

FernUniversität in Hagen

Designprinzipien und Erfolgsfaktoren von Applikationsarchitekturen in Dienstleistungsnetzwerken

Bachelorarbeit

Vorgelegt der Fakultät für Wirtschaftswissenschaft
der FernUniversität in Hagen
Lehrstuhl für Betriebswirtschaftslehre,
insbesondere Informationsmanagement

Von: Dipl.-Inf. Thomas Gawehns
Weichselstrasse 3
90419 Nürnberg
Matrikelnummer: 7499949

Gutachter: Univ.-Prof. Dr. Ulrike Baumöl
Betreuer: Dipl.-Kffr., Dipl.-Volksw. Martina Meschke

Abgabe am: 30.09.2010

Sommersemester 2010, 6. Studiensemester

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	II
Tabellenverzeichnis.....	III
Abkürzungsverzeichnis.....	IV
1 Applikationsarchitekturen im Kontext „Dienstleistungsnetzwerke“.....	1
2 Applikationsarchitekturen.....	2
2.1 Begriffsabgrenzung.....	2
2.2 Designprinzipien.....	4
2.3 Erfolgsfaktoren.....	9
2.3.1 Technische Erfolgsfaktoren.....	10
2.3.2 Projektbezogene Erfolgsfaktoren.....	15
3 Dienstleistungsnetzwerke.....	19
3.1 Beispiele.....	20
3.2 Typen von Dienstleistungsnetzwerken.....	21
4 Designprinzipien und Erfolgsfaktoren von Applikationsarchitekturen im Kontext von Dienstleistungsnetzwerken.....	25
4.1 Austausch von strukturierten Geschäftsdokumenten mit Wartezeit.....	27
4.2 Direkte Verarbeitung strukturierter Geschäftsdokumente.....	29
5 Zusammenfassung und Ausblick.....	33
Literaturverzeichnis.....	36
Eidesstattliche Erklärung & Einverständniserklärung.....	42

Abbildungsverzeichnis

Abbildung 2.1: Applikationsarchitektur.....	3
Abbildung 2.2: Das D&M IS Success Modell	11
Abbildung 2.3: Agilität von Applikationsarchitektur	14
Abbildung 3.1: Typologie nach Datenformat und erwarteter Antwortzeit	23

Tabellenverzeichnis

Tabelle 2.1: Designprinzipien.....	9
Tabelle 2.2: technische Erfolgsfaktoren.....	12
Tabelle 2.3: projektbezogene Erfolgsfaktoren.....	18
Tabelle 5.1: Designprinzipien und Erfolgsfaktoren im Kontext Dienstleistungsnetzwerke.....	35

Abkürzungsverzeichnis

BMW	Bayerische Motorenwerke
bzgl.	bezüglich
bzw.	beziehungsweise
CAD	Computer Aided Design
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
DIN69901	Deutsche Industrie Norm 69901 (Projektmanagementsysteme)
EDV	Elektronische Datenverarbeitung
ERP	Enterprise Resource Planning
et al.	Et alii (und andere)
EUSOP	End User Service Orchestration Platform
evtl.	eventuell
ggf.	gegebenenfalls
GPM	Deutsche Gesellschaft für Projektmanagement e.V.
GPn	Geschäftsprozess n
IS	Information System
IT	Informationstechnik
OMG	Object Management Group
QDX	Quality Data Exchange
PLM	Product Lifecycle Management
REST	Representational State Transfer
s.	siehe
s.o.	Siehe oben
SAP	SAP Aktiengesellschaft Systeme, Anwendungen und Produkte in der Datenverarbeitung
Sn	Service n
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SWIFT	Society for Worldwide Interbank Financial Telecommunication
TCP/IP	Transmission Control Protocol / Internet Protocol
UBL	universal business language
VDA	Verband der Automobilindustrie
Vgl.	Vergleiche
VPN	Virtual Private Network
XML	Extensible Markup Language
z.B.	zum Beispiel

1 Applikationsarchitekturen im Kontext „Dienstleistungsnetzwerke“

Die EDV-Applikationen von Unternehmen sollten in Form einer Applikationsarchitektur einer gewissen Ordnung und Struktur genügen. Musste man sich früher noch mit einer ungeplanten Entstehung von Applikationen mehr oder weniger abfinden, so liegen heutzutage bekannte Empfehlungen für Designprinzipien und Erfolgsfaktoren solcher Applikationsarchitekturen vor. (Vgl. Legner/Heutschi 2007, Winter 2003, DeLone/McLean 2003). Allerdings wird in diesen Empfehlungen der Aspekt der dauerhaften Vernetzung von Unternehmen nicht ausdrücklich beachtet.

Gerade eine Zusammenarbeit im Dienstleistungsbereich kann einen Austausch von vielfältigen Informationen erfordern. So sind Beiträge der einzelnen Akteure aufeinander abzustimmen und es ist der Zeitablauf zu planen. Je nach Dienstleistung sind Zwischenschritte zu überprüfen. Falls die Dienstleister sich dauerhaft vernetzen wollen und die Geschäftsprozesse aufeinander abstimmen, ergeben sich ggf. zusätzliche Anforderungen an die im Unternehmen eingesetzten EDV-Applikationen.

Ziel dieser Arbeit ist es zu untersuchen, ob die Designprinzipien und Erfolgsfaktoren, die für eine einzelne Applikationslandschaft eines Unternehmens empfohlen werden, auch dann gelten wenn die Unternehmen sich zu Dienstleistungsnetzwerken zusammenschließen.

Aufgebaut ist die Arbeit in drei Teile:

- Zunächst werden bekannte Designprinzipien und Erfolgsfaktoren von Applikationsarchitekturen im Unternehmenskontext geschildert.
- Anschließend werden Dienstleistungsnetzwerken im Hinblick auf die Art der Vernetzung kategorisiert.
- Zum Schluss werden dann die Kategorien in Bezug zu den Designprinzipien und Erfolgsfaktoren untersucht.

Ausgangspunkt der Arbeit sind Konferenzbeiträge zu den Wirtschaftsinformatik Konferenzen 2008 und 2010. Eine moderne Aufgabenstellung ist anscheinend genau die Vernetzung von Unternehmen und Computersystemen.

Die Arbeit setzt Kenntnisse bzgl. grundlegender Geschäftsprozesse, wie z.B. Einkauf, Logistik, Controlling usw., sowie grundlegender IT Begriffe, wie z.B. Hardware, Software, Schnittstelle, Datum usw. voraus.

2 Applikationsarchitekturen

Mit Applikationsarchitektur werden, je nach Sichtweise, unterschiedliche Sachverhalte bezeichnet. Es gibt in der wissenschaftlichen Literatur keine eindeutige Festlegung, was genau mit Applikationsarchitektur gemeint ist (vgl. Schönherr 2004, S. 7). Zunächst werden deswegen Grundbegriffe geklärt, bevor Designprinzipien und Erfolgsfaktoren untersucht werden.

2.1 Begriffsabgrenzung

Ausgehend vom einem statischen Architekturbegriff, in dem eine Architektur im wesentlichen Strukturen beschreibt, so dass ein Gesamtes aus Teilen besteht, die dann wiederum aus Teilen bestehen bzw. in anderen Beziehungen stehen (vgl. Schönherr 2004, S. 12), wird im Folgenden schrittweise der Begriff „Applikationsarchitektur“ eingeordnet.

So besteht die Architektur eines Unternehmens aus der Organisationsarchitektur, die die Organisation der Produktionsfaktoren beschreibt und der IT-Architektur, die die für das Funktionieren dieser Organisation notwendige IT beschreibt. Diese IT besteht aus einer Infrastruktur, in der die notwendige Hardware verschaltet und verkabelt ist, sowie einem Softwareanteil, in dem dann auch die Anwendungssoftware abläuft (vgl. Aier/Dogan 2004, S. 81).

In nächsten Schritt wird die Datenarchitektur von der Applikationsarchitektur getrennt (Winter 2003, S. 2), (Baumöl 2006, S. 320). Die gemäß Applikationsarchitektur strukturierten Applikationen manipulieren dann Daten in Datenbanken oder Dateisystemen, die gemäß einer irgendwie gearteten Datenarchitektur strukturiert sind. Die Architektur dieser Daten sollte allerdings nicht aus Sicht der Applikationen erfolgen, sondern aus Sicht der Unternehmensarchitektur (vgl. Baumöl/Meschke 2009, S. 64).

An der Schnittstelle zwischen Applikationsarchitektur und der Unternehmensarchitektur stehen Services, die die Schritte der Geschäftsprozesse unterstützen bzw. ermöglichen. Die einzelnen Geschäftsprozesse werden in der Organisationsarchitektur beschrieben und werden von den Fachabteilungen ausgeführt. Im Gegensatz zu anderen Autoren (z.B. Legner/Heutschi 2007, S. 1645) werden in dieser Arbeit die Services nicht von den eigentlichen Applikationen getrennt. Stattdessen wird die Repräsentation der Daten am Bildschirm von der eigentlichen Funktionalität getrennt. Diese Funktionalität besteht dann in der Manipulation einer Teilmenge der gemäß Datenarchitektur strukturierten Anwendungsdaten. Diese granulare Funktionalität wird als Service bezeichnet. Genau wie bei

den Schichten der Datenprotokolle, in denen die oberste Schicht die Anordnung der Bytes zu Zahlformaten festlegt (vgl. Clark/Tennenhouse 1990, S. 4), wird hier auch die Darstellung der Daten auf dem Bildschirm der Anwender aus der eigentlichen Applikationsarchitektur gelöst. In modernen webbasierten Implementierungen werden die Daten an den Webbrowser übermittelt, die eigentliche Darstellung ist in Skripten festgelegt (Fielding 2000, S. 86) und die Funktionalität wird durch Webservices realisiert.

Folgende Abbildung skizziert die Einordnung einer Applikationsarchitektur als Teil der Unternehmensarchitektur:

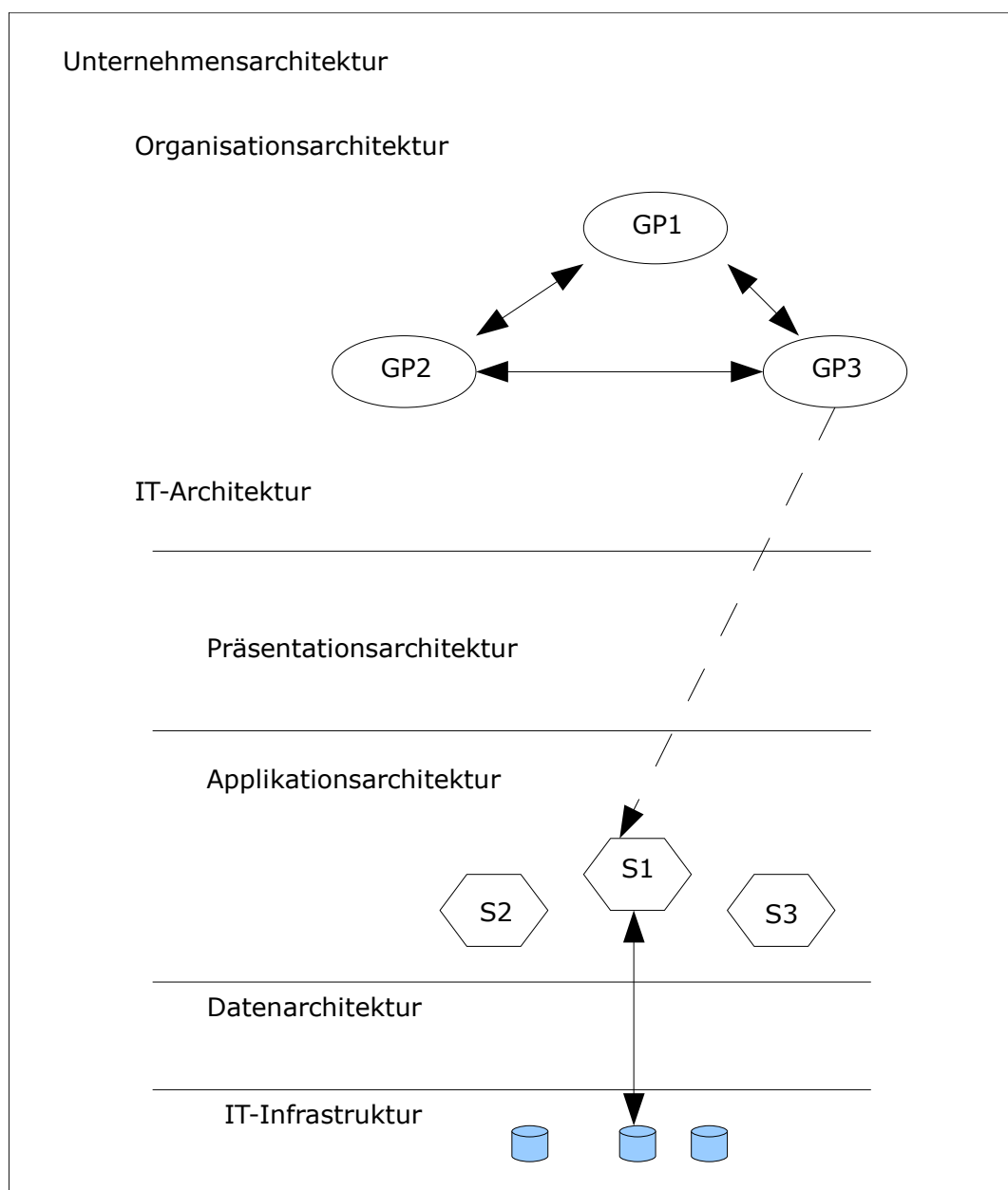


Abbildung 2.1: Applikationsarchitektur

Gezeigt wird, wie in der Organisationsarchitektur die Geschäftsprozesse GP1 bis GP3 zusammenarbeiten, wobei GP3 durch einen Service S1 unterstützt wird. Die Darstellung der Daten erfolgt durch die Repräsentationsschicht, die Abspeicherung und Manipulation der Daten wird durch die Datenarchitektur strukturiert und erfolgt in der IT-Infrastruktur. Nicht dargestellt ist der Computer und der Bildschirm des Arbeitsplatzes sowie der eventuell beteiligte Serverrechner oder Host.

Allerdings ist diese Darstellung recht vereinfachend, so kann die Repräsentationsschicht mehr umfassen als nur die Darstellung der Daten am Bildschirm. Ein Mashup, der mehrere Services zusammenfasst (vgl. Böhringer/Gluchowski 2010, S. 801), sollte dieser Schicht zugeordnet werden. Wenn die Präsentation abhängig vom jeweiligen Benutzern gehalten wird, so dass diese per Metadaten beschrieben wird (vgl. Hofmann/Ley/Dörner 2008, S. 1315), müssen diese direkt aus der IT-Infrastruktur gelesen werden, da es nicht sinnvoll scheint diese Daten durch eine Datenarchitektur zu beschreiben.

Zusammengefasst wird in einer *Applikationsarchitektur* beschrieben, welche Applikationen von der IT realisiert werden und wie diese strukturiert sind. Diese Applikationen manipulieren Daten gemäß Datenarchitektur. Die Geschäftsprozesse greifen auf die Services der Applikationen zu, die Darstellung der Ein- und Ausgabedaten der Services wird gemäß Präsentationsarchitektur beschrieben.

2.2 Designprinzipien

Eine Applikationsarchitektur, deren Design keinen fixierten Prinzipien genügt, kann zwar durchaus zu funktionsfähigen Lösungen führen, im Endeffekt entsteht meistens ein Wildwuchs von Applikationen, die je nach Entstehungsgeschichte über die IT-Struktur des Unternehmens verteilt sind (vgl. Winter 2003, S. 6). Designprinzipien werden einerseits vorgeschlagen zur Strukturierung des Designs der Applikationen einer Applikationslandschaft, damit diese dann leichter gemanagt werden kann. Andererseits gibt es auch Designprinzipien für die Services, die dann von den Geschäftsprozessen verwendet werden. Beide Vorschläge werden im folgenden vorgestellt.

Für die unternehmensinterne Strukturierung der Applikationslandschaft müssen unternehmensspezifische Designprinzipien eingehalten werden. Eine Herleitung solcher, von Unternehmen zu Unternehmen unterschiedlicher Prinzipien kann aus einer dreidimensionalen Visualisierung der Applikationen abgeleitet werden (vgl. Winter 2003, S. 5).

In solch einer Visualisierung werden die Applikationen entlang der folgenden drei Achsen angeordnet:

- Organisationseinheit oder Produktlinie
z.B. Einkauf, Vertrieb, Logistik, Produktion oder auch Niederlassungen.
- Informationen
z.B. Kunden- oder Produktdaten, Aufträge, Kennzahlen.
- Funktionalität
z.B. Analysewerkzeuge, Sicherheitssysteme, Planungswerkzeuge.

Mit Hilfe einer Graphik kann dann der Ist-Zustand visualisiert und anschließend eine Soll-Architektur diskutiert werden. Die Soll-Architektur versucht dabei ein Gleichgewicht entlang dieser Achsen zu erzielen, das immer bestrebt sein muss trade-offs aufzuzeigen, die dann von den Entscheidern auch zu tragen sind. So führt z.B. eine Aufteilung der Informationen auf die Organisationseinheiten einerseits zu Redundanzen, andererseits stehen diese duplizierten Informationen aber auch schneller zur Verfügung.

Für den Einsatz von Applikationsarchitekturen im Kontext Dienstleistungsnetzwerk ist die interne Strukturierung und die dabei ggf. einzuhaltenden Designprinzipien eigentlich uninteressant. Dienstleistungsnetzwerke sind Netzwerke von unabhängigen Unternehmen, die jeweils über getrennte Applikationslandschaften verfügen. Bei einer Vernetzung dieser Unternehmen müssen die Geschäftsprozesse aufeinander abgestimmt werden. Deswegen sind in dieser Arbeit die Designprinzipien für die Services, die ja die Schnittstelle zu den Geschäftsprozessen bilden, wichtiger.

Für das Design der Services werden in der Literatur (vgl. Klose/Knackstedt/Beverungen 2007, S. 1805), (vgl. Legner/Heutschi 2007, S. 1545 ff.) zehn Grundprinzipien beschrieben:

1. Abstraktion von der Service Implementierung
Ein Service soll verwendet werden können, ohne dass Kenntnis der Implementierung vorausgesetzt wird.
2. Klare Spezifikation des Services
Der Service soll klar dokumentiert sein.

3. Stabile, verwaltete Service Verträge

Die Services implementieren fixe Verträge, die nur in verwalteten Zyklen geändert werden können.

4. technische Standardisierung

z.B. CORBA, SOAP, REST

5. geschäftliche Standardisierung

z.B. SWIFT

6. Benutzung von offenen und weit genutzten Industrie Standards

z.B. TCP/IP, COM

7. kohäsive Servicedomänen

Eng zusammenhängende Services werden in gemeinsamen Domänen realisiert

8. schwach gekoppelte Kommunikation

Services sind möglichst asynchron miteinander verbunden und zustandslos, damit Laufzeitabhängigkeiten vermieden werden.

9. Servicegranularität ausgerichtet an Geschäftskonzepte

Ein Service sollte nicht mehrere Geschäftsprozessschritte umfassen.

10. Wiederverwendbarkeit von Services

Services sollten auch in anderen Zusammenhängen wiederverwendet werden können.

Die Einhaltung dieser Grundprinzipien führt so einer so genannten Service Orientierter Architektur (SOA), die als eine moderne Architektur gilt, deren Anwendung zunehmend auch von Anbietern von Standard Softwarepaketen aus dem Bereich ERP (Enterprise Resource Planning) angestrebt wird (Vgl. Frick/Schubert 2010, S. 1566).

Die Ausrichtung einer Architektur nach obigen Prinzipien eröffnet einem Unternehmen neue Möglichkeiten:

Durch Einhalten der Prinzipien (1) und (2) können Agentensysteme implementiert werden. Ein Agent ist eine Applikation, die ohne Benutzerinteraktion automatisch tätig wird, so dass kleinere Geschäftsprozesse automatisch ablaufen können (Vgl. Sunyaev et al. 2008, S. 1458). So können z.B. in einem Kranken-

haus die Patientendaten automatisch von angeschlossenen Ärzten zusammengesucht werden (Vgl. Sunyaev et al. 2008, S. 1460). Dies ist nur möglich, wenn kein nicht dokumentiertes Wissen bzgl. der aktuellen Implementierung des Services Vorbedingung für eine korrekte Verwendung desselben ist.

Das Einhalten von Standards (4), (5) und (6) ermöglicht die Kooperation mit externen Unternehmen. Ein unternehmensübergreifendes Projektmanagement muss Terminabsprachen einhalten. Diese Termininformationen können gemäß der Standards OMG PLM Services V2.0, QDX-Standard des VDA und des Vorschlags der GPM zur DIN69901 abgeleitet werden (Vgl. Bartlog/Boy 2010, S. 1578). Aber auch allgemeinere Dokumente können mit UBL spezifiziert werden, damit mit anderen Unternehmen zusammengearbeitet werden kann (Vgl. Becker/Janiesch/Pöppelbuß 2008, S. 819).

Falls ein Unternehmen plant, Grid Computing einzusetzen, müssen (3), (7) und (8) eingehalten werden. Beim Grid Computing sind die Services und Applikationen nicht mehr statisch den Servern zugeordnet, vielmehr wird über einen Vermittlungsservice die genaue Adresse eines Services ermittelt. Zur Laufzeit können neue Server, auch unternehmensexterne hinzugefügt werden. Dies funktioniert nur, wenn der Service einen fixen Vertrag erfüllt (3), er nur Applikationen in seiner Domäne verwendet (7) oder asynchron auf andere Services zugreift (8). Gerade die Verwendung des Protokolls REST für die Implementierung der Services (Vgl. Barton/Bach 2010, S. 2416) eignet sich für eine Verwendung im Grid. Für ein Unternehmen kann eine Gridfähigkeit neue Geschäftsmöglichkeiten eröffnen, in dem anderen Unternehmen die kostenpflichtige Benutzung von Services angeboten wird. Beispiele für solche Services können Standardservices wie XML-Schema Transformationen aber auch speziellere Services wie z.B. Anwendungen aus der Medizin (Vgl. Scholz/Breitner/Blaurock 2008 S. 1353) oder Risikoanalysen im Banking (Vgl. Blau/Schnizler 2008, S. 1359). Falls ein Marktplatz für solche Services eingerichtet werden soll, müssen Services zum Aushandeln von Preisen und Abrechnen von Leistungen implementiert werden (Vgl. Karaenke et al. 2008, S. 1387).

Diese zehn Grundprinzipien müssen zu leichter handhabbaren Designprinzipien verdichtet werden. Ein Vorschlag zu einer Zusammenfassung ist: (Vgl. Klose/Knackstedt/Beverungen 2007, S. 1806)(Vgl. Legner/Heutschi 2007, S. 1646):

- Vertragsorientierung

Die Aufgabenstellung des Services ist als klarer Vertrag zwischen Nutzer und IT formuliert. Es ist dem Nutzer verborgen, wie genau der Service realisiert wird.

- Offenheit

Der Service tauscht Daten in standardisierten Formaten aus und benutzt hierzu nur standardisierte Protokolle.

- Modularität

Semantisch zusammenhängende Services werden in Blöcke zusammengefasst, die wiederum nur wenig Datenverkehr untereinander haben.

- Geschäftsorientierung

Die Services werden aus Sicht der Geschäftsprozesse so definiert, dass die Funktionalitäten möglichst wiederverwendet werden können.

Eine alternative Darstellung von Designprinzipien (Mueller et al. 2007 S. 1610-1611) erkennt folgende Punkte:

- Modularität

Services realisieren spezifische, granulare Funktionalitäten.

- Lose Kopplung

Je zwei semantisch kompatible Services können gekoppelt werden.

- Offene Standards

Für die Kommunikation der Services werden offene Standards verwendet.

- Einfachheit

Schnittstellen, Datenformate und Funktionalitäten sollten so einfach, wie möglich gehalten werden.

Durch diese Vorschläge können die Grundprinzipien zu folgenden Designprinzipien zusammengefasst werden:

Designprinzip	Erläuterung
Service als Vertrag zwischen Nutzer und IT. (1), (2), (3)	In dem Vertrag werden Ein- und Ausgabe detailliert beschrieben. Dem Nutzer ist verborgen, wo und wie der Service realisiert wird.
Offene Standards für Datenprotokolle und -formate. (4), (5), (6)	Wichtig ist hier, dass ein Standard verwendet, ob dies jetzt auf Protokollebene HTTP, Corba oder Mqseries ist, ist irrelevant.
Verwendung lose gekoppelter Servicedomänen. (7), (8)	Semantisch zusammengehörende Services werden aus Gründen der Performance und Verwaltung zu möglichst unabhängigen Domänen zusammengefasst.
Geschäftsprozess orientiertes Design einfacher Services. (9), (10)	Von einem Modell der Geschäftsprozesse ausgehend, werden die Services so definiert, dass jeder Service eine einfache Funktionalität realisiert

Tabelle 2.1: Designprinzipien

Diese Designprinzipien werden unten (s. Abschnitt 4) auf ihre Bedeutung im Kontext Dienstleistungsnetzwerke überprüft.

2.3 Erfolgsfaktoren

Erfolgsfaktoren werden im deutschsprachigen Raum als „Einflussgrößen und Bedingungen, die für den Erfolg und Misserfolg unternehmerischen Handelns bestimmend sind“ definiert (Dömer 1998, S. 101). Im Zusammenhang mit Applikationsarchitekturen werden Erfolgsfaktoren in zweierlei Hinsicht genannt: Einerseits als Qualitätsanforderungen, die verwirklicht werden sollten, damit die Architektur erfolgreich benutzt werden kann, andererseits als organisatorische Rahmenbedingungen, die gelten müssen, damit die Architektur erfolgreich implementiert werden kann. Im folgenden wird für jede der beiden Ansichten je eine Liste von Erfolgsfaktoren zusammengestellt.

2.3.1 Technische Erfolgsfaktoren

Als technische Erfolgsfaktoren werden die Faktoren angesehen, die eine Architektur erfüllen muss, damit sie erfolgreich eingesetzt werden kann. Während die Designprinzipien als Handlungsempfehlung aufzufassen sind, beschreiben diese Faktoren eher Zustände und Ziele. Allerdings sind sie in der Literatur nicht unbedingt klar von Designprinzipien getrennt. Bei den im folgenden betrachteten Literaturstellen, werden zunächst alle Punkte aufgelistet, bevor die eigentlichen technischen Erfolgsfaktoren für Applikationsarchitekturen benannt werden.

Ein Standardmodell, mit dem der Erfolg von IT Systemen gemessen werden kann, scheint das 1992 publizierte „D&M IS Success Model“ von DeLone und McLean zu sein. Es wurde zehn Jahre später noch einmal überarbeitet und sein Ansatz wurde aktuell auch für Untersuchungen von Erfolgsfaktoren von social software benutzt (Vgl. Steinhüser/Räth 2010, S. 1783).

Die in diesem Modell beschriebenen Faktoren für den Erfolg eines IT Systemes sind (Vgl. DeLone/McLean 2003, S. 12):

- Systemqualität
Handhabbarkeit, Funktionalität, Zuverlässigkeit, Flexibilität, Datenqualität, Portierbarkeit, Integration und Bedeutung.
- Informationsqualität
Genauigkeit, Aktualität, Vollständigkeit, Relevanz und Konsistenz
- Benutzung
Die Benutzung des Systems sollte freiwillig sein und wird gemessen in Benutzungsfrequenz, -dauer, Anzahl Zugriffe, Nutzungsmuster und Abhängigkeit.
- Benutzerzufriedenheit
- Individueller Nutzen
ergibt sich als Qualität der Arbeitsumgebung, -leistung und Entscheidungsfindung.
- Unternehmensnutzen
Aggregiert die individuellen Nutzen

Diese Faktoren können zu einem Kausalmodell zusammengefügt, in dem gezeigt wird, welche Faktoren aufeinander einwirken, so dass aus den jeweiligen indivi-

duellen Nutzen der Benutzer schließlich ein aggregierter Unternehmensnutzen wird:

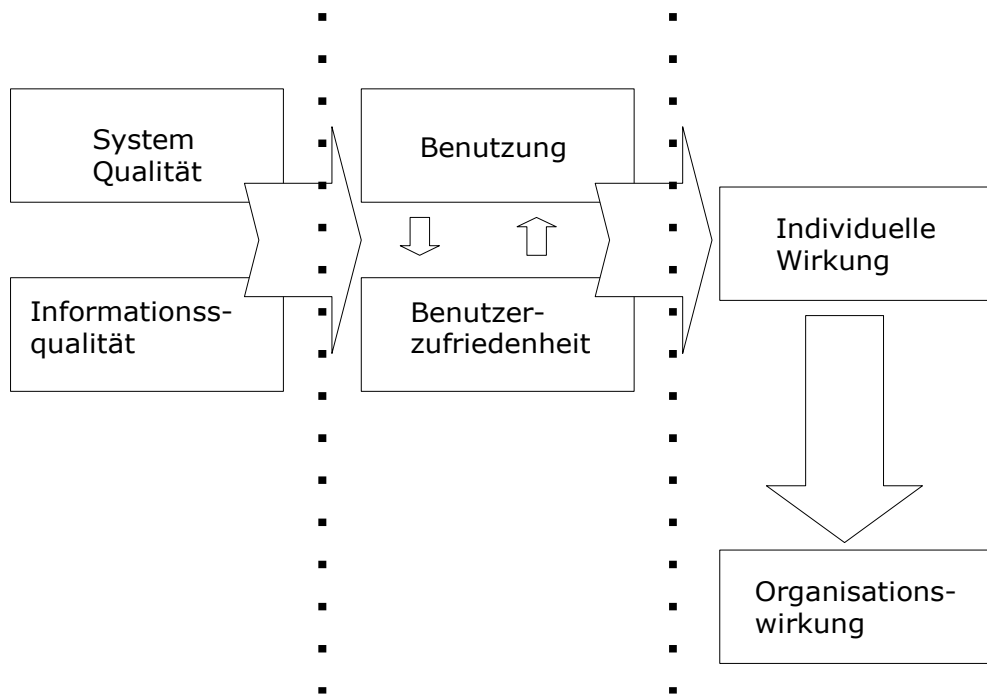


Abbildung 2.2: Das D&M IS Success Modell

Quelle: DeLone/McLean 2003, S. 12

Zahlreiche empirische Untersuchungen belegen diese Wirkungsketten (DeLone/McLean 2003, S. 14).

Es ergeben sich damit die System- und Informationsqualität als primäre Faktoren, die die anderen Faktoren beeinflussen. Allerdings sind nicht alle unter System- und Informationsqualität angegebenen Punkte auch Erfolgsfaktoren. Durch das Designprinzip „Geschäftsprozess orientiertes Design einfacher Services“ werden schon die Faktoren Funktionalität, Integration und Bedeutung berücksichtigt. Da die Faktoren, die die Datenqualität betreffen, der Datenarchitektur zugeordnet sind und der Faktor Handhabbarkeit besser durch die Repräsentationsschicht berücksichtigt wird, verbleiben als Erfolgsfaktoren:

- Zuverlässigkeit
Das System muss zuverlässig arbeiten.
- Flexibilität
Änderungen müssen leicht erfolgen.

- Portierbarkeit

Auf andere ggf. modernere Plattformen muss portiert werden können.

- Aktualität

Die Daten müssen schnell genug zur Verfügung stehen.

Ein Faktor, der erst durch die Möglichkeiten zur Vernetzung hinzukommt ist (Vgl. Bianco/Kotermanski/Merson 2007, S. 65), (Vgl. Bass/Klein/Moreno 2001, S. 7):

- Sicherheit

Aussenstehende dürfen keinen Zugang zu den Daten, Informationen oder Diensten haben

So dass als technische Erfolgsfaktoren verbleiben:

Technischer Erfolgsfaktor	Erläuterung
Zuverlässigkeit	Selbst bei Ausfall von Komponenten muss das System ansprechbar sein und ggf. mit eingeschränkter Funktionalität arbeitsfähig sein.
Agilität	Die Funktionalität muss schnell geändert werden können.
Portierbarkeit	Auf unterschiedlichen Hard- und Systemsoftwareplattformen müssen die Applikationen ablaufen können bzw. mit wenig Aufwand ablaufbar gemacht werden können.
Effizienz	Die Informationen sind zeitnah zur Verfügung zu stellen, Dienstanforderungen in definierten Zeitrahmen zu beantworten.
Sicherheit	Aussenstehende dürfen keinen Zugang zu Daten, Informationen oder Diensten haben.

Tabelle 2.2: technische Erfolgsfaktoren

Von allen Faktoren wird dem Faktor Agilität in der Literatur besondere Aufmerksamkeit geschenkt (Vgl. Becker/Buxmann/Widjaja 2009 S. 2096). Insbesondere das gut untersuchte Bankenumfeld erzwingt geradezu die Auseinandersetzung mit der Agilität. Zum einen sind gewachsene Strukturen entstanden, die zu Systemen strukturiert werden müssen, um überhaupt verwaltbar zu sein. Daneben müssen ständig neue Anforderungen und Regularien realisiert werden. Außer-

dem gibt durch Übernahmen die Anforderung übernommene Systeme zu integrieren bzw. umgekehrt im Falle einer Übernahme integriert zu werden (vgl. Baskerville et al. 2005, S. 764).

Faktoren und Designprinzipien, die die Agilität erhöhen und unterstützen sind (vgl. Schwinn/Winter 2007 S. 2180):

- Minimale Integrationskosten (Zeit und Geld)
Der Aufwand zur Integration von neuen oder geänderten Applikationen in die bestehende Applikationslandschaft.
- Wiederverwendung/funktionale Redundanz
Es sollte eigentlich jede Funktion nur einmal implementiert sein. Allerdings steigen dann ggf. die Laufzeitkosten, da die jeweiligen Funktionalitäten zentral verwaltet werden müssen.
- Komplexität der Applikationsarchitektur
Eine große Zahl von Applikationen kann nicht jeweils für sich verwaltet werden, deswegen werden die Applikationen zu Blöcken (Domänen) zusammengefasst.
- Kopplungsgrad der Applikationen an Geschäftsprozesse
Drückt das Verhältnis von Anzahl der Applikationen zu Anzahl der Geschäftsprozesse aus. Je weniger Applikationen für die Geschäftsprozesse verwendet werden, um so weniger Wartungsaufwand ist notwendig.
- Kosten und Komplexität der Infrastrukturkomponenten
Falls nur einige Komponenten verwendet werden, können diese professionell unterstützt werden. Allerdings schränken die Komponenten auch die Möglichkeiten ein.

Das Zusammenwirken dieser Faktoren erhöht die „Agilität“ der Architektur, d.h. die Fähigkeit auf Änderungen schnell reagieren zu können (Schwinn/Winter 2007, S. 2182):

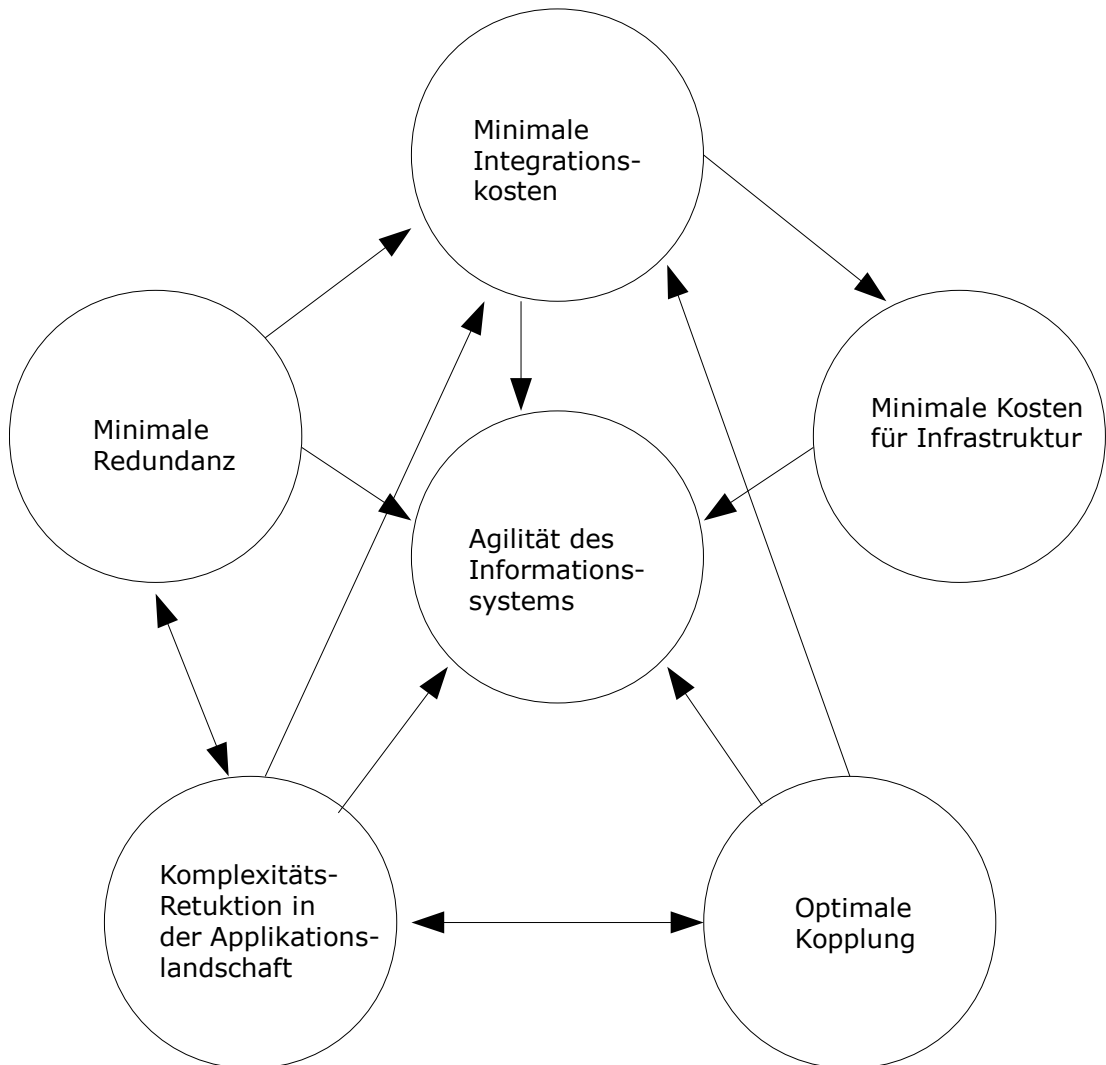


Abbildung 2.3: Agilität von Applikationsarchitektur

Quelle: Schwinn/Winter 2007, S. 2182

Eine weitere von verschiedenen Autoren vorgeschlagene Methode zur Erhöhung der Agilität besteht in der sogenannten „Orchestrierung“ von Services. Hierdurch sind die Services nicht fest einem Server zugeordnet, sondern die Anfragen werden durch ein Service Repository geroutet (Vgl. Knöfel/Barth 2010, S. 816/817; (Laures 2006 S. 157). Dadurch wird die, durch die unterschiedlich aktiven Geschäftsprozesse erzeugte, Arbeitslast auf vorhandenen Computersysteme flexibel verteilt.

Die Zusammenschaltung dieser Services kann mit einer speziellen Beschreibungssprache realisiert werden, wie im Projekt EUSOP vorgestellt. Hier können dann sogar abhängig vom Arbeitsplatz spezielle Services zusammenschaltet werden. (Vgl. Hofmann/Ley/Dörner 2008 S. 1315). Aber nicht nur einzelne Geschäftsprozesse können so konfigurierbar gestaltet werden, es ist auch möglich ganze Arbeitsabläufe, sogenannte work-flows, agil zu beschreiben. Dies kann dann von den Fachabteilungen selber gestaltet werden (Vgl. Minor/Schmalen/Bergmann 2008, S. 1991).

Ob allerdings die Agilität einer Applikationsarchitektur auch im Kontext Dienstleistungsnetzwerke eine überragende Stellung einnimmt, wird weiter unten untersucht.

2.3.2 Projektbezogene Erfolgsfaktoren

Unter Erfolgsfaktoren können auch die Bedingungen und Umstände verstanden werden, die für den Erfolg und Misserfolg unternehmerischen Handelns bestimmend sind (Vgl. Dömer 1998, S. 101). Da eine erfolgreiche Umsetzung von IT Projekten ebenfalls unternehmerisches Handeln beinhaltet, gibt es auch Erfolgsfaktoren, die die erfolgreiche Umsetzung von IT Projekten bestimmen. Insbesondere wenn hier eine geplante Applikationsarchitektur eingehalten werden soll. Eine gut untersuchte Klasse von Anwendungssystemen sind Systeme aus dem Umfeld ERP (Enterprise Resource Planning), die Daten und Informationen innerhalb des gesamten Unternehmens aggregieren und Prozesse automatisieren (Vgl. Dreiling 2010, S. 1597).

Diese werden in der Regel aus Standardpaketen zusammengestellt, die dann angepasst und insgesamt eingeführt werden. Für die Umsetzung solcher ERP Projekte werden folgende Erfolgsfaktoren vorgeschlagen (Vgl. Somers/Nelson 2004, S. 260 ff):

- Business Plan

Es muss eine Vision des Projektes vorliegen, die den Zweck und die Ziele des Projektes für alle Beteiligten verständlich beschreibt.

- Veränderungs- und Erwartungshaltungsmanagement

Mit der Einführung des Systems werden Veränderungen bezweckt, die vorbereitet werden müssen. Allerdings dürfen die Erwartungen nicht zu hoch geschraubt werden, damit das Projekt nicht als Fehlschlag wahrgenommen wird, obwohl es zum Unternehmenserfolg beiträgt.

-
- Datenanalyse und -konvertierung

Es müssen die richtigen Daten identifiziert und im richtigen Format in das System geladen werden.

- abteilungsübergreifende Kommunikation

In der Regel sind mehrere Abteilungen von der Einführung betroffen. Alle Schlüsselpersonen müssen regelmäßig über das Projekt unterrichtet werden.

- Business Process Reengineering

Mit der Einführung des System müssen die Geschäftsprozesse neu definiert bzw. angepasst werden.

- Projektbefürwortung und -verfechtung

des Topmanagements gilt als wichtigster Erfolgsfaktor des Projekts. Ohne Rückhalt im Topmanagement können Geschäftsprozesse nicht geändert werden und Kompromisse zwischen Abteilungszuständigkeiten nicht erzwungen werden.

- Projektmanagement

Der Verlauf des Projekts ist zu planen und zu überwachen, insbesondere im Hinblick auf die Erfahrung im Umgang mit den eingesetzten Werkzeugen und Hilfsmitteln.

- Bereitstellung von Ressourcen

Die betroffenen Abteilungen müssen genügend Ressourcen (Mitarbeiterzeit) zur Unterstützung bereitstellen. Der Bedarf muss frühzeitig geschätzt werden.

- Teamwork und -zusammensetzung

Ein Team aus IT-Professionals und Mitarbeitern der Fachabteilungen muss zusammengestellt werden. Die Arbeit innerhalb des Teams muss organisiert werden.

- Nutzung von Werkzeugen und Beschleunigern

Der geschickte Einsatz von Programmgeneratoren oder die Verwendung von parametrierbarer Software kann die Umsetzung der notwendigen Änderungen erheblich beschleunigen.

-
- Nutzertraining bzw. die Nutzerausbildung

Die Nutzer müssen mit Fertigstellung des Projekts ausreichend trainiert und ausgebildet sein.

Die Gültigkeit dieser Erfolgsfaktoren sollte empirisch für SAP Projekte in einem Großunternehmen nachgewiesen werden. Hierzu wurden die Projektverantwortlichen von 20 Projekten befragt, ob diese Faktoren eingehalten wurden und ob das Projekt erfolgreich war. Eine Auswertung ergab allerdings eine negative Korrelation dieser in der Theorie postulierten Erfolgsfaktoren (vgl. Dreiling 2010 S. 1602 ff)!

Aufschlussreicher erscheint eine Untersuchung bzgl. der Einführung von ERP Standardsoftware in mittelständischen Unternehmen. Hier wurden die Unternehmer, die solche Softwarepakete einsetzten, gefragt nach „Faktoren für dauerhaften Erfolg“ und folgende Aussage fasst das Gefundene zusammen:

„Es ist interessant zu sehen, welche Faktoren von den Autoren für dauerhaften Erfolg identifiziert werden. Deren Charakter ist nur in der Minderheit der Fälle harter Natur, z. B. zeitnaher Informationszugriff. Häufiger genannt werden weiche, nicht eindeutig messbare Kriterien wie Zukunftsfähigkeit, Agilität, Know-how sowie Vertrauensbeziehungen.“(vgl. Schubert/Williams 2010 S. 1557)

Insgesamt wurde folgende Liste von Faktoren gefunden (vgl. Schubert/Williams 2010 S. 1555-1556):

- Aktuelle Technologie, Know-how-Vorsprung
- Eigenschaften einer modernen Software (Zukunftsfähigkeit)
- Software ermöglicht Agilität (neue Funktionen leicht zu implementieren)
- Stabiler, zukunftsorientierter Softwarelieferant (Zukunftsfähigkeit)
- Branchen-Know-how des IT-Partners
- Vertrauen in den IT-Partner
- Zeitnaher Informationszugriff auf operative Zahlen, ad hoc Analysen
- Einhalten des Standards im Datenaustausch
- Einhalten des Standards einer Standardsoftware (keine individuellen Anpassungen)

Einige dieser Faktoren sind schon als Designprinzipien oder technische Erfolgsfaktoren benannt. Neu sind allerdings die Begriffe „Vertrauen“ und „Zukunftsfäh-

higkeit“. Hiermit soll wohl ausgedrückt werden, dass für den Erfolg des Projekts ein Vertrauen zwischen dem auftraggebenden Unternehmen und dem externen IT-Partner bzgl. der angestrebten Lösung herrschen muss.

Es wurde zwar in Schubert/Williams 2010 nur mittelständische Unternehmen untersucht, die externe Softwarepartner einsetzen. Allerdings könnten sich die Ergebnisse verallgemeinern lassen, in dem drei Rollen identifiziert werden: Management, Anwender und IT Verantwortlichen. Bei einem kleineren Unternehmen wird Management und Anwender in Personalunion realisiert, und der IT-Verantwortliche ist extern, bei größeren Unternehmen würden alle drei Rollen in Form von Abteilungen realisiert. Nachdem das Management die strategische Entscheidung für ein IT Projekt getroffen hat, müssen Anwender und IT Verantwortliche eine Form der Umsetzung finden, die beiderseitiges Vertrauen hat und den Rahmen der Ressourcen nicht sprengt. Diese drei Rollen müssten dann so zusammenwirken, dass das Top Management hinter der Projektentscheidung steht, damit auch willens ist genügend Ressourcen bereitzustellen. Außerdem müssen Anwender und IT Verantwortliche eine Lösung entwickeln, von deren Zukunftsfähigkeit beide überzeugt sind.

Zusammenfassend ergibt sich:

Projektbezogener Erfolgsfaktor	Erläuterung
Top Management Commitment	Wenn die Entscheider nicht hinter dem Projekt stehen, kann es nicht umgesetzt werden.
Ressourcenbereitstellung	Wenn die notwendigen Ressourcen nicht verfügbar sind, ist das Projekt ebenfalls zum Scheitern verurteilt.
Vertrauen in die Zukunftsfähigkeit der Lösung	Die Fachabteilungen und die IT Abteilung müssen überzeugt sein, dass das geplante Arbeiten muss für die absehbare Zukunft Bestand haben.

Tabelle 2.3: projektbezogene Erfolgsfaktoren

Bei unternehmensübergreifenden Problemstellungen müssen natürlich die Entscheider der Unternehmenspartner, sowie die jeweiligen Fachabteilungen und IT Verantwortlichen zusammenwirken.

3 Dienstleistungsnetzwerke

Die Definition von Dienstleistungen kann auf vielfältige Art und Weise geschehen. Am einfachsten scheint es zu sein, Dienstleistungen von gewöhnlichen Gütern abzugrenzen. So können gewöhnliche Güter weiterverkauft werden, während Dienstleistungen nach Erledigung nur neu erbracht werden können. Nach dieser Beschreibung sind Dienstleistungen nichts anderes als die Erbringung von Arbeitsleistung für einen Kunden. Im Gegensatz zu Gütern, die ihre Funktion und Eigenschaften schon haben, bevor der Kunde diese kauft, muss bei Dienstleistungen der Kunde bestimmen, was er wie erledigt haben will. Außerdem muss er das Material stellen, aus dem die Dienstleister etwas hervorbringen sollen (Vgl. Ahlert/Blaich/Evanschitzky 2003, S. 45).

Der Begriff des Netzwerks kann am einfachsten mathematisch als gleichzeitige Beziehung zwischen mindestens drei Teilnehmern definiert werden. Für die hier interessierenden Marktbeziehungen gilt, dass die Teilnehmer in die Gruppen Kunden und Produzenten fallen und in den interessanten Beziehungen immer Teilnehmer beider Gruppen sein müssen. Damit gibt es die minimalen Beziehungen, ein Kunde und zwei Produzenten, bzw. zwei Kunden und ein Produzent. Ein minimales Netzwerkprodukt ist damit etwas, das entweder zwei Produzenten hat und einen Kunden oder einen Produzenten und zwei Kunden.

Bei einem Dienstleistungsnetzwerk fragt nach dieser Definition ein Kunde eine Dienstleistung ab, die mehrere Dienstleister gemeinsam erbringen müssen, oder ein Dienstleister erbringt eine Dienstleistung in der Regel immer für mehrere Kunden gleichzeitig. Diese Gleichbehandlung zwischen Kunden und Dienstleister in dieser Definition führt dazu, dass z.B. ein Werksbusfahrt ein Dienstleistungsnetzwerk ist, da ja in dem Bus in der Regel immer mehrere Arbeiter befördert werden!

Diese Definition von Dienstleistungsnetzwerk geht über die in der Grundlagenliteratur angegebene Definition hinaus. Dort wird ein Dienstleistungsnetzwerk definiert als Dienstleister, die Dienstleistungen regelmäßig zusammen erbringen. Dabei haben sie ihre Zusammenarbeit dauerhaft angelegt und nicht nur marktmässig zusammengeführt (Vgl. Ahlert/Blaich/Evanschitzky 2003, S. 52). Ein minimales Netzwerk besteht hier aus zwei Dienstleistern und einem Kunden. Hier werden dann zum Beispiele solcher Dienstleistungsnetzwerke wie Tourismusdienstleistungen, in denen ein Tourist von verschiedenen Dienstleistern mit Hotel, Gaststätten und Touren versorgt wird, oder auch Finanzdienstleistungen, in

denen Kredite und Versicherungen unterschiedlicher Anbieter miteinander gekoppelt werden.

Im Konsumentenumfeld mag diese Einschränkung sinnvoll sein, da interessante Fälle von einem Dienstleister und zwei Kunden nicht vorstellbar sind. Allerdings gibt es im Bereich der Logistikdienstleistungen den Fall, dass Logistikdienstleister vernetzt mehrere Industriekunden gleichzeitig versorgen (Vgl. Lönngren/Kolbe/Rosenkranz 2008, S. 728). Hier ist die Zusammenarbeit auf Dauer angelegt und es handelt sich in jedem Fall um ein Netzwerk mit Dienstleistung.

Die für das Thema dieser Arbeit interessanteren Dienstleistungsnetzwerke befinden sich im Bereich der industriellen Dienstleistungen, in denen der Kunde selbst auch ein Unternehmen ist. Da diese Kunden in der Regel über IT Systeme verfügen, sind hier die Fragestellungen bzgl. Erfolg und Mißerfolg von Vernetzungen eher untersucht. Bevor eine Typologie der Netzwerke hergeleitet werden kann, erscheint es notwendig, Beispiele von in der verwendeten Literatur untersuchten Dienstleistungsnetzwerken vorzustellen.

3.1 Beispiele

Die folgenden Ausführungen dieser Arbeit bauen auf industriellen Netzwerken auf, die nicht hauptsächlich industrielle Partner als Kunden der Dienstleistung haben.

Die Geschäftsprozesse in Banken und Versicherungen sind in erster Linie informationsbasiert und großteils standardisiert. So müssen z.B. die Angaben bei Schecks überprüft werden bevor eine Zahlung angewiesen werden kann. Ähnliches gilt für Risiko- und Bonitätsanalysen. Hier ist eine Vernetzung möglich, in der einzelne Geschäftsprozesse ausgelagert und evtl. sogar weltweit verteilt werden (Vgl. Braun/Winter 2005, S. 352).

Die Zusammenarbeit bei der Entwicklung und Produktion zwischen Autoherstellern und Zulieferern weist bzgl. der Abwicklung von Entwicklungsprojekten Dienstleistungsmerkmale auf. Ein Automobilhersteller gibt die Entwicklung von Komponenten seinen Zulieferern in Auftrag. Diese müssen dann zusammenarbeiten und kooperieren (Bartlog/Boy 2010, S. 1571).

Bei industriellen Dienstleistungen werden Anlagen in aller Regel von unterschiedlichen Büros geplant und die Komponenten von unterschiedlichen Herstellern bezogen. Für die Durchführung von Störfallarbeiten entsteht ein mehr oder weniger

formales Dienstleistungsnetzwerk von Kunde, Ingenieur und Komponentenlieferanten (vgl. Beverungen et al. 2008, S. 736).

Im Medizinbereich können medizinische Versorgungszentren, in denen unterschiedliche Fachärzte, Hausärzte und andere medizinische Dienstleister wie Physiotherapeuten und Apotheker zusammenarbeiten, als Dienstleistungsnetzwerke aufgefasst werden (Pütz et al. 2010, S. 145).

Die Entwicklung von Konsumgütern erfolgt häufig in Dienstleistungsnetzwerken, in denen virtuelle Prototypen ausgetauscht werden. Virtuelle Prototypen sind Dateien mit CAD Spezifikationen, so dass der Empfänger ein Modell des Prototypen selbst erstellen oder auch visualisieren kann. Besondere Bedeutung haben diese Prototypen im Bereich virtueller Produktabbildungen. Von besonderer Bedeutung in diesen Netzwerken ist die sichere Übermittlung nicht unbeträchtlicher Datenvolumen (Vgl. Zagel/Löffler 2010, S. 1265).

3.2 Typen von Dienstleistungsnetzwerken

Wie oben dargestellt, gibt es die unterschiedlichsten Arten von Dienstleistungsnetzwerken. Für weitere Untersuchungen ist es unerlässlich diese Netzwerke in Kategorien bzw. Typen einzuteilen.

Ein Ansatz wäre zu unterscheiden, ob das Netzwerk von einem Unternehmen initiiert wurde oder es ob es ein Netzwerk von gleichrangigen Unternehmen ist (vgl. Stauss/Bruhns 2003, S. 9). Im ersten Fall, einem *hierarchischen* Netzwerk, gruppieren sich andere Dienstleister dann um einen Hauptdienstleister herum und liefern Teildienstleistungen zu. Ein typisches Beispiel wäre hier eine Sparkasse, die einen Baukredit mit dem Abschluss einer Lebensversicherung eines Tochterunternehmens sowie eines Bausparvertrages verknüpft. Im zweiten Fall, einem *heterarchischen* Netzwerk, haben alle Dienstleister einen Kontakt mit dem Kunden und vermitteln zusätzliche Leistungen an Dienstleister im Netzwerk. Typisch wäre hier eine Gemeinschaft von Handwerkern, die Umbau- oder Renovierungsmaßnahmen gemeinsam durchführen, so dass derjenige, der zuerst angesprochen wurde, die Teilleistungen vermittelt. Ob allerdings eine Vernetzung auf Ebene der IT-Architekturen stattfindet oder nicht, wird in diesen Kategorien (hierarchisch/heterarchisch) nicht unterschieden. Deswegen sind diese Kategorien für in die Zielsetzung dieser Arbeit nicht hilfreich.

Genausowenig ist eine Unterscheidung, ob das Netzwerk vom Anbieter oder vom Kunden initiiert wird (vgl. Stauss/Bruhns 2003, S. 10), hilfreich. Beide oben genannten Beispiele sind dem ersten Fall, einen *Anbieter initiierten Netzwerk*, zu

zuordnen. Ein Beispiel für den zweiten Fall, einem *Kunden initiierten Netzwerk*, ist ein Urlaubsort, in dem Touristen Dienstleistungen von unterschiedlichen Anbietern nachfragen. Das Netzwerk der Dienstleister entsteht dann durch die gemeinsame Nachfrage der Touristen. Aber auch bei dieser Unterscheidung wird keine Aussage bzgl. der verwendeten IT-Architektur gemacht.

Erfolgversprechender ist das Ergebnis einer Untersuchung von 14 Fallstudien betreffs der IT-Architektur von Unternehmensnetzwerken im allgemeinen. Untersucht wurden unterschiedlichste Unternehmen, von Detailhändlern über Buchhaltungsdienstleister zu Fertigungsketten. Je nachdem, ob eine gemeinsame Systembasis genutzt wurde, oder nicht (Schubert 2008 S. 892), wurden fünf Szenarien unterschieden:

1. Parallele Nutzung *unterschiedlicher* Informationssysteme, manuelle externe Systemzugriffe. Bsp. Chocolat Frey: Portal für Lieferanten
2. Parallele Nutzung *unterschiedlicher* Informationssysteme, Austausch *strukturierter Geschäftsdokumente* mit direkter Anbindung. Bsp. Intersport: Austausch von Franchisenehmern mit Mutterkonzern
3. Parallele Nutzung *unterschiedlicher* Informationssysteme, Austausch *strukturierter Geschäftsdokumente* mit Anbindung über einen Intermediär. Bsp. Edeka: zwischen Händler und Lieferanten wird ein IT-Dienstleister geschaltet
4. Gemeinsame Nutzung eines eigenen, *zentralen* ERP-Systems. Bsp. Vintothek Brancaia: das ERP System wird Aussendienstmitarbeitenden per VPN zur Verfügung gestellt.
5. Gemeinsame Nutzung des *zentralen* Systems eines Dienstleisters (Intermediärs). Bsp. Verein Swisscanto Asset Management: Verwaltungsgesellschaften und Leistungserbringer haben Zugriff auf eine zentral betriebene Plattform.

Für die Zielsetzung dieser Arbeit ist die Aussage, ob ein Intermediär eingeschaltet wurde oder nicht, genauso unerheblich, wie die, ob alle Teilnehmer dieselbe IT-Architektur haben. Der Aufbau eines Netzwerkes ist selbstverständlich einfacher, wenn alle Teilnehmer eine gemeinsame Basis haben bzw. wenn ein IT-Dienstleister den gesamten Vernetzungsaufwand auf sich nimmt.

Allerdings ist die Unterscheidung zwischen Austausch von strukturierten und unstrukturierten Geschäftsdokumenten hilfreich. Wenn strukturierte Dokumente

ausgetauscht werden, kann die Vernetzung enger sein, wenn diese nicht strukturiert sind, muss eine Arbeitskraft diese verarbeiten, die Vernetzung ist dann nicht so eng. Eine andere Unterscheidung kann aus den Fallstudien ebenfalls abgeleitet werden. Nämlich die der Verarbeitungszeit einer Nachricht. Soll sie sofort verarbeitet werden, in *Echtzeit* also, oder erst nach einer *Wartezeit*, wenn der Empfänger Zeit und Interesse hat, diese zu verarbeiten. Eine Unterteilung von Netzwerken in solches Schema würde dann vier Quadranten ergeben:

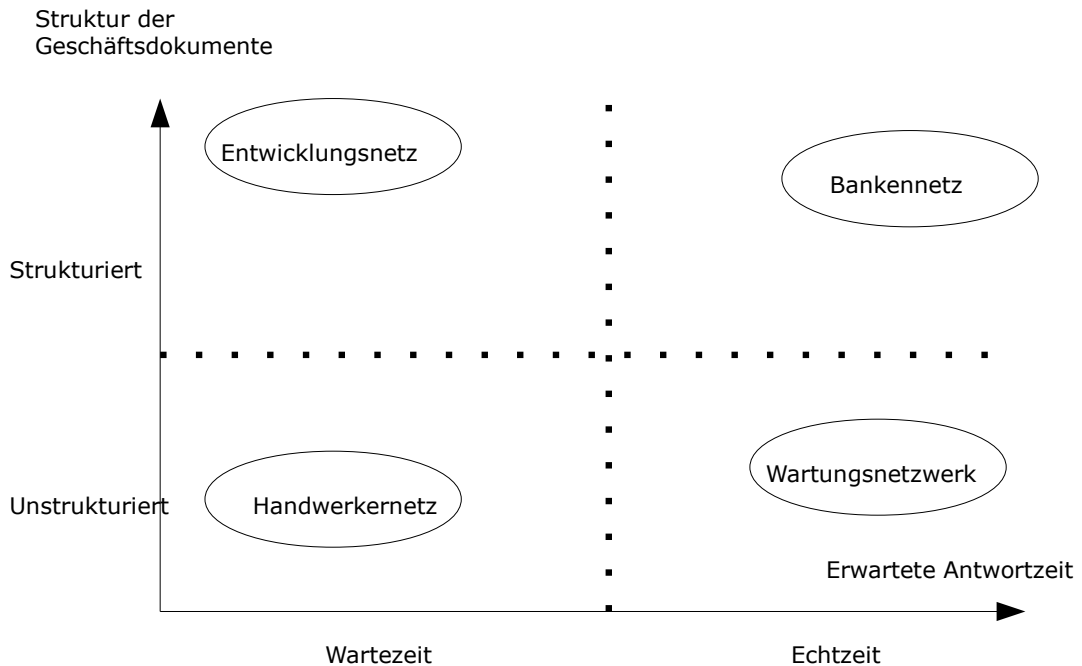


Abbildung 3.1: Typologie nach Datenformat und erwarteter Antwortzeit

Im Quadrant unstrukturiert/Wartezeit befindet sich z.B. das typische Handwerkernetzwerk. Der Austausch der Informationen über einen Auftrag findet in Form von Telefonaten und Gesprächen statt, nachdem der Empfänger geprüft hat, ob er überhaupt an dem Gesamtauftrag teilnehmen kann (will), wird eine Nachfrage beantwortet. Ein Einsatz von IT zur Verbesserung der Kommunikation wird nicht benötigt und, auch wenn angeboten, nicht genutzt (Vgl. Lönngren/Kolbe/Rosenkranz 2008, S. 731)

Bei strukturiert/Wartezeit befinden sich Netze, in denen generelle Entwicklung stattfindet, so z.B. Automobilkomponenten oder Konsumgüter oder auch Bauplanungen. Hier werden Arbeitsergebnisse in strukturierter Form ausgetauscht und weitergegeben. Ebenso kann die Planung und Verfolgung von Projekten in einheitlicher Form erfolgen. Aber auch wenn Kunden ihrem Logistikdienstleister die

geplanten Absatz- und Produktionszahlen bekannt machen, fällt dies in diese Kategorie (Vgl. Lönngren/Kolbe/Rosenkranz 2008, S. 728).

Die Wartungsfälle im Anlagenbau befinden sich im Quadranten unstrukturiert/Echtzeit, da ja eine Störfallmeldung sofort beantwortet werden muss. Der Störfall selbst kann aber nur in dem Sinne unstrukturiert gemeldet werden, dass ein Mensch die Meldung interpretieren und bewerten muss, auch wenn die Daten in Form von Tabellen und Logdateien übertragen werden. Wenn ja eine automatische Diagnose dieser Daten möglich wäre, wäre es ja kein Störfall.

Im Bereich des Quadranten strukturiert/Echtzeit befindet sich die komplette Auslagerung von ganzen Geschäftsprozessen, das sogenannte „business process outsourcing“. Dies findet sich in Deutschland hauptsächlich im Bankenbereich. Hier ist diese Auslagerung von Geschäftsprozessen am weitesten fortgeschritten (Vgl. Schömburg/Breitner 2010 S. 1262).

4 Designprinzipien und Erfolgsfaktoren von Applikationsarchitekturen im Kontext von Dienstleistungsnetzwerken

Die oben gefundenen Designprinzipien und Erfolgsfaktoren von Applikationsarchitekturen bezogen auf sich auf jeweils ein einzelnes Unternehmen. Zu klären ist jetzt die Bedeutung dieser Prinzipien und Faktoren, wenn ein Dienstleister sich mit anderen Dienstleistern vernetzen will bzw. sich einem Dienstleistungsnetzwerk anschließen möchte. Wie dargestellt lassen sich aus architektonischer Sicht Dienstleistungsnetzwerke entlang der zwei Achsen „Strukturiertheit der ausgetauschten Geschäftsdokumente“ und „erwartetes Antwortverhalten des Empfängers“ positionieren, so dass ein einzelnes Netzwerk mehr oder weniger ausgeprägt in einen der Quadranten strukturiert/Echtzeit, strukturiert/Wartezeit, unstrukturiert/Echtzeit oder unstrukturiert/Wartezeit fällt.

In jedem Fall werden Applikationsarchitekturen nur dann interessant, wenn strukturierte Daten ausgetauscht werden. Es können zwar auch unstrukturierte Geschäftsdokumente maschinell interpretiert werden, besonders einfach scheinen hier Excel Tabellen zu sein, allerdings bedeutet dies dann, dass der Empfänger die Daten für den Absender in eine strukturierte Form bringt. Die Vernetzung insgesamt verhält sich dann so, wie als wenn strukturierte Geschäftsdokumente ausgetauscht worden wären.

Eine Vernetzung unter Verwendung unstrukturierter Dokumente bedeutet aber nicht, dass die Zusammenarbeit der Dienstleister nur informell und flüchtig ist. Sondern nur, dass die Geschäftsfelder der beteiligten Unternehmen soweit von einander getrennt sind, dass für die Übertragungen von Informationen das Urteilsvermögen der Sachbearbeiter notwendig ist. Insofern ist ein Eintritt eines Unternehmens in ein Netzwerk mit Austausch unstrukturierter Dokumente so zu betrachten, wie eine normale Änderung oder Anpassung der Geschäftsprozesse auch.

Zu untersuchen sind folglich nur die Fälle „Antwort nach Wartezeit“ oder „Reaktion in Echtzeit“ bei Austausch von strukturierten Geschäftsdokumenten.

Zuvor sollte aber überlegt werden, ob wirklich alle gefundenen Prinzipien und Faktoren für den Vernetzungsfall interessant sind. Vielleicht sind einige dabei, die ausgeschlossen werden können, weil sie sich ja nicht auf die Vernetzung auswirken können.

Bei den Designprinzipien soll mit der Einhaltung des Prinzips „Verwendung lose gekoppelter Servicedomänen“ eine Strukturierung der Applikationslandschaft erhöht werden. Hierdurch soll diese dann leichter verwaltet und gewartet werden können. Für Vernetzungen mit anderen Unternehmen ist dies allerdings nur mittelbar von Bedeutung, etwa wenn hierdurch die allgemeine Agilität erhöht wird. Deswegen kann dieses Designprinzip vernachlässigt werden.

Von den technischen Erfolgsfaktoren sind die Faktoren Sicherheit, Zuverlässigkeit und Effizienz eigentlich nicht für Vernetzungen interessant. Zwar werden durch die Offenlegung von Schnittstellen und Systemzugängen die Anforderungen an die Sicherheit erhöht. Allerdings steht hier nur die Entscheidung an, welche zusätzlichen Sicherheitsmechanismen verwendet werden sollen. Mithin kann dieser Faktor für die weiteren Ausführungen vernachlässigt werden. Ein Service muss zuverlässig und effizient arbeiten, egal ob er einem Geschäftspartner angeboten wurde oder nicht. Anders ist die Situation bei den Faktoren Agilität und Portierbarkeit. Durch den Eintritt in ein Netzwerk müssen ja ggf. Umstellungen und Erweiterungen vorgenommen werden, weswegen Agilität gefordert sein sollte. Die Portierbarkeit muss bewiesen werden, wenn ggf. das System oder Teile davon portiert werden müssen, falls Zutritt zu einem Netzwerk dies erforderlich macht.

Bei den projektbezogene Erfolgsfaktoren sollten allerdings alle betrachtet werden, da ja der Eintritt in ein Unternehmensnetzwerk bzw. die Etablierung eines solchen ein Projekt darstellt. In diesem Zusammenhang ist nicht nur der Entschluss des eigenen Top-Managements gefordert, sondern auch die Zustimmung vom Management der Partner, ebenso müssen die IT-Abteilungen bzw. IT-Partner die Vernetzung unterstützen und gemeinsam mit der eigenen Fachabteilung und den Fachabteilungen der Partner Vertrauen in die Zukunftsfähigkeit der Lösung aufbauen. Selbstverständlich funktioniert das Vorhaben nur dann, wenn die ausreichend Ressourcen zur Umsetzung bereitgestellt werden.

Zusammenfassend sind folgende Punkte, je nach Art der Vernetzung zu untersuchen:

- Designprinzipien
 - Service als Vertrag zwischen Nutzer und IT.
 - Offene Standards für Datenprotokolle und -formate.
 - Geschäftsprozess orientiertes Design einfacher Services.

-
- Technische Erfolgsfaktoren
 - Agilität
 - Portierbarkeit
 - Projektbezogene Erfolgsfaktoren
 - Top Management Commitment
 - Ressourcenbereitstellung
 - Vertrauen in die Zukunftsfähigkeit der Lösung

4.1 Austausch von strukturierten Geschäftsdokumenten mit Wartezeit

Mit solchen Dienstleistungsnetzwerken werden hauptsächlich Entwicklungsaufgaben realisiert, seien es Projekte im Anlagenbau, bei der Entwicklung von Komponenten der Automobile der nächsten Generation oder auch von Modeartikeln. Hier werden hauptsächlich Angebote, Termin- und Zustandsinformationen sowie eben die Arbeitsergebnisse übertragen und kontrolliert.

Nun gilt es darzustellen, zum einen wie mehrere Dienstleister ein solches Netzwerk aufbauen und zum anderen wie ein Dienstleister sich einem bestehenden Netzwerk anschließt. Es wird immer vorausgesetzt, dass der Dienstleister über eine bestehende Applikationsarchitektur verfügt, deren Design unter Beachtung der oben genannten Prinzipien erfolgte und deren Realisierung die genannten Erfolgsfaktoren aufweist.

Die Bedeutung des Designprinzips „Offene Standards für Datenprotokolle und -formate“ liegt in diesem Zusammenhang auf der Hand. Wenn alle Beteiligten eines Netzwerkes dieselben Datenprotokolle und -formate verwenden würden, wäre eine Vernetzung ohne Probleme machbar. Die Protokollebene ist in diesem Zusammenhang nicht so sehr interessant, da die Geschäftsdokumente ja auch verschlüsselt per eMail übertragen werden und dann auf Empfängerseite entschlüsselt und in das jeweilige System eingespeist werden könnten. Für die Dokumente selbst kann es durchaus konkurrierende Standards geben, so werden z.B. bei den Office Werkzeugen von Microsoft und OpenOffice die Dokumente in zum Teil unterschiedlichem Format erzeugt.

Bei Etablierung eines Netzwerkes müssen die Partner diesen Standard festlegen, falls ein Dienstleister einem Netzwerk beitrifft, muss er den vorgegebenen Standard einhalten. In jedem Fall gilt: wenn ein weitverbreiteter, offener Standard

eingehalten wurde, ist die Chance groß einen Datenumsetzer in einen anderen Standard zu finden!

Die Designprinzipien „Service als Vertrag zwischen Nutzer und IT“ und „Geschäftsprozess orientiertes Design einfacher Services“ sind in diesem Zusammenhang eher von untergeordneter Bedeutung. Zwar müssen die ausgetauschten Geschäftsdokumente verarbeitet werden und hierzu werden dann auch services eingesetzt. Aber es ist für den Service unerheblich ob das Dokument intern erstellt wurde oder ausgetauscht wurde.

Die Bedeutung des Erfolgsfaktors „Agilität“ scheint offensichtlich, da die Anpassungskosten, die durch die Vernetzung entstehen um so größer sind, je weniger agil die Applikationsarchitektur des IT-System ist. Es muss allerdings eigentlich nur der Versand und der Empfang der Dokumente implementiert werden. Wenn die Geschäftsprozesse des Unternehmens durch die Vernetzung nicht geändert werden, sind die Anforderungen an die Agilität gar nicht so groß.

Falls ein Datenformat verwendet werden muss, das das aktuelle System nicht unterstützt, müssen Teile des Systems portiert werden. Dann wird der Erfolgsfaktor „Portabilität“ wichtig, nur wird jetzt der Zwang zur Portierung nicht von den Hardware- und Systemsoftwarezyklen bestimmt, sondern von einer Änderung in den Datenformaten und -protokollen.

Bezüglich der projektbezogenen Erfolgsfaktoren zeigen sich deutliche Unterschiede, je nachdem ob es sich um einen Anschluss an ein bestehendes Netzwerk handelt, oder ob es sich um die Gründung eines Netzwerkes handelt. Im Fall des Beitritts sind die Punkte „Top Management Commitment“, „Ressourcenbereitstellung“ und „Vertrauen in die Zukunftsfähigkeit der Lösung“ unternehmensintern zu beachten. Das Management muss wollen, die Kosten für die Umstellung müssen bereitgestellt werden und die Mitarbeiter an den Schnittstellen zu anderen Unternehmen müssen von der Perspektive überzeugt werden.

Anders liegt der Fall bei der Gründung eines Netzwerkes. Wenn es dort keinen mächtigen Hauptpartner gibt, der seine Vorstellungen durchsetzt, muss zunächst eine Willensbildung bei den Managern der wichtigen Partner stattfinden. Die Kosten der gegebenenfalls notwendigen Anpassungen sind von dem gewählten Dokumentenstandard abhängig. Das technische Know-how liegt jedoch normalerweise in den IT-Abteilungen der Partner. Aus diesen IT-Abteilungen werden in Vorbereitung einer Entscheidungsfindung Arbeitsgruppen gebildet, die entsprechende Standards dem Management vorschlägt. In diesem Zusammenhang liegt

asymmetrische Information vor, da ja jeder Partner nur seine eigenen Anpassungskosten kennt. Der Erfolg solcher Verhandlungen setzt dann entweder großes Vertrauen untereinander und eine Überzeugung von der Zukunftsfähigkeit des Ansatzes voraus.

Wie aufwändig sich die Etablierung solch einer Zusammenarbeit gestalten kann, zeigt das Projekt „Collaborative Project Management“, das von Daimler und BMW gemeinsam mit den Zulieferern ZF, Karmann und Kieper initiiert wurde. In dem Projekt sollte eine Umgebung zur unternehmensübergreifenden Koordination von Entwicklungsarbeiten definiert und implementiert werden.

Das Projekt wurde 2007 gestartet, 2008 gab es erste Vorschläge für ein XML-Schema, an denen eine Arbeitsgruppe immer noch arbeitet (vgl. Bartlog/Boy 2010, S. 1572). Obwohl die Daten eigentlich einfacher Art sind, eben nur Termine, Arbeitspunkte, Probleme und Aufgaben in unterschiedlicher Struktur, war nach zwei Jahren noch immer keine Implementierung vorzuweisen. Trotzdem haben in einer Umfrage zur Bedeutung eines Standards für kollaboratives Projektmanagement unter Projektleitern unterschiedlicher Unternehmen über 80% diesem eine wichtige Bedeutung zugesprochen.

Obige Überlegungen können wie folgt zusammengefasst werden:

- Die Verwendung *offener Standards für Datenprotokolle und -formate* ist normalerweise vorteilhaft, da Partner ggf. eben diese auch verwenden
- Die *Agilität* der Applikationsarchitektur ist nicht besonders gefordert
- Die *Portierbarkeit* kann wichtig werden, wenn die Teilnahme am Netzwerk andere Datenformate verlangt
- Das Top Management Commitment von allen Partnern wird verlangt.
- Die *Bereitstellung von Ressourcen* bestimmt die Verhandlungen bei Einrichtung eines Netzwerks.
- Das *Vertrauen in die Zukunftsfähigkeit der Lösung* muss von allen Beteiligten Partner und deren IT-Partnern gegeben sein.

4.2 Direkte Verarbeitung strukturierter Geschäftsdokumente

Bei der direkten Verarbeitung von strukturierten Geschäftsdokumenten wird auf Empfängerseite ein Geschäftsprozess angestoßen. Dies kann auf der einen Seite ein ausgelagerter Geschäftsprozess sein, nämlich dann, wenn das Resultat der

Verarbeitung, ebenfalls als strukturiertes Geschäftsdokument, wieder abgefragt wird. Es kann aber auch eine Zulieferung im Rahmen einer Geschäftsverbindung sein, dann werden die Aufträge vermittelt. Im ersten Fall wäre zum Beispiel eine Risikoanalyse oder auch eine Auswertung medizinischer Daten, im zweiten Fall wäre an eine Vermittlung von Reisen oder Versicherungen zu denken.

Ein gut untersuchter Bereich bezüglich dieser Art der Vernetzung ist der Bankenbereich. Hier herrscht ein großer Marktdruck zur Vernetzung. So bestehen die Aufgaben im wesentlichen in der Verarbeitung von Informationen. Es gilt diese Informationen in Standardabläufen zu überprüfen und zu bewerten. Somit gibt es ein großes Potential, dass sich Unternehmen auf einige Abläufe konzentrieren und andere Abläufe auslagern und mieten (vgl. Baskerville et al. 2005, S. 764). Aber auch im medizinischen Bereich gibt es einen Druck zur Zusammenarbeit bei der Bewertung von Patientendaten (vgl. Pütz et al. 2010, S. 145).

Zu untersuchen ist jetzt wieder die Fragestellung, welche Designprinzipien und Erfolgsfaktoren von Unternehmen bei Eintritt oder Gründung eines Verbundes zur gemeinsamen Verarbeitung von Geschäftsprozessen von besonderer Bedeutung sind. Im Unterschied zur vorherigen Betrachtung des Austauschs von strukturierten Geschäftsdokumenten, steht hier die Auslagerung von Geschäftsprozessen im Vordergrund. Wieder sei angenommen, dass die Unternehmen bereits eine Applikationsarchitektur verwenden, die den genannten Designprinzipien und Erfolgsfaktoren genügt.

Bei den Designprinzipien sind „Service als Vertrag zwischen Nutzer und IT“ und „Geschäftsprozess orientiertes Design einfacher Services“ von Bedeutung. Mit dem Eintritt eines Unternehmens in einen Verbund werden die Geschäftsprozesse eingestellt, die von dem Unternehmen nicht effizient ausgeführt werden, während sich auf die Geschäftsprozesse konzentriert wird, die effizient ausgeführt werden. Wenn für die Bedienung eines Services noch Spezialwissen nötig ist, also „Service als Vertrag“ nicht beachtet wurde, können die Sachbearbeiter eine höhere Entlohnung durchsetzen, da bei einem Wechsel neue Kräfte erst wieder anzulernen sind. Der so unterstützte Geschäftsprozess ist damit dann zu teuer. Ebenso ist ein Geschäftsprozess effizienter je mehr er durch einen Service unterstützt wird, der sich an eben dem Geschäftsprozess orientiert, also „Geschäftsprozess orientiertes Design einfacher Services“ erfüllt.

Das Designprinzip „Offene Standards für Datenprotokolle und -formate“ ist zwar wichtig, aber nicht so bedeutend. Zwar müssen auch hier Daten ausgetauscht werden, und es ist vorteilhaft hier über Standards zu verfügen. Aber die eventu-

ellen Anpassungskosten können mit den Einsparpotentialen gegengerechnet werden.

Die „Agilität“ der Applikationsarchitektur ist nicht so sehr gefordert, da mit der Konzentration auf Geschäftsprozesse eine Straffung der Architektur einhergeht und weniger eine Umstellung oder Änderung. So dass im Endeffekt eher weniger Geschäftsprozesse unterstützt werden.

Auf der anderen Seite ist die „Portierbarkeit“ der Applikationsarchitektur gefordert, da mit der Konzentration auf weniger Geschäftsprozesse feststeht, dass eben diese auch in Zukunft unterstützt werden müssen. Also müssen die unterstützenden Applikationen auf zukünftigen Hard- und Systemsoftware portiert werden können.

Bei den projektorientierten Erfolgsfaktoren muss beachtet werden, dass mit der Konzentration auf effiziente Geschäftsprozesse ein Abbau an Fachabteilung und IT-Abteilung einhergeht. Eine solche Entscheidung für einen Abbau kann vom Top-Management leichter umgesetzt werden, als eine Entscheidung für einen Umbau, da die Fach- und IT-Abteilungen ihre Informationsvorsprünge nutzen können, um die Umbauentscheidungen zu beeinflussen. Ebenso ist es mit der Entscheidung vermehrt in die Geschäftsprozesse zu investieren, die anderen Unternehmen zur Nutzung angeboten werden. Auch hier ist die Entscheidung leichter umzusetzen, da die Fachabteilungen nur aufgestockt werden brauchen.

Allerdings scheint eine gemeinsame, gleichzeitige Willensbildung mehrere Akteure mehr oder weniger zentral geplant ein Unternehmensverbund zu konstituieren schwierig zu sein. Hier können in den Vorbereitungsgesprächen alle IT-Abteilungen und Fachabteilungen der beteiligten Unternehmen gemeinsam die Vorgabe ihres Top Management hintertreiben. So geschehen bei dem fehlgeschlagenen Versuch der Irish League of Credit Unions (in etwa Volks- und Raiffeisenbanken) ihre IT auf eine gemeinsame Netzwerkbasis zu stellen. Das Projekt wurde 1998 angefangen, ist nie über die Pilotphase hinausgekommen und wurde 2001 eingestellt. Es sollte versucht werden 450 Computerinstallationen zu vernetzen. Auf diesen Computern liefen 33 verschiedene Softwaresysteme, die von 26 IT-Partnern betrieben wurden (Vgl. Mangan/Kelly 2003, S. 1224). Interessant ist, dass die Einstellungsgründe vorgeblich finanzieller Natur waren (Vgl. Mangan/Kelly 2003, S. 1225), während eine spätere Untersuchung zeigte, dass es eher kulturelle Gründe waren, die zu einem Widerstand der kleinen Genossenschaften gegen eine Aufgabe ihrer traditionellen Unabhängigkeit führten (Vgl. Mangan/Kelly 2003, S. 1227).

Auf der anderen Seite gibt es erfolgreiche Auslagerungen in eben dem Bereich finanzieller Dienstleistungen, die eher inkrementeller Natur sind. So sieht die Postbank ihre Hauptkompetenz in der Behandlung von finanziellen Transaktionen und bietet eben diese Dienstleistung auch extern an. Die Deutsche Bank und die Commerzbank haben die Bearbeitung ihres Zahlungsverkehrs seit 2004 an die Postbank ausgelagert (Vgl. Braun/Winter 2005, S. 360).

Die Beispiele deuten auf die überragende Bedeutung des Faktors „Vertrauen in die Zukunftsfähigkeit der Lösung“ hin. Wenn die Fachabteilung gemeinsam mit der IT-Abteilung bzw. dem IT-Partner von der Zukunftsfähigkeit überzeugt ist, kann das Management diese Prozesse auch extern anbieten. Partner können dann dieses Angebot annehmen und eine Vernetzung entsteht. Der Faktor „Top Management Commitment“ ist nicht so ausschlaggebend, da eine Entscheidung der Top Manager alleine zunächst eine Entscheidung gegen einige, noch nicht bestimmte Fachabteilungen ist. Hierdurch werden dann Widerstände organisiert. Die Bereitstellung von Ressourcen ist ebenso von untergeordneter Bedeutung, da durch die Auslagerung von Geschäftsfeldern Ausgaben eingespart werden. Auf der anderen Seite könne die Aufwände für einen Ausbau den Abnehmern in Rechnung gestellt werden.

Zusammenfassend können folgende Punkte angemerkt werden:

- Geschäftsprozesse, die durch einen *Service, der als Vertrag zwischen Nutzer und IT realisiert ist*, unterstützt werden, sind kostengünstiger, da die Anwender kein Spezial know-how benötigen.
- Die Verwendung *offener Standards für Datenprotokolle und -formate* ist nicht so wichtig.
- Ein *Geschäftsprozess orientiertes Design einfacher Services* senkt ebenfalls die Kosten der Geschäftsprozesse.
- Die *Agilität* ist nicht gefordert, da Geschäftsprozesse eingestellt oder exportiert werden.
- Die *Portierbarkeit* der exportierten Geschäftsprozesse muss gewahrt sein, da dort langfristige Verpflichtungen entstehen.

-
- Ein Top Management Commitment kann sich über die bestehenden Realitäten nicht hinwegsetzen.
 - Die *Ressourcenbereitstellung* muss durch Einsparungen wettgemacht werden.
 - Ein Vertrauen in die Zukunftsfähigkeit der Lösung der exportierten Geschäftsprozesse scheint das wichtigste Kriterium zu sein.

5 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde untersucht, welche der Designfaktoren und Erfolgsfaktoren von Applikationsarchitekturen im Kontext Dienstleistungsnetzwerke besonders wichtig sind. Zunächst wurde der Begriff Applikationsarchitektur genauer definiert als die Struktur der Anwendungssoftware zwischen Präsentationsschicht und Datenbankschicht. Von außen zugreifbar sind Services und strukturierte Geschäftsdokumente.

Es wurden Designprinzipien gefunden, sowie technische und projektbezogene Erfolgsfaktoren von Applikationsarchitekturen. Mit den Designprinzipien soll das Design, also die fachliche Spezifikation der Anwendungssoftware erfolgen, die Einhaltung der technischen Erfolgsfaktoren sollte zu einer erfolgreichen Verwendung der Architektur sorgen, die Beachtung der projektbezogenen Erfolgsfaktoren sollte zu einer erfolgreichen Umsetzung der Architektur führen.

Anschließend wurden Dienstleistungsnetzwerke definiert, in Beispielen untersucht und klassifiziert. Bei der Klassifikation wurde unterschieden, ob strukturierte Geschäftsdokumente ausgetauscht werden oder nicht. Des Weiteren wurde unterschieden, ob die ausgetauschten Daten auf der Empfängerseite sofort verarbeitet wurden (in Echtzeit) oder nicht (in Wartezeit).

Wenn kein Austausch von strukturierten Geschäftsdokumenten erfolgt, ist die Bedeutung für die Applikationsarchitektur zu vernachlässigen.

Wenn ein Austausch vorliegt, wurden zwei Fälle unterschieden, je nachdem wie auf Empfängerseite die Dokumente verarbeitet werden. Bei einer Verarbeitung mit Wartezeit handelt es sich eher um eine Information und Zulieferung, die zu einer weiteren Verarbeitung führt. Hier ist dann wichtig, das Format der Geschäftsdokumente mit den Partnern auszuhandeln bzw. Anpassungen an das ausgehandelte Format umzusetzen.

Im anderen Fall werden die Dokumente direkt und automatisch verarbeitet. Es wird hier also eher ein ganzer Geschäftsprozess ausgelagert. Für den Verbund betrachtet, bedeutet dies, dass jeder der Partner gewinnt, da jeder die Bereiche, die schlecht unterstützt werden, abgibt und die Bereiche, in denen er stark ist, ausbaut.

Die Bewertung der Designprinzipien und Erfolgsfaktoren erfolgte rein argumentativ, da keine eindeutige Datenbasis in Form von Umfrageergebnissen oder Fallstudien vorlag. Gleichwohl gibt es veröffentlichte Anekdoten, die die Überlegungen zu bestätigen scheinen.

Folgende Tabelle (-/- nicht von Belang, + wichtig, 0 unwichtig) fasst die Resultate zusammen:

Designprinzip/Erfolgsfaktor	Wartezeit	Echtzeit
Designprinzipien		
Service als Vertrag zwischen Nutzer und IT.	-/-	+
Offene Standards für Datenprotokolle und -formate.	+	0
Verwendung lose gekoppelter Servicedomänen.	-/-	-/-
Geschäftsprozess orientiertes Design einfacher Services.	-/-	+
Technische Erfolgsfaktoren		
Zuverlässigkeit	-/-	-/-
Agilität	0	0
Portierbarkeit	+	+
Effizienz	-/-	-/-
Sicherheit	-/-	-/-
Projektbezogenen Erfolgsfaktoren		
Top Management Commitment	+	0
Ressourcenbereitstellung	+	0
Vertrauen in die Zukunftsfähigkeit der Lösung	+	+

Tabelle 5.1: Designprinzipien und Erfolgsfaktoren im Kontext Dienstleistungsnetzwerke

Anscheinend ist die Agilität einer Applikationsarchitektur gar nicht so entscheidend für den Erfolg im Kontext Dienstleistungsnetzwerke. Stattdessen ist die Portierbarkeit wichtig, da ja immer auch eine langfristige Verpflichtung zur Erbringung einer Leistung eingegangen wird. Weiter ist das Vertrauen von Fachabteilung und IT Abteilung in die umgesetzte Lösung zu beachten.

Allerdings sind alle Überlegungen letztlich argumentativer und spekulativer Natur, da außer Anekdoten von Konferenzbeiträgen keine eigentliche Datenbasis vorliegt. In einer nachfolgenden Arbeit wäre empirisch zu überprüfen, ob, die durch obige Überlegungen begründeten, Hypothesen wirklich stimmig sind.

Für die weiteren Entwicklungen im Bereich Dienstleistungsnetzwerke sollte beachtet werden, dass diese Art der Vernetzung in der Regel unter gleichrangigen Partnern erfolgt, die unter den informationstechnischen Konzepten als peer-to-peer bekannt ist (Vgl. Walter/Werth 2008, S. 4). Hierdurch sind sehr robuste Netzwerke möglich, die den Ausfall eines Partners durch die Aufnahme neuer Partner ersetzen können. Dadurch ist die mehr oder weniger feste Vernetzung von Unternehmungen in fixen Netzwerken nicht immer unbedingt nötig.

Neuere Konzepte untersuchen genau solche Szenarien. So wird ein „Internet der Dienste“ postuliert (Vgl. Schroth 2008, S. 1373), in dem die Dienste somit ähnlich in einem Gütermarkt gehandelt werden können (Vgl. Karaenke et al. 2008, S. 1383; Blau/Schnizler 2008, S. 1359).

Allerdings bleibt anzumerken, daß diese Möglichkeiten technisch machbar und auch wirtschaftlich effizient sein mögen, aber die Vertrauenskomponente bei der unternehmensübergreifenden Zusammenarbeit nicht zu vernachlässigen ist. So ist der Einsatz von Dienstleister zur Bearbeitung elektronischer Rechnungen sicherlich effizient, aber es setzen sich elektronische Rechnungen am Markt nicht durch (Vgl. Schömburger/Breitner 2010, S. 1261).

Literaturverzeichnis

Aier/Dogan 2004: Aier, Stephan; Dogan, Turgut: Nachhaltig als Gestaltungsziel von Unternehmensarchitekturen. In: Aier, S., Schönherr, M. (Hrsg.): Enterprise ApplicationIntegration – Serviceorientierung und nachhaltige Architekturen, Gito, Berlin, S. 75–122

Ahlert/Blaich/Evanschitzky 2003: Ahlert, Dieter; Blaich, Günter; Evanschitzky, Heiner: Systematisierung von Dienstleistungsnetzwerken. In: In: Bruhns, Manfred; Stauss, Bernd (Hrsg.): Dienstleistungsnetzwerke, Jahrbuch Dienstleistungsmanagement 2003, Gabler.

Baumöl/Meschke 2009: Baumöl, Ulrike; Meschke, Martina: Das Management von Datenqualität. In: Controlling & Management 2009 Vol. 53, Nr 1, S. 62-65

Baumöl 2006: Baumöl, Ulrike: Methodenkonstruktion für das Business/IT Alignment. In: Wirtschaftsinformatik 2006 Vol 48, Nr 5, S 314-322

Bartlog/Boy 2010: Bartlog, Heiko; Boy, Jochen: Collaborative Project Management (CPM). In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Barton/Bach 2010: Barton, Thomas; Bach, Harriet: Modellierung eines Anwendungssystems zur Behälterlokalisierung und Behälterreservierung auf Basis des Architekturstils REST. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Baskerville/Cavallari/Hjort-Madsen/Pries-Heje/Sorrentino/Virili 2005: Baskerville, Richard; Cavallari, Marco; Hjort-Madsen, Kristian; Pries-Heje, Jan; Sorrentino, Maddalena; Virili, Francesco: Extensible Architectures: The Strategic Value of Service-Oriented Architecture in Banking. In Proceedings of the Thirteenth European Conference on Information Systems (Bartmann D, Rajola F, Kallinikos J, Avison D, Winter R, Ein-Dor P, Becker J, Bodendorf F, Weinhardt C eds.), 761-772, Regensburg, Germany.

Bass/Klein/Moreno 2001: Bass, Len; Klein, Mark; Moreno, Gabriel: Applicability of General Scenarios to the Architecture Tradeoff Analysis MethodSM. In: TECHNICAL REPORT CMU/SEI-2001-TR-014 ESC-TR-2001-014

Becker/Buxmann/Widjaja 2009: Becker, Alexander; Buxmann, Peter; Widjaja, Thomas: Value Potential and Challenges of Service-Oriented Architectures – a User and Vendor Perspective. In 17th European Conference on Information Sys-

tems (Newell S, Whitley EA, Pouloudi N, Wareham J, Mathiassen L eds.), 2085-2096, Verona, Italy.

Becker/Janiesch/Pöppelbuß 2008: Becker, Jörg; Janiesch, Christian; Pöppelbuß, Jens: Konfiguration kollaborativer Informationsmodelle. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Beverungen/Kaiser/Knackstedt/Krings/Stein 2008: Beverungen, Daniel; Kaiser, Uwe; Knackstedt, Ralf; Krings, Robin; Stein, Armin: Konfigurative Prozessmodellierung der hybriden Leistungserstellung in Unternehmensnetzwerken des Maschinen- und Anlagenbaus. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Bianco/Kotermanski/Merson 2007: Bianco, Phil; Kotermanski, Rick; Merson, Paulo: Evaluating a Service-Oriented Architecture. In TECHNICAL REPORT CMU/SEI-2007-TR-015 ESC-TR-2007-015 Software Architecture Technology Initiative

Blau/Schnizler 2008: Blau, Benjamin; Schnizler, Björn: Description Languages and Market Mechanisms for Trading Grid Services. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Böhringer/Gluchowski 2010: Böhringer, Martin; Gluchowski, Peter: Ubiquitous Microblogging als dezentrales Entwurfparadigma für leichtgewichtige Informationssysteme. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Braun/Winter 2005: Braun, Christian; Winter, Robert: Classification of Outsourcing Phenomena in Financial Services. In Proceedings of the Thirteenth European Conference on Information Systems (Bartmann D, Rajola F, Kallinikos J, Avison D, Winter R, Ein-Dor P, Becker J, Bodendorf F, Weinhardt C eds.), 349-360, Regensburg, Germany.

Clark/Tennenhouse 1990: Clark, David D.; Tennenhouse, David L.: Architectural considerations for a new generation of protocols. In: Proceedings of the ACM symposium on Communications architectures & protocols 1990

DeLone/McLean 2003: DeLone, William H.; McLean, Ephraim R.: The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. In Journal of Management Information Systems / Spring 2003, Vol. 19, No. 4, pp. 9-30.

Dömer 1998: Dömer, Fabian: Migration von Informationssystemen - Erfolgsfaktoren für das Management. Deutscher Universitäts-Verlag GmbH, Wiesbaden.

Dreiling 2010: Dreiling, Alexander: ERP-Einführung: Wirkung von kritischen Erfolgsfaktoren der Projektphase auf den Projekterfolg. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Fielding 2000: Fielding, Roy Thomas: Architectural Styles and the Design of Network-based Software Architectures. Dissertation 2000 University of California, Irvine, Dissertation Committee Professor Richard N. Taylor, Professor Mark S. Ackerman, Professor David S. Rosenblum

Frick/Schubert 2010: Frick, Norbert; Schubert, Petra: Flexibilität in ERP-Standardsoftware. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Hofmann/Ley/Dörner 2008: Hofmann, Markus; Ley, Benedikt; Dörner, Christian: Endbenutzergerechte Anpassung von serviceorientierten Softwaresystemen. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Karaenke/Bieser/Schuele/Kirn 2008: Karaenke, Paul; Bieser, Thomas; Schuele, Michael; Kirn, Stefan :Towards a Market-Centric OGSA-Compliant Architecture Model. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Klose/Knackstedt/Beverungen 2007: Klose, Karsten; Knackstedt, Ralf; Beverungen, Daniel: Identification of Services - A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning. In Proceedings of the Fifteenth European Conference on Information Systems (Österle H, Schelp J, Winter R eds.), 1802-1814, University of St. Gallen, St. Gallen.

Knöfel/Barth 2010: Knöfel, Markus; Barth Thomas: Kriterien für den Einsatz Service-orientierter Architekturen in der Unternehmens-IT. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Laures 2006: Laures, Guido: Flexibilitätsanalyse service-orientierter Architekturen zur Realisierung von Geschäftsprozessen. In EMISA 2006 Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen, Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung) P-95, 163-177 (2006).

Legner/Heutschi 2007: Legner, Christine ; Heutschi, Roger: SOA Adoption in Practice - Findings from Early SOA Implementations. In: Oesterle, Hubert (Hrsg.) ; Schelp, Joachim (Hrsg.) ; Winter, Robert (Hrsg.), 2007. - European Conference on Information Systems (ECIS 2007). - St. Gallen, S. 1643-1654. - Volltext unter <http://www.alexandria.unisg.ch/Publikationen/35214> (Stand: 2010-07-04)

Lönnngren/Kolbe/Rosenkranz 2008: Lönnngren, Hans-Martin; Kolbe, Harald; Rosenkranz, Christoph: Erfolgsfaktoren für hybride Wertschöpfungsnetze – Eine Fallstudienanalyse. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Mangan/Kelly 2003: Mangan, Anita; Kelly, Seamas: IS and the integrated network organisation: a cautionary tale from the financial services sector. In Proceedings of the Eleventh European Conference on Information Systems (Ciborra CU, Mercurio R, de Marco M, Martinez M, Carignani A eds.), 1220-1235, Naples, Italy.

Minor/Schmalen/Bergmann 2008: Minor, Mirjam; Schmalen, Daniel; Bergmann, Ralph: XML-based Representation of Agile Workflows. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Mueller/Viering/Ahlemann/Riempp 2007: Mueller, Benjamin; Viering, Goetz; Ahlemann, Frederik; Riempp, Gerold: Towards Understanding the Sources of the Economic Potential of Service-Oriented Architecture: Findings from the Automotive and Banking Industry. In Proceedings of the Fifteenth European Conference on Information Systems (Österle H, Schelp J, Winter R eds.), 1608-1619, University of St. Gallen, St. Gallen.

Pütz/Wagner/Ferstl/Sinz 2010: Pütz, Corinna; Wagner, Daniel; Ferstl, Otto K.; Sinz, Elmar J. : Konzeption eines generischen Geschäftsprozessmodells für Medizinische Versorgungszentren. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Scholz/Breitner/Blaurock 2008: Scholz, S.; Breitner, MH.; Blaurock, M.: A Sustainable Business Model Approach for Grid Computing – And a Life Sciences Example. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Schönherr 2004: Schönherr, Marten: Enterprise Architecture Frameworks. In: Aier, S., Schönherr, M. (Hrsg.): Enterprise ApplicationIntegration – Serviceorientierung und nachhaltige Architekturen, Gito, Berlin, S. 75–122

Schömburg/Breitner 2010: Schömburg, Harald; Breitner, Michael H. : Elektronische Rechnungen zur Optimierung der Financial Supply Chain: Status Quo, empirische Ergebnisse und Akzeptanzprobleme. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Schroth 2008: Schroth, Christoph: „Richness“ und „Reach“ organisationsübergreifender, Informations-intensiver Dienste. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Schubert 2008: Schubert, Petra: Business Collaboration: Erfahrungen aus der Unternehmenspraxis. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Schubert/Williams 2010: Schubert, Petra; Williams, Susan P.: Erwartungen und Nutzen beim Einsatz von Business Software Empirische Ergebnisse einer Analyse von Fallstudien. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Schwinn/Winter 2007: Schwinn, Alexander; Winter, Ropert: Success Factors and Performance Indicators for Enterprise Application Integration. In: Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE, USA August 11th-14th 2005

Somers/Nelson 2004: Somers, Toni M.; Nelson, Klara G.: A taxonomy of players and activities across the ERP project life cycle. In: Information & Management 41 (2004) S. 257-278

Stauss/Bruhns 2003: Stauss, Bernd; Bruhns, Manfred: Dienstleistungsnetzwerk – Eine Einführung in den Sammelband. In: Bruhns, Manfred; Stauss, Bernd (Hrsg.): Dienstleistungsnetzwerke, Jahrbuch Dienstleistungsmanagement 2003, Gabler.

Steinhüser/Räth 2010: Steinhüser, Melanie; Räth, Philip: Erfolgsdimensionen von Social Software: Eine fallstudienbasierte Untersuchung. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Sunyaev/Schweiger/Leimeister/Krcmar 2008: Sunyaev, Ali; Schweiger, Andreas; Leimeister, Jan Marco; Krcmar, Helmut: Software-Agenten zur Integration von Informationssystemen im Gesundheitswesen. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Walter/Werth 2008: Walter, Philipp; Werth, Dirk: Eine Peer-to-Peer Infrastruktur zur Konstruktion kollaborativer Geschäftsprozesse. In Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings

Winter 2003: Winter, Robert: An architecture model for supporting application integration decisions. In Proceedings of the Eleventh European Conference on Information Systems (Ciborra CU, Mercurio R, de Marco M, Martinez M, Carignani A eds.), 2188-2200, Naples, Italy.

Zagel/Löffler 2010: Zagel, Christian; Löffler, Carolin: Technisches Konzept zur Service Externalisierung virtueller Prototypen in der Modebranche. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner, Arne Frerichs (Hrsg.): Multikonferenz Wirtschaftsinformatik 2010

Eidesstattliche Erklärung & Einverständniserklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Bachelorarbeit selbständig und ohne Inanspruchnahme fremder Hilfe angefertigt habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Ich erkläre mich damit einverstanden, dass die Arbeit mit Hilfe eines Plagiaterkennungsdienstes auf enthaltene Plagiate überprüft wird.

Ort, Datum

Unterschrift