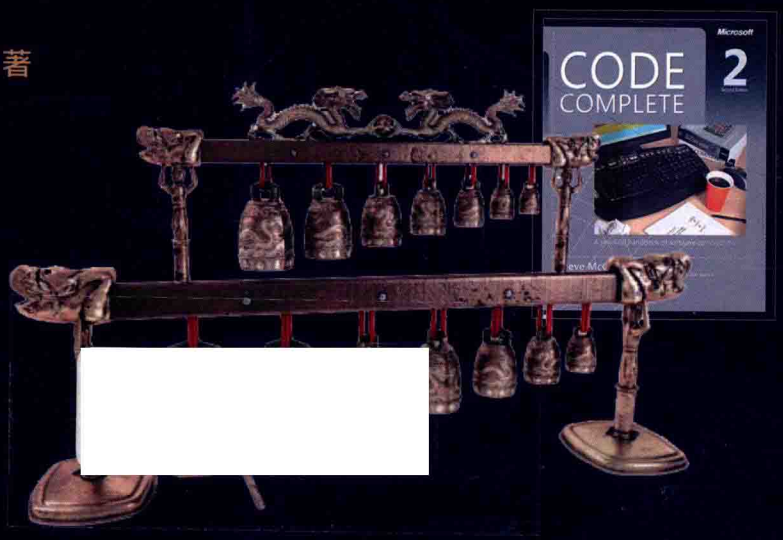
Broadview
www.broadview.com.cn
PEARSON

· 原味精品书系 ·

代码大全（第2版）（英文版）

Code Complete:
A Practical Handbook of Software Construction (2nd Edition)

[美] Steve McConnell 著



中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

· 原味精品书系 ·

代码大全（第2版）（英文版）

Code Complete:
A Practical Handbook of Software Construction（2nd Edition）

[美] Steve McConnell 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内容简介

本书是著名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。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。此外，本书还归纳总结了来自专家的经验、业界研究及学术成果，列举了大量软件开发领域的真实案例与统计数据。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。

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 =Code Complete: A Practical Handbook of Software Construction，2E：英文 /（美）迈克康奈尔（McConnell,S.）著. — 北京：电子工业出版社，2016.4
（原味精品书系）
ISBN 978-7-121-27315-5
I. ①代…II. ①迈…III. ①软件开发－英文 IV. ①TP311.52
中国版本图书馆CIP数据核字（2015）第233689号
策划编辑：张春雨　刘　芸
责任编辑：徐津平
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
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：59.5　　字数：1145千字
版　　次：2016年4月第1版
印　　次：2016年4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，联系及邮购电话：（010）88254888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
服务热线：（010）88258888。

对《代码大全（第2版）》的赞誉

“《代码大全（第2版）》是有关编程风格和软件构建的绝好指导书。”

▶ **Martin Fowler**,《重构》一书作者

“Steve McConnell 的《代码大全（第2版）》……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2版）》一书作者

“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，并且每年都要从头到尾读一遍。九年来我每年都读这本书，每次都能从中有新的收获。”

▶ **John Robbins**,《Microsoft .NET 和 Windows 应用程序调试》一书作者

“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。《代码大全（第1版）》出版后的十年里，没有出现比它更权威的书，直到现在《代码大全（第2版）》的出版。”

▶ **Michael Howard**,《编写安全的代码》一书作者

“《代码大全（第2版）》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* 一书作者

“对软件开发者而言，终极的百科全书就是 Steve McConnell 的《代码大全（第2版）》。这本厚达850页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、历经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* 一书作者

“Steve McConnell 的创新书籍《代码大全（第2版）》是详述软件开发方面最易懂的一本书……”

▶ **Erik Bethke**, *Game Development and Production* 一书作者

“《代码大全（第2版）》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* 一书作者

“如果你有意改进编程技术，就该有一本 Steve McConnell 的《代码大全（第 2 版）》。”

▶ **Jean J. Labrosse**, *Embedded Systems Building Blocks: Complete and Ready-To-Use Modules in C* 一书作者

“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* 一书作者

“每个时代都会有这样一本书，它为读者提供获得经验的捷径，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* 一书作者

“我认为 Microsoft 出版社出版的这本书是软件构建方面最好的书，每个软件开发人员的书架上都该有这本书。”

▶ **Warren Keuffel**, *Software Development* 一书作者

“每个程序员都该读读这本杰出的书籍。”

▶ **T.L. (Frank) Pappas**, *Computer* 一书作者

“假如你期望成为专业程序员，这将是你投资 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* 一书作者

“应当让在软件开发领域中的每个人都来读读《代码大全（第 2 版）》。”

▶ **Tommy Usher**, *C User's Journal* 一书作者

“我不遗余力地为 Steve McConnell 的《代码大全（第 2 版）》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* 一书作者

“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* 一书作者

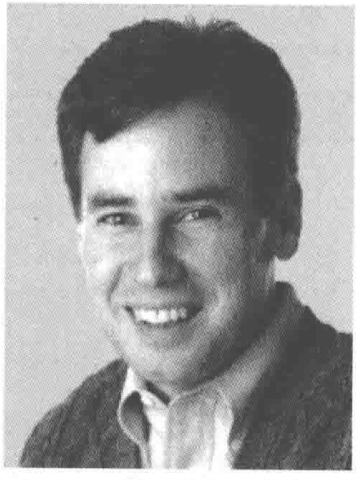
“这是我所读过的软件工程方面最好的书籍。”

▶ **Edward Kenworth**, *.Exe Magazine* 一书作者

“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* 一书作者

关于作者 Steve McConnell



Steve McConnell 是 Construx Software 公司的首席软件工程师，负责监管软件工程实践。他是软件工程知识体（SWEBOK）项目构建知识领域的领导。Steve 曾为微软、波音及西雅图地区的其他一些公司工作过。

Steve McConnell 是以下书籍的作者：1996 年的《快速软件开发》（*Rapid Development*）、1998 年的《软件项目生存指南》（*Software Project Survival Guide*）和 2004 年的《专业软件开发》（*Professional Software Development*）。他的优秀著作曾两度获得当年的《软件开发》（*Software Development*）杂志的优秀震撼大奖（Jolt Excellence award）。Steve 还曾是 SPC 评估专业版的开发领袖，软件开发生产力大奖（Software Development Productivity award）的获得者。1998 年《软件开发》的读者推选 Steve McConnell 为软件行业最有影响力的三个人之一，与 Bill Gates、Linus Torvalds 齐名。

Steve 从 Whitman 大学获学士学位，并从西雅图大学获软件工程硕士学位。他现居住在华盛顿的 Bellevue。

如果你对此书有任何疑问或评论，请与 Steve 联系，他的电子信箱是 stevemcc@construx.com，也可通过 www.stevemcconnell.com 与他联系。

前言

普通的软件工程实践与最优秀的软件实践差距巨大——多半比其他工程学科中的这种差距都要大。因此，传播优秀实践经验的工具是十分重要的。

—— Fred Brooks

我写这本书的首要目的，就是希望缩小本行业中一般商业实践与大师级人物及专家们之间的知识差距。许多强大的编程技术在被编程领域的大众接触之前，都已在学术论文和期刊里尘封了多年。

虽然近年来前卫的软件开发实践迅速发展，但普通的实践手段并没有太大变化。很多程序的开发仍然是漏洞百出、迟于交付并且超出预算，还有很多根本就无法满足用户的需求。软件业界及学术界的研究人员已经发现了不少行之有效的实践经验，足以解决自 20 世纪 70 年代以来编程领域中日益蔓延的大多数问题。可是这些实践经验很少在高度专业化的技术期刊之外对外发表，所以时至今日大多数编程的机构和组织还没能用上这些技术。有研究表明，一项研发成果从其诞生之日起，到进入商业实践阶段，通常要经历 5～15 年甚至更长的时间（Raghavan and Chand 1989，Rogers 1995，Parnas 1999）。这本手册就是想缩短这一漫长的过程，让那些关键性的研发成果现在就能为更多编程人员所用。

Who Should Read This Book
谁应当阅读本书

本书中所汇集的研究成果和编程经验，将帮助你创建更高质量的软件，使你能更快速地进行开发，遇到的问题更少。本书将帮你弄明白过去为什么会遇到那些问题，并告诉你如何在将来避免它们。这里所描述的编程实践将帮助你掌控更大型的项目，还能在项目的需求发生变动时帮助你成功地维护并修改已经开发出来的软件。

Experienced Programmers
经验丰富的程序员

对于经验丰富的程序员而言，本书正是他们想要的一本翔实、易用的软件开发指南。本书关注的是“构建（construction）”，即整个软件生命周期中最为人熟知的部分；本书把强大的软件开发技术写得让自学的程序员和参加过正规训练的程序员都能读懂。

Technical Leads
技术领导

许多技术领导（或者说是技术带头人）都曾在他们的团队中使用《代码大全（第1版）》来培训经验不足的程序员。当然，本书也可以用来填补你自己的知识缺陷。如果你是一位经验丰富的程序员，你不一定会同意我给出的所有结论（如果不是这样，我倒会觉得奇怪）。但如果你阅读本书并思索其中的每一个问题之后，那么几乎不会有人再能提出什么你未曾思考过的软件构建方面的问题了。

Self-Taught Programmers
自学的程序员

如果你没有受过太多的正规训练，本书正是你的良伴。每年约有 50 000 个新手进入这一专业领域（BLS 2004, Hecker 2004），但每年却只有 35 000 个人获得与软件相关的学位（NCES 2002）。从这些数据中我们可以很快得出一个结论——很多程序员并没有接受过软件开发方面的正规教育。在许多新兴的专业人员社群中都可以看到自学的编程人员——工程师、会计师、科学家、教师及小公司的老板们。编写程序是他们工作的一部分，但他们并不一定把自己看作是程序员。无论你在编程方面受过何种程度的教育，本手册都能让你对各种行之有效的编程实践有深入的了解。

Students
学生

与有经验但缺乏正规培训的程序员对应的，是那些刚刚毕业的大学生。新近毕业的学生大多拥有丰富的理论知识，但却缺乏创建产品级的程序（production programs）的实践技术。关于编写优秀代码的实践知识，就像部落里祭祀仪式上的舞蹈一样，只能慢慢地从软件架构师、项目负责人、分析师及更有经验的程序员那里传承下来。更多的时候，这些知识就是程序员个人反复的尝试和犯错后的结晶。本书则是这些缓慢、传统智慧传承方式的一种替代方案，它汇集了以往只能从他人经验中猎取和收集的大量实用的经验技巧和有效的开发策略。对于那些正在从学术环境转向专业环境的学生来说，这是一本必备的读物。

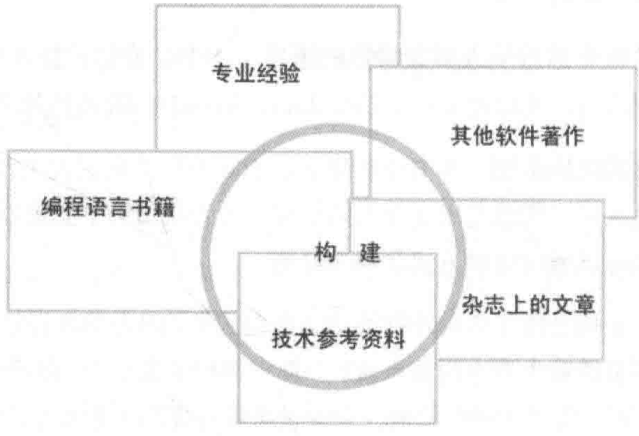
Where Else Can You Find This Information
还能从何处找到这些信息

本书综合整理了来自四面八方的多种软件构建技术。这些技术是软件构建领域长年累月积聚下来的智慧财富，它们不仅分散，而且其中大部分素材常年散落于纸面之外（Hildebrand 1989，McConnell 1997a）。其实，内行的程序员们所用的那些强大有效的编程技术并不神秘。但是这些内行人士面对手头日复一日紧张冲刺的项目，几乎没有谁花些时间和大家分享他们所学到的知识和技能。因此，程序员们可能很难找到

很好的关于编程的信息来源。

而本书所描述的技术则填补了入门图书和高级编程图书之间的空白。当你读过了“Java 编程入门”、“高级 Java 编程”和“超高级 Java 编程”之后，如果你还想学更多的编程知识，那还能读点什么呢？你可以阅读 Intel 或 Motorola 的硬件参考手册，阅读 Microsoft Windows 或 Linux 操作系统的函数手册，甚至是去阅读讲另外一门编程语言的书籍——你确实无法在一个缺乏这种详细参考资料的环境中使用语言或者程序。但本书是为数不多的探究编程本质的书籍之一。无论你在何种环境下、用何种语言编写程序，书中某些最有益处的编程技术都能派上用场。其他的书一般都忽略了这些实践知识，而这也正是本书专注于这些知识的原因。

本书中的信息是从许多来源中提炼出来的，如下图所示。想完全获得在本书中看到的这些信息的另外途径只有一条，那就是通读堆积如山的书籍和成百上千本技术期刊，还得再加上大量的实际经验。即便你把这些事情都做到了，本书仍然会对你很有益处，因为它把所有这些资料都集于一处，便于查阅。



Key Benefits of This Handbook
阅读本书的收益

无论你是何种背景，本书都能助你在更短的时间内写出更棒的程序，还不会那么头疼。

全面的软件构建参考 本书讨论了软件构建活动的方方面面，比如说软件的质量，还有编程的思维方式。它还会深入阐述构建活动中的重要细节，如创建一个类的步骤，使用数据和控制结构时的各种事项，还有调试、重构、代码调优的技术与策略等。你无须逐页通读所有主题。本书可以让你很容易就能找到感兴趣的特定话题。

随时备用的核对表 本书包括了大量的核对表（checklist），你可以用它来评估软件架构、设计方法、类和子程序的质量、变量命名、控制结构、代码格式、测试用例，等等。

与时俱进的信息 本书介绍了一些当今最为时兴的技术，其中有许多还未被广泛采用。正因为本书撷取了实践与研究两者的精髓，它所介绍的这些技术将经久不衰，受用多年。

以更广的视角检视软件开发 本书将给你一个机会，让你凌驾于日复一日、忙于救火的混乱场面之上，看看到底什么是可行的，而什么又是不可行的。实践中的程序员们很少有时间去阅读数以百计的书籍与期刊，而本手册萃取了其中的精华。本书所汇集的理论研究与实践经验将活跃你的思维，激励你对自己项目的思考，使你的行动更有策略，避免反复陷入完全一样的战斗。

绝不注水 有些软件书籍，其中精髓部分的净重也就 1 克，却注入了重达 10 克的水分。本书则会公平地探讨每项技术的优劣。关于你自己项目的特定需求，你了解得要比任何人都清楚。因此本书仅是给你公正客观的信息，让你能够具体情况具体分析，做出正确的决策。

有关概念适用于大多数常见的语言 本书中介绍的技术能让你可以更好地利用你的编程语言，无论是 C++、C#、Java、Visual Basic，还是其他类似语言。

丰富的代码示例 本书中收集了近 500 个用于展现优劣代码之差异的示例。之所以给出这么多示例也是出于个人的偏好。因为从示例中我最能学到东西，我想其他程序员也该可以通过这种方式学得更好吧。

这些示例是用了多种不同的语言所写成的，因为学习并掌握不止一门语言通常是专业程序员职业生涯中的分水岭。一旦一名程序员意识到编程原则是超越特定语言语法的东西时，通往能够实质地改善编程质量并提高工作效率的知识的大门也就向他敞开了。

为了避免以多种语言写成的例子成为读者的负担，我会尽量避免使用各语言中那些深奥的特性——除非当时就是需要探讨它。为了弄懂一个代码片段要表达的问题，你无须完全理解所有的细枝末节。如果你集中关注示例所展示的问题，那么无论它是用什么语言写成的，你都能读懂。为让你更容易理解这些示例，我还给其中的关键部分加了注解。

引用其他信息来源 本书汇集了为数众多关于软件构建方面的可用信息，但这并不算完。在本书所有的章节中，“更多资源（Additional Resources）”一节都会介绍其他一些书籍和文章，你希望进一步深入了解感兴趣的话题时可以阅读它们。

cc2e.com/1234 **配套网站** 在本书的配套网站 cc2e.com 上会提供更新的核对表、参考书目、杂志文章、网页链接等内容。要访问《代码大全（第 2 版）》中的相关信息，请如本段文字左侧所示，在浏览器中输入“cc2e.com/”，后跟一个四位阿拉伯数字即可。这样的网址参考链接在本书中会有很多。

Why This Handbook Was Written
为什么要写这本手册

在软件工程界，人们都清楚地认识到，应该把软件开发中行之有效的实践知识归纳、编撰成一本开发手册。计算机科学与技术委员会（Computer Science and Technology Board）的一份报告指出，要想大幅提高软件开发的质量和工作效率，需要把已知的行之有效的软件开发实践知识归纳、统一并广为传播（CSTB 1990, McConnell 1997a）。该委员会还指出，传播这些知识的策略应建立在软件工程手册这个概念的基础之上。

The Topic of Construction Has Been Neglected
软件构建的话题常被忽视

曾几何时，软件开发和编写代码被认为是同一件事情。但随着软件开发周期中的各个活动被人们逐渐认识，该领域中一些最棒的头头脑脑们就开始花更多时间去分析和争论诸如项目管理方法、需求、设计、测试等问题了。在这场学习研究新兴领域的浪潮中，代码构建这个与软件开发骨肉相连的环节反而被忽视了。

关于软件构建的讨论之所以步履蹒跚，也是因为有人认为，如果将构建活动视作软件开发中的一项特定**活动**，就暗示着也必须把它视作其中的一个特定**阶段**。然而实际上，软件开发中的各项**活动**和各个**阶段**无须以特定的关系一一对应起来；而且无论其他的软件活动是分阶段（phase）进行，还是迭代式（iteration）进行，或者以某种其他方式进行，都不妨碍我们探讨“构建活动”。

Construction Is Important
构建活动是重要的

构建活动被学者和作者所忽略的另一个原因是源于一个错误的观念，他们认为与其他软件开发活动相比，构建是一个相对机械化的过程，并没有太多可改进的机会。然而事实并非如此。

“代码构建”一般占据了小型项目 65%的工作量，而在中型项目上也达到了 50%。同时，“构建”也要为小型项目中 75%的错误负责，在中到大型项目上这一比例为 50%~75%。任何一个要为 50%~75%的错误负责的活动环节显然都是应该加以改善的。

（第 27 章中对这些统计数据有更多详细的探讨。）

也有一些评论家指出，虽然构建阶段发生的错误在所有错误中占有很大的比例，但修正这些错误的代价往往比“修正那些由于需求和架构所导致的错误”要低很多，这也就暗示着构建活动因此不那么重要。诚然，修正由构建活动所导致的错误的代价比较低这一说法是正确的，但它也引起了误导——因为如果不修正这些错误，代价反而会高得令人难以置信。研究人员发现，软件中一些代价最为昂贵的错误，其罪魁祸首常常是一些小范围的代码错误，其代价甚至可以飙至上亿美元的程度（Weinberg 1983，SEN 1990）。可以用较低代价修正的错误，并不意味着这些错误的修正不重要。

人们忽视构建活动的另一种原因则颇具讽刺意味——就因为它是软件开发中唯一一项肯定能完成的活动。对于需求，人们可以自以为是而不去潜心分析；对于架构，人们可以偷工减料而不去精心设计；对于测试，人们可以短斤少两甚至跳过不做，而不去整体计划并付诸实施。但只要写出来的是程序，总归要进行构建活动，这也说明，只要改进软件构建这一环节，就一定对软件开发实践有好处。

No Comparable Book Is Available
没有可媲美的同类书籍

既然看到构建活动有着如此清晰的重要性，我曾相信，当我构思此书时已有人写过关于有效的软件构建实践的书籍了。对这样一本介绍如何有效地进行编程的书籍的需求是显而易见的，但是我却只找到很少几本关于软件构建这一题材的书，而且那些书也仅仅是涉及这个话题的一部分罢了。有些书写于 15 年前，还是和一些深奥的语言——如 ALGOL、PL/I、Ratfor 及 Smalltalk 等——紧密相关的。有些则是出自并不实际编写产品代码的教授之手。教授们写出来的技术内容对于学生们的项目而言还行得通，但他们通常不知道如何在完整规模的开发环境中施展这些技术；还有些书是为了宣传作者最新钟情的某种方法论，却忽略了那些被时间反复证明是行之有效的成熟实践技术的巨大宝库。

当艺术评论家聚在一起的时候，他们谈论的都是关于版式、结构及意蕴之类的话题；而当真正的艺术家聚在一起的时候，他们谈论的则是到哪儿才能买到更便宜的松节油。
—Pable Picasso
（毕加索）

简而言之，我没有找到哪怕是一本试图归纳总结来自专家经验、业界研究及学术成果的实践编程技术的书籍。关于这个话题的讨论要能和现今的编程语言、面向对象编程技术及前沿的开发实践紧密结合。很明显需要有人写出一本这样的书出来，而他必须了解当今的理论发展水平，同时也编写过足够多的能反映实践状况的产品级代码。我把本书构思成关于代码构建活动的完整探讨——一个程序员给其他的程序员写的书。

Author Note
作者注

欢迎你对本书中所探讨的话题进行质询，如果你整理了勘误报告，或其他相关的内容，请发邮件与我联系，我的邮箱是 stevemmc@construx.com，也可以访问我的网站：www.SteveMcConnell.com。

Bellevue, Washington
Memorial Day, 2004

Microsoft Learning Technical Support

Every effort has been made to ensure the accuracy of this book. Microsoft Press provides corrections for books through the World Wide Web at the following address:

http://www.microsoft.com/learning/support/

To connect directly to the Microsoft Knowledge Base and enter a query regarding a question or issue that you may have, go to:

http://www.microsoft.com/learning/support/search.asp

If you have comments, questions, or ideas regarding this book, please send them to Microsoft Press using either of the following methods:

Postal Mail:

Microsoft Press
Attn: Code Complete 2E Editor
One Microsoft Way
Redmond, WA 98052-6399

E-mail:

mspinput@microsoft.com

致谢

从来没有哪本书是完全由一个人写出来的（至少我的书都是如此）。《代码大全（第 2 版）》更是许多人智慧的结晶。

我想感谢对本书提出重要意见的人：Hákon Ágústsson、Scott Ambler、Will Barns、William D.Bartholomew、Lars Bergstrom、Ian Brockbank、Bruce Butler、Jay Cincotta、Alan Cooper、Bob Corrick、Al Corwin、Jerry Deville、Jon Eaves、Edward Estrada、Steve Gouldstone、Owain Griffiths、Matthew Harris、Michael Howard、Andy Hunt、Kevin Hutchison、Rob Jasper、Stephen Jenkins、Ralph Johnson 及其在伊利诺伊斯大学的软件架构研究组、Marek Konopka、Jeff Langr、Andy Lester、Mitica Manu、Steve Mattingly、Gareth McCaughan、Robert McGovern、Scott Meyers、Gareth Morgan、Matt Peloquin、Bryan Pflug、Jeffrey Richter、Steve Rinn、Doug Rosenberg、Brian St.Pierre、Diomidis Spinellis、Matt Stephens、Dave Thomas、Andy Thomas-Cramer、John Vlissides、Pavel Vozenilek、Denny Williford、Jack Woolley 和 Dee Zsombor。

几百名读者为第 1 版发来了意见，对第 2 版给出评论的读者就更多了。感谢大家花时间以各种方式交换阅读本书的心得。

特别要感谢审定最终稿件的 Construx 软件公司同仁：Jason Hills、Bradey Honsinger、Abdul Nizar、Tom Reed 和 Pamela Perrott。虽然知道稿件之前有过许多次的仔细检查，你们仍然进行了彻底地审阅，认真程度着实令我吃惊。还要感谢 Bradey、Jason 和 Pamela 为 cc2e.com 网站付出的辛劳。

和本书的责任编辑 Devon Musgrave 共同工作，是一件尤为愉快的事。我曾和许多了不起的编辑在其他项目上合作过，Devon 更是少有的负责和容易相处。谢谢你，Devon！谢谢 Linda Engleman，是你支持着第 2 版的出版。没有你就没有《代码大全（第 2 版）》。还要谢谢 Microsoft 出版社的其他员工，包括 Robin Van Steenburgh、Elden Nelson、Carl Diltz、Joel Panchot、Patricia Masserman、Bill Myers、Sandi Resnick、Barbara Norfleet、James Kramer 和 Prescott Klassen。

我想向 Microsoft 出版社的下列员工致意，感谢你们为《代码大全》第 1 版的出版付出的汗水：Alice Smith、Arlene Myers、Barbara Runyan、Carol Luke、Connie Little、Dean Holmes、Eric Stroo、Erin O'Connor、Jeannie McGivern、Jeff Carey、Jennifer Harris、Jennifer Vick、Judith Bloch、Katherine Erickson、Kim Eggleston、Lisa Sandburg、Lisa Theobald、Margarite Hargrave、Mike Halvorson、Pat Forgette、Peggy Herman、Ruth Pettis、Sally Brunsman、Shawn Peck、Steve Murray、Wallis Bolz 和 Zaafar Hasnain。

感谢以下审阅者，感谢你们为第 1 版做出了突出贡献：Al Corwin、Bill Kiestler、Brian Daugherty、Dave Moore、Greg Hitchcock、Hank Meuret、Jack Woolley、Joey Wyrick、Margot Page、Mike Klein、Mike Zevenbergen、Pat Forman、Peter Pathe、Robert L.Glass、Tammy Forman、Tony Pisculli 和 Wayne Beardsley。特别要感谢 Tony Garland，谢谢你详尽的评论：12 年后再来回顾，我越发感激 Tony 那几千条评论是多么地难得。

目录