Bdd Confusion Using Behaviour Driven Development

Cognitive-Behavioral Therapy for Body Dysmorphic Disorder
BDD Confusion
Learning Behavior-driven Development with JavaScript
Unit Testing Principles, Practices, and Patterns
JUnit Recipes
Body Dysmorphic Disorder
The RSpec Book
The Art of Unit Testing
Event-Driven Architecture in Golang
More Agile Testing
Programming JavaScript Applications
JavaScript Testing with Jasmine
Writing Effective Use Cases
Five Lines of Code
Know What You're FOR
Specification by Example
The Professional Scrum Master Guide
Discovery
Practical Object-oriented Design in Ruby
BDD in Action, Second Edition
BDD in Action
Web Development with Node and Express
Behavior-Driven Development
Learn Microservices with Spring Boot
Sexual Obsessions in Obsessive-Compulsive Disorder
Pro-environmental Behaviors

Object Design
Overcoming Harm OCD
Body Dysmorphic Disorder
Java Testing with Spock
The Cucumber Book
97 Things Every Programmer Should Know
Domain-driven Design
ATDD by Example
Composing Software
Behavior-Driven Development with Cucumber
Cognitive Behavioural Processes Across Psychological Disorders
Planning Extreme Programming
Living Documentation
Developer Testing

BRAYLON ALEX

Cognitive-Behavioral Therapy for Body Dysmorphic Disorder "O'Reilly Media, Inc." Five Lines of Code teaches refactoring that's focused on concrete rules and getting any method down to five lines or less! There’s no jargon or tricky automated-testing skills required, just easy guidelines and patterns illustrated by detailed code samples. In Five Lines of Code you will learn: The signs of bad code Improving code safely, even when you don’t understand it Balancing optimization and code generality Proper compiler practices The Extract method, Introducing Strategy pattern, and many other refactoring patterns Writing stable code that enables change-by-addition Writing code that needs no comments Real-world practices for great refactoring Improving existing code—refactoring—is one of the most common tasks you’ll face as a programmer. Five Lines of Code teaches you clear and actionable refactoring rules that you can apply without relying on intuitive judgements such as “code smells.” Following the author’s expert perspective—that refactoring and code smells can be learned by following a concrete set of principles—you’ll learn when to refactor your code, what patterns to apply to what problem, and the code characteristics that indicate it’s time for a rework. Foreword by Robert C. Martin. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Every codebase includes

mistakes and inefficiencies that you need to find and fix. Refactor the right way, and your code becomes elegant, easy to read, and easy to maintain. In this book, you'll learn a unique approach to refactoring that implements any method in five lines or fewer. You'll also discover a secret most senior devs know: sometimes it's quicker to hammer out code and fix it later! About the book Five Lines of Code is a fresh look at refactoring for developers of all skill levels. In it, you'll master author Christian Clausen's innovative approach, learning concrete rules to get any method down to five lines—or less! You'll learn when to refactor, specific refactoring patterns that apply to most common problems, and characteristics of code that should be deleted altogether. What's inside The signs of bad code Improving code safely, even when you don't understand it Balancing optimization and code generality Proper compiler practices About the reader For developers of all skill levels. Examples use easy-to-read Typescript, in the same style as Java and C#. About the author Christian Clausen works as a Technical Agile Coach, teaching teams how to refactor code. Table of Contents 1 Refactoring refactoring 2 Looking under the hood of refactoring PART 1 LEARN BY REFACTORING A COMPUTER GAME 3 Shatter long function 4 Make type codes work 5 Fuse similar code together 6 Defend the data PART 2 TAKING WHAT YOU HAVE LEARNED INTO THE REAL WORLD 7 Collaborate with the compiler 8 Stay away from comments 9 Love deleting code 10 Never be afraid to add code 11 Follow the structure in the code 12 Avoid optimizations and generality 13 Make bad code look bad 14 Wrapping up

BDD Confusion "O'Reilly Media, Inc." "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.

Learning Behavior-driven Development with JavaScript Pearson Education

Don't let your thoughts and fears define you. In Overcoming Harm OCD, psychotherapist Jon Hershfield offers powerful cognitive behavioral therapy (CBT) and mindfulness tools to help you break free from the pain and self-doubt caused by harm OCD. Do you suffer from violent, unwanted thoughts and a crippling fear of harming others? Are you afraid to seek treatment for fear of being judged? If so, you may have harm OCD—an anxiety disorder associated with obsessive-compulsive disorder (OCD). First and foremost, you need to know that these thoughts do not define you as a human being. But they can cause a lot of real emotional pain. So, how can you overcome harm OCD and start living a better life? Written by an expert in treating harm OCD, this much-needed book offers a direct and comprehensive explanation of what harm OCD is and how to manage it. You'll learn why you have unwanted thoughts, how to identify mental compulsions, and find an overview of cognitive-behavioral and mindfulness-based treatment approaches that can help you reclaim your life. You'll also find tips for disclosing violent obsessions, finding adequate professional help, and working with loved ones to address harm OCD systemically. And finally, you'll learn that your thoughts are just thoughts, and that they don't make you a bad person. If you have harm OCD, it's time to move past the stigma and start focusing on solutions. This evidence-based guide will help light

the way.

Unit Testing Principles, Practices, and Patterns Zondervan

This landmark book is the first comprehensive edited volume on body dysmorphic disorder (BDD), a common and severe disorder. People with BDD are preoccupied with distressing or impairing preoccupations with non-existent or slight defects in their physical appearance. People with BDD think that they look ugly -- even monstrous -- although they look normal to others. BDD often derails sufferers' lives and can lead to suicide. BDD has been described around the world since the 1800s but was virtually unknown and unstudied until only several decades ago. Since then, research on BDD has dramatically increased understanding of this often-debilitating condition. Only recently, BDD was considered untreatable, but today, most sufferers can be successfully treated. This is the only book that provides comprehensive, in-depth, up-to-date information on BDD's clinical features, history, classification, epidemiology, morbidity, features in special populations, diagnosis and assessment, etiology and pathophysiology, treatment, and relationship to other disorders. Numerous chapters focus on cosmetic treatment, because it is frequently received but usually ineffective for BDD, which can lead to legal action and even violence toward treating clinicians. The book includes numerous clinical cases, which illustrate BDD's clinical features, its often-profound consequences, and recommended treatment approaches. This volume's contributors are the leading researchers and clinicians in this rapidly expanding field. Editor Katharine A. Phillips, head of the DSM-V committee on BDD, has done pioneering research on many aspects of this disorder, including its treatment. This book will be of interest to all clinicians who provide mental health treatment and to researchers in BDD, anxiety disorders, eating disorders, and other obsessive-compulsive and related disorders. It will be indispensable to surgeons, dermatologists, and other clinicians who provide cosmetic treatment. Students and trainees with an interest in psychology and mental health will also be interested in this book. This book fills a major gap in the literature by providing clinicians and researchers with cutting-edge, indispensable information on all aspects of BDD and its treatment.

JUnit Recipes Addison-Wesley Professional

Provides information on developing Rails 3 applications using RSpec and Cucumber.

Body Dysmorphic Disorder Packt Publishing Ltd

Your organization - business, church, or nonprofit - will experience unprecedented growth when you close the gap between these two game-changing questions: What are we known for? What do we want to be known for? In Know What You're FOR, entrepreneur and thought leader Jeff Henderson makes it clear that if we want to change the world with our products or our mission, then we must shift the focus of our messaging and marketing. Rather than self-promoting, we must transform our organizations to be people-centric. This sounds like a no-brainer, but looking closer shows just how little this is true and how impactful the change would be if it were. Whether you're a business leader, a change advocate, or a movement maker, Know What You're FOR will help you - and your organization - thrive. It's what happens when you create an organization focused on who it is FOR. This is the

future. Thriving organizations will be more concerned with becoming raving fans of their customers than they are trying to convince customers to become raving fans of the organization. This isn't theory. Jeff Henderson has experienced it. Working with companies like Chick-fil-A and the Atlanta Braves, then serving as a pastor for 15 years at one of the country's largest and most influential churches, North Point, Jeff knows what success looks like for healthy organizations and healthy lives. With fascinating stories from a host of entrepreneurs and Jeff's remarkable career, Know What You're FOR equips you with a simple strategy and the tools for extraordinary growth. You'll discover how to: Work FOR your current and future customers with a new, effective method Be FOR your team and help your people reach full potential Create a ripple impact by being FOR your community Live and work your best by caring FOR yourself In a hypercritical, cynical world, one that is often known for what it's against, let's be a group of people known for who and what we're FOR. It's a powerful strategy for business. But more importantly, it is a revolutionary way to live.

The RSpec Book Facets of Ruby

All software design is composition: the act of breaking complex problems down into smaller problems and composing those solutions. Most developers have a limited understanding of compositional techniques. It's time for that to change.In "Composing Software", Eric Elliott shares the fundamentals of composition, including both function composition and object composition, and explores them in the context of JavaScript. The book covers the foundations of both functional programming and object oriented programming to help the reader better understand how to build and structure complex applications using simple building blocks.You'll learn: Functional programmingObject compositionHow to work with composite data structuresClosuresHigher order functionsFunctors (e.g., array.map)Monads (e.g., promises)TransducersLensesAll of this in the context of JavaScript, the most used programming language in the world. But the learning doesn't stop at JavaScript. You'll be able to apply these lessons to any language. This book is about the timeless principles of software composition and its lessons will outlast the hot languages and frameworks of today. Unlike most programming books, this one may still be relevant 20 years from now.This book began life as a popular blog post series that attracted hundreds of thousands of readers and influenced the way software is built at many high growth tech startups and fortune 500 companies

The Art of Unit Testing Hogrefe Publishing GmbH

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

Event-Driven Architecture in Golang

Springer

Summary BDD in Action teaches you the Behavior-Driven Development model and shows you how to integrate it into your existing development process. First you'll learn how to apply BDD to requirements analysis to define features that focus your development efforts on underlying business goals. Then, you'll discover how to automate acceptance criteria and use tests to guide and report on the development process. Along the way, you'll apply BDD principles at the coding level to write more maintainable and better documented code. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology You can't write good software if you don't understand what it's supposed to do. Behavior-Driven Development (BDD) encourages teams to use conversation and concrete examples to build up a shared understanding of how an application should work and which features really matter. With an emerging body of best practices and sophisticated new tools that assist in requirement analysis and test automation, BDD has become a hot, mainstream practice. About the Book BDD in Action teaches you BDD principles and practices and shows you how to integrate them into your existing development process, no matter what language you use. First, you'll apply BDD to requirements analysis so you can focus your development efforts on underlying business goals. Then, you'll discover how to automate acceptance criteria and use tests to guide and report on the development process. Along the way, you'll apply BDD principles at the coding level to write more maintainable and better documented code. No prior experience with BDD is required. What's Inside BDD theory and practice How BDD will affect your team BDD for acceptance, integration, and unit testing Examples in Java, .NET, JavaScript, and more Reporting and living documentation About the Author John Ferguson Smart is a specialist in BDD, automated testing, and software lifecycle development optimization. Table of Contents PART 1: FIRST STEPS Building software that makes a difference BDD—the whirlwind tour PART 2: WHAT DO I WANT? DEFINING REQUIREMENTS USING BDD Understanding the business goals: Feature Injection and related techniques Defining and illustrating

features From examples to executable specifications Automating the scenarios PART 3: HOW DO I BUILD IT? CODING THE BDD WAY From executable specifications to rock-solid automated acceptance tests Automating acceptance criteria for the UI layer Automating acceptance criteria for non-UI requirements BDD and unit testing PART 4: TAKING BDD FURTHER Living Documentation: reporting and project management BDD in the build process

More Agile Testing Guilford Press

This is the fourth semi-fictional story from the Carnsa Development series. It focuses on Behaviour Driven Development, also known as BDD. It explores how BDD can be used to write effective acceptance criteria and support requirements. The story also includes a brief look at Test Driven Development (TDD) and Continuous Integration to support test automation. This book might be able to help you or your team with some of the following: Difficulty writing effective acceptance tests to support requirements? Not sure about the link between BDD and automated testing? Want understand the ideal mindset to use and prepare gherkins? Not sure how TDD and its relation to BDD? Not sure how continuous integration relates to testing? The story is set the life of the quirky Carnsa family, whose family projects are led by the mother and business analyst, Claudia. Granny has no problem voicing her opinion when she encounters some of the initial concepts that she does not agree with.Why not join the family in exploration of the subjects, with scenarios you can relate to and a quiz to test your knowledge?

Programming JavaScript Applications Addison-Wesley Professional

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).
JavaScript Testing with Jasmine Simon and Schuster
Without careful ongoing planning, the software development process can fall apart. Extreme Programming (XP) is a new programming discipline, or methodology, that is geared toward the way that the vast majority of software development projects are handled -- in small teams. In this new book, noted software engineers Kent Beck and Martin Fowler show the reader how to properly plan a software development project with XP in mind. The authors lay out a proven strategy that forces the reader to plan as their software project unfolds, and therefore avoid many of the nasty problems that can potentially spring up along the way.
Writing Effective Use Cases O'Reilly Media
Learn how to build dynamic web applications with Express, a key component of the Node/JavaScript development stack. In this hands-on guide, author Ethan Brown teaches you the fundamentals through the development of a fictional application that exposes a public website and a RESTful API. You'll also learn web architecture best practices to help you build single-page, multi-page, and hybrid web apps with Express. Express strikes a balance between a robust framework and no framework at all, allowing you a free hand in your architecture choices. With this book, frontend and backend engineers familiar with JavaScript will discover new ways of looking at web development. Create webpage templating system for rendering dynamic data Dive into request and response objects, middleware, and URL routing Simulate a production environment for testing and development Focus on persistence with document databases, particularly MongoDB Make your resources available to other programs with RESTful APIs Build secure apps with authentication, authorization, and HTTPS Integrate with social media, geolocation, and other third-party services Implement a plan for launching and maintaining your app Learn critical debugging skills This book covers Express 4.0.
Five Lines of Code Simon and Schuster
Deliver software that does what it's supposed to do! Behavior-Driven Development guides your software projects to success with collaboration, communication techniques, and concrete requirements you can turn into automated tests. In BDD in Action, Second Edition you'll learn how to: Implement and improve BDD practices Prioritize features from business goals Facilitate an example mapping session Write automated acceptance tests Scale up your automated acceptance tests Deliver accurate reporting and documentation Around half of all software projects fail to deliver on requirements. Behavior-Driven Development (BDD) helps make sure that yours isn't one of them. Behavior-Driven Development in Action, Second Edition teaches you how to ensure that everyone involved in a software project—from developers to non-technical stakeholders—are in agreement on goals and objectives. It lays out the communication skills, collaborative practices, and useful automation tools that will let you seamlessly succeed with BDD. Now in its second edition, this revised bestseller has been extensively updated with new techniques for incorporating BDD

into large-scale and enterprise development practices such as Agile and DevOps. Foreword by Daniel Terhorst-North. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Behavior-Driven Development is a collaborative software design technique that organizes examples of an application’s desired behavior into a concrete, testable specification. Because the BDD process gathers input from all areas of an organization, it maximizes the likelihood your software will satisfy both end users and business stakeholders. The established collaboration practices and automation strategies in this book will help you maximize the benefits of BDD for your dev team and your business clients. About the Book In BDD in Action, Second Edition, you’ll learn to seamlessly integrate BDD into your existing development process. This thoroughly revised new edition now shows how to integrate BDD with DevOps and large-scale Agile systems. Practical examples introduce cross-functional team communication skills, leading a successful requirements analysis, and how to set up automated acceptance criteria. What’s Inside How BDD positively affects teamwork, dynamics, and collaboration with stakeholders Help teams discover and analyze requirements, uncover assumptions, and reduce risks Make acceptance, integration, and unit testing more effective Automate reporting and living documentation to improve transparency About the Reader For all development teams. No experience with BDD required. Examples in Java, JavaScript, and TypeScript can be easily expressed in your chosen language. About the Author John Ferguson Smart is the creator of the Serenity BDD framework and founder of the Serenity Dojo training school. Jan Molak is the author of the Serenity/JS testing framework, Jenkins Build Monitor, and other CD and testing tools. Table of Contents PART 1 - FIRST STEPS 1 Building software that makes a difference 2 Introducing Behavior-Driven Development 3 BDD: The whirlwind tour PART 2 - WHAT DO I WANT? DEFINING REQUIREMENTS USING BDD 4 Speculate: From business goals to prioritized features 5 Describing and prioritizing features 6 Illustrating features with examples 7 From examples to executable specifications PART 3 - HOW DO I BUILD IT? CODING THE BDD WAY 8 From executable specifications to automated acceptance tests 9 Writing solid automated acceptance tests 10 Automating acceptance criteria for the UI layer 11 Test automation design patterns for the UI layer 12 Scalable test automation with the Screenplay Pattern 13 BDD and executable specifications for microservices and APIs 14 Executable specifications for existing systems with Serenity/JS 15 Portable test automation with Serenity/JS 16 Living documentation and release evidence

Know What You're FOR Addison-Wesley Professional

Your customers want rock-solid, bug-free software that does exactly what they expect it to do. Yet they can't always articulate their ideas clearly enough for you to turn them into code. You need Cucumber: a testing, communication, and requirements tool-all rolled into one. All the code in this book is updated for Cucumber 2.4, Rails 5, and RSpec 3.5. Express your customers' wild ideas as a set of clear, executable specifications that everyone on the team can read. Feed

those examples into Cucumber and let it guide your development. Build just the right code to keep your customers happy. You can use Cucumber to test almost any system or any platform. Get started by using the core features of Cucumber and working with Cucumber's Gherkin DSL to describe-in plain language-the behavior your customers want from the system. Then write Ruby code that interprets those plain-language specifications and checks them against your application. Next, consolidate the knowledge you've gained with a worked example, where you'll learn more advanced Cucumber techniques, test asynchronous systems, and test systems that use a database. Recipes highlight some of the most difficult and commonly seen situations the authors have helped teams solve. With these patterns and techniques, test Ajax-heavy web applications with Capybara and Selenium, REST web services, Ruby on Rails applications, command-line applications, legacy applications, and more. Written by the creator of Cucumber and the co-founders of Cucumber Ltd., this authoritative guide will give you and your team all the knowledge you need to start using Cucumber with confidence. What You Need: Windows, Mac OS X (with XCode) or Linux, Ruby 1.9.2 and upwards, Cucumber 2.4, Rails 5, and RSpec 3.5

Specification by Example Oxford University Press

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

The Professional Scrum Master Guide

Addison-Wesley Professional

Developers looking to keep their JavaScript code bug-free will want to unit test using Jasmine, one of the most popular unit testing frameworks around. Any project of meaningful size should be automatically tested to help catch bugs as early as possible. Jasmine, a testing framework for JavaScript, makes it easy to test JavaScript projects, from browser-based applications to Node.js. While a quick understanding of Jasmine can be gleaned from the project's homepage, the framework has a lot of details and exciting plugins. This book explores Jasmine in a depth that can't be found elsewhere. This book provides: Exposure to some Jasmine plugins, to extend Jasmine and allow for more functionality and more thorough testing An Understanding of Jasmine's main features, to allow code to be automatically tested and reduce bugs An Explanation of how to get Jasmine working in different environments (in the browser, in Node.js, through Rails, et cetera), to make Jasmine easier to work with

Discovery Oxford University Press, USA

Begin building event-driven microservices, including patterns to handle data consistency and resiliency Key Features Explore the benefits and tradeoffs of event-driven architectures with practical examples and use cases Understand synergy with event sourcing, CQRS, and domain-driven development in software architecture Build an end-to-end robust application architecture by the end of the book Book DescriptionEvent-driven architecture in Golang is an approach used to develop applications that shares state changes asynchronously, internally, and externally using messages. EDA applications are better suited at handling situations that need to scale up quickly and the chances of individual component failures are less likely to bring your system crashing down. This is why EDA is a great thing to learn and this book is designed to get you started with the help of step-by-step explanations of essential concepts, practical examples, and more. You'll begin building event-driven microservices, including patterns to handle data consistency and resiliency. Not only will you learn the patterns behind event-driven microservices but also how to communicate using asynchronous messaging with event streams. You'll then build an application made of several microservices that communicates using both choreographed and orchestrated messaging. By the end of this book, you'll be able to build and deploy your own event-driven microservices using asynchronous communication.What you will learn Understand different event-driven patterns and best practices Plan and design your software architecture with ease Track changes and updates effectively using event sourcing Test and deploy your sample software application with ease Monitor and improve the performance of your software architecture

Who this book is for This hands-on book is for intermediate-level software architects, or senior software engineers working with Golang and interested in building asynchronous microservices using event sourcing, CQRS, and DDD. Intermediate-level knowledge of the Go syntax and concurrency features is necessary.
Practical Object-oriented Design in Ruby Packt Publishing Ltd
When testing becomes a developer's habit good things tend to happen--good productivity, good code, and good job satisfaction. If you want some of that, there's no better way to start your testing habit, nor to continue feeding it, than with"" JUnit Recipes,"" In this book you will find one hundred and thirty-seven solutions to a range of problems, from simple to complex, selected for you by an experienced developer and master tester. Each recipe follows the same organization giving you the problem and its background before discussing your options in solving it. JUnit - the unit testing framework for Java - is simple to use, but some code can be tricky to test. When you're facing such code you will be glad to have this book. It is a how-to reference full of practical advice on all issues of testing, from how to name your test case classes to how to test complicated J2EE applications. Its valuable advice includes side matters that can have a big payoff, like how to organize your test data or how to manage expensive test resources. What's Inside: - Getting started with JUnit - Recipes for: servlets JSPs EJBs Database code much more - Difficult-to-test designs, and how to fix them - How testing saves time - Choose a JUnit extension: HTMLUnit XMLUnit ServletUnit EasyMock and more!
BDD in Action, Second Edition Apress
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