

<<.NET单元测试艺术>>

图书基本信息

书名：<<.NET单元测试艺术>>

13位ISBN编号：9787302269168

10位ISBN编号：7302269165

出版时间：2012-1

出版时间：清华大学出版社

作者：奥西洛夫

页数：297

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<.NET单元测试艺术>>

内容概要

《.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。

<<.NET单元测试艺术>>

作者简介

作者：(以色列)奥西洛夫(Roy Osherove) 译者：张博超 张昌贵 李丁山 注释 解说词：滕振宇奥西洛夫，Roy Osherove，Typemock首席架构师，ALT.NET创办人。
在全球各地主要从事单元测试和测试驱动开发的顾问和培训工作。
他也是TechEd和JAOO等国际性技术大会的明星发言人。

<<.NET单元测试艺术>>

书籍目录

《.net单元测试艺术》

第i部分 入门

第1章 单元测试的基本知识

1.1 单元测试——传统定义

1.1.1 编写“优秀单元测试”的重要性

1.1.2 我们都写过单元测试(或多或少)

1.2 优秀单元测试的特性

1.3 集成测试

1.4 优秀的单元测试——定义

1.5 一个简单的单元测试实例

1.6 测试驱动开发

1.7 小结

第2章 第一个单元测试

2.1 单元测试框架

2.1.1 单元测试框架的优势提供了什么

2.1.2 xunit测试框架

2.2 logan项目的介绍

2.3 使用nunit的第一步

2.3.1 安装nunit

2.3.2 加载解决方案

2.3.3 在代码中使用nunit特性

2.4 编写第一个测试

2.4.1 assert类

2.4.2 用nunit运行我们的第一个测试

2.4.3 修正代码让测试通过

2.4.4 从红色到绿色

2.5 更多nunit特性

2.5.1 setup和teardown

2.5.2 验证预期的异常

2.5.3 忽略测试

2.5.4 设置测试类别

2.6 针对状态的间接测试

2.7 小结

第ii部分 核心技术

第3章 使用桩对象解除依赖

3.1 桩对象

3.2 发现logan对文件系统的依赖

3.3 确认简化loganalyzer测试的方法

3.4 重构设计增强了可测性

3.4.1 抽取接口, 以允许替换底层实现

3.4.2 在被测类中注入桩对象

3.4.3 在构造函数级别上接收一个接口(构造函数注入)

3.4.4 接收一个接口作为属性的get或set的类型

3.4.5 在调用方法之前获取一个桩对象

3.5 重构技术的变种

<<.NET单元测试艺术>>

3.6 解决封装问题

3.6.1 使用internal和[internalvisibleto]

3.6.2 利用[conditional]属性标签

3.6.3 使用#if和#endif的条件编译

3.7 小结

第4章 用模拟对象做交互测试

4.1 基于状态的测试和交互测试

4.2 模拟对象和桩对象之间的区别

4.3 简单的手写模拟对象例子

4.4 同时使用模拟对象和桩对象

4.5 一个测试一个模拟对象

4.6 桩链：产生模拟对象或其他桩的一批桩对象

4.7 手写模拟对象和桩对象的问题

4.8 小结

第5章 隔离(模拟对象)框架

5.1 为什么使用隔离框架

5.2 动态创建伪对象

5.2.1 在测试中引入rhino mocks

5.2.2 使用动态模拟对象替换手写模拟对象

5.3 严格模拟对象与非严格模拟对象

5.3.1 严格模拟对象

5.3.2 非严格模拟对象

5.4 从伪对象返回值

5.5 用隔离框架创建智能桩对象

5.5.1 在rhino mocks框架中创建桩对象

5.5.2 结合使用动态桩对象和模拟对象

5.6 模拟对象和桩对象的参数约束

5.6.1 用字符串约束检查参数

5.6.2 使用约束检验参数对象的属性

5.6.3 执行回调检验参数

5.7 测试与事件相关的活动

5.7.1 测试一个事件已被订阅

5.7.2 在模拟对象和桩对象中触发事件

5.7.3 测试一个事件是否被触发

5.8 隔离框架中的设置-操作-断言语法

5.9 .net中现有的隔离框架

5.9.1 nunit.mocks

5.9.2 nmock

5.9.3 nmock2

5.9.4 typemock isolator

5.9.5 rhino mocks

5.9.6 moq框架

5.10 隔离框架的优势

5.11 避免使用隔离框架时的陷阱

5.11.1 测试代码缺乏可读性

5.11.2 对错误的事情做验证

5.11.3 一个测试包含多个模拟对象

<<.NET单元测试艺术>>

5.11.4 测试的细节太多

5.12 小结

第iii部分 测试的代码

第6章 测试层次及组织

6.1 让自动化构建运行自动化测试

6.1.1 自动构建剖析

6.1.2 触发构建和持续集成

6.1.3 自动化构建类型

6.2 根据速度和类型组织测试

6.2.1 分离单元测试与集成测试的人为因素

6.2.2 绿色安全区域

6.3 确保测试在代码库中

6.4 在测试类和被测代码之间建立映射

6.4.1 映射测试到项目

6.4.2 映射测试到类

6.4.3 映射测试到方法

6.5 为应用程序打造测试api

6.5.1 使用测试类的继承模式

6.5.2 新建测试工具类和方法

6.5.3 让程序员知道你的api

6.6 小结

第7章 优秀单元测试的支柱

7.1 编写可信赖的测试

7.1.1 决定何时删除或更改测试

7.1.2 避免测试的逻辑

7.1.3 只测试一件事情

7.1.4 让测试容易运行

7.1.5 确保测试覆盖率

7.2 编写可维护的测试

7.2.1 测试私有的或者受保护的方法

7.2.2 去除重复代码

7.2.3 让setup方法可维护

7.2.4 实施测试隔离

7.2.5 避免多个断言

7.2.6 避免测试同一个对象的多个方面

7.2.7 避免在测试里过度关注细节

7.3 编写可读的测试

7.3.1 为单元测试命名

7.3.2 为变量命名

7.3.3 让断言有意义

7.3.4 将断言和动作分离

7.3.5 setup和teardown

7.4 小结

第iv部分 设计与流程

第8章 在组织中引入单元测试

8.1 怎样成为变革推动者

8.1.1 备战棘手问题

<<.NET单元测试艺术>>

- 8.1.2 说服内部人士：拥护者与阻碍者
- 8.1.3 洞察切入机会
- 8.2 成功之路
 - 8.2.1 游击策略(自下而上)
 - 8.2.2 说服管理层(自上而下)
 - 8.2.3 从外面找一个专家
 - 8.2.4 让过程可见
 - 8.2.5 锁定目标
 - 8.2.6 意识到即将面对的阻碍
- 8.3 失败之路
 - 8.3.1 缺乏驱动力
 - 8.3.2 缺乏政治上的支持
 - 8.3.3 不好的实施和第一印象
 - 8.3.4 缺乏团队支持
- 8.4 棘手的问题及其答案
 - 8.4.1 在现有的流程上会增加多少时间
 - 8.4.2 测试人员的工作会因此受到威胁吗
 - 8.4.3 怎么知道这确实可行呢
 - 8.4.4 有什么可以证明单元测试的好处
 - 8.4.5 为什么测试部门还是能找到缺陷
 - 8.4.6 我们有很多没有测试的代码：该从哪里开始呢
 - 8.4.7 使用多种语言开发：单元测试适用吗
 - 8.4.8 如果是软硬件结合的开发，该怎么办
 - 8.4.9 怎么知道测试本身是否有缺陷
 - 8.4.10 我的调试器显示代码可以正常工作：为什么还需要测试
 - 8.4.11 必须用tdd的方式来编码吗
- 8.5 小结
- 第9章 修改遗留代码
 - 9.1 从哪里开始添加测试？
 - 9.2 确定抉择策略
 - 9.2.1 容易优先策略的优缺点
 - 9.2.2 困难优先策略的优缺点
 - 9.3 在重构前写集成测试
 - 9.4 重要的遗留代码单元测试工具
 - 9.4.1 使用typemock isolator轻松隔离依赖项
 - 9.4.2 使用depender找出可测性问题
 - 9.4.3 在java遗留代码里使用jmockit
 - 9.4.4 重构java代码时使用vise
 - 9.4.5 使用fitnesse在重构前做验收测试
 - 9.4.6 阅读michael feathers的关于遗留代码的书
 - 9.4.7 使用ndepend来审查生产代码
 - 9.4.8 使用resharper浏览和重构生产代码
 - 9.4.9 使用simian来检测重复代码(和缺陷)
 - 9.4.10 使用typemock racer来检测线程问题
 - 9.5 小结
 - 附录a 设计与可测试性

<<.NET单元测试艺术>>

附录b 工具和框架

<<.NET单元测试艺术>>

章节摘录

版权页：插图：开始推行变革时，通过调查辨别出你认为对自己有帮助的人。

他们会是你的拥护者。

通常他们是早期采用者，或者是那些愿意接受新事物、愿意尝试你倡议的人。他们可能半数已经被说服，但正在寻找开始变革的推动力。

他们甚至可能已经尝试过但是失败了。

赶在别人之前接近这些人，并针对你要做的事情向他们请教。

他们可能会告诉你一些你未曾考虑过的事情：可以从哪些团队开始，哪里的人更容易接受这项变革。

甚至他们从自己的经验角度出发，告诉你需要注意哪些事项。

通过接近这些人，有助于确保他们成为变革过程的参与者。

觉得自己是参与者，通常会努力帮助实现过程。

让他们成为你的拥护者；问他们是否可以帮助你，是否可以成为回答人们问题的人。

让他们为此做好准备。

阻碍者接下来，辨别出阻碍者。

这些人很可能在组织里抵制你倡导的变革。

例如，某个管理人员可能反对添加单元测试，宣称单元测试会增加太多开发时间，并增加需要维护的代码量。

通过在过程中给这些人（至少可以是那些有意愿并且有能力的人）一个正面角色，让他们成为过程的参与者，而不是过程的抵制者。

<<.NET单元测试艺术>>

媒体关注与评论

“这是我印象最深的TDD（测试驱动开发）最佳图书。
它从最基础的讲起，通过严谨、客观的方式自然而然地过渡到高级主题。

” ——Robert C.Martin (B.

b大叔)，《敏捷软件开发》作者“若干年前就应该面世的重要图书。” ——Michael Feathers，《修改遗留代码的艺术》作者“《.NET单元测试艺术》讲透了单元测试的方方面面，甚至论及单元测试的诟病。

” ——Franco Lombardo，Molteni InformatiCa“ Roy深谙测试之道。

” ——Wendy Friedlander，Agile Solutions“ 涵括单元测试的理论与实践。

” ——Francesco Goggi，软件工程师“ 精心打造的大师之作，凝聚着单元测试的精髓。

Bravo，Bravo，BraVo！

” ——Mohammad Azam，微软最有价值专家（Microsoft MVP），HighOnCoding“ 出神入化！

” ——Gabor Paller，OnRelay公司“ 以前针对单元测试的书都是写给外行看热闹的，但这本书是写给专业人员阅读和参考的。

” ——Josh Cronemayer，ThoughWorks“ 您会发现，这本书包含单元测试方方面面的内容。

我爱看涉及测试代码组织与重构这一章。

这是帮助您的团队/公司拥抱单元测试的理想图书，可谓字字珠玑。

” ——Alessandro Gallo，ASP.NET Ajax in Action作者“ 初学者可以很容易从这本书中学会开始写单元测试，高手也可以学到提高技能的一些思路和方法。

” ——Kent Beck

<<.NET单元测试艺术>>

编辑推荐

《.NET单元测试艺术》编辑推荐：以正确的方式进行单元测试，是决定项目成败的关键，是决定代码维护性强弱的分水岭，是决定你加班到深夜两点还是准点下班回家晚餐的重要因素。

<<.NET单元测试艺术>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>