Axel Van Lamsweerde Software Requirements Engineering

Problem FramesConceptual Modeling: Foundations and ApplicationsSoftware Quality EngineeringAdvanced Use Case ModelingEngineering Dependable Software SystemsRequirements EngineeringEngineering and Managing Software RequirementsProblem FramesDesign Requirements Engineering: A Ten-Year PerspectiveScenarios, Stories, Use CasesFrom Object-Orientation to Formal MethodsSoftware Requirements & SpecificationsMastering the Requirements ProcessSocial Modeling for Requirements EngineeringVisual Models for Software RequirementsRadical Innovations of Software and Systems Engineering in the FutureRequirements Engineering FundamentalsDefining and Using Requirements Patterns for Embedded SystemsUse CasesESEC '91Designing Complex Web Information Systems: Integrating Evolutionary Process EngineeringRequirements Engineering Fundamentals, 2nd EditionRequirements EngineeringESEC '91Proceedings of the 2000 International Conference on Software EngineeringTest and Analysis of Web ServicesAdventures of an It LeaderSoftware System Reliability and SecurityProduct Focused Software Process ImprovementRequirements EngineeringInformation Systems DevelopmentSoftware RequirementsA Formal Approach to Providing Assurance to Dynamically Adaptive SoftwareReliable Software Technologies, Ada-Europe Software Requirements and DesignSoftware Engineering in Health CareRequirements Writing for System EngineeringFundamental Approaches to Software EngineeringSoftware ReuseSoftware Reuse: Methods, Techniques, and Tools

Problem Frames

This book constitutes the thoroughly refereed post-proceedings of the 9th International Workshop on Radical Innovations of Software and Systems Engineering in the Future, RISSEF 2002, held in Venice, Italy, in October 2002. The 24 revised full papers presented were carefully reviewed and selected from the 36 invited workshop presentations. The authors evaluate all major paradigms and conceptual issues in software and systems design and analysis, especially regarding their potential for modifications to cope with future needs.

Conceptual Modeling: Foundations and Applications

Since its inception in 1968, software engineering has undergone numerous changes. In the early years, software development was organized using the waterfall model, where the focus of requirements engineering was on a frozen requirements document, which formed the basis of the subsequent design and implementation process. Since then, a lot has changed: software has to be developed faster, in larger and distributed teams, for pervasive as well as large-scale applications, with more flexibility, and with ongoing maintenance and quick release cycles. What do these ongoing developments and changes imply for the future of requirements engineering and software design? Now is the time to rethink the role of requirements and design for software intensive systems in transportation, life sciences, banking, e-government and other areas. Past

assumptions need to be questioned, research and education need to be rethought. This book is based on the Design Requirements Workshop, held June 3-6, 2007, in Cleveland, OH, USA, where leading researchers met to assess the current state of affairs and define new directions. The papers included were carefully reviewed and selected to give an overview of the current state of the art as well as an outlook on probable future challenges and priorities. After a general introduction to the workshop and the related NSF-funded project, the contributions are organized in topical sections on fundamental concepts of design; evolution and the fluidity of design; quality and value-based requirements; requirements intertwining; and adapting requirements practices in different domains.

Software Quality Engineering

Subtitled ""The Work of Michael Jackson,"" this book spans the career of one of software engineering's most important figures. Half the chapters are an anthology of Jackson's past writings, exemplifying the clarity, wisdom, and wit for which he is so well known. The other half of the book is new: Jackson and his colleagues gives their latest views on requirements, specifications, design, problem frames, and programming methods. Although many people have observed that software development should be more of an engineering discipline, few have drawn from the wider engineering literature more deeply or usefully than Jackson. Because of his work, many software engineers have a better perspective on their software and the real world it is intended to serve.

Advanced Use Case Modeling

Because almost all technical systems are more or less interfaced with software these days, attacks against computer systems can cause considerable economic and physical damage. For this reason, understanding the dependability of such systems, as well as the improvement of cyber security and its development process, are amongst the most challenging and crucial issues in current computer science research. This book contains the lectures from the NATO Advanced Study Institute (ASI) Summer School entitled Engineering Dependable Software Systems, held in Marktoberdorf, Germany, in July and August 2012. This two week course for young computer scientists and mathematicians working in the field of formal software and systems was designed to give an in-depth presentation of state-of-the-art topics in the field, as well as promoting international contacts and collaboration and the teaming up of leading researchers and young scientists. The 12 lectures delivered at the school and presented here cover subjects including: model-based testing, formal modeling and verification, deductively verified software, model checking, performance analysis, integrating risk analysis, embedded systems and model checking, among others. The book will be of interest to all those whose work involves the development of large-scale, reliable and secure software systems.

Engineering Dependable Software Systems

Most IT systems fail to meet expectations. They don't meet business goals and don't support users efficiently. Why? Because the requirements didn't address the

right issues. Writing a good requirements specification doesn't take more time. This book shows how it¿s done - many times faster and many times smarter. What are the highlights? Two complete real-life requirements specifications (the traditional and the fast approach) and examples from many others. Explanations of both traditional and fast approaches, and discussions of their strengths and weaknesses in different project types (tailor-made, COTS, and product development). Real-life illustrations of all types of requirements, stakeholder analysis, cost/benefit and other techniques to ensure that business goals are met. Proven methods for dealing with difficult or complex requirements, such as specifying ease-of-use, or dealing with 200 reports that might be needed because they are in the old system. Who is it for? Everyone involved in the software supply chain, from analysts and developers to end users, will learn new techniques, benefit from requirements written by other specialists, and discover successes and failures from other companies. Software suppliers will find ideas for helping customers and writing competitive proposals. Programmers and other developers will learn how to express requirements without specifying technical details, and how to reduce risks when developing a system. Students aspiring to IT careers will learn the theory and practice of requirements engineering, and get a strong foundation for case studies and projects. Who is the author? Soren Lauesen is currently professor at the IT-University of Copenhagen. He has worked in the IT industry for 20 years and has been a professor at Copenhagen Business School for 15. He has been co-founder of three educational and two industrial development organizations. His industry projects have encompassed compilers, operating systems, process control, temporal databases, and software quality assurance. His research interests include human-computer interaction, requirements specification, object-oriented design, quality assurance, marketing and product development, and interaction between research and industry. He has a broad range of other interests ranging from biology to dancing and foreign cultures.

Requirements Engineering

Engineering and Managing Software Requirements

Provides a complete view of the architectures, problems, and solutions linked to the design and development of modern web information systems.

Problem Frames

Requirements engineering tasks have become increasingly complex. In order to ensure a high level of knowledge and competency among requirements engineers, the International Requirements Engineering Board (IREB) developed a standardized qualification called the Certified Professional for Requirements Engineering (CPRE). The certification defines the practical skills of a requirements engineer on various training levels. This book is designed for self-study and covers the curriculum for the Certified Professional for Requirements Engineering Foundation Level exam as defined by the IREB. The 2nd edition has been thoroughly revised and is aligned with the curriculum Version 2.2 of the IREB. In addition, some minor corrections to the 1st edition have been included. About IREB:The mission of the IREB is to

contribute to the standardization of further education in the fields of business analysis and requirements engineering by providing syllabi and examinations, thereby achieving a higher level of applied requirements engineering. The IRE Board is comprised of a balanced mix of independent, internationally recognized experts in the fields of economy, consulting, research, and science. The IREB is a non-profit corporation. For more information visit www.certified-re.com.

Design Requirements Engineering: A Ten-Year Perspective

Learn how to create good requirements when designing hardware and software systems. While this book emphasizes writing traditional “shall” statements, it also provides guidance on use case design and creating user stories in support of agile methodologies. The book surveys modeling techniques and various tools that support requirements collection and analysis. You’ll learn to manage requirements, including discussions of document types and digital approaches using spreadsheets, generic databases, and dedicated requirements tools. Good, clear examples are presented, many related to real-world work the author has done during his career. Requirements Writing for System Engineeringantages of different requirements approaches and implement them correctly as your needs evolve. Unlike most requirements books, Requirements Writing for System Engineering teaches writing both hardware and software requirements because many projects include both areas. To exemplify this approach, two example projects are developed throughout the book, one focusing on hardware and the other on software. This book Presents many techniques for capturing requirements. Demonstrates gap analysis to find missing requirements. Shows how to address both software and hardware, as most projects involve both. Provides extensive examples of “shall” statements, user stories, and use cases. Explains how to supplement or replace traditional requirement statements with user stories and use cases that work well in agile development environments What You Will Learn Understand the 14 techniques for capturing all requirements. Address software and hardware needs; because most projects involve both. Ensure all statements meet the 16 attributes of a good requirement. Differentiate the 19 different functional types of requirement, and the 31 non-functional types. Write requirements properly based on extensive examples of good ‘shall’ statements, user stories, and use cases. Employ modeling techniques to mitigate the imprecision of words. Audience Writing Requirements teaches you to write requirements the correct way. It is targeted at the requirements engineer who wants to improve and master his craft. This is also an excellent book from which to teach requirements engineering at the university level. Government organizations at all levels, from Federal to local levels, can use this book to ensure they begin all development projects correctly. As well, contractor companies supporting government development are also excellent audiences for this book.

Scenarios, Stories, Use Cases

Extending the scenario method beyond interface design, this important book shows developers how to design more effective systems by soliciting, analyzing, and elaborating stories from end-users Contributions from leading industry consultants and opinion-makers present a range of scenario techniques, from the light, sketchy, and agile to the careful and systematic Includes real-world case

studies from Philips, DaimlerChrysler, and Nokia, and covers systems ranging from custom software to embedded hardware-software systems

From Object-Orientation to Formal Methods

Software Requirements & Specifications

"Mastering the Requirements Process: Getting Requirements Right" sets out an industry-proven process for gathering and verifying requirements, regardless of whether you work in a traditional or agile development environment. In this sweeping update of the bestselling guide, the authors show how to discover precisely what the customer wants and needs, in the most efficient manner possible.

Mastering the Requirements Process

"The third European Software Engineering Conference follows ESEC'87 and ESEC'89. This series of conferences was set up by the European societies with the aim of providing an international forum for researchers, developersand users of software engineering technology. The need for a meeting point to discuss new results and useful experiences was clear from the large amount of high-quality European software engineering researchin recent years, stimulated, for example, through major European research programmes. The 22 papers in these proceedings were selected from 133 papers submitted from 26 different countries. They cover a fairly broad range of themes such as formal methods and practical experiences with them, special techniques for real-time systems, software evolution and re-engineering, software engineering environments, and software metrics. Invited papers by well-known experts address further important areas: perspectives on configuration management, software factories, user interfacedesign, computer security, and technology transfer."--PUBLISHER'S WEBSITE.

Social Modeling for Requirements Engineering

Visual Models for Software Requirements

Radical Innovations of Software and Systems Engineering in the Future

Following an introductory chapter that provides an exploration of key issues in requirements engineering, this book is organized in three parts. It presents surveys of requirements engineering process research along with critical assessments of existing models, frameworks and techniques. It also addresses key areas in requirements engineering.

Requirements Engineering Fundamentals

After Ole-Johan's retirement at the beginning of the new millennium, some of us had thought and talked about making a "Festschrift" in his honor. When Donald Knuth took the initiative by sending us the ?rst contribution, the p- cess began to roll! In early 2002 an editing group was formed, including Kristen Nygaard, who had known Ole-Johan since their student days, and with whom he had developed the Simula language. Then we invited a number of prominent researchers familiar with Ole-Johan to submit contributions for a book hon- ing Ole-Johan on the occasion of his 70th birthday. Invitees included several members of the IFIP 2. 3 working group, a forum that Ole-Johan treasured and enjoyedparticipating in throughouthis career. In spite of the shortdeadline, the response to the invitations was overwhelmingly positive. The original idea was to complete the book rather quickly to make it a gift he could read and enjoy, because by then he had had cancer for three years, and his health was gradually deteriorating. Kristen had been regularly visiting Ole-Johan,who was in the hospitalat that time, and they were workingon their Turing award speech. Ole-Johan was grati?ed to hear about the contributions to this book, but modestly expressed the feeling that there was no special need to undertake a book project on his behalf. Peacefully accepting his destiny, Ole- Johan died on June 29, 2002.

Defining and Using Requirements Patterns for Embedded Systems

Written for those who want to develop their knowledge of requirements engineering process, whether practitioners or students. Using the latest research and driven by practical experience from industry, Requirements Engineering gives useful hints to practitioners on how to write and structure requirements. It explains the importance of Systems Engineering and the creation of effective solutions to problems. It describes the underlying representations used in system modeling and introduces the UML2, and considers the relationship between requirements and modeling. Covering a generic multi-layer requirements process, the book discusses the key elements of effective requirements management. The latest version of DOORS (Version 7) - a software tool which serves as an enabler of a requirements management process - is also introduced to the reader here. Additional material and links are available at: http://www.requirementsengineering.info

Use Cases

This book constitutes the refereed proceedings of the 8th International Conference on Software Reuse, ICSR-8, held in Madrid, Spain in July 2004. The 28 revised full papers presented were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on software variability: requirements; testing reusable software; feature modeling; aspect-oriented software development; component and service development; code level reuse; libraries, classification, and retrieval; model-based approaches; transformation and generation; and requirements.

ESEC '91

Essential comprehensive coverage of the fundamentals of requirements

engineering Requirements engineering (RE) deals with the variety of prerequisites that must be met by a software system within an organization in order for that system to produce stellar results. With that explanation in mind, this must-have book presents a disciplined approach to the engineering of high-quality requirements. Serving as a helpful introduction to the fundamental concepts and principles of requirements engineering, this guide offers a comprehensive review of the aim, scope, and role of requirements engineering as well as best practices and flaws to avoid. Shares state-of-the-art techniques for domain analysis, requirements elicitation, risk analysis, conflict management, and more Features in-depth treatment of system modeling in the specific context of engineering requirements Presents various forms of reasoning about models for requirements quality assurance Discusses the transitions from requirements to software specifications to software architecture In addition, case studies are included that complement the many examples provided in the book in order to show you how the described method and techniques are applied in practical situations.

Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering

Requirements Engineering Fundamentals, 2nd Edition

Apply best practices for capturing, analyzing, and implementing software requirements through visual models—and deliver better results for your business. The authors—experts in eliciting and visualizing requirements—walk you through a simple but comprehensive language of visual models that has been used on hundreds of real-world, large-scale projects. Build your fluency with core concepts—and gain essential, scenario-based context and implementation advice—as you progress through each chapter. Transcend the limitations of text-based requirements data using visual models that more rigorously identify, capture, and validate requirements Get real-world guidance on best ways to use visual models—how and when, and ways to combine them for best project outcomes Practice the book’s concepts as you work through chapters Change your focus from writing a good requirement to ensuring a complete system

Requirements Engineering

"Information security covers the protection of information against unauthorized disclosure, transfer, modification, and destruction, whether accidentally or intentionally. Quality of life in general and of individual citizens, and the effectiveness of the economy critically depends on our ability to build software in a transparent and efficient way. Furthermore, we must be able to enhance the software development process systematically in order to ensure software's safety and security. This, in turn, requires very high software reliability, i.e., an extremely high confidence in the ability of the software to perform flawlessly. Foundations of software technology provide models that enable us to capture application domains and their requirements, but also to understand the structure and working of software systems and software architectures. Based on these foundations tools allow to prove and ensure the correctness of software's functioning. New

developments must pay due diligence to the importance of security-related aspects, and align current methods and techniques to information security, integrity, and system reliability. The articles in this book describe the state-of-the-art ideas on how to meet these challenges in software engineering."

ESEC '91

With a spice of wit and illuminating illustration, this collection of 75 short pieces deals with topics in the field of software requirements analysis, specifications and design. The author emphasizes the need to structure and analyze problems, not just specify a solution.

Proceedings of the 2000 International Conference on Software Engineering

This volume is the result of the 11th International Conference on Information Systems Development: Methods and Tools, Theory and Practice, held in Riga, Latvia, September 12-14, 2002. The purpose of this conference was to address issues facing academia and industry when specifying, developing, managing, reengineering and improving information systems. This volume should be a useful reference for anyone in the fields of general management, systems and control theory, software engineering and operation systems.

Test and Analysis of Web Services

Requirements engineering tasks have become increasingly complex. In order to ensure a high level of knowledge and competency among requirements engineers, the International Requirements Engineering Board (IREB) developed a standardized qualification called the Certified Professional for Requirements Engineering (CPRE). The certification defines the practical skills of a requirements engineer on various training levels. This book is designed for self-study and covers the curriculum for the Certified Professional for Requirements Engineering Foundation Level exam as defined by the IREB. **The 2nd edition** has been thoroughly revised and is aligned with the curriculum Version 2.2 of the IREB. In addition, some minor corrections to the 1st edition have been included. **About IREB:** The mission of the IREB is to contribute to the standardization of further education in the fields of business analysis and requirements engineering by providing syllabi and examinations, thereby achieving a higher level of applied requirements engineering. The IRE Board is comprised of a balanced mix of independent, internationally recognized experts in the fields of economy, consulting, research, and science. The IREB is a non-profit corporation. For more information visit www.certified-re.com

Adventures of an It Leader

This Festschrift volume, published in honor of John Mylopoulos on the occasion of his retirement from the University of Toronto, contains 25 high-quality papers, written by leading scientists in the field of conceptual modeling. The volume has been divided into six sections. The first section focuses on the foundations of conceptual modeling and contains material on ontologies and knowledge

representation. The four sections on software and requirements engineering, information systems, information integration, and web and services, represent the chief current application domains of conceptual modeling. Finally, the section on implementations concentrates on projects that build tools to support conceptual modeling. With its in-depth coverage of diverse topics, this book could be a useful companion to a course on conceptual modeling.

Software System Reliability and Security

Product Focused Software Process Improvement

This book is a must-have for all IT professionals facing software development problems on a daily basis. If you are a systems analyst or requirements engineer it will provide an essential, practical guide from the task of identifying the problem to making the descriptions needed to resolve it. It will help you: decompose complex problems into simpler subproblems and see how the subproblems fit together; and build up a repertoire of simple, clear and easily applicable problem classes which you can access and reuse, drawing on the experience associated with each class. Features: numerous real-world example problems are analyzed, giving you insight into how to recognize and structure your own problems in practice; a mixture of large and small problems is presented, showing the stripped down essence of problem classes and discussing different aspects of each problem; problem frames are independent of any particular development method, so they can be easily applied in your own situation; and appendices summarizing the descriptive languages and notations; plus a glossary of terminology.

Requirements Engineering

This book describes a modeling approach (called the i* framework) that conceives of software-based information systems as being situated in environments in which social actors relate to each other in terms of goals to be achieved, tasks to be performed, and resources to be furnished.

Information Systems Development

This book is a must-have for all IT professionals facing software development problems on a daily basis. If you are a systems analyst or requirements engineer it will provide an essential, practical guide from the task of identifying the problem to making the descriptions needed to resolve it. It will help you: decompose complex problems into simpler subproblems and see how the subproblems fit together; and build up a repertoire of simple, clear and easily applicable problem classes which you can access and reuse, drawing on the experience associated with each class. Features: numerous real-world example problems are analyzed, giving you insight into how to recognize and structure your own problems in practice; a mixture of large and small problems is presented, showing the stripped down essence of problem classes and discussing different aspects of each problem; problem frames are independent of any particular development method, so they can be easily applied in your own situation; and appendices summarizing the descriptive

languages and notations; plus a glossary of terminology.

Software Requirements

The one resource needed to create reliable software This text offers a comprehensive and integrated approach tosoftware quality engineering. By following the author's clearguidance, readers learn how to master the techniques to producehigh-quality, reliable software, regardless of the softwaresystem's level of complexity. The first part of the publication introduces major topics insoftware quality engineering and presents quality planning as anintegral part of the process. Providing readers with a solidfoundation in key concepts and practices, the book moves on tooffer in-depth coverage of software testing as a primary means toensure software quality; alternatives for quality assurance,including defect prevention, process improvement, inspection,formal verification, fault tolerance, safety assurance, and damagecontrol; and measurement and analysis to close the feedback loopfor quality assessment and quantifiable improvement. The text's approach and style evolved from the author's hands-onexperience in the classroom. All the pedagogical tools needed tofacilitate quick learning are provided: * Figures and tables that clarify concepts and provide quick topicsummaries * Examples that illustrate how theory is applied in real-worldsituations * Comprehensive bibliography that leads to in-depth discussion ofspecialized topics * Problem sets at the end of each chapter that test readers'knowledge This is a superior textbook for software engineering, computerscience, information systems, and electrical engineering students,and a dependable reference for software and computer professionalsand engineers.

A Formal Approach to Providing Assurance to Dynamically Adaptive Software

"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

Reliable Software Technologies, Ada-Europe

Software Requirements and Design

This book describes how to gather and define software requirements using a process based on use cases. It shows systems analysts and designers how use cases can provide solutions to the most challenging requirements issues, resulting in effective, quality systems that meet the needs of users. Use Cases, Second Edition: Requirements in Context describes a three-step method for establishing requirements—an iterative process that produces increasingly refined requirements. Drawing on their extensive, real-world experience, the authors offer a wealth of advice on use-case driven lifecycles, planning for change, and keeping on track. In addition, they include numerous detailed examples to illustrate practical applications. This second edition incorporates the many advancements in use case methodology that have occurred over the past few years. Specifically, this new edition features major changes to the methodology's iterations, and the section on management reflects the faster-paced, more "chaordic" software lifecycles prominent today. In addition, the authors have included a new chapter on use case traceability issues and have revised the appendixes to show more clearly how use cases evolve. The book opens with a brief introduction to use cases and the Unified Modeling Language (UML). It explains how use cases reduce the incidence of duplicate and inconsistent requirements, and how they facilitate the documentation process and communication among stakeholders. The book shows you how to: Describe the context of relationships and interactions between actors and applications using use case diagrams and scenarios Specify functional and nonfunctional requirements Create the candidate use case list Break out detailed use cases and add detail to use case diagrams Add triggers, preconditions, basic course of events, and exceptions to use cases Manage the iterative/incremental use case driven project lifecycle Trace back to use cases, nonfunctionals, and business rules Avoid classic mistakes and pitfalls The book also highlights numerous currently available tools, including use case name filters, the context matrix, user interface requirements, and the authors' own "hierarchy killer."

Software Engineering in Health Care

This book constitutes revised selected papers from the jointly held conferences FHIES 2014, 4th International Symposium on Foundations of Health Information Engineering and Systems, and SEHC 2014, 6th International Workshop on Software Engineering in Health Care. The meeting took place in Washington, DC, USA, in July 2014. The 16 papers presented in this volume were carefully reviewed and

selected from 23 submissions. They deal with security aspects of health information systems; medical devices in cyberphysical systems; the process of providing healthcare and of monitoring patients; and patient safety and the assurance of medical systems.

Requirements Writing for System Engineering

Becoming an effective IT manager presents a host of challenges--from anticipating emerging technology to managing relationships with vendors, employees, and other managers. A good IT manager must also be a strong business leader. This book invites you to accompany new CIO Jim Barton to better understand the role of IT in your organization. You'll see Jim struggle through a challenging first year, handling (and fumbling) situations that, although fictional, are based on true events. You can read this book from beginning to end, or treat is as a series of cases. You can also skip around to address your most pressing needs. For example, need to learn about crisis management and security? Read chapters 10-12. You can formulate your own responses to a CIO's obstacles by reading the authors' regular "Reflection" questions. You'll turn to this book many times as you face IT-related issues in your own career.

Fundamental Approaches to Software Engineering

This volume presents the proceedings of the Third European Software Engineering Conference. Themes include formal methods and practical experiences with them, special techniques for real-time systems, software evolution and re-engineering, software engineering environments, and software metrics.

Software Reuse

Requirements engineering is the process of eliciting individual stakeholder requirements and needs and developing them into detailed, agreed requirements documented and specified in such a way that they can serve as the basis for all other system development activities. In this textbook, Klaus Pohl provides a comprehensive and well-structured introduction to the fundamentals, principles, and techniques of requirements engineering. He presents approved techniques for eliciting, negotiating and documenting as well as validating, and managing requirements for software-intensive systems. The various aspects of the process and the techniques are illustrated using numerous examples based on his extensive teaching experience and his work in industrial collaborations. His presentation aims at professionals, students, and lecturers in systems and software engineering or business applications development. Professionals such as project managers, software architects, systems analysts, and software engineers will benefit in their daily work from the didactically well-presented combination of validated procedures and industrial experience. Students and lecturers will appreciate the comprehensive description of sound fundamentals, principles, and techniques, which is completed by a huge commented list of references for further reading. Lecturers will find additional teaching material on the book’s website, www.requirements-book.com.

Software Reuse: Methods, Techniques, and Tools

The authors have here put together the first reference on all aspects of testing and validating service-oriented architectures. With contributions by leading academic and industrial research groups it offers detailed guidelines for the actual validation process. Readers will find a comprehensive survey of state-of-the-art approaches as well as techniques and tools to improve the quality of service-oriented applications. It also includes references and scenarios for future research and development.