

The Economics Of Software Quality

Thank you utterly much for downloading **The Economics Of Software Quality** .Maybe you have knowledge that, people have see numerous period for their favorite books gone this The Economics Of Software Quality , but end in the works in harmful downloads.

Rather than enjoying a good book considering a mug of coffee in the afternoon, instead they juggled when some harmful virus inside their computer. **The Economics Of Software Quality** is to hand in our digital library an online right of entry to it is set as public fittingly you can download it instantly. Our digital library saves in compound countries, allowing you to acquire the most less latency era to download any of our books with this one. Merely said, the The Economics Of Software Quality is universally compatible like any devices to read.

Estimating Software Costs - Capers Jones 2007-05-10

Deliver bug-free software projects on schedule and within budget Get a clear, complete understanding of how to estimate software costs, schedules, and quality using the real-world information contained in this comprehensive volume. Find out how to choose the correct hardware and software tools, develop an appraisal strategy, deploy tests and prototypes, and produce accurate software cost estimates. Plus, you'll get full coverage of cutting-edge estimating approaches using Java, object-oriented methods, and reusable components. Plan for and execute project-, phase-, and activity-level cost estimations Estimate regression, component, integration, and stress tests Compensate for inaccuracies in data collection, calculation, and analysis Assess software deliverables and data complexity Test design principles and operational characteristics using software prototyping Handle configuration change, research, quality control, and documentation costs "Capers Jones' work offers a unique contribution to the understanding of the economics of software production. It provides deep insights into why our advances in computing are not matched with corresponding improvements in the software that drives it. This book is absolutely required reading for an understanding of the limitations of our technological advances." --Paul A. Strassmann, former CIO of Xerox, the Department of Defense, and NASA

Thinking-Driven Testing - Adam Roman 2018-03-20

This book presents a new paradigm of software testing by emphasizing the role of critical thinking, system thinking and rationality as the most important skills for the tester. It thus approaches software testing from a different perspective than in past literature, as the vast majority of books describe testing in the context of specific tools, automation, documentation, particular test design techniques or test management. In addition, the book proposes a novel meta-approach for designing effective test strategies, which is based on recent advances in psychology, economics, system sciences and logic. Chapter 1 starts by introducing the fundamental ideas underlying software testing. Chapter 2 then describes meta-strategies in software testing, i.e. general approaches that can be adapted to many different situations that a software tester encounters. Next, Chapter 3 presents the concept of Thinking-Driven Testing (TDT). This approach utilizes the concepts discussed in the two previous chapters and introduces the main ideas that underlie a reasonable and optimal approach to software testing. Chapter 4 builds on this basis and proposes a specific approach to testing, called TQED, that makes it possible to increase creativity in the context of delivering effective, optimal test ideas. Chapter 5 provides an overview of different types of testing techniques in order to understand the fundamental concepts of test design, while Chapter 6 details various pitfalls a tester may encounter and that can originate from a wide range of testing process areas. Lastly, Chapter 7 puts all this into practice, as it contains several exercises that will help testers develop a number of crucial skills: logical thinking and reasoning, thinking out of the box, creativity, counting and estimating, and analytical thinking. By promoting critical, rational and creative thinking, this book invites readers to re-examine common assumptions regarding software testing and shows them how to become professional testers who bring added value to their company.

High Quality Low Cost Software Inspections - Ronald A. Radice 2004-02-01

Written with every software engineer's question in mind--"How much do inspections pay off?--this book demonstrates how to get the most value while controlling costs for investment and inspections. Covered are

the economics and ROI of inspections as well as defect prevention and controlling the cost of quality. Written for managers interested in successfully implementing inspection programs and using the data, this book can also be used by engineers and technical writers who want to learn how to maximize software quality and minimize costs as well as agents of change interested in introducing a method that has significant quality and productivity potential. Also described are how to develop and use checklists, how to apply inspections to other life cycle activities and work products, how to read documents for successful inspections, and how to maximize quality, cost, and cycle time objectives for an organization.

Software Testing and Quality Assurance - Kshirasagar Naik 2011-09-23

A superior primer on software testing and quality assurance, from integration to execution and automation This important new work fills the pressing need for a user-friendly text that aims to provide software engineers, software quality professionals, software developers, and students with the fundamental developments in testing theory and common testing practices. Software Testing and Quality Assurance: Theory and Practice equips readers with a solid understanding of: Practices that support the production of quality software Software testing techniques Life-cycle models for requirements, defects, test cases, and test results Process models for units, integration, system, and acceptance testing How to build test teams, including recruiting and retaining test engineers Quality Models, Capability Maturity Model, Testing Maturity Model, and Test Process Improvement Model Expertly balancing theory with practice, and complemented with an abundance of pedagogical tools, including test questions, examples, teaching suggestions, and chapter summaries, this book is a valuable, self-contained tool for professionals and an ideal introductory text for courses in software testing, quality assurance, and software engineering.

Applied Software Measurement - Capers Jones 2008-05-11

Effectively forecast, manage, and control software across the entire project lifecycle Accurately size, estimate, and administer software projects with real-world guidance from an industry expert. Fully updated to cover the latest tools and techniques, Applied Software Measurement, Third Edition details how to deploy a cost-effective and pragmatic analysis strategy. You will learn how to use function points and baselines, implement benchmarks and tracking systems, and perform efficiency tests. Full coverage of the latest regulations, metrics, and standards is included. Measure performance at the requirements, coding, testing, and installation phases Set function points for efficiency, cost, market share, and customer satisfaction Analyze quality and productivity using assessments, benchmarks, and baselines Design and manage project cost, defect, and quality tracking systems Use object-oriented, reusable component, Agile, CMM, and XP methods Assess defect removal efficiency using unit tests and multistage test suites

The Art of Software Testing - Glenford J. Myers 2004-07-22

This long-awaited revision of a bestseller provides a practical discussion of the nature and aims of software testing. You'll find the latest methodologies for the design of effective test cases, including information on psychological and economic principles, managerial aspects, test tools, high-order testing, code inspections, and debugging. Accessible, comprehensive, and always practical, this edition provides the key information you need to test successfully, whether a novice or a working programmer. Buy your copy today and end up with fewer bugs tomorrow.

Software Testing - Ali Mili 2015-06-15

Explores and identifies the main issues, concepts, principles and evolution of software testing, including software quality engineering and testing concepts, test data generation, test deployment analysis, and software test management This book examines the principles, concepts, and processes that are fundamental to the software testing function. This book is divided into five broad parts. Part I introduces software testing in the broader context of software engineering and explores the qualities that testing aims to achieve or ascertain, as well as the lifecycle of software testing. Part II covers mathematical foundations of software testing, which include software specification, program correctness and verification, concepts of software dependability, and a software testing taxonomy. Part III discusses test data generation, specifically, functional criteria and structural criteria. Test oracle design, test driver design, and test outcome analysis is covered in Part IV. Finally, Part V surveys managerial aspects of software testing, including software metrics, software testing tools, and software product line testing. Presents software testing, not as an isolated technique, but as part of an integrated discipline of software verification and validation Proposes program testing and program correctness verification within the same mathematical model, making it possible to deploy the two techniques in concert, by virtue of the law of diminishing returns Defines the concept of a software fault, and the related concept of relative correctness, and shows how relative correctness can be used to characterize monotonic fault removal Presents the activity of software testing as a goal oriented activity, and explores how the conduct of the test depends on the selected goal Covers all phases of the software testing lifecycle, including test data generation, test oracle design, test driver design, and test outcome analysis Software Testing: Concepts and Operations is a great resource for software quality and software engineering students because it presents them with fundamentals that help them to prepare for their ever evolving discipline.

The Technical and Social History of Software Engineering - Capers Jones 2014

Pioneering software engineer Capers Jones has written the first and only definitive history of the entire software engineering industry. Drawing on his extraordinary vantage point as a leading practitioner for several decades, Jones reviews the entire history of IT and software engineering, assesses its impact on society, and previews its future. One decade at a time, Jones assesses emerging trends and companies, winners and losers, new technologies, methods, tools, languages, productivity/quality benchmarks, challenges, risks, professional societies, and more. He quantifies both beneficial and harmful software inventions; accurately estimates the size of both the US and global software industries; and takes on "unexplained mysteries" such as why and how programming languages gain and lose popularity.

The Economics of Software Quality, Video Enhanced Edition - Capers Jones 2011-12-31

This is the video enhanced eBook version of the printed book. It contains 55 minutes of video conversations & tips from the industry's leading software management consultant, Capers Jones. Important note: The audio and video content included with this enhanced eBook can be viewed only using iBooks on an iPad, iPhone, or iPod touch. Due to the incredibly rich media included in your enhanced eBook, you may experience longer than usual download times. Please be patient while your product is delivered. "Whether consulting, working on projects, or teaching, whenever I need credible, detailed, relevant metrics and insights into the current capabilities and performance of the software engineering profession, I always turn first to Capers Jones' work. In this important new book, he and Olivier Bonsignour make the hard-headed, bottom-line, economic case, with facts and data, about why software quality is so important. I know I'll turn to this excellent reference again and again." —Rex Black, President, RBCS Poor quality continues to bedevil large-scale development projects, but few software leaders and practitioners know how to measure quality, select quality best practices, or cost-justify their usage. In *The Economics of Software Quality*, leading software quality experts Capers Jones and Olivier Bonsignour show how to systematically measure the economic impact of quality and how to use this information to deliver far more business value. Using empirical data from hundreds of software organizations, Jones and Bonsignour show how integrated inspection, structural quality measurement, static analysis, and testing can achieve defect removal rates exceeding 95 percent. They offer innovative guidance for predicting and measuring defects and quality; choosing defect prevention, pre-test defect removal, and testing methods; and optimizing post-release defect reporting and repair. This book will help you Move beyond functional quality to quantify non-functional and structural quality Prove that improved software quality translates into strongly positive ROI

and greatly reduced TCO Drive better results from current investments in Quality Assurance and Testing Use quality improvement techniques to stay on schedule and on budget Avoid "hazardous" metrics that lead to poor decisions

Taking Economics Seriously - Dean Baker 2010-04-02

A leading economist's exploration of what our economic arrangements might look like if we applied basic principles without ideological blinders. There is nothing wrong with economics, Dean Baker contends, but economists routinely ignore their own principles when it comes to economic policy. What would policy look like if we took basic principles of mainstream economics seriously and applied them consistently? In the debate over regulation, for example, Baker—one of the few economists who predicted the meltdown of fall 2008—points out that ideological blinders have obscured the fact there is no "free market" to protect. Modern markets are highly regulated, although intrusive regulations such as copyright and patents are rarely viewed as regulatory devices. If we admit the extent to which the economy is and will be regulated, we have many more options in designing policy and deciding who benefits from it. On health care reform, Baker complains that economists ignore another basic idea: marginal cost pricing. Unlike all other industries, medical services are priced extraordinarily high, far above the cost of production, yet that discrepancy is rarely addressed in the debate about health care reform. What if we applied marginal cost pricing—making doctors' wages competitive and charging less for prescription drugs and tests such as MRIs? *Taking Economics Seriously* offers an alternative Econ 101. It introduces economic principles and thinks through what we might gain if we free ourselves from ideological blinders and get back to basics in the most troubled parts of our economy.

Guide to the Software Engineering Body of Knowledge (Swebok(r)) - IEEE Computer Society 2014

In the *Guide to the Software Engineering Body of Knowledge (SWEBOK(R) Guide)*, the IEEE Computer Society establishes a baseline for the body of knowledge for the field of software engineering, and the work supports the Society's responsibility to promote the advancement of both theory and practice in this field. It should be noted that the Guide does not purport to define the body of knowledge but rather to serve as a compendium and guide to the knowledge that has been developing and evolving over the past four decades. Now in Version 3.0, the Guide's 15 knowledge areas summarize generally accepted topics and list references for detailed information. The editors for Version 3.0 of the SWEBOK(R) Guide are Pierre Bourque (Ecole de technologie superieure (ETS), Universite du Quebec) and Richard E. (Dick) Fairley (Software and Systems Engineering Associates (S2EA)).

The Economics of Software Quality Assurance - Tarek K Abdel-Hamid 2022-10-27

This work has been selected by scholars as being culturally important, and is part of the knowledge base of civilization as we know it. This work is in the "public domain in the United States of America, and possibly other nations. Within the United States, you may freely copy and distribute this work, as no entity (individual or corporate) has a copyright on the body of the work. Scholars believe, and we concur, that this work is important enough to be preserved, reproduced, and made generally available to the public. We appreciate your support of the preservation process, and thank you for being an important part of keeping this knowledge alive and relevant.

Introduction to the Personal Software Process(sm) - Watts S. Humphrey 1996-12-20

This newest book from Watts Humphrey is a hands-on introduction to basic disciplines of software engineering. Designed as a workbook companion to any introductory programming or software-engineering text, Humphrey provides here the practical means to integrate his highly regarded Personal Software Process (PSP) into college and university curricula. The book may also be adapted for use in industrial training or for self-improvement by practicing software engineers. Applying the book's exercises to their course assignments, students learn both to manage their time effectively and to monitor the quality of their work, good practices they will need to be successful in their future careers. The book is supported by its own electronic supplement, which includes spreadsheets for data entry and analysis. A complete instructor's package is also available. By mastering PSP techniques early in their studies, students can avoid--or overcome--the popular "hacker" ethic that leads to so many bad habits. Employers will appreciate new hires prepared to do competent professional work without, as now is common, expensive retraining and years of experience.

The Economics of Software Quality - Capers Jones 2012

Poor quality continues to bedevil large-scale development projects, but few software leaders and practitioners know how to measure quality, select quality best practices, or cost-justify their usage. In *The Economics of Software Quality*, leading software quality experts Capers Jones and Jitendra Subramanyam show how to systematically measure the economic impact of quality and how to use this information to deliver far more business value. Using empirical data from hundreds of software organizations, Jones and Subramanyam show how integrated inspection, static analysis, and testing can achieve defect removal rates exceeding 95 percent. They offer innovative guidance for predicting and measuring defects and quality; choosing defect prevention, pre-test defect removal, and testing methods; and optimizing post-release defect reporting and repair. This book will help you Prove that improved software quality translates into strongly positive ROI and greatly reduced TCO Drive better results from current investments in debugging and prevention Use quality techniques to stay on schedule and on budget Avoid "hazardous" metrics that lead to poor decisions Important note: The audio and video content included with this enhanced eBook can be viewed only using iBooks on an iPad, iPhone, or iPod touch.

Software Engineering Economics - Barry W. Boehm 1981

Software Engineering Economics is an invaluable guide to determining software costs, applying the fundamental concepts of microeconomics to software engineering, and utilizing economic analysis in software engineering decision making.

Software, Growth, and the Future of the U.S Economy - National Research Council 2006-05-06

Starting in the mid 1990s, the United States economy experienced an unprecedented upsurge in economic productivity. Rapid technological change in communications, computing, and information management continue to promise further gains in productivity, a phenomenon often referred to as the New Economy. To better understand this phenomenon, the National Academies Board on Science, Technology, and Economic Policy (STEP) has convened a series of workshops and commissioned papers on Measuring and Sustaining the New Economy. This major workshop, entitled *Software, Growth, and the Future of the U.S. Economy*, convened academic experts and industry representatives from leading companies such as Google and General Motors to participate in a high-level discussion of the role of software and its importance to U.S. productivity growth; how software is made and why it is unique; the measurement of software in national and business accounts; the implications of the movement of the U.S. software industry offshore; and related policy issues.

The Economics of Information Systems and Software - Richard Veryard 2014-05-15

The Economics of Information Systems and Software focuses on the economic aspects of information systems and software, including advertising, evaluation of information systems, and software maintenance. The book first elaborates on value and values, software business, and scientific information as an economic category. Discussions focus on information products and information services, special economic properties of information, culture and convergence, hardware and software products, materiality and consumption, technological progress, and software flexibility. The text then takes a look at advertising to finance software, perspectives on East-West relations in economics and information, and evaluation of information systems. Topics include research on information systems, knowledge on Eastern European information services, GDR information institutes, local databases, GDR databases, CMEA directions, and theoretical propositions. The manuscript reviews software reuse, software methodology in the harsh light of economics, quantitative aspects of software maintenance management, and calibrating a software cost-estimation model. Concerns cover the need for calibration, measuring maintainability, prognosis of maintenance effort, object-oriented programming, metaprogramming, and software quality and reuse. The text is a dependable reference for computer science experts and researchers wanting to explore further the economics of information systems and software.

Quality Software Project Management - Robert T. Futrell 2002

Drawing on best practices identified at the Software Quality Institute and embodied in bodies of knowledge from the Project Management Institute, the American Society of Quality, IEEE, and the Software Engineering Institute, *Quality Software Project Management* teaches 34 critical skills that allow any manager to minimize costs, risks, and time-to-market. Written by leading practitioners Robert T. Futrell,

Donald F. Shafer, and Linda I. Shafer, it addresses the entire project lifecycle, covering process, project, and people. It contains extensive practical resources-including downloadable checklists, templates, and forms.

Software Engineering - Shari Lawrence Pfleeger 1991

Occupational Outlook Handbook - United States. Bureau of Labor Statistics 1976

Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products - Mora, Manuel 2020-07-10

The highly dynamic world of information technology service management stresses the benefits of the quick and correct implementation of IT services. A disciplined approach relies on a separate set of assumptions and principles as an agile approach, both of which have complicated implementation processes as well as copious benefits. Combining these two approaches to enhance the effectiveness of each, while difficult, can yield exceptional dividends. *Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products* is an essential publication that focuses on clarifying theoretical foundations of balanced design methods with conceptual frameworks and empirical cases. Highlighting a broad range of topics including business trends, IT service, and software development, this book is ideally designed for software engineers, software developers, programmers, information technology professionals, researchers, academicians, and students.

The Economic Impacts of Inadequate Infrastructure for Software Testing - Gregory Tassej 2002-09-01

In 2000, total sales of software in the U.S. reached \$180 billion. Reducing the cost of software development and improving software quality are important objectives of the U.S. software industry. However, the complexity of the underlying software needed to support the U.S.'s computerized economy is increasing at an alarming rate. Software nonperformance and failure are expensive, but it is difficult to define and measure software quality. The objective of this study is to investigate the economic impact of an inadequate infrastructure for software testing in the U.S. This study was undertaken as part of joint planning between NIST and industry to help identify and assess technical needs that would improve the industry's software testing capabilities. Illustrated.

How to Reduce the Cost of Software Testing - Matthew Heusser 2018-09-03

Plenty of software testing books tell you how to test well; this one tells you how to do it while decreasing your testing budget. A series of essays written by some of the leading minds in software testing, *How to Reduce the Cost of Software Testing* provides tips, tactics, and techniques to help readers accelerate the testing process, improve the performance of the test teams, and lower costs. The distinguished team of contributors—that includes corporate test leaders, best paper authors, and keynote speakers from leading software testing conferences—supply concrete suggestions on how to find cost savings without sacrificing outcome. Detailing strategies that testers can immediately put to use to reduce costs, the book explains how to make testing nimble, how to remove bottlenecks in the testing process, and how to locate and track defects efficiently and effectively. Written in language accessible to non-technical executives, as well as those doing the testing, the book considers the latest advances in test automation, ideology, and technology. Rather than present the perspective of one or two experts in software testing, it supplies the wide-ranging perspectives of a team of experts to help ensure your team can deliver a completed test cycle in less time, with more confidence, and reduced costs.

Economics-Driven Software Architecture - Ivan Mistrik 2014-06-03

Economics-driven Software Architecture presents a guide for engineers and architects who need to understand the economic impact of architecture design decisions: the long term and strategic viability, cost-effectiveness, and sustainability of applications and systems. Economics-driven software development can increase quality, productivity, and profitability, but comprehensive knowledge is needed to understand the architectural challenges involved in dealing with the development of large, architecturally challenging systems in an economic way. This book covers how to apply economic considerations during the software architecting activities of a project. Architecture-centric approaches to development and systematic evolution, where managing complexity, cost reduction, risk mitigation, evolvability, strategic planning and

long-term value creation are among the major drivers for adopting such approaches. It assists the objective assessment of the lifetime costs and benefits of evolving systems, and the identification of legacy situations, where architecture or a component is indispensable but can no longer be evolved to meet changing needs at economic cost. Such consideration will form the scientific foundation for reasoning about the economics of nonfunctional requirements in the context of architectures and architecting. Familiarizes readers with essential considerations in economic-informed and value-driven software design and analysis Introduces techniques for making value-based software architecting decisions Provides readers a better understanding of the methods of economics-driven architecting

Principles of Quality Costs, Fourth Edition - Douglas C. Wood 2012-12-28

The last decade has seen wide changes in how quality standards are applied in industry. We now have two functions: quality assurance and process improvement. Quality assurance focuses primarily on product quality, while process improvement focuses on process quality; the principles of quality cost support both. The purpose of this book remains the same as the third edition: to provide a basic understanding of the principles of quality cost. Using this book, organizations can develop and implement a quality cost system to fit their needs. Used as an adjunct to overall financial management, these principles will help maintain vital quality improvement programs over extended timeframes. This fourth edition now includes information on the quality cost systems involved with the education, service, banking, and software development industries. You'll also find new material on ISO 9001, cost systems in small businesses, and activity based costing. Additional information on team-based problem-solving, customer satisfaction, and the costs involved with the defense industry are also offered.

Maximizing ROI on Software Development - Vijay Sikka 2004-10-28

Maximizing ROI on Software Development explains how to execute best quality software development and testing while maximizing business value. It discusses Applied ROI in the context of methodologies such as Agile and Extreme Programming, and traditional methodologies including Six Sigma, the Capability Maturity Model® (CMM®), Total Cost of Ownership (TCO), and Product Line Models (PLM). The text discusses what is important in global terms and details how best to choose teams and partners, including outsourcers, and how to employ the latest tools and technologies. It provides models, metrics, and detailed case studies to improve current and future development projects, whether in house or outsourced, near shore or off-shore. The book offers perspectives on how quality improvement through software quality assurance (SQA) testing, planning, and execution is a powerful and effective route toward maximizing return on investment. Divided into seven chapters, this friendly and informative guide can be read quickly, then used as a reliable reference by team leaders and members. It begins by reviewing software development, tools, and methodologies, followed by an examination of how development, maintenance, and integration have become more complex and will continue to do so. The book discusses best practices for managing this complexity and explores the business case for maximizing ROI. The text then provides a comprehensive analysis of ROI from several perspectives, covering nomenclature, project success and failure, mathematics, processes, work products, and techniques. It details how to make global teams successful and how to evaluate Applied ROI implementation, and it includes case studies for wireless, enterprise, and CRM systems.

What Drives Quality - Ben Linders 2017-09-30

With plenty of ideas, suggestions, and practical cases on software quality, this book will help you to improve the quality of your software and to deliver high-quality products to your users and satisfy the needs of your customers and stakeholders. Many methods for product quality improvement start by investigating the problems, and then work their way back to the point where the problem started. For instance audits and root cause analysis work this way. But what if you could prevent problems from happening, by building an understanding what drives quality, thus enabling to take action before problems actually occur? What Drives Quality explores how quality plays a role in all of the software development activities. It takes a deep dive into quality by listing the relevant factors of development and management activities that drive the quality of software products. It provides a lean approach to quality by analyzing the full development chain from customer requests to delivering products to users. I'm aiming this book at software developers and testers, architects, product owners and managers, agile coaches, Scrum masters, project managers, and

operational and senior managers who consider quality to be important. A book on quality should be practical. It should help you, the reader of this book, to improve the quality of your software and deliver better products. It should inspire you and give you energy to persevere on your quality journey. What drives quality tries to do just that, and more. This book is based on my experience as a developer, tester, team leader, project manager, quality manager, process manager, consultant, coach, trainer, and adviser in Agile, Lean, Quality and Continuous Improvement. It takes a deep dive into quality with views from different perspectives and provides ideas, suggestions, practices, and experiences that will help you to improve quality of the products that your organization is delivering. This book views software quality from an engineering, management, and social perspective. It explores the interaction between all involved in delivering high-quality software to users and provides ideas to do it quicker and at lower costs.

The Economics of Software Quality - Capers Jones 2011-07-19

"Whether consulting, working on projects, or teaching, whenever I need credible, detailed, relevant metrics and insights into the current capabilities and performance of the software engineering profession, I always turn first to Capers Jones' work. In this important new book, he and Olivier Bonsignour make the hard-headed, bottom-line, economic case, with facts and data, about why software quality is so important. I know I'll turn to this excellent reference again and again." —Rex Black, President, RBCS Poor quality continues to bedevil large-scale development projects, but few software leaders and practitioners know how to measure quality, select quality best practices, or cost-justify their usage. In *The Economics of Software Quality*, leading software quality experts Capers Jones and Olivier Bonsignour show how to systematically measure the economic impact of quality and how to use this information to deliver far more business value. Using empirical data from hundreds of software organizations, Jones and Bonsignour show how integrated inspection, structural quality measurement, static analysis, and testing can achieve defect removal rates exceeding 95 percent. They offer innovative guidance for predicting and measuring defects and quality; choosing defect prevention, pre-test defect removal, and testing methods; and optimizing post-release defect reporting and repair. This book will help you Move beyond functional quality to quantify non-functional and structural quality Prove that improved software quality translates into strongly positive ROI and greatly reduced TCO Drive better results from current investments in Quality Assurance and Testing Use quality improvement techniques to stay on schedule and on budget Avoid "hazardous" metrics that lead to poor decisions

Happy About Global Software Test Automation - Hung Quoc Nguyen 2006

This book addresses the fundamental issue of software testing and helps the reader understand the high-level elements necessary to better execute software test automation and outsourcing initiatives.

Reliability and Statistical Computing - Hoang Pham 2020-03-28

This book presents the latest developments in both qualitative and quantitative computational methods for reliability and statistics, as well as their applications. Consisting of contributions from active researchers and experienced practitioners in the field, it fills the gap between theory and practice and explores new research challenges in reliability and statistical computing. The book consists of 18 chapters. It covers (1) modeling in and methods for reliability computing, with chapters dedicated to predicted reliability modeling, optimal maintenance models, and mechanical reliability and safety analysis; (2) statistical computing methods, including machine learning techniques and deep learning approaches for sentiment analysis and recommendation systems; and (3) applications and case studies, such as modeling innovation paths of European firms, aircraft components, bus safety analysis, performance prediction in textile finishing processes, and movie recommendation systems. Given its scope, the book will appeal to postgraduates, researchers, professors, scientists, and practitioners in a range of fields, including reliability engineering and management, maintenance engineering, quality management, statistics, computer science and engineering, mechanical engineering, business analytics, and data science.

The Economics of Software Quality Assurance - Tarek K. Abdel-Hamid 2017-03-11

Excerpt from *The Economics of Software Quality Assurance: A System Dynamics Based Simulation Approach* The Human Resource Management Subsystem captures the hiring, training, assimilation, and transfer of the project's human resource. Such actions are not carried out in vacuum, but, as Figure 1 suggests, they are affected by the other subsystems. For example, the project's hiring rate is a function of

the workforce level needed to complete the project on a certain planned completion date. Similarly, what workforce is available has direct. About the Publisher Forgotten Books publishes hundreds of thousands of rare and classic books. Find more at www.forgottenbooks.com This book is a reproduction of an important historical work. Forgotten Books uses state-of-the-art technology to digitally reconstruct the work, preserving the original format whilst repairing imperfections present in the aged copy. In rare cases, an imperfection in the original, such as a blemish or missing page, may be replicated in our edition. We do, however, repair the vast majority of imperfections successfully; any imperfections that remain are intentionally left to preserve the state of such historical works.

Green in Software Engineering - Coral Calero 2015-04-03

This is the first book that presents a comprehensive overview of sustainability aspects in software engineering. Its format follows the structure of the SWEBOK and covers the key areas involved in the incorporation of green aspects in software engineering, encompassing topics from requirement elicitation to quality assurance and maintenance, while also considering professional practices and economic aspects. The book consists of thirteen chapters, which are structured in five parts. First the "Introduction" gives an overview of the primary general concepts related to Green IT, discussing what Green in Software Engineering is and how it differs from Green by Software Engineering. Next "Environments, Processes and Construction" presents green software development environments, green software engineering processes and green software construction in general. The third part, "Economic and Other Qualities," details models for measuring how well software supports green software engineering techniques and for performing trade-off analyses between alternative green practices from an economic perspective. "Software Development Process" then details techniques for incorporating green aspects at various stages of software development, including requirements engineering, design, testing, and maintenance. In closing, "Practical Issues" addresses the repercussions of green software engineering on decision-making, stakeholder participation and innovation management. The audience for this book includes software engineering researchers in academia and industry seeking to understand the challenges and impact of green aspects in software engineering, as well as practitioners interested in learning about the state of the art in Green in Software Engineering.

The Certified Software Quality Engineer Handbook - Linda Westfall 2016-09-23

A comprehensive reference manual to the Certified Software Quality Engineer Body of Knowledge and study guide for the CSQE exam.

The Future of Software Quality Assurance - Stephan Goericke 2019-11-19

This open access book, published to mark the 15th anniversary of the International Software Quality Institute (ISQI), is intended to raise the profile of software testers and their profession. It gathers contributions by respected software testing experts in order to highlight the state of the art as well as future challenges and trends. In addition, it covers current and emerging technologies like test automation, DevOps, and artificial intelligence methodologies used for software testing, before taking a look into the future. The contributing authors answer questions like: "How is the profession of tester currently changing? What should testers be prepared for in the years to come, and what skills will the next generation need? What opportunities are available for further training today? What will testing look like in an agile world that is user-centered and fast-paced? What tasks will remain for testers once the most important processes are automated?" ISQI has been focused on the education and certification of software testers for fifteen years now, and in the process has contributed to improving the quality of software in many areas. The papers gathered here clearly reflect the numerous ways in which software quality assurance can play a critical role in various areas. Accordingly, the book will be of interest to both professional software testers and managers working in software testing or software quality assurance.

Software Product Quality Control - Stefan Wagner 2013-07-25

Quality is not a fixed or universal property of software; it depends on the context and goals of its stakeholders. Hence, when you want to develop a high-quality software system, the first step must be a clear and precise specification of quality. Yet even if you get it right and complete, you can be sure that it will become invalid over time. So the only solution is continuous quality control: the steady and explicit evaluation of a product's properties with respect to its updated quality goals. This book guides you in

setting up and running continuous quality control in your environment. Starting with a general introduction on the notion of quality, it elaborates what the differences between process and product quality are and provides definitions for quality-related terms often used without the required level of precision. On this basis, the work then discusses quality models as the foundation of quality control, explaining how to plan desired product qualities and how to ensure they are delivered throughout the entire lifecycle. Next it presents the main concepts and techniques of continuous quality control, discussing the quality control loop and its main techniques such as reviews or testing. In addition to sample scenarios in all chapters, the book is rounded out by a dedicated chapter highlighting several applications of different subsets of the presented quality control techniques in an industrial setting. The book is primarily intended for practitioners working in software engineering or quality assurance, who will benefit by learning how to improve their current processes, how to plan for quality, and how to apply state-of-the-art quality control techniques. Students and lecturers in computer science and specializing in software engineering will also profit from this book, which they can use in practice-oriented courses on software quality, software maintenance and quality assurance.

Software Quality Assurance - Abu Sayed Mahfuz 2016-04-27

Software Quality Assurance: Integrating Testing, Security, and Audit focuses on the importance of software quality and security. It defines various types of testing, recognizes factors that propose value to software quality, and provides theoretical and real-world scenarios that offer value and contribute quality to projects and applications. The p

Effective Methods for Software Testing, CafeScribe - William E. Perry 2007-03-31

Written by the founder and executive director of the Quality Assurance Institute, which sponsors the most widely accepted certification program for software testing Software testing is a weak spot for most developers, and many have no system in place to find and correct defects quickly and efficiently This comprehensive resource provides step-by-step guidelines, checklists, and templates for each testing activity, as well as a self-assessment that helps readers identify the sections of the book that respond to their individual needs Covers the latest regulatory developments affecting software testing, including Sarbanes-Oxley Section 404, and provides guidelines for agile testing and testing for security, internal controls, and data warehouses CD-ROM with all checklists and templates saves testers countless hours of developing their own test documentation Note: CD-ROM/DVD and other supplementary materials are not included as part of eBook file.

Fuzzing for Software Security Testing and Quality Assurance - Ari Takanen 2008

Learn the code cracker's malicious mindset, so you can find worn-size holes in the software you are designing, testing, and building. Fuzzing for Software Security Testing and Quality Assurance takes a weapon from the black-hat arsenal to give you a powerful new tool to build secure, high-quality software. This practical resource helps you add extra protection without adding expense or time to already tight schedules and budgets. The book shows you how to make fuzzing a standard practice that integrates seamlessly with all development activities. This comprehensive reference goes through each phase of software development and points out where testing and auditing can tighten security. It surveys all popular commercial fuzzing tools and explains how to select the right one for a software development project. The book also identifies those cases where commercial tools fall short and when there is a need for building your own fuzzing tools.

Software Quality Assurance - Ivan Mistrik 2015-10-12

Software Quality Assurance in Large Scale and Complex Software-intensive Systems presents novel and high-quality research related approaches that relate the quality of software architecture to system requirements, system architecture and enterprise-architecture, or software testing. Modern software has become complex and adaptable due to the emergence of globalization and new software technologies, devices and networks. These changes challenge both traditional software quality assurance techniques and software engineers to ensure software quality when building today (and tomorrow's) adaptive, context-sensitive, and highly diverse applications. This edited volume presents state of the art techniques, methodologies, tools, best practices and guidelines for software quality assurance and offers guidance for future software engineering research and practice. Each contributed chapter considers the practical

application of the topic through case studies, experiments, empirical validation, or systematic comparisons with other approaches already in practice. Topics of interest include, but are not limited, to: quality attributes of system/software architectures; aligning enterprise, system, and software architecture from the point of view of total quality; design decisions and their influence on the quality of system/software architecture; methods and processes for evaluating architecture quality; quality assessment of legacy systems and third party applications; lessons learned and empirical validation of theories and frameworks on architectural quality; empirical validation and testing for assessing architecture quality. Focused on quality assurance at all levels of software design and development Covers domain-specific software quality assurance issues e.g. for cloud, mobile, security, context-sensitive, mash-up and autonomic systems Explains likely trade-offs from design decisions in the context of complex software system engineering and quality assurance Includes practical case studies of software quality assurance for complex, adaptive and context-critical systems

Software Quality Assurance - Neil Walkinshaw 2017-07-24

This textbook offers undergraduate students an introduction to the main principles and some of the most popular techniques that constitute 'software quality assurance'. The book seeks to engage students by placing an emphasis on the underlying foundations of modern quality-assurance techniques, using these to

highlight why techniques work, as opposed to merely focussing on how they work. In doing so it provides readers with a comprehensive understanding of where software quality fits into the development lifecycle (spoiler: everywhere), and what the key quality assurance activities are. The book focuses on quality assurance in a way that typical, more generic software engineering reference books do not. It is structured so that it can (and should) be read from cover to cover throughout the course of a typical university module. Specifically, it is Concise: it is small enough to be readable in its entirety over the course of a typical software engineering module. Explanatory: topics are discussed not merely in terms of what they are, but also why they are the way they are - what events, technologies, and individuals or organisations helped to shape them into what they are now. Applied: topics are covered with a view to giving the reader a good idea of how they can be applied in practice, and by pointing, where possible, to evidence of their efficacy. The book starts from some of the most general notions (e.g. quality and development process), and gradually homes-in on the more specific activities, assuming knowledge of the basic notions established in prior chapters. Each chapter concludes with a "Key Points" section, summarising the main issues that have been covered in the chapter. Throughout the book there are exercises that serve to remind readers of relevant parts in the book that have been covered previously, and give them the opportunity to reflect on a particular topic and refer to related references.