Foundations of Agile Python Development

■■■

Jeff Younker

Apress®

Foundations of Agile Python Development

Copyright © 2008 by Jeff Younker

ISBN-13 (pbk): 978-1-59059-981-5

ISBN-10 (pbk): 1-59059-981-0

ISBN-13 (electronic): 978-1-4302-0636-1

ISBN-10 (electronic): 1-4302-0636-5

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Tom Welsh
Technical Reviewer: Will McGugan
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Susannah Davidson Pfalzer
Copy Editor: Damon Larson
Associate Production Director: Kari Brooks-Copony
Production Editor: Elizabeth Berry
Compositor: Dina Quan
Proofreaders: Nancy Bell, April Eddy
Indexer: John Collin
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

The source code for this book is available to readers at http://www.apress.com.

Contents at a Glance

Contents

About the Author

JEFF YOUNKER is chief engineer of Data-Pipes (`www.data-pipes.com/`). His educational background carefully avoided computers, but he was drawn in anyway. Most of his misguided adulthood has been spent in large installation systems administration, tool smithing, and release engineering, with a peculiar obsession involving both monitoring and rapid deployment. Over the last several years, he's had the pleasure of working with Python full time. Having escaped Texas nearly a decade ago, he now lives in gray and rainy Northern California. When not suffering monitor-induced radiation burns, Jeff likes to do anything that doesn't involve a roof, unless the roof has been top-roped or covers a machine shop.

About the Technical Reviewer

WILL McGUGAN is a software developer and author currently working in London on a social networking site for games built with Django. See his blog at `www.willmcgugan.com/` for more information on Will's work and open source projects.

Acknowledgments

I'd like to thank Apress and Jason Gilmore for giving me the opportunity to write this book. Thanks to Tom Welsh for the unfaltering criticism that has made it something more than my incoherent ravings. I thank Susannah Davidson Pfalzer for playing midwife to Jason's child after he left the Apress family. This has been a huge undertaking. If I had really understood the magnitude, I might not have started to begin with, but Susannah kept everything on track and graciously coped with the events in my life impinging on the schedule.

Thanks to Will McGugan for the many improvements he made in the code, and for his meticulous attention to detail. Damon Larson did an amazing job turning my technobabble into coherent English. It was wonderful working with Liz Berry in the last few weeks of madness as the disjointed word-processed files turned into something that looks suspiciously like a book.

A whole slew of friends came out of the woodwork as they discovered that I was writing this book, and I was astonished to discover that they actually wanted to help. The last half of this book is much richer for the efforts of Matt Ho, Jacob Hoffman-Andrews, Nancy Huntingford, Rachel McConnell, and Erik Ziko.

In particular, Matt Ho's assistance with Chapter 11 was indispensable. I don't know if I could have finished this book on time without the days we spent in his living room. At a point where I was completely lost, his advice made the path clear. I wish we'd talked much earlier in the process.

My business partner David Birkhead bought time with our customers to give me an opportunity to finish this project, and I owe him a huge debt for that. He's been hugely supportive outside of our business relationship, too. I have to thank our customers Cynthia Walston and David Alban for being so supportive and accommodating during the last month of this process.

I owe a huge debt of gratitude to my father, William James Younker, who passed away somewhere around Chapter 5. No matter how poor my family was, he always saw to it that I had the tools to follow my curiosity. I wish I could show him this book you're reading. At least I'll get to show it to my mother, Hallie Younker, who did an astoundingly good job of bringing me up given the circumstances that enveloped our lives.

I'd like to thank my teachers, too—I had some amazing ones. I'd like to thank Adréa Shaw for giving me such wonderful anecdotes about Baylor College of Medicine. I remember the soft glow of the electron microscope fondly. I'd like to thank David Raikow. He writes more than anyone I know, and his advice kept me moving along when I got stuck. I'd like to give my thanks to Ethyl the Dog, too. She was remarkably good at alleviating writer's block.

I've received support in other ways, as well. Roxanne Williams and Sören Ragsdale lent me technical assistance when necessary. Blair Miller kept me from going insane and from time to time made sure that I stayed fed. Scott Calvert, Gwen Perry, and David Cutler made sure that I wasn't completely isolated from my friends. Kathryn Keslosky's assistance with revisions was immensely helpful.

My dear friends Jaron and Cherry Rothkop made sure that my time in Toledo was bearable. Clare and John Vrakking put up with my mental absence from our first Christmas together in over a decade. I'm sorry this book didn't let me make Passover this year. Finally, I don't know if I could have completed this book without Amy Woodward's love and support. Her calming voice and compassionate words kept me from coming unglued when I became immobilized by the seemingly unending tide of drafts and revisions. She assisted me emotionally and materially, and she believed in me more than I believed in myself at times. I owe her a debt of gratitude that I don't know if I can ever repay. A thousand love songs do her no justice.

Introduction

If you're embarking on a Python development project, then you should buy this book—there's nothing quite like it. I know this because I was looking for it last year, and I couldn't find it. This book introduces the tools you'll need to get started on agile projects in Python, and unlike any other book out there, it shows you how to tie them all together.

Sure, there are many good books on agile development. A lot of them cover the development processes in great detail, and this is a good thing. Agile development is very much about human interactions and the environment surrounding software development, but there is a whole ecology of tooling to make everything work at a practical level.

Agile development eschews extensive up-front specification, and it anticipates that the product will constantly change, but it puts in place rigorous checks to compensate for anticipated change. Testing is an integral part of agile development from the very start, and it is pursued with ferocious rigor. You need software tools to facilitate testing.

Agile projects have very short release cycles, and this has implications for tooling, too. There's no way to have two-week release cycles if it takes you days to integrate changes, days to perform QA, and days to package and deploy the software. This means that agile development puts a high value on build and release automation.

While agile development techniques can be applied to any project, both testing tools and build automation tend to be very language specific. These tools do exist in Python. They're widely available, and by and large they're free, too, but the documentation tends to be . . . um . . . spotty. And while there may be documentation on the individual tools, the documentation telling you how to tie these tools together is usually sparse to nonexistent. This book provides that missing documentation.

Who This Book Is For

This book is written for a person who knows how to program and is already familiar with Python. If you have some Python under your belt and you're thinking of starting a new project, but you don't know how to get started, then this book is for you. If you're an experienced Python programmer and you want to give this agile stuff a whirl, then this book is for you. If you're a release engineer who has been thrown headlong into the world of Python, then this book is for you, too. If you're brand new to programming or don't really know Python, this is not the best book to start with. There are some wonderful books out there that will introduce you to the language, but this isn't one of them.

What's Really in Here?

Each chapter in this book addresses a different aspect of tooling in an agile development environment. These are collected roughly into two parts, with the first focusing on basic tooling, and the second focusing on specific practices. If you're already familiar with Subversion, Setuptools, and Buildbot, then you should have no problem jumping between Chapters 6 through 11. If you're not, then you'll want to look at the earlier chapters first.

Chapter 1: What Is Agile Development?

Chapter 1 provides an overview of the methods that characterize agile development methodologies, with a focus on those not directly related to tooling.

Chapter 2: The IDE: Eclipsing the Command Line

This book uses the command line throughout, but modern IDEs provide many benefits. This chapter introduces you to Python development using Eclipse and the Pydev plug-in.

Chapter 3: Revision Control: Subverting Your Code

A revision control system is part of the core infrastructure for any agile development environment. Subversion is an excellent choice. I show you how to use it from the command line and from Eclipse using the Subversive plug-in.

Chapter 4: Setuptools: Harnessing Your Code

You can't replicate your work for testing purposes without some sort of a framework. In Python, a natural choice is Setuptools, which provides a solid basis for automated builds.

Chapter 5: A Build for Every Check-In

Automated build systems form the core of a continuous integration system. Here I introduce Buildbot, an excellent system that happens to be written in Python. It ensures that the code you check in builds correctly.

Chapter 6: Testing: The Horse and the Cart

Unit testing ensures that your code runs as you expect it to, and it prevents regression (reappearance of old bugs) when you change existing code. I introduce the unit-testing packages unittest and Nose, and I show how to use Nose to run tests from within Eclipse and Setuptools. Finally, I show how to link them into Buildbot.

Chapter 7: Test-Driven Development and Impostors

Test-driven development (TDD) is the practice of writing tests before writing the code they test. Imposters (a.k.a. mock objects) provide a powerful unit-testing technique to isolate units of code. I examine two mock object frameworks, pMock and PyMock, and I work through a sizable example to show how TDD, refactoring, and imposters are used, and how they affect the code that you produce with them.

Chapter 8: Everybody Needs Feedback

Improving your code requires feedback—useful information that sometimes comes from your coworkers, and sometimes from software. Accurate feedback requires standards. This chapter looks at code coverage, complexity measures, and development velocity. It also examines coding standards, how they can be enforced from within Eclipse, and how you can prevent bad code from reaching your repository by using Subversion pre-commit hooks.

Chapter 9: Databases

Databases are very widely used these days, and they pose their own special challenges for agile development. This chapter examines the object-relational mappers SQLObject and SQLAlchemy, and then examines how to version databases using the DBMigrate tool.

Chapter 10: Web Testing

The web is everywhere, and web development has its own set of issues. This chapter examines general approaches to testing web applications, and introduces HTML/XML verification using `ElementTree` and `BeautifulSoup`. It also looks into JavaScript unit testing with JsUnit.

Chapter 11: Functional Testing

This chapter examines functional testing with a particular emphasis on acceptance testing using PyFit. The chapter shows how to use PyFit, and more importantly, how to tie PyFit into Setuptools and Buildbot. (In my view, this alone is worth the price of the book.)

Contacting Me

Finally, please don't hesitate to give me feedback on the book at any time. This is my first book, my writing ability has improved immensely as the book has progressed, and I now have a much better understanding of what I wanted to say than when I started. I'll try to improve any sections that people find lacking and publish them to this book's web page at `http://www.apress.com/book/view/9781590599815`. Additional materials may be available on my blog (`www.theblobshop.com/blog`) under the tag `famip`. I'll present more information in these locations as it becomes available. This pertains but is not limited to notes about anything that I've fouled up, new thoughts, and additional materials that I think you may find useful.