Acces PDF Eliciting And Documenting Usability Requirements
Acces PDF Eliciting And Documenting Usability Requirements

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

Software Requirements

Developing today's complex systems requires more than just good software engineering solutions. Many are faced with complex systems projects, incomplete or inaccurate requirements, canceled projects, or cost overruns, and have their systems' users in revolt and demanding more. Others want to build user-centric systems, but fear managing the process. This book describes an approach that brings the engineering process together with human performance engineering and business process reengineering. The result is a manageable user-centered process for gathering, analyzing, and evaluating requirements that can vastly improve the success rate in the development of medium-to-large size systems and applications. Unlike some texts that are primarily conceptual, this volume provides guidelines, "how-to" information, and examples, enabling the reader to quickly apply the process and techniques to accomplish the following goals: * define high quality requirements, * enhance productive client involvement, * help clients maintain competitiveness, * ensure client buy-in and support throughout the process, * reduce missing functionality and corrections, and * improve user satisfaction with systems. This volume clearly details the role of user-centered requirements and knowledge acquisition within Scenario-Based Engineering Process (SEP) and identifies SEP products and artifacts. It assists project personnel in planning and managing effective requirements activities, including managing risks, avoiding common problems with requirements elicitation, organizing project participants and tools, and managing the logistics. Guidelines are provided for the following: selecting the right individual and group techniques to elicit scenarios and requirements from users; subject matter experts, or other shareholders; and ensuring engineers or analysts have the necessary skills.

Human-Centered and Error-Resilient Systems Development

Projects are inherently risky, since they involve some level of uncertainty, doing something new in the target environment, but the percentage of projects seen as a success is still disappointingly low, especially for IT projects. The 'Iron Triangle' of time/cost/quality suggests that all three aspects are equal, but with quantitative methods for monitoring project performance, the focus is primarily on managing cost and time. This book seeks to redress the balance, explaining the rationale and benefits of focusing more on quality (fitness for purpose and conformance to requirements) before detailing a range of tools and techniques to support rebalancing the management of projects, programmes and portfolios. It shows how managing project quality actively can reduce costs through minimising wastage, and reduce delays through avoiding rework, leading to improved project success rates and customer satisfaction.

Requirements Engineering: Foundation for Software Quality

Knowledge is power, but this is especially true for teams carrying out a project. As in other arenas, the effective use of knowledge is possible only if it is readily accessible, well organized, properly analyzed, and competently disseminated to meet the project needs. Knowledge gained from project failures or successes is vital for the long term sustainability of organizations to compete in the business environment. This book focuses on the proper access and delivery methods for explicit knowledge in projects and also concentrates on tacit knowledge unknown and unavailable to most people in project environments. Every project is unique with start and end dates, detailed project plan, budget, schedule, human resources, and deliverables, and all these areas have a high volume of rich knowledge. Knowledge is created and flows through all nine project knowledge areas: Project Integration Management; Project Scope Management; Project Time Management; Project Cost Management; Project Quality Management; Project Human Resources Management; Project Communications Management; Project Risk Management; and Project Procurement Management. This book discusses the benefits of managing knowledge in projects and provides techniques that will increase the rate of return on projects. Addressing strategy and deployment issues, this volume also provides case studies, making this an invaluable tool for the success of projects and sustainability/growth of organizations.

The UX Book

Penetrates the human computer interaction (HCI) field with breadth and depth of comprehensive research.
This book presents research on the most recent technological developments in all fields of knowledge or disciplines of computer games development, including planning, design, development, marketing, business management, users and behavior.--Provided by publisher.

Usability- and Accessibility-Focused Requirements Engineering

User-centered Requirements

In a down-to-the-earth manner, the volume lucidly presents how the fundamental concepts, methodology, and algorithms of Computational Intelligence are efficiently exploited in Software Engineering and opens up a novel and promising avenue of a comprehensive analysis and advanced design of software artifacts. It shows how the paradigm and the best practices of Computational Intelligence can be creatively explored to carry out comprehensive software requirement analysis, support design, testing, and maintenance. Software Engineering is an intensive knowledge-based endeavor of inherent human-centric nature, which profoundly relies on acquiring semiformal knowledge and then processing it to produce a running system. The knowledge spans a wide variety of artifacts, from requirements, captured in the interaction with customers, to design practices, testing, and code management strategies, which rely on the knowledge of the running system. This volume consists of contributions written by widely acknowledged experts in the field who reveal how the Software Engineering benefits from the key foundations and synergistically existing technologies of Computational Intelligence being focused on knowledge representation, learning mechanisms, and population-based global optimization strategies. This book can serve as a highly useful reference material for researchers, software engineers and graduate students and senior undergraduate students in Software Engineering and its sub-disciplines, Internet engineering, Computational Intelligence, management, operations research, and knowledge-based systems.

CBAP / CCBA Certified Business Analysis Study Guide

Industrial engineering affects all levels of society, with innovations in manufacturing and other forms of engineering oftentimes spawning cultural or educational shifts along with new technologies. Industrial Engineering: Concepts, Methodologies, Tools, and Applications serves as a vital compendium of research, detailing the latest research, theories, and case studies on industrial engineering. Bringing together contributions from authors around the world, this three-volume collection represents the most sophisticated research and developments from the field of industrial engineering and will prove a valuable resource for researchers, academics, and practitioners alike.

Human-System Integration in the System Development Process

This book constitutes the proceedings of the first Asia Pacific Requirements Engineering Symposium, APRES 2014, held in Auckland, New Zealand, in April 2014. The 16 papers presented were carefully reviewed and selected from 30 submissions. The focus of the papers is on the following topics: novel ideas, methods, tools, and techniques for improving and enhancing Requirement Engineering products and processes.

The Usability Engineering Lifecycle

"This book provides innovative ideas and methods on the development, operation, and maintenance of secure software systems and highlights the construction of a functional software system and a secure system simultaneously"--Provided by publisher.

The UX Careers Handbook

Includes articles in topic areas such as autonomic computing, operating system architectures, and open source software technologies and applications.
Acces PDF Eliciting And Documenting Usability Requirements

Taking a unique approach to systems analysis and design, this insightful book provides learners with a critical personal framework for considering and developing knowledge and practice of systems analysis and design. Each chapter begins by highlighting what can be learned on its completion and ends with a critical skills development section containing activities, tasks and discussion questions. Chapters cover: * systems analysis and design in concept and action * structured data modelling * making systems analysis and design inclusive. Although the discussion and examples in this text are drawn primarily from business information systems, the lessons apply to both government and healthcare information systems and to systems development in general. Critical Systems Analysis and Design makes a complex area of study accessible and relevant and as such is an indispensable textbook for both advanced students and professionals concerned with the innovation of information systems.

Online Communities and Social Computing

Most IT systems fail to meet expectations. They don't meet business goals and don't support users efficiently. Why? Because the requirements didn't address the right issues. Writing a good requirements specification doesn't take more time. This book shows how it's done - many times faster and many times smarter. What are the highlights? Two complete real-life requirements specifications (the traditional and the fast approach) and examples from many others. Explanations of both traditional and fast approaches, and discussions of their strengths and weaknesses in different project types (tailor-made, COTS, and product development). Real-life illustrations of all types of requirements, stakeholder analysis, cost/benefit and other techniques to ensure that business goals are met. Proven methods for dealing with difficult or complex requirements, such as specifying ease-of-use, or dealing with 200 reports that might be needed because they are in the old system. Who is it for? Everyone involved in the software supply chain, from analysts and developers to end users, will learn new techniques, benefit from requirements written by other specialists, and discover successes and failures from other companies. Software suppliers will find ideas for helping customers and writing competitive proposals. Programmers and other developers will learn how to express requirements without specifying technical details, and how to reduce risks when developing a system. Students aspiring to IT careers will learn the theory and practice of requirements engineering, and get a strong foundation for case studies and projects. Who is the author? Soren Lauesen is currently professor at the IT-University of Copenhagen. He has worked in the IT industry for 20 years and has been a professor at Copenhagen Business School for 15. He has been co-founder of three educational and two industrial development organizations. His industry projects have encompassed compilers, operating systems, process control, temporal databases, and software quality assurance. His research interests include human-computer interaction, requirements specification, object-oriented design, quality assurance, marketing and product development, and interaction between research and industry. He has a broad range of other interests ranging from biology to dancing and foreign cultures.

Information Systems Development

Your go-to guide on business analysis Business analysis refers to the set of tasks and activities that help companies determine their objectives for meeting certain opportunities or addressing challenges and then help them define solutions to meet those objectives. Those engaged in business analysis are charged with identifying the activities that enable the company to define the business problem or opportunity, define what the solutions looks like, and define how it should behave in the end. As a BA, you lay out the plans for the process ahead. Business Analysis For Dummies is the go to reference on how to make the complex topic of business analysis easy to understand. Whether you are new or have experience with business analysis, this book gives you the tools, techniques, tips and tricks to set your project's expectations and on the path to success. Offers guidance on how to make an impact in your organization by performing business analysis Shows you the tools and techniques to be an effective business analysis professional Provides a number of examples on how to perform business analysis regardless of your role If you're interested in learning about the tools and techniques used by successful business analysis professionals, Business Analysis For Dummies has you covered.

Project Success and Quality

"This book explores different applications in V & V that spawn many areas of software development - including real time applications - where V & V techniques are required, providing in all cases examples of the applications." - Provided by publisher.

Requirements Engineering for Sociotechnical Systems
The UX Book: Process and Guidelines for Ensuring a Quality User Experience aims to help readers learn how to create and refine interaction designs that ensure a quality user experience (UX). The book seeks to expand the concept of traditional usability to a broader notion of user experience; to provide a hands-on, practical guide to best practices and established principles in a UX lifecycle; and to describe a pragmatic process for managing the overall development effort. The book provides an iterative and evaluation-centered UX lifecycle template, called the Wheel, for interaction design. Key concepts discussed include contextual inquiry and analysis; extracting interaction design requirements; constructing design-informing models; design production; UX goals, metrics, and targets; prototyping; UX evaluation; the interaction cycle and the user action framework; and UX design guidelines. This book will be useful to anyone interested in learning more about creating interaction designs to ensure a quality user experience. These include interaction designers, graphic designers, usability analysts, software engineers, programmers, systems analysts, software quality-assurance specialists, human factors engineers, cognitive psychologists, cosmic psychics, trainers, technical writers, documentation specialists, marketing personnel, and project managers. A very broad approach to user experience through its components—usability, usefulness, and emotional impact with special attention to lightweight methods such as rapid UX evaluation techniques and an agile UX development process Universal applicability of processes, principles, and guidelines—not just for GUIs and the Web, but for all kinds of interaction and devices: embodied interaction, mobile devices, ATMs, refrigerators, and elevator controls, and even highway signage Extensive design guidelines applied in the context of the various kinds of affordances necessary to support all aspects of interaction Real-world stories and contributions from accomplished UX practitioners A practical guide to best practices and established principles in UX A lifecycle template that can be instantiated and tailored to a given project, for a given type of system development, on a given budget

Human-Centered Software Engineering

In April 1991 BusinessWeek ran a cover story entitled, “I Can’t Work This ?#!!@ Thing,† about the difficulties many people have with consumer products, such as cell phones and VCRs. More than 15 years later, the situation is much the same—but at a very different level of scale. The disconnect between people and technology has had society-wide consequences in the large-scale system accidents from major human error, such as those at Three Mile Island and in Chernobyl. To prevent both the individually annoying and nationally significant consequences, human capabilities and needs must be considered early and throughout system design and development. One challenge for such consideration has been providing the background and data needed for the seamless integration of humans into the design process from various perspectives: human factors engineering, manpower, personnel, training, safety and health, and, in the military, habitability and survivability. This collection of development activities has come to be called human-system integration (HSI). Human-System Integration in the System Development Process reviews in detail more than 20 categories of HSI methods to provide invaluable guidance and information for system designers and developers.