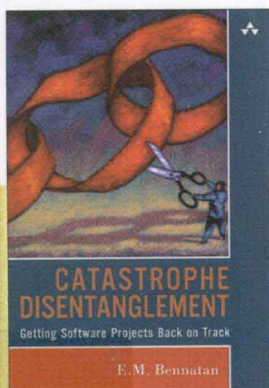


灾难拯救 让软件项目重回轨道

Catastrophe Disentanglement
Getting Software Projects Back on Track

[美] E.M. Bennatan 著
侯艳飞 侯玉芳 李 萌 译



Jolt 大奖精选丛书

灾难拯救 让软件项目重回轨道

Catastrophe Disentanglement
Getting Software Projects Back on Track

[美] E.M.Bennatan 著
侯艳飞 侯玉芳 李 萌 译

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



内 容 简 介

Jolt大奖素有“软件业之奥斯卡”的美称，本丛书精选自Jolt历届获奖图书，以植根于开发实践中的独到工程思想与杰出方法论为主要甄选方向。本书是作者在几十年软件项目管理实践经验的基础上写成的，从软件项目是否需要拯救的判断到具体拯救的步骤，面面俱到，为拯救陷入灾难的软件项目提供了一套易理解且便于操作的有效方法。

本书荣获2007年Jolt世界图书大奖，适用于软件项目经理和高级经理，也可供软件开发人员和与软件项目有关的其他人员参考，还可作为软件工程和项目管理方面的教材。

Authorized translation from the English language edition, entitled *Catastrophe Disentanglement: Getting Software Projects Back on Track*, 1E, 0321336623 by Bennatan, E.M., published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2006 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., And PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2012

本书简体中文版专有出版权由 Pearson Education 培生教育出版集团亚洲有限公司授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标贴，无标签者不得销售。

版权贸易合同登记号图字：01-2007-4097

图书在版编目（CIP）数据

灾难拯救：让软件项目重回轨道 / （美）本拿塔（Bennatan, E.M.）著；侯艳飞，侯玉芳，李萌译. —北京：电子工业出版社，2012.6

（Jolt 大奖精选丛书）

书名原文：Catastrophe Disentanglement: Getting Software Projects Back on Track

ISBN 978-7-121-15421-8

I. ①灾… II. ①本… ②侯… ③侯… ④李… III. ①软件开发—项目管理 IV. ①TP311.52

中国版本图书馆 CIP 数据核字（2011）第 252547 号

策划编辑：张春雨 符隆美

责任编辑：徐津平

印 刷：

装 订：北京中新伟业印刷有限公司

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：787×1092 1/16 印张：17.5 字数：294 千字

印 次：2012 年 6 月第 1 次印刷

定 价：59.00 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。

经久不息的回荡

今时的读书人，不复有无书可读之苦，却时有品种繁多而无从择优之惑，甚而专业度颇高的技术书领域，亦日趋遭逢乱花迷眼的境地。此时，若得觅权威书评，抑或有公信力的排行榜，可按图索骥，大大增加选中好书的命中率。然而，如此良助，不可多得，纵观中外也唯见一枝独秀——素有“软件业奥斯卡”之美誉的 Jolt 奖！

震撼世界者为谁

在计算设备已经成为企业生产和日常生活之必备工具的今天，专业和大众用户对于软件的功能、性能和用户体验的要求都在不断提高。在这样的背景下，如何能够发挥出软件开发的最高效率和最大效能，已经是摆在每一个从业者面前的重大课题，而这也正是 Jolt 大奖横空出世的初衷及坚持数年的宗旨。

Jolt 大奖历时 20 余年，在图书及软件业知名度极高，广受推崇。奖如其名，为引领计算机科学与工程发展主流，Jolt 坚持将每年的奖项只颁给那些给整个 IT 业界带来震撼结果的图书、工具、产品及理念等，因一流的眼光及超高的专业度而得以闻名遐迩，声名远播。

除图书外，Jolt 针对软件产品设有诸多奖项分类，如配置管理、协作工具、数据库引擎/数据库工具、设计工具/建模、开发环境、企业工具、库/框架、移动开发工具等。但图书历来是 Jolt 大奖中最受瞩目且传播最广的一个奖项分支。Jolt 曾设有通用类图书、技术类图书等分类，每个分类又设有“卓越奖”（Jolt Award，一般为一个）和“生产力奖”（Productivity Award，一般为 2 或 3 个）。

获奖技术图书一经公布，即打上经典烙印，可谓一举“震撼全世界”（赞助商 Jolt 可乐的广告词）。

作为计算机技术图书的厚爱者，我们总在追问——是谁在震撼世界，是谁在照亮明天？Jolt 大奖恰似摆在眼前的橱窗，让我们可以近距离观看潮流在舞蹈，倾听震撼在轰鸣！

朝花夕拾为哪般

Jolt 像是一年一度的承诺，在茫茫书海中为我们淘砺出一批批经得起岁月冲刷的杰作，头顶桂冠的佳作也因而得以一批批引进中国，为国人开阔了眼界，滋补了技术养分。然而，或因技术差距造就的生不逢时、水土不服，或因翻译、制作的不如人意，抑或是疏于宣传等诸多原因，这些经典著作在国内出版后，尽管不乏如获至宝的拥趸，却仍不为诸多人所知，从而与大量本应从中获益的读者擦肩而过。既然这生生错失的遗憾本不该发生，则更不应延续。为此，我们邀国外出版同行、国内技术专家一道，踏上朝花夕拾之路，竭力为广大读者筛选出历久弥新、震撼依旧的 Jolt 图书精品。

Jolt 获奖图书皆由业界专家一致评出，并得到软件从业人员的高度认可，虽然这些书今天读来，不再能看到上世纪史诗时代那般日新月异的理论突破，以及依赖于高深繁复的科学研究所取得的系统化成果，更多是在日复一日的开发实践中总结和提炼出来的工程思想和方法论。重新选材之所以有所弃取，从 Jolt 多年来的评奖规律中可窥端倪——

一万小时真理见

凡是在工程思想领域取得革命性、颠覆性突破的图书，就被归于“震撼”获奖分类。比如，从基于过程的程序设计模型过渡到面向对象的全新模型，就是软件开发思想上的一次带来巨大震撼的革命；再比如，打破传统的瀑布模型而转向持续集成的软件交付模型，这也是一场业界的重大思想转变。像这样的重大思想突破，可以说是数年甚至数十年一遇的，而荣获 Jolt 大奖的图书中更为常见的，则是基于最佳实践的“生产效率”获奖者。获得此类殊荣的图书，都是作者们从平凡的、重复的，甚至用一般人的眼光看来不怎么起眼的日常开

发实践中，以独具的慧眼、过人的耐心和大胆的创新，闯开一条不平常道路的心血与经验总结。

这些图书所涉及的主题，都是普通的软件开发人员每天要面对的工作——代码阅读、撰写测试用例、修复软件问题……但就是这样貌似平淡无奇的工作，是否能每一天、每一个项目都做好，着实拉开了软件开发人员素质的差距，也决定了软件企业开发出来的产品和服务的质量。我们中国有一句古话，叫做熟能生巧；某位著名企业家也说过一句家喻户晓的名言：“把简单的事千百万次地做好，就是不简单的。”这些朴素而实际的真理，同样也是本套丛书最能彰显的所谓程序员精神。它建立在脚踏实地的实践基础之上，也充满了对于自由和创新的向往。

名作可堪比名曲

就不因岁月流逝而褪色来说，与这些 Jolt 名作相媲美者，只有那些百年响彻、震撼古今的经典名曲。希望本丛书带给大家的每部著作，也如百听不厌的乐曲，掩卷良久方余音绕梁，真知存心。仔细想来，软件开发与古典音乐岂非有异曲同工之妙？既是人类心智索问精确科学的探究，亦是寻觅美学享受的追求。工程是艺术的根基，而艺术是工程的极致。衷心地希望各位读者能够认真阅读本丛书的本本珍品，并切实地用于自己的日常工作中，在充分享受大师魅力的同时，为中国的软件事业谱写更多、更震撼的乐章。

电子工业出版社博文视点

二零一二年春

译者序

自 20 世纪 60 年代“软件危机”提出以来，软件工程理论和方法得到了长足的发展，软件项目开发效率和成功率也有了显著提高。但是，五十多年的历史也表明，软件项目的高失败率现象并不会在短期内消失。每一位从事软件项目开发或管理的人在其职业生涯中，应该都遇到过进度、成本及（或）质量未能达到预期结果的情况，这些情况很多时候发展成了项目灾难并导致项目失败。然而，纵观市面上浩如烟海的关于软件工程的书籍，不难发现，关于如何预防项目灾难和项目失败的书籍成千上万，但关于如何处理软件项目陷入灾难面临失败情况的书籍却寥寥无几。一旦软件项目陷入灾难或已处在失败的边缘，我们该怎么办呢？任项目按照之前的行为方式继续下去，期盼会有奇迹发生吗？还是能做些什么扭转形势反败为胜呢？

引发软件项目灾难的原因多种多样，但软件项目灾难在表现上可归纳为项目在进度、开支和质量这三个方面有一个或多个方面失控。如果对软件项目陷入灾难判定及时，那么多数情况下，将能够通过本书所介绍的方法化解灾难，使项目重新回到正规并走向成功。本书并没有详细介绍很多理论知识，而是本着实用原则，结合一些真实案例，重点给我们提供了一套经过很多成功拯救经验积累出来的、易于理解和便于操作的、有效的和理智的识别项目是否陷入灾难和拯救陷入灾难的项目的方法。

本书作者 E. M. Bennatan 在软件项目管理方面有着丰富的实践经验，并且是一位资深顾问和演讲台上的一位常客。他所提出的拯救陷入灾难的软件项目的“10 个步骤”，单看每一步的话并不会发现什么神秘之处，每一步都可能被很多软件项目开发或管理人员使用过，但在将这些步骤组合在一起，将它们作为一个有序的且必须完整实施的集合体来使用，并在不超过两周的时间表内完

成时，就具有了异乎寻常的强大效力，而且无论对于传统项目还是敏捷项目来说都有这样的效力。

本书填补了软件项目管理领域一个明显的空白。对于软件项目经理、软件开发机构高级经理、软件开发人员及所有那些认为自身参与的软件项目正陷入灾难的人来说，本书都是一本值得阅读和学习的参考书。

本书英文原著通俗易懂，但由于时间仓促，2008 年出版的本书中文版第一版在一些章节上存在翻译质量不尽如人意之处。借此次再版的机会，我们对这些章节进行了修订，以期达到更好的翻译品质。同时，借这个机会，我们想向购买过本书中文版第一版的读者朋友表示诚挚的歉意。

本书在翻译过程中，得到了贾丽冰、陈宇、刘金哲、高晋芳、黄晓波、赵金福、张春雨、游军玲等人的大力帮助和支持，在此一并表示感谢。由于译者水平有限，错误和不妥之处敬请广大读者朋友批评指正。

侯艳飞

2011 年 11 月于北京

关于作者

E. M. Bennatan 拥有丰富的管理实战经验。这些经验源自他在摩托罗拉公司多年担任高级主管的经历。在任期间，他带领团队开发了很多大型软件系统，并领导过多个跨国设计中心。他还曾是米德威公司（Midway Company）的工程副总裁，任职期间管理着数百名软件和硬件工程师。Bennatan 先生经常在软件项目管理方面做演讲。他还是《在预算范围内按时完成：软件项目管理、实践和技术（第三版）》（*On Time Within Budget: Software Project Management, Practices and Techniques, Third Edition*）一书的作者。Bennatan 先生目前是先进项目解决方案公司（Advanced Project Solution Inc., www.AdvancedPS.com）的总裁以及波士顿 Cutter 联合公司（Boston Cutter Consortium）的高级顾问。

前言

几年前，我听了这样一个故事。一个俄国人、一个法国人、一个日本人和一个美国人不幸被食人族抓住了。在被扔进沸水锅之前，酋长告诉他的俘虏，每人可提一个请求，而他会满足他们在这世上的最后要求。

俄国人请求喝最后一杯伏特加，法国人请求让一位年轻的本地姑娘给他最后一吻，日本人说他请求做最后一次关于质量的演讲。最后轮到了美国人说自己的请求，他说：“请先把我扔进沸水锅吧，这样我就不必再听一次关于质量的演讲了！”

什么事都要讲个时机和场合。当一个软件项目陷入严峻的困境时，软件开发机构最想听到的是他们应怎样扭转败局，使项目进行下去。然而此时，并没有可遵循的 PMI、IEEE、SEI 或 ISO 标准帮助他们拯救项目。PMI、IEEE、SEI 和 ISO 这些组织提供了预防项目失败的方案，而没有提供拯救项目于危难的办法。当项目迫近“被扔入沸水锅”的结局之时，它最后的请求是“救我”，而不是“再告诉我一次如何避免陷入困境”。

本书是一本“救治”之书。本书探讨了如何拯救面临失败的软件项目使之回到正轨，虽然其中也偶有内容论及灾难预防。本书描述了拯救（或解救）面临失败的软件项目（或称软件项目灾难）的 10 个步骤。本书有广泛的目标读者群，包括软件开发人员、项目经理、高级管理人员以及软件项目利益攸关者（同软件项目之间存在很大利益关系的人）。

本书也旨在成为一本教材。在本书每一章的末尾，都有本章小结，并提供了习题。

本书中有的内容要求读者具备一些软件工程知识，不过这样的内容很少，

具备软件工程知识并不是使用本书的必要条件。本书对于缺少管理知识的软件工程师和对于缺少软件开发知识的项目管理人员一样有用。

本书共包括 13 章：

- 第 1 章为绪论。本章介绍了灾难拯救的概念，探讨了软件项目何时需要拯救行动的介入。在本章中，还对本书中使用的几个基本术语进行了阐释。
- 第 2 章论述了判断项目灾难来临的方法。对于遇到麻烦的项目来说，本章是决定是否需要对它采取后面各章中描述的拯救步骤的一章。
- 第 3 章至第 12 章描述了灾难拯救过程的 10 个步骤（每章描述一个步骤）。
- 第 13 章为结束语，标题是“把最后的拼图放入位置”。在本章中，有些内容是关于灾难预防的。本章综览之前各章描述的 10 个步骤，并阐述了如何进行时间安排以使全部拯救过程在两周时间里完成。

本书所介绍的拯救过程中，很多步骤之间存在交叠，且每一步骤都依赖于它前面的步骤，因此，本书不适合随兴翻到哪里看哪里、“东一榔头西一棒槌”的阅读方式。如果读者从头到尾地了解了整个拯救过程，就更容易理解每个拯救步骤。因此，强烈建议您在实施拯救过程之前从头到尾学习本书全部内容。不过，您也不必因为我的上述建议而气馁，本书每一章都有“本章小结”，您可以仅仔细阅读与正在实施的拯救步骤有关章节而对其他各章只阅读“本章小结”的简化方式来学习。

本书内容偏重实践而非理论。本书在介绍很多方法和技术时，并未说明其理论基础。不过，本书中提供了大量的参考书目，以飨对理论背景感兴趣的读者。参考书目信息见本书末尾部分。

在本书出版之前，有一些关于项目灾难局面能否扭转的讨论，也就是说，当我们想拯救一个项目的时候，将这个项目称为灾难是否合适。对于任何一名在大型技术公司工作过并听到过某位沮丧的高级经理说“这个项目是个灾难！”的人来说，答案都是很明显的。如果灾难局面不可扭转，那么，项目在彼时彼

地就会被放弃了，但是，高级经理接下来通常会说的是：“我们需要马上使它回到轨道！”是的，这正是本书所要论述的。

我在摩托罗拉公司和其他一些技术公司从事了多年的软件项目管理工作，并对数百家软件开发机构的软件项目开发数据进行了收集和分析，这是形成本书中“拯救”概念的基础。在本书之前，我曾写过一篇同名的论文，这篇论文发表在了美国国防部出版的软件工程期刊 *CrossTalk* 上。

感谢埃米尔（Amir）在本书问世过程中给予的极大帮助，他的贡献和评论是无价的。

E. M. Bennatan

2006 年 1 月

目录

第 1 章 绪论	1
1.1 灾难拯救过程概述	4
1.1.1 案例研究	4
1.1.2 做出拯救决定	6
1.1.3 拯救过程	6
1.2 一些调查数据	9
1.3 一些提示	11
1.4 本章小结	14
第 2 章 确定项目是否陷入灾难	17
2.1 进度	20
2.1.1 设置进度警报器	21
2.1.2 调整进度警报器	24
2.1.3 监视延长后的时间表	26
2.2 预算	29
2.2.1 设置预算警报器	30
2.2.2 其他需考虑的事项	34
2.3 质量	35
2.3.1 问题列表警报器	36
2.3.2 顾客满意度警报器	39

2.4 学会利用经验	40
2.5 本章小结	41
习题	43

第3章 第1步——停止 47

3.1 停止项目	48
3.1.1 为什么停止项目	48
3.1.2 谁来停止项目	49
3.1.3 项目停止程序	50
3.2 准备下一步	52
3.3 开展团队行动	53
3.4 处理反对意见	55
3.5 可能出现哪些问题及如何解决	56
3.6 本章小结	58
习题	60

第4章 第2步——选定评估者 62

4.1 该选谁——合格评估者的素质要求	65
4.2 案情陈述	66
4.2.1 应包含的内容	67
4.2.2 管理者的承诺	69
4.2.3 评估者的承诺	70
4.3 大型软件项目	72
4.4 可能出现哪些问题及如何解决	73
4.5 本章小结	76
习题	77

第5章 第3步——评估项目现状 79

5.1 评审	82
--------------	----

5.1.1 软件项目评审概述.....	82
5.1.2 评审面临失败的软件项目	84
5.2 项目状态信息的来源	87
5.2.1 口头的状态信息	88
5.2.2 操作性的状态信息	89
5.2.3 文档类信息	90
5.3 评估大型软件项目	91
5.3.1 大型项目有何不同	91
5.3.2 评估团队	93
5.3.3 评估大型项目的指导方针	93
5.4 拼拼图	95
5.5 可能出现哪些问题及如何解决	97
5.6 本章小结	100
习题	101

第 6 章 第 4 步——评估项目团队 103

6.1 一般原则	105
6.2 评审团队整体	108
6.3 评审项目管理	110
6.4 评审团队成员	113
6.5 整合信息	115
6.6 可能出现哪些问题及如何解决	116
6.7 本章小结	118
习题	120

第 7 章 第 5 步——确定最低目标 121

7.1 项目目标和拯救过程	123
7.1.1 区分目标、具体目标、需求和交付成果	123
7.1.2 项目目标由谁制定	125

7.1.3 同目标监督者结成同盟	126
7.2 目标最低化的准则	127
7.2.1 降低目标的过程	128
7.2.2 一个降低目标的案例	129
7.2.3 处理反对意见	132
7.3 大型项目的目标最低化	134
7.4 可能出现哪些问题及如何解决	135
7.5 本章小结	139
习题	141

第 8 章 第 6 步——确定最低目标能否实现

142

8.1 可实现的目标	143
8.1.1 可行性分析方法	144
8.1.2 被拯救项目的可实现目标	146
8.1.3 如果目标不可实现	148
8.2 中期报告	149
8.3 可能出现哪些问题及如何解决	150
8.4 本章小结	152
习题	154

第 9 章 第 7 步——重建项目团队

155

9.1 回顾团队评估	157
9.2 识别问题	158
9.3 重建团队	162
9.3.1 应对变更	162
9.3.2 实施变更	164
9.3.3 处理反对意见	167
9.4 重建大型项目团队	169
9.5 可能出现哪些问题及如何解决	171

9.6 本章小结.....	173
习题.....	175

第 10 章 第 8 步——风险分析..... 177

10.1 风险分析概述.....	179
10.2 风险分析过程.....	181
10.2.1 预测问题.....	181
10.2.2 分析阶段.....	183
10.2.3 实施风险行动方案.....	185
10.3 风险分析的一个例子.....	187
10.4 可能出现哪些问题及如何解决.....	189
10.5 本章小结.....	192
习题.....	193

第 11 章 第 9 步——修改计划..... 195

11.1 软件项目计划制定综述.....	196
11.1.1 软件项目计划概念.....	197
11.1.2 软件项目开发计划.....	198
11.1.3 项目计划制定工具.....	201
11.2 制定一个被拯救项目的计划.....	203
11.2.1 为被拯救项目制订计划的指导方针.....	204
11.2.2 其他需考虑的事项.....	207
11.3 可能出现哪些问题及如何解决.....	210
11.4 本章小结.....	214
习题.....	215

第 12 章 第 10 步——创建早期预警系统..... 217

12.1 早期预警系统的组成要素.....	220
12.2 开发数据收集.....	224