

Ch6 单元测试

2018年12月11日 12:28

6.1什么是单元测试：

- 按阶段进行测试是一种基本的测试策略，单元测试是执行过程中的第一阶段。
- 单元测试是软件开发过程中最小的可测试的应用程序的某些部分，**称为单元**，是单独和独立的scrutinized(审查)正确操作。
 - **单元**是软件设计的**最小单元**
 - 软件单元必须与程序的其他部分分开，并**独立测试**。
 - 一个单元，一个特定的函数或类方法，以及一个模块或一组模块
 - 在Java中，单元是**类或类方法**。
 - 在C语言中，单元是**一个函数或子进程**
- 一般情况下，被测单元能够实现一个特定的功能，并与其他单元有明确的接口定义，这样才可以与其他单元隔离开来
- 单元测试的**目的**就是**确保模块被正确的编码**。使用详细的规格说明作为指南，对重要控制路径进行测试以发现模块内的错误。

目标：确保模块被正确的编码

依据：系统详细的规格说明

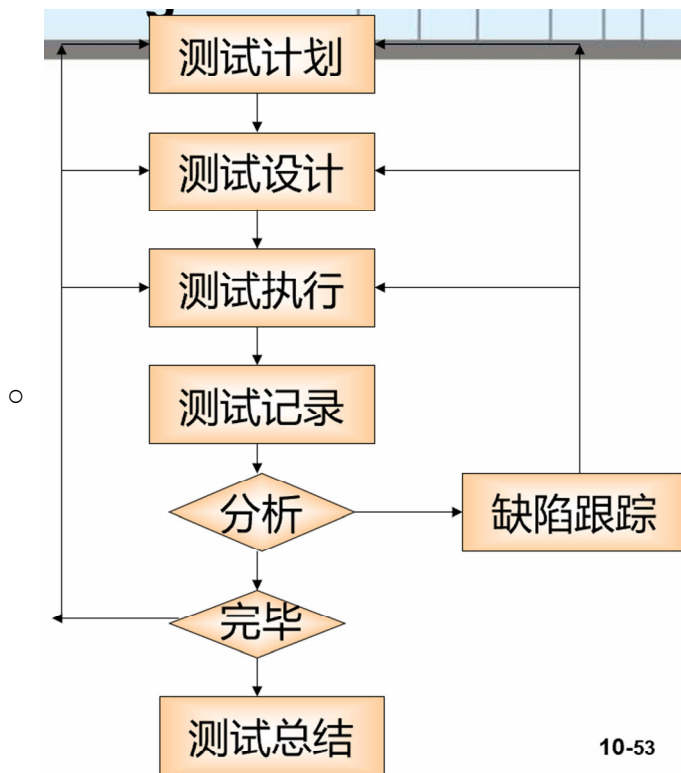
过程：经过设计、脚本开发、执行、调试和分析结果一个过程

执行者：由程序开发人员和测试人员共同完成

方法：以**白盒测试方法**为主，辅以黑盒测试方法

如何进行评估：通过所有单元测试用例，代码没有严重缺陷

- 单元测试的过程：
 - 1、在详细设计阶段完成单元测试计划
 - 2、建立单元测试环境，完成测试设计和开发
 - 3、执行单元测试用例，并且详细记录测试结果
 - 4、判定单元测试是否通过
 - 5、提交单元测试报告



- 单元测试的好处：

- 1、单独进行，一起进行，降低软件质量成本，缩短开发周期；
- 2、便于跟踪错误；
- 3、集成后错误会放大，集成后复杂性高，很难发现问题；
- 4、无需而外的设备和人员。

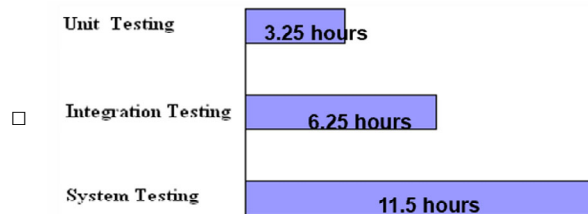
- 单元测试的分类：

- 静态测试

- 它主要是对代码进行语法检查和/或手动检查代码或文档以查找错误。
 - 编写代码的开发人员可以单独使用这种类型的测试。
 - 编码标准和规格说明：如C++编码规范
 - 代码评审review：
 - 走查Walkthroughs
 - （正式会议）审查inspections

项目	走查	审查
准备	通读设计和编码	应准备好需求描述文档、程序设计文档、程序源代码清单、代码编码标准和代码缺陷检查表
形式	非正式会议	正式会议
参加人员	开发人员为主	项目组成员包括测试人员
主要技术方法	无	缺陷检查表
注意事项	限时、不要现场修改代码	限时、不要现场修改代码
生成文档	会议记录	静态分析错误报告
目标	代码标准规范、无逻辑错误	代码标准规范、无逻辑错误

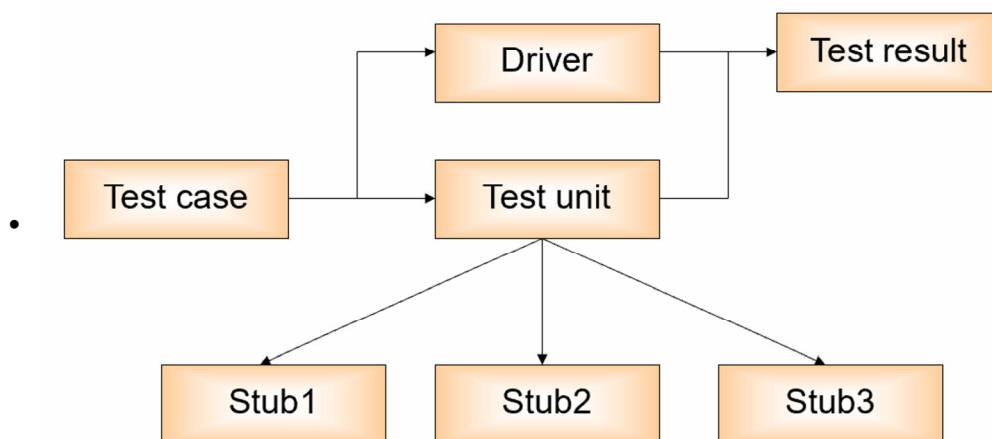
- 在不同的测试阶段内发现的一个缺陷所需要修复的时间



- 当代码评审没有问题，再进行动态测试
 - 动态测试

6.2单元测试的策略

- 存根/桩：Stub（被测模块的下层）
 - 用于模拟测试模块工作过程中的调用模块。通常它们只处理很少的数据。
- 驱动模块：Driver Module（被测模块的上层）
 - 用于模拟被测模块的上级模块。接收测试数据，传送测试模块相关信息，启动测试模块，打印测试结果。



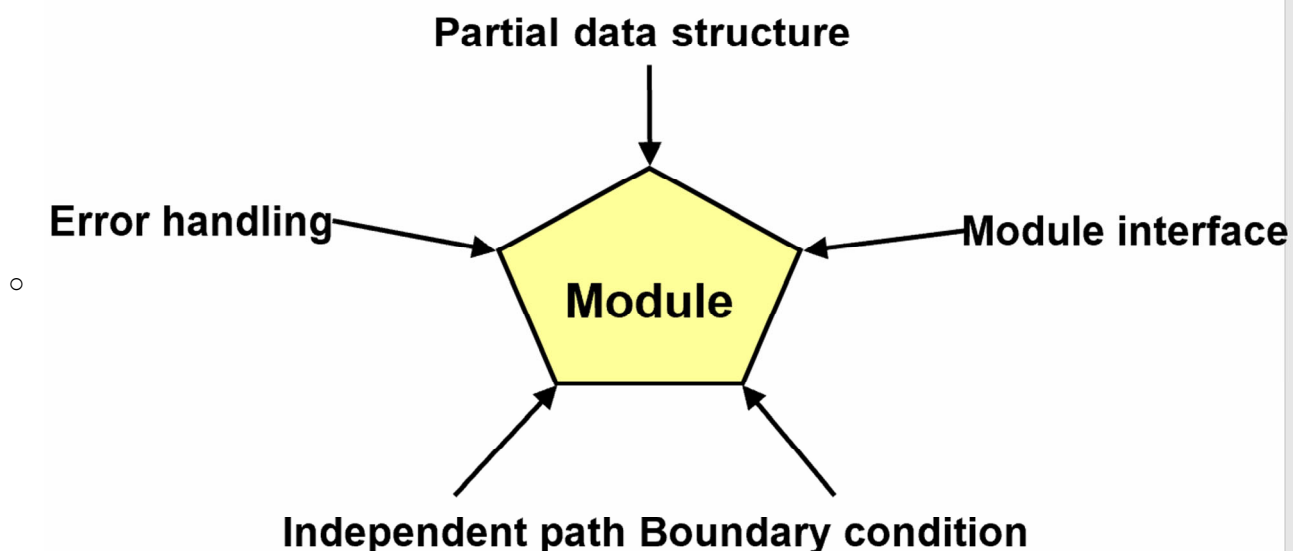
- 自顶向下的单元测试
 - 测试顶级首先,单元由顶层调用存根,然后测试第二层次,作为驱动模块的单元测试顺

序。

- 自底向上的单元测试
 - 首先对底层模块进行测试，为其设置驱动模块，然后对上层模块进行测试，将已测试模块依次作为存根进行测试。
- 隔离测试（最方便，减少了对其他单元的依赖）
 - 不考虑模块之间的关系，设计每个模块的子模块和驱动模块。

6.3单元测试分析

- 单元测试阶段关注的5个方面：



6.4单元测试准则

- 注意控制单元测试的范围，不要把单元测试变成集成测试
1. 保持单元测试小而快速
 2. 单元测试应该是完全自动化和**非交互式的**（没有用户输入），一旦触发，自动运行直到得到比对结果
 3. 使单元测试易于运行
 4. 测量测试
 5. 立即修复失败的测试
 6. 保持单元级别的测试，不要扩大到集成测试
 7. 简单的开始
 8. 保持测试独立
 9. 将测试保持在被测试的类附近
 10. 适当的命名测试
 11. 测试公共API
 12. 考虑黑盒测试
 13. 考虑白盒测试
 14. 测试一些简单的情况
 15. 首先关注执行范围
 16. 覆盖边界情况
 17. 提供一个随机生成器
 18. 测试每个特性一次

19. 使用明确的断言
20. 提供负面测试
21. 设计代码时要考虑测试
22. 不要连接到预定义的外部资源
23. 了解测试的成本
24. 优先测试
25. 为失败准备测试代码
26. 编写测试来重现
27. 知道的限制

6.5怎样使用JUnit

- 单元测试的工具
 - 单元测试一般采用白盒测试的方法。
 - 单元测试工具根据不同的语言有对应的版本。
 - JUnit:java
 - CppUnit:C++