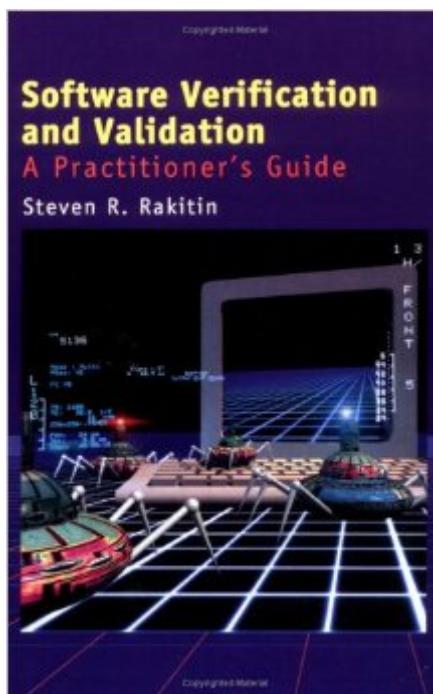


The book was found

Software Verification And Validation: A Practitioner's Guide (Artech House Computer Library (Hardcover))



Synopsis

This book highlights the advantages and disadvantages of various software development lifecycle models, and describes when to apply testing -- and when to use other, more cost-effective techniques. It also shows how to incorporate V&V techniques if your organization does not have a written procedure, and explains how to implement the inspection process.

Book Information

Series: Artech House Computer Library (Hardcover)

Hardcover: 296 pages

Publisher: Artech House Publishers (February 1, 1997)

Language: English

ISBN-10: 0890068895

ISBN-13: 978-0890068892

Product Dimensions: 6 x 0.8 x 9 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 4.0 out of 5 starsÂ See all reviewsÂ (2 customer reviews)

Best Sellers Rank: #3,044,655 in Books (See Top 100 in Books) #66 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control #1207 inÂ Books > Computers & Technology > Computer Science > Systems Analysis & Design #2694 inÂ Books > Textbooks > Computer Science > Operating Systems

Customer Reviews

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, ...

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.

[Download to continue reading...](#)

Software Verification and Validation: A Practitioner's Guide (Artech House Computer Library (Hardcover)) Software Verification and Validation for Practitioners and Managers, Second Edition Verification and Validation of Modern Software-Intensive Systems Practical Guide to Software Quality Management (Artech House Computer Science Library) Integrated Microwave Front-Ends with Avionics Applications (Artech House Microwave Library (Hardcover)) Software Fault Tolerance Techniques and Implementation (Artech House Computing Library) Broadband Networking ATM, Adh and SONET (Artech House Telecommunications Library) Convolutional Coding: Fundamentals

and Applications (Artech House Communications Library) Codes for Error Control and Synchronization (Artech House Communication & Electronic Defense Library) Agile Systems with Reusable Patterns of Business Knowledge: A Component-Based Approach (Artech House Computing Library) The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP Systems and Software Verification: Model-Checking Techniques and Tools Verification of Computer Codes in Computational Science and Engineering Family Psychiatric & Mental Health Nurse Practitioner Exam Flashcard Study System: NP Test Practice Questions & Review for the Nurse Practitioner Exam (Cards) The High-Conflict Couple: A Dialectical Behavior Therapy Guide to Finding Peace, Intimacy, and Validation The Validation of Risk Models: A Handbook for Practitioners (Applied Quantitative Finance) Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Tiny Houses: Tiny House Plans, Woodworking on a Tiny House and Living Mortgage Free (Tiny Houses, Tiny House Living, Tiny House Plans, Small Homes, Woodworking Book 1) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection Python: Python Programming For Beginners - The Comprehensive Guide To Python Programming: Computer Programming, Computer Language, Computer Science

[Dmca](#)