

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals
in den Bereichen Softwareentwicklung,
Internettechnologie und IT-Management aktuell
und kompetent relevantes Fachwissen über
Technologien und Produkte zur Entwicklung
und Anwendung moderner Informationstechnologien.

Dieter Masak

Moderne Enterprise Architekturen

Mit 217 Abbildungen

 Springer

Dieter Masak
plenum Systems GmbH
Hagenauer Str. 53
65203 Wiesbaden
dieter.masak@plenum.de

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

ISSN 1439-5428
ISBN 3-540-22946-9 Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Verlag und Autor können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Satz und Herstellung: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142/YL - 5 4 3 2 1 0

Danksagung

... meiner Familie, für ihre Unterstützung und Verständnis ...
... meinen Kollegen, für ihre Denkanstöße ...
... Martin Spenger, für die Freiräume ...

Dr. Dieter Masak

Prolog

Ein Architekt wird der sein, . . . , der gelernt hat, mittels eines bestimmten und bewundernswerten Planes und Weges sowohl in Gedanken und Gefühl zu bestimmen, als auch in der Tat auszuführen, was unter der Bewegung von Lasten und der Vereinigung und Zusammenfügung von Körpern den hervorragendsten menschlichen Bedürfnissen am ehesten entspricht und dessen Erwerbung und Kenntnis unter allen wertvollen und besten Sachen nötig ist. Derart wird also ein Architekt sein.

Leon Battista Alberti
1404–1472

Inhaltsverzeichnis

1	Einleitung	1
2	Umfeld	5
2.1	Geschichte	5
2.2	Warum Enterprise Architektur?	9
2.3	Treibende Kräfte	10
2.4	Lebensdauer von Applikationen	11
2.5	Outsourcing	13
2.6	Flexibilität und Skalierbarkeit	14
2.7	Offenheit und Interoperabilität	15
2.8	Modularität	15
2.9	Wartbarkeit und Zukunftssicherheit	16
2.10	Wiederverwendbarkeit	16
2.11	Teile der Enterprise Architektur	17
3	Framework	19
3.1	Frameworks, Standards und Techniken	19
3.2	Einbettung der Enterprise Architektur	20
3.3	Gesamtframework	21
3.4	Architektur-governance-Framework	22
3.5	Enterprise Architektur-Framework	23
3.6	Entwicklung der Enterprise Architektur	24
3.6.1	Prinzipien	24
3.6.2	Vision	25
3.6.3	Geschäftsprozess-Architektur	25
3.6.4	Systemarchitektur	26
3.6.5	Technologie-Architektur	27
3.6.6	Lösungen	29
3.6.7	Implementierungsgovernance	29
3.6.8	Migration	30
3.6.9	Architektur-Changemanagement	30

3.7	Architektur und Lösungen	32
3.8	Nutzung	34
3.9	Zachman-Framework	34
3.10	C ⁴ ISR	36
4	Topologien	39
4.1	Einleitung	39
4.2	Entwicklung der Verteilung	40
4.2.1	Mainframe-Ära	40
4.2.2	Client-Server-Ära	43
4.2.3	Service-Architektur-Ära	44
4.3	Systemarchitektur	45
4.3.1	Grafische Elemente	45
4.3.2	Applikationen	45
4.3.3	Verbindungen	46
4.3.4	Blackbox-Graphen	48
4.3.5	Kardinalitäten	49
4.4	Einfache Topologien	49
4.4.1	Einfache Anwendung	50
4.4.2	Datawarehouse	51
4.4.3	Einfaches Webbeispiel	52
4.4.4	Hub and Spoke	52
4.4.5	Portale	52
4.4.6	Edge-Service-Topologie	53
4.4.7	Enterprise Application Integration	55
4.4.8	Application-Server	55
4.4.9	Extended Enterprise	56
4.4.10	Virtuelles Unternehmen	56
5	Metriken	59
5.1	Einleitung	59
5.2	Graphen	63
5.2.1	Senke	64
5.2.2	Quelle	64
5.2.3	Gewicht	64
5.3	Arbeitslast	66
5.4	Zeiten	67
5.5	Latenz	69
5.6	Komplexität	69
5.6.1	Average Node Degree	69
5.6.2	McCabe-Metrik	70
5.6.3	Card-Metrik	71
5.7	Small Worlds	72
5.8	Entropie	74
5.9	Temperatur	78

5.10	Volatilität	79
5.11	Skalierbarkeit	81
6	Evolution und Revolution	85
6.1	Einleitung	85
6.2	Alterungsprozess	86
6.3	Komplexität	88
6.4	Lehmans Gesetz	89
6.5	Volatilität	90
6.6	Entropie	92
6.7	Kosten	94
7	Systemarchitektur	97
7.1	Einleitung	97
7.2	Netzwerk	98
7.3	Speicherarchitektur	100
8	Datenarchitektur	103
8.1	Einleitung	103
8.2	Datenquellen	104
8.3	Entwicklung	105
8.4	Metadaten	107
9	Mikroarchitekturen	111
9.1	Einleitung	111
9.2	Monolithische Systeme	111
9.3	Schichtenarchitektur	116
9.3.1	Client-Server	116
9.3.2	n-Tier	120
9.4	Komponenten	123
9.5	Service Oriented Architecture	125
9.5.1	Einleitung	125
9.5.2	Metriken	127
9.5.3	Semantik	132
9.6	Webservices	132
9.6.1	Einleitung	132
9.6.2	Modell	133
9.6.3	Services	133
9.6.4	SOAP	139
9.6.5	UDDI	141
9.6.6	Taxonomie	142
9.6.7	WSDL	143
9.7	CORBA	145
9.7.1	Einleitung	145
9.7.2	OMA	146

9.7.3	Basisstruktur	150
9.7.4	Interface Description Language	154
9.7.5	DSI	155
9.7.6	Interface Repository	156
9.7.7	GIOP	157
9.7.8	Metrik	157
9.7.9	CORBA vs. SOAP	160
9.8	Application-Server	160
9.8.1	Einleitung	160
9.8.2	Topologie	162
9.9	Messaging	164
9.9.1	Integration	164
9.9.2	JMS	166
9.10	Enterprise Application Integration	168
9.11	Portale	176
10	Produktplattformen	183
10.1	Einleitung	183
10.2	Java 2 Enterprise Edition	183
10.2.1	Geschichte	183
10.2.2	Überblick	185
10.2.3	Container	187
10.2.4	Servlets	189
10.2.5	JSP	191
10.2.6	Enterprise Java Beans	193
10.2.7	Session Beans	195
10.2.8	Entity Beans	196
10.2.9	Message Beans	199
10.2.10	JMX	200
10.2.11	Java Connector Architecture	202
10.2.12	JBoss	203
10.2.13	WebSphere	205
10.3	.NET	208
11	Migration und Integration	215
11.1	Einleitung	215
11.2	Legacy-Migration	216
11.2.1	Applikationen	218
11.2.2	Fusionen	219
11.3	Rekonstruktion	220
11.4	Webservices und SOA	221
11.4.1	Zufällige Integration	221
11.4.2	Webservice-Wrapper	222
11.4.3	Managed Webservices	224
11.4.4	Paradigmaverschiebung	225

11.5	Enterprise Application Integration	226
11.5.1	Legacy-Applikationen	228
11.5.2	Application-Server	229
11.5.3	Real Time Enterprise	230
11.6	Legacy-Migration Roadmap	230
11.6.1	User-Interface-Integration Roadmap	231
11.6.2	Message Driven Integration Roadmap	234
12	Entwicklungsprozess	235
12.1	Komplexe Systeme	235
12.2	Rational Unified Process	239
12.3	Enterprise Unified Process	243
13	Model Driven Architecture	249
13.1	Einleitung	249
13.2	Pattern	250
13.3	Struktur	251
13.4	Common Services	255
13.5	Dynamik	256
13.6	Meta Object Facility	256
13.7	XML Metadata Interchange	257
13.8	MDA-Implementierung	259
13.9	Perspektiven	259
14	Agility	261
14.1	Agiles Manifest	261
14.2	Agile Modellierung	268
14.2.1	eXtreme Programming	268
14.2.2	Dynamic System Development Method	270
14.2.3	Adaptive Software Development	270
14.2.4	Scrum	271
14.2.5	Feature-Driven Development	272
14.2.6	Agile Modelle und Dokumente	273
14.3	Agile Architektur	274
14.4	AMDD	279
15	Patterns und Antipatterns	283
15.1	Einleitung	283
15.2	Allgemeine Patterns	284
15.2.1	Data Warehouse	284
15.2.2	Integration Broker	285
15.2.3	Channel-Architektur	286
15.2.4	Softwaredarwinismus	286
15.2.5	Kleine Oberfläche	287
15.2.6	Service Layer	288

15.2.7	Gateway	289
15.2.8	Deprecation	289
15.2.9	Teile und modernisiere	290
15.3	Mikroarchitekturpatterns	290
15.3.1	Filter	290
15.3.2	Schichten	291
15.3.3	Repository	292
15.3.4	Distributed Object	294
15.3.5	Microkernel	294
15.3.6	Broker	295
15.4	Antipatterns	296
15.4.1	Über den Wolken	296
15.4.2	Detailitis	297
15.4.3	Elfenbeinturm	297
15.4.4	Perfektionitis	297
15.4.5	Modernitis	298
15.4.6	Spielzeugladen	298
15.4.7	Vergoldung	298
15.4.8	Conway's Law	299
15.4.9	Servicitis	299
15.4.10	Silver-Bullet	299
15.4.11	Schichtenfieber	300
16	Epilog	301
	Glossar	303
	Literaturverzeichnis	319
	Index	327

Einleitung

*Nicht weil etwas schwer ist
wagen wir es nicht,
sondern weil wir es nicht wagen
ist es schwer . . .*

Seneca

Was ist eine Enterprise Architektur? Eine Enterprise Architektur ist eine Art Plan oder Planung der Informationssysteme eines Unternehmens. Stellen wir uns eine Stadt ohne eine Stadtplanung vor. Was entsteht ist ein mehr oder minder gewachsenes System mit der Eigenschaft, dass die Infrastrukturteile verschiedener Stadtteile nicht mehr zusammenpassen. Es gibt mehrere parallele Kanalisationen, mehrere Stromnetze, Straßen mit unterschiedlicher Breite und Belag usw. Eine solche Stadt ist zwar lebensfähig, aber ihre Infrastruktur reicht nicht aus, um rapides Bevölkerungs- oder Geschäftswachstum zu verkraften. Ähnliches entsteht bei dem Wachstum von Unternehmen mit deren Informationssystemen. Die Enterprise Architektur ist die Stadtplanung der Informationssysteme.

Das große Interesse an Enterprise Architekturen ist darauf zurückzuführen, dass die gesamte IT-Branche an einem kritischen Punkt angelangt ist. Das steile Wachstum der Informationstechnologie in den letzten vierzig Jahren war auf die schnell ansteigende Automatisierung zurückzuführen. Im Rahmen dieses Prozesses wurden Abläufe, die bisher durch Menschen durchgeführt wurden, einfach in analoge Computersysteme umgesetzt und nun von diesen durchgeführt. Hierbei stand jedoch der Ablauf, der bisher in Unternehmen stattfand, Pate für die Entwicklung von Software, angefangen von so genannten Sachbearbeitern bis hin zu den heutigen Use Cases. In allen Fällen ist das menschliche Handeln, was in der Regel aus der Unternehmens- oder Branchenhistorie entstammt, Leitlinie für die Software. Durch dieses Vorgehen lassen sich echte Quantensprünge in der Entwicklung als auch latente Bedürfnisse nur sehr schwer abdecken. Eine solche zukünftige Entwicklung ist erst möglich, wenn die vollständige Innovationskraft unabhängig von menschlichem Tun eingesetzt wird.

Heutige große Softwaresysteme in Unternehmen bestehen aus einer fast unüberschaubaren Vielzahl von verschiedenen Applikationen, welche alle der Automatisierung dienen, jedoch stets eine partikuläre Sicht haben. Diese Tat-

sache erklärt auch, warum die Kosten für den IT-Betrieb als auch für Softwareentwicklungen immer drastischer ansteigen. Diese zunehmende Komplexität steigert die Kosten und lähmt das Handeln. In den letzten Jahren wurde immer deutlicher, dass die Schaffung und Verwaltung von Enterprise Architekturen ein möglicher Weg zur Lösung dieser Probleme sein kann. Die durch die Enterprise Architektur erzwungene und gelebte Standardisierung führt sowohl zu einer Senkung der Kosten im Betrieb als auch in der Entwicklung von Software.

Die Parallele hierzu ist das so genannte lean manufacturing in der produzierenden Industrie, speziell im Automobilsektor. Hier zeigte sich in den Siebzigerjahren des letzten Jahrhunderts, dass die Autos immer komplexer und teurer wurden, da die Hersteller fast alle Teile selbst produzierten und versuchten, jedes noch so winzige Teil auch zu kontrollieren. Relativ schnell konnte hier, wie auch in der Flugzeugindustrie, gezeigt werden, dass die Nutzung von vorgefertigten und zugelieferten Komponenten zu einem Königsweg in der Produktion wurde. Ein Fahrzeug wird durch diese großen Komponenten, beispielsweise die komplette Armaturenelektronik oder das Getriebe, viel effektiver gebaut. Das Geheimnis dahinter liegt in der Standardisierung der Schnittstellen zwischen den einzelnen Komponenten und der daraus implizit folgenden Austauschbarkeit der Komponenten.

Der gleiche Weg steht der IT-Branche noch bevor. Im Vergleich zur Automobilbranche oder der Flugzeugindustrie befindet sie sich praktisch noch in den Kinderschuhen, trotzdem wird sie sich rasant weiterentwickeln. Um die Einleitungsparallele der Abbildung des menschlichen Vorgehens an dieser Stelle weiter zu tragen: Wir befinden uns, relativ zur Entwicklung des Autos, innerhalb der IT-Branche in der Zeit, wo Autos noch aussahen wie Kutschen! Trotz alledem, Softwaresysteme sind das Komplexeste, was je von Menschen geschaffen wurde, obwohl die Einfachheit ihrer Nutzung uns anderes suggeriert. Aus dem Blickwinkel der unterliegenden Komplexität betrachtet sind große Softwaresysteme erstaunlich stabil, obwohl ein einzelner Anwender im Tagesgeschäft da oft anderer Ansicht ist.

Die Tatsache, dass unsere heutige IT-Technologie sich rasch ändert, impliziert im Umkehrschluss, dass eine Enterprise Architektur in der Lage sein muss, diese Veränderung nicht nur zu überstehen, sondern sie auch aktiv zu beeinflussen. Neben reiner technischer Veränderungen muss sich eine Enterprise Architektur auch auf ein großes Maß an Veränderungen im Bereich der Geschäftswelt einstellen. Neben Änderungen einzelner Geschäftsprozesse stehen in der jetzigen Zeit sowohl durch Fusionen und Übernahmen als auch, in Ausnahmefällen, durch explosives Wachstum ein großes Maß an Flexibilität ins Haus. Ein solcher Schritt versucht aber gleichzeitig das bestehende Investment, welches in Applikationen getätigt wurde, zu erhalten. Die rasante Geschwindigkeit der Änderungen innerhalb der IT-Technologie, zusammen mit der Tatsache, dass Informationsverarbeitung ein immer wichtigerer Bestandteil jedes Unternehmens wird, zeigt die Dringlichkeit auf, sich dem Thema Enterprise Architektur zu stellen.

Vermutlich wird die Abhängigkeit der Unternehmen von ihrer IT-Landschaft in den nächsten Jahren noch stärker zunehmen, da hier die einfachsten Effizienzgewinne für Hochlohnländer zu erreichen sind. Trotz dieser Wichtigkeit der IT-Landschaft, oder gerade weil sie so fundamental ist, herrscht ein hoher Grad an Unzufriedenheit in den Vorständen an der Leistungsfähigkeit der eigenen IT-Abteilungen. Die meisten Vorstände glauben nicht, dass ihre IT-Abteilungen optimal arbeiten. Die Tatsache, dass weniger als 25% aller Softwareprojekte ihr Ziel auch nur ansatzweise erreichen, trägt ein Übriges zu diesem Negativbild bei.

In Bezug auf die Enterprise Architektur trägt die Analogie zur Flugzeugindustrie noch weiter: Genau wie bei einem Flugzeug bleiben heute große IT-Infrastruktursysteme strukturell intakt, wenn Teile ausgetauscht werden, in den meisten Fällen wird sogar der Betrieb vollständig aufrechterhalten.

Die Architekten der Enterprise Architekturen sind stets damit beschäftigt, welche Auswirkungen ihre Tätigkeit für große Systeme, d.h. auch jenseits des eigenen Unternehmens, für die Gegenwart und vor allen Dingen für die Zukunft hat. Sie müssen jenseits der einzelnen Applikation schauen und sich bei jeder neuen Anwendung die Frage stellen:

„Welchen Einfluss oder Auswirkung hat diese Anwendung auf die Enterprise Architektur?“

Im Rahmen der Enterprise Architektur muss der Blick über den aktuellen Handlungsbedarf hinausgehen, da hier die Eckpfeiler für die Zukunft formuliert werden.

Es ist wichtig darauf hinzuweisen, dass Enterprise Architekturen keine Abfolge von statischen Momentaufnahmen eines komplexen Systems darstellen, sondern permanent überarbeitet und redefiniert werden müssen, genauso wie die Geschäftsentwicklung am Markt es diktiert.

Umfeld

*Chief architect and plotter of these woes
The villain is alive in Titus house. . .*

Marcus Andronicus,
William Shakespeare

2.1 Geschichte

Ein Enterprise, wörtlich übersetzt eine Unternehmung, ist im ursprünglichen Sinne eine Aktivität, welche eine wohl definierte Zielsetzung beinhaltet. Heute verstehen wir darunter ein Unternehmen oder eine Menge von Organisationen, welche eine gemeinsame Zielsetzung haben oder ein gemeinsames Ergebnis produzieren. Ein Enterprise in diesem Sinne kann alles von einem großen Konzern bis hin zu einer staatlichen Institution oder einem Ministerium sein. Große Konzerne und staatliche Stellen bestehen oft aus mehreren Enterprises. Folglich besitzen diese auch mehrere Enterprise Architekturen. Ein Enterprise, in dem hier verwendeten Sinn, kann aber auch ein so genanntes extended Enterprise sein, dieses beinhaltet dann, neben dem eigentlichen Kernunternehmen, auch alle Partner, Lieferanten und Kunden des eigentlichen Unternehmens.

Neben den extended Enterprises spielen in der derzeitigen Diskussion die virtuellen Unternehmen eine immer stärkere Rolle. Ein virtuelles Unternehmen ist eine Organisationsform, welche unabhängige Partner vereint, um einen einmaligen Auftrag zu erfüllen, und danach wieder aufgelöst wird, beispielsweise die Arbeitsgemeinschaften beim Bau von Autobahnen. Ein virtuelles Unternehmen wird für jede Teilaufgabe den bestmöglichen Anbieter einsetzen, dessen Kernkompetenz identisch mit der Aufgabe sein sollte. Die sich daraus ableitenden Problemstellungen tauchen immer wieder auf, dazu zählen effiziente und durchaus verletzbare Verbindungen zwischen den einzelnen Unternehmen. Diese Verbindungen müssen in Bezug auf Flexibilität und Versatilität ein ungleich höheres Maß an Anforderungen erfüllen, als das sonst, im Rahmen eines normalen Unternehmens, notwendig ist. Als Organisationsform zerfallen diese virtuellen Unternehmen in first- und second-level Organisationen. Zu den typischen first-level virtuellen Unternehmen zählen Projekte innerhalb eines Konzerns, s. Abb. 2.1, während eine echte virtuelle Organisation zum second-level gehört. Im Gegensatz zu den extended Enterprises ist

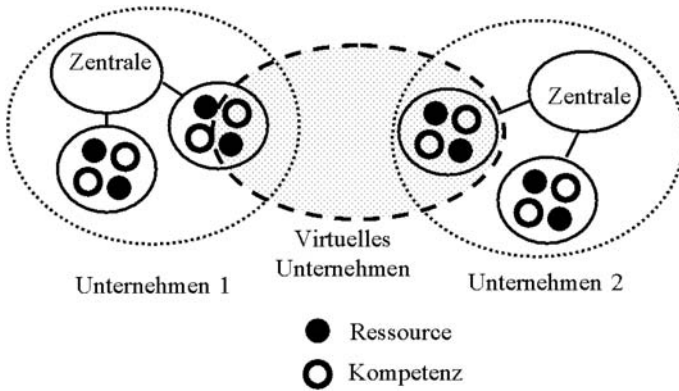


Abb. 2.1. Das virtuelle Unternehmen

die zeitliche Begrenzung der virtuellen das Schlüsselement. Der Grund, hinter dem Bestreben virtuelle Unternehmen zu bilden, liegt in den entstehenden Kosten. Folgt man der Transaktionstheorie, so lassen sich die Kosten für die verschiedenen Unternehmensformen recht leicht visualisieren, s. Abb. 2.2. Dabei wird klar, dass erst bei sehr speziellen Vorgängen die Transaktionskosten innerhalb eines Unternehmens denen des freien Marktes überlegen sind. Solche Vorgänge müssen so speziell sein, dass sie keiner Skalenökonomie unterliegen, dann kann auch ein hierarchisches Unternehmen niedrige Transaktionskosten erzielen.

Der Unterschied zwischen einem extended Enterprise und einem virtuellen Unternehmen lässt sich an zwei Stereotypen beider Ausrichtungen aufzeigen:

- Dell. Dieses amerikanische Unternehmen produziert nichts selbst. Alle Produkte sowie der Zusammenbau innerhalb der Produktion werden durch Dritte gefertigt. Dell ist jedoch immer bestrebt, eine stabile Partnerschaft mit seinen Lieferanten einzugehen und sichert diese auch vertraglich ab.
- Open-Source-Projekte. Die Open-Source-Projekte sind das Paradebeispiel für virtuelle Organisationen. Es handelt sich zwar um non-profit Organisationen, trotzdem stellen sie im Rahmen der obigen Definition Enterprises dar. Hier arbeiten projektbezogen eine Reihe von Individuen auf zeitlich befristeter Basis zwecks eines gemeinsamen Zieles zusammen. Das Betriebssystem Linux ist durch ein solches virtuelles Unternehmen entstanden.

Obwohl die Diskussion bezüglich virtuellen Unternehmen und extended Enterprises in der Fachpresse intensiv geführt wird, stehen die meisten Unternehmen vor der Herausforderung, zunächst einmal ein Real TimeEnterprise, RTE, aufzubauen. Die mehr oder minder offizielle Definition eines Real TimeEnterprise stammt von der Gartner Group:

The RTE is fundamentally about squeezing lag time or „information float“ out of core operational and managerial processes. Business has

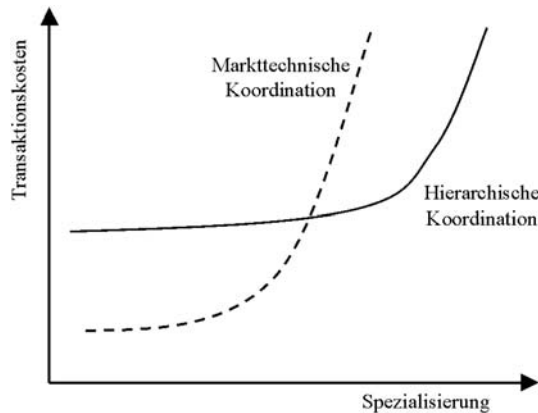


Abb. 2.2. Transaktionskosten als Funktion der Spezialisierung

long understood that time is money and speed is a competitive advantage. However, „time“ is taking on new meaning as access to markets and talent pools becomes global, and as innovation cycle times and product life cycles shrink. The ability to remove latency or lag times from key business processes, to extend processes globally to „follow the sun“, and to access real-time financial, managerial and operating data will increasingly separate industry leaders from also-rans. In time, RTE capability will be a requirement for staying in business.

Danach ist ein Real TimeEnterprise ein Unternehmen, was alle geschäftsprozessrelevanten Informationen in „Echtzeit“ zur Verfügung stellen kann. Hier zwei Beispiele von aktuellen Real Time Enterprises:

- Wal-Mart. Obwohl noch nicht ganz in Echtzeit, ist Wal-Mart das bekannteste Handelsunternehmen, welches IT als strategisches Werkzeug einsetzt und sehr zeitnah Informationen über Produkte, Filialen und Kassen verarbeiten und analysieren kann.
- H&M. Hennes und Mauritz sind ein integrierter Textilanbieter. H&M kann Textilien in sehr viel kürzeren Zyklen auf den Markt werfen als ihre Konkurrenz. Hier ist das Real TimeEnterprise im Produktionssektor bzw. in der Qualität der Supply-Chain vorhanden.

Leider wird die heutige Diskussion um das Real TimeEnterprise sehr auf technischer Ebene geführt, dies ist jedoch der falsche Ansatz, denn eine entsprechende Strategie kann nur aus dem Geschäftsprozess und dem Kundenverhalten, beispielsweise Wünschen, abgeleitet werden. Denn eine Betrachtung aus rein technologischer Sicht führt in den seltensten Fällen zu einem echten Mehrwert. Im Rahmen von technikgetriebenen Strategien werden sehr oft Prozesse beschleunigt, die keine oder nur eine sehr geringe Kunden- oder Geschäftsprozessrelevanz besitzen.

Die hinter allen diesen Ansätzen stehende Problematik der Unternehmen liegt in einigen wenigen, dafür aber um so schmerzhafteren Punkten begründet.

- **Kundenkontakt.** Die traditionelle funktionsorientierte Organisation verhindert de facto eine ganzheitliche Kundenbetrachtung. Produkt- und funktionsorientierte Unternehmen bilden für den Kunden ein oft verwirrendes Schema mit unterschiedlichen Aussagen aus demselben Unternehmen.
- **Lieferantenkontakt.** Hier herrscht in der Industrie noch immer der Bullwhip-Effekt vor, der auf die nicht vorhandene Koppelung in Echtzeit zurückzuführen ist. Der Bullwhip-Effekt beruht darauf, dass bei erhöhtem Bestellaufkommen der Lieferant nicht schnell genug reagiert, und wenn er reagiert, die Nachfrage schon wieder gesunken ist. Traditionell wird dies durch den Aufbau von Lagerbeständen abgemildert. Allerdings sind die Lagerbestände in der Regel sehr kapitalintensiv, was eine große Ressourcenverschwendung darstellt.

Im Gegensatz zum privaten Sektor waren bei den öffentlichen Händen Fragen nach Investitionen in Informationstechnologien in der Vergangenheit selten ein Thema. Traditionell wurde Informationstechnologie für Ämter nie als eine zentrale Investition gesehen. Erst der Kostendruck bzw. der Legitimationsdruck für Kosten hat die Fragen nach Enterprise Architekturen und Return of Investment gestellt. Das zweite große Problem der öffentlichen Hände ist, dass sie, obwohl sie die gleichen Kunden, die Bürger, haben, völlig separat voneinander agieren. Diese Separation schlägt sich direkt in der heute existierenden Softwarelandschaft nieder. Mit der Einführung einer behördenweiten Enterprise Architektur könnte genau wie in der Industrie dieses Problem gelöst werden.

Bis in die Neunzigerjahre hinein war es für ein Unternehmen durchaus legitim, mit einer großen Zahl von fragmentierten monolithischen Systemen zu leben, da Informationssysteme meistens eine interne Aufgabe wahrnahmen, die auch nur eigenen Mitarbeitern zugänglich wurde. Diese Mitarbeiter waren hoch spezialisiert und führten, bzw. führen auch heute noch, Detailaufgaben aus. Daher konnte man sie sowohl mit einem kryptischen Benutzerinterface als auch einer sehr eingeschränkten und zum Teil widersprüchlichen Funktionalität und dem entsprechenden Datenhaushalt zufrieden stellen. Das Aufkommen des Internets öffnete diese Welt an Dritte, Kunden wie auch andere Geschäftspartner. Die Folge hiervon war, dass Brüche immer offensichtlicher wurden und nicht mehr akzeptiert werden konnten.

Die zunehmende Spezialisierung der Unternehmen durch den immer höher werdenden Kostendruck führt zu einer stärkeren Fragmentierung der Geschäftsprozesse. Heutige Allgemeinbanken zum Beispiel prozessieren kaum noch ihren eigenen Wertpapierhandel, sondern überlassen das Processing meistens spezialisierten Transaktionsbanken. Ein weiteres Beispiel für ein stark ausgedehntes extended Enterprise sind die Automobilproduzenten mit allen ihren komplexen Zuliefererstrukturen. Bezeichnenderweise waren die Au-

tomobilproduzenten die ersten Unternehmen in Deutschland, welche ein standardisiertes Austauschprotokoll für Bestell- und Fakturierdaten, EDIFACT, bewusst forcierten und der Idee des extended Enterprises und damit einer Enterprise Architektur sehr nahe kamen.

Die Architektur ist die abstrakte Struktur einer Aktivität. Unter Enterprise Architektur im engeren Sinne wird die IT-Struktur eines Unternehmens bezeichnet. Die offizielle Definition des Begriffs Architektur nach dem ANSI/IEEE-Standard 1471-2000 im Umfeld der Informationstechnologie lautet:

An architecture is the fundamental organisation of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

Die hier verwendete Definition ist allerdings etwas enger:

Eine Architektur ist eine formale Beschreibung eines Systems, ein detaillierter Plan des Systems und seiner Komponenten, die Struktur der Komponenten, ihre Wechselwirkungen, ihre Prinzipien und Richtlinien, die ihren Entwurf, ihre Entwicklung und Implementierung steuern.

Bei größeren Konzernen können durchaus mehrere unterschiedliche Enterprise Architekturen parallel zueinander existieren. In allen Fällen jedoch überspannt eine Enterprise Architektur immer mehrere technische Systeme.

Das erweiterte Konzept von Enterprise Architekturen geht zurück auf die Achtzigerjahre des letzten Jahrhunderts. Einer der führenden Köpfe der Architekturbewegung, John Zachman, erkannte den Wert der Nutzung einer abstrakten Architektur für die Integration von Systemen und ihrer Komponenten. Zachman entwickelte die Analogien zum Gebiet der klassischen Bauarchitektur und nutzte später Konzepte aus der Flugzeugindustrie, um Geschäftsprozessaspekte in seinem Zachman-Framework abzudecken. Seit diesen Anfängen sind eine Reihe von Frameworks publiziert worden, die alle dieselbe Zielsetzung verfolgen, ein Unternehmen strukturell zu beschreiben.

2.2 Warum Enterprise Architektur?

Der Einsatz einer Enterprise Architektur unterstützt jedes Unternehmen, welches auf Informationstechnologie angewiesen ist, in hohem Maße. Innerhalb einer Enterprise Architektur gibt es keine effektiven Grenzen bezüglich der Fähigkeit zum Informationsaustausch zwischen verschiedenen Unternehmensteilen. Eine Enterprise Architektur erhöht den Wert der akkumulierten Daten, da jetzt verschiedene Datenbestände abgeglichen werden können. Die Enterprise Architektur wird zur Roadmap, von der sich alle zukünftigen IT-Investitionen leiten lassen können, und schließt die Lücke zwischen den Geschäftsprozessanforderungen und der IT-Infrastruktur. Zusätzliche Gemeinsamkeiten in Bezug auf Sicherheit und Ausbildung der Mitarbeiter in

der Enterprise Architektur schaffen einen Mehrwert für den Einsatz einer Enterprise Architektur. Jedoch sollte auch klar sein, dass die Einführung und Implementierung einer Enterprise Architektur eine hohe Investitionsleistung in einem Unternehmen darstellt. Die meisten Unternehmen haben nicht die nötigen Ressourcen, um eine neue Enterprise Architektur komplett neu aus eigener Kraft, quasi durch Schaffung auf der grünen Wiese, zu erzeugen. Von daher ist die Migration und Koexistenz die einzig valide Alternative im normalen Alltag, da stets eine große Menge von Applikationen innerhalb des Unternehmens schon vorhanden sind.

Viele Entwickler betrachten Enterprise Architekturen als einen Zwang, der sie in der Implementierung von Geschäftsprozessen oder Software einengt und deren Befolgung zusätzliche Projektkosten produzieren würde. Auf das einzelne Projekt bezogen mag dies in Ausnahmefällen sogar korrekt sein, projektübergreifend jedoch erzeugen die Gemeinsamkeiten innerhalb der Enterprise Architekturen ein hohes Einsparpotenzial. Dieses ist nicht unbedingt innerhalb der Entwicklung, sondern immer im Bereich der Bereitstellung und Nutzung von Informationen geschäftsbereichsübergreifend vorzufinden.

Die Nutzung bekannter und gemeinsamer Enterprise Architekturen ermöglicht Skaleneffekte und limitiert die Projektrisiken, erhöht den Projekterfolg und kann auch die Vergleichbarkeit von Angeboten über zu erwerbende Software unterstützen, insofern auch das Interesse von Unternehmen, die nur Standardsoftware einsetzen, an einer Enterprise Architektur.

2.3 Treibende Kräfte

Bei einem recht komplexen und technischen Thema, wie dem von Enterprise Architekturen, stellt sich rasch die Frage: Was sind die treibenden Kräfte, welche hinter einer Enterprise Architektur stehen? Bevor wir diese Frage beantworten, sollten wir jedoch noch einen kleinen Unterschied klären: den zwischen einer Softwarearchitektur und einer Enterprise Architektur.

Eine Softwarearchitektur zeigt auf, in welche logische Teile ein in sich geschlossenes Softwarepaket zerlegt werden kann. Die Enterprise Architektur stellt die Summe aller Softwarepakete und ihrer Verteilung innerhalb eines Unternehmens dar. Beide Architekturen haben notwendigerweise gegenseitige Abhängig- und Gemeinsamkeiten. Zur Beurteilung einer Enterprise Architektur ist es jedoch nicht unbedingt nötig, die einzelne Softwarearchitektur en detail zu kennen. Am besten lässt sich dies mit einem Hochhaus vergleichen. Die Enterprise Architektur zeigt das Haus als Ganzes, während die Softwarearchitektur die Gliederung des einzelnen Stockwerkes beschreibt. Diese Parallele trägt noch weiter: So wie ein ungewöhnliches Stockwerk innerhalb eines Hochhauses das Empfinden des Gesamtbauwerkes stört, so stört auch eine abweichende Softwarearchitektur, zumindest lokal, die Enterprise Architektur.

Was sind nun die treibenden Kräfte hinter einer Enterprise Architektur? Wie nicht anders zu erwarten, liegen sie alle außerhalb der eigentlichen tech-

nischen Domäne. Sie sind sowohl im Bereich der Ökonomie als auch der Geschäftsprozesse des Unternehmens vorzufinden. Zu diesen Kräften zählen neben der Wirtschaftlichkeit und Effizienz die Lebensdauer der Applikationen, Flexibilität, Skalierbarkeit, Offenheit, Interoperabilität, Modularität, Wartbarkeit und Zukunftssicherheit sowie auch das allgemeine Problem des Outsourcings.

Alle diese Kräfte und Zwänge wirken auf ein Unternehmen ein und bestimmen die Notwendigkeit einer Enterprise Architektur.

2.4 Lebensdauer von Applikationen

Innerhalb des letzten Jahrzehnts haben Architekturen in großen Softwaresystemen immer mehr an Bedeutung gewonnen. Dieser Trend liegt darin begründet, dass die einzelne Applikation eine geringere Lebensdauer im Vergleich zu früher hat. Mitte der 70er-Jahre konnte man von einer mittleren Lebenserwartung von etwa 10 bis 15 Jahren für eine große Applikation, beispielsweise ein Buchhaltungssystem, ausgehen. Heute ist diese Lebenserwartung für Buchhaltungssysteme auf 5 Jahre geschrumpft. Einzelne Applikationen mögen durchaus noch länger einsetzbar sein, im Mittel jedoch geht der Trend eindeutig zu immer kürzeren Zeiten.

Der Schlüsselfaktor für diese Entwicklung ist sowohl direkt als auch indirekt unsere Wirtschaft. Der zunehmende technische Fortschritt speziell im Bereich Informationstechnologie wie auch der Telekommunikation, bzw. seit Anfang der Neunzigerjahre der Internettechnologie, zwingen die Hardware- und Softwarehersteller zu immer kürzeren Produkt- und Releasezeiten. Diese Entwicklung übt massiven Druck auf die bestehende IT-Landschaft jedes Unternehmens aus. Auf dem PC-Sektor sind die Hardwarezeiten auf etwa 18 Monate gesunken, d.h., alle 18 Monate kann mit einer neuen Chipgeneration mit neuen Treibern und Eigenschaften gerechnet werden. Umgekehrt werden alte Systeme und Treiber mit fast der gleichen Geschwindigkeit vom Markt genommen. Zwar werden, speziell von größeren Unternehmen, immer wieder Forderungen laut, bestehende Systeme länger zu unterstützen, aber auf Dauer ist der ökonomische Druck auf die Hersteller zu hoch. Manche große IT-Dienstleister gehen sogar so weit, sich an dem Hersteller ihrer Hardware finanziell zu beteiligen, aber selbst diese Einzelfälle zögern das Ende nur etwas hinaus, da diese Strategie sich für keinen der Beteiligten betriebswirtschaftlich rechnet. Diese Form der freiwilligen Solidarität hat sich – in der Vergangenheit – noch nie als effektiv herausgestellt.

Aber nicht nur die Hardware beeinflusst die Lebensdauer der IT-Landschaft eines Unternehmens. Auch die darunterliegenden Betriebssystemplattformen, Datenbanken und Middleware haben direkte Auswirkungen auf die Qualität und Stabilität der eingesetzten Applikationen. Betrachtet man die Releasezyklen und damit die Releasedauer von Datenbanken und Middleware, so scheint

sie sich heute auf etwa 3 Jahre einzupendeln. Auf dem Gebiet der Betriebssysteme ist es noch drastischer; die Marktdominanz von Microsoft resultiert in einem neuen Betriebssystem alle 24 Monate. Diverse Linux-Derivate sind nicht besser, in manchen Distributionen liegt die Releasedauer unter 12 Monaten, d.h., alle 12 Monate gibt es einen neuen Major Release.

Die Releasedauer, d.h. die Zeit von einem Releasewechsel zu dem nächsten, führt zu einer Lebenserwartung von etwa der zwei- bis dreifachen Releasedauer für das Herstellersystem. Kein ökonomisch denkender Softwarehersteller wird eine größere Zahl von verschiedenen miteinander konkurrierenden Betriebssystemen aus dem gleichen Haus lange unterstützen. Auf Dauer werden die für ihn veralteten Systeme vom Markt genommen, folglich erhält der Kunde weder Pflege noch Support für die nun veralteten Softwareteile.

Werden die Teile Betriebssysteme, Datenbanken und Middleware, für diese Betrachtung zählen JDBC/ODBC-Treiber zur Middleware, so ist hier keineswegs ein wechselseitiger Einfluss dieser Teile untereinander zu sehen. Die Betriebssysteme und Datenbanken sind für den Endbenutzer recht gut entkoppelt, wobei die Datenbankhersteller heute die notwendigen Anpassungen an die neuesten Betriebssystemvarianten in der Regel recht zeitnah zur Verfügung stellen. Aufgrund des Marktdrucks der Datenbankhersteller untereinander sind diese jedoch gezwungen, häufig neue Releases mit neuen Funktionalitäten zur Verfügung zu stellen, schon um ein Alleinstellungsmerkmal auf dem hart umkämpften Markt zu haben. Die Middleware spielt die Rolle des Nachzüglers, sie wartet die Veränderungen der beiden anderen ab, um sich dann anzupassen. Auch wenn bestimmte Funktionalitäten aktueller Datenbankreleases zurzeit nicht genutzt werden, so erfüllen sie doch ein latentes Bedürfnis und werden mit hoher Wahrscheinlichkeit von einer späteren Applikation bewusst eingesetzt und somit zu einer Vorbedingung für den Einsatz.

Die heute eingesetzten verteilten Systeme mit diversen Serverplattformen, sowie auch eine gewisse Anzahl von unterschiedlichen Clients, verstärken das Problem noch um einen Faktor, da sich die Hersteller der unterschiedlichen Plattformen nicht bezüglich ihrer jeweiligen Releasezyklen absprechen. Insofern hat die Verteilung nicht zur Lösung des Lebensdauerproblems direkt beigetragen. Wir werden jedoch später sehen, dass die Verteilung von Systemen die Gesamtstabilität durchaus erhöhen kann.

Die zweite große treibende Kraft bei der Reduktion der Lebenserwartung einer Applikation ist der Mitbewerber für das gleiche Marktsegment, in dem die jeweilige IT-Landschaft Unterstützung leistet. Informationstechnologie ist in Unternehmen nie Selbstzweck, sondern immer ein Mittel zum Zweck. Informationstechnologie stellt immer ein mögliches Mittel zur Umsetzung und Unterstützung von Geschäftsprozessen dar. Je besser der Prozess und die Produkte durch die IT-Landschaft unterstützt werden, desto wettbewerbsfähiger ist ein Unternehmen. Neben den rein betriebswirtschaftlichen Faktoren eines IT-Systems ist dessen Adaptivität extrem wichtig. Im Rahmen von immer kürzer werdender Produktzyklen ist neben der Veränderung der Produkte auch eine möglichst kleine Time-to-Market-Zeit für die Konkurrenzfähigkeit

eines Unternehmens auf dem Markt ausschlaggebend. Diese Zwänge gelten aber nicht nur für das Unternehmen selbst, sondern auch für die Hersteller der Applikationen. Im Fall von Individualsoftware ist es das Unternehmen selbst, im Falle von Standardsoftware der Lieferant, der möglichst rasch neue Produkte auf den Markt bringen und konsequenterweise alte vom Markt nehmen muss.

Neue oder drastisch veränderte Produkte sind aber nicht immer mit den bestehenden Applikationen abzudecken. Die Folge: Diese Applikationen müssen ersetzt oder durch neue ergänzt werden. Dieser Ersatz führt zum sofortigen Ende der vorhergehenden Applikation. Folglich sinkt die Lebensdauer der alten Applikation und damit auch die durchschnittliche Lebenserwartung aller Applikationen innerhalb der IT-Landschaft. Gleichgültig ob Individual- oder Standardsoftware, die Ersetzung findet statt und die Lebenserwartung ist effektiv drastisch gesunken. An dieser Stelle wird oft argumentiert, dass Individualsoftware länger lebt als Standardsoftware. Diese Vermutung erweist sich in der Praxis als falsch. Eine der treibenden Kräfte hinter der Veränderung, nämlich der Wechsel im Bereich Betriebs- und Datenbanksysteme, trifft auf beide Softwareformen zu. Da Individualsoftware im Gegensatz zu Standardsoftware in der Praxis eine sehr geringe Pflegbarkeit aufweist, ist der Aufwand, sie lebensfähig zu halten, sehr viel höher als bei der Standardsoftware, wo der Lieferant über eine Skalenökonomie die Veränderungen wirtschaftlich verkraften kann.

Extrapolieren wir unsere Erfahrungen aus den letzten Jahrzehnten, so ist zu vermuten, dass sich diese Entwicklung nicht verlangsamen wird, folglich müssen wir uns darauf einstellen, dass innerhalb großer IT-Landschaften die durchschnittliche Lebensdauer einer Applikation etwa 4 Jahre beträgt. Heutige große IT-Dienstleister haben oft mehrere Hundert Applikationen in ihrer Gesamtlandschaft, welche von ihnen betreut werden. Selbst bei einer konservativen Schätzung von 200 Applikationen müssen in dem Gesamtsystem im Durchschnitt 4 Applikationen pro Monat ausgetauscht werden. Allein diese Zahlen sagen uns klar und deutlich, dass die einzelne Applikation nicht der stabilisierende Faktor innerhalb der IT-Landschaft sein kann. Einzig die von der einzelnen Applikation, dem einzelnen Produkt losgelöste Gemeinsamkeit, die Enterprise Architektur, bleibt unverändert und damit der stabilisierende Faktor.

2.5 Outsourcing

Unsere heutige Wirtschaft beruht auf der Grundlage, dass das Unternehmen, welches eine Tätigkeit am effektivsten, sei es über eine Skalen- oder eine Scope-Ökonomie, ausführt, einen immensen Wettbewerbsvorteil besitzt. Dieser Vorteil wird aber in der Regel durch zunehmende Spezialisierung erreicht. Das Angebot dieser Unternehmen an andere wächst ständig an. Diese hohe Spezialisierung führt dazu, dass das Unternehmen nur noch kleinere, aber besonders

wichtige, Operationen durchführt. Ist ein Unternehmen in der Lage, einen Teil seiner nicht essenziellen Prozesse an Dritte zu delegieren, die diese dann autonom ausführen, spricht man von Outsourcing.

Der Vorreiter auf dem Gebiet des Outsourcing ist der Produktionsprozess in der Automobilindustrie. Bei Automobilherstellern ist das Outsourcing so weit gediehen, dass ein großer Teil der Mitarbeiter am Fließband nicht mehr zur eigenen Belegschaft gehört, sondern von den Lieferanten der Teil- und Halbfertigprodukte gestellt wird. Die Automobilindustrie beschränkt sich nur noch auf wenige, aber essenzielle Teile.

Auch im Sektor von IT-Dienstleistungen ist dieser Ansatz zu beobachten. So können schon heute Funktionalitäten wie Adressprüfung oder Bonitätsauskünfte sowie auch Kreditbewertungen von Auskunfteien online genutzt werden. Diese Nutzungsmodelle sind erst der Anfang. Mit zunehmendem Kostendruck werden große Unternehmen wie Banken und Versicherungen weitere Teile ihres Geschäftsprozessablaufes an Dritte delegieren. Für dieses Outsourcing gibt es, auf technischer Basis, zwei grundlegende Szenarien. Zum einen hostet der Anbieter die Applikation im Sinne eines Application Service Providers, ASP, bei sich selbst, zum anderen stellt der Anbieter nur eine Betriebsmannschaft sowie auch die Applikation innerhalb des Gesamtsystems des Auftraggebers zur Verfügung, eine Art inhouse-Hosting.

Diese Fragmentierung der Geschäftsprozesse in verschiedene Hoheiten muss notwendigerweise durch die IT-Architektur unterstützt werden. Erst wenn die IT-Architektur solche Vorgehensweise ad hoc ermöglicht, erfüllt sie eine der notwendigen Voraussetzungen für den Einsatz innerhalb eines modernen Unternehmens. In späteren Abschnitten wird zu sehen sein, welche Auswirkungen diese Forderung auf die Enterprise Architektur hat.

2.6 Flexibilität und Skalierbarkeit

Welche Auswirkungen hat die zunehmend reduzierte Lebenserwartung der einzelnen Applikation auf die Anforderungen an die Enterprise Architektur?

Die Enterprise Architektur muss flexibel genug sein, um beliebig strukturierte Applikationen aufnehmen zu können. Diese Flexibilität ist eine der Schlüsselfaktoren für den Erfolg einer Enterprise Architektur. Sobald diese nicht mehr gewährleistet ist, bedarf der Einbau einer weiteren oder der Ersatz einer bestehenden Applikation stets eines immensen Kraftaufwandes mit sehr hohem Risiko.

Auch auf die Kostenfragen hat die Flexibilität einen großen Einfluss. Bedingt durch die immer kürzer werdenden Lebensdauern einzelner Applikationen wird der Integrations- und Einführungsaufwand sehr viel wichtiger als der Betriebsaufwand. Die klassischen Kriterien, in denen nur der Betrieb bewertet wird, werden dadurch immer stärker in den Hintergrund gerückt, da sie nur noch einen kleinen Teil an den effektiven Gesamtkosten einer Applikation einnehmen.

Neben der reinen Flexibilität zeigt sich im schnell ändernden Geschäftsleben, dass die Skalierbarkeit eine wichtige Größe darstellt. Skalierbarkeit im Softwaresinne bedeutet, dass das System in der Lage sein muss, sich auf geänderte Kapazitätsanforderungen dynamisch anzupassen, wobei diese Anforderungen sowohl plötzlich durch Hinzufügen neuer Applikationen als auch zeitlich durch Tagesschwankungen der Benutzer ausgelöst worden sein kann. Im Grunde ist die Skalierbarkeit eine betriebswirtschaftliche Größe, da ihr Vorhandensein erst eine Kalkulation der benötigten Ressourcen sowie auch der Kosten ermöglicht.

2.7 Offenheit und Interoperabilität

Wie schon bei der Diskussion um die Lebensdauer von Applikationen aufgezeigt, ändern sich die einzelnen Applikationen recht schnell. Damit diese aber in den Gesamtkontext integriert werden können, ist es notwendig, dass die Enterprise Architektur ein hohes Maß an Offenheit gegenüber den möglichen Formen von Applikationen besitzt. Folglich muss unsere Enterprise Architektur in der Lage sein, eine Menge von diversen Applikationen zu verkraften.

In der Praxis geht diese Forderung noch weiter. Da nicht überall dieselbe Enterprise Architektur vorhanden sein kann, muss jede Enterprise Architektur mit einer anderen zusammenarbeiten können, was sich in der Forderung nach Interoperabilität niederschlägt. Diese Interoperabilität ist nicht zu verwechseln mit der Interoperabilität, die vorliegt, wenn interne Teile der Gesamtarchitektur miteinander wechselwirken und noch nicht dem notwendigen Standard der Enterprise Architektur genügen. Solche Formen werden im Folgekapitel Migration und Integration besprochen. Das Management solcher Integrationsarchitekturen ist für ein Unternehmen eine immanent wichtige Disziplin, da sich die mögliche Ziel-Enterprise-Architektur im Laufe der Zeit, bedingt durch die technologische Entwicklung im Bereich Netzwerke, Festplatten etc., permanent verändert. Folglich befindet sich eine Enterprise Architektur in einem stetigen Wandel. Allerdings ist die Lebenserwartung einer gewählten Architektur deutlich höher als die Lebenserwartung einer Applikation. Vermutlich liegt die Lebenserwartung einer Enterprise Architektur in der Größenordnung von 10 Jahren.

2.8 Modularität

Die Modularitätsforderung an die Enterprise Architektur ist eine Forderung an die Fähigkeit, möglichst schnell und effektiv bestehende Applikationen zu ersetzen bzw. neue Applikationen zu integrieren. Interessanterweise wird die Fähigkeit, Applikationen zu entfernen und das Gesamtsystem stabil zu halten, als eine oft vernachlässigte Disziplin betrachtet. Die gewählte Enterprise Architektur muss die beliebte Plug-and-Play-Forderung gegenüber ganzen

Applikationen erfüllen. Für eine sinnvolle Enterprise Architektur ist es unabdingbar, dass es Mechanismen für die Ersetzung von Applikationen gibt. Neben den inneren Eigenschaften der Applikationen kann nur die Enterprise Architektur den notwendigen Rahmen setzen, um die entsprechenden Austauschaktivitäten vorzunehmen.

2.9 Wartbarkeit und Zukunftssicherheit

Obwohl betriebswirtschaftliche Größen generell eine Rolle spielen, treten Wartbarkeit und Zukunftssicherheit ganz klar in den Vordergrund. Beide Größen erscheinen mittlerweile als selbstverständlich, sind es jedoch nicht.

Sie sind schon allein deswegen wichtig, da sich Enterprise Architekturen ändern und somit die Frage nach der Zukunftssicherheit gestellt werden muss. Für ein Unternehmen ist es aber besser, sich der Herausforderung einer sich ändernden Enterprise Architektur zu stellen und den Wechselprozess bewusst zu managen, als von einer statischen Sicht auf die Enterprise Architektur auszugehen.

Warum sollen uns Enterprise Architekturen in der Zukunft helfen können? Ein großer Teil der Antworten beruht simplerweise auf Glaubensgrundsätzen in der Informationstechnologie, welche zwar nicht beweisbar, aber durch ständiges Wiederholen in Verkaufsgesprächen zu akzeptierten Tatsachen mutieren. Zu diesen Glaubensgrundsätzen zählen:

- Die Fähigkeit, durch eine Enterprise Architektur die gesamte Infrastruktur schnell zu ändern, ist ein großer Wettbewerbsvorteil gegenüber der Konkurrenz.
- Durch die Austauschbarkeit von Teilen oder ganzen Subsystemen sinken die Kosten für die Integration neuer Partner und den Austausch von bestehenden Lieferanten. Die Austauschbarkeit wird durch die Enterprise Architektur gewährleistet.
- Eine flexible Enterprise Architektur bietet die besten Services zu den niedrigsten Kosten.
- Eine gute Enterprise Architektur kann die bestehenden Systeme kostengünstig am Leben erhalten.

Diese Annahmen beinhalten im Grunde ein hohes Maß an Wunschenken. Wir werden im Nachfolgenden noch sehen, dass die Wirklichkeit doch etwas differenzierter und zum Teil widersprüchlicher ist.

2.10 Wiederverwendbarkeit

Eine gute Enterprise Architektur muss so entworfen sein, dass ihre Komponenten wiederverwendet werden können. Diese Wiederverwendung ist vorbereitet, vorgedacht und hochgradig erwünscht. Die Konsequenz aus dieser Forderung

nach Wiederverwendbarkeit gibt ein besonderes Gewicht auf die Services, speziell auch Service Oriented Architectures. Solche Services müssen nicht nur technischer Natur sein, sondern auch aus Geschäftsprozessteilen bestehen. Jede Funktionalität, die sich für eine Wiederverwendung eignen könnte, wird besonders vorteilhaft durch eine Service Oriented Architecture unterstützt.

2.11 Teile der Enterprise Architektur

Eine Enterprise Architektur im ausgedehnten Sinne besteht in ihrer Ganzheit aus vier separaten Teilbereichen:

- Geschäftsprozess-Architektur
- Applikationsarchitektur
- Datenarchitektur
- Systemarchitektur

Der Bereich der Geschäftsprozess-Architektur wird im vorliegenden Buch explizit ausgeklammert, da hier völlig andere Größen und Betrachtungsweisen notwendig sind, um zu einer sinnvollen Beschreibung zu kommen. Die drei anderen Architekturkategorien werden stets gemeinsam betrachtet, da eine explizite Separation für eine übergreifende Betrachtung nicht besonders sinnvoll erscheint. Die Abhängigkeiten und Wechselwirkungen dieser verschiedenen Teile sind, wie in Kapitel 3 aufgezeigt, viel zu groß.

Die in Abschn. 2.3 angesprochene Analogie zwischen der Enterprise Architektur und einem Gebäude trägt noch weiter. Die Architektur eines großen Gebäudes ist mehr als eine Anhäufung von Blaupausen. Holzmodelle und Außenansichten erklären die Grundzüge der Bauarchitektur dem Auftraggeber, damit dieser sich entscheiden kann, ob die Architektur den Bedürfnissen des Auftraggebers gerecht werden kann. Detailzeichnungen zeigen jeweils das Fundament, die Stockwerke und das Dach. Diese einzelnen Teile müssen sinnvollerweise immer im Kontext der anderen Teile entwickelt werden. Das Fundament muss die notwendige Stärke und Stabilität besitzen, um das gesamte Gebäude zu unterstützen. Je nach Anzahl der Stockwerke muss das Fundament umso stabiler ausfallen. Größere elektrische Anlagen benötigen stärkere Leitungen usw. Der Bau eines Gebäudes benötigt eine Reihe von Blaupausen, eine für die elektrischen Leitungen, eine für die Wasserversorgung, eine für die Heizungsanlage, eine für die Klimatisierung. Im Endeffekt aber müssen alle Blaupausen ein gemeinsames Bild für die Architektur bilden.

Die Situation ist für eine Enterprise Architektur ähnlich. Ein Stapel von Dokumenten ist noch lange keine Enterprise Architektur. Damit alle zusammen einen gemeinsamen Wert haben, müssen sie integriert werden. Die so entstandene integrierte Gesamtmenge mit all ihren Querbezügen und Referenzen bildet dann eine Enterprise Architektur.

Genauso wie die Blaupause eines Gebäudes dem jeweiligen Investor als Entscheidungsgrundlage dient, liefert die Enterprise Architektur die Entscheidungsgrundlage für die jeweilige Unternehmensleitung. Die Enterprise Architektur ermöglicht es dem Management eines Unternehmens über das Unternehmen als Ganzes nachzudenken.

Die obige Analogie trägt noch weiter. Ein Gebäude hat mehr als einen Nutzer. Die unterschiedlichen Nutzer haben zum Teil völlig verschiedene Erwartungen an die Struktur des Gebäudes. Außerdem ist ein solches Gebäude aus verschiedenen Baumaterialien, zum Teil auch durch Fertigprodukte, aufgebaut, je nach struktureller Notwendigkeit oder ökonomischem Zwang. Die Wahl der Materialien und des Bodens bestimmt die Stabilität, Performanz und Kosten des Gebäudes und muß beim Bau des Gebäudes berücksichtigt werden. Ähnlich ist auch ein Unternehmen aufgebaut. Auch hier entscheidet letztendlich die Qualität des Materials, sprich IT-Produkte, über die Stabilität und Performanz der IT-Landschaft.

Framework

*... he picked a bit of paper from the walk
and nailed it to the framework with a knife.*

The Bride comes to Yellow Sky,
Stephen Crane

3.1 Frameworks, Standards und Techniken

Die Unterstützung der Systementwicklung durch Architekturen ist in den letzten Jahren immer wichtiger geworden. Einige Standards und Techniken wurden schon von diversen Quellen vorgeschlagen. Ein Architektur-Framework zerlegt die komplexe Aufgabe in mehrere Teilschichten, welche dann getrennt modelliert werden können. Jede dieser Teilschichten, Layers, muss im Meta-modell des Frameworks spezifiziert sein.

Es ist, unabhängig vom konkreten Framework, immer sinnvoll, zwischen konzeptioneller und operationeller Sicht auf eine Architektur zu unterscheiden.

Die konzeptionelle Norm für Enterprise Architekturen ist der IEEE-Standard 1471. Dieser definiert eine theoretische Basis für die Definition, Analyse und Beschreibung einer Systemarchitektur, was einen Teil der Enterprise Architektur im engeren Sinne darstellt. Im Grunde beschreibt die IEEE-1471-Norm eine Reihe von Elementen und die Beziehungen, die diese Elemente untereinander haben.

Zusätzlich zur IEEE-1471-Norm existieren noch folgende drei Normen:

- ISO 14258: Beinhaltet die Konzepte und Regeln für das Modellieren von Enterprises mit dem Schwerpunkt auf den Geschäftsprozessen.
- ISO 15704: Voraussetzungen für Referenzarchitekturen und auch die entsprechenden Methodiken zur Entwicklung der Referenzarchitekturen.
- CEN ENV 4003: Das CIMOSA ist die Europeanorm des Comite Europeen de Normalisation für Systemarchitekturen.

Die beiden wohl bekanntesten Frameworks für die Entwicklung von Enterprise Architekturen sind:

- Das Zachman-Framework: Framework for Enterprise Architecture, welches eine logische Struktur zur Beschreibung und Klassifikation sowie der Organisation von Darstellungen liefert, die für die Entwicklung einer Architektur wichtig sind.