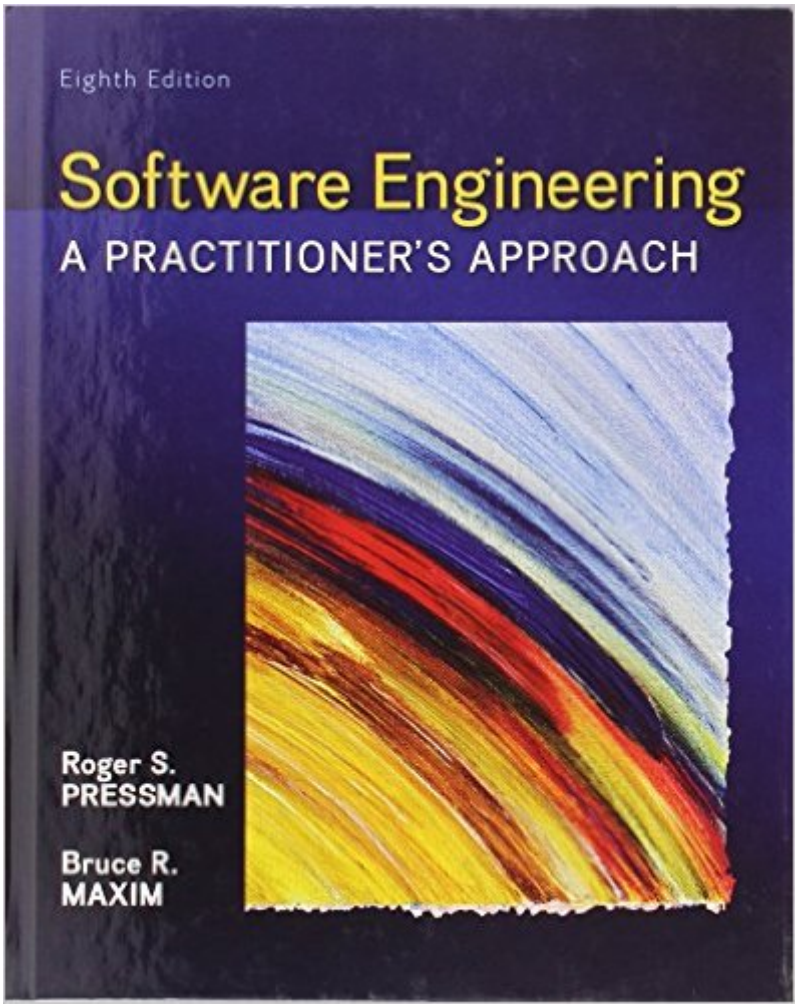
Software Engineering: A Practitioner's Approach



Synopsis

For almost three decades, Roger Pressman’s Software Engineering: A Practitioner’s Approach has been the world’s leading textbook in software engineering. The new edition represents a major restructuring and update of previous editions, solidifying the book’s position as the most comprehensive guide to this important subject. The chapter structure will return to a more linear presentation of software engineering topics with a direct emphasis on the major activities that are part of a generic software process. Content will focus on widely used software engineering methods and will de-emphasize or completely eliminate discussion of secondary methods, tools and techniques. The intent is to provide a more targeted, prescriptive, and focused approach, while attempting to maintain SEPA’s reputation as a comprehensive guide to software engineering. The 39 chapters of this edition are organized into five parts - Process, Modeling, Quality Management, Managing Software Projects, and Advanced Topics. The book has been revised and restructured to improve pedagogical flow and emphasize new and important software engineering processes and practices.McGraw-Hill Education’s Connect, is also available as an optional, add on item. Connect is the only integrated learning system that empowers students by continuously adapting to deliver precisely what they need, when they need it, how they need it, so that class time is more effective. Connect allows the professor to assign homework, quizzes, and tests easily and automatically grades and records the scores of the student’s work. Problems are randomized to prevent sharing of answers an may also have a "multi-step solution" which helps move the students’ learning along if they experience difficulty.

Book Information

Hardcover: 976 pages

Publisher: McGraw-Hill Education; 8 edition (January 23, 2014)

Language: English

ISBN-10: 0078022126

ISBN-13: 978-0078022128

Product Dimensions: 7.5 x 1.9 x 9.4 inches

Shipping Weight: 3.6 pounds (View shipping rates and policies)

Average Customer Review: 3.4 out of 5 starsÂ Â See all reviewsÂ (16 customer reviews)

Best Sellers Rank: #73,367 in Books (See Top 100 in Books) #93 inÂ Books > Textbooks > Computer Science > Software Design & Engineering #195 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development #339

Customer Reviews

Bloated, so bloated. The authors seem to say the same things over and over again, using slightly different "keywords" which all blend together and "methodologies" which are basic common sense to anyone with a grade school education.. The fact that the other five reviews are all five stars blows my mind. I'm convinced they are almost entirely fake, considering the 7th edition of this book is currently rated somewhere around 1.5 stars and the books are nearly identical minus a few additions about mobile development and some chapter reshuffling. I feel very sorry for anyone who is unlucky enough to have this as a required text for a Software Engineering course, myself included.

There's really not any software engineering in this book. The fact that it's somehow made it to an 8th edition is simply marketing magic. You'd learn more software engineering from reading the wikipedia page on the topic, or from the Tale of Two Systems in Pete Goodliffe's book on Becoming a Better Programmer, or from How Linux Works, or from the Linux Mint Tutorial section on what a package management system does. Or really from just taking an introduction to programming course. Not worth the money (or even close). Engineering is about doing something. Nothing gets done in this book. Nothing at all.

This is the most BORING, REPETITIVE and REDUNDANT book you can get. Pressman must believe that writing the same thing repeatedly makes the book fatter and better, which is absolutely wrong. The text is so "academic", to put it some way, that it appears this guy has never ever written a single line of code. Sadly my university uses it for two semesters, and so I had to suffer it. I would give it 0 stars if allowed it. I can not comprehend how this book got to is 8th edition. About the 5 star reviews, they are probably fake. Most of them have zero actual review content.

This book is a simple scan of the hard copy book. It's just a bunch of images in a PDF file. I made the mistake of renting this book and there doesn't seem to be a way to return the book. The book is completely useless. Never, never buy a McGraw-Hill etext. You can't read the text on a small screen, like a phone. You can't adjust the text size. It's easily the worst purchase I have made on . I'm pissed off for paying $90 for this piece of crap. You can get the same quality by pirating the book off the internet.

This is a terrible book which makes the implicit claim that software engineering is a much more organized discipline than it actually is, and most likely than it ever can be. I originally gave it one star, but bumped it to two, simply because I'm not sure there is a better book available. Yet, this doesn't make the problem go away.Software engineering, to put it in computer science terms, is carried out in the world of natural language rather than formal language. There are limits to the precision of the meaning of words, and limits to the precision with which formal processes can define human interactions. This would not be a problem if the book accepted these limits, but it attempts to push well beyond them. Pretending to say more than we can say about something results in saying less than we could have said if we hadn't pretended. The book follows the approach of defining words in terms of other words which are themselves badly defined, or promised to be defined later (a promise which is rarely kept), or are self-referential (architecture is architectural). And then, maybe, another model will be introduced which defines the same words differently (demonstrating the words don't really have a particular meaning) or uses different words for the same things, so that terms don't even have consistent meaning across a chapter. Then, these terms are used to discuss processes as if they've been rigorously defined, when they really haven't been. This leaves us pretending to have an engineering discussion with the precision of equations when we are having at best a social discussion with much less precision than that. If you enjoy reading well-built arguments you will constantly find jarring transitions in this book, along the lines of "it follows that," when quite simply, it does NOT follow.And then this is made worse by the fact that this is a modern textbook, with a modern test bank, in which it's assumed that the *correct* answer depends on the exact keywords the author used on page 391, despite the fact that even the author would certainly admit that this is just one way of looking at the thing.As I already said, there might not be a better effort available, but that doesn't mean this book isn't bad. In fact, it's bad enough that taking the class associated with this book made me want to quit studying computer science or even leave college. I won't do that, because the cost of doing so is too great in terms of what I need to accomplish for other reasons, but I view reading this book almost entirely as a cost to be borne in the service of a greater good. Someone, for Gods's sake, please write a better one.

Horrible... it's just the worst. Sucks all the fun and life out of my soul. A few sentences are enough to make your eyes glaze over. Waste of a semester's worth of reading.

The book itself was alright for the material being taught. The case study story that ran throughout

the book helps pull things together. Avoid renting this as an eBook, either get the hardback or softback book. The eBook was not designed or reflowed to be used on a e-reader. The experience of reading this book on an eReader was comparable to viewing a pdf file on a small screen, having to zoom in and out of pages to read it.

Required for a course I am taking. Well written and easy to follow. Some topics could use a bit more depth but most are very detailed. This is a survey level textbook in practice. The authors long experience as a lecturer and educator makes this book a very strong resource.

Software Engineering: A Practitioner's Approach Non-Functional Requirements in Software Engineering (International Series in Software Engineering) Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP Family Psychiatric & Mental Health Nurse Practitioner Exam Flashcard Study System: NP Test Practice Questions & Review for the Nurse Practitioner Exam (Cards) Software Components With Ada: Structures, Tools, and Subsystems (The Benjamin/Cummings Series in Ada and Software Engineering) Global Software Development Handbook (Applied Software Engineering Series) 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) 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 Software Reuse: Guidelines and Methods (Software Science and Engineering) Enterprise Software Platform: A Textbook for Software Engineering Students 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 Software Verification and Validation: A Practitioner's Guide (Artech House Computer Library (Hardcover)) The Renaissance of Legacy Systems: Method Support for Software-System Evolution (Practitioner Series)