

Explore It Elisabeth Hendrickson

Explore It!
 A Radical Enterprise
 The Evolution of International Security Studies
 Epilepsy Across the Spectrum
 Agile Testing
 Lessons Learned in Software Testing
 The Unicorn Project
 Buddha in Testing
 Fifty Quick Ideas to Improve Your User Stories
 Experiences of Test Automation
 The DevOps Handbook
 Software Testing
 ATDD by Example
 Head First HTML with CSS & XHTML
 Agile Conversations
 Roundtable on Project Management
 The Software Test Engineer's Handbook
 Pragmatic Guide to Git
 Leading Quality
 "Dear Evil Tester"
 Perfect Software--and Other Illusions about Testing
 Dragons Out!
 The Way of the Web Tester
 The Cucumber Book
 The Agile Self-assessment Game
 Software Testing
 The Phoenix Project
 Clean Craftsmanship
 More Agile Testing
 How Google Tests Software
 Exploring Requirements
 Test-Driven React
 How to Break Software
 The Art of Application Performance Testing
 The Art of Agile Development
 Agile Retrospectives
 The Art of Clean Code
 Exploratory Software Testing
 Hemingway's Boat
 Behavior-Driven Development with Cucumber

Explore It Elisabeth Hendrickson

Downloaded from process.ogleschool.edu by guest

LUCERO WARD

Explore It! Addison-Wesley

Learn eight principles to simplify your code and become a more effective (and successful) programmer. Most software developers waste thousands of hours working with overly complex code. The eight core principles in *The Art of Clean Coding* will teach you how to write clear, maintainable code without compromising functionality. The book's guiding principle is simplicity: reduce and simplify, then reinvest energy in the important parts to save you countless hours and ease the often onerous task of code maintenance. Bestselling author Christian Mayer leverages his experience helping thousands perfect their coding skills in this new book. With expert advice and real-world examples, he'll show you how to: Concentrate on the important stuff with the 80/20 principle -- focus on the 20% of your code that matters most Avoid coding in isolation: create a minimum viable product to get early feedback Write code cleanly and simply to eliminate clutter

Avoid premature optimization that risks over-complicating code Balance your goals, capacity, and feedback to achieve the productive state of Flow Apply the Do One Thing Well philosophy to vastly improve functionality Design efficient user interfaces with the Less is More principle Tie your new skills together into one unifying principle: Focus The Python-based *The Art of Clean Coding* is suitable for programmers at any level, with ideas presented in a language-agnostic manner.

A Radical Enterprise Pearson

Over a half-million sold! And available now, the Wall Street Journal Bestselling sequel *The Unicorn Project* "Every person involved in a failed IT project should be forced to read this book."—TIM O'REILLY, Founder & CEO of O'Reilly Media "The Phoenix Project is a must read for business and IT executives who are struggling with the growing complexity of IT."—JIM WHITEHURST, President and CEO, Red Hat, Inc. Five years after this sleeper hit took on the world of IT and flipped it on its head, the 5th Anniversary Edition of *The Phoenix Project* continues to guide IT in the DevOps revolution. In this newly updated and expanded edition of the bestselling *The Phoenix Project*, co-author Gene Kim includes a new afterword and a deeper delve into the

Three Ways as described in *The DevOps Handbook*. Bill, an IT manager at Parts Unlimited, has been tasked with taking on a project critical to the future of the business, code named Phoenix Project. But the project is massively over budget and behind schedule. The CEO demands Bill must fix the mess in ninety days or else Bill's entire department will be outsourced. With the help of a prospective board member and his mysterious philosophy of The Three Ways, Bill starts to see that IT work has more in common with a manufacturing plant work than he ever imagined. With the clock ticking, Bill must organize work flow streamline interdepartmental communications, and effectively serve the other business functions at Parts Unlimited. In a fast-paced and entertaining style, three luminaries of the DevOps movement deliver a story that anyone who works in IT will recognize. Readers will not only learn how to improve their own IT organizations, they'll never view IT the same way again. "This book is a gripping read that captures brilliantly the dilemmas that face companies which depend on IT, and offers real-world solutions."—JEZ HUMBLE, Co-author of *Continuous Delivery*, *Lean Enterprise*, *Accelerate*, and *The DevOps Handbook* [The Evolution of International Security Studies](#) Addison-Wesley Professional

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

[Epilepsy Across the Spectrum](#) John Wiley & Sons

How to Write Code You're Proud of . . . Every Single Day ". . . [A] timely and humble reminder of the ever-increasing complexity of our programmatic world and how we owe it to the legacy of humankind--and to ourselves--to practice ethical development. Take your time reading Clean Craftsmanship. . . . Keep this book on your go-to bookshelf. Let this book be your old friend--your Uncle Bob, your guide--as you make your way through this world with curiosity and courage." -- From the Foreword by Stacia Heimgartner Viscardi, CST & Agile Mentor In Clean Craftsmanship, the legendary Robert C. Martin ("Uncle Bob") has written the principles that define the profession--and the craft--of software development. Uncle Bob brings together the disciplines, standards, and ethics you need to deliver robust, effective code and to be proud of all the software you write. Robert Martin, the best-selling author of Clean Code, provides a pragmatic, technical, and prescriptive guide to the foundational disciplines of software craftsmanship. He discusses standards, showing how the world's expectations of developers often differ from their own and helping you bring the two in sync. Bob concludes with the ethics of the programming profession, describing the fundamental promises all developers should make to their colleagues, their users, and, above all, themselves. With Uncle Bob's insights, all programmers and their managers can consistently deliver code that builds trust instead of undermining it--trust among users and throughout societies that depend on software for their survival. Moving towards the "north star" of true software craftsmanship: the state of knowing how to program well Practical, specific guidance for applying five core disciplines: test-driven development, refactoring, simple design, collaborative programming, and acceptance tests How developers and teams can promote productivity, quality, and courage The true meaning of integrity and teamwork among programmers, and ten specific commitments every software professional should make Register your book for convenient access to the book's companion videos, updates, and/or corrections as they become available. See inside book for details.

Agile Testing Pearson Education

Need to learn how to wrap your head around Git, but don't need a lot of hand holding? Grab this book if you're new to Git, not to the world of programming. Git tasks displayed on two-page spreads provide all the context you need, without the extra fluff.

[Lessons Learned in Software Testing](#) Pearson Education

Uncover surprises, risks, and potentially serious bugs with exploratory testing. Rather than designing all tests in advance, explorers design and execute small, rapid experiments, using what they learned from the last little experiment to inform the next. Learn essential skills of a master explorer, including how to analyze software to discover key points of vulnerability, how to design experiments on the fly, how to hone your observation skills, and how to focus your efforts. Software is full of surprises. No matter how careful or skilled you are, when you create software it can behave differently than you intended. Exploratory testing mitigates those risks. Part 1 introduces the core, essential skills of a master explorer. You'll learn to craft charters to guide your exploration, to observe what's really happening (hint: it's harder than it sounds), to identify interesting variations, and to determine what expected behavior should be when exercising software in unexpected ways. Part 2 builds on that foundation. You'll learn how to explore by varying interactions, sequences, data, timing, and configurations. Along the way you'll see how to incorporate analysis techniques like state modeling, data modeling, and defining context diagrams into your explorer's arsenal. Part 3 brings the techniques back into the context of a software project. You'll apply the skills and techniques in a variety of contexts and integrate exploration into the development cycle from the very beginning. You can apply the techniques in this book to any kind of software. Whether you work on embedded systems, Web applications, desktop applications, APIs, or something else, you'll find this book contains a wealth of concrete and practical advice about exploring your software to discover its capabilities, limitations, and risks.

[The Unicorn Project](#) CRC Press

Janet Gregory and Lisa Crispin pioneered the agile testing discipline with their previous work, Agile Testing. Now, in More Agile Testing, they reflect on all they've learned since. They address crucial emerging issues, share evolved agile practices, and cover key issues agile testers have asked to learn more about. Packed with new examples from real teams, this insightful guide offers detailed information about adapting agile testing for your environment; learning from experience and continually improving your test processes; scaling agile testing across teams; and overcoming the pitfalls of automated testing. You'll find brand-new coverage of agile testing for the enterprise,

distributed teams, mobile/embedded systems, regulated environments, data warehouse/BI systems, and DevOps practices. You'll come away understanding • How to clarify testing activities within the team • Ways to collaborate with business experts to identify valuable features and deliver the right capabilities • How to design automated tests for superior reliability and easier maintenance • How agile team members can improve and expand their testing skills • How to plan “just enough,” balancing small increments with larger feature sets and the entire system • How to use testing to identify and mitigate risks associated with your current agile processes and to prevent defects • How to address challenges within your product or organizational context • How to perform exploratory testing using “personas” and “tours” • Exploratory testing approaches that engage the whole team, using test charters with session- and thread-based techniques • How to bring new agile testers up to speed quickly--without overwhelming them Janet Gregory is founder of DragonFire Inc., an agile quality process consultancy and training firm. Her passion is helping teams build quality systems. For almost fifteen years, she has worked as a coach and tester, introducing agile practices into companies of all sizes and helping users and testers understand their agile roles. She is a frequent speaker at agile and testing software conferences, and is a major contributor to the agile testing community. Lisa Crispin, an experienced agile testing practitioner and coach, regularly leads conference workshops on agile testing and contributes frequently to agile software publications. She enjoys collaborating as part of an awesome agile team to produce quality software. Since 1982, she has worked in a variety of roles on software teams, in a wide range of industries. She joined her first agile team in 2000 and continually learns from other teams and practitioners.

Buddha in Testing IT Revolution

From a National Book Critics Circle Award winner, a brilliantly conceived and illuminating reconsideration of a key period in the life of Ernest Hemingway that will forever change the way he is perceived and understood. Focusing on the years 1934 to 1961—from Hemingway's pinnacle as the reigning monarch of American letters until his suicide—Paul Hendrickson traces the writer's exultations and despair around the one constant in his life during this time: his beloved boat, Pilar. We follow him from Key West to Paris, to New York, Africa, Cuba, and finally Idaho, as he wrestles with his best angels and worst demons. Whenever he could, he returned to his beloved fishing cruiser, to exult in the sea, to fight the biggest fish he could find, to drink, to entertain celebrities and friends and seduce women, to be with his children. But as he began to succumb to the diseases of fame, we see that Pilar was also where he cursed his critics, saw marriages and friendships dissolve, and tried, in vain, to escape his increasingly diminished capacities. Generally thought of as a great writer and an unappealing human being, Hemingway emerges here in a far more benevolent light. Drawing on previously unpublished material, including interviews with Hemingway's sons, Hendrickson shows that for all the writer's boorishness, depression, and alcoholism, and despite his choleric anger, he was capable of remarkable generosity—to struggling writers, to lost souls, to the dying son of a friend. We see most poignantly his relationship with his youngest son, Gigi, a doctor who lived his adult life mostly as a cross-dresser, and died squalidly and alone in a Miami women's jail. He was the son Hemingway forsook the least, yet the one who disappointed him the most, as Gigi acted out for nearly his whole life so many of the tortured, ambiguous tensions his father felt. Hendrickson's bold and beautiful book strikingly makes the case that both men were braver than we know, struggling all their lives against the complicated, powerful emotions swirling around them. As Hendrickson writes, “Amid so much ruin, still the beauty.” Hemingway's Boat is both stunningly original and deeply gripping, an invaluable contribution to our understanding of this great American writer, published fifty years after his death.

[Fifty Quick Ideas to Improve Your User Stories](#) Pragmatic Bookshelf

Crispin and Gregory define agile testing and illustrate the tester's role with examples from real agile teams. They teach you how to use the agile testing quadrants to identify what testing is needed, who should do it, and what tools might help. The book chronicles an agile software development iteration from the viewpoint of a tester and explains the seven key success factors of agile testing.

[Experiences of Test Automation](#) Addison-Wesley Professional

Master BDD to deliver higher-value software more quickly To develop high-value products quickly, software development teams need better ways to collaborate. Agile methods like Scrum and Kanban are helpful, but they're not enough. Teams need better ways to work inside each sprint or work item. Behavior-driven development (BDD) adds just enough structure for product experts,

testers, and developers to collaborate more effectively. Drawing on extensive experience helping teams adopt BDD, Richard Lawrence and Paul Rayner show how to explore changes in system behavior with examples through conversations, how to capture your examples in expressive language, and how to flow the results into effective automated testing with Cucumber. Where most BDD resources focus on test automation, this guide goes deep into how BDD changes team collaboration and what that collaboration looks like day to day. Concrete examples and practical advice will prepare you to succeed with BDD, whatever your context or role. · Learn how to collaborate better by using concrete examples of system behavior · Identify your project's meaningful increment of value so you're always working on something important · Begin experimenting with BDD slowly and at low risk · Move smoothly from informal examples to automated tests in Cucumber · Use BDD to deliver more frequently with greater visibility · Make Cucumber scenarios more expressive to ensure you're building the right thing · Grow a Cucumber suite that acts as high-value living documentation · Sustainably work with complex scenario data · Get beyond the “mini-waterfalls” that often arise on Scrum teams

The DevOps Handbook Neuri Consulting Llp

In this work, over 40 pioneering implementers share their experiences and best practices in 28 case studies. Drawing on their insights, you can avoid the pitfalls associated with test automation, and achieve powerful results on every metric you care about: quality, cost, time to market, usability, and value.

Software Testing Dorset House

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?

ATDD by Example Cambridge University Press

Because performance is paramount today, this thoroughly updated guide shows you how to test mission-critical applications for scalability and performance before you deploy them—whether it's to the cloud or a mobile device. You'll learn the complete testing process lifecycle step-by-step, along with best practices to plan, coordinate, and conduct performance tests on your applications. Set realistic performance testing goals Implement an effective application performance testing strategy Interpret performance test results Cope with different application technologies and architectures Understand the importance of End User Monitoring (EUM) Use automated performance testing tools Test traditional local applications, web applications, and web services Recognize and resolves issues often overlooked in performance tests Written by a consultant with over 15 years' experience with performance testing, The Art of Application Performance Testing thoroughly explains the pitfalls of an inadequate testing strategy and offers a robust, structured approach for ensuring that your applications perform well and scale effectively when the need arises.

Head First HTML with CSS & XHTML IT Revolution

The Agile Self-Assessment Game is used by teams and organizations to self-assess their agility. Playing the game enables teams to reflect on their own team interworking, discover how agile they are and decide what they can do to increase their agility to deliver more value to their customers

and stakeholders. This is the first book specifically about Agile Self-assessments. In this book, Ben Linders explains what self-assessments are and why you would do them, and explores how to do them using the Agile Self-assessment Game. He's also sharing experience stories from people who played the game. This book is based on his 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 self-assessments, viewing them from different perspectives and provides ideas, suggestions, practices, and experiences that will help you to do effective agile self-assessments with your teams. The book is aimed at Scrum masters, agile coaches, consultants leading agile transformations, developers and testers, project managers, line managers, and CxOs; basically for anyone who is looking for an effective way to help their agile teams improve and to increase the agility of their organization. With plenty of ideas, suggestions, and practical cases on Agile Self-assessments, this book will help you to apply assessments and help teams to improve. Note: The agile coaching cards needed to play the games described in the book can be downloaded for a nominal fee at benlinders.com/downloads.

[Agile Conversations](#) Pragmatic Bookshelf

This guide provides practical insight into the world of software testing, explaining the basic steps of the testing process and how to perform effective tests. It also presents an overview of different techniques, both dynamic and static, and how to apply them.

[Roundtable on Project Management](#) No Starch Press

Although epilepsy is one of the nation's most common neurological disorders, public understanding of it is limited. Many people do not know the causes of epilepsy or what they should do if they see someone having a seizure. Epilepsy is a complex spectrum of disorders that affects an estimated 2.2 million Americans in a variety of ways, and is characterized by unpredictable seizures that differ in type, cause, and severity. Yet living with epilepsy is about much more than just seizures; the disorder is often defined in practical terms, such as challenges in school, uncertainties about social situations and employment, limitations on driving, and questions about independent living. The Institute of Medicine was asked to examine the public health dimensions of the epilepsies, focusing on public health surveillance and data collection; population and public health research;

health policy, health care, and human services; and education for people with the disorder and their families, health care providers, and the public. In *Epilepsy Across the Spectrum*, the IOM makes recommendations ranging from the expansion of collaborative epilepsy surveillance efforts, to the coordination of public awareness efforts, to the engagement of people with epilepsy and their families in education, dissemination, and advocacy for improved care and services. Taking action across multiple dimensions will improve the lives of people with epilepsy and their families. The realistic, feasible, and action-oriented recommendations in this report can help enable short- and long-term improvements for people with epilepsy. For all epilepsy organizations and advocates, local, state, and federal agencies, researchers, health care professionals, people with epilepsy, as well as the public, *Epilepsy Across the Spectrum* is an essential resource.

[The Software Test Engineer's Handbook](#) IT Revolution

You work in a loop: write code, get feedback, iterate. The faster you get feedback, the faster you can learn and become a more effective developer. *Test-Driven React* helps you refine your React workflow to give you the feedback you need as quickly as possible. Write strong tests and run them continuously as you work, split complex code up into manageable pieces, and stay focused on what's important by automating away mundane, trivial tasks. Adopt these techniques and you'll be able to avoid productivity traps and start building React components at a stunning pace!

[Pragmatic Guide to Git](#) Pragmatic Bookshelf

2012 Jolt Award finalist! Pioneering the Future of Software Test Do you need to get it right, too?

Then, learn from Google. Legendary testing expert James Whittaker, until recently a Google testing leader, and two top Google experts reveal exactly how Google tests software, offering brand-new best practices you can use even if you're not quite Google's size...yet! *Breakthrough Techniques You Can Actually Use* Discover 100% practical, amazingly scalable techniques for analyzing risk and planning tests...thinking like real users...implementing exploratory, black box, white box, and acceptance testing...getting usable feedback...tracking issues...choosing and creating tools...testing "Docs & Mocks," interfaces, classes, modules, libraries, binaries, services, and infrastructure...reviewing code and refactoring...using test hooks, presubmit scripts, queues,

continuous builds, and more. With these techniques, you can transform testing from a bottleneck into an accelerator—and make your whole organization more productive!

[Leading Quality](#) Pearson Education

Increase profitability, elevate work culture, and exceed productivity goals through DevOps practices. More than ever, the effective management of technology is critical for business competitiveness. For decades, technology leaders have struggled to balance agility, reliability, and security. The consequences of failure have never been greater—whether it's the healthcare.gov debacle, cardholder data breaches, or missing the boat with Big Data in the cloud. And yet, high performers using DevOps principles, such as Google, Amazon, Facebook, Etsy, and Netflix, are routinely and reliably deploying code into production hundreds, or even thousands, of times per day. Following in the footsteps of *The Phoenix Project*, *The DevOps Handbook* shows leaders how to replicate these incredible outcomes, by showing how to integrate Product Management, Development, QA, IT Operations, and Information Security to elevate your company and win in the marketplace.

["Dear Evil Tester"](#) Pragmatic Bookshelf

This updated and reorganized Fifth edition of *Software Testing: A Craftsman's Approach* applies the strong mathematics content of previous editions to a coherent treatment of software testing. Responding to instructor and student survey input of previous editions, the authors have streamlined chapters and examples. The Fifth Edition: Has a new chapter on feature interaction testing that explores the feature interaction problem and explains how to reduce tests Uses Java instead of pseudo-code for all examples including structured and object-oriented ones Presents model-based development and provides an explanation of how to conduct testing within model-based development environments Explains testing in waterfall, iterative, and agile software development projects 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, Fifth Edition* is sure to become a standard reference for those who need to stay up to date with evolving technologies in software testing. Carrying on the tradition of previous editions, it is a valuable reference for software testers, developers, and engineers.

Best Sellers - Books :

- [The Mountain Is You: Transforming Self-sabotage Into Self-mastery](#)
- [Brown Bear, Brown Bear, What Do You See?](#)
- [Chicka Chicka Boom Boom \(board Book\)](#)
- [Spare](#)
- [Can't Hurt Me: Master Your Mind And Defy The Odds By David Goggins](#)
- [Spare By Prince Harry The Duke Of Sussex](#)
- [It Starts With Us: A Novel \(2\) \(it Ends With Us\) By Colleen Hoover](#)
- [The Alchemist, 25th Anniversary: A Fable About Following Your Dream](#)
- [Guess How Much I Love You By Sam Mcbratney](#)
- [A Court Of Mist And Fury \(a Court Of Thorns And Roses, 2\)](#)