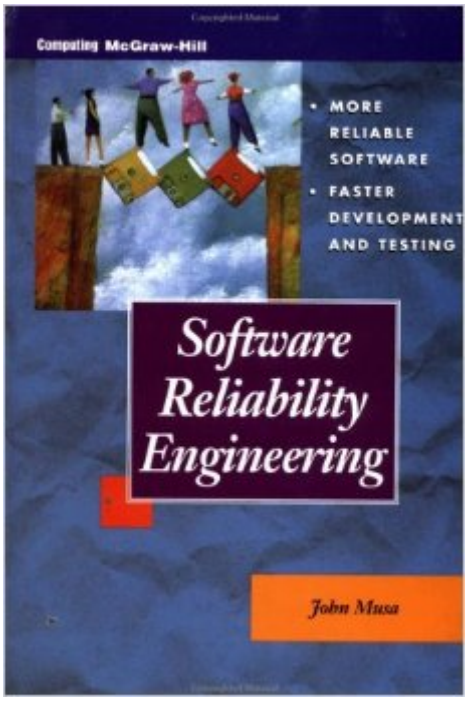
Software Reliability Engineering



Synopsis

This hands-on tutorial shows how to develop tests that ensure the reliability of software systems. Software developers learn how to establish reliable objectives, develop operational profiles, and prepare and execute test cases, as well as about useful formulae and recommended software tools. FAQ sections in each chapgter provide a useful way to review or reference specific information, and practical exercises allow readers to immediately apply what they have learned. John D. Musa is the author of "Software Reliability: Measurement, Prediction, Application".

Book Information

Series: McGraw Hill Series on Software Development

Hardcover: 384 pages

Publisher: Osborne/McGraw-Hill (July 31, 1998)

Language: English

ISBN-10: 0079132715

ISBN-13: 978-0079132710

Product Dimensions: 9.4 x 6.3 x 1.3 inches

Shipping Weight: 1.6 pounds

Average Customer Review: 5.0 out of 5 starsÂ Â See all reviewsÂ (3 customer reviews)

Best Sellers Rank: #1,848,776 in Books (See Top 100 in Books) #41 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control #4975 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development #11456 inÂ Books > Computers & Technology > Programming > Languages & Tools

Customer Reviews

This book takes Mr. Musa’s out-of-print classic, Software Reliability: Measurement, Prediction, Application, to the next level. Where his first book spawned an entire body of knowledge and approach to software reliability, this one adds a structured process and extends the foundation provided by the original book into a discipline that is practiced by mature organizations.The process, called SRET or software reliability engineered testing, is six-step model comprised of the following steps:(1) List associated systems - includes base products and variations to identify scope and coverage.(2) Develop operational profiles - break the system down into logical tasks and rate of occurrence (expressed as probabilities)(3) Define "just right" reliability - this is the tough part and is thoroughly covered. The essential elements of this step include: determining failure (discrepancy

between system behavior and user requirements) and faults (system implementation defects that trigger failures). You next determine the "just right" level of reliability by determining a strategy for measuring failure intensities. This is done by computing a failure intensity objective (FIO) for each system. Brush up on probability and statistics for this step because it is performed using hard quantitative methods.(4) Prepare for testing - this is the traditional approach employing a test plan and associated test cases, with a distinct difference: the test cases are tied to operational profiles, breaking down a complex process into manageable elements. The more complex the software being tested the more manageable the test process becomes using this structured approach.