

Univ.-Prof. Dr. Ulrike Baumöl
Dr. Martina Meschke

Informationsmanagement

Kurseinheit 4:
Architekturen und Integration

Fakultät für
**Wirtschafts-
wissenschaft**

9611711

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Übersicht zum Kurs 41760 „Informationsmanagement“

Kurseinheit 1:	Grundlagen des Informationsmanagements
Kurseinheit 2:	Informationslogistik: Entscheidungsunterstützung
Kurseinheit 3:	IT-Governance: Planung, Steuerung, Kontrolle
Kurseinheit 4:	Architekturen und Integration
Kurseinheit 5:	IT-Sicherheitsmanagement
Kurseinheit 6:	IT als Enabler

9611711

Inhaltsübersicht

Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	VII
Abkürzungsverzeichnis	VIII
1 Einführung	1
1.1 Architektur als Ordnungsrahmen für die Gestaltung von Organisationen	1
1.2 Inhalte und Lernziele der Kurseinheit.....	2
2 Architekturen.....	5
2.1 Grundlagen und Begriffe.....	5
2.2 Applikationsarchitektur.....	11
2.2.1 Ziele und Aufgaben der Applikationsarchitektur	11
2.2.2 Komponenten der Applikationsarchitektur.....	13
2.3 Architekturmodelle	19
2.3.1 Ziele und Aufgaben von Architekturmodellen	19
2.3.2 Ausgewählte Architekturmodelle	21
2.4 Übungsaufgaben.....	31
3 Integration.....	32
3.1 Begriff und Perspektiven der Integration	32
3.1.1 Automationsgrad.....	33
3.1.2 Integrationsrichtung	33
3.1.3 Integrationsreichweite.....	34
3.1.4 Integrationsgegenstand	35
3.2 Ziele und Probleme der Integration.....	38
3.3 Integrationslösungen	41
3.3.1 Enterprise Application Integration.....	41
3.3.2 Serviceorientierte Architektur.....	51
3.4 Übungsaufgaben.....	58
4 Architekturmanagement.....	59
4.1 Ziele und Aufgaben des Architekturmanagements	59
4.2 Architekturlebenszyklus.....	62
4.2.1 Dokumentation.....	62
4.2.2 Analyse	64

4.2.3	Planung	67
4.2.4	Entwicklung	70
4.2.5	Steuerung	71
4.3	Vorgehensmodelle zum Architekturmanagement	81
4.3.1	IT-Architektur-Engineering	81
4.3.2	Management von IT-Architekturen	82
4.3.3	The Open Group Architecture Framework - TOGAF	84
4.3.4	Architecture Capability Maturity Model - ACMM	87
4.4	Übungsaufgaben	91
5	Zusammenfassung	92
	Literaturverzeichnis	94
	Lösungen zu den Übungsaufgaben	100

Abbildungsverzeichnis

Abbildung 1: Einordnung der KE in das „Informationsmanagement“	3
Abbildung 2: Architekturen im Unternehmen	7
Abbildung 3: Übergang von einem Ist- in einen Soll-Zustand	19
Abbildung 4: Strategische IT-Planung und Architektur	19
Abbildung 5: Zachman Framework for Enterprise Architecture	23
Abbildung 6: Ganzheitliches Modell der Informationssystem-Architektur	24
Abbildung 7: Architekturpyramide von Dern	27
Abbildung 8: Geschäftsarchitektur	27
Abbildung 9: Informationsarchitektur	28
Abbildung 10: IT-Architektur	29
Abbildung 11: IT-Basisinfrastruktur	30
Abbildung 12: Perspektiven der integrierten Informationsverarbeitung	33
Abbildung 13: Integrationsrichtung	34
Abbildung 14: Integrationsreichweite	35
Abbildung 15: Integrationsebenen und Funktionalitäten von EAI-Systemen	42
Abbildung 16: Hub-and-spoke Architektur	47
Abbildung 17: Bus-Architektur	48
Abbildung 18: Punkt-zu-Punkt Anbindung der Bosch Gruppe	50
Abbildung 19: Neue Architektur der Bosch Gruppe	51
Abbildung 20: SOA-Ebenen und -sichten	54
Abbildung 21: IT-Strategie-Framework	60
Abbildung 22: Architekturlebenszyklus	62
Abbildung 23: Marktanteils-/Marktwachstums-Portfolio einer Applikationsarchitektur	73
Abbildung 24: Marktattraktivitäts-/Wettbewerbsvorteilsportfolio einer Applikationsarchitektur	75
Abbildung 25: Kontext der CBAM	77
Abbildung 26: Kosten-Nutzen-Diagramm	78
Abbildung 27: Ziele und Wechselwirkung der Applikationsintegration	79
Abbildung 28: Veränderungskreislauf der IT-Architektur	81
Abbildung 29: Modell für das Management von IT-Architekturen	83
Abbildung 30: Balancedreieck	84

Abbildung 31: TOGAF Architecture Development Method	86
---	----

9611711

Tabellenverzeichnis

Tabelle 1: Einordnung der Architektur-Terminologie	10
Tabelle 2: Applikationsarchitekturkomponenten und Funktionsbereiche	16
Tabelle 3: Nutzen und Probleme von EAI-Systemen	46
Tabelle 4: Rollen im Architekturmanagement	64
Tabelle 5: Bereiche für die Architekturanalyse am Beispiel der Applikationsarchitektur	66
Tabelle 6: Architekturprinzipien des Innenministeriums NRW	68
Tabelle 7: Kennzahlen für die Applikationsintegration	79
Tabelle 8: Strukturierte Metriken zur Applikationssteuerung	80
Tabelle 9: ACMM-Methode zur Berechnung des gewichteten Mittels	89
Tabelle 10: ACMM-Methode zur Berechnung des prozentualen Anteils	89
Tabelle 11: Fokussierung der Architekturmanagementphasen	90

Abkürzungsverzeichnis

ACMM.....	Architecture Capability Maturity Model
ADM.....	Architecture Development Method
APO	Advanced Planer and Optimiser
APS	Advanced Planning & Scheduling
ARIS	Architektur integrierter Informationssysteme
ATAM.....	Architecture Trade-off Analysis Method
BCG	Boston Consulting Group
BSC.....	Balanced Scorecard
CBAM	Cost Benefit Analysis Method
CIO	Chief Information Officer
CRM	Customer Relationship Management
DV.....	Datenverarbeitung
DW.....	Data-Warehouse
EAI.....	Enterprise Application Integration
EC	Electronic Commerce
EDI.....	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
EIS	Executive Information Systems
ERP	Enterprise-Resource-Planning
ETL.....	Extraktion, Transformation, Laden
IS.....	Informationssystem
ISA	Informationssystemarchitektur
IT.....	Informationstechnik
KE.....	Kurseinheit
KM.....	Knowledge Management
MDM	Master Data Management
SAAM.....	Scenario-based Architecture Analysis Method
SCM.....	Supply Chain Management
SE.....	Softwareentwicklung
SIB	Standard Information Base
SOA	Serviceorientierte Architekturen

TOGAF	The Open Group Architecture Framework
WMS	Workflow-Management-Systeme
XML.....	Extensible Markup Language

9611711

Diese Seite bleibt aus technischen Gründen frei!

9611711

1 Einführung

Vielleicht kein anderes Thema wird im Kontext der Informatik in Unternehmen so kontrovers und teilweise emotional diskutiert, wie Architektur und Architekturmanagement. Die Befürworter sehen darin die Lösung nahezu aller fachlichen und technischen Problemstellungen. Die ablehnende Seite empfindet die Arbeit in und an der Architektur als Treiben im Elfenbeinturm ohne konkreten Nutzen. Die Wahrheit wird, wie so oft, irgendwo in der Mitte liegen.

Unter dem Begriff der Architektur werden Konzepte verschiedener Reichweite gefasst. Das weitreichendste Konzept ist das der „Enterprise Architecture“, das von der Architektur des Geschäftsmodells bis hin zur Architektur der technischen Infrastruktur alles umfasst und idealerweise in ein konsistentes Gesamtmodell fassen will. Als Subkonzepte existieren Architekturen zu Prozessen, Applikationen, Daten, Schnittstellen und anderen relevanten Teilbereichen der Organisation.

Unerheblich, ob nun eine Gesamt- oder nur eine Teilsicht eingenommen wird, die Zielsetzung der Beschäftigung mit Architektur ist es, ein systematisches Abbild bestimmter Teilbereiche in der Organisation zu erhalten, um diese analysieren und weiterentwickeln zu können. Dazu ist es erforderlich, ähnlich wie durch eine Landkarte, eine Sicht auf das konkrete „Gelände“ zu erhalten, in dem Veränderungen geplant werden. Nur so ist es möglich, die Potenziale, aber auch die Konsequenzen zu verstehen.

In dieser Kurseinheit (KE) wird also das Thema Architektur nicht nur als Planungsinstrument, sondern auch als Instrument organisationaler Veränderungsprozesse diskutiert. So wird dem „wahren“ Charakter dieses scheinbar „trockenen“ Themas Rechnung getragen und gezeigt, dass es höchst spannend ist, sich damit auseinanderzusetzen und die Prinzipien zur Anwendung zu bringen.

1.1 Architektur als Ordnungsrahmen für die Gestaltung von Organisationen

Nicht erst mit dem nahezu inflationär verwendeten Begriff der „Digitalisierung“ stehen Unternehmen vor der Herausforderung, ihre Organisation zielorientiert zu gestalten. Zielorientiert bedeutet dabei, dass alle Strukturen auf die Ziele des Geschäftsmodells ausgerichtet sind und dabei entsprechend optimiert zusammenspielen. Mit den Entwicklungen rund um die Digitalisierung nimmt die technische Seite der Architektur eine wichtigere Rolle ein, denn sie ist noch mehr in die Wertschöpfung integriert als es vielleicht schon zuvor der Fall war. Eine gegenseitige Abstimmung von fachlichen Anforderungen und technischem Potenzial kann eine zentrale Aufgabe der Architektur in diesem Kontext sein.

Das bedeutet gleichzeitig, dass eine Beschränkung auf Teilsichten von Architekturen nicht ausreichend ist. So ist die zuvor bereits erwähnte, umfassende Perspektive der „Enterprise Architecture“ als Ausgangspunkt zu wählen. Nur auf dieser Basis können die Gesamtzusammenhänge, die zwischen der Ausgestaltung des Geschäftsmodells als Geschäftsarchitektur, der umsetzenden Prozessarchitektur,

Rolle der Architektur
in der Organisations-
gestaltung

der technischen Architektur, aber auch der Architektur der kulturellen Artefakte und des Steuerungssystems liegen, abgebildet werden.

Die **Komplexität und Vielgestaltigkeit** dieser Aufgabe lässt sich nur durch stabile Strukturen realisieren. Die entstehenden Teilarchitekturen, aber auch die letztlich entstehende Gesamtarchitektur - die Enterprise Architecture - bieten einen visualisierten Ordnungsrahmen, der helfen soll, die Komplexität durch die **Integration** von Prozessen, Funktionen und Daten zu reduzieren und damit gleichzeitig die **Flexibilität** in Anbetracht der Herausforderungen zu erhöhen (vgl. Baumöl und Winter 2003, 46 f.).

Architekturbegriff

Der Begriff **Architektur** stammt aus dem Griechischen und ist aus den Wörtern [arché] = Anfang, Ursprung, Grundlage, das Erste und [techné] = Kunst, Handwerk, auch aus dem Lateinischen [tectum] = Gebäude zusammengesetzt. Architektur enthält also einerseits die Bedeutung für ein gestaltprägendes Element, andererseits kann damit auch eine strukturell organisierte Beziehung von materiellen Elementen beschrieben werden (vgl. Schmidt 2009, S. 15).

In der Informatik beschreibt die Architektur das Zusammenspiel von Komponenten eines komplexen Systems, wie Rechnersysteme, Informationssysteme (IS) oder Applikationssysteme. Die Modellierung solch komplexer Systeme kann aus unterschiedlichsten Zwecken und Sichten erfolgen. Durch eine hierarchische Bildung von Modellierungsebenen und der Differenzierung verschiedener Modellierungssichten wird versucht, dieser Komplexität entgegenzuwirken.

Informationssystemarchitekturen

Im Rahmen des Informationsmanagements beschäftigt sich diese KE mit Informationssystemarchitekturen. Dabei betrachten wir - laut zugrunde liegender Definition - nicht ausschließlich die technischen Komponenten eines Informationssystems, sondern ebenso die fachlichen und organisatorischen Komponenten. Die unterschiedlichen Ebenen und Sichten werden in Kapitel 2 näher erläutert und die differierende Terminologie in der heutigen Literatur zum Architekturthema aufgezeigt.

Eine weitere zentrale Problemstellung im Kontext von Informationssystemarchitekturen ist die Integration von Prozessen, Funktionen und Daten. Aus diesem Grund beschäftigt sich Kapitel 3 mit den Zielen und Aufgaben der Integration sowie deren besonderen Herausforderungen. Anschließend werden ausgewählte Integrationslösungen vorgestellt.

1.2 Inhalte und Lernziele der Kurseinheit

Wie Abbildung 1 zeigt, findet sich die KE „Architektur und Integration“ im Bereich *Management der Informatik* wieder. Das hat seine Begründung darin, dass das Architekturmanagement, auch wenn es Aufgabenstellung aus den Fachbereichen, wie z. B. die Erhebung und Dokumentation der Prozessarchitektur, bearbeitet, auch heute noch häufig in der Informatik angesiedelt ist. Es muss aber festgehalten werden, dass es eine Brückenfunktion hat und dabei das Potenzial, Fachbereiche und die Informatik als heute oftmals noch zwei getrennte Welten, zu ver-

binden, indem es durch die Modellierung der Strukturen und Zusammenhänge ein Verständnis für die Aufgabenstellungen und Möglichkeiten der Entwicklung aufzeigt.

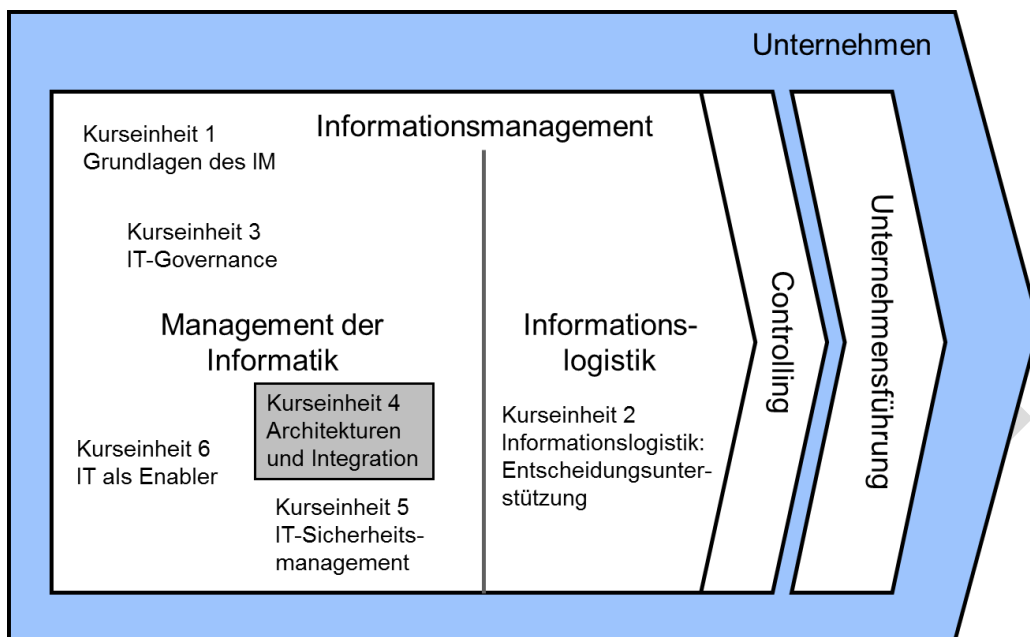


Abbildung 1: Einordnung der KE in das „Informationsmanagement“

Nach einer kurzen Einführung der begrifflichen Grundlagen und dem Aufzeigen von grundsätzlichen Architekturebenen, wird auf die Applikationsarchitektur fokussiert, da diese speziell auf die Erfüllung der Geschäftsprozesse ausgelegt ist und somit den wesentlichen Beitrag im Rahmen des Informationsmanagements leistet. Abschließend werden grundlegende Architekturmodelle vorgestellt und kritisch betrachtet.

Die Integration stellt ein primäres Element von Architekturen dar und wird gesondert betrachtet. Dabei wird insbesondere das Problem der Funktions-, Prozess- und Datenintegration angesprochen. Darauf folgend werden Enterprise Application Integration (EAI) und Serviceorientierte Architekturen (SOA) als mögliche Integrationslösungen aufgezeigt.

Das Architekturmanagement soll dafür sorgen, dass die im Unternehmen vorhandenen Architekturen wirksam auf- und ausgebaut werden. In der Literatur existieren bereits verschiedene Vorgehensmodelle zum Architekturmanagement, die hier vorgestellt und beurteilt werden. Vorab wird das Architekturmanagement anhand des Architekturlebenszyklus verdeutlicht und die einzelnen Phasen der Dokumentation, Analyse, Planung, Entwicklung und Steuerung erläutert.

Die **Lernziele** dieser KE lassen sich wie folgt zusammenfassen:

Kapitel 2: Architekturen

- Sie können erklären, was unter Architekturen allgemein und im speziellen Kontext des Informationsmanagements verstanden wird.
- Sie verstehen, welche verschiedenen Begrifflichkeiten für Architekturen in der Literatur zu finden sind und können diese voneinander abgrenzen.
- Sie können die Applikationsarchitektur von anderen Architekturen unterscheiden und die Ziele und Aufgaben sowie mögliche Komponenten der Applikationsarchitektur erklären.
- Sie verstehen, was unter Architekturmodellen verstanden wird und können deren Ziele und Aufgaben erläutern.
- Sie kennen verschiedene Architekturmodelle aus der Literatur und können diese voneinander abgrenzen.

Kapitel 3: Integration

- Sie wissen, was unter dem Begriff Integration verstanden wird und können die verschiedenen Perspektiven der Integration beschreiben.
- Sie kennen die Ziele und Probleme der Integration und können deren Zusammenhang erläutern.
- Sie kennen Enterprise Application Integration und Serviceorientierte Architekturen und können deren Aufbau und Aufgaben erklären.

Kapitel 4: Architekturmanagement

- Sie verstehen, welche Ziele und Aufgaben das Architekturmanagement umfasst und können die einzelnen Lebenszyklusphasen beschreiben.
- Sie können die verschiedenen Vorgehensmodelle zum Architekturmanagement aus der Literatur erklären und voneinander abgrenzen.

2 Architekturen

Architekturen veranschaulichen und modellieren komplexe Strukturen innerhalb eines Unternehmens, wie z. B. Prozesse oder die Aufbauorganisation. Im Folgenden werden unterschiedliche Modellierungsebenen und -sichten für Architekturen verdeutlicht. Des Weiteren soll das bis heute immer noch uneinheitliche Begriffsverständnis in der Literatur aufgezeigt werden. Die Applikationsarchitektur hat aufgrund ihrer Schnittstelle zwischen betriebswirtschaftlicher und technischer Ebene eine besondere Bedeutung und wird in Abschnitt 2.2 erläutert. Abschließend werden in Abschnitt 2.3 ausgewählte Architekturmodelle aufgezeigt. Das von *Zachman* entwickelte Modell stellt dabei eines der ersten Architekturansätze von Informationssystemen dar. Krcmar entwickelte diesen Ansatz weiter und nahm eine ganzheitliche Sicht auf das Informationssystem mit allen vorhandenen Interdependenzen ein. Das umfassende Architekturmodell von *Dern* wird als aktueller Ansatz erläutert.

Modellierung komplexer Strukturen

2.1 Grundlagen und Begriffe

Im Rahmen des Informationsmanagements finden sich unterschiedliche Architekturverständnisse wieder. Die Aufteilung erfolgt dabei in der Regel nach betriebswirtschaftlicher oder technischer Sicht. Die **betriebswirtschaftliche Sicht** betrachtet die prozessualen und organisationalen Aspekte eines Unternehmens. Hierzu zählen z. B. die Ablauf- und Aufbauorganisation, aber auch strategische Aspekte, wie z. B. das Geschäftsmodell als Grundlage für Architekturentscheidungen. Die **technische Sicht** umfasst die Hardware- und Softwarestrukturen als strukturgebende Elemente der Informationstechnik (IT).

Sichten

Anhand dieser groben Einteilung können Architekturen nach unterschiedlichen Abstraktionsgraden eingeordnet werden. Die umfassendste Betrachtung mit dem höchsten Abstraktionsgrad¹ bietet die **Unternehmensarchitektur** (Enterprise Architecture). Hier wird das gesamte Unternehmen von der Strategie über die Geschäftsprozesse bis zur IT abgebildet. Die Unternehmensarchitektur beinhaltet dabei sowohl Organisationsstrukturen, als auch die darunterliegenden Subarchitekturen mit niedrigeren Abstraktionsgraden. Hierzu gehören die Prozessarchitektur, die Applikationsarchitektur sowie die technische Systemarchitektur.

Ebenen

Die **Prozessarchitektur** dient dazu, die Ablauforganisation strukturiert abzubilden. Sie erfasst die Management-, Leistungs- und Unterstützungsprozesse, die die in der Unternehmensarchitektur ebenfalls definierte Strategie unterstützen. Die Strategieebene selber kann ihrerseits gut durch ein Modell abgebildet werden –

¹ Der Abstraktionsgrad einer Architektur erläutert die Genauigkeit, wie die Details der Architektur erfasst werden. Ein hoher Abstraktionsgrad impliziert eine abstraktere Sicht auf das Unternehmen. Hierbei werden viele Details ausgeblendet und so die Komplexität des Modells verringert. Ein niedriger Abstraktionsgrad impliziert viele Details und ist z. B. bei der Softwareentwicklung von Vorteil.

dem Geschäftsmodell. Die Dokumentation einzelner Prozesse ist, je nach Aufgabenstellung, ebenso wichtig, wie die Analyse und Dokumentation des Zusammenwirkens und der gegenseitigen Abhängigkeiten (Interdependenzen). Diese Interdependenzen spielen z. B. bei der Formulierung von Prozesszielen (z. B. Optimierung der Angebotserstellung) eine Rolle, wenn die Prozessanalyse nicht nur für die Ist-Dokumentation, sondern auch für eine Optimierung eingesetzt wird.

Die Unterscheidung in die drei Prozesskategorien ist für die Entscheidungsunterstützung durch die Prozessarchitektur von Bedeutung. Zum einen wird dadurch transparent, um welche Kategorie von Prozess es sich handelt:

- **Managementprozess:** Prozess zur Steuerung des Unternehmens (z. B. Strategieplanung).
- **Leistungsprozess:** Prozess, der in der direkten Wertschöpfung, also er Leistungserstellung eingesetzt wird (z. B. Produktentwicklung).
- **Unterstützungsprozess:** Prozess, der grundlegend für die Leistungserstellung ist, aber nicht direkt beteiligt ist (z. B. Budgeterstellung).

Eine wichtige Funktion im Rahmen der Nutzung der Prozessarchitektur ist das Prozessmanagement. Diese Steuerungsfunktion muss die Rahmenbedingungen und Steuerungshebel definieren, damit die Prozessarchitektur bestimmte Qualitätskriterien erfüllt. So gehören zu diesen Kriterien z. B. die Konsistenz des Modells, die Widerspruchsfreiheit oder auch die Korrektheit. Darüber hinaus muss es die tägliche Nutzung unterstützen und die Weiterentwicklung überwachen. Damit diese beiden Bereiche sichergestellt sind, sind z. B. Kennzahlen für die Prozessführung zu definieren, Rollen im Rahmen der Prozessführung festzulegen und entsprechende Gremien für die übergreifende Kommunikation zu etablieren.

Bei der Abbildung der Prozessarchitektur ist es durchaus üblich, auch die Aufbauorganisation einzubeziehen, z. B. um zeigen zu können, wo die entsprechenden Prozesse ablaufen, also welche Bereiche der Aufbauorganisation mit welchen Prozessen befasst sind.

Die **Applikationsarchitektur** bildet typischerweise die gesamten Applikationen eines Unternehmens und deren Schnittstellen ab. In der Regel werden in Zusammenhang mit den Applikationen auch die Prozesse abgebildet, die unterstützt werden (vgl. Abbildung 2). Damit entsteht eine wichtige Landkarte, die zur Analyse der Effizienz des bestehenden Applikationsportfolios nutzbar ist. Es kann z. B. geprüft werden, ob die Leistungsprozesse angemessen unterstützt werden. Aufgrund ihrer besonderen Bedeutung im Rahmen des Informationsmanagements werden Applikationsarchitekturen in Abschnitt 2.2 näher erläutert.

Die **technische Architektur** beinhaltet die technische Infrastruktur, wie z. B. die Hardware, Software, die nicht als Applikationen eingestuft werden, sondern für die technische Lauffähigkeit genutzt werden, oder Netzwerkstrukturen. Somit wird mit dieser Teilarchitektur eine technische Sicht verfolgt (vgl. Aier und Dogan 2006, 80 ff.; Hafner et al. 2006, 103 f.) Auf dieser Ebene können mehrere verschiedene Architekturmodelle entwickelt und genutzt werden, z. B. Datenar-

chitektur, Schnittstellenarchitektur oder andere.

Abbildung 2 veranschaulicht die Zusammenhänge der Architekturen im Unternehmen.

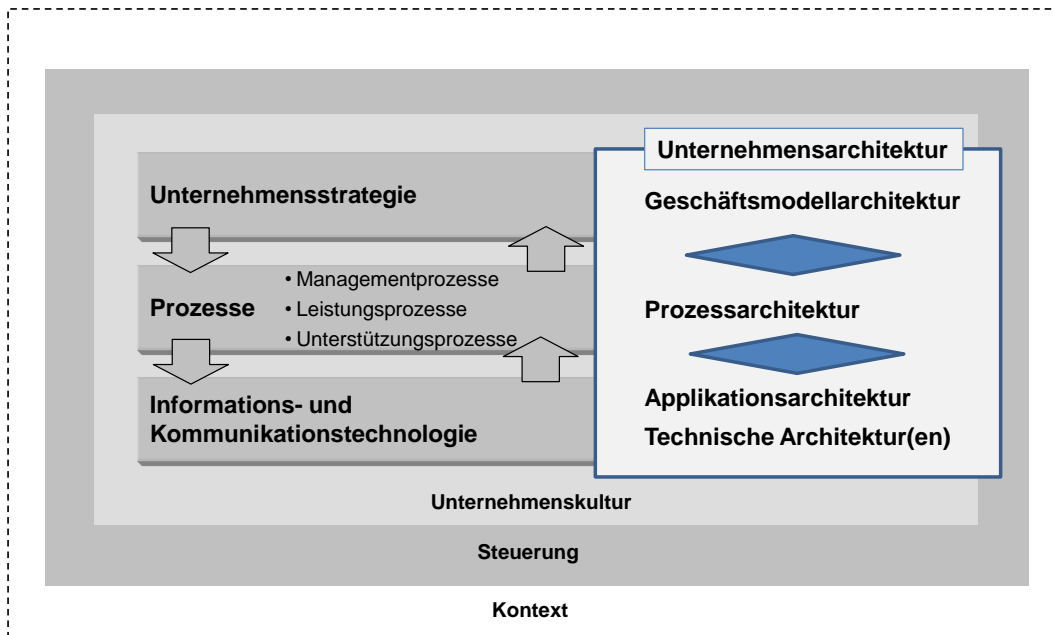


Abbildung 2: Architekturen im Unternehmen

In der Literatur ist eine Vielzahl **unterschiedlicher Begriffsanwendungen und Verständnisse für Architekturen** zu finden, wobei die Begriffe oftmals sowohl synonym als auch homonym² verwendet werden. Im Folgenden werden einige ausgewählte Autoren und deren Terminologieverständnis aufgezeigt. Dieser Schritt ist wichtig, damit die unterschiedlichen Auffassungen, die es geben kann, transparent und damit interpretierbar werden.

Synonyme und
Homonyme

Hafner und Winter (2005, S. 628) unterscheiden vier verschiedene Architekturmodelle. Dazu gehört die Geschäftsarchitektur (Geschäftsmodell), die den Gesamtzusammenhang der Leistungsverflechtung eines Unternehmens beschreibt. Der Gesamtzusammenhang der Leistungsentwicklung, -erstellung und des -vertriebs einer Organisation wird durch die Prozessarchitektur abgebildet. Die Applikationsarchitektur beschreibt die informationsbezogenen Verknüpfungen von Applikationen. Die unterste Ebene bildet die IT-Architektur (technische Architektur), die die gesamten Zusammenhänge von Software- und Datenstrukturen beschreibt.

Hafner und Winter

² Ein Homonym ist das Gegenteil von einem Synonym. Ein Synonym bezeichnet mehrere Worte, die die gleiche Bedeutung haben. Ein Homonym bezeichnet ein Wort, das für unterschiedliche Begrifflichkeiten stehen kann (z. B. Schicht bezeichnet einerseits eine Ebene, andererseits kann aber auch eine gesellschaftliche Schicht oder eine Arbeitsschicht gemeint sein).

- Dern** *Dern* (2006, 16 ff.) beschreibt in seiner Architekturpyramide ebenfalls die gesamte Unternehmensarchitektur mit unterschiedlichen Teilarchitekturen. An der Spitze der Pyramide steht bei *Dern* ebenfalls die Strategie. Darunter befindet sich die Ebene der Geschäftsarchitektur und dann folgt die Informationsinfrastruktur. Darunter liegt die IT-Architektur und die unterste Ebene bildet die IT-Basisinfrastruktur. Jede Architekturebene der Pyramide besteht dabei wieder aus weiteren Teilarchitekturen, die in Abschnitt 2.3.2 im Einzelnen erläutert werden. Die von *Dern* definierte Pyramide entspricht wiederum der Unternehmensarchitektur. Die verschiedenen Ebenen lassen sich nicht ganz trennscharf von der Systematik aus Abbildung 2 abgrenzen. Die Ebene der Geschäftsarchitektur enthält Elemente der Prozessarchitektur und bildet diese in Verbindung mit der Informationsinfrastruktur ab. Die Applikationsarchitektur setzt sich bei *Dern* aus der Informationsinfrastruktur und der IT-Architektur zusammen. Elemente der IT-Architektur sowie die Basisinfrastruktur bilden die technische Architektur ab.
- Krcmar** *Krcmar* (2015, S. 104) beschreibt ebenso unterschiedliche Architekturmodelle in dem umfassenden Modell der „ganzheitlichen Informationssystemarchitektur“ (ISA) (vgl. Abschnitt 2.3.2.2). Er spricht hier von Strategie, Prozess-, Aufbauorganisations-, Anwendungs-, Daten- und Kommunikationsarchitektur sowie der Infrastruktur. Der ISA liegt eine unternehmensweite Betrachtung zugrunde und entspricht damit auch einem Unternehmensarchitekturmodell. Die in Abbildung 2 dargestellten Architekturmodelle lassen sich auch hier deutlich wiederfinden.
- Zachman** *Zachmann* 1987 spricht auch von Unternehmensarchitekturen und entwickelte dazu das Modell „Framework for Enterprise-Architecture“ (vgl. Abschnitt 2.3.2.1). Eine Unternehmensarchitektur versteht er als die beschreibende Darstellung eines Unternehmens, die relevant ist, um den Managementanforderungen gerecht zu werden und auch um das Unternehmen über die benötigte Zeit erhalten zu können. *Zachmans* Unternehmensarchitektur beinhaltet ebenfalls mehrere Teilarchitekturen, die sich in denen zuvor vorgestellten Architekturmodellen wiederfinden.
- Scheer** *Scheer* (1998, 3 ff.) versteht unter dem Begriff Architektur die Komponenten eines Informationssystems, d. h. eines Unternehmens, und deren Beziehungen zueinander. Er entwickelte ausgehend von betrieblichen Vorgangskettenmodellen ein Architekturmodell, welches nach den Sichten Daten, Funktionen, Steuerung und Organisation sowie den Entwicklungsstufen Fachkonzept, DV-Konzept und Implementierung aufgeteilt ist. Subarchitekturen werden nicht explizit angesprochen, entsprechen jedoch den zuvor bereits vorgestellten Architekturmodellen.
- Heinrich** *Heinrich et al.* (2014, S. 85 f.) gehen ebenfalls von mehreren voneinander abhängigen Architekturen aus. Sie beschreiben den Begriff Architektur als „Organisation eines Systems, die sich in seinen Komponenten und deren Beziehungen zueinander sowie zum Systemumfeld manifestiert“ (Heinrich et al. 2014, S. 85). Sie machen darauf aufmerksam, dass der Begriff Architektur im Rahmen des Informationsmanagements nicht nur eine Gliederung oder Struktur beschreibt, wie es bei den Begriffen Hardware- oder Datenbank-Architektur der Fall ist. Die Architekturen sind einerseits durch Modellierungsobjekte, wie z. B. Daten und Tech-

nikkomponenten, und andererseits durch die Sicht des Nutzers, wie z. B. Entwickler-, Anbieter- oder Anwendersicht, gekennzeichnet. Unter IT-Architektur versteht er die Zusammenfassung der Applikationen (Applikationsarchitektur) in einem Unternehmen einschließlich der Basisinfrastruktur (technische Architektur), welche aus den Betriebssystemen, Netzwerken etc. besteht. Als zusammenfassende Bezeichnung für die Architektur der Aufbau- und Ablauforganisation führt er den Begriff Organisationsarchitektur ein.

Ferstl und Sinz (2013, S. 195 f.) interpretieren eine Architektur analog zur Baukunst als Bauplan, der die Komponenten eines Informationssystems (Unternehmensarchitektur inkl. aller Subarchitekturen) spezifiziert und deren Beziehungen und Betrachtungsweisen auf die Komponenten beschreibt. Zudem müssen nach *Ferstl und Sinz* Konstruktionsregeln bei der Bauplanerstellung berücksichtigt werden.

Ferstl und Sinz

Jung (2006, S. 55 f.) versteht unter Architektur ebenfalls „... die Komponenten eines Systems und ihre Beziehungen zueinander..., wobei die verwendeten Konstruktionsregeln (Regeln ... zur Konstruktion von Architekturen mit vorgegebenen Eigenschaften) Bestandteil der Architektur sein können ...“ (Jung 2006, S. 55). Er unterscheidet zum einen zwischen einer Prozessarchitektur (Unternehmens- und Geschäftsprozessarchitektur), die aus aufeinander aufbauenden oder voneinander abhängigen Prozessen besteht sowie aus Organisationseinheiten, die in Beziehung zu den Prozessen stehen können. Zum anderen geht er auf die Applikationsarchitektur ein, die aus Applikationen, Datenverwaltungssystemen, Schnittstellen und Kommunikationsverbindungen besteht. Die Systemarchitektur wird nicht explizit erwähnt.

Jung

Laut *Brenner et al.* (2003, S. 157) sind Architekturen „...Konzepte bzw. Bebauungspläne betriebswirtschaftlicher und technischer Art, die aufzeigen, wie die Strategie realisiert werden soll. Sie beschreiben sowohl die zu realisierenden Elemente als auch die Felder des ‚Nicht Wissens‘.“ Hierbei unterscheiden *Brenner et al.* zwischen der Organisations- (Unternehmens- und Geschäftsprozessarchitektur), der Applikations-, der Technologie- und der Sicherheitsarchitektur (Systemarchitektur).

Brenner et al.

In den Diskussionen zur Bedeutung des Architekturbegriffs ist zu erkennen, dass sie noch zu keinem einheitlichen Begriffsverständnis geführt haben. Z. B. in der gewählten Definition für „Architektur“ von *Ferstl und Sinz* liegt die Überlegung zugrunde, dass jedes System (z. B. ein Gebäude) eine individuelle Architektur besitzt. Eine solche Architektur entsteht in der Regel in zwei Schritten: Zunächst wird im Rahmen eines Konstruktionsprozesses ein Modell erstellt, das anschließend als Vorgabe für den Realisierungsprozess verwendet wird. Bereits im Konstruktionsprozess werden Regeln angewendet (Konstruktionsregeln) um sicherzustellen, dass die spätere Architektur bestimmte, gewünschte Eigenschaften besitzt. Die Konstruktionsregeln werden in der Definition von *Ferstl und Sinz* für den etwas spezielleren Begriff Informationssystemarchitektur als ein „Muss-Element“ betrachtet. Die Informationssystemarchitektur beschreibt die Architektur eines be-

trieblichen Informationssystems (vgl. KE 1) mit seinen Komponenten Organisationsstruktur, Prozesse und Menschen und deren Beziehungen und Interaktionen untereinander. Die für diesen Lehrbrief gewählte Definition schließt sich dieser Definition an.

Definition:
Architektur

Als **Architektur** werden die Komponenten eines Systems (z. B. Informationssystem) und ihre Beziehungen zueinander bezeichnet. Es gibt Konstruktionsregeln, die vorgeben, wie die Komponenten mit Blick auf die angestrebten Eigenschaften der Architektur miteinander in Beziehung zu setzen sind.

In Tabelle 1 werden die vorgestellten Terminologien in die vorab beschriebenen Architektursichten eingeordnet.

	Unternehmens- architektur	Geschäftsprozess- architektur	Applikations- architektur	System- architektur
Hafner und Winter	Geschäfts- architektur	Prozess- architektur	Applikations- architektur	IT-Architektur
Dern	Strategie	Informationsinfrastruktur und IT-Architektur		IT-Basis- infrastruktur
	Geschäftsarchitektur und Informationsinfrastruktur		IT-Architektur und IT-Basisinfrastruktur	
Krcmar	Strategie	Prozess- und Auf- bauorganisations- architektur	Anwendungs-, Da- ten-, Kommunika- tions- architektur	Infrastruktur
Zachman	Unternehmens- architektur	Geschäftsprozess- architektur	Applikations- architektur	System- architektur
Scheer	Unternehmens- architektur	Geschäftsprozess- architektur	Applikations- architektur	System- architektur
Heinrich	Organisationsarchitektur		IT-Architektur	
Ferstl und Sinz	Unternehmens- architektur	Geschäftsprozess- architektur	Applikations- architektur	System- architektur
Jung	Prozessarchitektur		Applikationsarchitektur	
Brenner et al.	Organisationsarchitektur		Applikations- architektur	Technologie- und Sicherheits- architektur

Tabelle 1: Einordnung der Architektur-Terminologie

Die Bedeutung der Applikationsarchitektur, die bereits angeführt wurde, ist auch hier wiederzuerkennen. In den Definitionen findet sie explizite Beachtung als Bindeglied zwischen der Geschäftsprozess- und Systemarchitektur. Daher wird im folgenden Abschnitt ein Überblick zur Applikationsarchitektur gegeben.

2.2 Applikationsarchitektur

Die Applikationsarchitektur beinhaltet die Gesamtheit aller Applikationen in einer Organisation. Sie visualisiert modellartig den Aufbau und das Zusammenspiel aller Applikationen sowie deren Schnittstellen. Hierbei werden der Informationszusammenhang und die Abhängigkeiten in aggregierter Form dargestellt (vgl. Hafner und Winter 2005, S. 627). Der Applikationsbegriff wurde bereits in KE 1 erläutert und wird hier wiederholt. Als **Applikation** wird die Software bezeichnet, die unmittelbar durch die Benutzer verwendet wird und die Aufgabenerfüllung unterstützt. Eine Applikation kann zum einen vom Markt eingekauft werden und damit verschiedene Grade der Standardisierung erreichen (Standardsoftware). Sie kann zum anderen selbstentwickelt sein und damit individuell auf das Unternehmen angepasst werden (Individualsoftware).

Im Folgenden wird die historische Weiterentwicklung von Applikationen sowie deren Ziele und Aufgaben aufgezeigt. Anschließend erfolgen die Erläuterung der Funktion einer Applikationsarchitektur und deren mögliche Komponenten.

2.2.1 Ziele und Aufgaben der Applikationsarchitektur

Die Entstehungsgeschichte der Applikationsarchitektur beginnt bereits in den **70er Jahren**. Applikationen, die Geschäftsprozesse wie den Einkauf, die Produktion oder die Finanzbuchhaltung unterstützten, wurden separat von den auf den zentralen Großrechnern vorhandenen Applikationen ausgeführt. Somit bestand zwischen diesen beiden Systemen keine Integration. In den **80er Jahren** wurde mittels Standardsoftware versucht, den Integrationsgrad zu erhöhen. Eine Standardsoftware ist eine vorgefertigte Applikation, die generelle Funktionen eines Unternehmens abdeckt und in die vorhandene Infrastruktur integriert werden muss. Den hohen Entwicklungskosten der davor bestehenden Individualsoftware konnte damit entgegengewirkt werden. Es konnten jedoch noch nicht alle Geschäftsprozesse umfassend unterstützt werden. Die Applikationen liefen weiterhin auf zentralen Rechnern. Der Einsatz von Client-Server und damit dezentralen Systemen begann in den **90er Jahren**. **Heutzutage** kommt noch die Herausforderung hinzu, die Integration der Informationssysteme und der damit verbundenen Applikationen über Unternehmensgrenzen hinaus zu gestalten, z. B. zur Anbindung von Lieferanten oder Kunden im Bereich Supply Chain Management (SCM) oder Customer Relationship Management (CRM). Hierzu hat die rasante Entwicklung der Internettechnologie beigetragen (vgl. Huber et al. 2002, 167 ff.).

Entstehungsgeschichte der Applikationsarchitektur

Die Planung und Umsetzung einer Applikationsarchitektur ist sehr aufwendig. Ausschlaggebend ist daher, eine wirtschaftliche Planung zur Erreichung der verfolgten Ziele der Applikationsarchitektur vorzunehmen. Hierbei wird zwischen strategischen und operativen Zielen unterschieden.

Die strategischen und damit langfristigen Ziele (ca. fünf bis zehn Jahre) verfolgen eine **Erhöhung der Flexibilität** und der **Anpassungsgeschwindigkeit** an veränderte Anforderungen (z. B. veränderte Prozessabläufe oder Veränderungen im IT-Bereich) sowie eine **Minimierung des Risikos** im Unternehmen. Es soll eine Ge-

Strategische Ziele

samtsicht, durch Bündelung von Funktionalitäten, Verantwortlichkeiten und Daten erreicht werden. Diese Gesamtsicht visualisiert die Zusammenhänge und unterstützt somit ein erfolgreiches Geschäftsprozessmanagement. Die Anpassung (Erweiterung und Änderung) der Applikationslandschaft aufgrund veränderter Anforderungen, kann somit flexibler und schneller gestaltet werden. Durch die visualisierte Gesamtsicht wird auch die Risikominimierung im Rahmen des Geschäftsprozessmanagements unterstützt. Die einzelnen, durch die Applikationen unterstützten Prozesse, können Ineffizienzen aufweisen und müssen im Rahmen der Strategieerreichung dennoch bestehen bleiben. Hier kann die Applikationsarchitektur die Risiken der entstehenden Ineffizienzen aufzeigen und dadurch eine mögliche Risikominimierung verdeutlichen.

Huber *et al.* (2002) sprechen in diesem Zusammenhang auch von Qualitätsmerkmalen, denen die Applikationsarchitektur entsprechen soll. Sie sprechen neben der Flexibilität (Anpassbarkeit) auch von der **Standardisierung (Wiederverwendbarkeit)** und der **Integration**. Die Standardisierung beschreibt die Wiederverwendbarkeit von Architekturkomponenten. Durch den Einsatz von Standardsoftware-Paketen, die von den verschiedenen Organisationen einfach konfiguriert und auf die jeweiligen Bedarfe angepasst werden können, wird eine hohe Standardisierung erreicht. Hierbei werden Templates³ eingesetzt, die den Informationsaustausch der unterschiedlichen Organisationseinheiten vereinfachen. Dadurch ist ebenso eine einfache unternehmensübergreifende Integration von Applikationen möglich. Die Applikationen werden durch entsprechende Schnittstellen einfach und schnell mit anderen Applikationen inner- und überbetrieblich vernetzt und erreichen dadurch einen hohen Integrationsgrad (vgl. auch Winter 2006, S. 3 ff.).

Operative Ziele

Operative Ziele haben einen kurzfristigen Horizont von ca. ein bis fünf Jahren und sind auf die **Optimierung von Kosten, Qualität und Zeit** ausgerichtet. Sie beschreiben hauptsächlich Prozessziele, welche sich insbesondere auf qualitative und zeitliche Aspekte auswirken.

Applikationsarchitekturen optimieren Kosten durch die weitgehende Standardisierung von Architekturkomponenten. Hierbei wird, wie oben bereits erläutert, die Wiederverwendbarkeit gewährleistet. Zudem werden in einer Applikationsarchitektur die Beziehungen der Applikationen mit deren Schnittstellen definiert und dadurch können Kosten gesenkt werden. Die Vorgabe von Komponenten hilft, Fehlinvestitionen zu vermeiden und eine optimale Budgetallokation der einzelnen IT-Projekte sicherzustellen. Eine anschließende Projektbewertung hinsichtlich der Relevanz des IT-Projektes unterstützt nachfolgend bei der kontinuierlichen Pflege, Gestaltung und Erweiterung der Applikationsarchitektur, so dass auch die Folgekosten optimiert werden können.

³ In einem physischen System lassen sich Funktionen, Prozesse und Daten mittels Templates abbilden. Dabei stellen die Templates vorgegebene Vorlagen dar.

Im Bereich der Prozessunterstützung kann durch die Applikationsarchitektur eine Steigerung der Qualität erreicht werden. In der Applikationsarchitektur sind bestimmte Funktionen⁴ konkreten Prozessen zugeordnet und dadurch erfolgt eine Koordination der Applikationen untereinander. Je umfassender die Prozesse vorab analysiert werden, umso besser können die Applikationsfunktionen zur Unterstützung der Geschäftsprozesse bereitgestellt werden. Der daraus erfolgende schnelle und effiziente Ablauf der Geschäftsprozesse reduziert zudem die Durchlaufzeiten und unterstützt bei der Optimierung der Zeit.

Aus den Zielen ergeben sich Aufgaben an die Applikationsarchitektur. Zum einen muss die Applikationsarchitektur im Rahmen der Integration die **Koordination** von Daten, Funktionen und Prozessen gewährleisten (vgl. auch Abschnitt 3.1.4). Durch die Integration von Applikationen werden Geschäftsprozesse mit den dazu benötigten Daten und Funktionen unterstützt. Das bedeutet, dass das Geschäftsprozessmodell, welches sich aus der Strategie des Unternehmens ableitet, auf die IT umgesetzt werden muss. Dies impliziert einen Koordinationsprozess zwischen den einzelnen Funktionen. Es ist wichtig, die Applikationsarchitektur optimal zu strukturieren und zu implementieren. Bei der **optimalen Strukturierung** wird die betriebswirtschaftliche Sicht auf die Geschäftsprozesse vorgenommen. Dabei wird eine zielgerichtete Unterstützung von automatisierbaren Aktivitäten gewährleistet sowie effiziente Applikationsschnittstellen geschaffen. Im Rahmen der **optimalen Implementierung** werden aus technischer Sicht Implementierungskomponenten, wie Module und Datenstrukturen bzw. Objekte, zur Wiederverwendung geschaffen und benutzt (vgl. Winter 2003, S. 107).

Aufgaben der Applikationsarchitektur

Eine weitere Aufgabe von Applikationsarchitekturen liegt in der **Realisierung von Geschäftsprozessen**, die sich im Rahmen einer Unternehmensstrategie ergeben können. Die Geschäftsprozesse können z. B. aus Strategien, wie Lean Management oder Outsourcing, abgeleitet sein. Die Applikationsarchitektur hat hierbei die Aufgabe, einen Rahmen zu bieten, um solche Strategien erfolgreich umsetzen zu können. Diese Umsetzung kann intern sowie unternehmensübergreifend unterstützt werden (vgl. Huber et al. 2002, S. 166 ff.). Hierfür werden in einer Applikationsarchitektur verschiedene Komponenten verknüpft, die im folgenden Abschnitt erläutert werden.

2.2.2 Komponenten der Applikationsarchitektur

Eine Applikationsarchitektur besteht aus verschiedenen Applikationstypen, die miteinander vernetzt sind und die betriebswirtschaftlichen Funktionsbereiche eines oder mehrerer Unternehmen wirksam unterstützen sollen. Applikationen können in Geschäfts- und Universalapplikationen eingeteilt werden. Eine **Geschäfts-**

⁴ Mit einer Funktion ist generell eine (Teil-)Aufgabe oder (Teil-)Zweck einer Applikation gemeint, um einen Prozess bei seiner Durchführung zu unterstützen (z. B. Ermittlung des Mindestbestellwertes).

applikation ist eine Applikation, die zur Unterstützung von Geschäftsprozessen entwickelt wurde und zu diesem Zweck entsprechend eingesetzt wird. Eine **Universalapplikation** ist eine Applikation, die nicht ausschließlich zur Unterstützung von Geschäftsprozessen, sondern auch universell, z. B. im Privatbereich, einsetzbar ist (vgl. KE 1).

Huber et al. (2002, S. 178 ff.) verwenden ebenfalls die Begriffe Geschäfts- und Universalapplikationen. Sie unterscheiden acht Architekturkomponenten, die wichtige Funktionsbereiche im Unternehmen unterstützen. Dabei unterstützen **Geschäftsapplikationen** Geschäftsprozesse durch Automatisierung von Aktivitäten und Schnittstellen (geschäftsprozessorientiert). Zu ihnen gehören z. B. (vgl. hier und im Folgenden auch Alt et al. 2004, S. 44; Puschmann 2003, S. 207):

- **Enterprise Resource Planning (ERP) Systeme:** Diese bilden in der Regel den Kern einer Applikationsarchitektur, da sie die meisten der betriebswirtschaftlichen Funktionsbereiche abdecken. Die großen Firmen, wie SAP, Oracle, Sage oder Microsoft, decken die meisten der in Tabelle 2 aufgelisteten Funktionen ab. Zusätzlich stellen diese Hersteller Module für die Bereiche SCM oder CRM zur Verfügung, die dadurch relativ gut in das bestehende ERP-System integriert werden können (vgl. Huber et al. 2002, S. 179 f.; Mertens et al. 2012, S. 122 ff.).
- **Supply Chain Management (SCM)/Advanced Planning & Scheduling (APS) Systeme:** Die Funktionen, die von diesen Systemen angeboten werden, sind größtenteils auch in ERP Systemen zu finden. Hierzu gehören Funktionsbereiche wie Bedarfsplanung, Bestandsplanung, Produktionsplanung, Distributionsplanung oder Transportplanung. In den vorhandenen APS- bzw. SCM Standardapplikationen sind diese betrieblichen Aufgabenbereiche jedoch wesentlich detaillierter und umfassender implementiert, so dass eine wirtschaftlichere Planung erfolgen kann. Zudem ist die Anbindung an Kunden- oder Lieferantensysteme zum Austausch von Bestands- und Planungsdaten oft nur mittels dieser Applikationen möglich (vgl. Grünauer et al. 2002, S. 197 ff.).
- **Electronic Commerce (EC) Systeme:** Durch diese Systeme besteht für Unternehmen die Möglichkeit, ihre Kunden internetbasiert und relativ unkompliziert anzusprechen und mit ihnen in Interaktion zu treten. EC Applikationen stellen Funktionalitäten im Bereich des elektronischen Handels zur Verfügung, wie Online Shops, die Kataloge und Bestellmöglichkeiten bereitstellen. In diesem Fall wird von Sell-side Electronic Commerce (eSale) gesprochen. Ebenso können zu internen Zwecken Beschaffungsmaßnahmen von vor allem indirekten Gütern vereinfacht werden. Diese Applikationen werden Buy-side Electronic Commerce (eProcurement) genannt (vgl. Alt et al. 2002, 88 ff.).
- **Customer Relationship Management (CRM) Systeme:** Applikationen, die das CRM unterstützen, beinhalten kundenorientierte Funktionen für die Bereiche Marketing, Verkauf und Service. Für den Marketingprozess

stellen CRM Applikationen Informationen zu Kunden, Produkten oder Preisaktionen bereit. Diese Informationen dienen den Mitarbeitern sowie den Kunden eines Unternehmens. Im Verkaufsprozess werden die Kunden durch Produktinformationen oder Call Center-Aktivitäten unterstützt. Problemanalysen, Online Support oder Garantieabwicklungen unterstützen die Kunden bei Serviceprozessen. Dabei kann zwischen analytischen (Geschäftsprozessauswertung) und operativen (Geschäftsprozessunterstützung) CRM Systemen unterschieden werden. CRM Applikationen werden häufig bereits als Zusatzmodule für ERP-Systeme angeboten, die eine möglichst einfache Integration in die bestehende ERP-Landschaft ermöglichen sollen (vgl. Alt et al. 2002, 93 f.).

Universalapplikationen können geschäftsprozessunabhängig eingesetzt werden und unterstützen z. B. Funktionen wie die Datenhaltung, das Wissens- oder Dokumentenmanagement. Zu ihnen gehören z. B.:

- **Master Data Management (MDM) Systeme:** Diese Systeme verwalten benötigte Stammdaten. Um die Daten aus den heterogenen Applikationen zu integrieren, muss eine spezielle Applikation das Anlegen und automatische Verteilen an die entsprechenden Organisationseinheiten übernehmen. Das MDM kann auch von einem ERP System übernommen werden.
- **Data Warehouse (DW) Systeme:** Ein DW-System speichert relevante Daten aus den heterogenen operativen Applikationen. Diese sollen in konsistenter und möglichst redundanzfreier Form den einzelnen Organisationseinheiten oder dem Management von Unternehmen zur Entscheidungsfindung zur Verfügung gestellt werden. Diese Daten fließen in der Regel aus den ERP Systemen ein und werden für Auswertungen von EIS-, EC-, APS- oder CRM Applikationen abgerufen. Das Ziel eines DW ist es, die operativen Daten über längere Perioden hinweg in unveränderter Form zu halten, um einerseits die operativen Applikationen zu entlasten und andererseits Daten aus Vorperioden für Datenanalysen zur Verfügung zu stellen. Hierzu werden die Daten in ein einheitliches Format extrahiert und können aus internen, als auch externen Datenquellen stammen (vgl. Krcmar 2015, 150 ff.; Mertens et al. 2012, S. 39, 50 ff.).
- **Knowledge Management (KM) Systeme:** Applikationen zum KM dienen der Wissensgenerierung und Dokumentation in Unternehmen. Durch Funktionen wie Datenbanken, Archivierungs-, Imaging-, Groupware- oder Workflow Systeme wird die Informationsverwaltung im Rahmen des Wissensmanagements unterstützt. Die Speicherung, Verteilung und das Wiederauffinden der Informationen dient der Wissensvermittlung in Organisationen und muss deshalb einer regelmäßigen Aktualisierung folgen (vgl. Mertens et al. 2012, S. 59).
- **Executive Information Systems (EIS):** EIS dienen der Unterstützung von Managementaufgaben. Sie verarbeiten große Datenmengen, meist gekoppelt an ein DW, um diese in aufbereiteter Form als Informationen zur Ent-

scheidungsunterstützung zur Verfügung zu stellen. Hierbei werden interne und auch externe Informationen verarbeitet. Die Darstellung kann auf unterschiedlichen Ebenen stattfinden. Z. B. ist eine Aufbereitung in aggregierter Form vorstellbar. Ebenso sind die Informationen in graphischer Form abrufbar (vgl. Huber et al. 2002, S. 182).

Tabelle 2 zeigt mögliche Funktionsbereiche und Applikationsarchitekturkomponenten, welche die Funktionsbereiche unterstützen können. Diese Einteilung ist beispielhaft und kann je Unternehmen individuell gestaltet sein.

Applikations- komponenten								
Funktionsbereiche	ERP	SCM/ APS	EC	CRM	MDM	DW	KM	EIS
Entwicklung								
Forschung								
Nachfrageplanung								
Bedarfsplanung								
Produktionsplanung								
Einkauf								
Lagerhaltung								
Produktion								
Distribution/Transport								
Marketing								
Verkauf								
Service								
Personal (Human Relations)								
Lohnbuchhaltung								
Kreditorenbuchhaltung								
Debitorenbuchhaltung								
Interne/Externe FIBU								
Beschaffung indirekter Güter								
Führungsinformationen								
GroupWare								
Dokumentenmanagement								
Datenstandardisierung								
Datensammlung								

Tabelle 2: Applikationsarchitekturkomponenten und Funktionsbereiche

Quelle: Entnommen aus Huber et al. 2002, S. 180

In den vorhergehenden Abschnitten wurden die Grundlagen zu Architekturen und im Speziellen zu Applikationsarchitekturen vorgestellt. Das folgende Beispiel soll eine mögliche Modellierung in der Praxis anhand einer Bankenarchitektur verdeutlichen.

Bankenarchitektur

In diesem Kapitel haben wir die Merkmale und Eigenschaften von Architekturen, sowie die in der Literatur herrschenden verschiedenen Definitionen betrachtet. An dieser Stelle soll beispielhaft die spezielle Form einer Architektur, in diesem Fall die Bankenarchitektur, näher betrachtet werden.

Die Bankenarchitektur ist ein Architekturvorschlag für das Retail Banking, der sich an den Gestaltungsebenen des Business Engineering orientiert und dem Kompetenzzentrum Bankenarchitekturen im Informationszeitalter (CC BAI) zugrunde liegt. Seine Entwicklung ist dadurch begründet, um dem auf die Banken einwirkenden Strukturwandel (bspw. deregulierte Märkte, anspruchsvollere Kunden, variantenreiche Produkte, Eindringen neuer Wettbewerber, Zusammenwachsen von Bank- und Versicherungsleistungen, etc.) entgegen zu wirken (vgl. Heinrich und Leist 2000, 152 ff.; Leist 2002, S. 19 ff.). Sie soll dazu dienen, „Unternehmen bei der Transformation in das Informationszeitalter zu unterstützen“ (Leist 2002, S. 5), indem sie restrukturiert oder neu geschaffen werden.

Der bereits angesprochene Strukturwandel macht eine Untersuchung der bestehenden betriebswirtschaftlichen, organisatorischen und informationstechnischen Systeme erforderlich, um festzustellen, ob die bestehenden Systeme durch Anpassungen beibehalten werden können oder ob eine Neuentwicklung in Betracht gezogen werden muss.

Beim Aufbau der Bankenarchitektur wird zwischen Zustandsmodellen, die die Struktur und das Verhalten einer Bank zu einem bestimmten Zeitpunkt beschreiben, und Vorgehensmodellen, die Maßnahmen und Aktivitäten wiedergeben, „welche die Zustandsmodelle von einem Ist-Zustand zu einem Soll-Zustand überführen“ (Leist 2002, S. 19) unterschieden. Die beiden vorgenannten Modelle werden den so genannten Gestaltungsebenen zugeordnet.

Die Banken-Architektur umfasst drei Gestaltungsebenen: Die Geschäftsebene, die Prozessebene und die Applikationsebene, die im Folgenden näher erläutert werden.

Geschäftsebene

Auf dieser Ebene erfolgt die betriebswirtschaftliche Ausrichtung eines Unternehmens, indem die für den Unternehmensbereich relevanten Fragestellungen gefunden und in einem Geschäftsmodell strukturiert werden. Dazu zählen z. B. Fragen nach zukünftig zu erbringenden Leistungen, sich ändernden Bedürfnissen der Kunden, im Unternehmen vorhandenen Kompetenzen usw.

Neben der Erstellung des Geschäftsmodells erfolgt auf dieser Ebene eine Analyse des Kundenprozesses und der Frage, welche Leistungen vom Unternehmen selbst erbracht werden bzw. welche sich für eine Kooperation anbieten. Die Darstellung der Ergebnisse dieser Analyse, d. h. die Festlegung, was dem Kunden wie angeboten wird, erfolgt innerhalb eines Modells, eines sog. Vertriebswegemix.

Weiterhin wird versucht, für die Strategien, die im Rahmen des Geschäftsmodells festgelegt wurden, geeignete Maßnahmen zur Umsetzung zu ermitteln, die dann in die Prozessmodellierung einfließen.

Prozessebene

Auf dieser Ebene werden die Aufgaben und ihre Abläufe definiert, mit dem Ziel, die Prozessabläufe und die betriebswirtschaftlichen und informationstechnischen Anforderungen abzustimmen. Die Prozessebene bildet innerhalb der Bankenarchitektur die zentrale Ebene zwischen der betriebswirtschaftlichen und informationstechnischen Sicht, da ein verändertes Prozessmodell die Vorgabe für die beiden anderen Ebenen bildet.

Applikationsebene

„Das auf dieser Ebene verfolgte Ziel besteht darin, die Ausführung der Prozesse mit Informationstechnik zu unterstützen und damit die betriebswirtschaftlichen Anforderungen umzusetzen“ (Leist 2002, S. 20). Dazu entwickelte das CC BAI ein Konzept, das sog. PIT-Modell, das die Beziehung zwischen der Prozess- und der Applikationsebene herstellt und das Beziehungsmuster von Prozessen und Komponenten abbildet, und damit zwei wichtige Ziele verfolgt: die Wiederverwendbarkeit und die Transparenz (vgl. Leist 2002, 20 f.).

Zustands- und Vorgehensmodelle

„Entlang der Ebenen wurden im Kompetenzzentrum Zustandsmodelle mit festgelegter Syntax definiert, die in Zusammenarbeit mit den Partnerunternehmen zu unternehmensspezifischen Modellen und auch in Ansätzen als Referenz- bzw. generische Modelle entwickelt wurden“ (Leist 2002, S. 21).

Für jede der drei Gestaltungsebenen sind zwei Arten von Zustandsmodellen entwickelt bzw. verwendet worden – Metamodelle und die jeweiligen Objektmodelle.

Im Bereich der Vorgehensmodelle werden grundsätzlich zwei Arten unterschieden: Das generalisierte Vorgehensmodell beinhaltet die Methoden zur Unterstützung der Soll-Modelle, wohingegen das spezifische Vorgehensmodell die Maßnahmen beinhaltet, die tatsächlich durchgeführt werden sollen (vgl. Leist 2002, 21 ff.).

Für die Vorgehensweise, die vom CC BAI gewählt wurde, war zunächst ein generalisiertes Vorgehensmodell zu entwickeln, das dann durch den Unternehmenskontext spezialisiert wurde (vgl. die nachfolgende Abbildung 3).

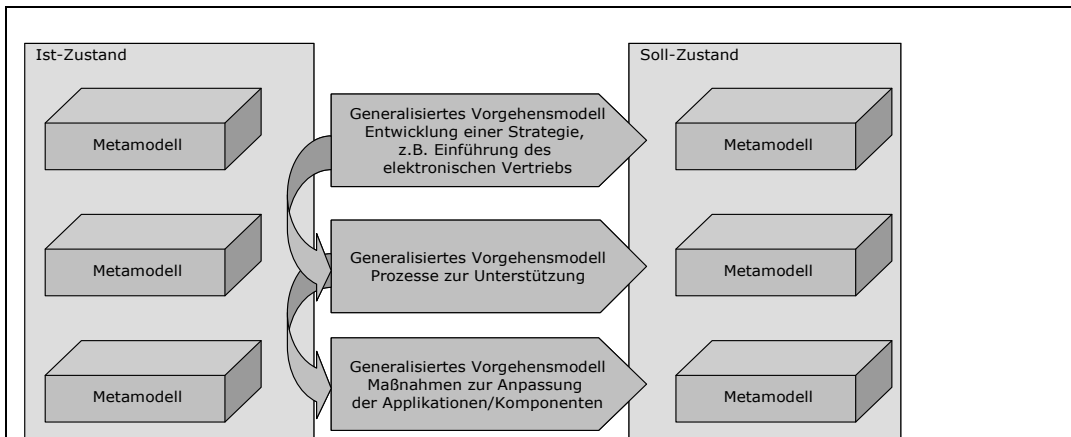


Abbildung 3: Übergang von einem Ist- in einen Soll-Zustand

Quelle: Entnommen aus Leist 2002, S. 24

Die Modellierung kann dabei von allen drei Gestaltungsebenen ausgehen. Der Modellierungsumfang kann dabei eine, zwei oder alle drei Ebenen umfassen. Je nachdem, welche Ebene die Modellierung ausgelöst hat, variiert auch der Umfang der Informationen des Ausgangszustandes.

2.3 Architekturmodelle

In diesem Abschnitt werden einige ausgewählte Architekturmodelle vorgestellt, die in der Wirtschaftsinformatik-Literatur zu finden sind. Vorab wird gezeigt, welche Ziele und Aufgaben mit Architekturmodellen verfolgt werden.

2.3.1 Ziele und Aufgaben von Architekturmodellen

Architekturen sind eng mit der strategischen IT-Planung im Unternehmen verknüpft (vgl. Abbildung 4). So beeinflussen die vorhandenen Architekturen die strategische IT-Planung (**architekturbasierte strategische Planung**), da sie durch ihre Strukturen einen gewissen Handlungsrahmen festlegen. Durch die strategische Planung wird andererseits aber auch die Architektur verändert (**Architektur als Gegenstand strategischer Planung**).

Wechselwirkung zwischen strategischer IT-Planung und Architektur

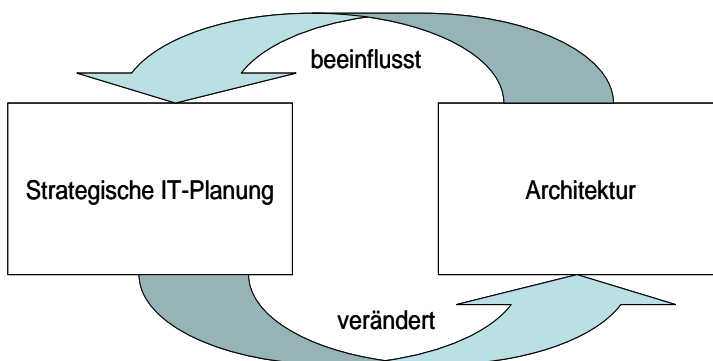


Abbildung 4: Strategische IT-Planung und Architektur

Es ist ersichtlich, dass die Architektur nie unabhängig von der IT-Planung betrachtet werden darf und umgekehrt. Um die daraus gewachsene Komplexität im

Rahmen der Architekturplanung und -entwicklung beherrschbar zu gestalten, wurden bereits zahlreiche Architekturmodelle entwickelt, von denen drei ausgewählt im Folgenden erläutert werden.

Mit Architekturmodellen werden Zielsetzungen verfolgt, von denen nachfolgend einige aufgezeigt werden (vgl. Heinrich et al. 2014, 86 f.; Hofmann und Schmidt 2010, S. 225 ff.).

Ziele von Architekturmodellen

Bereitstellung einer graphischen Präsentation und Realitätsabbildung: Durch die **Visualisierung** von Prozessen, Systemen, Rollen und Aufgaben kann das Unternehmen bzw. können Unternehmensteile realitätsgetreu abgebildet werden.

Verständlichkeit durch Genauigkeit der Abbildung unterschiedlicher Unternehmenskomponenten: Durch die graphische Abbildung von Unternehmenskomponenten wird ein **einheitliches Verständnis** im Unternehmen hergestellt.

Kommunikationsmittel: Ein Architekturmodell dient durch das einheitliche Verständnis auch als Basiswerkzeug für die Kommunikation zwischen den Fachabteilungen und der Informatik. Es hilft den Fachabteilungen, die Anforderungen präziser zu formulieren und der Informatik, alle relevanten Anforderungen zu berücksichtigen, einzuordnen und auf Machbarkeit zu prüfen.

Einfache Benutzung und Veränderungsmöglichkeit: Architekturmodelle helfen, die Komplexität überschaubar zu gestalten. Alle relevanten Komponenten werden abgebildet sowie deren Abhängigkeiten dargestellt. Dadurch ist eine bessere Beherrschbarkeit möglich und Veränderungen sind wirtschaftlicher durchzuführen.

Koordinationsbasis für Applikationen und Technologien: Architekturmodelle bieten auf Ebene der Applikationsarchitektur eine Grundlage für die Integration der unternehmensweiten Applikationen und Technologien.

Erhöhung der **Datenkonsistenz und Datensicherheit:** Durch die Modellierung der z. B. Applikationsarchitektur kann die Wahrung der Datenkonsistenz und Datensicherheit unterstützt werden, da alle Daten und Datenflüsse übersichtlich und vollständig abgebildet werden. Hierbei können Inkonsistenzen oder Synergiepotenziale aufgedeckt werden.

Erhöhung der Planbarkeit der Systeme: Die Planung der Architekturen ist ein komplexer Prozess, der mit Hilfe von Architekturmodellen, insbesondere auf der Ebene der Systemarchitektur, übersichtlich gestaltet und damit verbessert werden kann.

Erhöhung der Wartbarkeit der Systemarchitektur: Durch die Modellierung der Systemarchitektur wird die gesamte IT-Infrastruktur übersichtlich abgebildet. Alle Komponenten und deren Verbindungen werden aufgezeigt, was eine wirtschaftliche Wartbarkeit unterstützt.

Aufgaben von Architekturmodellen

Aus den Zielen lassen sich vier grundlegende Aufgaben von Architekturmodellen ableiten (vgl. Heinrich et al. 2014, 86 ff.):

Flexibilität ermöglichen: Architekturen bilden einen Puffer zwischen den strategischen Unternehmenszielen (inklusive der strategischen IT-Ziele), den Prozessen

und den implementierten Informations- und Kommunikationssystemen. Daher können strategische Ziele ohne eine Veränderung der Informations- und Kommunikationssysteme angepasst werden.

Struktur schaffen: Architekturen schaffen einen Ordnungsrahmen und stellen damit Transparenz auf allen Unternehmensebenen her. Durch die übersichtliche und strukturierte Darstellung von Unternehmenskomponenten, wie z. B. Daten, Funktionen und Prozessen kann vermeidbarer Komplexität, wie z. B. Redundanzen, entgegengewirkt werden.

Integration unterstützen: Architekturen helfen bei der Zusammenführung von Objekten, wie Daten, Applikationen oder Hardwarekomponenten. Es können auch unterschiedliche Teilarchitekturen zu einer übergeordneten Gesamtarchitektur abstrahiert werden. Z. B. enthält die Unternehmensarchitektur in der Regel implizit die Geschäftsprozessarchitektur, die Applikationsarchitektur und die Systemarchitektur. Durch den abstrakteren Überblick kann die übergeordnete Architektur helfen, die Interdependenzen zwischen den untergeordneten Architekturen übersichtlicher darzustellen.

Standardisierung gewährleisten: Durch ihre Struktur und die vorgegebenen Ablaufsteuerungen unterstützen Architekturen die Standardisierung von Prozessen.

2.3.2 Ausgewählte Architekturmodelle

Im Folgenden wird als klassisches Modell das Framework for Enterprise-Architecture von *Zachman* vorgestellt. Darauf aufbauend modifizierte *Krcmar* als ganzheitlichen Ansatz die Informationssystemarchitektur. Ein modernes Architekturmodell stellt die Architekturpyramide von *Dern* dar.

2.3.2.1 Framework for Enterprise-Architecture von *Zachman*

Das von *Zachman* 1987 entwickelte Framework for Enterprise-Architecture stellt eines der ersten Architekturansätze von Informationssystemen dar, die die Idee der strukturellen Sichtweise mit der Visualisierung durch Modellierung verfolgen. Sein Ansatz geht davon aus, dass sich in einem Gesamtsystem nicht nur eine Architektur befindet, sondern immer mehrere Architekturen die Gesamtstruktur beschreiben.

Das Framework soll einen Ordnungsrahmen für Anwender darstellen. Es soll helfen die Komplexität eines Gesamtsystems zu bewältigen, indem die Aufteilung in leichter handhabbare Teilstücke erfolgt. Diese können sogar unabhängig voneinander isoliert werden, damit auftretende Fehler oder Störungen sich nicht auf das restliche Gesamtsystem auswirken.

Ordnungsrahmen für
Anwender

Zachman beschreibt das Modell anhand der Perspektiven **Scope** (Planer), **Enterprise Model** (Owner), **System Model** (Designer), **Technology Model** (Builder), **Components** (Subcontractor) und **Functioning Enterprise**. Diese Perspektiven bilden die eigentliche Betrachtung auf das Gesamtsystem und orientieren sich an den verschiedenen Rollen (Personen) analog zur Architektur bei einem Hausbau.

Perspektiven
und Rollen

Sichten

Den Rollen Planer, Owner, Designer, Builder und Subcontractor werden unterschiedliche Sichten zugeordnet, die den Fokus repräsentieren. Jede **Sicht** ist durch eine **spezielle Fragestellung** charakterisiert (vgl. hier und im Folgenden Krcmar 2015, 102 f.; Masak 2006, S. 96 ff.; Frank 1994, S. 152 ff.):

- **Data:** Was ist im Hinblick auf die Geschäftstätigkeit wichtig für die Organisation? (What?)
- **Function:** Wie funktioniert die Organisation? (How?)
- **Network:** Wo befindet sich die Organisation? (Where?)
- **People:** Wer ist für welche Schritte verantwortlich? (Who?)
- **Time:** Wann müssen die Ereignisse ablaufen? (When?)
- **Motivation:** Wieso werden die verschiedenen Ziele und Strategien so und nicht anders verfolgt? (Why?)

Zachman entwickelte Ende der achtziger Jahre die erste Version seines Frameworks, welches den Fokus nur auf Daten, Funktionen und Netzwerke legte. Die Dimensionen Personen, Zeit und Motivation folgten erst in den neunziger Jahren.

Abbildung 5 verdeutlicht die Matrix mit den unterschiedlichen Rollen und Sichten des *Zachman* Frameworks. In den Zellen finden sich mögliche Lösungen zur Unterstützung der jeweiligen Rolle hinsichtlich der geforderten Sicht auf den Unternehmensbereich.

In den Zellen der Matrix werden pro Perspektive und Sicht beispielhaft Instrumente zur Umsetzung aufgezeigt. Z. B. aus Motivationssicht und der Perspektive des Enterprise Models kann ein Business Plan eingesetzt werden, um die verfolgten Ziele und Strategien darzulegen. Aus Funktionssicht und der Perspektive des System Models gibt die Applikationsarchitektur Auskunft darüber, wie die Organisation funktioniert. Aus Datensicht und der Perspektive des Technology Models unterstützt ein physisches Datenmodell dabei, die relevanten Daten im Hinblick auf die Geschäftstätigkeit der Organisation aufzuzeigen.

Kritik

Das *Zachman* Framework bietet einen Ordnungsrahmen, der möglichst vollständig die unterschiedlichen Sichten auf das Gesamtunternehmen abbilden soll. Hierbei eignet es sich insbesondere bei der Systementwicklung. Eine Methodenvwahl wird dabei nicht vorgegeben. Es unterstützt jedoch nur bedingt die Beschreibung der Ausrichtung zwischen IT und Business, da die Kopplung zwischen der Strategieebene und den operativen Zielen nicht eindeutig gelöst ist (vgl. Masak 2006, S. 98). Zudem erschwert die Aufteilung in Matricelemente die ganzheitliche Sicht der Informationssystem-Architektur, da die Interdependenzen zwischen den Matricelementen keine Berücksichtigung finden. Aus diesem Grund entwickelte *Krcmar* ein ganzheitliches Modell der Informationssystem-Architektur (vgl. Krcmar 1990, S. 399).

	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why
Scope (contextual) ► Planner	List of things important for the business	List of processes the business performs	List of locations in which the business operates	List of organizations important to the business	List of events/cycles significant to the business	Lists of business goals/strategies
Enterprise Model (conceptual) ► Owner	e.g. Semantic Model	e.g. Business Prozess Model	e.g. Business Logistics System	e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan
System Model (logical) ► Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model
Technology Model (physical) ► Builder	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Components (out-of-context) ► Subcontractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Specification
Function Enterprise	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY

Abbildung 5: Zachman Framework for Enterprise Architecture

Quelle: In Anlehnung an Zachmann 1987; Sowa und Zachman 1992

2.3.2.2 Ganzheitliches Modell der Informationssystem-Architektur von Krcmar

Nach *Krcmar* (1990, S. 399 ff.) können Unternehmensziele nur durch eine Informationssystem-Architektur unterstützt werden, bei der alle Schichten aufeinander und alle Sichten untereinander abgestimmt sind. Aus dieser Erkenntnis heraus entwickelte er sein ganzheitliches Modell der Informationssystem-Architektur (ISA) in Kreiselform. Die Kreiselform wurde von ihm deswegen gewählt, da nach seiner Aussage das Gesamtsystem durcheinander gerät, wenn nur eine Komponente der ISA entfernt wird. Das heißt, dass in seinem Modell das Gleichgewicht aller Komponenten immer gewährleistet sein soll. Abbildung 6 veranschaulicht das Modell mit seinen Komponenten. Wie bereits in Abschnitt 2.1 aufgezeigt, unterscheidet *Krcmar* in seiner ISA verschiedene Teil-Architekturen, die nachfolgend erläutert werden.

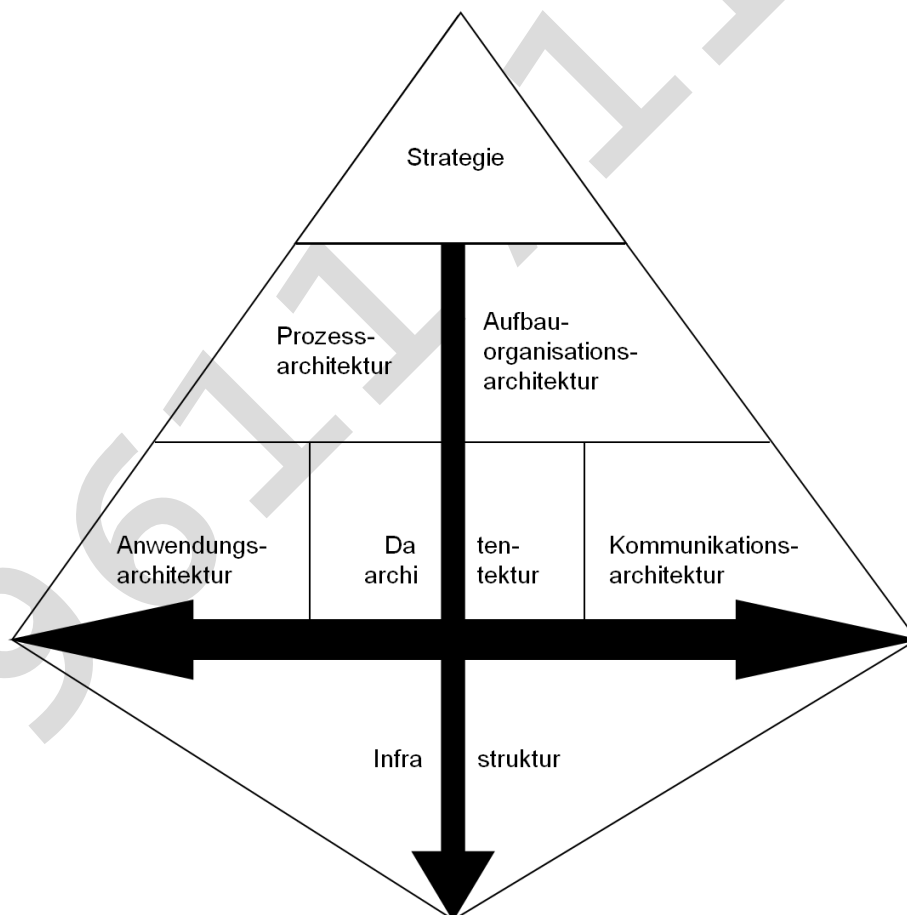


Abbildung 6: Ganzheitliches Modell der Informationssystem-Architektur

Quelle: Entnommen aus Krcmar 1990, S. 399

Das Besondere an diesem Modell, ergänzend zu früheren Ansichten, ist die explizite Berücksichtigung der Geschäftsziele und der daraus resultierenden organisatorischen Struktur. *Krcmar* (1990, S. 399 ff.) erläutert die einzelnen Modellschichten anhand von Kosten, Nutzen, Flexibilität und Risiko.

Strategie

Die ISA enthält in der obersten Schicht Elemente der **Geschäftsstrategie**. *Krcmar* stützt sich dabei auf den Gedanken des unternehmensweiten Informationsmana-

gements (vgl. Krcmar 1990, S. 28 ff.), wodurch die Berücksichtigung der Strategie begründet ist. Zu beachten ist, dass sich die IT und die Unternehmensstrategie gegenseitig beeinflussen. Das Modell veranschaulicht diesen Zusammenhang anhand des schwarzen Pfeils. Die Strategie zieht sich durch das ganze Unternehmen und somit auch durch alle Sub-Architekturen.

Auf der zweiten Schicht befindet sich die Geschäftsprozessarchitektur, auf der die Strategie umgesetzt wird. Die **organisatorische Schicht** besteht aus der Prozess- und der Aufbauorganisationsarchitektur. Diese beiden Sichtweisen sind eng miteinander verknüpft.

Prozess- und
Aufbauorganisations-
architektur

Die dritte Schicht lehnt sich an das *Zachman* Framework an und enthält die Elemente Anwendungsarchitektur⁵, Datenarchitektur und Kommunikationsarchitektur (vgl. hier und im Folgenden Krcmar 1990, S. 399 ff.).

Anwendungs-, Daten-
und Kommunikati-
onsarchitektur

In der **Anwendungsarchitektur** werden Funktionen beschrieben, die sich aus den Geschäftsprozessen und deren Unterstützung zusammensetzen. Die organisatorische Umgestaltung bei der Umstellung auf eine neue Applikation ist sehr komplex, weshalb im Vorfeld in der Regel eine Wirtschaftlichkeitsrechnung vorangestellt wird. Hierbei werden Kosten und Nutzen der einzelnen Applikationen analysiert. Bei der Gestaltung von Anwendungsarchitekturen steht, wie bereits in Abschnitt 2.2.1 beschrieben, die Beachtung von Flexibilität und Risiko im Vordergrund. Im Rahmen der Flexibilität ist zu berücksichtigen, ob weitere strategische Ausrichtungen unterstützt oder verhindert werden. Ein Risiko entsteht dann, wenn Prozesse weiterverfolgt werden müssen, obwohl deren Ineffizienz bereits identifiziert wurde. Besonders bei verteilten Unternehmen müssen eventuelle Synergieeffekte zwischen den heterogenen Applikationen erkannt werden.

Datenarchitekturen beschreiben Datenstrukturen, die den statischen Zusammenhang der unternehmensrelevanten Daten darstellen. Hieraus können Datenmodelle abgelesen werden, die die Geschäftsentwicklung aufzeigen und in den Applikationen verwirklicht wurden. Aufgrund ihrer Stabilität erlauben Datenarchitekturen einen langfristigen Betrachtungshorizont auf die Geschäftsentwicklung. Wie bei der Wirtschaftlichkeitsbetrachtung der Anwendungsarchitektur können auch bei der Datenarchitektur die Flexibilität und das Risiko der Datenstrukturen bewertet werden.

Kommunikationsarchitekturen bieten die Möglichkeit Informationsflüsse abzubilden und sollten unabhängig von der Daten- und Anwendungsarchitektur Beachtung finden. Eine Wirtschaftlichkeitsbetrachtung ist hier sehr schwierig, da Nutzen und Kosten in der Regel nicht eindeutig identifiziert werden können. Es lassen sich zwar die Gesamtkosten der Infrastruktur-Entwicklung erfassen, die Kosten,

⁵ Die von Krcmar definierte Anwendungsarchitektur entspricht nicht der nach unserem Verständnis definierten Applikationsarchitektur.

die hinsichtlich der Informationsflüsse entstehen, sind jedoch schwer messbar. Im Rahmen der Flexibilität können z. B. Unterlassensrisiken und Handlungschancen analysiert werden, die Auswirkungen auf die Informationsflüsse in der Kommunikationsarchitektur haben.

Die dritte Schicht bildet mit ihrer Anwendungs-, Daten- und Kommunikations-Architektur einen Puffer zwischen der darunter liegenden Infrastruktur und der Prozess- und Aufbauorganisationsarchitektur. Die sich möglicherweise schnell verändernden Strategien und die daraus resultierenden Prozess- und Aufbauorganisationselemente müssen mit der technischen Infrastruktur harmonisieren. Hieraus ergibt sich die geforderte Flexibilität, die nur bei einer Gesamtbetrachtung der ISA realisiert werden kann.

Infrastruktur

In der vierten Schicht ist die technische Infrastruktur verankert. Sie beschreibt, welche IT wo im Unternehmen verwendet werden. Sie stellt somit die **Systemarchitektur** dar. In ihr werden alle Technologien erfasst, die der **Datenverarbeitung und -speicherung** dienen. Besondere Beachtung findet die Infrastruktur, wenn technische Innovationen einwirken, auf die das Unternehmen reagieren muss.

Kritik

Positiv zu bewerten ist die ganzheitliche Sicht der ISA auf das Unternehmen, von der Strategie bis zur technischen Infrastruktur. Die ISA betrachtet die unterschiedlichen Sub-Architekturen auf jeder Ebene und stellt deren Wirkungszusammenhang dar. Ein Kritikpunkt der ISA ist jedoch ihre vereinfachende Darstellung. Die einzelnen Schichten lassen sich in sich selbst beschreiben. Die Zusammenhänge im Gesamtmodell sind allerdings schwieriger zu beantworten, stellen jedoch die wesentliche Herausforderung dar. Zudem muss das ISA-Konzept verständlich dargestellt sein. Dazu ist es ebenso notwendig, sich die Fähigkeit anzueignen, die ISA lesen und interpretieren zu können.

2.3.2.3 Architekturpyramide von Dern

Die Architekturpyramide von *Dern* (2006, S. 15 ff.) baut auf der Meinung auf, dass jedes Unternehmen einen komplexen Organismus darstellt und jeder Bestandteil der Pyramide nur eine Momentaufnahme dieses dynamischen Systems widerspiegelt. Jeder Sicht sind daher Prozesse zugeordnet, die dieser Dynamik entsprechen.

Abbildung 7 zeigt die Architekturpyramide von *Dern*. Die Pyramide legt einen besonderen Fokus auf das Management der vorhandenen Architekturen eines Unternehmens (vgl. auch Abschnitt 4.3).

Der Aufbau der Architekturpyramide von *Dern* ist ähnlich dem Aufbau Krcmars ISA. Jedoch beschreibt *Dern* die einzelnen Ebenen ausführlicher und geht im Einzelnen auf die darin liegenden Architekturen ein. Diese werden im Folgenden ebenso detailliert aufgezeigt.

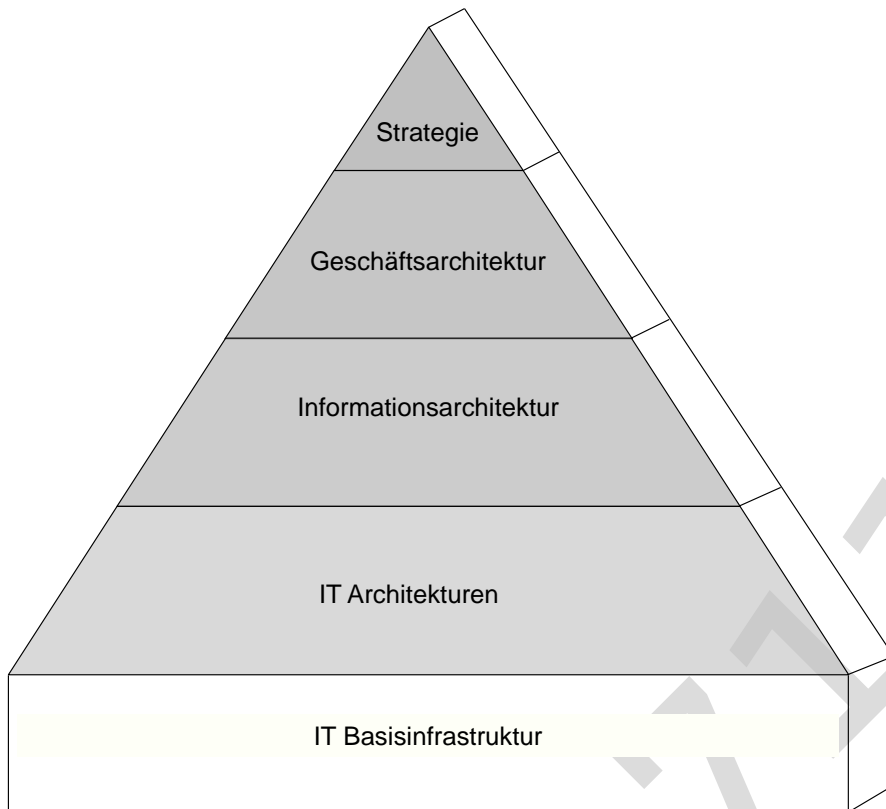


Abbildung 7: Architekturpyramide von Dern

Quelle: Entnommen aus Dern 2006, S. 6

Die obere Ebene der Pyramide bildet die Strategie des Unternehmens ab, in der die **strategischen Ziele** sowie deren **Maßnahmen zur Erreichung** beschrieben werden. Sie bilden die **Geschäftstreiber** für die darunter liegende Geschäftsarchitektur. Durch Geschäftstreiber werden die wesentlichen Geschäftsfelder des Unternehmens aufgedeckt und die daraus resultierende Ausrichtung vorangetrieben.

Strategie

In der Geschäftsarchitektur werden die **Geschäftsfunktionen** und der **Informationsbedarf** formalisiert beschrieben. In ihr ist die Prozess- und die Organisationsarchitektur verankert.

Geschäftsarchitektur

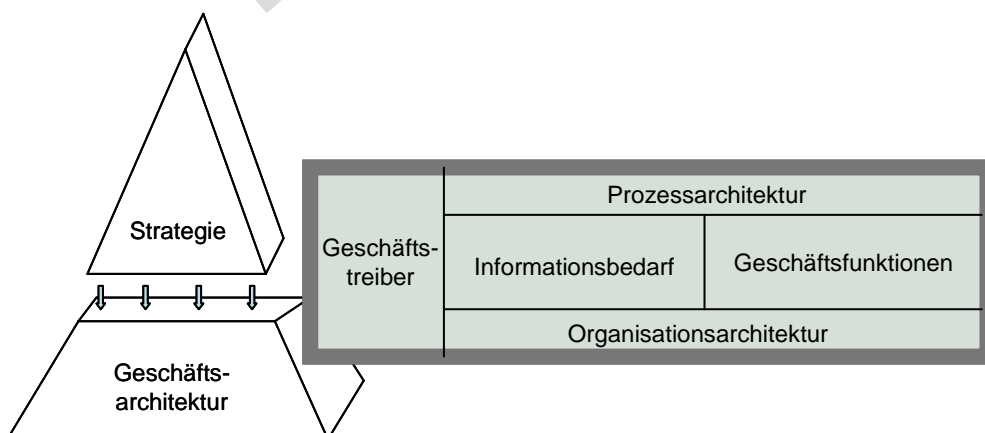


Abbildung 8: Geschäftsarchitektur

Quelle: Entnommen aus Dern 2006, S. 24

Innerhalb der **Prozessarchitektur** werden die Geschäftsprozesse eines Unternehmens definiert und strukturiert. In ihr werden außerdem die internen und externen Kommunikationsbeziehungen (z. B. zum Kunden oder zu Lieferanten) dargestellt sowie die zu deren Durchführung benötigten Geschäftsfunktionen und der Informationsbedarf. Die **Organisationsarchitektur** bildet in diesem Kontext die Aufbauorganisation ab, die so gestaltet werden muss, dass sie die Geschäftsziele optimal unterstützt.

Die Informationsarchitektur beinhaltet **Strukturprinzipien** und **Regeln** für die Gestaltung der **Informationssystem(IS)**⁶-Landschaft. Sie bildet somit das Fundament für die Architekturplanung. Besondere Beachtung findet dabei das IS-Portfolio mit seinen Ist- und Soll-Zuständen.

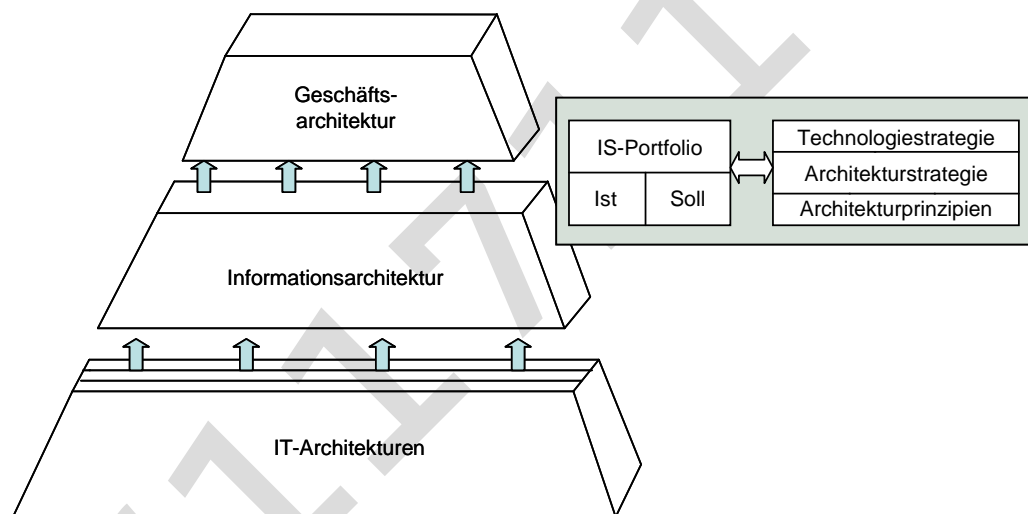


Abbildung 9: Informationsarchitektur

Quelle: Entnommen aus Dern 2006, S. 26

Im **IS-Portfolio** werden die Aufstellung und das Zusammenwirken der IS eines Unternehmens abgebildet (vgl. hier und im Folgenden Dern 2006, S. 71 ff.). Das **Ist-IS-Portfolio** umfasst zum einen die Aufstellung der bestehenden IS, zum anderen die Analyse und Bewertung dieser IS. Des Weiteren werden im Ist-IS-Portfolio der Integrationsgrad der IS und die Darstellung der wichtigsten Informationsflüsse und Schnittstellen beschrieben. Somit stellt das Ist-IS-Portfolio eine strukturierte Gesamtsicht auf alle aktuell vorhandenen IS eines Unternehmens dar. Das **Soll-IS-Portfolio** beinhaltet die geplante Aufstellung der IS unter Berücksichtigung von deren Lebenszyklus. Anhand ausgewählter Kriterien wird die geplante IS-Landschaft strukturiert und eine Integrationsstrategie erarbeitet. Ebenso werden die geplanten wichtigsten Informationsflüsse und Schnittstellen dargestellt.

⁶ Der von Dern verwendete Begriff Informationssystem entspricht dem der in dieser KE eingeführten Definition Applikation. Um der ursprünglichen Definition Dernas zu folgen, wird in diesem Abschnitt die Definition Informationssystem verwendet. Diese Definition weicht von der in diesem Kurs verwendeten Definition Informationssystem (vgl. KE 1) ab.

Weitere Elemente der Informationsarchitektur sind die Technologiestrategie, die Architekturstrategie und die Architekturprinzipien. Die **Technologiestrategie** setzt die Entwicklungsleitlinie der IT-Basisinfrastruktur fest. Die Rahmenbedingungen sind durch die Geschäftstreiber, die vorhandenen IS-Landschaften, das Soll-IS-Portfolio, die bestehende IT-Basisinfrastruktur und die absehbare Entwicklung der IT gegeben. Die Technologiestrategie hat unmittelbaren Einfluss auf die zwei unteren Ebenen der Pyramide, die IT-Architektur und die IT-Basisinfrastruktur. Mit der **Architekturstrategie** werden der Zielzustand und die Maßnahmen beschrieben, die im Rahmen des Architekturmanagements (vgl. Kapitel 4) erreicht werden sollen. **Architekturprinzipien** bilden den Rahmen für Richtlinien und Standards bei der Weiterentwicklung des IS-Portfolios eines Unternehmens. Hier werden alle Anforderungen der Architekturentwicklung und Softwareentwicklung (SE) subsumiert und können ebenso Technologie- und Produktvorgaben umfassen.

Die IT-Architektur beschreibt in abstrakter Weise die Struktur der vorhandenen und geplanten IS. Sie bietet für alle an der IS-Gestaltung Beteiligten eine gemeinsame Kommunikationsplattform. Damit wird der Abstimmungsprozess bei der Planung und Steuerung der IS vereinfacht. Die IT-Architektur ist aufgeteilt in eine statische und eine dynamische Sicht.

IT-Architektur

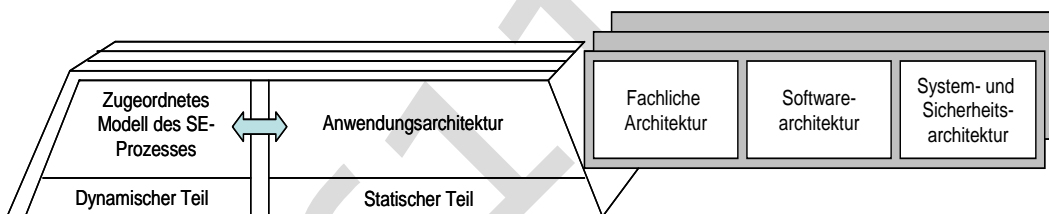


Abbildung 10: IT-Architektur

Quelle: Entnommen aus Dern 2006, S. 20

Im **statischen Teil** wird die Anwendungsarchitektur beschrieben. Sie umfasst die übergeordnete Gesamtstruktur eines oder mehrerer IS. Durch die Definition von logischen Komponenten und ihrer Informationsflüsse wird das Zusammenwirken der wichtigsten Bauteile abgebildet. Die Anwendungsarchitektur teilt *Dern* in die fachliche-, die Software- und die System- und Sicherheitsarchitektur auf (vgl. hier und im Folgenden Dern 2006, S. 20 ff.). Die **fachliche Architektur** strukturiert die betriebswirtschaftliche Funktionalität und die dafür relevanten Daten (Datenarchitektur). Die **Softwarearchitektur** beinhaltet eine softwaretechnische Strukturierung von IS sowie deren Entitäten (wie Schichten, Dienste und Arten von Komponenten). Weiterhin werden die Eigenschaften von IS sowie deren Interaktionen klassifiziert. Die Softwarearchitektur bietet eine Organisations- sowie Steuerungsstruktur für alle in ihr abgebildeten Softwarekomponenten. Dabei werden sowohl die Schnittstellen definiert, als auch die Kommunikation, Synchronisation und die Datenzugriffe aufgezeigt. Innerhalb der **System- und Sicherheitsarchitektur** werden die technischen Komponenten, wie z. B. Hardware und deren Sicherheitsanforderungen, aufgezeigt, die im Rahmen der SE beachtet werden müssen, um diese auf der unten beschriebenen IT-Basisinfrastruktur umzusetzen.

Im **dynamischen Teil** wird das Modell eines SE-Prozesses dargestellt. Hierbei werden die benötigten Aktivitäten und Artefakte festgelegt, um ein Informationssystem zu beschreiben, zu implementieren und freizugeben. In diesem Rahmen müssen die funktionalen und nicht-funktionalen Anforderungen beachtet werden. Im Modell des SE-Prozesses wird die SE-Umgebung abgebildet. Diese kann je nach Applikationstyp unterschiedlich sein.

IT-Basisinfrastruktur

Die unterste Schicht bildet die IT-Basisinfrastruktur. Hier werden alle Hardwarekomponenten abgebildet. In der IT-Basisinfrastruktur ist die Plattform- sowie die Sicherheitsstrategie verankert. Mit der Plattformstrategie werden verschiedene Basisplattformen für die aus der IS-Architektur abgeleitete Technologiestrategie festgelegt. In der Technologiestrategie wird, wie bereits erläutert, die Weiterentwicklung der IT-Basisinfrastruktur formuliert. Die Sicherheitsstrategie legt die Weiterentwicklung der Sicherheitsarchitektur fest.

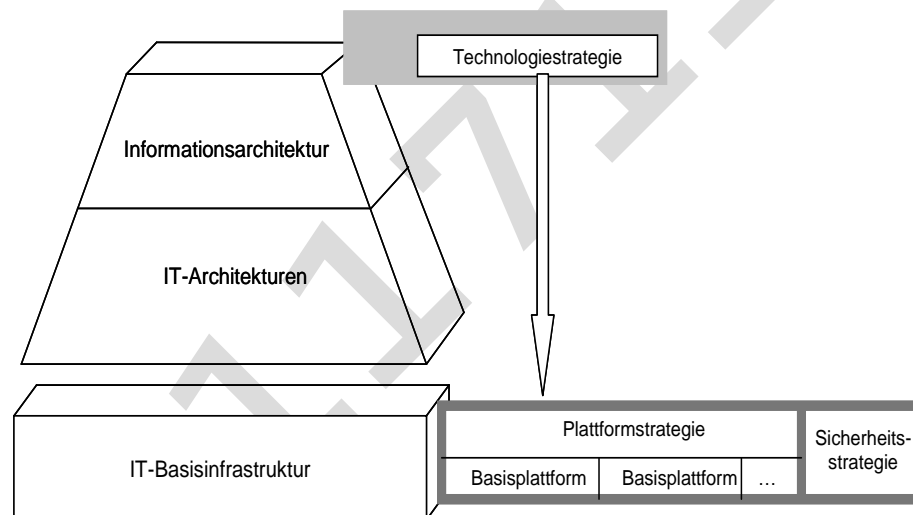


Abbildung 11: IT-Basisinfrastruktur

Quelle: Entnommen aus Dern 2006, S. 29

Kritik

Die Architekturpyramide von *Dern* berücksichtigt die gesamte Unternehmensarchitektur einschließlich der Strategie und allen Sub-Architekturen. Hierbei berücksichtigt er besonders das IS-Portfoliomanagement. Bei jeder Architekturebene wird insbesondere das Architekturmanagement hervorgehoben (vgl. auch Abschnitt 4.3.2), wobei der Fokus hierbei auf der Planung und Entwicklung der Architektur liegt. Die Dokumentation und Steuerung der Architektur findet in *Derns* Architekturpyramide wenig Beachtung.

In diesem Kapitel wurde ein allgemeines Architekturverständnis gelegt. Hierbei wurde deutlich, dass Architekturen die Basis für mehr Flexibilität, Struktur, Standardisierung und Integration bilden. Die Integration stellt einen weiteren komplexen Bereich in der Wirtschaftsinformatik dar und wird im folgenden Kapitel 3 näher erläutert. Dabei kann sich Integration auf z. B. Daten, Funktionen und Prozesse beziehen, die innerbetrieblich, als auch zwischenbetrieblich zusammengeführt werden müssen, um einen optimalen Redundanzgrad und Konsistenz zu gewährleisten.

2.4 Übungsaufgaben

1. Was ist unter einer Unternehmensarchitektur zu verstehen?
2. Was ist unter einer Applikationsarchitektur zu verstehen?
3. Welche Ziele werden mit einer Applikationsarchitektur verfolgt?
4. Nennen Sie fünf Ziele, die mit Architekturmodellen verfolgt werden.
5. Von welchem Ansatz geht Zachman bei seinem Framework for Enterprise-Architecture aus? Nennen Sie die von Zachman verwendeten Perspektiven (Rollen) und Sichten.
6. Beschreiben Sie die Informationssystem-Architektur (ISA) nach Krcmar.
7. Nennen Sie die Ebenen der Architekturpyramide nach Dern und erläutern Sie die Informationsarchitektur näher.

3 Integration

Zusammenführen und
Koordination

Unternehmen sehen sich im Wettbewerb immer wieder Integrationsprojekten gegenübergestellt, z. B. im Rahmen von **Merger & Acquisition** oder **Kooperationen**. Hierbei beschreibt die Integration das Zusammenführen bzw. die Koordination von z. B. Daten, Funktionen und Prozessen. Die Integration kann aus verschiedenen Blickwinkeln, wie des Gegenstandes, der Reichweite, der Richtung oder des Automationsgrads betrachtet werden. Im Folgenden wird die Terminologie aus der Wirtschaftsinformatik zum Thema Integration aufgezeigt und ein Überblick über die unterschiedlichen Perspektiven gegeben. Die Perspektive des Gegenstandes wird aufgrund der hohen Komplexität besonders hervorgehoben. Aus den Perspektiven lassen sich Ziele und Probleme der Integration ableiten, die im Einzelnen erläutert werden. Abschließend werden zwei in der Literatur aktuell diskutierten Integrationslösungen vorgestellt.

3.1 Begriff und Perspektiven der Integration

Die Wirtschaftsinformatik betrachtet die Integration einerseits aus **technischer Sicht**, wie z. B. IT-Infrastruktur und beachtet andererseits ebenso **betriebswirtschaftliche Aspekte**, wie z. B. Prozesse und Daten.

Dabei wird unter Integration ganz allgemein „die (Wieder-)Herstellung eines Ganzen oder einer Einheit“ verstanden (vgl. Kaib 2002, S. 10), unabhängig vom technischen oder betriebswirtschaftlichen Aspekt. Die Integration kann entweder durch Verbinden oder durch Vereinigen von Objekten erreicht werden (vgl. Rosemann 1999, S. 7). *Picot et al.* (2003, S. 180) definieren die Integration als eine Zusammenführung von bestimmten funktionalen Eigenschaften. Die für diesen Lehrbrief gewählte Definition schließt sich diesen Definitionen an.

Definition:
Integration

Integration bezeichnet das Herstellen oder Wiederherstellen von Objekten oder funktionalen Eigenschaften durch Verbindung oder Zusammenführung.

Mertens (2009, S. 1 ff.) betrachtet ebenso die Wiederherstellung und vermerkt, dass zu integrierende Elemente Menschen, Technik und Aufgaben sind. Er hat zudem den Begriff der „Integrierten Informationsverarbeitung“ geprägt und betrachtet in diesem Rahmen unterschiedliche **Integrationsperspektiven**. In der nach *Jung* (2006, S. 27) dargestellten Abbildung 12 werden die verschiedenen Integrationsperspektiven der „Integrierten Informationsverarbeitung“ übersichtlich aufgezeigt.

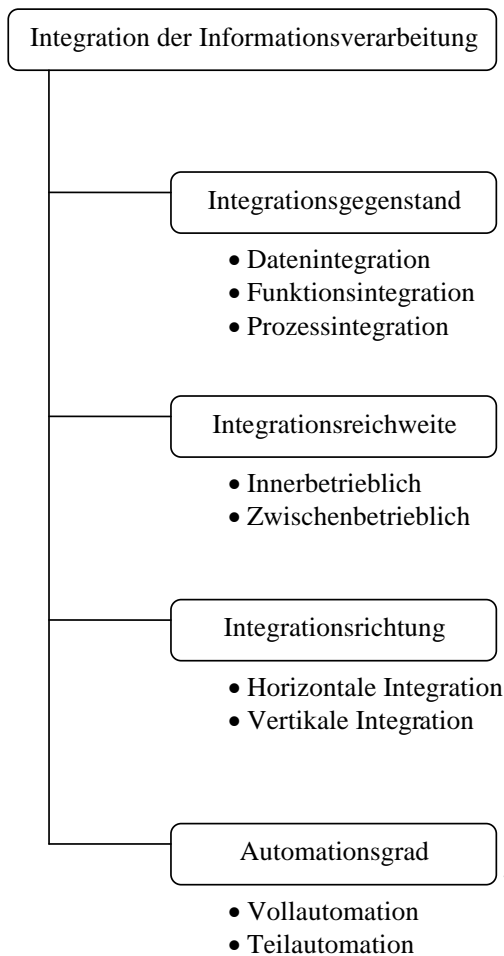


Abbildung 12: Perspektiven der integrierten Informationsverarbeitung

Quelle: Entnommen aus Jung 2006, S. 27

Die Integrationsperspektiven beschreiben dabei die grundsätzliche Sichtweise (z. B. technische oder betriebswirtschaftliche), nach der eine Integration stattfinden kann. Hierbei wird nach dem Automationsgrad, der Richtung, der Reichweite und dem Gegenstand der Integration unterschieden (vgl. hier und im Folgenden Jung 2006, S. 26 ff.; Mertens 2009, S. 1 ff.).

3.1.1 Automationsgrad

Der Automationsgrad beschreibt, inwieweit die Integrationsprozesse selbstständig ablaufen. Man spricht von **Teilautomation**, wenn der Anstoß zur Integration noch den Einfluss menschlicher Arbeit, wie z. B. das Starten einer Applikation, benötigt. **Vollautomation** bedeutet, dass die Applikationen direkt miteinander kommunizieren und Aktionen vollautomatisiert ablaufen, ohne externe Eingriffe von Menschen.

3.1.2 Integrationsrichtung

Mit der Integrationsrichtung wird die Ausrichtung definiert, in die ein Integrationsprozess ausgeführt werden kann. Das kann sowohl in horizontaler, als auch in vertikaler Richtung stattfinden (vgl. Abbildung 13). Bei der **horizontalen In-**

Integration erfolgt die Zusammenführung von z. B. Daten entlang des Wertschöpfungsprozesses. Das kann entweder innerbetrieblich, entlang eines Auftragsdurchlaufs, oder auch zwischenbetrieblich, z. B. durch Integration von Funktionen unternehmensübergreifender Geschäftsprozesse geschehen. Hierbei sollten sich die Unternehmen auf gleicher Wertschöpfungsstufe befinden, wie z. B. Zulieferer und Produzent. Bei der **vertikalen Integration** wird innerbetrieblich die Integration zwischen Planungs- und Kontrollsystemen und den Administrations- und Dispositionssystemen verstanden. Zwischenbetrieblich handelt es sich um Unternehmen unterschiedlicher Wertschöpfungsstufen. Ein Beispiel für die zwischenbetriebliche vertikale Integration stellt die Weitergabe verdichteter Kennzahlen von der Unternehmenstochter an das Mutterunternehmen dar.

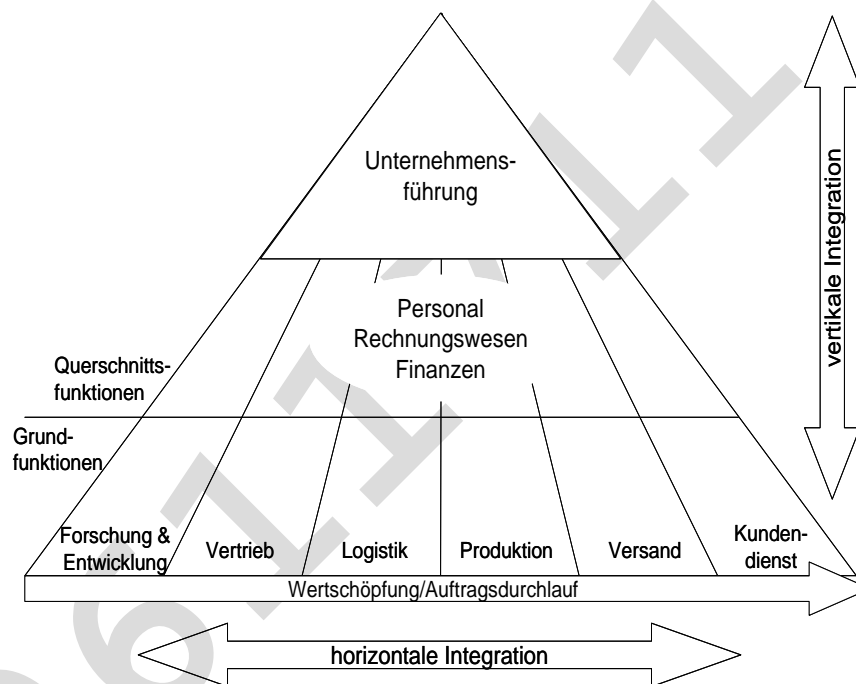


Abbildung 13: Integrationsrichtung

Quelle: Leicht verändert, entnommen aus Mertens 2009, S. 6

3.1.3 Integrationsreichweite

Die Integrationsreichweite beschreibt die „organisatorische Entfernung“ zwischen den von der Integration betroffenen Organisationseinheiten. Hierbei wird in innerbetriebliche und zwischenbetriebliche Reichweite unterschieden. Die **innerbetriebliche** Integration bezieht sich auf die Integration innerhalb einer Organisation, die sowohl bereichsintern innerhalb einer Abteilung, als auch bereichsübergreifend über Abteilungsgrenzen hinweg stattfinden kann. Z. B. führt die Implementierung eines ERP-Systems zu einer innerbetrieblichen Integration von Daten und Funktionen, die von mehreren Arbeitsplätzen abteilungsintern sowie abteilungsübergreifend genutzt werden können. Die **zwischenbetriebliche** Integration beschreibt das Zusammenführen oder Austauschen von Daten und Prozessen über Unternehmensgrenzen hinweg, wie es z. B. in Unternehmensnetzwerken erforderlich sein kann. Diese Integration entsteht z. B. bei der Nutzung von SCM-Systemen, auf die in der Regel unterschiedliche Unternehmen zugreifen.

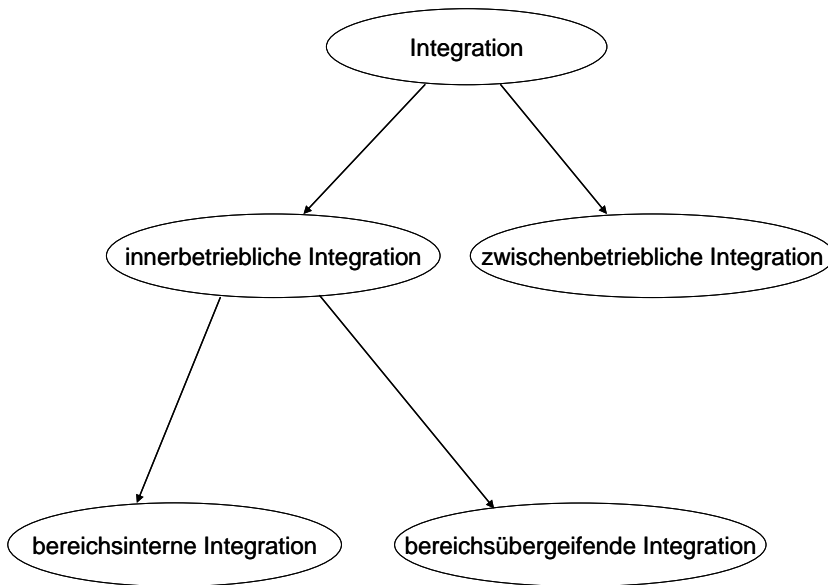


Abbildung 14: Integrationsreichweite

3.1.4 Integrationsgegenstand

Der Integrationsgegenstand beschreibt das Objekt, nach dem die Integrationsüberlegungen stattfinden. Hierbei kann nach *Mertens* zwischen Daten, Funktionen, Prozessen, Methoden⁷ und Programmen unterschieden werden. Im Hinblick auf die in dieser KE hauptsächlich betrachteten Applikationsarchitekturen, werden im Folgenden die Daten-, die Funktions- und die Prozessintegration näher betrachtet. Diese Integrationsgegenstände decken indirekt die Methoden- und Programmintegration mit ab. Bei der **Methodenintegration** werden im Unternehmen verwendete Methoden aufeinander abgestimmt bzw. zusammengeführt, z. B. um Kosten im Bereich der Bedarfsplanung bei Supply-Chains zu sparen. Dies wird im weitesten Sinne auch durch die Überlegungen der Funktionsintegration mit abgedeckt, da die Methoden zur Realisierung von betrieblichen Funktionen eingesetzt werden (vgl. Jung 2006, S. 35). Die **Programmintegration** bezieht sich auf technische Aspekte. Hierbei werden einzelne Programmbausteine aufeinander abgestimmt. Dies dient hauptsächlich zur Integration von Benutzerschnittstellen, der Medienintegration und der Hardwareintegration und soll in diesem Rahmen nicht ausführlich betrachtet werden (zur Vertiefung vgl. Mertens 2009, S. 2 ff.).

3.1.4.1 Datenintegration

Der zentrale Integrationsgegenstand der Wirtschaftsinformatik sind Daten. Bei der Datenintegration werden Daten aus den unterschiedlichen Applikationen zusammengeführt. Dies kann sowohl auf physischer als auch auf logischer Ebene betrachtet werden.

⁷ Eine Methode beschreibt eine geplante Vorgehensweise, mit der systematisch die Erreichung eines bestimmten Arbeitsergebnisses erzielt werden kann (vgl. Wilde und Hess 2006, S. 2 ff.)

Bei der **physischen Datenintegration** steht die Zusammenführung von Daten aus heterogen gewachsenen Applikationslandschaften im Vordergrund, ungeachtet ihrer funktionalen Verwendung. Als Beispiel sei hier der Aufbau eines DW genannt, welches die Daten vollständig, konsistent und zeitnah bereitstellt. Hier werden ETL (Extraktion, Transformation, Laden)-Werkzeuge benötigt, die den Vorgang der Extraktion, des Zusammenführens und Bereitstellens der Daten unterstützen. Dabei entsteht die Integration durch Vereinigung der Daten, indem eine umfassende Datenbasis geschaffen wird, auf die alle Funktionseinheiten (z. B. Auftragsbearbeitung, Logistik, Finanzbuchhaltung) einen gemeinsamen Zugriff haben. Die Speicherung im DW muss redundanzfrei erfolgen und die Daten sollen zeitnah für die entsprechenden dispositiven Bereiche zur Verfügung stehen.

Bei der **logischen Datenintegration** werden Daten, die in einem betriebswirtschaftlichen Kontext zueinander stehen, über Schnittstellen zwischen Applikationen ausgetauscht, ohne in einer gemeinsamen Datenbasis gehalten zu werden. Im Rahmen dieses Vorgangs bleiben die einzelnen Daten unverändert und können Redundanzen aufweisen. Diese Redundanzen werden bei der Verbindung jedoch dokumentiert und können somit auf Widersprüchlichkeiten überprüft werden. Dieser Vorgang findet sich häufig bei der automatisierten Datenübertragung (vgl. Heine 1999, S. 65; Jung 2006, S. 38; Mertens 2009, S. 1; Rosemann 1999, S. 7).

3.1.4.2 Funktionsintegration

Durch die Funktionsintegration sollen Aufgaben und Lösungsverfahren miteinander verknüpft werden. Dies geschieht anhand von Kommunikationskanälen zwischen Applikationen. Hierbei wird zwischen aufgabenträgerorientierter⁸ und datenflussorientierter Funktionsintegration unterschieden.

Die **aufgabenträgerorientierte Funktionsintegration** bezieht sich auf die Vereinigung von teilautomatisierten Aufgaben, um die Bearbeitung dieses Aufgabenblocks von nur einer Person zu gewährleisten. Hierbei soll die Kommunikation über einen Kommunikationskanal zwischen Mensch und Applikation geschehen. Dadurch werden personelle und rechnergestützte Aufgabenteile miteinander verbunden. Ebenso können mehrere teilautomatisierte Aufgaben durch eine Person vereinigt werden, sofern sie dem gleichen Aufgabenträger zugeordnet sind. Die so entstandenen Aufgabenblöcke können an einem Arbeitsplatz bearbeitet werden. Zu beachten ist hierbei allerdings die Qualifikation der bedienenden Person. Die Zusammenfassung von Aufgaben unterschiedlicher Anforderungsklassen kann ineffizient sein. Z. B. sollten Aufgaben, wie die Stammdatenerfassung, nicht an einem Arbeitsplatz erfolgen, der die Qualifikation eines Controllers benötigt, da dies unwirtschaftlich wäre.

⁸ Der Begriff Aufgabenträger wurde von Ferstl und Sinz (2013, S. 98) definiert. Hierbei wird zwischen maschinellen (Rechner) und personellen (Menschen) Aufgabenträgern unterschieden. Ein Aufgabenträger bearbeitet eine Aufgabe selbstständig, z. B. Transportsysteme im Fertigungsbereich, die vollautomatisiert Aufgaben verrichten.

Die **datenflussorientierte Funktionsintegration** betrachtet die datenflussorientierten Funktionsstrukturen in einer Applikation. Die Applikation besteht aus einem Netz solcher Funktionen, sowie Datenspeichern und Ein- und Ausgabekanälen. Hierüber werden Dateien und Nachrichten gesendet. Bei der datenflussorientierten Funktionsintegration wird über eine bestimmte Kommunikationsstruktur versucht, die benötigten Kommunikationskanäle zu automatisieren und Automationsinseln zu verbinden. Als Beispiel kann hier die Synchronisation von Adressbüchern oder Terminkalendern eines Desktop-Computers mit einem mobilen Endgerät genannt werden. Teilansätze können auch im Bereich der Vorgangssteuerung über Nachrichtenstandards realisiert werden. Hierfür dienen z. B. XML-basierte Standards oder EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport) (vgl. Ferstl und Sinz 2013, S. 245 f.; Jung 2006, S. 38; Rosemann 1999, S. 7).

3.1.4.3 Prozessintegration

Im Rahmen der Prozessintegration werden einzelne Prozesse und Vorgänge aufeinander abgestimmt und zusammengeführt. Die einzelnen Prozesse können entweder zu einer Prozesskette verbunden werden, um den Gesamtablauf eines Vorgangs wirtschaftlicher zu gestalten (z. B. Warenausgang und Rechnungsstellung) oder ähnlich gelagerte Prozesse können vereinigt werden, um dadurch eine höhere Kosteneffizienz zu erzielen (z. B. gleichzeitiger Ausdruck von Rechnung und Lieferschein und Versendung der Rechnung im Warenpaket) (vgl. Mertens 2009, S. 1, 24 f.; Rosemann 1999, S. 8).

Bei der Prozessintegration werden die Prozessabläufe sowohl innerbetrieblich, als auch zwischenbetrieblich aktiv gesteuert. Dazu gehört auch die Koordination aller beteiligten Benutzer an dem Prozess. Dabei kann die Prozessintegration einerseits als Zustand und andererseits als Vorgang definiert werden (vgl. hier und im Folgenden Vogler 2006, S. 127 ff.).

Die **Prozessintegration als Zustand** definiert sich hauptsächlich aus der Steuerung von Prozessabläufen. Einerseits können verschiedene Arten der Prozessablauffolge differenziert werden, andererseits können unterschiedliche Varianten von Aufgabenträgern für die Prozessablaufsteuerung definiert werden. Somit wird die Prozessintegration nach unterschiedlichen Zuständen klassifiziert.

Die Art der AbLauffolge kann standardisiert, flexibel oder ad hoc erfolgen. Die **standardisierte** Form eignet sich besonders bei Routine- oder strukturierten Prozessen, für die Standardabläufe und -steuerungsprozesse vordefiniert werden. Hier wird ein immer gleicher Ablauf vorgegeben. Bei der **flexiblen** Form ist lediglich die Gesamtabwicklung aller Aktivitäten eines Prozesses ausschlaggebend. Die AbLauffolge kann flexibel gewählt werden. Hier ist auch eine teilweise standardisierte Vorgabe denkbar. Die flexible Steuerung gestaltet sich hauptsächlich in der Überwachung der vollständigen Bearbeitung eines Prozesses, nicht jedoch in der Reihenfolge des Prozessablaufes. Dabei werden lediglich die Folgeaktivitäten eines Prozesses aufgezeigt und keine standardisierten Vorgaben gemacht. Bei unbekannten AbLauffolgen von Prozessen kann weder die AbLauffolge noch deren

Art der
AbLauffolge

Art der Ablaufsteuerung

Steuerung spezifiziert werden. Hier agiert der Verantwortliche **ad hoc** auf den ihm unbekannten Prozess. Er erhält dabei von der Applikation nur Unterstützungsfunktionen in Form von z. B. Auswahllisten oder auch einer Protokollierung vorheriger Prozessabläufe. Die Protokollierung kann sinnvoll sein, wenn Prozesse im Falle eines Systemfehlers zurückgespielt oder wiederholt werden müssen.

Weiterhin kann im Hinblick auf den Aufgabenträger zwischen manueller, teilautomatisierter oder vollautomatisierter Ablaufsteuerung unterschieden werden. Hierbei wird die nötige Interaktion zwischen Mensch und Computer definiert. Bei der **manuellen** Ablaufsteuerung ist der Aufgabenträger der Mensch, der alle Aktivitäten des Prozesses manuell anstoßen muss. Bei der **teilautomatisierten** Steuerung wird die jeweilige automatisierte Abfolge vom Menschen ausgelöst. Der Computer unterstützt hier z. B. durch das Aufzeigen aller nachfolgenden Prozessaktivitäten. Die **vollautomatisierte** Steuerung bedarf keiner menschlichen Aktivität mehr. Hier wird der Prozessablauf allein durch den Computer durchgeführt.

Nachdem die Prozesse eindeutig klassifiziert wurden, muss das weitere Vorgehen bei der Prozessintegration geplant werden. Somit wird bei der **Prozessintegration als Vorgang** die Vorgehensweise betrachtet, wie die verschiedenen Prozessaktivitäten zusammengeführt werden können. Hierbei wird zwischen der Wahl einer geeigneten Steuerungsvariante, der Zusammenstellung von Aufgabenblöcken sowie der Realisierungsplanung unterschieden.

Die einzelnen **Abläufe eines Prozesses** müssen in einem ersten Schritt analysiert und voneinander abgegrenzt werden. Hierdurch wird die Komplexität des Prozesses verringert und das weitere Vorgehen kann detaillierter geplant werden. Für jeden identifizierten Prozessablauf muss dann eine geeignete Steuerungsvariante definiert werden. Im Weiteren werden die Aufgabenblöcke nach zuvor beschriebenen Zuständen klassifiziert, das heißt, die genaue **Ablauffolge des Aufgabenblockes** und die dafür verantwortlichen Aufgabenträger werden abgegrenzt. Hierbei sind Aspekte, wie die Ablaufsteuerung und die Benutzerergonomie sowie die notwendige Technologie zur Realisierung von Bedeutung. Dabei ist zu beachten, dass die Zusammenstellung der Aufgabenblöcke aus einer benutzerorientierten Sichtweise erfolgen soll und damit in der Regel von der Sichtweise des ursprünglichen Prozessentwurfes abweicht. Abschließend müssen noch, im Rahmen der Realisierungsplanung, die **Verantwortlichkeiten** für die abgegrenzten Aufgabenblöcke definiert werden. Hierbei werden insbesondere die Abläufe der Aufgabenblöcke identifiziert.

3.2 Ziele und Probleme der Integration

Ziele der Integration

Aus den zuvor vorgestellten Integrationsperspektiven können generelle Ziele der Integration abgeleitet werden. Hierzu gehören (vgl. Mertens 2009, 10 f.):

Reduktion von künstlichen Grenzen zwischen Organisationseinheiten, Funktionsbereichen und Prozessen: Durch die Integration werden negative Auswirkungen, die innerbetrieblich als auch zwischenbetrieblich durch die vorhandene

Aufgabenteilung entstehen, gemildert. Hierbei muss eine Verbindung oder Vereinigung der einzelnen Informationssystemkomponenten erfolgen, um deren Kooperationsfähigkeit zu erhöhen. Dies geschieht z. B. mittels eines ERP-Systems, welches den Akteuren ermöglicht, auf eine gemeinsame Datenbasis zuzugreifen (vgl. Frank 1994, S. 22). Ebenso verändert z. B. die Verringerung eines Einkaufspreises im Artikelstamm die Höhe der Lagerbewertung und reduziert gleichzeitig den Bilanzposten in der Bilanzbuchhaltung.

Reduktion des personellen Eingabeaufwandes: Durch die Integration von Daten aus unterschiedlichen Applikationen kann der Eingabeaufwand der Benutzer reduziert werden. Z. B. kann ein ERP-System die Fakturierungsdaten an ein EIS übertragen und benötigte Vertriebsstatistiken können sofort ausgeführt werden, ohne dass die Daten noch einmal separat in eine weitere Applikation eingegeben werden müssen. *Frank* (1994, S. 27) spricht hierbei von **Verfügbarkeit**. Er meint damit die Zugriffsmöglichkeit einer Applikation oder einer Person, auf die Daten und Funktionen einer anderen Applikation über Schnittstellen.

Reduktion von Erfassungsfehlern: Die Reduktion des personellen Eingabeaufwandes trägt auch dazu bei, dass Eingabefehler vermindert werden, da keine Medienbrüche entstehen. Medienbrüche entstehen z. B. dann, wenn erst eine Liste aus einer Applikation auf Papier ausgedruckt werden muss, um die Daten dann wieder durch manuelle Eingabe in einer zweiten Applikation zu digitalisieren.

Realisierung betriebswirtschaftlicher Konzepte: Durch die Reduzierung des Eingabeaufwandes wird ebenso eine Basis für die Einbettung von Vorhersage-, Planungs- und Optimierungsmodellen geschaffen. Daraus können etablierte betriebswirtschaftliche Konzepte, wie z. B. SCM oder CRM realisiert werden. Hierbei kann erst durch die Nutzung von integrierten Lösungen das volle Potenzial dieser Konzepte ausgeschöpft werden (vgl. Jung 2006, S. 72 ff.).

Steigerung der Qualität von betrieblichen Prozessen: Durch die Prozessintegration können Prozessketten vorgegeben werden, die in der Regel teil- oder vollautomatisiert ablaufen. Hierbei hilft die standardisierte Vorgehensweise, alle Aufgabenblöcke auszuführen und dabei nichts zu vergessen. Außerdem wird durch die Standardisierung eine wirtschaftliche Bearbeitung der Prozesse gewährleistet.

Verringerung von Redundanzen: Eine wichtige Zielsetzung der Integration ist die Verringerung von Redundanzen. Durch die Eliminierung doppelt vorhandener Daten, Funktionen oder Prozesse soll die Komplexität in einem Unternehmen verringert werden, um dadurch eine höhere Effizienz zu generieren. Mit einem geringen Redundanzgrad wird zudem eine hohe Systemintegrität verbunden und dadurch eine Konsistenz im Zeitverlauf. Dabei ist jedoch zu beachten, dass ein gewisser Redundanzgrad auch eine Absicherung im Falle einer Systemstörung bedeutet. Demzufolge ist weder eine zu hohe, noch eine zu niedrige Redundanz vorteilhaft. Daher ist die Zielsetzung die Erreichung einer **optimalen** Redundanz (vgl. Ferstl und Sinz 2013, S. 241 f.; Frank 1994, S. 27 f.).

Probleme der Integration

Aufdecken von Fehlern durch Mehrfachnutzung: Durch die Integration von Daten, Funktionen und Prozessen erfolgt deren Mehrfachnutzung in unterschiedlichen Organisationseinheiten. Dadurch ist die Wahrscheinlichkeit größer, dass Fehler in diesen Objekten schneller oder überhaupt erst aufgedeckt werden. Z. B. greifen auf den Artikelstamm die Abteilungen Auftragsbearbeitung, Bestellwesen und Kommissionierung zu. Liegt ein Fehler in einer Artikelbezeichnung vor, wird dieser eventuell schneller entdeckt, als bei getrennt vorhandenen Artikelstämmen.

Aufgrund der hohen Komplexität der Integration stehen den aufgezeigten Zielen generelle Probleme bei der Realisierung gegenüber. Zu diesen gehören (vgl. Mertens 2009, S. 11 f.; Vogler 2006, S. 24 ff.):

Lange Realisierungslaufzeiten: Wie oben bereits erläutert, sind die Überlegungen im Rahmen der Integration vielschichtig und umfangreich. Daher erreichen Integrationsprojekte einen hohen, oft nicht voraussehbaren Komplexitätsgrad und damit in der Regel sehr lange Realisierungslaufzeiten. Häufig sind nicht alle möglichen Lösungsvarianten bekannt oder das Ausmaß der Integrationsentscheidung kann nicht überblickt werden. Dadurch ergeben sich unvorhergesehene Konsequenzen durch Interdependenzen, die z. B. zwischen Prozessabläufen und Applikationen entstehen und zusätzlichen Anpassungen an den Applikationen oder Schnittstellen bedürfen. Generell sollten diese Interdependenzen dokumentiert werden. Damit kann ein Überblick erfolgen, der z. B. zukünftige, nicht benötigte Redundanzen zu vermeiden hilft. Dies wird häufig unterlassen, da die Integrationsprojekte in der Regel einem hohen Zeitdruck unterliegen. Die fehlenden Dokumentationen führen allerdings häufig zu noch längeren Projektlaufzeiten, da durch die fehlende Transparenz keine Kontrollmöglichkeiten im Rahmen der Integrationsbeziehungen vorhanden sind. Zudem erschweren fehlendes methodisches Vorgehen sowie fehlende Standardisierungsmöglichkeiten die Realisierung von Integrationsprojekten.

Lange Amortisationsdauer: Es ist nachvollziehbar, dass sich bei Integrationsprojekten hohe Kosten der Realisierung ergeben. Diese können z. B. bei der Erstellung der Infrastruktur, durch Planungs- und Entwicklungskosten des Projektes, durch spätere Inbetriebnahme und Wartung sowie auch evtl. nachträglich anfallender Desintegrationskosten (z. B. bei Unternehmensteilungen) entstehen (vgl. Jung 2006, S. 71). Der gewünschte Nutzen der Integration entsteht oft in viel späteren Phasen. Häufig entstehen im Laufe der Zeit auch noch Insellösungen aufgrund von Projekten, die unter hohem Zeitdruck durchgeführt worden sind. Diese verzögern zusätzlich die Nutzenentstehung. Integrationsprojekte weisen aus diesen Gründen in der Regel lange Amortisationszeiten auf.

Schlechte Integrationsfähigkeit: Ein weiteres Problem stellt die häufig schlechte Integrationsfähigkeit von Daten, Funktionen oder Prozessen in die vorhandene Applikationsarchitektur dar. Das liegt in der Regel an einer im Zeitablauf stark heterogen gewachsenen Applikationslandschaft. Die Heterogenität entsteht sowohl auf technischer, als auch auf betriebswirtschaftlicher Ebene und erschwert deren Abstimmung im Rahmen der Integration. Des Weiteren entstehen Probleme aus

unflexiblen Altsystemen. Diese können aufgrund starrer Prozessabfolgen, veralteter Datenhaltungssysteme oder Benutzeroberflächen oft nicht auf die neuen Anforderungen angepasst werden. Begünstigt wird dieser Faktor durch fehlende oder schlecht anzupassende Schnittstellen. Insgesamt weisen solche Altsysteme eine schlechte Wartbarkeit auf, da sie häufig Eigenentwicklungen darstellen, die nicht dokumentiert wurden und der Entwickler nicht mehr im Unternehmen arbeitet. Aus dem Problem der schlechten Integrationsfähigkeit aufgrund heterogener Applikationslandschaften und unflexibler Altsysteme ergibt sich ein nicht optimaler Integrationsgrad, der Ineffizienzen im Unternehmen hervorrufen kann.

Mitarbeiterfähigkeit: Integrationsprojekte fordern eine hohe Mitarbeiterqualifikation. Die Systemplaner müssen umfassende technische sowie betriebswirtschaftliche Kenntnisse besitzen, um bei der Planung des Integrationsprojektes alle wichtigen Aspekte einschätzen zu können. Z. B. sollte ein Projektverantwortlicher für die Einführung eines EIS die Anforderungen der Führungskräfte richtig verstehen und beurteilen können, um das Integrationsprojekt erfolgreich voranzutreiben.

Ausnahmeregelungen: Wichtig ist auch die Einbeziehung von nicht standardisierbaren oder automatisierbaren Vorgängen. Diese Sonderfälle müssen ebenso flexibel in den Arbeitsprozess integrierbar sein. Z. B. können Retourenabwicklungen in der Auftragsbearbeitung höchst differenziert und müssen gleichwohl wirtschaftlich ausführbar sein.

Hoher Wartungsaufwand: Aufgrund der vorgestellten Interdependenzen der Integrationskomponenten und der damit verbundenen Komplexität, gestaltet sich die Wartung dieser Komponenten aufwändig. So hat das Einspielen von Updates bei einer Komponente in der Regel die Anpassungen weiterer, mit dieser Komponente verknüpften Komponenten oder der verbindenden Schnittstellen zur Folge.

3.3 Integrationslösungen

Die vorgestellten Integrationsziele verdeutlichen, dass ein hoher Zusatznutzen durch die Integration erreicht werden kann. Anhand der aufgezeigten Probleme wird aber auch ersichtlich, dass damit Schwierigkeiten und ein hoher Aufwand verbunden sind. Um die Komplexität und die daraus resultierenden Problemen der Integration beherrschbar zu machen, wurden in der Wirtschaftsinformatik unterschiedliche Integrationslösungen entwickelt. Im Folgenden werden zwei aktuelle Lösungsansätze vorgestellt.

3.3.1 Enterprise Application Integration

Enterprise Application Integration (EAI) kann nicht als eigenständige Architektur oder Technologie betrachtet werden (vgl. Jung 2006, S. 174). Vielmehr bezeichnet EAI die Integration von Daten, Funktionen und Prozessen. Dazu werden aufeinander abgestimmte Vorgehensweisen, Architekturkonzepte und Instrumente benötigt. Mit EAI sollen die gewachsenen heterogenen Systemlandschaften in Organisationen zusammengeführt werden. Dies kann innerbetrieblich sowie zwi-

schenbetrieblich geschehen. Die vorhandenen Automationsinseln der Applikationen sollen dabei weitestgehend unverändert bleiben, was die Schwierigkeit der Einführung von EAI deutlich macht (vgl. Linthicum 1999, S. 3 ff.; Ferstl und Sinz 2013, S. 260 f.).

Mit EAI-Systemen sollen vorhandene Koordinations- und Integrationsprobleme gelöst werden. Diese entstehen auf syntaktischer, semantischer und pragmatischer Ebene⁹ (vgl. hier und im Folgenden Fleisch 2001, S. 124 ff.; Puschmann 2003, S. 140 ff.). In Abbildung 15 werden die verschiedenen Integrationsebenen und die darin eingebetteten Dienste von EAI-Systemen veranschaulicht. Anschließend werden diese ausführlich erläutert.

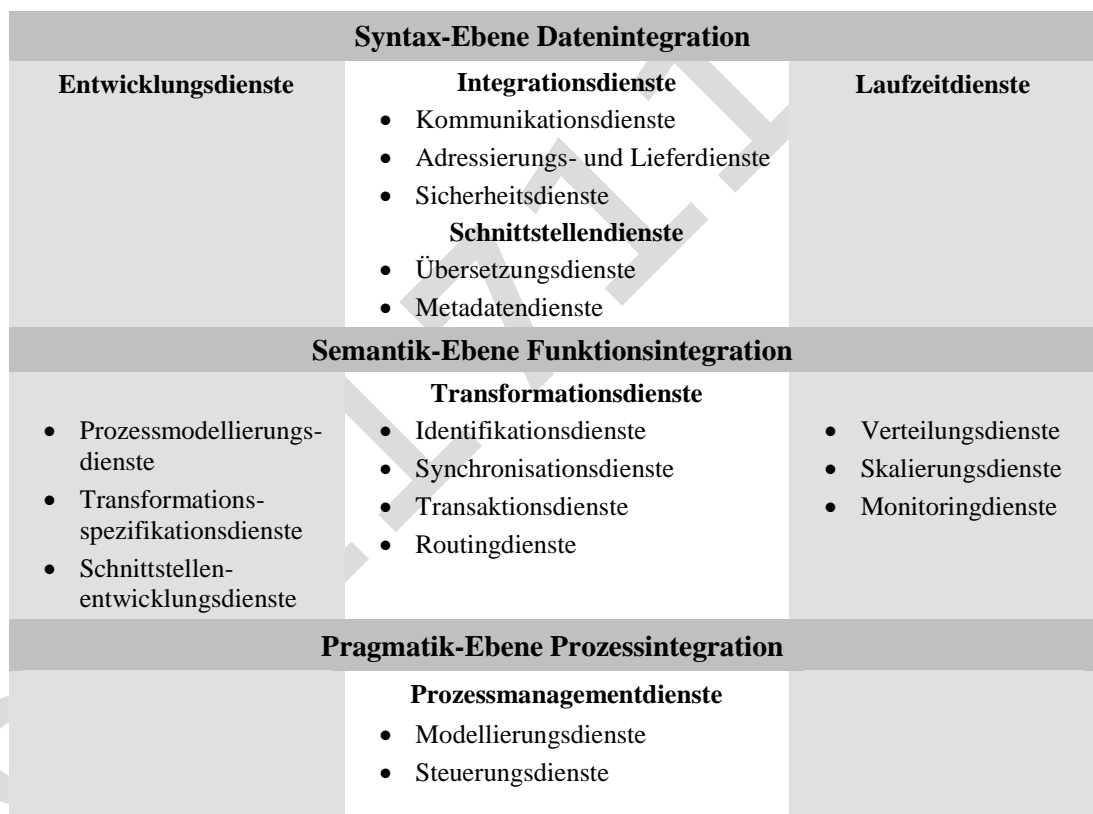


Abbildung 15: Integrationsebenen und Funktionalitäten von EAI-Systemen

Quelle: In Anlehnung an Puschmann 2003, S. 143 und Fleisch 2001, S. 126

Syntax-Ebene

Auf der Syntax-Ebene unterstützen EAI-Systeme die Datenintegration. Hierbei soll der Informationsaustausch zwischen den Applikationen in fehlerfreier und vollständiger Weise sichergestellt werden. Die Applikationen können untereinander Daten versenden und empfangen. Die Datenintegration dient als Basis der semantischen Integration und wird durch EAI-Systeme mit folgenden Integrations- und Schnittstellendiensten unterstützt:

⁹ Die Syntax-Ebene enthält die reine Darstellung von Informationen als Zeichen. In der Semantik-Ebene wird den Zeichen aus der Syntax-Ebene eine Bedeutung zugewiesen und auf der Pragmatik-Ebene wird diese zweckorientiert angewendet.

- **Kommunikationsdienste** übertragen Daten in synchroner oder asynchroner Weise.
- **Adressierungs- und Liefersdienste** unterstützen verschiedene Adressierungsschemata für die Datenübertragung zwischen den zu integrierenden Systemen.
- **Sicherheitsdienste** regeln den Zugriff der Applikationen auf die Informationen und garantieren die Integrität, Authentizität und Autorisierung von diesen Informationen.
- **Übersetzungsdienste** unterstützen die Datenintegration durch die Übersetzung von EAI-Programmierschnittstellen und Programmierschnittstellen der zu integrierenden Systeme.
- **Metadatendienste unterstützen** die EAI-Systeme bei der Bereitstellung von Daten und Informationen für die Semantik- und Pragmatik-Ebene. Hierfür werden bestimmte Regeln für den Eingang und Ausgang der Daten definiert.

Auf der Semantik-Ebene unterstützen EAI-Systeme die Funktionsintegration¹⁰. Die hier übertragenen Informationen sollen mit der richtigen Bedeutung bei der empfangenden Applikation ankommen. Hierfür muss das EAI-System die Informationen einer sendenden Applikation in die Semantik der empfangenden Applikation transformieren. Gleichzeitig übernimmt das EAI-System die Koordination mehrerer Nachrichten und bündelt diese. Folgende Transformationsdienste wendet das EAI-System dazu an:

Semantik-Ebene

- **Identifikationsdienste** unterstützen die Identifikation und Validierung von empfangenen Nachrichten. Mittels zugeordneter Metadaten¹¹, die die Datenstrukturen der integrierten Applikationen beinhalten, können die eingehenden Nachrichten erkannt und überprüft werden.
- **Synchronisationsdienste** unterstützen bei der zeitlichen Abstimmung zwischen den gesendeten und empfangenen Nachrichten.
- **Transaktionsdienste** stellen besonders in verteilten Systemen sicher, dass eine Funktion entweder vollständig erfolgt oder gar nicht ausgeführt wird, damit keine inkonsistenten Zustände zwischen Quell- und Zielsystemen entstehen.
- **Routingdienste** übernehmen die Verteilung von eingehenden Nachrichten

¹⁰ Puschmann 2003, S. 140 spricht hier von Objektintegration. Für ihn sind Objekte z. B. Kunde oder Produkt. Im Folgenden wird von Funktionsintegration als Adaption der Objektintegration gesprochen. Eine Funktion ist z. B. das Anlegen eines Kunden und basiert somit auf dem Objekt Kunde.

¹¹ Metadaten sind übergeordnete Daten, die die ihnen untergeordneten Daten beschreiben, z. B. können Metadaten Berechtigungen, Sicherheits- oder Leistungsanforderungen sein (vgl. Tilkov und Starke 2007, S. 21).

an die geforderten unterschiedlichen Zielsysteme. Das EAI-System benötigt die Routingdienste, um bei gewissen Dateninhalten bestimmte Geschäftsregeln auszuführen.

Pragmatik-Ebene

Auf der Pragmatik-Ebene unterstützen EAI-Systeme die Prozessintegration. Dabei werden hauptsächlich die unterschiedlichen Funktionen bzw. Prozessschritte aus den heterogenen Applikationslandschaften bzw. Organisationsstrukturen zweckgebunden koordiniert sowie die richtige Verwendung der übertragenen Informationen sichergestellt. Im Rahmen der Pragmatik werden folgende Funktionalitäten für das Prozessmanagement von verteilten Prozessen angeboten:

- Die **Modellierungsdienste** unterstützen bei der Gestaltung von Geschäftsprozessen in verteilten Organisationen (z. B. Kooperationen). Dies ermöglicht z. B. die Verteilung der einzelnen Prozessschritte auf die jeweiligen Geschäftseinheiten in einer Supply Chain.
- Die **Steuerungsdienste** vereinfachen die Koordination von innerbetrieblichen sowie zwischenbetrieblichen Prozessen.

Neben den Diensten, die hier für jede einzelne beschrieben wurden, bieten EAI-Systeme weitere Dienste an, die für alle Ebenen gemeinsam angeboten werden. Hierzu gehören die Entwicklungs- und die Laufzeitdienste.

Entwicklungsdienste

Entwicklungsdienste unterstützen die Prozessmodellierung und die Transformationsspezifikation. Ebenso bieten sie Methoden und Hilfsmittel bei der Anpassung und Entwicklung von neuen Schnittstellen an, die z. B. bei der Integration von Individualsystemen benötigt werden. Im Folgenden werden die einzelnen Unterstützungsfunktionen aufgezeigt, die ein EAI-System hierfür zur Verfügung stellt:

- **Prozessmodellierungsdienste** unterstützen durch die Bereitstellung einer grafischen Prozessmodellierungsumgebung die Modellierung von z. B. Zustandsdiagrammen. Ein besonderer Fokus liegt hierbei auf Simulationsumgebungen, die benötigte Validitätstests gezielter durchführbar machen.
- **Transformationsspezifikationsdienste** werden bei der Entwicklung von Mappingmechanismen eingesetzt. Einfache Transformationen können durch Drag-and-Drop Umgebungen bedient werden. Diese enthalten Verzeichnisstrukturen zum Auffinden von Quell- und Zielsystemen sowie Arbeitsflächen zum Konsolidieren von Quell- und Zielstrukturen. Komplexe Transformationen benötigen Programmbibliotheken, womit die Definition von Skripten unterstützt wird.
- **Schnittstellenentwicklungsdienste** bieten Methoden und Hilfsmittel zur Anpassung und Neuentwicklung von Adaptern an, die es ermöglichen, modifizierte Applikationen an Standardsysteme zu koppeln.

Laufzeitdienste dienen zur Verbesserung der Leistung, Skalierbarkeit¹², Verfügbarkeit und Zuverlässigkeit der Applikationen. Hier werden folgende Dienste bereitgestellt.

Laufzeitdienste

- **Verteilungsdienste** unterstützen bei der Implementierung eines EAI-Systems. Hierbei kann nach einer zentralen oder einer verteilten Architektur vorgegangen werden. Zentrale Architekturen implementieren die Integrations- und Schnittstellendienste lokal an den einzelnen Applikationen, während die Integrations-, Transformations- und Prozessmanagementdienste zentral installiert werden. Bei einer verteilten Architektur werden die einzelnen Integrations-, Transformations- und Prozessmanagementdienste nur an den Applikationen implementiert, an denen sie sinnvoll erscheinen.
- **Skalierungsdienste** verbessern die Skalierbarkeit und die Verfügbarkeit der Applikationen. Sie stellen Verfahren zur Verteilung der Arbeitslasten auf unterschiedlichen Servern bereit. Durch die generelle statische oder dynamische Verteilung der Arbeitslast auf verschiedene Server wird einerseits die Skalierbarkeit verbessert. Andererseits kann die Verfügbarkeit der Applikationen erhöht werden, indem mittels des Skalierungsdienstes Anfragen an so genannte Backup-Server umgeleitet werden. Der Backup-Server enthält die gleichen Daten, wie der Primärserver und ist sofort einsatzbereit, falls z. B. ein Ausfall des Primärservers droht.
- **Monitoringdienste** unterstützen die Überwachung der Systemverfügbarkeit und der Systemleistung.

Mit diesen Diensten und Funktionalitäten verfolgen EAI-Systeme bestimmte Zielsetzungen, die sich an die in Abschnitt 3.2 vorgestellten Integrationsziele anlehnen. Hier können als Überbegriffe die Redundanz, Verknüpfung, Konsistenz und Zielorientierung genannt werden (vgl. hier und im Folgenden Ferstl und Sinz 2013, S. 260 f.):

Nutzen

- EAI-Systeme helfen, **Redundanzen** in den heterogenen Systemlandschaften zu **erkennen** und mittels Eliminierung oder Deaktivierung von redundanten Komponenten deren Komplexität zu **reduzieren**.
- EAI-Systeme bieten die Möglichkeit, **Verknüpfungen** zwischen heterogenen Applikationen **herzustellen**. Dazu müssen entsprechende Kommunikationskanäle zwischen den Systemen geschaffen und die Schnittstellen einschließlich der Datenkonvertierung angepasst werden. In den meisten Fällen wird über eine Sterntopologie die Übertragung, Verteilung und

¹² Die Skalierbarkeit beschreibt das Leistungsverhalten z. B. einer Applikation auf eine Arbeitslaständerung. Demnach bezeichnet eine gute Skalierbarkeit einer Applikation die lineare Leistungssteigerung bei einer Erhöhung von Arbeitslast. Die Applikation hat dem gegenüber eine schlechte Skalierbarkeit, wenn die Leistung bei Arbeitslastenerhöhung unterproportional steigt oder sogar sinkt (vgl. Röder 2007, S. 256)

Konvertierung von Daten und Ereignissen gesteuert.

- Durch EAI-Systeme können **Konsistenzen**, die insbesondere durch auftretende Redundanzen verlorengegangen sind, durch Transaktionsmonitorsysteme **wieder erlangt werden**. Diese überwachen und steuern die Integritätsregeln und -bedingungen in der heterogenen Applikationslandschaft.

Es wird deutlich, dass EAI-Systeme Integrationsprobleme lösen, die durch heterogen gewachsene Applikationslandschaften in Unternehmen entstanden sind. Durch die Implementierung geeigneter Integrationstechnologien, wie z. B. Hub-and-Spoke- oder Bus-Architekturen, die im Folgenden erklärt werden, wird eine flexible Integration von geschäftsbereichsübergreifenden Prozessen in verteilten Organisationen (z. B. Kooperationen) oder Kunden und Lieferanten durch CRM-, EC- oder SCM-Systeme ermöglicht. Die Implementierung neuer und die Wartung von bestehenden Applikationen kann durch EAI-Systeme mit geringerem Aufwand bzw. Kosten erfolgen.

Probleme

Demgegenüber entstehen jedoch auch Probleme bei der Einführung und dem Gebrauch von EAI-Systemen. Diese ergeben sich hauptsächlich aus gestiegenen Anforderungen an die Mitarbeiter, die sich zusätzliches Wissen über prozessübergreifende Aufgaben, die neu integrierten Systeme und die neu entstandene Infrastruktur aneignen müssen. Zudem ist die Erzeugung der Konsistenz aufwändig und mit hohen Kosten verbunden.

Tabelle 3 gibt einen Überblick über den Nutzen und die Probleme die bei bzw. nach einer Einführung eines EAI-Systems entstehen können. Dabei wird deutlich, dass der Nutzen von EAI-Systemen die Probleme aus dem gewachsenen Aufwand und den Kosten übersteigt.

Nutzen eines EAI-Systems	Probleme eines EAI-Systems
<ul style="list-style-type: none"> • Redundanzen werden erkannt und können verringert werden • Verknüpfung von Applikationen wird ermöglicht • Konsistenz wird erhalten bzw. wiedererlangt • Grundlage für Automatisierung in internen und stabilen Netzwerken wird geschaffen • Verteilte Organisationsstrukturen werden ermöglicht • Reduktion der Softwarewartungs-, Integrations- und Migrationskosten sowie Kosten des Informationsaustausches 	<ul style="list-style-type: none"> • Gestiegene Anforderungen an Mitarbeiterfähigkeiten (prozessübergreifendes Prozesswissen, Wissen über die zu integrierenden Systeme, Wissen über Infrastruktur etc.) • Schaffung einer neuen Infrastruktur • Erzeugung von Konsistenz ist aufwändig und mit hohen Kosten verbunden

Tabelle 3: Nutzen und Probleme von EAI-Systemen

Quelle: In Anlehnung an Fleisch 2001, S. 127

Zur Umsetzung von EAI stehen, wie vorangegangen erwähnt, unterschiedliche Technologien bereit. Die einfachste Möglichkeit besteht in einer Punkt-zu-Punkt-Verbindung zwischen zwei Applikationen. Diese Art der Kommunikation zwischen den Applikationen birgt eine unüberschaubare Komplexität. Heutzutage

finden sich in der Regel im Rahmen von EAI die Architekturansätze Hub-and-Spoke und die Bus-Architektur.

Hub-and-Spoke Architekturen verknüpfen verschiedene Applikation über eine Vermittlungsstation (= Hub) in Form einer Sterntopologie. Jede Applikation hat einen gleichberechtigten Zugriff auf den Hub, der den gesamten Datenaustausch zwischen den verschiedenen Applikationen steuert und überwacht.

Die Hub-and-Spoke Architektur eignet sich insbesondere im Rahmen der datenflussorientierten Funktionsintegration für die Datenübertragung und Kommunikation zwischen den Applikationen. Ebenso ist die Unterstützung prozessorientierter Vorgehensweisen durch zentral abgelegte Arbeitsabläufe auf dem Hub gegeben.

Durch die zentrale Vermittlungsstation kann, gegenüber einer herkömmlichen Punkt-zu-Punkt-Verbindung, die Anzahl der Schnittstellen und damit die Komplexität enorm reduziert werden. Nachteilig ist allerdings die Kapazitäts- und damit Leistungsschranke, die durch den zentralen Hub vorgegeben ist (vgl. Ferstl und Sinz 2013, S. 262; Aier und Schönherr 2006, S. 190). Abbildung 16 veranschaulicht in einfacher Form den sternförmigen Aufbau einer Hub-and-Spoke Architektur.

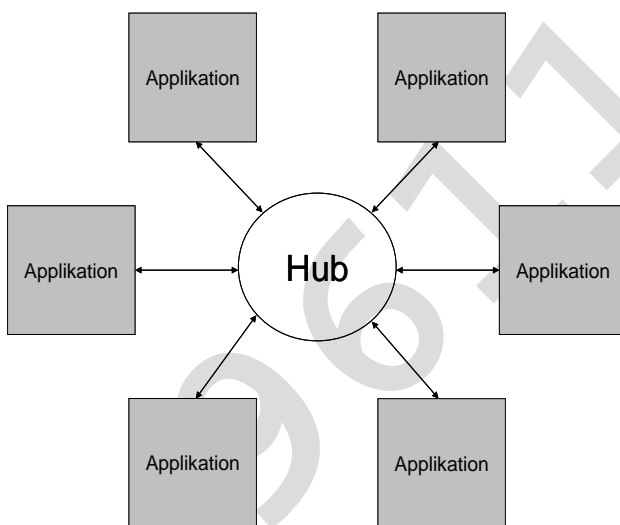


Abbildung 16: Hub-and-spoke Architektur

Quelle: Entnommen aus Ferstl und Sinz 2013, S. 262

Bei der **Bus-Architektur** wird der so genannte Bus in Form eines passiven Mediums verwendet und die Kommunikationsteilnehmer dieser dezentralen Topologie steuern die Ablaufkoordination selbst. Die angeschlossenen Applikationen können Informationen über den Bus zur Verfügung stellen und diese sind von allen anderen angeschlossenen Applikationen je nach Bedarf abrufbar.

Die Bus-Architektur ist besonders im Rahmen des Datenaustauschs einsetzbar. Hier kann sie speziell in Szenarien, in denen eine Applikation Daten an eine hohe Anzahl weiterer Applikationen abgeben muss, von Vorteil sein. Auch der umgekehrte Fall ist denkbar. Die Bus-Architektur findet also insbesondere beim Austausch von großen Datenmengen vieler unterschiedlicher Applikationen Anwen-

dung. Die einzelnen Applikationen verwalten dezentral ihre Ablaufsteuerung selber. Zentral kann die gesamte Ablaufsteuerung aller Applikationen, die an den Bus angeschlossen sind, über Ablaufsteuerungsobjekte verwaltet werden.

Die Bus-Architektur unterliegt dabei ebenso Kapazitäts- und Leistungsrestriktionen, wie die Hub-and-Spoke Architektur (vgl. Ferstl und Sinz 2013, S. 262 ff.). Abbildung 17 verdeutlicht den Aufbau einer Bus-Architektur mit den integrierten Ablaufsteuerungsobjekten.

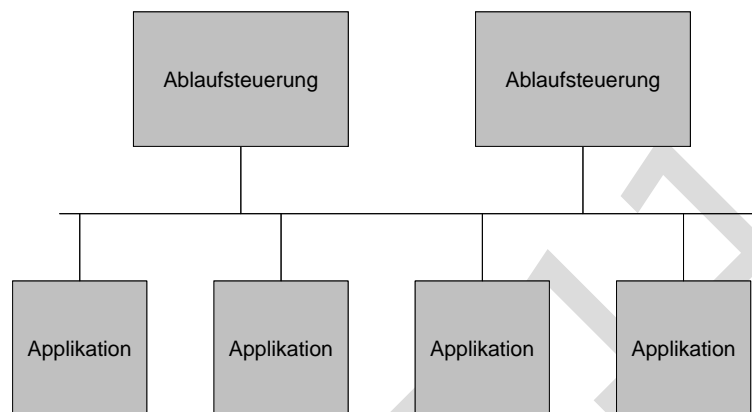


Abbildung 17: Bus-Architektur

Quelle: In Anlehnung an Ferstl und Sinz 2013, S. 263

In folgendem Beispiel wird abschließend der Aufbau einer EAI in der Praxis aufgezeigt.

Beispiel: EAI bei der Bosch Gruppe

Die international tätige Bosch Gruppe beschäftigt 190.000 Arbeitnehmer in über 132 Ländern und hat einen Jahresumsatz von mehr als 25 Milliarden US\$ (vgl. hier und im Folgenden Puschmann und Alt 2004). Sie besteht aus ungefähr 250 Tochtergesellschaften und zugehörigen Unternehmen. Von den insgesamt 185 Produktionsstandorten befinden sich 43 in Deutschland während die anderen weltweit, z. B. in Europa, Afrika, Asien oder Nord- und Südamerika, ansässig sind. Die geschäftlichen Aktivitäten der Bosch Gruppe können grob in die vier Geschäftsbereiche Automobilzubehör, Kommunikationstechnik, Konsumgüter und Investitionsgüter unterteilt werden.

Ein wesentlicher Erfolgsfaktor für ein Industrieunternehmen wie die Bosch Gruppe ist eine gute Unterstützung der internen und externen Geschäftsprozesse durch eine homogene IT-Infrastruktur. Zur Erreichung dieses Ziels sind die europäischen IT Aktivitäten in einer übergreifenden Abteilung namens QI gebündelt. Als eine interne Service-Abteilung soll die Automatisierung von Geschäftsprozessen mittels IT für geschäftliche und technische Funktionen unterstützt werden. Aufgrund der historisch gewachsenen und aus diesem Grund heterogenen IT-Architektur stand die Abteilung QI bei der Integration der IT-Systeme vor einer Reihe von Herausforderungen.

- Die Migration von bestehenden Systemen zu SAP R/3 und die Integration speziell angefertigter Systeme in eine neue IT-Architektur nach dem Best-of-breed Ansatz, erfordert die kostenintensive Entwicklung hunderter neuer Schnittstellen.
- Die Aufteilung nach verschiedenen Geschäftsbereichen erfordert flexible und bereichsübergreifende Kerngeschäftsprozesse, welche auf einer homogenen IT-Architektur aufsetzen und leicht konfiguriert werden können.
- Außerdem erfordert die Integration von Kunden oder Zulieferern in die Geschäftsprozesse die Integration neuer Systeme in die vorhandene Applikationsarchitektur. Zusätzlich sollen SCM- oder CRM-Systeme integriert werden.
- Zur Lösung dieser Herausforderungen sind ca. 700 Schnittstellen zu entwickeln und zu pflegen.

Nachfolgend werden einige Problemfelder näher erläutert.

Integration der ERP-Systeme

Die meisten der Tochtergesellschaften und Produktionsstandorte verfügen über individuell entwickelte Systeme, die genau auf die eigenen Bedürfnisse abgestimmt sind. Zur Harmonisierung und Erlangung einer flexiblen und standardisierten IT-Architektur will die Bosch Gruppe daher zu dem Standard-ERP-System SAP R/3 migrieren. Eines der größten Probleme stellen dabei die unterschiedlichen Anforderungen der einzelnen Unternehmen dar. Allein durch die Einführung von SAP R/3 wird erwartet, dass zusätzliche 300 Schnittstellen zu entwickeln sind.

Integration der EC-Systeme

Für ein einheitliches Auftreten gegenüber Geschäfts- und Privatkunden sollen die EC-Applikationen unter einem Bosch-Portal vereinigt werden. Dazu müssen verschiedene IT-Systeme unterschiedlicher Bereiche integriert werden. Die dafür notwendigen Schnittstellen sind unbekannt.

Integration der SCM-Systeme

Unter anderem durch die Einführung des einheitlichen Bosch-Portals bestehen zusätzliche Anforderungen an die bestehenden SCM-Systeme, wie verkürzte Durchlaufzeiten oder Verfügbarkeitsprüfungen. Z. B. wird die Einführung des SAP Advanced Planer and Optimiser (APO) geplant, was die weitere Integration verschiedener Systeme bedeutet und die Entwicklung von ungefähr 200 Schnittstellen bedeuten würde.

Um diesen Problemen zu entgehen, hat sich die Bosch Gruppe dazu entschlossen, eine standardisierte Integrationsarchitektur mit dem Namen „business bus“ zu implementieren. Bei der Umsetzung dieses Vorhabens erwies es sich als schwierig, ein passendes EAI-System zu finden. Zu unterschiedlich waren die einzelnen Systeme aufgrund der ungleichen Schwerpunktlegung auf die verschiedenen Integrationsebenen (Prozess, Objekt, Daten) und zu umfangreich waren die Anforderun-

gen von Seiten der Bosch Gruppe. Die Integration verschiedener individuell entwickelter Systeme ließ die Verwendung von Standardadaptern nicht zu. Außerdem musste eine Kompatibilität mit bereits vorhandenen Integrationsstandards vorherrschen. Bei der Datenintegration musste sowohl die asynchrone (bspw. ERP-System) als auch die synchrone (bspw. EC-Systeme) Integration möglich sein. Auch musste die Integration alter Systeme wie Siemens BS2000 gewährleistet sein.

Damit alle Geschäftsprozesse und Applikationen unterstützt wurden, entschied sich die Bosch Gruppe bei der Umsetzung des „business bus“ für drei unterschiedliche EAI-Systeme, die in dem „business bus“ vereint wurden. Durch die Entwicklung des „business bus“ mittels der Verwendung der EAI-Systeme konnte folgender Nutzen erzielt werden:

- Umsetzung des Best-of-breed Ansatzes.
- Eine größere Flexibilität bei der Geschäftsprozessintegration.
- Bietet die Möglichkeit für eine flexiblere Integration von Geschäftspartnern und Kunden mittels CRM, EC und SCM-Systemen.
- Kosteneinsparung bei der Implementierung bestehender und neuer Systeme.

Die folgende Abbildung zeigt die ursprüngliche Punkt-zu-Punkt Anbindung neuer System bei der Bosch Gruppe.

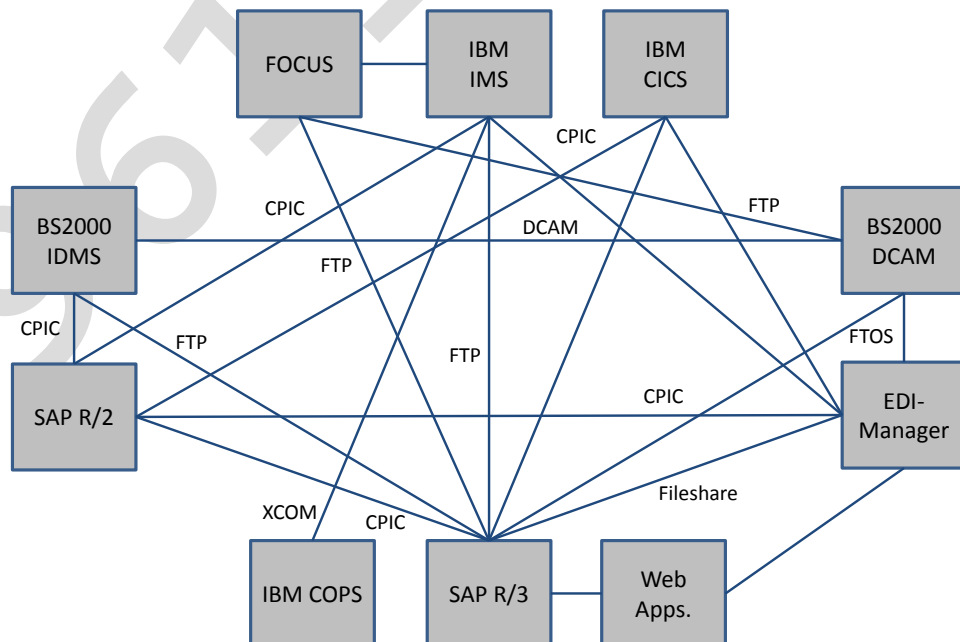
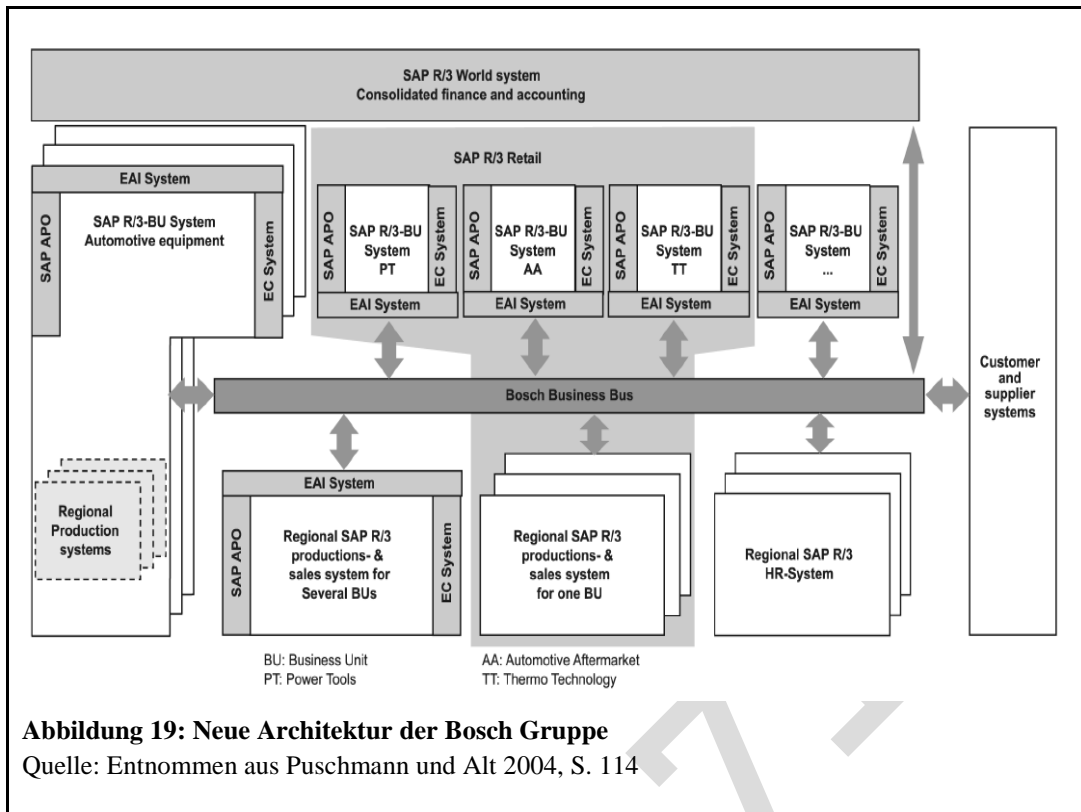


Abbildung 18: Punkt-zu-Punkt Anbindung der Bosch Gruppe

Quelle: Entnommen aus Puschmann und Alt 2004, S. 111

Die nachstehende Abbildung zeigt die neue Architektur, umgesetzt mit dem „business bus“ und den drei EAI-Systemen.



EAI stellt also ein systematisches Konzept dar, um die Integration auf allen Unternehmensebenen voranzutreiben. Aufgrund aber der dabei eingesetzten und damit für den jeweiligen Kontext festgeschriebenen Technologien wird die Flexibilität und Kapazität eingeschränkt. Dieses Problem versucht die so genannte Serviceorientierung aufzuheben. Serviceorientierte Architekturen bieten technologie-unabhängige Integrationslösungen und damit eine hohe Flexibilität mit geringen Kapazitätsgrenzen. Der folgende Abschnitt erfasst die Grundlagen zu Serviceorientierten Architekturen.

3.3.2 Serviceorientierte Architektur

Serviceorientierte Architekturen (SOA) können auch als prozessorientierte Integrationsarchitekturen beschrieben werden. Sie enthalten unterschiedliche Dienste, auch Services genannt, die die Verbindung zwischen der Geschäftsprozess- und der Applikationsarchitektur herstellen. Der Aufbau einer SOA basiert dabei nicht wie bei EAI auf technischen Komponenten (z. B. Hub-and-Spoke- oder Bus-Architekturen). Vielmehr beschreibt eine SOA die funktionalen Unterstützungen im Rahmen des Geschäftsprozessmanagements mittels standardisierter Services.

Eine SOA beschäftigt sich mit dem Entwurf von im Laufe der Zeit gewachsenen, heterogenen Applikationslandschaften. Diese enthalten in der Regel Redundanzen im Daten- und Funktionsbereich. Durch eine SOA wird versucht, die darin bereits bestehenden Gestaltungsansätze zu erweitern und über standardisierte Schnittstellen den Integrationsgrad dieser heterogenen Applikationen zu erhöhen (vgl. Heutschi 2007, S. 3).

Die zentralen Begriffe zu SOA sind (vgl. Conrad 2006, S. 89 f.):

- **Service:** Ein Service beschreibt eine nachrichtenorientierte Funktionalität, die über standardisierte Schnittstellen zwischen Applikationen abgerufen werden kann. Hierbei sind Services untereinander lose gekoppelt und können als wieder verwendbare Teil-Geschäftsprozesse bezeichnet werden. Es kann zwischen wertschöpfenden Services (z. B. Service für einen Bestellprozess oder eine Produktlizenzvergabe), unterstützenden Services (z. B. Service für die Kundenverwaltung über ein Call Center oder die Lohnabrechnung) und technischen Services (z. B. Infrastrukturservices) unterschieden werden (vgl. Stein und Ivanov 2007, S. 214 ff.).
- **Schnittstellenbeschreibung:** Hier werden die Funktionalitäten und Aktivitäten der einzelnen Services beschrieben und deren Qualitätskriterien (z. B. Sicherheitsmechanismen) und Transaktionen erfasst. Die Schnittstellenbeschreibung bildet ein wichtiges Konzept im Rahmen von SOA, da hier erst ein genauer Überblick über die zahlreichen Services und deren Abhängigkeiten untereinander geschaffen wird und diese dadurch verwaltbar sind.
- **Service-Repository:** Im Rahmen von Entwicklungsprozessen müssen die Services oder Metadaten zentral verwaltet werden. Das Service-Repository unterstützt die Entwicklung oder betriebswirtschaftliche Konzeption über den gesamten Lebenszyklus hinweg.
- **Service-Registry:** In der Service-Registry werden die Services lediglich beschrieben. In ihr werden alle Details zu den Services gesammelt. Während der Laufzeit dient die Service-Registry dem einfachen Suchen und Finden der einzelnen Services.

Die Einsatzgründe für eine SOA sind vielfältig und je Unternehmen höchst individuell. Als allgemeingültige Gründe können jedoch folgende Probleme genannt werden:

- Die Geschäftsprozesse der Organisation werden nicht ausreichend unterstützt.
- Die Geschäftsanforderungen können nicht flexibel genug umgesetzt werden.
- Der IT-Betrieb unterliegt insgesamt sehr hohen Kosten.
- Die IT-Architektur kann nur unzureichend an notwendige technische Neuerungen angepasst werden.

Ziele

Um diese Probleme zu beheben, verfolgt die SOA die Ziele der Standardisierung, Entkopplung und Flexibilisierung. Die SOA soll zur **Standardisierung** dienen, indem sie in Form einer Integrationsinfrastruktur die Integration von heterogenen Applikationen durch die bereitgestellten Services erleichtert und dadurch die Anzahl der Schnittstellentechnologien reduziert. Dadurch können Integrations- und Wartungskosten gesenkt werden. Die SOA hilft auch bei der **Entkopplung** von Applikationsinseln, die in der Regel im Laufe der Zeit in Unternehmen entstehen.

Die Redundanzen und Abhängigkeiten zwischen den Applikationen können durch eine SOA verringert werden. Dadurch werden Kosten und Projektrisiken gesenkt, die im Rahmen des IT-Betriebs und für Neu- und Weiterentwicklungen der Applikationen anfallen. Im Rahmen der Benutzer- und Prozessintegration kann durch SOA eine **Flexibilisierung** erreicht werden. Die Anpassung oder Neuentwicklung von applikationsübergreifenden Prozessen gestaltet sich durch SOA einfacher und schneller.

Eine SOA dient im Rahmen des Geschäftsprozessmanagements als Infrastruktur. Ihr Aufbau kann nach Ebenen und Sichten erfolgen. Die zu unterscheidenden Ebenen sind (vgl. hier und im Folgenden Heutschi 2007, S. 12 ff.):

- **Applikationsebene:** Auf der Applikationsebene werden alle betrieblichen Applikationen abgebildet. Sie präsentiert somit eine Ressourcensicht und unterstützt die im Unternehmen implementierten Applikationsfunktionen und Datensammlungen.
- **Serviceebene:** Die Serviceebene enthält Services, die die betrieblichen Applikationen auf der Applikationsebene als Schnittstellen unterstützen. Sie dient dabei als Schnittstellen- und Kommunikationsschicht. Ziel ist, die Applikationsfunktionen und Daten möglichst redundanzfrei zu strukturieren und zu kapseln.
- **Workflowintegrationsebene:** Im Rahmen des Geschäftsprozessmanagements können Prozesse auch (teil-)automatisiert ablaufen. Workflows steuern die Ablauflogik von Prozessen durch Weiterleitung von Dokumenten, Informationen und Aufgaben von einer Ressource zur nächsten. Dieser Ablauf geschieht anhand vorgegebener Regeln für applikationsübergreifende Geschäftsprozesse.
- **Desktopintegrationsebene:** Die Desktopintegrationsebene fokussiert die Mitarbeiter- bzw. Benutzerrollensichtweise. Hier werden alle von einer Person bzw. einem Arbeitsplatz zur Aufgabenerfüllung benötigten Applikationsfunktionen gebündelt.

Auf die vier Ebenen wird zusätzlich eine unterschiedliche Sicht vorgenommen:

- Die **applikationsbezogene Sicht** bezieht sich auf Architekturkomponenten, die die Umsetzung der betriebswirtschaftlichen Logik, wie Applikationslogik oder Workflows unterstützen.
- Die **applikationsneutrale Sicht** betrachtet die Architektur von der Integrations- und Infrastrukturseite. Hier erfolgt die Bereitstellung von Implementierungsdiensten und -protokollen zur Systemintegration.

Abbildung 20 veranschaulicht die Architekturkomponenten einer SOA mit ihren verschiedenen Ebenen und Sichten.

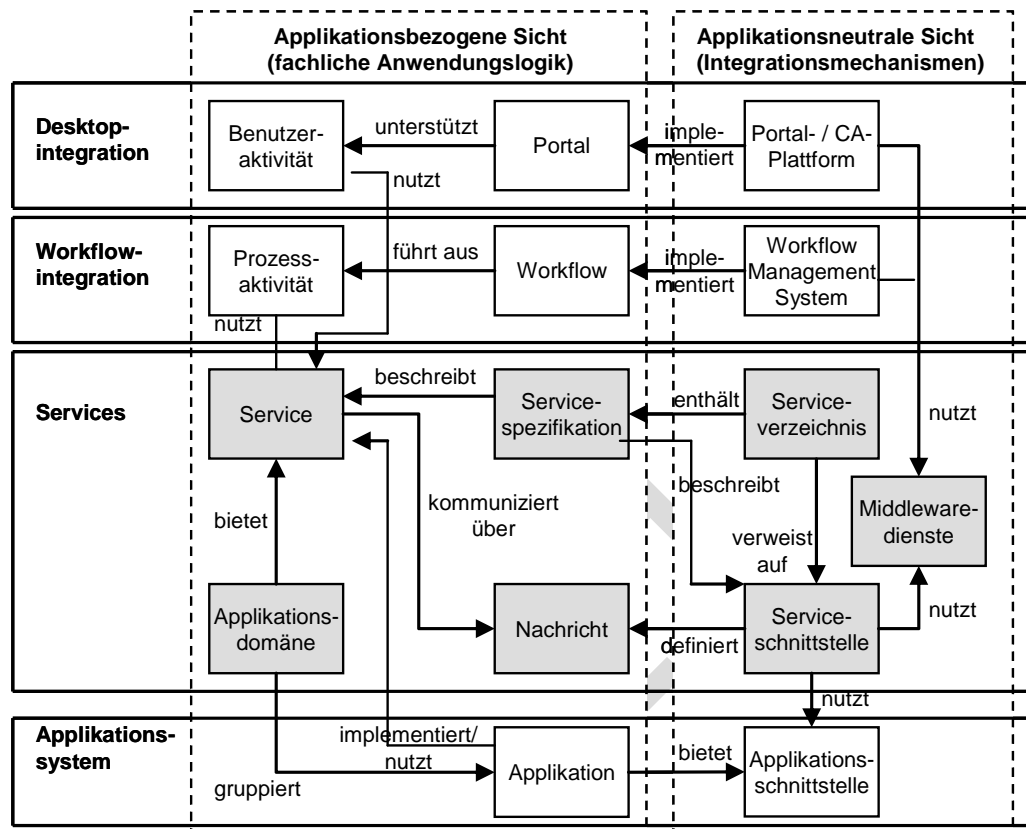


Abbildung 20: SOA-Ebenen und -sichten

Quelle: Entnommen aus Heutschi et al. 2006, S. 364

Designprinzipien

Dieser Aufbau einer SOA und deren Services werden nach unterschiedlichen Designprinzipien gestaltet. Folgende grundlegende Designprinzipien für eine SOA sind anzuführen und gliedern sich in vier Klassen (vgl. hier und im Folgenden Heutschi 2007, S. 34 ff.):

Schnittstellenorientierung: Bei der Schnittstellenorientierung werden Implementierungsdetails abstrahiert, da sie aus Sicht eines Servicenutzers irrelevant sind. Hierfür können folgende Designprinzipien aufgezeigt werden:

- Abstraktion von der Serviceimplementierung
- Umfassende, einheitliche Servicespezifikation
- Stabile, gesteuerte Serviceverträge

Interoperabilität: Beim Servicedesign müssen verschiedene Standardisierungsentscheidungen erfolgen, die einerseits technische Aspekte, wie z. B. Schnittstellentechnologien und andererseits betriebswirtschaftliche Aspekte, wie z. B. eine standardisierte Begriffswelt betrachten muss. Dadurch werden die Interoperabilität von Services und die kontextunabhängige Benutzung gewährleistet. Folgende Designprinzipien können aufgezeigt werden:

- Technische Schnittstellenstandardisierung
- Betriebswirtschaftliche Schnittstellenstandardisierung
- Verwendung offener und verbreiteter Industriestandards

Autonomie und Modularität: Um die Autonomie und Modularität einer SOA zu wahren, werden die relevanten Ressourcen oder Funktionen, die hohe Interdependenzen aufweisen, anhand bekannter Modul- oder Komponentenbildungsprinzipien zusammengefasst. Durch diese Auflösung der ursprünglich vorhandenen Applikationsarchitekturen entstehen teilautonome Subarchitekturen, wie z. B. Domänen und Services, die gegenüber den anderen Subarchitekturen nur eine lose Kopplung mit geringen Interdependenzen aufweisen. Hierbei gelten folgende Designprinzipien:

- Hohe Servicekohäsion und schwache logische Kopplung
- Lose gekoppelte Kommunikation

Bedarfsorientierung: Im Rahmen eines betriebswirtschaftlich sinnvollen und wertschöpfenden Leistungsumfangs orientieren sich Services am geschäftsspezifischen Bedarf der Objekte und Prozessaktivitäten. Folgende Designprinzipien sind dazu anwendbar:

- An Geschäftsprozessen orientierte, grobe Servicegranularität
- Generalisierung der Serviceleistung

Anhand der genannten Designprinzipien kann die Implementierung einer SOA erfolgen. Die Implementierung einer SOA ist sehr komplex. Eine schrittweise Planung und Einführung anhand von Teilprojekten hilft, diese Komplexität zu beherrschen. Die Planung der einzelnen Projektschritte kann z. B. folgendermaßen aufgebaut sein:

Implementierung

Die **Rollen und Verantwortlichkeiten im SOA-Projekt** müssen festgelegt werden. Hier können entweder neue Stellen besetzt oder bereits bestehenden Rollen neue Kompetenzen und Aufgaben zugeteilt werden. Sinnvoll ist außerdem eine Verankerung des Projekts in der IT-Organisation des Unternehmens.

Die **Ziele und Einsatzbereiche der SOA** müssen genau definiert werden. Hierzu können die Architekturverantwortlichen Situations- und Potenzialanalysen durchführen.

Anhand der definierten Ziele können dann die **Designprinzipien** für die Architektur festgelegt werden. Hierfür müssen die Serviceentwicklung sowie die Servicenutzung festgelegt werden. Die Integrationsinfrastruktur ist ebenso abzustimmen. Im Rahmen des Servicedesigns sind hierfür standardisierte Entwicklungs- und Bewertungsprozesse sowie Designprinzipien nötig.

Um **Zuständigkeits- und Kopplungsentscheidungen**¹³ zu unterstützen, müssen die Unternehmen die Applikationsarchitekturen aus betriebswirtschaftlicher Sicht strukturieren.

¹³ Hierbei wird entschieden, an welchen Stellen Applikationen entkoppelt werden können und über Services integriert werden bzw. ob andere Kopplungsmechanismen nötig sind.

Die Unternehmen müssen entscheiden, auf welcher **Unternehmensebene** die SOA implementiert werden soll.

Um die SOA laufend zu steuern und zu überwachen, ist ein **Architekturmanagement** (vgl. Kapitel 4) nötig. Anhand von Kennzahlen können die Wirksamkeit und Wirtschaftlichkeit der Architektur betrachtet werden.

Eine SOA stellt kein einmaliges Implementierungsprojekt dar. Vielmehr müssen alle Geschäftsprozesse durchleuchtet und teilweise reorganisiert werden. Nach der Einführung ist eine regelmäßige Überwachung und Steuerung nötig. Somit ist die SOA als dynamischer Prozess in das Unternehmen zu integrieren.

Abschließend wird die Einführung einer SOA am Beispiel der Deutschen Post Brief erläutert.

Beispiel: Die Implementierung einer SOA bei der Deutschen Post Brief

(vgl. Heutschi und Schemm 2005)

Aufgrund der Erkenntnis, dass die existierende, historisch gewachsene IT-Architektur den Agilitätsanforderungen nicht mehr gerecht wurde, hat sich die Deutsche Post Brief 1999 zur Neugestaltung der IT-Architektur nach Prinzipien einer Serviceorientierten Architektur entschlossen. In der Vergangenheit wurden Projekte der Fachbereiche zusammen mit der zentralen IT durchgeführt, jedoch fehlte eine projektübergreifende Koordination. Das Ergebnis war eine heterogene Applikationslandschaft mit großen funktionalen und datenbezogenen Redundanzen. Dies hatte zur Folge, dass IT-Projekte, z. B. durch unzureichend dokumentierte Abhängigkeiten, zunehmend risikoreicher wurden und die bestehende IT-Architektur zusätzlich immer teurer in der Unterhaltung wurde. Mit der Zielsetzung unternehmerische Agilität und technische Flexibilität zu erlangen, beschloss die Deutsche Post Brief daraufhin die Entwicklung einer Serviceorientierten Architektur.

Zur Verwirklichung wurde ein strategisches Projekt initiiert, das eine logische Applikationsarchitektur als Generalbebauungsplan für eine redundanzfreie Implementierung von Funktionalitäten und eine konsistente Datenpflege als Ziel hatte. Auf Basis der geschäftlichen Anforderungen der Fachbereiche wurde begonnen, die betriebswirtschaftliche Applikationsarchitektur zu entwickeln. Die relevanten Geschäftsprozesse wurden in betriebswirtschaftlich funktionale Bestandteile zerlegt und die generischen Geschäftsfunktionen und Datenflüsse bestimmt. Geschäftsfunktionen mit Bedarf an IT-Unterstützung wurden anhand wechselseitiger Abhängigkeiten in betriebswirtschaftlichen Applikationsdomänen zusammengefasst. Mit Hilfe einer Sammlung von Fachklassen wurde aus den in den Domänen zusammengefassten Geschäftsfunktionen und -objekten Services gebildet.

Zur technischen Umsetzung wurde eine Integrationsarchitektur entwickelt. Der so genannte „Service Backbone“ stellt dabei den Kernbestandteil der technischen Umsetzung der SOA dar. Er unterstützt als Kernfunktion die Kommunikation

zwischen den Serviceanbietern und -nutzern. Bei der Implementierung des „Service Backbones“ wurde großer Wert auf eine Anbieterunabhängigkeit gelegt. Mittels des Best-of-Breed Ansatzes wurden Infrastrukturkomponenten verschiedenster Anbieter und Eigenentwicklungen zusammengeführt.

Bis zum Jahr 2005 konnte die Deutsche Post Brief durch die Einführung einer Serviceorientierten Architektur insbesondere folgende Nutzenpotenziale realisieren:

- Wiederverwendung von Services und Vermeidung redundanter Softwareentwicklung
- Kürzere Projektlaufzeiten
- Niedrige Wartungskosten
- Dezentralisierung der Applikationsentwicklung und bessere Reaktionsfähigkeit auf neue Geschäftsanforderungen

Das Unternehmen sieht weitere geschäftliche und technische Nutzenhebel wie z. B. den Schutz bereits bestehender IT-Investitionen.

Die Architekturverantwortlichen nennen als wesentliche Erfolgsfaktoren für die erfolgreiche Umsetzung bei der Deutschen Post Brief

- das Starten der Entwicklung aus einer geschäftsorientierten und betriebswirtschaftlich-logischen Sicht,
- die Unterstützung durch das Top-Management und
- die Herstellerunabhängigkeit und Standardkonformität der Serviceinfrastruktur zur Wahrung der technischen Zukunftsfähigkeit und Flexibilität.

Aufgrund des Erfolges der Einführung einer Serviceorientierten Architektur bei der Deutschen Post Brief wurde eine Ausweitung auf andere Unternehmensbereiche wie Express/Logistik vorgenommen.

3.4 Übungsaufgaben

1. Was wird unter dem Begriff Integration im Zusammenhang mit dem Bereich des Informationsmanagements verstanden?
2. Nennen Sie die unterschiedlichen Integrationsperspektiven und geben Sie mindestens zwei Ausprägungen je Perspektive an.
3. Was wird unter Datenintegration verstanden?
4. Was wird unter Funktionsintegration verstanden?
5. Nennen Sie fünf Ziele, die mit der Integration verfolgt werden.
6. Worin besteht der Nutzen von EAI-Systemen?
7. Was wird unter einer SOA verstanden?

4 Architekturmanagement

Dieses Kapitel beschäftigt sich mit dem Management von Architekturen. Hierbei werden zunächst die Ziele und Aufgaben des Architekturmanagements vorgestellt und daran anschließend die Architekturmanagementprozesse im Rahmen des Architekturlebenszyklus aufgezeigt. Abschließend werden vier Vorgehensmodelle zum Architekturmanagement beschrieben.

4.1 Ziele und Aufgaben des Architekturmanagements

Die in den vorherigen Kapiteln aufgezeigte Komplexität im Bereich der Architekturen und Integration kann nur durch ein ganzheitliches Architekturmanagement bewältigt werden. *Müller et al.* (2006, S. 187) verstehen unter Architekturmanagement „[...] alle Aspekte der Erstellung, Pflege und des Einsatzes von IT-Architekturmodellen. Neben der technischen Umsetzung kümmert sich das [...] Architekturmanagement auch um die Etablierung geeigneter Organisationsstrukturen und Prozesse im Unternehmen.“ Es ist erkennbar, dass sich Architekturmanagement nicht nur auf die technische Ebene bezieht, sondern immer auch die betriebswirtschaftlichen Aspekte beachtet. Im Rahmen des Architekturmanagements werden somit Bebauungspläne entwickelt und gepflegt, Prozesse, Funktionen, Rollen und Verantwortlichkeiten definiert und auch Architekturbewertungen im Rahmen der Steuerung etabliert.

Eine radikale Veränderung der Unternehmens- bzw. der Informationssystemstrukturen ist dabei nicht das Hauptziel des Architekturmanagements. Es dient eher der Flexibilisierung und der Nachhaltigkeit von Architekturen und folgt einem **evolutionären Charakter**. Damit lässt sich das Architekturmanagement der in KE 1 beschriebenen Kybernetik II zuordnen. Es verfolgt folgende Ziele (vgl. Hafner 2006, S. 90 f.; Jung 2006, S. 313; Schelp et al. 2004, S. 64):

Ziele

- Unterstützung des Business/IT-Alignment (vgl. hierzu KE 3)
- Erhaltung der Transparenz und Flexibilität von Geschäftszielen, Prozessen, Applikationen und der Infrastruktur
- Optimierung von Kosten im Hinblick auf z. B. Infrastruktur- und Entwicklungskosten
- Unterstützung der Standardisierung und Redundanzvermeidung von Prozessen, Applikationen und benötigter Infrastruktur
- Erhöhung der Wartbarkeit und Entwicklungsgeschwindigkeit von Architekturkomponenten
- Aufrechterhaltung und Weiterentwicklung von Architekturen durch Einführung neuer Technologien bzw. Abschaltung nicht benötigter Technologien
- Sicherstellung der Architekturkonformität durch Berücksichtigung von betriebswirtschaftlichen Anforderungen und technischen Restriktionen

Das Architekturmanagement dient somit, im Rahmen des gesamten Informatikmanagementprozesses, der Wirksamkeit, Wirtschaftlichkeit und Sicherheit der Informatik. Diese Ziele sind konform mit dem IT-Governanceprozess (vgl. KE 3). Hieraus ergeben sich spezifische Aufgaben für das Architekturmanagement, welche nach *Niemann* in strategische und operative Architekturaufgaben aufgeteilt werden. Das Architekturmanagement lässt sich neben Anforderungs- und Portfoliomanagement (strategisches Architekturmanagement) sowie Programm- und Servicemanagement (operatives Architekturmanagement) in das IT-Strategie-Framework eingliedern. Abbildung 21 stellt die Zusammenhänge mit den oben genannten Zielen des Architekturmanagements grafisch dar (vgl. hier und im Folgenden Niemann 2006, 38 ff.).

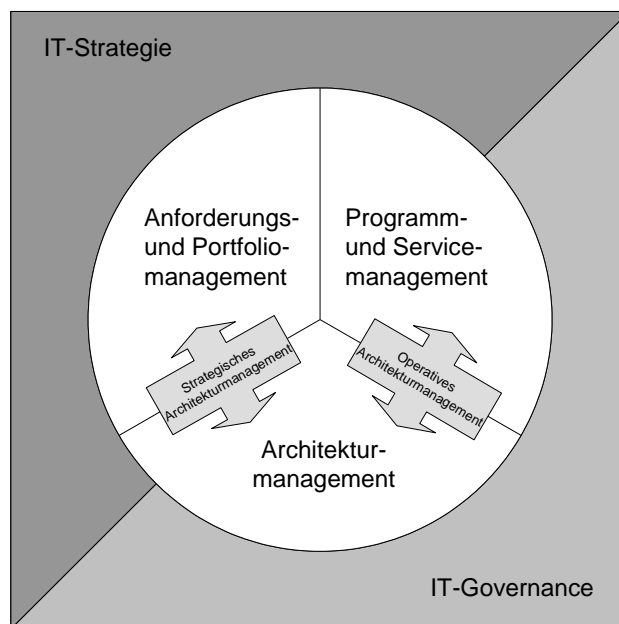


Abbildung 21: IT-Strategie-Framework

Quelle: Entnommen aus Niemann 2006, S. 9

Strategische Aufgaben

Zu den strategischen Aufgaben des Architekturmanagements gehört die **Dokumentation, Analyse und Planung** von Architekturen. Hierzu zählt im Einzelnen:

- die Erhebung der Ist-Architektur
- die Analyse der Ist-Architektur
- die Auswertung des Anforderungs- und Portfoliomanagements
- die Synchronisation zwischen Anforderungs- und Portfoliomanagement und den Ergebnissen der Ist-Architekturanalyse
- die Definition einer Soll-Architektur.

Operative Aufgaben

Die operativen Aufgaben des Architekturmanagements umfassen die durchgängige **Entwicklung, Umsetzung und Steuerung** der Architekturen. Hierunter fallen z. B.

- die Definition von Referenzarchitekturen und Einsatzszenarien
- die Festlegung von Vorgehensweisen und Verantwortlichkeiten
- die Entwicklung und Pflege wieder verwendbarer Komponenten
- die Architekturkommunikation (Publikation, Schulung, Beratung)

- die Prüfung der Architekturkonformität von Änderungsvorhaben.

Um diese Aufgaben zielgerichtet zu erfüllen, muss das Architekturmanagement folgenden Anforderungen gerecht werden (vgl. Hafner 2006, S. 91 ff.; Umek und Tannhäuser 2006, S. 57):

Anforderungen

- **Skalierbarkeit:** Das Management von Architekturen muss nicht nur deren Standardisierung sicherstellen, sondern auch Ansätze zur Skalierbarkeit liefern. Hiermit soll die Komplexität und Dynamik von Architekturen beherrschbar gestaltet werden.
- **Evolutionärer Charakter:** Wie oben bereits erläutert, hat das Architekturmanagement einen evolutionären Charakter, der Kybernetik II entsprechend. Die Architektur muss sich im Rahmen der Zielsetzung der Flexibilität stetig an das dynamische Unternehmensumfeld anpassen können.
- **Fokussierung auf Inkonsistenzen:** Diese Anforderung knüpft an die erste und zweite Anforderung an. Ausgehend von der Annahme, dass Architekturen immer Inkonsistenzen in sich bergen, soll der Fokus hierbei auf den situationsbedingten Umgang mit diesen gelegt werden. Eine Beseitigung aller Inkonsistenzen ist in der Regel nicht ausführbar und sollte daher auch nicht das Ziel sein.
- **Visionsbildung:** Die Formulierung einer Architekturvision, initiiert durch betriebswirtschaftliche Anforderungen und technische Restriktionen von Architekturen soll helfen, ein Rahmenwerk mit Zieldefinitionen und Meilensteinen zu schaffen.
- **Einsatz von Modellen, Methoden und Kennzahlensystemen:** Der Einsatz von Modellen, Methoden und Kennzahlensystemen im Rahmen des Architekturmanagements kann messbar zum Erfolg der Architektursteuerung beitragen.
- **Anschlussfähigkeit:** Die Anschlussfähigkeit des Architekturmanagements gegenüber weiteren Aufgaben des Informationsmanagements oder des IT-Controllings sollte stets gewährleistet sein. Hierzu dient z. B. die Definition von Informationsschnittstellen.
- **Organisatorische Einbettung:** Das Architekturmanagement sollte, der Anschlussfähigkeit folgend, in die gesamte Aufbau- und Ablauforganisation eines Unternehmens eingebettet werden, z. B. als eigene Abteilung oder auch mittels Gremien.

An diesem umfangreichen Anforderungskatalog ist die Komplexität eines gut funktionierenden Architekturmanagements erkennbar. Hierbei ist die Orientierung am Architekturlebenszyklus sinnvoll, um in jeder Phase ein wirtschaftliches Management zu gewährleisten.

4.2 Architekturlebenszyklus

Die Architektur folgt einem Lebenszyklus, der sich in regelmäßigen Abständen wiederholt, um sich den veränderten Betriebs- und Wettbewerbsbedingungen anzupassen. Abbildung 22 präsentiert die einzelnen Phasen, welche in den folgenden Abschnitten ausführlich vorgestellt werden.

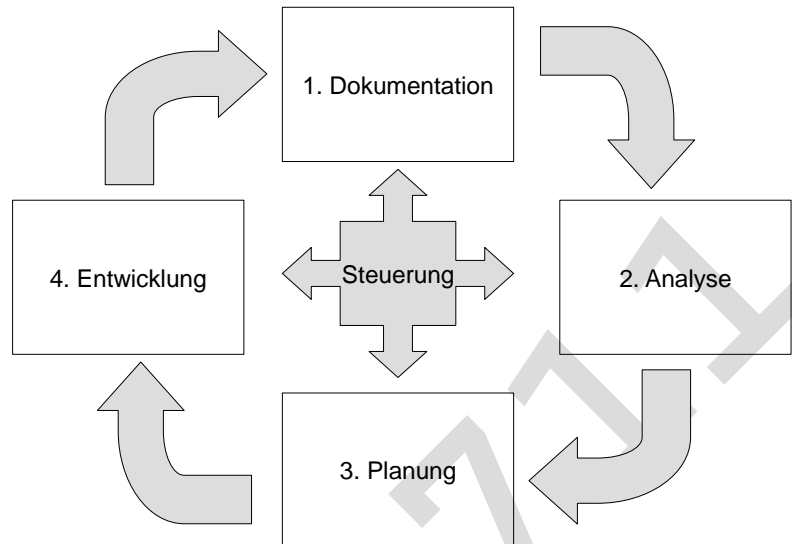


Abbildung 22: Architekturlebenszyklus

Quelle: Entnommen aus Niemann 2005, S. 3

4.2.1 Dokumentation

Am Anfang eines zielgerichteten Architekturmanagements muss die **Dokumentation der Ist-Situation** der Architektur stehen. Die Dokumentation soll u. a. folgende Kernfragen beantworten können (vgl. hier und im Folgenden Niemann 2005, S. 82 ff.):

Fragen im Rahmen
der Dokumentation

- Welche Prozesse werden durch die Architektur unterstützt?
- Wie wird diese Unterstützung betriebswirtschaftlich und technisch realisiert?
- Welche Schnittstellen und Kommunikationsbeziehungen existieren zwischen den Systemen?
- Gibt es Lücken, Brüche oder Redundanzen in der bestehenden Architektur?
- Welchen Zweck erfüllen die vorhandenen Systeme?
- Welchen Anforderungen müssen die Systeme gerecht werden?
- Welche Verfügbarkeit und Ausfallsicherheit haben die Systeme?
- Welche Daten werden von den Systemen verwendet?

Die **Ist-Situation** wird durch die Dokumentation transparent abgebildet. Die unterschiedlichen Interessengruppen (Rollen) gewinnen dadurch Überblick über die vorhandene Architektur. Dabei benötigen die verschiedenen **Rollen** in einem Unternehmen unterschiedliche Detaillierungsgrade. So benötigt z. B. ein Applikationsdesigner eine Dokumentation mit möglichst ausführlichen Details, da er zur Entwicklung einer neuen Applikation möglichst viele Einzelheiten kennen muss.

Der Projektleiter benötigt vergleichsweise weniger Einzelheiten als der Applikationsdesigner und der Unternehmensplaner braucht einen hohen Abstraktionsgrad, um einen möglichst guten Überblick über die Gesamtunternehmenssituation zu erhalten. Diese Abgrenzung wurde bereits von *Zachman* (1987) in seinem Framework for Enterprise-Architecture erkannt (vgl. Abschnitt 2.3.2.1). Er unterscheidet ebenfalls die Rollen Planer, Eigentümer, Designer, Entwickler und Zulieferer und begründet damit die verschiedenen Abstraktionsgrade und Themengebiete, die für die jeweiligen Rollen interessant sind.

Auch *Dern* (2006, S. 118 f.) unterscheidet im Rahmen des Architekturmanagement verschiedene Rollen. Diese zeigt er anhand der Sichten seiner Architekturpyramide (vgl. Abschnitt 2.3.2.3) auf. Tabelle 4 zeigt ausführlich die Rollen im Unternehmen, die sich mit Architekturen auseinander setzen. Sie verdeutlicht die verschiedenen Anforderungen an die Dokumentation der Architekturen, die sich im Architekturmanagementprozess ergeben.

Rolle	Rollenverständnis
Geschäftssicht	
Geschäfts-Architekt	Gestaltung der Businessarchitektur; Analyse und Bewertung des IS-Portfolios im Hinblick auf die Elemente der Businessarchitektur. Unterstützung von Softwareentwicklungsprojekten bei der betriebswirtschaftlichen Analyse und der Entwicklung betriebswirtschaftlicher Architekturen.
Architektursicht	
IT-Architekt (auf Unternehmensebene)	Gestaltung der Informationsarchitektur; Durchführung der übergreifenden Architekturplanung und Architekturentwicklungsplanung.
IT-Architekt (auf Projektebene)	Durchführung von Architekturentwicklungen; Mitarbeit in bzw. Beratung von Entwicklungsprojekten.
Infrastruktursicht	
Service-Manager	Abstimmung und Koordination von Anforderungen an die IT-Basisinfrastruktur und deren Umsetzung; Einbringen von Anforderungen der IT-Basisinfrastruktur in Architekturentwicklungen.
Sicherheits-Ingenieur	Unterstützung und Beratung der Entwicklung von Sicherheitsarchitekturen auf der Grundlage der Sicherheitsstrategie und bestehender bzw. geplanter Basisplattformen.
System-Ingenieur	Unterstützung und Beratung der Entwicklung von Systemarchitekturen auf Grundlage der Plattformstrategie und bestehender bzw. geplanter Basisplattformen.
Softwareentwicklungssicht	
Projektleiter	Planung und Steuerung von Softwareentwicklungsprojekten nach den Kriterien Inhalt, Budget und Zeit.
Software-Ingenieur	Design, Test und Realisierung von Softwarebausteinen von Informationssystemen auf Basis der zugehörigen IT-Architektur. Unterstützung bei der Entwicklung der Softwarearchitektur und der Definition des Entwicklungsprozesses.

Managementsicht	
IT-Controller	Bewertung und Steuerung von Aktivitäten der IT nach Wirksamkeit und Wirtschaftlichkeit
Prozess-Eigentümer	Verantwortung für Prozesse innerhalb des Unternehmens und deren Etablierung in der Aufbau- und Ablauforganisation.
IS-Eigentümer	Fachbereichsseitige Eigentümerschaft eines Informationssystems.
IS-Verantwortlicher	Zentraler Ansprechpartner für ein IS gegenüber dem Fachbereich (IS-Eigentümer). Service Level Management für das IS und Kanalisation sowie Abstimmung von Change Requests.
Sonstige	
IT-Forscher	Beobachtung, Analyse und Bewertung von Entwicklungen und Trends in der IT.

Tabelle 4: Rollen im Architekturmanagement

Quelle: Entnommen aus Dern 2006, S. 118 f.

Die Dokumentation liefert die Vorlage für eine wirksame Architekturanalyse und beinhaltet alle Schichten und Komponenten, die für den vertikalen und horizontalen Aufbau der Architektur definiert werden. Zusätzlich werden hierfür auch Konstruktionsprinzipien und Verwendungsrichtlinien erfasst. Die Dokumentation muss aufzeigen, nach welchen Prinzipien Schnittstellen und Kommunikationsbeziehungen in der Architektur aufgebaut sind und welche betriebswirtschaftlichen Einsatzszenarien vorherrschen. Zudem müssen Einsatzheuristiken, wie z. B. Mengengerüste, Infrastrukturnutzung, Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Ausfallsicherheit beinhaltet sein.

4.2.2 Analyse

Nach der Dokumentation der vorhandenen Architektur erfolgt ihre Analyse. Bei der Analyse wird die Architektur gedanklich in ihre Einzelteile zerlegt und logisch nach Prozessen und Systemen neu geordnet. Hierbei müssen alle **Abhängigkeiten**, **Abdeckungen** und **Schnittstellen** aufgedeckt werden, die die Applikationsarchitektur umfasst. Zudem muss die **Heterogenität**, **Komplexität** und **Konformität** geprüft werden sowie eine **Analyse** der in der Dokumentationsphase identifizierten **Kosten und des Nutzens** erfolgen.

In jeder Applikationsarchitektur bestehen **Abhängigkeiten** zwischen Applikationen oder Prozessen. Diese können sowohl intern gelagert, als auch extern gegeben sein, z. B. durch externe Datenbanken von Kunden, Lieferanten oder Datenanbietern. Hierbei muss genau analysiert werden, welche direkten und indirekten Verknüpfungen bestehen, und welche Auswirkungen eine Eliminierung oder Integration von Komponenten hat.

Im Weiteren wird die **Abdeckung** geprüft, welche die Applikationen für die Prozesse erreichen. Es soll daraus ersichtlich werden, ob Lücken oder Redundanzen vorhanden sind, die beseitigt werden müssen. Dabei wird genau analysiert, aus welchen Gründen diese Lücken oder Redundanzen entstanden sind und welche

Risiken und Auswirkungen sich dadurch für das laufende Geschäft ergeben.

Von besonderer Bedeutung im Rahmen der Applikationsarchitektur ist die Analyse der **Schnittstellen**. Alle Verbindungen zwischen den Applikationssystemen müssen auf Art, Anzahl, Komplexität, Häufigkeit, Aktualität, Leistung, Stabilität und Verfügbarkeit untersucht werden. Ziel ist es, Ineffizienzen und Optimierungspotenziale aufzudecken. Ein besonderer Fokus liegt hierbei auf der Schnittstellenreduktion. Zu prüfen ist, ob Schnittstellen durch z. B. Middleware oder Hub-and-Spoke-Architekturen eliminiert bzw. zusammengelegt werden können. Um die Art der Schnittstellen und ihre Anforderungen bzgl. Komplexität, Häufigkeit, Leistung, Stabilität und Verfügbarkeit zu bestimmen, ist es sinnvoll, ihre ursächliche Entstehung zu betrachten. Die Art der Geschäftsprozessunterstützung gibt Auskunft über die benötigten Anforderungen. Es ist bei der Optimierung der Applikationsarchitektur ebenfalls zu überlegen, ob eine Anpassung der Geschäftsprozesse wirtschaftlicher ist als die Anpassung der Applikationsarchitektur.

In diesem Rahmen ist ebenfalls eine Analyse der **Heterogenität** notwendig. In den Unternehmen wächst in der Regel über Jahre hinweg eine heterogene Applikationslandschaft, die sich aus den ständig wachsenden Anforderungen im Wettbewerb und den technischen Innovationen im IT-Bereich ergeben. In den seltensten Fällen arbeiten Unternehmen von Anfang an mit einem Architekturmanagement. Bei einer wirtschaftlichen Planung der Applikationsarchitektur würde deren Heterogenitätsgrad optimal ausfallen. Dabei ist der kleinste Heterogenitätsgrad nicht zwingend der optimale. Eine zu homogene Applikationslandschaft kann je nach Unternehmensstrategie suboptimal sein, z. B. bei hoch differenzierten Kundengruppen, die unterschiedliche Geschäftsprozesse bedingen. Sind diese Prozesse nicht effizient durch eine einzelne Applikation abzuwickeln, kann es erforderlich sein, mehrere Applikationen zu integrieren. Mit einem wirksamen Anforderungsmanagement (vgl. KE 1) kann das Applikationsportfolio hierbei optimal aufgestellt werden. Bei der Heterogenitätsanalyse soll im Ist-Portfolio geprüft werden, ob Potenziale zur Konsolidierung von Applikationen bestehen.

Aus der steigenden Heterogenität ergibt sich auch eine gesteigerte **Komplexität** der gesamten Applikationsarchitektur. Es bestehen derzeit noch keine Methoden zur Messung der Komplexität einer ganzen Applikationslandschaft. Allerdings erscheint es logisch, dass mit steigender Anzahl von Applikationen und Schnittstellen die Komplexität steigt. Durch Optimierung des Heterogenitätsgrades und der Schnittstellen kann demnach automatisch die Komplexität reduziert werden.

Die Analyse der **Konformität** stellt einen weiteren Aspekt dar. Die Applikationsarchitektur muss auf die vorherrschenden gesetzlichen Regelungen angepasst sein (z. B. im Rahmen von Basel II, Sarbanes Oxley Act oder Solvency II). Die Architektur muss einerseits so ausgerichtet sein, dass sie die schnelle Ausführung und Informationsbereitstellung der im Regelwerk geforderten Anforderungen erfüllt. Andererseits muss sie sich flexibel und kurzfristig an neue oder geänderte gesetzliche Vorschriften anpassen können. Diese Konformitätsprüfung ist im Rahmen der IT-Governance eingebettet.

Bei der **Kosten-Nutzen-Analyse** werden die Kosten der Architektur sowie der Nutzen, welcher sich durch ihre Umsetzung ergibt, erfasst und ausgewertet. Kosten entstehen bei der Beschaffung, dem Betrieb oder der Wartung einer Architektur. Hier sind sowohl materielle (z. B. Hardware, Software) als auch personelle Ressourcen gemeint. Die Kostenanalyse kann mittels üblicher Kostenrechnungsmethoden erhoben werden. Wichtig ist hierbei, den ermittelten Kosten den Nutzen gegenüber zu stellen. Applikationen unterstützen Geschäftsprozesse und Unternehmensziele. Hieraus generieren sie den Nutzen für das Unternehmen. Die Applikationen können auch einen Beitrag zur Wertschöpfung leisten, z. B. durch neue Vertriebswege im e-Business-Bereich. Mittels unterschiedlichster Methoden zur Messung des Ziel- oder Wertschöpfungsbeitrags können die Nutzenpotenziale der Applikationen operationalisiert werden. Dadurch kann neben der Effizienzsteigerung, die sich aus den vorher genannten Analysebereichen ergeben, ebenso eine Effektivitätssteigerung erlangt werden (vgl. Niemann 2005, S. 125 ff.).

In Tabelle 5 sind diese Analysebereiche nochmals zusammengefasst.

Analysebereich	Beschreibung des Analyseverfahrens	Typische Fragestellungen
Abhängigkeit	Selektion von direkt und indirekt verknüpften Elementen sowie Aufzeigen von Beziehungen und (Aus-) Wirkungen der Applikationsarchitektur.	Welche Auswirkungen auf andere Architekturelemente ergeben sich bei Eliminierung alter oder Integration neuer Komponenten?
Abdeckung	Untersuchung der Abdeckung der Geschäftsprozesse, die durch die Applikationen erreicht wird.	Welche Lücken oder Redundanzen liegen in der Applikationsarchitektur vor und welche Auswirkungen und Risiken ergeben sich dadurch für das Unternehmen?
Schnittstellen	Analyse der Applikationsschnittstellen bezüglich Art, Anzahl, Komplexität, Häufigkeit, Aktualität, Leistung, Stabilität und Verfügbarkeit.	Welche Brüche oder Heterogenitäten ergeben sich bei der Prozessunterstützung und welche Schnittstellen können dadurch eliminiert oder zusammengelegt werden?
Heterogenität	Erkennen von Heterogenitäten in der gewachsenen Applikationslandschaft und von Konsolidierungsmöglichkeiten der Applikationen.	Welche Applikationsfunktionen werden nicht mehr verwendet oder können von anderen übernommen werden und welche Prozesse werden redundant von mehreren Applikationen unterstützt?
Komplexität	Analyse der Anzahl von Applikationen und deren Schnittstellen	Wie viele Applikationssysteme mit wie vielen Schnittstellen existieren?
Konformität	Einhaltung von gesetzlichen Vorgaben und Identifizierung flexibler Anpassungsmöglichkeiten.	Werden gesetzliche Vorgaben und Normen eingehalten?
Kosten-Nutzen	Erfassung von Kosten und Bewertung des Nutzens der Applikationsarchitektur hinsichtlich der Zielunterstützung und des Wertschöpfungsbeitrages im Unternehmen.	Welche Kosten-Nutzen-Relation liegt bei der Prozessunterstützung durch die einzelnen Applikationen vor?

Tabelle 5: Bereiche für die Architekturanalyse am Beispiel der Applikationsarchitektur

Quelle: In Anlehnung an Niemann 2005, S. 128 ff.

4.2.3 Planung

Nachdem die Dokumentation der Ist-Architektur und deren ausführliche Analyse abgeschlossen sind, beginnt die Planung der Soll-Architektur. Im Rahmen der Planung werden die in der Analyse aufgedeckten Schwachstellen vorab bewertet. Dadurch soll festgestellt werden, welche Ziele und Maßnahmen zur Veränderung dringend erfolgen müssen, und welche Schwachstellen eine geringere Priorität besitzen. Diese **Bewertung** erfolgt im Rahmen der Steuerung des Architekturmanagements. Verschiedene Instrumente hierzu werden in Abschnitt 4.2.5 erläutert.

Im Rahmen der Planung werden Ziele und Maßnahmen festgelegt, die zur Erreichung der Soll-Architektur führen. Hierbei müssen vorhandene Architekturstandards und -prinzipien beachtet werden. Wird ein neues Architekturdiseign entwickelt, ist es auch erforderlich, neue Standards und Prinzipien zu formulieren.

Architekturstandards beschreiben die **Standardisierung der Komponenten, wie z. B. Daten, Prozesse, Schnittstellen und Applikationen** innerhalb der Architektur. Die Architekturstandards werden in der Regel schriftlich fixiert und dienen als Handlungsrahmen für die zukünftige Entwicklung. Diese können sowohl auf betriebswirtschaftlicher als auch auf technischer Ebene definiert werden. Architekturstandards auf **betriebswirtschaftlicher Ebene** sind z. B. geforderte Mitarbeiterfähigkeiten, wie spezielle Architekturkenntnisse oder Kommunikationskompetenz. Auf **technischer Ebene** lassen sich Architekturstandards, wie z. B. geforderte Serverleistung festlegen (vgl. Bohand und Yellin 2006, S. 166 f.).

Architekturstandards

Architekturprinzipien bilden ebenfalls einen Handlungsrahmen durch **definierte Regeln und Richtlinien**. Mit ihnen wird die Erarbeitung, Weiterentwicklung und Anwendung der Architektur festgelegt. Die Architekturprinzipien dienen der Transparenz für die Beteiligten und als Basis für Entscheidungsprozesse. Daher sollten sie immer in Abstimmung der Stakeholder definiert und regelmäßig aktualisiert und ergänzt werden. Veraltete Architekturprinzipien führen zu Entscheidungsblockaden bis hin zu Fehlentscheidungen im Rahmen der Architekturentwicklung (vgl. Innenministerium des Landes Nordrhein-Westfalen 2005, S. 8).

Architekturprinzipien

Tabelle 6 zeigt beispielhaft einen Auszug der Architekturprinzipien für den Geschäftsbereich des Innenministeriums NRW bei der Erstellung einer Applikationsarchitektur. Sie veranschaulicht, was durch festgelegte Architekturprinzipien geregelt werden kann. Grundsätzlich können die Prinzipien in folgende Kategorien unterteilt werden (vgl. auch Dern 2006, S. 271):

- Gestaltung von Architekturen
- Gestaltung und Management des Portfolios
- Planung und Steuerung von Architekturen
- Weiterentwicklung von Architekturen
- Technische Standards (Architekturstandards)
- Organisation (Strategie, Prozesse, Menschen, Kultur)

Architekturprinzip	Beschreibung
Vorrang der Leitsätze	Diese Leitsätze der Informationsverarbeitung gelten verbindlich für alle Behörden und Einrichtungen im Geschäftsbereich des Innenministeriums.
Wirtschaftlichkeit	Alle Entscheidungen im Bereich IT sind unter wirtschaftlichen Gesichtspunkten zu prüfen und werden so getroffen, dass der größtmögliche Nutzen für alle Behörden und Einrichtungen erzielt wird.
Beteiligung	Alle Behörden und Einrichtungen werden an der Entscheidungsfindung in Bezug auf die IT-Strategie beteiligt.
Verfügbarkeit und Zuverlässigkeit der IT	Verwaltungsabläufe dürfen nicht durch mangelhafte Verfügbarkeit der IT-Systeme behindert oder unterbrochen werden.
Mehrfachnutzung von Applikationen	Bei Beschaffung oder Entwicklung von Applikationen ist sicherzustellen, dass gleiche Applikationen für gleiche Anforderungen bei unterschiedlichen Einrichtungen zum Einsatz kommen.
IT-Verantwortlichkeit	Die IT-Organisation ist als Dienstleister für die IT-Landschaft verantwortlich mit dem Ziel, den vom Nutzer definierten Anforderungen an Funktionalität, Service Levels, Kosten und Zeit zu genügen.
Datenverwaltung	Den Nutzern ist der Zugang zu allen benötigten Daten zu ermöglichen. Daten sind konsistent und redundanzfrei zu verwalten.
Einheitliche Begriffs-, Prozess- und Datendefinition	Begriffe, Prozesse und Daten werden bei allen betroffenen Organisationseinheiten einheitlich definiert und beschrieben. Die Definitionen sind verständlich und allen Nutzern zugänglich.
Sicherheit	Daten, Verfahren und Systeme sind gegen einen unautorisierten Zugriff und gegen unautorisierte Verwendung zu schützen.
Unabhängigkeit	Applikationen werden herstellerunabhängig entwickelt und können daher auf unterschiedlichen technischen Plattformen eingesetzt werden.
Benutzerfreundlichkeit	Applikationen sind leicht zu bedienen und zu benutzen. Die darunterliegende Technologie hat keinen Einfluss auf das Verhalten der Applikationen.
Änderungen	Änderungen an Applikationen oder Technologie erfolgen nur aufgrund von neuen oder geänderten betriebswirtschaftlichen Anforderungen. Änderungen sind zeitgerecht zu implementieren.
Vorgehensmodell	Bei allen Entwicklungen und Änderungen von IT-Lösungen ist ein definiertes Vorgehensmodell zu nutzen.
Standards	Software und Hardware müssen den definierten Standards entsprechen, um die Interoperabilität der Daten, Applikationen und Technologie sicherzustellen.

Tabelle 6: Architekturprinzipien des Innenministeriums NRW

Quelle: Entnommen aus Innenministerium des Landes Nordrhein-Westfalen 2005, S. 8

Im Rahmen der Planung werden verschiedene Szenarien entwickelt und anschließend bewertet, um eine optimale Soll-Architektur zu identifizieren. Die Entwicklung unterschiedlicher Szenarien folgt dabei fünf Gestaltungskriterien (vgl. Krüger und Seelmann-Eggebert 2003, S. 33 ff.):

- **Funktionalität:** Die Einhaltung der betriebswirtschaftlichen Anforderungen muss gewährleistet sein.
- **Nachhaltigkeit:** Zukünftige Anforderungen müssen integrierbar sein.
- **Methodik:** Die Design- und Entwicklungsmethode für die Architektur muss sorgfältig und gewissenhaft ausgewählt werden.
- **Wiederverwendbarkeit:** Die Architekturkomponenten müssen so gewählt werden, dass ihre Korrektheit gewährleistet ist und sie auch zukünftig wiederverwendet werden können.
- **Standardisierung:** Definierte betriebswirtschaftliche und technische Architekturstandards müssen eingehalten werden.

Die **Soll-Architektur** dient der Optimierung der vorhandenen und besitzt einen Modellcharakter. Das Architekturmodell muss funktionale Eigenschaften erfüllen. Hiermit sind die Funktionalitätseigenschaften der Architekturkomponenten und deren Verknüpfungen gemeint. Zu ihnen gehören (vgl. hier und im Folgenden Müller et al. 2006, S. 195 ff.):

Funktionale
Eigenschaften

- **Eindeutigkeit:** Die Soll-Architektur muss Transparenz und ein allgemeines Verständnis im Unternehmen schaffen. Diese Eindeutigkeit kann nur durch eine entsprechend klare Semiotik, wie Syntax, Semantik oder Pragmatik erreicht werden.
- **Konsistenz:** Im Rahmen der Planung muss immer wieder die Konsistenz der Architekturkomponenten durchleuchtet werden. Dabei ist zu prüfen, ob weiterhin alle Geschäftsprozesse unterstützt sowie alle benötigten betriebswirtschaftlichen (z. B. Organisationseinheiten) und technischen Komponenten (z. B. Hardware, Software) berücksichtigt und integriert werden. Inkonsistenzen im Planungsmodell können schwerwiegende Fehler verursachen und müssen unbedingt im Vorfeld eliminiert werden.
- **Aktualität:** Wichtig bei dem Entwurf des Soll-Zustands ist außerdem die Aktualität. Wenn während des Planungsprozesses Änderungen an z. B. vorgenommen werden, müssen diese sofort dokumentiert werden. Hierzu empfiehlt sich ein automatisches Update aller Änderungen. Dokumentationen mit veralteten Informationen können ansonsten unbrauchbar werden.
- **Vollständigkeit:** Das Architekturmodell muss vollständig alle Funktionen und relevanten Daten abdecken. Besteht Unvollständigkeit bei der Umsetzung in die Soll-Architektur, kann das dazu führen, dass Akzeptanz- und Nutzungsprobleme bei den Anwendern aufgrund fehlender Informationen entstehen.
- **Verständlichkeit:** Eng verzahnt mit der Eindeutigkeit des Architekturmodells ist auch dessen Verständlichkeit. Dabei müssen die bereits in der Dokumentation erwähnten zielgruppenorientierten Abstraktionsstufen berücksichtigt werden. Nur ein gut verständliches Architekturmodell dient

dem Betrachter als Informationsbasis. Je nach Anwender muss der Detaillierungsgrad unterschiedlich ausfallen.

- **Erweiterbarkeit:** Ebenso ist eine Erweiterbarkeit der Soll-Architektur im dynamischen Unternehmensumfeld wichtig. Nur ein Modell, das sich flexibel anpassen lässt, ist langfristig erfolgversprechend. Aufgrund unvorhersehbarer zukünftiger Entwicklungen stellt die Modifikationsfähigkeit des Architekturmodells eine hohe Anforderung dar.
- **Auswertbarkeit:** Abschließend soll das Architekturmodell mittels Kennzahlen auswertbar sein. Das heißt, die enthaltenen Informationen müssen strukturierbar sein, um sie z. B. in Kennzahlen ausdrücken bzw. aggregieren zu können.

Nicht-funktionale Eigenschaften

Neben den funktionalen Anforderungen, soll die Soll-Architektur auch nicht-funktionale Eigenschaften erfüllen. Hier sind vor allem folgende zu nennen:

- **Akzeptanz:** Ein besonders wichtiger Aspekt, nicht nur im Rahmen des Architekturmodells, sondern im gesamten Architekturmanagementprozess, ist die Akzeptanz der Mitarbeiter für die Veränderungsprozesse, die angestoßen werden. Eine Reorganisation der Applikationsarchitektur bedeutet unter Umständen wesentliche Veränderungen in den Geschäftsprozessen oder in den eingesetzten Technologien. Die Anwender müssen die Veränderung akzeptieren, da ansonsten die Anpassungen ineffizient und wirkungslos sind.
- **Wirtschaftlichkeit:** Eng mit der Akzeptanz gekoppelt ist die Thematik der Wirtschaftlichkeit. Eine Architektur, die unwirtschaftlich ist und die Anwender nicht effizient bei ihren Aufgaben unterstützt, wird sowohl bei den Entscheidern als auch bei den Mitarbeitern wenig Akzeptanz finden. Der Wertbeitrag der Architektur ist schwierig ermittelbar und nur indirekt und mittel- bis langfristig zu identifizieren. Dennoch muss versucht werden, dem erheblichen Aufwand der Dokumentation, Analyse, Planung und Entwicklung einer Architektur einen Nutzen gegenüber zu stellen, indem z. B. wirtschaftlichere Prozesse als Kennzahl dargestellt werden.

Nachdem die umfassende Planung der Soll-Architektur abgeschlossen ist, ist die Soll-Architektur in der Entwicklungsphase zu realisieren.

4.2.4 Entwicklung

In dieser Phase wird die Soll-Architektur entwickelt. Die Entwicklung ist eng gekoppelt mit der Planung. Die Soll-Architektur muss für die Umsetzung operationalisiert werden. Hierzu dienen die unterschiedlichen Architekturmodelle, die einen Rahmen für den Umsetzungsprozess vorgeben.

Referenzarchitektur

Der Entwicklungsprozess kann außerdem durch den Einsatz von Referenzarchitekturen unterstützt werden. Referenzarchitekturen beschreiben abstrakte, generische Architekturen, die produkt- und unternehmensunabhängig anwendbar sind. Sie können sowohl Bottom-up, durch nachträgliche Abstraktion mehrerer Archi-

tekturkomponenten, als auch Top-down oder begleitend bei der Entwicklung der Architektur gestaltet werden. Referenzarchitekturen dienen der Einhaltung von Architekturstandards und der Wiederverwendbarkeit von Entwicklungskomponenten. Die betriebswirtschaftlichen und technischen Komponenten der Architektur werden dabei aufeinander abgestimmt. Dadurch wird eine höhere Effizienz bei der Architekturentwicklung erreicht und ein gemeinsames Architekturverständnis der einzelnen Stakeholder gefördert (vgl. Dern 2006, S. 51). Sind keine für die gewünschte Soll-Architektur passende Referenzarchitekturen vorhanden, so kann in dieser Phase zunächst eine solche Architektur entwickelt werden.

Die Entwicklung der Soll-Architektur geschieht dann in zwei großen Schritten: als erstes werden die notwendigen Architekturkomponenten entwickelt bzw. vorhandene Komponenten werden an die neuen Anforderungen angepasst. Abschließend erfolgt die Integration der Architekturkomponenten zu einer neuen Architektur.

4.2.5 Steuerung

Um ein erfolgreiches Architekturmanagement durchführen zu können, muss dieses gesteuert werden. Hierzu ist der Bedarf in den einzelnen Phasen des Architekturmanagementlebenszyklus bereits erkannt worden. Im Folgenden werden einige Instrumente aufgezeigt, die für die Steuerung des Architekturmanagements eingesetzt werden können.

Bei der Dokumentation und Analyse der Ist-Architektur wird die Ist-Situation erfasst und die Risiken und Schwachstellen der Architektur analysiert, um Gefahren- und Verbesserungspotenziale zu erkennen. Je nach Branche kann auch eine Wettbewerbsanalyse durchgeführt werden, um evtl. Wettbewerbsvorteile durch die Architektur der Mitbewerber zu identifizieren. Bei der Planung und Entwicklung werden im Rahmen eines Soll-Ist-Vergleichs Maßnahmen und Vorgehensweisen zur Erreichung der Soll-Architektur festgelegt. Alle Phasen müssen durch Erfolgskontrollen, z. B. mittels Kennzahlen, laufend überprüft werden. Hinsichtlich des Architekturmanagements hat sich auch der von *Kaplan und Norton* entwickelte Ansatz der Balanced Scorecard (BSC) etabliert. Die BSC wird hierbei auf die speziellen Anforderungen des Architekturmanagements angepasst (vgl. z. B. Krüger und Seelmann-Eggebert 2003, S. 245 ff.; Niemann 2005, S. 210 f.).

Besonders die Bewertung der Applikationsarchitektur erhält hier große Beachtung. Sowohl die Bewertung der dokumentierten Ist-Architektur in Bezug auf Risiken und Schwachstellen, als auch die Bewertung der unterschiedlichen Szenarien auf Wirksamkeit und Wirtschaftlichkeit sind ausschlaggebend für die erfolgreiche Repositionierung der Applikationsarchitektur. Hierbei bietet sich folgendes allgemeines Vorgehen zur Architekturbewertung an (vgl. hier und im Folgenden Posch et al. 2011, S. 175 ff.):

1) Festlegung der Bewertungskriterien

Zuerst müssen die relevanten Kriterien festgelegt werden, nach denen die Risiken und Schwachstellen der Architektur zu bewerten sind. Hierbei müssen alle relevanten Einflussfaktoren, die auf die Architektur einwirken können, beachtet wer-

den. Zu den Einflussfaktoren gehören organisatorische, technische und produktbezogene Faktoren. Organisatorische Einflussfaktoren können sich z. B. aus der Unternehmensstruktur, den Prozessen oder Menschen im Unternehmen ergeben. Die technischen Einflussfaktoren beziehen sich auf Hardware, Software und Standards. Produktbezogene Einflussfaktoren betreffen z. B. die funktionalen Anforderungen an die Applikationen, deren Schnittstellen, Systemleistung, Zuverlässigkeit oder Dienste (vgl. auch Krüger und Seelmann-Eggebert 2003, S. 314 ff.). Die Konzentration auf die wichtigsten Risikofaktoren und Schwachstellen ist ausschlaggebend, da die Bewertung ansonsten zu umfangreich und teuer wird.

2) Sicherstellung der Messbarkeit der Kriterien

Danach müssen die Risiken und Schwachstellen, die sich durch die kritischen Einflussfaktoren ergeben, sauber definiert und abgegrenzt werden. Es muss außerdem sichergestellt sein, dass die Faktoren messbar sind.

3) Festlegung der Bewertungsarten und Methoden

Nachdem die kritischen Faktoren identifiziert, präzisiert und analysiert wurden, kann die Bewertungsmethode festgelegt werden. Hierbei wird in der Regel zwischen quantitativen und qualitativen Architekturbewertungsmethoden unterschieden. Quantitative Methoden sind z. B. messende oder simulationsbasierte Methoden. Sie werden häufig zur Identifizierung von Qualitätseigenschaften angewendet. Zu den qualitativen Methoden zählen Fragebögen, szenariobasierte Bewertungen und Checklisten. Diese Methoden beruhen in der Regel auf Erfahrungswissen und werden in der Praxis häufig angewendet, da sie kostengünstiger als die quantitativen Methoden sind (vgl. Posch et al. 2011, S. 178 f.).

4) Durchführung der Bewertung und

5) Ableiten von Maßnahmen anhand der Ergebnisse

Im Folgenden werden **ausgewählte Bewertungsmethoden** für die unterschiedlichen Architekturmanagementphasen erläutert sowie die **Handlungsalternativen** aufgezeigt, die sich daraus ableiten lassen (vgl. im Folgenden Krüger und Seelmann-Eggebert 2003, S. 145 ff.).

Portfolio-Analyse

Die Bewertung der Ist-Architektur ist anhand der Portfolio-Analyse möglich, welche individuell auf die Anforderungen z. B. der Applikationsarchitektur modifiziert wird. Hierbei werden die Dimensionen nach speziellen Architekturkriterien aufgestellt. Ziel ist es, einen umfassenden Überblick des Ist-Applikationsportfolios zu erhalten und tendenzielle Handlungsbedarfe daraus ableiten zu können.

Marktanteils-/ Marktwachstums- Analyse

Die Anwendung der Vier-Felder-Matrix (Marktanteils-/Marktwachstums-Analyse) auf die Applikationsarchitektur kann nach folgenden Kriterien modifiziert werden. In der **Marktanteilsdimension** eignen sich zur Bewertung z. B. Kriterien wie der benötigte Ressourcenbedarf, das Budget oder die Kosten, die mit der Applikationsarchitektur verbunden sind. Die **Marktwachstumsdimension** bezieht sich auf den internen Nutzen der Applikationsarchitektur im Unternehmen. Wird die zukünftige Entwicklung und Ausrichtung des Unternehmens auf neue Märkte

durch die Applikationsarchitektur unterstützt, dann wächst auch der interne Markt für diese Applikation. Die Auswahl der Kriterien ist für jedes Unternehmen individuell zu gestalten, da je nach Strategie und Geschäftsprozessen unterschiedliche Kriterien von Bedeutung sind. Anhand der Einteilung in die vier Felder der BCG-Matrix können sehr anschaulich Schwachstellen sowie Entwicklungsmöglichkeiten der Applikationen abgebildet werden. Besonders die Konformität zur Unternehmensstrategie ist hierbei aufzeigbar. Jedoch gilt der allgemeine Nachteil der zu starken Verdichtung der Parameter in der BCG-Matrix, aufgrund der Einteilung in vier Schwerpunktfelder, auch im Hinblick auf die Bewertung der Applikationsarchitektur. Daher ist empfehlenswert, nachdem ein allgemeiner Überblick über die Ist-Situation erfasst wurde, eine anschließende Detailanalyse durchzuführen. Abbildung 23 zeigt eine mögliche Ausprägung eines Marktwachstums-/ Marktanteils-Portfolios für eine Applikationsarchitektur auf.

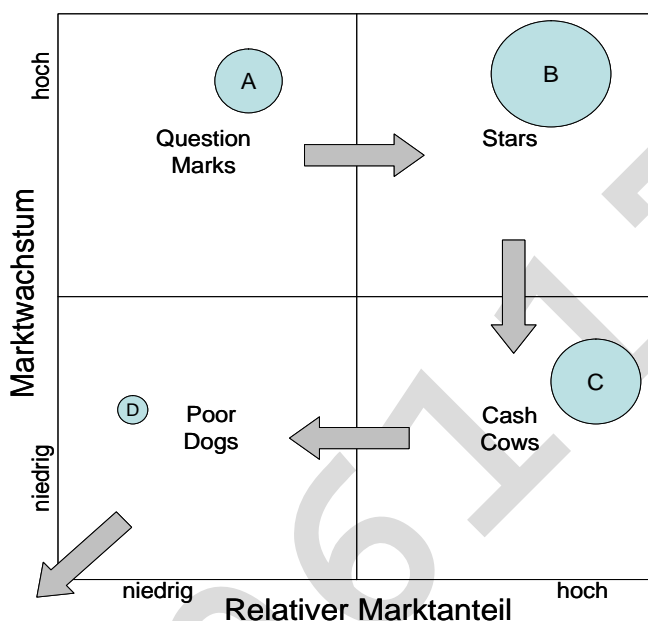


Abbildung 23: Marktanteils-/Marktwachstums-Portfolio einer Applikationsarchitektur

Quelle: Entnommen aus Krüger und Seelmann-Eggebert 2003, S. 147

In dem Portfolio sind vier Applikationen der betrachteten Architektur abgebildet. Es lässt sich leicht erkennen, dass z. B. Applikation D keine hohe Bedeutung mehr für das Unternehmen hat und deswegen eliminiert werden sollte. Hingegen weist Applikation A einen hohen erwarteten Nutzen auf. Hier sollte weiter investiert werden, damit die Applikation eine größtmögliche Unterstützungsfunktion im Unternehmen bieten kann.

Marktattraktivitäts-/ Wettbewerbsvorteils- analyse

Die Neun-Felder-Matrix von McKinsey (Marktattraktivitäts-/Wettbewerbsvorteilsanalyse) (vgl. Abbildung 24) integriert wesentlich mehr Kennzahlen, aufgrund qualitativer¹⁴ und quantitativer Faktoren. In der Dimension der **Marktattraktivität** fließen externe Faktoren, wie das Marktrisiko, das Marktwachstum oder die Nachfrage eines bestimmten Geschäftssegments ein. Je attraktiver dieses Segment für das Unternehmen ist, desto attraktiver sind auch die Applikationen zu bewerten, die diese Geschäftseinheit unterstützen. Auch hier ist immer die Konformität zur Unternehmensstrategie zu berücksichtigen. Zu den **relativen Wettbewerbsvorteilen** einer Applikation zählen Faktoren wie die betriebswirtschaftliche und technische Qualität oder die Güte von organisatorischen Arbeitsabläufen, welche direkt beeinflussbar sind. Sie drücken das Erfolgspotenzial der Applikationen für die zu unterstützenden Geschäftseinheiten aus. Der wesentliche Vorteil der Marktattraktivitäts-/Wettbewerbsvorteilsanalyse ist die Berücksichtigung qualitativer Faktoren. Die darin einfließende Subjektivität ist jedoch gleichzeitig als Nachteil zu bewerten. Grundsätzlich kann dieses Portfolio auch zur Gewinnung eines Überblicks über die Applikationsarchitektur dienen, welcher danach im Einzelnen noch genauer zu analysieren ist. Anhand der Einteilung der Neun-Felder-Matrix in drei Sektionen können strategische Entscheidungen abgeleitet werden. Die Architekturkomponenten mit der Bezeichnung A besitzen eine geringe Attraktivität sowie geringe Wettbewerbsvorteile für das Unternehmen. Hier sollten keine verstärkten Investitionen erfolgen, evtl. sogar Komponenten aus der Applikationsarchitektur eliminiert werden. In Sektion B sollte die Entscheidung anhand detaillierter Einzelanalysen der Komponenten erfolgen. Hier kann je nach Einzelfall eine Investitions-, Konsolidierungs- oder auch eine Eliminationsentscheidung sinnvoll sein. Die Architekturkomponenten in Sektion C haben dahingegen eine eindeutig hohe Bedeutung und Unterstützungsfunktion für das Unternehmen und sollten weiter ausgebaut werden.

¹⁴ Qualitative Kennzahlen (z. B. Mitarbeiter- oder Kundenzufriedenheit) basieren häufig auf einer Vielzahl von Einflussfaktoren, die multidimensional sind und daher selten isoliert betrachtet werden können. Demnach sind qualitative Kennzahlen sehr komplex und die Datenerhebung basiert auf qualitativen Methoden, wie z. B. Interviews oder Fragebögen.

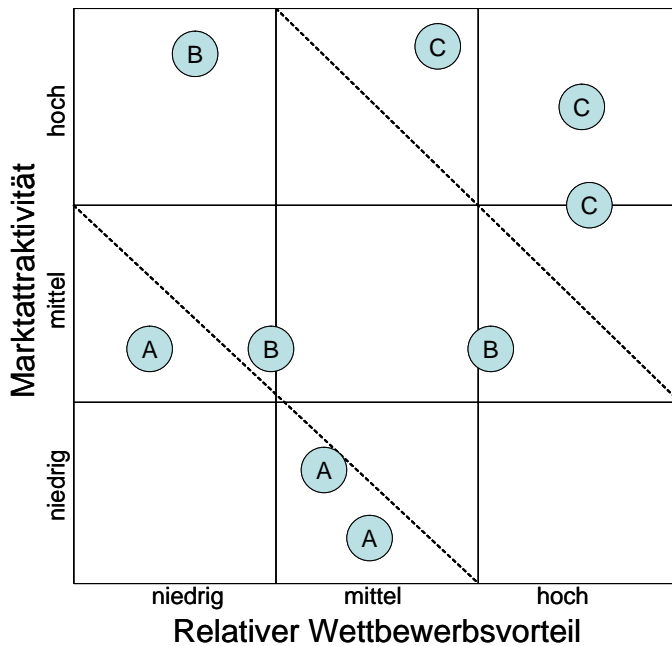


Abbildung 24: Marktattraktivitäts-/Wettbewerbsvorteilsportfolio einer Applikationsarchitektur

Quelle: Entnommen aus Krüger und Seelmann-Eggebert 2003, S. 147

Mit der Portfolio-Analyse kann also ein guter Überblick von der Ist-Architektur geschaffen werden. Mittels der gewonnenen Handlungsalternativen kann die Planungs- und Entwicklungsphase eingeleitet werden. Hierbei müssen die entwickelten Planszenarien bewertet werden. Das Software-Engineering bietet verschiedene Bewertungsmethoden¹⁵, die in diesen Kontext übertragen werden können. Im Folgenden werden zwei solche Bewertungsmethoden vorgestellt.

Die Architecture Trade-off Analysis Method (ATAM) ist eine Bewertungsmethode, die Qualitätsmerkmale in einer Applikationsarchitektur, wie z. B. die Zuverlässigkeit, die Wartbarkeit, die Verfügbarkeit, die Sicherheit, die Echtzeitfähigkeit, die Modifizierbarkeit, die Erweiterbarkeit, die Wirtschaftlichkeit und die Portierbarkeit der Applikationen, bewertet. Sie wurde von (Kazman et al. 1998, 2 ff.) entwickelt und baut auf der Scenario-based Architecture Analysis Method (SAAM)¹⁶ auf.

Architecture Trade-off Analysis Method (ATAM)

Bei der ATAM wird berücksichtigt, dass die Qualitätsmerkmale untereinander Abhängigkeiten besitzen können. Dieser gegenseitigen Beeinflussung sollte bei der Entwurfsentscheidung Rechnung getragen werden. Dazu werden die einzelnen Entwurfsentscheidungen, die auf die Qualitätsmerkmale Einfluss haben, nach Risiko und Nicht-Risiko sowie Sensitivity Points und Tradeoff Points kategorisiert.

¹⁵ Für einen Überblick verschiedener szenariobasierter Bewertungsmethoden siehe Grunske (2006, S. 286 ff.) und Grunske und Kaiser (2006, S. 295 ff.)

¹⁶ Zur detaillierten Beschreibung siehe Grunske (2006, S. 286 ff.)

Eine Architekturentscheidung ist mit Risiko behaftet, wenn ihre Auswirkungen auf die Qualitätsmerkmale und damit auf die Gesamtarchitektur nicht vorhersehbar sind. Sind die Konsequenzen jedoch im Vorfeld absehbar, kann die Entscheidung als Nicht-Risiko eingestuft werden. Architekturentscheidungen, die ausschlaggebend sind, um ein gewisses Qualitätsmerkmal zu erfüllen, werden als Sensitivity Point kategorisiert. Z. B. kann ein bestimmtes Sicherheitskonzept, wie spezielle Benutzerberechtigungen, ausschlaggebend für die Sicherheit des Gesamtsystems sein. Als Tradeoff Point werden Architekturentscheidungen kategorisiert, die sich auf mehrere Qualitätsmerkmale gleichzeitig auswirken und deshalb einen besonders kritischen Faktor darstellen.

Das Vorgehensmodell der ATAM umfasst folgende Phase:

ATAM

Vorgehensmodell:

1) Vorstellung der Bewertungsmethode ATAM, der Geschäftsziele, der Ist-Architektur und der geplanten Soll-Architekturen

In der ersten Phase muss im Architekturbewertungsteam ein einheitliches Verständnis geschaffen werden. Hierzu ist es wichtig, die ATAM, die relevanten Geschäftsziele sowie die Ist-Architektur und die geplanten Architekturszenarien vorzustellen.

2) Erstellen des Qualitätsattributbaums

In der zweiten Phase wird ein Qualitätsattributbaum erstellt, der an oberster Stelle ein Qualitätshauptmerkmal erfasst. Nachfolgend werden alle Qualitätsmerkmale aufgeschlüsselt, die das Obermerkmal beeinflussen. Darunter finden sich dann die Szenarien, die die jeweiligen Qualitätsmerkmale beeinflussen. Dadurch kann schnell erkannt werden, welche Szenarien sich auf welche Qualitätsmerkmale auswirken.

3) Analyse der Architekturentscheidungen

In der dritten Phase werden die Architekturentscheidungen aus den Soll-Szenarien analysiert, indem sie den Ergebnissen des Qualitätsattributbaums gegenüber gestellt werden. Hierauf erfolgt auch die Kategorisierung in Risiko und Nicht-Risiko bzw. in Sensitivity Points oder Tradeoff Points.

4) Dokumentation der Ergebnisse

In der vierten Phase müssen die Bewertungsergebnisse genauestens dokumentiert werden, um für spätere Projekte oder auch Projektschritte das gewonnene Wissen zu behalten. So können Änderungen nachverfolgt und mögliche, voneinander abhängige, Qualitätsanforderungen können bei zukünftiger Szenarienbildung berücksichtigt werden.

Vor- und Nachteile ATAM

Die Vorteile der ATAM liegen vor allem in der strukturierten Vorgehensweise, welche eine Wiederholbarkeit der Bewertung für zukünftige Szenarien vereinfacht. Zudem wird Wissen generiert, welches in eine zukünftige Architekturentwicklung einfließen kann. Die Betrachtung mehrerer, sich gegenseitig beeinflussender Qualitätsmerkmale und deren übersichtliche Darstellung in einem Qualitätsbaum sowie deren anschließende Kategorisierung (Risiko, Nicht-Risiko, Sensitivity Point, Tradeoff Point) erleichtern die Entscheidungsfindung für die richti-

ge Soll-Architektur. Nachteilig ist jedoch die Abhängigkeit der Bewertungsergebnisse von der Qualität des Bewertungsteams und der Szenarien. Demnach können, bei geringem Erfahrungswissen des Bewertungsteams, suboptimale Ergebnisse bei der Bewertung entstehen (vgl. Kazman et al. 1998, 2 f.).

Mit der ATAM wird somit die optimale Architektur anhand des größtmöglichen Nutzens ermittelt. In der Regel ist für Unternehmen jedoch nicht nur der Nutzen, sondern insgesamt die Wirtschaftlichkeit eines Vorhabens von Bedeutung. Demzufolge sind bei der Bewertung auch die Kosten zu berücksichtigen.

Aufbauend auf der ATAM kann eine Kosten-Nutzen-Analyse mit der Cost Benefit Analysis Method (CBAM) durchgeführt werden (vgl. hier und im Folgenden Kazman et al. 2001, 297 ff.). Die mithilfe der ATAM kategorisierten Architekturentscheidungen (Risiko, Nicht-Risiko, Sensitivity Point, Tradeoff Point) werden als Grundlage genommen. Dem durch die Stakeholder anhand verschiedener Qualitätsmerkmale (z. B. Zuverlässigkeit, Wartbarkeit, Verfügbarkeit, Sicherheit) ermittelten Nutzen werden dann die voraussichtlichen Kosten gegenüber gestellt. Somit lässt sich für jedes Architekturszenario die Wirtschaftlichkeit ermitteln. Abbildung 25 verdeutlicht die Zusammenhänge der CBAM.

Cost Benefit Analysis Method (CBAM)

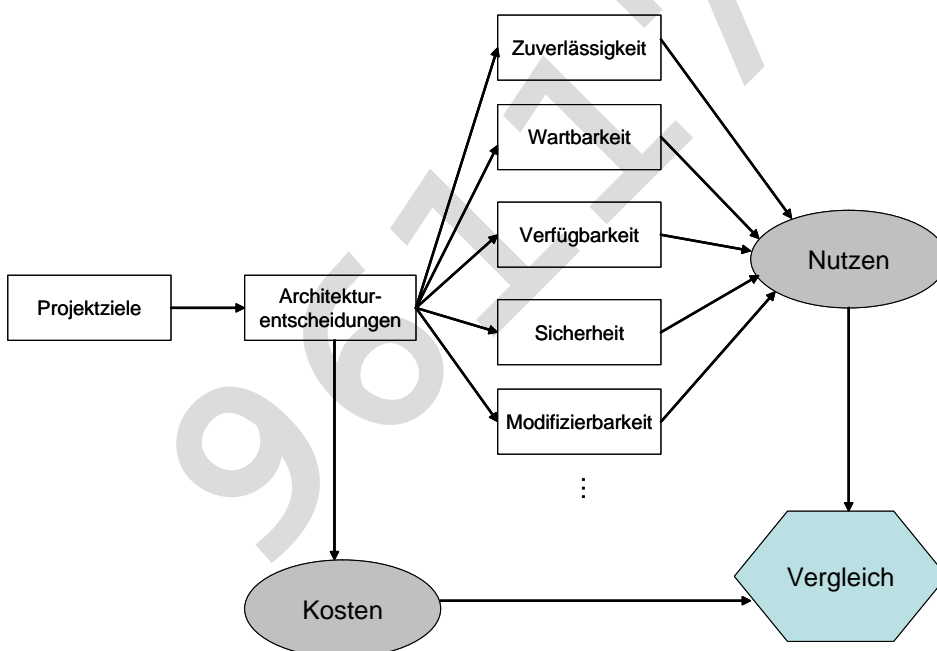


Abbildung 25: Kontext der CBAM

Quelle: Entnommen aus Kazman et al. 2001, S. 298

Die Ergebnisse der Kosten-Nutzen-Analyse können in ein Portfolio-Diagramm übertragen werden, um einen Überblick über die wirtschaftlichen Entwicklungsszenarien zu gewinnen. Die **Achsenabschnitte** werden mit Kosten und Nutzen beschriftet. Die Aufteilung kann in vier **Quadranten** erfolgen. Abbildung 26 zeigt ein Portfolio-Diagramm mit vier Quadranten. In Quadrant „I“ sind die Szenarien abgebildet mit einem hohen Nutzen und geringen Kosten. Quadrant „II“ enthält Szenarien mit ebenso hohem Nutzen, jedoch hohen Kosten. Die Quadranten „III“ und „IV“ weisen jeweils einen geringen Nutzen auf, wobei die Szenarien in Quad-

rant „III“ geringe Kosten aufweisen und die Szenarien im Quadrant „IV“ hohe. Dabei wird deutlich, dass die Szenarien, die in dem Quadranten „I“ mit einem hohen Nutzen und geringen Kosten platziert sind, am wirtschaftlichsten sind und somit umgesetzt werden sollten.

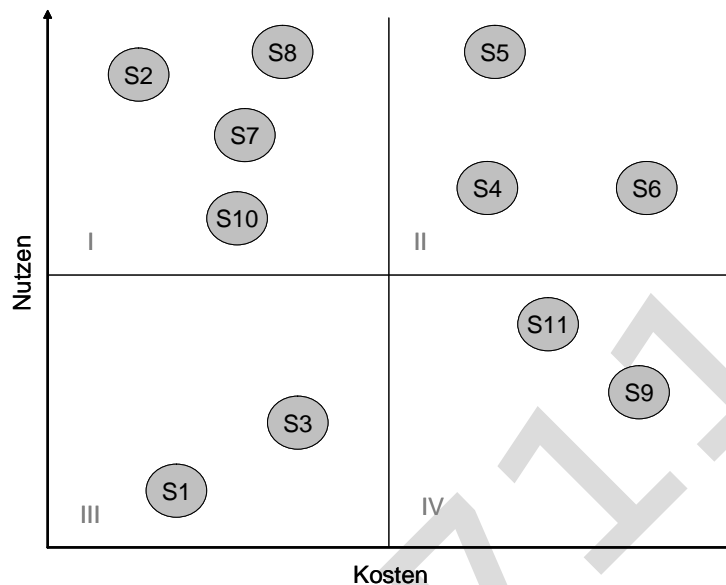


Abbildung 26: Kosten-Nutzen-Diagramm

Quelle: Entnommen aus Kazman et al. 2001, S. 301

Die bereits vorgestellten Bewertungsmethoden basieren hauptsächlich auf qualitativen Größen und dienen der Bewertung von Ist- und Soll-Architekturen sowie der Ableitung von Handlungsalternativen. Zur laufenden Steuerung einer Architektur eignen sich quantitative Bewertungsverfahren, z. B. mittels Kennzahlen.

Steuerung der Applikationsintegration

Schwinn und Winter (2005, S. 587 ff.) definieren mögliche Kennzahlen zur Steuerung der Applikationsintegration. Hierfür betrachten sie Erfolgsfaktoren und deren Wechselwirkungen, die im Rahmen der Steuerung der Integration in Applikationsarchitekturen relevant sind. Als Hauptziel definieren sie die Agilität einer Applikationsarchitektur. Dabei beschreibt die Agilität, wie schnell und flexibel die einzelnen Komponenten der Applikationsarchitektur, sowohl auf betriebswirtschaftlicher als auch auf technischer Ebene angepasst werden können. Auf die Agilität wirken unterschiedliche Erfolgsfaktoren ein, wie der optimale Integrationsgrad der Applikationen, die Komplexitätsreduktion der Applikationslandschaft, die optimale Wiederverwendung und Redundanzvermeidung, die minimalen Projektaufwände für die Applikationsintegration sowie die minimale Infrastrukturkomplexität und -kosten. Im Weiteren untersuchen *Schwinn und Winter* ausführlich die Interdependenzen zwischen den einzelnen Erfolgsfaktoren. Abbildung 27 stellt die Wechselwirkungen dieser Erfolgsfaktoren grafisch dar.

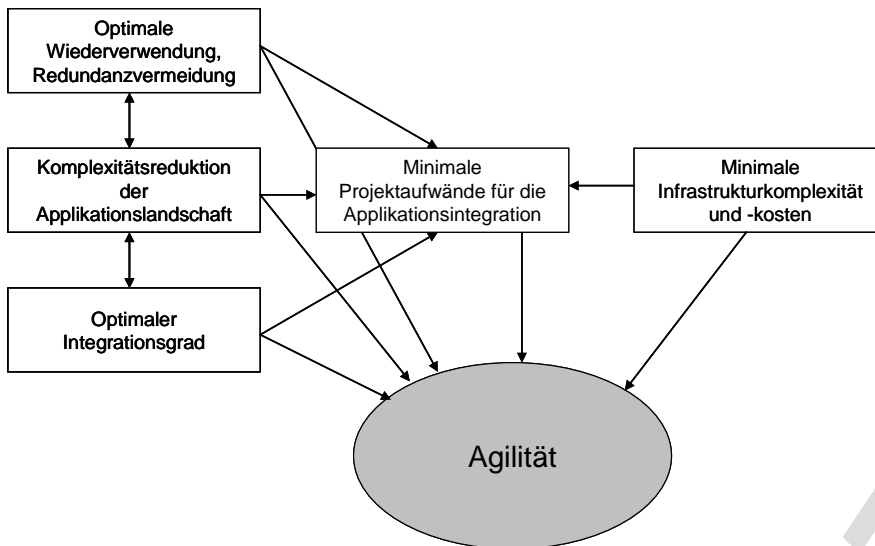


Abbildung 27: Ziele und Wechselwirkung der Applikationsintegration

Quelle: Entnommen aus Schwinn und Winter 2005, S. 593

Die Ermittlung geeigneter Kennzahlen für die Erfolgsfaktoren ist in der Regel schwierig. In Tabelle 7 sind einige mögliche Kennzahlen aufgeführt.

Zielgröße	Mögliche Kennzahl
Agilität	z. B. Agilitätsgrad = Kosten der Änderung, Erweiterung der Architektur für Zeitraum x/Größe durchgeführter Änderungen
Minimale Infrastrukturkomplexität und -kosten	z. B. Infrastrukturkosten, Anzahl eingesetzter Technologien, Grad des Einsatzes einheitlicher Produkte, Erfüllungsgrad der technischen Anforderungen in Projekten
Minimale Projektaufwände für die Applikationsintegration	z. B. Integrationskosten pro Jahr (Summe über alle Projekte)/Gesamt-Integrationskomplexität
Optimale Wiederverwendung, Redundanzvermeidung	z. B. durchschnittliche Wiederverwendung pro Funktion
Komplexitätsreduktion der Applikationslandschaft	z. B. Grad der Desintegration = Anzahl lose gekoppelter Verbindungen/Gesamtanzahl der Verbindungen der Applikationslandschaft
Optimale Enge der Kopplung	z. B. Kosten für Schnittstellenanpassung, Entwicklungs- und Laufzeitgemeinkosten für lose gekoppelte Systeme

Tabelle 7: Kennzahlen für die Applikationsintegration

Quelle: Entnommen aus Schwinn und Winter 2005, S. 593 ff.

Auch *Krüger* und *Seelmann-Eggebert* (2003, S. 154 ff.) definieren Kennzahlen zur Steuerung der Applikationsarchitektur. Sie sprechen in diesem Rahmen von Metriken und messen dabei die Leistung, den Umfang, die Komplexität sowie die Qualität der Applikationsarchitektur. Leistungsparameter betrachten die Unterstützungsfunktion der Applikationen für die Geschäftsprozesse. Je besser eine Applikation die Geschäftsprobleme löst, desto höher ist deren Leistung zu bewerten. Weiterhin kann die Komplexität der Applikationen variieren. Dies beeinflusst die Bedien- und Wartbarkeit der Applikationen sowie deren Kosten. Ebenso von

Bedeutung ist die durch die Anwender und Entwickler empfundene Qualität der Applikationen. Zur wirtschaftlichen Messung empfehlen die Autoren eine Strukturierung der Metriken nach Prozess, System, Ressourcen und dem Lebenszyklus (vgl. Tabelle 8).

Metrik	Mögliche Kennzahl
Prozessmetrik	z. B. Prozesslaufzeiten, Automatisierungsgrad, Anzahl beteiligter Systeme
Systemmetrik	z. B. Nutzungsgrad, Kosten-Nutzen-Verhältnis bei Erweiterungen, Integrationsfähigkeit, technische Qualität
Ressourcenmetrik	z. B. Produktivität der Mitarbeiter, Systemverfügbarkeit, Zuverlässigkeit
Lebenszyklusmetrik	z. B. Wartungsaufwände, Umsetzungszeiten, Aufwände des Änderungsmanagements, Überalterungszustand der Applikationsarchitektur

Tabelle 8: Strukturierte Metriken zur Applikationssteuerung

Quelle: Entnommen aus Krüger und Seelmann-Eggebert 2003, S. 157 ff.

Institutionalisierung
des Architektur-
managements

Abschließend ist anzuführen, dass zu einer zielgerichteten Steuerung der Applikationsarchitektur ebenso die Institutionalisierung des Architekturmanagements in das Unternehmen gehört. Es müssen Verantwortliche für die unterschiedlichen Aufgaben (z. B. Prüfung, Absicherung, Entscheidungsfindung) definiert werden. Hierzu eignen sich Gremien wie (vgl. Niemann 2005, S. 205 ff.):

- **Governance-Board:** Dieses besteht überwiegend aus IT-Führungskräften, wie z. B. dem Chief Information Officer (CIO) oder dem IT-Architekten. Hier werden die IT-Prozesse entschieden und das IT-Governance-Regelwerk definiert.
- **Architektur-Board:** Dieses besteht in der Regel aus Applikationsarchitekturspezialisten, IT-Architekten und Vertretern des Architekturmanagementbereichs. Es bildet eine Prüfungsinstanz zur Überwachung der im Governance-Board definierten Prozesse und Regelwerke für das Architekturmanagement.
- **Sounding-Board:** Dieses ist hauptsächlich für die Integration von Wissen, Anforderungen und Erfahrungen aus den unterschiedlichen IT-Bereichen in den Architekturmanagementprozess verantwortlich. Es lenkt die Kommunikation zwischen allen Projektbeteiligten. Das Sounding-Board muss einerseits Informationen aus der Projekt- und Linienpraxis in den Architekturmanagementprozess übertragen und andererseits alle Projektbeteiligten immer auf dem aktuellen Informationsstand halten.

In diesem Abschnitt wurde der Architekturmanagementprozess anhand des Architekturlebenszyklus aufgezeigt. Hierbei durchläuft das Architekturmanagement die Phasen der Dokumentation, der Analyse, der Planung und der Entwicklung. In allen Phasen ist die Steuerung für ein wirtschaftliches Architekturmanagement wichtig. Im folgenden Abschnitt werden ausgewählte Vorgehensmodelle zum Architekturmanagement vorgestellt.

4.3 Vorgehensmodelle zum Architekturmanagement

In der Literatur finden sich verschiedene Vorgehensmodelle zum Architekturmanagement. Sie alle folgen in gewisser Weise dem evolutionären Charakter des zuvor beschriebenen Architekturmanagementlebenszyklus. Im Folgenden werden vier Modelle beschrieben.

4.3.1 IT-Architektur-Engineering

Krüger und Seelmann-Eggebert (2003, S. 26 ff.) beschreiben in ihrem Ansatz des IT-Architektur-Engineerings die IT-Architektur als Gesamtheit aller Applikationen und Daten in einem Unternehmen inklusive der Organisationseinheiten bzw. Geschäftsprozesse, die durch sie unterstützt werden. Damit wird deutlich, dass die technische ebenso wie die betriebswirtschaftliche Sicht berücksichtigt wird. Die IT-Architektur wird als ein „Unternehmensökosystem“ betrachtet, wobei die Wertschöpfung die Grundmotivation und die Strategie den Treiber für Veränderungen darstellt.

Das Vorgehensmodell zum IT-Architektur-Engineering umfasst die Bereiche Dokumentation, Bewertung sowie Organisation und beschreibt im Wesentlichen die wirtschaftliche Transformation der Ist-Architektur in die geplante Soll-Architektur (vgl. Abbildung 28).

Vorgehensmodell

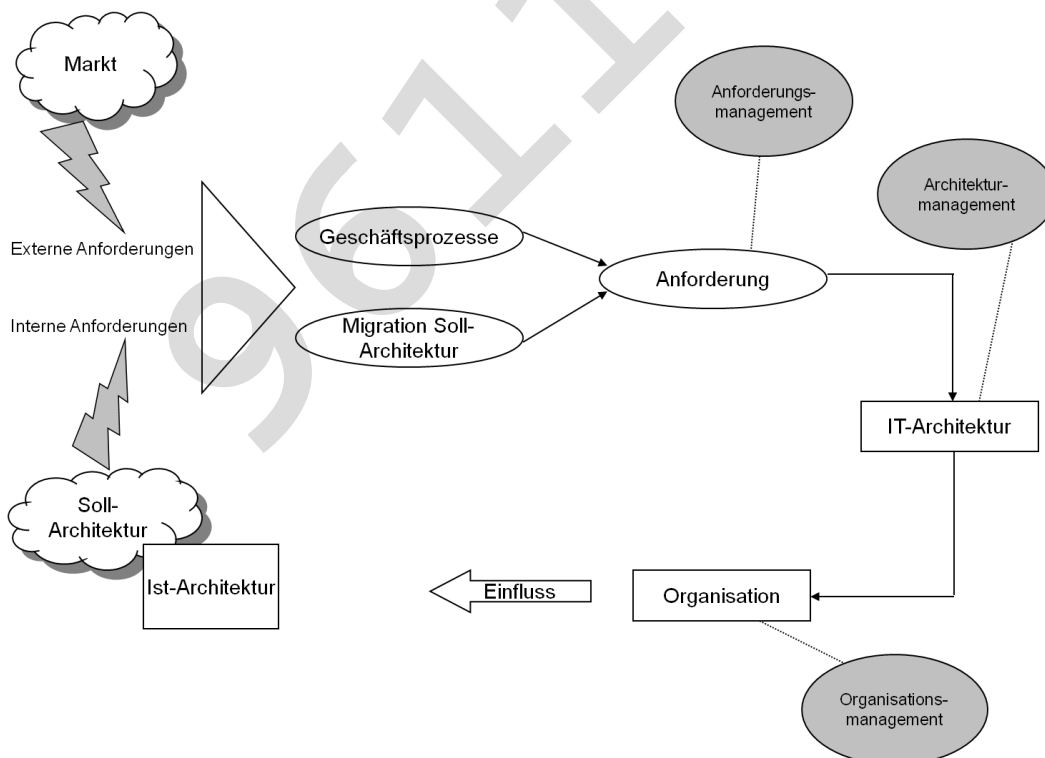


Abbildung 28: Veränderungskreislauf der IT-Architektur

Quelle: Entnommen aus Krüger und Seelmann-Eggebert 2003, S. 331

Der Architekturmanagementprozess ist auch hier eng mit dem Projektmanagement des Unternehmens gekoppelt. Der Anstoß wird durch **äußere Einflüsse** gegeben. Hier sind sowohl externe (Marktveränderungen, technische Innovationen)

als auch interne Anforderungen (z. B. Systemvorgaben, technische Einschränkungen, Bedürfnisse der Fachbereiche) zu nennen. Durch die **Geschäftsprozesse**, die sich daran anpassen müssen und der Repositionierungsplanung in Richtung **Soll-Architektur**, ergeben sich neue **Anforderungen** an die IT-Architektur, die im Rahmen des Anforderungsmanagements dokumentiert werden müssen. Nachfolgend muss die **IT-Architektur** bewertet werden. Hierzu dienen Situations-, Risiko-, Wettbewerbs- und Schwachpunktanalysen anhand derer die Konformität hinsichtlich des Gesamtgeschäfts geprüft werden. Danach kann die IT-Architektur im Rahmen des eigentlichen Architekturmanagements sowie die Ablauf- und Aufbauorganisation im Rahmen des Organisationsmanagements an die neuen Anforderungen angepasst werden. Das Organisationsmanagement beinhaltet Methoden zur Steuerung der Komplexität zwischen Strategie, Technologie und **Organisation**. Weiterhin werden Aufgaben definiert, die sich hinsichtlich des Transformationsprozesses an die IT-Architekten ergeben (vgl. Krüger und Seelmann-Eggebert 2003, S. 317 ff.).

Vor- und Nachteile

Der Ansatz beinhaltet sehr ausführlich die Dokumentation und stellt darüber hinaus auch Architekturstandards, -prinzipien, Prozesse und Rahmenwerke zur Verfügung. Ebenso positiv ist die Integration des Anforderungs- und Organisationsmanagements sowie die kontinuierliche Analyse der Einflussfaktoren zu beurteilen. Nachteilig ist allerdings die geringe Orientierung an der Applikationsarchitektur. Die Formulierung einer Vision wird nicht explizit mit einbezogen und dem evolutionären Charakter wird im Rahmen des Architekturmanagements nur gering gefolgt. Zudem bedingt der Ansatz nur eine grobe Darstellung des eigentlichen Architekturmanagements (vgl. Hafner 2005, 106 f.).

4.3.2 Management von IT-Architekturen

Die verschiedenen Ebenen der Architekturpyramide von *Dern* (2006, S. 6) wurden bereits in Abschnitt 2.3.2.3 beschrieben und werden an dieser Stelle nicht noch einmal erläutert. Das Management der IT-Architektur wird anhand der verschiedenen Ebenen der Architekturpyramide in Abbildung 29 verdeutlicht.

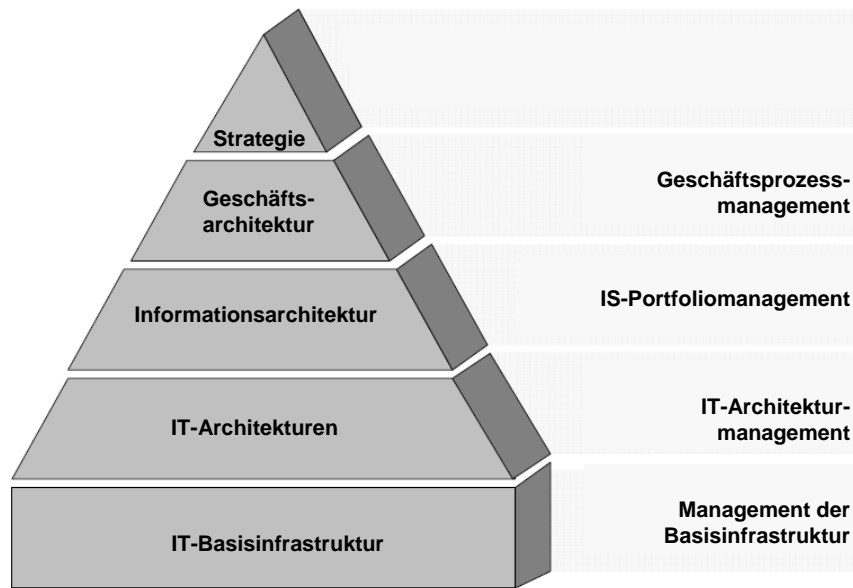


Abbildung 29: Modell für das Management von IT-Architekturen

Quelle: Entnommen aus Dern 2006, S. 6

In seinem Ansatz beschreibt der Autor mittels definierter Workflows¹⁷, wie das IS-Portfolio von der Ist- in die gewünschte Soll-Architektur überführt werden kann, und welche Anpassungsprozesse der Business- und IT-Basisinfrastruktur damit verbunden sind. Im Rahmen des IT-Architekturmanagements beschreibt *Dern* hauptsächlich die Phasen der Architekturplanung und -entwicklung.

Die Architekturplanung umfasst im Wesentlichen eine zyklische Anforderungserhebung und die anschließende Analyse des Ist-IS-Portfolios. Bevor über die Umsetzung in das Soll-IS-Portfolio entschieden werden kann, muss das aus den Anforderungen abgeleitete Soll-IS-Portfolio bewertet werden. Hieraus ergeben sich dann die Maßnahmen, die in die Architekturentwicklung einfließen. Die Architekturentwicklung beschreibt die eigentliche Weiterentwicklung der IT-Architektur. Hierbei wird ebenso der Applikationsentwicklungsprozess berücksichtigt (vgl. Dern 2006, S. 151 ff.).

Weiterhin betrachtet *Dern* (2006, S. 111 ff.) im Rahmen der Einführung des Architekturmanagements insbesondere den Reifegrad eines Unternehmens bzgl. des Architekturmanagements, die Erwartungen und Anforderungen der Stakeholder sowie konkrete Aufträge. Hierbei wird ein Balancedreieck aufgespannt (vgl. Abbildung 30), welches verdeutlichen soll, dass die einzelnen Elemente bei der Einführung ausgewogen beachtet werden müssen. Ebenso werden bei der Einführung verschiedene Sichten im Rahmen eines Rollenmodells betrachtet und Gremien für architekturelevante Entscheidungen empfohlen.

¹⁷ *Dern* definiert in seinem Ansatz sechs „Workflows“, die bei der Umsetzung der Architekturplanung und -entwicklung die Arbeitsabläufe im Rahmen des Applikationsentwicklungsprozesses abbilden.

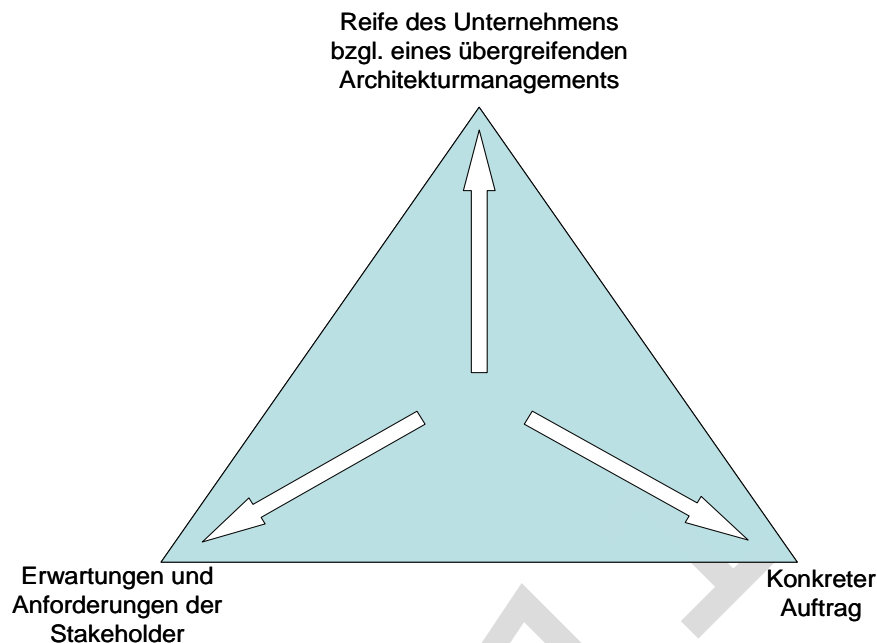


Abbildung 30: Balancedreieck

Quelle: Entnommen aus Dern 2006, S. 112

Vor- und Nachteile

Besonders positiv zu bewerten ist die besondere Berücksichtigung des Wertschöpfungsbeitrages der Applikationsarchitektur im Rahmen des Architekturmanagements. Die Workflows, die die Arbeitsabläufe für die Architekturplanung und auch die -entwicklung darstellen, werden sehr ausführlich beschrieben und die Verknüpfungen des Architekturmanagements mit dem Geschäftsprozess-, IS-Portfolio- sowie Basisinfrastrukturmanagements werden gut dargestellt. Die ausgewogene Betrachtung im Balancedreieck des Reifegrades eines Unternehmens hinsichtlich des Architekturmanagements und der Einbeziehung konkreter Aufträge mit Berücksichtigung der Stakeholder folgt dem gewünschten evolutionären Charakter des Architekturmanagements. Nachteilig sind allerdings die geringe Beachtung der Dokumentations- und Steuerungsphase (vgl. Hafner 2005, 103 f.).

4.3.3 The Open Group Architecture Framework - TOGAF

The Open Group Architecture Framework (TOGAF) stellt eine Entwurfsmethode zur Architekturentwicklung dar. Das Framework wurde von der Open Group erstmals 1995 entwickelt und hatte in den Versionen 1 bis 7 einen technischen Fokus. Mit der Version 8.1, die 2003 erschien, wurde auch die betriebswirtschaftliche Komponente von Architekturen mit berücksichtigt. Die Dokumentation zu TOGAF setzt sich aus vier Bestandteilen zusammen (vgl. hier und im Folgenden The Open Group 2003):

- **Introduction:** Im Rahmen der Einführung werden die Grundlagen zu Unternehmensarchitekturen und im Besonderen der Ansatz von TOGAF erläutert.
- **Architecture Development Method (ADM):** ADM bildet den Kern von TOGAF. Mit ihr wird schrittweise ein Entwicklungsansatz für Architektu-

ren beschrieben. Der Ansatz beschreibt ein sequenzielles Top-Down-Vorgehen und gliedert sich in folgende Phasen:

- **Preliminary Phase – Framework und Prinzipien:** Festlegung von Prinzipien, Methoden und Verantwortlichkeiten im Rahmen der Architekturentwicklung.
- **A – Architektur Vision:** Definition von Zielen, Treibern, Umfang, Fokus, Einflüssen und Vorgaben im Rahmen der Architekturentwicklung.
- **B – Geschäftsarchitektur:** Beschreibung der Ist-Geschäftsarchitektur und Entwicklung einer Soll-Geschäftsarchitektur. Anschließende Abweichungsanalyse zwischen Ist- und Soll-Geschäftsarchitektur.
- **C – IS-Architektur:** Beschreibung der Ist-Daten- und Applikationsarchitektur und Entwicklung einer Soll-Daten- und Applikationsarchitektur. Anschließende Abweichungsanalyse zwischen Ist- und Soll-Daten- und Applikationsarchitektur.
- **D – Technologiearchitektur:** Beschreibung der Ist-Technologiearchitektur und Entwicklung einer Soll-Technologiearchitektur. Anschließende Abweichungsanalyse zwischen Ist- und Soll-Technologiearchitektur.
- **E – Möglichkeiten und Lösungen:** Definition von Lösungsmöglichkeiten und Strategien für die Migration und den Implementierungsplan.
- **F – Migrationsplan:** Priorisierung von Projekten und Abschätzen der benötigten und zur Verfügung stehenden Ressourcen sowie der geschätzten Kosten und Risiken. Migrationsvorgehen definieren und dokumentieren.
- **G – Governanceimplementierung:** Formulierung von Projektvorschlägen und Dokumentation von Architekturvorhaben sowie die Überwachung von Projekt- und Architekturkompatibilität.
- **H – Architektur-Changemanagement:** Beobachtung der Geschäfts- und Systemänderungen sowie deren Bewertung für zukünftige Änderungsvorhaben.

Die Phasen A bis H werden durch ein kontinuierliches Anforderungsmanagement begleitet. Abbildung 31 stellt den Ablauf der ADM dar.

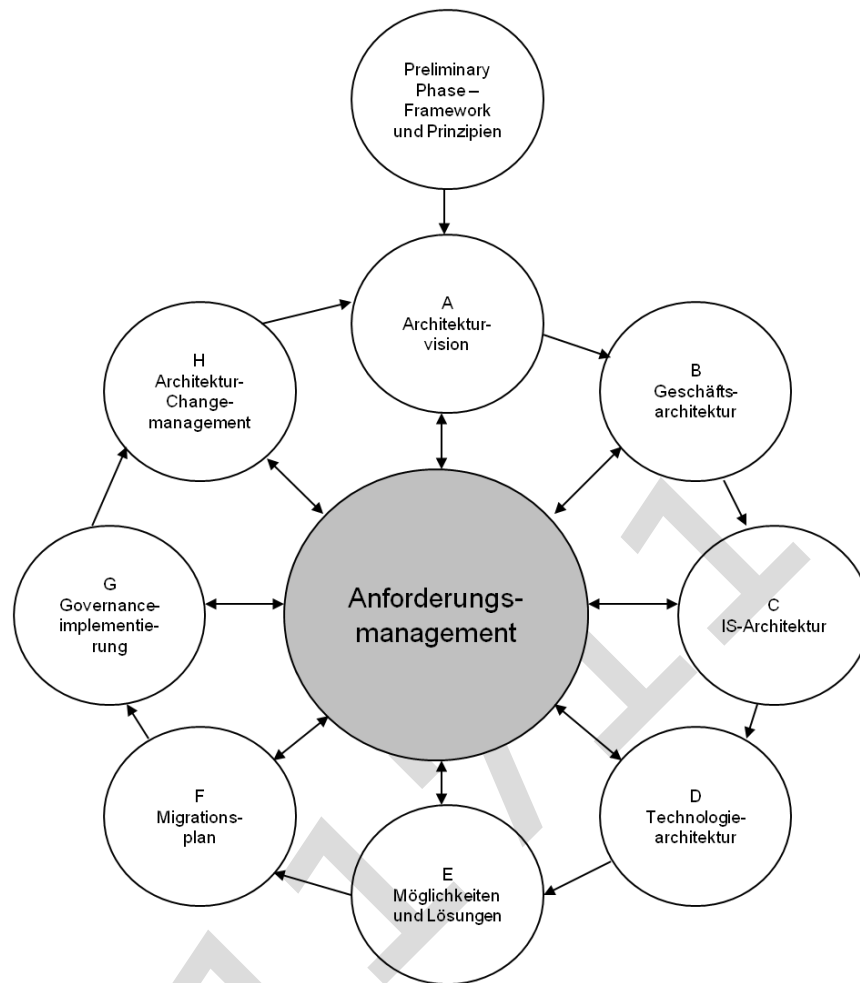


Abbildung 31: TOGAF Architecture Development Method

Quelle: Entnommen aus The Open Group 2003

- **Enterprise Continuum:** Hier werden Architekturmodelle klassifiziert (Architectural Continuum) sowie Lösungsmöglichkeiten der Implementierung aufgezeigt (Solutions Continuum). In diesem Rahmen stellt TOGAF mit der Foundation Architecture das Technical Reference Modell (TRM) sowie die Standard Information Base (SIB) zur Verfügung. Das TRM beinhaltet eine umfassende und übergreifende Serviceplattform mit z. B. Datenmanagement- oder Netzwerkservices. Die SIB ist eine Datenbank, die Standards, wie z. B. Produkt- oder Technologiestandards enthält.
- **Resource Base:** Mit der Resource Base werden Templates, Leitlinien und Hintergrundinformationen bereitgestellt, die im Rahmen des Entwicklungsprozesses z. B. die Strategie, die Prinzipien oder die Aufstellung von Projektteam und -leitung unterstützen.

Vor- und Nachteile

Das TOGAF stellt einen umfassenden Ansatz zur Architekturentwicklung dar und führt alle Phasen des Architekturmanagementprozesses ausführlich auf. Besonders hervorzuheben sind hierbei die Berücksichtigung der Skalierbarkeit, die Beachtung einer Architekturvision sowie die Bereitstellung von Dokumentations- und Vorgehensmodellen. Im Rahmen der Architektursteuerung fehlen allerdings geeignete Kennzahlen zur Bewertung. Der gewünschte evolutionäre Charakter des

Architekturmanagements ist durch den Ansatz ebenfalls eingeschränkt, da TOGAF ein sequenzielles Vorgehen und keinen Maßnahmenkatalog bei Abweichung vorschlägt (vgl. Hafner 2005, S. 99).

4.3.4 Architecture Capability Maturity Model - ACMM

Das Architecture Capability Maturity Model (ACMM) (vgl. hier und im Folgenden US DoC 2007) wurde vom *US Department of Commerce* (DoC) entwickelt, um die interne Bewertung von IT-Architekturprojekten und -entwicklungen besser unterstützen zu können. Ziel ist es, den Architekturmanagementprozess von der Dokumentation bis zur Entwicklung besser steuern zu können. Anhand von Schwachstellenanalysen sollen Vorgehen aufgezeigt werden, wie vorhandene Erfolgspotenziale besser ausgeschöpft werden können. Die Architektur muss mit Hilfe eines Änderungsmanagements kontinuierlich gewartet und weiterentwickelt werden. Hierbei soll der evolutionäre Charakter Beachtung finden. Zudem muss eine ständige Verbesserung der Fähigkeiten einer Organisation im Hinblick auf die Architekturprozessreife gewährleistet sein.

Das ACMM besteht aus drei Teilbereichen. Die ersten beiden Teile, das IT-Architekturreifegradmodell sowie die IT-Architektureigenschaften der einzelnen operativen Geschäftseinheiten in den verschiedenen Reifegraden, bilden eine Grundlage zur Beschreibung und Bewertung der Fähigkeiten der Organisation im Rahmen der Architekturreifegrade. Hier werden die einzelnen Reifegrade erläutert und die Fähigkeiten hinsichtlich der Architektureigenschaften aufgezeigt. Die gewonnenen Ergebnisse dienen als Messgrößen im Rahmen des Bewertungsprozesses. Der dritte Teil besteht aus der ACMM-Scorecard, mit der der Reifegrad der Architekturfähigkeiten der gesamten Organisation abgeleitet werden kann, der dann an den CIO berichtet wird.

Das ACMM unterscheidet sechs Architekturreifegrade einer Organisation:

Architekturreifegrade

- **Keine Architektur (No Architecture):** Keine Beschäftigung mit IT-Architektur.
- **Architektur ist im Anfangsstadium (Initial):** Es bestehen keine generellen Vorgänge, jedoch sind einige Prozesse bereits definiert. Nur ein kleines Projektteam ist involviert und die Architekturanstrengungen werden wenig kommuniziert. Es besteht kaum Verankerung mit der Geschäftsstrategie oder den Geschäftsprozessen.
- **Architektur ist in der Entwicklung (Under Development):** Es sind bereits eine Vision, Prinzipien, Standards und Ziele der Architektur definiert. Die Architekturstandards sind jedoch nicht zwingend mit den Zielen verknüpft. Ein technisches Referenzmodell ist bereits etabliert. Rollen und Verantwortlichkeiten im Architekturprozess wurden bestimmt. Es existiert ein Bewusstsein zur Architektursteuerung in der Organisation, jedoch bestehen nur geringe bis keine Steuerungsmechanismen sowie Investitions- und Akquisitionsstrategien.

- **Architektur ist festgelegt (Defined):** Die IT-Architektur enthält detailliert beschriebene Abläufe sowie ein umfassend entwickeltes technisches Referenzmodell. Die Soll/Ist-Analyse und ein Migrationsplan sind abgeschlossen. Es bestehen eine Finanzplanung sowie eine Investitionskontrolle und Steuerungsmechanismen hinsichtlich der IT-Architektur.
- **Architektur wird gesteuert (Managed):** Die IT-Architektur ist vollständig in der Organisation verankert und die Architekturprozesse sind Teil der Unternehmenskultur. Die Dokumentation der Architektur erfolgt in regelmäßigen Abständen. Im Rahmen der Architekturprozesssteuerung werden quantitative sowie qualitative Kennzahlen erhoben.
- **Architektur wird optimiert (Optimizing):** Die vorab erhobenen Kennzahlen werden zur kontinuierlichen Verbesserung der Architekturprozesse verwendet. Im Rahmen der systematischen Optimierung wird ein entsprechendes Änderungsmanagement etabliert. Architekturinvestitionen und -akquisitionen werden systematisch geplant und gesteuert.

Architekturmerkmale

Weiterhin definiert das ACMM neun Architekturmerkmale, anhand derer die Architektur bewertet wird und den einzelnen Reifegradstufen zugeordnet wird:

- IT-Architekturprozess
- IT-Architecturentwicklung
- Geschäftsverankerung
- Einbeziehung des Top-Managements
- Partizipation der operativen Geschäftseinheiten
- Kommunikation der Architekturprojekte
- IT-Sicherheit
- Architekturgovernance
- IT-Investitions- und -akquisitionsstrategie

Für jedes Architekturmerkmal beschreibt das ACMM die Ausprägung pro Reifegradstufe und bietet im Rahmen der ACMM-Scorecard eine Art Checkliste zur Bewertung an. Das ACMM bemisst hierbei den Reifegrad der gesamten Organisation. Dieser wird mittels der Architektureigenschaften in Verbindung mit den Reifegradstufen der Architektur erhoben. Anhand zwei verschiedener Methoden kann die Architekturfähigkeit der Gesamtorganisation bewertet werden.

Mit der ersten Methode wird das gewichtete Mittel berechnet, indem zuerst alle IT-Architektureigenschaften aufgelistet werden. Anschließend wird die Reifegradstufe pro IT-Architektureigenschaft bestimmt. Die vorkommenden Ereignisse pro Reifegrad werden abschließend summiert und durch neun (Anzahl der Architektureigenschaften) dividiert (vgl. Tabelle 9).

IT-Architektureigenschaft	Reifegradstufe
1	3
2	2
3	4
4	3
5	1
6	3
7	5
8	2
9	1
Gewichtetes Mittel	24/9 = 2,7

Tabelle 9: ACMM-Methode zur Berechnung des gewichteten Mittels

Quelle: Entnommen aus US DoC 2007, S. 15

Nach Tabelle 9 hat die gesamte Organisation bereits einen Reifegrad von 2,66 erreicht und befindet sich damit aufgerundet auf der Stufe 3 „Architektur ist festgelegt“.

Die zweite Methode berechnet den prozentualen Anteil pro Reifegradstufe (vgl. Tabelle 10). Auch an diesem Beispiel ist erkennbar, dass sich die Organisation bereits auf der dritten Reifegradstufe befindet, da sich die meisten Architekturmerkmale auf Reifegradstufe 3 befinden (33,3%).

Reifegradstufe	Anzahl pro Stufe	Prozentualer Anteil
0	0	0%
1	2	22,2%
2	2	22,2%
3	3	33,3%
4	1	11,1%
5	1	11,1%

Tabelle 10: ACMM-Methode zur Berechnung des prozentualen Anteils

Quelle: In Anlehnung an US DoC 2007, S. 16

Das ACMM bietet eine systematische Vorgehensweise, um die IT-Architektur zu pflegen und weiter zu entwickeln. Es dient der Qualitätssicherung der IT-Architektur, indem die Fähigkeit der Gesamtorganisation hinsichtlich des Managements der IT-Architektur kontinuierlich verbessert wird. Besonders positiv ist zu bewerten, dass das Architekturmanagement nicht allein im Architekturteam und dem Management verankert ist, sondern sich in alle operativen Geschäftsbereiche hineinzieht und die Unternehmenskultur maßgeblich mit beeinflussen soll. Zu kritisieren ist die geringe Beachtung von Integrationslösungen im Rahmen des Architekturmanagements. So gibt das ACMM keine Handlungsempfehlungen zur Komplexitätsreduktion von Architekturen, lediglich die Architekturreife der Or-

Vor- und Nachteile

ganisation steht im Fokus. Eine Kombination mit einem Architekturmanagementansatz, der die Architekturplanung stärker fokussiert und damit ein Anforderungsmanagement beinhaltet und zur Komplexitätsreduktion beiträgt, ist zu empfehlen.

Anhand der hier vorgestellten Ansätze zum Architekturmanagement ist erkennbar, wie unterschiedlich die verschiedenen Phasen des Architekturmanagementprozesses fokussiert werden. Ebenso unterschiedlich fällt der Detaillierungsgrad der vorgeschlagenen Vorgehensweisen aus. Daher ist die Anwendung eines Ansatzes jeweils von der individuellen Situation und des individuellen Bedarfes des Unternehmens abhängig.

Tabelle 11 fasst die Fokussierung der Ansätze hinsichtlich der einzelnen Architekturmanagementphasen noch einmal zusammen.

	Krüger/Seelmann-Eggebert	Dern	The Open Group	Department of Commerce
Dokumentation	X	-	X	X
Analyse	X	X	X	X
Planung	X	X	X	-
Entwicklung	-	X	X	X
Steuerung	X	-	-	X

Tabelle 11: Fokussierung der Architekturmanagementphasen

Legende: X : vorhanden

- : wenig bis nicht vorhanden

4.4 Übungsaufgaben

1. Was wird unter Architekturmanagement verstanden?
2. Nennen Sie fünf Ziele, die mit dem Architekturmanagement verfolgt werden.
3. Welche Aufgaben hat das Architekturmanagement?
4. Nennen und erklären Sie die Phasen des Architekturlebenszyklus.
5. Was wird unter dem Architecture Capability Maturity Model (ACMM) verstanden?

9611711

5 Zusammenfassung

In dieser KE wurden Grundlagen, Modelle und Methoden zu Architekturen und Integration vorgestellt. Dieses Themengebiet gliedert sich in die Aufgaben des Informatik Managements ein und stellt einen Teilbereich des Informationsmanagements dar.

Die Zielsetzung, die mit Architekturmanagement und Integrationsmaßnahmen verfolgt wird, ist eine Verringerung der Komplexität in Unternehmen und gleichzeitiger Erhaltung ihrer Flexibilität. Unternehmen sehen sich immer härteren Wettbewerbsbedingungen gegenübergestellt, die eine schnelle Anpassung an veränderte Kundenwünsche oder neue Technologien verlangt. Häufig können Unternehmen nur reagieren statt agieren, da ihnen die nötige strukturelle Übersicht sowie ein flexibler Handlungsrahmen fehlt. Architekturen entsprechen diesen Anforderungen und unterstützen im Rahmen umfassender Veränderungsprozesse.

In der vorliegenden KE wurde die Notwendigkeit von Architekturen und Integrationsmaßnahmen herausgearbeitet, die im Rahmen eines effektiven und effizienten Informationsmanagement entsteht. Hierfür stehen bereits verschiedene Modelle und Methoden zur Verfügung, die zum einen verdeutlichen, welche Komponenten beim Aufbau einer Architektur, im Speziellen der Applikationsarchitektur, betrachtet werden müssen und wie diese im Rahmen des Architekturlebenszyklus zu steuern sind. Zum anderen werden damit verschiedene Integrationslösungen dargestellt.

In Kapitel 2 wurden die Grundlagen zu Architekturen gelegt. Hierbei wurde deutlich, welches unterschiedliche Begriffsverständnis in der Literatur vorherrscht. Die Applikationsarchitektur wird aufgrund ihrer besonderen Bedeutung der Schnittstelle zwischen betriebswirtschaftlicher und technischer Ebene ausführlich erläutert. Anschließend folgte die Erläuterung der Ziele und Aufgaben von Architekturen und Architekturmodellen. Als bekannte Beispiele für Architekturmodelle wurden das Framework for Enterprise-Architecture von *Zachman*, das Ganzheitliche Modell der Informationssystem-Architektur von *Krcmar* und die Architekturpyramide von *Dern* aufgeführt.

Die Definition und Grundlagen zur Integration wurden in Kapitel 3 aufgezeigt. Hierbei wurden die verschiedenen Integrationsperspektiven und -ziele erläutert. Anhand der unterschiedlichen Integrationsziele und -perspektiven wurden die Komplexität des Themas und die sich daraus ergebenden Probleme verdeutlicht. Als Integrationslösung wurden Enterprise Application Integration und Serviceorientierte Architekturen vorgestellt.

Abschließend wurde in Kapitel 4 das Architekturmanagement behandelt. Anhand des Architekturlebenszyklus konnten die einzelnen Phasen der Dokumentation und Analyse sowie der Planung und Entwicklung erläutert werden. Diese Phasen müssen kontinuierlich im Sinne eines evolutionären Charakters durchlaufen, um die Wirksamkeit und Wirtschaftlichkeit der Architekturen im Unternehmen zu gewährleisten. Jede Phase wird durch Steuerungsprozesse begleitet.

In der Literatur sind unterschiedliche Vorgehensmodelle zum Architekturmanagement zu finden, von denen beispielhaft das IT-Architektur Engineering von *Krüger und Seelmann-Eggebert*, das Management von IT-Architekturen von *Dern*, TOGAF von der *Open Group* sowie das ACMM von *US Department of Commerce* beschrieben wurde. Auch hieran wurde deutlich, dass das Architekturmanagement, ebenso wie die Architektur selber, individuell gestaltbar ist und sich nach der Situation und den Anforderungen des Unternehmens richten muss.

9611711

Literaturverzeichnis

Aier, S.; Dogan, T. (2006): Nachhaltigkeit als Gestaltungsziel von Unternehmensarchitekturen. In: Aier, Stephan, Schönherr, M. (Hg.): Enterprise Application Integration. Serviceorientierung und nachhaltige Architekturen. 2. Aufl. Berlin: GITO-Verl. (Enterprise architecture), S. 75–122.

Aier, Stephan; Schönherr, M. (2006): Status quo geschäftsprozessorientierter Architekturintegration. In: WIRTSCHAFTSINFORMATIK, Bd. 48, S. 188–197.

Alt, Rainer; Cäsar, Marc; Leser, Florian; Österle, Hubert; Puschmann, Thomas; Reichmayr, Christian (2004): Architektur des Echtzeit-Unternehmens. In: Rainer Alt und Hubert Österle (Hg.): Real-time Business. Lösungen, Bausteine und Potenziale des Business Networking. Berlin: Springer, S. 19–52.

Alt, Rainer; Puschmann, Thomas; Reichmayr, Christian (2002): Strategien zum Business Networking. In: Österle et al. (Hg.): Business Networking in der Praxis. Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten: Springer, Berlin, S. 77–101.

Baumöl, Ulrike; Winter, Robert (2003): Qualifikation für die Veränderung. In: H. Österle und R. Winter (Hg.): Business Engineering. Auf dem Weg zum Unternehmen des Informationszeitalters. 2. Aufl.: Springer Berlin Heidelberg, S. 45–61.

Bohand, W.; Yellin, D. (2006): Using Enterprise Architecture Standards in Managing Information Technology. In: Journal of Management Information Systems, Vol. 23, S. 163–207.

Brenner, Walter; Zarnekow, Rüdiger; Pörtig, Fritz (2003): Entwicklungstendenzen im Informationsmanagement. In: H. Österle und R. Winter (Hg.): Business Engineering. Auf dem Weg zum Unternehmen des Informationszeitalters. 2. Aufl.: Springer Berlin Heidelberg, S. 147–168.

Conrad, Stefan (2006): Enterprise Application Integration. Grundlagen, Konzepte, Entwurfsmuster, Praxisbeispiele. 1. Aufl. Heidelberg: Spektrum Akademischer Verlag.

Dern, Gernot (2006): Management von IT-Architekturen. 2. Aufl. Wiesbaden: Springer Fachmedien.

Ferstl, Otto K.; Sinz, Elmar J. (2013): Grundlagen der Wirtschaftsinformatik. 7. Aufl. München: De Gruyter Oldenbourg.

Fleisch, Elgar (2001): Das Netzwerkunternehmen. Strategien und Prozesse zur Steigerung der Wettbewerbsfähigkeit in der "Networked economy": Springer Berlin Heidelberg (Business Engineering).

Frank, Ulrich (1994): Multiperspektivische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. München, Wien: Oldenbourg.

Grünauer, Karl Maria; Fleisch, Elgar; Österle, Hubert (2002): Systeme für das Supply Chain Management. Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten. In: Hubert Österle, Elgar Fleisch und Rainer Alt (Hg.): Business Networking in der Praxis. Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten, Bd. 3. Berlin, Heidelberg: Springer (Business Engineering), S. 185–206.

Grunske, L. (2006): Bewertungstechniken - eine allgemeine Übersicht. In: R. Reussner und W. Hasselbring (Hg.): Handbuch der Software-Architektur. 1. Aufl. Heidelberg: dpunkt, S. 277–293.

Grunske, L.; Kaiser, B. (2006): Bewertungstechniken für die Systemsicherheit. In: R. Reussner und W. Hasselbring (Hg.): Handbuch der Software-Architektur. 1. Aufl. Heidelberg: dpunkt, S. 295–310.

Hafner, Martin (2005): Entwicklung einer Methode für das Management der Informationssystemarchitektur im Unternehmen. Dissertation. Universität St. Gallen.

Hafner, Martin (2006): Entwicklung eines Zielsystems für ein systemisch-evolutionäres Management der IS-Architektur im Unternehmen. In: Joachim Schelp und Robert Winter (Hg.): Integrationsmanagement. Planung, Bewertung und Steuerung von Applikationslandschaften. 1. Aufl.: Springer Berlin Heidelberg (Business Engineering), S. 61–97.

Hafner, Martin; Schelp, Joachim; Winter, Robert (2006): Berücksichtigung des Architekturmanagements in serviceorientierten IT-Managementkonzepten am Beispiel von ITIL. In: Joachim Schelp und Robert Winter (Hg.): Integrationsmanagement. Planung, Bewertung und Steuerung von Applikationslandschaften. 1. Aufl.: Springer Berlin Heidelberg (Business Engineering), S. 99–121.

Hafner, Martin; Winter, Robert (2005): Vorgehensmodell für das Management der unternehmensweiten Applikationsarchitektur. In: Otto K. Ferstl, Elmar J. Sinz, Sven Eckert und Tilman Isselhorst (Hg.): Wirtschaftsinformatik 2005. EEconomy, eGovernment, eSociety: Physica-Verlag Heidelberg, S. 627–646.

Heine, Peter (1999): Unternehmensweite Datenintegration. Modular-integrierte Datenlogistik in betrieblichen Informationssystemen. Wiesbaden: Vieweg+Teubner Verlag (Teubner-Reihe Wirtschaftsinformatik).

Heinrich, Bernd; Leist, Susanne (2000): Bankenarchitekturen im Informationszeitalter — Zur Rolle des Geschäftsmodells. In: Hubert Österle und Robert Winter (Hg.): Business Engineering. Auf dem Weg zum Unternehmen des Informationszeitalters. Berlin, Heidelberg: Springer (Business Engineering), S. 141–165.

Heinrich, Lutz Jürgen; Riedl, René; Stelzer, Dirk (2014): Informationsmanagement. Grundlagen, Aufgaben, Methoden. 11. Aufl. Berlin: De Gruyter.

Heutschi, R.; Schemm, J. W. (2005): Fallstudie: Serviceorientierte Architektur bei Deutsche Post Brief. Universität St. Gallen. Institut für Wirtschaftsinformatik.

- Heutschi, Roger (2007): Serviceorientierte Architektur. Architekturprinzipien und Umsetzung in die Praxis: Springer Berlin Heidelberg (Business Engineering).
- Heutschi, Roger; Legner, Christine; Österle, Hubert (2006): Serviceorientierte Architekturen. Vom Konzept zum Einsatz in der Praxis. In: Joachim Schelp, Robert Winter, Frank Ulrich, Bodo Rieger und Klaus Turowski (Hg.): DW2006 - Integration, Informationslogistik und Architektur. P-90. Bonn: Köllen Druck+Verlag GmbH (Lecture Notes in Informatics (LNI)), S. 361–382.
- Hofmann, Jürgen; Schmidt, Werner (2010): Masterkurs IT-Management. Grundlagen, Umsetzung und erfolgreiche Praxis für Studenten und Praktiker. 2. Aufl. Wiesbaden: Vieweg + Teubner.
- Huber, T.; Alt, Rainer; Barak, Vladimir; Österle, Hubert (2002): Entwurf einer Applikationsarchitektur für die Pharmaindustrie. In: Hubert Österle, Elgar Fleisch und Rainer Alt (Hg.): Business Networking in der Praxis. Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten. Berlin, Heidelberg: Springer (Business Engineering), S. 165–183.
- Innenministerium des Landes Nordrhein-Westfalen (Hg.) (2005): IT-Architektur. Eine gemeinsame Initiative der Bezirksregierungen und des Innenministeriums des Landes NRW zur Erstellung einer Leitlinie für die Entwicklung und den Betrieb von IT-Systemen. Juli. Düsseldorf. Online verfügbar unter http://www.mik.nrw.de/fileadmin/user_upload/editors/import/imshop/shopdocs/it_architektur.pdf, zuletzt geprüft am 27.07.2016.
- Jung, Reinhard (2006): Architekturen zur Datenintegration. Gestaltungsempfehlungen auf der Basis fachkonzeptueller Anforderungen. Wiesbaden: GWV.
- Kaib, Michael (2002): Enterprise Application Integration. Grundlagen, Integrationsprodukte, Anwendungsbeispiele. 1. Aufl. Wiesbaden: Dt. Univ.-Verl.
- Kazman, R.; Asundi, Jai; Klein, M. (2001): Quantifying the Costs and Benefits of Architectural Decisions. In: 23rd International Conference on Software Engineering. ICSE 2001. Toronto, Ont., Canada, 12-19 May 2001. IEEE Computer Society, S. 297–306.
- Kazman, R.; Klein, M.; Barbacci, M.; Longstaff, T.; Lipson, H.; Carriere, J. (1998): The Architecture Tradeoff Analysis Method. In: Fourth IEEE International Conference on Engineering of Complex Computer Systems. ICECCS '98. Monterey, CA, USA, 10-14 Aug. 1998, S. 68–78.
- Krcmar, H. (1990): Bedeutung und Ziele von Informationssystem-Architekturen. In: WIRTSCHAFTSINFORMATIK, Bd. 32, S. 395–402.
- Krcmar, Helmut (2015): Informationsmanagement. Strukturierte und umfassende Darstellung des Informationsmanagement. 6. Aufl. Berlin Heidelberg: Springer.
- Krüger, Sascha; Seelmann-Eggebert, Jörg (2003): IT-Architektur-Engineering. Systemkomplexität bewältigen, Kosten senken, Potenziale freisetzen. 1. Aufl. Bonn: Galileo Press (Galileo Computing).

- Leist, Susanne (2002): Bankenarchitektur des Informationszeitalters — Zielsetzung und Gestaltungsebenen. In: Susanne Leist und Robert Winter (Hg.): Retail Banking im Informationszeitalter. Integrierte Gestaltung der Geschäfts-, Prozess- und Applikationsebene. Berlin: Springer (Business Engineering), S. 3–28.
- Linthicum, David S. (1999): Enterprise Application Integration. 1. Aufl. Amsterdam: Addison-Wesley (Addison-Wesley information technology series).
- Masak, Dieter (2006): IT-Alignment. IT-Architektur und Organisation: Springer Berlin Heidelberg.
- Mertens, Peter (2009): Integrierte Informationsverarbeitung 1. Operative Systeme in der Industrie. 17. Aufl. Wiesbaden: Gabler.
- Mertens, Peter; Bodendorf, Freimut; König, Wolfgang; Picot, Arnold; Schumann, Matthias; Hess, Thomas (2012): Grundzüge der Wirtschaftsinformatik. 11. Aufl. Berlin, Heidelberg: Springer.
- Müller, S.; Kuhn, Walter; Meiler, Christian; Petrov, Ilia; Jablonski, Stefan (2006): Integratives IT-Architekturmanagement. In: R. Reussner und W. Hasselbring (Hg.): Handbuch der Software-Architektur. 1. Aufl. Heidelberg: dpunkt.
- Niemann, Klaus D. (2005): Von der Unternehmensarchitektur zur IT-Governance. Bausteine für ein wirksames IT-Management. 1. Aufl. Wiesbaden: Vieweg.
- Niemann, Klaus D. (2006): From Enterprise Architecture to IT Governance. Elements of Effective IT Management. 1. Aufl. Wiesbaden: Vieweg.
- Picot, Arnold; Reichwald, Ralf; Wigand, Rolf T. (2003): Die grenzenlose Unternehmung. Information, Organisation und Management. Lehrbuch zur Unternehmensführung im Informationszeitalter. 5. Aufl. Wiesbaden: Gabler.
- Posch, Torsten; Birken, Klaus; Gerdorf, Michael (2011): Basiswissen Softwarearchitektur. Verstehen, entwerfen, wiederverwenden. 3. Aufl. Heidelberg: dpunkt.
- Puschmann, Thomas (2003): Collaboration Portale. Architektur, Integration, Umsetzung und Beispiele. Dissertation. St. Gallen.
- Puschmann, Thomas; Alt, Rainer (2004): Enterprise application integration systems and architecture – the case of the Robert Bosch Group. In: *Journal of Enterprise Information Management* 17 (2), S. 105–116.
- Röder, C. (2007): SOA und Qualität. In: Starke und Tilkov (Hg.): SOA - Expertenwissen. Methoden, Konzepte und Praxis serviceorientierter Architekturen. Heidelberg: dpunkt, S. 241 - 262.
- Rosemann, M. (1999): Gegenstand und Aufgaben des Integrationsmanagements. Arbeitsbericht Nr. 65. Universität Münster. Institut für Wirtschaftsinformatik.
- Scheer, August-Wilhelm (1998): Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse. 2. Aufl. Berlin: Springer.
- Schelp, J.; Hafner, Martin; Winter, Robert (2004): Architekturmanagement als Basis effizienter und effektiver Produktion von IT-Services. In: *HMD - Praxis der*

Wirtschaftsinformatik 237 (41), S. 54–66.

Schmidt, Christian (2009): Management komplexer IT-Architekturen. Empirische Analyse am Beispiel der internationalen Finanzindustrie. Technische Univ. Diss.-- Darmstadt, 2009. Wiesbaden: Gabler Verlag / GWV Fachverlage GmbH Wiesbaden.

Schwinn, Alexander; Winter, Robert (2005): Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration. In: Otto K. Ferstl, Elmar J. Sinz, Sven Eckert und Tilman Isselhorst (Hg.): *Wirtschaftsinformatik 2005. EEconomy, eGovernment, eSociety*: Physica-Verlag Heidelberg, S. 587–606.

Sowa, J. F.; Zachman, J. A. (1992): Extending and formalizing the framework for information systems architecture. In: *IBM Systems Journal* 31 (3), S. 590–616.

Stein, S.; Ivanov, K. (2007): Fachliche Beschreibung von Services. In: Starke und Tilkov (Hg.): *SOA-Expertenwissen. Methoden, Konzepte und Praxis serviceorientierter Architekturen*: dpunkt, Heidelberg, S. 213–227.

The Open Group (2003): *The Open Group Architecture Framework (TOGAF) (Version 8.1)*.

Tilkov, S.; Starke, G. (2007): Einmaleins der serviceorientierten Architekturen. In: Starke und Tilkov (Hg.): *SOA - Expertenwissen. Methoden, Konzepte und Praxis serviceorientierter Architekturen*. Heidelberg: dpunkt, S. 9–36.

Umek, A.; Tannhäuser, C. (2006): Wert schaffen durch nutzenorientiertes Architekturmanagement in EAI-Projekten. In: Aier, Stephan, Schönherr, M. (Hg.): *Enterprise Application Integration. Serviceorientierung und nachhaltige Architekturen*. 2. Aufl. Berlin: GITO-Verl. (Enterprise architecture), S. 49–74.

US DoC (Hg.) (2007): *Enterprise Architecture Capability Maturity Model*. Enterprise Architecture Program Support. United States Department of Commerce. Online verfügbar unter http://ocio.os.doc.gov/s/idcplg?IdcService=GET_FILE&dDocName=PROD01_004935&RevisionSelectionMethod=LatestReleased&Rendition=Alternate, zuletzt geprüft am 25.07.2016.

Vogler, Petra (2006): *Prozess- und Systemintegration. Evolutionäre Weiterentwicklung bestehender Informationssysteme mit Hilfe von Enterprise Application Integration*. Wiesbaden: GWV.

Wilde, T.; Hess, T. (2006): *Methodenspektrum der Wirtschaftsinformatik. Überblick und Portfoliobildung*. Arbeitsbericht Nr. 2/2006,. Ludwig-Maximilians-Universität München. Institut für Wirtschaftsinformatik und Neue Medien.

Winter, Robert (2003): Modelle, Techniken und Werkzeuge im Business Engineering. In: H. Österle und R. Winter (Hg.): *Business Engineering. Auf dem Weg zum Unternehmen des Informationszeitalters*. 2. Aufl.: Springer Berlin Heidelberg, S. 87–118.

Winter, Robert (2006): Ein Modell zur Visualisierung der Anwendungslandschaft als Grundlage der Informationssystem-Architekturplanung. In: Joachim Schelp und Robert Winter (Hg.): Integrationsmanagement. Planung, Bewertung und Steuerung von Applikationslandschaften. 1. Aufl.: Springer Berlin Heidelberg (Business Engineering), S. 1–29.

Zachmann, J. A. (1987): A Framework for Information Systems Architecture. In: IBM Systems Journal, Vol. 26 No. 3, S. 276–292.

9611711

Diese Seite bleibt aus technischen Gründen frei!

9611711

Lösungen zu den Übungsaufgaben

Aufgaben zu Kapitel 2

1. Was ist unter einer Unternehmensarchitektur zu verstehen?

Im Rahmen des Informationsmanagements finden sich unterschiedliche Architekturverständnisse wieder, die in der Regel nach betriebswirtschaftlicher oder technischer Sicht unterschieden werden. Anhand dieser groben Einteilung können Architekturen nach unterschiedlichen Abstraktionsgraden eingeordnet werden. Die Unternehmensarchitektur bietet den höchsten Abstraktionsgrad. Bei ihr wird das gesamte Unternehmen von der Strategie über die Geschäftsprozesse bis zur IT abgebildet. Die Unternehmensarchitektur beinhaltet dabei sowohl Organisationsstrukturen, als auch die darunterliegenden Subarchitekturen mit niedrigeren Abstraktionsgraden. Hierzu gehören die Geschäftsprozessarchitektur, die Applikationsarchitektur sowie die Systemarchitektur.

2. Was ist unter einer Applikationsarchitektur zu verstehen?

Eine Applikationsarchitektur beinhaltet die Gesamtheit aller Applikationen in einer Organisation. Sie visualisiert modellartig den Aufbau und das Zusammenspiel aller Applikationen sowie deren Schnittstellen. Hierbei werden der Informationszusammenhang und die Abhängigkeiten in aggregierter Form dargestellt.

3. Welche Ziele werden mit einer Applikationsarchitektur verfolgt?

Bei einer Applikationsarchitektur kann zwischen strategischen und operativen Zielen unterschieden werden. Die strategischen und damit langfristigen Ziele verfolgen eine Erhöhung der Flexibilität und der Anpassungsgeschwindigkeit an veränderte Anforderungen sowie eine Minimierung des Risikos im Unternehmen. Es soll eine Gesamtsicht, durch Bündelung von Funktionalitäten, Verantwortlichkeiten und Daten erreicht werden. Die operativen Ziele haben einen kurzfristigen Horizont und sind auf die Optimierung von Kosten, Qualität und Zeit ausgerichtet. Sie beschreiben hauptsächlich Prozessziele, welche sich insbesondere auf qualitative und zeitliche Aspekte auswirken.

4. Nennen Sie fünf Ziele, die mit Architekturmodellen verfolgt werden.

- Verständlichkeit durch Genauigkeit der Abbildung unterschiedlicher Unternehmenskomponenten.
- Kommunikationsmittel für die Kommunikation zwischen den Fachabteilungen und der Informatik.
- Einfache Benutzung und Veränderungsmöglichkeit von Architekturmodellen.
- Erhöhung der Datenkonsistenz und Datensicherheit.
- Erhöhung der Planbarkeit der Systeme.

Diese Seite bleibt aus technischen Gründen frei!

9611711

5. Von welchem Ansatz geht Zachman bei seinem Framework for Enterprise-Architecture aus? Nennen Sie die von Zachman verwendeten Perspektiven (Rollen) und Sichten.

Zachman geht davon aus, dass in der Unternehmensarchitektur eine Reihe von Informationssystemarchitekturen vorhanden sind. D. h., dass sich in einem Gesamtsystem nicht nur eine Architektur befindet, sondern dass die Gesamtstruktur von mehreren Architekturen beschrieben wird.

Die von Zachmann identifizierten Perspektiven (Rollen) sind: Scope (Planer), Enterprise Model (Owner), System Model (Designer), Technology Model (Builder), Components (Subcontractor) und Functioning Enterprise.

Die von ihm verwendeten Sichten sind: Data (What?), Function (How?), Network (Where?), People (Who?), Time (When?), Motivation (Why?)

6. Beschreiben Sie die Informationssystem-Architektur (ISA) nach Krcmar.

Die in Kreiselform dargestellte Informationssystem-Architektur (ISA) nach Krcmar besteht aus vier Schichten:

- In der obersten Schicht sind die Elemente der Geschäftsstrategie abgebildet, deren Vision sich durch das ganze Unternehmen und alle Sub-Architekturen zieht.
- Auf der zweiten Schicht befindet sich die Organisationsarchitektur, auf der die Strategie der Geschäftsarchitektur umgesetzt wird. Die organisatorische Schicht besteht aus der Prozess- und der Aufbauorganisationsarchitektur.
- Die dritte Schicht enthält die Elemente Anwendungsarchitektur, Daten-Architektur und Kommunikationsarchitektur. Die Anwendungsarchitektur beschreibt dabei die Funktionen, die Datenarchitektur, den statischen Zusammenhang zwischen den Daten. Die Kommunikationsarchitektur bildet die Möglichkeit Informationsflüsse abzubilden.
- Die vierte Schicht stellt die technische Infrastruktur dar. Sie beschreibt, welche IT wo im Unternehmen verwendet werden.

Die Darstellung der gesamten Informationssystem-Architektur als Kreisel, soll die Notwendigkeit der Abstimmung aller Schichten verdeutlichen.

7. Nennen Sie die Ebenen der Architekturpyramide nach Dern und erläutern Sie die Informationsarchitektur näher.

Die Architekturpyramide von Dern besteht aus den Ebenen: Strategie, Geschäftsarchitektur, Informationsarchitektur, IT-Architekturen, IT-Basisinfrastruktur.

Die Informationsarchitektur beinhaltet Strukturprinzipien und Regeln für die Gestaltung der Informationssystemlandschaft. Sie bildet somit das Fundament für die Architekturplanung. Besondere Beachtung innerhalb der Ebene Informationsarchitektur findet dabei das IS-Portfolio mit seinen Ist- und Soll-Zuständen.

Das Ist-IS-Portfolio umfasst neben der Aufstellung der vorhandenen Informa-

tions- und Subsysteme die Analyse und Bewertung der Informationssysteme und die Beschreibung des Integrationsgrades der Anwendungen, sowie die Darstellung der wichtigsten Informationsflüsse und Schnittstellen.

Das Soll-IS-Portfolio beinhaltet die geplante Aufstellung der Informationssysteme unter Berücksichtigung von deren Lebenszyklus. Anhand ausgewählter Kriterien wird die geplante Informationssystemlandschaft strukturiert und eine Integrationsstrategie erarbeitet. Ebenso werden die geplanten wichtigsten Informationsflüsse und Schnittstellen dargestellt.

Weitere Elemente der Informationsarchitektur sind die Technologiestrategie, die Architekturstrategie und die Architekturprinzipien. Die Technologiestrategie setzt die Entwicklungsleitlinie der IT-Basisinfrastruktur fest. Mit der Architekturstrategie werden der Zielzustand und die Maßnahmen beschrieben, die im Rahmen des IT-Architekturmanagements erreicht werden sollen. Die Architekturprinzipien bilden den Rahmen für Richtlinien und Standards bei der Weiterentwicklung des IS-Portfolios.

Aufgaben zu Kapitel 3

1. Was wird unter dem Begriff Integration im Zusammenhang mit dem Bereich des Informationsmanagements verstanden?

Die Wirtschaftsinformatik betrachtet die Integration einerseits aus technischer Sicht, wie z. B. IT-Infrastruktur und beachtet andererseits ebenso betriebswirtschaftliche Aspekte, wie z. B. Prozesse und Daten. Dabei wird unter Integration ganz allgemein die (Wieder-)Herstellung eines Ganzen oder einer Einheit verstanden, unabhängig vom technischen oder betriebswirtschaftlichen Aspekt. Die Integration wird entweder durch Verbinden oder durch Vereinigen von Objekten erreicht und kann auch als eine Zusammenführung von bestimmten funktionalen Eigenschaften definiert werden.

2. Nennen Sie die unterschiedlichen Integrationsperspektiven und geben Sie mindestens zwei Ausprägungen je Perspektive an.

Integrationsgegenstand: Datenintegration, Funktionsintegration, Prozess-/Vorgangsintegration

Integrationsrichtung: horizontale Integration, vertikale Integration

Integrationsreichweite: Bereichsintegration, Innerbetriebliche Integration, Zwischenbetriebliche Integration

Automationsgrad: Vollautomation, Teilautomation

3. Was wird unter Datenintegration verstanden?

Daten sind der zentrale Integrationsgegenstand der Wirtschaftsinformatik. Bei der Datenintegration werden Daten aus den unterschiedlichen Applikationen zusammengeführt. Dies kann sowohl auf physischer als auch auf logischer Ebene betrachtet werden. Bei der physischen Datenintegration steht die Zusammenführung von Daten aus heterogen gewachsenen Applikationslandschaften im Vordergrund, ungeachtet ihrer funktionalen Verwendung. Die logische Datenintegration hat einen eher interdisziplinären Charakter. Hier geht es um die eher lose Verbindung von Prozessdaten, die in einem betriebswirtschaftlichen Kontext zueinander stehen. Dabei werden Daten über Schnittstellen zwischen den Applikationen ausgetauscht, ohne in einer gemeinsamen Datenbasis gehalten zu werden.

4. Was wird unter Funktionsintegration verstanden?

Durch die Funktionsintegration sollen Aufgaben und Lösungsverfahren miteinander verknüpft werden. Dies geschieht anhand von Kommunikationskanälen zwischen Applikationen. Hierbei wird zwischen aufgabenträgerorientierter und datenflussorientierter Funktionsintegration unterschieden.

Die aufgabenträgerorientierte Funktionsintegration bezieht sich auf die Vereinigung von teilautomatisierten Aufgaben, um die Bearbeitung dieses Aufgabenblocks von nur einer Person zu gewährleisten. Hierbei soll die Kommunikation über einen Kommunikationskanal zwischen Mensch und Applikation geschehen. Dadurch werden personelle und rechnergestützte Aufgabenteile miteinander ver-

bunden. Ebenso können mehrere teilautomatisierte Aufgaben durch eine Person vereinigt werden, sofern sie dem gleichen Aufgabenträger zugeordnet sind. Die so entstandenen Aufgabenblöcke können nun an einem Arbeitsplatz bearbeitet werden.

Die datenflussorientierte Funktionsintegration betrachtet die datenflussorientierte Funktionsstruktur in einer Applikation. Die Applikation besteht aus einem Netz solcher Funktionen, sowie Datenspeichern und Ein- und Ausgabekanälen. Hierüber werden Dateien und Nachrichten gesendet. Bei der datenflussorientierten Funktionsintegration wird über eine bestimmte Kommunikationsstruktur versucht, die benötigten Kommunikationskanäle zu automatisieren und Automationsinseln zu verbinden.

5. Nennen Sie fünf Ziele, die mit der Integration verfolgt werden.

- Reduktion von künstlichen Grenzen zwischen Organisationseinheiten, Funktionsbereichen und Prozessen
- Reduktion des personellen Eingabeaufwandes
- Reduktion von Erfassungsfehlern
- Realisierung betriebswirtschaftlicher Konzepte
- Steigerung der Qualität von betrieblichen Prozessen

6. Worin besteht der Nutzen von EAI-Systemen?

Die Zielsetzungen von EAI-Systemen können unter die Überbegriffe Redundanz, Verknüpfung, Konsistenz und Zielorientierung gefasst werden. EAI-Systeme lösen Integrationsprobleme, die durch heterogen gewachsene Applikationslandschaften in einem Unternehmen entstehen. Durch Implementierung geeigneter Integrationstechnologien, wie z. B. Hub-and-Spoke- oder Bus-Architekturen, ermöglichen sie eine flexible Integration von geschäftsbereichsübergreifenden Geschäftsprozessen in verteilten Organisationen oder von Kunden und Lieferanten durch CRM-, EC- oder SCM-Systeme. Dadurch wird auch die Basis für die Automatisierung von internen und stabilen Netzwerken geschaffen. Die Implementierung neuer und die Wartung bestehender Applikationen kann außerdem mit geringerem Aufwand erfolgen.

7. Was wird unter einer SOA verstanden?

Eine Serviceorientierte Architektur (SOA) kann als prozessorientierte Integrationsarchitektur beschrieben werden. Serviceorientierte Architekturen enthalten unterschiedliche Dienste und Funktionalitäten, auch Services genannt, die die Verbindung zwischen der Geschäftsprozessarchitektur und der Applikationsarchitektur herstellen sollen. Eine SOA beschäftigt sich mit dem Entwurf von im Laufe der Zeit gewachsenen, heterogenen Systemlandschaften. Diese enthalten in der Regel Redundanzen im Daten- und Funktionsbereich. Durch eine SOA wird versucht, die bereits bestehenden Gestaltungsansätze von Informationssystemen zu erweitern und über standardisierte Schnittstellen den Integrationsgrad dieser heterogenen Systemlandschaften zu erhöhen.

Aufgaben zu Kapitel 4

1. Was wird unter Architekturmanagement verstanden?

Unter Architekturmanagement werden alle Aspekte der Erstellung, Pflege und des Einsatzes von Architekturmodellen verstanden werden. Neben der technischen Umsetzung kümmert sich das Architekturmanagement auch um die Etablierung geeigneter Organisationsstrukturen und Prozesse im Unternehmen. Das Architekturmanagement soll dabei die Dokumentation, Analyse, Planung, Entwicklung und Steuerung der Architektur überwachen und umsetzen. Es dient der Flexibilisierung und der Nachhaltigkeit von Architekturen.

2. Nennen Sie fünf Ziele, die mit dem Architekturmanagement verfolgt werden.

Das Architekturmanagement verfolgt folgende:

- Unterstützung des Business/IT-Alignment
- Erhaltung der Transparenz und Flexibilität von Geschäftszielen, Prozessen, Applikationen und der Infrastruktur
- Optimierung von Kosten im Hinblick auf z. B. Infrastruktur- und Entwicklungskosten
- Unterstützung der Standardisierung und Redundanzvermeidung von Prozessen, Applikationen und benötigter Infrastruktur
- Erhöhung der Wartbarkeit und Entwicklungsgeschwindigkeit von Architekturkomponenten

3. Welche Aufgaben hat das Architekturmanagement?

Die Aufgaben des Architekturmanagements können in operative und strategische Aufgaben unterschieden werden. Zu den strategischen Aufgaben des Architekturmanagements gehört die Dokumentation, Analyse und Planung. Hierzu zählen z. B. die Erhebung der Ist-Architektur, die Auswertung des Anforderungs- und Portfoliomanagements, die Analyse der Ist-Architektur, die Synchronisation zwischen Anforderungs- und Portfoliomanagement und den Ergebnissen der Architekturanalyse. Die operativen Aufgaben des Architekturmanagements umfassen die durchgängige Entwicklung bzw. Umsetzung und Steuerung der Architektur. Hierunter fallen z. B. die Umsetzung der strategischen Aufgaben des Architekturmanagements, die Entwicklung von Architekturkomponenten sowie von Referenzarchitekturen.

4. Nennen und erklären Sie die Phasen des Architekturlebenszyklus.

- Dokumentation: Am Anfang eines zielgerichteten Architekturmanagement muss die Dokumentation der Ist-Architektur stehen. Die Ist-Situation wird durch die Dokumentation transparent abgebildet. Die Dokumentation liefert die Vorlage für eine effiziente Architekturanalyse und beinhaltet alle Schichten und Komponenten, die für den vertikalen und horizontalen Aufbau der Architektur definiert werden. Zusätzlich werden hierfür auch Konstruktionsprinzipien und Verwendungsrichtlinien erfasst. Die Dokumenta-

tion muss aufzeigen, nach welchen Prinzipien Schnittstellen und Kommunikationsbeziehungen in der Architektur aufgebaut sind und welche betriebswirtschaftlichen Einsatzszenarien vorherrschen. Zudem müssen Einsatzheuristiken, wie z. B. Mengengerüste, Infrastrukturnutzung, Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Ausfallsicherheit beinhaltet sein.

- **Analyse:** Nachdem die vorhandene Ist-Architektur umfassend dokumentiert wurde, muss diese genau analysiert werden. Bei der Analyse wird die Architektur gedanklich in ihre Einzelteile zerlegt und logisch nach Prozessen und Systemen neu geordnet. Hierbei müssen alle Abhängigkeiten, Abdeckungen und Schnittstellen aufgedeckt werden, die die Applikationsarchitektur umfasst. Zudem muss die Heterogenität, Komplexität und Konformität geprüft werden sowie eine Kosten-Nutzen-Analyse erfolgen, um Schwachstellen in der vorherrschenden Architektur zu erkennen.
- **Planung:** Im Rahmen der Planung werden die in der Analyse aufgedeckten Schwachstellen vorab bewertet. Dadurch soll festgestellt werden, welche Ziele und Maßnahmen zur Veränderung dringend erfolgen müssen und welche Schwachstellen eine geringere Priorität besitzen. Diese Bewertung erfolgt im Rahmen der Steuerung des Architekturmanagements. Im Rahmen der Planung werden Ziele und Maßnahmen festgelegt, die zur Erreichung der gewünschten Soll-Architektur führen. Hierbei müssen vorhandene Architekturstandards und -prinzipien beachtet werden. Wird ein neues Architekturdiseign entwickelt, ist es auch erforderlich, neue Standards und Prinzipien zu formulieren.
- **Entwicklung:** In dieser Phase wird die Soll-Architektur entwickelt. Die Entwicklung ist eng gekoppelt mit der Planung. Die Soll-Architektur muss für die Umsetzung operationalisiert werden. Hierzu dienen die unterschiedlichen Architekturmodelle, die einen Rahmen für den Umsetzungsprozess vorgeben. Der Entwicklungsprozess kann außerdem durch den Einsatz von Referenzarchitekturen unterstützt werden. Sind keine für die gewünschte Soll-Architektur passende Referenzarchitekturen vorhanden, so kann in dieser Phase zunächst eine solche Architektur entwickelt werden. Die Entwicklung der Soll-Architektur geschieht dann in zwei großen Schritten: als erstes werden die notwendigen Architekturkomponenten entwickelt bzw. vorhandene Komponenten werden an die neuen Anforderungen angepasst. Abschließend erfolgt die Integration der Architekturkomponenten zu einer neuen Architektur.
- **Steuerung:** Die Steuerung begleitet alle anderen Phasen und stellt den Erfolg und die Wirtschaftlichkeit des Architekturmanagements sicher. Dabei werden unterschiedliche Bewertungsinstrumente eingesetzt. Z. B. im Rahmen der Analyse werden Risiken, Schwachstellen sowie Verbesserungspotentiale durch Portfolio-Analysen bewertet und priorisiert. Im Rahmen der Planung und Entwicklung werden Planungs- und Umset-

zungsszenarien anhand unterschiedlicher Qualitätskriterien (z. B. Zuverlässigkeit, Wartbarkeit, Sicherheit, Modifizierbarkeit) mit der Architecture Trade-off Analysis Method, oder ihre Wirtschaftlichkeit mit der Cost-Benefit Analyse methode bewertet. Zur laufenden, ganzheitlichen Steuerung der Architektur kann die Balanced Scorecard angewendet werden.

5. Was wird unter dem Architecture Capability Maturity Model (ACMM) verstanden?

Das Architecture Capability Maturity Model (ACMM) wurde vom US Department of Commerce (DoC) entwickelt, um die interne Bewertung von IT-Architekturprojekten und -entwicklungen besser unterstützen zu können. Ziel ist es, den Architekturmanagementprozess von der Dokumentation bis zur Entwicklung besser steuern zu können. Anhand von Schwachstellenanalysen sollen Vorgehen aufgezeigt werden, wie vorhandene Erfolgspotenziale besser ausgeschöpft werden können. Die Architektur muss mit Hilfe eines Änderungsmanagements kontinuierlich gewartet und weiterentwickelt werden. Hierbei soll der evolutionäre Charakter Beachtung finden. Zudem muss eine ständige Verbesserung der Fähigkeiten einer Organisation im Hinblick auf die Architekturprozessreife gewährleistet sein. Das ACMM besteht aus drei Teilbereichen. Die ersten beiden Teile, das IT-Architekturreifegradmodell sowie die IT-Architectureigenschaften der einzelnen operativen Geschäftseinheiten in den verschiedenen Reifegraden, bilden eine Grundlage zur Beschreibung und Bewertung der Fähigkeiten der Organisation im Rahmen der Architekturreifegrade. Hier werden die einzelnen Reifegrade erläutert und die Fähigkeiten hinsichtlich der Architectureigenschaften aufgezeigt. Die gewonnenen Ergebnisse dienen als Messgrößen im Rahmen des Bewertungsprozesses. Der dritte Teil besteht aus der ACMM-Scorecard, mit der der Reifegrad der Architekturfähigkeiten der gesamten Organisation abgeleitet werden kann, der dann an den CIO berichtet wird.

Diese Seite bleibt aus technischen Gründen frei!

9611711

9611711

002516497
(10/17)

41760-6-04-S 1



Alle Rechte vorbehalten
© 2017 FernUniversität in Hagen
Fakultät für Wirtschaftswissenschaft