

Daniel M. Berry

## Aybüke Aurum and Claes Wohlin (eds): Engineering and managing software requirements

Published online: 25 January 2006  
© Springer-Verlag London Limited 2006

This book is a collection of chapters about various topics in requirements engineering (RE). Each chapter is written by one or two authors, at least one of which is well known in the topic of the chapter. Thus, each chapter is written by at least one expert. Indeed, as I was reading each chapter, I sensed the authority of the author and that the author had made good choices as to what to cover in the chapter. Each chapter describes the relevance of the topic of the chapter in RE, describes what has been done in the past and what is being done now. Usually, a chapter ends with some directions for future research in the topic.

The book begins with an introductory chapter setting the context for RE and describing the structure of the rest of the book. This chapter is followed by three parts:

- Part 1 is about the current art and practice in RE for traditional systems.
- Part 2 is about the art and practice of RE for various kinds of emerging systems and in emerging practices.
- Part 3 is about case studies about RE practice in industry.

The prime advantage of this book over traditional RE textbooks and earlier collections of papers is in Parts 2 and 3. Other books have not covered the same kinds of systems covered in Part 3 because these kinds of systems have emerged only too recently. Other books have not treated case studies in any depth, probably because the importance of the topic, of validating in practice what the research has shown, has been realized only recently.

The chapters and topics of Part 1 are:

2. Requirements elicitation
3. Specifications of requirements models

4. Requirements prioritization
5. Requirements interdependencies
6. Impact analysis
7. Requirements negotiation
8. Quality assurance in RE

The chapters and topics of Part 2 are:

9. Modeling goals and reasoning with them
10. Managing large repositories of natural language requirements
11. Understanding ambiguity in RE
12. Decision support in RE
13. Market-driven RE for software products
14. RE for agile methods
15. RE for Web-based information systems

The chapters and topics of Part 3 are:

16. A case of developing and managing quality SW systems in the public sector
17. Good quality requirements in the Unified Process
18. Requirements experience in practice: studies of six companies
19. An analysis of an empirical RE survey
20. RE: solutions and trends

I, as a reviewer, read every chapter. Each was a good survey of its topic. Each is written at the right level, so that I was neither bored from a lack of new information nor overwhelmed with a bombardment of new ideas. I got most excited about the chapters that touched on my own research interests and the chapters that presented topics about which I was quite unfamiliar. I will comment specifically on these chapters to give the reviewer a flavor of the chapters. The lack of coverage of any chapter should not be construed as a negative comment about the chapter. There is not enough room in this review to cover the entire book, and I had to find some criterion for deciding what to cover in this review.

*Chapter 2 (Requirements elicitation)* The section on Ethnography should cite the work of Joseph Goguen and Marina Jirotko, in particular their book titled

---

D. M. Berry  
Cheriton School of Computer Science, University of Waterloo,  
200 University Ave. West, Waterloo, ON, Canada, N2L 3G1  
E-mail: dberry@uwaterloo.ca  
Fax: +1-519-8884305  
URL: <http://www.se.uwaterloo.ca/~dberry>

*Requirements Engineering: Social and Technical Issues*, published by Academic Press, London, UK in 1994.

*Chapter 3 (Specifications of requirements models)* One rarely sees the kind of analysis of specification methods and languages that is found in Sect. 3.4.

*Chapter 4 (Requirements prioritization)* The running example on Sect. 4.7 is a nice example to make the chapter's necessarily brief description of techniques for ranking requirements by priority more concrete.

*Chapter 7 (Requirements negotiation)* The chapter's introduction says, "Identified disagreements represent major risks and need to be addressed by project management." There is no disagreement with this statement. However, it seems to me that *unidentified* disagreements represent an even greater risk.

*Chapter 8 (Quality assurance in RE)* The discussion of open research questions in Sect. 8.5 is particularly nice and forward looking. I recommend it to anyone doing a Ph.D. in the area of quality assurance of requirements specifications.

*Chapter 10 (Managing large repositories of natural language requirements)* I found particularly useful the criteria for tools dealing with natural language specifications. These criteria prevent a tool from being useless because it inundates the user with mostly useless information. A tool must provide the advantage of not overlooking anything important, while not presenting so much information that the user might well have just looked at the original document manually. In particular, Sect. 10.5.2 has a good explanation of usability of a tool.

*Chapter 11 (Understanding ambiguity in RE)* The most interesting idea in Chapter 11 is the recognition of a new class of ambiguity. Everyone is familiar with the four slightly overlapping categories of lexical, syntactic, semantic, and pragmatic ambiguities. The chapter introduces a new category, that of software engineering ambiguity, for sentences that are ambiguous without contextual information that should be provided by the specification in which the sentence is embedded. The chapter provides also several methods for surfacing ambiguities in a requirements specification; it describes empirical evaluations of how well these methods work.

*Chapter 12 (Decision support in RE)* This chapter was an eye opener. I was simply not aware of the distinction between decision support and decision making. The latter refers to the processing of finding the optimal solution to a problem on the assumption that it exists. The former refers to the process of finding a satisfactory solution to a problem, recognizing that an optimal solution does not exist or is not feasible to find. Because most problems requiring an RE process are wicked, RE needs decision support more than decision making. This

chapter is effectively advocating a new subarea of RE and an important one at that!

*Chapter 13 (Market-driven RE for software products)* This chapter on market-driven RE for mass-market software was another eye opener. This ever-more-and-more relevant topic has been largely ignored by the mainline RE research, which has focused on RE for bespoke systems. The chapter does a good job of describing the RE problems posed by mass-market software and RE techniques used in industry. It is a must read for those who have not been exposed to this topic before. An essential subprocess of market-driven RE is scoping of requirements into a doable consistent whole that will capture the market.

*Chapter 16 (A case of developing and managing quality SW systems in the public sector)* This chapter presents a case study of the development and quality management of software systems in the public sector. There are lots of useful lessons learned. The most interesting to me was the specific conclusion that "In some instances, organizations fail to recognize that building ICT systems, particularly those involving software, states long before any programmers are hired or code is cut." In other words, there is a whole requirements gathering and specification process before development can start.

*Chapters 18 (Requirements experience in practice: studies of six companies) and 19 (An analysis of an empirical RE survey)* These two chapters present case studies of RE practice that are chock-full of interesting data and conclusions about how helpful RE is to system development. These are useful to provide ammunition to convince an organization to improve its RE process. The first of these chapters examines six different projects and the second of these chapters analyzes survey data.

The only topic I can find that lacks but should have a chapter of its own is that of RE for systems built out of COTS and reused software. Parts of this topic are touched on briefly in several chapters, but I think it deserves a chapter by itself. What is special about such systems is that, even more than for systems built completely from scratch, RE cannot proceed strictly top down from requirements to specification and then on to implementation. Very often, one is willing to adjust requirements to match features that are already present in the COTS or reused software and to entirely avoid features that are not present in the COTS or reused software. A discussion of this process requires more than just the scattered passing treatment that COTS systems have in the book.

---

### Some general comments about all chapters

The future-directions section of most chapters is valuable to researchers seeking problems and topics.

The bibliographies of the chapters are valuable to identify sources of information outside the usual collection of sources in RE. There are items in social studies, experimentation, management, and information technology literature that are just not familiar to the average computer-science based RE researcher or practitioner. It would have been nice to have a common bibliography. As it is, the articles share many common articles. Having a common bibliography would save some pages. It would promote greater uniformity in the completeness of bibliographical items and would make for greater accuracy in that it is easier to check the accuracy of fewer items.

Unfortunately the quality of the contents is not matched by the quality of the final production. The copy editors need to have taken more care in putting the book together. There are lots of little typos and formatting glitches, but nothing terribly serious. If a revised version is done, these problems should be fixed.

Dear reader, do not get me wrong. If you want a good overview of RE, this book is well worth buying for its contents. Despite the little annoying problems in the book's final production, the chapters read very well and are loaded with useful information for both the practitioner and the researcher.



<http://www.springer.com/978-3-540-25043-2>

Engineering and Managing Software Requirements

Aurum, A.; Wohlin, C. (Eds.)

2005, XVIII, 478 p., Hardcover

ISBN: 978-3-540-25043-2