

微软技术丛书



Microsoft

- 案例翔实, 传授名家过经过脉之实践
- 名著新解, 演绎快速软件开发之精髓

快速软件开发

(珍藏版)



(美) Steve McConnell 著
席相林 译 冯炳根 审校

清华大学出版社

微软技术丛书



快速软件开发

(珍藏版)



(美) Steve McConnell 著
席相林 译 冯炳根 审校

清华大学出版社
北 京

*Best
Practices*

内 容 简 介

进度失控,几乎是每一个软件开发项目挥之不去的噩梦。如何从容赶急,如何通过正确的开发策略和原则,避免典型错误,有效地进行风险管理,从多个方面贯彻执行快速软件开发,都可以从本书中找到答案。本书借助于实际案例和数据,阐述了快速软件开发方法的要领和精髓。

本书前两部分描述快速开发的策略和理念,其中的案例讨论有助于读者清楚地领略到策略和理念在实践中的作用。第III部分则由27个快速开发实践构成,对于技术领导、程序员和项目经理具有重要的参考和指导意义。

Rapid Development (978-1-55615-900-8)

Copyright © 1996 by Steve McConnell

Original English Language Edition Copyright © 1996 by Steve McConnell

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字:01-2007-2829

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

快速软件开发(珍藏版)/(美)麦克康奈尔(McConnell, S.)著;席相林等译;冯炳根审校. —北京:清华大学出版社,2008.8

(微软技术丛书)

书名原文:Rapid Development

ISBN 978-7-302-17813-2

I. 软… II. ①麦… ②席… ③冯… III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字(2008)第 081427 号

责任编辑:汤斌浩

装帧设计:杨玉兰

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:33.25 字 数:654 千字

版 次:2008 年 8 月第 1 版 印 次:2008 年 8 月第 1 次印刷

印 数:1~4000

定 价:69.00 元

本书如存在文字不清、漏印、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)62770177 转 3103 产品编号:026268-01



读者回执卡

欢迎您立即填写回函

您好！感谢您购买本书，请您抽出宝贵的时间填写这份回执卡，并将此页剪下寄回我公司读者服务部。我们会在以后的工作中充分考虑您的意见和建议，并将您的信息加入公司的客户档案中，以便向您提供全程的一体化服务。您享有的权益：

- ★ 免费获得我公司的新书资料；
- ★ 寻求解答阅读中遇到的问题；

- ★ 免费参加我公司组织的技术交流会及讲座；
- ★ 可参加不定期的促销活动，免费获取赠品；

读者基本资料

姓 名 _____ 性 别 ☐男 ☐女 年 龄 _____
电 话 _____ 职 业 _____ 文化程度 _____
E-mail _____ 邮 编 _____
通讯地址 _____

请在您认可处打√（6至10题可多选）

- 1、您购买的图书名称是什么：_____
- 2、您在何处购买的此书：_____
- 3、您对电脑的掌握程度：
☐不懂 ☐基本掌握 ☐熟练应用 ☐精通某一领域
- 4、您学习此书的主要目的是：
☐工作需要 ☐个人爱好 ☐获得证书
- 5、您希望通过学习达到何种程度：
☐基本掌握 ☐熟练应用 ☐专业水平
- 6、您想学习的其他电脑知识有：
☐电脑入门 ☐操作系统 ☐办公软件 ☐多媒体设计
☐编程知识 ☐图像设计 ☐网页设计 ☐互联网知识
☐书名 ☐作者 ☐出版机构 ☐印刷、装帧质量
- 7、影响您购买图书的因素：
☐内容简介 ☐网络宣传 ☐图书定价 ☐书店宣传
☐封面、插图及版式 ☐知名作家（学者）的推荐或书评 ☐其他
- 8、您比较喜欢哪些形式的学习方式：
☐看图书 ☐上网学习 ☐用教学光盘 ☐参加培训班
- 9、您可以接受的图书的价格是：
☐20 元以内 ☐30 元以内 ☐50 元以内 ☐100 元以内
- 10、您从何处获知本公司产品信息：
☐报纸、杂志 ☐广播、电视 ☐同事或朋友推荐 ☐网站
- 11、您对本书的满意度：
☐很满意 ☐较满意 ☐一般 ☐不满意
- 12、您对我们的建议：_____

1 0 0 0 8 4

北京100084—157信箱

读者服务部

收

邮政编码：□□□□□□

贴 票
邮 处

请剪下本页填写清楚，放入信封寄回，谢谢！

《快速软件开发(珍藏版)》出版前言

在 IT 界，是资深程序员又是知名软件工程专家的不胜枚举，但同时又是深受程序员喜爱的技术图书作家的就不可多得了。如果三者兼而有之，而且还曾经写过多部书，而且部部都引人入胜，令人醍醐灌顶，这样的人更是凤毛麟角，而 Steve McConnell 就是其中为数不多的让人膜拜的偶像。

在程序员眼里，他与 Bill Gates、Linus Torvalds 齐名，对软件行业做出了卓越贡献。他身兼数职，既是国际知名软件工程专家，又是 IEEE Software 杂志主编和 IEEE 计算机杂志高级评审，同时又是《软件工程知识体系》(SWEBOK)项目的专家顾问小组成员和 IEEE 计算机学会专业实践委员会特聘成员。从 1984 年起一直浸染于软件行业，熟谙快速开发方法、项目估算、软件构建实践、性能优化、系统集成和第三方合同管理等方面。

Steve McConnell 先后出版了 6 本图书。其中多本获得《软件开发》杂志 Jolt 大奖的殊荣。如果说一本技术书籍只有在获得 Jolt 奖后才能真正奠定经典的地位，那么 Steve McConnell 就是经典的缔造者。他的书，经历了时间的沉淀和实践的锤炼，往往能发人深省，动人心弦。身处软件行业的读者，几乎都能如数家珍地列举出他的著作。

对于 *Rapid Development: Strategy and best practices for optimizing software development schedules*(《快速软件开发》)一书，在经历过时间的沉淀和锤炼之后，更是为很多读者津津乐道，纷纷表示其中的实践就是真实的再现，非常值得借鉴和参考。此书曾获得 1996 年《软件开发》杂志 Jolt 大奖。历经数年，她的风采依旧。作者在书中旁征博引，在诠释快速软件开发原理与方法的同时，提供了大量实践，旨在帮助软件开发人员掌握好开发进度与软件质量之间的平衡，使他们在工作中有张有弛，既能玩命工作，又能有时间学习和娱乐。坦白地说，本书是技术领导、程序员和项目经理不可或缺的工作指南。

在征求很多读者的意见以及相关编者和译者的意见之后，在经过慎重调研和权衡之后，我们决定推陈出新，以全新的风格重现这部经典，使之能成为广大程序员职业生涯中的工作指南和良师益友。

一旦确定这个项目，我们立即联系《快速软件开发》的主译席老和冯老。席老和冯老是中国科学院计算技术研究所的专家。二位老人虽然已近古稀之年，但身体依然健硕，精神矍铄，欣然担当起重新修订的重任。他们对译稿重新进行严谨、认真的推敲和演绎。席老抽出宝贵的休息时间，



额外新增具有画龙点睛之妙用的“译者心得”部分。冯老对于编辑提出的疑问，仔细推敲，并借游泳健身的机会虚心向国外软件同行请教，以求进一步完善书稿。在此衷心感谢席老和冯老。我们的责任编辑也一丝不苟地审阅和润饰译稿，并提出很多建设性的意见。

一本经典著作，不仅要内容精确，还少不了版式上的精心编排。为此，我们不仅精心设计了版式，还考虑到读者的需要(一本经典的好书，往往会让人爱不释手，看了又看)，特别选用价格不菲的轻型纸。轻型纸质地轻柔，颜色自然，表面细腻光滑，不仅环保，还有利于视力保护，即使长时间阅读也不容易产生视觉疲劳。

作为出版工作者，我们期望能得到著译者和读者的支持和鼓励，没有你们，就没有出版业的繁荣和发展。关于本书的任何意见和建议，欢迎发送邮件到 wenkq@tup.tsinghua.edu.cn，我们一直在聆听您的心声。

清华大学出版社
2008年7月

译者序

第29届奥林匹克运动会即将在北京隆重举行的前夕, Steve McConnell 的《快速软件开发(珍藏版)》由清华大学出版社出版发行, 实为一件极为庆幸的大好事。

随着信息技术的飞速发展, 我国从事软件开发业务的公司数量在急剧增加, 软件开发队伍成倍膨胀, 软件开发项目数目和规模也越来越大。长期以来, 软件开发人员一直受估算不准、功能蔓延、团队人员流失等问题的困扰。巨大的进度压力, 突然而至的风险, 总是使这支队伍的开发工作征程步履艰难。尽早、有效地解决这些问题是每个软件人员的期待。我的工作方向转移到项目管理领域以后, 也在力图以现代项目的知识体系和工具、方法找到一些解决这些问题的途径。

2001年初, 我有幸阅读了 McConnell 撰写的这本《快速软件开发》(英文版)一书, 书的内容深深地吸引着我, 从头至尾逐章浏览本书以后, 收获颇丰, 如何解决这些困扰的思路日益清晰起来。

Steve McConnell 是微软公司的一位经验丰富的软件顾问, 他最初的工作是开发分布式商业软件, 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, 他也进行更多的理论探讨, 撰写了许多论文和书籍。“Rapid Development”就是佳作之一。除了这本书之外, 他还撰写了“Code Complete: A Practical Handbook of Software Construction”, “Software Project Survivals Guide”和“After the Gold Rush: Creating a True Profession of Software Engineering”等多部著作。

在《快速软件开发(珍藏版)》这本书中, 他先从正面提出了完整的软件开发战略, 然后详细地阐述各种最佳实践, 同时提供众多有价值的运作技巧, 这些都有助于帮助项目开发人员缩短和控制开发的进度, 保持软件项目健康推进。

本书分为三部分, 共43章。

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

第II部分, 以清醒的思路逐个专题地阐述了软件开发中的几个核心问题, 包括进行估算、开发原型、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III部分, 向读者推荐了有利于快速开发的27种行之有效的最佳实践, 诸如变更控制委员会、选择恰当的生命期模型、有原则地谈判、W理



论等,对于每条最佳实践的潜在效率和风险都进行了恰如其分的描述和分析。

在本书的前16章中,McConnell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、经理和程序人员撰写的一本好书,是使软件开发工作更高效、更为现实的工作指南。

参与本书初次翻译和校对的人员有丁丽、朱莉、李华、陈新、陈蔚力、林鄂华、高福春、韩梅、韩蓬和黎纓。由席相林进行全文的统稿工作。冯炳根教授和王利文教授对全部译文进行了精心地校对,其中王教授校对了前两部分(第1~16章),冯教授校对了第Ⅲ部分(第17~43章)。

为了本珍藏版的出版发行,本人对于前一版的全部译稿又进行了一次认真细致的整理和加工,纠正了存在的一些不足和问题,并在前16章的每一章,写了“译者注”,把本人对该章内容的理解和体会呈现出来与读者分享。冯炳根教授对本译文再次进行认真的校对和完善。

在整个过程中,清华大学出版社文开琪老师给予了详细的指导,汤斌浩老师对译文逐字逐句进行校对和推敲,不放过丝毫的疑点,他们严谨治学的态度令我非常敬佩。此外,北京中科项目管理研究所给予了项目管理方面的帮助。杜端甫教授、许书珍、许成绩、赵欣如、沈天阳、王星、张嘉飞等也提供了其他方面的帮助。本书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。

书中译文中如有不当之处,恳请读者批评指正。

席相林

2008年7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殚精竭虑，几乎没有时间去钻研有效的实用技术；另一方面，如果不学习掌握软件快速开发方法，他们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 天才按计划完成开发项目，指望他们投入很多精力研究软件可用性方面的问题又从何谈起？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，但有些组织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达到 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得可以控制，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，而无需阅读整本书。不管你的项目处于何种阶段，你都会在这本书中找到能够帮助你改善当时境况的实用内容。

本书适用的对象

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，而不必分心去描述“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的，而不仅仅像字面想象的那样。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。其实，作为一名项目经理，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，做一些自己的小小变革。

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。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、让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. 提高开发速度

你可以在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，你总会在本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，或工作在嘈杂不利于生产的环境中有关。

2. 快速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，这些通常被看作是典型的“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的，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、使用这些方法使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。我无需讲述任何有关 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，以及你如果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，我并不要求你将所有的鸡蛋放到同样尺寸的篮子里。我认为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. 为什么本书会如此之厚

信息系统、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。本书从各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了。作为补救措施，我将本书的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的部分来阅读了。

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(Jones 1994)，将近 60% 的开发人员说他们感到工作压力在增加(Glass 1994c)，美国的开发人员每周工作时间在 48 ~ 50 小时(Krantz 1995)，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，导致开发人员开发工作负担过重，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，因而我们并不苟同编写软件只是为了获得报酬的说法。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间设置一道堤坝了，本书中我试图树立一个堤坝标杆，以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 Kim Eggleston 为本书所做的美丽设计,感谢 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 对本项目的顺利出版所做出的贡献。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,但我也衷心地向他们表示感谢。(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 为此付出了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、Greg 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 的最后一个图片提出了一个好建议,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工作了,以便她能停止取笑我,还管我叫业余作者!

Steve McConnell
Bellevue, Washington
1996年6月

目 录

第 I 部分 有效开发

第 1 章 欢迎学习快速开发	3	4.1.2 计划编制	49
1.1 什么是快速开发	3	4.1.3 跟踪	49
1.2 实现快速开发	4	4.1.4 度量	50
第 2 章 快速开发策略	7	4.2 技术的基本原则	52
2.1 快速开发的总体策略	10	4.2.1 需求管理	53
2.2 开发速度的四维	11	4.2.2 设计	54
2.2.1 人员	12	4.2.3 构建	55
2.2.2 过程	14	4.2.4 软件配置管理	56
2.2.3 产品	16	4.3 质量保证的基本原则	58
2.2.4 技术	17	4.3.1 易错模块	61
2.2.5 协同	17	4.3.2 测试	61
2.3 快速开发的一般分类	17	4.3.3 技术回顾	61
2.3.1 有效开发	18	4.4 按照指导来做	64
2.3.2 侧重于最佳进度的有效开发	19	深入阅读	65
2.3.3 全面快速开发	19	第 5 章 风险管理	67
2.4 哪一维更重要	20	5.1 风险管理要素	69
2.5 快速开发的权衡策略	21	5.1.1 风险评估	70
深入阅读	25	5.1.2 风险控制	70
第 3 章 典型错误	27	5.2 风险识别	70
3.1 典型错误案例研究	27	5.2.1 最常见的进度计划风险	71
3.2 错误对开发进度的影响	33	5.2.2 进度计划风险的完整列表	71
3.3 典型错误一览	34	5.3 风险分析	75
3.3.1 人员	35	5.3.1 风险暴露量	75
3.3.2 过程	39	5.3.2 估计损失的大小	76
3.3.3 产品	41	5.3.3 评估损失发生的概率	76
3.3.4 技术	42	5.3.4 整个项目的延期和缓冲	76
3.4 逃离格里甘岛	44	5.4 风险优先级	77
深入阅读	44	5.5 风险控制	78
第 4 章 软件开发的基本原则	45	5.5.1 风险管理计划	78
4.1 管理原则	48	5.5.2 风险化解	79
4.1.1 项目估算和进程安排	48	5.5.3 风险监控	81
		5.6 风险、高风险和冒险	83
		深入阅读	85

第 II 部分 有效开发

第 6 章 快速开发中的核心问题	89	深入阅读	128
6.1 一个标准是否适合所有情况	89	第 8 章 估算	129
6.2 你需要什么样的开发方法	90	8.1 软件估算的故事	130
6.2.1 进度计划有严格限制的产品	91	8.1.1 软件和建筑	131
6.2.2 表面上的快速开发	92	8.1.2 软件开发是一个改进的过程	132
6.2.3 你真正需要的是否是全力以 赴地开发	94	8.1.3 可能细化的数量	132
6.3 按时完成的可能性	94	8.1.4 估算与控制	134
6.4 感知与现实	97	8.1.5 合作	135
6.4.1 不切实际的用户期望	98	8.1.6 估算实例概要	136
6.4.2 克服慢速开发的感觉	98	8.2 估算步骤概述	137
6.5 时间花到哪里去了	99	8.3 规模估算	137
6.5.1 典型的观点	99	8.3.1 功能点估算	138
6.5.2 可以改善的问题	100	8.3.2 估算技巧	140
6.6 开发速度的平衡	102	8.3.3 估算的表达式	142
6.6.1 进度、费用和产品的平衡	102	8.4 工作量估算	144
6.6.2 质量的权衡	103	8.5 进度估算	144
6.6.3 个人效率的权衡	103	8.5.1 基于承诺的进度安排	145
6.7 典型的进度改进模式	103	8.5.2 Jones 的一阶估算实践	145
6.8 向快速开发前进	105	8.6 大致的进度估算	146
深入阅读	106	8.6.1 背景	146
第 7 章 生命期计划	107	8.6.2 可能的最短进度	148
7.1 纯瀑布模型	109	8.6.3 有效进度	152
7.2 编码修正模型	112	8.6.4 普通进度	153
7.3 螺旋模型	113	8.6.5 对大致的进度首先应怎么办	155
7.4 经过修改的瀑布模型	115	8.7 估算修正	155
7.4.1 生鱼片模型	115	深入阅读	159
7.4.2 具有子项目的瀑布模型	116	第 9 章 进度计划	161
7.4.3 能够降低风险的瀑布模型	117	9.1 过分乐观的进度计划	162
7.5 渐进原型	117	9.1.1 一个关于过分乐观的进度 计划的实例	163
7.6 阶段交付	118	9.1.2 产生过分乐观的进度计划的 根源	164
7.7 面向进度的设计	119	9.1.3 过分乐观的进度计划产生的 不良后果	165
7.8 渐进交付	120	9.1.4 超负荷的进度压力	168
7.9 面向开发工具的设计	121	9.1.5 底线	171
7.10 商品软件	122		
7.11 为你的项目选择最快速的生命期	123		

9.2 战胜进度压力	172
9.2.1 原则谈判法	173
9.2.2 将人和问题分开	174
9.2.3 关注共同利益，不要过分 坚持立场	175
9.2.4 提出对双方均有利的备选方案 ...	176
9.2.5 坚持客观标准	177
深入阅读	180

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
10.1.1 提高效率	186
10.1.2 减少返工	186
10.1.3 降低风险	186
10.1.4 消除矛盾	187
10.2 面向客户的开发方法	187
10.2.1 规划	187
10.2.2 需求分析	188
10.2.3 设计	190
10.2.4 实现	190
10.3 合理控制客户的期望值	191
深入阅读	193

第 11 章 激励机制 195

11.1 开发人员的典型激励	196
11.2 最重要的 5 个激励因素	199
11.2.1 成就感	199
11.2.2 发展机遇	201
11.2.3 工作乐趣	202
11.2.4 个人生活	203
11.2.5 成为技术主管的机会	204
11.3 利用其他激励因素	204
11.3.1 奖赏和鼓励	204
11.3.2 试验性项目	205
11.3.3 对业绩的评价	206
11.4 士气杀手	207
11.4.1 保健因素	207
11.4.2 其他士气杀手	208
深入阅读	212

第 12 章 团队合作 215

12.1 团队合作中软件的使用	217
-----------------------	-----

12.2 团队合作对于快速开发的重要性	217
12.2.1 团队生产率的变化	218
12.2.2 凝聚力和业绩	218
12.3 创建高绩效团队	219
12.3.1 共同的、可提升的愿景或 目标	220
12.3.2 团队成员的认同感	221
12.3.3 结果驱动的结构	222
12.3.4 胜任的团队成員	222
12.3.5 对团队的承诺	224
12.3.6 相互信任	224
12.3.7 团队成员间相互依赖	225
12.3.8 有效的沟通	225
12.3.9 自主意识	226
12.3.10 授权意识	226
12.3.11 团队规模较小	226
12.3.12 高层次的乐趣	227
12.3.13 如何管理高绩效团队	227
12.4 团队为什么会失败	228
12.5 长期的团队建设	230
12.6 团队合作指导方针总结	232
深入阅读	233

第 13 章 团队结构 235

13.1 团队结构应考虑的因素	237
13.1.1 团队的种类	237
13.1.2 其他团队设计特征	239
13.1.3 何种类型的团队最适用于 快速开发	239
13.2 团队模式	240
13.2.1 业务团队	241
13.2.2 主程序员团队	241
13.2.3 科研项目团队	242
13.2.4 特征团队	243
13.2.5 搜索救援团队	243
13.2.6 SWAT 团队	244
13.2.7 专业运动员团队	244
13.2.8 戏剧团队	245
13.2.9 大型团队	246
13.3 管理者和技术领导	247
深入阅读	250