

# 《测试驱动开发:实用指南(平装)》

## 图书基本信息

书名：《测试驱动开发:实用指南(平装)》

13位ISBN编号：9787508324821

10位ISBN编号：750832482X

出版时间：2004-9

出版社：中国电力出版社

作者：阿斯特尔斯 (AstelsDavid)

页数：478

译者：崔凯

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《测试驱动开发:实用指南(平装)》

## 内容概要

测试驱动开发(TDD)是近年来“敏捷革命”中最热的话题之一,但是,市面上鲜见与此相关的有深度的作品。这本书的出版填补了这个空白,满足了众多一线程序员的需要。作者Davis Astels将这本书看作是Kent Beck的经典作品《Test Driven Development By Example》(英文影印版及中文版由中国电力出版社出版)的补充,重点阐述利用TDD开发所必要的技术和工具上,因此对实际开发更具实用性。

本书共分四部分。首先介绍了与TDD有关的基本主题,例如重构、意图导向编程(programming by intention),当然还有TDD背后的原理剖析。接着深入分析了一些工具和技术,例如JUnit、模拟对象(mock objects)。第三部分主要讲解如何将TDD应用到一个真实Java项目,这是本书的精华所在,作者亲自带你进入TDD世界,告诉你如何将看似复杂的“测试为先”在项目开发中尽情演绎。第四部分逐一概述了xUnit家族成员,如RUBYUNIT、SUNIT等。附录包含极限编程和敏捷开发的入门材料,以及一些在线资源。

本书还包含了大量练习题并附有详尽的参考答案,以方便读者的学习,加深读者对本书内容的理解。

著名的敏捷开发专家Dave Astels通过一个从头至尾使用Java和Unit测试框架创建的项目,向你展示了TDD的精妙之处。你将会领悟到“测试为先”的工作机理,它之所以成为一种有效开发方法的原因,使用过程中将会碰到的困难以及如何现实中取得TDD所许诺的一切。

- 最具实用性!书中遍布了可下载的示例代码、练习题以及网上资源链接
- 介绍强大的TDD工具和技术——包括关键的JUnit扩展,相关内容由它们的创建者(Scott Ambler、Tim Bacon、Mike Bowler、Mike Clark、Bryan Dollery、James Newkirk、Bob Payne、Kay Pentacost和Jens Uwe Pipka)撰写。

# 《测试驱动开发:实用指南 (平装)》

## 作者简介

David Astels早在1985年就开始编写软件，涉及领域从嵌入式环境控制到知识产权保护系统，直到电能交易系统。他创建了Adaption([www.adaptionsoft.com](http://www.adaptionsoft.com))，专门提供极限编程 (XP) 和软件工艺(Software Craftsmanship)方面的咨询、培训、指导和编程业务。他从事The Coad Letter中TDD一版的编辑工作，并且在一些最为重要的有关极限编程和敏捷过程的大会上发表演讲。同时他也是《A Practical Guide to eXtreme Pronramming》(Prectice Hall PTR)作者之一。

# 《测试驱动开发:实用指南(平装)》

## 书籍目录

序言前言第1部分 背景知识介绍 第1章 测试驱动开发 什么是测试驱动开发? 让计算机来告诉你 一个简短的例子 小节 第2章 重构 什么是重构? 什么时候进行重构 如何重构 一些重要的重构方法 通过重构来引入模式 小节 第3章 意图导向的编程 名字 简单 要做有根据的假设 如何进行意图导向的编程 “无需注释” 小节第2部分 工具与技术 第4章 JUnit 体系结构概览 断言 编写一个测试用例 (TestCase) 运行测试 使用setUp()和tearDown() 使用TestSuite() 相互关系 测试的归属问题 一些建议 小节 第5章 JUnit的扩展 (extensions) 标准扩展 使用MockObjects补充的断言 使用JUnitPerf来测试性能与可伸缩性 Daedalus JUnit扩展 使用xmlUnit来编写基于XML的测试 Gargoyle 软件公司的JUnit扩展 第6章 与JUnit有关的工具 Jester NoUnit Clover Eclipse IDEA 第7章 模拟对象 (Mock Objects) 模拟对象 例子解说 模拟对象的用途 如果能这样该有多好? 一个普通的例子 MockObjects框架 MockMaker EasyMock 小节 第8章 以测试优先的方式开发一个GUI应用 例子 AWT机器人 蛮力测试 (Brute Force) JFCUnit Jemmy 超瘦 (Ultra-Thin) GUI 小节第3部分 一个彻头彻尾的采用TDD开发的Java软件项目 第9章 项目描述 概述 用户描述及相关任务 第10章 电影列表 为电影制作一个容器 为电影列表创建GUI 通过GUI增加电影 回顾 第11章 可以为电影更名 支持对电影名的编辑 对电影更名的GUI 回顾 第12章 电影是唯一的 电影是唯一的 发现不唯一时的出错信息 回顾 第13章 评分 (ratings) 给电影打分 在GUI中显示评分 对评分进行编辑 回顾 第14章 类别 增加一种类别 在GUI中显示类别 提供可供选择的类别 回顾 第15章 根据类别进行筛选 获得基于类别的子列表 支持一种称做ALL (所有) 的类别 在GUI上添加类别选择器 处理电影类别的变更 界面修整 回顾 第16章 持续性存储 (persistence) 将数据输出到平直文件中 GUI中的Save-As菜单项 GUI中的Save菜单项 从平直文件中读取数据 加载到GUI中 回顾 第17章 排序 电影比较 对电影列表进行排序 从MovieListEditor中获取排序后的列表 在GUI中增加排序操作 回顾 第18章 多重评分 多重评分 评分来源 修正后的持续性存储 在GUI中显示电影的多重评分 在GUI中添加增加评分的功能 删除单一评分域 回顾 第19章 评论 为评分附加评论 保存评论 加载评论 显示评论 增加评论 回顾 第20章 项目回顾 设计 测试代码 vs. 应用代码 测试质量 模拟对象的使用 总体评述 调试 测试清单 小节第4部分 xUnit一族 第21章 RUBYUNIT 框架描述 范例 第22章 SUNIT 框架描述 范例 第23章 CPPUNIT 框架描述 范例 第24章 NUNIT 框架描述 范例 第25章 PYUNIT 框架描述 范例 第26章 VBUNIT 框架描述 范例附录A 极限编程 敏捷革命 极限编程 四种变化量 (The Four Variables) 价值观 (Values) 实践方法 (Practices) 小节附录B 敏捷建模 与建模有关的荒诞言论 敏捷建模 (AM) 导论 什么是敏捷模型? 附录C 联机资源 论坛 与敏捷建模有关的信息 与极限编程有关的信息 与JUnit有关的软件 与JUnit有关的信息 各种工具 XUnit家族的其他成员 相关公司 杂类附录D 练习答案参考文献

# 《测试驱动开发:实用指南(平装)》

## 精彩短评

1、TDD的入门级教材，深入浅出，介绍了很多工具，同时所举的项目例子也很务实，适合TDD初学者。

2、这是一本值得推荐的书，本书的重点在第三章，整个章节是以一个TDD驱动开发的Java项目作为示例——这也印证了本书的标题《测试驱动开发：实用指南》中的实用二字。作者试图展示TDD在项目中是全能的，但我个人却认为不必要对界面使用单元测试。

理由一、从本书中的java项目来看：即开发一个可以对电影进行常规操作、分类、评价、评分、数据可以持久化的程序，界面也不复杂。但这样简单的界面却拥有大多数的单元测试代码（与业务逻辑的单元测试代码相比较）。这似乎产生一个80/20问题：我们用80%的单元测试代码去解决一个只占整体项目20%的界面问题。

理由二、使用RAD的方式开发界面可以极大的提高开发效率，WYSIWYG（所见即所得）的方式已经被主流的开发环境所支持。另外，在分层系统中，表示层本身应该是一个非常thin的层，其中并不包含业务逻辑，需要测试的内容应该是较少的。从界面的开发效率来说，RAD的方式比TDD高很多。

理由三、是不是项目中任何代码都要求单元测试？答案是否定的。单元测试是有范围的。比如，一些getter/setter方法是不需要测试的。同样花力气写代码去测试界面上是否存在某个Textbox，在我个人看来也是不需要。

理由四、如果对界面进行测试，仍然需要传统的调试器或者是使用脚本语言驱动的工具。

3、里面的一些测试技术都很老了，比如junit，现在都注解了。

本来是想跟着里面的电影列表的例子来实践一把的，不过对swing不熟悉，而且没有源代码可以下载的，遇到了一些书面上没有提到的代码，导致中间没法继续下去。

前面几章的一些理论还是不错的，有些以前自己已经知道了，现在再看一遍，有一种顿悟的感觉。

如果能拿java最擅长的web应用做例子就完美了，毕竟做swing应用的人比较少，至少在国内是如此。

4、一本教你怎么真正运用先进开发方法的书，很棒

5、不错的书。。

6、简单的扫了几眼

7、有些土，不过理不糙

8、书中的范例都是用户图形界面方面的，要是也有WEB方面的范例就更好了

1、这是一本值得推荐的书，本书的重点在第三章，整个章节是以一个TDD驱动开发的Java项目作为示例——这也印证了本书的标题《测试驱动开发：实用指南》中的实用二字。作者试图展示TDD在项目中是全能的，但我个人却认为不必要对界面使用单元测试。理由一、从本书中的java项目来看：即开发一个可以对电影进行常规操作、分类、评价、评分、数据可以持久化的程序，界面也不复杂。但这样简单的界面却拥有大多数的单元测试代码（与业务逻辑的单元测试代码相比较）。这似乎产生一个80/20问题：我们用80%的单元测试代码去解决一个只占整体项目20%的界面问题。理由二、使用RAD的方式开发界面可以极大的提高开发效率，WYSIWYG（所见即所得）的方式已经被主流的开发环境所支持。另外，在分层系统中，表示层本身应该是一个非常thin的层，其中并不包含业务逻辑，需要测试的内容应该是较少的。从界面的开发效率来说，RAD的方式比TDD高很多。理由三、是不是项目中任何代码都要求单元测试？答案是否定的。单元测试是有范围的。比如，一些getter/setter方法是不需要测试的。同样花力气写代码去测试界面上是否存在某个Textbox，在我个人看来也是不需要。理由四、如果对界面进行测试，仍然需要传统的调试器或者是使用脚本语言驱动的工具。

2、里面的一些测试技术都很老了，比如junit，现在都注解了。本来是想跟着里面的电影列表的例子来实践一把的，不过对swing不熟悉，而且没有源代码可以下载的，遇到了一些书面上没有提到的代码，导致中间没法继续下去。前面几章的一些理论还是不错的，有些以前自己已经知道了，现在再看一遍，有一种顿悟的感觉。如果能拿java最擅长的web应用做例子就完美了，毕竟做swing应用的人比较少，至少在国内是如此。

3、今天差不多把这本书看完了。这本书出版于2004年，对于现在的新人来说不太合适，因为里面使用的测试相关的库都太老了，有一些测试的实践可能也发生了一些变化。对于已经有了一些TDD经验的人来说还是有一些价值，可以看到历史的变迁，更好的理解现在的情况。前几章有一些关于clean code以及测试方面的示例感觉比较有用，也澄清了我的一些疑惑。中间一大块是一个使用TDD来真实开发的实例项目，对于没怎么接触TDD的人来说比较有用，可以照着做，但对于已经熟悉的人来说意义不是很大，快速翻一遍即可。后面介绍的一些xUnit系列的工具，可能也过时了。最后有一些关于敏捷开发和敏捷建模的介绍性的资源，有一些价值。我比较关注的“怎么分解任务”和“怎么重构到模式”，书上只是简单提了两句，有点遗憾。总体来说，在十一年之后看这本书，价值不大，但是也值得花一天的时间去阅读。

# 《测试驱动开发:实用指南(平装)》

## 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:[www.tushu000.com](http://www.tushu000.com)