As recognized, adventure as without difficulty as experience roughly lesson, amusement, as capably as understanding can be gotten by just checking out a books free software architecture in practice second edition bass paul clements rick kazman also it is not directly done, you could assume even more nearly this life, something like the world.

We meet the expense of you this proper as competently as simple acquire to acquire those all. We have enough money free software architecture in practice second edition bass paul clements rick kazman and numerous book collections from fictions to scientific research in any way. in the midst of them is this free software architecture in practice second edition bass paul clements rick kazman that can be your partner.
Software Systems Architecture-Nick Rozanski 2005-04-20 Software Systems Architecture is a practitioner-oriented guide to designing and implementing effective architectures for information systems. It is both a readily accessible introduction to software architecture and an invaluable handbook of well-established best practices. It shows why the role of the architect is central to any successful information-systems development project, and, by presenting a set of architectural viewpoints and perspectives, provides specific direction for improving your own and your organization’s approach to software systems architecture. With this book you will learn how to Design an architecture that reflects and balances the different needs of its stakeholders Communicate the architecture to stakeholders and demonstrate that it has met their requirements Focus on architecturally significant aspects of design, including frequently overlooked areas such as performance, resilience, and location Use scenarios and patterns to drive the creation and validation of your architecture Document your architecture as a set of related views Use perspectives to ensure that your architecture exhibits important qualities such as performance, scalability, and security The architectural viewpoints and perspectives presented in the book also provide a valuable long-term reference source for new and experienced architects alike. Whether you are an aspiring or practicing software architect, you will find yourself referring repeatedly to the practical advice in this book throughout the lifecycle of your projects. A supporting Web site containing further information can be found at www.viewpoints-and-perspectives.info

97 Things Every Software Architect Should Know-Richard Monson-Haefel 2009-02-05 In this truly unique technical book, today’s leading software architects present valuable principles on key development issues that go way beyond technology. More than four dozen architects – including Neal Ford, Michael Nygard, and Bill de Bona – offer advice for communicating with stakeholders, eliminating complexity, empowering developers, and many more practical lessons they’ve learned from years of experience. Among the 97 principles in this book, you’ll find useful advice such as: Don’t Put Your Resume Ahead of the Requirements (Nitin Borwankar) Chances Are, Your Biggest Problem Isn’t Technical (Mark Ramm) Communication Is King; Clarity and Leadership, Its Humble Servant (John Edwards) Stop Thinking in Generalities Use Scenarios Before Use Cases (Kevin Henney) For the End User, the Interface Is the System (Vinayak Hegde) It’s Never Too Early to Think About Performance (Rebecca Parsons) To be a successful software architect, you need to master both business and technology. This book tells you what top software architects think is important and how they approach a project. If you want to enhance your career, 97 Things Every Software Architect Should Know is essential reading.

Building Evolutionary Architectures-Neal Ford 2017-09-18 The software development ecosystem is constantly changing, providing a constant stream of new tools, frameworks, techniques, and paradigms. Over the past few years, incremental developments in core engineering practices for software development have created the foundations for rethinking how architecture changes over time, along with ways to protect important architectural characteristics as it evolves. This practical guide ties those parts together with a new way to think about architecture and time.

Software Architecture Patterns for Serverless Systems-John Gilbert 2021-07-30 A professional’s guide to solving complex problems while designing modern software Key Features Learn best practices for designing enterprise-grade software systems Understand the importance of building reliable, maintainable, and scalable systems Become a professional software architect by learning the most effective software design patterns and architectural concepts Book Description As businesses are undergoing a digital transformation to keep up with competition, it is now more important than ever for IT professionals to design systems to keep up with the rate of change while maintaining stability. This book takes you through the architectural patterns that power enterprise-grade software systems and the key architectural elements that enable change such as events, autonomous services, and microfrontends, along with demonstrating how to implement and operate anti-fragile systems. You’ll divide up a system and define boundaries so that teams can work autonomously and accelerate the pace of innovation. The book also covers low-level event and data patterns that support the entire architecture, while getting you up and running with the different autonomous service design patterns. As you progress, you’ll focus on best practices for security, reliability, testability, observability, and performance. Finally, the book combines all that you’ve learned, explaining the methodologies of continuous experimentation, deployment, and delivery before providing you with some final thoughts on how to start making progress. By the end of this book, you’ll be able to architect your own event-driven, serverless systems that are ready to adapt and change so that you can deliver value at the pace needed by your business. What you will learn Explore architectural patterns to create anti-fragile systems that thrive with change Focus on DevOps practices that empower self-sufficient, full-stack teams Build enterprise-scale serverless systems Apply microservices principles to the frontend Discover how SOLID principles apply to software and database architecture Create event stream processors that power the event sourcing and COQS pattern Deploy a multi-regional system, including regional health checks, latency-based routing, and replication Explore the Stranger pattern for migrating legacy systems Who this book is for This book is for software architects and aspiring software architects who want to learn about different patterns and best practices to design better software. Intermediate-level experience in software development and design is required. Beginner-level knowledge of the cloud will also help you get the most out of this software design book.

Software Architecture-Oliver Vogel 2011-09-18 As a software architect you work in a wide-ranging and dynamic environment. You have to understand the needs of your customer, design architectures that satisfy both functional and non-functional requirements, and lead development teams in implementing the architecture. And it is an environment saturated with changing requirements, service-oriented architectures, cloud computing, and model-driven procedures open up new architectural possibilities. This book will help you to develop a holistic architectural awareness and knowledge base that extends beyond concrete methods, techniques, and technologies. It will also help you to acquire or expand the technical, methodological, and social competences that you need. The authors place the spotlight on you, the architect, and offer you long-term architectural orientation. They give you numerous guidelines, checklists, and best practices to support you in your practical work. “Software Architecture” offers IT students, software developers, and software architects a holistic and consistent orientation across relevant topics. The book also provides valuable information and suggestions for system architects and enterprise architects, since many of the topics presented are also relevant for their work. Furthermore, IT project leaders and other IT managers can use the book to acquire an enhanced understanding of architecture. Further information is available at www.software-architecture-book.org.


Enterprise Software Architecture and Design-Dominic Duggan 2012-02-28 This book fills a gap between high-level overview texts that are often too general and low-level detail oriented technical handbooks that lose sight the "big picture". This book discusses SOA from the low-level perspective of middleware, various XML-based technologies, and basic service design. It also examines broader implications of SOA, particularly where it intersects with business process management and process modeling. Concrete overviews will be provided of the methodologies in those fields, so that students will have a hands-on grasp of how they may be used in the context of SOA.
Software Architecture Knowledge Management—Muhammad Ali Babar 2010-05-03 A software architecture manifests the major early design decisions, which determine the system’s development, deployment and evolution. Thus, making better architectural decisions is one of the largest challenges in software engineering. Software architecture knowledge management is about capturing practical experience and translating it into generalized architectural knowledge, and using this knowledge in the communication with stakeholders during all phases of the software lifecycle. This book presents a concise description of knowledge management in the software architecture discipline. It explains the importance of sound knowledge management practices for improving software architecture processes and products, and makes clear the role of knowledge management in software architecture and software development processes. It presents many approaches that are in use in software companies today, approaches that have been used in other domains, and approaches under development in academia. After an initial introduction by the editors, the contributions are grouped in three parts on “Architecture Knowledge Management”, “Strategies and Approaches for Managing Architectural Knowledge”, and “Tools and Techniques for Managing Architectural Knowledge”. The presentation aims at information technology and software engineering professionals, in particular software architects and software architecture researchers. For the industrial audience, the book gives a broad and concise understanding of the importance of knowledge management for improving software architecture process and building capabilities in designing and evaluating better architectures for their mission- and business-critical systems. For researchers, the book will help to understand the applications of various knowledge management approaches in an industrial setting and to identify research challenges and opportunities.

Model-Driven Architecture in Practice—Oscar Pastor 2007-06-14 This book introduces all the relevant information required to understand and put Model Driven Architecture (MDA) into industrial practice. It clearly explains which conceptual primitives should be present in a system specification, how to use UML to properly represent this subset of basic conceptual constructs, how to identify just those diagrams and modeling constructs that are actually required to create a meaningful conceptual schema, and how to accomplish the transformation process between the problem space and the solution space. The approach is fully supported by commercially available tools.

Agile Software Architecture—Muhammad Ali Babar 2013-11-27 Agile software development approaches have had significant impact on industrial software development practices. Today, agile software development has penetrated to most IT companies across the globe, with an intention to increase quality, productivity, and profitability. Comprehensive knowledge is needed to understand the architectural challenges involved in adopting and using agile approaches and industrial practices to deal with the development of large, architecturally challenging systems in an agile way. Agile Software Architecture focuses on gaps in the requirements of applying architecture-centric approaches and principles of agile software development and demystifies the agile architecture paradox. Readers will learn how agile and architectural cultures can co-exist and support each other according to the context. Moreover, this book will also provide useful leads for future research in architecture and agile development of appropriate structures, topics, and case studies in agile methods. Presents a consolidated view of the state-of-art and state-of-practice as well as the newest research findings Identifies gaps in the requirements of applying architecture-centric approaches and principles of agile software development and demystifies the agile architecture paradox Explains whether or not and how agile and architectural cultures can co-exist and support each other depending upon the context Provides useful leads for future research in both architecture and agile to bridge such gaps by developing appropriate approaches, which incorporate architecturally sound practices in agile methods

Essential Software Architecture—Jan Gorton 2011-04-27 Job titles like “Technical Architect” and “Chief Architect” nowadays abound in software industry, yet many people suspect that “architecture” is one of the most overlooked and least understood terms in professional software development. Gorton’s book tries to resolve this dilemma. It concisely describes the essential elements of knowledge and key skills required to be a software architect. The explanations encompass the essentials of architectural thinking, practices, and supporting technologies. They range from a general understanding of structure and quality attributes through technical issues like middleware components and service-oriented architectures to recent technologies like model-driven architecture, software product lines, aspect-oriented design, and the Semantic Web, which will presumably influence future software systems. This second edition contains new material covering enterprise architecture, agile development, enterprise service bus technologies, RESTful Web services, and a case study on how to use the MedICI integration framework. All approaches are illustrated by an ongoing real-world example. So if you work as an architect or senior designer (or want to someday), or if you are a student in software engineering, here is a valuable and yet approachable knowledge source for you.

Continuous Architecture—Murat Erder 2015-10-21 Continuous Architecture provides a broad architectural perspective for continuous delivery, and describes a new architectural approach that supports and enables it. As the pace of innovation and software releases increases, IT departments are tasked to deliver value quickly and inexpensively to their business partners. With a focus on getting software into end-users hands faster, the ultimate goal of daily software updates is in sight to allow teams to ensure that they can release every change to the system simply and efficiently. This book presents an architectural approach to support modern application delivery methods and provide a broader architectural perspective, taking architectural concerns into account when deploying agile or continuous delivery approaches. The authors explain how to solve the challenges of implementing continuous delivery at the project and enterprise level, and the impact on IT processes including application testing, software deployment and software architecture. Covering the application of enterprise and software architecture concepts to the Agile and Continuous Delivery models Explains how to create an architecture that can evolve with applications Incorporates techniques including refactoring, architectural analysis, testing, and feedback-driven development Provides insight into incorporating modern software development when structuring teams and organizations

Fowler—Martin Fowler 2012-03-09 The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not easily implemented. Common failures in enterprise applications often occur because their developers do not understand the architectural lessons that experienced object developers have learned. Patterns of Enterprise Application Architecture is written in direct response to the stiff challenges that face enterprise application developers. The author, noted object-oriented designer Martin Fowler, noticed that despite changes in technology—from Smalltalk to CORBA to Java to .NET—the same basic design ideas can be adapted and applied to solve common problems. With the help of an expert group of contributors, Martin distills over forty recurring solutions into patterns. The result is an indispensable handbook of solutions that are applicable to any enterprise application platform. This book is actually two books in one. The first section is a short tutorial on developing applications, which you can read from start to finish to understand the scope of the book’s lessons. The next section, the bulk of the book, is a detailed reference to the patterns themselves. Each pattern provides usage and implementation information, as well as detailed code examples in Java or C#. The entire book is also richly illustrated with UML diagrams to further explain the concepts. Armed with this book, you will have the knowledge necessary to make important architectural decisions about building an enterprise application and the proven approaches that take these decisions to scale. Dividing an enterprise application into layers The major approaches to organizing business logic: An in-depth treatment of mapping between objects and relational databases Using Model-View-Controller to organize a Web presentation Handling concurrency for data that spans multiple transactions Designing distributed object interfaces

The Great Japan Exhibition—Royal Academy of Arts 1981


Software Architect’s Handbook—Joseph Ingham 2018-08-30 A comprehensive guide to exploring software architecture concepts and implementing best practices Key Features Enhance your skills to grow your career as a software architect.
Software Architecture in Practice—Len Bass 1998 The concepts and practice of software architecture are introduced—what a system is designed to do and how its components are meant to interact with each other. The authors cover not only essential technical topics for specifying and validating a system, but, for the first time, emphasize the importance of the business context in which large systems are designed.

Collaborative Enterprise Architecture—Stefan Bente 2012-09-01 Ever-changing business needs have prompted large companies to rethink their enterprise IT. Today, businesses must allow interaction with their customers, partners, and employees at more touch points and at a depth never thought previously. At the same time, rapid advances in information technologies, like business digitization, cloud computing, and Web 2.0, demand fundamental changes in the enterprises’ management practices. These changes have a drastic effect not only on IT and business, but also on policies, processes, and people. Many companies therefore embark on enterprise-wide transformation initiatives. The role of Enterprise Architecture (EA) is to architect and supervise this transformational journey. Unfortunately, today’s EA is often a ponderous and detached exercise, with most of the EA initiatives failing to create visible impact. The enterprises need an EA that is agile and responsive to business dynamics. Collaborative Enterprise Architecture provides the innovative solutions today’s enterprises require, informed by real-world experiences and experts’ insights. This book, in its first part, provides a systematic compendium of the current best practices in EA, analyzes current ways of doing EA, and identifies its constraints and shortcomings. In the second part, it leaves the beaten tracks of EA by introducing Lean, Agile, and Enterprise 2.0 concepts to the traditional EA methods. This blended approach to EA focuses on practical aspects, with recommendations derived from real-world experiences. A truly thought provoking and pragmatic guide to manage EA.

Designing Embedded Hardware—John Catsoulis 2002 Intelligent readers who want to build their own embedded computer systems—installed in everything from cell phones to cars to handheld organizers to refrigerators—will find this book to be the most in-depth, practical, and up-to-date guide on the market. Designing Embedded Hardware carefully steers between the practical and philosophical aspects, so developers can both create their own devices and gadgets and customize and extend the off-the-shelf systems. There are hundreds of books to choose from if you need to learn programming, but only a few are available if you want to learn to create hardware. Designing Embedded Hardware provides software and hardware engineers with no prior experience in embedded systems with the necessary components and building blocks to understand the architectures of embedded systems. Written to provide the depth of coverage and real-world examples developers need, Designing Embedded Hardware also provides a road-map to the pitfalls and traps to avoid in designing embedded systems. Designing Embedded Hardware covers such essential topics as: The principles of developing computer hardware Core hardware designs Assembly language concepts Parallel I/O Analog-digital conversion Timers (internal and external) UART Serial Peripheral Interface Inter-Integrated Circuit Bus Controller Area Network (CAN) Data Converter Interface (DCI) Low-power operation This invaluable and eminently useful book gives you the practical tools and skills to develop, build, and program your own application-specific computers.

Design It—Michael Keeling 2017-10-18 Don’t engineer by coincidence-design it like you mean it! Filled with practical techniques, Design It! is the perfect introduction to software architecture for programmers who are ready to grow their design skills. Lead your team as a software architect, ask the right stakeholders the right questions, explore design options, and help your team implement a system that promotes the right -ilities. Share your design decisions, facilitate collaborative design workshops that are fast, effective, and fun-and develop more awesome software! With dozens of design methods, examples, and practical know-how, Design It! shows you how

Software Systems Architecture—Nick Rozanski 2011-10-01 Software Systems Architecture, Second Edition is a highly regarded, practitioner-oriented guide to designing and implementing effective architectures for information systems. It is both a readily accessible introduction to software architecture and an invaluable handbook of well-established best practices. With this book you will learn how to Design and communicate an architecture that reflects and balances the different needs of its stakeholders Focus on architecturally significant aspects of design, including frequently overlooked areas such as performance, resilience, and location Use scenarios and patterns to drive the creation and validation of your architecture Document your architecture as a set of related views Reflect the new standards and developments in this area The new edition extends and updates much of the content, and Adds a “system context viewpoint” that documents the system’s interactions with its environment. The book begins by covering the fundamentals, benefits, and purpose of software architecture. You will discover how software architecture relates to an organization, followed by identifying its significant quality attributes. Once you explore design patterns and real-world case studies, you will learn how to develop effective software architectures using patterns and best practices. Explore the different considerations for designing software architecture Discover what it takes to continuously improve as a software architect Create loosely coupled systems who act together to achieve goals and objectives Discover how to create software architectures that evolve as the market, business requirements, frameworks, tools, and best practices change over time. By the end of this book, you will not only have studied software architecture concepts but also built the soft skills necessary to grow in this field. What you will learn Design software architectures using patterns and best practices. Answer the question: Who this book is for? The Software Architect’s Handbook is for you if you are a software architect, chief technical officer (CTO), or senior developer looking to gain a firm grasp of software architecture.

DevOps—Len Bass 2015-05-08 The First Complete Guide to DevOps for Software Architects DevOps promises to accelerate the release of new software features and improve monitoring of systems in production, but its crucial to your role as a software architect. This book presents a new lightweight perspective for software architects who are responsible for the design and implementation of effective software architectures and how it relates to the DevOps paradigm. This guide demonstrates the authors’ ideas in action with three real-world case studies: datacenter replication for business continuity, management of a continuous deployment pipeline, and re-architect legacy applications. Who this book is for? The Software Architect’s Handbook is for you if you are a software architect, chief technical officer (CTO), or senior developer looking to gain a firm grasp of software architecture.

to become a software architect. Walk through the core concepts every architect must know, discover how to apply them, and learn a variety of skills that will make you a better programmer, leader, and designer. Uncover the big ideas and patterns in software architecture and gain confidence in your ability to implement, and evaluate software architectures and collaborate with your team, stakeholders, and other architects. Identify the right stakeholders and understand their needs, dig for architecturally significant requirements, write amazing quality attribute scenarios, and make confident decisions. Choose technologies based on their architectural impact, facilitate architecture-centric design workshops, and evaluate architectures using whiteboards, documents, and code, and apply architecture-focused design methods in your day-to-day practice. Hands-on exercises, real-world scenarios, and practical team-based decision-making tools will get you everyone on board and give you the experience you need to become a confident software architect.

Clean Architecture—Robert C. Martin 2017-09-12 Practical Software Architecture Solutions from the Legendary Robert C. Martin ("Uncle Bob") By applying universal rules of software architecture, you can dramatically improve developer productivity throughout the life of any software system. Now, building upon the success of his best-selling books Clean Code and The Clean Coder, legendary software craftsman Robert C. Martin ("Uncle Bob") reveals those rules and helps you apply them. Martin’s Clean Architecture doesn’t merely present options. Drawing on over a half-century of experience in software environments of every imaginable type, Martin tells you what choices to make and why they are critical to your success. As you’ve come to expect from Uncle Bob, this book is packed with direct, no-nonsense solutions for the real challenges you’ll face—the ones that will make or break your projects. Learn what software architects need to achieve—and core disciplines and practices for achieving them! This guide provides design principles for address space, context, components, collaboration patterns, data and data management. See how programming paradigms impose discipline by restricting what developers can do. Understand what’s critically important and what’s merely a “detail”! Implement optimal, high-level structures for web, database, thick-client, console, and embedded applications. Define appropriate boundaries and layers, and organize components and services. See why designs and architectures go wrong, and how to prevent (or fix) these failures! Clean Architecture is essential reading for everyone currently engaging in architectural design, systems analyst, system designer, and software manager—and for every programmer who must execute someone else’s designs. Register your product for convenient access to downloads, updates, and/or corrections as they become available.

Becoming an Agile Software Architect—Rajesh R V 2021-03-19 A guide to successfully operating in a lean-agile organization for solutions architects and enterprise architects Key Features Develop the right combination of processes and technical excellence to address architectural challenges Explore a range of architectural techniques to modernize legacy systems Discover how to design and continuously improve well-architected sustainable software Book Description Many organizations have embraced Agile methodologies to transform their business. Now, building upon the success of his best-selling books Clean Code and The Clean Coder, legendary software craftsman Robert C. Martin ("Uncle Bob") reveals those rules and helps you apply them. Martin’s Clean Architecture doesn’t merely present options. Drawing on over a half-century of experience in software environments of every imaginable type, Martin tells you what choices to make and why they are critical to your success. As you’ve come to expect from Uncle Bob, this book is packed with direct, no-nonsense solutions for the real challenges you’ll face—the ones that will make or break your projects. Learn what software architects need to achieve—and core disciplines and practices for achieving them! This guide provides design principles for address space, context, components, collaboration patterns, data and data management. See how programming paradigms impose discipline by restricting what developers can do. Understand what’s critically important and what’s merely a “detail”! Implement optimal, high-level structures for web, database, thick-client, console, and embedded applications. Define appropriate boundaries and layers, and organize components and services. See why designs and architectures go wrong, and how to prevent (or fix) these failures! Clean Architecture is essential reading for everyone currently engaging in architectural design, systems analyst, system designer, and software manager—and for every programmer who must execute someone else’s designs. Register your product for convenient access to downloads, updates, and/or corrections as they become available.

Architecture and Health—Dina Battisto 2019-10-30 Architecture and Health recognizes the built environment as a major driver of health and wellness. Commonly constricted by normative building practices and transforms the dialogue into one of creativity and innovation. This book brings a timely focus to a subject matter that affects everyone: from individuals to entire societies. By synthesizing ideas from diverse fields, this book provides a comprehensive and integrative approach to how architecture and health are inextricably linked. The book features case studies from professionals across various disciplines, including architects, engineers, and health care providers, to illustrate how design choices can impact health outcomes. From the design of hospitals and healthcare facilities to the creation of sustainable communities, this book explores the role of architecture in promoting health and well-being. Through original research and expert insights, Architecture and Health offers a fresh perspective on the relationship between architecture and health, making it an essential resource for students, professionals, and anyone interested in the intersection of design and health. Whether you’re an architect, planner, or health professional, this book will inspire you to think differently about how we design our built environment and how it affects our health and well-being.

The Cathedral & the Bazaar—Eric S. Raymond 2001-02-01 Open source provides the competitive advantage in the Internet Age. According to the August Forrester Report, 56 percent of IT managers interviewed at Global 2,500 companies are already using some type of open source software in their infrastructure and another 6 percent will install it in the next two years. This revolutionary model for collaborative software development is being embraced and studied by many of the biggest players in the high-tech industry, from Sun Microsystems to IBM to Intel. The Cathedral & the Bazaar is a must for anyone who cares about the future of the computer industry or the dynamics of the information economy. Already, billions of dollars have been made and lost based on the ideas in this book. Its conclusions will be studied, debated, and implemented for years to come. According to Bob Young, "This is Eric Raymond’s great contribution to the success of the open source revolution, to the adoption of Linux-based operating systems, and to the success of open source users and the companies that supply them.” The interest in open source software development has grown enormously in the past year. This revised and expanded paperback edition includes new material on open source developments in 1999 and 2000. Raymond’s clear and effective writing style accurately describing the benefits of open source software has been key to its success. With major vendors creating acceptance for open source within companies, independent vendors will become the open source story in 2001.

Designing Data-Intensive Applications—Martin Kleppmann 2017-03-16 Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords? In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse landscape by examining the pros and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications. Peer under the hood of the systems you already use, and learn how to use and operate them more effectively. Make informed decisions by identifying the strengths and weaknesses of different tools. Navigate the trade-offs around consistency, scalability, fault tolerance, and complexity. Understand the distributed systems research upon which modern databases are built. Peek behind the scenes of major online services, and learn from their architectures.
Microsoft .NET - Architecting Applications for the Enterprise - Dino Esposito 2014-08-28 A software architect’s digest of core practices, pragmatically applied. Designing effective architecture is your best strategy for managing project complexity and improving your results. But the principles and practices of software architecture—what the authors call the “science of hard decisions”—have been evolving for cloud, mobile, and other shifts. Now fully revised and updated, this book shares the knowledge and real-world perspectives that enable you to design for success—and deliver more successful solutions. In this fully updated Second Edition, you will: Learn how only a deep understanding of domain can lead to appropriate architecture Examine domain-driven design in both theory and implementation. Shift your approach to code first, model later—including multilayer architecture. Capture the benefits of prioritizing software maintainability. See how readability, testability, and extensibility lead to code quality. Take a user experience (UX) first approach, rather than designing for data Review patterns for organizing business logic. Use event sourcing and CQRS together to model complex business domains more effectively. Delve inside the persistence layer, including patterns and implementation.

Affect, Architecture, and Practice - Akari Nakai Kidd 2021-05-17 Affect, Architecture, and Practice builds on and contributes to work in theories of affect that have risen within diverse disciplines, including geography, cultural studies, and media studies, challenging the nature of textual and representational-based research. Although numerous studies have examined how affect emerges in architectural spaces, little attention has been paid to the creative process of architectural design and the role that affect plays in the many contingencies and uncertainties that arise in the process. The book traces the critical, philosophic, and architectural theories to examine how affect, architecture, and practice are interlinked. Through a series of conversations and reflections, it examines three key contemporary architects, their practices and projects, all within a single coherent theme. Reiser + Umemoto (BIR Architecture DPC), USA, Kerstin Thompson Architects, Australia, and Shigeru Ban Architects, Japan, are critically studied through the lens of different aspects of practice, namely image-making, the design process, and the making of an everyday object/material. Through this investigation, author Akari Nakai Kidd demonstrates how affect theory allows a critical interrogation of the in-betweens of practice, its liminality and limits. It questions the stability of objects, the smooth temporality of practice, and its often under-conceptualised non-human dimensions. More significantly, the book demonstrates architectural practice’s contribution to the reconceptualisation of theories of affect.

Software Architecture Design Patterns in Java - Partha Kuchana 2004-04-27 Software engineering and computer science students need a resource that explains how to apply design patterns at the enterprise level, allowing them to design and implement systems of high stability and quality. Software Architecture Design Patterns in Java is a detailed explanation of how to apply design patterns and develop software architectures. It provides in-depth examples in Java, and guides students by detailing when, why, and how to use specific patterns. This textbook presents 42 design patterns, including 23 GoF patterns. Categories include: Basic, Creational, Collectional, Structural, Behavioral, and Concurrency, with multiple examples for each. The discussion of each pattern includes an example implemented in Java. The source code for all examples is found on a companion Web site. The author explains the content so that it is easy to understand, and each pattern discussion includes Practice Questions to aid instructors. The textbook concludes with a case study that pulls several patterns together to demonstrate how patterns are not applied in isolation, but collaborate within domains to solve complicated problems.

The Social (Re)Production of Architecture - Doina Petrescu 2017-07-14 The Social (Re)Production of Architecture brings the debates of the ‘right to the city’ into today’s context of ecological, economic and social crises. Building on the 1970s' discussions about the ‘production of space’, which French sociologist Henri Lefebvre considered a civic right, the authors question who has the right to make space, and explore the kinds of relations that are produced in the process. In the emerging post-capitalist era, this book addresses urgent social and ecological imperatives for change and opens up questions around architecture's engagement with new forms of organization and practice. The book asks what (new) kinds of ‘social’ can architecture (re)produce, and what kinds of politics, values and actions are needed. The book features 24 interdisciplinary essays written by leading theorists and practitioners including social thinkers, economic theorists, architects, educators, urban curators, feminists, artists and activists from different generations and global contexts. The essays discuss the diverse, global locations with work taking different and specific forms in these different contexts. A cutting-edge, critical text which rethinks both practice and theory in the light of recent crises, making it key reading for students, academics and practitioners.

Beautiful Architecture - Domidis Spinellis 2009-01-15 What are the ingredients of robust, elegant, flexible, and maintainable software architecture? Beautiful Architecture answers this question through a collection of intriguing essays from more than a dozen of today’s leading software designers and architects. In each essay, contributors present a notable software architecture, and analyze what makes it innovative and ideal for its purpose. Some of the engineers in this book reveal how they developed a specific project, including decisions they faced and tradeoffs they made. Others take a step back to investigate how certain architectural aspects have influenced computing as a whole. With this book, you’ll discover: How Facebook’s architecture is the basis for a data-centric application ecosystem The effect of Zen’s well-designed architecture on the way operating systems evolve How community processes within the KDE project help software architectures evolve from rough sketches to beautiful systems How creeping featurism has helped GNU Emacs gain unanticipated functionality The magic behind the Jikes RVM self-optimizable, self-hosting runtime Design choices and building blocks that made Tandem’s choice platform in high-availability environments for over two decades Differences and similarities between object-oriented and functional architectural views How architectures can affect the software's evolution and the developers’ engagement. Go behind the scenes to learn what it takes to design elegant software architecture, and how it can shape the way you approach your own projects, with Beautiful Architecture.