Extracted from:

Effective Testing with RSpec 3

Build Ruby Apps with Confidence

This PDF file contains pages extracted from *Effective Testing with RSpec 3*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

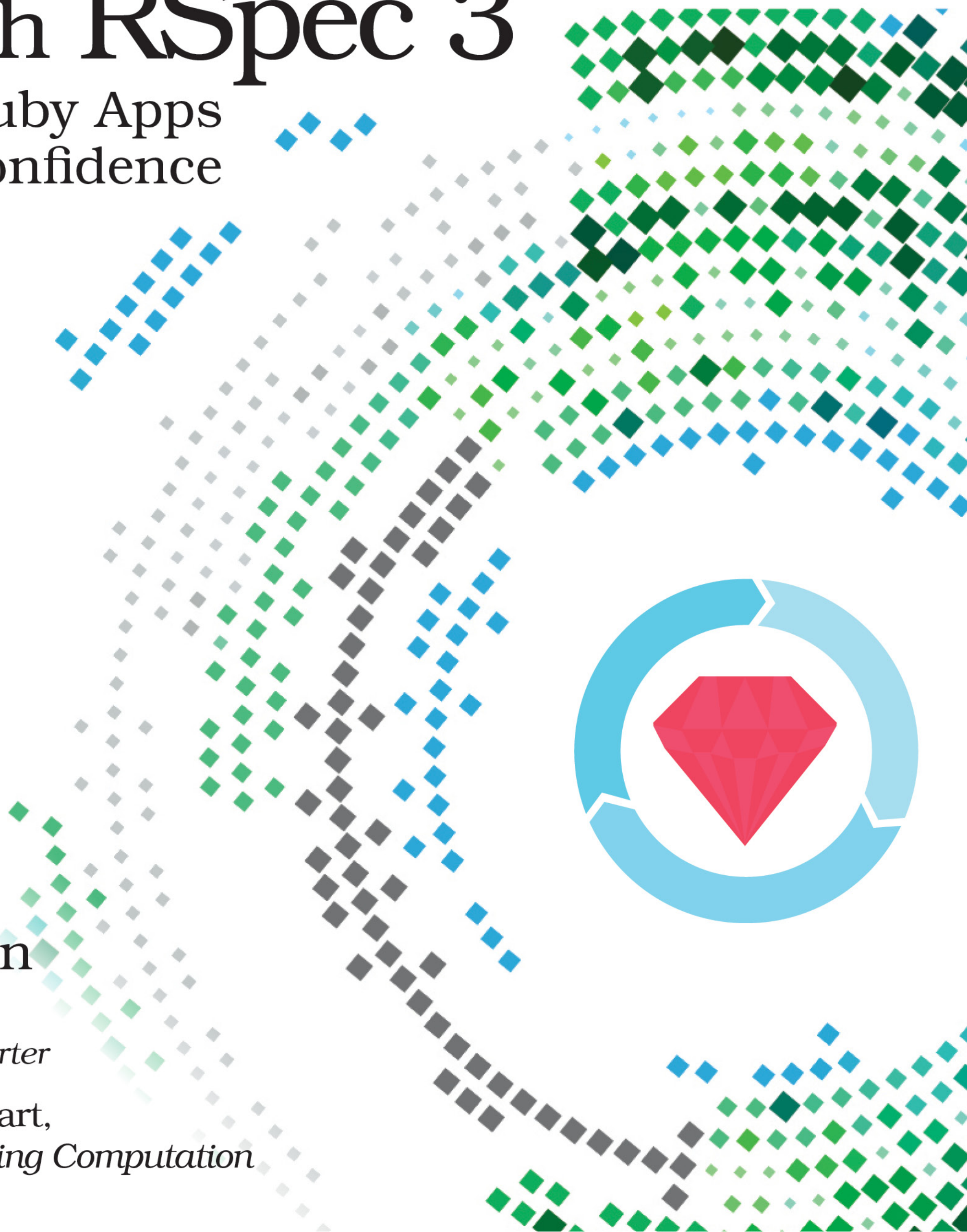
Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

Effective Testing with RSpec 3

Build Ruby Apps with Confidence



Myron Marston
and Ian Dees
edited by Jacquelyn Carter

Foreword by Tom Stuart,
author of *Understanding Computation*

Effective Testing with RSpec 3

Build Ruby Apps with Confidence

Myron Marston
Ian Dees

The Pragmatic Bookshelf
Raleigh, North Carolina

Pragmatic Bookshelf

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Executive Editor: Susannah Davidson Pfalzer
Development Editor: Jacquelyn Carter
Indexing: Potomac Indexing, LLC
Copy Editor: Liz Welch
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

Printed in the United States of America.
ISBN-13: 978-1-68050-198-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—August 2017

Foreword

Making software is hard, and it's difficult to know how to get better at it. There's always so much stuff to learn, and so many competing opinions about the right way to do things. And yet somehow here you are, about to read another book full of ideas you're apparently supposed to remember.

Well, the truth is that you can make software just fine without writing tests, and you can write tests just fine without using RSpec, so you'll be okay if you stop reading now. But if you decide to keep going, you'll discover something interesting: writing tests with RSpec is a great way to get really good at making software.

That's because RSpec isn't just a testing framework. It's a tool for learning how to think critically, patiently, and systematically about the design of your code, and how to make software in a methodical way so you have confidence that it's well organized, clear, and correct.

It's also the product of an ongoing experiment. Over the last decade or so, RSpec's users have refined the way they write programs and the strategies they use to test them, and RSpec itself has continually changed to reflect these emerging decisions. In recent years it's settled down into a stable, crystalline form: a kind of executable document of what all those programmers learned from their iterated successes and failures.

As a result, each feature in RSpec is there for a good reason. It was put there by a developer like you, to address a problem like the ones you face every time you write code. There's an art to appreciating this gallery of features as a unified whole and settling into the state of mind it wants you to adopt. You could spend months memorizing all of RSpec's methods and options and fiddly bits, yet still be none the wiser about how to use your tests to drive your implementation decisions and steer your design toward simplicity.

After all, the big challenge of test-driven development isn't knowing how to *write* tests, but knowing how to *listen* to them. For your tests to be worth the bytes they're written in, they must be able to speak to you about how well the underlying program is designed and implemented—and, crucially, you must be able to hear them. The words and ideas baked into RSpec are carefully chosen to heighten your sensitivity to this kind of feedback. As you play with its expressive little pieces you'll develop a taste for what a good test looks like, and the occasional stumble over a test that now seems harder or uglier or way more annoying than necessary will start you on a journey of discovery that leads to a better design for that part of your codebase.

Myron and Ian have excellent taste and a keen ear for the feedback that tests can provide. This book unpacks the hard-won wisdom of RSpec's feature set and translates it into gentle, practical, actionable advice for your own code. By the time you've completely internalized everything here, you'll have acquired that elusive meta-skill: judgment, the ability to make your own decisions about how to do things right.

Tom Stuart

London, May 2017

Author of *Understanding Computation*