

The Art of

/THEORY/IN/PRACTICE

Agile Development

敏捷开发艺术 (影印版)



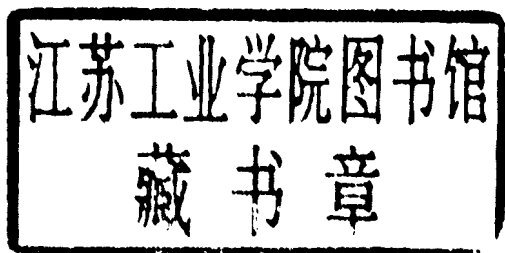
O'REILLY®

東南大學出版社

James Shore & Shane Warden 著

The Art of Agile Development

敏捷开发艺术 (影印版)



James Shore and Shane Warden

O'REILLY®

Beijing • Cambridge • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

敏捷开发艺术: 英文 / (美) 肖尔 (Shore, J.), (美) 活登 (Warden, S.) 著. — 影印本. — 南京: 东南大学出版社, 2008.8

书名原文: The Art of Agile Development

ISBN 978-7-5641-1241-7

I. 敏… II. ①肖… ②活… III. 软件开发 — 英文
IV. TP311.52

中国版本图书馆 CIP 数据核字 (2008) 第 104533 号

江苏省版权局著作权合同登记

图字: 10-2008-129 号

©2007 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2008. Authorized reprint of the original English edition, 2007 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 2007。

英文影印版由东南大学出版社出版 2008。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。

敏捷开发艺术

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 汉

网 址: <http://press.seu.edu.cn>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 27 印张

字 数: 454 千字

版 次: 2008 年 8 月第 1 版

印 次: 2008 年 8 月第 1 次印刷

书 号: ISBN 978-7-5641-1241-7/TP · 210

印 数: 1~2500 册

定 价: 59.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

The Art of Agile Development

敏捷开发艺术(影印版)

To our families.

Preface

Q: How do you get to Carnegie Hall?

A: Practice, man, practice!

We want to help you master the art of agile development.

Agile development, like any approach to team-based software development, is a fundamentally *human* art, one subject to the vagaries of individuals and their interactions. To master agile development, you must learn to evaluate myriad possibilities, moment to moment, and intuitively pick the best course of action.

How can you possibly learn such a difficult skill? Practice!

First and foremost, this book is a detailed description of *one* way to practice agile development: Extreme Programming (XP). It's a practical guide that, if followed mindfully, will allow you to successfully bring agile development in the form of XP to your team—or will help you decide that it's not a good choice in your situation.

Our second purpose is to help you master the art of agile development. Mastering agility means going beyond our cookbook of practices. Agile development is too context-sensitive for one approach to be entirely appropriate, and too nuanced for any book to teach you how to master it. Mastery comes from within: from experience and from an intuitive understanding of ripples caused by the pebble of a choice.

We can't teach you how your choices will ripple throughout your organization. We don't try. *You* must provide the nuance and understanding. This is the only way to master the art. Follow the practices. Watch what happens. Think about why they worked... or didn't work. Then do them again. What was the same? What was different? Why? Then do it again. And again.

At first, you may struggle to understand how to do each practice. They may look easy on paper, but putting some practices into action may be difficult. Keep practicing until they're easy.

As XP gets easier, you will discover that some of our rules don't work for you. In the beginning, you won't be able to tell if the problem is in our rules or in the way you're following them. Keep practicing until you're certain. When you are, break the rules. Modify our guidance to work better for your specific situation.

Parts I and II of this book contain our approach to XP. Part I helps you get started with Extreme Programming; Part II provides detailed guidance for each of XP's practices. Parts I and II should keep you occupied for many months.

When you're ready to break the rules, turn to Part III. A word of warning: there is nothing in Part III that will help you practice XP. Instead, it's full of ideas that will help you understand XP and agile development more deeply.

One day you'll discover that rules no longer hold any interest for you. After all, XP and agile development aren't about following rules. "It's about simplicity and feedback, communication and trust," you'll think. "It's about delivering value—and having the courage to do the right thing at the right time." You'll evaluate myriad possibilities, moment to moment, and intuitively pick the best course of action.

When you do, pass this book on to someone else, dog-eared and ragged though it may be, so that they too can master the art of agile development.

For the Pragmatists

What if you don't want to master a so-called art? What if you just want to develop good software?

Don't worry—this book is for you, too. Parts I and II are just what you need. We took our years of experience with agile development and Extreme Programming and distilled them into a single, clearly defined, comprehensive approach.

This approach allows us to use plain, straightforward language without caveats or digressions. We get to include a lot of practical tips. We candidly describe when our approach won't work and what alternatives to consider when it doesn't.

There's a downside to discussing just one approach: no single methodology is appropriate for everyone. Our advice may not be appropriate for your team or situation. Be sure to read Chapter 4 before putting our advice into practice.

You may be able to adopt part of XP even if you can't adopt all of it. The "Contraindications" section of each practice in Part II describes when a practice is inappropriate. If this applies to your situation, the "Alternatives" section will help you decide what to do instead.

Don't go too far and automatically assume that a particular practice won't work for you. Some of the ideas in this book are counterintuitive or just don't sound like fun. Most of them work best in concert with the others. If you can, try the practices as written for a few months, gain some real-world experience on how they work in your environment, and *then* change them.

We've been putting these ideas into practice for years. In the right environment, they really work. Agile development has been more fun, and more successful, than any other approach to team software development we've tried. Come join the ride.

Who Should Read This Book

This book is for anyone who is, will be, or wants to be part of an agile team. That includes programmers, of course, but it also includes domain experts, testers, projects managers, architects, designers, and business analysts. Agile teams are cross-functional; this book reflects that fact.

If you're a leader or you're interested in bringing agile development to your team or organization, you should read the whole book from cover to cover. Part I introduces agile concepts and describes how to adopt XP. Part II describes each of XP's practices in detail. Part III goes beyond XP, looking at the principles that allow you to create your own agile method by customizing XP to your particular situation.

If you just want to learn enough to do your job, you can focus primarily on Part II. Start with Chapter 3 in Part I to get an overview, then read through the practices in Part II that apply to your work. Each practice starts with a description of the audience it applies to, such as “Programmers,” “Customers,” or “Testers.”

If you’re merely curious about agile development, start by reading Part I. Again, Chapter 3 provides a good introduction. Afterwards, take a look at the practices in Part II. Start with the ones that look most interesting; you can read them in any order.

About the Études

Have you ever heard a musician playing scales? That’s an étude (if a boring one). An étude teaches mastery through precise and careful repetition. Eventually, the étude is abandoned, but the skills remain.

Extreme Programming is our étude for software development. We hope that practicing Extreme Programming week after week will help you master agile development. Eventually, you’ll change your practices, but the underlying principles will remain.

Besides the overarching étude of Extreme Programming, we’ve included a *mini-étude* for each major theme of agile development. Beginning agile teams can use the études to refine their practice of agile development. As you gain experience, look deeper; use the études to help connect Part II’s detailed practices to Part III’s general principles.

NOTE

These études are also useful for teams not currently practicing XP.

Either way, the études require thought to be effective. Each étude provides information, but it doesn’t tell you how to act on that information. Think about it. What did you learn from the étude? What was frustrating or exciting? How does that information affect your work? What will you do about it? Without attention and reflection—that is, mindfulness—the études are just games.

NOTE

It’s no coincidence that you need mindfulness to master the art of agile development as well. You have to think about more than XP’s practices for it to be effective.

Like musical études, our mini-études work best when you repeat them. We’ve designed them to take half an hour, so you can (and should) practice them every day for a week or more. Unlike musical études, these agile exercises work best when you include your whole team. The more you all understand about the process and where everyone fits in, the better you will work together.

To start, you need a quiet work area capable of holding everyone in your team comfortably. There should be a whiteboard or wall where you can hang or post index cards between meetings. There must also be sufficient space to break into groups of two or three people and to talk softly without disturbing other groups.

We’ve found it valuable to use a timer, whether a stopwatch or a kitchen timer, to keep the session moving. Each étude has a few parts of 5 to 10 minutes. Although that time will seem to flow quickly

on your first day, respect the time limits. You'll perform the étude again tomorrow, so it's OK if you don't finish everything.

Choose one member of your team to facilitate the étude by watching the time (perhaps calling out "one minute remaining" when appropriate) and promoting discussion. Again, the first session may be hard, but a good facilitator can encourage everyone to continue.

At the end of each étude, we recommend spending a few minutes debriefing. What did you learn? Are there questions and ideas you can follow up on during your regular work? If you've been trying this exercise for more than a week, are you still getting valuable results?

If you're new to XP and agile development, we strongly recommend that you perform each étude while you study the related chapter as a team. Besides exploring one particular theme in agile development, each étude can illuminate an aspect of how your team works together on your agile project.

About Pronouns

We speak in first-person singular rather than first-person plural in the rest of this book. (We say "I," not "we.") We include a lot of personal anecdotes and experiences, and the singular form works better as a result. However, this book is unquestionably the result of a partnership between two authors and our use of the word "I" is merely a convenience.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*The Art of Agile Development* by James Shore and Shane Warden. Copyright 2008 O'Reilly Media, Inc., 978-0-596-52767-9."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

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://www.oreilly.com/catalog/9780596527679>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

<http://www.oreilly.com>

Acknowledgments

We owe a debt of gratitude to Elisabeth Hendrickson for her contribution of the “Exploratory Testing” section. Her expertise and writing skill made the material sparkle. For more of Elisabeth’s writing, visit <http://www.testobsessed.com/>.

We stole good ideas wherever we could find them. Kent Beck, Ron Jeffries, and Ward Cunningham each had a hand in the ideas that led to XP, and we stole liberally from those. In addition to XP itself, Kent Beck introduced us to the idea of XP practices as études. Ward Cunningham introduced us to the notion of technical debt, a concept we use heavily. Brian Marick’s series of essays, “Agile Testing Directions,”* influenced our thoughts on agile testing and the role of testers on agile teams.

James had the opportunity to work with Joshua Kerievsky on an *Industrial XP (IXP)*[†] project for half a year. He learned a lot from that project; the Vision practice in particular was inspired by IXP’s Project Chartering practice. David Schwartz and Amy Schwab of True North pgs, Inc.,[‡] provided the specific vision format that we use, as well as the term *project community*. Their *Mastering Projects* workshop is excellent; take it when you have the opportunity.

Thank you to our editor, Mary Treseler O’Brien, for providing vision (where necessary), trust (where appropriate), and deadlines (where frightening). This book would not be what it is without you gently nudging us in directions different from our original ideas.

Thanks also to our army of reviewers, who provided over one thousand comments and suggestions on our mailing list. In particular, thanks to Adrian Howard, Adrian Sutton, Ann Barcomb, Andy Lester,

* <http://www.testing.com/cgi-bin/blog/2004/05/26>

† <http://www.industriabxp.org/>

‡ <http://www.projectcommunity.com/>

Anthony Williams, Bas Vodde, Bill Caputo, Bob Corrick, Brad Appleton, Chris Wheeler, Clarke Ching, Daði Ingólfsson, Diana Larsen, Erik Petersen, George Dinwiddie, Ilja Preuß, Jason Yip, Jeff Olfert, Jeffery Palermo, Jonathan Clarke, Keith Ray, Kevin Rutherford, Kim Gräsman, Lisa Crispin, Mark Waite, Nicholas Evans, Philippe Antras, Randy Coulman, Robert Schmitt, Ron Jeffries, Shane Duan, Tim Haughton, and Tony Byrne for their extensive comments. Special thanks to Brian Marick, Ken Pugh, and Mark Streibeck for their comments on the completed draft.

James Shore

Every work builds on what came before. I was fortunate to have not just one, but many giants to stand on. Without the inspired work of Kent Beck, Alistair Cockburn, Ward Cunningham, Tom DeMarco, Martin Fowler, Ron Jeffries, Timothy Lister, Steve McConnell, and Gerald Weinberg, I wouldn't have anything close to the understanding of software development I have today. It's thanks to their example that this book exists. In particular, thanks to Alistair Cockburn for generously inviting me to his roundtable and introducing me to the agile community.

If giants enabled me to contribute to this book, then Kim Eaves and the Denali team brought the stepladder. Without their enthusiastic support, I never would have been able to try that crazy XP thing. Thanks also to Rob Myers for his ever-friendly consideration of my rants.

I also gratefully acknowledge my coauthor and friend, Shane Warden. This project morphed from a little 100-page second edition into a 400-page monster. You didn't complain once. Thanks for putting up with me. (And hey! Nice book.)

Finally, thank you, Neeru, my loving and patient wife. I used to think authors thanking their families was cliché. Now I understand. I couldn't have finished this book without your support.

Shane Warden

Thanks to Jim for arguing with me while writing the first version of this book (it's better for it) and for convincing me that the second edition was worth doing.

Thanks to Allison and Andrew for the tools we used to write this book.

Thanks to my family for supporting me (and not grumbling too much while I sat upstairs and wrote very slowly), and to my friends for dragging me out of my house once in a while.

Thanks also to the other contributors to Parrot and Perl 6 for being unwitting collaborators, examples, and victims of some of the ideas in this book. The work we do continually amazes me.

O'Reilly Media, Inc. 介绍

O'Reilly Media, Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用,人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而,计算机领域的技术更新速度之快也是众所周知的,为了帮助国内技术人员在第一时间了解国外最新的技术,东南大学出版社和美国 O'Reilly Meida, Inc.达成协议,将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作,以影印版或者简体中文版的形式呈献给读者。其中,影印版书籍力求与国外图书“同步”出版,并且“原汁原味”展现给读者。

我们真诚地希望,所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助,对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的影印版图书,包括:

- 《Ruby 程序设计语言》(影印版)
- 《WPF 编程 第二版》(影印版)
- 《高性能网站》(影印版)
- 《Ajax 权威指南》(影印版)
- 《学习 ASP.NET 2.0 与 Ajax》(影印版)
- 《ASP.NET 3.5 构建 Web 2.0 门户网站》(影印版)
- 《Mac OS X: The Missing Manual, Leopard Editon》(影印版)
- 《深入浅出 JavaScript》(影印版)
- 《敏捷开发艺术》(影印版)
- 《Intermediate Perl》(影印版)
- 《学习 Python 第三版》(影印版)
- 《ASP.NET AJAX 编程》(影印版)

Table of Contents

Preface	xiii
----------------------	-------------

Part I. Getting Started

1. Why Agile?	3
Understanding Success	4
Beyond Deadlines	4
The Importance of Organizational Success	5
Enter Agility	6
2. How to Be Agile	9
Agile Methods	9
Don't Make Your Own Method	10
The Road to Mastery	11
Find a Mentor	12
3. Understanding XP	15
The XP Lifecycle	18
The XP Team	27
XP Concepts	39
4. Adopting XP	43
Is XP Right for Us?	43
Go!	51
Assess Your Agility	62

Part II. Practicing XP

5. Thinking	69
Pair Programming	71
Energized Work	79
Informative Workspace	83

Root-Cause Analysis	88
Retrospectives	91
6. Collaborating	99
Trust	102
Sit Together	112
Real Customer Involvement	120
Ubiquitous Language	124
Stand-Up Meetings	129
Coding Standards	133
Iteration Demo	138
Reporting	144
7. Releasing	153
"Done Done"	156
No Bugs	160
Version Control	169
Ten-Minute Build	177
Continuous Integration	183
Collective Code Ownership	191
Documentation	195
8. Planning	199
Vision	201
Release Planning	206
The Planning Game	219
Risk Management	224
Iteration Planning	233
Slack	246
Stories	253
Estimating	260
9. Developing	271
Incremental Requirements	273
Customer Tests	278
Test-Driven Development	285
Refactoring	303
Simple Design	314
Incremental Design and Architecture	321
Spike Solutions	331
Performance Optimization	335
Exploratory Testing	341

Part III. Mastering Agility

10. Values and Principles	353
Commonalities	353
About Values, Principles, and Practices	354
Further Reading	354
11. Improve the Process	357
Understand Your Project	357
Tune and Adapt	358
Break the Rules	359
12. Rely on People	361
Build Effective Relationships	361
Let the Right People Do the Right Things	363
Build the Process for the People	364
13. Eliminate Waste	367
Work in Small, Reversible Steps	367
Fail Fast	369
Maximize Work Not Done	370
Pursue Throughput	371
14. Deliver Value	375
Exploit Your Agility	375
Only Releasable Code Has Value	376
Deliver Business Results	378
Deliver Frequently	379
15. Seek Technical Excellence	381
Software Doesn't Exist	381
Design Is for Understanding	382
Design Trade-offs	383
Quality with a Name	383
Great Design	383
Universal Design Principles	384
Principles in Practice	387
Pursue Mastery	388
References	391
Index	397

PART I

Getting Started