

极客

极客练功房·TDD

测试驱动开发实战

2019年12月20日星期五 北京

主办方: **eXtreme**
Programming

联合主办: **极客学院**
jikexueyuan.com

ThoughtWorks®
Community

媒体伙伴:  **掘金**

TDD

极客练功房

- 「极客」就是「极限编程实践者」的简称。
- 「极客练功房」就是「极限编程实践者的练功房」、「极限编程实践的练功房」。

两个主旨：

- 认同“基本功”对做好软件的决定性价值
- 采用“刻意练习”方法提升基本功

极客练功房

- 基于极限编程的核心实践，直指个人和团队基本能力
- 全程采用刻意练习方法，不讲废话，只练真功夫
- 社群资深教练现场点拨，打通关窍
- 温馨网友聚会，面基志同道合的小伙伴

在线版：<https://www.jiker.com>

关于讲师

刘清

极客学院·架构师

中国极限编程社群·高级讲师

前 **ThoughtWorks** 咨询师

王祎

ThoughtWorks 咨询师

极客

练功

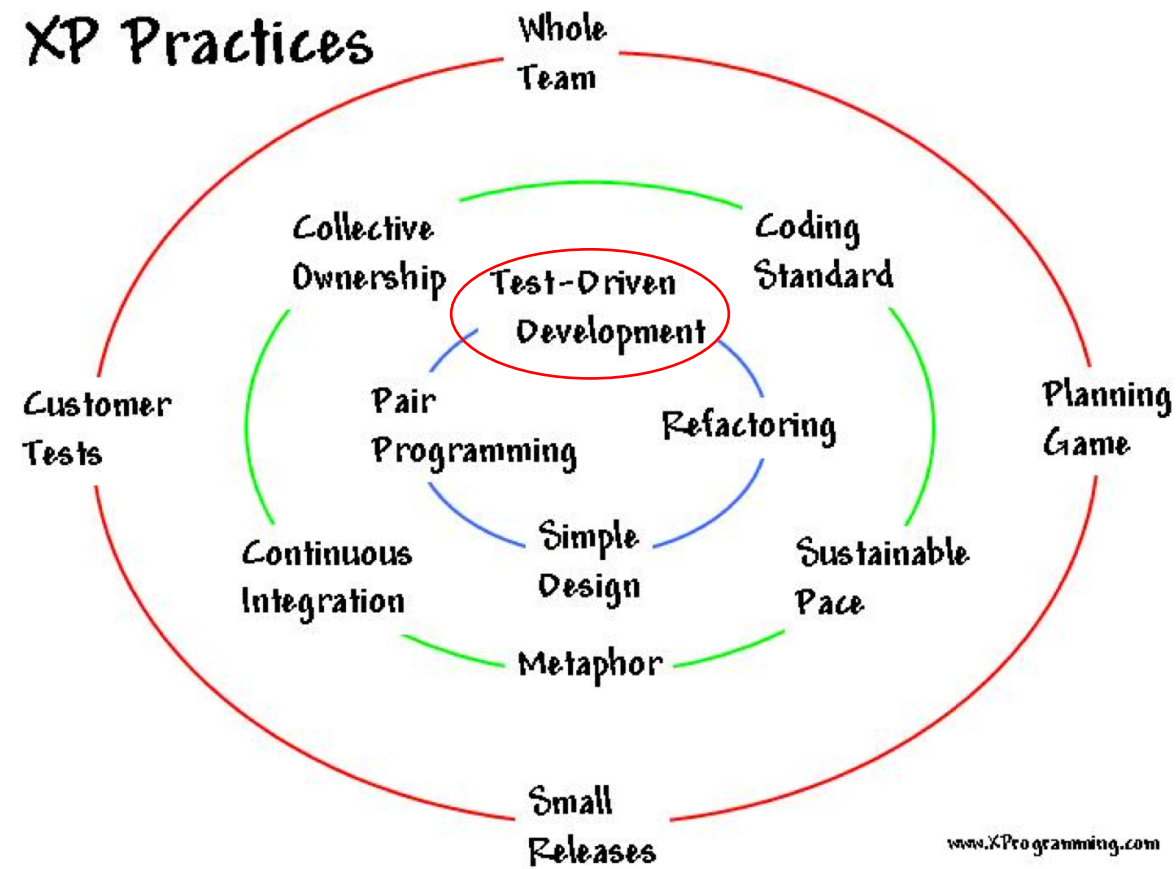
之

TDD

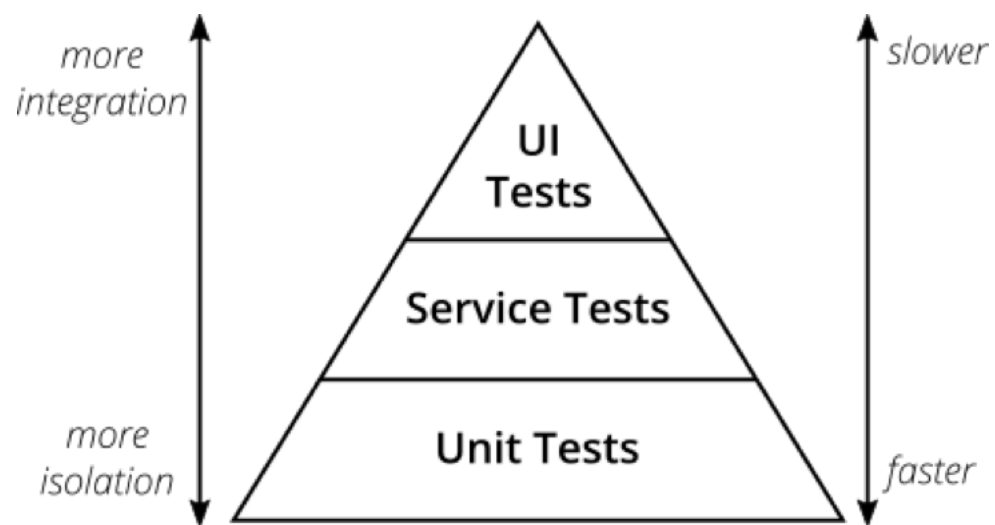
测试驱动开发

Test Driven Development

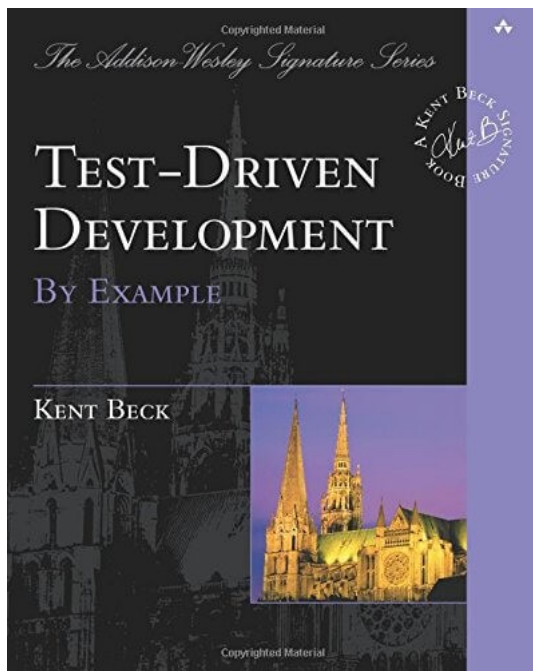
TDD 是什么？



T - 指什么？



测试驱动开发 (TDD)



测试驱动开发不是一种测试技术。
它是一种分析技术、设计技术，
更是一种组织所有开发活动的技术。
-- 《Test Driven Development》

Case：非TDD的编码场景

- 需求分析，想不清楚细节，管他呢，先开始写
- 发现需求细节不明确，去跟业务人员确认
- 确认好几次终于写完所有逻辑
- 运行起来测试一下，果然不工作，调试
- 调试好久终于工作了...
- 转测试，QA 测出 bug，debug，打补丁
- 终于，代码可以工作了...
- 一看代码烂的像坨屎，不敢动，动了还得手工测试，还得让 QA 测试，还得加班...

需求遗漏

反馈周期长

打地鼠软件

测试先行

需求~~X~~遗漏

提前对业务需求进行拆分，确保所有测试场景覆盖住需求

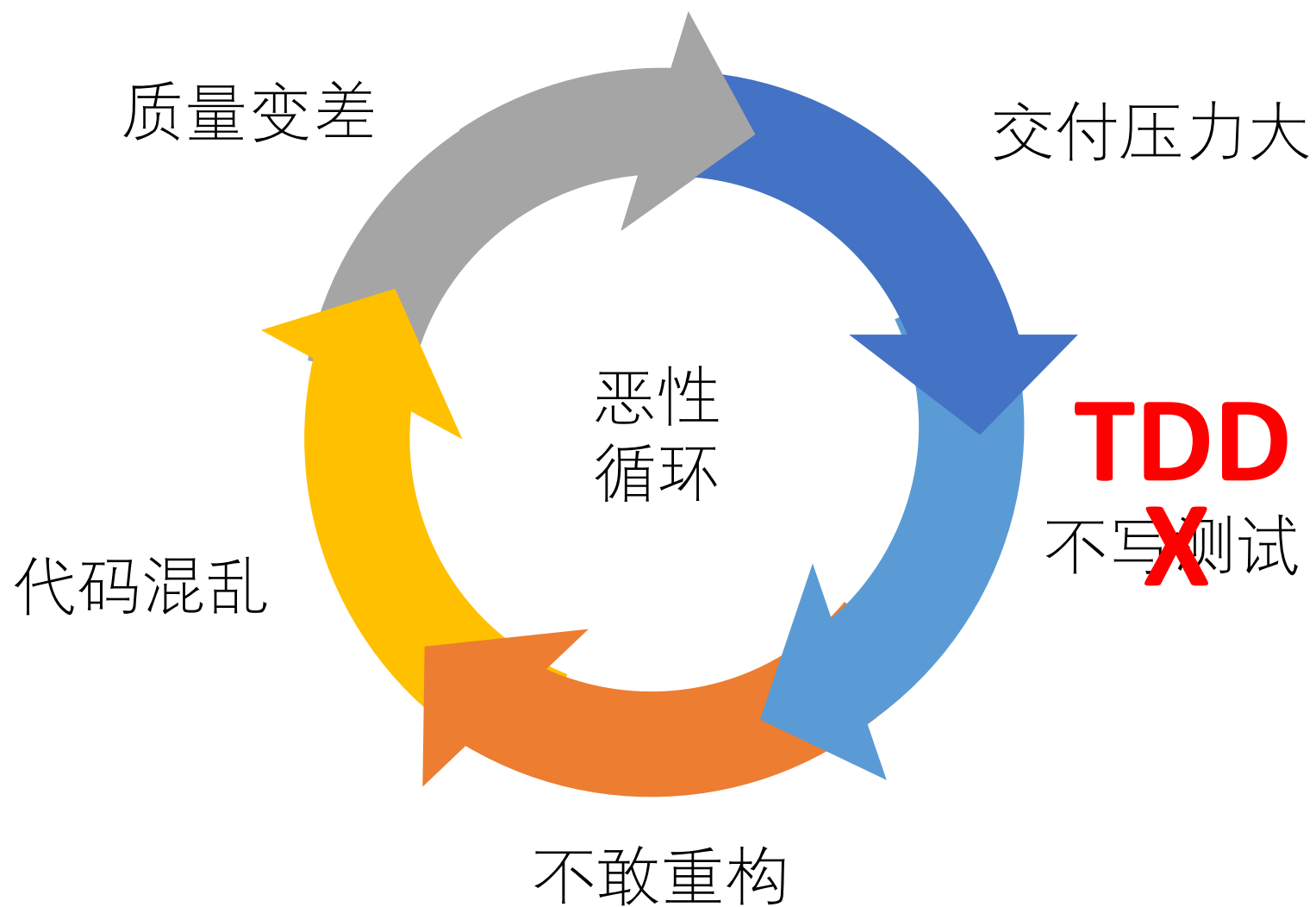
反馈~~X~~期长

能够快速对功能代码是否可行进行反馈。无需等到所有功能写完

打地~~X~~软件

有测试的保障，可以放心大胆地重构，无需担心新功能对老功能的破坏

TDD：避免恶性循环



TDD的目标

代码

简洁

可用

TDD规则

简洁

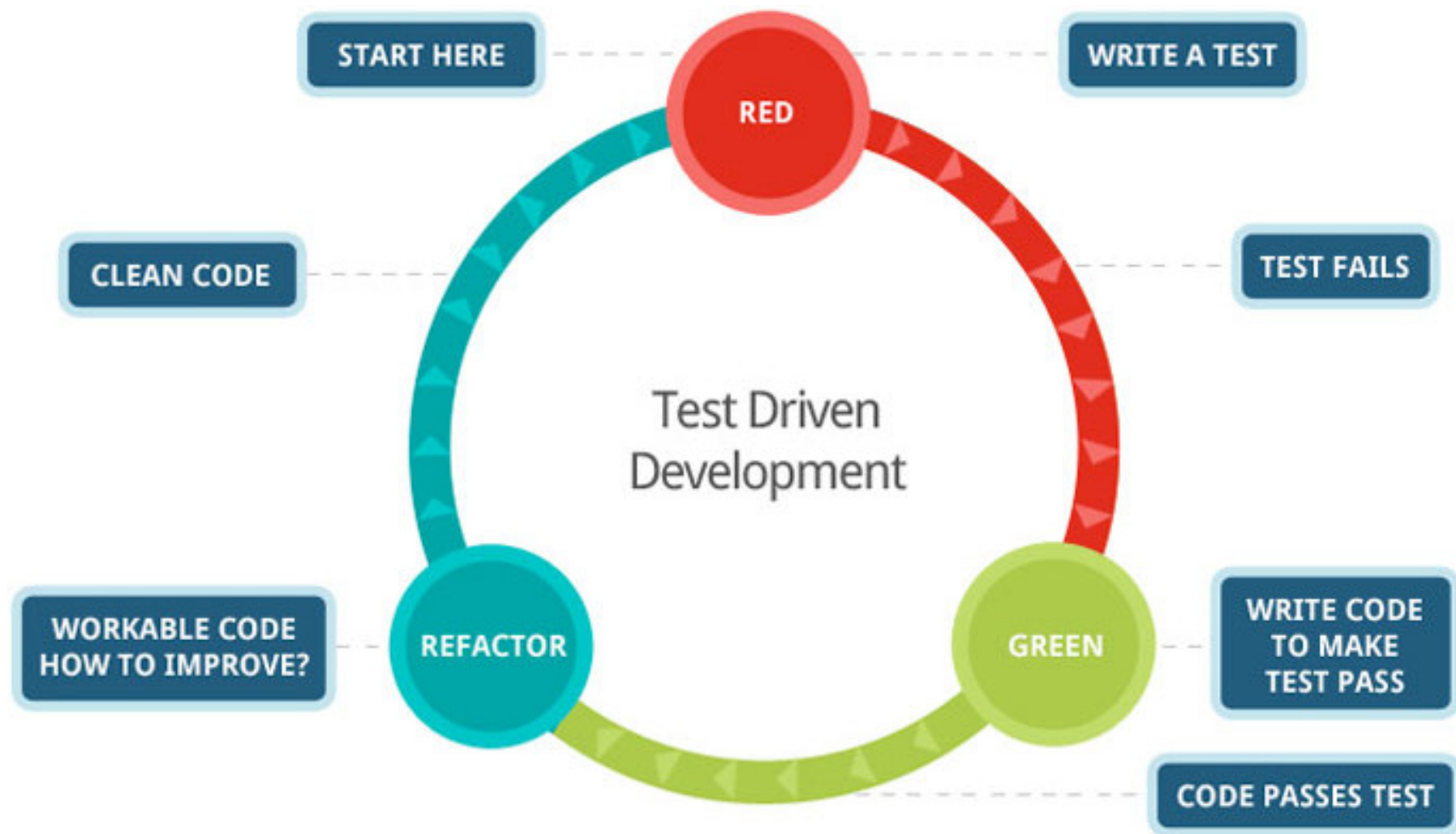
1. 小步前进 —— 每次允许写一个刚好失败的测试
2. **每次只编写刚刚好使测试通过的代码**
3. 相互独立的测试 —— 测试之间必须互不干扰
4. **消除重复设计，优化代码结构**

TDD步骤

可用

1. Tasking 分解任务
2. 列出测试列表
3. TDD Cycle

TDD Cycle : 红-绿-重构



TDD编码方式

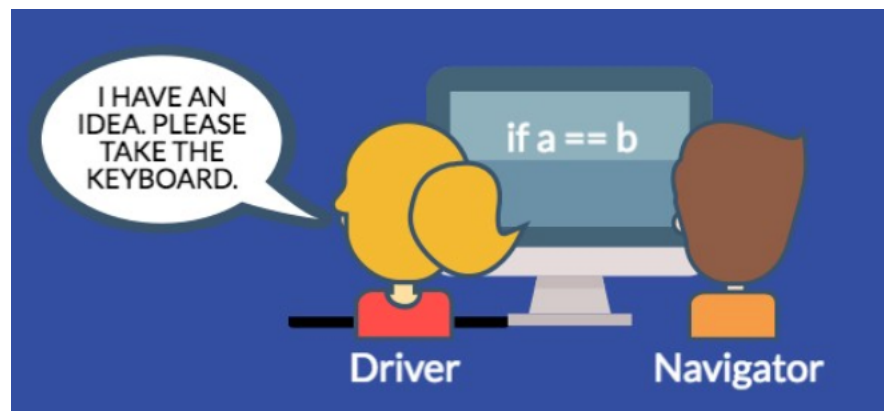
- 先**分解任务**，分离关注点
- 列 Example，用**实例化**需求，澄清需求细节
- 写测试，只关注需求，程序的**输入输出**，不关心中间过程
- 写实现，不考虑别的需求，用**最简单的方式**满足当前这个小需求即可
- **重构**，用手法消除代码里的**坏味道**
- 写完，手动测试一下，基本没什么问题，有问题补个用例，修复
- 转测试，小问题，补用例，修复
- 代码整洁且用例齐全，**信心满满**地提交。

更好的开始TDD

1. 断言优先
2. 足够简单的测试数据

Pair 进行TDD的方式

Driver - Navigator



Ping Pong



极客

练功

之

TDD

TDD 实践

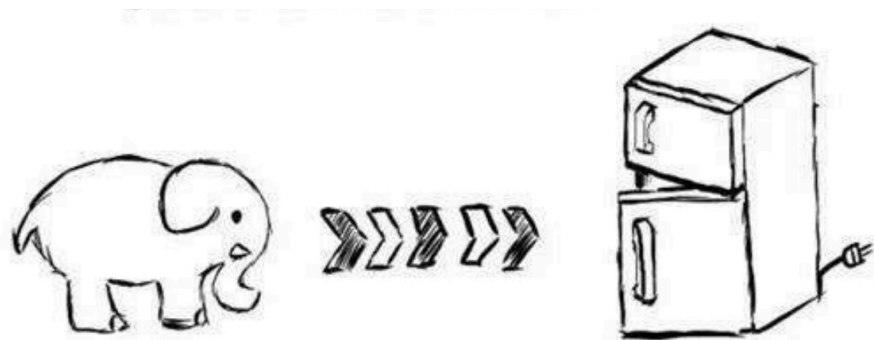
TDD in Action

TDD 实践

- 任务分解
- 单元测试

任务分解

把大象放进冰箱需要几步？



任务分解

把长颈鹿放进冰箱需要几步？

任务分解

森林里开动物大会谁没去？

任务分解

一个人过鳄鱼湖为什么没有被鳄鱼吃掉？

任务分解的核心要点

1. 各自独立
2. 完全穷尽

练习（3分钟）

- 列 Task：把大象装进冰箱。

把大象装进冰箱

- 把大象装进空的冰箱，成功
- 把大象装进恰好有空间容纳大象的冰箱，成功
- 把大象装进满的冰箱，失败
- 把大象装进容纳不下大象的冰箱，失败
- 把大象装进null，*&^\$#&**%
-

解决问题关注点

- 输入从哪来，输出到哪去
- 函数的触发事件、权限
- 代码里写死的数据hard code有哪些
- 处理过程：描述 - 函数名；怎样实现；为什么这样实现

任务分解关注点

- 输入从哪来，输出到哪去

单元测试

单元测试应该怎么写？

单元测试

三段式：

- GIVEN：测试准备
- WHEN：测试动作
- THEN：测试断言

练习（3分钟）

- 列 Case：把大象装进冰箱。

极客

练功

之

TDD

实战：FizzBuzz

Practice: FizzBuzz

需求

写一个程序打印1到100这些数字：

但是遇到数字为3的倍数的时候，打印 “Fizz” 替代数字，

5的倍数用 “Buzz” 代替，

既是3的倍数又是5的倍数打印 “FizzBuzz” 。

Showcase

- Task 列表
- 测试
- 实现代码

需求 2

遇到包含3的数字打印 “Fizz”

遇到包含5的数字打印 “Buzz”

Showcase

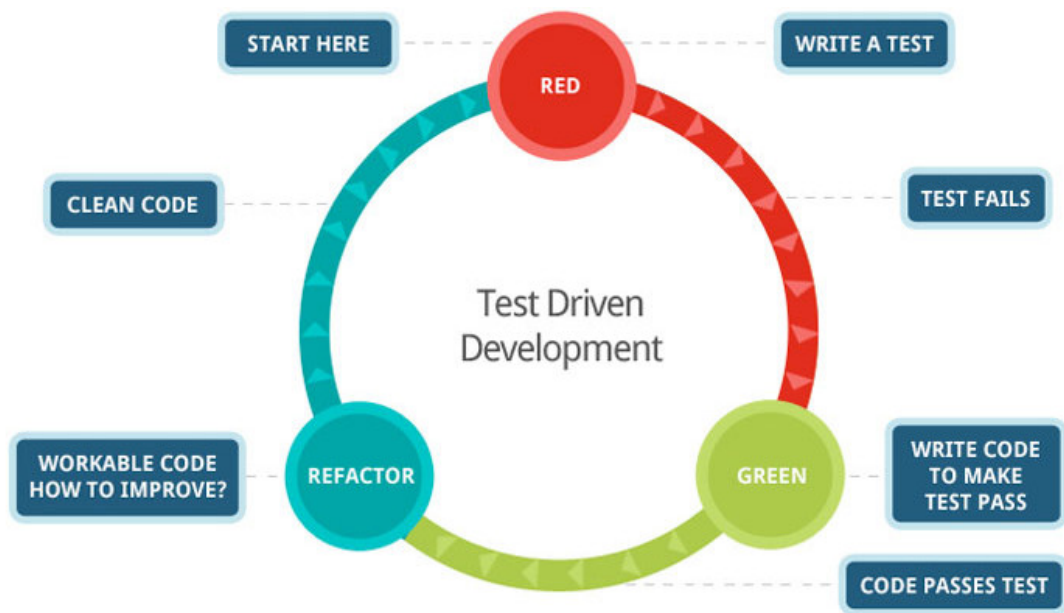
- Task 列表
- 测试
- 实现代码

Recap

- TDD循环 / 步骤
- TDD纪律
- 任务分解要点
- 单元测试三段式

红-绿-重构：TDD循环

1. 快速新增一个测试
2. 运行所有的测试，发现新增的测试不能通过
3. 用最简单的代码让测试程序可运行
4. 运行所有的测试，并且全部通过
5. 重构代码，以消除重复代码，优化设计结构



TDD 纪律

1. 没有失败的测试就不能写任何功能代码
2. 每次只允许写一个刚好失败的测试
3. 只允许写「刚好让失败的测试通过」的功能代码

TDD 实践

任务分解要点：

1. 各自独立
2. 完全穷尽

单元测试三段式：

- Given
- When
- Then

极客

练功

之

TDD

实战：停车场

Practice: Parking Lot

需求

停车场（Parking Lot）可以停车、取车；

不同的停车小哥（Parking Boy）能够依次或者按优先级（空置率、空闲车位数）在多个停车场中停放车辆；

停车经理（Parking Manger）指挥多个停车小哥，也可以自己去停车。

Showcase

- Task 列表
- 测试
- 实现代码

需求

- 停车场（Parking Lot）可以停车、取车

Showcase

- Task 列表
- 测试
- 实现代码

需求

- 停车小哥 (Parking Boy) 能够依次在多个停车场停放车辆

Showcase

- Task 列表
- 测试
- 实现代码

需求

- 机智的 (Smart) 停车小哥 (Parking Boy) 能够优先将车停在空位最多的停车场

Showcase

- Task 列表
- 测试
- 实现代码

需求

- 超级（Super）停车小哥（Parking Boy）能够优先将车停在空置率最高的停车场

Showcase

- Task 列表
- 测试
- 实现代码

需求

- 停车经理 (Parking Manger) 指挥多个停车小哥, 也可以自己去停车

Showcase

- Task 列表
- 测试
- 实现代码

Recap (again)

- TDD 是什么？
- TDD 解决了什么问题？
- TDD 怎么做？

Retro Time

每张sticker只写一件事情

- 收获了什么
- 存在的疑问