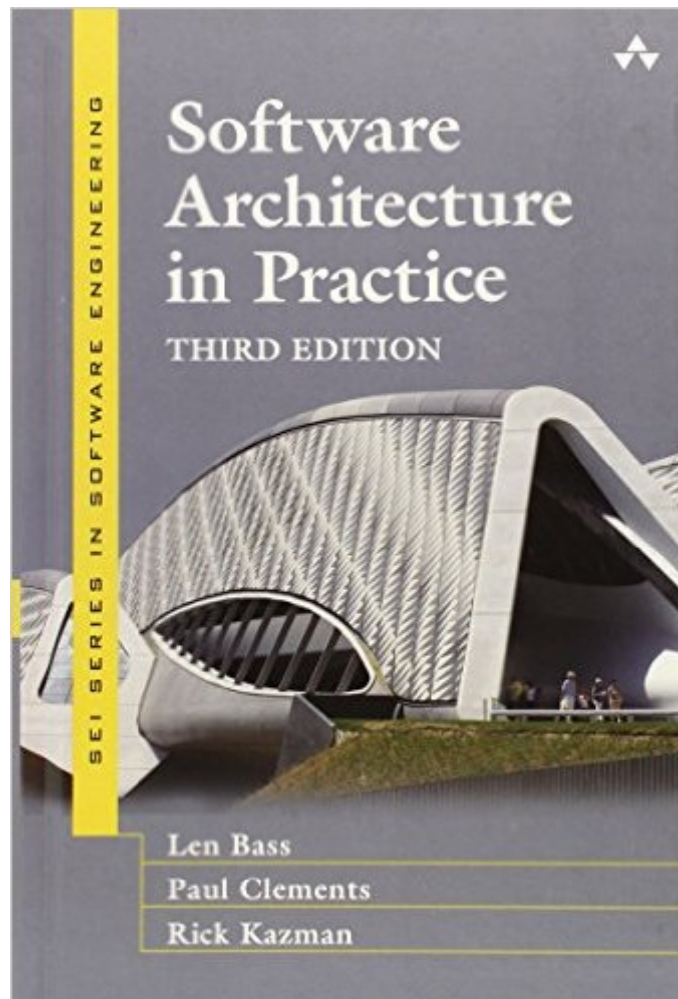The book was found

Software Architecture In Practice (3rd Edition) (SEI Series In Software Engineering)

Synopsis

The award-winning and highly influential Software Architecture in Practice, Third Edition, has been substantially revised to reflect the latest developments in the field. In a real-world setting, the book once again introduces the concepts and best practices of software architectureâ"how a software system is structured and how that systemâ™s elements are meant to interact. Distinct from the details of implementation, algorithm, and data representation, an architecture holds the key to achieving system quality, is a reusable asset that can be applied to subsequent systems, and is crucial to a software organizationâ™s business strategy. Â The authors have structured this edition around the concept of architecture influence cycles. Each cycle shows how architecture influences, and is influenced by, a particular context in which architecture plays a critical role. Contexts include technical environment, the life cycle of a project, an organizationâ™s business profile, and the architectâ™s professional practices. The authors also have greatly expanded their treatment of quality attributes, which remain central to their architecture philosophyâ"with an entire chapter devoted to each attributeâ"and broadened their treatment of architectural patterns. Â If you design, develop, or manage large software systems (or plan to do so), you will find this book to be a valuable resource for getting up to speed on the state of the art. Â Totally new material coversÂ Â Contexts of software architecture: technical, project, business, and professional Architecture competence: what this means both for individuals and organizations The origins of business goals and how this affects architecture Architecturally significant requirements, and how to determine them Architecture in the life cycle, including generate-and-test as a design philosophy; architecture conformance during implementation; architecture and testing; and architecture and agile development Architecture and current technologies, such as the cloud, social networks, and end-user devices

Book Information

Series: SEI Series in Software Engineering

Hardcover: 640 pages

Publisher: Addison-Wesley Professional; 3 edition (October 5, 2012)

Language: English

ISBN-10: 0321815734

ISBN-13: 978-0321815736

Product Dimensions: 6.5 x 1.6 x 9.4 inches

Shipping Weight: 2.2 pounds (View shipping rates and policies)

Average Customer Review: 4.6 out of 5 stars See all reviews (37 customer reviews)

Best Sellers Rank: #58,386 in Books (See Top 100 in Books)   #67 in Books > Textbooks > Computer Science > Software Design & Engineering   #144 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development   #259 in Books > Textbooks > Computer Science > Programming Languages

Customer Reviews

It was hard to imagine this book getter any better than it already was. I guess that was because I never expected so many changes. This is the most I have ever seen a new edition of a book change. If you own the second edition, you will definitely want this new one. It is almost like another book.The big case studies are gone. They have been replaced with a ton of new material. If you don't own the first and second editions of the book, you can get the case studies on the publisher's web site.Below are the chapters in this third edition.Part One. Introduction1. What Is Software Architecture?2. Why Is Software Architecture Important?3. The Many Contexts of Software ArchitecturePart Two. Quality Attributes4. Understanding Quality Attributes5. Availability6. Interoperability7. Modifiability8. Performance9. Security10. Testability11. Usability12. Other Quality Attributes13. Architectural Tactics and Patterns14. Quality Attribute Modeling and AnalysisPart Three. Architecture in the Life Cycle15. Architecture in Agile Projects16. Architecture and Requirements17. Designing an Architecture18. Documenting Software Architectures19. Architecture, Implementation, and Testing20. Architecture Reconstruction and Conformance21. Architecture Evaluation22. Management and GovernancePart Four. Architecture and Business23. Economic Analysis of Architectures24. Architecture Competence25. Architecture and Software Product LinesPart Five. The Brave New World26. Architecture in the Cloud27. Architectures for the Edge28. EpilogueTo show you just how much the book has changed I have included the table of contents to the second edition below.Pt. One. Envisioning ArchitectureCh. 1. The Architecture Business CycleCh. 2. What Is Software Architecture?Ch. 3. A-7E Avionics System: A Case Study in Utilizing Architectural StructuresPt. Two. Creating an ArchitectureCh. 4. Understanding Quality AttributesCh. 5. Achieving QualitiesCh. 6. Air Traffic Control: A Case Study in Designing for High AvailabilityCh. 7. Designing the ArchitectureCh. 8. Flight Simulation: A Case Study in an Architecture for IntegrabilityCh. 9. Documenting Software ArchitecturesCh. 10. Reconstructing Software ArchitecturesPt. Three. Analyzing ArchitecturesCh. 11. The ATAM: A Comprehensive Method for Architecture EvaluationCh. 12. The CBAM: A Quantitative Approach to Architecture Design Decision MakingCh. 13. The World Wide Web: A Case Study in InteroperabilityPt. Four.

Moving From One System to ManyCh. 14. Software Product Lines: Re-using Architectural AssetsCh. 15. CelsiusTech: A Case Study in Product Line DevelopmentCh. 16. J2EE/EJB: A Case Study of an Industry-Standard Computing InfrastructureCh. 17. The Luther Architecture: A Case Study in Mobile Applications Using J2EECh. 18. Building Systems from Off-the-Shelf ComponentsCh. 19. Software Architecture in the FutureThis book has been my go to book for most of my software architecture career. It is what taught me about quality attributes, tactics, and scenarios. It provided my first introduction to the Architecture Tradeoff Analysis Method (ATAM), the Quality Attribute Workshop (QAW), the Cost Benefit Analysis Method (CBAM), Active Reviews for Intermediate Designs (ARID), and the Attribute-Driven Design (ADD) method. In this new edition of the book they introduce the lightweight architecture evaluation. It is a slimmed-down version of ATAM. It is intended to be used on lower ceremony projects.This new version has a ton of new material. One of the biggest changes is that the quality attributes covered now have their own chapter. There is a new chapter for Availability, Interoperability, Modifiability, Performance, Security, Testability, and Usability.There is a new chapter dedicated to the technical, project, business, and professional contexts of software architecture. This chapter shows how architecture supports and is informed by other forces and activities in the different contexts.Part four, Architecture and Business, contains three chapters dedicated to showing the relationships between the architecture and the business.There is a new chapter on architecturally significant requirements and how to solicit those requirements from the goals set down by the business. This chapter introduces a new method for eliciting and documenting business goals in order to discover architecturally significant requirements. The new method is called Pedigreed Attribute eLicitation Method or PALM for short.There is a new chapter on individual and organizational architectural competence. The chapter covers the technical duties of a software architect, the non-technical duties of a software architect, and the knowledge areas of the software architect. It also provides a framework for organizational architectural competence.Part three, Architecture in the Life Cycle, shows how architecture relates to agile projects, shows how to elicit architecturally significant requirements, shows how to design the architecture and document the architecture, also how to test the architecture. Part three also covers how to reconstruct an architecture and evaluate an architecture.One of my favorite new additions was chapter 27, Architectures for the Edge. An Edge dominant system is one that depends on the input of their users for success. Edge architectures that exist today are Wikipedia, YouTube, Facebook, and Twitter. It was just a really interesting read.I could go on and on about the value of the content in this book. The bottom line is that if you are a software architect, or want to be one, this is mandatory reading, period.

Used this text book for a Master's of Software Engineering course on System Architecture. I would say the book is more of a reference than a read and learn type of book. I say this as the first group of chapters (1-12) covers Engineering Objectives and Quality Attributes. For each Quality Attribute, there are a few examples of tactics to solve them (for instance, for availability they discuss hot-spares for swapping out a failed component with an active replacement). Throughout the course, these chapters were extremely useful when building the architecture of an online collaboration system. In addition, the book discuses how tactics (and specifically which ones) cause trade offs and how to analyze them for your unique use. I now use this book at work when discussing the structure of new/existing products and it comes in handy when needing to make a tough decision. It also helps the book is an easy and understandable read.

I would give five starts for the content so far, but I decided to give only two stars because the formatting of the e-book edition is really sub-standard. Right at the start of the first chapter, the quotation is printed with one character per line over many pages. Going on, formatting switches between left-aligned and justified which is hard to read because the text looks different in each section. Some sections are indented so much that there are only a few words per line, so the text looks like a small column over pages. I wonder how this could pass quality checks by the publisher because it looks like someone converted the text and did not check the result before selling it on .All in all, I would advice every programmer or software architect to buy the book, but not the e-book.

As a software designer and engineer for over 1/4 century, I am always on the look-out for a process which claims to separate high-quality from low-quality. Software is not a tangible, 3-dimensional, manufactured product, but software has "quality". Initially someone at work started pushing this new "quality" rating methodology for software......distributed or centralized.I have dedicated the last several years to wrestling with "quality".....everywhere in my life but particularly at work with software. When I heard "Architecture Quality Attributes" I was intrigued and bought the book. Each AQA is broken down into Concerns and Common Tactics. Thinking back over the last 1/4 century....they all rang true. Quality in technical design and quality in technical management all come together here. Buy this book.My only peeve......why such a half-hearted writeup Probabilty Statistics, Fault-tree Analysis, and FMEA ? There is a dis-organized and scattered treatment through out the book. Page 84 (3rd edition) discusses AND gates, OR gates, etc etc...and then statistics shows up again from pages 255 - 259. In today's distributed environment, probability

statistics is very revealing. The more components bolted together to create a network, the more OR-gates you have that could fail and make your network inoperable. This is the Achilles heel of distributed environments.Ironically, that entire concept is hidden from the reader of this book. Even on page 256, at the very bottom, the authors give a formula for "probability of failure". In fact, the formula is EXACTLY WRONG !! The formula shown is the formula for "probability of success". (probabilty of failure + probability of success = 1). Oh well.Perhaps, I thought chuckling to myself as I put the book down, this is why the young kids keep buying more and more servers whenever they run into capacity constraints.......page 256 assures them that each component approaches zero chance of failing as time approaches infinity and the component is left in service longer and longer and longer....... DOH !! :-)

Software Architecture in Practice (3rd Edition) (SEI Series in Software Engineering) Minimum Design Loads for Buildings and Other Structures, 3rd Printing (Standard ASCE/SEI 7-10) Non-Functional Requirements in Software Engineering (International Series in Software Engineering) Comparing ISO 9000, Malcolm Baldrige, And the SEI CMM for Software: A Reference and Selection Guide Software Failure: Management Failure: Amazing Stories and Cautionary Tales (Wiley Series in Software Engineering Practice) Error-Free Software: Know-How and Know-Why of Program Correctness (Wiley Series in Software Engineering Practice) Practical Software Reuse (Wiley Series in Software Engineering Practice) Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Sei personaggi in cerca d'autore (Italian Edition) The Pillow Book of Sei Shonagon The Pillow Book of Sei Shonagon (Classics) Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition) [Economy Edition] Software Components With Ada: Structures, Tools, and Subsystems (The Benjamin/Cummings Series in Ada and Software Engineering) Global Software Development Handbook (Applied Software Engineering Series) Constraint-Based Design Recovery for Software Reengineering: Theory and Experiments (International Series in Software Engineering) Practice Problems for the Civil Engineering PE Exam: A Companion to the Civil Engineering Reference Manual, 15th Ed Resilience Engineering in Practice: A Guidebook (Ashgate Studies in Resilience Engineering) Re-Engineering Software: How to Re-Use Programming to Build New, State-of-the-Art Software Object-oriented software development: Engineering software for reuse Software Reuse: Guidelines and Methods (Software Science and Engineering)