

▶ **Book review**

Writing Better Requirements

by Ian F. Alexander and Richard Stevens

Addison-Wesley, 2002
ISBN: 0-932633-55-2
Cover price: US\$34.99
176 pages

Whether you are new to writing requirements or an old pro, this book is a must read. For novices, the authors provide a step-by-step methodology for writing requirements. For the pro, the layout of the requirements gathering and writing process is an excellent refresher. Do you want to achieve greater clarity when using common terms or improve the structure of your requirements document? This book has the required level of detail to help.

The first chapter covers the preliminaries and defines common terms such as user, customer, and different names used for requirements. It finishes with a description of requirements writing within the larger context of system development that is, unfortunately, buried in the text and treated too lightly. All too often, project teams do not really understand the discipline of requirements gathering and writing. This sometimes results in animosity from team members who mistakenly think their agendas conflict with that of the analysts and resent the amount of time these activities require. A separate chapter expanding upon the relationship between requirements and the rest of the development process would have helped readers understand that writing requirements is a crucial activity; it affects the entire project team as well as the quality of the product.

In subsequent chapters, the authors do a good job of describing a process for writing better requirements, providing useful techniques for each process step. The most difficult step in the requirements process is gathering proper, well-articulated requirements from stakeholders, and the techniques presented in the chapter on requirements gathering are particularly useful. Conducting interviews, holding workshops, and observing users at work are some of the techniques the authors identify and describe. In many cases users do not really know what their requirements are, so augmenting the interview or workshop with a prototype usually spurs their imagination. Regardless of the format interviewers chose for gathering requirements, the authors encourage

[▶ subscribe](#)[▶ contact us](#)[▶ submit an article](#)[▶ rational.com](#)[▶ issue contents](#)[▶ archives](#)[▶ mission statement](#)[▶ editorial staff](#)

them to keep asking the "Why?" question until the real requirement is revealed. Other useful questions are "So what?" and "Who cares?"

The exercises in these chapters are also well presented and well thought out -- every chapter has at least one exercise that assigns the reader tasks based on ideas and techniques covered in the text. For example, the exercises in Chapter 4 -- "Other Sources of Requirements" -- provide the reader with sample specifications. The task is to read the specification, extract statements from the text, and categorize these statements according to their type (User Requirements, System Specification, Design Elements, etc.). The first exercise identifies statements that need categorizing, and the second exercise is "free-form" -- the reader must pick out the statements for categorizing. Completing all of the exercises gives the user a practical way to learn and experience the process of gathering and writing requirements.

A chapter on how *not* to write requirements adds some humor to the book. For example, "The battery low warning lamp shall light up when the voltage drops below 3.6 volts, and the current workspace or input data shall be saved," is an example of including multiple requirements in one statement. (I don't think the two requirements are related, but you never know -- they just might be!) The example, "The fire alarm shall always be sounded when smoke is detected, unless the alarm is being tested or the engineer has suppressed the alarm," points out that "let-out" clauses, such as those that begin with "unless," can be dangerous. Actually, a fire alarm built to this requirement would be a real danger! (I am sure I have written a few laughable requirements in my lifetime, too.)

In addition to the few minor weaknesses I've cited, the authors include a single snippet of the European Space Agency's Software Development Standards but then do not reference it again. This left me frustrated. I felt they should either have provided more information and excerpts as an Appendix or simply not mentioned the standards at all.

Overall, however, I am confident that this book will help all who read it not only to improve their ability to write requirements but also to convey the importance of good requirements writing to other team members. I have found it particularly useful to give people examples of bad requirements. In our daily speech we use all sorts of constructs -- let-out clauses, wishful thinking, rambling, and so forth -- that we must avoid when writing requirements, and the examples this book provides are a great way to demonstrate the confusion they can cause.

As a product manager, I constantly write requirements documents; I know that I will use this book as a reference to gather and categorize them, and to lay them out clearly.

[-Peter Hanschke](#)

Rational Software Canada
IBM Rational Software Group