Best Practices For Software Performance Engineering

In this comprehensive introduction to software measurement, Ebert and Dumke detail knowledge and experiences about the subject in an easily understood, hands-on presentation. The book describes software measurement in theory and practice as well as provides guidance to all relevant measurement tools and online references. In addition, it presents hands-on experience from industry leaders and provides many examples and case studies from Global 100 companies. Besides the many practical hints and checklists, readers will also appreciate the large reference list, which includes links to metrics communities where project experiences are shared.

What are the concrete Software performance results? Which Software performance data should be retained? Is there any existing Software performance governance structure? How are consistent Software performance definitions important? How would you define Software performance leadership? Defining, designing, creating, and implementing a process to solve a challenge or meet an objective is the most valuable role... In EVERY group, company, organization and department. Unless you are talking a one-time, single-use project, there should be a process. Whether that process is managed and implemented by humans, AI, or a combination of the two, it needs to be designed by someone with a complex enough perspective to ask the right questions. Someone capable of asking the right questions and step back and say, 'What are we really trying to accomplish here? And is there a different way to look at it?' This Self-Assessment empowers people to do just that - whether their title is entrepreneur, manager, consultant, (Vice-)President, CxO etc... - they are the people who rule the future. They are the person who asks the right questions to make Software Performance investments work better. This Software Performance All-Inclusive Self-Assessment enables You to be that person. All the tools you need to an in-depth Software Performance Self-Assessment. Featuring 949 new and updated case-based questions, organized into seven core areas of process design, this Self-Assessment will help you identify areas in which Software Performance improvements can be made. In using the questions you will be better able to: - diagnose Software Performance projects, initiatives, organizations, businesses and processes using accepted diagnostic standards and practices - implement evidence-based best practice strategies aligned with overall goals - integrate recent advances in Software Performance and process design strategies into practice according to best practice guidelines Using a Self-Assessment tool known as the Software Performance Scorecard, you will develop a clear picture of which Software Performance areas need attention. Your purchase includes access details to the Software Performance self-assessment dashboard download which gives you your dynamically prioritized projects-ready tool and shows your organization exactly what to do next. You will receive the following contents with New and Updated specific criteria: - The latest quick edition of the book in PDF - The latest complete edition of the book in PDF, which criteria correspond to the criteria in... - The Self-Assessment Excel Dashboard - Example pre-filled Self-Assessment Excel Dashboard to get familiar with results generation - In-depth and specific Software Performance Checklists - Project management checklists and templates to assist with implementation INCLUDES LIFETIME SELF ASSESSMENT UPDATES Every self assessment comes with Lifetime Updates and Lifetime Free Updated Books. Lifetime Updates is an industry-first feature which allows you to receive verified self assessment updates, ensuring you always have the most accurate information at your fingertips.

Page 26: How can I avoid off-by-one errors? Page 143: Are Trojan Horse attacks for real? Page 158: Where should I look when my application can't handle its workload? Page 256: How can I detect memory leaks? Page 309: How do I target my application to international markets? Page 394: How should I name my code's identifiers? Page 441: How can I find and improve the code coverage of my tests? Diomidis Spinellis' first book, Code Reading, showed programmers how to understand and modify key functional properties of software. Code Quality focuses on non-functional properties, demonstrating how to meet such critical requirements as reliability, security, portability, and maintainability, as well as efficiency in time and space. Spinellis draws on hundreds of examples from open source projects--such as the Apache web and application servers, the BSD Unix systems, and the HSQLDB Java database--to illustrate concepts and techniques that every professional software developer will be able to appreciate and apply immediately. Complete files for the open source code illustrated in this book are available online at: http://www.spinellis.gr/codequality/

Consistent success does not happen by chance. It occurs by having an understanding of what is happening in the environment and then having the skills to execute the necessary changes. Ideal for project, IT, and systems development managers, IT Best Practices: Management, Teams, Quality, Performance, and Projects details the skills, knowledge, and a

SUPPLY CHAIN MANAGEMENT

How to Use Metrics to Improve Project and Process Performance ; 37 Tables

The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations

Software Engineering Best Practices

PERFORMANCE, PRODUCTIVITY AND INNOVATION

IBM FlashSystem Best Practices and Performance Guidelines for IBM Spectrum Virtualize Version 8.4.2

Chapter 10. Software Performance Engineering for Embedded Systems

Power and Performance: Software Analysis and Optimization is a guide to solving performance problems in modern Linux systems. Power-efficient chips are no help if the software those chips run on is inefficient. Starting with the necessary architectural background as a foundation, the book demonstrates the proper usage of performance analysis tools in order to pinpoint the cause of performance problems, and includes best practices for handling common performance issues those tools identify. Provides expert perspective from a key member of Intel ' s optimization team on how processors and memory systems influence performance Presents ideas to improve architectures running mobile, desktop, or enterprise platforms Demonstrates best practices for designing experiments and benchmarking throughout the software lifecycle Explains the importance of profiling and measurement to determine the source of performance issues

The objective of APM Best Practices: Realizing Application Performance Management is to establish reliable application performance management (APM) practices–to demonstrate value, to do it quickly, and to adapt to the client circumstances. It's important to balance long-term goals with short-term deliverables, but without compromising usefulness or correctness. The successful strategy is to establish a few reasonable goals, achieve them quickly, and then iterate over the same topics two more times, with each successive iteration expanding the skills and capabilities of the APM team. This strategy is referred to as “Good, Better, Best”. The application performance monitoring marketplace is very focused on ease of installation, rapid time to usefulness, and overall ease of use. But these worthy platitudes do not really address the application performance management processes that ensure that you will deploy effectively, synergize on quality assurance test plans, triage accurately, and encourage collaboration across the application life cycle that ultimately lowers overall application cost and ensures a quality user experience. These are also fine platitudes but these are the ones that are of interest to your application sponsors. These are the ones for which you need to show value. This CA Press book employs this iterative approach, adapted pragmatically for the realities of your organizational and operational constraints, to realize a future state that your sponsors will find useful, predictable and manageable–and something that they will want to fund. In the meantime, you will learn the useful techniques needed to set up and maintain a useful performance management system utilizing best practices regardless of the software provider(s).

This IBM ® Redbooks ® publication provides performance tuning tips and best practices for IBM Business Process Manager (IBM BPM) V8.5.5 (all editions) and IBM Business Monitor V8.5.5. These products represent an integrated development and runtime environment based on a key set of service-oriented architecture (SOA) and business process management (BPM) technologies. Such technologies include Service Component Architecture (SCA), Service Data Object (SDO), Business Process Execution Language (BPEL) for web services, and Business Processing Modeling Notation (BPMN). Both IBM Business Process Manager and Business Monitor build on the core capabilities of the IBM WebSphere ® Application Server infrastructure. As a result, Business Process Manager solutions benefit from tuning, configuration, and best practices information for WebSphere Application Server and the corresponding platform Java virtual machines (JVMs). This book targets a wide variety of groups, both within IBM (development, services, technical sales, and others) and customers. For customers who are either considering or are in the early stages of implementing a solution incorporating Business Process Manager and Business Monitor, this document proves a useful reference. The book is useful both in terms of best practices during application development and deployment and as a reference for setup, tuning, and configuration information. This book talks about many issues that can influence performance of each product and can serve as a guide for making rational first choices in terms of configuration and performance settings. Similarly, customers who already implemented a solution with these products can use the information presented here to gain insight into how their overall integrated solution performance can be improved.

Embedded systems often have one or more real-time requirements. The complexity of modern embedded software systems requires a systematic approach for achieving these performance targets. An ad hoc process can lead to missed deadlines, poorly performing systems and cancelled projects. There is a maturity required to define, manage, and deliver on multiple real-time performance requirements. Software performance engineering (SPE) is a discipline within the broader systems engineering area that can improve the maturity of the performance engineering process. SPE is a systematic, quantitative approach to constructing software systems that meet performance objectives. SPE is a software-oriented approach; it focuses on architecture, design, and implementation choices. It focuses on the activities, techniques, and deliverables that are applied at every phase of the embedded software development life-cycle, especially responsiveness and scalability, to ensure software is being architected and implemented to meet the performance-related requirements of the system.

A Complete Guide for Software Project Estimators

How to use metrics to improve project and process performance

The Open Source Perspective

Power and Performance

Best Practices in Software Measurement

IBM SAN Volume Controller Best Practices and Performance Guidelines

Management, Teams, Quality, Performance, and Projects

Is there any additional Software performance testing definition of success? Are you assessing Software performance testing and risk? Think of your Software performance testing project, what are the main functions? Who will be responsible for documenting the Software performance testing requirements in detail? What qualifications do Software performance testing leaders need? Defining, designing, creating, and implementing a process to solve a challenge or meet an objective is the most valuable role... In EVERY group, company, organization and department. Unless you are talking a one-time, single-use project, there should be a process. Whether that process is managed and implemented by humans, AI, or a combination of the two, it needs to be designed by someone with a complex enough perspective to ask the right questions. Someone capable of asking the right questions and step back and say, 'What are we really trying to accomplish here? And is there a different way to look at it?' This Self-Assessment empowers people to do just that - whether their title is entrepreneur, manager, consultant, (Vice-)President, CxO etc... - they are the people who rule the future. They are the person who asks the right questions to make Software Performance Testing investments work better. This Software Performance Testing All-Inclusive Self-Assessment enables You to be that person. All the tools you need to an in-depth Software Performance Testing Self-Assessment. Featuring 944 new and updated case-based questions, organized into seven core areas of process design, this Self-Assessment will help you identify areas in which Software Performance Testing improvements can be made. In using the questions you will be better able to: - diagnose Software Performance Testing projects, initiatives, organizations, businesses and processes using accepted diagnostic standards and practices - implement evidence-based best practice strategies aligned with overall goals - integrate recent advances in Software Performance Testing and process design strategies into practice according to best practice guidelines Using a Self-Assessment tool known as the Software Performance Testing Scorecard, you will develop a clear picture of which Software Performance Testing areas need attention. Your purchase includes access details to the Software Performance Testing self-assessment dashboard download which gives you your dynamically prioritized projects-ready tool and shows your organization exactly what to do next. You will receive the following contents with New and Updated specific criteria: - The latest quick edition of the book in PDF - The latest complete edition of the book in PDF, which criteria correspond to the criteria in... - The Self-Assessment Excel Dashboard - Example pre-filled Self-Assessment Excel Dashboard to get familiar with results generation - In-depth and specific Software Performance Testing Checklists - Project management checklists and templates to assist with implementation INCLUDES LIFETIME SELF ASSESSMENT UPDATES Every self assessment comes with Lifetime Updates and Lifetime Free Updated Books. Lifetime Updates is an industry-first feature which allows you to receive verified self assessment updates, ensuring you always have the most accurate information at your fingertips. Winner of the Shingo Publication Award Accelerate your organization to win in the marketplace. How can we apply technology to drive business value? For years, we've been told that the performance of software delivery teams doesn't matter—that it can't provide a competitive advantage to our companies. Through four years of groundbreaking research to include data collected from the State of DevOps reports conducted with Puppet, Dr. Nicole Forsgren, Jez Humble, and Gene Kim set out to find a way to measure software delivery performance—and what drives it—using rigorous statistical methods. This book presents both the findings and the science behind that research, making the information accessible for readers to apply in their own organizations. Readers will discover how to measure the performance of their teams, and what capabilities they should invest in to drive higher performance. This book is ideal for management at every level.

Pro .NET Best Practices is a practical reference to the best practices that you can apply to your .NET projects today. You will learn standards, techniques, and conventions that are sharply focused, realistic and helpful for achieving results, steering clear of unproven, idealistic, and impractical recommendations. Pro .NET Best Practices covers a broad range of practices and principles that development experts agree are the right ways to develop software, which includes continuous integration, automated testing, automated deployment, and code analysis. Whether the solution is from a free and open source or a commercial offering, you will learn how to get a continuous integration server running and executing builds every time code changes. You will write clearer and more maintainable automated testing code that focuses on prevention and helping your .NET project succeed. By learning and following the .NET best practices in this book, you will avoid making the same mistakes once. With this book at your side, you'll get: Real-world, no-nonsense approaches to continuous integration, automated testing, automated deployment, and code analysis Tips and tricks you'll need to clear hurdles that keep others from putting these common sense ideas into common practice Guidance from the minimal, essential approach all the way to what's necessary to deliver at the highest levels of quality and effectiveness Benefit immediately, even before finishing it, from the knowledge, workable advice, and experience found in Pro .NET Best Practices.

Almost every software project begins with the utterances, “What will this cost?” and “When will this project be done?” Once those words are spoken, project stakeholders begin to wrestle with how to produce an estimate. Accurately estimating the cost or time to complete a software project is a serious problem for many software engineers, developers and project managers who struggle with costs running double original estimates, putting their careers at risk. It is reported that nearly 50% of all software projects are shelved and that one of the major causes is poor estimation practices. If developing software for internal use, poor estimates can represent a significant drain on corporate profits. Worldwide growth in the number of companies specializing in the development of software for use by other companies is staggering. India alone has nearly 20,000 such companies. Intense competition has led to an increased demand for fixed-bid pricing in client/vendor relationships, and has made effective cost estimation even more important and, in many cases, critical to a firm's survival. There are many methods of estimation. Each method has its strengths and weaknesses, proponents and opponents. Knowing how and which one to use on a given project is key to developing acceptable estimates for either internal or external projects.Software Estimation Best Practices, Tools, & Techniques covers all facets of software estimation. It provides a detailed explanation of the various methods for estimating software size, development effort, cost, and schedule, including a comprehensive explanation of Test Effort Estimation. Emphasizing that software estimation should be based on a well-defined process, it presents software estimation best practices and shows how to avoid common pitfalls. This guide offers direction on which methods are most appropriate for each of the different project types commonly executed in the software development space and criteria for selecting software estimation tools. This comprehensive desk reference explains software estimation from scratch to help the beginner and features advanced techniques for more experienced estimators. It details project scheduling, including resource leveling and the concept of productivity, as applicable to software estimators, demonstrating the many benefits of moving from the current macro-productivity approach to a micro-productivity approach in software estimation. Software Estimation Best Practices, Tools, & Techniques: A Complete Guide for Software Project Estimators caters to the needs of all software project stakeholders, from novice to expert. It provides the valuable guidance needed to estimate the cost and time required to complete software projects within a reasonable margin of error for effective software development.

Introduction, Management, and Performance

Refactor your legacy C# code base and improve application performance by applying best practices

IT Best Practices

Best Practices in ERP Software Applications

Performance Management of Integrated Systems and its Applications in Software Engineering

How Google Runs Production Systems

Realizing Application Performance Management

Practical approach to software measurement Contains hands-on industry experiences

This Expert Guide gives you the techniques and technologies in software engineering to optimally design and implement your embedded system. Written by experts with a solutions focus, this encyclopedic reference gives you an indispensable aid to tackling the day-to-day problems when using software engineering methods to develop your embedded systems. With this book you will learn: The principles of good architecture for an embedded system Design practices to help make your embedded project successful Details on principles that are often a part of embedded systems, including digital signal processing, safety-critical principles, and development processes Techniques for setting up a performance engineering strategy for your embedded system software How to develop user interfaces for embedded systems Strategies for testing and deploying your embedded system, and ensuring quality development processes Practical techniques for optimizing embedded software for performance, memory, and power Advanced guidelines for developing multicore software for embedded systems How to develop embedded software for networking, storage, and automotive segments How to manage the embedded development process Includes contributions from: Frank Schirrmeister, Shelly Gretlein, Bruce Douglass, Erich Styger, Gary Stringham, Jean Labrosse, Jim Trudeau, Mike Brogioli, Mark Pitchford, Catalin Dan Udma, Markus Levy, Pete Wilson, Whit Waldo, Inga Harris, Xinxin Yang, Srinivasa Addepalli, Andrew McKay, Mark Kraeling and Robert Oshana. Road map of key problems/issues and references to their solution in the text Review of core methods in the context of how to apply them Examples demonstrating timeless implementation details Short and to- the- point case studies show how key ideas can be implemented, the rationale for choices made, and design guidelines and trade-offs

A practical guide to help you undertand the ability of Apache jMeter to load and performance test various server types in a more efficient way. About This Book Use jMeter to create and run tests to improve the performance of your webpages and applications Learn to build a test plan for your websites and analyze the results Unleash the power of various features and changes introduced in Apache jMeter 3.0 Who This Book Is For This book is for software professionals who want to understand and improve the performance of their applications with Apache jMeter. What You Will Learn See why performance testing is necessary and learn how to set up JMeter Record and test with JMeter Handle various form inputs in JMeter and parse results during testing Manage user sessions in web applications in the context of a JMeter test Monitor JMeter results in real time Perform distributed testing with JMeter Get acquainted with helpful tips and best practices for working with JMeter In Detail JMeter is a Java application designed to load and test performance for web application. JMeter extends to improve the functioning of various other static and dynamic resources. This book is a great starting point to learn about JMeter. It covers the new features introduced with JMeter 3 and enables you to dive deep into the new techniques needed for measuring your website performance. The book starts with the basics of performance testing and guides you through recording your first test scenario, before diving deeper into JMeter. You will also learn how to configure JMeter and browsers to help record test plans. Moving on, you will learn how to capture form submission in JMeter, dive into managing sessions with JMeter and see how to leverage some of the components provided by JMeter to handle web application HTTP sessions. You will also learn how JMeter can help monitor tests in real-time. Further, you will go in depth into distributed testing and see how to leverage the capabilities of JMeter to accomplish this. You will get acquainted with some tips and best practices with regard to performance testing. By the end of the book, you will have learned how to take full advantage of the real power behind Apache JMeter. Style and approach The book is a practical guide starting with introducing the readers to the importance of automated testing. It will then be a beginner's journey from getting introduced to Apache jMeter to an in-detail discussion of more advanced features and possibilities with it.

Proven techniques for software engineering success This in-depth volume examines software engineering topics that are not covered elsewhere: the question of why software engineering has developed more than 2,500 programming languages; problems with traditional definitions of software quality; and problems with common metrics, "lines of code," and "cost per defect" that violate standard economic assumptions. The book notes that a majority of "new" projects are actually replacements for legacy applications, illustrating that data mining for lost requirements should be a standard practice. Difficult social engineering issues are also covered, such as how to minimize harm from layoffs and downsizing. Software Engineering Best Practices explains how to effectively plan, size, schedule, and manage software projects of all types, using solid engineering procedures. It details proven methods, from initial requirements through 20 years of maintenance. Portions of the book have been extensively reviewed by key engineers from top companies, including IBM, Microsoft, Unisys, and Sony. Manage Agile, hierarchical, matrix, and virtual software development teams Optimize software quality using JAD, OFD, TSP, static analysis, inspections, and other methods with proven success records Use high-speed functional metrics to assess productivity and quality levels Plan optimal organization, from small teams through more than 1,000 personnel

IBM Business Process Manager V8.5 Performance Tuning and Best Practices

Software Engineering for Embedded Systems

Best Practices for the Formal Software Testing Process

Software Performance A Complete Guide - 2020 Edition

Performance Best Practices for Web Developers

Even Faster Web Sites

Performance Engineering

The overwhelming majority of a software system’s lifespan is spent in use, not in design or implementation. So, why does conventional wisdom insist that software engineers focus primarily on the design and development of large-scale computing systems? In this collection of essays and articles, key members of Google’s Site Reliability Team explain how and why their commitment to the entire lifecycle has enabled the company to successfully build, deploy, monitor, and maintain some of the largest software systems in the world. You’ll learn the principles and practices that enable Google engineers to make systems more scalable, reliable, and efficient—lessons directly applicable to your organization. This book is divided into four sections: Introduction—Learn what site reliability engineering is and why it differs from conventional IT industry practices Principles—Examine the patterns, behaviors, and areas of concern that influence the work of a site reliability engineer (SRE) Practices—Understand the theory and practice of an SRE’s day-to-day work: building and operating large distributed computing systems Management—Explore Google's best practices for training, communication, and meetings that your organization can use

Develop your programming skills by exploring essential topics such as code reviews, implementing TDD and BDD, and designing APIs to overcome code inefficiency, redundancy, and other problems arising from bad code Key FeaturesWrite code that cleanly integrates with other systems while maintaining well-defined software boundariesUnderstand how coding principles and standards enhance software qualityLearn how to avoid common errors while implementing concurrency or threadingBook Description Traditionally associated with developing Windows desktop applications and games, C# is now used in a wide variety of domains, such as web and cloud apps, and has become increasingly popular for mobile development. Despite its extensive coding features, professionals experience problems related to efficiency, scalability, and maintainability because of bad code. Clean Code in C# will help you identify these problems and solve them using coding best practices. The book starts with a comparison of good and bad code, helping you understand the importance of coding standards, principles, and methodologies. You’ll then get to grips with code reviews and their role in improving your code while ensuring that you adhere to industry-recognized coding standards. This C# book covers unit testing, delves into test-driven development, and addresses cross-cutting concerns. You’ll explore good programming practices for objects, data structures, exception handling, and other aspects of writing C# computer programs. Once you’ve studied API design and discovered tools for improving code quality, you’ll look at examples of bad code and understand which coding practices you should avoid. By the end of this clean code book, you’ll have the developed skills you need in order to apply industry-approved coding practices to write clean, readable, extendable, and maintainable C# code. What you will learnWrite code that allows software to be modified and adapted over timeImplement the fail-pass-refactor methodology using a sample C# console applicationAddress cross-cutting concerns with the help of software design patternsWrite custom C# exceptions that provide meaningful informationIdentify poor quality C# code that needs to be refactoredSecure APIs with API keys and protect data using Azure Key VaultImprove your code’s performance by using tools for profiling and refactoringWho this book is for This coding book is for C# developers, team leads, senior software engineers, and software architects who want to improve the efficiency of their legacy systems. A strong understanding of C# programming is required.

This book presents a key solution for current and future technological issues, adopting an integrated system approach with a combination of software engineering applications. Focusing on how software dominates and influences the performance, reliability, maintainability and availability of complex integrated systems, it proposes a comprehensive method of improving the entire process. The book provides numerous qualitative and quantitative analyses and examples of varied systems to help readers understand and interpret the derived results and outcomes. In addition, it examines and reviews foundational work associated with decision and control systems for information systems, to inspire researchers and industry professionals to develop new and integrated foundations, theories, principles, and tools for information systems. It also offers guidance and suggests best practices for the research community and practitioners alike. The book’s twenty-two chapters examine and address current and future research topics in areas like vulnerability analysis, secured software requirements analysis, progressive models for planning and enhancing system efficiency, cloud computing, healthcare management, and integrating data-information-knowledge in decision-making. As such it enables organizations to adopt integrated approaches to system and software engineering, helping them implement technological advances and drive performance. This in turn provides actionable insights on each and every technical and managerial level so that timely action-based decisions can be taken to maintain a competitive edge. Featuring conceptual work and best practices in integrated systems and software engineering applications, this book is also a valuable resource for all researchers, graduate and undergraduate students, and management professionals with an interest in the fields of e-commerce, cloud computing, software engineering, software & system security and analysis, data-information-knowledge systems and integrated systems.

This book constitutes the refereed proceedings of the 5th Software Quality Days Conference (SWQD) held in Vienna, Austria, in January 2013. This professional symposium and conference offers a range of comprehensive and valuable opportunities for advanced professional training, new ideas, and networking with a series of keynote speeches, professional lectures, exhibits, and tutorials. The seven scientific full papers accepted for SWQD were each peer-reviewed by three or more reviewers and selected out of 18 high-quality submissions. Further, two keynotes and six short papers on promising research directions were also presented and included in order to spark discussions between researchers and practitioners. The papers are organized into topical sections on risk management; software and systems testing; test processes; model-based development; and process improvement and measurement.

Efficient MySQL Performance

Designing Data-Intensive Applications

Automated Software Testing

Software Performance Testing A Complete Guide - 2020 Edition

Information Theory and Best Practices in the IT Industry

Software Performance Testing a Complete Guide

Pro .NET Best Practices

Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords? In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse landscape by examining the pros and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications. Peer under the hood of the systems you already use, and learn how to use and operate them more effectively Make informed decisions by identifying the strengths and weaknesses of different tools Navigate the trade-offs around consistency, scalability, fault tolerance, and complexity Understand the distributed systems research upon which modern databases are built Peek behind the scenes of major online services, and learn from their architectures

With the urgent demand for rapid turnaround on new software releases--without compromising quality--the testing element of software development must keep pace, requiring a major shift from slow, labor-intensive testing methods to a faster and more thorough automated testing approach. Automated Software Testing is a comprehensive, step-by-step guide to the most effective tools, techniques, and methods for automated testing. Using numerous case studies of successful industry implementations, this book presents everything you need to know to successfully incorporate automated testing into the development process. In particular, this book focuses on the Automated Test Life Cycle Methodology (ATLM), a structured process for designing and executing testing that parallels the Rapid Application Development methodology commonly used today. Automated Software Testing is designed to lead you through each step of this structured program, from the initial decision to implement automated software testing through test planning, execution, and reporting. Included are test automation and test management guidance for: Acquiring management support Test tool evaluation and selection The automated testing introduction process Test effort and test team sizing Test team composition, recruiting, and management Test planning and preparation Test procedure development guidelines Automation reuse analysis and reuse library Best practices for test automation

Performance is critical to the success of any web site, and yet today's web applications push browsers to their limits with increasing amounts of rich content and heavy use of Ajax. In this book, Steve Souders, web performance evangelist at Google and former Chief Performance Yahoo!, provides valuable techniques to help you optimize your site's performance. Souders' previous book, the bestselling High Performance Web Sites, shocked the web development world by revealing that 80% of the time it takes for a web page to load is on the client side. In Even Faster Web Sites, Souders and eight expert contributors provide best practices and pragmatic advice for improving your site's performance in three critical categories: JavaScript—Get advice for understanding Ajax performance, writing efficient JavaScript, creating responsive applications, loading scripts without blocking other components, and more. Network—Learn to share resources across multiple domains, reduce image size without loss of quality, and use chunked encoding to render pages faster. Browser—Discover alternatives to iframes, how to simplify CSS selectors, and other techniques. Speed is essential for today's rich media web sites and Web 2.0 applications. With this book, you'll learn how to shave precious seconds off your sites' load times and make them respond even faster. This book contains six guest chapters contributed by Dion Almaer, Doug Crockford, Ben Galbraith, Tony Gentilcore, Dylan Schiemann, Stoyan Stefanov, Nicole Sullivan, and Nicholas C. Zakas.

Description: The book, Software Development Teams, offers a new and unique approach to developing software project teams. It guides IT experts and managers for forming, assessing and developing successful project management teams for effective performance and productivity. Focusing on the management side of the software industry, this text-cum-reference book discusses key aspects of the management such as performance measurement, organisational structure and development, motivation of the team with awards and rewards to bring innovative ideas, and the best practices followed in the modern software industry for measuring the team effectively. The book begins with an introduction of software teams, explaining how software projects are different. It then discusses the characteristics, skills and competencies that are required for a perfect programmer or a project manager, in addition to many other dimensions of software development teams. It further includes empirical studies on team climate, team performance, team productivity and team innovation. Next, it explores the factors that are important for maintaining the software development team climate, and the impact of conflicts on teams, which may ultimately have negative impact on the organisation. Tools and techniques to measure performance of software development team are explained along with the factors that influence the teams’ performance, relationship between team cohesion, productivity and finally the performance. Different types of possible innovation in software teams and organisations, innovation cycle and framework, role of top management and leadership in team management are also given due weightage. Providing an exhaustive description of the origin and present status of the Indian software industry using statistical data, the book is useful for the students of MBA (IT), BE/B.Tech (CS and IT), M.Tech (CS and IT) and M.Tech (Software Engineering). The book is also useful as a reference for professionals in the field of information systems, software project management, software engineering, team management and organisational development. Key features of the book • Highlights the latest studies in the field and cites inferences of various researchers. • Includes numerous figures, tables, graphs, and abbreviations to clarify the concepts. • Provides chapter-end questions and quick quiz (multiple choice questions with answers) to test the knowledge acquired. • Incorporates keywords and adequate number of references, which make the book an ideal tool for learning the concepts of software development teams. • Includes case studies to show the application of concepts of software development teams in real life scenarios.

Establish – Extract – Evaluate – Execute

Clean Code in C#

Software Performance Engineering Third Edition

CONCEPTS AND CASES

The Big Ideas Behind Reliable, Scalable, and Maintainable Systems

A Menu of Testing Tasks

APM Best Practices

Best Practices in Software MeasurementHow to Use Metrics to Improve Project and Process Performance ; 37 TablesSpringer Science & Business Media

What are the barriers to increased Software Performance engineering production? How do you manage Software Performance engineering Knowledge Management (KM)? What situation(s) led to this Software Performance engineering Self Assessment? How do you make it meaningful in connecting Software Performance engineering with what users do day-to-day? Is the measure of success for Software Performance engineering understandable to a variety of people? Defining, designing, creating, and implementing a process to solve a challenge or meet an objective is the most valuable role... In EVERY group, company, organization and department. Unless you are talking a one-time, single-use project, there should be a process. Whether that process is managed and implemented by humans, AI, or a combination of the two, it needs to be designed by someone with a complex enough perspective to ask the right questions. Someone capable of asking the right questions and step back and say, 'What are we really trying to accomplish here? And is there a different way to look at it?' This Self-Assessment empowers people to do just that - whether their title is entrepreneur, manager, consultant, (Vice-)President, CxO etc... - they are the people who rule the future. They are the person who asks the right questions to make Software Performance engineering investments work better. This Software Performance engineering All-Inclusive Self-Assessment enables You to be that person. All the tools you need to an in-depth Software Performance engineering Self-Assessment. Featuring 673 new and updated case-based questions, organized into seven core areas of process design, this Self-Assessment will help you identify areas in which Software Performance engineering improvements can be made. In using the questions you will be better able to: - diagnose Software Performance engineering projects, initiatives, organizations, businesses and processes using accepted diagnostic standards and practices - implement evidence-based best practice strategies aligned with overall goals - integrate recent advances in Software Performance engineering and process design strategies into practice according to best practice guidelines Using a Self-Assessment tool known as the Software Performance engineering Scorecard, you will develop a clear picture of which Software Performance engineering areas need attention. Your purchase includes access details to the Software Performance engineering self-assessment dashboard download which gives you your dynamically prioritized projects-ready tool and shows your organization exactly what to do next. You will receive the following contents with New and Updated specific criteria: - The latest quick edition of the book in PDF - The latest complete edition of the book in PDF, which criteria correspond to the criteria in... - The Self-Assessment Excel Dashboard - Example pre-filled Self-Assessment Excel Dashboard to get familiar with results generation - In-depth and specific Software Performance engineering Checklists - Project management checklists and templates to assist with implementation INCLUDES LIFETIME SELF ASSESSMENT UPDATES Every self assessment comes with Lifetime Updates and Lifetime Free Updated Books. Lifetime Updates is an industry-first feature which allows you to receive verified self assessment updates, ensuring you always have the most accurate information at your fingertips.

Initially, computer systems performance analyses were carried out primarily because of limited resources. Due to ever increasing functional complexity of computational systems and user requirements, performance engineering continues to play a major role in software development. This book assesses the state of the art in performance engineering. Besides revised chapters drawn from two workshops on performance engineering held in 2000, additional chapters were solicited in order to provide complete coverage of all relevant aspects. The first part is devoted to the relation between software engineering and performance engineering; the second part focuses on the use of models, measures, and tools; finally, case studies with regard to concrete technologies are presented. Researchers, professional software engineers, and advanced students interested in performance analysis will find this book an indispensable source of information and reference.

Software is important because it is used by a great many people in companies and institutions. This book presents engineering methods for designing and building software. Based on the author’s experience in software engineering as a programmer in the defense and aerospace industries, this book explains how to ensure a software that is programmed operates according to its requirements. It also shows how to develop, operate, and maintain software engineering capabilities by instilling an engineering discipline to support programming, design, builds, and delivery to customers. This book helps software engineers to: Understand the basic concepts, standards, and requirements of software engineering. Select the appropriate programming and design techniques. Effectively use software engineering tools and applications. Create specifications to comply with the software standards and requirements. Utilize various methods and techniques to identify defects. Manage changes to standards and requirements. Besides providing a technical view, this book discusses the moral and ethical responsibility of software engineers to ensure that the software they design and program does not cause serious problems. Software engineers tend to be concerned with the technical elegance of their software products and tools, whereas customers tend to be concerned only with whether a software product meets their needs and is easy and ready to use. This book looks at these two sides of software development and the challenges they present for software engineering. A critical understanding of software engineering empowers developers to choose the right methods for achieving effective results. Effective Methods for Software Engineering guides software programmers and developers to develop this critical understanding that is so crucial in today’s software-dependent society.

Software Estimation Best Practices, Tools & Techniques

Performance Testing with JMeter 3

SOFTWARE DEVELOPMENT TEAMS

Methods, Practical Techniques, and Applications

Code Quality

Effective Methods for Software Engineering

Effective Teamwork, Practical Integration

This IBM® Redbooks® publication captures several of the preferred practices and describes the performance gains that can be achieved by implementing the IBM FlashSystem® products that are powered by IBM Spectrum® Virtualize Version 8.4.2. These practices are based on field experience. This book highlights configuration guidelines and preferred practices for the storage area network (SAN) topology, clustered system, back-end storage, storage pools and managed disks, volumes, Remote Copy services, and hosts. It explains how you can optimize disk performance with the IBM System Storage Easy Tier® function. It also provides preferred practices for monitoring, maintaining, and troubleshooting. This book is intended for experienced storage, SAN, IBM FlashSystem, SAN Volume Controller, and IBM Storwize® administrators and technicians. Understanding this book requires advanced knowledge of these environments.

Who is the main stakeholder, with ultimate responsibility for driving Software performance testing forward? How will you know that the Software performance testing project has been successful? How do we keep improving Software performance testing? Do we all define Software performance testing in the same way? How to deal with Software performance testing Changes? Defining,

designing, creating, and implementing a process to solve a business challenge or meet a business objective is the most valuable role... In EVERY company, organization and department. Unless you are talking a one-time, single-use project within a business, there should be a process. Whether that process is managed and implemented by humans, AI, or a combination of the two, it needs to be designed by someone with a complex enough perspective to ask the right questions. Someone capable of asking the right questions and step back and say, 'What are we really trying to accomplish here? And is there a different way to look at it?' This Self-Assessment empowers people to do just that - whether their title is entrepreneur, manager, consultant, (Vice-)President, CxO etc... - they are the people who rule the future. They are the person who asks the right questions to make Software performance testing investments work better. This Software performance testing All-Inclusive Self-Assessment enables You to be that person. All the tools you need to an in-depth Software performance testing Self-Assessment. Featuring 710 new and updated case-based questions, organized into seven core areas of process design, this Self-Assessment will help you identify areas in which Software performance testing improvements can be made. In using the questions you will be better able to: - diagnose Software performance testing projects, initiatives, organizations, businesses and processes using accepted diagnostic standards and practices - implement evidence-based best practice strategies aligned with overall goals - integrate recent advances in Software performance testing and process design strategies into practice according to best practice guidelines Using a Self-Assessment tool known as the Software performance testing Scorecard, you will develop a clear picture of which Software performance testing areas need attention. Your purchase includes access details to the Software performance testing self-assessment dashboard download which gives you your dynamically prioritized projects-ready tool and shows your organization exactly what to do next. Your exclusive instant access details can be found in your book.

Learn how to: § Select the best ERP software for your organization § Choose the most effective wrap around software to enhance the performance of an existing ERP system § Align software selection with business goals and objectives § Budget for the software and the hidden costs involved in its implementation At times a daring, maddening, and even frightening process, finding and implementing a suitable software package is never an easy task. The cost of the software package is often a fraction of the overall expense. Unless carefully selected, a major software package implementation can consume a considerable amount of your organization's time and energy. An ill-informed purchase can cost your organization it's customers, dollars, and reputation. Maximizing Business Performance through Software Packages: Best Practices for Justification, Selection, and Implementation explores the business challenges involved in justifying, selecting, and implementing software packages. It contains practical advice and insights on how to select "good fitting" software packages, how to justify them in terms of their ability to enable business process change or improvement, and most importantly, how to implement them successfully. Selecting and implementing enterprise architecture technology software solutions involves a large expenditure across all the resources of an organization. The process has become increasingly complex as business functions have become increasingly integrated. Maximizing Business Performance through Software Packages: Best Practices for Justification, Selection, and Implementation provides a definitive source that will help you select the solutions that best fit your business needs.

This IBM® Redbooks® publication describes several of the preferred practices and describes the performance gains that can be achieved by implementing the IBM SAN Volume Controller powered by IBM Spectrum® Virtualize V8.4. These practices are based on field experience. This book highlights configuration guidelines and preferred practices for the storage area network (SAN) topology, clustered system, back-end storage, storage pools, and managed disks, volumes, Remote Copy services, and hosts. Then, it provides performance guidelines for IBM SAN Volume Controller, back-end storage, and applications. It explains how you can optimize disk performance with the IBM System Storage Easy Tier® function. It also provides preferred practices for monitoring, maintaining, and troubleshooting IBM SAN Volume Controller. This book is intended for experienced storage, SAN, and IBM SAN Volume Controller administrators and technicians. Understanding this book requires advanced knowledge of the IBM SAN Volume Controller, IBM FlashSystem, and SAN environments.

5th International Conference, SWQD 2013, Vienna, Austria, January 15-17, 2013, Proceedings

Best Practices for Justification, Selection, and Implementation

Site Reliability Engineering

Lessons from Successful Projects in the Top Companies

Software Configuration Management Patterns

Maximizing Business Performance through Software Packages

Software Measurement

In response to the increasing significance attached to supply chain management in both academic and professional areas, this text intends to build a bridge and highlight the relationship between various disciplines of SCM like demand planning, manufacturing planning, logistics planning, analytical IT management, global e-biz modeling, perfor-mance benchmarking etc. Primarily intended to address the typical and general syllabus requirements of postgraduate management programmes, and undergraduate and postgraduate engineering programmes, this book also caters to the needs of the industry professionals in the supply chain domain.

You'll find several books on basic or advanced MySQL performance, but nothing in between. That's because explaining MySQL performance without addressing its complexity is difficult. This practical book bridges the gap by teaching software engineers mid-level MySQL knowledge beyond the fundamentals, but well shy of deep-level internals required by database administrators (DBAs). Daniel Nichter shows you how to apply the best practices and techniques that directly affect MySQL performance. You'll learn how to improve performance by analyzing query execution, indexing for common SQL clauses and table joins, optimizing data access, and understanding the most important MySQL metrics. You'll also discover how replication, transactions, row locking, and the cloud influenceMySQL performance. Understand why query response time is the North Star of MySQL performance Learn query metrics in detail, including aggregation, reporting, and analysis See how to index effectively for common SQL clauses and table joins Explore the most important server metrics and what they reveal about performance Dive into transactions and row locking to gain deep, actionable insight Achieve remarkable MySQL performance at any scale

The importance of benchmarking in the service sector is well recognized as it helps in continuous improvement in products and work processes. Through benchmarking, companies have strived to implement best practices in order to remain competitive in the product- market in which they operate. However studies on benchmarking, particularly in the software development sector, have neglected using multiple variables and therefore have not been as comprehensive. Information Theory and Best Practices in the IT Industry fills this void by examining benchmarking in the business of software development and studying how it is affected by development process, application type, hardware platforms used, and many other variables. Information Theory and Best Practices in the IT Industry begins by examining practices of benchmarking productivity and critically appraises them. Next the book identifies different variables which affect productivity and variables that affect quality, developing useful equations that explaining their relationships. Finally these equations and findings are applied to case studies. Utilizing this book, practitioners can decide about what emphasis they should attach to different variables in their own companies, while seeking to optimize productivity and defect density.

Software Analysis and Optimization

Accelerate

State of the Art and Current Trends

Software Quality. Increasing Value in Software and Systems Development