# 白盒测试深入理解与总结

## 白盒测试的用处/好处：

1. 可以设计与程序中某个模块或单元无关的测试用例。
2. 充分设计用例，使分支语句的True或false分支都得到测试。
3. 设计循环边界执行的用例。
4. 测试内部数据结构。

## 白盒测试的阶段：

在unit, integration, function/system, acceptance, regression,and beta六个测试阶段中，白盒测试可以用于unit,integration和regression三个阶段。

## Driver和stub

简单来说,driver负责调用，传参，监控待测试模块，stub用来暂时代替待测试模块中需要用到但尚未完成的部分，等完成后stub可以被替换回来。

## 设计测试用例

1. 基本路径测试

计算基本路径个数的方法：分支语句的个数+1即为基本路径数。

1. Equivalence Partitioning/Boundary Value Analysis

## Control-flow/Coverage Testing

1. Coverage:一种度量，衡量对流程的覆盖程度，具体有不同的覆盖测试形式。

3.1 Method Coverage

指测试样例应该完全调用所定义的方法或函数，简单来说就是如果定义了十个方法或函数，那么就最好在测试样例（或整个测试流程中）保证10个方法都被调用到，即覆盖率达到100%。

3.2 Statement Coverage

就是语句覆盖率，即要测试样例能保证所有的语句都被至少执行过一次。

3.3 Branch Coverage

即每个关键的分支语句处的布尔表达式（将所有and，or之后的结果看作一个整体）

3.4 Condition Coverage

对Branch Coverage进一步细化至子表达式，比如（（a==b）&&(c==d)）,要求a==b,和c==d的两个子表达式的true,false取值在测试样例中都被取过。

**4 Data Flow Testing 数据流测试**

**关注数据流向的测试。将一个引用注解用三元是(d,u,V)表示其中V代表某个变量，d代表被定义的地方,u代表被使用的某个节点。**

**5 Failure (“Dirty”) Test Cases 故障测试**

**指从用户使用角度，考虑某些突发或恶意而可能导致故障的情况。**

**比如用户输入处输入过大或是不符合格式的数据；**

**用户能否恶意造成程序安全问题，比如通过输入数据使缓冲区出现溢出；**

**注意每个计算点，比如有没有可能出现除以0，或是数据过大而溢出。**

**6 Flow Graphs Revisited**

**通过划分更细致的Flow graph,并计算**cyclomatic number，可以估计100%各种覆盖所需的测试用例数量下界。

**7 Summary**

**尽可能在单元测试阶段早进行测试，而不是拖到后期的系统测试，有助于排查和解决问题。**

**白盒测试与黑盒测试的对比：**

**白盒可以根据代码设计用例的输入和输出，黑盒只能凭文档或常识；**

**白盒测试可以反应故障（fault），黑盒测试只能反应失效（failure）.**