

从范围管理、时间管理、成本管理、质量管理、风险管理等多角度全面揭示软件项目管理中最容易导致项目失败的10个隐秘假设

以敏捷软件开发方法为主导，通过对一个典型案例的重构，以实践的方式指出了使软件项目获得成功的秘诀

项目管理之殇

为什么你的软件项目会失败

Software Project Secrets *Why Software Projects Fail*

(美) George Stepanek 著 陈宗斌 等译



机械工业出版社
China Machine Press

014022717

TP311.52
380

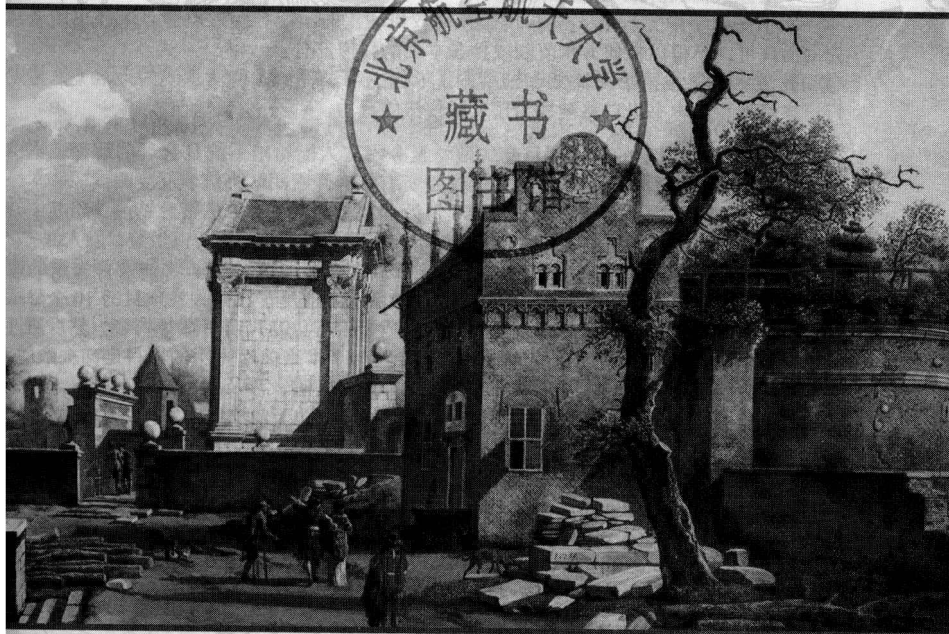
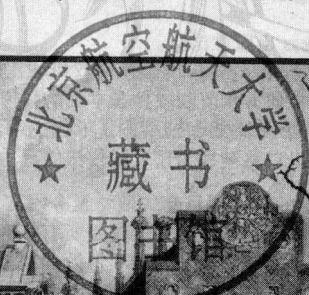
软件工程技术

项目管理之殇

为什么你的软件项目会失败

Software Project Secrets Why Software Projects Fail

(美) George Stepanek 著 陈宗斌 等译



机械工业出版社



机械工业出版社

014022717

TP311.52
380

图书在版编目 (CIP) 数据

项目管理之殇：为什么你的软件项目会失败 / (美) 斯泰潘内克 (Stepanek, G.) 著；陈宗斌等译. —北京：机械工业出版社，2014.1
(软件工程技术丛书)

书名原文：Software Project Secrets: Why Software Projects Fail

ISBN 978-7-111-45287-4

I. 项… II. ①斯… ②陈… III. 软件开发—项目管理 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2013) 第 306604 号

版权所有·侵权必究

封底无防伪标均为盗版

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：01-2013-8508

George Stepanek: Software Project Secrets: Why Software Projects Fail (ISBN: 978-1-430-25101-9).

Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright © 2012 by George Stepanek. Simplified Chinese-language edition copyright © 2014 by China Machine Press. All rights reserved.

This edition is licensed for distribution and sale in the People's Republic of China only, excluding Hong Kong, Taiwan and Macao and may not be distributed and sold elsewhere.

本书原版由 Apress 出版社出版。

本书简体字中文版由 Apress 出版社授权机械工业出版社独家出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

此版权仅限在中华人民共和国境内（不包括中国香港、台湾、澳门地区）销售发行，未经授权的本书出口将被视为违反版权法的行为。

为什么 70% 以上的软件项目会失败？至今没有人能给出系统且合理的解释，本书试图探究其中的原因并给出解决方案。这是所有软件开发团队都应该反复阅读的一本经典著作，是一位拥有十几年软件开发和项目管理经验的资深专家的智慧结晶，这其中有成功的经验，更多的则是在项目中经历的挫折和失败的教训总结，可以借鉴，发人深省。

本书分为两部分。第一部分（第 1 ~ 4 章）比较详细地描述了软件项目失败的原因，阐释使软件开发与众不同的 12 个特征，并揭示不适用于软件开发项目的 10 个隐藏假设，涉及范围管理、时间管理、成本管理、质量管理以及风险管理等内容，然后通过一个模拟案例的研究，说明这些问题是如何导致一个本来前途光明的项目走向失败的。第二部分（第 5 ~ 7 章）详细介绍项目成功的方式与方法，涉及水晶方法、极限编程和 Rational 统一过程，并重点介绍一些战略，它们可能有助于使软件项目获得成功。

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：秦 健

北京诚信伟业印刷有限公司印刷

2014 年 1 月第 1 版第 1 次印刷

145mm × 210mm · 5.625 印张

标准书号：ISBN 978-7-111-45287-4

定 价：49.00 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

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzjsj@hzbook.com

译者序

敏捷开发是当今 IT 界的一个热门话题，因为它是针对软件开发项目失败率过高的现实应运而生的。本书是分析迄今为止被忽视的一个因素（即项目管理方法本身）的第一本书，以查明它怎样在软件开发项目上产生问题。

本书共分为两个部分，第一部分通过比较软件开发与其他行业的异同，介绍了软件的 12 个关键特征，正是它们使软件显得与众不同。在这个基础上，揭示了项目管理中对于软件开发无效的 10 个隐藏的假设，并据此组织和展开内容，深入讨论了这些假设导致项目失败的原因。第二部分则着重介绍一些敏捷方法及其特点和具体应用，以及各种方法的长处和局限性。

本书在两个部分的末尾分别使用两个案例研究来比较不同软件开发方法所带来的截然不同的结果，并在其中穿插介绍了敏捷方法如何规避可能引发问题的项目管理假设，从而使项目走向成功。通过真实的案例，读者可以学习不同敏捷方法的具体应用，认识它们之间的差别，并切实感受到它们的强大威力。

本书最后还包含一个术语表，方便读者查阅相关的术语。

本书非常适合从事敏捷开发的软件项目开发人员 and 项目经理阅读，对于有志于从事软件开发的读者，也有很高的参考价值。

参加本书翻译的人员有：陈宗斌、戴锋、许瑛琪、易小丽、陈婷、管学岗、王新彦、金惠敏、张海峰、徐晔。

由于时间紧迫，加之译者水平有限，错误在所难免，恳请广大读者批评指正。

致 谢

下面这些人绝对应该出现在致谢名单中：

Richard Cheeseman

Andrew Chessum

Kimson Co

Alistair Cockburn

Greg Forsythe

Dave Horton

Jasmine Kamante

Eugene Sergejew

Dominic Shakeshaft

Olga Stepanek

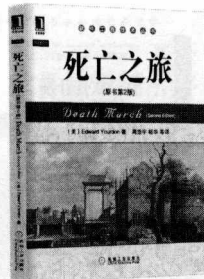
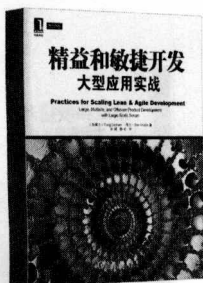
当然，还包括我的甜心 Erica。

对于上面这些人，我深深感谢他们提供的建议、意见、激励、鼓舞、专业意见等。如果没有他们的帮助，本书将远没有现在这样好。

我要特别感谢 Unisys 在我编写本书期间提供的支持，感谢整个 Apress 团队所付出的耐心和辛勤工作，还要感谢 Emily Cotlier 提供了大量的建议并做了许多文字编辑工作，这些都提高了本书质量。

George Stepanek

推荐阅读



■ 精益和敏捷开发：大型应用实战

作者：(加) Craig Larman 等
ISBN: 978-7-111-32647-2
定价：69.00元

■ 代码之殇 (原书第2版)

作者：(美) Eric Brechner
ISBN: 978-7-111-41682-1
定价：79.00元

■ 测试驱动开发：实战与模式解析

作者：(美) Kent Beck
ISBN: 978-7-111-42386-7
定价：59.00元

■ 领导力、团队精神和信任

有竞争力软件团队的管理原则、方法和实践
作者：(美) Watts S. Humphrey 等
ISBN: 978-7-111-38225-6
定价：59.00元

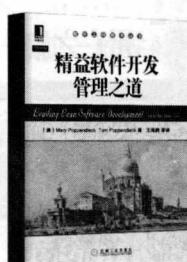
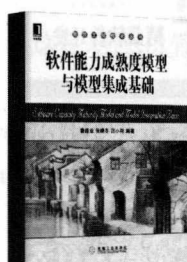
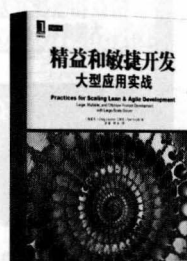
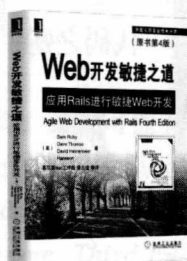
■ 死亡之旅 (原书第2版)

作者：(美) Edward Yourdon
ISBN: 978-7-111-35998-2
定价：49.00元

■ 精益软件开发管理之道

作者：(美) Mary Poppendieck 等
ISBN: 978-7-111-34082-9
定价：49.00元

推荐阅读



■ 个体与交互：敏捷实践指南

作者：(美) Ken Howard 等
ISBN: 978-7-111-37428-2
定价：45.00元

■ Web开发敏捷之道：应用Rails进行敏捷Web开发（原书第4版）

作者：(美) Sam Ruby 等
ISBN: 978-7-111-37404-6
定价：59.00元

■ 精益和敏捷开发大型应用实战

作者：(加拿大) Craig Larman 等
ISBN: 978-7-111-32647-2
定价：69.00元

■ 软件能力成熟度模型与模型集成基础

作者：黎连业 等编
ISBN: 978-7-111-33735-5
定价：49.00元

■ 精益软件开发管理之道

作者：(美) Mary Poppendieck 等
ISBN: 978-7-111-34082-9
定价：49.00元

■ 敏捷技能修炼：敏捷软件开发与设计的最佳实践

作者：(美) Alan Shalloway 等
ISBN: 978-7-111-39527-0
定价：59.00元



北航

C1710271

目 录

译者序

致谢

第一部分 为什么软件项目会失败

第 1 章 简介	2
第 2 章 为什么软件与众不同	6
2.1 软件是复杂的	7
2.2 软件是抽象的	9
2.3 需求不完整	11
2.4 技术在快速变化	12
2.5 最佳实践不成熟	13
2.6 技术是一个庞大的领域	15
2.7 技术经验不完整	16
2.8 软件开发就是调查研究	17
2.9 自动处理重复性工作	19
2.10 构造实际上就是设计	20
2.11 改变被认为很容易	22
2.12 改变是不可避免的	23
2.13 小结	24

第 3 章 项目管理假设	26
3.1 隐藏的假设	27
3.2 范围管理	28
3.3 时间管理	32
3.3.1 活动定义	32
3.3.2 活动排序	35
3.3.3 活动持续时间估计	39
3.3.4 进度安排	42
3.4 成本管理	43
3.4.1 资源规划	44
3.4.2 软件文档	46
3.4.3 开发人员生产率	48
3.4.4 成本估计	50
3.5 质量管理	51
3.5.1 指标	51
3.5.2 检查表	52
3.6 风险管理	53
3.6.1 风险接受	53
3.6.2 风险转移	55
3.6.3 风险避免	55
3.6.4 风险缓解	55
3.7 小结	56
第 4 章 案例研究：计费系统项目	57
4.1 需求	57
4.2 规划	58
4.3 设计	60
4.4 构造	61

4.4.1 编码	61
4.4.2 集成	62
4.5 测试	64
4.6 后果	67
4.7 小结	68

第二部分 怎样使软件项目获得成功

第 5 章 新的敏捷方法	72
5.1 所选的方法	73
5.2 水晶方法	75
5.2.1 频繁交付	76
5.2.2 反思改进	77
5.2.3 密切或渗透式交流	78
5.2.4 人身安全	80
5.2.5 专注	81
5.2.6 容易访问专家级用户	82
5.2.7 具有自动化测试、配置管理和频繁集成的 技术环境	82
5.2.8 使用水晶方法	84
5.3 极限编程	84
5.3.1 规划策略	86
5.3.2 测试	87
5.3.3 结对编程	87
5.3.4 重构	88
5.3.5 简单设计	89
5.3.6 代码集体所有权	89
5.3.7 持续集成	90
5.3.8 现场客户	91

5.3.9	小型发布	91
5.3.10	每周 40 小时工作制	92
5.3.11	编码标准	92
5.3.12	系统隐喻	93
5.3.13	使用 XP	94
5.4	Rational 统一过程	95
5.4.1	阶段	97
5.4.2	迭代	98
5.4.3	角色	98
5.4.4	工件	99
5.4.5	活动和工作流	99
5.4.6	过程配置	100
5.4.7	用例驱动的开发	100
5.4.8	可视化建模	101
5.4.9	使用 RUP	102
5.5	利用敏捷缓解风险	103
5.5.1	不完整的需求和范围改变	103
5.5.2	工具和技术没有像预期的那样工作	104
5.5.3	开发人员缺乏技能和专业知识	104
5.5.4	新软件有缺陷并且需要返工	105
5.5.5	参与项目的员工离职	105
5.6	小结	106
第 6 章 规划敏捷项目的预算		108
6.1	软件开发的预算	109
6.2	持续开发	111
6.3	按需编程	113
6.4	SWAT 团队	115
6.5	子团队封装	116

6.6 特性权衡	118
6.7 分诊	119
6.8 范围研究	121
6.9 结合这些技术	123
6.9.1 主要的遗留系统	123
6.9.2 次要的遗留应用程序	124
6.9.3 主要的新系统	124
6.9.4 次要的新应用程序	125
6.10 敏捷离岸外包	126
6.11 小结	128
第 7 章 案例研究：再论计费系统	129
7.1 方法	129
7.2 初始阶段	130
7.3 范围研究	131
7.4 细化	136
7.5 构造	139
7.6 交付	142
7.7 结局	143
7.8 小结	144
后记	146
附录 敏捷宣言	148
术语表	150
参考资料	159

第一部分

为什么软件项目会失败

第 1 章 简介

第 2 章 为什么软件与众不同

第 3 章 项目管理假设

第 4 章 案例研究：计费系统项目

第 1 章

简 介

你的老板要求你监督新计费系统的开发，你把一位能干的项目经理和一组精挑细选的开发人员召集在一起。他们选择了最新的技术和工具来构建该系统。商业分析师与会计经理促膝长谈，并且写出了一组详细的需求。项目具有获得成功所需的一切条件，不是吗？

显然不是。6 个月后，项目已经延期了，并且超出了预算。开发人员连续加班了几个星期，并且其中一位已经离职了，尽管如此，这个软件似乎从来没有接近完成。一部分问题在于会计团队一直声称该软件没有做他们需要的工作，并且他们总是不断地提出一连串“基本的”变更要求，更不要说泛滥成灾的错误报告了。你的老板在听到这些后，感到异常震怒。

那是什么地方出了问题呢？

无论怎样，大多数公司都会出这样的问题。依据 Standish Group [2001] 的研究报告，2000 年只有 28% 的软件项目完全获得了成功（参见图 1-1）。另外有大约 23% 的软件项目被取消了，其余的则出现了各种各样的问题，比如严重延期（平均有 63%）、超出预算（45%）、功能缺乏（33%），或者更常见的是，所有这些问题集中出现。

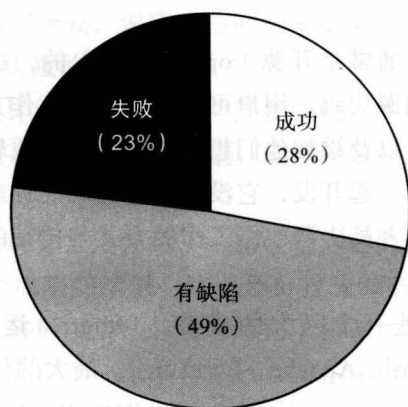


图 1-1 2000 年软件项目的成功和失败

在新西兰司法部，耗资 4200 万美元的新案件管理系统超出了预算 800 万美元，并且延期一年才完成，于 2003 年推广使用。人们本来翘首期盼该系统提供 27 种功能，但是只实现了其中的 16 种。这个系统不仅没有提高工作效率，反而由于数据输入量增加了一倍，实际上增加了法院管理案件的时间。系统实现后的审查确定了超过 1 400 个显著的问题。但是，“开发人员面临的唯一挑战是大型、复杂的系统常见的那些问题” [Bell 2004]。

与之相反，在工程和建筑业中将有很大的差别。依据 *Engineering News-Record* 的观点，对于客户的质疑，94% 的项目都能够满足他们想要的项目结果，这暗示建筑项目的失败率比软件项目要低得多。这也就是为什么 2004 年 5 月在戴高乐机场（巴黎）新建的 2E 航站楼管形屋顶倒塌事件成为全世界的头条新闻：它是如此罕见。失败的软件项目太常见了，很难获得这样的关注。

通过查看商业和非商业软件开发，我们可以知晓其原因。商业软件是公司出于盈利目的而制作的。一些软件是为个别客户量身定制的，比如计费系统，但是也有一些像 Microsoft Word 这样通用的“现成”产品。这些软件几乎全都是在项目或者一系

列项目内创建的。

非商业软件通常是开源（open source）的，这意味着任何人都可以阅读它的源代码。用户可以查明它的工作方式，并做出修改以修正错误，以及添加他们想要的功能。开源软件由来自世界各地的开发人员一起开发，它没有固定的功能列表、预算或最后期限。开源开发人员协调他们工作的方式与传统的项目管理差别相当大。

开源软件是一个巨大的成功。“Internet 运行在开源软件（BIND、Sendmail、Apache、Perl）上”，最大的计算机图书出版商 O'Reilly & Associates 的 CEO Tim O'Reilly 说道。开源软件一般具有比商业软件少得多的可靠性问题或错误。但是，如果按照我们用于衡量商业项目的相同标准，那么它还是成功的吗？毕竟，如果没有时间限制，任何项目不都将获得成功吗？

事实上，无限的时间足以弥补低下的工作效率。不过，开源工作人员的工作效率名扬四海。1991 年，Linus Torvalds 在不到一年的时间内基本上靠他自己一个人就编写了一个完整的、稳定的操作系统内核（Linux）。与此同时，8 位核心贡献者一起组建了 Apache Group，之后用了不到一年的时间就创建了 Apache 1.0，它是一款如此有吸引力的软件，以至于它变成了 Internet 上使用最广泛的网页服务器。

这些成功暗示可以很好地在传统的项目管理之外进行软件开发。考虑到项目管理技术非常适合于大多数其他的领域，这一点就令人困惑了。在建筑和工程领域就可以看到这样的情况。软件开发必定存在着某个与众不同的方面，它使得项目管理走向失败。

第 2 章将开始具体分析，首先将确定软件的特征，在软件开发流程中，正是软件的特征使它们显得与众不同。然后将这些特征与项目管理的最佳实践做比较，以便发现针对软件开发的项目管理流程是在什么地方开始出问题的。本书的第一部分末尾展示