Effective Unit Testing A For Java Developers

Testing is a cornerstone of XP, as tests are written for every piece of code before it is programmed. This workbook helps testers learn XP, and XP devotees learn testing. This new book defines how an XP tester can optimally contribute to a project, including what testers should do, when they should do it, and how they should do it.

Few .NET developers have the luxury of unlimited code testing once their application is complete, and rushing through the testing process is both problematic and stressful. The open source NUnit framework provides an excellent and efficient way to test and improve .NET code as it's written, saving hundreds of QA hours and headaches.NUnit is one of the most mature and widely-used .NET open source projects even Microsoft uses it internally. NUnit is a unit-testing framework for all .Net languages. Written entirely in C#, NUnit takes advantage of many .NET language features, such as custom attributes and other reflection related capabilities. It automates unit testing and reduces the effort required to frequently test code while developing it. NUint is invaluable for .NET developers in test-driven development under agile methodologies such as Extreme Programming (XP)

as well as for developers who use for unit testing for software quality assurance. Unfortunately, some of those valuable hours saved by using NUnit can be wasted trying to master this powerful but under-documented frameworkProof that good things come in small packages, the NUnit Pocket Reference is a complete reference to NUnit, filling in the blanks left by the existing documentation and online discussion. It offers developers everything they need to know to install, configure, and use NUnit; the NUnit user interface; and a reference to the NUnit framework classes in a slim but well-organized package. This handy little book even offers practical, real world NUnit examples. And with the NUnit Pocket Reference, IT managers will know to expect when they implement unit testing in their projects. It is the only book you'll need on this popular and practical new open source framework.

Explore the new way of building and maintaining test cases with Java test driven development (TDD) using JUnit 5. This book doesn't just talk about the new concepts, it shows you ways of applying them in TDD and Java 8 to continuously deliver code that excels in all metrics. Unit testing and test driven development have now become part of every developer's skill set. For Java developers, the most popular testing tool has been JUnit, and JUnit 5 is built using the latest features of Java. With Java Unit Testing with JUnit 5, you'll master these new

features, including method parameters, extensions, assertions and assumptions, and dynamic tests. You'll also see how to write clean tests with less code. This book is a departure from using older practices and presents new ways of performing tests, building assertions, and injecting dependencies. What You Will Learn Write tests the JUnit 5 way Run your tests from within your IDE Integrate tests with your build and static analysis tools Migrate from JUnit 4 to JUnit 5 Who This Book Is For Java developers both with and without any prior unit testing experience.

From the basics to the most advanced quality of service (QoS) concepts, this all encompassing, first-of-its-kind book offers an in-depth understanding of the latest technical issues raised by the emergence of new types, classes and qualities of Internet services. The book provides end-to-end QoS guidance for real time multimedia communications over the Internet. It offers you a multiplicity of hands-on examples and simulation script support, and shows you where and when it is preferable to use these techniques for QoS support in networks and Internet traffic with widely varying characteristics and demand profiles. This practical resource discusses key standards and protocols, including real-time transport, resource reservation, and integrated and differentiated service models, policy based management, and mobile/wireless QoS. The book

features numerous examples, simulation results and graphs that illustrate important concepts, and pseudo codes are used to explain algorithms. Case studies, based on freely available Linux/FreeBSD systems, are presented to show you how to build networks supporting Quality of Service. Online support material including presentation foils, lab exercises and additional exercises are available to text adopters.

This book explains in detail how to implement unit tests using two very popular open source Java technologies: JUnit and Mockito. It presents a range of techniques necessary to write high quality unit tests - e.g. mocks, parametrized tests and matchers. It also discusses trade-offs related to the choices we have to make when dealing with some real-life code issues. It stresses the importance of writing readable and maintainable unit tests, and puts a lot of stress on code quality. It shows how to achieve testable code and to eliminate common mistakes by following the Test Driven Development approach. Every topic discussed in the book is illustrated with code examples, and each chapter is accompanied by some exercises

Can a system be considered truly reliable if it isn't fundamentally secure? Or can it be considered secure if it's unreliable? Security is crucial to the design and operation of scalable systems in production, as it plays an important part in product

quality, performance, and availability. In this book, experts from Google share best practices to help your organization design scalable and reliable systems that are fundamentally secure. Two previous O'Reilly books from Google—Site Reliability Engineering and The Site Reliability Workbook—demonstrated how and why a commitment to the entire service lifecycle enables organizations to successfully build, deploy, monitor, and maintain software systems. In this latest guide, the authors offer insights into system design, implementation, and maintenance from practitioners who specialize in security and reliability. They also discuss how building and adopting their recommended best practices requires a culture that's supportive of such change. You'll learn about secure and reliable systems through: Design strategies Recommendations for coding, testing, and debugging practices Strategies to prepare for, respond to, and recover from incidents Cultural best practices that help teams across your organization collaborate effectively

These proceedings include tutorials and papers presented at the Sixth CSR Confer ence on the topic of Large Software Systems. The aim of the Conference was to identify solutions to the problems of developing and maintaining large software systems, based on approaches which are currently being undertaken by software practitioners. These

proceedings are intended to make these solutions more widely available to the software industry. The papers from software practitioners describe: • important working systems, highlighting their problems and successes; • techniques for large system development and maintenance, including project management, quality management, incremental delivery, system security, in dependent V & V, and reverse engineering. In addition, academic and industrial researchers discuss the practical impact of current research in formal methods, object-oriented design and advanced environ ments. The keynote paper is provided by Professor Brian Warboys of ICL and the University of Manchester, who masterminded the development of the ICL VME Operating System, and the production of the first database-driven software en gineering environment (CADES). The proceedings commence with reports of the two tutorial sessions which preceded the conference: • Professor Keith Bennett of the Centre for Software Maintenance at Durham University on Software Maintenance; • Professor John McDermid of the University of York on Systems Engineering Environments for High Integrity Systems. The remaining papers deal with reports on existing systems (starting with Professor Warboys' keynote paper), approaches to large systems development, methods for large systems maintenance and the expected impact of current

research.

2nd edition of the step-by-step guide that helps developers to write test sets that are maintainable, readable and trustworthy.

Unit Testing Principles, Practices, and PatternsManning Publications

Using Test-Driven Development (TDD), thousands of agile developers are delivering software that is more rigorously tested, better designed, more robust, and easier to maintain. Until recently, however, database developers have been stymied by the unique characteristics of SQL code and the scarcity of useful tools. If you're an SQL Server developer who wants to use TDD, solutions now exist. In Database Unit Testing for SQL Server Using tSQLt , two pioneering database developers introduce those solutions and show you exactly how to apply them. Dennis Lloyd, Jr. and Sebastian Meine introduce their powerful new tSQLt open source unit testing framework for SQL Server, and show how to take full advantage of it. Lloyd and Meine first explain the key TDD concepts that have made unit testing of non-database projects so effective, including isolating functions under test, using mocks to break up dependencies, and incrementally guiding designs. Next, they explain why these concepts haven't translated neatly to database development, and show how to overcome those disconnects. Then, building on this foundation, they introduce tSQLt and

show you how to use it to create higher quality SQL Server code. Coverage includes: Why TDD can be as valuable in database development as it is in object-oriented development Writing effective unit tests for SQL Server T-SQL code Quick, time-saving heuristics for identifying test cases Advanced use cases for tSQLt and TDD Applying TDD to new code Refactoring existing database code through TDD Integrating tSQLt and TDD with other development processes Building and deploying databases in continuous improvement environments

"Structured Software Testing- The Discipline of Discovering Software Errors" is a book that will be liked both by readers from academia and industry. This book is unique and is packed with software testing concepts, techniques, and methodologies, followed with a step-by-step approach to illustrate real-world applications of the same. Well chosen topics, apt presentation, illustrative approach, use of valuable schematic diagrams and tables, narration of best practices of industry are the highlights of this book and make it a must read book. Key Features of the Book: - Well chosen and sequenced chapters which make it a unique resource for test practitioners, also, as a text at both graduate and post-graduate levels. - Apt presentation of Testing Techniques covering Requirement Based: Basic & Advanced, Code Based: Dynamic & Static, Data Testing, User Interface, Usability, Internationalization

& Localization Testing, and various aspects of bugs which are narrated with carefully chosen examples. - Illustrative approach to demonstrate software testing concepts, methodologies, test case designing and steps to be followed, usefulness, and issues. - Valuable schematic diagrams and tables to enhance ability to comprehend the topics explained - Best practices of industry and checklists are nicely fitted across different sections of the book.

The second XP Universe and ?rst Agile Universe brought together many p- ple interested in building software in a new way. Held in Chicago, August 4–7, 2002 it attracted software experts, educators, and developers. Unlike most c- ferences the venue was very dynamic. Many activities were not even well de?ned in advance. All discussions were encouraged to be spontaneous. Even so, there were some written words available and you are holding all of them now. We have collected as much material as possible together into this small volume. It is just the tip of the iceberg of course. A reminder to us of what we learned, the people we met, and the ideas we expressed. The conference papers, including research and experience papers, are rep- duced in these proceedings. Forty-one (41) papers were submitted. Each subm- ted paper received three reviews by program committee members. The program committee consisted of 40 members. Papers submitted by program committee members

were refereed separately. This ensured that reviewers could provide an honest feedback not seen by the paper submitters. In many cases, the program committee shepherded authors to signi?cantly improve their initial submission prior to completing the version contained in these proceedings. In the end, the program committee chose 25 papers for publication (60% acceptance). Summary Effective Unit Testing is written to show how to write good tests—tests that are concise and to the point, expressive, useful, and maintainable. Inspired by Roy Osherove's bestselling The Art of Unit Testing, this book focuses on tools and practices specific to the Java world. It introduces you to emerging techniques like behavior-driven development and specification by example, and shows you how to add robust practices into your toolkit. About Testing Test the components before you assemble them into a full application, and you'll get better software. For Java developers, there's now a decade of experience with well-crafted tests that anticipate problems, identify known and unknown dependencies in the code, and allow you to test components both in isolation and in the context of a full application. About this Book Effective Unit Testing teaches Java developers how to write unit tests that are concise, expressive, useful, and maintainable. Offering crisp explanations and easy-to-absorb examples, it introduces emerging

techniques like behavior-driven development and specification by example. Programmers who are already unit testing will learn the current state of the art. Those who are new to the game will learn practices that will serve them well for the rest of their career. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. About the Author Lasse Koskela is a coach, trainer, consultant, and programmer. He hacks on open source projects, helps companies improve their productivity, and speaks frequently at conferences around the world. Lasse is the author of Test Driven, also published by Manning. What's Inside A thorough introduction to unit testing Choosing best-of-breed tools Writing tests using dynamic languages Efficient test automation Table of Contents PART 1 FOUNDATIONS The promise of good tests In search of good Test doubles PART 2 CATALOG Readability Maintainability Trustworthiness PART 3 DIVERSIONS Testable design Writing tests in other JVM languages Speeding up test execution

Kotlin is a powerful and pragmatic language, but it's not enough to know about its features. We also need to know when they should be used and in what way. This book is a guide for Kotlin developers on how to become excellent Kotlin developers. It presents and explains in-depth the best practices for Kotlin development. Each item is presented as a clear rule

of thumb, supported by detailed explanations and practical examples.

Fearlessly change the design of your iOS code with solid unit tests. Use Xcode's built-in test framework XCTest and Swift to get rapid feedback on all your code - including legacy code. Learn the tricks and techniques of testing all iOS code, especially view controllers (UIViewControllers), which are critical to iOS apps. Learn to isolate and replace dependencies in legacy code written without tests. Practice safe refactoring that makes these tests possible, and watch all your changes get verified quickly and automatically. Make even the boldest code changes with complete confidence. Manual code and UI testing get slower the deeper your navigation hierarchy goes. It can take several taps just to reach a particular screen, never mind the actual workflow tests. Automatic unit testing offers such rapid feedback that it can change the rules of development. Bring testing to iOS development, even for legacy code. Use XCTest to write unit tests in Swift for all your code. iOS developers typically reserve unit tests for their model classes alone. But that approach skips most of the code common to iOS apps, especially with UIViewControllers. Learn how to unit test these view controllers to expand your unit testing possibilities. Since good unit tests form the bedrock for safe refactoring, you're empowered to make bold changes. Learn how to

avoid the most common mistakes Swift programmers make with the XCTest framework. Use code coverage to find holes in your test suites. Learn how to identify hard dependencies. Reshape the design of your code quickly, with less risk and less fear.

Over the past decade, software engineering has developed into a highly respected field. Though computing and software engineering education continues to emerge as a prominent interest area of study, few books specifically focus on software engineering education itself. Software Engineering: Effective Teaching and Learning Approaches and Practices presents the latest developments in software engineering education, drawing contributions from over 20 software engineering educators from around the globe. Encompassing areas such as student assessment and learning, innovative teaching methods, and educational technology, this much-needed book greatly enhances libraries with its unique research content.

* 1st and only book to market on the open source Spring MVC and Web Flows, positioned to become the new "Struts." * Will be the only authoritative solution, by the Spring MVC and Spring Web Flows project leads themselves. * Two markets for this book. 1) Ex-patriots from the Struts world who have developed numerous web applications, but are looking for more and willing to take the initiative to experiment with new solutions; and

2) early adopter web developers into Web Flow, which has created a lot of buzz and will generate interest around this book as well as Spring MVC.

Beginner to Expert in Web development with JavaScript: From HTML to React-Redux KEY FEATURES a- Acquire web development skills to build independent applications a- Understand the basics of HTML, CSS, JavaScript, React and Reduxa- Create build beautiful applications using HTML, CSS, JavaScript, React and Reduxa- Learn how to debug and unit test your applications properly to build good end productsa- Follow best practices to write good quality code and build performant applicationsDESCRIPTION This book will take you on a complete journey of learning web development, starting right with the basics. The book begins with the history of web development and JavaScript, how it has evolved over these years, and how it still keeps growing with new features. Next, you will learn the basic pillars of web development - HTML, CSS, and JavaScript. You will learn about the functional, object-oriented programming and asynchronous behaviour, and how JavaScript provides for these. Empowered with the basics, you will proceed to learn the new features of JavaScript, ES2015, and the latest ES2019. Next, you will apply your learning to build a real application to see how the Web takes shape.At the end, you will also have an introductory section on ReactJS, one of the modern frameworks for UI development and also develop a simple weather application using React. You will be introduced to Redux as the state container for React applications. This book will conclude with an introductory

look at additional topics which can be taken up to become a professional and in building enterprise level applications.WHAT WILL YOU LEARN By the end of the book, you will be building real web applications to put your knowledge to practice. This book introduces all the concepts to get started with web application development. To further excel in this field, you really need to practice by building a lot many applications, implementing your own ideas or imitating existing websites. Also remember to practice additional examples provided in the code bundle of the book to master this field.WHO THIS BOOK IS FOR This book can be used by people who are completely new to software development and want to get into front-end web development by starting from basics. This book can also be used by JavaScript users for a quick reference to the fundamentals of HTML, CSS, JS, and learn ReactJS with Redux, as well as the new features in JavaScript ES2019.Table of Contents1. History of JS and how it has revolutionized web development2. HTML: Creating Web Content 3. CSS: Making content beautiful4. JavaScript Programming: Making application Interactive5. Functional programming with JavaScript6. Object-Oriented JavaScript7. Asynchronous Programming8. What's new in ES2019 JavaScript9. Building an application with JavaScript10. Debugging JavaScript Applications11. Unit test automation12. Build and Deploy an Application13. JavaScript Best Practices14. Introduction to React15. Building an application with React 16. State Management in React applications17. Debugging, Testing and Deploying React applications18.

What is next - for becoming a pro?About the AuthorsAlok Ranjan is a B.Tech in Computer Science and Engineering from IIT Delhi. After graduating in 2001, Alok worked with companies like Newgen, Virtusa, and Convergys before starting his entrepreneurial journey. Abhilasha Sinha is BE Computer Science from Osmania University. After completing her graduation in 2003, she started her career as a Software Engineer with Infosys. She went on to become Senior Technology Architect in a long and fulfilling association of 12 years. Ranjit Battewad has over 8 years of experience in full-stack web and mobile application development. Associated with WalkingTree Technologies since past over 8 years and playing the role of senior technical lead, he has exposure to complex application architecture design and development.

??????????20??????.????????????,???????????????,????????????????????.??,????60???????,????????????????????????.

Objective-C Programmer's Reference provides the tools necessary to write software in Objective-C—the language of choice for developing iOS and OS X applications. Author Carlos Oliveira begins from the basic building blocks of the language. He shows how to create correct and efficient applications by applying your knowledge of object-oriented and structured programming. This book: Takes you quickly through fundamental concepts such as interfaces and class implementations. Provides a concise reference to the Foundation Framework that is all-important when programming in Objective-C. Highlights key differences between Objective-C and

other popular languages such as Java or Python. Provides the fundamentals of Cocoa and Cocoa Touch, which are the standard for OS X and iOS development. Objective-C Programmer's Reference makes extensive use of concepts already mastered by developers who are fluent in other languages such as C++, Java, Perl, and Python. The author's approach is logical and structured, and even novice developers will have an easy time absorbing the most important topics necessary to program in Objective-C. Objective-C Programmer's Reference is a book for professional developers in Objective-C, or those who are moving to Objective-C from other languages. The book is written for readers who lack the time to invest in more traditional books, which usually spend hundreds of pages to explain concepts that are part of the working programmer's standard vocabulary.

Good Code, Bad Code is a clear, practical introduction to writing code that's a snap to read, apply, and remember. With dozens of instantly-useful techniques, you'll find coding insights that normally take years of experience to master. In this fast-paced guide, Google software engineer Tom Long teaches you a host of rules to apply, along with advice on when to break them!

The Pernambuco School on Software Engineering (PSSE) 2007 was the second in a series of events devoted to the study of advanced computer science and to the promotion of international scienti?c collaboration. The main theme in 2007 was testing. Testing is nowadays a key activity for assuring software quality. The summer school and its proceedings were intended

to give a detailed tutorial introduction to the scienti?c basis of this activity and its state of the art. Theseprocee dingsrecordthecontributionsfromtheinvitedlecturers.Each of thechaptersistheresultofathoroughrevisionoftheinitialno tesprovidedtothe participants of the school. The revision was inspired by the synergy generated by the opportunity for the lecturers to present and discuss their work among themselves and with the school's attendees. The editors have tried to produce a coherent view of the topic by harmonizing these contributions, smoothing out di?erences in notation and approach, and providing links between the lectures. We apologize to the authors for any errors introduced by our extensive editing. Although the chapters are linked in severalways, each one is su?ciently se- contained to be read in isolation. Nevertheless, Chap. 1 should be read ?rst by those interested in an introduction to testing. Chapter 1 introduces the terminology adopted in this book. It also provides an overview of the testing process, and of the types (functional, structural, and so on)anddimensions(u nit,integration,andsoon)ofthetestingactivity.Themain strategies employed in the central activity of test selection are also discussed. Most of the material presented in this introductory chapter is addressedin more depth in the following chapters.

This book contains most of the papers presented at the 4th International C- ference on Extreme Programming and Agile Processes in Software Engineering (XP 2003), held in Genoa, Italy, May 2003. The XP 200n series of conferences were started in 2000 to promote the - change of new ideas, research and applications in the

emerging ?eld of agile methodologies for software development. Over the years, the conference has - come the main world forum for all major advances in this important ?eld. Also this year the contributions to Agile Methodologies and Extreme P- gramming were substantial. They demonstrate that the topic is continuing to gain more and more momentum. In spite of some criticism of agile meth- ologies, everyone agrees that they address some unresolved needs of software practitioners. People still do not know how to develop software on time, with the desired features, and within the given budget! This volume is divided into several thematic sections, easing reader's na- gation through the content. Full papers are presented ?rst, followed by research reports, papers from the Educational Symposium, and papers from the Ph.D. Symposium. The presentations given during three panel sessions held at the conference conclude the book. The section on Managing Agile Processes includes contributions highlighting the sometimes di?cult relationship between agile methodologies and mana- ment, and includes approaches and suggestions that should facilitate the acc- tance of agile methodologies at the di?erent levels of management.

The Pragmatic Programmers classic is back! Freshly updated for modern software development, Pragmatic Unit Testing in Java 8 With JUnit teaches you how to write and run easily maintained unit tests in JUnit with confidence. You'll learn mnemonics to help you know what tests to write, how to remember all the boundary conditions, and what the qualities of a good test are.

You'll see how unit tests can pay off by allowing you to keep your system code clean, and you'll learn how to handle the stuff that seems too tough to test. Pragmatic Unit Testing in Java 8 With JUnit steps you through all the important unit testing topics. If you've never written a unit test, you'll see screen shots from Eclipse, IntelliJ IDEA, and NetBeans that will help you get past the hard part--getting set up and started. Once past the basics, you'll learn why you want to write unit tests and how to effectively use JUnit. But the meaty part of the book is its collected unit testing wisdom from people who've been there, done that on production systems for at least 15 years: veteran author and developer Jeff Langr, building on the wisdom of Pragmatic Programmers Andy Hunt and Dave Thomas. You'll learn: How to craft your unit tests to minimize your effort in maintaining them. How to use unit tests to help keep your system clean. How to test the tough stuff. Memorable mnemonics to help you remember what's important when writing unit tests. How to help your team reap and sustain the benefits of unit testing. You won't just learn about unit testing in theory--you'll work through numerous code examples. When it comes to programming, hands-on is the only way to learn!

A team of Microsoft insiders shows programmers how to use Visual Studio 2005 Team System, the new suite of products from Microsoft that can be used for software modeling, design, testing, and deployment Focuses on practical application of the tools on code samples, development scenarios, and automation scripting This timely book serves as both as a step-by-step guide and as a reference for modeling, designing, and coordinating enterprise solutions at

every level using Team System The book begins with an overview of Team System and then offers nuts-and-bolts guidance on practical implementation Code examples are provided in both VB.NET and C#

Effective Software Testing explores fifty critically important best practices, pitfalls, and solutions. Gleaned from the author's extensive practical experience, these concrete items will enable quality assurance professionals and test managers to immediately enhance their understanding and skills, avoid costly mistakes, and implement a state-of-the-art testing program. This book places special emphasis on the integration of testing into all phases of the software development life cycle--from requirements definition to design and final coding. The fifty lessons provided here focus on the key aspects of software testing: test planning, design, documentation, execution, managing the testing team, unit testing, automated testing, nonfunctional testing, and more. You will learn to: Base testing efforts on a prioritized feature schedule Estimate test preparation and execution Define the testing team roles and responsibilities Design test procedures as soon as requirements are available Derive effective test cases from requirements Avoid constraints and detailed data elements in test procedures Make unit-test execution part of the build process Use logging to increase system testability Test automated test tools on an application prototype Automate regression tests whenever possible Avoid sole reliance on capture/playback Conduct performance testing with production-sized databases Tailor usability tests to the intended audience Isolate the test environment from the development environment Implement a defect tracking life cycle Throughout the book, numerous real-world case studies and concrete examples illustrate the successful application of these important principles and techniques. Effective Software Testing provides ready access to the expertise and advice of

one of the world's foremost software quality and testing authorities. 0201794292B12032002

JUnit in Action, Third Edition has been completely rewritten for this release. The book is full of examples that demonstrate JUnit's modern features, including its new architecture; nested, tagged, and dynamic tests; and dependency injection. Summary JUnit is the gold standard for unit testing Java applications. Filled with powerful new features designed to automate software testing, JUnit 5 boosts your productivity and helps avoid debugging nightmares. Whether you're just starting with JUnit or you want to ramp up on the new features, JUnit in Action, Third Edition has you covered. Extensively revised with new code and new chapters, JUnit in Action, Third Edition is an up-to-date guide to smooth software testing. Dozens of hands-on examples illustrate JUnit 5's innovations for dependency injection, nested testing, parameterized tests, and more. Throughout, you'll learn how to use JUnit 5 to automate your testing, for a process that consumes less resources, and gives you more time for developing. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology The JUnit framework is the gold standard for unit testing Java applications—and knowing it is an essential skill for Java developers. The latest version, JUnit 5, is a total overhaul, now supporting modern Java features like Lambdas and Streams. About the book JUnit in Action, Third Edition has been completely rewritten for this release. The book is full of examples that demonstrate JUnit's modern features, including its new architecture; nested, tagged, and dynamic tests; and dependency injection. You'll benefit from author Catalin Tudose's unique "pyramid" testing strategy, which breaks the testing process into layers and sets you on the path to bug-free code creation. What's inside Migrating from JUnit 4 to 5 Effective test automation Test-

driven development and behavior-driven development Using mocks for test isolation Connecting JUnit 5 with Maven or Gradle About the reader For intermediate Java developers. About the author Catalin Tudose has a Ph.D. in Computer Science, and over 15 years of experience as a Senior Java Developer and Technical Team Lead. Previous editions were authored by Petar Tahchiev, Felipe Leme, Gary Gregory, and Vincent Massol. Table of Contents PART 1 - JUNIT 1 JUnit jump-start 2 Exploring core JUnit 3 JUnit architecture 4 Migrating from JUnit 4 to JUnit 5 5 Software testing principles PART 2 - DIFFERENT TESTING STRATEGIES 6 Test quality 7 Coarse-grained testing with stubs 8 Testing with mock objects 9 In-container testing PART 3 - WORKING WITH JUNIT 5 AND OTHER TOOLS 10 Runing JUnit tests from Maven 3 11 Running JUnit tests from Gradle 6 12 JUnit 5 IDE support 13 Coninuous integration with JUnit 5 PART 4 - WORKING WITH MODERN FRAMEWORKS AND JUNIT 5 14 JUnit 5 extension model 15 Presentation-layer testing 16 Testing Spring applications 17 Testing Spring Boot applications 18 Testing a REST API 19 Testing database applications PART 5 - DEVELOPING APPLICATIONS WITH JUNIT 5 20 Test-driven development with JUnit 5 21 Behavior-driven development in JUnit 5 22 Implementing a test pyramid strategy with JUnit 5

Diagramming and process are important topics in today's software development world, as the UML diagramming language has come to be almost universally accepted. Yet process is necessary; by themselves, diagrams are of little use. Use Case Driven Object Modeling with UML - Theory and Practice combines the notation of UML with a lightweight but effective process - the ICONIX process - for designing and developing software systems. ICONIX has developed a growing following over the years. Sitting between the free-for-all of Extreme Programming and overly rigid processes such

as RUP, ICONIX offers just enough structure to be successful.

Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

A guide to alternatives to EJB covers such topics as managing transactions, designing applications, accessing data, using open source products to increase productivity, and solving problems.

Unit Testing Principles, Patterns and Practices shows you how to refine your existing unit tests by implementing modern best practices. You'll learn to spot which tests are performing, which need refactoring, and which need to be deleted

entirely! Upgrade your testing suite with new testing styles, good patterns, and reliable automated testing. Unit Testing Principles, Practices and Patterns is a practical guide to modern unit testing best practices. Microsoft MVP Vladimir Khorikov takes you hands-on with examples of the ideal unit test and unit testing practices, building your skills step by step on a solid foundation. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

A developer's guide that covers everything needed to build Windows phone 7.5 mobile apps.

This is the first International Conference on Advances in Computing (ICAdC-2012). The scope of the conference includes all the areas of New Theoretical Computer Science, Systems and Software, and Intelligent systems. Conference Proceedings is a culmination of research results, papers and the theory related to all the three major areas of computing mentioned above. Helps budding researchers, graduates in the areas of Computer Science, Information Science, Electronics, Telecommunication, Instrumentation, Networking to take forward their research work based on the reviewed results in the paper by mutual interaction through e-mail contacts in the proceedings.

Persistence is an important set of techniques and technologies for accessing and transacting data, and ensuring that data is mobile regardless of specific applications and contexts. In Java development, persistence is a key factor in enterprise, e-

commerce, and other transaction-oriented applications. Today, the Spring framework is the leading out-of-the-box solution for enterprise Java developers; in it, you can find a number of Java Persistence solutions. This book gets you rolling with fundamental Spring Framework 3 concepts and integrating persistence functionality into enterprise Java applications using Hibernate, the JavaTM Persistence API (JPA) 2, and the Grails Object Relational Mapping tool, GORM. Covers core Hibernate fundamentals, demonstrating how the framework can be best utilized within a Spring application context Covers how to use and integrate JPA 2, found in the new Java EE 6 platform Covers how to integrate and use the new Grails persistence engine, GORM

With over 75 million downloads per month, Spring Boot is the most widely used Java framework available. Its ease and power have revolutionized application development from monoliths to microservices. Yet Spring Boot's simplicity can also be confounding. How do developers learn enough to be productive immediately? This practical book shows you how to use this framework to write successful mission-critical applications. Mark Heckler from VMware, the company behind Spring, guides you through Spring Boot's architecture and approach, covering topics such as debugging, testing, and deployment. If you want to develop

cloud native Java or Kotlin applications with Spring Boot rapidly and effectively--using reactive programming, building APIs, and creating database access of all kinds--this book is for you. Learn how Spring Boot simplifies cloud native application development and deployment Build reactive applications and extend communication across the network boundary to create distributed systems Understand how Spring Boot's architecture and approach increase developer productivity and application portability Deploy Spring Boot applications for production workloads rapidly and reliably Monitor application and system health for optimal performance and reliability Debug, test, and secure cloud-based applications painlessly

Agile methods are gaining more and more interest both in industry and in research. Many industries are transforming their way of working from traditional waterfall projects with long duration to more incremental, iterative and agile practices. At the same time, the need to evaluate and to obtain evidence for different processes, methods and tools has been emphasized. Lech Madeyski offers the first in-depth evaluation of agile methods. He presents in detail the results of three different experiments, including concrete examples of how to conduct statistical analysis with meta analysis or the SPSS package, using as evaluation indicators the number of acceptance tests passed (overall and per hour)

and design complexity metrics. The book is appropriate for graduate students, researchers and advanced professionals in software engineering. It proves the real benefits of agile software development, provides readers with in-depth insights into experimental methods in the context of agile development, and discusses various validity threats in empirical studies.