

XP 系列丛书

极限编程研究

**E~~x~~treme
Programming
*E~~x~~amined***

[加] Giancarlo Succi 编
[意] Michele Marchesi
张辉 译

3
Person
Education
人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

◆
Addison
Wesley

Pearson Education
出版集团

XP 系列丛书

极限编程研究

Extreme Programming Examined

[加] Giancarlo Succi

编

[意] Michele Marchesi

张辉

译



A1013238

 人民邮电出版社

 Addison
Wesley

Pearson Education 出版集团

图书在版编目(CIP)数据

极限编程研究/ (加) 苏西 (Succi,G.), (意) 马切西 (Marchesi,M.) 编:
张辉译. —北京: 人民邮电出版社, 2002.6
(XP 系列丛书)
ISBN 7-115-10381-X

I. 极... II. ①苏...②马...③张... III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字(2002)第 043171 号

版 权 声 明

Simplified Chinese edition Copyright © 2001 by PEARSON EDUCATION NORTH ASIA
LIMITED and Posts & Telecommunications Press.

Extreme Programming Examined

By Giancarlo Succi, Michele Marchesi

Copyright © 2001

All Rights Reserved.

Published by arrangement with Addison-Wesley, Pearson Education, Inc.

This edition is authorized for sale only in People's Republic of China (excluding the Special
Administrative Region of Hong Kong and Macau).

本书封面贴有 **Pearson Education** 出版集团激光防伪标签, 无标签者不得销售。

XP 系列丛书

极限编程研究

◆ 编 [加] Giancarlo Succi [意] Michele Marchesi
译 张 辉
责任编辑 俞 彬

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

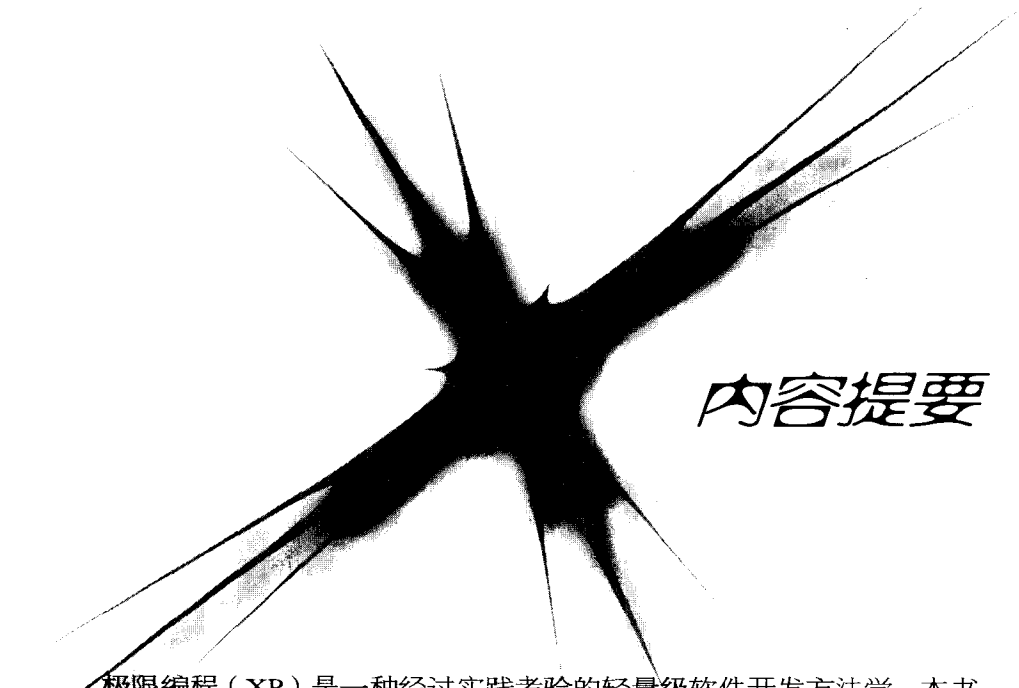
◆ 开本: 800×1000 1/16
印张: 33.75
字数: 557 千字 2002 年 6 月第 1 版
印数: 1-4 000 册 2002 年 6 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2001 - 4827 号

ISBN 7-115-10381-X/TP • 2932

定价: 70.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223



内容提要

极限编程 (XP) 是一种经过实践考验的轻量级软件开发方法学, 本书将首届极限编程和柔性过程年度会议上的 33 篇最富有深刻见解的论文专题结集成册, 提供了高水平的 XP 编程理论, 并且提出了特定的方法论、过程、技术、工具以及案例介绍。

本书共分 7 部分, 第 1 部分探讨了 XP 和其他柔性方法论的实质。第 2 部分探讨了方法论和过程。第 3 部分对柔性方法论和 UML 进行了深入探讨。第 4 部分研究了三种关键 XP 实践的经验和考虑事项: 结对编程、测试以及重构。第 5 部分回顾了着眼于重构提供结对编程支持的现有工具、于 XP 团队的支持以及测试。第 6 部分介绍了在 XP 上的实践经验。第 7 部分就如何解决 XP 某些棘手的方面介绍了若干闪光的思想。

本书汇集了 XP 理论家和实践家们的真知卓见, 专题探讨的形式让读者可以轻而易举地研习和运用这些材料。本书适合于软件开发人员、软件项目管理人员、软件工程研究人员, 以及所有想要深入了解 XP 的各界人士参考。

XP 系列丛书

Kent Beck, 丛书顾问

极限编程, 通常称为 XP, 是一种针对业务和软件开发的规则, 它的作用在于将两者的力量集中在共同的、可以达到的目标上。XP 团队以可持续的步调生产优质软件。选择 XP “书” 中的实践时考虑的是它们对人类创造力的依赖和对人类弱点的包容。

虽然 XP 经常作为一系列实践出现, 但它可不是终点线。只有最终达到了期望的目标以后, 在做 XP 时你才能越做越好。XP 是一个起跑线。它提出这样的问题: “如何能少干活而生产出优秀的软件来?”

最初的答案是如果我们希望软件开发条理分明, 必须准备好完全接受那些要采用的实践。行事中庸就是把现在不解决的问题留到更三心二意的时候去处理。最终你会被无数不彻底的措施所包围, 以至于再也看不出程序员所创造的价值核心来自编程。

我说“最初的答案……”是因为并不存在任何最终的答案。XP 丛书的作者都是过来人, 他们回过头来讲述自己的故事。这套丛书正是他们沿途立下的路标: “当心暴龙”、“前方 15 公里为风景区”、“雨天路滑”。

不好意思, 我得回去编程了。

丛书书目:

《解析极限编程——拥抱变化》, Kent Beck

《规划极限编程》, Kent Beck 和 Martin Fowler

《极限编程实施》, Ron Jeffries、Ann Anderson 和 Chet Hendrickson

《极限编程研究》, Giancarlo Succi 和 Michele Marchesi

《极限编程实践》, James Newkirk 和 Robert C. Martin

《探索极限编程》, William C. Wake

《应用极限编程——积极求胜》, Ken Auer 和 Roy Miller

有关更多信息, 请访问丛书的 Web 站点 <http://www.aw.com/cseng/series/XP/>



中文版丛书序言

发展我国软件产业的重要性已成共识。软件工作者和许多有识之士已决心投入软件产业领域。多年努力，已取得不少进展。但也出现无法回避的进一步发展瓶颈。

对软件开发的认识已有很大进步，从只要简单地照搬规范行事，到关注软件工程、软件开发方法、软件过程改进等方面。人民邮电出版社这次组织翻译的这套 XP 系列图书，全面介绍了当前软件开发方法中一种有影响的流派：XP。其主要特征是要适应环境变化和需求变化，充分发挥开发人员的主动精神。

XP 属于轻量开发方法中较有影响的一种方法。轻量开发方法是相对于传统的重量开发方法而言。简单地理解，“量”的轻重是指用于软件过程管理和控制的、除程序量以外的“文档量”的多少。文档从不同角度提供软件开发的可见性，作为测量、预见、管理、决策和控制的客观依据。XP 等轻量开发方法认识到，在当前很多情况下，按传统观念建立的大量文档，一方面需要消耗大量开发资源，同时却已失去帮助“预见、管理、决策和控制的依据”的作用。因此必须重新审视开发环节，去除臃肿累赘，轻装上阵。

当然，轻量带来一些好处的同时，也应看到它的局限性。就是开发中常缺乏一定的预见性，容易造成结构性的质量问题，必须花大力气补救。在学习 XP 过程中，应特别注意和认识 XP 的对策。

XP 用“沟通、简单、反馈和勇气”来减轻开发压力和包袱；无论是术语命名、专著叙述内容和方式、过程要求，都可以从中感受到轻松愉快和主动奋发的态度和气氛。这是一种帮助理解和更容易激发人的潜力的手段。XP 用自己的实践，在一定范围内成功地打破了软件工程“必须重量”才能成功的传统观念。

网络技术使世界的距离缩短，促进了经济全球化的进程，快速变化和多样性（包括软件开发技术的多样性）成为当今时代的特点，也加剧了学习难度和多样性所导致的无所适从的问题。XP 精神可以启发我们如何学习和对待快速变化、多样的开发技术。成功学习 XP（及其他软件开发方法）的关键，是用“沟通、简单、反馈和勇气”来对待 XP：轻松地来感受 XP 的许多实践思想和我们的开发常识多么接近；根据自己认真实践后，对真实反馈的分析，来决定 XP 对自己的价值；有勇气接受它，或改进它，或打破它。

我很高兴了解到，人民邮电出版社正在引进其他有关软件开发方法学方面的系列著作，其中不乏荣获《Software Development》杂志年度图书大奖的好书，相信这些图书在国内的陆续出版，会对软件开发方法学的研究及教学产生积极的推动作用。



2002 年 5 月于清华园



前言

—— Paolo Succi 和 Michele Marchesi

只有天才才能让困难的事情变得简单，而任何傻瓜都会让一项简单的任务变得困难！阿尔伯特·爱因斯坦也曾经说过类似的话，我们认为这句话用在极限编程上相当贴切。极限编程——也称为 XP——和其他柔性方法论都是大规模简化软件系统开发过程的一种极端尝试，它把注意力放在了提供价值的东西上面：对系统的需求以及实现系统的代码。没有更多别的东西！

在进行极限编程的过程中，一切都是从需求开始的，而且采用了用户素材（user story）的形式。客户提供用户素材并确定用户素材的优先级。开发人员分析这样的素材并为它们编写测试……一切都归结到了代码上。代码由结对的开发人员进行开发以提高质量。对代码进行重构以促使其更加简单。参照需求对代码不断进行测试……其间没有任何东西！

需求和代码两者都要受到仔细的审查：正如所说的那样，代码只是准确地完成用户素材告诉它的任务，它什么也不会多做，并且始终保持为它最有可能的形式。

这本论文集包含了在极限编程和其他柔性方法论上的经验。它讨论了什么是极限编程——需求加代码——以及我们如何能改进极限编程。

本书里的各个主题是按照自顶而下的顺序编排的。我们从基础知识开始，然后转向过程、实践、工具支持以及经验，最后以研讨可能的新途径作为结束。

本书从讨论 XP 和其他柔性方法论的实质开始。Martin Fowler 论述了设计的作用。Peter Merel 综合介绍了 XP 的原则。

第二部分着眼于方法论和过程。在 Dirk Riehle 对 XP 和其他方法论的准则系统进行的讨论之后，Lars-Göran Andersson、Mark Collins-Cope、Carsten Jacobi、Even-André Karlsson、Martin Lipper、Hubert Matthews、Stefan Roock、Bernhard Rumpe、Henning Wolf 和 Heinz Züllighoven 等人的文稿对框架和其他大型软件开发实践进行了一次全面的分析。

在第三部分，几位作者试图将其他人认为不相容的东西结合起来：柔性方法论和 UML。Jutta Eckstein 和 Rolf Katzenberger 对统一软件开发过程（Unified Software Development Process, USDP）和 XP 进行了比较。Giuliano Armano 和 Michele Marchesi 介绍了在新成立公司中的实地经验。Marko Boger、Toby Baier、Frank Wienberg 和 Winfried Lamersdorf 详细说明了一种给软件系统快速建模的有意思的方法。Sergio Focardi 和我们两个人阐述了为什么 XP 是一种不同的使用形式化的随机图开发软件的方法。Joshua Kerievsky 将设计模式与 XP 进行了调和。

第四部分包含三种关键 XP 实践的经验 and 考虑事项：结对编程（Alistair Cockburn 和 Laurie Williams）、测试（Philip Craig、Steve Freeman、Peter Gassmann、Tim Mackinnon 和 Kevin Rutherford），以及重构（Neelam Soundarajan）。

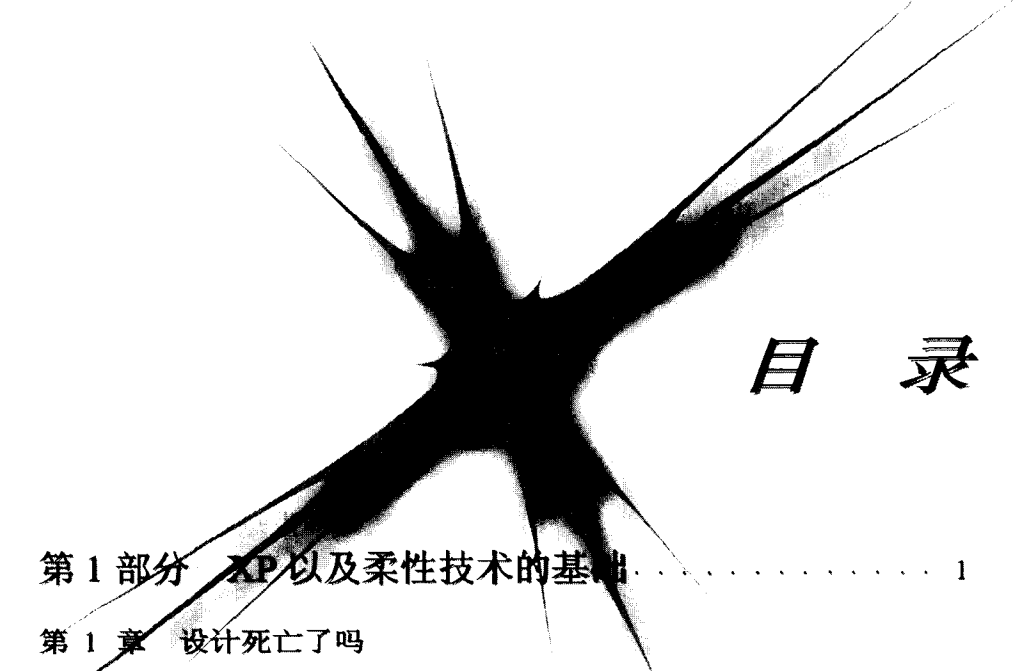
第五部分回顾了着眼于重构提供结对编程支持的现有工具（Ralph Johnson），特定于 XP 团队的支持（Jim des Rivières、Erich Gamma、Ivan Moore、Kai-Uwe Mätzel、Jan Schümmer、Till Schümmer、André Weinand 和 John Wiegand），以及测试（Renato Cerqueira 和 Roberto Ierusalimschy）。

Karl Boutin、Michael Kircher、Manfred Lange、David Levid、Peter Sommerlad 和 Don Wells 在第六部分介绍了他们在 XP 上的经验。

香槟酒留在了最后！本书的末尾就如何解决 XP 某些棘手的方面介绍了若干闪光的思想。Christian Wege 和 Frank Gerhardt 概述了他们教授 XP 的方法。Arie van Deursen、Tobias Kuipers 和 Leon Moonen 解决了遗留代码的问题。Andrei Alexandrescu 阐述了如何用 C++ 语言实现柔性方法论。Paolo Predonzani、Giancarlo Succi 和 Tullio Vernazza 讨论了如何在一个极限编程的环境下处理好对变体的管理。Luigi Benedicenti、Raman Paranjape 和 Kevin Smith 则讨论了软件智能体内在的极端灵活性。Jason Yip、Giancarlo Succi 和 Eric Liu 阐述了几个使用 XP 开发的产品如何组织到一个生产线中，但又不会变成重量级的方法。

关于作者

Michele Marchesi 是意大利卡利亚里大学（Università di Cagliari）电气与电子工程系（Dipartimento di Ingegneria Elettrica e Elettronica）的一位教授。Giancarlo Succi 是加拿大阿尔伯塔大学（University of Alberta）电气与计算机工程系（Development of Electrical and Computer Engineering）的一位教授。



目 录

第 1 部分	XP 以及柔性技术的基础	1
第 1 章	设计死亡了吗	
	—— Martin Fowler	3
第 2 章	极限编程之道	
	—— Peter Merel	17
第 2 部分	方法与过程	31
第 3 章	自适应软件开发和极限编程的准则系统的比较： 方法论之间可以怎样相互学习	
	—— Dirk Riehle	33
第 4 章	让我们进行分层：一种在 XP 中用于重构的建议 参考体系结构	
	—— Mark Collins-Cope 和 Hubert Matthews	49
第 5 章	极限架构：如何把应用瞄准演进的框架	
	—— Stefan Roock	67
第 6 章	层次结构的 XP：类比重组过程为大型项目对 XP 进行改进	
	—— Carsten Jacobi 和 Bernhard Rumpe	79

第 7 章	JWAM 和 XP: 使用 XP 进行框架开发	
	—— Martin Lippert, Stefan Roock, Henning Wolf	
	和 Heinz Züllighoven	97
第 8 章	XP 和大型分布式软件项目	
	—— Even-André Karlsson 和 Lars-Göran Andersson . . .	111
第 3 部分	柔性技术和 UML	125
第 9 章	特洛伊木马中的 XP: 重构统一软件开发过程	
	—— Jutta Eckstein 和 Rolf F. Katzenberger	127
第 10 章	一种用于新兴机构的柔性软件开发过程	
	—— Giuliano Armano 和 Michele Marchesi	145
第 11 章	极限建模	
	—— Marko Boger, Toby Baier, Frank Wienberg	
	和 Winfried Lamersdorf	163
第 12 章	一种软件维护的随机模型及其与极限编程过程的关系	
	—— Sergio Focardi, Michele Marchesi	
	和 Giancarlo Succi	177
第 13 章	模式和 XP	
	—— Joshua Kerievsky	193
第 4 部分	结对编程、测试和重构	209
第 14 章	结对编程的代价和收益	
	—— Alistair Cockburn 和 Laurie Williams	211
	附录“身边的行家”项目管理模式	231
第 15 章	在一个 Java 项目中的单元测试	

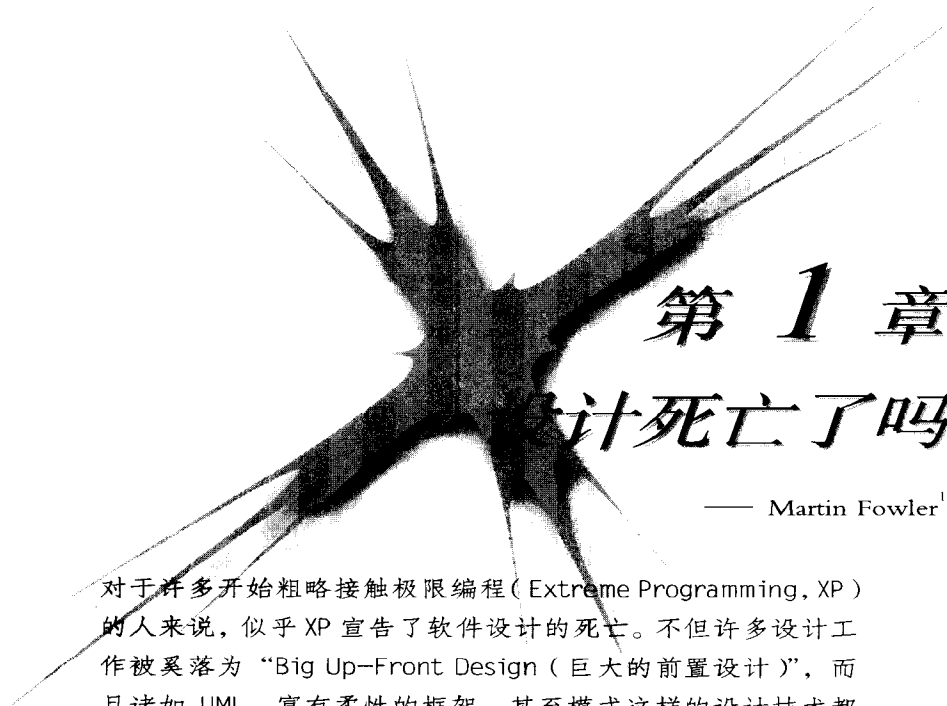
	— Peter Gassmann	235
	样本代码	253
第 16 章	使用 JUnit 翻新单元测试	
	— Kevin Rutherford	257
第 17 章	内部测试: 使用虚拟对象进行单元测试	
	— Tim Mackinnon, Steve Freeman 和 Philip Craig . . .	273
第 18 章	重构和重新推理	
	— Neelam Soundarajan	291
第 5 部分	XP 开发工具	307
第 19 章	开发重构浏览器	
	— Ralph Johnson	309
第 20 章	团队流: 极限团队支持	
	— Jim des Rivières, Erich Gamma, Kai-Uwe Mätzel, Ivan Moore, André Weinand 和 John Wiegand . . .	317
第 21 章	在极限编程中支持分布式团队	
	— Till Schümmer, Jan Schümmer	337
第 22 章	一种基于 CORBA 的分布式系统的自动测试	
	— Renato Cerqueira 和 Roberto Ierusalimsky	357
第 6 部分	实践经验	375
第 23 章	VCAPS 项目: 一个向 XP 过渡的例子	
	— Don Wells 和 Trish Buckley	377
第 24 章	采用 XP	
	— Peter Sommerlad	397
第 25 章	在一个研究和开发实验室引入极限编程	

	— Karl Boutin	407
第 26 章	在事实背后：将 XP 引入一个现有的 C++ 项目	
	— Manfred Lange	421
第 27 章	TAO 的 XP：大型、开放源代码框架的极限编程	
	— Michael Kircher 和 David L. Levine	435
第 7 部分	XP 及其之外	457
第 28 章	学习 XP：举办一个新兵训练营	
	— Christian Wega 和 Frank Gerhardt	459
第 29 章	遗留到极限	
	— Arie van Deursen, Tobias Kuipers 和 Leon Moonen	471
第 30 章	设计就在代码之中：C++ 中的增强性重用技术	
	— Andrei Alexandrescu	485
第 31 章	跟踪开发过程：一种变化性的观点	
	— Giancarlo Succi, Paolo Predonzani 和 Tullio Vernazza	501
第 32 章	软件智能体的柔性制造技术	
	— Luigi Benedicenti, Raman Paranjape 和 Kevin Smith	507
第 33 章	柔性过程怎样和软件产品线相关	
	— Jason Yip, Giancarlo Succi 和 Eric Liu	515



第 1 部分

XP 以及柔性技术的基础



第 1 章 设计死亡了吗

—— Martin Fowler¹

对于许多开始粗略接触极限编程 (Extreme Programming, XP) 的人来说, 似乎 XP 宣告了软件设计的死亡。不但许多设计工作被奚落为 “Big Up-Front Design (巨大的前置设计)”, 而且诸如 UML、富有柔性的框架, 甚至模式这样的设计技术都不被重视或者近乎于忽略了。实际上, XP 包含有许多设计思想, 只是采用了一种与既定的软件过程不同的方式来进行设计。XP 借助的多种实践让演进成为一种可行的设计策略, 以此使演进设计的概念焕发青春。设计者需要学会如何进行简单设计、如何使用重构技术保持设计清晰, 以及如何以演进的方式使用模式, 因此 XP 也带来了新的挑战 and 技巧。

极限编程 (Extreme Programming, XP) 向软件开发方面的许多共识发出了挑战。其中最富有争议的一项就是它赞成采取一种更为循序渐进的方法, 抵制在前置设计 (up-front design) 上花费巨大精力。对于它的批评者来说, 这是走 “编码加修正 (code and fix)” 开发方式——这往往被嘲笑为 “hacking”——的回头路。就连它的倡导者也常常将它看作是对设计技术 (比如 UML)、原理和模式的一种抵制。不必担心设计: 如果你聆听代码, 良好的设计就会出现。

1. 版权所有 © 2000 Martin Fowler。保留所有权利。本章的在线版本可以在 <http://www.martinfowler.com/articles/designDead.html> 找到, 在线版本偶尔会得到更新。