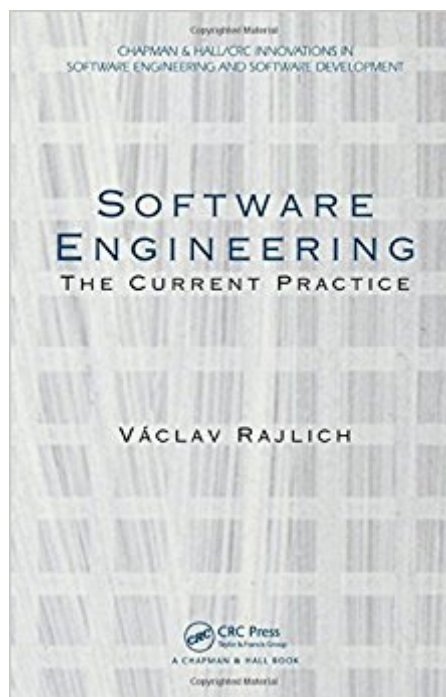The book was found

Software Engineering: The Current Practice (Chapman & Hall/CRC Innovations In Software Engineering And Software Development Series)

Synopsis

Software Engineering: The Current Practice teaches students basic software engineering skills and helps practitioners refresh their knowledge and explore recent developments in the field, including software changes and iterative processes of software development. After a historical overview and an introduction to software technology and models, the book discusses the software change and its phases, including concept location, impact analysis, refactoring, actualization, and verification. It then covers the most common iterative processes: agile, directed, and centralized processes. The text also journeys through the software life span from the initial development of software from scratch to the final stages that lead toward software closedown. For ProfessionalsThe book gives programmers and software managers a unified view of the contemporary practice of software engineering. It shows how various developments fit together and fit into the contemporary software engineering mosaic. The knowledge gained from the book allows practitioners to evaluate and improve the software engineering processes in their projects. For InstructorsInstructors have several options for using this classroom-tested material. Designed to be run in conjunction with the lectures, ideas for student projects include open source programs that use Java or C++ and range in size from 50 to 500 thousand lines of code. These projects emphasize the role of developers in a classroom-tailored version of the directed iterative process (DIP). For StudentsStudents gain a real understanding of software engineering processes through the lectures and projects. They acquire hands-on experience with software of the size and quality comparable to that of industrial software. As is the case in the industry, students work in teams but have individual assignments and accountability.

Book Information

Series: Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series

Hardcover: 315 pages

Publisher: Chapman and Hall/CRC; 1 edition (November 17, 2011)

Language: English

ISBN-10: 1439841225

ISBN-13: 978-1439841228

Product Dimensions: 9.4 x 6.2 x 0.8 inches

Shipping Weight: 1.3 pounds (View shipping rates and policies)

Average Customer Review: 4.8 out of 5 stars   7 customer reviews

Best Sellers Rank: #407,543 in Books (See Top 100 in Books)   #116 inÂ Books > Textbooks > Computer Science > Algorithms   #271 inÂ Books > Computers & Technology > Programming > Algorithms   #503 inÂ Books > Textbooks > Computer Science > Software Design & Engineering

Customer Reviews

"â | a great read â | this [is] an entirely different approach to teaching software engineering and it could really help students (and practitioners) understand recent advances in software engineering and become better software engineers. â | this book explains software engineering not from a constructionist point of view, but from a change/maintenance perspective, meaning most of the time you need to read/analyze programs rather than write them (though there is plenty of material in the book to support green field development)."â •Will Tracz, ACM SIGSOFT Software Engineering Notes, November 2013

VÃ¡clav Rajlich is a professor and former chair of computer science at Wayne State University. Dr. Rajlich is an editorial board member of the Journal of Software Maintenance and Evolution and the founder and permanent steering committee member of the IEEE International Conference on Program Comprehension (ICPC). His research focuses on software evolution and comprehension.

Actually got to take my software engineering class with the author of this book. It is easy to read and has a great flow. There is a lot of information to absorb, but the ideas are presented in a clear concise manner. This is a great text book!!!

The book begins with an introduction to process models and a difference between traditional and agile methodologies, then moves to an explanation of each step in the model as it pertains to agile to later wrap up with a couple chapters with examples using the entire process. The author does not go into a great level of detail of each process, but does provide simple examples by painting a picture in the reader's mind and providing diagrams that accompany the text. Challenge questions are at the end of each chapter, accompanied by referenced material in "Further Reading" sections that include other books and papers to further expand one's knowledge on each of the topics covered.

This is a real INTRODUCTION book, any newcomer to the software engineering field will be pleased.

great book! for software engineering

I have been using this text since it was in preprint, and should have written this review sooner. This is the text that I would have written, but now there is no need! I hold that there really is no such thing as development; only maintenance and evolution on the empty system. Given that assumption, this text shows students what their professional practice will be like. Using its directions and such for projects, I have been able to put together a sophomore level software engineering course that students actually can reference in job interviews. ["I was part of a team that performed successful maintenance and evolution on a mature 200K LOC SourceForge project. The entire project was under configuration control. ..."]Here is why I recommend no changes, and why I unreservedly recommend it for an introductory software engineering course. The students in my classes have heard the term "software engineering', but have no idea what it means. This -text- presents the basics to the uninitiated and permits a substantial team project. That is its purpose and it does so handsomely. I put the dashes around 'text' above because once a student masters the material in the text, there is no further need of it; it won't be kept next to a workstation for reference. That is OK imnsho -- in fact much better than OK! Students can read it and understand it, as many have commented to me. The chapter organization fits nicely into a semester and goes hand-in-hand with an assigned project. To add more material would steal from its essential pedagogical purpose, which is to present the basics of software engineering; there is time to dig deeper into other topics in other courses -once the students have some basic ideas about what is going on in real projects.

The book is excellent for the topic of evolutionary software development. I used it in combination with a real-world software evolution project in a one semester capstone project class. The text is perfect for one semester at the undergraduate level. The slides, problems and answer keys provided with the book are very helpful for preparing my own lectures. Some students express the desire to have a true design oriented project rather than evolutionary ones like what I gave them. I will have to think about how to address that concern in future. For the semester I got the highest student evaluation in 9 years of teaching.

I have used this book in teaching software engineering this semester (Winter 2012). I was looking for a suitable software engineering book for a long time, and this one finally has what I need. It covers most topics that I can think of for a software engineering course. I particularly like the

emphasis on iterative software development, and the combination of concrete code examples with the high-level notions.As I was teaching an advanced course on Software Engineering (fourth year undergrad + grad students), I have had many other materials to cover. However, still I covered most part of this book in my lectures. It was enjoyable in reading (and teaching) the book. The provided slides were also great. One doesn't possibly need to spend a lot of efforts in updating/making the slides on their own (if wishes). The assignments and projects, and the solutions provided were also very helpful. The most useful part of this book is the "Software Change", where you learn all the different aspects of software change in a nutshell but also in a comprehensive way with appropriate references. The other part excellent is the "software process", where the author extensively talked about the iterative software processes. The book also provides several example scenarios of the software processes. All these together provided a comprehensive and enjoyable introduction to Iterative Software Development, which is one of the state of the art Software Development processes. Great work Vaclav. However, for the next edition you may want to add a few more things such as software design, design patterns, more on testing, software architecture and so on.

Software Engineering: The Current Practice (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series) Flexible, Reliable Software: Using Patterns and Agile Development (Chapman & Hall/CRC Textbooks in Computing) Using R for Numerical Analysis in Science and Engineering (Chapman & Hall/CRC The R Series) Access Control, Security, and Trust: A Logical Approach (Chapman & Hall/CRC Cryptography and Network Security Series) Handbook of Financial Cryptography and Security (Chapman & Hall/CRC Cryptography and Network Security Series) Data Classification: Algorithms and Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series) Introduction to Scientific Programming and Simulation Using R (Chapman & Hall/CRC The R Series) Variational Methods in Image Processing (Chapman & Hall/CRC Mathematical and Computational Imaging Sciences Series) Introduction to Scientific Programming and Simulation Using R, Second Edition (Chapman & Hall/CRC The R Series) Introduction to Modern Cryptography, Second Edition (Chapman & Hall/CRC Cryptography and Network Security Series) Statistical Computing with R (Chapman & Hall/CRC The R Series) Introduction to Stochastic Processes (Chapman & Hall/CRC Probability Series) A Concise Introduction to Pure Mathematics, Fourth Edition (Chapman Hall/CRC Mathematics Series) Analyzing Baseball Data with R (Chapman & Hall/CRC The R Series) Introductory Fisheries Analyses with R (Chapman & Hall/CRC The R Series) Statistics and Data Analysis for Microarrays Using R and Bioconductor, Second Edition (Chapman & Hall/CRC Mathematical and Computational

Biology) Measure and Integral: An Introduction to Real Analysis, Second Edition (Chapman & Hall/CRC Pure and Applied Mathematics) Introduction to Set Theory, Third Edition, Revised and Expanded (Chapman & Hall/CRC Pure and Applied Mathematics) Introduction to Proteins: Structure, Function, and Motion (Chapman & Hall/CRC Mathematical and Computational Biology) An Introduction to Systems Biology: Design Principles of Biological Circuits (Chapman & Hall/CRC Mathematical and Computational Biology)