白盒测试方法

一、静态结构分析法

程序的结构形式是白盒测试的主要依据。研究表明程序员38%的时间花费在理解软件系统上，因为代码以文本格式被写入多重文件中，这是很难阅读理解的，需要其它一些东西来帮助人们阅读理解，如各种图表等，而静态结构分析满足了这样的需求。

在静态结构分析中，测试者通过使用测试工具分析程序源代码的系统结构、数据结构、数据结构、内部控制逻辑等内部结构，生成函数调用关系图、模块控制流图、内部文件调用关系图、子程序表、宏和函数参数表等各类图形图标，可以清晰地标识整个软件系统的组成结构，使其便于阅读和理解，然后可以通过分析这些图标，检查软件有没有存在缺陷或错误。

其中函数调用关系图通过应用程序中各函数之间的调用关系展示了系统的结构。通过查看函数调用关系图，可以检查函数之间的调用关系是否符合要求，是否存在递归调用，函数的调用曾是是否过深，有没有存在独立的没有被调用的函数。从而可以发现系统是否存在结构缺陷，发现哪些函数是重要的，哪些是次要的，需要使用什么级别的覆盖要求......

模块控制流图是与程序流程图相类似的由许多节点和连接节点的边组成的一种图形，其中一个节点代表一条语句或数条语句，边代表节点间控制流向，它显示了一个函数的内部逻辑结构。模块控制流图可以直观地反映出一个函数的内部逻辑结构，通过检查这些模块控制流图，能够很快发现软件的错误与缺陷

二、代码检查

代码检查包括桌面检查、代码审查和走查等，主要检查代码和设计的一致性，代码对标准的遵循、可读性，代码逻辑表达的正确性，代码结构的合理性等方面；发现违背程序编写标准的问题，程序中不安全、不明确和模糊的部分，找出程序中不可移植部分、违背程序编程风格的内容，包括变量检查、命名和类型审查、程序逻辑审查、程序语法检查和程序结构检查等内容。

代码检查方法

1、代码检查法

（1）桌面检查：这是一种传统的检查方法，由程序员检查自己编写的程序。程序员在程序通过编译之后，对源程序代码进行分析、检验，并补充相关文档，目的是发现程序中的错误。由于程序员熟悉自己的程序及其程序设计风格，桌面检查由程序员自己进行可以节省很多的检查时间，但应避免主观片面性

（2）代码审查

由若干程序员和测试员组成一个审查小组，通过阅读、讨论和争议，对程序进行静态分析的过程。代码审查分两步：第一步，小组负责人提前把设计规格说明书、控制流程图、程序文本及有关要求、规范等分发给小组成员，作为审查的依据。小组成员在充分阅读这些材料后，进入审查的第二步，召开程序审查会。在会上，首先由程序员逐句简介程序的逻辑。在此过程中，程序员或其他小组成员可以提出问题，展开讨论，审查错误是否存在。实践表明，程序员在讲解过程中能发现许多原来自己没有发现的错误，而讨论和争议则促进了问题的暴露。

在会前，应当给审查小组每个成员准备一份常见错误的清单，把以往所有可能发生的常见错误罗列出来，供与会者对照检查，以提高审查的失效。这个常见的错误清单也成为检查表，它把程序中可能发生的各种错误进行分类，对每一类错误列出尽可能多的典型错误，然后把它们制成表格，供再审查时使用

（3）走查

与代码审查基本相同，分为两步，第一步也是把材料分给走查小组的每个成员，让他们认真研究程序，然后再开会。开会的程序与代码审查不同，不是简单地读程序和对照错误检查表进行检查，而是让与会者“充当”计算机，即首先由测试组成员为所测试程序准备一批有代表性的测试用例，提交给走查小组。走查小组开会，集体扮演计算机角色，让测试用例沿程序的逻辑运行一遍，随时记录程序的踪迹，供分析和讨论用。

人们借助测试用例的媒介作用，对程序的逻辑和功能提出各种疑问，结合问题开展热烈的讨论和争议，能够发现更多的问题。

代码检查应在编译和动态测试之前进行，在检查前，应准备好需求描述文档、程序设计文档、程序的源代码请当、代码编译标准和代码缺陷检查表等。在实际使用中，代码检查能快速找到缺陷，发现30%～70%的逻辑设计和编码缺陷，而且代码检查看到的问题本身而非征兆。但是代码检查非常耗费时间，而且代码检查需要知识和经验的积累。

代码检查可以使用测试软件进行自动化测试，以利于提高测试效率，降低劳动强度，或者使用人工进行测试，以充分发挥人力的逻辑思维能力

2、代码检查项目

变量交叉引用表；标号的交叉引用表；检查子程序、宏、函数；等价性检查；常量检查；标准检查；风格检查；比较控制流；选择、激活路径；补充文档

根据检查项目可以编制代码规则、规范和检查表等作为测试用例，如编码规范、代码检查规范、缺陷检查表等

3、编码规范

编码规范是指程序编写过程中必须遵循的规则，一般会详细制定代码的语法规则、语法格式等

4、代码检查规范

在代码检查中，需要依据被测软件的特点，选用适当的标准与规则规范。在使用测试软件进行自动化代码检查时，测试工具一般会内置许多的编码规则。在自动化测试基础上使用桌面检查、代码走查、代码审查等人工检查的方法仔细检查程序的结构、逻辑等方面的缺陷

5、缺陷检查表

在进行人工代码检查时，代码缺陷检查表是我们用到的测试用例。

代码缺陷检查表中一般包括容易出错的地方和在以往的工作中遇到的典型错误

三、静态质量度查法

根据ISO/IEC 9126质量模型作为基础，我们可以构造质量度量模型，用于评估软件的各个方面。该模型从上到下分为3层：质量因素（Factors）、分类标准（Criteria）和度量规则（metrics）。其中质量因素对应ISO 9126质量模型的质量特性，分类标准对应ISO 9126质量模型的子特性，度量规则用于规范软件的各种行为属性。以下例子按照可维护性进行分析。

1、度量规则

度量规则使用了代码行数、注释频度等参数度量软件的各种行为属性，具体参数定义如表1

2、分类标准

软件的可维护性采用以下四个分类标准来评估：可分析性（ANALYZABILITY）、可修改性（CHANGEABILITY）、稳定性（STABILITY）、可测性（TESTABILITY）。每个分类标准由一系列度量规则组成，各个规则分配一个权重，由规则的取值与权重值计算出每个分类标准的取值。

function\_TESTABILITY\_DRCT\_CALLS+LEVL+PATH+PARA

3、质量因素

质量因素的取值与分类标准的计算方式类似：依据各分类标准取值组合权重方法计算.

function\_MAINTAINABILITY=function\_ANALYZABILITY

+function\_CHANGEABILITY

+function\_ATABILITY

+function\_TESTABILITY

四、基本路径测试法

设计出的测试用例要保证每一个基本独立路径至少要执行一次。

**欢迎您的下载，**

**资料仅供参考！**

致力为企业和个人提供合同协议，策划案计划书，学习课件等等

打造全网一站式需求