

软件项目管理系列丛书



Rapid Development

Taming Wild Software Schedules

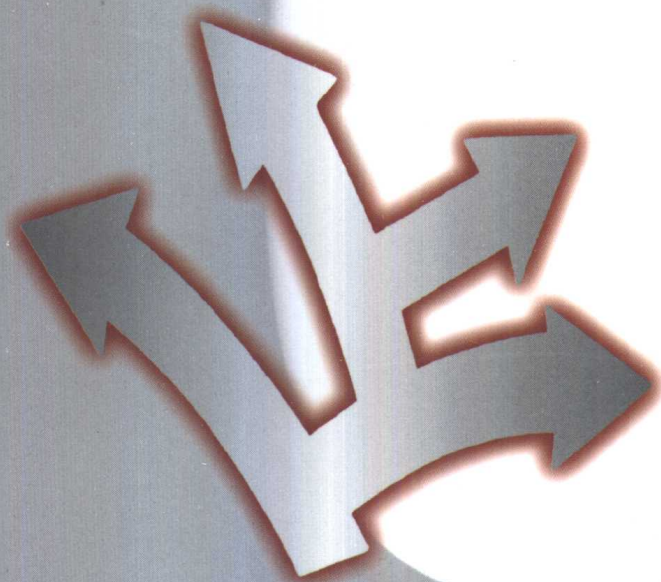
快速软件开发

有效控制与完成进度计划

(美) 斯蒂夫·迈克康奈尔 著

席相霖 等 译

冯炳根 王利文 审校



Microsoft® Press



电子工业出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

TP311.52

24

件项目管理系列丛书

Rapid Development

Taming Wild Software Schedules

快速软件开发

有效控制与完成进度计划

(美) 斯蒂夫·迈克康奈尔 著

席相霖 等译

冯炳根 王利文 审校

北方工业大学图书馆



00501889

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

提高软件开发的速度,按进度计划完成项目,是软件开发项目管理中最常见和最难解决的问题。本书在总结了包括微软公司在内的美国软件业成千上万个软件开发项目的实践经验、研究成果、经验教训的基础上,详细列出了几十种经实践证明可以直接在软件开发中应用,以提高开发速度的最佳实践方法、开发策略、实用技巧等,帮助开发人员和项目经理在了解软件开发中最常见错误的基础上,根据自身实际情况,制定出满足项目进度、成本、质量与其他目标要求的最佳方案。

本书获得美国Jolt卓越软件开发图书奖,被誉为软件开发最好的实践指南。特别适合技术领导、软件开发人员、软件项目经理、软件企业管理人员阅读。

Original English language edition Copyrights ©1996 by Steve McConnell.

All rights published by arrangement with the original publisher, Microsoft Press, A division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体字版由电子工业出版社出版,未经出版者书面许可,本书任何部分,不得以任何方式复制或抄袭。

版权所有,翻印必究。

图书在版编目(CIP)数据

快速软件开发——有效控制与完成进度计划/(美)迈克康奈尔(McConnell, S.)著;席相霖等译. —北京:电子工业出版社, 2002.1

(软件项目管理系列丛书)

书名原文: Rapid Development

ISBN 7-5053-7285-8

I. 快… II. ①迈… ②席… III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字(2001)第 086818 号

丛 书 名: 软件项目管理系列丛书

书 名: 快速软件开发——有效控制与完成进度计划

著 者: (美)斯蒂夫·迈克康奈尔

译 者: 席相霖 等

策划编辑: 傅豫波

排版制作: 电子工业出版社计算机排版室监制

印 刷 者: 北京市增富印刷有限责任公司

装 订 者: 三河市金马印装有限公司

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 30.5 字数: 660 千字

版 次: 2002 年 1 月第 1 版 2002 年 2 月第 2 次印刷

书 号: ISBN 7-5053-7285-8
TP·4176

定 价: 58.00 元

版权贸易合同登记号 图字: 01-2001-4207

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;

若书店售缺,请与本社发行部联系调换。电话 68279077

向着软件产业国际化迈进

——《软件项目管理系列丛书》总序

软件产业是一个朝阳的、新兴的知识型产业。一个国家软件业的发达程度，也体现了国家的综合国力，决定着国家未来的国际竞争地位。

目前，中国的软件企业正处于高速发展、急需规范管理并以项目为主导的环境中。企业每天所面对的不仅仅是几个越来越大的大型项目，而将是成百上千不断发生和进行的项目。产生这种变化的因素是多方面的，这包括客户需求的不断提高导致产品生命周期缩短；产品开发项目数量大增；新技术导致了对研究和开发项目需求的增加；为了提高业务赢利能力，改进业务模式的项目需求大增等。在这种多项目并发、技术含量高、变化速度快、资源有限的环境下，如何对企业、项目、资源实施科学的管理，加强团队能力，实现软件企业的生产规模化、规范化、国际化，是当前我国软件业面临的最大挑战。

一些调查表明，大约70%的软件开发项目超出了估算的时间，大型项目平均超出计划交付时间20%到50%，90%以上的软件项目开发费用超出预算，并且项目越大，超出项目计划的程度越高。

在中国成为WTO成员的今天，如何科学地管理企业，激励开发人员，提高软件开发的生产率，按时、按预算提供满足客户需求、具有国际市场竞争力的软件产品，一直困扰着软件企业的管理者，同时也成为阻碍中国软件业向国际化发展的绊脚石。

中国目前软件业的落后状况，实际已经不是技术本身的问题，而是科学管理的问题，软件工程理论与软件项目管理的实际应用与经验，是我国软件企业急待学习与渴求的。中国软件业各个层次上的管理问题，目前可以说已经制约了中国

软件业走向国际化的进程。

电子工业出版社针对以上现状，组织出版了“软件项目管理系列丛书”。该套丛书针对如何有效进行软件企业管理、软件项目管理、软件质量管理这一主题，以软件企业CMM的实现与软件开发项目管理的问题为核心展开，围绕软件业中的技术、过程、产品、人四个维度进行讨论。该套丛书的出版，为中国的软件企业提供了科学的管理方法与技巧，特别是贯穿整套丛书中的国外软件企业和软件项目的管理实施方法、管理经验与教训、成功与失败的案例，对帮助中国的软件企业实施科学的管理，把项目变为可以控制，从而以更短的时间、更高的质量、可预测的成本生产功能更为丰富的软件产品，具有重要的作用。

该套丛书的内容独特、实用，多数均为世界软件开发领域的经典名著，是国际上软件项目管理大家智慧的结晶，代表了软件大国的先进管理理念与水平，可以说是一套很好的软件业“MBA”丛书。该套丛书的出版，对推动中国软件业向世界软件大国迈进，具有重要的意义，也希望中国软件企业，能够从该套丛书中，学习国外先进的管理经验，尽快成为具有国际管理与生产水平的软件企业，为使中国尽快成为世界软件大国而努力，为推进中国信息化建设做出贡献。

陈冲

信息产业部电子信息产品管理司副司长

中国软件行业协会副理事长

2001年12月12日



译者序

斯蒂夫·迈克康奈尔（McConnell）撰写的《快速软件开发》一书中文译本与中国读者见面了，这实为一件值得庆幸的事。

随着信息技术的飞速发展，我国软件开发公司的数量在急剧增加，软件开发队伍成倍扩大，软件开发项目不仅越来越多，而且规模也越来越大。但是长期以来，软件开发人员一直被估算不准、功能蔓延、团队人员流失等问题所困扰，巨大的进度压力、突然而至的风险，总是使这支队伍在成功地完成项目的征途中步履艰难。无论是公司内部还是商业的软件开发团队，都希望尽早地解决这些重要的问题，能够有效地控制他们高度压力下的进度计划。近年来，我的工作方向转移到项目管理方面以后，也非常想以现代项目管理的知识体系和工具、方法来进行总结，力图找到一些解决这些问题的途径，但一直感到头绪不够清晰。

今年盛夏，电子工业出版社送来了斯蒂夫·迈克康奈尔撰写的《快速软件开发》这本书，征求我的意见是否将其译为中文提供给中国的读者。我用3天的时间从头至之尾逐章地阅读了这本书，收获颇丰。

斯蒂夫·迈克康奈尔是微软和其他许多公司的一位经验丰富的软件顾问，他最初的方向是开发分布式商业微机软件，在5年中编写了5万行产品代码，积累了大量的实践经验。与此同时，他也进行了很多的理论研究，撰写了许多论文和书籍。《快速软件开发》就是其佳作之一。

在本书中，他从正面提出了完整的软件开发战略，详细地阐述最佳实践，提供了有价值的运作技巧，这些都对项目开发人员缩短和控制开发的进度，保持项目健康向前推进提供了极大的帮助。

本书分为三部分，共43章。

在第一部分，他列举了软件开发中经常犯的36种错误，分析了常遇到的120多种风险，论述了实现快速开发的基本原则和战略。

第二部分，则以清醒的头脑逐个专题地阐述了软件开发中的几个核心问题，包括如何估算、做原型、编制现实的计划、激励、团队工作、快速开发语言、控制需求蔓延等的原则、技术和方法。

第三部分中，他向读者推荐了有利于快速开发的27种行之有效的最佳的实践，例如，变更控制委员会、选择恰当的生命周期模型、有原则地谈判、W理论等等，对于每条建议都对其潜在的效率和风险进行了恰如其分的描述和分析。

在全书的每一章中，作者都向读者提供了失败的案例和成功的案例，用不可辩驳的数据进行证明，以富有洞察力的趣事进行点缀。

这是一本为团队领导、项目经理和程序人员撰写的一本好书，是使应用开发更高效、更为现实的工作指南。

有10人参加了本书的初译工作，承担的工作分别是：丁丽翻译第12~13章，朱莉翻译第19章，李华翻译第15~16章，陈新翻译第7章，陈蔚力翻译第8章，林鄂华翻译第17~18章，高福春翻译前言和第1~5章，韩梅翻译第9~11章，韩蓬翻译20~43章，黎纓翻译第6、14章。

冯炳根教授和王利文副教授对于全部译文进行了精心地校对，王老师校对了第一、二部分（第1~16章），冯教授校对了第三部分（第17~43章）。

在翻译的整个过程中，电子工业出版社傅豫波副总编给予了指导，并亲自参与审校，北京中科项目管理研究所给予了项目管理方面的帮助，高箐华参与了录入工作，杜端甫教授、许成绩、沈天阳、王星、徐永成、毕素缘等也提供了其他方面的帮助。本书翻译的完成是大家共同努力的结果，在此向参与该书翻译、校对和提供支持和关注的所有人员一并致谢。

书中释译中如有不当之处，恳请读者批评指正。

席相霖
中科项目管理研究所 所长
2002年1月

前 言

软件开发人员基本上处于进退两难的境地中，一方面他们为解决开发中所碰到的各种问题，工作太刻苦，挤不出时间钻研有效的实用技术；另一方面，如果他们不学习掌握软件快速开发的方法，他们就永远不会有足够的时间。

摆在他们面前的问题是，在“尽快交工”计划的压力下，如何在开发进度与软件质量之间达到最理想的平衡。如果开发人员需要放弃看电影、读报纸、购物、休闲、锄草或与孩子玩耍的所有时间，连续工作20天才能按计划完成开发项目，那么，怎么可能指望他们投入很多精力研究软件可用性方面的问题呢？也就是说，除非我们把对项目进度的控制作为软件从业人员的必修课程，并为开发人员和经理们留出学习这方面专业知识的时间，否则，对开发人员来说，将会很难有足够的时间进行有关方面知识的学习。

软件开发时间的问题普遍存在。一些调查表明，2/3的项目超出了估算的时间（Lederer and Prasad 1992, Gibbs 1994, Standish Group 1994）。大型项目平均超出计划交付时间的20%到50%，项目越大，超出计划的时间越长（Jones 1994）。一直以来，开发速度的问题都是软件开发业必须解决的头等问题（Symons 1991）。

虽然软件开发速度缓慢的现象普遍存在，但有些机构还是在进行快速的开发。调查人员发现，同一行业的两家公司生产效率的差别有可能为10:1，甚至更大（Jones 1994）。

本书的目的在于为当前是“10:1”中“1”的一方提供所需的信息，帮助他们向“10”的一方转移。本书将帮助你把项目变为可以控制，从而以更短的时间向用户交付功能更为丰富的产品。

在阅读本书时，你可以只看你感兴趣的内容，而无须阅读整本书，不管你的项目处于何种阶段，你都会在书中找到能够帮助你改善当时境况的实用内容。

本书适用对象

慢速的开发会对软件开发中的每个人造成影响，包括开发人员、项目经理、项目委托人和软件的最终用户，甚至包括其家庭和朋友，每个项目组在解决开发速度缓慢的问题时都有其各自的困难，本书将对常见难点逐一加以讨论。

本书旨在帮助开发人员和项目经理理解什么是可行的，帮助经理和用户认识哪些是可实现的，同时也讲述了开发人员、项目经理和用户之间可行的沟通方式与方法，从而使得他们能够共同找到一条最佳的途径以满足项目计划、成本、质量与其他目标的要求。

1. 技术领导

本书主要为技术领导或项目经理而编写，如果你是这样的角色，你可能经常要为提高软件开发速度承担主要责任，本书将告诉你如何去做，同时也描述了开发速度的限制，这将为你准确区分可实现过程与想象过程打下坚实基础。

本书中有些实用方法完全是技术性的，作为技术领导，学习并实施这些方法你应该没有问题，而另外一些实用方法则更多的是基于管理方面的考虑。可能你会问：“此处为什么包括这些内容？”在编写本书时，我做了这样一个简单的假设，即你是一个超级技术领导，你比快速的黑客还快，比疯狂的经理还强，能够同时解决技术问题与管理问题。我知道，有时这可能并不现实，但做这样的假设，可以让我专注地讨论中心的议题，不必分心去描述“如果你是经理，这样做，如果你是开发人员，那样做。”我做的另一个假设是，技术领导同时肩负技术与管理工作，而不仅像字面想像的那样。技术领导经常肩负着为高层领导提供技术建议的职责，本书将帮助他们更好地完成这方面的工作。

2. 程序员

很多软件项目都是由程序员个人或自行管理的项目组来承担的，事实上，这就将参与项目的技术人员推到了技术领导的角色上。如果你是处于这一角色，那么本书将帮助你提高开发速度，基于同样的原因，也可以帮助你成为一个好的技术领导。

3. 项目经理

项目经理有时认为实现快速软件开发主要是技术工作。然而，作为一个项目经理，常常也会像开发人员一样，在改善软件开发速度方面有很多事情要做。本书将描述许多管理层面上的快速开发实用方法，当然，你也可以阅读技术方面的实用方法，以便更好地理解开发人员所做的一切。

本书的主要特色

我是以“软件开发人员的直觉”，围绕“为什么我们常见的快速开发方法都基本失败了”这一中心来构思本书的。书中讲述的所有实践活动都是特定环境下开发人员实际工作的真实写照，正是这个原因，本书提倡在学习使用书中的方法时，应根据自身情况，自己做一些小小的变革。

我个人对于软件开发的观点是，软件项目可以基于多种目的进行优化，如基于最低错误率、基于最快执行速度、基于最佳用户接受度、基于最佳维护性、基于最低的成本、基于最短开发周期等等。软件工程方法的一个主要目的就是要平衡得失：你能够通过降低质量要求、降低可用性要求、开发人员超时工作等手段优化开发时间吗？当项目结束的时间逼近时，你最终要压缩哪些环节？本书将回答这些关键性的权衡问题及其他一些问题。

1. 立刻用于提高开发速度

可以在你特定的项目中采用本书所描述的策略和最佳实践方法，尽可能地提高开发速度。通常，大多数人都能够通过采用本书中的策略和实践方法使开发速度大大改善。我可以说，对任何软件项目，你总会在书中找到若干适用的方法。根据你的项目情况，“最佳开发速度”有可能并不像你期望的那样快，这不仅仅是运气，也与你没有使用快速开发语言，或还在使用过时的代码，或工作在不利于生产的嘈杂环境中有关。

2. 快速软件开发对传统理论的倾斜

本书中描述的有些方法并不属于典型的快速开发方法，如风险管理、软件开发原理和生命期计划，这些通常被看做典型的“好的软件开发方法”，而非快速开发方法的范畴。然而，这些方法具有深刻的开发速度内涵，许多情况下，也称它们属于快速开发这一主题。本书将这些方法在提高开发速度方面的实践，纳入到了与此相关的其他一些快速开发的实践方法中。

3. 实践方法的重点

对有些人而言，“实践”意味着“编码”，对这些人，我必须承认本书并不“很实践”。我避免基于编码的“实践方法”主要有两个方面的原因：第一，我已经写过一本800页的有关编码实践方法与技巧的书籍（Code Complete, Microsoft Press, 1993），有关编码，我已没有更多要说的了；第二，许多有关快速开发的关键要素并非基于编码，而是一种策略和理念，有时，更是一种实践而非理论。

4. 快速阅读结构

我尽可能以更为实用的方式来组织本书的快速开发资料。本书的前两部分描述了快速开发的策略和理念，其中有约50页左右的案例讨论，所以你可以清楚地看到策略和理念在实践中的作用。案例以明显的形式排版，如果你不喜欢这些案例研究，可以很方便地跳过。本书的其余部分是由一系列的快速开发实践构成，这些实践也以便于快速查找的方式编排，你可以根据自己项目的需要，很方便地找到感兴趣的内容。本书描述了怎样使用这些实践中的方法，使用该方法使计划进度缩短的程度，以及伴随而来的风险。

我在书中也采用了一些图标和特殊文字，用来帮助你快速找到与你目前阅读内容有关的其他信息，指出应避免的典型错误，最佳实践中容易忽视的盲点，或查找书中提到的定量的支持信息。

5. 有关快速软件开发的新思路

在软件开发领域，快速开发方面的谈论颇多。最近许多毫无价值的开发方法都打着“快速开发实践”的招牌，开发人员对这些所谓的“开发实践”非常气愤。尽管有些“实践”也还是有用的，但它们真正的作用被开发人员的冷嘲热讽所掩盖。

每种工具的提供者和每种方法的倡导者都想让人相信，他们的工具与方法可以满足你在开发中的所有需要。而本书的目的，就是指导你正确分析快速开

发方面的各种资源，就像要将小麦从谷壳中分离出来一样，帮助你找到快速开发的真谛。

本书提供了一个可操作的模型，该模型可以帮助你客观地分析快速开发的工具，以及决定怎样将其为你所用。当一个人来到你的办公室说：“我刚从GigaCorp公司获知，一种功能强大的新型工具可以将我们进行软件开发的时间缩减80%！”，你可能想知道这时你应做何反应。我无须讲述任何有关GigaCorp公司和它们新型工具的内容，当你读完本书的时候，你就会知道这种时候应该提什么样的问题，应该如何一分为二地判断GigaCorp公司这种说法的可信程度，以及如果你决定采用这样的工具，应该怎样将它们集成到你的开发环境中去。

与其他的快速开发书籍不同，我并不认为所有的鸡蛋都可以放到同样尺寸的篮子里。不同的项目具有不同的需求，而一种方法很难解决项目计划中的所有问题。我一直以不能说是刻薄，也至少是挑剔的眼光看待这些快速开发实践的有效性，并一再假设这些实践不能很好地工作。书中我也再访了一些曾大肆宣传的实践方法与工具，即便它们没有以前宣传的那样有用，但对其真正有用部分还是应该倡导的。

6. 为什么本书会有如此之厚

信息系统、商业封装软件、军事应用以及软件工程领域的开发人员都各自发现了一些有价值的快速开发实践方法，但不同领域的人员之间可能很难彼此交流经验。本书从每个领域收集最有价值的实践活动，很多快速开发的资料是首次汇总在了一起。

是否每个想了解快速开发的人都有时间读完这几百页的书呢？可能性很小，大多是读到一半的时候可能就不得不简化那些无关紧要的部分。作为补救措施，我将本书结构组织成便于快速阅读与选择性阅读的形式，在旅行途中或等车期间，也可以只翻阅一下简短的摘录部分。第1章和第2章包含有理解快速开发产品所必须的一些内容，读完这些章节后，你就可以任意选择自己感兴趣部分来阅读。

为何编写本书

项目客户和项目经理对慢速开发问题的第一个反应经常是加大项目计划进度的压力，让开发人员超时工作。有75%的大型项目和将近100%的超大型项目计划进度压力过重（Jones1994），将近60%的开发人员说他们感到工作压力在增加（Glass1994c），美国的开发人员每周工作时间在48~50小时（Krantz1995），甚至更多，许多人认为工作负担过重。

在这样的环境中，软件开发人员的总体工作满意度在过去的15年中大幅度下降就不足为奇了（Zawacki 1993）。项目的进度计划依靠对开发人员工作的拼命挤压来完成，导致开发人员工作负担过重，他们很自然会告诉他们的朋友与家人说，这一领域毫无乐趣可言。



显然这一领域还是有乐趣的，我们大多数人以前都从事过这样的工作，而我们并不苟同编写软件只是为获得报酬的说法。当然，在开发过程中的讨论会上确实会有一些不愉快的事情发生，这些不愉快大多会与快速开发的话题密切相关。

是该在软件开发人员与项目进度这个海洋中设置一道堤坝了，本书中我试图树立一个堤坝标杆，确保大海那边的狂潮不致打乱开发人员的正常生活。

致谢

首先，衷心感谢本书的项目编辑Jack Litewka，感谢他在本书编辑出版过程中提出的建设性意见。同时也感谢Peggy Herman和Kim Eggleston为本书所做的精美设计，Michael Victor的图表、Mark Monlux的插图与说明。感谢Sally Brunsman, David Clark, Susanne Freet, Dean Holmes, Wendy Maier和Heidi Saastamoinen对本项目的顺利运作所做出的贡献。还有很多朋友为本书的出版做出了各种各样的努力，有些人没有直接打过交道，但我也衷心向他们表示感谢（主要有艺术家Jeanie McGivern, ArtSource的生产经理Jean Trenary和微软出版社排版及校样管理员Brenda Morris, Richard Carey, Roger LeBlanc, Patrick Forgette, Ron Drummond, Patricia Masserman, Paula Thurman, Jocelyn Elliott, Deborah Long和Devon Musgrave）。

通过对数以百计图书与文章的挖掘与分析奠定了本书的基础构架，微软公司的技术资料馆为此提供了无法估量的帮助。Keith Barland为此付出艰辛的劳动与宝贵的时间，对此提供帮助的技术资料馆其他工作人员包括Janelle Jones, Christine Shannon, Linda Shaw, Amy Victor, Kyle Wagner, Amy Westfall和Eilidh Zuvich。

本书也从大量的他人审阅中获益匪浅。Al Corwin, Pat Forman, Tony Garland, Hank Meuret和Matt Peloquin从头到尾对本书进行了仔细推敲，感谢他们对本书所做的工作，使得你手中的本书并不像当初我写完的样子！同时我也从Wayne Beardsley, Duane Bedard, Ray Bernard, Bob Glass, Sharon Graham, Grey Hitchcock, Dave Moore, Tony Pisculli, Steve Rinn和Bob Stacy收到大量有益的意见与建议。David Sommer（11岁）也为本书的图14-3的最后定稿提出了一个好建议，谢谢David。最后，我要感谢我的夫人Tammy，感谢她精神上的支持与诙谐和幽默。我必须迅速开始我第三本书的编写工作了，以便她能停止取笑我，还管我叫业余作者！

Bellevue, Washington
1996年6月

目 录

第一部分 有效开发	1
第1章 欢迎学习快速软件开发	2
1.1 什么是快速软件开发	2
1.2 实现快速开发	3
第2章 快速开发策略	6
2.1 快速开发的总体战略	9
2.2 开发速度的四维	11
2.3 快速开发的一般分类	17
2.4 哪一维更重要	20
2.5 快速开发的权衡策略	21
第3章 典型错误	27
3.1 典型错误案例研究	27
3.2 错误对开发进度的影响	34
3.3 常见的典型错误	36
3.4 逃离Gilligan岛	44
第4章 软件开发的基本原则	45
4.1 管理原则	48
4.2 技术的基本原则	51
4.3 质量保障的基本原则	56
4.4 按照指导来做	61
第5章 风险管理	63
5.1 风险管理要素	65
5.2 风险识别	67
5.3 风险分析	72
5.4 风险的优先级	74
5.5 风险控制	76
5.6 风险、高风险和冒险	80
第二部分 快速开发	85
第6章 快速开发中的核心问题	86
6.1 一个标准是否可以适合所有情况	86
6.2 你需要什么样的开发方法	87
6.3 按时完成的可能性	92
6.4 感知与现实	94

6.5	时间到哪里去了	96
6.6	开发速度的权衡	100
6.7	典型的进度改进模式	102
6.8	向快速开发前进	103
第7章	生命期计划	105
7.1	纯瀑布模型	108
7.2	编码修正模型	110
7.3	螺旋模型	111
7.4	经过修改的瀑布模型	113
7.5	渐进原型	116
7.6	阶段交付	117
7.7	面向进度的设计	118
7.8	渐进交付	119
7.9	面向开发工具的设计	120
7.10	商品软件	121
7.11	为你的项目选择最快速的生命期	122
第8章	估算	129
8.1	软件估算的故事	131
8.2	估算步骤概述	137
8.3	规模估算	138
8.4	工作量估算	144
8.5	进度估算	145
8.6	大致的 (Ballpark) 进度估算	147
8.7	估算修正	156
第9章	进度计划	162
9.1	过分乐观的进度计划	163
9.2	战胜进度压力	173
第10章	面向客户开发	183
10.1	客户对于快速开发的重要性	185
10.2	面向客户的开发方法	187
10.3	合理控制客户的期望值	191
第11章	激励机制	194
11.1	开发人员的典型动机	196
11.2	最重要的5个激励因素	198
11.3	利用其他激励因素	203
11.4	士气杀手	206

第12章 团队合作	212
12.1 团队合作中软件的使用	214
12.2 团队合作对于快速开发的重要性	215
12.3 创造高业绩团队	217
12.4 团队为什么会失败	225
12.5 长期的团队建设	228
12.6 团队合作指导方针总结	229
第13章 团队结构	231
13.1 团队结构应考虑的因素	233
13.2 团队模式	236
13.3 管理者和技术领导	244
第14章 功能限定	248
14.1 项目早期：功能的简化	250
14.2 项目中期：功能蔓延的控制	257
14.3 项目后期：功能剪切	265
第15章 生产率工具	268
15.1 快速开发中生产率工具的作用	271
15.2 生产率工具的战略	274
15.3 生产率工具的获取	275
15.4 生产率工具的使用	279
15.5 银弹综合症	283
第16章 项目修复	289
16.1 一般的修复方案	291
16.2 修复计划	293
第三部分 最佳实践	305
最佳实践简介	306
III.1 最佳实践章节的安排方式	308
III.2 关于挑选最佳实践的总结	310
III.3 最佳实践评估之总结	315
第17章 变更委员会 (Change Board)	317
第18章 日创建和冒烟测试	318
18.1 使用日创建和冒烟测试	320
18.2 管理日创建和冒烟测试的风险	324
18.3 日创建和冒烟测试的附带效果	325
18.4 日创建和冒烟测试与其他方法的交互作用	325
18.5 日创建和冒烟测试的底线	325

18.6	使用日创建和冒烟测试成功的关键	325
第19章	变更设计	327
19.1	采用变更的设计	328
19.2	管理变更设计带来的风险	332
19.3	变更设计的附带效果	333
19.4	变更的设计与其他方法的关系	333
19.5	变更设计的底线	333
19.6	使用变更的设计方法成功的关键因素	333
第20章	渐进交付	334
20.1	渐进交付法的应用	335
20.2	控制渐进交付的风险	338
20.3	渐进交付的附带效果	338
20.4	渐进交付与其他方法的相互关系	339
20.5	渐进交付的底线	339
20.6	成功运用渐进交付法的关键	339
第21章	渐进原型	340
21.1	使用渐进原型	341
21.2	控制渐进原型的风险	342
21.3	渐进原型的附带效果	346
21.4	渐进原型与其他方法的相互影响	346
21.5	渐进原型的底线	347
21.6	成功使用渐进原型法的关键	347
第22章	目标设定	348
第23章	检查	349
第24章	联合应用开发 (JAD)	350
24.1	使用JAD	351
24.2	控制JAD的风险	358
24.3	JAD的附带效果	359
24.4	JAD与其他方法的关系	359
24.5	JAD方法的底线	360
24.6	成功使用JAD的关键	360
第25章	生命期模型的选择	361
第26章	测量	362
26.1	应用测量	363
26.2	管理测量的风险	369
26.3	测量的附带效果	371
26.4	测量与其他活动的相互关系	371

26.5	测量的底线	371
26.6	成功应用测量的关键	371
第27章	小型里程碑	372
27.1	应用小型里程碑	374
27.2	管理小型里程碑的风险	377
27.3	小型里程碑的附带效果	377
27.4	小型里程碑与其他一些活动的相互关系	377
27.5	小型里程碑的底线	378
27.6	成功应用小型里程碑的关键	378
第28章	外包	380
28.1	应用外包	381
28.2	管理外包风险	387
28.3	外包的附带效果	388
28.4	外包与其他一些活动的相互关系	388
28.5	外包的底线	388
28.6	成功应用外包的关键	388
第29章	原则性谈判	389
第30章	高效开发环境	390
30.1	应用高效开发环境	392
30.2	管理高效开发环境的风险	394
30.3	高效开发环境的附带效果	395
30.4	高效开发环境与其他活动的相互关系	395
30.5	高效开发环境的底线	395
30.6	成功应用高效开发环境的关键	396
第31章	快速开发语言 (RDLs)	397
31.1	应用快速开发语言 (RDLs)	400
31.2	管理快速开发语言 (RDLs) 的风险	401
31.3	快速开发语言的附带效果	402
31.4	快速开发语言与其他活动的相互关系	402
31.5	快速开发语言的底线	403
31.6	成功应用快速开发语言的关键	403
第32章	需求修正	404
第33章	重用	405
33.1	应用重用	406
33.2	重用中的风险管理	412
33.3	重用的附带效果	413
33.4	重用与其他实践的相互影响	413