对比黑盒测试方法和白盒测试方法

一．概念分析：

 黑盒测试方法：黑盒测试方法，也称数据驱动测试方法，在测试时，把程序看成是一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，测试人员针对软件直接进行测试。如：检查系统功能是否按照需求规格说明书的规定正常使用、是否能适当地接受输入数据而输出正确的结果等，检查相应的模板是否采用了正确的模板、是否满足规范要求。

白盒测试方法：白盒测试也称结构化测试或逻辑驱动测试，也就是已知产品的内部工作过程，清楚最终软件产品的计算机程序结构及其语句，按照程序内部的结构测试程序，测试程序内部的变量状态、逻辑结构、运行路径等，检验程序中的每条通路是否能按预定要求正确工作，检查程序内部动作或运行是否符合设计规格要求，所有内部结构是否按规定正常运行。

二．方法对比分析：

①黑盒测试：

测试特点：测试功能；

测试依据：需求规格说明书

方法举例：等价类划分、边界值测试

优点：能站在用户的立场上进行测试

缺点：不能测试程序内部特定部位，如程序有误，则无法发现。

黑盒测试主要是为了发现以下错误：

1. 有错误的功能或遗漏了某项功能；
2. 不能正确地接收输入数据，输出错误地结果；
3. 功能操作逻辑不合理、不够方便；
4. 界面出错、扭曲或不美观；
5. 安装过程中出现问题，安装步骤不清晰、不够灵活；
6. 系统初始化问题等。

黑盒的测试用例技术设计有三种：  边界值分析、等价类划分、错误推测法。

黑盒测试有助于对被测软件产品进行总体功能验证，在进行黑盒测试时需要注意：

（1）黑盒测试基于需求实施。黑盒测试除了可以发现各种有关系统整体上的问题，还有发现不一致、不完备的需求。

（2）黑盒测试检查已描述的需求和隐含的需求。并不是所有需求都经过明确描述，有些需求是隐含的。例如，打印报表，有可能在需求规格说明书上没有明确给出报表上要给出日期。但是，这些功能在向客户交付产品时应该提供，以提高更好的可读性和可用性。

（3）黑盒测试要包括最终用户视角。由于要从外部视角测试产品的行为，因此，最终用户视角是黑盒测试的一个组成部分。

（4）黑盒测试采用有效输入和无效输入。用户在使用产品时出错是很正常的，因此，黑盒测试只采用有效输入时不够的，还应包括无效条件。

②白盒测试

测试特点：测试程序接口与结构

测试依据：软件程序

方法举例：逻辑覆盖

优点：对程序内部特定部位进行覆盖测试。

缺点：无法检验程序外部特性。

白盒测试方法主要是想对程序模块进行以下检查：

 （1）对程序模块的所有独立的执行路径至少测试一遍；

 （2）对所有的逻辑判定，取真与假的两种情况都能至少测一遍；

 （3）如果有更高的质量要求，测试对象流程图中所有独立的路径至少被执行一次；

 （4）测试内部数据结构的有效性，等等；

 （5）静态白盒测试  ：即代码审查，正式审查和检验设计和程序代码；

 （6）动态白盒测试 ：利用查看代码功能和实现方式得到的信息来设计和执行测试，也叫结构测试；

白盒测试用例技术包括逻辑覆盖和基本路径测试。

  逻辑覆盖：是以程序内在逻辑结构为基础的测试用例设计技术，这一方法要求测试人员对程序的逻辑结构有清楚的了解。

  基本路径测试：在程序控制流程图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例。

白盒测试方法具体测试步骤：

步骤1：由源代码出发，构造程序图，如基本路径法的流图等。

步骤2：根据程序图或控制流图，生成测试用例。

步骤3：编译被测源程序，生成可执行代码。

步骤4：测试用例的输入条件驱动，以执行程序测试。

步骤5：计算测试结果的实际覆盖率，如果达不到既定的覆盖率，则返回步骤2，否则就结束测试。

步骤6：对于测试结果，进行代码覆盖、测试通过率、失败率和可靠性等分析。

三．概括分析：

|  |  |
| --- | --- |
| 白盒测试 | 黑盒测试 |
| 考察程序逻辑结构 | 不涉及程序结构 |
| 用程序结构信息生成测试用例 | 用软件规格说明书生成测试用例 |
| 主要适用于单元测试和集成测试 | 可适用于单元测试到验收测试 |
| 对所有逻辑路径进行比较 | 某些代码得不到测试 |