

The Addison Wesley Signature Series

用户故事 与敏捷方法

BOOK A KENT BECK SIGNATURE

- 敏捷大师Mike Cohn的软件需求方法圣经
- 小型团队(项目)不可或缺的敏捷开发宝典
- 亚马逊五星级长销图书，敏捷社区重点推荐
- 结合精髓和实例，充分演绎用户故事的智慧

(美) Mike Cohn 著
Kent Beck 作序
石永超 张博超 译
李国彪 滕振宇 审校



清华大学出版社

用户故事与敏捷方法

(美)Mike Cohn 著

石永超 张博超 译

李国彪 滕振宇 审校

清华大学出版社

北 京

内 容 简 介

本书详细介绍了用户故事与敏捷开发方法的结合,诠释了用户故事的重要价值,用户故事的实践过程,良好用户故事编写准则,如何搜集和整理用户故事,如何排列用户故事的优先级,进而澄清真正适合用户需求的、有价值的功能需求。

本书对于软件开发人员、测试人员、需求分析师和管理者,具有实际的指导意义和重要的参考价值。

Simplified Chinese edition copyright © 2010 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: User Stories Applied: For Agile Software Development © 2010
EISBN: 978-0-321-20568-1

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2009-6813

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

用户故事与敏捷方法/(美)科恩(Cohn M.)著;石永超,张博超译;李国彪,滕振宇审校. —北京:清华大学出版社, 2010.4

书名原文: User Stories Applied: For Agile Software Development

ISBN 978-7-302-22340-5

I. 用… II. ①科… ②石… ③张… ④李… ⑤滕… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字(2010)第 056499 号

责任编辑: 文开琪

责任印制: 杨 艳

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市李旗庄少明装订厂

经 销: 全国新华书店

开 本: 185×230 印 张: 15.25 字 数: 355 千字

版 次: 2010 年 4 月第 1 版 印 次: 2010 年 4 月第 1 次印刷

印 数: 1~4000

定 价: 39.00 元

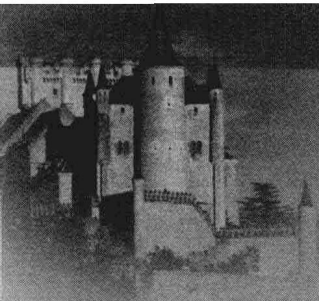
产品编号: 034966-01

献给 Laura，因你用心阅读本书；

献给 Savannah，因你对阅读的喜爱；

献给 Delaney，因你总是设法让大家相信你将会阅读。

有了你们这样的读者，写作变得如此轻松愉快。



Kent 推荐序

你如何确定一个软件系统应该做什么？然后，你怎样和不同的人去沟通你的决定？本书就是研究这个复杂问题的。这个问题很难回答，因为不同的参与者有不同的需求。项目经理想跟踪进度，开发人员想实现系统，产品经理想要灵活性，测试人员想要度量，而用户想要一个可用的系统。在这些充满冲突的视角中，想要做出一个人都支持、皆大欢喜的决定，并且持续数月或者数年都保持这种平衡是很困难的事情。

与早期的种种方法(需求分析、用户实例和场景)一样，在这本书中，Mike Cohn 探索的解决方案也是试图解决这个问题。是什么如此复杂？你写下想做的事情然后照着它做。日益出现的多个解决方案表明这个问题并不像最初那样简单，这些方案的不同在于你写下了什么，何时写的。

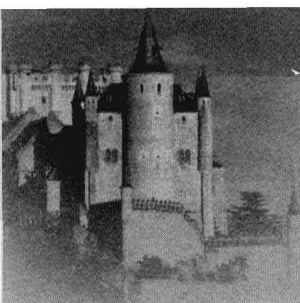
采用用户故事这一方法，是从写下两条信息开始的：每一个系统需要实现的目标和实现那个目标所需要的大致成本。这只需要几句话，却能给你其他方法没有给出的信息。遵循“最后责任时刻”(Last Responsible Moment)的原则，团队要等到开始实现软件特性前才写下特性的具体细节。

这个简单的时刻调换可以带来两个主要效果。首先，团队很容易在早期开发就开始实现最重要的特性，而那时其他特性还很模糊。当你加入新特性的时候，定义了每个特性行为细节的自动化测试可以保证此前的特性仍然正常工作。其次，在早期对特性进行成本考虑可以驱使我们起始阶段就对软件特性排列优先级，而不是在最后为了赶上发布日期而惊慌地缩减功能。

Mike 在用户故事方面积累的经验使得本书充满实用的建议，让用户故事可以在你们的开发团队里正常运转。祝愿你们有一个明确的、充满自信的开发过程。

Kent Beck

Three Rivers Institute



乐在用户故事与敏捷方法

软件开发是充满挑战的探险。行业文献所记载的事实及实战经验告诉我们，这些挑战常常导致痛苦的结果：失败的项目。任何参与过软件开发的人毫无疑问都碰到过一次或多次的失败。我自己的经历也是确凿的见证。

虽然软件开发项目失败的原因是方方面面的，但是软件需求确实被识别为最常见的痛苦根源。不论是行业的研究，还是权威人士的讨论，或者是我自身的经历，都指向此环节。针对此挑战，多年以来行业中的解决办法是使用越来越冗长的文档，尝试用精确的语言来记录越来越细化、越来越具体的所谓全面的需求。制定标准的各个组织提供各种的文件模板和相应的语言书写规范让大家来清楚地定义需求。数不尽的书籍和文章论述怎样用详细的文档来描述需求。不知道多少森林因为各个组织产出堆积如山的需求文档而被砍伐。即便如此，失败的项目仍然持续不断地出现而且大家仍然发现需求依然是主要问题所在。为什么会这样？

或许使用固化呆板的文档及精确的语言的想法是错误的方向？问题的出路是否不在于复杂，而在于简单？解决方法是否不在于尽力在前期定义出细节，而在于及时响应，开发随需而动(just in time)？是否不在于文档，而在于密切的交流？

Mike Cohn 的这本 *User Stories Applied* 英文版是这一领域的奠基之作。现在终于有此书的中文版供中国广大的敏捷实践者参考。此书的译者们是国内敏捷界被认可的资深人士，确保了翻译的质量和准确性。

通过捕获一目了然的格式一致的用户故事，我们掌握了足够的信息可以继续前行。我们给用户故事排优先级，并且先开发最重要的。我们按需及时展开，通过交谈获取所需要的细节。我们产出具有真正价值的软件，成果可以被审核和验证甚至交付使用。然后我们继续开发后续最重要的用户故事。

这是否行得通？

是的！我最近在年终派对上有机会与我的团队反思了这个话题。在这个充满挑战、项目失败乃家常便饭的行业中，我们的组织有理由为我们的成果庆祝。因为在过去两年

半时间中，我们交付了 20 多个项目而没有一个失败。当然，需求只是项目成功的一个因素，但是我们确实深深体会到敏捷需求管理方式减轻了挑战无处不在的软件开发探险过程中的痛苦。

读好这本书。遵循它的原则。在此希望读者也能够享受到敏捷的好处。

Vernon Stinebaker(史文林)

Scrum 及 Agile Evangelist Perficient 技术总监

2010 年春于杭州



使用用户故事，不仅仅是为了快

如果去 Amazon.com 搜索 “user story” 这个关键字，你会惊奇地发现：*User Stories Applied*——这是唯一一本以用户故事为主题的书籍。随意翻开一页，你都可以从中发现有关用户故事乃至敏捷的真知灼见，无怪乎读者会称之为“用户故事圣经”。在“圣经”出现之前，我们又是如何应对需求的？

数年之前，每当我接手一个项目的时候，总会有几大本厚厚的文档扔在面前，不外乎需求规格说明、概要设计和详细设计。尤其是需求规格说明，拿起来翻开一看，那些格式化的语言就变成了世界上最好的催眠曲。读尚且如此，遑论写乎？人的大脑同时处理事物的能力是有限的，写这种正式的需求规格说明，既要思考内容是否表述客户的真实意图，还要想着符合公司对于格式、用词等等方面要求，怎能不心生厌惧？更何况大脑里还总回响着一个问题：“这东西写出来，能有人认真看么？”在 Wikipedia 的 User Story 页面上，这样讲述使用用户故事的目的：以更快的速度、更少的消耗应对现实世界需求的快速变化。高速互联网时代，更是如此。你还在吭哧吭哧地写需求规格说明，竞争对手的系统很可能都已经上线试运行了。

使用用户故事，不仅仅是为了快。

从大脑认知的角度来看，正式的规约说明，主要以格式化的文字为表现形式。而我们的大脑对此是很难提起兴趣的。O’ Reilly 有一个 Head First 系列广为人知，并在市场上热卖，这套书就是在这方面做出了突破。在每一本 Head First 书籍前，都有同样几页图文并茂的说明，介绍我们大脑的认知方式。面对同样一个主题，只有通过多种不同方式、不同活动的刺激，大脑才能深刻理解并记忆。显然，单一的需求规格说明无法做到这一点。回过头来看用户故事，著名的极限编程创始人之一 Ron Jeffries 提出了“3C”原则：Card、Conversation 和 Confirmation。使用卡片记录用户故事，一方面可以隐藏低层细节，另一方面也方便各方人员在白板上将其移来摆去，以整体图形的方式将与客户需求有关的内容深深印在团队的脑海中，更不用说这样给项目规划带来的好处。对话，是为了促使团队与客户之间的沟通，让大家谈论需求，大声说出来，这种活动也调动了大脑的不同区域，让人们能把相关内容学得更快，记得更牢，同时还可促进团队和客户之

间的沟通，加强人际联系，何乐而不为？用户故事的确认环节，则是以反复的方式，与用户确认某个具体使用场景中的关键细节，从而不会导致遗漏。

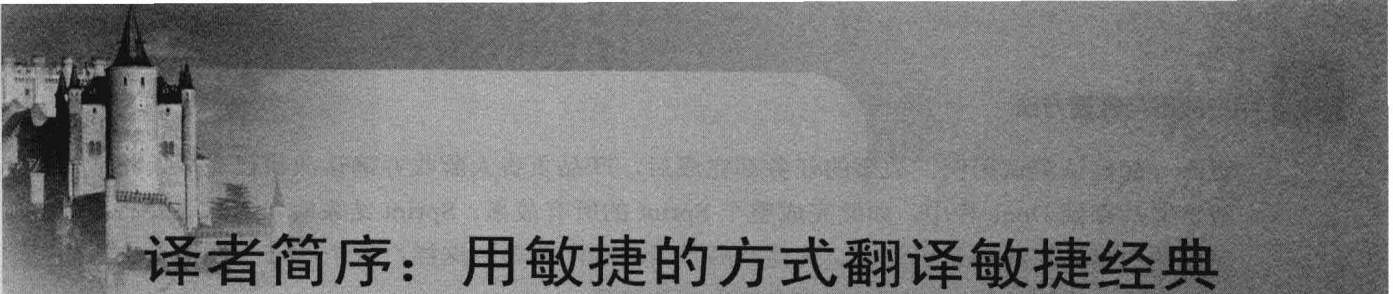
从软件开发的角入手，使用用户故事，从用户角度出发描述功能，这让我们可以站在最终用户的立场考虑问题，避免程序员的自行其是。同时还能促使团队按功能特性实现需求，而不是按架构层次，这样可以降低系统开发进入后期出现整体风险的可能，带来更大的灵活性。

不过，我想指出的是：用户故事、规划会议等类似系列非技术实践，实施起来可能并不复杂，但是必须要结合 TDD、持续集成、重构等技术实践，否则要想产生高质量的代码就是空谈，由此而完成的软件产品或是项目也必将成为沙滩上的城堡，连一个小小的浪头都无法抵挡，一瞬间即可坍塌。这两三年里，国内的敏捷声势一浪高过一浪，可大家更多讨论的也都是 Scrum 这个管理概念框架，如何减少技术债务、如何做好单元测试等方面的话题可谓寥寥。倘若写代码的基本功不扎实，长此以往，出现问题的时候，就会有人跳出来归罪于敏捷。我们作为软件开发从业人员，对此不可轻视。

回到本书，虽然它第一次正式出版是在 6 年之前，然而其中的内容却绝不过时。更值得指出的是，本书的译者团队本着快速迭代、频繁沟通的原则，以敏捷的方式完成了翻译，这个过程中，他们甚至还使用了持续集成！作为 InfoQ 中文站敏捷社区的首席编辑，我希望他们能尽快分享自己在“敏捷翻译”方面的经验，让更多的人将水平更高、质量更好的敏捷乃至技术内容介绍到国内，推动国内软件开发行业的进一步发展。

郑 柯

InfoQ 中文站敏捷社区首席编辑



译者简序：用敏捷的方式翻译敏捷经典

数年前，Mike Cohn 写了这本 *User Stories Applied: For Agile Software Development*，而我从 2007 年才真正开始接触敏捷，没想到在 2009 年我竟有机会能够参与翻译这本有关用户故事的经典著作，我感到十分的荣幸。

敏捷开发近些年在国内软件开发公司中十分流行，因为它为软件开发指引了一个方向。而用户故事是敏捷实践中一个十分重要的环节。它能帮助我们高效地收集客户真正的需求。软件开发都起始于需求收集与分析。如果一开始需求都弄错了，软件的成功也就无从谈起。同时，用户故事带来了一个十分重要的作用，即高效沟通，不论是开发团队与客户的沟通，还是团队内部成员之间的沟通。沟通使客户和团队成员都朝同一个方向前进，这意味着更少的错误，更少的浪费、风险和成本。用户故事还是敏捷计划与估算的重要基础。

我十分有幸能在 Irdeto BSS 进行敏捷开发。在这里，我们使用 Scrum，结合 XP 进行开发。用户故事自然是不可或缺的。在这里，每个团队都有一个白板，上面贴有一些卡片。它们是这个 Sprint 团队计划完成的用户故事以及这些故事划分的任务。这些用户故事卡片概要描述了需求，形如“作为(角色)，我想要(功能)，以此(商业价值)”，有时上面还附着另一张卡片写着验收条件。在做计划和开发的时候，团队可以拿着这个用户故事讨论故事细节，故事如何才算完成，等等，正如 Ron Jeffries 所描述的 3C：Card(卡片)，Conversation(交谈)和 Confirmation(确认)。

用户故事从始至终贯穿于整个开发流程。首先产品负责人根据收集来的需求编写用户故事，放入产品 Backlog 中。在 Sprint 计划会议中，团队成员讨论其中的一些用户故事，细化故事细节，确定验收标准，使用 Planning Poker(计划扑克)估算故事点。然后，将故事分成一些小的任务，并估算工作时间。最后，将故事放入 Sprint Backlog 中，并按优先级排序。Sprint 开始时，故事卡片和任务卡片都放在白板的 To Do 栏，团队成员按故事的优先级挑选任务，将任务卡片挪到 Doing 栏。任务完成后，将任务卡片挪到 Done 栏。团队尽可能地先完成优先级高的故事。在故事开发的初始阶段，测试人员和产品负



责人一起确认测试用例。故事的任务都完成后，产品负责人验收并确认故事已完成，将故事卡片挪到 Done 栏中。如此完成整个 Sprint 的所有故事。Sprint 结束时，团队还要将完成的故事演示给利益相关人、其他产品负责人和团队等。这样，每个 Sprint 团队都会通过完成一系列的用户故事来向客户输出商业价值。

这次翻译我们一共四个人参与，我和石永超主要负责翻译，滕振宇和李国彪主要负责审核。前期的第一个月，我们几乎没有什么太大的进展。我们讨论过后，发现这次翻译其实是一个具有固定交付时间、固定范围(21 章和两个附录)且依赖虚拟兼职开发团队的项目，包含开发(翻译)和测试(审核)等工作，何不用敏捷的方式来开展？于是，第二个月开始，我们使用敏捷的思想来指导翻译工作。我们首先把每一章当作一个用户故事，每个故事估算一个故事点(这个是我们做的不足的地方，一个故事点显然不能准确反映各章的不同篇幅)。我们以一个星期为一个迭代，用一份 Excel 文档作电子白板。每个周末固定时间用 Skype 开一次网上语音会议。如同 Scrum 的每日例会一样，大家回答三个问题：这个星期我做了什么，下个星期准备做什么，有什么困难。然后讨论大家遇到的一些翻译难点，统一一些术语的翻译。大家在完成每个星期的翻译工作的同时，必须及时简单地更新 Excel 中故事的状态。这样一来，每个人都能及时知道每一章的进度，哪一章可以审核，哪一章没有人翻译，可以任领。同时，Excel 中还有燃尽图，告诉我们离目标还有多远，是否需要调整。如此，这份 Excel 文档就成为我们的另一个沟通工具。另外，我们还有十分重要的工具，如同我们的软件项目一样，对于项目中的所有文件(包括电子白板和术语表)，我们使用版本管理工具 Subversion 和持续集成工具 CruiseControl。我们每次签入，持续集成工具都会立即发一封邮件通知大家有新的改动，邮件包含有签入的描述，这样可以提醒大家某一章翻译完了，应该审核，等等^①。这样，从第二个月开始，我们的翻译工作就始终保持稳定的进度，团队通过更多更频繁的回馈不断学习成长，持续改善译稿和翻译过程，最终按时交付了翻译稿。

这次翻译成为我们软件开发之外的一次敏捷实践，获益良多。同样，我们也希望能够让更多的人了解敏捷，让更多从事软件开发的人和其他行业的人从中获益。翻译这本书也希望能传播敏捷思想与方法尽一份力。让更多的人了解用户故事，使用用户故事，带来更多成功的项目。

^① 其实，这远没有达到持续集成的终极效果，如果有时间，可以开发一个小程序，将现有各章的 Word 文档合并成一个最终可交付的 Word 文档，甚至做一些简单的关键字检查等单元测试^②。

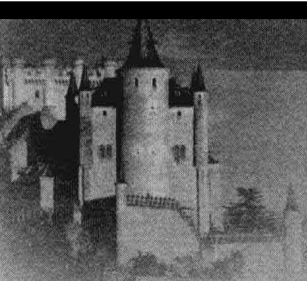
译者简序：用敏捷的方式翻译敏捷经典

在此感谢一起翻译的伙伴们：李国彪、滕振宇和石永超。感谢你们让我能够获得参与翻译此书的机会，我从中学习了很多很多，不仅仅是对用户故事更深入的了解，还有在这次翻译过程中对敏捷更全面、深刻的理解。

张博超

译者团队代表

2010 年春于上海



前言

在 20 世纪 90 年代中期，我一直感到非常愧疚。那时我在为一家公司工作，这家公司每年都会收购一家公司。每次收购一家公司后，我就会被派去负责他们的软件开发部门。收购的每个软件开发部门都有华丽而厚重的需求文档。我当然会感到愧疚，因为我自己的部门从没有做过这样漂亮的需求说明。然而，我的部门在生产软件方面远比我们收购的部门要成功得多。

我清楚我们成功的动因。但是我仍然有一种挥之不去、怅然若失的感觉，如果我们能写出长长的需求文档，我们可能会更成功。毕竟那是在我当时读到的书和文章里所描述的做法。如果成功的软件开发团队都在编写华丽的需求文档，看起来我们也应该这么做。但是我们从来没有时间做。我们的项目总是太重要，时间非常紧迫，以至于我们从一开始就没有时间做文档。

因为我们从来没有时间写长篇大论的漂亮的需求文档，所以便选定了一个可以和用户交谈的方法。我们不是把需求写下来，传来传去，并在时间不够用的时候还在商谈，而是和客户交谈。我们会在纸上画一些界面样例，有时做一些原型，我们经常开发一点就给潜在的用户展示我们开发了些什么。我们至少每月一次邀请一些典型用户，向他们演示我们开发的功能。我们贴近我们的用户，向用户展示我们一点一滴的进度，这样便在不知不觉中发现一个不需要漂亮的需求文档就可以成功的方法。

我还是觉得愧疚，觉得我们并没有按规范来做事。

1999 年，Kent Beck 出版了一本革命性的书《解析极限编程——拥抱变化》。一夜之间，我的这份愧疚烟消云散。终于有人旗帜鲜明地提出开发人员和客户之间用交谈取代“写文档——商谈——再写文档”。Kent 阐明了许多事情，带给我许多有用的方法。但最重要的是，他证明我在实践中领悟到的是正确的。

大量预先的需求收集和文档会以很多方式导致项目失败。最常见的是需求文档变成软件开发的目的。应当只在对交付软件有用时才写需求文档。

大量预先的需求收集和文档导致项目失败的另一个方式是记录语言的不准确性。我记得许多年前听到过的一个小孩洗澡的故事。小孩的父亲在浴缸中放满了水，帮助孩子

进入水中。小孩子大概只有两三岁，她先用脚趾头试了下水温，告诉父亲说“水要暖点”(make it warmer)。父亲把手放入水中，惊奇地发现，水并不冷，水已经比他女儿习惯的水温更热了。

父亲思考了一下孩子的要求，发现他们的沟通出现了问题，相同的词代表不同的意思。孩子的要求“水要暖点”对任何大人的理解都和“提高水温”是一样的。然而对孩子而言，“水要暖点”意思却是“让水温更接近我认为的暖的温度”。

语句，尤其是写在书面上的时候，对于表达像软件这么复杂的需求是比较有限的。由于它们可能被误解，所以需要与开发人员、客户和用户频繁沟通。用户故事提供了一个方法，让我们可以写下我们不会遗忘且我们可以估算和计划的，同时还鼓励沟通。

读完本书第 I 部分后，你即可改变以前总是严谨地记录下每个需求细节的方式。读完本书，你将知道在实现故事驱动流程的所有必要信息。本书由 4 部分和两个附录组成。

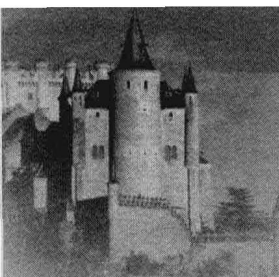
第 I 部分“起步” 描述用户故事编写须知。用户故事的目的之一是让大家交谈而不是写。第 I 部分的目的是让你尽快开始交谈。第 1 章概要介绍什么是用户故事，如何使用故事。接下来 9 章详细介绍编写用户故事，通过用户角色建模收集故事，在不能直接访问真实的最终用户时编写故事，测试用户故事。第 I 部分结束时，用一章的篇幅介绍用户故事改进指南。

第 II 部分“估算和计划” 有了一系列用户故事后，首先要做的是回答“要花多长时间来开发？”第 II 部分的各章全面介绍如何用故事点估算故事，如何做一个 3~6 个月的发布计划，具体如何为即将到来的一轮迭代做计划，如何测量进度和评估项目是否如期进行。

第 III 部分“经常讨论的话题” 第 III 部分开始讲述故事与用例，软件需求说明和交互设计场景之间的区别。紧接着的各章介绍用户故事特有的优势，如何发现错误，如何在敏捷过程 Scrum 中使用故事。第 III 部分的最后一章研究了一些小问题，例如故事写在纸质卡片上还是记录在软件系统中，如何处理非功能性需求。

第 IV 部分“一个完整的实例” 用一个扩充的实例做一个综述。如果我们认为开发人员能通过故事充分理解用户的需求，那么用一个扩展的案例展示用户故事的所有方面来总结本书是非常重要的。

第 V 部分“附录” 用户故事源自极限编程。虽然读本书不需要熟悉极限编程，不过，附录 A 仍然大致介绍了极限编程。附录 B 则包含每章问题的答案。



致 谢

本书受益于众多审阅者的评论。我需要特别感谢 Marco Abis, Dave Astels, Steve Bannerman, Steve Berczuk, Lyn Bain, Dan Brown, Laura Cohn, Ron Crocker, Ward Cunningham, Rachel Davies, Robert Ellsworth, Doris Ford, John Gilman, Sven Gorts, Deb Hartmann, Chris Leslie, Chin Keong Ling, Philip, Keith Ray, Michele Sliger, Jeff Tatelman, Anko Tijman, Trond Wingard, Jason Yip 以及其他几位匿名的审阅者。

我向本书正式的审阅者致以诚挚的感谢: Ron Jeffries, Tom Poppendieck 及 Bill Wake。Ron 使我保持诚实和敏捷。Tom 让我看得到我原来忽视的一些想法。而 Bill 则使我不会偏离方向以及与我分享其 INVEST 缩写词模型。我很骄傲能与这三位在此项目中共事, 因为他们任何一位提出的很多建议都对改善本书做出了不可估量的贡献。

我也要感谢 Lisa Crispin, *Testing Extreme Programming* 一书的作者。她鼓励我写这本书, 并告诉我她与 Addison-Wesley 出版社的良好合作经历。没有她的鼓舞, 我可能现在都还没有动笔。

在过去 9 年中, 我经常就我所知的东西与 Tod Golding 进行争论。其实我们的共识远远超过我们之间的争执。不管怎样, 我常常从这些争论中学到东西。感谢 Tod 这些年来他所教我的。本书的很多内容因为我与他的那些交谈而增色不少。

感谢 Alex Viggio 和 Denver XP 小组的每一位, 让我有机会展示本书中的很多早期想法。也感谢 Mark Mosholder 和 J. B. Rainsberger, 告诉我他们怎样使用软件来替代记录卡片。也要感谢 Kenny Rubin, *Succeeding With Objects* 的作者之一(与 Adele Goldberg 合著)。他们书中显而易见的自豪感促使我在搁笔数年后重新开始写作。

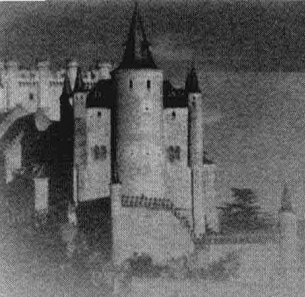
衷心感谢 Fast401k 的创办人 Mark 和 Dan Gutrich。他们由衷地拥抱用户故事和 Scrum。还要感谢 Fast401k 我的每一位同事, 我们正在越来越接近实现我们的目标——成为科罗拉多州最好的团队之一。

千言万语都无法表达对我家人的感激, 因为那么多的时间我无法与他们相伴。感谢我那两个令人惊奇的女儿兼小公主, Savannah 和 Delaney。特别感谢我曼妙美丽的太太

Laura, 她因为我的繁忙而需要额外的操劳。

我还欠 Addison-Wesley 团队太多的谢意。Paul Petralia 使过程由始至终都非常愉快。Michele Vincenti 是事情的推进者。Lisa Iarkowski 无私地帮助我使用 FrameMaker。Gail Cocker 为我的图表润色不少。Nick Radhuber 在最后使所有一切努力成功地整合在一起。

最后一点, 同时也是非常重要的一点, 谢谢 Kent Beck 的真知灼见和时间, 感谢他把本书包含在他的签名系列中。



目 录

第I部分 起 步

第1章 概览.....	3	第3章 用户角色建模.....	27
什么是用户故事?	4	用户角色.....	27
细节在哪里?	5	角色建模的步骤.....	28
“必须多长时间完成?”	6	通过头脑风暴, 列出初始的用户	
客户团队	7	角色集合	29
使用故事的过程是怎么样的?	7	整理最初的角色集合	30
规划发布和迭代	9	整合角色	31
什么是验收测试?	11	提炼角色	32
为什么要变?	12	两个额外的技术.....	33
小结	13	虚构人物	33
问题	14	极端人物	34
第2章 编写故事.....	15	如果有现场用户该如何?	35
独立的	15	小结	35
可讨论的	16	开发人员职责	35
对用户或客户有价值的.....	18	客户职责.....	35
可估计的	19	问题	36
小的	20	第4章 搜集故事.....	37
分割故事.....	21	引出和捕捉是不合用的	37
合并故事.....	23	够用就行, 不是吗?	38
可测试的	23	方法	38
小结	24	用户访谈	39
开发人员职责	25	问卷调查	41
客户团队职责	25	观察	41
问题	25		