乐学 Team 分享系列之

Test Driven Development

熊俊

Roadmap

- TDD 是什么？
- 为什么考虑 TDD?
- 如何开始 TDD?
- TDD 实施难点
- Summary

TDD 是什么？

- TDD 不是什么？

- 测试驱动的开发（与设计方法）
- 文档（接口及代码使用规范）
- 编码后的设计验证与功能回归
- 形式上恰好是可执行的 UnitTests

TDD 是什么？

- 从写 Unit Test 开发，失败的测试驱动开发过程
- 描述被实现对象应该具有的行为，而不仅是验证对象是否正确
- 对函数或 oo 中类级别的测试
- T 并不明确要求对实现代码的覆盖率

实战

敏捷该产出什么样的系统？

- 据 2002 年 Standish Group 的报告，软件系统中有 65% 的功能是客户从来不用或者很少用到的。传统意义上大家认为敏捷开发应该让团队开发得更快，生产率更高
- 　与提高效率相比，使用 Scrum,TDD 能够帮我们从整个需求中确定真正有用的那 35% ，而且往往这 35% 的功能往往实现起来并不是那么困难。因为做得少，所以做得更快。

为什么考虑 TDD?

对需求 / 设计 / 代码的影响

- 减少无用功 , 迫使从需求验收角度入手
- 锻炼自顶向下的分述与设计能力
- 提高代码的可测试性 , 做到简单设计 (够用)

如何开始 TDD?

- 理解观念
- 改变开发习惯
- 结对 / 有经验者指导
- 理解在短期内可能呈现开发速度更慢现象
- 团队敏捷能力：迭代增量式开发，持续集成等
- 个人设计能力提升
- 组合拳：TDD + Code Review + Refactory

资料推荐

- Kent Beck “ 测试驱动开发”
- Martin Fowler “ 重构”
- Uncle Bob “ 敏捷软件开发原则、模式、实践”
- Head First OOA & OOD
- Eric Evans 领域驱动设计 (DDD)
- Mike Cohn “User Story Applied”

TDD 实施难点

- 需要足够的设计能力
  - 熟练的编程基础
  - 清晰的 MVC 设计分层与解耦能力
  - OOA&OOD 与建模能力
- 需要实践者坚持
- TDD 不是万能的

实施 TDD 易犯错误

- 当你使用一把锤子时，你能犯的最大的错误就是尝试用它把钉子撬出来而不是砸进去 .(熊节)
- 过度编码 : 生产代码不止使测试通过，还实现了很多别的东西
- 不做重构 : UnitTest, Code

Summary

- 编码前的设计，编码完成后的设计验证
- 形式上以 Unit Test 呈现
- 从需求角度入手，锻炼自顶向下分析能力
- 不了解设计，不知道解耦，TDD 很痛苦
- TDD 不是万能的

Thanks!

Q & A