

软件测试 Software Testing

04. 软件测试方法

程适

cheng@snnu.edu.cn

计算机科学学院

2016 年 10 月 13 日



陕西师范大学
SHAANXI NORMAL UNIVERSITY

Outline

- 白盒测试
- 黑盒测试
- 静态测试和动态测试
- 主动测试和被动测试
- 形式化测试
- 风险测试
- 模糊测试
- 随机测试

软件测试方法

- 通过对软件缺陷的产生、分类、构成所做的讨论，可以更容易理解软件测试的目的，软件测试是为了更快、更早地将软件产品或软件系统中所存在的各种问题找出来，并促使系统分析人员、设计人员和程序员尽快地解决这些问题，最终及时地向客户提供高质量的软件产品。
- 要做到这一点，确保在有限的时间内找出系统中绝大部分的软件缺陷，必须依赖有效的软件测试方法。
- 在掌握了软件测试的背景、基本概念和过程之后，需要掌握软件测试的方法。

软件测试方法

- 软件测试方法，可以从宏观和微观两个方面看。从宏观看软件测试的方法，也就是讨论软件测试的方法论。
 - 从方法论看，更多体现了一种哲学的思想，例如辩证统一的方法，在测试中有许多对立的统一体，如静态测试和动态测试、白盒测试和黑盒测试、自动化测试和手工测试等。
 - 软件测试的方法论来源于软件工程的方法论，例如有面向对象的开发方法，就有面向对象的测试方法；有敏捷方法，就有和敏捷方法对应的测试方法。

软件测试方法

- 从微观看软件测试方法，就是软件测试过程中所使用的、具体的测试方法，例如等价类划分、边界值分析、正交实验方法等。

软件测试方法

- 白盒测试方法
- 黑盒测试方法
- 静态测试和动态测试
- 主动测试和被动测试
- 形式化测试方法
- 基于风险的测试
- 模糊测试方法
- ALAC 测试和随机测试方法

白盒测试

- 从哲学观点看，分析问题和解决问题的方法有两种：白盒方法和黑盒方法。
- 所谓**白盒方法**就是能够看清楚事物的内部，即了解事物的内部结构和运行机制，通过剖析事物的内部结构和运行机制，来处理和解决问题。
- 如果没有办法或不去了解事物的内部结构和运行机制，而把整个事物看成一个整体—黑盒子，通过分析事物的输入、输出及周边条件来分析处理问题，这种方法就是**黑盒方法**。

白盒测试

- 根据是针对软件系统的内部结构，还是针对软件系统的外部表现行为来采取不同的测试方法，分别被称为白盒测试（white-box testing）方法和黑盒测试（black-box testing）方法。
- 白盒测试，也称结构测试或逻辑驱动测试。

白盒测试

- 白盒测试主要用于单元测试，其基本原则有：
 - 保证每个模块中所有独立路径至少被使用一次；
 - 完成所有逻辑值分别为真值（true）和假值（false）的条件下的测试；
 - 在上下边界及可操作范围内运行所有循环，完成循环覆盖测试；
 - 检查内部数据结构以确保其有效性，完成边界条件的测试。

白盒测试

- 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
- 基本路径测试法

白盒测试

- **语句覆盖**：语句覆盖法的基本思想是设计若干测试用例，运行被测程序，使程序中的每个可执行语句至少被执行一次。
- **判定覆盖**：判定覆盖法的基本思想是设计若干用例，运行被测程序，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，即判定真假值均曾被满足。一个判定往往代表着程序的一个分支，所以判定覆盖也被称为分支覆盖。

白盒测试

- **条件覆盖**：条件覆盖的基本思想是设计若干测试用例，执行被测程序以后，要使每个判定中每个条件的可能取值至少满足一次。
- **判定-条件覆盖**：判定-条件覆盖实际上是将前两种方法结合起来的一种设计方法，它是判定和条件覆盖设计方法的交集，即设计足够的测试用例，使得判定条件中的所有条件可能取值至少执行一次，同时，所有判断的可能结果至少执行一次。

白盒测试

- **条件组合覆盖**：条件组合覆盖的基本思想是设计足够的测试用例，使得判断中每个条件的所有可能至少出现一次，并且每个判断本身的判定结果也至少出现一次。它与条件覆盖的差别是它不是简单地要求每个条件都出现“真”与“假”两种结果，而是要求让这些结果的所有可能组合都至少出现一次。

白盒测试

- **路径覆盖**：路径覆盖就是设计所有的测试用例，来覆盖程序中的所有可能的执行路径。
- **基本路径测试法**：基本路径测试法是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，从而设计测试用例的方法。设计出的测试用例要保证被测试程序的每个可执行语句至少被执行一次。

黑盒测试

- 黑盒测试方法常用于发现以下缺陷：
 - 是否有不正确或遗漏了的功能；
 - 能否正确地接受输入数据，能否产生正确的输出信息；
 - 功能操作逻辑不合理；
 - 界面是否出错、是否不美观；
 - 安装过程中出现问题，安装步骤不清晰、不够方便灵活；
 - 系统初始化问题。

黑盒测试

- 1 等价类划分法
- 2 边界值分析法
- 3 判定表方法
- 4 因果图法
- 5 正交实验法
- 6 功能图法
- 7 错误推测法

等价类划分法

- 等价类划分法：用一组有限的数去代表近似无限的数，这就是“等价类划分”方法的基本思想。等价类划分法就是解决如何选择适当的数子集来代表整个数据集的问题，通过降低测试的数目去实现“合理的”覆盖，覆盖了更多的可能数，以发现更多的软件缺陷。

边界值分析法

- 边界值分析法就是在某个输入输出变量范围的边界上，验证系统功能是否正常运行的测试方法。

判定表方法

- 一个判定表由“条件和活动”两部分组成，也就是列出了一个测试活动执行所需的条件组合，所有可能的条件组合定义了一系列的选择，而测试活动需要考虑每一个选择。

因果图法

- 因果图法（Cause-effect Diagram）借助图形，着重分析输入条件的各种组合，每种组合的条件就是“因”，它必然有一个输出的结果，这就是“果”。

正交实验法

■ 正交实验设计法（Orthogonal Test Design Method, OTDM）

功能图法

- 功能图法就是使用功能图形式化地表示程序的功能说明，并机械地生成功能图的测试用例。
- 功能图模型由状态迁移图和逻辑功能模型组成。

错误推测法

- 错误推测法（error guess method），或叫探索性测试方法（exploratory test）

静态测试和动态测试

- 1 静态测试就是静态分析，对模块的源代码进行研读，查找错误并收集一些度量数据，并不需要对代码进行编译和仿真运行。
- 2 动态测试是通过真正运行程序发现错误，通过观察代码运行过程，来获取系统行为、变量实时结果、内存、堆栈、线程以及测试覆盖度等各方面的信息，来判断系统是否存在问题，或者通过有效地测试用例，对应的输入输出关系来分析被测试程序的运行情况，来发现缺陷。

主动测试和被动测试

- 1 主动测试：测试人员主动向被测试对象发送请求，或借助数据、事件驱动被测试对象的行为，从而验证被测试对象的反应或输出结果。在主动测试中，测试人员和被测试对象之间发生直接相互作用的关系，而且被测试对象完全受测试人员控制，被测试对象处于测试状态，而不是实际工作状态。
- 2 被动测试：为了解决产品在线测试，被动测试方法应运而生。在被动测试方法中，软件产品运行在实际环境中，测试人员不干预产品的运行，而是被动地监测产品的运行，通过一定的被动机制来获得系统运行的数据，包括输入、输出数据。

形式化测试

- 1 形式化方法：形式化方法实际上就是基于数学的方法来描述目标软件系统属性的一种技术。
- 2 形式化验证：根据某些形式规范或属性，使用数学方法（形式逻辑方法）证明其正确性或非正确性。
- 3 基于模型的软件测试
- 4 扩展有限状态机方法

风险测试

- 1 列出软件的所有功能和特性;
- 2 确定每个功能出错的可能性
- 3 如果某个功能出错或缺某个特征, 需要评估对用户使用的软件产品的
影响程度;
- 4 根据上面两个步骤, 计算风险度;
- 5 根据可能出错的迹象, 来修改风险度;
- 6 确定测试的范围, 编写测试方案。

模糊测试

- 1 模糊测试（Fuzz testing）方法，简单地说，就是构造大量的随机数据作为系统的输入，从而检验系统在各种数据情况下是否会出现问题。
- 2 模糊测试方法缺乏严密的逻辑，不去推导哪个数据会造成系统破坏，而是设定一些基本框架，在这框架内产生尽可能多的杂乱数据进行测试，发现一些意想不到的系统缺陷。

随机测试

- 1 ALAC: act like a customer （向客户那样做）
- 2 ALAC 测试方法是一种基于客户使用产品的知识开发出来的测试方法。

Pareto 80/20 规律

- 1 一个软件产品或系统中全部功能的20%是常用的功能，用户的80%时间都在使用这20%的功能；而软件产品或系统中剩下的80%不是常用的功能，用户使用的比较少，只有20%时间在使用剩下的80%功能。
- 2 测试发现的所有错误的80%很可能集中在20%的程序模块中，另外20%的错误很可能集中在80%的程序模块中。

随机测试 (ad hoc test)

- 1 模糊测试强调大随机量的数据，并由工具自动产生，用于容错性和安全性测试；
- 2 随机测试强调人工的自由测试，不受测试用例的束缚，放开思想，灵活地进行各种测试。

小结

- 白盒测试
- 黑盒测试
- 静态测试和动态测试
- 主动测试和被动测试
- 形式化测试
- 风险测试
- 模糊测试
- 随机测试

测试方法的4 种基本组合

	白盒测试方法	黑盒测试方法
静态测试方法	静态-白盒测试方法（对源程序代码的语法检查、扫描、评审等）	静态-黑盒测试方法（对需求文档、需求规格说明书的审查活动，一些非技术性文档测试等）
动态测试方法	动态-白盒测试方法（在单元测试中，一边运行代码，一边对结果进行检查、验证和调试等）	动态-黑盒测试方法（在运行程序时，通过数据驱动对软件进行功能测试，从用户角度验证软件的各项功能）

致谢

谢谢! 欢迎提问!