Software Product Line Engineering Klaus Pohl 2005-08-03 Software product line engineering has proven to be the methodology for developing a diversity of software products, especially in large enterprises and in software-intensive industries, such as automotive, aeronautics, telecommunications, and industrial control. Klaus Pohl and his co-authors present a framework for software product line engineering which they have developed based on their academic as well as industrial experience gained in projects over the last eight years. They derive their framework from the latest academic research and describes the underlying guidelines and techniques. They explicitly point out the key differences of software product line engineering compared to traditional single software system development, as the need for two distinct development processes for domain and application engineering respectively, or the need to define and manage variability.

Thinking-Howard Eizer 2009-01-14 Thinking: A Guide to Systems Engineering Problem-Solving focuses upon articulating ways of thinking in today's world of systems and engineering. It also explores how the masters made the advances they made, hundreds of years ago. Taken together, these considerations represent novel problem-solving and new pathways to answers for modern times. Special techniques are illustrated using examples from systems engineering, automotive, corporate, social networking, and innovation as well as some famous classical examples from the history of thinking.

Seminal Contributions to Information Systems Engineering Janis Bubenko 2013-06-15 In 2013, the International Conference on Advanced Information Systems Engineering (CAiSE) turns 20, initially launched in 1993, for all these years the conference has provided a broad forum for researchers working in the area of Information Systems Engineering. To reflect on the work done so far and to examine prospects for future work, the CAiSE Steering Committee decided to present a selection of seminal papers published for the conference during these years and to ask their authors, all prominent researchers in the field, to comment on their work and how it has developed over the years. The scope of the papers selected covers a broad range of topics related to modeling and designing information systems, collecting and managing requirements, and with special attention to how information systems are engineered through their final development and deployment as such for both complex systems and software-as-a-service solutions. This book is not only a historical analysis of research in information systems engineering, but also a lively overview of the fascinating social network analysis of the research community. Additionally, many inspiring ideas for future research and new perspectives in this area are sparked by the intriguing comments of the renowned authors.

Requirements Engineering for Software and Systems—Philip A. Laplante 2017-10-24 Solid requirements engineering has increasingly been recognized as the key to providing and, on-time and ultimately successful deployment of software systems. This textbook provides a comprehensive treatment of the theoretical and practical aspects of requirements engineering. The book discusses the connection between requirements engineering and other software engineering disciplines such as software architecture, project management, and testing. It brings into play a variety of formal methods, construction tools, and modern requirements for writing techniques to be useful to the practicing engineer. This book will provide a practical and comprehensive guide for software engineers and system analysts involved in requirements engineering activities. Advanced exercises are available as a research assignment or independent study and are denoted by an asterisk. Varied examples illustrate problems throughout the book and include scenarios from information systems engineering. Solutions to selected exercises have been included. The book introduces the reader to the many requirements engineering tools currently available, and provides a description of their capabilities. Solutions to selected exercises have been included. The book introduces the reader to the many requirements engineering tools currently available, and provides a description of their capabilities.

Learning and Teaching Systems Engineering—George Koelsch 2016-10-20 Learn how to create good requirements in designing hardware and software systems. While this book emphasizes writing traditional “shell” statements, it also provides guidance on use case design and creating user stories in support of agile development. It includes a collection of sample requirements and varied techniques 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 includes 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 and emphasizes the importance of good systems engineering practices. It includes a comprehensive reference section on teaching systems engineering, including a chapter written by the developers of the textbook. It also includes a complete list of all tools and resources that can be found online.
The Principles of Health Care Administration - Joseph Truex
2020-04-15 Textbook to assist candidates studying for the national examination to become a nursing home administrator.

Managed Software Evolution - Ralph Ressemer 2019-01-01 This open access book presents the outcomes of the "Design for Future - Managed Software Evolution" project program 103, which was launched by the German Research Foundation ("Deutsche Forschungsgemeinschaft (DFG)") to develop new approaches to software engineering with a specific focus on long-lived software systems. The different lifecycles of software and hardware platforms lead to interoperability problems in such systems. Instead of separating the development, adaptation and evolution of software and its platforms, as well as aspects like operation, monitoring and maintenance, they should all be integrated into one overarching process. Accordingly, the book is split into three major parts, the first of which includes an introduction to the nature of software evolution, followed by an overview of the specific challenges and a general introduction to the case studies used in the project. The second part of the book consists of the main chapters on knowledge carrying software, and cover tacit knowledge in software evolution, continuous design decision support, model-based round-trip engineering for software product lines, performance analysis strategies, maintaining security in software evolution, learning from evolution for evolution, and formal verification of evolutionary changes. In turn, the last part of the book presents key findings and spin-offs. The individual chapters describe various case studies, along with their benefits, deliberations and the respective learning insights. An overview of future research topics rounds out the coverage. The book was mainly written for scientific researchers and advanced professionals with an academic background. They will benefit from its comprehensive treatment of various topics related to problems that are now gaining in importance, given the high speed of maintenance and evolution in comparison to the initial development, and the fact that today, most software is not developed from scratch, but as a part of a continuum of former and future releases.

Product-Focused Software Process Improvement - Xavier Franch 2020-01-14 This book constitutes the refereed proceedings of the 29th International Conference on Product-Focused Software Process Improvement, PROFES 2019, held in Barcelona, Spain, in November 2019. The 2.4 revised full papers 4 industry papers, and 11 short papers presented were carefully reviewed and selected from 304 submissions. The papers cover a broad range of topics related to professional software development and process improvement driven by product and service quality needs. They are organized in topical sections on testing, software development, technical debt, estimations, continuous delivery, agile, project management, microservices, and continuous experimentation. This book also includes papers from the co-located events: 10 project papers, 8 workshop papers, and 4 tutorial summaries.

Advanced Information Systems Engineering - Eric Dubois 2017-06-01 This book constitutes the refereed proceedings of the 29th International Conference on Advanced Information Systems Engineering, CAiSE 2017, held in Essen, Germany, in June 2017. The 37 papers presented together with 3 keynote papers in this volume were carefully reviewed and selected from 175 submissions. The papers cover a broad range of topics related to professional software development and process improvement driven by product and service quality needs. They are organized in topical sections on testing, software development, technical debt, estimations, continuous delivery, agile, project management, microservices, and continuous experimentation. This book also includes papers from the co-located events: 10 project papers, 8 workshop papers, and 4 tutorial summaries.

Introduction to Mechatronic Design - James Edward Cayer 2011 Introduction to Mechatronic Design is ideal for upper level and graduate Mechatronics courses in Electrical, Computing, or Mechanical & Aerospace Engineering. Unlike other texts on mechatronics that focus on derivations and calculations, Introduction to Mechatronics, 1e, takes a narrative approach, emphasizing the importance of building intuition and understanding before diving into the math. The authors believe that integration of the core of mechatronics and students must have a command of each of the domains to create the balance necessary for successful mechatronic design and that a section on system design and engineering. A robust package of teaching and learning resources accompanies the book.

Situational Method Engineering - Brian Henderson-Sellers 2014-06-03 While previously available methodologies for software - like those published in the early days of object technology - claimed to be appropriate for every conceivable project, situational method engineering (SME) acknowledges that most projects typically have individual characteristics and situations. Thus, finding the most effective methodology for a particular project needs specific tailoring to that situation. Such a tailored software development methodology needs to take into account all the facts and pieces needed for an organization to develop software, including the software process, the input and output work products, the people involved, the languages used to describe requirements, design, code, and eventually also measures of success or failure.

The authors have structured the book into three parts. Part I deals with all the basic concepts, terminology and overall ideas underpinning situational method engineering. As a summary of this part, they present a formal meta-model that enables readers to create their own quality methods and supporting tools. In Part II, they explain how to implement SME in practice, i.e., how to find method components and put them together and how to evaluate the resulting method. For illustration, they also include several industry case studies of customized or constructed processes, highlighting the impact that high-quality engineered methods can have on the success of an industrial software development. Finally, Part III summarizes some of the more recent and forward-looking ideas. This book presents the first summary of the state of the art for SME. For academics, it provides a comprehensive conceptual framework and discusses new research areas. For lecturers, thanks to its step-by-step explanations from basics to the customization and quality assessment of constructed methods, it serves as a solid basis for comprehensive courses on the topic. For industry methodologists, it offers a reference guide on features and technologies to consider when developing in-house software development methods or customizing and adopting off-the-shelf ones.

Perspectives on Software Requirements - Julio Cesar Sampaio do Prado Leite 2012-12-06 Perspectives On Software Requirements presents perspectives on current approaches to software requirements. Each chapter addresses a specific problem where the authors summarize their experiences and results to produce well-fit and traceable requirements. Chapters highlight familiar issues with recent results and experiences, which are accompanied by chapters discussing well-tuned new methods for specific domains.


Situational Method Engineering: Fundamentals and Experiences - Julia Ralpht 2007-09-10 Over the last decade, Method Engineering, defined as the engineering discipline to design, construct and adapt methods, including supportive tools, has emerged as the research and application area for using methods for systems development. This book contains the papers from the IFIP Working Group 8.1 conference on Situational Method Engineering.

The Future of Software Quality Assurance - Stephan Goerzke 2011-11-19 This open access book, published to mark the 15th anniversary of the International Software Quality Institute (ISQI), is intended to raise the profile of software testers and their profession. It gathers contributions by respected software testing experts in order to highlight the state of the art as well as future challenges and trends. In addition, it covers current and emerging technologies like test automation, DevOps, and artificial intelligence methodologies used for software testing, before taking a look into the future. The contributing authors answer questions like: "How is the profession of tester currently changing? What should testers be prepared for in the years to come, and what skills will the next generation need? What opportunities are available for further training today?" What will testing look like in an agile world that is user-centered and fast-paced? What tasks will remain for testers once the most important processes are automated? ISQI has been focused on the education and certification of software testers for fifteen years now, and in this process has contributed to improving the quality of software in many areas. The papers gathered here clearly reflect the numerous ways in which software quality assurance can play a critical role in various areas. Accordingly, the book will be of interest to both professional software testers and managers working in software testing or software quality assurance.

Requirements Engineering and Management for Software Development Projects - Murali Chematari 2012-09-27 Requirements Engineering and Management for Software Development Projects presents a complete guide on requirements for software development including engineering, computer science and management activities. It is the first book to cover all aspects of requirements management in software development projects. This book introduces the understanding of the requirements, elicitation and gathering, requirements analysis, verification and validation of the requirements, establishment of requirements, different methodologies in brief, requirements traceability and change management among other topics. The best practices, pitfalls, and metrics used for efficient software requirements management are also covered. Intended for the professional market, including software engineers, programmers, designers and researchers, this book is also suitable for advanced-level students in computer science or engineering courses as a textbook or reference.