

敏捷项目管理系列丛书



发“15890”到  
106900292251  
成为世纪畅优会员, 可阅读更多图  
书, 获得增值在线课程(教学资源)

PMI-ACP<sup>SM</sup>参考用书

# 敏捷项目管理

## 企业级实践与案例

# P

Project Management

the Agile Way:

Making it Work in the Enterprise

(美) 约翰·C·古德帕斯丘 著  
(John C. Goodpasture)

陈秋萍 译



NLIC2970762312



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

敏捷项目管理系列丛书

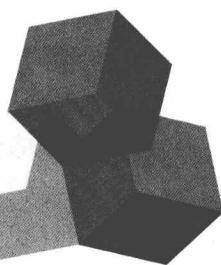
PMI-ACP<sup>SM</sup>参考用书

# 敏捷项目管理

## P 企业级实践与案例

Project Management  
the Agile Way:

Making it Work in the Enterprise



NLIC 2970762312

(美) 约翰·C·古德帕斯丘 著  
(John C. Goodpasture)

陈秋萍 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

John C. Goodpasture, PMP: Project Management the Agile Way: Making it Work in the Enterprise

Copyright © 2010 J. Ross Publishing, Inc.

This translation is published by arrangement with J. Ross Publishing, Inc. All rights reserved.

本书中文简体字版由 J. Ross Publishing, Inc. 授权电子工业出版社独家出版发行。未经书面许可, 不得以任何方式抄袭、复制或节录本书中的任何内容。

版权贸易合同登记号 图字: 01-2011-5771

#### 图书在版编目(CIP)数据

敏捷项目管理: 企业级实践与案例 / (美) 古德帕斯丘 (Goodpasture, J.C.) 著; 陈秋萍译.  
北京: 电子工业出版社, 2012.4

(敏捷项目管理系列丛书)

书名原文: Project Management the Agile Way: Making it Work in the Enterprise

ISBN 978-7-121-15890-2

I. ①敏… II. ①古… ②陈… III. ①企业管理: 项目管理—应用软件 IV. ①F270  
②TP317

中国版本图书馆 CIP 数据核字(2012)第 025008 号

责任编辑: 杨洪军

印 刷: 三河市鑫金马印装有限公司  
装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 720×1000 1/16 印张: 21.25 字数: 346 千字

印 次: 2012 年 4 月第 1 次印刷

定 价: 48.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

---

# 前言

---

本书从项目的管理的角度来讨论敏捷方法论。

迪尔伯特：我们还需要 3 个程序员。

老板：使用敏捷方法。

迪尔伯特：敏捷编程并不意味着用更少的人做更多的工作。

老板：那就给我找出意味着那样的词，然后再来找我。¹

[(迪尔伯特 (Dilbert™) 是斯科特·亚当斯创造的一个卡通人物。)]

在项目管理中出现了一些具有挑战性的新观点，特别适合管理创新性和技术性的项目——尤其是软件项目，这样的项目对于用户来说日益复杂。对于我们所讨论的这些观点，有一个总称，即敏捷。

## 敏捷的使命

敏捷意味着小的团队共同协作以完成如下使命：

将新功能和特征以经常性的、增量的发布形式交付，功能和特征的优先级根据需求和交付能力来排列；功能和特征根据用户的反映和反馈迭代演化，并力求实现可能的最佳价值。²

敏捷的方法包括很多：Scrum、极限编程 (Extreme Programming, XP)、Crystal 族、EVO 以及 RAD 的敏捷模式——动态系统开发方法 (Dynamic System Development Mothod, DSDM)。³ 此外还有特征驱动开发 (Feature-driven Design,

FDD)、自适应软件开发 (Adaptive Software Development)、精益开发 (Lean Development)、团队软件过程 (Team Software Process, TSP) 和个人软件过程 (Personal Software Process, PSP)。<sup>4</sup> 在这些之前, 还有一些更早的方法为敏捷这一平台奠定了基础, 如螺旋模型 (Spiral)、统一软件开发过程 (Rational Unified Process, RUP)、联合应用开发 (Joint Application Development, JAD) 和快速应用开发 (Rapid Application Development, RAD)。<sup>5</sup> 也有一些新的方法致力于开发出更可靠的产品, 如净室开发 (CleanRoom), 但这些方法并不是敏捷方法。

所有的敏捷方法都有一个共同的特性: 每种方法都用自己的方式来解决在创建复杂的、无形的、具有用户接口的交付物时所遇到的困境, 也就是, 用户会说他们所需要的和想要的总是无法确定。的确, 这种解决方案通常就决定了需求——“当我看到, 我就知道!”

敏捷方法使小团队能够对不断变化的环境迅速做出反应, 并在更长的商业和市场周期中更好地为客户提供价值。敏捷团队实现较短时间内能够稳定的一小部分需求, 在解决方案形成的过程中不断征求客户和用户的意见, 经常性地发布产品增量, 并在每次发布之后要求严肃的评论。

## 敏捷方法解决根本问题

---

敏捷方法在业界中并不是一夜形成的。许多年来, 用于解决客户需求的过程不断被改进——因为项目和项目管理方法总是无法可靠地满足企业需求。有太多时候, 要么项目交付了错误的结果, 要么本是正确的事情被错误地执行, 要么就是根本交付不出什么。在众多项目中, 似乎仅有一小部分项目按照正确的方法得到正确的结果。

为此, 人们提出了许多解决方案, 并不断改进。例如, 在瀑布模型中加入了反馈和迭代,<sup>6</sup> 引进了成熟度模型来度量和鼓励员工及组织, 越来越强调需求的重要性。还引入了面谈和故事板、关联分析、跟踪数据库、UML 模型以及其他许多 CASE 工具和框架, 来保证各个方面都覆盖到。<sup>7</sup>

现在,由于越来越多的项目复杂程度几乎不可想象,因此也就更难将事情做对。过去曾强调改进项目管理艺术和科学,现在则转为强调按照正确的方法来做正确的事情,将质量包含到过程以及 workflow 中。毫无疑问,近年来许多行业的项目经理和项目集经理已经提升到专业水平。项目管理的不同知识体系之间也有了交叉。其中的几个例子是项目管理协会 (Project Management Institute, PMI) 的《项目管理知识体系》(Project Management Body of Knowledge, PMBOK®)<sup>8</sup> 及其支持标准和成熟度模型、软件工程研究所 (Software Engineering Institute) 的《能力成熟度模型:集成》(Capability Maturity Model: Integration), 英国的 PRINCE2, 以及美国国防部的程序和采购管理。

但是,这些体系并不是敏捷方法。它们在每个细节上并不完全一样。它们都提倡项目经理具有强有力的命令和控制角色;它们都重视明确的过程能力高于生产可重复的结果,它们都是集中计划的系列化的项目方法。

近来,除了“正确的方法”之外,人们也越来越重视“做正确的事情”。人们开始认识到,当按传统方法收集需求并建立基线时,什么是正确的通常都还是未知的或不可知的。经验表明,开发人员可能要等到用户看到实际产品的样子并有一些体验之后才能知道。很多人开始认识到,复杂性、与遗留系统的互操作性以及用户体验,都存在着太多变数和复杂的因素,以至于很难开始就把一切都想象好并且开始凭想象实施。所以,在一些重要的方法上,敏捷方法与传统的思考方式不同。

#### 敏捷方法与传统方法不同

- 需求是如此重要,不能只在项目开始进行分析之后就置之不顾;随着用户交互和解释的进行,所有隐含条件都显示出来时,需求也必须进行演化。
- 过程是逐渐形成来适应环境的;控制度量是根据经验来决定的,而不是像六西格玛那样根据历史绩效来决定的。<sup>9</sup>
- 计划很重要,但是遵循计划并没有满足客户那么重要。

传统方法

传统方法的跟踪记录存在着不少问题。

StandishGroup 的《Chaos 报告》(1994)发现了一个令人震惊的结果：在它所调查的软件项目中，近 1/3 的项目没有完成，有一半大大超过成本预算。只有 16%按计划完成。<sup>10</sup>

不过，到 2007 年，《Chaos 报告》发现了可观的改善：根据《SDTimes》的在线评论，成功的项目从原来的 16%增长到约 32%。<sup>11</sup>

该评论还指出，StandishGroup 的创始人及联合董事长吉姆·约翰逊将改善的原因归为以下三点：更好的项目管理、迭代开发和新兴的 Web 基础设施。<sup>12</sup>

在 2009 年的报告中，StandishGroup 的报告指出，项目成功率又有所下降。<sup>13</sup>

□ 敏捷方法是否可行

本书的目的在于解决这些问题：

- 当遇到客户没有说出来的需求或未知的需求时，能否用敏捷方法？项目经理对使用敏捷方法生产出可接受的项目结果有多大信心？
- 对于大规模的项目、有遗留投资需要保护的项目、因得不到充分信任而寸步难行的项目以及需要对投资人及企业管理层做出肯定的项目，敏捷方法能有多大适用性？

现在，微软和 IBM 在一些项目上也在使用敏捷方法，这一事实可能会让人觉得有一点保证。

关于敏捷方法，必须指出，敏捷方法是可行的，确实成功过，确实能够缩短进度，提供高质量的产品。<sup>14</sup>但是敏捷并不是银弹。它的方法并不适合任何情况，而只适合拥有合适环境和管理人才的项目。

|            |                                                                                                                                  |
|------------|----------------------------------------------------------------------------------------------------------------------------------|
| 项目管理<br>贴士 | 敏捷方法                                                                                                                             |
|            | <ul style="list-style-type: none"><li>• 本书所说的敏捷方法与传统的项目协议在管理范围、成本和进度上有所不同。</li><li>• 敏捷是需求在用户看到实际产品之前不断变化、未知或者不可知时的选择。</li></ul> |

**项目管理  
贴士**

- 敏捷方法最适合几个小团队、通常不超过 50 个开发人员的情境。
- 敏捷方法更适用于内部项目，比较不适用于有合同约定的情况；不适用于固定总价合同。
- 敏捷更适用于在一起工作的团队，而较不适用于具有文化差异、沟通渠道也有限的虚拟团队。
- 对于安全性至关重要的项目，必须使用以过程为中心的方法，如净室。这样的项目用敏捷方法并不适合。

**□ 敏捷可能是答案**

项目经理应该认真地了解一下敏捷方法。绩效和客户价值实现的不足——这些问题在 Web 时代的快速企业周期下变得更加严重——使业界的创新之士开始以完全不同的眼光来看待整件事情。从产品开发社区、软件工程社区到系统工程社区，完全虚构的和实际的协议，都有人提出并应用于实践中。敏捷方法及实践不仅改变了利用项目人才的方式，也改变了几十年来那种直观地记录项目事件的方式。随着近来对主流敏捷方法的推动，众多项目管理专业人员也开始认真地对待这些观点。

在本书中，读者将读到一些实践，这些实践将提供新的协作、分配工作和度量结果的方法。客户担任不同的角色，作为几乎实时的产品负责人；客户的参与度和可靠性与项目成功与否的关系更加密切。重视满足客户胜于循规蹈矩地遵循计划。当然，成功的部分秘诀在于项目能够较早地获得收益，对于自费项目则能够较早地看到可行性。另外，敏捷方法也能够大大提高客户的满意度，因为它们比传统的序列化方法更早地实现价值。

能够处理改变的需求，甚至在项目生命周期的较后期也能够处理，这一点是敏捷方法的优势。能够在项目的较后期处理需求，有助于降低风险与损失量的比例曲线，因此也改变了项目治理的需要。

本书要讨论下面四种方法：

1. Scrum。Scrum 是主流的管理框架。虽然有一套 Scrum 规则，但它并不是实际技术实践的规范。Scrum 在敏捷方法中是最简单的，可能也是当今最流行的。

2. XP。XP 的纪律性很强，包含特定的、明确的软件实践。与 Scrum 相比，XP 更面向工程，在管理实践上的指令性不像 Scrum 那么强。

3. Crystal 族。这是最带有感情色彩的方法，自称是以人为动力的。它承认，方法必须适合项目规模。Crystal 是不那么强调个人纪律的 XP；它强调文档较多一些，而较少依赖于个人沟通。

4. EVO。这是真正的系统工程师的方法，使用增量和演化的软件实践。EVO 提倡计划—实施—检查—处理的周期；它是一系列的增量瀑布。

本书也会讨论螺旋方法。螺旋方法的历史比上述四种敏捷方法还早 20 年。螺旋方法比任何敏捷方法都更好地解决了可行性问题。在项目的生产阶段，螺旋方法学家可能会转换到一些其他的方法上。

## 本书的目标读者

---

本书是由专业人士为专业人士而写的。本书的目标读者是项目经理、架构师和系统分析师，他们应该是具有经验的、掌握经典和传统的项目管理方法，而今要开始未知的旅程。在本书中，经理、架构师和系统分析师不仅可以找到对于新的、不同的实践简洁而实用的解释，也能找到将敏捷方法与更主流的、他们更熟悉的方法相结合的提示和建议。

如果你涉及技术性的项目和项目集，并且：

- 想了解新的、面向结果的替代方法；
- 希望提高项目的价值；
- 由于被分配到使用敏捷方法的项目上而寻求关于敏捷的知识；

那么，你就应该读一读本书。

## 本书的章节

---

第1章是关于本书主体所要讨论的四种方法的快速阅读。接下来的几章将围绕敏捷方法讨论项目管理的几个特定主题。

第2章是关于商业计划的。要实现有益于企业及其受益人的策略，项目是手段。敏捷商业计划尊重和鼓励将企业周期与客户需求的重要性和紧迫性融合在一起。本章将讨论如何有效地调整商业计划的实践，使之适应敏捷方法。

第3章将讨论质量，这可能是采用敏捷方法最重要的动机之一。质量不仅是没有错误，更是一个整体的概念：适应能力、功能和形式；对实现客户要求的时间以及价值主张的承诺；对经济的使用和维护的适合程度；以及对满足干系人对业务绩效期望的承诺。

第4章将讨论测试。测试是敏捷及迭代方法中主要的质量工具之一。实际上，测试驱动的开发被推荐作为设计工具、回归的辅助工具和质量工具。因为测试在敏捷方法中具有突出的地位，所以第4章介绍了如何计划测试、安排测试计分卡以及如何整合其他方法，如抽样测试和假设测试。

第5章是关于范围以及收集与组织需求的方法。第5章介绍了敏捷项目的工作分解结构、评估复杂性的方法以及推荐用于分配需求的方法。

第6章提供了关于成本与进度计划的指导。计划从商业计划开始，但从那之后，计划就是关于团队如何能够交付用户所要求的全部特征和功能。主计划模式是一个计划波形，然后分为一个个时间箱开发周期。<sup>15</sup>

第7章解释了如何用数学术语来估计成本和进度。成本是团队所有人力投入的总和，但是总人力投入是依赖于进度的：需求多快能被转化为已完成的产品。成本和进度依赖于整个敏捷团队的产出。速率是产出的通用度量方法。速率通过测量燃尽率来度量项目的产出——完成产品增量的节奏。<sup>16</sup>

第8章是关于团队的，团队是敏捷方法的组织模型的核心。每种方法在如何组织团队上都略有不同，但其思想是一致的：员工以小团队的形式协作以取得更好的协同效果，最终总的结果对于涉及的各方来说都是双赢。

第9章讨论治理。治理如果能与常识一起应用，就是好事。它为干系人创造

了演化范围和交付时间线的机会。治理提供对资源的承诺，并为项目顺利进行提供稳定的基础。

第 10 章描述挣值及对结果的管理。挣值意味着满足客户、投资取得回报并建立起收益流。价值的跟踪不能给项目经理或团队带来太大的开销。本章中我们将看一看那些在实际应用中不仅有效而且高效的实践。

第 11 章提供了关于扩大规模以及使合同和外包协议成为一部分项目活动的观点。敏捷方法是为自我组织的小团队而设计的。调整敏捷实践使之适应企业级别，这是一个挑战，也正是本章所要讨论的。

第 12 章讨论收益。如果项目不能为组织、干系人或客户带来可持续的收益，项目就不值得做。收益大部分在项目结束以后才能产生，但是敏捷方法引进及早交付的概念，使得在项目进展中获得收益成为可能，甚至有可能使项目变得自给自足。

附录 A 包含本书所讨论的四种敏捷方法的细节。附录 B 是具有特殊意义的术语的词汇表。

## 前言尾注

---

1. Dilbert™ 插图于 2008 年 12 月 21 日发表于 Techcrunchit.com (<http://www.techcrunchit.com/2008/12/21/will-this-economy-finally-push-the-toyota-way-into-software-development/>) 检索于 2009 年 7 月。

2. 在本书中，产品基线、产品、系统、交付物及结果可互用，皆用于表示项目结束时用户或客户所拥有的或使用的。适用于敏捷方法的项目是软件密集型的，但可能也有许多复杂的硬件组件。项目可能只产生支持软件的过程，也可能产生内部使用的系统或应用程序，或者是为企业或消费者所使用的系统或产品。项目也可能对已安装的系统或产品——本书称遗留系统或产品——进行或大或小的修改。

3. XP 是极限编程 (Extreme Programming) 的缩写；EVO 是指汤姆·吉尔 (Tom Gilb) 倡导的演化方法；RAD 是快速应用开发 (Rapid Application

Development), 后来发展为 DSDM。

4. 团队软件过程 (Team Software Process, TSP) 和个人软件过程 (Personal Software Process, PSP), 是卡耐基梅隆大学的服务标记。

5. 本书中, RUP 指统一软件开发过程 (Rational Unified Process), 是 IBM/Rational 的产品; RAD 是快速应用开发, JAD 是联合应用开发 (Joint Application Development)。

6. 瀑布是一种序列化的项目计划, 这种计划大致由收集需求、设计解决方案、开发与测试解决方案然后交付结果等步骤组成。瀑布得名于它的图表里的一系列层叠的步骤看起来像瀑布。在 20 世纪 70 年代, 为了改进瀑布序列, 才加入了回到前一步骤的迭代。

7. UML 是统一建模语言 (Unified Modeling Language), 用于画图表示和分析需求, 特别是在包含人机交互的系统中。CASE 是计算机辅助的软件工程 (Computer Aided Software Engineering) 的缩写。它指一系列用于帮助软件设计、开发和维护的工具。

8. PMBOK 是项目管理协会 (Project Management Institute) 的商标, 它在美国和其他国家注册。

9. 六西格玛是一种定义规范的控制方法, 它包含了一个由多个步骤组成的确定问题的实践, 还有一个缺陷控制标准, 该标准要求 100 万个机会中在控制边界之外的缺陷必须少于 3.4 个。而实际的控制边界是由分析和历史度量数据来决定的。

10. StandishGroup 自 1994 年起定期在《Chaos 报告》上发表报告。(参见 StandishGroup.com。)

11. 参见《SDTimes》在线于 2007 年 3 月 1 日对 2007 年报告的评论——Rubinstein, D. “Standish Group Report: There’s Less Development Chaos” *SDTimes*, March 2007, <http://www.sdtimes.com/content/article.aspx?ArticleD;eq30247>, 检索于 2009 年 7 月。

12. 业界的一些人士看了 StandishGroup 的首页及报告, 认为它的结果用其他调查并不可重复。但是, 1994 年的结果被广泛引用, 并作为事实被收录进词典。

13. 参见《SDTimes》在线于 2009 年 4 月的报告 [http://www.standishgroup.com/newsroom/chaos\\_2009.php](http://www.standishgroup.com/newsroom/chaos_2009.php), 检索于 2009 年 6 月。

14. 关于敏捷项目跟踪记录的度量信息, 参见 Boehm, B. 和 Turner, R. 于 2004 年由波士顿 Addison-Wesley 出版的《Balancing Agility and Discipline》中的附录 E “经验信息”。

15. 本书的许多部分都会详细地介绍时间箱概念。简单地说, 时间箱是预先为某一活动计划好的期间, 它定义的时间约束适合所有人。在时间箱的结尾, 工作要么完成, 要么没有完成——不会有部分完成。迭代指项目中的开发步骤。在敏捷方法中迭代是按照时间箱来划分的。每次迭代都在时间箱的时间限制内实现一部分特征和功能。

16. 敏捷方法使用一些新的术语来表示旧的概念。速率这一术语在敏捷方法中用于表示产出。它度量团队在一次迭代期间生产了多少产品。燃尽 (burn-down) 则是 Scrum 中的一个挣值概念 (在 XP 中则是 burn-up, 概念基本相同)。当对象交付到产品中时, 它就从未结订单中除去。随着团队不断完成清单中的对象, 到最后未结订单就是空的了。燃烧指的是人力投入。当人力投入到每个对象上, 最终对象就被完全“燃尽”而可以发布到产品中了。

---

# 目 录

---

|                           |    |
|---------------------------|----|
| 第 1 章 快速阅读 .....          | 1  |
| 简短的历史回顾 .....             | 2  |
| 敏捷宣言与敏捷原则树立了敏捷方法 .....    | 4  |
| 项目开发生命周期涵盖商业计划到商业交付 ..... | 6  |
| 便于读者理解的一些术语 .....         | 11 |
| 计划驱动的经验教训 .....           | 12 |
| 四种具有代表性的敏捷方法 .....        | 18 |
| 螺旋方法降低风险 .....            | 25 |
| 总结与要点 .....               | 30 |
| 本章尾注 .....                | 31 |
| 表格注释 .....                | 32 |
| 第 2 章 敏捷的商业计划 .....       | 34 |
| 商业计划增加项目的价值 .....         | 34 |
| 商业价值模型支撑商业计划 .....        | 39 |
| 项目平衡表有助于与商业决策者进行沟通 .....  | 45 |
| 敏捷商业计划按照层次来创建 .....       | 49 |
| 总结与要点 .....               | 57 |
| 本章尾注 .....                | 57 |

## XIV 敏捷项目管理：企业级实践与案例

|                              |     |
|------------------------------|-----|
| <b>第3章 敏捷领域的质量</b>           | 59  |
| 质量是在价值观、原则和实践的基础上创建的         | 60  |
| 思想领袖树立敏捷质量价值观和原则             | 62  |
| 敏捷方法计划了质量的价值观和原则             | 70  |
| 总结与要点                        | 77  |
| 本章尾注                         | 77  |
| 表格注释                         | 78  |
| <b>第4章 管理测试</b>              | 79  |
| 原则和实践指导质量测试                  | 79  |
| 测试驱动的开发是起点                   | 81  |
| 测试计划对好的测试度量必不可少              | 87  |
| 抽样测试节省时间和金钱                  | 96  |
| 测试假设建立信心                     | 100 |
| 总结与要点                        | 102 |
| 本章尾注                         | 103 |
| <b>第5章 开发范围和需求</b>           | 105 |
| 最有能力提供的范围是最佳价值               | 106 |
| 愿景是起点                        | 108 |
| 敏捷项目团队与客户的合作伙伴关系建立在权利与责任的基础上 | 110 |
| 需求的过程                        | 112 |
| 团队使用故事、模型和原型                 | 117 |
| WBS 是组织范围的工具                 | 121 |
| 用计划范围来管理范围变化                 | 126 |
| 总结与要点                        | 130 |
| 本章尾注                         | 130 |
| 表格注释                         | 133 |

|              |                   |            |
|--------------|-------------------|------------|
| <b>第 6 章</b> | <b>计划成本和进度</b>    | <b>134</b> |
|              | 既然是敏捷，为什么还要计划     | 135        |
|              | 为敏捷项目制定敏捷的计划      | 141        |
|              | 时间箱是敏捷方法进度计划的基本单元 | 149        |
|              | 总结与要点             | 156        |
|              | 本章尾注              | 157        |
| <b>第 7 章</b> | <b>估计成本与进度</b>    | <b>159</b> |
|              | 估计的特点影响可预测性       | 160        |
|              | 范围、复杂性和速率都影响估计    | 168        |
|              | 总结与要点             | 178        |
|              | 本章尾注              | 179        |
|              | 表格注释              | 180        |
| <b>第 8 章</b> | <b>团队就是一切</b>     | <b>181</b> |
|              | 团体是从社会单元中形成的      | 182        |
|              | 团队是敏捷项目的基本单元      | 189        |
|              | 有的团队成功，有的团队失败     | 200        |
|              | 矩阵管理敏捷领域的资源       | 207        |
|              | 敏捷团队自己招募成员        | 210        |
|              | 总结与要点             | 211        |
|              | 本章尾注              | 212        |
|              | 表格注释              | 214        |
| <b>第 9 章</b> | <b>治理</b>         | <b>215</b> |
|              | 治理建立在质量原则上        | 216        |
|              | 治理核实符合性           | 225        |
|              | 总结与要点             | 227        |
|              | 本章尾注              | 228        |
|              | 表格注释              | 229        |

|                  |     |
|------------------|-----|
| 第 10 章 挣值        | 230 |
| 价值的定义            | 231 |
| 价值的核算            | 232 |
| 挣值收回投资           | 234 |
| 每次敏捷交付都获得价值      | 241 |
| 每次发布的计划—执行—检查—处理 | 247 |
| 总结与要点            | 252 |
| 本章尾注             | 253 |
| 第 11 章 扩大规模和合同   | 254 |
| 规模会放大每一个问题       | 254 |
| 网络使得规模扩大         | 259 |
| 虚拟团队可以扩大产出       | 262 |
| 敏捷通过合同来扩大规模      | 265 |
| 总结与要点            | 273 |
| 本章尾注             | 274 |
| 表格注释             | 275 |
| 第 12 章 收益的实现     | 276 |
| 收益是计划的一部分        | 276 |
| 结果的度量推动改进        | 282 |
| 总结与要点            | 288 |
| 本章尾注             | 288 |
| 附录 A 方法论         | 289 |
| 附录 B 词汇表         | 314 |