

# The Unified Software Development Process By Ivar Jacobson

Thank you certainly much for downloading **the unified software development process by ivar jacobson**. Most likely you have knowledge that, people have seen numerous times for their favorite books taking into account this the unified software development process by ivar jacobson, but stop occurring in harmful downloads.

Rather than enjoying a good book past a cup of coffee in the afternoon, instead they juggled in imitation of some harmful virus inside their computer. **the unified software development process by ivar jacobson** is within reach in our digital library an online right of entry to it is set as public suitably you can download it instantly. Our digital library saves in compound countries, allowing you to get the most less latency epoch to download any of our books following this one. Merely said, the the unified software development process by ivar jacobson is universally compatible taking into consideration any devices to read.

Developing Applications with Visual Basic and UML Paul R. Reed 2000 Reed's guide includes detailed coverage of architecting VB enterprise applications and features working examples and step-by-step instructions for planning and development of an order entry system, detailing do's and don't's for analysis, design and construction. CD-ROM contains several templates for applying UML, as well as complete Rational Rose models for the sample applications.

*The Essentials of Modern Software Engineering* Ivar Jacobson 2019-07-19 The first course in software engineering is the most critical. Education must start from an understanding of the heart of software development, from familiar ground that is common to all software development endeavors. This book is an in-depth introduction to software engineering that uses a systematic, universal kernel to teach the essential elements of all software engineering methods. This kernel, Essence, is a vocabulary for defining methods and practices. Essence was envisioned and originally created by Ivar Jacobson and his colleagues, developed by Software Engineering Method and Theory (SEMAT) and approved by The Object Management Group (OMG) as a standard in 2014. Essence is a practice-independent framework for thinking and reasoning about the practices we have and the practices we need. Essence establishes a shared and standard understanding of what is at the heart of software development. Essence is agnostic to any particular method, lifecycle independent, programming language independent, concise, scalable, extensible, and formally specified. Essence frees the practices from their method prisons. The first part of the book describes Essence, the essential elements to work with, the essential things to do and the essential competencies you need when developing software. The other three parts describe more and more advanced use cases of Essence. Using real but manageable examples, it covers the fundamentals of Essence and the innovative use of serious games to support software engineering. It also explains how current practices such as user stories, use cases, Scrum, and micro-services can be described using Essence, and illustrates how their activities can be represented using the Essence notions of cards and checklists. The fourth part of the book offers a vision how Essence can be scaled to support large, complex systems engineering. Essence is supported by an ecosystem developed and maintained by a community of experienced people worldwide. From this ecosystem, professors and students can select what they need and create their own way of working, thus learning how to create ONE way of working that matches the particular situation and needs.

**The Unified Software Development Process** Ivar Jacobson 2003

**Managing Software Requirements** Dean Leffingwell 2000 A classic treatise that defined the field of applied demand analysis, *Consumer Demand in the United States: Prices, Income, and Consumption Behavior* is now fully updated and expanded for a new generation. Consumption expenditures by households in the United States account for about 70% of America's GDP. The primary focus in this book is on how households adjust these expenditures in response to changes in price and income. Econometric estimates of price and income elasticities are obtained for an exhaustive array of goods and services using data from surveys conducted by the Bureau of Labor Statistics, providing a better understanding of consumer demand. Practical models for forecasting future price and income elasticities are also demonstrated. Fully revised with over a dozen new chapters and appendices, the book revisits the original Taylor-Houthakker models while examining new material as well, such as the use of quantile regression and the stationarity of consumer

preference. It also explores the emerging connection between neuroscience and consumer behavior, integrating the economic literature on demand theory with psychology literature. The most comprehensive treatment of the topic to date, this volume will be an essential resource for any researcher, student or professional economist working on consumer behavior or demand theory, as well as investors and policymakers concerned with the impact of economic fluctuations.

*Software Project Management* Walker Royce 1998 Software Project Management explains the latest management strategies and techniques in software developments. It covers such issues as keeping the team motivated, cost-justifying strategies, deadlines and budgets.

**OMT Insights** James Rumbaugh 1996 This book presents the collected writings of OMT guru Dr James Rumbaugh. These articles encompass the development, refinement, and current state of OMT.

UML 2 and the Unified Process Jim Arlow 2005-06-27 "This book manages to convey the practical use of UML 2 in clear and understandable terms with many examples and guidelines. Even for people not working with the Unified Process, the book is still of great use. UML 2 and the Unified Process, Second Edition is a must-read for every UML 2 beginner and a helpful guide and reference for the experienced practitioner." --Roland Leibundgut, Technical Director, Zuehlke Engineering Ltd. "This book is a good starting point for organizations and individuals who are adopting UP and need to understand how to provide visualization of the different aspects needed to satisfy it." --Eric Naiburg, Market Manager, Desktop Products, IBM Rational Software This thoroughly revised edition provides an indispensable and practical guide to the complex process of object-oriented analysis and design using UML 2. It describes how the process of OO analysis and design fits into the software development lifecycle as defined by the Unified Process (UP). UML 2 and the Unified Process contains a wealth of practical, powerful, and useful techniques that you can apply immediately. As you progress through the text, you will learn OO analysis and design techniques, UML syntax and semantics, and the relevant aspects of the UP. The book provides you with an accurate and succinct summary of both UML and UP from the point of view of the OO analyst and designer. This book provides Chapter roadmaps, detailed diagrams, and margin notes allowing you to focus on your needs. Outline summaries for each chapter, making it ideal for revision, and a comprehensive index that can be used as a reference. New to this edition: Completely revised and updated for UML 2 syntax. Easy to understand explanations of the new UML 2 semantics. More real-world examples. A new section on the Object Constraint Language (OCL). Introductory material on the OMG's Model Driven Architecture (MDA). The accompanying website provides a complete example of a simple e-commerce system. Open source tools for requirements engineering and use case modeling. Industrial-strength UML course materials based on the book.

The Rational Unified Process Philippe Kruchten 2000 The second edition of this text brings the content up to date and in compliance with Rational Unified Process 2000. It defines the process, putting it into a proper software development context, reviewing the RUP's history and providing detailed coverage of its structure.

*The Unified Modeling Language Reference Manual* James Rumbaugh 2010 "If you are a serious user of UML, there is no other book quite like this one. I have been involved with the UML specification process for some time, but I still found myself learning things while reading through this book-especially on the changes and new capabilities that have come with UML." -Ed Seidewitz, Chief Architect, IntelliData Technologies Corporation The latest version of the Unified Modeling Language-UML 2.0-has increased its capabilities as the

standard notation for modeling software-intensive systems. Like most standards documents, however, the official UML specification is difficult to read and navigate. In addition, UML 2.0 is far more complex than previous versions, making a thorough reference book more essential than ever. In this significantly updated and expanded edition of the definitive reference to the standard, James Rumbaugh, Ivar Jacobson, and Grady Booch-the UML's creators-clearly and completely describe UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Whether you are capturing requirements, developing software architectures, designing implementations, or trying to understand existing systems, this is the book for you. Highlights include: Alphabetical dictionary of articles covering every UML concept Integrated summary of UML concepts by diagram type Two-color diagrams with extensive annotations in blue Thorough coverage of both semantics and notation, separated in each article for easy reference Further explanations of concepts whose meaning or purpose is obscure in the original specifications Discussion sections offering usage advice and additional insight into tricky concepts Notation summary, with references to individual articles An enhanced online index available on the book's web site allowing readers to quickly and easily search the entire text for specific topics The result is an indispensable resource for anyone who needs to understand the inner workings of the industry standard modeling language.

**Systems Engineering with SysML/UML** Tim Weilkiens 2011-08-29 UML, the Universal Modeling Language, was the first programming language designed to fulfill the requirement for "universality." However, it is a software-specific language, and does not support the needs of engineers designing from the broader systems-based perspective. Therefore, SysML was created. It has been steadily gaining popularity, and many companies, especially in the heavily-regulated Defense, Automotive, Aerospace, Medical Device and Telecomms industries, are already using SysML, or are planning to switch over to it in the near future. However, little information is currently available on the market regarding SysML. Its use is just on the crest of becoming a widespread phenomenon, and so thousands of software engineers are now beginning to look for training and resources. This book will serve as the one-stop, definitive guide that provide an introduction to SysML, and instruction on how to implement it, for all these new users. \*SysML is the latest emerging programming language--250,000 estimated software systems engineers are using it in the US alone! \*The first available book on SysML in English \*Insider information! The author is a member of the SysML working group and has written sections of the specification \*Special focus comparing SysML and UML, and explaining how both can work together

**The Essentials of Modern Software Engineering** Ivar Jacobson 2019-07-19 The first course in software engineering is the most critical. Education must start from an understanding of the heart of software development, from familiar ground that is common to all software development endeavors. This book is an in-depth introduction to software engineering that uses a systematic, universal kernel to teach the essential elements of all software engineering methods. This kernel, Essence, is a vocabulary for defining methods and practices. Essence was envisioned and originally created by Ivar Jacobson and his colleagues, developed by Software Engineering Method and Theory (SEMAT) and approved by The Object Management Group (OMG) as a standard in 2014. Essence is a practice-independent framework for thinking and reasoning about the practices we have and the practices we need. Essence establishes a shared and standard understanding of what is at the heart of software development. Essence is agnostic to any particular method, lifecycle independent, programming language independent, concise, scalable, extensible, and formally specified. Essence frees the practices from their method prisons. The first part of the book describes Essence, the essential elements to work with, the essential things to do and the essential competencies you need when developing software. The other three parts describe more and more advanced use cases of Essence. Using real but manageable examples, it covers the fundamentals of Essence and the innovative use of serious games to support software engineering. It also explains how current practices such as user stories, use cases, Scrum, and micro-services can be described using Essence, and illustrates how their activities can be represented using the Essence notions of cards and checklists. The fourth part of the book offers a vision how Essence can be scaled to support large, complex systems engineering. Essence is supported by an ecosystem developed and maintained by a community of experienced people worldwide. From this ecosystem, professors and students can select what they need and create their own way of working, thus

learning how to create ONE way of working that matches the particular situation and needs.

**Just Enough Software Architecture** George Fairbanks 2010-08-30 This is a practical guide for software developers, and different than other software architecture books. Here's why: It teaches risk-driven architecting. There is no need for meticulous designs when risks are small, nor any excuse for sloppy designs when risks threaten your success. This book describes a way to do just enough architecture. It avoids the one-size-fits-all process tar pit with advice on how to tune your design effort based on the risks you face. It democratizes architecture. This book seeks to make architecture relevant to all software developers. Developers need to understand how to use constraints as guiderails that ensure desired outcomes, and how seemingly small changes can affect a system's properties. It cultivates declarative knowledge. There is a difference between being able to hit a ball and knowing why you are able to hit it, what psychologists refer to as procedural knowledge versus declarative knowledge. This book will make you more aware of what you have been doing and provide names for the concepts. It emphasizes the engineering. This book focuses on the technical parts of software development and what developers do to ensure the system works not job titles or processes. It shows you how to build models and analyze architectures so that you can make principled design tradeoffs. It describes the techniques software designers use to reason about medium to large sized problems and points out where you can learn specialized techniques in more detail. It provides practical advice. Software design decisions influence the architecture and vice versa. The approach in this book embraces drill-down/pop-up behavior by describing models that have various levels of abstraction, from architecture to data structure design.

**Visual Modeling with Rational Rose 2000 and UML** Terry Quatrani 2000 The first UML book to cover Rational Rose 2000, this brand-new edition reviews the three key interrelated components of state-of-the-art software system design: the Rational Unified process, the Unified Modeling Language, and Rational Rose 2000. Then, through a simplified case study, it walks developers through a real-world business system. Includes screen shots demonstrating UML at work in the Rational Rose 2000 modeling tool.

**The Unified Process Explained** Kendall Scott 2002 Since its inception Research in Labor Economics has published over 350 articles encompassing a wide range of themes and spanning an array of labor economics topics. Authors have ranged from young scholars with much potential to mature leaders in the field, including Nobel Prize and John Bates Clark award winners. Over the years Research in Labor Economics has continued to present important new research in labor economics. It covers themes such as labor supply, work effort, schooling, on-the-job training, earnings distribution, discrimination, migration, and the effects of government policies on worker well-being. It aims to apply economic theory and econometrics to analyze important policy-related questions, often with an international focus. To commemorate Research in Labor Economics's 35th anniversary, this retrospective edition contains 20 of the most influential Research in Labor Economics articles along with new introductory prefatory updates written by the original authors. These new prefaces emphasize recent developments that each article might have inspired and also discuss remaining unanswered questions.

**Head First Object-Oriented Analysis and Design** Brett McLaughlin 2006-11-27 Provides information on analyzing, designing, and writing object-oriented software.

**The Unified Modeling Language User Guide** Grady Booch 2017-07-12 For nearly ten years, the Unified Modeling Language (UML) has been the industry standard for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. As the de facto standard modeling language, the UML facilitates communication and reduces confusion among project stakeholders. The recent standardization of UML 2.0 has further extended the language's scope and viability. Its inherent expressiveness allows users to model everything from enterprise information systems and distributed Web-based applications to real-time embedded systems. In this eagerly anticipated revision of the best-selling and definitive guide to the use of the UML, the creators of the language provide a tutorial to its core aspects in a two-color format designed to facilitate learning. Starting with an overview of the UML, the book explains the language gradually by introducing a few concepts and notations in each chapter. It also illustrates the application of the UML to complex modeling problems across a variety of application domains. The in-depth coverage and example-driven approach that made the first edition of The Unified Modeling Language User Guide an indispensable resource remain unchanged. However, content has been thoroughly updated to

reflect changes to notation and usage required by UML 2.0. Highlights include: A new chapter on components and internal structure, including significant new capabilities for building encapsulated designs New details and updated coverage of provided and required interfaces, collaborations, and UML profiles Additions and changes to discussions of sequence diagrams, activity diagrams, and more Coverage of many other changes introduced by the UML 2.0 specification With this essential guide, you will quickly get up to speed on the latest features of the industry standard modeling language and be able to apply them to your next software project.

*Agility and Discipline Made Easy* Per Kroll 2006-05-19 "The Japanese samurai Musashi wrote: 'One can win with the long sword, and one can win with the short sword. Whatever the weapon, there is a time and situation in which it is appropriate.' "Similarly, we have the long RUP and the short RUP, and all sizes in between. RUP is not a rigid, static recipe, and it evolves with the field and the practitioners, as demonstrated in this new book full of wisdom to illustrate further the liveliness of a process adopted by so many organizations around the world. Bravo!" --Philippe Kruchten, Professor, University of British Columbia "The Unified Process and its practices have had, and continue to have, a great impact on the software industry. This book is a refreshing new look at some of the principles underlying the Unified Process. It is full of practical guidance for people who want to start, or increase, their adoption of proven practices. No matter where you are today in terms of software maturity, you can start improving tomorrow." --Ivar Jacobson, Ivar Jacobson Consulting "Kroll and MacIsaac have written a must-have book. It is well organized with new principles for software development. I encounter many books I consider valuable; I consider this one indispensable, especially as it includes over 20 concrete best practices. If you are interested in making your software development shop a better one, read this book!" --Ricardo R. Garcia, President, Global Rational User Group Council, [www.rational-ug.org/index.php](http://www.rational-ug.org/index.php) "Agile software development is real, it works, and it's here to stay. Now is the time to come up to speed on agile best practices for the Unified Process, and this book provides a great starting point." --Scott W. Ambler, practice leader, Agile Modeling "IBM and the global economy have become increasingly dependent on software over the last decade, and our industry has evolved some discriminating best practices. Per and Bruce have captured the principles and practices of success in this concise book; a must for executives, project managers, and practitioners. These ideas are progressive, but they strike the right balance between agility and governance and will form the foundation for successful systems and software developers for a long time." --Walker Royce, Vice President, IBM Software Services-Rational "Finally, the RUP is presented in digestible, byte-size pieces. Kroll and MacIsaac effectively describe a set of practices that can be adopted in a low-ceremony, ad hoc fashion, suited to the culture of the more agile project team, while allowing them to understand how to scale their process as needed." --Dean Leffingwell, author and software business advisor and executive "This text fills an important gap in the knowledge-base of our industry: providing agile practices in the proven, scalable framework of the Unified Process. With each practice able to be throttled to the unique context of a development organization, Kroll and MacIsaac provide software teams with the ability to balance agility and discipline as appropriate for their specific needs." --Brian G. Lyons, CTO, Number Six Software, Inc. In *Agility and Discipline Made Easy*, Rational Unified Process (RUP) and Open Unified Process (OpenUP) experts Per Kroll and Bruce MacIsaac share twenty well-defined best practices that you and your team can start adopting today to improve the agility, predictability, speed, and cost of software development. Kroll and MacIsaac outline proven principles for software development, and supply a number of supporting practices for each. You'll learn what problems each practice addresses and how you can best leverage RUP and OpenUP (an open-source version of the Unified Process) to make the practice work for you. You'll find proactive, prescriptive guidance on how to adopt the practices with minimal risk and implement as much or as little of RUP or OpenUP as you want. Learn how to apply sample practices from the Unified Process so you can Execute your project in iterations Embrace and manage change Test your own code Describe requirements from the user perspective Architect with components and services Model key perspectives Whether you are interested in agile or disciplined development using RUP, OpenUP, or other agile processes, this book will help you reduce the anxiety and cost associated with software improvement by providing an easy, non-intrusive path toward improved results--without overwhelming you and your team.

*The Rational Unified Process* Philippe Kruchten 1999 This concise book offers a quick introduction to the

concepts, structure, content, and motivation of the Rational Unified Process. This revolutionary software development process provides a disciplined approach to assigning, managing, and completing tasks within a software development organization and is the first development process to exploit the full capabilities of the industry-standard Unified Modeling Language. The Rational Unified Process is unique in that it captures many of the proven best practices in modern software development and presents them in a form that can be tailored to a wide range of projects and organizations. In this book, you will discover: what the Rational Unified Process is - and what it is not; the concepts used in the Rational Unified Process, as well as its structure; the best practices that have been synthesized into this process; and how this process can provide the guidance you need for your specific project responsibilities.

*Use Case Modeling* Kurt Bittner 2003 Discusses how to define and organize use cases that model the user requirements of a software application. The approach focuses on identifying all the parties who will be using the system, then writing detailed use case descriptions and structuring the use case model. An ATM example runs throughout the book. The authors work at Rational Software. Annotation copyrighted by Book News, Inc., Portland, OR

*The Rational Unified Process Made Easy* Per Kroll 2003 The authors explain the underlying software development principles behind the RUP, and guide readers in its application in their organization.

*Model-Driven Software Engineering in Practice* Marco Brambilla 2017-03-30 This book discusses how model-based approaches can improve the daily practice of software professionals. This is known as Model-Driven Software Engineering (MDSE) or, simply, Model-Driven Engineering (MDE). MDSE practices have proved to increase efficiency and effectiveness in software development, as demonstrated by various quantitative and qualitative studies. MDSE adoption in the software industry is foreseen to grow exponentially in the near future, e.g., due to the convergence of software development and business analysis. The aim of this book is to provide you with an agile and flexible tool to introduce you to the MDSE world, thus allowing you to quickly understand its basic principles and techniques and to choose the right set of MDSE instruments for your needs so that you can start to benefit from MDSE right away. The book is organized into two main parts. The first part discusses the foundations of MDSE in terms of basic concepts (i.e., models and transformations), driving principles, application scenarios, and current standards, like the well-known MDA initiative proposed by OMG (Object Management Group) as well as the practices on how to integrate MDSE in existing development processes. The second part deals with the technical aspects of MDSE, spanning from the basics on when and how to build a domain-specific modeling language, to the description of Model-to-Text and Model-to-Model transformations, and the tools that support the management of MDSE projects. The second edition of the book features: a set of completely new topics, including: full example of the creation of a new modeling language (IFML), discussion of modeling issues and approaches in specific domains, like business process modeling, user interaction modeling, and enterprise architecture complete revision of examples, figures, and text, for improving readability, understandability, and coherence better formulation of definitions, dependencies between concepts and ideas addition of a complete index of book content In addition to the contents of the book, more resources are provided on the book's website <http://www.mdse-book.com>, including the examples presented in the book.

*Object-oriented Software Engineering* Timothy Christian Lethbridge 2004 This book covers the essential knowledge and skills needed by a student who is specializing in software engineering. Readers will learn principles of object orientation, software development, software modeling, software design, requirements analysis, and testing. The use of the Unified Modelling Language to develop software is taught in depth. Many concepts are illustrated using complete examples, with code written in Java.

*The Road to the Unified Software Development Process* Ivar Jacobson 2000-07-24 Ivar Jacobson, one of the Three Amigos of Rational, follows his fellow amigos, Grady Booch and James Rumbaugh, with the publication of *The Road to the Unified Software Development Process*, his own collection of the best of his work.

Together with Stefan Bylund, Dr. Jacobson has gathered the best of his articles from *Object Magazine*, *JOOP*, and *ROAD*, and updated them to reflect current trends in the industry. This book not only presents the best of his work, but it also tracks the development of the new Unified Software Development Process. This book is an excellent reference for software professionals who are interested in analysis and design. It provides real-world experience in developing quality software through disciplined engineering.



*The Unified Software Development Process* Ivar Jacobson 1999

**Executable UML** Stephen J. Mellor 2002 Executable UML can help organizations implement working software systems. This book shows how UML can be used to execute code.

UML Distilled Martin Fowler 2018-08-30 More than 300,000 developers have benefited from past editions of UML Distilled . This third edition is the best resource for quick, no-nonsense insights into understanding and using UML 2.0 and prior versions of the UML. Some readers will want to quickly get up to speed with the UML 2.0 and learn the essentials of the UML. Others will use this book as a handy, quick reference to the most common parts of the UML. The author delivers on both of these promises in a short, concise, and focused presentation. This book describes all the major UML diagram types, what they're used for, and the basic notation involved in creating and deciphering them. These diagrams include class, sequence, object, package, deployment, use case, state machine, activity, communication, composite structure, component, interaction overview, and timing diagrams. The examples are clear and the explanations cut to the fundamental design logic. Includes a quick reference to the most useful parts of the UML notation and a useful summary of diagram types that were added to the UML 2.0. If you are like most developers, you don't have time to keep up with all the new innovations in software engineering. This new edition of Fowler's classic work gets you acquainted with some of the best thinking about efficient object-oriented software design using the UML--in a convenient format that will be essential to anyone who designs software professionally.

**The Unified Software Development Process** Ivar Jacobson 1999 This landmark book provides a thorough overview of the Unified Process for software development, with a practical focus on modeling using the Unified Modeling Language (UML). The Unified Process goes beyond mere object-oriented analysis and design to spell out a proven family of techniques that supports the complete software development life cycle. The result is a component-based process that is use-case driven, architecture-centric, iterative, and incremental. The Unified Process takes full advantage of the industry-standard Unified Modeling Language. This book demonstrates how the notation and process complement one another, using UML models to illustrate the new process in action. The authors clearly describe the semantics and notation of the different higher-level constructs used in the models. Constructs such as use cases, actors, subsystems, classes, interfaces, active classes, processes, threads, nodes, and most relations are described in the context of a model. Object technology practitioners and software engineers familiar with the authors' past work will appreciate *The Unified Software Development Process* as a useful means of learning the current best practices in software development.

**Object-oriented Software Engineering** Ivar Jacobson 1993

**The Economics of Iterative Software Development** Walker Royce 2009-03-23 Results-Based Software Management: Achieve Better Outcomes with Finite Resources Effective software development is no longer merely an IT concern: today, it is crucial to the entire enterprise. However, most businesspeople are not ready to make informed decisions about software initiatives. *The Economics of Iterative Software Development: Steering Toward Better Business Results* will prepare them. Drawing on decades of software development and business experience, the authors demonstrate how to utilize practical, economics-based techniques to plan and manage software projects for maximum return on technology investments. The authors begin by dispelling widespread myths about software costs, explaining why traditional, “engineering-based” software management introduces unacceptable inefficiencies in today’s development environments. Next, they show business and technical managers how to combine the principles of economics and iterative development to achieve optimal results with limited resources. Using their techniques, readers will learn how to build systems that enable maximum business innovation and process improvement—and implement software processes that allow them to do so consistently. Highlights include How to repeatedly quantify the value a project is delivering and quickly adjust course as needed How to reduce software project size, complexity, and other “project killers” How to identify and eliminate software development processes that don’t work How to improve development processes, reduce rework, mitigate risk, and identify inefficiencies How to create more proficient teams by improving individual skills, team interactions, and organizational capability Where to use integrated, automated tools to improve effectiveness What to measure, and when: specific metrics for project inception, elaboration, construction, and transition *The Economics of Iterative*

*Software Development: Steering Toward Better Business Results* will help both business and technical managers make better decisions throughout the software development process—and it will help team and project leaders keep any project or initiative on track, so they can deliver more value faster.

*Enterprise Modeling with UML* Chris Marshall 2000 CD-ROM contains: Java and XML implementations of ideas and models described in the appendix.

**Explanatory Model Analysis** Przemyslaw Biecek 2021-02-15 Explanatory Model Analysis Explore, Explain and Examine Predictive Models is a set of methods and tools designed to build better predictive models and to monitor their behaviour in a changing environment. Today, the true bottleneck in predictive modelling is neither the lack of data, nor the lack of computational power, nor inadequate algorithms, nor the lack of flexible models. It is the lack of tools for model exploration (extraction of relationships learned by the model), model explanation (understanding the key factors influencing model decisions) and model examination (identification of model weaknesses and evaluation of model's performance). This book presents a collection of model agnostic methods that may be used for any black-box model together with real-world applications to classification and regression problems.

The Rational Unified Process Philippe Kruchten 2004 bull; Reflects all of the changes that were integrated into RUP v2003-the latest version of the very popular product bull; Learn the key concepts, fundamentals of structure, integral content, and motivation behind the RUP bull; Covers all phases of the software development lifecycle -from concept, to delivery, to revision

*Advanced Use Case Modeling* Frank Armour 2000-12-29 "This book isn't just another introduction to use cases. The authors have used their wealth of experience to produce an excellent and insightful collection of detailed examples, explanations, and advice on how to work with use cases." –Maria Ericsson The toughest challenge in building a software system that meets the needs of your audience lies in clearly understanding the problems that the system must solve. *Advanced Use Case Modeling* presents a framework for discovering, identifying, and modeling the problem that the software system will ultimately solve. Software developers often employ use cases to specify what should be performed by the system they're constructing. Although use case-driven analysis, design, and testing of software systems has become increasingly popular, little has been written on the role of use cases in the complete software cycle. This book fills that need by describing how to create use case models for complex software development projects, using practical examples to explain conceptual information. The authors extend the work of software visionary Ivar Jacobson, using the Unified Modeling Language (UML) as the notation to describe the book's models. Aimed primarily at software professionals, *Advanced Use Case Modeling* also includes information that relates use case technique to business processes. This book presents a process for creating and maintaining use case models in a framework that can be fully customized for your organization. The authors, pioneers in the application of use cases in software development, bring their extensive experience to cover topics such as: A process model for applying a use case model How to keep your use case modeling effort on track Tips and pitfalls in use case modeling How to organize your use case model for large-system development Similarities between *Advanced Use Case Modeling* and the Rational Unified Process framework Effect of use cases on user interface design Guidelines for quality use case modeling

**Object-Oriented Analysis and Design for Information Systems** Raul Sidnei Wazlawick 2014-01-28 *Object-Oriented Analysis and Design for Information Systems* clearly explains real object-oriented programming in practice. Expert author Raul Sidnei Wazlawick explains concepts such as object responsibility, visibility and the real need for delegation in detail. The object-oriented code generated by using these concepts in a systematic way is concise, organized and reusable. The patterns and solutions presented in this book are based in research and industrial applications. You will come away with clarity regarding processes and use cases and a clear understanding of how to expand a use case. Wazlawick clearly explains clearly how to build meaningful sequence diagrams. *Object-Oriented Analysis and Design for Information Systems* illustrates how and why building a class model is not just placing classes into a diagram. You will learn the necessary organizational patterns so that your software architecture will be maintainable. Learn how to build better class models, which are more maintainable and understandable. Write use cases in a more efficient and standardized way, using more effective and less complex diagrams. Build true object-oriented code with division of responsibility and delegation.

**Managing Iterative Software Development Projects** Kurt Bittner 2006-06-27 The Practical, Start-to-Finish Guide to Planning and Leading Iterative Software Projects Iterative processes have gained widespread acceptance because they help software developers reduce risk and cost, manage change, improve productivity, and deliver more effective, timely solutions. But conventional project management techniques don't work well in iterative projects, and newer iterative management techniques have been poorly documented. Managing Iterative Software Development Projects is the solution: a relentlessly practical guide to planning, organizing, estimating, staffing, and managing any iterative project, from start to finish. Leading iterative development experts Kurt Bittner and Ian Spence introduce a proven, scalable approach that improves both agility and control at the same time, satisfying the needs of developers, managers, and the business alike. Their techniques are easy to understand, and easy to use with any iterative methodology, from Rational Unified Process to Extreme Programming to the Microsoft Solutions Framework. Whatever your role—team leader, program manager, project manager, developer, sponsor, or user representative—this book will help you Understand the key drivers of success in iterative projects Leverage “time boxing” to define project lifecycles and measure results Use Unified Process phases to facilitate controlled iterative development Master core concepts of iterative project management, including layering and evolution Create project roadmaps, including release plans Discover key patterns of risk management, estimation, organization, and iteration planning Understand what must be controlled centrally, and what you can safely delegate Transition smoothly to iterative processes Scale iterative project management from the smallest to the largest projects Align software investments with the needs of the business Whether you are interested in software development using RUP, OpenUP, or other agile processes, this book will help you reduce the anxiety and cost associated with software improvement by providing an easy, non-intrusive path toward improved results—without overwhelming you and your team.

**Object-oriented Development at Work** Ruth Malan 1996 Fusion is an easy-to-learn, easy-to-use method for designing object-oriented (oo) software that was created at Hewlett-Packard Labs by merging (fusing) the

best practices from other oo analysis and design methods. This is the first book that describes the actual experiences of Fusion users, both inside and outside of Hewlett-Packard.

**Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development: 3rd Edition** Craig Larman 2012

Topological UML Modeling Janis Osis 2017-06-16 Topological UML Modeling: An Improved Approach for Domain Modeling and Software Development presents a specification for Topological UML® that combines the formalism of the Topological Functioning Model (TFM) mathematical topology with a specified software analysis and design method. The analysis of problem domain and design of desired solutions within software development processes has a major impact on the achieved result – developed software. While there are many tools and different techniques to create detailed specifications of the solution, the proper analysis of problem domain functioning is ignored or covered insufficiently. The design of object-oriented software has been led for many years by the Unified Modeling Language (UML®), an approved industry standard modeling notation for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system, and this comprehensive book shines new light on the many advances in the field. Presents an approach to formally define, analyze, and verify functionality of existing processes and desired processes to track incomplete or incorrect functional requirements Describes the path from functional and nonfunctional requirements specification to software design with step-by-step creation and transformation of diagrams and models with very early capturing of security requirements for software systems. Defines all modeling constructs as extensions to UML®, thus creating a new UML® profile which can be implemented in existing UML® modeling tools and toolsets

*Implementing the IBM® Rational Unified Process® and Solutions: A Guide to Improving Your Software Development Capability and Maturity* Barnes Joshua 2007

Learning UML 2.0 Russ Miles 2006-04-25 With its clear introduction to the Unified Modeling Language (UML) 2.0, this tutorial offers a solid understanding of each topic, covering foundational concepts of object-orientation and an introduction to each of the UML diagram types.