



FAKULTÄT FÜR **INFORMATIK**

OPTIMIERTE RISIKOSTEUERUNG
BEI ERP-ENTWICKLUNGEN
MITTELS ITIL CHANGE MANAGEMENT

Diplomarbeit

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Siegfried Zeilinger

Matrikelnummer 9926306

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung:

Betreuer: Univ.-Prof. Dipl.-Ing. Dr. Thomas Grechenig

Wien, 15. November 2008

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Technische Universität Wien

A-1040 Wien

Karlsplatz 13

Tel. +43/(0)1/58801-0

<http://www.tuwien.ac.at>



OPTIMIERTE RISIKOSTEUERUNG BEI ERP-ENTWICKLUNGEN MITTELS ITIL CHANGE MANAGEMENT

Diplomarbeit

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Siegfried Zeilinger

Matrikelnummer 9926306

ausgeführt am

Institut für Rechnergestützte Automation

Forschungsgruppe Industrial Software

der Fakultät für Informatik der Technischen Universität Wien

Betreuung:

Betreuer: Univ.-Prof. Dipl.-Ing. Dr. Thomas Grechenig

Wien, 15. November 2008

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 15. November 2008

Siegfried Zeilinger

Danksagung

Ich bedanke mich bei meinen Eltern für ihre Unterstützung während meiner Ausbildung sowie bei allen, die zur Entstehung dieser Arbeit beigetragen haben.

Kurzfassung

Durch die fortschreitende Globalisierung sowie das Outsourcing von Geschäftsbereichen von Unternehmen ist die Standardisierung der internen Prozesse und die effektive Kontrolle mittlerweile zu einem Muss geworden. Corporate Governance, ein Begriff, der nicht so einfach übersetzt werden kann, wird zum Schlagwort und zum Lösungsansatz für viele Probleme erklärt.

Was steckt nun hinter diesem Schlagwort? Erhoffen sich Unternehmen dadurch eine vereinfachte Kontrolle der internen Strukturen, eine „Nachvollziehbarkeit“ ähnlich den Grundsätzen guter Buchhaltung?

Aus diesem weitläufigen Thema soll im Bereich dieser Arbeit zunächst der IT Bereich ausgegliedert werden. Dieser beschäftigt sich u.a. mit der Entwicklung, der Anpassung, der Wartung und dem Betreiben von ERP Software. ERP Software selbst dient der Datenverarbeitung der Daten, die bei der Ausübung der gewöhnlichen Geschäftstätigkeit anfallen. Aufgrund der Fülle und der Sensibilität dieser Daten müssen ERP Systeme gewissen Kriterien genügen. Die Umsetzung der Governance Regeln muss sich an diesen Kriterien orientieren.

Die Arbeit befasst sich mit dem Software Lifecycle von ERP Systemen und den Regeln der Governance. Beim Change Management geht es darum, Software zu warten bzw. auf eine neue Release zu bringen. Für diesen Prozess gibt es im britischen ITIL Standard Vorgaben, die in ERP Lösungen bereits integriert wurden.

Ziel der Arbeit ist es, diese Ansätze für Governance im Bereich Change Management zu beleuchten. Anhand eines praktischen Beispiels in SAP wird untersucht, wie weit aktuell erhältliche Change Management Tools für eine gute IT Governance brauchbar sind. Durch dieses praktische Beispiel kann als Resultat eine Einschätzung mitgegeben werden, wie viel IT Governance Unternehmen verschiedener Größe einsetzen sollten.

Die entwickelte Modellbildung erklärt ausführlich, welche Teile von ITIL für kleine, mittlere, große und Konzernunternehmen relevant sind. Diese Aussagen sind besonders wichtig, weil das IT Governance Umfeld noch immer zu einem der schwierigsten und weitläufigsten interdisziplinären Bereiche zwischen Informationstechnologie, Recht und Sozialwissenschaften zählt.

Abstract

Given proceeding globalisation and the trend to perform outsourcing of complete business areas of companies, the standardisation of company-internal processes as well as effective measures of control have become a must. Corporate governance has become the keyword and has been declared the solution of choice to many problems.

What lies behind the term corporate governance? Do companies head for simplified control over internal structures, like the „traceability“ known from accounting standards? From this broad area of interest, this thesis will extract the issue of information technology. Among others, IT deals with the development, the adoption, the maintenance and operation of ERP software. ERP software itself is used for processing all the data that are generated out of a business operation. Because this usually represents a lot of data which might as well be sensitive information, ERP systems have to fulfill certain criteria. The realisation of governance rules has to be adopted to these criteria.

This piece of work will mainly treat the topic of the Software Lifecycle around ERP systems and the rules for governance. In this context, Change Management means software maintenance and release adoption. For this process, the British ITIL standard provides some guidelines which have already been integrated into ERP solutions.

The goal of this thesis is to shed some light on the rudiments of governance within the area of Change Management. Investigated by a practical example in SAP it is the aim to find out, how useful Change Management solutions can actually be for ensuring proper IT governance. Based on this practical example, as a result, estimations can be provided on how much IT governance is useful for companies of different sizes.

The model developed explains in detail, which parts of the ITIL Library are relevant to small, medium, large enterprises as well as for corporations. The propositions within this model are very important since the environment around IT governance has been one of the most complex and most ample interdisciplinary fields linking IT, law and social sciences.

Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Abkürzungsverzeichnis	xi
1 Einleitung	1
1.1 Ausgangspunkt und Problemstellung	1
1.2 Ziel der Arbeit	3
1.3 Struktur	4
2 Grundlagen von Enterprise Resource Planning	7
2.1 Der Begriff Enterprise Resource Planning	7
2.1.1 Einführung	7
2.1.2 Von der klassischen Host Anwendung zu integrierten ERP Systemen	7
2.1.3 ERP 2	9
2.1.4 ERP Systeme und die Zukunft	10
2.2 Abriss der Architektur von ERP Systemen	11
2.2.1 Architekturelle Einführung	11
2.2.2 Client / Server	12
2.2.3 Schichtarchitektur	13
2.2.4 Bestandteile eines ERP Systems	14
2.2.5 Schnittstellen und Kommunikation nach außen	15
2.2.6 Webservices und SOA	16
2.2.7 Multiplexing	16
2.2.8 Abbildungen	18
2.3 Systemlandschaften in Großunternehmen	19
2.3.1 Begriff der Systemlandschaft	19
2.3.2 Evolution der IT Infrastruktur	20
2.3.3 Typische Systemlandschaften von Referenzunternehmen	22
2.3.4 Planung von hoch integrierten Systemlandschaften	24

3	ERP Software Lifecycle Management	29
3.1	Begriff und Definition	29
3.2	Besonderheiten im Lifecycle von ERP Standardsoftware	33
3.2.1	Änderungs- und Programmierkonzepte bei Standardsoftware	34
3.2.2	Auswirkungen und Zusammenfassung	36
3.3	Prozess der Anpassung	36
3.4	Schnittstellen zum Projektmanagement	38
4	Corporate Governance versus IT Governance	40
4.1	Zum Begriff der Corporate Governance	40
4.1.1	Definition	40
4.1.2	Corporate Governance Framework	40
4.1.3	Compliance	42
4.1.4	Sarbanes-Oxley Act	42
4.1.5	8. EU Richtlinie	43
4.1.6	Der Österreichische Corporate Governance Kodex	44
4.1.7	Zusammenfassung	45
4.2	Zum Begriff der IT-Governance	46
4.2.1	Definition	46
4.2.2	IT Governance Framework	48
4.2.3	Frameworks	49
4.2.4	Einführung und Zertifizierung	52
4.3	Zusammenhang zwischen Corporate Governance und IT Governance	54
5	IT Infrastructure Library	57
5.1	Definition und Einführung	57
5.1.1	Zielsetzung	57
5.1.2	Teilbereiche	57
5.1.3	Zertifizierungen	59
5.2	Generisches Prozessmodell	62
5.3	Grundbegriffe	63
5.4	Prozess Change Management	67
5.4.1	Aufgaben und Ziele von Change Management	67
5.4.2	Das Change Management Umfeld	67
5.4.3	Fahrplan für die Einführung von Change Management	71
5.4.4	Beteiligte Rollen	74
5.4.5	Der Prozess im Detail	76

5.4.6	Prozessindikatoren	81
6	Change Management im SAP Umfeld	84
6.1	Systemarchitektur	84
6.1.1	Programmiersprachen	84
6.1.2	Systemkomponenten	86
6.1.3	Mandantenkonzept	89
6.1.4	Entwicklungsumgebungen	90
6.2	Das Korrektur- und Transportwesen	92
6.2.1	Architektur des TMS	92
6.2.2	Einrichtung	93
6.2.3	Prozess	94
6.2.4	Transporte in Systemlandschaften	95
6.2.5	Ausblick: CTS+	95
6.3	Change Management mit dem SAP Solution Manager 4.0	96
6.3.1	Allgemeines	96
6.3.2	Der Prozess im Detail	97
7	Analyse und Reflexion	100
7.1	Korrektur- und Transportwesen und Solution Manager - ein Vergleich . .	100
7.2	Risiken beim Einsatz der SAP Softwarelogistik	101
7.3	SAP Change Request Management und ITIL	102
7.4	Kosten und erwarteter Nutzen	103
8	Praxisbeispiel	105
8.1	Zu den beteiligten Firmen	105
8.2	Ausgangssituation	105
8.3	Potentiale	108
8.4	Kosten- und Nutzenabwägung	109
8.5	Prototyp	109
8.6	Ergebnisse	110
8.7	Bedeutung für andere Unternehmen	111
8.8	Ableitung eines Modells	112
9	Zusammenfassung und Ausblick	116
9.1	Zusammenfassung	116
9.2	Ausblick	118

Inhaltsverzeichnis

9.3 Schlussbemerkung des Autors	119
Literaturverzeichnis	120

Abbildungsverzeichnis

1.1	Kapitelaufbau mit Darstellung von Theorie/Praxis Schwerpunkt	4
2.1	Input/Output Prozess	8
2.2	softwareseitiges Client/Server Prinzip	12
2.3	hardwareseitiges Client/Server Prinzip	13
2.4	3-Schichtmodell	15
2.5	Multiplexer Funktionsweise	18
2.6	Dispatcher Architektur	18
2.7	Zusammenhang zwischen Landschaft, System und Server	19
2.8	Zwei Systeme, eine erste „Systemlandschaft“	20
2.9	Erweiterte Systemlandschaften mit Kommunikations- und Änderungspfaden	21
2.10	Wartungs- und Releasetracks in einer Systemlandschaft	23
2.11	Globale Systemlandschaft	24
3.1	Software Lifecycle	29
3.2	Technik des Customizing	34
3.3	Technik der Erweiterung von Standardcode mit eigenem Code	35
3.4	Typischer Prozess der Anpassung von ERP Standardsoftware	37
3.5	Informationsaustausch zwischen Projektmanagement und dem Entwicklungsprozess	38
4.1	Corporate Governance Framework	41
4.2	IT Governance Lifecycle	48
4.3	IT Governance Framework	49
5.1	ITIL Core	58
5.2	Generisches Prozessmodell von ITIL	63
5.3	ITIL Change Management und sein ITIL Prozessumfeld	68
5.4	ITIL Fahrplan	71
5.5	Prozessablauf eines Normal Change	78

Abbildungsverzeichnis

5.6	Change Authorization Model	79
5.7	Emergency Change	80
6.1	Schema einer interpretierten Programmiersprache	85
6.2	Darstellung der Komponenten des ABAP Applikationsservers	87
6.3	Darstellung der Komponenten des Java Applikationsservers	87
6.4	Vereinfachte Darstellung des SAP Web Dispatcher	89
6.5	Screenshot: Transportwege modellieren in der Transaktion STMS	93
6.6	Der Prozess von der Erstellung des Transportauftrags bis zum Import im Zielsystem	94
6.7	Der Change Request Management Prozess	98
8.1	Die Projektzuordnung als Kundenentwicklung. Ein Beispiel.	107

Abkürzungsverzeichnis

ABAP bzw. ABAP/4	Früher: Allgemeiner Berichtsaufbereitungsprozessor. Heute: Advanced Business Application Programming Language
ANSI	American National Standards Institute
ATX	Austrian Traded Index
BSI	British Standards Institution
CA	Change Authority, zu Deutsch: Änderungsgremium
CAB	Change Advisory Board, zu Deutsch: Änderungs-Beirat
CCC	Customer Competence Center
CeBit	Centrum für Büroautomation, Informationstechnologie und Telekommunikation
CI	Configuration Item, zu Deutsch: Konfiguration
CIO	Chief Information Officer, zu Deutsch: Leiter der IT Abteilung
CMDB	Configuration Management Database, zu Deutsch: Konfigurationsdatenbank
CMM	Capability Maturity Model
CMS	Change Management System
CO	Compliance Officer
COBIT	Control Objectives for Information and related Technology, zu Deutsch: Kontrollziele für die IT und verwandte Technologien
CS	Central Services, zu Deutsch: Zentralkomponenten
CTS	Change and Transport System, zu Deutsch: das Änderungs- und Transportmanagement
DEV	development system, zu Deutsch: Entwicklungssystem
Dynpro	dynamisches Programm
EAI	Enterprise Application Integration, zu Deutsch: Integration von Geschäftsanwendungen
ECAB	Emergency Change Advisory Board, zu Deutsch: Notfallbeirat für Änderungen
ERP	Enterprise Resource Planning

Abkürzungsverzeichnis

ESS	Employee Self Service
EuroSOX	Richtlinie 2006/43/EG zur Abschlussprüfung in Jahresabschlüssen
GesRÄG	Gesellschaftsrechts- Änderungsgesetz
HA	high availability, zu Deutsch: Hochverfügbarkeit
HR	Human Resources, zu Deutsch: Personalwirtschaft
ICM	Internet Communication Manager
IEEE	Institute of Electrical and Electronics Engineers
IMS	Information Management System
IS	Informationssystem
ISMS	Information Security Management System
IT	Informationstechnologie, vgl. Englisch information technology
ITEC	IT Executive Comitee
ITIL	IT Infrastructure Library
JCO	SAP Java Connector
JEE	Java Enterprise Edition
JSE	Java Standard Edition
KPI	Key Process Indicator, zu Deutsch: kritischer Prozessindikator
KTW	Korrektur- und Transportwesen
MRP	Material Requirement Planning
MVC	Model View Controller
NWA	Netweaver Administrator
OGC	Office of Government Commerce
PIR	Post Implementation Review, zu Deutsch: Implementierungsbericht
PM	Projektmanagement
POC	Proof of Concept zu Deutsch: Beweis des Konzepts
PRD	production system, zu Deutsch: Produktivsystem
PSO	Planned Service Outage, zu Deutsch: geplante Serviceunterbrechungen
QAS	quality assurance, zu Deutsch: Qualitätssicherungssystem (Testsystem)
RFC	Remote Function Call (SAP) bzw. Request for Change (ITIL)
ROI	Return of Investment
SAP	Systeme, Anwendungen, Programme in der Datenverarbeitung
SAS	Statement on Auditing Standards

Abkürzungsverzeichnis

SD	Sales and Distribution, zu Deutsch kurz: Vertrieb(smodul)
SLA	Service Level Agreement
SLD	System Landscape Directory
SMART	Simple, Measurable, Achievable, Realistic, Time-Driven
SOA	Serviceorientierte Architektur
SOAP	Simple object access protocol
SolMan	SAP Solution Manager
SOX	Sarbanes-Oxley-Act
SPOC	Single Point of Contact, zu Deutsch: zentrale Kontaktstelle
SQL	Structured Query Language
TMS	Transportmanagementsystem
URL	Uniform Resource Locator
WWW	World Wide Web
XML	eXtended Markup Language
ZBV	Zentrale Benutzerverwaltung
ÖCGK	Österreichischer Corporate Governance Kodex

Sämtliche Übersetzungen aus englischsprachiger Literatur stammen vom Autor.

Die Arbeit wurde gänzlich mit L^AT_EX und OpenOffice produziert.

1 Einleitung

1.1 Ausgangspunkt und Problemstellung

Die Umwandlung von einer Industrie- zu einer Dienstleistungs- und nunmehr Wissensgesellschaft macht auch vor der Branche der Datenverarbeitung nicht Halt. Eine Weiterentwicklung von einer Produktionsunterstützung zu einem Dienstleistungsinstrument ist bereits durchlaufen worden. Immer neue Informationssilos, Verknüpfungen von Daten und Filterung nach verschiedensten Kriterien und bunte Auswertungen bestimmen über den Glanz oder den Niedergang von Unternehmen. Wissen, bisher versteckt als Nebenprodukt von Geschäftsabläufen, wird zutage gefördert. Der Physiker würde sagen, messen oder messbar machen¹.

Die Informationstechnologie ist keine reine Datenverarbeitung mehr. Früher galt es, anfallende Daten in einem Teilbereich des Unternehmens zu verarbeiten, um dort Arbeitsabläufe effizienter zu gestalten und Mitarbeiter einzusparen. Heute sind diese Informationen aus den einzelnen Teilbereichen und Abteilungen untereinander verknüpft, werden periodisch aufbereitet, mit der Konkurrenz und weiterem Datenmaterial verglichen und die Mitarbeiter in Führungspositionen danach entlohnt. Die IT ist allgegenwärtig.

Systeme und Systemverbünde bestehen aus vielen Einzelteilen, sowohl was die Hardware betrifft als auch hinsichtlich der Anzahl der ablaufenden Programme. Wird ein einzelner Satz verbucht, so werden Ereignisse ausgelöst, die weitere Buchungen zur Folge haben können. Die Daten eines Tages werden in der Nacht in ein Data Warehouse geladen und dort wirkt sich die Buchung wieder aus, z.B. auf das Ergebnis des letzten Tages, der letzten Woche usw.. Alle diese Zusammenhänge erfordern es, dass große Unternehmen sich in ausreichendem Maße damit auseinandersetzen, wie diese Systeme richtig betreut und gewartet werden. Dies setzt in einigen Bereichen sicher einen höheren Formalismus und eine größere Genauigkeit beim Testen von neuen Funktionen voraus, damit nicht einzelne Teile der Architektur die gesamte Architektur gefährden.

Getrieben vom Willen der Verbesserung versuchen sich viele Unternehmen in einer guten Governance. Eine solche Governance kann im Bereich der IT durch Einhaltung von

¹Frei nach Galileo Galilei.

1 Einleitung

Richtlinien wie ITIL dann auch zertifiziert werden. Es bleibt die Frage, ob eine Zertifizierung nach ITIL und ein sauberes Definieren von Change Requests es zu verhindern vermag, dass fehlerhafte Software auf einem produktiven, geschäftskritischen System eingespielt wird und die Frage, wie wahrscheinlich so etwas ist. Viele Unternehmen machen hier eigene Erfahrungen und genau diese können sich dann nicht mit dem von Prozess- und Revisionsberatern versprochenen Nutzen decken.

Ein oftmals vernachlässigter Aspekt ist hierbei auch, dass immer mehr Unternehmen Standardsoftware einsetzen, die über Einstellungen, auch Customizing genannt, an die Unternehmensprozesse angepasst wird. In diesem Fall ist ein Change in der Softwareauslieferung seitens des Herstellers noch kritischer zu sehen. Nachdem das Unternehmen nicht selbst an den Anpassungen programmiert hat, ist zunächst unklar, ob der Hersteller auch wirklich alle Aspekte des Customizings in diesem speziellen Fall berücksichtigt hat. Dazu kommen dann noch die Änderungen des Unternehmens selbst. Hier hat ein Change mit größter Vorsicht zu erfolgen.

Große Change Management Lösungen, wo jede einzelne Änderung an einem EDV System akribisch aufgelistet und in eigenen Workflows genehmigt werden muss, machen natürlich auch nur ab einer gewissen Unternehmensgröße Sinn. Bei kleineren Unternehmen verschwimmen oft die Grenzen zwischen Entwicklung und Test, oder es gibt gar keine eigene Testabteilung. Dann testen die Fachabteilungen neben ihrem täglichen Business. In solchen Unternehmen sind andere Konzepte vielleicht praktikabler, um eine Trennung von Test und der Freigabe von Änderungen zu erreichen. Eine allgemeingültige Lösung ist in Bezug auf eine Change Strategie für mittlere Unternehmen ist noch nicht gefunden. Dieses Thema scheint derzeit in der Literatur noch nicht hinreichend behandelt worden zu sein.

Gerade auf dem österreichischen Markt gibt es viele dieser größeren Unternehmen, welche gerade noch den kleinen und mittleren Unternehmen zugerechnet werden, für den lokalen Markt allerdings Großunternehmen sind. Global ein kleiner Player haben solche Unternehmen eher angespannte Budgets, müssen aber international mit großen Konzernen mithalten können. Hier gilt es eine Grundlage zu finden, auf der man bei einer derartigen Unternehmensgröße Entscheidungen über die Art und den Umfang von Sicherheits- und Kontrollmechanismen treffen kann.

In der Literatur, z.B. mit [Pop07] existieren sehr gute Arbeiten, welche die Theorie hinter Governance sehr verständlich aufbereiten. Die meisten Autoren gehen sehr wenig auf Kosten und Nutzen von solchen Maßnahmen ein. Sei es, dass es von größeren Unternehmen ohnehin erwartet wird, dass Governance Maßnahmen getroffen werden oder seien es gesetzliche Maßnahmen, die dazu führen (z.B. der Sarbanes-Oxley-Act, SOX in

1 Einleitung

den USA). Einige wenige praxisnähere Werke sparen den Bereich der Kosten/Nutzen Rechnung, des Return on Investment (ROI), eher aus, wie beispielsweise [Lie06].

Es ergibt sich zusammenfassend ein Bedarf vornehmlich in zwei Bereichen

- IT Governance für mittlere Unternehmen: Was ist ein nice-to-have Prozess bzw. Methode und was ist ein must?
- Betrachtungen zu Kosten und Nutzen der Einführung von ITIL konformen Prozessen in Unternehmen, die leicht abzuschätzen sind.

1.2 Ziel der Arbeit

Diese Arbeit zeigt, wie ERP Systeme funktionieren. Der Aufbau, die besonderen Aspekte von Standardsoftware und die genauen Details bei der Wartung von darauf ablaufenden Programmen mögen sich dem Leser erschließen. Speziell der Kontext von großen, miteinander vernetzten Systemlandschaften, deren reibungslosen Funktion für den laufenden Betrieb eines Unternehmens unabdingbar ist, bedarf des Einsatzes von besonderen Verfahren und Tools für ein effektives Change Management. Die Bedeutung von guter Governance, genauer gesagt von IT Governance, soll diskutiert werden und die Auswirkungen von Governance auf bisherige Wartungs- und Change Prozesse. Diese Bedeutung möge anhand eines zusätzlichen praktischen Beispiels noch besser erlebbar gemacht werden.

ITIL ist ein Konzept, welches sich in der IT Branche zunehmend als de-facto Standard durchsetzt. Die genaue Bedeutung von ITIL, die Bedeutung der einzelnen Tools und Methoden wird vorgestellt und im Bereich des Change Management genau behandelt. Speziell der genaue Ablauf von Änderungen an der System- und Softwarelandschaft durch ITIL konforme Change Requests ist hier von Interesse. Die verfügbaren Tools des führenden ERP Herstellers, SAP, sollen vorgestellt und verglichen werden.

Die Entstehung und Behandlung eines Change Requests in der Praxis mit der möglichen Annahme bzw. Ablehnung, der Durchführung inkl. Validierung und des Releaseprozesses sollen erläutert werden. Dazu wird sich die Arbeit eines Praxisbeispiels bedienen. In dieser wird eine Einführung von SAP Change Request Management durchgeführt. Dadurch soll aufgezeigt werden, wie hier in der Praxis vorgegangen wird. Diese Einführung erfolgt unter Aufzeichnung relevanter Faktoren wie z.B. Zeit- und Schulungsbedarf. Durch die Zusammenarbeit mit der IT Tochter einer großen Österreichischen Bank sollen auch Einblicke in Alternativen zur SAP-eigenen Lösung gewonnen werden. Das resultierende Delta soll verglichen werden, in Bezug auf die Zeit und den Nutzen, kurz gesagt auf den

1 Einleitung

ROI.

Unternehmen beliebiger Größe mögen durch diese Arbeit ein Mittel in die Hand bekommen, für sich zu entscheiden, ob Change Management nach ITIL für sie eine gute Wahl ist. Vielleicht sind ja Alternativen angezeigt. Letztlich darf der Leser zu einer weiterführenden Diskussion in den täglichen Unternehmensanforderungen animiert werden.

1.3 Struktur

Die Arbeit gliedert sich in theoretische und praktische Teile. In den ersten Kapiteln wird die nötige Theorie vertieft, mit zahlreichen Möglichkeiten zu weiterführender Lektüre. In den weiteren Kapiteln wird diese Theorie in einem Praxisbeispiel aus dem SAP Solution Manager verwendet und letztlich genau hinsichtlich des Aufwands analysiert.

Als Verbindung von Theorie und Praxis soll dann eine Abschätzung getroffen werden, ab welcher Unternehmensgröße ein Einsatz von Verfahren und Vorgehensweisen nach ITIL empfohlen werden kann. In Abbildung 1.1 wird die Aufteilung zwischen Theorie und Praxis der einzelnen Kapitel dieses Werkes veranschaulicht.

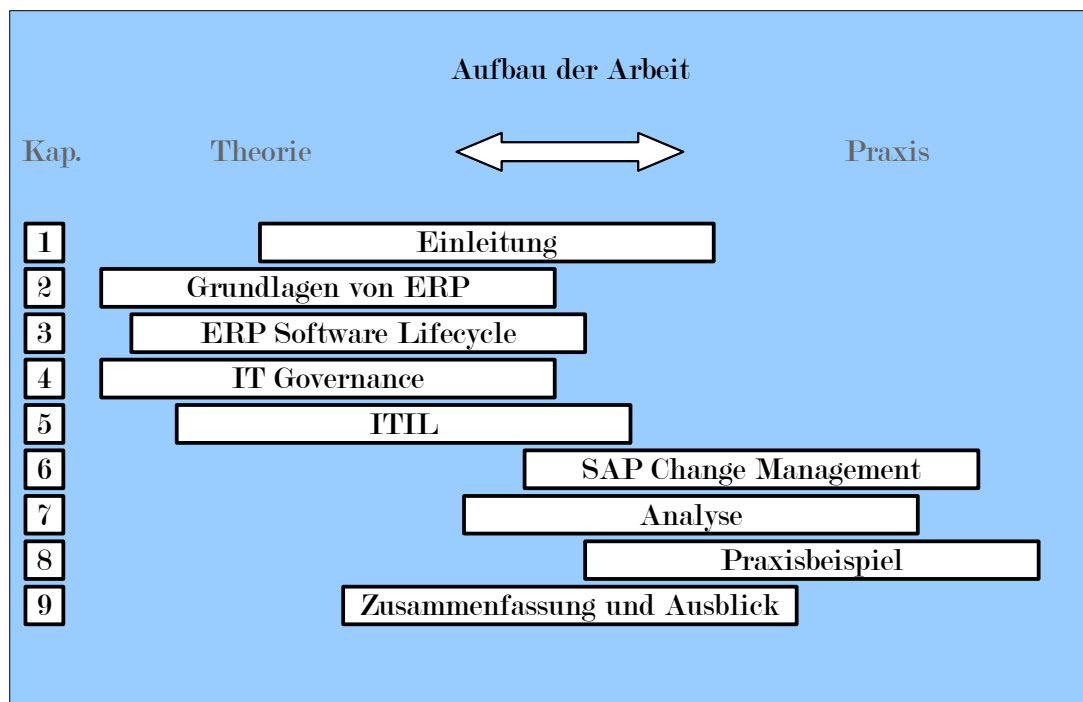


Abbildung 1.1: Kapitelaufbau mit Darstellung von Theorie/Praxis Schwerpunkt

Kapitel 2 geht auf die Grundlagen von ERP Systemen ein. Der Begriff wird mitsamt

1 Einleitung

seiner historischen Bedeutung und Evolution erläutert. Es wird ein Ausblick auf die Zukunft gegeben und aktuelle Trends vorgestellt. Architekturen und Systemlandschaften in Unternehmen runden das Kapitel ab und sollen dem Leser das notwendige Verständnis der Materie für die nachfolgenden Kapitel ermöglichen. Da das Verständnis von Change Management auch Kenntnisse über die genaue Architektur von ERP Systemen erfordert, gehen einige Teile des Kapitels sehr ins Detail. Der komplette Abschnitt der Grundlagen ist sehr theoretisch orientiert.

Kapitel 3 erläutert, wie ein Software Lifecycle bei ERP Systemen zu verstehen ist; dies unter Bezugnahme auf das zuvor vorgestellte Grundlagenwissen. Hier soll vor allem auf die Eigenheiten von Wartungs- und Erweiterungskonzepten bei betrieblicher Standardsoftware gegenüber Individualsoftware eingegangen werden, um letztlich auch die Aufgabe der verschiedenen Tools in diesem Bereich zu erkennen. Auch die Rollenverteilung innerhalb eines Unternehmens ist hier wichtig. Dieses Kapitel ist wiederum eher theoretischer Natur, mit gelegentlichen Bezügen auf konkrete Anforderungen in Unternehmen.

Kapitel 4 widmet sich der IT Governance und beschäftigt sich mit den Möglichkeiten und Pflichten von Unternehmen, IT Prozesse zu überwachen und effektiv zu gestalten. Es werden gute Standardwerke exzerpiert und das Verständnis für die Materie geschaffen.

Kapitel 5 erklärt den Begriff der ITIL Library in seiner Historie. Auch wird die Arbeit vom Umfang her so abgegrenzt, dass diese sich auf Change Management fokussieren kann. Die nötigen Grundbegriffe und -prozesse der ITIL Bibliothek werden vorgestellt und speziell der ITIL Prozess Change Management wird erklärt. Die darin vorkommenden Begriffe sollen den Leser über den gesamten Rest der Arbeit begleiten. Schnittstellen zu anderen Prozessen sind von entscheidender Bedeutung für das Verständnis des Begriffes des Change Requests und werden ebenfalls angerissen. Eine eventuelle Implementierung von Change Request Management und anderer Kernprozesse wird angeregt und diskutiert. Dieser Bereich geht auch stärker auf Praxisthemen ein.

Kapitel 6 behandelt die Umsetzung von Change Management Prozessen speziell auf SAP ERP Systemen. Die klassischen Transportprozesse zwischen Systemen sowie die neuen Prozesse des SAP Change Request Management werden vorgestellt. Auch neueste Entwicklungen, die Change Management über verteilte Architekturen behandeln (CTS +), werden erläutert. In diesem Teil dominiert ein stärkerer Praxisbezug.

Kapitel 7 (Analyse) präsentiert eine Abschätzung der Einsatztauglichkeit der vorgestellten SAP Change Management Tools. Es werden die klassischen Prozesse dem SAP Change Request Management gegenübergestellt und es wird eine erste Abwägung ge-

1 Einleitung

troffen.

Kapitel 8, das Praxisbeispiel, versucht Aufwände und auch den Nutzen zu schätzen, welche bei einer Einführung erwartet werden. Die getroffenen Überlegungen werden so dann mittels eines Prototypen verifiziert. Der Leser möge vom vorgestellten Beispiel und den damit einhergehenden Überlegungen Anregungen für die Situation in seinem bzw. in anderen Unternehmen erhalten. Diese Kapitel sind fast ausschließlich praktischer Natur. **Kapitel 9** fasst die Arbeit zusammen. Es wird diskutiert, wie der Begriff Return on Investment (ROI) im Falle von IT Governance zu sehen ist und wo bzw. wie denn der genaue Return liegt. Unter dem Titel Zusammenfassung und Ausblick versucht der Autor auch, Kritik am gezeigten Modell vorzubringen sowie weiterführende Ideen aufzuzeigen. Themen, die nicht in der Arbeit behandelt werden konnten oder nicht in entsprechender Ausführlichkeit, werden angeschnitten. Eine Verknüpfung von Theorie und Praxis wird vorgenommen.

Die einzelnen Kapitel der Arbeit sind grundsätzlich als voneinander abhängig und aufeinander aufbauend zu sehen. Sollte der Leser aber bereits über hinreichende Kenntnisse im SAP Transportwesen verfügen, so wäre ein Einstieg mit Kapitel 4 oder 5 zu überlegen. Bei Vorwissen in allen Bereichen empfiehlt sich eine Lektüre ab der Analyse in Kapitel 7.

Nach dem nunmehrigen Festlegen des Zieles der Arbeit, der Motivation und der Ausgangslage wird nun im folgenden Kapitel auf die Grundlagen von ERP Systemen eingegangen. Dieses Kapitel möge das Grundverständnis von ERP in historischer und aktueller Sicht begründen und dient als wichtige Grundlage für die weiteren Kapitel der Arbeit. Auch der Begriff der Systemlandschaft soll erläutert werden.

2 Grundlagen von Enterprise Resource Planning

2.1 Der Begriff Enterprise Resource Planning

2.1.1 Einführung

Der Begriff ERP, oder Enterprise Resource Planning, stellt den Anspruch, Eigenschaften einer Software zusammenzufassen, die der Verarbeitung der in Unternehmen - im oft so genannten Business Umfeld - anfallenden Daten ermöglicht.

Man kann auch sagen, dass ein ERP System nichts Anderes ist als ein Modell des Unternehmens, in dem sich Material- als auch Informationsflüsse wiederfinden ².

Die Abgrenzung des Begriffes lässt sich wohl am besten durch eine Betrachtung der Historie des Begriffes ERP bewerkstelligen, was im Folgenden versucht wird. Das Verständnis des Begriffes ERP soll im nächsten Kapitel die spezifischen Anforderungen an ERP Systemlandschaften begründen.

2.1.2 Von der klassischen Host Anwendung zu integrierten ERP Systemen

Letztlich ging es historisch in der elektronischen Datenverarbeitung in Unternehmen immer schon darum abzubilden, wie aus einem Input ein Output wird. Inputs lassen sich in Kapital, Arbeit (Humankapital) und Material (inkl. Fremdleistungen, Miete, etc.) unterteilen³, wie in Abbildung 2.1 ersichtlich.

Als Output versteht man je nach Branche des Unternehmens etwas anderes. Generell könnte man wohl jene Dinge hier nennen, die aus der Erfüllung des Unternehmenszwecks resultieren.

Historisch gesehen besitzen ERP Systeme Vorläufersysteme in einzelnen Satellitensystemen, wie z.B. Fertigungssystemen, die in den 1970er und 1980er Jahren in Unternehmen eingeführt wurden. Diese Systeme liefen unter Plattformen wie IMS auf Großrechnern⁴.

²Vgl. ausführlicher in [Hum04, S.34].

³Vgl. genauere Erläuterung in [Hum04, S.35].

⁴Ausführliche historische Erläuterung in [SAW00, S.93].



Abbildung 2.1: Input/Output Prozess

Aber auch MRP Systeme, die für die Materialplanung eingesetzt wurden, können als Vorläufer gesehen werden⁵. Ein durchschnittliches Unternehmen besaß mehrere solcher Satellitensysteme, die untereinander nicht bis kaum kommunizierten. Die Entwicklung an solchen Systemen geschah auf das Unternehmen zugeschnitten und für den spezifischen Zweck. Dies hatte hohe Kosten aber auch eine hohe Passgenauigkeit der Systeme zur Folge.

Historisch gesehen erkennt man einige Revolutionen. Eine immer bessere Rechnerleistung erlaubte eine Buchung von Daten in Echtzeit. Bei vielen Vorgängen in einem Betrieb löst eine Buchung weitere Buchungen aus, z.B. bei Zusammenhängen zwischen Ein-/Verkauf und der Lagerverwaltung. Es mussten Aktualisierungen in anderen Systemen nun nicht mehr in Schwachlastzeiten mit eigenen Nachtläufen dort abgeglichen werden. Diese konnten technisch sofort in andere Systembestandteile übernommen werden. Durch diese technischen Möglichkeiten kamen integrierte Systeme in Betracht, welche die Informationen, die bisher in Satellitensystemen verarbeitet wurden, zentral erfassten und zueinander in Beziehung brachten.

Auf das Beispiel bezogen: Statt einem System für Rechnungen und Finanzen und einem für die Lagerverwaltung kann man in einem integrierten System etwas ausliefern und direkt die Rechnung erstellen und den Lagerstand aktualisieren.

Nachdem integrierte Systeme natürlich aufwändiger zu programmieren waren, kamen die Hersteller dieser Systeme - allen voran die deutsche SAP - auf die Idee, Software als Standardsoftware zu programmieren, die - einstellbar - für möglichst viele Unternehmen passen würde. Dadurch ließen sich die Entwicklungskosten auf mehrere Firmen verteilen und eine größtmögliche Integration realisieren. Eines der ersten Systeme, welches über verschiedene Applikationsmodule solche Standardabbildungen zuließ, war R/2. Es besaß mehrere Module, die integriert und auf der gleichen Datenbasis betrieben werden konnten. Über umfangreiche Einstellungen, das Customizing, konnten Anpassungen für unterschiedlich ablaufende Unternehmensprozesse getroffen werden, ohne zu program-

⁵Zur Evolution vgl. [Gal05, S.44].

mieren.

Zum Erfolg der betriebswirtschaftlichen Standardsoftware wie sie heute verstanden wird und als ERP System haben einige Faktoren besonders beigetragen. Allen voran die Frage der Architektur.

Frühere Architekturen setzten auf den Einsatz von Großrechnern, welche in ihrer Kapazität limitiert waren und die im Wesentlichen nur durch Aufrüstung des Innenlebens (mehr Prozessoren, mehr Speicher, etc.) skaliert werden konnten. Die technische Evolution in der Leistung von Workstations (zunächst unter UNIX) machte es möglich, Rechenleistung kostengünstiger zur Verfügung zu stellen. Eine Verwendung von mehreren Workstations machte allerdings eine Parallelisierung von Aufgaben und eine Synchronisation zwischen den Geräten erforderlich.

Anstatt einem Großrechner, der für alles zuständig war, teilte man die Aufgaben in einzelne Komponenten:

- die Datenbank
- den Applikationscode und
- den Präsentationscode

(siehe auch Kapitel 2.2.4). Dies galt damals als schier unmöglich, da eine entsprechende Zuverlässigkeit zunächst nur dem Großrechner zugetraut wurde. Nach einer erfolgreichen Demonstration auf der CeBit Messe durch SAP entschieden sich aber mehr und mehr Unternehmen für eine solche Architektur, umgesetzt beispielsweise in SAP R/3⁶. Heute kann man sagen, dass sich solche Mehrschicht - Architekturen durchgesetzt haben und der Begriff Cluster - also ein Rechnerverbund - aus modernen IT Landschaften nicht mehr wegzudenken ist.

2.1.3 ERP 2

Mit dem Fortschreiten der Entwicklung an ERP Systemen schafften es die verschiedenen Hersteller in wenigen Jahren, sämtliche Branchen und auch Unternehmen des öffentlichen Sektors mit Standardsoftware zu versorgen. Es hatte also Mitte der 90er, Anfang der 2000er Jahre nahezu jedes größere Unternehmen eine Art von ERP Software im Einsatz. Was nun in diesem, dem oft auch als ERP1 bezeichneten, Umfeld fehlte war die Öffnung nach „außen“. Prozesse eines Unternehmens beginnen nicht unbedingt an der Eingangstür, sondern mitunter beim Geschäftsessen oder bei einer Internetrecherche durch den

⁶Genauer in [SAW00, S.98ff.] und nachfolgend in Kapitel 6.

Abnehmer und enden auch nicht beim Verlassen des Gebäudes sondern vielleicht erst dann, wenn der Kunde die Ware erhalten und für gut befunden hat (klassisches Fernabsatzgeschäft).

Für die Anforderung, ERP näher zu den Kunden und Lieferanten zu rücken, wurden Systeme konzipiert, die mit den klassischen Unternehmensgrenzen brachen.

- CRM Systeme, oder Customer Relationship Management, bringen den Kunden näher an das Unternehmen und unterstützen alle Arten von Verkauf, z.B. Internet-, Telefon- oder Außendienstverkauf.
- SCM Systeme, oder Supply Chain Management, unterstützen den Einkauf und können z.B. optimal zwischen Lieferanten wählen, unter Berücksichtigung von Terminen, Mengen und Preisen.
- Elektronische Marktplätze ermöglichen die direkte Interaktion von ERP Systemen zwischen Unternehmen (direktes Auffinden eines Lieferanten, Vertragsabwicklung und Verbuchung ohne einen Papierbeleg)

Alle diese Szenarien haben gemeinsam, dass sie sich wiederum in das klassische ERP integrieren und dieses offener machen und erweitern. Das Ziel einer völlig offenen Architektur in einem globalen Markt muss hier mit dem Risiko von Wettbewerbsnachteilen durch zu große Offenheit abgewogen werden, so findet sich bei sehr wenigen Unternehmen außerhalb der Einzelhandelsbranche, in einem Internetauftritt auch ein Webshop mit öffentlichen Preisen.

2.1.4 ERP Systeme und die Zukunft

Nach der Erweiterung von ERP um die ERP2 Komponenten sind die führenden Hersteller dazu übergegangen, die Systeme mehr und mehr von den ERP Fachbenutzern weg zu allen Mitarbeitern zu öffnen. Jeder Mitarbeiter soll seine Urlaube und Arbeitszeiten direkt im ERP System erfassen können, seine Bankverbindung ändern oder beispielsweise Bestellungen tätigen und Trainings buchen können. Dies erfordert einerseits Anpassungen in der Bedienbarkeit (intuitive Lösungen), andererseits aber auch die Öffnung der Netze. Von überall im Intranet soll die ERP und ERP2 Landschaft erreicht werden können. Um den Überblick nicht zu verlieren werden noch Portallösungen für den Zugriff auf die verschiedenen Services geboten, die nach Benutzerrollen unterscheiden können, mit welchen Applikationen der Benutzer versorgt werden soll.

Dies dient nicht zuletzt wieder einer Kostensenkung: Die Erfassung der Daten wird von der jeweiligen Abteilung (z.B. der Personalabteilung) zum Mitarbeiter verlagert,

die Fachabteilung muss lediglich eine Prüfung durchführen. Genauso verhält es sich mit Portallösungen: Je weniger der Mitarbeiter geschult werden muss, um zu seinen Applikationen zu kommen und mit ihnen zu arbeiten, desto größer ist der Nutzen für das Unternehmen.

Aber auch vor dem Internet machen die ERP Hersteller nicht halt. Galt es früher als hohes Sicherheitsrisiko, ERP Daten als wichtigste Unternehmensdaten nicht nach außen zu tragen, so gibt es nun die Vernetzung der Systeme über elektronische Marktplätze, den Bestell- und Verkaufsvorgang als XML Stream oder Web Service zwischen den Geschäftspartnern. Grundlage dafür sind Integrationsumgebungen (Enterprise Application Integration, EAI), die XML Dokumente an die richtigen Empfänger versenden und zustellen können). Es gilt auch als schick, den Lieferanten die Rückmeldung über die Qualität der gelieferten Produkte online rückzumelden oder auch Daten über den Absatz gelieferter Fertigwaren, analytisch aufbereitet, dem Lieferanten zurück zu verkaufen.

Eine totale Vernetzung bahnt sich an und wird als Service Architektur in den Fachbüchern gelistet. Eine Vielzahl an Services wird von innerhalb und außerhalb eines Unternehmens konsumiert um neue Applikationen zu schaffen, die keine Systemgrenzen mehr kennen und rein prozessorientiert ablaufen.

Eine weitere Neuerung scheint der Trend zu besserem Design und Usability zu sein. ERP Systeme haben aufgrund der starren Strukturen und der Vielzahl an Daten, die erfasst werden können (bzw. müssen) einen „altmodischen“ Touch. Der krasse Gegensatz zu den anderen täglich verwendeten Büroapplikationen hat führende Hersteller dazu ermutigt, die Systeme bedienfreundlicher zu gestalten. Hier darf man sich auch in Zukunft noch weitere Neuerungen erwarten.

2.2 Abriss der Architektur von ERP Systemen

2.2.1 Architekturelle Einführung

Ein ERP System unterscheidet sich in einigen Dingen wesentlich von einem z.B. wissenschaftlichen System oder einem Einzelplatzcomputer. Zunächst dadurch, dass ein größerer Benutzerkreis, oft das gesamte Unternehmen, am gleichen Datenbestand arbeitet.

Ein zentraler Datenbestand ermöglicht eine weitgehende Automation von Prozessschritten, da die gleichen Daten von allen Applikationsbestandteilen zugegriffen werden können⁷.

⁷[Gal05, S.40] erläutert ERP gänzlich am Beispiel des zentralen Datenbestandes.

Dieser Datenbestand wird in einzelnen Transaktionen verarbeitet. Eine Transaktion versetzt hier das System von einem betriebswirtschaftlich konsistenten Zustand in einen anderen. Das kann man als abgeschlossenen Prozessschritt verstehen. Im Gegensatz zu einem wissenschaftlichen System spielt nicht die massenhafte Analyse von verschiedenen Daten eine Rolle, sondern in aller Regel eine systematische Erfassung nach den gleichen Schemata. Der Unterschied zum Einzelplatzrechner besteht in der Vernetzung und der gleichzeitigen Benutzung der (zentralen) Ressourcen untereinander.

Im Folgenden soll näher auf signifikante Details von Client/Server Architekturen eingegangen werden, dem häufigsten Architekturmodell von ERP Systemen. Im weiteren Verlauf wird im Kapitel 6 auch noch auf die spezifische Implementierung der Programmierer von SAP eingegangen.

2.2.2 Client / Server

Das Prinzip von Client/Server Architekturen ist aus der modernen IT Welt nicht mehr wegzudenken. Jedermann hat heutzutage beim Zugriff auf eine Internetseite mit einem Server zu tun, der Informationen im WWW bereitstellt.

Der Begriff Client/Server kann in zweierlei Bedeutung auftreten. Generell versteht man unter Client/Server aber das Prinzip, dass ein Server eine Dienstleistung für einen Klienten erbringt.

Softwareorientiertes Client/Server Prinzip

Unter einem softwareseitigem Client/Server Modell versteht man, dass auf einem Rechner mehrere Programme laufen, von denen eines einem anderen eine Dienstleistung erbringt, siehe Abbildung 2.2.

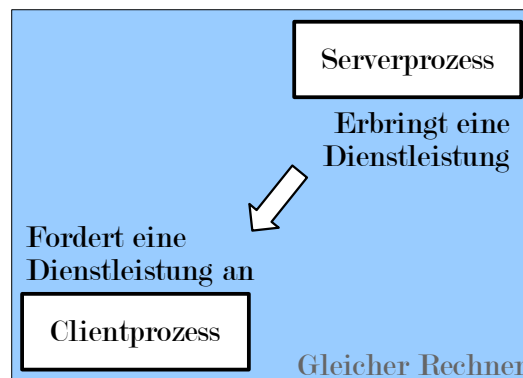


Abbildung 2.2: softwareseitiges Client/Server Prinzip

Zum Beispiel ein lokal laufender Webserver, der einem lokal laufenden Browser eine Webseite zur Anzeige übermittelt. Man kann aber auch innerhalb eines Programms mit Servern und Klienten arbeiten, vor allem in objektorientierten Programmiersprachen, wenn ein Objekt Methoden eines anderen, parallel ablaufenden Prozesses abfragt.

Hardwareorientiertes Client/Server Prinzip

Beim hardwareorientierten Client/Server Prinzip verhält es sich ähnlich wie bei den Softwareprozessen. Hier läuft der Serverprozess aber auf einem anderen Rechner ab als der Clientprozess. Dies wird in Abbildung 2.3 gezeigt.

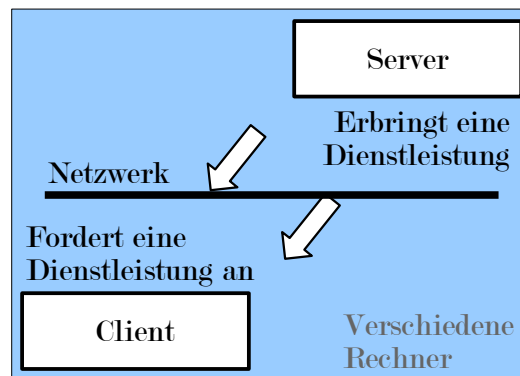


Abbildung 2.3: hardwareseitiges Client/Server Prinzip

Dies ist also das Prinzip, welches beim zuvor zitierten Beispiel (Webserver) Anwendung findet.

In modernen Architekturen spielt gerade diese, die verteilte, Eigenschaft von Architekturen eine zentrale Rolle. Aber sie bringt natürlich auch Probleme mit sich, wie z.B. die Sicherheit der Übertragung, die Warteschlangenmechanismen wenn ein Server nicht verfügbar ist usw..

Ein modernes EDV System, wie auch ein ERP System, kombiniert beides, software- und hardwareorientierte Sicht.

2.2.3 Schichtarchitektur

Ein weiterer Begriff, der in ERP Umgebungen wie in anderen großen EDV Umgebungen Bedeutung hat, ist der Begriff der multitier Architektur, zu deutsch Schichtarchitektur. Hier wird ein Programm so implementiert, dass es in Schichten zur Verfügung steht, wobei eine Schicht auf die jeweils darunterliegende aufbaut. Man kann dies auch als Ebene modellieren.

Die unterste Ebene hätte in diesem Fall die Grundfunktionen, die Ebene darüber würde mit dieser eine Anwendung bauen und die Ebenen darüber können sich z.B. um die Darstellung kümmern. Aber es kann auch mehrere Schichten geben, die nur der Ablaufsteuerung und Datenhaltung dienen. Der große Vorteil solcher Schichtarchitekturen ist, dass bei jeder Schicht klar definierte Zugriffswege und Schnittstellen definiert sind, wodurch diese leichter getauscht bzw. von anderen Programmteilen verwendet werden können. Auch eine Aufteilung der Schichten auf verschiedene Hardwaresysteme wird dadurch vereinfacht.

Im Falle von ERP Systemen (siehe auch folgendes Kapitel 2.2.4) wird die Datenbank meist von einem anderen Hersteller bezogen oder ist gar austauschbar, somit ist die Schicht unabhängig von der verwendeten Sprache und kann unabhängig gewartet werden.

2.2.4 Bestandteile eines ERP Systems

In einem ERP System werden jene Daten, die aus den verschiedenen Prozessen im Unternehmen anfallen, gespeichert und verarbeitet. Zum Beispiel ist das Einbuchen der Werte eines Papierbelegs in ein ERP System ein solcher Prozess.

Zur Datenspeicherung wird eine Datenbank verwendet welche sicherstellt, dass die Daten dauerhaft und unter definierten Integritätsregeln⁸ abgelegt werden. Eine Datenbank hat zusätzlich die Eigenschaft über Optimierungs- und Indizierungsmethoden Massenzugriffe zu beschleunigen und bietet als Zugriffssprache eine Form von SQL (Structured Query Language) Implementierung (oftmals als Dialekt) an.

Die Daten in der Datenbank werden nun von den verschiedenen Benutzern bearbeitet. Hier kann zwischen einem Online- und einem Offline Zugriff unterschieden werden, wobei in dieser Arbeit wegen der komplexen Synchronisationsmechanismen bei Offlinedatenbeständen zunächst die Onlinezugriffe bearbeitet werden sollen.

Beim Zugriff und der Verarbeitung der Daten unterscheiden gute Programme zumindest zwischen Applikations- und Präsentationskomponenten. Die einen dienen dazu, die einzelnen Schritte der Anwendung zu steuern, die Präsentationskomponente zeigen die Daten dann dem Anwender und lassen diesen mit der Applikation interagieren.

Das Hauptmodell von ERP Software ist ein Dreischicht-Modell, welches in Abbildung 2.4 zu sehen ist.

Manche Autoren beschreiben zusätzlich noch eine Schicht Infrastruktur zwischen der Datenbank und den Applikationsprozessen⁹.

Aus Gründen der Skalierbarkeit und Modularisierung des Systems werden die einzelnen

⁸Siehe weiterführend auch [ZW01, S.315ff].

⁹Wie z.B. in [ZW01, S.126].

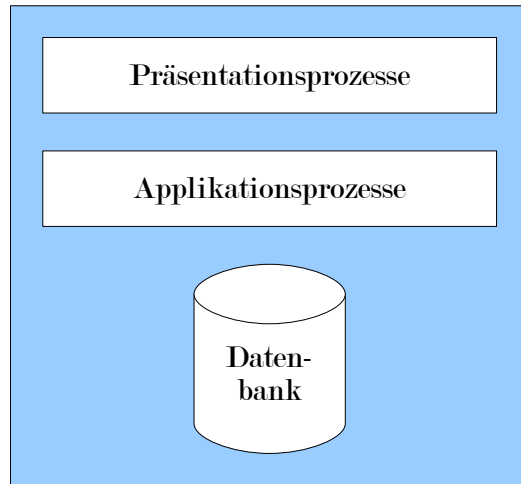


Abbildung 2.4: 3-Schichtmodell

Schichten heute in der Regel als Instanzen ausgeführt. Dies bedeutet, dass man diese Komponenten auftrennen und auf jeweils getrennte Hardware installieren kann. In einer 3-schichtigen, voll skalierten Landschaft hat man somit zumindest eine Hardware für die Datenbank, eine für die Applikationskomponenten und eine für die Präsentationskomponenten¹⁰.

2.2.5 Schnittstellen und Kommunikation nach außen

Ein ERP System besitzt die Rolle eines Informationssilos, wo Daten zu den wesentlichen Prozessen des Unternehmens gespeichert und verarbeitet werden. Diese Daten werden aber nicht alle händisch erfasst, vieles kann aus anderen Systemen gewonnen werden.

Aus diesem Grund ist es von großer Wichtigkeit, dass ein ERP System Schnittstellen nach außen besitzt, welche den gängigen Standards genügen. So gibt es beispielsweise für die verschiedensten Anwendungsfälle standardisierte Datenaustauschformate. Diese zu berücksichtigen, sowie einen Zugriff von den verschiedensten Programmiersprachen und -umgebungen zu ermöglichen, ist ein wichtiges Ziel von ERP Software.

Eine Schnittstelle definiert sich in der Regel durch verschiedene nach außen angebotene Funktionen, die Eingabe- und Ausgabestrukturen besitzen. Diese Strukturen bestehe aus einer Menge von Einzeldatenfeldern. Beim Aufruf einer solchen Schnittstellenfunktion kann nun aus den Eingaben über den Aufruf eine Ausgabe erzeugt werden. Der Code, der auf dem Zielsystem ausgeführt wird, ist gegenüber dem Klienten der Schnittstelle nicht transparent.

¹⁰Siehe weiterführend [Gal05, S.59ff.] für eine genaue Beschreibung.

Abhängig von den verwendeten Protokollen ist für die meisten ERP Systeme eine eigene Zugriffsbibliothek vonnöten, was eine individuelle Programmierung für die externe Ansteuerung bedingt.

2.2.6 Webservices und SOA

Durch die Problematik verschiedenster proprietärer Schnittstellen (z.B. RFC Schnittstelle bei SAP) im Softwareumfeld ist in den letzten Jahren die Komplexität von Unternehmenssoftware stetig gestiegen. Um hier Vereinfachungen zu erreichen, hat sich die IT Industrie auf ein neues Konzept eingeschworen, den Web Services Standard. Aufbauend auf diesem Standard soll eine völlig neue Methodik der Softwareentwicklung eingeführt werden, die serviceorientierte Architektur, kurz SOA.

Bei modernen Programmiersprachen, wie Java, steht die Wiederverwendung des Codes und eine gewisse Objektorientierung im Vordergrund. Diese Paradigmen gab es bisher zwischen Systemen nicht. Wohl gab es die Schnittstellen, um das jeweils andere System anzusteuern. Die Möglichkeit, Code des jeweils anderen Systems wiederzuverwenden scheiterte aber oft am großen Aufwand, die entsprechenden Schnittstellen zur Verfügung zu stellen.

Mit SOA wird versucht, mittels Webservices die Möglichkeit zu schaffen, über das verbreitete Internetprotokoll die Daten in einem plattformunabhängigen Format zu übertragen. Über Repositories werden Services zentral verwaltet und können dann konsumiert werden.

Die Architektur für das Offerieren von Web Services ist heutzutage in alle ERP Systeme fix integriert und kann in der Regel einfach aus bestehenden Schnittstellen nachgeneriert werden¹¹.

2.2.7 Multiplexing

Eine der wichtigsten Aufgabenstellungen an die Architektur von ERP Systemen ist die Skalierbarkeit für den Zugriff von mehreren hundert bis tausenden Benutzern gleichzeitig. Dies ist notwendig, da die zentrale Datenbasis des Unternehmens von fast allen Benutzern, quer über die Abteilungen, benötigt wird.

Wenn nun Benutzer am zentralen ERP System verbunden sind, muss die Software mittels der verschiedenen Client-/Server-Techniken¹² in der Lage sein, die Benutzeranfragen optimal auf die vorhandenen Serverprozesse aufzuteilen. Dies geschieht mittels sogenann-

¹¹Siehe auch weiterführend [Gal05, S.48].

¹²Siehe Abschnitt 2.2.2.

tem Multiplexing.

Das Prinzip dieser Technik beruht darauf, dass bei gleichzeitiger Verwendung eines Systems durch Menschen die meiste Zeit von den Benutzern dafür verwendet wird, die Daten einzugeben, zu betrachten und zu „denken“ (vgl. Englisch think time). Während dieser Zeit ist der Server nicht direkt mit der Anfrage bzw. Transaktion des Klienten beschäftigt. Durch diesen Umstand ist es nicht notwendig, während der Benutzerzeit den Server zu blockieren. Währenddessen ist es durchaus möglich, andere Benutzeranfragen mit den gleichen Ressourcen zu bearbeiten.

Die Konzeption eines multiplexing-Systems erfolgt durch die Definition von Arbeitsprozessen (Workprozessen), welche die Abarbeitung der Anfragen übernehmen. Zusätzlich dazu wird ein Prozess dafür geschaffen, die angemeldeten Klienten zu verwalten und ihnen Arbeitsprozesse zuzuweisen wenn diese eine Anfrage stellen. Dieser Prozess wird dann Dispatcher genannt. Man könnte eine Anfrage an einen ERP Server grob wie folgt skizzieren:

- Die Anfrage vom Klienten an den Server schlägt auf dem Dispatcherprozess auf
- Der Dispatcher weist dem Klienten einen Serverprozess zu, der gerade Anfragen verarbeiten kann (nicht von anderen Benutzern verwendet wird)¹³
- Der Serverprozess bearbeitet die Anfrage und schickt die Antwort an den Klienten
- Der Serverprozess wird beim Dispatcher wieder als bereit gemeldet.

Nach einem solchen Durchlauf ist das Ergebnis der Anfrage beim Benutzer und kann weiter bearbeitet werden bzw. in einer weiteren Anfrage enden.

Im Detail muss man sich hier natürlich auch um den Benutzerkontext kümmern, der die Statusinformationen zum Benutzer kennt und der Werte zwischenspeichern kann, solange ein Vorgang durch eine einzige Anfrage nicht abgeschlossen werden kann (Programme mit mehreren Teilmasken bzw. Arbeitsschritten). Grafisch lässt sich ein Multiplexer wie in Abbildung 2.5 darstellen.

Die Serverprozesse haben hier keine eindeutige Benutzerzuordnung und werden dynamisch vom Dispatcher zugeteilt. In Architekturzeichnungen findet sich in der Regel eine Darstellungsform wie in Abbildung 2.6.

Die einzelnen Serverprozesse können noch einen eigenen Typ besitzen, um Anfragen eines bestimmten Typs einfacher oder besser verarbeiten zu können (z.B. eigene Druck-

¹³Ist gerade kein Prozess verfügbar, so wird die Anfrage in eine Warteschlange gestellt, bis ein Prozess frei geworden ist.

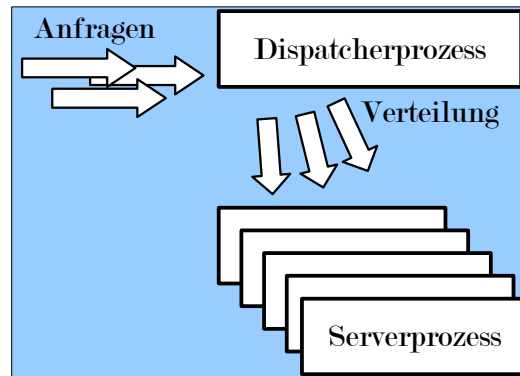


Abbildung 2.5: Multiplexer Funktionsweise

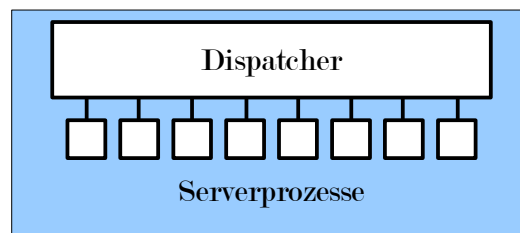


Abbildung 2.6: Dispatcher Architektur

prozesse, Hintergrundprozesse, etc). Das hat dann ebenfalls Auswirkungen auf die Skalierbarkeit des Systems.

2.2.8 Abbildungen

Um ERP Systeme abzubilden, werden die einzelnen physischen Server meist in einem Diagramm gezeichnet, wobei die klassische Benutzeranfrage ein gutes Vorbild dafür ist. Selten wird auf Ebene der Serverprozesse angezeigt, wie der innere Aufbau des entsprechenden Systems ist. Eine beispielhafte Abbildung eines ERP Systems wäre Abbildung 2.4 aus Abschnitt 2.2.4, erweitert um die Serveradressen, auf denen die drei Komponenten laufen.

Aus diesen Abbildungen muss ein IT Mitarbeiter herauslesen können, welcher Server welche Aufgabe(n) besitzt, ob es sich um ein ausfallsicheres System handelt und idealerweise auch, welche Netzwerkinterfaces vorhanden sind. Die Abbildung kann herangezogen werden bei der Systemplanung, bei Ausfällen oder auch bei dringenden Wartungsarbeiten (wenn es um die Frage geht, welche Auswirkungen das Herunterfahren eines Systembestandteiles hat).

2.3 Systemlandschaften in Großunternehmen

2.3.1 Begriff der Systemlandschaft

Eine Systemlandschaft lässt sich am besten beschreiben als eine logische Anordnung von EDV Systemen. Die Gruppierung selbst kann auf verschiedene Arten gewählt werden. Es kann z.B. eine Landschaft der SAP Systeme gewählt werden oder eine Landschaft der Entwicklungsserver aber auch die produktive Landschaft der Server für ein bestimmtes Land, wenn die komplette IT Landschaft noch größer wird. In der Regel ist ein System Mitglied von mehreren Landschaften im Unternehmen. Die SAP bietet etwa mit dem System Landscape Directory (SLD) die Möglichkeit, Systeme zu Landschaften zusammenzufassen und diese dann über das Web-Tool Netweaver Administrator gemeinsam zu verwalten.

Zusammenfassend:

- Eine Systemlandschaft kann mehrere Systeme enthalten
- Eine System bezeichnet ein ERP- oder Fremdsystem, das sich über mehrere Server erstrecken kann
- Ein System kann in mehreren Landschaften vorkommen
- Die Systemlandschaft ist z.B. nach Anwendungen oder nach Aufgaben der Systeme gegliedert

Dieser Zusammenhang ist in Abbildung 2.7 zu sehen.

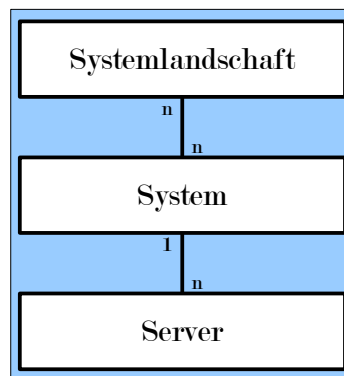


Abbildung 2.7: Zusammenhang zwischen Landschaft, System und Server

Dem Begriff der Systemlandschaft kommt vor allem bei der ganzheitlichen Betrachtung der IT Infrastruktur eine Bedeutung zu: Welches System ist mit welchem gleichzusetzen?

Welche Systeme gehören zu den produktiven Systemen (und benötigen daher besonderen Schutz)?

2.3.2 Evolution der IT Infrastruktur

Historisch gesehen entstanden ab der Mitte des letzten Jahrhunderts in vielen Firmen erste EDV Systeme, die für einzelne Prozesse, wie Produktion, Verkauf oder Lager- und Finanzverwaltung genutzt werden. Diese Systeme waren vermutlich in mehreren Ausprägungen vorhanden. In der Regel existierten diese als ein System zum Entwickeln und einem System, wo die entwickelte Software dann eingespielt und produktiv genutzt wurde (vgl. ein einfaches Beispiel in Abbildung 2.8).

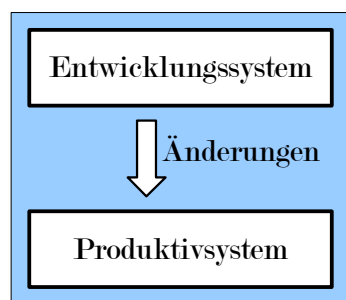


Abbildung 2.8: Zwei Systeme, eine erste „Systemlandschaft“

Bei diesen Altsystemen endeten die Prozesse der EDV aber meist an der Systemgrenze, das Lagerverwaltungssystem kommunizierte in der Regeln nicht dauerhaft oder automatisiert mit dem Verkaufssystem.

Mit den integrierten ERP Lösungen (siehe auch Kapitel 2.1.2) wurden die Systeme erstmals auf gemeinsame Systeme integriert. Dies führte in der Umstellungsphase zunächst dazu, dass viele Altsysteme mit ihren Daten nun in ein neues System überführt werden mussten. Vielfach entschied man sich aber auch, einzelne Funktionen in den alten Systemen zu belassen und die Altsysteme dann für automatisierte Anfragen zu nutzen¹⁴. Die ersten Systemabhängigkeiten waren damit gegeben. Es war nun nicht mehr einfach möglich, eines der beiden Systeme ohne Rücksicht auf das jeweils andere abzuschalten, zu warten oder in einen anderen Zustand zu versetzen.

In den modernen Landschaften von ERP² und den aufkeimenden SOA Lösungen ist es

¹⁴Tatsächlich ist es heute noch in vielen Kreditinstituten so, dass die aus den 1970er Jahren stammenden, Host-basierten, sogenannten „Kernbankensysteme“ immer noch für die Kontenverwaltung genutzt werden.

von entscheidender Wichtigkeit, dass Systeme entsprechend ihrer Abhängigkeiten behandelt werden. Hier ist der entsprechende Landschaftsgedanke ganz besonders wichtig. Änderungen am Gesamtsystem werden dabei nur mehr an gesamten Systemlandschaften vorgenommen. Wenn in einer SOA Architektur Services von mehreren produktiven Systemen genutzt werden, so kann die übergreifende Anwendung, welche diese Services konsumiert, nur dann korrekt funktionieren, wenn die einzelnen Services miteinander getestet und im gleichen Releasestand sind. Änderungen am Gesamt-SOA-System werden dann auf Systemen in einer Entwicklungslandschaft vorgenommen und dann zusammen möglichst gleichzeitig auf der Testumgebung eingespielt. Sind die Tests dort abgeschlossen, dann wird auf der produktiven Landschaft gleichartig vorgegangen. Ohne Testumgebung könnte das reibungslose Zusammenspiel der einzelnen Komponenten nicht gewährleistet werden. Diese Aktualisierungen kann man als „Einspielen eines Transports“¹⁵ oder aber auch als „Aktivieren eines Changes“¹⁶ bezeichnen. Eine Skizze für zwei miteinander kommunizierende, z.B. SOA basierte Systeme und die entsprechenden Landschaften ist in Skizze 2.9 umrissen.

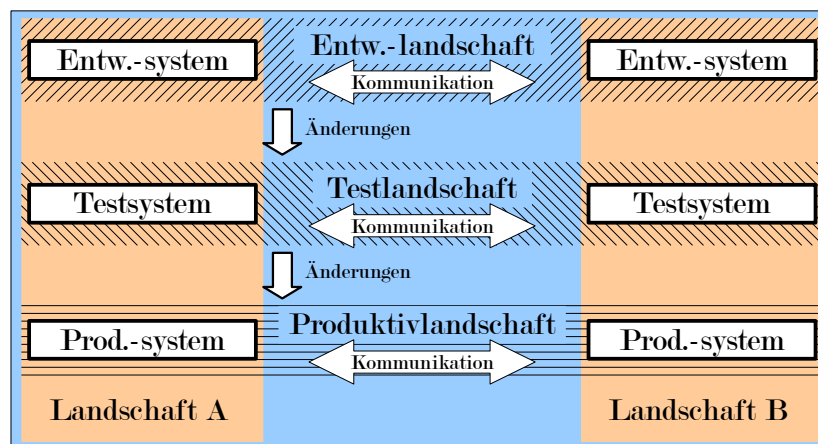


Abbildung 2.9: Erweiterte Systemlandschaften mit Kommunikations- und Änderungspfaden

Betrachtet man die Skizze genauer, so erkennt man zwei Landschaften A und B. Diese seien ein ERP und ein ERP2 System, die im täglichen Betrieb miteinander kommunizieren. Für jedes System gibt es ein Entwicklungs-, ein Test- und ein Produktionssystem. Dies wird auch als 3-Systemlandschaft bezeichnet¹⁷.

¹⁵Die entspricht eher dem SAP Vokabular.

¹⁶Ein Change ist eher ITIL Vokabular.

¹⁷Dies ist von der SAP für ERP Systeme et al. empfohlen und kann bei [MFMS00, S.74ff.] weiterführend nachgelesen werden.

Die Entwicklungssysteme unter sich sind wiederum zu einer weiteren Landschaft, der Entwicklungslandschaft zusammengefasst, ebenso die Test- und Produktionssysteme. Für die Aufrechterhaltung des normalen Betriebs ist es nun notwendig, dass die Schnittstellen und Programme, über die das ERP und das ERP2 System miteinander kommunizieren, gleichartig beschaffen sind und genau so funktionieren, wie das eine dies vom jeweils anderen System erwartet. Dies kann nur dann gewährleistet werden, wenn Änderungen an einem System nicht unabhängig vom jeweils anderen System durchgeführt werden. Man aktiviert die Changes also immer auf einer gesamten Landschaft. Zuerst geschieht dies auf der Testlandschaft, nach erfolgreichem Test werden die Änderungen dann auf der Produktivlandschaft aktiviert.

Kombiniert mit den Erläuterungen in Abschnitt 2.3.1 bedeutet dies, dass bei einem Transport von Änderungen von der Entwicklungslandschaft auf die Testlandschaft alle Server von der Aktualisierung betroffen sind, die zu den jeweiligen Testsystemen der Landschaft A und B gehören (die Server sind in der Abbildung nicht ersichtlich).

2.3.3 Typische Systemlandschaften von Referenzunternehmen

Die Praxis der existierenden Systemlandschaften in Unternehmen ist nun weit komplexer als das zuvor genannte Beispiel der „erweiterten Systemlandschaft“ (Abbildung 2.9). Wird Standardsoftware eingesetzt, so erwirbt man zusätzlich zu den entsprechenden Programmen auch einen Wartungsvertrag, der sicherstellt, dass die Software den aktuellen gesetzlichen und sicherheitstechnischen Anforderungen entspricht. Weiters ist in dieser Wartung ein Kundensupport oder eine Kundenhotline des Herstellers inkludiert, die auch aktuelle Probleme in den einzelnen Releases der Standardsoftware beheben kann. Zu festgelegten Zeiten liefert der Hersteller neue Releases, Hilfspakete oder dringende Korrekturen aus, die dann auf den Entwicklungssystemen des Kunden eingespielt werden. Sollte der Kunde Änderungen an den Standardsoftwarekomponenten vorgenommen haben, muss nun bei jeder Aktualisierung auf diese Änderungen Rücksicht genommen werden. Ein sogenannter Abgleich findet statt¹⁸. Dies hat zur Folge, dass zeitweilig das Entwicklungs- bzw. auch das Testsystem nicht den gleichen Versionsstand wie das Produktionssystem aufweisen. Während dieser Abgleichszeit sollen bei einer lediglich aus drei Systemen bestehenden Landschaft nun keine dringenden Änderungen am laufenden Produktionssystem vorgenommen werden, da die verschiedenen Versionsstände Un-

¹⁸Abhängig von der Kompatibilität der jeweiligen Anpassung spricht man im SAP Umfeld von Erweiterungen, Kundenentwicklungen und Modifikationen. Die genaue Definition an dieser Stelle würde den Rahmen der Arbeit sprengen. Der interessierte Leser sei z.B. auf [MFMS00, S.48ff sowie S.42ff] und auf die Ansätze in Kapitel 3 verwiesen.

verträglichkeiten aufweisen könnten. Es kann also sein, dass ein Unternehmen mehrere Entwicklungs- und Testsysteme besitzt, um neue Versionen in einem sicheren Umfeld zu adaptieren, während auf den anderen Systemen die aktuelle Version im Produktionssystem gewartet werden kann. Es kann hier auch von Release- und Maintenance-“Tracks“ gesprochen werden. Ein Beispiel für solche Tracks wird in Abbildung 2.10 gezeigt.

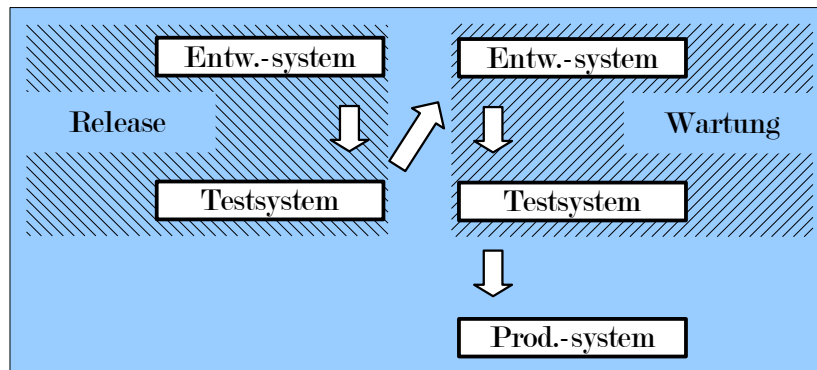


Abbildung 2.10: Wartungs- und Releasetracks in einer Systemlandschaft

Im rechten schraffierten Bereich kann so lange die aktuell auf dem Produktionssystem vorhandene Version gehalten werden wie die Entwickler auf den Systemen im linken Bereich brauchen, um die neue Release der Software anzupassen und durch Tests verifizieren zu lassen. Ist die Release fertig gebaut, wird es in die Wartung (rechts) übernommen und produktiv gesetzt. Der einzige Nachteil einer solchen Konstruktion ist, dass die in der Zwischenzeit im Maintenance Track vorgenommenen Änderungen bei Übernahme in die Wartung eventuell nochmals angeglichen werden müssen.

Ein anderer Aspekt, der die Komplexität von Systemlandschaften erhöht, ist meist jener der Internationalisierung. Wenn Software in verschiedenen Ländern von verschiedenen Töchtern des Unternehmens eingesetzt wird, so wird in Konzernen gerne eine globale Systemlandschaft eingerichtet¹⁹. Hier errichtet die Konzernzentrale oder auch ein eigenes zentrales IT Service einen Prototypen, oder auch „installation master“ genannt, wo Programme und Einstellungen an die entsprechenden Tochtergesellschaften ausgeliefert werden, die im ganzen Unternehmen gebraucht werden können. Die Auslieferung ist aber noch keineswegs ein fertiges System und muss von den jeweiligen Konzerntöchtern noch fertig adaptiert werden. Es ist auch möglich, dass die einzelnen verbundenen Unternehmen eigene Zusatzentwicklungen vornehmen.

Bei sehr großen Konzernen, mit besonders vielen Mitarbeitern kann es nun auch noch der Fall sein, dass solche „installation master“ für einzelne Teilbereiche, wie z.B. das

¹⁹Weiterführend beschrieben in [MFMS00, S.90ff.].

Personalwesen existieren. Diese sehr seltenen Fälle werden hier außer Acht gelassen. Ein Beispiel einer globalen Systemlandschaft befindet sich in Abbildung 2.11.

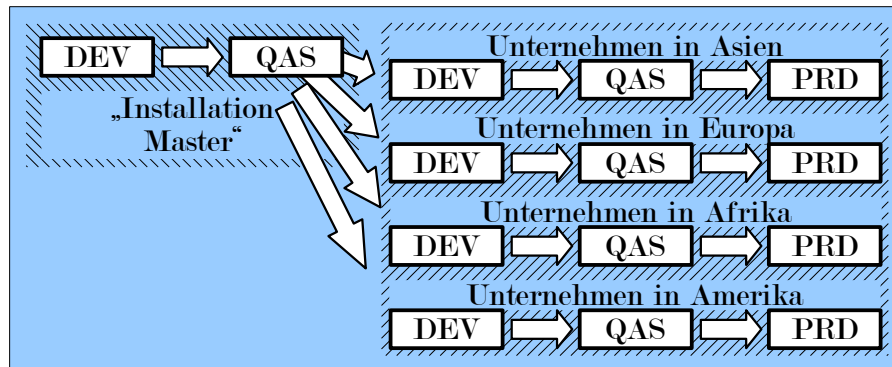


Abbildung 2.11: Globale Systemlandschaft

In der Abbildung steht DEV für Entwicklungssysteme, QAS für die Qualitätssicherung und PRD für die Produktivsysteme. Man kann erkennen, dass hier bereits $2 + (3 \cdot \text{Anzahl der Unternehmen})$ Systeme benötigt werden. Aus der Erkenntnis, dass ein System sich in der Regel aus mehreren Servern zusammensetzt, kann man bereits die hohen Anforderungen an die Hardware ableiten.

2.3.4 Planung von hoch integrierten Systemlandschaften

Bei der Planung von Systemlandschaften wird so vorgegangen, dass die jeweiligen Kunden in der Regel die IT Abteilung oder den Outsourcing Partner (im Folgenden werden beide als Hosting Provider bezeichnet) damit beauftragen, die Konzeption für eine Systemlandschaft vorzunehmen. Kunde kann in diesem Fall das gesamte Unternehmen sein (z.B. unternehmensweite ERP Lösungen) oder aber auch nur ein Fachbereich (z.B. Personalwirtschaftssysteme²⁰ oder Vertriebsunterstützung). Der Kunde hat in der Regel genaue Vorstellungen davon, welche Systeme er einführen möchte (z.B. ein ERP und ein ERP2 System oder die Erweiterung eines ERP Systems und zusätzlich ein HR System), aber es ist nicht unbedingt seine Aufgabe, die Abhängigkeiten innerhalb der integrierten Gesamtlandschaft zu kennen. Kunde und Hosting Provider verhandeln diese gemeinsam. Dabei verfolgen der Kunde und der Hosting Provider verschiedenartige Ziele.

Der Kunde

- möchte, dass seine Systeme möglichst immer verfügbar und mehrfach abgesichert sind.

²⁰Diese werden auch HR Systeme genannt.

2 Grundlagen von Enterprise Resource Planning

- ist daran interessiert, bei Ausfällen schnellstmöglich eine Wiederherstellung des betroffenen Systems zu erhalten.
- will alle seine Probleme auf den Hosting Provider übergehen lassen und jegliche Symptomatik von diesem abgeklärt wissen.
- benötigt besonders schnelle Systeme, um effizient arbeiten zu können.

Im Gegensatz dazu will der Hosting Provider

- viele Wartungsfenster zur Verfügung haben, um Tests und Änderungen an der Konfiguration möglichst oft vornehmen zu können.
- mehrere Stunden Zeit haben, bis ein gestörtes System wieder korrekt laufen muss.
- nur die Probleme lösen, für die er auch wirklich zuständig ist, und das mit möglichst wenig Ressourcen.
- möglichst wenige Abhängigkeiten zu anderen Systemen, um die Landschaft „wartungsfreundlich“ zu halten.

Im den folgenden Abschnitten wird auf die vier wichtigsten Faktoren²¹ bei der Systemlandschaftsplanung eingegangen. In Abwägung der vier Aspekte: Verfügbarkeit, maximale Zeit für Fehlerbehebung, maximale Antwortzeit und Supportleistungen erzielen die beiden Partner in der Regel einen Kompromiss, der in einem Service Level Agreement (SLA) niedergeschrieben wird²².

Aus diesem Service Level Agreement leitet nun der Hosting Provider ab, welche Infrastruktur benötigt wird. Daraus ergibt sich das Design der Systeme (Anzahl, die Anzahl der Server, die dafür benötigte Hardware und letztlich auch das Aussehen der Systemlandschaft). Wichtige Inputs geben dem Hosting Provider dabei die Hardwarehersteller und auch die Erfahrung mit entsprechenden Vergleichssystemen oder -kunden. Findet die Systemlandschaft und das SLA Zustimmung, so wird eine entsprechende Bestellung seitens des Kunden getätigt.

Verfügbarkeit

Die gewünschte Verfügbarkeit eines Systems ist wichtig für das Ausmaß der Redundanz, welches bei der Hardwareplanung definiert wird. Wenn ein System nur von einem Land

²¹Diese Darstellung entspricht der Sicht des Autors, eine weitergehende Auflistung möglicher Anforderungen findet man z.B. bei [KB04, S.117 ff.].

²²Für weitere Ausführungen und Erfolgsfaktoren von Service Level Agreements siehe [Lie06, S.5 ff.].

genutzt wird, so kann man meist davon ausgehen, dass z.B. an Feiertagen des jeweiligen Landes oder auch in den Nachtstunden keine bis wenige Personen am System arbeiten müssen. Hier wird man sich entscheiden, eine Verfügbarkeit von 8 bis 12 Stunden am Tag, an 5 Tagen die Woche zu gewährleisten. In den restlichen Zeiten kann der Hosting Provider am System Wartungen durchführen. Diese Variante ist für den Kunden deutlich kostengünstiger, als wenn eine durchgehend hohe Verfügbarkeit gewünscht wird.

Ein höherwertiges, sogenanntes HA²³-Szenario ist jedoch bei Schichtbetrieb oder auch bei internationalem Einsatz des Systems unumgänglich. In diesen Fällen muss das System wegen verschiedener Zeitzonen, Wochenenden, Feiertagen oder wegen nächtlichen Kundenzugriffen an 7 Tagen 24 Stunden lang laufen. Für die Wartung müssen Backupsysteme errichtet werden, die während der Wartungsfenster genutzt werden können.

Bei beiden Szenarien, mit und ohne HA, hängen die tatsächlichen Kosten maßgeblich von der maximalen Zeit für die Fehlerbehebung ab (siehe nächster Abschnitt). Ist die Zeit sehr kurz bemessen, muss das System mit einem höheren Grad von Redundanz ausgelegt werden, sodass Teile der Hardware jederzeit ausfallen können und auch Netzverbindungen mehrfach abgesichert sind.

Fehlerbehebung

Ein weiterer Faktor, welcher sich auf das Design der Systemlandschaft auswirkt, ist die zwischen Kunde und Hosting Provider vereinbarte maximale Zeit der Betriebsunterbrechung im Fehlerfall. Abhängig von Systemart (DEV, QAS, PRD) und Einsatz im Projekt können hier unterschiedliche Zeiten vereinbart sein. Ist das Projekt ein Einführungsprojekt, wo sehr viel Entwicklung auf dem DEV System vorgenommen wird, dies auch noch unter Einbeziehung vieler, teurer, externer Ressourcen, so ist das Entwicklungssystem innerhalb kürzester Zeit wieder in seinen Normalzustand zu versetzen. Wenn jedoch wenig entwickelt wird, so kann es sein, dass eine Fehlerbehebungszeit von ein paar Tagen zweckmäßig erscheint. Ähnliches gilt für Produktionssysteme. Hier stellen sich Kunde und Hosting Provider meist die Frage, wie geschäftskritisch ein Ausfall des Systems ist. Wenn die komplette Produktion des Unternehmens bei Ausfall des Systems zum Erliegen kommt, ist natürlich viel besser abzusichern, als wenn lediglich Teile des Unternehmens betroffen wären.

Aus Kostengründen wird hier meistens der Sparstift angesetzt. Der Autor konnte in seiner Berufspraxis allerdings bemerken, dass die Service Level Agreements in verschiedenen Unternehmen nach dem Auftreten eines Zusammenbruchs sehr oft in diesem Punkt neu

²³Englisch: high availability.

verhandelt wurden, weil die ursprünglich vereinbarte Zeit als zu lange angesehen wurde.

Antwortzeit

Die Antwortzeit eines Systems von einer Anfrage bis zu einer beim Benutzer aufscheinenden Antwort hängt von einigen Faktoren ab, u.a. von

- der Geschwindigkeit des Client PCs (Aufbereitung der Antwort des Systems).
- der Netzverbindung zwischen dem Client und dem System.
- der Geschwindigkeit des Systems.
- der Art des Systems (Reporting, Onlinetransaktionen, Offline...)
- der Komplexität der gestellten Anfrage.

Aus diesen Gründen kann keine exakte Garantie für eine tatsächliche einzelne Anfrage gegeben werden. Es existieren für die verschiedenen ERP Produkte jedoch Benchmark Systeme²⁴, die einen Vergleich zulassen. Die durchschnittliche Antwortzeit für ein System wird im SLA vereinbart und gleichzeitig werden Messmethoden und -intervalle definiert, die idealerweise auch auf die regionale Verteilung der Kunden abgestimmt sind, sodass die Zeiten auch von entfernten Orten aus eingehalten werden können.

Bei Systemen mit vielen Benutzern wird in der Vereinbarung zwischen Kunden und Hosting Provider in der Regel viel mehr Augenmerk auf die Antwortzeit gelegt als bei Systemen mit kleinem Benutzerkreis. Dies hat hauptsächlich eine Begründung darin, dass die Produktivität eines Endbenutzers maßgeblich gesteigert werden kann, wenn das System schneller arbeitet.

Supportleistungen

Die Qualität von Supportleistungen ist höher, wenn Personen den Support übernehmen, die eine gute Kenntnis des Unternehmens haben. Für den Endbenutzer ist es besonders praktisch, wenn der Support für alle ihn betreffenden Angelegenheiten über eine zentrale Stelle erreichbar ist. Der Support wird meist in verschiedene Support Level eingeteilt und je nach Komplexität der Anfrage wird das Anliegen dann so lange die einzelnen Support Level hinauf gehoben, bis das Anliegen gelöst werden konnte.

Beispielhafte Aufgaben für einzelne Support Level:

²⁴Im Falle von SAP gibt es den SD Benchmark, dieser entspricht der benötigten Hardware für einen durchschnittlichen Vertriebsbenutzer auf einem System.

2 Grundlagen von Enterprise Resource Planning

- First Level Support: Gängige, bekannte Probleme beim Benutzen von Software oder beim Auffinden von Informationen.
- Second Level Support: Problemstellungen, die genaueres Wissen voraussetzen.
- Third Level Support: Probleme, die nur mehr vom Entwickler gelöst werden können, oder die Spezialwissen erfordern.

Es ist für den Kunden in der Regel günstiger, wenn First und Second Level Support über ein Kompetenzzentrum selbst erledigt werden und nur der Third Level Support vom Hosting Provider vorgenommen werden muss.

Meistens setzt man aber in Unternehmen bereits auf eine vorhandene Support Infrastruktur auf und muss über das neue System lediglich den Third Level Support sowie entsprechende Schulungsmaßnahmen definieren. Dies kann die Kosten weiter senken.

Anmerkung: Die Aufgaben und die Prozesse im Service Desk nach ITIL werden in Kapitel 5 noch eingehender behandelt.

Im Rahmen der drei Abschnitte des gegenständlichen Kapitels wurden die Funktionsweise und der Einsatz von ERP Systemen besprochen. Die Historie von ERP Systemen wurde dargelegt, mit einem klaren Ausblick auf zukünftige Entwicklungen. Eine architekturelle Einführung brachte dem Leser das Grundverständnis von gängigen Systemkonzepten näher. Mit der Definition des Begriffs der Systemlandschaft, umfasst von typischen Beispielen, schließt dieses Kapitel. Später, im Bereich der SAP Change Prozesse (Kapitel 6) wird dieses Wissen dem Leser als Grundlage dienen.

Im nun folgenden Kapitel wird eine Übersicht über die Besonderheiten der Softwareentwicklung im Unternehmensbereich gegeben, mit den verschiedenen Anpassungsmodellen sowie einem Ausblick auf das spezifische Projektmanagement in ERP Softwareprojekten.

3 ERP Software Lifecycle Management

3.1 Begriff und Definition

Der Kreislauf des Software Lifecycle in einem Unternehmen wird in Abbildung 3.1 illustriert.

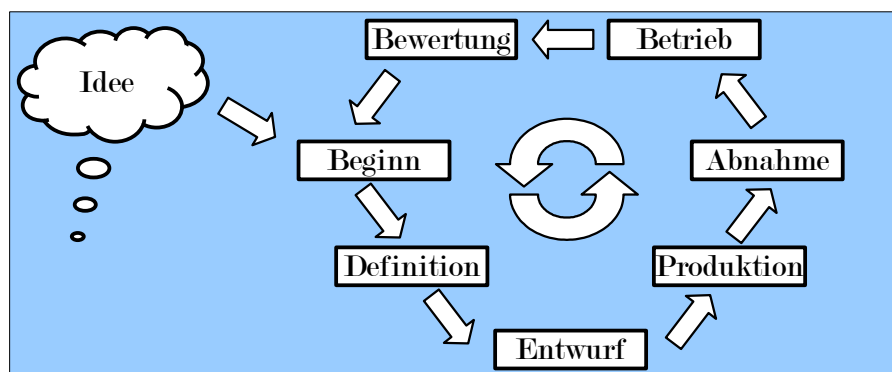


Abbildung 3.1: Software Lifecycle nach [ZW01, S.46].

Bei [ZW01, S.45] wird Software Lifecycle wie folgt definiert:

Während der Lebensdauer eines Systems ist eine zumindest einmalige sequenzielle Durchführung aller notwendigen Arbeitsschritte (Anforderungen, Analyse, Entwurf, Implementierung, Test und Inbetriebnahme/Wartung) vorgesehen. Änderungen der Anforderungen werden erst im Zuge eines neuen Projekts durchgeführt, welches wiederum alle Arbeitsschritte in der vorgegebenen Reihenfolge durchläuft. Das größte Problem bei einem solchen starren Vorgehen stellt die fehlende Möglichkeit eines oder mehrerer Schritte zurück dar, wenn bereits während der Durchführung eines Projekts (auch im Zuge der Wartung) Änderungen der Anforderungen notwendig werden sollten.

Auf dieser Idee basierend gibt es verschiedene Vorgehensmodelle, die sich vom intuitiven Build-And-Fix-Cycle (vor sich hin programmieren während man gedanklich spezi-

fiziert²⁵) durch die Strukturiertheit des Vorgehens unterscheiden. Einige Modelle seien hier erwähnt, eine Behandlung würde den Rahmen sprengen, somit sei der interessierte Leser auf [ZW01, S.45 ff.] verwiesen: Wasserfallmodell, Spiralmodell, V-Modell. Für eine ausführliche Erläuterung inklusive der entsprechenden Prozesse sei noch die Abhandlung in [Spi07, S.26 ff.] referenziert.

Am Beginn des Software Lifecycle steht immer die Idee, bei ERP Systemen wohl die Absicht oder die Entscheidung der IT Leitung oder der Geschäftsführung, eine ERP Software einzuführen. Ausgehend von dieser Idee wird dann der Prozess des Software Lifecycle durchlaufen.

Jedes am Software Lifecycle orientierte Vorgehensmodell hat gemeinsam, dass immer bei Erreichung des Status **Betrieb** eine sogenannte Release vorliegt²⁶, welches in weiteren Durchläufen verbessert werden kann.

Zu den einzelnen Schritten aus Abbildung 3.1:

Beginn: Am Beginn des Software Lifecycle steht die Entscheidung, ein neues Softwareprodukt einzuführen oder der Beschluss, ein bestehendes Softwareprodukt an die Anforderungen des Unternehmens anzupassen. Zu diesem Schritt wird auch das Projekt mit den verschiedenen Stakeholdern aufgesetzt. Dazu gehören der Projektsponsor (z.B. der Geschäftsführer) sowie das Projektteam. Dieses besteht wiederum aus der IT Abteilung (CIO²⁷), den Fachabteilungen, einem allfälligen Outsourcing Provider und eventuell Vertretern des Herstellers. Das Projekt wird mit dem KickOff Meeting gestartet und kann dann in die nächste Phase übergehen.

Definition: Bei der Definition des Umfangs ist in hohem Maße der Kunde gefragt. Es gilt seine Anforderungen verständlich in einer Art niederzuschreiben, dass die Entwickler dies in ihre Konzeption einbauen können. Aus den Definitionen des Kunden ergibt sich das Lastenheft, welches in [KB04, S.124 ff.] derart angeregt wird, dass folgende Punkte darin unbedingt vorkommen sollen²⁸:

1. Zielbestimmung: Hier wird beschrieben, welche Ziele durch den Einsatz des Produktes erreicht werden sollen.
2. Produkteinsatz: Es wird festgelegt, für welche Anwendungsbereiche und für welche Zielgruppen das Produkt vorgesehen ist.
3. Produktübersicht: Gibt einen meist grafischen Überblick über die Produktumge-

²⁵Eine schöne Erläuterung des Begriffes Build-And-Fix-Cycle kann bei [ZW01, S.45] nachgelesen werden.

²⁶Siehe weiterführend [Kru00, S.7].

²⁷Der Term Chief Information Officer bezeichnet in der Regel den Leiter der IT Abteilung eines Unternehmens.

²⁸Konträr zu den Pflichtenheften im nächsten Abschnitt existiert hier keine internationale Normierung.

bungen.

4. Produktfunktionen: Die Hauptfunktionen des Produktes aus Auftraggebersicht sind auf oberster Abstraktionsebene zu beschreiben. Das bedeutet, dass die typischen Arbeitsabläufe, die mit dem zu erstellenden Produkt durchgeführt werden sollen, zu nennen sind. Die Funktionalität kann mit Hilfe von Akteuren und Geschäftsprozessen oder mit Hilfe von Schnittstellen und Datenflüssen systematisch dargestellt werden.
5. Produktdaten: Die langfristig zu speichernden Hauptdaten und deren voraussichtlicher Umfang (Mengengerüst) sind aus Benutzersicht aufzuführen.
6. Produktleistungen: Werden an Hauptfunktionen und Hauptdaten Leistungsanforderungen bzgl. Zeit und Genauigkeit gestellt, dann werden sie hier aufgeführt. Zu prüfen ist, ob die gewünschten Leistungen mit den unter 5. genannten Datenmengen erreicht werden können.
7. Qualitätsanforderungen: Die wichtigsten Qualitätsanforderungen und die jeweils geforderte Qualitätsstufe sind hier aufzuführen, wie Zuverlässigkeit, Benutzerfreundlichkeit, Effizienz, etc..
8. Ergänzungen: Hier werden Ergänzungen beschrieben oder besondere Anforderungen, z.B. außergewöhnliche Anforderungen an die Benutzerschnittstelle oder Ähnliches.

Es empfiehlt sich, Dinge wie die Benutzerschnittstelle oder komplexe Abläufe in kleineren Prototypen zu erbringen, um sicherzustellen, dass die Anforderungen korrekt umsetzbar sind²⁹. **Entwurf:** Die Phase des Entwurfs beinhaltet eine Ausarbeitung der zuvor gesammelten Anforderungen durch die Entwickler. Das aus der Definition resultierende Pflichtenheft sollte enthalten³⁰:

1. Einleitung: Gibt einen Überblick über die Anforderungsdefinition.
 - a) Zielsetzung
 - b) Produktumfang (Scope)
 - c) Definitionen, Akronyme und Abkürzungen
 - d) Referenzen

²⁹Vgl. die Anleitungen in [ZW01, S.90 ff.].

³⁰Nach dem ANSI/IEEE Standard 830, [KB04, S.125 ff.].

- e) Überblick
2. Übersichtsbeschreibung: Gibt einen Überblick über das Produkt und die allgemeinen Faktoren, die seine Konzeption beeinflussen.
 - a) Produkt-Umgebung
 - b) Produkt-Funktionen
 - c) Benutzer-Eigenschaften
 - d) Restriktionen
 - e) Annahmen und Abhängigkeiten
 3. Spezifische Anforderungen³¹: Beschreibung aller Details, die für den Systementwurf benötigt werden.
 - a) Externe Schnittstellenanforderungen
 - b) Funktionale Anforderungen
 - c) Leistungsanforderungen
 - d) Entwurfsrestriktionen
 - e) Qualitätsmerkmale
 - f) Andere Anforderungen

Während der Erstellung des Pflichtenheftes kann es zu Nachfragen beim Kunden kommen, zumal nicht immer alle Punkte des Lastenhefts mit entsprechender Kenntnis der Umsetzung definiert werden können. Zusätzlich zum Pflichtenheft wird auch eine erste Kostenschätzung abgegeben, die vom Projektsponsor geprüft und freigegeben wird.

Es ist auch sinnvoll, in diesem Schritt die Datenmodellierung vorzunehmen, da auf diese Weise nochmals die Umsetzbarkeit der Anforderungen verifiziert werden kann. **Produktion:** Der Entwicklungsverantwortliche erhält vom Projektleiter den Auftrag, die Anforderungen aus dem Pflichtenheft umzusetzen. Die Entwickler beginnen ihre Arbeit und übergeben diese nachher an die Tester, welche Testszenarien erstellen und damit die die Funktionsfähigkeit der Software testen und protokollieren.

Abnahme: Bei der Abnahme kontrolliert der Kunde, ob die in Lasten- und Pflichtenheft spezifizierten Anforderungen korrekt umgesetzt wurden. In der Abnahme liegt die spezielle Verantwortung, die Geschäftslogik nochmals zu überprüfen. Am Ende der Abnahme gibt der Projektsponsor das Produkt frei und es geht in Produktion (für einen

³¹Anmerkung: Hier werden als Unterpunkte in der Quelle die wichtigsten Punkte genannt. Die original IEEE Spezifikation kennt hier 8 Vorschläge.

technischen Vergleich siehe Kapitel 2.2.4). Die Verantwortung geht von der Entwicklung zur IT Organisation (oder zum Outsourcer) über und das Projektteam zur Einführung kann aufgelöst werden.

Betrieb: Während des Betriebes des Systems werden keine Änderungen an der Logik vorgenommen. Dringende Änderungen, die im laufenden Betrieb entdeckt werden sowie Wartungen an Systemkomponenten sind allerdings zulässig, sofern diese den Funktionsumfang nicht erweitern.

Bewertung: Wenn ein System längere Zeit genutzt wurde, wird in der Phase der Bewertung darüber resümiert, wie erfolgreich der Einsatz des Produktes ist. Zum Zeitpunkt einer Bewertung ist es möglich, wieder neue Ideen zu generieren und ein Nachfolgeprojekt zu lancieren. Daraus entsteht dann eine Spirale, in welcher das Softwareprodukt immer umfangreicher wird. Bei ERP Standardsoftware kann so eine Spirale auch dadurch geschlossen bzw. der Zustand des Beginns so angesteuert werden, dass der ERP Hersteller neue Funktionalitäten in den Standardumfang verpackt und das Unternehmen beschließt, diese Programmteile nutzbar zu machen. Der Input kommt hier aber wohl von außen, durch Informationen seitens des Herstellers oder durch Forschungsarbeit des Kunden.

3.2 Besonderheiten im Lifecycle von ERP Standardsoftware

Aus [Mil89, S.45]:

Ziel des Software - Entwicklungsprozesses ist die Erstellung eines Programms oder Programmsystems und dessen Dokumentation.

Diese Definition aus dem Jahre 1989 ist sicherlich immer noch gültig. Der Software Lifecycle von ERP Standardsoftware verhält sich aber in einigen Punkten anders als der klassische Software Lifecycle. Dies ist zunächst dadurch bedingt, dass das System an sich zum größten Teil außerhalb, nämlich beim Hersteller der ERP Lösung, entwickelt wird. Es stellt sich also vielmehr die Frage der Anpassung, denn die Frage der Entwicklung. Damit ist es mit einem wesentlich geringeren Aufwand möglich, ERP Systeme für das eigene Unternehmen zu betreiben, als wenn man diese selbst programmieren müsste³². Bei der Anpassung von Standardsoftware kombinieren sich die Elemente des Software Lifecycle mit den Vorgehensweisen und Zielen des Projektmanagements (siehe Abschnitt

³²Dies ist übrigens einer der Hauptgründe für die oft kritisierten, mitunter hohen Preise von ERP Systemen: Individualsoftware wäre noch teurer. An dieser Stelle sei dem Leser die teils ironische, aber im Kern sehr ernste Lektüre von [DeM97, S.5 ff.] empfohlen.

3.4). Alle im ERP Bereich vorgeschlagenen Projekte sollten sich an der Strategie des gesamten Unternehmens ausrichten³³.

Es ist anzufügen, dass die meisten ERP Hersteller den vollständigen, betriebswirtschaftlich relevanten Code der jeweiligen Anwendungen zur Verfügung stellen und auf die eine oder andere Weise Änderungen erlauben. Dies ermöglicht einen genauen Einblick in die Funktionsweise der Programme und kann über Debugging vom Kunden nachvollzogen werden. Die möglichen Änderungen am Code werden in den folgenden Abschnitten vorgestellt.

3.2.1 Änderungs- und Programmierkonzepte bei Standardsoftware

Customizing

Standardsoftware ist derart konzipiert, dass sie für viele Unternehmen ohne zusätzliche Programmierung einsetzbar ist. Dies geschieht über das - bei den meisten Herstellern so genannte - Customizing. Dabei kann man über Einstellungen die Funktion der Software beeinflussen. Dies geschieht ohne jede Programmierung³⁴. Allerdings muss dazu gesagt werden, dass diese Einstellungen mitunter sehr komplex und voneinander abhängig sind. Siehe Abbildung 3.2, wo als Beispiel der Programmablauf durch Customizing beeinflusst wird.

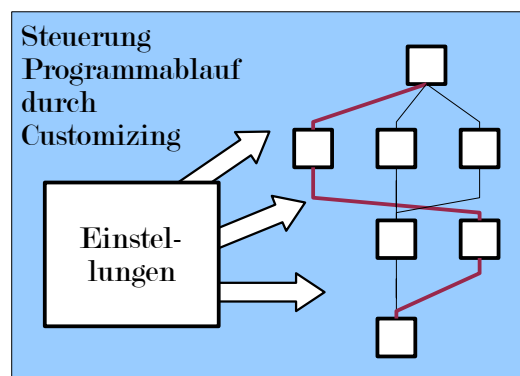


Abbildung 3.2: Technik des Customizing

Erweiterungen

Als Erweiterungen bezeichnet man alle Änderungen an Standardsoftware, die über Customizing hinausgehen. Hier schreiben die Entwickler eigenen Code, allerdings über-

³³Vgl. [Mel04, S.31 ff.] zu Strategien und ihre Bedeutung für Softwareprojekte.

³⁴Eine schöne Darstellung findet sich in [SAP04, S.77 ff].

schreiben sie damit keinen Code in der Standardsoftware. Vielmehr hat der Hersteller der Software einige Punkte vorgesehen, wo klar war bzw. wurde, dass an der betreffenden Stelle einige Kunden eigene Logiken, eigene Systematiken oder eigene Regeln einbauen würden. Hier macht es keinen Sinn, alles über Einstellungen abbilden zu wollen. Es wurde also dem Kunden bewusst die Möglichkeit offen gelassen, in den Ablauf des Programms einzugreifen. Vorteil gegenüber einer direkten Änderung oder Erweiterung im Originalcoding ist, dass der Punkt, an dem das eigene Programm eingefügt wird (wie in Abbildung 3.3 ersichtlich), auch bei Änderungen durch den Hersteller so verbleibt. Im Idealfall muss also nicht alles auf Kundenseite neu erdacht werden, wenn der Hersteller sich entschließt, seine Programme neu zu schreiben³⁵.

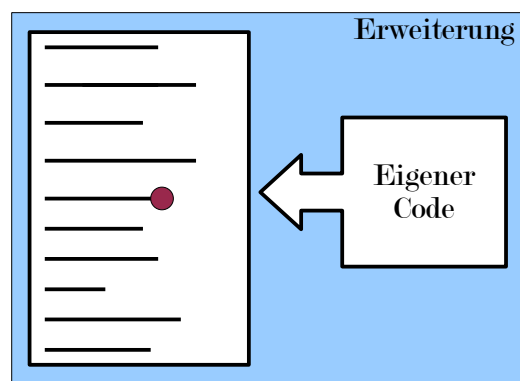


Abbildung 3.3: Technik der Erweiterung von Standardcode mit eigenem Code

Eigenentwicklungen

Eigenentwicklungen sind Programme, die von den Kunden eines ERP Herstellers selbstständig entwickelt werden und welche unabhängig von bestehenden Standardprogrammen ablaufen. Solche Eigenentwicklungen können Schnittstellen zu den Standardkomponenten der ERP Lösung verwenden, sind aber grundsätzlich von diesen unabhängig. Bei diesen Programmen ist es mitunter nötig, einen entsprechenden Abgleich zu machen, um Änderungen im ERP Standard in den Kundenentwicklungen nachziehen zu können. Wichtig für Kundenentwicklungen ist, dass der Hersteller nicht im Nachhinein durch eine Aktualisierung den kundeneigenen Code überschreibt. Das wird dadurch verhindert, dass Kunden bei den meisten Lösungen einen eigenen Bereich in der Programmierumgebung, identifiziert durch eine spezielle Namenskonvention oder durch eigene Präfixe in den Programmnamen, verwenden sollen.

³⁵Eine SAP-spezifische, aber sehr gute Zusammenfassung von Erweiterungskonzepten findet sich auch in [MFMS00, S.46].

Modifikationen

Eine Modifikation ändert den Originalcode des Herstellers in Code, der vom Kunden konzipiert wurde. Solche Änderungen sind insofern heikel, als einerseits der Support des Herstellers für die betreffenden Programmteile nicht mehr übernommen wird, und andererseits ein genaues Verständnis des entsprechenden Programms vorausgesetzt werden muss. Modifikationen sind auch nicht gefeit gegenüber Aktualisierungen, die vom Hersteller kommen. Im Aktualisierungsfall werden diese überschrieben. In den meisten Produkten existieren hier Werkzeuge, mit denen ein solcher Abgleich zwischen Original und Änderung teilweise automatisiert vorgenommen werden kann. In jedem Fall sind Modifikationen bei Standardsoftware zu vermeiden, wenn dies möglich ist.

3.2.2 Auswirkungen und Zusammenfassung

Abgesehen von Layer-orientierten Standardprogrammen³⁶ bieten die meisten Hersteller die Möglichkeit des Customizing, der Erweiterung und der Modifikation, zusätzlich zu den Eigenentwicklungen des Kunden. Diese vier Techniken können also im Software Lifecycle auf Kundenseite eingesetzt werden. Nur der ERP Hersteller hat die Möglichkeit, eine Änderung direkt am Code vorzunehmen, ohne dass dies eine Modifikation wäre.

Zusammenfassend ist es für den Kunden eines ERP Herstellers wichtig, sofern möglich im Standard der Anwendungen zu verbleiben und im Notfall die Technik vom kundeneigenen Entwicklungen zu nutzen. In zahlreichen Standardverfahren für die Projektgestaltung, wie sie bei verschiedenen Unternehmen, bei denen der Autor bereits tätig wurde, eingesetzt werden, ist eine Zustimmung des Projektsteuerungsgremiums für jede einzelne Modifikation an Standardsoftware einzuholen.

Der daraus resultierende Umstand, dass man ein Standardprodukt nicht beliebig ändern soll³⁷, führt meist bei Einführungsprojekten zu folgender Diskussion: Soll sich das Unternehmen in seiner Arbeitsweise an die vordefinierten Prozesse, die von der Software unterstützt werden, anpassen oder soll die Software sich an das Unternehmen bestmöglich anpassen (auch wenn das Einspielen eines neuen Releases dann konstant teurer wird)?

3.3 Prozess der Anpassung

Wird ein ERP System neu eingesetzt, so gilt es, zu evaluieren, wie es am besten angepasst werden kann. Externe Berater als Experten für die ERP Lösung konzipieren hier

³⁶Zum Beispiel ist Microsoft Axapta ein solches.

³⁷Für eine genauere Beschreibung in welchen Fällen dies doch zulässig sein könnte siehe [MFMS00, S.51 ff.].

mit internen Prozessexperten des Unternehmens, wie die im Unternehmen aktuell vorgefundene Situation wohl am besten abgebildet werden kann. Eine aus diesem Prozess abgeleitete Entscheidungsmatrix ist in Abbildung 3.4 zu sehen. Hat eine Anforderung

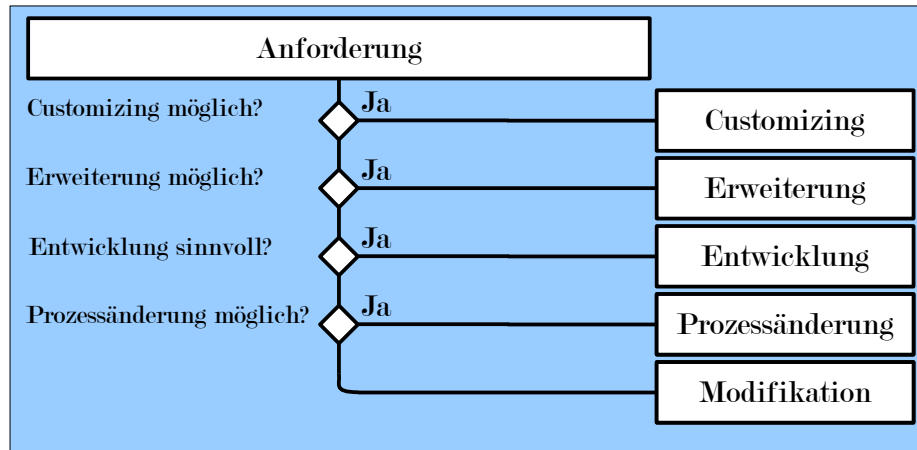


Abbildung 3.4: Typischer Prozess der Anpassung von ERP Standardsoftware

den Weg in das Lastenheft geschafft, so wird bei den meisten Unternehmen nach diesem System vorgegangen:

1. Zunächst wird geprüft, ob die Anforderung rein über Customizing gelöst werden kann. Informationsquellen sind hier die Dokumentation, die Prozessexperten oder auch andere Unternehmen, die eine ähnliche Aufgabenstellung bereits lösen mussten und zu einem Erfahrungsaustausch bereit sind.
2. Wenn die Anforderung nicht über Customizing gelöst werden kann, wird die Möglichkeit einer Erweiterung geprüft. Diese erfordert, wie zuvor erläutert, einen bereits existierenden Anknüpfungspunkt im Coding des ERP Herstellers.
3. Wenn weder eine Erweiterung noch eine Anpassung durch Customizing in Frage kommen, wird meist eine Eigenentwicklung angedacht. Viele Standardprozesse sind aber derart integriert, dass eine Eigenentwicklung einer Neuentwicklung von Programmteilen gleichkommt.
4. Ist auch eine Eigenentwicklung nicht das Mittel der Wahl, sind die Prozessexperten gefragt. Oftmals können durch kleine Änderungen im Unternehmen Abläufe so verändert werden, dass diese besser zur eingesetzten Software passen.
5. Manchmal sind Änderungen eines Ablaufes im Unternehmen nicht gewünscht oder schlicht nicht möglich. In einem solchen Fall ist es angezeigt, eine Modifikation

vorzunehmen.

Man kann erkennen, dass eine Entwicklung im Standardsoftwareumfeld wohl die im Software Lifecycle definierten Schritte durchläuft, dass sich dabei aber noch ganz andere Aspekte ergeben.

Eine Möglichkeit, die Zahl der ungeliebten Modifikationen gering zu halten, ist mitunter eine verstärkte Serviceorientierung. In einer SOA (siehe Abschnitt 2.1.4 sowie 2.2.6) werden von einzelnen Prozessschritten nur die Services genutzt und in einer übergreifenden Anwendung neu zusammengesetzt. So sind mitunter Prozesse mit anderen als den vorgegebenen Abläufen umsetzbar, deren Programmierung andernfalls einer Modifikation bedurft hätte.

3.4 Schnittstellen zum Projektmanagement

In Softwareprojekten mischen sich mitunter die Vorgehensmodelle nach dem Software Lifecycle mit dem Ablauf des Projektmanagements. Im Folgenden wird kurz auf diesen Aspekt eingegangen.

In Abbildung 3.5 wird der Zusammenhang zwischen Projektmanagement und dem

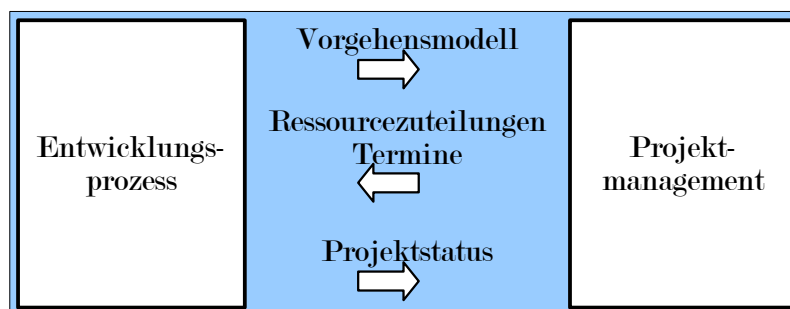


Abbildung 3.5: Informationsaustausch zwischen Projektmanagement und dem Entwicklungsprozeß, aus [Mil89, S.65]

gewählten Vorgehensmodell des Software Lifecycle gezeigt. Diese Interaktionen zwischen Projektmanagement und dem Lifecycle können nur dann gut funktionieren, wenn die Anforderungen, die im Lastenheft vom Kunden beschrieben wurden, sich auch in einzelne Teilbereiche zerlegen lassen³⁸. Auch das Pflichtenheft muss über eine Struktur entsprechend abgegrenzter Aufgaben verfügen, die zu den im Lastenheft eingeführten Teilbereichen passen und welche dezidierten Personen oder Teams zugeteilt werden können.

³⁸Zur Datenqualität von Anforderungen siehe [Spi07, S.95].

3 ERP Software Lifecycle Management

Zusammenfassend sind zwei Faktoren für ein erfolgreiches ERP Einführungs- oder Erweiterungsprojekt besonders wichtig:

- die „Datenqualität“ von Lasten- und Pflichtenheft.
- die Verknüpfung des Projektcontrollings mit den genauen Zielen des Softwareproduktes³⁹, um abweichende Ergebnisse zu vermeiden.

Der Software Lifecycle mitsamt seinen Implikationen auf die spezifischen Anforderungen von ERP Systemen wurde nun erläutert. Die Änderungsstrategien, speziell in einer risiko- und aufwandsminimierenden Art, wurden klargestellt und ihre Auswirkungen dokumentiert. Eine Darstellung der Schnittstellen mit den Prozessen des betrieblichen Projektmanagements hat das Bild von Softwareprojekten im Unternehmen abgerundet. In dem nun folgenden Kapitel wird das tägliche unternehmerische Risiko, sowie das Risiko der Investoren aufgezeigt und es werden Begrenzungen - titulierte als Governance oder IT Governance Strategien - vorgestellt. Diese Regelungen und das Umfeld, aus dem sie entstehen, sind von großer Wichtigkeit für das Verständnis von Standardverfahren, die ab Kapitel 5 vorgestellt werden.

³⁹Im Controlling muss der mitunter sehr komplexen Architektur von Software Rechnung getragen werden, vgl. [Jos03].

4 Corporate Governance versus IT Governance

4.1 Zum Begriff der Corporate Governance

4.1.1 Definition

Corporate Governance lässt sich als ein definierter Rahmen verstehen, innerhalb dessen ein Unternehmen gesteuert werden darf und in welchem es kontrolliert werden muss. In neueren Definitionen, der sogenannten New Corporate Governance, kommen zu diesen Aspekten noch weitere dazu⁴⁰. Frei übersetzt von [Hil05, S.9]:

Neue Corporate Governance ist ein System, durch das Unternehmen strategisch ausgerichtet, integrativ gemanaged und ganzheitlich kontrolliert werden, sowohl in einem ethischen als auch in einem unternehmerischen Sinn, und auf eine für den speziellen Kontext angepasste Weise.

Eine Corporate Governance zu entwickeln bedeutet, ein Regelwerk zu schaffen, nach welchem die Prozesse im Unternehmen ablaufen. Dieses Regelwerk ist mitunter durch Gesetze wie den Sarbanes-Oxley Act (SOX), die 8. EU Richtlinie, Basel II, div. Fachgutachten und Empfehlungen, etc. notwendig⁴¹. Als Beispiele seien in den Abschnitten 4.1.4 und 4.1.5 SOX und die 8. EU Richtlinie kurz erwähnt. In Abschnitt 4.1.6 wird auf den Österreichischen Corporate Governance Kodex Bezug genommen, einer Empfehlung⁴², zugeschnitten auf den Österreichischen Markt.

4.1.2 Corporate Governance Framework

Das Corporate Governance Framework⁴³ wird in Abbildung 4.1 illustriert. Dieses kann als Darstellung des kompletten Umfeldes von Corporate Governance im und

⁴⁰Vgl. die Definition bei [Hau07, S.6].

⁴¹Vgl. eine genauere Erläuterung bei [Hau07, S.19ff.].

⁴²Zum dessen genauem Status sei der Leser auf [Ple05, S.9] verwiesen.

⁴³Es handelt sich um eine zusammenfassende und vereinfachte Darstellung.

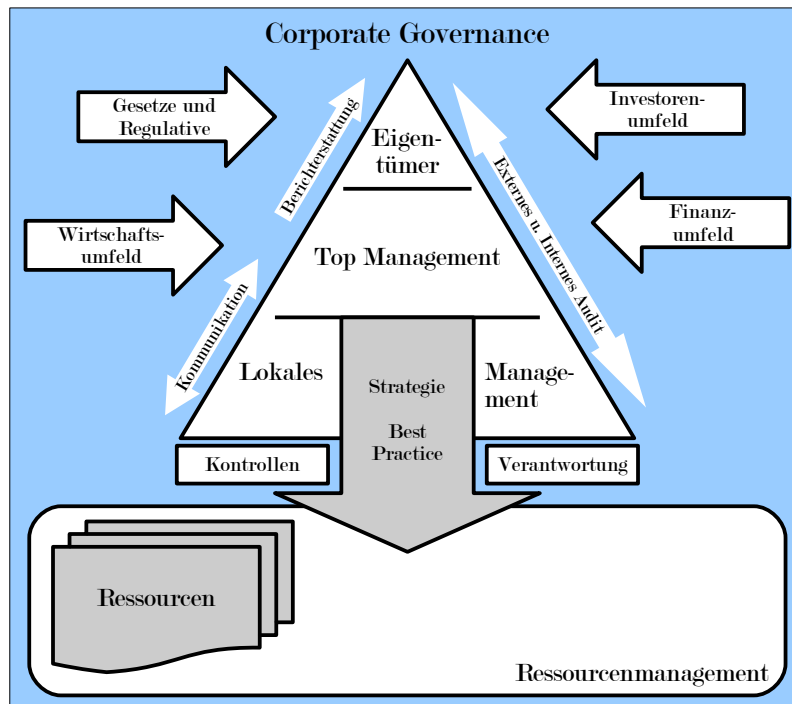


Abbildung 4.1: Corporate Governance Framework nach [Hau07, S.7]

um das Unternehmen verstanden werden. Verschiedene externe Faktoren, wie Gesetze und Regulative, das Wirtschafts-, Finanz- und Investoren-umfeld (und auch Kombinationen aus diesen) wirken sich auf die Möglichkeiten und letztendlich auch auf die Entscheidungen des Managements aus. Über die verschiedenen Managementebenen werden die Ressourcen verwaltet. Dies geschieht nach entsprechenden Strategien oder Best Practices, welche dann genau das Corporate Governance Modell ausmachen.

Die Darstellung berücksichtigt auch die Informationsflüsse zwischen den Gremien. Das bedeutet, dass die Managementebenen untereinander kommunizieren. Außerdem berichtet das Top Management an den oder die Eigentümer. Die interne und externe Prüfung (Revision) wird auf allen Ebenen über Audits realisiert.

Das Framework stellt über die Strategien und Best Practices sicher, dass entsprechende Kontrollen erfolgen und, dass das Management einer gewissen Verantwortung Rechnung trägt.

Die Notwendigkeit, ein solches Framework formell zu definieren ergibt sich schon alleine aus den verschiedenen Interessen der einzelnen involvierten Gruppen. So haben die Eigentümer und auch das Management das Ziel der Gewinnmaximierung, während

der einzelne Arbeitnehmer vielleicht am Ziel der Arbeitsplatzsicherung festhält⁴⁴. Über die im Corporate Governance Kodex festgelegten Grundregeln und -werte werden diese Interessen in der Unternehmensführung berücksichtigt⁴⁵.

4.1.3 Compliance

Unter Compliance versteht man die Einhaltung aller im Corporate Governance Modell definierten Richtlinien, welche über Audits geprüft werden.

Weiters handelt es sich bei Compliance

[...] um eine entsprechende Adressierung der unternehmerischen Risiken.⁴⁶

Unabdingbar ist auch die regelmäßige Selbstkontrolle. Aus [Mül08]:

Ein wichtiger Aspekt von Compliance [...] ist die Überprüfung der Wirksamkeit der eingesetzten Kontrollen. Üblicherweise werden in Firmen nur die wichtigsten Prozesse definiert. Es gibt auch ansatzweise Tätigkeiten, die einen gewissen Kontrollcharakter haben. Was jedoch sehr selten gemacht wird ist die regelmäßige Überprüfung der Durchführung. Man verlässt sich darauf, dass alle sich an die erfassten Vorgaben halten.[...]

Die durch den Gesetzgeber verordnete Selbstkontrolle gibt den Firmen eine Gelegenheit, sich selbst über den Zustand der Prozesse und Kontrollen zu vergewissern. Dabei gewonnene Erkenntnisse ermöglichen, die noch vorhandenen Schwachstellen zu erkennen und zu beseitigen.

Bei der Überprüfung der Compliance eines Unternehmens wird sichergestellt, dass die Regeln und gesetzlichen Vorschriften eingehalten bzw. befolgt werden⁴⁷. Zusammenfassend könnte man Compliance als die Steuerung von unternehmerischen Risiken innerhalb eines definierten Rahmens ansehen, welcher regelmäßig und systematisch geprüft wird.

4.1.4 Sarbanes-Oxley Act

Der Sarbanes-Oxley Act of 2002, kurz SOX, ist ein Resultat der zwischen 2000 und 2005 in den USA aufgetretenen Bilanzfälschungsskandale. Das Gesetz, im Umfang von 66 Seiten und 11 Sektionen, regelt hauptsächlich die Transparenz und Fehlerfreiheit von

⁴⁴Zur Agency Problematik, die den Eigentümer-/Management Grundkonflikt behandelt, siehe [Mit04, S.10 ff.].

⁴⁵Vgl. Modell bei [Hau07, S. 8], auch für einige Prinzipien von Corporate Governance.

⁴⁶[Hau07, S.15]

⁴⁷Vgl. bei [FHG08], dort wird dies noch genauer umrissen.

Bilanzdaten über verschiedene Regulative⁴⁸.

Sämtliche Unternehmen, für die eines der folgenden Kriterien zutrifft, sind verpflichtet, sich an die SOX Kriterien⁴⁹ zu halten:

- Unternehmen mit einem Listing an einer US-amerikanischen Börse
- Unternehmen, die mehr als 500 Aktionäre und im abgelaufenen Geschäftsjahr eine Bilanzsumme von mehr als 10 Millionen US-Dollar besitzen.

Die Bereiche, die hauptsächlich von SOX tangiert werden, sind⁵⁰:

- Projekt- und Budgetplanung sowie Controlling
- Das IT Infrastruktur Management
- Pflege und Wartung
- Security

Die Bereiche ergeben sich aus dem Umstand, dass zu einer Absicherung der genauen Bilanzdaten einerseits das Zustandekommen der Bilanzzahlen maßgeblich ist. Daher wird der Budget- und Controllingbereich verstärkt normalisiert. Andererseits ist der Zugriff, die Wartung und die fälschungssichere Aufbewahrung der generierten Zahlen von größter Wichtigkeit. Eine SOX Einführung hat also auch viele Berührungspunkte mit der IT. Der Zweck dieser Vorschriften ist klarerweise die Verbesserung der Bilanzierungspraxis in Unternehmen.

Damit soll die Unternehmensberichterstattung verbessert und das Vertrauen der Anleger in die veröffentlichten Finanzdaten hergestellt werden. Im Rahmen der Section 404 des Sarbanes-Oxley Acts müssen Unternehmensprozesse beschrieben, definiert und Kontrollverfahren festgelegt werden, die das Risiko des falschen Bilanzausweises minimieren sollen⁵¹.

4.1.5 8. EU Richtlinie

Die Richtlinie 2006/43/EG (auch 8. EU Richtlinie genannt) ist eine Festlegung der Europäischen Union, welche die Abschlussprüfung von Jahresabschlüssen standardisiert⁵².

⁴⁸Vgl. die Gliederung bei [Hau07, S.19].

⁴⁹[Hau07, S.19]

⁵⁰[RC07]

⁵¹[RC07]

⁵²Vgl. weiterführend [Hau07, S.21].

Auch EuroSOX⁵³ genannt tritt diese in den meisten EU Staaten im Laufe des Jahres 2008 in Kraft. Die Ziele der 8. EU Richtlinie ähneln den Zielen des SOX und haben ebenfalls die Intention, die Qualität von Bilanzzahlen zu erhöhen.

Jedes einzelne EU Land hat diese Richtlinie nun in nationales Recht zu übernehmen. Die EU hat allerdings ein paar Freiheiten eingeräumt. Jedes Land darf eine eigene Definition von „wichtigen Firmen“ in die jeweiligen Gesetze einbauen, welche zusätzlich zu den Aktiengesellschaften und Banken zu den Inhalten der Richtlinie verpflichtet werden. Aus diesem Grund werden sich die einzelnen Europäischen Länder in den genauen Vorschriften unterscheiden. Daher und weil die Umsetzung in Unternehmen erst in den nächsten Monaten erste Erfahrungen und Ergebnisse bringen wird, kann an dieser Stelle noch keine genauere Aussage zu EuroSOX getroffen werden.

Zu den Vorgaben von SOX und EuroSOX kann gesagt werden, dass diese weitgehend harmonisieren. Das US-amerikanische Modell (SOX) geht bei der unternehmensinternen Kontrolle der Finanzgebarung jedoch etwas weiter⁵⁴. Das europäische Modell verschärft jedoch die Haftung des Wirtschaftsprüfers erheblich, sodass der Prüfer von Konzernbuchhaltungen für die Prüfung der Einzelabschlüsse von verbundenen Unternehmen haftbar gemacht werden kann⁵⁵.

4.1.6 Der Österreichische Corporate Governance Kodex

Der österreichische Corporate Governance Kodex (ÖCGK) ist eine Empfehlung des österreichischen Arbeitskreises für Corporate Governance. Er adressiert sich primär an Aktiengesellschaften, die an der Börse notieren. In dieser Richtlinie versucht der Arbeitskreis, die für österreichische Unternehmen zutreffenden rechtlichen Vorschriften⁵⁶ mit den jeweiligen Empfehlungen zu kombinieren. Für Unternehmen bringt das den Vorteil, dass idealerweise nicht bei Einführung von Corporate Governance Richtlinien im Unternehmen auf rechtliche Aspekte vergessen werden kann. Andererseits muss „das Rad nicht neu erfunden werden“. Es werden weiterführende Ideen aufgezeigt⁵⁷.

Die Einhaltung des Kodex wird, bedingt durch den Umstand, dass es sich ja nicht um ein Gesetz mit verpflichtendem Charakter handelt, über eine freiwillige Selbstverpflichtung, die Compliance Erklärung, kommuniziert⁵⁸. Dadurch bindet sich das Unternehmen an

⁵³Siehe Bericht bei [EC08].

⁵⁴Vgl. [Bra07, S.125].

⁵⁵Vgl. genaue Vorschau bei [Hau07, S.25].

⁵⁶In Österreich ist das in erster Linie das Gesellschaftsrecht, zu dem laufend entsprechende GesRÄG Novellen beschlossen werden.

⁵⁷Vgl. die Ideen bei [Ple05, S.5 ff.].

⁵⁸Vgl. [Ple05, S.6].

den ÖCGK⁵⁹.

Ziel des ÖCGK ist es, auch hierzulande das Vertrauen des Kapitalmarktes in ein Unternehmen herzustellen. Es gilt als weiter ausbaufähig⁶⁰. Eine empirische Untersuchung bei [Hai03, S.91 ff.] ergab, dass bereits im Jahre 2003 eine große Zahl an Unternehmen, die an der Wiener Börse gelistet waren, zu einer Einhaltung des ÖCGK bereit waren. Manche Autoren berichten von Auswirkungen einer Selbstverpflichtung zu einem Corporate Governance Kodex in einer Höhe bis zu 20%⁶¹.

Generell ist das österreichische Gesellschaftsrecht im Vergleich zu seinen angelsächsischen Pendanten aber als eher hochgradig reguliert zu bezeichnen⁶², wodurch sehr viel mehr Regulative bereits damit abgedeckt werden.

4.1.7 Zusammenfassung

Ein Unternehmen benötigt, um am Markt erfolgreich wahrgenommen zu werden, eine Unternehmensstrategie, die sich an gewissen Kriterien, wie z.B. der Ethik und auch an den beteiligten Umwelten und Personen orientiert. Diese Strategie, gemeinsam mit den Regeln für alle Teile des Unternehmens, zu verhandeln, zu Papier zu bringen und auch die entsprechende Kontrolle zu sichern, ist im Interesse aller Beteiligten. Das Corporate Governance Framework veranschaulicht, wie sich externe und interne Faktoren auf das Management von Ressourcen auswirken können. Die externen Faktoren wurden in den letzten Jahren in Amerika und in der EU auch legislativ formalisiert und dienen der Aufstellung „gemeinsamer Spielregeln“ für Unternehmen. Governance an sich kann für Unternehmen eine Chance sein, Prozesse zu verbessern, aber auch eventuell ohnehin schon existierende Leitsätze niederzuschreiben.

Für die nächsten Jahre wird erwartet, dass sich europäische Vorschriften immer weiter an die US-amerikanischen annähern⁶³.

Ein sehr oft begangener Fehler von Unternehmen ist es, Governance als eine weitere Einschränkung zu sehen und das Management nicht danach auszurichten⁶⁴.

⁵⁹Für eine weitergehende Analyse mit Beispielen von Unternehmen im ATX, dem wichtigsten österreichischen Aktienindex, vgl. bei [Ple05, S.5 ff.].

⁶⁰Vgl. Statements in [Ple05, S.36 ff.].

⁶¹Siehe dazu ausführlich bei [Sta07, S.40] sowie den dort genannten Quellen.

⁶²Siehe [Hai03, S.13].

⁶³Siehe die Gedankengänge in [Hai03, S.86].

⁶⁴Dies ist die Grundaussage in [SL08].

4.2 Zum Begriff der IT-Governance

4.2.1 Definition

IT Governance hat sich historisch seit der Mitte der 90er Jahre des 20. Jahrhunderts aus der Corporate Governance und den strategischen Informationssystemen (IS) abgeleitet. Strategische IS befassen sich hierbei mit dem optimalen Einsatz von IT Systemen, dem Erkennen von Bedrohungen und strategischen Möglichkeiten sowie den zukünftigen Entwicklungen bei IT Systemen⁶⁵.

IT Governance weist ähnliche Ziele und Bedingungen auf wie Corporate Governance und wird daher zumeist als Teilmenge dieser gesehen⁶⁶. In [Pop07, S.3] wählt der Autor ein Zitat, welches IT Governance wohl am besten beschreibt. Dieses stammt vom IT Governance Institute, einer Organisation, die sich ausschließlich mit IT Governance beschäftigt⁶⁷.

„Das Hauptziel von IT Governance ist es, die Anforderungen an die IT sowie die strategische Bedeutung von IT zu verstehen, um eine optimale Unterstützung der Unternehmensziele sicherzustellen und Strategien für die zukünftige Erweiterung des Geschäftsbetriebs zu schaffen.“ [Div03, S.6]

IT Governance fasst Grundsätze, Verfahren und Maßnahmen zusammen, welche alleamt sicherstellen sollen, dass die Geschäftsziele mit den gegebenen Ressourcen erreicht werden; dies unter einer ständigen, angemessenen Überwachung⁶⁸. Ergänzend zur Corporate Governance geht es bei der IT Governance darum, die Unternehmensziele und die -strategie durch das Management der IT Ressourcen zu unterstützen.

IT Governance gründet sich auf drei Säulen⁶⁹:

- die IT Organisation
- die IT Managementprozesse
- die Wertschöpfung

⁶⁵Vgl. [Pop07, S.5 ff.]

⁶⁶Aus einem sehr guten Standardwerk: [Pop07, S.3]

⁶⁷Eine weitere, hier nicht ausgewählte Definition wäre bei [SL08] zu finden.

⁶⁸Vgl. [TAM05]

⁶⁹Ausführlicher in: [SL08]; andere Autoren (z.B.[TAM05]) sehen die drei Säulen Struktur, Prozesse und Kommunikation als Basis von IT Governance.

IT Organisation

In dem verwendeten IT Governance Modell muss definiert sein, aus welchen Personen die IT Organisation aufgebaut ist, wie die Hierarchien aussehen und wer welche Zuständigkeiten innehat. Zu Zeiten einer zentralen IT Organisation war diese Frage noch leicht zu beantworten. Nachdem immer mehr Unternehmen zu Matrix- oder anderen alternativen Organisationsformen konvertieren, oder auch Teile der IT extern ausgelagert werden, ist es wichtig, zu jedem Zeitpunkt zu wissen, wie die IT Organisation denn aussieht. Oftmals passt auch die IT Organisation nicht besonders gut zur restlichen Organisation und somit zur Ausrichtung des Unternehmens. Hier kann man bei der Einführung eines neuen Governance Modells auch nachjustieren, wenngleich nur mehr in geringerem Umfang.

Die wichtigsten Verantwortlichen sind in diesem Zusammenhang der CIO und der Compliance Officer (CO). Sie sind die Hauptverantwortlichen und -treiber der IT Governance.

IT Managementprozesse

Die IT Managementprozesse sind jene Prozesse, welche die IT selbst am Leben erhalten, sie kontrollieren, lenken und letztlich (über die Abrechnung) auch finanzieren. Diese Prozesse müssen zur Unternehmensstrategie passen (vgl. Corporate Governance) und in dem dargelegten Modell enthalten sein.

IT Wertschöpfung

Die Prozesse der IT Wertschöpfung sind all jene Prozesse, bei denen die IT den Prozessen dient, welche zum Output des Unternehmens beitragen. Das sind also all jene Prozesse, wo operativ von der IT am Unternehmen gearbeitet wird (vgl. Managementprozesse: hier arbeitet die IT an der IT). Bei den Prozessen der Wertschöpfung ist es wichtig, dass die IT diese auf einer guten Preis-/Leistungsbasis erzielt. Sekundär ist, ob die IT diese Leistung selbst erbringt oder fremd vergibt.

IT Governance Lifecycle

Die IT Governance kann ähnlich dem Software Lifecycle (vgl. Kapitel 3.1) in einem IT Governance Lifecycle abgebildet werden. Dieser ist in Abbildung 4.2 ersichtlich.

Zunächst wird die Ausgangslage untersucht und definiert, welche Ziele im IT Bereich gesetzt werden sollen. Diese Ziele und ihre gewünschte Wirkung können durch externe Faktoren (Vorschriften, an die sich das Unternehmen halten muss (Corporate Governance), aber auch durch interne Faktoren (Kontrolle, Flexibilität, ...)) bestimmt werden.

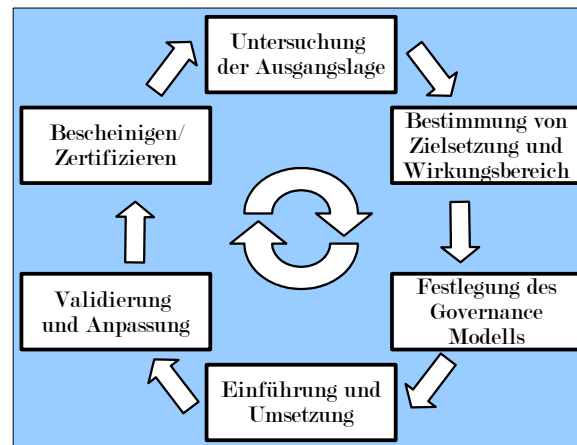


Abbildung 4.2: IT Governance Lifecycle, aus [SL08]

Nach Evaluierung der Ziele versucht das IT Management, ein geeignetes Governance Modell festzulegen, welches dann sukzessive eingeführt und umgesetzt wird.

Im Schritt der Validierung und Anpassung folgt die Qualitätskontrolle. Hier besteht die Möglichkeit, das Modell noch genauer auf den gewünschten Wirkungsbereich abzustimmen.

Im Schritt der Bescheinigung wird bei Bedarf eine Zertifizierung der Zielerreichung vorgenommen. Abhängig von gesetzlichen Erfordernissen kann diese Phase auch weggelassen werden.

4.2.2 IT Governance Framework

Will man die IT Governance, wie die Corporate Governance, in ein Framework eingliedern (vgl. Abschnitt 4.1.2), so muss das Corporate Governance Framework mit dem IT Governance Framework verbunden sein. Dies ist ohnehin naheliegend, zumal die beiden Governance Modelle doch an der gleichen Strategie im gleichen Unternehmen arbeiten. Abbildung 4.1 veranschaulicht diesen Zusammenhang.

Man kann erkennen, dass sich die Unternehmensstrategie, welche in der Corporate Governance festgeschrieben wurde, über die Strategie und die Compliance Vorgaben direkt auf die IT Governance auswirkt. Die IT Governance Strategie legt hier den Entscheidungsspielraum für die Manager fest, wie die IT Ressourcen auf die Wertschöpfungs- und Managementprozesse verteilt werden.

Der CIO ist hier das Bindeglied zwischen der IT Governance und dem Top Management. Er setzt die Vorgaben gemeinsam mit seiner Organisation um, hat aber Gestaltungsspielraum. Es zählt auch zu den Aufgaben der IT Governance, sich strategischen Themen

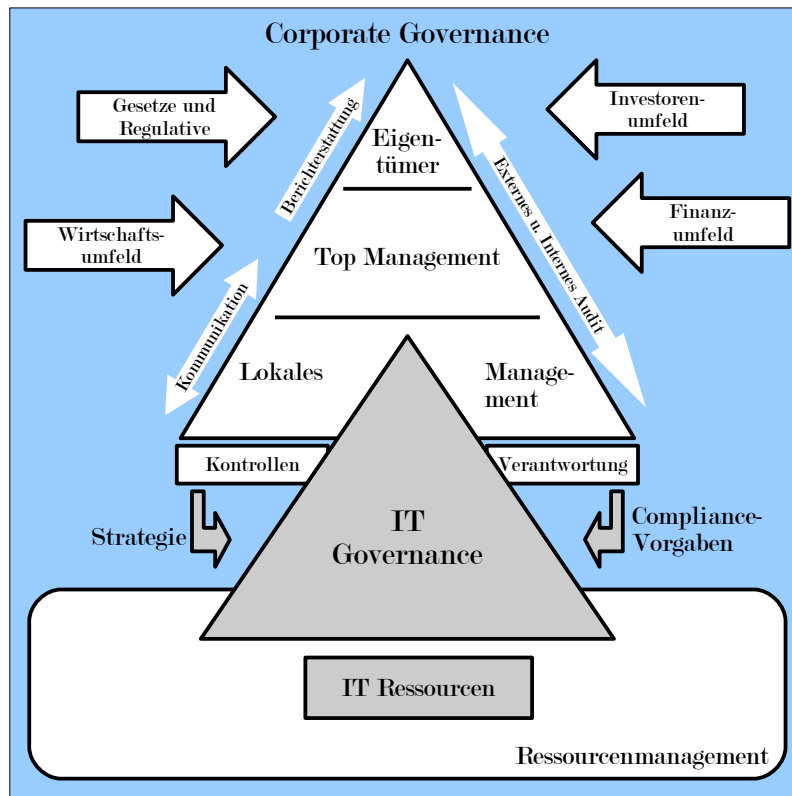


Abbildung 4.3: IT Governance Framework, aus [Hau07, S.11]

zu widmen. Diese umfassen die Ausrichtung der IT an der Unternehmensstrategie aber auch die proaktive Herangehensweise an neue Themen (neue Technologien, etc.)⁷⁰.

4.2.3 Frameworks

Für die Gestaltung der IT Governance existieren eigene Frameworks und Rahmenwerke, die sich zumeist auf die Säule der IT Managementprozesse beschränken⁷¹. Die folgende Festlegung ist zweckdienlich:

Frameworks wie ITIL und COBIT beschreiben dabei nicht die konkrete Umsetzung sondern definieren den Rahmen, die Anforderungen und die Ergebnisse des IT Managements. Die Ausformulierung [...] muss in Einklang mit den branchen- und unternehmensspezifischen Geschäftsprozessen erfolgen⁷².

⁷⁰Mehr zu den Rollen des CIO und des Managements bei [Hau07, S.11 ff].

⁷¹Diese Abgrenzung wird in [SL08] getroffen.

⁷²[TAM05]

Die im Zitat genannten Frameworks, ITIL und COBIT gelten als die anerkanntesten⁷³ bezüglich des praktischen Einsatzes.

Für das jeweilige Unternehmen bieten die Frameworks Prozessmodelle an, die nicht nur Regularien enthalten, sondern auch messbare Größen⁷⁴, um die Prozesse und ihre Ergebnisse auch überwachen zu können. Diese Prozesse haben sich in der Praxis derart bewährt, dass sie, wenn sie gelebt werden, die Standards einer guten IT Governance treffen.

Ein kurzes Beispiel: Lebt man einen verantwortungsvollen Umgang mit Berechtigungen im Unternehmen, so muss man sicherstellen, dass von dem Personenkreis, der Berechtigungen auf kritischen Systemen vergeben kann, niemand im Alleingang kritische Berechtigungen nur für sich selbst vergeben darf.

Da sich Kapitel 5 ausschließlich mit dem ITIL Framework befassen wird, welches nicht ganz so ausführlich, jedoch weit bekannter ist, soll im Rahmen dieses Abschnitts COBIT vorgestellt, sowie der Querschnitt zwischen COBIT und ITIL vollzogen werden.

COBIT

Allgemein gilt COBIT als das von der Breite her umfangreichste Standardmodell im Rahmen der IT Governance⁷⁵. Es wird vom IT Governance Institute betreut und weiterentwickelt. Oberstes Prinzip bei COBIT ist die gemeinsame Betrachtung von IT und dem Rest des Unternehmens, also dem IT Governance Framework mit seinen Umwelten. COBIT hat von einem Kontroll- und Risk Management Modell der IT eine Transition zu einem geschäftsorientierten IT-Steuerungsmodell durchlaufen. Bei COBIT wird wieder ein Lebenszyklus beschrieben, der sich unter den IT Governance Zyklus (Kapitel 4.2.1) einzuordnen vermag. Dieser Zyklus geht von den Erfordernissen, in IT Ressourcen zu investieren bis zu deren Anwendung in IT Prozessen, um Unternehmensinformationen zu liefern. Diese Informationen können wieder zu Investitionen führen und so schließt sich ein Kreislauf, der - weniger ausführlich - dem der IT Governance aus Abbildung 4.2 nicht unähnlich ist⁷⁶.

COBIT umfasst folgende Teilbereiche⁷⁷:

- „Plan & Organise“, Planung und Organisation
- „Acquire & Implement“, Beschaffung & Einführung

⁷³Siehe [TAM05]

⁷⁴Vgl. [Hes02, S.8]

⁷⁵Siehe dazu [Hes02, S.67]

⁷⁶Siehe weiterführend [Hau07, S.45 ff]

⁷⁷Vgl. Aufstellung bei [Hes02, S.70] auf Deutsch und [Pop07, S.15 f.] auf Englisch.

4 Corporate Governance versus IT Governance

- „Deliver & Support“, Auslieferung & Unterstützung
- „Monitor & Evaluate“, Überwachung

Über die Informationen, die in der Überwachung der Prozesse Planung, Beschaffung und Auslieferung gewonnen werden, können IT Ressourcen und auch wiederum neue Planungen wirkungsvoll gesteuert werden.

Im Bereich **Planung und Organisation** geht es darum, strategisch und adäquat zu planen und diese Planungen mit dem Management abzustimmen. Die Organisation muss nach geeigneten Regeln ablaufen. Sie muss die entsprechenden Ressourcen zur Verfügung haben, dies effizient zu tun.

Der Bereich **Beschaffung und Einführung** ist für die Umsetzung der in der Planung beschlossenen Strategie verantwortlich. Hier beschäftigt man sich mit der Wartung und den Änderungen an Systemen sowie an Prozessen.

Der COBIT Bereich **Auslieferung & Unterstützung** ist für die Datenaufbereitung, die Lieferung von Leistungen und die Kundenbetreuung (den Support) zuständig.

Im Bereich der **Überwachung** werden IT Prozesse bewertet und idealerweise unabhängig zertifiziert.

Jeder dieser Bereiche untergliedert sich in eine Reihe von Musterprozessen, die dann genaue Mindestanforderungen besitzen.

Für das COBIT Framework existiert weiters eine eigene Road-Map zur Erfüllung einer SOX Compliance, welche Unternehmen dabei helfen soll, SOX Compliance zu überschaubaren Kosten zu erreichen. Es ist jedoch nur als Richtwerk zu sehen. Seine Einhaltung bedeutet noch nicht automatisch, dass ein Unternehmen voll SOX Compliant ist⁷⁸.

COBIT und ITIL

ITIL deckt, wie in Kapitel 5 gezeigt werden wird, nicht alle Bereiche von COBIT hinreichend ab. Für das Verständnis von Change Request Management, dem Hauptforschungsgebiet der Arbeit, ist allerdings nur ein kleiner Teil der Materie relevant. In [Pop07, S.14 ff.] versucht der Autor eine Zuordnung gängiger Frameworks zu den einzelnen Teilbereichen von COBIT und untersucht diese auf deren Abbildungsverhalten im COBIT. Konkret geht es dem Autor um die Frage, ob ITIL eine vollständige Abbildung des COBIT Teilbereichs „Deliver & Support“ (Auslieferung & Unterstützung) ist oder ob blinde Flecken (können nicht direkt abgebildet werden) vorhanden sind.

Der Autor kommt in [Pop07, S.73] zu dem Schluss, dass ITIL die von COBIT definierten Prozesse abdeckt und verfeinert.

⁷⁸Vgl. weiterführend mit vielen Details bei [Bra07, S.115 ff.]

Die Untersuchung der einzelnen ITIL Prozesse auf ihre Abbildung von COBIT ergab, dass sich sämtliche wichtigen Anforderungen und Ziele von COBIT in diesem Bereich in ITIL wieder finden. ITIL kann daher als Referenzmodell im Bereich „Deliver & Support“ für eine Implementierung nach dem COBIT-Modell herangezogen werden⁷⁹.

Aus diesem Grund ist eine Betrachtung von ITIL alleine für eine spätere Eingliederung des Themas des Change Request Managements in eine übergreifende, breitere IT Governance Strategie möglich. Es sind dadurch keine Einschränkungen zu befürchten.

4.2.4 Einführung und Zertifizierung

Einführung

Die Umsetzung von IT Governance im Unternehmen erfolgt heute zumeist auf Basis von international abgestimmten Rahmenwerken und Standards⁸⁰.

Es zeigt sich in der Praxis jedoch, dass der Aufwand, IT Governance umzusetzen, vielfach unterschätzt wird. Das liegt wohl auch daran, dass die vorhandenen Frameworks wie ITIL nicht branchenspezifisch formuliert sind und daher die entsprechende Auslegung und die Best-Practices teuer als externe Beratung zugekauft werden müssen⁸¹. Man kann auch Unterschiede in der Auffassung von IT Governance sehen. Manche Unternehmen sehen IT Governance als die Aufgabe, sich an SOX oder EuroSOX Standards halten zu müssen. Mehr als die „Erfüllung der Standards“ wird in der Planung nicht angedacht. Dies kann sich als Fehler erweisen. IT Governance lebt davon, dass im Top Management, ausgehend von einer Corporate Governance eine Strategie definiert wird, die über IT und bis zu den einzelnen Prozessen durchgängig ist. Ansonsten hält sich die IT an Prozesse, die im Management als störend oder auch nicht beachtenswert gesehen werden. Gerade vom obersten Managementgremium muss aber die Überwachung der entsprechenden Prozesse erfolgen, da eine nicht kontrollierte (IT) Governance Strategie zahnlos ist:

IT-Governance wird in den meisten Fällen auf den Compliance-Aspekt im Kontext mit dem COBIT Framework reduziert. Das Verständnis von IT-Governance als ein IT-Managementsystem wird häufig vernachlässigt wenn nicht sogar gänzlich unberücksichtigt⁸².

⁷⁹Grammatikalisch richtiggestellt, ansonsten wörtlich übernommen aus: [Pop07, S.73]

⁸⁰[FHG08]

⁸¹Als Exkurs sei eine Lektüre von [Lie06, S.38 ff.] empfohlen.

⁸²[SL08]

Manche Unternehmen fokussieren sich aber auch zu sehr auf Einzelprozesse und übersehen dabei das Gesamtkonzept⁸³.

Risiko

Wie im Kapitel zur Corporate Governance (4.1) bereits angeschnitten, ist das Ziel dort unter anderem, dass der Kapitalanleger weniger Risiko bei der Investition seiner Mittel hat, weil er sich auf die festgelegten Standards der Unternehmens verlassen kann. Bei IT Governance ist das nun ähnlich. Das Top Management hat durch genaue Vorgaben und entsprechende Kontrollen die Möglichkeit das Risiko zu minimieren.

Bei [Sta07, S.172 ff.] werden einige Erkenntnisse aus IT-Governance Risikoüberlegungen gewonnen, die im Folgenden zur weiteren Sensibilisierung gegenüber IT Governance dienen sollen.

- IT-Risikomanagement ist Chefsache
- Risikomanagement ist Chancenmanagement mit positiv darstellbarem Kapitalwert
- IT generiert Wertschöpfung
- Das Unternehmen unterliegt einer externen Risikobewertung
- Die Bedeutung der IT als kritischer Erfolgsfaktor nimmt weiter zu
- Standards und Normen verlangen eine quantitative Risikobewertung
- Die rechtliche Relevanz beim Einsatz der IT im Unternehmen wird strenger reguliert und geprüft

Durch die zunehmende Bedeutung der IT im Unternehmen gehen immer mehr Risiken von diesem Unternehmensteil und den von ihm administrierten Ressourcen aus. Das IT ist keine reine Dienstleistungsorganisation für die administrativen Abläufe im Unternehmen, sondern generiert Wertschöpfung. Diese Umstände führen dazu, dass die Risiken in der IT Abteilung eines Unternehmens hinreichend berücksichtigt werden.

Zusammenfassend hilft im IT Bereich eine stimmige und gelebte IT Governance das Gesamtrisiko des Unternehmens zu verringern und trägt letztlich positiv zur Wertschöpfung bei. Mitunter ist allerdings eine hohe Grundinvestition nötig.

⁸³Vgl. bei [TAM05]

Zertifizierung

Es gab bisher im Kontext der ITIL und COBIT Rahmenwerke lediglich Personenzertifizierungen, das heißt, dass die einzelnen Frameworks Prüfungen anbieten, bei denen sich Projektmitarbeiter ein entsprechendes Wissen über die jeweilige Methode bescheinigen lassen konnten (z.B. bei ITIL in drei Stufen vom Foundation Level zum Practitioner). Unternehmen nach ITIL oder COBIT zu zertifizieren war nicht möglich. Lediglich bei COBIT ist eine Zertifizierungs-ähnliche Form des Audits vorgesehen.

Mit den Normen BS15000 (seit 2002) von der BSI (British Standards Institution) und ISO 20000 wird versucht, auch Unternehmen zertifizierbar zu machen⁸⁴. Schwerwiegende Bedeutung hat der englische Standard allerdings nicht, sodass die ISO 20000 zu einer Standardzertifizierung für Unternehmen aufsteigen dürfte.

Sie beinhaltet zum einen Anforderungsdefinitionen an das IT Servicemanagement für die verantwortlichen Akteure und zum anderen einen „Code of practice“, der auch einen Leitfaden für Wirtschaftsprüfung darstellt⁸⁵.

Eine weitere Konsolidierung ist in der Form zu erwarten, dass ITIL und COBIT in ihrem Schnittbereich (siehe Abschnitt 4.2.3) weiter angenähert und als Ganzes dann an die ISO 20000 angepasst werden⁸⁶.

4.3 Zusammenhang zwischen Corporate Governance und IT Governance

Das Top Management von großen Konzernen muss in der heutigen Zeit, nach Skandalen wie der ENRON Pleite in den USA im Jahr 2002, immer wieder darlegen können, dass es keinen hochriskanten und vor allem illegalen Geschäften nachgeht. Um diese Aussage untermauern zu können, empfiehlt sich die Etablierung eines Corporate Governance Kodex.

Durch die steigende Bedeutung der IT im gesamten Unternehmenskontext ist der Geschäftserfolg vermehrt von dieser abhängig. Somit werden auch immer größere Teile des Gesamtrisikos des Unternehmens in die IT verlagert. Dies hat zur Folge, dass neben einer funktionierenden Unternehmensstrategie auch eine korrekte IT Strategie angezeigt ist. Der IT Governance Kodex muss Mittel und Prozesse kennen, die einerseits nach den Unternehmenszielen ausgerichtet sind und andererseits das Risiko innerhalb der IT

⁸⁴Siehe weiterführend bei [TAM05]

⁸⁵[FHG08]

⁸⁶Die Mehrheit der Autoren sieht dies wie [FHG08]

4 Corporate Governance versus IT Governance

minimieren (und somit im Gesamtunternehmen).

Die IT Governance ist in die gleiche Umgebung eingebettet wie die Corporate Governance und wird über die Strategie und die Compliance Vorgaben von dieser gesteuert. Innerhalb der beteiligten Organisationen, also der IT und dem Top Management, ist eine entsprechende Kontrolle unabdingbar, um den Erfolg der gesamte Governance Architektur sicherstellen zu können.

Zur weiteren Erklärung einige beispielhafte Fragestellungen der Corporate- und IT-Governance im Vergleich⁸⁷:

Fragestellung aus der Corporate Governance	IT Governance Fragestellung
Wie können Investoren die Manager dazu bringen, ihnen einen Teil der Profite abzutreten?	Wie bekommen die Top Manager die IT Organisation und den CIO dazu, für Sie geschäftlich nützlichen Output zu liefern?
Wie stellen die Investoren sicher, dass die Manager ihres Unternehmens das Geld nicht stehlen oder in schlechte Projekte investieren?	Wie kann das Top Management sicherstellen, dass der CIO und die IT Organisation ihr Geld nicht stehlen oder in schlechte Projekte investieren?
Wie können Investoren die Manager kontrollieren?	Wie kann das Top Management die IT Organisation und ihren CIO kontrollieren?

Wie man sehen kann übertragen sich hier die Ziele des Gesamtunternehmens in abgewandelter Form auf die IT.

IT- wie auch Corporate Governance werden Top-Down, also vom Management aus, in einem Unternehmen ausgerollt, beginnend bei der Corporate Governance.

Maßnahmen zur Risikosteuerung, induziert durch die verschiedenen Umwelten des Unternehmens, wurden hiermit erläutert. Es wurden die Begriffe Governance und IT Governance in Bezug gesetzt und die Abhängigkeiten diskutiert. Gesetzliche Regelungen wie auch freiwillige Verpflichtungen von und für Unternehmen haben ebenso eine Nennung erfahren wie Frameworks, die eine Umsetzung von Governance Strategien erleichtern. Mit der Behandlung von Absicherungsmechanismen schließt das Kapitel.

In dem nun folgenden Teil der Arbeit wird die IT Infrastructure Library mit ihren Ansätzen und Prozessmodellen vorgestellt. Die Grundbegriffe, die spezifischen Ziele, Rollen und Prozessdetails von Change Management werden vorgestellt und mit einem

⁸⁷[Bra07, S.12], übersetzt vom Autor

4 Corporate Governance versus IT Governance

Fahrplan und Indikatoren abgerundet.

5 IT Infrastructure Library

5.1 Definition und Einführung

ITIL steht als Abkürzung für die IT Infrastructure Library, zu Deutsch wohl am besten mit „Bibliothek zur IT Infrastruktur“ zu übersetzen. ITIL ist eine stetig wachsende Publikation des OGC, des Britischen Office of Government Commerce. Die Publikation gliedert sich je nach Version in viele einzelne Bücher.

Generell ist anzumerken, dass die ITIL zwar sehr umfangreich ist, jedoch auf Grund einer mangelnden Strukturierung, die auch durch Bemühungen in der jüngeren Vergangenheit nicht verbessert werden konnte, unübersichtlich und nicht anwendungsfreundlich ist⁸⁸.

Zusammengefasst, der Inhalt ist gut, die Übersicht im original ITIL Text mangelhaft. Zahlreiche erschienene Publikationen mit Praxisbezug erleichtern den Umgang mit der Materie.

5.1.1 Zielsetzung

Der Fokus liegt in ITIL nicht auf einem kompletten IT Governance Modell sondern behandelt in der Regel nur Teilbereiche aus COBIT (siehe dazu Abschnitt 4.2.3).

Man konzentriert sich auf die Publikation von Informationen, die ein Unternehmen zum Aufbau und Betrieb von IT Dienstleistungen benötigt. Dies vor dem Hintergrund, dass Unternehmen in zunehmendem Maße von ihrer IT abhängig sind⁸⁹.

5.1.2 Teilbereiche

Das ITIL Framework gliedert sich⁹⁰, wie in Abbildung 5.1 ersichtlich, in folgende Teilbereiche:

⁸⁸[Hes02, S.52]

⁸⁹Sehr gut beschrieben in: [Hes02, S.50 ff.].

⁹⁰Dies betrifft die aktuelle, im Oktober 2008 verfügbare Version von ITIL, Version 3, welche im Jahr 2007 publiziert wurde. In den Vorversionen wurde der Cycle, also die Kreiskonzeption, noch nicht verwendet.

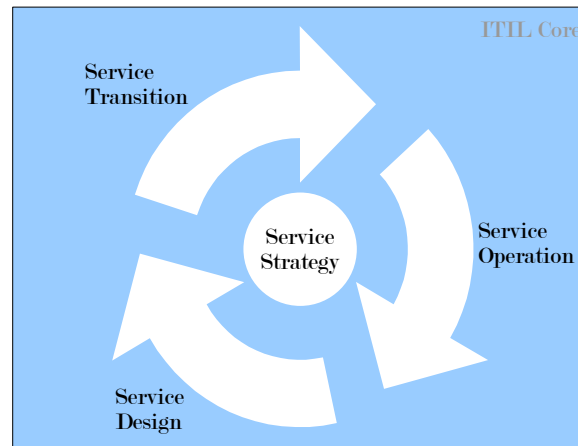


Abbildung 5.1: ITIL Core, vereinfacht, nach [Fan07, S.5]

- Service Strategy, zu Deutsch: Servicestrategie
- Service Design, zu Deutsch: Servicedesign
- Service Transition, zu Deutsch: Serviceeinführung und -überleitung
- Service Operation, zu Deutsch: Servicebetrieb
- Continual Service Improvement, zu Deutsch: Laufende Serviceverbesserung (nicht in der Abbildung verzeichnet)

Die Phasen Design, Transition und Operation folgen einem Kreislauf, dem Kreislauf der **fortlaufenden Serviceverbesserung** (Continual Service Improvement), welcher den gesamten ITIL Kern (Core) umfasst. Wie im Software Lifecycle (siehe Abschnitt 3.1) könnte man auch hier den kompletten Ablauf als eine Spirale, bei der die Qualität der Services immer höher wird, zeichnen. Die Serviceverbesserung hilft aber nicht nur bei der Qualitätsverbesserung, sondern auch bei der Verbesserung der Möglichkeiten der Organisation als Ganzes. Schließlich gehört zur Service Improvement Komponente auch noch ein geschlossener Feedbackzyklus nach dem Plan-Do-Check-Act Model (PDCA)⁹¹. Die **Servicestrategie** bildet gewissermaßen das Fundament des ITIL Kerns. Sie steht im Mittelpunkt des Kreislaufes, weil sie sich klassischerweise nicht mit jeder Änderung im Unternehmen ändert. Alle anderen Bereiche werden klarerweise von der Strategie beeinflusst. Innerhalb der Strategie finden sich Dinge wie die Einschätzung der Marktentwicklung, der Servicekatalog und finanzielle Einschätzungen und Abwägungen zu neuen

⁹¹Vgl. [Fan07, S.7] und weiterführend [Hau07, S.51 ff.]

Geschäftsgelegenheiten für die IT Organisation. Für bereits bestehende Strategien ist eine Möglichkeit zum Review der Strategie vorgesehen⁹².

Der Teilbereich des **Service Designs** enthält Gedanken und Methoden zum Design und zur Entwicklung von Services. Hier geht es um Designgrundsätze und um das Formen von integrierten Serviceangeboten. Wichtig ist in diesem Bereich auch, dass die neuen Services in Einklang mit bestehenden Regularien und Standards konzipiert werden⁹³.

Im Bereich der **Service Transition** werden Mittel und Wege vorgestellt, neue und geänderte Services auszurollen und dann an die operativen Prozesse zu übergeben. Hier geht es auch um die Schnittstellen von Servicemanagement zu Programm-, Release- und Risikomanagement. Beim Ausrollen von Services muss eine Balance zwischen Risikoscheue („Never change a running system“) und Innovation gefunden werden. Das in dieser Arbeit behandelte Change Management ist auch im Teilbereich Service Transition beschrieben⁹⁴.

Die **Service Operation** betrifft einen Bereich im Unternehmen, der meist so beschrieben wird, dass er „einfach funktionieren muss“. Diese Prozesse der IT Organisation kümmern sich um den Betrieb und die Wartung der von der IT zur Verfügung gestellten Services. ITIL will in diesem Bereich eine Effektivität sicherstellen. Ein weiterer, sehr wichtiger Punkt ist die Sicherstellung einer Kontinuität, obwohl sich Teile des Umfeldes laufend ändern (neue Hard- und Software, neue Technologien; das alles während Altbewährtes „einfach laufen muss“)⁹⁵.

ITIL versteht sich als öffentlich verfügbares Framework wie COBIT (siehe Abschnitt 4.2.3). Dies hat den Vorteil, dass die in ITIL angeregten Prozesse frei zugänglich sind (sieht man von dem vergleichsweise hohen Buchpreis ab). Es fallen keinerlei Nutzungs- oder Lizenzgebühren an. Es war und ist daher vielen Unternehmen möglich, in diesem Umfeld Erfahrungen zu sammeln. Dadurch konnten ITIL Prozesse u.a. weiter verbessert werden. Zu jedem der fünf vorgestellten Themenbereiche gibt es ein eigenes Buch, welches im Falle von Service Transition ([Fan07]) etwa 250 Seiten umfasst.

5.1.3 Zertifizierungen

Wie in Abschnitt 4.2.4 bereits erwähnt gibt es für ITIL selbst lediglich eine Personenzertifizierung in drei Levels. Alle drei Zertifizierungen haben vorausgehende Seminare, an deren Abschluss eine Prüfung steht.

⁹²Vgl. dazu [Fan07, S.6].

⁹³Vgl. [Fan07, S.6]

⁹⁴Nach früherem ITIL Wording wäre dieser Bereich im Service Support zu finden, vgl. [Bon05, S.40 ff.].

⁹⁵Vgl. [Fan07, S.6 ff.]

- ITIL Foundation: Diese Zertifizierung beinhaltet Grundlagen aller ITIL Disziplinen mit einer Herausstellung ihrer Verbindung zur IT Organisation. Es ist sehr wünschenswert, wenn in einem Unternehmen alle IT Mitarbeiter eine solche Zertifizierung bzw. gleichwertige Kenntnisse erlangen⁹⁶. Zumal ITIL aber auch ein Mittel ist, um die IT Governance Strategie zu leben, ist es von entscheidender Bedeutung, dass sich auch die Manager diesem Thema angemessen widmen. Das bedeutet, dass ein Besuch durch Manager und Mitarbeiter aus dem IT Bereich sinnvoll ist.
- ITIL Service Manager: Bei einer Ausbildung zum Service Manager wird bei den einzelnen ITIL Kernbereichen ins Detail gegangen. Der Inhalt ist bunt gemischt⁹⁷:
 - Analyse und Beurteilung einer Organisation
 - Detailbetrachtungen zur Rolle des ITIL Service Managers sowie seines Arbeitsumfeldes
 - Anforderungen an die Funktion eines Service Desks sowie an alle operativen Prozesse (Service Support): Incident Management, Problem Management, Configuration Management, Change Management, Release Management
 - Detaillierte Darstellung aller taktischen Prozesse (Service Delivery) in einem Unternehmen und ihre Beurteilung (Service Level Management, Financial Management, Availability Management, IT Service Continuity Management, Capacity Management). Der zusätzliche Prozess „Security Management“ ist ebenfalls enthalten.
- ITIL Practitioner: Die Zertifizierung des Practitioners baut auf der ITIL Foundation auf und kann als das praktische Gegenstück zum eher theoretischen Service Manager gesehen werden. Hier geht es darum, einen oder einzelne Prozesse vonseiten seiner praktischen Umsetzung zu beherrschen. Die Practitioner Ausbildungen werden zumeist zu Praxisbereichen angeboten, z.B. im Bereich Release & Control.

Neben diesen Personenzertifizierungen gibt es Standards, an denen sich die ITIL Publikation zunehmend orientiert. Da ITIL sich ja als Sammlung von Best Practices versteht, kann man ITIL auch so einführen, dass man sich nicht immer voll und ganz an diese Practices hält⁹⁸.

⁹⁶Details dazu siehe [Lie06, S.31].

⁹⁷Übernommen aus: [Lie06, S.32]; teilweise ist hier noch die alte ITILv2 Nomenklatur in Verwendung, die Erklärung ist hier aber maßgeblich.

⁹⁸Man geht u.U. bei der Einführung immer noch nach ITIL vor!

Diese Orientierung bedeutet also: Wenn man sich an die ITIL Vorgaben hält ist es bei entsprechenden Begleitmaßnahmen leicht, eine Standardzertifizierung zu bekommen.

Man beachte: Die hier vorgestellten Normen haben ihren Bezug zur ITIL und IT Governance Materie. Normen wie die ISO 9001 und 9002, die sich mit dem Qualitätsmanagement in einem Unternehmen beschäftigen⁹⁹, können und werden auch weiterhin eine Rolle spielen. Aufgrund des in dieser Arbeit gesteckten Rahmens seien nur diese drei wichtigsten Normen erwähnt.

ISO 20000 - IT Service Management

Die ISO Norm 20000 wird historisch vom britischen Standard BS 15000 begründet und hat dessen wesentliche Inhalte übernommen und aktualisiert. Sie trägt den Titel „IT Service Management System“. Inhaltlich ist die Norm in zwei Teile gegliedert. Der erste Teil befasst sich mit dem Aufbau der IT Organisation und der erforderlichen Qualität der Prozesse. Der zweite Teil ist praxisorientiert und zeigt Best Practices auf, die eine Entsprechung im ITIL Framework haben¹⁰⁰.

Eine Zertifizierung nach ISO 20000 bedingt einerseits die Einhaltung der diversen Best Practices, die ihre Entsprechung in einem ISO Kapitel haben (z.B. hat die Norm in Kapitel 9.2. die Anforderungen an das Change Management formuliert). Andererseits muss die Organisation auch einen gewissen Reifegrad mitbringen. Dieser kann eventuell auch nach dem Reifegrad Modell bestimmt werden (CMM Standard), welches die Reife einer Prozessorganisation in fünf Ebenen zu bestimmen vermag: Initial, Repeatable, Defined, Managed, Optimizing. Ist ein Unternehmen in der Lage, Prozesse initial auszuführen, dann geschehen diese Ad Hoc. Je näher die Optimizing Ebene erreicht wird, desto eher ist das Unternehmen in der Lage, die Prozesse auch weiter zu optimieren bzw. hat es Kontrolle über diese¹⁰¹.

Eine Zertifizierung nach ISO 20000 ist für Unternehmen sinnvoll, die ITIL erfolgreich eingeführt haben. Dies ist gut für die Erfolgssicherung und nicht zuletzt das Marketing.

ISO 27001 - Implementierung von Sicherheitssystemen

Der ISO 27001 Standard trägt den Titel „Information technology - Security techniques - information security management systems - Requirements“ und wurde 2005 als Nachfolger des britischen Standard BS7799 publiziert. Wichtigstes Kernelement ist das in der

⁹⁹Detaillierte Ausführungen sind bei [Pop07, S.42 ff.] zu finden.

¹⁰⁰Vgl. [Pop07, S.43]

¹⁰¹Zu CMM vgl. [Bon05, S.21]

Einführung bereits erwähnte PDCA Modell. Dieses beschreibt die Schritte im Lebenszyklus eines Information Security Management Systems (ISMS). Diese Schritte laufen von einer Planung, über die Umsetzung (Implementierung & Betrieb), die Kontrolle (Überwachen & Überprüfen) bis zu einer Phase des Handelns (Wartung & Optimierung). Geht ein Unternehmen bei der Einführung, dem Betrieb und der Wartung nach diesem Modell vor, so ist dies ISO 27001 konform und kann in diesem Bereich zertifiziert werden¹⁰².

ISO 27002 - Kontrollmechanismen

Der ISO 27002 Standard ist eine Fortführung des ISO 17799 Standards und trägt den Titel „Information technology - Security techniques - Code of practice for information security management“. Im Gegensatz zum bereits vorgestellten ISO 27001 Standard geht es hier nicht um den Lebenszyklus eines Security Systems sondern um die Kontrolle im praktischen Sinne. Die behandelten Themengebiete, denen jeweils ein Kapitel gewidmet wird, sind: Security Policy, Organization of Information Security, Asset Management, Human Resources Security, Physical Security, Communications and Operation Management, Access Control, Information Systems Acquisition & Development & Maintenance, Information Security Incident management, Business Continuity und Compliance. Die von sämtlichen Herstellern vertriebenen Sicherheitslösungen orientieren sich an diesen Standards¹⁰³.

5.2 Generisches Prozessmodell

Alle in ITIL modellierten Prozesse stützen sich auf ein gemeinsames Prozessmodell mit relativ ähnlichen Begrifflichkeiten. Bevor in dieser Arbeit der Prozess des Change Managements näher vorgestellt wird, soll noch ein Blick auf die generelle Gestaltung von Prozessen innerhalb des ITIL Frameworks geworfen werden.

In Abbildung 5.2 wird das generische Prozessmodell von ITIL gezeigt.

Mit gewissen Inputs wird der Prozess versorgt. Abhängig von den Prozessbedingungen werden seine Aktivitäten und Subprozesse ausgeführt, welche vom Prozessinhaber anhand der Prozessziele und verschiedener Parameter überwacht werden können. Das Resultat des Prozesses sind Outputs, z.B. eine erbrachte Leistung. Inputs und Outputs sind spezifiziert, das heißt es ist von vornherein klar, was alles an Informationen in den Prozess genommen werden kann (an verschiedenen Punkten) und was genau am Schluss

¹⁰²Vgl. [Hau07, S.51 ff.]

¹⁰³Zur Nachlese auch zu einzelnen Kapiteln vgl. bei [Hau07, S.52 ff.]

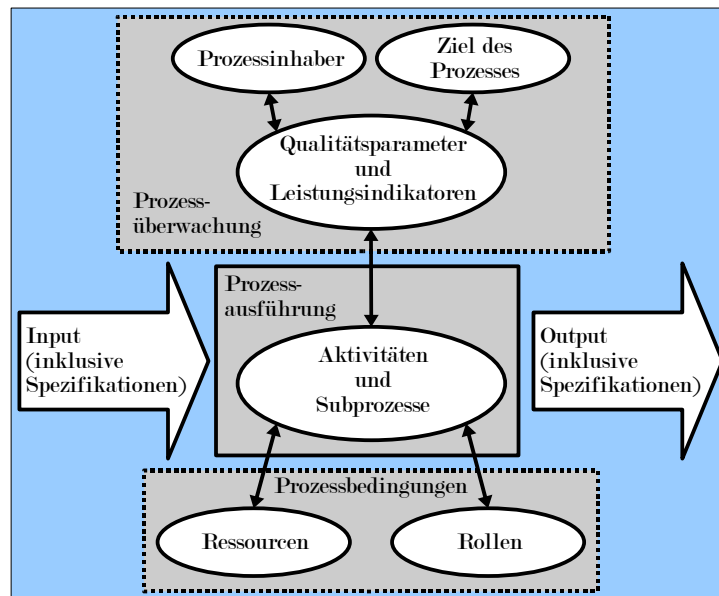


Abbildung 5.2: Generisches Prozessmodell von ITIL, aus [Pop07, S.45]

als Outputs erwartet werden darf. Diese Definition kann sich an bestehenden Normen und Eigenschaften orientieren.

Besondere Bedeutung wird bei ITIL dem Prozessziel beigemessen. Dieses muss klar formuliert sein und definiert für den Prozessinhaber auch klare Parameter und Kriterien, durch welche dieser den Grad der Zielerreichung messen kann. Die Überwachung ist hierbei ein aus der Thematik der IT Governance bekannter Begriff, der das Risiko der Prozesse minimieren soll.

Die Aktivitäten und Sub-Aktivitäten, die während des Ablaufs von Prozessen vonnöten sind, benötigen allesamt Ressourcen, definieren aber auch die verschiedensten Rollen, welche im Unternehmen auf die verschiedenen (IT-) Mitarbeiter zu verteilen sind.

5.3 Grundbegriffe

Im Folgenden sollen einige zentrale, im ITIL Umfeld auftretende Begriffe definiert werden. Diese Begriffe begleiten die weiteren Ausführungen. Die Auswahl orientiert sich an verschiedenen Abschnitten in [Lie06] und [Bon05] und wurde in Anlehnung an das Change und Release Management weiter reduziert.

Change

Ein Change bezeichnet bei ITIL jegliche Änderung an der IT Infrastruktur eines Unternehmens. Das kann die Änderung einer Konfigurationseinstellung, der Austausch von Hardware, aber z.B. auch das Erstellen eines neuen Programms sein.

Change Management System

Ein Change Management System (CMS¹⁰⁴) ist ein EDV-basiertes System, in dem Change Management Statusinformationen und Dokumente während ihres gesamten Lebenszyklus eingepflegt, erweitert und verwaltet werden können.

Konfigurationen

Unter einem CI, einem Configuration Item, zu Deutsch einer Konfiguration, versteht man

[...] die Betriebsmittel und die daraus resultierenden IT Services als Konfigurations-Elemente (Configuration Items - CIs). Jedes Betriebsmittel, dessen Existenz und Version erfasst wird, ist ein CI¹⁰⁵.

Ein CI wird charakterisiert durch¹⁰⁶:

- **Kategorien:** Jede Art von Software, Hardware oder Dokumentation wird gruppiert und so eindeutig zugeordnet, wie zB. Software - ERP Software - Finanzwesen
- **Relationen:** Die wohl wichtigste Ausprägung in der CMDB sind die relationalen Aussagen. Hier wird modelliert, welches CI ein jeweils anderes benutzt, mit ihm verbunden oder Teil eines anderen ist. Auch eine Hierarchie wird über primäre und sekundäre CIs abbildbar gemacht.
- **Attribute:** Jedes CI wird exakt beschrieben über Farben, Formen, Marken Vertrags- und Seriennummern.
- **Status:** Jedes Item kann verschiedene Stati durchlaufen, z.B. geplant, aktiv, in Revision, bestellt, etc.

Alle Hardwarebauteile, die Software und die jeweiligen Einstellungen, die in eine Version mit einem definierten Stand gebracht werden können, werden als CIs registriert.

¹⁰⁴Nicht zu verwechseln mit Content Management Systemen!

¹⁰⁵[Bon05, S.72]

¹⁰⁶Vgl. [Lie06, S.17]

Configuration Management Database

Die Configuration Management Database (CMDB) ist das Verzeichnis aller im Unternehmen existierenden CIs. Das bedeutet, dass die CMDB eine Sicht auf alle Datenbanken mit Betriebsmitteln ist. Entscheidend ist, dass die CIs untereinander verknüpft werden. Das heißt es wird möglich, zu erkennen, auf welche anderen Systembestandteile eine Änderung an einem CI sich auswirken wird.

Ein Beispiel: Wenn der zentrale Backbone Switch neu eingestellt werden muss, so hat ein dadurch ausgelöster Neustart des Switches eine Auswirkung auf alle an dem Backbone operierenden Knoten in einem Netzwerk. Diese Beziehung ist in der CMDB vermerkt. Die CMDB hat ihre Rolle als wichtigstes Instrument zur Abwägung von Risiken und zur korrekten Klassifizierung von Changes (mehr dazu im folgenden Abschnitt)¹⁰⁷.

Request for Change

Der Request for Change (RFC) bezeichnet das zentrale Anforderungsdokument für jegliche Änderung im Rahmen des Change Managements. Das RFC Dokument beinhaltet folgende Informationen¹⁰⁸:

- Die RFC Art (Engl.: Type):
- Eine eindeutige Nummer: Anhand dieser Nummer, der ID, kann man das Dokument eindeutig identifizieren.
- Den Ursprung (Trigger): Der Ursprung gibt an, auf welche Quelle sich der RFC bezieht. Das kann die Nummer einer Problemmeldung, eine Bestellnummer oder die Bezeichnung der rechtlichen Anforderung sein.
- Eine Beschreibung: Im Bereich des RFC Dokumentes ist eine Zusammenfassung der Änderung aufzuführen.
- Objekte, die geändert werden müssen: Eine kurze Zusammenfassung, welche Objekte von der Änderung betroffen sind.
- Änderungsgrund: Eine kurze Darstellung, z.B. der Geschäftsvorfall.
- Umkehrereffekt: Darstellung, was passiert, wenn die Änderung nicht vorgenommen wird.

¹⁰⁷Dazu ergänzend siehe [Bon05, S.72 ff.].

¹⁰⁸Man beachte, dass es sich bei der ITIL Library immer um Empfehlungen handelt. Unternehmen haben immer die Wahl, dies abweichend zu gestalten!

Nicht alle Unterpunkte eines RFC sind vollständig detailliert. Es muss aber ein ausreichender Detaillierungsgrad vorliegen, um die Änderung erfassen und darüber entscheiden zu können.

Post Implementation Review

Der Post Implementation Review, kurz PIR, ist ein Bericht, der nach der Implementierung eines Changes angefertigt wird. Anhand dieses Dokumentes wird entschieden, ob eine Änderung als abgenommen gilt. Auch dazu mehr im nächsten Abschnitt. In einem PIR Dokument können folgende Informationen vorkommen¹⁰⁹:

- Der Zeitpunkt des Auftretens des Problems
- Die Art und Weise, wie das Problem entdeckt wurde
- Die Ursache des Problems, falls diese ermittelt werden konnte
- Vermeidungsstrategien des Problems in Zukunft
- Verbesserungsvorschläge zum Alternativplan
- Eine Dokumentation des Informationsflusses zu den betreffenden Personen
- Die Auswirkungen auf das Tagesgeschäft und wann diese bemerkt wurden
- Die Implikationen des Problems für die Ressourcen- und Changeplanung
- Eine Stellungnahme zur Korrektheit der angenommenen Auswirkungen des Changes
- Die Schlüsse, die aus dem Informationsfluss gezogen werden können (Qualität der Informationswege)
- Eine Reflexion, ob externe Hilfe notwendig gewesen wäre

Release Package

Ein Release Package ist eine Sammlung von Release Units. Eine Release Unit bezeichnet einen Teil eines IT Services oder der IT Infrastruktur, der gemeinsam freigegeben wird. Somit sind in einem Release Package eine oder mehrere Änderungen zusammengefasst, die dann miteinander freigegeben und installiert bzw. umgesetzt werden. Eine Release hat immer eine eindeutige Kennung, z.B. eine Release Nummer¹¹⁰.

¹⁰⁹Vgl. [Lie06, S.92 ff.]

¹¹⁰Vgl. [Fan07, S.86 f.]

5.4 Prozess Change Management

Nicht jede Änderung bedeutet eine Verbesserung, aber jede Verbesserung ist eine Änderung¹¹¹.

5.4.1 Aufgaben und Ziele von Change Management

Der Prozess des Change Management lässt sich wie folgt sehr gut zusammenfassen¹¹²:

Der Prozess, welcher für die Kontrolle des Lifecycle aller Änderungen verantwortlich ist. Das primäre Ziel ist es, nützliche Änderungen unter einer minimalen Beeinträchtigung der IT Services und des Business zuzulassen

Change Management bündelt also die Ressourcen, die für eine Änderung an der IT Infrastruktur notwendig sind, qualifiziert die Risiken und führt die Änderung dann kontrolliert durch. Die kontrollierte Durchführung stellt hierbei sicher, dass die Regeln der IT Governance (bzw. übergeordnet die Regeln der Corporate Governance, vertraglich auch die SLAs¹¹³), nicht verletzt werden.

Zunächst wird versucht, einen Überblick über alle eingehenden Changes zu erlangen. Diese werden priorisiert und dann nach ihrer Art und dem Ausmaß ihrer Auswirkungen auf die IT Infrastruktur einem entsprechenden Genehmigungsverfahren zugeführt. Wenn die Genehmigung erteilt wurde, so wird die Änderung umgesetzt.

Es ist hierbei wichtig, dass sämtliche Prozessschritte (Eingang des RFC, Ausarbeitung, Genehmigung, etc.) genau dokumentiert werden. Diese Informationen sind einerseits wichtig für das Reporting, bei welchem gesehen werden kann, wie gut der Change Management Prozess funktioniert¹¹⁴, andererseits machen genau diese Statusinformationen das Change Management aus. Anfragen, wer gerade einen Change bearbeitet, was damit passiert ist, etc., können genau zurückverfolgt werden und sind (zeitsparend) zentral im Zugriff.

5.4.2 Das Change Management Umfeld

Nachdem bei so ziemlich jeder Aktivität im Tagesgeschäft einer IT Abteilung eine Änderung ausgelöst werden kann - sei es eine Anregung, ein Fehler im System oder eine gesetzliche Anpassung - hat das Change Management Schnittstellen zu sämtlichen

¹¹¹[Bon05, S.91]

¹¹²[Blo08, S.12]

¹¹³Zu SLAs siehe Kapitel 2.3.4.

¹¹⁴Siehe dazu auch die Beiträge zur Überprüfbarkeit von IT Governance in Kapitel 4.2.

ITIL Prozessen. Jede Änderung, die dort gewünscht wird, wird über das Change Management verarbeitet. In Abbildung 5.3 werden diese Zusammenhänge aufgezeigt.

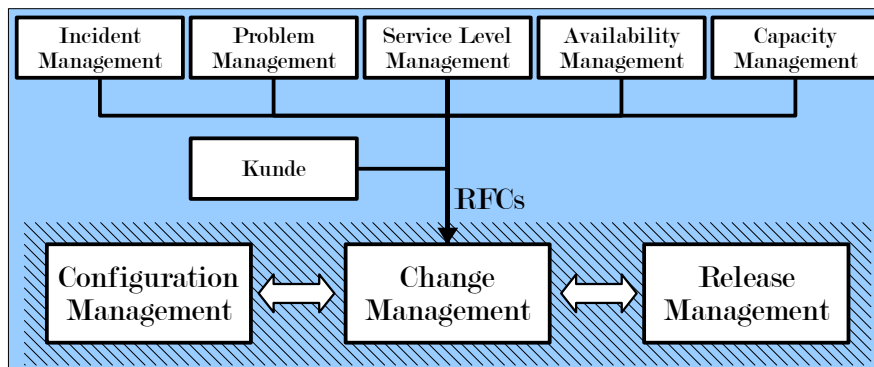


Abbildung 5.3: ITIL Change Management und sein ITIL Prozessumfeld, nach [Bon05, S.94]

Die RFCs (Requests for Change), siehe Abschnitt 5.3 werden in den einzelnen, im folgenden aufgelisteten Prozessen generiert und enthalten standardisierte Informationen. Auch ein Kunde hat die Möglichkeit, einen Change zu beauftragen. Da dies außerhalb der IT passiert werden die Kundenprozesse nicht in ITIL behandelt.

Für ein wirksames Change Management hat dieses aber auch Schnittstellen zu anderen Prozessen, vornehmlich dem Configuration Management und dem Release Management. Auch diese Zusammenhänge werden im Folgenden erläutert.

Kundenwünsche

Ein Servicerequest ist eine Möglichkeit für den Kunden, Änderungen an der IT Landschaft zu beantragen. Die Anforderungen werden direkt¹¹⁵ von einem Service Request ins Change Management übernommen und dort wie eine aus der IT (den anderen ITIL Prozessen) kommende Änderung in einem RFC behandelt.

Incident Management

Die Prozesse des Incident Management befassen sich mit dem Management von Störungen. Wenn beispielsweise ein Teil der Hardware eines EDV Systems nicht korrekt funktioniert, so wird der Austausch als RFC an das Change Management übergeben.

¹¹⁵Dieser Umstand erklärt die hervorgehobene Position des Kunden in der Abbildung 5.3.

Problem Management

Das Problem Management dient als zentrale Anlauf- und Sammelstelle für Probleme. Hier werden alle gemeldeten Probleme gefunden und priorisiert. Um das Problem zu beseitigen wird es sehr häufig notwendig sein, eine Änderung an der IT Infrastruktur vorzunehmen. In dem Fall wird vonseiten des Problem Management ein RFC an das Change Management übermittelt.

Service Level Management

Beim Prozess des Service Level Management geht es darum, die Einhaltung von SLAs mit dem Kunden zu überwachen, nötigenfalls diese mit dem Kunden auch neu zu verhandeln. Bei einer solchen Etablierung eines SLAs - bzw. bei dessen Anpassung - ergeben sich oft weitreichende Änderungen, meist an der Hardwarelandschaft, die wiederum über RFCs im Change Management eingemeldet werden.

Availability Management

Das Availability Management befasst sich mit dem Thema, die Verfügbarkeiten innerhalb der IT Landschaft zu verbessern. Aus den dort angestellten Überlegungen kommen ebenfalls RFCs im Change Management an.

Capacity Management

Über das Capacity Management lässt sich die Kapazität von Systemen über längere Zeiträume planen. Ist beispielsweise eine Erweiterung des Unternehmens geplant, kümmert sich das Capacity Management darum, die Erweiterungen im IT Bereich rechtzeitig über RFCs vorzunehmen.

Configuration Management

Das Configuration Management verwaltet die Konfigurationen der einzelnen Bestandteile einer IT Landschaft in einer Datenbank¹¹⁶. Das Change Management hat einige Schnittstellen mit dem Configuration Management, insbesondere wenn es darum geht, Auswirkungen der geplanten Änderung(en) auf andere Systeme zu beurteilen¹¹⁷.

Die beiden Prozesse Change Management und Configuration Management eignen sich sehr gut für eine gemeinsame Integration, da das Change Mana-

¹¹⁶In der CMDB, siehe Abschnitt 5.3.

¹¹⁷Für ein passendes Beispiel siehe wiederum Kapitel 5.3.

gement für die Autorisierung von Änderungen in der IT Infrastruktur verantwortlich ist und das Configuration Management die Überwachung des Status von Konfigurations-Elementen [...] übernimmt¹¹⁸.

Nach einer über einen RFC durchgeführten Änderung ist natürlich auch der neue Status des oder der geänderten CI(s) an das Configuration Management zurück zu melden. Der Aufbau einer vollständigen CMDB ist sicher einer der zeitraubendsten Prozesse bei einer ITIL Kompletteinführung.

Release Management

Während im Change Management einzelne Änderungen in eher granularer Ausprägung erfasst werden, so ist es Aufgabe des Release Managements, den gesamthaften Blick über die Änderungen in der IT Landschaft zu bewahren. Jede Änderung im Change Management wird, wenn Release Management verwendet wird, einer von drei Release Arten zugeordnet. Die Release Arten:

- **Major Release:** Meist zu einem bestimmten Hauptrelease-Termin werden Major Releases ausgerollt. Hier wird die IT Landschaft meist in großen Schritten auf einen neuen Stand gehoben. Das bedeutet, dass hier Erweiterungen und auch Fehlerbehebungen durchgeführt werden.
- **Minor Release:** Kleinere Änderungen, die die Funktionalität nicht wesentlich erweitern, werden in Minor Releases zusammengefasst.
- **Emergency Fix:** Änderungen, die nicht bis zum nächsten Releasetermin warten können, weil diese dringend umgesetzt werden müssen, werden in eine Notfall-Release zusammengepackt.

Es ist außerdem in der Verantwortung des Release Managements, die Änderungen, die im Change Management bearbeitet wurden, entsprechend zu testen, freizugeben und sich um alle Begleitmaßnahmen zu kümmern (Einschulung, Kommunikation, Gesamtdokumentation).

Zusammenfassung

Change Management steht mit sehr vielen Prozessen der ITIL Library in Verbindung, meist zur Bearbeitung von RFCs, mit der CMDB und der Releaseplanung ist aber auch ein hohes Potential zur Integration mit Configuration und Release Management gegeben.

¹¹⁸[Bon05, S.95]

5.4.3 Fahrplan für die Einführung von Change Management

Aufgrund der zuvor skizzierten Abhängigkeit von Change Management und anderen Prozessen stellt sich die Frage nach einem optimalen Vorgehensmodell für die Einführung. Es ergibt nämlich keinen Sinn, lediglich Change Management einzuführen und die anderen ITIL Prozesse zu ignorieren. Man hat dadurch wohl ein ITIL konformes Change Management aufgebaut, aber der Mehrwert geht verloren, weil die anderen Prozesse sich nicht an den Change Prozess anpassen.

Aus den Erfahrungen in [Lie06] wurde ein „ITIL Fahrplan“ für eine Einführung von ITIL von den Anfängen bis zum Change- und Releasemanagement skizziert, welcher versucht, den soeben vorgestellten Abhängigkeiten Rechnung zu tragen. Ein derartiges, stufenweises Vorgehen erleichtert die Einführung von ITIL und stellt sicher, dass einzelne Prozesse nicht offen im Raum stehen, sondern aufeinander aufbauend gestaltet sind¹¹⁹.

Der Fahrplan wird in Abbildung 5.4 dargestellt.

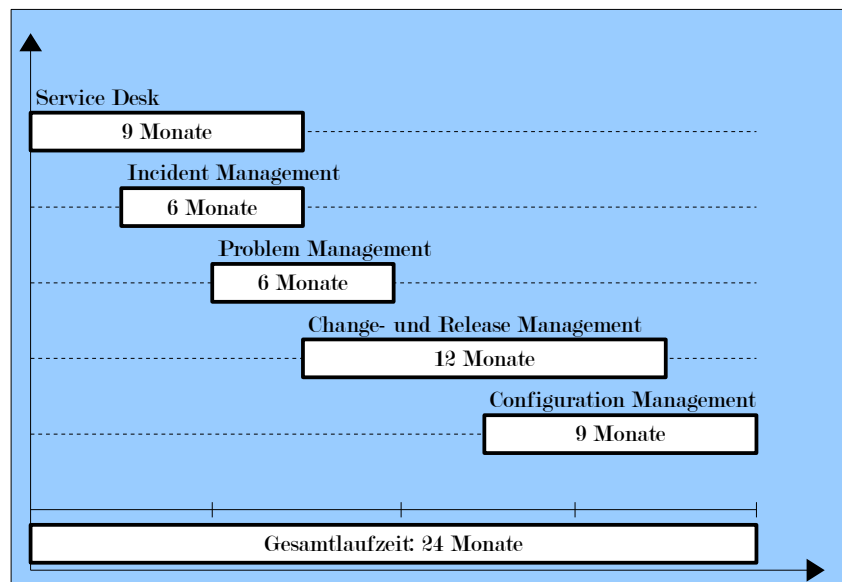


Abbildung 5.4: ITIL Fahrplan, aus: [Lie06, S.48]

Service Desk

Zunächst empfiehlt Lienemann, den Service Desk einzuführen. Dieser ITIL Terminus beschreibt das Konzept, für alle Anliegen, die der Kunde an die IT stellt, eine zentra-

¹¹⁹Es sei aber an dieser Stelle auch gesagt, dass kaum ein Unternehmen hier „bei Null“ beginnt, sondern dass eine ITIL Einführung oft nur bedeutet, einen gelebten Prozess auszudefinieren und ihn strikter zu leben als vorher.

le Service- und Supportstelle einzurichten, die die eingehenden Anfragen, Beschwerden oder Probleme aufnimmt und dann entsprechend weiterleitet. So wird eine direkte Kommunikation zwischen den IT Spezialisten unterbunden. Das hilft den Spezialisten, ihrer Tätigkeit nachzugehen und sich nicht um Probleme kümmern zu müssen, die nicht ihre eigentliche Aufgabe oder bereits bekannt sind.

Die Einführung des Service Desk schafft auch Ordnung in der IT Organisation, weil dafür die Zuständigkeiten der einzelnen Mitarbeiter festgeschrieben und der genaue Prozess für die Bearbeitung von Kundenanfragen definiert wird.

Für den Kunden bedeutet die Einführung einer zentralen Anlaufstelle, eines Single Point of Contact (SPOC), eine höhere Benutzerfreundlichkeit. Die Mitarbeiter des Service Desk sind nämlich idealerweise keine IT Cracks, also reines Fachpersonal, sondern verfügen auch über gute Kommunikationsfertigkeiten¹²⁰.

Incident Management

Das Incident Management ist der Prozess, der von der IT Abteilung mit Service Desk gelebt wird. Hier wird der Support in mehrere Level unterteilt, wobei der Service Desk bereits den ersten Supportlevel übernimmt. Kann dieser den „Incident“, also die Meldung des Kunden, nicht lösen, so wandert der Incident zu höheren Support Levels in Richtung der zuständigen Spezialisten. Dort wird versucht, den Incident abzuarbeiten. Es empfiehlt sich, das Incident Management kurz nach dem Anlaufen des Support Desks einzuführen, da dieser die logische Fortsetzung der Etablierung einer zentralen Anlaufstelle darstellt.

Problem Management

Wenn die IT Abteilung gelernt hat, mit Incidents korrekt umzugehen, diese auch bei den richtigen Personen landen und der Kunde den Service Desk auch angenommen hat, wird mit dem Problem Management der nächste ITIL Prozess in der IT Organisation definiert. Dies hat seine Begründung wiederum in der Definition eines Incidents und eines Problems. Tritt ein Incident (wohl am besten mit einer „Störung“ übersetzt) nämlich häufig auf, so wird dieser dem Problem Management übergeben, welches versucht, die Ursache der Störung zu finden. Man kann also sagen, dass Service Desk, Incident Management und Problem Management derartig eng miteinander verknüpft sind, dass diese nur gemeinsam eingeführt werden können¹²¹.

¹²⁰Vgl. [Lie06, S.46].

¹²¹Vgl. wiederum [Lie06, S.46].

Change- und Release Management

Erst nach der Einführung einer geordneten „Vorsortierung“ von Kundenanfragen, nach einer Reduzierung auf Probleme sowie einer Erforschung der entsprechenden Ursachen macht es Sinn, mit Änderungsdokumenten zu arbeiten. So lässt sich wohl am besten die Abfolge begründen. Ein Change Management ohne die vorherige Einführung bzw. Adaptierung der Anfrage- und Problembearbeitung in der IT macht sehr wenig Sinn.

An Lienemanns Zeitplan fällt aber auch auf, dass Problem Management noch nicht voll eingeführt sein muss, damit der Change Management Prozess bereits begonnen werden kann. Das rührt daher, dass aus dem Incident Management bereits erste Änderungen per RFC angefordert werden und dass die ersten Anwendungen des Problem Management Prozesses bereits gute Testszenarios für Change Management darstellen¹²².

Die Dauer der Einführung ist mit 12 Monaten eher lange bemessen. Dies hat sich in der Praxis dadurch erforderlich gezeigt, weil die IT Mitarbeiter ihre Arbeitsweise ändern müssen. Es ist nun bei jeder Änderung der entsprechende ITIL-orientierte Prozess zu leben, es müssen Rollen geschaffen und auch verankert werden. Um so etwas intern zu „verkaufen“ bedarf es Zeit und Gestaltungsmöglichkeiten durch die Mitarbeiter.

Configuration Management

Wie bereits in Abschnitt 5.3 erwähnt sind bei Änderungen die Abhängigkeiten zwischen den einzelnen Komponenten einer IT Landschaft von entscheidender Bedeutung. Ein Change Management sollte also auch immer die Schnittstelle zum Configuration Management implementieren. Die Einführung beginnt während der Umsetzung von Change Management und ermöglicht so, bereits in einem frühen Stadium erste Abhängigkeiten zu sehen. Es werden die über die Changes durchgeführten Änderungen dann bereits an den CIs im Configuration Management vorgenommen und über den PIR¹²³ existiert auch ein Werkzeug, vom Change Management ins Configuration Management zurück zu berichten. Die letzte Phase, in der noch etwa 3 Monate nur am Configuration Management implementiert wird, ist häufig mit dem Optimieren von Performance, einer genauen Evaluation sowie Nachjustierung der gewählten CI Gruppierungen und auch mit dem Aufbau des kompletten CMDB Datenbestands verbunden.

¹²²Siehe weiterführend bei [Lie06, S.47 und S.68 ff.].

¹²³Siehe Kapitel 5.3.

5.4.4 Beteiligte Rollen

Bevor nun der Prozess des Change Managements an sich beleuchtet wird noch ein paar Anmerkungen zu den benötigten Rollen im Unternehmen. Wie bereits an anderer Stelle erwähnt inkludiert eine Ausrichtung an ITIL Vorgehensmodellen auch eine Neuausrichtung der IT Organisation. Dies führt zu neuen Rollen und Verantwortlichkeiten. In den folgenden Zeilen werden die wichtigsten Rollen - im Unternehmen wie im Change Management Prozess - vorgestellt. Diese werden in der Detailbeschreibung von Change Management in dem anschließenden Kapitel [Lie06] benötigt.

Change Advisory Board

Das Change Advisory Board, kurz CAB, übersetzt man wohl am besten mit dem Wort Änderungsbeirat. Der Änderungsbeirat ist ein nicht ständig tagendes Gremium, welches schwerwiegende Changes genehmigen muss (siehe nächster Abschnitt). Es wird vom Change Manager einberufen.

Das CAB setzt sich in der Regel aus einem Personenkreis wie folgt zusammen¹²⁴:

- Change Manager
- Service-(Level)-Manager¹²⁵
- Vertreter aus dem Incident-, Problem- und Releasemanagement
- Vertreter der Anwendungsentwicklung
- IT Spezialisten
- Bereichsmanager und Manager der Finanzabteilung
- Vertreter der Kundenumgebung
- Vertreter der Dienstleister

Der Sinn des Change Advisory Boards ist es, bei Änderungen, welche die IT gesamthaft betreffen, ein vordefiniertes Gremium zur Verfügung zu haben. Diese Gruppe kann ausreichend die verschiedenen Aspekte der IT Landschaft berücksichtigen, um über Änderungen gemeinsam zu entscheiden.

Global agierende oder sehr große Unternehmen können mehrere CABs besitzen, die z.B. nach Regionen zuständig sind.

¹²⁴[Bon05, S.92]

¹²⁵Dieser ist zuständig für das Service Level Management.

Change Manager

Der Change Manager repräsentiert den Prozessverantwortlichen für das Change Management im Unternehmen. Er steuert die Strategie des Change Management Prozesses, ist aber nicht notwendigerweise für alle Prozessaktivitäten verantwortlich¹²⁶.

Er hat die Letztentscheidung bei der Genehmigung von kleineren Änderungswünschen und bereitet größere Änderungen für das Change Advisory Board auf.

Emergency Change Advisory Board

Zu einem CAB Meeting sind besonders viele Teilnehmer geladen. Bei dringenden Änderungen, die schwerwiegend sind, ist es nicht praktikabel, das komplette CAB zusammentreten zu lassen. Diese Aufgabe fällt dem Emergency Change Advisory Board, kurz ECAB, zu. Es ist ein ähnlicher, wenngleich etwas reduzierter Personenkreis wie das CAB, und wird im Notfall vom Change Manager einberufen.

Change Authority

Die Change Authority (CA) ist eine Rolle, die von einer Person oder meist von einem Team wahrgenommen wird. Das CA Team ist für die Freigabe einer jeden Änderung, die im Unternehmen mittels Change Management realisiert wird, zuständig. In ihrer Entscheidung stützen sich die Mitarbeiter des CA Teams auf Vorschriften, Meinungen oder ihre Erfahrung. Sie sind der Minimierung des Geschäftsrisikos und der maßvollen finanziellen Gebarung verpflichtet. Das gesamte Team untersteht dem Change Manager und nimmt bei der Umsetzung der IT Governance (Stichwort: Risikominimierung) einen bedeutenden Platz ein¹²⁷.

IT Executive Comitee

Das IT Executive Comitee (ITEC) ist eine Abkürzung für das höhere Management der IT Organisation¹²⁸. Das ITEC kann durchaus das gleiche Gremium sein, welches auch für die Erstellung der IT Governance Vorschriften verantwortlich zeigt.

¹²⁶Dies wird ausführlichst bei [Lie06, S.78] beschrieben.

¹²⁷Siehe weiterführend bei [Blo08, S.89].

¹²⁸Der Begriff wird bei [Blo08, S.18] geprägt.

5.4.5 Der Prozess im Detail

Um den Prozess des Change Management im Detail zu beschreiben verwendet das OGC das generische Prozessmodell, welches in Kapitel 5.2 bereits beschrieben wurde¹²⁹.

Input

Als Input in den Change Prozess dienen zunächst die verschiedenen **Änderungs- und Releasevorschriften** die in den Unternehmen unterschiedlich gehandhabt werden können. Ein gängiges Vorgehen ist es, Änderungen nur zu gewissen Wartungsfenstern zuzulassen und immer zum Halbjahr eine Release vorzusehen. Dies kann aber natürlich pro Unternehmen variieren. Die **Planungen für die Changes und die Planned Service Outages (PSOs)**, zu denen die Systeme nicht verfügbar sind, sind somit ebenfalls Bestandteil der Inputs. Ein weiterer Input ist der **Request for Change**, der RFC, welcher die Grundlage der Änderungen bildet (siehe Kapitel 5.3). Das **Change Proposal** ist der Vorschlag, wie eine Änderung durchgeführt wird. Er ist die Grundlage der nachfolgenden Umsetzung. Planungen sind auch an einigen Stellen des Change Prozesses als Input gefragt. An dieser Stelle sind die **Änderungs-, Wechsel-, Release-, Einsatz-, Test-, Evaluierungs- und Nachbesserungsplanung** als Input möglich. Aus dem Release Management kommen **CI's und/oder Release Pakete** als Input in den Prozess. **Definitiv geplante Konfigurationen**, die im Configuration Management bereits autorisiert wurden sowie **Testergebnisse, -berichte und Evaluationsberichte** schließen die Inputs ab.

Output

Aus dem Prozess werden als Output die **genehmigten und abgelehnten RFCs** generiert. Weiters sind als Output alle **Änderungen an Infrastruktur und den Services**, die von genehmigten RFCs herrühren, zu betrachten. **Anlagen und CI's können neu, geändert oder entfernt werden**, während der Change implementiert wird. Sie stellen daher auch einen Output dar. Ein **Änderungszeitplan für die Durchführung der Änderungen** sowie **überarbeitete PSOs** gehören auch zu den Outputs wie auch allfällig **veränderte Change Planungen**. Schließlich gehören auch noch **Änderungsentscheidungen, die gesetzten Aktionen, Dokumente, Aufzeichnungen und Berichte** zu den Outputs des Change Management Prozesses.

¹²⁹Zu den Ausführungen dieses Abschnittes empfiehlt sich als Detaillektüre der englische Originaltext bei [Fan07, S.42 ff.].

Informationen

All jene Informationen, die für eine erfolgreiche Bearbeitung eines Changes notwendig sind, werden im englischen mit den sieben „R“s des Change Management zusammengefasst¹³⁰:

- Who **raised** the change?: Wer hat die Änderung beantragt?
- What is the **Reason** for the Change?: Was ist der Grund für die Änderung?
- What are the **risks** involved in the change?: Welche Risiken bestehen bei dem Change?
- What **resources** are required to deliver the changes?: Welche Ressourcen werden für die Änderung benötigt?
- Who is **responsible** for the build, test and implementation of the change?: Wer ist für den Zusammenbau, den Test und die Implementierung zuständig?
- What is the **relationship** between this change and other changes?: Wie hängt dieser Change mit anderen Changes zusammen? Anmerkung: Diese Abhängigkeiten ergeben sich oft erst über die CMDB, welche gegen die Abhängigkeiten involvierter CIs zu prüfen imstande ist.

Je genauer diese Informationen bereits zu Beginn vorliegen, umso einfacher wird es sein, den Change schnell umgesetzt zu bekommen.

Prozessablauf eines „Normal Change“

Beim „Normal Change“ handelt es sich, wie der Name schon sagt, um eine normale Änderung. Das heißt alle mit Ausnahme von sehr dringenden Änderungen durchlaufen diesen Prozess. Der zugehörige Prozessablauf ist in Abbildung 5.5 dargestellt. Es beginnt damit, dass der Anforderer einen RFC mit optionalen Umsetzungsvorschlägen (Change Proposal) erstellt. Je nachdem ob das Unternehmen einen Service Desk einsetzt, geschieht die nun folgende Erfassung und Prüfung dort oder in einem individuellen Verfahren¹³¹. Bei der Erfassung wird vom Service Desk eine eindeutige Kennung vergeben¹³². Die Prüfung beinhaltet nun einerseits eine formale Prüfung, andererseits

¹³⁰Vgl. bei [Blo08, S.30].

¹³¹Aufbauend auf die Erläuterungen in Kapitel 5.4.3 geht diese Arbeit in der Folge davon aus, dass ein Service Desk in Verwendung ist.

¹³²Siehe sehr anschaulich bei [Lie06, S.74].

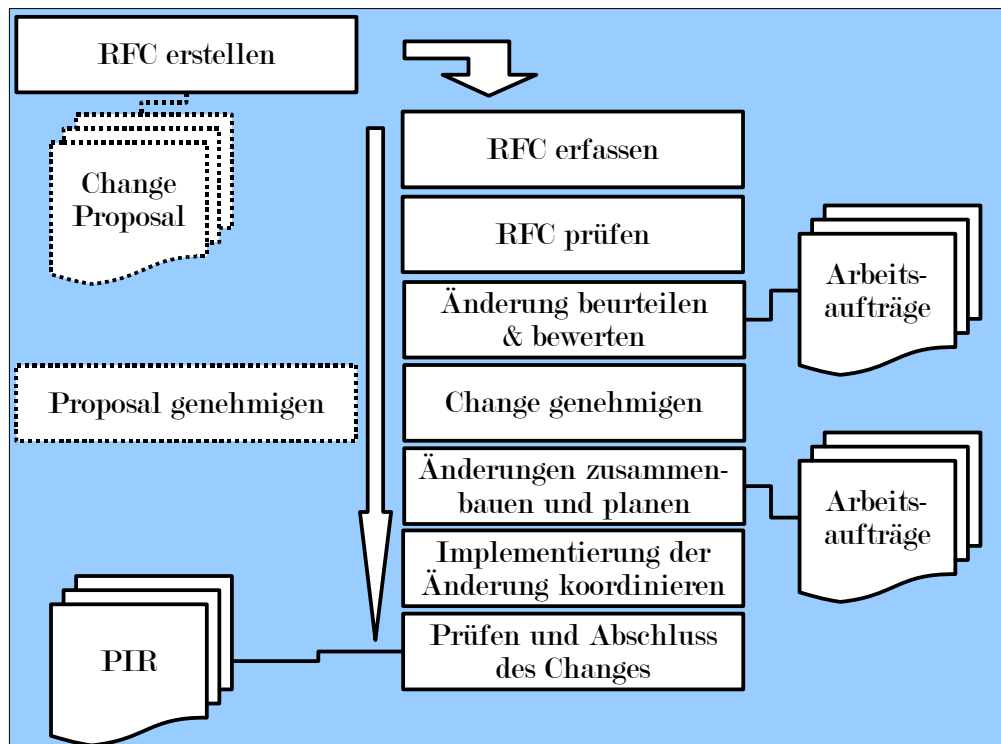


Abbildung 5.5: Prozessablauf eines Normal Change, vereinfacht und übersetzt aus [Fan07, S.49]

wird hier bereits eine Dringlichkeitsstufe ermittelt. Die hier verwendeten Stufen werden vonseiten ITIL lediglich als Beispiele aufgeführt, das heißt, dass ein Unternehmen hier seine eigenen Dringlichkeitsstufen verwenden kann. In der Praxis bewähren sich vier Stufen, nämlich weniger dringend, normale Dringlichkeit und hohe Dringlichkeit. Als vierte Stufe ist der Emergency Change anzuführen, welcher eine sofortige Ausführung erfordert¹³³.

Ausgestattet mit dieser Grundkategorisierung läuft der RFC nun in das Change Management Team. Dort wird die Änderung beurteilt und bewertet. Dies bedeutet noch nicht, dass über die Ausführung oder die Ablehnung des RFC direkt entschieden wird. In diesem Schritt werden Unterlagen und Informationen beschafft, die für eine Entscheidung notwendig sind. Über allfällige Arbeitsaufträge werden Personen damit beauftragt, diese Informationen zu erstellen. Das kann z.B. ein Auftrag für die Erstellung eines Proof of Concept (POC, zu Deutsch: Beweis des Konzepts) sein, um zu sehen, dass ein solcher Change überhaupt umsetzbar wäre. An dieser Stelle wird auch über die CMDB ge-

¹³³Sollte es sich um einen Emergency Change handeln, wird der zugehörige Prozess, erläutert im nächsten Textblock weiter unten, ausgeführt.

prüft, welche Abhängigkeiten bestehen. Im Teilschritt der Bewertung urteilt der Change Manager darüber, wer den Change genehmigen muss. Hier wird nach einem im Unternehmen zu definierenden Autorisierungskonzept vorgegangen. Ein gängiges Konzept wird in Abbildung 5.6 gezeigt. Je nach betroffenen Bereichen, Services und nach einer Kosten/Risiko Schätzung werden die Changes in verschiedene Level eingeteilt, die entweder lokal vom Change Manager bzw. seinem Team, vom CAB oder ECAB¹³⁴, vom ITEC oder, im Falle besonders hoher Kosten oder eines besonders hohen Risikos, direkt von der Geschäftsleitung zu genehmigen sind.

Auf die Bewertung folgt im nächsten Schritt die Genehmigung des RFC. Allfällige Vor-

Level 1	Geschäftsleitung	Hohe Kosten Hohes Risiko
Level 2	ITEC	Änderung betrifft viele Services oder Bereiche
Level 3	CAB (ECAB)	Änderung betrifft einen Service oder Bereich
Level 4	Change Management Gruppe	Alltägliche Änderung

Abbildung 5.6: Change Authorization Model nach [Fan07, S.57]

schläge, die mit dem RFC eingebracht wurden, werden ebenfalls genehmigt. Sollte die Genehmigung von dem zuvor ermittelten Gremium nicht erteilt werden, so geht der RFC mit Begründung zurück an den Initiator.

Wenn die Genehmigung erteilt wurde, ist das Change Management nun dafür zuständig, die Änderung zu organisieren. Hier setzt die Schnittstelle zum Release Management an. Das Release Management würde sich hier um die Erstellung, das Testen und auch das ausrollen des Changes zu einem bestimmten, passenden Releasetermin kümmern. Die entsprechenden Arbeitsaufträge würden auch innerhalb des Release Managements erstellt.

Unter dem nächsten Schritt, der Koordinierung, versteht man, dass Change Management oder CAB noch genau über die Implementierung befinden können. Das heißt, dass sie den genauen Zeitplan für die Produktivsetzung noch beeinflussen können.

Mit dem Schritt „Prüfung und Abschluss“ wird der Change geschlossen. Idealerweise

¹³⁴Das ECAB ist zuständig für Emergency Changes und wird weiter unten im Zuge des Emergency Change Prozesses erläutert.

wird in einem PIR noch ein Abschlussbericht erstellt (siehe dazu unter 5.3).

Prozessablauf eines „Emergency Change“

Liegt ein Notfall vor, der eine Änderung erfordert, so wird ein verkürztes Verfahren angewendet, welches in Abbildung 5.7 dargestellt wird.

Zunächst wird festgestellt, dass ein Notfall eine dringende Änderung erfordert. In einer

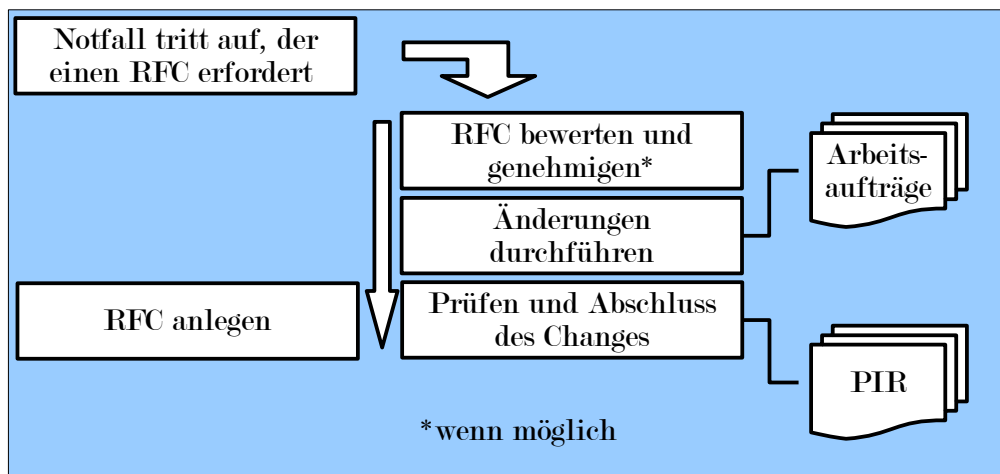


Abbildung 5.7: Emergency Change in Anlehnung an [Lie06, S.78]

ersten Evaluierung wird entschieden, wer diesen Emergency Change genehmigen darf. Das kann der Change Manager oder - bei schweren Auswirkungen - das ECAB sein. Wurde eine Genehmigung erteilt oder ist diese in der kurzen Zeit nicht möglich, so wird die erforderliche Änderung durchgeführt oder mittels Arbeitsaufträgen zur sofortigen Durchführung weitergeleitet.

Erst nach der Änderung wird ein RFC angelegt und gleichzeitig auch ein PIR erstellt. Bei dringenden Changes ist das PIR Dokument wichtiger als bei normalen Changes, weil diese Dokumente bei entsprechender Analyse dabei helfen können, die Anzahl der dringenden Changes zu verringern.

Zusammenfassung

In vielen Unternehmen existierten auch vor ITIL Einführung Systeme, wie Änderungen zu genehmigen sind. Die Vorteile der ITIL Vorgehensweise mit Change Management und CMBD liegen im Bereich der Risikominimierung, weil alle Änderungen, sowohl an Hard-, Software und auch an Inventar, berücksichtigt werden können. Die Personen, welche die Genehmigungen erteilen können, sind vorab definiert und es muss nicht bei jeder

Änderung zuerst geklärt werden, wer für die Freigabe zuständig ist.

Klare Strukturen in diesem Bereich haben natürlich positive Auswirkungen auf das Gesamtrisiko im Unternehmen und stehen somit in klarem Zusammenhang mit der Corporate und IT-Governance.

5.4.6 Prozessindikatoren

Wie im Kapitel 4.2 erläutert, müssen Prozesse für eine wirksame IT Governance Strategie auch auf ihre korrekte Funktion überprüft werden können. Hier stellt das OGC in der Originalpublikation¹³⁵ sehr viele Key Performance Indicators (KPIs), also kritische Prozessindikatoren vor. Blokdijk fasst in [Blo08, S.44 und 97] entsprechend zusammen: Ein KPI sollte immer dem SMART Prinzip folgen. Dies bedeutet auf Englisch, dass ein SMART-Ziel **S**imple, **M**easurable, **A**chievable, **R**ealistic und **T**ime-Driven ist.

Auf Deutsch würde man folgende Kriterien anwenden:

- **Einfachheit:** Ist ein Ziel nicht einfach formuliert, so kann es nicht sehr schnell erfasst werden. Je einfacher ein Ziel nachzuvollziehen ist, desto besser ist es geeignet.
- **Messbarkeit:** Wenn ein Ziel nicht messbar ist lassen sich nur ungenaue Quantitative Aussagen treffen. Da in einem Unternehmen mehr als eine Handvoll Change Requests anfallen ist vor allem die Mengenbetrachtung interessant.
- **Erreichbarkeit:** Sind Ziele so definiert, dass sie nicht erreichbar sind, so erzeugt dies mitunter Unwillen und Frustration bei den Angestellten.
- **Realistische Zielsetzung:** Ist eine Zielsetzung für diejenigen, denen sie gesetzt wird, unrealistisch, dann wird ihre Einhaltung nicht zwangsweise beachtet werden. Das gewonnene Datenmaterial wird unbrauchbar.
- **Zeitgetriebene Qualifizierung:** Die Evaluation eines Zieles sollte immer in einer sinnvollen Zeitachse erfolgen. In Unternehmen empfiehlt sich beispielsweise die Zeit zwischen den verschiedenen Releases als gute Zeitachse. Zum Beispiel: Wie viele Change Requests gab es zwischen Release 01/2008 und Release 2/2008.

Blokdijk schlägt zwei Gruppen von Reports vor, die eine für das IT interne Qualitätsmanagement, die andere für das Management. Alle diese KPIs müssen von den jeweiligen Organisationen mit Zielwerten ausgestattet werden, die zur entsprechenden IT Governance Strategie passen.

¹³⁵[Fan07, S.64 ff.]

KPIs für IT-interne Qualitätssicherung

Blokdijk empfiehlt, folgende KPIs intern pro gewählter Zeitachse zu erheben¹³⁶:

- Welche CIs im Configuration Management werden fortlaufend geändert?
- Anzahl der neu erstellten RFCs
- Anzahl der erfolgreichen Änderungen
- Anzahl der abgelehnten RFCs
- Anzahl der zurückgenommenen Änderungen, eventuell mit der Begründung
- Verteilung des Ursprungs der RFCs (aus welchem ITIL Prozess)
- Korrelation der aufgetretenen Incidents mit den erfolgten Changes
- Anzahl erledigter Nachbearbeitung von Änderungen
- Menge an Changes, die einen Hinweis auf die betroffene CI enthielten

Diese KPIs beobachten die Qualität, die Menge und die Art der Behandlung der RFCs. Abweichungen zu prüfen ist bei den internen Reports die spezifische Aufgabe der IT Abteilung; idealerweise ist dies auch eine direkte Möglichkeit des Feedbacks für den Change Manager.

KPIs für Managementberichte

Die folgenden Fragestellungen helfen dem Management laut [Blo08, S.101 f.] dabei, den Erfolg und die Strategie des Change Managements zu verifizieren um fallweise Korrekturen vorzunehmen:

- Anzahl der behandelten RFCs, Ablehnungsquote und Quote der Emergency Changes
- Anzahl der noch zu verarbeitenden Changes, also die Change-Warteschlange
- Detailinformationen, was noch alles an Arbeit offen ist und wo es zeitlich eng wird
- Anzahl der Änderungswünsche pro Geschäftsbereich
- Personalauslastung, das heißt wie lange haben Mitarbeiter an verschiedenen Changes mitgewirkt (falls erfasst)

¹³⁶Aus dem Englischen übernommen von [Blo08, S.100].

- Vollständigkeitsreporting für RFCs und für Beschlussprotokolle des CAB
- Eine gemeinsam mit den Finanzbereichen erstellte Betrachtung der Kosten pro Change

Diese KPIs zielen eher auf die Managementaufgaben der Steuerung des IT Bereiches ab. Hier wird transparent, wie gut sich alles im Plan befindet, ob es budgetär in Ordnung ist und welcher der Bereiche (Kunden) mehr Zeit braucht als der andere.

Mit dem Abschluss dieses Kapitels wurden nun die Grundlagen von ERP Systemen, die speziellen Implikationen von ERP Systemen auf den Software Lifecycle, die Einflüsse von Regulativen auf die IT (in Form von Governance Strategien) und Vorgehensmodelle erläutert. Bei den Vorgehensmodellen wurden die derzeit gängigen Modelle diskutiert, in Form von ITIL detailliert beschrieben und mit Indikatoren versehen. Dieses Prozessmodell wird in weiterer Folge in der Arbeit noch große Bedeutung gewinnen.

Es folgt Kapitel 6, welches den Ablauf von Änderungsvorgängen im SAP System, als ein konkretes Beispiel eines ERP Systems, erklärt. Die bewährten, von SAP angebotenen Hilfsmittel werden evaluiert, ihre neuen Pendant vorgestellt und ein Vergleich zwischen beiden Lösungen durchgeführt.

6 Change Management im SAP Umfeld

6.1 Systemarchitektur

In Kapitel 2.2 wurden bereits die wichtigsten Merkmale der Architektur von ERP Systemen erläutert. In diesem Abschnitt soll nun speziell auf die Systemarchitektur von SAP eingegangen werden, welche für den nachfolgenden Abschnitt der verfügbaren Transportkonzepte wichtig ist.

6.1.1 Programmiersprachen

Die SAP Welt kennt derzeit (Stand Ende 2008) im Wesentlichen zwei verschiedene Programmiersprachen, ABAP und Java¹³⁷. Auch wenn in der SAP Welt hauptsächlich ABAP Verwendung findet, wird im Folgenden auf beide Programmiersprachen eingegangen. Die öffentliche Positionierung von ABAP vs. Java vonseiten der SAP ist die, dass ABAP für ERP Anwendungen und die Datenaufbereitung verwendet werden kann. Java hingegen wird eher Bedeutung bei der Visualisierung, den Programmierer-Communities sowie den Interfaces zu anderen Plattformen zugeschrieben. Offiziell heißt es dazu freilich meistens, ABAP und Java können gleichwertig verwendet werden.

ABAP

ABAP, die Advanced Business Application Programming Language, ist eine von SAP entwickelte, an COBOL orientierte Programmiersprache, die für die Verarbeitung von Massendaten optimiert wurde. Die Syntax ist über den OpenSQL Dialekt in der Lage, direkt aus ABAP Anweisungen heraus Datenbankabfragen zu generieren.

Auch wenn COBOL und ABAP schon zu den verhältnismäßig „alten“ Programmiersprachen zählen, so ist die Performance im Datenbankbereich noch immer einer der großen Pluspunkte, welche für ABAP sprechen.

ABAP ist eine plattformunabhängige¹³⁸, interpretierte Sprache. Mit ABAP Objects existiert

¹³⁷Auch wenn einzelne Komponenten auf Programmiersprachen wie Microsoft Visual Basic oder PHP setzen, so ist dies nur für einen kleinen Bruchteil der SAP Anwender von Belang.

¹³⁸Von SAP werden hierfür die meisten gängigen Betriebssysteme unterstützt.

tiert auch eine Möglichkeit, ABAP objektorientiert zu programmieren, wenngleich der Kern der Programme immer noch linear umgesetzt wird.

Java

Java ist eine in den späten 1990er Jahren entwickelte Programmiersprache, die vollständig objektorientiert, plattformunabhängig und interpretiert konzipiert ist. Java hat seinen Erfolg vor allem der frühen Ausrichtung auf das Internet (z.B. über die Browser Applet Technologie) zu verdanken. Durch die vollständige Objektorientierung können Java Programme, wenn diese größere Bibliotheken verwalten und darauf zugreifen müssen, eher schwerfällig laden, um dann schnell ausgeführt zu werden.

Exkurs: Interpretierte Programmiersprachen

Beide von der SAP verwendete Programmiersprachen werden interpretiert. Die Bedeutung des Begriffes der interpretierten Programmiersprache wird anhand von Abbildung 6.1 veranschaulicht.

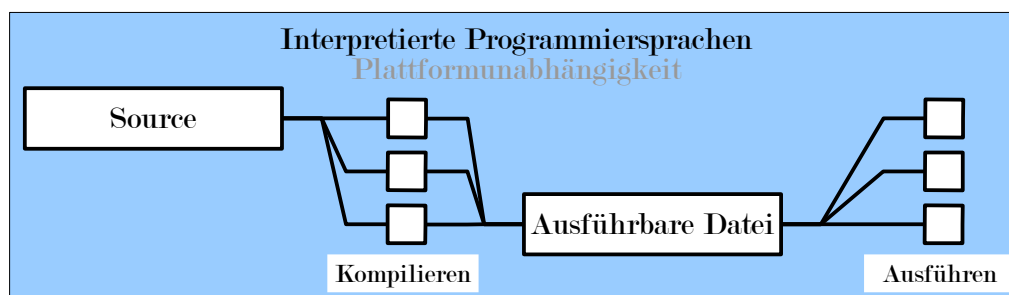


Abbildung 6.1: Schema einer interpretierten Programmiersprache

Die Grafik zeigt im linken Bereich den Sourcecode, der zur Ausführung gelangen soll. Dieser wird durch kompilieren von der ursprünglichen Source in einen interpretierbaren Zwischenzustand umgesetzt. Dieser Vorgang kann auf allen angebotenen Plattformen (in der Grafik drei verschiedene) ablaufen. Der daraus resultierende, ausführbare Code ist wiederum plattformunabhängig. Das bedeutet, dass dieser Code von den jeweiligen Ausführungsprogrammen (Runtimes) auf allen unterstützten Plattformen ausgeführt werden kann.

Ein Beispiel: Ein Java Programm wird unter Windows geschrieben und dann als .java Datei im Textformat gespeichert. Dort wird dann mittels des Java Compilers (javac) eine ausführbare Datei erzeugt (.class Datei), die dann z.B. auch auf Unix mit der entsprechenden Java Runtime ausgeführt werden kann.

Dieses Prinzips bedienen sich - wie bereits erwähnt - sowohl ABAP als auch die Programmiersprache Java.

6.1.2 Systemkomponenten

Durch die Verwendung der beiden Programmiersprachen ABAP und Java bietet die SAP auch zwei verschiedene Applikationsserver-Architekturen an. Beide verfolgen einen ähnlichen Aufbau wie generell in Kapitel 2.2.4 beschrieben¹³⁹.

ABAP

Der ABAP Teil besteht aus einer Datenbank, die über einen Applikationsserver zugegriffen wird. Der Endbenutzer kann sich mittels SAPGui, einer Anzeigesoftware für die SAP Programme, sogenannte DynPros (dynamische Programme) an den Applikationsserver verbinden oder alternativ einen Webbrowser verwenden. Zusätzlich zu diesen Komponenten gibt es noch die sogenannten Central Services (CS), bzw. Zentralkomponenten, welche für eine reibungslose Clusterkommunikation zwischen mehreren Applikationsservern, die Lastverteilung und die transaktionale Integrität sorgen (zentraler Datensperremechanismus).

Im Applikationsserver selbst sorgt der Dispatcher für die Zuteilung von Requests auf Workprozesse nach dem Multiplexing Prinzip (siehe Abschnitt 2.2.7). Für die Zugriffe über das Internetprotokoll steht ein eigener Dispatcher zur Verfügung, der Internet Communication Manager (ICM), welcher Objekte ausfiltert, die vom Applikationsserver berechnet werden müssen (dynamische Web Inhalte). Diese werden weitergereicht. Statische Internetanfragen werden direkt beantwortet. Eine schematische Darstellung des ABAP Applikationsservers befindet sich in Abbildung 6.2.

Für eine Skalierung dieser Systemarchitektur können mehrere Applikationsserver eingesetzt und die verwendeten Datenbanken verteilt installiert werden. Dadurch ist die Anzahl der zugreifenden Benutzer praktisch nicht limitiert.

Java

Die Systemarchitektur für die Java Plattform entspricht der zuvor vorgestellten ABAP Architektur. Im Mittelpunkt steht ein JEE (Java Enterprise Edition¹⁴⁰) kompatibler Applikationsserver, welcher für die Datenhaltung ebenfalls eine Datenbank benützt. Die

¹³⁹Eine sehr gute Einführung gibt [SAP03, S.3 ff.].

¹⁴⁰Java Standardisierung für Serverdienste, diese erweitern den JSE, die Java Standardfunktionalität, um Services wie einen Webserver, Messaging Komponenten etc..

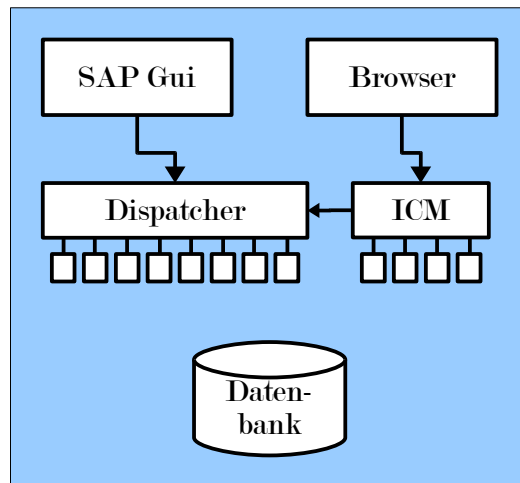


Abbildung 6.2: Darstellung der Komponenten des ABAP Applikationsservers

Visualisierung geschieht meistens über Webanwendungen, somit ist der Browser meist die „Präsentationskomponente“ einer Java-basierten Applikation. Zu diesen drei Komponenten kommen wieder die CS Dienste für die Clusterkommunikation, die Lastverteilung und die Sperrverwaltung.

Die SAP hat, wie auch andere Hersteller von JEE-kompatiblen Applikationsservern, eigene Services auf die bestehenden Services des JEE Standards aufgesetzt. So existiert beispielsweise ein eigenes Java Connector (JCo) Service zur Verbindung auf SAP ABAP Systeme oder die WebDynpro Runtime, eine eigene MVC¹⁴¹- und Metadaten-basierte Servlet Architektur.

Die Systemarchitektur des Java Teils kann in Abbildung 6.3 betrachtet werden.

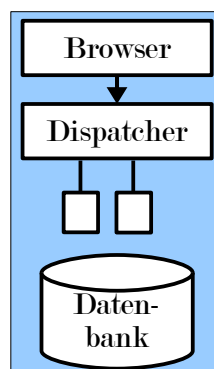


Abbildung 6.3: Darstellung der Komponenten des Java Applikationsservers

¹⁴¹Das MVC Konzept ist sehr gut bei [ZW01, S.151 ff.] erklärt.

Im Bild wurden wiederum die CS Dienste weggelassen. Einer der wesentlichen Unterschiede zum ABAP Teil in Abbildung 6.2 ist die Größe und Anzahl der Prozessknoten, die unter dem Dispatcher zu sehen sind. Dies ist durch die vollständige Objektorientierung von Java bedingt, die ein Vorausladen von wesentlich mehr Programmen (Klassen) erfordert, um einen schnellen Zugriff auf die Serverfunktionen zu ermöglichen. Der Ressourcenbedarf ist also beim Java Teil höher als beim ABAP Teil, daher werden die Prozesse unter dem Dispatcher anders dargestellt als beim ABAP Teil. Die restlichen Eigenschaften des Java Teiles gleichen jenen des ABAP Teiles.

Kombination ABAP + JAVA

Wie bereits dargelegt ist die Positionierung vonseiten SAP jene, dass der Applikationsserver ABAP eher für Datenbankanwendungen und der Applikationsserver Java eher für die Visualisierung und die Connectivity einer EDV gestützten Lösung eingesetzt wird. Daraus ergibt sich der Umstand, dass drei mögliche Kombinationen auftreten können:

- **Nur ABAP:** In Szenarien, die keine aufwändigen Webanwendungen benötigen oder in solchen, die noch in einer Zeit umgesetzt wurden, zu der es den Java Applikationsserver nicht gab, wird nur der ABAP Teil verwendet. In einem Großteil der Standardszenarien des SAP ERP (früher: R/3) kommt man ohne den Java Teil aus.
- **Nur Java:** In reinen Webanwendungen, die sehr wenig in einer Datenbank speichern müssen, wird zumeist nur der Java Server benötigt. Die SAP liefert beispielsweise ihr Enterprise Portal als reine Java-Anwendung aus.
- **ABAP + Java:** Es gibt Szenarien, bei denen eine Applikation einerseits viele Daten verarbeiten muss und eine starke Datenbankaffinität besitzt und andererseits aber in der Darstellung sehr verstärkt auf Webinhalte setzt. Ein Beispiel für solche Szenarien wäre die Employee Self Service Lösung (ESS), bei der Mitarbeiter ihre HR Services aus dem Intranet beziehen können (z.B. Gehaltszettel, etc.).

Das Kombinationsszenario aus beiden Systemen kann also bedeuten, dass zwei Systeme zugleich verwendet werden. Wenn der Zugriff nun mittels Browser erfolgt, so ist davon auszugehen, dass es zwei verschiedene Zugriffswege gibt. Einen auf das ABAP System und einen auf das Java System. Über einen eigenen Verteilungsmechanismus können die Systeme so verbunden werden, dass es für den Endbenutzer den Anschein hat, als wenn er auf ein einziges System unter der gleichen Webadresse (URL) zugreift. Die zugehörige Komponente ist der SAP Web Dispatcher, welcher in dieser Arbeit nicht von Relevanz

sein wird. Eine kleine Gesamt-Architekturskizze sei in Abbildung 6.4 angefügt.

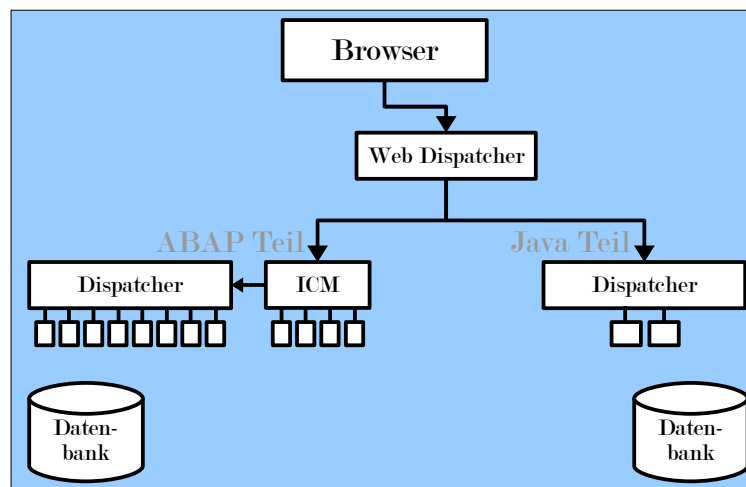


Abbildung 6.4: Vereinfachte Darstellung des SAP Web Dispatcher

Der Request des Browsers trifft am Web Dispatcher ein und wird dort nach dem URL Präfix (dem Beginn des nicht-serverspezifischen Teiles der URL) kategorisiert und dann entweder an den ABAP oder an den Java Teil geschickt. Die Antwort, der Response, wird wieder zurück an den Browser übermittelt¹⁴².

Mit der in dieser Arbeit behandelten Release, Netweaver 7.0 waren auch schon Details über die nächste Version, Netweaver 7.1 bekannt. So wird von SAP eine leichtgewichtige, abgespeckte Java Virtual Machine auf dem ABAP Applikationsserver zusätzlich zu den ABAP Workprozessen angeboten werden. Dies soll ein Coding von Java und ABAP im gleichen Programm ermöglichen oder kleinere Java Abläufe direkt aus dem ABAP heraus. Es wird aber von der SAP dementiert, dass dies den JEE Server komplett ablösen wird.

In der weiteren Folge dieser Arbeit wird der Java Teil eine weniger große Rolle spielen.

6.1.3 Mandantenkonzept

Das SAP ABAP System ist mandantenfähig ausgestaltet. Ein Mandant ist hierbei semantisch als ein Datenbereich beschreibbar, der von anderen technisch getrennt ist. Somit laufen Vorgänge, die sich in diesem Datenbereich abspielen, von anderen Vorgängen getrennt und von anderen Teilnehmern am System nicht einsehbar.

Technisch bedeutet die Mandantenfähigkeit, dass sämtliche Datentabellen auf der Datenbank des Systems einen zusätzlichen Primärschlüssel besitzen, nämlich den Mandan-

¹⁴²Mehr dazu in [KK05, S.487 ff.].

tencode. Dies ist bei SAP eine Zahl zwischen 000 und 999. Datentabellen werden im System so definiert, dass diese Tabellen Daten halten, die für betriebswirtschaftliche Vorgänge nötig sind sowie z.B. auch Tabellen, die die Benutzerstammsätze enthalten. Nicht mandantenfähig wurden die Objekte des Repository in Tabellen abgelegt, wie z.B. Programme und Datenstrukturen.

Ein Mandant findet bei einer beabsichtigten betriebswirtschaftlichen Trennung von Daten zwischen Benutzern eines Systems Verwendung. In der Praxis kann es sein, dass man auf einem System z.B. einen eigenen Schulungsmandanten einrichtet. Schulungsteilnehmer können also mit Daten im System arbeiten, die nicht die auf dem gleichen System befindlichen Produktivdaten verändern. Man verwendet öfters auch Mandanten, um verschiedene Tochtergesellschaften auf ein zentrales System zugreifen zu lassen. Da man nicht möchte, dass sich das Unternehmen A die Betriebsdaten von Unternehmen B ansehen und verändern kann benützt man einen Mandant, der dies verhindert.

Ein Mandant hat jedoch auch seine Grenzen: Wenn ein Unternehmen eigene Erweiterungen oder Modifikationen (siehe dazu Abschnitt 3.2.1) an den Programmen vornehmen will, welche die betriebswirtschaftlichen Abläufe kontrollieren, so ist dies nur dann möglich, wenn das System alleinig benutzt wird. Andernfalls würde das Unternehmen die Programme auch für das andere angeschlossene Unternehmen ändern.

Im folgenden wird bei Transporten von neuen Entwicklungen immer davon gesprochen werden, dass diese in einem speziellen Mandant durchzuführen sind. Nachdem das Transportieren nicht zwingend ein betriebswirtschaftlicher Vorgang ist, sei nur die Tatsache relevant, dass sich über den Mandant verschiedene Benutzer anmelden können und diese pro Mandant eigens zu definieren sind. Noch kürzer: Bei der Anmeldung an ein SAP System (ABAP Teil) müssen Sie einen Benutzernamen und ein Passwort angeben und wissen, in welchem Mandant dieser angelegt wurde.

Der Vollständigkeit halber sei erwähnt, dass das SAP Java System nicht mandantenfähig ist. Will man also Anwendungen in der Programmiersprache Java entwickeln, so muss man für die Abbildung mehrerer eigenständiger Unternehmen ein weiteres System installieren¹⁴³.

6.1.4 Entwicklungsumgebungen

Für beide Programmiersprachen existieren bei SAP unterschiedliche Entwicklungsumgebungen. Diese werden in den nachfolgenden Unterabschnitten erläutert.

¹⁴³Weitere, sehr detaillierte Ausführungen zu Mandanten bei [SAP04, S.14 ff.].

ABAP

Nachdem ABAP den kompletten Quellcode für alle Programme sowie sämtliche Definitionen (z.B. von Datenbanktabellen) in der Datenbank selbst ablegt, wird mit dem ABAP System auch ein zentrales Verzeichnis, das sogenannte Repository, installiert.

R/3 Repository

Zentrale Ablage für alle Entwicklungsobjekte der ABAP Workbench. Diese Entwicklungsobjekte beinhalten ABAP-Programme, Bildschirme und Dokumentation¹⁴⁴.

Die ABAP Workbench ist die zentrale Entwicklungsumgebung und bietet einen Zugriff auf die verschiedenen Entwicklungswerkzeuge, wie z.B. den ABAP Editor oder den Menu Painter. Die ABAP Workbench selbst ist in ABAP geschrieben und läuft auf dem Applikationsserver ab. Somit arbeiten alle Entwickler zentral mit der gleichen Entwicklungsumgebung. Die Entwicklungsumgebung verwaltet Änderungsaufträge, durch die Änderungen an Objekten für einen spezifischen Entwickler reserviert werden können¹⁴⁵. Das Repository, welches durch die einzelnen Programmieraufgaben erweitert bzw. geändert wird, ist mit zahlreichen Features ausgestattet, wie z.B. der Möglichkeit der Versionierung und dem automatischen Abgleich von Modifikationen und Erweiterungen (siehe Kapitel 3.2.1).

Java

Die Java Entwicklungsumgebung wird in der Folge nicht weiter erwähnt werden. Dennoch soll an dieser Stelle kurz auf diese Entwicklungsumgebung eingegangen werden. Die Java Entwicklung kann aufgrund der hohen Abhängigkeit bei der vollständig objektorientierten Programmierung nicht gemeinsam auf einem zentralen System erfolgen, zu sehr würden sich die Änderungen mehrerer Entwickler an den gleichen Quellen aufeinander auswirken.

Hier ist die SAP den Weg gegangen, eine bestehende Lösung, die Plattform Eclipse, zu verwenden und mit eigenen Werkzeugen (Plugins) auszustatten. Der daraus resultierende Developer Workplace bietet einem Entwickler die Möglichkeit, lokal den Quellcode eines Programms von einem Server zu synchronisieren, diesen zu ändern, zu testen und dann einer zentrale Weiterverarbeitung (Test, Packaging) zuzuführen.

Die Möglichkeiten und Techniken werden ausführlich in einem kompletten Kapitel bei [KK05, S.37 ff.] erläutert.

¹⁴⁴[MFMS00, S.634]

¹⁴⁵Dazu mehr in Abschnitt 6.2.

6.2 Das Korrektur- und Transportwesen

Das Korrektur- und Transportwesen (KTW) ist bei SAP Systemen unter anderem dafür zuständig, dass Änderungen von den Entwicklungssystemen bis zu den Produktivsystemen nach entsprechender Freigabe vollständig deployed werden können¹⁴⁶. Der Teil von KTW, welcher für die Übertragung und Installation am Zielsystem verantwortlich ist, wird auch Transportmanagementsystem (TMS) genannt.

In den folgenden Abschnitten soll die Architektur, die Einrichtung und der Prozess bei der täglichen Benutzung angeschnitten werden. In Kapitel 7 wird dann ein Vergleich zwischen TMS und SAP Change Request Management (siehe nächster Abschnitt 6.3) erfolgen.

6.2.1 Architektur des TMS

Wenn zwischen zwei Systemen Inhalte übertragen werden, so produziert das exportierende System eine Datei, die sogenannte Exportdatei¹⁴⁷. Diese enthält sämtliche Programme und Daten, die für den Transport zwischen den Systemen im Rahmen eines Transportauftrags selektiert wurden¹⁴⁸. Wird der Transportauftrag als „freigegeben“ markiert, so wird die Datei auf ein zentrales Verzeichnis abgelegt, auf welches beide Systeme Zugriff haben. Alternativ können auch dezentrale Verzeichnisse gewählt und die Datei von Hand kopiert werden¹⁴⁹.

Das importierende System liest aus dem dort konfigurierten Verzeichnis und bildet eine Importwarteschlange, die Import Queue. Der Verantwortliche kann einzelne Dateien auswählen und diese im System importieren. Das entspricht dem Installieren. Bei der ersten Ausführung werden allfällig geänderte Programme dann kompiliert (siehe Kapitel 6.1).

Der Weg vom Entwicklungssystem bis in die produktiven Systeme wird als Transportweg bezeichnet. Ein Transportweg kann über die erweiterte Transportsteuerung auch von einem auf mehrere Systeme gehen und eine komplexe Struktur aufweisen. So wird das TMS den Anforderungen von internationalen Systemlandschaften gerecht, die in Kapitel 2.3.3 anhand von Abbildung 2.11 bereits erläutert wurden.

Transportwege werden verwendet, um die Rolle jedes Systems und den Fluss der Änderungsaufträge anzugeben¹⁵⁰.

¹⁴⁶Wie bereits in Kapitel 2.3.2 skizziert.

¹⁴⁷Gelegentlich wird diese Datei auch schlicht „der Export“ genannt.

¹⁴⁸Dazu mehr im Unterpunkt „Prozess“.

¹⁴⁹Dies wird als Transportgruppe bezeichnet, der interessierte Leser sei auf [MFMS00, S.236] verwiesen.

¹⁵⁰[MFMS00, S.237]

Besondere Beachtung beim Transportieren verdient noch die Art der transportierten Daten. Im Normalfall werden über Transporte entweder Programme oder Customizing übertragen. Beide ergeben im Zusammenspiel eine betriebswirtschaftliche Arbeitsumgebung. Wie in Kapitel 6.1 beschrieben haben geänderte Repository Objekte, zu denen Programme dazuzählen¹⁵¹, eine Auswirkung auf alle Mandanten. Applikatorisches Customizing ist aber mandantenspezifisch, muss also in jeden Mandant, wo es benötigt wird, separat übernommen werden.

6.2.2 Einrichtung

Die komplette Einrichtung von SAP KTW, im Speziellen hier TMS, geschieht immer im Mandanten 000. Dies ist der Systemmandant und wird für alle übergreifenden technischen Konfigurationen verwendet. Sämtliche TMS Einstellungen werden in einer Transportdomäne vollzogen. Dies ist eine logische Gruppierung für Systeme, die zu einem Transportverbund gehören.

Alle R/3-Systeme¹⁵², die Sie mit dem TMS zentral verwalten möchten, bilden eine Transportdomäne. Innerhalb einer Transportdomäne müssen alle R/3 Systeme eindeutige Systemidentifikationen sowie identische Transportwege aufweisen¹⁵³.

Nach dem Anlegen einer Transportdomäne wird ein System gewählt, welches als Transport Domänencontroller dient. Auf diesem System werden die Transportwege modelliert und dann auf alle Server in der Transportdomäne publiziert. Somit wissen alle Systeme in der Domäne über den aktuellen Stand der Transportwege Bescheid¹⁵⁴. Die Transportwege werden in dem über den Schnellcode („Transaktionscode“) STMS erreichbaren Programm administriert. Einen Bildschirmabgriff eines Transportwegs über 4 Systeme kann man in Abbildung 6.5 sehen. Um nun mit den Transportwegen arbeiten zu können,



Abbildung 6.5: Screenshot: Transportwege modellieren in der Transaktion STMS

müssen die entsprechenden Berechtigungen am SAP System vergeben, sowie die Benutzer fallweise als Entwickler registriert sein. Da Berechtigungen den Umfang dieses Arbeit

¹⁵¹Der SAP Begriff lautet hier Workbench Transportaufträge; siehe auch [MFMS00, S.338].

¹⁵²Zum Zeitpunkt der Drucklegung des Buches sprach man bei ABAP Systemen immer von R/3 Systemen. Die Aussage ist universell für ABAP Systeme gültig, Anmerkung des Autors.

¹⁵³[MFMS00, S.234]

¹⁵⁴Backupkonzepte für Ausfallsicherheit findet der interessierte Leser wiederum bei [MFMS00, S.235 ff.].

sprengen würden sei auf die SAP Online Hilfe zu TMS verwiesen. Dort findet sich für jegliche Aufgabe das entsprechende Berechtigungsobjekt.

Die Zuordnung von Transporten zu Releases kann über das Projektsystem erfolgen. Dazu muss jede definierte Release im SAP Projektsystem angelegt sein. Eine übergreifende Verwaltung von Releases über die Systemlandschaft ist im SAP Standard nicht vorgesehen und bedarf einer Erweiterung des SAP Standard¹⁵⁵.

6.2.3 Prozess

Der TMS Prozess wird in Abbildung 6.6 dargestellt¹⁵⁶.

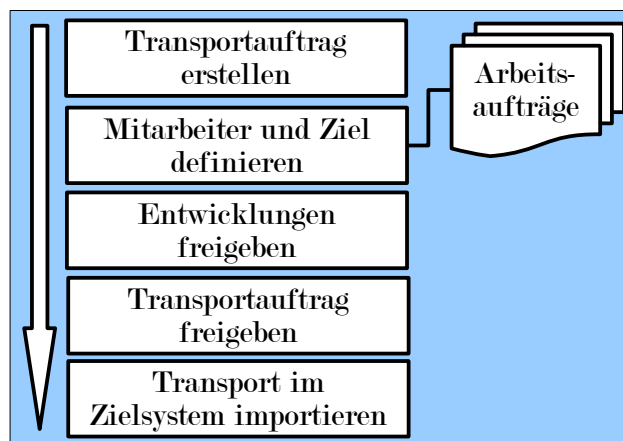


Abbildung 6.6: Der Prozess von der Erstellung des Transportauftrags bis zum Import im Zielsystem

Der Projektleiter (oder, in Anlehnung an ITIL: der Change Manager) legt einen Transportauftrag an.

Dem Auftrag werden die Mitarbeiter zugewiesen, die Objekte in den Auftrag hinzufügen sollen. Diese Mitarbeiter bekommen dann Arbeitsaufträge zugeteilt. Weiters ist zu spezifizieren, auf welches Ziel der Auftrag kommen soll. Dies ist nichts Anderes als der Transportweg. Zusätzlich kann man ein Projekt zuweisen. Hier gäbe es die Möglichkeit, ein Projekt im SAP Projektsystem für jede Release zu erstellen.

Alle zugewiesenen Mitarbeiter können nun beim Sichern ihrer Änderungen diese an den Transportauftrag binden. Sind alle Änderungen erledigt und getestet, kann jeder Entwickler für sich die Änderungen freigeben.

Nach der Freigabe aller Änderungen sollte noch ein vom Projektleiter bzw. Change

¹⁵⁵Siehe Kapitel 3.2.1.

¹⁵⁶Weitere Informationen zum Standardtransportprozess finden sich bei [Hüb02, S.539 ff.].

Manager koordinierter Test stattfinden. Ist dieser erfolgreich, so gibt er den Transportauftrag frei. Mit der Freigabe wird der Export erstellt und in das Transportverzeichnis gestellt.

Der letzte Schritt ist dann am Zielsystem der Import des Transports über die Importqueue. Dort kann über das Transportprotokoll geprüft werden ob der Import erfolgreich abgelaufen ist.

6.2.4 Transporte in Systemlandschaften

Sind mehrere Systeme vorhanden, beispielsweise in einer 3-System Landschaft (Entwicklungs-, Test- und Produktivsystem), so läuft der im vorigen Abschnitt erläuterte Prozess einmal ab, um den Transport vom Entwicklungs- in das Testsystem zu übertragen. Ab dem Testsystem muss in der Regel (je nach Konfiguration) keine weitere Freigabe erteilt werden. Es genügt, den Transport im nachfolgenden System (Produktionssystem) ebenfalls zu importieren.

Besteht eine Systemlandschaft aus mehreren verbundenen Systemen (z.B. ein ERP und ein ERP2 System parallel), so ist zu beachten, dass für beide Systeme, das ERP und das ERP2 System nicht synchron importiert werden kann. Es kommt also für kurze Zeit zu einem inkonsistenten Zustand zwischen den Releaseständen, weil der Projektleiter auf beiden Systemen nacheinander die jeweiligen importieren muss. Dies kann problematisch sein, wenn sich die Änderungen aufeinander beziehen. Es muss in solch einem Fall sichergestellt sein, dass keine Benutzer am System mit den zu aktualisierenden Komponenten arbeiten. Dazu kann man verschiedene Verfahren anwenden, z.B. eine kurzfristige Sperre der entsprechenden Programme.

6.2.5 Ausblick: CTS+

Das aktuelle ¹⁵⁷ Korrektur- und Transportwesen (Change and Transport System, CTS in Englisch) ist nicht in der Lage, synchron zu transportieren. es ist auch nicht in der Lage, bei Systemen mit ABAP und Java Teil den Java Teil mit entsprechenden Aktualisierungen zu versorgen. An dieser Stelle sei erwähnt, dass mit der nächsten technischen SAP Release das CTS um die Möglichkeit erweitert werden wird, ABAP Transportaufträge mit externen Files, wie z.B. Java Programmen auszustatten und dann entsprechend auf mehrere Systeme zu transportieren.

Zum Zeitpunkt der Drucklegung dieser Arbeit war ein Termin für die generelle Freigabe an Kunden noch nicht bekannt, die Vorabversion, die dem Autor zur Verfügung stand,

¹⁵⁷Stand Anfang 2008, Release kleiner gleich SAP Netweaver 7.0.

war allerdings einsatzfähig¹⁵⁸.

6.3 Change Management mit dem SAP Solution Manager 4.0

Mit dem SAP Solution Manager (SolMan) hat die SAP in den letzten Jahren ein Tool auf den Markt gebracht, das einige Problemstellungen von komplexen, hoch integrierten Systemlandschaften lösen soll. Außerdem sind einige der angebotenen Funktionalitäten besser an ITIL angenähert als bisherige Ansätze der deutschen Softwareschmiede.

Der Solution Manager selbst ist ein SAP CRM (ERP2)-System mit einigen zusätzlichen Entwicklungen. Dass ein CRM System als Basis gewählt wurde liegt an der Konzeption von CRM. Es ermöglicht, Geschäftsfälle generisch anzulegen und dazu jegliche Daten abzulegen. Ob es nun Angebote, Bestellungen oder auch Anfragen sind, alles hat eine eindeutige Geschäftsfallnummer. So etwas eignet sich gut als Basis für z.B. einen Service Desk Betrieb.

Weitere Funktionalitäten des Solution Manager wären ein zentrales Monitoring (auch auf Prozessebene) sowie eine zentrale, grafische Systemlandschaftsübersicht.

6.3.1 Allgemeines

Überblick

Change Request Management versucht die offensichtlichen Lücken zwischen den ITIL Change Prozessen und dem klassischen SAP Releaseprozess über TMS zu schließen. Dabei wurde die Applikation vornehmlich für drei Arten von Änderungen vorgesehen¹⁵⁹:

- Änderungen an Geschäftsprozessen, Implementierungs- und Migrationsprojekte
- Laufende Wartung
- Dringende Korrekturen

Änderungen werden in Change Projekten realisiert, die - ähnlich wie beim TMS - im Projektsystem abgebildet werden. Zusätzlich bietet Change Request Management die Möglichkeit, die Projekte in die Satellitensysteme zu synchronisieren, sodass die Releaseinformationen in der gesamten Systemlandschaft zur Verfügung stehen. Dies vereinfacht die durchaus komplexe Releasepflege in großen Systemumgebungen.

Die drei Säulen von Change Request Management sind die Administration des Changes

¹⁵⁸Zum gesamten Thema CTS+ findet sich bereits Doku in [SAP07, S.5-1 ff.].

¹⁵⁹Vgl. bei [SAP07, S.7-5].

an sich (Dokumentation, Kategorisierung, Freigabe, Berichtswesen), das Projektmanagement (Dokumentation, Konzeption und Einstellungen, Schulungsorganisation) sowie die Transportlogistik (Customizing, Transporttools, -einplanung und -verfolgung)¹⁶⁰. In all diesen Bereichen bietet Change Request Management umfangreiche Tools an.

Systemlandschaft

Für große Systemlandschaften ist das Change Request Management sehr gut geeignet, weil der Change Manager direkt auf die verschiedenen betroffenen Systeme transportieren kann, ohne sich auf diese separat verbinden zu müssen.

Es werden neben den Korrekturen auch die in ERP Szenarien oft benötigten Abgleiche unterstützt¹⁶¹. So kann sichergestellt werden, dass eine neue Release des ERP Herstellers mit den Eigenentwicklungen zusammenstimmt. Diese Retrofits oder Umzüge von Kopien werden in dieser Arbeit allerdings nicht weiter erläutert, da sie für den theoretischen Aspekt gar nicht und für die tägliche Praxis kaum von Bedeutung sind.

Eine Konfigurationsdatenbank ist im SAP Ansatz derzeit nicht enthalten. Es gibt aber Möglichkeiten, zentral die Releasestände in einem System abzufragen sowie zentral technische Details anzeigen zu lassen. Dinge wie das Hardwareinventar oder Clusterkonfigurationen sind nicht enthalten.

Berichte

Die SAP stellt in der Standardauslieferung des Solution Manager eine Berichtsoberfläche zur Verfügung. Leider sind Standardberichte, wie z.B. die Anzahl der monatlichen Nottransporte noch nicht vorkonfiguriert.

Das Berichtswesen selbst ähnelt den ERP Berichten, denen von Kunden oft nachgesagt wird, dass man sie am besten in Excel nachbearbeitet.

6.3.2 Der Prozess im Detail

Der Prozess, wie er im Change Request Management implementiert wurde, ist in Abbildung 6.7 dargestellt¹⁶².

Hier wird der RFC gleich nach seiner Erfassung gefiltert und genehmigt. Danach wird die Dringlichkeitsentscheidung getroffen. Handelt es sich um einen dringenden Change, so wird für die Änderung ein eigener Arbeitsauftrag mit Mitarbeitern wie im TMS

¹⁶⁰[SAP07, S.7-7]

¹⁶¹Siehe dazu auch Kapitel 3, im Speziellen Kapitel 3.3.

¹⁶²Vgl. [SAP07, S.7-20 und 7-32].

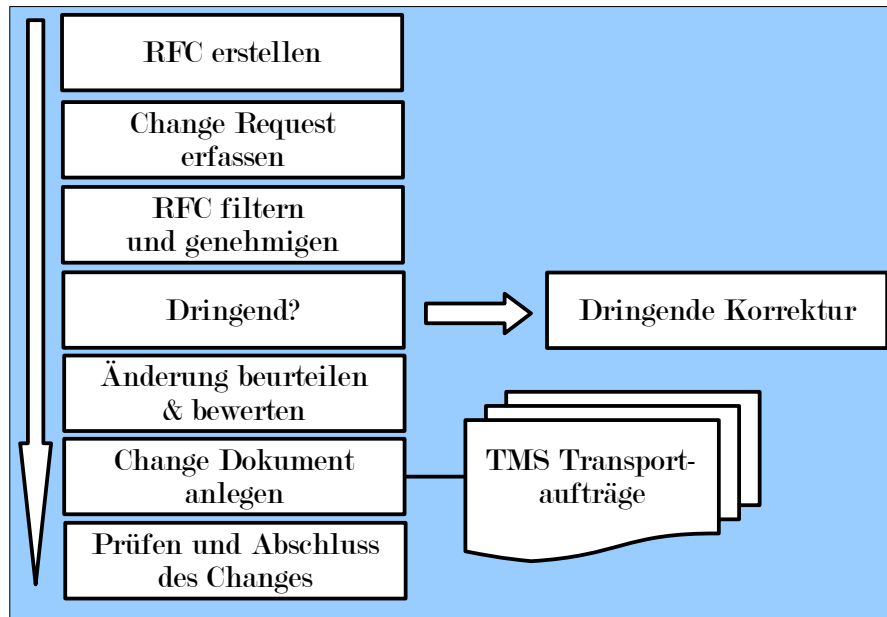


Abbildung 6.7: Der Change Request Management Prozess

dafür angelegt. Innerhalb dieser Aufträge geschieht die Abarbeitung des Changes. Ist die Änderung ein normaler Change, so erfolgt nun die Klassifizierung. Über das nun angelegte Change Dokument werden dann einzelne Transportaufträge von Hand angelegt und zugeordnet. Diese lassen sich dann zentral mit dem Change transportieren. Wenn alle releaseabhängigen Schritte erfolgreich im Change Dokument aktualisiert wurden, kann der Change Manager zentral den Transport auf die betroffenen Zielsysteme durchführen; zunächst auf die Qualitätssicherungssysteme und dann auf die Produktivsysteme. Ein direktes Login ist nicht mehr erforderlich. Auch die Transportprotokolle sind zentral verfügbar. Sind alle Änderungsaufträge durch den Freigabe- und Deploymentprozess gelaufen, kann der RFC geschlossen werden.

Der Entwicklungsverantwortliche kann einzelne Objekte als kritische Objekte definieren, deren Änderung zusätzlich genehmigungspflichtig ist. So können unerwünschte Änderungen schon bei der Entwicklung erkannt werden. Hierbei ist lediglich zu erwähnen, dass in einem ERP System sehr viele solcher Objekte existieren, die nicht geändert werden sollten.

Es gibt ein paar wesentliche Unterschiede zwischen normaler und dringender Änderung, die wie folgt zusammengefasst werden können¹⁶³:

¹⁶³[SAP07, S.7-40]

Normale Änderung

- Es gibt keine spezifische Aufgabenliste, es wird die des Wartungszyklus (TMS) benutzt.
- Transportaufträge müssen manuell erstellt werden.
- Transporte werden in den Standardwartungszyklen durchgeführt.
- Entwickler und Tester müssen nicht zwingend verschieden sein, es ist lediglich eine Warnmeldung¹⁶⁴.

Dringende Änderung

- Es gibt eine individuelle Aufgabenliste für den Change.
- Transportaufträge werden automatisch erstellt.
- Transporte können automatisch auf die Folgesysteme verteilt werden, verbleiben aber in der Warteschlange zum weiteren Import.
- Entwickler und Tester müssen zwingend verschiedene Personen sein.

Im gegenständlichen Kapitel hat die Systemarchitektur von SAP Change Lösungen besondere Beachtung gefunden. Gleichsam wurde der Prozess im Korrektur- und Transportwesen behandelt und mit SAP Change Request Management verglichen. Die Ergebnisse wurden bewertet und detailliert beschrieben.

Im nachfolgenden Kapitel wird dieser Vergleich weiter fortgeführt und mit den bisherigen Kapiteln abgeglichen. Weiters werden Überlegungen zum Risikomanagement und eine Kosten/Nutzen Betrachtung angefügt und damit das Praxisbeispiel vorbereitet.

¹⁶⁴Warnmeldungen können im SAP System einfach bestätigt („weggeklickt“) werden.

7 Analyse und Reflexion

7.1 Korrektur- und Transportwesen und Solution Manager - ein Vergleich

Vergleicht man das Management von Änderungen über SAP KTW (Kapitel 6.2) und Solution Manager (Kapitel 6.3), so fällt auf, dass der Solution Manager eine einzige Aufgabe erfüllt, nämlich die Zentralisierung von Änderungen an SAP Programmen und Applikationen.

Die zugrundeliegende Technik ist die des KTW, auch die verwendeten Programme sind die gleichen. Für den Entwickler erfordert das zusätzlich zu den üblichen Freigabeschritten noch die Pflege des Change Dokumentes.

Der Projektleiter bzw. Change Manager kann sich durch die Konzeption der Dokumentpflege beim Change Management über den Solution Manager allerdings darauf verlassen, dass zumindest ein Tester das Programm getestet hat, bevor überhaupt die Möglichkeit zum Transport besteht. Die Testfreigabe durch den Entwickler entfällt.

Ein weiterer Unterschied bei der Verwendung von Change Management ist, dass in den einzelnen, im Zuge des Changes erstellten Änderungsaufträgen, kein unmittelbarer Transport und Import im Zielsystem erfolgt. Der eigentliche (externe) KTW Prozess endet bei der Freigabe. Der Transport erfolgt dann zentral und für das gesamte Projekt (=Release) auf einmal. Pessimistisch ausgedrückt geht ein wenig Flexibilität verloren, allerdings zugunsten einer konsistenten Transportlogistik.

Zu den entsprechenden Berechtigungen - an sich nicht Thema der Arbeit - sei ergänzt, dass jegliche Berechtigung sowohl im Solution Manager als auch in den Satellitensystemen separat gesetzt werden kann. Dadurch können Projektteams (oder auch Change Manager) entsprechend den in Unternehmen verwendeten Konzepten für die einzelnen Funktionen von Change Request Management berechtigt werden. Anzumerken ist in diesem Zusammenhang, dass sich für größere Landschaften mit Solution Manager auch noch die Einführung einer zentralen Rollen- und Benutzerverwaltung (ZBV) anbietet, die eine übergreifende Berechtigungsadministration über das gesamte Unternehmen ermöglicht.

7.2 Risiken beim Einsatz der SAP Softwarelogistik

Die Risiken des Deployments von Änderungen im ERP Umfeld sind - wie bereits in Kapitel 3 erläutert, mitunter beträchtlich. KTW und speziell der Einsatz von Change Request Management kann diese minimieren, aber nicht komplett ausschalten.

Die folgenden häufig auftretenden Risiken sind bei der Planung zu berücksichtigen¹⁶⁵:

Vergessene Transporte

Wenn man in einer Systemlandschaft sehr viele Änderungen durchführt, kann es sein, dass man bei einer Strategie der einzelnen Importe Transporte vergisst. Dies kann insbesondere dann passieren, wenn ein Entwickler das Programm eines anderen Entwicklers verwendet, welches, von jedem der beiden, in einen anderen Transportauftrag gelegt wurde. Ist nun einer der Transporte bereits importiert worden, der andere aber z.B. noch nicht freigegeben, so kommt es zu Inkonsistenzen.

Wie hoch dieses Risiko ist hat der Autor selbst bei einem Kunden in der Papierindustrie erlebt. Die beiden Entwickler führten Änderungen am zentralen Eingang von Kundenbestellungen durch. Dabei wurde die Bestellung vom Programm des einen an eine Funktion des anderen Entwicklers übergeben. Einer der beiden Aufträge wurde in die Produktion transportiert, der andere war noch nicht fertiggestellt. Nach 6 Stunden bemerkte man den Fehler. Das Resultat war, dass alle Kundenbestellungen innerhalb dieser 6 Stunden verloren waren und die Aufträge bei den Kunden neu eingefordert werden mussten!

Mit Change Request Management ist die Angabe eines Releases zwingend erforderlich, welches als Projekt in den Satellitensystemen angelegt wird. Über den zentralen Import können nur gesamte Releaseprojekte importiert werden. Wenn also ein durchgängiges Change Management verwendet wird, können auf diese Weise keine Inkonsistenzen innerhalb von Komponenten, die für die gleiche Release entwickelt werden, mehr auftreten. Vorsicht ist immer noch bei gleichzeitigen Releases bzw. gleichzeitigen Fixes am alten und Entwicklung an eine neue Release geboten.

Inkonsistenzen beim Transport

Vielfach müssen Transporte nicht nur kombiniert importiert werden, sondern auch in der richtigen Reihenfolge. Wenn die SAP mit einem Transport eine Änderung einführt, die der Kunde mit einem anderen Transport modifiziert, so muss konsequenterweise der SAP Transport früher importiert werden.

¹⁶⁵Für eine weiterführende, detailliertere Ausführung vgl. [SAP07, S.6-11 ff.]!

Eine weitere Quelle von Inkonsistenzen sind zurückgestellte Entwicklungen. Wird während der Entwicklung einer Komponente diese Arbeit zurückgestellt, so verbleiben die Komponenten im Importpuffer der weiteren Systeme während diese in der Entwicklung „halbfertig“ verweilen. Wird in der Zwischenzeit einer der Transporte unabsichtlich weiter importiert, kann das schlechte Folgen haben.

Parallele Projekte

Wenn Unternehmen nicht mit Releases arbeiten sondern projektspezifische Deploymentpläne entwerfen, so setzen sie sich einem höheren Risiko aus. Einer der Gründe liegt darin, dass die Test- und Qualitätssicherungssysteme mitunter nie den Stand der Produktion erreichen. Das kann bedingt sein durch die unterschiedlichen Testphasen in denen sich Produkte befinden oder durch mögliche Abhängigkeiten unter den Projekten. Es kann auch vorkommen, dass Fehler in einem neuen Produkt das Testen des fast fertigen Produktes verhindern.

7.3 SAP Change Request Management und ITIL

SAP Change Request Management unterstützt laut Aussage von SAP den ITIL Prozess. Aufgrund der Zertifizierungsthematik bei ITIL (siehe Kapitel 5.1.3) ist hier natürlich ein gewisser Spielraum gegeben. Change Request Management vereint - etwas undifferenziert - den Release- und Change Prozess nach ITIL.

Diese unspezifische Sichtweise soll am Beispiel der Verantwortlichkeiten aufgezeigt werden. ITIL kennt eine klare Trennung von Release- und Change Manager. Dies aus gutem Grund. Die Aufgabe des Change Managers besteht darin, die Änderungswünsche zu qualifizieren, deren Umsetzung evaluieren zu lassen und dann die Entscheidung über die Umsetzung einzuleiten. Erst dann geht der Auftrag für die Entwicklung und die Qualitätssicherung an den Release Manager. Der Release Manager entscheidet, wie die Entwicklung vorgenommen wird. Im Falle von SAP Change Request Management ist diese Trennung unscharf. Sie ist wohl umsetzbar, aber gerade bei den Abhängigkeiten zwischen dem Change Projekt und den einzelnen Transportaufträgen scheint ein weit höherer Aufwand als nötig gegeben zu sein, wenn hier zwei Personen die verschiedenen Rollen einnehmen.

Der SAP Prozess unterscheidet sich von ITIL in einem wesentlichen Punkt, der die oben beschriebene Problematik noch verschärft. Ist im ITIL Prozess eine Evaluierungsphase gefolgt von einer Entscheidungsphase, und dann einer Umsetzungsphase, so beginnt bei SAP die Entscheidung über Annahme oder Ablehnung des RFC noch vor dessen Bear-

beitung¹⁶⁶. Es ist zu bezweifeln, dass ein Change Manager (nicht der Release Manager!), der sämtliche Changes für das gesamte Unternehmen durchführt, über hinreichende Informationen verfügt, um über einen Change mit fachspezifischen Inhalten zu entscheiden. Der SAP Prozess sieht eine Evaluierungsphase nicht vor. Es würde also zu einem Implementierungsabbruch kommen, wenn der RFC fachlich abgelehnt wird. Hier geht speziell im Reporting wichtige Information verloren.

Dringende RFCs müssen vorher kategorisiert werden, erst dann werden diese dringend bearbeitet. ITIL sieht vor, dass dringende RFCs gleich als solche eingemeldet werden. Dies kann natürlich kritisch gesehen werden. Würden alle Anfragen dringend sein hätte man nie einen sauberen Prozess. Aber was, wenn ein Notfall auftritt, bei dem, wie in ITIL formuliert, zuerst gearbeitet wird und dann der RFC angelegt wird (siehe Kapitel 5.4.5). Das wäre in der aktuellen Release von Change Management nicht zu realisieren. SAP Change Request Management orientiert sich wohl an ITIL, hat derzeit aber nur wenig Nutzen, wenn man Änderungen durchführen möchte, die nicht in SAP abgebildet werden. Wenn man Hardware tauschen will muss man alles ganz genau beschreiben, eine CMDB ist im SAP Ansatz noch nicht realisiert. Gerade wenn man die Empfehlungen aus der Praxis betrachtet und die enge Verzahnung zwischen z.B. Betriebssystemupdates, Downtimes und notwendigen Softwareaktualisierungen sehen möchte, ist das problematisch.

Nachdem ein Service Desk angeboten wird, müsste man konsequenterweise aus dem SAP Service Desk nur jene Meldungen, die mit SAP Software zu tun haben, an das Change Request Management weitergeben. Die anderen Meldungen wären in einer Fremdsoftware mit angeschlossener CMDB zu pflegen. Dies hätte dann aber wieder Nachteile bei Verflechtungen (wie weiter oben beschrieben).

Andere Produkte im Change Management besitzen die Fähigkeit, direkt aus abweichenden Monitoringdaten (zu viel Serverlast, keine Antwort, etc.) RFCs zu erstellen. Hier muss leider derzeit auch noch programmiert werden, um eine automatische Ticketgenerierung zu erreichen. Ein letzter Punkt wären die Interfaces. Derzeit werden noch keine praktikablen Interfaces dokumentiert, mit denen man SAP Change Request Management fernsteuern könnte (z.B. zur Anlage von RFCs).

7.4 Kosten und erwarteter Nutzen

SAP hält sich über die Einführungskosten von Change Request Management bedeckt. Die Lizenz für den Solution Manager sei in der ERP Lizenz enthalten.

¹⁶⁶Vgl. im Detail bei [SAP07, S.7-19].

7 Analyse und Reflexion

Bei einer Einführung muss speziell in folgenden Bereichen mit einem hohen Aufwand gerechnet werden:

- Einrichten der Systeme: Für alle an den Solution Manager angeschlossenen Systeme müssen die TMS Komponenten an den Solution Manager angeschlossen werden.
- Berechtigungsthematik: Die Berechtigungen des Change Managers und der Entwickler am Solution Manager System müssen eingerichtet und geprüft werden.
- Aufsetzen des Reporting: Change Request Management muss „überprüfbar“ gemacht werden. In Anlehnung an die in Kapitel 5.4.6 genannten Indikatoren ist ein entsprechendes Reporting zu etablieren.
- CMDB Thematik: Wenn keine CMDB verwendet wird muss eine Alternativlösung konzipiert werden, mit welcher es möglich ist, Interdependenzen zwischen einzelnen ERP Programmteilen abzubilden und im Change Fall zu erkennen.

Es sind nun sämtliche Zusammenhänge zwischen den bisher vorgestellten Themen hergestellt worden. Die Begriffe ERP, Software Lifecycle, Governance, IT Governance, ITIL, COBIT und SAP Change Management wurden miteinander in Beziehung gesetzt und grundlegend analysiert. Kosten sowie Nutzen und auch Risiken wurden erläutert.

Im nun folgenden Praxisbeispiel wird das bisher erworbene Wissen nun auf ein praktisches Beispiel übertragen, sowie eine Modellbildung für die Praxis versucht werden.

8 Praxisbeispiel

8.1 Zu den beteiligten Firmen

In seiner Tätigkeit als Berater bei der SAP Österreich GmbH kommen dem Autor immer wieder Kunden mit den unterschiedlichsten Aufgabenstellungen unter. Bei einem dieser Kundentermine kam es zu einer Aufgabenstellung, die aus seiner Sicht noch unzureichend betrachtet wurde. Aus weiteren Gesprächen ergab sich sehr schnell ein Thema für eine Diplomarbeit. Aufgrund des guten Gesprächsklimas wurde eine Zusammenarbeit zwischen TU Wien, der SAP Österreich GmbH und dem Kunden beschlossen.

Diese Zusammenarbeit hat der Arbeit ihren Praxisbezug verliehen und war für die Qualität der Arbeit sehr förderlich.

SAP Österreich GmbH

Die SAP Österreich GmbH wurde im Jahre 1986 als erste Zweigniederlassung der deutschen SAP AG gegründet und beschäftigt heute 450 Mitarbeiter, von denen 100 in einem der größten deutschsprachigen Supportzentren, etwa 100 im Vertrieb und 100 Kollegen in der Beratung tätig sind. Der Rest verteilt sich auf regionale Teams, da der SAP Österreich in der Region Ost- und Zentraleuropa eine starke Rolle zukommt.

Bankdienstleister

Das Unternehmen ist der IT Dienstleister einer österreichischen Großbank, die auch Tochtergesellschaften im Ausland unterhält. Das Unternehmen betreibt eine weitgehend zentrale Infrastruktur am Standort Wien.

8.2 Ausgangssituation

Die Bank betreibt in ihrer Systemlandschaft eine Vielzahl von Softwarelösungen, die sich im Wesentlichen aus den üblichen Büropaketen (Mailserver), Bankensoftware und ERP Systemen zusammensetzt. Als ERP Software wird SAP verwendet. In jüngster Zeit sind auch EAI Tools und ein Data Warehouse im Aufbau.

Das produktive Hauptsystem ist in einer Dreisystemlandschaft angesiedelt. Das Hauptsystem läuft auf einem sehr belastungsstark ausgelegten Cluster. Durch die hohen Anforderungen, die Banken an ihre IT Systeme stellen, existieren weitere Systeme in der Umgebung dieser drei Teile, welche gelegentlich für Kerneltests oder für Zeitreisen¹⁶⁷ genutzt werden.

Mit dem Aufbau der EAI Infrastruktur¹⁶⁸ werden die Abhängigkeiten des ERP Systems zu den anderen Systemen komplexer. Im Moment existiert keine CMDB. Ein Service Desk für SAP Probleme existiert bereits und wird im Rahmen eines SAP Customer Competence Center (CCC) als Abteilung im Unternehmen geführt. Jeder Endbenutzer kann sich an diese Einrichtung wenden.

Dieser Helpdesk ist in vier Supportschichten aufgebaut:

- **Level 1:** Der SAP Helpdesk ist die erste Anlaufstelle und idealerweise auch die beste Stelle, bei der Probleme mit den SAP Anwendungen gelöst oder Anfragen gestellt werden können.
- **Level 2:** Kann der Helpdesk das Problem nicht lösen oder sind weitere Informationen bzw. tiefergehendes Wissen erforderlich, so wird das Problem zu den Fachverantwortlichen für das jeweilige ERP Modul bzw. die jeweilige Anwendung weitergeleitet. Dort wird mit dem nötigen Hintergrundwissen an dem Problem bzw. der Anfrage gearbeitet.
- **Level 3:** Ist die Anfrage mit einer Entwicklung verbunden oder kann der Fachverantwortliche einen Fehler im Programm nicht ausschließen, so ist in der dritten Schicht die Entwicklungsabteilung des CCC gefragt, eine Lösung zu implementieren bzw. zu suchen.
- **Level 4:** Kann sich die Entwicklung den Fehler selbst nicht erklären, oder ist dieser auf SAP Standardcode zurückzuführen, so stellt der SAP Support die 4. und letzte Ebene im Supportkonzept der Bank dar, wobei einer der Entwickler „Eigentümer“ des Problems bleibt und für die Kommunikation mit der SAP zuständig ist.

Bestehende ITIL Annäherungen

Der Helpdesk arbeitet bereits gemäß einem ITIL Service Desk konformen Prozess. Eine Ausnahme besteht insofern, als es einen eigenen SAP Helpdesk gibt, während in der

¹⁶⁷Bei einer Zeitreise wird das System auf einen Zeitpunkt in näherer oder ferner Zukunft geschickt, um zu sehen, dass es auch in Zukunft noch richtig arbeitet.

¹⁶⁸Siehe dazu Kapitel 2.1.

ITIL Empfehlung von einem zentralen Helpdesk für die gesamte IT ausgegangen wird. Im Bereich des Release Managements existiert bereits eine eigenständige Lösung der Bank, welche die zentrale, systemweite Projektpflege ermöglicht. Das ist ein Feature, welches im SAP Standard von Change Request Management übernommen werden kann¹⁶⁹, welches hier aber schon länger existiert und sich direkt am ITIL Originaltext orientiert. Die Administration der Projekte ist so aufgebaut, dass spezifisch ausgewählt werden kann, welche SAP Projekte (Releases oder Kundenprojekte, beides wird verwendet) auf den jeweiligen im Unternehmen vorhandenen Systemen ausgerollt werden sollen. Weiters wurde das KTW so angepasst, dass die Zuordnung eines Transportauftrags zu einem Projekt verpflichtend ist¹⁷⁰. Somit wird jede Änderung automatisch einer Release oder einem Projekt zugebucht. Neben den Releases (2 Mal im Jahr) und den Kundenprojekten gibt es hier für dringende Änderungen eigene Sammelprojekte für die Wartung, die nach Abstimmung auch antizyklisch transportiert werden. Ein Schema dieser Projektzuordnungen ist in Abbildung 8.1 zu erkennen.

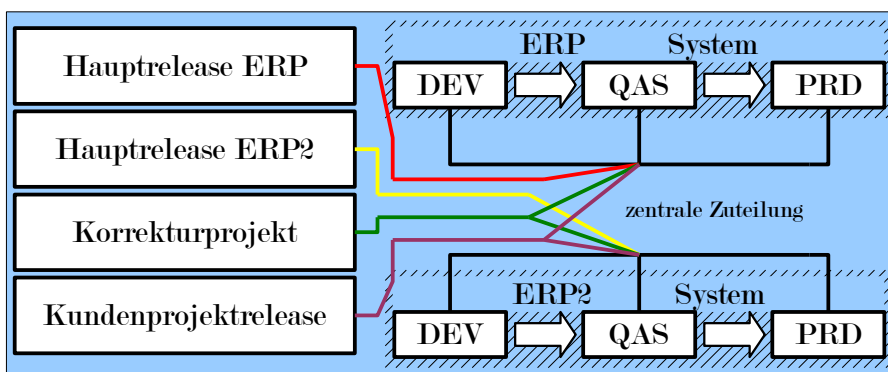


Abbildung 8.1: Die Projektzuordnung als Kundenentwicklung. Ein Beispiel.

Verschiedene Releases für ERP, ERP2, Kunden- und Notfallprojekte werden im Projekt-system zentral angelegt. Mit der Anlage und der Zuordnung wird auf den angeschlossenen Systemen die Projektinformation deployed. Im konkreten Beispiel wird das ERP Release der ERP Landschaft und das ERP2 Release der ERP2 Landschaft zugeordnet. Die Notfall- und Kundenprojekte werden auf beiden Landschaften angelegt. Somit können alle Transportaufträge eindeutig für Systemlandschaften im Sinne des Release Managements identifiziert werden.

Der bisher angefallene Aufwand für diese Anpassungen wird von der Bank auf etwa 1,5-3 Personenmonate geschätzt.

¹⁶⁹Siehe dazu Kapitel 6.3.

¹⁷⁰Diese Information wird in den Transportattributen abgelegt.

Das Unternehmen wird laufend nach SAS (Statement on Auditing Standards) geprüft und zertifiziert. Der Audit nach dem SAS No. 70 Act ist ein amerikanischer, sehr detaillierter und weithin bekannter und akzeptierter Standard, der in Version 2 auch mit SOX konform läuft¹⁷¹. In dieser Arbeit wird SAS nicht behandelt, kann aber als einer ISO 20000 Zertifizierung ähnlich gesehen werden.

8.3 Potentiale

In einer Potentialanalyse wurden Verbesserungsvorschläge der derzeitigen Infrastruktur gesammelt und hinsichtlich des Risikos bewertet. Es ergab sich folgende Reihung:

1. **Errichtung einer zentralen CMDB:** Derzeit werden Zusammenhänge zwischen Applikationen, der Hardware und auch zwischen den einzelnen Modulen und Releases nur über Grafiken und über das Wissen der Modulverantwortlichen getragen. Dies ist aus Sicht des Potentials zur Risikominimierung der Hauptansatzpunkt. Eine groß angelegte CMDB, die nicht nur das Inventar an SAP Programmen, sondern auch die laufende Hardware wie auch die anderen im Haus verwendeten und eingebundenen Applikationen berücksichtigt, erscheint hier besonders praktisch.
2. **Übergreifendes Change Management:** Sekundär erscheint eine Einrichtung eines übergreifenden Change Managements über Hardware und alle Applikationen, inklusive SAP, als hilfreich. Dieses Change Management kann aber nur dann gut funktionieren, wenn es auch auf eine CMDB zurückgreifen kann. Es wäre für eine Senkung des Risikos allerdings nicht unvorteilhaft, alle Changes gemeinsam zu verwalten, da auf diese Weise auch von anderen Abteilungen Changes in der abhängigen Infrastruktur angesehen werden können.
3. **Einrichtung eines zentralen Service Desk für sämtliche IT Inputs:** Also für SAP Anfragen und Probleme als auch solche in den anderen Bereichen der IT Infrastruktur. Es könnte unter Umständen Vorteile bringen, wenn Meldungen zentral erfasst und an die verschiedenen Bereiche weitergeleitet werden können. Was man hier allerdings beachten muss ist, dass in diesem Fall auch die Bankkunden sich wegen Problemen mit dem Online Banking an diesen Service Desk wenden würden. Hier ist irgendeine Art von Abgrenzung nötig, zumindest eine Trennung in internen und externen Service Desk.

¹⁷¹Siehe weiterführend bei [Coo07].

8.4 Kosten- und Nutzenabwägung

Die Implementierung der Änderungen im KTW für eine bessere Unterstützung von Release Management mit den SAP Transportprozessen kostete die Bank etwa 1,5-3 Personenmonate. Für Punkt 1, eine vollständige CMDB würden mehrere Personenjahre Aufwand entstehen, wenn man diesen Ansatz vollständig wählen würde, einfach deshalb, weil sich in jeder Bankfiliale (theoretisch) z.B. Hardware befindet, die auch inventarisiert werden müsste. Der Nutzen wäre in den Filialen eher gering, in den Rechenzentren allerdings nicht unerheblich. Nebenbei bemerkt müsste für die CMDB ein Fremdprodukt angeschafft oder eine Eigenentwicklung in SAP durchgeführt werden, da, wie bereits beschrieben, keine SAP Standardlösung existiert.

In einem Zeitraum von etwa 3-6 Personenmonaten ließe sich Punkt 2, SAP Change Request Management, einführen. Die Erfahrung hat allerdings gezeigt, dass ein Einführen von Change Management ohne CMDB nicht sehr viel verbessern kann¹⁷². Der Vorteil der zentralen Projektpflege geht durch die bestehende Erweiterung im SAP System der Bank nicht verloren, im Gegenteil, die eigene Implementierung ist sogar besser, da die Projekte nur auf betroffene Systeme verteilt werden und nicht auf alle Systeme, die Change Request Management nutzen.

Eine Umsetzung von Punkt 3, die Einrichtung eines zentralen Service Desk, kann nur dann Sinn machen, wenn zumindest der erste Punkt, eine CMDB, umgesetzt wurde, ansonsten sind die Mehrwerte hier begrenzt, da zum Support von SAP Anwendungen ohnehin geschultes Personal vorhanden sein muss. Hier kann es außerdem zum Einziehen eines weiteren Supportlevel kommen (Level 1 = genereller Support, Level 2 = SAP Support, usw.). Das würde nicht zwingend bedeuten, dass hier eine Verringerung des Risikos bewerkstelligt wird. Daher wurde dieser Punkt auch keiner Aufwandsschätzung unterzogen.

8.5 Prototyp

Umsetzung

Nachdem die Umsetzung einer CMDB ein sehr langfristiges Projekt ist (insbesondere wenn es um die Modellierung der gegenseitigen Beeinflussungen von Hard- und Software geht), wurde beschlossen, für die Umsetzung von Punkt 2, dem Change Request Management, einen kleinen Prototypen zu bauen, um erkennen zu können, ob diese Lösung für einen produktiven Einsatz sinnvoll wäre.

¹⁷²Siehe auch Kapitel 5.4.3.

Die Umsetzung des Prototypen hatte zum Ziel, dass eine Dreisystemlandschaft (allesamt Testsysteme) von einem Solution Manager aus mit Transporten beschickt wird. Es wurde eine Testrelease als Projekt im SolMan angelegt und die notwendigen Konfigurationen vorgenommen.

Die fertige Konfiguration wurde getestet, indem eine Kundenentwicklung über einen Change angefordert, die Mitarbeiter und Tester zugeordnet, alles entsprechend protokolliert und der Change nach erfolgreicher Produktivsetzung geschlossen wurde.

Das Ergebnis war, dass durch die zusätzlichen Schritte und die Eintragungen im Hauptdokument eine Steigerung des zeitlichen Aufwandes von etwa 5 Minuten (exklusive Entwicklungs- und Testzeit) auf etwa 9 Minuten stattfand. Dies entspricht fast einer Steigerung um 100% für die administrativen Aufwände einer jeden Änderung. Das Thema Berechtigungen wurde hierbei noch ausgeklammert, ebenso die Tatsache, dass in der Praxis, wenn ohne RFCs gearbeitet wird, oft auch mehrere Changes in einen Transport übernommen werden.

8.6 Ergebnisse

Fazit

Change Request Management erinnert an eine Sammlung von Links in einer Mappe, in welcher immer das zentrale Changedokument vorhanden ist und alle Funktionen dann mit Links verbunden sind. Ohne eine zusätzliche Nutzung einer CMDB oder einem ähnlichen Tool zur Modellierung von Abhängigkeiten von Software Komponenten ist die Risikominimierung äußerst gering, einzig die Zuständigkeiten sind in manchen Konstellationen klarer. Es ist aus Sicht der Evaluierung und auch aus Sicht der Bank im derzeitigen Status nicht sinnvoll, Change Request Management einzuführen.

Die Funktionalität der zentralen Pflege von transportrelevanten Einstellungen und die zentrale Durchführung konnten praktisch überzeugen und es sind keine Fehler aufgetreten.

ROI

Aus diesen Ergebnissen heraus stellt sich die Frage, ab welcher Unternehmensgröße und bei welchen Konstellationen ein Change Management, welches sich an ITIL orientiert, überhaupt Sinn macht.

Gesucht ist gewissermaßen der ITIL-Return on Investment (ROI)¹⁷³:

¹⁷³[Lie06, S.60]

Im Zusammenhang mit ITIL ist es relativ schwierig, einen „Return on Investment“ [...] zu bestimmen. Dies liegt einerseits daran, dass eventuell messbare Verbesserungen oder Sparpotentiale erst über einen längeren Zeitraum sichtbar werden und andererseits eine exakte Bewertung oft nicht möglich ist. Ein Unternehmen tritt zunächst mit erheblichen Kosten in Vorlage (z.B. Ausbildung, Tools) bevor sich die ersten Verbesserungen einstellen.

Die vereinfachte klassische Formel für den Return on Investment wäre¹⁷⁴:

Gewinn/Gesamtkapital

Diese ist hier nicht anwendbar, da sich ja der Gewinn nur bedingt direkt monetär ausdrücken lässt.

Die relevante Kennzahl liegt vielmehr in einer Kombination aus Kostensenkung, Leistungsverbesserung und Erhöhung des Zufriedenheitsgrads der Benutzer einerseits und den für die ITIL Einführung und den ITIL Betrieb eingesetzten Kosten andererseits¹⁷⁵.

Empfehlungen

In der derzeitigen Situation der Bank lautet die aus den Erfahrungen mit dem Prototypen gewonnene Information, dass es zum derzeitigen Zeitpunkt keiner Einführung von Change Request Management bedarf, sondern des Aufsetzens einer Konfigurationsdatenbank, die zumindest die SAP Applikationen mit den jeweiligen abhängigen Hard- und Softwarekonfigurationen verknüpft. Eine direkte Ableitung der betroffenen Konfigurationen bei einer Änderung kann im Alltag der IT Mitarbeiter eine deutliche Verbesserung im Arbeitsablauf bewirken.

Die Risikokomponente der IT Infrastruktur ist bei SAP nicht ganz so hoch, weil die für Banken geschäftskritischen Systeme in diesem Institut nicht in SAP laufen. Ein Ausfall des SAP Systems hat keine unmittelbaren Folgen für das Tagesgeschäft der Bank. Das Risiko kann aber durch die in Abschnitt 8.3 ausgewählten Potentiale weiter gesenkt werden, wenn dies in der (IT) Governance Strategie zum Ziel gemacht wird.

8.7 Bedeutung für andere Unternehmen

Während im gegenständlichen Fall eine klare Entscheidung getroffen wurde, ist es in der Praxis vielleicht nicht so einfach zu entscheiden, wie weit ein Prozessmodell wie

¹⁷⁴Vgl. [Lie06, S.63].

¹⁷⁵[Lie06, S.63]

COBIT oder ITIL in einem Unternehmen implementiert werden soll. Im Zuge der praktischen Realisierung konnte das Projektteam einige Erfahrungen gewinnen, die verschiedene Schlüsse zulassen.

Eine Überzeugung, die gereift ist, war, dass die von Lienemann vorgeschlagene und in Kapitel 5.4.3 beschriebene Roadmap ein sehr guter und vernünftiger Vorschlag ist. In kurzen Worten bedeutet das, dass mit dem Service Desk begonnen, dann über Incident und Problem Management fortgesetzt werden soll. Mit der Einführung von Change und Release Management wird der Fahrplan abgeschlossen. Große Bedeutung kommt der Konfigurationsdatenbank zu.

Ein weiteres Erkenntnis ist, dass ein Change Management selbst erst ab einer gewissen Komplexität der internen Organisation und der Systemlandschaft sinnvoll erscheint. Es sollte nur bei einer vorhandenen und gepflegten CMDB eingesetzt werden, ansonsten ist dies oft ein übertriebener Aufwand.

Bei Teams bis zu 5 Personen im IT ist ein Change Management eher nicht erforderlich. Hier ist das Risiko gering, dass eine Abstimmung nicht erfolgen konnte. Was hierbei noch berücksichtigt werden muss, ist die Anzahl externer Berater.

Bei mittleren Unternehmen bis etwa 20 Mitarbeitern im IT ist - der Einschätzung des Projektteams nach - die Einführung von Change Management in der Regel ein Schritt der Überadministration. Hier macht ein Service Desk Sinn, die Kommunikation kann aber bei einer solchen Größenordnung noch direkt erfolgen.

Großen Unternehmen und Unternehmen, die über gesetzliche Vorschriften einem Regulierungsgesetz unterworfen sind, wird dringend angeraten, ihre Prozesse an ITIL anzunähern. Es sollte nicht unbedingt Wert darauf gelegt werden, diese immer exakt wie in der ITIL Spezifikation erläutert umzusetzen, aber es ist besonders wichtig, dass die einzelnen Prozesse genau dokumentiert sind und sodann genau nach der entstandenen Spezifikation gelebt werden.

8.8 Ableitung eines Modells

Die Ergebnisse und Überlegungen zu Change Management bei betrieblicher Standardsoftware führen zu einer Empfehlung, die von folgenden Fakten ausgeht:

- Die Unternehmen setzen eine ERP Standardsoftware für ihre täglichen Geschäftsprozesse ein. Eine Arbeitsunterbrechung ist somit kritisch.
- Die Anzahl der IT Mitarbeiter ist eine Kennzahl, die jene Mitarbeiter umfasst, die direkt der IT zugeteilt sind, also organisatorisch unter dem CIO arbeiten. Sie

befassen sich mit der internen IT.

Kleine Unternehmen

Bei kleinen Unternehmen handelt es sich beim vorliegenden Modell um Unternehmen mit einer IT Abteilung von bis zu 5 Mitarbeitern.

Empfehlungen:

- Etablierung einer Telefonhotline, die nach dem ITIL Service Desk Standard arbeitet.
- Abbildung der Abhängigkeiten von Hard- und Software innerhalb der IT Landschaft in Konfigurationsdatenbanken oder einem entsprechenden grafischen Tool.
- SAP Changes werden mit den TMS Tools durchgeführt, eventuell unter Benutzung des SAP Projektsystems als Releasedatenbank.
- Das Monitoring ist ein manueller Vorgang, welcher von den Administratoren regelmäßig durchgeführt werden muss.

Mittlere Unternehmen

Ein mittleres Unternehmen wird hier definiert als ein Unternehmen mit einer IT Abteilung von bis zu 20 Mitarbeitern. Hier gelten folgende Empfehlungen:

- Etablierung einer Telefonhotline, die nach dem ITIL Service Desk Standard arbeitet.
- Die Einführung zumindest dreier Support Level ist erforderlich.
- Abbildung der Abhängigkeiten von Hard- und Software innerhalb der IT Landschaft in einer zentralen Konfigurationsdatenbank oder einem entsprechenden grafischen Tool.
- Eventuell ist es sinnvoll, einen Prozess gemäß ITIL Problem Management zu etablieren.
- SAP Changes werden mit den TMS Tools durchgeführt oder über eine Erweiterung im SAP Standard wie die Bank im gegenständlichen Praxisbeispiel (Kapitel 8.2). SAP Change Request Management wird nicht eingesetzt.
- SAP Releases werden zentral über den Solution Manager in das SAP Projektsystems als Releasedatenbank übernommen.

- Das Monitoring wird zentral durchgeführt und ist ein manueller oder automatischer Vorgang, welcher von den Administratoren überwacht wird.

Im Unterschied zum kleinen Unternehmen setzt die IT Abteilung des mittleren Unternehmens auf die Zentralisierung von Services. Durch die Größe von bis zu 20 Mitarbeitern ist die IT Abteilung noch überschaubar und es ist für die Risikominimierung noch nicht erforderlich, die gesamte Bandbreite an ITIL Prozessmodellen einzuführen.

Große Unternehmen

Ein großes Unternehmen definiert sich als ein Unternehmen mit einer IT Abteilung von mehr als 20 Mitarbeitern. Das Unternehmen ist als Kapitalgesellschaft begründet und ist seinen Investoren verpflichtet. Es agiert nicht international; das bedeutet, es gibt bei mehreren Standorten nur einen Standort mit einer IT Abteilung. Die Empfehlungen für Unternehmen dieser Größe lauten:

- Etablierung einer Telefonhotline, die nach dem ITIL Service Desk Standard arbeitet.
- Die Einführung von vier Support Levels erscheint sinnvoll.
- Die Abhängigkeiten von Hard- und Software innerhalb der IT Landschaft werden in einer zentralen Konfigurationsdatenbank administriert.
- Es existiert ein Corporate Governance Kodex sowie eine IT Governance Strategie, die genau die Erwartungen von Investoren, Management und den übrigen Stakeholdern (Gesetze, etc.) an die IT Abteilung wiedergibt.
- Es werden die ITIL Prozesse Incident Management, Problem Management, Change Management und Release Management umgesetzt¹⁷⁶.
- SAP Changes werden mit SAP Change Request Management durchgeführt oder mit einer gleichwertigen externen Lösung.
- Es gibt im Bereich des Change Management einen definierten Change Manager inklusive Change Management Team sowie ein CAB, ECAB und ein ITEC.
- SAP Releases werden zentral über den Solution Manager in das SAP Projektsystem als Releasedatenbank übernommen.

¹⁷⁶Diese Liste orientiert sich an den in dieser Arbeit vorgestellten Prozessen und ist keinesfalls vollständig!

- Das Monitoring wird zentral durchgeführt und vollständig automatisiert. Dies bedeutet, dass bei Störungen in den Schwellwerten automatisch RFCs im Change Management angelegt werden.

Große Unternehmen müssen sich noch nicht mit den globalen Anforderungen eines Konzerns auseinandersetzen, haben aber durch gesetzliche Regulierungen und dem Umstand, dass sie teilweise außenstehenden Investoren verpflichtet sind, einen hohen Druck, Risiko zu minimieren. Hier ist es erforderlich, eine gute (IT) Governance zu leben und sich der vorhandenen Standardprozess-Frameworks wie COBIT oder ITIL zu bedienen.

Konzernunternehmungen

Für Konzernunternehmen gelten prinzipiell die gleichen Regeln wie bei großen Unternehmen. Bei der Umsetzung des Change Managements sollte Bedacht genommen werden, ob eine vollständig zentrale Infrastruktur für Changes sinnvoll ist. Es ist anzudenken, ob nicht mehrere Change Manager, mehrere CABs und auch regionalisierte ITECs zu definieren sind, allesamt koordiniert von einer globalen Change Management Gruppe und einem globalen CAB. Über diese Globalisierung und Regionalisierung kann in der Regeln anhand der Systemlandschaft entschieden werden. Existieren eigene lokalisierte Länderversionen von konzernweit angepasster Standardsoftware, so sind regionalisierte Strukturen zu bevorzugen.

Mit der nunmehr erfolgten praktischen Umsetzung von Change Management und der Entwicklung eines entsprechenden Modells wurde der Versuch unternommen, SAP Change Request Management auf seine Einsatztauglichkeit zu prüfen. Die gewonnenen Ergebnisse mögen in den verschiedenen Unternehmen von Nutzen sein.

Im nächsten Kapitel werden die Arbeit, die Ergebnisse und der praktische Teil nochmals zusammengefasst. Im Abschnitt Ausblick werden Ansätze zur weiteren Bearbeitung des Themas geliefert. Die Arbeit wird mit einer Bemerkung des Autors abgeschlossen.

9 Zusammenfassung und Ausblick

9.1 Zusammenfassung

Die vorliegende Arbeit wurde mit dem Ziel verfasst, im Bereich von Change Management in der Informationstechnologie einen Überblick und eine konkrete Anwendung in der Praxis aufzuzeigen. Wie komplex, weitläufig und interdisziplinär dieses Thema ist, konnte anhand der vielen Details aus den Bereichen der Informatik-, der Rechts- und der Sozialwissenschaften in diesem Umfang nur unvollständig wiedergegeben werden. Bereits bei den Grundlagen zu ERP Systemen wie auch in den fortlaufenden Kapiteln mussten immer wieder Abgrenzungen getroffen werden, damit die Fülle an Informationen auch auf das konkrete Ziel der Arbeit hin heruntergebrochen wurde. Architekturelle Grundlagen wie auch z.B. der Begriff der Systemlandschaft wurden allerdings genau aufbereitet. Diese sind im Zuge der Arbeit immer wieder in Erscheinung getreten. Weiter fortgehend in den Ausführungen wurde der Begriff des Software Lifecycle allgemein für ERP Systeme definiert und durch Details und Beispiele greifbar gemacht. Anpassungstechniken von Standardsoftware wurden hierbei ebenso behandelt wie die Auswirkungen und Wechselwirkungen mit Softwareprojekten. Änderungen an der ERP Software im Unternehmen bergen immer ein gewisses Risiko. Es wurde festgestellt, dass eines der Ziele eines jeden Unternehmens sowie seiner Investoren die Minimierung von Risiken ist. Zu diesem Zweck existieren verschiedene Richtlinien und Gesetze, die im Rahmen einer Governance Strategie implementiert werden. Im Abschnitt zu Governance und IT Governance sind auch die Umwelten von Unternehmen berücksichtigt worden. Die Stakeholder, die die Unternehmen beeinflussen und mitunter ganz unterschiedliche Interessen verfolgen, wurden in ihren Absichten beleuchtet und die verschiedenen Ausprägungen und Regulative rund um Governance und IT Governance wurden erläutert. Dies alles diente als Grundlage des Verständnisses von Governance Framework. Richtwerke für die Implementierung einer guten IT Governance sind COBIT, aber auch das IT Infrastructure Library (ITIL) Framework, welches im Detail behandelt wurde. Die IT Infrastructure Library des britischen Office of Government Commerce wurde sowohl historisch betrachtet als auch systematisch beschrieben. Das generische Prozessmodell

und die Ausführungen zu Change Management haben die Vorarbeit für die Behandlung der Prozessindikatoren geleistet. Mit diesen kann der Erfolg oder Misserfolg eines Prozesses gemessen werden kann. In weiterer Folge ging die Arbeit dann konkret auf einen Anbieter und das Lösungsportfolio ein, indem Erläuterungen zum SAP Transportwesen vorgebracht wurden. SAP Transportmanagement und Change Request Management wurden vom Prozess her beschrieben und analysiert. Ein Ausblick gab entsprechende Perspektiven für die Zukunft. Letztlich wurde all dies noch in den Kontext von internationalen Systemlandschaften gestellt, wo durch die Verteilung der Komponenten wieder neue Risikoquellen zu Tage treten könnten.

Mit einem ausreichenden Basiswissen ausgestattet wurden die vorgestellten Teile analysiert und Hypothesen für eine Implementierung aufgestellt. Die Betrachtung der Kosten/Nutzen Aspekte sowie der Risiken erfuhr eine besondere Berücksichtigung. Sind doch der Kosten/Nutzen sowie der Risikoaspekt sehr wichtige Kriterien bei der Beurteilung von IT Governance Strategien wie auch bei der Freigabe der für deren Einführung benötigten Mittel.

Das Praxisbeispiel zum Change Request Management mit SAP wurde in Kooperation mit namhaften Firmen durchgeführt und liefert ein klares Bild über die derzeitigen Entwicklungstendenzen im Change Bereich bei SAP. Es wurden Empfehlungen für das gezeigte Unternehmen ausgearbeitet, die dort auch vorgelegt und diskutiert wurden. Die Inputs wurden freundlich angenommen. Sie zeigten unter anderem, dass die eigenen Überlegungen des Unternehmens korrekt waren, aber auch, dass die Inputs, die aus der Durchführung des Prototypings gewonnen werden konnten, wichtig für die weitere Planung sind. Aus den gewonnenen Erfahrungen sowie den Diskussionen erschließt sich ein Modell, welches eine Einordnung von Firmen in Strategien für IT Governance anhand des ITIL Frameworks ermöglicht. Diese Modellbildung ist einer der wichtigsten Aspekte der Arbeit. Ein weiteres wichtiges Erkenntnis ist wohl, dass die IT und auch die dort gelebten Prozesse Chefsache sind also, dass das Unternehmensmanagement die Ziele der IT und somit auch der ERP Strategie festzulegen hat. Das Selbstverständnis bzw. die Vorgabe, dass „IT einfach zu funktionieren hat“, scheint hier wenig zeitgemäß und bedarf einer differenzierteren Betrachtung: IT kann ganz einfach funktionieren, wenn diese richtig gelenkt wird. Möglicherweise selbstverständliches muss dokumentiert werden, um sicherzustellen, dass die IT dies auch in Zukunft tut.

Die von SAP derzeit angebotenen Lösungen im Change und Release Management sind gute Ansätze, wenn auch nicht in allen Punkten vollständig ITIL konform. Dass allerdings zusätzlich zu den SAP Tools eine eigene, externe Konfigurationsdatenbank benötigt wird, um das komplette Spektrum der ITIL Anforderungen abzudecken, hat enttäuscht.

Der Funktionsumfang der getesteten Lösung kann als vollständig empfunden werden, während die Bedienung zu Beginn ein gewisses Geschick erfordert. Der Gesamteindruck ist somit positiv, die Lösung selbst deckt aber nicht alle Bereiche der Anforderungen ab. Auch die undokumentierte Schnittstellenthematik zwischen Change Request Management und möglichen externen Konfigurationsdatenbanken war entgegen den Erwartungen, die zu Beginn des Prototyps bestanden haben.

IT Verantwortliche in anderen Unternehmen können aus den Ausführungen und Entscheidungshilfen in dieser Arbeit wertvolle Informationen für das jeweilige eigene Unternehmen gewinnen und den Governance-Fahrplan im jeweiligen Unternehmen entsprechend gestalten. Im Bereich der ERP Systeme fiel auf, dass vielfach die IT und die ERP Welt¹⁷⁷ in sehr vielen Unternehmen getrennt betrachtet werden. Diese Sichtweise ist im Sinne der Risikominimierung zu überdenken. Fachlich haben ein Netzwerkadministrator und ein ERP Spezialist vielleicht nicht so viele Dinge gemeinsam, aber in der Realität sind sich die beiden näher als auf den ersten Blick auffällt.

9.2 Ausblick

In den aktuellen Zeiten mit einer rezessiven Wirtschaft in Nordamerika und einer befürchteten Deflation in Europa werden Unternehmen zu vorsichtigen Strategien und einer vom Sparen geprägten Kostenpolitik genötigt werden.

Neben der Minimierung von unnötigen Ausgaben gilt die Aufmerksamkeit in den nächsten Jahren aber auch einer weiteren Reduktion des Risikos in den internen Bereichen. Es kann davon ausgegangen werden, dass Governance Themen auch weiterhin einen hohen Stellenwert in Unternehmen genießen werden. Dies zum einen, weil gesetzliche Regulative diese Themen weiter fordern werden, zum anderen wird aber auch bei den Investoren die Nachfrage nach effektiven Kontrollmechanismen ansteigen.

Die in dieser Arbeit durchgeführten Betrachtungen zum Return on Investment von Prozessmodellen wie ITIL Change Management sind bei Weitem nicht vollständig. Auch ist ohne eine konkrete Studie zu diesem Thema das gebildete Modell nicht verifizierbar. In einer weiterführenden wissenschaftlichen Studie zu diesem Thema wäre es interessant, den tatsächlichen ROI von ITIL zu berechnen, und das hier gebildete Modell mit Zahlen zu untermauern. Auch externe Effekte, wie eingebundene Dienstleister und Berater, die mitunter das Risiko enorm beeinflussen, wären geeignet abzubilden. Eine gute Grundlage für weitere Forschungen auf diesem Gebiet wäre die Arbeit von [Sta07], welche bereits eine erste Risikobezifferung im IT Governance Bereich versucht. Dieses Risiko nun auf

¹⁷⁷In diesem Beispiel ist die SAP Welt gemeint.

ITIL herunterzuberechnen und die zusätzlichen positiven Effekte aufzuzeigen ist ein sehr spannendes weiterführendes Thema.

Wie aktuell das Thema ROI von ITIL tatsächlich ist, ergibt ein aktuell durchgeführter Aufruf der Internet-Suchmaschine www.google.com mit den Suchbegriffen ITIL und ROI . Die erste Ergebnisseite besteht aus fast gleichlautenden Links in den verschiedenen Ländern. Eingeleitet wird das erste Suchergebnis mit den Worten: „ITIL's ROI hard to measure“, zu Deutsch: Der ROI von ITIL ist schwer zu messen. Die Aussage kommt von einer Studie unter IT Managern. Es kommt wohl nicht von ungefähr, dass diese Aussage das erste Ergebnis in Google darstellt.

9.3 Schlussbemerkung des Autors

Auch wenn ich zu Beginn der Arbeit am Thema Change Management einen völlig anderen Fokus auf das Thema gelegt hatte, so denke ich, dass mir die vorliegende Arbeit sehr viele und auch neue Einsichten zur Wichtigkeit des Themas Corporate Governance in Unternehmen gebracht hat. Die Autoren der zahlreichen verwendeten Quellen haben aber auch zwischen den Zeilen ein Bild in mir hinterlassen, welches ich an dieser Stelle der Arbeit noch auszudrücken versuchen möchte:

Ich denke, es wird niemals ein Unternehmen geben, in welches man deswegen investieren kann und soll, weil das ohne jedes Risiko ist. Gesetze wie der Sarbanes Oxley Act oder EuroSOX greifen zwar tief in Unternehmen ein und fordern Grundsätze der ordentlichen Unternehmensführung anhand von festgelegten Kriterien; dies vor allem in der IT. Jedoch wird ein Investment in ein Unternehmen den Anleger - mit oder ohne SOX - nicht von der Verpflichtung entheben, sich die vom Unternehmen vorgelegten Prospekte und Bilanzen genauestens anzusehen. Auf diesen basierend soll er dann seine Investmententscheidung treffen. Volkswirtschaftlich haben allerdings alle ein Interesse daran, dass Unternehmen korrekt wirtschaften, weil dies die Insolvenzsicherung und die korrekte Steuertaxierung erleichtert.

Literaturverzeichnis

- [Blo08] BLOKDIJK, G.: *The Change Management Toolkit - The Missing IT Change Management Planning, Process, Theory and Tools Guide - ITIL Compliant*. Emereo Pty Ltd, 2008
- [Bon05] BON, J. van (Hrsg.): *IT Service Management basierend auf ITIL*. Van Haren Publishing, 2005
- [Bra07] BRANDNER, G. C.: *The Challenges Facing IT Governance with Regard to Compliance-Expectations*, Johannes Kepler Universität Linz, Diplomarbeit, 2007
- [Coo07] COOLIDGE, Scott: *About SAS 70*. Scott Coolidge, 2000-2007 <http://www.sas70.com/about.htm>; 08.11.2008
- [DeM97] DEMARCO, Tom: *Warum ist Software so teuer? ... und andere Rätsel des Informationszeitalters*. Carl Hanser Verlag München Wien, 1997
- [Div03] DIV.: *Board Briefing on IT Governance, 2nd edition*. IT Governance Institute, 2003 http://www.itgi.org/Template_ITGI.cfm?Section=ITGI&Template=/ContentManagement/ContentDisplay.cfm&ContentFileID=4667; 03.09.2008
- [EC08] ECONET CONSULTING, n.n.: Hilfe bei der EURO-SOX-Einführung. In: *it-Management* 07/08 (2008), S. 28
- [Fan07] FANNING, P. (Hrsg.): *ITIL Service Transition*. Office of Government Commerce, London, England, 2007
- [FHG08] FILL H.-G., Tjoa A M.: Governance in der IT. In: *Magazin der Oesterreichischen Computer Gesellschaft* 1 (2008), S. 10 ff.
- [Gal05] GALL, M.: *E-Business und die daraus resultierenden Anforderungen an ERP Systeme*, TU Wien, Diplomarbeit, 2005

Literaturverzeichnis

- [Hai03] HAINBUCHER, M.: *Corporate Governance in Österreich - Quo Vadis?*, Johannes Kepler Universität Linz, Diplomarbeit, 2003
- [Hau07] HAUSEGGER, H.: *Auswirkungen von Compliance Änderungen auf IT Governance im Hinblick auf das interne Kontrollsystem*, Johannes Kepler Universität Linz, Diplomarbeit, 2007
- [Hüb02] HÜBNER, W. (Hrsg.): *SAP Guide Systemadministration*. Galileo Press GmbH, Bonn, 2002
- [Hes02] HESCHL, K.: *IT Governance*, Johannes Kepler Universität Linz, Diplomarbeit, 2002
- [Hil05] HILB, M.: *New corporate governance: successful board management tools*. Springer Verlag, 2005
- [Hum04] HUMPLIK, F.: *Eine Mustersprache für ERP-Systeme*, TU Wien, Diplomarbeit, 2004
- [Jos03] JOST, C.: IT Controlling - am Beispiel des Software-Projektcontrollings. In: *Zeitschrift für Unternehmensentwicklung und Industrial Engineering* 6 (2003), S. 246 ff.
- [KB04] KITTLAUS B., Rau C. und Schulz J.: *Software-Produkt-Management*. Springer Verlag, 2004
- [KK05] KESSLER K., Tillert P. und Dobrikov P.: *Java Programmierung mit dem SAP Web Application Server*. Galileo Press GmbH, Bonn, 2005
- [Kru00] KRUCHTEN, Philippe: *The Rational Unified Process - An Introduction Second Edition*. Addison - Wesley, 2000
- [Lie06] LIENEMANN, G.: *ITIL - Change Management*. Heise Zeitschriften Verlag, Hannover, 2006
- [Mel04] MELLIS, Werner: *Projektmanagement der SW-Entwicklung. Eine umfassende und fundierte Einführung*. Friedr. Vieweg & Sohn Verlag, Wiesbaden, 2004
- [MFMS00] MC FARLAND METZGER S., Röhrs S.: *SAP R/3 Änderungs- und Transportmanagement*. 1. Auflage. Bonn : Galileo Press GmbH, Bonn, 2000
- [Mil89] MILLION, J.: *Integration von Projektmanagement und dem Software-Entwicklungsprozeß*, TU Wien, Diplomarbeit, 1989

Literaturverzeichnis

- [Mit04] MITTERRUTZNER, G.: *Die Stellung des Aufsichtsrates - Corporate Governance in Österreich und Deutschland*, Johannes Kepler Universität Linz, Diplomarbeit, 2004
- [Mül08] MÜLLER, K.: IT Compliance - Eine Verpflichtung oder auch ein Vorteil. In: *Magazin der Oesterreichischen Computer Gesellschaft* 1 (2008), S. 18 ff.
- [Ple05] PLECR, D.: *Der Österreichische Corporate Governance Kodex*, Johannes Kepler Universität Linz, Diplomarbeit, 2005
- [Pop07] POPP, T.: *IT-Governance-Modelle*, TU Wien, Diplomarbeit, 2007
- [RC07] ROPARDO CONSULTING, n.n.: SO Einführung bei Toyota Deutschland. In: *itfokus* September/Oktober (2007), S. 10
- [SAP03] SAP AG (Hrsg.): *Schulungsunterlage Kurs ADM100 SAP Web AS Administration 1*. 2005/Q2. Walldorf DE: SAP AG, 2003
- [SAP04] SAP AG (Hrsg.): *Schulungsunterlage ADM325 Software - Logistik*. 2004/Q4. Walldorf DE: SAP AG, 2004
- [SAP07] SAP AG (Hrsg.): *Schulungsunterlage E2E200 - Change Control Management*. 2007/Q2. Walldorf DE: SAP AG, 2007
- [SAW00] SCHEER A.-W., Wendt S.: *Dem Wandel voraus*. Galileo Press GmbH, Bonn, 2000
- [SL08] SCHWARZE L., Fabritius und D. Tranwnitschek D.: Risikointelligente Steuerung der IT. In: *itManagement* 06 (2008), S. 27 ff.
- [Spi07] SPIEGL, Klaus: *Projektmanagement Life - Best Practices und Significant Events im Software Projektmanagement*, TU Wien, Diplomarbeit, 2007
- [Sta07] STALLINGER, M.: *IT-Governance im Kontext Risikomanagement*, Johannes Kepler Universität Linz, Diss., 2007
- [TAM05] TJOA A M., Karagiannis D.: IT Governance - Definition, Standards & Zertifizierung. In: *Magazin der Oesterreichischen Computer Gesellschaft* 4 (2005), S. 18 ff.
- [ZW01] ZUSER W., Grechenig T. und Köhle M. Biff S. S. Biff S.: *Software Engineering mit UML und dem Unified Process*. Pearson Education Deutschland GmbH, 2001