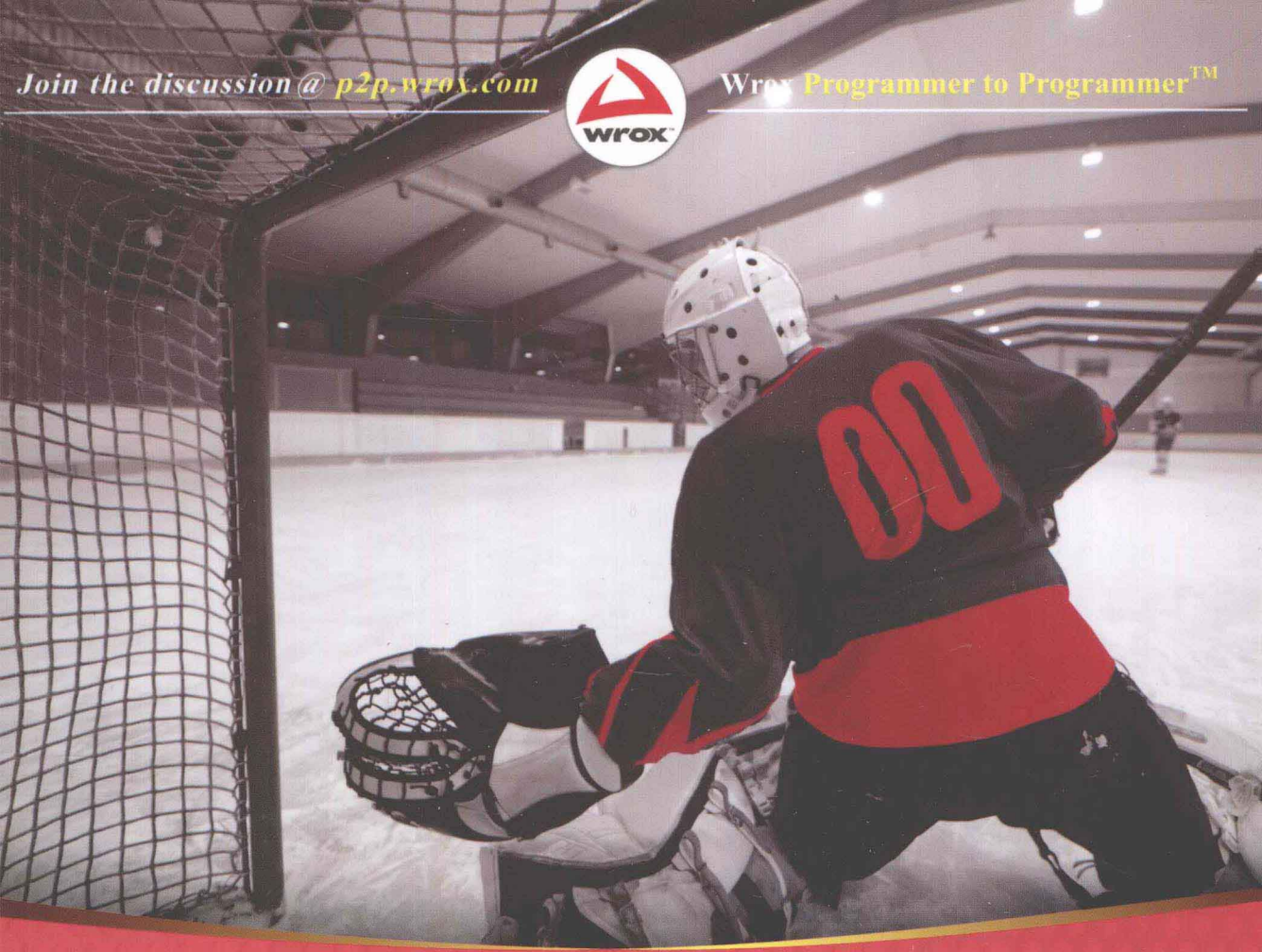

Professional Test Driven Development with C#

C#测试驱动开发

[美] James Bender 著
Jeff McWherter
贾洪峰 李菊彦 译

清华大学出版社

C# 测试驱动开发

(美) James Bender
Jeff McWherter 著

贾洪峰 李菊彦 译

清华大学出版社

北 京

James Bender，Jeff McWherter
Professional Test Driven Development with C#
EISBN：978-0-470-64320-4

北京市版权局著作权合同登记号 图字：01-2011-3827

图书在版编目(CIP)数据

C#测试驱动开发 /(美) 本德(Bender, J.)，(美) 麦格瓦特(McWherter, J.) 著；贾洪峰，李菊彦 译.
—北京：清华大学出版社，2012.3
书名原文：Professional Test Driven Development with C#

ISBN 978-7-302-27971-6

Ⅰ. C…　Ⅱ. ①本…　②麦…　③贾…　④李…　Ⅲ. C 语言—程序设计　Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2012)第 028902 号

责任编辑：王　军　于　平
装帧设计：牛艳敏
责任校对：成凤进
责任印制：王静怡

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
　　社 总 机：010-62770175　　邮　　购：010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：20　　**字　数**：487 千字
版　　次：2012 年 3 月第 1 版　　**印　次**：2012 年 3 月第 1 次印刷
印　　数：1～3000
定　　价：58.00 元

产品编号：041243-01

作 者 简 介

James Bender 是 Improving Enterprises 公司的技术副总裁，从事软件开发和基础体系结构设计已有 17 年。作为开发人员和体系结构设计师，他参与过各种各样的软件设计，既有小型的单用户应用程序，也有企业规模的多用户系统。他是一位经验丰富的顾问和作家，擅长于.NET 开发与体系结构设计、SOA、WCF、WF、云计算和敏捷开发方法。

James 致力于推动软件开发的封装，追求新的、更好的应用程序开发方式。在他的职业生涯之初，是用 C++在基于 SCO Unix 的系统上开发信用卡处理应用程序。在 20 世纪 90 年代后期，James 开始研究用基于 Java 的 JSP 页面和微软的 ASP 技术进行 Web 开发。他从.NET 首次发布公共测试版就开始使用此工具。后来他继续研究.NET 技术，重点是分布式计算范例(.NET Web 服务使之成为可能)，之后很自然地发展到对微软的 Windows Communication Foundation(WCF)极为着迷。

从 2003 年开始，James 就开始实际应用敏捷方法，包括 Scrum 和极限编程(eXtreme Programming，XP)。在研究敏捷方法的同时，James 开始探索测试驱动开发。他致力于向客户及软件开发社区的众多开发人员介绍在敏捷软件开发和测试驱动开发中用到的概念和技术。

James 是 Visual C#方面的微软 MVP、开发社区的活跃会员和 Central Ohio .NET Developers Group(www.condg.org)的现任主席，并继续领导着 Columbus Architects Group(www.colarc.org)，还是 nplus1.org 第一方(first-party)内容的高级编辑，nplus1.org 是一个面向体系结构设计师的教育网站。James 的博客地址为 www.jamescbender.com。

Jeff McWherter 是 Gravity Works Design and Development 的合伙人和开发经理，这家公司的办公室位于密歇根州兰辛市的老城区(Old Town District)。Jeff 毕业于密歇根州大学，拥有 12 年以上的专业软件开发经验，他还在微软获得了很多证书，包括微软认证解决方案开发员(MCSD)、微软认证数据库管理员(MCDBA)、微软认证应用程序开发员(MCAD)和微软认证技术工程师(MCTS)。

2010 年，Jeff 连续第三年被评为“微软最有价值专家(MVP)”。同年，他荣获了由兰辛区域商会颁发的“Ten Over The Next Ten”奖项，这一奖项评选出会在接下来的 10 年间担负起重任的 10 位年轻专家。Jeff 还是一位作家，Wrox 出版社出版了他的 *Testing ASP.NET*

Web Applications 一书。

作为一名作家和软件开发人员，Jeff 还积极参与国内程序设计社区的开发，例如在会议上发言、组织 Lansing Give Camp 等活动，这些活动将开发人员和一些非盈利组织联合起来，共同参加一些志愿者项目。

技术编辑简介

Mitchel Sellers 擅长使用微软技术进行软件开发。他是 IowaComputerGurus 有限公司的 CEO、微软 C# MVP 和微软认证专家，出版了两本书，为很多书籍做技术编辑。经常会看到 Mitchel 与非常大型的软件开发社区进行互动，既包括活动/会议，也包括网上讨论论坛。如需有关 Mitchel 的专业经验、认证和出版作品方面的更多信息，请参阅他在 MitchelSellers.com 上的简历。

致　谢

我首先要感谢我的女朋友 Gayle (如果幸运的话，等您看到这本书时她已经是我的未婚妻了)。在编写本书期间，她非常支持和理解我，她的付出远远超出了应当付出的。

我要感谢我的父母，因为有他们才有了我，也才有了这本书。我的妈妈因为马上可以看到这本书而感到非常自豪。保佑她的心脏吧！我希望当她发现我在撒谎，这本书一点也不像 Stephen King 的小说时，还能如此自豪。

在这个充满理解和支持的部门里，我要感谢 Daniel Grey、Mark Kovacevich、Jeff Perry 和 Improving 公司的每一个成员。我还要感谢 Pete Klassen。我们想念你，兄弟！

我要感谢 Jeff McWherter 和 Michael Eaton 为本书所做的贡献。Jeff，感谢你为我分担一些工作；Mike，感谢你促使我将“非 Web”人员也包含在内。我还要感谢我的编辑 Sydney，是她让这本书看起来好像我具有丰富的写作经验。

Brian Prince，感谢你促使我加入开发社区。我本来想在这儿写点有趣的事情，但却都想不起来了。

当我有机会编写这本书时，我差点拒绝了。我要感谢 Ted Neward 说服我最终写了这本书。

我还要感谢我在 NPlus1.org Mike Wood 和 Chris Woodruff 的合作伙伴，在过去的几个月里，当我对于撰写本书有所懈怠时，他们会督促我。

要感谢的人还有很多：Brahma Ghosh、Brian Sherwin、Bill Sempf、Jeff Blankenburg、Carey Payette、Caleb Jenkins、Jennifer Marsman、Sarah & Kevin Dutkiewicz、Steve Harman、Josh Holmes。感谢 Matt Groves，他对本书的付出几乎和我一样多。我肯定漏了某个人，在这里表示歉意！

James

首先感谢我富有耐心的妻子 Carla。感谢你在我努力工作期间给予我的所有支持、耐心和理解。感谢我在 Gravity Works 的同事——Amelia Marschall、Lauren Colton、Scott Gowell 和 Dave Smith，感谢他们回答我随时碰到的各种问题。最后，还要感谢 James，感谢他的努力工作、奉献和友谊。

Jeff

前　言

作为一名咨询师，我与开发人员合作过。在每一位客户那里，我都会遇到一个新团队，要了解他们是如何开发软件的。我见过非常优秀的团队，也看到过分崩离析以致从未有过一个成功项目的团队。这几年，我注意到不同的团队获得成功具有不同的特点。我开始总结，是什么使一个开发团队开发和部署的应用程序优质且能为企业提供价值。

大多数人都预计我的观察会得出这样一条结论：成功的团队拥有更聪明、更有能力的人员，他们当然能够成功了。但那些失败的团队中也有许多非常聪明的人。显然，智力不是成功的关键因素。

我在成功团队中所看到的，是他们对技术的热情，对自己作品的自豪感。他们总在学习新工具和新技术，希望能够提高软件开发速度，减少错误。而那些不太成功的团队则满足于坚持旧的工作方式，对周边发生的变化从来不感兴趣。

在我第一次遇到这些成功的、富有激情的开发团队时，他们并非都采用了测试驱动开发(test-driven development，TDD)。但在了解到这一概念之后，他们大多都会快速而迫切地锁定它。这些团队发现，在开发软件的过程中，增加测试驱动开发实践，可以马上得到非常出色的结果，提高了所交付应用程序的质量，减少了其中的缺陷。

要培养激情很难，扼杀它却很容易。在缺乏激情的团队里，通过介绍测试驱动开发，在许多情况下都能重新点起开发人员心中的激情。特别是对那些已经厌倦了日复一日地重复相同开发工作的开发人员而言，尤其如此！

除了激情之外，研究测试驱动开发还有另外一个非常有说服力的理由。有证据表明，近年来的两个最大变化可能会吸引极多的开发人员，这两大变化就是敏捷方法的兴起和测试驱动开发。这两者经常一起发展。我不相信，敏捷方法能够在不使用测试驱动开发的情况下获得长期成功，也想象不出测试驱动开发如何在一个瀑布环境中发挥作用。

敏捷方法已经比较成熟。它不再是一些小型开发部门使用的“疯狂牛仔式编码”方法了。一些大型公司在设计其 IT 部门的结构时大规模采用瀑布形式，现在也开始采用一种敏捷方法来开发越来越多的项目。即使是最官僚化的组织、政府也开始研究敏捷方法，并取得了巨大成功。这些开发清楚地表明了一个现实：那些可以在敏捷环境中工作(包括采用测

试驱动开发)的开发人员，其价值很快就能超过那些不能在此环境中工作的开发人员。

驱动测试开发并不是存在于真空中的。在过去几年里，许多团体和运动都旨在提高所开发软件的质量，并使企业参与到这一过程中。新的工作理念与方式已经向前发展，可以帮助开发人员开发出能够满足企业需求的可维护应用程序。诸如“软件工艺”和 SOLID 等术语已经进入了世界各地富有激情的开发人员的词汇表当中。一些开发人员甚至更进一步，将自己称为软件工匠或软件技工。

为了满足人们对于学习测试驱动开发及相关知识的要求，已经有了许多书籍、网站和研讨会，其中有很多是非常出色的。但也有一些只不过是将一些常见的、可移植的工作方式加以商业化运作，为其套上一层昂贵的、专用解决方案的外衣。许多聪明的、热情的开发人员都在讨论和传播测试驱动开发。但是，没有任何一种“一站式”资源能够将一位开发人员(具体来说，是.NET 开发人员)从一个新手变为一个……嗯，还是一个新手，不过是掌握了一些信息的新手。

您正在阅读本书，就凭这一点，就说明您对测试驱动开发是有兴趣的。您可能是一位开发人员，听了关于测试驱动开发的信息，但却从未真正有机会研究它。您也可能是一位很有经验的测试驱动开发人员，只是想看看这本书在讨论这一主题时与其他书有什么不同。无论是哪一种情况，就凭您正在阅读本书，就表明测试驱动开发已经成为主流，值得花时间来学习、练习和改进。

本书读者对象

测试驱动开发是一种非常有效的方式，从项目伊始就能保证应用程序的质量。测试驱动开发的相关原理与实务使您和您的团队能够快速编写出便于维护的软件，更好地满足企业的需要。如果您是一位希望提高自己技能的开发人员，那么本书就是为您准备的。

如果您刚刚接触测试驱动开发，那就从第 1 章开始学习。这样会使您了解足够的背景知识，理解测试驱动开发为什么会如此引人入胜。第 1 章还会介绍一些有关面向对象编程、SOLID 原则和重构的概念。为了实际运用测试驱动开发，这些技术都是至关重要的基础。

如果曾经研究过测试驱动开发，可以从第 3 章开始阅读，该章回顾了面向对象开发、SOLID 原则和重构。即使是非常有经验的开发人员，有时也需要复习一下这些概念与应用程序开发有什么关系。本书的其他部分(从第 4 章开始)为这些开发人员提供了测试驱动开发的形式和结构。

测试驱动开发经验非常丰富的开发人员可能希望从第III部分开始阅读。如果从这里开始，您应当已经拥有了很高程度的测试驱动开发、面向对象编程(OOP)和 SOLID 技术。这一部分主要讨论.NET 开发人员所面对的具体情景。其中包括如何在以下应用程序中采用测试驱动开发：基于 Web 的应用程序(包括 Web 窗体、ASP.NET MVC 和 JavaScript)、用“模型-视图-视图模型(MVVM)”模式在 WPF(Windows Presentation Foundation)上构建的应用程

序、用微软的 WCF(Windows Communication Foundation)构建的服务应用程序。一个应用程序中最难测试的部分就是边缘。这些章节将说明如何尽可能缩小应用程序的边缘，从而使其更容易测试。

本书内容

要使测试驱动开发在软件行业中得以繁荣兴盛，需要一些条件，本书从讨论这些条件开始。软件开发发展到今天，有其历史和特定的条件，理解这些很重要。避免重复过去的错误也很重要。在自己当前的开发实践中找出这些反面模式则更为重要。

为了支持测试驱动开发的应用，本书还全面介绍了面向对象编程、敏捷方法以及 SOLID 软件设计和编码原理。

当然，本书还介绍了测试驱动开发中一些固有的、必需的概念。首先介绍的一些测试都是很简单的，很容易理解。您会看到如何在 Visual Studio 中使用 NUnit 单元测试框架编写单元测试。

之后将介绍依赖注入模式。其中包括如何实现这一模式，以及依赖项注入框架(如 Ninject)如何帮助管理应用程序中的依赖。本书还会介绍“模拟”(mocking)与“模拟框架”(mocking framework)实践，包括对模拟框架 Moq 的介绍。

关于行为驱动开发的基础知识，也在介绍范围之内，但不会深入讨论这一主题。本书解释了行为驱动开发背后的思想，并突出介绍了命名测试的业务驱动开发风格。本书还介绍了 NBehave 测试框架。NBehave 有许多功能，但本书只用它为测试提供“句法糖”。

本书的组织结构

设计本书的内容结构花费了很大的力气，希望每一章都是以上一章的课程为基础。前几章主要提供一个基础，说明测试驱动开发的重要性以及有效使用它所需要的基础技巧。每个章节都以一个概念为基础，如依赖项注入(dependency injection)和模拟，直到使您掌握实际运用测试驱动开发所需要的全部工具和技巧为止。

第 III 部分结合前几章讲授的测试驱动开发技巧，阐述了如何利用微软的几个框架来练习测试驱动开发，这几个框架是用来为应用程序开发界面的，包括 ASP.NET MVC、WPF 和 WCF。

本书的最后是附录，列出了一些备用工具，有助于使用测试驱动开发来开发应用程序。它还列出了一些可能出现的用户情景，如果在日常工作中还没有使用测试驱动开发，可以把这些用户情景当作练习。

完成本书练习的必备条件

要完成本书给出的示例，并使用可下载的演示应用程序，需要以下工具：

- Visual Studio 2010(任意版本)
- NUnit version 2.5.2.9222 或更新版本，可从 nunit.org 获取
- Moq version 4 beta 4(build 4.0.10827.0)或更新版本，可从 code.google.com/p/moq 获取
- Ninject version 2(build 2.1.0.91)或更新版本，可从 ninject.org 获取
- NBehave version 0.4.5.183 或更新版本，可从 nbehave.org 获取
- Fluent NHibernate version 1.1 或更新版本，可从 fluentnhibernate.org 获取
- 数据库管理系统(Database Management System，DBMS)供示例应用程序使用。本书中的示例使用了 Microsoft SQL Server Developer，但任何关系数据库系统都是可以的。

源代码

使用本书的示例时，既可以采用人工方式输入所有代码，也可以使用随本书提供的源代码文件。本书中用到的所有源代码都可以从 www.wrox.com 和 http://tupwk.com.cn/downpage 下载。在该网站上，只要找到本书的标题(使用搜索框或标题列表)，并单击本书详细信息页中的 Download Code 链接，就可以获得本书的全部源代码。该网站提供的代码在本书中用以下图标突出显示：

列表中包含了标题中的文件名。如果只是一个代码段，可以在代码提示中找到其文件名，如下所示：

代码段文件名

提示：

由于许多书的书名类似，所以用 ISBN 进行搜索是最容易的；本书的 ISBN 为 978-0-470- 64320–4。

下载代码之后，用最喜欢的压缩工具解压即可。或者，可以进入 Wrox 的代码下载主页 www.wrox.com/dynamic/books/download.aspx，查看本书及所有其他 Wrox 书籍的可用代码。

勘误表

尽管我们已经尽力保证正文或代码中不出现错误，但是错误总是难免的。如果您在本书中找到了错误，如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们就会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 http://www.wrox.com，通过 Search 工具或书名列表查找本书，然后在本书的详细信息页面单击 Book Errata 链接。在打开的页面可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表链接，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，可以给您传送感兴趣的论题。Wrox 作者、编辑、其他业界专家和读者都会到这个论坛上来探讨问题。

在 http://p2p.wrox.com 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发您自己的应用程序。加入论坛的步骤如下：

(1) 进入 p2p.wrox.com，单击 Register 链接。

(2) 阅读使用协议，并单击 Agree 按钮。

(3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。

(4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

提示：

不加入 P2P 也可以阅读论坛上的消息，但要发布自己的消息，必须加入该论坛。

加入论坛后，就可以发表新消息，响应其他用户发表的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目　录

第Ⅰ部分　入　门

第 I 部分

入　门

- 第 1 章　通向测试驱动开发之路
- 第 2 章　单元测试简介
- 第 3 章　重构速览
- 第 4 章　测试驱动开发：以测试为指南
- 第 5 章　模拟外部资源