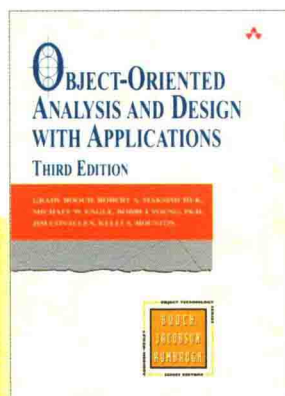




面向对象 分析与设计

第3版
修订版

Object-Oriented Analysis and
Design with Applications (Third Edition)



Grady Booch
Robert A. Maksimchuk
Michael W. Engle 著
Bobbi J. Young, PH.D.
Jim Conallen
Kelli A. Houston
王海鹏 潘加宇 译



Jolt 大奖精选丛书

面向对象 分析与设计

第3版
修订版

Object-Oriented Analysis and
Design with Applications (Third Edition)

Grady Booch
Robert A. Maksimchuk
Michael W. Engle
[美] Bobbi J. Young, PH.D. 著
Jim Conallen
Kelli A. Houston
王海鹏 潘加宇 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



内 容 简 介

Jolt 大奖素有“软件业之奥斯卡”的美称，本丛书精选自 Jolt 历届获奖图书，以植根于开发实践中的独到工程思想与杰出方法论为主要甄选方向。本书第 1 版和第 2 版分别于 1991 年和 1993 年荣获 Jolt 震撼奖。

本书是 UML 创始人 Grady Booch 的代表作之一。

全书分为理论和应用两部分。理论部分深刻剖析了面向对象分析与设计 (OOAD) 的概念和方法。应用部分连续列出了 5 个不同类型、不同领域的应用，描述如何从初始阶段到交付阶段，将 OOAD 理论和方法应用到项目中。应用部分所涉及的领域包括系统架构、数据获取、密码分析、控制系统和 Web 开发，还给出了一些关于重要问题的有效建议，包括分类、实现策略和高性价比的项目管理。书中介绍的概念都基于牢固的理论基础。同时，作者十分注重实效，基于其丰富的经验，面向软件工程实践者的实际需要，提出了改进的对象开发方法，用于解决系统和软件开发者面临的复杂问题；运用大量例子说明了基本概念，解释了方法，并展示了在不同领域的成功应用。

书中的表示法采用最新的 UML 2.0，因此本书是学习 UML 2.0 不可多得的参考书。本书非常适合实际系统和软件的开发、系统分析师或架构师、项目经理阅读，也可以作为高等院校软件工程和高级编程课程的教材使用。

Authorized translation from the English language edition, entitled OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APPLICATIONS, 3E, 9780201895513 by BOOCH, GRADY; MAKSIMCHUK, ROBERT A.; ENGLE, MICHAEL W.; YOUNG, BOBBI J.; CONALLEN, JIM; HOUSTON, KELLI A., published by Pearson Education, Inc., publishing as Addison-Wesley Professional, Copyright © 2007 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2017.

本书简体中文版专有版权由 Pearson Education 培生教育集团亚洲有限公司授予电子工业出版社。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育集团激光防伪标签，无标签者不得销售。

版权贸易合同登记号图字：01-2012-4068

图书在版编目 (CIP) 数据

面向对象分析与设计：第 3 版：修订版 / (美) 布奇 (Booch, G.) 等著；王海鹏，潘加宇译。—北京：电子工业出版社，2016.5

(Jolt 大奖精选丛书)

书名原文：Object-Oriented Analysis and Design with Applications (Third Edition)

ISBN 978-7-121-28666-7

I. ①面… II. ①布… ②王… ③潘… III. ①面向对象语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 089337 号

策划编辑：张春雨 符隆美

责任编辑：刘 舫

印 刷：北京中新伟业印刷有限公司

装 订：三河市华成印务有限公司

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：38 字数：850 千字

版 次：2012 年 7 月第 1 版 (原著第 3 版)

2016 年 5 月第 2 版

印 次：2016 年 5 月第 1 次印刷

印 数：3000 册 定价：109.00 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, 88258888。

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。

本书咨询联系方式：010-51260888-819 faq@phei.com.cn。

经久不息的回荡

今时的读书人，不复有无书可读之苦，却时有品种繁多而无从择优之感，甚而专业度颇高的技术书领域，亦日趋遭逢乱花迷眼的境地。此时，若得觅权威书评，抑或有公信力的排行榜，可按图索骥，大大增加选中好书的命中率。然而，如此良助，不可多得，纵观中外也唯见一枝独秀——素有“软件业奥斯卡”之美誉的 Jolt 奖！

震撼世界者为谁

在计算设备已经成为企业生产和日常生活之必备工具的今天，专业和大众用户对于软件的功能、性能和用户体验的要求都在不断提高。在这样的背景下，如何能够发挥出软件开发的最高效率和最大效能，已经是摆在每一个从业者面前的重大课题，而这也正是 Jolt 大奖横空出世的初衷及坚持数年的宗旨。

Jolt 大奖历时 20 余年，在图书及软件业知名度极高，广受推崇。奖如其名，为引领计算机科学与工程发展主流，Jolt 坚持将每年的奖项只颁给那些给整个 IT 业界带来震撼结果的图书、工具、产品及理念等，因一流的眼光及超高的专业度而得以闻名遐迩，声名远播。

除图书外，Jolt 针对软件产品设有诸多奖项分类，如配置管理、协作工具、数据库引擎/数据库工具、设计工具/建模、开发环境、企业工具、库/框架、移动开发工具等。但图书历来是 Jolt 大奖中最受瞩目且传播最广的一个奖项分支。Jolt 曾设有通用类图书、技术类图书等分类，每个分类又设有“卓越奖”（Jolt Award，一般为一个）和“生产力奖”（Productivity Award，一般为 2 或 3 个）。

获奖技术图书一经公布，即打上经典烙印，可谓一举“震撼全世界”（赞助商 Jolt 可乐的广告词）。

作为计算机技术图书的后进，我们总在追问——是谁在震撼世界，是谁在照亮明天？Jolt 大奖恰似摆在眼前的橱窗，让我们可以近距离观看潮流在舞蹈，倾听震撼在轰鸣！

朝花夕拾为哪般

Jolt 像是一年一度的承诺，在茫茫书海中为我们淘砺出一批批经得起岁月冲刷的杰作，头顶桂冠的佳作也因而得以一批批引进中国，为国人开阔了眼界，滋补了技术养分。然而，或因技术差距造就的生不逢时、水土不服，或因翻译、制作的不如人意，抑或是疏于宣传等诸多原因，这些经典著作在国内出版后，尽管不乏如获至宝的拥趸，却仍不为诸多人所知，从而与大量本应从中获益的读者擦肩而过。既然这生生错失的遗憾本不该发生，则更不应延续。为此，我们邀国外出版同行、国内技术专家一道，踏上朝花夕拾之路，竭力为广大读者筛选出历久弥新、震撼依旧的 Jolt 图书精品。

Jolt 获奖图书皆由业界专家一致评出，并得到软件从业人员的高度认可，虽然这些书今天读来，不再能看到 20 世纪史诗时代那般日新月异的理论突破，以及依赖于高深繁复的科学研究所取得的系统化成果，更多是在日复一日的开发实践中总结和提炼出来的工程思想和方法论。重新选材之所以有所弃取，从 Jolt 多年来的评奖规律中可窥端倪——

一万小时真理见

凡是在工程思想领域取得革命性、颠覆性突破的图书，就被归于“震撼”获奖分类。比如，从基于过程的程序设计模型过渡到面向对象的全新模型，就是软件开发思想上的一次带来巨大震撼的革命；再比如，打破传统的瀑布模型而转向持续集成的软件交付模型，这也是一场业界的重大思想转变。像这样的重大思想突破，可以说是数年甚至数十年一遇的，而荣获 Jolt 大奖的图书中更为常见的，则是基于最佳实践的“生产效率”获奖者。获得此类殊荣的图书，都是作者们从平凡的、重复的，甚至用一般人的眼光看来不怎么起眼的日常开

发实践中，以独具的慧眼、过人的耐心和大胆的创新，闯开一条不平常道路的心血与经验总结。

这些图书所涉及的主题，都是普通的软件开发人员每天要面对的工作——代码阅读、撰写测试用例、修复软件问题……但就是这样貌似平淡无奇的工作，是否能每一天、每一个项目都做好，着实拉开了软件开发人员素质的差距，也决定了软件企业开发出来的产品和服务的质量。我们中国有一句古话，叫作熟能生巧；某位著名企业家也说过一句家喻户晓的名言：“把简单的事千百万次地做好，就是不简单的。”这些朴素而实际的真理，同样也是本套丛书最能彰显的所谓程序员精神。它建立在脚踏实地的实践基础之上，也充满了对于自由和创新的向往。

名作可堪比名曲

就不因岁月流逝而褪色来说，与这些 Jolt 名作相媲美者，只有那些百年响彻、震撼古今的经典名曲。希望本丛书带给大家的每部著作，也如百听不厌的乐曲，掩卷良久方余音绕梁，真知存心。仔细想来，软件开发与古典音乐岂非有异曲同工之妙？既是人类心智索问精确科学的探究，亦是寻觅美学享受的追求。工程是艺术的根基，而艺术是工程的极致。衷心地希望各位读者能够认真阅读本丛书的本本珍品，并切实地用于自己的日常工作中，在充分享受大师魅力的同时，为中国的软件事业谱写更多、更震撼的乐章。

谨以本书献给 Jan:

我的朋友，我的爱人，我的妻子。

——Grady

重看面向对象

软件的本质是什么？从不同的角度来看，会有不同的答案。有人认为是程序加文档，有人认为是人机交互，有人认为是增删改查，有人认为是抽象模型，而我认为算法。

计算机科学的基石是图灵机抽象：一个输入集合，一个输出集合，一个内部状态集合，一个计算规则集合。这个抽象十分强大，我们甚至可以认为一头奶牛也是一个图灵机：吃的是草，挤的是奶。

一个表达式也是一个图灵机，其中的操作数是输入，求值的结果是输出。一个函数也是图灵机，参数是输入，返回值是输出。编程或设计软件就是在通用图灵机的基础上，设计一个具体的图灵机。我们设计软件接受怎样的输入，设计软件内部的状态，设计表示计算规则的代码，设计软件的输出。

近年来逐渐流行的函数式编程，就是建立在这个抽象的基础上。而且函数式编程的思想由来已久，可以追溯到最古老的高级语言之一：Lisp。这种思想非常简单：给定一个输入集合，经过函数的处理，给出一个输出集合。由此也导出了 Map-Reduce 等流行的架构设计模式：一个计算集群仍然是一个图灵机。

纯粹的函数有一点不足，它没有内部状态。可以说，它是简化了的图灵机。但在有些时候，我们确实需要内部状态。根据内部状态的不同，对于同样的输入，可能给出不一样的输出。于是便有了闭包的概念，它是一个函数加上相关的上下文环境状态。这样，我们可以毫无困难地构建任何具体的图灵机（好吧，正确实现一个算法还是比较困难的）。

闭包可以看成是拥有内部状态的函数，这就相当于一个简单的对象，它只有一个方法。反过来，对象可以看成是几个闭包，它们共享了内部状态。所以有人说：闭包是懒人的对象，对象是懒人的闭包。因此，函数式编程和面向对

象思想，在底层基础上是一致的。

面向对象思想的历史和函数式一样久远。实际上，它们都是我们在设计算法时的一种抽象。只有利用抽象概念，才能实现人与人之间的沟通。“你想吃苹果吗？”这里的“苹果”就是一个抽象概念，它隐藏了苹果实现的许多细节。人的大脑喜欢工作在一组抽象概念上。名词是结构或存在的抽象，动词是行为或过程的抽象。

我们在设计算法时，既需要函数抽象，也需要对象抽象。今天，面向对象和函数式编程的思想在各种编程语言中融合，可以说是殊途同归。

抽象是强大的工具，但用得不好，也会产生不良的后果。最重要的问题，就是创建太多不必要的抽象。毕竟，抽象只是我们脑中的概念，我们可以创造出任何概念。比如上帝和各种鬼神，直到科学家说，在科学的系统里不需要假设存在一个上帝。面向对象在这方面遇到的问题比较多。举例来说，一个 Java 的 Hello world 程序，就要涉及好几个概念，直接导致程序的代码比较长。而在函数式编程中，这通常只是一次函数调用。又比如，在一个使用 Struts、Spring、Hibernate 构建的 Java Web 应用程序中，处理一个 Get 请求的调用栈，可能是长长的一串。数据在不同的概念抽象之间反复倒手，白白浪费了计算资源。

任何两种观点都是互补的。面向对象思想在过去的软件开发中取得了辉煌的成绩。函数式编程让我们能从另一个角度审视面向对象，更进一步体会面向对象抽象的强大，也发现面向对象中一些误用的地方。如无必要，勿增实体。也许我们不需要假设以太的存在，就能解释光在真空中的传播。

多年后重读这本书，促使我重新思考，需要利用哪些抽象来设计我的算法。

这些年来，这本书让我受益良多，再次向大家郑重推荐。

王海鹏

2016 年 1 月 5 日

特别的Booch，特别的书

以下几点也许是您购买本书的理由。

1. 读过 Robert C. Martin 的 *Agile Principles* 的读者，很容易注意到该书前言的第一句话，“Bob，你说过去年就能写完这本书的——Claudia Frers 在 1999 年 UML World 大会上抱怨。”“这本书”指的是 Robert C. Martin 在 20 世纪 90 年代的代表作 *Designing Object-Oriented C++ Application using the Booch Method*。*Agile Principles* 本来是作为 *Designing* 一书的第 2 版。由此可见，Robert C. Martin 深受 Booch 的影响。

2. Grady Booch 是最早提出面向对象分析设计方法的方法学家之一。20 多年来，Grady Booch 一直担任 Rational 公司的首席科学家，随着 Rational 成长，收购别人，被别人收购，CEO 换了又换，他也没有离开。Grady Booch 是 UML 三友中唯一的 IBM 院士。

3. Grady Booch 非常“与时俱进”，其 Blog（<http://www.ibm.com/developerworks/blogs/page/gradybooch>）更新频繁，即使在心脏主动脉瘤手术期间，他也一直在病床上更新 Blog。他还热衷于 Second Life，认为 Second Life 是一条协作软件开发的新道路。Grady Booch 在 Second Life 上的化身叫作 Alem Theas。Dr. Dobb 在给他颁发程序设计杰出奖（Excellence in Programming Award）的时候，就是在 Second Life 上进行的。

4. 本书是真正展示 Booch 思想的书。我们可以在 UML 三友署名的 UML 系列书籍封面上看到 Grady Booch 的名字，但其中大多数工作是由 James Rumbaugh 完成的。

5. 本书分为理论和应用两部分。理论部分（概念和方法）的叙述朴实无华，标题简洁：“复杂性”、“对象模型”、“类和对象”、“分类”、“表示法”、“过

程”……用平实的语言把内容剖析得很透彻。应用部分连续列出了 5 个不同类型、不同领域的应用，描述如何从初始阶段到交付阶段，把前面所授方法应用到项目中。

- 基于卫星的导航系统：聚焦于系统架构；
- 列车交通控制系统：聚焦于系统需求；
- 人工智能解密系统：聚焦于分析；
- 气象站数据采集系统：聚焦于分析到初步的设计；
- 雇员休假跟踪 Web 应用系统：聚焦于详细设计和实现。

6. 书中的表示法采用最新的 UML 2.0，画图工具是 IBM Rational Software Architect 和 Sparx Systems Enterprise Architect。也就是说，Grady Booch 使用了非 IBM 公司的 UML 工具来写自己的书。

潘加宇

前言

人类渴望得到精神上的宁静、美学上的成就、家庭的安全、正义和自由。这一切都不能通过工业化的生产效率来直接满足。但是，生产效率让人们得到充足的物质享受，而不至于与匮乏苦苦斗争。这为精神、美学和家庭事务赢得了时间，也使得社会能够将一些特殊的技能赋予司法机构以及维护权利的机构。

Harlan Mills

DPMA and Human Productivity

作为计算机专业人员，我们努力地去构建能工作而且有用的系统；作为软件工程师，我们面临着在计算资源和人力资源有限的条件下创建复杂系统的任务。面向对象（OO）技术已经发展为管理许多不同种类的系统内复杂性的手段。对象模型已被证明是非常有力和统一的概念。

对第 2 版的改动

在本书第 2 版出版以后，我们看到了一些重要的技术进步，其中一些突出的进步如下。

- 与因特网的高带宽、无线连接已经非常普遍；
- 纳米技术已经出现，并开始提供有价值的产品；
- 机器人在火星表面漫步；
- 计算机生成的特效使得在电影中能够完全逼真地再现任何想象中的世界；
- 出现了个人气垫船；
- 手机已无处不在，使用非常方便；
- 获得了人类基因图谱；

- 面向对象技术已经在工业软件开发中成为主流技术。

在世界各地都能见到面向对象技术被使用。但是，我们仍然遇到许多人，他们还没有采用面向对象的开发方式。对于这两类人，本书的新版本都很有价值。

对于面向对象分析与设计（OOAD）的新手，本书提供了下列信息。

- 面向对象的概念支持和演进式的观点；
- 如何在系统开发生命周期中应用 OOAD 的例子；
- 对系统和软件开发中使用的标准表示法统一建模语言（UML 2.0）的介绍。

对于有经验的 OOAD 实践者，本书从不同的角度提供了价值。

- 即使对于有经验的实践者，UML 2.0 也是新的。这是可以看到的表示法方面的重要区别。
- 根据前一版本所收到的反馈，更加关注建模。
- 通过本书的概念部分的学习，可以了解在面向对象的世界中，“为什么事情总是像它们现在的样子”。许多人可能从没研究过面向对象（OO）概念本身的发展，即使有所了解，在初次学习 OO 方法时，也许未能理解其重要性。

本书这一版和以前的版本相比有四项主要区别，如下所示。

1. UML 2.0 已经正式得到了通过，第 5 章将介绍 UML 2.0。为了加强读者对这种表示法的理解，特别区分了它的基本元素和高级元素。

2. 这一版在应用程序的章节中引入了一些新的领域和背景。例如，应用程序的领域范围很广，包括从高级系统架构到基于 Web 的系统的设计细节等各种不同层次的抽象。

3. 在前一版出版时，作为 OO 编程的概念来说，C++ 相对还是比较新的。读者告诉我们，这种强调不再是主要的考虑。现在有大量的 OO 编程和技术书籍及培训，还有许多为 OO 开发而设计的编程语言。因此，大部分关于编码的讨论被删除了。

4. 最后, 响应读者的要求, 这一版更关注 OOAD 建模方面。应用程序章节将展示如何利用 UML, 其中每一章强调了整个开发生命周期中的一个阶段。

本书的目标

本书在面向对象系统构建方面提供了实用指导。它的具体目标如下。

- 提供对对象模型的基础概念及其发展变化的正确理解;
- 帮助读者掌握面向对象分析和设计的表示法和过程;
- 介绍在不同的问题域中面向对象分析和设计的实际应用。

本书介绍的概念都基于牢固的理论基础, 但本书首先是一本注重实效的书, 面向架构师和软件开发者等软件工程实践者的实际需要。

读者对象

本书既是为计算机专业人员也是为学生编写的。

- 对于实际系统和软件的开发者, 本书将展示如何高效地利用面向对象技术来解决实际问题。
- 对于系统分析师或架构师, 本书将利用面向对象的分析与设计, 提供一条从需求到实现的途径。我们帮助分析人员或架构师提高识别能力, 以区分不好的面向对象的结构与好的面向对象的结构, 并在现实情况反常时权衡可选的设计方案。也许最重要的就是, 我们提供了一些让复杂系统变得有条理的新方法。
- 对于项目经理, 本书可以帮助他们更好地理解开发团队的资源分配、软件品质、测量指标以及管理与复杂软件系统相关的风险。
- 对于学生, 本书提供了一些必要的指导, 使得学生能够开始掌握复杂系统开发的科学与艺术中的一些重要技巧。

本书不仅适合专业研讨班和个人学习使用, 也适合作为高等院校本科生和研究生课程的教材。因为它主要阐述了软件开发的方法, 所以非常适合软件工

程和高级编程等课程，也可以作为涉及具体面向对象编程语言的课程的补充阅读材料。

本书的组织结构

本书分成3篇：概念、方法和应用，其中穿插了大量的补充材料。

概念

第1篇研究软件的内在复杂性及其表现方式。本书将对象模型作为一种手段来帮助我们管理这种复杂性，详细地研究了对象模型的基本元素——抽象、封装、模块化、层次结构，讨论了“什么是类？”以及“什么是对象？”等基本问题。由于确定有意义的类和对象是面向对象开发中的关键任务，因此我们花了相当多的时间来研究分类的本质。具体来说，我们研究了生物学、语言学和心理学等其他学科中的分类方法，然后将这些经验应用到发现软件系统中类和对象的问题上。

方法

第2篇基于对象模型提出了复杂系统开发的一种方法。针对面向对象的分析与设计，首先提出了一套图形表示法（即UML），然后是一个通用的过程框架。还研究了面向对象开发的实践，具体来说，就是它在软件开发生命周期中的位置以及它对于项目管理意味着什么。

应用程序

第3篇提供了一组（5个）不简单的例子，涉及不同问题域：系统架构、控制系统、密码分析、数据获取和Web开发。之所以选择这些问题域，是因为它们是软件工程师实践过程中遇到的复杂问题的代表。展示某些原则如何应用于简单的问题是很容易的，但是因为我们关注的是为现实世界构建有用的系统，所以我们对如何将对象模型应用于复杂应用程序更加感兴趣。软件系统的

开发不同于按菜谱做菜，因此我们强调应用程序的增量式开发，这种开发以一些正确的原则和良好的模型作为指导。

补充材料

本书中穿插了大量的补充材料。多数章节中都有补充材料，这些材料对重要的主题提供了相关的信息。本书包括了一个关于面向对象编程语言的附录，其中总结了一些常见语言的特征，还提供了常用术语的词汇表，以及一个扩展的分类参考书目，列出了关于对象模型的参考资料。

工具说明

读者总是会问创建本书中的图使用了什么工具。我们主要使用两个很好的工具来画图：IBM Rational Software Architect 和 Sparx Systems Enterprise Architect。为什么不只用一个？市场的实际情况是，没有哪一种工具可以做所有的事情。实践 OOAD 的时间越长，最后就会发现有些特别的情况是所有工具都不支持的。（在这种情况下，可能需要寻求基本的绘图工具来展示你的想法。）但是，不要让这些很少的情况阻止你使用健壮的 OOAD 工具，如我们提到的这两种工具。

本书的阅读方法

对于本书可以一页一页地读，也可以不按现有的组织形式阅读。如果想对对象模型中的基本概念或面向对象开发的动机有较深的理解，那么就应该从第 1 章开始依次读下去。如果只对面向对象开发分析与设计中的表示法和过程感兴趣，就从第 5 章和第 6 章开始阅读。第 7 章对使用这种方法管理项目的管理者来说特别有用。如果对针对特定问题域的面向对象技术的应用程序更感兴趣，则可以在第 8~12 章中任选一章或者全部阅读。

致 谢

我把本书献给我的妻子 Jan，感谢她的爱和支持。

在第 1 版和第 2 版的写作过程中，一些人促成了我的面向对象开发思想。对于他们的贡献，我特别要感谢，他们是：Sam Adams、Mike Akroid、Glenn Andert、Sid Bailin、Kent Beck、Dave Bernstein、Daniel Bobrow、Dick Bolz、Dave Bulman、Kayvan Carun、Dave Collins、Damian Conway、Steve Cook、Jim Coplien、Brad Cox、Ward Cunningham、Tom DeMarco、Mike Devlin、Richard Gabriel、William Genemaras、Adele Goldberg、Ian Graham、Tony Hoare、Jon Hopkins、Michael Jackson、Ralph Johnson、James Kempf、Norm Kerth、Jordan Kreindler、Doug Lea、Phil Levy、Barbara Liskov、Cliff Longman、James MacFarlane、Masoud Milani、Harlan Mills、Robert Murray、Steve Neis、Gene Ouye、Dave Parnas、Bill Riddel、Mary Beth Rosson、Kenny Rubin、Jim Rumbaugh、Kurt Schmucker、Ed Seidewitz、Dan Shiffman、Dave Stevenson、Bjarne Stroustrup、Dave Thomas、Mike Vilot、Tony Wasserman、Peter Wegner、Iseult White、John Williams、Lloyd Williams、Niklaus Wirth、Mario Wolczko 和 Ed Yourdon。

本书的相当一部分实践来自我参与并在世界各地开发的复杂软件系统，这些系统的开发公司包括 Alcatel、Andersen Consulting、Apple、AT&T、Autotrol、Bell Northern Research、Boeing、Borland、Computer Sciences Corporation、Contel、Ericsson、Ferranti、General Electric、GTE、Holland Signaal、Hughes Aircraft Company、IBM、Lockheed、Martin Marietta、Motorola、NTT、Philips、Rockwell International、Shell Oil、Symantec、Taligent 和 TRW。我曾有机会与数百名专业软件工程师和他们的经理协作，我要谢谢他们的帮助，是他们让本书与真实世界的问题相关。