.NET单元测试艺术

The Art of **Unit Testing: With Examples in .NET**

（以色列）Roy Osherove 著
张昌贵 张博超 李丁山 译
滕振宇 审校

清华大学出版社

.NET 单元测试艺术

(以色列) Roy Osherove 著

张昌贵 张博超 李丁山 译

滕振宇 审校

清华大学出版社

北 京

内 容 简 介

《.NET 单元测试艺术》针对这个重要主题展开讨论，引导读者从简单的测试开始，逐渐过渡到如何写出可维护、可读、可信赖的测试。同时，还涉及 mock，stub 和框架(如 Typemock Isolator 和 Rhino Mocks)等高级主题，旨在帮助读者逐步掌握高级的测试模式和结构，高效地为遗留代码和甚至根本不可测试的代码编写测试。书中还讨论了测试数据库时需要的工具和其他技术。本书为广大.NET 开发人员而写，但其他读者也可以从中受益。

图书在版编目(CIP)数据

.NET 单元测试艺术/(以色列)奥西洛夫(Osherove, R.)著；张昌贵，张博超，李丁山译；滕振宇审校.
--北京：清华大学出版社，2012.1
书名原文：The Art of Unit Testing: With Examples in .NET
ISBN 978-7-302-26916-8
Ⅰ. ①N… Ⅱ. ①奥… ②张… ③张… ④李… ⑤滕… Ⅲ. ①开发工具－程序设计 Ⅳ. ①TP393.092
中国版本图书馆 CIP 数据核字(2011)第 187899 号

责任编辑：文开琪　汤涌涛
封面设计：杨玉兰
版式设计：北京东方人华科技有限公司
责任校对：周剑云
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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 装 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185×230　**印　张**：20　**字　数**：434 千字
版　　次：2012 年 1 月第 1 版　　**印　　次**：2012 年 1 月第1 次印刷
印　　数：1～4000
定　　价：49.00 元

产品编号：042223-01

译 者 序

在软件开发行业中，一个项目或产品的成功与否取决于很多因素，可靠的代码质量是其中的关键因素之一。而为了确保代码质量，相信很多人都尝试过测试驱动开发(TDD)，或者知道它能在提高代码质量方面提供帮助。但是真正能把测试驱动开发用得好的人并不多见，而且有不少人认为只要是先写测试再写代码，那就是测试驱动开发。很多人都浅尝辄止，结果发现测试驱动开发不但没有带来好处，反而造成很多的困扰。

记得今年春节前后，有人在个人博客上评论了 TDD 的优缺点，而后 InfoQ 中文网站就专门针对这篇博客，邀请了一些社区活跃分子就 TDD 的经验和体会做了一个虚拟会谈(具体内容请查阅 InfoQ 中文网站的文章)。由于每个人对于 TDD 的理解和见识不尽相同，所以争论也就无法避免。

而本书绕开 TDD 这个概念，针对的是这个概念背后真正保证代码质量的最关键部分——单元测试。无论是先写单元测试，还是后写单元测试，最终都是通过单元测试来保证代码质量。所以，单元测试才是重中之重，它是程序员的一门必修课，如何编写优秀的单元测试是每个程序员都必须学会的一种技能，甚至可以说是一种生存技能。

所以，作为在软件开发行业里摸爬滚打了十多年的先行者，我们觉得有义务也有责任为程序员推荐一本好书，一本可以在日常开发工作中提供很大帮助的书——特别是为那些新入行的程序员兄弟姐妹，或者需要在单元测试方面系统学习的朋友。

北欧敏捷实践者 Jurgen Appelo(被誉为 21 世纪管理圣经的《管理 3.0》一书的作者)在其著名的博客 *www.noop.nl* 推荐了一百本敏捷书籍，其中不乏一些名人大家的作品，比如 Martin Fowler 的《重构》，Kent Beck 的《测试驱动开发》。而本书在 2011 年全球敏捷大会期间排名第 1。如果你们有.NET 基础或者能看懂 C#代码，那么这本书就是帮助你学习单元测试的不二选择。

最后，希望读者可以从这本书中获益，并把学到的知识和技能应用到实际工作中，为项目和产品带来直接的好处。

张昌贵　张博超　李丁山

2011 年 7 月于上海

序　言

当 Roy Osherove 告诉我他正在写一本关于单元测试的书时，我感到非常高兴。测试基因(meme)在业内已经提了好多年，但单元测试的相关资源仍然比较缺乏。在我的书架上，我能找到测试驱动开发方面的书，一般性的测试方面的书，但到目前为止还没有任何一本全面介绍单元测试的参考书——没有一本书介绍这个主题并引导读者从第一步走到广泛接受的最佳实践。这个事实的真实性很令人震惊。单元测试不是一项新的实践，但我们为何会发展到这种境况？

虽说“我们这个产业还非常年轻”这话是老生常谈，但事实也的确如此。数学家早在近百年前就为我们现在的工作奠定了基础，但直到最近六十年前，我们才有足够快速的硬件得以利用他们的洞察力。在我们这个产业，理论与实践之间天生就有差距，而我们直到现在才发现它是如何影响了我们这个领域的。

在早期，机器运行周期是相当昂贵的，我们以批量方式来运行程序。程序员有一个预定的时间段，他们必须以打孔的方式把程序打到纸带上并送到机房。如果程序不正确，你的时间就没了，只得趴在桌子上用铅笔和纸对自己的程序进行“案头检查”(desk-check)，人工走查所有场景和所有边界用例。我甚至怀疑当时压根儿没有人能够想象到“自动化单元测试”这个概念。机器本来是用来解决问题的，在可以用它解决问题时，为什么要用它来做测试呢？机器的稀缺性让我们一直处于这种蒙昧状态。

后来，机器越来越快，我们也越来越陶醉于交互计算。我们可以输入

代码而后心血来潮想改就改。“案头检查”代码的概念逐渐消失，我们逐渐失去早年所有的一些规范。我们知道编程很难，但那只是意味着我们必须在计算机上花更多的时间更改代码行和符号，直到找到行之有效的魔咒。

我们从稀缺到过剩，唯独错过了中间地带，但现在我们正重新返回中间地带。自动化单元测试结合了“案头检查”的规范与把计算机当作开发资源的新理念。我们可以使用我们开发所用的语言，写自动化测试来检查我们的工作——不只是一次，而是尽量经常运行。我觉得在软件开发中没有其他实践可以如此强大了。

在 2009 年，我写这段文字的时候，我们很高兴见到 Roy 的书出版了。这是一本实用指南，它可以帮助你入门，同时也是一本帮助你深入测试领域的优秀参考书。这不是一本关于理想化场景的书。它以这个领域中存在的方式来教你如何测试代码，如何利用广泛使用的框架，并且最重要的是如何编写更容易测试的代码。

《.NET 单元测试艺术》很重要，这样的书本应该在几年前就出现，但那时我们尚未准备就绪。现在，我们已经准备妥当，请好好享用吧！

Michael Feathers

资深咨询师，OBJECT MENTOR

关 于 本 书

如何使用本书

如果你以前从未写过单元测试，最好从头读到尾，这样你才能了解大局。如果你已经有这方面的经验，你可以跳到任意章节。

本书读者对象

本书是针对那些编写代码并且有兴趣学习单元测试最佳实践的读者而编写的。所有代码范例都是在Visual Studio 2008里用C#写的，所以程序员会发现这些范例特别有用。但是我讲授的这些课程，即使不是全部，大都也适用于大部分静态的面向对象语言(举几个例子，如VB.NET、Java和C++)。如果你是程序员、团队领导者、测试人员(写代码的)或者新手，本书也同样适用。

路线图

本书分成四部分。

第 I 部分带领你在编写单元测试方面从零分到六十分。第 1 章与第 2 章涵盖了基础知识，例如如何使用测试框架(NUnit)，也介绍了基本的自动化测试特性，例如[Setup]和[TearDown]。这两章也介绍断言、忽略的测试和基于状态测试的一些概念。

第 II 部分讨论了打破依赖的高级技术：模拟对象、桩对象、模拟框架，以及使用这些技术来重构代码的模式。第 3 章介绍了桩对象的概念以及如何手工创建并使用它。第 4 章介绍了使用手写模拟对象做交互测试。第 5

章合并桩对象与模拟对象这两个概念，并展示了隔离(模拟)框架如何结合这两种想法以及如何让它们自动化。

第III部分谈到了组织测试代码的方式、运行并重构其结构的模式，以及编写测试的最佳实践。第 6 章讨论了测试层次，如何使用测试基架 API，以及在自动化构建过程中如何合并测试。第 7 章讨论了创建可维护、可阅读与可信赖的单元测试的最佳实践。

第IV部分谈到了如何在一个组织内实施变革以及如何在已有代码上工作。第 8 章讨论了当试图把单元测试引入到一个组织内时将会碰到的问题和解决方案，同时还提出了一些你可能被问到的问题与解答。第 9 章谈论的是如何在已有代码中引入单元测试。它提出了一些确定从哪里开始测试的方法，并讨论了一些工具用来测试难以测试的代码。

最后是两个附录。附录 A 讨论了可测试性设计的话题以及其他当前存在的变通方式。附录 B 里列出了一些你可能觉得能在测试中提供帮助的工具。

代码约定以及下载

你可以从本书的网站 *www.ArtOfUnitTesting.com* 下载到源代码，也可以到出版社网站 *www.manning.com/TheArtofUnitTesting* 下载。

本书列出的所有源代码都使用等宽字体。在代码清单中，**粗体代码**表示的是与前一个范例相比所改变的代码，或者在下一个范例中会改变的代码。

代码清单中伴随的一些代码注解用于突出重要的概念。在某些情况下，带编号的项目在文中有注释。

软件需求

要使用本书中的代码，至少需要 Visual Studio C# Express(免费的)或者更高级的版本(例如专业版或 Team Suite)，还将需要 NUnit(开源的免费

框架)及其他相关工具。上述所有工具要么是免费开源的，要么有可以免费使用的试用版。

作者在线

凡是购买了本书原版的读者，都可以免费使用 Manning 出版社的论坛，你可以在论坛里讨论本书，提技术问题，并且会获得来自作者及其他读者的帮助。如需进入论坛，请在浏览器访问 *www.manning.com/TheArtofUnitTesting*，会告诉你一些信息，包括如何在注册后进入论坛，会有哪些帮助，以及论坛行为准则。

Manning 出版社对读者的帮助，包括提供一个场所供读者之间，以及读者与作者之间进行有意义的对话。但是并不保证作者的参与程度，作者对本书论坛的贡献是自愿的(并且是无偿的)。我们建议你问一些有挑战性的问题，以免他不感兴趣。

作者在线论坛与以前的讨论归档，只要本书还在出版，都将可以在出版社网站访问到。

致　　谢

非常感谢 Manning 出版社的 Michael Stephens 和 Nermina Miller，他们在我写这本书的整个过程中都保持了巨大的耐心。

感谢 Jim Newkirk、Michael Feathers、Gerard Meszaros 以及其他很多人，他们给了我本书的灵感并出谋划策。还特别感谢 Michael 同意为本书写序。

以下评审者在本书撰写过程的不同阶段阅读了原稿。我想感谢他们提供了非常宝贵的意见和建议：Svetlana Christopher，Wendy Friedlander，Jay Flowers，Jean-Paul S. Boodhoo，Armand du Plessis，James Kovacs，Carlo Bottiglieri，Ken DeLong，Dusty Jewett，Lester Lobo，Alessandro Gallo，Gabor Paller，Eric Raymond，David Laribee，Christian Siegers，Phil Hanna，Josh Cronemeyer，Mark Seemann，Francesco Goggi，Franco Lambardo，Dave Nicolette 和 Mohammad Azam。还感谢 Rod Coffin 在本书即将出版前做了技术校对。

最后感谢那些 ManningEAP(Early Access Program)的早期读者在论坛里提供的评论。他们对本书的定稿提供了帮助。

前　言

我曾参与过一个最失败的项目，至少我是这么觉得的，它拥有单元测试。我当时带领着一组程序员写一个计费程序，而且我们完全是以测试驱动的方式开发的——写测试，然后写代码，测试先失败然后通过，重构，而后重新再来。

这个项目的最初几个月非常好，一切都很顺利，并且我们有测试足以证明代码可以正常工作。但随着时间的推移，需求发生了变化。我们不得不改变代码以适应新的需求，然而在修改代码之后，测试却失败了，我们只好回头修复代码。代码仍然可以工作，但测试却很脆弱，任何一点微小的代码改动都会导致测试失败，即便代码本身根本没有问题。因为我们必须同时修复代码的相关测试，所以改变类或方法变成了一项艰巨的任务。

更糟糕的是，一些测试变得无法使用了，因为写这些测试的人离开了该项目，而且没有人知道如何维护这些测试，或者它们测的是什么。单元测试的方法命名不够清晰，而且有些测试依赖于其他测试。结果项目开始不到六个月，我们就抛弃了大部分测试。

这个失败的项目是一个悲剧，因为我们所写的测试做了弊大于利的事情。在长期运行中，它们消耗了更多的时间来维护和理解，而非节约时间，所以我们就停止继续使用。后来我到了另一个项目，在该项目中我们编写单元测试方法好了很多，而且通过使用单元测试我们取得了巨大的成功，节省了大量调试与集成的时间。自从那个失败的项目之后，我就已经开始整理单元测试的最佳实践，而且在后续项目中使用这些最佳实践。在我参

与的每个项目中我都发现了更多的最佳实践。

理解如何编写单元测试——如何使其拥有可维护性、可读性和可依赖性——是本书所述的主要内容，无论你使用何种语言或何种 IDE。本书涵盖了编写单元测试的基础，集成测试的基础，而后介绍了在现实世界中编写、管理和维护单元测试的最佳实践。

目　　录

第I部分　入　　门

第 II 部分 核 心 技 术

第 III 部分 测试的代码