



PEARSON



软件开发 经济学

The Economics of Iterative
Software Development

Steering Toward Better
Business Results

(美) Walker Royce Kurt Bittner Mike Perrow 著
苏敬凯 金振林 译

TP311.52



机械工业出版社
China Machine Press



软件开发 经济学



机械工业出版社
China Machine Press

本书讲的是如何管理软件开发项目,从经济学的角度介绍了如何有效地管理有限的资源以达到最优结果,描述了在迭代式的项目中频繁修正航向的好处,以及如何统筹兼顾和度量中间结果。

本书第一部分讲述在当今商业气候下软件开发和管理的情景、成功的难点以及失败的结果;第二部分介绍以迭代式开发原则和实践为基础的软件工程的现代方法;第三部分更详细地讲解了如何通过度量来确定所采用技术的优劣。本书适合商业组织中的各级领导人以及任何希望成功管理软件项目的人员阅读。

Simplified Chinese edition copyright © 2010 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *The Economics of Iterative Software Development: Steering Toward Better Business Results* (ISBN 978-0-321-50935-2) by Walker Royce, Kurt Bittner, Mike Perrow. Copyright © 2009.

All rights reserved.

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

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。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2009-7322

图书在版编目(CIP)数据

软件开发经济学/(美)罗伊斯(Royce, W.)等著;苏敬凯,金振林译.——北京:机械工业出版社,2010.3

书名原文: *The Economics of Iterative Software Development: Steering Toward Better Business Results*

ISBN 978-7-111-30146-2

I. 软… II. ①罗… ②苏… ③金… III. 软件开发-项目管理 IV. TP311.52

中国版本图书馆CIP数据核字(2010)第048490号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:刘立卿

三河市明辉印装有限公司印刷

2010年4月第1版第1次印刷

170mm×242mm·11.5印张

标准书号:ISBN 978-7-111-30146-2

定价:29.80元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010)88378991;88361066

购书热线:(010)68326294;88379649;68995259

投稿热线:(010)88379604

读者信箱:hzjsj@hzbook.com

前 言

假设你是一位富有并且经验丰富的旅行家，正在计划一个为期一个月的多国假期。听起来不错吧！接下来，你要选择一个大陆，世界上总有些你想要去探索的地方。你会从哪里开始呢？如果你实际地考虑一下这个打算，可能很快就能决定自己这个假期的起点和终点。你会想象对某些著名城市、名山大川及海滨胜地的一连串的小旅行。但是，当你开始确定这次旅行的总体路线时，你会开始计划每一顿饭、每一次晚间散步和每一次购物吗？当然不会。没有人能预先准确地计划在那些陌生的地方做些什么，或者一路上如何使用时间和资源。另外，你也知道，如果你执著于出发前设计好的那份严格的计划，那么游览的质量肯定会受到损害。

本书讲的是如何管理软件开发项目，这确实很难与悠长的假期相提并论。但从管理的角度来看，两者有很多共同点，这些共同点都与不熟悉有关——不熟悉的地区、语言、个人行为、习惯、成本和基础设施。本书所基于的经验涉及那些软件开发团队在每个项目中都会遇到这些不确定性的遭遇。那些最成功的团队会先在高层次上对自己的项目进行计划，然后（就像那些经验丰富的旅行者一样），他们会随着旅程的进展而以一些较小的步骤进行计划，这就是迭代。

对于大多数对改进业务结果感兴趣的经理们，我们将尽量使用一些他们能理解的词汇来解释这些成功。考虑到这种情况，对于这部分读者而言，我们没有假设他们有大量的技术经验，但是我们必须假设他们承诺要成为成功的领导者。本书的目标读者是那些在某个商业组织中处于不同领导职位的人，特别是那些要采购、管理或开发软件并以此作为业务策略一部分的组织。我们的目标是描述在迭代式的项目中频繁修正航向的好处，如何度量那些中间结果，以及统筹兼顾的方法是如何对底线做出有意义的贡献的。

最后这一点使得所有的事情都与本书的根本主题有关，那就是：经济学。从最广泛的意义上来讲，良好经济性意味着有效地管理有限的资源以获得最优结果。软件经济学同样以这些原则为基础。我们花了些时间来探索差的经济学——以老式的管理风格为基础的，包括那些把软件项目当作传统工程项目（例如造桥）来管理所发生的低效率。当到了软件构造阶段，这些低效率在时间、预算和在市场上失去的机会方面都代价高昂。相反，现代的迭代式开发方法能够以实用性地管理团队的有限资源为基础来改进这一结果。因此就有了本书的书名。

本书安排章节的顺序很显而易见。第一部分“软件驱动的经济”讲述在今天这样的商业气候下的软件开发和管理的情景、成功的难点以及失败的结果。第二部分“提高软件开发的经济效益”关注的是以迭代式开发的原则和实践为基础的软件工程的现代方法。当然，我们并非无所不知。但是，通过几十年对于“我们知道什么不起作

用”的观察，我们对于“什么能起作用”也了解了很多。第三部分“软件工程的实用度量”更详细地讲解了如何通过度量来确定这些技术能够起作用。正如“你不能管理你不能度量的东西”这句更新过的箴言那样，本书的最后一部分关注度量的目的，包括可以用怎样的方式在整个项目周期中降低差异。

无论你是一位正在寻找迭代式开发原则概述的经验丰富的软件项目经理，还是一个正在寻找这些概念的入门知识的新手，你都会发现本书的价值。

Mike Perrow
Medford, Massachusetts

目 录

前 言

第一部分 软件驱动的经济

第 1 章 软件项目管理的挑战	3
1.1 软件开发中的高风险	4
1.2 制度上的关注	5
1.3 传统的软件项目管理	8
1.4 瀑布方式的问题	9
1.5 小结	12
第 2 章 达成结果：软件经济学的案例	15
2.1 迭代式开发	17
2.2 基于结果的方式的益处	19
2.3 成功的标志	20
2.4 小结	21

第二部分 提高软件开发的经济效益

第 3 章 软件经济学的趋势	25
3.1 软件经济学的简化模型	26
3.2 软件工程：40 年历史回顾	28
3.3 改进的关键：平衡的方法	30

3.4 小结	33
第4章 降低软件项目的规模或者复杂度	35
4.1 管理范畴	35
4.2 降低人工编写的代码的规模	37
4.3 使用面向服务的架构增强复用	38
4.4 小结	39
第5章 改进开发的过程	41
5.1 项目过程	41
5.2 使用迭代式的过程	42
5.3 及早应对重大风险	45
5.4 增量式地改进实践以达到目标	48
5.5 小结	50
第6章 提高团队效率	53
6.1 提高个人绩效	54
6.2 改进项目的团队协作	55
6.3 提高组织级能力	57
6.4 小结	59
第7章 通过集成工具来提高自动化	61
7.1 提高个人生产力	62
7.2 减少错误源	63
7.3 使过程改进可能进行	64
7.4 小结	65
第8章 通过常识来加速文化的改变	67
8.1 成功组织的特征	67
8.2 成功的关键	69
8.2.1 低中级管理者是关键的执行者	71

8.2.2	需求、设计和计划要可变并且具体	71
8.2.3	鼓励进行雄心勃勃的演示	72
8.2.4	项目性能的好坏在生命周期的早期更明显	72
8.2.5	早期的迭代应该是不完全的	73
8.2.6	详细完整的工件在早期不太重要，在后期比较重要	74
8.2.7	真正的问题会系统化地出现，也应系统化地加以解决	74
8.2.8	质量保证是每个人的职责，不是一个单独的行当	74
8.2.9	应该把在自动化方面的投资看作是必要的	75
8.3	建议：选择正确的项目、正确的人、正确的目标	75
8.4	小结	76

第三部分 软件工程的实用度量

第9章	实用的软件开发度量观	81
9.1	度量与目标	81
9.2	易变性和目标	82
9.3	度量与迭代开发	83
9.4	度量工作何时会出错	85
9.5	事先制定详细计划错在哪里	87
9.6	在每个阶段分别确定要度量什么	88
9.7	小结	89
第10章	在起始阶段度量什么	91
10.1	评估财务可行性	92
10.2	评估技术可行性和估计总体项目成本	93
10.3	起始阶段中的迭代	95
10.4	其他度量	97
10.5	小结	99

第 11 章 在细化阶段和构造阶段度量什么	103
11.1 细化阶段的度量工作	103
11.2 如何度量风险的降低	106
11.3 如何度量进展情况	108
11.4 构造阶段的度量工作	110
11.5 如何度量 backlog	111
11.6 如何度量测试覆盖率	112
11.7 如何度量构建稳定性	113
11.8 构造阶段的预期进展趋势	114
11.9 小结	115
11.9.1 细化阶段	115
11.9.2 构造阶段	116
第 12 章 在交付阶段度量什么	119
12.1 交付阶段的度量工作	120
12.1.1 度量什么	121
12.1.2 分析趋势和评估测试结果	123
12.1.3 影响可发布性的其他一些需要考虑的事项	123
12.2 如何对交付阶段进行总结	125
12.3 小结	126
第 13 章 在嵌入规划的项目中度量什么	129
13.1 将项目组织起来构成规划	130
13.2 如何度量规划的进阶	133
13.3 小结	134
附录 A 迭代式项目管理起步	137
推荐书目	167

第一部分

软件驱动的经济

软件无处不在，对我们的日常工作来说，软件不可或缺。从微波炉、烤面包机到消费电器、汽车和玩具，我们所使用的所有一切都包含了越来越多的软件。它们在教育中都离不开软件的支持。

我们的医疗保健系统极度地依靠软件来处理账单、病历，而且在某些情况下甚至病人护理也要依靠软件来完成。很多人都在不同程度上依赖因特网来访问信息和完成在线的商务活动。

没有软件系统，大多数行业根本不能运转下去，它们要依靠软件系统来完成业务运转和雇员间的协作。

第1章 软件项目管理的挑战

尽管软件在我们生活中的重要性在不断增长，然而软件项目的难以管理却依旧令人抓狂——进度滞后、项目失败，或者即便是完成也通常不能交付当初想要的东西。对于那些以交付软件解决方案为目的的人来说，这样的失败比比皆是；每个人都有自己的一些项目出岔子的恐怖故事。由于软件在今天的重要性，这种失败会很快导致商业机会的丧失，甚至导致生意的失败。

相比之下，成功的企业则能够理解软件开发能力的重要性，知道它是一种确保项目健康和业务长期生存的手段。但懂得如何管理软件开发的公司却很少，无论他们是为了使自己的优势最大化而运行业务系统的使用者，还是针对商业市场而开发软件的开发者，或者两者兼而有之的企业。真实的情况是，大多数成长中的企业都没有准备好创建这样的软件——这种软件定义了如何跟客户交互或者能帮助他们交付货物或服务。

为什么会这样？原因之一是：大多数行业仍然抱着那些已过时了的 IT 部门的模型不放，把 IT 部门当作是一个提供内部服务的部门，

业务方面的经理们要与之签订“合同”才能得到他们的便利服务。相反，我们应当从一开始就把软件和用来创建软件的过程看作是业务价值链中不可或缺的一部分。真正被集成到公司商业过程中的软件可以改变和提升整个运作，这一点 Amazon.com 和 Apple 这些创新者已经给我们做出了示范。将软件能力集成到业务价值链中的优势日显，所以我们需要改变公司管理与 IT 之间的那种传统关系。

在高层领导的层面上，如果他们能够领悟到改变游戏规则式的创新能带来好处，那么他们就一定会把软件开发能力看成是一种核心的商业竞争力。在 IT 管理的层面上，必须把软件项目理解成是一项能强化公司的竞争优势的价值观念，而不只是一项工程上的成绩。

在这一节中，我们将讨论 IT 失败以认识到软件创新的潜力，以及这种失败的一些典型原因，然后再引入一个软件管理的新概念——用一个植根于软件经济学的更实用的方法来替代传统软件工程学。

1.1 软件开发中的高风险

自 20 世纪 80 年代以来，我们已经走了很长一段路。在那时，我们相信个人计算将会对商业、国防乃至整个社会都产生深远的影响。而刚到 20 世纪 90 年代，计算能力就已发展到了我们无法想象的地步——闪电般的处理速度、充足的内存和实际上无限的存储能力。我们非常容易就能看到这一计算能力所带来的好处，我们尽力对用户接口进行了许多简化，让访问电子邮件、访问因特网、学习新软件

都变得更加简单，这使我们的生活从方方面面都得到了改善。不仅如此，我们还得到了之前无法想象的带宽，尽管至今还有些 Web 用户和部分应用没有充分利用这一带宽。我们正生活在一个通过网线或者移动设备与因特网相连的世界之中，因特网的迅速成长使得众多媒体每个月都会给它一个新定义。

然而，尽管这个产业在发掘计算能力方面取得了如此巨大的进步，我们仍不能完全地利用这些计算能力所带来的好处，除非软件也能取得同样的进步。换句话说，这个世界不会自己实现在数字化方面的潜力，除非软件系统的能力能在今天的计算机硬件和网络条件下提高到一个新水平。开发软件的方式正是问题的所在。

1.2 制度上的关注

我们在软件领域所要面对的困难有些是可以度量的。马萨诸塞州的 Standish Group 所发布的著名 CHAOS 报告一直在测算软件项目的成功/失败比，在 1994 年的第一份报告中，从预算计划、最后期限和功能方面来看，在所有软件项目中只有 16.2% 成功了。尽管在 Standish 的 2007 年的报告中成功率已经提升到 35%^①，但这仍然说明在所有的项目中有近三分之二的项目没有按照预期完成。

在别的工程领域，有资金和管理保障的项目有如此之差的成功率

① 参见 David Rubenstein 的 “Standish Group Report: There's Less Development Chaos Today,” *SD Times*, 2007 年 3 月 1 日 (www.sdtimes.com)。

是难以想象的。可以肯定的是，一些业内人士^①已经对 Standish 所采用的关于构成“失败”的基本假设提出了质疑。作为现代经济中一个充满生机的部分，软件业怎么会充斥着项目的失败呢？答案很复杂，并且涉及所谓工程（engineering）一词的核心思想。后面我们还将讨论这一术语。而现在我们要讨论的突出的事实是人们对于 Standish 年度研究的关注度，因为它强调了软件项目经理们在寻找改进项目结果的方式方面的失败（这一点他们也承认），这种失败比他们之前的希望要差很多。

虽然 Standish Group 指出了失败，但其他的一些专业机构，比如项目管理协会（PMI）、电气及电子工程师协会（IEEE）、软件工程研究所（SEI）以及 IBM 的 Rational 公司，他们也投入了很大一部分资源，用于专门改进软件项目的管理。PMI 的项目管理知识体系（PMBOK）就是一个著名的指引人们处理软件项目中固有的复杂性的指南。长期以来，IEEE 一直被人们看作是在各种工程技术方面都居于全球领导地位的一家专业权威机构，它发行了一些在软件产业中最受人尊重的刊物，这些刊物所关注的是软件项目管理方面的最佳实践。SEI 的能力成熟度管理集成（CMMI）指南已经发展成为度量组织级软件开发能力的黄金标准，特别是在军工行业。IBM 在 Rational Software 的品牌下推广了 Rational 统一过程，以此作为一种可以根据各种规模的软件开发的组织级需要进行裁剪的实践

① 可参见 www.infoq.com/news/Standish-Chaos-Report-Questioned。

框架。

除了这些机构以外，软件业的专业人士们也在频繁地求教于其他行业所公布的一些指南，比如 Six Sigma（发源自 Motorola 公司的一套以降低缺陷为目标的工作规范）和精益制造（丰田公司开创的一种“以减少制造过程中的浪费为目标”的概念）。无论是精益制造还是 Six Sigma，都不是专门为软件产业所设计的，但是这种对于解决软件项目管理难点的执著还是一直激励着业界领袖和从业者们去思考各种通用的为项目管理而编制的指南。

此外，自从 2000 年以来，各种组织和专业联盟已经提出了很多不同的新方法来解决软件项目中天生的难点，比如敏捷联盟、Scrum 联盟、极限编程社区，更别说还有如 OpenUP（the Open Unified Process）那样的各种开源的（非商业性的）项目了。每个组织都推出了不同的开发方法，每种方法都对特定的软件开发团队有所帮助，团队的规模、风格、集体的经验程度决定了其中一种方法更适合自己。

尽管这些专业团体和公司的观点和目标不同，但它们的方法都强调了一个共同的目的，那就是：要克服那些从 20 世纪 60 年代就开始流行的传统软件开发技术的不足。总结起来说，在这些新方法与那些较传统的方法之间，最主要的差别就是：新方法代表的是迭代式的增量开发的实践，而旧方法代表的是瀑布式的开发实践。这在某种程度上似乎是一种粗略的特征划分，但这一区分足够合理，能够达到我们想要的目的。