

Axel Van Lamsweerde Requirements Engineering

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.

This volume contains lectures presented at the 21st International Summer School on Engineering Theories of Software Construction (Marktoberdorf, Germany July/August 2000). Eleven contributions from professionals in industry and academia trace the path from the scientific foundations of programming theory through the development of toolsets and methods and on to practical application by working engineers. A sampling of topics includes unifying theories for logic programming, performance modeling using probabilistic process algebra, and extended static checking. The volume is not indexed. Annotation copyrighted by Book News, Inc., Portland, OR.

We can now say that it is really a big pleasure for us to welcome all of you to the proceedings of CAiSE 2005 which was held in Porto.

Discusses the technological developments, main issues, challenges, opportunities, and trends impacting every part of small to medium sized enterprises.

"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."

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.

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, this book gives useful hints to practitioners on how to write and structure requirements. - Explains the importance of Systems Engineering and the creation of effective solutions to problems - Describes the underlying representations used in system modeling - data flow diagrams; statecharts; object-oriented approaches - Covers a generic multi-layer requirements process - Discusses the key elements of effective requirements management - Includes a chapter written by one of the developers of rich traceability - Introduces an overview of DOORS - a software tool which serves as an enabler of a requirements management process Additional material and links are available at: <http://www.requirementsengineering.info> "In recent years we have been finding ourselves with a shortage of engineers with good competence in requirements engineering. Perhaps this is in part because requirements management tool vendors have persuaded management that a glitzy tool will solve their requirements engineering problems. Of course, the tools only make it possible for engineers who understand requirements engineering to do a better job. This book goes a long way towards building a foundational set of skills in requirements engineering, so that today's powerful tools can be used sensibly. Of particular value is a recognition of the place software requirements have within the system context, and of ways for dealing with that sensitive connection. This is an important book. I think its particular value in industry will be to bring the requirements engineers and their internal customers to a practical common understanding of what can and should be achieved." (Byron Purves, Technical Fellow, The Boeing Company)

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.

NLDB 2005, the 10th International Conference on Applications of Natural Language to Information Systems, was held on June 15–17, 2005 at the University of Alicante, Spain. Since the first NLDB conference in 1995 the main goal has been to provide a forum to discuss and disseminate research on the integration of natural language resources in information system engineering. The development and convergence of computing, telecommunications and information systems has already led to a revolution in the way that we work, communicate with each other, buy goods and use services, and even in the way that we entertain and educate ourselves. The revolution continues, and one of its results is that large volumes of information will increasingly be held in a form which is more natural for users than the data presentation formats typical of computer systems of the past. Natural language processing (NLP) is crucial in solving these problems, and language technologies will make an indispensable contribution to the success of information systems. We hope that NLDB 2005 was a modest contribution to this goal. NLDB 2005 contributed to advancing the goals and the high international standing of these conferences, largely due to its Program Committee, composed of renowned researchers in the field of natural language processing and information system engineering. Papers were reviewed by three reviewers from the Program Committee. This clearly contributed to the significant number of papers submitted (95). Twenty-nine were accepted as regular papers, while 18 were accepted as short papers.

This book provides a coherent methodology for Model-Driven Requirements Engineering which stresses the systematic treatment of requirements within the realm of modelling and model

transformations. The underlying basic assumption is that detailed requirements models are used as first-class artefacts playing a direct role in constructing software. To this end, the book presents the Requirements Specification Language (RSL) that allows precision and formality, which eventually permits automation of the process of turning requirements into a working system by applying model transformations and code generation to RSL. The book is structured in eight chapters. The first two chapters present the main concepts and give an introduction to requirements modelling in RSL. The next two chapters concentrate on presenting RSL in a formal way, suitable for automated processing. Subsequently, chapters 5 and 6 concentrate on model transformations with the emphasis on those involving RSL and UML. Finally, chapters 7 and 8 provide a summary in the form of a systematic methodology with a comprehensive case study. Presenting technical details of requirements modelling and model transformations for requirements, this book is of interest to researchers, graduate students and advanced practitioners from industry. While researchers will benefit from the latest results and possible research directions in MDRE, students and practitioners can exploit the presented information and practical techniques in several areas, including requirements engineering, architectural design, software language construction and model transformation. Together with a tool suite available online, the book supplies the reader with what it promises: the means to get from requirements to code "in a snap".

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.

This book is dedicated to the memory of Ole-Johan Dahl who passed away in June 2002 at the age of 70, shortly after he had received, together with his colleague Kristen Nygaard, the ACM Alan M. Turing Award: "For ideas fundamental to the emergence of object-oriented programming, through their design of the programming languages Simula I and Simula 67." This Festschrift opens with a short biography and a bibliography recollecting Ole-Johan Dahl's life and work, as well as a paper he wrote entitled: "The Birth of Object-Orientation: the Simula Languages." The main part of the book consists of 14 scientific articles written by leading scientists who worked with Ole-Johan Dahl as students or colleagues. In accordance with the scope of Ole-Johan Dahl's work and the book's title, the articles are centered around object-orientation and formal methods.

This volume focuses on the education of researchers, teachers, students and practitioners. As usual in engineering, a study and application of the relevant branches of mathematics is crucial both in education and practice.

The more our society relies on electronic forms of communication, the more the security of these communication networks is essential for its well-functioning. Topics in this volume include developments in: security protocols; secure software engineering; and mobile agent security.

Trustworthiness is a key success factor in the acceptance and adoption of cyber-physical systems. The author first discusses various existing definitions of trust and trustworthiness and extends them to cyber-physical systems. A comprehensive framework is proposed, including methods that cover all phases of development: requirements engineering, system design, trustworthiness evaluation, run-time maintenance, and evidence-based assurance. To support a smooth integration of the methods into development projects, these methods are provided in the form of so-called capability patterns. A running example from the ambient assisted living domain is used to demonstrate the application of the methods. About the Author: Nazila Gol Mohammadi is currently working as an associate researcher at paluno – The Ruhr Institute for Software Technology in Essen, Germany. Her research interests include software engineering, requirements engineering, digitalization, cloud computing, cyber-physical systems, and trustworthiness of software systems.

The Third International Conference on Product Focused Software Process Improvement (PROFES 2001) continued the success of the PROFES'99 and PROFES 2000 conferences. PROFES 2001 was organized in Kaiserslautern, Germany, September 10-13, 2001. The PROFES conference has its roots in the PROFES Esprit project (<http://www.ele.vtt.fi/profes/>), but it quickly evolved into a full fledged general purpose conference in 1999 and since then it has gained wide spread international popularity. As in previous years, the main theme of PROFES 2001 was professional software process improvement (SPI) motivated by product and service quality needs. SPI is facilitated by software process assessment, software measurement, process modeling, and technology transfer and has become a practical tool for quality software engineering and management. The conference addresses both the solutions found in practice as well as relevant research results from academia. The purpose of the conference is to bring to light the most recent findings and results in the area and to stimulate discussion between the researchers, experienced professionals, and technology providers for SPI.

"This book is not only of practical value. It's also a lot of fun to read." Michael Jackson, The Open University. Do you need to know how to create good requirements? Discovering Requirements offers a set of simple, robust, and effective cognitive tools for building requirements. Using worked examples throughout the text, it shows you how to develop an understanding of any problem, leading to questions such as: What are you trying to achieve? Who is involved, and how? What do those people want? Do they agree? How do you envisage this working? What could go wrong? Why are you making these decisions? What are you assuming? The established author team of Ian Alexander and Ljerka Beus-Dukic answer these and related questions, using a set of complementary techniques, including stakeholder analysis, goal modelling, context modelling, storytelling and scenario modelling, identifying risks and threats, describing rationales, defining terms in a project dictionary, and prioritizing. This easy to read guide is full of carefully-checked tips and tricks. Illustrated with worked examples, checklists,

summaries, keywords and exercises, this book will encourage you to move closer to the real problems you're trying to solve. Guest boxes from other experts give you additional hints for your projects. Invaluable for anyone specifying requirements including IT practitioners, engineers, developers, business analysts, test engineers, configuration managers, quality engineers and project managers. A practical sourcebook for lecturers as well as students studying software engineering who want to learn about requirements work in industry. Once you've read this book you will be ready to create good requirements!

This book is the result of the 11 th 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. Recently many new concepts and approaches have emerged in the Information Systems Development (ISD) field. Various theories, methodologies, methods and tools available to system developers also created new problems, such as choosing the most effective approach for a specific task, or solving problems of advanced technology integration into information systems. This conference provides a meeting place for ISD researchers and practitioners from Eastern and Western Europe as well as from other parts of the world. Main objectives of this conference are to share scientific knowledge and interests and to establish strong professional ties among the participants. The 11th International Conference on Information Systems Development (ISD'02) continues the tradition started with the first Polish-Scandinavian Seminar on Current Trends in Information Systems Development Methodologies, held in Gdansk, Poland in 1988. Through the years this Seminar has evolved into the International Conference on Information Systems Development. ISD'02 is the first ISD conference held in Eastern Europe, namely, in Latvia, one of the three Baltic countries.

This book comprises a set of papers selected from those presented at the fifth « International Conference on Enterprise Information Systems », (ICEIS'2003) held in Angers, France, from 23 to 26 April 2003. The conference was organised by École Supérieure d'Électronique de l'Ouest (ESEO) of Angers, France and the Escola Superior de Tecnologia of Setúbal, Portugal. Since its first edition in 1999, ICEIS focuses on real world applications and aims at bringing together researchers, engineers and practitioners interested in the advances and business applications of information systems. As in previous years, ICEIS'2003 held four simultaneous tracks covering different aspects of enterprise computing: Databases and Information Systems Integration, Artificial Intelligence and Decision Support Systems, Information Systems Analysis and Specification and Software Agents and Internet Computing. Although ICEIS'2003 received 546 paper submissions from over 50 countries, only 80 were accepted as full papers and presented in 30-minutes oral presentations. With an acceptance rate of 15%, these numbers demonstrate the intention of preserving a high quality forum for future editions of this conference. From the articles accepted as long papers for the conference, only 32 were selected for inclusion in this book. Additional keynote lectures, tutorials and industrial sessions were also held during ICEIS'2003, and, for the first time this year, the 1st Doctoral Consortium on Enterprise Information Systems gave PhD students an opportunity to present their work to an international audience of experts in the field of information systems.

This book constitutes the strictly refereed post-workshop proceedings of the International Workshop on Requirements Targeting Software and Systems Engineering, RTSE '97, held in Bernried, Germany in October 1997. The 15 revised full papers presented in the book were carefully revised and reviewed for inclusion in the book. Among the authors are internationally leading researchers. The book is divided in sections on foundations of software engineering, methodology, evaluation and case studies, and tool support and prototyping.

This book constitutes the refereed proceedings of the Third International Conference on Generative and Component-Based Software Engineering, GCSE 2001, held in Erfurt, Germany, in September 2001. The 14 revised full papers presented together with one invited paper were carefully reviewed and selected from 43 submissions. The papers are organized in topical sections on software product lines, aspects, generic and generative approaches, and components and architectures.

This three-volume collection, titled Enterprise Information Systems: Concepts, Methodologies, Tools and Applications, provides a complete assessment of the latest developments in enterprise information systems research, including development, design, and emerging methodologies. Experts in the field cover all aspects of enterprise resource planning (ERP), e-commerce, and organizational, social and technological implications of enterprise information systems.

A revolutionary concept-based approach to thinking about, designing, and interacting with software As our dependence on technology increases, the design of software matters more than ever before. Why then is so much software flawed? Why hasn't there been a systematic and scalable way to create software that is easy to use, robust, and secure? Examining these issues in depth, The Essence of Software introduces a theory of software design that gives new answers to old questions. Daniel Jackson explains that a software system should be viewed as a collection of interacting concepts, breaking the functionality into manageable parts and providing a new framework for thinking about design. Through this radical and original perspective, Jackson lays out a practical and coherent path, accessible to anyone—from strategist and marketer to UX designer, architect, or programmer—for making software that is empowering, dependable, and a delight to use. Jackson explores every aspect of concepts—what they are and aren't, how to identify them, how to define them, and more—and offers prescriptive principles and practical tips that can be applied cost-effectively in a wide range of domains. He applies these ideas to contemporary software designs, drawing examples from leading software manufacturers such as Adobe, Apple, Dropbox, Facebook, Google, Microsoft, Twitter, and others. Jackson shows how concepts let designers preserve and reuse design knowledge, rather than starting from scratch in every project. An argument against the status quo and a guide to improvement for both working designers and novices to the field, The Essence of Software brings a fresh approach to software and its creation.

Recommender systems use information filtering to predict user preferences. They are becoming a vital part of e-business and are used in a wide variety of industries, ranging from entertainment and social networking to information technology, tourism, education, agriculture, healthcare, manufacturing, and retail. Recommender Systems: Algorithms and Applications dives into the theoretical underpinnings of these systems and looks at how this theory is applied and implemented in actual systems. The book examines several classes of recommendation algorithms, including Machine learning algorithms Community detection algorithms Filtering algorithms Various efficient and robust product recommender systems using machine learning algorithms are helpful in filtering and exploring unseen data by users for better prediction and extrapolation of decisions. These are providing a wider range of solutions to such challenges as

imbalanced data set problems, cold-start problems, and long tail problems. This book also looks at fundamental ontological positions that form the foundations of recommender systems and explain why certain recommendations are predicted over others. Techniques and approaches for developing recommender systems are also investigated. These can help with implementing algorithms as systems and include A latent-factor technique for model-based filtering systems Collaborative filtering approaches Content-based approaches Finally, this book examines actual systems for social networking, recommending consumer products, and predicting risk in software engineering projects.

"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

ETAPS 2002 was the 7th instance of the European Joint Conferences on Theory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised 5 conferences (FOSSACS, FASE, ESOP, CC, TACAS), 13 satellite workshops (ACL2, AGT, CMCS, COCV, DCC, INT, LDFA, SC, SFEDL, SLAP, SPIN, TPTS, and VISS), 8 invited lectures (not including those specific to the satellite events), and several tutorials. The events that comprise ETAPS address various aspects of the system development process, including specification, design, implementation, analysis, and improvement. The languages, methodologies, and tools which support these activities are all well within its scope. Different blends of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and soundly-based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive.

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 Marktobendorf, 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.

Requirements Engineering From System Goals to UML Models to Software Specifications John Wiley & Sons Incorporated

The size, complexity, and integration level of software systems is increasing constantly. Companies in all domains identify that software defines the competitive edge of their products. These developments require us to constantly search for new approaches to increase the productivity and quality of our software development and to decrease the cost of software maintenance. Generative and component-based technologies hold considerable promise with respect to achieving these goals. GCSE 2001 constituted another important step forward and provided a platform for academic and industrial researchers to exchange ideas. These proceedings represent the third conference on generative and component-based software engineering. The conference originated as a special track on generative programming from the Smalltalk and Java in Industry and Education Conference (STJA), organized by the working group "Generative and Component-Based Software Engineering" of the "Gesellschaft für Informatik" FG 2.1.9 "Object-Oriented Software Engineering." However, the conference has evolved substantially since then, with its own, independent stature, invited speakers, and, most importantly, a stable and growing community. This year's conference attracted 43 submissions from all over the world, indicating the broad, international interest in the research field. Based on careful review by the program committee, 14 papers were selected for presentation. I would like to thank the members of the program committee, all renowned experts, for their dedication in preparing thorough reviews of the submissions.

In the past ten years or so, software architecture has emerged as a central notion in the development of complex software systems. Software architecture is now accepted in the

software engineering research and development community as a manageable and meaningful abstraction of the system under development and is applied throughout the software development life cycle, from requirements analysis and validation, to design and down to code and execution level. This book presents the tutorial lectures given by leading authorities at the Third International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2003, held in Bertinoro, Italy, in September 2003. The book is ideally suited for advanced courses on software architecture as well as for ongoing education of software engineers using formal methods in their day-to-day professional work.

For more and more systems, software has moved from a peripheral to a central role, replacing mechanical parts and hardware and giving the product a competitive edge. Consequences of this trend are an increase in: the size of software systems, the variability in software artifacts, and the importance of software in achieving the system-level properties. Software architecture provides the necessary abstractions for managing the resulting complexity. We here introduce the Third Working IEEE/IFIP Conference on Software Architecture, WICSA3. That it is already the third such conference is in itself a clear indication that software architecture continues to be an important topic in industrial software development and in software engineering research. However, becoming an established field does not mean that software architecture provides less opportunity for innovation and new directions. On the contrary, one can identify a number of interesting trends within software architecture research. The first trend is that the role of the software architecture in all phases of software development is more explicitly recognized. Whereas initially software architecture was primarily associated with the architecture design phase, we now see that the software architecture is treated explicitly during development, product derivation in software product lines, at run-time, and during system evolution. Software architecture as an artifact has been decoupled from a particular lifecycle phase.

SYSMOD is an MBSE toolbox for pragmatic modeling of systems. It is well-suited to be used with SysML. The book provides a set of methods with roles and outputs. Concrete guidances and examples show how to apply the methods with SysML. * Requirements modeling * System Context * Use Cases * Functional, Physical, Logical and Product Architectures * Guidances how to create a SysML model * Full-fledged SysML example * Complete definition of a profile for SYSMOD This book is also available as an eBook at leanpub.com/sysmod.

Modelling of information is necessary in developing information systems. Information is acquired from many sources, by using various methods and tools. It must be recognized, conceptualized, and conceptually organized efficiently so that users can easily understand and use it. Modelling is needed to understand, explain, organize, predict, and reason on information. It also helps to master the role and functions of components of information systems. Modelling can be performed with many different purposes in mind, at different levels, and by using different notions and different background theories. It can be made by emphasizing users' conceptual understanding of information on a domain level, on an algorithmic level, or on representation levels. On each level, the objects and structures used on them are different, and different rules govern the behavior on them. Therefore the notions, rules, theories, languages, and methods for modelling on different levels are also different. It will be useful if we can develop theories and methodologies for modelling, to be used in different situations, because databases, knowledge bases, and repositories in knowledge management systems, developed on the basis of models and used to technically store information, are growing day by day. In this publication, the interest is focused on modelling of information, and one of the central topics is modelling of time. Scientific and technical papers of high quality are brought together in this book.

This volume contains the proceedings of the 17th IFIP TC6/WG6.1 International Conference on Testing of Communicating Systems (TestCom 2005). The conference was held at Concordia University, Montreal, Canada, from May 31 to June 2, 2005. TestCom 2005 was organized by Concordia University and was sponsored by IFIP.

This book constitutes the thoroughly refereed post-proceedings of the International Software Process Workshop, SPW 2005, held in Beijing, China in May 2005. The 30 papers presented here, together with 11 keynote addresses are organized in topical sections on process content, process tools and metrics, process management, process representation and analysis, as well as experience reports.

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.

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.

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

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 Engineering advantages 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.

The book presents both the current state of the art in requirements engineering and a systematic method for engineering high-quality requirements, broken down into four parts. The first part introduces fundamental concepts and principles including the aim and scope of requirements engineering, the products and processes involved, requirements qualities to aim at and flaws to avoid, and the critical role of requirements engineering in system and software engineering. The second part of the book is devoted to system modeling in the specific context of engineering requirements. It presents a multi-view modeling framework that integrates complementary techniques for modeling the system-as-is and the system-to-be. The third part of the book reviews goal-based reasoning techniques to support the various steps of the KAOS method. The fourth part of the book goes beyond requirements engineering to discuss the mapping from goal-oriented requirements to software specifications and to software architecture. Online software will accompany the book and will add value to both classroom and self-study by enabling students to build models and specifications involved in the book's exercises and case studies, helping them to discover the latest RE technology solutions. Instructor resources such as slides, figures and handouts are available from an accompanying website.

[Copyright: f1876fe837cbaf11f982228fcb7f6e0a](#)