System Architecture Document

Integrative Document & Content ManagementSoftware ArchitectureEmbedded Systems ArchitectureTraceability Analysis of a High-Level Automotive System Architecture DocumentDocumenting Software ArchitecturesThe Oregon ExperimentGuide to Understanding Design Documentation in Trusted SystemsAdvanced Use Case ModelingModel-Based System ArchitectureHow to Become an It ArchitectProduct Focused Software Process ImprovementSoftware Architecture: System Design, Development and MaintenanceArc42 by ExampleCommon System and Software Testing PitfallsDocument Analysis Systems VIISoftware Systems ArchitectureProcess for System Architecture and Requirements EngineeringGuide to Efficient Software DesignDesigning Software ArchitecturesNew Trends in Databases and Information SystemsSoftware and Systems Architecture in ActionSoftware Architecture with PythonDesigning Software-Intensive Systems: Methods and PrinciplesSoftware Architecture in PracticeSoftware Systems ArchitectureSystem Architecture with XMLFrom Enterprise Architecture to IT GovernanceThe Architecture of Computer Hardware and System Software: An Information Technology Approach, 5th EditionStructured Software TestingDocument Analysis Systems IIBuilding Evolutionary ArchitecturesThe Method Framework for Engineering System ArchitecturesArmy-NASA Aircrew/Aircraft Integration Program (A3I) Software Detailed Design Document: Phase IIISystem Architecture and System DesignModeling and Simulation Support for System of Systems Engineering ApplicationsBuilding Natural Language Generation SystemsDatabase and Expert Systems ApplicationsSun Certified Enterprise Architect for J2EE Technology Study GuideDocument Architecture in Open Systems: The ODA StandardGuide to Enterprise IT Architecture

Integrative Document & Content Management

This book constitutes the refereed proceedings of the 7th International Conference on Document Analysis Systems, DAS 2006, held in Nelson, New Zealand, in February 2006. The 33 revised full papers and 22 poster papers presented were carefully reviewed and selected from 78 submissions. The papers are organized in topical sections on digital libraries, image processing, handwriting, document structure and format, tables, language and script identification, systems and performance evaluation, and retrieval and segmentation.

Software Architecture

An up-to-date and comprehensive overview of information and database systems design and implementation. The book provides an accessible presentation and explanation of technical architecture for systems complying with TOGAF standards, the accepted international framework. Covering nearly the full spectrum of architectural concern, the authors also illustrate and concretize the notion of traceability from business goals, strategy through to technical architecture, providing the reader with a holistic and commanding view. The work has two mutually supportive foci. First, information technology technical architecture, the in-depth, illustrative and contemporary treatment of which comprises the core and majority of the book; and secondly, a strategic and business context.

Embedded Systems Architecture

Presents modeling approaches that can be performed in SysML and other modeling languages This book combines the emerging discipline of systems architecting with model-based approaches using SysML. The early chapters of the book provide the fundamentals of systems architecting; discussing what systems architecting entails and how it benefits systems engineering. Model-based systems engineering is then defined, and its capabilities to develop complex systems on time and in a feasible quality are discussed. The remainder of the book covers important topics such as: architecture descriptions; architecture patterns; perspectives, viewpoints, views and their relation to system architecture; the roles of a system architect, their team, and stakeholders; systems architecting processes; agile approaches to systems architecting; variant modeling techniques; architecture frameworks; and architecture assessment. The book's organization allows experts to read the chapters out of sequence. Novices can read the chapters sequentially to gain a systematic introduction to system architecting. Model-Based System Architecture: Provides comprehensive coverage of the Functional Architecture for Systems (FAS) method created by the authors and based on common MBSE practices Covers architecture frameworks, including the System of Systems, Zachman Frameworks, TOGAF®, and more Includes a consistent example system, the “Virtual Museum Tour” system, that allows the authors to demonstrate the systems architecting concepts covered in the book Model-Based System Architecture is a comprehensive reference for system architects and systems engineers in technology companies. This book will also serve as a reference to students and researchers interested in functional architectures. Tim Weilkiens is the CEO at the German consultancy oose Innovative Informatik and co-author of the SysML specification. He has introduced model-based systems engineering to a variety of industry sectors. He is author of several books about modeling and the MBSE methodology SYSMOD. Jesko G. Lamm is a Senior Systems Engineer at Bernafon, a Swiss manufacturer for hearing instruments. With Tim Weilkiens, Jesko G. Lamm founded the Functional Architectures working group of the German chapter of INCOSE. Stephan Roth is a coach, consultant, and trainer for systems and software engineering at the German consultancy oose Innovative Informatik. He is a state-certified technical assistant for computer science from Physikalisch-Technische Lehranstalt (PTL) Wedel and a certified systems engineer (GfSE)®- Level C. Markus Walker works at Schindler Elevator in the research and development division as elevator system architect. He is an INCOSE Certified Systems Engineering Professional (CSEP) and is engaged in the committee of the Swiss chapter of INCOSE.

Traceability Analysis of a High-Level Automotive System Architecture Document

“Don’s book is a very good addition both to the testing literature and to the literature on quality assurance and software engineering… . [It] is likely to become a standard for test training as well as a good reference for professional testers and developers. I would also recommend this book as background material for negotiating outsourced software contracts. I often work as an expert witness in litigation for software with very poor quality, and this book might well reduce or eliminate these lawsuits....” –Capers Jones, VP and CTO, Namcook Analytics LLC Software and system testers repeatedly fall victim to the same pitfalls. Think of them as “anti-patterns”: mistakes that make testing far less effective and efficient than it ought to be. In Common System and Software Testing Pitfalls, Donald G. Firesmith catalogs 92 of these pitfalls. Drawing on his 35 years of software and system engineering experience, Firesmith shows testers and technical managers and other stakeholders how to avoid falling into these pitfalls, recognize when they have already fallen in, and escape while minimizing their negative consequences. Firesmith writes for testing professionals and other stakeholders involved in large or medium-sized projects. His anti-patterns and solutions address both “pure software” applications and “software-reliant systems,” encompassing heterogeneous subsystems, hardware, software, data, facilities, material, and personnel. For each pitfall, he identifies its applicability, characteristic symptoms, potential negative consequences and causes, and offers specific actionable recommendations for avoiding it or limiting its consequences. This guide will help you Pinpoint testing processes that need improvement–before, during, and after the project Improve shared understanding and collaboration among all project participants Develop, review, and optimize future project testing programs Make your test documentation far more useful Identify testing risks and appropriate risk-mitigation strategies Categorize testing problems for metrics collection, analysis, and reporting Train new testers, QA specialists, and other project stakeholders With 92 common testing pitfalls organized into 14 categories, this taxonomy of testing pitfalls should be relatively complete. However, in spite of its comprehensiveness, it is also quite likely that additional pitfalls and even missing categories of pitfalls will be identified over time as testers read this book and compare it to their personal experiences. As an enhancement to the print edition, the author has provided the following location on the web where readers can find major additions and modifications to this taxonomy of pitfalls: http://donald.firesmith.net/home/common-testing-pitfalls Please send any recommended changes and additions to dgf (at) sei (dot) cmu (dot) edu, and the author will consider them for publication both on the website and in future editions of this book.

Documenting Software Architectures

"This book addresses the complex issues associated with software engineering environment capabilities for designing real-time embedded software systems"--Provided by publisher.

The Oregon Experiment

Scenario -- Groundwork -- Structure -- Meaning -- Modeling processes -- Communication -- Navigation and discovery -- Presentation formats -- Infrastructure -- Solutions.

Guide to Understanding Design Documentation in Trusted Systems

This book provides an overview of the state of the art in research and development of systems for document image analysis. Topics covered include a variety of systems and architectures for processing document images as well as methods for converting those images into formats that can be manipulated by a computer. The chapters are written by recognized experts in the field and describe Systems and Architectures, Recognition Techniques, Graphics Analysis, Document Image Retrieval, and World Wide Web Applications.

Advanced Use Case Modeling

A set of good practices related to design documentation in automated data processing systems employed for processing classified and other sensitive information. Helps vendor and evaluator community understand what deliverables are required for design documentation and the level of detail required of design documentation at all classes in the Trusted Computer Systems Evaluation Criteria.

Model-Based System Architecture

The software development ecosystem is constantly changing, providing a constant stream of new tools, frameworks, techniques, and paradigms. Over the past few years, incremental developments in core engineering practices for software development have created the foundations for rethinking how architecture changes over time, along with ways to protect important architectural characteristics as it evolves. This practical guide ties those parts together with a new way to think about architecture and time.

How to Become an It Architect

Portals present unique strategic challenges in the academic environment. Their conceptualization and design requires the input of campus constituents who seldom interact and whose interests are often opposite. The implementation of a portal requires a coordination of applications and databases controlled by different campus units at a level that may never before have been attempted at the institution. Building a portal is as much about constructing intra-campus bridges as it is about user interfaces and content. Designing Portals: Opportunities and Challenges discusses the current status of portals in higher education by providing insight into the role portals play in an institution's business and educational strategy, by taking the reader through the processes of conceptualization, design, and implementation of the portals (in different stages of development) at major universities and by offering insight from three producers of portal software systems in use at institutions of higher learning and elsewhere.

Product Focused Software Process Improvement

This book shows its readers how to achieve the goal of genuine IT governance. The key here is the successful development of enterprise architecture as the necessary foundation. With its capacity to span and integrate business procedures, IT applications and IT infrastructure, enterprise architecture opens these areas up to analysis and makes them rich sources of critical data. Enterprise architecture thereby rises to the status of a crucial management information system for the CIO. The focused analysis of the architecture (its current and future states) illuminates the path to concrete IT development planning and the cost-effective and beneficial deployment of IT. Profit from the author's firsthand experience - proven approaches firmly based in enterprise reality.

Software Architecture: System Design, Development and Maintenance

Arc42 by Example

"This book isn't just another introduction to use cases. The authors have used their wealth of experience to produce an excellent and insightful collection of detailed examples, explanations, and advice on how to work with use cases." –Maria Ericsson The toughest challenge in building a software system that meets the needs of your audience lies in clearly understanding the problems that the system must solve. Advanced Use Case Modeling presents a framework for discovering, identifying, and modeling the problem that the software system will ultimately solve. Software developers often employ use cases to specify what should be performed by the system they're constructing. Although use case-driven analysis, design, and testing of software systems has become increasingly popular, little has been written on the role of use cases in the complete software cycle. This book fills that need by describing how to create use case models for complex software development projects, using practical examples to explain conceptual information. The authors extend the work of software visionary Ivar Jacobson, using the Unified Modeling Language (UML) as the notation to describe the book's models. Aimed primarily at software professionals, Advanced Use Case Modeling also includes information that relates use case technique to business processes. This book presents a process for creating and maintaining use case models in a framework that can be fully customized for your organization. The authors, pioneers in the application of use cases in software development, bring their extensive experience to cover topics such as: A process model for applying a use case model How to keep your use case modeling effort on track Tips and pitfalls in use case modeling How to organize your use case model for large-system development Similarities between Advanced Use Case Modeling and the Rational Unified Process framework Effect of use cases on user interface design Guidelines for quality use case modeling

Common System and Software Testing Pitfalls

Database and information systems technologies have been rapidly evolving in several directions over the past years. New types and kinds of data, new types of applications and information systems to support them raise diverse challenges to be addressed. The so-called big data challenge, streaming data management and processing, social networks and other complex data analysis, including semantic reasoning into information systems supporting for instance trading, negotiations, and bidding mechanisms are just some of the emerging research topics. This volume contains papers contributed by six workshops: ADBIS Workshop on GPUs in

Databases (GID 2012), Mining Complex and Stream Data (MCSD'12), International Workshop on Ontologies meet Advanced Information Systems (OAIS'2012), Second Workshop on Modeling Multi-commodity Trade: Data models and processing (MMT'12), 1st ADBIS Workshop on Social Data Processing (SDP'12), 1st ADBIS Workshop on Social and Algorithmic Issues in Business Support (SAIBS), and the Ph.D. Consortium associated with the ADBIS 2012 conference that report on the recent developments and an ongoing research in the aforementioned areas.

Document Analysis Systems VII

Introduction. Architectural styles. Case studies. Shared information systems. Architectural design guidance. Formal models and specifications. Linguistics issues. Tools for architectural design. Education of software architects.

Software Systems Architecture

The Architecture of Computer Hardware and System Software provides the right amount of technical detail needed to succeed in the field. This accessible introduction provides the basic principles of computer system architecture and organization in the context of the current technological landscape. The author provides chapters on the fundamentals of networking as it relates to computer systems as well as all kinds of business systems, from entrepreneurial to small business, networked, distributed, and more. This valuable book provides IT professionals with several real-world case studies that clearly show how the concepts are applied in the field.

Process for System Architecture and Requirements Engineering

Defining the various types of IT architecture in the industry, this one-of-a-kind resource highlights the rewards of becoming an architect and explores the details of the deliverables, project structure, and how to approach their creation. --

Guide to Efficient Software Design

Architect and design highly scalable, robust, clean, and highly performant applications in Python About This Book Identify design issues and make the necessary adjustments to achieve improved performance Understand practical architectural quality attributes from the perspective of a practicing engineer and architect using Python Gain knowledge of architectural principles and how they can be used to provide accountability and rationale for architectural decisions Who This Book Is For This book is for experienced Python developers who are aspiring to become the architects of enterprise-grade applications or software architects who would like to leverage Python to create effective blueprints of applications. What You Will Learn Build programs with the right architectural attributes Use Enterprise Architectural Patterns to solve scalable problems on the Web Understand design patterns from a Python perspective Optimize the performance testing tools in Python Deploy code in remote environments or on the Cloud using Python Secure architecture applications in Python In Detail This book starts off by explaining how

Python fits into an application architecture. As you move along, you will understand the architecturally significant demands and how to determine them. Later, you'll get a complete understanding of the different architectural quality requirements that help an architect to build a product that satisfies business needs, such as maintainability/reusability, testability, scalability, performance, usability, and security. You will use various techniques such as incorporating DevOps, Continuous Integration, and more to make your application robust. You will understand when and when not to use object orientation in your applications. You will be able to think of the future and design applications that can scale proportionally to the growing business. The focus is on building the business logic based on the business process documentation and which frameworks are to be used when. We also cover some important patterns that are to be taken into account while solving design problems as well as those in relatively new domains such as the Cloud. This book will help you understand the ins and outs of Python so that you can make those critical design decisions that not just live up to but also surpass the expectations of your clients. Style and approach Filled with examples and use cases, this guide takes a no-nonsense approach to help you with everything it takes to become a successful software architect.

Designing Software Architectures

For more and more systems, software has moved from a peripheral to a central role, replacing mechanical parts and hardware and giving the product a competitive edge. Consequences of this trend are an increase in: the size of software systems, the variability in software artifacts, and the importance of software in achieving the system-level properties. Software architecture provides the necessary abstractions for managing the resulting complexity. We here introduce the Third Working IEEFIIFIP Conference on Software Architecture, WICSA3. That it is already the third such conference is in itself a clear indication that software architecture continues to be an important topic in industrial software development and in software engineering research. However, becoming an established field does not mean that software architecture provides less opportunity for innovation and new directions. On the contrary, one can identify a number of interesting trends within software architecture research. The first trend is that the role of the software architecture in all phases of software development is more explicitly recognized. Whereas initially software architecture was primarily associated with the architecture design phase, we now see that the software architecture is treated explicitly during development, product derivation in software product lines, at run-time, and during system evolution. Software architecture as an artifact has been decoupled from a particular lifecycle phase.

New Trends in Databases and Information Systems

In 1989, the ISO Standard 8613 "Office Document Architecture (ODA) and Interchange Format" was published. The Standard is intended for the interchange of documents in an Open Systems environment. ISO 8613 is technically identical to the CCITT Recommendations of the T.410 series called "Open Document Architecture (ODA) and Interchange Format" published in 1988. Almost all major companies in the office automation and telecommunication area are currently developing products based on this Standard. In this book, all important aspects of the presently publishedeight parts of the Standard are discussed. The book provides a comprehensiveand detailed introduction to the technical specifications of ISO 8613 and the concepts on which these specifications are based, including the extensions which were added to the ODA Standard in 1991. The book isprimarily addressed to readers who want to investigate the applicability of the Standard for their document interchange problems, plan to acquire products based on the Standard, or intend to develop document processingsystems conforming to the Standard.

Software and Systems Architecture in Action

Software Architecture with Python

Embedded Systems Architecture is a practical and technical guide to understanding the components that make up an embedded system's architecture. This book is perfect for those starting out as technical professionals such as engineers, programmers and designers of embedded systems; and also for students of computer science, computer engineering and electrical engineering. It gives a much-needed 'big picture' for recently graduated engineers grappling with understanding the design of real-world systems for the first time, and provides professionals with a systems-level picture of the key elements that can go into an embedded design, providing a firm foundation on which to build their skills. Real-world approach to the fundamentals, as well as the design and architecture process, makes this book a popular reference for the daunted or the inexperienced: if in doubt, the answer is in here! Fully updated with new coverage of FPGAs, testing, middleware and the latest programming techniques in C, plus complete source code and sample code, reference designs and tools online make this the complete package Visit the companion web site at http://booksite.elsevier.com/9780123821966/ for source code, design examples, data sheets and more A true introductory book, provides a comprehensive get up and running reference for those new to the field, and updating skills: assumes no prior knowledge beyond undergrad level electrical engineering Addresses the needs of practicing engineers, enabling it to get to the point more directly, and cover more ground. Covers hardware, software and middleware in a single volume Includes a library of design examples and design tools, plus a complete set of source code and embedded systems design tutorial materials from companion website

Designing Software-Intensive Systems: Methods and Principles

This is the digital version of the printed book (Copyright © 2000). Derek Hatley and Imtiaz Pirbhai—authors of Strategies for Real-Time System Specification—join with influential consultant Peter Hruschka to present a much anticipated update to their widely implemented Hatley/Pirbhai methods. Process for System Architecture and Requirements Engineering introduces a new approach that is particularly useful for multidisciplinary system development: It applies equally well to all technologies and thereby provides a common language for developers in widely differing disciplines. The Hatley-Pirbhai-Hruschka approach (H/H/P) has another important feature: the coexistence of the requirements and architecture methods and of the corresponding models they produce. These two models are kept separate, but the approach fully records their ongoing and changing interrelationships. This feature is missing from virtually all other system and software development methods and from CASE tools that only automate the requirements model. System managers, system architects, system engineers, and managers and engineers in all of the diverse engineering technologies will benefit from this comprehensive, pragmatic text. In addition to its models of requirements and architecture and of the development process itself, the book uses in-depth case studies of a hospital monitoring system and of a multidisciplinary groundwater analysis system to illustrate the principles. Compatibility Between the H/H/P Methods and the UML: The Hatley/Pirbhai architecture and requirements methods—described in Strategies for Real-Time System Specification—have been widely used for almost two decades in system and software development. Now known as the Hatley/Hruschka/Pirbhai (H/H/P) methods, they have always been compatible with object-oriented software techniques, such as the UML, by defining architectural elements as classes, objects, messages, inheritance relationships, and so on. In Process for System Architecture and Requirements Engineering, that compatibility is made more specific through the addition of message diagrams, inheritance diagrams, and new notations that go with them. In addition, state charts, while never excluded, are now specifically included as a representation of sequential machines. These additions make definition of the system/software boundary even more straightforward, while retaining the clear separation of requirements and design at the system levels that is a hallmark of the H/H/P methods—not shared by most OO techniques. Once the transition to software is made, the developer is free to continue using the H/H/P methods, or to use the UML or any other software-specific technique.

Software Architecture in Practice

Designing Software Architectures will teach you how to design any software architecture in a systematic, predictable, repeatable, and cost-effective way. This book introduces a practical methodology for architecture design that any professional software engineer can use, provides structured methods supported by reusable chunks of design knowledge, and includes rich case studies that demonstrate how to use the methods. Using realistic examples, you’ll master the powerful new version of the proven Attribute-Driven Design (ADD) 3.0 method and will learn how to use it to address key drivers, including quality attributes, such as modifiability, usability, and availability, along with functional requirements and architectural concerns. Drawing on their extensive experience, Humberto Cervantes and Rick Kazman guide you through crafting practical designs that support the full software life cycle, from requirements to maintenance and evolution. You’ll learn how to successfully integrate design in your organizational context, and how to design systems that will be built with agile methods. Comprehensive coverage includes Understanding what architecture design involves, and where it fits in the full software development life cycle Mastering core design concepts, principles, and processes Understanding how to perform the steps of the ADD method Scaling design and analysis up or down, including design for pre-sale processes or lightweight architecture reviews Recognizing and optimizing critical relationships between analysis and design Utilizing proven, reusable design primitives and adapting them to specific problems and contexts Solving design problems in new domains, such as cloud, mobile, or big data

Software Systems Architecture

Details the master architectural design plan currently being implemented at the University of Oregon, illustrating the participation of all members of a small community in the designing of their own environment

System Architecture with XML

Software Systems Architecture is a practitioner-oriented guide to designing and implementing effective architectures for information systems. It is both a readily accessible introduction to software architecture and an invaluable handbook of well-established best practices. It shows why the role of the architect is central to any successful information-systems development project, and, by presenting a set of architectural viewpoints and perspectives, provides specific direction for improving your own and your organization's approach to software systems architecture. With this book you will learn how to Design an architecture that reflects and balances the different needs of its stakeholders Communicate the architecture to stakeholders and demonstrate that it has met their requirements Focus on architecturally significant aspects of design, including frequently overlooked areas such as performance, resilience, and location Use scenarios and patterns to drive the creation and validation of your architecture Document your architecture as a set of related views Use perspectives to ensure that your architecture exhibits important qualities such as performance, scalability, and security The architectural viewpoints and perspectives presented in the book also provide a valuable long-term reference source for new and experienced architects alike. Whether you are an aspiring or practicing software architect, you will find yourself referring repeatedly to the practical advice in this book throughout the lifecycle of your projects. A supporting Web site containing further information can be found at www.viewpoints-and-perspectives.info

From Enterprise Architecture to IT Governance

The Architecture of Computer Hardware and System Software: An Information Technology Approach, 5th Edition

This is the eagerly-anticipated revision to one of the seminal books in the field of software architecture which clearly defines and explains the topic.

Structured Software Testing

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

Document Analysis Systems II

Modern-day projects require software and systems engineers to work together in realizing architectures of large and complex software-intensive systems. To date, the two have used their own tools and methods to deal with similar issues when it comes to the requirements, design, testing, maintenance, and evolution of these architectures. Software and Systems Architecture in Action explores practices that can be helpful in the development of architectures of large-scale systems in which software is a major component. Examining the synergies that exist between the disciplines of software and systems engineering, it presents concepts, techniques, and methods for creating and documenting architectures. The book describes an approach to architecture design that is driven from systemic quality attributes determined from both the business and technical goals of the system, rather than just its functional requirements. This architecture-centric design approach utilizes analytically derived patterns and tactics for quality attributes that inform the architect’s design choices and help shape the architecture of a given system. The book includes coverage of techniques used to assess the impact of architecture-centric design on the structural complexity of a system. After reading the book, you will understand how to create architectures of systems and assess their ability to meet the business goals of your organization. Ideal for anyone involved with large and complex software-intensive systems, the book details powerful methods for engaging the software and systems engineers on your team. The book is also suitable for use in undergraduate and graduate-level courses on software and systems architecture as it exposes students to the concepts and techniques used to create and manage architectures of software-intensive systems.

Building Evolutionary Architectures

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

The Method Framework for Engineering System Architectures

This book constitutes the refereed proceedings of the Second International Conference on Product Focused Software Process Improvement, PROFES 2000, held in Oulu, Finland, in June 2000. The 30 revised full papers presented were carefully reviewed and selected from a total of 60 submitted full papers. The book is divided into topical sections on process improvement, empirical software engineering, industrial experiences, methods and tools, software process and modeling, software and process measurement, and organizational learning and experience factory.

Army-NASA Aircrew/Aircraft Integration Program (A3I) Software Detailed Design Document: Phase III

Explains how to build computer software systems which generate understandable texts in human languages.

System Architecture and System Design

“a much-needed handbook with contributions from well-chosen practitioners. A primary accomplishment is to provide guidance for those involved in modeling and simulation in support of Systems of Systems development, more particularly guidance that draws on well-conceived academic research to define concepts and terms, that identifies primary challenges for developers, and that suggests fruitful approaches grounded in theory and successful examples.” Paul Davis, The RAND Corporation Modeling and Simulation Support for System of Systems Engineering Applications provides a comprehensive overview of the underlying theory, methods, and solutions in modeling and simulation support for system of systems engineering. Highlighting plentiful multidisciplinary applications of modeling and simulation, the book uniquely addresses the criteria and challenges found within the field. Beginning with a foundation of concepts, terms, and categories, a theoretical and generalized approach to system of systems engineering is introduced, and real-world applications via case studies and examples are presented. A unified approach is maintained in an effort to understand the complexity of a single system as well as the context among other proximate systems. In addition, the book features: Cutting edge coverage of modeling and simulation within the field of system of systems, including transportation, system health management, space mission analysis, systems engineering methodology, and energy State-of-the-art advances within multiple domains to instantiate theoretic insights, applicable methods, and lessons learned from real-world applications of modeling and simulation The challenges of system of systems engineering using a systematic and holistic approach Key concepts, terms, and activities to provide a comprehensive, unified, and concise representation of the field A collection of chapters written by over 40 recognized international experts from academia, government, and industry A research agenda derived from the contribution of experts that guides scholars and researchers towards open questions Modeling and Simulation Support for System of Systems Engineering Applications is an ideal reference and resource for academics and practitioners in operations research, engineering, statistics, mathematics, modeling and simulation, and computer science. The book is also an excellent course book for graduate and PhD-level courses in modeling and simulation, engineering, and computer science.

Modeling and Simulation Support for System of Systems Engineering Applications

This classroom-tested textbook presents an active-learning approach to the foundational concepts of software design. These concepts are then applied to a case study, and reinforced through practice exercises, with the option to follow either a structured design or object-oriented design paradigm. The text applies an incremental and iterative software development approach, emphasizing the use of design characteristics and modeling techniques as a way to represent higher levels of design abstraction, and promoting the model-view-controller (MVC) architecture. Topics and features: provides a case study to illustrate the various concepts discussed throughout the book, offering an in-depth look at the pros and cons of different software designs; includes discussion questions and hands-on exercises that extend the case study and apply the concepts to other problem domains; presents a review of program design fundamentals to reinforce understanding of the basic concepts; focuses on a bottom-up approach to describing software design concepts; introduces the characteristics of a good software design, emphasizing the model-view-controller as an underlying architectural principle; describes software design from both object-oriented and structured perspectives; examines additional topics on human-computer interaction design, quality assurance, secure design, design patterns, and persistent data storage design; discusses design concepts that may be applied to many types of software development projects; suggests a template for a software design document, and offers ideas for further learning. Students of computer science and software engineering will find this textbook to be indispensable for advanced undergraduate courses on programming and software design. Prior background knowledge and experience of programming is required, but familiarity in software design is not assumed.

Building Natural Language Generation Systems

Document the architecture of your software easily with this highly practical, open-source template. Key Features Get to grips with leveraging the features of arc42 to create insightful documents Learn the concepts of software architecture documentation through real-world examples Discover techniques to create compact, helpful, and easy-to-read documentation Book Description When developers document the architecture of their systems, they often invent their own specific ways of articulating structures, designs, concepts, and decisions. What they need is a template that enables simple and efficient software architecture documentation. arc42 by Example shows how it's done through several real-world examples. Each example in the book, whether it is a chess engine, a huge CRM system, or a cool web system, starts with a brief description of the problem domain and the quality requirements. Then, you'll discover the system context with all the external interfaces. You'll dive into an overview of the solution strategy to implement the building blocks and runtime scenarios. The later chapters also explain various cross-cutting concerns and how they affect other aspects of a program. What you will learn Utilize arc42 to document a system's physical infrastructure Learn how to identify a system's scope and boundaries Break a system down into building blocks and illustrate the relationships between them Discover how to describe the runtime behavior of a system Know how to document design decisions and their reasons Explore the risks and technical debt of your system Who this book is for This book is for software developers and solutions architects who are looking for an easy, open-source tool to document their systems. It is a useful reference for those who are already using arc42. If you are new to arc42, this book is a great learning resource. For those of you who want to write better technical documentation will benefit from the general concepts covered in this book.

Database and Expert Systems Applications

Software Systems Architecture, Second Edition is a highly regarded, practitioner-oriented guide to designing and implementing effective architectures for information systems. It is both a readily accessible introduction to software architecture and an invaluable handbook of well-established best practices. With this book you will learn how to Design and communicate an architecture that reflects and balances the different needs of its stakeholders Focus on architecturally significant aspects of design, including frequently overlooked areas such as performance, resilience, and location Use scenarios and patterns to drive the creation and validation of your architecture Document your architecture as a set of related views Reflecting new standards and developments in the field, this new edition extends and updates much of the content, and Adds a “system context viewpoint” that documents the system's interactions with its environment Expands the discussion of architectural principles, showing how they can be used to provide traceability and rationale for architectural decisions Explains how agile development and architecture can work together Positions requirements and architecture activities in the project context Presents a new lightweight method for architectural validation Whether you are an aspiring or practicing software architect, you will find yourself referring repeatedly to the practical advice in this book throughout the lifecycle of your projects. A supporting Web site containing further information can be found at www.viewpoints-and-perspectives.info.

Sun Certified Enterprise Architect for J2EE Technology Study Guide

Helps readers understand the goals of system architecture, design patterns, identify the appropriate J2EE technologies and APIs, maximize security and scalability, and evaluate existing architectures.

Document Architecture in Open Systems: The ODA Standard

Structured Software Testing- The Discipline of Discovering Software Errors is a book that will be liked both by readers from academia and industry. This book is unique and is packed with software testing concepts, techniques, and methodologies, followed with a step-by-step approach to illustrate real-world applications of the same. Well chosen topics, apt presentation, illustrative approach, use of valuable schematic diagrams and tables, narration of best practices of industry are the highlights of this book and make it a must read book. Key Features of the Book: Well chosen and sequenced chapters which make it a unique resource for test practitioners, also, as a text at both graduate and post-graduate levels. Apt presentation of Testing Techniques covering Requirement Based: Basic & Advanced, Code Based: Dynamic & Static, Data Testing, User Interface, Usability, Internationalization & Localization Testing, and various aspects of bugs which are narrated with carefully chosen examples. Illustrative approach to demonstrate software testing concepts, methodologies, test case designing and steps to be followed, usefulness, and issues. Valuable schematic diagrams and tables to enhance ability to comprehend the topics explained Best practices of industry and checklists are nicely fitted across different sections of the book.

Guide to Enterprise IT Architecture

This volume constitutes the proceedings of the 5th International Conference on Database and Expert Systems Applications (DEXA '94), held in Athens, Greece in September 1994. The 78 papers presented were selected from more than 300 submissions and give a comprehensive view of advanced applications of databases and expert systems. Among the topics covered are object-oriented, temporal, active, geographical, hypermedia and distributed databases, data management, cooperative office applications, object-oriented modelling, industrial applications, conceptual modelling, legal systems, evolving environments, knowledge engineering, information retrieval, advanced querying, medical systems, and CIM.