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

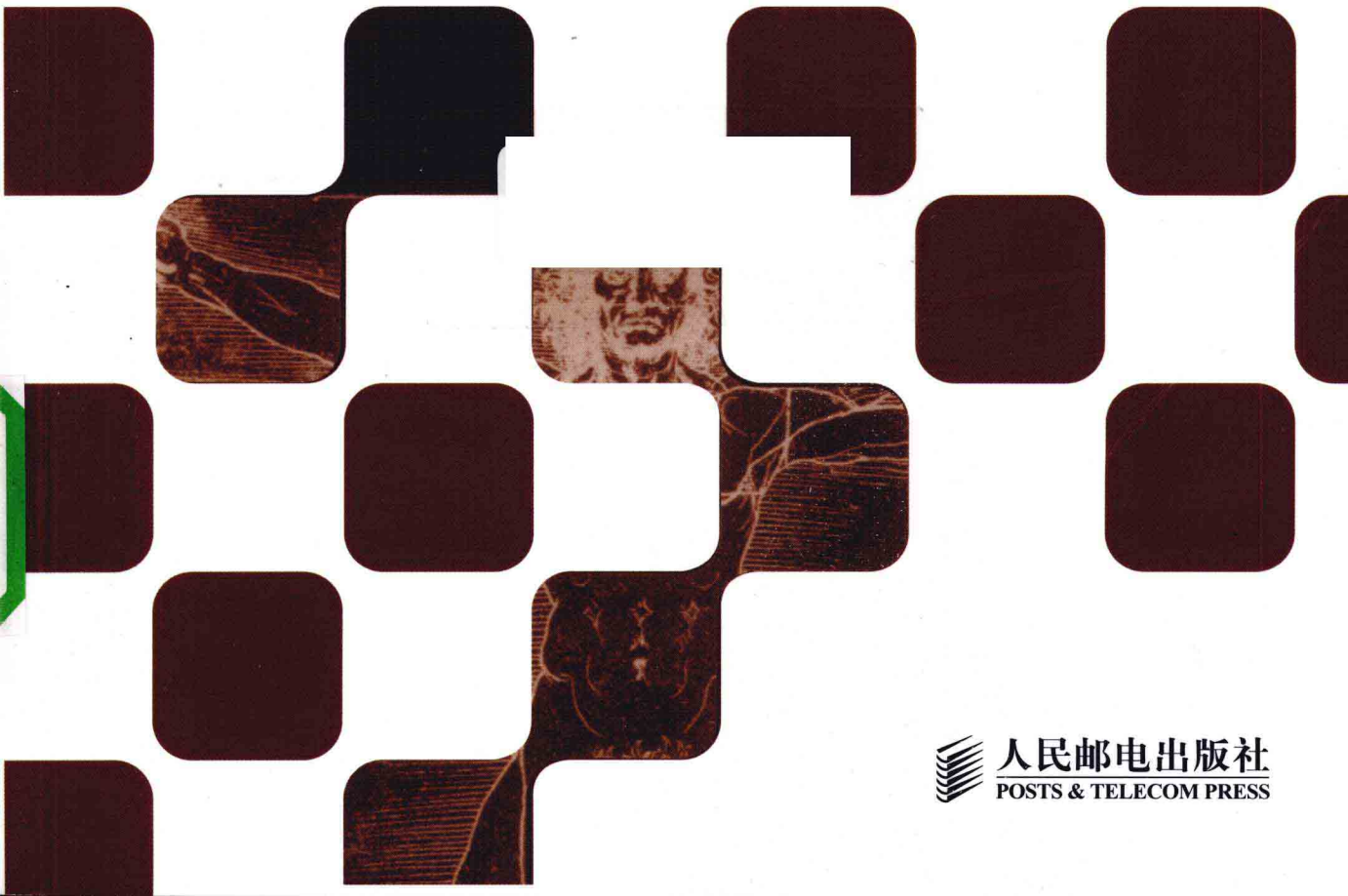
约定、惯用法与模式

第2版

附赠DVD光盘

[美] Krzysztof Cwalina
Brad Abrams 著

葛子昂 译



人民邮电出版社
POSTS & TELECOM PRESS

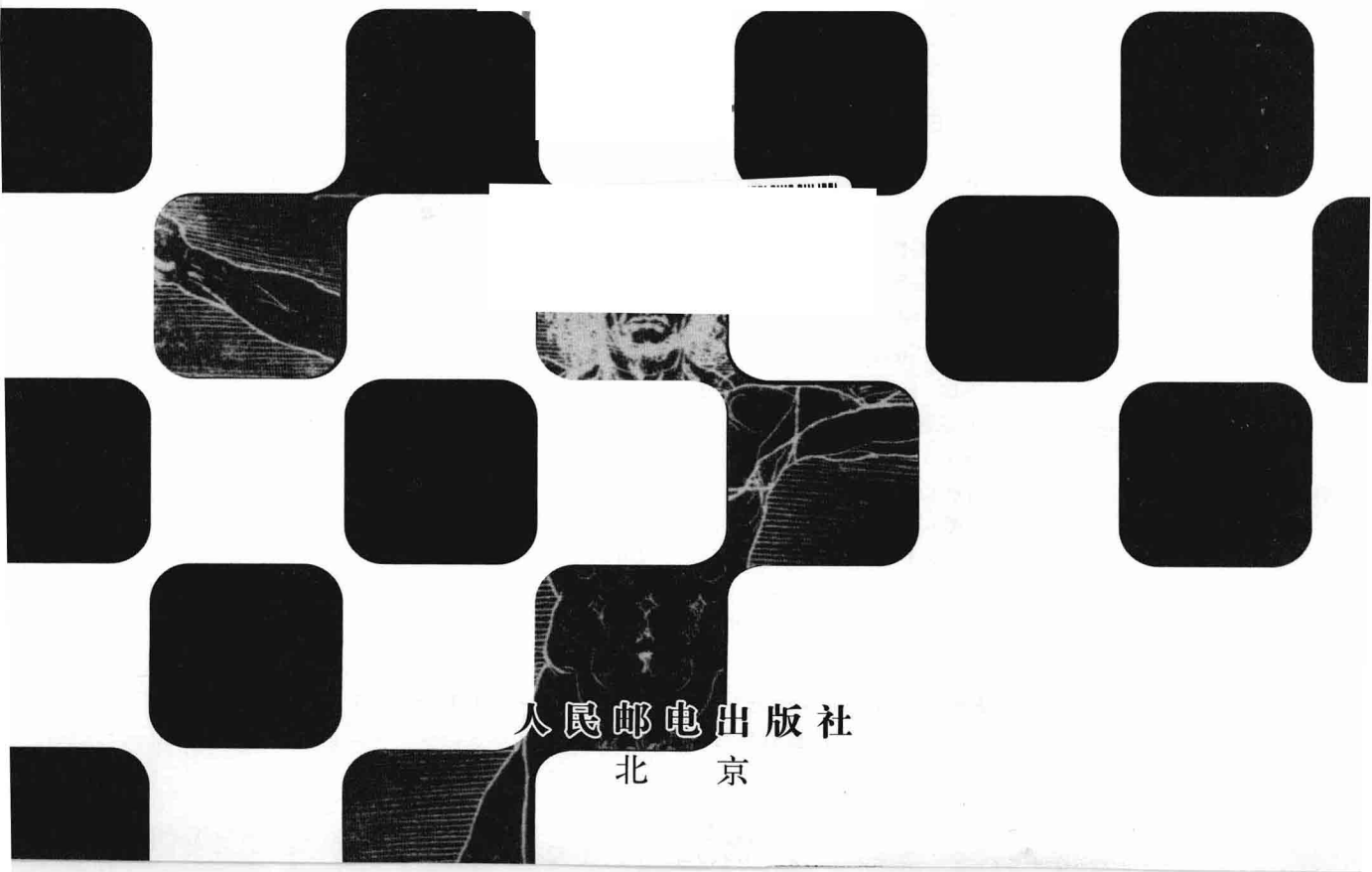
TURING 图灵程序设计丛书 微软技术系列

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



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 / (美) 克瓦林纳 (Cwalina,K.), (美) 艾布拉姆斯 (Abrams,B.) 著 ; 葛子昂译. -- 北京 : 人民邮电出版社, 2010.5
(图灵程序设计丛书)
书名原文: Framework Design Guidelines:Conventions,Idioms,and Patterns for Reusable .NET Libraries,2E
ISBN 978-7-115-22651-8

Ⅰ. ①N… Ⅱ. ①克… ②艾… ③葛… 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(2010)第056722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，为名字空间、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）

◆ 著 [美] Krzysztof Cwalina Brad Abrams
译 葛子昂
责任编辑 傅志红
执行编辑 丁晓昀

◆ 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
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：24.5
字数：579千字 2010年5月第1版
印数：1－3 000册 2010年5月北京第1次印刷

著作权合同登记号 图字：01-2009-5710号

ISBN 978-7-115-22651-8

定价：69.00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

序

.NET 框架一推出，我就立即对它着了迷。这种技术让 CLR（Common Language Runtime，公共语言运行库）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 框架在成长，其开发者社区也在成长，人们不断地确定新的实践、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 框架就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，我们视其为无价的规范，努力将其贯彻在自己的程序和库的设计过程中。第 2 版不仅继承了第 1 版的成功之处，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

墨西哥自由软件开发者，GNOME 和 Mono 项目创建者

于马萨诸塞州波士顿市

第一版序

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码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

第二版译序

从 *Framework Design Guidelines* 的第一版出版到现在，转眼间已经过去了四年。四年前，.NET 框架 2.0 和 Visual Studio 2005 刚刚发布。之后我们先后迎来了 .NET 框架 3.0、.NET 框架 3.5 和 Visual Studio 2008，而现在我们又即将迎来 .NET 框架 4.0 和 Visual Studio 2010。事实上，在我忙于翻译本书的时候，也正是我忙于 Visual Studio 2010 的开发和收尾工作的时候。

本书保留了第一版中的大部分内容，对原有内容进行了适当的更新。这充分证明了书中的设计规范是经得起时间考验的，而这也正是它的核心价值所在。除此之外，本书还增加了许多新的内容，包括序列化、LINQ、依赖属性等等，从而涵盖了 .NET 框架 3.0 和 .NET 框架 3.5 中的核心特性。

在过去的四年中，无论是我对技术的理解还是对语言的把握，都有了相当的提高，这一点将在书中得以体现。第一版的读者可能会发现，中文第二版不仅更新了英文版中更新的那些部分，同时还对第一版已有的内容进行了修订和润色。我相信本书不仅是《.NET 设计规范（第 2 版）》，而且是一本更好的《.NET 设计规范》。

虽然我竭尽所能力求译文的准确和流畅，但鉴于时间和精力有限，难免会有翻译不当甚至是错误之处。为此我建立了一份网上勘误表，如果读者发现任何错误，都请通过该网页与我联系，一旦确认我会立即将其更新到勘误表中。勘误表的网址为：www.gesoftfactory.com/ge/FrameworkDesignGuidelines2/。

我要感谢本书的编辑丁晓昀对我的理解和支持，我们之间的合作非常愉快。我还要感谢我的同事吴宇进、田超、唐友、张羿和王彩霞，他们在繁忙的工作之余审阅译稿，发现了不少难以觉察的问题，并提出了许多宝贵的意见和建议，从而使得本书的质量更上一层楼。最后，我要感谢我的妻儿，他们的支持和鼓励，是我前进的动力。

葛子昂

2010 年 3 月

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 框架。
- 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 Enhancement）。
- 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[①]）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 框架的开发早期形成的，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 框架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，它们有的评注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 `Courier` 字体表示。

---

① 包括一些公共类型，和这些类型的公有、受保护的和显式实现的成员。

本书的读者应该基本知道如何使用 .NET 框架编程，有一些规范要求读者熟悉 .NET 框架 3.5 版引入的新特性。如果你想找一本介绍框架编程的书，那么可以参考书后的推荐读物，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，前提是该类型的实例较小、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，前提是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。

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。

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

语言选择和样例代码

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 语言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

本书提供了自上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。

关于评注者

Mark Alcazar曾经想成为一名著名的运动员。但是，当发现自己没有很好的手眼协调性和体育天赋时，他转向了计算机行业。在过去九年中，Mark一直都在微软公司工作。他从事过Internet Explorer中HTML渲染引擎的开发，他还是Windows Presentation Foundation开发组的一员，从头到尾参与了Windows Presentation Foundation的开发。Mark热衷于让空白区域保持一致，喜欢喝Talking Rain的桃子和油桃口味饮料，还喜欢吃辛辣的食物。他拥有西印度大学的学士学位和宾夕法尼亚大学的硕士学位。

Chris Anderson是微软公司互联系统部门的一名架构师。Chris主要关注那些用来实现下一代应用程序和服务的.NET技术，以及相关的设计和架构。从2002年开始到不久以前，他都一直是WPF开发组的主管架构师。Chris曾经撰写过许多文章和白皮书，他不仅在全球的许多大会（包括Microsoft Professional Developers Conference、Microsoft TechEd、WinDev、DevCon等等）做过演讲，而且还做过许多主题演讲。他在www.simplegeek.com中的博客也广受欢迎。

Christopher Brumme于1997年加入微软公司，那时CLR开发组还正在筹建的过程中。此后，他广泛参与了CLR的设计，还参与了其中执行引擎部分的实现。目前他正在关注托管代码中的并发问题。在加入CLR开发组之前，Chris曾先后在Borland和Oracle公司担任架构师。

Pablo Castro是SQL Server开发组的技术主管。在SQL Server和.NET框架中的许多领域，他都做出了诸多贡献。其中包括SQL-CLR集成、类型系统的扩展性、TDS客户/服务器协议，以及ADO.NET API。Pablo目前正在从事ADO.NET实体框架（Entity Framework）的开发。除此之外，他还领导着ADO.NET数据服务（Data Services）项目的开发，该项目寻求的是一种将数据和Web技术集成起来的方法。在加入微软公司之前，Pablo在许多公司从事过各种软件的开发，其中包括用于信用卡评估和风险分析的分布式推理系统、协作软件，以及群件（groupware）。

Jason Clark是微软公司的一名软件架构师。他在微软公司的软件开发生涯包括三个版本的Windows、三个版本的.NET框架，以及WCF。在2000年他出版了他的第一本关于软件开发的书。他还一直参与各种杂志和其他出版物的工作。他目前负责Visual Studio的数据库版本。除了软件开发之外，Jason最热爱的就是他的妻子和孩子们，他们一家幸福地生活在西雅图地区。

Steven Clarke自从1999年开始就在微软的开发者部门担任用户体验研究员。他主要的兴趣

是观察和理解开发人员在使用API时的体验，创建模型并协助产品组设计出能为用户提供最佳体验的API。

Joe Duffy是微软公司的一名软件开发主管，负责对.NET进行扩展来支持并行计算。他不仅管理一组开发人员并为他的团队定义长远的愿景和策略，而且还编写大量的代码。Joe曾在CLR开发组从事与并行处理有关的开发，之前他是EMC公司的一名软件工程师。空闲的时候，Joe会弹吉他、研究音乐理论，并在www.bluebytesoftware.com写博客。

Patrick Dussud是微软公司的一名Technical Fellow，他担任CLR和.NET框架的架构组中的首席架构师。他负责处理整个公司范围内与.NET框架有关的问题，并帮助各个开发团队以最好的方式使用CLR。他尤其关注如何利用CLR提供的抽象层来优化程序的执行。

Michael Fanning目前是微软公司Expression Web开发组的开发主管。他曾是FxCop开发组的早期成员之一，FxCop原来只是一个供微软内部使用的工具，但最终随Visual Studio 2005一起公开发布。

Kit George是微软公司.NET框架开发组的一名程序经理。他于1995年毕业于新西兰惠灵顿维多利亚大学，拥有心理学、哲学和数学学士学位。在加入微软公司之前，他曾经是一名技术培训师，主要负责与Visual Basic相关的培训。他曾花了两年的时间参与过.NET框架前两个版本的设计和实现。

Jan Gray是微软公司的一名架构师。他现在从事并发编程模型及相关基础设施的开发。他曾经是CLR中负责性能的架构师，在上世纪90年代他协助编写了早期的MS C++编译器（例如：语法、运行时对象模型、预处理头文件、PDB文件、增量编译，以及链接）和Microsoft Transaction Server。Jan的兴趣包括用FPGA来搭建自己的多处理器。

Brian Grunkemeyer自1998年起一直是.NET框架开发组的一名软件设计工程师。他实现了框架类库中的许多部分，他还参与了ECMA/ISO CLI标准中许多类的细节的制定。Brian目前正在从事下一个版本的.NET框架的开发，涉及的领域包括泛型、托管代码的可靠性、版本管理、代码中的契约，以及提高开发人员的体验。他拥有卡耐基梅隆大学计算机科学和认知科学双学士学位。

Eric Gunnerson于1994年加入微软公司，此前他曾经在航空业和其他一些不景气的行业工作过。他曾经在C++编译器开发组工作，也曾经是C#语言设计组的一员，并在DevDiv社区工作开展之初成为早期的追随者之一。在Vista开发期间，他参与了Windows DVD Maker界面的设计和实现，并于2007年初加入Microsoft Health Vault开发组。空闲的时候，他会骑自行车、滑雪、把肋骨摔断、制造船舱板、写博客，以及用第三人称来描写自己。

Phil Haack在ASP.NET开发组中从事ASP.NET MVC框架的程序经理，该框架是由社区驱动，以一种透明的方式开发的。该框架的目标是要体现并鼓励一些好的软件设计原则：关注点

分离、可测试性，以及单一职责原则等等。Phil还热衷于编写代码，他不仅编写软件，而且在他的博客中谈论软件开发。

Anders Hejlsberg是微软开发者部门的一名Technical Fellow。他是C#编程语言的主设计师，也是.NET框架开发的一名重要参与者。Anders于1996年加入微软公司，之前他是Borland International的Principal Engineer。作为Borland公司的第一名雇员，他是Turbo Pascal的原作者，也是后来的Delphi产品线的首席架构师。Anders曾在丹麦科技大学学习工程学。

David Kean是微软公司.NET框架开发组的一名开发人员，他从事托管扩展框架（Managed Extensibility Framework，MEF）的开发，该框架是一些组件，用来开发可扩展的动态应用程序。之前他曾经参与开发备受喜爱但也经常被误解的工具FxCop，以及它的兄弟Visual Studio Code Analysis。他毕业于澳大利亚墨尔本的迪肯大学并拥有学士学位。现在他和他的妻子Lucy和两个孩子Jack和Sarah一起居住在西雅图。

Rico Mariani早在1988年就开始了他在微软公司的职业生涯，那时他从事编程语言产品的开发。一开始是Microsoft C 6.0版，一直到Microsoft Visual C++ 5.0版发布。1995年，Rico成为“Sidewalk”项目的软件开发经理，从此他开始了长达七年的平台开发工作，涉及各种MSN技术。2002年夏天，Rico回到开发者部门，担任CLR开发组负责性能的架构师。由于在CLR性能方面的出色工作，他最近被任命为Visual Studio的首席架构师。Rico的兴趣包括编译器和编程语言理论、数据库、三维艺术，以及科幻小说。

Anthony Moore是互联系统部门的一名开发主管。在2001到2007年之间，他是CLR基础类库的一名开发主管，参与过从1.0版到3.5版的框架的开发。Anthony于1999年加入微软公司，一开始从事Visual Basic和ASP.NET的开发。之前他在自己的家乡澳大利亚从事了八年的企业开发，其中包括三年在快餐行业工作。

Vance Morrison是微软公司的一名架构师，负责.NET运行时的性能。他关注运行时性能的各个方面，目前的注意力主要集中在启动时间。自从.NET运行时开始开发，他参与了许多组件的设计。之前他负责过.NET中间语言（Intermediate Language，IL）的设计，还曾经是JIT编译器的开发主管。

Christophe Nasarre是Business Object的一名架构师和开发主管，该公司是SAP旗下的一家跨国公司，主要从事商业智能解决方案的开发。在空闲的时间，Christophe为MSDN Magazine、MSDN，以及ASPToday撰写文章。自从1996年开始，他还担任过许多书籍的技术编辑，涵盖的主题包括Win32、COM、MFC、.NET，以及WPF。他于2007年出版了自己的第一本书，Windows via C/C++，该书由微软出版社出版。

Dare Obasanjo是微软公司MSN通讯服务平台开发组的一名程序经理。他热衷于用XML来解决问题，并以此构建服务器基础设施，供MSN Messenger、MSN Hotmail和MSN Spaces开发

组使用。之前他是XML开发组的一名程序经理，负责.NET框架中核心的XML API以及与W3C XML Schema相关的技术。

Brian Pepin是微软公司的一名软件架构师，目前正在从事Visual Studio中的WPF和Silverlight设计器的开发。他从事开发工具和框架的开发已经有14年了，参与设计的产品包括Visual Basic 5、Visual J++、.NET框架、WPF、Silverlight，还有一个以上从未上市的试验性产品。

Jonathan Pincus是微软研究院系统和网络组的一名资深研究员，他主要的研究方向是软件和基于软件的系统中的安全性、隐私性和可靠性。之前他曾是Instrinsa的创办人和CTO，此外他还在GE Calma和EDA Systems从事过设计自动化（用于IC和CAD框架的布局和走线）的工作。

Jeff Prosise是Wintellect（www.wintellect.com）的创办人之一。他最近的一本书Programming Microsoft .NET由微软出版社于2002年出版，他的文章经常见诸于MSDN Magazine和其他开发人员杂志。Jeff的技术专长主要是ASP.NET、ASP.NET AJAX和Silverlight。毕业后他发现生活不仅仅是计算支撑架的负载，自此他成为了一个更好的工程师。Jeff出了名的是，他会不嫌麻烦到世界上最好的潜水点去潜水，还会花许多时间来造遥控飞机和玩遥控飞机。

Brent Rector是微软公司一名从事技术策略孵化的程序经理。他在软件开发行业有三十多年的经验，他开发过的产品包括编程语言编译器、操作系统、ISV应用程序，以及其他许多产品。Brent自己写过也与别人合写过许多Windows软件开发方面的书，其中包括ATL Internals、Win32 Programming（这两本都由Addison-Wesley出版社出版），以及Introducing WinFX（由微软出版社出版）。在加入微软公司之前，Brent是Wise Owl Consulting, Inc.的创办人和总裁，还是.NET obfuscator和Demeanor for .NET这两个软件的首席架构师。

Jeffrey Richter是Wintellect（www.wintellect.com）的创办人之一。该公司是一家培训、调试和咨询公司，致力于帮助客户以更快的速度开发出更好的软件。他是许多.NET和Win32畅销编程书的作者，其中包括Applied Microsoft .NET Framework Programming（由微软出版社出版）。Jeffrey还是MSDN Magazine的特约编辑，并为其撰写“Concurrent Affairs”专栏。自从1999年起Jeffrey就担任微软.NET框架开发组的顾问，同时他还是微软Web Services和Messaging开发组的顾问。

Greg Schechter从事API的设计和实现已经超过20年，主要在二维和三维图形领域，但也包括媒体、图像、通用用户界面系统，以及异步编程等领域。Greg目前是微软Windows Presentation Foundation和Silverlight开发组的一名架构师。在1994年加入微软公司之前，Greg在Sun Microsystems工作了六年。除此之外，Greg还喜欢以第三人称来描写自己。

Chris Sells是微软公司互联系统部门的一名程序经理。他写了许多书，包括Programming WPF、Windows Forms 2.0 Programming和ATL Internals。在闲暇之余，Chris会主办各种会议，并参与微软产品开发组的各个内部讨论组中。

Steve Starck是微软公司ADO.NET开发组的一名技术主管。在过去的十年中，他一直都在开发和设计数据访问技术，包括ODBC、OLE DB和ADO.NET。

Herb Sutter是软件开发方面的权威人士。在Herb的职业生涯中，他曾经是多种商业技术的创建者和主要设计者，其中包括用于异构分布式数据库的PeerDirect点对点复制系统、用来让C++支持.NET编程的C++/CLI语言扩展，以及最近的Concur并发编程模型。目前他是微软公司的一名架构师，他还担任ISO C++标准委员会的主席，出版过四本备受称赞的书籍，发表过上百篇关于软件开发的技术论文和文章。

Clemens Szyperski于1999年作为软件架构师加入微软公司。他主要关注如何利用组件软件来高效地构建新型的软件。Clemens是Oberon Microsystems及其衍生公司Esmertec的创办人之一，他还是澳大利亚昆士兰科技大学计算机学院的一名助理教授。他是卓越图书大奖得主Component Software（由Addison-Wesley出版社出版）一书的作者，他也是Software Ecosystem（由MIT出版社出版）一书的合著者。他拥有瑞士联邦理工学院的计算机科学博士学位和亚琛工业大学的电子工程/计算机工程硕士学位。

Mircea Trofin是微软公司.NET应用框架核心组的一名程序经理。他主要负责保证和改进.NET框架的架构。他还负责.NET基于组件编程领域中一些即将出现的新特性。他拥有滑铁卢大学的计算机工程学学士学位和都柏林大学的计算机科学博士学位。

Paul Vick是Visual Basic的语言架构师，并领导整个语言设计组。Paul在微软的职业生涯从1992年加入Microsoft Access开发组开始，并参与开发了从Access 1.0一直到Access 97的所有版本。1998年他转到Visual Basic开发组，参与Visual Basic编译器的设计和实现，并为了Visual Basic支持.NET框架而推进语言的重新设计。他是Visual Basic .NET Language Specification和The Visual Basic .NET Language（由Addison-Wesley出版社出版）的作者。可以在www.panopticoncentral.net访问他的博客。

目　录