**概述**

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，即清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。

白盒测试的测试方法有代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、基本路径测试法、域测试、符号测试、路径覆盖和程序变异。 [1]

白盒测试法的覆盖标准有逻辑覆盖、循环覆盖和基本路径测试。其中逻辑覆盖包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。六种覆盖标准发现错误的能力呈由弱到强的变化：

1.语句覆盖每条语句至少执行一次。

2.判定覆盖每个判定的每个分支至少执行一次。

3.条件覆盖每个判定的每个条件应取到各种可能的值。

4.判定/条件覆盖同时满足判定覆盖条件覆盖。

5.条件组合覆盖每个判定中各条件的每一种组合至少出现一次。

6.路径覆盖使程序中每一条可能的路径至少执行一次。

**原则**

(1)一个模块中的所有独立路径至少被测试一次。

(2)所有逻辑值均需测试true和false两种情况。

(3)检査程序的内部数据结构，保证其结构的有效性。

(4)在取值的上、下边界及可操作范围内运行所有循环。

**实施阶段**

1.测试计划阶段：根据需求说明书，制定测试进度。

2.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。

3.测试执行阶段：输入测试用例，得到测试结果。

4.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。

**三步法**

1. 根据代码的功能，人工设计测试用例进行基本功能测试。

2. 统计白盒覆盖率，为未覆盖的白盒单位设计测试用例，实现完整的白盒覆盖，比较理想的覆盖率是实现100%语句、条件、分支、路径覆盖。

3. 自动生成大量的测试用例，捕捉"程序员未处理某些特殊输入"形成的错误。

第1步的测试用例通常是现成的，因为详细设计文档会规定程序的基本功能，没有文档的，程序员在编程时也要想清楚程序的功能，这些基本功能就是基本测试用例。

第2步是在第1步的基础上，检查未覆盖的白盒单位，由于未覆盖的逻辑单位通常对应未测试的等价类，因此第2步可以找出第1步所遗漏的测试用例。

第3步用自动动态测试弥补第2步的固有缺陷。

"三步法"尽量避免重复工作，白盒方法和黑盒方法相结合，人工方法和自动方法相补充，如果第2步的覆盖率比较理想，那么基本上可以保证找出所有等价类。在开发过程允许的限度内，"三步法"已接近极限，当得起"彻底测试"四个字。

**优缺点**

优点

(1)迫使测试人员去仔细思考软件的实现。

(2)可以检测代码中的每条分支和路径。

(3)揭示隐藏在代码中的错误。

(4)对代码的测试比较彻底。

(5)让软件最优化。

缺点

(1)昂贵。

(2)无法检测代码中遗漏的路径和数据敏感性错误。

(3)不验证规格的正确性。