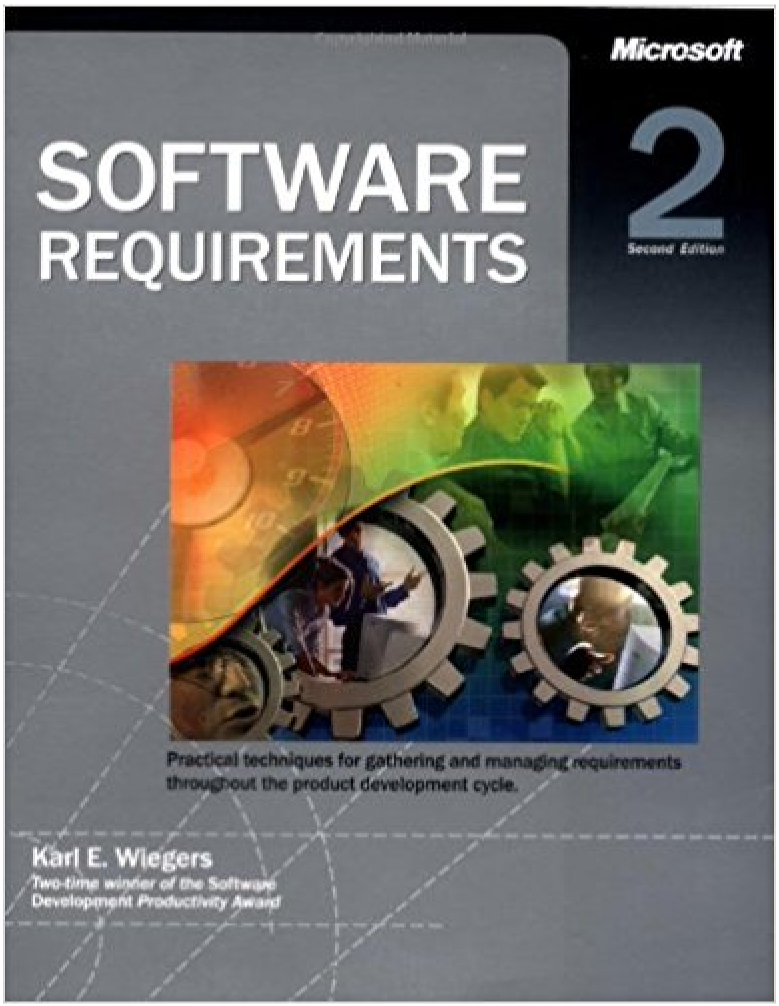
The book was found

Software Requirements 2


Microsoft
SOFTWARE
REQUIREMENTS
2
Second Edition
Practical techniques for gathering and managing requirements throughout the product development cycle.
Karl E. Wiegers
Two-time winner of the Software Development Productivity Award

Synopsis

Without formal, verifiable software requirementsÃ¢â ¬â •and an effective system for managing themÃ¢â ¬â •the programs that developers think theyÃ¢â ¬â„¢ve agreed to build often will not be the same products their customers are expecting. In SOFTWARE REQUIREMENTS, Second Edition, requirements engineering authority Karl Wiegers amplifies the best practices presented in his original award-winning text?now a mainstay for anyone participating in the software development process. In this book, youÃ¢â ¬â„¢ll discover effective techniques for managing the requirements engineering process all the way through the development cycleÃ¢â ¬â •including dozens of techniques to facilitate that all-important communication between users, developers, and management. This updated edition features new case examples, anecdotes culled from the authorÃ¢â ¬â„¢s extensive consulting career, and specific Next Steps for putting the bookÃ¢â ¬â„¢s process-improvement principles into practice. YouÃ¢â ¬â„¢ll also find several new chapters, sample documents, and an incisive troubleshooting guide. Discover how to: Set achievable expectations for functionality and quality NEW: Incorporate business rules into application development Employ use cases to discover user requirements Arrest creeping requirements and manage change requests NEW: Deal with requirements on maintenance, outsourced, and package solution projects Curb the impulse to Ã¢â ¬Å“gold-plateÃ¢â ¬Â• your programs NEW: Grow effective requirements analysts Cut revisionsÃ¢â ¬â •and costsÃ¢â ¬â •dramatically Produce better software! No matter what kind of software you build, or what your role in the development process, SOFTWARE REQUIREMENTS, Second Edition, delivers expert guidance and field-tested techniques for engineering software success.

Book Information

Paperback: 544 pages

Publisher: Microsoft Press; 2nd ed. edition (March 26, 2003)

Language: English

ISBN-10: 0735618798

ISBN-13: 978-0735618794

Product Dimensions: 7.5 x 1.5 x 9 inches

Shipping Weight: 1.8 pounds

Average Customer Review: 4.5 out of 5 stars 51 customer reviews

Best Sellers Rank: #426,464 in Books (See Top 100 in Books) #37 inÃ Â Books > Computers & Technology > Computer Science > AI & Machine Learning > Expert Systems #84 inÃ Â Books >

Computers & Technology > Programming > Software Design, Testing & Engineering > Tools   #87 inÃ Â Books > Computers & Technology > Computer Science > AI & Machine Learning > Machine Theory

Customer Reviews

Karl E. Wiegers is a leading speaker, author, and consultant on requirements engineering, project management, and process improvement. As Principal Consultant with Process Impact, he conducts training seminars for corporate and government clients worldwide. Karl has twice won the Software Development Productivity Award, which honors excellence in productivity-enhancing products and books.

I did not find this book as good as other reviewers seemed to think.To give a bit of a background, I'm work on a fairly complex robotics system with both hardware and software components. Determining system requirements is one of the major tasks that I perform on a daily basis.I did find this book useful, as after reading it I had a much better perspective from the 5000 foot level of how requirements affect the entire business organization and product development process. It really drills down the importance of understanding what it is you are trying to build before designing a product. This book is 100% useful to engineers that focus on hardware as well as software - don't be fooled by the title.My main gripe with the book is that it reads like an overly verbose company work instruction (imagine a dry and somewhat boring employee orientation manual). There are entire pages of material that could have been summed up much better in a paragraph or so, or with an effective picture. Many of the flow charts in the book for example read like your typical overly complex and useless business process charts that no one would ever actually reference when they do their job.For the record, I don't have a problem with flow-charts, but they need to be simple to be effective, and despite what I stated above, there are a couple flow-chart gems in the book.On to the specifics...Section I of the book focuses on a very general overview of requirements, roles that different people perform in relation to requirements (i.e. designers, product managers, project managers, system analyst, testers, etc.). I found most of this material to be useless, except for comparing how my company structured itself versus standard practice across the industry. I imagine that anyone working at a company with any sort of formalized requirements process wouldn't get too much out of this section either.Section II is a bit better. It begins with a focus on the business aspects of requirements. Without a clear scope document and business vision, the requirements are meaningless. It goes on to argue the importance of customer feedback during the requirements development process. I think some clearer explanation on gathering customer feedback, with specific case studies for how to drill down customer desires into tangible features to incorporate into a product would have gone a long way here... but I generally agree with the philosophy. Much of the use case / customer feedback chapters were too general to be useful though.Section II also covers good practices in documenting requirements. I took this part for granted because my company has many of these practices already in place, but had I been working for a smaller company or startup, I would have found these organizational tips to be invaluable. This is really great material if you don’t have any formalized process in place.Section III covers a lot of issues related to version control, changing requirements during the development process, maintaining traceable requirements... This section is boring, and could have benefited by being more concise as well. Reading it did give me a better perspective of the requirements process however.Section IV, a very short section, was really one of the best parts of the book. It ties together the other sections in some of the effective flow charts of the book as to how requirements management is a PROCESS, and one that lies at the heart of good product development. If you’ve ever been through a project and had a gut feeling that major decisions were being rushed without due consideration, or that the wrong tasks were being prioritized, this section will crystallize how things should have gone in that project. It covered a few things I hadn’t seen before, such as measuring requirements volatility, which is a good way to get a handle on how well the product is defined over a long period of time.Finally in the appendices, there are a couple hidden gems that cover the maturity level of a company, and what level of requirements management are actually NEEDED by company depending on the project. For a simple project, requirement management tools would be major overkill.Overall a good book for managing requirements. People that work on large, technically challenging projects in large groups / organizations would find this book especially useful. It suffers though from being way too long for the information that it is trying to impart.My review might a little harsh, as I could see myself re-reading a couple selected parts of this book from time to time, but I simply would never rate this book anything close to 5 stars, and I’m surprised by how high other reviewers have esteemed this book.

"Software Requirements" was an excellent read, with vastly more infomation and topics covered than I expected. Requirements were presented as an "ecosystem", with Good Practices, Setting Vision/Scope, and listening to the Customer well presented in opening paragraphs. Since software requirements cover such a vast area, no one particular domain was addressed in detail. Instead, their general traits were well addressed.I particularly appreciated the added details for

Requirements Manaqement, handling Changes, Traceability, and Prototyping. Also, an entire chapter was devoted to Risk Management, and thus risk reduction, which was of very practical benefit to me. Like everyone, I operate under intense schedule and budget pressure; if I had unlimited time and budget, I could simply iterate "forever". Instead, this book aptly provided practical day-to-day guidance for my real-life software projects. I will share the learned insight with my co-workers and clients alike. A very valuable read, with something for everyone. (My only suggestion would be to publish several sequels addressing unique needs of different domains.)

When I first purchased this book back in 2006, I went through most of it fairly quickly, and even downloaded some of Wiegers supplementary material from processimpact.com. Some of my colleagues took notice and together we formed a mini-workshop using the downloaded materials, which we used to gather requirements for an enhancement to one of the applications I assist in managing.A merger and subsequent reorganization later, I found the time to finish the book. I recommend it for the developer or technical manager who finds themselves in a project that lacks thorough requirements development. The book uses appropriate tone and terminology to address its' intended audience; it is neither too simplistic nor overly dense. It has enough supplementary material to preclude the need to build a requirements development process from scratch without looking too much like a cookbook. Its' bibliography includes several classics and many references not familiar to me. All in all, a balanced book about requirements development and management.

I'm somewhat of a software engineering/process geek. I find the process of creating a product more interesting than the actual code these days (though I like to code). Wiegers' book is THE bible, in my opinion, for eliciting and maintaining requirements.He covers the issues involved in gathering requirements and keeping them up to date, often offering multiple ways to resolve issues. Wiegers, unlike many academic oriented books, fully acknowledges the political and cultural difficulties that arise when trying to institute a requirements program. Much of his advice is practical and he gives good pointers on the highst ROI practices, so you can inject a little at a time, rather than trying to change culture wholesale.I'd give a 4.5 out of 5 if I could, due only to the "Next Steps" sections at the end of each chapter. The "Next Steps" are supposedly be small steps you can take to start using the advice Wiegers offers. Unfortunately, most of the steps start with "Take a page/chapter from your current requirements document...." I've worked at few companies that even have a requirements document, so I'm not sure how useful the "Next Steps" really are.But, that complaint aside, this book is the best combination of reference information for techniques and advice on how to use them on the job.

Download to continue reading...

The Software Requirements Memory Jogger: A Pocket Guide to Help Software And Business Teams Develop And Manage Requirements (Memory Jogger) Software Engineering: The Current Practice (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series) Veterinary Medical School Admission Requirements (VMSAR): 2017 Edition for 2018 Matriculation (Veterinary Medical School Admission Requirements in the United States and Canada) Veterinary Medical School Admission Requirements (VMSAR): 2016 Edition for 2017 Matriculation (Veterinary Medical School Admission Requirements in the United States and Canada) ACI 318.2-14: Building Code Requirements for Concrete Thin Shells (ACI 318.2-14) and Commentary on Building Code Requirements for Concrete Thin Shells (ACI 318.2R-14) Nutrient Requirements of Dogs and Cats (Nutrient Requirements of Domestic Animals) Mastering the Requirements Process: Getting Requirements Right (3rd Edition) IEC 61511-1 Ed. 1.0 b:2003, Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements Software Requirements 2 Managing Software Requirements (paperback): A Use Case Approach (Addison-Wesley Object Technology (Paperback)) Software Requirements (3rd Edition) (Developer Best Practices) Head First Software Development: A Learner's Companion to Software Development Agile Project Management: Agile Revolution, Beyond Software Limits: A Practical Guide to Implementing Agile Outside Software Development (Agile Business Leadership, Book 4) Don't Buy Software For Your Small Business Until You Read This Book: A guide to choosing the right software for your SME & achieving a rapid return on your investment Software Agreements Line by Line, 2nd ed.: A Detailed Look at Software Agreements and How to Draft Them to Meet Your Needs IEC 62304 Ed. 1.0 b:2006, Medical device software - Software life cycle processes Agile Software Development with Scrum (Series in Agile Software Development) Anonymous Security Systems and Applications: Requirements and Solutions Energy Transitions: History, Requirements, Prospects Hard Work:Defining Physical Work Performance Requirements