# 软件测试基础

1. 什么是软件测试?

设计一批测试用例(输入数据，验证预期的输出结果)，运行程序发现程序错误的过程。

1. “软件测试流程”是什么？

测试计划: 根据用户需求, 规格说明书，定义相应的测试需求报告。

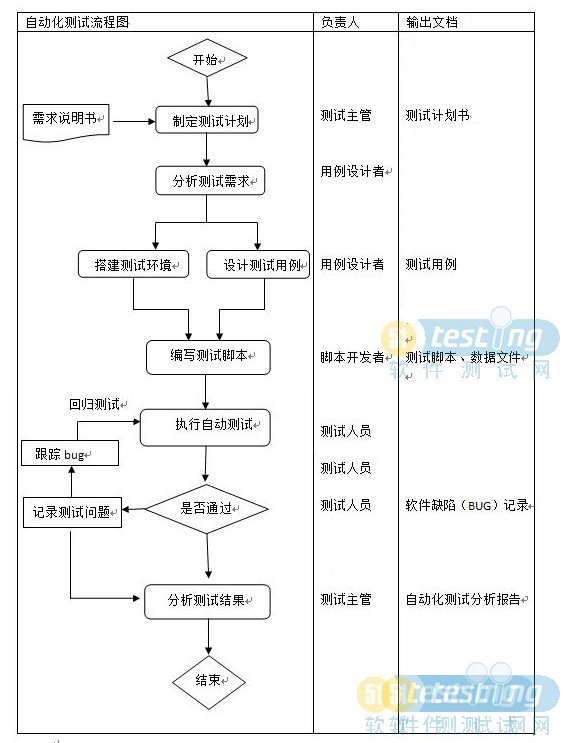
测试设计: 将测试需求分解为若干个可执行的测试过程(设计测试用例)。

测试开发: 建立自动化测试过程。

测试执行: 测试执行一般由单元测试、集成测试、系统测试、回归测试、验收测试 等组成。

测试评估: 评估测试报告。

1. “自动化测试流程”是什么？



1. 软件生存周期是什么？

1.需求分析

2.软件设计

3.编码

4.软件测试

5.运行与维护

1. 什么是V模型？（一套完整的测试应该由哪些阶段组成）

需求分析                             验收测试

    概要设计                      系统测试

        详细设计              集成测试

             编码           单元测试

                         V

1. 什么是测试用例 ？

向被测试系统提供的输入数据、操作以及期望结果的一个的集合。

1. 软件测试类型都有哪些？

按测试策略分类：1、静态与动态测试2、黑盒与白盒测试 3、手工和自动测试 4、冒烟测试（针对具体bug 覆 盖率低） 5、回归测试；

按测试阶段分类：单元测试、集成测试、系统测试，验收测试

其他常见测试方法：1、功能测试 2、安全性测试 3、性能测试 （压力测试，负载测试） 4、易用性测试 5、 安装测试 6、界面测试 7、兼容性测试 8、健壮性测试

1. “测试用例”设计方法有哪些？

黑盒：

1) 等价类划分: 有效等价类，无效等价类。划分互不相交的子集。（比如测一个电话项的输入）

2) 边界值分析: 错误很容易出现在边界上。（白黑都可以用。1,2合并叫等值分析法。）

3) 错误推测法: 根据经验或直觉推测程序中可能存在的各种错误。

4) 判定表法: 策略表，严谨。通过穷举条件获得结果，对结果再进行优化合并。

例子：C1&(c2|C3) => yes (比如1和2可以优化合并)



1. 正交法：缩减一些用例，达到尽量少的用例 覆盖 尽量大的范围。

白盒：

逻辑覆盖（6种覆盖），循环覆盖（代码里有循环，嵌套循环）

1. 白盒测试的“六种覆盖方法”是什么？

以下由弱到强。

1）语句覆盖：每个语句至少被执行一次；

2）判定覆盖：所有判定可能（yes或no）被执行一次；

3）条件覆盖：判定中的每个条件(true or false)都获得各种可能；

4）判定/条件覆盖：判定中的每个条件取到各种可能的值，并使每个判定取到各种可能的结果；

5）条件组合覆盖：每个判定中条件的各种可能“组合”都至少出现一次；

6）路径覆盖：包括所有的可能性。

1. 什么是“驱动函数”“桩函数”?

1) 驱动函数: 在一个main中构造一点实验数据传入，然后调用被测试函数，检验测试结果。

2) 桩函数:（打桩）被测函数可能要调用一个未被实现的函数，测试者写个临时的模拟的函数去顶替。

1. 什么是α测试?什么是β测试?

都属于用户验收测试，不能由程序员或测试员完成。

1. α测试：功能性，可靠性。公司内部进行内部体验测试，一般在编码结束之后开始。
2. β测试： α 测试已经可靠了，大部分bug已经修复。上市之前让用户或者企业合作伙伴用户体验测试。
3. Γ 测试：市场测试。
4. 什么是“黑盒测试”？“白盒测试”？

1)黑盒测试:

a.什么是？只关心输入后会得出怎么样的输出结果，验证软件是否符合需求文档。把软件看成一个黑盒， 不必深入了解其内部.

b.哪里用？集成测试。系统测试。验收测试。

c.优点：简单，直观。 缺点：代码覆盖率低。

2)白盒测试

a.什么是？要求测试人员了解开发人员的代码实现，了解内部逻辑。

b.怎么做？使用多种逻辑覆盖（6种）

c.哪里用？一般用于单元测试。

d..优点：代码覆盖率高。 缺点：专业人员，测试成本大。

1. 什么是“单元测试”？
2. 简介：一般为开发人员做的白盒测试，在软件开发的初始阶段发现软件bug，降低开发成本。

2) 时间？ 编码阶段（设计好接口）。有的甚至要先编写测试代码，再进行软件开发。

3) 怎么做？（策略）针对开发人员做好的接口，对比input和output。编写驱动模块（main），编写桩模块。

4) 侧重点？接口的正确。

1. 什么是“集成测试”？

1)简介：发现引进新的组件所引起的bug。

2)时间？ 各单元编码完成，集成到一个组件进行测试。

3)怎么做？（策略）

a.自顶向下(需要写桩模块)：main开始，较早接触主要功能模块，不需要写驱动。

b.自底向上(需要驱动模块)：较早验证基础模块，不需要写桩模块。

c.大爆炸集成法：所有模块放到一起集成测试。

4) 侧重点? 模块的衔接。

1. 什么是“系统测试”？

1.简介：模拟真实测试环境测试。

2时间？ 所有模块集成之后。

3.怎么做？（策略）6。功能测试，性能测试（压力测试, 负载测试）,安全性测试, 可用性测试，健壮性测试（突然关机），兼容性测试（新旧版本），安装测试。

4.侧重点？整个系统的运行，和其他软件兼容。

1. 什么是“回归测试”？

1.简介：验证是不是引入了新的bug，新系统是否稳定，验证此版本是否能成为新的基线版本。

2.时间？基线版本做出改变（比如修复bug，加入新需求）

3.怎么做？（策略）沿用上一基线版本的测试用例，针对新功能加入新的测试用例。

PS：基线版本（Baseline）：一个个稳定版本，作为继续在其上面开发的依据。如果不稳定，则需要rollback。

1. 什么是“验收测试”？

就是α测试和β测试。

1. Bug等级怎么划分？

严重性：1.致命错误（死机）2.严重错误（异常退出）3.一般错误（部分失效） 4.建议模块

优先级：1.P1（立即解决）2.P2（高优先级）3.P3（正常排队） 4.P4(有时间的时候修复)

1. Bugzilla管理流程是怎么样的？（看图）
2. 测试用例的内容 （看图）

用例编号|功能描述|步骤描述|期望结果|测试结果|前置条件

1. Bug的内容

Bug描述|等级|重现步骤|bug类型|程序版本

1. 测试计划的内容

主管写的。包含(背景、测试策略、测试环境、测试开始结束时间、测试人员安排、风险评估等。)

1. 什么是软件质量？

软件质量是软件符合“需求文档”，“功能和性能需求”程度。

质量特性：功能性，性能，安全性，可用性，健壮性，兼容性。

1. 测试人员在软件开发过程中的任务是什么？

尽可能早的找出系统中的Bug，确保软件的质量。

1. 做好测试计划的关键是什么？

5W

What：明确测试的范围和内容，

Why：理解测试的目的

When：确定测试的开始和结束日期

How：测试的方法和工具

Where：测试文档的存放位置

1. 您认为做好测试用例设计工作的关键是什么？

以较少的用例覆盖尽可能多路径，在单位的时间内发现最多的问题。