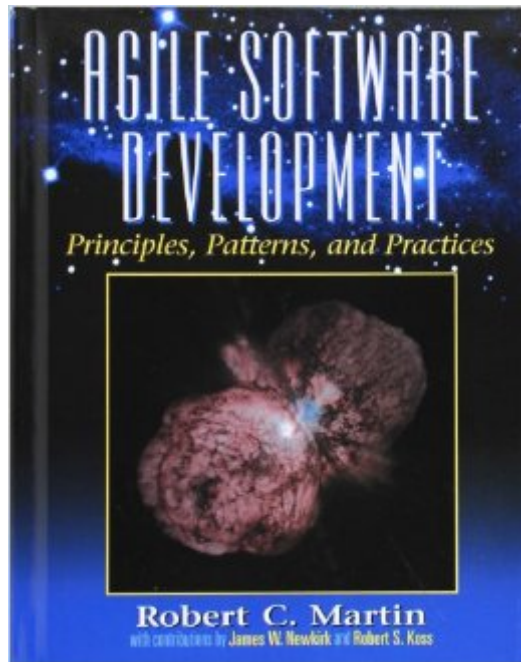


The book was found

Agile Software Development, Principles, Patterns, And Practices



Synopsis

Written by a software developer for software developers, this book is a unique collection of the latest software development methods. The author includes OOD, UML, Design Patterns, Agile and XP methods with a detailed description of a complete software design for reusable programs in C++ and Java. Using a practical, problem-solving approach, it shows how to develop an object-oriented applicationâfrom the early stages of analysis, through the low-level design and into the implementation. Walks readers through the designer's thoughts â showing the errors, blind alleys, and creative insights that occur throughout the software design process. The book covers: Statics and Dynamics; Principles of Class Design; Complexity Management; Principles of Package Design; Analysis and Design; Patterns and Paradigm Crossings. Explains the principles of OOD, one by one, and then demonstrates them with numerous examples, completely worked-through designs, and case studies. Covers traps, pitfalls, and work arounds in the application of C++ and OOD and then shows how Agile methods can be used. Discusses the methods for designing and developing big software in detail. Features a three-chapter, in-depth, single case study of a building security system. For Software Engineers, Programmers, and Analysts who want to understand how to design object oriented software with state of the art methods.

Book Information

Hardcover: 529 pages

Publisher: Pearson; 1st edition (October 25, 2002)

Language: English

ISBN-10: 0135974445

ISBN-13: 978-0135974445

Product Dimensions: 8 x 1.9 x 10 inches

Shipping Weight: 2.6 pounds (View shipping rates and policies)

Average Customer Review: 4.6 out of 5 starsÂ See all reviewsÂ (54 customer reviews)

Best Sellers Rank: #126,084 in Books (See Top 100 in Books) #51 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design #73 inÂ Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming #89 inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C++

Customer Reviews

I don't think I've given another design/programming book 5 stars before. This book deserves it-- it could easily replace a half dozen books on my shelf, and it probably will. Martin focuses on the why's

and the wherefores of current OOD methodologies. He doesn't try to sell Agile Processes in this book. Instead, he explains a number of current practices that might be loosely grouped under the 'Agile' name. He anchors his discussion in a set of principles that drive the design process. Then he shows how software patterns can be used to put these principles into practice. Patterns are explained and demonstrated in the context of three case studies. The case studies (a payroll system, a weather monitoring system, and an exam testing system) have the feel of day-to-day problems. One of my chief complaints with other books has been the use of esoteric case studies-- unless I work for Microsoft, I'm not likely to write a word processor anytime soon. Okay, so maybe I won't write a weather station either, but it comes a lot closer to what I will do! The patterns discussion in this book is down-to-earth and easily understood. I have struggled over the 'Gang of Four' book ('Gamma et Al, 'Design Patterns') for well over a year. Bob Martin's book has cut through a lot of the clutter and confusion. It has been a great help to me in understanding why, where, and when to use different patterns. And the explanation of UML in the book's appendices is one of the best I have seen. I can't think of a better way to learn UML than to sit down with these Appendices and Martin Fowler's 'UML Distilled'. This is one of the two books I would recommend to an OOD newbie. The other would be 'Object Design' by Rebecca Wirfs Brock and Alan McKean. These books provide a solid grounding in object-oriented design, while requiring a very reasonable expenditure of time and effort.

The bulk of this book describes OO design principles. They're presented in a readable, useful, and well-organized way. Often they just clarify and put a name to something you've probably been doing anyway. The standard Dependency Inversion Principle is there, for one. (I'm glad to see that other people have trouble with the name. By today's reckoning, there's nothing inverted about it, but the name dates back to less enlightened times.) Others, like the Interface Segregation Principle, are less well known but reinforce lots of other good practices, such as data hiding and prevention of "interface leakage". The "Agile" section is blessedly short, and doesn't much contaminate the otherwise good presentation elsewhere in the book. There's a lot of good to be extracted from the agility movement, but there's a lot of rabid dogmatism too. Martin managed to keep it well under control. He presented the Manifesto (ugh) early on, but that was the worst of it. A few points marred the book, but only slightly. The drawings came across as "cute" - unprofessional and tangential to the topics at hand. Semi-fictional conversations in books like this always seem fatuous to me, and Ch.6 was no exception. The technical content managed to withstand this presentation anyway. This book has lots of good ideas. It relates those ideas well to common and useful design patterns. A few

aspects of the book tried to be funny, but came across as more annoying than anything else. That was only a few, though - the meaningful content of the book came through despite those flaws. I recommend this book to any serious student or practitioner of OO design and implementation. I really mean "any," since even project-scarred veterans are likely to see some of their hard won knowledge set into clear text and into the context of other ideas.

I knew the book would be a great one before read it. But now, after I read some of its chapters, I know I underestimated it. I love to read Uncle Bob's books and articles. His previous book "Designing Object-Oriented C++ Application with Booch Method" is a real gem, I learn much a lot from it, maybe more than any other books on designing. The author's style is unique, he tries to guide readers to reach a good design instead of just putting the final solution in front of you. He presents the whole process of design, shows you how to think, how to verify, how to test and modify. His is a real mentor who gives you solid knowledge and solid experience by solid examples. So, I expected learn a lot from this new book. The book shows that it's more than my expectation. It keeps the good style, plus very valuable contents. It present at least 4 aspects which are very important and useful for today's programmers: * Agile method: The author show you you how to USE agile method. Still he tell you a lot about "Why". I'm not a XPer, but after reading the first several chapters, I think I'd give a try. * Object-Oriented Design Principles: The book concludes 11 O-O design principles. Only these principles are worth the price of the book to me. * Design patterns: This book teach you 23 design patterns with concrete examples -- 15 are GoF patterns, 8 are new. The emphasis is how to use patterns in real applications, instead of telling you what design patterns are and how to document them. * UML: This book is not about UML, but it uses UML to demonstrate designs. To make you feet wet, it includes two appendics, show you basic UML with, again, concrete example. I find it's much easier for me to learn UML this way. Well, IMHO, this book is the best O-O design textbook this year, and I wonder whether there will be a better one in the next several years.

[Download to continue reading...](#)

Agile Software Development, Principles, Patterns, and Practices Agile in a Flash: Speed-Learning Agile Software Development (Pragmatic Programmers) Crochet: Easy Crochet Patterns: Crochet Patterns for Beginners (Crochet: Step by Step Crochet, Crochet Patterns, Easy Crochet Patterns, Crochet Patterns for Beginners, and Crochet Projects) Agile Principles, Patterns, and Practices in C# Small Memory Software: Patterns for systems with limited memory (Software Patterns Series) Exploring Mac App Development Strategies: Patterns & Best Practices for Clean Software

Architecture on the Mac with Swift 2.0 and Tests Engineering Software as a Service: An Agile Approach Using Cloud Computing Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio (Portuguese Edition) Engineering Software as a Service: An Agile Approach Using Cloud Computing + \$10 AWS Credit Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process Parallel Programming with Microsoft Visual C++: Design Patterns for Decomposition and Coordination on Multicore Architectures (Patterns & Practices) 11 Crochet Shawl Patterns: Crochet Poncho Patterns, Free Easy Crochet Patterns and More Crochet Mandala For Beginners Learn To Create 15 Amazing Crochet Mandala Patterns: (Crochet Mandala Patterns, Crochet for Beginners) (crochet books patterns, cute and easy crochet) His Forge Burns Hot for Mosaic Damascus: Knife Patterns & Techniques: Damascus pattern making & techniques. Learn how to make mosaic Damascus patterns ... techniques for making Damascus patterns. Crochet Mandala: 12 Most Gorgeous Patterns With Easy Instructions: (Crochet Hook A, Crochet Accessories, Crochet Patterns, Crochet Books, Easy Crochet ... Crocheting For Dummies, Crochet Patterns) Crochet Mandala: 15 Best Fabulous Patterns With Easy Instructions: (Crochet Hook A, Crochet Accessories, Crochet Patterns, Crochet Books, Easy Crochet ... Crocheting For Dummies, Crochet Patterns) Agile Web Development with Rails 5 Algorithms: C++: Data Structures, Automation & Problem Solving, w/ Programming & Design (app design, app development, web development, web design, jquery, ... software engineering, r programming) Theories and Practices of Development (Routledge Perspectives on Development) Theories and Practices of Development: Volume 8 (Routledge Perspectives on Development)

[Dmca](#)