

# Preface

Requirements engineering continues to be a hot topic in the software industry. More and more development organizations realize that they cannot succeed unless they get the software requirements right. Too often, the people responsible for leading the requirements process are ill-equipped for this challenging role. They do the best they can, but it's an uphill climb without adequate training, coaching, resources, and experience.

As a consultant, trainer, and author, I receive many questions from practitioners about how to handle difficult requirements issues. Certain questions come up over and over again. Alas, there aren't simple answers for many of these. Many books on requirements engineering have been published during the last several years, including my own, *Software Requirements, Second Edition* (Wieggers 2003a). These books provide solid guidance on the challenges of requirements elicitation, analysis, specification, validation, and management. However, additional requirements topics are not covered well by the existing books. Also, some books contain guidance that I believe is ill-founded.

This book addresses some of these recurrent questions that puzzle and frustrate requirements analysts, such as the following:

- “How do I keep too much design from being embedded in the requirements?” (I heard this question again the day before I wrote these words.)
- “When should I baseline my requirements?”
- “How can I convince my managers that we need to do a better job on our project requirements?”
- “What are some good questions to ask in requirements interviews?”
- “Are use cases all I need for documenting the requirements?”
- “We can't get our customers to review the requirements specification. What should I do?”
- “What are some good metrics our organization should collect about our requirements?”
- “We're collecting requirements for multiple releases concurrently. How should I store those?”
- “How can I use requirements to estimate how long it will take to finish the project?”
- “How can I write better requirements?”

I've addressed some other topics in this book simply because little has been written about them. For instance, everyone talks about project scope, but the current books on requirements engineering say little about how to actually define scope. See Chapter 17, “Defining Project Scope,” for some recommendations. Still other topics are included because I don't see

**xiv Preface**

practitioners using some of the established techniques that can help them do a better job. As an example, nearly all requirements specifications I see consist entirely of written text—there’s not a picture to be found. However, the skilled analyst should have a rich tool kit of techniques available for representing requirements information. Text is fine in many cases, but other sorts of requirements “views” sometimes are more valuable. Chapter 19, “The Six Blind Men and the Requirements,” addresses this topic.



The suggestions I propose in this book augment the “good practices” approach I took in my earlier book. Many cross-references are provided to chapters in *Software Requirements, Second Edition*, marked with the icon shown to the left of this paragraph. As with all such advice, you need to think about how best to apply these suggestions to your specific situation. Organizations are different, projects are different, and cultures are different, so techniques that work in one situation might not be just right for another. To illustrate the application of these practices, I’ve included many examples of actual project experiences, marked with the “true stories” icon shown to the left here.

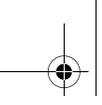


Anyone involved with defining or understanding the requirements for a new or enhanced software product will find this book useful. The primary audience consists of those team members who perform the role of the requirements analyst on a software development project, be this their full-time job or just something they do once in awhile. Part II of this book, “On the Management View of Requirements,” is focused on aspects of requirements engineering that are of particular interest to project managers and senior managers. Customer representatives who work with the software team will also find certain chapters valuable, particularly those in Part III, “On Customer Interactions,” and Part IV, “On Use Cases.”

I should point out that all the practices I recommend assume that you’re dealing with reasonable people. Sometimes an unreasonable customer will insist on a specific solution that isn’t a good fit for the problem. Unreasonable funding sponsors might impose their own inappropriate preferences, overriding the thoughtful decisions made by actual user representatives. Senior managers or influential customers sometimes demand impossible delivery dates for an overly constrained project. If you face such a situation, try educating the difficult people to help them understand the risks posed by the approaches they are demanding and the value of using a better approach. People who appear unreasonable often are just uninformed. Sometimes, though, they truly are unreasonable. I can’t help you much with that.

You may download the templates and other process assets described in this book from the Process Impact Web site, <http://www.processimpact.com>. Feel free to share your experiences with me at [kwiegers@acm.org](mailto:kwiegers@acm.org).

I hope you’ll find this book a valuable supplement to your other resources for software requirements engineering. But don’t just read the chapters and say, “That’s interesting.” Set yourself a personal goal of finding at least three new practices that you want to try the next few times you wear your analyst hat.



## Acknowledgments

First, I thank the many people in my seminars who have asked some of these challenging questions over the past several years, as well as the readers who have sent me e-mails with their own thorny requirements issues. I'm grateful for the review input provided by Wayne Allen, Michael Beshears, Steven Davis, Chris Fahlbusch, Lynda Fleming, Betty Luedke, Jeannine McConnell, Terry Nooyen-Coyner, Debbie Shyne, David Standerford, Donna Swaim, and Robin Tucker. The many comments I received from reviewers Ellen Gottesdiener, Andre Gous, and Shannon Jackson were especially valuable. A special thanks goes to Erik Simmons, who provided incisive suggestions on every chapter and greatly helped me sharpen the message. Thanks also to the Microsoft Learning editorial and production teams, including acquisitions editor Ben Ryan, copy editor Michelle Goodman, proofreaders Becka McKay and Sandi Resnick, and artist Joel Panchot. I especially enjoyed the opportunity to work with project editor Devon Musgrave again.

And finally, a big thank-you once again to my ever-cheerful wife, Chris Zambito. Fooled you this time, hon!



