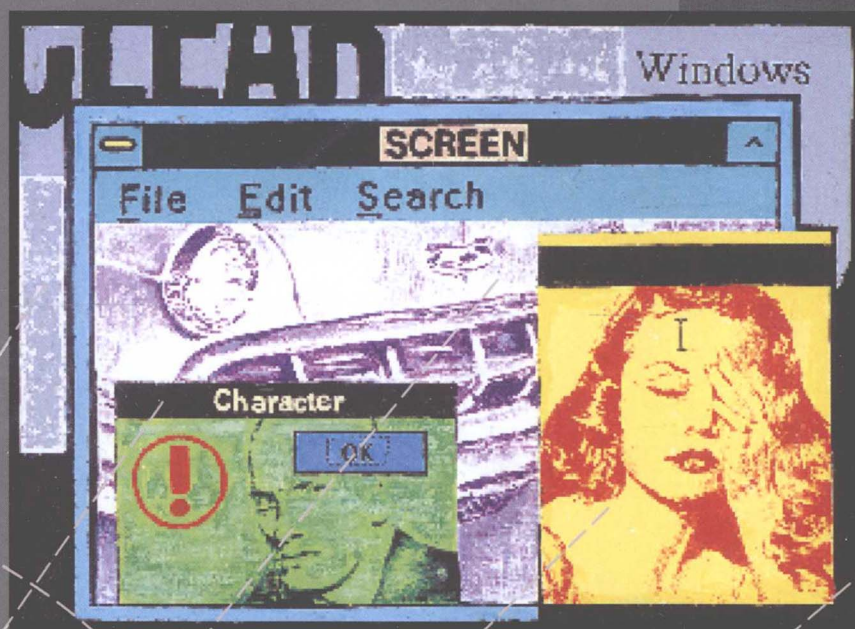


Dynamics of Software Development

软件研发之道

微软开发团队的经验法则



[美] Jim McCarthy 著
Michele McCarthy
赵俐 石华耀 译



人民邮电出版社

POSTS & TELECOM PRESS

TURING 图灵程序设计

Dynamics of Software Development

软件开发之道

微软开发团队的经验法则



[美] Jim McCarthy 著
Michele McCarthy
赵俐 石华耀 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

软件开发之道 : 微软开发团队的经验法则 / (美) 麦卡锡 (McCarthy, J.), (美) 麦卡锡 (McCarthy, M.) 著; 赵俐, 石华耀译. — 北京 : 人民邮电出版社, 2011.8

(图灵程序设计丛书)

书名原文: Dynamics of Software Development

ISBN 978-7-115-25260-9

I. ①软… II. ①麦… ②麦… ③赵… ④石… III. ①软件开发—经验—美国 IV. ①TP311.52

中国版本图书馆CIP数据核字(2011)第082408号

内 容 提 要

本书叙述了微软 Visual C++ 开发团队的故事, 通过作者的总结和归纳告诉读者如何构建一个优秀的软件开发团队, 如何在一段时间内成功地交付一个软件。作为升级版, 作者将言简意赅的法则扩展到 57 条, 包括了软件开发及营销的全部内容, 相信每一位项目经理都会对此书爱不释手。

本书适合软件设计者、开发人员、营销人员及技术主管等阅读。

图灵程序设计丛书

软件开发之道：微软开发团队的经验法则

◆ 著 [美] Jim McCarthy Michele McCarthy

译 赵 俐 石华耀

责任编辑 卢秀丽

◆ 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 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 12.75

字数: 260千字

2011年8月第1版

印数: 1—4 000册

2011年8月北京第1次印刷

著作权合同登记号 图字: 01-2008-1719号

ISBN 978-7-115-25260-9

定价: 45.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154



译者序

本书是软件工程领域的经典之作，它的第一版创作于 1995 年，到现在已经有 15 年历史了，你现在看到的是它的第二版（2006 版）。这一版包括 3 个部分，第一部分是原来第一版内容，第二部分介绍了人际关系准则和行为模式的系统，第三部分是作为多媒体补充的视频资料^①。值得注意的是，原来 1995 版内容大部分没有做修改，因为这些思想经过 10 余年的事实检验，被证明一直是正确的，作者对这部分内容只是增加了一些补充性的文字信息，并指出了这些思想的后续发展。我想 15 年前的大部分软件书籍可能都早已过时了吧，而这本书经过岁月的洗礼，依然熠熠生辉。虽然软件领域瞬息万变，技术革命的步伐也越来越快，但有些东西是永远不变的，就像牛顿三大定律一样。

把软件工程理论与软件开发实践结合到一起，生产力就会呈几何级增长。自从软件工程成为一门正式的学科以来，人们对软件工程理论的探索和研究就从未停止过，人们对于开发的认识也从单纯逐步走向成熟，并对它进行科学的分析和研究，进而实现人们把软件工程理论成功应用于软件开发项目的理想。本书所介绍的很多思想，特别是第二部分中的思想，将会给软件开发方式带来一场革命，也将会把软件开发推向一个全新的阶段。和历史上每一次超越的过程一样，只有依靠严谨的科学理论和不断的实践，才能够真正实现自我发展与超越，这也正是本书的主旨。

本书中有些经验法则是我们所熟知的，但大部分人可能只是下意识地使用它们，而没有形成一个清晰的认识，通过仔细阅读本书，你可以系统地掌握它们。有些经验法则则是我们以前从不知道的，例如有关人际竞争的 alpha 理论。这些新鲜的思想使我有所顿悟，相信读者也会同样受到启发。

翻译本书是一个巨大的挑战，毕竟，要在短短数月之间读懂作者多年职业生涯中在各个方面积累的思想、感悟和智慧并不容易。虽然我几乎放弃了所有的周末和节假日休息时间，但仍然感觉时间紧张而且压力巨大。这是一本博大精深的书，如果由于我的理解问题而致使没有完整地表达出作者的深邃思想，那将是我最大的过

^① 该视频资料详见 www.turingbook.com。——编者注

错和遗憾。因此，我全力以赴，尽最大努力保证翻译的准确，但由于水平有限，在翻译过程中难免会出现错误，恳请读者批评指正。在此我必须感谢我的同事为我分担了大量我份内的工作，使我能够腾出更多时间来翻译本书。最后感谢图灵各位编辑所付出的努力，感谢他们在翻译过程中给予的帮助和提出的宝贵意见。

2006 版前言

本书 1995 版的写作是一项非常困难的任务。虽然书很小，但却给我一种永远也写不完的感觉。不安的情绪贯穿整个写作过程，我不敢肯定是否能够把它完成，甚至怀疑我是否应该写它。书中大部分内容都是过去演讲过的。我对于演讲这种更直接的媒介应付自如。毕竟，演讲或发言是一个高度集中的交互式过程，在很短的时间内就可以完成，这种几个小时的压力是我完全能够承受的。此外，我付出的努力也得到了回报，那就是我的听众能够集中注意力听我演讲，而且听众会为我提供直接的、不断的反馈，时刻指引我把主要精力投入到哪里。

在写作期间，我一度后悔过，如果当时决定只发行一个视频录像就好了。但出于多种原因，我还是决定写这本书。微软出版社的朋友们支持我做出的这个决定，现在我有义务把承诺要写的书交给他们。整个写书的过程时断时续，我曾经逃避过，也曾经精心编制各种借口，但在我未婚妻 Michele 的全力支持下，经过了所有这一切，我终于把这本历时已久的书写完了。

事实上，我为本书写的最后一句话最终却出现在了书的前面——序言中。这句话就是：

当然，我希望本书中包含的线索能在未来的版本或后续著作中被扩展为一个更全面的框架。

因此，这一版的修订有两个目的。第一是文字上的修订，包括插入图片、扩充原来的经验法则，以及说明这些经验法则的进一步发展演变。第二个目的，也是我认为更重要的目的，就是（至少）宣告这一愿望，今后（或许）实现上述希望。

这一版分为 3 个部分，这里先简要说明一下，下面将会更详细地介绍。

第一部分是原来的 1995 版内容。

第二部分向本书的读者介绍了人际关系准则和行为模式的系统，我把这个系统称为核心（The Core）。核心体现了这些经验法则自出版以来 10 年间的运用结果。注意，我们还增加了以下经验法则。

经验法则 55：做完美的老板。

经验法则 56：老板就是你最重要的客户。

经验法则 57：支付木材税和下阿尔法赌注。

第三部分是专门为本书（第二版）提供的多媒体补充资料，包括以下内容。

- 1995 年做的一场演讲的视频——“及时交付优秀软件的二十三条半经验法则”（ $23\frac{1}{2}$ Rules of Thumb for Shipping Great Software on Time），这个演讲一直以来吸引了众多热情的支持者。
- McCarthy Show 的 4 段视频。

就第一部分而言，原来的 1995 版内容保留没变，仍是一个独立的短文集，描写了几个相关的微软软件开发小组在 Web 即将诞生之前的年代共同工作的经历，并以这些经历作为基础总结出了一些经验法则。虽然时光荏苒，但并没有人质疑过这些故事的准确性，先前得出的结论也并没有面临信任危机，而且对于我们从软件开发经历总结出的经验法则也没有人提出过令人信服的反对意见。因此，这些经验法则大部分没有做修改，它们基本上已被人们接受而且是令人满意的。我们顶住了编辑提出的修改要求，而只是增加了一些补充性的文字信息，并指出了这些思想的后续发展。

上面提过，本书起源于 20 世纪 90 年代我曾反复做过的一个演讲，后来它逐渐变得有一丝表演的味道了。我们很高兴把录得最好的一次演讲作为这一版的主要部分（也是本书原本该有的部分），以视频资料的形式提供给读者。演讲本身是在一些软件开发会议上做的研讨讲座，题目是“拖延而不失败”（Slipping Without Falling）。在那个年代，“进度落后”（即没能按计划完成）是软件开发的一个非常突出的现实，但人们却避而不谈。如何应对软件开发进度的极度不确定性是软件项目的核心问题，尽管人们并不常讨论它。我认为，承认这个事实并强调它具有典型性，是本书吸引读者的关键，也是本书获得成功的基础。虽然人们从不谈及如何应对进度落后，但这却是软件公司的核心活动现状。在一些好奇的人看来，整个行业都不愿意谈论成功所需的基本技能，确实是一个很有意思的问题。这种彻头彻尾的自我损害不可能永远不被注意到。

我相信我们的研讨会最有吸引力的地方就是大胆地指出了这个现实。经过了六七年后，“拖延而不失败”演变为“及时交付优秀软件的二十一条经验法则”。研讨会的很多参加者和演讲听众都建议应该出一本书。微软出版社采纳了这个建议，请我写一个演讲摘要，于是我就写了一篇短文 *21 Rules of Thumb for Creating Great Software on Time*。写完之后发给了微软出版社的编辑，后来就没有再想这件事。再后来，我发现这篇短文像某种“Web 地下出版物”一样传播开来（我不知该归功于

谁还是归罪于谁)。甚至在今天,你在 Google 上搜索一下“21 Rules of Thumb”,仍能找到数千条引用和链接。

为了写一本完整的书,我把 21 条经验法则扩充为 54 条,这就是本书第一版 *Dynamics of Software Development*,也就是这一版的第一部分。

在这一版的第二部分中,我们试图为读者呈现一个我们认为更全面的框架,这是自第一版出版 10 年来我们一直希望的,也是一直在努力追求的。我们把这个“框架”称为“核心”。然而,就像任何有着非凡目标的软件一样,这个“框架”的最终形式有点难以理解。我们在后续工作中,特别是在“核心准则”中,交叉引用了第一版中的很多思想,显示了这些思想在演进,也表明了我们対原有观点的坚持。

坚持的意义往往要等到事后才会明白,在此之前,这常常令人感觉有点愚蠢,而且很容易被人误解为固执己见。不管怎样,在这些年中,我一直念念不忘当初创作本书第一版时所怀的希望,而且我也越来越坚信我们正在真真正正兑现当时的承诺。

上一版结稿时,我得到了 Michele Frame 的大力协助,她是微软的一位模范项目经理。当时她对这些思想的支持为我提供了必不可少的动力,使我终于能够推出本书。就在第一版出版后不久,我们一起离开了微软,更积极和坚持不懈地去尝试“更全面的框架”。

我们认识到,软件开发本身实际上只不过是人类协作的一个案例分析罢了,如果我们能够找到一种途径使任何团队都能够可靠地形成共同前景,那么我们就可以驯服软件开发这头猛兽,把所有人的力量汇集到一起,从而缔造世界的文明。遗憾的是,在企业环境中无法足够快地实现这一点。任何软件经理或软件团队也都无法在一个项目的生命周期内进行足够多的开发周期和实验来找出这种途径。

我们开始在一个为期一周的课程(我们把它叫做“训练营”)中让真实世界的团队来模拟开发周期。尽管我们有许多“思路”可以教给学员,但我们只是要求他们开发出一个单一的、可重复的过程,通过这个过程就可以造就一个具有共同愿景的成功团队。我们一向以胆大著称,给前来学习的团队只布置一项简单的任务:“设计、实现和交付一个过程。它将教给你(每次)按时推出优秀软件所需知道的一切。”

这是参加模拟的团队必须要在一周内完成的产品。这使得这一周的学习非常紧张,因为如果没有交付产品,那么就有可能被判定为不及格。过去 10 年来,这项任务一直没有太大的变化,每次课程结束时都会出现更多协作方面的“最佳实践”。我们从每个团队纷乱的工作中把这些最佳实践提炼出来,利用训练营的间隙在我们自己的团队中实验它们,然后再把它们提供给下一个团队,让他们以之为起点继续探索。

最后我们得出了一个神奇的结论，那就是软件开发和其他复杂协作活动的解决方案（实际上）是一种新型的软件，这是一组代码化、标准化、非个性化且版本化的人际准则、定义和行为模式，我们把它命名为“核心准则”。

或许我们本来不应该对软件开发动力最终会产生软件这个结果感到惊奇，但这确实让我们吃惊了。这个软件及其使用就是本书第二部分的写作背景。核心准则 1.0 版的细节及其相关思想和历史在 *Software for Your Head* (Addison-Wesley, 2002) 一书中做了介绍。

本书首次介绍了核心准则 3.0 版。

我们相信你（至少）会看到我们所期望的框架的轮廓。我们一直的宏大理想是在它的后续版本中提供更鲜明、优美和实用的框架。

Jim McCarthy
Michele McCarthy
2006 年 7 月

系 统 需 求

我们推荐使用 Windows Media Player 观看视频。以下是使用 Windows Media Player 10 的最低硬件需求。

组 件	要 求
操作系统	Microsoft Windows XP Home Edition、Windows XP Professional、Windows XP Media Center Edition或 Windows XP Tablet PC Edition
处理器	233 MHz处理器，例如Intel奔腾 II 或AMD处理器
RAM	64 MB
空闲磁盘空间	100 MB
光驱	CD或DVD驱动器
调制解调器	28.8 Kbps
声卡	16位声卡
监视器	Super VGA分辨率 (800×600)
显卡	带有64 MB RAM和支持DirectX的显卡
声音输出设备	扬声器或耳机
Internet浏览器	Microsoft Internet Explorer 6或Netscape 7.1

本书的支持信息

我们已尽一切努力确保本书和配套视频资料内容的准确性。微软出版社为本书及其配套视频资料提供支持，网址为 <http://www.microsoft.com/learning/support/books/>。

问题和评论

如果你对本书及其配套视频资料有任何评论、问题和想法，或者有任何问题在以上网址没有找到答案，请把问题寄给微软出版社。

电子邮件地址：mspinput@microsoft.com

通信地址：

Microsoft Press

Attn: Dynamics of Software Development Editor

One Microsoft Way

Redmond, WA 98052-6399

请注意，以上地址不提供微软产品支持。

第一版序

你想知道从本书中能够学到什么吗？不妨想象一下达芬奇、汤姆·彼得斯^①和罗伯特·傅刚^②合著了一本讲软件开发的书，你就会知道能够学到什么了。

Jim 的这本书富有远见、生动活泼且极为实用，读懂了这本书，保证你几乎每次都能按时开发出伟大的软件，甚至是划时代的软件。更重要的是，Jim 的书还讲述了如何培养出伟大的软件团队。

Jim McCarthy 是我的搭档。我们同在一个团队中工作，为微软设计和开发软件。我们开发的软件 Visual C++，如果用微软内部的标准来评判，算是略有小成，但用外人的标准来看，则是获得了巨大的成功。自从两年前发布 1.0 版本以来，已售出 100 多万份。

但本书并不是讲 Visual C++ 的，也不是讲微软的。本书讲的是我们碰巧在微软工作而又碰巧开发 Visual C++ 时所获得的经验教训。

即使是在强手如林的微软，我们的团队也凭借按时交付（甚至是超前交付）的良好声誉闯出了不小的名头。虽然在某些情况下项目逾期已成为惯例，有时甚至拖上好几年的时间人们都认为没什么，但我们一直能够按约定交付软件。这在几年前是难以想象的，即使是现在，99 % 的软件团队也认为这是不可思议的。几年前，我们的产品发布周期约为 12~24 个月，现在已缩短至 4 个月。

我以个人名义保证，如果你的工作与软件开发有关，哪怕是只有一点点关联，你也会在 Jim 的故事中找到你自己的影子，并领悟他的经验法则所表现出来的智慧。Jim 的洞察力和建议如此敏锐、如此恰当、如此准确，你会发现他着实精通他所讨论的内容。

我还保证没人会指责 Jim 采用了一种学术（或者说正式的）方法来讨论软件开发或管理。本书来源于亲身体验，来源于艰苦的磨炼，同时也来源于我们的一个愿望——我们再也不想在下一次软件评审中失败。

① 美国著名管理学家、管理类畅销书作家，被誉为“商界教皇”。——译者注

② 美国著名作家，演讲家，出版过多部畅销书。——译者注

本书是从 Jim 的一个讲演“交付优秀软件的 21 条经验法则”发展而来。Jim 在世界各地做这个巡回演讲，所到之处可谓是座无虚席。成千上万人听着、笑着，发现他所讲的内容与自己密切相关，对 Jim 用来构建优秀软件和卓越软件团队的简单却意义深远的方法抱以赞同的掌声。

Jim 的经验法则（一如其本人）随着时间的推移不断发展和完善，始终紧紧围绕着我们所深爱的这个行业。这个行业混沌无序、不断变化、广阔无边。此外，这个行业还非常有趣。比如说吧，你会碰到一些像 Jim McCarthy 这样非常有意思的人。

我也正在努力超越自我。

我第一次见到 Jim McCarthy 的时候，他只有一条经验法则：虽跌倒，别趴下。至少这是 1992 年我面试他的时候他告诉我的。当时，我刚刚就任微软系统语言小组（现在这个部门已经不存在了）的工程部经理。Jim 来自 Whitewater 公司，先前做过一种名为 Actor 的早期面向对象编程语言的开发工作。这次面试我印象最深的是 Jim 穿着一双牛仔靴，当时我想这是我所经历过的唯一一次面试官和被面试者都穿着牛仔靴的面试。这一定是一个预兆。

然而，我们并没有仅仅因为 Jim 穿了牛仔靴而决定聘用他，也不是因为他的那条经验法则。回想起来，我不得不说我们是因为他口袋里的另外一份清单而雇佣了他，那份清单的标题是“你应该雇用 Jim McCarthy 的十大理由”。

这就是 Jim 干的事儿。什么他都喜欢列出一二三条。他列出他的教训、他的经验和有用的事情。多年以来，Jim 所开创的经验法则不断成长，就像杰克的魔豆^①一样，最后终于成就了微软出版社出版的这本书，也成就了软件领域最受欢迎的主题演讲。

Jim 和我最近几年共同经历了一些令人难忘的冒险活动，本书会与你分享其中一部分，希望能对你有所帮助。

Denis Gilbert
Visual C++ 首席讲师
微软公司
1995 年 6 月

① 《杰克与魔豆》（也译作《杰克与仙豆》），英国童话故事。讲的是杰克用自己家的奶牛换来了一瓶魔幻豆，妈妈很生气，但是杰克通过它换来了很多奇特的经历和宝贵的财富。——译者注

第一版前言

在一个已郑重承诺的长期软件开发项目中，会有一些动力因素互相作用，本书就是尝试描述其中的一些因素。这些动因都是在我亲身经历的项目中出现的，我甚至可以说它们是项目中的决定性因素。正如你可能会想到的，我对这些动因的观察和解释也在随着时间而演变，而且还会继续演变。

用一本书把这些真实的历史事件讲述出来难免会产生一个副作用，那就是本书提到的每个人对这些事件的记忆和解释可能会各不相同。本书只是表达了我的观点，书中的重点是我切身感受到的有意义且重要的经历。在有些地方我改变了一些不重要的细节，要么是为了保护个人隐私使用了化名，要么是为了突出那些我认为特别重要的内容。

人们对任何事件或一连串事件的解释肯定（至少可能）会随着时间的推移和经验的积累而变得深入和成熟。本书历时一年半，我都是利用晚上和周末的时间来写作。在这段时间内，我仍在不断学习软件开发。可以肯定的是，书中讨论的一些动因和实践，得益于我的深入学习以及开始写书时已经或将要获得的更成熟的观点。毕竟，软件的年头就像“狗龄”^①一样——人的一年等于软件的 n 年，其中 n 与人在一年中要实现（或试图实现）的开发里程碑的数目成比例。在我写作期间，有很多软件已经创建完成，等于又过去了很多软件年。

经过与足够多的开发人员和组织讨论，我相信书中的分析论述是很多正在开发商业软件和内部软件的个人和团队很熟悉的。我希望我的思想、观点和经验法则有助于降低软件开发不确定性导致的风险和带来的恐惧。我还希望这些内容可以成为一个框架，从而帮助人们理解什么是软件开发的常态。在软件开发工作中，如果没有一个标准的、广为接受的软件开发实践，我们注定永远无法分清在软件团队的驱动因素中，哪些是健康的，哪些是病态的，而且这种状况无药可救。

因此，本书的基本主题就是讲述在特定的时期里，一批优秀人员组成的成功团队在全力地多次按时交付优秀软件的过程中，是如何体验、阐释甚至（通过反馈）在相当程度上管理好这些软件开发的动因的。我相信本书所描述的见解和实践对那

^① 西方有种说法认为人的1年等于狗的7年。——译者注

些我有幸合作过的项目团队的成功做出了重大贡献。我也相信今后再回过头来反思所有这些内容时，一定会看到本书的一些不足之处，这是由于我们原先的技术水平所致，而并非我们判断错误。

你可以把本书看做一部纪录片。你可以说它记录了软件开发或某个软件开发团队成长过程中的一个真实阶段，也可以说它忠实地记录了这个行业奇特的组织行为和分析结果。

当然，我希望本书中包含的线索能在未来的版本或后续著作中被扩展为一个更全面的框架。

Jim McCarthy
1995 年于华盛顿州雷德蒙德市

致 谢

首先感谢 1991~1995 年在微软的 AFX^①和 Visual C++ 事业部工作的人们，本书中的很多思想都是从他们那里“挖”来的。

还要感谢原来在 Whitewater 集团工作的 Mark Achler 和 Chuck Duff，以及原来在贝尔实验室工作的 Bob Gianni 和 Tom Collura，他们对我的软件开发生涯产生了很大影响。

在此对第一版的编辑 Erin O'Connor 表示我的敬意和感激，感谢她的远见、贡献和耐心。

Patrick McCarthy 为本书绘制了绝妙的插图，用这样的插图来表达软件开发文化和信息文化我还是第一次看到。当我的文字未能尽意时，他的丹青之笔或许能更传神。

2006 版补充致谢

我如何才能正确评价一门技术？有没有一种有意义的方式能让我表达出对一场新文化运动的感谢？我能够脱帽向一个席卷全球、海纳百川的行业致敬吗？怎样才能表达出我的感激之情？我是否能够为软件贴上一个价签，标示出这个并非活物但却生机勃勃、汹涌澎湃、生生不息且不断演进的事物价值几何？或许最终我能，但现在却不能。那么，我如何才能有效地表达出我对一个时代的万分崇敬之情？怎样准确地形容我生活的这个时代？就软件本身而言，谁值得感谢？

如果我在这里说，是软件让我实现了年轻时的梦想，让我的生活充满了创造的乐趣，那么这是否稍稍表达出了我饮水思源的感激之情？至少我知道我说的是事实，可以令我心稍安，但不管这是多么真切的事实，即使它引起了部分读者的共鸣，最终也不能表达出我的万分崇敬之心。

如果我在这里提及几位巨人——比尔·盖茨、史蒂夫·乔布斯、史蒂夫·鲍尔

① AFX 是微软 MFC (Microsoft Foundation Class, 微软基础类库) 开发小组早期的代号。AFX 指的是 Application Framework Extensions。——编者注

默，他们在软件冲击波到来伊始就彻底为之折服，从而建立了现在令我感到惬意的软件世界，但他们在软件或事业狂潮中又从未屈服，那么这是否就充分表达了我的谢意呢？

又或者，我是否还应该再把感谢之辞送给那些我认为的真正奠基者——新泽西州默里山贝尔实验室的柯尼汉（Kernighan）、里奇（Richie）和汤普森（Thompson），伯克利和 Sun 的乔伊（Joy），PARC 的凯（Kay）、戈德堡（Goldberg）、梅特卡夫（Metcalf）及其他人，麻省理工学院的斯托曼（Stallman）、约翰·麦卡锡（John McCarthy）和高德纳（Knuth）。当然，此处一笔带过是根本不足以感谢他们所缔造的世界的。

还有我的前辈们，弗雷德·布鲁克斯、汤姆·迪马科和埃德·尤顿，我的感谢之辞对他们来说微不足道，他们应该得到更多的致敬。

但最重要的是，感谢产品开发训练营的数千位参与者，以及他们的赞助单位，在此表示我最衷心的赞誉和最深的谢意。

我把续写的本书这一版作为一个见证，献给上述所有这些人。我希望它可以使软件世界中的人工作更轻松，成绩更出色。

最后还要感谢微软出版社的以下人员：Ben Ryan，他使得本书再次焕发勃勃生机；还有 Devon Musgrave，感谢他的远见和为这一版所做的编辑工作。

目 录

第一部分 1995

简介	2
有关软件的思考	3
项目的阶段	6
开局阶段	7
组织	7
质保人员是“少数民族”吗	8
到底谁负责产品设计	8
经验法则 1 建立共同前景	9
经验法则 2 使大家主动投入	18
经验法则 3 制定多版本的技术计划	20
经验法则 4 不要认为别人是笨蛋	25
死亡行军	27
经验法则 5 搜集情报	30
经验法则 6 注意团队成员组成的比例	32
经验法则 7 组建功能小组	33
经验法则 8 项目经理的重要性	41
团队精神	42
经验法则 9 做权威,而非掌权者	44
竞争	47
人类学分析	47
软件竞争	49
经验法则 10 缺乏竞争对手?未必是好事	50
经验法则 11 与竞争对手不相上下?进行功能竞赛	52
经验法则 12 落后于竞争对手?更频繁地推出新版本	52
经验法则 13 领先于竞争对手?绝不放松	55
经验法则 14 紧跟潮流	55