



Xpert.press

Dieter Masak

Moderne Enterprise Architekturen

 Springer



Xpert.press

Dieter Masak

Moderne Enterprise Architekturen

 Springer

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-Change-management	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-