



Professional Software Testing
with Visual Studio 2005 Team System

Tools for Software Developers and Test Engineers

**Visual Studio 2005 Team
System 软件测试专家教程**



(美) Tom Arnold 等著
Dominic Hopton
颜 炯 译

Visual Studio 2005 Team System

软件测试专家教程

(美) Tom Arnold 等著
Dominic Hopton
颜 炯 译

清华大学出版社

北 京

Tom Arnold, Dominic Hopton, et al

Professional Software Testing with Visual Studio 2005 Team System: Tools for Software Developers and Test Engineers

EISBN: 978-0-470-14978-2

Copyright © 2007 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2008-0826

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Visual Studio 2005 Team System 软件测试专家教程/(美)阿纳尔德(Arnold, T.) 等著; 颜炯 译. —北京: 清华大学出版社, 2008.10

书名原文: Professional Software Testing with Visual Studio 2005 Team System: Tools for Software Developers and Test Engineers

ISBN 978-7-302-18552-9

I. V… II. ①阿… ②颜… III. 计算机网络—程序设计 IV.TP393

中国版本图书馆 CIP 数据核字(2008)第 140766 号

责任编辑: 王 军 于 平

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 何 芊

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 刷 者: 北京市世界知识印刷厂

装 订 者: 三河市漂源装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 24.75 字 数: 602 千字

版 次: 2008 年 10 月第 1 版

印 次: 2008 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 49.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 027545-01

前言

我有幸参与过许多创建性的工作。当时，我们三个人着手创建一个公司，这个公司后来发展为一个拥有 250 名员工，实际年销售额达到一千四百万美元的公司。在这个公司，我还参与创建了一个新部门。在各种条件下，我们都认真思考过这类问题：在一个开发团队中，到底都需要什么样的开发工具，如何将这些开发工具集成起来，团队应该如何有效地利用这些工具，在尽可能短的时间内开发出高质量的产品，从而满足市场的需要。

在典型情况下，这意味着我们必须构建我们自己的工具“仓库”，在这个“仓库”中，应该包括以下软件：诸如 CVS(Concurrent Versions System)之类的开源版本控制系统、Serena Software 公司开发的用于缺陷跟踪和问题跟踪的软件工具 Team Track、Mercury Interactive 公司开发的用于 Web 测试和负载测试的自动化工具、诸如 JUnit 和 NUnit 之类的开源单元测试工具、Bullseye Testing Technology 公司开发的代码覆盖分析工具 C Cover、Rational Software 公司开发的测试库管理工具、BEA Systems 公司开发的 Plumtree 或微软公司开发的 SharePoint 之类的文档共享管理工具，等等。

一旦确定了所需要的工具仓库，我们就可以对软件项目生命周期进行讨论，并就开发过程取得一致意见。我们是采用瀑布方法、敏捷方法、螺旋开发方法，还是 Rational 统一过程(RUP)、微软解决框架(MSF)，还是其他某种开发方法？如果我们希望开发团队能够高效地工作，那么我们就必须回答这些问题。

选定了工具并确定了开发过程之后，开发团队就要着手利用这些工具完成版本控制、代码覆盖、动态分析和结果测试等工作，并使用诸如微软公司开发的 Excel 之类的应用程序来记录项目的进展情况。这样，我们可以知道是否能够按时发布软件。

对开发团队而言，掌握项目总体进度是一项非常重要的工作。如果我们再考虑到投资人的期望——无论他们是风险投资人，还是对新技术提供资金资助的公司，抑或是朋友或家庭，他们对我们进行资助并希望能够帮助我们的事业起步——那么掌握项目总体进度就更为重要。因为投资人慷慨投资时承担了巨大的风险，所以他们有权利知道项目的进展情况。

我们再看看 Visual Studio 2005 Team System(VSTS)。这是一个工具套件，主要用于为软件架构设计人员、开发人员、测试工程师、程序开发管理人员和为项目提供资金的人士提供相关支持。VSTS 不仅提供了工具库，同时还为这些工具能够协同工作提供了支持。我们知道：无论这些工具的功能是每天晚上执行一次完整构建，还是对当前 bug 的现况进行记录，将这些工具集成起来使之能够协同工作，是一项非常关键的功能。微软公司开发的这个软件套件确实是一项创举，目前已经得到了广泛的好评。

当我回想起 VSTS 的开发过程时,我必须指出:我所经历的所有一切都令我感到畏惧,但是,这个套件毕竟能够使我们将需求逐步分解为规模可控制的软件组件。本书仅关注 VSTS 提供的两项功能,专门讨论为开发人员和测试人员提供的工具。也就是说,本书关注 Visual Studio Team Edition for Software Developers(VSTESD)和 Visual Studio Team Edition for Software Testers(VSTEST),这两项内容也是 Visual Studio Team System 的组成部分。

本书的目的在于:

- 向读者介绍 Visual Studio Team System 的总体概况。
- 引导读者初步了解开发和测试工具 VSTEST 和 VSTESD 的用户界面。
- 深入研究测试工程师和软件开发人员可以使用的每一类测试类型。
- 帮助读者理解如何高效地使用代码分析工具和动态分析工具。
- 对工具提供的各项功能进行总结,掌握在团队开发过程中如何有机地使用工具的各项功能。

本书的合作作者、本书编辑,以及本人之所以决定编写本书,是因为我们当中的许多人或者参与了微软公司开发团队并开发了本书所讨论的测试工具,或者能够在软件开发实践中熟练高效地使用这些工具,我们希望能够与读者分享我们的经验,从而使读者能够掌握工具的使用环境,并了解如何使用这些工具,使这些工具能够发挥最大效用,并且使读者在参与某个软件开发团队并作为软件开发人员或者软件测试工程师的情况下,能够尽快熟练高效地使用这些工具。

本书读者对象

VSTS 使得软件测试工程师和软件开发人员的职责不再泾渭分明,因此,本书的主要读者是软件测试工程师和软件开发人员。这两类读者在使用测试和分析工具时,会感到本书非常有用。软件开发人员会对单元测试、代码分析和事件探查工具(profiling tool)充满兴趣,因为利用这些工具有利于他们开发高质量的代码。测试工程师需要使用软件开发人员编写的单元测试代码,可能也需要自己开发部分单元测试代码,还需要自己开发 Web 测试、负载测试和手工测试代码,从而能够为软件产品提供一个完整的测试方案。测试人员和开发人员都需要利用测试执行结果来进行代码分析,他们还希望能够利用图形用户界面来创建、组织和维护相关测试。同时,他们还希望能够利用自行创建的测试库,针对每个最新构建的版本进行测试,并根据测试结果确定是否可以对软件进行最后的安装测试。无论是开发人员还是测试人员,都希望了解如何将其他工具作为插件安装到 VSTS 中,并作为 VSTS 的一部分与 VSTS 集成使用。

本书假定读者基本掌握了 Visual Studio,并可以利用 Visual Studio 完成相关工作。这个假定对大多数开发人员都是成立的,因为很多开发人员都学习使用 Visual Studio 作为开发工具。但是,如果某位软件测试人员以前主要是使用测试工具进行测试,并且从来没有编写过代码,那么本书内容对他们就不是很容易理解,这是因为本书所讨论的工具是 Visual Studio 的有机组成部分,而且这些工具是从底层实现的测试工具,所以对这些测试人员而言,掌握本书的内容会存在一些难度。此外,本书不打算花费很多笔墨来讨论 Visual Studio

本身，我们主要关注 VSTS 中 VSTEST 和 VSTESD 所提供的测试和分析工具。

本书不讨论与任何具体测试方法学有关的细节，也不推荐使用任何具体的软件开发生命周期。VSTEST 和 VSTESD 本身并不依赖于任何特定的测试理论，也不依赖于任何具体的软件开发生命周期，VSTEST 和 VSTESD 可以与任何开发方法集成，并且可以应用于任何项目。在附录 F 中，我们给出了与上述内容相关的链接，读者可以通过阅读链接内容学习相关知识。

本书主要内容

本书关注 Visual Studio 2005 Team System 中的 Visual Studio Team Edition for Software Developers(VSTESD)和 Visual Studio Team Edition for Software Testers(VSTEST)所提供的工具。

本书具体内容包括：

- 用于创建、授权、执行和管理测试的用户界面，具体包括
 - 单元测试(包括数据库单元测试)
 - Web 测试
 - 负载测试
 - 手工测试
 - 顺序测试
 - 一般测试
 - 第三方测试
- 分析工具：
 - 代码分析(也称为静态分析)
 - 动态分析(事件探查和代码覆盖)
- 与 Visual Studio Team Foundation Server 的集成

本书所讨论的具体内容是非常清楚的，因为这些测试类型和分析工具是第一次被集成到 Visual Studio 中，所以我们不需要考虑这些工具是否存在先前的版本，只需要学习如何将它们应用于项目中就足够了。

本书组织结构

在一本书里同时满足两类完全不同读者的需要确实不是一件容易的事。本书主要面向两类读者，这两类读者分别是测试人员和开发人员，虽然这两类读者都属于工程师，并且都需要使用测试工具来保证软件项目的质量，然而，如果开发团队采用了不同的开发方法，遵循了不同的软件开发方法学，那么，在软件开发生命周期中，测试工具的应用阶段必然有所不同。因此本书并不局限于某种特定的开发方法学，也没有局限于某个具体的软件开发生命周期，而是集中关注如何使用 VSTEST 和 VSTESD，以及如何将这些工具集成到读

者使用的开发过程中。

本书内容如下：

- 第1章, VSTEST和VSTESD简介——本章主要从整体上介绍了VSTEST和VSTESD的作用和功能。
- 第2章, VSTEST和VSTESD快速浏览——本章介绍了开发创建、授权、执行和管理测试时所使用的用户界面。
- 第3章, 使用VSTEST和VSTESD进行单元测试——本章介绍了使用VSTEST和VSTESD进行单元测试的一般方法。
- 第4章, 对数据库进行测试——本章讨论了如何利用单元测试对数据库进行测试。
- 第5章, Web测试——利用Web测试, 可以将我们与一个Web网站的交互记录下来。本章还介绍了如何将这种交互转化为源代码。
- 第6章, 如何使用手工测试、顺序测试和一般测试类型——本章讨论了如何在一个“容器”中容纳一个或多个测试, 这样不仅提高了测试灵活性, 而且还有效地提高了测试价值。
- 第7章, 负载测试——本章通过模拟多位用户同时访问应用程序, 指导我们如何对系统进行负载测试。
- 第8章, 使用代码分析和动态分析——本章讨论了如何在编译过程中和程序执行过程中对代码进行分析。
- 第9章, 在软件开发生命周期中应用VSTEST和VSTESD——本章讨论了如何在团队开发中使用VSTEST工具和VSTESD工具, 特别讨论了在使用Visual Studio Team Foundation Server(VSTFS)的条件下如何使用VSTEST工具和VSTESD工具。
- 附录——主要介绍了如何使用工具的某些具体功能, 并帮助我们快速浏览每种测试类型的工作流。
 - 附录A, 安装Team Explorer——介绍如何将Visual Studio与开发团队的VSTFS服务器连接起来。
 - 附录B, 创建并运行一个Web测试: 循序渐进指导——简要介绍了如何使用Web测试。
 - 附录C, 创建并运行一个单元测试: 循序渐进指导——简要介绍了如何使用单元测试。
 - 附录D, 创建并运行一个负载测试: 循序渐进指导——简要介绍了如何使用负载测试。
 - 附录E, 创建并运行一个手工测试: 循序渐进指导——简要介绍了如何使用手工测试。
 - 附录F, 其他信息资源——提供了与本书主题相关的Web网站、论坛、会议以及其他信息。

如果读者对VSTS还很陌生, 那么请读者从第1章和第2章开始阅读, 这样, 读者可以了解掌握VSTEST工具和VSTESD工具提供的功能。在此基础上, 读者可以开始阅读与其所从事工作相关的章节。如果读者具备这些工具的使用经验并希望能够深入了解某种具

体的测试类型,或者希望能够深入了解某种工具的使用方法,那么可以直接阅读相关章节。如果读者需要了解掌握工具是如何与 Visual Studio Team Foundation Server (VSTFS)进行集成的,那么就可以直接阅读本书第 9 章。如果读者阅读的目的是学习掌握如何将某种具体的测试类型添加到当前开发的项目中,并编写相关测试,最终能够顺利执行测试,那么请阅读附录 B 到附录 E 的内容。

源代码

读者在阅读本书提供的代码时,既可以手工输入所有代码,也可以使用随书提供的代码文件。本书所有源代码均可以从本书合作站点 <http://www.wrox.com> 或 www.tupwk.com.cn/download 上下载。进入该网站后,请读者查找本书,本书书名为“*Professional Software Testing with Visual Studio 2005 Team System*”(读者既可以使用搜索框进行查找,也可以使用书名标题列表进行查找),然后单击本书详细内容页面上提供的 Download Code 链接,就可以下载本书提供的所有代码。

因为许多书籍名称与本书类似,因此读者也可以通过 ISBN 进行查找,本书的 ISBN 为: 978-0-470-14978-2。

下载代码后,读者可以利用某种压缩工具将代码解压。此外,读者还可以通过访问网站 <http://www.wrox.com/dynamic/books/download.aspx> 中的 Wrox 代码下载页面来获取本书提供的代码,也可以下载 Wrox 出版的其他书籍提供的代码。

勘误表

为了避免本书文字和代码中存在错误,我们已经竭尽全力。然而世界上并不存在完美无缺的事物,所以本书可能仍然存在错误。如果读者在我们编写的某本书籍中发现了诸如拼写错误或代码缺陷等问题,那么请告诉我们,我们对此表示感谢。利用勘误表反馈错误信息,可以为其他读者节省大量时间,同时,有助于我们编写出质量更高的专业著作。

请给 wkservice@tup.tsinghua.edu.cn 发电子邮件,我们就会查阅您的信息,如果是正确的,我们将在本书的后续版本中采用。

如果读者需要参考本书的勘误表,请在网站 <http://www.wrox.com> 中用搜索框或书名列表查找本书书名。然后,在本书的详细内容页面上,单击 Book Errata 链接。在随后显示的页面中,读者可以看到与本书相关的所有勘误信息,这些信息是由读者提交、并由 Wrox 的编辑们加上的。通过访问 www.wrox.com/misc-pages/booklist.shtml,读者还可以看到 Wrox 出版的所有书籍的勘误表。

关于 p2p.wrox.com

如果读者希望能够与作者进行讨论,或希望能够参与到与其他读者的共同讨论中,那么请加入 p2p.wrox.com 的论坛。该论坛是一个基于 Web 的系统,读者可以在论坛发表与 Wrox 出版的书籍有关的技术信息,并与其他读者和技术用户进行讨论。论坛提供了订阅功能,可以将与读者所选定主题相关的新帖子定期发送到读者的电子邮箱。Wrox 的作者、编辑、工业界专家,以及其他读者都会参与论坛中的讨论。

读者可以在 <http://p2p.wrox.com> 参与多个论坛的讨论,这些论坛不仅能够帮助读者更好地理解本书,还有助于读者更好地开发应用程序。如果希望加入论坛,那么请按照以下步骤执行:

- (1) 进入 p2p.wrox.com 页面,单击 Register 链接。
- (2) 阅读使用条款,单击 Agree 按钮。
- (3) 填写必要的信息,必要时也需要填写可选信息,单击 Submit 按钮。
- (4) 随后读者会收到一封电子邮件,邮件中说明了如何验证账户并完成整个加入过程。

为了阅读论坛信息,读者无须加入 P2P。但是如果读者需要发表主题或发表回复,那么读者必须加入论坛。

成功加入论坛后,读者就可以发表新主题了。此时,读者还可以回复其他主题。读者在任何时间都可以阅读论坛信息。如果读者需要论坛将新的信息发送到自己的电子邮箱,那么可以单击论坛列表中论坛名称旁的 Subscribe to this Forum 图标完成这项功能。

如果读者需要获得更多与使用 Wrox P2P 相关的信息,请阅读 P2P FAQ,这样可以获得许多论坛软件如何使用的回答,以及大量与 P2P 和 Wrox 出版的书籍相关的具体信息。阅读 FAQ 时,请单击 P2P 页面上的 FAQs 链接。

译者序

自从成功推出了 Visual Studio 2005 Team System(VSTS), 微软公司第一次为软件开发人员和软件测试工程师提供了一个能够覆盖整个软件开发生命周期的团队协作开发平台。VSTS 提供的测试及开发工具 Visual Studio Team Edition for Software Testers(VSTEST)和 Visual Studio Team Edition for Software Developers(VSTESD)为软件测试工程师和软件开发人员提供了多种测试类型, 可以帮助软件测试工程师和软件开发人员完成多种测试工作, 还提供了代码静态分析和程序动态分析功能, 从而使得软件测试工程师和软件开发人员之间的界线不再泾渭分明。

当前, 越来越多的开发团队已经开始使用 VSTEST 和 VSTESD 作为测试工具, 并基于 VSTS 作为整个开发的软件基础结构。然而, 开发应用程序的过程中, 开发人员常常会遇到一些意想不到的困难, 例如, 如何有效地对应用程序后台数据库进行高效率的单元测试, 如何对 Web 应用程序进行负载测试, 以及如何分析测试数据等, 入门似乎很容易, 但是一旦涉及到如何在软件开发生命周期内, 并且在整个开发团队协作的前提下完成相关测试工作, 许多开发人员就不得不在 MSDN 联机手册中以及互联网上苦苦寻找答案。

本书正是基于上述原因, 讨论了如何在软件开发生命周期内并在整个开发团队协作的前提下, 如何对应用程序进行高效的测试, 而这些主题或者在别的书籍中很少讨论到, 或者分散在不同的角落里, 需要读者花费大量的精力才能找到。读者可以从本书了解到 VSTS 的总体概况, 学习到 VSTEST 和 VSTESD 提供的每一类测试类型, 并且可以掌握如何高效地使用代码分析工具和动态分析工具。

在翻译本书的过程中, 我们感到作者对如何将 VSTEST 和 VSTESD 应用于软件开发生命周期过程中的认识极为深刻, 并对整个开发团队协作开发过程中如何使用 VSTS 具有深刻的认识。特别是, 作者的讨论时刻不偏离团队开发过程和软件开发生命周期这两个重要的概念, 强调了在团队开发过程和软件开发生命周期中如何有机地使用工具的各项功能, 这也是本书与其他介绍 VSTEST 和 VSTESD 的技术书籍的最大区别, 也是本书的技术亮点所在。

本书的主要读者是软件测试工程师和软件开发人员。软件开发人员可以使用单元测试、代码分析和事件探查工具, 测试工程师需要使用软件开发人员编写的单元测试代码, 也需要自己开发部分单元测试代码, 还需要自己开发 Web 测试、负载测试和手工测试代码, 最终能够为软件产品提供一个完整的测试方案。

本书作者均为 VSTS 测试及开发工具的开发团队成员, 因此, 本书所讲述的工具使用方式都具有高度的针对性。通过阅读本书, 我们可以了解掌握每一种测试类型, 学习到如何高效地使用代码分析工具和动态分析工具, 从而进一步提高自身的技术水平。

在本书的翻译过程中，我们本着对读者认真负责的精神，力求做到技术内涵的准确无误和专业术语的规范统一，力求做到翻译的准确性和灵活性的有效结合。由于我们的水平有限，错误与不到位之处在所难免。欢迎广大读者提出宝贵的意见。

译者

2008年1月27日

关于作者

Tom Arnold 是 Microsoft Visual Studio 2005 Team System 的程序开发主管，主要负责测试工具开发。他还曾经负责过微软公司和 Rational Software 公司的其他商用测试工具项目的开发工作，并在 STAR、Microsoft Tech Ed、Internet World 等会议上担任过发言人。作为本书作者之一，Tom Arnold 还编写过其他两部有关软件测试的著作，并出版了三部与软件测试相关的录像带。此外，作为创始人之一，Tom Arnold 参与创建了一个拥有 250 位员工的软件测试公司(后来该公司被 Lionbridge/Veritest 收购)。目前，Tom 是微软公司 OfficeLive.com 项目小组的程序开发主管。如果读者希望与 Tom 进行联系，请浏览网站 <http://www.msvstest.com>。

Dominic Hopton 是微软公司 Visual Studio Enterprise Test 项目小组的软件开发工程师，作为该小组的最早成员之一，Dominic 帮助对 Visual Studio 2005 中的开发和测试工具进行了革新。他目前从事 Visual Studio 新一代测试工具的开发工作。

Andy Leonard 是一位 SQL Server 数据库开发人员，同时还是微软公司的工程师，而且，他还是微软公司的 MVP。Andy Leonard 是 *Professional SQL Server 2005 Integration Services* 一书的作者之一，他创建了 VSTeamSystemCentral.com，同时负责该网站的管理工作。Andy 曾经从事过 Web 应用程序架构设计和程序开发工作，并参与过 VB.NET、ASP、ASP.NET、SQL Server Integration Services(SSIS)和数据仓库的开发工作(分别使用 SQL Server 2000 和 SQL Server 2005)，还参与过测试驱动的数据库开发工作。

Mike Frost 是微软公司一位资深的“测试方面的软件设计工程师”(也就是说，他是一位以编写代码为主要工作的软件测试工程师)。在最近十年中，Mike Frost 参与了多个版本的 Visual Studio 的开发工作，他主要关注与软件测试相关的底层结构设计和架构设计。他目前从事新一代 OfficeLive.com 服务产品的开发工作。

目 录

第 1 章	VSTEST 和 VSTESD 简介	1
1.1	软件开发过程和软件开发生命周期	1
1.1.1	规划、分析、设计和开发	1
1.1.2	市场上现有的工具	3
1.2	VSTEST 和 VSTESD 适用于软件开发生命周期的哪个阶段	6
1.3	VSTEST 和 VSTESD 能够完成什么工作	7
1.3.1	VSTEST 提供的功能	7
1.3.2	VSTESD 的特点	11
1.3.3	VSTEST 和 VSTESD 如何在 SDLC 中发挥作用	12
1.3.4	为什么选择了 VSTEST 而没有选择其他工具集	13
1.4	小结	15
第 2 章	VSTEST 和 VSTESD 快速浏览	17
2.1	测试类型	17
2.1.1	手工测试类型	18
2.1.2	单元测试类型	19
2.1.3	Web 测试类型	19
2.1.4	负载测试类型	20
2.1.5	顺序测试类型	20
2.1.6	一般测试类型	20
2.1.7	未来/第三方测试类型	20
2.2	测试用户界面	21
2.2.1	创建和编写测试的用户界面	21
2.2.2	管理测试的用户界面	30
2.2.3	执行测试和观察测试执行结果的用户界面	33
2.3	分析工具	40
2.3.1	代码分析	40
2.3.2	性能度量	41
2.4	小结	42
第 3 章	使用 VSTEST 和 VSTESD 进行单元测试	45
3.1	单元测试概要	45
3.1.1	为什么进行单元测试	45

3.1.2	单元测试方法	46
3.1.3	VSTEST 和 VSTESD 的单元测试有什么独特之处	48
3.2	创建第一个单元测试	49
3.2.1	创建一个示例库	49
3.2.2	生成单元测试	51
3.2.3	运行单元测试	53
3.3	VSTEST 中的单元测试	53
3.3.1	剖析一个测试类	53
3.3.2	测试通过和测试失效	54
3.3.3	代码生成	54
3.3.4	定制测试方法	59
3.3.5	单元测试断言	62
3.3.6	对单元测试进行调试	63
3.4	高级单元测试	63
3.4.1	MSTest 命令行工具	66
3.4.2	单元测试生命周期	71
3.4.3	ASP.NET	75
3.4.4	数据驱动的测试	76
3.4.5	测试上下文	78
3.5	小结	79
第 4 章	对数据库进行测试	81
4.1	数据库测试是不同的	81
4.2	开发一个数据库项目	81
4.2.1	数据库项目探索研究	83
4.2.2	为数据库添加信息架构、数据表和约束	86
4.2.3	添加视图和存储过程	92
4.2.4	当前开发数据库的内部工作过程	96
4.3	重命名重构	97
4.4	数据库项目部署	100
4.5	生成测试数据	101
4.6	单元测试	104
4.7	开发定制的单元测试	126
4.7.1	添加定制的测试条件项目	126
4.7.2	构建定制的测试条件项目	127
4.7.3	构建“挂钩”	132
4.7.4	使用定制的测试条件开发单元测试	136
4.8	小结	137

第 5 章 Web 测试	139
5.1 基本 Web 测试	139
5.1.1 录制一个测试	140
5.1.2 使用 Web 测试编辑器	142
5.1.3 Web 测试的组件	143
5.2 使 Web 测试动态化	151
5.2.1 上下文参数	151
5.2.2 数据驱动的 Web 测试	155
5.3 在 Web 测试过程中捕获数据	158
5.3.1 提取规则	159
5.3.2 默认(即自动捕获的)字段	160
5.4 使用确认规则对响应进行验证	161
5.4.1 内置的确认规则	161
5.4.2 为一个请求添加一个确认规则	162
5.4.3 确认等级	163
5.5 自定义规则	164
5.6 响应页面	167
5.7 对 Web Service 进行测试	167
5.8 基于 HTTP 请求的测试无法测试用户界面	168
5.8.1 我们不能用 Web 测试对 Web 用户界面和客户端脚本进行测试	169
5.8.2 受脚本影响的动态测试数据	169
5.9 运行一个 Web 测试	170
5.9.1 .testrunconfig 文件中的重要设置	170
5.9.2 运行测试	171
5.10 检查测试结果	171
5.10.1 在 Web Test Viewer 中检查 Web 测试的请求和响应	172
5.10.2 Request 面板	172
5.10.3 Details 面板	172
5.10.4 失效研究	173
5.11 维护测试	176
5.12 可编程 Web 测试	176
5.12.1 可编程 Web 测试和基本的 Web 测试	176
5.12.2 将一个 Web 测试转换为可编程 Web 测试	177
5.12.3 可编程 Web 测试的方方面面	178
5.13 插件	180
5.13.1 Web 测试插件	181
5.13.2 Web 测试请求插件	181
5.13.3 可编程 Web 测试插件	181
5.14 Web 测试和 AJAX	182

5.15 小结	183
第 6 章 如何使用手工测试、顺序测试和一般测试类型	185
6.1 手工测试类型	185
6.1.1 定义一个手工测试场景	186
6.1.2 创建一个手工测试	187
6.1.3 手工测试的属性	191
6.1.4 执行一个手工测试	193
6.2 顺序测试类型	198
6.2.1 创建一个顺序测试	198
6.2.2 顺序测试的属性	199
6.2.3 执行一个顺序测试	202
6.3 一般测试类型	203
6.3.1 创建一个一般测试	204
6.3.2 一般测试的属性	208
6.3.3 执行一个一般测试	212
6.4 小结	212
第 7 章 负载测试	213
7.1 基本术语	213
7.2 使用 VSTEST 的负载测试功能	215
7.3 创建一个负载测试	216
7.4 一个负载测试的组成部分	217
7.4.1 场景	217
7.4.2 计数器集	223
7.4.3 运行设置	228
7.5 SQL 跟踪	232
7.5.1 与 SQL 跟踪有关的运行设置	232
7.5.2 观察 SQL 跟踪的数据	233
7.6 使用远程测试机组	234
7.6.1 确定硬件需求	235
7.6.2 设置远程测试机组	238
7.6.3 管理远程测试机组	239
7.7 负载测试结果存储区	243
7.7.1 在一个现有的 SQL Server 中创建一个负载测试结果存储区	243
7.7.2 负载测试结果存储区数据库架构	244
7.8 最佳实践	245
7.8.1 在明确的目标指导下开始工作	246
7.8.2 了解掌握测试环境	247
7.8.3 了解何时进行负载测试	247

7.8.4	欲跑先学走	248
7.9	运行一个负载测试	250
7.9.1	.testrunconfig 文件中的重要设置	250
7.9.2	启动负载测试	251
7.9.3	检查生成的测试数据	251
7.9.4	使用 Load Test Analyzer	253
7.10	对测试结果进行研究	258
7.10.1	充分理解被测应用程序	259
7.10.2	使用规则	259
7.10.3	检查参与测试的计算机	260
7.10.4	对测试运行进行分析	261
7.10.5	没有发生错误时的处理方法	266
7.11	测试结果报表	267
7.12	小结	267
第 8 章	使用代码分析和动态分析	269
8.1	代码分析	269
8.1.1	静态分析工具	269
8.1.2	代码编写标准	270
8.1.3	启用代码分析	270
8.1.4	理解错误信息	272
8.1.5	停止显示错误信息	274
8.2	动态分析	277
8.2.1	其他工具	277
8.2.2	探查事件	283
8.2.3	代码覆盖	293
8.2.4	命令行越野赛	297
8.3	小结	300
第 9 章	在软件开发生命周期中应用 VSTEST 和 VSTESD	301
9.1	VSTFS 概览	301
9.2	软件开发生命周期	303
9.3	与开发团队协同	304
9.3.1	安装 Team Explorer	305
9.3.2	连接到一个项目	306
9.3.3	项目的高级视图	307
9.4	VSTEST 和 VSTESD 工具是如何协同工作的	311
9.4.1	团队项目源代码控制策略	312
9.4.2	将测试与工作项进行关联	323
9.4.3	团队构建过程	326