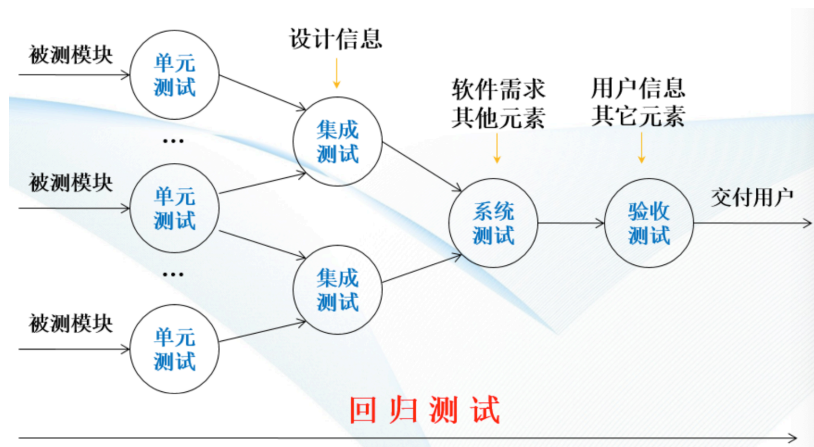


第四章 软件测试过程

1. 软件测试过程

四个测试阶段

- 单元测试 ⇒ 对软件中的最小被测模块进行测试的过程
- 集成测试 ⇒ 把多个单元模块组装在一起进行测试的过程。（Integration Testing）
- 系统测试 ⇒ 对已经集成之后的完整的软件系统进行测试的过程。（System Testing）
- 验收测试 ⇒ 模拟用户的实际环境，在交付之前对软件进行的最后一次测试过程。
- 回归测试 ⇒ 开发修改旧代码后，测试人员重新进行测试的过程。



1.1. 单元测试

单元：软件设计的最小单位，又叫模块，规定的最小被测功能模块

测试对象：最小可测单元或基本组成单元

时间：最小的单元模块编写完成、进行代码检查后，集成测试前

人员：大多数由开发人员测，采用互测/走查/评审等方式

测试类型：白盒测试

与开发的对应关系：在详细设计阶段进行单元测试准备工作

功能：是其它类型测试的基础

目的

- 验证代码是否与设计相符
- 发现在编码过程中引入的错误

意义

- 程序员：可以自信交付代码
 - 项目：尽早发现错误，减少成本
-

输入

- 《软件需求规格说明书》
 - 《软件详细设计说明书》
-

输出

- 单元测试用例
 - 代码静态检查记录
 - 问题记录与追踪（缺陷管理系统）
 - 软件代码开发版本
-

辅助模块

- 驱动模块：模拟被测试模块上一级模块
 - 桩模块：模拟被测模块工作过程中调用的模块
-

1.2. 集成测试

又叫组装测试

前提：假设各个软件单元已经通过了单元测试

集成：把多个单元组合起来形成的更大的单元

测试对象：各单元之间的接口以及集成后的功能

时间：单元测试之后，系统测试之前

人员：开发/测试

测试类型：灰盒

与开发的对应关系：开发的概要设计阶段，进行集成测试准备工作

层次

- 类内集成测试
 - 类间集成测试
-

关注点

- 模块间的**数据传递**是否正确
- 模块间是否存在**资源竞争问题**
- 模块**组合后的功能**是否满足要求
- 新增模块是否**影响其他模块的功能**

测试方式

- 体系结构分析
 - 分析子系统间依赖关系，确定集成测试的粒度
- 模块分析
 - 确定本次要测试的模块
 - 找出与该模块相关的所有模块，按优先级进行排练
 - 从高优先级开始集成
- 接口分析
- 风险分析
 - 识别 \implies 评估 \implies 规划 \implies 控制
- 集成测试策略分析
 - 根据被测对象选择合适的集成测试策略

集成测试原则

- 所有公共接口必须被测试到;
- 关键模块必须进行充分测试;
- 集成测试应当按一定层次进行;
- 集成测试策略选择要综合考虑质量、成本和进度三者之间的关系
- 集成测试应当尽早开始，并以概要设计为基础;
- 在模块和接口的划分上，测试人员应和开发人员进行充分沟通
- 当接口发生修改时，涉及到的相关接口都必须进行回归测试
- 测试执行结果应当如实的记录。

策略

在对测试对象分析的基础上，描述软件单元集成的方式和方法，是各种活动基础

基于分解的集成策略（最常用）：

- 非增量式集成
 - 一次性集成（大棒集成）
 - 把所有模块进行单元测试之后，一次性集成在一起进行测试。
 - 目的：短时间组装系统，使用最少测试来验证整个系统
 - 优点：节省时间
 - 缺点：出现问题不好定位
 - 适用范围：对于维护型项目，产品比较稳定的项目，为了节省时间，可以采用这种一次性集成的方式。
- 增量式集成：
 - 自顶向下集成
 - 深度：纵向集成
 - 广度：横向集成
- 自底向上集成
- 混合式集成

1.3. 系统测试

测试对象：产品系统的软件、硬件、外设、数据、支持软件等

时间：集成测试之后，验收测试之前

与开发的关系：开发过程的需求分析阶段，就可以进行测试要点的分析等相关工作，为系统测试做准备。

概念：对集成好的系统进行测试，目的是验证软件的功能和性能是不是满足需求。

人员：测试人员

类型：黑盒测试

系统测试类型

1) **功能测试：**系统测试的最基本类型，属于黑盒测试。

2) **性能测试：**对软件的性能指标进行测试的过程，目的是检测软件在实际的使用环境中能不能稳定可靠运行。

常用性能指标：

- 响应时间：我发出请求，到客户端返回结果所用的时间；
- 并发用户数：同一时间段内访问系统的用户数量
- 吞吐量：单位时间内系统处理的用户请求数量

3) **压力测试：**在软件超负荷运行时，查看系统的运行情况的测试。目的是找到软件的最大承受点在哪。

4) **安全性测试：**检测系统对于外部的非法侵入的抵御能力，目的是检查软件是否存在安全漏洞。

5) **兼容性测试：**在不同的软硬件平台/操作系统/浏览器等环境下测试同一个软件的过程。目的是验证软件能否在各种环境下正常运行。

分类：

1. 操作系统兼容性
2. 浏览器兼容性
3. 同一软件不同版本的兼容性

6) **恢复性测试：**测试系统的恢复能力，看系统有没有自动恢复状态和数据的能力。

7) **安装测试：**测试软件成功安装的能力：安装/卸载/升级

8) **GUI测试：**测试软件的界面是否美观，按钮是否可用等（界面测试）

9) **文档测试：**对软件实施过程中的各种文档进行测试的过程。

1.4. 验收测试

又叫交付测试，系统测试之后，把用户的相关信息和用户环境加入进来，进行最后一轮测试的过程。

验收测试类型：

- α 测试：在公司模拟用户的实际环境进行测试
- β 测试：在用户现场，由用户参与实施测试的过程

1.5. 回归测试

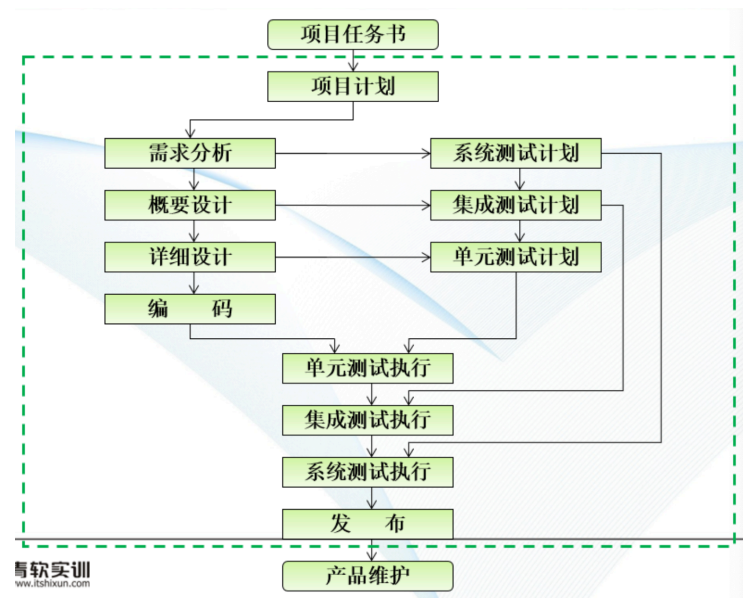
回归测试

开发修改旧代码后，测试人员重新进行测试确定修改没有引入新的错误或导致其它代码产生错误的过程。

- 软件开发的各个阶段都会进行多次回归测试
- 1) 验证缺陷是否修复
- 2) 验证修复了该缺陷，其他关联模块有没有产生新缺陷

2. 软件测试过程模型

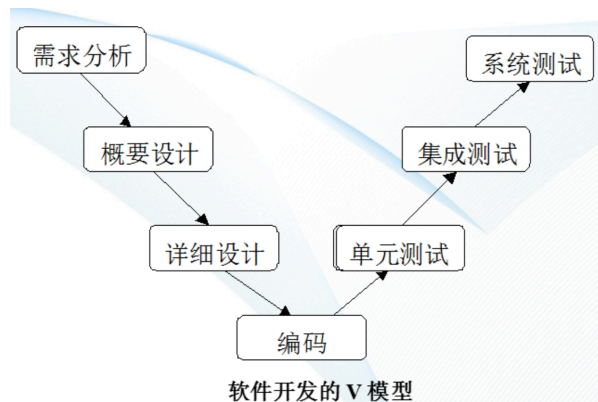
- 软件测试过程



- 瀑布模型

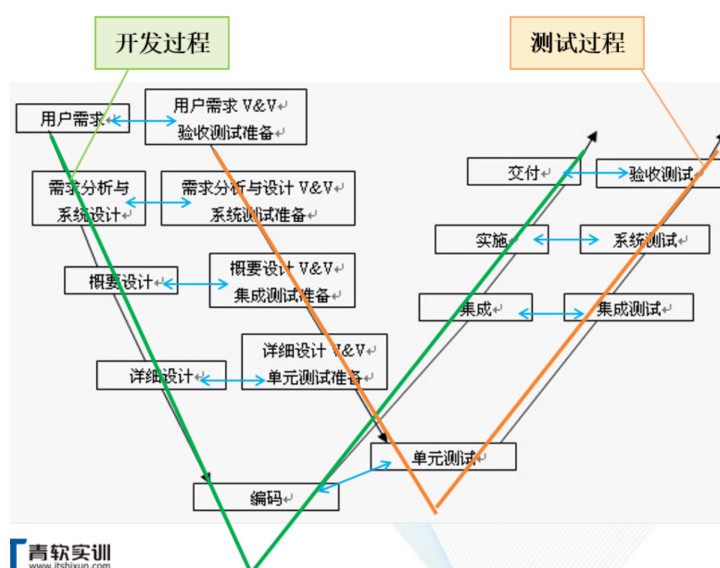
从上而下按部就班地执行工作，测试在编码之后才能进行；

- V 模型



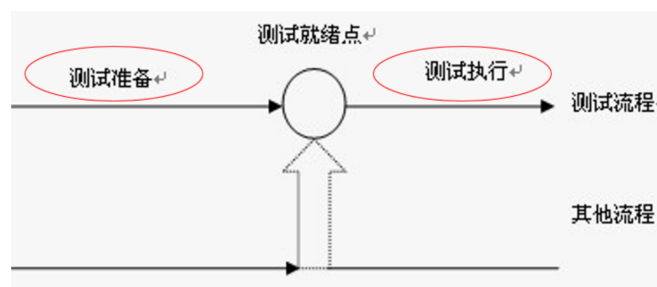
把测试按测试阶段进行了细分，而且有了开发过程和测试过程的区分，但是测试依然在编码之后进行，没有体现**尽早/不断测试**

• W 模型



体现了开发过程和测试过程的并行关系，但是依然要等上一阶段完全做完才开始下一个阶段工作。

• H 模型



测试就绪点满足之后，就进行测试，多条测试流程可以并行进行。太过于模型化了，没有太大使用价值。

3. 软件测试过程管理

测试过程三大元素

- 测试相关人员
 - 测试过程分解
 - 测试工作产品
-

测试过程的主要工作产品有

- 测试计划
 - 测试用例
 - 缺陷报告
 - 测试总结报告
 - 测试日志
-

测试过程管理原则

- 测试需求明确
- 测试计划先行
- 建立任务优先级
- 尽早测试
- 全面测试
- 全过程测试
- 独立的、迭代的测试