

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, Implementierung, Installation und Betrieb« (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 der Entwurf, die Implementierung, die Installation und der Betrieb einer Anwendungssoftware behandelt. Dieses Buch besteht aus folgenden Teilen:

I Der Entwurf

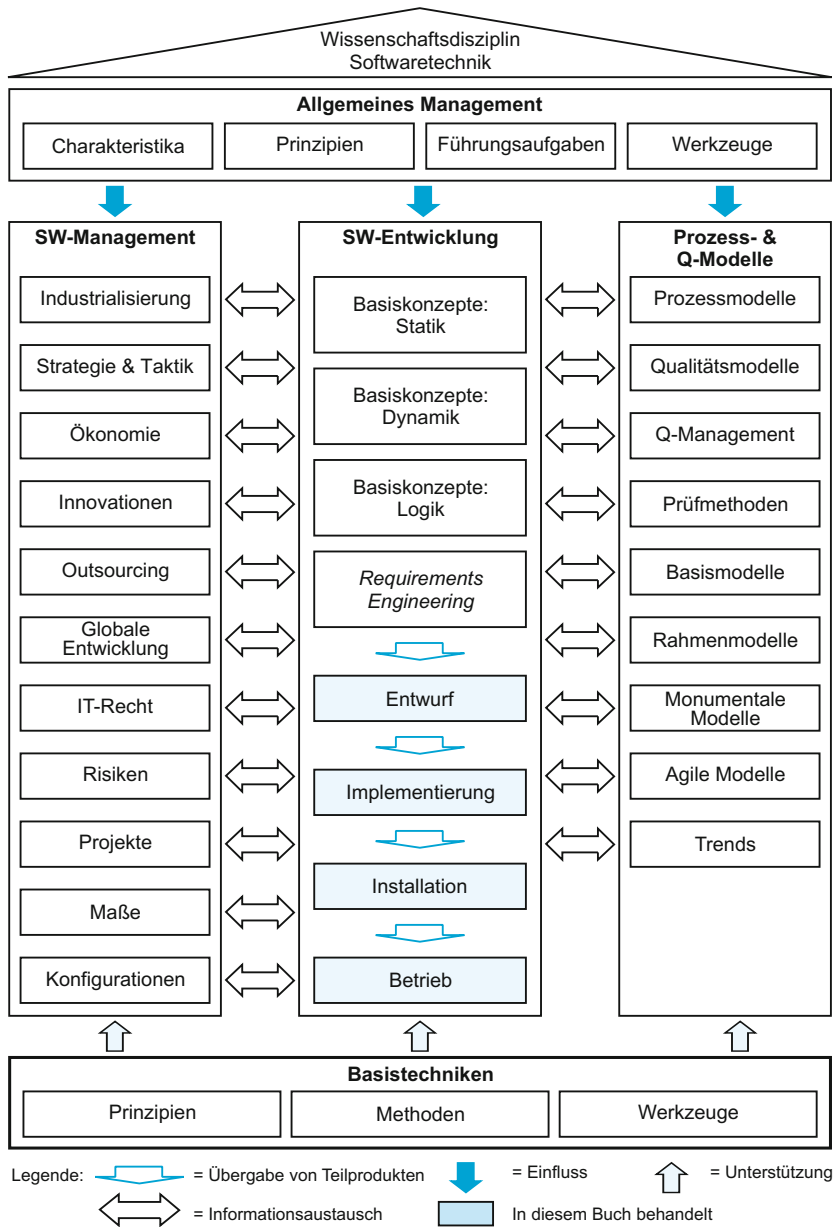
II Die Implementierung

III Verteilung, Installation, Abnahme und Einführung

IV Der Betrieb

Vorwort zur 3. Auflage

Abb. 0.0-1:
Gliederung der
Softwaretechnik.



Eine Softwareentwicklung darf nicht isoliert gesehen werden, sondern es muss immer der gesamte Software-Lebenszyklus betrachtet werden. Daher beginnt dieses Buch mit einer Betrachtung des Lebenszyklus:

■ »Der Software-Lebenszyklus«, S. 1

Vorwort zur 3. Auflage

Im Mittelpunkt dieses Buches steht der Entwurf von Softwarearchitekturen. Dabei wird eine ganzheitliche Sicht eingenommen. Die Kunst besteht darin, eine Softwarearchitektur zu entwerfen, die die funktionalen und nichtfunktionalen Anforderungen unter Berücksichtigung von Architekturprinzipien, Architektur- und Entwurfsmustern sowie weiteren Einflussfaktoren erfüllt. Dabei sind vielfältige Abhängigkeiten zu berücksichtigen. Nicht umsonst wird daher der Entwurf auch als »Königsdisziplin« der Softwaretechnik bezeichnet.

Entwurf im
Mittelpunkt

Erfahrungen haben gezeigt, dass viele Studierende der Informatik zwar in der Lage sind, »Entwürfe im Kleinen« vorzunehmen, aber oft überfordert sind, eine globale Softwarearchitektur zu entwerfen. Betrachtet man heutige Lehrbücher zur Softwarearchitektur, dann stellt man zwei Extreme fest:

- Es gibt Lehrbücher, die die Softwarearchitektur sehr abstrakt und auf einer Modellebene betrachten und beschreiben.
- Auf der anderen Seite gibt es Lehrbücher, die sehr konkret spezifische plattformorientierte Architekturen vorstellen und auch die zugehörige Programmierung dazu zeigen, zum Beispiel Lehrbücher zu Java EE (*Enterprise Edition*) oder .Net.

2 Extreme

Beide Wege sind *nicht* zufriedenstellend. Bei der abstrakten Behandlung von Softwarearchitekturen fehlt die eigene praktische Erfahrung für die Umsetzung der Konzepte. Bei der Behandlung einer plattformspezifischen Architektur nimmt die Plattformarchitektur dem Entwerfer viel Arbeit ab, ohne dass er sich über die einzelnen Aspekte jeweils im Klaren ist. Außerdem werden Architekturalternativen meist nicht behandelt.

In diesem Buch wird daher versucht, einen Mittelweg zwischen beiden Extremen einzuschlagen. Ausgehend von globalen Architekturmustern werden zunächst Einzelaspekte mit ihren Alternativen behandelt. Damit immer der Bezug zur Realität vorhanden ist, wird eine durchgängige Fallstudie in verschiedenen Varianten zunächst für Einzelaspekte entworfen und implementiert. Dadurch wird es auch möglich, gute Softwarearchitekturen zu entwerfen, auch wenn keine Standardplattform, wie z. B. Java EE, zur Verfügung steht, nicht geeignet ist oder nicht benötigt wird:

Mittelweg

- »Der Entwurf«, S. 5

Natürlich ist es in einem Lehrbuch nicht möglich, alle Aspekte einer Softwarearchitektur im Detail zu behandeln. Zur Implementierung der Fallstudie wird daher immer die Programmiersprache Java einschließlich verfügbaren Java-Frameworks verwendet. Neben der Java EE-Plattform wird aber auch die .Net-Plattform behandelt. Auf architektonische Unterschiede wird eingegangen. Zusätzlich werden die Besonderheiten bei softwareintensiven Systemen behandelt.

Vorwort zur 3. Auflage

Programme zum
Herunterladen

Im kostenlosen E-Learning-Kurs zu diesem Buch finden Sie alle Programme zum Herunterladen sowie Beschreibungen zu den eingesetzten Entwicklungswerkzeugen und Plattformen.

Auf die Programmierung bzw. Implementierung von entworfenen Softwarekomponenten wird nur ein kurzer Blick geworfen, da die meisten Studierenden der Informatik sich mit diesem Teil der Softwareentwicklung meist intensiv auseinandergesetzt haben:

■ »Die Implementierung«, S. 491

Die Verteilung und Installation – im Englischen *Deployment* genannt – von Komponenten eines Softwaresystems auf Client- und Serversysteme spielt heute eine wichtige Rolle. Die Anforderungen an die Effizienz bei der Verteilung und Installation haben Rückwirkungen auf die Softwarearchitektur:

■ »Verteilung, Installation, Abnahme und Einführung«, S. 519

Früher ging man davon aus, dass Softwaresysteme nur eine begrenzte Lebensdauer haben. Man war der Meinung, wenn ein Softwaresystem den Anforderungen nicht mehr genügt, dass man es dann durch ein neues Softwaresystem abgelöst.

Die Erfahrung hat gezeigt, dass der Aufwand, um Altsysteme durch vollständig neue Systeme zu ersetzen, oft so groß ist, dass ein Austausch nicht rentabel ist. Daher muss man – wie auch bei Städten mit alten und neuen Stadtteilen – in der Lage sein, Altsysteme in neue Umgebungen einzubetten. Eine Konsequenz aus dieser Erkenntnis ist außerdem, dass die Weiterentwickelbarkeit von neuen Systemen heute eine wesentlich größere Priorität besitzt.

Daher spielt der Betrieb von Softwaresystemen einschließlich der Wartung, Pflege und Weiterentwicklung heute eine große Rolle, mit Rückwirkungen auf die Softwarearchitektur:

■ »Der Betrieb«, S. 529

Zur 3. Auflage

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.

Viele Themen, die in der 2. Auflage von 2001 behandelt wurden, können heute als bekannt vorausgesetzt werden, z.B. Einführung in relationale Datenbanksysteme und SQL. Objektorientierte Datenbanksysteme haben sich nicht durchgesetzt und werden daher in diesem Buch *nicht* mehr behandelt. Ebenso werden die heute veralteten Methoden SD (*Structured Design*) und MD (*Modular Design*) *nicht* mehr betrachtet.

An diesen Beispielen sieht man auch die dynamische Entwicklung der Softwaretechnik und insbesondere der Softwarearchitekturen. Sehr viel mehr Wert wird daher in diesem Buch auf Architektur- und Entwurfsmuster gelegt.

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.

Das Buch besteht aus 33 Kapiteln. Die meisten Kapitel enthalten Unterkapitel. Am Ende des Buches finden Sie ein umfangreiches Glossar, das Literaturverzeichnis und das Sachregister. Dadurch kann das Buch auch ideal als Nachschlagewerk benutzt werden.

Buchaufbau

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).

Didaktik &
Methodik

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 Fragen an Sie. Diese Fragen sollen 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).

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.

Querverweise

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, der zusätzlich zahlreiche Animationen und Selbsttestaufgaben mit Musterlösungen enthält. 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 und dem Berufsprofil »Software-Architekt/in (FH Dortmund)« im Rahmen der »Wissenschaftlichen Informatik-Weiterbildung Online«. Sie finden Informationen dazu und den E-Learning-Kurs auf der W3L-Plattform (<http://www.W3L.de>).

Einsatz des
Buches

Vorwort zur 3. Auflage

Kostenloser E-Learning-Kurs Ergänzend zu diesem Buch gibt es den kostenlosen E-Learning-Kurs »Nichtfunktionale Anforderungen«, der Installationsanleitungen und zusätzlich zahlreiche Tests enthält, mit denen Sie Ihr Wissen überprüfen können. Sie finden den Kurs auf der Website <http://Akademie.W3L.de>. Unter Startseite & Aktuelles finden Sie in der Box E-Learning-Kurs zum Buch den Link zum Registrieren. Nach der Registrierung und dem Einloggen geben Sie bitte die folgende Transaktionsnummer (TAN) ein: 5403882130.

Zielgruppen Dieses Buch ist für folgende Zielgruppen geschrieben:

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

Vorkenntnisse Vorausgesetzt werden gute Kenntnisse der Programmiersprache Java, der Modellierungssprache UML, der objektorientierten Analyse sowie Erfahrungen bei der Entwicklung von Softwaresystemen.

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

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

Hinweis Viele der komplexen UML-Diagramme wurden mit einem UML-Werkzeug erstellt. Da dieses Werkzeug keine Vektorgrafiken erzeugt, ist die Darstellungsqualität dieser Grafiken im Buch nicht optimal. Um Übertragungsfehler zu vermeiden, wurde jedoch auf ein Neuzeichnen verzichtet.

Begriffe, Glossar halbfett, blau 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.

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 – insbesondere wenn ergänzend dazu der E-Learning-Kurs eingesetzt wird.

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

Vorwort zur 3. Auflage

Fachbegriffe. Auf der anderen Seite gibt es jedoch für viele Bereiche der Softwaretechnik sowohl übliche als auch sinnvolle deutsche Bezeichnungen, z. B. Entwurf für *Design*.

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 *nicht* 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.

Gibt es noch keinen eingebürgerten deutschen Begriff, dann wird der englische Originalbegriff verwendet. Englische Bezeichnungen sind immer *kursiv* gesetzt, sodass sie sofort ins Auge fallen.

Englische Begriffe
kursiv gesetzt

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.

Lesen des Buches:
sequenziell

Eine andere Möglichkeit besteht darin, jeweils eine der Teildisziplinen »Entwurf«, »Implementierung«, »Verteilung, Installation, Abnahme und Einführung« und »Betrieb« durchzuarbeiten. Auf Querbezüge und notwendige Voraussetzungen wird jeweils hingewiesen.

Nach
Teildisziplinen

Außerdem kann das Buch themenbezogen gelesen werden. Möchte man sich in die Architektur- und Entwurfsmuster einarbeiten, dann kann man die dafür relevanten Kapitel durcharbeiten.

Themenbezogen

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 Globalisierung.

Punktuell

Durch ein ausführliches Sachregister und Glossar kann dieses Buch auch gut zum Nachschlagen verwendet werden.

Zum
Nachschlagen

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.

Vorwort zur 3. Auflage

Koautoren 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 zwei Kollegen gebeten, Teilgebiete durch eigene Beiträge abzudecken.

Prof. Dr.-Ing. Peter Liggesmeyer, Technische Universität Kaiserslautern und Institutsleiter des Fraunhofer Instituts für Experimentelles Software Engineering, und seine Mitarbeiter Dr. Mario Trapp, Dr. Martin Becker und Thorsten Keuler haben das Buch durch Beiträge zu eingebetteten Systemen ergänzt:

- »Betriebssicherheit und Funktionssicherheit«, S. 121 (Ergänzung des Abschnitts zur Funktionssicherheit)
- »Zuverlässigkeit«, S. 124 (Ergänzung)
- »Architekturen »Eingebetteter Systeme««, S. 397

Dr. Holger Schwichtenberg, Technologieberater und Softwarearchitekt, hat das Buch um Microsoft-spezifische Gesichtspunkte bereichert:

- »Transaktionen in .NET«, S. 186
- »Netzkommunikation in .NET«, S. 301
- »Die .NET-Plattform«, S. 360 (Kapitel und Unterkapitel)
- »Persistenz in .NET«, S. 442
- »GUIs in der .NET-Plattform«, S. 467

Dank Ich danke den Koautoren für ihre Beiträge zu diesem Lehrbuch. Mein besonderer Dank gilt meinem wissenschaftlichen Mitarbeiter M.Sc. Michael Goll, der viele Beispiele in diesem Buch implementiert hat. Mein wissenschaftlicher Mitarbeiter M.Sc. Jakob Pyttlik hat die Fallstudie zum Subsystem Benutzungsoberfläche (siehe »Fallstudie: Kundenverwaltung – GUI«, S. 458) konzipiert und programmiert. Danke dafür.

Um nichts zu übersehen, habe ich zwei Fachleute gebeten, den Inhalt dieses Buches kritisch zu reflektieren und mir eine entsprechende Rückmeldung zu geben. Ich möchte daher folgenden Personen herzlich danken:

- Dr.-Ing. Olaf Zwintzsch, Geschäftsführer der W3L GmbH in Witten und Lehrbeauftragter für Softwaretechnik an der Ruhr-Universität Bochum.
- Dr.-Ing. Doga Arinir, Leiter der Softwareentwicklung der W3L GmbH in Witten und Lehrbeauftragter für Softwaretechnik an der Ruhr-Universität Bochum.

Die Grafiken erstellte meine Mitarbeiterin Frau Anja Scharthl. 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.

Vorwort zur 3. Auflage

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

A handwritten signature in black ink, reading "Helmut Beier". The signature is fluid and cursive, with a long horizontal stroke at the end.

Lehrbuch der Softwaretechnik: Entwurf,
Implementierung, Installation und Betrieb

Balzert, H.

2011, XVIII, 596 S. 320 Abb. in Farbe., Hardcover

ISBN: 978-3-8274-1706-0