

Helmut Balzert

# Lehrbuch der Softwaretechnik

## Basiskonzepte und Requirements Engineering

3. Auflage

Lehrbücher der Informatik

**Spektrum**  
AKADEMISCHER VERLAG

# Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering

# Lehrbücher der Informatik

Herausgegeben von  
Prof. Dr.-Ing. habil. Helmut Balzert

Heide Balzert  
Lehrbuch der Objektmodellierung  
Analyse und Entwurf mit der UML 2, 2. Auflage

Helmut Balzert  
Lehrbuch Grundlagen der Informatik  
Konzepte und Notationen in UML 2, Java 5, C# und C++,  
Algorithmik und Softwaretechnik, Anwendungen, 2. Auflage

Klaus Zeppenfeld  
Lehrbuch der Grafikprogrammierung  
Grundlagen, Programmierung, Anwendung

Helmut Balzert  
Objektorientierte Programmierung  
mit Java 5

Helmut Balzert  
Lehrbuch der Softwaretechnik:  
Softwaremanagement  
2. Auflage

Helmut Balzert

# Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering

3. Auflage

Unter Mitwirkung von  
Heide Balzert  
Rainer Koschke  
Uwe Lämmel  
Peter Liggesmeyer  
Jochen Quante

**Autor**

Prof. Dr. Helmut Balzert

E-Mail: hb@W3L.de

**Wichtiger Hinweis für den Benutzer**

Der Verlag und der Autor 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](http://springer.de)

3. 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

Redaktion und Gestaltung: M.Sc. Kerstin Kohl, Witten; Dagmar Fraude, Witten

Umschlaggestaltung: SpieszDesign, Neu-Ulm

Titelbild: Anna Solecka-Zach: „Ohne Titel“ (1995)

Satz: W3L GmbH, Witten – automatischer Satz aus der W3L-Plattform.

ISBN 978-3-8274-1705-3

# Vorwort zur 3. Auflage

Die Softwaretechnik bildet einen Grundpfeiler der Informatik. Sie hat sich zu einem umfassenden Wissenschaftsgebiet entwickelt. Um Ihnen, liebe Leserin, lieber Leser, ein optimales Erlernen dieses Gebietes zu ermöglichen, habe ich – zusammen mit Koautoren – ein dreibändiges Lehr- und Lernbuch über die Softwaretechnik geschrieben. Es behandelt die Kerngebiete »Basiskonzepte und Requirements Engineering« (Band 1), »Entwurf und Architekturen« (Band 2) und »Softwaremanagement« (Band 3) sowie die Abhängigkeiten zwischen diesen Gebieten.

Ich habe das Gebiet der Softwaretechnik in mehrere große Bereiche gegliedert – der Tempel der Softwaretechnik (Abb. 0.0-1) veranschaulicht diese Gliederung. Ziel jeder Softwareentwicklung ist es, ein lauffähiges Softwareprodukt zu erstellen, zu warten und zu pflegen. Im Mittelpunkt der Gliederung steht daher die Softwareentwicklung (mittlere Säule). Jede der Aktivitäten der Softwareentwicklung trägt dazu bei, Teilprodukte zu erstellen, die dann in ein Gesamtprodukt münden.

Aufbau &  
Gliederung

Eine Softwareentwicklung läuft aber *nicht* von alleine ab. Sie basiert auf und nutzt Basistechniken, das sind Prinzipien, Methoden, Werkzeuge, *Best Practices*.

Die eigentliche Softwareentwicklung wird durch die zwei Säulen »Softwaremanagement« und »Prozess- und Qualitätsmodelle« und das Dach »Allgemeines Management« eingerahmt. Bei den Aktivitäten des Managements und der Prozess- & Qualitätssicherung handelt es sich um *begleitende* Aktivitäten, deren Ergebnisse aber selbst *nicht* Bestandteil des Endprodukts sind. Dennoch sind sie eminent wichtig – was die große Anzahl fehlgeschlagener Softwareprojekte deutlich zeigt.

In diesem Buch werden die Basistechniken, die Basiskonzepte und das *Requirements Engineering* behandelt. Dieses Buch besteht aus folgenden Teilen:

**I Die Wissenschaftsdisziplin Softwaretechnik**

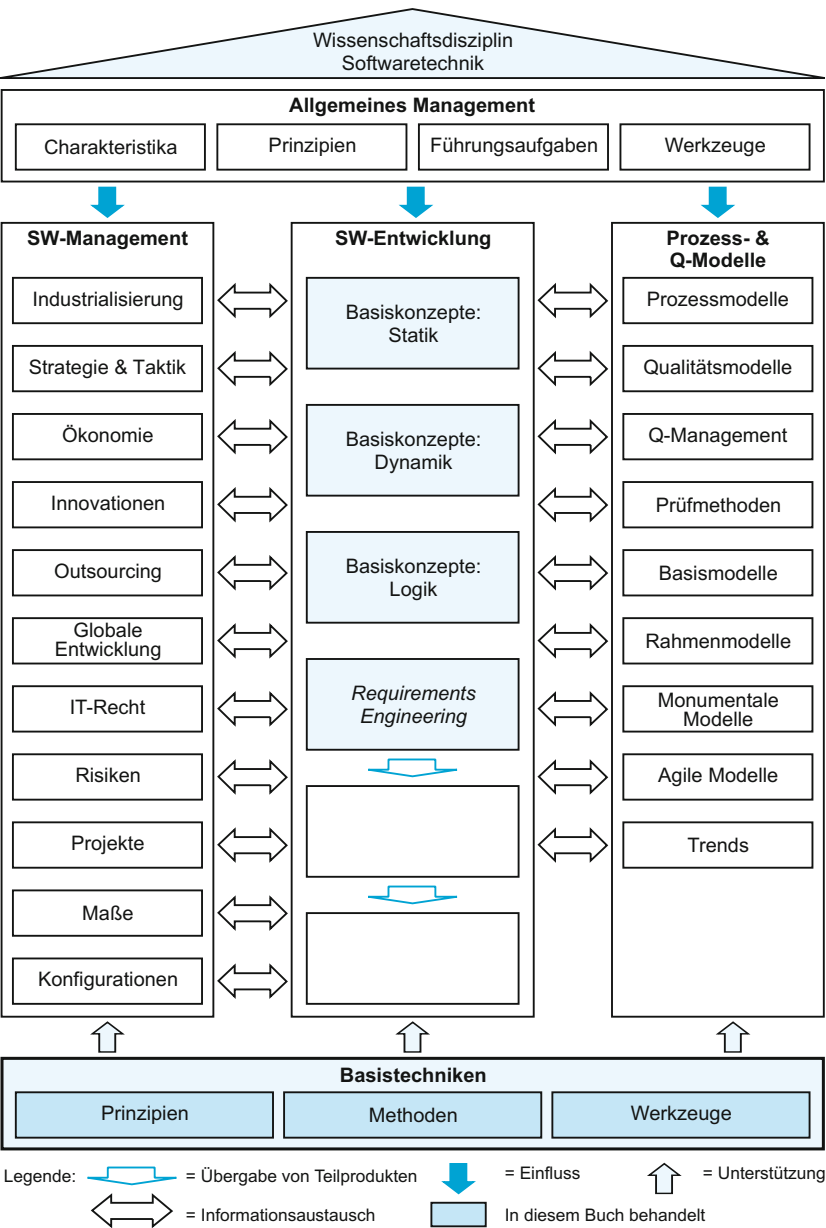
**II Basistechniken**

**III Basiskonzepte**

**IV Requirements Engineering**

Vorwort zur 3. Auflage

Abb. 0.0-1:  
Gliederung der  
Softwaretechnik.



Zu einem dreibändigen Lehrbuch der Softwaretechnik gehört natürlich auch eine Reflexion über die Wissenschaftsdisziplin:  
■ »Die Wissenschaftsdisziplin Softwaretechnik«, S. 1  
Wie in anderen dynamischen Disziplinen auch, so gibt es auch in der Softwaretechnik Trends und »Hypes«. Nach einer Euphoriephase erfolgt dann oft die Ernüchterung. Für die Lehre und das Lernen ist es

daher wichtig, das Konstante in der Softwaretechnik zu identifizieren – das unabhängig von aktuellen Trends gilt. Ich habe allgemeingültige Prinzipien, Methoden und Werkzeuge unter dem Oberbegriff Basistechniken zusammengefasst:

■ »Basistechniken«, S. 23

Analog habe ich versucht, bei den heute in der Softwaretechnik angewandten Konzepten die grundlegenden Konzepte zu isolieren und in die drei Kategorien Statik, Dynamik und Logik einzuordnen:

■ »Basiskonzepte«, S. 99

Softwaresysteme lassen sich grob in Informationssysteme und softwareintensive Systeme gliedern. Bei softwareintensiven Systemen ist die Software nur ein – wenn auch wichtiger – Bestandteil des Systems. Je nachdem, um was für eine Art von Softwaresystem es sich handelt, werden die Basiskonzepte unterschiedlich intensiv und in verschiedenen Kombinationen eingesetzt.

Um diese Unterschiede zu verdeutlichen, werden zwei Fallstudien verwendet, auf die sich in den Beispielen zu den Basiskonzepten immer wieder bezogen wird:

■ »Fallstudie: SemOrg – Die Spezifikation«, S. 107

■ »Fallstudie: Fensterheber – Die Spezifikation«, S. 117

Zu wissen, was der Kunde will, und dies zu ermitteln, zu beschreiben, zu spezifizieren, zu analysieren und in Form einer fachlichen Lösung zu modellieren ist die Aufgabe des *Requirements Engineering* – im Deutschen auch als Systemanalyse bezeichnet. Da im *Requirements Engineering* viele Basistechniken und Basiskonzepte benötigt und eingesetzt werden, habe ich die Basistechniken und Basiskonzepte an den Anfang des Buches gestellt. Das *Requirements Engineering* wird oft noch unterschätzt – hier werden jedoch die Grundlagen für den Erfolg oder Misserfolg einer Softwareentwicklung gelegt:

■ »Requirements Engineering«, S. 433

Die Inhalte der 2. Auflage wurden in zwei Bände aufgeteilt, um die Bücher handlicher und zielgruppenorientierter zu gestalten. Im Gegensatz zur 2. Auflage ist die 3. Auflage unabhängig von einem Vorgehensmodell. Die Basistechniken und Basiskonzepte sind prozessneutral beschrieben und nach sachlogischen Gesichtspunkten strukturiert – das gilt auch für das *Requirements Engineering*.

Die Methoden SA (*Structured Analysis*) und SA/RT (*Real Time Analysis*) werden nicht mehr behandelt, da ihr Einsatz in der Praxis abnimmt. Das gesamte Gebiet der Software-Ergonomie wird aus Umfangsgründen nicht mehr behandelt – obwohl es zunehmend wichtiger wird. Das Gebiet ist jedoch inzwischen so umfangreich, dass es dazu eigene Lehrbücher gibt und an Hochschulen oft auch in eigenen Lehrveranstaltungen unterrichtet wird.

2 Fallstudien

Zur 3. Auflage



## Vorwort zur 3. Auflage



Kapitel aus der 2. Auflage, die nicht mehr in dieser neuen Auflage berücksichtigt werden konnten, finden Sie als *Open Content* auf der Website [www.W3L.de/OpenContent](http://www.W3L.de/OpenContent).

Buchaufbau

Das Buch besteht aus 28 Kapiteln. Jedes Kapitel besteht aus Unterkapiteln. Am Ende der meisten Kapitel befindet sich eine Zusammenfassung des behandelten Lehrstoffs. Am Ende des Buches sind das umfangreiche Glossar sowie das Literaturverzeichnis angeordnet. Dadurch kann das Buch auch ideal als Nachschlagewerk benutzt werden.

Didaktik &  
Methodik

Ziel der Didaktik ist es, einen Lehrstoff so zu strukturieren und aufzubereiten, dass der Lernende sich leicht ein mentales Modell von dem Lehrstoff aufbauen kann und genügend Übungsmöglichkeiten erhält, um zu überprüfen, ob er den Lehrstoff – nach der Beschäftigung mit ihm – entsprechend den vorgegebenen Lernzielen beherrscht. Dieses didaktische Ziel habe ich versucht in diesem Lehrbuch umzusetzen. Die Übungsmöglichkeiten befinden sich jedoch nicht im Lehrbuch, sondern in dem ebenfalls verfügbaren E-Learning-Kurs (siehe unten).

In den meisten Lehrbüchern wird die Welt so erklärt, wie sie ist – ohne dem Lernenden vorher die Möglichkeit gegeben zu haben, über die Welt nachzudenken. Ich stelle daher in vielen Kapiteln am Anfang an Sie eine Frage. Diese Frage soll Sie dazu anregen, über ein Thema nachzudenken. Erst nach dem Nachdenken sollten Sie weiter lesen. (Vielleicht sollten Sie die Antwort nach der Frage zunächst durch ein Papier abdecken).

Querverweise

Die Softwaretechnik ist komplex. Es gibt viele gegenseitigen Abhängigkeiten. Um diese Abhängigkeiten zu verdeutlichen, enthält dieses Buch viele (absolute) Querverweise auf andere Kapitel, damit Sie sich mit verschiedenen Perspektiven auf ein Themengebiet befassen können.

Einsatz des  
Buches



Dieses Buch kann zur Vorlesungsbegleitung, zum Selbststudium und zum Nachschlagen verwendet werden. Um den Umfang und die Kosten des Buches zu begrenzen, enthält dieses Buch *keine* Tests und Aufgaben. Für diejenigen Leser unter Ihnen, die Ihr Wissen durch Tests und Aufgaben aktiv überprüfen möchten, gibt es einen (kostenpflichtigen) E-Learning-Online-Kurs. Mentoren und Tutoren betreuen Sie bei der Bearbeitung von Tests und Aufgaben. Das Bestehen eines Abschlusstests und einer Abschlussklausur wird durch Zertifikate dokumentiert. Dieser Online-Kurs ist Bestandteil des Online-Bachelor-Studiengangs »Web- und Medieninformatik« der FH Dortmund. Sie finden den Kurs auf der W3L-Plattform (<http://www.W3L.de>).

Kostenloser  
E-Learning-Kurs

Ergänzend zu diesem Buch gibt es den kostenlosen E-Learning-Kurs »Prinzipien der Softwaretechnik«, der zusätzlich zahlreiche Tests enthält, mit denen Sie Ihr Wissen überprüfen können. Sie fin-

## Vorwort zur 3. Auflage

den den Kurs auf der E-Learning-Plattform [www.W3L.de](http://www.W3L.de). Bitte wählen Sie auf dem Reiter »Online-Kurse« den Link »Zur TAN-Einlösung«. Registrieren Sie sich als neuer Benutzer und geben Sie anschließend folgende Transaktionsnummer (TAN) ein: 3645112138.

Dieses Buch ist für folgende Zielgruppen geschrieben:

Zielgruppen

- Studierende der Informatik und Softwaretechnik an Universitäten, Fachhochschulen und Berufsakademien.
- Software-Ingenieure, Softwaremanager und Software-Qualitätssicherer in der Praxis.

Vorausgesetzt werden Kenntnisse, wie sie normalerweise in einer Einführungsvorlesung zur Informatik vermittelt werden.

Vorkenntnisse

Zur Vermittlung der Lerninhalte werden Beispiele und Fallstudien verwendet. Um Ihnen diese unmittelbar kenntlich zu machen, sind sie in blauer Schrift gesetzt.

Beispiele,  
Fallstudien

Da ein Bild oft mehr aussagt als 1000 Worte, habe ich versucht, möglichst viele Sachverhalte zu veranschaulichen.

Visualisierung

In diesem Lehrbuch wurde sorgfältig überlegt, welche Begriffe eingeführt und definiert werden. Ziel ist es, die Anzahl der Begriffe möglichst gering zu halten. Alle wichtigen Begriffe sind im Text halbfett und blau gesetzt. Die so markierten Begriffe sind am Ende des Buches in einem Glossar alphabetisch angeordnet und definiert. Dabei wurde oft versucht, die Definition etwas anders abzufassen, als es im Text der Fall war, um Ihnen noch eine andere Sichtweise zu vermitteln.

Begriffe, Glossar  
halbfett, blau

Um in einem Buch deutlich zu machen, dass Leser und Leserinnen gemeint sind, gibt es verschiedene Möglichkeiten für den Autor:

Weibliche vs.  
männliche Anrede

- 1** Man formuliert Bezeichnungen in der 3. Person Singular in ihrer männlichen Form. In jüngeren Veröffentlichungen verweist man in den Vorbemerkungen dann häufig darauf, dass das weibliche Geschlecht mit gemeint ist, auch wenn es nicht im Schriftbild erscheint.
- 2** Man redet beide Geschlechter direkt an, z. B. Leserinnen und Leser, man/frau.
- 3** Man kombiniert die beiden Geschlechter in einem Wort, z. B. StudentInnen.
- 4** Man wechselt das Geschlecht von Kapitel zu Kapitel.

Aus Gründen der Lesbarkeit und Lesegewohnheit habe ich mich für die 1. Variante entschieden. Die Variante 4 ist mir an und für sich sehr sympathisch, jedoch steigt der Aufwand für den Autor beträchtlich, da man beim Schreiben noch nicht die genaue Reihenfolge der Kapitel kennt.

### Vorwort zur 3. Auflage

Als Begleitunterlage & zum Selbststudium	Bücher können als Begleitunterlage oder zum Selbststudium ausgelegt sein. In diesem Buch versuche ich einen Mittelweg einzuschlagen. Ich selbst verwende das Buch als begleitende und ergänzende Unterlage zu meinen Vorlesungen. Viele Lernziele dieses Buches können aber auch im Selbststudium erreicht werden.
Englische vs. deutsche Begriffe	Ein Problem für ein Informatikbuch stellt die Verwendung englischer Begriffe dar. Da die Wissenschaftssprache der Softwaretechnik Englisch ist, gibt es für viele Begriffe – insbesondere in Spezialgebieten – keine oder noch keine geeigneten oder üblichen deutschen Fachbegriffe. Auf der anderen Seite gibt es jedoch für viele Bereiche der Softwaretechnik sowohl übliche als auch sinnvolle deutsche Bezeichnungen, z. B. Entwurf für <i>Design</i> . Da mit einem Lehrbuch auch die Begriffswelt beeinflusst wird, bemühe ich mich in diesem Buch, sinnvolle und übliche deutsche Begriffe zu verwenden. Ist anhand des deutschen Begriffs <i>nicht</i> unmittelbar einsehbar oder allgemein bekannt, wie der englische Begriff lautet, dann wird in Klammern und kursiv der englische Begriff hinter dem deutschen Begriff aufgeführt. Dadurch wird auch das Lesen der englischsprachigen Literatur erleichtert.
Englische Begriffe kursiv gesetzt	Gibt es noch keinen eingebürgerten deutschen Begriff, dann wird der englische Originalbegriff verwendet. Englische Bezeichnungen sind immer <i>kursiv</i> gesetzt, sodass sie sofort ins Auge fallen.
Lesen des Buches: sequenziell	Ziel der Buchgestaltung war es, Ihnen als Leser viele Möglichkeiten zu eröffnen, dieses Buch nutzbringend für Ihre eigene Arbeit einzusetzen. Sie können dieses Buch sequenziell von vorne nach hinten lesen. Die Reihenfolge der Kapitel ist so gewählt, dass die Voraussetzungen für ein Kapitel jeweils erfüllt sind, wenn man das Buch sequenziell liest.
Nach Teildisziplinen	Eine andere Möglichkeit besteht darin, jeweils eine der Teildisziplinen »Basistechniken«, »Basiskonzepte«, » <i>Requirements Engineering</i> « durchzuarbeiten. Auf Querbezüge und notwendige Voraussetzungen wird jeweils hingewiesen.
Themenbezogen	Außerdem kann das Buch themenbezogen gelesen werden. Möchte man sich in die Grundlagen der statischen Basiskonzepte einarbeiten, dann kann man die dafür relevanten Kapitel durcharbeiten. Will man sich auf das <i>Requirements Engineering</i> konzentrieren, dann kann man auch nur diese Kapitel lesen.
Punktuell	Durch das Buchkonzept ist es natürlich auch möglich, punktuell einzelne Kapitel durchzulesen, um eigenes Wissen zu erwerben, aufzufrischen und abzurunden, z. B. Durchlesen des Kapitels über Petrinetze.
Zum Nachschlagen	Durch ein ausführliches Sachregister und Glossar sowie durch Zusammenfassungen kann dieses Buch auch gut zum Nachschlagen verwendet werden.

Soll das Buch begleitend zu einem Softwareprojekt eingesetzt werden, dann kann zunächst der »Requirements Engineering«, S. 433, behandelt werden. Parallel zum Projektverlauf können dann die für das Projekt relevanten Basiskonzepte behandelt werden.

Projekt-  
bezogen

Ich habe versucht, ein innovatives wissenschaftliches Lehrbuch der Softwaretechnik zu schreiben. Ob mir dies gelungen ist, müssen Sie als Leser selbst entscheiden.

Ein Buch soll aber nicht nur vom Inhalt her gut sein, sondern Form und Inhalt sollten übereinstimmen. Daher wurde auch versucht, die Form anspruchsvoll zu gestalten. Da ich ein Buch als »Gesamtkunstwerk« betrachte, ist auf der Buchtitelseite ein Bild der Malerin Anna Solecka-Zach abgedruckt. Sie setzt den Computer als Hilfsmittel ein, um ihre künstlerischen Vorstellungen umzusetzen.

Die dynamische Entwicklung der Softwaretechnik und die Themenbreite machen es für einen Autor fast unmöglich, alleine ein Lehrbuch der Softwaretechnik zu schreiben. Ich habe daher einige Kollegen gebeten, Teilgebiete durch eigene Beiträge abzudecken.

Koautoren

Prof. Dr. rer. nat. Rainer Koschke, Universität Bremen, hat das Kapitel zum Thema »Software-Werkzeuge« verfasst:

■ »Werkzeuge«, S. 59

Prof. Dr.-Ing. Uwe Lämmel, Hochschule Wismar, hat ein Kapitel zum Thema »Regelbasierte Systeme« beigetragen:

■ »Regeln«, S. 404

Prof. Dr.-Ing. Peter Liggesmeyer, Technische Universität Kaiserslautern und Institutsleiter des Fraunhofer Instituts für Experimentelles Software Engineering, und seine Mitarbeiter Dr. Robert Eschbach, Dr. Mario Trapp, Johannes Kloos und Bastian Zimmer haben mit der »Fallstudie Fensterheber« und entsprechenden Beispielen in den Basiskonzepten sowie mit dem Kapitel »Formale Logik« zu diesem Buch beigetragen:

■ »Fallstudie: Fensterheber – Die Spezifikation«, S. 117

■ »Fallstudie: Fensterheber – Die fachliche Lösung«, S. 575

■ »Formale Logik«, S. 357

■ »Markov-Ketten«, S. 292

■ »Generalisierte stochastische Petrinetze«, S. 316

Dr.-Ing. Jochen Quante, Universität Bremen, hat das Thema »Aufwandsschätzmethoden« bearbeitet:

■ »Schätzen des Aufwands«, S. 515

Von meiner Frau, Prof. Dr. Heide Balzert, Fachhochschule Dortmund, habe ich aus ihrem »Lehrbuch der Objektmodellierung«, 2. Auflage, eine Reihe von Abschnitten übernommen [Balz05]. Die Boxen mit Checklisten stammen – in etwas abgewandelter Form – ebenfalls aus dem »Lehrbuch der Objektmodellierung«.

### Vorwort zur 3. Auflage

Dank Ich danke allen Koautoren für ihre Beiträge zu diesem Lehrbuch. Zusätzlich danke ich Dr.-Ing. Olaf Zwintzsch, Geschäftsführer der W3L GmbH in Witten und Lehrbeauftragter für Softwaretechnik an der Ruhr-Universität Bochum, für die kritische Durchsicht des gesamten Buches und Prof. Dr. Roland Gabriel, Lehrstuhl für Wirtschaftsinformatik an der Ruhr-Universität Bochum, für Anregungen zum Kapitel »Multidimensionale Datenstrukturen«.

Die Grafiken erstellte meine Mitarbeiterin Frau Anja Scharlt. Danke!

Trotz der Unterstützung vieler Personen bei der Erstellung dieses Buches enthält ein so umfangreiches Werk sicher immer noch Fehler und Verbesserungsmöglichkeiten: »*Nobody is perfect*«. Kritik und Anregungen sind daher jederzeit willkommen. Eine aktuelle Liste mit Korrekturen und Informationen zu diesem Buch finden Sie im kostenlosen E-Learning-Kurs zu diesem Buch (siehe oben).

Nach soviel Vorrede wünsche ich Ihnen nun viel Spaß beim Lesen. Möge Ihnen dieses Buch – trotz der manchmal »trockenen« Materie – ein wenig von der Faszination und Vielfalt der Softwaretechnik vermitteln. Werden Sie ein guter Softwareingenieur – Sie werden gebraucht!

Ihr



# Inhalt

## **I      Die Wissenschaftsdisziplin Softwaretechnik** 1

- 1      Was ist Software?** 3
- 2      Warum ist Software so schwer zu entwickeln?** 9
- 3      Was ist Softwaretechnik?** 17

## **II     Basistechniken** 23

- 4      Prinzipien** 25
  - 4.1    Prinzip der Abstraktion 26
  - 4.2    Prinzip der Strukturierung 34
  - 4.3    Prinzip der Bindung und Kopplung 37
  - 4.4    Prinzip der Hierarchisierung 38
  - 4.5    Prinzip der Modularisierung 40
  - 4.6    Geheimnisprinzip 42
  - 4.7    Prinzip der Lokalität 45
  - 4.8    Prinzip der Verbalisierung 46
  - 4.9    Abhängigkeiten zwischen den Prinzipien 48
  - 4.10   Zusammenfassung 50
- 5      Methoden** 53
- 6      Werkzeuge** 59
  - 6.1    Menschen, Methoden, Werkzeuge 59
  - 6.2    Klassifikation von Werkzeugen 60
    - 6.2.1   Von Werkzeugen behandelte Artefakte 62
    - 6.2.2   Von Werkzeugen unterstützte Operationen 62
    - 6.2.3   Werkzeuge zur Kollaboration und Kommunikation 73
    - 6.2.4   Unterstützung von Prozessmodellen und Methoden 75
  - 6.3    Integrierte Entwicklungsumgebungen 76
  - 6.4    Modellgetriebene Entwicklung 79
  - 6.5    Auswahlkriterien bei der Anschaffung von Werkzeugen 87
  - 6.6    Evaluationsverfahren für die Anschaffung 90
  - 6.7    Zusammenfassung 97

**III      Basiskonzepte    99**

**7           Fallstudie: SemOrg – Die Spezifikation    107**

**8           Fallstudie: Fensterheber – Die Spezifikation    117**

**9           Statik    127**

- 9.1      Funktionalität    127
- 9.1.1   Einzelne Funktionen    128
- 9.1.2   Zusammenfassung von Funktionen    131
- 9.1.3   Box: Klassen – Methode und Checkliste    137
- 9.2      Funktions-Strukturen    142
- 9.2.1   Funktionsbaum    143
- 9.2.2   Pakete    145
- 9.2.3   Box: Pakete – Methode und Checkliste    148
- 9.2.4   Vererbung    150
- 9.2.5   Box: Vererbung – Methode und Checkliste    155
- 9.2.6   Assoziation    158
- 9.2.7   Box: Assoziationen – Methode und Checkliste    166
- 9.2.8   Box: Multiplizitäten – Methode und Checkliste    169
- 9.2.9   Aggregation und Komposition    171
- 9.2.10   Box: Komposition und Aggregation – Methode und  
Checkliste    175
- 9.2.11   Weitere Strukturen    177
- 9.3      Daten    181
- 9.4      Box: Attribute – Methode und Checkliste    187
- 9.5      Daten-Strukturen    190
- 9.5.1   XML, DTD und XML-Schemata    190
- 9.5.2   Entity-Relationship-Modell    199
- 9.5.2.1   ER-Konzepte und OO-Konzepte im Vergleich    200
- 9.5.2.2   Schlüssel, Tabellen und Dateien    204
- 9.5.2.3   Beispiele für semantische Datenmodelle    207
- 9.5.2.4   Unternehmensdatenmodelle und Weltmodelle    209
- 9.5.2.5   Zusammenfassung    213
- 9.5.3   Multidimensionale Datenstrukturen    214
- 9.5.3.1   *Data Warehouse* und *Data Marts*    215
- 9.5.3.2   OLAP und Hyperwürfel    217
- 9.5.3.3   Modellierungsansätze    222
- 9.5.3.4   Zusammenfassung    226

**10          Dynamik    227**

- 10.1      Kontrollstrukturen    227
- 10.1.1   Die Sequenz    229
- 10.1.2   Die Auswahl    229
- 10.1.3   Die Wiederholung    231
- 10.1.4   Der Aufruf    235