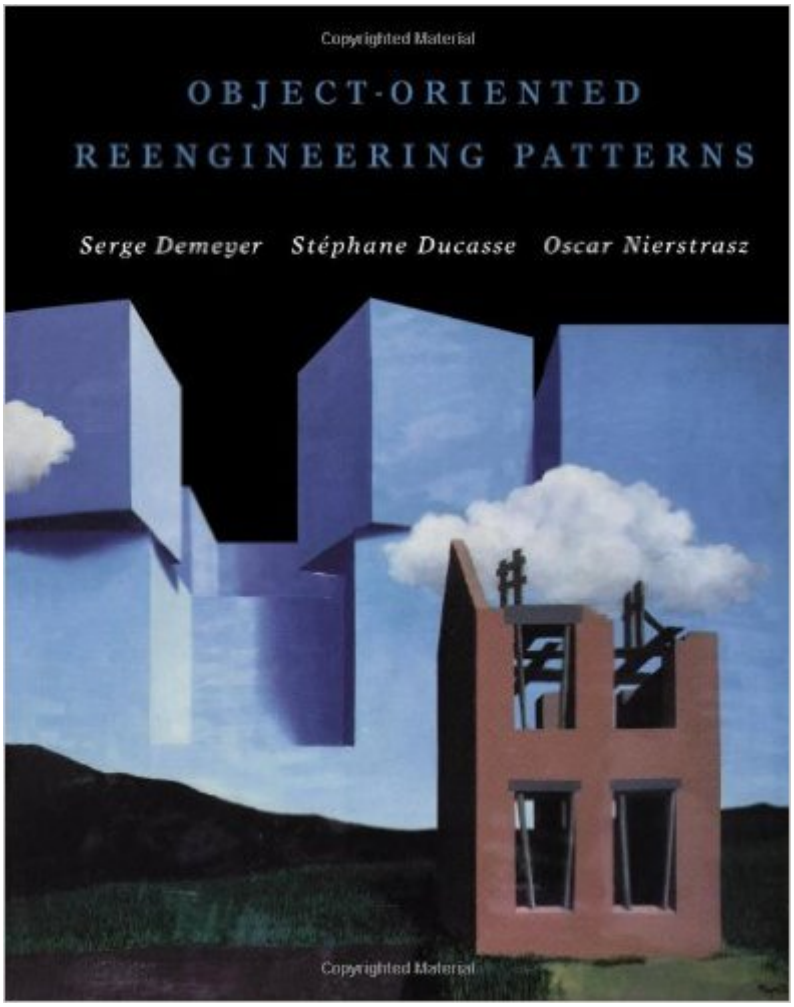
The book was found

Object-Oriented Reengineering Patterns (The Morgan Kaufmann Series In Software Engineering And Programming)



PDF
Adobe
DOWNLOAD EBOOK

Synopsis

The documentation is missing or obsolete, and the original developers have departed. Your team has limited understanding of the system, and unit tests are missing for many, if not all, of the components. When you fix a bug in one place, another bug pops up somewhere else in the system. Long rebuild times make any change difficult. All of these are signs of software that is close to the breaking point.Many systems can be upgraded or simply thrown away if they no longer serve their purpose. Legacy software, however, is crucial for operations and needs to be continually available and upgraded. How can you reduce the complexity of a legacy system sufficiently so that it can continue to be used and adapted at acceptable cost?Based on the authors' industrial experiences, this book is a guide on how to reverse engineer legacy systems to understand their problems, and then reengineer those systems to meet new demands. Patterns are used to clarify and explain the process of understanding large code bases, hence transforming them to meet new requirements. The key insight is that the right design and organization of your system is not something that can be evident from the initial requirements alone, but rather as a consequence of understanding how these requirements evolve. * Describes how to reverse engineer a monolithic system to understand how it really works and how to identify potential problems.* Includes reengineering patterns that tackle well-known reengineering techniques often encountered in object-oriented programming, such as introducing polymorphism, factoring out common behavior, detecting duplicated code, and understanding design.* Shows how to build a culture of continuous reengineering for achieving flexible and maintainable object-oriented systems.

Book Information

Series: The Morgan Kaufmann Series in Software Engineering and Programming
Hardcover: 282 pages
Publisher: Morgan Kaufmann; 1st edition (July 17, 2002)
Language: English
ISBN-10: 1558606394
ISBN-13: 978-1558606395
Product Dimensions: 9.5 x 7.6 x 0.9 inches
Shipping Weight: 1.8 pounds
Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (2 customer reviews)
Best Sellers Rank: #3,187,812 in Books (See Top 100 in Books) #12 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Reengineering #892

inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design #3038 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

The authors wrote better than they knew. My own experience is that reverse engineering is at least 20% of a typical software engineer's typical day. It's the study that a developer puts towards understanding a system well enough to make necessary changes, rediscovering enough about the system to enable the task at hand. The authors addressed their book to specific re-engineering projects, as isolated events, but they really wrote about the everyday life of almost every programmer.As Johnson points out in the foreword, much if the information has an "everybody knows that" feel about it. I found a few new tips or new phrases, but mostly I found a clear, systematic organization of facts and techniques that are widely applied. The authors' arrangement of known techniques makes them especially valuable, much the way an arrangement of ordinary playing cards can become a valuable hand in poker.Among other things, these authors are the first to convince me that software metrics can be a net gain to a developer. My own, sad exposure to metrics has been normative, a stick wielded blindly in the name of misunderstood "quality". This book shows how to use metrics in a constructive, exploratory way. The complexity (or whatever) scores are not the output of the process, they are intermediate results to be discarded as soon as they've pointed the way to the real problems.I found only a very few points to disagree with. For example, the authors point out pros and cons of prototypes, but missed the biggest danger of a working prototype: that, no matter what caveats are given, it can be mistaken for a real system. Over-eager clients or managers driven by a false sense of efficiency may demand that the developers just add a feature or two (usually, system's entire capability) and ship it tomorrow.

Download to continue reading...

Object-Oriented Reengineering Patterns (The Morgan Kaufmann Series in Software Engineering and Programming) Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition) Object-Oriented Software Engineering Using UML, Patterns, and Java Enterprise Software Procurement: Tools and Techniques for Successful Software Procurement and Business Process Reengineering for Municipal Executives and Managers ActionScript 3.0 Design Patterns: Object Oriented Programming Techniques (Adobe Developer Library) 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) Introduction to Programming with

Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) Programming #8:C Programming Success in a Day & Android Programming In a Day! (C Programming, C++programming, C++ programming language, Android , Android Programming, Android Games) Programming #57: C++ Programming Professional Made Easy & Android Programming in a Day (C++ Programming, C++ Language, C++for beginners, C++, Programming ... Programming, Android, C, C Programming) Computer Organization and Design, Fourth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) Computer Organization and Design, Third Edition: The Hardware/Software Interface, Third Edition (The Morgan Kaufmann Series in Computer Architecture and Design) Computer Organization and Design: The Hardware Software Interface: ARM Edition (The Morgan Kaufmann Series in Computer Architecture and Design) ARM System Developer's Guide: Designing and Optimizing System Software (The Morgan Kaufmann Series in Computer Architecture and Design) Small Memory Software: Patterns for systems with limited memory (Software Patterns Series) Learning Processing, Second Edition: A Beginner's Guide to Programming Images, Animation, and Interaction (The Morgan Kaufmann Series in Computer Graphics) Real-Time Shader Programming (The Morgan Kaufmann Series in Computer Graphics) C#: Design Patterns: The Easy Way Standard Solutions for Everyday Programming Problems; Great for: Game Programming, System Administration, App Programming, ... & Database Systems (Design Patterns Series) Learn PHP 7: Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide Ada 95: The Craft of Object-Oriented Programming

Dmca