软件工程研究院

可伸缩敏捷开发：企业级最佳实践

Scaling Software Agility:Best Practices for Large Enterprises

(美) Dean Leffingwell 著
李冬冬 冯 雁 娄嘉鹏 译
飞思科技产品研发中心 监制

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

软件工程研究院

可伸缩敏捷开发：企业级最佳实践

(美) Dean Leffingwell 著
李冬冬 冯 雁 娄嘉鹏 译
飞思科技产品研发中心 监制

Scaling Software Agility: Best Practices for Large Enterprises

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

版权

版权贸易合同登记号 图字：01-2009-0647

图书在版编目（CIP）数据

可伸缩敏捷开发：企业级最佳实践 /（美）兰芬维奥（Leffingwell,D.）著；李冬冬，冯雁，娄嘉鹏译.—北京：电子工业出版社，2009.5

（软件工程研究院）

书名原文：Scaling Software Agility: Best Practices for Large Enterprises

ISBN 978-7-121-08216-

I. 可… II.①兰…②李…③冯…④娄… III.企业管理—应用软件—软件开发 IV.F270.7

中国版本图书馆 CIP 数据核字（2009）第 013186 号

责任编辑：宋兆武　吴亚芬

印　　刷：北京机工印刷厂

装　　订：三河市鹏成印业有限公司

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：20.5　字数：365 千字

印　　次：2009 年 5 月第 1 次印刷

印　　数：3 500 册　　定价：49.00 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 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

如果你刚刚涉入敏捷方法的领域，那么打开这本书时你可能会充满忧虑和怀疑，这不足为怪。关于敏捷方法似乎有一些奇怪的表述，例如，冲刺（sprint）、速度（velocity）、并列争球（scrum）（这是橄榄球比赛吗？）、极限编程（extreme programming）（我们是穿着滑雪板跳过悬崖吗？）、用户故事（user stories）、事迹和传奇（epics and sagas）（这是作家工作室吗？）等，还有一些怪异的社交形式，例如，结对编程（pair programming）、用户审查（user retrospectives）、聚集或者每日站立例会（huddles or daily stand-up meetings）（开发人员和测试人员在工作中互相拥抱吗？）等。敏捷团队似乎也消耗大量的彩色便签和 4×6 索引卡，他们在可以触及的任何墙面贴满了这些东西，整个事情似乎都不太对劲。这一切似乎是 Scott Adams 的 Dilbert 漫画的好素材！但是请不要误会：敏捷软件开发过程就是“穿越鸿沟”，这里用到了 Jeffrey Moore 创造的一个术语。我们今天已经远远不再是说些搞笑且令人讨厌的行话了，而是形成了有效的、有生产力的并且可扩展的开发方法。你必须向前迈进，否则就要落后。

另外，你可能看到或听说过，敏捷开发通常是反对“计划驱动”开发的，你可能认为这是一件非常混乱或者无法无天的事情，比起既系统又有计划的行动倒退了一大步。但是实际上，敏捷项目是经过精心策划的，只是他们的计划不同，并且该计划的修订和完善更为频繁。可能也有人告诉过你，敏捷方法能够很好地支持小型团队（7~12 人）、短期项目（2~9 个月），但是它不能适用于大型的、长期的和分布在全球各地的软件开发项目。然而，请你不要合上这本书。随着世界各地的众多项目对这些界限的推动，以及敏捷在软件成果的高生产力和高质量方面取得的成功，一切都在快速地变化。

这就是 Dean Leffingwell 在本书中的重要贡献，他以 XP、Scrum、Lean、DSDM、FDD、Unified Process 等不同的敏捷过程之间的争论为基础，找到了这些方法之间的共性并作为基准，然后才进入他的主要目标，说明如何扩展这些敏捷方法并使其超越当前的适用范围。他不是在本已很长的名册中补充一个新的敏捷过程方法，相反，他利用一套新的实践方法扩展了敏捷方法，并把这些包括技术和管理在内的更高层次的实践方法融合集成到现有的已经建立的敏捷实践方法（名字很有趣）中。除了综合并且扩充了敏捷方法中所共有的最佳工程实践方法之外，他还描述了用于大型敏捷项目管理的方法：发布计划等主题，协调大规模的分布式团队，建立项目的企业价值观，处理大型的、长生命周期的开发过程，等等，我这里仅举这几例。

作者的工作不是学术性的，他不只是提出一些新的、大胆的猜测让你去尝试。他的建议根植于他多年积极的自身实践，这些实践来自于许多公司的众多项目，

所涉及的行业范围极为广泛，从维持生命的医疗设备到软件工具，从游乐园骑乘设备到大型 IT 基础设施的应用。我知道这第一手资料是因为我曾有幸先后在 Rational Software 公司和 Rally Software Development 公司与 Dean 一起工作了大约 10 年。

我再补充一句我自己的建议。看完这本书后（可能有很多人都在寻求好的方法），不要期望像食谱中的配方一样能够立即使用并且肯定成功。房地产商的座右铭是“位置，位置，位置”，我想说，采用新的软件开发实践方法的关键是“背景，背景，背景”。了解你的背景，将其与 Leffingwell 所介绍的背景相比；了解相关的目标、项目范围和规模，以及文化和改进业务的合约。然后，调整、适应和改变 Dean 的建议以适合你的需要。背景包括很多方面：你的领域（行业的类型），待开发系统的规模，团队的历史，他们自己的个人和集体经验或教育，所使用的技术，系统的危险程度，甚至企业和国家的文化。Dean 介绍给你的东西是在他自己的背景基础上形成的，他曾有作为一个企业家、软件开发经理、主管、方法学家及顾问的经历，从这些经历中他形成了自己的一套价值观和规范。你的背景会有所不同。如果有任何疑问，就回到基本的原则，即敏捷的基本原则，包括本书所描述的那些原则，而不能被束缚在具体的术语和战术中。

我们一直在说“敏捷过程”，但是实际上，不论是正式的还是在人们头脑中所默许的描述，过程本身都不是敏捷的。人可以敏捷；团队或者组织可以敏捷。你是“敏捷的”，但是你不能“做敏捷”。如果你只是做敏捷而不是敏捷地去做，那么你就会失败，并且不会知道为什么失败，接下来你可能会将失败归咎于一些书。因此，你要跨越这个鸿沟，就不仅是决定采用“指定的”敏捷方法和/或 Dean 等人所描述的这样或那样的实践方法。这需要改变观念或态度，这不是很容易做的事。仅仅读一本书或者参加一个学习班是不够的。你和你的组织想要把敏捷原则变成自己的思想方式，并获得真正敏捷性的好处，你必须要尝试这些新的实践，并且是一次又一次地实践。然后，总结你自己的经验并进行反思，回头看看你已得到（或者没有得到）什么，继而再进一步调整和适应以及采用敏捷的原则。换句话说，就是不要相信一本书能够使你敏捷。它需要的不仅仅是去做，还需要思考、感觉和变化。

然而，如果你是敏捷领域里的一位老资格的人，如果在 20 世纪 70 年代你就已经了解敏捷了，那么这本书有什么给你看的呢？有很多东西值得你看。我个人学到了很多东西，并开始反思自己的经验和实践。我同意 Dean 写的所有内容吗？不。我的背景和我的经验与他是有些不同的，但是如果不去深究具体的战术或者词汇的话，Dean 和我在大多数问题上最终都是一致的，特别是在原则和总体战略

方面。（此外，他最近指出，我们似乎很享受辩论的过程和辩论本身，辩论可以作为我们明确自身思想的有效渠道，也可以作为一种了解他人想法的交流工具。）有了这一观点，我猜想，即使是最有经验的“敏捷人员”也会学到一些宝贵的经验教训。

说永远不如做。从前言开始，翻开这本书，开始阅读吧。欣赏和学习敏捷吧！

Philippe Kruchten, Ph.D., P.Eng.

软件工程的教授，英属哥伦比亚大学（University of British Columbia），温哥华，加拿大

软件方法论生涯的第 1 阶段：在 RELA 和 Requisite 公司的经历

在我的职业生涯中，我一直致力于改进软件工程及软件开发管理实践方法。即使这是一个需要持续努力的目标，但是现在我根据我对软件开发实践方法的理解，将我的经历总结为 3 个不同的阶段。在第 1 个阶段，我是 RELA 公司的首席执行官，在这个公司我们与他人签订合同为其开发软件。RELA 开发各种各样的软件应用程序，从令人反胃的冒险乐园骑乘设备到维持生命的医疗设备。因为总是为他人编写软件，我们逐渐意识到需要“建立正确的东西”。个人的生计、我们的公司及其他利益相关者都依赖于我们理解方案需要解决什么问题的能力，以及在实现方案时怎样应用有效的最佳实践方法的能力。

为了做到这一点，那个时候我们使用严格的基于瀑布模型的实践方法。事实上，我们的一些客户及 FDA（美国食品和药物管理局）等一些主要监管机构指定使用这种方法，因此，我们遵照要求使用这种方法，并且曾试图对其进行改进。虽然现在我们中的很多人以批评和取笑我们曾使用的方法为乐，但事实是瀑布方法比起过去的试验性方法是一个极大的进步。更重要的是，使用这种方法能够交付成果。当时，我主要关注需求过程，因为在这个过程中，要进行重要的分析，要定义解决方案并且要作为合同的基础。

这段经历促使我进入了下一个职业，担任 Requisite 公司的创始人和总裁，并开发了产品需求管理的解决方案 RequisitePro。在 Requisite 公司，我们提出并且开发了需求实践方法和产品，因此，在一定意义上成为软件生命周期前端的专家。在 1997 年我们将 Requisite 出售给了 Rational 公司，接着我开始了软件开发过程职业生涯的第 2 个阶段。

软件方法论生涯的第 2 阶段：在 Rational 软件公司的经历

在这个阶段，我是 Rational 软件公司的一名高级行政人员，参与了统一建模语言（Unified Modeling Language，UML ）和 Rational 统一过程（Rational Unified Process，RUP）的颁布。在 Rational 软件公司，我有幸与 Grady Booch、Ivar Jacobson、James Rumbaugh、Walker Royce 和 Philippe Kruchten 等软件思想领袖一起工作。在这段时间内，Don Widrig 和我也出版了 Addison-Wesley 教材《管理软件需求》的第 1 版（2000）。

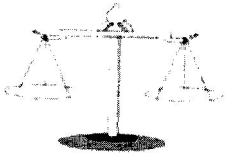
接着，我们开始考虑基于面向对象技术，这项技术为我们的开发方法提供了更多的灵活性，为我们所编写的软件提供了更多的伸缩性。同时也带来了一种新的软件开发过程，这个过程完全不同于瀑布模型，它的特点是迭代和增量。在此方法中，每一次迭代是编写一段可以被客观地估量和评价的代码。这种方法远比我以前使用的方法敏捷：我们不必依靠文件和设计审查等类的中间产品，就可以看到和衡量实际工作的进展。

Rational 公司在一份书面过程描述中，将这个过程命名为 Rational 统一过程（Rational Unified Process），之后，它的销售和应用在整个行业内获得了成功。此外，在需要 4 个国家多达 800 人的团队成员合作的项目开发与发布中，我们也应用了这个过程。Rational Suites 每年发布两次，每一次都是一套集成的产品和一个共同的安装程序。Rational 最终被 IBM 公司购买，今天 RUP 的销售由 IBM 的 Rational 软件事业部（Rational Software Division）负责，数十万的从业人员都在使用它。

软件方法论生涯的第 3 阶段：使用敏捷和在 Rally 公司的经历

离开 Rational 公司之后，我成为发展阶段软件企业的独立顾问和指导，我指导 6 个新企业的经营策略和软件开发实践。我利用这个机会应用一些更为创新的、轻量级的方法，包括 XP 和 Scrum ，并亲眼目睹了这些方法给小规模团队所带来的工作效率和质量的改进。

经过一段短暂的时间，这些方法征服了我，因此，很快我开始拒绝参加不理解敏捷方法的任何企业或者团队。否则企业的风险太大了！在同一时间，我开始觉察到这些方法的局限性。随着团队和应用程序的扩大，团队重构代码变得不切实际，并且我们也注意到，需要更多地保证需求的沟通。在这个时候，我还是 Rally 软件公司的专家顾问，帮助发展其分布式敏捷开发的安置解决方案。在 Rally 软件公司，与 Ryan Martens、Ken Schwaber、Jim Highsmith、Mike Cohn、Tom、Mary Poppendieck 和 Jeff Sutherland 等敏捷思想领导者的交流对我产生了重大影响。



在企业级规模应用敏捷的经历

此时，我遇到了在一些大型组织中应用敏捷方法的挑战。我很不安地接受了这项任务，并在接下来的几年内，将敏捷的核心原则应用到大型组织中，同时也应用了我在 BMC 软件公司的大规模开发的经验。在 BMC

软件公司，我们曾为交付超大规模的新应用程序与数以百计的高度分布的开发人员一起工作。

在这个过程中，我很高兴地发现敏捷方法所提供的很多最佳实践方法给企业带来了立竿见影的价值。同时，我也发现这些最佳实践方法并不能完全解决企业规模的挑战。因此，我们逐渐形成了获得更好的敏捷所必需的一套扩展实践方法。当我发现市场上几乎没有出版物可供大型公司阅读参考时，我下定决心写这本书。我这样做是希望你的企业可以吸取我们的经验，并应用这些经验给客户提供更高的生产力和更好的品质。在软件占主导地位的世界中，很难想象我们的行业是在一个较高的支点上，确实，我们的经济是一个整体。

如何阅读本书

第 1 部分：软件敏捷概述

本书共分为 3 部分。第 1 部分介绍了敏捷运动的简短历史，讨论了一些目前使用的主要敏捷方法，包括 XP 和 Scrum，也讨论了应用在敏捷方式中的 RUP，RUP 是一种迭代和增量方法。此外，我们也简要地介绍了其他一些促进敏捷运动的方法，包括精益软件开发（Lean Software Development）、动态系统开发方法（Dynamic System Development Method，DSDM），以及特性驱动开发（Feature-Driven Development，FDD）。我们介绍这些方法并不是为了教授这些方法本身，而是为理解第 2 部分和第 3 部分的内容打下基础。你会发现，每一种方法都为软件开发实践带来了大量的新思路，每一种方法都为技术发展做出了重大贡献。此外，你也将看到形成了一套最佳敏捷实践方法，其中的许多方法都已经得到大规模应用，我们将利用这些作为建立企业敏捷性的基础。

第 2 部分：7 种可伸缩的敏捷团队实践

第 2 部分介绍了可伸缩的 7 种敏捷团队实践方法，每章介绍一种。从一定意义上讲，这些实践方法可以被视为敏捷的本质，因为所有的敏捷方法都明确或含蓄地应用了这些实践。对于那些刚刚涉入敏捷方法的人或者意图实施这些实践的大机构来说，本书的第 2 部分应该是些安慰，因为通过理解所描述的一些敏捷方法，或者更进一步，基于公司的背景进行一些必要的融合和匹配，这些最佳实践方法很自然就出现了并为在任何范围内的实际应用提供了益处。这些方法不是微不足道的，它们的作用已经在各种各样的项目背景中得到了证实，采用这些方法的所有团队都将受益。

本书第 1 部分和第 2 部分介绍了软件敏捷的概要，并描述了可以应用在任何

规模的 7 种最佳实践方法，每个实践方法都可以直接并且立即促进采用此方法的团队的生产力和成果的质量。

第 3 部分：创建敏捷企业

然而，要真正实现企业级敏捷还有更多的工作要做，这就是第 3 部分的内容。我们描述了另一套能力、指导方针、原则、实践和见解，这将使该组织可以在几乎任何规模的应用程序或系统中应用敏捷方法。这些实践方法都来自于在大型环境中应用敏捷方法的经验。它们包括：分布在多个国家有 40~50 个开发人员的小型团队的“零起点”项目，其中包括广泛的外包及多达 1000 人的大型组织，所有开发人员在要求这些团队高度协作的系统中为达到共同目标而一起工作。第 3 部分中的一些原则似乎是显而易见的，也有一些更精细的原则是在大型环境中应用敏捷方法的经历中总结出来的。许多原则都是团队在对其前期的努力进行反省时提出来的，然后这些原则随着时间推移而调整其行为以不断地改进结果。

总之，我们希望本书能够帮助大型组织，使其和应用敏捷方法的小型团队一样获得 200%的生产力和质量。然后，这些结果将带来更快的产品上市时间、较高的开发投资回报率，以及提高客户对企业的满意度等好处。并且，我们不要忘记，组织机构沿着这条道路前进还有其他一些无形的好处：团队本身对敏捷方法的热爱推动他们去实践并改进他们的方法，这样就形成了良性循环，充满活力——不断地改进过程——促进项目成果——个人和专业的成长——更高的工作满意度。在产业面临为世界上大部分知识产权编码的挑战时，有什么能比敏捷更有效力呢！

有很多人对本书的出版做出了贡献，在此我无法一一提及，但我还是想特别地感谢一些对本书有直接贡献的人。首先，我想感谢 Rally 软件公司的创立者和 CTO，Ryan Martens，他教给我很多有关敏捷方法的东西，并提供了很多相关的概念。具体地说，Ryan 帮助我准备了第 7 章，敏捷的本质，正是这一章最清晰地描述了为什么敏捷是如此出众并且强大的。Rally 公司中的 Richard Leavitt、Shai Koenig、Tom 和 Mary Poppendieck、Randy Stafford、Dave Muirhead 及 Philippe Kruchten 等人也给予了帮助。此外，Rally 公司允许我将一些白皮书中的内容写入书中，为此，我深为感谢。

我也想感谢 Pete Behrens、Bob Cotton 和 Bill Wood，他们协助我在其公司将理论转化为实践，并且审阅了本书中的各个引用。也要感谢 Grady Booch 审阅了第 16 章，有意识的架构。

特别要感谢 Ken Schwaber，感谢他领导了 Scrum 的开发、应用及推广；也感谢他对敏捷方法所要求的新组织和动态管理的理解。

我也想感谢 Addison-Wesley 的主编 Chris Guzikowski，感谢他对项目的支持；还要感谢 Addison-Wesley 的评审，Robert Bogetti、Susan Burk 和 Carol A. Wellington 及制作助理 Kim Arney、Carol Lallier、Diane Freed 和 Richard Evans，没有他们我不能写成这本有价值的书。

我还要感谢 Philippe Kruchten，感谢他审阅了本书及对第 16 章所做的贡献；感谢他非常独立的观点和在过去的 20 年中对软件开发实践方法的巨大贡献，感谢他为本书做序。

最后，特别要感谢 BMC 软件集团公司分布式系统管理副总裁 Israel Gat，他委托我及其他人帮助他的团队快速地转向大型的敏捷开发模式；感谢 BMC 软件公司的 Paul Beavers、Becky Strauss、Roy Ritthaler、Walter Bodwell 和 Mike Lunt 等同事，他们帮助开发及应用了这些方法；感谢敏捷方法指导 Michelle Sliger，Jean Tabaka 和 Stacia Broderick。他们的经验和专业知识为本书所做的贡献是不可估量的。

Dean Leffingwell 是一位知名的软件开发方法论者和作者，也是一个软件团队指导，他用自己的经历帮助软件开发团队实现他们的目标。他是 Requisite 公司的创始人和前 CEO，是 RequisitePro 的创造者，也是 Rational 公司的前副总裁并在 Rational 公司负责 RUP 商业化。在过去的五年里，他的工作角色是独立顾问，并担任 Rally 软件公司的顾问兼方法论者。Leffingwell 先生致力于将敏捷方法应用于跨国公司分布式大型开发团队，他以在此过程中获得的经验为基础，写成了此书。Leffingwell 先生也是《软件需求管理：用例方法（第 2 版）》（Addison-Wesley 公司，2003）的第一作者。

第 1 部分　软件敏捷概述

Contents

第 2 部分 7 种可伸缩的敏捷团队实践