

BRIDGING THE COMMUNICATION GAP SPECIFICATION BY EXAMPLE AND AGILE ACCEPTANCE TESTING GOJKO ADZIC

Right here, we have countless books **BRIDGING THE COMMUNICATION GAP SPECIFICATION BY EXAMPLE AND AGILE ACCEPTANCE TESTING GOJKO ADZIC** and collections to check out. We additionally have enough money variant types and then type of the books to browse. The adequate book, fiction, history, novel, scientific research, as skillfully as various supplementary sorts of books are readily affable here.

As this BRIDGING THE COMMUNICATION GAP SPECIFICATION BY EXAMPLE AND AGILE ACCEPTANCE TESTING GOJKO ADZIC, it ends up inborn one of the favored books BRIDGING THE COMMUNICATION GAP SPECIFICATION BY EXAMPLE AND AGILE ACCEPTANCE TESTING GOJKO ADZIC collections that we have. This is why you remain in the best website to see the amazing books to have.

SAP MM Training Tutorials - Michael M. Martinez 2009-10
SAPCOOKBOOK Training Tutorials Guides are designed to help you understand what you need to know to get started working in SAP. Written from the end-user's perspective, SAPCOOKBOOK Resource Manuals and Training Tutorials guides provide step-by-step instruction on how to execute the critical transactions in each functional area of SAP. This is not a 1000-page encyclopedia filled with obscure configuration items that you will never use - this book shows you what people actually do in the SAP system, and how to do it faster. Michael Martinez will help you understand the document flow and the business process steps you need to work in the exciting area of SAP. SAPCOOKBOOK Training Tutorials Guides are the fastest way to learn SAP, period.

Bridging the Communication Gap - Gojko Adzic 2009
Bridging the Communication Gap is a book about improving communication between customers, business analysts, developers and testers on software projects, especially by using specification by example and agile acceptance testing. These two key emerging software development practices can significantly improve the chances of success of a software project. They ensure that all project participants speak the same language, and build a shared and consistent understanding of the domain. This leads to better specifications, flushes out incorrect assumptions and ensures that functional gaps are discovered before the development starts. With these practices in place you can build software that is genuinely fit for purpose.

Specification by Example - Gojko Adzic 2011-06-02
Summary Specification by Example is an emerging practice for creating software based on realistic examples, bridging the communication gap between business stakeholders and the dev teams building the software. In this book, author Gojko Adzic distills interviews with successful teams worldwide, sharing how they specify, develop, and deliver software, without defects, in short iterative delivery cycles. About the Technology Specification by Example is a collaborative method for specifying requirements and tests. Seven patterns, fully explored in this book, are key to making the method effective. The method has four main benefits: it produces living, reliable documentation; it defines expectations clearly and makes validation efficient; it reduces rework; and, above all, it assures delivery teams and business stakeholders that the software that's built is right for its purpose. About the Book This book distills from the experience of leading teams worldwide effective ways to specify, test, and deliver software in short, iterative delivery cycles. Case studies in this book range from small web startups to large financial institutions, working in many processes including XP, Scrum, and Kanban. This book is written for developers, testers, analysts, and business people working together to build great software. 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. What's Inside Common process patterns How to avoid bad practices Fitting SBE in your process 50+ case studies =====
Table of Contents Part 1 Getting started Part 2 Key process patterns Part 3 Case studies Key benefits Key process patterns Living documentation Initiating the changes Deriving scope from goals Specifying collaboratively Illustrating using examples Refining the specification Automating validation without changing specifications Validating frequently Evolving a documentation system uSwitch RainStor Iowa Student Loan Sabre Airline Solutions ePlan Services Songkick Concluding thoughts

ATDD by Example - Markus Gärtner 2013
With Acceptance Test-Driven Development (ATDD), business customers, testers, and developers can collaborate to produce testable requirements that help them build higher quality software more rapidly. However, ATDD is still widely misunderstood by many practitioners. ATDD by Example is the first practical, entry-level, hands-on guide to implementing and successfully applying it. ATDD pioneer Markus Gärtner walks readers step by step through deriving the right systems from business users, and then implementing fully automated, functional tests that accurately reflect business requirements, are intelligible to stakeholders, and promote more effective development. Through two end-to-end case studies, Gärtner demonstrates how ATDD can be applied using diverse frameworks and languages. Each case study is accompanied by an extensive set of artifacts, including test automation classes, step definitions, and full sample implementations. These realistic examples illuminate ATDD's fundamental principles, show how ATDD fits into the broader development process, highlight tips from Gärtner's extensive experience, and identify crucial pitfalls to avoid. Readers will learn to Master the thought processes associated with successful ATDD implementation Use ATDD with Cucumber to describe software in ways businesspeople can understand Test web pages using ATDD tools Bring ATDD to Java with the FitNesse wiki-based acceptance test framework Use examples more effectively in Behavior-Driven Development (BDD) Specify software collaboratively through innovative workshops Implement more user-friendly and collaborative test automation Test more cleanly, listen to test results, and refactor tests for greater value If you're a tester, analyst, developer, or project manager, this book offers a concrete foundation for achieving real benefits with ATDD now-and it will help you reap even more value as you gain experience.

Living Documentation - Cyrille Martraire 2018-11-14
Use an Approach Inspired by Domain-Driven Design to Build Documentation That Evolves to Maximize Value Throughout Your Development Lifecycle Software documentation can come to life, stay dynamic, and actually help you build better software. Writing for developers, coding architects, and other software professionals, Living Documentation shows how to create

documentation that evolves throughout your entire design and development lifecycle. Through patterns, clarifying illustrations, and concrete examples, Cyrille Martraire demonstrates how to use well-crafted artifacts and automation to dramatically improve the value of documentation at minimal extra cost. Whatever your domain, language, or technologies, you don't have to choose between working software and comprehensive, high-quality documentation: you can have both. · Extract and augment available knowledge, and make it useful through living curation · Automate the creation of documentation and diagrams that evolve as knowledge changes · Use development tools to refactor documentation · Leverage documentation to improve software designs · Introduce living documentation to new and legacy environments
The Nexus Framework for Scaling Scrum - Kurt Bittner
2017-12-04

Improve and Accelerate Software Delivery for Large, Distributed, Complex Projects The Nexus Framework is the simplest, most effective approach to applying Scrum at scale across multiple teams, sites, and time zones. Created by Scrum.org—the pioneering Scrum training and certification organization founded by Scrum co-creator Ken Schwaber—Nexus draws on decades of experience to address the unique challenges teams face in coming together, sharing work, and managing and minimizing dependencies. The Nexus™ Framework for Scaling Scrum is a concise book that shows how Nexus helps teams to deliver a complex, multi-platform, software-based product in short, frequent cycles, without sacrificing consistency or quality, and without adding unnecessary complexity or straying from Scrum's core principles. Using an extended case study, the authors illustrate how Nexus helps teams solve common scaling challenges like reducing cross-team dependencies, preserving team self-organization and transparency, and ensuring accountability. Understand the challenges of delivering working, integrated product increments with multiple teams, and how Nexus addresses them Form a Nexus around a new or existing product and learn how that Nexus sets goals and plans its work Run Sprints within a Nexus, provide transparency into progress, conduct effective Nexus Sprint reviews, and use Nexus Sprint Retrospectives to continuously improve Overcome the distributed team collaboration challenges

Writing Great Specifications - Kamil Nicieja 2017-10-25
Summary Writing Great Specifications is an example-rich tutorial that teaches you how to write good Gherkin specification documents that take advantage of the benefits of specification by example. Foreword written by Gojko Adzic. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology The clearest way to communicate a software specification is to provide examples of how it should work. Turning these story-based descriptions into a well-organized dev plan is another matter. Gherkin is a human-friendly, jargon-free language for documenting a suite of examples as an executable specification. It fosters efficient collaboration between business and dev teams, and it's an excellent foundation for the specification by example (SBE) process. About the Book Writing Great Specifications teaches you how to capture executable software designs in Gherkin following the SBE method. Written for both developers and non-technical team members, this practical book starts with collecting individual feature stories and organizing them into a full, testable spec. You'll learn to choose the best scenarios, write them in a way that anyone can understand, and ensure they can be easily updated by anyone.management. What's Inside Reading and writing Gherkin Designing story-based test cases Team Collaboration Managing a suite of Gherkin documents About the Reader Primarily written for developers and architects, this book is accessible to any member of a

software design team. About the Author Kamil Nicieja is a seasoned engineer, architect, and project manager with deep expertise in Gherkin and SBE. Table of contents Introduction to specification by example and Gherkin PART 1 - WRITING EXECUTABLE SPECIFICATIONS WITH EXAMPLES The specification layer and the automation layer Mastering the Given-When-Then template The basics of scenario outlines Choosing examples for scenario outlines The life cycle of executable specifications Living documentation PART 2 - MANAGING SPECIFICATION SUITES Organizing scenarios into a specification suite Refactoring features into abilities and business needs Building a domain-driven specification suite Managing large projects with bounded contexts
Managing Change in Organizations - Project Management Institute 2013-08-01

Managing Change in Organizations: A Practice Guide is unique in that it integrates two traditionally disparate world views on managing change: organizational development/human resources and portfolio/program/project management. By bringing these together, professionals from both worlds can use project management approaches to effectively create and manage change. This practice guide begins by providing the reader with a framework for creating organizational agility and judging change readiness.

The New Turing Omnibus - A. K. Dewdney 2001

"No other volume provides as broad, as thorough, or as accessible an introduction to the realm of computers as A. K. Dewdney's *The Turing Omnibus*. Updated and expanded, *The Turing Omnibus* offers 66 concise, brilliantly written articles on the major points of interest in computer science theory, technology, and applications. New for this tour: updated information on algorithms, detecting primes, noncomputable functions, and self-replicating computers--plus completely new sections on the Mandelbrot set, genetic algorithms, the Newton-Raphson Method, neural networks that learn, DOS systems for personal computers, and computer viruses." -- Book cover.

Secure by Design - Daniel Sawano 2019-09-03

Summary *Secure by Design* teaches developers how to use design to drive security in software development. This book is full of patterns, best practices, and mindsets that you can directly apply to your real world development. You'll also learn to spot weaknesses in legacy code and how to address them. About the technology Security should be the natural outcome of your development process. As applications increase in complexity, it becomes more important to bake security-mindedness into every step. The secure-by-design approach teaches best practices to implement essential software features using design as the primary driver for security. About the book *Secure by Design* teaches you principles and best practices for writing highly secure software. At the code level, you'll discover security-promoting constructs like safe error handling, secure validation, and domain primitives. You'll also master security-centric techniques you can apply throughout your build-test-deploy pipeline, including the unique concerns of modern microservices and cloud-native designs. What's inside *Secure-by-design* concepts Spotting hidden security problems Secure code constructs Assessing security by identifying common design flaws Securing legacy and microservices architectures About the reader Readers should have some experience in designing applications in Java, C#, .NET, or a similar language. About the author Dan Bergh Johnson, Daniel Deogun, and Daniel Sawano are acclaimed speakers who often present at international conferences on topics of high-quality development, as well as security and design.

Lean-agile Acceptance Test-driven Development - Kenneth Pugh 2011

How to scale ATDD to large projects --

Scaling Lean & Agile Development - Craig Larman
2008-12-08

Lean Development and Agile Methods for Large-Scale Products: Key Thinking and Organizational Tools for Sustainable Competitive Success Increasingly, large product-development organizations are turning to lean thinking, agile principles and practices, and large-scale Scrum to sustainably and quickly deliver value and innovation. However, many groups have floundered in their practice-oriented adoptions. Why? Because without a deeper understanding of the thinking tools and profound organizational redesign needed, it is as though casting seeds on to an infertile field. Now, drawing on their long experience leading and guiding large-scale lean and agile adoptions for large, multisite, and offshore product development, and drawing on the best research for great team-based agile organizations, internationally recognized consultant and best-selling author Craig Larman and former leader of the agile transformation at Nokia Networks Bas Vodde share the key thinking and organizational tools needed to plant the seeds of product development success in a fertile lean and agile enterprise. Coverage includes Lean thinking and development combined with agile practices and methods Systems thinking Queuing theory and large-scale development processes Moving from single-function and component teams to stable cross-functional cross-component Scrum feature teams with end-to-end responsibility for features Organizational redesign to a lean and agile enterprise that delivers value fast Large-scale Scrum for multi-hundred-person product groups In a competitive environment that demands ever-faster cycle times and greater innovation, applied lean thinking and agile principles are becoming an urgent priority. Scaling Lean & Agile Development will help leaders create the foundation for their lean enterprise—and deliver on the significant benefits of agility. In addition to the foundation tools in this text, see the companion book Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum for complementary action tools.

Domain-driven Design - Eric Evans 2004

"Domain-Driven Design" incorporates numerous examples in Java-case studies taken from actual projects that illustrate the application of domain-driven design to real-world software development.

Impact Mapping - Gojko Adzic 2012-10

A practical guide to impact mapping, a simple yet incredibly effective method for collaborative strategic planning that helps organizations make an impact with software.

Three Pillars of Agile Quality & Testing: Achieving Balanced Results in Your Journey Towards Agile Quality - Robert Galen 2015-01-24

There are a few books on the market that discuss agile testing from a practitioner perspective. But this is the first book that looks at the organizational moves that are required to pull together an effective Agile Quality and Testing strategy. One that shows leaders and coaches how to effectively establish agile practices using the Three Pillars model. The book is chock-full of real world stories from two coaches who

Run Grow Transform - Steven Bell 2017-07-27

Today's customers want it all and they want it now: innovation, speed, agility, and value. How can you drive operational excellence, stimulate growth, and accelerate idea-to-value innovation throughout your enterprise? Shingo Prize-winning author Steve Bell, joined by other thought leaders, offers useful insights and examples you can start using now. Run Grow Transform takes the next logical step to driving enterprise value. This could be the game-changing playbook for IT 3.0. -Mark Katz, CIO & Senior Vice President, Esselte Corporation A powerful read detailing how companies can leverage their Lean IT

transformation to supercharge the business. -Tom Paider, IT Build Capability Leader, Nationwide The consistent application of the practices described in this book has enabled Embraer to reap huge gains. I recommend this book as a desktop companion. -Alexandre Baule, Vice President Information Systems, Embraer Run Grow Transform takes the reader a leap forward, ready for immediate application to bridge Lean and innovation. -Melissa Barrett, Enterprise Architecture & IT Strategy, Premera Blue Cross This book focuses on the most critical and challenging issue for any aspect of the development or use of IT: creating a collaborative learning culture. -Jeffrey K. Liker, Shingo Prize-Winning Author of The Toyota Way Run Grow Transform sets out the principles and practices necessary for success in the new economy. -Jez Humble, author of Continuous Delivery Steve Bell has hit another home run with this book. Either your organization will adopt the wisdom contained in this book and thrive, or your competitors will do so and put you out of business. -Scott Ambler, author Disciplined Agile Delivery and 19 other books It's rare to see truly new insight added to the Lean discussion. Steve Bell does just that by continuing to push the frontiers of Lean thinking. -Alexander Brown, COO, Scrum Inc. A powerful read detailing how companies can leverage their Lean IT transformation to supercharge the business. -Tom Paider, IT Build Capability Leader, Nationwide Steve Bell has mapped a new trajectory. I challenge any CIO to read Bell's breakthrough work and not be compelled to start this journey to become a transformative leader in the creation of real and sustainable value. -Jeffrey Barnes, Society for Information Management (SIM), Regional Director, Advanced Practices Council All too often the IT organization is viewed as an impediment to lean transformation, when it truly can be a catalyst. Steve's book sorts out all the noise, the jargon, and the "hero culture", guiding the reader to what is so obvious, yet so hard to see: build your culture around your customer! -Josh Rapoza, Director of Web Strategy and Operations, Lean Enterprise Institute Aligning Lean and IT is a great challenge with a big payoff. This book really shows how Lean and IT can create a strong enterprise; it's a great inspiration. -Klaus Lyck Petersen, Solar A/S A must read for any organization that is pursuing continuous improvement. In today's world, real business improvement cannot be achieved without the IT factor; this book will help any organization achieve the improvement that they are seeking. -Barry J. Brunetto, Vice President, Information Systems, Blount International Precise, concise, and entertaining, this book provides the reader with crucial tips on how IT can help enterprises survive and thrive in a fast-paced technological and economic environment. This is mandatory reading not only for businesses and IT organizations, but also for universities and policy makers. -Fuat Alican, PhD, Vice President, Central American Scientific Research and Education Center Not just for Lean Practitioners, Run Grow Transform is a must-have reference for any IT organization, regardless of size, age or industry, looking to move to the next level of performance. -Sarah Topham, Lean Deployment Leader, Information Technology & Product Management, Paychex, Inc. This is a long overdue book that addresses the key challenges for today's IT organization and puts Lean IT into a context that is too often lacking. -James Finister, Tata Consultancy Services The 'business as usual' scenario is not an option in today's economy and global challenges. A transformation in methods, tools and frameworks is needed to guide our business decisions. This book is your first step! -Khuloud Odeh, IT Director, Grameen Foundation Delivering beyond the helpful folk wisdom and narrow techniques and technologies found elsewhere, Steve Bell and his contributors provide practical full value stream

lifecycle methods for continuous improvement using Lean in an IT and customer (business) setting. -Martin Erb, Director of Professional Services, Pink Elephant Run Grow Transform clarifies the eternal quest of IT: to simply "running" of the business and to create innovative solutions to grow the business and create sustainable competitive advantage to transform the way customers interact with your business, in plain, actionable advice from one who has been on the front lines. -Tom Foco, Value Stream Solution

Agile Extension to the BABOK Guide, Version 2 - Iiba 2017-09-22

The Agile Extension to the BABOK(R) Guide (Agile Extension) version 2 describes the benefits, activities, tasks, skills, and practices required for effective agile business analysis with a constant focus on delivering business value. The Agile Extension version 2: describes the agile mindset and positions agile business analysis beyond software development introduces a 3-tier rolling planning model to help organizations, teams, and practitioners deliver greater business value incorporates the Business Analysis Core Concept Model(TM) (BACCM(TM)) details the seven principles of agile business analysis The Agile Extension to the BABOK(R) Guide is an ongoing initiative of Agile Alliance and the International Institute of Business Analysis(TM) (IIBA(R)) since 2009. The Agile Extension provides guidance for Agile practitioners or anyone interested in leveraging effective Agile business analysis to create better business outcomes that add real business and customer value

Explore It! - Elisabeth Hendrickson 2013-02-21

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.

Developer Testing - Alexander Tarlinder 2016-09-07

How do successful agile teams deliver bug-free, maintainable software—iteration after iteration? The answer is: By seamlessly combining development and testing. On such teams, the developers write testable code that enables them to verify it using various types of automated tests. This approach keeps regressions at bay and prevents “testing crunches”—which otherwise may

occur near the end of an iteration—from ever happening. Writing testable code, however, is often difficult, because it requires knowledge and skills that cut across multiple disciplines. In *Developer Testing*, leading test expert and mentor Alexander Tarlinder presents concise, focused guidance for making new and legacy code far more testable. Tarlinder helps you answer questions like: When have I tested this enough? How many tests do I need to write? What should my tests verify? You'll learn how to design for testability and utilize techniques like refactoring, dependency breaking, unit testing, data-driven testing, and test-driven development to achieve the highest possible confidence in your software. Through practical examples in Java, C#, Groovy, and Ruby, you'll discover what works—and what doesn't. You can quickly begin using Tarlinder's technology-agnostic insights with most languages and toolsets while not getting buried in specialist details. The author helps you adapt your current programming style for testability, make a testing mindset “second nature,” improve your code, and enrich your day-to-day experience as a software professional. With this guide, you will Understand the discipline and vocabulary of testing from the developer's standpoint Base developer tests on well-established testing techniques and best practices Recognize code constructs that impact testability Effectively name, organize, and execute unit tests Master the essentials of classic and “mockist-style” TDD Leverage test doubles with or without mocking frameworks Capture the benefits of programming by contract, even without runtime support for contracts Take control of dependencies between classes, components, layers, and tiers Handle combinatorial explosions of test cases, or scenarios requiring many similar tests Manage code duplication when it can't be eliminated Actively maintain and improve your test suites Perform more advanced tests at the integration, system, and end-to-end levels Develop an understanding for how the organizational context influences quality assurance Establish well-balanced and effective testing strategies suitable for agile teams

Domain Storytelling - Stefan Hofer 2021-09-27

Storytelling is at the heart of human communication--why not use it to overcome costly misunderstandings when designing software? By telling and visualising stories, domain experts and team members make business processes and domain knowledge tangible. *Domain Storytelling* enables everyone to understand the relevant people, activities, and work items. With this guide, the method's inventors explain how domain experts and teams can work together to capture insights with simple pictographs, show their work, solicit feedback, and get everyone on the same page. Stefan Hofer and Henning Schwentner introduce the methods easy pictographic language, scenario-based modeling techniques, workshop format, and relationship to other modeling methods. Using step-by-step case studies, they guide you through solving many common problems: Fully align all project participants and stakeholders, both technical and business-focused Master a simple set of symbols and rules for modeling any process or workflow Use workshop-based collaborative modeling to find better solutions faster Draw clear boundaries to organise your domain, software, and teams Transform domain knowledge into requirements, embedded naturally into an agile process Move your models from diagrams and sticky notes to code Gain better visibility into your IT landscape so you can consolidate or optimise it This guide is for everyone who wants more effective software--from developers, architects, and team leads to the domain experts, product owners, and executives who rely on it every day. **Implementing Domain-driven Design** - Vaughn Vernon 2013 Vaughn Vernon presents concrete and realistic domain-driven design (DDD) techniques through examples from familiar domains, such as a Scrum-based project

management application that integrates with a collaboration suite and security provider. Each principle is backed up by realistic Java examples, and all content is tied together by a single case study of a company charged with delivering a set of advanced software systems with DDD.

Lessons Learned in Software Testing - Cem Kaner

2011-08-02

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

Domain-Driven Design Quickly - Floyd Marinescu

2007-12-01

Domain Driven Design is a vision and approach for dealing with highly complex domains that is based on making the domain itself the main focus of the project, and maintaining a software model that reflects a deep understanding of the domain. This book is a short, quickly-readable summary and introduction to the fundamentals of DDD; it does not introduce any new concepts; it attempts to concisely summarize the essence of what DDD is, drawing mostly Eric Evans' original book, as well other sources since published such as Jimmy Nilsson's *Applying Domain Driven Design*, and various DDD discussion forums. The main topics covered in the book include: Building Domain Knowledge, The Ubiquitous Language, Model Driven Design, Refactoring Toward Deeper Insight, and Preserving Model Integrity. Also included is an interview with Eric Evans on Domain Driven Design today.

Adaptive Code - Gary McLean Hall 2017-04-18

Write code that can adapt to changes. By applying this book's principles, you can create code that accommodates new requirements and unforeseen scenarios without significant rewrites. Gary McLean Hall describes Agile best practices, principles, and patterns for designing and writing code that can evolve more quickly and easily, with fewer errors, because it doesn't impede change. Now revised, updated, and expanded, *Adaptive Code, Second Edition* adds indispensable practical insights on Kanban, dependency inversion, and creating reusable abstractions. Drawing on over a decade of Agile consulting and development experience, McLean Hall has updated his best-seller with deeper coverage of unit testing, refactoring, pure dependency injection, and more. Master powerful new ways to:

- Write code that enables and complements Scrum, Kanban, or any other Agile framework
- Develop code that can survive major changes in requirements
- Plan for adaptability by using dependencies, layering, interfaces, and design patterns
- Perform unit testing and refactoring in tandem, gaining more value from both
- Use the "golden master" technique to make legacy code adaptive
- Build SOLID code with single-responsibility, open/closed, and Liskov substitution principles
- Create smaller interfaces to support more-diverse client and architectural needs

Leverage dependency injection best practices to improve code adaptability • Apply dependency inversion with the Stairway pattern, and avoid related anti-patterns About You This book is for programmers of all skill levels seeking more-practical insight into design patterns, SOLID principles, unit testing, refactoring, and related topics. Most readers will have programmed in C#, Java, C++, or similar object-oriented languages, and will be familiar with core procedural programming techniques.

Practices for Scaling Lean & Agile Development - Craig Larman 2010-01-26

Lean and Agile Development for Large-Scale Products: Key Practices for Sustainable Competitive Success

Increasingly, large product-development organizations are turning to lean thinking, agile principles and practices, and large-scale Scrum to sustainably and quickly deliver value and innovation. Drawing on their long experience leading and guiding lean and agile adoptions for large, multisite, and offshore product development, internationally recognized consultant and best-selling author Craig Larman and former leader of the agile transformation at Nokia Networks Bas Vodde share the key action tools needed for success. Coverage includes Frameworks for large-scale Scrum for multihundred-person product groups Testing and building quality in Product management and the end of the "contract game" between business and R&D Envisioning a large release, and planning for multiteam development Low-quality legacy code: why it's created, and how to stop it Continuous integration in a large multisite context Agile architecting Multisite or offshore development Contracts and outsourced development In a competitive environment that demands ever-faster cycle times and greater innovation, the practices inspired by lean thinking and agile principles are ever-more relevant. *Practices for Scaling Lean & Agile Development* will help people realize a lean enterprise—and deliver on the significant benefits of agility. In addition to the action tools in this text, see the companion book *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum* for complementary foundation tools.

Cucumber Recipes - Ian Dees 2013-02-12

You can test just about anything with Cucumber. We certainly have, and in *Cucumber Recipes* we'll show you how to apply our hard-won field experience to your own projects. Once you've mastered the basics, this book will show you how to get the most out of Cucumber--from specific situations to advanced test-writing advice. With over forty practical recipes, you'll test desktop, web, mobile, and server applications across a variety of platforms. This book gives you tools that you can use today to automate any system that you encounter, and do it well. The *Cucumber Book* showed you how your team can work together to write executable specifications--documents that tell a clear story and also happen to be working test code. We'll arm you with ready-rolled solutions to real-world problems: your tests will run faster, read more clearly, and work in any environment. Our first tips will help you fit Cucumber into your workflow. Powerful filters will tame tables full of test data, transforming them into the format your application needs. Custom output formatters will generate reports for any occasion. Continuous Integration servers will run your Cucumber tests every time the code changes. Next, you'll find recipes tailored to the platform you're running on. Ever wanted to know how to test a Grails app from Cucumber? Need to put a Windows program through its paces? How about a mobile app running on Android or iOS? We'll show you how to do all of these. Throughout the book, you'll see how to make Cucumber sing as you interoperate with different platforms, languages, and environments. From embedded circuits to Python and PHP web apps, Cucumber has something for you. What You Need: You'll need basic working knowledge of

Cucumber and Ruby. Individual recipes may have additional requirements; for example, a recipe on Windows automation might pull in an open source GUI driver. We've written the recipes for compatibility with Ruby 1.9.3 and 1.8.7, plus Cucumber 1.1.4. Other versions may work as well, but these are the ones we test with.

Agile Practice Guide (Hindi) - 2019-08-05

Agile Practice Guide – First Edition has been developed as a resource to understand, evaluate, and use agile and hybrid agile approaches. This practice guide provides guidance on when, where, and how to apply agile approaches and provides practical tools for practitioners and organizations wanting to increase agility. This practice guide is aligned with other PMI standards, including A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition, and was developed as the result of collaboration between the Project Management Institute and the Agile Alliance.

Managing Digital - Charles Betz

About This Book This book, "Managing Digital: Concepts and Practices", is intended to guide a practitioner through the journey of building a digital-first viewpoint and the skills needed to thrive in the digital-first world. As such, this book is a bit of an experiment for The Open Group; it isn't structured as a traditional standard or guide. Instead, it is structured to show the key issues and skills needed at each stage of the digital journey, starting with the basics of a small digital project, eventually building to the concerns of a large enterprise. So, feel free to digest this book in stages – the section Introduction for the student is a good guide. The book is intended for both academic and industry training purposes. This book seeks to provide guidance for both new entrants into the digital workforce and experienced practitioners seeking to update their understanding on how all the various themes and components of IT management fit together in the new world. About The Open Group Press The Open Group Press is an imprint of The Open Group for advancing knowledge of information technology by publishing works from individual authors within The Open Group membership that are relevant to advancing The Open Group mission of Boundaryless Information Flow™. The key focus of The Open Group Press is to publish high-quality monographs, as well as introductory technology books intended for the general public, and act as a complement to The Open Group Standards, Guides, and White Papers. The views and opinions expressed in this book are those of the author, and do not necessarily reflect the consensus position of The Open Group members or staff.

Agile in a Flash - Jeff Langr 2011

Real agilists don't weigh themselves down with libraries of books, they keep their important information handy with them at all times. Jeff and Tim pack over two decades of experience coaching and doing agile into Agile in a Flash, a unique deck of index cards that fit neatly in your pocket and tack easily onto the wall. Agile in a Flash cards run the gamut of agile, covering customer, planning, team, and developer concepts to help you succeed on agile projects. You can use cards from the deck in many ways: as references, reminders, teaching tools, and conversation pieces. Why not get sets for your entire team or organization? This comprehensive set of cards is an indispensable resource for agile teams. The deck of Agile in a Flash cards teaches leadership, teamwork, clean programming, agile approaches to problem solving, and tips for coaching agile teams. Team members can use the cards as reference material, ice breakers for conversations, reminders (taped to a wall or monitor), and sources of useful tips and hard-won wisdom. The cards are: Bite-sized! Read one practice or aspect at a time in a couple of minutes. Smart! Each card has years of practical experience behind it. Portable! Cards fit easily in your pocket or

backpack. An indispensable tool for any agile team, and a must-have for every agile coach or Scrum Master. The Agile in a Flash deck is broken into four areas: planning, team, coding, and agile concepts. The front of each card is a quick list - a summary of the things you want to know and remember. The back provides further detail on each of the bullet points, and offers sage nuggets of knowledge based on extensive professional experience. Tape the cards to your wall, stick them on your monitor, and get agile fast.

Applying Domain-Driven Design and Patterns - Jimmy Nilsson 2006-05-08

Patterns, Domain-Driven Design (DDD), and Test-Driven Development (TDD) enable architects and developers to create systems that are powerful, robust, and maintainable. Now, there's a comprehensive, practical guide to leveraging all these techniques primarily in Microsoft .NET environments, but the discussions are just as useful for Java developers. Drawing on seminal work by Martin Fowler (Patterns of Enterprise Application Architecture) and Eric Evans (Domain-Driven Design), Jimmy Nilsson shows how to create real-world architectures for any .NET application. Nilsson illuminates each principle with clear, well-annotated code examples based on C# 1.1 and 2.0. His examples and discussions will be valuable both to C# developers and those working with other .NET languages and any databases—even with other platforms, such as J2EE. Coverage includes · Quick primers on patterns, TDD, and refactoring · Using architectural techniques to improve software quality · Using domain models to support business rules and validation · Applying enterprise patterns to provide persistence support via NHibernate · Planning effectively for the presentation layer and UI testing · Designing for Dependency Injection, Aspect Orientation, and other new paradigms

Fifty Quick Ideas to Improve Your User Stories - Gojko Adzic 2014-10-15

This book will help you write better stories, spot and fix common issues, split stories so that they are smaller but still valuable, and deal with difficult stuff like crosscutting concerns, long-term effects and non-functional requirements. Above all, this book will help you achieve the promise of agile and iterative delivery: to ensure that the right stuff gets delivered through productive discussions between delivery team members and business stakeholders. Who is this book for? This is a book for anyone working in an iterative delivery environment, doing planning with user stories. The ideas in this book are useful both to people relatively new to user stories and those who have been working with them for years. People who work in software delivery, regardless of their role, will find plenty of tips for engaging stakeholders better and structuring iterative plans more effectively. Business stakeholders working with software teams will discover how to provide better information to their delivery groups, how to set better priorities and how to outrun the competition by achieving more with less software. What's inside? Unsurprisingly, the book contains exactly fifty ideas. They are grouped into five major parts: - Creating stories: This part deals with capturing information about stories before they get accepted into the delivery pipeline. You'll find ideas about what kind of information to note down on story cards and how to quickly spot potential problems. - Planning with stories: This part contains ideas that will help you manage the big-picture view, set milestones and organise long-term work. - Discussing stories: User stories are all about effective conversations, and this part contains ideas to improve discussions between delivery teams and business stakeholders. You'll find out how to discover hidden assumptions and how to facilitate effective conversations to ensure shared understanding. - Splitting stories: The ideas in this part will help

you deal with large and difficult stories, offering several strategies for dividing them into smaller chunks that will help you learn fast and deliver value quickly.

- Managing iterative delivery: This part contains ideas that will help you work with user stories in the short and mid term, manage capacity, prioritise and reduce scope to achieve the most with the least software. About the authors: Gojko Adzic is a strategic software delivery consultant who works with ambitious teams to improve the quality of their software products and processes. Gojko's book Specification by Example was awarded the #2 spot on the top 100 agile books for 2012 and won the Jolt Award for the best book of 2012. In 2011, he was voted by peers as the most influential agile testing professional, and his blog won the UK agile award for the best online publication in 2010. David Evans is a consultant, coach and trainer specialising in the field of Agile Quality. David helps organisations with strategic process improvement and coaches teams on effective agile practice. He is regularly in demand as a conference speaker and has had several articles published in international journals.

Object Design Style Guide - Matthias Noback 2019-12-23 "Demystifies object-oriented programming, and lays out how to use it to design truly secure and performant applications." -Charles Soetan, Plum.io Key Features Dozens of techniques for writing object-oriented code that's easy to read, reuse, and maintain Write code that other programmers will instantly understand Design rules for constructing objects, changing and exposing state, and more Examples written in an instantly familiar pseudocode that's easy to apply to Java, Python, C#, and any object-oriented language Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About The Book Well-written object-oriented code is easy to read, modify, and debug. Elevate your coding style by mastering the universal best practices for object design presented in this book. These clearly presented rules, which apply to any OO language, maximize the clarity and durability of your codebase and increase productivity for you and your team. In Object Design Style Guide, veteran developer Matthias Noback lays out design rules for constructing objects, defining methods, and much more. All examples use instantly familiar pseudocode, so you can follow along in the language you prefer. You'll go case by case through important scenarios and challenges for object design and then walk through a simple web application that demonstrates how different types of objects can work together effectively. What You Will Learn Universal design rules for a wide range of objects Best practices for testing objects A catalog of common object types Changing and exposing state Test your object design skills with exercises This Book Is Written For For readers familiar with an object-oriented language and basic application architecture. About the Author Matthias Noback is a professional web developer with nearly two decades of experience. He runs his own web development, training, and consultancy company called "Noback's Office." Table of Contents: 1 | Programming with objects: A primer 2 | Creating services 3 | Creating other objects 4 | Manipulating objects 5 | Using objects 6 | Retrieving information 7 | Performing tasks 8 | Dividing responsibilities 9 | Changing the behavior of services 10 | A field guide to objects 11 | Epilogue

Beyond Requirements - Kent J. McDonald 2015-08-29 Satisfy Stakeholders by Solving the Right Problems, in the Right Ways In Beyond Requirements , Kent J. McDonald shows how applying analysis techniques with an agile mindset can radically transform analysis from merely "gathering and documenting requirements" to an important activity teams use to build shared understanding. First, McDonald discusses the unique agile mindset, reviews the key principles underlying it, and shows how these

principles link to effective analysis. Next, he puts these principles to work in four wide-ranging and thought-provoking case studies. Finally, he drills down on a full set of techniques for effective agile analysis, using examples to show how, why, and when they work. McDonald's strategies will teach you how to understand stakeholders' needs, identify the best solution for satisfying those needs, and build a shared understanding of your solution that persists throughout the product lifecycle. He also demonstrates how to iterate your analysis, taking advantage of what you learn throughout development, testing, and deployment so that you can continuously adapt, refine, and improve. Whether you're an analysis practitioner or you perform analysis tasks as a developer, manager, or tester, McDonald's techniques will help your team consistently find and deliver better solutions. Coverage includes Core concepts for analysis: needs/ solutions, outcome/output, discovery/delivery Adapting Lean Startup ideas for IT projects: customer delivery, build-measure-learn, and metrics Structuring decisions, recognizing differences between options and commitments, and overcoming cognitive biases Focusing on value: feature injection, minimum viable products, and minimum marketable features Understanding how analysis flows alongside your project's lifecycle Analyzing users: mapping stakeholders, gauging commitment, and creating personas Understanding context: performing strategy (enterprise) analysis Clarifying needs: applying decision filters, assessing project opportunities, stating problems Investigating solutions: impact and story mapping, collaborative modeling, and acceptance criteria definition Kent J. McDonald uncovers better ways of delivering value. His experience includes work in business analysis, strategic planning, project management, and product development in the financial services, health insurance, performance marketing, human services, nonprofit, and automotive industries. He has a BS in industrial engineering from Iowa State University and an MBA from Kent State University. He is coauthor of Stand Back and Deliver: Accelerating Business Agility (Addison-Wesley, 2009).

Complete Guide to Test Automation - Arnon Axelrod 2018-09-22

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 team. Leverage 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. *Modern C++ Programming with Test-Driven Development* - Jeff Langr 2013-10-10

If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD--until now. In this book, Jeff Langr gives you hands-on lessons in the challenges and rewards of doing TDD in C++. *Modern C++ Programming With Test-Driven Development*, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++. What You Need A C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include: cURL JsonCpp Boost (filesystem, date_time/gregorian, algorithm, assign) Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp.

Agile Analytics - Ken Collier 2012

Using Agile methods, you can bring far greater innovation, value, and quality to any data warehousing (DW), business intelligence (BI), or analytics project. However, conventional Agile methods must be carefully adapted to address the unique characteristics of DW/BI projects. In *Agile Analytics*, Agile pioneer Ken Collier shows how to do just that. Collier introduces platform-agnostic Agile solutions for integrating infrastructures consisting of diverse operational, legacy, and specialty systems that mix commercial and custom code. Using

working examples, he shows how to manage analytics development teams with widely diverse skill sets and how to support enormous and fast-growing data volumes. Collier's techniques offer optimal value whether your projects involve "back-end" data management, "front-end" business analysis, or both. Part I focuses on Agile project management techniques and delivery team coordination, introducing core practices that shape the way your Agile DW/BI project community can collaborate toward success. Part II presents technical methods for enabling continuous delivery of business value at production-quality levels, including evolving superior designs; test-driven DW development; version control; and project automation. Collier brings together proven solutions you can apply right now--whether you're an IT decision-maker, data warehouse professional, database administrator, business intelligence specialist, or database developer. With his help, you can mitigate project risk, improve business alignment, achieve better results--and have fun along the way.

Test Driven .NET Development with FitNesse - Gojko Adzic 2008-02-01

Test Driven .NET Development with FitNesse takes you on a journey through the wonderful world of FitNesse, a great web-based tool for software acceptance testing. FitNesse enables software developers and business people to build a shared understanding of the domain and helps produce software that is genuinely fit for purpose.

Lessons in Agile Management - David J. Anderson 2012-06-01

The Missing Link in the Evolution of Kanban -- From Its Roots in Agile David J. Anderson developed the Kanban Method over years spent managing and coaching Agile development teams, at companies such as Sprint and Microsoft, by integrating Lean thinking with Agile principles and practices. This compendium of anecdotes and epiphanies shares this journey on the road to Kanban, now a popular method for improving predictability while managing change and risk in organizations worldwide. Topics include: -Why people resist change -The role of the manager in Agile development -Flow and variability -Timeboxes and delivery cadence -Estimation and metrics

More Agile Testing - Janet Gregory 2014-09-30

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 The eBook edition of More Agile Testing also is available as part of a two-eBook collection, The Agile Testing Collection (9780134190624).

Discovery - Gáspár Nagy 2018-02-07

Written by the creator of SpecFlow and the author of The Cucumber for Java Book, this book provides inside information on how to get the most out of the discovery phase of Behaviour Driven Development (BDD). This practical guide demonstrates good collaboration techniques, illustrated by concrete examples. This book

is written for everyone involved in the specification and delivery of software (including product owners, business analysts, developers, and testers). The book starts by explaining the reasons BDD exists in the first place and describes techniques for getting the most out of collaboration between business and delivery team members. This is the first in the BDD Books series that will guide you through the entire development process, including specific technical practices needed to successfully drive development using collaboratively-authored specifications and living documentation.