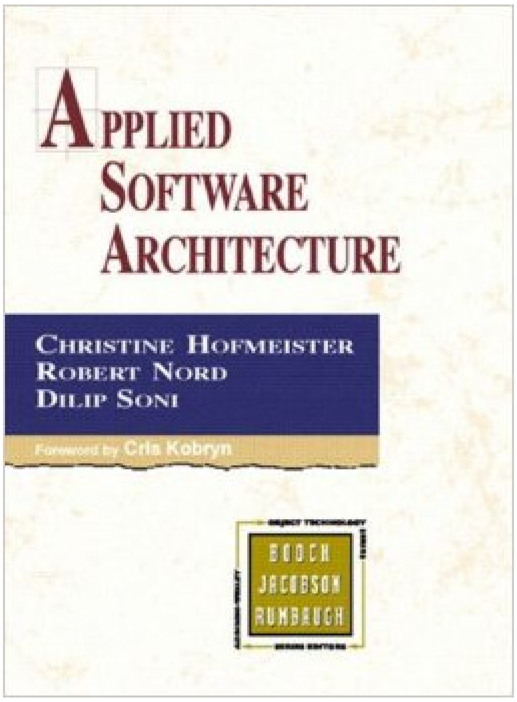
Applied Software Architecture



Synopsis

"Designing a large software system is an extremely complicated undertaking that requires juggling differing perspectives and differing goals, and evaluating differing options. Applied Software Architecture is the best book yet that gives guidance as to how to sort out and organize the conflicting pressures and produce a successful design." -- Len Bass, author of Software Architecture in Practice. Quality software architecture design has always been important, but in today’s fast-paced, rapidly changing, and complex development environment, it is essential. A solid, well-thought-out design helps to manage complexity, to resolve trade-offs among conflicting requirements, and, in general, to bring quality software to market in a more timely fashion. Applied Software Architecture provides practical guidelines and techniques for producing quality software designs. It gives an overview of software architecture basics and a detailed guide to architecture design tasks, focusing on four fundamental views of architecture--conceptual, module, execution, and code.Through four real-life case studies, this book reveals the insights and best practices of the most skilled software architects in designing software architecture. These case studies, written with the masters who created them, demonstrate how the book’s concepts and techniques are embodied in state-of-the-art architecture design. You will learn how to: *create designs flexible enough to incorporate tomorrow’s technology; *use architecture as the basis for meeting performance, modifiability, reliability, and safety requirements; *determine priorities among conflicting requirements and arrive at a successful solution; and *use software architecture to help integrate system components. Anyone involved in software architecture will find this book a valuable compendium of best practices and an insightful look at the critical role of architecture in software development. 0201325713B07092001

Book Information

Hardcover: 432 pages

Publisher: Addison-Wesley Professional; 1 edition (November 14, 1999)

Language: English

ISBN-10: 0201325713

ISBN-13: 978-0201325713

Product Dimensions: 7.7 x 0.9 x 9.7 inches

Shipping Weight: 2 pounds

Average Customer Review: 4.0 out of 5 starsÂ Â See all reviewsÂ (6 customer reviews)

Best Sellers Rank: #1,479,310 in Books (See Top 100 in Books)   #207 inÂ Books > Computers &

Technology > Hardware & DIY > Microprocessors & System Design > Computer Design #684 inÂ Books > Computers & Technology > Hardware & DIY > Design & Architecture #4045 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

OK. So you are a seasoned software engineer, 4th and 5th generation languages hold no secrets for you, design patterns are your credo, and you even have tackled this good old OMT technique. However, you still feel uneasy when it comes to translating use cases to risk management, and especially to take into account those interns who will develop part of the software.This book answers your questions by proposing both a technique and a language (UML extended), that will help you list the different factors affecting your project, infer the right design decisions, and document them throughout the project. For those with an analytical mind, the architecture process itself is decomposed and re-engineered. No consultant talk here : everything is explained, both in words and figures, using real world examples.Some will regret that the application field used for the demonstration is too narrow, since only real time applications are used, and there is no reference to database architecture or e-business ! But for those of the embedded world, such a book was awaited, and browsing (too) quickly through various application fields would have contented no one, anyhow.It is still a long reading, if you want to study all examples in depth - fortunately, you can start your own design after the first case study.Lastly, using UML throughout the project eases the communication with the development engineers, and it really helps when your team tackles detailed design.

This book is clear, solid, and workmanlike. It could work well as a textbook, or one of several texts for a term course.It gives a systematic introduction to several high-level notations, describing the conceptual, executable, structural (or module), and code views. Most of the notation is well-formed UML, and the authors take care to add semantic notes to every part of the graphical notation. They supplement the standard notations with a few text-based extensions. These capture requirements, archtiectural decisions, risks and risk mitigation, and other operating features of a living software project.One real asset is the related set of brief case studies at the end of the book, three separate products with a common conceptual base. This book is aging, it dates back to 1999 - five years, as I write this. That’s old in the "architecture" literature, and the authors fail to apply the "product line" notion. I take this book for its good, though, and lack of one buzzword is a small enough fault.The book uses a process-and-pipe model pervasively for architectural description. It’s a good tool, but other tools are good for other purposes, and their omission is a problem here.Still, the book is competent on the whole. Its sustained product-line example ties the whole together, and it focusses on practice intead of mainfestos and brand-name methodologies. There’s a lot of good here, and you can pick out out easily.//wiredweird

This book spends a great deal of time discussing what it suggests one does to architect a system. However, there is very little on how to actually do it. The steps to do things are detailed, but what doing the steps means is not well articulated. Further, some parts of the process are very poorly explained, but still used as a foundation of many other parts of the process.The best practices are simply case studies that really impart no wisdom to the reader (or, at least, to this reader).I tried to "get something" out of this book several times, and read it fully twice. However, I’m convinced that there really isn’t much there.

Download to continue reading...

Applied Software Architecture Rendering in SketchUp: From Modeling to Presentation for Architecture, Landscape Architecture, and Interior Design Computer Architecture, Fifth Edition: A Quantitative Approach (The Morgan Kaufmann Series in Computer Architecture and Design) Computer Architecture: A Quantitative Approach (The Morgan Kaufmann Series in Computer Architecture and Design) Material Strategies: Innovative Applications in Architecture (Architecture Briefs) Mansilla Y Tunon: Recent Work (2G: International Architecture Review S.) (2G: International Architecture Review Series) (Spanish and English Edition) Chinese Architecture and Metaphor: Song Culture in the Yingzao Fashi Building Manual (Spatial Habitus: Making and Meaning in Asia’s Architecture) The Architecture of Computer Hardware, Systems Software, and Networking: An Information Technology Approach 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) Software Architecture: Foundations, Theory, and Practice Inside Cisco IOS Software Architecture (CCIE Professional Development) The Architecture of Computer Hardware and System Software: An Information Technology Approach, 5th Edition The Architecture of Computer Hardware, Systems Software, & Networking: An Information Technology Approach Beyond Software Architecture: Creating and Sustaining Winning Solutions Software Architecture

And Design Illuminated (Jones and Bartlett Illuminated (Paperback)) Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives ARM System Developer’s Guide: Designing and Optimizing System Software (The Morgan Kaufmann Series in Computer Architecture and Design) Symbian OS Platform Security: Software Development Using the Symbian OS Security Architecture (Symbian Press)