Extracted from:

The RSpec Book

Behaviour-Driven Development with RSpec, Cucumber, and Friends

This PDF file contains pages extracted from The RSpec Book, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Programmers

The RSpec Book

Behaviour-Driven Development with RSpec, Cucumber, and Friends

David Chelimsky
with Dave Astels,
Zach Dennis,
Aslak Hellesøy,
Bryan Helmkamp,
and Dan North

Foreword by Robert C. Martin (Uncle Bob)

Edited by Jacquelyn Carter

The Facets of Ruby Series

Contents

Foreword **13**

About the Authors **16**

Acknowledgments **18**

Preface **20**
- Ruby and Gem Versions . . . 20
- Downloading the Code Examples . . . 21
- What's in This Book . . . 21

I Getting Started with RSpec and Cucumber **23**

1 Introduction **24**
- 1.1 Test-Driven Development: Where It All Started . . . 24
- 1.2 Behaviour-Driven Development: The Next Step . . . 26
- 1.3 RSpec . . . 27
- 1.4 Cucumber . . . 28
- 1.5 The BDD Cycle . . . 30

2 Hello **32**
- 2.1 Installation . . . 32
- 2.2 Hello RSpec . . . 33
- 2.3 Hello Cucumber . . . 35

3 Describing Features **39**
- 3.1 Introducing Codebreaker . . . 40
- 3.2 Planning the First Release . . . 40
- 3.3 Planning the First Iteration . . . 46
- 3.4 What We've Learned . . . 54

4 Automating Features with Cucumber **55**
4.1 Steps and Step Definitions 57
4.2 Step Definition Methods 58
4.3 Test Double 62
4.4 What We've Learned 64

5 Describing Code with RSpec **65**
5.1 Getting Started with RSpec 65
5.2 Red: Start with a Failing Code Example 68
5.3 Green: Get the Example to Pass 70
5.4 Refactor 74
5.5 What We've Learned 78

6 Adding New Features **79**
6.1 Scenario Outlines in Cucumber 79
6.2 Responding to Change 82
6.3 What We've Learned 85

7 Specifying an Algorithm **86**
7.1 Begin with the Simplest Example 86
7.2 Refactor to Remove Duplication 90
7.3 Refactor to Express Intent 91
7.4 What We've Learned 96

8 Refactoring with Confidence **98**
8.1 Sniffing Out Code Smells 98
8.2 One Step at a Time 100
8.3 Updating Specs After Refactoring 109
8.4 Exploratory Testing 113
8.5 What We've Learned 114

9 Feeding Back What We've Learned **116**
9.1 Use Cucumber for Collaboration 116
9.2 Experimenting with a New Implementation 120
9.3 What We've Learned 124

Click Here to purchase this book now.

II Behaviour-Driven Development 126

10 The Case for BDD 127
10.1 How Traditional Projects Fail 127
10.2 Why Traditional Projects Fail 128
10.3 Redefining the Problem 132
10.4 The Cost of Going Agile 135
10.5 What We've Learned 138

11 Writing Software That Matters 140
11.1 A Description of BDD 140
11.2 The Principles of BDD 140
11.3 The Project Inception 141
11.4 The Cycle of Delivery 143
11.5 What's in a Story? 147
11.6 What We've Learned 150

III RSpec 151

12 Code Examples 152
12.1 Describe It! 153
12.2 Pending Examples 157
12.3 Hooks: Before, After, and Around 160
12.4 Helper Methods 164
12.5 Shared Examples 166
12.6 Nested Example Groups 168
12.7 What We've Learned 171

13 RSpec::Expectations 172
13.1 should, should_not, and matchers 173
13.2 Built-in Matchers 174
13.3 Predicate Matchers 182
13.4 Be True in the Eyes of Ruby 183
13.5 Have Whatever You Like 184
13.6 Operator Expressions 188
13.7 Generated Descriptions 189
13.8 Subjectivity 190
13.9 What We've Learned 192

CLICK HERE to purchase this book now.

14 RSpec::Mocks **193**
14.1 Test Doubles 194
14.2 Method Stubs 195
14.3 Message Expectations 197
14.4 Test-Specific Extensions 199
14.5 More on Method Stubs 203
14.6 More on Message Expectations 205
14.7 When to Use Test Doubles and Test-Specific Extensions 215
14.8 Risks and Trade-Offs 220
14.9 Choosing Other Test Double Frameworks 222
14.10 What We've Learned 224

15 Tools and Integration **226**
15.1 The rspec Command 226
15.2 TextMate 231
15.3 Autotest 232
15.4 Rake 232
15.5 RCov 234
15.6 What We've Learned 235

16 Extending RSpec **236**
16.1 Metadata 236
16.2 Configuration 237
16.3 Filtering 237
16.4 Extension Modules 240
16.5 Global Hooks 241
16.6 Mock Framework 241
16.7 Custom Matchers 242
16.8 Macros 246
16.9 Custom Formatters 249
16.10 What We've Learned 252

IV Cucumber **253**

17 Intro to Cucumber **254**
17.1 From 20,000 Feet 255
17.2 Features 255
17.3 Customer Acceptance Tests 259
17.4 Gherkin 259
17.5 Scenarios 261

17.6 Steps 262
17.7 The cucumber Command 263
17.8 Given/When/Then 264
17.9 Declarative and Imperative Scenario Styles 265
17.10 Organizing Features 266
17.11 Tags 267
17.12 What We've Learned 269

18 Cucumber Detail 271
18.1 Step Definitions 271
18.2 World 273
18.3 Calling Steps Within Step Definitions 274
18.4 Hooks 275
18.5 Background 277
18.6 Multiline Text 278
18.7 Tables in Steps 279
18.8 Scenario Outlines 280
18.9 Configuration 282
18.10 What We've Learned 282

V Behaviour-Driven Rails 284

19 BDD in Rails 285
19.1 Outside-In Rails Development 286
19.2 Setting Up a Rails 3 Project 288
19.3 Setting Up a Rails 2 Project 291
19.4 What We've Learned 291

20 Cucumber with Rails 292
20.1 Step Definition Styles 292
20.2 Direct Model Access 294
20.3 What We've Learned 299

21 Simulating the Browser with Webrat 300
21.1 Writing Simulated Browser Step Definitions 301
21.2 Navigating to Pages 306
21.3 Manipulating Forms 310
21.4 Specifying Outcomes with View Matchers 315
21.5 Building on the Basics 318
21.6 What We've Learned 321

CLICK HERE to purchase this book now.

22 Automating the Browser with Webrat and Selenium **322**
22.1 Getting Started 323
22.2 Writing Step Definitions for Selenium 326
22.3 Debugging Selenium Issues 330
22.4 What We've Learned 332

23 Rails Views **333**
23.1 Writing View Specs 333
23.2 Mocking Models 336
23.3 Specifying Helpers 343
23.4 When Should I Write View Specs? 345
23.5 What We've Learned 345

24 Rails Controllers **347**
24.1 Controller Specs 347
24.2 Context-Specific Examples 354
24.3 Specifying ApplicationController 363
24.4 What We've Learned 365

25 Rails Models **367**
25.1 Writing Model Specs 367
25.2 Specifying Business Rules 375
25.3 Exercise 384
25.4 Useful Tidbits 385
25.5 What We've Learned 388

A RubySpec **390**
A.1 The Project 391
A.2 Syntax 391
A.3 Guards 393
A.4 Extensibility 394
A.5 MSpec 396
A.6 Tags 396
A.7 Community 397

B RSpec's Built-in Expectations **399**

C Bibliography **403**

Index **405**

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

The RSpec Book's Home Page
http://pragprog.com/titles/achbd
Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates
http://pragprog.com/updates
Be notified when updates and new books become available.

Join the Community
http://pragprog.com/community
Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy
http://pragprog.com/news
Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/achbd.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)