

The RSpec Book

Behaviour-Driven Development
with RSpec, Cucumber,
and Friends

David Chelmsky

with Dave Astels,

Zach Dennis,

Aslak Hellesøy,

Bryan Helmkamp,

and Dan North

*Foreword by Robert C. Martin
(Uncle Bob)*

Edited by Jacquelyn Carter



Contents

Foreword	xiii
About the Authors	xvii
Acknowledgments	xix
Preface	xxi
Ruby and Gem Versions	xxi
Downloading the Code Examples	xxii
What's in This Book	xxii
I Getting Started with RSpec and Cucumber	1
1 Introduction	3
1.1 Test-Driven Development: Where It All Started . . .	3
1.2 Behaviour-Driven Development: The Next Step . . .	5
1.3 RSpec	6
1.4 Cucumber	7
1.5 The BDD Cycle	9
2 Hello	11
2.1 Installation	11
2.2 Hello RSpec	12
2.3 Hello Cucumber	14
3 Describing Features	19
3.1 Introducing Codebreaker	20
3.2 Planning the First Release	20
3.3 Planning the First Iteration	26
3.4 What We've Learned	34

4	Automating Features with Cucumber	35
4.1	Steps and Step Definitions	37
4.2	Step Definition Methods	38
4.3	Test Double	42
4.4	What We've Learned	44
5	Describing Code with RSpec	45
5.1	Getting Started with RSpec	45
5.2	Red: Start with a Failing Code Example	48
5.3	Green: Get the Example to Pass	50
5.4	Refactor	54
5.5	What We've Learned	58
6	Adding New Features	59
6.1	Scenario Outlines in Cucumber	59
6.2	Responding to Change	62
6.3	What We've Learned	65
7	Specifying an Algorithm	67
7.1	Begin with the Simplest Example	67
7.2	Refactor to Remove Duplication	71
7.3	Refactor to Express Intent	72
7.4	What We've Learned	77
8	Refactoring with Confidence	79
8.1	Sniffing Out Code Smells	79
8.2	One Step at a Time	81
8.3	Updating Specs After Refactoring	90
8.4	Exploratory Testing	94
8.5	What We've Learned	95
9	Feeding Back What We've Learned	97
9.1	Use Cucumber for Collaboration	97
9.2	Experimenting with a New Implementation	101
9.3	What We've Learned	105

II	Behaviour-Driven Development	107
10	The Case for BDD	109
10.1	How Traditional Projects Fail	109
10.2	Why Traditional Projects Fail	110
10.3	Redefining the Problem	114
10.4	The Cost of Going Agile	117
10.5	What We've Learned	120
11	Writing Software That Matters	123
11.1	A Description of BDD	123
11.2	The Principles of BDD	123
11.3	The Project Inception	124
11.4	The Cycle of Delivery	126
11.5	What's in a Story?	130
11.6	What We've Learned	133
III	RSpec	135
12	Code Examples	137
12.1	Describe It!	138
12.2	Pending Examples	142
12.3	Hooks: Before, After, and Around	145
12.4	Helper Methods	149
12.5	Shared Examples	151
12.6	Nested Example Groups	153
12.7	What We've Learned	156
13	RSpec::Expectations	157
13.1	should, should_not, and matchers	158
13.2	Built-in Matchers	159
13.3	Predicate Matchers	167
13.4	Be True in the Eyes of Ruby	168
13.5	Have Whatever You Like	169
13.6	Operator Expressions	173
13.7	Generated Descriptions	174
13.8	Subjectivity	175
13.9	What We've Learned	177

14	RSpec::Mocks	179
14.1	Test Doubles	180
14.2	Method Stubs	181
14.3	Message Expectations	183
14.4	Test-Specific Extensions	185
14.5	More on Method Stubs	189
14.6	More on Message Expectations	191
14.7	When to Use Test Doubles and Test-Specific Extensions	201
14.8	Risks and Trade-Offs	206
14.9	Choosing Other Test Double Frameworks	208
14.10	What We've Learned	210
15	Tools and Integration	213
15.1	The rspec Command	213
15.2	TextMate	218
15.3	Autotest	219
15.4	Rake	219
15.5	RCov	221
15.6	What We've Learned	222
16	Extending RSpec	223
16.1	Metadata	223
16.2	Configuration	224
16.3	Filtering	224
16.4	Extension Modules	227
16.5	Global Hooks	228
16.6	Mock Framework	228
16.7	Custom Matchers	229
16.8	Macros	233
16.9	Custom Formatters	236
16.10	What We've Learned	239
IV	Cucumber	241
17	Intro to Cucumber	243
17.1	From 20,000 Feet	244
17.2	Features	244
17.3	Customer Acceptance Tests	248
17.4	Gherkin	248
17.5	Scenarios	250

17.6	Steps	251
17.7	The cucumber Command	252
17.8	Given/When/Then	253
17.9	Declarative and Imperative Scenario Styles	254
17.10	Organizing Features	255
17.11	Tags	256
17.12	What We've Learned	258
18	Cucumber Detail	261
18.1	Step Definitions	261
18.2	World	263
18.3	Calling Steps Within Step Definitions	264
18.4	Hooks	265
18.5	Background	267
18.6	Multiline Text	268
18.7	Tables in Steps	269
18.8	Scenario Outlines	270
18.9	Configuration	272
18.10	What We've Learned	272
V	Behaviour-Driven Rails	275
19	BDD in Rails	277
19.1	Outside-In Rails Development	278
19.2	Setting Up a Rails 3 Project	280
19.3	Setting Up a Rails 2 Project	283
19.4	What We've Learned	283
20	Cucumber with Rails	285
20.1	Step Definition Styles	285
20.2	Direct Model Access	287
20.3	What We've Learned	292
21	Simulating the Browser with Webrat	293
21.1	Writing Simulated Browser Step Definitions	294
21.2	Navigating to Pages	299
21.3	Manipulating Forms	303
21.4	Specifying Outcomes with View Matchers	308
21.5	Building on the Basics	311
21.6	What We've Learned	314

22	Automating the Browser with Webrat and Selenium	315
22.1	Getting Started	316
22.2	Writing Step Definitions for Selenium	319
22.3	Debugging Selenium Issues	323
22.4	What We've Learned	325
23	Rails Views	327
23.1	Writing View Specs	327
23.2	Mocking Models	330
23.3	Specifying Helpers	337
23.4	When Should I Write View Specs?	339
23.5	What We've Learned	339
24	Rails Controllers	341
24.1	Controller Specs	341
24.2	Context-Specific Examples	348
24.3	Specifying ApplicationController	357
24.4	What We've Learned	359
25	Rails Models	361
25.1	Writing Model Specs	361
25.2	Specifying Business Rules	369
25.3	Exercise	378
25.4	Useful Tidbits	379
25.5	What We've Learned	382
A	RubySpec	385
A.1	The Project	386
A.2	Syntax	386
A.3	Guards	388
A.4	Extensibility	389
A.5	MSpec	391
A.6	Tags	391
A.7	Community	392
B	RSpec's Built-in Expectations	395
C	Bibliography	399
	Index	401