

Practices of an Agile Developer

高效程序员的45个习惯

敏捷开发修炼之道

[美] Venkat Subramaniam 著
Andy Hunt
钱安川 郑柯 译



TURING 图灵程序设计丛书

Practices of an Agile Developer

高效程序员的45个习惯

敏捷开发修炼之道

[美] Venkat Subramaniam 著
Andy Hunt
钱安川 郑柯 译



人民邮电出版社
北京

人民邮电出版社

样书

专用章

图书在版编目 (CIP) 数据

高效程序员的 45 个习惯：敏捷开发修炼之道 / (美)
苏帕拉马尼亚姆 (Subramaniam, V.), (美) 亨特
(Hunt, A.) 著; 钱安川, 郑柯译. —北京: 人民邮电
出版社, 2010.1

(图灵程序设计丛书)

书名原文: Practices of an Agile Developer:
Working in the Real World
ISBN 978-7-115-21553-6

I. ①高… II. ①苏… III. ①程序设计 IV.
① TP311.1

中国版本图书馆CIP数据核字 (2009) 第178801号

内 容 提 要

本书总结并生动地阐述了成为高效的开发人员所需具备的 45 个习惯、思想观念和方法, 涵盖了软件开发进程、编程和调试工作、开发者态度、项目和团队管理以及持续学习等几个方面。

本书适合所有程序员阅读。

图灵程序设计丛书

高效程序员的45个习惯：敏捷开发修炼之道

- ◆ 著 [美] Venkat Subramaniam Andy Hunt
译 钱安川 郑柯
责任编辑 傅志红
执行编辑 陈兴璐
- ◆ 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 电子函件 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: 800×1000 1/16
印张: 12.5
字数: 212千字 2010年1月第1版
印数: 1-3 000册 2010年1月北京第1次印刷

著作权合同登记号 图字: 01-2008-3859号

ISBN 978-7-115-21553-6

定价: 35.00元

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

反盗版热线: (010)67171154

版权声明

Copyright © 2006 Venkat Subramaniam and Andy Hunt. Original English language edition, entitled *Practices of an Agile Developer: Working in the Real World*.

Simplified Chinese-language edition copyright © 2010 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由The Pragmatic Programmers, LLC授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

கற்க கசடறக் கற்பவை கற்றபின்
நிற்க அதற்குத் தக.

திருக்குறள்-391

学有所成，学有所用。

《提鲁库拉尔》中1330条警句第391条

提鲁瓦鲁瓦 (Thiruvalluvar)，印度诗人和哲学家，公元前31年

Almost every wise saying
has an opposite one,
no less wise,
to balance it.

—George Santayana

几乎每句所谓至理名言都有句意思相反的话与之对应，
而且后者也同样在理。

——乔治·桑塔耶纳（美国哲学家、诗人）

对本书的赞誉

书中“切身感受”的内容非常有价值——通过它我们可以做到学有所思，思有所悟，悟有所行。

——Nathaniel T. Schutta, 《Ajax基础教程》作者

在我眼中，这就是一本名不虚传的Pragmatic书架系列里的图书：简短、易读、精炼、深入、深刻且实用。它对于那些想要采用敏捷方法的人很有价值。

——Forrest Chang, 软件主管

我从一开始看这本书时就一直在想：“哇噻！很多开发者都需要这本书。”我很快就认识到这正是我需要的书，故我向各个级别的开发者强烈推荐。

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

此书通过常理和经验，阐述了为什么你应该在项目中使用敏捷方法。最难得的是，这些行之有效的实战经验，竟然从一本书中得到了。

——Matthew Johnson, 软件工程师

我买过Pragmatic书架系列的其他书籍，从中看到过这本书提到的一些习惯。但是，本书把这些思想整合到一起，而且用了简明、易懂的方式组织起来。我在此向开发新手和想要变得“敏捷”的团队强烈推荐此书。

——Scott Splavec, 资深软件工程师

2 对本书的赞誉

伴随着敏捷实践席卷了整个行业，有越来越多的人需要理解什么是真正的“敏捷”。这本简明、实用的书，正符合他们的需求。

——Marty Haught, Razorstream公司软件工程师和架构师

也许，你已经听说过了敏捷方法学，并在思索着如何才能每天改进自己的工作。我的答案是好好读这本书，倾听这天籁之音，融会贯通这些最佳习惯吧。

——David Lázaro Saz, 软件开发者

这是一本深入浅出地讲解敏捷核心实践的书。我最欣赏本书的地方在于，它不是在推销一个具体的敏捷方法学，而是把各种敏捷方法中的有效实践有机地串联成一个整体。它适合那些渴望快速而可靠地交付高质量软件的人们。

——Matthew Bass, 软件咨询师

这本书是《程序员修炼之道》的完美续篇！

——Bil Kleb, NASA研究员

推荐序 一

仅仅还在几年前，XP还被认为是方法异教，FDD属于黑客程序方法。如今，敏捷俨然已经成为主流学说，敏捷方法成为人们学习和讨论的热点。敏捷方法的应用也更加广泛，以至于不少外包项目都要求采用某种敏捷方法。它不仅仅是小团队和小项目在使用，甚至连微软都开始使用Scrum。

敏捷已经成为一种炙手可热的新时尚。

因为火热，各种不同的说法就多起来；因为时尚，原本有些不认同敏捷的人也开始追捧起来。人们反复地讨论敏捷方法，涉及从哲学思想到实现细节的各个层面。人们不断地推出各种不同版本的敏捷方法，甚至有些方法显得如此矛盾、如此不同。

同时，一些误解也一直在坊间流行。一般误认为敏捷就是快，越快就是越敏捷——字典上的名词解释是其依据。岂不知它本来要以“lightweight processes”（轻量级过程）命名，只不过有些参会者不喜欢被看做是在拳台上跳来跳去的轻量级拳手，所以才用了“敏捷”这个词。还有其他一些误解是，敏捷就是只写代码不写文档；敏捷需要重构而无需设计；敏捷迭代就是尽量做到最小，以至于一个小时就好几次；敏捷需要天才的程序员才能应用，其他人都会水土不服；如此这般。

可以看到，市面上以敏捷为题目的图书俯拾皆是，似乎软件开发的书不加上敏捷这个词就是落伍一样。敏捷体系下存在多种方法，介绍每种方法的图书就有一大堆。再加上每种方法采用不同的技术，每本书采用不同的组织形式，存在这么多书也不奇怪，就更不用提那些仅仅为了跟风而敏捷的作品了。

2 推荐序一

面对如此百花齐放、百家争鸣的现象，你该从什么地方开始呢？有没有一本图书可以作为入门的第一读物呢？

这本书就可以胜任这样的角色！

这是一本很容易理解并掌握，不需要太多基础就可以阅读的书。不管你是开发人员，还是管理人员、财务等后勤人员、学生、编程爱好者，只要你对敏捷有兴趣，就可以读懂这本书。你不会被众多的概念和曲折的逻辑所迷惑，不会被高难度技巧所困扰。这本书为你打开了了解和学习敏捷方法的一扇大门，并指出继续前进的道路。

你会很悠闲自在地读完这本小书，然后说：“原来敏捷就是这么一回事啊！”

自由软件顾问 刘新生 (ozzzzz)

推荐序二

我很喜欢本书的中文书名——高效程序员的45个习惯，比直译成“敏捷开发者实践”含蓄多了。敏捷不是目的，只是手段。只要某个手段适合某个场景，有助于提升质量，提高交付能力，提高开发者水平……总而言之，有好处的事情，我们尽管做就是了，何必冠以敏捷之名？

记得第一次读本书还是两年前。这时又细细读来，越来越觉得“习惯”一词比“实践”更有味道。所谓“流水不腐，户枢不蠹”，厨房脏了就擦一下，总比满墙都是油烟以后再去清理的代价小得多。有价值的东西——比如回顾、测试、重构，一切有利于团队建设、提高生产力的实践都应该频繁且持续做，然后日积月累就养成了习惯。

有些习惯很容易养成，有些则很难。我们大都常常许愿，做计划，比如要做一个至少100人同时在线的成熟应用，参加义工活动，每周至少一篇博客……然后在计划落空的时候，用各种理由来安慰自己。

李笑来老师在《把时间当作朋友》一书中提到：“所有学习上的成功，都只靠两件事：策略和坚持，而坚持本身就应该是最重要的策略之一。”那么，为什么我们会在某些事情上坚持不下去？或者换个角度来看，哪些事情是容易坚持下去的？

以前我是标准的宅男，CS、网络小说、魔兽世界几乎是休闲的全部，等到后来得了腰肌劳损，又得了颈椎病，这才痛定思痛，开始游泳锻炼身体。每天游两千米，一个月以后，游泳就成了习惯。再举个例子，我老婆生完孩子以后体型变化

很大，立志想要减肥。为了坚持下去，她把怀孕前的照片放在电脑桌面上，时时督促自己。后来，减肥也就变成了一种生活方式。

从我的个人体验来看，难以坚持下去的事情，基本都是因为没有迫切的欲望和激情。单说锻炼身体，无论是为了减肥、祛病，还是塑形美体等，做这些事情至少都有明确的目的，这样才能驱使着人们一直坚持下去。没有动机，没有欲望，哪里来的毅力呢？

那么，当我们决定做一件事情的时候，首先就要多问问自己：为什么要做这件事情？它所带来的好处是什么？如果不做它又会有哪些坏处？有了清晰的目的和思路后再去做事，遇到变化时就知道孰轻孰重，该怎么调整计划，同时也不至于被重复和乏味消磨了一时的意气。翻开本书之后，你同样也该对自己提问：“为什么要有自动验收测试，有了足够的单元测试是不是就能保证质量了？”“写自动验收测试有哪些成本，会带来哪些收益？”只有明白了“为什么做”，才能够解决“如何做”的问题。

本书的两名译者与我都是故交。钱安川是我的同事，是ThoughtWorks资深咨询师，有丰富的敏捷实施经验。郑柯与我同是InfoQ中文站敏捷社区的编辑，一起翻译过数十篇稿件。他翻译的《项目管理修炼之道》也即将由图灵公司出版。这次二人联手的作品，定会给读者以赏心悦目的阅读体验。我有幸已经从样章中感受到了这一点。

希望你能够带着问题，踏上愉快的阅读之旅。

希望你能够养成好习惯。

李 剑

ThoughtWorks咨询师

2009.10.10

译者序

“武者，包括内功、外功、武术技击术之总和。有形的动作，如支撑格拒，招式回环，变化万千，外部可见，授受较易，晨操夕练，不难熟练。而无形的内功指内部之灵惠素质，即识、胆、气、劲、神是也，此乃与学练者整个内在世界的学识水平密切相关，是先天之慧根悟性与后天智能的总成，必需寻得秘籍方可炼成。”

——摘自《武林秘籍大全》

公元21世纪，软件业江湖动荡，人才辈出，各大门派林立，白道黑帮，都欲靠各自门派的武功称霸武林。

在那些外家功门派（传统的瀑布开发方法、CMM、ISO和RUP等）和非正统教（中国式太极敏捷UDD等）当道之际，一股新势力正在崛起——以敏捷方法为总称的一批内家功门派。

下面的歌诀是对内家武功招数的概述：

迭代开发，价值优先
分解任务，真实进度

站立会议，交流畅通
用户参与，调整方向

结对编程，代码质量

测试驱动，安全可靠

持续集成，尽早反馈

自动部署，一键安装

定期回顾，持续改进

不断学习，提高能力

上面的每种招式，都可寻得一本手册，介绍其动作要领和攻防章法。几乎每个内家功门派都有自己的拳法和套路。

但，正所谓“练拳不练功，到老一场空”。学习招数和套路不难，难的是如何练就一身真功夫。内家功，以练内为主，内外结合，以动作引领内气，以内气催领动作，通过后天的修炼来弥补先天的不足。

本书是一本内功手册。它注重于培养软件开发者的态度、原则、操守、价值观，即识、胆、气、劲、神是也。

敏捷的实践者Venkat Subramaniam和Andy Hunt携手著下此书。望有志之士有缘得到此书，依法修习，得其精要；由心知到身知，入筋、入骨、入髓，修炼得道。而后，匡扶正义，交付高质量的软件，为人类造福。

安 川
2008年4月于北京

目 录

第 1 章 敏捷——高效软件开发之道	1	22 自动验收测试	90
第 2 章 态度决定一切	10	23 度量真实的进度	93
1 做事	12	24 倾听用户的声音	96
2 欲速则不达	15	第 6 章 敏捷编码	98
3 对事不对人	18	25 代码要清晰地表达意图	100
4 排除万难，奋勇前进	23	26 用代码沟通	105
第 3 章 学无止境	26	27 动态评估取舍	110
5 跟踪变化	28	28 增量式编程	113
6 对团队投资	31	29 保持简单	115
7 懂得丢弃	34	30 编写内聚的代码	117
8 打破砂锅问到底	37	31 告知，不要询问	121
9 把握开发节奏	40	32 根据契约进行替换	124
第 4 章 交付用户想要的软件	43	第 7 章 敏捷调试	128
10 让客户做决定	45	33 记录问题解决日志	129
11 让设计指导而不是操纵开发	48	34 警告就是错误	132
12 合理地使用技术	52	35 对问题各个击破	136
13 保持可以发布	55	36 报告所有的异常	139
14 提早集成，频繁集成	58	37 提供有用的错误信息	141
15 提早实现自动化部署	61	第 8 章 敏捷协作	146
16 使用演示获得频繁反馈	64	38 定期安排会面时间	148
17 使用短迭代，增量发布	69	39 架构师必须写代码	152
18 固定的价格就意味着背叛承诺	73	40 实行代码集体所有制	155
第 5 章 敏捷反馈	76	41 成为指导者	157
19 守护天使	78	42 允许大家自己想办法	160
20 先用它再实现它	82	43 准备好后再共享代码	162
21 不同环境，就有不同问题	87	44 做代码复查	165

2 目 录

45 及时通报进展与问题	168	9.4 引入敏捷：程序员指南	174
第9章 尾声：走向敏捷	170	9.5 结束了吗	175
9.1 只要一个新的习惯	170	附录A 资源	176
9.2 拯救濒临失败的项目	171	索引	182
9.3 引入敏捷：管理者指南	172		

第 1 章

敏捷——高效软件开发之道

不管路走了多远，错了就要重新返回。

——土耳其谚语

这句土耳其谚语的含义显而易见，你也会认同这是软件开发应该遵守的原则。但很多时候，开发人员（包括我们自己）发现自己走错路后，却不愿意立即回头，而是抱着迟早会步入正轨的侥幸心理，继续错下去。人们会想，或许差不多吧，或许错误不像想象的那么严重。假使开发软件是个确定的、线性的过程，我们随时可以撤回来，如同谚语中所说的那样。然而，它却不是。

相反，软件开发更像是在冲浪——一直处于动态、不断变化的环境中。大海本身无法预知，充满风险，并且海里还可能有鲨鱼出没。

冲浪之所以如此有挑战性，是因为波浪各不相同。在冲浪现场，每次波浪都是独一无二的，冲浪的动作也会各不相同。例如，沙滩边的波浪和峭壁下的波浪就有很大的区别。

在软件开发领域里，在项目研发过程中出现的需求变化和挑战就是你在冲浪时要应对的海浪——它们从不停止并且永远变化，像波浪一样。在不同的业务领域和应用下，软件项目具有不同的形式，带来了不同的挑战。甚至还有鲨鱼以各种伪装出没。

软件项目的成败，依赖于整个项目团队中所有开发成员的技术水平，对他们的培训，以及他们各自的能力高低。就像成功的冲浪手一样，开发人员必须也是技术扎实、懂得掌握平衡和能够敏捷行事的人。不管是预料之外的波浪冲击，还是意想不到的设计失败，在这两种情况下敏捷都意味着可以快速地适应变化。

敏捷开发宣言

我们正通过亲身实践和帮助他人实践，揭示了一些更好的软件开发方法。通过这项工作，我们认为：

- 个体和交互胜过过程和工具
- 可工作的软件胜过面面俱到的文档
- 客户协作胜过合同谈判
- 响应变化胜过遵循计划

虽然右项也有价值，但我们认为左项具有更大的价值。

敏捷宣言作者，2001年版权所有。

更多详细信息可以访问agilemanifesto.org。

敏捷的精神

那么，到底什么是敏捷开发方法？整个敏捷开发方法的运动从何而来呢？

2001年2月，17位志愿者（包括作者之一Andy在内）聚集在美国犹他州雪鸟度假胜地，讨论一个新的软件开发趋势，这个趋势被不严格地称为“轻量型软件开发过程”。

我们都见过了因为开发过程的冗余、笨重、繁杂而失败的项目。世上应该有一种更好的软件开发方法——只关注真正重要的事情，少关注那些占用大量时间而无甚裨益的不重要的事情。

这些志愿者们给这个方法学取名为**敏捷**。他们审视了这种新的软件开发方法，并且发布了敏捷开发宣言：一种把以人为本、团队合作、快速响应变化和可工作的软件作为宗旨的开发方法（本页最开始的方框里就是宣言的内容）。

敏捷方法可以快速地响应变化，它强调团队合作，人们专注于具体可行的目标（实现真正可以工作的软件），这就是敏捷的精神。它打破了那种基于计划的瀑布式软件开发方法，将软件开发的实际重点转移到一种更加自然和可持续的开发方式上。

它要求团队中的每一个人（包括与团队合作的人）都具备职业精神，并积极地期