

# 测试开发工程师面经总结

## 1.你是怎么写测试用例的?

### A.黑盒测试

- 1) 等价类划分法: 输入的数据划分为有效等价类和无效等价类
- 2) 边界值划分法: 边界值划分法使被测程序在边界值及其附近运行, 从而更有效地暴露程序中潜藏的错误。不仅根据输入条件, 它还根据输出情况设计测试。(如果输入条件规定了取值范围分别对最大、最小个数及稍小于最小、稍大于最大个数作为测试用例)
- 3) 错误推测法: 猜测被测程序在哪些地方容易出错, 针对可能的薄弱环节来设计测试用例。
- 4) 因果图法: 检查输入条件的各种组合情况, 从功能说明中找出因(输入条件)和果(输出或程序状态的修改, 通过因果图功能说明转换成一张判定表, 然后为判定表的每一例设计测试用例。

### B.白盒测试

- 1) 语句覆盖法: 使得程序中的每一个语句至少被遍历一次。
- 2) 判定覆盖(分支): 使得程序中每一个分支至少被遍历一次。
- 3) 条件覆盖: 使得每个判定的条件获取各种可能的结果。
- 4) 判定/条件覆盖: 使得判定中的条件取得各种可能的值, 并使得每个判定取得各种可能的结果。
- 5) 条件组合覆盖: 使得每个判定条件的各种可能组合都至少出现一次。
- 6) 路径覆盖: 覆盖程序中所有可能的路径。(满足判定覆盖)

## 2.测试的相关流程是什么?

- 1) 需求: 阅读需求, 理解需求, 与客户、开发、架构多方交流, 深入了解需求。-testing team
- 2) 测试计划: 根据需求估算测试所需资源(人力、设备等)、所需时间、功能点划分、如何合理分配安排资源等。-testing leader or testing manager。
- 3) 用例设计: 根据测试计划、任务分配、功能点划分, 设计合理的测试用例。-testing leader, senior tester
- 4) 执行测试: 根据测试用例的详细步骤, 执行测试用例。-every tester(主要是初级测试人员)
- 5) 执行结果记录和bug记录: 对每个case记录测试的结果, 有bug的在测试管理工具中编写bug记录。-every tester(主要是初级测试人员)
- 6) defect tracking: 追踪leader分配给你追踪的bug.直到 bug fixed。-every tester。
- 7) 测试报告: 通过不断测试、追踪, 直到被测软件达到测试需求要求, 并没有重大bug。。
- 8) 用户体验、软件发布等

## 3.设计一个微信朋友圈点赞的测试用例

- 1) 是否可以正常点赞和取消;
- 2) 点赞的人是否在可见分组里;
- 3) 点赞状态是否能即时更新显示;
- 4) 点赞状态, 共同好友是否可见;
- 5) 不同手机, 系统显示界面如何;

- 6) 性能检测，网速快慢对其影响；
- 7) 点赞显示的是否正确，一行几个；
- 8) 点赞是否按时间进行排序，头像对应的是否正确；
- 9) 是否能在消息列表中显示点赞人的昵称、备注；
- 10) 可扩展性测试，点赞后是否能发表评
- 11) 是否在未登录时可查看被点赞的信息。

4.分别介绍单元测试、集成测试、验收测试（确认测试）、系统测试、回归测试等等。

#### A.单元测试

目的：通过模块测试，使其代码达到模块设计的要求。

任务：

- 1) 对模块代码进行编译，发现其语法错误；
- 2) 确定模块的测试策略（通常采用白盒测试法），并据此设计一组测试用例；
- 3) 用选定的测试用例对模块进行测试，直至满足测试终止标准为止；
- 4) 编制单元测试评估报告。

#### B.集成测试

目的：将经过单元测试的模块逐步组装成具有良好一致性的完整的程序

任务：

- 1) 制订集成测试实施策略
- 2) 确定集成测试的实施步骤，设计测试用例
- 3) 逐一地添加模块，进行测试

#### C.确认测试

目的：确认组装好的程序是否满足软件需求（SRS）

任务：

- 1) 有效性测试（黑盒测试）
- 2) 配置复审（configuration review）
- 3) 验收测试—针对专用应用软件
- 4) alpha与beta测试—针对通用产品软件

alpha与beta测试：Alpha 测试( $\alpha$ 测试)是由公司内部的用户在模拟实际操作环境下进行的受控测试，Alpha测试不能由程序员或测试员完成。Beta测试( $\beta$ 测试)是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场，Beta测试不能由程序员或测试员完成。

两者区别：

Alpha测试由开发人员或测试人员在场，可随时记录下错误和使用中出现的问题；

Beta测试开发人员和测试人员都不在场。

#### D.系统测试

目的：软件安装到系统中以后，能否与系统的其余部分协调运行，还包括功能实现等。

任务：

- 1) 测试是否与硬件协调运行
- 2) 测试是否和原来就有的其它软件协调运行
- 3) 测试是否完成系统需求对它的要求

系统测试技术

- 1) 安全和存取控制测试

- 2) 故障及恢复测试
- 3) 性能测试
- 4) 强度测试 (Stress Testing )

## 测试用例

测试用例是指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标，测试环境，输入数据，测试步骤，预期结果，测试脚本等并形成文档。

### 方法（黑盒）

#### 1) 等价类划分、边界值

- 1
- 2

- 2) 因果图、判定表
- 3) 正交排列方法、场景法
- 4) 状态转换图方法
- 5) 测试大纲方法

### 等价类划分&&边界值

按照控件的数据要求，把输入的数据分为有效等价类和无效等价类。有效等价类是对程序有意义、合法的输入，输入有效等价类后，程序应该正确计算或正常运行。无效等价类是对程序无意义、不合法的输入，输入无效等价类后，程序应该给出错误提示，或者根本不让用户输入。在测试的时候，分别从每个等价类中选取一组数据编写用例即可。比如输入“姓名”，要求1——20个字符，不能为空。这样，有效等价类就是1——20之间的数据，比如：张三；无效等价类为：小于1的字符，就是为空，另外还有大于20个字符的，比如准备一个30个字符的姓名进行测试。

边界值方法就是对输入数据的有效无效的边界点及其两边的数据进行测试。比如，刚才的姓名，要测试1个字符，0个字符和2个字符，20个字符、19个和21个字符。

### 因果图&&判定表

因果图法最终生成的是判定表，它适用于检查程序输入条件的各种组合情况步骤大概是这样~分析软件规格说明描述中的因果关系（输入与输出的因果关系）找出原因与结果、原因与原因之间的对应关系，画出因果图在因果图上标记约束或限制条件把因果图转化为判定表将判定表中的每一列拿出来设计测试用例。

### 正交排列&&场景法

正交表是一整套规则的设计表格，用 L 为正交表的代号，n 为试验的次数，t 为水平数，c 为列数，也就是可能安排最多的因素个数。正交表的构造需要用到组合数学和概率学知识，现在广泛使用的  $L_n(t^c)$  类型的正交表构造思想比较成熟。比如测试一个功能，该功能有两个条件，不同的条件得出不同的结果；那么根据正交排列有  $2 \times 2 = 4$  种方法

## 状态转换图

通过描绘系统的状态及引起系统状态转换的事件，来表示系统的行为。此外状态转换图还指明了作为特定事件的结果系统将做哪些动作（例如，处理数据）。因此状态转换图提供了行为建模机制。

在状态转换图中，每一个节点代表一个状态，其中双圈是终结状态。许多单片机教材上对工作模式的表达通常采用状态图的形式。

### 步骤

- 1) 找出所有的动作
- 2) 找出所有的状态
- 3) 把所有的动作单独加载到初始状态中，产生新的状态，然后再把这些动作单独加载到新的状态中，再次产生更新的状态，以此类推。

### 状态转换图的核心

动作：用户能够向软件做的操作

状态：软件运行到某个时刻所处的情况

## 错误推测法

错误推测法是指：在测试程序时，人们可以根据经验或直觉推测程序中可能存在的各种错误，从而有针对性地编写检查这些错误的测试用例的方法。

错误推测方法的基本思想：列举出程序中所有可能有的错误和容易发生错误的特殊情况，根据他们选择测试用例。例如，在单元测试时曾列出的许多在模块中常见的错误。以前产品测试中曾经发现的错误等，这些就是经验的总结。还有，输入数据和输出数据为 0 的情况。输入表格为空格或输入表格只有一行。这些都是容易发生错误的情况。可选择这些情况下的例子作为测试用例。

总之，就是进行错误的操作。

例如，输入数据和输出数据为 0 的情况；输入表格为空格或输入表格只有一行。这些都是容易发生错误的情况。可选择这些情况下的例子作为测试用例。

例如，测试手机终端的通话功能，可以设计各种通话失败的情况来补充测试用例：

- 1) 无 SIM 卡插入时进行呼出（非紧急呼叫）
- 2) 插入已欠费 SIM 卡进行呼出
- 3) 射频器件损坏或无信号区域插入有效 SIM 卡呼出
- 4) 网络正常，插入有效 SIM 卡，呼出无效号码（如 1、888、333333、不输入任何号码等）
- 5) 网络正常，插入有效 SIM 卡，使用“快速拨号”功能呼出设置无效号码的数字

例如，测试一个对线性表（比如数组）进行排序的程序，可推测列出以下几项需要特别测试

的情况：

- 1) 输入的线性表为空表；
- 2) 表中只含有一个元素；
- 3) 输入表中所有元素已排好序；
- 4) 输入表已按逆序排好；
- 5) 输入表中部分或全部元素相同。

## 测试大纲法

有多个窗口，每个窗口有多个动作，这些动作之间有相互联系，为了弄清动作之间的联系或者窗口之间的关系，可以使用测试大纲方法。

### 操作步骤

- 1) 列提纲，列出所有的窗口以及每个窗口中的动作
- 2) 找出各窗口中动作的联系，编写用例。

# 功能测试

Functional testing（功能测试），也称为behavioral testing（行为测试），根据产品特性、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。功能测试是为了确保程序以期望的方式运行而按功能要求对软件进行的测试，通过对一个系统的所有的特性和功能都进行测试确保符合需求和规范。

功能测试也叫黑盒测试或数据驱动测试，只需考虑需要测试的各个功能，不需要考虑整个软件的内部结构及代码。一般从软件产品的界面、架构出发，按照需求编写出来的测试用例，输入数据在预期结果和实际结果之间进行评测，进而提出更加使产品达到用户使用的要求。

# 黑盒测试

黑盒测试（Black-box Testing，又称为功能测试或数据驱动测试）是把测试对象看作一个黑盒子。利用黑盒测试法进行动态测试时，需要测试软件产品的功能，不需测试软件产品的内部结构和处理过程。

比如黑盒技术设计测试用例的方法有：等价类划分、边界值分析、错误推测、因果图和综合策略。

黑盒测试注重于测试软件的功能性需求，也即黑盒测试使软件工程师派生出执行程序所有功能需求的输入条件。黑盒测试并不是白盒测试的替代品，而是用于辅助白盒测试发现其他类型的错误。

黑盒测试试图发现以下类型的错误：

- （1）功能错误或遗漏
- （2）界面错误

- (3) 数据结构或外部数据库访问错误
- (4) 性能错误
- (5) 初始化和终止错误

## 兼容性测试

兼容测试（Compatibility Test Suite）简称CTS。

指对所设计程序与硬件、软件之间的兼容性的测试。

测试分类：

一、浏览器兼容测试

二、分辨率兼容测试

一般来说，兼容性指能同时容纳多个方面，在计算机术语上兼容是指几个硬件之间、几个软件之间或是软硬件之间的相互配合程度。

兼容性测试是指测试软件在特定的硬件平台上、不同的应用软件之间、不同的操作系统平台上、不同的网络等环境中是否能够很友好的运行的测试。

## 意义

Application Compatibility Test（兼容性测），主要目的是为了兼容第三方软件，确保第三方软件能正常运行，也难怪乎不受影响。

具体表现如下：

- 1、待测项目在不同的操作系统平台上正常运行，包括待测试项目能在同一操作系统平台的不同版本上正常运行；
- 2、待测项目能与相关的其他软件或系统“协调工作”；
- 3、待测项目能在指定的硬件环境中正常运行；
- 4、待测项目能在不同的网络环境中正常运行。

[2]

## 作用

- 1、兼容性测试能够进一步提高产品的质量
- 2、兼容性测试能使软件与尽可能多的其他软件“和平共处”，尽可能达到平台无关性
- 3、兼容性测试能尽可能的保证软件存在的价值，它是衡量一个软件质量的重要指标
- 4、兼容性测试能使软件产品的市场更广阔

## 交叉事件测试

交叉测试又叫事件或冲突测试，是指一个功能正在执行过程中，同时另外一个事件或操作对该过程进行干扰的测试。如；App在前/后台运行状态时与来电、文件下载、音乐收听等关键运用的交互情况测试等。交叉事件测试非常重要，能发现很多应用中潜在的性能问题。

- 1) 多个App同时运行是否影响正常功能

- 2) App运行时前/后台切换是否影响正常功能
- 3) App运行时拨打/接听电话
- 4) App运行时发送/接收信息
- 5) App运行时发送/收取邮件
- 6) App运行时切换网络 (2G、3G、wifi)
- 7) App运行时浏览网络
- 8) App运行时使用蓝牙传送/接收数据
- 9) App运行时使用相机、计算器等手机自带设备

参考文章 <http://blog.csdn.net/jusulysunbeamy/article/details/49932839>

## 接口测试

接口测试一般会用于多系统间交互开发，或者拥有多个子系统的应用系统开发的测试。接口测试适用于为其他系统提供服务的底层框架系统和中心服务系统，主要测试这些系统对外部提供的接口，验证其正确性和稳定性。接口测试同样适用于一个上层系统中的服务层接口，越往上层，其测试的难度越大。接口测试在淘宝的应用是一个自下而上的发展过程。

### 方法

- 1: 写一个程序去调用该接口，看是否能够达到该接口所定义的功能
- 2: 根据该接口参数，构造不同的用例，测试接口在参数合法及非法情况下能否达到预期效果
- 3: 根据该接口中的逻辑，设计不同条件的用例，测试该接口实现代码的逻辑
- 4: 进行容错及健壮性测试
- 5: 静态检测代码，看是否有内存泄露、或永远走不到的分支、代码规范及逻辑是否合理。
- 6: 对于一些接口，需要进行多线程测试

## 自动化测试

一般是指软件测试的自动化，软件测试就是在预设条件下运行系统或应用程序，评估运行结果，预先条件应包括正常条件和异常条件。

### 定义

自动化测试是把以人为驱动测试行为转化为机器执行的一种过程。通常，在设计了测试用例并通过评审之后，由测试人员根据测试用例中描述的规程一步步执行测试，得到实际结果与期望结果的比较。在此过程中，为了节省人力、时间或硬件资源，提高测试效率，便引入了自动化测试的概念。

## 测试报告

项目简介：一些需要介绍的内容，项目简称的解释，项目背景等等。

测试内容：测试内容的大纲。

测试环境：测试环境的描述，包括客户端和网络环境。

测试资源：测试过程中的测试资源使用。

测试数据分析：对于整个过程测试的一个分析，得出结论。

遗留问题：对于软件遗留问题有详细说明。报告的内容每个人都可以说清楚，但是仅仅简单的罗列，也能使看的人很费劲。

## 如何展现这些东西使你的测试报告丰满而又有说服力，并且易读易看呢？

不罗列详细数据，挑拣一些能说明问题分析数据的：比如缺陷走势图，模块的bug分布等等。加必要的简短的分析。图形简单易懂，且比较直观。如果不能说明问题或者一些不重要的图表就不用都一一列在报告中了，会显得报告比较啰嗦。

分析结论一定要给出，并且明显的位置。让项目经理清楚你的测试结论是什么，当时间比较紧的时候他看到结论心里就有数了。