

# Reviews

## SOFTWARE ENGINEERING PROCESS

### Software Engineering Handbook

Jessica Keyes.

2003. Auerbach Publications:  
A CRC Press Company.  
(<http://www.auerbach-publications.com>). 874 pages.  
ISBN 0-8493-1479-8

CSQE Body of Knowledge  
area: Software Quality  
Management and Software  
Engineering Processes

Reviewed by John D. Richards  
[john.richards1@amedd.army.mil](mailto:john.richards1@amedd.army.mil)

This is a large, comprehensive volume that serves as an excellent reference tool for software engineers and managers of software projects. The author set quite a task for herself as put forth in the foreword:

"This book was written to push the information technology industry up that learning curve in one fell swoop. Collected here are 65 chapters, 191 illustrations, and 19 appendices filled with practical (the keyword here is practical) techniques, policies, issues, checklists, and guidelines, and complete "working" examples on methodology, quality, productivity, and reliability." (Page xvii).

This work was collected over the author's 25 years of experience as a consultant and professor of computer sciences. Her experience includes business and technology. She was managing director of research and development for the New York Stock Exchange and has been an officer with Banker's Trust and Swiss Bank Company. She has published more than 200 articles and is the author of 16 books on business issues and technology.

The handbook is divided into three sections. The first section

consists of 20 chapters on software engineering. The chapters cover a project from feasibility and cost/benefit analysis, through plan writing, requirement elicitation, and outsourcing decisions to testing and documentation. This section is very detailed, providing examples, charts, checklists, formulas, and so on necessary for running a project.

In the second section, in the author's words, "We change gears from method to metrics." The focus is on such things as productivity, quality, and reliability. This section consists of 45 chapters. It covers such topics as the Baldrige Award, Corbin's methodology for establishing a software development environment, Motorola's Six Sigma defect reduction effort, Byrne's reverse engineering technique, and the IEEE framework for measures, to name only a few. Each chapter has an abstract that further increases the value of this handbook as a reference.

The third section contains 19 appendices, providing examples of such things as: sample project plan, sample data dictionary, sample cost/benefit analysis worksheets, and test plan. All these assist the project manager in being efficient in his or her efforts, in that forms or templates do not have to be created but copied and populated.

This is not a volume that is normally read cover to cover but rather kept on the desk for frequent reference. It is well-organized, complete, easy to use, and an invaluable resource for project managers who do not want to "reinvent the wheel" but rather improve on it.

### Design for Trustworthy Software

Bijay K. Jayaswal

and Peter C. Patton.

2007. Pearson Education, Inc.  
(<http://www.pearsoned.com>).  
805 pages.

ISBN 0-13-187250-8

CSQE Body of Knowledge  
area: Software Engineering  
Processes

Reviewed by Scott Duncan  
[sduncan@westfullteam.com](mailto:sduncan@westfullteam.com)

In the preface, the authors state that "Any quality method employed to improve software reliability and hence trustworthiness [has] to be applied *as far upstream as possible*" since it is design, not manufacturing, that characterizes software. Therefore, their goal is to offer "a framework of tools, techniques, and methodologies . . . based on the principles of transformational leadership, best practices of learning organizations, management infrastructure, and quality strategy and systems" focused on software design. To do this, they have chosen to apply Taguchi methods "used in the context of other upstream customer-oriented methods, such as analytical hierarchical process (AHP), quality function deployment (QFD), TRIZ, Pugh concept selection, and failure mode and effects analysis (FMEA), all of which may be applied before a single line of code is written." (It should be noted, however, that at other places, the authors say their design for trustworthy software (DFTS) approach uses things like "the iterative robust software development model, software design optimization engineering, and object-oriented design technology" to address "producing trustworthy software.")

The book is divided into five parts and 27 chapters:

- Part I: Covers contemporary software development process, including life-cycle models, an introduction to Taguchi methods, software quality metrics, financial perspectives (that is, cost of software quality (CoSQ)), and organizational leadership.