TURING 图灵程序设计丛书 微软技术系列

Addison Wesley

Framework Design Guidelines

Conventions, Idioms, and Patterns for Reusable .NET Libraries Second Edition

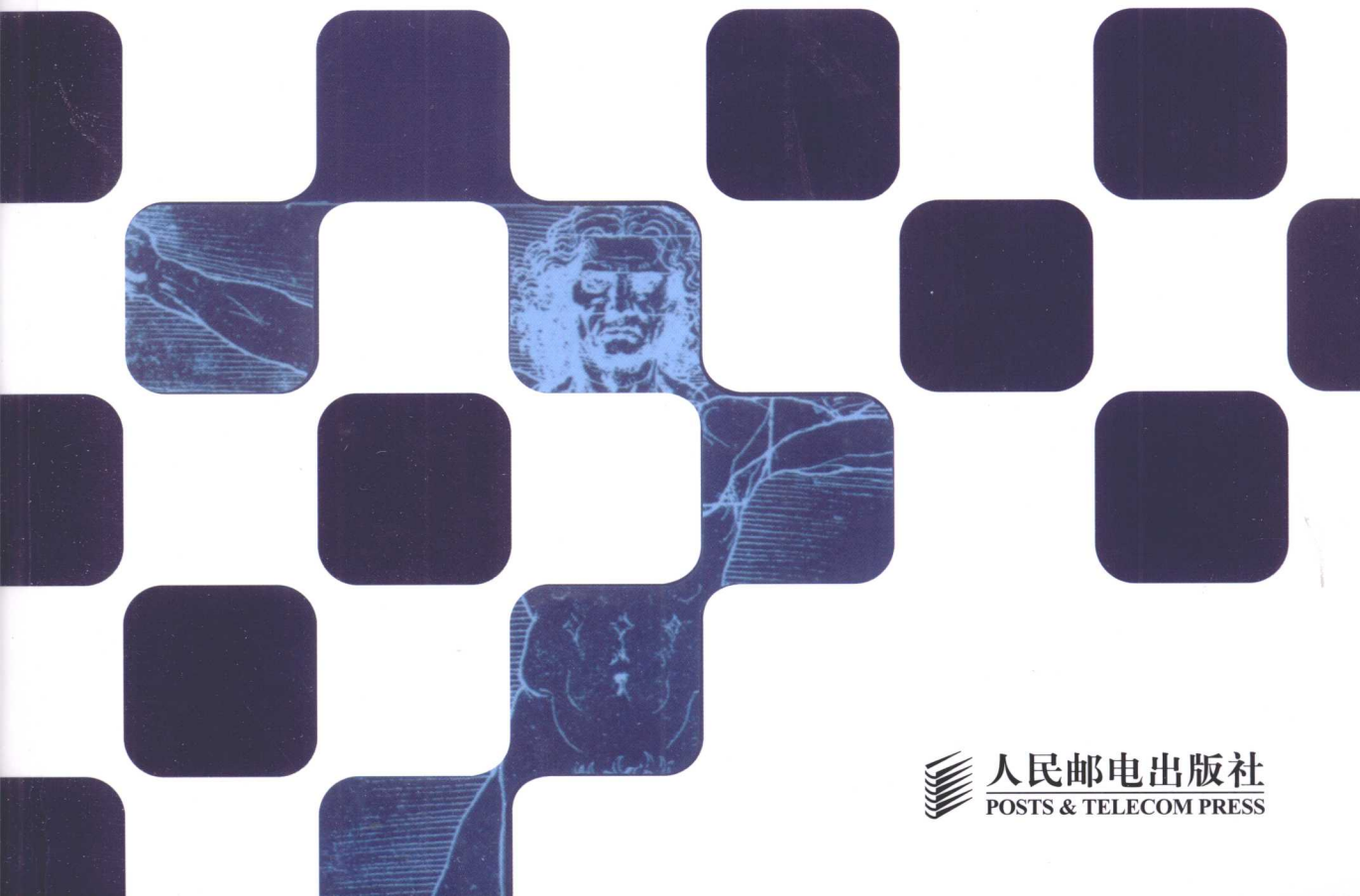
.NET设计规范

约定、惯用法与模式

第2版·英文版

附赠DVD光盘

[美] Krzysztof Cwalina
Brad Abrams 著



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 微软技术系列

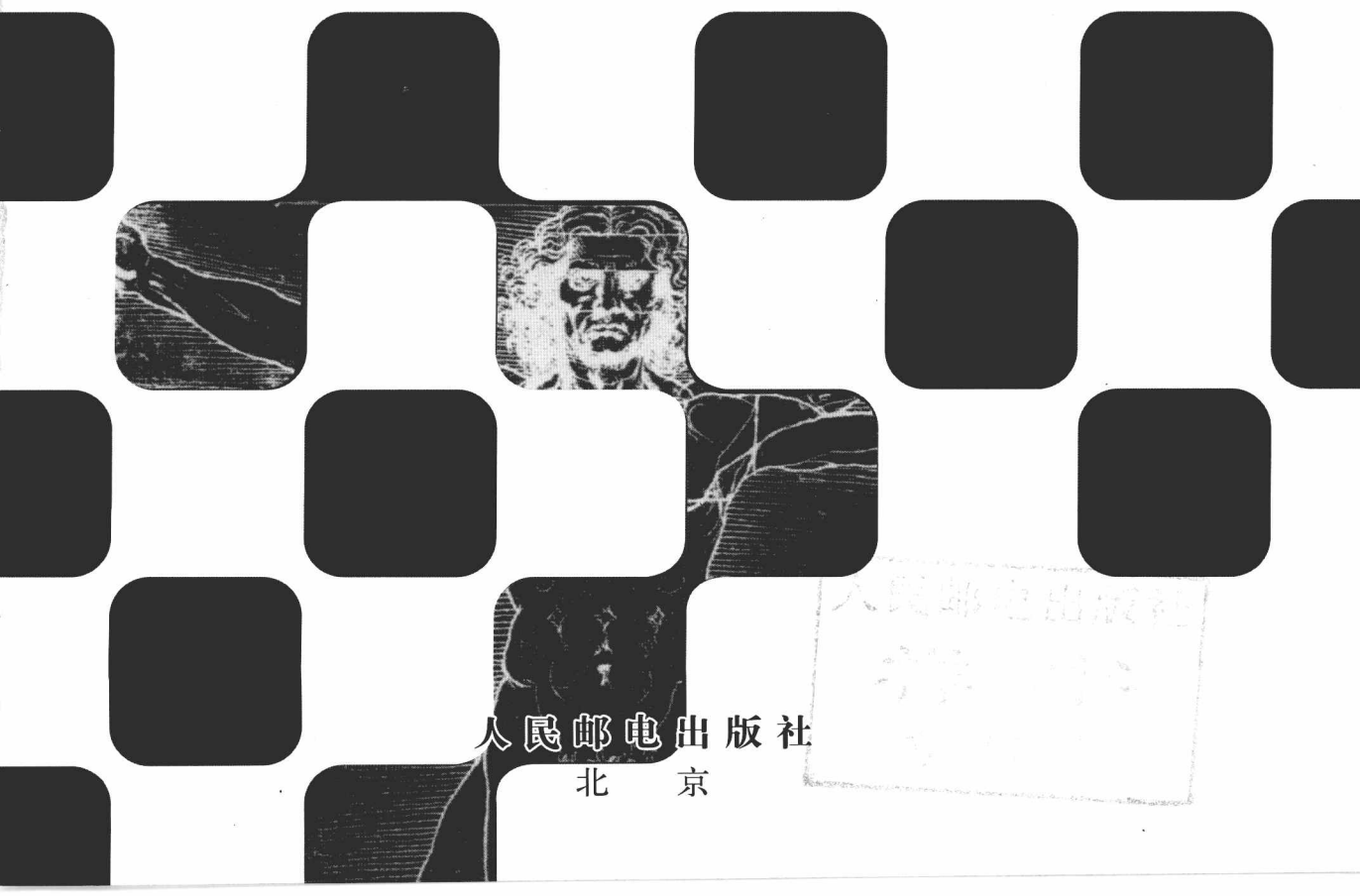
Framework Design Guidelines

Conventions, Idioms, and Patterns for Reusable .NET Libraries Second Edition

.NET设计规范

约定、惯用法与模式 第2版·英文版

[美] Krzysztof Cwalina
Brad Abrams 著



人民邮电出版社
北 京

图书在版编目（CIP）数据

.NET设计规范：约定、惯用法与模式：第2版：英文/（美）克瓦林纳（Cwalina, K.），（美）艾布拉姆斯（Abrams, B.）著. —北京：人民邮电出版社，2010.1
（图灵程序设计丛书）
书名原文：Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries, Second Edition
ISBN 978-7-115-21445-4

Ⅰ.①N… Ⅱ.①克…②艾… Ⅲ.①计算机网络－程序设计 Ⅳ.①TP393.09

中国版本图书馆CIP数据核字（2009）第179584号

内 容 提 要

本书关注直接影响框架可编程能力的设计问题，为框架设计师和广大开发人员设计高质量的软件提供了权威的指南，这一版更新至.NET 3.5。书中内容涉及框架设计的基本原则和规范，常用设计惯用法，为命名空间、类型、成员等框架各部分命名的规范，框架中常用设计模式的规范等。同时，书中添加了来自经验丰富的框架设计师、业界专家及用户给出的评注，为书中的许多规范增色不少。

本书为框架设计师必读之作，也可用作.NET开发人员的技术参考书。

图灵程序设计丛书

.NET设计规范——约定、惯用法与模式（第2版·英文版）

◆ 著 [美] Krzysztof Cwalina Brad Abrams
责任编辑 傅志红

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市潮河印业有限公司印刷

◆ 开本：800×1000 1/16
印张：29
字数：557千字 2010年1月第1版
印数：1–3 000册 2010年1月河北第1次印刷

著作权合同登记号 图字：01-2009-5711号
ISBN 978-7-115-21445-4

定价：59.00元（附光盘）

读者服务热线：(010)51095186 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版 权 声 明

献给我的妻子 Ela，

感谢她在我漫长的写作过程中给予我的支持。

献给我的父母 Jadwiga 和 Janusz，

感谢他们对我的鼓励。

——Krzysztof Cwalina

献给我的妻子 Tamara，

你的爱和耐心使我更加坚强。

——Brad Abrams

对本书的赞誉

“《.NET 设计规范》是少有的几本可以从不同深度阅读、对各类开发者都有益的书。无论你是想设计一种有效的对象模型，还是想加深对 .NET 框架的理解，是想借鉴软件大师的经验，还是想避免常见的编程错误，或者只是想了解一下人们通过怎样巨大的努力才走到 .NET 这条路上来，你都必须要看一看这本书。”

——Francesco Balena，意大利 VB 移植伙伴团队公司（www.vbmigration.com）
代码架构师，技术作家，微软地区经理

“框架极有价值，但却出了名地难以构建，因为你在构建过程中的每一个决策都必须环环相扣，使框架容易被正确使用，而又很难被用错。本书带给你一条条中肯的建议，可以让你不必事后后悔不迭地说‘我早点儿知道这个就好了’。我要是早点儿看到这本书就好了。”

——Paul Besley，QA 公司首席技术官

“自从 Brooks 的《人月神话》以来，已经很久没有大的软件厂商为现代软件开发人员写这样一本充满优秀建议的书了。本书会在我的书架上永久占据一席之地，我会经常去查阅的。”

——George Byrkit, Genomic Solutions 公司高级软件工程师

“本书根据 .NET Framework 3.0 和 3.5 语言的新特性做了全面更新，完全可以继续作为想要设计类库框架的 .NET 开发者和架构师的权威参考书。某些既有规范加了新的注解，更加详细，并且又增加了一些涉及扩展方法和可空类型等特性的新规范。每条规范都能帮助开发者写出清晰易懂的代码，而本版增加的注解内容将帮助你深入了解 .NET Framework 演变过程中的重大设计决策。”

——Scott Dorman, 微软 MVP，坦帕湾区国际软件架构师协会主席

“本书充满了对各层次开发者和架构师极为有用的信息，提供给大家实用的指导和理解这些规则的专家级背景信息。新版更趋完美，你若想写出能很好地集成在 .NET 环境中的应用，就非看此书不可。”

——软件工程师 Cristof Falk

“本书绝对是所有 .NET 开发人员的必读之作。它清楚地告诉你在设计 .NET 类库的时候该做什么，不该做什么。它还揭示了 .NET 设计和创建过程中的一些内幕，使开发人员了解事情的来龙去脉。这些宝贵信息将帮助开发人员写好自己的类库，并能更加有效地利用 .NET 类库。”

——Jeffrey Richter, 技术作家 / 培训师 / 咨询师，Wintellect 公司

“《.NET 设计规范》第 2 版给予大家一些崭新的重要启示，可应用于自己的类库设计。两位作者开诚布公地讨论了面临的挑战：为已发布的产品增加新特性，却要对现有代码影响最小。你可以看到 .NET 类库团队在创建 2.0、3.0 和 3.5 版的时候是怎么做的，他们的优秀范例可以指导你为自己的软件创造出 N+1 版。他们在尽量少影响现存 API 的情况下，增加了泛型、WCF、WPF、WF、LINQ 等新特性，甚至让客户可以只使用部分新特性，而保持对原有库的兼容。”

——Bill Wagner, SRT Solutions 公司创始人及顾问，
Effective C# 和 *More Effective C#* 的作者

“本书是所有从事框架工作的架构师和软件开发人员必读的作品。书中揭示了 .NET Framework 设计时所做进行的一些思考。对于承担应用框架创建任务的人员来说，本书不可不读。”

——Peter Winkler, Balance 技术公司高级软件工程师

序

.NET Framework 一推出，我就立即对它着了迷。这种技术让 CLR（通用语言运行库）及其大量 API，以及 C# 语言的优势立刻彰显无遗。这些技术无不体现了 API 的通用设计风格和始终贯彻的一系列约定。这就是 .NET 文化。一旦你了解了这种文化，就容易把有关知识运用到框架设计的其他领域中去。

过去 16 年里，我一直在从事开源软件工作。由于开源的特点，开发者不仅文化背景不同，连开发时间也会跨越好几年，这时坚守一种风格和编码约定就显得特别重要。虽然我们有维护人员，其日常的工作就是重写或修改提交来的软件，保证代码遵守项目编码标准和风格，但是，如果参与软件项目的所有人一开始就使用项目的约定，当然就更为理想。通过实践和标准所传达出的信息越多，未来新加入的人员就越容易上手。这有利于项目汇聚和融合新老代码。

.NET Framework 在成长，其开发者社区也在成长，人们不断确定了新的实践、模式和约定。Brad 和 Krzysztof 挺身而出，成为了这些规则的“监护人”，他们把这些新知识转变为现在这本规范。他们通常会在博客里介绍新的约定，收集社区的反馈，再记录下这些规范。在我看来，任何人若有兴趣用好 .NET Framework，则必须要常看他们的博客。

这本书第 1 版出版以后，立即成为整个 Mono 社区传诵的经典，其原因大体有两个。首先，我们可以由此了解各种 .NET API 实现的原委和方式；其次，我们珍视其为无价的规范，也努力贯彻在自己的程序和库的设计过程中。第 2 版不仅建立在第 1 版成功的基础之上，而且还添加了不少新的知识。许多规范还增加了注解，注解者本身就是行业里首屈一指的 .NET 架构师和伟大的程序员，正是他们帮助确定了这些规范。

最后我要说，这本书远不止是一本规范。它是我们热爱的一部经典，可以帮助我们成为一名杰出的程序员，而这样的程序员在我们行业里现在还为数不多。

Miguel de Icaza

GNOME 和 Mono 项目创建者

于马萨诸塞州波士顿市

前一版序

在 .NET 框架开发的早期（那时甚至 .NET 框架这个名字还没有诞生呢），我花了无数时间与各开发组一起对框架的设计进行评审，以确保最终得到的平台是易于理解、内在一致的。我始终认为对框架来说，最关键的品质应该是一致性。一旦用户理解了框架的一部分，那么就应该能立即理解框架的其他部分。

可以想象，一大群聪明人在一起肯定会有许多不同意见，我们的开发组正是如此——再没有其他什么事情能比编程约定更能激发出生动而热烈的辩论了。但是，为了保证一致性，我们逐渐化解了各种不同意见，并将结果编纂为一组通用的规范，这样程序员就能容易地理解并使用框架。

Brad Abrams 和 Krzysztof Cwalina 先后帮助我们把这些规范整理到文档中，并不断对其进行更新和完善。本书就是他们的工作成果。

这些规范有效地帮助我们完成了 .NET 框架的三个版本和许多小项目。而且，这些规范还在指导着微软 Windows 操作系统的下一代 API——WinFX 的开发。

希望通过阅读本书，读者也能使自己的框架、类库及组件变得易于理解，易于使用。

希望框架设计能给你带来快乐。祝好运！

Anders Hejlsberg

微软杰出工程师，C# 和 Delphi 之父

2005 年 6 月于美国华盛顿州雷德蒙德市

前　言

本书介绍了设计框架的最佳实践。所谓框架，即可重用面向对象程序库。书中所描述的规范普遍适用于下述规模不同、可重用程度不同的框架。

- 大规模的系统框架。这些框架通常都有成千上万个类型，并且为大量的开发人员所使用，如 .NET Framework。
- 中等规模的程序库。这既可以是大型分布式应用程序的可重用层，也可以是对系统框架的可重用扩展，如 Web 服务扩展（Web Services Enhancements）。
- 小规模组件。为多个应用程序所共享，如网格控件（grid control）库。

值得注意的是，本书关注的是直接影响框架（可以公开访问的 API）可编程能力的设计问题。因此，我们不会过多涉及实现细节。正如一本介绍用户界面设计的书不会讨论有关如何实现碰撞测试（hit testing）的细节一样，本书也不会讲解如何实现二叉排序。这样的定位使我们能够提供针对框架设计师的完整可靠的指南，而不是又一本关于编程方面的书。

书中的规范是在 .NET Framework 的开发早期形成的，它们最早只是少量的命名和设计约定，但是经过不断改进、检验、提炼，这些规范最终成为了微软内部公认的框架设计规范。这些规范历经 .NET Framework 三个版本的长期开发，凝聚了数千名开发人员累积的经验和智慧。我们并不希望本书只是基于一些理想化的设计理念，我们认为本书是极其注重实效的，微软的各开发组将之用于日常开发就是很好的证明。

本书包含许多评注，它们有的解释了相应规范的利弊权衡，有的介绍了其历史，有的给出了进一步的说明，有的提出了自己的批评意见。所有评注都来自经验丰富的框架设计师、业界专家及用户。这些源于开发一线的故事，为书中的许多规范增色不少。

为了使命名空间名、类、接口、方法、属性及类型能够在正文中一目了然，我们用 `Consolas` 字体表示。

本书的读者应该基本知道如何使用 .NET Framework 编程，有一些规范要求读者熟悉 .NET Framework 3.5 版引入的新特性。如果你想找一本介绍 Framework 编程的书，那么可以参考书后的 Suggested Reading List 部分，其中有一些非常好的建议。

规范的表示方法

我们通过要（Do）、考虑（Consider）、避免（Avoid）、不要（Do not）这些词把书中的规范组织成一条条简单的建议。每一条规范都描述了一种好的或是不好的做法，并用统一的方式来表示。好的做法，在其前面会用 ✓ 表示；与此对应，不好的做法则用 × 表示。每一条规范的措词也会明确表示出这条规范的重要性。例如，“要……”描述的是必须[①]遵循的规范（下面所有的例子都摘自本书）：

✓ **DO 要**在命名自定义的 attribute 类时加上“Attribute”后缀。

```
public class ObsoleteAttribute : Attribute { ... }
```

另一方面，“考虑……”描述的是在一般情况下应该遵循的规范，但如果完全理解规范背后的道理，并有很好的理由不遵循它时，也不要畏惧打破常规：

✓ **CONSIDER 考虑**将类型定义为结构，而不要定义为类，如果该类型的实例较小、存活期较短或通常内嵌在其他对象中。

同样，“不要……”描述的是一些绝对不应该违反的规范：

× **DO NOT 不要**把可变类型的实例赋给只读字段。

“避免……”就没有那么绝对，它描述的做法虽然通常并不好，但却存在一些已知的可以违反该规范的情况：

× **AVOID 避免**使用 `ICollection<T>` 或 `ICollection` 作为参数，如果其目的仅仅只是为了访问 `Count` 属性。

对那些更为复杂的规范，我们会另外提供一些背景知识、代码示例及基本原理：

✓ **DO 要**为值类型实现 `IEquatable<T>`。

值类型的 `Object.Equals` 方法会导致装箱，而且由于使用了反射，因此默认实现的效率不高。`IEquatable<T>.Equals` 能够提供更好的性能，而且它的实现可以避免装箱操作。

```
public struct Int32 : IEquatable<Int32> {
    public bool Equals(Int32 other){ ... }
}
```

① “必须”这一措词可能有点太强。虽然有些规范我们的确应该始终遵守，但此类规范极其罕见。另一方面，在一些非常特殊的情况下，你可能需要违反一些“要……”规范，而框架的用户仍能从中受益。

语言选择和样例代码

公共语言运行库（CLR）的目标之一就是支持多种编程语言，这其中既包括微软提供的语言，如 C++、VB、C#、F#、Python 和 Ruby，也包括第三方语言，如 Eiffel、COBOL 和 Fortran 等。因此，本书适用于开发和使用现代框架的多种语言。

为了加强多语言框架设计的概念，我们曾考虑过使用几种不同的编程语言来编写样例代码。但最终我们还是放弃了这种想法，因为使用不同的语言虽然有助于理念的表达，但这样做可能会迫使读者学习好几种新语言，而这并不是本书的目的。

我们决定使用一种易于为广大开发人员阅读的语言，最终选择了 C#，因为 C# 是一种简单的语言，它源自 C 语言家族——一个在框架开发方面有着悠久历史的家族（C 家族中的其他语言包括 C、C++、Java 和 C#）。

语言的选择对于许多程序员来说至关重要，对那些不适应 C# 的读者，我们谨在此表示歉意。

关于本书

本书提供了自顶向下的框架设计规范。

- 第 1 章简要介绍了本书，讨论了框架设计的理念，这是书中唯一没有规范的一章。
- 第 2 章为框架的总体设计提供了基本的原则和规范。
- 第 3 章包含了常用设计惯用法以及为命名空间、类型、成员等框架各部分命名的规范。
- 第 4 章为类型设计提供了规范。
- 第 5 章为类型成员的设计提供了进一步的规范。
- 第 6 章讨论了一些对保证框架的扩展性来说至关重要的问题和规范。
- 第 7 章为与异常有关的工作提供了规范，及推荐使用的错误报告机制。
- 第 8 章为扩展和使用框架中的常用类型提供了规范。
- 第 9 章为框架中常用的设计模式提供了规范和示例。
- 附录 A 对本书使用的编程约定做了简要的描述。
- 附录 B 介绍了 FxCop 工具。可以用该工具来分析框架的二进制文件，以验证框架是否符合本书所描述的规范。本书的附带光盘中包括了此工具。
- 附录 C 是一份 API 规范样例，这些规范通常是微软公司的框架设计人员在设计 API 时创建的。

本书附带光盘还包括了另一份 API 规范样例和其他有用的资源。

致　　谢

本书实质上是对数千人智慧结晶的一个汇总，对所有这些人我们深表谢意。

微软公司的许多人在过去几年中长期努力地工作，是他们提出建议，进行辩论，并最终撰写了书中的许多规范。虽然不可能列出所有的参与者，但在此我们要特别感谢：Chris Anderson、Erik Christensen、Jason Clark、Joe Duffy、Patrick Dussud、Anders Hejlsberg、Jim Miller、Michael Murray、Lance Olson、Eric Gunnerson、Dare Obasanjo、Steve Starck、Kit George、Mike Hillberg、Greg Schecter、Mark Boulter、Asad Jawahar、Justin Van Patten 和 Mircea Trofin。

许多人审阅了本书并提供了评注，我们在此同样表示感谢：Mark Alcazar、Chris Anderson、Christopher Brumme、Pablo Castro、Jason Clark、Steven Clarke、Joe Duffy、Patrick Dussud、Mike Fanning、Kit George、Jan Gray、Brian Grunkemeyer、Eric Gunnerson、Phil Haack、Anders Hejlsberg、David Kean、Rico Mariani、Anthony Moore、Vance Morrison、Christophe Nasarre、Dare Obasanjo、Brian Pepin、Jon Pincus、Jeff Prosise、Brent Rector、Jeffrey Richter、Greg Schechter、Chris Sells、Steve Starck、Herb Sutter、Clemens Szyperski、Mircea Trofin 和 Paul Vick。

上述同仁对本书做了非常必要的补充增益。他们为正文添加了注解内容，补充了有关历史背景知识，让文风变得风趣生动，这些都十分出彩，令本书增色不少。

附录 B 对 FxCop 的介绍实际上是 Sheridan Harrison 和 David Kean 编写的，在此我们对他们表示感谢。

感谢 Martin Heller，他从技术上、精神上给我们以帮助和支持。感谢 Pierre Nallet、George Byrkit、Khristof Falk、Paul Besley、Bill Wagner 和 Peter Winkler，他们提出了中肯而有益的意见。

特别感谢 Susann Ragsdale，是她把诸多不甚连续的想法化为本书。她耐心、极具幽默感，她的创作思路完美无瑕，所有这些使得本书的写作过程变得格外轻松。

Contents