

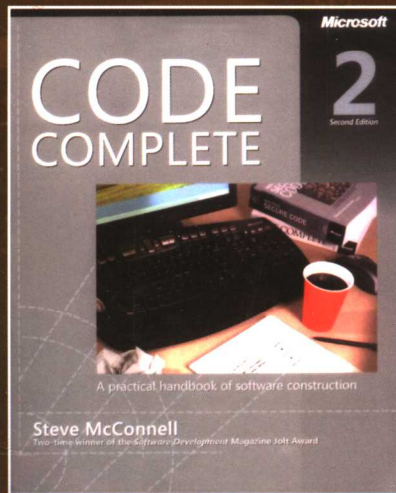
CODE COMPLETE

Second Edition

代码 2 大全

第 2 版 英文版

[美] Steve
McConnell 著



两届
Software Development Magazine
Jolt Award
震撼大奖得主

软件构建之实践指南
A practical handbook of software construction



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

代 码 大 全

— (第2版) 英文版 —

CODE COMPLETE Second Edition

[美] Steve McConnell 著

江苏工业学院图书馆
藏书章

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

《代码大全(第2版)》是著名IT畅销书作者、*IEEE Software*杂志前主编、具有20年编程与项目管理经验的Steve McConnell十余年前的经典著作的全新演绎;第2版做了全面的更新,增加了很多与时俱进的内容,包括对新语言、新的开发过程与方法论的讨论,等等。这是一本百科全书式的软件构建手册,涵盖了软件构建活动的方方面面,尤其强调提高软件质量的种种实践方法。

作者特别注重源代码的可读性,详细讨论了类和函数命名、变量命名、数据类型和控制结构、代码布局等编程的最基本要素,也讨论了防御式编程、表驱动法、协同构建、开发者测试、性能优化等有效开发实践,这些都服务于软件的首要技术使命:管理复杂度。为了培养程序员编写高质量代码的习惯,书中展示了大量高质量代码示例(以及用作对比的低质量代码),提高软件质量是降低开发成本的重要途径。除此之外,本书归纳总结了来自专家的经验、业界研究以及学术成果,列举了大量软件开发领域的真实案例与统计数据,提高本书的说服力。

本书中所论述的技术不仅填补了初级与高级编程实践之间的空白,而且也为程序员们提供了一个有关软件开发技术的信息来源。本书对经验丰富的程序员、技术带头人、自学的程序员及没有太多编程经验的学生都是大有裨益的。可以说,只要您具有一定的编程基础,想成为一名优秀的程序员,阅读本书都不会让您失望。

Copyright © 2006 by Microsoft Corporation. All rights reserved.

Original English language edition ©2004 by Microsoft by Steve C. McConnell.

All rights reserved. English Reprint edition published by arrangement with the original publisher, Microsoft Corporation, Redmond, Washington, U.S.A.

本书英文影印版专有出版权由Microsoft Corporation授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号:图字:01-2006-6059

图书在版编目(CIP)数据

代码大全:第2版:英文版/(美)迈克康奈尔(McConnell,S.)著.一北京:电子工业出版社,2006.12

书名原文:Code Complete, Second Edition

ISBN 7-121-03362-3

I.代... II.迈... III.软件开发—手册—英文 IV.TP311.52

中国版本图书馆CIP数据核字(2006)第128636号

责任编辑:周筠 陈元玉

印 刷:北京市天竺颖华印刷厂

装 订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×980 1/16 印张:61 字数:1000千字

印 次:2006年12月第1次印刷

定 价:148.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010)68279077;邮购电话:(010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

《代码大全 (第2版)》所得到的赞誉

《代码大全》是我早在好几年前便已经阅读过的好书。这几年来我不知买过多少书籍，也清理过许多书房再也放不下的书籍，但是《代码大全》这本书始终占据着我书架上重要的位置而不曾移开过，因为好书是经得起时光考验的。

▶ CodeGear 公司大中华首席技术官 (CTO) 李维

在众多的编程类书籍中，如果只让我挑一本书来阅读，那我一定选择《代码大全》，因为它是最不可或缺。

▶ 微软亚洲研究院 研究员 潘爱民

《代码大全 (第2版)》是一本真正体贴开发者的经典巨著！

▶ CSDN 副总经理 曾登高

能把软件做好的人很多，但是把方法总结出来却很难。《代码大全》做到了，它堪称一本构建软件的百科全书。我在阅读过程中因无数次的共鸣而欣喜。

▶ 《游戏之旅——我的编程感悟》作者 云风

《代码大全 (第2版)》曾伴我走过研习软件开发的岁月，读来收获颇丰，唇齿留香，是技术书中少见的“好吃又有营养”的佳作。

▶ 《JUnit IN ACTION 中文版》译者 鲍志云

《代码大全 (第2版)》是最佳的软件构建论著，在中国已经创造了软件开发技术图书出版的奇迹。而我们更希望这个奇迹能够实实在在地推动中国软件开发者能力的提升。

▶ 《程序员》杂志社

国内读者对《代码大全（第2版）》的赞誉

以下读者评价摘自中国互动出版网 (www.china-pub.com)、第二书店 (www.dearbook.com.cn)、华储网 (www.huachu.com.cn)。

10年前，每当我成功解决一个数学难题，我就有一种莫名的成就感。解题的思路越巧妙，成就感越强，直到我看到著名的数学教育家波利亚 (Polya G.) 的《怎样解题》。这是一本小册子，但却是对我影响最深的数学书之一。……我看《代码大全（第2版）》……不禁想起了波利亚的这本名著。现在对《代码大全（第2版）》的褒奖太多了，我想说明的是，如果你曾经写过很多代码，常常为自己的代码组织和设计困惑，那就看看这本书吧。他对我的冲击就如《怎样解题》一样，让我在 Steve McConnell 循序渐进的讲述方式中对那些未成熟的编码思路变得豁然开朗。……不仅让你正确地理解开发行为，还给出了很多设计和开发涉及到的细节。如果说类似《人月神话》这样的软件工程的图书离开发者还有点遥远的话，《代码大全（第2版）》就是真正体贴开发者的经典巨著。

► 曾登高

McConnell 是 15 年来国际知名的软件工程专家，他的书本本都很精彩。但最关键的，这是一本宣扬科学精神和工程态度的书，作者是在科学、工程的范畴内探讨、总结程序员的基础技能、开发技艺。……软件工程是一个极其庞大、无所不包的知识体系，编程 (programming) 或构建 (construction) 之技艺正是国际软件工程界几十年苦心研究的核心内容，绝不是我们所想的学点高级程序设计语言、学点 API 那么简单。如果此书早引进 10 年，可能我国今日之软件面貌将是另一番景象。形成这种错误的观念不能怪谁，怪的是这些年来落后的软件工程教育。那么在编程技艺的总结、归纳和发展上，我们与世界领导者差距有多大，此书就是回答。

► 张恂

2004 年看过第一遍，觉得相见恨晚，打印了几十份，分发给新员工看；2005 年，再次阅读，并且以《代码大全》为内容，作了半年的软件创建培训；2006 年，购买 35 本新版《代码大全（第2版）》，在部门推广，我觉得能把这本好书推广给每个开发人员，就是对他的巨大帮助。要求是每人必须完成读书报告。

► coolwhy

代码大全 (第2版) 英文版

翻译得不错，如果所有的书都翻译成这样，翻译版的图书名声也就不会那么差了。

► miqx

这本书我看了一下……内容确实很不错。我觉得对于中国来说，各个阶层的程序员都有必要看看。……有书评说这本书只适合初、中级别的程序员看，我看不然……我觉得有必要好好去思考一些问题，好好去吸收别人的思想。这本书我觉得不是一本能很快提高你编程能力的书，但是确实是一本教你如何思考的书。

► 孤独的鹰

十几年前，这本书让我第一次认识到编程风格的重要，曾经买了N本，都推荐给朋友们了，到现在反而自己手里没有这本书。刚发现这本书出了新版……迫不及待地推荐给大家。

► iasc

我只想说是我见过的最好的关于编程经验的书，前几天买回来一看就被它震撼了。这本书很生动，也翻译得很好，我完全没有在读翻译版的感觉，对于一些英文术语在中文的旁边都附上了该英文。所以如果你真的是想获得这本书的精华经验，我还是建议你们花点钱买纸版的好。否则我想你不会有什么收获的。

► morky Zhang

我是搞程序开发的，我们用C++比较多，听说这本书不错，所以我就买了一本，买回来放在办公室，我还没有来得及看，结果被我一个同事看到了，他拿过去翻了下，结果他不还了，说什么让他先看完再还，因为他认为写的实在是太好了。后来我也拿过来仔细地翻了一下，果然不错。后来我向公司的同事极力推荐这本书，好多人都买了。

► linux_xxx

这本书给我的感觉就是很爽，写的都是一些很具体的问题，而不像普通计算机书一样，空话一堆。刚拿到书时，随手一翻，都不知道要从哪里开始看比较好，因为每一页都很精彩，语言也很平实。更爽的是这本书很厚，是一本可以让你慢慢享受的书，感觉很超值！

► scutteng

……开始读的时候就被书里的那些平实的话深深吸引，开始我是有选择地读了其中的有关程序员心理部分内容……内容更是于平淡中透着深刻，于是又翻开第一页，决定从头好好细细咀嚼了……当然，这本书读一遍是肯定不够的，好多人在读了不下四遍以后还有新的收获。希望在接下来的日子里有CC2相伴，

能发现其更多妩媚之处！

► derry

七年前看过了本书的第1版原版。的确是很好的书。深入浅出，几乎说清楚了软件开发的方方面面。作者的确是那种又能干又能说的牛人。现在看到这里的第2版……译者处理得比较好的部分就是关键的术语都是中英并列，比较方便理解。

► songma

我是在2005年年底开始在一家网络通信公司实习时知道这本《代码大全（第2版）》的。它是公司推荐新入职研发部员工阅读书目之一……

► woody107

如此重量级的图书，值得收藏。特别那些刚进公司的新员工……趁早买一本。

► thtcstar

经典著作，包罗万象！程序员必读！

► 兵工厂

感觉这本书很好，深入浅出。……买回来后觉得物有所值。

► huangmo98

这本书讲什么

《代码大全》这本书的原名叫 *Code Complete*，那么 *code complete* 在这里是何含义呢？首先，它不代表现代集成开发环境（IDE）中的代码自动补全功能，本书也不打算向您讲解 Eclipse 或 Visual Studio 2005 中的代码自动补全功能是如何实现的^①。其次，*code complete* 也不是真正的软件源代码“大全”的意思^②，这本书既没有列出连接各种数据库的代码、也没有列出网页中常用的各种 JavaScript 代码。书中的代码示例恐怕也不能直接 copy&paste 代码到您自己的项目中。

那么 *code complete* 到底是什么意思？中译本为什么又要取名为“代码大全”呢？虽然从网上讨论的情况看，各位网友对书名含义的理解有出入，但是译者有充分的理由相信，*code complete* 是“编码完成”的意思，是一个软件项目开发过程中的重要里程碑（milestone）。软件项目进行到这里，表明已经完成了所有的编码工作，即将开始系统测试。

这本书讲的正是为了到达“编码完成”这一重要里程碑所必需的软件构建技术，确切地说，就是如何编写高质量的代码。作者认为，应该首先为人编写代码，其次才是为机器（第 34.3 节）；代码主要是供人阅读的。遍布全书的提高代码质量的实实在在的技术和诀窍，是本书最有价值的部分。事实上，我们认为第 6、7、10 至 19 章这 300 多页的内容是本书的精华内容，在其他书里恐怕很难找到如此详尽的对变量、语句、子程序等编程基本要素的讨论。

十多年前，本书第 1 版以《代码大全》为名翻译出版，在过去的 10 余年中，这本书影响了整整一代程序员，“代码大全”四个字已成为一个响当当的名字。鉴于此，本书第 2 版决定保留这个无伤大雅的“错误”，沿用“代码大全”作为书名，也借此向原书第 1 版各位译者、修订者们的辛勤劳动表示我们的敬意。无论如何，对 *code complete* 的理解不会影响对整本书的理解。

本书除了讲如何构建高质量的软件，还讲如何成为一名优秀的程序员（第 33 章“个人性格”、第 4.3 节“你在技术浪潮中的位置”、第 34.4 节“深入一门语言去编程”）。

这本书适合谁看，该怎么看

任何想写出好程序的人，或者想带领一群程序员写出好软件的人，都不应该错过这本好书。作者在前言中指明了本书的读者群（包括经验丰富的程序员、技术带头人、自学的程序员、学生等），请您参阅。

这是一本 800 多页的大部头，从头到尾阅读要花不少时间，谁都希望能尽快找到对自己有用的内容。译者大致针对不同的读者群提一点阅读建议，仅供参考。

- 初级程序员，请先看第 18 章“表驱动法”：将复杂的逻辑判断转换为查表，从而简化代码的编写与维护。另外，本章中的一个示例说明了，面向对象设计并不只是因为它是“面向对象”，就一定会好于其他的设计。
- 高级程序员，请先看第 4 章“关键的‘构建’决策”，本章关注的焦点是程序员和技术带头人个人必须（直接或间接）负责的项目准备工作。
- 项目经理，请先看第 33 章“个人性格”，程序设计是一项纯粹的脑力劳动，本章对挑选和培养优秀程序员提出了建议。事实证明，相对于聪明程度（智商），个人性格（情商）对于造就出程序员高手更具有决定性的意义。
- 低年级学生，请先看第 11 章“变量名的力量”。这本书用了整整一章（30 多页）的篇幅来讲解“为变量命名”这一编程中最常见的活动，这里提供的建议在别的书里是很难见到的。
- 高年级学生，请先看第 8 章“防御式编程”，本章讲述如何面对严酷的充斥非法数据的真实世界，在遇到“绝不会发生”的事件和其他程序员犯下的错误时如何保护自己。对于那些正在从学术环境转向专业开发环境的学生来说，这是必备的一课。
- 制定编码标准的人，请先看第 32 章“自说明代码”，本章中有一段关于注释的精彩对话，它可能会改变您在制定编码标准时对注释的要求。
- 自学编程的人，请先看第 7 章“高质量的子程序”，本章详细讨论了子程序的命名和参数选择等问题，其中对子程序最佳长度的讨论颇有借鉴意义。
- 喜欢参与网上争论的人，请先看第 13.3 节“全局数据”和第 17.3 节“goto 语句”，听听学术界在这些问题上的争论也挺有意思。

当然，这整本书都非常值得一读，准确地说，值得反复阅读。书中不仅有实实在在的数据和论述，也有一些有趣的比喻，作者偶尔还开开心，读起来一点也不枯燥。

另外需要说明的一点是，书中出现的诸如“(Yourdon 1986b)”表示的是参考文献，可以从第 863 页起的参考文献列表中查到文献的原名和出处，例如，(Yourdon 1986b) 代表的是 Edward Yourdon 写

的 *Nations at Risk* 一书。如果只出现“(2000)”字样，那么请您从上下文中推断出作者姓名。

配套网站

这本书英文版的配套网站是 <http://www.cc2e.com>，书中左侧出现的类似 [cc2e.com/1234](http://www.cc2e.com/1234) 的标志的含义请参阅前言中的说明。本书中文版的配套网站是 <http://www.cc2e.com.cn>，凡是书中出现的 [cc2e.com/1234](http://www.cc2e.com/1234) 均可对应访问 [cc2e.com.cn/1234](http://www.cc2e.com.cn/1234)。

译者
2006年2月初

对《代码大全》的更多赞誉

“《代码大全》是有关编程风格和软件构建的绝好指导书。”

► **Martin Fowler**, 《重构》

“Steve McConnell 的《代码大全》……为程序员提供了通向智慧的捷径……他的书读起来饶有趣味，要知道他可是有切实的亲身经验的。”

► **Jon Bentley**, 《编程珠玑（第二版）》

“这无疑是我所看过的软件构建方面最好的书籍。每个开发人员都应该有一本，并且每年都从头到尾读一遍。九年来我每年都读这本书，仍能从中有新的收获。”

► **John Robbins**, 《Microsoft .NET 和 Windows 应用程序调试》

“当今的软件必须是健壮、有弹性的，而安全的代码始于规范的构建。第 1 版出版后的十年里，没有出现比《代码大全》更权威的书。”

► **Michael Howard**, 《编写安全的代码》

“《代码大全》广泛剖析编程工艺的各种实战话题。McConnell 的著作涵盖软件架构、编码标准、测试、集成以及软件工艺的本质等内容。”

► **Grady Booch**, *Object Solutions*

“对软件开发者而言，终极的百科全书就是 Steve McConnell 的《代码大全》。这本长达 850 页厚的书确如其副标题所说，是一本实用手册。它旨在缩短‘业界大师与教授’（例如 Yourdon 和 Pressman）的知识与一般商业实践之间的距离，帮助读者用较短的时间、碰到较少的麻烦去编写更好的程序……每个开发者都应该拥有这本书，其风格和内容是切实可用的。”

► **Chris Loosley**, *High-Performance Client/Server*

“Steve McConnell 的创新书籍《代码大全》是详述软件开发方面最易懂的一本书……”

► **Erik Bethke**, *Game Development and Production*

“《代码大全》是关于设计与生产优秀软件的实用信息与建议的宝藏。”

► **John Dempster**, *The Laboratory Computer: A Practical Guide for Physiologists and Neuroscientists*

代码大全（第2版）英文版

“如果你有意改进编程技术，就该有一本 Steve McConnell 的《代码大全》。”

► **Jean J. Labrosse**, *Embedded Systems Building Blocks: Complete and Ready-To-Use Modules in C*

“Steve McConnell 写出了一本独立于特定计算机环境的软件开发方面最好的书籍。”

► **Kenneth Rosen**, *Unix: The Complete Reference*

“每个时代你都会遇到一本书，提供你获得经验的捷径，节省数年走弯路的时间……千言万语都无法说明这本书有多好。标题《代码大全》尚不足以表达出该作品的全部智慧与内涵。”

► **Jeff Duntemann**, *PC Techniques*

“Microsoft 出版社出版了我认为是软件构建方面很好的书，每个软件开发人员的书架上都该有这本书。”

► **Warren Keuffel**, *Software Development*

“每个程序员都该读读这本杰出的书籍。”

► **T.L. (Frank) Pappas**, *Computer*

“假如你期望成为专业程序员，这将是投资 35 美元能得到的最好回报。不要只是看看这个书评，赶快冲出去买一本回来！McConnell 声称此书意在拉近业界大师的知识与一般商业实践之间的距离……令人称奇的是他做到了。”

► **Richard Mateosian**, *IEEE Micro*

“应当让在软件开发领域中的每个人都来读读《代码大全》。”

► **Tommy Usher**, *C User's Journal*

“我不遗余力地为 Steve McConnell 的《代码大全》拍手叫好……这本书取代了 API 参考手册，成为伴我干活的最亲密的书。”

► **Jim Kyle**, *Windows Tech Journal*

“这本编纂精良的巨著有望成为软件实现的实践方面最好的专著。”

► **Tommy Usher**, *Embedded Systems Programming*

“这是我所读过的软件工程方面最好的书籍。”

► **Edward Kenworth**, *.Exe Magazine*

“该书必将成为一部经典的、所有开发人员及其管理者必备的读物。”

► **Peter Wright**, *Program Now*

Further Praise for **Code Complete**

“An excellent guide to programming style and software construction.”

—Martin Fowler, *Refactoring*

“Steve McConnell’s *Code Complete* . . . provides a fast track to wisdom for programmers. . . . His books are fun to read, and you never forget that he is speaking from hard-won personal experience.” —Jon Bentley, *Programming Pearls*, 2d ed.

“This is simply the best book on software construction that I’ve ever read. Every developer should own a copy and read it cover to cover every year. After reading it annually for nine years, I’m still learning things from this book!”

—John Robbins, *Debugging Applications for Microsoft .NET and Microsoft Windows*

“Today’s software *must* be robust and resilient, and secure code starts with disciplined software construction. After ten years, there is still no better authority than *Code Complete*.”

—Michael Howard, Security Engineering, Microsoft Corporation; Coauthor, *Writing Secure Code*

“A comprehensive examination of the tactical issues that go into crafting a well-engineered program. McConnell’s work covers such diverse topics as architecture, coding standards, testing, integration, and the nature of software craftsmanship.”

—Grady Booch, *Object Solutions*

“The ultimate encyclopedia for the software developer is *Code Complete* by Steve McConnell. Subtitled ‘A Practical Handbook of Software Construction,’ this 850-page book is exactly that. Its stated goal is to narrow the gap between the knowledge of ‘industry gurus and professors’ (Yourdon and Pressman, for example) and common commercial practice, and ‘to help you write better programs in less time with fewer headaches.’ . . . Every developer should own a copy of McConnell’s book. Its style and content are thoroughly practical.”

—Chris Loosley, *High-Performance Client/Server*

“Steve McConnell’s seminal book *Code Complete* is one of the most accessible works discussing in detail software development methods. . . .”

—Erik Bethke, *Game Development and Production*

“A mine of useful information and advice on the broader issues in designing and producing good software.”

—John Dempster, *The Laboratory Computer: A Practical Guide for Physiologists and Neuroscientists*

iv Further Praise for Code Complete

“If you are serious about improving your programming skills, you should get *Code Complete* by Steve McConnell.”

—Jean J. Labrosse, *Embedded Systems Building Blocks: Complete and Ready-To-Use Modules in C*

“Steve McConnell has written one of the best books on software development independent of computer environment . . . *Code Complete*.”

—Kenneth Rosen, *Unix: The Complete Reference*

“Every half an age or so, you come across a book that short-circuits the school of experience and saves you years of purgatory. . . . I cannot adequately express how good this book really is. *Code Complete* is a pretty lame title for a work of brilliance.”

—Jeff Duntemann, *PC Techniques*

“Microsoft Press has published what I consider to be the definitive book on software construction. This is a book that belongs on every software developer’s shelf.”

—Warren Keuffel, *Software Development*

“Every programmer should read this outstanding book.” —T. L. (Frank) Pappas, *Computer*

“If you aspire to be a professional programmer, this may be the wisest \$35 investment you’ll ever make. Don’t stop to read the rest of this review: just run out and buy it. McConnell’s stated purpose is to narrow the gap between the knowledge of industry gurus and common commercial practice. . . . The amazing thing is that he succeeds.”

—Richard Mateosian, *IEEE Micro*

“*Code Complete* should be required reading for anyone . . . in software development.”

—Tommy Usher, *C Users Journal*

“I’m encouraged to stick my neck out a bit further than usual and recommend, without reservation, Steve McConnell’s *Code Complete*. . . . My copy has replaced my API reference manuals as the book that’s closest to my keyboard while I work.”

—Jim Kyle, *Windows Tech Journal*

“This well-written but massive tome is arguably the best single volume ever written on the practical aspects of software implementation.”

—Tommy Usher, *Embedded Systems Programming*

“This is the best book on software engineering that I have yet read.”

—Edward Kenworth, *.EXE Magazine*

“This book deserves to become a classic, and should be compulsory reading for all developers, and those responsible for managing them.” —Peter Wright, *Program Now*

*To my wife, Ashlie, who doesn't have much to do with computer programming
but who has everything to do with enriching the rest of my life
in more ways than I could possibly describe*

Preface

The gap between the best software engineering practice and the average practice is very wide—perhaps wider than in any other engineering discipline. A tool that disseminates good practice would be important.

—Fred Brooks

My primary concern in writing this book has been to narrow the gap between the knowledge of industry gurus and professors on the one hand and common commercial practice on the other. Many powerful programming techniques hide in journals and academic papers for years before trickling down to the programming public.

Although leading-edge software-development practice has advanced rapidly in recent years, common practice hasn't. Many programs are still buggy, late, and over budget, and many fail to satisfy the needs of their users. Researchers in both the software industry and academic settings have discovered effective practices that eliminate most of the programming problems that have been prevalent since the 1970s. Because these practices aren't often reported outside the pages of highly specialized technical journals, however, most programming organizations aren't yet using them today. Studies have found that it typically takes 5 to 15 years or more for a research development to make its way into commercial practice (Raghavan and Chand 1989, Rogers 1995, Parnas 1999). This handbook shortcuts the process, making key discoveries available to the average programmer now.

Who Should Read This Book?

The research and programming experience collected in this handbook will help you to create higher-quality software and to do your work more quickly and with fewer problems. This book will give you insight into why you've had problems in the past and will show you how to avoid problems in the future. The programming practices described here will help you keep big projects under control and help you maintain and modify software successfully as the demands of your projects change.

Experienced Programmers

This handbook serves experienced programmers who want a comprehensive, easy-to-use guide to software development. Because this book focuses on construction, the most familiar part of the software life cycle, it makes powerful software development techniques understandable to self-taught programmers as well as to programmers with formal training.

Technical Leads

Many technical leads have used *Code Complete* to educate less-experienced programmers on their teams. You can also use it to fill your own knowledge gaps. If you're an experienced programmer, you might not agree with all my conclusions (and I would be surprised if you did), but if you read this book and think about each issue, only rarely will someone bring up a construction issue that you haven't previously considered.

Self-Taught Programmers

If you haven't had much formal training, you're in good company. About 50,000 new developers enter the profession each year (BLS 2004, Hecker 2004), but only about 35,000 software-related degrees are awarded each year (NCES 2002). From these figures it's a short hop to the conclusion that many programmers don't receive a formal education in software development. Self-taught programmers are found in the emerging group of professionals—engineers, accountants, scientists, teachers, and small-business owners—who program as part of their jobs but who do not necessarily view themselves as programmers. Regardless of the extent of your programming education, this handbook can give you insight into effective programming practices.

Students

The counterpoint to the programmer with experience but little formal training is the fresh college graduate. The recent graduate is often rich in theoretical knowledge but poor in the practical know-how that goes into building production programs. The practical lore of good coding is often passed down slowly in the ritualistic tribal dances of software architects, project leads, analysts, and more-experienced programmers. Even more often, it's the product of the individual programmer's trials and errors. This book is an alternative to the slow workings of the traditional intellectual potlatch. It pulls together the helpful tips and effective development strategies previously available mainly by hunting and gathering from other people's experience. It's a hand up for the student making the transition from an academic environment to a professional one.

Where Else Can You Find This Information?

This book synthesizes construction techniques from a variety of sources. In addition to being widely scattered, much of the accumulated wisdom about construction has resided outside written sources for years (Hildebrand 1989, McConnell 1997a). There is nothing mysterious about the effective, high-powered programming techniques used by expert programmers. In the day-to-day rush of grinding out the latest project, however, few experts take the time to share what they have learned. Consequently, programmers may have difficulty finding a good source of programming information.