

单元测试说明

本文档是对单元测试的流程和相关概念进行说明。

单元测试步骤

1. 步骤一：制定单元测试计划
2. 步骤二：设计单元测试用例
3. 步骤三：搭建单元测试环境
4. 步骤四：执行单元测试
5. 步骤五：撰写单元测试报告

制定单元测试计划

1. 内容
 - a) 目的
 - b) 测试方法
 - c) 测试范围
 - d) 测试交付件
 - e) 测试过程准则
 - f) 工作任务分布
 - g) 测试进度
 - h) 测试资源
 - i) 测试用例结构及其用例
 - j) 测试结论约定
2. 测试方法

根据项目要求和被测单元特征，指明在本次单元测试中所采用的发现缺陷的技术，例如常规的白盒测试、黑盒测试、自动化测试或者复用类似的测试等。
3. 测试范围

明确此次单元测试“做什么”和“不做什么”，依据项目安排测试哪些单元，每个单元需要测试哪些内容。
4. 测试范围

按照常规观点，围绕单元的设计功能，单元测试常需要包括单元的接口测试、局部数据结构测试、边界条件测试、所有独立执行通路测试和各条错误处理测试等几大方面。
5. 测试交付件
 - a) 《XXXX 单元测试计划》
 - b) 《XXXX 单元测试计划评审报告》
 - c) 《XXXX 单元测试用例》
 - d) 《XXXX 单元测试用例评审报告》
 - e) 《XXXX 单元测试脚本》
 - f) 《XXXX 单元测试脚本评审报告》

- g) 《XXXX 单元测试报告》
- h) 《XXXX 单元测试报告评审报告》
- i) 《缺陷问题单》若干

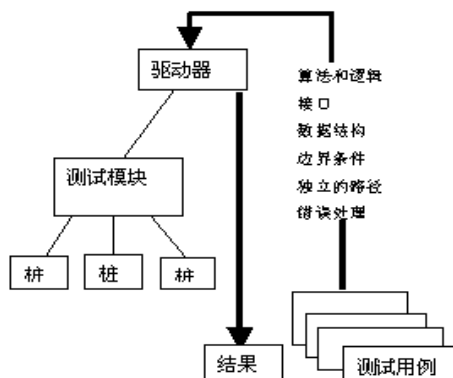
设计单元测试用例

主要任务是根据各项测试需求确定单元测试方案，包括

1. 测试所依据的标准和文档
2. 测试使用的方法例如白盒、黑盒或其它
3. 缺陷属性的说明
4. 结论的约定等。
5. 如果需要编写测试代码或测试工具还需准备测试代码与工具的设计描述

搭建单元测试环境

1. 执行单元测试的软硬件环境
2. 待测单元
3. 单元测试用例
4. 因为单元是整个系统的一部分，它不能单独运行。为了执行单元测试用例需要开发如下的测试代码，一般用如下的图表示：



- a) 驱动：模拟调用函数的一段代码，它可以替代调用被测单元的模块
- b) 桩：模拟被测单元所调用函数的一段代码，它可以替代被被测单元调用的模块

5. 单元测试工具：JUnit/TestNG
6. 持续集成工具：Ant/maven

单元测试实现

依据规范开发单元测试用例并确保满足测试需求，测试用例可以是手工测试用例，也可以是自动化测试脚本。

单元测试执行

主要任务是搭建测试环境，运行测试用例以发现被测单元中的缺陷，当发现缺陷后提交缺陷问题报告单并在缺陷修复后对缺陷的修正进行验证。

单元测试报告

对测试过程进行总结，提供相关测试数据说明和缺陷说明，评价被测对象并给出改进意见，输出《软件单元测试报告》。

其它

1. 单元测试中还有一些辅助性但也非常重要的活动
 - a) 进行需求跟踪以验证分配到该软件单元的需求是否已完全实现
 - b) 跟踪和解决单元测试缺陷
 - c) 更新用户文档
 - d) 阶段评审
 - e) 单元过程资产基线
 - f) 编写任务总结报告等
2. 指明了单元测试的过程，明确此次单元测试的目的

下面的建议可能很有帮助

1. 单元测试前执行代码检视
2. 由开发被测单元的程序员执行单元测试
3. 完成代码后修改单元测试用例
4. 记得测试运行时问题
5. 测试程序越简单越好
6. 开发测试程序编程规范
7. 在单元测试时使用裁减的缺陷跟踪流程
8. 确保修改缺陷时不引入新的问题
9. 报告单元测试缺陷
10. 保存单元缺陷报告

判定标准

单元测试覆盖率达到 40% 算合格。