[白盒测试概述及其方法简介](http://blog.chinaunix.net/uid-21411227-id-1826810.html) 2010-03-16 17:00:26

分类： 项目管理

**引言：**

白盒测试作为测试人员常用的一种测试方法，越来越受到测试工程师的重视。白盒测试并不是简单的按照代码设计用例，而是需要根据不同的测试需求，结合不同的测试对象，使用适合的方法进行测试。因为对于不同复杂度的代码逻辑，可以衍生出许多种执行路径，只有适当的测试方法，才能帮助我们从代码的迷雾森林中找到正确的方向。

**1．白盒测试的概述**

由于逻辑错误和不正确假设与一条程序路径被运行的可能性成反比。由于我们经常相信某逻辑路径不可能被执行, 而事实上,它可能在正常的情况下被执行。由于代码中的笔误是随机且无法杜绝的，因此我们要进行白盒测试。

白盒测试又称结构测试,透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的,你清楚盒子内部的东西以及里面是如何运作的。

**1） 白盒的测试用例需要做到：**

·保证一个模块中的所有独立路径至少被使用一次

·对所有逻辑值均需测试 true 和 false

·在上下边界及可操作范围内运行所有循环

·检查内部数据结构以确保其有效性

**2）白盒测试的目的：**

通过检查软件内部的逻辑结构，对软件中的逻辑路径进行覆盖测试；在程序不同地方设立检查点，检查程序的状态，以确定实际运行状态与预期状态是否一致。

**3）白盒测试的特点：**

依据软件设计说明书进行测试、对程序内部细节的严密检验、针对特定条件设计测试用例、对软件的逻辑路径进行覆盖测试。

**4）白盒测试的实施步骤：**

测试计划阶段：根据需求说明书，制定测试进度。

测试设计阶段：依据程序设计说明书，按照一定规范化的方法进行软件结构划分和设计测试用例。

测试执行阶段：输入测试用例，得到测试结果。

测试总结阶段：对比测试的结果和代码的预期结果，分析错误原因，找到并解决错误。

**5）白盒测试的方法：总体上分为静态方法和动态方法两大类。**

静态分析是一种不通过执行程序而进行测试的技术。静态分析的关键功能是检查软件的表示和描述是否一致,没有冲突或者没有歧义。

动态分析的主要特点是当软件系统在模拟的或真实的环境中执行之前、之中和之后 , 对软件系统行为的分析。动态分析包含了程序在受控的环境下使用特定的期望结果进行正式的运行。它显示了一个系统在检查状态下是正确还是不正确。在动态分析技术中,最重要的技术是路径和分支测试。下面要介绍的六种覆盖测试方法属于动态分析方法。

**6）白盒测试的优缺点**

 优点

·迫使测试人员去仔细思考软件的实现

·可以检测代码中的每条分支和路径

·揭示隐藏在代码中的错误

·对代码的测试比较彻底

·最优化

　缺点

·昂贵

·无法检测代码中遗漏的路径和数据敏感性错误

·不验证规格的正确性

**2．本文介绍六种白盒子测试方法：（强度由低到高）语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。**

1）所谓语句覆盖：就是设计若干个测试用例，运行被测程序，使得每一可执行语句至少执行一次。这里的“若干个”，意味着使用测试用例越少越好。语句覆盖率的公式可以表示如下：

语句覆盖率=被评价到的语句数量/可执行的语句总数 x 100%

2）判定覆盖：使设计的测试用例保证程序中每个判断的每个取值分支（t or f）至少经历一次

[优点]：判定覆盖具有比语句覆盖更强的测试能力，而且具有和语句覆盖一样的简单性，无需细分每个判定就可以得到测试用例。

[缺点]：往往大部分的判定语句是由多个逻辑条件组合而成（如，判定语句中包含AND、OR、CASE），若仅仅判断其整个最终结果，而忽略每个条件的取值情况，必然会遗漏部分测试路径。

　　例如:

　　int a,b;

　　if(a || b)

　　执行语句1

　　else

　　执行语句2

要达到这段程序的判断覆盖,我们采用测试用例:1)a = true , b = true ;2)a = flase, b = flase

3）条件覆盖：条件覆盖是指选择足够的测试用例，使得运行这些测试用例时，判定中每个条件的所有可能结果至少出现一次，但未必能覆盖全部分支

条件覆盖要检查每个符合谓词的子表达式值为真和假两种情况，要独立衡量每个子表达式的结果，以确保每个子表达式的值为真和假两种情况都被测试到。

4） 判定条件覆盖：判定-条件覆盖就是设计足够的测试用例，使得判断中每个条件的所有可能取值至少执行一次，同时每个判断的所有可能判断结果至少执行，即要求各个判断的所有可能的条件取值组合至少执行一次。

5） 条件组合覆盖：在白盒测试法中，选择足够的测试用例，使所有判定中各条件判断结果的所有组合至少出现一次，满足这种覆盖标准成为条件组合覆盖。

6）路径覆盖：是每条可能执行到的路径至少执行一次；

说明：其中语句覆盖是一种最弱的覆盖，判定覆盖和条件覆盖比语句覆盖强，满足判定/条件覆盖标准的测试用例一定也满足判定覆盖、条件覆盖和语句覆盖，条件组合覆盖是除路径覆盖外最强的，路径覆盖也是一种比较强的覆盖，但未必考虑判定条件结果的组合，并不能代替条件覆盖和条件组合覆盖。

**举例：**

if A and B then Action1

if C or D then Action2

1）语句覆盖最弱，只需要让程序中的语句都执行一遍即可 。上例中只需设计测试用例使得A=true B=true C=true 即可。

2）分支覆盖又称判定覆盖：使得程序中每个判断的取真分支和取假分支至少经历一次，即判断的真假均曾被满足。上例需要设计测试用例使其分别满足下列条件即可（1）A=true，B=true，C＝true，D=false（2）A=true，B=false，C＝false，D=false。

3）条件覆盖：要使得每个判断中的每个条件的可能取值至少满足一次。上例中第一个判断应考虑到A=true，A=false，B=true，B=false第二个判断应考虑到C＝true，C＝false，D=true，D=false，所以上例中可以设计测试用例满足下列条件（1）A=true，B=true，C＝true，D=true（2）A=false，B=false，C＝false，D=false。

4） 路径覆盖：要求覆盖程序中所有可能的路径。所以可以设计测试用例满足下列条件（1）A=true，B=true，C＝true，D=true（2）A=false，B=false，C＝false，D=false（3）A=true，B=true，C＝false，D=false（4）A=false，B=false，C＝true，D=true。