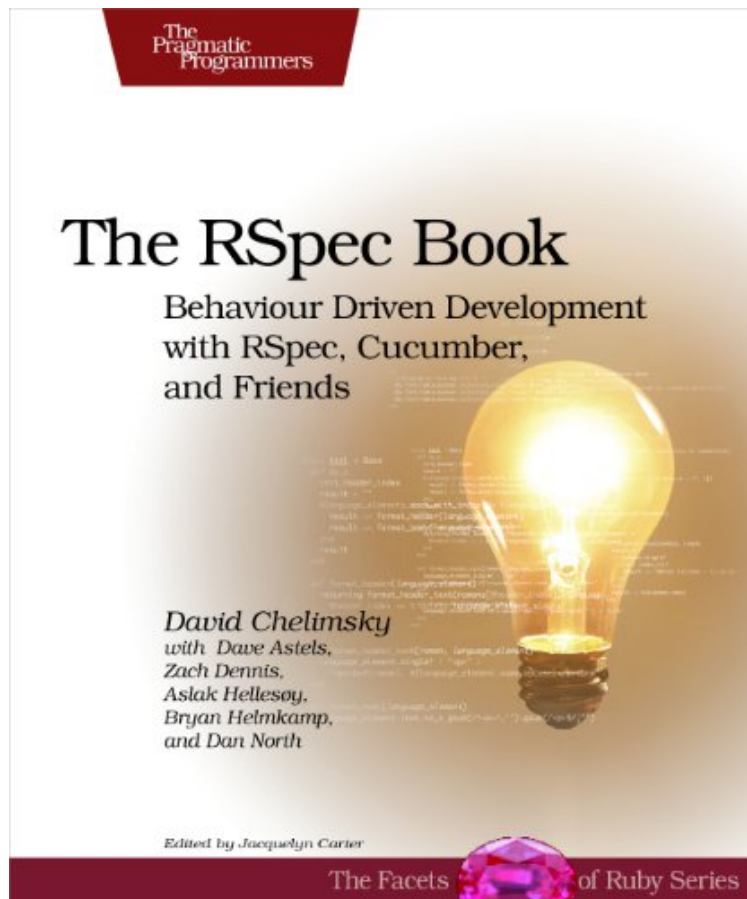


[Mobile library] The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends (Facets of Ruby)

The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends (Facets of Ruby)

David Chelimsky, Dave Astels, Bryan Helmkamp, Dan North, Zach Dennis, Aslak Hellesoy
*ebooks | Download PDF | *ePub | DOC | audiobook*



#780648 in eBooks 2010-12-15 2012-11-06File Name: B00A32NY44 | File size: 16.Mb

David Chelimsky, Dave Astels, Bryan Helmkamp, Dan North, Zach Dennis, Aslak Hellesoy : The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends (Facets of Ruby) before purchasing it in order to gage whether or not it would be worth my time, and all praised The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends (Facets of Ruby):

2 of 2 people found the following review helpful. Good but outdatedBy SiddhardhaI bought this book to obtain a more in-depth treatment of rspec that some of the other books I have read didn't have. Rspec is the framework used for testing where I work and I needed to get up to speed on it. Part I of the book starts with an introduction of rspec and cucumber and then proceeds to illustrate BDD with these two tools for a simple game known as a code breaker. I thought this part presented very well. Real world applications tend to be much more complex than the simple one presented here, but hopefully the principles will apply (with some modifications as appropriate). Part 2 describes BDD in a little more detail while Part III and Part IV cover rspec and cucumber respectively. The part about Rspec is quite

good - there is a fair amount of detail with ideas on when to use what along with examples. Part IV is also okay but it seemed to not have enough depth - this isn't necessarily an issue as there is another book on Cucumber I intend to read soon. Part 5 is about applying BDD to Rails. I usually test out the code samples in every book I buy by typing them out from the book and not downloading the source (unless absolutely required). Once concern I had when I purchased this book is that the versions of the gems used in the book are outdated and therefore the code may not work on newer version of gems. As a matter of fact, the book recommends using the versions of the gems that the original was source was tested with. I had instead used the latest version of all gems. In Parts I through IV, not many changes are required to get the source to work properly with the newer versions of the gems. Occasionally there may be warning (for instance, in RSpec, Stub is deprecated, use double instead) and this was easy to fix and once fixed, the warning went away. The only part that I didn't try the samples was Part V - BDD with Rails - the reason being the book uses Webrat but Capybara seems to be the popular choice now in the community. The book does cover Selenium and includes ideas for testing different components of Rails, so the ideas are usable, but the source code isn't out of the box. In other words, this section of the book is outdated. Notwithstanding the caveats, this book is still worth a read. 2 of 2 people found the following review helpful. Good intro to RSpec... and more! By Bas Vodde I find the RSpec book hard to review as on one hand, it's excellent, but on the other hand, I felt it could be still much better. Also, with the introduction of the more recent Cucumber book, makes it hard to appreciate the extra stuff in this book as it is quite a bit overlapping with that of the Cucumber book. I still enjoyed the book. The book is divided in five parts. It feels a bit like the parts are written by different authors independently, which isn't unlikely considering the amount of authors. That's too bad as some cross-referring would have made the book better. The first part is tutorial style where it simulates implementing a project Behavior-Driven-Development-style (BDD). The project is the traditional problem of code-breaker where you need to guess a code and get some hints on how well you did. It drives the project by first writing cucumber specs, then test-driving the implementation using RSpec. The tutorial is simple (perhaps even simplistic) and goes on for about 100 pages. The second part of the book is the philosophy behind BDD. It is short, consisting of only 2 chapters. The first describes how traditional projects work... or actually how they do not work. Then it quickly runs over some ideas behind agile development and how that is different. The next chapter makes a case for by showing how BDD actually focuses on the communication between developers and customers. The third part is the RSpec part of the book (where it got its name :P). It's about a 100 pages and does a pretty good job in describing RSpec. The only thing that I was missing was that it could explain a bit more about the RSpec internals... at least I would have found that interesting. The fourth chapter is the "and more!" part of the book where it introduces cucumber (in the RSpec book :P). The Cucumber introduction isn't fantastic and is pretty short (and has quite some overlap with the tutorial). I wouldn't recommend this book for the cucumber introduction anyways but instead it would probably be better to read... the cucumber book (published about 2 years later). The last chapter is the rails chapter where it takes rspec and cucumber and explain how to use the extensions for writing features and specs for rails apps. Here, some of the text is already a bit obsolete as the technology moves fast. Still, the concepts are about the same. It contains 3 chapters on cucumber and 3 on rspec. All in all, I found it a pretty good read and enjoyed most of it. It could be a bit thinner and faster for me as it felt a bit repetitive at times. Partly this was because because some chapters could be integrated better together. For the RSpec part, I'd rate the book 3 stars as it does a good job in describing rspec. But the book is much more than The RSpec Book... therefore I'll give it 4 stars (yet, for Cucumber, it is better to pick up the Cucumber book) 16 of 17 people found the following review helpful. If you are a hard core developer By David A. Taylor I am a long time Web Developer, ramping back up on RoR. I was looking for a book that would help me develop a good solid testing environment on this new Rails 3 project that I am creating. As I read and tried many of the little tiny examples in the book, and eventually decided that I do not want to do Cucumber (I do not need to spend the extra time to generate code to translate requirements from English, RSpec is clear enough for me). Unfortunately (from my perspective), much of the book rambles on about Cucumber and integrating it with RSpec. As I went through the book and I found a section of code that interested me, it too frequently told me that I would hear more details later on, which I found quite frustrating. I was ready for the down-low, and never seemed to find it, until I eventually jumped to Chapters 23, 24 and 25. Chapters 23, 24 and 25 are the chapters that walk you through the process of developing Test/Behavior driven View, Controllers and Models. This is what I needed to get my project going. This book is worth it, even if you only look at the RSpec chapters. Oh, by the way, when you are looking into the tools you want to use for integration testing, I recommend looking into Capybara, which is not talked about in the book. I hope this helps. Dave Taylor (tayloredwebsites.com)

Behaviour-Driven Development (BDD) gives you the best of Test Driven Development, Domain Driven Design, and Acceptance Test Driven Planning techniques, so you can create better software with self-documenting, executable tests that bring users and developers together with a common language. Get the most out of BDD in Ruby with The RSpec Book, written by the lead developer of RSpec, David Chelimsky. You'll get started right away with RSpec 2 and Cucumber by developing a simple game, using Cucumber to express high-level requirements in language your customer understands, and RSpec to express more granular requirements that focus on the behavior of individual

objects in the system. You'll learn how to use test doubles (mocks and stubs) to control the environment and focus the RSpec examples on one object at a time, and how to customize RSpec to "speak" in the language of your domain. You'll develop Rails 3 applications and use companion tools such as Webrat and Selenium to express requirements for web applications both in memory and in the browser. And you'll learn to specify Rails views, controllers, and models, each in complete isolation from the other. Whether you're developing applications, frameworks, or the libraries that power them, The RSpec Book will help you write better code, better tests, and deliver better software to happier users.

"Some authors would be satisfied with just writing the definitive guide for a technology. These folks go a step further, and show you insider tips that will keep your tests clean and maintainable."