
Download File PDF Lawrence Shari Practice Theory Engineering Software

Thank you very much for reading **Lawrence Shari Practice Theory Engineering Software**. Maybe you have knowledge that, people have look hundreds times for their favorite books like this Lawrence Shari Practice Theory Engineering Software, but end up in infectious downloads.

Rather than reading a good book with a cup of tea in the afternoon, instead they juggled with some harmful virus inside their desktop computer.

Lawrence Shari Practice Theory Engineering Software is available in our book collection an online access to it is set as public so you can download it instantly.

Our book servers spans in multiple countries, allowing you to get the most less latency time to download any of our books like this one.

Kindly say, the Lawrence Shari Practice Theory Engineering Software is universally compatible with any devices to read

KEY=LAWRENCE - VIRGINIA PAGE

Software Engineering

Theory and Practice

Prentice Hall Featuring an associated Web page, and consistently combining theory with real-world practical applications, this text includes thought-provoking questions about legal and ethical issues in software engineering.

Software Engineering

Pearson Education India

Software Engineering

Theory and Practice

Pfleeger divides her study into three major sections: a motivational treatise on why knowledge of software engineering is important, the major steps of development and maintenance including requirements analysis and architecture, and evaluation and improvement needs after delivery for future redesign and redevelopment.

Studyguide for Software Engineering

Theory and Practice by Pfleeger, Shari Lawrence

Cram101 Never HIGHLIGHT a Book Again Includes all testable terms, concepts, persons, places, and events. Cram101 Just the FACTS101 studyguides gives all of the outlines, highlights, and quizzes for your textbook with optional online comprehensive practice tests. Only Cram101 is Textbook Specific. Accompanys: 9780872893795. This item is printed on demand.

Outlines and Highlights for Software Engineering

Theory and Practice by Shari Lawrence Pfleeger

Academic Internet Pub Incorporated Never HIGHLIGHT a Book Again! Virtually all of the testable terms, concepts, persons, places, and events from the textbook are included. Cram101 Just the FACTS101 studyguides give all of the outlines, highlights, notes, and quizzes for your textbook with optional online comprehensive practice tests. Only Cram101 is Textbook Specific. Accompanys: 9780136061694 .

Software Engineering: Theory and Practice: Fourth Edition

Pearson Education India

Student Study Guide for Software Engineering

Theory and Practice

Prentice Hall Featuring an associated Web page, and consistently combining theory with real-world practical applications, this text includes thought-provoking questions about legal and ethical issues in software engineering.

Analyzing Computer Security

A Threat/vulnerability/countermeasure Approach

Prentice Hall Professional In this book, the authors of the 20-year best-selling classic *Security in Computing* take a fresh, contemporary, and powerfully relevant new approach to introducing computer security. Organised around attacks and mitigations, the Pfleegers' new *Analyzing Computer Security* will attract students' attention by building on the high-profile security failures they may have already encountered in the popular media. Each section starts with an attack description. Next, the authors explain the vulnerabilities that have allowed this attack to occur. With this foundation in place, they systematically present today's most effective countermeasures for blocking or weakening the attack. One step at a time, students progress from attack/problem/harm to solution/protection/mitigation, building the powerful real-world problem solving skills they need to succeed as information security professionals. *Analyzing Computer Security* addresses crucial contemporary computer security themes throughout, including effective security management and risk analysis; economics and quantitative study; privacy, ethics, and laws; and the use of overlapping controls. The authors also present significant new material on computer forensics, insiders, human factors, and trust.

Applying Software Metrics

John Wiley & Sons Features a useful collection of important and practical papers on applying software metrics and measurement. The book details the importance of planning a successful measurement program with a complete discussion of why, what, where, when, and how to measure and who should be involved. Each chapter addresses these significant questions and provides the essential answers in building an effective measurement program. The book differs from others on the market by focusing on the application of the metrics rather than the metrics themselves. The author's provide information based on actual experience with successful metrics programs. Each chapter includes a case study focusing on technology transfer and a set of recommended references. The book serves as a guide on the use and application of software metrics in industrial environments. It is specially designed for managers, product supervisors, and quality assurance personnel who want to know how to implement a metrics program.

Software Engineering

The Production of Quality Software

Macmillan College

Product Focused Software Process Improvement

Third International Conference, PROFES 2001, Kaiserslautern, Germany, September 10-13, 2001. Proceedings

Springer The Third International Conference on Product Focused Software Process Improvement (PROFES 2001) continued the success of the PROFES'99 and PROFES 2000 conferences. PROFES 2001 was organized in Kaiserslautern, Germany, September 10-13, 2001. The PROFES conference has its roots in the PROFES Esprit project (<http://www.ele.vtt.fi/profes/>), but it quickly evolved into a full fledged general purpose conference in 1999 and since then it has gained wide spread international popularity. As in previous years, the main theme of PROFES 2001 was professional software process improvement (SPI) motivated by product and service quality needs. SPI is facilitated by software process assessment, software measurement, process modeling, and technology transfer and has become a practical tool for quality software engineering and management. The conference addresses both the solutions found in practice as well as relevant research results from academia. The purpose of the conference is to bring to light the most recent findings and results in the area and to stimulate discussion between the researchers, experienced professionals, and technology providers for SPI.

EBOOK: Object-Oriented Software Engineering: Practical Software Development Using UML and Java

McGraw Hill EBOOK: Object-Oriented Software Engineering: Practical Software Development Using UML and Java

Multi Pack Software Engineering

Theory and Practise and Extreme Programming Explained: Embracing Change

Prentice Hall This multi pack is made up of the following components: Pfleeger/ Software Engineering: Theory and Practice 0130931292 Beck/ Extreme Programming Explained: Embrace Change 020161641

Hermeneutical Engineering of Requirements: A Technical Approach to Improving the Elicitation and the Evaluating of the Software Requirements (The manuscript, in its essence)

Scientific Research Publishing, Inc. USA Software Engineering aims to develop software in a “systematic, controlled and quantifiable” way, through the application of a series of combined and integrated activities. In order to define what the software should do, it includes the execution of the activity Requirements Engineering, whose purpose is to identify, examine and specify the context of the software to be developed. Software development depends primarily on this activity. However, there is a growing problem: to understand the context of the software to be developed. This manuscript presents a proposal that face this problem through the application of Hermeneutical Engineering of Requirements, which is made up of two instruments: Hermeneutical Elicitation of Requirements and Hermeneutical Theodolite of Requirements. The Hermeneutical Elicitation of Requirements uses hermeneutic methods suitable specifically for Requirements Engineering, which will help the requirements engineer to better understand the original business needs to be met. The Hermeneutical Theodolite of Requirements is an instrument composed of two mechanisms: one that evaluates and presents the levels of understanding and difficulty that the requirements engineer has in relation to the domain of the application, and another that evaluates and presents the quality grades, and to the levels of difficulty, of the software requirements. Thus, it will be possible to establish strategies to improve the application of Hermeneutical Elicitation of Requirements. With this, the Hermeneutical Engineering of Requirements will help the requirements engineer to better understand the context of the software being developed and thus be him able to determine and better build the software requirements.

Planning Smarter

Creating Blueprint-Quality Software Specifications

Prentice Hall Professional In this volume, the author shows how to dramatically reduce the time and resources expended in project planning, while producing more effective project specifications. It gives real-world insight into project planning and refreshing perspectives on how to tackle the software planning challenge. It provides simple techniques.

Basics of Software Engineering Experimentation

Springer Science & Business Media Basics of Software Engineering Experimentation is a practical guide to experimentation in a field which has long been underpinned by suppositions, assumptions, speculations and beliefs. It demonstrates to software engineers how Experimental Design and Analysis can be used to validate their beliefs and ideas. The book does not assume its readers have an in-depth knowledge of mathematics, specifying the conceptual essence of the techniques to use in the design and analysis of experiments and keeping the mathematical calculations clear and simple. Basics of Software Engineering Experimentation is practically oriented and is specially written for software engineers, all the examples being based on real and fictitious software engineering experiments.

Mult Pack Software Engg (International Edition)

Prentice Hall This multi pack is made up of the following components: Pfleeger/ Software Engineering: Theory and Practice 0130931292 Whittaker/ How to Break Software: A Practical Guide to Testing 020179619

Computer Safety, Reliability, and Security

19th International Conference, SAFECOMP 2000, Rotterdam, The Netherlands, October 24-27, 2000

Proceedings

Springer This book constitutes the refereed proceedings of the 19th International Conference on Computer Safety, Reliability, and Security, SAFECOMP 2000, held in Rotterdam, The Netherlands in October 2000. The 33 revised full papers presented together with three invited papers were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on verification and validation; software process improvement; formal methods; safety guidelines, standards and certification; hardware aspects; safety assessment; design for safety; and transport and infrastructure.

Software Engineering

A Methodical Approach

Apress This text provides a comprehensive, but concise introduction to software engineering. It adopts a methodical approach to solving software engineering problems proven over several years of teaching, with outstanding results. The book covers concepts, principles, design, construction, implementation, and management issues of software systems. Each chapter is organized systematically into brief, reader-friendly sections, with itemization of the important points to be remembered. Diagrams and illustrations also sum up the salient points to enhance learning. Additionally, the book includes a number of the author's original methodologies that add clarity and creativity to the software engineering experience, while making a novel contribution to the discipline. Upholding his aim for brevity, comprehensive coverage, and relevance, Foster's practical and methodical discussion style gets straight to the salient issues, and avoids unnecessary topics and minimizes theoretical coverage.

Mastering the Requirements Process

Getting Requirements Right

Addison-Wesley "If the purpose is to create one of the best books on requirements yet written, the authors have succeeded."
—Capers Jones Software can solve almost any problem. The trick is knowing what the problem is. With about half of all software errors originating in the requirements activity, it is clear that a better understanding of the problem is needed. Getting the requirements right is crucial if we are to build systems that best meet our needs. We know, beyond doubt, that the right requirements produce an end result that is as innovative and beneficial as it can be, and that system development is both effective and efficient. Mastering the Requirements Process: Getting Requirements Right, Third Edition, sets out an industry-proven process for gathering and verifying requirements, regardless of whether you work in a traditional or agile development environment. In this sweeping update of the bestselling guide, the authors show how to discover precisely what the customer wants and needs, in the most efficient manner possible. Features include The Volere requirements process for discovering requirements, for use with both traditional and iterative environments A specification template that can be used as the basis for your own requirements specifications Formality guides that help you funnel your efforts into only the requirements work needed for your particular development environment and project How to make requirements testable using fit criteria Checklists to help identify stakeholders, users, non-functional requirements, and more Methods for reusing requirements and requirements patterns New features include Strategy guides for different environments, including outsourcing Strategies for gathering and implementing requirements for iterative releases "Thinking above the line" to find the real problem How to move from requirements to finding the right solution The Brown Cow model for clearer viewpoints of the system Using story cards as requirements Using the Volere Knowledge Model to help record and communicate requirements Fundamental truths about requirements and system development

Managed Software Evolution

Springer This open access book presents the outcomes of the "Design for Future - Managed Software Evolution" priority program 1593, which was launched by the German Research Foundation ("Deutsche Forschungsgemeinschaft (DFG)") to develop new approaches to software engineering with a specific focus on long-lived software systems. The different lifecycles of software and hardware platforms lead to interoperability problems in such systems. Instead of separating the development, adaptation and evolution of software and its platforms, as well as aspects like operation, monitoring and maintenance, they should all be integrated into one overarching process. Accordingly, the book is split into three major parts, the first of which includes an introduction to the nature of software evolution, followed by an overview of the specific challenges and a general introduction to the case studies used in the project. The second part of the book consists of the main chapters on knowledge carrying software, and cover tacit knowledge in software evolution, continuous design decision support, model-based round-trip engineering for software product lines, performance analysis strategies, maintaining security in software evolution, learning from evolution for evolution, and formal verification of evolutionary changes. In turn, the last part of the book presents key findings and spin-offs. The individual chapters there describe

various case studies, along with their benefits, deliverables and the respective lessons learned. An overview of future research topics rounds out the coverage. The book was mainly written for scientific researchers and advanced professionals with an academic background. They will benefit from its comprehensive treatment of various topics related to problems that are now gaining in importance, given the higher costs for maintenance and evolution in comparison to the initial development, and the fact that today, most software is not developed from scratch, but as part of a continuum of former and future releases.

INFORMATION SYSTEMS MANAGEMENT IN BUSINESS AND DEVELOPMENT ORGANIZATIONS (Text and Cases)

PHI Learning Pvt. Ltd. Management Information Systems (MIS) has fast emerged as a multi-disciplinary area having strategic interfaces to achieve organizational objectives. This comprehensive book discusses the underlying principles of business and development organizations, identifies their core areas and prescribes approaches to develop MIS. Divided into five parts, Part I—Understanding Organizations for MIS deals with organizational issues and focuses on the rationale behind creating organizations, especially business and development organizations, to understand their distinguishing features. Part II—Systems Approach to Organizations covers conceptualization, identification, design and development of Information System (IS) for the organization in order to have better systems in place to support organizational goals. Part III—Understanding MIS discusses the relevance of MIS in organizations and the forms it can take to meet the strategic needs of the respective organizations. Part IV—Understanding Information Technologies describes possible approaches to plan, identify and deploy ICT in the acquiring organizations and provides insight into the barriers that creep in during identification and deployment of IS and ICT keeping in view the organizational objectives. Part V—Planning and Implementation of MIS concludes with a discussion on preparation of MIS plan and issues related to its implementation. The book is intended for the postgraduate students of management specializing in rural management and IT. **Key Features** • Describes life cycle approach and systems approach to organizations. • Contains a large number of case studies. • Provides real-life examples to put the concepts in the right perspective.

Impossible Certainty

Cost Risk Analysis for Air Force Systems

Rand Corporation Uncertainty and risk affect countless aspects of our everyday life: What investments should I make? Is eating this food good for my health? Cost analysts too must predict—sometimes very far—into the future how much money the military will spend on weapon systems. Similar to those making decisions about everyday life uncertainties, an analyst does not have perfect knowledge about tomorrow's technology, economic conditions, or any other future event. Actual costs could be substantially higher or lower than originally anticipated. Based on literature reviews, numerous interviews, and analysis of historical cost data, the authors of this book look at how estimates for future weapon systems can be more realistic than in the past and how cost uncertainty analyses can be more comprehensive and informative. To help set the Air Force's cost uncertainty analysis policy, the authors recommend that the Air Force flexibly use multiple methods for different cases; have consistent, uniform communications formats between analysts and decisionmakers; periodically track and update cost estimate records; and consider risk reserves to fund costs that arise from unforeseen circumstances.

Database Systems

A Pragmatic Approach, 3rd edition

CRC Press This book provides a concise but comprehensive guide to the disciplines of database design, construction, implementation, and management. Based on the authors' professional experience in the software engineering and IT industries before making a career switch to academia, the text stresses sound database design as a necessary precursor to successful development and administration of database systems. The discipline of database systems design and management is discussed within the context of the bigger picture of software engineering. Students are led to understand from the outset of the text that a database is a critical component of a software infrastructure, and that proper database design and management is integral to the success of a software system. Additionally, students are led to appreciate the huge value of a properly designed database to the success of a business enterprise. The text was written for three target audiences. It is suited for undergraduate students of computer science and related disciplines who are pursuing a course in database systems, graduate students who are pursuing an introductory course to database, and practicing software engineers and information technology (IT) professionals who need a quick reference on database design. **Database Systems: A Pragmatic Approach, 3rd Edition** discusses concepts, principles, design, implementation, and management issues related to database systems. Each chapter is organized into brief, reader-friendly, conversational sections with itemization of salient points to be remembered. This pragmatic approach includes adequate treatment of database theory and practice based on strategies that have been tested, proven, and refined over several years. Features of the third edition include: Short paragraphs that express the salient aspects of each subject **Bullet points** itemizing important points for easy memorization **Fully revised and updated diagrams and figures** to illustrate concepts to enhance the student's understanding **Real-world examples** **Original methodologies** applicable to database design **Step-by-step, student-friendly guidelines** for solving generic database systems problems **Opening**

chapter overviews and concluding chapter summaries Discussion of DBMS alternatives such as the Entity-Attributes-Value model, NoSQL databases, database-supporting frameworks, and other burgeoning database technologies A chapter with sample assignment questions and case studies This textbook may be used as a one-semester or two-semester course in database systems, augmented by a DBMS (preferably Oracle). After its usage, students will come away with a firm grasp of the design, development, implementation, and management of a database system.

Six Sigma Software Development, Second Edition

CRC Press Even though Six Sigma programs have successfully been implemented in practice, many IT departments remain skeptical of the process or are unaware of how the tools can be used to improve system development. Removing the mystique surrounding this technique, *Six Sigma Software Development, Second Edition* demonstrates how Six Sigma tools and concepts can be used to enhance the system development process. Revised and updated, this second edition clearly explains Six Sigma concepts and their application, maps Six Sigma concepts and tools to all aspects of system development, and proposes the use of Six Sigma tools to evaluate and improve the overall performance of the IT department. In addition to classic Six Sigma, the book introduces Design for Six Sigma (DFSS) and illustrates when and how its tools and techniques can be used to increase the robustness and reliability of a new system. It also shows how the judicious application of lean tools can reduce the complexity of IT processes, thus shortening the time needed to translate customer requirements into completed systems and increasing customer satisfaction.

The Genie in the Machine

How Computer-automated Inventing is Revolutionizing Law and Business

Stanford University Press *The Genie in the Machine* examines how computers are being used to automate the process of inventing, and explains the steps that high-tech companies, patent lawyers, inventors, and consumers should take to thrive in the upcoming Artificial Invention Age.

Testing Web Security

Assessing the Security of Web Sites and Applications

John Wiley & Sons

Software Sizing, Estimation, and Risk Management

When Performance is Measured Performance Improves

CRC Press To achieve consistent software project success under the pressures of today's software development environment, software organizations require achievable plans including viable estimates of schedule, resources, and risks. To estimate realistically, you must understand how to apply sound estimation processes, tools, and data. *Software Sizing*

Guide to Advanced Empirical Software Engineering

Springer Science & Business Media This book gathers chapters from some of the top international empirical software engineering researchers focusing on the practical knowledge necessary for conducting, reporting and using empirical methods in software engineering. Topics and features include guidance on how to design, conduct and report empirical studies. The volume also provides information across a range of techniques, methods and qualitative and quantitative issues to help build a toolkit applicable to the diverse software development contexts

Security in Computing

Prentice Hall Professional This third edition of the all time classic computer security book provides an overview of all types of computer security from centralized systems to distributed networks. The book has been updated to make the most current information in the field available and accessible to today's professionals.

Agile!

The Good, the Hype and the Ugly

Springer Science & Business Media Are you attracted by the promises of agile methods but put off by the fanaticism of many agile texts? Would you like to know which agile techniques work, which ones do not matter much, and which ones will harm your projects? Then you need *Agile!*: the first exhaustive, objective review of agile principles, techniques and tools. Agile methods are one of the

most important developments in software over the past decades, but also a surprising mix of the best and the worst. Until now every project and developer had to sort out the good ideas from the bad by themselves. This book spares you the pain. It offers both a thorough descriptive presentation of agile techniques and a perceptive analysis of their benefits and limitations. Agile! serves first as a primer on agile development: one chapter each introduces agile principles, roles, managerial practices, technical practices and artifacts. A separate chapter analyzes the four major agile methods: Extreme Programming, Lean Software, Scrum and Crystal. The accompanying critical analysis explains what you should retain and discard from agile ideas. It is based on Meyer's thorough understanding of software engineering, and his extensive personal experience of programming and project management. He highlights the limitations of agile methods as well as their truly brilliant contributions — even those to which their own authors do not do full justice. Three important chapters precede the core discussion of agile ideas: an overview, serving as a concentrate of the entire book; a dissection of the intellectual devices used by agile authors; and a review of classical software engineering techniques, such as requirements analysis and lifecycle models, which agile methods criticize. The final chapters describe the precautions that a company should take during a transition to agile development and present an overall assessment of agile ideas. This is the first book to discuss agile methods, beyond the brouhaha, in the general context of modern software engineering. It is a key resource for projects that want to combine the best of established results and agile innovations.

Creating Maintainable APIs

A Practical, Case-Study Approach

Apress Build straightforward and maintainable APIs to create services that are usable and maintainable. Although this book focuses on distributed services, it also emphasizes how the core principles apply even to pure OOD and OOP constructs. The overall context of Creating Maintainable APIs is to classify the topics into four main areas: classes and interfaces, HTTP REST APIs, messaging APIs, and message payloads (XML, JSON and JSON API as well as Apache Avro). What You Will Learn Use object-oriented design constructs and their APIs Create and manage HTTP REST APIs Build and manage maintainable messaging APIs, including the use of Apache Kafka as a principal messaging hub Handle message payloads via JSON Who This Book Is For Any level software engineers and very experienced programmers.

Computational Intelligence In Software Engineering, Advances In Fuzzy Systems: Applications And Theory

World Scientific This unique volume is the first publication on software engineering and computational intelligence (CI) viewed as a synergistic interplay of neurocomputing, granular computation (including fuzzy sets and rough sets), and evolutionary methods. It presents a unified view of CI in the context of software engineering. The book addresses a number of crucial issues: what is CI, what role does it play in software development, how are CI elements built into successive phases of the software life cycle, and what is the role played by CI in quantifying fundamental features of software artifacts? With contributions from leading researchers and practitioners, the book provides the reader with a wealth of new concepts and approaches, complete algorithms, in-depth case studies, and thought-provoking exercises. The topics coverage include neurocomputing, granular as well as evolutionary computing, object-oriented analysis and design in software engineering. There is also an extensive bibliography.

Software Development Measurement Programs

Development, Management and Evolution

Springer This book seeks to promote the structured, standardized and accurate use of software measurement at all levels of modern software development companies. To do so, it focuses on seven main aspects: sound scientific foundations, cost-efficiency, standardization, value-maximization, flexibility, combining organizational and technical aspects, and seamless technology integration. Further, it supports companies in their journey from manual reporting to automated decision support by combining academic research and industrial practice. When scientists and engineers measure something, they tend to focus on two different things. Scientists focus on the ability of the measurement to quantify whatever is being measured; engineers, however, focus on finding the right qualities of measurement given the designed system (e.g. correctness), the system's quality of use (e.g. ease of use), and the efficiency of the measurement process. In this book, the authors argue that both focuses are necessary, and that the two are complementary. Thus, the book is organized as a gradual progression from theories of measurement (yes, you need theories to be successful!) to practical, organizational aspects of maintaining measurement systems (yes, you need the practical side to understand how to be successful). The authors of this book come from academia and industry, where they worked together for the past twelve years. They have worked with both small and large software development organizations, as researchers and as measurement engineers, measurement program leaders and even teachers. They wrote this book to help readers define, implement, deploy and maintain company-wide measurement programs, which consist of a set of measures, indicators and roles that are built around the concept of measurement systems. Based on their experiences introducing over 40,000 measurement systems at over a dozen companies, they share essential tips and tricks on how to do it right and how to avoid common pitfalls.

Conceptual Modeling - ER 2006

25th International Conference on Conceptual Modeling, Tucson, AZ, USA, November 6-9, 2006, Proceedings

Springer This book constitutes the refereed proceedings of the 25th International Conference on Conceptual Modeling, ER 2006, held in Tucson, AZ, USA in November 2006. The 37 revised full papers presented together with two keynote talks, two panel session papers, six industrial papers, and five demo/posters papers were carefully reviewed and selected from 158 submissions.

Security in Computing

Pearson Education India

Software Engineering

S. Chand Publishing This book is a comprehensive, step-by-step guide to software engineering. This book provides an introduction to software engineering for students in undergraduate and post graduate programs in computers.

Advances in Intelligent IT

Active Media Technology 2006

IOS Press " In the great digital era, we are witnessing many rapid scientific and technological developments in human-centered, seamless computing environments, interfaces, devices and systems with applications ranging from business and communication to entertainment and learning. These developments are collectively best characterized as Active Media Technology (AMT), a new area of intelligent information technology and computer science that emphasizes the proactive, seamless roles of interfaces and systems as well as new media in all aspects of digital life. An AMT based computer system offers services that enable the rapid design, implementation, deploying and support of customized solutions. This book brings together papers from researchers from diverse areas, such as Web intelligence, data mining, intelligent agents, smart information use, networking and intelligent interface. The book includes papers on the following topics: Active Computer Systems and Intelligent Interfaces; Adaptive Web Systems and Information Foraging Agents; Web mining, Wisdom Web and Web Intelligence; E-Commerce and Web Services; Data Mining, Ontology Mining and Data Reasoning; Network, Mobile and Wireless Security; Entertainment and Social Applications of Active Media; Agent-Based Software Engineering and Multi-Agent Systems; Digital City and Digital Interactivity; Machine Learning and Human-Centered Robotics; Multi-Modal Processing, Detection, Recognition, and Expression Analysis; Personalized, Pervasive, and Ubiquitous Systems and their Interfaces; Smart Digital Media; and Evaluation of Active Media and AMT Based Systems. "

Software Metrics

A Rigorous and Practical Approach

Course Technology Ptr PART I: FUNDAMENTALS OF MEASUREMENT AND EXPERIMENTATION 1. Measurement: What Is It and Why Do It? 2. The Basics of Measurement 3. A Goal-Based Framework for Software Measurement 4. Empirical Investigation 5. Software Metrics Data Collection 6. Analyzing Software-Measurement Data PART II: SOFTWARE-ENGINEERING MEASUREMENT 7. Measuring Internal Product Attributes: Size 8. Measuring Internal Product Attributes: Structure 9. Measuring Internal Product Attributes 10. Software Reliability: Measurement and Prediction 11. Resource Measurement: Productivity, Teams, and Tools 12. Making Process Predictions PART III: MEASUREMENT AND MANAGEMENT 13. Planning a Measurement Program 14. Measurement in Practice 15. Empirical Research in Software Engineering APPENDIXES: A. Solutions to Selected Exercises / B. Metric Tools / C. Acronyms and Glossary / ANNOTATED BIBLIOGRAPHY / INDEX

Computational Intelligence in Software Engineering

World Scientific This unique volume is the first publication on software engineering and computational intelligence (CI) viewed as a synergistic interplay of neurocomputing, granular computation (including fuzzy sets and rough sets), and evolutionary methods. It presents a unified view of CI in the context of software engineering. The book addresses a number of crucial issues: what is CI, what role does it play in software development, how are CI elements built into successive phases of the software life cycle, and what is the role played by CI in quantifying fundamental features of software artifacts? With contributions from leading researchers and practitioners, the book provides the reader with a wealth of new concepts and approaches, complete algorithms, in-depth case studies, and thought-provoking exercises. The topics coverage include neurocomputing, granular as well as evolutionary computing, object-oriented analysis and design in software engineering. There is also an extensive bibliography.