

软件需求 第3版

Software Requirements Third Edition



经过实践检验的需求工程——全面更新和补充

在第3版中，这本需求工程方面的经典书籍得到了全面更新，包括新的主题、样例和认知。在两位软件需求社区领导者的带领下，你将获得覆盖所有需求开发和管理活动的最新实践和工具。你可以把这些可行的、切合实际的技术带到工作中去——它能够帮助你减少开销、降低维护成本和减少为了交付更佳目标解决方案而不得不进行的返工。

改进你的团队：

- 确定关键决策者并与之协作
- 在关注商业目标的前提下提取和分析需求
- 对需求进行记录、优先级排序、验证和重用
- 实现需求的原型化和创建可视化模型
- 管理需求变更、范围渐变和需求风险
- 理解和明确客户质量期望

本版中的新内容：

- 把特定实践带入敏捷项目的指导
- 聚焦商业分析师的角色和成功要素
- 为商业流程自动化、打包解决方案、外包、改进或替换以及嵌入系统项目推荐的方法
- 数据和报告需求方面的指导

在线内容

请从 <http://aka.ms/SoftwareReq3E/files> 处

获取本书中的模板、清单、表格以及其他工作辅助。

“商业分析的经典著作之一，该第3版很好地诠释了业界的最新趋势。”

Kevin Brennan, 首席商业分析师和商业分析国际学院执行副总裁

关于读者

本书适合于商业分析师和需求工程师，以及软件架构师、开发人员、项目主管和其他决策者。

作者简介

Karl Wiegers 博士是软件需求工程、过程改进和软件质量方面备受推崇的专家。作为一名有多项荣誉的作家，Karl 撰写了多本著作，包括《软件需求》的前一版。

Joy Beatty, 软件需求社区的领导者，培训过数千计的商业分析师并帮助过大多数组织搭建绝佳的商业分析中心。她是《Visual Models for Software Requirements》的作者之一。

ISBN 978-7-5641-4988-8



9 787564 149888 >

定价：99.00 元



软件需求 第3版(影印版)

Software Requirements, 3E

Karl Wieggers, Joy Beatty 著

南京 东南大学出版社

图书在版编目 (CIP) 数据

软件需求: 第3版: 英文 / (美) 维格 (Wieggers, K.),
(美) 比蒂 (Beatty, J.) 著. —影印本. —南京: 东南大学出版社, 2014.9

书名原文: Software Requirements, 3E

ISBN 978-7-5641-4988-8

I. ① 软… II. ① 维… ② 比… III. ① 软件需求—英文
IV. ① TP311.52

中国版本图书馆 CIP 数据核字 (2014) 第 107963 号

江苏省版权局著作权合同登记

图字: 10-2014-168 号

©Southeast University Press[2014]

Authorized the English edition of Software Requirements, 3rd Edition
©[Karl Wieggers, Joy Beatty]. This English edition is published by
permission of O'Reilly Media, Inc., the owner of all rights to publish
and sell the same.

软件需求 第3版 (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼2号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seupress.com

印 刷: 常州市武进第三印刷有限公司

开 本: 787毫米×980毫米 12开本

印 张: 56

字 数: 1019千字

版 次: 2014年9月第1版

印 次: 2014年9月第1次印刷

书 号: ISBN 978-7-5641-4988-8

定 价: 99.00元

本社图书若有印装质量问题, 请直接与营销部联系。电话 (传真): 025-83791830

Praise for this book

"Software Requirements, Third Edition, is the most valuable requirements guidance you will find. Wiegers and Beatty cover the entire landscape of practices that today's business analyst is expected to know. Whether you are a veteran of requirements specification or a novice on your first project, this is the book that needs to be on your desk or in your hands."

—Gary K. Evans, Agile Coach and Use Case Expert, Evanetics, Inc.

*"It's a three-peat: Karl Wiegers and Joy Beatty score again with this third edition. From the first edition in 1999 through each successive edition, the guidance that *Software Requirements* provides has been the foundation of my requirements consulting practice. To beginning and experienced practitioners alike, I cannot recommend this book highly enough."*

—Roxanne Miller, President, Requirements Quest

"The best book on requirements just got better! The third edition's range of new topics expands the project circumstances it covers. Using requirements in agile environments is perhaps the most significant, because everyone involved still needs to understand what a new system must do—and agile developers are now an audience who ought to have a good grasp of what's in this book."

—Stephen Withall, author of Software Requirement Patterns

*"The third edition of *Software Requirements* is finally available—and it was worth waiting so long. Full of practical guidance, it helps readers identify many useful practices for their work. I particularly enjoy the examples and many hands-on solutions that can be easily implemented in real-life scenarios. A must-read, not only for requirements engineers and analysts but also for project managers."*

—Dr. Christof Ebert, Managing Director, Vector Consulting Services

"Karl and Joy have updated one of the seminal works on software requirements, taking what was good and improving on it. This edition retains what made the previous versions must-have references for anyone working in this space and extends it to tackle the challenges faced in today's complex business and technology environment. Irrespective of the technology, business domain, methodology, or project type you are working in, this book will help you deliver better outcomes for your customers."

—Shane Hastie, Chief Knowledge Engineer, Software Education

"Karl Wiegers's and Joy Beatty's new book on requirements is an excellent addition to the literature. Requirements for large software applications are one of the most difficult business topics of the century. This new book will help to smooth out a very rough topic."

—T. Capers Jones, VP and CTO, Namcook Analytics LLC

"Simply put, this book is both a must-read and a great reference for anyone working to define and manage software development projects. In today's modern software development world, too often sound requirements practices are set aside for the lure of "unencumbered" agile. Karl and Joy have detailed a progressive approach to managing requirements, and detailed how to accommodate the ever-changing approaches to delivering software."

—Mark Kulak, Software Development Director, Borland, a Micro Focus company

"I am so pleased to see the updated book on software requirements from Karl Wiegers and Joy Beatty. I especially like the latest topic on how to apply effective requirements practices to agile projects, because it is a service that our consultants are engaged in more and more these days. The practical guide and real examples of the many different requirement practices are invaluable."

—Doreen Evans, Managing Director of the Requirements and Business Analysis Practice for Robbins Gioia Inc.

"As an early adopter of Karl's classic book, *Software Requirements*, I have been eagerly awaiting his new edition—and it doesn't disappoint. Over the years, IT development has undergone a change of focus from large, new, 'green-field' projects towards adoption of ready-made off-the-shelf solutions and quick-release agile practices. In this latest edition, Karl and Joy explore the implications of these new developments on the requirements process, with invaluable recommendations based not on dogma but on what works, honed from their broad and deep experience in the field."

—Howard Podeswa, CEO, Noble Inc., and author of *The Business Analyst's Handbook*

"If you are looking for a practical guide into what software requirements are, how to craft them, and what to do with them, then look no further than *Software Requirements, Third Edition*. This usable and readable text walks you through exactly how to approach common requirements-related scenarios. The incorporation of multiple stories, case studies, anecdotes, and examples keeps it engaging to read."

—Laura Brandenburg, CBAP, Host at *Bridging the Gap*

"How do you make a good requirements read better? You add content like Karl and Joy did to address incorporating product vision, tackling agility issues, covering requirements reuse, tackling packaged and outsourced software, and addressing specific user classes. You could take an outside look inside of requirements to address process and risk issues and go beyond just capturing functionality."

—Donald J. Reifer, President, Reifer Consultants LLC

"This new edition keeps pace with the speed of business, both in deepening the foundation of the second edition and in bringing analysts down-to-earth how-to's for addressing the surge in agile development, using features to control scope, improving elicitation techniques, and expanding modeling. Wiegers and Beatty have put together a must-read for anyone in the profession."

—Keith Ellis, President and CEO, Enfocus Solutions Inc., and author of *Business Analysis Benchmark*

For Chris, yet again. Eighth time's the charm.

—K.W.

For my parents, Bob and Joanne, for a lifetime of encouragement.

—J.B.

Introduction

Despite decades of industry experience, many software organizations struggle to understand, document, and manage their product requirements. Inadequate user input, incomplete requirements, changing requirements, and misunderstood business objectives are major reasons why so many information technology projects are less than fully successful. Some software teams aren't proficient at eliciting requirements from customers and other sources. Customers often don't have the time or patience to participate in requirements activities. In many cases, project participants don't even agree on what a "requirement" is. As one writer observed, "Engineers would rather decipher the words to the Kingsmen's 1963 classic party song 'Louie Louie' than decipher customer requirements" (Peterson 2002).

The second edition of *Software Requirements* was published 10 years prior to this one. Ten years is a long time in the technology world. Many things have changed in that time, but others have not. Major requirements trends in the past decade include:

- The recognition of business analysis as a professional discipline and the rise of professional certifications and organizations, such as the International Institute of Business Analysis and the International Requirements Engineering Board.
- The maturing of tools both for managing requirements in a database and for assisting with requirements development activities such as prototyping, modeling, and simulation.
- The increased use of agile development methods and the evolution of techniques for handling requirements on agile projects.
- The increased use of visual models to represent requirements knowledge.

So, what *hasn't* changed? Two factors contribute to keeping this topic important and relevant. First, many undergraduate curricula in software engineering and computer science continue to underemphasize the importance of requirements engineering (which encompasses both requirements development and requirements management). And second, those of us in the software domain tend to be enamored with technical and process solutions to our challenges. We sometimes fail to appreciate that requirements elicitation—and much of software and systems project work in general—is primarily a human interaction challenge. No magical new techniques have come along to automate that, although various tools are available to help geographically separated people collaborate effectively.

We believe that the practices presented in the second edition for developing and managing requirements are still valid and applicable to a wide range of software projects. The creative business analyst, product manager, or product owner will thoughtfully adapt and scale the practices to best meet the needs of a particular situation. Newly added to this third edition are a chapter on handling requirements for agile projects and sections in numerous other chapters that describe how to apply and adapt the practices in those chapters to the agile development environment.

Software development involves at least as much communication as it does computing, yet both educational curricula and project activities often emphasize the computing over the communication aspect. This book offers dozens of tools to facilitate that communication and to help software practitioners, managers, marketers, and customers apply effective requirements engineering methods. The techniques presented here constitute a tool kit of mainstream “good practices,” not exotic new techniques or an elaborate methodology that purports to solve all of your requirements problems. Numerous anecdotes and sidebars present stories—all true—that illustrate typical requirements-related experiences; you have likely had similar experiences. Look for the “true stories” icon, like the one to the left, next to real examples drawn from many project experiences.



Since the first edition of this book appeared in 1999, we have each worked on numerous projects and taught hundreds of classes on software requirements to people from companies and government agencies of all sizes and types. We’ve learned that these practices are useful on virtually any project: small projects and large, new development and enhancements, with local and distributed teams, and using traditional and agile development methods. The techniques apply to hardware and systems engineering projects, too, not just software projects. As with any other technical practice, you’ll need to use good judgment and experience to learn how to make the methods work best for you. Think of these practices as tools to help ensure that you have effective conversations with the right people on your projects.

Benefits this book provides

Of all the software process improvements you could undertake, improved requirements practices are among the most beneficial. We describe practical, proven techniques that can help you to:

- Write high-quality requirements from the outset of a project, thereby minimizing rework and maximizing productivity.

- Deliver high-quality information systems and commercial products that achieve their business objectives.
- Manage scope creep and requirements changes to stay both on target and under control.
- Achieve higher customer satisfaction.
- Reduce maintenance, enhancement, and support costs.

Our objective is to help you improve the processes you use for eliciting and analyzing requirements, writing and validating requirements specifications, and managing the requirements throughout the software product development cycle. The techniques we describe are pragmatic and realistic. Both of us have used these very techniques many times, and we always get good results when we do.

Who should read this book

Anyone involved with defining or understanding the requirements for any system that contains software will find useful information here. The primary audience consists of individuals who serve as business analysts or requirements engineers on a development project, be they full-time specialists or other team members who sometimes fill the analyst role. A second audience includes the architects, designers, developers, testers, and other technical team members who must understand and satisfy user expectations and participate in the creation and review of effective requirements. Marketers and product managers who are charged with specifying the features and attributes that will make a product a commercial success will find these practices valuable. Project managers will learn how to plan and track the project's requirements activities and deal with requirements changes. Yet another audience is made up of stakeholders who participate in defining a product that meets their business, functional, and quality needs. This book will help end users, customers who procure or contract for software products, and numerous other stakeholders understand the importance of the requirements process and their roles in it.

Looking ahead

This book is organized into five parts. Part I, "Software requirements: What, why, and who," begins with some definitions. If you're on the technical side of the house, please share Chapter 2, on the customer-development partnership, with your key customers. Chapter 3 summarizes several dozen "good practices" for requirements development

and management, as well as an overall process framework for requirements development. The role of the business analyst (a role that also goes by many other names) is the subject of Chapter 4.

Part II, “Requirements development,” begins with techniques for defining the project’s business requirements. Other chapters in Part II address how to find appropriate customer representatives, elicit requirements from them, and document user requirements, business rules, functional requirements, data requirements, and nonfunctional requirements. Chapter 12 describes numerous visual models that represent the requirements from various perspectives to supplement natural-language text, and Chapter 15 addresses the use of prototypes to reduce risk. Other chapters in Part II present ways to prioritize, validate, and reuse requirements. Part II concludes by describing how requirements affect other aspects of project work.

New to this edition, Part III contains chapters that recommend the most effective requirements approaches for various specific classes of projects: agile projects developing products of any type, enhancement and replacement projects, projects that incorporate packaged solutions, outsourced projects, business process automation projects, business analytics projects, and embedded and other real-time systems.

The principles and practices of requirements management are the subject of Part IV, with emphasis on techniques for dealing with changing requirements. Chapter 29 describes how requirements tracing connects individual requirements both to their origins and to downstream development deliverables. Part IV concludes with a description of commercial tools that can enhance the way your teams conduct both requirements development and requirements management.

The final section of this book, Part V, “Implementing requirements engineering,” helps you move from concepts to practice. Chapter 31 will help you incorporate new requirements techniques into your group’s development process. Common requirements-related project risks are described in Chapter 32. The self-assessment in Appendix A can help you select areas that are ripe for improvement. Two other appendices present a requirements troubleshooting guide and several sample requirements documents so you can see how the pieces all fit together.

Case studies

To illustrate the methods described in this book, we have provided examples from several case studies based on actual projects, particularly a medium-sized information system called the Chemical Tracking System. Don’t worry—you don’t need to know anything about chemistry to understand this project. Sample discussions among

participants from the case studies are sprinkled throughout the book. No matter what kind of software your organization builds, you'll be able to relate to these dialogs.

From principles to practice

It's difficult to muster the energy needed for overcoming obstacles to change and putting new knowledge into action. As an aid for your journey to improved requirements, most chapters end with several "next steps," actions you can take to begin applying the contents of that chapter immediately. Various chapters offer suggested templates for requirements documents, a review checklist, a requirements prioritization spreadsheet, a change control process, and many other process assets. These items are available for downloading at the companion content website for this book:

<http://aka.ms/SoftwareReq3E/files>

Use them to jump-start your application of these techniques. Start with small improvements, but start today.

Some people will be reluctant to try new requirements techniques. Use this book to educate your peers, your customers, and your managers. Remind them of requirements-related problems encountered on previous projects, and discuss the potential benefits of trying some new approaches.

You don't need to launch a new development project to begin applying better requirements practices. Chapter 21 discusses ways to apply many of the techniques to enhancement and replacement projects. Implementing requirements practices incrementally is a low-risk process improvement approach that will prepare you for the next major project.

The goal of requirements development is to accumulate a set of requirements that are *good enough* to allow your team to proceed with design and construction of the next portion of the product at an acceptable level of risk. You need to devote enough attention to requirements to minimize the risks of rework, unacceptable products, and blown schedules. This book gives you the tools to get the right people to collaborate on developing the right requirements for the right product.

Errata & book support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site at oreilly.com:

<http://aka.ms/SoftwareReq3E/errata>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://aka.ms/tellpress>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *<http://twitter.com/MicrosoftPress>*.

Acknowledgments

Writing a book like this is a team effort that goes far beyond the contributions from the two authors. A number of people took the time to review the full manuscript and offer countless suggestions for improvement; they have our deep gratitude. We especially appreciate the invaluable comments from Jim Brosseau, Joan Davis, Gary K. Evans, Joyce Grapes, Tina Heidenreich, Kelly Morrison Smith, and Dr. Joyce Statz. Additional review input was received from Kevin Brennan, Steven Davis, Anne Hartley, Emily Iem, Matt Leach, Jeannine McConnell, Yaaqub Mohamed, and John Parker. Certain individuals reviewed specific chapters or sections in their areas of expertise, often providing highly detailed comments. We thank Tanya Charbury, Mike Cohn, Dr. Alex Dean, Ellen Gottesdiener, Shane Hastie, James Hulgán, Dr. Phil Koopman, Mark Kulak, Shirley Sartin, Rob Siciliano, and Betsy Stockdale. We especially thank Roxanne Miller and Stephen Withall for their deep insights and generous participation.

We discussed aspects of the book's topics with many people, learning from their personal experiences and from resource materials they passed along to us. We appreciate such contributions from Jim Brosseau, Nanette Brown, Nigel Budd, Katherine Busey, Tanya Charbury, Jennifer Doyle, Gary Evans, Scott Francis, Sarah Gates, Dr. David Gelperin, Mark Kerin, Norm Kerth, Dr. Scott Meyers, John Parker, Kathy Reynolds, Bill Trosky, Dr. Ricardo Valerdi, and Dr. Ian Watson. We also thank the many people who let us share their anecdotes in our "true stories."

Numerous staff members at Seilevel contributed to the book. They reviewed specific sections, participated in quick opinion and experience surveys, shared blog material they had written, edited final chapters, drew figures, and helped us with operational issues of various sorts. We thank Ajay Badri, Jason Benfield, Anthony Chen, Kell Condon, Amber Davis, Jeremy Gorr, Joyce Grapes, John Jertson, Melanie Norrell, David Reinhardt, Betsy Stockdale, and Christine Wollmuth. Their work made ours easier. The editorial input from Candase Hokanson is greatly appreciated.

Thanks go to many people at Microsoft Press, including acquisitions editor Devon Musgrave, project editor Carol Dillingham, project editor Christian Holdener of S4Carlisle Publishing Services, copy editor Kathy Krause, proofreader Nicole Schlutt, indexer Maureen Johnson, compositor Sambasivam Sangaran, and production artists Balaganesan M., Srinivasan R., and Ganeshbabu G. Karl especially values his long-term relationship, and friendship, with Devon Musgrave and Ben Ryan.

The comments and questions from thousands of students in our requirements training classes over the years have been most helpful in stimulating our thinking about

requirements issues. Our consulting experiences and the thought-provoking questions we receive from readers have kept us in touch with what practitioners struggle with on a daily basis and helped us think through some of these difficult topics. Please share your own experiences with us at karl@processimpact.com or joy.beatty@seilevel.com.

As always, Karl would like to thank his wife, Chris Zambito. And as always, she was patient and good-humored throughout the process. Karl also thanks Joy for prompting him into working on this project and for her terrific contributions. Working with her was a lot of fun, and she added a great deal of value to the book. It was great to have someone to bounce ideas off, to help make difficult decisions, and to chew hard on draft chapters before we inflicted them on the reviewers.

Joy is particularly grateful to her husband, Tony Hamilton, for supporting her writing dreams so soon again; to her daughter, Skye, for making it easy to keep her daily priorities balanced; and to Sean and Estelle for being the center of her family fun times. Joy wants to extend a special thanks to all of the Seilevel employees who collaborate to push the software requirements field forward. She particularly wants to thank two colleagues and friends: Anthony Chen, whose support for her writing this book was paramount; and Rob Sparks, for his continued encouragement in such endeavors. Finally, Joy owes a great deal of gratitude to Karl for allowing her to join him in this co-authorship, teaching her something new every day, and being an absolute joy to work with!

Contents at a glance

<i>Introduction</i>	xxv
<i>Acknowledgments</i>	xxxix

PART I	SOFTWARE REQUIREMENTS: WHAT, WHY, AND WHO	
CHAPTER 1	The essential software requirement	3
CHAPTER 2	Requirements from the customer's perspective	25
CHAPTER 3	Good practices for requirements engineering	43
CHAPTER 4	The business analyst	61
PART II	REQUIREMENTS DEVELOPMENT	
CHAPTER 5	Establishing the business requirements	77
CHAPTER 6	Finding the voice of the user	101
CHAPTER 7	Requirements elicitation	119
CHAPTER 8	Understanding user requirements	143
CHAPTER 9	Playing by the rules	167
CHAPTER 10	Documenting the requirements	181
CHAPTER 11	Writing excellent requirements	203
CHAPTER 12	A picture is worth 1024 words	221
CHAPTER 13	Specifying data requirements	245
CHAPTER 14	Beyond functionality	261
CHAPTER 15	Risk reduction through prototyping	295
CHAPTER 16	First things first: Setting requirement priorities	313
CHAPTER 17	Validating the requirements	329
CHAPTER 18	Requirements reuse	351
CHAPTER 19	Beyond requirements development	365
PART III	REQUIREMENTS FOR SPECIFIC PROJECT CLASSES	
CHAPTER 20	Agile projects	383
CHAPTER 21	Enhancement and replacement projects	393
CHAPTER 22	Packaged solution projects	405
CHAPTER 23	Outsourced projects	415

CHAPTER 24	Business process automation projects	421
CHAPTER 25	Business analytics projects	427
CHAPTER 26	Embedded and other real-time systems projects	439
PART IV	REQUIREMENTS MANAGEMENT	
CHAPTER 27	Requirements management practices	457
CHAPTER 28	Change happens	471
CHAPTER 29	Links in the requirements chain	491
CHAPTER 30	Tools for requirements engineering	503
PART V	IMPLEMENTING REQUIREMENTS ENGINEERING	
CHAPTER 31	Improving your requirements processes	517
CHAPTER 32	Software requirements and risk management	537
	<i>Epilogue</i>	549
	<i>Appendix A</i>	551
	<i>Appendix B</i>	559
	<i>Appendix C</i>	575
	<i>Glossary</i>	597
	<i>References</i>	605
	<i>Index</i>	619

Contents

<i>Introduction</i>	xxv
<i>Acknowledgments</i>	xxxi

PART I SOFTWARE REQUIREMENTS: WHAT, WHY, AND WHO

Chapter 1	The essential software requirement	3
	Software requirements defined	5
	Some interpretations of "requirement"	6
	Levels and types of requirements	7
	Working with the three levels	12
	Product vs. project requirements	14
	Requirements development and management.	15
	Requirements development.	15
	Requirements management.	17
	Every project has requirements	18
	When bad requirements happen to good people	19
	Insufficient user involvement.	20
	Inaccurate planning.	20
	Creeping user requirements.	20
	Ambiguous requirements.	21
	Gold plating	21
	Overlooked stakeholders	22
	Benefits from a high-quality requirements process	22
Chapter 2	Requirements from the customer's perspective	25
	The expectation gap.	26
	Who is the customer?.	27
	The customer-development partnership	29
	Requirements Bill of Rights for Software Customers.	31
	Requirements Bill of Responsibilities for Software Customers.	33