

The
Pragmatic
Programmers

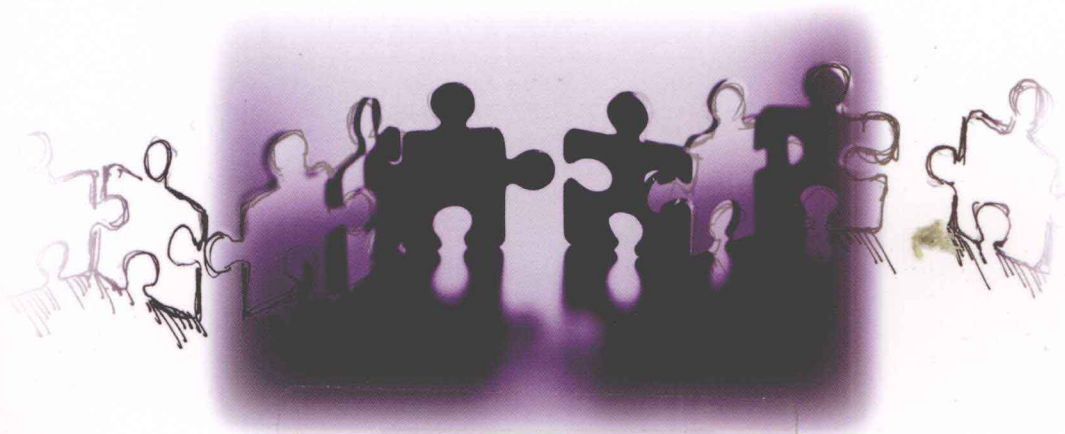
TURING

图灵程序设计丛书

Ship It!

A Practical Guide to Successful Software Projects

软件项目成功之道



[美] Jared R. Richardson 著
William A. Gwaltney Jr.
苏金国 王少轩 等译



人民邮电出版社
POSTS & TELECOM PRESS

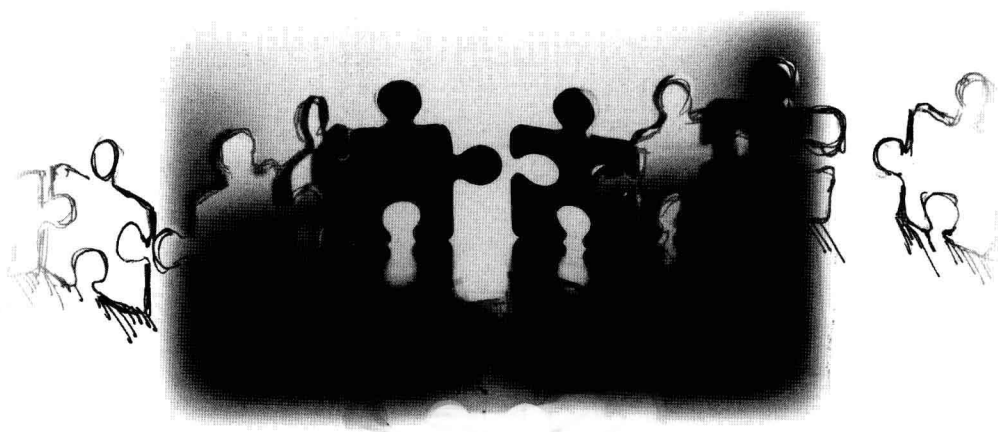
TURING

图灵程序设计丛书

Ship It!

A Practical Guide to Successful Software Projects

软件项目成功之道



[美] Jared R. Richardson 著
William A. Gwaltney Jr.
苏金国 王少轩 等译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

软件项目成功之道 / (美) 理查森
(Richardson, J. R.), (美) 格沃特尼 (Gwaltney, W. A.)
著; 苏金国等译. -- 北京: 人民邮电出版社, 2011.8
(图灵程序设计丛书)

书名原文: Ship It! A Practical Guide to
Successful Software Projects
ISBN 978-7-115-25965-3

I. ①软… II. ①理… ②格… ③苏… III. ①软件开
发—项目管理 IV. ①TP311.52

中国版本图书馆CIP数据核字(2011)第141577号

内 容 提 要

作者以精炼、风趣的语言揭开了项目管理过程的神秘面纱。所涵盖的内容涉及工具、使用项目技术、曳光弹开发以及常见问题的解决办法, 并提供大量实用建议, 且总结出方方面面的“技巧”, 帮助读者在阅读过程中快速消化所看内容。

本书适合软件研发专业人士阅读, 对软件项目管理感兴趣的社会各界人士也能从中获益。

图灵程序设计丛书 软件项目成功之道

- ◆ 著 [美] Jared R. Richardson William A. Gwaltney Jr.
译 苏金国 王少轩 等
责任编辑 傅志红 卢秀丽
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
- ◆ 开本: 700×1000 1/16
印张: 12.25
字数: 213千字
印数: 1-4 000册

2011年8月第1版

2011年8月北京第1次印刷

著作权合同登记号 图字: 01-2011-1378号

ISBN 978-7-115-25965-3

定价: 39.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154



读者对本书的赞誉

真是让人难以置信的一本书。刚开始读到这本书，我惊讶得几乎要从座位上掉下来，因为那时我参与的一个项目正在遭遇同样的问题，简直与你描述的一模一样，这本书会让我获益多多。

► Matthew Bass, 软件工程师

就像 Mac 一样，这本书“相当实用”，因为它从团队领导人和成员的经验教训中汲取了精华，揭开了软件项目管理过程的神秘面纱……这本书语言非常简洁，很快就能读完，如果你是软件开发行业中的一员，花几个小时读读这本书绝对值得。

► Robert Pritchett, macCompanion 公司

在软件行业，如果我们都不再像晕头转向的小丑那样整天忙得团团转，而是能够充分发挥自己的潜能，那就太好了。这本书可以提供帮助，只要那些真正需要帮助的人能够关注本书。

► Mike Gunderloy, Larkware 公司

如果你交给别人一本书，而它居然能改变他们的思维和行为方式，这种感觉确实很奇妙。我真的很高兴读到这样一本书，相信以后还会不时拿来翻看，这本书就是《软件项目成功之道》。

► Jeffery Fredrick, CruiseControl 平台

很少有一本书能让开发人员和经理们都信服，不过这本书确实做到了……这本书中我最喜欢的是最后一部分，汇集了一些篇幅不长文章，不仅描述了软件项目经常遇到的问题，还指出了如何应用书中给出的原则和实践解决这些问题。与我读过的另外一些牵强附会的“反模式”总汇不同，这部分非常实用，

确实很有用。

▶ Ernest Friedman-Hill, Java Ranch

真是一本好书！作者用通俗自然的笔调将内容娓娓道来，没有落入刻板介绍方法的俗套，实在太棒了。

▶ Roberto Gianassi, IT 咨询师

《程序员修炼之道》为我们介绍了软件开发人员个人的技术和能力，这本书则是面向软件开发团队。如果你认识的同在软件领域打拼的某个人工作上不顺心，可以让他读一读这本书。

▶ Guerry A. Semones, 资深软件工程师, Appistry 公司

一本关于软件的书居然如此有趣，实属难得。这些想法睿智、中肯而且非常重要。此时此刻，我读到的这些内容会让我立刻成为一个更棒的程序员。

▶ Joe Fair

这本书保持了 Pragmatic 图书一贯的风格，读起来轻松流畅，简明扼要。我只用两天就读完了这本书，收获颇丰，而且从中掌握的精髓要义可以直接加以应用。如果你想让自己的软件开发生涯更上一个台阶，强烈推荐这本书！

▶ Anil John

如果你是一个开发团队领导人或者开发经理，不读这本书会被“炒鱿鱼”的。

▶ David Starr, Elegant Code 公司

译者序

说来也巧，拿到原书正在翻看时，我所在的一个项目组临时召集开会。这个项目去年就已经启动，甚至单位最高领导亲自动员，其重要性可见一斑。开始时大家确实士气高昂，成效明显，每周例会都会汇报新的进展，可惜第一个演示版本提交后，这种势头就没能再保持下去。大家都忙于处理自己手上其他的繁杂事务，在这个项目上投入的精力越来越少，每周例会也不再召开，已经成为回忆。由于中期检查临近，这才不得不召开“例会”了解目前工作状态，并讨论下一步工作安排。

会议开始后，一个同事的发言让我觉得似曾相识。他负责的部分需要与外单位开发的4个模块集成，提交演示版本时，只有一个模块能成功集成，另外3个模块一直有问题。这期间改改停停，问题一直不能干净利落地得到解决。最近联系过一次联调，经过几天努力，终于搞定了这3个模块。可是最后一天外单位人员临走时过于匆忙，不小心覆盖了那几天的成果。所以这位同事说，只好再安排一个时间重新联调，不过时间还没有定下来……我心里想：这不正是我刚才在书里偶然看到的情况吗？

这时另一位同事也表示出对这个单位合作态度的不满，开始大谈他们的种种不是，大家只能百无聊赖地听着，不好意思打断。接下来话题扯得越来越远，一下午的会议结束后，尽管大家都很是辛苦，甚至头昏脑胀，但还是不清楚目前到底有哪些新的进展，也没有讨论好接下来的具体安排。

回到自己的办公室，重新拿起这本书，我发现这一切正是书中所说的一个真实场景的再现，当即把它推荐给了我们的项目负责人。现如今，我们的每周例会已经恢复（尽管没有做到书中建议的每日例会），项目在有条不紊地向前推进。

关于这本书的内容，我不想多讲，只想说：这本小书会带给你实实在在的

好处，如果你手上有迟迟无法交付的项目，也许这就是你的曙光！

“好风凭借力，送我上青云”！借着大师们的力量，站在巨人的肩上，我们会走得更远！

全书主要由苏金国、王少轩翻译，并得到王小振、李璜、刘亮、李新宏的大力协助。若译文有不当之处，敬请读者批评指正。

教条并不意味着没有思想，而是思想的终结。

► 吉尔伯特·基思·切斯特顿 (1874—1936)

序 言

你可能已经注意到了，书架上关于软件开发的书并不只有这一本。

实际上，关于设计和构建软件的书林林总总，多得让人眼花缭乱，不仅如此，它们相互之间还不太一致。遗憾的是，这些方法上的分歧带来的感受往往让人“炙热难当”：作为软件行业的从业人员，我们体会到的不是光明，而是焦头烂额的感觉。另外，我们的项目总是延期。

我们在不断探寻更好的软件开发方法，希望能找到适合自己 and 团队的好办法。不过，基于既有的教条，关于各种开发方法孰优孰劣的讨论最终总会演变成激烈的争吵。字典中教条的定义是“一种权威性观点，但并没有充分的依据”。我们经常会看到，各种方法的拥护者们都坚持认为自己的方法才是开发软件唯一正确的方法。我们不断听到一些从业人员这么讲，他们执着地采用某种方式开发软件，即使这种方法明显危害到团队的其他人甚至整个组织，却仍然固执己见。

事实上，开发软件根本没有所谓“绝对正确的方法”。倒是有很多错误的方法，不过没有哪一种方法、观点、哲学或工具能“以不变应万变”，在所有时间、所有场合对所有项目和所有人都适用。软件是人创建的，不会有两个人完全一样。

所以对于软件开发实践，我们还是沿用“务实”的观点。我们要强调的是目标：你希望得到一堆签名，还是希望所有人都能理解？你希望匆忙间随便抛出一堆东西来，还是生成一个可以真正帮助别人完成工作的软件？

我们想尝试新鲜事物，并对我们的实践不断评价和修正；我们希望找到适用的方法。不过着手这个工作很费功夫：我们要做大量研究，需要耗费大量时

间，而大多数参与实际工作的程序员根本没有这么多时间。

这让我萌生出一个想法：请 Jared 和 Will 写这样一本书。这是一本快速入门的指南，你能从中了解开发可靠代码所需的基本、有效的工具和技术。

Jared 和 Will 是《程序员修炼之道》最早的读者，他们用心领会书中的道理，运用我们的方法和技术，并结合其他流行的敏捷方法实践，形成了一种独特的方法，让他们不论在小的初创公司还是世界上最大的私营软件公司，都能得心应手地开展工作的。

这本书汇集了他们最钟爱的技术和实践。运用这些内容，你将在改善开发过程时如虎添翼，快速入手，当然可能还需要补充 Pragmatic Starter Kit 系列介绍的其他一些技术细节。随着时间的推移，你可能希望进一步扩展，尝试另外一些实践和技术。毕竟，这正是务实之道——根据当务之急，找到最适用的方法。

希望这本书对你有帮助，能让你轻轻松松地成功交付系统！

Andy Hunt

The Pragmatic Programmers 公司

2005 年 4 月

andy@pragmaticprogrammer.com

前 言

不论是对你自己还是对你的职业发展，最明智的一项投资就是让你身边有一些“合适”的人——他们会成为你能找到的最好的资源。这些人可能已经做过你打算做或者想学着做的事情。如果你想做一些事情，最好找到那些做过这些工作的人，或者至少找到一些真正睿智的人，能够告诉你该怎样完成这些工作。尽可能和他们多待些时间，通过相互帮助来向他们学习。与这些高水平的人相处，会让你学到很多东西，不论你的工作是什么，这都会让你表现得更出色。

这是一个很好的想法，但是要想与那些顶尖人物接触可能很困难。像 Martin Fowler、Kent Beck 和 Pragmatic Programmers 的作者等大师级人物，并不是我们大多数人有机会见到的，不过我们可以看到他们的书、文章和演示文稿。所以要开始读书。一个月读一本书应该不会太费劲。但是不要就此止步，接下来可以学习一种新的编程语言，或者研究一个不同的开发过程。在学习和读书时，要想办法把这些新思想应用到当前的工作中。这样一来，你不仅能帮助你的公司提升，更重要的是，还可以让你自己得到提高。

要让自己开放地接受新思想。不要闭塞，应当想办法把这些新思想应用到你现在的工作中。也许你会放弃，并声称某种新思想不适用，这样做当然更轻松，不过我们的目标是学习采用另外一种不同的方式考虑问题。要打破条条框框（或者至少建一个更大的框框）。多学习掌握一些看似不太相关的概念和思想。

通过对环境和过程进行分析和评判，你可以找出弱点。也许这会帮助你对这个项目或者下一个项目做出改进。但同时你也练习了一种新的思维方式，不论在哪里工作这对你都会有好处。大多数人从来不曾了解这个概念，擅长的人更是少之又少。

所以，读完这本书的每个实践后，请停下来，花 5 分钟时间试着想想有没

有办法在你今天做的工作中具体运用各个概念。你可能不假思索地回答：无法做到。这种回答当然最不费劲，不过要记住，不能这么懈怠，你应当更加努力！如果你自己找不出一种方法来应用这个概念，可以找一位同事来帮忙。如果仅凭自己的双眼看不到，可以借助别人的视角来了解。不论在什么领域，知道如何利用同事的经验，绝对是高手特有的标志。

希望你掌握从本书（和整个 Pragmatic Starter Kit 系列）读到的内容，想办法在工作中应用每一个概念。你会看到读这本书给你带来的直接好处，其中最大的好处就是你会学习如何真正加以应用。

希望你喜欢！

致谢

首先要感谢 Andy 和 Dave 让我们为 Pragmatic Bookshelf 写一本书。能为你们写一本书真是荣幸。Andy，你做的已经远远超出了你的份内职责，甚至花好几个小时与我们一同修改手稿，给我们上了一堂写作速成课，尽管有时对我们的写作天份……有些发愁（我们确信你肯定失望过）。Dave，我们曾经问过很多关于图书生成系统和标记语言细节的问题，感谢你花那么多时间回答这些邮件。非常感谢你们二位！

我们有很多非常棒的审校人员，另外还有很多人做出了贡献，他们详细而且有建设性的反馈确实意义重大。没有你们投入的时间，没有你们丰富的经验，这本书不可能出版。Susan Henshaw 和 Jim Weiss 花了大量时间审校我们粗糙的文字，而且读过不止一遍。谢谢你们。

还要感谢 Mike Clark、David Bock、Ken Pugh、Dominique Plante、Justin McCarthy、Al Chou、Bryan Ewbanks、Graham Brooks、Grant Bremer、Guerry Semones、Joe Fair、Mark Donoghue、Roberto Gianassi、Rob Sartin、Shae Erisson、Stefan Schmiedl 和 Andy Lester。你们当中很多人曾忍受过这本书很早的版本，最近我们重新读了你们原先读过的版本，对你们曾经遭受的“折磨”我们深表歉意。说真的，所有反馈都很棒，正是有了这些反馈，这本书才得到了如此显著的改进。

在我们的职业生涯中，曾经与很多人共事过，其中有些人对我们的工作以及这本书产生了直接的影响。我们要特别感谢 Jim Weiss、Randy Humes、Graham Wright、Flint O'Brien、Toby Segaran 和 John Wilbanks。还要感谢我们现在的经

理 Oita Coleman 对我们的鼓励和支持。我们很幸运，能够在 SAS 这样的世界级公司工作。

如果没有敏捷开发社区的智慧，没有大家出色的作品，这本书绝无可能问世。我们读过 XP、Scrum、Crystal 以及很多其他软件领域专家的书和文章。没有你们辛苦而忘我的工作，软件行业可能还在黑暗岁月中挣扎。也许我们还没有完全走出黑暗，不过起码正朝着正确的方向前进。你们孜孜不倦的工作让大家受益匪浅。

开源社区共同为全世界提供了这么多非凡的工具和想法，同样要向你们表示感谢。正是因为全世界开发人员的无私奉献，我们这里讨论的大多数工具才可以免费使用。在此要特别提到 SourceForge 团队和 Apache 软件基金会。你们提供的服务和工具不仅让我们提高了生产效率，还改变了整个世界。

最后，也是最重要的，要感谢我们的主，耶稣基督，愿主永得荣耀！

► Jared Richardson 和 William Gwaltney

我的妻子 Debra 为这本书投入了大量的时间和精力。甚至有好几个星期 Debra 花在这本书上的时间比 Will 和我还要多。其余的时间她既做母亲又做父亲，才让我得以安心地完成这本书。我发自内心地相信，如果没有她的帮助和支持，我绝对不可能完成这个工作。谢谢你！

我的孩子们，Hannah 和 Elisabeth，很多个夜晚和周末爸爸都把自己锁在办公室里一心扑在这本书上，感谢你们忍受了这一切。谢谢你们的理解 and 爱！

► Jared

非常非常感谢我的家人，很多个漫长的夜晚和周末我都在写书，谢谢你们能理解，还要忍受我在不顺的时候发脾气。是你们让一切付出都变得有意义。

► William

目 录

第 1 章	绪论	1
1.1	习惯性优秀	2
1.2	务实观点	3
1.3	路线图	5
1.4	继续前进	7
1.5	怎样读这本书	7
第 2 章	工具和基础设施	11
1	在沙箱中开发	15
2	管理资产	18
3	建立构建脚本	23
4	自动构建	27
5	跟踪问题	32
6	跟踪特性	36
7	使用自动化测试框架	39
8	选择工具	46
9	何时结束试验	48
第 3 章	实用项目技术	51
10	按照任务清单工作	53
11	技术领导人	65
12	每天都要协调和沟通	73
13	审查所有代码	82
14	发送代码变更通知	91
15	大汇总	96
第 4 章	曳光弹开发	99
第 5 章	常见问题及解决办法	119
16	救命! 我继承了遗留代码	120
17	测试不可测试的代码	122
18	特性不断破坏	123

19	测试？我们早就不用了.....	124
20	不过我这里没问题！.....	126
21	集成代码很痛苦.....	127
22	不能可靠地构建产品.....	129
23	客户不满意.....	130
24	有一个另类的开发人员.....	131
25	你的经理不满意.....	134
26	团队不能很好地合作.....	136
27	在根本问题上无法得到“认可”.....	137
28	新实践没有帮助.....	140
29	没有自动测试.....	143
30	我们只是低级别开发人员，没有人指导我们.....	144
31	我们在一个“死亡之旅”项目中.....	145
32	特性不断蔓延.....	147
33	我们永远也完不了.....	148
附录 A	技巧汇总.....	151
附录 B	源代码管理.....	153
附录 C	脚本构建工具.....	157
附录 D	持续集成系统.....	161
附录 E	问题跟踪软件.....	165
附录 F	开发方法.....	169
附录 G	测试框架.....	173
附录 H	建议阅读书目.....	177

如果我们坚持不懈，那么，优秀就不再是一种行为，而成为一个习惯。

►亚里士多德

第 1 章

绪 论

如今，很多软件开发人员都很困惑。他们夜以继日、废寝忘食地工作，可是他们的团队还是无法顺利地完成任务。不是他们的努力不够，也不是愿望不强烈：团队里的每一个人都希望能干净利落地把项目搞定，不过大家都不知道如何共同努力完成工作。你很难有时间坐下来静心读点东西，做些试验，得出哪些做法可行以及如何在你的工作室合理运用。大多数人都太忙于手头的工作，而无暇顾及这种研究。

本书就这样应运而生。这本书汇集了大量基本而实用的建议，这些建议已经在这个领域的多个项目以及大小公司中得到充分证明。我们的亲眼所见和亲身经历可以证明这些方法确实可行。我们不同于那些只在公司出出进进几个星期就离开的顾问，而是天天都在这些公司全力工作。我们并不只是提出一些听起来不错的想法，就匆匆转向下一个预约项目。如果这些想法没有效果，我们还会留在公司查看为什么会失败，哪里出了问题。另一方面，我们必须等到情况好转，工作能够顺利进行下去。

我们介绍的想法有些是从一些众所周知的软件方法中得来的，我们会尽量指明它们的出处。另外一些想法则是我们用热血、汗水和泪水“凝结”而成的。我们尝试过很多工具、技术和最佳实践，如果可行就会将其保留；如果失败，就会断然地放弃。在这里你几乎看不到我们通过盲目摸索得来的一手经验（尽管这是好东西）。相反，我们会“站在巨人的肩上”，精选出这个行业最睿智的思想，把它们转化为你将看到的文字。

如今，50%~70%的软件开发团队并没有使用那些基本的、众所周知的软件

实践 ([Cus03])。很多情况下,这并不是因为他们不知道要做什么,而是因为他们不清楚该如何立即开始运用这些实践。我们会告诉你如何向管理层推销这些想法,给出能够让你迅速上手的明确的实用步骤,然后指出要注意哪些警告信号以免脱离正轨。

本书由“一线”开发人员倾力编写。这本书凝聚了我们在不同公司的实践中得出的经验(从初创的小型公司,到全世界最大的私营软件公司),而不是刻板的理论。这是一个不依赖具体方法论的指南,更注重如何让项目顺利完成。

我们努力使这本书沿袭 Pragmatic Bookshelf 图书一贯秉承的风格:实用、简洁,轻松阅读。希望能将这一风格发扬光大。

1.1 习惯性优秀

为什么这里要引用亚里士多德的名言?“如果我们坚持不懈,那么,优秀就不再是一种行为,而成为一个习惯。”能够生产出一个或一些不错的产品算不上优秀。优秀体现在我们每天所做的点点滴滴,也就是我们的习惯。一流的产品只不过是好习惯的副产品。

把这句名言应用到我们自己身上(包括工作和个人生活中),要求我们能够认识到生活实际上只是习惯的副产品,因此最好审慎地选择我们的习惯。大多数人会毫无计划地草率陷入当前的工作模式而不能自拔,这有很多原因:也许你原先就是这么学的,或者这是你的老板一贯的做法,诸如此类。但是,我们确实能做得更好。

要有意识地搜寻好习惯,并把它们加到你的日常生活当中。

可以做个小实验。找一种要研究的开发方法,挑选一个看上去对你来说不错的习惯(而且这个习惯可以单独运用)。实际使用一个星期。如果你喜欢,而且看来有好处,就继续用上一个月。不断实践这个新习惯,把它变成为你生活模式中很自然的一部分,然后再选择另一个新习惯重新开始这个过程。如同你一砖一瓦地建造地基那样,一次一个新习惯重复这个过程,你就正在为优秀奠定坚实的基础。有时有些做法并不适用于你的环境,请大胆舍弃这些方法,也不要因为某个实践很有名或者很流行就勉强地把它保留下来。要走你自己的路,根据自身情况做出正确的选择。

“我们的每一天怎样度过，一生就会怎样度过”。^①既然如此，就必须仔细考虑如何度过我们的每一天。

技巧 1

选择习惯

不要偶然地养成某些习惯，要有意识地主动选择习惯。

1.2 务实观点

这本书并不是一个学术著作，单纯地分析为什么有些做法可行或者有些做法不可行，也不会简单地罗列一堆实践和方法供你选择。

相反，这本书会提供现实的软件项目中比较有用的方法。我们会先引入一个新工具或实践，具体加以运用，弄明白它是否可行。接下来，会把那些可行的工具保留在我们的软件开发工具箱中，并“随身携带”。最终看来，我们所做的确实很奏效！希望这些工具和实践也适用于你。

我们参与过很多小型初创公司的工作，这些小公司条件有限，并不是某种方法“正确”就能够加以使用——他们无法做到如此“奢侈”。我们所处的环境要求我们找出能够立即投入使用的解决问题之道。我们也在较大规模的公司工作过，这些公司拥有丰富的资源和技术。但我们发现，大公司也不会因为某个工具很优秀或者某个专家很推崇就使用这个工具。他们想要的是能够快速而经济地解决当务之急的方案。所以我们会取舍地挑选合适的习惯，最终建立一个足够通用可以移植，并且仍能有效地解决具体问题的工具箱。这本书汇集了我们使用过的很多好习惯，它们也会对你的工作室产生深远影响——结果可能是惊人的。

为了便于说明，下面来讲一个有关两个软件工作室的“双室记”故事（因为篡改了《双城记》的名字，在此向查尔斯·狄更斯致歉）。

第一个工作室简直一团糟。他们购买了非常昂贵的源代码管理软件，但是从来没有安装过。结果，他们甚至把原本打算向潜在客户展示的演示版的源代码都弄丢了。没有人能确定产品中应当包含哪些特性，但不管怎样，整个开发团队还在兢兢业业地工作着。开发出的代码很不稳定，每 5 分钟左右就会崩溃

^① Annie Dillard，美国作家，诗人，1975 年因其散文《溪畔天问》获得普利策奖。