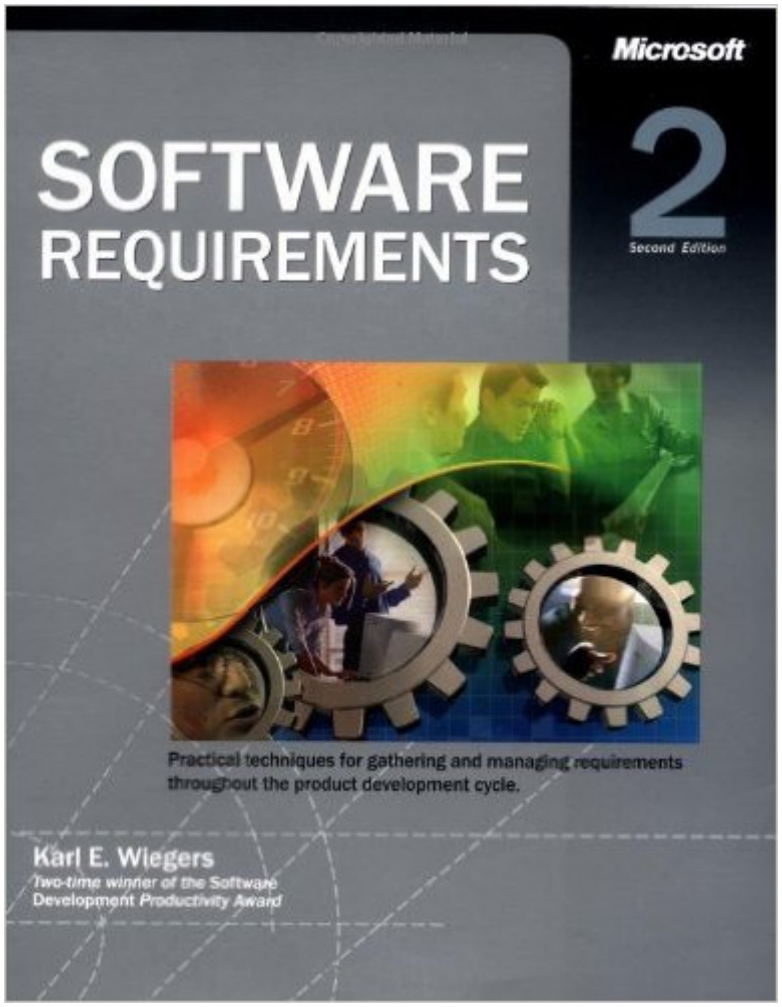
Software Requirements 2



Synopsis

Without formal, verifiable software requirementsâ ”and an effective system for managing themâ ”the programs that developers think theyâ ™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, youâ ™ll discover effective techniques for managing the requirements engineering process all the way through the development cycleâ ”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 authorâ ™s extensive consulting career, and specific Next Steps for putting the bookâ ™s process-improvement principles into practice. Youâ ™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 â œgold-plateâ • your programs NEW: Grow effective requirements analysts Cut revisionsâ ”and costsâ ”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

Book Information

Paperback: 544 pages

Publisher: Microsoft Press; 2 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: 2.3 pounds

Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (48 customer reviews)

Best Sellers Rank: #547,043 in Books (See Top 100 in Books) #34 inÂ Books > Computers & Technology > Computer Science > AI & Machine Learning > Expert Systems #97 inÂ Books > Computers & Technology > Computer Science > AI & Machine Learning > Machine Theory #115

inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Tools

Customer Reviews

How do you know if you have good software requirements? Some use the simple technique of checking if the requirements definition is complete, clear, and consistent. Every book on requirements engineering has some variation of this theme and in this book, you are advised to check if the requirements statement is complete, correct, feasible, necessary, prioritized, unambiguous, and verifiable.If you haven't used techniques like this one before, it is definitely a good idea to pick up a solid book like this one on the best practices in requirements engineering. There are several good books in the market on the topic of software requirements and this is one of the best ones out there.I found three other books that complement this one - Requirements Engineering by Kotonya and Sommerville (used more as a textbook), Managing Software Requirements by Leffingwell and Widrig (part of the Object Technology Series), and Effective Requirements Practices by Ralph R. Young (comes with a CD-ROM).If you are a project manager, business analyst or anyone that has a lot to lose because of bad requirements, you will benefit tremendously from this current book being reviewed. The book is divided into three parts - What and Why, Development, and Management of Software Requirements. The part names are self explanatory. This book is very readable and is full of best practices that stand true to their name!The unique things about this book - in chapter 2, the author outlines the Requirements Bill of Rights for Software Customers and the Requirements Bill of Responsibilities for Software Customers. When I first read this, I felt like every customer has to read this before attempting a software project. Chapter 10 has an excellent description of different diagrams useful in requirements documentation - DFD (data flow diagram), ERD (entity-relationship diagram), STD (state transition diagram), dialog map, and class diagrams. I think all books on software requirements should ideally have some variation of these topics.Important topics like traceability are given an excellent treatment in this book but the only thing lacking is how to manage requirements in software processes involving iterations (the mainstay of the Rational Unified Process and other newer software development methodologies). There are only 13 pages devoted to this topic and even then it is indirect - Chapter 12: Risk Reduction Through Prototyping.Otherwise, I have no complaints about this book and I believe that it is a basic to intermediate in level (definitely not an advanced book). Overall, I believe it indeed captures the best practices in the field of requirements engineering. It is also a good price, so enjoy!

This book faces a lot of competition from other books, which are supposed to tell you how to manage software projects in general, and the requirements gathering process in particular.However, what sets this book apart from the vast majority of others is its absolute relevance (as opposed to being an arbitrary textbook). For example, this book recognizes the fact that often enough process improvements are deferred due to political reasons alone. The more you read it, the more you realize it addresses the same problems you have encountered while managing the requirements process.But what really sets this book apart is that it actually tells you how to solve these problems, by offering feasible solutions that could be easily implemented, gradually, in real life scenarios. This, basically, means that the book could actually HELP you.

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.

When it comes to the development life cycle, there are generally two broad schools of thought: rigorous, waterfall approach; and the agile, iterative approach. This text sits in the heart of the rigorous, waterfall approach.Iterative approaches are proven to be more effective at eliciting requirements, a fact which is somewhat embraced in the author's discussion of use cases; however, Jacobson originally envisioned use cases to replace other requirements documents as a central element in elicitation, rather than just being a quick diversion.In reality, most of us strike a middle ground. Projects can't be run in most organizations without rigor, and Software Requirements is a

thorough treatment or requirements development and management. The well-organized book is a quick read, and is filled with prescriptive advice, risks, sample forms, and checklists that can be applied to your requirements effort. No wonder the author won a Software Productivity Award for the effort!

MCPD Self-Paced Training Kit (Exams 70-536, 70-528, 70-547): MicrosoftÃ Â® .NET Framework Web Developer Core Requirements: Microsoft .Net Framework Web ... Requirements (Microsoft Press Training Kit) Nutrient Requirements of Dogs (Nutrient Requirements of Domestic Animals) Veterinary Medical School Admission Requirements (VMSAR): 2015 Edition for 2016 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) Non-Functional Requirements in Software Engineering (International Series in Software Engineering) Software Requirements (3rd Edition) (Developer Best Practices) Software Requirements: Encapsulation, Quality, and Reuse Software Requirements 2 Requirements Engineering: From System Goals to UML Models to Software Specifications Just Enough Requirements Management: Where Software Development Meets Marketing Agile Product Management: User Stories: How to capture, and manage requirements for Agile Product Management and Business Analysis with Scrum (scrum, ... development, agile software development) Agile Product Management: Product Owner (Box set) : 27 Tips To Manage Your Product, Product Backlog: 21 Tips To Capture and Manage Requirements with Scrum ... development, agile software development) Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection UML Requirements Modeling For Business Analysts ISO 14001:2004, Environmental management systems - Requirements with guidance for use Nutrient Requirements of Horses: Sixth Revised Edition (Animal Nutrition Series) Capital Requirements, Disclosure, and Supervision in the European Insurance Industry: New Challenges towards Solvency II Tax Facts on Insurance & Employee Benefits 2015: Annuities, Cafeteria Plans, Compensation, Disclosure Requirements, Estate and Gift Taxation, Health ... Facts on Insurance and Employee Benefits) Agile Product Management and Product Owner Box Set: 27 Tips to Manage Your Product, Product Backlog and 21 Tips to Capture and Manage Requirements with Scrum