektteamsitzungen (in der Regel einmal im Monat – mit Protokoll)
- Terminplanung Projektausschusssitzungen (in der Regel alle 3 Monate – mit Protokoll)
- Informationswesen institutionalisieren (projektintern z.B. E-Mail, Periodizität Info GL, etc.)
- Abwesenheitsplan für alle Projektbeteiligten (Ferien, regelmäßige Termine, ...)
- Projektname festlegen
- Kick-off Sitzung organisieren

Abb. 235: Checkliste einer Projektplanung

**Dokumentation Projektplan und -handbuch**

Als wichtigstes Ergebnis der Projektplanung entsteht ein so genannter Projektplan, welcher idealerweise sauber dokumentiert wird. Der Projektplan lässt sich durch einige wenige Erweiterungen zu einem "Projekt-handbuch" erweitern, wenn zu Beginn eine konzise Projektbeschreibung mit den Projektzielen aus dem Projektantrag übernommen wird. Der Projektplan muss sich fortlaufend der Situation des Projektes anpassen, was sich bei den anderen Teilen des Projekthandbuchs nicht zwingend so verhält. Die nachfolgende Beispielsstruktur des Projektplans richtet sich nach der Struktur der oben stehenden Projektmanagementfunktionen.

**I ALLGEMEINES**

- Projektbeschreibung
- Erfolgsfaktoren und Ziele

**II PROJEKTORGANISATION**

- Projektorganisation
- Verantwortlichkeiten, Projektrollen, Zuständigkeitsmatrix

**III PROJEKTIINHALT**

- Ergebnisstrukturplan
- Teilprojekte
- Änderungsverfahren bei inhaltlichen Anpassungen

**IV PROJEKTABLAUF UND TERMINPLAN**

- Vorgehensmodell (Phasenplanung)
- Termin und Meilensteinplanung mit Entscheidungspunkten

**V RESSOURCENPLAN**

- Kostenplanung, Personal- und Sachmittelplan

**VI AUFTRAGSMANAGEMENT**

- Verträge
- Interne Vereinbarungen
- Verwaltung der zu erledigenden Aufgaben

**VII KOMMUNIKATIONSMANAGEMENT**

- Dokumentenmanagement
- Informationsmanagement und Projektinformationsplan

**VIII RISIKOMANAGEMENT****IX QUALITÄTSSTRATEGIE UND -PLANUNG****X ARBEITSTECHNIKEN, STANDARDS UND RICHTLINIEN**

### 18.12.2 Projektausführung

Nach der Projektplanung erfolgt meist im Rahmen eines Kick-off-Meetings die Inangsetzung und Überleitung in die operative Projektausführung. Die Projektausführung beinhaltet auf Projektmanagementebene die laufende Wahrnehmung aller Projektmanagementfunktionen. Dies umfasst die integrierte Pflege und Führung der Projektorganisation, das Auftrags- & Kommunikationsmanagement sowie die fortlaufende Abstimmung von Inhalts-, Termin- und Kostenfortschritt (engl: "Integration Management", PMI 2000). Die Projektausführung ist nicht mit der inhaltlichen Projektbearbeitung zu verwechseln, auch wenn diese eng ineinander greifen.

Eine zentrale Aufgabe der Projektführung besteht in der Anordnung und Überwachung von Arbeitsaufträgen. Das gängige Instrument hierzu ist ein sauberes Aufgabenmanagement (s. Abb. 236).

Während im Rahmen der Projektausführung eine fortlaufende Überwachung und Steuerung des Projektfortschrittes erfolgt, erfasst demgegenüber das institutionalisierte Projektcontrolling auf viel formellere Art und Weise den Projektstatus. Man könnte dies mit einer laufenden Buchhaltung und regelmäßigen Quartalsabschlüssen vergleichen. Ein Quartalsabschluss erfasst gewisse formelle Bewertungen, welche sich auf einen klaren Zeitpunkt beziehen.

Pend. Nr.	Team TP	Beschreibung	Priorität	Status	erstellt am	fällig am	Verantwortlich	Ergebnisstatus
1	alle	Beschreibung der Ziele je Teilprojekt	2	Erl.	11.3.06	18.3.06	Thees	Die Ziele werden noch mit GL abgestimmt.
2	1	Abklärung der JDE Schnittstelle	1	offen	11.3.06	18.3.06	Gemperli	...
3	alle	Auflistung der Risiken je Teilprojekt	2	Erl.	11.3.06	18.3.06	König	...
4	2	Definition der Kampagnen-Selektionskriterien	1	Erl.	18.3.06	25.3.06	Meer	...
5	2	Definition des Kundendatenblattes	1	Erl.	18.3.06	25.3.06	Warwick	...
..	...	...	...	...	...	...	...	...

Abb. 236: Beispiel einer Aufgabenliste mit den zu erledigenden Punkten

### 18.12.3 Projektcontrolling

Der Begriff “Projektcontrolling” steht hier wie oben erwähnt für die periodische und phasenbezogene Beurteilung des Projektstatus. Das Projektstatuscontrolling muss über alle Elemente eines Projektes Auskunft geben können. Dieser Zustand wird in einem Projektcontrollingbericht zusammengefasst: Stand der Projektorganisation, Stand der Aufträge/Verträge, Inhaltsfortschritt, Terminfortschritt, Kostenfortschritt, Risikostatus & erreichter Qualitätsstand.

Eine besondere Schwierigkeit betrifft hier die Abschätzung des Status des Inhalts-, Termin- und Kostenfortschrittes. Häufig wird fälschlicherweise davon ausgegangen, dass die absorbierten Kosten einen Anhaltspunkt geben, wo das Projekt in Bezug auf den Inhaltsfortschritt steht. Dies ist leider nicht der Fall, da gewisse Arbeiten unter Umständen nicht effizient wahrgenommen werden. Daher muss der Inhaltsfortschritt (Ist-Leistung) autonom ermittelt werden (s. Abb. 237). Man bezeichnet diese wichtigste Größe eines Projektes als den “Earned Value” – der aktuell erarbeitete Wert – des Projektes. Sie lässt sich z.B. ermitteln, indem der Restaufwand zur Fertigstellung des Projektes abgeschätzt wird. Steht diese Größe fest, so kann sie mit den aufgelaufenen Kosten und den geplanten Kosten (aufgrund der terminierten Arbeitspakete) verglichen werden. Idealerweise werden die drei Kurven Plankostenkurve, Istkostenkurve und Ist-Leistungskurve grafisch dargestellt und dem Controllingbericht beigelegt.

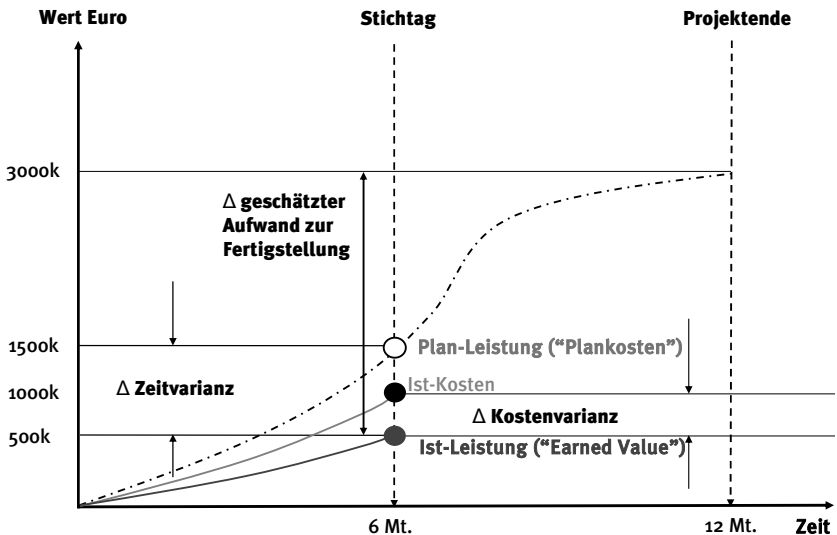


Abb. 237: Controlling vergleicht die “Ist-Leistung” mit den Ist- und Plankosten

### Projektcontrollingbericht

Der regelmäßige Bericht über den Projektstatus erfolgt einerseits durch einen periodischen Arbeitsbericht des Projektleiters zu Händen des Projektausschusses. Dieser Tätigkeitsbericht umfasst eine verbale Beschreibung des inhaltlichen Fortschritts und der Veränderungen im Projekt. In Ergänzung zu diesem Inhaltsbericht der Projektleitung muss ein regelmäßiger Projektcontrollingbericht erstellt werden. Er gibt Auskunft über den Status des Projektes in Bezug auf alle Projektfunktionen.

- **Zusammenfassende Beurteilung des Gesamtprojektes**  
Zusammenfassung des Projektstandes. Dies beinhaltet Inhaltsfortschritt, Termin- & Ressourcensituation, Risikostatus etc. In der Regel wird das Projekt zusätzlich mit einem Gesamtampelstatus versehen (rot, gelb, grün).
- **Status Inhalt**  
Projektstatus in Bezug auf den Inhalt. Dies umfasst den erreichten Projektstand, die Ist-Leistung (Earned Value) und eine Übersicht der wichtigsten Änderungen.
- **Status Termine**  
Projektstatus in Bezug auf die Terminsituation. Es werden die erreichten Meilensteine und Terminabweichungen aufgezeigt.
- **Status Ressourcen**  
Projektstatus in Bezug auf die Kosten- und Ressourcensituation. Die aufgelaufenen Kosten müssen klar dargelegt werden.
- **“Earned Value“-Kennzahlen**  
Die “Earned Value“-Kennzahlen setzen den Inhaltsfortschritt in Beziehung zur Planleistung und den Ist-Kosten. Die Termin- und Kostenabweichungen werden visualisiert.
- **Status Verträge, Vereinbarungen und Aufträge**  
Status des Projektes in Bezug auf die Verträge, interne Vereinbarungen und die Erfüllung von Arbeitsaufträgen.
- **Status des Kommunikationsmanagements**  
Status und Ergebnisse des Projektes in Bezug auf die Information durch das aufgesetzte Kommunikationsmanagement.
- **Status Risiken**  
Status der Projektrisiken. Aufzeigen des Risikoverlaufs und eventueller neuer oder geänderter Risiken.
- **Status Qualität**  
Status des Projektes in Bezug auf die qualitativen Aspekte wie die Umsetzung der Projektgrundlagen, Organisation etc.
- **Empfehlungen**  
Abschließende Zusammenfassung von wichtigen Punkten und Empfehlungen aus dem Statusbericht zuhanden des Projektes.

## 18.13 Projektorganisation und -rollen

### 18.13.1 Formen der Projektorganisation

Wie jedes System verfügt auch das Projektsystem über eine so genannte Projektorganisation. Die Projektorganisation besteht in einem Unternehmen parallel zur permanenten Aufbauorganisation des Unternehmens. In Bezug auf die Projektorganisation können daher drei Grundformen unterschieden werden, die "reine Projektorganisation", die "Projektkoordination" und die "Matrix-Projektorganisation". Diese Formen unterscheiden sich primär im Verhältnis zwischen Projekt- und Aufbauorganisation.

In der reinen Projektorganisation sind die Projektmitarbeiter organisatorisch ausschließlich dem Projekt unterstellt. Das Projekt bildet eine eigene temporäre Organisationseinheit. Der Projektleiter besitzt alle Kompetenzen, welche auch einer normalen Linienführung zukommen würden.

Bei der "Projektkoordination" verfügt der Projektleiter über keine direkten Weisungsbefugnisse. Die Projektmitarbeiter gehen ihren angestammten Tätigkeiten im Unternehmen nach und wirken parallel im Projekt.

Die Matrix-Projektorganisation stellt eine Zwischenform dieser beiden Organisationsformen dar (s. Abb. 238). Es besteht aber eine klare Doppelunterstellung. Oftmals bleiben die Mitarbeiter administrativ dem Fachvorgesetzten unterstellt. In diesen Fällen können die erwähnten "Internen Vereinbarungen" hilfreich sein, da sie den Grad des Einsatzes klar festhalten.

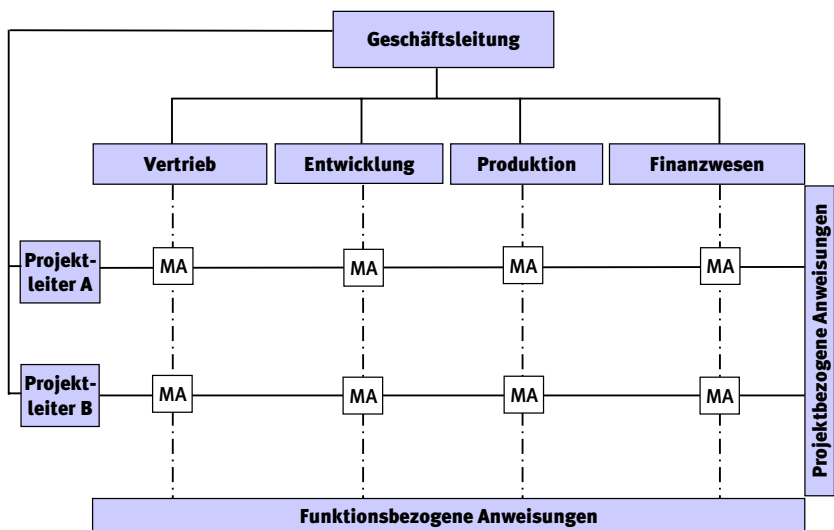


Abb. 238: Matrix Projektorganisation

### 18.13.2 Projekt-Stellenfunktionendiagramm

Das Stellenfunktionendiagramm zeigt beispielhafte Aufgaben von Projektleiter (PL), Projektleiterstellvertreter (STV), Teilprojektleiter (TPL), Projektmitarbeiter (MA), Projektcontroller (CO) und Projektausschuss (PA) sowie die jeweiligen Verantwortlichkeiten und Entscheidungskompetenzen.

<b>Projektfunktionen</b>	<b>PL</b>	<b>STV</b>	<b>TPL</b>	<b>MA</b>	<b>CO</b>	<b>PA</b>
<b>Projektorganisation</b>						x/E
<b>Auftragsmanagement</b>						
Projektverträge	x				x	E
Claimmanagementverfahren	E	x	x	x		
Interne Vereinbarungen	E	x	x	x	x	
Aufgabenmanagement	x					
<b>Kommunikationsmanagement</b>						
Dokumentenablageplanung		x				
Projektinformationsplan	E	x				
Projektmarketingplan	E	x				
<b>Inhaltsmanagement</b>						
Zieldefinition	x	x	x	x		E
Projektstrukturplan	x	x	x	x		E
Ergebnisstrukturplan	x	x	x	x		E
Änderungsverfahren	E	x	x	x		
<b>Terminmanagement</b>						
Vorgehensphasenplan	...	...	...	...		
Termin- und Meilensteinplan						
Projektablaufstrukturplan						
Interne/externe Liefertermine						
<b>Ressourcenmanagement</b>						
Ressourcen- / Kostenplan						
Personal- / Sachmittelplan						
Aktivitätenaufwandsplan						
Projektinfrastrukturplan						
<b>Risikomanagement</b>						
Risikomanagementplan						
Initialer Risikokatalog						
Initiale Massnahmen						
<b>Qualitätsmanagement</b>						
Qualitätssicherungsplan						
QS-Modell erarbeiten						
Prüfplan erarbeiten						
Methoden & Standards						
<b>Berichte</b>						
Projektplan						
Projekthandbuch						
Projektübersicht						

Abb. 239 Beispielhaftes Funktionendiagramm (x: Ausführen, E: Entscheidung)



### 18.13.3 Teamzusammensetzung

In Projekten ist zu berücksichtigen, dass eine ausgewogene Partizipation und die Ermöglichung von Mitsprache bzw. die Einbeziehung verschiedener Interessengruppen zu höherer Akzeptanz der Lösung führen werden.

Partizipation erhöht neben der Sicherheit einer breit abgestützten Lösung zudem die Geschwindigkeit der Projektabwicklung bei komplexen Systemen. Bei komplexen Sachverhalten ist das Wissen in der Regel derart verteilt, dass es ein Ding der Unmöglichkeit wäre, ohne Einbeziehung der relevanten Know-how-Träger innerhalb einer angemessenen Frist zum Ziel zu kommen. Wie bereits oben erwähnt, wird auch keine einzelne Person in der Lage sein, alle Aspekte alleine zu berücksichtigen und die Gesamtzusammenhänge eines komplexen Unternehmens zu verstehen.

Die Beteiligten sind bei partizipativen Ansätzen sodann in der Lage, ein gemeinsames Verständnis für Begriffe zu entwickeln und somit eine gemeinsame Sprache. Beispiel: In einem Projekt wurde darüber diskutiert, was nun ein "Vertriebsauftrag" genau ist. Die Vertriebsabteilung hatte eine andere Definition als die Fertigung. Für die Fertigung stellte jeder Auftrag, der aus dem Vertrieb kam, ein "Vertriebsauftrag" dar. Der Vertrieb verstand indessen darunter nur jene Aufträge, die kundenauftragsunabhängig der Produktion erteilt werden. Beide haben aber stets dasselbe Wort verwendet. Im Zuge der Projektabwicklung konnten sich daran extrem emotionale Diskussionen entzünden.

Bereits in den frühen Phasen eines Projektes muss daher auf die Teamzusammensetzung geachtet werden. Ohne enge Einbindung von Mitarbeitern aus der Geschäftsleitung wird indessen keine größere Veränderung im organisatorischen Sinn möglich sein. Mögliche sinnvolle sozio-technische Neukonzeptionen werden damit scheitern. Es ist daher ratsam, in allen Phasen des Projektes in das Projektteam auch repräsentative Personen mit Einfluss auf die Führungsebene zu integrieren und die Spannweite des möglichen Handlungsraumes bereits früh zu kommunizieren.

### 18.13.4 Team-, Mitarbeiterführung und Konfliktmanagement

Im Rahmen der Projektorganisation und des Teams sind zusätzlich generelle Führungsaufgaben wahrzunehmen. Dies betrifft Aspekte der Mitarbeiterführung, Mitarbeitermotivation und den Umgang mit Problemen und Konflikten. Wir verweisen bei diesen Themengebiete auf die spezialisierte Führungsliteratur (Steiger 2004). In diesem Zusammenhang bleibt anzufügen, dass sich viele Probleme mit einer adäquaten Projektmethodik oftmals vor ihrer Entstehung verhindern lassen würden. Dies gilt als abschliessendes Plädoyer für eine korrekte und methodisch saubere Projektabwicklung, wie sie ja im zweiten Teil des Buches vorgestellt wurde.

## 18.14 Ergebnisübersicht Projektmanagement

	<b>Projektplanung/-start</b>	<b>Projektausführung</b>	<b>Projektcontrolling</b>	<b>Phasen-, Projektende</b>
<b>Projektorganisation</b>	Projektorganisation; Verantwortlichkeiten; Projektrollen; Zuständigkeitsmatrix.	Organisationsänderungen; Projektrollen zuweisen; Verantwortlichkeiten & Zuständigkeiten anpassen.	Status Projektorganisation; Projektausschuss, Stellvertretung Projektleiter vorhanden, ...	Anpassung / Auflösung der Projektorganisation; Überführung in laufende Organisation.
<b>Auftragsmanagement</b>	Projektverträge; Interne Vereinbarungen; Aufgabenliste.	Arbeitsaufträge erteilen; Aufgabenliste führen; Verträge anpassen und laufende Überprüfung.	Status etwaiger "Claims"; Status Vertragsänderungen; Grad der Umsetzung interner Vereinbarungen.	Vertragsbeendigung; Inkraftsetzen laufende Verträge; Planung Nachbesserungen.
<b>Kommunikationsmanagement</b>	Dokumentations- & Dokumentenablageplan; Projektinformationsplan; Projektmarketingplan.	Dokumente verwalten; Informationen verteilen; Projektmarketingaktionen; Releases verwalten.	Status der Dokumentation, Vollzug der Kommunikation; Status der Kommunikation von Projektzielen / -vision.	Dokumentenarchivierung; Kommunikation der Projektzielerreichung.
<b>Inhaltsmanagement</b>	Erfolgsfaktoren & Ziele; Projektstrukturplan; Ergebnisstrukturplan; Änderungsverfahren.	Änderungen melden; Änderungsstatusliste führen Fehler melden; Fehlerstatusliste führen.	Status Inhaltsfortschritt (Earned Value); Status von Änderungen (Change Control).	Feststellen Zielerreichung; Analyse der Abweichungen.
<b>Terminmanagement</b>	Vorgehensphasenplan; Termin- & Meilensteinplan; Projektbaufstrukturplan; Liefertermine (In- & Extern).	Termine koordinieren; Meilensteine überwachen; Liefertermine überwachen; Terminplan nachführen.	Status Terminabweichung; Status der Meilensteine; Analyse Liefertermine.	Begründung von etwaigen Terminabweichungen.
<b>Ressourcenmanagement</b>	Kostenplan; Zahlungsplan; Personal- / Sachmittelplan; Aktivitätsaufwandsplan; Projektinfrastrukturplan.	Personal führen; Aufwand überwachen; Zahlungen vornehmen; Kostenabrechnung.	Status der Kostensituation (Stand der Ist-Kosten).	Abschlussrechnung; Begründung von Kostenüberschreitungen.
<b>Risiko-</b> <b>management</b>	Risikomanagementplan; Initialer Risikokatalog; Initiale Massnahmen.	Risikokatalog revidieren; Massnahmen umsetzen	Anzahl & Status der Risiken; Beurteilung Massnahmen; Empfehlung Massnahmen.	Beurteilung der Risiken für laufenden Betrieb; Massnahmen.
<b>Qualitätsmanagement</b>	Qualitätssicherungsplan; QS-Modell; Prüfplan; Methoden & Standards.	Qualität sicherstellen; Prüfprozeduren definieren; Prüfspezifikationen ausführen.	Beurteilung Qualitätsstatus; Stand der Umsetzung von Qualitätsstandards in allen Projektfunktionsbereichen.	Qualitätsschlussreview; Lehren für andere Projekte.
<b>Berichte</b>	Projektplan (inkl. Projektbeschreibung); Projektübersicht.	Aktueller Projektplan; Sitzungsprotokolle; Projektberichte/-reporting; Projektentscheide.	Projektcontrollingbericht / Projektstatusberichte	Projektphasenbericht; Projektabschlussbericht.

Abb. 240: Ergebnisübersicht der Projektmanagementtätigkeit

**18.15 Literatur zum Projektmanagement**

- Campagna, C.A. (1998): Projektmanagement der Entwicklung individueller betriebswirtschaftlicher Anwendungssoftware: Darstellung und Begründung einer umfassenden Sichtweise des Projektmanagements unter besonderer Berücksichtigung personeller Faktoren. Berlin: Technische Universität Berlin.
- Carr, N.G. (2003): IT Doesn't Matter. In: Harvard Business Review, May 2003.
- Gareis, R. (1991): Projektmanagement im Maschinen- und Anlagenbau. Wien: Manz.
- Gemünden, G. (1998): Schlüsselfaktoren des Projekterfolgs. In: Human Resource Management. Köln: Verlag Deutscher Wirtschaftsdienst.
- Hermes (2003): Hermes: Führen und Abwickeln von Projekten der Informations- und Kommunikationstechnik (IKT); Handbuch Systementwicklung. Bern: Informatikstrategieorgan Bund ISB.
- Litke, H.D. (2004): Projektmanagement. München: Carl Hanser.
- Jenny, B. (2003): Projektmanagement: Das Wissen für eine erfolgreiche Karriere. Zürich: vdf Hochschulverlag.
- Martin, R. (1993): Einflussfaktoren auf Akzeptanz und Einführungsumfang von Produktionsplanung und -steuerung (PPS). Frankfurt: Peter Lang.
- Steiger, T., Lippmann, E. (2004). Handbuch angewandte Psychologie für Führungskräfte. Berlin: Springer.
- Vetter, H. (2004). Projektmanagement. In: Handbuch angewandte Psychologie für Führungskräfte (Steiger, Th.; Lippmann, E.; Hrsg.). Führungskompetenz und Führungswissen, Band 2. Berlin: Springer.
- Project Management Institute (2000). A Guide to the Project Management Body of Knowledge (PMBOK guide). Pennsylvania: Project Management Institute.



## Literaturverzeichnis

- Agiplan (1993): PPS erfolgreich einführen und nutzen, Praxis-Studie. Mülheim/Ruhr.
- Alard, R. (2002): Internetbasiertes Beschaffungsmanagement direkter Güter: Konzept zur Gestaltung der Beschaffung durch Nutzung internetbasierter Technologien; Dissertation ETH Zürich.
- Anderson, J.R. (2001): Kognitive Psychologie: Eine Einführung. (3. Aufl.). Heidelberg: Spektrum der Wissenschaft Verlagsgesellschaft.
- Armbruster, S., Springer, J., Luczak, H. (1998): Bewertung von PPS-Systemen hinsichtlich der Erfüllung von Benutzeranforderungen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 327–375.
- Atkins, P. (1926). Factory Management. New York: Prentice-Hall.
- Bainbridge, L., Sanderson, P. (1995): Verbal Protocol Analysis. In: Evaluation of Human Work: A Practical Ergonomics Methodology (Wilson, J.R., Corlett, E.N.; Hrsg.), 2<sup>nd</sup> edition. London: Taylor & Francis Ltd. 169–201.
- Balzert, H. (1996): Lehrbuch der Software-Technik: Software Entwicklung. Heidelberg/Berlin: Spektrum Akademischer Verlag.
- Bauknecht, K. (Hrsg.) (1992): Informatik-Anwendungsentwicklung – Praxiserfahrungen mit CASE: Probleme, Lösungen und Erfahrungen bei Einführung und Einsatz von CASE. Stuttgart: Teubner.

- Bauknecht, K., Zehnder, C.A. (1997): Grundlagen für den Informatikeinsatz. Stuttgart: Teubner.
- Beck K., Cunningham W. (1989). A Laboratory for Teaching Object-Oriented Thinking. ACM SIGPLAN Notices vol. 24 (10).
- Becker, M., Haberfellner, R., Liebetrau, G. (2000): EDV-Wissen für Anwender: ein Informatik-Handbuch für die Praxis (11. Aufl.). Zürich: Industrielle Organisation.
- Beer, S. (1962): Kybernetik und Management.). Frankfurt am Main: S. Fischer.
- Bellotti, V. (1988): Implications of Current Design Practice for the Use of HCI Techniques. In: People and Computers IV (Jones, D.M., Winder, R.; Hrsg.). Cambridge: Cambridge University Press. 13–34.
- Bertin, J. (1982): Graphische Darstellungen und die graphische Weiterverarbeitung der Information. Berlin: Walter de Gruyter.
- Blessing, D. (2002): Wissensmanagement in Beratungsunternehmen. Nordestedt: BoD.
- Boehm, B.W. (1986): Wirtschaftliche Software-Produktion. Wiesbaden, Forkel.
- Booch, G. (1991): Object oriented design with applications. Redwood City, CA: Benjamin/Cummings.
- Booch, G., Rumbaugh, J., Jacobson, I. (1999): The Unified Modeling Language User Guide. Reading, MA: Addison Wesley Longman.
- Brynjolfsson, E. (1993): The Productivity Paradox of Information Technology: Review and Assessment. In: Communications of the ACM, Dec.
- Burger, G. (2004): Handover-Management: Informationsversorgung für den After-Sales Service. Zürich: vdf Hochschulverlag.
- Burkhard, R. (1997): UML – Unified Modeling Language: Objektorientierte Modellierung für die Praxis. Bonn: Addison Wesley Longman.
- Campagna, C.A. (1998): Projektmanagement der Entwicklung individueller betriebswirtschaftlicher Anwendungssoftware: Darstellung und Begründung einer umfassenden Sichtweise des Projektmanagements unter besonderer Berücksichtigung personeller Faktoren. Berlin: Technische Universität Berlin.
- Chapin, N. (1970): Flowcharting with the ANSI Standard: A Tutorial. ACM Computing Surveys, Vol. 2, No. 2. New York: ACM Publications.

- Chestnut, H. (1966): Systems Engineering Tools. New York: John Wiley & Sons.
- Coad, P., Yourdon, E. (1991, 2<sup>nd</sup> ed.): Object-Oriented Analysis. Englewood Cliffs, New Jersey: Yourdon Press/Prentice Hall.
- Cox, B., Novobilski, A. (1991, 2<sup>nd</sup> ed.): Object-oriented Programming: An Evolutionary Approach. Reading, MA: Addison-Wesley.
- Daenzer, W. (Hrsg.) (1976): Systems Engineering: Leitfaden zur methodischen Durchführung umfangreicher Planungsvorhaben. Zürich: Verlag Industrielle Organisation.
- Davenport, T. (1993): Process Innovation: Reengineering Work through Information Technology. Boston, MA: Harvard Business School Press.
- DeMarco, T. (1978): Structured Analysis and System Specification. Englewood Cliffs, NJ: Prentice Hall.
- Dunckel, H., Volpert, W., Zölch, M. et al. (1993): Kontrastive Aufgabenanalyse im Büro. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 7. Zürich/Stuttgart: Verlag der Fachvereine/Teubner.
- Dunckel, H. (Hrsg.) (1999): Handbuch psychologischer Arbeitsanalyseverfahren. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 14. Zürich: vdf Hochschulverlag.
- Ehn, P., Kyng, M. (1991): Cardboard Computers: Mocking-it-up or Hands-on the Future. In: Design at Work: Cooperative Design of Computer Systems; (Greenbaum, J., Kyng, M.; Hrsg.). Hillsdale, NJ: Lawrence Erlbaum. 169–196.
- Emery, F.E. (1967): Analytical Model for Socio-Technical Systems. In: The Emergence of a New Paradigm of Work (Emery, F.E.; Hrsg.); Canberra: Australian National University. 95–106.
- Ericsson, K.A., Simon, H.A. (1984): Protocol Analysis: Verbal Reports as Data. Cambridge, MA: MIT Press.
- Fleisch, E. (2001): Das Netzwerkunternehmen: Strategien und Prozesse zur Steigerung der Wettbewerbsfähigkeit in der “Networked economy”. Berlin: Springer.
- Frei, F. (1981): Psychologische Arbeitsanalyse – eine Einführung zum Thema. In: Beiträge zur psychologischen Tätigkeitsanalyse; (Frei, F., Ulich, E.; Hrsg.) Schriften zur Arbeitspsychologie, Band 31. Bern: Huber. 11–36.

- Gane, C., Sarson, T. (1975): Structured Systems Analysis: Tools and Techniques, Englewood Cliffs, N.J.: Prentice Hall
- Gareis, R. (1991): Projektmanagement im Maschinen- und Anlagenbau. Wien: Manz.
- Gause, D., Weinberg, G. (1993): Software Requirements: Anforderungen erkennen, verstehen und erfüllen. München: Carl Hanser.
- Gemünden, G. (1998): Schlüsselfaktoren des Projekterfolgs. In: Human Resource Management. Köln: Verlag Deutscher Wirtschaftsdienst.
- Glinz, M. (1999). Eine geführte Tour durch die Landschaft der Software-Prozesse und -Prozessverbesserung. In: Informatik/Informatique Nr. 6, Dez. 1999.
- Graham, I. (2001): Object Oriented Methods. Reading, MA: Addison-Wesley.
- Grote, G. (1991): Effekte der Nutzung eines Bürokommunikationssystems auf Arbeitsprozesse und -strukturen. In: German Chapter of the ACM, Band 34: Computergestützte Gruppenarbeit (CSCW). (Friedrich, J., Rödiger, K.H.; Hrsg.); Stuttgart: Teubner; 221–249.
- Grupp, B. (1994). Standard-Software richtig auswählen und einführen: Mit System zur kostengünstigen und umfassenden DV-Lösung. Wuppertal: TAW-Verlag.
- Grupp, B. (1995). EDV-Pflichtenheft zur Hardware- und Softwareauswahl: Praktische Anleitung. Köln: TÜV Rheinland.
- Grull, W. (1928). Die Organisation von Fabrikbetrieben. Leipzig: Glöckner.
- Grün, J., (1995): Führen durch visuelles Management. In: io Management Zeitschrift, 64/10. Zürich: Verlag Industrielle Organisation. 23–27.
- Haberfellner, R., Nagel, P. et al. (2002): Systems Engineering: Methodik und Praxis, 11. Aufl. (Daenzer, W., Huber, F.; Hrsg.). Zürich: Verlag Industrielle Organisation.
- Hacker, W. (1978): Allgemeine Arbeits- und Ingenieurpsychologie: Psychische Struktur und Regulation von Arbeitstätigkeiten; Schriften zur Arbeitspsychologie, Band 20. Bern: Hans Huber.
- Hacker, W. (1992): Expertenkönnen: Erkennen und Vermitteln. Göttingen: Verlag für Angewandte Psychologie.
- Hall, A. (1978): A Methodology for Systems Engineering. Princeton: Van Nostrand.
- Hammer, M., Champy, J. (1993): Reengineering the Corporation. New York: Harper Business.



- Hebeisen, W. (1999): Taylor und der Taylorismus. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 23. Zürich: vdf Hochschulverlag.
- Heiderich, T., Schotten, M. (1998): Prozesse. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 75–143.
- Heinrich, L.J. (1996): Systemplanung: Planung und Realisierung von Informatik-Projekten. 7. Auflage. München: Oldenbourg.
- Heinrich, L.J. (1997): Management von Informatik-Projekten. München: Oldenbourg.
- Hermes (2003): Hermes: Führen und Abwickeln von Projekten der Informations- und Kommunikationstechnik (IKT); Handbuch Systementwicklung. Bern: Informatikstrategieorgan Bund ISB.
- Hess, T., Brecht, L. (1996): State of the Art des Business Process Redesign: Darstellung und Vergleich bestehender Methoden. 2. Auflage. Wiesbaden: Gabler.
- HIPO 1974: HIPO – A Design Aid and Documentation Technique. White Planes, N.Y.: IBM.
- Holz auf der Heide, B. (1993): Welche software-ergonomischen Evaluationsverfahren können was leisten? In: German Chapter of the ACM, Band 39: Software-Ergonomie '93. (Rödiger, K.H.; Hrsg.); Stuttgart: Teubner. 157–171.
- IDEF (1981): Standard Users Manual for the ICAM Function Modeling Method – IDEF0, IEEE draft standard, P1320-1-1-1997. Piscataway, N.J.: IEEE Publications.
- Jacob, O. (1989). Aufgabenintegrierte Büroinformationssysteme: Allgemeines Datenmodell und Probleme der Realisierung. Heidelberg: Physica.
- Jacobson I. (1992): Object-Oriented Software Engineering: A Use Case Driven Approach. Reading, MA: Addison-Wesley.
- Jacobson I. (2001): The Object Advantage: Business Process Reengineering with Object Technology. Reading, MA: Addison-Wesley.
- Jacobson I. (1995): The Use-Case Construct in Object-Oriented Software Engineering. In: Scenario-Based Design – Envisioning Work and Technology in System Development. (Carroll, J.; ed.). New York: John Wiley & Sons. 309–336.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999): The Unified Software Development Process. Reading, MA: Addison Wesley Longman.

- Jantzen-Homp, D. (2000): Projektportfolio-Management: Multiprojektarbeit im Unternehmenswandel. Wiesbaden: Gabler.
- Jenny, B. (2003): Projektmanagement: Das Wissen für eine erfolgreiche Karriere. Zürich: vdf Hochschulverlag.
- Johannson, H., McHugh, P. et al. (1996): Business Process Reengineering: Break Point Strategies for Market Dominance. Chichester, West Sussex: John Wiley & Sons.
- Johnson, P. (1992): Human Computer Interaction: Psychology, Task Analysis and Software Engineering. Cambridge, MA: Harvard University Press.
- Jungk, R. (1994): Zukunftswerkstätten: mit Phantasie gegen Routine und Resignation. München: Heyne.
- Kaiser, H., Paegert, C., Schotten, M. (1998): Auswahl von PPS-Systemen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 292–326.
- Kaplan, R., Norton, D.P. (1997). Balanced Scorecard: Strategien erfolgreich umsetzen. Stuttgart: Schäffer-Poeschel.
- Kay, A. (1990): User Interface: A Personal View. In: The Art of Human-Computer Interface Design. (Laurel, B.; Hrsg.); Reading, MA: Addison-Wesley. 191–207.
- Keil, M., Carmel, E. (1995): Customer-Developer Links in Software Development. In: Communications of the ACM, 38(5). 33–44.
- Kuhnt, B., Müllert, N.R. (2000). Moderationsfibel Zukunftswerkstätten: verstehen, anleiten, einsetzen; das Praxisbuch zur Sozialen Problemlösungsmethode Zukunftswerkstatt (3. Aufl.). Münster: Ökoptopia.
- Kyng, M. (1995): Creating Contexts for Design. In: Scenario-Based Design – Envisioning Work and Technology in System Development. (Carroll, J.; ed.). New York: John Wiley & Sons. 85–107.
- Lechner, S. (1993). Betriebstypologische Auswahl von Standardsoftware zur Produktions-Planung und Steuerung (PPS). Fortschrittsberichte VDI Reihe 20, Nr. 89. Düsseldorf: VDI Verlag.
- Litke, H.D. (2004): Projektmanagement. München: Carl Hanser.
- Leont'ev, A.N. (1977): Tätigkeit, Bewusstsein, Persönlichkeit. Stuttgart: Klett.
- Macaulay, L. (1988): Requirements Engineering. Berlin: Springer.

- Martin, C.F. (1988): User-Centered Requirements Analysis. Englewood Cliffs, NJ: Prentice Hall.
- Martin, R. (1993): Einflussfaktoren auf Akzeptanz und Einführungsumfang von Produktionsplanung und -steuerung (PPS). Frankfurt: Peter Lang.
- Matern, B. (1983): Psychologische Arbeitsanalyse. Spezielle Arbeits- und Ingenieurpsychologie (Hacker, W.; Hrsg.), Lehrtext 3. Berlin: Deutscher Verlag der Wissenschaften.
- Mayring, P. (2002): Qualitative Inhaltsanalyse: Grundlagen und Techniken. Weinheim: Deutscher Studien Verlag.
- Mertens, P. (2000). Integrierte Informationsverarbeitung 1: Administrations- und Dispositionssysteme in der Industrie, 12. überarbeitete Auflage. Wiesbaden: Gabler.
- Mertens, P., Bodendorf, F. et al. (2004): Grundzüge der Wirtschaftsinformatik. 8. Auflage. Berlin: Springer.
- Meyer, B. (1988): Object-oriented Software Construction. Englewood Cliffs, New Jersey: Prentice-Hall.
- Miles, M.B., Hubermann, M. (1994): Qualitative Data Analysis: An expanded sourcebook. Thousand Oaks: SAGE Publ.
- Mumford, E., Weir, M. (1979). Computer systems in work design – the ETH-ICS method: Effective Technical and Human Implementation of Computer Systems. New York: John Wiley & Sons.
- Müller, R. (1983): Zur Geschichte des Modelldenkens und des Modellbegriffs. In: Modelle – Konstruktion der Wirklichkeit. (Stachowiak, H.; Hrsg.); München: Wilhelm Fink.
- Murer, T. (1997). Project Gipsy: Facing the Challenge of Future Integrated Software Engineering Environments. Zürich, ETH: TIK Schriftenreihe, Nr. 21.
- Norman, D., Draper, S. (1986): User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum.
- NOVE-IT (2003): Informatikprozesse der Bundesverwaltung: Definierte Abläufe in der Informatik. Bern: Informatikstrategieorgan Bund ISB.
- Oestereich, B. (2001): Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified Modeling Language, 5. völlig überarbeitete Auflage. München: Oldenbourg.

- Oesterreich, R. (1984): Zur Analyse von Planungs- und Denkprozessen in der Produktion. Das Arbeitsanalyseinstrument VERA. In: *Diagnostica* 30, 216–234.
- Oppermann, R., Murchner, B., Reiterer, H. & Koch, M. (1994): Software-ergonomische Evaluation. Der Leitfaden EVADIS II, 2. überarbeitete Auflage. Berlin: DeGruyter.
- Osterloh, M., Frost, J. (2003): Prozessmanagement als Kernkompetenz: wie Sie Business Reengineering strategisch nutzen können. 3. Auflage. Wiesbaden: Gabler.
- Österle, H. (Hrsg.) (1990): Integrierte Standardsoftware: Entscheidungshilfen für den Einsatz von Softwarepaketen. Band 2: Auswahl, Einführung und Betrieb von Standardsoftware. München: Angewandte Informations Technik – AIT.
- Österle, H. (1995): Business Engineering: Prozess- und Systementwicklung. Berlin: Springer.
- Pleschak, F., Sabisch, H. (1996): Innovationsmanagement. Stuttgart: Schäffer-Poeschel.
- Pohl, K. (1996): Process-Centered Requirements Engineering. New York: John Wiley & Sons.
- Project Management Institute (2000). A Guide to the Project Management Body of Knowledge (PMBOK guide). Pennsylvania: Project Management Institute.
- Rauterberg, M. (1995): Ein Konzept zur Quantifizierung software-ergonomischer Richtlinien. Zürich: Institut für Arbeitspsychologie, Eidgenössische Technische Hochschule.
- Rauterberg, M., Spinas, P. et al. (1994): Benutzerorientierte Software-Entwicklung: Konzepte, Methoden und Vorgehen zur Benutzerbeteiligung. Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 3. Zürich/Stuttgart: Verlag der Fachvereine/Teubner.
- Rieder, B. (1988): Die Gestaltung des Implementierungsprozesses bei der Einführung von integrierter Standardsoftware. Dissertation, Regensburg.
- Roos, E. (1992). Informationsmodellierung für PPS-Systeme: Ein Konzept zur aufgabenorientierten Systementwicklung. Berlin: Springer.
- Ross, D. (1985): Applications and extensions of SADT. Piscataway, N.J.: IEEE Publications.

- Rosson, M.B., Maass, S., Kellogg, W.A. (1988): The Designer as User: Building Requirements for Design Tools from Design Practice. In: Communications of the ACM, 31(11). 1288–1298.
- Rosson, M.B., Carroll, J. (1995): Narrowing the Specification-Implementation Gap in Scenario-Based Design. In: Scenario-Based Design – Envisioning Work and Technology in System Development. (Carroll, J.; ed.). New York: John Wiley & Sons. 247–278.
- Rumbaugh, J. (1991): Object-oriented Modeling and Design. Englewood Cliffs, NJ: Prentice Hall.
- Saad, K.N. et al. (1991): Management der F&E-Strategie. Wiesbaden: Gabler.
- Schank, R.C. & Abelson, R.P. (1977): Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures. Hillsdale, NJ: Lawrence Erlbaum.
- Scheer, A.-W. (1990). Wirtschaftsinformatik: Informationssysteme im Industriebetrieb. Berlin: Springer.
- Scheer, A.W. (2001): ARIS: Modellierungsmethoden, Metamodelle, Anwendungen. Berlin: Springer.
- Scheer, A.-W. (1998): Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. Berlin: Springer
- Schmager, B. (1989). Betriebsdatenerfassung und -verarbeitung als Instrument der betrieblichen Informationsgewinnung: Ein Ansatz zur Reduzierung der Distanzen bei der BDEV unter Berücksichtigung ziel- und benutzerorientierter Aspekte. Hamburg: Technische Universität Hamburg-Harburg.
- Schmitz, R. (1998): Einführung von PPS-Systemen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 327–375.
- Schnetzler M., Sennheiser, A. (2003): Identification of Improvement Strategies in Production Networks. In: Building the Knowledge Economy – Issues, Applications, Case Studies (Cunningham, P. et al.; ed.). Amsterdam: IOS Press. 306–312.
- Schönsleben, P., Specker, A. et al. (1999): Unternehmerische Behandlung der Softwareproduktion – mehr als den Softwareentwicklungsprozess im Griff haben. In: io Management Zeitschrift, 6/99. 90–96.

- Schönsleben, P. (2000): Integrales Logistikmanagement: Planung und Steuerung von umfassenden Geschäftsabläufen. Berlin: Springer.
- Schönsleben, P. (2001): Integrales Informationsmanagement: Informationssysteme für Geschäftsprozesse – Management, Modellierung, Lebenszyklus und Technologie. Berlin, Heidelberg, New York: Springer.
- Schotten, M. (1998): Aachener PPS-Modell. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 9–28.
- Schotten, M. (1998a): Funktionen. In: Produktionsplanung und -steuerung; (Luczak, H., Eversheim, W.; Hrsg.). Berlin: Springer. 144–218.
- Sennheiser, A. (2004): Determinant based selection of Benchmarking Partners and Logistics Performance Indicators; Dissertation ETH Zürich.
- Sommerville, I. (2000): Software Engineering, 6<sup>th</sup> ed. Reading, MA: Addison-Wesley.
- Specker, A. (1993): Die Informatikwelt wird zum sozialen Gefüge: Vom Personal Computing zum Interpersonal Computing. In: Neue Zürcher Zeitung, Nr. 225, ORBIT-Beilage.
- Specker, A. (1997): Kognitives Software Engineering: Ein schema- und scriptbasierter Ansatz. Zürich: vdf Hochschulverlag.
- Specker, A. (1999): Gegenständliches CAD. In: Design Thinking: Analyse und Unterstützung konstruktiver Entwurfstätigkeiten (Sachse, P.; Specker, A.; Hrsg.) (Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 22. Zürich: vdf Hochschulverlag.
- Specker, A. (2000): Unterstützung kritischer Erfolgsfaktoren durch ERP-Systeme: Resultate einer Expertenbefragung anlässlich der Zürcher PPS-Tage 2000. [www.tagungen.bwi.ethz.ch/rueckblickpps](http://www.tagungen.bwi.ethz.ch/rueckblickpps).
- Spisak, M. (1999). Organisation als Führungsaufgabe. In: Handbuch angewandte Psychologie für Führungskräfte (Steiger, Th.; Lippmann, E.; Hrsg.). Führungskompetenz und Führungswissen, Band 2. Berlin: Springer.
- Stahlknecht, P. (2002). Einführung in die Wirtschaftsinformatik, 10. überarbeitete Auflage. Berlin: Springer.
- Standish Group (2003): Chaos Report. West Yarmouth, Ma.
- Steiger, T., Lippmann, E. (2004). Handbuch angewandte Psychologie für Führungskräfte. Berlin: Springer.

- Sterman, J.D. (2000): Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill.
- Störmer, W., Droege, M. (1997). Personalzeiten und Betriebsdaten: Konzepte, Lösungen und Erfahrungen aus der Praxis. München: Hanser.
- Straub, W. (2004): Informatikrecht: Einführung in Softwareschutz, Projektverträge und Haftung. Zürich/Bern: vdf Hochschulverlag/Stämpfli.
- Strohm, O., Ulich, E. (Hrsg.) (1997): Unternehmen arbeitspsychologisch bewerten: ein Mehr-Ebenen-Ansatz unter besonderer Berücksichtigung von Mensch, Technik und Organisation (Schriftenreihe Mensch – Technik – Organisation (Ulich, E.; Hrsg.), Band 10. Zürich: vdf Hochschulverlag.
- Strohm, O. (1990): Arbeitsorganisation, Methodik und Benutzerorientierung bei der Softwareentwicklung. Projektberichte Nr. 2 zum Forschungsprojekt Benutzer-Orientierte Softwareentwicklung und Schnittstellengestaltung (P. Spinas, M. Rauterberg, O. Strohm, D. Waeber & E. Ulich; Hrsg.). Zürich: ETH, Institut für Arbeitspsychologie.
- Sturm, R., Morris, W., Jander, M. (2000): Foundations of Service Level Management. Indiana: SAMS.
- SVD (1985): Evaluation von Informatiklösungen: Verfahren – Methoden – Beispiele. Schweizerische Vereinigung für Datenverarbeitung (Hrsg.). Bern/Stuttgart: Haupt.
- Taylor, D.A. (1990): Object Oriented Technology: A Manager's Guide. Reading, MA: Addison-Wesley.
- Taylor, F.W. (1913): Die Grundsätze wissenschaftlicher Betriebsführung. Berlin: Oldenburg.
- Tipotsch, C. (1997): Business Modelling: Vorgehensmethodik und Gestaltungsmodelle. Graz: Technische Universität.
- Trist, E.L. & Bamforth, K.W. (1951): Some social and psychological consequences of the longwall method of coal getting. In: Human Relations 4; 3–38.
- Tschirky, H., Koruna, S. (Hrsg.) (1998): Technologie-Management: Idee und Praxis. Zürich: Verlag Industrielle Organisation.
- UML (2003): OMG Unified Modeling Language Specification. Needham, MA: Object Management Group.

- Ulich, E. (1978): Über das Prinzip der differentiellen Arbeitsgestaltung. In: *io Management Zeitschrift*, 47/12. 566–568.
- Ulich, E. (2001): *Arbeitspsychologie*, 5. Aufl. Zürich/Stuttgart: vdf Hochschulverlag/Schäffer-Poeschel.
- Vetter, H. (2004). Projektmanagement. In: *Handbuch angewandte Psychologie für Führungskräfte* (Steiger, Th.; Lippmann, E.; Hrsg.). *Führungskompetenz und Führungswissen*, Band 2. Berlin: Springer.
- Vogt, W. (2000): *Nutzen ohne Frust: IT Service kundenorientiert planen und steuern*. Basel: Perseo.
- Volpert W., Oesterreich, R., Gablenz-Kolakovic, S., Krogoll, T., Resch, M. (1983): *Verfahren zur Ermittlung von Regulationserfordernissen in der Arbeitstätigkeit (VERA). Analyse von Planungs- und Denkprozessen in der industriellen Produktion*. Köln: Verlag TÜV Rheinland.
- von Weizsäcker, C.F. (1991). Wahrnehmung des Denkens. In: *Erfahrung des Denkens – Wahrnehmung des Ganzen: Carl Friedrich von Weizsäcker als Physiker und Philosoph* (Ackermann, P., Eisenberg, W.; Hrsg.). Berlin: Akademie-Verlag.
- Weber, W.G. (1994): *Psychologische Analyse und Bewertung computergestützter Facharbeit*. Berlin: Quintessenz.
- Weber, W.G. (1996): *Analyse von Gruppenarbeit in der Produktion: Versuch einer Integration von Konzepten zur kollektiven Handlungsregulation in soziotechnischen Systemen*. Habilitationsschrift. Zürich: Eidgenössische Technische Hochschule.
- Wehner, T., Clases, C., Endres, E. (1996): *Situiertes Lernen und kooperatives Handeln in Praxisgemeinschaften*. In: *Zwischenbetriebliche Kooperation: Die Gestaltung von Lieferbeziehungen* (Endres, E., Wehner, T.; Hrsg.). Weinheim: Psychologie Verlags Union. 71–85.
- White, I. (1994): *Using the Booch Method: A Rational Approach*. Redwood City, CA: Benjamin/Cummings.
- Wiener, N. (1948, dt. 1963): *Kybernetik. Regelung und Nachrichtenübertragung in Lebewesen und Maschine*. Düsseldorf: ECON.
- Wirth, N. (1971): *Program Development by Stepwise Refinement*. In: *Communications of the ACM*, Vol. 14, No. 4. New York: ACM Publications.
- Yourdon, E., Constantine, L. (1975): *Structured Design: Fundamentals for a Discipline of Computer Program and System Design*. New York: Yourdon Press.



- Yourdon, E. (1989): Modern Structured Analysis. Englewood Cliffs, NJ: Prentice Hall.
- Zehnder, C.A. (2002): Informatik-Projektentwicklung: Eine Einführung für Informatikstudenten und Praktiker, 4. Aufl. Zürich: vdf Hochschulverlag.
- Zehnder, C.A. (1998): Informationssysteme und Datenbanken, 6. Aufl. Zürich: vdf Hochschulverlag.
- Zuboff, S. (1988): In the Age of the Smart Machine: The Future of Work and Power. New York: Basic Books.



# 20

## Stichwortverzeichnis

### A

Abnahme .....	357, 366
Abnahmeentscheidung .....	369
Abnahmeprotokoll .....	369
Aktivitäten .....	54
Akzeptanz .....	184
Analyse .....	26
aspektweise .....	33
Beziehungen.....	36
funktionsorientierte.....	80
prozessorientierte .....	56
Änderungsmanagement.....	394
Anforderungsmanagement .....	394
Anforderungsvielfalt .....	268
Anpassungsfähigkeit .....	208
Arbeitsgestaltung, differentielle .....	332
Arbeitsobjektdiagramm .....	40, 45
Definition .....	126
Arbeitsplatzmetapher .....	158
Arbeitspsychologie .....	132
ARIS .....	74
Aufgabe .....	
Aufgabenhierarchie .....	118
Bewertung .....	121
Definition.....	35, 116

Detailkonzeption .....	332
Identifikation .....	119
Modellierung .....	115
Spezifikation .....	120
Vererbung .....	118
Vollständigkeit .....	121
Aufgabenmodell .....	45
Aufgabensicht .....	116, 129
Definition .....	35
Aufgabenstellung .....	206
Auftragsmanagement .....	397
Ausschreibung.....	295
Auswahl.....	196
Autonomie .....	268, 272

### B

Basistechnologie .....	217
Baumstruktur .....	47
Bearbeitung .....	
hierarchisch-sequentielle .....	59
Bearbeitungselemente .....	33
Beobachtungstechniken .....	192
Berechtschaftsanalyse .....	359
Berichtswesen .....	402
Betrieb und Unterhalt .....	373
Betriebskosten .....	212

Bewertung .....	196, 245, 298
Beziehungen .....	26
Analyse der .....	36
Darstellung von .....	38
Use-Cases .....	91
von Systemaspekten .....	40
Black-box .....	31, 63
Bottom-up .....	83, 84
Brainstorming .....	195, 238
Bubbles .....	88
Business Process Reengineering .....	262

## C

Change Request .....	394
Class-Responsibilities-Collaborators .....	44, 108
Definition .....	108
Contracts .....	397
Customer Developer Links .....	184

## D

Darstellungen .....	
aspektweise .....	39
geschachtelte .....	47
reine .....	39
Datenbestände .....	147
Datenflussdiagramm .....	40, 43, 88
Definition .....	88
Hierarchische Gliederung .....	89
Datenobjektdiagramm .....	147
Datenspeicher .....	88
Datenübernahme .....	365
DeMarco .....	92
Design .....	341
Detailkonzept .....	321
Dezentralisierung .....	132
Dienstleistungsvertrag .....	312
Dokumentenmanagement .....	398
Durchlaufzeit .....	66, 257

## E

Ebene 0 .....	89
Einführung .....	
gestaffelte .....	361
Strategien .....	363

Einführungsdauer .....	168
Einführungskosten .....	396
Entity Relationship Model .....	92
Erfolgsfaktoren .....	
kritische .....	66
Ergonomie .....	292
Erhebungstechniken .....	192
Evaluation .....	295, 296, 389

## F

Flexibilität .....	208, 267
Flow-Chart .....	42
Flussdiagramm .....	42, 325
Definition .....	68
<i>Funktion</i> .....	
<i>Definition</i> .....	35, 80
Elementarfunktion .....	81, 84, 86
Identifikation .....	83
Modellierung .....	79
Spezifikation .....	86
Systemabgrenzung .....	83, 85
Funktionale Integration .....	267
Funktionen-Blockdiagramm .....	43, 87
Definition .....	87
Funktionsmodell .....	43, 80, 82, 265
Definition .....	82
Funktions-Prozess-Matrix .....	125
Funktionssicht .....	80
Definition .....	80
Funktionsunterstützungsdiagramm .....	150

## G

Gane & Searson .....	92
Ganzheitlichkeit .....	268
Generalunternehmervertrag .....	310
Geschäftsprozess .....	58, 239
Geschäftsregeln .....	65
Grobkonzept .....	223
Gruppen .....	236

## H

Handlungsabläufe .....	56
Handlungsautonomie .....	389
Hardwarevertrag .....	314

## Hierarchie

Aufgabenhierarchie .....	118
Bestandteilhierarchie .....	32, 59, 101
flache.....	61
Funktionshierarchie .....	32
Objekthierarchie .....	101
Prozesshierarchie .....	54
Systemarchitektur .....	140
Vererbungshierarchie .....	32, 102
Hierarchieebene .....	47
HIPO .....	82, 92

**I**

## Identifikation

Aufgaben .....	119
Funktionen .....	83
Objekte.....	104
Prozesse .....	62
Techniksysteme.....	143
Implementation .....	341, 349
Inbetriebnahme .....	357
Informatikgeschichte .....	232
Informationssystem	
aufgabenorientiertes .....	158
Dimensionen von .....	37
funktionsorientiertes .....	82, 155
integriertes .....	153
objektorientiertes .....	156
prozessorientiertes.....	154
Inhaltsmanagement .....	392
Inkremental-Entwicklung .....	181
Installation.....	364
Interaktion, soziale .....	268
Interne Projektvereinbarungen .....	397
Interview .....	192
Investitionsantrag.....	204
Investitionskosten .....	212

**K**

Kenngrossen .....	66
Klassen .....	105
Kollaborationsdiagramm .....	44, 109
Kommunikation .....	208
Kommunikationsmanagement .....	398

Komplettumstellung .....	361
Komplexität .....	27, 36, 169
Konfiguration.....	350
Konfigurationsmanagement .....	394
Konsistenzcheck.....	89
Kontextdiagramm .....	62, 85
Konzept .....	241, 251
Kosten .....	66, 257
Kosten- und Aufwandsplanung.....	396
Kriterien.....	291
Kriterienkatalog.....	291
Kritische Erfolgsfaktoren .....	184
Kundenorientierung.....	66

**L**

Lebensphasen .....	169, 218
Lösungssuche.....	194

**M**

Matrix .....	39
Medils .....	74
Mehrdimensionalität .....	27
Meilensteine.....	173
Meilensteinmanagement .....	395
Meinungsdifferenzen.....	366
Methoden	
aufgabenorientierte .....	116
funktionsorientierte .....	81
Grobkonzept.....	226
objektorientierte .....	99
prozessorientierte .....	57
Minispec .....	86
Mission Statement.....	207
Modelle .....	57
Modellierung	
aspektororientierte.....	41
Aufgaben.....	45
Funktionen .....	43
Informationssysteme.....	25
Morphologie der.....	46
Objekte .....	44
Probleme bei .....	18
Prozesse.....	42, 54, 57
Technik.....	137

Zielsetzung der .....	28
Modellierungsmatrix .....	40
Modellierungstechniken	
aspektbezogene .....	40
Morphologischer Kasten .....	244
MTO-Analyse .....	127

## N

Nacharbeit .....	365
Nachfolger, direkte .....	46, 47
Nutzwertanalyse .....	196, 298

## O

### Objekt

Attribute .....	105
Beziehungen .....	100
Definition .....	35, 97
Hierarchie .....	101
Identifikation .....	104
Kapselung .....	156
Methoden .....	105
Modellierung .....	95
Spezifikation .....	105
Objektmodell .....	44, 100, 347
Definition .....	100
Objektorientierung .....	98

### Objektsicht

Definition .....	35, 97
Objektzugriffsdiagramm .....	151
Objektzugriffsdiagramm .....	271
Operationen .....	33, 56
Organigramm .....	229
Organisation, prozessorientierte .....	131
Organisationsentwicklung .....	173
Organisationsmodell .....	45, 117, 229
Definition .....	118

## P

Parallellauf .....	362
Partizipation .....	183, 184, 195, 409
Pflichtenheft .....	281
Ausprägungen .....	287
funktionale Beschreibung .....	285
Inhalt .....	284

Kritik .....	286
Raster .....	296
Phasenmanagement .....	395
Phasenmodell .....	170
Potenzial .....	218
Problemlösungszyklus .....	190
Produktinnovation .....	209
Produktionsplanung und -steuerung	
Erfolgsfaktoren .....	360
Probleme .....	168
Wirtschaftlichkeit .....	168
Produktivstart .....	359
Projekt .....	385
Projektabwicklung	
Methoden .....	169
Modellierungsmethoden .....	189
Probleme .....	166
Vorgehensmodelle .....	165, 181
Zielerfüllung .....	168
Projektantrag .....	203, 219
Projektcontrolling .....	405
Projekterfolg .....	389
Projektkategorien .....	386
Projektleiter .....	390
Projektmanagement .....	383, 387, 389, 402
Projektmanagementprozess .....	401
Projektorganisation .....	407
Projektplan .....	402
Projektplanungstools .....	389
Projektstrukturmanagement .....	392
Projektteam .....	402
Projektzielsetzung .....	207
Prototyp .....	177, 344

### Prozess

Beziehungen .....	63
Charakterisierung .....	67
Definition .....	35, 54
Detailkonzept .....	328
end-to-end .....	64, 66
Geschäftsprozess .....	58
Hauptprozess .....	58
Hierarchie .....	58
Hilfsmittel .....	67
Identifikation .....	62

In- und Outputs .....	63, 67
Managementprozess .....	61
Messgrößen .....	66, 67
Modellierung .....	53
Parallelitäten .....	64
Regeln .....	65
Segmentierung .....	60
Spezifikation.....	64, 262
Supportprozess .....	61
Teilprozess .....	59
Vielfalt .....	56
Wertschöpfungskette .....	64
Prozessinnovation .....	209
Prozesslandkarte .....	61, 75, 228, 239, 375
Prozessmodell .....	42, 58, 328
Definition.....	58
Prozessowner .....	67
<i>Prozesssicht</i> .....	35, 54
Prozesstechnologiediagramm.....	149

## Q

Qualifizierung .....	267
Qualität.....	66
Qualitätskontrolle.....	173
Qualitätsmanagement .....	400

## R

Rational Unified Process (RUP) .....	110
Realisierungsvorbereitung.....	402
Regelwissen.....	65
Rendite .....	214
Reports.....	331, 396
Requirements Prototyping .....	184
Ressourcenmanagement .....	396
Risiken .....	218
Risikoanalyse.....	218
Risikomanagement .....	399

## S

SADT.....	92
Schlüsseltechnologie.....	217
Schrittmachertechnologie.....	217
Schulung.....	360, 389
Schwankungen und Störungen .....	230

Segmentierung .....	60
Selbstregulation .....	267
Sequenzdiagramm.....	42, 69, 348
Definition .....	69
Sichtweise .....	46
Sichtweise .....	39
Sinnhaftigkeit .....	268
Situationsanalyse .....	191
Software Krise .....	169
Software Lifecycle.....	169
Softwarekosten .....	396
Software-Lizenzvertrag.....	311
Sozio-technisches System .....	236
Spezifikation	
Aufgaben.....	120, 121
Funktionen .....	86
Komplettheitsanspruch .....	89
Objekte .....	105
Prozesse.....	64
Technik.....	144
Spiralmodell	
Vor- und Nachteile.....	185
Stammdaten.....	359
Stärken und Schwächen .....	233
<i>Stelle</i> .....	35
Stellenbeschreibung.....	120
Stellenfunktionendiagramm .....	40, 45, 124
Definition .....	124
Stellenorientierter Informationsfluss .....	45, 122
Definition .....	122
Stellenorientiertes Ablaufdiagramm .....	42, 70, 325
Definition .....	70
Stichtagsumstellung.....	362
Strukturierte Analyse.....	92
Strukturierter Text .....	86
Subprozess.....	54
Support und Wartungsvertrag .....	314
System	
Analyse .....	30
Aspektsysteme .....	35
aspektweise Analyse .....	33
aspektweise Modellierung .....	34
Hierarchie.....	32

sozio-technisches .....	33
Systembegriff .....	26
Systemabgrenzung .....	30, 83, 227
aufgabenorientierte .....	119
funktionsorientierte .....	85
objektorientierte .....	104
prozessorientierte .....	62
technikorientierte .....	143
Systemablösung .....	232
Systemanalyse .....	345
Systemanalyse, sozio-technische .....	127
Systemanforderungen .....	251
Systemarchitektur .....	231, 292
technische .....	333
Systemarchitekturmodell .....	140
Systemdesign .....	346
Systeme	
relativ isolierte .....	30
sozio-technische .....	27
Systemelemente .....	36
Systemfunktionendiagramm .....	146
Systemniveau .....	54
Systemnutzungsdiagramm .....	148
Systems Engineering .....	26, 170
Systemschnittstellen .....	145
Systemschnittstellendiagramm .....	145, 270
Definition .....	145
Systemspezifikation .....	40

## T

Tätigkeitsanalyse .....	121
Tätigkeitsdiagramm .....	110
Taylorismus .....	130, 236
Teamzusammensetzung .....	409
Technik .....	137
Schnittstellen .....	145
Technikmodell .....	140
Techniksysteme .....	138
Technikeinsatzdiagramm .....	152, 242
Technikgestützte Arbeitssysteme .....	236
Techniksticht .....	138
Technologiemanagement .....	217
Teilprojekte .....	393
Teilprozess .....	54, 256

Terminatoren .....	88
Terminmanagement .....	395
Test .....	352
Top-down .....	47, 83, 84, 104, 143
Topfunktion .....	89
Transparenz .....	208

## U

Umfeldanalyse .....	227
UML .....	14, 101, 109, 110, 187
Umsystem .....	63
Unified Modeling Language (UML) .....	110
Untersysteme .....	32
Use-Case-Diagramm .....	43, 90
Use-Cases .....	40
User Interface Prototyping .....	184

## V

Vererbung	
Aufgaben .....	118
Objekte .....	102
Prozesse .....	60
Verifikation .....	305, 307
Vertrag .....	305, 307
Vertragsmanagement .....	397
Vollständiges Modell .....	159
Vorgehensmodelle .....	163

## W

Wartbarkeit .....	292
Wasserfallmodell .....	165, 172
Vor- und Nachteile .....	177
Werkvertrag .....	312
Wiederholhäufigkeit .....	66
Wirtschaftlichkeit .....	168, 211
Wirtschaftlichkeitsverfahren .....	214
Workflow .....	154

## Y

Yourdon .....	92
---------------	----

## Z

Zielformulierung .....	193
Zielsetzungen .....	290



---

Zielsetzungsmanagement.....	392	Zukunftswerkstätten .....	238
Zukunftsanalyse .....	234	Zustandsübergangsdiagramm .....	44, 107
Zukunftstechnologie .....	217	Zyklische Optimierung.....	184

Weitere aktuelle vdf-Publikationen finden Sie in unseren **Fachbereichsprospekten**:



- Bauwesen
- Naturwissenschaften, Umwelt und Technik
- Wirtschaft
- Geistes- und Sozialwissenschaften, Interdisziplinäres, Militärwissenschaft, Politik, Recht
- Informatik, Wirtschaftsinformatik, Mathematik

Bücher als Websites und mehr...

**vdf**-online.ch

- unser aktuelles Angebot von Büchern als Websites
- BEN-Blog zum Thema Bau-Energie-Nachhaltigkeit

**vdf**

Hochschulverlag AG  
an der ETH Zürich



NEWSLETTER ABONNIEREN

Dieses Methodenhandbuch ist ein unverzichtbarer, praxisgerechter Leitfaden zur Modellierung und Einführung von Informationssystemen. Er richtet sich insbesondere an Informatik-Projektleiter, Studenten, Wirtschaftsinformatiker und IT-Berater.

### **Modellierung**

In Informatikprojekten stellt sich immer die Frage nach dem optimalen Einsatz von Modellierungs- und Spezifikationsmethoden. Obwohl eine Vielzahl von Methoden propagiert wird – beispielsweise zur Modellierung von Geschäftsprozessen oder Objekten –, kann noch immer eine hohe Unsicherheit festgestellt werden, wenn es im konkreten Praxisfall darum geht, die geeigneten Methoden auszuwählen.

Das Buch gibt erstmals einen umfassenden und ausgewogenen Überblick über alle verbreiteten Modellierungstechniken für Informationssysteme. Im Zentrum steht ein methodischer Rahmen, der die bislang konkurrierenden Methoden der geschäftsprozess-, funktions-, objekt- und aufgabenorientierten Modellierung nahtlos integriert. Die im Verlauf der Jahre entwickelten Methoden (z.B. ARIS, Structured Analysis, UML/Rational Unified Process, MTO-Ansatz) lassen sich auf dieser Basis zu einem tieferen Verständnis verschmelzen. Es gelingt, die gängigen „Modelbreaks“ zwischen „prozessorientierten Beratern“ und „objektorientierten Informatikern“ zu überwinden.

### **Projektabwicklung**

Im zweiten Teil des Buches werden die Vorgehensmodelle zur IT-Projektabwicklung beschrieben: das Wasserfall- und Spiralmodell. Es wird im Sinne eines praxisgerechten Leitfadens aufgezeigt, welche Tätigkeitsschritte ein IT-Projekt zu durchlaufen hat und welche Modellierungsmethoden in welcher Detaillierung zur Anwendung gelangen sollten.

Die 2. Auflage dieses Buches führt den Leser anhand von vielen Praxisbeispielen und praktischen Tabellen durch Projektantrag, Grobkonzept, Konzept, Pflichtenheft, Evaluation, Vertrag, Detailkonzept, Systemdesign, Implementation, Abnahme, Inbetriebnahme und Betrieb.

### **Deutsche Fachzeitschrift „Wirtschaftsinformatik“:**

„Das Buch eignet sich gut als Grundlagenliteratur für Studenten und Praktiker, die sich einen Überblick über die verschiedenen Methoden der Informationsmodellierung verschaffen wollen (...).“