

<<测试驱动的iOS开发>>

图书基本信息

书名：<<测试驱动的iOS开发>>

13位ISBN编号：9787111399193

10位ISBN编号：7111399196

出版时间：2012-10

出版时间：机械工业出版社

作者：Graham Lee

页数：206

译者：爱飞翔

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>

<<测试驱动的iOS开发>>

前言

我向其他开发者讲授测试驱动开发很大程度上是出于偶然。

本来安排我在会议上关于另一个话题发表演讲，而我的一个朋友则讲述TDD（Test Driven Development，测试驱动开发）。

不过朋友的妻子选择那周末生双胞胎（我认为是这样的，我可不是这方面的专家），所以Chunk，也就是委托我写这本书的人，问我能不能也关于TDD发表演讲。

由此机缘，最终使我开始了为期一年的写书过程。

通常来说，真实状况并不像我们彼此之间讲述的那样美好。

实际上，我初次接触单元测试是在很多年前了。

在成为专业的软件开发者之前，我曾在一家制作基于GNUstep（由自由软件基金会所制作，可运行于Linux与其他平台的Cocoa库）产品的公司里当测试人员。

据我当时所知，单元测试就是确保软件产品的每个小部件都能正常运行的一种开发方法。

当这些小部件合并为大的组件时，按理说它也能正常运行才是。

这种理解一直持续到了我从事第一份编程工作时，那个工作是负责某个跨平台安全软件的Mac系统版本开发。

（这里我又一次省略了一件事——几年前，我曾做过一个为期6周的LISP外包程序。

每个人都曾做过一些并不为之骄傲的事情。

）在做这份工作时，我参加了一个TDD培训课程，这项课程是由经常活跃于面向对象编程讨论会上的Kevlin Henney所开设的。

他除了写过很多文章之外，还曾编辑过一本书，名叫《每个程序员都应该知道的97件事》。

就是通过学习这个课程，我才最终意识到测试驱动开发的主旨是使得开发者对其编写的代码更有信心。

随着学习的深入，我还领悟到测试驱动开发可以使开发者在修改代码时心里更加踏实。

当对TDD有足够的了解，并能够从实践的教训中总结经验时，我终于将TDD当成了日常开发的一部分，并弄懂了其中哪些方法适合我，哪些不适合。

几年之后，我能够接受Chuck的邀请，来做一场关于TDD的演讲了。

真心希望这本书能够让读者领悟到测试驱动开发的好处，并将其纳入日常编码工作中，同时也希望读者能稍花点时间就掌握它，不要像笔者这样花费了5年左右的时间。

有很多关于单元测试的书，其中某些书的作者也曾参与测试框架的编写与设计。

这些书都很好，不过，它们都没有特别针对Cocoa Touch的开发者。

笔者的这本书提供了以Objective-C语言所编写的范例，讲述了Xcode及相关工具的用法，并以Cocoa的编程风格书写代码。

但愿本书能够将测试驱动开发的原理讲得更加易懂，并使它与iOS开发者的工作联系起来。

另外，还有测试工具的问题。

对于写单元测试有很多种工具可用，具体采用哪个，还要取决于众多不同的工具与框架所提供的功能。

尽管本书会提到它们之间的一些差别，但是笔者打算专注于讲解Apple所提供Xcode开发环境及其所附OCUnit测试框架的用法。

这么做的原因很简单，那就是适用性：任何想尝试单元测试或TDD的开发者，只要学习了本书所讲的知识，使用标准的开发工具，并具备一定的决心，就可以立刻投入工作中。

如果读者觉得标准的测试工具缺少某些功能或者不太好用，那你当然可以研究其他工具的用法，甚至自己写一个测试工具——只是记得要测试它哦！

在成长为一个迷恋测试的程序员的过程中，我学到了很多。

其中之一就是，想要当一个好的软件工程师，最好的办法就是与其他开发者交流。

如果你对本书内容或者通用的TDD开发方法有任何评论或建议，欢迎在Twitter上与我交流讨论（我的用户名是iamleeg）。

<<测试驱动的iOS开发>>

致谢牛顿曾说过，“如果我比别人看得更远，那是因为我站在巨人的肩上”，当然了，他的这个说法综合了数个世纪以来作者们不断拓展与完善的那个隐喻。

与此相似，本书也不是凭空写出来的，有很多要感谢的人，若全部写出他们的名字的话，要从爱达勒芙蕾丝伯爵夫人开始，用好多页才能写完。

如果要写一份简洁一些的致谢表，那么首先要感谢的就是培生（Pearson）出版集团的诸位同仁，是他们促成了本书的出版发行。

还要感谢Chunk、Trina与Olivia，是他们持续督促我完成此书的。

此外还有本书的技术审校者Saul、Tim、Alan、Andrew、两位名为Richard的先生、Simon、Patrick以及Alexander，感谢你们在排查本书草稿中错误的过程中所做的出色工作，如果还有错误的话，则应归咎于我。

感谢Andy与Barbara，你们所做的文字润色工作，将一些由程序员草草写就的文字变成了优雅的英语文句。

感谢xUnit测试框架的设计者Kent Beck先生，没有他对于测试的见解，我将无处下笔，同样，也感谢xUnit框架的Objective-C版本制作方Sente SA公司。

还必须提到的是Apple的开发工具制作团队，他们为了让全世界的iOS开发者能够使用单元测试所做的努力比其他人都要多。

与别人相比，Kevlin Henney更让我领略到了测试驱动开发之美，让我避免了很多bug，谢谢你！

最后，感谢Freya在没日没夜的写书过程中对我的支持与理解。

如果此刻你正在读这段话，我想你可能会更加了解我的。

<<测试驱动的iOS开发>>

内容概要

本书是目前仅有的一本讲解如何将测试驱动的开发方法应用到iOS开发中的权威教程，从测试工具、测试驱动开发方法和技巧、基于测试驱动开发方法的软件设计等多角度完美地展现了测试驱动开发方法与iOS开发的结合，不仅能使iOS开发者迅速掌握测试驱动的开发方法，还能使他们在产品需求、软件设计、测试有效性与开发效率之间达成达到很好的平衡。

全书一共13章：第1章简单介绍了通用的软件测试知识，以及软件测试的目标；第2章介绍如何利用测试驱动开发与单元测试来达到这个目标；第3章将如何进行单元测试的设计与编写；第4章将深入学习如何使用Apple的开发工具中绑定的OCUnit测试框架；第5章揭示在iOS应用程序的开发过程中如何从最初的需求规格书演进到最终产品；第6章阐述如何实现从需求描述中提取出来的数据模型；第7章将实现应用程序的业务逻辑；第8章以测试用例为指导，为BrowseOverflow应用程序设计并实现网络通信功能；第9章讲述如何编写视图控制器的代码，让BrowseOverflow程序将这些信息展示给用户；第10章将以上述类整合起来，实现一个功能完备且能够正常运行的应用程序；第11章讲解一些用于移除类之间的依赖性、代码运行环境依赖性以及线程依赖性的设计范式；第12章回答了在什么情况下适合采用TDD开发方式；第13章展望了业界一些更为新颖的功能、一些对已有技术的扩展以及一些有用的开发工具。

本书适合从事iOS测试驱动的软件开发人员和程序员阅读。

<<测试驱动的iOS开发>>

作者简介

Graham Lee, 资深安全顾问与iOS/Mac开发专家, 在全球iOS/Mac开发者社区中享有盛名。多次受邀在Voices

That Matter、NSConference、Association of C and C++

Users以及Qcon等国际技术会议上展示高质量的Cocoa软件工程实践。

在软件测试领域有深厚的积累和实践经验, 从6年前主管一个基于GNUstep的服务器应用程序的测试开发工作开始深入研究OCUnit与单元测试技术。

在研究iOS开发前, 曾为OS

X、NeXTSTEP及其他类UNIX系统开发过应用程序。

他还是一位经验丰富的技术作家, 著有《Professional Cocoa Application Security》一书。

<<测试驱动的iOS开发>>

书籍目录

译者序

前言

第1章 软件测试与单元测试简介

1.1 软件测试的目标

1.2 软件测试由谁来做

1.3 何时进行软件测试

1.4 测试实践举例

1.5 单元测试的适用范围

1.6 测试驱动开发对iOS开发者的意义

第2章 测试驱动开发技巧

2.1 测试先行

2.2 “失败、成功、重构”三部曲

2.3 设计易于测试的应用程序

2.4 更多有关重构的知识

2.5 不要实现目前用不到的功能

2.6 在编码前、编码中及编码后进行测试

第3章 如何写单元测试

3.1 需求

3.2 使用已知的输入数据来运行代码

3.3 查看运行结果是否符合预期

3.4 验证结果

3.5 使测试代码更具可读性

3.6 将多个测试用例组织起来

3.7 重构

3.8 总结

第4章 测试工具

4.1 Xcode附带的OCUnit测试框架

4.2 OCUnit的替代方案

4.2.1 GTM

4.2.2 GHUnit

4.2.3 CATCH

4.2.4 OCMock

4.3 持续集成工具

4.3.1 Hudson

4.3.2 CruiseControl

4.4 总结

第5章 针对iOS应用程序的测试驱动开发

5.1 产品目标

5.2 用例

5.3 执行方案

5.4 开始制作程序

第6章 数据模型

6.1 Topic类

6.2 Question类

6.3 Person类

<<测试驱动的iOS开发>>

- 6.4 将Question类与其他类关联起来
- 6.5 Answer类
- 第7章 应用程序逻辑
 - 7.1 执行方案
 - 7.2 建立Question对象
 - 7.3 用JSON数据构建Question对象
- 第8章 网络相关代码
 - 8.1 NSURLConnection类的设计
 - 8.2 实现StackOverflowCommunicator类
 - 8.3 总结
- 第9章 视图控制器
 - 9.1 类结构
 - 9.2 视图控制器类
 - 9.3 TopicTableDataSource类与TopicTableDelegate类
 - 9.4 通过已有的视图控制器建立新的控制器对象
 - 9.5 提问列表的数据源
 - 9.6 接下来的任务
- 第10章 整合所有部件
 - 10.1 完成应用程序的工作流程
 - 10.2 显示用户头像
 - 10.3 收尾与清理
 - 10.4 发布应用
- 第11章 为测试驱动开发进行软件设计
 - 11.1 针对接口进行设计，而不要针对实现
 - 11.2 用命令代替查询
 - 11.3 简洁而专注的类与方法
 - 11.4 封装
 - 11.5 使用比重用更好
 - 11.6 测试并发代码
 - 11.7 别耍没有必要的小聪明
 - 11.8 优先选择宽而浅的继承体系
 - 11.9 综述
- 第12章 在既有项目中运用测试驱动开发
 - 12.1 第一个测试用例是最重要的
 - 12.2 通过重构使得代码更易于测试
 - 12.3 编写测试使得代码更易于重构
 - 12.4 真的需要写这么多测试用例吗
- 第13章 测试驱动开发展望
 - 13.1 使用一系列输入输出值构建测试用例
 - 13.2 行为驱动开发
 - 13.3 自动生成测试用例
 - 13.4 自动生成能够通过测试的代码
 - 13.5 综述

<<测试驱动的iOS开发>>

章节摘录

第1章 软件测试与单元测试简介要想最大限度地从单元测试中受益，就必须理解它的目标及它是如何改进软件开发过程的。

在本章中，读者将会学到一些通用的软件测试知识，这些知识也适用于单元测试。

这一章也会讲到软件测试的优点和缺点。

1.1 软件测试的目标很多软件项目的目标都是盈利，实现这个目标的通常方式即通过应用商店来出售软件或者以其他方式授权给用户使用并收取费用。

那种为了程序开发者内部使用所制作的软件，则会通过提高某个业务流程的效率，减少该流程所耗的时间来间接地盈利。

如果通过提高业务流程效率节省的成本大于开发该软件的花销，那么这个软件项目就是盈利的。

开源软件的开发者通常以出售“支援服务包”（support package）来获利，他们也会使用自己开发的软件，在这种情况下，前面的论断依然成立。

所以说，软件开发经济学的基本原则就是，如果某个软件项目的目标是盈利—不管是向客户出售最终产品还是供开发者内部使用，那么它要想成功地达成此目标，必须创造某种高于软件制作开销的价值才行。

笔者也知道这并非一个具有非凡意义的论断，不过可以将它推及到软件测试领域中。

如果软件测试（也叫做“质量保证”（Quality Assurance, QA））是为了支持软件项目，则它必须对实现盈利有帮助才行。

这一点很重要，因为它对软件测试做出了限定：如果软件测试开销过大，导致项目亏损，那么这种测试就不适合去做。

不过对软件进行测试可以保证产品能正常运行，而产品又包含了客户所需的功能。

如果你不能展示这些功能的价值，那么客户就不会购买这个产品。

注意，测试的目标是证明产品能够正常运行，而不是发现bug。

软件测试是在做“质量保证”，而不是“质量介入”。

查找bug通常是个坏主意。

为什么呢？

因为要修复bug就必须有开销，而这部分资金本来是付给开发者的，让其一开始就写出无bug的软件，现在却被浪费了。

在理想的情况下，大家可能会认为开发者只需写出无bug的软件，通过快速的测试确保它们没有问题，然后将其上传到iTunes Connect账户，就可以坐等财源滚滚而来了。

不过别急，这么做也会以另一种方式导致同样的问题：在测试软件之前，需要多长的时间来编写100%无bug的软件呢？

这样做的开销是多少？

这么说的话，合适的软件测试方案看起来是一种折中：既要保证对软件开发进度有一定程度的控制，又要在工程开销许可的范围内进行一定程度的检查，以确保产品确实能够正常运行。

这种平衡应该着眼于将所发行产品的运行风险降低到一个可以接受的水平上。

所以说，“最具风险的组件”，也就是那些对于软件的运行至关重要的组件或者那些最有可能隐藏bug的组件，应当首先测试，然后测试那些风险稍低的组件，依次测试，直到你觉得所有剩下的风险因素都不值得再投入时间和资金去测试为止。

最终的结果应该是让客户看到软件实现了预期功能，从而值得为此付费购买才对。

<<测试驱动的iOS开发>>

编辑推荐

《测试驱动的iOS开发》为Amazon五星级畅销书，作者权威，在全球iOS/Mac开发者社区享有盛誉！完美地展现了测试驱动开发方法与iOS开发的结合，能使iOS开发者在产品需求、软件设计、测试有效性与开发效率之间达成达到很好的平衡。

<<测试驱动的iOS开发>>

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>