

Online Documentation Software

"This book explores some of the most recent developments in robotic motion, artificial intelligence, and human-machine interaction, providing insight into a wide variety of applications and functional areas"--Provided by publisher.

The Web is a global information space consisting of linked documents and linked data. As the Web continues to grow and new technologies, modes of interaction, and applications are being developed, the task of the Semantic Web is to unlock the power of information available on the Web into a common semantic information space and to make it available for sharing and processing by automated tools as well as by people. Right now, the publication of large datasets on the Web, the opening of data access interfaces, and the encoding of the semantics of the data extend the current human-centric Web. Now, the Semantic Web community is tackling the challenges of how to create and manage Semantic Web content, how to make Semantic Web applications robust and scalable, and how to organize and integrate information from different sources for novel uses. To foster the exchange of ideas and collaboration, the International Semantic Web Conference brings together researchers and practitioners in relevant disciplines such as artificial intelligence, databases, social networks, distributed computing, Web engineering, information systems, natural language processing, soft computing, and human-computer interaction. This volume contains the main proceedings of ISWC 2008, which we are cited to offer to the growing community of researchers and practitioners of the Semantic Web. We got a tremendous response to our call for research papers from a truly international community of researchers and practitioners from 41 countries submitting 261 papers. Each paper received an average of 3.

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

This book is designed to address the randomness of the literature on software documentation. As anyone interested in software documentation is aware, the field is highly synthetic; information about software documentation may be found in engineering, computer science training, technical communication, management, education and so on. "Perspectives on Software Documentation" contains a variety of perspectives, all tied together by the shared need to make software products more usable.

The tools you need to follow your dream of starting and running an online business! With the right knowledge and resources, you can take action to start the online business you've been dreaming of. This comprehensive guide provides tips and tricks for turning your dream into a reality. The sixth edition of Starting an Online Business: All-in-One For Dummies will teach you the basics and beyond. It will prepare you to set up your business website, offer your products in an online store, and keep accurate books. The authors help you navigate the primary legal, accounting, and security challenges related to running an online business. Fund your business for success and future growth Use SEO strategically to drive traffic to a well-designed site Market your business effectively as an entrepreneur Stand out, build customer relationships, and sell on social media Keep up with ecommerce trends to stay a step ahead With some guidance, you can find your market niche, create a business plan, and decide on a revenue model. Then, it's time to set up shop! Starting an Online Business can help bring your dream of an online business to life and guide you on the road to success.

PCMag.com is a leading authority on technology, delivering Labs-based, independent reviews of the latest products and services. Our expert industry analysis and practical solutions help you make better buying decisions and get more from technology.

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

Presents an illustrated A-Z encyclopedia containing approximately 600 entries on computer and technology related topics.

Designed for beginners and intermediate project team, this book serves as a detailed reference guide to the preparation of effective documentation for computer applications. It is intended for those who wish to develop software documentation and requires no prior knowledge or experience of writing software documentation. This book equips the project team with software documentation writing skills leaving behind a blue print of how each kind of software documentation is written in the real world. It showcases real world samples of the most required project documentation. This is something the project team is really going to appreciate. They can quickly get started by simply looking at the samples. Key Topics Audience Analysis SDLC/DDLC Case Study SRS User Manual HLDD LLDD Data Dictionary Online Help Installation Manual Editing Proofreading Formatting Guidelines What You'll Learn? How to: Prepare for the Technical Writing job Create a resume for the Technical Writing job Understand: The software documentation process The skills set required for software documentation Make a note of the various Publishing, Help Authoring, Graphic and Screen Capturing tools Learn how to choose the most appropriate software documentation tool Learn how to analyze the audience Gain insight into: Software Development Life Cycle [SDLC] Document Development Life Cycle [DDLC] Learn how SDLC relates to DDLC About The Authors The author Sharanam Shah [www.sharanamshah.com] has 9+ years of IT experience and is currently a technical writer for Saba Software Inc. He also consults with several software houses in Mumbai, India, to help them design and manage database applications. Aarti Shah, a technical writer, has a rich experience of churning out huge technical documents. She works as a freelancer for a lot of software houses to help them document their applications.

This report explains the technical and scientific communication internship I performed after completing my MTSC classes at Miami University. In the report, I orient the reader to the context in which I performed the internship by introducing Ontario Systems, its products, organization, corporate culture, and the work I performed there. I proceed to describe the activities and projects in which I spent time during the internship. I then extensively detail my

largest project, the letter management online help, outlining the process I used to complete the project. In the final chapter, I analyze the process I used on the letter management project by using the Anderson Problem-Solving Method for Technical and Scientific Communication. I show how the process I used at Ontario Systems represents an example of the problem-solving method.

Designed for translators and other professional linguists, this work attempts to clarify, explain and exemplify the impact that computers have had and are having on their profession. The book concerns machine translation, computer-aided translation and the future of translation and the computer.

Software documentation forms the basis for all communication relating to a software project. To be truly effective and usable, it should be based on what needs to be known. Agile Documentation provides sound advice on how to produce lean and lightweight software documentation. It will be welcomed by all project team members who want to cut out the fat from this time consuming task. Guidance given in pattern form, easily digested and cross-referenced, provides solutions to common problems. Straightforward advice will help you to judge: What details should be left in and what left out When communication face-to-face would be better than paper or online How to adapt the documentation process to the requirements of individual projects and build in change How to organise documents and make them easily accessible When to use diagrams rather than text How to choose the right tools and techniques How documentation impacts the customer Better than offering pat answers or prescriptions, this book will help you to understand the elements and processes that can be found repeatedly in good project documentation and which can be shaped and designed to address your individual circumstance. The author uses real-world examples and utilises agile principles to provide an accessible, practical pattern-based guide which shows how to produce necessary and high quality documentation.

Use an Approach Inspired by Domain-Driven Design to Build Documentation That Evolves to Maximize Value Throughout Your Development Lifecycle Software documentation can come to life, stay dynamic, and actually help you build better software. Writing for developers, coding architects, and other software professionals, Living Documentation shows how to create documentation that evolves throughout your entire design and development lifecycle. Through patterns, clarifying illustrations, and concrete examples, Cyrille Martraire demonstrates how to use well-crafted artifacts and automation to dramatically improve the value of documentation at minimal extra cost. Whatever your domain, language, or technologies, you don't have to choose between working software and comprehensive, high-quality documentation: you can have both.

- Extract and augment available knowledge, and make it useful through living curation
- Automate the creation of documentation and diagrams that evolve as knowledge changes
- Use development tools to refactor documentation
- Leverage documentation to improve software designs
- Introduce living documentation to new and legacy environments

The path for developing an internationally usable product with a human-machine interface is described in this textbook, from theory to conception and from design to practical implementation. The most important concepts in the fields of philosophy, communication, culture and Ethnocomputing as the basis of intercultural user interface design are explained. The book presents directly usable and implementable knowledge that is relevant for the processes of internationalization and localization of software. Aspects of software ergonomics, software engineering and human-centered design are presented in an intercultural context; general and concrete recommendations and checklists for immediate use in product design are also provided. Each chapter includes the target message, its motivation and theoretical justification as well as the practical methods to achieve the intended benefit from the respective topic. The book opens with an introduction illuminating the background necessary for taking culture into account in Human Computer Interaction (HCI) design. Definitions of

concepts are followed by a historical overview of the importance of taking culture into account in HCI design. Subsequently, the structures, processes, methods, models, and approaches concerning the relationship between culture and HCI design are illustrated to cover the most important questions in practice.

Part of the new Allyn & Bacon series in technical communication, *Writing Software Documentation* features a step-by-step strategy to writing and describing procedures. This task-oriented book is designed to support both college students taking a course and professionals working in the field. Teaching apparatus includes complete programs for students to work on and a full set of project tracking forms, as well as a broad range of examples including Windows-style pages and screens and award-winning examples from STC competitions. This series in *Computers and Medicine* had its origins when I met Jerry Stone of Springer-Verlag at a SCAMC meeting in 1982. We determined that there was a need for good collections of papers that would help disseminate the results of research and application in this field. I had already decided to do what is now *Information Systems for Patient Care*, and Jerry contributed the idea of making it part of a series. In 1984 the first book was published, and—thanks to Jerry's efforts—*Computers and Medicine* was underway. Since that time, there have been many changes. Sadly, Jerry died at a very early age and cannot share in the success of the series that he helped found. On the bright side, however, many of the early goals of the series have been met. As the result of equipment improvements and the consequent lowering of costs, computers are being used in a growing number of medical applications, and the health care community is very computer literate. Thus, the focus of concern has turned from learning about the technology to understanding how that technology can be exploited in a medical environment.

Looking for a way to invigorate your technical writing team and grow that expertise to include developers, designers, and writers of all backgrounds? When you treat docs like code, you multiply everyone's efforts and streamline processes through collaboration, automation, and innovation. Second edition now available with updates and more information about version control for documents and continuous publishing.

Pro Git (Second Edition) is your fully-updated guide to Git and its usage in the modern world. Git has come a long way since it was first developed by Linus Torvalds for Linux kernel development. It has taken the open source world by storm since its inception in 2005, and this book teaches you how to use it like a pro. Effective and well-implemented version control is a necessity for successful web projects, whether large or small. With this book you'll learn how to master the world of distributed version workflow, use the distributed features of Git to the full, and extend Git to meet your every need. Written by Git pros Scott Chacon and Ben Straub, *Pro Git (Second Edition)* builds on the hugely successful first edition, and is now fully updated for Git version 2.0, as well as including an indispensable chapter on GitHub. It's the best book for all your Git needs.

"This unique resource provides you with a practical approach to quickly learning the software-defined radio concepts you need to know for your work in the field. By prototyping and evaluating actual digital communication systems capable of performing "over-the-air" wireless data transmission and reception, this volume helps you attain a first-hand understanding of critical design trade-offs and issues. Moreover you gain a sense of the actual "real-world" operational behavior of these systems. With the purchase of the book, you gain access to several ready-made Simulink experiments at the publisher's website. This collection of laboratory experiments, along with several examples, enables you to successfully implement the designs discussed the book in a short period of time. These files can be executed using MATLAB version R2011b or later. "

A manual about a computer program that constitutes the electronic version of Novell print manuals on LAN maintenance and management.

The #1 guide to creating effective online documentation is now updated and expanded to reflect the latest technological advances, including multimedia.

"...online documentation is a different medium, as different from books as television is from radio or movies from novels. This edition treats online documentation as the new electronic medium it is." -William Horton
Written by an internationally renowned pioneer in the field of technical communication, this is an incomparable guide to the art and science of creating online documents and documentation systems. Rather than concentrating on any one particular program or operating system, William Horton cuts to the heart of effective human-computer interaction and extrapolates a set of universal principles that can be applied to any form of online documentation-from messages, menus, and help files, to computer tutorials and hypertexts. Maintaining an end-user's perspective throughout, he guides you step by step through every crucial design decision without ever losing sight of the final goal-clear, effective online documentation that people enjoy using. Proven techniques that help reduce support and training costs for software products, eliminate the need for paper documentation, make programs more appealing and easier to use, and more * A practical, hands-on approach, supported by the latest research and supplemented with dozens of case studies and illustrations * Includes new chapters on multimedia and computer-based training * Comprehensive coverage of all online documentation media-words, graphics, animation, and sound * Updated information on organizing and structuring documents-with examples from Windows, OS/2, and Macintosh interfaces

IT organizations face pressure to increase productivity, improve application performance, support global collaboration, improve data protection, and minimize costs. In today's WAN-centered environments, traditional LAN-oriented infrastructure approaches are insufficient to meet these goals. Application Acceleration and WAN Optimization Fundamentals introduces a better solution: integrating today's new generation of accelerator solutions to efficiently and effectively scale networks beyond traditional capabilities while improving performance and minimizing costs through consolidation. Ted Grevers and Joel Christner begin by reviewing the challenges network professionals face in delivering applications to globally distributed

workforces. You learn how accelerators are transforming application business models, enabling IT departments to centralize and consolidate resources while also delivering consistently superior performance. Grevers and Christner show how to identify network consumers, prioritize traffic, and guarantee appropriate throughput and response times to business-critical applications. You learn how to use quality of service techniques such as packet classification and marking and traffic policing, queuing, scheduling, and shaping. Next, you compare options for integrating accelerators and optimization services into your network and for optimizing content delivery. The authors show how to address application protocol-related performance problems that cannot be resolved through compression or flow optimization alone. In the final chapter, the authors walk you through several real-world scenarios for utilizing accelerator technology. Ted Grevers, Jr., is the solution manager for the Cisco® Video IPTV Systems Test and Architecture (C-VISTA) team. He has extensive experience in the content delivery network (CDN) market, focusing on enterprise and service provider content delivery and application optimization needs. Joel Christner, CCIE® No. 15311, is the manager of technical marketing for the Cisco Application Delivery Business Unit (ADBU). He has extensive experience with application protocols, acceleration technologies, LAN/WAN infrastructure, and storage networking. Grevers and Christner are key contributors to the design and architecture of Cisco application delivery and application acceleration solutions. Provide high-performance access to remote data, content, video, rich media, and applications Understand how accelerators can improve network performance and minimize bandwidth consumption Use NetFlow to baseline application requirements and network utilization Ensure network resources are allocated based on business priorities Identify performance barriers arising from networks, protocols, operating systems, hardware, file systems, and applications Employ application-specific acceleration components to mitigate the negative impact of latency and bandwidth consumption Integrate content delivery networks (CDN) to centrally manage the acquisition, security, and distribution of content to remote locations Leverage WAN optimization technologies to improve application throughput, mitigate the impact of latency and loss, and minimize bandwidth consumption Optimize the performance of WANs and business-critical WAN applications This book is part of the Cisco Press® Fundamentals Series. Books in this series introduce networking professionals to new networking technologies, covering network topologies, sample deployment concepts, protocols, and management techniques. Category: Cisco Press/Networking Covers: Network Optimization

Maximize the impact and precision of your message! Now in its fourth edition, the Microsoft Manual of Style provides essential guidance to content creators, journalists, technical writers, editors, and everyone else who writes about computer technology. Direct from the Editorial Style Board at Microsoft—you get a comprehensive glossary of both general technology terms and those specific to Microsoft; clear, concise usage and style guidelines with helpful examples and alternatives; guidance on grammar, tone, and voice; and best practices for writing content for the web, optimizing for accessibility, and communicating to a worldwide audience. Fully updated and optimized for ease of use, the Microsoft Manual of Style is designed to help you communicate clearly, consistently, and accurately about technical topics—across a range of audiences and media.

This special issue of the Journal of Automated Software Engineering contains four extended papers from the 10th Knowledge-Based Software Engineering Conference, held in Boston, Massachusetts in November 1995. The conference provides a forum for researchers and practitioners to discuss applications of automated reasoning, knowledge representation, and artificial intelligence techniques to software engineering problems. The papers included herein are the best paper award winners, or candidates for same.

"A clearly written book that is a useful primer for a very complicated set of topics." --Capers

Jones, Chief Scientist Emeritus, Software Productivity Research LLC Practical Software Estimation brings together today's most valuable tips, techniques, and best practices for accurately estimating software project efforts, costs, and schedules. Written by a leading expert in the field, it addresses the full spectrum of real-world challenges faced by those who must develop reliable estimates. M. A. Parthasarathy draws on the immense experience of Infosys, one of the world's largest and most respected providers of IT-enabled business solutions, to bring you the only book with detailed guidance on estimating insourced and outsourced software projects, as well as projects that blend both approaches. He demonstrates how to successfully utilize Function Point (FP) methods, the industry's leading estimation model. Then, using real case studies, he systematically identifies pitfalls that can lead to inaccurate estimates--and offers proven solutions. Coverage includes How to estimate all types of software projects, including "fresh" development, reengineering, and maintenance How to incorporate the impact of core project elements on estimates: scope, environment, experience, and tools FP analysis from start to finish: data and transaction functions, general system characteristics, and more FP methods for any platform or business function Innovative re-estimation methods to track progress How to quote RFPs and prepare contracts: fixed price, time/material, and project execution lifecycle models Alternatives to FP: Delphi, COCOMO II, and COSMIC-FFP How to choose the right estimation tools Practical Software Estimation is the definitive reference for anyone who must estimate software projects accurately: project and IT managers, individual developers, system designers, architects, executives, consultants, and outsourcers alike. List of Figures List of Tables Foreword Preface Acknowledgments Chapter 1: Introduction Chapter 2: Role of Estimation in Software Projects Chapter 3: A Study of Function Point Analysis Chapter 4: Data Functions Chapter 5: Transactional Functions Chapter 6: General System Characteristics Chapter 7: Size, Effort, and Scheduling of Projects Chapter 8: Estimation Flavors Chapter 9: A Sense of Where You Are Chapter 10: Tips, Tricks, and Traps Chapter 11: Insourcing versus Outsourcing Chapter 12: Key Factors in Software Contracts Chapter 13: Project Estimation and Costing Chapter 14: Other Estimation Methods Chapter 15: Estimation Tools Chapter 16: Estimation Case Study Appendix A: Reference Tables: Transaction Function Counts Appendix B: Reference Tables: Data Function Points Bibliography Index

Software development has been a troubling since it first started. There are seven chronic problems that have plagued it from the beginning: Incomplete and ambiguous user requirements that grow by >2% per month. Major cost and schedule overruns for large applications > 35% higher than planned. Low defect removal efficiency (DRE) Cancelled projects that are not completed: > 30% above 10,000 function points. Poor quality and low reliability after the software is delivered: > 5 bugs per FP. Breach of contract litigation against software outsource vendors. Expensive maintenance and enhancement costs after delivery. These are endemic problems for software executives, software engineers and software customers but they are not insurmountable. In Software Development Patterns and Antipatterns, software engineering and metrics pioneer Capers Jones presents technical solutions for all seven. The solutions involve moving from harmful patterns of software development to effective patterns of software development. The first section of the book examines common software development problems that have been observed in many companies and government agencies. The data on the problems comes from consulting studies, breach of contract lawsuits, and the literature on major software failures. This section considers the factors involved with cost overruns, schedule delays, canceled projects, poor quality, and expensive maintenance after deployment. The second section shows patterns that lead to software success. The data comes from actual companies. The section's first chapter on Corporate Software Risk Reduction in a Fortune 500 company was based on a major telecom company whose CEO was troubled by repeated software failures. The other chapters

in this section deal with methods of achieving excellence, as well as measures that can prove excellence to C-level executives, and with continuing excellence through the maintenance cycle as well as for software development.

This book constitutes the refereed proceedings of the 9th International Conference on Principles and Practice of Constraint Programming, CP 2003, held in Kinsale, Ireland in September/October 2003. The 48 revised full papers and 34 revised short papers presented together with 4 invited papers and 40 abstracts of contributions to the CP 2003 doctoral program were carefully reviewed and selected from 181 submissions. A wealth of recent results in computing with constraints is addressed ranging from foundational and methodological issues to solving real-world problems in a variety of application fields.

For more than 40 years, Computerworld has been the leading source of technology news and information for IT influencers worldwide. Computerworld's award-winning Web site (Computerworld.com), twice-monthly publication, focused conference series and custom research form the hub of the world's largest global IT media network.

Just like vinyl LPs, static sites are making a comeback, evidenced by the wide array of static-site generators now available. This practical book shows you hands-on how to build these simple sites for blogs and other use cases, and how to make them more powerful. In the process, you'll work with some of today's more mature and popular static-site generators. Authors Raymond Camden and Brian Rinaldi explain the advantages of using static-site generators for building fast and secure sites. Web and frontend designers and developers will also explore methods for adding dynamic elements and for migrating an existing CMS to a static site. Build a basic four-page static site with the Harp generator Create a simple blog with Jekyll Develop a documentation site with Hugo by generating site files and creating the layout Add dynamic elements, such as forms, comments, and search Integrate a CMS with tools such as CloudCannon and Netlify CMS Use one of several options to deploy your static files Learn methods for moving an existing CMS to a static site

Designed to address the randomness of the literature on software documentation. This book contains a variety of perspectives, tied together by the need to make software products more usable.

This book contains both relevant real-world research, as well as reviews of different areas of interest in the software engineering literature, such as clone identification. The contents of the various sections will provide a better understanding of known problems and detailed treatment of advanced topics. Consequently, the book consolidates the work and findings from leading researchers in the software research community in key areas such as maintainability, architectural recovery, code analysis, software migration, and tool support.

Includes articles in topic areas such as autonomic computing, operating system architectures, and open source software technologies and applications.

Perspectives on Software DocumentationInquiries and InnovationsRoutledge

[Copyright: 94d8eb17933012b2ed3d4d902b3e7d73](https://doi.org/10.1002/9781119999999)