

O'REILLY®



Learning Agile

学习敏捷开发 (影印版)

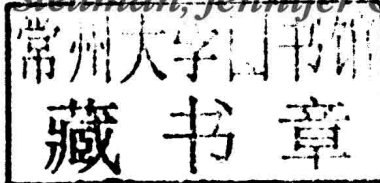
东南大学出版社

Andrew Stellman, Jennifer Greene 著

学习敏捷开发 (影印版)

Learning Agile

Andrew Stellman, Jennifer Greene 著



Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目(CIP)数据

学习敏捷开发: 英文/(美)斯特尔曼(Stellman, A.), (美)格林(Greene, J.)著. —影印本. —南京: 东南大学出版社, 2015.8

书名原文: Learning Agile

ISBN 978-7-5641-5919-1

I. ①学… II. ①斯… ②格… III. ①软件开发—英文 IV. ①TP311.52

中国版本图书馆 CIP 数据核字(2015)第 165681 号

江苏省版权局著作权合同登记

图字: 10-2014-163 号

© 2014 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2015. Authorized reprint of the original English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2014。

英文影印版由东南大学出版社出版 2015。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

学习敏捷开发(影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seupress.com

印 刷: 常州市武进第三印刷有限公司

开 本: 787 毫米×980 毫米 16 开本

印 张: 26.25

字 数: 514 千字

版 次: 2015 年 8 月第 1 版

印 次: 2015 年 8 月第 1 次印刷

书 号: ISBN 978-7-5641-5919-1

定 价: 76.00 元

本社图书若有印装质量问题, 请直接与营销部联系。电话(传真): 025-83791830

Praise for *Learning Agile*

Another amazing book by the team of Andrew and Jennifer. Their writing style is engaging, their mastery of all things agile is paramount, and their content is not only comprehensive, it's wonderfully actionable.

—Grady Booch
IBM Fellow

The biggest obstacle to overcome in building a high-performance agile team is not learning how, but learning why. Helping teams discover the why is the key to unlock their potential for greater commitment and more creative collaboration. With a focus on values and principles Andrew and Jennifer have provided an outstanding tool to help you and your team discover the why. I can't wait to share it.

—Todd Webb
Technical Product Leader at a global e-commerce company

While I was already sold on agile, what I learned from *Learning Agile* will help my efforts to sell agile across my organization. This book provides more than “just” an engaging way to gain a deep understanding of agile's principles and practices. The easily relatable stories will help make agile compelling to members across your team, so you can begin reaping its rewards.

—Mark Denovich
Senior Business Consultant and Head of US development, Centriq Group

An excellent guide for any team member looking to deepen their understanding of agile. Stellman and Greene cover agile values and practices with an extremely clear and engaging writing style. The humor, examples, and clever metaphors offer a refreshing delivery. But where the book really shines is how it pinpoints frequent problems with agile teams, and offers practical advice on how to move forward to achieve deeper results.

—Matthew Dundas
CTO, Katori

As an engineer, I always thought the problems that Agile practices help to solve are a direct hit for the industry. As it turns out, becoming Agile is hard; it's more than just the practices. A piecemeal approach to Agile gives, as the authors call it, "better-than-not-doing-it" results. If you are just getting started, or Agile is only "better-than-not-doing-it", Andrew and Jennifer have a lot of practical advice on how to read between the lines of the Agile Manifesto and really become Agile.

—James W Grenning

Founder of Wingman Software and co-author of the Agile Manifesto

Andrew Stellman and Jennifer Greene have done an impressive job putting together a comprehensive, practical resource that is easily accessible for anyone who is trying to 'get' Agile. They cover a lot of ground in *Learning Agile*, and have taken great care to go beyond simply detailing the behaviors most should expect of Agile teams. In exploring different elements of Agile, the authors present not just the standard practices and desired results, but also common misconceptions, and the positive and negative results they may bring. The authors also explore how specific practices and behaviors might impact individuals in different roles. This book is a great resource for new and experienced Agile practitioners alike.

—Dave Prior PMP CST PMI-ACP
Agile Consultant and Trainer

If you want to learn about any of the specific approaches to agile, you need to read the specific relevant books. That means you know what you want to do in advance. Not very agile of you, is it? What Andrew and Jenny have done is create an approachable, relatable, understandable compendium of what agile is. You don't have to decide in advance what your agile approach is. You can read about all of them, and then decide. On your way, you can learn the system of agile and how it works.

—Johanna Rothman
Author and Consultant, www.jrothman.com

The culture of a software development team often has a greater impact than their expertise or tools do on the success of their project. Stellman and Greene's advice on how to transform an assortment of fragmented individual perspectives into a collaborative unit with shared values and practices should help any software manager regardless of the organization's official methodology. Their comparison of Scrum, XP, Lean, and Kanban techniques analyze the many ways in which Agile principles can be applied. The entertaining case studies illustrate the human dilemmas—and the rewards—of learning to become Agile.

—Patricia Ensworth
President, Harborlight Management Services LLC

Learning Agile is thorough, approachable, practical, and interesting. The values, principles, and methodologies explained in this book are thought-provoking and illuminating, and I look forward to applying them to my team's work.

—*Sam Kass*

Software architect and tech lead in the financial sector

Andrew Stellman and Jennifer Greene have been there, seen that, bought the T-Shirt, and now written the book! This is a truly fantastic introduction to the major Agile methodologies for software professionals of all levels and disciplines. It will help you understand the common pitfalls faced by development teams, and learn how to avoid them.

—*Adam Reeve*

Engineer and team lead at a major social networking site

*For Nisha and Lisa,
who have been very patient with us*

Foreword

It seems that people always need something to debate. Was Van Halen better with David Lee Roth or Sammy Hagar? Pepsi or Coke? Lennon or McCartney? Cats or dogs? One such debate in the early days of agile was principles versus practices. Early agilists agreed on a set of principles enshrined in the Agile Manifesto, and many practices were shared across multiple agile approaches. However, there was fierce debate about whether a team should start by understanding the principles of agile software development or whether they should begin by performing the practices even before developing a deep understanding of why.

Proponents of starting with practices took a wax-on/wax-off view of the world. If a team were to act agile, they would be agile. By going through the motions of being agile—pair programming, automating tests and builds, using iterations, working closely with a key stakeholder, and so on—a team would gradually develop an understanding of the principles of agile.

Proponents of starting with principles, on the other hand, contended that practices without principles were hollow. Going through the motions without understanding why did not lead to agility. Agility was (and still is) a focus on continuous improvement. The argument went that a team could not improve continuously if they didn't understand why they were doing the things they were doing.

In *Learning Agile*, Andrew Stellman and Jennifer Greene do the best job I've encountered of stressing the principles of agile without de-emphasizing its practices. They point out that following practices without knowing why is likely to lead only to what they call a "better-than-not-doing-it" level of success. That is, implementing practices alone is helpful, but it falls far short of the true promise of what becoming agile can truly deliver.

I first met Andrew and Jennifer six years ago when they interviewed me for their *Beautiful Teams* book. Although that book does not include *agile* in its title, in many ways the book was about agile. A team that has embraced the principles of agile, mastered the practices it needs, and discarded practices it found unnecessary is truly a

beautiful team. In *Learning Agile*, Andrew and Jennifer focus their discussion on agile by concentrating on today's three most common agile methods—Scrum, Extreme Programming, and Kanban. You'll see how their shared principles have led to different practices within each approach. For example, if you've wondered why Scrum requires end-of-sprint retrospectives but Extreme Programming does not, you'll find the answer here.

In joining Andrew and Jennifer through their exploration of Scrum, Extreme Programming, Lean, and Kanban, you'll read lots of stories. This makes sense—after all, a common practice for many agile teams is telling user stories to describe what a system's users want. You'll meet teams struggling to build the right functionality, taking too long to deliver last year's requirements, mistaking agile for just another form of command-and-control management, being whipsawed by change rather than embracing it, and more. More importantly, you'll read how those teams overcame those problems and how you can, too.

Learning Agile puts an end, once and for all, to the question of which should come first, practices or principles. Its engaging stories and discussions illustrate the simple truth that in agile there can be no separation between principles and practices. In these pages, you'll gain a deeper understanding of how to get started (or get back on track) on your journey to becoming a genuinely beautiful team.

Mike Cohn

Author, *Succeeding with Agile*

Boulder, Colorado

Acknowledgments

We wrote this book to help you learn agile—but we didn’t do it alone. First and foremost, we want to thank our fantastic editor, Mary Treseler. She championed this project from the day we first discussed it with her in an Indian restaurant in downtown Manhattan all the way through to the finished book that you’re reading today. She’s been an important part of everything we’ve done with O’Reilly, and we couldn’t have done this without her.

We’d also like to thank other friends at O’Reilly, without whom this would not be possible: Mike Hendrickson, Laurie Petrycki, Tim O’Reilly, Ally MacDonald, Andy Oram, Nicole Shelby, and especially Marsee Henon, Sara Peyton, and all of the fantastic press and PR folks in Sebastopol.

We want to thank Mike Cohn for his wonderful foreword, as well as the really good advice he’s given us over the years. We also want to thank him for his great books, because we learned a lot from them! We’d also like to thank David Anderson for giving us some really excellent feedback on Chapters 8 and 9. We want to thank Grady Booch, Scott Ambler, James Grenning, Scott Berkun, Steve McConnell, Karl Wiegers, Johanna Rothman, Patricia Ensworth, Tommy Tarka, Keoki Andrus, Neil Siegel, Karl Fogel, and Auke Jilderda for giving us some really good material and input over the years, especially in *Beautiful Teams*—and especially Barry Boehm, not only for contributing a really fantastic story to that book, but more importantly for laying the intellectual groundwork that much of agile is built on. And we want to thank Kent Beck, Alistair Cockburn, Ken Schwaber, Jeff Sutherland, Ron Jeffries, Tom Poppendieck, Mary Poppendieck, Lyssa Adkins, and Jim Highsmith for their groundbreaking work in agile. We literally would not have been able to write this book without them.

We also want to thank all of our technical reviewers for their excellent feedback and thorough review: Faisal Jawdat, Adam Reeve, Anjanette Randolph, Samuel Weiler, Dave Prior, Randy DeFauw, Todd Webb, Michael DeWitt, and Paul Ellarby.

And finally, we'd like to thank the hundreds of software team members who have been kind enough to talk to us about their problems, solutions, stories, and experiences over the years.

Andrew would like to thank Lisa Kellner. He'd also like to thank everyone in the Computer Science department at Carnegie Mellon who helped him learn some really important ideas, especially Bob Harper, Mark Stehlik, and Randy Bryant. He'd like to thank Tony Visconti, who's been a true mentor and a friend over the years. He'd like to thank his friends Sara Landeau, Greg Gassman, Juline Koken, Kristeen Young, and Casey Dunmore—thinking back, it's kind of amazing how much he's learned about teamwork from these great musicians. He'd also like to thank some really fantastic teammates he's had over his career, including Dan Faltyn, Ned Robinson, Debra Herschmann, Mary Gensheimer, Lana Sze, Warren Pearson, Bill DiPierre, Jonathan Weinberg, and Irene O'Brien. And last but not least, he'd like to thank the two best software teams he's ever worked with, especially Mark Denovich, Eric Renkey, and Chris Winters from Optiron, and Mike Hickin, Nick Lai, Sunanda Bera, and Rituraj Deb Nath from Bank of America.

Jenny would like to thank Nisha Sondhe. She'd also like to thank Christopher Wenger, Brian Romeo, LaToya Jordan, Mazz Swift, Rekha Malhotra, Courtney Nelson, Anjanette Randolph, Shona McCarthy, Ethan Hill, Yeidy Rodriguez, Kyle Mosier, Achinta McDaniel, Jaikaran Sawhny, and Kit Cole for all of their support, laughter, and shenanigans. She'd like to thank her family for their patience and encouragement during the nearly three years this book was under construction. She'd like to thank Tanya and Dilan Desai for their support and help. Finally, she'd like to thank the many colleagues she's learned from over the years. There are too many people who deserve thanks to list them all, but here are a few: Joe Madia, Paul Oakes, Jonathan Weinberg, Bianka Buschbeck, Thor List, Oleg Fishel, Brian Duperrouzel, Dave Murdock, Flora Chen, Danny Wunder, David San Filippo, and Rasko Ristic.

Safari® Books Online



Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-

Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at <http://bit.ly/learning-agile>.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Table of Contents

Foreword.....	xiii
Preface.....	xv
1. Learning Agile.....	1
What Is Agile?	2
Who Should Read This Book	7
Our Goals for This Book	8
Getting Agile into Your Brain by Any Means Necessary	8
How This Book Is Structured	12
2. Understanding Agile Values.....	15
A Team Lead, Architect, and Project Manager Walk into a Bar...	16
No Silver Bullet	19
Agile to the Rescue! (Right?)	22
A Fractured Perspective	26
The Agile Manifesto Helps Teams See the Purpose Behind Each Practice	33
Understanding the Elephant	39
Where to Start with a New Methodology	45
3. The Agile Principles.....	51
The 12 Principles of Agile Software	52
The Customer Is Always Right...Right?	53
Delivering the Project	55
Communicating and Working Together	64
Project Execution—Moving the Project Along	74
Constantly Improving the Project and the Team	78
The Agile Project: Bringing All the Principles Together	81

4. Scrum and Self-Organizing Teams.....	87
The Rules of Scrum	89
Act I: I Can Haz Scrum?	92
Everyone on a Scrum Team Owns the Project	94
Act II: Status Updates Are for Social Networks!	109
The Whole Team Uses the Daily Scrum	110
Act III: Sprinting into a Wall	119
Sprints, Planning, and Retrospectives	120
Act IV: Dog Catches Car	129
5. Scrum Planning and Collective Commitment.....	137
Act V: Not Quite Expecting the Unexpected	138
User Stories, Velocity, and Generally Accepted Scrum Practices	140
Act VI: Victory Lap	159
Scrum Values Revisited	160
6. XP and Embracing Change.....	175
Act I: Going into Overtime	176
The Primary Practices of XP	178
Act II: The Game Plan Changed, but We're Still Losing	187
The XP Values Help the Team Change Their Mindset	189
An Effective Mindset Starts with the XP Values	195
Act III: The Momentum Shifts	200
Understanding the XP Principles Helps You Embrace Change	201
7. XP, Simplicity, and Incremental Design.....	219
Act IV: Going into Overtime, Part 2: Second Overtime	220
Code and Design	222
Make Code and Design Decisions at the Last Responsible Moment	237
Incremental Design and the Holistic XP Practices	245
Act V: Final Score	260
8. Lean, Eliminating Waste, and Seeing the Whole.....	269
Lean Thinking	270
Act I: Just One More Thing...	279
Creating Heroes and Magical Thinking	280
Eliminate Waste	283
Gain a Deeper Understanding of the Product	288
Deliver As Fast As Possible	295
9. Kanban, Flow, and Constantly Improving.....	315
Act II: Playing Catch-Up	317

The Principles of Kanban	318
Improving Your Process with Kanban	325
Measure and Manage Flow	339
Emergent Behavior with Kanban	358
10. The Agile Coach.....	369
Act III: Just One More Thing (Again?!)...	371
Coaches Understand Why People Don't Always Want to Change	371
Coaches Understand How People Learn	376
Coaches Understand What Makes a Methodology Work	380
The Principles of Coaching	382
Index.....	387

Learning Agile

The most important attitude that can be formed is that of desire to go on learning.
—John Dewey, *Experience and Education*

It's an exciting time to be agile! For the first time, our industry has found a real, sustainable way to solve problems that generations of software development teams have been struggling with. Here are just a few of the solutions that agile promises:

- Agile projects come in on time, which is great for teams that have struggled with projects delivered very late and far over budget.
- Agile projects deliver high-quality software, which is a big change for teams that have been bogged down by buggy, inefficient software.
- Code built by agile teams is well constructed and highly maintainable, which should be a relief to teams accustomed to maintaining convoluted and tangled spaghetti code.
- Agile teams make their users happy, which is a huge change from software that fails to deliver value to the users.
- Best of all, developers on an effective agile team find themselves working normal hours, and are able to spend their nights and weekends with their friends and families—possibly for the first time in their careers.

Agile is popular because many teams that have “gone agile” report great results: they build better software, work together better, satisfy their users, and do it all while having a much more relaxed and enjoyable working environment. Some agile teams seem to have finally made headway in fixing problems that have vexed software teams for decades. So how do great teams use agile to build better software? More specifically, how can *you* use agile to get results like this?

In this book, you will learn about the two most popular agile methodologies, Scrum and Extreme Programming (XP). You'll also learn about Lean and Kanban, and how they help you understand the way you build software today and help you evolve to a better state tomorrow. You'll learn that while these four agile schools of thought focus on different areas of software development, they all have one important thing in common: they focus on **changing your team's mindset**.

It's that mindset shift that takes a team from superficially adding a few token agile practices to one that has genuinely improved the way it builds software. The goal of this book is to help you learn both sides of agile: the practices that make up the day-to-day work, and the values and principles that help you and your team fundamentally change the way that you think about building software.

What Is Agile?

Agile is a **set of methods and methodologies** that help your team to think more effectively, work more efficiently, and make better decisions.

These methods and methodologies address all of the areas of traditional software engineering, including project management, software design and architecture, and process improvement. Each of those methods and methodologies consists of **practices** that are streamlined and optimized to make them as easy as possible to adopt.

Agile is *also* a **mindset**, because the right mindset can make a big difference in how effectively a team uses the practices. This mindset helps people on a team share information with one another, so that they can make important project decisions together—instead of having a manager who makes all of those decisions alone. An agile mindset is about opening up planning, design, and process improvement to the entire team. An agile team uses practices in a way where everyone shares the same information, and each person on the team has a say in how the practices are applied.

The reality of agile for many teams that have not had as much success is quite different from its promise, and the key to that difference is often the mindset the team brings to each project. The majority of companies that build software have experimented with agile, and while many of them have found success, some teams have gotten less-than-stellar results. They've achieved some improvement in how they run their projects—enough to make the effort to adopt agile worth it—but they haven't seen the substantial changes that they feel agile promised them. This is what that mindset shift is all about; “going agile” means helping the team find an effective mindset.

But what does “mindset shift” really mean? If you're on a software team, then what you do every day is plan, design, build, and ship software. What does “mindset” have to do with that? As it turns out, the practices you use to do your everyday work depend a lot on the attitude that you and your teammates have toward them.