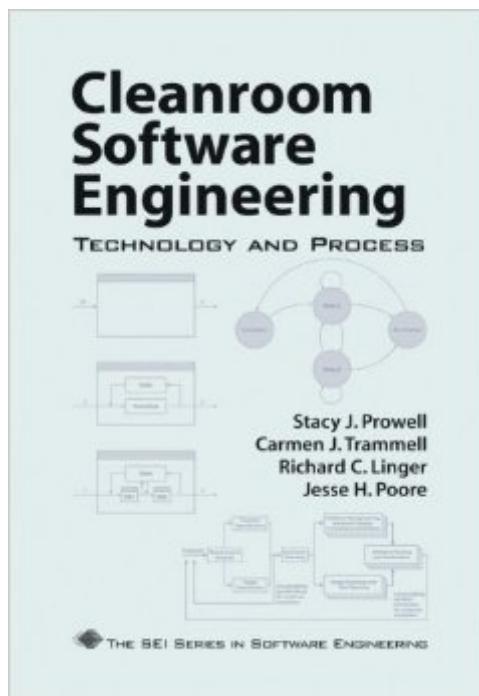


The book was found

Cleanroom Software Engineering: Technology And Process



Synopsis

Cleanroom software engineering is a process for developing and certifying high-reliability software. Combining theory-based engineering technologies in project management, incremental development, software specification and design, correctness verification, and statistical quality certification, the Cleanroom process answers today's call for more reliable software and provides methods for more cost-effective software development. Cleanroom originated with Harlan D. Mills, an IBM Fellow and a visionary in software engineering. Written by colleagues of Mills and some of the most experienced developers and practitioners of Cleanroom, *Cleanroom Software Engineering* provides a roadmap for software management, development, and testing as disciplined engineering practices. This book serves both as an introduction for those new to Cleanroom and as a reference guide for the growing practitioner community. Readers will discover a proven way to raise both quality and productivity in their software-intensive products, while reducing costs. Highlights Explain basic Cleanroom theory Introduces the sequence-based specification method Elaborates the full management, development, and certification process in a Cleanroom Reference Model (CRM) Shows how the Cleanroom process dovetails with the SEI's Capability Maturity Model for Software (CMM) Includes a large case study to illustrate how Cleanroom methods scale up to large projects.

Book Information

Paperback: 416 pages

Publisher: Addison-Wesley Professional; 1 edition (March 19, 1999)

Language: English

ISBN-10: 0201854805

ISBN-13: 978-0201854800

Product Dimensions: 6.2 x 1 x 9.1 inches

Shipping Weight: 14.4 ounces (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 starsÂ See all reviewsÂ (7 customer reviews)

Best Sellers Rank: #1,190,178 in Books (See Top 100 in Books) #26 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control #1490 in Books > Textbooks > Computer Science > Software Design & Engineering #1968 in Books > Computers & Technology > Certification

Customer Reviews

Frequently, software development methods describe concepts in a way that suggests that one can

get better only by divine inspiration. Thankfully, this book's premise is that software development can be done in a deterministic and algorithmic fashion, rather than a heuristic which some are better at applying than others. The importance of this point is that while some people design software much like artists make paintings, people can be trained in Cleanroom Engineering technology. For 95% of all software we don't need artists. Sadly, current software development methods assume they are continuously available. My background is in controls engineering, both hardware and software, and when things machines break due to an error in its control, there is frequent economic loss and, regrettably but occasionally, loss of life. So, in order to avoid these things engineers in controls development do effectively the same thing described in this book. We develop in a stepwise fashion while always proving the implementation in the small before integrating it in the large. I learned of Cleanroom Engineering in 1994 from the STARS project. I formalized my controls engineering to the techniques identified in that literature to great success. When I entered software design and engineering as a full-time effort, most of my colleagues and fellow employees thought I was nuts when I developed software using Cleanroom Engineering. However, my software always arrived on time, without defect, and well reused. Cleanroom Software Engineering identifies the necessary techniques to deliver zero-defect software.

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

Cleanroom Software Engineering: Technology and Process Measuring the Software Process: Statistical Process Control for Software Process Improvement Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Non-Functional Requirements in Software Engineering (International Series in Software Engineering) Modelling Techniques for Business Process Re-engineering and Benchmarking (IFIP Advances in Information and Communication Technology) Software Process Design: Out of the Tar Pit (Mcgraw-Hill International Software Quality Assurance) Software Components With Ada: Structures, Tools, and Subsystems (The Benjamin/Cummings Series in Ada and Software Engineering) Software Reuse: Guidelines and Methods (Software Science and Engineering) Software Failure: Management Failure: Amazing Stories and Cautionary Tales (Wiley Series in Software Engineering Practice) Error-Free Software: Know-How and Know-Why of Program Correctness (Wiley Series in Software Engineering Practice) Constraint-Based Design Recovery for Software Reengineering: Theory and Experiments (International Series in Software Engineering) Object-Oriented Software Engineering: Practical Software Development Using UML and Java Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for

Software Protection Global Software Development Handbook (Applied Software Engineering Series) Re-Engineering Software: How to Re-Use Programming to Build New, State-of-the-Art Software Software Architecture in Practice (3rd Edition) (SEI Series in Software Engineering) Practical Software Reuse (Wiley Series in Software Engineering Practice) Object-oriented software development: Engineering software for reuse Enterprise Software Platform: A Textbook for Software Engineering Students ADA as a Second Language (McGraw-Hill series in software engineering and technology)

[Dmca](#)