Get Free Engineering Software To Related Projects

Eventually, you will unquestionably discover a further experience and skill by spending more cash. yet when? realize you tolerate that you require to get those all needs in the manner of having significantly cash? Why dont you attempt to get something basic in the beginning? Thats something that will guide you to comprehend even more just about the globe, experience, some places, in imitation of history, amusement, and a lot more?

It is your definitely own grow old to take effect reviewing habit. accompanied by guides you could enjoy now is **Engineering Software To Related Projects** below.

KEY=SOFTWARE - GROSS JAMARCUS

PROJECT-BASED SOFTWARE ENGINEERING

AN OBJECT-ORIENTED APPROACH

Addison-Wesley *Project-Based Software Engineering is the first book to provide hands-on process and practice in software engineering essentials for the beginner. The book presents steps through the software development life cycle and two running case studies that develop as the steps are presented. Running parallel to the process presentation and case studies, the book supports a semester-long software development project. This book focuses on object-oriented software development, and supports the conceptualization, analysis, design and implementation of an object-oriented project. It is mostly language-independent, with necessary code examples in Java. A subset of UML is used, with the notation explained as needed to support the readers' work. Two running case studies a video game and a library check out system show the development of a software project. Both have sample deliverables and thus provide the reader with examples of the type of work readers are to create. This book is appropriate for readers looking to gain experience in project analysis, design implementation, and testing.*

REAL-WORLD SOFTWARE PROJECTS FOR COMPUTER SCIENCE AND ENGINEERING STUDENTS

Developing projects outside of a classroom setting can be intimidating for students and is not always a seamless process. Real-World Software Projects for Computer Science and Engineering Students is a quick, easy source for tackling such issues. Filling a critical gap in the research literature, the book: Is ideal for academic project supervisors. Helps researchers conduct interdisciplinary research. Guides computer science students on undertaking and implementing research-based projects This book explains how to develop highly complex, industry-specific projects, touching on real-world complexities of software developments. It shows how to develop projects *for students who have not yet had the chance to gain real-world experience, providing opportunity to become familiar with the skills needed to implement projects using standard development methodologies. The book is also a great source for teachers of undergraduate students in software engineering and computer science as it can help students prepare for risk and uncertainty that is typical of software development in industrial settings*

REQUIREMENTS ENGINEERING AND MANAGEMENT FOR SOFTWARE DEVELOPMENT PROJECTS

Springer Science & Business Media *Requirements Engineering and Management for Software Development Projects presents a complete guide on requirements for software development including engineering, computer science and management activities. It is the first book to cover all aspects of requirements management in software development projects. This book introduces the understanding of the requirements, elicitation and gathering, requirements analysis, verification and validation of the requirements, establishment of requirements, different methodologies in brief, requirements traceability and change management among other topics. The best practices, pitfalls, and metrics used for efficient software requirements management are also covered. Intended for the professional market, including software engineers, programmers, designers and researchers, this book is also suitable for advanced-level students in computer science or engineering courses as a textbook or reference.*

MANAGING AND LEADING SOFTWARE PROJECTS

John Wiley & Sons *The book is organized around basic principles of software project management: planning and estimating, measuring and controlling, leading and communicating, and managing risk. Introduces software development methods, from traditional (hacking, requirements to code, and waterfall) to iterative (incremental build, evolutionary, agile, and spiral). Illustrates and emphasizes tailoring the development process to each project, with a foundation in the fundamentals that are true for all development methods. Topics such as the WBS, estimation, schedule networks, organizing the project team, and performance reporting are integrated, rather than being relegating to appendices. Each chapter in the book includes an appendix that covers the relevant topics from CMMI-DEV-v1.2, IEEE/ISO Standards 12207, IEEE Standard 1058, and the PMI® Body of Knowledge. (PMI is a registered mark of Project Management Institute, Inc.)*

EVALUATING PROJECT DECISIONS

CASE STUDIES IN SOFTWARE ENGINEERING

Addison-Wesley Professional *Effective decisions are crucial to the success of any software project, but to make better decisions you need a better decision-making process. In Evaluating Project Decisions , leading project management experts introduce an innovative decision model that helps you tailor your decision-making process to systematically evaluate all of your decisions and avoid the bad choices* *that lead to project failure. Using a real-world, case study approach, the authors show how to evaluate software project problems and situations more effectively, thoughtfully assess your alternatives, and improve the decisions you make. Drawing on their own extensive research and experience, the authors bridge software engineering theory and practice, offering guidance that is both well-grounded and actionable. They present dozens of detailed examples from both successful and unsuccessful projects, illustrating what to do and what not to do. Evaluating Project Decisions will help you to analyze your options and ultimately make better decisions at every stage in your project, including: Requirements–Elicitation, description, verification, validation, negotiation, contracting, and management over the software life cycle Estimates–Conceptual solution design, decomposition, resource and overhead allocation, estimate construction, and change management Planning–Defining objectives, policies, and scope; planning tasks, milestones, schedules, budgets, staff and other resources; and managing projects against plans Product–Proper product definition, development process management, QA, configuration management, delivery, installation, training, and field service Process–Defining, selecting, understanding, teaching, and measuring processes; evaluating process performance; and process improvement or optimization In addition, you will see how to evaluate decisions related to risk, people, stakeholder expectations, and global development. Simply put, you'll use what you learn here on every project, in any industry, whatever your goals, and for projects of any duration, size, or type.*

SOFTWARE ENGINEERING RISK MANAGEMENT

GRIN Verlag *Research Paper (undergraduate) from the year 2004 in the subject Computer Science - Commercial Information Technology, grade: 1,0 (A), University Karlsruhe (TH) (Institute for Computer Science), 73 entries in the bibliography, language: English, abstract: While computer scientists have developed and provided several powerful computer languages and techniques in the last decades, facilitating the development of modular, maintainable and e±cient code, software development itself has changed fundamentally. Software development today treats often with large-scale projects, immense development costs, and complex sys- tems which typically deploy multiple technologies and require multiple participants for their development. As with any large development exercise, the development of a complex system must be systematic and structured in order to manage this complexity, and in order to make possible the future maintenance and evolution of the system. Thus, while systematic and structured approaches are necessary for the development of such systems, software engineers have attempted to provide the structured methodologies and formalisms so often lacking in large software development projects. However, software development projects are still related with many di®erent high risks. These risks cause software engineering projects to exceed bud- gets, miss deadlines, or deliver less than satisfactory products. As an example, U.S. companies alone spent an estimated $59 billion in cost overruns on IT projects and another $81 billion on cancelled software projects in 1995 (Johnson 1995). One reason for these high costs is that managers are not using adequate measures and* *executing e±cient risk management assess and mitigate the risks involved in these projects. Although risk taking is essential to progress, and failure is often a key part of learning, the inevitability of risks does not imply the inability to recognize and manage risks to minimize potential negative consequences while retaining the opportunities for creating new and better software. Obviously, this risk management process is particularly di±cult for large-scale soft- ware projects and be handled in the same way as for small project, or just by providing more resources for all development factors.*

KNOWLEDGE-BASED SOFTWARE ENGINEERING

PROCEEDINGS OF THE SEVENTH JOINT CONFERENCE ON KNOWLEDGE-BASED SOFTWARE ENGINEERING

IOS Press *"This publication addresses the research in theoretical foundations, practical techniques, software tools, applications and / or practical experiences in knowledge-based software engineering. The book also includes a new field: research in web services and semantic web. This is a rapidly developing research area promising to give excellent practical outcome, and interesting for theoretically minded as well as for practically minded people. The largest part of the papers belongs to a traditional area of applications of artificial intelligence methods to various software engineering problems. Another traditional section is application of intelligent agents in software engineering. A separate section is devoted to interesting applications and special techniques related in one or another way to the topic of the conference."*

SOFTWARE ENGINEERING PROJECT MANAGEMENT

John Wiley & Sons *About The Book: Richard Thayer s popular; bestselling book presents a top-down, practical view of managing a successful software engineering project. The book builds a framework for project management activities based on the planning, organizing, staffing, directing, and controlling model. Thayer provides information designed to help you understand and successfully perform the unique role of a project manager. This book is a must for all project managers in the software field. The text focuses on the five functions of general management by first describing each function and then detailing the project management activities that support each function. This new edition shows you how to manage a software development project, discusses current software engineering management methodologies and techniques, and presents general descriptions and project management problems. The book serves as a guide for your future project management activities. The text also offers students sufficient background and instructional material to serve as a main supplementary text for a course in software engineering project management. · Introduction to Management · Software Engineering · Software Engineering Project Management · Planning s Software Engineering Project · Planning: Software Cost, Schedule, and Size · Organizing a Software Engineering Project · Staffing a Software Engineering Project · Directing a Software Engineering Project · Controlling a Software Engineering Project ·*

Controlling: Software Metrics and Visibility of Progress

NEW SOFTWARE ENGINEERING PARADIGM BASED ON COMPLEXITY SCIENCE

AN INTRODUCTION TO NSE

Springer Science & Business Media *This book describes a complete revolution in software engineering based on complexity science through the establishment of NSE – Nonlinear Software Engineering paradigm which complies with the essential principles of complexity science, including the Nonlinearity principle, the Holism principle, the Complexity Arises From Simple Rules principle, the Initial Condition Sensitivity principle, the Sensitivity to Change principle, the Dynamics principle, the Openness principle, the Self-organization principle, and the Self-adaptation principle. The aims of this book are to offer revolutionary solutions to solve the critical problems existing with the old-established software engineering paradigm based on linear thinking and simplistic science complied with the superposition principle, and make it possible tohelp software development organizations double their productivity, halve their cost, and remove 99% to 99.99% of the defects in their software products, and efficiently handle software complexity, conformity, visibility, and changeability. It covers almost all areas in software engineering. The tools NSE_CLICK- an automatic acceptance testing platform for outsourcing (or internally developed) C/C++ products, and NSE_CLICK_J - an automatic acceptance testing platform for outsourcing (or internally developed) Java products are particularly designed for non-technical readers to view/review how the acceptance testing of a software product developed with NSE can be performed automatically, and how the product developed with NSE is truly maintainable at the customer site.*

REQUIREMENTS IN ENGINEERING PROJECTS

Springer *This book focuses on various topics related to engineering and management of requirements, in particular elicitation, negotiation, prioritisation, and documentation (whether with natural languages or with graphical models). The book provides methods and techniques that help to characterise, in a systematic manner, the requirements of the intended engineering system. It was written with the goal of being adopted as the main text for courses on requirements engineering, or as a strong reference to the topics of requirements in courses with a broader scope. It can also be used in vocational courses, for professionals interested in the software and information systems domain. Readers who have finished this book will be able to: - establish and plan a requirements engineering process within the development of complex engineering systems; - define and identify the types of relevant requirements in engineering projects; - choose and apply the most appropriate techniques to elicit the requirements of a given system; - conduct and manage negotiation and prioritisation processes for the requirements of a given engineering system; - document the requirements of the system under development, either in natural language or with graphical and formal models. Each chapter includes a set of exercises.*

WATER RELATED EDUCATION, TRAINING AND TECHNOLOGY TRANSFER

EOLSS Publications *Water Related Education, Training and Technology Transfer is a component of Encyclopedia of Water Sciences, Engineering and Technology Resources in the global Encyclopedia of Life Support Systems (EOLSS), which is an integrated compendium of twenty one Encyclopedias. Learning processes offer knowledge, skills, and competencies to the individual through different methods of education and training. The learning society and the concept of lifelong learning form the basis for the so-called “knowledge-based” economy. Since water resources development and management are an essential part of this economy, education, training, and transfer of technology for water resources should be seen as important aspects of societal policies for a sustainable future. This book starts with a little history, and introduces several issues related to water resources in the learning environment. What does the water profession expect from education? We must consider the methods and tools used the need to match demand and supply, and quality assessment of education and training. Transfer of technology to close the technology gap between countries can only be effective if an enabling learning environment exists. Capacity building must ensure that this environment is sustainable. This volume is aimed at the following five major target audiences: University and College students Educators, Professional practitioners, Research personnel and Policy analysts, managers, and decision makers and NGOs.*

SKILLS FOR MANAGING RAPIDLY CHANGING IT PROJECTS

IGI Global *"This book covers many aspects related to IT project management, such as human relationships, team management, software methodologies, and tools and techniques for project management"--Provided by publisher.*

REAL-WORLD SOFTWARE PROJECTS FOR COMPUTER SCIENCE AND ENGINEERING STUDENTS

CRC Press *Developing projects outside of a classroom setting can be intimidating for students and is not always a seamless process. Real-World Software Projects for Computer Science and Engineering Students is a quick, easy source for tackling such issues. Filling a critical gap in the research literature, the book: Is ideal for academic project supervisors. Helps researchers conduct interdisciplinary research. Guides computer science students on undertaking and implementing research-based projects This book explains how to develop highly complex, industry-specific projects touching on real-world complexities of software developments. It shows how to develop projects for students who have not yet had the chance to gain real-world experience, providing opportunity to become familiar with the skills needed to implement projects using standard development methodologies. The book is also a great source for teachers of undergraduate students in software engineering and computer science as it can help students prepare for the risk and uncertainty that is typical of software development in industrial settings.*

THE PROJECT MANAGER'S GUIDE TO SOFTWARE ENGINEERING'S BEST PRACTICES

Wiley-IEEE Computer Society Press *Since the earliest days of the computer industry, managing a software project has been a complex and demanding activity. While the technical content of software products and the technical methods used to build them have changed over time, the fundamental issues that determine the success or failure of software projects have remain fairly constant. That is, the same fundamental management mistakes continue to be made. To cite a few examples; requirements are unclear at the beginning of projects and are not managed during the project, the product is not tested adequately, schedules are misestimated or not tracked in sufficient detail. The contents of this book, together with the underlying IEEE Standards, are dedicated to helping the reader in their work: The continuing quest to produce quality software products in a predictable manner. This book, containing all original material, is based on the proposition that the IEEE Software Engineering Standards capture many of the fundamental 'best practices' of software project management. It is written to assist the reader in applying those standards to their projects and company. To meet this goal, the authors discuss and elaborate the standards that bear on the three key management areas of: Software systems engineering, Processes for developing software products, Planning and control of software project activities. The body of the book is correspondingly organized into three parts. Software Systems Engineering, which argues that software development projects are most successful when developed using a systems level viewpoint. Process Management and Control, which describes the key activities needed to define, support, and manage a project's software development processes. Project Planning and Management completes the book, integrating the elements of cost and schedule estimation and control, risk management, and the role metrics play in performing those tasks.*

KNOWLEDGE-BASED SOFTWARE ENGINEERING

PROCEEDINGS OF THE EIGHTH JOINT CONFERENCE ON KNOWLEDGE-BASED SOFTWARE ENGINEERING

IOS Press *The papers in this publication address many topics in the context of knowledge-based software engineering, including new challenges that have arisen in this demanding area of research. Topics in this book are: knowledge-based requirements engineering, domain analysis and modeling; development processes for knowledge-based applications; knowledge acquisition; software tools assisting the development; architectures for knowledge-based systems and shells including intelligent agents; intelligent user interfaces and human-machine interaction; development of multi-modal interfaces; knowledge technologies for semantic web; internet-based interactive applications; knowledge engineering for process management and project management; methodology and tools for knowledge discovery and data mining; knowledge-based methods and tools for testing, verification and validation, maintenance and evolution; decision support methods for software engineering and cognitive systems; knowledge management for business* *processes, workflows and enterprise modeling; program understanding, programming knowledge, modeling programs and programmers; and software engineering methods for intelligent tutoring systems.*

CONCISE GUIDE TO SOFTWARE ENGINEERING

FROM FUNDAMENTALS TO APPLICATION METHODS

Springer Nature *This textbook presents a concise introduction to the fundamental principles of software engineering, together with practical guidance on how to apply the theory in a real-world, industrial environment. The wide-ranging coverage encompasses all areas of software design, management, and quality. Topics and features: presents a broad overview of software engineering, including software lifecycles and phases in software development, and project management for software engineering; examines the areas of requirements engineering, software configuration management, software inspections, software testing, software quality assurance, and process quality; covers topics on software metrics and problem solving, software reliability and dependability, and software design and development, including Agile approaches; explains formal methods, a set of mathematical techniques to specify and derive a program from its specification, introducing the Z specification language; discusses software process improvement, describing the CMMI model, and introduces UML, a visual modelling language for software systems; reviews a range of tools to support various activities in software engineering, and offers advice on the selection and management of a software supplier; describes such innovations in the field of software as distributed systems, service-oriented architecture, software as a service, cloud computing, and embedded systems; includes key learning topics, summaries and review questions in each chapter, together with a useful glossary. This practical and easy-to-follow textbook/reference is ideal for computer science students seeking to learn how to build high quality and reliable software on time and on budget. The text also serves as a self-study primer for software engineers, quality professionals, and software managers.*

REQUIREMENTS ENGINEERING AND MANAGEMENT FOR SOFTWARE DEVELOPMENT PROJECTS

Springer Science & Business Media *Requirements Engineering and Management for Software Development Projects presents a complete guide on requirements for software development including engineering, computer science and management activities. It is the first book to cover all aspects of requirements management in software development projects. This book introduces the understanding of the requirements, elicitation and gathering, requirements analysis, verification and validation of the requirements, establishment of requirements, different methodologies in brief, requirements traceability and change management among other topics. The best practices, pitfalls, and metrics used for efficient software requirements management are also covered. Intended for the professional market, including software engineers, programmers, designers and researchers, this book is* *also suitable for advanced-level students in computer science or engineering courses as a textbook or reference.*

MANAGING SYSTEMS AND IT PROJECTS

Jones & Bartlett Learning *This book is designed for software engineering students and project management professional in the IT and software industry. It focuses on the four phases of management -- planning, organizing, monitoring, and adjusting (POMA) -- and tailors to systems and applications on software projects. The tasks and techniques utilized in each of the POMA management phases are discussed with specific software engineering and IT related examples. Drawing from years of experience in the industry, the author presents material within a framework of real-world examples and exercises that help readers apply new concepts to everyday situations.*

INFORMATION ASSURANCE APPLICATIONS IN SOFTWARE ENGINEERING PROJECTS

One USNA requirement (for computer science or IT undergraduates) is a capstone project. Students--in groups of three or four on a project of their choosing--must find a customer, define requirements, and meet key milestone dates in providing a software or system artifact. Projects require about 150 hours per person and must be completed and fully documented within the 15-week semester. Over the past two years, there has been increased student motivation to choose IA-related projects. Like software or systems engineering projects in other fields, students found it especially challenging to define customer requirements and meet expectations and milestones. Faculty use these challenges as learning opportunities by allowing students to make their own project decisions, even if poor decision making leads to a mid-project failure, because these failures will teach the students much more than a perfectly executed plan. Students found that taking on projects in the IA field of study created additional challenges in subject matter knowledge, system design, and implementation.

SOFTWARE ENGINEERING RESEARCH, MANAGEMENT AND APPLICATIONS

Springer *This edited book presents scientific results of the 12th International Conference on Software Engineering, Artificial Intelligence Research, Management and Applications (SERA 2014) held on August 31 – September 4, 2014 in Kitakyushu, Japan. The aim of this conference was to bring together researchers and scientists, businessmen and entrepreneurs, teachers, engineers, computer users, and students to discuss the numerous fields of computer science and to share their experiences and exchange new ideas and information in a meaningful way. Research results about all aspects (theory, applications and tools) of computer and information science, and to discuss the practical challenges encountered along the way and the solutions adopted to solve them. This publication captures 17 of the conference's most promising papers.*

PROJECT RELIABILITY ENGINEERING

PRO SKILLS FOR NEXT LEVEL MAKER PROJECTS

Apress *Turn your projects from a weekend hack to a long-living creation! Loosely drawing from the field known in large software companies as Site Reliability Engineering (SRE), this book distills from these disciplines and addresses issues that matter to makers: keeping projects up and running, and providing means to control, monitor, and troubleshoot them. Most examples use the Raspberry Pi, but the techniques discussed apply to other platforms as well. This book is all about breadth, and in the spirit of making, it visits different technologies as needed. However, the big goal in this book is to create a shift in the reader's mindset, where weekend hacks are pushed to the next level and are treated as products to be deployed. In that regard, this book can be a stepping stone for hobbyist makers into developing a broader, professional skill set. First, the book describes techniques for creating web-browser based dashboards for projects. These allow project creators to monitor, control, and troubleshoot their projects in real-time. Project Reliability Engineering discusses various aspects of the process of creating a web dashboard, such as network communication protocols, multithreading, and web design, and data visualization. Later chapters cover configuration of the project and the machine it's running on, and additional techniques for project monitoring and diagnosis. These include good logging practices; automatic log and metrics monitoring; and alerting via email and text messages; A mixture of advanced concepts forms the last chapter of the book, touching on topics such as usage of microservices in complex projects; debugging techniques for object-oriented projects; and fail-safing the project's software and hardware. What You'll Learn Monitor and control projects, keep them up and running, and troubleshoot them efficiently Get acquainted with available tools and libraries, and learn how to make your own tools Expand your knowledge in Python, JavaScript and Linux Develop deeper understanding of web technologies Design robust and complex systems Who This Book Is For Members of the maker community with some development skills.*

SOFTWARE ENGINEERING EDUCATION

7TH SEI CSEE CONFERENCE, SAN ANTONIO, TEXAS, USA, JANUARY 5-7, 1994. PROCEEDINGS

Springer Science & Business Media *While vols. III/29 A, B (published in 1992 and 1993, respectively) contains the low frequency properties of dielectric crystals, in vol. III/30 the high frequency or optical properties are compiled. While the first subvolume 30 A contains piezooptic and elastooptic constants, linear and quadratic electrooptic constants and their temperature coefficients, and relevant refractive indices, the present subvolume 30 B covers second and third order nonlinear optical susceptibilities. For the reader's convenience an alphabetical formula index and an alphabetical index of chemical, mineralogical and technical names for all substances of volumes 29 A, B and 30 A, B are included.*

THE DARK SIDE OF SOFTWARE ENGINEERING

EVIL ON COMPUTING PROJECTS

John Wiley & Sons *Betrayal! Corruption! Software engineering? Industry experts Johann Rost and Robert L. Glass explore the seamy underbelly of software engineering in this timely report on and analysis of the prevalance of subversion, lying, hacking, and espionage on every level of software project management. Based on the authors' original research and augmented by frank discussion and insights from other well-respected figures, The Dark Side of Software Engineering goes where other management studies fear to tread -- a corporate environment where schedules are fabricated, trust is betrayed, millions of dollars are lost, and there is a serious need for the kind of corrective action that this book ultimately proposes.*

SOFTWARE ENGINEERING: A HANDS-ON APPROACH

Springer Science & Business Media *This textbook provides a progressive approach to the teaching of software engineering. First, readers are introduced to the core concepts of the object-oriented methodology, which is used throughout the book to act as the foundation for software engineering and programming practices, and partly for the software engineering process itself. Then, the processes involved in software engineering are explained in more detail, especially methods and their applications in design, implementation, testing, and measurement, as they relate to software engineering projects. At last, readers are given the chance to practice these concepts by applying commonly used skills and tasks to a hands-on project. The impact of such a format is the potential for quicker and deeper understanding. Readers will master concepts and skills at the most basic levels before continuing to expand on and apply these lessons in later chapters.*

ENGINEERING AND MANAGING SOFTWARE REQUIREMENTS

Springer Science & Business Media *Requirements engineering is the process by which the requirements for software systems are gathered, analyzed, documented, and managed throughout their complete lifecycle. Traditionally it has been concerned with technical goals for, functions of, and constraints on software systems. Aurum and Wohlin, however, argue that it is no longer appropriate for software systems professionals to focus only on functional and non-functional aspects of the intended system and to somehow assume that organizational context and needs are outside their remit. Instead, they call for a broader perspective in order to gain a better understanding of the interdependencies between enterprise stakeholders, processes, and software systems, which would in turn give rise to more appropriate techniques and higher-quality systems. Following an introductory chapter that provides an exploration of key issues in requirements engineering, the book is organized in three parts. Part 1 presents surveys of state-of-the art requirements engineering process research along with critical assessments of existing models, frameworks and techniques. Part 2 addresses key areas in requirements engineering, such as market-driven requirements engineering, goal modeling, requirements ambiguity, and others. Part 3 concludes the book with* *articles that present empirical evidence and experiences from practices in industrial projects. Its broader perspective gives this book its distinct appeal and makes it of interest to both researchers and practitioners, not only in software engineering but also in other disciplines such as business process engineering and management science.*

SOFTWARE ENGINEERING

PHI Learning Pvt. Ltd. *The concepts, trends and practices in different phases of software development have taken sufficient advancement from the traditional ones. With these changes, methods of developing software, system architecture, software design, software coding, software maintenance and software project management have taken new shapes. Software Engineering discusses the principles, methodologies, trends and practices associated with different phases of software engineering. Starting from the basics, the book progresses slowly to advanced and emerging topics on software project management, process models, developing methodologies, software specification, testing, quality control, deployment, software security, maintenance and software reuse. Case study is a special feature of this book that discusses real life situation of dealing with IT related problems and finding their practical solutions in an easy manner. Elegant and simple style of presentation makes reading of this book a pleasant experience.Students of Computer Science and Engineering, Information Technology and Computer Applications should find this book highly useful. It would also be useful for IT technology professionals who are interested to get acquainted with the latest and the newest technologies.*

REQUIREMENTS ENGINEERING PROJECT-BASED LEARNING MODEL USING THE ELECTRONIC LEARNING SOFTWARE ENGINEERING SYSTEM (ELINS)

GRIN Verlag *Doctoral Thesis / Dissertation from the year 2014 in the subject Computer Science - Software, grade: Pass, University of Technology, Malaysia (Faculty of Computing), course: Software Engineering Education, language: English, abstract: The success of software project depends on how well it fits the needs of its user and its environment. This research strongly believes that future Requirement Engineering (RE) engineers should have the necessary generic skills in order to improve the quality of producing Software Requirement Specification. The software industry claims that the software engineering graduates are not able to meet their requirements for employability. Thus, confronting the problems right from the Higher Learning Education level that lead to this disparity will save the software industry the cost of sending new employees for additional training. The objectives of this research are to develop new learning environment model that can be implemented in RE education; construct a prototype namely Electronic Learning Software Engineering System (ELINS) that allows the industry, educators and Software Engineering (SE) undergraduate students to actively communicate and collaborate; and measure the effectiveness of the proposed learning model in teaching RE and enhancing the generic skills of SE undergraduates. This research comprises of pilot* *and main study to gather the requirement from experience of software industry personnel before evaluating the students after they involve in experimental test. The interview findings from the pilot study provided inputs which guide this research to develop the actual questionnaire for the main study. The study discusses the factors, causes, expected attributes, and importance of allowing undergraduates to improve their generic skills through actual hands-on participation. Rasch Measurement Model's software, WinStep, is used to analyze the raw data. In experimental test, students are provided with opportunities to practise how to deliver the SRS by doing several case studies from the software industry. The analysis and results have shown a positive improvement of the generic skills among the students who were involved in the Requirement Engineering Project Based-Learning (RE PjBL) model environment compared to those who were taught the course by traditional methods with minimal cost. The results conclude that the RE PjBL which are facilitated by ELINS can enhance student's knowledge, skills and attitude effectively.*

SOFTWARE SECURITY ENGINEERING

A GUIDE FOR PROJECT MANAGERS

Addison-Wesley Professional *Software Security Engineering draws extensively on the systematic approach developed for the Build Security In (BSI) Web site. Sponsored by the Department of Homeland Security Software Assurance Program, the BSI site offers a host of tools, guidelines, rules, principles, and other resources to help project managers address security issues in every phase of the software development life cycle (SDLC). The book's expert authors, themselves frequent contributors to the BSI site, represent two well-known resources in the security world: the CERT Program at the Software Engineering Institute (SEI) and Cigital, Inc., a consulting firm specializing in software security. This book will help you understand why Software security is about more than just eliminating vulnerabilities and conducting penetration tests Network security mechanisms and IT infrastructure security services do not sufficiently protect application software from security risks Software security initiatives should follow a risk-management approach to identify priorities and to define what is "good enough"–understanding that software security risks will change throughout the SDLC Project managers and software engineers need to learn to think like an attacker in order to address the range of functions that software should not do, and how software can better resist, tolerate, and recover when under attack*

SERVICE-ORIENTED SOFTWARE SYSTEM ENGINEERING

CHALLENGES AND PRACTICES

IGI Global *Current IT developments like component-based development and Web services have emerged as effective ways of building complex enterprise-scale information systems and providing enterprise application integration. To aid this process, platforms such as .NET and WebSphere have become standards in web-based systems development. However, there are still a lot of issues that need to be addressed before service-oriented software engineering (SOSE) becomes a* *prominent and widely accepted paradigm for enterprise information systems development and integration. This book provides a comprehensive view of SOSE through a number of different perspectives. Some of those perspectives include: service-based concepts, modeling and documentation, service discovery and composition, service-oriented architecture, model-driven development of service-oriented applications, service security and service-orientation in mobile settings. The book provides readers with an in-depth knowledge of the main challenges and practices in the exciting, new world of service-oriented software engineering. Addressing both technical and organizational aspects of this new field, it offers a balance making it valuable to a variety of readers, including IT architects, developers, managers, and analysts.*

HANDBOOK OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING

IN 2 VOLUMES

World Scientific *Readership: Graduate students, researchers, programmers, managers and academics in software engineering and knowledge engineering.Key Features: There are no other handbooks in the market in this area.Keywords:*

PROJECT MANAGEMENT WITH THE IBM RATIONAL UNIFIED PROCESS

LESSONS FROM THE TRENCHES

Prentice Hall Professional *· Master win–win techniques for managing outsourced and offshore projects, from procurement and risk mitigation to maintenance · Use RUP to implement best-practice project management throughout the software development lifecycle · Overcome key management challenges, from changing requirements to managing user expectations The Hands-On, Start-to-Finish Guide to Managing Software Projects with the IBM® Rational Unified Process® This is the definitive guide to managing software development projects with the IBM Rational Unified Process (RUP®). Drawing on his extensive experience managing projects with the RUP, R. Dennis Gibbs covers the entire development lifecycle, from planning and requirements to post-mortems and system maintenance. Gibbs offers especially valuable insights into using the RUP to manage outsourced projects and any project relying on distributed development teams—outsourced, insourced, or both. This “from the trenches” guidebook is invaluable for anyone interested in best practices for managing software development: project managers, team leaders, procurement and contracting specialists, quality assurance and software process professionals, consultants, and developers. If you’re already using the RUP, Gibbs will help you more effectively use it. Whatever your role or the RUP experience, you’ll learn ways to · Simplify and streamline the management of any large-scale or outsourced project · Overcome the challenges of using the RUP in software project management · Optimize software procurement and supplier relationships, from Request for Proposals (RFPs) and contracts to delivery · Staff high-performance project teams and project management offices · Establish productive, consistent development environments · Run effective project kickoffs · Systematically identify and mitigate* *project risks · Manage the technical and business challenges of changing requirements · Organize iterations and testing in incremental development processes · Transition new systems into service: from managing expectations to migrating data · Plan system maintenance and implement effective change control · Learn all you can from project post-mortems—and put those lessons into practice*

SOFTWARE PROJECT MANAGEMENT

A PROCESS-DRIVEN APPROACH

CRC Press *To build reliable, industry-applicable software products, large-scale software project groups must continuously improve software engineering processes to increase product quality, facilitate cost reductions, and adhere to tight schedules. Emphasizing the critical components of successful large-scale software projects, Software Project Management: A Process-Driven Approach discusses human resources, software engineering, and technology to a level that exceeds most university-level courses on the subject. The book is organized into five parts. Part I defines project management with information on project and process specifics and choices, the skills and experience needed, the tools available, and the human resources organization and management that brings it all together. Part II explores software life-cycle management. Part III tackles software engineering processes and the range of processing models devised by several domestic and international organizations. Part IV reveals the human side of project management with chapters on managing the team, the suppliers, and the customers themselves. Part V wraps up coverage with a look at the technology, techniques, templates, and checklists that can help your project teams meet and exceed their goals. A running case study provides authoritative insight and insider information on the tools and techniques required to ensure product quality, reduce costs, and meet project deadlines. Praise for the book: This book presents all aspects of modern project management practices ... includes a wealth of quality templates that practitioners can use to build their own tools. ... equally useful to students and professionals alike. —Maqbool Patel, PhD, SVP/CTO/Partner, Acuitec*

SOFTWARE MANAGEMENT

John Wiley & Sons *This Seventh Edition of Donald Reifer's popular, bestselling tutorial summarizes what software project managers need to know to be successful on the job. The text provides pointers and approaches to deal with the issues, challenges, and experiences that shape their thoughts and performance. To accomplish its goals, the volume explores recent advances in dissimilar fields such as management theory, acquisition management, globalization, knowledge management, licensing, motivation theory, process improvement, organization dynamics, subcontract management, and technology transfer. Software Management provides software managers at all levels of the organization with the information they need to know to develop their software engineering management strategies for now and the future. The book provides insight into management tools and techniques that work in practice. It also provides sufficient instructional* *materials to serve as a text for a course in software management. This new edition achieves a balance between theory and practical experience. Reifer systematically addresses the skills, knowledge, and abilities that software managers, at any level of experience, need to have to practice their profession effectively. This book contains original articles by leaders in the software management field written specifically for this tutorial, as well as a collection of applicable reprints. About forty percent of the material in this edition has been produced specifically for the tutorial. Contents: * Introduction * Life Cycle Models * Process Improvement * Project Management * Planning Fundamentals * Software Estimating * Organizing for Success * Staffing Essentials * Direction Advice * Visibility and Control * Software Risk Management * Metrics and Measurement * Acquisition Management * Emerging Management Topics "The challenges faced by software project managers are the gap between what the customers can envision and the reality on the ground and how to deal with the risks associated with this gap in delivering a product that meets requirements on time and schedule at the target costs. This tutorial hits the mark by providing project managers, practitioners, and educators with source materials on how project managers can effectively deal with this risk." -Dr. Kenneth E. Nidiffer, Systems & Software Consortium, Inc. "The volume has evolved into a solid set of foundation works for anyone trying to practice software management in a world that is increasingly dependent on software release quality, timeliness, and productivity." -Walker Royce, Vice President, IBM Software Services-Rational*

BUSINESS COMPONENT-BASED SOFTWARE ENGINEERING

Springer Science & Business Media *Business Component-Based Software Engineering, an edited volume, aims to complement some other reputable books on CBSE, by stressing how components are built for large-scale applications, within dedicated development processes and for easy and direct combination. This book will emphasize these three facets and will offer a complete overview of some recent progresses. Projects and works explained herein will prompt graduate students, academics, software engineers, project managers and developers to adopt and to apply new component development methods gained from and validated by the authors. The authors of Business Component-Based Software Engineering are academic and professionals, experts in the field, who will introduce the state of the art on CBSE from their shared experience by working on the same projects. Business Component-Based Software Engineering is designed to meet the needs of practitioners and researchers in industry, and graduate-level students in Computer Science and Engineering.*

SOFTWARE ENGINEERING: PRINCIPLES AND PRACTICES, 2ND EDITION

Vikas Publishing House *This revised edition of Software Engineering-Principles and Practices has become more comprehensive with the inclusion of several topics. The book now offers a complete understanding of software engineering as an engineering discipline. Like its previous edition, it provides an in-depth coverage of fundamental principles, methods and applications of software engineering. In addition, it covers some advanced approaches including Computer-aided Software*

Engineering (CASE), Component-based Software Engineering (CBSE), Clean-room Software Engineering (CSE) and formal methods.Taking into account the needs of both students and practitioners, the book presents a pragmatic picture of the software engineering methods and tools. A thorough study of the software industry shows that there exists a substantial difference between classroom study and the practical industrial application. Therefore, earnest efforts have been made in this book to bridge the gap between theory and practical applications. The subject matter is well supported by examples and case studies representing the situations that one actually faces during the software development process.The book meets the requirements of students enrolled in various courses both at the undergraduate and postgraduate levels, such as BCA, BE, BTech, BIT, BIS, BSc, PGDCA, MCA, MIT, MIS, MSc, various DOEACC levels and so on. It will also be suitable for those software engineers who abide by scientific principles and wish to expand their knowledge. With the increasing demand of software, the software engineering discipline has become important in education and industry. This thoughtfully organized second edition of the book provides its readers a profound knowledge of software engineering concepts and principles in a simple, interesting and illustrative manner.

EMPIRICAL METHODS AND STUDIES IN SOFTWARE ENGINEERING

EXPERIENCES FROM ESERNET

Springer Science & Business Media *Nowadays, societies crucially depend on high-quality software for a large part of their functionalities and activities. Therefore, software professionals, researchers, managers, and practitioners alike have to competently decide what software technologies and products to choose for which purpose. For various reasons, systematic empirical studies employing strictly scientific methods are hardly practiced in software engineering. Thus there is an unquestioned need for developing improved and better-qualified empirical methods, for their application in practice and for dissemination of the results. This book describes different kinds of empirical studies and methods for performing such studies, e.g., for planning, performing, analyzing, and reporting such studies. Actual studies are presented in detail in various chapters dealing with inspections, testing, object-oriented techniques, and component-based software engineering.*

AGILE SOFTWARE DEVELOPMENT QUALITY ASSURANCE

IGI Global "*This book provides the research and instruction used to develop and implement software quickly, in small iteration cycles, and in close cooperation with the customer in an adaptive way, making it possible to react to changes set by the constant changing business environment. It presents four values explaining extreme programming (XP), the most widely adopted agile methodology"--Provided by publisher.*

SOFTWARE PROJECT SURVIVAL GUIDE

Microsoft Press *Equip yourself with SOFTWARE PROJECT SURVIVAL GUIDE. It's for everyone with a stake in the outcome of a development project--and especially for* *those without formal software project management training. That includes top managers, executives, clients, investors, end-user representatives, project managers, and technical leads. Here you'll find guidance from the acclaimed author of the classics CODE COMPLETE and RAPID DEVELOPMENT. Steve McConnell draws on solid research and a career's worth of hard-won experience to map the surest path to your goal--what he calls "one specific approach to software development that works pretty well most of the time for most projects." Nineteen chapters in four sections cover the concepts and strategies you need for mastering the development process, including planning, design, management, quality assurance, testing, and archiving. For newcomers and seasoned project managers alike, SOFTWARE PROJECT SURVIVAL GUIDE draws on a vast store of techniques to create an elegantly simplified and reliable framework for project management success. So don't worry about wandering among complex sets of project management techniques that require years to sort out and master. SOFTWARE PROJECT SURVIVAL GUIDE goes straight to the heart of the matter to help your projects succeed. And that makes it a required addition to every professional's bookshelf.*

SOFTWARE ENGINEERING

PRINCIPLES AND PRACTICE

This work aims to provide the reader with sound engineering principles, whilst embracing relevant industry practices and technologies, such as object orientation and requirements engineering. It includes a chapter on software architectures, covering software design patterns.

SOFTWARE ENGINEERING PROJECT MANAGEMENT KNOWLEDGE AREAS

VOLUME 12: THE ENGIEERING OF SOFTWARE PROJECTS

This book serves four separate but connected audiences: (1) This book expands on the software engineering outline expressed in SWEBOK, Version 3.0, i.e., to provide the "meat-on-the bones" where SWEBOK is the "bones. (2) When used as a software engineering tutorial, it can be used to provide a detailed software engineering education to university-level software engineering students. (3)When used as a software engineering study guide, this document can impart software engineering knowledge to assist practicing software engineers to take and pass the new IEEE Professional Software Engineering Master (PSEM) Certification exams. (4) When used as a software engineering overview, this book can be referenced by journeyman programmers to improve their background and understanding of software engineering fundamentals. This book will provide a comprehensive overview of software engineering knowledge and skills necessary for a well-qualified programmer to become an entry level "software engineer."