

# 《设计驱动测试》

## 图书基本信息

书名：《设计驱动测试》

13位ISBN编号：9787111440668

出版时间：2014-1-1

作者：Matt Stephens,Doug Rosenberg

页数：276

译者：郑静

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)

# 《设计驱动测试》

## 内容概要

本书主要介绍了设计驱动测试（DDT）的思想和一种全新的软件开发过程—ICONIX。作者希望通过一个个真实而具体的案例告诉读者，如何在实践中达到测试的最佳平衡和优化。全书共分12章，第1~3章介绍了全新的DDT和传统的TDD之间的差异。第4~8章通过一个真实的Web地图案例，讲解了如何在项目实践中运用DDT的思想。第9~12章主要描述了如何在自动化测试、算法测试、单元测试等环节中使用DDT。

本书可供软件开发人员、测试人员以及项目管理人员阅读和参考。

# 《设计驱动测试》

## 作者简介

Matt Stephens是一家位于伦敦中心的金融机构的软件咨询顾问，并且是独立出版社Fmgerpress的创始人。他为很多杂志和网站(包括The Register和ApplicationDevelopment Trends)供稿。你可以在Software Reality在线找到他。

Doug Rosenberg于1984年在自己的家里创建了ICONIX，在花了几年时间建立了CASE工具后，从1990年开始他为各个企业讲解面向对象分析与设计。ICONIX专门从事UML与SysML的培训，并且提供现场培训与公开课。Doug在早于UML问世之前的1993年开发出了统一Booch / Rumbaugh / Jacobson建模方法，并且在1995年开始写书。

## 书籍目录

译者序

序

关于作者

关于技术评审人

致谢

开场白

第一部分 DDT vs. TDD

第1章 有人弄反了

DDT要解决的问题

很难知道什么时候完成

将测试放在后期代价更大

测试设计糟糕的代码很困难

用户级测试很容易被遗忘

开发人员变得自负

测试有时缺少目标

对DDT的与工具无关的快速概览

DDT的结构

DDT实战

TDD与DDT的不同之处

示例项目：Mapplet 2.0介绍

小结

第2章 使用TDD的Hello World

TDD的十大特性

10. 测试驱动设计

9. 完全没有文档

8. 所有东西都是单元测试

7. TDD测试不是完全的单元测试

6. 验收测试提供针对需求的反馈

5. TDD导致盲目自信的变更

4. 设计在不断增长

3. 有一些预先设计就可以了

2. TDD产生了大量测试

1. TDD实在太难了

使用TDD实现登录用例

理解需求

考虑设计

编写第一个测试先行的测试

编写登录检查代码从而使测试通过

创建模拟对象

从重构代码看设计的浮现

TDD中的验收测试

结论：TDD实在太难了

小结

第3章 使用DDT的Hello World

ICONIX/DDT的十大特性

10. DDT包含业务需求测试

9. DDT包含场景测试

8. 测试是被设计驱动的
7. DDT包含控制器测试
6. DDT测试更灵活，更简单
5. DDT中的单元测试是“经典”的单元测试
4. DDT中的测试用例可以转换成测试代码
3. DDT测试用例指导测试计划
2. DDT测试对开发和测试团队都很有用
1. DDT可以消除重复工作

使用DDT实现登录

步骤1：创建健壮性图

步骤2：创建控制器测试

步骤3：添加场景

步骤4：将控制器测试用例转换为类

步骤5：生成控制器测试代码

步骤6：绘制序列图

步骤7：创建单元测试用例

步骤8：填充测试代码

小结

第二部分 真实世界中的DDT：Mapplet 2.0旅游网站

第4章 Mapplet项目简介

ICONIX 流程/DDT十大“ To-Do ” 列表

10. 创建架构
9. 对需求达成共识并进行测试
8. 从问题域驱动设计
7. 使用UI故事板编写用例
6. 编写场景测试验证用例
5. 测试概要设计和详细设计
4. 经常更新模型
3. 保持测试脚本与需求同步
2. 更新自动化测试
1. 比较待发布版本和原始用例

小结

第5章 详细设计和单元测试

单元测试十大“ To-Do ” 列表

10. 从序列图开始
9. 在设计中标识测试用例
8. 为每个测试用例编写场景
7. 聪明测试：避免重叠测试
6. 把测试用例转换为UML类
5. 编写单元测试和相关的代码
4. 编写白盒单元测试
3. 使用模拟对象框架
2. 用单元测试测试算法逻辑
1. 编写集成测试的独立套件

小结

第6章 概要设计和控制器测试

控制器测试十大“ To-Do ” 列表

10. 从健壮性图开始
9. 为控制器标识测试用例

8. 为每个测试用例定义一个或者多个场景
7. 填写描述、输入和验收标准
6. 生成测试类
5. 实现测试代码
4. 编写容易测试的代码
3. 编写“灰盒”控制器测试
2. 串联控制器测试
1. 编写集成测试的独立套件

小结

第7章 验收测试：扩展用例场景

场景测试的十大“To-Do”列表

Maplet用例

10. 从一个叙述性用例开始
9. 把这个用例转换成一个结构化的场景
8. 确保涵盖所有的可选方案和意外场景
7. 增加前置条件和后置条件，将每个场景分支连接起来
6. 生成活动图来检查结构化场景
5. 创建外部测试集来细化场景
4. 把测试用例放进用例图
3. 进入EA测试视图
2. 根据需要细化场景
1. 为测试团队生成测试计划文档

这个过程精髓是……

小结

第8章 验收测试：业务需求

十大需求测试“To-Do”列表

10. 从一个域模型开始
9. 编写业务需求测试
8. 对需求进行建模和整理
7. 从需求创建测试用例
6. 与用户一起审查你的计划
5. 编写手工测试脚本
4. 编写自动化需求测试
3. 导出需求测试用例
2. 使测试用例可见
1. 让你的团队参与其中！

小结

第三部分 高级DDT

第9章 单元测试的反模式（反面案例）

末日圣殿（特指某一种代码）

大背景

HotelPriceCalculator类

支持类

服务类

反模式

10. 复杂的构造函数
9. 滥用类继承
8. 静态微触发器
7. 静态方法和变量

6. 单例设计模式
5. 紧耦合
4. UI代码里实现业务逻辑
3. 滥用私有属性
2. 声明为final的服务对象
1. 热心的程序员开发的成熟的功能

小结

## 第10章 为易于测试而设计

十大为测试而设计的“ To-Do ”列表

末日圣殿——彻底修正

用例——确定我们需要做什么

识别控制器测试

计算总价格测试

获取最新价格测试

为易于测试而设计

10. 将初始化代码放在构造函数之外

9. 慎用继承

8. 避免使用静态初始化块

7. 使用对象级别的方法和变量

6. 避免使用单例设计模式

5. 保持类解耦合

4. 将业务逻辑放在UI代码之外

3. 使用“黑盒”和“灰盒”测试

2. 为常量预留“final”修饰符——通常避免修饰复杂类型（如Service Objects）为final

1. 坚持使用用户用例和设计

Quote Hotel Price用例的详细设计

控制器测试：计算总价

控制器测试：获得最新价格的测试

重构设计和代码

小结

## 第11章 自动化的集成测试

十大集成测试“ To-Do ”列表

10. 在概要设计里寻找测试模式

9. 不要忘记安全性测试

安全性测试：SQL注入攻击

安全性测试：建立安全会话

8. 决定编写哪个“等级”的集成测试

三个等级的不同点

了解编写哪个等级的集成测试

7. 概要设计驱动单元/控制器级别的集成测试

6. 从用例场景驱动场景测试

5. 编写端到端场景测试

模拟一个场景中的步骤

共享测试数据库

Mapplet例子：“高级搜索”用例

Vanilla xUnit场景测试

4. 使用“业务友好”型测试框架

3. 将测试GUI代码作为场景测试的一部分

2. 不要低估集成测试的难度

网络延迟

数据库元数据变化

随机变化的（又名“敏捷”）接口

远程系统中的bugs

阴雨天

1. 不要低估集成测试的价值

编写集成测试的关键点

小结

第12章 单元测试算法

十大算法测试“ To-Do ”列表

10. 从概要设计的控制器开始工作

9. 将控制器扩展成算法设计

8. 把图和域模型对应起来

7. 分割那些看上去不止做一个检查的判断结点

6. 为每个结点（活动和判断结点）建立一个测试用例

5. 为每个测试用例定义测试场景，一组输入和期望结果

4. 按照算法，从不同的源中创建输入数据

3. 把逻辑流程对应到独立的方法和类上

2. 编写“白盒”单元测试

1. 在其他类型的设计图上使用DDT技术

小结

附录 爱丽丝漫游用例国

介绍

第1部分

爱丽丝在看书的时候睡着了

用例驱动开发的承诺

一种把用例文本和对象连接起来的分析模型

简洁且直接

<<包含>> 还是<<扩展>>

我们迟到了! 我们必须开始编码了!

爱丽丝想知道如何才能把用例变成代码

抽象的.....基本的

有点太过抽象了?

目的中心化.....

我们真的打算为每个用例都指定这些东西吗?

第2部分

爱丽丝口渴了

爱丽丝感到头晕

设想.....（敬请约翰列侬原谅，这首歌改编自他的作品）

结对编程意味着再也不用把需求写下来了

没时间去写需求了

你也许也会说“代码就是设计”

谁在乎用例?

C3项目被中止了

一次且只有一次?

没有写下需求之前，爱丽丝拒绝开始写代码

你因为预先设计而被定罪.....

CMM已经死了，砍掉她的脑袋!

一些严肃的设计重构



## 第3部分

爱丽丝醒了

缩小“什么”和“如何”之间的距离

静态模型和动态模型被连接在了一起

行为被定位到序列图里

这里面的教训在于.....

尾声——乱七八糟的测试.....

索引

# 《设计驱动测试》

## 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)