
Site To Download Guide To The Software Engineering Body Of Knowledge Swebok R

If you ally infatuation such a referred **Guide To The Software Engineering Body Of Knowledge Swebok R** ebook that will provide you worth, get the unconditionally best seller from us currently from several preferred authors. If you desire to comical books, lots of novels, tale, jokes, and more fictions collections are next launched, from best seller to one of the most current released.

You may not be perplexed to enjoy all books collections Guide To The Software Engineering Body Of Knowledge Swebok R that we will unquestionably offer. It is not a propos the costs. Its approximately what you habit currently. This Guide To The Software Engineering Body Of Knowledge Swebok R, as one of the most functioning sellers here will totally be among the best options to review.

KEY=KNOWLEDGE - YOUNG SANTOS

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE (SWEBOK(R))

VERSION 3.0

In the Guide to the Software Engineering Body of Knowledge (SWEBOK(R) Guide), the IEEE Computer Society establishes a baseline for the body of knowledge for the field of software engineering, and the work supports the Society's responsibility to promote the advancement of both theory and practice in this field. It should be noted that the Guide does not purport to define the body of knowledge but rather to serve as a compendium and guide to the knowledge that has been developing and evolving over the past four decades. Now in Version 3.0, the Guide's 15 knowledge areas summarize generally accepted topics and list references for detailed information. The editors for Version 3.0 of the SWEBOK(R) Guide are Pierre Bourque (Ecole de technologie superieure (ETS), Universite du Quebec) and Richard E. (Dick) Fairley (Software and Systems Engineering Associates (S2EA)).

SWEBOK

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE ; TRIAL VERSION ; A PROJECT OF THE SOFTWARE ENGINEERING COORDINATING COMMITTEE ; TRIAL VERSION 1.00

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE

IEEE Computer Society Press The purpose of the Guide to the Software Engineering Body of Knowledge is to provide a validated classification of the bounds of the software engineering discipline and topical access that will support this discipline. The Body of Knowledge is subdivided into ten software engineering Knowledge Areas (KA) that differentiate among the various important concepts, allowing readers to find their way quickly to subjects of interest. Upon finding a subject, readers are referred to key papers or book chapters. Emphases on engineering practice lead the Guide toward a strong relationship with the normative literature. The normative literature is validated by consensus formed among practitioners and is concentrated in standards and related documents. The two major standards bodies for software engineering (IEEE Computer Society Software and Systems Engineering Standards Committee and ISO/IEC JTC1/SC7) are represented in the project.

SOFTWARE ENGINEERING - GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE (SWEBOK).

SOFTWARE ENGINEERING

ARCHITECTURE-DRIVEN SOFTWARE DEVELOPMENT

Newnes **Software Engineering: Architecture-driven Software Development** is the first comprehensive guide to the underlying skills embodied in the IEEE's Software Engineering Body of Knowledge (SWEBOK) standard. Standards expert Richard Schmidt explains the traditional software engineering practices recognized for developing projects for government or corporate systems. Software engineering education often lacks standardization, with many institutions focusing on implementation rather than design as it impacts product architecture. Many graduates join the workforce with incomplete skills, leading to software projects that either fail outright or run woefully over budget and behind schedule. Additionally, software engineers need to understand system engineering and architecture—the hardware and peripherals their programs will run on. This issue will only grow in importance as more programs leverage parallel

computing, requiring an understanding of the parallel capabilities of processors and hardware. This book gives both software developers and system engineers key insights into how their skillsets support and complement each other. With a focus on these key knowledge areas, Software Engineering offers a set of best practices that can be applied to any industry or domain involved in developing software products. A thorough, integrated compilation on the engineering of software products, addressing the majority of the standard knowledge areas and topics Offers best practices focused on those key skills common to many industries and domains that develop software Learn how software engineering relates to systems engineering for better communication with other engineering professionals within a project environment

BUILDING SOFTWARE

A PRACTITIONER'S GUIDE

CRC Press Novel in its approach to software design, development, and management, **Building Software: A Practitioner's Guide** shows you how to successfully build and manage a system. The approach the authors recommend is a simple, effective framework known as Solution Engineering Execution (SEE). Through SEE, you create a successful solution by following a high

THE CERTIFIED SOFTWARE QUALITY ENGINEER HANDBOOK

Quality Press A comprehensive reference manual to the Certified Software Quality Engineer Body of Knowledge and study guide for the CSQE exam.

GUIDE TO EFFICIENT SOFTWARE DESIGN

AN MVC APPROACH TO CONCEPTS, STRUCTURES, AND MODELS

Springer Nature This classroom-tested textbook presents an active-learning approach to the foundational concepts of software design. These concepts are then applied to a case study, and reinforced through practice exercises, with the option to follow either a structured design or object-oriented design paradigm. The text applies an incremental and iterative software development approach, emphasizing the use of design characteristics and modeling techniques as a way to represent higher levels of design abstraction, and promoting the model-view-controller (MVC) architecture.

Topics and features: provides a case study to illustrate the various concepts discussed throughout the book, offering an in-depth look at the pros and cons of different software designs; includes discussion questions and hands-on exercises that extend the case study and apply the concepts to other problem domains; presents a review of program design fundamentals to reinforce understanding of the basic concepts; focuses on a bottom-up approach to describing software design concepts; introduces the characteristics of a good software design, emphasizing the model-view-controller as an underlying architectural principle; describes software design from both object-oriented and structured perspectives; examines additional topics on human-computer interaction design, quality assurance, secure design, design patterns, and persistent data storage design; discusses design concepts that may be applied to many types of software development projects; suggests a template for a software design document, and offers ideas for further learning. Students of computer science and software engineering will find this textbook to be indispensable for advanced undergraduate courses on programming and software design. Prior background knowledge and experience of programming is required, but familiarity in software design is not assumed.

IEEE COMPUTER SOCIETY REAL-WORLD SOFTWARE ENGINEERING PROBLEMS

A SELF-STUDY GUIDE FOR TODAY'S SOFTWARE PROFESSIONAL

John Wiley & Sons **Key problems for the IEEE Computer Society Certified Software Development Professional (CSDP) Certification Program** IEEE Computer Society Real-World Software Engineering Problems helps prepare software engineering professionals for the IEEE Computer Society Certified Software Development Professional (CSDP) Certification Program. The book offers workable, real-world sample problems with solutions to help readers solve common problems. In addition to its role as the definitive preparation guide for the IEEE Computer Society Certified Software Development Professional (CSDP) Certification Program, this resource also serves as an appropriate guide for graduate-level courses in software engineering or for professionals interested in sharpening or refreshing their skills. The book includes a comprehensive collection of sample problems, each of which includes the problem's statement, the solution, an explanation, and references. Topics covered include: * Engineering economics * Test * Ethics * Maintenance * Professional practice * Software configuration * Standards * Quality assurance * Requirements * Metrics * Software design * Tools and methods * Coding * SQA and V & V IEEE Computer Society Real-World Software Engineering Problems offers an invaluable guide to preparing for the IEEE Computer Society Certified Software Development Professional (CSDP) Certification Program for software professionals, as well as providing students with a

practical resource for coursework or general study.

SOFTWARE ENGINEERING ESSENTIALS

Software Management Training

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE

INCOSE SYSTEMS ENGINEERING HANDBOOK

A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES

John Wiley & Sons **A detailed and thorough reference on the discipline and practice of systems engineering The objective of the International Council on Systems Engineering (INCOSE) Systems Engineering Handbook is to describe key process activities performed by systems engineers and other engineering professionals throughout the life cycle of a system. The book covers a wide range of fundamental system concepts that broaden the thinking of the systems engineering practitioner, such as system thinking, system science, life cycle management, specialty engineering, system of systems, and agile and iterative methods. This book also defines the discipline and practice of systems engineering for students and practicing professionals alike, providing an authoritative reference that is acknowledged worldwide. The latest edition of the INCOSE Systems Engineering Handbook: Is consistent with ISO/IEC/IEEE 15288:2015 Systems and software engineering—System life cycle processes and the Guide to the Systems Engineering Body of Knowledge (SEBoK) Has been updated to include the latest concepts of the INCOSE working groups Is the body of knowledge for the INCOSE Certification Process This book is ideal for any engineering professional who has an interest in or needs to apply systems engineering practices. This includes the experienced systems engineer who needs a convenient reference, a product engineer or engineer in another discipline who needs to perform systems engineering, a new systems engineer, or anyone interested in learning more about systems engineering.**

THE ROAD MAP TO SOFTWARE ENGINEERING

A STANDARDS-BASED GUIDE

Wiley-IEEE Computer Society Press **This book presents describes in detail each of the 40 software and systems engineering**

standards contained in the collection of the IEEE. The book is organized to allow users to quickly pinpoint a subject of interest, find an overall description of the subject, and obtain a clear explanation of best-practice standards for that subject.

SOFTWARE ENGINEERING

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE (SWEBOK).

ESSENTIALS OF SOFTWARE ENGINEERING

Jones & Bartlett Learning **Computer Architecture/Software Engineering**

SOFTWARE ENGINEERING

GUIDE TO SOFTWARE ENGINEERING BODY OF KNOWLEDGE (SWEBOK).

SOFTWARE ENGINEERING. GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE (SWEBOK)

Systemology, Software engineering techniques, Life cycle, Management, Planning, Project management

TEAM GEEK

A SOFTWARE DEVELOPER'S GUIDE TO WORKING WELL WITH OTHERS

"O'Reilly Media, Inc." **In a perfect world, software engineers who produce the best code are the most successful. But in our perfectly messy world, success also depends on how you work with people to get your job done. In this highly entertaining book, Brian Fitzpatrick and Ben Collins-Sussman cover basic patterns and anti-patterns for working with other people, teams, and users while trying to develop software. This is valuable information from two respected software engineers whose popular series of talks—including "Working with Poisonous People"—has attracted hundreds of thousands of followers. Writing software is a team sport, and human factors have as much influence on the outcome as technical factors. Even if you've spent decades learning the technical side of programming, this book teaches you about the often-overlooked human component. By learning to collaborate and investing in the "soft skills" of software**

engineering, you can have a much greater impact for the same amount of effort. Team Geek was named as a Finalist in the 2013 Jolt Awards from Dr. Dobb's Journal. The publication's panel of judges chose five notable books, published during a 12-month period ending June 30, that every serious programmer should read.

PRINCIPLES OF CASE TOOL INTEGRATION

[Oxford University Press on Demand](#) **Computer Aided Software Engineering (CASE) tools typically support individual users in the automation of a set of tasks within a software development process. Such tools have helped organizations in their efforts to develop better software within budget and time constraints. However, many organizations are failing to take full advantage of CASE technology as they struggle to make coordinated use of collections of tools, often obtained at different times from different vendors. This book provides an in-depth analysis of the CASE tool integration problem, and describes practical approaches that can be used with current CASE technology to help your organization take greater advantage of integrated CASE.**

SYSTEMS ENGINEERING GUIDEBOOK

A PROCESS FOR DEVELOPING SYSTEMS AND PRODUCTS

[CRC Press](#) **Systems Engineering Guidebook: A Process for Developing Systems and Products is intended to provide readers with a guide to understanding and becoming familiar with the systems engineering process, its application, and its value to the successful implementation of systems development projects. The book describes the systems engineering process as a multidisciplinary effort. The process is defined in terms of specific tasks to be accomplished, with great emphasis placed on defining the problem that is being addressed prior to designing the solution.**

A GUIDE TO THE WIRELESS ENGINEERING BODY OF KNOWLEDGE (WEBOK)

[John Wiley & Sons](#) **"Responding to requests from industry for a program to ensure consistency and excellence among wireless technical professionals, the IEEE Communications Society (ComSoc) created its Wireless Communication Engineering Technologies (WCET) Certification. Many of the wireless experts who helped develop the WCET certification exam, the WEBOK, and ComSoc's wireless training courses, and helped update it in 2010, are editors or contributing authors to this book. Their inherent familiarity with the field and with each other as authors gives the**

WEBOK a credibility and a thoroughness that cannot be matched"--

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE 2004 SWEBOK.

AGILE MODEL-BASED SYSTEMS ENGINEERING COOKBOOK

IMPROVE SYSTEM DEVELOPMENT BY APPLYING PROVEN RECIPES FOR EFFECTIVE AGILE SYSTEMS ENGINEERING

Packt Publishing Ltd **The Agile Model-Based Systems Engineering Cookbook distills the most relevant MBSE workflows and work products into a set of easy-to-follow recipes, complete with examples of their application. This book serves as a quick and reliable practical reference for systems engineers looking to apply agile MBSE to real-world projects.**

RESEARCH SOFTWARE ENGINEERING WITH PYTHON

BUILDING SOFTWARE THAT MAKES RESEARCH POSSIBLE

CRC Press **Writing and running software is now as much a part of science as telescopes and test tubes, but most researchers are never taught how to do either well. As a result, it takes them longer to accomplish simple tasks than it should, and it is harder for them to share their work with others than it needs to be. This book introduces the concepts, tools, and skills that researchers need to get more done in less time and with less pain. Based on the practical experiences of its authors, who collectively have spent several decades teaching software skills to scientists, it covers everything graduate-level researchers need to automate their workflows, collaborate with colleagues, ensure that their results are trustworthy, and publish what they have built so that others can build on it. The book assumes only a basic knowledge of Python as a starting point, and shows readers how it, the Unix shell, Git, Make, and related tools can give them more time to focus on the research they actually want to do. Research Software Engineering with Python can be used as the main text in a one-semester course or for self-guided study. A running example shows how to organize a small research project step by step; over a hundred exercises give readers a chance to practice these skills themselves, while a glossary defining over two hundred terms will help readers find their way through the terminology. All of the material can be re-used under a Creative Commons license, and all royalties from sales of the book will be donated to The Carpentries, an organization that teaches foundational coding and data science skills to**

researchers worldwide.

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**

BEGINNING SOFTWARE ENGINEERING

John Wiley & Sons **A complete introduction to building robust and reliable software Beginning Software Engineering demystifies the software engineering methodologies and techniques that professional developers use to design and build robust, efficient, and consistently reliable software. Free of jargon and assuming no previous programming, development, or management experience, this accessible guide explains important concepts and techniques that can be applied to any programming language. Each chapter ends with exercises that let you test your understanding and help you elaborate on the chapter's main concepts. Everything you need to understand waterfall, Sashimi, agile, RAD, Scrum, Kanban, Extreme Programming, and many other development models is inside! Describes in plain English what software engineering is Explains the roles and responsibilities of team members working on a software engineering project Outlines key phases that any software engineering effort must handle to produce applications that are powerful and dependable Details the most popular software development methodologies and explains the different ways they handle critical development tasks Incorporates exercises that expand upon each chapter's main ideas Includes an extensive glossary of software engineering terms**

SOFTWARE QUALITY ASSURANCE

John Wiley & Sons **The most comprehensive General, Organic, and Biochemistry book available, Introduction to General, Organic, and Biochemistry, 11th Edition continues its tradition of a solid development of problem-solving skills, numerous examples and practice problems, along with coverage of current applications. Written by an experienced author team, they skillfully anticipate areas of difficulty and pace the book accordingly. Readers will find the right mix**

of general chemistry compared to the discussions on organic and biochemistry. Introduction to General, Organic, and Biochemistry, 11th Edition has clear & logical explanations of chemical concepts and great depth of coverage as well as a clear, consistent writing style which provides great readability. An emphasis on Real-World aspects of chemistry makes the reader comfortable in seeing how the chemistry will apply to their career.

OCCUPATIONAL OUTLOOK HANDBOOK

THE COMPUTER ENGINEERING HANDBOOK

CRC Press There is arguably no field in greater need of a comprehensive handbook than computer engineering. The unparalleled rate of technological advancement, the explosion of computer applications, and the now-in-progress migration to a wireless world have made it difficult for engineers to keep up with all the developments in specialties outside their own

LEAN SOFTWARE SYSTEMS ENGINEERING FOR DEVELOPERS

MANAGING REQUIREMENTS, COMPLEXITY, TEAMS, AND CHANGE LIKE A CHAMP

Apress Get to the next level of your software development career, learning the tools you need to successfully manage the complexity of modern software systems. Whether you are developer at a small software company or a large enterprise, your success is directly related to the ability of your development team to rapidly respond to change. What makes this task challenging is that the tech challenges we strive to overcome are becoming increasingly more complex: requirements, solution, hosting, support, pace of change, etc. A good developer manages every aspect of the program and understands that when details and decisions are left to chance, outcomes can be negatively impacted and result in increased errors due to substandard quality. It is the difference between being a professional software engineer and a programmer. You will know how look at the entire spectrum of the software development process and learn valuable concepts and apply these principles through meaningful examples, exercises, case studies, and source code. What You Will Learn Know what it means to be a professional software engineer Spend more time doing software development and minimize the pain of dealing with inefficient processes Integrate Lean and Agile practices to reduce errors in judgment and provide predictable outcomes, while still maintaining agility and responsiveness Ensure a shared understanding in the group of stakeholders Validate user experience early and often to minimize costly re-work

Develop software designs and architectures that age well and enable long-term business agility Implement patterns and processes that result in developers “falling into the pit of success” instead of into the “pit of failure” Adopt the necessary processes and patterns that will result in “institutionalized” quality that is pervasive Redefine the important role of technical leadership to ensure team maturity and growth Who This Book Is For Software developers and team leaders who have struggled to implement design and development best practices due to lack of in-depth knowledge or experience, and want a book designed to provide the confidence and foundational skills needed to achieve success

ARCHITECTURE-CENTRIC SOFTWARE PROJECT MANAGEMENT

A PRACTICAL GUIDE

Addison-Wesley Professional To fully leverage the value of software architecture in enterprise development projects, you need to expressly and consciously link architecture with project management. This book shows how, drawing on powerful lessons learned at Siemens, one of the world's leading software development organizations. The authors offer insight into project management for software architects, insight into software architecture for project managers, and above all, insight into integrating the two disciplines to maximize the effectiveness of both of them. Learn how to develop cost and schedule estimates for development projects, based on software architecture; how to clarify architecture so projects can be more effectively planned and managed; and then how to use architecture to organize, implement, and measure the project iteratively as work progresses.

PEER REVIEWS IN SOFTWARE

A PRACTICAL GUIDE

Addison-Wesley Professional This practical introduction to peer reviews covers different methods of peer review, from the formal method of inspection to other less formal methods, and addresses the cultural and practical aspects of both.

REVERSING

SECRETS OF REVERSE ENGINEERING

John Wiley & Sons Beginning with a basic primer on reverse engineering-including computer internals, operating

systems, and assembly language-and then discussing the various applications of reverse engineering, this book provides readers with practical, in-depth techniques for software reverse engineering. The book is broken into two parts, the first deals with security-related reverse engineering and the second explores the more practical aspects of reverse engineering. In addition, the author explains how to reverse engineer a third-party software library to improve interfacing and how to reverse engineer a competitor's software to build a better product. * The first popular book to show how software reverse engineering can help defend against security threats, speed up development, and unlock the secrets of competitive products * Helps developers plug security holes by demonstrating how hackers exploit reverse engineering techniques to crack copy-protection schemes and identify software targets for viruses and other malware * Offers a primer on advanced reverse-engineering, delving into "disassembly"-code-level reverse engineering-and explaining how to decipher assembly language

SOFTWARE ENGINEERING

GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE (SWEBOK).

SOFTWARE ENGINEERING MEASUREMENT

CRC Press The product of many years of practical experience and research in the software measurement business, this technical reference helps you select what metrics to collect, how to convert measurement data to management information, and provides the statistics necessary to perform these conversions. The author explains how to manage software development measurement systems, how to build software measurement tools and standards, and how to construct controlled experiments using standardized measurement tools. There are three fundamental questions that this book seeks to answer. First, exactly how do you get the measurement data? Second, how do you convert the data from the measurement process to information that you can use to manage the software development process? Third, how do you manage all of the data? Millions of dollars are being spent trying to secure software systems. When suitable instrumentation is placed into the systems that we develop, their activity can be monitored in real time. Measurement based automatic detection mechanisms can be designed into systems. This will permit the detection of system misuse and detect incipient reliability problems. By demonstrating how to develop simple experiments for the empirical validation of theoretical research and showing how to convert measurement data into meaningful and valuable information, this text fosters more precise use of software measurement in the computer science and

software engineering literature. Software Engineering Measurement shows you how to convert your measurement data to valuable information that can be used immediately for software process improvement.

SOFTWARE PROCESSES AND LIFE CYCLE MODELS

AN INTRODUCTION TO MODELLING, USING AND MANAGING AGILE, PLAN-DRIVEN AND HYBRID PROCESSES

[Springer](#) This book provides a comprehensive overview of the field of software processes, covering in particular the following essential topics: software process modelling, software process and lifecycle models, software process management, deployment and governance, and software process improvement (including assessment and measurement). It does not propose any new processes or methods; rather, it introduces students and software engineers to software processes and life cycle models, covering the different types ranging from “classical”, plan-driven via hybrid to agile approaches. The book is structured as follows: In chapter 1, the fundamentals of the topic are introduced: the basic concepts, a historical overview, and the terminology used. Next, chapter 2 covers the various approaches to modelling software processes and lifecycle models, before chapter 3 discusses the contents of these models, addressing plan-driven, agile and hybrid approaches. The following three chapters address various aspects of using software processes and lifecycle models within organisations, and consider the management of these processes, their assessment and improvement, and the measurement of both software and software processes. Working with software processes normally involves various tools, which are the focus of chapter 7, before a look at current trends in software processes in chapter 8 rounds out the book. This book is mainly intended for graduate students and practicing professionals. It can be used as a textbook for courses and lectures, for self-study, and as a reference guide. When used as a textbook, it may support courses and lectures on software processes, or be used as complementary literature for more basic courses, such as introductory courses on software engineering or project management. To this end, it includes a wealth of examples and case studies, and each chapter is complemented by exercises that help readers gain a better command of the concepts discussed.

FUNDAMENTALS OF COMPUTER PROGRAMMING WITH C#

THE BULGARIAN C# BOOK

[Faber Publishing](#) The free book "Fundamentals of Computer Programming with C#" is a comprehensive computer

programming tutorial that teaches programming, logical thinking, data structures and algorithms, problem solving and high quality code with lots of examples in C#. It starts with the first steps in programming and software development like variables, data types, conditional statements, loops and arrays and continues with other basic topics like methods, numeral systems, strings and string processing, exceptions, classes and objects. After the basics this fundamental programming book enters into more advanced programming topics like recursion, data structures (lists, trees, hash-tables and graphs), high-quality code, unit testing and refactoring, object-oriented principles (inheritance, abstraction, encapsulation and polymorphism) and their implementation the C# language. It also covers fundamental topics that each good developer should know like algorithm design, complexity of algorithms and problem solving. The book uses C# language and Visual Studio to illustrate the programming concepts and explains some C# / .NET specific technologies like lambda expressions, extension methods and LINQ. The book is written by a team of developers lead by Svetlin Nakov who has 20+ years practical software development experience. It teaches the major programming concepts and way of thinking needed to become a good software engineer and the C# language in the meantime. It is a great start for anyone who wants to become a skillful software engineer. The books does not teach technologies like databases, mobile and web development, but shows the true way to master the basics of programming regardless of the languages, technologies and tools. It is good for beginners and intermediate developers who want to put a solid base for a successful career in the software engineering industry. The book is accompanied by free video lessons, presentation slides and mind maps, as well as hundreds of exercises and live examples. Download the free C# programming book, videos, presentations and other resources from <http://introprogramming.info>. Title: Fundamentals of Computer Programming with C# (The Bulgarian C# Programming Book) ISBN: 9789544007737 ISBN-13: 978-954-400-773-7 (9789544007737) ISBN-10: 954-400-773-3 (9544007733) Author: Svetlin Nakov & Co. Pages: 1132 Language: English Published: Sofia, 2013 Publisher: Faber Publishing, Bulgaria Web site: <http://www.introprogramming.info> License: CC-Attribution-Share-Alike Tags: free, programming, book, computer programming, programming fundamentals, ebook, book programming, C#, CSharp, C# book, tutorial, C# tutorial; programming concepts, programming fundamentals, compiler, Visual Studio, .NET, .NET Framework, data types, variables, expressions, statements, console, conditional statements, control-flow logic, loops, arrays, numeral systems, methods, strings, text processing, StringBuilder, exceptions, exception handling, stack trace, streams, files, text files, linear data structures, list, linked list, stack, queue, tree, balanced tree, graph, depth-first search, DFS, breadth-first search, BFS, dictionaries, hash tables, associative arrays, sets, algorithms, sorting algorithm, searching algorithms, recursion, combinatorial algorithms, algorithm complexity, OOP, object-oriented programming, classes,

objects, constructors, fields, properties, static members, abstraction, interfaces, encapsulation, inheritance, virtual methods, polymorphism, cohesion, coupling, enumerations, generics, namespaces, UML, design patterns, extension methods, anonymous types, lambda expressions, LINQ, code quality, high-quality code, high-quality classes, high-quality methods, code formatting, self-documenting code, code refactoring, problem solving, problem solving methodology, 9789544007737, 9544007733

SOFTWARE ENGINEERING, THE DEVELOPMENT PROCESS

Wiley-IEEE Computer Society Press **Volume 1 of Software Engineering, Third Edition** includes reprinted and newly authored papers that describe the technical processes of software development and the associated business and societal context. Together with Volume 2, which describes the key processes that support development, the two volumes address the key issues and tasks facing the software engineer today. The two volumes provide a self-teaching guide and tutorial for software engineers who desire to qualify themselves as Certified Software Development Professionals (CSDP) as described at the IEEE Computer Society Web site (www.computer.org/certification), while also gaining a fuller understanding of standards-based software development. Both volumes consist of original papers written expressly for the two volumes, as well as authoritative papers from the IEEE archival journals, along with papers from other highly regarded sources. The papers and introductions of each chapter provide an orientation to the key concepts and activities described in the new 2004 version as well as the older 2001 version of the Software Engineering Body of Knowledge (SWEBOK), with many of the key papers having been written by the authors of the corresponding chapters of the SWEBOK. Software Engineering is further anchored in the concepts of IEEE/EIA 12207.0-1997 Standard for Information Technology--Software Life Cycle Processes, which provides a framework for all primary and supporting processes, activities, and tasks associated with software development. As the only self-help guide and tutorial based on IEEE/EIA 12207.0--1997, this is an essential reference for software engineers, programmers, and project managers. This volume can also form part of an upper-division undergraduate or graduate-level engineering course. Each chapter in this volume consists of an introduction to the chapter's subject area and an orientation to the relevant areas of the SWEBOK, followed by the supporting articles and, where applicable, the specific IEEE software engineering standard. By emphasizing the IEEE software engineering standards, the SWEBOK, and the contributions of key authors, the two volumes provide a comprehensive orientation to the landscape of software engineering as practiced today. Contents: * Key concepts and activities of software and systems engineering * Societal and legal contexts in which software development takes place * Key IEEE software engineering standards *

Software requirements and methods for developing them * Essential concepts and methods of software design * Guidelines for the selection and use of tools and methods * Major issues and activities of software construction * Software development testing * Preparation and execution of software maintenance programs

THE SECOND MEDIA AGE

John Wiley & Sons This book examines the implications of new communication technologies in the light of the most recent work in social and cultural theory and argues that new developments in electronic media, such as the Internet and Virtual Reality, justify the designation of a "second media age".

A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK® GUIDE) - SEVENTH EDITION AND THE STANDARD FOR PROJECT MANAGEMENT (BRAZILIAN PORTUGUESE)

Project Management Institute **PMBOK® Guide** is the go-to resource for project management practitioners. The project management profession has significantly evolved due to emerging technology, new approaches and rapid market changes. Reflecting this evolution, The Standard for Project Management enumerates 12 principles of project management and the **PMBOK® Guide - Seventh Edition** is structured around eight project performance domains. This edition is designed to address practitioners' current and future needs and to help them be more proactive, innovative and nimble in enabling desired project outcomes. This edition of the **PMBOK® Guide**: • Reflects the full range of development approaches (predictive, adaptive, hybrid, etc.); • Provides an entire section devoted to tailoring the development approach and processes; • Includes an expanded list of models, methods, and artifacts; • Focuses on not just delivering project outputs but also enabling outcomes; and • Integrates with **PMI standards+™** for information and standards application content based on project type, development approach, and industry sector.