Download Ebook Pdf Paperback Engineering Software In Series Sei Implementation To Architecture From Warehouse Data

Recognizing the artifice ways to get this book **Pdf Paperback Engineering Software In Series Sei Implementation To Architecture From Warehouse Data** is additionally useful. You have remained in right site to start getting this info. get the Pdf Paperback Engineering Software In Series Sei Implementation To Architecture From Warehouse Data member that we pay for here and check out the link.

You could purchase lead Pdf Paperback Engineering Software In Series Sei Implementation To Architecture From Warehouse Data or acquire it as soon as feasible. You could quickly download this Pdf Paperback Engineering Software In Series Sei Implementation To Architecture From Warehouse Data after getting deal. So, with you require the book swiftly, you can straight acquire it. Its suitably categorically easy and consequently fats, isnt it? You have to favor to in this ventilate

KEY=PAPERBACK - EDWARDS JESSIE

Modernizing Legacy Systems

Software Technologies, Engineering Processes, and Business Practices

Addison-Wesley Professional Most organizations rely on complex enterprise information systems (EISs) to codify their business practices and collect, process, and analyze business data. These EISs are large, heterogeneous, distributed, constantly evolving, dynamic, long-lived, and mission critical. In other words, they are a complicated system of systems. As features are added to an EIS, new technologies and components are selected and integrated. In many ways, these information systems are to an enterprise what a brain is to the higher species--a complex, poorly understood mass upon which the organism relies for its very existence. To optimize business value, these large, complex systems must be modernized--but where does one begin? This book uses an extensive real-world case study (based on the modernization of a thirty year old retail system) to show how modernizing legacy systems can deliver significant business value to any organization.

Future-Proof Software-Systems

A Sustainable Evolution Strategy

Springer This book focuses on software architecture and the value of architecture in the development of long-lived, mission-critical, trustworthy software-systems. The author introduces and demonstrates the powerful strategy of "Managed Evolution," along with the engineering best practice known as "Principle-based Architecting." The book examines in detail architecture principles for e.g., Business Value, Changeability, Resilience, and Dependability. The author argues that the software development community has a strong responsibility to produce and operate useful, dependable, and trustworthy software. Software should at the same time provide business value and guarantee many quality-of-service properties, including security, safety, performance, and integrity. As Dr. Furrer states, "Producing dependable software is a balancing act between investing in the implementation of business functionality and investing in the quality-of-service properties of the software-systems." The book presents extensive coverage of such concepts as: Principle-Based Architecting Managed Evolution Strategy The Future Principles for Business Value Legacy Software Modernization/Migration Architecture Principles for Changeability Architecture Principles for Resilience Architecture Principles for Dependability The text is supplemented with numerous figures, tables, examples and illustrative quotations. Future-Proof Software-Systems provides a set of good engineering practices, devised for integration into most software development processes dedicated to the creation of software-

systems that incorporate Managed Evolution.

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.

Secure Coding in C and C++

Pearson Education "The security of information systems has not improved at a rate consistent with the growth and sophistication of the attacks being made against them. To address this problem, we must improve the underlying strategies and techniques used to create our systems. Specifically, we must build security in from the start, rather than append it as an afterthought. That's the point of Secure Coding in C and C++. In careful detail, this book shows software developers how to build high-quality systems that are less vulnerable to costly and even catastrophic attack. It's a book that every developer should read before the start of any serious project." --Frank Abagnale, author, lecturer, and leading consultant on fraud prevention and secure documents Learn the Root Causes of Software Vulnerabilities and How to Avoid Them Commonly exploited software vulnerabilities are usually caused by avoidable software defects. Having analyzed nearly 18,000 vulnerability reports over the past ten years, the CERT/Coordination Center (CERT/CC) has determined that a relatively small number of root causes account for most of them. This book identifies and explains these causes and shows the steps that can be taken to prevent exploitation. Moreover, this book encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow's attacks, not just today's. Drawing on the CERT/CC's reports and conclusions, Robert Seacord systematically identifies the program errors most likely to lead to security breaches, shows how they can be exploited, reviews the potential consequences, and presents secure alternatives. Coverage includes technical detail on how to Improve the overall security of any C/C++ application Thwart buffer overflows and stack-smashing attacks that exploit insecure string manipulation logic Avoid vulnerabilities and security flaws resulting from the incorrect use of dynamic memory management functions Eliminate integer-related problems: integer overflows, sign errors, and truncation errors Correctly use formatted output functions without introducing format-string vulnerabilities Avoid I/O vulnerabilities, including race conditions Secure Coding in C and C++ presents hundreds of examples of secure code, insecure code, and exploits, implemented for Windows and Linux. If you're responsible for creating secure C or C++ software--or for keeping it safe--no other book offers you this much detailed, expert assistance.

The Method Framework for Engineering System Architectures

CRC Press The architects of today's large and complex systems all too often struggle with the lack of a consistent set of principles and practices that adequately address the entire breadth of systems architecture. The Method Framework for Engineering System Architectures (MFESA) enables system architects and process engineers to create methods for effective

Agile Manufacturing Systems

ALPHA SCIENCE INTERNATIONAL LIMITED Agility has become very important for the industries today as the lifetimes of the products are continuously shrinking. This book provides an excellent opportunity for updating understanding of agile methods from the design, manufacturing and business process perspectives, whether one is an industrial practitioner, academic researcher engineer or business graduate student. This volume is a compilation of various important aspects of agility consisting of systemic considerations in

manufacturing, agile software systems, agile business systems, agile operations research, flexible manufacturing systems, advanced manufacturing systems with improved materials and mechanical behavior of products, agile aspects of design, clean and green manufacturing systems, environment, agile defence systems.

A Discipline for Software Engineering

Addison-Wesley Professional This new work from Watts Humphrey, author of the influential book, Managing the Software Process, broadens his orderly view of software process management, and lays the foundation for a disciplined approach to software engineering. In his earlier book, the author developed concrete methods for managing software development and maintenance. These methods, now commonly practiced in industry, provide programmers and managers with specific steps they can take to evaluate and improve their software capabilities. In this new book, Humphrey scales those methods down to a personal level, helping software engineers develop the skills and habits needed to plan, track, and analyze large, complex projects. Humphrey and others have used material from this book to train professionals and students around the world in a projects-oriented software engineering course. First establishing the need for discipline in software engineering, and the benefits to practitioners of learning how to manage their personal software process, Humphrey then develops a model that they can use to monitor, test, and improve their work. Examples drawn from industry enhance the practical focus of the book, while project exercises give readers the opportunity to practice software process management as they learn it. Features: presents concepts and methods for a disciplined software engineering process; scales down industrial practices for planning, tracking, analysis, and defect management to fit the needs of small-scale program development; and shows how small project disciplines provide a solid base for larger projects.

Software Architecture in Practice

Addison-Wesley The award-winning and highly influential Software Architecture in Practice, Third Edition, has been substantially revised to reflect the latest developments in the field. In a real-world setting, the book once again introduces the concepts and best practices of software architecture—how a software system is structured and how that system's elements are meant to interact. Distinct from the details of implementation, algorithm, and data representation, an architecture holds the key to achieving system quality, is a reusable asset that can be applied to subsequent systems, and is crucial to a software organization's business strategy. The authors have structured this edition around the concept of architecture influence cycles. Each cycle shows how architecture influences, and is influenced by, a particular context in which architecture plays a critical role. Contexts include technical environment, the life cycle of a project, an organization's business profile, and the architect's professional practices. The authors also have greatly expanded their treatment of quality attributes, which remain central to their architecture philosophy—with an entire chapter devoted to each attribute—and broadened their treatment of architectural patterns. If you design, develop, or manage large software systems (or plan to do so), you will find this book to be a valuable resource for getting up to speed on the state of the art. Totally new material covers Contexts of software architecture: technical, project, business, and professional Architecture competence: what this means both for individuals and organizations The origins of business goals and how this affects architecture Architecturally significant requirements, and how to determine them Architecture in the life cycle, including generate-and-test as a design philosophy; architecture conformance during implementation; architecture and testing; and architecture and agile development Architecture and current technologies, such as the cloud, social networks, and end-user devices

Competitive Engineering

A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage

Elsevier Competitive Engineering documents Tom Gilb's unique, ground-breaking approach to communicating management objectives and systems engineering requirements, clearly and unambiguously. Competitive Engineering is a revelation for anyone involved in management and risk control. Already used by thousands of project managers and systems engineers around the world, this is a handbook for initiating, controlling and delivering complex projects on time and within budget. The Competitive Engineering methodology

provides a practical set of tools and techniques that enable readers to effectively design, manage and deliver results in any complex organization - in engineering, industry, systems engineering, software, IT, the service sector and beyond. Elegant, comprehensive and accessible, the Competitive Engineering methodology provides a practical set of tools and techniques that enable readers to effectively design, manage and deliver results in any complex organization - in engineering, industry, systems engineering, software, IT, the service sector and beyond. Provides detailed, practical and innovative coverage of key subjects including requirements specification, design evaluation, specification quality control and evolutionary project management Offers a complete, proven and meaningful 'end-to-end' process for specifying, evaluating, managing and delivering high quality solutions Tom Gilb's clients include HP, Intel, CitiGroup, IBM, Nokia and the US Department of Defense

Knowledge Networks: The Social Software Perspective

The Social Software Perspective

IGI Global "This book concentrates on strategies that exploit emerging technologies for the knowledge effectiveness in social networks"--Provided by publisher.

Touch of Class

Learning to Program Well with Objects and Contracts

Springer Science & Business Media This text combines a practical, hands-on approach to programming with the introduction of sound theoretical support focused on teaching the construction of high-quality software. A major feature of the book is the use of Design by Contract.

Evaluating Software Architectures

Pearson Education India This Book Describes Systematic Methods For Evaluating Software Architectures And Applies Them To Real-Life Cases. Evaluating Software Architectures Introduces The Conceptual Background For Architecture Evaluation And Provides A Step-By-Step Guide To The Process Based On Numerous Evaluations Performed In Government And Industry.

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.

Process Assessment and ISO/IEC 15504

A Reference Book

Springer Science & Business Media Helps readers understand the power and benefits of a process approach and process assessment. Guides the reader through the various parts of the standard in an understandable and practical manner.

Safety and Security of Cyber-Physical Systems

Engineering dependable Software using Principle-based Development

Springer Nature Cyber-physical systems (CPSs) consist of software-controlled computing devices communicating with each other and interacting with the physical world through sensors and actuators. A CPS has, therefore, two parts: The cyber part implementing most of the functionality and the physical part, i.e., the real world. Typical examples of CPS's are a water treatment plant, an unmanned aerial vehicle, and a heart pacemaker. Because most of the functionality is implemented in software, the software is of crucial importance. The software determines the functionality and many CPS properties, such as safety, security, performance, real-time behavior, etc. Therefore, avoiding safety accidents and security incidents in the CPS requires highly dependable software. Methodology Today, many methodologies for developing safe and secure software are in use. As software engineering slowly becomes disciplined and mature, generally accepted construction principles have emerged. This monograph advocates principle-based engineering for the development and operation of dependable software. No new development process is suggested, but integrating security and safety principles into existing development processes is demonstrated. Safety and Security Principles At the core of this monograph are the engineering principles. A total of 62 principles are introduced and catalogized into five categories: Business & organization, general principles, safety, security, and risk management principles. The principles are rigorous, teachable, and enforceable. The terminology used is precisely defined. The material is supported by numerous examples and enriched by illustrative quotes from celebrities in the field. Final Words «In a cyber-physical system's safety and security, any compromise is a planned disaster» Audience First, this monograph is for organizations that want to improve their methodologies to build safe and secure software for mission-critical cyber-physical systems. Second, the material is suitable for a two-semester, 4 hours/week, advanced computer science lecture at a Technical University. This textbook has been recommended and developed for university courses in Germany, Austria and Switzerland.

Java Coding Standards

Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs (SEI

Series in Software Engineering)

Harry Hariom Choudhary "A must-read for all Java developers. . . . Every developer has a responsibility to author code that is free of significant security vulnerabilities. This book provides realistic guidance to help Java developers implement desired functionality with security, reliability, and maintainability goals in mind." –Mary Ann Davidson, Chief Security Officer, Oracle Corporation Organizations worldwide rely on Java code to perform mission-critical tasks, and therefore that code must be reliable, robust, fast, maintainable, and secure. Java™ Coding Guidelines brings together expert guidelines, recommendations, and code examples to help you meet these demands. Written by the same team that brought you The CERT® Oracle ® Secure Coding Standard for Java™, this guide extends that previous work's expert security advice to address many additional quality attributes. You'll find 75 guidelines, each presented consistently and intuitively. For each guideline, conformance requirements are specified; for most, noncompliant code examples and compliant solutions are also offered. The authors explain when to apply each guideline and provide references to even more detailed information. Reflecting pioneering research on Java security, Java™ Coding Guidelines offers updated techniques for protecting against both deliberate attacks and other unexpected events. You'll find best practices for improving code reliability and clarity, and a full chapter exposing common misunderstandings that lead to suboptimal code. With a Foreword by James A. Gosling, Father of the Java Programming Language

Advances in Software Engineering, Education, and e-Learning

Proceedings from FECS'20, FCS'20, SERP'20, and EEE'20

Springer Nature This book presents the proceedings of four conferences: The 16th International Conference on Frontiers in Education: Computer Science and Computer Engineering + STEM (FECS'20), The 16th International Conference on Foundations of Computer Science (FCS'20), The 18th International Conference on Software Engineering Research and Practice (SERP'20), and The 19th International Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government (EEE'20). The conferences took place in Las Vegas, NV, USA, July 27-30, 2020 as part of the larger 2020 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'20), which features 20 major tracks. Authors include academics, researchers, professionals, and students. This book contains an open access chapter entitled, "Advances in Software Engineering, Education, and e-Learning". Presents the proceedings of four conferences as part of the 2020 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'20); Includes the tracks Computer Engineering + STEM, Foundations of Computer Science, Software Engineering Research, and e-Learning, e-Business, Enterprise Information Systems, & e-Government; Features papers from FECS'20, FCS'20, SERP'20, EEE'20, including one open access chapter.

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

Handbook Of Software Engineering And Knowledge Engineering, Vol 3: Recent Advances

World Scientific The book covers the recent new advances in software engineering and knowledge engineering. It is intended as a supplement to the two-volume handbook of software engineering and knowledge engineering. The editor and authors are well-known international experts in their respective fields of expertise. Each chapter in the book is entirely self-contained and gives in-depth information on a specific topic of current interest. This book will be a useful desktop companion for both practitioners and students of software engineering and knowledge engineering.

Beyond Algorithms

Delivering AI for Business

CRC Press With so much artificial intelligence (AI) in the headlines, it is no surprise that businesses are scrambling to exploit this exciting and transformative technology. Clearly, those who are the first to deliver business-relevant AI will gain significant advantage. However, there is a problem! Our perception of AI success in society is primarily based on our experiences with consumer applications from the big web companies. The adoption of AI in the enterprise has been slow due to various challenges. Business applications address far more complex problems and the data needed to address them is less plentiful. There is also the critical need for alignment of AI with relevant business processes. In addition, the use of AI requires new engineering practices for application maintenance and trust. So, how do you deliver working AI applications in the enterprise? Beyond Algorithms: Delivering AI for Business answers this question. Written by three engineers with decades of experience in AI (and all the scars that come with that), this book explains what it takes to define, manage, engineer, and deliver end-to-end AI applications that work. This book presents: Core conceptual differences between AI and traditional business applications A new methodology that helps to prioritise AI projects and manage risks Practical case studies and examples with a focus on business impact and solution delivery Technical Deep Dives and Thought Experiments designed to challenge your brain and destroy your weekends

Agile Enterprise Engineering: Smart Application of Human Factors

Models, Methods, Practices, Case Studies

Springer Nature This concise book provides a survival toolkit for efficient, large-scale software development. Discussing a multi-contextual research framework that aims to harness human-related factors in order to improve flexibility, it includes a carefully selected blend of models, methods, practices, and case studies. To investigate mission-critical communication aspects in system engineering, it also examines diverse, i.e. cross-cultural and multinational, environments. This book helps students better organize their knowledge bases, and presents conceptual frameworks, handy practices and case-based examples of agile development in diverse environments. Together with the authors' previous books, "Crisis Management for Software Development and Knowledge Transfer" (2016) and "Managing Software Crisis: A Smart Way to Enterprise Agility" (2018), it constitutes a comprehensive reference resource adds value to this book.

Agile Data Warehousing Project Management

Business Intelligence Systems Using Scrum

Newnes You have to make sense of enormous amounts of data, and while the notion of "agile data warehousing might sound tricky, it can yield as much as a 3-to-1 speed advantage while cutting project costs in half. Bring this highly effective technique to your organization with the wisdom of agile data warehousing expert Ralph Hughes. Agile Data Warehousing Project Management will give you a thorough introduction to the method as you would practice it in the project room to build a serious "data mart. Regardless of where you are today, this step-by-step implementation guide will prepare you to join or even lead a team in visualizing, building, and validating a single component to an enterprise data warehouse. Provides a thorough grounding on the mechanics of Scrum as well as practical advice on keeping your team on track Includes strategies for getting accurate and actionable requirements from a team's business partner Revolutionary estimating techniques that make forecasting labor far more understandable and accurate Demonstrates a blends of Agile methods to simplify team management and synchronize inputs across IT specialties Enables you and your teams to start simple and progress steadily to world-class performance levels

Cyber Security Engineering

A Practical Approach for Systems and Software Assurance

Addison-Wesley Professional Cyber Security Engineering is the definitive modern reference and tutorial on the full range of capabilities associated with modern cyber security engineering. Pioneering software assurance experts Dr. Nancy R. Mead and Dr. Carol C. Woody bring together comprehensive best practices for building software systems that exhibit superior operational security, and for considering security throughout your full system development and acquisition lifecycles. Drawing on their pioneering work at the Software Engineering Institute (SEI) and Carnegie Mellon University, Mead and Woody introduce seven core principles of software assurance, and show how to apply them coherently and systematically. Using these principles, they help you prioritize the wide range of possible security actions available to you, and justify the required investments. Cyber Security Engineering guides you through risk analysis, planning to manage secure software development, building organizational models, identifying required and missing competencies, and defining and structuring metrics. Mead and Woody address important topics, including the use of standards, engineering security requirements for acquiring COTS software, applying DevOps, analyzing malware to anticipate future vulnerabilities, and planning ongoing improvements. This book will be valuable to wide audiences of practitioners and managers with responsibility for systems, software, or quality engineering, reliability, security, acquisition, or operations. Whatever your role, it can help you reduce operational problems, eliminate excessive patching, and deliver software that is more resilient and secure.

Documenting Software Architectures

Views and Beyond

Pearson Education Software architecture—the conceptual glue that holds every phase of a project together for its many stakeholders—is widely recognized as a critical element in modern software development. Practitioners have increasingly discovered that close attention to a software system's architecture pays valuable dividends. Without an architecture that is appropriate for the problem being solved, a project will stumble along or, most likely, fail. Even with a superb architecture, if that architecture is not well understood or well communicated the project is unlikely to succeed. Documenting Software Architectures, Second Edition, provides the most complete and current guidance, independent of language or notation, on how to capture an architecture in a commonly understandable form. Drawing on their extensive experience, the authors first help you decide what information to document, and then, with guidelines and examples (in various notations, including UML), show you how to express an architecture so that others can successfully build, use, and maintain a system from it. The book features rules for sound documentation, the goals and strategies of documentation, architectural views and styles, documentation for software interfaces and software behavior, and templates for capturing and organizing information to generate a coherent package. New and improved in this second edition: Coverage of architectural styles such as service-oriented architectures, multi-tier architectures, and data models Guidance for documentation in an Agile development environment Deeper

treatment of documentation of rationale, reflecting best industrial practices Improved templates, reflecting years of use and feedback, and more documentation layout options A new, comprehensive example (available online), featuring documentation of a Web-based service-oriented system Reference guides for three important architecture documentation languages: UML, AADL, and SySML

Software Engineering Mathematics

Addison-Wesley

Availability, Reliability, and Security in Information Systems

IFIP WG 8.4, 8.9, TC 5 International Cross-Domain Conference, CD-ARES 2014 and 4th InternationalWorkshop on Security and Cognitive Informatics for Homeland Defense, SeCIHD 2014, Fribourg, Switzerland, September 8-12, 2014. Proceedings

Springer This volume constitutes the refereed proceedings of two workshops: the International Cross-Domain Conference and Workshop on Availability, Reliability and Security, CD-ARES 2014, and the 4th International Workshop on Security and Cognitive Informatics for Homeland Defense, SeCIHD 2014, co-located with the International Conference on Availability, Reliability and Security, ARES 2014, held in Fribourg, Switzerland, in September 2014. The 23 revised full papers presented were carefully reviewed and selected from numerous submissions. The papers deal with knowledge management, software security, mobile and social computing, enterprise information systems, homeland security and information processing.

Documenting Software Architectures

Views and Beyond

Addison-Wesley Professional Architecture is crucial to the success of any large software system -- but even a superb architecture will fail if it isn't communicated well. Now, there's a language- and notation-independent guide to capturing architecture so it can be used successfully by every analyst, software designer, and developer. The authors review the diverse goals and uses of software architecture documentation, providing documentation strategies for several common scenarios. They identify the basic unit of software architecture documentation: the viewtype, which specifies the type of information to be provided in an architectural view. For each viewtype -- Modules, Component-and-Connectors, and Allocation -- they offer detailed guidance on documenting what really matters. Next, they demonstrate how to package architecture documentation in coherent, usable form: augmenting architectural views with documentation of interfaces and behavior; accounting for architectural variability and dynamic systems; and more.

Handbook of Research on Distributed Medical Informatics and E-Health

IGI Global Provides coverage of specific topics and issues in healthcare, highlighting recent trends and describing the latest advances in the field.

Reconstruction of Software Component Architectures and Behaviour Models Using Static and Dynamic Analysis

KIT Scientific Publishing Model-based performance prediction systematically deals with the evaluation of software performance to avoid for example bottlenecks, estimate execution environment sizing, or identify scalability limitations for new usage scenarios. Such performance predictions require up-to-date software performance models. This book describes a new integrated reverse engineering approach for the reconstruction of parameterised software performance models (software component architecture and behaviour).

Evaluating Software Architectures

Methods and Case Studies

Addison-Wesley Professional The foundation of any software system is its architecture. Using this book, you can evaluate every aspect of architecture in advance, at remarkably low cost -- identifying improvements that can dramatically improve any system's performance, security, reliability, and maintainability. As the practice of software architecture has matured, it has become possible to identify causal connections between architectural design decisions and the qualities and properties that result downstream in the systems that follow from them. This book shows how, offering step-by-step guidance, as well as detailed practical examples -- complete with sample artifacts reflective of those that evaluators will encounter. The techniques presented here are applicable not only to software architectures, but also to system architectures encompassing computing hardware, networking equipment, and other elements. For all software architects, software engineers, developers, IT managers, and others responsible for creating, evaluating, or implementing software architectures.

Innovations in Computing Sciences and Software Engineering

Springer Science & Business Media Innovations in Computing Sciences and Software Engineering includes a set of rigorously reviewed world-class manuscripts addressing and detailing state-of-the-art research projects in the areas of Computer Science, Software Engineering, Computer Engineering, and Systems Engineering and Sciences. Topics Covered: •Image and Pattern Recognition: Compression, Image processing, Signal Processing Architectures, Signal Processing for Communication, Signal Processing Implementation, Speech Compression, and Video Coding Architectures. •Languages and Systems: Algorithms, Databases, Embedded Systems and Applications, File Systems and I/O, Geographical Information Systems, Kernel and OS Structures, Knowledge Based Systems, Modeling and Simulation, Object Based Software Engineering, Programming Languages, and Programming Models and tools. •Parallel Processing: Distributed Scheduling, Multiprocessing, Real-time Systems, Simulation Modeling and Development, and Web Applications. •Signal and Image Processing: Content Based Video Retrieval, Character Recognition, Incremental Learning for Speech Recognition, Signal Processing Theory and Methods, and Vision-based Monitoring Systems. •Software and Systems: Activity-Based Software Estimation, Algorithms, Genetic Algorithms, Information Systems Security, Programming Languages, Software Protection Techniques, Software Protection Techniques, and User Interfaces. •Distributed Processing: Asynchronous Message Passing System, Heterogeneous Software Environments, Mobile Ad Hoc Networks, Resource Allocation, and Sensor Networks. •New trends in computing: Computers for People of Special Needs, Fuzzy Inference, Human Computer Interaction, Incremental Learning, Internet-based Computing Models, Machine Intelligence, Natural Language.

Encyclopedia of Software Engineering Three-Volume Set (Print)

CRC Press Software engineering requires specialized knowledge of a broad spectrum of topics, including the construction of software and the platforms, applications, and environments in which the software operates as well as an understanding of the people who build and use the software. Offering an authoritative perspective, the two volumes of the Encyclopedia of Software Engineering cover the entire multidisciplinary scope of this important field. More than 200 expert contributors and reviewers from industry and academia across 21 countries provide easy-to-read entries that cover software requirements, design, construction, testing, maintenance, configuration management, quality control,

and software engineering management tools and methods. Editor Phillip A. Laplante uses the most universally recognized definition of the areas of relevance to software engineering, the Software Engineering Body of Knowledge (SWEBOK®), as a template for organizing the material. Also available in an electronic format, this encyclopedia supplies software engineering students, IT professionals, researchers, managers, and scholars with unrivaled coverage of the topics that encompass this ever-changing field. Also Available Online This Taylor & Francis encyclopedia is also available through online subscription, offering a variety of extra benefits for researchers, students, and librarians, including: Citation tracking and alerts Active reference linking Saved searches and marked lists HTML and PDF format options Contact Taylor and Francis for more information or to inquire about subscription options and print/online combination packages. US: (Tel) 1.888.318.2367; (E-mail) e-reference@taylorandfrancis.com International: (Tel) +44 (0) 20 7017 6062; (E-mail) online.sales@tandf.co.uk

The CERT C Coding Standard

98 Rules for Developing Safe, Reliable, and Secure Systems

Pearson Education The CERT C Coding Standard, Second Edition enumerates the coding errors that are the root causes of current software vulnerabilities in C, prioritizing them by severity, likelihood of exploitation, and remediation costs. "Secure programming in C can be more difficult than even many experienced programmers realize," said Robert C. Seacord, technical manager of the CERT Secure Coding Initiative and author of the CERT C Coding Standard. "Software systems are becoming increasing complex as our dependency on these systems increases. In our new CERT standard, as with all of our standards, we identify insecure coding practices and present secure alternatives that software developers can implement to reduce or eliminate vulnerabilities before deployment."

Software Product Lines in Action

The Best Industrial Practice in Product Line Engineering

Springer Science & Business Media Software product lines represent perhaps the most exciting paradigm shift in software development since the advent of high-level programming languages. Nowhere else in software engineering have we seen such breathtaking improvements in cost, quality, time to market, and developer productivity, often registering in the order-of-magnitude range. Here, the authors combine academic research results with real-world industrial experiences, thus presenting a broad view on product line engineering so that both managers and technical specialists will benefit from exposure to this work. They capture the wealth of knowledge that eight companies have gathered during the introduction of the software product line engineering approach in their daily practice.

Performance Improvement Quarterly

Corporate Defense and the Value Preservation Imperative

Bulletproof Your Corporate Defense Program

CRC Press This is the first book to finally address the umbrella term corporate defense, and to explain how an integrated corporate defense program can help an organization address both value creation and preservation. The book explores the value preservation imperative, which represents an organization's obligation to implement a comprehensive corporate defense program in order to deliver long-term sustainable value to its stakeholders. For the first time the reader is provided with a complete picture of how corporate defense operates all the way from the boardroom to the front-lines, and vice versa. It provides comprehensive guidance on how to implement a robust corporate defense program

by addressing this challenge from strategic, tactical, and operational perspectives. This arrangement provides readers with a holistic view of corporate defense and incorporates the management of the eight critical corporate defense components. It includes how an organization needs to integrate its governance, risk, compliance, intelligence, security, resilience, controls and assurance activities within its corporate defense program. The book addresses the corporate defense requirement from various perspectives and helps readers to understand the critical interconnections and inter-dependencies which exist at strategic, tactical, and operational levels. It facilitates the reader in comprehending the importance of appropriately prioritizing corporate defense at a strategic level, while also educating the reader in the importance of managing corporate defense at a tactical level, and executing corporate defense activities at an operational level. Finally the book looks at the business case for implementing a robust corporate defense program and the value proposition of introducing a truly world class approach to addressing the value preservation imperative. Cut and paste this link (https://m.youtube.com/watch?v=u5R_eOPNHbI) to learn more about a corporate defense program and how the book will help you implement one in your organization.

Software Architecture

First European Conference, ECSA 2007, Madrid, Spain, September 24-26, 2007, Proceedings

Springer Science & Business Media This book constitutes the refereed proceedings of the First European Conference on Software Architecture, ECSA 2007, held in Aranjuez, Spain. The 12 revised long papers presented together with four short papers cover description languages and metamodels, architecture-based code generation, run-time monitoring, requirements engineering, service-oriented architectures, aspect-oriented software architectures, ontology-based approaches, autonomic systems, middleware and web services.

Reflections on Management

How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself

Pearson Education A Lifetime of Invaluable Management Insights from Legendary Software Quality Guru Watts S. Humphrey In 1986, Watts S. Humphrey made an outrageous commitment: a promise to transform software development. As the pioneering innovator behind SEI's Capability Maturity Model (CMM), Personal Software Process (PSP), and Team Software Process (TSP), Humphrey has more than met that promise. But his contributions go beyond methodology: For decades, his deeply personal writings on project management have been admired by software engineers worldwide. Reflections on Management brings together Humphrey's best and most influential essays and articles--sharing insights that will be indispensable for anyone who must achieve superior results in software or any other endeavor. Collected here for the first time, these works offer compelling insights into everything from planning day-to-day work to improving quality, encouraging teamwork to becoming a truly great leader. All of these writings share a powerful vision, grounded by a life in software that has extended across nearly six decades. The vision is this: To succeed, professionals must effectively manage for more than plans, schedules, and code--they must manage teams, bosses, and above all, themselves.

CMMI for Development

Guidelines for Process Integration and Product Improvement

Pearson Education CMMI® for Development (CMMI-DEV) describes best practices for the development and maintenance of products and services across their lifecycle. By integrating essential bodies of knowledge, CMMI-DEV provides a single, comprehensive framework for organizations to assess their development and maintenance processes and improve performance. Already widely adopted throughout the world for disciplined, high-quality engineering, CMMI-DEV Version 1.3 now accommodates other modern approaches as well,

including the use of Agile methods, Lean Six Sigma, and architecture-centric development. CMMI® for Development, Third Edition, is the definitive reference for CMMI-DEV Version 1.3. The authors have revised their tips, hints, and cross-references, which appear in the margins of the book, to help you better understand, apply, and find information about the content of each process area. The book includes new and updated perspectives on CMMI-DEV in which people influential in the model's creation, development, and transition share brief but valuable insights. It also features four new case studies and five contributed essays with practical advice for adopting and using CMMI-DEV. This book is an essential resource–whether you are new to CMMI-DEV or are familiar with an earlier version–if you need to know about, evaluate, or put the latest version of the model into practice. The book is divided into three parts. Part One offers the broad view of CMMI-DEV, beginning with basic concepts of process improvement. It introduces the process areas, their components, and their relationships to each other. It describes effective paths to the adoption and use of CMMI-DEV for process improvement and benchmarking, all illuminated with fresh case studies and helpful essays. Part Two, the bulk of the book, details the generic goals and practices and the twenty-two process areas now comprising CMMI-DEV. The process areas are organized alphabetically by acronym for easy reference. Each process area includes goals, best practices, and examples. Part Three contains several useful resources, including CMMI-DEV-related references, acronym definitions, a glossary of terms, and an index.

TSP(SM) Coaching Development Teams

Pearson Education Most modern software development projects require teams, and good teamwork largely determines a project's success. The Team Software Process (TSP), created by Watts S. Humphrey, is a set of engineering practices and team concepts that produce effective teams, thereby helping developers deliver high-quality products on time and within budget. TSP bridges Humphrey's seminal work on the Capability Maturity Model (CMM), an improvement framework for the entire software organization, and his Personal Software Process (PSP), practices designed to improve the work of individual developers. Typical first-time TSP teams increase productivity by more than 50 percent while greatly increasing the quality of their delivered products. However, TSP teams only continue to improve under the guidance of a capable coach. One industrial-strength team, for example, increased its productivity by an additional 94 percent and reduced test defects by 85 percent through three consecutive TSP quarterly product release cycles. Without competent coaching, teams often do not progress much beyond the initial one-time improvement seen after the introduction of the TSP. Humphrey distinguishes between TSP coaching and TSP leadership, explaining why the skillful performance of both functions is critical. In this practical guide, he shares coaching methods that have repeatedly inspired TSP teams and steered them toward success. With the help of a coach, TSP teams undergo a brief but intense project launch in which they define their own processes, make their own plans, and negotiate their commitments with management, resulting in dramatically enhanced performance. Whether you are considering the TSP or are actively implementing it, TSPSM–Coaching Development Teams provides the invaluable examples, guidelines, and suggestions you need to get started and keep developing as a team coach. It's meant to complement Humphrey's other books, TSPSM–Leading a Development Team and PSPSM: A Self-Improvement Process for Software Engineers. Together, the three works offer a rich resource for improving your software development capabilities.