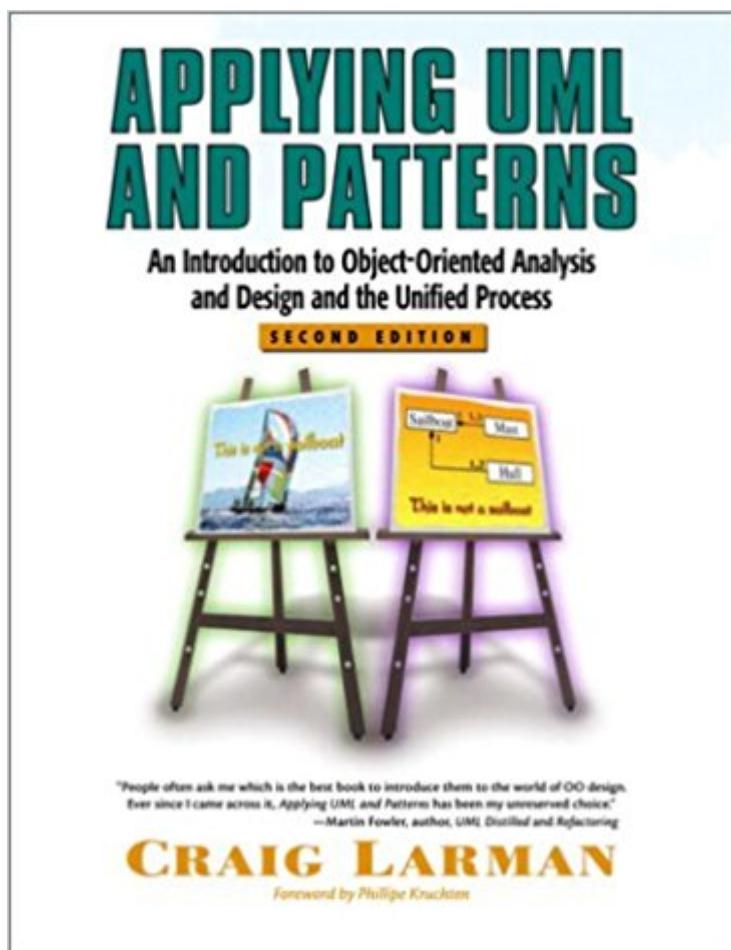


The book was found

Applying UML And Patterns: An Introduction To Object-Oriented Analysis And Design And The Unified Process (2nd Edition)



Synopsis

People often ask me which is the best book to introduce them to the world of OO design. Ever since I came across it, *Applying UML and Patterns* has been my unreserved choice. Martin Fowler, author, *UML Distilled* and *Refactoring*; The first edition of *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* quickly emerged as the leading OOA/D introduction; translated to many languages and adopted in universities and businesses worldwide. In this second edition, well-known object technology and iterative methods leader Craig Larman refines and expands this text for developers and students new to OOA/D, the UML, patterns, use cases, iterative development, and related topics. The book helps newcomers to OOA/D learn how to *think in objects*; by presenting three iterations of a single, cohesive case study, incrementally introducing the requirements and OOA/D activities, principles, and patterns that are most critical to success. It introduces the most frequently used UML diagramming notation, while emphasizing that OOA/D is much more than knowing UML notation. All case study iterations and skills are presented in the context of an *agile*; version of the Unified Process -- a popular, modern iterative approach to software development. Throughout, Larman presents the topics in a fashion designed for learning and comprehension. Among the topics introduced in *Applying UML and Patterns* are: * requirements and use cases, * domain object modeling, * core UML, * designing objects with responsibilities, * *Gang of Four*; and other design patterns, * mapping designs to code (using Java as an example), * layered architectures, * architectural analysis, * package design, * iterative development, * the Unified Process. Foreword by Philippe Kruchten, the lead architect of the Rational Unified Process. *Too few people have a knack for explaining things. Fewer still have a handle on software analysis and design. Craig Larman has both.*; John Vlissides, author, *Design Patterns*; and *Pattern Hatching*; *This edition contains Larman's usual accurate and thoughtful writing. It is a very good book made even better.*; Alistair Cockburn, author, *Writing Effective Use Cases*; and *Surviving OO Projects*;

Book Information

Hardcover: 656 pages

Publisher: Prentice Hall PTR; 2 edition (July 13, 2001)

Language: English

ISBN-10: 0130925691

ISBN-13: 978-0130925695

Product Dimensions: 8.3 x 1.2 x 10.4 inches

Shipping Weight: 3.2 pounds

Average Customer Review: 4.4 out of 5 stars [See all reviews](#) (159 customer reviews)

Best Sellers Rank: #711,787 in Books (See Top 100 in Books) #54 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > UML #256 in Books > Textbooks > Computer Science > Object-Oriented Software Design #926 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

Craig Larman did a splendid job simplifying UML. I bought this book along with the official UML user guide and I prefer reading this book because of how it made UML look so easy. What's so impressive about this book is that the author stays with one example throughout the book, and clearly illustrates the relationships and dependencies amongst different UML diagrams and artifacts. He even shows you how to map the diagrams to actual Java code! Larman's explanations of the different design patterns is also invaluable. I haven't heard of design patterns before, and this book helped me have a deeper appreciation of patterns to create better object-oriented software.

This book is the best introduction I have seen to object-oriented analysis and design. It explains the 'Unified Process' of analysis and design, including UML and the use of patterns. I am a big fan of this book because it emphasizes the "why"--the reason for each element of the process from a business perspective. For example, instead of simply offering a recipe for writing use cases, the book talks about why one might include or exclude specific elements to achieve specific objectives. Those (like me) who have avoided Unified Process because it looks too rigid, bureaucratic, and form-laden will love this book. The author argues convincingly that UP is best implemented in an iterative process that looks more like Extreme Programming than the cumbersome waterfall process one typically associates with UP. Then the book shows how to implement a UP process in enough detail, with enough examples, and with enough flexibility, that a project leader can readily put this form of UP into practice. I have avoided UP for years, but this book may have made a convert out of me. It's the first book I have seen that makes an effective business case for UP and presents a practical guide to its implementation. I have no hesitation recommending it to anyone, from novice project leaders to seasoned software architects.

This book IS what the subtitle says: An Intro to OO Analysis and Design and the Unified Process. It is a GREAT introduction. It also uses patterns and does a good job of explaining why various approaches are what *good* programmers do. But it does strike me that the main title was invented to get the most "hits" on a variety of searches one might try. It's not primarily about "applying UML" which is a good thing, despite the title. It does use UML as its notation whenever notation is needed and it explains the usage well. I am using this book as a text in the first of a 3-course graduate series on software engineering. The reactions from students (all with industry experience, most without OO experience) have been quite positive. The use of an example that runs throughout the book provides a vehicle for getting deeper and deeper into certain topics. Larman writes in a very readable style but he doesn't write "down" to the reader. His motivations for various techniques/approaches are reasoned and appropriate. He references excellent books as well. As the professor, I have used some of these to develop certain topics more deeply in my accompanying lectures.

Craig Larman provides the reader with simple, easy-to-follow examples of UML and pattern application, with few deep theoretical or philosophical discussions of development process. This style offers the new user of UML a good guide to applied modeling concepts, without bogging he/she down in academic arguments. If you are going to buy a single book to learn UML, I highly recommend this one, as it goes beyond simply describing the notation, and illustrates its use through simple guidelines. Other books are necessary to fully understand the complexities of UML, patterns, and design process, but this one fills a huge gap for the beginner market.

I feel ripped off. I was excited to see that this well-rated book had appeared in course form, and bought it to help my development group get started with UML and thinking in terms of design patterns. The book is interesting, and the tapes are not bad, but there are only about 3 hours of tape. This means that the examples are all very simple and there is only time to lightly touch on each topic. Personally, I don't consider 3 hours and 25 minutes -- minus the time spent on administrative matters -- to be anything close to a "complete video course." I'm a big fan of the idea of video courses, but in this case I feel like I paid 3 to 4 times as much as the price of the book just to get a few hours of introductory video that aren't enough to help us out. Come on, Prentice Hall -- you can do better than that! You guys know how to provide good value -- do it! Until then, my recommendation is to buy the book and spend the rest of the money on something else.

Despite the main title, this book is not just about the UML (UML is not OOA/D or a method, it is SIMPLY NOTATION). The book helps a developer learn core skills in the art and science of building object systems, rather than notation. In fact, it is not so helpful to learn syntactically correct UML diagramming, but then not be able to create an excellent design, or evaluate, or improve an existing one. This is the harder and more valuable skill. "How should responsibilities be allocated to classes of objects? How should objects interact? What classes should do what?" These are some critical questions in the design of a system. Knowing an object-oriented language is a necessary but insufficient first step to create object systems. This book helps a developer: Apply principles and patterns to create better object designs; Follow a set of common activities in analysis and design; Create frequently used diagrams in the UML notation.

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

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition) UML and the Unified Process: practical object-oriented analysis and design Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Object-Oriented Analysis and Design with the Unified Process (Available Titles CengageNOW) Object Success : A Manager's Guide to Object-Oriented Technology And Its Impact On the Corporation (Object-Oriented Series) Object-Oriented Software Engineering: Using UML, Patterns and Java (2nd Edition) Object-Oriented Analysis and Design for Information Systems: Modeling with UML, OCL, and IFML Systems Analysis and Design: An Object-Oriented Approach with UML Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach Object-Oriented Modeling and Design with UML (2nd Edition) Reusable Software : The Base Object-Oriented Component Libraries (Prentice Hall Object-Oriented Series) Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition) [Economy Edition] Fundamentals of Object-Oriented Design in UML The Object-Oriented Approach: Concepts, Systems Development, and Modeling with UML, Second Edition Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition, Volume 1 The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) Object-Oriented Software Engineering: Practical Software Development Using UML and Java Object-Oriented Technology: From Diagram to Code with Visual Paradigm for UML Visual Object-Oriented Programming Using Delphi With CD-ROM (SIGS: Advances in Object Technology) Design Patterns CD: Elements of Reusable Object-Oriented Software (Professional Computing)

[Dmca](#)