

Software-Architektur

In dieser Reihe sind bisher erschienen:

Martin Backschat / Bernd Rücker
Enterprise JavaBeans 3.0
Grundlagen – Konzepte – Praxis

Peter Liggesmeyer
Software-Qualität
Testen, Analysieren und Verifizieren von Software

Michael Englbrecht
Entwicklung sicherer Software
Modellierung und Implementierung mit Java

Klaus Zeppenfeld
Objektorientierte Programmiersprachen
Einführung und Vergleich von Java, C++, C#, Ruby

Martin Backschat / Stefan Edlich
J2EE-Entwicklung mit Open-Source-Tools
Coding – Automatisierung – Projektverwaltung – Testen

Marco Kuhrmann / Jens Calamé / Erika Horn
Verteilte Systeme mit .NET Remoting
Grundlagen – Konzepte – Praxis

Peter Liggesmeyer / Dieter Rombach (Hrsg.)
Software Engineering eingebetteter Systeme
Grundlagen – Methodik – Anwendungen

Stefan Conrad / Wilhelm Hasselbring / Arne Koschel / Roland Tritsch
Enterprise Application Integration
Grundlagen – Konzepte – Entwurfsmuster – Praxisbeispiele

Ingo Melzer et al.
Service-orientierte Architekturen mit Web Services, 3. Auflage
Konzepte – Standards – Praxis

Oliver Vogel / Ingo Arnold / Arif Chughtai / Edmund Ihler /
Timo Kehrer / Uwe Mehlig / Uwe Zdun
Software-Architektur, 2. Auflage
Grundlagen – Konzepte – Praxis

Marco Kuhrmann / Gerd Beneken
Windows® Communication Foundation
Konzepte – Programmierung – Konzeption

Andreas Korff
Modellierung von eingebetteten Systemen mit UML und SysML

Oliver Vogel / Ingo Arnold / Arif Chughtai /
Edmund Ihler / Timo Kehrer / Uwe Mehlig /
Uwe Zdun

Software-Architektur

Grundlagen – Konzepte – Praxis

2. Auflage

Unter Mitwirkung von Markus Völter

Autoren:

Oliver Vogel, Ingo Arnold, Arif Chughtai, Edmund Ihler, Timo Kehrer, Uwe Mehlig,
Uwe Zdun

E-Mail: autoren@software-architektur-buch.de

Weiterführende Informationen zum Buch:

www.software-architektur-buch.de

Wichtiger Hinweis für den Benutzer

Der Verlag und die Autoren haben alle Sorgfalt walten lassen, um vollständige und akkurate Informationen in diesem Buch zu publizieren. Der Verlag übernimmt weder Garantie noch die juristische Verantwortung oder irgendeine Haftung für die Nutzung dieser Informationen, für deren Wirtschaftlichkeit oder fehlerfreie Funktion für einen bestimmten Zweck. Ferner kann der Verlag für Schäden, die auf einer Fehlfunktion von Programmen oder ähnliches zurückzuführen sind, nicht haftbar gemacht werden. Auch nicht für die Verletzung von Patent- und anderen Rechten Dritter, die daraus resultieren. Eine telefonische oder schriftliche Beratung durch den Verlag über den Einsatz der Programme ist nicht möglich. Der Verlag übernimmt keine Gewähr dafür, dass die beschriebenen Verfahren, Programme usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Der Verlag hat sich bemüht, sämtliche Rechteinhaber von Abbildungen zu ermitteln. Sollte dem Verlag gegenüber dennoch der Nachweis der Rechtsinhaberschaft geführt werden, wird das branchenübliche Honorar gezahlt.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

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

2. Auflage 2009

© Spektrum Akademischer Verlag Heidelberg 2009

Spektrum Akademischer Verlag ist ein Imprint von Springer

09 10 11 12 13

5 4 3 2 1

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Planung und Lektorat: Dr. Andreas Rüdinger, Barbara Lühker

Herstellung: Ute Kreutzer

Umschlaggestaltung: SpieszDesign, Neu-Ulm

Satz: Autorenatz

Layout/Gestaltung: Graphik & Text Studio, Barbing

Druck und Bindung: Krips b.v., Meppel

Printed in The Netherlands

ISBN 978-3-8274-1933-0

| Geleitwort

Seit vielen Jahren leite ich die *IT Architect Profession* bei der IBM in Europa. Meine Aufgabe ist es, die Entwicklung von IT-Architekten zu fördern und dafür zu sorgen, dass sie ihr Wissen auf dem neuesten Stand halten. Immer mehr Kunden und Mitbewerber sind daran interessiert, ihre eigenen Architektur-Fähigkeiten auszubauen. Die Open Group, ein Technologie- und Anbieter-unabhängiges Konsortium, bietet seit 2006 das sogenannte *Open Group Information Technology Architect Certification Program* an. Dieses wird inzwischen von vielen unserer Kunden und Mitbewerbern genutzt, um die Qualifikation ihrer Mitarbeiter entsprechend evaluieren zu können.

In diesem Zusammenhang bin ich erfreut über die neue Auflage dieses Buches. Es beschreibt und erklärt sehr anschaulich und gut strukturiert, was Architekten von IT-Systemen tun und womit sich die IT- respektive Software-Architektur überhaupt beschäftigt. Das Buch bietet somit eine gute Grundlage, um sich mit dem Thema vertraut zu machen und um seine Architektur-Fähigkeiten zu verbessern. Es passt genau in den Trend der Zeit, den ich sowohl in der Open Group als auch bei unseren Kunden und Mitbewerbern sehe. Es spiegelt die Denkweise wieder, die wir seit vielen Jahren in der IBM fördern und fordern.

Es ist eine sehr gute Zeit für IT-Architekten. Die IT- und Technologietrends entwickeln sich immer weiter und schneller. Eine Software-Architektur als Basis bei der Entwicklung von IT-Systemen ist zunehmend wichtiger geworden, um mit diesen rasanten Veränderungen umzugehen. Nicht zuletzt die ganze Diskussion rund um das Thema Service Oriented Architecture (SOA) hat das mehr als deutlich gemacht.

Deshalb finde ich das Buch sehr empfehlenswert für alle, die die Notwendigkeit erkannt haben, sich mit dem Thema Software-Architektur auseinander zu setzen. Es bietet einen sehr umfassenden Einstieg in das bewusste Architektur-Denken.

*Karin Dürmeyer
IBM Distinguished Engineer
IBM IOT Northeast IT Architect Profession Leader*

**Architektur-
Fähigkeiten werden
immer wichtiger**

**Dieses Buch hilft beim
Aufbau und Ausbau
dieser Fähigkeiten**

**Die Zeit ist reif, um in
das spannende Thema
einzusteigen...**

**...und ein architektoni-
sches Bewusstsein zu
entwickeln**

| Vorwort zur 2. Auflage

Seit dem Erscheinen der 1. Auflage dieses Buches vor nunmehr fast drei Jahren ist vieles geschehen. Architektur konnte sich als eigene Disziplin innerhalb der angewandten Informatik fest etablieren und *der Architekt* avancierte zur tragenden Rolle in großen oder komplexen Software-Projekten. Diese Entwicklung kann gut verglichen werden mit derjenigen im Bereich der klassischen Architektur, in der – ausgehend vom zupackenden Maurer, Zimmermann und Dachdecker – nachfolgend erst die Rolle des Architekten entstand, der als Generalist auch ästhetische, zeitliche und bautechnische Aspekte mit ganzheitlichem Blick integrierte. Auch innerhalb der Informatik vollzieht sich ein Wandel weg vom praktischen Handwerker hin zum konzeptuellen Planer. Kurzum: Das Thema *Architektur* hat signifikant an Bedeutung gewonnen.

Konzeptuell planerische, ausbildende oder organisatorische Beiträge haben in dem Maß an Bedeutung gewonnen, in dem technisches Spezialwissen in Länder ausgelagert wird, deren Lohnstrukturen und Expertenbasis diesen Trend fördern. Die Rolle des *Architekten* bildet in unseren Augen mit ihrer ganzheitlichen und integrativen Sicht auf die IT-Herausforderungen die Speerspitze einer neuen Generation von Ausbildungsprofilen innerhalb der Informatik und angrenzender Domänen.

Die Informatik ist eine der am schnellsten wachsenden und sich wandelnden Wissenschaften unserer Zeit. Drei Jahre Entwicklung im Bereich Informationstechnologie entsprechen Dekaden in vielen anderen Technologiebereichen. Einer unserer zentralen Wünsche für die vorliegende 2. Auflage war es daher, die architektonischen Entwicklungen, Veränderungen und Trends angemessen zu berücksichtigen und damit unser Buch auf den *neuesten Stand* zu bringen

Darüber hinaus haben wir in den Jahren seit Erscheinen der 1. Auflage viele Erfahrungen gemacht in der direkten und praktischen Arbeit mit unserem Buch. Wir setzten es z. B. ein, um in Projekten ein einheitliches Vokabular und Verständnis von Architektur zu etablieren; wir verwendeten es, um uns selber in weniger gut bekannten Wissensgebieten zu orientieren; wir stellten es in Vorträgen und an Konferenzen vor und diskutierten viele Aspekte desselben mit einem sehr interessierten Publikum; wir verwendeten es als Basis für Vorlesungen an verschiedenen Hochschulen und erhielten zahlreiche E-Mails von unseren Lesern mit wertvollen Rückmeldungen. Die Verbesserungen, die wir aus

Feste Etablierung von Architektur als eigene Disziplin in der Informatik

Der Architekt als zukunftsorientiertes Rollenprofil

**Motivation 1:
Buch auf den *neuesten Stand* bringen – aktualisieren, korrigieren, entfernen.**

**Motivation 2:
Eigene, praktische Erfahrungen mit dem Buch einbringen**

unseren eigenen Erfahrungen ableiteten sowie die Rückmeldungen und Erweiterungswünsche unserer Leser waren uns ein weiteres starkes Motiv, eine 2. Auflage zu veröffentlichen.

Motivation 3: Konsistenz und didaktische Qualität verbessern

Der letzte wichtige Anlass für uns, das Werk noch einmal zu verbessern, war unser eigener Qualitätsanspruch. Wir haben daher für die 2. Auflage sehr großen Wert gelegt auf eine weitere Steigerung der begrifflichen und inhaltlichen Konsistenz sowie die Ergänzung durch verständnisfördernde Mittel wie beispielsweise Concept-Maps oder Checklisten. Zudem haben wir die thematischen Beziehungen zwischen verwandten Buchabschnitten noch deutlicher gemacht.

Obwohl auch die 2. Auflage äußerst arbeitsintensiv war und uns Autoren wieder viele Stunden unserer Freizeit gekostet hat, sind wir alle doch froh, diesen Anlauf genommen zu haben, denn unser Buch hat dadurch noch einmal stark an Relevanz, Vollständigkeit und Qualität gewonnen.

Viel Freude beim Lesen

Wir hoffen, dass Sie das Buch in seiner überarbeiteten Form genauso interessiert, aufmerksam und begeistert lesen werden wie wir es in den vergangenen fast 12 Monaten überarbeitet haben. Wir wünschen Ihnen viel Freude bei der Lektüre.

Unser Dank

Abschließend wollen wir uns auch hier wieder bei all jenen bedanken, die uns für die Arbeit am vorliegenden Werk frei stellten und unterstützten. Dies waren unsere Partnerinnen und Kinder, unsere Eltern und Geschwister, unsere Freunde und Kollegen, unsere Arbeitgeber und Vorgesetzte. All denen möchten wir danken, die uns ihre Zeit opferten und immer wieder neue Kraft gaben.

Für die wertvollen Kommentare, Hilfe und Verbesserungsvorschläge möchten wir Kerstin Gans, Cordula Kleinschmidt, Johannes Blöcker, Martin Gottschalk, Peter Jess, Christoph Miksovic, Bernhard Polarzyk, Bernhard Scheffold, Gilbert Semmer, Martin Sieber sowie Marco Kuhmann von der Technischen Universität München ganz herzlich danken.

Ein besonderer Dank gilt Karin Dürmeyer für Ihre Worte, die sie uns und der 2. Auflage unseres Buches mit auf den Weg gegeben hat. Sie waren für uns ein wertvoller Antrieb.

Ebenso danken wir Barbara Lühker und Dr. Andreas Rüdinger von Spektrum Akademischer Verlag für ihre Unterstützung.

| Vorwort zur 1. Auflage

Im IT-Arbeitsalltag ist der Begriff „Software-Architektur“ bzw. ganz allgemein „Architektur“ allgegenwärtig. Auf Visitenkarten stehen Rollenbezeichnungen wie Software-, Sicherheits-, Daten- oder Systemarchitekt. Für Kunden werden Dokumente erstellt, die mit „Lösungsarchitektur“ überschrieben sind oder aber man befindet sich selbst in der Kundenrolle und vergibt Architektur-Aufträge an Lieferanten. Obwohl der Begriff „Architektur“ hierbei so häufig verwendet wird, verstehen (wenn man genau hinsieht) Architekten, Projektleiter oder Entwickler diesen Begriff nicht auf die gleiche Art und Weise.

Für manche von uns ist „Architektur“ die Auswahl und der Einsatz einer Technologie, für andere ist „Architektur“ vor allem ein Prozess, für viele ist „die Architektur“ eine Mappe mit Zeichnungen, auf denen miteinander verbundene geometrische Figuren zu sehen sind, für noch andere mag „Architektur“ schlicht all das sein, was „der Architekt“ produziert. Der Begriff „Architektur“ ist in seiner praktischen Verwendung breit ausgelegt – sprich: wenig einheitlich definiert oder verstanden. Dies macht es oft schwer, in der Architektur-Domäne und im Berufsalltag über einzelne Personen hinweg effizient zusammenzuarbeiten und zu kommunizieren.

Als wir uns entschlossen, ein Buch über Software-Architektur zu schreiben und in unser Vorhaben mit einer ersten Bestandsaufnahme starteten, lernten wir schnell, dass sich Software-Architektur selbst in einer scharf begrenzten Gruppe von erfahrenen Software-Architekten nicht so klar definieren ließ, wie wir das erwartet hatten. Wir stellten fest, dass – obwohl wir alle jahrelange Erfahrung mit dem Entwurf, der Beschreibung oder auch Überprüfung von Software-Architekturen hatten – wir doch nicht über ein einheitliches, präzises Ordnungs- und Begriffsverständnis in der Architektur-Domäne verfügten.

Je länger wir uns damit auseinandersetzten, desto mehr wurde uns die Herausforderung bewusst, die der Entwurf und die Beschreibung eines Erklärungsmodells bedeutete, mit dem wir in der Lage sein würden, das Thema Architektur differenziert zu betrachten und zu erklären. Auf der anderen Seite spürten wir, dass ein solches Erklärungsmodell genau das war, was uns fehlte. Ein Modell als fester Architektur-Bezugspunkt also, von dem aus wir das Architektur-Universum konsistent erschließen und auf das wir uns stets würden beziehen können.

Architektur als Begriff ist allgegenwärtig ...

... und vielseitig interpretiert ...

... zunächst auch in unserem Autorenteam

Unser Wunsch nach einem Erklärungsmodell ...

... und nach Orientierung

Wir erinnerten uns an die Zeit zurück, in der wir selber primär Software-Entwickler waren und mit dem Begriff Software-Architektur das erste Mal konfrontiert wurden. Wir erkannten ganz speziell in dieser Zeit unser Streben nach einem inneren und stabilen Erklärungsmodell, nach einem Satz von Mustern, die stabil und unabhängig von einer konkreten Lösung universell einsetzbar waren. Die Suche nach einem solchen Grundmodell, das die wichtigen Dimensionen der Architektur-Domäne aufdeckt, lief in jedem von uns lange Zeit unbewusst bzw. intuitiv ab. Am Anfang unserer „Reise durch die Informatik“ benötigten wir sehr viel Fach- und Detailwissen, konzentrierten wir uns auf den Erwerb von Wissen rund um Techniken und Technologien, Prozessmodelle, Methoden und Organisationen. Ständig und teilweise, ohne uns dies bewusst zu machen, leiteten wir im Laufe unseres Berufs- und damit Lern- und Ausbildungslebens aus dieser Sammlung isolierter Einzelerkenntnisse ein jeder für sich *sein* Erklärungsmodell der Architektur-Domäne ab. Nun waren wir endlich an den Punkt gekommen, an dem wir unsere individuellen Erklärungsmodelle miteinander abstimmen, gemeinsam formulieren und zum Kern unseres Buches machen konnten.

Unser architektonisches Denken entwickelte sich über die Zeit

Wir alle wussten, dass es nicht die eine Architekt-Klausur, das eine Architekt-Zertifikat gibt, das man bestehen oder erwerben kann, um sich anschließend ausgebildeter, diplomierter oder sonstwie zertifizierter Architekt nennen zu dürfen. Im Laufe unseres Informatiker-Lebens hatten wir alle bereits in vielen Rollen gearbeitet. Wir wussten als Entwickler, Tester, Projektleiter oder Designer, dass Architektur viele Gesichter hat, dass der Architektur-Aspekt für viele Rollen – nicht nur für den Architekten selbst – von entscheidender Bedeutung ist. Wir hatten aber auch die Erfahrung gemacht, dass wir neben der fachlichen Aus- und Weiterbildung zunächst einen Reife- und Reflektionsprozess durchleben mussten, bevor wir begannen, uns gegebenen Problem- wie auch Lösungsbetrachtungen aus einem stärker ganzheitlichen Blickwinkel zu nähern – bevor wir begannen, „architektonisch“ zu denken.

Unsere Buchvision

Mit unserem Buch verfolgen wir das primäre Ziel, Lesern Orientierung in der Architektur-Domäne zu ermöglichen. Viele Bücher über Architektur stellen in unseren Augen zu sehr das Thema Technologie in den Vordergrund der Betrachtung. Andere Bücher, die wir kennen, konzentrieren sich auf Architektur-Darstellungen und Nomenklaturen sowie die mit diesen verbundenen Techniken. Wieder andere Bücher betrachten Lösungsmuster für Architektur-Probleme. Und schließlich befassen sich regelmäßig einschlägige Computer-Magazine mit Projekterfahrungsberichten, in denen sehr häufig der Architektur-Aspekt einer

entsprechend vorgestellten Lösung dem Artikel erst Substanz gibt. Es strebte jedoch – in unseren Augen zumindest – keines dieser Werke an, dem Leser eine umfassende Orientierung zum Thema Architektur zu geben. Die uns bekannten Bücher konzentrieren sich jeweils lediglich auf ausgewählte Architektur-Teilgebiete.

Damit stellten wir Autoren uns zwei großen Herausforderungen. Die erste Herausforderung bestand darin, eine Buchstruktur zu entwerfen, welche die uns gleichermaßen wichtigen Aspekte Orientierung, Theorie und Praxis adressierte. Die zweite Herausforderung war für uns, ein Erklärungsmodell der Software-Architektur zu entwerfen und zu beschreiben, das uns im Folgenden erlaubte, die Vieldimensionalität dieses Themas angemessen aufzuarbeiten und als stabilen geistigen Kern für unser Buch zu nutzen. Das Ergebnis dieser ersten und grundlegenden Arbeit an unserem Buch war – wenn man so will – die Architektur des Buches selbst, die ausführlich in Kapitel 1 beschrieben wird und sich ganz grob wie folgt gliedert:

- > Darstellung der Architektur-Dimensionen und eines entsprechenden Ordnungs- und Orientierungsrahmens.
- > Inhaltlich weiterführende Aufarbeitung der Architektur-Dimensionen im Rahmen von Architektur-Theoriekapiteln.
- > Verbindungen zwischen Ordnungsrahmen- wie Theoriekapiteln aus dem Blickwinkel von Projekten in den Fallbeispielkapiteln dieses Buches.

Das vorliegende Buch ist demnach das Resultat unseres Wunsches nach einem Werk, welches den Themenkomplex Architektur sinnvoll strukturiert, an der Praxis orientiert ist sowie entsprechende Praxiserfahrungen vermittelt. Das Buch ist in besonderem Maße technologieneutral und zeitlos. Damit zählt dieses Buch für uns zu der Gruppe der Grundlagenwerke, die Ihnen ein stabiles Referenzsystem auch über aktuelle technologische Trends hinaus liefert. Die Aufgabe, die wir uns mit dem Schreiben dieses Buches gestellt hatten, war nicht leicht zu bewältigen und erforderte intensive und tiefe Auseinandersetzung aller Autoren mit dem Thema Architektur – und zwar über das sonst übliche Niveau eher isolierter Betrachtungen hinaus. In der Zeit, in der unser Buch entstand, haben wir sehr viel gelernt. Wir haben miteinander diskutiert und gerungen. Unser Autoren-Team gewann durch die gemeinsame Arbeit am vorliegenden Werk viele wertvolle neue Erkenntnisse und ein gemeinsames Architektur-Verständnis.

Unsere Herausforderungen

Unser Buch

Unser Verständnis von Architektur halten Sie nun in Ihren Händen. Wir hoffen, dass unser Anspruch, das Thema Architektur für Sie zu ordnen, zu erläutern und praktisch zu verankern, Ihnen hilft, sich in Ihrem Berufsleben oder Ihrem Studium mit diesem interessanten und wichtigen Gebiet zu befassen.

Unser Dank

Wir wollen uns an dieser Stelle bei all denen bedanken, die uns für die Arbeit am vorliegenden Werk frei stellten und uns unterstützten. Dies waren unsere Partner und Kinder, unsere Freunde und Kollegen, unsere Arbeitgeber und Vorgesetzte. All denen möchten wir danken, die uns ihre Zeit opferten und immer wieder neue Kraft gaben.

Für die wertvollen Kommentare, Hilfe und Verbesserungsvorschläge möchten wir Kerstin Gans, Dorothee Küpper, Christian Dennler, Martin Fabini, Martin Gottschalk, Peter Jess, Elmar Küpper, Arthur Neudeck, Bernhard Polarzyk, Bernhard Scheffold, Gilbert Semmer, Ralf Steck, Marco Kuhrmann sowie Bernd Oestereich ganz herzlich danken.

Ebenso danken wir Barbara Lühker und Dr. Andreas Rüdinger von Spektrum Akademischer Verlag für ihre Unterstützung.

| Inhaltsverzeichnis

1 Einleitung	1
1.1 Ausgangslage und Zielsetzung des Buches	2
1.2 Was ist Software-Architektur?	8
1.3 Leser-Leitfaden.....	11
1.3.1 Buchaufbau	11
1.3.2 Zielpublikum	15
1.3.3 Kapitelüberblick.....	15
1.3.4 Kapitel im Detail	17
2 Architektonischer Ordnungsrahmen	23
2.1 Motivation.....	24
2.2 Ordnungsrahmen im Überblick.....	26
2.3 Architekturen und Architektur-Disziplinen (WAS).....	30
2.4 Architektur-Perspektiven (WO)	31
2.5 Architektur-Anforderungen (WARUM)	32
2.6 Architektur-Mittel (WOMIT).....	33
2.7 Organisationen und Individuen (WER).....	36
2.8 Architektur-Vorgehen (WIE)	37
2.9 Zusammenfassung	38
3 Architekturen und Architektur-Disziplinen (WAS)	41
3.1 Klassische Architektur als Ausgangspunkt.....	42
3.2 Von der klassischen Architektur zur Software-Architektur	46
3.3 Architektur und der Systemgedanke.....	57
3.4 Architektur und die Bausteine eines Systems	62
3.5 Zusammenfassung	68
4 Architektur-Perspektiven (WO)	71
4.1 Architektur-Ebenen.....	72
4.1.1 Organisationsebene	80
4.1.2 Systemebene.....	81
4.1.3 Bausteinebene	82
4.2 Architektur-Sichten.....	83
4.2.1 Zachman-Framework	94
4.2.2 Reference Model for Open Distributed Processing	97
4.2.3 4+1-Sichtenmodell	98
4.3 Zusammenfassung	100

5 Architektur-Anforderungen (WARUM)	103
5.1 Allgemeines	104
5.2 Anforderungen im Überblick	107
5.3 Anforderungen im Detail	111
5.3.1 Organisationsanforderungen	111
5.3.2 Systemanforderungen	113
5.3.3 Bausteinanforderungen	114
5.3.4 Laufzeitanforderungen.....	114
5.3.5 Entwicklungszeitanforderungen	116
5.3.6 Organisatorische Rahmenbedingungen.....	118
5.4 Anforderungen im Architektur-Kontext.....	119
5.5 Zusammenfassung	123
6 Architektur-Mittel (WOMIT).....	125
6.1 Architektur-Prinzipien	128
6.1.1 Prinzip der losen Kopplung.....	130
6.1.2 Prinzip der hohen Kohäsion.....	133
6.1.3 Prinzip des Entwurfs für Veränderung.....	135
6.1.4 Separation-of-Concerns-Prinzip.....	137
6.1.5 Information-Hiding-Prinzip	140
6.1.6 Abstraktionsprinzipien	142
6.1.7 Modularitätsprinzip.....	145
6.1.8 Rückverfolgbarkeitsprinzip	148
6.1.9 Selbstdokumentationsprinzip.....	148
6.1.10 Inkrementalitätsprinzip	149
6.1.11 Weitere Architektur-Prinzipien.....	150
6.1.12 Zusammenfassung.....	151
6.2 Grundlegende architektonische Konzepte	152
6.2.1 Prozedurale Ansätze.....	153
6.2.2 Objektorientierung	155
6.2.3 Komponentenorientierung	161
6.2.4 Meta-Architekturen und Reflection	164
6.2.5 Generative Erzeugung von Systembausteinen	166
6.2.6 Modellgetriebene Software-Entwicklung.....	170
6.2.7 Aspektorientierung.....	181
6.2.8 Skriptsprachen und dynamische Sprachen	185
6.2.9 Wartung von Software-Architekturen	189
6.2.10 Zusammenfassung.....	193
6.3 Architektur-Taktiken, -Stile und -Muster.....	194
6.3.1 Qualitätsattributszenarien	196
6.3.2 Architektur-Taktiken.....	197
6.3.3 Architektur-Stile.....	199
6.3.4 Architektur-Muster.....	202

6.3.5 Mustersprachen	211
6.3.6 Zusammenfassung.....	215
6.4 Basisarchitekturen	216
6.4.1 Schichtenarchitekturen	217
6.4.2 Datenflussarchitekturen	219
6.4.3 Repositories	220
6.4.4 Zentralisierung gegenüber Dezentralisierung.....	221
6.4.5 n-Tier-Architektur.....	224
6.4.6 Rich Client gegenüber Thin Client.....	226
6.4.7 Peer-to-Peer-Architektur.....	228
6.4.8 Publish/Subscribe-Architektur.....	228
6.4.9 Middleware	229
6.4.10 Komponentenplattformen	233
6.4.11 Serviceorientierte Architekturen.....	235
6.4.12 Sicherheitsarchitekturen.....	243
6.4.13 Zusammenfassung	252
6.5 Referenzarchitekturen.....	253
6.5.1 Definition und Bestandteile	254
6.5.2 Einsatz und Vorteile von Referenzarchitekturen	256
6.5.3 Anforderungen an Referenzarchitekturen	257
6.5.4 Arten von Referenzarchitekturen	257
6.5.5 Beispiel für eine Referenzarchitektur	258
6.5.6 Zusammenfassung.....	263
6.6 Architektur-Modellierungsmittel.....	264
6.6.1 Grundlegende Konzepte der Modellierung	265
6.6.2 Unified Modeling Language (UML).....	268
6.6.3 Domain Specific Languages (DSL)	276
6.6.4 Architecture Description Languages (ADL).....	279
6.6.5 Unified Method Architecture (UMA)	283
6.6.6 Zusammenfassung.....	290
6.7 Architekturrelevante Technologien.....	291
6.7.1 Middleware-Systeme	292
6.7.2 Datenbanken und Persistenz von Geschäftsobjekten	297
6.7.3 Datenaustausch und Datentransformation mit XML.....	300
6.7.4 Dynamische Web-Seiten und Web-Anwendungsserver	303
6.7.5 Komponentenplattformen	305
6.7.6 Web Services	308
6.7.7 Zusammenfassung.....	310
7 Organisationen und Individuen (WER).....	311
7.1 Allgemeines	312
7.2 Organisationen.....	316
7.3 Individuen.....	321

7.4 Individuen und Gruppen.....	324
7.5 Architektur und Entscheidungen	328
7.6 Architekt als zentrale Rolle	332
7.7 Zusammenfassung	337
8 Architektur-Vorgehen (WIE).....	341
8.1 Architektur und Entwicklungsprozesse	342
8.2 Architektonisches Vorgehen im Überblick.....	350
8.3 Erstellen der Systemvision	357
8.4 Verstehen der Anforderungen.....	367
8.5 Entwerfen der Architektur.....	377
8.6 Umsetzen der Architektur	406
8.7 Kommunizieren der Architektur	413
8.8 Anwendungsszenario: Enterprise Application Integration	428
8.8.1 Erstellen der Systemvision.....	430
8.8.2 Verstehen der Anforderungen.....	432
8.8.3 Entwerfen der Architektur.....	435
8.8.4 Kommunizieren und Umsetzen der Architektur.....	447
9 Risikofallmanagementsystem.....	449
9.1 Überblick.....	450
9.2 Architektur-Anforderungen (WARUM)	451
9.2.1 Systemvision.....	451
9.2.2 Organisationsanforderungen	451
9.2.3 Systemanforderungen	452
9.2.4 Bausteinanforderungen	456
9.3 Architekturen und Architektur-Disziplinen (WAS).....	460
9.3.1 Disziplinen	460
9.3.2 Entscheidungen zur Software-Architektur	461
9.4 Architektur-Perspektiven (WO)	462
9.4.1 Systemebene.....	462
9.4.2 Bausteinebene	463
9.5 Architektur-Mittel (WOMIT).....	465
9.5.1 Architektur-Prinzipien	465
9.5.2 Grundlegende architektonische Konzepte	467
9.5.3 Generative und generische Verfahren	467
9.6 Organisationen und Individuen (WER).....	470
9.6.1 Organisation	470
9.6.2 Individuen	471
9.7 Architektur-Vorgehen (WIE)	472

10 CRM-Kundendatenbank	473
10.1 Überblick	474
10.2 Architektur-Anforderungen (WARUM)	475
10.2.1 Ausgangssituation	476
10.2.2 Anforderungen	478
10.2.3 Anwendungsfälle	481
10.2.4 Architekturrelevante Anforderungen	482
10.3 Architekturen und Architektur-Disziplinen (WAS)	484
10.3.1 Disziplinen	484
10.3.2 Architektonische Entscheidungen	485
10.3.3 Entscheidungen zur Software-Architektur	489
10.4 Architektur-Perspektiven (WO)	493
10.5 Architektur-Mittel (WOMIT)	494
10.6 Organisationen und Individuen (WER)	495
10.7 Architektur-Vorgehen (WIE)	496
10.8 Fazit	497
 Glossar	499
 Abkürzungsverzeichnis	523
 Literaturverzeichnis	528
 Index	546

| Verzeichnis der Autoren

Oliver Vogel ist zertifizierter IT-Architekt und -Berater bei IBM Global Business Services. Sein Tätigkeitsfeld umfasst die Leitung, Schulung und Beratung von internationalen Projekten und Kunden in diversen Architektur-Themen, wie beispielsweise Architektur-Entwurf, -Umsetzung, -Beurteilung und -Governance. Darüber hinaus beschäftigt er sich intensiv mit modellgetriebener Software-Entwicklung, serviceorientierten Architekturen und Offshoring. Neben Software-Architektur ist Enterprise-Architektur für ihn ein weiterer Interessensschwerpunkt. In seiner Freizeit engagiert er sich als Referent, Dozent und Autor in den genannten Themengebieten.

Ingo Arnold arbeitet als Enterprise-Architekt für die Novartis AG in der Schweiz und ist dort zuständig für die globale Architektur-Planung zentraler Betriebsplattformen. Darüber hinaus gibt er als Dozent Vorlesungen in den Gebieten Software-Architektur, Software Engineering und Software Design Patterns an der Berufsakademie Lörrach sowie der Universität Basel. Auch stellt Ingo Arnold auf internationalen Konferenzen regelmäßig ausgewählte Themen seines Wirkungsbereiches, wie beispielsweise SOA oder Sicherheitsarchitekturen, einem breiten Publikum vor.

Arif Chughtai ist als selbständiger IT-Berater und -Trainer tätig. Sein besonderes Interesse gilt der Verbesserung der technischen Software-Qualität. Software-Engineering gehört deshalb zu den zentralen Gegenständen seiner Arbeit. Er beschäftigt sich dabei insbesondere mit Software-Architektur und -Entwurf, objektorientierter und modellgetriebener Software-Entwicklung sowie serviceorientierten Architekturen. Teile aus den aufgeführten Themenfeldern lässt er als Autor, Dozent und Referent regelmäßig in Fachartikel, Vorlesungen und Vorträge einfließen.

Edmund Ihler war zunächst in der Informatikforschung und später als Architekt und Projektmanager in der Software-Entwicklung für Banken und Versicherungen tätig. Seit 2000 lehrt er als Professor für Informatik an der Hochschule der Medien in Stuttgart mit den Schwerpunkten objektorientierte Software-Modellierung und modellgetriebenes Software Engineering.

Timo Kehrer promoviert an der Universität Siegen und ist Mitarbeiter der Fachgruppe Praktische Informatik an der Fakultät Elektrotechnik und Informatik. Seine Forschungsgebiete sind modellgetriebene Software-Entwicklung, Modell-Versionsmanagement, Modell-Evolution und Analyse von Modell-Repositories. Zeitgleich ist er derzeit noch wissenschaftlicher Mitarbeiter im Studiengang Medieninformatik an der Hochschule der Medien in Stuttgart. Die Schwerpunkte seiner Lehrtätigkeit liegen auf der objektorientierten Programmierung und Modellierung. Während des Studiums, insbesondere im Rahmen seiner Diplomarbeit, spezialisierte er sich auf dem Gebiet der Software-Modellierung.

Uwe Mehlig ist als IT-Architekt bei der IBM Deutschland GmbH im Bereich Global Business Services tätig. Sein aktueller Schwerpunkt liegt auf dem Entwurf von Integrationslösungen basierend auf offenen Standards wie XML, SOAP und Web Services.

Uwe Zdun ist Universitätsassistent in der Distributed Systems Group an der Technischen Universität Wien. Er hat 2001 an der Universität Essen in Informatik promoviert und 2006 an der Wirtschaftsuniversität Wien habilitiert. Seine Forschungsgebiete sind Software Patterns, Software-Architektur, Language Engineering, SOA, verteilte Systeme und Objektorientierung. Er hat viele Software-Systeme entwickelt, darunter Open-Source-Systeme wie die Sprachen Frag und Extended Object Tcl (XOTcl) und kommerzielle Systeme. Darüber hinaus war er in zahlreichen Beratungs- und Forschungsprojekten tätig. Er ist Autor zahlreicher Forschungspublikationen und Koautor des Buches „Remoting Patterns“. Uwe Zdun ist Associate Editor-in-Chief für das Magazin IEEE Software und European Editor für das Journal Transactions on Pattern Languages of Programming (TPLoP).