



# 测试驱动开发

厦门大学信息学院 赵江声

2023-11

# 目录

Content

01

基本概念

概念

02

演练：通过测试资源管理器进行  
测试驱动开发

单击此处添加文本具体内容

03

单击此处添加标题

单击此处添加文本具体内容



01

# 基本概念

简要介绍



## 1.1 什么是测试驱动开发

参考资料：

[https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E9%A9%B1%E5%8A%A8%E5%BC%80%E5%8F%91/3328831?](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E9%A9%B1%E5%8A%A8%E5%BC%80%E5%8F%91/3328831?fr=ge_al)

[fr=ge\\_al](#)

测试驱动开发，英文全称Test-Driven Development，简称TDD，是一种不同于传统软件开发流程的新型的开发方法。它要求在编写某个功能的代码之前先编写测试代码，然后只编写使测试通过的功能代码，通过测试来推动整个开发的进行。这有助于编写简洁可用和高质量的代码，并加速开发过程。

Kent Beck先生最早在其极限编程（XP）方法论中，向大家推荐“测试驱动”这一最佳实践，还专门撰写了《测试驱动开发》一书，详细说明如何实现。经过几年的迅猛发展，测试驱动开发已经成长为一门独立的软件开发技术，其名气甚至盖过了极限编程。



## 1.2 基本原理

- 测试驱动开发的基本思想就是在开发功能代码之前，先编写测试代码，然后只编写使测试通过的功能代码，从而以测试来驱动整个开发过程的进行。
- 这有助于编写简洁可用和高质量的代码，有很高的灵活性和健壮性，能快速响应变化，并加速开发过程。
- **不可运行/可运行/重构**——测试驱动开发的口号

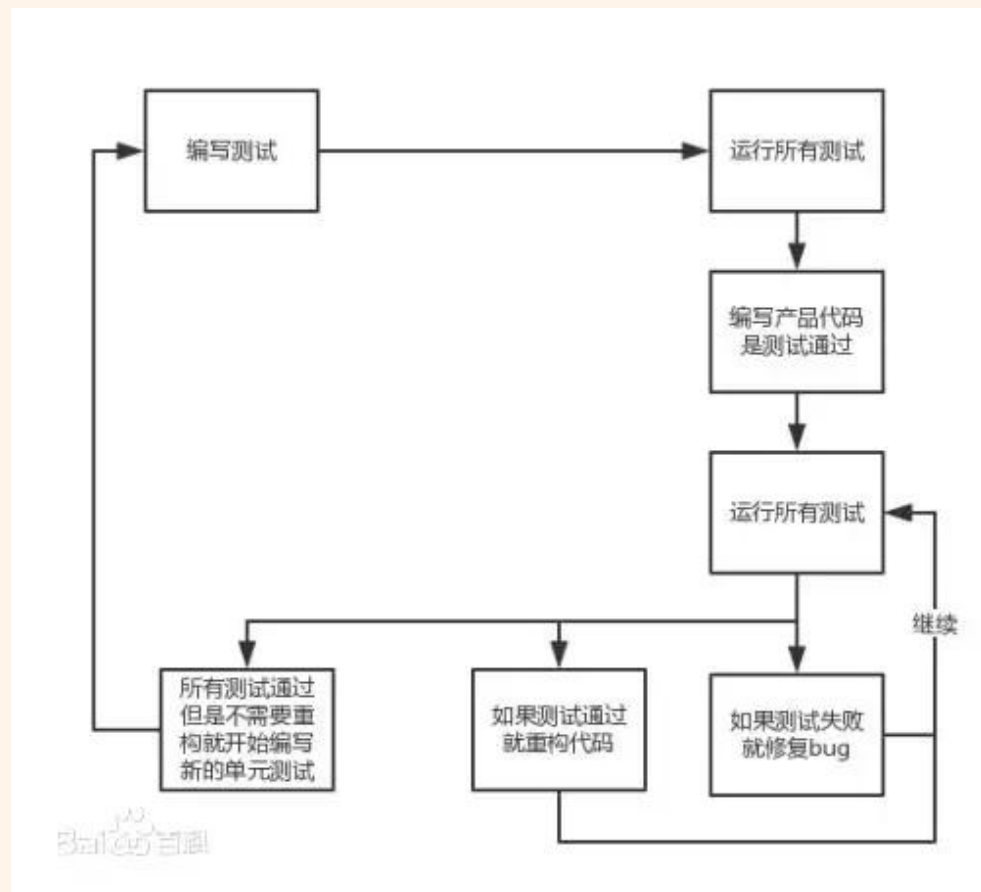


## 1.3 测试驱动开发的基本过程

测试驱动开发的基本过程如下：

- ① 快速新增一个测试
- ② 运行所有的测试（有时候只需要运行一个或一部分），发现新增的测试不能通过
- ③ 做一些小小的改动，尽快地让测试程序可运行，为此可以在程序中使用一些不合情理的方法
- ④ 运行所有的测试，并且全部通过
- ⑤ 重构代码，以消除重复设计，优化设计结构

简单来说，就是不可运行/可运行/重构——这正是测试驱动开发的口号。



## 1.4 优势

相对于传统的结构化开发过程方法，它具有以下优势：

1. TDD根据客户需求编写测试用例，对功能的过程和接口都进行了设计，而且这种从使用者角度对代码进行的设计通常更符合后期开发的需求。因为关注用户反馈，可以及时响应需求变更，同时因为从使用者角度出发的简单设计，也可以更快地适应变化。
2. 出于易测试和测试独立性的要求，将促使我们实现松耦合的设计，并更多地依赖于接口而非具体的类，提高系统的可扩展性和抗变性。而且TDD明显地缩短了设计决策的反馈循环，使我们几秒或几分钟之内就能获得反馈。
3. 将测试工作提到编码之前，并频繁地运行所有测试，可以尽量地避免和尽早地发现错误，极大地降低了后续测试及修复的成本，提高了代码的质量。在测试的保护下，不断重构代码，以消除重复设计，优化设计结构，提高了代码的重用性，从而提高了软件产品的质量。
4. TDD提供了持续的回归测试，使我们拥有重构的勇气，因为代码的改动导致系统其他部分产生任何异常，测试都会立刻通知我们。完整的测试会帮助我们持续地跟踪整个系统的状态，因此我们就不需要担心会产生什么不可预知的副作用了。
5. TDD所产生的单元测试代码就是最比较好的开发者文档，它们展示了所有的API该如何使用以及如何运作的，而且它们与工作代码保持同步，永远是最新的。
6. TDD可以减轻压力、降低忧虑、提高我们对代码的信心、使我们拥有重构的勇气，这些都是快乐工作的重要前提。
7. 快速的提高开发效率。





## 1.5 讨论

- 正面评价

可以有效的避免过度设计带来的浪费。但是也有人强调在开发前需要有完整的设计再实施可以有效的避免重构带来的浪费。

可以让开发者在开发中拥有更全面的视角。

- 负面评价

开发者可能只完成满足了测试的代码，而忽略了对实际需求的实现。有实践者认为用结对编程的方式可以有效的避免这个问题。

会放慢开发实际代码的速度，特别对于要求开发速度的原型开发造成不利。这里需要考虑开发速度需要包含功能和品质两个方面，单纯的代码速度可能不能完全代表开发速度。

对于GUI,资料库和Web应用而言。构造单元测试比较困难，如果强行构造单元测试，反而给维护带来额外的工作量。

有开发者认为这个是由于设计方法，而不是开发方法造成的困难。

使得开发更为关注用例和测试案例，而不是设计本身。当前对于这个观点有较多的争议。

测试驱动开发会导致单元测试的覆盖度不够，比如可能缺乏边界测试。在实际的操作中，和非测试驱动开发一样，当代码完成以后还是需要补充单元测试，提高测试的覆盖度。





## 02

演练：通过测试资源  
管理器进行测试驱动  
开发



## 2.1 基本概念

- 参考资料：<https://learn.microsoft.com/zh-cn/visualstudio/test/quick-start-test-driven-development-with-test-explorer?view=vs-2022>





谢谢！