Software Engineering: Theory and Practice (2nd Edition)

By Shari Lawrence Pfleeger

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger

This thorough book blends concepts with real, practical applications by providing an abundance of examples. Dr. Pfleeger applies state-of-the-art software engineering methods consistently to two case studies that are integrated throughout: a realistic information system and a requirements rich real-time system. *NEW -Chapter 6: Considering Objects- Significantly expands the coverage of object-oriented development, a programming perspective being implemented by many companies. *NEW -Chapter 14: The Future of Software Engineering- Discussion of the importance of technology transfer and the role of decision-making in software engineering. *NEW -Term Project-This hands-on project weaves the concepts of the book into a tangible example. The author applies concepts to real world settings at the macro level for development teams and at the micro level for individual developers.

Software Engineering: Theory and Practice (2nd Edition)

By Shari Lawrence Pfleeger

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger

This thorough book blends concepts with real, practical applications by providing an abundance of examples. Dr. Pfleeger applies state-of-the-art software engineering methods consistently to two case studies that are integrated throughout: a realistic information system and a requirements rich real-time system. *NEW -Chapter 6: Considering Objects- Significantly expands the coverage of object-oriented development, a programming perspective being implemented by many companies. *NEW -Chapter 14: The Future of Software Engineering- Discussion of the importance of technology transfer and the role of decision-making in software engineering. *NEW -Term Project-This hands-on project weaves the concepts of the book into a tangible example. The author applies concepts to real world settings at the macro level for development teams and at the micro level for individual developers.

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger Bibliography

- Rank: #3364524 in Books
- Published on: 2001-02-07
- Original language: English
- Number of items: 1
- Dimensions: 9.24" h x 1.41" w x 7.26" l, 2.47 pounds
- Binding: Hardcover
- 659 pages

Download Software Engineering: Theory and Practice (2nd Edi ...pdf

Read Online Software Engineering: Theory and Practice (2nd E ...pdf

Editorial Review

From the Inside Flap
Preface BRIDGING THE GAP BETWEEN RESEARCH AND PRACTICE

Software engineering has come a long way since 1968, when the term was first used at a NATO conference. And software itself has entered our lives in ways that few had anticipated, even a decade ago. So a firm grounding in software engineering theory and practice is essential for understanding how to build good software and for evaluating the risks and opportunities that software presents in our everyday lives. This text represents the blending of the two current software engineering worlds: that of the practitioner, whose main focus is to build high-quality products that perform useful functions, and that the researcher, who strives to find ways to improve the quality of products and the productivity of those who build them.

Designed for an undergraduate software engineering curriculum, this book paints a pragmatic picture of software engineering research and practices. Examples speak to a student's limited experience but illustrate clearly how large software development projects progress from need to idea to reality.

The book is also suitable for a graduate course offering an introduction to software engineering concepts and practices, or for practitioners wishing to expand their knowledge of the subject. It includes examples that represent the many situations readers are likely to experience: large projects and small, object-oriented and procedural, real-time and transaction processing, development and maintenance. In particular, Chapters 12, 13, and 14 present thought-provoking material designed to interest graduate students in current research topics. KEY FEATURES

This text has many key features that distinguish it from other books.

Unlike other software engineering books that consider measurement a separate issue, this book blends measurement with software engineering. Measurement issues are considered as an integral part of software engineering strategy, rather than as a separate discipline. This approach shows students how to involve quantitative assessment and improvement in their daily activities. They can evaluate their progress on an individual, team, and project basis. Similarly, concepts such as reuse, risk management, and quality engineering are embedded in the software engineering activities that are affected by them, instead of treating them as separate issues. Each chapter applies its concepts to two common examples: one that represents a typical information system, and another that represents a real-time system. Both examples are based on actual projects. The information system example describes the software needed to determine the price of advertising time for a large British television company. The real-time system is the control software for the Ariane-5 rocket; we look at the problems reported, and explore how software engineering techniques could have helped to locate and avoid some of them. Students can follow the progress of two typical projects, seeing how the various practices described in the book are merged into the technologies used to build systems. At the end of every chapter, the results are expressed in three ways: what the content of the chapter means for development teams, what it means for individual developers, and what it means for researchers. The student can easily review the highlights of each chapter and see the chapter's relevance to both research and practice. The book has an associated Web page, containing current examples from the literature, links to Web pages for relevant tool and method vendors, and a study guide for students. It is on the Web that students can find real requirements documents, designs, code, test plans, and more, so they can see real software engineering project artifacts. Students seeking additional in-depth information are pointed to reputable accessible publications and Web sites. The Web pages are updated regularly to keep the material in the textbook current and include a facility for feedback to the author and the publisher. The book is replete with case studies and examples from the literature. Many of the one-page case studies shown as sidebars in the book are expanded on the Web page. The student can see how the book's theoretical concepts are applied to real-life situations. Each chapter ends with thought-provoking questions about legal and ethical issues in software engineering. Students see software engineering in its social and political contexts. As with other sciences, software engineering decisions must be viewed in terms of the people their consequences will affect. Every chapter addresses both procedural and object-oriented development. In addition, a new chapter on object-oriented development explains the steps of an object-oriented development process. Using UML for common notation, each step is applied to a common example, from requirements specification through program design. The book has an annotated bibliography that points to many of the seminal papers in software engineering. In addition, the Web page points to annotated bibliographies and discussion groups for specialized areas, such as software reliability, fault tolerance, computer security, and more. The book has a solutions manual, available from Prentice Hall, as are PowerPoint slides with the figures, tables, and sample instructional slides. Each chapter includes a description of a term project, involving development of software for a mortgage processing system. The instructor may use this term project, or a variation of it, in class assignments. Each chapter ends with a list of key references for the concepts in the chapter, enabling students to find in-depth information about particular tools and methods discussed in the chapter. CONTENTS AND ORGANIZATION

This text is organized in three parts. The first part (Chapters 1 to 3) motivates the reader, explaining why knowledge of software engineering is important to practitioners and researchers alike. Part I also discusses the need for understanding process issues and for doing careful project planning. Part II (Chapters 4 to 11) walks through the major steps of development and maintenance, regardless of the process model used to build the software: eliciting and checking the requirements, designing a solution to the problem, writing and testing the code, and turning it over to the customer. Part III (Chapters 12 to 14) focuses on evaluation and improvement. It looks at how we can assess the quality of our processes and products, and how to take steps to improve them. Chapter 1: Why Software Engineering?

In this chapter we address our track record, motivating the reader and highlighting where in later chapters certain key issues are examined. In particular, we look at Wasserman's key factors that help define software engineering: abstraction, analysis and design methods and notations, modularity and architecture, software life cycle and press, reuse, measurement, tools and integrated environments, and user interface and prototyping. We discuss the difference between computer science and software engineering, explaining some of the major types of problems that can be encountered, and laying the groundwork for the rest of the book. We also explore the need to take a systems approach to building software, and we introduce the two common examples that will be used in every chapter. We also introduce the context for the term project. Chapter 2: Modeling the Process and Life Cycle

In this chapter, we present an overview of different types of process and life-cycle models, including the waterfall model, the V -model, the spiral model, and various prototyping models. We also describe several modeling techniques and tools, including systems dynamics, SADT, and other commonly-used approaches. Each of the two common examples is modeled in part with some of the techniques introduced here. Chapter 3: Planning and Managing the Project

Here, we look at project planning and scheduling. We introduce notions such as activities and milestones, work breakdown structure, activity graphs, risk management, and costs and cost estimation. Estimation models are used to estimate the cost and schedule of the two common examples. We focus on actual case studies, including management of software development for the F-16 airplane and for Digital's alpha AXP programs. Chapter 4: Capturing the Requirements

In this chapter, we look at requirements analysis and specification. We explain the difference between functional and nonfunctional requirements, present several ways to describe different kinds of requirements, and discuss how to prototype requirements. We see how several types of formal methods can be used in specifying and evaluating requirements. Other topics discussed include requirements documentation, requirements reviews, requirements quality and how to measure it, requirements testability, and how to select a specification method. The chapter ends with application of some of the methods to the two common examples. Chapter 5: Designing the System

This chapter focuses on architectural issues, and we begin by discussing Shaw and Garlan's fra

From the Back Cover

This thorough book blends concepts with real, practical applications by providing an abundance of examples.

Dr. Pfleeger applies state-of-the-art software engineering methods consistently to two case studies that are integrated throughout: a realistic information system and a requirements rich real-time system.

Read Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger for online ebook

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger books to read online.

Online Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger ebook PDF download

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger Doc

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger Mobipocket

Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger EPub

JW2CU5A19ZI: Software Engineering: Theory and Practice (2nd Edition) By Shari Lawrence Pfleeger