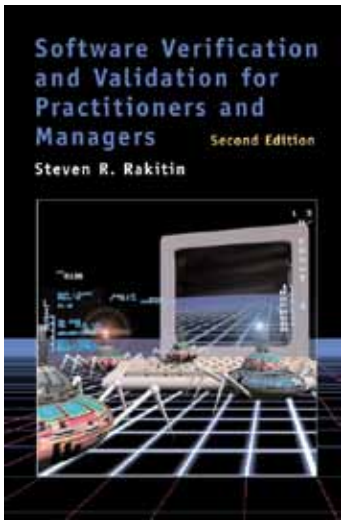
Software Verification and Validation for Practitioners and Managers

Second Edition, by Steven R. Rakitin

Review for Second Edition



Bill Gates is famous for his yearly retreats where he takes a pile of books and reads them. Maybe that's the one thing today's managers should emulate. If you do, bring this book along on your first sabbatical.

This all new edition of the book (first published in 1997) was even renamed to include its new, extended charter. "Clearly management must take a leadership role in helping the organization behave in a more predictable way. It is for this reason that the title of the book has been changed to include managers. [the book] includes specific actions that managers can take to help organizations behave in a more predictable manner."

Software development is a difficult disciple to master. Even so, technical acumen is no guarantee of quality. "As observed by Dr. Edwards Deming, 'The quality of a product is directly related to the quality of the process used to create it.'" Verification ("are we building the product right?") and validation ("are we building the right product?") are the first questions one must ask to begin the path of process improvement. This book addresses those questions and more.

The book is divided into four large sections. The first three sections are brought forward from the previous edition with some noticeable improvements. The new edition is not just a rehash of the old stuff, however. Over 30 new pages of appendices and an entire new section of the book aimed at management with over 70 new pages have been added.

Other improvements are also evident including better formatting. For example, the font selection and layout are much easier on the eyes than in the previous edition. If you own the first edition, this is a "must have" update. If you don't own the first edition, consider this work an SQA department "starter kit." It is well organized, well annotated, and filled with practical artifacts such as checklists for inspections, suggested document outlines, and the like. This is stuff you can use.

Chapter 12 is specifically for the CEO and should be required reading by anyone running a company with a software development function. "Managers and executives need to understand that having a predictable software development process is vitally important to the long-term success of their business." Rakitin shows how, then shows what can be done. The prose is crisp and to-the-point. Well done.

Review for First Edition

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 gives 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.