

Testing Computer Software Second Edition

This celebrated primer presents an introduction to all of the key ingredients in understanding computerized adaptive testing technology, test development, statistics, and mental test theory. Based on years of research, this accessible book educates the novice and serves as a compendium of state-of-the-art information for professionals interested in computerized testing in the areas of education, psychology, and other related social sciences. A hypothetical test taken as a prelude to employment is used as a common example throughout to highlight this book's most important features and problems. Changes in the new edition include:
* a completely rewritten chapter 2 on the system considerations needed for modern computerized adaptive testing;
* a revised chapter 4 to include the latest in methodology surrounding online calibration and in the modeling of testlets; and
* a new chapter 10 with helpful information on how test items are really selected, usage patterns, how usage patterns influence the number of new items required, and tools for managing item pools.

This book is designed for use as an introductory software engineering course or as a reference for programmers. Up-to-date text uses both theory applications to design reliable, error-free software. Includes a companion CD-ROM with source code third-party software engineering applications.

This book teaches test managers what they need to know to achieve advanced skills in test estimation, test planning, test monitoring, and test control. Readers will learn how to define the overall testing goals and strategies for the systems being tested. This hands-on, exercise-rich book provides experience with planning, scheduling, and tracking these tasks. You'll be able to describe and organize the necessary activities as well as learn to select, acquire, and assign adequate resources for testing tasks. You'll learn how to form, organize, and lead testing teams, and master the organizing of communication among the members of the testing teams, and between the testing teams and all the other stakeholders. Additionally, you'll learn how to justify decisions and provide adequate reporting information where applicable. With over thirty years of software and systems engineering experience, author Rex Black is President of RBCS, is a leader in software, hardware, and systems testing, and is the most prolific author practicing in the field of software testing today. He has published a dozen books on testing that have sold tens of thousands of copies worldwide. He is past president of the International Software Testing Qualifications Board (ISTQB) and a director of the American Software Testing Qualifications Board (ASTQB). This book will help you prepare for the ISTQB Advanced Test Manager exam. Included are sample exam questions, at the appropriate level of difficulty, for most of the learning objectives covered by the ISTQB Advanced Level Syllabus. The ISTQB certification program is the leading software tester certification program in the world. With about 300,000 certificate holders and a global reach over 50 countries, you can be confident in the value and international stature that the Advanced Test Manager certificate can offer you. This second edition has been thoroughly updated to reflect the new ISTQB Advanced Test Manager 2012 Syllabus, and the latest ISTQB Glossary. This edition reflects Rex Black's unique insights into these changes, as he was one of the main participants in the ISTQB Advanced Level Working Group.

Extensively class-tested, this textbook takes an innovative approach to software testing: it defines testing as the process of applying a few well-defined, general-purpose test criteria to a structure or model of the software. It incorporates the latest innovations in testing, including techniques to test modern types of software such as OO, web applications, and embedded software. The book contains numerous examples throughout.

An instructor's solution manual, PowerPoint slides, sample syllabi, additional examples and updates, testing tools for students, and example software programs in Java are available on an extensive website.

Code Complete

A Practitioner's Guide to Software Test Design

Effective Methods for Software Testing

Handbook of Test Development

The Elements of Computing Systems

How to Find and Fix the Killer Software Bugs that Evade Conventional Testing In Exploratory Software Testing, renowned software testing expert James Whittaker reveals the real causes of today's most serious, well-hidden software bugs--and introduces powerful new "exploratory" techniques for finding and correcting them. Drawing on nearly two decades of experience working at the cutting edge of testing with Google, Microsoft, and other top software organizations, Whittaker introduces innovative new processes for manual testing that are repeatable, prescriptive, teachable, and extremely effective. Whittaker defines both in-the-small techniques for individual testers and in-the-large techniques to supercharge test teams. He also introduces a hybrid strategy for injecting exploratory concepts into traditional scripted testing. You'll learn when to use each, and how to use them all successfully. Concise, entertaining, and actionable, this book introduces robust techniques that have been used extensively by real testers on shipping software, illuminating their actual experiences with these techniques, and the results they've achieved. Writing for testers, QA specialists, developers, program managers, and architects alike, Whittaker answers crucial questions such as:
• Why do some bugs remain invisible to automated testing--and how can I uncover them?
• What techniques will help me consistently discover and eliminate "show stopper" bugs?
• How do I make manual testing more effective--and less boring and unpleasant?
• What's the most effective high-level test strategy for each project?
• Which inputs should I test when I can't test them all?
• Which test cases will provide the best feature coverage?
• How can I get better results by combining exploratory testing with traditional script or scenario-based testing?
• How do I reflect feedback from the development process, such as code changes?

Testing Computer SoftwareJohn Wiley & Sons

Decades of software testing experience condensed into the most important lessons learned. The world's leading software testing experts lend you their wisdom and years of experience to help you avoid the most common mistakes in testing software. Each lesson is an assertion related to software testing, followed by an explanation or example that shows you the how, when, and why of the testing lesson. More than just tips, tricks, and pitfalls to avoid, Lessons Learned in Software Testing speeds you through the critical testing phase of the software development project without the extensive trial and error it normally takes to do so. The ultimate resource for software testers and developers at every level of expertise, this guidebook features:
* Over 200 lessons gleaned from over 30 years of combined testing experience
* Tips, tricks, and common pitfalls to avoid by simply reading the book rather than finding out the hard way
* Lessons for all key topic areas, including test design, test management, testing strategies, and bug reporting
* Explanations and examples of each testing trouble spot help illustrate each lesson's assertion

As the title states, this is a simply introduction to software testing. It covers the basics of testing theory and terminology, how to write test plans, and how defects are found and reported. It also goes over more advanced testing topics such as performance testing, security testing, combinatorial testing and others. Written by a software engineer with more than fifteen years of software development and quality assurance experience, this book provides an industry-focused introduction to the field of software testing.

A Friendly Introduction to Software Testing

What To Do When Software Fails

Software Testing Techniques

Computerized Adaptive Testing

A Study Guide for the ISTQB Test Analyst and Technical Test Analyst Advanced Level Certificates 2012

Designing Embedded Hardware

Testing Computer Software provides a realistic, pragmatic introduction to testing consumer and business software under normal business conditions. This book will teach you how to test computer software under real-world conditions. The authors have all been test managers and software development managers at well-known Silicon Valley software companies. Successful consumer software companies have learned how to produce high quality products under tight time and budget constraints. The book explains the testing side of that success. This is a cookbook packed with code examples and step-by-step instructions to ease your learning curve. This book is intended for software quality assurance/testing professionals, software project managers, or software developers with prior experience in using Selenium and Java for testing web-based applications. This book also provides examples for Python, and Ruby users.

Summary The Art of Unit Testing, Second Edition guides you step by step from writing your first simple tests to developing robust test sets that are maintainable, readable, and trustworthy. You'll master the foundational ideas and quickly move to high-value subjects like mocks, stubs, and isolation, including frameworks such as Moq, FakeItEasy, and Typemock Isolator. You'll explore test patterns and organization, working with legacy code, and even "untestable" code. Along the way, you'll learn about integration testing and techniques and tools for testing databases and other technologies. About this book You know you should be unit testing, so why aren't you doing it? If you're new to unit testing, if you find unit testing tedious, or if you're just not getting enough payoff for the effort you put into it, keep reading. The Art of Unit Testing, Second Edition guides you step by step from writing your first simple unit tests to building complete test sets that are maintainable, readable, and trustworthy. You'll move quickly to more complicated subjects like mocks and stubs, while learning to use isolation (mocking) frameworks like Moq, FakeItEasy, and Typemock Isolator. You'll explore test patterns and organization, refactor code applications, and learn how to test "untestable" code. Along the way, you'll learn about integration testing and techniques for testing with databases. The examples in the book use C#, but will benefit anyone using a statically typed language such as Java or C++-. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. What's Inside Create readable, maintainable, trustworthy tests Fakes, stubs, mock objects, and isolation (mocking) frameworks Simple dependency injection techniques Refactoring legacy code About the Author Roy Osherover has been coding for over 15 years, and he consults and trains teams worldwide on the gentle art of unit testing and test-driven development. His blog is at ArtOfUnitTesting.com. Table of Contents PART 1 GETTING STARTED The basics of unit testing A first unit test PART 2 CORE TECHNIQUES Using stubs to break dependencies Interaction testing using mock objects Isolation (mocking) frameworks Digging deeper into isolation frameworks PART 3 THE TEST CODE Test hierarchies and organization The pillars of good unit tests PART 4 DESIGN AND PROCESS Integrating unit testing into the organization Working with legacy code Design and testability

The hardware and software of computing have changed markedly inthe three decades since the first edition of The Art of SoftwareTesting, but this book's powerful underlying analysis has stoodthe test of time. Whereas most books on software testing target particular development techniques, languages, or testing methods,The Art of Software Testing, Third Edition provides a briefbut powerful and comprehensive presentation of time-proven softwaretesting approaches. If your software development project is missioncritical, this book is an investment that will pay for itself withthe first bug you find. The new Third Edition explains how to apply the book'sclassic principles to today's hot topics including: Testing apps for iPhones, iPads, BlackBerry's, Androids, andother mobile devices Collaborative (user) programming and testing Testing for Internet applications, e-commerce, and agileprogramming environments Whether you're a student looking for a testing guide you'll usefor the rest of your career, or an IT manager overseeing a softwaredevelopment team, The Art of Software Testing, Third Editionis an expensive book that will pay for itself many times over.

Introduction to Software Testing

Handbook of Bioequivalence Testing, Second Edition

Help for Programmers and Quality Assurance

Theory and Practice

The Art of Unit Testing

Testing Applications on the Web

Technical and legal issues and be your own consumer advocate. Most software products are released with known defects. Misleading advertising is rampant in the industry, and few software publishers provide real warranties for their products. And as we all know, most software companies provide woefully inadequate technical support. Quite simply, consumers usually get the short end of the stick in the software industry. Not for long, if the authors of Bad Software can help it. This book pulls no punches in explaining why things are so bad, and how consumers can best stand up for themselves. The authors provide guidance on how to troubleshoot faulty software and when to call for help; exactly what to demand of software companies when defective products cost you time and money; how to ensure a replacement or refund; how best to deal with intransigent companies and their personnel; and much more. Written by industry insiders with software management, technical support management, and legal experience, this book will show you how to fight for your rights and get valuable results. Companion Web site features legislative and regulatory news and commentary, court cases, and contact information for protection agencies.

Usability Testing Essentials presents a practical, step-by-step approach to learning the entire process of planning and conducting a usability test. It explains how to analyze and apply the results and what to do when confronted with budgetary and time restrictions. This is the ideal book for anyone involved in usability or user-centered design—from students to seasoned professionals. Filled with new examples and case studies, Usability Testing Essentials, 2nd Edition is completely updated to reflect the latest approaches, tools and techniques needed to begin usability testing or to advance in this area. Provides a comprehensive, step-by-step guide to usability testing, a crucial part of every product's development Discusses important usability issues such as international testing, persona creation, remote testing, and accessibility Presents new examples covering mobile devices and apps, websites, web applications, software, and more Includes strategies for using tools for moderated and unmoderated testing, expanded content on task analysis, and on analyzing and reporting results

As the fourth edition of the Quality Assurance Institute, which sponsors the most widely accepted 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.

Intelligent readers who want to build their own embedded computer systems-- installed in everything from cell phones to cars to handheld organizers to refrigerators-- will find this book to be the most in-depth, practical, and up-to-date guide on the market. Designing Embedded Hardware carefully steers between the practical and philosophical aspects, so developers can both create their own devices and gadgets and customize and extend off-the-shelf systems. There are hundreds of books to choose from if you need to learn programming, but only a few are available if you want to learn to create hardware. Designing Embedded Hardware provides software and hardware engineers with no prior experience in embedded systems with the necessary conceptual and design building blocks to understand the architectures of embedded systems. Written to provide the depth of coverage and real-world examples developers need, Designing Embedded Hardware also provides a road-map to the pitfalls and traps to avoid in designing embedded systems. Designing Embedded Hardware covers such essential topics as: The principles of developing computer hardware Core hardware designs Assembly language concepts Parallel I/O Analog-digital conversion Timers (internal and external) UART Serial Peripheral Interface Inter-Integrated Circuit Bus Controller Area Network (CAN) Data Converter Interface (DC) Low-power operation This invaluable and eminently useful book gives you the practical tools and skills to develop, build, and program your own application-specific computers.

Lessons Learned in Software Testing

The Art of Application Performance Testing

A Context-Driven Approach

Bad Software

Exploratory Software Testing

This book will teach you how to test computer software under real-world conditions. The authors have all been test managers and software development managers at well-known Silicon Valley software companies. Successful consumer software companies have learned how to produce high-quality products under tight time and budget constraints. The book explains the testing side of that success. Who this book is for:
• Testers and Test Managers
• Project Managers-Understand the timeline, depth of investigation, and quality of communication to hold testers accountable for.
• Programmers-Gain insight into the sources of errors in your code, understand what tests your work will have to pass, and why testers do the things they do.
• Students-Train for an entry-level position in software development. What you will learn:
• How to find important bugs quickly
• How to describe software errors clearly
• How to create a testing plan with a minimum of paperwork
• How to design and use a bug-tracking system
• Where testing fits in the product development process
• How to test products that will be translated into other languages
• How to test for compatibility with devices, such as printers
• What laws apply to software quality

Many books cover functional testing techniques, but relatively few also cover technical testing. The Software Test Engineer's Handbook-2nd Edition fills that gap. Authors Graham Bath and Judy McKay are core members of the ISTQB Working Party that created the new Advanced Level Syllabus-Test Analyst and Advanced Level Syllabus-Technical Test Analyst. These syllabi were released in 2012. This book presents functional and technical aspects of testing as a coherent whole, which benefits test analysts/engineers and test managers. It provides a solid preparation base for passing the exams for Advanced Test Analyst and Advanced Technical Test Analyst, with enough real-world examples to keep you intellectually invested. This book includes information that will help you become a highly skilled Advanced Test Analyst and Advanced Technical Test Analyst. You will be able to apply this information in the real world of tight schedules, restricted resources, and projects that do not proceed as planned.

As the generic pharmaceutical industry continues to grow and thrive, so does the need to conduct adequate, efficient bioequivalence studies. In recent years, there have been significant changes to the statistical models for evaluating bioequivalence. In addition, advances in the analytical technology used to detect drug and metabolite levels have made bioequivalence testing more complex. The second edition of Handbook of Bioequivalence Testing has been completely updated to include the most current information available, including new findings in drug delivery and dosage form design and revised worldwide regulatory requirements. New topics include: A historical perspective on generic pharmaceuticals New guidelines governing submissions related to bioequivalency studies, along with therapeutic code classifications Models of noninferiority Biosimilarity of large molecule drugs Bioequivalence of complementary and alternate medicines Bioequivalence of bisimilar therapeutic proteins and monoclonal antibodies New FDA guidelines for bioanalytical method validation Outsourcing and monitoring of bioequivalence studies The cost of generic drugs is rising much faster than in the past, partly because of the increased costs required for approval—including those for bioequivalence testing. There is a dire need to re-examine the science behind this type of testing to reduce the burden of development costs—allowing companies to develop generic drugs faster and at a lower expense. The final chapter explores the future of bioequivalence testing and proposes radical changes in the process of bioawiaers. It suggests how the cost of demonstrating bioequivalence can be reduced through intensive analytical investigation and proposes that regulatory agencies reduce the need for bioequivalence studies in humans.

Backed by science and updated with the latest research, this book is destined to spark continued debate on the efficacy of the current bioequivalence testing paradigm.

CD-ROM contains: Canned HEAT v.2.0 -- Holodeck Lite v. 1.0.

Practical Software Testing

How to Break Software

The Cucumber Book

Software Testing

A Process-Oriented Approach

Principles and Practice

"Software Testing: Principles and Practices is a comprehensive treatise on software testing. It provides a pragmatic view of testing, addressing emerging areas like extreme testing and ad hoc testing"--Resource description page.

The second edition of the Handbook of Test Development provides graduate students and professionals with an up-to-date, research-oriented guide to the latest developments in the field. Including thirty-two chapters by well-known scholars and practitioners, it is divided into five sections, covering the foundations of test development, content definition, item development, test design and form assembly, and the processes of test administration, documentation, and evaluation. Keenly aware of developments in the field since the publication of the first edition, including changes in technology, the evolution of psychometric theory, and the increased demands for effective tests via educational policy, the editors of this edition include new chapters on assessing noncognitive skills, measuring growth and learning progressions, automated item generation and test assembly, and computerized scoring of constructed responses. The volume also includes expanded coverage of performance testing, validity, fairness, and numerous other topics. Edited by Suzanne Lane, Mark R. Raymond, and Thomas M. Haladyna, The Handbook of Test Development, 2nd edition, is based on the revised Standards for Educational and Psychological Testing, and is appropriate for graduate courses and seminars that deal with test development and usage, professional testing services and credentialing agencies, state and local boards of education, and academic libraries serving these groups.

The taxonomy of bugs, Flowcharts and path testing, Path testing and transaction flows. Graphs, paths and complexity. Paths, path products, and regular expressions. Data validation and syntax testing, Data-base-driven test design. Decision tables and boolean algebra. Boolean algebra the easy way. States, state graphs, and transition testing. Graph matrices and applications. Based on the needs of the educational community, and the software professional, this book takes a unique approach to teaching software testing. It introduces testing concepts that are managerial, technical, and process oriented, using the Testing Maturity Model (TMM) as a guiding framework. The TMM levels and goals support a structured presentation of fundamental and advanced test-related concepts to the reader. In this context, the interrelationships between theoretical, technical, and managerial concepts become more apparent. In addition, relationships between the testing process, maturity goals, and such key players as managers, testers and client groups are introduced. Topics and features - Process/engineering-oriented test - Promotes the growth and value of software testing as a profession - Introduces both technical and managerial aspects of testing in a clear and precise style - Uses the TMM framework to introduce testing concepts in a systematic, evolutionary way to facilitate understanding - Describes the role of testing tools and measurements, and how to integrate them into the testing process Graduate students and industry professionals will benefit from the book, which is designed for a graduate course in software testing, software quality assurance, or software validation and verification Moreover, the number of universities with graduate courses that cover this material will grow, given the evolution in software development as an engineering discipline and the creation of degree programs in software engineering.

Test-Driven Development with Python

Software Engineering and Testing

Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects

Building a Modern Computer from First Principles

Validation, Verification, and Testing of Computer Software

with examples in C#

Software Testing Techniques, 2nd Edition is the first book-length work that explicitly addresses the idea that design for testability is as important as testing itself not just by saying that testability is a desirable goal, but by showing the reader how it to do it. Every chapter has testability guidelines that illustrate how the technique discussed in the chapter can be used to make software more easily tested and therefore more reliable and maintainable. Application of all techniques to unit, integration, maintenance, and system testing are discussed throughout this book.As a self-study text, as a classroom text, as a working reference, it is a book that no programmer, independent software tester, software engineer, testing theorist, system designer, or software project manager can be without. This title gives students an integrated and rigorous picture of applied computer science, as it comes to play in the construction of a simple yet powerful computer system.

Your customers want rock-solid, bug-free software that does exactly what they expect it to do. Yet they can't always articulate their ideas clearly enough for you to turn them into code. You need Cucumber: a testing, communication, and requirements tool--all rolled into one. All the code in this book is updated for Cucumber 2.4, Rails 5, and RSpec 3.5. Express your customers' wild ideas as a set of clear, executable specifications that everyone on the team can read. Feed those examples into Cucumber and let it guide your development. Build just the right code to keep your customers happy. You can use Cucumber to test almost any system or any platform. Get started by using the core features of Cucumber and working with Cucumber's Gherkin DSL, to describe-in plain language--the behavior your customers want from the system. Then write Ruby code that interprets those plain-language specifications and checks them against your application. Next, consolidate the knowledge you've gained with a worked example, where you'll learn more advanced Cucumber techniques, test asynchronous systems, and test systems that use a database. Copies highlight some of the most difficult and commonly seen situations the authors have helped teams solve. With these patterns and techniques, test Ajax-heavy web applications with Capybara and Selenium, REST web services, Ruby on Rails applications, command-line applications, legacy applications, and more. Written by the creator of Cucumber and the co-founders of Cucumber Ltd., this authoritative guide will give you and your team all the knowledge you need to start using Cucumber with confidence. What You Need: Windows, Mac OS X (with XCode) or Linux, Ruby 1.9.2 and upwards, Cucumber 2.4, Rails 5, and RSpec 3.5

Software testing is one of the invisible jobs in the software industry. Everyone has heard of computer programmers but few people realize there are nearly as many people behind the scenes with job titles such as Software Tester, Software Quality Assurance Engineer, Software Test Engineer, and Software Test Technician. Microsoft alone hires hundreds of people for these positions each year. There are also many companies whose sole purpose is providing software test consulting services. The first edition of Software Testing was published in November 2000. Although the processes and techniques used in testing computer software are timeless, this title will be brought up-to-date by adding a chapter that specifically deals with testing software for security bugs and revisiting the rest of the book to update examples and references.

A Practical Guide to Testing

Testing Computer Software Second Edition

Code

Complete Guide to Test Automation

The Art of Software Testing

A Craftsman's Approach, Fourth Edition

Rely on this robust and thorough guide to build and maintain successful test automation. As the software industry shifts from traditional waterfall paradigms into more agile ones, test automation becomes a highly important tool that allows your development teams to deliver software at an ever-increasing pace without compromising quality. Even though it may seem trivial to automate the repetitive tester's work, using test automation efficiently and properly is not trivial. Many test automation endeavors end up in the "graveyard" of software projects. There are many things that affect the value of test automation, and also its costs. This book aims to cover all of these aspects in great detail so you can make decisions to create the best test automation solution that will not only help your test automation project to succeed, but also allow the entire software project to thrive. One of the most important details that affects the success of the test automation is how easy it is to maintain the automated tests. Complete Guide to Test Automation provides a detailed hands-on guide for writing highly maintainable test code. What You'll Learn Know the real value to be expected from test automation Discover the key traits that will make your test automation project succeed Be aware of the different considerations to take into account when planning automated tests vs. manual tests Determine who should implement the tests and the implications of this decision Architect the test project and fit it to the architecture of the tested application Design and implement highly reliable automated tests Begin gaining value from test automation earlier Integrate test automation into the business processes of the development teamLeverage test automation to improve your organization's performance and quality, even without formal authority Understand how different types of automated tests will fit into your testing strategy, including unit testing, load and performance testing, visual testing, and more Who This Book Is For Those involved with software development such as test automation leads, QA managers, test automation developers, and development managers. Some parts of the book assume hands-on experience in writing code in an object-oriented language (mainly C# or Java), although most of the content is also relevant for nonprogrammers.

Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking—and help you build the highest quality code. Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity Reap the benefits of collaborative development Apply defensive programming techniques to reduce and flush out errors Exploit opportunities to refactor—or evolve—code, and do it safely Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project This practical book provides a step-by-step approach to testing mission-critical applications for scalability and performance before they're deployed -- a vital topic to which other books devote one chapter, if that. Businesses today live and die by network applications and web services.

Because of the increasing complexity of these programs, and the pressure to deploy them quickly, many professionals don't take the time to ensure that they'll perform well and scale effectively. The Art of Application Performance Testing explains the complete life cycle of the testing process, and demonstrates best practices to help you plan, gain approval for, coordinate, and conduct performance tests on your applications. With this book, you'll learn to: Set realistic performance testing goals Implement an effective application performance testing strategy Interpret performance test results Grapple with different application technologies and architectures Use automated performance testing tools Test traditional local applications, web-based applications, and web services (SOAs) Recognize and resolves issues that are often overlooked in performance tests Written by a consultant with 30 years of experience in the IT industry and over 12 years' experience with performance testing, this easy-to-read book is illustrated with real-world examples and packed with practical advice. The Art of Application Performance Testing thoroughly explains the pitfalls of an inadequate testing strategy and offers you a robust, structured approach for ensuring that your applications perform well and scale effectively when the need arises. Ian has maintained a vendor-agnostic methodology beautifully in this material. The metrics and graphs, along with background information provided in his case studies, eloquently conveyed to the reader, "Methodology above all, tools at your discretion..." Ian's expertise shines through throughout the entire reading experience!-- Matt St. Onge, Enterprise Solution Architect, HCL Technologies America / Teradyne

Written by a leading expert in the field, this unique volume contains current test design approaches and focuses only on software test design. Copeland illustrates each test design through detailed examples and step-by-step instructions. Selenium Testing Tools Cookbook

Test Planning for Internet-Based Systems

Software Testing and Quality Assurance

A Primer

Includes Complete Guidelines, Checklists, and Templates

Obey the Testing Goat: Using Django, Selenium, and JavaScript

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.

Research Methods in Human-Computer Interaction is a comprehensive guide to performing research and is essential reading for both quantitative and qualitative methods. Since the first edition was published in 2009, the book has been adopted for use at leading universities around the world, including Harvard University, Carnegie-Mellon University, the University of Washington, the University of Toronto, HiOA (Norway), KTH (Sweden), Tel Aviv University (Israel), and many others. Chapters cover a broad range of topics relevant to the collection and analysis of HCI data, going beyond experimental design and surveys, to cover ethnography, diaries, physiological measurements, case studies, crowdsourcing, and other essential elements in the well-informed HCI researcher's toolkit. Continual technological evolution has led to an explosion of new techniques and a need for this updated 2nd edition, to reflect the most recent research in the field and newer trends in research methodology. This Research Methods in HCI revision contains updates throughout, including more detail on statistical tests, coding qualitative data, and data collection via mobile devices and sensors. Other new material covers performing research with children, older adults, and people with cognitive impairments. Comprehensive and updated guide to the latest research methodologies and approaches, and now available in EPUB3 format (choose any of the ePub or Mobi formats after purchase of the eBook). Expanded discussions of online datasets, crowdsourcing, statistical tests, coding qualitative data, laws and regulations relating to the use of human participants, and data collection via mobile devices and sensors New material on performing research with children, older adults, and people with cognitive impairments, two new case studies from Google and Yahoo!, and techniques for expanding the influence of your research to reach non-researcher audiences, including software developers and policymakers

This edition of Foundations of Software Testing is aimed at the undergraduate, the graduate students and the practicing engineers. It presents sound engineering approaches for test generation, ion, minimization, assessment, and enhancement. Using numerous examples, it offers a lucid description of a wide range of simple to complex techniques for a variety of testing-related tasks. It also discusses the comparative analyses of commercially available testing tools to facilitate the tool ion.

A software testing survival guide for those who work in Internet time With Internet applications spreading like wildfire, the field of software testing is increasingly challenged by the brave new networked world of e-Business. This book brings you up to speed on the technologies, testing concepts, and tools you'll need to run e-business applications on the Web. Written by Hung Nguyen, a coauthor of the bestselling software testing book of all time, Testing Computer Software, this new guide takes you to the next level, helping you apply your existing skills to the testing of B2B (Business-to-Business), B2C (Business-to-Consumer), and internal Web-based applications. You'll learn how to test transactions across networks, explore complex systems for errors, and work efficiently with the many components at play—from servers to browsers to protocols. Most importantly, you'll get detailed instructions on how to carry out specific test types along with case studies and error examples for each test. Software testers, test leads and test managers, QA analysts and managers, and IT managers and staff will find this an invaluable resource for their testing projects. With an emphasis on achievable goals and necessary rather than nice-to-have features, Testing Applications on the Web provides: An analysis of the Web-application model and the difference between Web testing and traditional testing A tutorial on the methodology and techniques for networking technologies and component-based testing Strategies for test planning, test case designing, and error analysis on the Web Effective real-world practices for UI (User Interface) tests, security tests, installation tests, load and stress tests, database tests, and more A survey of commercial tools and a sampling of proven test matrices and templates

Tips, Tricks, Tours, and Techniques to Guide Test Design

Foundations of Software Testing, 2/e

Testing Computer Software

Behaviour-Driven Development for Testers and Developers

Advanced Software Testing - Vol. 2, 2nd Edition

Usability Testing Essentials: Ready, Set...Test!

This updated and reorganized Fourth edition of Software Testing: A Craftsman's Approach applies the strong mathematics content of previous editions to a coherent treatment of model-based (structural) and specification-based (functional) testing. These techniques are extended from the usual unit testing discussions to full coverage of less understood levels integration and system testing. The Fourth Edition: Emphasizes technical inspections and is supplemented by an appendix with a full package of documents required for a simple Use Case technical inspection Introduces an innovative approach that merges the Even-Driven Petri Nets from the earlier editions with the "Swim Lane" concept from the Unified Modeling Language (UML) that permits model-based testing for four levels of interaction among constituents in a System of Systems Introduces model-based development and provides an explanation of how to conduct testing within model-based development environments Presents a new section on methods for testing software in an Agile programming environment Explores test-driven development, reexamines all-pairs testing, and explains the four contexts of software testing. Thoroughly revised and updated, Software Testing: A Craftsman's Approach, Fourth Edition is sure to become a standard reference for those who need to stay up to date with modern technologies in software testing. Carrying on the tradition of previous editions, it will continue to serve as a valuable reference for software testers, developers, and engineers.

By taking you through the development of a real web application from beginning to end, the second edition of this hands-on guide demonstrates the practical advantages of test-driven development (TDD) with Python. You'll learn how to write and run tests before building each part of your app, and then develop the minimum amount of code required to pass those tests. The result? Clean code that works. In the process, you'll learn the basics of Django, Selenium, Git, JQuery, and Mock, along with current web development techniques. If you're ready to take your Python skills to the next level, this book—updated for Python 3.6—clearly demonstrates how TDD encourages simple designs and inspires confidence. Dive into the TDD workflow, including the unit test/code cycle and refactoring. Use unit tests for classes and functions, and functional tests for user interactions within the browser. Learn when and how to use mock objects, and the pros and cons of isolated vs. integrated tests. Test and automate your deployments with a staging server. Apply tests to the third-party plugins you integrate into your site. Run tests automatically by using a Continuous Integration environment. Use TDD to build a REST API with a front-end Ajax interface.

The Software Test Engineer's Handbook

The Hidden Language of Computer Hardware and Software

Research Methods in Human-Computer Interaction