



Professional Application Lifecycle Management with Visual Studio 2012

Visual Studio 2012

应用生命周期管理高级教程 (第2版)

涵盖 Visual Studio 2012 和 Team Foundation Server 2012

Mickey Gousset
[美] Brian Keller 著
Martin Woodward
朱永光 吴慧峰 译



清华大学出版社

.NET 开发经典名著

Visual Studio 2012 应用生命周期管理 高级教程 (第 2 版)

Mickey Gousset

[美] Brian Keller 著

Martin Woodward

朱永光 吴慧峰 译

清华大学出版社

北 京

Mickey Gousset, Brian Keller, Martin Woodward
Professional Application Lifecycle Management with Visual Studio 2012
EISBN: 978-1-118-31408-1
Copyright © 2012 by John Wiley & Sons, Inc., Indianapolis, Indiana
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: 01-2013-5116

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

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 2012 应用生命周期管理高级教程: 第2版 / (美)古塞(Gousset, M.), (美)凯勒(Keller, B.), (美)伍德沃(Woodward, M.) 著; 朱永光, 吴慧峰 译. —北京: 清华大学出版社, 2014
(.NET 开发经典名著)

书名原文: Professional Application Lifecycle Management with Visual Studio 2012

ISBN 978-7-302-34942-6

I. ①V… II. ①古… ②凯… ③伍… ④朱… ⑤吴… III. ①程序语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 000288 号

责任编辑: 王 军 韩宏志

装帧设计: 孔祥峰

责任校对: 成凤进

责任印制: 沈 露

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: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 33.25 字 数: 809 千字

版 次: 2014 年 2 月第 1 版 印 次: 2014 年 2 月第 1 次印刷

印 数: 1~3000

定 价: 69.80 元

译者序

记得还在上大学期间，为自学 Visual Basic，我安装了 Visual Basic 5.0，因此首次接触到 Visual Studio 的第一个版本(当时，几个开发工具合称为 Visual Studio 97)。在这些年的开发历程中，一旦 Visual Studio 的新版本发布，我都会立即将其应用于实际工作中。一路走来，感慨颇多，Visual Studio 发布的每个新版本除增强了编码、调试的基础功能外，也与时俱进，引入了诸多顺应时代潮流的新功能。

时代不断发展，开发理念亦推陈出新，应用程序生命周期管理(Application Lifecycle Management, ALM)这一概念也得到巨大发展。ALM 关注应用程序的整个开发过程：从需求收集、架构设计、编码测试到交付部署，乃至项目管理。近年来一些炙手可热的技术，如测试驱动 TDD、领域设计 DDD、敏捷开发 Agile、Scrum 项目管理、持续集成和持续交付等概念无一不是 ALM 的创新和扩展。本人近几年一直从事的工作——生命周期评价(Life Cycle Assessment, LCA)——也有“生命周期”这四个字。这体现着一种理念，即对于应用程序的开发或产品的环境评价，都要着眼全局，通盘考虑。

在 Visual Studio 2005 发布前，微软只通过几个零散工具和功能来支撑 ALM 的各个部分，如通过 VSS 解决版本控制问题，通过 Visual Studio 2003 企业架构版解决架构建模问题。在开发 Visual Studio 2005 时，微软意识到，Visual Studio 应成为解决企业在整个开发项目生命周期遇到的各方面问题的一体化方案，并发布了 Visual Studio Team System 来实现端到端的 ALM 解决方案。Visual Studio Team System 将 Team Foundations Server 作为核心和服务器，为项目经理、需求分析人员、开发人员、测试人员、运维人员提供所需的功能。目前，Visual Studio 已能很好地解决开发团队在应用程序生命周期管理过程中遇到的一些困难：工具集成、地理分布的团队、角色细分、糟糕的报告、缺乏过程指导、对测试重视不够和沟通问题。

作为一本全面介绍 Visual Studio 应用程序生命周期管理方案的书籍，本书几乎涵盖了 ALM 的各个方面，涉及如何利用 Visual Studio 2012 进行需求获取、架构设计、编码开发、各类测试和项目管理。当然，由于篇幅所限，本书无法对所有主题都进行十分深入的剖析。还好，本书在恰当的地方都给出了丰富的参考书籍和资料。

本书全部章节由朱永光、吴慧峰翻译，参与翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。在翻译过程中，尽管我们尽了最大努力，但疏漏之处在所难免，望广大读者提出宝贵意见和建议。

作者简介



Mickey Gousset 是 Infront Consulting Group 公司(主要从事 Microsoft System Center 产品系列咨询业务)的高级咨询师。他已经连续 7 年荣获 Microsoft Application Lifecycle Management MVP, 同时是 *Professional Team Foundation Server*(Wiley, 2006)一书和 *Professional Application Lifecycle Management with Visual Studio 2010* (Wiley, 2010)一书的合著者。Mickey 维护着社区网站 ALM Rocks! (<http://www.almrocks.com>), 在这里撰写了很多关于 Visual Studio、TFS 和 ALM 的博客文章。他还与他人合办了广受欢迎的 Team Foundation Server 播客 Radio TFS (<http://www.radiotfs.com>)。他曾在世界各地发表有关 ALM 和 System Center 的演讲。Mickey 的业余爱好十分广泛, 包括玩 Xbox Liv(玩家标识为 HereBDragons)以及参加地方社团剧院的演出。有时也与爱妻 Amye 及两个吉娃娃 Lucy 和 Linus 依偎在沙发上共度闲暇时光。



Brian Keller 是 Microsoft 的高级技术布道师, 他的专业方向是 Visual Studio 和应用程序生命周期管理。Brian 自 2002 年起便与 Microsoft 结缘, 出席在世界各地举办的会议, 包括 TechEd、Professional Developers Conference(PDC)、Build 和 MIX。这是 Brian 为 Wiley 撰写的第三部著作。他也是 MSDN 的 Channel 9 网站的常客, 常与他人共同主持热门节目 This Week on Channel 9。Brian 在工作之余最喜欢户外运动, 如攀岩、背包旅行、滑雪和冲浪等, 或者和他的所爱 Elisa 共度美好时光。



Martin Woodward 目前担任 Microsoft Visual Studio Team Foundation Server 的资深研发经理, 专注于 Eclipse 和跨平台工具开发。在加盟 Microsoft 之前, Martin 就已被评选为年度 Visual Studio Application Lifecycle Management 的 MVP, 且多次在国际会议上发表有关 Team Foundation Server 的演讲。Martin 通过总结多年来在各种规模的公司中使用 Team Foundation Server 的实际经验, 对该产品的内部原理有了深刻独到的见解, 同时也乐于与大家交流心得。除了工作和演讲, 还可以在他的博客 <http://www.woodwardweb.com> 或播客 <http://radiotfs.com> 上找到 Martin。

致 谢

首先，我想感谢 Brian 和 Martin 再次和我一起共同完成本书的写作。你们为本书做出了难以置信的贡献，没有你们，本书不可能完成。我愿意未来任何时候和你们再次著书立说。还要感谢 Wiley 和 Wrox 中的每个人，特别是编辑 Bob Elliot 和 Tom Dinse。没有他们的帮助和对细节的持续关注，本书不会问世。Ed Blankenship 和 Steven St. Jean，谢谢你们卓越的技术编辑工作。你们让我时刻保持清醒。Gene 先生，谢谢你没有让我的草坪杂草丛生，以及 Kathy 夫人，你做的奶油朱古力饼是我从事写作的那些夜晚享用的美味佳肴。妈妈，谢谢你在写作过程中给予的全部支持和帮助。最后，最需要感谢的是 Amye、Emma 和 Meg，在深夜和周末我躲到办公室写作期间给予的理解、爱和支持。

——Mickey Gousset

首先，要再次感谢 Elisa 对我的信任和给予的鼓励。你让我们每天都精神百倍，在生命中有你的支持无比幸运。谢谢微软的开发部门构建出如此优秀的产品和技术，让我可以书写和教授这些内容——你们让我的工作更轻松。谢谢 Ed Blankenship、Steven St. Jean 和 Ravi Shanker 对本书提供的专家指导。当然也要谢谢 Mickey、Martin 以及 Wiley 的优秀团队——我依然无法相信，你们会继续邀请我撰写本书，不过感觉这是非常值得去做的事情。

——Brian Keller

首先感谢两位合著者 Brian 和 Mickey，让我继续在本书中和你们共同写作——让我感受到两位出色的工作，非常幸运能和两位成为朋友。

还要感谢来自微软 Visual Studio 团队内外朋友的帮助、建议和协助。要把特别的谢意送给 Jamie Cool、Buck Hodges、Grant Holliday、Matthew Mitrik、Ed Holloway、Peter Provost 和 William Bartholomew——没有你们，我无法完成本书中的内容。也要谢谢 Rob Caron、Jeff Beehler、Corey Steffen、Jamie Cool 和 Brian Harry 鼓励我在过去 8 年中参与到 Visual Studio 和 Eclipse 社区中。

最后感谢妻子 Catherine 给予的鼓励、支持，以及为了让我参与本书的写作而牺牲的无数夜晚和周末。否则对于如此知书达理的聪慧女子，很难理解最初为什么愿意嫁给我。就算经过 10 年，她依然如大学时代那样美丽，伴我左右，对我关怀备至。

——Martin Woodward

前言

在过去15年中，微软的软件开发工具已经成熟到不仅用来满足单个程序员的需要，也可以满足整个软件开发团队的需要。能满足包括业务分析师、项目经理、架构师、测试人员、程序员、经理、相关利益者的需要，甚至包括部署和维护应用程序的运维人员。撰写本书就是要帮助整个团队了解和采用这些工具达成终极目标——让他们更有凝聚力、提高生产效率，并最终按时按预算产出高质量的软件。

不管你是否已经拥有 Visual Studio 2012 或正考虑购买，本书都能帮助你为你的项目评估和采用适合的工具。本书会考虑组成现代软件开发项目的各种角色的需要。与每个角色相关的工具和技术都会详细讲解，甚至包含一些操作步骤来帮助你在自己的团队学习和应用每个工具。

0.1 本书读者对象

本书主要的对象是那些在业务或企业软件开发领域中的专业团队——换句话说，是针对中高级用户。如果你是如下任何一类用户，都可以从本书中获益匪浅：

- 期望学习如何使用 Visual Studio 2012 产品来帮助完成工作的开发人员、测试人员或架构师
- 必须管理软件开发项目的项目经理

本书包含的内容并非针对完全的初学者。其关注的是对工具、代码示例和实践场景的实际应用。本书的组织和编排十分合理，既是逐步学习的指南，也是在各种级别上建模、设计、测试和整合企业解决方案的参考书。

Visual Studio 2012 是为所有规模的软件开发团队而打造的。所以，不管是 5 人的团队，还是 2000 人的团队，本书包含的 Visual Studio 2012 和应用程序生命周期管理的信息都是有用的。不像大部分 Wrox 出版的书籍，本书面向的是软件开发组织中的所有角色——架构师、开发人员、测试人员、项目领导和管理者——而不仅仅针对开发人员。

0.2 本书内容

本书包含 Visual Studio 2012 应用程序生命周期管理功能的各个方面的内容。所以，本书基于这些方面分为 6 大部分：

- 第 I 部分：Team Foundation Server
- 第 II 部分：生成正确的软件

- 第III部分：项目管理
- 第IV部分：架构
- 第V部分：软件开发
- 第VI部分：测试

第 I 部分：Team Foundation Server

由于 Team Foundation Server 是微软应用程序生命周期管理解决方案的核心，因此本书从它所具备的功能开始研究。我们会讨论 Team Foundation Server 2012 的架构，接着探究版本控制系统和一些利用 Team Foundation Server 进行分支合并的最佳实践。最后，会深入钻研自动化构建过程——Team Foundation Build，并用一些示例来讲解如何对 Team Foundation Server 进行常见的自定义。

第 II 部分：生成正确的软件

微软的应用程序生命周期管理功能在 Visual Studio 2012 这个版本中获得扩展，考虑了软件开发过程中利益相关者所扮演的角色。利益相关者可能是应用程序的未来用户，影响应用程序购买的决策者，需要验证应用程序符合规章制度要求的律师，或开发团队以外的任何可能对特定开发项目的收入利益感兴趣的人。在该部分，你会了解到一些新工具，可让利益相关者更早更频繁地参与到整个开发过程中。这些工具可以促进开发出高质量的软件，在尽量减少返工的情况下既能满足需求又能持续地交付价值。

第III部分：项目管理

该部分会讲到 Visual Studio 2012 和 Team Foundation Server 2012 中用于处理项目和过程管理的功能。会研究与这个产品一同发布的过程模板，涵盖新的基于 Web 的敏捷计划和跟踪功能。第III部分也会讨论内置在 Team Foundation Server 中的报表功能。不管你是使用注入 Scrum 这样的轻量级开发方法学，还是更正式严格的开发过程，都会发现 Team Foundation Server 为你提供了管理项目所需的工具。

第IV部分：架构

该部分会研究 Visual Studio 2012 中为定义和分析应用程序架构而提供的一些工具。在简要介绍架构概念后，会深入讨论新的 UML 工具，包括用例图、活动图、序列图、类图和组件图。你接着还会学到 Architecture Explorer，以及如何使用它来了解应用程序的架构。该部分最后讨论分层图(layer diagrams)。

第 V 部分：软件开发

该部分会涉及使用 Visual Studio 2012 的软件开发人员最感兴趣的主体。这些主题的选择最适合通过团队的力量来构建复杂应用程序的情况。例如，该部分会解释如何进行单元测试、静态代码分析、调优、代码覆盖率分析和新的代码克隆分析特性，以便提高应用程序的整体质量和可维护性。第 V 部分也会介绍内置的全新代码审查功能，讲解如何使用它来和其他开发人员协作。你会了解到这个功能如何在开发过程中挂起和恢复，让工作被打

断的情况处理起来更加容易。最后，该部分还会深入研究利用 IntelliTrace 来调试应用程序，甚至会提到使用 IntelliTrace 的一种新方式在生产环境中调试应用程序。

第VI部分：测试

Visual Studio 2012为测试人员提供了大量工具供其所用。我们会从Microsoft Test Manager 中提供的手动测试功能开始研究，并研究如何利用可编码用户界面测试所提供的自动化用户界面测试能力。Web性能测试和负载测试能让你创建测试来帮助确保你的网站用户会体验到最佳性能，即使在繁重负载下也是如此。该部分的最后讲解Visual Studio 2012针对实验室管理功能的加强，这种增强让你可以利用物理或虚拟的环境来使用自动化的“生成-部署-测试”工作流。

Team Foundation Server 管理员

如果你是团队中负责管理 Team Foundation Server 部署的那个人，那么你应该考虑购买本书的姊妹书——《Team Foundation Server 2012 高级教程》——这本书会深入研究 Team Foundation Server 2012 的安装、配置和管理。你可以从中找到更多关于部署拓扑结构的内容，学到如何修改过程模板、进行高级安全设置、考虑灾难恢复和地理分布团队的需求，还可以了解到更多额外的管理配置内容。

0.3 本书约定

为了帮助你更好地阅读其中的文字，跟上内容讲述的节奏。我们会在整本书中使用一些约定。



类似这样的带有警告图标的方框，会包含与周围文字直接相关的、重要的、不应该忽视的信息。



类似这样缩进且斜体的文字表明这是一个注释、小窍门、提示、小技巧。

旁白

类似这样缩进的文字是针对当前讨论内容的旁白。

我们以两种不同的方式来表示代码：

- 对于大部分代码示例，使用单一的无高亮的文本方式。
- 在特别重要需要表示上下文的地方，我们用粗体来强调代码。

0.4 源代码

读者在学习本书中的示例时，既可以手动输入所有代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从站点 <http://www.wrox.com/> 或 www.tupwk.com.cn/downpage 下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有源代码。



由于许多图书的书名都很类似，所以按 ISBN 进行搜索是最简单的，本书英文版的 ISBN 是 978-1-118-31408-1。

在下载了代码后，只需要用自己喜欢的解压缩软件对它们进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载页面，查看本书和其他 Wrox 图书的源代码。

0.5 勘误表

尽管我们已经尽了最大的努力来保证文章或代码中不出现错误，但是错误总是难免的，如果你在本书中找到了错误，例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走入误区，当然，这还有助于提供更高质量的信息。

要在网站上找到本书英文版的勘误表，可以登录 www.wrox.com，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Errata 链接。在这个页面上可以查看到 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，网址是 www.wrox.com/misc-pages/booklist.shtml。

如果你在勘误表上没有找到错误，那么可以到 www.wrox.com/contact/techsupport.shtml 上，完成上面的表格，并把找到的错误发送给我们。我们将会核查这些信息，如果无误的话，会把它放置到本书的勘误表中，并在本书的后续版本中更正这些问题。

也可以给 wkservice@vip.163.com 发电子邮件，我们会检查你的信息，如果是正确的，我们将在本书后续版本中采用。

0.6 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，便于你张贴与 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，它可以给你传送感兴趣的论题。Wrox 作者、编辑和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，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (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, 并单击 **Submit** 按钮。
- (4) 你会收到一封电子邮件, 其中的信息描述了如何验证账户和完成加入过程。

不加入 P2P 也可以阅读论坛上的消息, 但要张贴自己的消息, 就必须加入该论坛。

加入论坛后, 就可以张贴新消息, 回复其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要想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的解答。要阅读 FAQ, 可在任意 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 Visual Studio 2012 应用

生命周期管理介绍	1
1.1 应用生命周期管理	2
1.2 Visual Studio 2012 产品系列	2
1.3 应用生命周期管理挑战	4
1.4 进入 Visual Studio 2012 的世界	5
1.5 应用生命周期管理活动简介	6
1.5.1 需求	6
1.5.2 系统设计与建模	6
1.5.3 代码生成	6
1.5.4 测试	7
1.5.5 反馈	7
1.5.6 运营	7
1.5.7 一切尽在本书中	8
1.6 小结	8

第 I 部分 Team Foundation Server

第 2 章 Team Foundation

Server 简介	11
2.1 Team Foundation Server 简介	11
2.2 获取 Team Foundation Server	12
2.2.1 托管 Team Foundation Server	13
2.2.2 独立安装	14
2.3 Team Foundation Server 核心概念	15
2.3.1 Team Foundation Server	15
2.3.2 团队项目集合	16
2.3.3 团队项目	16
2.3.4 团队	19

2.3.5 过程模板	20
2.3.6 工作项跟踪	21
2.3.7 版本控制	22
2.3.8 团队生成	24
2.4 访问 Team Foundation Server	25
2.4.1 从 Visual Studio 访问 Team Foundation Server	25
2.4.2 通过 Web 浏览器访问 Team Foundation Server	27
2.4.3 在微软 Excel 中使用 Team Foundation Server	28
2.4.4 在微软 Project 中使用 Team Foundation Server	29
2.4.5 Team Foundation Server 命令行工具	29
2.4.6 从 Eclipse 访问 Team Foundation Server	29
2.4.7 Team Foundation Server 与 Windows Explorer 集成	30
2.4.8 通过其他第三方集成 工具访问 Team Foundation Server	30
2.5 Team Foundation Server 2012 中的新内容	31
2.5.1 版本控制	31
2.5.2 Web 访问	32
2.5.3 Team Explorer	32
2.5.4 团队	32
2.5.5 工作项跟踪	32
2.5.6 云平台	33
2.6 采用 Team Foundation Server	33
2.7 小结	34

第 3 章 Team Foundation 版本

控制 35

3.1 Team Foundation 版本 控制和 VSS 2005	37
3.2 设置版本控制	37
3.3 Source Control Explorer	38
3.3.1 设置工作区	39
3.3.2 获取代码	40
3.3.3 在版本控制中共享项目	41
3.4 签入挂起变更	44
3.4.1 签入一项	44
3.4.2 创建和管理签入策略	48
3.4.3 查看历史	50
3.4.4 标记文件	51
3.5 搁置	52
3.5.1 工作区	53
3.5.2 服务器工作区	56
3.5.3 本地工作区	56
3.6 命令行工具	57
3.7 小结	58

第 4 章 分支与合并 59

4.1 揭秘分支	59
4.1.1 分支	60
4.1.2 合并	60
4.1.3 冲突	60
4.1.4 分支关系	61
4.1.5 无基合并	61
4.1.6 正向/逆向集成	62
4.2 常见的分支策略	62
4.2.1 不分支	62
4.2.2 每个发布都分支	63
4.2.3 代码升级分支	64
4.2.4 功能分支	65
4.3 实施分支策略	66
4.3.1 场景	66
4.3.2 方案	67
4.3.3 实施	68
4.3.4 处理变更集	73

4.3.5 通过分支跟踪变更	83
----------------------	----

4.4 小结	84
--------------	----

第 5 章 Team Foundation Build 85

5.1 Team Foundation Build 简介	86
5.2 Team Foundation Build 2012 的新功能	87
5.2.1 托管生成服务	88
5.2.2 输出到版本控制	88
5.2.3 批处理限制门签入	89
5.2.4 生成服务协议的变化	90
5.2.5 更新的生成报告	90
5.3 Team Foundation Build 架构	90
5.4 生成管理	92
5.4.1 Team Explorer	92
5.4.2 Build Explorer	92
5.4.3 生成细节视图	93
5.4.4 创建生成定义	94
5.4.5 生成排队	101
5.4.6 生成通知	103
5.5 团队生成过程	105
5.5.1 DefaultTemplate 过程	105
5.5.2 生成过程的参数	107
5.6 小结	112

第 6 章 Team Foundation Server

常用自定义 113

6.1 对象模型	114
6.1.1 客户对象模型	115
6.1.2 服务器对象模型	115
6.1.3 生成过程对象模型	115
6.1.4 简单的对象模型示例	115
6.1.5 用于 Java 的 TFS SDK	117
6.2 自定义 Team Foundation Build	117
6.2.1 创建自定义生成过程模板	117
6.2.2 创建自定义生成 工作流活动	119
6.3 自定义 Team Foundation 版本控制	119

6.4 Team Foundation Server 事件服务	120	10.1.1 丰富的工作项关系	162
6.5 自定义工作项跟踪	121	10.1.2 敏捷计划工具	163
6.5.1 修改工作项类型定义	121	10.1.3 测试用例管理	164
6.5.2 创建自定义工作项控件	122	10.1.4 反馈管理	164
6.6 小结	123	10.1.5 增强报表	164
		10.1.6 SharePoint Server 仪表板	165
第 II 部分 生成正确的软件		10.2 工作项	165
第 7 章 生成正确的软件简介	127	10.2.1 工作项类型	165
7.1 利益相关者	128	10.2.2 区域和迭代	168
7.2 故事板	129	10.3 过程模板	170
7.3 捕获利益相关者的反馈	130	10.3.1 MSF for Agile Software Development	170
7.4 单独的工作项视图	131	10.3.2 MSF for CMMI Process Improvement	172
7.5 第三方需求管理解决方案	132	10.3.3 Visual Studio Scrum	175
7.5.1 TeamSpec	132	10.3.4 第三方过程模板	176
7.5.2 TeamLook	133	10.3.5 自定义过程模板	176
7.5.3 inteGREAT	134	10.4 管理工作项	176
7.6 小结	135	10.4.1 使用 Visual Studio	177
第 8 章 故事板	137	10.4.2 使用 Microsoft Excel	181
8.1 使用故事板的原因	137	10.4.3 使用 Microsoft Project	182
8.2 PowerPoint Storyboarding	139	10.4.4 使用 Team Web Access	183
8.2.1 故事板形状	140	10.5 Project Server 集成	184
8.2.2 版式	141	10.6 小结	185
8.2.3 屏幕截图	142		
8.2.4 My Shapes 分类	146	第 11 章 敏捷计划和跟踪	187
8.2.5 动画	147	11.1 定义团队	188
8.2.6 超链接	148	11.2 维护产品积压	192
8.2.7 故事板链接	148	11.3 计划迭代	194
8.3 小结	150	11.4 跟踪工作	196
第 9 章 获取利益相关者的反馈	151	11.5 自定义选项	199
9.1 请求反馈	152	11.6 小结	199
9.2 小结	157		
		第 12 章 报表、门户和面板	201
第 III 部分 项目管理		12.1 Team Foundation Server 报表功能	202
第 10 章 项目管理简介	161	12.2 使用 Team Foundation Server 报表	204
10.1 Team Foundation Server 2012 中的项目管理增强	162	12.2.1 创建报表的工具	204

12.2.2	使用 Microsoft Excel 报表.....	205	14.2.3	将活动图添加 到用例图.....	238
12.2.3	使用 RDL 报表.....	214	14.3	顺序图.....	238
12.3	小结.....	215	14.3.1	创建顺序图.....	238
第IV部分 架构					
第 13 章	软件架构介绍.....	219	14.3.2	顺序图工具箱.....	241
13.1	设计可视化.....	219	14.4	组件图.....	241
13.2	Microsoft 的建模策略.....	221	14.4.1	创建组件图.....	242
13.2.1	了解模型驱动开发.....	221	14.4.2	组件图工具箱.....	246
13.2.2	了解领域特定语言.....	222	14.5	类图.....	247
13.2.3	“理解代码”体验.....	222	14.5.1	创建类图.....	247
13.3	Visual Studio Ultimate 2012 提供的架构工具.....	223	14.5.2	类图工具箱.....	249
13.3.1	用例图.....	223	14.5.3	从UML类图生成代码.....	250
13.3.2	活动图.....	224	14.6	小结.....	251
13.3.3	顺序图.....	225	第 15 章	使用 Architecture Explorer 和依赖图 分析应用程序.....	253
13.3.4	组件图.....	225	15.1	了解基本代码.....	254
13.3.5	类图.....	226	15.2	Architecture Explorer 基础.....	255
13.3.6	层次图.....	226	15.2.1	了解 Architecture Explorer 窗口.....	255
13.3.7	Architecture Explorer.....	227	15.2.2	Architecture Explorer 选项.....	256
13.4	Visual Studio Ultimate 2012 中的新架构工具.....	227	15.2.3	Architecture Explorer 的导航功能.....	257
13.4.1	架构菜单选项.....	227	15.2.4	命名空间的浏览选项.....	258
13.4.2	依赖图增强.....	228	15.2.5	类的浏览选项.....	260
13.4.3	Visual Studio 可视化 和建模 SDK.....	228	15.2.6	成员的浏览选项.....	261
13.5	小结.....	228	15.3	依赖图.....	262
第 14 章	使用用例图、活动图、 顺序图、组件图和类图 进行自上而下的设计.....	229	15.3.1	创建第一个依赖图.....	263
14.1	用例图.....	230	15.3.2	不用 Architecture Explorer 创建依赖图.....	264
14.1.1	创建用例图.....	230	15.3.3	依赖图的导航功能.....	265
14.1.2	用例图工具箱.....	233	15.3.4	依赖图图例.....	268
14.2	活动图.....	234	15.3.5	依赖图工具栏.....	269
14.2.1	创建活动图.....	234	15.3.6	Visual Studio 2012 中 对依赖图所做的增强.....	269
14.2.2	活动图工具箱.....	237	15.4	小结.....	271

第 16 章	使用层次图建模和实施	
	应用程序架构	273
16.1	创建层次图	274
16.2	层次图的层定义	275
16.2.1	为单独的工件项	
	创建层	276
16.2.2	在层次图中添加	
	多个对象	276
16.2.3	Layer Explorer	277
16.3	定义依赖关系	278
16.4	验证层次图	280
16.5	层次图和生成过程	281
16.6	小结	282

第 V 部分 软件开发

第 17 章	软件开发简介	285
17.1	Visual Studio 2012 为开发	
	人员提供的新功能	286
17.1.1	单元测试	286
17.1.2	改进的代码分析功能	287
17.1.3	代码度量	287
17.1.4	代码克隆分析	287
17.1.5	性能分析器	287
17.1.6	IntelliTrace 的高级	
	调试功能	288
17.2	My Work	288
17.2.1	暂停和恢复	289
17.2.2	代码评审	289
17.3	小结	292
第 18 章	单元测试	293
18.1	单元测试的基本概念	294
18.1.1	单元测试的优点	294
18.1.2	编写有效的单元测试	295
18.1.3	第三方工具	296
18.2	Visual Studio 单元测试	296
18.2.1	创建第一个单元测试	297
18.2.2	管理和运行单元测试	299

18.2.3	调试单元测试	300
18.3	使用单元测试框架	
	进行编程	301
18.3.1	单元测试的初始化	
	和清除	301
18.3.2	使用 Assert 方法	304
18.3.3	使用 CollectionAssert 类	307
18.3.4	使用 StringAssert 类	308
18.3.5	期望的异常	309
18.3.6	定义自定义单元	
	测试属性	309
18.3.7	TestContext 类	310
18.4	Microsoft Fakes 简介	311
18.4.1	存根与填充码的选择	311
18.4.2	使用存根	312
18.4.3	使用填充码	314
18.5	测试适配器	316
18.6	小结	317

第 19 章	代码分析、代码度量和	
	代码克隆分析	319
19.1	分析工具的必要性	320
19.2	使用代码分析	320
19.2.1	内置的代码分析规则	321
19.2.2	代码分析规则集	322
19.2.3	启用代码分析	323
19.2.4	执行静态代码分析	325
19.2.5	违反规则的处理	327
19.3	使用命令行分析工具	330
19.3.1	FxCopCmd 选项	330
19.3.2	FxCopCmd 项目文件	332
19.3.3	将代码分析集成在	
	生成过程中	333
19.4	创建代码分析规则	334
19.5	代码度量	334
19.6	代码克隆分析	336
19.6.1	找出代码克隆	336
19.6.2	检查代码克隆分析	
	的结果	337

19.6.3	代码克隆分析的 工作原理	337
19.6.4	从代码克隆分析中 排除项	338
19.7	小结	339
第 20 章	性能和性能分析	341
20.1	性能分析概述	342
20.1.1	性能分析器类型	342
20.1.2	Visual Studio性能分析	343
20.2	使用性能分析器	343
20.2.1	创建示例应用程序	343
20.2.2	创建性能会话	345
20.2.3	使用 Performance Explorer	348
20.2.4	配置采样式会话	356
20.2.5	配置插装式会话	357
20.2.6	配置.NET 内存 分配会话	357
20.2.7	配置并发分析会话	358
20.2.8	执行性能会话	358
20.2.9	管理会话报告	359
20.2.10	解读会话报告	360
20.3	命令行分析实用工具	368
20.3.1	JavaScript 性能分析	369
20.3.2	仅分析自己的代码	370
20.4	常见的性能分析问题	371
20.4.1	调试符号	371
20.4.2	插装和代码覆盖	371
20.5	小结	371
第 21 章	使用IntelliTrace进行调试	373
21.1	IntelliTrace 的基本功能	373
21.1.1	IntelliTrace 示例	374
21.1.2	导航 IntelliTrace 事件视图	376
21.1.3	收集方法调用信息	376
21.1.4	收集详细信息	379
21.1.5	保存 IntelliTrace 会话	380
21.1.6	IntelliTrace 选项	384

21.2	IntelliTrace 在生产环境 中的应用	386
21.2.1	安装 IntelliTrace 独立 收集器	386
21.2.2	配置IntelliTrace PowerShell commandlet	388
21.2.3	收集执行信息	388
21.3	小结	389

第VI部分 测试

第 22 章	软件测试简介	393
22.1	基于角色的测试工具	394
22.2	测试类型	394
22.3	诊断数据适配器	395
22.4	Microsoft Test Manager	397
22.5	使用 Visual Studio 管理 自动测试	398
22.5.1	测试项目类型	399
22.5.2	Test Explorer	400
22.5.3	代码覆盖	402
22.5.4	使用顺序测试	402
22.5.5	测试设置	404
22.6	小结	405
第 23 章	手动测试	407
23.1	Microsoft Test Manager	407
23.2	使用测试计划	409
23.2.1	配置测试设置	410
23.2.2	版本	412
23.2.3	测试影响分析	413
23.2.4	测试配置定义	414
23.2.5	使用计划内容	415
23.3	测试运行和结果记录	420
23.3.1	使用 Test Runner	422
23.3.2	动作记录的支持技术	425
23.3.3	记录 bug 并保存 测试结果	426
23.4	探索式测试	427
23.5	运行自动测试	430