

Murray Cantor 著
朱剑平 等 译



软件领导

——成功开发软件的指导准则

清华大学出版社

软件领导

——成功开发软件的指导准则

Murray Cantor 著

朱剑平 等 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Software Leadership, 1st by Murray, Copyright © 2002

EISBN: 0201700441

All Rights Reserved.

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

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

本书中文简体翻译版由 Addison-Wesley 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字:01-2002-4498 号

版权所有,翻印必究。

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。

图书在版编目(CIP)数据

软件领导:成功开发软件的指导准则/坎托(Cantor, M.)著;朱剑平等译.

—北京:清华大学出版社,2004.11

书名原文:Software Leadership: A Guide to Successful Software Development

ISBN 7-302-09783-6

I. 软… II. ①坎… ②朱… III. 软件开发—项目管理 IV. TP311.52

中国版本图书馆 CIP 数据核字(2004)第 108661 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:常晓波

封面设计:立日新

印 刷 者:世界知识印刷厂

装 订 者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:148×210 印张:6.375 字数:177千字

版 次:2004年11月第1版 2004年11月第1次印刷

书 号:ISBN 7-302-09783-6/TP·6755

印 数:1~3000

定 价:19.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或 (010)62795704

序

Scott Adams 绘制的漫画 Dilbert 描述了一个非常有趣但非常愚蠢的角色，叫做“Pointy Hair Manager(尖发经理)”，他管理一家高科技公司的一个部门。除了他的发型以外，这个人的主要特点是看上去对他下属(如 Dilbert、Alice、Wally 或者 Asok)所做的事情毫不知情，也不知道什么事情能够激励他们。可笑、可恶、完全不合体统？我不知道。我们的行业实际上也有着许多“尖发经理”，无论他们是否真的留有这样的发型。这些人被拉到(有时候也违背他们自己的意愿)管理“IT(信息技术)”项目或者软件开发项目的位置上来。

您是一个“尖发经理”吗？再想一想，照照镜子。某一天可能会发现自己的门上贴上了漫画 Dilbert，如果您认为以下内容是正确的。

- 软件开发主要与编程有关。
- 管理软件项目与管理任何其他项目没有什么区别。
- 最好的方法是一开始就制定一个详细的计划，并确保每个人每天都在遵守这个计划。
- 持续开发产品的安全做法是及早将所有的需求都确定下来，以便把计划落空和以后出现意想不到的事情的风险降到最低。
- 等等。

为了真正地进行管理，而不是简单地监控，好的管理者必须理解一些本质性的东西，比如团队正在生产什么、团队的动力、团队使用的工具和工作的过程，以及与开发相关的所有风险。任何因素

II 软件领导

都可能阻碍您获得成功。软件的本质使其与其他行业不同，适用于其他产品（比如发动机或桥梁）的管理技术在软件行业将没有多大用处，或者至少不能在软件领域开发出高质量的产品。

在以下方面，软件不同于其他产品。

- 它不遵守“物理学定律”，并且还没有发现等价的软件基础定律。如果不通过某种形式的实验或者原型获得反馈，就不能精确地进行预测和计划。
- 软件本来就是“软的”、可修改的，而且外部环境（客户、用户）期望对软件进行更改、完善、进化，甚至在开发软件时也是如此。您的管理方法必须要承认这一点。
- 用于设计和开发软件的技术和工具发展得很快。这些技术对软件的生产效率、成本、风险和质量有很大的影响——必须将它们结合到计划过程中。
- 软件开发成本几乎完全与工程有关。软件产品可以用极低的成本“大批量生产”，但它的经济模型与其他制造产品的经济模型有非常大的区别。

软件的这些特点和其他特点对于软件领域的管理者来讲具有更大的挑战性。

本书有助于您不至于成为“尖发经理”。它将让您洞察到软件的开发过程和所使用的技术和工具的本质，因此您可以聪明地对开发进行思考、调节、计划、重新计划和管理风险，并领导团队走向成功，而不仅仅只是记录它们的失败。

我很高兴我们的同事 Murray Cantor 选择将他的新书以“Rational 统一过程”为基础。这将使他的新书成为团队经理们使用和将要使用 Rational 统一过程或任何变体时的首选参考书，这些过程包含的大量技术性材料可能使经理们望而却步。本书提供了有关统一过程方面的内容适当的知识，足以使经理们让他们的 Dilbert 们受到激励，保持工作效率并且获得成功。但是，无论最终决定遵循哪种过

程，都将发现这里给出的指导对于领导团队都是有价值的。

希望您阅读愉快，预祝您在下一个 IT 项目中取得成功。

Philippe Kruchten, Rational Fellow

过程开发部高级经理

Rational 软件公司

前 言

我写这本书主要是由于挫折感。作为 Rational 软件公司战略服务组织首席咨询师，我发现软件开发对许多公司来说是一个灾难性的过程。当然，从过去担任开发经理和技术领导的经验中，我知道事情不至如此。在绝大多数情况下，问题都可以追溯到不正确的领导方法。我的挫折感来自于自己知道这些问题本来能够被纠正。

很多书和文章集中讨论了软件开发管理是多么的糟糕。这不是写作本书的目的，我是要带给读者好消息：在过去的 40 年中，领导软件组织的经验积累了很多。通过这些经验教训，读者能够成为任何软件开发组织中的更好的领导者（跟随者）。

编写本书的一个原因是看到了对软件领导技巧日益增长的需要。不仅是软件行业在持续的增长，很多其他领域的公司也开始认识到他们在软件上支付的越来越多的资金。例如，让我们看一个设计医疗超声波成像仪的案例，这种成像仪用于透视子宫中的胚胎。有两种设计方法。

- 硬件为中心的方法：一家传统医疗器械供应商销售的超声波仪器，大小类似于写字台，重量和一架小型钢琴差不多，它包括了大量特殊的硬件，不仅发射和接收声波，还处理信号和生成图形。这个设备非常昂贵，难于维护和升级。
- 软件为中心的方法：一家新的小型竞争企业认识到可以利用现成的硬件制造一个超声波成像系统。使用 ASIC（可编程芯片）、微软公司的操作系统、一些传感器和接收器、液晶显示屏和其他设备即可。为了把设备推向市场，新公

司做了一些工业设计，将按钮和刻度盘布置好，还编制了大量的专用软件。新产品采用便携形式，重量为 5.4 磅，市场价格不到老设备的一半，盈利率很高。如果他们的软件设计得好，他们的产品就会比老式的系统更容易维护和扩展。upstart 公司的便携式设备在 1999 年 3 月得到了美国政府的批准，在 2000 年的前 9 个月里实现了 25 000 000 美元的销售收入，比前一年的 745 000 美元增长了很多。他们还没有完全替代老公司，但是他们的目标是这样的。为了在医疗器械领域保持竞争力，传统的设备制造商进入了软件行业。

这种现象表明大量硬件公司越来越认识到他们产品的价值在于他们的软件。自动化系统（那些盖子下面的黑盒子和发动机控制器）、电话设备（中心局交换机、移动电话）、玩具、娱乐设备和几乎所有其他设备都受到这种现象的影响。我曾经同一个大型一级汽车部件供应商的部门经理共进午餐。他的分部制作汽车座椅。我请他说出什么是他最大的难题。他的回答是控制高档座椅的软件，这软件应该记住不同司机的偏好。

阅读本书的读者

在写作本书时，我脑子里想到 3 类读者，每一类读者可能以不同的方式阅读本书：

人事主管，特别是那些背景不是软件开发的人，而是其他领域如工程、产品营销或者一般管理方面的管理者。

有经验的软件经理，正在寻求各种方式使他们的公司更加具有竞争力的人。

技术人员同时也是领导者，有兴趣成为领导者的人或者希望更

好地被领导的人。

具有技术背景的读者对熟悉的领域可以得到新的感受。他们可以获得一种更好的想法，如何提供技术领导以及从他们的领导那里获得什么。用人主管将得到新的见解，有可能发现新的更有效的领导方式。软件项目的管理者和团队领导人可以获得如何更好开展工作的想法。

内容安排

本书内容基于的前提是有竞争力的软件开发领导者应该对软件质量、最行之有效的开发实践、团队动力和恰当的领导风格有良好的认识。根据这个前提，本书将按照以下部分组织。

- “简介”部分对今天软件开发的状态提供了背景资料，定义了软件开发工作的成功标准。
- “第1章 高质量软件”，提供关于您的团队必须开发和交付产品的概况。包括对于高质量软件特性的认识，介绍了一些为保证交付质量可以采用的步骤。还讨论了为什么交付质量既是一件有关生存的大事，也可能是一件无足轻重的事情。
- “第2章 软件体系结构”，概述了如何定制软件设计。简要地描述了关于软件设计的不同观点，以及这些观点的目的和在完成质量中扮演的角色。阅读本章之后，您虽然不会成为一名架构师，但可以对软件的体系结构提出高标准的问题。
- “第3章 软件项目”，给出了大方向。本章认为软件开发与一般产品开发很相象，可以从一般产品开发领域吸取很多的经验。提供了项目开发解决问题本质的理解。软件开

发的本质是非线性的，这种非线性对于领导项目有深刻的影响。本章勾画出软件领导的关键问题，是理解全书的结构框架。

- “第4章 软件开发生产率”，介绍了软件生产率的一个概念模型，给出了帮助您公司改善完成开发任务能力的3种方法。这些方法构成了在第5章和第6章中讨论的现代软件开发实践的基础。
- “第5章 Rational 统一过程”，介绍了现代软件开发项目管理的最好方法。另外，这一章还提供了怎样组织软件开发项目的指导。第5章解释了如何确定适量过程，给出关于重用(reuse)的一个重要观点。
- “第6章 管理和领导”，描述了按照前面章节所介绍的原则和过程管理的组织中最合适的领导风格。解释了生产线和项目管理者所需要的领导风格。特别描述了软件开发管理者应该扮演的投入、建设性开发团队成员的角色。探讨了如何在不确定情况下做出和完成承诺。
- “附录A”，软件开发的3种失败方法，基于业界的经验教训。给出了为什么这些普遍的方法在软件开发中失败的观点，提出了一些不应采取的行为。

贯穿每一章的中心都围绕着领导任务。

虽然推荐每一个人都阅读所有内容，但是对于某些读者来说，一些章节的意义更大。第1章和第2章关于软件本身。这两章对于那些缺乏技术背景的读者特别有帮助。第3章到第5章提供了对开发过程的理解。对于前两章内容比较熟悉但在管理方面是个新手的读者会从这些章节中获益更多。当然，本书内容对于很多有经验的并有兴趣改善组织竞争态势的管理者也有很大的好处。第6章对于所有的读者都适用。

介绍一下附录。决定收入这些材料的原因是希望读者可以从中

VIII 软件领导

发现自己的领导方法中的一个或多个错误，并下决心尝试一些不同的东西。

这本书的内容故意编排得很简短，重点在于掌握大的方向。对全书中很多论题和内容的完整讨论是另外一本书的职责。为了给希望了解更多内容的读者指出方向，在每章的最后提供了一个注释过的参考书和文章列表。全书末尾还有一个完整的参考书目列表。

致谢

这本书得到了很多同事的帮助：他们中的有些人是复审者，有些人与我在谈话中交换过想法。特别要感谢 Walker Royce、Philippe Krutchen、William Lochrie、Robert Brownsword、Michael Drexler、David Lubanko 和两位匿名评审人，感谢他们对本书初稿内容的坦率和建设性评论。

简 介

本书并不谈论编程、面向对象方法，甚至也不谈论软件项目管理。只讨论如何领导开发小组良好地开展这些活动。本书将从领导角度讨论这些话题，同时也讨论其他内容。

企业改革方面的主要专家 John P. Kotter，将领导 (leadership) 定义为指明方向、统一思想和激励人心。另一方面，管理包括计划、组织和安排人力资源，以及控制和解决问题 [Kotter, 1996]。不是每位领导者都能成为一个合格的管理者，也不是每个管理者就能成为一个合格的领导者。胜任管理任务是必需的，但还不足以成为一名高效的领导者。本书所涉及的管理问题是为关于领导的讨论提供必要的背景知识。

有竞争力的软件组织

软件不仅是高科技企业的主要资产，对传统经济企业来说也是如此。例如，一家因生产高质量汽车而久负盛名的大型汽车公司，因为在汽车出厂后缺乏跟踪、预测和分配汽车的良好系统而损失数百万美元。类似地，他们也正在尽力从供应商那里要回保修成本。如果他们能够开发或者获得更好的软件来支持这些功能，就可以给股东增加很多的价值。

试想，在上个世纪，厂长的工作发生了什么样的改变。在 1905 年，厂长可能雇用最好的技术工人，让他们操作设备，自己定期检查事情的进展状态。到 1920 年，厂长可能实现了流水线作

X 软件领导

业，雇佣没有技能的工人来执行非常有限的工作，让生产线运转尽可能快，使产量最大化。厂长依赖于生产工艺，而不是工人的技术。现代的厂长需要领导专业技术过硬的专家团队，包括生产工程师、系统分析师和运筹研究人员。他们必须理解生产制造、供应链、配置管理和存货控制过程，还有这些过程的自动化和测量方法。他们可能每天都察看测量值，看看生产量和质量是否产生变化，让相应部门的管理者找出改变的原因，并采取适当的行动。厂长可能每个月都需要与员工们回顾一下工厂的业绩。厂长将改进生产工艺并跟踪它们的状态，定期与下面的管理层和员工就状态方面进行交流。简而言之，厂长将确切理解所有生产过程和实现这些过程自动化的工具，并且能够有效地领导技术过硬的员工。

直到最近，软件开发还特别像 1905 年或者 1920 年的生产制造，没有多少自动化。一些软件管理者雇用了最好的工匠，并给他们分配工作，而且希望得到最好的结果。许多软件公司还在以这种方式运作。他们为这个方法下了一个积极的定义，那就是“信任员工”。信任人是必要的，但是还不够。其他的管理者建立了一个详细的过程，将工作划分为一系列有序的步骤。这些管理者希望过程本身为开发软件提供方法。对于他们来讲，一旦理解了需求，开发软件就是连接各点的过程。这些管理者总是在想，为什么他们的项目实际上是那么的不可预测。本书将做出解释：虽然过程是必要的，但还不充分。团队开发软件需要领导。

作为软件开发组织的领导者，您可能并不知道如何编写代码，但是需要理解公司如何运作，并且需要参与到项目中来领导公司走向成功。或者将自己的事业作为赌注，请求您的部下按时交付软件，而在他们“辜负”您的时候您将极度失望。最好是在自己的领导岗位上学作一名专家，与员工一起工作，充分理解和认识他们从事的工作和面对的困难。

进度表上的发明

开发软件和生产实物之间存在一些明显的相似点。它们都需要具备以下内容。

- 由自动化工具支持的专家团队
- 管理纪律
- 高效的领导

它们的区别在于对创造性的需求。软件开发有时候被描述为根据需求进行的创造。团队必须创造新的程序，在一定的时间和运算范围内，满足一系列的独特功能。

由于需要创造性，所以软件开发的工作流程和经济模型、制造或建筑项目有很大的区别。事实上，软件开发与其他将创造性与商业要求相结合的其他活动有一些共同的特点，比如消费品开发或者电影开发。在 Karl Sabbagh 的著作《21 世纪喷气式飞机》(Scribner, 1966) 中就可以找到这样一个例子，这本书叙述了波音 777 喷气式飞机的开发过程。读者将找到飞机设计(不是生产)与软件项目之间惊人的相似点。

软件开发的起点被定得很高。现代软件开发工具和过程能够开发相当复杂的程序。今天，包括 100 万行新的和重用代码的程序并不罕见。我们周围到处都有软件开发在技术上取得成功的证据。

- 因特网及其协议、服务器和客户应用程序。
- 电子商务的很大一部分。
- 呼叫者从一个网络移动到另一个网络时，跟踪呼叫的蜂窝电话交换机。
- 覆盖全世界的信用卡交易系统。
- 航空订票系统。

XII 软件领导

- 所有执行企业内勤工作的企业资源计划软件，包括存货控制和人力资源管理。
- 嵌入在汽车发动机控制器中的软件。
- 支持设计波音 777 飞机无纸化传输信息模式的 CAD/CAM 软件。
- 能够让从空气动力学角度来讲不稳定的军用飞机(比如 B2)飞起来的软件。
- 能够让太空探测器精确进入遥远行星轨道上的太空导航系统。
- 制作电影的计算机动画软件，比如《侏罗纪公园》、《玩具总动员》和《恐龙》。

进一步讲，这些程序正在以前所未有的生产效率进行开发。在 20 世纪 80 年代，通常一个普通的程序员在一个大项目中可以编写大约 1 500 到 2 000 行的经测试成功的代码。今天，各种案例研究和个人经验表明生产效率为以前的 10 倍是有可能的。

每个人都可以使用提供更具有竞争力的软件开发工具和过程，区别在于领导。

成功的领导者

本书的目的是说明每个人都需要成为软件开发组织的成功的领导者。第 1 步是理解任务。



成功的软件组织承诺在预算范围内如期交付满足利害关系人(stakeholder)需求的高质量软件。

这个成功的定义与其他的定义不同。一些人将项目成功定义为能够在最初的计划安排和预算下实现最初的计划。我们的定义更实

际：要看事情的结果，而不看是否遵照了某项计划。该定义关注于领导使项目获得成功结果的组织。当在既定预算下按时开发出满足客户需求并在现场采用的系统时，您将庆祝团队的成功，您的上司也是如此。事实是，只要管理者、市场营销组织和客户不感到惊讶，即使系统没有包含最初列表上的所有功能也并不重要。

为了应用这个定义，领导者必须理解利害关系人和他们的需求、高质量软件的特点和如何让企业在既定的时间和预算约束下取得实质性的成功。进一步讲，领导者需要面对真正的用户需求和想法，并在需要发明满足这些需求的方法时，必须知道如何处理内部的不确定性。

领导是士气的关键

优秀的企业和优秀的员工相辅相成。留住一流的员工是竞争的关键。而留住最好的员工也需要好的领导。另一方面，领导工作不善则可能导致士气低落，最终会失去公司的关键人才。我曾经做过软件开发、技术领导、软件架构师、架构设计经理、项目经理和计划经理。这些经验让我知道软件管理不善对员工造成的伤害。程序员们经常会变得神情紧张、沮丧，感觉浪费了大量的精力。工程截止日期制定得毫无根据，并且管理者并不知道项目进展状况。在项目评估会议期间，他们诉说自己的困境时，程序员们有时候竟然流出眼泪来。完全可以避免这种情况的发生。

软件开发并不是一个残忍的过程。我曾经在商业和合同条件下交付过代码。当遵循本书中描述的原则和技术时，我的团队成功地开发出了软件，并没有成为这个行业中还普遍存在的“敢死队”，也没有经历疲劳战。

职业成就感

作为一个负责开发软件的组织的领导一般并不让人羡慕。如果领导者不熟悉管理成功软件开发组织的原则，就更是如此。他们发现软件对于公司的成功来讲非常关键，经常处于关键的环节，并且经常被推迟完成。管理者可能非常恐慌而员工可能非常愤慨。领导者做出的反应只是避过一个又一个危机。许多管理者甚至准备两套工作计划：一套用于向管理者报告进展状态，一套用于跟踪真实的过程。这样的领导不可能获得工作成就感或者事业的进步。其实事情不必如此。使用本书描述的背景知识和技术，领导者的精力就可以花费在履行承诺、提高员工的生活质量并构建一个具有竞争力的软件组织上面。有什么能比这让人更满意呢？