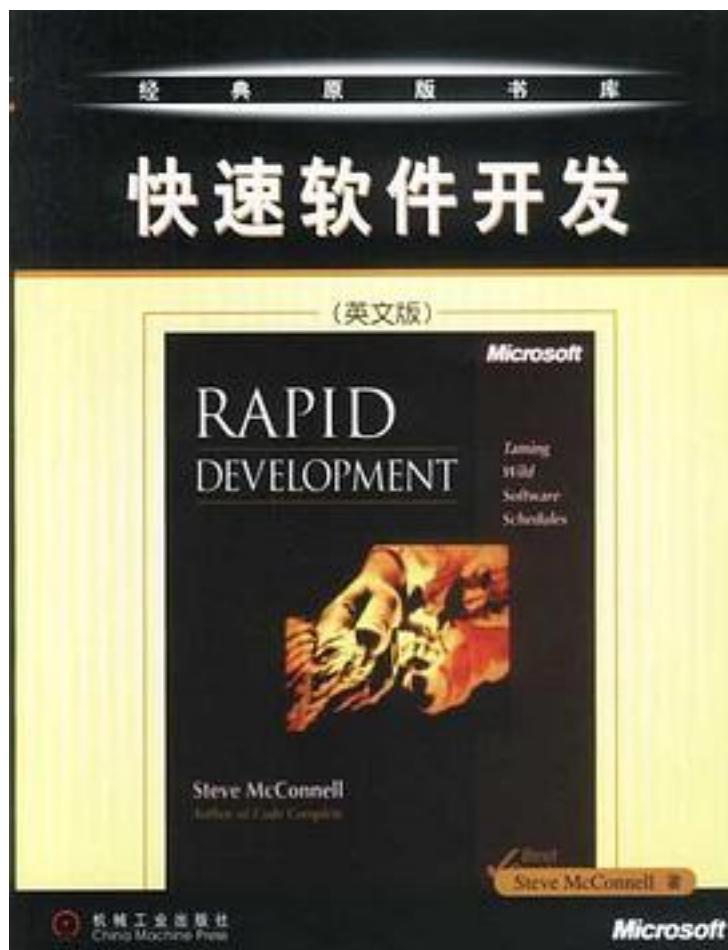


快速软件开发



[快速软件开发 下载链接1](#)

著者:[美] 斯蒂夫·迈克康奈尔

出版者:电子工业出版社

出版时间:2002-1

装帧:平装

isbn:9787505372856

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最常见和最难解决的问题。这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。

作者介绍：

斯蒂夫·迈克康奈尔（Steve McConnell）是IEEE Software的总编，Construx Software的总工程师兼总裁，多家世界知名软件公司的顾问，在美国软件业享有很高的声誉。他编著的图书包括获得1993年度美国Jolt图书大奖的《完美的编码法则》，获得1999年Jolt图书大奖的《淘金热的背后——成为专业的软件工程人员》《微软项目：求生法则》。

目录: 第1部分 有效开发

第1章 欢迎学习快速开发

1.1 什么是快速开发

1.2 实现快速开发

第2章 快速开发策略

2.1 快速开发的总体策略

2.2 开发速度的四维

2.2.1 人员

2.2.2 过程

2.2.3 产品

2.2.4 技术

2.2.5 协同

2.3 快速开发的一般分类

2.3.1 有效开发

2.3.2 侧重于最佳进度的有效开发

2.3.3 全面快速开发

2.4 哪一维更重要

2.5 快速开发的权衡策略

深入阅读

第3章 典型错误

3.1 典型错误案例研究

3.2 错误对开发进度的影响

3.3 典型错误一览

3.3.1 人员

3.3.2 过程

3.3.3 产品

3.3.4 技术

3.4 逃离格里甘岛

深入阅读

第4章 软件开发的基本原则

4.1 管理原则

4.1.1 项目估算和进程安排

4.1.2 计划编制

4.1.3 跟踪

4.1.4 度量

4.2 技术的基本原则

4.2.1 需求管理

4.2.2 设计

4.2.3 构建

4.2.4 软件配置管理

- 4.3 质量保证的基本原则
- 4.3.1 易错模块
- 4.3.2 测试
- 4.3.3 技术回顾
- 4.4 按照指导来做
- 深入阅读
- 第5章 风险管理
- 5.1 风险管理要素
- 5.1.1 风险评估
- 5.1.2 风险控制
- 5.2 风险识别
- 5.2.1 最常见的进度计划风险
- 5.2.2 进度计划风险的完整列表
- 5.3 风险分析
- 5.3.1 风险暴露量
- 5.3.2 估计损失的大小
- 5.3.3 评估损失发生的概率
- 5.3.4 整个项目的延期和缓冲
- 5.4 风险优先级
- 5.5 风险控制
- 5.5.1 风险管理计划
- 5.5.2 风险化解
- 5.5.3 风险监控
- 5.6 风险、高风险和冒险
- 深入阅读
- 第 II 部分 有效开发
- 第6章 快速开发中的核心问题
- 第7章 生命期计划
- 第8章 估算
- 第9章 进度计划
- 第10章 面向客户开发
- 第11章 激励机制
- 第12章 团队合作
- 第13章 团队结构
- 第14章 功能限定
- 第15章 生产率工具
- 第16章 项目修复
- 第 III 部分 最佳实践简介
- 第17章 变更委员会
- 第18章 日创建和冒烟测试
- 第19章 变更设计
- 第20章 渐进交付
- 第21章 渐进原型
- 第22章 目标设定
- 第23章 检查
- 第24章 联合应用程序开发
- 第25章 生命期模型的选择
- 第26章 度量
- 第27章 小型里程碑
- 第28章 外包
- 第29章 原则谈判法
- 第30章 高效开发环境
- 第31章 快速开发语言
- 第32章 需求提炼

第33章 重用
第34章 签约
第35章 螺旋型生命期模型
第36章 阶段性交付
第37章 W理论管理
第38章 舍弃型原型法
第39章 限时开发
第40章 工具组
第41章 前十大风险清单
第42章 构建用户接口原型
第43章 自愿加班
参考文献
· · · · · (收起)

[快速软件开发](#) [下载链接1](#)

标签

项目管理

软件工程

软件开发

计算机

快速软件开发

管理

编程

PM

评论

看过了，感觉写得还可以。

虽然和清华出版社出的珍藏版是同一个翻译，但翻译质量差得很。

受益匪浅

非常实用 但是北图竟然没有！

快速开发后面就是敏捷开发了（XP极限编程）

一年前读过一部分

17年前+厚达500页+无独到观点+学究气=没必要读。

翻了下,太过学院派的风格,讲究大而全,就象图书馆里的馆藏虽多.却不适合由此得学习之道.

关于外包管理和加班讲得很中肯

闪光点很多

牛人，幽默 项目开发解决方案全书

有一定借鉴价值，尤其对于小团队。

软件项目管理必读，而且值得反复研读和思考。

8年之后再重读，一本经得起时间考验的经典

奇怪，是我么有快速开发的迫切需要吗？稍微的冗长，稍微的不确定，软件工程，总是在思考后……继续思考。实践已离我远去

90年代就能有如此思想，快20年了，还未过时。。。

很不错

还不错的项目过程控制，值得经常读~~

软件工程管理的错误经验总结，在上面我看到了很多我曾经犯过的或是经历过的典型错误。。。总体上很值得一看。不过没有代码上的展示对我而言，可读性稍微差点，需要更多的时间慢慢阅读。

快速浏览各种模式的开发过程

[快速软件开发 下载链接1](#)

书评

本人经验并不丰富,只是不小心做了一段时间的管理者:软件开发真的是到处是陷阱.现在看了此书,里面很多的失败的实例就和自己在管理过程中遇到的一样.
书上提到的实践,绝对值得一试!因为软件发现在还没有银弹!

如果说, 像<人月神话>,<设计模式>,<代码大全>,这样的著作来得过于宏大、经典, 请千万不要错过这本很经典, 但非常有趣, 充满着阅读乐趣的书----<快速软件开发>
<http://www.douban.com/subject/1007738/> 与其把此书当作技术管...

还能说什么, 虽然书里没有敏捷, 也没有单元测试, 更没有AOP/SOA等时髦名词, 但是这绝对不影响这本书的价值! 其实我一直觉得, 国内那么多小软件公司, 搞什么RUP/XP/SOA/AOP, 其实好好看看这本书, 没有什么项目做不成的~

工期和质量是一个永远存在于软件开发中的矛与盾么? ? 看看书中提到的所有失败, 以及不成功的例子多是由于不断缩短工期而造成的, 但是如果不能及时抢占市场, 即使开发出完美的软件, 却也犹如空有利剑而无用武之地。所以工期和质量的矛盾, 只有能找到他们之间平衡点的项目才能...

chapter 2 best possible schedule = Classic-Mistake Avoidance + Development Fundamentals + Risk Management + Schedule-Oriented Practices

这书虽然是1996年出的, 但是现在看来还是相当的经典, 姜还是老的辣, 现在的新鲜玩意满天飞, 但是真正经典的还是这些传世之作, 其实很多公司天天也不知道在忙啥, 人员还不知道怎么管理, 就天天XP,RUP,AOP,先把这本书看透了再说, 足够做项目了。

这书虽然是1996年出的, 但是现在看来还是相当的经典, 姜还是老的辣, 现在的新鲜玩意满天飞, 但是真正经典的还是这些传世之作

这是我所见软件开发领域最好的书, 任何从事软件开发, 或者与软件开发相关行业的, 又或者想了解软件开发行业的人, 不管你是程序员、项目经理、还是老板, 都应该读这

本书。任何软件学院、计算机相关专业都应该把这本书作为教材开一门必修课！用这本书给国内软件业所有从业人员集...

看了6章了，深刻体会了。上一个负责的项目很大，风险没有控制好，结果造成了较大的延迟。新项目开始了，也正在努力将书中的知识运用进去，真的是一本难得的好书，从《代码大全》到这本书，真的对让我获益良多

无论对于软件工程的入门者还是正在管理软件开发项目的PM，都非常有帮助。
这本书是对敏捷技术出现前的软件工程实践/方法的绝妙总结。
书中有许多短小的案例，把现实中软件开发项目可能遇到的难题生动地展现在读者面前，为本书增加了许多趣味。

[快速软件开发 下载链接1](#)