

7 of 7 people found the following review helpful:

★★★★★ **Comprehensive**, January 15, 2001

Reviewer: [B. Scott Andersen](#) (Acton, MA USA) - [See all my reviews](#)

REAL NAME

I wish I could count the times I've seen second and third rounds of development occur because the first round produced a working version of a product the customer didn't want. With all of the emphasis lately on rapid development, especially now that the web has everyone working on "internet time", there has been a noticeable lack of discussion on ensuring the software produced fits the needs of the customer and is of reasonable quality.

Rakitin addresses these issues and more in this book. As the title indicates, he concentrates on Verification ("are we building the product right?") and Validation ("are we building the right product?"). However, the subtitle to the work "A Practitioner's Guide" provides much more insight into the actual scope of this work. In the discussion of software inspection meetings, for example, Rakitin give guidelines regarding not only the mechanics of who should attend and when materials should be distributed but he also provides insight into what to expect as a moderator and how much should be expected to be accomplished in the meetings themselves. There are a number of statements in the book that begin "Experience has shown..." Rakitin's extensive experience has manifested itself throughout the book transforming the dry, checklist-like discussions found in so many other books into discussions about how people work and communicate with each other.

This isn't to say there couldn't be more. Although what's presented is very good, there are points in the book where I found myself wishing for additional discussion. Perhaps in future editions Rakitin will be able to expand upon, say, requirements collection or configuration management.

There are also things that could be updated if the book were to have a revision. For example, a brief discussion on OO methodologies is provided where Fusion from HP is outlined. This could obviously be expanded to cover the Rational Unified Process, Rational's effort to provide UML with "meat" the modeling language alone could not have.

As Deming observed and Rakitin noted, "The quality of a product is directly related to the quality of the process used to create it." To this end, Rakitin attempts to provide the reader with ready-made tools, checklists, outlines, and forms to aid them in the maturation of their software engineering department. These items, which appear in approximately 80 pages of appendices, give the reader a variety of starting places for just such an initiative.

Brooks said "no silver bullet" and he was right. Quality software is possible only through a methodical, rational, and scientific approach. Rakitin goes a long way towards that in this work. I highly recommend it.

Was this review helpful to you? yes no yes no [\(Report this\)](#) [\(Report this\)](#)

7 of 7 people found the following review helpful:

★★★★☆ **Practical Book giving practical approach to complex subject**, June 14, 2000

Reviewer: [John P. Rooney "John Peter"](#) (Plymouth, MA USA) - [See all my](#)

[reviews](#)
REAL NAME

Software Verification and Validation by Steven R. Rakitin. A practical book by a practical author, presenting the steps required to set up a formal and complete Verification and Validation program for software production. Mr. Rakitin sprinkles pertinent quotations throughout the book to support his case that software quality programs (detection) cost less than simply releasing a "buggy" program to your Customers. The author also teaches the lessons of this book at IEEE lectures; I attended one in March 1998. At these lectures, the author fleshes out and makes more interesting many of the points made in his book. As an ASQ Certified Software Engineer, S. Rakitin has the credentials to expound on many aspects of quality programs set up to deal effectively with the major & crucial issue of the lack of software quality today. There are 169 pages of text, describing a good program, but the author has also included some 87 pages of appendices (from "A" to "H") which give you a cook book or recipe approach to different aspects of software quality. The author gives you good leads to other sources of information on software quality. The book would profit from another good editorial review, reducing the wordiness in some chapters and tying the whole story all together. For example, Rakitin continuously uses "SEI" which (as far as I can see) is not defined in his book. SEI = Software Engineering Institute. In Chapter 4, the author attempts to establish a cost/benefit justification for implementing a software quality program, but misses and then, later, on page 90 (Chapter 7), he DOES lists the pertinent reasons as, "...lower support costs, ... fewer maintenance releases, ... higher customer satisfaction and, as a result, increased sales". An astute editor would have melded all of this together. Overall, this book is directed at the practitioner, whether a hardware quality engineer required to set up a software quality program, or a novice who has the same task. It is a practical book, generally well written,