

Prof. Dr. Bettina Schwarzer

# Quick Guide to Migration

Entscheidungshilfe und Wegleitung bei IT-Migrations-Projekten

**Prof. Dr. Bettina Schwarzer**

# **Quick Guide to Migration**

**Entscheidungshilfe und Wegleitung bei IT-Migrations-Projekten**

Herausgeber: Business Solution Group AG, Zürich

Druck: Spillmann Druck AG, Zürich

Auflage: 500 Ex.

Nachdruck und Vervielfältigung nach Absprache.

© Business Solution Group AG, November 2007

## **Wer steckt hinter der ITM?**

Die ITM GmbH wurde 1995 von Herrn Prof. Dr. Krcmar (TU München) als unabhängiges Beratungsunternehmen gegründet, um Know How aus der Wissenschaft in die Praxis zu transferieren. Noch heute haben wir enge Kontakte zu Hochschulen um diesen Transfer aufrecht zu erhalten.

Unsere Stärke liegt in der Verbindung innovativer und praktikabler Ideen mit wissenschaftlichen Ansätzen des Informations- und Technologiemanagements. Mit grossem Engagement richten wir unsere erprobten Vorgehensweisen immer wieder neu an den Anforderungen unserer Kunden aus, um für diese State-of-the-Art Lösungen zu erarbeiten.

## **Prof. Dr. Bettina Schwarzer**

Studium der Wirtschaftswissenschaften in Hohenheim und an der Anderson Graduate School of Management, Los Angeles. 1991-1997 Wissenschaftliche Mitarbeiterin am Lehrstuhl für Wirtschaftsinformatik an der Universität Hohenheim. 1996-2003 Geschäftsführende Gesellschafterin der ITM und Projektleitung im Bereich Strategie- und Umsetzungsberatung mit den Schwerpunkten Entwicklung von IT-Strategien, Prozessreorganisation, IT-Controlling.

Seit 2003 Professorin mit dem Lehrgebiet "Betriebliche Informationssysteme" und Studiendekanin des Studiengangs E-Services an der Hochschule für Medien, Stuttgart.



**business|solution|group**  
WE CREATE VALUE.

## **Business Solution Group - we create value**

Die Business Solution Group ist eine eigenständige, finanziell unabhängige, schweizerische Firmengruppe mit Sitz in Zürich und beschäftigt über 180 Informatik- und Business-Spezialisten. Wir gehören zu den Top-3 Schweizer Lösungsanbietern.

## **Software-Migration – Viel mehr als ein technisches Thema**

*«Der vorliegende „Quick Guide to Migration“ erlaubt einen schnellen Einstieg in die Thematik. Ohne auf technische Details einzugehen, werden betriebswirtschaftliche Zusammenhänge aufgezeigt, Vorgehensweisen erläutert und kritische Erfolgsfaktoren herausgearbeitet.»*

*Prof. Dr. Helmut Krcmar  
TU München*

**Business Solution Group AS**

Buckhauserstrasse 24

8048 Zürich

Telefon: +41 (0)44 497 77 77

Fax: +41 (0)44 497 77 78

[www.bsgroup.ch](http://www.bsgroup.ch)

ISBN 3-905712-04-0

CHF 30.00    EUR 18.00

## Vorwort

In 2006 zitierte Hunter (Group Vice President des Analysten Gartner) ein Mitglied des Gartner Forschungsteams:

„A legacy system is a hindrance that fills a business need – so you can't just get rid of it.”  
(Hunter, 2006, 5).

Gut ein Jahr später lesen sich die Überschriften in den managementorientierten IT-Zeitschriften jedoch wie folgt: „Kernbanksysteme kein Tabu mehr“ (Kufer / Bereszewski, 2007), „Flexible Banken mit SOA“ (Wolff, 2007a), „Banking von der Stange“ (Halbich / Fritsch, 2007). In anbetracht des harten Wettbewerbsumfeldes machen viele Finanzdienstleister inzwischen auch vor dem „Allerheiligsten“ nicht mehr halt sondern tauschen historisch gewachsene Individuallösungen gegen Standardsoftwaresysteme.

Dieser Austausch kann aber nicht „einfach so“ erfolgen. Die Migration eines komplexen Systems auf eine neue Plattform ist immer eine große Herausforderung für das Unternehmen, die mit erheblichen Risiken verbunden ist.

Eine Migration ist nur bei oberflächlicher Betrachtung ein IT-Projekt – vielmehr kann es eine strategische Entscheidung für eine flexible und zukunftsorientierte Ausrichtung des gesamten Unternehmens sein, wenn entsprechend damit umgegangen wird. Daher sollte das Thema nicht ausschließlich den IT-Spezialisten übertragen werden, sondern das Management sollte sich mit der Thematik beschäftigen, um die mit einer Migration verbundenen Potentiale für das Gesamtunternehmen zu identifizieren und entsprechende Ausschöpfungsmöglichkeiten zu schaffen.

Der vorliegende “Quick Guide to Migration” erlaubt einen schnellen Einstieg in die Thematik. Ohne auf technische Details einzugehen, werden betriebswirtschaftliche Zusammenhänge aufgezeigt, Vorgehensweisen erläutert und kritische Erfolgsfaktoren herausgearbeitet. Mit dieser Vorbereitung sollte jeder Manager in der Lage sein, bei Diskussionen über ein Migrationsprojekt die „richtigen“ Fragen zu stellen, um aus einem Migrationsprojekt mehr als nur ein IT-Projekt zu machen.

Prof. Dr. Helmut Krcmar  
(TU München)

# Inhalt

Vorwort

Inhalt

Verzeichnis der Abbildungen

Verzeichnis der Tabellen

<b><u>1</u></b>	<b><u>WARUM IST MIGRATION EIN THEMA?</u></b>	<b><u>1</u></b>
1.1	AUSGANGSSITUATION IN DEN UNTERNEHMEN	1
1.2	MIGRATION ALS LÖSUNGSANSATZ	4
<b><u>2</u></b>	<b><u>WAS VERSTEHT MAN UNTER „MIGRATION“?</u></b>	<b><u>7</u></b>
2.1	MIGRATION IM LEBENSZYKLUS VON SYSTEMEN	7
2.2	MIGRATIONSTYPEN	15
<b><u>3</u></b>	<b><u>SOFTWARE-MIGRATION – NOTFALL ODER STRATEGISCHE NEUORIENTIERUNG ?</u></b>	<b><u>21</u></b>
3.1	AUSLÖSER EINER MIGRATION	21
3.2	POTENTIALE EINER MIGRATION	35
3.3	WAS UND WANN SOLLTE MIGRIERT WERDEN?	38
3.4	KOSTEN DER MIGRATION	43
3.5	RISIKEN EINER MIGRATION	50
<b><u>4</u></b>	<b><u>MIGRATION – MEHR ALS NUR EIN TECHNISCHES THEMA</u></b>	<b><u>55</u></b>
4.1	TECHNISCHE RAHMENBEDINGUNGEN	55
4.2	ORGANISATORISCHE RAHMENBEDINGUNGEN	58



<u>5</u>	<u>HERAUSFORDERUNGEN DER MIGRATION .....</u>	<u>61</u>
5.1	PROBLEMBEREICHE EINER MIGRATION.....	61
5.2	KRITISCHE ERFOLGSFAKTOREN FÜR MIGRATIONSPROJEKTE 64	
<u>6</u>	<u>MIGRATIONSSTRATEGIEN.....</u>	<u>71</u>
6.1	INKREMENTELLE MIGRATION.....	71
6.2	BIG BANG UMSTELLUNG.....	77
6.3	VOR- UND NACHTEILE DER MIGRATIONSSTRATEGIEN.....	78
6.4	ANFORDERUNGEN AN EINE MIGRATION .....	84
<u>7</u>	<u>VORGEHENSMODELL FÜR MIGRATIONSPROJEKTE.</u>	<u>87</u>
7.1	ÜBERBLICK .....	87
7.2	PHASE 1: VORBEREITUNG .....	91
7.3	PHASE 2: MIGRATIONS KONZEPT .....	94
7.4	PHASE 3: UMSETZUNG .....	97
7.5	PHASE 4: TEST .....	98
7.6	PHASE 5: EINFÜHRUNG UND KONSOLIDIERUNG .....	99
<u>8</u>	<u>VORGEHEN BEI DER DATEN-MIGRATION.....</u>	<u>101</u>
8.1	BEDEUTUNG DER DATENMIGRATION .....	101
8.2	ABLAUF EINER DATEN-MIGRATION .....	104
8.3	PROBLEMBEREICHE DER DATEN-MIGRATION.....	109
<u>9</u>	<u>FALLSTUDIE.....</u>	<u>111</u>
9.1	DIE AUSGANGSSITUATION .....	111
9.2	DIE PROJEKTORGANISATION.....	113
9.3	VORGEHENSWEISE.....	115
9.4	KRITISCHE ERFOLGSFAKTOREN.....	126
<u>10</u>	<u>LITERATURVERZEICHNIS.....</u>	<u>129</u>

## Abbildungsverzeichnis

Abb. 1: Aufbau des Quick Guide .....	6
Abb. 2: Ansätze zur Erneuerung von Systemen. ....	10
Abb. 3: Kapselung. ....	11
Abb. 4: Renovierung. ....	12
Abb. 5: Partielle Migration .....	14
Abb. 6: Vollmigration. ....	15
Abb. 7: Sichten auf IT-Landschaft von Unternehmen .	22
Abb. 8: Bandbreite der Auslöser .....	24
Abb. 9: Entwicklungsjahr der Kernapplikation. ....	26
Abb. 10: Finanzdienstleister und ihre IT-Probleme. ....	32
Abb. 11: Aufteilung der IT-Budgets. ....	37
Abb. 12: Legacy Portofolio .....	40
Abb. 13: Schichten voll zerlegbarer Systeme .....	56
Abb. 14: Inkrementelle Migration. ....	73
Abb. 15: Big Bang Umstellung. ....	78
Abb. 16: Phasen in Migrationsprojekten. ....	90
Abb. 17: Ablauf Daten-Migration. ....	107
Abb. 18: Projektorganisation. ....	113
Abb. 19: Vorgehen zur Umsetzung. ....	119

## Tabellenverzeichnis

Tab. 1: Teilbereiche der Migration .....	18
Tab. 2: Personalaufwand für ein Migrationsprojekt. ....	49
Tab. 3: Typische Projektrisiken. ....	51
Tab. 4: Durchführung von Migrationsprojekten .....	89



# 1 Warum ist Migration ein Thema?

## 1.1 Ausgangssituation in den Unternehmen

Heutige IT-Landschaften in den Unternehmen sind oftmals ein komplexes Gefüge aus PCs, Midrange-Systemen und Mainframes, die mit einer Vielzahl von Betriebssystemen, Anwendungen, Datenbanken, Programmiersprachen und systemnaher Software arbeiten.

*Gewachsene  
IT-Land-  
schaft*

Obwohl diese über die Jahre gewachsenen Landschaften häufig nicht mehr dem technischen State-of-the-Art entsprechen, halten viele Unternehmen dennoch daran fest und arbeiten mit (technisch) veralteter Software (so genannten Altsystemen oder Legacy-Systemen; Legacy = Vermächtnis). In der Literatur werden Legacy-Systeme definiert als „any information system that significantly resists modification and evolution“ (Brodie / Stonebraker, 1995). Parareda / Pizka (2006) weisen darauf hin, dass Legacy keine Frage des Alters ist. „Junge“ Systeme, die unter Zeitdruck, mit knappen Budgets und unzureichen-

*Legacy-  
Systeme*

der Qualifikation erstellt werden, entarten oft schon während ihrer Entwicklung zu Legacy-Systemen, deren Wartung überproportional teuer wird.

*Basis für Informationsflüsse*

Häufig bilden diese so genannten Legacy-Systeme die Basis für sämtliche Informationsflüsse sowie für die Konsolidierung der Daten im Unternehmen. Ein Ausfall eines derartigen Systems kann somit unter Umständen erhebliche Auswirkungen auf die Geschäftstätigkeit des Unternehmens haben (Bennett, 1995).

*Häufig genutzt*

Trotz der damit verbundenen Kosten und Risiken sind noch viele Legacy-Systeme im Einsatz. So zeigten beispielsweise Studien, dass noch im Jahr 2002 rund 80 % der IT-Systeme auf Legacy-Plattformen liefen. Gleichzeitig schätzte die International Data Corporation, dass ca. 200 Mrd. Zeilen Legacy Code in Verwendung waren (Zoufaly, 2002).

*Schwachstellen*

Die Programmarchitektur und Softwareinfrastruktur der alten Systeme erfüllen jedoch oftmals nur unzureichend die heutigen Anforderungen, die an Sicherheit, Leistungsfähigkeit, Adaptierbarkeit und Verfügbarkeit moderner Zielplattformen gestellt werden. Darüber hinaus gefährdet rückläufiges Programmierwissen in den Sprachen dieser

Systeme die weitere Existenz der Systeme und hohe Wartungs- und Lizenzkosten belasten die IT-Budgets.

In Anbetracht dieser über die Jahre gewachsenen Landschaften und ihren Schwachstellen stehen die IT-Leiter und Chief Information Officers (CIO) immer häufiger aus technischen und/ oder funktionalen Gründen vor der Herausforderung, Teile der vorhandenen Altsysteme zu modernisieren. Dieser Druck wird umso größer, je mehr sich einerseits die technischen Innovationszyklen verkürzen und andererseits die IT-Budgets schrumpfen.

Durch einfaches Abschalten der Systeme sind diese Probleme in vielen Unternehmen aber nicht zu lösen. Nicht selten unterstützen die über die Jahre gewachsenen Systeme unternehmensspezifische, geschäftskritische Prozesse und sind daher wettbewerbsrelevant für die Unternehmen. Auch verkörpern sie das Wissen vieler Expertengenerationen und unzähliger investierter Arbeitsstunden. Sie stellen damit einen unschätzbaren Wert für das Unternehmen dar, ganz zu schweigen von den historischen Datenbeständen, die in diesen Systemen abgelegt sind (Seacord, 2002; Aminian, 2003).

*Herausforderung*

*Wert der Legacy-Systeme*

Somit ist die Frage, wie mit diesen Systemen umzugehen ist, ein zentrales Thema für IT-Leiter und CIOs, um einerseits die getätigten Investitionen zu schützen, andererseits aber zukunftsorientiert agieren zu können.

## 1.2 Migration als Lösungsansatz

### *Lösungsansätze*

In Wissenschaft und Praxis werden verschiedene Ansätze zur Lösung des Problems diskutiert. Diese reichen von der Ummantelung bis zur kompletten Neuimplementierung (vgl. dazu ausführlich Kap. 2.1).

### *Migration als Lösung*

Ein häufig angeführtes Verfahren ist die so genannte Migration. Grundgedanke einer Migration ist, ein existierendes, im Betrieb befindliches System auf eine neue Plattform zu überführen und dabei zum einen die Funktionalität des alten Systems uneingeschränkt beizubehalten und zum anderen so wenig wie möglich Störungen im täglichen Ablauf zu verursachen (Bisbal et al., 1999c).

### *Migration als Managementthema*

Da eine Migration immer mit erheblichen Investitionen verbunden ist, sollte eine Entscheidung für eine Migration niemals als rein technische Fragestellung betrachtet werden. Vielmehr sollte aus einer Managementperspektive die Sicht des Gesamtunternehmens eingenommen wer-

den und eine sowohl betriebswirtschaftlich als auch technisch fundierte Entscheidung getroffen werden.

Während es eine Fülle von Artikeln und Konferenzen gibt, die sich mit technischen Aspekten der Migration beschäftigen, gibt es nur wenig Literatur, die sich unter Managementgesichtspunkten mit diesem Thema auseinandersetzt. Vor diesem Hintergrund greift der vorliegende „Quick Guide to Migration“ die wichtigsten Managementfragestellungen im Zusammenhang mit Migrationsüberlegungen auf.

*Managementliteratur fehlt*

Die Darstellung gliedert sich in vier Teile (Abb. 1):

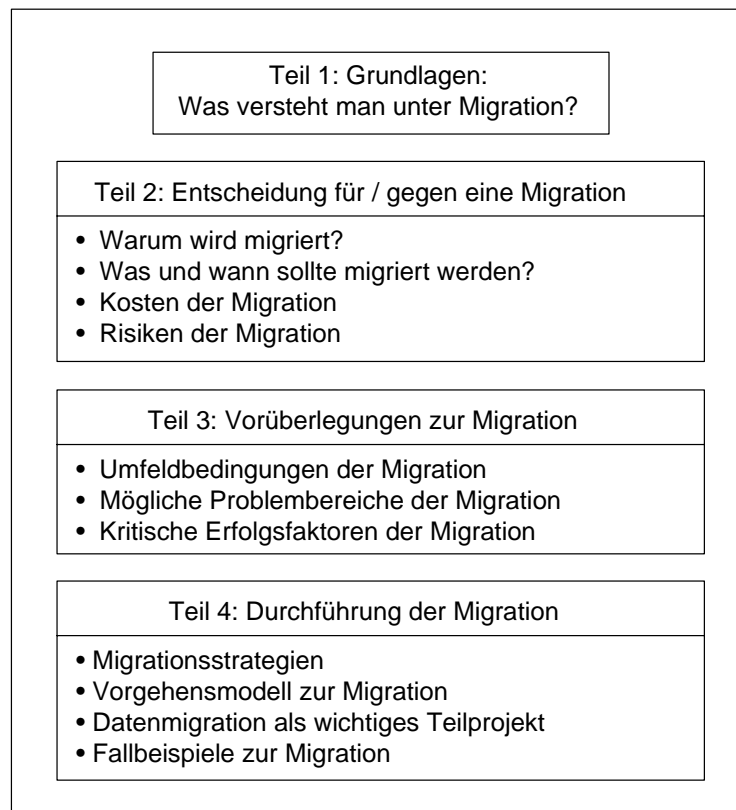
- *Teil 1* erklärt den Begriff „Migration“ und ordnet ihn in den Gesamtzusammenhang des Software-Lebenszyklus ein.
- *Teil 2* beleuchtet unterschiedliche Aspekte des Entscheidungsprozesses für oder gegen eine Migration.
- *Teil 3* zeigt, basierend auf der Annahme, dass eine Migration durchgeführt werden soll, Teilbereiche auf, die proaktiv gemanaged werden sollten, damit die Migration erfolgreich verlaufen kann.
- *Teil 4* geht auf die eigentliche Migration ein. Neben verschiedenen Migrationsstrategien wird ein Vor-

*Aufbau Migrationsguide*



gehensmodell vorgestellt sowie das Thema der Datenmigration beleuchtet. Den Abschluss bilden Fallstudien, die die dargestellten Inhalte anhand verschiedener Praxisbeispiele verdeutlichen.

Abb. 1: Aufbau des Quick Guide (Quelle: Eigene Darstellung)



## 2 Was versteht man unter „Migration“?

### 2.1 Migration im Lebenszyklus von Systemen

Das Stichwort „Migration“ (lat. migrare – wandern) findet sich in vielen Büchern, Broschüren und Internetseiten – eine Definition des Begriffs jedoch nur außerordentlich selten. So erstaunt es nicht, dass dieser Begriff schwierig zu fassen ist und die Interpretationen des Begriffs auseinanderklaffen.

*Keine einheitliche Definition*

Um ein einheitliches Begriffsverständnis als Basis für die weiteren Ausführungen in diesem Buch zu schaffen, wird der Begriff der „Migration“ in Zusammenhang zum Lebenszyklus von IT-Systemen gestellt und gegenüber anderen Strategien zum Umgang mit Altsystemen abgegrenzt.

Typischerweise wird der Lebenszyklus eines IT-Systems in die Phasen Plan, Build, Run, Sun Down unterteilt, wobei die Phasen Plan und Build vor dem eigentlichen Ein-

*Lebenszyklus*

satz des Systems liegen. Wenn das System produktiv ist (Phase Run) und im Unternehmen genutzt wird, setzen Wartung und Weiterentwicklung (Maintenance & Enhancement) ein. Hierbei geht es um kleinere Weiterentwicklungen sowie die Ausbesserung von Fehlern.

*Wartung* Je länger das System im Einsatz ist, desto mühsamer und schwieriger wird die Wartung (Lehmann / Belady, 1985). Die Arbeiten, die im Rahmen der Wartung zu leisten sind, können in vier Kategorien eingeteilt werden. Dazu gehören im Wesentlichen (Lientz et al., 1980, S. 68):

*Korrektur*

- Die korrigierende Wartung, d.h. die Eliminierung von Entwurfs- und / oder Implementierungsfehlern (ca. 22 % des Gesamtwartungsaufwands).

*Adaption*

- Die adaptive Wartung, d.h. die Anpassung des Systems an neue Hardware und / oder neue Basissoftware (ca. 24 % des Gesamtwartungsaufwands).

*Optimierung*

- Die optimierende Wartung, d.h. die Speicher- und Laufzeitoptimierung, die Verbesserung der Systemqualität sowie die Verbesserung bzw. Erstel-

lung der Programmdokumentation (ca. 10 % des Gesamtwartungsaufwands).

- Die erweiternde Wartung, d.h. die Anpassung an Benutzerwünsche in Bezug auf die Erweiterung und / oder Änderung existierender Funktionalitäten (ca. 42 % des Gesamtwartungsaufwands).

*Erweiterung*

Eine Studie von Eick et al. (2001) zeigte, dass der Verlust an Softwarequalität mit der Anzahl der durchgeführten Änderungen korreliert. Dieses wurde bereits 1990 von Sneed / Kaposi (1990) gezeigt, die zu dem Ergebnis kamen, dass die Anpassung eines Systems, die nicht auf eine Qualitätsverbesserung abzielt, stets die Wartbarkeit des Systems weiter verschlechtert. Ihrer Untersuchung nach steigt die Komplexität eines Programms nach einer Korrektur durchschnittlich um 4 %, nach einer Änderung um 17 % und nach einer Erweiterung um 26 %. Das bedeutet, dass der für die nächste Modifikation notwendige Wartungsaufwand stetig zunimmt. Am Ende steht ein Softwareprodukt, das praktisch nicht mehr wartbar ist.

*Komplexität durch Wartung*

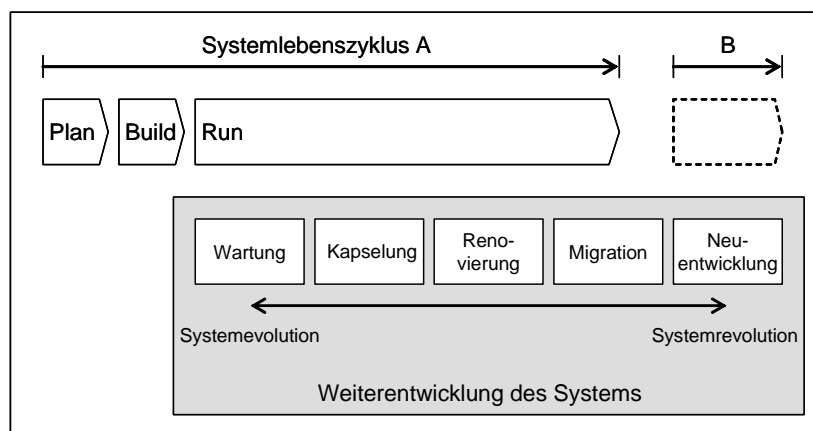
In Literatur und Praxis werden daher für diese Fälle andere Ansätze beschrieben bzw. verfolgt, um ein Altsystem an neue Anforderungen anzupassen (Bisbal et al.

*Anpassung von Altsystemen*

Was versteht man unter „Migration“?

1999a; Weidermann et al., 1997). Die folgende Abbildung zeigt die unterschiedlichen Ansätze, geordnet nach dem Umfang der nötigen Änderungen am Altsystem.

Abb. 2: Ansätze zur Erneuerung von Systemen. (Quelle: Eigene Darstellung)



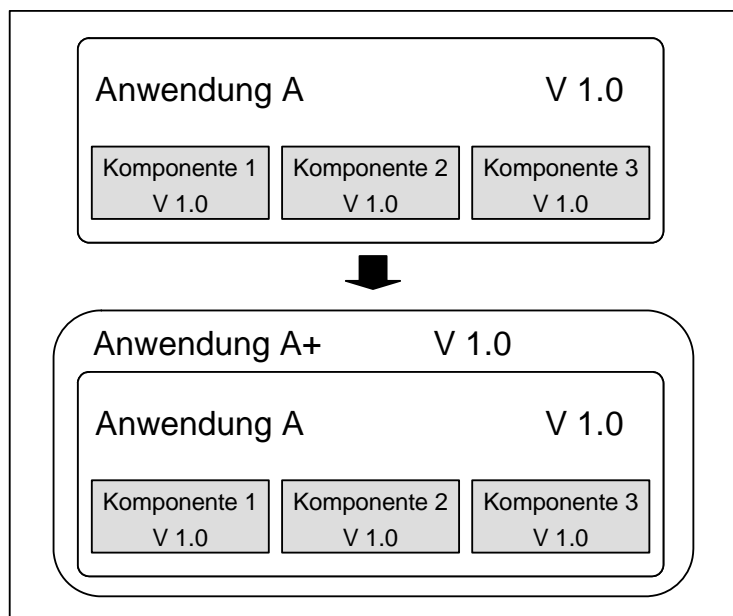
### Kapselung

Bei der Kapselung wird das Altsystem mit einer Softwareschicht „ummantelt“ (engl. Wrapping), die die ungewünschte Komplexität des Altsystems verbirgt und eine moderne Schnittstelle bereitstellt (Shaw, 1995).<sup>1</sup> Darüber hinaus kann das System auf diese Weise um zusätzliche Funktionalitäten außerhalb der Kapselung erweitert werden. Der Kern, d.h. die Funktionalität, des alten Systems bleibt unverändert bestehen (vgl. Abb. 3) - eine Ablösung

<sup>1</sup> Ein Spezialfall der Kapselung ist das Screen Scraping. Von Screen Scraping spricht man, wenn ein Altsystem mit textbasierter Oberfläche, i.d.R. 3270-Terminals, um eine graphische Benutzerschnittstelle erweitert wird, jedoch ohne die Altanwendung zu ändern.

des Altsystems findet nicht statt. Neue Komponenten und Altsystem werden auf Dauer parallel betrieben. Auf diese Weise kann das System von außen an geänderte Anforderungen angepasst werden.

Abb. 3: Kapselung. (Quelle: Eigene Darstellung)



Bei der Renovierung (in der Literatur teilweise auch als Sanierung bezeichnet) werden größere Änderungen am Altsystem vorgenommen.<sup>2</sup> Die bestehenden Komponenten werden mehrheitlich in einem einmaligen Reenginee-

*Renovierung*

<sup>2</sup> Sneed et al., 2005, 229ff weisen darauf hin, dass bei Sanierungsprojekten mehr als 50% des Codes überarbeitet wird. Zu Sanierungsprojekten vgl. ausführlich Miller, 1998.

## 2

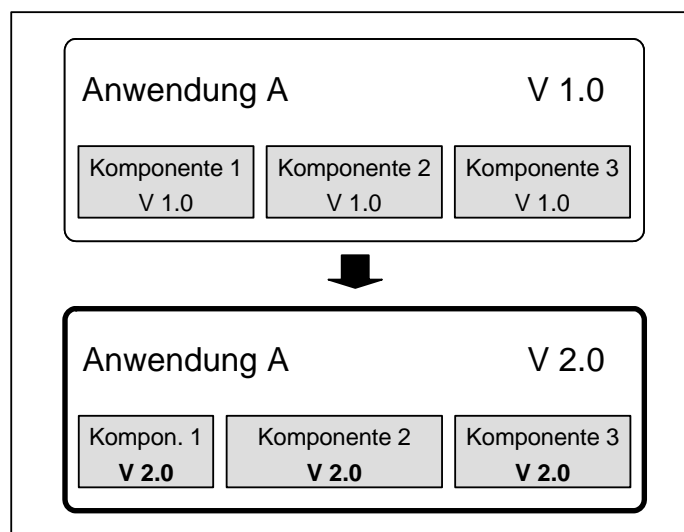
### Ziele der Renovierung

Was versteht man unter „Migration“?

ring-Projekt bereinigt und / oder restrukturiert (vgl. Abb. 4).

So können z.B. erodierte Strukturen wiederhergestellt werden oder der Code wird in unabhängige Module zerlegt. Ziel einer Renovierung ist es, sowohl die Komplexität der Anwendung zu reduzieren und dadurch die Erhaltungskosten zu senken, als auch die Qualität zu erhöhen. Insgesamt soll die Anwendung flexibler und offener für zukünftige Anforderungen gemacht werden.

Abb. 4: Renovierung. (Quelle: Eigene Darstellung)



### Software-Migration

Bei der (Software-)Migration werden wesentliche Teile eines Systems komponentenweise durch ein anderes

System ersetzt und die Daten aus einer Umgebung in eine andere transferiert. Auslöser sind geänderte Anforderungen an das Softwaresystem, wie beispielsweise organisatorische Änderungen der Geschäftsprozesse, Erweiterungen der Softwarefunktionalität oder externe (z.B. gesetzliche) Anforderungen, die Überarbeitungen oder Erweiterungen der Softwaresysteme erfordern. Ohne eine Überführung des bestehenden Systems in eine neue Umgebung sind diese Anforderungen nicht oder nur mit großem Aufwand realisierbar (Gimmich / Winter, 2005).

So kann es z. B. sein, dass das vorhandene Anwendungssystem auf einem anderen Rechner, unter einem anderen Betriebssystem oder mit einem anderen Datenbanksystem laufen soll. Es könnte aber auch sein, dass die Programmiersprache gewechselt wird. Unter Umständen kommen alle vier Aspekte gleichzeitig zum Tragen, zum Beispiel bei einer Migration vom Host in eine verteilte objektorientierte Client-/Server-Umgebung.

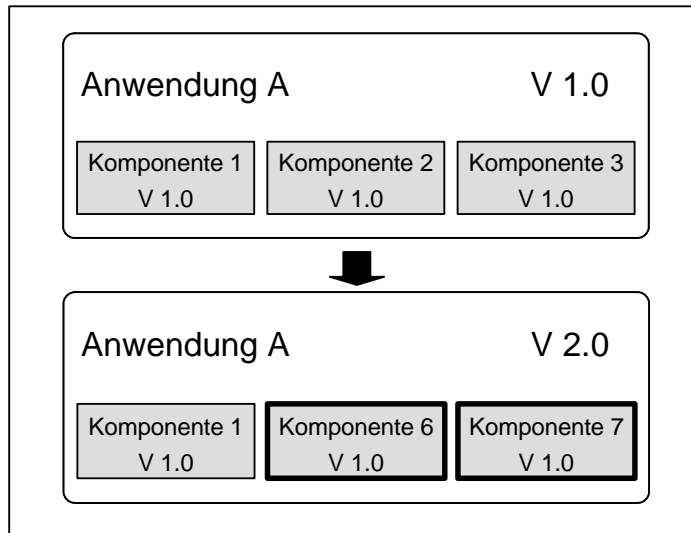
Ist der Umfang der Änderungen bei einer Migration geringer als bei einer vollständigen Neuimplementierung des Systems, wird von einer partiellen Migration gesprochen (vgl. Abb. 5).

*Beispiele*

*Partielle Migration*



Abb. 5: Partielle Migration (Quelle: Eigene Darstellung)

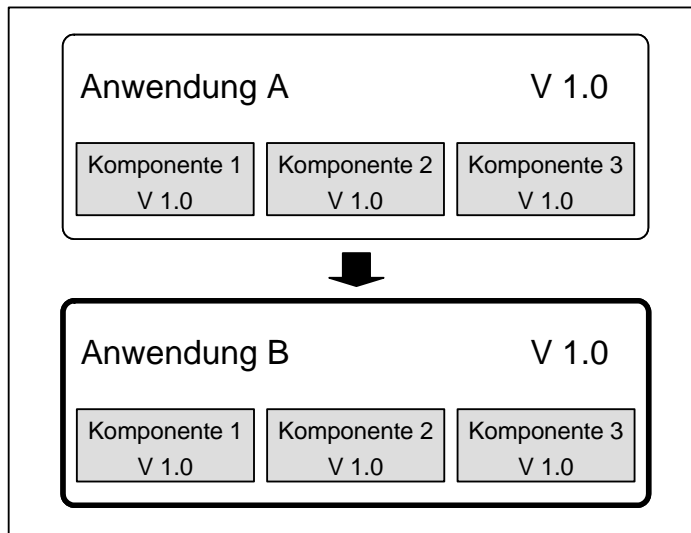
*Vollmigration*

Eine so genannte Vollmigration ist dann gegeben, wenn das komplette System abgelöst wird. In diesem Fall ist die Migration gleichzusetzen mit einer Neuimplementierung (vgl. Abb. 6).

*Migration als technische Transformation*

Eine Migration ist immer eine rein technische Transformation mit einer klaren Anforderungsdefinition – funktionale Veränderungen am System sind i. d. R. nicht damit verbunden. Das zu migrierende Altsystem beschreibt eindeutig die Systemfunktionalität, deren Erhalt nach erfolgreicher Migration durch Regressionstests überprüft werden kann (Gimmich / Winter, 2005).

Abb. 6: Vollmigration (Quelle: Eigene Darstellung)



Nach dieser Abgrenzung der Migration gegenüber den anderen Verfahren zum Umgang mit Altsystemen wird nun die Migration näher betrachtet.

## 2.2 Migrationstypen

Migrationen sind in der Regel Teil größerer Reengineering-Maßnahmen, bei denen vorhandene Softwareartefakte in eine neue Umgebung überführt werden, ohne hierbei die Fachlichkeit zu verändern. Bei genauerer Betrachtung kann eine Migration auf technischer Ebene unterschiedlichste Bereiche betreffen (Gimmich / Winter, 2005):

*Teilbereiche  
der Migration*

## 2

Was versteht man unter „Migration“?

### *Hardware-Migration*

- Eine Hardware-Migration beinhaltet den Wechsel der zugrunde liegenden Hardwareumgebung wie z.B. den Wechsel von einer Mainframe-Umgebung zu einer Client-/Server-Umgebung.

### *Migration der Laufzeitumgebung*

- Eine Migration der Laufzeitumgebung bezieht sich auf den Wechsel der dem System zugrunde liegenden Systemsoftware wie beispielsweise Betriebssysteme oder Datenbankmanagement-Systeme.

### *Architektur-Migration*

- Eine Architektur-Migration umfasst eine grundlegende Änderung der Systemstruktur, wie sie beispielsweise beim Übergang von monolithischen Systemen zu Mehr-Schichten-Architekturen auftritt.

### *Migration der Entwicklungsumgebung*

- Eine Migration der Entwicklungsumgebung bezeichnet den Wechsel der Programmierumgebung. Dazu gehört z.B. die (teil)automatische Konvertierung von Programmen aus traditionellen Programmiersprachen wie z.B. COBOL oder PL/I in moderne(re) Sprachen wie C++ und Java, .net.

Je nach Zielsetzung des Vorhabens sind von einer Software-Migration folgende Aspekte betroffen: Daten, Benutzerschnittstellen und Programme (Sneed, et al. 2005).

*Software-Migration*

Eine Daten-Migration umfasst die Überführung der Datenbestände eines Systems in ein anderes System. Hierbei sind neben den eigentlichen Daten auch die den Daten zugrunde liegenden Datenstrukturen oder Schemata zu migrieren.

*Daten-Migration*

Eine Benutzerschnittstellen-Migration umfasst die Übertragung der Interaktionskomponenten mit dem Anwender, d. h. die Masken des alten Systems werden in eine neue GUI (Graphical User Interface) transformiert.

*Benutzerschnittstellen-Migration*

Eine Programm-Migration beschäftigt sich mit der Transformation der ausführbaren Programmlogik.

*Programm-Migration*

Vereinfacht bietet sich die folgende Sichtweise an: Eine Migration lässt sich unterteilen in die oft eng miteinander verknüpften Prozesse der eigentlichen System-Migration und der Daten-Migration (vgl. Tab. 1). Dennoch kann es sowohl eine System-Migration ohne Daten-Migration als auch eine Daten-Migration ohne System-Migration geben.

*System- vs. Daten-Migration*

Tab. 1: Teilbereiche der Migration (Quelle: Eigene Darstellung)

Migrationstypen	
System-Migration	Daten-Migration
z. B.: - Austausch der Programmiersprache  - Portieren der Anwendung auf neue Plattform  - Hardware-Migration	z. B.: - Modernisierung der Datenhaltung  - Übernahme des Datenbestands auf neue Datenbank

*Daten-Migration vernachlässigt*

Während Fragen der System-Migration durchaus große Beachtung in Wissenschaft und Praxis geschenkt wird, wird das Thema Daten-Migration in der einschlägigen Literatur kaum berücksichtigt (Willinger/Gradl, 2006, 11).

Dieses ist aus verschiedenen Gründen erstaunlich:

*Häufige Notwendigkeit*

- Eine Daten-Migration ist in der Regel bei jeder Ablösung eines bestehenden Softwaresystems durch ein Nachfolgesystem durchzuführen und damit eine sehr häufige Notwendigkeit.

- Nicht nur die Menge sondern insbesondere auch der Aussagegehalt der migrierten Daten, d.h. die Datenqualität, spielt eine entscheidende Rolle für die Abwicklung des operativen Geschäfts in dem neuen System.
- Die Daten-Migration kann je nach Projektumfang unter Umständen einen Großteil der Projektressourcen binden, was zu personellen und finanziellen Engpässen führen kann.
- Da die Daten-Migration typischerweise erst recht spät im Projektverlauf stattfindet, hat jede Verzögerung unmittelbare Auswirkungen auf den Endtermin des Projektes. Aus dieser Sicht ist die Daten-Migration ein kritisches Teilprojekt.
- Oftmals ist es noch möglich ein verzögertes Projekt dadurch zu „retten“, dass das System mit zunächst reduzierter Funktionalität eingeführt wird. Eine Einführung mit fehlenden oder nur teilweise migrierten Daten ist aber niemals möglich.
- Daten-Migrationsprojekte können sehr unterschiedliche Anlässe haben. Der klassische Grund

*Datenqualität kritisch*

*Große Ressourcenbindung*

*Verzögerung Endtermin*

*Voraussetzung für Produktivstart*

*Wiederkehrende Problematik*

ist die Einführung eines neuen Systems, in das Daten anderer Systeme übernommen werden müssen. Daneben können aber beispielsweise auch Systemkonsolidierungen oder Upgrades bestehender Systeme Auslöser für Daten-Migrationen sein. Unabhängig davon, welcher Auslöser der Daten-Migration zugrunde liegt, die im Rahmen der Migration zu lösenden Probleme sind immer die gleichen (Morris, 2006, 1).

Vor diesem Hintergrund wird im Folgenden neben allgemeinen Betrachtungen zum Thema Migration dem Aspekt der Daten-Migration besondere Beachtung geschenkt.

## 3 Software-Migration - Notfall oder strategische Neuorientierung ?

### 3.1 Auslöser einer Migration

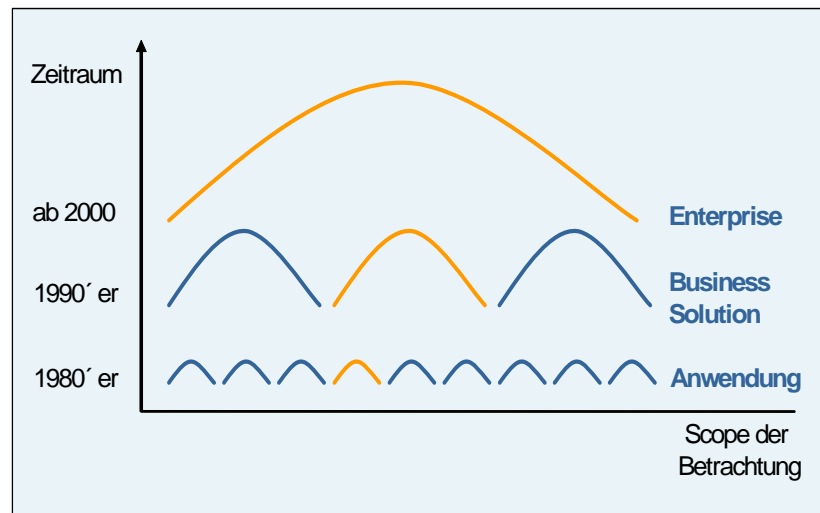
Vielfach wird die Auffassung vertreten, dass Migrationsprojekte lästige Pflichtübungen sind, um veraltete Hardware und / oder Software abzulösen und ein entsprechendes Handeln somit unvermeidbar ist. Neben diesen unvermeidbaren „Pflichtübungen“, die sich häufig auf einzelne Systeme beziehen, gibt es jedoch ein breites Spektrum anderer Aspekte, weshalb ein Migrationsprojekt durchaus notwendig, sinnvoll oder gar von strategischer Bedeutung für das Unternehmen sein kann. Diese Anlässe werden im Folgenden anhand der Betrachtungsebene systematisiert.

Die nachfolgende Abbildung zeigt verschiedene Blickwinkel bzw. Ebenen, aus denen heraus die IT-Landschaft eines Unternehmens betrachtet werden kann. Je nach gewählter Perspektive ergeben sich andere Motivationslagen für eine Migration.

*Migration als „Pflichtübung“*



Abb. 7: Sichten auf die IT-Landschaft von Unternehmen  
(Quelle: Eigene Darstellung)



### Funktions-orientierung

In den 1980' er Jahren war die Sichtweise sehr stark von der Betrachtung einzelner Anwendungen geprägt. Dem Gedanken der Spezialisierung und Funktionsorientierung folgend stand das einzelne System im Mittelpunkt der Betrachtung. Es unterstützte einen eng abgegrenzten Aufgabenbereich, der in der Regel eindeutig definiert war.

### Prozess-orientierung

Mit dem Aufkommen der Geschäftsprozessorientierung in den 1990' er Jahren verlagerte sich der Fokus hin zu so genannten Business Solutions, d.h. einem Geflecht mehrerer Anwendungen, die gemeinsam eine durchgängige

betriebswirtschaftliche Aufgabenstellung (einen Geschäftsprozess) unterstützen.

Die heutige Diskussion ist stark von einer unternehmensweiten Sicht geprägt. Schlagworte wie Enterprise Application Integration (EAI), Enterprise Architecture Management (EAM) oder serviceorientierte Architekturen (SOA), die heute in nahezu allen IT-Veröffentlichungen die Headlines beherrschen, spiegeln diesen Trend wider. Nicht mehr die einzelne Anwendung, sondern die IT-Landschaft als Ganzes im Zusammenwirken mit den Geschäftsprozessen rückt in den Mittelpunkt der Betrachtungen des IT-Managements.

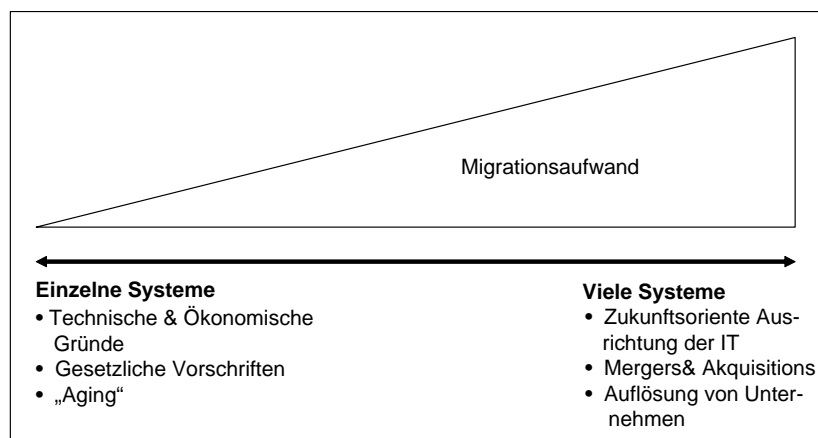
Entsprechend der unterschiedlichen Sichten auf die IT-Landschaft kann auch die Motivation für ein Migrationsprojekt sehr unterschiedlich sein. Geht es in dem einen Fall um die Migration eines einzelnen Anwendungssystems, um beispielsweise das operative Doing zu erleichtern und zu verbessern, so kann im anderen Fall eine strategische Neuausrichtung der kompletten IT-Architektur des Unternehmens ein Auslöser für Migrationsprojekte sein.

*Unternehmenssicht*

*Bandbreite der Auslöser*

Die folgende Abbildung zeigt die Bandbreite der Auslöser und den damit verbundenen Migrationsaufwand, bevor einzelne Auslöser im Detail erläutert werden.

Abb. 8: Bandbreite der Auslöser (Quelle: Eigene Darstellung)



### Vielfältige Auslöser

Im Folgenden werden aus dem breiten Spektrum drei Aspekte aufgegriffen und eingehender betrachtet. Im letzten Unterkapitel werden weitere Auslöser kurz aufgelistet, ohne diese im Detail zu erläutern.

### 3.1.1 Ablösung von Legacy-Systemen aus technischen und ökonomischen Gründen

Veränderungen am Markt zwingen Unternehmen kontinuierlich auf Neuerungen zu reagieren, um wettbewerbsfähig zu bleiben. Diese Veränderungen wirken sich auf Legacy Systeme aus und machen Anpassungen an diesen erforderlich. So stehen beispielsweise Finanzdienstleister vor der Herausforderung, ihren Kunden und Geschäftspartnern zu jeder Zeit und von jedem Ort aus Zugriff auf Informationen und angebotene Dienstleistungen zu ermöglichen. Dieses setzt häufig einen Zugriff auf Legacy-Systeme voraus, die nicht für diese Arbeitsweise konzipiert wurden und daher denkbar ungeeignet dafür sind (Aminian, 2003).

Viele Unternehmen, insbesondere aber zahlreiche große Finanzdienstleister, stehen vor der Anforderung, ihre immer komplexer werdenden Systemlandschaften aus technischen Gründen zu migrieren. So zeigte eine Studie der Forschungseinrichtung E-Finance Lab im Jahr 2006, dass die eingesetzte Kernapplikation bei fast der Hälfte aller befragten Banken bereits seit 1999 im Einsatz ist (vgl. Abb. 9), wobei das Einführungsjahr der Applikation negativ mit der Größe des Instituts korreliert. Das bedeu-

*Wettbewerbsdruck*

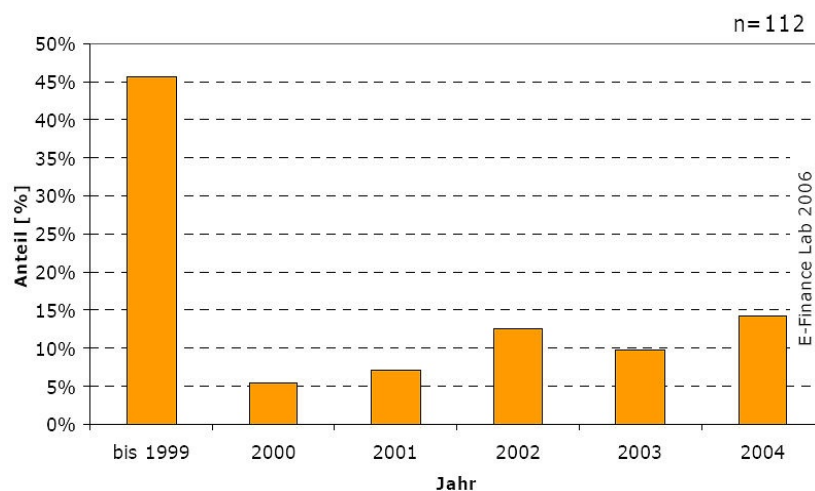
*Kernapplikation älter als 7 Jahre*

Nur 51%  
finden SW  
„modern“

tet: Je größer das Haus, desto älter die Kernapplikation (Wolff, 2006).<sup>3</sup>

Nur rund 50,8% der befragten Banken schätzen die genutzte Software als modern ein und nur rund die Hälfte schöpft den Funktionsumfang der Software aus. Unklar blieb jedoch, ob dieses so gewollt ist oder durch dominante modernere Technologien substituiert wird (Wolff, 2006).

Abb. 9: Entwicklungsjahr der Kernapplikation. (Quelle: Wolff, 2006)



<sup>3</sup> Für die Studie „Kritische Erfolgsfaktoren in Finanzprozessen“ befragte E-Finance Lab 1.020 der größten deutschen Kreditinstitute. 136 Fragebögen wurden zur Auswertung zurückgesandt.

Darüber hinaus zeigen sich insbesondere bei Finanzdienstleistern folgende typische Mängel:

- Viele Unternehmen arbeiten nach wie vor mit veralteten und auslaufenden Technologien und Plattformen. Häufig sind Server und Software nicht skalierbar, wodurch ein hoher Aufwand und folglich hohe Betriebskosten resultieren.
- Durch eine über die Jahre gewachsene IT-Infrastruktur ist die Umgebung durch eine heterogene Serverlandschaft geprägt. Die Hardware stammt von diversen Herstellern aus unterschiedlichen Jahren. Sie ist sehr teuer zu unterhalten.
- Hersteller bieten keine Weiterentwicklung und keinen Support mehr für ihre alten Produkte. So löst beispielsweise die Abkündigung von Betriebssystemversionen häufig eine Welle von Migrationsprojekten aus.
- Die Systeme sind nicht oder nur unzureichend dokumentiert. Am System durchgeführte Änderungen wurden nicht systematisch erfasst und sind daher nicht mehr nachvollziehbar.

*Fehlende Skalierbarkeit*

*Heterogene Serverlandschaft*

*Abkündigungen*

*Mangelhafte Dokumentation*

### 3

*Fehlendes Know-how*

*Fehlende Wirtschaftlichkeit*

*Software-Migration - Notfall oder strategische Neuorientierung ?*

- Viele Mitarbeiter, die ursprünglich an der Entwicklung des Systems beteiligt waren, sind aus dem Unternehmen ausgeschieden oder arbeiten in anderen Bereichen. Daher verstehen nur noch wenige Experten den inneren Aufbau des Systems. Häufig versteht, aufgrund der vielen Änderungen im Laufe der Zeit, niemand mehr das System als Ganzes.
- Es ist nicht selten, dass die Altanwendungen über zehn Millionen Zeilen Code umfassen, von denen in der Realität aber nur ca. 30 % geschäftsrelevante Prozesse abbilden (Wölfle et al., 2006). Ab einem gewissen Punkt sind daher der finanzielle und zeitliche Aufwand für Anpassung und Wartung so hoch, dass ein Weiterbetreiben des Systems aus wirtschaftlichen Gesichtspunkten nicht mehr sinnvoll ist. Laut Berbner et al. (2006) werden bis zu 40% der IT-Ausgaben in den Unternehmen für die Wartung und den Betrieb der über die Jahre gewachsenen IT-Landschaft ausgegeben.<sup>4</sup>

---

<sup>4</sup> Bereits Ende der 1970'er Jahre kamen zwei Untersuchungen zu dem Ergebnis, dass ca. 50% des DV-Personals für Wartungsarbeiten eingesetzt wurden (Fjeldsad/Hamlet, 1979, Lientz et al., 1980). In den 1990'er Jahren wurde der Anteil der Wartung von der Gartner Group sogar auf über 80% veranschlagt (CAP debis GEI, 1992, S. 2).

### 3.1.2 Mergers & Akquisitions

Viele Branchen sind derzeit von Fusionstendenzen gekennzeichnet. Insbesondere bei Finanzdienstleistern wird derzeit auf diesem Weg versucht, politisch oder managementmotiviert einen größeren Marktanteil zu erzielen (Dewal / Schnichels, 2000).

Erfolgreiches Banking wird in der Zukunft jedoch nur möglich sein, wenn es gelingt, einerseits die internen Strukturen der fusionierten Banken effizient und flexibel an die Markteinflüsse anzupassen, andererseits die heterogenen IT-Systeme der einstmals eigenständigen Unternehmen so zu integrieren, dass die operativen Prozesse optimal ablaufen können.

Da die früher eigenständigen Unternehmen jedoch in der Vergangenheit unabhängig voneinander ihre Systeme entwickelt haben, verwenden die Systeme unterschiedliche Kommunikationsprotokolle und stellen Daten in unternehmensspezifischen Formaten zur Verfügung. Nicht selten behindern sie so Fortschritt und Innovation.

Die großen IT-technischen Herausforderungen bei Mergers & Akquisitions liegen somit in der

*Fusionen*

*Herausforderungen*

*Technische Barrieren*



# 3

*Software-Migration - Notfall oder strategische Neuorientierung ?*

*Gemeinsame  
IT-Landschaft*

- Definition einer gemeinsamen IT-Landschaft, in der die Plattformen und Systeme der verschiedenen Welten zusammengeführt werden.

*Gemeinsame  
Datenbestände*

- Bereinigung und Überführung der getrennten Datenbestände in einem gemeinsam nutzbaren Datenbestand.

### 3.1.3 Zukunftsorientierte Ausrichtung der IT-Architektur

*Zukunftsorientierung*

Nahezu alle Branchen sind seit Jahren von einem verschärften Wettbewerb gekennzeichnet. Insbesondere die Finanzdienstleistungsbranche befindet sich derzeit im Umbruch. Als wichtigste Herausforderung im Bankenmarkt sehen die Bankmanager laut einer Studie den verstärkten Preiswettbewerb bei Standardprodukten. Darüber hinaus haben sich die Bankkunden daran gewöhnt, „einfache“ Geschäfte online abzuwickeln. Zur Steigerung der Ertragskraft bedarf es daher Neuerungen an der Kundenschnittstelle, die ergänzt um effiziente Back-Office Systeme zur Gesamteffizienz der Bank beitragen.

Ralf Steinmetz vom E-Finance Lab stellt vor diesem Hintergrund den Zusammenhang zur IT her:

*„Wer im internationalen Wettbewerb bestehen will, darf nicht weiter auf unflexible, monolithische und schwer zu wartende IT-Lösungen setzen“. (zit. nach Wolff, 2007a).*

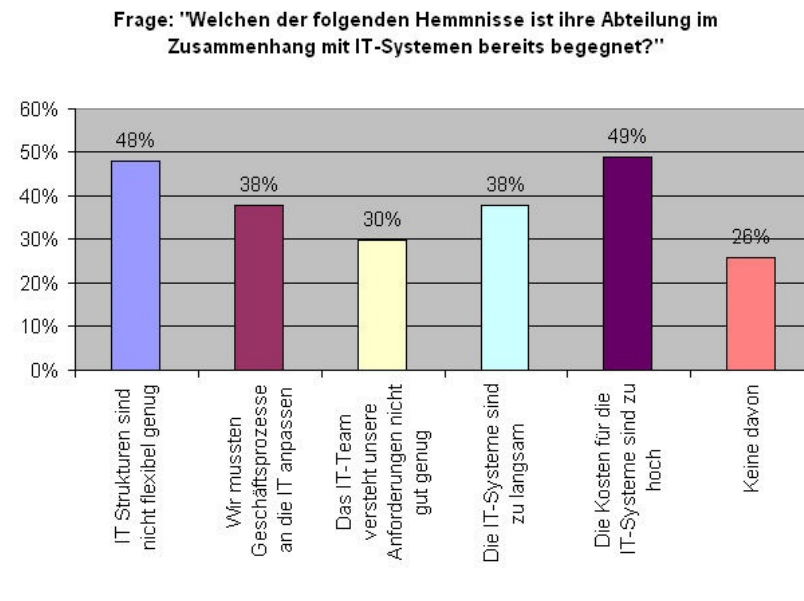
Eine Studie des Marktforschungsinstituts Vanson Burne ergab ein ernüchterndes Bild für die Finanzdienstleister (vgl. Abb. 10): 48 % der Befragten gaben an, die IT sei nicht flexibel genug, um die Geschäftsanforderungen angemessen abzubilden. 38 % erklärten sogar, sie müssten die Geschäftsprozesse an die IT anpassen, nicht umgekehrt (Püttner, 2007).

Als Ausweg aus dem Dilemma sehen viele Analysten und Berater den Einstieg in so genannte serviceorientierte Architekturen (SOA), durch die sich die Finanzdienstleister schneller auf neue Gegebenheiten einstellen könnten (Borbner et al., 2006).

*Fehlende  
Flexibilität*

*Service-  
orientierte  
Architekturen*

Abb. 10: Finanzdienstleister und ihre IT-Probleme.  
(Quelle: Püttner, 2007)



### Wiederverwendbare Services

Serviceorientierte Architekturen haben in den letzten Jahren erheblich an Popularität gewonnen. Eine SOA beruht auf der losen Kopplung wiederverwendbarer Softwarebausteine (Services). Services spezifizieren gekapselte (Fach-)Funktionalitäten, die unabhängig von einer konkreten Implementierung sind.

### Vorteile von SOA

Der Vorteil der serviceorientierten Architekturen wird in einer größeren Anpassungsfähigkeit, schnelleren Implementierungszeiten und geringeren Kosten für die Anwendungsentwicklung und Integration gesehen.

Eine aktuelle branchenübergreifende Erhebung der Experton Group zum Stand der SOA-Aktivitäten in Deutschland zeigt, dass sich bislang nur wenige Unternehmen vertiefend mit der SOA-Thematik auseinandergesetzt haben. Rund 55% der befragten 144 Unternehmen plant jedoch eventuell die Umsetzung eines SOA-Projekts (Zacher / Fritsch, 2007).

*SOA kaum umgesetzt*

Bei den Banken und Versicherungen sieht die Situation jedoch noch anders aus. So gaben in der Studie von Vanson Bourne 68 % der Befragten an, noch nie etwas von SOA gehört zu haben und nur einer von 100 Studienteilnehmern hat eine entsprechende SOA Strategie aufgesetzt (Püttner, 2007).

*Kaum SOA Kenntnisse*

Basierend auf ihren Umfrageergebnissen geht die Experton Group davon aus, dass sich in den nächsten fünf bis 10 Jahren jedoch erhebliche Veränderungen am SOA-Markt ergeben können, wenn viele Unternehmen stärker sensibilisiert sind und die bestehenden Applikationsinfrastrukturen neu organisieren (Zacher / Fritsch, 2007). Die Analysten von Gartner erwarten sogar, dass noch in diesem Jahr die Hälfte aller geschäftskritischen Anwendungen mit serviceorientierten Architekturen gestaltet wird. Bis 2010 sollen es sogar 80% sein (Jung, 2007).

*SOA im Aufwind*

*Herausforderung*

Die Herausforderung für die Unternehmen auf dem Weg von Legacy Systemen zu serviceorientierten Architekturen wird darin bestehen, das angestrebte serviceorientierte Unternehmensmodell mit den vorhandenen Altsystemen in Übereinstimmung zu bringen (vgl. hierzu ausführlich Winter / Ziemann, 2005; Gimmich, 2006). So sind auf der einen Seite die Services und ihre Interaktion bezüglich der geschäftlichen Anforderungen zu definieren. Auf der anderen Seite müssen Komponenten in den Legacy Systemen identifiziert werden, die benötigte Funktionalitäten des serviceorientierten Zielsystems beinhalten (Ziemann et al., 2006).

#### 3.1.4 Sonstige Gründe

Neben den ausführlich beschriebenen Gründen gibt es weitere Auslöser für Migrationen, die im Folgenden nur kurz aufgelistet werden, wobei die Aufstellung keinen Anspruch auf Vollständigkeit erhebt:

*Auflösung von Unternehmen*

- Auseinandergehen von zuvor fusionierten Unternehmen oder die Abspaltung von einzelnen Unternehmensteilen in eigenständige Gesellschaften, z.B. durch Verkauf oder Outsourcing.

- Erzielung von Synergieeffekten innerhalb von Konzernen durch Harmonisierung von IT-Systemen und Prozessen.
- Neue gesetzliche Rahmenbedingungen wie z.B. Rechnungslegung / IAS, Basel II.

*Synergieeffekte*

*Gesetzliche Regelungen*

### 3.2 Potentiale einer Migration

Wurden bislang die Auslöser für Migrationsprojekte betrachtet, so soll nun der Nutzen derartiger Projekte beleuchtet werden. Zusammenfassend betrachtet können Unternehmen durch die Ablösung veralteter Systeme Mehrwert in drei Bereichen erzielen:

*Nutzen einer Migration*

- Produktivitätssteigerungen,
- Kostensenkungen,
- Zukunftssicherheit.

Produktivitätssteigerungen können in verschiedenen Bereichen erzielt werden. So beugt die Ablösung der Alt-systeme den Folgen eines zunehmenden Know-how-Verlustes in der IT-Abteilung vor. Auf diese Weise können Reibungsverluste und Qualitätsmängel verhindert werden, die zum einen die IT-Abteilungen belasten wür-

*Produktivitätssteigerungen*

den, zum anderen die Arbeit in den Fachbereichen. Darüber hinaus sind schlanke und durchgängige IT-Prozesse eine wesentliche Voraussetzung für eine höhere Produktivität in den IT-Abteilungen. In den Fachbereichen erhöht eine optimale und flexible IT-Unterstützung der Geschäftsprozesse die Wettbewerbsfähigkeit der Unternehmen.

*Kosteneinsparung*

Die Harmonisierung der Systemlandschaft führt zu einer Verringerung der System-, Daten- und Schnittstellenkomplexität und damit zu geringeren Betreuungskosten. Darüber hinaus werden durch die Ablösung von Altsystemen Wartungs- und Pflegeaufwände reduziert.

*TCO Senkung*

Ein erfolgreiches Migrationsprojekt kann die Total Cost of Ownership (TCO), also die Summe der Betriebskosten eines Systems, erheblich senken, wobei sich die Einsparungen auf bis zu 40 % belaufen können (Eberhardt / Rüdiger, 2006).

*Einsparpotentiale*

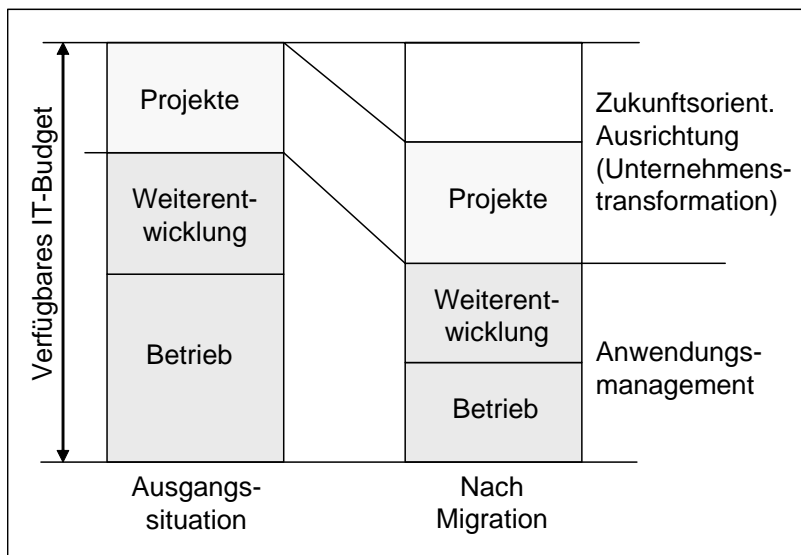
Diese hohen Einsparpotentiale belegt auch eine Studie des auf Finanzdienstleister spezialisierten Beratungshauses Celent: Bei der Migration eines Altsystems auf ein Core-Banking-System (Standardsoftware) einer weltweit agierenden Bank wurde geschätzt, dass das Unter-

nehmen durch den verringerten Wartungsaufwand und kürzere Entwicklungszyklen in der Lage sein wird, jährlich 110 Millionen Dollar einzusparen (Halbich, 2007).

Die durch die Migration eingesparten Mittel im Systembetrieb können von den Unternehmen gezielt eingesetzt werden, um z. B. strategische Investitionen zu tätigen und die zukunftsorientierte Ausrichtung des Unternehmens sicherzustellen (Ross, 2005; vgl. dazu auch Kapitel 3.1.3). Die folgende Abbildung verdeutlicht diese Möglichkeit.

*Strategische Investitionen*

Abb. 11: Aufteilung der IT-Budgets. (Quelle: In Anlehnung an Ross, 2005)





*Zukunfts-  
sicherheit*

Die Zukunftssicherheit kann durch die Schaffung von Prozessen und Systemen erhöht werden, die es dem Unternehmen ermöglichen, flexibel auf neue Anforderungen zu reagieren. Darüber hinaus minimiert eine moderne Systemlandschaft das Risiko für das Unternehmen durch eine qualitative Verbesserung der Anwendungen und Bestände.

### **3.3 Was und wann sollte migriert werden?**

*Strukturierte  
Entscheidung*

Die Entscheidung für eine Migration sollte niemals eine „Bauchentscheidung“ sein, sondern Ergebnis eines wohlstrukturierten Entscheidungsprozesses, der neben rein technischen Aspekten immer auch betriebswirtschaftliche Betrachtungen umfasst.

*Legacy  
Portfolio*

Eine erste Annäherung daran, welche Systeme überhaupt für eine Migration in Frage kommen, ist eine Einordnung der Legacy Systeme nach ihrem Wert, d.h. ihrer Bedeutung, für das Unternehmen und ihrer technischen Qualität (Seacord et al, 2003).

*Wert des  
Systems*

Zur Erstellung des Portfolios sind zunächst alle Legacy Applikationen zu inventarisieren und ihr derzeitiger und zukünftiger Wert für das Unternehmen entlang der ver-

schiedenen Geschäftsprozesse abzuschätzen. Folgende Kriterien können z. B. zur Bestimmung des Wertes verwendet werden: Beitrag zum Umsatz / Gewinn, Anzahl der unterstützten Geschäftsziele, Kundenzufriedenheit, Umfang der Nutzung.

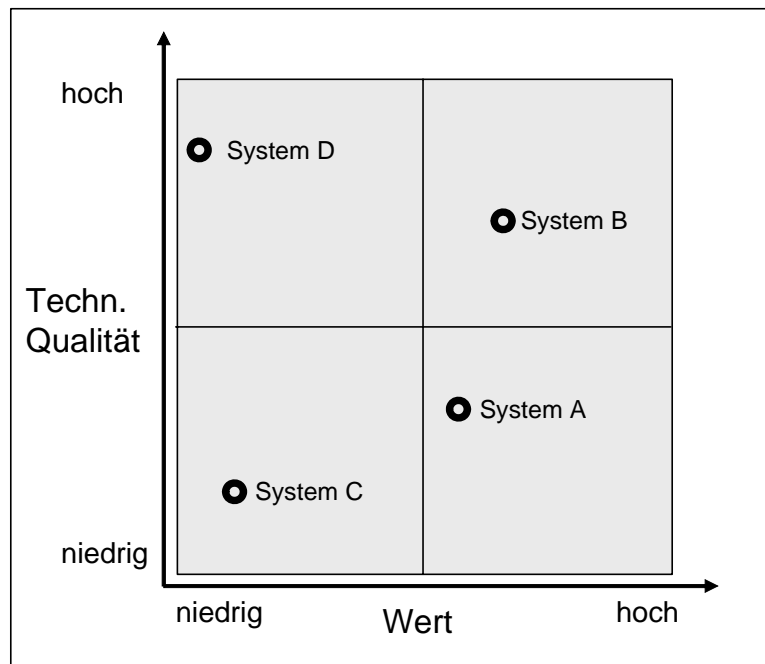
Danach ist für jede Applikation die technische Qualität zu beurteilen. Beispielkriterien zur Operationalisierung dieser Dimension sind: Häufigkeit von neuen Releases, Veränderbarkeit, Hardware- und Softwarezuverlässigkeit, Performance, Einfachheit der Bedienung usw.<sup>5</sup> Technische Qualität und Wert werden dann für alle Applikationen in das Portfolio eingetragen (vgl. Abb. 12).

*Technische  
Qualität*

---

<sup>5</sup> Andere Autoren verwenden anstelle der Dimension „technische Qualität“ die Dimension „Veränderbarkeit“ (vgl. Jacobson / Lindstrom, 1991) oder Risiko (vgl. Hunter, 2006).

Abb. 12: Legacy Portofolio (Quelle: In Anlehnung an Seacord et al, 2003, 29)



*Entscheidungsgrundlage*

Das Portfolio kann dem CIO / IT-Leiter wichtige Grundlagen für die Entscheidung liefern: So kann abgelesen werden, welche Systeme aus welchen Gründen hin zu neueren Plattformen migriert werden sollten.

*Ersetzen durch Standardsoftware*

Systeme, die einen geringen Wert und eine geringe technische Qualität haben, wie das System C in Abbildung 12, sollten durch Standardsoftware ersetzt werden. Zwei Gründe sprechen dafür:

1. Da sie nur eine geringe technische Qualität haben, sollten sie abgelöst werden, um zukünftige Folgekosten zu vermeiden.

2. Da sie nur einen geringen Wert für das Unternehmen haben, stellen sie keine unternehmenskritischen Funktionalitäten zur Verfügung. Es wird sich dabei in der Regel um Systeme der Gehaltsabrechnung o.ä. handeln, die nicht das Kerngeschäft unterstützen und für die ausreichend Standardsoftwareprodukte gibt.

Systeme, die eine hohe technische Qualität, aber nur einen geringen Wert für das Unternehmen haben, wie das System D in Abbildung 12, sollten derzeit keinerlei Maßnahmen erforderlich machen.

*Kein Handlungsbedarf*

Systeme, die eine hohe technische Qualität haben und sehr wertvoll für das Unternehmen sind, wie das System B in der Abbildung, sollten unbedingt erhalten und weiterentwickelt werden. Zunächst sollten evolutionäre Ansätze weiterverfolgt werden, bevor irgendwann eine Migration durchgeführt werden muss.

*Weiter-zuentwickelnde Systeme*

Kritisch sind Systeme, die eine geringe technische Qualität haben, aber gleichzeitig sehr wertvoll für das Unternehmen sind. Diese Systeme bedürfen der besonderen

*Kritische Systeme*

Aufmerksamkeit des IT-Leiters. Für diese Systeme sollte umgehend eine passende Migrationsstrategie ausgewählt werden (vgl. dazu Kapitel 6).

*SRAH* Ein strukturiertes Vorgehen zur Beurteilung der Situation einzelner Systeme bietet das vom amerikanischen Verteidigungsministerium (United States Department of Defence, DoD) entwickelte Software Reengineering Assessment Handbook (SRAH). Ziel des darin beschriebenen Beurteilungsprozesses ist es, zu bestimmen, wie mit einem Altsystem zu verfahren ist. SRAH verwendet verschiedene Kriterien und definiert ein strukturiertes Vorgehen für die Bewertung der einzelnen Faktoren aus den Bereichen „Technik“, „Wirtschaftlichkeit“ und „Management“ (Alyokhin, 2005).

*Technische  
Beurteilung*

Bei der technischen Beurteilung ist zu prüfen, ob ein System grundsätzlich für eine Migration in Frage kommt und welche Alternativsysteme zur Verfügung stehen. Beispielhafte Fragen in diesem Zusammenhang sind etwa:

- Gibt es für System noch Support vom Hersteller?
- Ist noch genügend Know-how zum Einsatz des Systems vorhanden, so dass im Problemfall

schnell reagiert und eine Lösung gefunden werden kann?

- Läuft das System auch auf aktueller Hardware, falls kein gleichwertiger Ersatz für defekte Komponenten gefunden werden kann?

Bei der betriebswirtschaftlichen Betrachtung werden, aufbauend auf der technischen Beurteilung, Punkte wie Kostenschätzung und Risikoanalyse behandelt.

*Betriebswirtschaftliche Betrachtung*

Basierend auf den Ergebnissen dieser beiden Beurteilungen werden im Bereich „Management“ Aspekte wie Kundenwirkung, strategische Entscheidung für einen bestimmten Hersteller u.a. in die Entscheidung einbezogen.

### **3.4 Kosten der Migration**

Die Migration eines Altsystems ist immer mit erheblichen Kosten für das Unternehmen verbunden. Ein sehr großer Anteil dieser Kosten sind Personalkosten für interne Mitarbeiter und externe Consultants.<sup>6</sup> Personalkosten fallen an für:

*Personalkosten*

Rekonstruk-  
tion

- **Analyse des Altsystems**

Legacy Systeme sind häufig nicht oder nur unzureichend dokumentiert. Die im System implementierten fachlichen Funktionen müssen daher häufig mühsam nachvollzogen und dokumentiert werden.

Gewöhnlich ist es nur schwer möglich, die im System implementierten Prozesse direkt aus dem Quellcode zu extrahieren. Daher muss mit erheblichem personellen Aufwand von Anwendern und Betreuern des Altsystems versucht werden, die Prozesse zu rekonstruieren und zu modellieren.

- **Anforderungsanalyse**

Grundsätzlich sollten geänderte Anforderungen erfasst werden, damit das neue System dem heutigen Zustand entspricht. Dazu sollten unbedingt auch Mitarbeiter aus den betroffenen Fachabteilungen eingebunden werden. Ballast in Form ungenutzter Funktionalitäten und Komponenten sollte abgeworfen werden und dafür neue, bislang nicht vorhandene, aber durchaus gewünschte Funktionalitäten sollten identifiziert werden.

Abgleich  
Anforderun-  
gen

---

<sup>6</sup> Zur Einbindung externer Consultants vgl. Kapitel 5.2 „Kritische Erfolgsfaktoren für Migrationsprojekte“.

- **Entwicklung des fachlichen Feinkonzeptes**

Die Erstellung des Pflichtenheftes, die Prüfung der technischen Machbarkeit und die weitere Detailplanung des Projekts sind in der Konzeptionsphase die wichtigsten Aufgaben, die einen nicht zu unterschätzenden Personalaufwand mit sich bringen.

*Pflichtenheft-  
erstellung*

- **Erstellung des neuen Systems**

Dazu gehört nicht nur die eigentliche Implementierung, sondern auch der komplette Systemtest (incl. Testfallentwicklung) und der Pilotbetrieb sowie die Integration des neuen Systems in die bestehende Systemlandschaft des Unternehmens.

*Implementie-  
rung*

- **Änderungen am Altsystem**

Um einen Parallelbetrieb mit dem neuen System während der Migration zu ermöglichen, müssen unter Umständen umfangreiche Änderungen am Altsystem vorgenommen werden. Diese können von einfachen Wrappern (Ummantelungen) über komplexe Umstrukturierungen bis hin zu einem Service in einer SOA reichen.

*Anpassung  
Altsystem*



*Schulung*

- **Schulungen der Mitarbeiter**

Wenn ein neues System eingeführt wird, müssen sowohl die Mitarbeiter aus dem IT-Bereich, dem Help Desk als auch in den betroffenen Fachabteilungen geschult werden. Dafür ist zum einen ein entsprechendes Schulungskonzept zu entwickeln, zum anderen ist der Personalaufwand auf Seiten sowohl der Trainer als auch der zu schulenden Mitarbeiter zu berücksichtigen.

*Migration*

- **Einführung und Rollout**

Im Rahmen der Einführung und des Rollouts müssen die Systeme und Daten endgültig migriert und den Anwendern zur Verfügung gestellt werden. Wichtig ist insbesondere auch die Anwenderbetreuung in der Übergangsphase.

*Überstunden-  
den-  
zuschläge*

Viele Migrationen, insbesondere Datenmigrationen, werden außerhalb der typischen Arbeitszeiten und an Wochenenden durchgeführt, um zu vermeiden, dass die Migration das operative Geschäft stört. Je nach Vertragsgestaltung der Mitarbeiter kann dieses zu erhöhten Personalkosten führen, wenn Überstunden und Wochenendarbeit zusätzlich entlohnt werden müssen.

- **Dokumentation**

Ein häufig vernachlässigter, aber sehr wichtiger Aspekt ist die Dokumentation. Da es aufgrund der schnellen Weiterentwicklungen in der IT sehr wahrscheinlich ist, dass die heute konstruierten Systeme demnächst auch zu Altsystemen werden und dann auch migriert werden müssen, sollte großer Wert auf eine lückenlose, konsistente Dokumentation gelegt werden. Dadurch kann bei späteren Migrationen erheblicher Projektaufwand eingespart werden (Kroha / Rosenhainer, 2006).

*Dokumen-  
tation*

- **Projektmanagement**

Aufgrund der Komplexität von Migrationsprojekten und der Bedeutung, die ein fundiertes und systematisches Projektmanagement für den Projekterfolg hat, sollte das Projektmanagement unbedingt gesondert als eigenständiger Kostenfaktor in der Planung betrachtet werden. Typischerweise empfiehlt es sich, dafür einen prozentualen Aufschlag von 10-15 % vom Gesamtpersonalaufwand anzusetzen.

*Projekt-  
management*

*Qualitäts-  
sicherung*

- **Qualitätssicherung**

Auch die Qualitätssicherung stellt einen wesentlichen Kostenfaktor dar, der häufig in der Projektplanung unterschätzt wird. Da es sich dabei um einen kontinuierlichen Prozess handelt, der parallel zum Projektmanagement läuft, sollte ein ähnlicher prozentualer Anteil, also ca. 10-15% des Gesamtpersonalaufwands, angesetzt werden.

*Aufteilung  
Personal-  
aufwand*

Die folgende Tabelle zeigt exemplarisch die Aufteilung des Personalaufwands (in Personentagen) für ein Migrationsprojekt. Das zugrunde liegende Beispiel bezieht sich auf die Migration von Basissoftwarekomponenten (Serverdienste, Standardsoftware, Bürokommunikation, Dokumente, Makros) auf Server und Arbeitsplatzsystemen in einer mittelgroßen Behörde.

Tab. 2: Personalaufwand für ein Migrationsprojekt  
(Quelle: BMI, 2006, S. 13).

Aufwand in Personentagen		
Aktivität	Intern	Extern
Planung	162	220
Realisierung	41	30
Einsatz	20	30
Dokumentation	50	50
<b>Summe Aufwand</b>	<b>273</b>	<b>330</b>
Projektmanagement <sup>7</sup>	41	50
Qualitätssicherung	41	50
Anwenderbetreuung	250	100
<b>Gesamtaufwand</b>	<b>605</b>	<b>530</b>

Darüber hinaus fallen Sachkosten in folgenden Bereichen an:

- **Lizenzen**

Werden neue Softwareprodukte verwendet, so fallen für diese in der Regel Lizenzkosten an, wenn sie bislang im Unternehmen nicht eingesetzt wurden.

*Lizenzkosten*

<sup>7</sup> Für Projektmanagement und Qualitätssicherung wurde jeweils ein pauschaler Aufschlag von 15% auf den Aufwand (ohne Anwenderbetreuung) gerechnet.

*Hardware*

- **Hardware**

Häufig wird im Zuge der Migration auch die technische Basis ausgetauscht. In diesem Fall sind die Kosten für Server, Clients und Infrastrukturkomponenten in die Berechnung einzubeziehen. Sollte kein kompletter Austausch stattfinden, so wird häufig zumindest in Kapazitätserweiterungen der Speichermedien oder der Rechnerleistung investiert.

*Werkzeuge*

- **Werkzeuge zur Unterstützung der Migration**

Eine Migration ist nicht ohne unterstützende Werkzeuge durchführbar. Da diese in der Regel nicht im Unternehmen vorhanden sind, müssen sie entweder entwickelt oder gekauft werden.

*Allgemeine  
Projekt-  
risiken*

### 3.5 Risiken einer Migration

Zu den Risiken einer Migration gehören zunächst einmal ganz allgemein alle Risiken, die die Entwicklung und Einführung großer Systeme mit sich bringt. Diese werden in der folgenden Tabelle nur überblicksmäßig aufgelistet, aber nicht ausführlich erläutert, da sie nicht migrationspezifisch sind.

Tab. 3: Typische Projektrisiken (Quelle: In Anlehnung an Houdek et al, 2006)

Entwicklungsrisiken	
	Einführungsrisiko
	Applikations-Entwicklungs-Risiko
	Materialzulieferungs-Risiko
Management-Risiken	
	Projektleitungs-Risiko
	Planungs-Risiko
	Informations- und Kommunikations-Risiko
Soziale Risiken	
	Motivations-Risiko
	Politische Risiken
	Mitarbeiter-Risiken

Neben diesen allgemeinen Projektrisiken gibt es aber auch Risiken, die spezifisch für Migrationsprojekte sind. Eine Untersuchung zum Thema Daten-Migration zeigte, dass in 83% aller Projekte Probleme auftraten. Als häufigstes Problem wurden unerwartete Downtimes, d. h. Zeiten, in denen das System nicht funktioniert, angegeben, gefolgt von technischen Kompatibilitätsproblemen, Datenkorruption, Performance-Problemen und Datenverlust (Softek, 2006, 6).

*Downtime*

### 3

#### *Software-Migration - Notfall oder strategische Neuorientierung ?*

##### *Finanzielle Gefährdung*

Wird berücksichtigt, dass die Kosten einer Downtime sich auf einige Millionen Dollar pro Stunde belaufen können, so kann eine ungeplante Downtime unter Umständen ganz erhebliche Auswirkungen auf die finanzielle Situation des Unternehmens haben. Vor diesem Hintergrund sollte dem Risikomanagement bei Migrationsprojekten unbedingt von Anfang an Beachtung geschenkt werden.

Darüber hinaus sollten folgende Risiken im Auge behalten werden:

##### *Fachliche Komplexität*

- Die Komplexität der im Altsystem enthaltenen Fachlichkeit wird unterschätzt und Teile der Geschäftsregeln gehen verloren.

##### *Prozesse noch „unreif“*

- Organisatorische Risiken, da bei der Umstellung möglicherweise neue Geschäftsprozesse eingeführt werden, die noch „reifen“ müssen.

##### *Widerstände*

- Widerstände der Mitarbeiter und Systemeigner, da die Umstellung oder Ablösung des Systems Machtverluste für einzelne Mitarbeiter oder Organisationseinheiten bedeuten kann.

Nachdem die Entscheidung für ein Migrationsprojekt gefallen ist, sollte jedoch nicht unmittelbar in „operative Hektik“ ausgebrochen werden. Vielmehr ist es wichtig, eine wohlüberlegte und strukturierte Vorgehensweise zu wählen, die gegebenen Rahmenbedingungen zu analysieren, die kritischen Erfolgsfaktoren frühzeitig zu berücksichtigen und ein gezieltes Projekt- und Risikomanagement aufzubauen.





## 4 Migration - mehr als nur ein technisches Thema

Ein Migrationsprojekt sollte niemals nur aus einem rein technischen Blickwinkel betrachtet werden. Zwar ist es unbedingt notwendig, die technische Migration zu beherrschen, für den Gesamterfolg des Projektes stellt dieses jedoch nur einen Teilaspekt dar.

*Technik nur  
Teilaspekt*

Je größer das Umstellungsprojekt ist, desto wichtiger sind auch andere Einflussfaktoren wie beispielsweise das technische und / oder organisatorische Umfeld.

### 4.1 Technische Rahmenbedingungen

Die Möglichkeiten der Migration werden ganz erheblich von der Struktur des Altsystems beeinflusst. Brodie / Stonebraker (1995) unterscheiden drei Typen von Altsystemen nach ihrer Strukturierung:

*Zerlegbarkeit*

- Voll zerlegbare Systeme
- Teilweise zerlegbare Systeme
- Nicht zerlegbare Systeme.

# 4

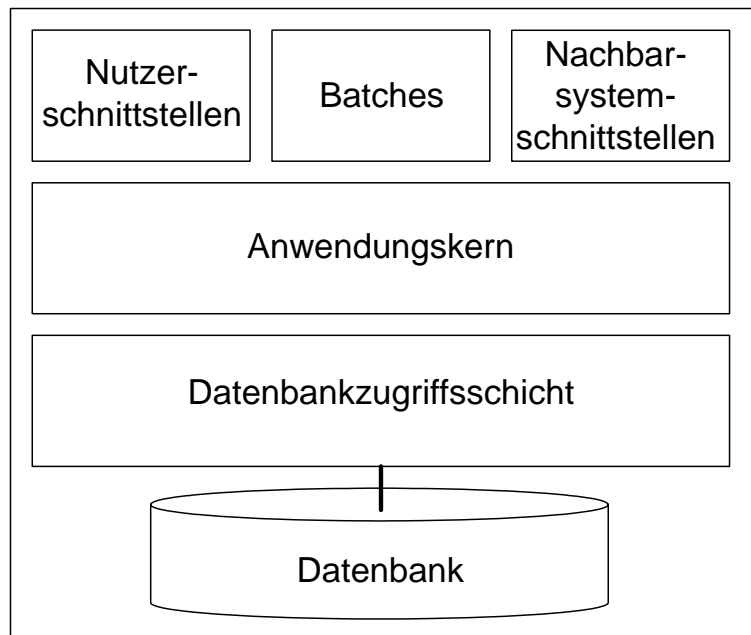
## Voll zerlegbare Systeme

Migration - mehr als nur ein technisches Thema

Voll zerlegbare Systeme sind gut strukturiert und können sowohl in vertikale Schichten als auch horizontal in fachliche Module unterteilt werden. Beispiele für fachliche Module sind die Kundenverwaltung, die Vertragsverwaltung usw.

Vertikal kann ein gut strukturiertes Altsystem in drei Schichten aufgeteilt werden, die in der folgenden Abbildung dargestellt sind:

Abb. 13: Schichten voll zerlegbarer Systeme (Quelle: In Anlehnung an Softwarekompetenz, 2007).



Der Anwendungskern ist weitgehend in eine Reihe von fachlichen Modulen mit jeweils zugehörigen eigenen Schnittstellen zu Nachbarsystemen und zum Nutzer aufteilbar. Die Module verwenden häufig eine gemeinsame Datenbankzugriffsschicht, agieren ansonsten aber weitgehend unabhängig voneinander, d.h. es existieren keine bzw. kaum gegenseitigen Funktionsaufrufe. Eine systematische Zerlegbarkeit wie in Abbildung 13 dargestellt, findet sich jedoch bei Altsystemen sehr selten. Typischer sind teilweise oder gar nicht zerlegbare Systeme.

Im Gegensatz zu den komplett zerlegbaren Systemen ist bei den teilweise zerlegbaren Systemen der Anwendungskern monolithisch, d.h. nicht mehr in einzelne Komponenten zerlegbar. Nur die Benutzer- und Nachbarsystemschnittstellen lassen sich noch als separate Bestandteile identifizieren.

Bei nicht zerlegbaren Systemen sind weder Schnittstellen noch der Anwendungskern in Module unterteilt, das Altsystem ist ein Monolith.

Darüber hinaus beeinflussen folgende (technische) Rahmenbedingungen ein Migrationsprojekt:

- Die Qualität der heutigen Datenbanken und die Datenintegrität.
- Die Anzahl und Komplexität der Schnittstellen zu anderen internen und externen Systemen.
- Der Umfang und die Aktualität der verfügbaren Systemdokumentation.

## 4.2 Organisatorische Rahmenbedingungen

*Einbindung  
Fachbereich*

Geeignete organisatorische Rahmenbedingungen bilden eine wesentliche Voraussetzung für das Gelingen eines Migrationsprojektes. Wichtig ist dabei insbesondere die frühzeitige Einbindung aller betroffenen Fach- / Geschäftsbereiche.

*Mitwirkung  
als Chance*

Eine frühzeitige Einbindung der betroffenen Fachbereiche ist nicht nur wichtig, um die Akzeptanz zu sichern, sondern insbesondere auch, um auf das Know-how der Mitarbeiter zurückgreifen zu können. Da die Mitarbeiter in den Fachbereichen die Altsysteme oft jahrelang betreut und genutzt und mit ihnen die Unternehmensprozesse abgewickelt haben, sprich das Unternehmen am Leben gehalten haben, wissen sie am besten, welche Daten in

welcher Qualität für die Fortführung der Prozesse benötigt werden und wo die entsprechenden Datenquellen und Verantwortlichen zu finden sind. Dabei ist es wichtig, dass die Mitwirkung von den Beteiligten nicht als notwendige Pflichtübung, sondern als Chance zum Mitgestalten betrachtet wird (Fein, 1998, S. 59).

Für die Akzeptanzsicherung in den Fachbereichen und die Motivation des Migrationsteams ist darüber hinaus eine deutlich spürbare Unterstützung durch das Management wichtig. Nur wenn das Management hinter dem Projekt steht und allen Mitarbeitern die Bedeutung des Projektes für das ganze Unternehmen kommuniziert, werden die Mitarbeiter bereit sein, die Veränderungen mit zu tragen.

*Einbindung  
Management*



## 5 Herausforderungen der Migration

Damit eine Migration erfolgreich verlaufen kann, sollten bereits zu Projektbeginn sowohl mögliche Problemfelder als auch die kritischen Erfolgsfaktoren analysiert und in den Planungen berücksichtigt werden.

### 5.1 Problembereiche einer Migration

Es gibt eine Reihe von Gegebenheiten, die eine Migration erschweren bzw. unter Umständen erheblich verzögern und / oder verteuern können. Diese sollten rechtzeitig untersucht und, sofern möglich, hierfür Abhilfe geschaffen werden. Im Folgenden werden zwei grundlegende Problembereiche vorgestellt:

- **Fehlende Dokumentation des Altsystems**

Altsysteme sind häufig kaum dokumentiert. In ihnen befindet sich über die Jahre hinweg gewachsenes Wissen über die Geschäftsprozesse, das durch fortwährende Änderungen am System weiterentwickelt wurde. Das Wissen ist jedoch häufig nur im Code implementiert, eine dazugehörige Be-

*Mangelhafte  
Dokumen-  
tation*



*Mühsame  
Rekonstruk-  
tion*

schreibung existiert nicht. Damit dieses Wissen bei der Ablösung des Systems nicht verloren geht, muss es nachträglich dokumentiert werden.

Da die in einem System implementierten Prozesse jedoch gewöhnlich nur schwer direkt aus dem Quellcode extrahiert werden können, müssen sie in der Regel mühsam in Workshops mit den Anwendern und ggf. Entwicklern rekonstruiert werden (Wölfle et al., 2006).

*Fehlendes  
Know-how*

Erschwerend kommt hinzu, dass viele der ursprünglichen Programmierer und Experten, die an der Entwicklung des Systems beteiligt waren, das Unternehmen schon verlassen oder den Aufgabenbereich gewechselt haben und als Ansprechpartner nicht mehr zur Verfügung stehen.

*Verständnis-  
probleme*

In engem Zusammenhang mit der beschriebenen Situation stehen Verständnisprobleme, die durch lückenhafte, inkonsistente oder gar fehlende Dokumentation verursacht werden (Kroha / Rosenhainer, 2006). Der Auftraggeber bzw. Nutzer kann seine Wünsche / Anforderungen an das zukünftige System häufig nur teilweise oder ungenau be-

schreiben. Diese Beschreibung kann von den Entwicklern nur partiell nachvollzogen werden, was bei der Entwicklung zu Implementierungen führt, die die eigentlichen Anforderungen des Auftraggebers nur unzureichend abdecken.

- **Mängel im Altdatenbestand**

Die Daten in den Altsystemen weisen meist eine Reihe von Defiziten auf. Typische Mängel sind beispielsweise (vgl. dazu Fischbach et al, 2002):

- Für identische Sachverhalte finden sich verschiedene Schreibweisen.
- Sonderzeichen sind als Platzhalter in Felder eingefügt worden.
- Auch identische Attribute haben in unterschiedlichen Altsystemen und im Zielsystem inkongruente Wertemengen.
- Die Daten in Altsystemen sind aus aktueller, dem neuen Zieldatenmodell entsprechender Sicht meist unvollständig.
- Unterschiedliche Quellen spiegeln unterschiedliche Bearbeitungsstände von Dokumenten beziehungsweise Vorgängen wider, wobei es „Dateileichen“ gibt, die jahrelang nicht gepflegt wurden.

*Qualität  
Altdaten*

- Gleiche Anwendungen arbeiten an verschiedenen Standorten mit unterschiedlichen Datenformaten, die sich aus oft undokumentierten Abweichungen zu einem inzwischen obsoleten Schema ergeben.

## 5.2 Kritische Erfolgsfaktoren für Migrationsprojekte

Kritische Erfolgsfaktoren beschreiben jene Verhaltens- und Vorgehensweisen, die für eine erfolgreiche Migration besonders relevant sind. Für Migrationsprojekte wurden folgende kritische Erfolgsfaktoren identifiziert:

- **Systematische Planung**

Migrationsprojekte sind häufig sehr komplex und viele Unternehmen unterschätzen den tatsächlichen Aufwand, insbesondere für die Datenmigration (Albrecht, 2002, Fischbach et al. 2002). Eine systematische und detaillierte Projektplanung, die vorzugsweise die Datenmigration als eigenständiges Teilprojekt berücksichtigt, ist eine Grundvoraussetzung für ein erfolgreiches Projektmanagement.

*Planung*

- **Effiziente und effektive Projektleitung**

Der Projektleiter sollte über umfangreiche Projektmanagementenerfahrungen, am besten im Zusammenhang mit Migrationen, verfügen. Aufgrund seiner Erfahrungen sollte er in der Lage sein, ein effektives Ressourcen Management bezüglich Time, People und Budget aufzubauen und geeignete Steuerungsinstrumente zu nutzen.

*Projektleitung*

- **Einsatz von Werkzeugen**

Komplexe Migrationsprojekte sind nur unter Einsatz von Werkzeugen effizient und effektiv im gegebenen Zeit- und Budgetrahmen zu realisieren. Benötigt werden in der Regel folgende Werkzeugtypen:

*Migrationswerkzeuge*

- Eigentliche Konvertierungswerkzeuge
- Werkzeuge zur Testunterstützung
- Allgemeine Werkzeuge wie Debugger, Datenbank-Laden / -Entladen.

Am Markt gibt es Tools verschiedener Hersteller, die den Migrationsprozess in unterschiedlichen Phasen und in unterschiedlicher Qualität unterstützen (Uhlig, 2006). Der teilweise von den Herstellern erweckte Eindruck, dass Migrationsprojek-

*Auswahl geeigneter Tools*

*Projekt-  
Know-how*

*Spezialwis-  
sen kaufen*

te durch den Einsatz ihrer Werkzeuge quasi „nebenbei“ erledigt werden könnten, ist jedoch irreführend (Borchers, 1997, 93). Die Auswahl und Nutzung geeigneter Tools für das jeweilige Projekt ist somit eine wesentliche Voraussetzung für den Projekterfolg.

- **Know-how aus ähnlichen Projekten**

Migrationsprojekte sind wegen ihrer hohen technischen Spezialisierung für ein Outsourcing oder zumindest für die Einbindung von externen Spezialisten prädestiniert. Sie setzen anspruchsvolle Werkzeuge und Spezialisten, die damit umgehen können, voraus. Außerdem ist es von Vorteil, wenn der Migrationsverantwortliche und die wichtigsten Mitarbeiter Erfahrungen mit etlichen Projekten dieser Art gesammelt haben.

Über derartig umfangreiche Erfahrungen verfügen jedoch nur die wenigsten Produkthersteller und noch weniger die Anwenderbetriebe (Sneed et al., 2005, 243). Es empfiehlt sich daher, Migrationsprojekte outzusourcen oder spezialisierte Consultants in die Projekte einzubinden (Brodie / Stonebaker, 1995).

- **Ein Team, das die alten und die neuen Anwendungen und Systemumgebungen gut kennt**

Das Projektteam sollte zum einen Mitarbeiter umfassen, die die Altsysteme sowohl von der Anwendung (d.h. in der Regel Mitarbeiter aus den Fachabteilungen) als auch von der technischen Realisierung (d.h. in der Regel Mitarbeiter aus der IT-Abteilung) her gut kennen. Zum anderen sollten technische Experten zum Team gehören, die die zukünftige Betreuung des Neusystems übernehmen (Albrecht, 2002).

*Internes  
Know-how*

- **Profunde Kenntnisse der alten und neuen Daten-Objekte, -Strukturen, -Umsetzungsregeln**

Die Datenintegrität ist im Rahmen von Migrationsprojekten sehr ernst zu nehmen, da die Daten den eigentlichen Unternehmenswert darstellen. Daher ist zum einen von zentraler Bedeutung, dass „echte“ Spezialisten für die Datenmigration zur Verfügung stehen, zum anderen ist eine enge Zusammenarbeit zwischen Datenmodellexperten für das Alt- und das Neusystem unabdingbar.

*Spezialisten  
für Daten-  
migration*

*Tests kritisch*

- **Einhaltung eines rigiden Testkonzepts**

Der Erfolg oder Misserfolg eines Migrationsprojektes hängt zum überwiegenden Teil von der Durchführung strenger Regressionstests ab (vgl. dazu Kap. 7.5).

*Regressionstest*

Der Regressionstest ist die mit Abstand ressourcenintensivste Aufgabe eines Migrationsprojektes. Zum einen werden deutlich über 50% der Personalressourcen benötigt, zum anderen werden die höchsten Anforderungen an die Rechnerumgebung gestellt.

*Testkonzept  
frühzeitig  
entwickeln*

Alleine der Plattenbedarf für Testbestände beträgt insgesamt gesehen ein Mehrfaches des Produktionsbestands (Borchers, 1997, 92). Aus diesem Grund sollte bereits frühzeitig ein umfassendes Testkonzept erarbeitet werden, das dann strikt einzuhalten ist.

*Eigentliche  
Migration  
testen*

Darüber hinaus ist es außerordentlich wichtig, die eigentliche Migration im Voraus zu testen, um z.B. abschätzen zu können, ob die eingeplante Zeit tatsächlich ausreicht. Dazu empfehlen sich Testwochenenden, an denen das Migrationsszenario

durchgespielt werden kann, ohne den operativen Betrieb zu gefährden. Auf der Basis der neuen Stände können dann die Regressionstests durchgeführt werden.





## 6 Migrationsstrategien

Für die Migration von Systemen sind verschiedene Migrationsstrategien entwickelt worden. Welche dieser Strategien für das jeweilige Projekt geeignet ist, muss im Einzelfall vor dem Hintergrund der unternehmensspezifischen Ausgangslage geklärt werden (Erdle, 2005, S 22).

*Migrationsstrategien*

Im Folgenden werden zwei Migrationsstrategien<sup>8</sup>, die inkrementelle Migration und die Stichtagsumstellung vorgestellt und deren Vor- und Nachteile herausgearbeitet. Im Anschluss daran werden allgemeine Anforderungen an eine Migration aufgezeigt.

### 6.1 Inkrementelle Migration

Die inkrementelle Migration basiert auf einem Stufenkonzept. Nach dem Prinzip des „Divide & Conquer“ wird die Migration in viele kleine Schritte aufgeteilt, die iterativ durchlaufen werden.

*Kleine Schritte*

---

<sup>8</sup> Die Beschreibung basiert auf den entsprechenden Wissensbausteinen unter [www.softwarekompetenz.de](http://www.softwarekompetenz.de).

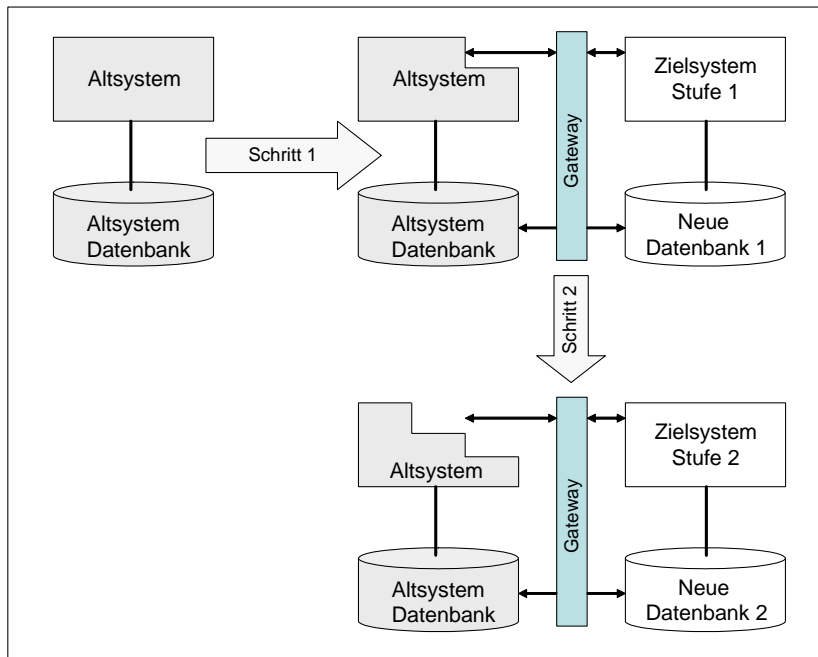
*Gateways  
als Vermittler*

So wird das Zielsystem in kleinen Schritten, beispielsweise Komponente für Komponente, aufgebaut und zunächst parallel zum Altsystem betrieben. Beide Systeme arbeiten an bestimmten Stellen über so genannte Gateways zusammen. Diese treten als Vermittler zwischen Komponenten des Alt- und des Zielsystems auf und setzen Funktionsaufrufe oder zu übermittelnde Daten in das jeweils auf der anderen Seite erwartete Format um. Das neue System übernimmt dann Schritt für Schritt mehr Funktionsumfang des Altsystems, bis das Altsystem vollständig abgelöst ist (vgl. Abb.14).

*Voraussetzung:  
Zerlegbarkeit*

Eine inkrementelle Migration setzt zwingend voraus, dass das Altsystem zerlegbar ist, da sonst eine modulweise erfolgenden Ablösung der Komponenten nicht realisierbar ist. Wie in Kapitel 4.1 dargestellt, gibt es jedoch auch Systeme, die nur teilweise oder sogar gar nicht zerlegbar sind. In diesen Fällen müssen Zwischenschritte unternommen werden, um ein inkrementelles Vorgehen zu ermöglichen.

Abb. 14: Inkrementelle Migration. (Quelle: Softwarekompetenz, 2007)



Zunächst wird der „einfachste“ Fall, die inkrementelle Migration bei komplett zerlegbaren Systemen beschrieben. Im Anschluss daran wird kurz auf die Möglichkeiten einer inkrementellen Migration bei teilweise bzw. nicht zerlegbaren Systemen eingegangen.<sup>9</sup>

*Komplett  
zerlegbare  
Systeme*

<sup>9</sup> Eine ausführliche Darstellung der Vorgehensweisen findet sich unter [www.softwarekompetenz.de](http://www.softwarekompetenz.de)

*Modulweise Migration*

Bei komplett zerlegbaren Systemen kann der Anwendungskern modulweise migriert werden. Problematisch ist jedoch der Zugriff auf die Datenbank, da sich die Daten in der Datenbank in der Regel nicht auf die Module aufteilen lassen, so dass verschiedene Module auf dieselben Daten zugreifen. Je nach Zeitpunkt der Umstellung der Datenbasis werden drei Ansätze unterschieden:

- Vorwärts-Migration
- Rückwärts-Migration und
- Allgemeine Migration

*Vorwärts-Migration*

Bei der so genannten Vorwärts-Migration (auch „Database First“ Ansatz genannt) werden zuerst die Datenbestände vollständig auf ein modernes System migriert, bevor die Anwendung und die Benutzeroberfläche migriert werden. Für den Zugriff der Komponenten des Alt-systems auf das neue Datenbanksystem werden so genannte Forward Gateways verwendet. Nach vollständiger Migration der Anwendungskomponenten und Benutzeroberflächen wird das Forward Gateway deaktiviert.

*Rückwärts-Migration*

Bei der so genannten Rückwärts-Migration (auch „Database Last“ genannt) werden nach und nach die einzelnen Komponenten des Alt-systems auf das neue System mig-

riert und erst zuletzt die Datenbasis. Für den gleichzeitigen Zugriff von Komponenten des Alt- und des Neusystems auf den Datenbestand müssen alle Komponenten des Neusystems den Zugriff über Reverse Gateways abwickeln. Der letzte Schritt ist die Migration des Datenbanksystems auf die neue Plattform und die Abschaltung der Reverse Gateways.

Bei der so genannten Allgemeinen Migration werden Datenbestände und Anwendungskern gemeinsam migriert. Die Komponenten des Zielsystems müssen ggf. über einen Reverse Gateway auf die Daten des Altsystems zugreifen und umgekehrt müssen die Module des Altsystems über einen Forward Gateway auf die Daten des Zielsystems zugreifen.

Bei teilweise zerlegbaren Systemen ist die Modularisierung des Anwendungskerns nicht möglich, nur die Benutzerschnittstellen können als einzelne Module betrachtet werden. In diesem Fall wird ein Application-Gateway, das aus einem Forward- und einem Reverse-Gateway sowie einem Coordinator besteht, zwischen Anwendungskern und Nutzer- bzw. Nachbarsystemschnittstellen gesetzt. Bei der Migration werden nun iterativ die Schnittstellen des Altsystems in Schnittstellen des Zielsystems umgesetzt.

*Allgemeine Migration*

*Teilweise zerlegbare Systeme*

*Schrittweise  
Implementierung*

Die den Schnittstellen zugeordneten Teile des Anwendungskerns werden parallel dazu neu implementiert. Vom Vorgehen her ist es unerheblich, ob zuerst die Schnittstelle und dann die zugeordnete Business Logik migriert wird oder umgekehrt. Allerdings ist es wichtig, die den Schnittstellen zugeordneten Funktionen in logisch zusammengehörende Komponenten aufzuteilen und entsprechend zu implementieren. Nur so kann sichergestellt werden, dass die im Altsystem existierenden internen Abhängigkeiten, die für die Nicht-Zerlegbarkeit verantwortlich waren, nicht ins Zielsystem übernommen werden.

*Monolithische  
Anwendungssysteme*

Bei monolithischen Anwendungssystemen sind weder Schnittstellen noch Anwendungskern in Module unterteilt. In diesem Fall gibt es zwei Ansätze für eine inkrementelle Migration.

*Teilweise  
Renovierung*

- **Teilweise Renovierung und Restrukturierung des Altsystems**

Durch eine teilweise Renovierung und Restrukturierung wird das System zerlegbar gemacht. Sobald die Renovierung vollendet und das System zerlegbar ist, kann nach den vorstehend beschriebenen Migrationsstrategien für zerlegbare bzw.

teilweise zerlegbare Systeme vorgegangen werden.

- **Gateway oberhalb der Schnittstellen**

Bei dieser Vorgehensweise wird ein Gateway oberhalb der Schnittstellen platziert und somit das System gekapselt. Das Altsystem wird als Blackbox betrachtet.

*Altsystem  
als Blackbox*

## 6.2 Big Bang Umstellung

Unter einem "Big Bang"-Ansatz, auch Stichtagsumstellung oder „Cold Turkey“ genannt, versteht man eine Vorgehensweise, bei der das komplette System auf einmal umgestellt wird. Parallel zum Betrieb des Altsystems wird das neue System entwickelt und getestet. Sind alle notwendigen Tests durchgeführt und bestanden, wird in einem letzten Schritt, dem „Big Bang“, das Altsystem deaktiviert und das neue System übernimmt die Arbeit (vgl. Abb. 15).

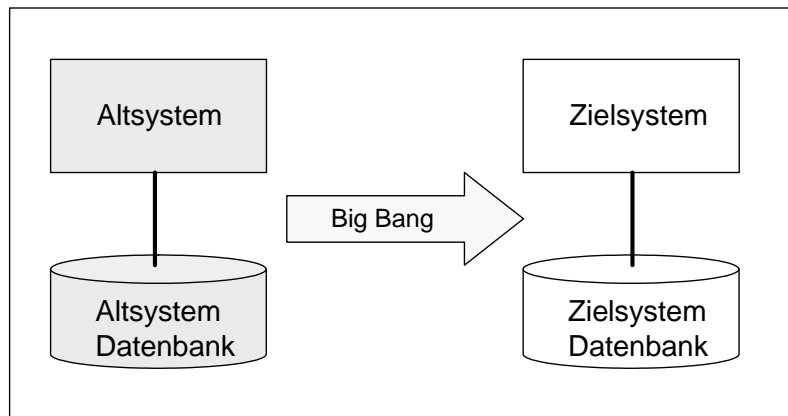
*Stichtags-  
umstellung*

Wie nachfolgend erläutert, ist die Stichtagsumstellung mit großen Risiken verbunden. Aus diesem Grund empfiehlt es sich, bei einer Stichtagsumstellung zunächst beide Systeme parallel zu betreiben.

*Parallelbe-  
trieb ratsam*



Abb. 15: Big Bang Umstellung. (Quelle: In Anlehnung an Softwarekompetenz, 2007)



### 6.3 Vor- und Nachteile der Migrationsstrategien

*Keine generelle Überlegenheit*

Sowohl die inkrementelle Migration als auch der Big Bang Ansatz haben sowohl Vor- als auch Nachteile. Eine generelle Aussage, welcher Ansatz besser ist, ist daher nicht möglich. Vielmehr sollte das System an sich sowie die Ausgangssituation im Unternehmen und im Projekt berücksichtigt werden und dann auf dieser Basis die passende Strategie ausgewählt werden.

Im Folgenden werden die Vor- und Nachteile der beiden Ansätze beschrieben, um Argumente für die Entscheidungsfindung zu liefern.

Die Strategie der inkrementellen Migration bringt folgende Vorteile mit sich:

- **Reduzierung der Komplexität**

Das Vorgehen in kleineren Schritten reduziert die Komplexität jedes einzelnen Schrittes und steigert die Erfolgsaussichten. Wenn ein einzelner Schritt scheitert, muss nur dieser wiederholt werden. Das Projekt insgesamt wird dadurch wenig gefährdet.

*Geringere Komplexität*

- **Vergleichsweise geringes Risiko**

Da immer nur einzelne Komponenten umgestellt werden, kann nach jedem Schritt getestet werden, ob sich das System den Wünschen entsprechend verhält. Sollte ein Fehler auftreten, so ist die Ursachenforschung relativ einfach, da bekannt ist, wo gesucht werden muss. Das Risiko wird dadurch möglichst klein gehalten.

*Geringeres Risiko*

- **Streckung der Investitionen über längeren Zeitraum**

Jede Umstellungsstufe für sich hat einen geringeren Umfang als die Gesamtumstellung. Die Investition für die Gesamtumstellung kann somit über einen längeren Zeitraum gestreckt werden.

*Streckung Investition*

*Zusätzliche  
Kosten*

Diesen Vorteilen stehen folgende Nachteile gegenüber:

- **Zusätzliche Kosten durch Gateways**

Das Stufenkonzept setzt voraus, dass die schon vorhandenen Teile des Zielsystems auf die alte Applikation und deren Datenbestand zurückgreifen können und umgekehrt. Dazu wird während der Projektlaufzeit als weitere Komponente ein Gateway benötigt, das diese Zugriffe ermöglicht. Die Entwicklung eines Gateways kann unter Umständen sehr komplex und aufwändig sein.

Wegen der zusätzlichen Aufwände für die Entwicklung des Gateways und die zusätzlichen Test- und Abnahmephasen ist die inkrementelle Migration in der Regel teurer als der Big Bang Ansatz.

- **Zerlegbarkeit in Komponenten ist Voraussetzung**

Eine Ablösung in Stufen ist nur möglich, wenn das Altsystem in Komponenten zerlegbar ist. Eventuell sind vor der eigentlichen Ablösung noch Restrukturierungen im Altsystem erforderlich, die es erlauben, das Altsystem in einzelnen Schritten durch

*Zerlegbarkeit ist  
Voraussetzung*

neue Bestandteile zu ersetzen, oder das ganze System muss gekapselt werden.

Die Big Bang Strategie bringt folgende Vorteile mit sich:

- **Geringere Kosten**

Eine Big Bang Ablösung ist in der Regel billiger als eine stufenweise Umstellung, da Aufwände für die Entwicklung von Gateways und Kapseln entfallen. Auch fallen Aufwände für Test und Einführung nur einmalig an.

*Geringere  
Kosten*

- **Größere Freiheitsgrade**

Da bei der Big Bang Umstellung das neue System unabhängig vom Altsystem entwickelt wird, sind die Freiheitsgrade bei der Entwicklung in der Regel größer, da Strukturen des Altsystems nicht berücksichtigt werden müssen.

*Freiheits-  
grade*

Diese Migrationsstrategie ist jedoch durch eine Reihe von Nachteilen gekennzeichnet:

- **Sehr hohes Risiko**

Da die Umstellung aller Komponenten zu einem Zeitpunkt geschieht, ist das Risiko extrem hoch.

*Hohes Risiko*

*Fehlersuche  
schwierig*

Da Altsysteme häufig sehr groß sind, wird auch das Zielsystem groß sein. Mit der Projektgröße steigt jedoch die Wahrscheinlichkeit, dass das Projekt scheitert (Boehm, 1991).

Sollte nur an einer Stelle ein Fehler auftreten, bricht unter Umständen das komplette System zusammen. Auch gestaltet sich die Fehlersuche unter Umständen sehr schwierig, da nicht sofort ersichtlich ist, wo der Fehler aufgetreten ist.

*Neue Anfor-  
derungen*

- **Dauer des Projektes**

Eine Big Bang Umstellung ist aufgrund der Größe und Komplexität des Projektes mit einem erheblichen Zeitaufwand verbunden und dauert unter Umständen mehrere Jahre. Während der Projektlaufzeit entwickelt sich die Technologie jedoch weiter und auch die geschäftlichen Anforderungen können unter Umständen eine ganz neue Richtung bekommen haben. Wenn die Migration dann abgeschlossen ist, finden sich die Unternehmen unter Umständen in der Situation, dass das neue System die Anforderungen erfüllt und technisch bereits wieder veraltet ist (Bisbal et al, 1999b).

- **Risiken der Datenmigration**

Die im Altsystem gespeicherten Daten müssen in einem Schritt in das neue System migriert werden. Wenn die Daten im Altsystem Fehler oder Inkonsistenzen aufweisen, kann die Migration unter Umständen fehlschlagen und das neue System nicht in Betrieb gehen.

Dieses Problem tritt zwar im Prinzip bei jeder Migration auf, es ist jedoch bei einer Big Bang Migration besonders gravierend, da alle Daten betroffen sind und nur sehr wenig zeitlicher Spielraum für deren Korrektur verfügbar ist.

- **Hoher gleichzeitiger Aufwand**

Altsysteme haben häufig viele Nutzer, die bei einer Stichtagsumstellung alle gleichzeitig „umgestellt“ werden. D.h. der Schulungsaufwand ist im Vorfeld extrem hoch, da alle gleichzeitig fertig geschult sein müssen und direkt nach der Umstellung besteht unter Umständen hoher Supportbedarf.

*Daten-  
Migration*

*Support-  
bedarf*

## 6.4 Anforderungen an eine Migration

Damit eine Migration erfolgreich sein kann, sind laut Brodie und Stonebraker (1995) mindestens folgende Anforderungen zu erfüllen:

*Zuverlässiger  
Betrieb*

- **Der ununterbrochene, sichere und zuverlässige Betrieb ist zu garantieren.**

Ausfälle zentraler Informationssysteme kann ein Unternehmen nicht über längere Zeit verkraften. Bereits kurzzeitige Ausfälle führen in der Regel zu finanziellen Verlusten.

*Nur  
notwendige  
Änderungen*

- **Nur so viele Änderungen durchführen, wie notwendig, um aktuelle und zukünftige Anforderungen abzudecken.**

Hierdurch wird sichergestellt, dass nicht bereits kurz nach dem Abschluss der Migration das neue System angepasst werden muss, was wiederum eine neue Migration erforderlich machen könnte.

*Risiko  
minimieren*

- **Nur so wenige Änderungen durchführen wie möglich, um den Umfang und das Risiko der Migration zu minimieren.**

Je komplexer eine Migration ist, desto größer ist das damit verbundene Risiko. Die Komplexität der

Migration steigt mit der Anzahl der durchzuführenden Änderungen.

- **Alten Code so wenig wie möglich abändern, um Risiken zu minimieren.**

Solange der Code funktioniert und keine neuen Funktionalitäten notwendig sind, sollte er übernommen werden wie er ist, bzw. sollten nur minimale Änderungen durchgeführt werden. Änderungen können Fehler in der Implementierung nach sich ziehen.

Dieses Prinzip wird jedoch häufig nicht angewendet, da das ganze System ohne Übernahme des Codes komplett neu entwickelt wird.

- **Alten Code soweit abändern, dass er die Migration unterstützt.**

Wenn durch Änderungen am Code, die mit vertretbarem Aufwand durchgeführt werden können, die Migration vereinfacht wird, sollten diese durchgeführt werden.

*Alten Code wenig verändern*

*Alter Code soll Migration unterstützen*



*Flexibilität einbauen*

- **Möglichst große Flexibilität einbauen, um zukünftige Änderungen zu erleichtern.**

Da davon auszugehen ist, dass auch das neue System irgendwann verändert bzw. migriert werden muss, sollte eine flexible Systemarchitektur gewählt werden. Beispielsweise können die Kapselung der Funktionen und die Bereitstellung eines APIs (Application Programming Interface) zukünftige Entwicklungen und Anpassungen erleichtern.

*Negative Auswirkungen minimieren*

- **Negative Auswirkungen der Änderungen möglichst minimieren.**

Bei Änderungen am System sollte geprüft werden, ob diese Änderungen noch mit dem System verträglich sind, um hierdurch möglichst frühzeitig Fehlentwicklungen zu vermeiden.

*Moderne Technologien nutzen*

- **Nutzen moderner Technologien und Methoden maximieren**

Durch den Einsatz moderner Technologien und Methoden werden zum einen zukünftige Anpassungen erleichtert, zum anderen lassen sich Systemwerte, die die Entscheidung für eine Migration unterstützen, positiv beeinflussen.

## 7 Vorgehensmodell für Migrationsprojekte

Nachdem in den vorangegangenen Kapiteln die Voraussetzungen, Rahmenbedingungen und Strategien für Migrationsprojekte betrachtet wurden, soll nun der Ablauf von Migrationsprojekten eingehender betrachtet werden. Nach einem kurzen Überblick über verschiedene Ansätze wird ein ausgewähltes Vorgehensmodell detaillierter beschrieben.

### 7.1 Überblick

Eine standardisierte Vorgehensweise zur Durchführung von Migrationsprojekten gibt es bislang nicht. So finden sich in der Literatur auf der einen Seite zahlreiche Modelle zur Softwareentwicklung, auf der anderen Seite Prozessmodelle des Reengineering oder der Migration. Einen integrierten Ansatz im Sinne eines generischen Referenz-Prozesses für Migrationsprojekte sucht man, bis auf den Ansatz von Ackermann et al. (2005), jedoch vergeblich.

*Kein  
Standard*

*Unterschiedliche Ansatzpunkte*

Betrachtet man die in der Literatur beschriebenen Ansätze zur Migration, so wird deutlich, dass diese jeweils unterschiedliche Schwerpunkte legen. Während beispielsweise der Chicken Little Ansatz (Brodie / Stonebraker, 1995) eine Vorgehensweise zur Migration in elf iterativen Schritten beschreibt, beschäftigt sich Borchers (1997) in dem Konzept der "Reengineering Factory" mit dem Versuch, die in allen Software-Reengineering-Projekten gleichartig vorkommenden Prozesse und Organisationsmuster zu standardisieren und im Sinne einer Fließfertigung anzuwenden.

Einen ganz anderen Fokus haben hingegen Seacord et al. (2003), die in ihrem Ansatz der Risk-Managed Modernization Methoden des Software-Engineerings mit Ansätzen des Risikomanagements verbinden.

Die nachfolgende Tabelle gibt einen Überblick über die in der Literatur beschriebenen Ansätze und deren Autoren, sortiert nach dem jeweiligen Fokus.

Tab. 4: Ansätze zur Durchführung von Migrationsprojekten (Quelle: Eigene Darstellung)

<b>Fokus: Allgemeine Evolutionsprozesse</b>	
Chicken Little Ansatz	Brodie /Stonebraker (1995)
Allgemeiner Migrationsprozess	Sneed et al. (2004)
Reengineering Factory	Borchers (1997)
Renaissance	Warren (1999)
<b>Fokus: Objektorientierung</b>	
Objektorientierte Software-Migration	Sneed (1999)
FAMOOS	Bär et al. (1999)
<b>Fokus: Management</b>	
Risk-Managed-Modernization	Seacord et al. (2003)
<b>Fokus: Daten</b>	
Butterfly	Wu et al. (1997)
Mikado	Aebi (1997)

Aus diesen verschiedenen Ansätzen lässt sich eine Reihe von Tätigkeiten ableiten, die (unabhängig vom gewählten Ansatz) in jedem Migrationsprojekt durchgeführt werden müssen. Diese sind:

- Anforderungsanalyse
- Legacy-Analyse und Aufbereitung
- Ziel- und Brückendesign
- Strategieauswahl

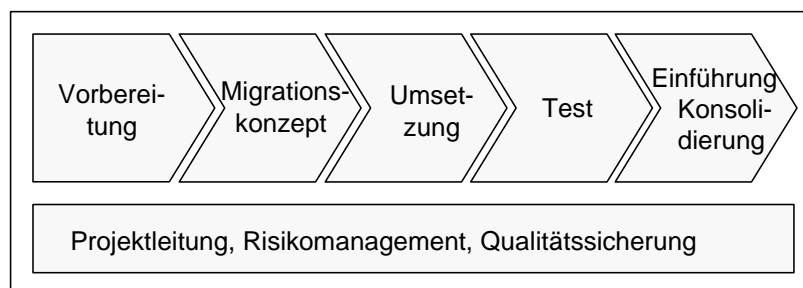
- Implementierung
- Test (Qualitätssicherung)
- Produktivsetzung (Übergabe / Cut Over).

### Eigene Vorgehensmodelle

Die auf Migrationsprojekte spezialisierten IT-Dienstleister haben eigene Vorgehensmodelle für ihre Projekte definiert, die die genannten Aktivitäten eines Migrationsprozesses in mehreren Phasen zusammenfassen.

So hat die Business Solution Group ein strukturiertes, fünfphasiges Vorgehensmodell entwickelt, das in Abbildung 16 dargestellt ist. Die Aufgaben und Inhalte der einzelnen Phasen werden in den folgenden Unterkapiteln beschrieben.

Abb. 16: Phasen in Migrationsprojekten. (Quelle: Eigene Darstellung)



## 7.2 Phase 1: Vorbereitung

Ein Migrationsprojekt ist immer mit erheblichen Kosten und Risiken für das Unternehmen verbunden (vgl. dazu Kapitel 3). Bevor die Entscheidung für eine Migration getroffen wird, sollte eine genaue Analyse durchgeführt werden, um Risiken und Nutzen zu quantifizieren und die Migration zu rechtfertigen (Sneed, 1995).

*Vorüberlegungen*

Im ersten Schritt ist zunächst zu prüfen, inwieweit sich die alte und die zukünftige neue technische Umgebung voneinander unterscheiden. So bestehen beispielsweise bei einer Migration von einem Mainframe zu einer verteilten Client-/Server-Umgebung erhebliche Unterschiede bei den Datenzugriffstechniken, bei den Benutzeroberflächen und in den Programmstrukturen. Um derartige Unterschiede identifizieren und dokumentieren zu können, ist eine detaillierte Analyse und Bewertung des Legacy-Systems durchzuführen.

*Soll-Ist-Vergleich*

Auf der anderen Seite sind die architektur- bzw. migrationsrelevanten fachlichen und technischen (Soll-)Anforderungen zu erfassen. Durch einen Abgleich von Ist und Soll kann dann festgestellt werden, ob es sich überhaupt lohnt, das System zu migrieren und mit welchen Kosten zu rechnen ist.

*Anforderungen*

*Schätzung  
des Projektes*

Geschätzt werden Migrationsprojekte anhand der Anzahl der zu migrierenden Objekte und der bisherigen Migrationsproduktivität aus früheren oder anderen vergleichbaren Migrationsprojekten. Zu den Objekten zählen die Benutzeroberflächen oder Masken, die Dateien und Datenbanken, die Systemschnittstellen, die Programme und Jobsteuerungsprozeduren. Alle diese Softwareobjekte müssen analysiert und gemessen werden. Dabei wird nicht nur ihre Größe in Zeilen und Anweisungen, sondern auch ihre innere Qualität und ihre Komplexität gemessen.<sup>10</sup>

*Kostenermittlung*

Basierend auf diesen Bewertungen können die ungefähren Kosten des Projektes ermittelt werden (zu Kosten der Migration vgl. ausführlich Kapitel 3.4). Die Kosten sollten dann mit den Risiken kalibriert werden. Der Projektrisikofaktor ist das Produkt aller einzelnen Risiken. Diese wiederum sind das Produkt der Risikoauswirkung mal der Risikowahrscheinlichkeit mit dem Resultat der Gegensteuerungsmaßnahmen. Die potentiellen Kosten des Projektes sind somit die geschätzten Kosten multipliziert mit dem Risikofaktor.

---

<sup>10</sup> Eine ausführliche Beschreibung findet sich z.B. bei Sneed, 1999.

Darüber hinaus sollte auch der Nutzen in die Betrachtung einbezogen werden. Dieser kann unterteilt werden in monetäre Einsparungen und nicht-monetäre Kriterien. Monetäre Einsparungen ergeben sich z.B. durch geringere Betriebs- und Wartungskosten, wegfallende Lizenzgebühren, günstigere Supportverträge für das neue System im Vergleich zum alten System.

*Monetäre  
Einsparungen*

Für die nicht-monetären Kriterien empfiehlt sich eine Nutzwert-Analyse.<sup>11</sup> Nicht-monetäre Kriterien können z.B. in den Bereichen Dringlichkeit, Qualität und strategische Bedeutung betrachtet werden.

*Nicht-monetäre  
Kriterien*

Im Bereich Dringlichkeit wären z.B. folgende Aspekte anhand einer Notenskala zu bewerten:

*Dringlichkeit*

- Unterstützungskontinuität Altsystem
- Stabilität Altsystem
- Fehler und Ausfälle (Downtime)
- Flexibilität
- Ausbau- und Erweiterungsgrenzen
- Bedienbarkeit und Ergonomie (Benutzerfreundlichkeit).

---

<sup>11</sup> Vgl. dazu ausführlich BMI, 2006.



*Qualität* Im Bereich Qualität könnten z.B. folgende Aspekte bewertet werden:

- Einhaltung von Gesetzen und Vorschriften
- Erfüllung von Datenschutz und Datensicherheit
- Sicherstellung ordnungsgemäßer Arbeitsabläufe
- Erfüllung von Auflagen und Empfehlungen.

*Strategie* Zu den qualitativ-strategischen Kriterien zählen z. B.:

- Bedeutung innerhalb der IT-Strategie / Architektur
- Auswirkungen auf Kommunikationspartner
- Qualitätsverbesserung bei der Aufgabenerfüllung
- Imageverbesserung
- Verbesserung der Arbeitssituation.

*Projektplanung* Wenn nach Abschluss der Wirtschaftlichkeitsbetrachtung die Entscheidung für ein Migrationsprojekt gefallen ist, ist dieses zu planen und aufzusetzen. Dazu sind eine eigenständige Projektorganisation aufzubauen und die finanziellen und personellen Ressourcen zur Verfügung zu stellen.

### **7.3 Phase 2: Migrationskonzept**

*Entwicklung Migrationskonzept* Im Mittelpunkt der zweiten Phase steht die Entwicklung des Migrationskonzeptes. Hierbei geht es um die Definiti-

on des Ziel- und Brückensystems sowie um die Festlegung der Migrationsstrategie.

Ausgehend von der Legacy-Analyse und den dabei gewonnenen Erkenntnissen zum Legacy Design sowie der Anforderungsanalyse wird die Zielarchitektur entwickelt. Dabei ist zu beachten, dass das neue System den (benötigten) Funktionsumfang des alten Systems beibehalten muss. Festzulegen sind insbesondere die Hardwareumgebung, Systemsoftware, Entwicklungsumgebung und Software-Architektur sowie die hiervon betroffenen Aspekte der Benutzerschnittstellen, Daten und Programme.

*Ziel-  
architektur*

Das Legacy-Design und die angestrebte Zielarchitektur beeinflussen die Festlegung der Migrationsstrategie. Bei großen, nicht strukturierten Altsystemen bleibt häufig nur eine Big Bang-Ablösung, während bei besser strukturierten Systemen die stufenweise Ablösung denkbar ist (vgl. dazu Kap. 6).

*Migrations-  
strategie*

In Abhängigkeit von gewählter Migrationsstrategie und angestrebter Zielarchitektur kann dann das Brückensystem, d.h. die Übergangsarchitektur, die für den Wechsel von einem System zum anderen benötigt wird, gestaltet werden.

*Brücken-  
system*

*Verstehen  
des Alt-  
systems*

Darüber hinaus sollte auch die Daten-Migration rechtzeitig vorbereitet werden. Zum einen ist in diesem Zusammenhang die Logik der Geschäftsprozesse zu untersuchen und zu prüfen, welche Konsequenzen sich daraus für die Daten-Migration ergeben.

*Daten-  
strukturen*

Zum anderen müssen die Datenstrukturen des Altsystems genau verstanden werden. Wird das Altsystem in Stufen abgelöst, so können altes und neues System auf die Daten gemeinsam zugreifen. Vor diesem Hintergrund sind beispielsweise häufige Zugriffsmuster sowie das Sperrverhalten des Altsystems zu untersuchen, um Zugriffskonflikte beim parallelen Zugriff beider Systeme zu vermeiden.

*Daten-  
qualität*

Auch die Datenqualität muss frühzeitig berücksichtigt werden. So können die über die Jahre gewachsenen Datenbestände Daten enthalten, die unvollständig, nicht mehr aktuell oder sogar falsch sind. Vor der Migration ist zu festzulegen, wie mit diesen Problemen umgegangen werden soll. Geht man beispielsweise davon aus, dass das Altsystem 5 Mio. Datensätze umfasst und dass davon nur 1 % der Datensätze Fehler enthält, so bedeutet dieses ggf. die manuelle Prüfung von 50.000 Datensätzen.

## 7.4 Phase 3: Umsetzung

In der nächsten Phase erfolgt die eigentliche Umsetzung. Dabei empfiehlt sich folgende Reihenfolge (Sneed, 1995):

1. Daten in die neue Datenbank übertragen
2. Programme konvertieren und kapseln
3. Oberflächen anpassen.

Für die Daten-Migration wird zumeist ein bestimmter Zeitpunkt festgelegt, an dem alle Datenbanken auf einmal übertragen werden. Ausführlichere Erläuterungen zum Ablauf der Daten-Migration finden sich in Kapitel 8.

Die Programme oder Komponenten werden häufig paketweise umgesetzt. Dabei spielt die richtige Paketierung eine wichtige Rolle: Es sollten immer Pakete definiert werden, die möglichst kohäsiv, d.h. in sich geschlossen sind und möglichst wenige Verknüpfungen zu anderen Paketen haben. Es sollte auch möglich sein, die einzelnen Pakete unabhängig voneinander zu testen.

Der Aufbau der neuen Oberflächen als GUI- oder Web-Clients setzt gute Kenntnisse der neuen Zielumgebung voraus. Um Oberflächen unabhängig von der Geschäfts-

*Reihenfolge*

*Daten-Migration*

*Paketierung*

*Oberflächen*

logik testen zu können, müssen sie von den dahinter liegenden Komponenten getrennt sein.

### *Schulungen*

Weiterhin muss beachtet werden, dass sowohl die Mitarbeiter, die mit der Migration betraut sind, als auch die Mitarbeiter, die später mit dem System arbeiten sollen, durch Schulungen entsprechend qualifiziert werden.

## **7.5 Phase 4: Test**

### *Regressi- onstest*

Der größte Kostentreiber in einem Migrationsprojekt ist der so genannte Regressionstest (auch Back to Back Test genannt) (Sneed et al., 2005). Im Rahmen des Regressionstests wird gezeigt, dass sich die neue Software funktional äquivalent zur alten verhält. Dieses ist auch dann zu tun, wenn der Transformationsprozess voll automatisiert war, denn niemand kann wissen, ob sich die neue Lösung genauso verhält wie die alte.

### *Funktionale Äquivalenz*

Um die funktionale Äquivalenz zu zeigen, müssen sowohl altes als auch neues System mit den gleichen Eingaben und Daten getestet werden und die Ergebnisse hinterher im Detail abgeglichen werden. Nur die 100%-Übereinstimmung der alten und neuen Ergebnisse kann den Erfolg der Migration bestätigen. In der Regel ist es

hierzu erforderlich, spezielle Abgleichwerkzeuge einzusetzen, die Datenbanken, Dateien und Maskeninhalte über Umgebungsgrenzen hinaus logisch vergleichen können.

Darüber hinaus empfiehlt es sich, an Testwochenenden den Ablauf der eigentlichen Migration rechtzeitig zu proben. Nur so kann z. B. überprüft werden, ob die eingeplante Zeit ausreicht, um die Migration durchzuführen, oder ob eine Aufteilung auf mehrere Termine notwendig ist. Die migrierten Stände können dann zu weiteren Regressionstests verwendet werden.

*Testwochenenden*

## **7.6 Phase 5: Einführung und Konsolidierung**

Wenn alle Tests erfolgreich abgeschlossen sind, kann die eigentliche Migration entsprechend der gewählten Migrationsstrategie stattfinden. Hierbei geht es darum, den aktiven Betrieb des Altsystems möglichst ohne Unterbrechungen und Datenverlust in den Betrieb des neuen Zielsystems zu überführen.

*Cut Over*

Häufig wird zur Minderung der Risiken zunächst ein Parallelbetrieb durchgeführt, d.h. altes und neues System arbeiten für einen bestimmten Zeitraum ganz oder teil-

*Parallelbetrieb*

weise parallel. Während dieser Zeit kann das Verhalten des neuen Systems unter realen Bedingungen beobachtet werden. Treten keine Probleme und Störungen auf, so kann das Altsystem mit geringerem Risiko abgeschaltet werden.

## 8 Vorgehen bei der Daten-Migration

Im vorhergehenden Kapitel wurde eine allgemeine Vorgehensweise für Migrationsprojekte beschrieben, ohne im Detail auf die Daten-Migration einzugehen. Aufgrund der großen Bedeutung, die die Daten-Migration in der Praxis hat, wird ihr in diesem Kapitel noch einmal besondere Beachtung geschenkt.

### 8.1 Bedeutung der Datenmigration

Sehr häufig wird die Daten-Migration als „Aschenputtel“ der IT betrachtet und an das Ende des Projektplans eines groß angelegten Migrationsprojektes „angehängt“. Dabei wird verkannt, dass eine Daten-Migration unter Umständen einen Großteil der finanziellen und personellen Projektressourcen binden kann. So haben viele Unternehmen – häufig erst mitten im Projekt – die Erfahrung gemacht, dass die Daten-Migration bis zu 40% des gesamten Projektbudgets „verschlingen“ kann. (Informatica, 2003, S. 1).

*Großer  
Kostenblock*



*Daten-  
qualität*

Neben der finanziellen Belastung, die eine Daten-Migration für ein Unternehmen mit sich bringt, sollte ein anderer Aspekt unbedingt berücksichtigt werden: Der Aussagegehalt der migrierten Daten, d. h. die Datenqualität.

*Fachliche  
Dimension*

Bei einer Daten-Migration geht es nicht einfach darum, Daten von einem Platz an einen anderen „zu bewegen“. Daten-Migration ist nicht nur ein technisches Thema, sondern hat auch eine sehr wichtige fachliche Dimension, denn die Qualität der Daten spielt eine sehr wichtige Rolle für die Abwicklung der Geschäftsprozesse. Demzufolge sollte die Ausgangsfrage in einem Daten-Migrationsprojekt nicht lauten „Wie können wir diese Daten migrieren?“ sondern vielmehr „Welche Daten sollten wir migrieren?“

*Basis für  
Geschäfts-  
prozesse*

Werden beispielsweise offene Posten aus einem Altsystem in ein neues System übernommen, so reicht es nicht aus, dass sich die Positionen in beiden Systemen sowohl einzeln als auch kumuliert entsprechen. Vielmehr müssen auch zusätzliche Daten / Informationen übernommen werden, die für eine reibungslose Abwicklung der Folgeprozesse wichtige Steuerungsqualitäten besitzen (Willinger / Gradl, 2003). In Bezug auf die offenen Posten muss

beispielsweise gewährleistet sein, dass alle die Informationen im neuen System vorliegen, die benötigt werden, um eine korrekte Berechnung der Fälligkeiten zu ermöglichen, denn dieses hat unmittelbare Auswirkungen auf die Liquiditätsplanung des Unternehmens. Fehlende oder falsche Informationen könnten die Entscheider zu nicht beabsichtigten Entscheidungen verleiten.

Darüber hinaus ist bei der Datenmigration darauf zu achten, dass alle gesetzlichen Vorschriften sowie die internen Governance Compliance Kriterien eingehalten werden.

Neben der großen Bedeutung, die die Datenmigration für ein Unternehmen besitzt, sieht Morris (2006, 123ff) vier weitere Gründe, warum ein Daten-Migrationsprojekt immer als eigenständiges (Teil-)Projekt aufgesetzt werden sollte:

- Die „Produkte“ eines Daten-Migrationsprojektes unterscheiden sich grundlegend von den Produkten eines Softwareentwicklungsprojektes.
- Das Daten-Migrationsteam nimmt eine Vermittlerposition zwischen den Fachbereichen des Unter-

*Einhaltung  
Regelungen*

*Eigenes  
Projekt*

nehmens und den Programmierern des übrigen Projektes ein.

- Für eine Daten-Migration werden Spezialisten benötigt.
- Die Projektstruktur eines Daten-Migrationsprojektes unterscheidet sich grundlegend von der in Softwareentwicklungsprojekten.

## 8.2 Ablauf einer Daten-Migration

### *Vorbereitungen*

Bevor mit der eigentlichen Daten-Migration begonnen wird, sollten einige Vorüberlegungen angestellt werden, die für die weitere Projektplanung wegweisend sind. Dazu gehören:

- Die Definition des zu migrierenden Datenbestands
- Die Identifikation von „Karteileichen“
- Maßnahmen zur Reduktion des Datenvolumens.

### *Zu migrierende Daten*

Die zu migrierenden Daten lassen sich unterscheiden in Stammdaten und Bewegungsdaten. Stammdaten bleiben über einen längeren Zeitraum unverändert und werden immer wieder benötigt. Bewegungsdaten sind vorgangsbezogene Daten, die relativ kurzlebig sind.

Vor der Migration sollten so genannte „Karteteileichen“, die keinerlei operativen Bezug mehr haben, d.h. nicht mehr verwendet werden, im Altsystem gekennzeichnet werden, damit sie von der Datenübernahme ausgeschlossen bleiben. Darüber hinaus sollten redundante Altdaten, d.h. identische Stammdaten, die im Altsystem unter verschiedenen Identifikationsnummern abgespeichert sind, identifiziert und entsprechend gekennzeichnet werden.

*Karteteileichen*

Für die Bewegungsdaten, d.h. die kurzlebigen, vorgangsbezogenen Daten (auch Buchungsbelege genannt), ist zu klären, wie detailliert die Historie des Altsystems im neuen System abgebildet werden soll. Das heißt, es ist zu klären, ob die Einzelposten oder nur die kumulierten Werte (die Salden) übernommen werden sollen. Darüber hinaus ist festzulegen, für wie viele zurückliegende Geschäftsjahre die Daten übernommen werden sollen. Diese Entscheidungen beeinflussen das zu migrierende Datenvolumen ganz entscheidend.

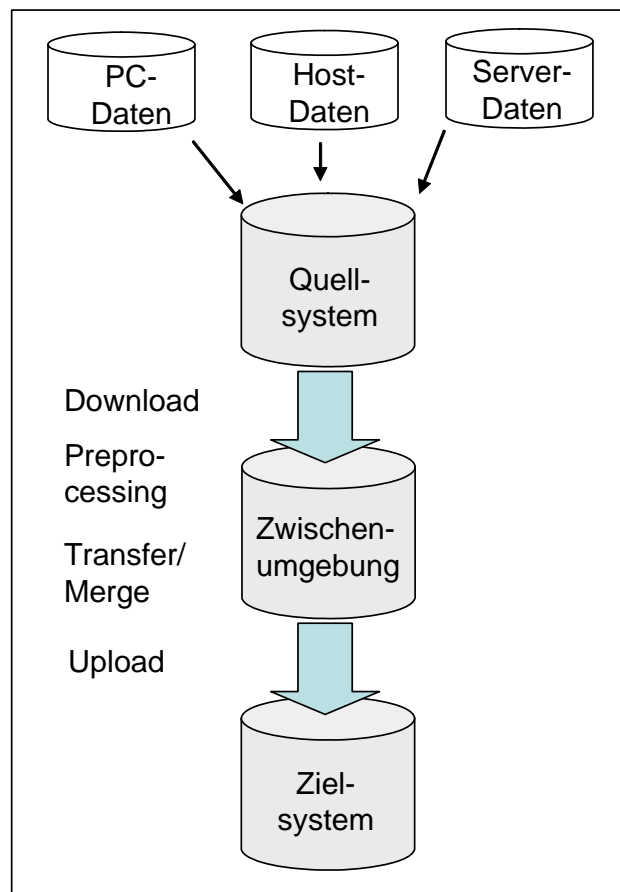
*Einzelposten  
oder Salden*

Neben diesen Überlegungen sollte frühzeitig geprüft werden, ob das Altsystem generell in der Lage ist, alle Stamm- und Bewegungsdaten in eine Datei oder mehrere Dateien auszulesen. Darüber hinaus ist zu prüfen, ob das System für einen solchen Datenextrakt Standard-

*Datenextrakt*

	<p>funktionalitäten bereitstellt oder ob hierfür ggf. eigene Programmierkapazitäten zur Verfügung gestellt werden müssen.</p>
<i>Ablauf der Daten-Migration</i>	<p>Wenn alle vorbereitenden Tätigkeiten abgeschlossen sind, kann mit der eigentlichen Daten-Migration begonnen werden, die in Abbildung 17 gezeigt und im Folgenden beschrieben wird.</p>
<i>Download</i>	<p>Im Rahmen des Downloads wird ein Roh-Export der Daten aus den Altsystemen durchgeführt. Zielsetzung ist es, die Daten in einer Zwischenumgebung in einem Transport- und Weiterverarbeitungsmedium, z.B. Flat Files, zu haben. Als Hilfsmittel in dieser Phase kommen z.B. Export-Utilities der Altsysteme, Datenbank-Prozeduren oder Datenbank-Export-Tools zum Einsatz.</p>
<i>Preprocessing</i>	<p>Beim Preprocessing werden alle Quelldaten in ein gemeinsames (Zwischen-)Format überführt. Darüber hinaus findet eine Vorverarbeitung (z.B. Konsolidierung, Formattierung) der Daten aus den Altsystemen statt. Ziel ist es, ein einheitliches Format zur Weiterverarbeitung der Quelldaten zu bekommen. Als Tools kommen z.B. Skriptsprachen, Texteditor-Makros und Programmiersprachen zum Einsatz.</p>

Abb. 17: Ablauf Daten-Migration. (Quelle: Eigene Darstellung)



Beim Transfer/ Merge werden die Daten zum Zielsystem transferiert und die Daten der verschiedenen Altsysteme zusammengeführt, damit alle Daten im Neusystem verfügbar sind. In diesem Zusammenhang spielt das Data Mapping eine wesentliche Rolle.

*Transfer / Merge*

<i>Data Mapping</i>	Beim Data Mapping (Feldabgleich) geht es darum, auf Feldebene zu untersuchen, wo die Daten genau herkommen und wohin sie im neuen System übertragen werden. In den seltensten Fällen werden die Feldbezeichnungen des Altsystems mit den korrespondierenden Begrifflichkeiten des neuen Systems übereinstimmen. Liegen unterschiedliche Begrifflichkeiten vor, so müssen die jeweiligen Bezeichnungen festgehalten und entsprechend zugeordnet werden.
<i>Transformation</i>	Darüber hinaus ist zu prüfen, was auf diesem Weg getan werden muss. Treffen unterschiedliche Konzeptionen aufeinander, so müssen Felder des Altsystems, die das neue System nicht kennt, zunächst transformiert und dem entsprechenden Feld des neuen Systems zugewiesen werden. So kann es z. B. sein, dass ein Wert aus dem Altsystem in mehrere Werte im neuen System aufgeteilt werden muss oder mehrere Werte des Altsystems zu einem Wert im neuen System zusammengefasst werden müssen.
<i>Upload</i>	Der Upload dient der Bereitstellung der Daten in der Datenbank des Zielsystems. Dazu werden die Daten über die Schnittstellenstrukturen importiert. Als mögliche Tools

kommen z.B. Upload Tools der Datenbanken oder Datenbank-Prozeduren zum Einsatz.

### 8.3 Problembereiche der Daten-Migration

Es lassen sich drei große Problembereiche der Daten-Migration identifizieren:

- Download von Altsystemen,
- Datenbereinigung,
- Upload.

Der Download kann sich unter Umständen schwierig gestalten, weil zu wenig Know-how über das Altsystem vorhanden ist und / oder die Anforderungen der neuen Systeme an die zu liefernden Daten nur unzureichend bekannt sind. Darüber hinaus kommt es teilweise zu Problemen weil die Umsetzungsregeln und die Datenintegrität nicht genügend beachtet werden.

*Download*

Im Bereich der Datenbereinigung spielt die Datenqualität eine wesentliche Rolle. Auf jeden Fall sollte die Bereinigung der Daten bereits im Altsystem stattfinden (vgl. dazu Kap. 8.2). Häufig sind jedoch die Anforderungen der

*Datenbereinigung*



neuen Systeme an die Datenqualität nicht oder nicht ausreichend bekannt.

*Zu hoher  
Qualitäts-  
anspruch*

Dieses kann zum einen dazu führen, dass die Ansprüche an die Bereinigung zu hoch gesteckt werden. Die „Altlasten“ sollen komplett bereinigt werden und eine 100% Datenqualität wird angestrebt. Dadurch uferf die Datenbereinigung aus und läuft sich unter Umständen tot.

*Qualitäts-  
mängel*

Zum anderen führt eine zu geringe Datenqualität zu Problemen beim Upload in das neue System und damit zu zeitintensiven Rückkoppelungsprozessen zwischen Download- und Upload-Programmierung.

## 9 Fallstudie

Erfahrungsbericht von R. Fischer, Leiter Migration der Business Solution Group AS, Zürich

### 9.1 Die Ausgangssituation

Die Finanzdienstleistungsbranche ist derzeit von Fusionen gekennzeichnet. Die im Folgenden dargestellte Fallstudie beschreibt ein Migrationsprojekt, in dem drei Privatbanken fusioniert und in einen Großkonzern integriert wurden.

Voraussetzung für die erfolgreiche Umsetzung der One Brand-Geschäftsstrategie des Mutterkonzerns war die Integration der drei übernommenen Banken und die Migration auf eine gemeinsame technische Plattform.

Zwei der Banken hatten ursprünglich die Gesamtbanklösung Olympic von ERI Bancaire mit diversen unterschiedlichen dezentralen Systemen (Trading, Reconciliation, Portfolio-Management etc.) im Einsatz, die dritte Bank das Bankenpaket Penta der Firma Sobaco mit verschiedenen Umsystemen. Der Mutterkonzern hatte eine

*Banken-  
fusion*

*Strategische  
Notwendig-  
keit*

*Technische  
Vorausset-  
zungen*

IBM Mainframe Eigenentwicklung mit einer heterogenen dezentralen Systemlandschaft, auf die migriert werden sollte.

Dabei stand das Projekt vor drei wesentlichen Herausforderungen:

*Kulturelle  
Integration*

- Die (organisatorische) Integration der Banken erforderte ein Zusammenführen unterschiedlicher Bankkulturen aus verschiedenen geografischen und sprachlichen Kulturkreisen.

*Vereinheitlichung  
Prozesse*

- Auf der bankfachlichen Seite musste eine Vereinheitlichung der unterschiedlichen Prozesse und Geschäftsarten erzielt sowie die Verkaufsstruktur auf die neue Marktausrichtung angepasst werden.

*Technische  
Komplexität*

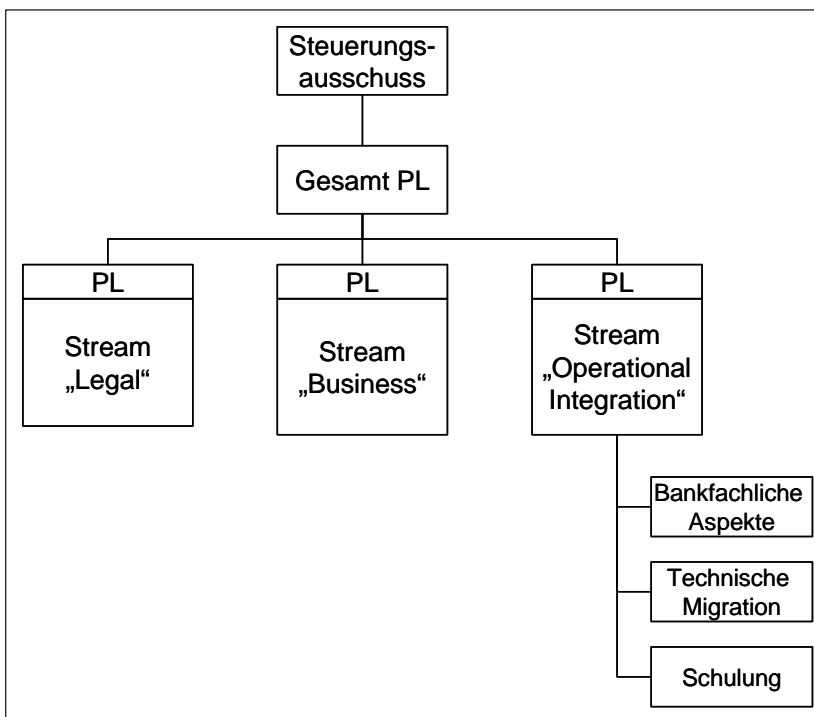
- Die technische Migration war von sehr hoher Komplexität gekennzeichnet, die zum einen durch die Größe der involvierten Banken, zum anderen durch einen sehr engen Zeitplan bedingt wurde. So war geplant, dass die erste Bank bereits 9 Monate nach Bekanntgabe der Fusion, die zweite Bank nach 12 Monaten und die dritte Bank nach 18 Monaten vollständig integriert sein sollten.

## 9.2 Die Projektorganisation

Für die Abwicklung eines Projektes dieser Größe war eine eigenständige, Projektorganisation und Projektzusammensetzung notwendig (vgl. Abb. 18):

*Projektorganisation*

Abb. 18: Projektorganisation. (Quelle: Eigene Darstellung)



Der Steuerungsausschuss bestand aus Geschäftsleitungsmitgliedern der verschiedenen Banken sowie des Mutterhauses. Er überwachte das Projekt hinsichtlich der Termine, Finanzen und Ressourcen und traf Entschei-

*Steuerungsausschuss*

dungen bei geschäftspolitischen oder migrationsrelevanten Differenzen. Ihm direkt unterstellt war der interne Gesamtprojektleiter, der aus dem Mutterhaus kam.

*Teilprojekte* Das Gesamtprojekt „Fusion“ war in drei (Teil)Projekte (Streams) unterteilt, die jeweils von einem Teilprojektleiter geführt wurden:

*Stream  
„Legal“*

- Der Stream „Legal“ war verantwortlich für alle rechtlichen und gesetzlichen Aspekte der Fusion sowie der daraus resultierenden Umsetzung der Namensänderung und dem gesamten Decommissioning-Prozess nach erfolgter Übernahme.

*Stream  
„Business“*

- Der Stream „Business“ war zuständig für den Aufbau bzw. die Anpassung der Verkaufsstruktur, die Harmonisierung der Geschäftsprozesse, die Information der Kunden sowie für die Benachrichtigung der externen Geschäftspartner.

*Stream  
„Operational  
Integration“*

- Der Stream „Operational Integration“ war verantwortlich für die termingerechte, fachlich korrekte und vollständige Migration der Datenbestände, sowie die Schulungen auf den neuen Systemen. Dieser Stream bestand aus IT- und Fachspezialisten.

ten aller involvierten Banken und wurde durch externes Migrations-Know-how unterstützt.

Die nachfolgenden Beschreibungen befassen sich mit den Tätigkeiten des Streams „Operational Integration“, da innerhalb dieses Teilprojektes die technische Migration durchgeführt wurde.

## 9.3 Vorgehensweise

### 9.3.1 Vorbereitung: Ist-Aufnahme und GAP-Analyse

Als Basis für die Entwicklung des Migrationskonzeptes erfolgte zuerst eine Ist-Aufnahme der verschiedenen Quellsysteme und des Ziel-Systems mit anschließender Zuordnung der einzelnen Applikationen bzw. Systeme zueinander. Dabei wurde auch festgelegt, ob diese migrationsrelevant waren oder nicht.

*Ist-Aufnahme*

Parallel dazu erfolgte eine GAP-Analyse. Dabei wurden die Funktionalitäten der Quell-Systeme und des Ziel-Systems miteinander verglichen. Bei fehlender Funktionalität wurde entschieden, ob die Problematik mittels eines Work-Arounds gelöst werden konnte, oder ob eine

*GAP-Analyse*

Anpassung des Ziel-Systems notwendig war. Dabei wurde die Devise „so wenig Anpassungen wie möglich“ konsequent umgesetzt und trug dazu bei, die Komplexität des Ziel-Systems und damit verbunden die Migrations-Risiken nicht drastisch zu erhöhen.

### 9.3.2 Migrationskonzept

#### *Aufgabenpakete*

In der nächsten Phase wurden folgende Aufgabenpakete bearbeitet, die im Folgenden beschrieben werden:

- Festlegen der Migrationsreihenfolge
- Festlegen der Migrationsstrategie
- Erarbeitung des Migrationskonzeptes
- Erarbeitung des Abstimmkonzeptes.

#### *Migrations-Reihenfolge*

Auf Basis der Resultate der IST-Aufnahme und der GAP-Analyse wurde die Reihenfolge der zu migrierenden Banken festgelegt und die Machbarkeit bezüglich der vom Management gewünschten Einführungstermine überprüft.

#### *Big Bang Strategie*

Aufgrund der Komplexität entschied man sich für eine „Big Bang“-Strategie für jede der drei zu migrierenden Banken. Diese wurde in zwei Stufen umgesetzt:

1. Stufe: Migration sämtlicher Stammdaten
2. Stufe: Migration aller Bewegungsdaten

Zwischen diesen beiden Einführungsstufen lag ein Zeitraum von vier Wochen, in dem die Stammdaten im Zielsystem überprüft und veredelt wurden. Dabei musste für diese Zeit ein Parallelbetrieb von Quell- und Ziel-System mit entsprechenden Abgleich-Prozeduren geplant werden.

*Parallelbetrieb*

Für die Erarbeitung des Migrationskonzeptes wurde pro Themenbereich eine Arbeitsgruppe gebildet, die jeweils aus Vertretern der abgebenden Bank und des Mutterhauses bestand.

*Migrationskonzept*

Pro Geschäftsprozess bzw. Applikation wurden im Migrationskonzept folgende Punkte beschrieben:

- Quell-System der Daten
- Ziel-System der Daten
- Mengen und Häufigkeiten
- Migrationsprozedur (automatisch / manuell / keine Migration) auf Seiten der abgebenden und der empfangenden Bank
- Vorbedingungen, die erfüllt sein müssen
- Vorbereitung für die Migration



*Abstimm-  
konzept*

- Prozedur am Migrations-Wochenende
- Datenbereinigungen vor (Prior) und nach (Post) erfolgter Migration.

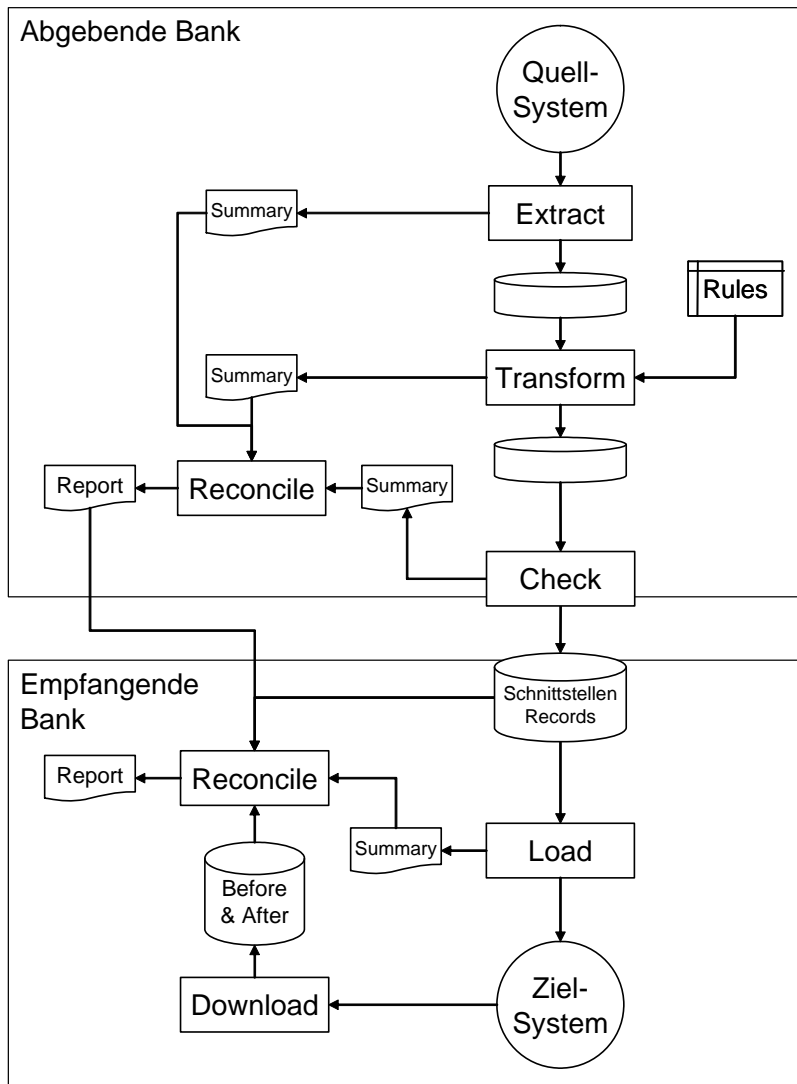
Um die Nachvollziehbarkeit der Migration aus Sicht der Revision (interne und externe Audits) zu gewährleisten, wurde ein Abstimmkonzept entwickelt. Dieses beinhaltet folgende Komponenten:

- Prüfpfad pro Schnittstellen-Record
- Quantitative Abstimmung mittels Anzahl und Hash-Totalen, sowohl für Stamm- als auch für Bewegungsdaten
- Abstimmung der Cash-Seite auf Basis der Bilanzen
- Abstimmung der Wertschriften-Seite auf Basis von Positionslisten und Depotbewertungen.

### 9.3.3 Umsetzung

Das Migrations-Konzept diente als Grundlage für die Festlegung der Migrations-Prozeduren und die Definition der entsprechenden Schnittstellen-Records.

Abb. 19: Vorgehen zur Umsetzung. (Quelle: Eigene Darstellung)



Im Rahmen der Umsetzung wurde wie folgt vorgegangen:

- Download* Für den Download von den Quellsystemen wurden zum einen Standard-Utilities verwendet, zum anderen wurden neue Entlade-Programme entwickelt.
- Transform* Für die Transformation wurden Transformations-Programme auf Basis der unter Punkt „Data-Mapping“ (siehe nachfolgend) definierten Schnittstellen-Records und Transformations-Regeln entwickelt.
- Check* Für den Check wurde eine eigene MS-Access Umgebung entwickelt. Die Plausibilisierung jedes einzelnen Schnittstellen-Records wurde aufgrund der Plausi-Regeln der zugehörigen Online-Transaktion / Batchprogramm vorgenommen. Darüber hinaus wurden Abhängigkeiten zwischen verschiedenen Schnittstellen-Records geprüft.
- Upload* Für den Upload wurden sowohl Scripts (TCET Trans Century Enterprise Tester) mit Einbindung bestehender Online-Transaktionen für die Stammdaten als auch bereits bestehende Batch-Abläufe für die Bewegungsdaten sowie Standard-Schnittstellen für dezentrale Systeme verwendet. Sofern notwendig wurden neue Lade-Programme entwickelt, falls diese nicht durch bestehenden Transaktionen, Abläufe oder Schnittstellen abgedeckt werden konnten.

Das Abstimm-Konzept war die Basis für die Festlegung der Abstimm-Prozeduren und die Definition der entsprechenden Abstimm-Records.

*Abstimm-Konzept*

Der Reconcile umfasste folgende Schritte:

*Reconcile*

- Entwicklung einer eigenen MS-Access Umgebung,
- Mengenmäßige Abstimmung bezüglich Anzahl und Hash-Totalen,
- Abstimmung der Bilanzen,
- Abstimmung der Wertschriften.

Die Check-, Upload- und Reconcile-Prozeduren konnten für alle drei involvierten Banken mit geringfügigen Anpassungen wieder verwendet werden. Die Download- und Transform-Prozeduren mussten für jede zu migrierende Bank entwickelt werden.

*Wiederverwendbarkeit*

Beim Data-Mapping handelte es sich um das Kernstück der Migration. Dieses hat folgende Aktivitäten zum Inhalt:

*Data Mapping*

- Definition der Schnittstellen-Records
- Bestimmung der Quell-Felder
- Festlegung der Beziehung(en) Quell-Felder zu Schnittstellen-Record (1:1, n:1, 1:n, n:n)
- Definition der Transformations-Regeln.

*Schulungen*

Für die Schulung der Mitarbeiter/innen auf dem neuen System (Ziel-System) wurde ein Schulungskonzept erstellt. Pro integrierter Bank wurden jeweils 250 - 300 Mitarbeiter/innen geschult. Das Schulungskonzept umfasste folgende Bausteine:

*Grundschulung*

- Ein Trainingsmodul für die Grundschulung jedes Mitarbeiters/Mitarbeiterin. Darin wurde ein Überblick über das System sowie dessen Handhabung und Grundfunktionen aufgezeigt.

*Rollen-spezifische Module*

- Pro Arbeitsgebiet wurde ein Rollen-Profil mit entsprechendem Trainingsmodul und Handbuch entwickelt. Jeder Mitarbeiter/Mitarbeiterin wurde einem bis mehreren Rollen-Profilen zugeordnet und besuchte die entsprechenden Trainingsmodule.

*Support*

- Es wurde eine Support-Organisation vor Ort aufgebaut. Nach dem erfolgreichen Cut-Over Wochenende unterstützten sowohl Key-User als auch Projektmitarbeiter des Schulungsteams als so genannte „Floor-Runners“ die neuen Mitarbeiter/innen bei auftretenden Fragen oder Problemen an deren Arbeitsplatz. Dieser Support wurde wäh-

rend der ersten drei Wochen nach dem Cut-Over angeboten.

#### 9.3.4 Test

Zur Absicherung der Projektergebnisse wurden verschiedene Testverfahren eingesetzt.

- Einzel- und Unit-Test

Die unzähligen Einzel- und Unit-Tests dienten zum Testen der einzelnen Migrationsprozeduren und –schritte. Durch sie wurde das Data-Mapping verifiziert. Die Tests wurden über fünf Monate iterativ durchgeführt, bis die jeweilige Prozedur vom Benutzer abgenommen wurde. Darüber hinaus dienten sie der Überprüfung der Systemleistung bezüglich Laufzeiten und der Space-Allozierung für die Datenbanken.

*Einzel-/Unit-Test*

- Kettentest

Im Kettentest wurden die einzelnen Prozeduren aus dem Unit-Test in einer zeitlich logischen Reihenfolge durchgeführt und die nachgelagerten dezentralen Systeme wurden über die Schnittstellen beliefert. Diese Tests verteilen sich auf mehrere Tage und wurden durch Benutzerkontrollen unterbrochen. Die End-Resultate dieser Tests wurden von den Fachabteilungen abgenommen. Pro mig-

*Kettentest*

rierte Bank wurden ca. sechs Mal je eine Woche Ketten-Tests durchgeführt.

### *Test- Wochen- ende*

- Test-Wochenende

An den Test-Wochenende, auch „Dress-Rehearsals“ genannt, wurde das Cut-Over Wochenende 1:1 durchgeführt. Die Testwochenenden dienten der Überprüfung der Planung sowie des korrekten Zusammenspiels aller involvierten Komponenten und Fachstellen. Neben allen automatischen Prozeduren wurden auch sämtliche manuellen Eingaben und Prozesse durch die Fachabteilungen durchgeführt. Pro migrierter Bank wurden je zwei Test-Wochenenden für die Stammdaten und zwei Wochenenden für die Bewegungsdaten durchgeführt.

### *Optimierung*

Parallel dazu erfolgte die Überprüfung und die Abnahme (Sign-Off) der Resultate der einzelnen Migrationsschritte. Auftretende Fehler wurden protokolliert und, wenn möglich, vor Ort korrigiert. Optimierungsmöglichkeiten wurden in der Planung berücksichtigt und anlässlich des nächsten „Dress-Rehearsals“ verifiziert. Nach Abschluss des jeweiligen Test-Wochenendes erfolgte eine Gesamtbeurteilung der Resultate und ein GO- / NO GO-Entscheid.

- **Regressions-Test**

Im Anschluss an die Ketten-Tests wurden drei Mal pro Bank bzw. nach Abschluss der Test-Wochenenden mit den Bewegungsdaten wurden zweimal pro Bank die Regressions-Tests durchgeführt, um ein einwandfreies Funktionieren des Systems mit den neuen Daten zu gewährleisten.

#### 9.3.5 Einführung und Konsolidierung

Das Cut-Over Wochenende wurde strikt nach dem Zeit- und Ablaufplan der „Dress-Rehearsals“ durchgeführt. Auftretende Fehler wurden protokolliert und, wenn möglich, vor Ort korrigiert. Nach Überprüfung aller Resultate erfolgte eine Gesamtbeurteilung und eine GO- / NOGO-Entscheidung durch den Steuerungsausschuss

*Cut-Over*

Im Rahmen der Konsolidierung wurden aufgetretene Fehler behoben und die im Migrationskonzept definierten „Post-Migration“ Bereinigungen durchgeführt und abgeschlossen. In einer täglich stattfindenden Telefonkonferenz zwischen dem Migrationsteam des Mutterhauses und dem Migrationsteam / den Benutzern der abgeben-

*Konsolidierung*



den Banken wurde der Status der offenen Punkte und allfällig auftretende Probleme behandelt.

## 9.4 Kritische Erfolgsfaktoren

Die drei Banken wurden erfolgreich und zum geplanten Termin in die Bankengruppe integriert. Fünf Aspekte wurden als erfolgsentscheidend identifiziert:

*Management-Support*

- Die Integration der drei Banken hatte innerhalb der Bankengruppe die höchste Priorität bezüglich Ressourcen und finanzieller Mittel. Die notwendige Aufmerksamkeit und Unterstützung durch das Top Management wurde dadurch sichergestellt, dass Mitglieder der Geschäftsleitung aller beteiligten Banken im Steuerungsausschuss saßen. Darüber hinaus wurden die für den Projekterfolg notwendigen Entscheidungen immer termingerecht getroffen.

*Migration eigenes Projekt*

- Die Migration wurde innerhalb des Gesamtprojektes „Fusion“ als eigener Stream bzw. als eigenständiges (Teil-)Projekt aufgesetzt und mit den entsprechenden Ressourcen versorgt.

- Dadurch, dass sich im Stream „Operational Integration“ zwei Teams in enger Zusammenarbeit mit den bankfachlichen Aspekten und der technischen Migration befassten sowie durch externe Mitarbeiter mit vertieftem Migrations-Know-how verstärkt wurden, konnte eine „optimale“ Teamzusammensetzung erreicht werden.
- Dank frühzeitiger Datenbereinigungen auf den Quell-Systemen konnte eine hohe Datenqualität für die eigentliche Datenmigration erreicht werden.
- Durch die Verwendung bestehender Prozeduren (Online-Transaktionen und Batch-Abläufe) konnte das Migrations-Risiko minimiert und die Datenqualität erhöht werden.

*Teamzusammensetzung*

*Frühzeitige Datenbereinigung*

*Verwendung bestehender Prozeduren*



## 10 Literaturverzeichnis

Ackermann, E.; Winter, A.; Gimmich, R. (2005): Reference Migration Process ReMiP – Ein Referenzprozess der Softwaremigration. Vortrag beim 2. Workshop Reengineering Prozesse - Software-Migration, Chemnitz, 7.10.2005. <http://www.proetcon.de/migration/Images/ackermann.pdf>, Zugriff am 21.5.2007.

D. Aebi (1997): A process model for re-engineering, migration and multi-use of business data. In: *csmr*, p. 106, 1st Euromicro Working Conference on Software Maintenance and Reengineering (CSMR '97).

Albrecht, S. (2002): Datenmigration – Ein unterschätztes Thema. Vortrag beim Arbeitskreis Objekttechnologie Norddeutschland, 11.11.2002.

Alyokhin, Vadim (2005): Systembewertung: Metriken und Prozess. Vortrag im Seminar Management von Softwaresystemen, Technische Universität München. [www4.in.tum.de/lehre/seminare/hs/WS0506/mvs/files/Vortrag\\_Alyokhin.pdf](http://www4.in.tum.de/lehre/seminare/hs/WS0506/mvs/files/Vortrag_Alyokhin.pdf), Zugriff am 13.3.2007.

Aminian, H.A. (2003): The Legacy System Dilemma: Making the Right Choices. Technical Report. Insurity.

Bär, H.; Bauer, M.; Ciupke, O.; Demeyer, S.; Ducasse, S.; Lanza, M.; Marinescu, R.; Nebbe, R.; Nierstrasz, O.; Richner, T.; Rieger, M.; Riva, C.; Sassen, A.-M.; Schulz, B.; Steyaert, P.; Tichelaar, S.; Weisbrod, J. (1999): The FAMOOS Object-Oriented Reengineering Handbook. FZI Publikation.

Bennett, K. (1995): Legacy Systems: Coping with Success. In: IEEE Software, S. 19 – 22, Jan. 1995.

Berbner, R.; Repp, N.; Heckmann, O.; Steinmetz, R. (2006): SOA zum Durchbruch verhelfen. In: IT Management, 3/2006, S. 22-27.

Alessandro Bianchi, Danilo Caivano, Vittorio Marengo, Giuseppe Visaggio (2003): "Iterative Reengineering of Legacy Systems," *IEEE Transactions on Software Engineering*, Vol. 29, Nr. 3, S. 225-241.

Bisbal, J.; Lawless, D.; Wu, B.; Grimson, J. (1999a): Legacy Information Systems: Issues and Directions. IEEE Software, Vol. 16, Heft 5, S. 103-111.

Bisbal, J.; Lawless, D.; Wu, B.; Grimson, J. (1999b): Legacy Information System Migration: A Brief Review of Problems, Solutions and Research Issues. Technical Report TCD-CS-1999-38, Computer Science Department, Trinity College Dublin.

Bisbal, J.; Lawless, D.; Wu, B.; Grimson, V.; Richardson, R.; O'Sullivan, D. (1999c): A Survey of Research into Legacy Systems Migration.

BMI (2006): Wirtschaftlichkeitsbetrachtung für Software-Migrationen. [http://www.kbst.bund.de/cIn\\_046/nn\\_910238 / SharedDocs/Anlagenbstbst/Wirtschaftlichkeit/ wirtschaftlichkeitsbetrachtungen\\_\\_fuer\\_\\_software \\_\\_migrationen,templateId=raw,property=publicationFile.p df/wirtschaftlichkeitsbetrachtungen\\_fuer\\_ software\\_migrationen.pdf](http://www.kbst.bund.de/cIn_046/nn_910238/SharedDocs/Anlagenbstbst/Wirtschaftlichkeit/wirtschaftlichkeitsbetrachtungen__fuer__software__migrationen,templateId=raw,property=publicationFile.pdf/wirtschaftlichkeitsbetrachtungen_fuer_software_migrationen.pdf)

Boehm, B.W. (1991): Software Risk Management: Principles and Practices. In: IEEE Software 8, 1, S. 32-41.

Borchers, J. (1997): Erfahrungen mit dem Einsatz einer Reengineering Factory in einem großen Umstellungsprojekt. In: HMD, 34. Jg., Heft 194, S. 77-94.

Brodie, M. L.; Stonebraker, M. (1995): Migrating Legacy Systems: Gateways, Interfaces & the incremental approach. Morgan Kaufmann Publishers Inc., San Francisco.

CAP debis GEI (1992): ESW – Existing Systems Workbench. Produktinformation.

Dewal, S.; Schnichels, L. (2000): Bank 2010: Eine fachliche und technische Vision. In: Softwaretechnik-Trends, Vol. 20, Nr. 3.

Dueck, G. (2007): Erst BOA dann SOA. In: Information Week, 9.3.2007. <http://www.informationweek.de/soa/>, Zugriff am 7.5.2007.

Eberhardt, T.; Rüdiger, A. (2006): IT-Infrastruktur migrieren und dabei sparen. In: Information Week, 9.2.2006.

Eick, S.; Graves, T.; Karr, A.; Marron, J.; Mockus, A. (2001): Does code decay? Assessing the evidence from change management data. In: IEEE Transactions on Software Engineering, Vol. 27, No. 1, S. 1-12.

Erdle, C. (2005): Legacy Migrationsstrategien. Vortrag im Hauptseminar "Management von Softwaresystemen", TUM, München, 13.12.2005.

Feathers, M. C. (2004): Working Effectively with Legacy Code. Prentice Hall.

Fein, R. (1998): Migration von Host-Systemen in eine Client-Server-Welt: Strategie statt Risiko. In: ExperPraxis 98/99, S. 55-60.

Fjeldstad, R.K.; Hamlen, W.T. (1979): Application Program Maintenance Study. In: Proceedings of GUIDE 48, Philadelphia.

Fischbach, R.; Lachmann, I.; Winnemuth, M. (2002): Altdaten-Migration wird oft unterschätzt. In: Computerwoche, 10/2002. [http://korinar/presse/computerwoche\\_2002\\_10.html](http://korinar/presse/computerwoche_2002_10.html), Zugriff am 14.3.2002.

Gimmich, R. (2006): SOA Migration – Approaches and Experience. In: Informatik Bericht Nr. 2/2006, Mainzer Informatik Berichte.



Gimmich, R.; Winter, A. (2005): Workflows der Software-Migration. WSR, 2005, Softwaretechnik-Trends, 25:2, Mai 2005.

Halbich, O. (2007): Banking von der Stange. In: Information Week, 8.3.2007. <http://www.informationweek.de/soa/>; Zugriff am 7.5.2007.

Houdek, F.; Metzker, E.; Omasreiter, H. (2006): Vorlesung „Management von Softwareprojekten“, Folie 435. [http://www.informatik.uni-ulm.de/pm/fileadmin/pm/home/frank/pm\\_slides\\_kap6.pdf](http://www.informatik.uni-ulm.de/pm/fileadmin/pm/home/frank/pm_slides_kap6.pdf); Zugriff am 12.7.2007.

Hunter, R. et al. (2006): High Value, High Risk: Managing the Legacy Portfolio. Gartner, USA.

Informatica (2003): Accelerate Application Data Migration with Informatica. Solution Brief, Informatica.

Jacobson I., Lindström F. (1991): Reengineering of old systems to an object-oriented architecture. In: ACM SIGPLAN Notices 26(11): 340-350.

Jung, M. (2007): Gartner: SOA in 50% aller Anwendungen. In: Information Week, 25.4.2007

<http://www.informationweek.de/news/showArticle.jhtml;jsessionid=ODKAL2ZDKDYRSQSNLPSKH0CJUNN2JVN?articleID=199201558>; Zugriff am 7.5.2007.

Kapoor, R.; Stroulia, E. (2001): Mathaino: Simultaneous Legacy Interface Migration to Multiple Plattformen. In: Proceedings of the 9<sup>th</sup> International Conference on Human-Computer Interaction, 5-10 August, New Orleans.

Kroha, P.; Rosenhainer, L. (2006): Textuelle Anforderungen und Software-Migration. In: Informatik Bericht Nr. 2/2006, Mainzer Informatik Berichte.

Kufer, J.; Bereszkeski, M. (2007): Kernbanksysteme kein Tabu mehr. In: Information Week, 8.6.2007, <http://www.informationweek.de/schwerpunkt/showArticle.jhtml;jsessionid=JR1OCV0WCFATCQSNLPCCKHSCJUNN2JVN?articleID=199902689>, Zugriff am 30.7.2007.

Lehman, M.M.; Belady, L. (1985): Program Evolution: Processes of Software Change. London.

Lientz, B. P.; Bennet, P.; Swanson, E. B.; Burton, E. (1980): Software Maintenance Management. A Study of the Maintenance of Computer Application Software in

487 Data Processing Organisations. Addison Wesley, Reading.

Merlo, E.; Girard, J.; Kontogiannis, K.; Panangaden, P.; Mori, R.D. (1993): Reverse engineering of user interfaces. In: Proceedings of the Working Conference on Reverse Engineering, Baltimore, May 1993, S. 171-179.

Miller, H. (1998): Reengineering Legacy Software Systems. Digital Press, Boston.

Morris, J. (2006): Practical Data Migration. British Computer Society, Swindon.

o.V. (2007): IT bei Finanzdienstleistern: SOA soll's richten. <http://www.cio.de/financeit/strategien/834118/index1.html>, Zugriff am 8.5.2007.

Parareda, B.; Pizka, M. (2006): Reengineering Webbasierter und anderer junger Systeme - Erfahrungsbericht. Technical report, itestra GmbH, Garching, Germany. <http://www4.in.tum.de/~pizka/mp06f.pdf>; Zugriff am 12.7.2007.

Radding, U. (2005): Datenmigration: Handarbeit hat ausgedient. In: Information Week, 28.4.2005. Zugriff am 29.4.2007.

Püttner, C. (2007): IT bei Finanzdienstleistern: SOA soll's richten. In: CIO Magazine, 28.3.2007. [http://www.cio.de/\\_misc/article/printmicrosite/index.cfm?pid=345&pk=834118&IM](http://www.cio.de/_misc/article/printmicrosite/index.cfm?pid=345&pk=834118&IM), Zugriff am 9.5.2007.

Ross, S. (2005): Migrationen in Versicherungsunternehmen – Systeme erneuern, Portfolios qualifizieren, Bestände migrieren. Vortrag beim 2. Workshop Reengineering Prozesse – Software Migration, Chemnitz, 7.10.2005. <http://www.proetcon.de/migration/Images/ross.pdf>, Zugriff am 21.5.2007.

Schneider, T.: (2003): Datenmigration mit PROMET-SSW bei CWS. Vortrag bei der Veranstaltung „Praxisberichte“, 13.11.2003, Zürich.

Seacord, R. (2002): Modernizing Legacy Systems. In: [news@sei](mailto:news@sei) interactive, 4 Q. 2002. <http://interactive.sei.cmu.edu>. Zugriff am 12.7.2007.

Seacord, R.; Plakosh, D.; Lewis, G. (2003): Modernizing Legacy Systems: Software Technologies, Engineering Processes and Business Practices. Addison-Wesley, New York, 2003.

Shaw, M. (1995): Architecture Issues in Software Reuse: It's Not Just the Functionality, It's the Packaging. In: Proceedings of the IEEE Software Symposium on Software Reusability, April 1995.

Sneed, H. (1995): Planning the Re-engineering of Legacy Systems. In: IEEE Software, Band 12, Nr. 1, S. 24-34.

Sneed, H. (1999): Objekt-orientierte Softwaremigration. Addison Wesley Longman, Bonn.

Sneed, H.; Kaposi, M. (1990): A Study of the Effect of Reengineering upon Software Maintainability. In: Proceedings of the 6<sup>th</sup> International Conference on Software Maintenance, San Diego, Nov. 1990, S. 223-232.

Sneed, H.; Hasitschka, M.; Teichmann, M.-T. (2005): Software-Produktmanagement. Dpunkt.Verlag, Heidelberg.

Softek (2006): The Hidden Costs of Data Migration.

Softwarekompetenz (2007): [www.softwarekompetenz.de](http://www.softwarekompetenz.de),  
Zugriff am 13.5.2007.

Uhlig, D. (2006): Eine integrierte Entwicklungsumgebung (IDE) für die BS2000-Migration auf der Basis von Eclipse. In: Informatik Bericht Nr. 2/2006, Mainzer Informatik Berichte.

Weiderman, N.; Nelson, H.; Bergey, J.K.; Smith, D.B. (1997): Approaches to Legacy System Evolution. Technical Report, CMU/SEI-97-TR-014, Carnegie Mellon University, Pittsburgh, PA, USA.

Willinger, M.; Gradl, J. (2003): Datenmigration in SAP R/3. Galileo Press.

Winter, A.; Ziemann, J. (2005): Model-based Migration to Service-oriented Architectures.

Wölfle, T.; Jäschke, C. (2006): Legacy in der SOA Welt. In: Information Week, 19.10.2006. <http://informationweek.de/showArticle.jhtml?articleID=194400186&queryText=Migr>; Zugriff am 29.4.2007.

Wolff, T. (2006): Schlechtes IT-Alignment schwächt Banken. In: [www.CIO.de](http://www.CIO.de), 20.7.2006, <http://www.cio.de/financeit/strategien/824853/index.html>, Zugriff am 7.5.2007.

Wolff, T. (2007a): Flexible Banken mit SOA. In: [www.CIO.de](http://www.CIO.de), 15.2.2007, <http://www.cio.de/financeit/strategien/832750/index.html>, Zugriff am 8.5.2007.

Wolff, T. (2007b): Neue IT-Lösungen für Banken. In: [www.CIO.de](http://www.CIO.de), 21.2.2007, <http://www.cio.de/financeit/strategien/832864/index.html>, Zugriff am 8.5.2007.

Wu, B.; Lawless, D.; Bisbal, J.; Richardson, R.; Grimson, W.; Wade, V.; O'Sullivan, D. (1997): The Butterfly Methodology : A Gateway-free Approach for Migrating Legacy Information Systems. In: *iceccs*, Third IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '97), S. 200 - 206.

Zacher, M.; Fritsch, W. (2007): Die SOA-Evolution. In: *Information Week*, 12.4.2007. <http://www.informationweek.de/soa/>; Zugriff am 7.5.2007.

Ziemann, J.; Leyking, K.; Kahl, T.; Werth, D. (2006): Enterprise Model driven Migration from Legacy to SOA. In: Gimmich, R.; Winter, A.: Workshop Software-Reengineering and Services, MKWI, Passau, 2006.

Zoufaly, F. (2002): Issues and Challenges Facing Legacy Systems. <http://www.developer.com/mgmt/article.php/1492531>; Zugriff am 2.5.2007.