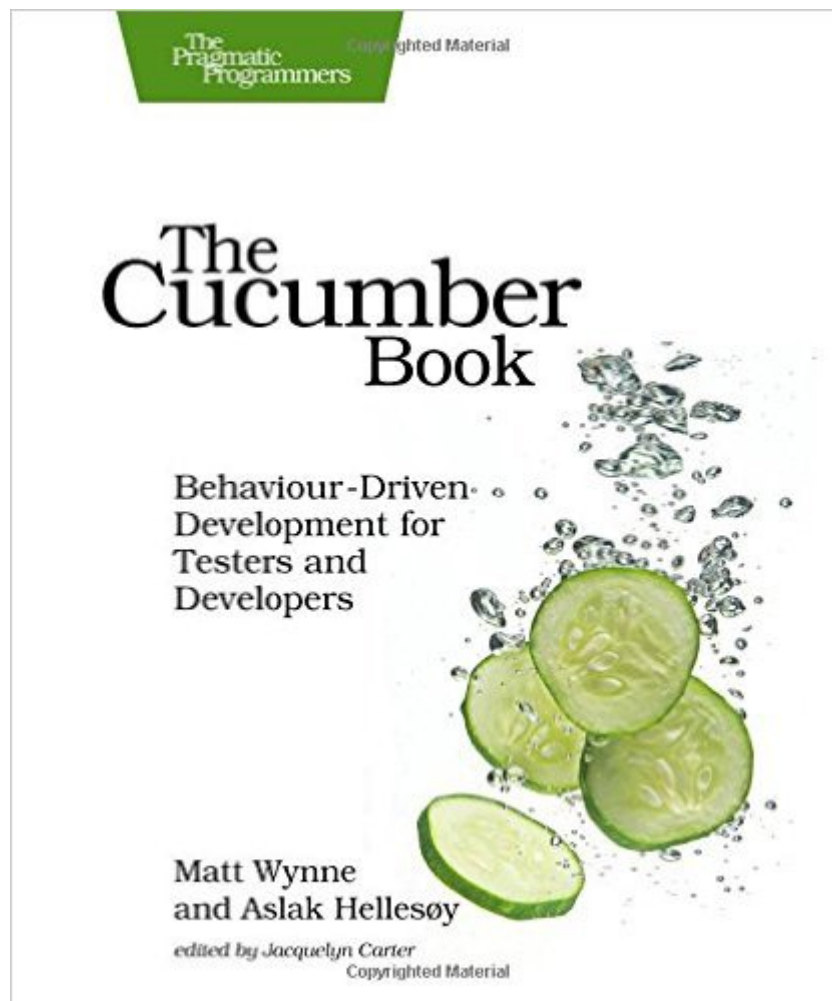# The Cucumber Book: Behaviour-Driven Development For Testers And Developers (Pragmatic Programmers)

Synopsis

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. The Cucumber Book dives straight into the core of the problem: communication between people. Cucumber saves the day; it's a testing, communication, and requirements tool - all rolled into one.We'll show you how to express your customers' wild ideas as a set of clear, executable specifications that everyone on the team can read. You'll learn how to feed those examples into Cucumber and let it guide your development. You'll build just the right code to keep your customers happy, and not a line more.The first part of the book teaches you how to use the core features of Cucumber. You'll learn how to use Cucumber's Gherkin DSL to describe-- in plain language - the behavior your customers want from the system. You'll learn how to write Ruby code that interprets those plain language specifications and checks them against your application. In Part 2, you'll consolidate the knowledge you just gained with a worked example.Although it was born in the Ruby community, you can use Cucumber to test almost any system, from a simple shell script or a Perl script, to enterprise PHP or a Java web application. In Part 3, you'll find a selection of recipes for some of the most common situations you'll encounter using Cucumber in the wild. You'll learn how to test Ajax-heavy web applications with Capybara and Selenium, REST web services, Ruby on Rails applications, command-line applications, legacy applications and lots more!Written by the creator of Cucumber and one of its most experienced users and contributors, The Cucumber Book is an authoritative guide that 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 LinuxRuby 1.9.2 and upwards

Book Information

Series: Pragmatic Programmers

Paperback: 336 pages

Publisher: Pragmatic Bookshelf; 1 edition (February 10, 2012)

Language: English

ISBN-10: 1934356808

ISBN-13: 978-1934356807

Product Dimensions: 7.5 x 0.6 x 9.2 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 stars See all reviews (24 customer reviews)

Best Sellers Rank: #74,068 in Books (See Top 100 in Books)   #13 in Books > Computers &

Technology > Programming > Languages & Tools > Ruby   #39 inÂ Books > Computers & Technology > Computer Science > Systems Analysis & Design   #190 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

I was looking forward to the Cucumber book and excepted it to be great yet not excellent. Especially as there is already an excellent introduction to Cucumber called "The Secret Cucumber Ninja Scrolls." However, I was surprised by the book, it was better and broader than I expected. I learned a lot from it and not just about Cucumber but also about the latest on ruby test automation techniques and tools.The Cucumber book consists of 3 different parts. The first part is an basic introduction to Cucumber, the second part provides a 3-chapter long example and the last part shows how to test different type of application.Part one starts by introducing the concepts of BDD (or A-TDD or "Specification by Example" which are all very similar) and explains that cucumber, in the end, is a collaboration tool where the developers, testers and users learn to speak the same language and that way improve development of software. Cucumber provides a way of expressing and automating that shared language. The next couple of chapters introduce the basic features of cucumber one at the time. The last chapter of the first part talks about common test automation problems and that their causes are and what you can do about this. Most of this chapter and the concepts expressed in the book are valid for any of the BDD/A-TDD frameworks such as Fitnesse or RobotFramework.Part two started out disappointing to me. The authors decided to use an ATM as example of their test. ATMs has frequently been used in software development books (like calculators, which unfortunately it also used) and I had hoped for a different, less stereotypical, domain. Yet, as part two progressed, I started liking the example more and eventually part two became my favorite part of the book. Its style reminded me of the excellentÂ Growing Object-Oriented Software, Guided by Tests. The example starts by writing tests that directly talk to the application domain logic. After getting the logic to work, they switch the interface to a web interface without making a change to the written specifications (excellent!) and clearly show that the specification and application logic are and should be separated. After this they explain how to deal with asynchronous events and testing with a database.Part three is sort-of the left-over chapter of what the authors still wanted to say about cucumber that didn't fit in the other chapters. It starts with exploring the command line options to cucumber and then dives into testing different type of applications, namely: REST, legacy, rails, ajax and CLI. Each of these chapters (except legacy code) provides excellent examples of working outside-in with cucumber specifications and each

chapter introduces additional useful ruby (testing) tools.So, in the end, I was positively surprised about the book and enjoyed every page. What drawbacks does the book have? The collaborative part of BDD was only stressed in the first couple of chapters, which was too bad. As mentioned, I found the examples a bit too stereotypical. I guess the book wouldn't be easy to read for non-ruby developers as there is a fair amount of ruby code in it and the alternative language cucumber clones aren't covered. Yet, I found these drawbacks minimal and would still rate it 5 stars. For explaining BDD (A-TDD/Specification by Example), I'd still recommend Gojko Adzic'sÂ Specification by Example: How Successful Teams Deliver the Right SoftwareÂ over the cucumber book. ButÂ Specification by Example: How Successful Teams Deliver the Right SoftwareÂ doesn't cover the concrete implementation in a tool, for that, I'd definitively recommend the cucumber book. They compliment each other nicely.

After some experience in software testing I decided to try Cucumber. With no prior Cucumber or Ruby experience I bought this book and read it (and completed most of the examples) on a rainy weekend. Not only did this book teach me cucumber but it also gave me some cool Behavior Driven Development (BDD) ideas and concepts and even gave some ideas on how to program with Ruby. At my current place of employment, my colleague also bought the book and after reading it we confidently implemented a solid BDD test framework using Cucumber in a couple of days. I highly recommend this book for anyone who wants to learn Cucumber and also for anyone who wants to understand how BDD frameworks (like Cucumber) can be valuable on a software project.

One of the cool things about Pragmatic Publishing is the fact that they make it possible to get your hands on Beta books, meaning you get the chance to see a book as its actively being developed. The Cucumber Book was one of those books, and as such, I've had the benefit of looking at and reviewing this book for the past several months, and have watched it grow into the book that is today (and now available in print form).Most people who have a passing understanding of Test Driven Development or Behavior Driven Development have likely heard of Cucumber. It's a language that allows anyone who wants to define tests and requirements for applications the ability to do so in plain English (or fill in the blank language if supported). In truth, Cucumber isn't really a programming language at all, but a symbolic phrase library that matches to various underlying commands and blocks of code (represented in Ruby in this book and referencing a variety of tools including Capybara, Rspec and others).Matt Wynne and Aslak HellesÃƒÂ¸y have put together a very readable and focused text that help the user get familiar with the basics of the language. The

book also focuses the reader on understanding the underpinnings needed to create expressions that work with their respective technologies. Granted, if you are a tester and you want to take advantage of this framework, there is plenty in here to keep you busy. The Cucumber Book starts out by explaining what Cucumber is and the niche it is meant to fill (specifications based tests and requirements). If you are a developer, there is likewise plenty in here to keep you interested, too.The process in the Cucumber book is heavy on examples and showing how the examples work. Yes, for those who want to know how to use the syntax and language specific details of Cucumber, that stuff is covered. What is also covered, and covered well, is the Behavioral Driven Development approach needed to effectively create tests and have them work effectively. Along with creating feature files and steps for those feature files, the underlying step definitions also have to be coded. Not only do they have to be coded, but they have to have assertions written that will effectively confirm if the step has passed, or if it fails, and why.Since the book is primarily based on Cucumber, there is a large section that covers Cucumber fundamentals, including basic Gherkin (the underlying syntax that Cucumber uses), and the ability of using expressive options such as Scenario Outlines, Data tables, Doc Strings, tags, and dealing with some of the pain points seen in your tests (such as "flickering scenarios", where the tests pass some of the time but fail some times, too). More than just using Cucumber to define steps and have step definitions defined, the third part of the book deals with applying Cucumber to a number of different technologies; working with various databases, testing with RESTful web services, working with Rails, running tests and using capybara to simulate common browser actions and many other options that may come to play in your everyday testing life.Bottom Line:If you have ever been interested in looking at Cucumber and your testing environment is built around Ruby, then this will be an ideal book to use. If you are interested in deploying Cucumber in another type of environment, such as testing with Java or .NET, many of the ideas in this book will also carry over, but have a look at "The Secret Ninja Cucumber Scrolls" by David de Florinier and Gojko Adzic. It provides information about how to apply Cucumber to those environments. Regardless of your particular focus and environment needs, for a practical and effective book for learning and using Cucumber in a meaningful way, The Cucumber Book is an excellent addition to any tester or developer's library.

The book's title might lead one to believe that its contents are only for people that use the Cucumber tool for BDD with Ruby, when in fact the authors cover most of their topics in such a way that most of the principles can be applied to whichever BDD tool and programming language one is using.Although the book is geared towards a more technical reader, the authors offer what I think

are some of the best general overviews of BDD available.The book is sliced into three main parts:Part 1 - Cucumber FundamentalsPart 2 - Working ExamplePart 3 - Cucumber AppliedI would recommend this book to anyone who is new to BDD, as well as enginers who are familiar with Cucumber or BDD but are looking to expand their skills. The biggest benefit for me was that after reading it I feel I have the information I need to successfully create killer automation for my next BDD project.Joe Colantonio@JoeColantonio.com

The Cucumber Book: Behaviour-Driven Development for Testers and Developers (Pragmatic Programmers) The Cucumber for Java Book: Behaviour-Driven Development for Testers and Developers The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends (Facets of Ruby) Web Design for Developers: A Programmer's Guide to Design Tools and Techniques (Pragmatic Programmers) Pragmatic Guide to Git (Pragmatic Guides) 3D Game Programming for Kids: Create Interactive Worlds with JavaScript (Pragmatic Programmers) Programming Clojure (Pragmatic Programmers) Programming Ruby 1.9 & 2.0: The Pragmatic Programmers' Guide (The Facets of Ruby) Seven Concurrency Models in Seven Weeks: When Threads Unravel (The Pragmatic Programmers) The Agile Samurai: How Agile Masters Deliver Great Software (Pragmatic Programmers) Programming Groovy: Dynamic Productivity for the Java Developer (Pragmatic Programmers) Programming Groovy 2: Dynamic Productivity for the Java Developer (Pragmatic Programmers) Practical Programming: An Introduction to Computer Science Using Python 3 (Pragmatic Programmers) Object-Oriented Frameworks Using C++ and CORBA Gold Book: The Must-have Guide to CORBA for Developers and Programmers An Illustrated Guide for z/Architecture Assembler Programmers: A compact reference for application programmers Delphi Nuts & Bolts for Experienced Programmers: For Experienced Programmers Android 6 for Programmers: An App-Driven Approach (3rd Edition) (Deitel Developer Series) What Customers Want: Using Outcome-Driven Innovation to Create Breakthrough Products and Services: Using Outcome-Driven Innovation to Create Breakthrough Products and Services Everyday Scripting with Ruby: For Teams, Testers, and You Agile Testing: A Practical Guide for Testers and Agile Teams