

# 2021 软件测试面试

Software test interview

**你若盛开，清风自来！**

你的好友：程序员一凡

第3版

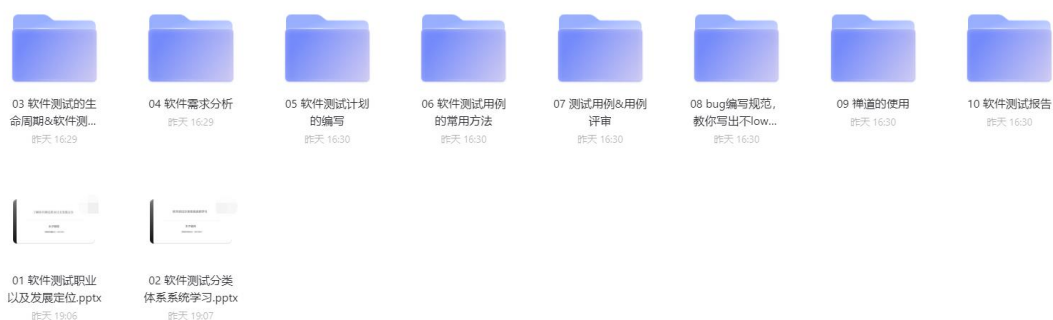
Software test interview

## 写在最前面：

你好，很高兴认识你！我是一凡。

之前面试题第一版，收到了很多朋友的好评！

索性今天把面试反馈回来的最新面试题，整理好在原有的基础上做一个新的 PDF 文档。



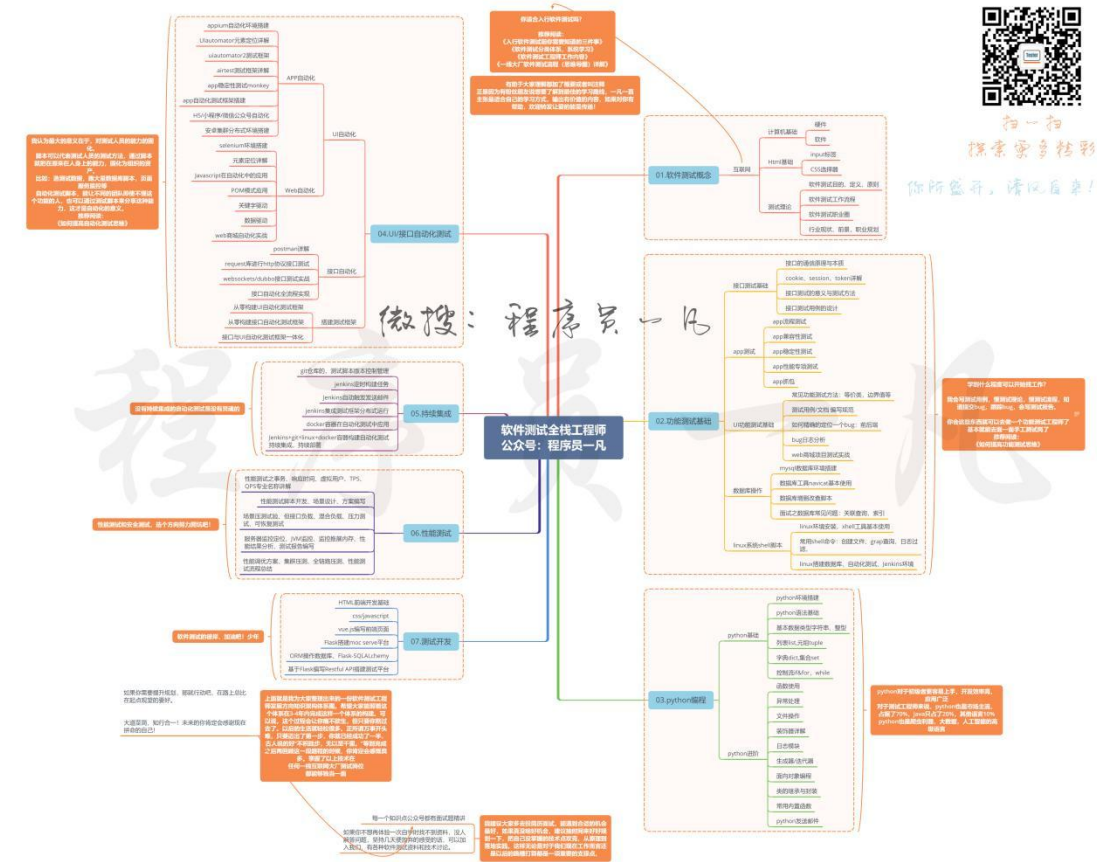
哔哩哔哩账号也即将更新，从软件测试零基础，测试理论开始。

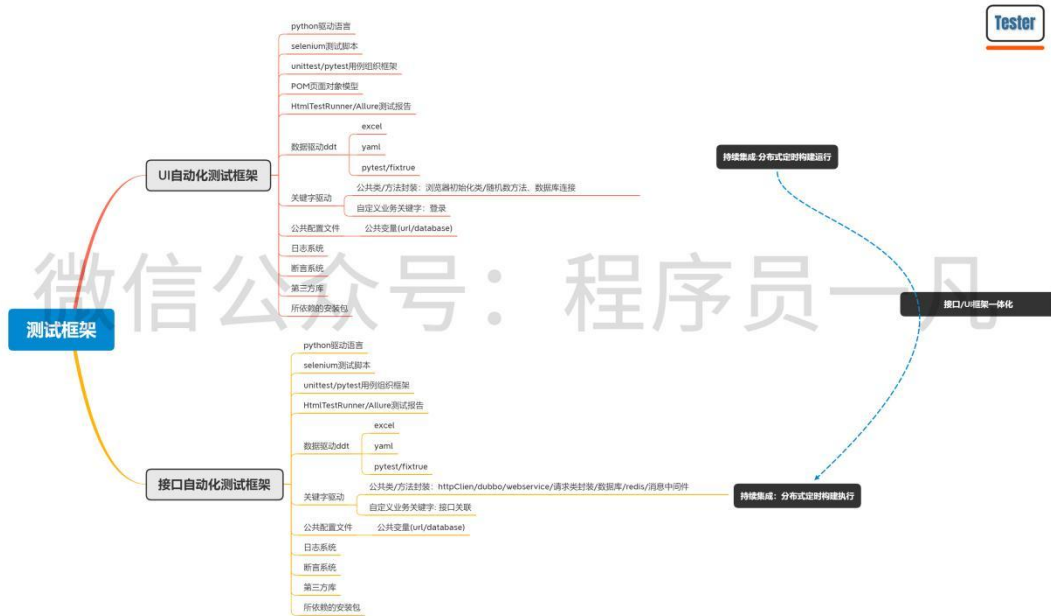


【可以加 QQ 技术交流群】



或者围观我的微信朋友圈，每一期视频都是干货精剪，需要学的知识点也都分类好，如果对你有帮助，记得三连一下





(软件测试行业现状前景文章+各类思维导图+学习路线+视频实战讲解。 微信公众号：程序员一凡皆有收录)

一个用心码了这么多文字的人，往往渴望得到大家的认可。如果我的分享对你有那么一点点启发和帮助，给我点个赞！

面试题文档中的一些知识点有些会附带知识点拓展链接地址，花了几个月收集整理，一天时间的编辑。“匠心之作”希望对你有帮助！

每一个成功的人，在前进的路上都会收到很多的鼓励，也会愿意鼓励他人，我深深感受到鼓励带给我的帮助，你们的每一次点赞，就是对我最大的支持，让我一直坚持的做出更好的内容。

愿你我相遇皆有收获！祝一切顺利！

真正优质的课程，会随时间的推移，慢慢浮出水面，被大众接受，也只有那个时候，培训行业的春天，才会到来；

认真做好服务，真正的收费点是让学员真心认可你你们，用心服务好每个学员打造优质的原创内容，才是唯一能在这样的战争中，活下去的途径。这个行业最终一定会产生独角兽企业，而这个企业一定是这样的企业。

我们其实都愿意知识付费！我自己就是学习 Adoube 全家桶，都是买了付费课程，很多人说那么多白嫖的资源，完全可以自学。其实学习这件事，很大一部分还是看我们自己，我选择了购买收费课程就是他们整理好了很适合我的学习路线，以及实战项目，省下了很多宝贵的时间去琐碎的拼凑知识点。

发现网上有关测试的学习资料还是太分散了，不像开发的学习资料那么系统化。希望能帮到刚刚入行的测试新手们，希望能够给大家带来帮助！

(善用 Ctrl+F，快速找到你要的面试题及答案) 祝大家顺利拿到心仪的 offer。



## 测试理论基础

### 什么是软件测试？

答：软件测试是在规定的条件下对程序进行操作，以发现错误，对软件质量进行评估。

---

### 软件测试的目的是什么？

答：软件测试的目的在于（1）发现软件的缺陷和错误（2）保证软件的质量，确保能够满足用户以及产品的需求。

（标重点）软件测试的目的是为了找 **bug**，并不是验证软件没有 **bug**。

---

### 白盒测试用例设计常用方法

答：

静态测试：不用运行程序的测试，如文档测试、代码检查等

动态测试：需要执行代码，接口测试、覆盖率分析、性能分析、内存分析等。

**逻辑覆盖法**：主要包括语句覆盖，判断覆盖，条件覆盖，判断/条件覆盖，条件

**组合覆盖，路径覆盖**等。

六种覆盖标准发现错误的能力由弱到强的变化：

1.

**语句覆盖**，每条语句至少执行一次。

2.

**判断覆盖**，每个判断的每个分支至少执行一次。

3.

**条件覆盖**，每个判段的每个条件应取到的各种可能的值。

4.

判断/条件覆盖，同时满足判断覆盖条件覆盖。

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。

---

### 黑盒测试用例设计常用方法

答：等价划分类，边界值分析，错误推测法、因果图法、场景法、正交试验设计法、判定表驱动分析法、功能图分析法等。

---

### 什么是灰盒测试？

答：灰盒测试，是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段。目前互联网的测试大多数都是灰盒测试。

### 列举出你所了解的软件测试方式

答：

按照软件的生命周期划分：单元测试、集成测试、系统测试、回归测试、验收测试。

按照测试关注点划分：功能测试、性能测试、稳定性测试、易用性测试、安全性测试。

按照测试实施者划分：开发方测试（ $\alpha$ 测试）、用户测试（ $\beta$ 测试）、第三方测试。

按照技术/测试用例设计划分：白盒测试、黑盒测试、灰盒测试。

按照分析方法划分：静态测试、动态测试。

按照测试执行方式划分：手工测试、自动化测试。

按照测试对象划分：程序测试、文档测试。

---

### 什么是单元测试

答：完成最小的软件设计单元（模块）的验证工作，确保模块被正确编码。通常情况下是白盒的，对代码风格和规则、程序设计和结构、业务逻辑等进行静态测



试，及早发现和解决不易显现的错误。

---

单元测试、集成测试、系统测试、验收测试、回归测试这几步最重要的是哪一步？

答：这些测试步骤分别在软件开发的阶段对软件进行测试，我认为对软件完整功能进行测试的系统测试很重要，因为此时单元测试和集成测试已完成，系统测试能够对软件所有功能进行功能测试，能够覆盖系统所有联合的部件，是针对整个产品系统进行的测试，能够验证系统是否满足需求规格的定义，因此，我认为系统测试很重要。

集成测试和系统测试的区别，以及应用场景分别是什么？

答：

区别：

- 

执行顺序：先执行集成测试，待集成测试问题修复后，再做系统测试。

- 

用例粒度：集成测试比系统测试用例更详细，集成测试对于接口部分也要重点写，而系统测试的用例更接近用户接受的测试用例。

应用场景：

- 

集成测试：一般包含接口测试，对程序的提测部分进行测试。测试方法一般选用黑盒测试和白盒测试相结合。

- 

系统测试：针对整个产品的全面测试，既包含各模块的验证性测试和功能性测试，又包含对整个产品的健壮性、安全性、可维护性及各种性能参数的测试。测试方法一般采用黑盒测试法。

---

测试开发需要哪些知识？具备哪些能力？

答：

需要的知识：

软件测试基础理论知识，如黑盒测试、白盒测试等；

编程语言基础，如 C/C++、java、python 等；

自动化测试工具，如 Selenium、Appium 等；

计算机基础知识，如数据库、Linux、计算机网络等；



测试卡框架，如 JUnit、Pytest、Unititest 等。

具备的能力：

业务分析能力、缺陷洞察能力、团队协作能力、专业技术能力、逻辑思考能力、问题解决能力、沟通表达能力和宏观把控能力。

---

请说一下手动测试与自动化测试的优缺点

答：

手工测试缺点：

1. 重复的手工回归测试，代价昂贵、容易出错。
2. 依赖于软件测试人员的能力。

手工测试的优点：

1. 测试人员具有经验和对错误的猜测能力。
2. 测试人员具有审美能力和心理体验。
3. 测试人员具有是非判断和逻辑推理能力。

自动化测试的缺点：

1. 不能取代手工测试。
2. 无法运用在测试复杂的场景
3. 手工测试比自动化测试发现的缺陷更多。
4. 对测试质量的依赖性极大。
5. 自动化测试不能提高有效性。
6. 比手动测试脆弱，需要维护成本。
7. 工具本身并无想象力。

自动化测试的优点：

1. 对程序的回归测试更方便。
- 2.

可以运行更多更繁琐的测试。

3.

可以执行一些手工测试困难或不可能进行的测试。

4.

更好地利用资源。

5.

测试具有一致性和可重复性。

6.

测试的复用性。

7.

增加软件的信任度。

---

### 自动化测试的运用场景举例

答：

1.

线上巡检（UI+接口）

2.

简单场景监控

3.

稳定性测试（monkey+遍历测试）

---

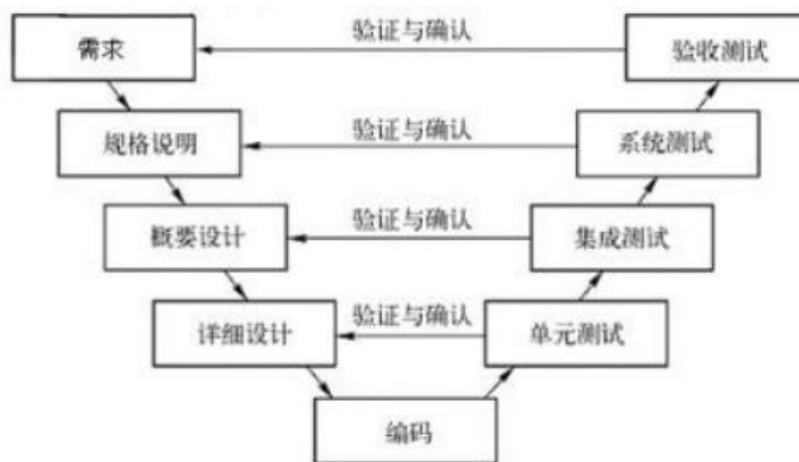
### 软件测试的核心竞争力是什么？

答：早发现问题和发现别人无法发现的问题。

### 测试和开发要怎么结合才能使软件的质量得到更好的保障

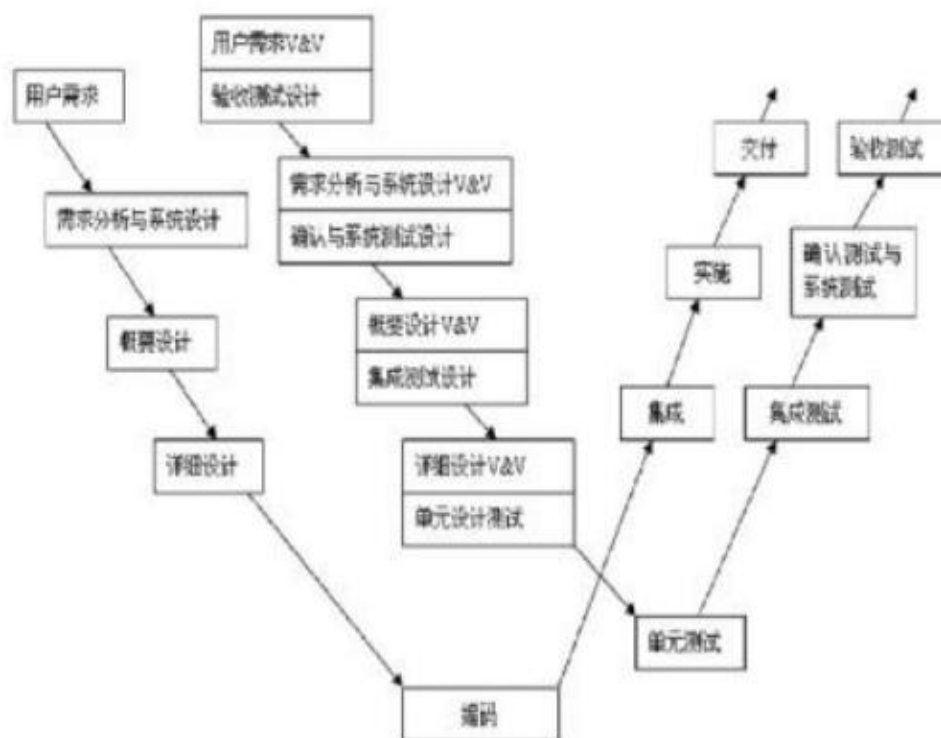
答：测试和开发可以按照 V 模型或 W 模型的方式进行结合。但应该按照 W 模型的方式进行结合比较合理。

V 模型：



测试过程加在开发过程的后半段，比较被动。

W 模型：



测试提前，甚至和开发是同步进行，测试不仅是程序，还包括需求和设计。W 模型有利于尽早地全面的发现问题，降低软件开发的成本，风险前置。

怎么实施自动化测试

答：

1.  
首先判断项目适不适合进行自动化测试。
  2.  
对项目做需求分析。
  3.  
制定测试计划和测试方案。
  4.  
搭建自动化测试框架。
  5.  
设计或编写测试用例。
  6.  
执行自动化测试。
  7.  
评估。
- 

## 测试的相关流程

答：

按 W 模型：

需求测试 -> 概要设计测试 -> 详细设计测试 -> 单元测试 -> 集成测试  
-> 系统测试 -> 验收测试

我工作中实际测试流程：

需求评审 -> 技术评审 -> case 评审 -> 开发自测以及冒烟测试 -> 整体提测  
(集成测试) -> 回归测试 -> 系统测试 -> 验收测试

---

## 测试项目具体工作是什么

答：

1.  
搭建测试环境
2.  
撰写测试用例

3.  
执行测试用例
  4.  
写测试计划、测试报告
  5.  
测试并提交 BUG
  6.  
跟踪 BUG 修改情况
  7.  
自动化测试
  8.  
性能测试、压力测试、安全测试等其他测试
- 

## BUG 分级

答：两个维度去划分

### 1.

按 BUG 严重程度划分等级：

- **blocker**：系统无法执行，崩溃，或严重资源不足，应用模块无法启动或异常退出，无法测试，系统不稳定。常见的：严重花屏、内存泄漏、用户数据丢失或破坏、系统崩溃/死机/冻结、模块无法启动或异常退出、严重的数值计算错误、功能设计与需求严重不符、服务器 500 等。
- **critical**：影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统的稳定性。常见的有：功能未实现，功能错误、系统刷新错误、数据通讯错误、轻微的数值计算错误、影响功能及界面的错别字或拼写错误。
- **major**：界面、性能缺陷、兼容性。常见的有：操作界面错误、边界条件错误、提示信息错误，长时间操作无法提示、系统未优化、兼容性问题。
- **minor/trivial**：易用性及建议性的问题。

### 2. 按 BUG 处理优先级划分：

- immediate: 马上解决
  - urgent: 急需解决
  - high: 高度重视，有时间马上解决
  - low: 在系统发布前解决或确认可以不用解决
- 

### **APP 性能指标有哪些？**

答：内存、CPU、流量、电量、启动速度、滑动速度、界面切换速度、与服务器交互的网络速度。

---

### **APP 测试工具有哪些？**

接口测试：postman

性能测试：jmeter

抓包工具：chales、fiddler

UI 自动化：uiautomator2、appium、atx

稳定性测试：monkey、maxim、uicrawler、appcrawler

兼容性测试：wetest、testin

内存、cpu、电量测试：GT、soloPi

弱网测试：chales

---

### **BUG 的生命周期**

答：

复杂版：

1. New（新的）

2. Assigned（已指派）
3. Open（打开的）
4. Fixed（已修复）
5. Pending Reset（待测试）
6. Reset（再测试）
7. Closed（已关闭）
8. Reopen（再次打开）
9. Pending Reject（拒绝中）
10. Rejected（被拒绝的）
11. Postponed（延期）

简单版：

1. 创建 bug
  2. 分配 bug
  3. 修复完待测试
  4. 关闭
  5. 重新开启
  6. 无效
- 

什么是 $\alpha$ 测试和 $\beta$  测试？

答：

$\alpha$ 测试：在受控的环境下进行，由用户在开发着的场所进行，开发者指导用户测试，开发者负责记录发现的错误和使用中遇到的问题。

$\beta$  测试：在开发者不可控的环境下进行，由软件最终用户在一个或多个客户场所下进行，用户记录测试中遇到的问题，并定期上报给开发者。

---

谈谈对敏捷的理解



答：

提到敏捷，不得不联想到瀑布开发，瀑布开发按项目为核心，一般都会有一个相对较长的项目周期，一开始把项目设计得大而全，完成项目并交付后，工作重心就会转移到另一个项目去。而敏捷开发是以需求为核心，一开始不会把产品设计得大而全，而是通过快速迭代的方式，不断采集需求，不断更新迭代。敏捷开发的开发周期更短，能够快速试错，快速迭代，敏捷开发比瀑布开发更顺应目前的软件开发趋势。

敏捷开发也对应着有敏捷测试，测试环节贯穿整个迭代周期，从需求评审到发布上线，都离不开测试快速跟进。测试左移：需求评审、用例设计、自测工具、静态代码扫描等；测试中：业务测试，接口测试，性能测试等；测试右移：稳定性测试，回归测试，灰度测试等

## 什么是压力测试？压力测试需要考虑哪些因素？

答：

压力测试是在高负载情况下，对系统稳定性进行测试。在高负载的情况下，系统出现异常的概率要比正常负载时要高。高负载包含长时间运行、大数据、高并发等情况。

在做压力测试时，一般要考虑环境因素、性能指标、运行时间等要素。

压测环境最好和生产环境一致，假如要在生产环境进行压测，需要在凌晨等在线用户量极少的情况下进行。在生产环境测试时要做好数据隔离，生产环境需提供虚拟数据，采用虚拟账号，避免对真实线上用户造成影响。

性能指标包括，内存、CPU、TPS、QPS、网络流量、错误统计等，这些指标需要监控。

压测一般需要运行长时间，最好能够通过长时间的压测，绘制出曲线图，这样更容易观察到性能瓶颈。

## 软件测试扫盲

### 按测试技术划分

- 黑盒测试、白盒测试、灰盒测试

### 被测试对象是否运行

- 动态测试、静态测试 (文档检查、代码走查)

### 按不同的测试手段划分

- 

手工测试(点工) 、自动化测试(工具+代码)

#### 按测试包含的内容划分

- 

功能测试、界面测试、安全测试、兼容性测试、易用性测试

- 

功能测试：测试业务逻辑(手工、自动化)--核心重要

- 

界面测试：UI (user interface) --外观美好、设计合理、友好、---主观性强=需求文档 (原型图 UI 切图)--优先级会低

- 

安全测试：高级类型-工具 (扫描--appsan) 代码(脚本-sql 注入) --漏洞，薄弱

- 

性能测试： 高级类型-双十一(访问人数多)--并发 (10000) ---资源内存  
--正常处理

- 

易用性测试： --人性化 ， 舒适，用户体验==提 bug ===站在用户角度考虑，考虑成熟产品

- 

兼容性测试： 软件+硬件(windows, Linux, MacOS,Android,Ios); 软件+软件--调用;  
软件不同版本之前 App(老功能，数据)

#### 按测试阶段划分

- 

单元测试、集成测试、系统测试、验收测试、a 测试、b 测试

#### 其他测试

回归测试、冒烟测试、探索性测试/自由测试(测试思维)

- 

回归测试: regression test : 测试-bug ， 开发修复 bug(修改代码)== 验证 bug==其他没被修改的代码模块测试，影响：上线之前-很多轮 (2-4 轮) 的回归冒烟测试（重复）

==自动化测试

- 

冒烟测试：来源--硬件测试：电路板-冒烟-短路被烧了=打回开发重做--软件测试：软件提测-核心业务功能  
主流程 ， 提高测试效率

- 

探索性测试: 发散测试 -能力要求非常的高 ， 没有依据，方法 ， 只能靠 经验、积累、直觉

## 需求评审：

### 什么是需求评审：

项目相关人员就产品需要进行确认和评估的相关活动

### 为什么要需求评审：

项目组成员理解需求，以便后期高效的进行、开发、设计工作

### 测试人员参与需求评审的职责

1. 确保主机的理解与产品设计理念一直明确实现的需求范围

2. 提出主机对产品需求不合理或遗漏

产品需求评审案例：只看功能是否正常，看控件位置，对流程的控制

有需求提需求，让产品更加完美

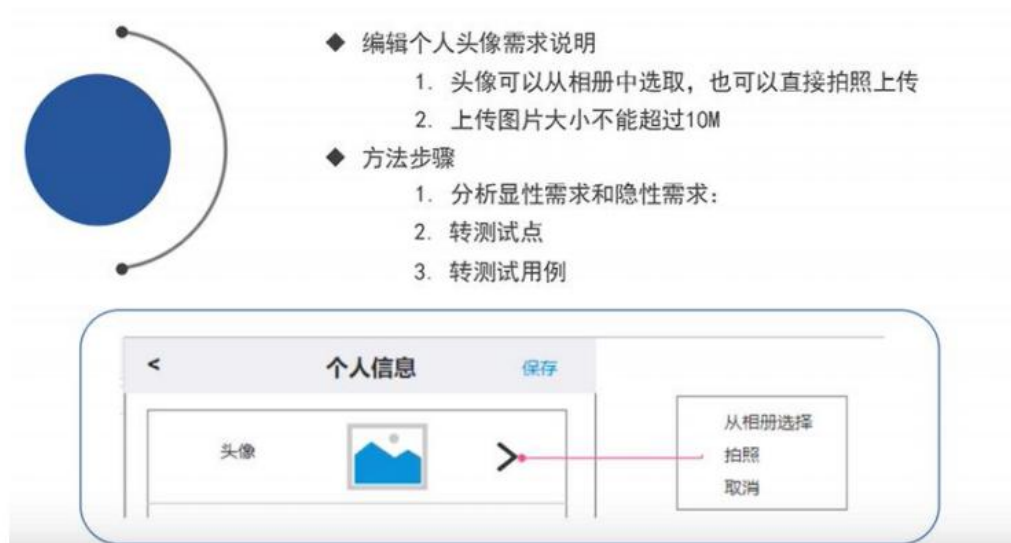
测试计划，就是一个 word 文档

测试用例测试思路

用例设计思路：帮助测试人员构建清晰的测试思维，指导测试思路



个人头像功能测试需求分析



分析显性，和隐性



显性：正常文章描述的东西

隐性： 那边边边角角没有显示出来的细节

测试用例的编写



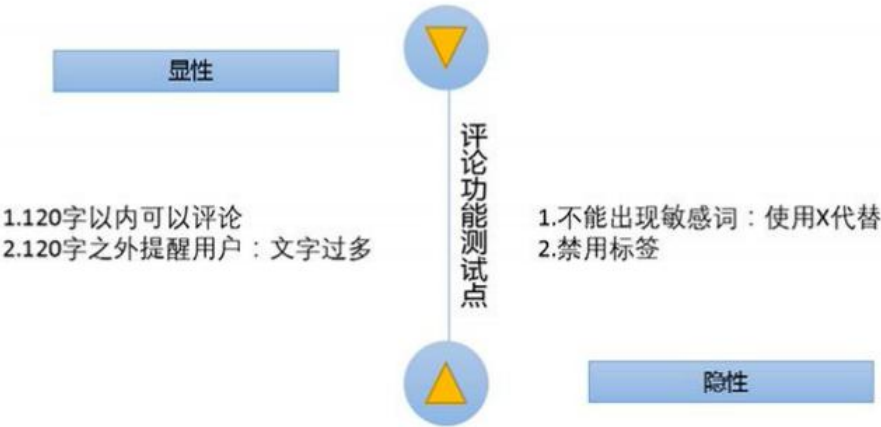
使用 Xmind 列出要测试的点，根据测试的需求来定，简单一句，就是找测试点  
最后使用 Excel 进行功能测试  
就是这么玩的

用例设计									
用例编号	功能模块	用例标题	用例级别	前置条件	测试数据	操作步骤	预期结果	测试结果	测试人员
1 我的-修改个人头像	验证有权限。进入个人相册	高	1 登录 2 进入到个人相册页面	/	1 点击头像按钮 2 点击从相册选择	进入个人相册			
2 我的-修改个人头像	验证无权限。进入个人相册	高	1 登录 2 进入到个人相册页面	/	1 点击头像按钮 2 点击从相册选择	不能进入个人相册 提醒用户从设置--相册--权限			
3 我的-修改个人头像	验证小于等于10M的图片上传	高	1 登录 2 进入到个人相册页面	/	1 点击头像按钮 2 点击从相册选择 3 选择一张小于等于10M的照片	上传成功 替换了头像			
4 我的-修改个人头像	验证大于10M的图片上传	高	1 登录 2 进入到个人相册页面	/	1 点击头像按钮 2 点击从相册选择 3 选择一张大于10M的照片	上传失败 提醒用户图片超过10M			
5 我的-修改个人头像	验证图片顺序为顺序	高	1 登录 2 进入到个人相册页面	/	1 点击头像按钮 2 点击从相册选择 3 选择一张大于10M的照片	顺序显示相册中的图片			
6 我的-修改个人头像	验证png图片格式的上传	高	1 登录 2 进入到个人相册页面	/	1 点击头像按钮 2 点击从相册选择 3 选择png格式的图片上传	正常替换头像 上传成功			

需求

◆ 评论需求说明

超过120字，提示用户：文字过多



测试用例

B6 X ✓ / ✗ 正常提交，不会弹出对话框									
A	B	C	D	E	F	G	H	I	J
ID	模块	优先级	测试场景	前置条件	步骤描述	测试数据	预期结果	测试结果	测试版本号
001	评论	高	验证文字120字	1. 已登录 2. 进入文章页面	1. 进入评论页面 2. 写一条120字的评论 3. 点击输入法右下角确定	/	新评论内容显示在评论列表第一条		1.2
002	评论	高	验证文字119字	1. 已登录 2. 进入文章页面	1. 进入评论页面 2. 写一条119字的评论 3. 点击输入法右下角确定	/	新评论内容显示在评论列表第一条		1.2
003	评论	高	验证文字121字	1. 已登录 2. 进入文章页面	1. 进入评论页面 2. 写一条121字的评论 3. 点击输入法右下角确定	/	弹窗提醒用户文字过多		1.2
004	评论	高	验证敏感词	1. 已登录 2. 进入文章页面	1. 进入评论页面 2. 写一条有敏感词的评论 3. 点击输入法右下角确定	/	敏感词以X代替		1.2
005	评论	高	验证是否禁用标签	1. 已登录 2. 进入文章页面	1. 进入评论页面 2. 写一条带标签的评论 3. 点击输入法右下角确定	<script>alert('弹窗啦')</script>	正常提交，不会弹出对话框		1.2

## 弱网测试

### 弱网测试和fiddler工具

#### ◆ 弱网测试

属于健壮性测试的内容。用户在地铁里，巴士上，甚至是电梯，车库等场景使用APP，我们就需要验证出现丢包、延时软件的处理机制，增强用户体验。

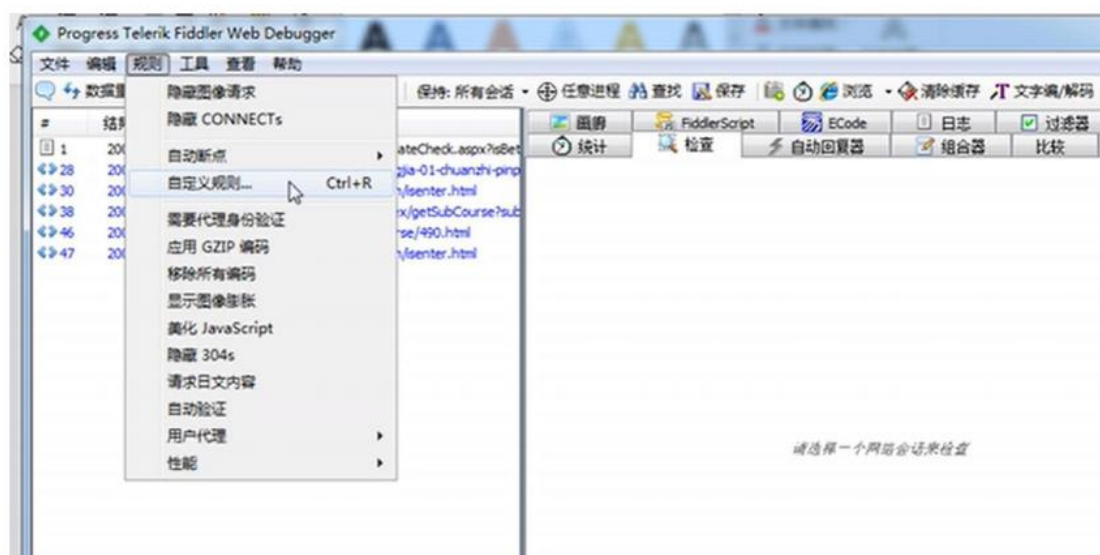


- 应用在网速慢的情
- 使用Fiddler进行

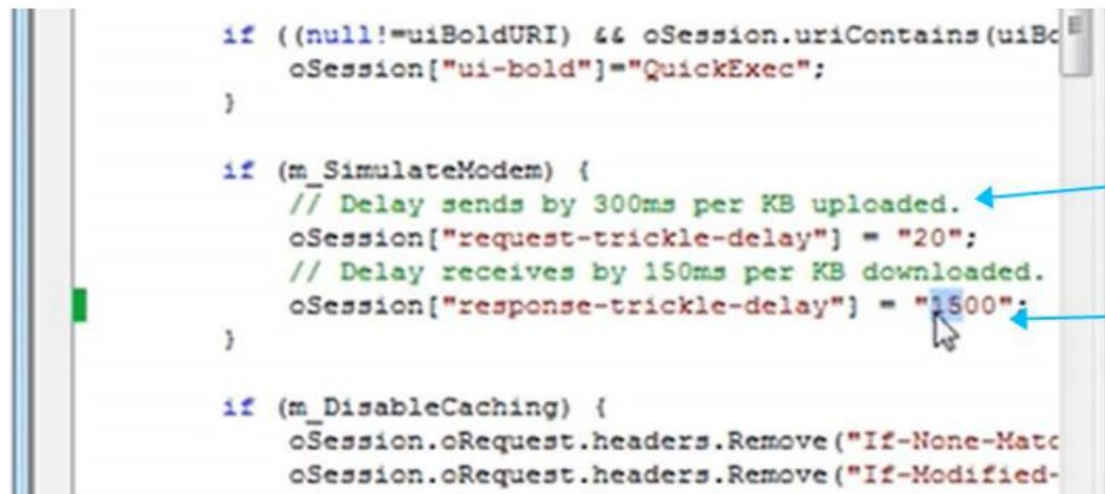
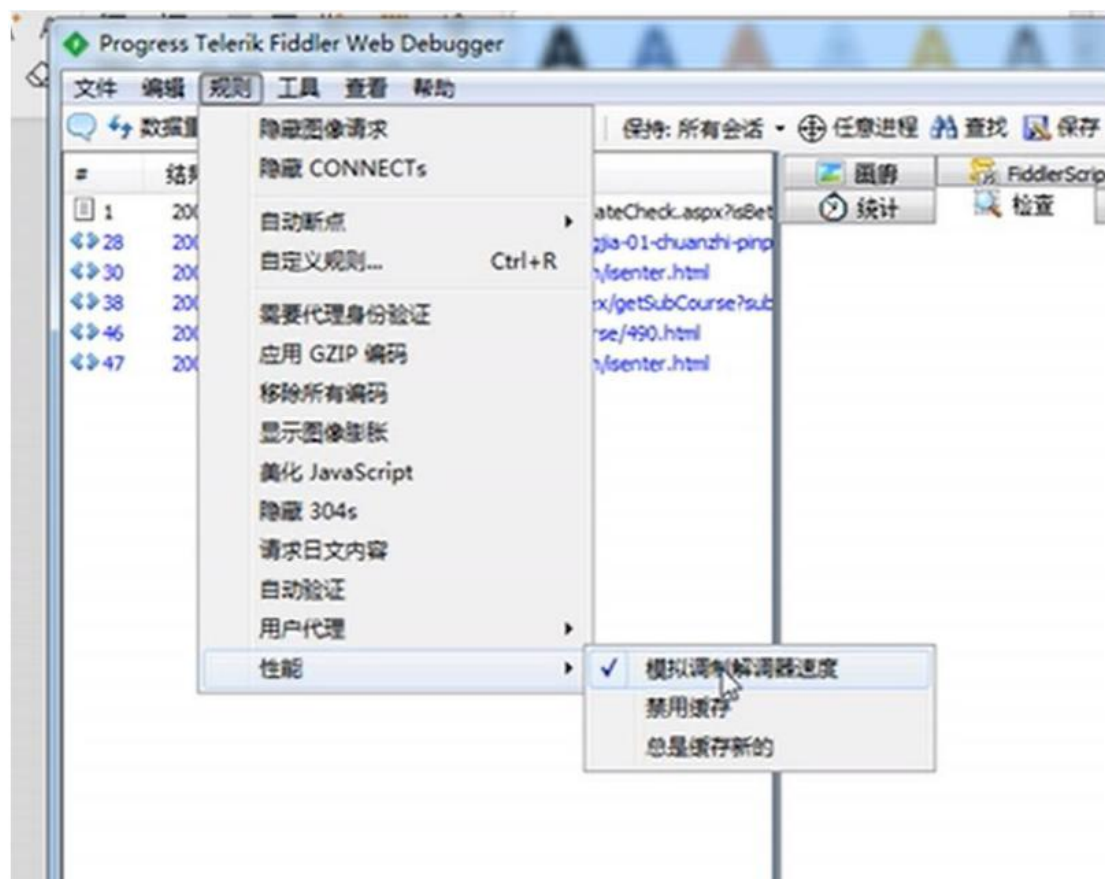
#### ◆ Fiddler

Fiddler提供了代理服务器，可以模拟弱网测试

使用 Fiddler 模拟 3G ， 4G 的访问速度，也就是具体时间







交叉测试

交叉事件测试

软件正在执行过程中，另外一个事件或操作对该过程进行干扰的测试



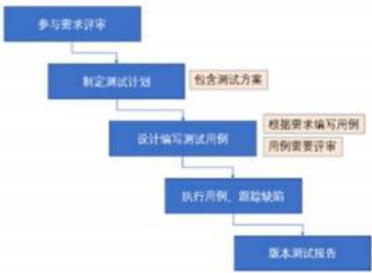
ID	模块	严重程度	优先级	Bug类型	状态	Bug标题	前置条件	再现步骤	预期结果	实际结果
1	我的-修改个人头像	一般	p1	代码错误	新建	有权限，无法进入个人相册	1. 登录 2. 进入到个人相册页面	1. 点击头像按钮 2. 点击从相册选择	可以进入个人相册	无法进入个人相册

◆ 用例执行：

开发提交测试包后，测试人员依据测试用例，验证被测软件功能实现是否与预期结果一致。

◆ 执行注意事项：

- ① 时间不足，及时上报，质量优先，进度其次
- ② 按优先级顺序执行
- ③ 未执行、无效、删除的用例备注原因





(4) 软件的开发和运行常常受到计算机系统的制约，对计算机系统有着不同程度的依赖

性。为了解除这种依赖性，在软件开发中提出了软件移植的问题。

(5)

软件的开发至今尚未完全摆脱人工艺的开发方式。

(6)

软件本身是复杂的。软件的复杂性可能来自它所反映的实际问题的复杂性，也可能

来自程序逻辑结构的复杂性。

(7) 软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，

它的成本是比较高的。

(8) 相当多的软件工作涉及到社会因素。许多软件的开发和运行涉及机构、体制及管理

方式等问题，它直接影响到项目的成败。

2. 软件的分类方法都有哪些？

软件的分类方法有如下 4 种：

(1) 按软件的功能分类

(2) 按软件服务对象的范围分类

(3) 按开发软件所需要的人力、时间以及完成的源程序行数分类。

(4) 按软件工作方式分类

按软件的工作方式分为：实时处理软件、分时软件、交互式软件、批处理软件。

3. 软件测试的概念

软件测试是软件工程中的一个环节，是开发项目整体的一部分。软件测试是有计划有组

织的，是保证软件质量的一种手段，它是软件工程中一个非常重要的环节。因此，可以认为

它是伴随软件工程的诞生而诞生的，伴随着软件复杂程度的增加、规模的增大，软件测试作

为一种能够保证软件质量的有效手段，越来越受到人们的重视，软件测试最终目的是使产品

达到完美。

4. 软件测试的方法有哪些？

软件的测试方法有 3 种，即用试题测试、用新旧两个系统作平行处理测试和软件测试自

动化工具测试。

5. 请简要说明软件测试阶段的任务。

软件测试阶段有以下几方面的任务：

(1)

制定测试大纲；

(2)

制作测试数据；

(3) 程序测试；

- (4) 功能测试;
- (5) 子系统测试;
- (6) 系统测试;
- (7) 系统接口测试;
- (8) 写出测试报告书;
- (9) 向下阶段工作提交系统运行、维护手册的草案。
- (10) 制定测试大纲。

6.

说明软件测试人员需要的知识结构。

★

需要具有懂得计算机的基本理论，又有一定开发经验的人员；

★

需要具有了解软件开发的基本过程和特征，对软件有良好的理解能力，掌握软件测

试相关理论及技术的人员；

★

需要具有软件业务经验的人员；

★

需要根据测试计划和方案进行软件测试；针对软件需求开发测试模型，制定测试方

案，安排测试计划，搭建测试环境，进行基本测试，设计简单的测试用例；

★ 需要具有规划设计环境；编制测试大纲并设计测试用例；对软件进行全面测试工作

的人员；

★ 需要具有编制测试计划；评审测试方案，规范测试流程及测试文档；分析测试结果，

管理测试项目；

★

需要会操作软件测试工具的人员。

7.

软件测试人员需要的素质都有哪些？请简要说出。

①

沟通能力

②

技术能力

③

自信心

④

洞察力

⑤

探索精神

⑥

不懈努力

⑦

创造性

⑧

追求完美

⑨

判断准确

⑩

老练稳重和说服力

## 第 2 章 白盒测试的实用技术

1.

白盒测试有哪两个分类？

(1) 静态测试

静态测试是测试中很重要的方法之一。它不要求在计算机上实际执行所测程序，主要以

一些人工的模拟技术对软件进行分析和测试。静态测试大约可以找出 25%—60% 的逻辑错误。

(2) 动态测试：

输入一组预先按照一定的测试准则设计的实例数据驱动运行程序，检查程序功能是否符合设计要求，发现程序中错误的过程。

2. 说出白盒测试的 4 个原则。

- 1) 保证一个模块中所有路径至少被测试一次；
- 2) 所有逻辑值都要测试真和假两种情况；
- 3) 检查程序的内部数据结构是否有效；
- 4) 再上、下边界及可操作范围内运行所有循环。

3. 详细说明白盒测试方法要注意的问题。

在白盒测试中，可以使用各种测试方法进行测试。但是，测试要考虑五点问题。

(1) 测试中，尽量先用自动化工具来进行静态结构分析；

(2) 测试中建议先从静态测试开始，如：静态结构分析、代码走查和静态质量度量，然

后进行动态测试，如：覆盖率测试；

(3) 利用静态分析的结果作为依据，再使用代码检查和动态测试的方式对静态分析结果

进行进一步确认，提高测试效率及准确性；

(4) 覆盖率测试是白盒测试中的重要手段，在测试报告中可以作为量化指标的依据，对

于软件的重点模块，应使用多种覆盖率标准衡量代码的覆盖率；

(5) 在不同的测试阶段，测试的侧重点不同：

★

在单元测试阶段，以代码检查、逻辑覆盖为主；

★

在集成测试阶段：需要增加静态结构分析、静态质量度量；

★

在系统测试阶段：在黑盒测试的基础上，白盒测试技术配合黑盒测试技术进行系统测试。

4.

请简要写出白盒测试常用的 7 类技术。

1)

逻辑覆盖法

2)

插桩技术

3)

基本路径测试法

4)

域测试法

5)

符号测试

6)

2 路径覆盖法

7)

程序变异测试法

5.

逻辑覆盖主要测试哪 8 各方面的覆盖率？

(1) 语句覆盖

(2)

判定覆盖

(3)

条件覆盖

(4)

条件判定组合覆盖

(5)

多条件覆盖

(6)

修正条件判定覆盖

(7)

组合覆盖

(8)

路径覆盖



### 第 3 章 黑盒测试的实用技术

1.

请详细叙述黑盒测试的基本概念。

黑盒测试（Black-Box Testing）又称为数据驱动测试或基于规格说明的测试。

黑盒测试

就是把程序看作一个不能打开的黑盒子，不考虑程序内部逻辑结构和内部特性的情况下，

测试程序的功能，测试者要在软件的接口处进行，它只检查程序功能是否按照规格说明书的

规定正常使用，程序是否能接收输入数据而产生正确的输出信息，以及性能是否满足用户的

需求，并且保持数据库或外部信息的完整性。通过测试来检测每个功能是否都能正常运行，

因此黑盒测试又可称为从用户观点和需求进行出发的测试。

2. 黑盒测试都有哪些优点？请说明。

黑盒测试的优点：

★

从产品功能角度测试可以最大程度满足用户的需求。

★

相同动作可重复执行，最枯燥的部分可由机器完成。

★

依据测试用例针对性地找寻问题，定位更为准确，容易生成测试数据。

★

将测试直接和程序/系统要完成的操作相关联。

3.

黑盒测试都有哪些缺点？请说明。

黑盒测试的缺点：

★

代码得不到测试。

★

如果规格说明设计有误，很难发现。

★

测试不能充分的进行。

★

结果取决于测试用例的设计。

4.

请详细说明黑盒测试的方法。

因为黑盒测试是一种基于证明功能需求和用户最终需求的测试方法，所以在选择测试，

设计测试方法方面有如下几种。

★

等价类划分法；

★

边界值分析法；

★

因果图法；

★

判定表驱动测试；

★

场景法；

★

功能图法；

★

错误推测法；

★

正交试验设计法。

在实际测试工作中，往往是综合使用各种方法才能有效地提高测试效率和测试覆盖

率，这就需要认真掌握这些方法的原理，积累更多的测试经验，以有效地提高测试水平和测试的效率。

5.

黑盒测试的原则都有哪些？

★

根据软件规格说明书设计测试用例，规格说明书的正确性是至关重要的。

★

有针对性地找问题，并且正确定位等价类

★

功能是否有缺陷或错误现象？

★

根据测试的重要性来确定测试等级和测试重点，减少程序可能出现的缺陷。

★

在接口处，输入的信息是否能正确接受？接受后能否输出正确的结果？

★

认真选择测试策略，尽可能发现程序的数据结构错误或外部信息访问错误，站在用户立场上进行测试。

## 第 4 章 测试用例设计方法

1.

什么是测试用例。

测试用例（Test Case）通俗一点来讲就是编写（编制）一组前提条件、输入、执行条

件、预期结果以完成对某个特定需求或目标测试的数据，体现测试方案、方法、技术和策略的文档。

2. 测试用例主要包括哪些内容。

完整的测试用例通常包括：

★

测试用例的编号；

★

测试日期；

★

测试用例设计人员和测试人员；

★

测试用例的优先级；

★

测试标题；

★

测试目标；

★

测试环境；

★

输入数据/动作；

★

测试的操作步骤；

★

测试预期的结果。

3.

请写出设计测试用例所需的文档资料。

设计测试用例所需要的文档资料包括：

★

软件需求说明书；

★

软件设计说明书；

★

软件测试需求说明书；

★

成熟的测试用例（案例库或财富库）。

4.

简述白盒测试用例的设计技术和目的。

（1）白盒测试用例的设计技术如下：

★

逻辑覆盖；

★

基本路径测试。

(2) 采用白盒测试技术设计用例的目的主要是：

- ★  
每个模块中的所有独立路径至少被执行一次；
- ★  
所有的逻辑值必须测试真、假两个分支；
- ★  
在边界值内和可操作范围至少循环一次；
- ★  
检查数据的内部结构保证其有效的实现预定功能。

5.

简述黑盒测试用例的设计技术和目的。

(1) 黑盒测试用例设计技术如下：

- ★  
等价类划分；
- ★  
边界值分析；
- ★  
错误推测；
- ★  
因果图。

(2) 采用黑盒测试技术设计用例的主要目的是：

- ★  
检查功能是否实现或遗漏；
- ★  
检查人机交互界面是否出错；
- ★  
数据库读取、更新操作出错；
- ★  
性能特性是否得到满足。

## 第 5 章 对本书测试模型的介绍

略

## 第 6 章 单元测试的实用技术

1.

简述单元测试的目的。

单元测试目的主要有以下几点：

- (1) 检查单元模块内部的错误，为软件的评审验收提供依据。
- (2) 单元测试是以程序设计说明书和之前所作的测试数据（正常的和错误的）

为指导，

测试模块内重要的路径，以检查出错误；

(3) 检验信息能否正确地流入和流出单元；

(4) 在单元测试工作过程中，其内部数据能否保持其完整性，包括内部数据的形式、

内容及相互关系不发生错误，也包括全局变量在单元中的处理和影响。

(5) 在为限制数据加工而设置的边界处，能否正确工作。

(6) 单元的运行能否做到满足特定的逻辑覆盖。

(7) 单元中发生了错误，其中的出错处理措施是否有效。

2. 简述单元测试的主要任务。

单元测试的主要任务有：程序语法检查、程序逻辑检查、模块接口测试、局部数据结构

测试、路径测试、边界条件测试、错误处理测试、代码书写规范检查。

3. 单元测试主要需要测试哪 8 点？

程序语法检查、程序逻辑检查、模块接口测试、局部数据结构测试、路径测试、边界条

件测试、错误处理测试，代码书写规范检查。

4. 局部数据结构测试主要表现形式是哪 6 个方面？

(1) 局部数据结构测试最常见的积累错误；

(2) 不适合或者不相容的类型说明；

(3) 变量无初值；

(4) 变量初始化或者缺省值有错；

(5) 不正确的变量名或不正确的截断；

(6) 出现上溢、下溢或地址异常。

5. 边界条件测试主要测试的是哪 3 点？

(1) 程序内有一个  $n$  次循环，这个  $n$  次循环应该是  $1 \sim n$ ，而不是  $0 \sim n$ ；

(2) 由小于 小于等于 等于 大于 大于等于 不等于确定的比较值出错；

(3) 出线上溢、下溢和地址异常问题。

## 第 7 章 功能测试的实用技术

1.

功能测试的基本概念是什么？请简述之。

功能测试就是对产品的各功能进行验证，根据功能测试用例，逐项测试，检查产品是否

达到用户要求的功能。

功能测试一般须在完成单元测试后集成测试前进行，而且是针对应用系统进行各功能测

试。一般应用系统有多个功能（子系统），功能测试是基于产品功能说明书，是在已知产品

所应具有的功能，从用户角度来进行功能验证，以确认每个功能是否都能正常使

用、是否实

现了产品规格说明书的要求、是否能适当地接收输入数据而产生正确的输出结果等。功能测

试，包括用户界面测试、各种操作的测试、不同的数据输入、逻辑思路、数据输出和存储等

的测试。对于功能测试，针对不同的应用系统，其测试内容的差异很大，但一般都可归为界

面、数据、操作、逻辑、接口等几个方面。

2. 功能测试的基本要求是什么？请简述之。

功能测试（Functional testing）是基于产品功能说明书并根据产品特征、操作描述和

用户方案，来测试产品的每个功能是否都能正常使用、是否达到了产品规格说明书的要求。

功能测试只需要考虑它的功能点不需要考虑软件的内部结构及代码等。功能测试包括用户界

面测试、各种操作的测试、不同的数据输入、逻辑思路、数据输出和存储等的测试。

3. 请说明功能测试的重点。

功能测试工作一般由程序员担当，测试的结果交系统设计、测试人员审核通过。

功能测试的重点应注意如下两大点内容：

A 整体性

(1)

符合标准和规范；

(2)

直观性；

(3)

一致性；

(4)

灵活性。

B 重点性

(1)

确认每个功能是否都能正常使用， 每项功能符合实际要求；

(2)

是否实现了产品规格说明书的要求；

(3)

否能适当地接收输入数据而产生正确的输出结果；

(4)

用户界面测试、是否有相应的提示框、适当的错误提示；

(5)

系统的界面是否清晰、美观；

(6)

菜单、按钮操作正常、灵活，能处理一些异常操作；

(7)

是否能接受不同的数据输入（能接受正确的数据输入，对异常数据的输入可以

进行提示、容错处理）；

(8)

数据的输出结果准确，格式清晰，可以保存和读取；

(9)

功能逻辑清楚，符合使用者习惯；

(10) 系统的各种状态按照业务流程而变化，并保持稳定；

(11) 支持各种应用的环境，能配合多种硬件周边设备，与外部应用系统的接口有效；

(12) 软件升级后，能继续支持旧版本的数据。

4. 请详细说明 Web 功能测试的方法主要包括的内容。

Web 功能测试通常又称为网站（网页）测试。测试的方法主要有如下几点：

1.

页面链接检查：每一个链接都要有对应的页面，并且页面之间切要正确。

2.

相关性检查：检查删除/增加其中每一项是否会对其他项产生影响，如果产生影响，

这些影响是否都正确。

3.

检查按钮的功能是否正确，如 Add, delete, save, update 功能键。

4.

字符串长度检查：输入超出所要求的字符串长度的内容，看系统检查字符串长度

时会不会出错。

5. 字符类型检查：在应该输入指定类型的地方输入其他类型的内容，例如在应该输

入浮点型的地方输入其他字符类型，看系统是否检查字符类型时是否报错。

6. 标点符号检查：输入内容包括各种标点符号，特别是逗号、句号、空格、回车键、

回格键。看系统处理是否正确。

7. 中文字符处理：在可以输入中文的地方输入中文，看是否出现乱码或出现错误。

8. 检查带出信息的完整性：在查看信息和更新信息时，查看所填写的信息是否全部

带出以及带出和添加的信息是否一致。

9. 信息重复：在一些需要命名并且名字是唯一的信息中输入重复的名字，看系统是

否处理、报错；重名包括是否区分大小写；以及在输入内容的前后输入空格，系统是否作出

正确处理。



10. 检查删除功能：在一些可以一次删除多个信息的地方，不选择任何信息，按“delete”键，看系统如何处理，是否出错；然后选择一个和多个信息，进行删除，看是否正确处理。

11. 检查添加和修改是否一致：检查添加和修改信息的要求是否一致，例如添加要求必填的项，修改也应该必填；添加规定为浮点型的项，修改也必须为浮点型。

12. 检查修改重名：修改时把不能重名的项改为已存在的内容，看能否处理、报错。

同时也要注意，会不会报和自己重名的错。

13. 重复提交表单：一条已经成功提交的纪录，回格后再提交，看看系统是否做了处理。

14. 检查多次使用回格键的情况：在有回格的地方回格，回到原来页面，再回格，重复多次，看会否出错。

15. Search 检查：在有 search 功能的地方输入系统存在和不存在的内容，看搜索结果

是否正确。如果可以输入多个搜索条件，可以同时添加合理和不合理的条件，看系统处理是否正确。

16. 输入信息位置：注意在光标停留的地方输入信息时，光标和所输入的信息会不会跳动。

17. 上传下载文件检查：上传下载文件的功能是否实现，上传文件能否打开。对上传文件的格式有何规定，系统是否有解释信息，并检查系统能否做到。

18. 必填项检查：应该填写的项没有填写时系统是否都做了处理，对必填项是否有提示信息。

19. 快捷键检查：是否支持常用快捷键，如 Ctrl+C，Ctrl+V 等，对一些不允许输入信息的字段，如选人，选日期对快捷方式是否也做了限制。

20. 回车键检查：在输入结束后直接按回车键，看系统处理如何，是否报错。

5. 请详细说明 Web 翻页功能测试的方法主要包括的内容。

A. 首页、上一页、下一页、尾页。

★

有无数据时控件的显示情况；

★

在首页时，首页和上一页是否能点击；

★

在尾页时，下一页和尾页是否能点击；

★

在非首页和非尾页时，四个按钮功能是否正确；

★

翻页后，列表中的记录是否仍按照指定的排序列进行了排序。

B. 总页数，当前页数

★

总页数是否等于总的记录数/指定每页条数；

★

当前页数是否正确。

C. 指定跳转页

★

是否能正常跳转到指定的页数；

★

输入的跳转页数非法时的处理。

D. 指定每页显示条数

★

是否有默认的指定每页显示条数；

★

指定每页的条数后，列表显示的记录数，页数是否正确；

★

输入的每页条数非法时的处理。

6.

请详细说明搜索功能测试的方法主要包括的内容。

对于搜索功能，主要通过以下八点测试：

1.

页面检查；

2.

默认条件搜索；

3.

修改可选条件搜索；

4.

修改输入条件搜索；

5.

修改区间条件搜索；

6.

组合可选、输入条件搜索；

7.

操作后检查搜索条件及查询结果；

8.

错误、空记录搜索。

## 第 8 章 集成测试技术

1. 请详细说明集成测试的内容。

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：

两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意

义上讲，组

件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组

的模块一起

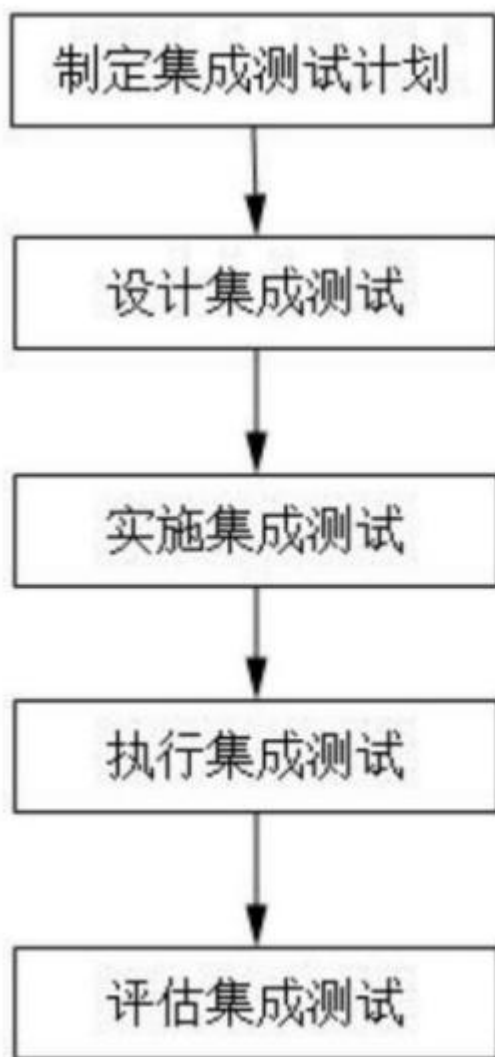
测试。最后，将构成进程的所有模块一起测试。此外，如果程序由多个进程组成，应该成对

测试它们，而不是同时测试所有进程。

2. 请说明集成测试的过程，可以用图表表示。

集成测试的过程包括：制定集成测试计划，设计集成测试，实施集成测试，执行集成测试，

评估集成测试。如下图所示。



3. 简述集成测试的五个步骤。

1. 首先确定子系统有哪些模块组成，保证这些模块都进行过单元测试。

2. 有开发人员组装这些模块，生成一个子系统，并保证在此子系统中，各个模块的功

能尽可能发挥出来。

3. 测试前，要设计测试用例，所以一个关键的模块为核心展开，以功能和性能为

两条

主线，注意模块间接口。

4. 搭建必要的测试环境，按照所写测试用例，进行模块连接的充分测试。

5. 记录测试结果，总结测试问题。

4. 请详细说明集成测试过程中要注意的事项。

## 1. 测试中问题的处理

（1）问题的定位，由谁定位，定位的时间

在测试过程中发现与测试计划中测试项预期结果有所不同，既是问题。如果测试人员有

能力定位问题，需明确程序代码中出错的地方，并记录下来；否则找开发人员到现场来定位。

定位的时间最好是在问题产生之前，这样有利于保护现场和问题重现，但时间不能太长，否

则影响测试进度，原则上说，集成测试中发现的问题都应该定位到语句，除非涉及到方案设

计上的错误。

（2）环境问题的处理

集成测试的环境可以是单机、双机或机架。测试过程中需要有独立的、稳定的和良好的

实验环境。但在实际中由于条件限制，测试环境是大家共享的，为保证本次测试不影响下次

测试工作或其他人测试工作的开展，所以测试人员需要做以下的工作：

★

测试环境的申请；

★

测试环境的维护；

★

测试环境的移交。

（3）测出问题的记录与提交

测试过程中发现的现象和问题由测试人员做详细记录，测出的问题最好先由开发人员确

认，然后以内部问题报告单的形式提交，这样防止测试人员提交的问题并非是程序的问题，

（可能是环境因素或其他因素造成），同时保证发现的问题能够被跟踪到回归测试，即被彻

底解决为止。

## 2. 测试过程记录

测试人员在测试过程中完成必要的测试记录，记录的内容包括：测试版本；测试任务；

使用环境；测试项目；测试结果；问题描述；产生原因。

每一阶段性的测试任务结束后，应向测试负责人提交测试记录，测试负责人做存

档处理。

3. 测试人员在测试过程中应不断地与开发人员进行经验交流，讨论程序中的疑问以及

问题的解决，加深程序的理解，以积极合作的方式来完成测试工作。

4. 测试用例、CHECKLIST、测试进度的适当修正

随着集成测试的进一步进行，对程序代码的理解不断加深，会发现以前的测试集不够理

想，这就需要及时更新测试用例，以提高测试覆盖率和达到需要的异常测试，相应也要修改

CHECKLIST 和调整测试进度。

5.

判断集成测试过程完成与否，需要注意哪些方面？

★

成功地执行了测试计划中规定的所有集成测试；

★

修正了所发现的错误；

★

测试结果通过了专门小组的评审。

## 第 9 章 性能测试的实用技术

1.

请详细说明性能测试的目的。

性能测试主要是验证软件系统是否能够达到用户提出的性能指标，同时发现软件系统中

存在的性能瓶颈及问题，找到软件的可扩展点，优化软件，最后起到优化系统的目的。

性能测试的目的主要有以下几点：

（1）评估系统的能力

性能测试主要考查系统的能力，它对系统的负荷和响应时间是相当重要的，也是验证系

统能力的依据之一。

（2）识别体系中的弱点

性能测试考查系统受控的负荷还存在有哪些缺陷，并为解决这些缺陷提供路径。

（3）系统调优

性能测试的系统调优就是重复运行测试，验证系统的活动是否得到了预期的结果，从而

改进系统性能。检测软件中的问题：长时间的测试执行可导致程序发生由于内存泄露引起的

失败，揭示程序中隐含的问题或冲突。

（4）验证稳定性及可靠性

验证稳定性及可靠性是在一个生产负荷下，执行一定时间的测试，是评估系统稳定性和

可靠性是否满足要求的唯一方法。

2. 请列举性能测试的先决条件。

性能测试的先决条件包括：

- (1) 针对性能测试对象的技术要成熟；
- (2) 性能测试的测试环境要稳定；
- (3) 进行性能测试的准备要充分；
- (4) 性能测试的目标要明确；
- (5) 性能测试的计划要详细；
- (6) 性能测试的数据要精确以及要有代表性；
- (7) 性能测试的描述要精练。

满足了这些之后我们能够进入测试阶段。

3. 请说明性能测试的主要分类，并简介之。

性能测试主要分为三类：

(1) 应用在客户端性能的测试

应用在客户端性能测试的目的是考察客户端应用的性能，测试的入口是客户端。它主要

包括并发性能测试、疲劳强度测试、大数据量测试和速度测试等，其中并发性能测试是重点。

(2) 应用在网络上性能的测试

应用在网络上性能的测试重点是利用成熟先进的自动化技术进行网络应用性能监控、网

络应用性能分析和网络预测。

(3) 应用在服务器端性能的测试

对于应用在服务器上性能的测试，可以采用工具监控，也可以使用系统本身的监控命令，

例如 Tuxedo 中可以使用 Top 命令监控资源使用情况。实施测试的目的是实现服务器设备、

服务器操作系统、数据库系统、应用在服务器上性能的全面监控。

4. 请列举在进行性能测试之前我们应掌握的相关文档。

- (1) 用户需求规格说明及其相关文档；
- (2) 软件开发的前期数据；
- (3) 前期工作的详细资料（单元测试、集成测试、功能测试等的相关文档）；
- (4) 在真正进入性能测试之前的软件数据的备份等；
- (5) 性能测试的测试大纲；
- (6) 性能测试的审批文稿及所签署的合同等。

5. 一个标准的性能调优过程是是什么？

- (1) 确定基准环境、基准负载和基准性能指标；
  - (2) 调整系统运行环境和实现方法，执行测试。
- （包括硬件环境的调优、

Weblogic

调优、Oracle 调优）；

(3) 记录测试结果、进行分析。

## 第 10 章 系统测试的实用技术

1.

简述系统测试的测试类型。

系统测试一般要考虑功能测试、性能测试、负载测试、容量测试、安全性测试、用户界

面测试、配置测试、安装测试、回归测试等。

2.

请分析系统测试的目标，并列举出来。

1)

确认系统测试的过程是按需求说明书进行的；

2)

确认新系统是否与需求说明书有不同或者缺陷；

3)

对新系统在进行测试的过程中出现的不足或不符合要的地方时行记录；

4)

建立完善的系统测试缺陷记录跟踪库；

5)

将测试过程中出现的问题进行修改，使之能达到令用户满足。

3.

系统测试策略的内容是什么？请详细说明。

测试策略用于说明测试工作的方法和目标，系统测试策略主要是对系统测试的需求，确

定测试类型和怎样进行测试的方法和技术。

测试策略应包括如下内容：

★

要进行的测试类型和测试目标；

★

进行测试时要采用的技术；

★

对测试的结果制定标准；

★

对测试过程中所出现问题存在的影响的特殊事项；

★

进行系统测试的对是应是完整的、集成的计算机系统；

★

按照设计说明书的规定，逐项测试系统的功能、性能等特性。

4.

系统测试的方法比较多，其中常用的方法是哪三个？



- 1)  
多任务测试
- 2)  
临界测试
- 3)  
中断测试。

5.

中断测试可分为：人为中断、硬件异常中断、程序执行中断以及意外中断 4 种情况，

请分别说明这 4 种情况。

1. 在测试中人为中段是为了表现测试结果而设置的中断。在测试中是常用的一种手段。
2. 而硬件异常中断是由计算机硬件异常或故障引起的中断，对硬件异常中断主要测试系统设备有没有在线功能和备用设备。
3. 程序执行中断主要是由程序中执行了中断指令引起的中断，也称为软中断。这是不应该发生的，对这类问题测试重点审查集成测试的过程和结果。
4. 意外中断主要是外部原因引起的中断，对他的测试主要是审查有没有备用电源、有没有安装 UPS。

## 第 11 章 验收测试的实用技术

1.

请详细列举验收测试的首要条件。

验收测试的首要条件有以下几点：

1.

软件开发已经完成，并全部解决了已知的软件缺陷；

2.

验收测试计划已经过评审并批准，并且置于文档控制之下；

3.

对软件需求说明书的审查已经完成；

4.

对概要设计、详细设计的审查已经完成；

5.

对所有关键模块的代码审查已经完成；

6.

对单元、集成、系统测试计划和报告的审查已经完成；

7.

所有的测试脚本已完成，并至少执行过一次，且通过评审；

8.

使用配置管理工具且代码置于配置控制之下；

9.

软件问题处理流程已经就绪；



10. 新系统已通过尝试运行工作；
11. 所被测的新系统应该是稳定的，要符合技术文档和标准的规定；
12. 已经制定、评审并批准验收测试完成标准；

13. 合同、附件规定的各类文档齐全。

2. 请详细说明验收测试的目的。

验收测试的目的主要是：

★ 新建系统产品是否是按照用户需求开发的，体验该产品是否能够满足用户使用要求、

有没有达到原设计水平、完成的功能怎样；

★

对照合同的需求进行验收测试，是否符合双方达成的共识；

★

新建系统产品的可靠性和可维护性好不好？

★

新建系统产品通过运行的结果表明，对业务处理的能力；

★

新建系统产品对用户操作的容错能力；

★

新建系统产品新系统对系统运行时发生故障的恢复能力；

★

承建单位向业主单位提交的有关技术资料是否俱全。

3.

请列举验收测试过程中所涉及到的相关文档。

测试过程中涉及到的文档有：

1.

测试任务说明书；

2.

测试计划说明书；

3.

测试用例说明书；

4.

测试报告说明书；

5.

测试总结说明书；

6.

测试验收说明书；

7.

缺陷跟踪报告说明书。

4.

正式验收测试是什么？它的优缺点又是什么？请介绍之。

正式验收测试，是系统测试的后续，也就是说正式测试的测试工作和系统测试差

不多，  
测试计划和测试用例设计都应很详细，在这个测试过程中应用的测试用例应是系统测试的用  
例的子集，不能对系统的测试方向有所偏离，在很多测试过程中，正式验收是自动进行测试  
的。

正式验收测试的优点是：

- 1)  
要进行验收测试的软件的功能和特性都是已知的；
- 2)  
可以对测试的过程进行评测；
- 3)  
正式验收测试可以自动进行测试；
- 4)  
对软件的要求是由用户需求说明书所决定的。

正式验收测试的缺点：

- 1)  
进行正式验收测试需要大量的资源和计划；
- 2)  
正式验收测试可能和系统测试差不多；
- 3)  
正式验收测试过程中可能不能发现某些缺陷。

5.

请介绍非正式验收测试的两个过程。

非正式验收测试过程分为 Alpha 测试 和 Beta 测试。

Alpha 测试

Alpha 测试是用户在开发环境下所进行的测试，或者是开发内部的人员在模拟实际环境

下进行的测试。Alpha 测试没有正式验收测试那样严格，在 Alpha 测试中，主要是对用户

使

用的功能和用户运行任务进行确认，测试的内容由用户需求说明书决定。

Beta

测试

进行

Beta 测试时，各测试员应负责创建自己的测试环境、选择数据，决定要研究的功

能、特性或任务，并负责确定自己对于系统当前状态的接受标准。

## 第 12 章 回归测试的实用技术

1.

请详细说明回归测试的定义。

在软件开发过程当中，只要软件发生改动，就可能给该软件带来诸多的问题，我们就必

须重新测试现有的功能模块。软件的改动可能是源于功能的变更、模块的增加或者 bug 的

修

改，具体表现在以下几个方面：

(1)

跟踪和管理系统不够健全，遗漏对 bug 的修改；

(2)

开发者对 bug 理解不够深入，只修改了 bug 的表面现象，而没有对 bug 做本质修改；

(3)

本 bug 被修改，之前版本 bug 掩盖的其他错误暴露出来；

(4)

bug 被修改，但并没有考虑到与此问题相关联的其他功能模块。

回归测试正是为了验证以上几个方面是否发生，以便确定修改是否达到了预期的目的，

验证修改是否损害了原有的正常功能。与此同时，还需要补充新的测试用例来测试新增的、

被修改了的功能模块。验证修改的正确性及其影响，即为回归测试。

回归测试不是特定的测试级别，软件开发的各个阶段都会进行多次回归测试。

2. 请说明回归测试的范围是什么。

在进行回归测试的时候，必须决定回归测试的范围，具体表现在以下几个方法：

(1)

测试所有修改或修正过的功能模块；

(2)

测试与被修改的模块相关的模块；

(3)

测试所有新增加的功能模块；

(4)

测试整个系统。

方法(1)、方法(2)和方法(3)中只进行了部分的回归测试，这样的测试是不健全的，因

为在软件系统中，对本地代码的修改可能导致整个系统产生副作用。

3. 请简要列举回归测试用例库的维护方法。

软件测试项目组在进行测试的过程中会将所用到的测试用例保存到“测试用例库”中，

并进行维护，回归测试用例库的维护方法如下。

1)

删除过时的测试用例

2)

改进不受控的测试用例

3)

删除冗余的测试用例

4)

增添新的测试用例

4.

列举常用的回归测试的方法。

再测试全部用例

基于风险进行测试

基于操作进行测试

仅测试修改部分

5. 请列出回归测试可遵循的基本过程。

(1)

识别出软件中被修改的部分。

(2)

从测试用例库中，排除所有不再适用的测试用例，确定那些对新的软件版本依然有

效的测试用例，其结果是建立一个新的测试用例库。

(3)

依据一定的策略从新的测试用例库中选择测试用例测试被修改的软件。

(4)

如果必要，可生成新的测试用例集，用于测试新的测试用例库无法充分测试的软件

部分。

(5) 用测试用例集执行修改后的软件。

其中第(2)和第(3)步测试验证修改是否破坏了现有的功能，第(4)和第(5)步测试验证修

改工作本身。

## 第 13 章 配置测试的实用技术

1.

配置测试的目标都有哪些？请列举之。

正如同所有测试的目标都是为了保证软件功能的强大，性能的优越，bug 报错率小配置

测试的目标也是相同的，它的目标有以下几点：

★

验证应用程序（即，确定它是否满足了它的配置要求）。

★

确定配置问题的软件出错。

★

帮助识别那些不能有效地在单元和集成测试发现的一些缺陷。

★

决定增加或修改，如硬件资源的影响：内存、磁盘和磁带资源、处理器、负

载均衡。

★

确定最佳的系统配置。

2.

请列举进行配置测试的几个前提条件。

进行配置测试需要以下几个前提条件：

★

进行配置测试的需求分析已经完成。

★

已完成应用程序的多个版本。

★

相关的软件组件已通过单元测试。

★ 软件集成测试已经进行，但在配置测试开始之前软件组件必须已经安装在被测硬件

设备上。

★

相关系统组件已通过系统集成测试。

★

在独立的测试小组配备足够的人员进行配置测试和训练。

★

配置测试环境准备完成。

3.

简要说明进行配置测试的两个范围所包括的内容。

配置测试的目标是为了使软件在尽可能多的硬件平台上运作，那么进行配置测试一般需

要测试它的硬件环境和软件环境。

1. 硬件环境

硬件环境主要包括：

★

不同的主机；

★

不同的组件；

★

不同的外设；

★

不同的接口以及可选项的测试。

2. 软件环境

软件环境包括：

★

对操作系统平台的兼容测试；

★

对同一操作系统平台不同版本的测试；

★

软件自身向前向后更新操作时的测试；

★

同其他软件产品兼容性测试以及数据兼容性（主要是数据共享）的测试。

4.

列举进行配置测试工作前和工作后所需的相关文档。

1. 工作开始前所需的文档

配置测试进行前需要以下文档资料：

★

测试计划；

★

需要进行的测试列表；

★

被测程序源码；

★

配置测试软硬件设备清单；

★

配置测试用例。

2. 工作结束后递交的文档

配置测试结束后需要递交以下文档资料：

★

配置测试报告；

★

配置测试总结报告。

5.

配置测试设计的要点包括哪 8 点？请说明之。

★

确定哪些功能是软件需要用到的，例如一个办公程序可能对显卡要求是很低的，没有必要去测试太多。又或者一个大型游戏根本不需要打印功能，那么就不需要管打印机了；

★ 看看要对哪些牌子，型号，具体那些驱动程序的硬件是可用的。一般都会选用市场上比较流行的软件；

★

看看哪些硬件特性，模式和选项是可用的；

★

在已有的测试集合里面挑选出一个可维护可管理的测试集，还是挑出表常见的硬件；

★

找出软件中对配置特别敏感的特有功能；

★

不同配置下的测试用例需要分别设计；

★

在每个配置环境下至少执行一边测试用例；

★

反复执行测试用例直到结果具有说服力。

## 第 14 章 软件的其他测试技术

1.

简要说明可用性测试的概念。

可用性测试的概念主要表现为：

1.

可用性是产品的一个基本的自然属性，是最终用户使用产品的可用的程度。

2.

可用性测试是依照可用性标准对 GUI 的系统评估。

3.

可用性是在产品和用户的相互作用中体现出来。

4.

可用性测试是用户在和系统（网站，软件应用程序，移动技术或任何用户操作的

设备）5. 交互时对用户体验质量的度量。

6.

可用性的基本评价指标是效率、满意和安全（容错，无错）。

2.

什么是压力测试？请说明之。

压力测试（Stress Test）也就是强度测试，压力测试是指模拟巨大的工作负荷来测试

应用程序在峰值情况下如何执行操作。在实际的软硬件环境下，压力测试主要是以软件响应

速度为测试目标，尤其是针对在较短时间内大量并发用户访问时软件的抗压能力。其目的是

找到系统在哪里失效以及如何失效的地方。

3. 请详细说明确认测试的内容（功能测试和性能测试）。

确认测试内容主要包括功能和性能两部分。

（1）功能测试

功能测试考察软件对功能需求完成的情况，应该设计测试用例使需求规定的每一个软件

功能得到执行和确认。

★

按照系统给出的功能列表，逐一设计测试案例；

★

对于需要资料合法性和资料边界值检查的功能，增加相应的测试案例；

★

运行测试案例；

★

检查测试结果是否符合业务逻辑；

★

评审功能测试结果。

## （2）性能测试

性能测试是检验软件是否达到需求规格说明中规定的各类性能指标，并满足一些与性能相关的约束和限制条件。

★

测试软件在获得定量结果时程序计算的精确性；

★

测试在有速度要求时完成功能的时间；

★

测试软件完成功能时所处理的数据量；

★

测试软件各部分工作的协调性，如高速操作、低速操作的协调性；

★

测试软件/硬件中因素是否限制了产品的性能；

★

测试产品的负载潜力及程序运行时占用的空间。

## 4.

请简要说明容错性测试的内容。

容错性测试包括两个方面：

输入异常数据或进行异常操作，以检验系统的保护性。如果系统的容错性好，系统只给

出提示或内部消化掉，而不会导致系统出错甚至崩溃。

灾难恢复性测试。通过各种手段，让软件强制性地发生故障，然后验证系统已保存的用

户数据是否丢失，系统和数据是否能尽快恢复。

## 5. 请详细说明容错性测试需考虑的特殊事项。

从容错性测试的概念和内容可以看出，当软件出现故障时如何进行故障的转移与恢复有

用的数据是十分重要的。对于如何进行容错性测试，这是我们关心的事情，所以进行容错性

测试需要考虑以下的特殊事项。

故障发生时数据的转移与数据的恢复

故障发生时数据的转移是为了确保在出现故障时能成功的转移有效的数据，防止因故障

的发生导致意外的破坏各种硬件、软件和网络设备。数据的恢复是为了能够继续运行系统，

同时，一旦系统发生故障，备用系统将不失时机地“顶替”已发生故障的系统。

容错性测试目前主要做的事情表现为：

服务器断电；

网络设备断电；

数据库系统发生故障；



应用系统文件发生故障

系统软件发生故障。

6. 请用简短的语言介绍一下易用性测试。

易用性是交互的适应性、功能性和有效性的集中体现。易用性一般分为两个层次，即用

户界面的易用性和操作系统的易用性。易用性测试包括针对应用程序的测试，同时还包括对

用户手册系统文档的测试。通常采用质量外部模型来评价易用性。

7. 请详细说明易用性测试中的用户界面测试的内容。

用于与软件交互的方式称为用户界面或 UI，易用性包括如下方面的测试：

（1）符合标准和规范

用户界面要素要符合软件现行的标准和规范。

（2）直观

用户界面是否洁净、不拥挤；

布局是否合理；

是否有多余功能。

（3）一致

如果软件或者平台有一个标准，就要遵守它。如果没有，就要注意软件的特性，确保相

似的操作以相似的方式进行。

（4）灵活

多种视图的选择；

状态跳转；

状态终止和跳过；

数据输入和输出。

（5）舒适

软件使用起来应该舒适，不能给用户工作制造障碍和困难。

（6）实用

是否实用是优秀用户界面的最后一个要素。

8. 请详细说明安全性测试的方法。

（1）应用程序（应用系统）级别的安全性测试方法

★ 对数据或业务功能的访问，在预期的安全性情况下，操作者只能访问应用程序

的特定功能、有限的数据；

★

操作者只能访问其所属用户类型已被授权访问的那些功能或数据；

★

不同权限的用户类型，创建各用户类型并用各用户类型所特有的事务来核实其

权限，最后修改用户类型并为相同的用户重新运行测试。

★

测试结果的安全性分析

●

分析所有测试用例，测试是否通过。

●

测试代码是否按照要求分析，并达到相应的测试覆盖率。

●

对测试结果进行分析，以验证所有的安全性需求是否得到了满足。

(2) 系统级别的安全性测试策略和方法

★ 只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访

问，包括对系统的登录或远程访问；

★

只有具备系统和应用程序访问权限的操作者才能访问系统和应用程序。

9.

请说明需求分析测试的内容。

需求分析测试的内容主要讨论以下 3 点：

1. 功能是否能满足用户的需求？

2. 性能是否能满足用户的需求？

3. 需求说明书所讨论的内容是否得到了用户的认可？

10. 请详细说明软件可靠性测试中需注意的问题。

软件可靠性测试需要注意的问题主要有 3 点：

1. 功能识别

软件可靠性测试首先考虑的是功能识别，确定系统所使用的功能。

功能识别的目标是：

★

识别系统所确定的功能（依据系统功能说明书进行审核）；

★

识别系统功能所需的相关条件。

2.

可靠性对时间的要求

软件可靠性对时间的要求是比较高的，测试时应将“运行时间”作为衡量可靠性的重要

指标，所谓运行时间就是软件运行时应在“规定的时间”内完成所要完成的工作。

对于时间

的要求应根据系统性能说明书的要求进行审核。

3. 可靠性对环境条件的要求

环境条件是指软件系统运行时所需的各种支持要素，主要表现为：硬件环境（服务器、

路由器、交换机、防火墙、磁盘阵列）、网络操作系统、软件工具、应用系统的操作规程等。

11. 请说明风险测试的内容。

风险是指在软件开发过程中遇到的预算、进度、开发不成功等方面的问题引起损失的可

能性，这种风险会导致软件开发的失败。

软件测试的风险是指软件测试过程出现的或潜在的问题，造成的原因主要是测试计划的

不充分、测试方法有误或测试过程的偏离，造成测试的补充以及结果不准确。测试的不成功

导致软件交付潜藏着问题，一旦在运行时爆发，会导致软件失败。

软件测试风险主要是对测试计划执行的风险分析与制定要采取的应急措施，降低软件测

试产生的风险造成的危害。

12. 请列举缺陷测试应注意的问题。

在缺陷测试过程中需要注意的问题有：

1)

由于市场的压力而造成的产品最终发行的时间限制；

2)

因测试员不正确操作或错误理解引出的缺陷；

3)

错误的修改影响的模块较多，带来的风险较大；

4)

在缺陷报告中提出很难被重现的问题；

5)

修改性价比太低的缺陷。

13.

请简要说明 Web 测试的内容。

Web

测试与一般应用系统的测试不同，链接的吻合性是 web 应用系统的一个主要特征，

需要检查和验证是否按照设计的要求运行，而且测试系统在不同用户的浏览器的显示是否合适。更重要的是，还要从最终用户的角度进行安全性和可用性测试。

14. 请说明接口测试的目的。

接口测试（Interface-Testing）的目的是：

●

测试系统相关联的外部接口；

●

测试的重点是要检查数据的交换；

●

传递和控制管理过程；

●

提高测试质量；

●

提高测试覆盖；

●

更好地重现软件缺陷；

●

更好定位错误。

作为接口测试主要考虑的问题是模块接口、系统接口。

15. 请详细介绍接口测试的测试项目。

接口测试的测试项目主要包括以下几点：

1. 数据类型问题

★

变量的数据类型是否错误；

★

是否存在不同数据类型的赋值；

★

是否存在不同数据类型的比较。

2. 变量值问题

★

变量的初始化或缺省值是否有错误；

★

变量是否发生上溢或下溢；

★

变量的精度是否足够。

3. 逻辑判断问题

★

是否由于精度原因导致比较无效；

★

表达式中的优先级是否有误；

★

逻辑判断结果是否颠倒。

4. 文件 I/O 问题

★

对不存在的或者错误的文件是否进行操作；

★

文件是否以不正确的方式打开；

★

文件结束判断是否正确；

★

是否正确地关闭了文件。

16. 请列举安装和反安装测试的 4 个目标。

安装和反安装测试的目标有 4 点：

1.

安装/卸载程序能正确运行；

2.

程序安装正确；卸载时完全清除；

3.

程序安装后能正确运行；卸载后系统的影响；

4.

完善性安装后程序能正确运行。

1.

对安装进行测试要注意如下内容：

- (1) 安装程序是否正确；
- (2) 程序安装后能否正确运行；
- (3) 安装过程是否符合安装手册的安装步骤；
- (4) 安装过程中所有缺省选项是否得到了验证；
- (5) 安装过程中典型的选项是否得到了验证；
- (6) 安装过程中是否出现异常配置状态（非法和不合理配置）；
- (7) 安装后是否能产生正确的目录结构和文件属性；
- (8) 安装后的动态库是否正确；
- (9) 安装后的软件能否正确运行；
- (10) 安装后该系统是否对其他的应用程序造成不正常影响。

2. 对反安装进行测试要注意如下内容：

- (1) 文件——安装目录里的文件及文件夹；
- (2) 非安装目录（向系统其它地方添加的文件及文件夹）；
- (3) 快捷方式——（桌面，菜单，任务栏，系统栏，控件面板，系统服务列表等）；
- (4) 复原方面——卸载后，系统能否恢复到软件安装前的状态；
- (5) 卸载方式——程序自带卸载程序/系统的控件面板卸载/其它自动卸载工具；
- (6) 卸载状态——程序在运行/暂停/终止等状态时的卸载；
- (7) 非正常卸载情况——卸载软件过程中，取消卸载进程，然后，观察软件能否

继续使用；

常使用；

(8) 冲击卸载——在卸载的过程中，中断电源，然后，启动计算机后，重新卸载软件，

如果软件无法卸载，则重新安装软件，安装之后再重新卸载；

(9) 卸载环境——不同的(操作系统，硬件环境，网络环境等)下进行卸载；卸载后，

该系统是否对其他的应用程序造成不正常影响。

## 测试面试题及一些资料

01. 为什么要在一个团队中开展**软件测试工作**？

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

02. 您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

我曾经做过 web 测试，后台测试，客户端软件，其中包括**功能测试**，**性能测试**，用户体

验测试。最擅长的是功能测试

03. 您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同 04. 的测试类型的区别与联系（如功能测试、性能测试.....）

测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。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。采用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。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范（快捷键之类的），是否美观（能否吸引用户的注意力），是否安全（尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告）？做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的，然后再考虑该功能点的性能测试

04. 您认为做好测试用例设计工作的关键是什么？

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

05. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

- 1、是否有不正确或遗漏的功能？
- 2、在接口上，输入是否能正确的接受？能否输出正确的结果？
- 3、是否有数据结构错误或外部信息（例如数据文件）访问错误？
- 4、性能上是否能够满足要求？
- 5、是否有初始化或终止性错误？

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行



如下检查：

- 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”与取“假”的两种情况都能至少测一遍。
- 3、在循环的边界和运行的界限内执行循环体。
- 4、测试内部数据结构的有效性，等等。

单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与[其他](#)组的模块一起测试。最后，将构成进程的所有模块一起测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。（常见的联调测试）

系统测试的目的是对最终软件系统全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

06. 测试计划工作的目的是什么？测试计划工作的内容都包括什么？其中哪些是最重要的？

软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计

划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

所以其中最重要的是测试策略和测试方法（最好是能先评审）

07. 您认为做好测试计划工作的关键是什么？

1. 明确测试的目标，增强测试计划的实用性

编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

2. 坚持“5W”规则，明确内容与过程

“5W”规则指的是“What（做什么）”、“Why（为什么做）”、“When（何时做）”、“Where

（在哪里）”、“How（如何做）”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的（Why），明确测试的范围和内容（What），确定测试的开始和结束日期（When），指出测试的方法和工具（How），给出测试文档和软件的存放位置（Where）。

### 3. 采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

### 4. 分别创建测试计划与测试详细规格、测试用例

应把详细的**测试技术**指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理**数据库**中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

08. 您所熟悉的测试用例设计方法都有哪些？请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

#### 1. 等价类划分

划分等价类：等价类是指某个输入域的子集合。在该子集中，各个输入数据对于揭露程序中的错误都是等效的。并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试。

因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，就可以用少量代表性的测试数据，取得较好的测试结果。等价类划分可有两种不同的情况：有效等价类和无效等价类。

#### 2. 边界值分析法

边界值分析方法是等价类划分方法的补充。测试工作经验告诉我，大量的错误是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。

使用边界值分析方法设计测试用例，首先应确定边界情况。通常输入和输出等价类的边界，就是应着重测试的边界情况。应当选取正好等于、刚刚大于或刚刚小于边界的值作为测试数据，

而不是选取等价类中的典型值或任意值作为测试数据。

#### 3. 错误推测法

基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法。

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。

#### 4. 因果图方法



前面介绍的等价类划分方法和边界值分析方法,都是着重考虑输入条件,但未考虑输入条件之间的联系,相互组合等.考虑输入条件之间的相互组合,可能会产生一些新的情况.但要检查输入条件的组合不是一件容易的事情,即使把所有输入条件划分成等价类,他们之间的组合情况也相当多.因此必须考虑采用一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例.这就需要利用因果图(逻辑模型).因果图方法最终生成的就是判定表.它适合于检查程序输入条件的各种组合情况.

09. 请以您以往的实际工作为例, 10. 详细的描述一次测试用例设计的完整的过程。

就说最近的这次网站功能的测试吧

首先: 得到相关文档(需求文档和设计文档), 理解需求和设计思想后, 想好测试策略(测试计划简单点就 OK 了), 考虑到测试环境, 测试用例, 测试时间等问题。

第二步: 设计测试用例, 测试策略是: 把网站部分的功能点测试完, 然后在进行系统测试(另外个模块呢有另一个测试人员负责, 可以进行联调测试), 网站模块的测试基本是功能测试和界面测试(用户并发的可能性很小, 所以不考虑): 这次的网站的输入数据呢是使用数据库中的某张表记录, 如果表中某一数据记录中新加进来的(还没有被处理的, 有个标志位), 网站启动后会立刻去刷那张表, 得到多条数据, 然后在进行处理。处理过程中, 会经历 3 个步骤, 网站才算完成了它的任务。有 3 个步骤呢, 就可以分别对这 3 个步骤进行测试用例的设计, 尽量覆盖到各种输入情况(包括数据库中的数据, 用户的输入等), 得出

了差不多 50 个用例。界面测试, 也就是用户看的到的地方, 包括发送的邮件和用户填写资料的页面展示。

第三步: 搭建测试环境(为什么这个时候考虑测试环境呢? 因为我对网站环境已经很熟了, 只有有机器能空下来做该功能测试就可以做了), 因为网站本身的环境搭建和其他的系统有点不同, 它需要的测试环境比较麻烦, 需要

web 服务器(Apache,tomcat), 不过这

次需求呢, 网站部分只用到了 tomcat, 所以只要有 tomcat 即可

第四步: 执行测试

11. 您以往是否曾经从事过性能测试工作? 如果有, 12. 请尽可能的详细描述您以往的性能测试工作的完整过程。

是的, 曾经做过网站方面的性能测试, 虽然做的时间并不久(2 个月吧), 当时呢, 是有位网站性能测试经验非常丰富的前辈带着我一起做。

性能测试类型包括负载测试, 强度测试, 容量测试等

负载测试: 负载测试是一种性能测试指数据在超负荷环境中运行, 程序是否能够承担。

强度测试: 强度测试是一种性能测试, 他在系统资源特别低的情况下软件系统运行情况

容量测试: 确定系统可处理同时在线的最大用户数

在网站流量逐渐加大的情况下, 开始考虑做性能测试了, 首先要写好性能测试计划, 根据运营数据得出流量最大的页面(如果是第一次的话, 一般是首页, 下载页, 个人帐户页流量最大, 而且以某种百分比),

Web 服务器指标指标:

\* Avg Rps: 平均每秒钟响应次数=总请求时间 / 秒数;

- \* Successful Rounds：成功的请求；
- \* Failed Rounds ：失败的请求；
- \* Successful Hits ：成功的点击次数；
- \* Failed Hits ：失败的点击次数；
- \* Hits Per Second ：每秒点击次数；
- \* Successful Hits Per Second ：每秒成功的点击次数；
- \* Failed Hits Per Second ：每秒失败的点击次数；
- \* Attempted Connections ：尝试链接数；

13. 您在从事性能测试工作时是否使用过一些测试工具？如果有请试述该工具的工作原理，并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。

17.

您认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？

18.

在您以往的工作中，一条软件缺陷（或者叫 Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？

20. 您以往所从事的软件测试工作中是否使用了一些工具来进行软件缺陷（Bug）的管理？如果有请结合该工具描述软件缺陷（Bug）跟踪管理的流程。

23. 您认为在测试人员同开发人员的沟通过程中如何提高沟通的效率和改善沟通的效果？维持测试人员同开发团队中其他成员良好的人际关系的关键是什么？

27. 在您以往的测试工作中最让您感到不满意或者不堪回首的事情是什么？

您是如何来对待这些事情的？

31. 在即将完成这次笔试前，32. 您是否愿意谈一些自己在以往的学习和工作中获得的工作经验和心得体会？（可以包括软件测试、过程改进、软件开发或者与此无关的其他方面）

33. 你对测试最大的兴趣在哪里？为什么？

最大的兴趣就是测试有难度，有挑战性！做测试越久越能感觉到做好测试有多难。曾经在无忧测试网上看到一篇文章，是关于如何做好一名测试工程师。一共罗列了 11, 12 点，有部分是和人的性格有关，有部分需要后天的努力。但除了性格有关的 1, 2 点我没有把握，其他点我都很有信心做好它。

刚开始进入测试行业时，对测试的认识是从无忧测试网上了解到的一些资料，当时是冲着做测试需要很多技能才能做的好，虽然入门容易，但做好很难，比开发更难，虽然当时我很想做开发（学校专业课我基本上不缺席，因为我喜欢我的专业），但看到测试比开发更难更有挑战性，想做好测试的意志就更坚定了。

不到一年半的测试工作中，当时的感动和热情没有减退一点（即使环境问题以及自身经验，技术的不足，做测试的你一定也能理解）。

我觉得做测试整个过程中有 2 点让我觉得很有难度（对我来说，有难度的东西我就非常感兴趣），第一是测试用例的设计，因为测试的精华就在测试用例的设计上了，要在版本出来之前，把用例写好，用什么测试方法写？（也就是测试计划或测试策略），如果你刚测试一个新任务时，你得花一定的时间去消化业务需求和技术基础，业务需求很好理解（多和产品经理和开发人员沟通就能达到目的），而技术基础可就没那么简单了，这需要你自觉的学习能力，比如说网站吧，最基本的技术知识你要知道网站内部是怎么运作的，后台是怎么响应用户请求的？测试环境如何搭建？这些都需要最早的学好。至少在开始测试之前能做好基本的准备，可能会遇到什么难题？需求细节是不是没有确定好？这些问题都能在设计用例的时候发现。

第二是发现 BUG 的时候了，这应该是测试人员最基本的任务了，一般按测试用例开始

测试就能发现大部分的 bug，还有一部分 bug 需要测试的过程中更了解所测版本的情况获得更

多信息，补充测试用例，测试出 bug。还有如何发现 bug？这就需要在测试用例有效的情况下，

通过细心和耐心去发现 bug 了，每个用例都有可能发现 bug，每个地方都有可能出错，所以测试

过程中思维要清晰（测试过程数据流及结果都得看仔细了，bug 都在里面发现的）。如何描述

bug 也很有讲究，bug 在什么情况下会产生，如果条件变化一点点，就不会有这个 bug，以

哪些最少的操作步骤就能重现这个 bug，这个 bug 产生的规律是什么？如果你够厉害的话，可

以帮开发人员初步定位问题。

34. 你的测试[职业发展](#)是什么？

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级测试工程师奔去。而且我也有初步的职业规划，前 3 年累积测试经验，按如何做好测试工程

师的 11, 12 点要求自己，不断的更新自己改正自己，做好测试任务。

35. 你自认为测试的优势在哪里？

优势在于我对测试坚定不移的信心和热情，虽然经验还不够，但测试需要的基本技能我有信心在工作中得以发挥。

36. 你以前工作时的测试流程是什么？

公司对测试流程没有规定如何做，但每个测试人员都有自己的一套测试流程。我说下我 1 年来不断改正（自己总结，吸取同行的方法）后的流程吧。需求评审（有开发人员，产品经理，测试人员，项目经理）—>需求确定(出一份确定的需求文档)—>开发设计文档（开发人员在开始写代码前就能输出设计文档）—>想好测试策略，写出测试用例—>发给开发人员和测试经理看看（非正式的评审用例）—>接到测试版本—>执行测试用例（中间可能会补充用例）—>提交 bug（有些 bug 需要开发人员的确定（严重级别的，或突然发现的在测试

用例范围之外的，难以重现的），有些可以直接录制进 TD）—>开发人员修改（可以在测试

过程中快速的修改）—>回归测试（可能又会发现新问题，再按流程开始跑）。

37. 当开发人员说不 38. 是

BUG 时，39. 你如何应付？

开发人员说不是 bug，有 2

种情况，一是需求没有确定，所以我可以这么做，这个时候

可以找来产品经理进行确认，需不需要改动，3 方商量确定好后再看要不要改。二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是 BUG 的依据是什么？

如果被用户发现或出了问题，会有什么不良结果？程序员可能会给你很多理由，你可以对他的解释进行反驳。如果还是不行，那我可以给这个问题提出来，跟开发经理和测试经理进行确认，如果要修改就改，如果不要修改就不改。其实有些真的不是 bug，我也只是建议的方式写进 TD 中，如果开发人员不修改也没有大问题。如果确定是 bug 的话，一定要坚持自己的立场，

让问题得到最后的确认。

23. 你为什么想离开目前的职务？

因为公司运作情况并不理想，公司需要调整部门体系，公司考虑到缩减部门人员，所以大批量的裁员（有 6, 7 个），这是我的第一份工作，对公司也有较深的感情，因为在这里我

找到了职业理想（就是测试），所以公司需要精简人员，我自愿退出。虽然很舍不得，但我将会有新的发挥能力的舞台。

24: 你对我们公司了解有多少？

25: 你找工作时，最重要的考虑因素为何？

工作的性质和内容是否能让我发挥所长，并不断成长。

26: 为什么我们应该录取你？

您可以由我过去的工作表现所呈现的客观数据，明显地看出我全力以赴的工作态度。

27: 请谈谈你个人的最大特色。

我的坚持度很高，事情没有做到一个令人满意的结果，绝不罢手。

28.白箱测试和黑箱测试是什么?什么是回归测试?

29.单元测试、集成测试、系统测试的侧重点是什么？

30.设计用例的方法、依据有那些？

31.一个测试工程师应具备那些素质和技能？

32.集成测试通常都有那些策略？

33.你用过的测试工具的主要功能、性能及其他？

34.一个缺陷测试报告的组成

35.基于 WEB 信息管理系统测试时应考虑的因素有哪些？

36.软件测试项目从什么时候开始，？为什么？

37.需求测试注意事项有哪些？

38.简述一下缺陷的生命周期

39.测试分析测试用例注意（事项）？

你在你所在的公司是怎么开展测试工作的？是如何组织的？

你认为理想的测试流程是什么样子？

你是怎样工作的？

软件测试活动的生命周期是什么？

请画出软件测试活动的流程图？

针对缺陷采取怎样管理措施？

什么是测试评估？测试评估的范围是什么？

如果能够执行完美的黑盒测试，还需要进行白盒测试吗？为什么？

测试结束的标准是什么？

软件验收测试除了 alpha,beta 测试以外,还有哪一种?

做测试多久了？

以前做过哪些项目？

你们以前测试的流程是怎样的？

<答：测试计划—测试用例设计—测试执行—测试分析报告>

用过哪些测试工具？

为什么选择测试这行？

<答：它是一个新兴的行业，有发展潜力，而且很锻炼人，需要掌握更多的技能，比做

开发要更难>

为什么值得他们公司雇用？

如果我雇用你，你能给部门带来什么贡献？

如何从工作中看出你是个自动自觉的人

你的工作通常能在时限内完成吗。（我想问一下就是她问这个问题的动机是什么）

通常你对于别人批评你会有什么样的反应

如果明知这样做不对，你还会依主管的指过去做吗

如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理

你觉得什么样的人最难相处

为什么值得他们公司雇用？

帮助公司提高软件质量和测试部门的技术水平

如果我雇用你，你能给部门带来什么贡献？

分享我的测试经验和测试技能，提高测试部门技术水平

如何从工作中看出你是个自动自觉的人

自动自觉范围太广

1.

工作成果

2.

工作质量

你的工作通常能在时限内完成吗。（我想问一下就是她问这个问题的动机是什么）

在有足够的资源和合理的工作量的情况下，完全可以按时完成，并能比一般人做的更好

通常你对于别人批评你会有什么样的反应

有错即改，无措勉之

如果明知这样做不对，你还会依主管的指过去做吗

在公司内部下级是否有申诉渠道？

如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理

为什么抱怨？是怎么样问题？

如果是客服问题，提交客服部门解决

如果是质量问题，分析原因，下一版本改进

你觉得什么样的人最难相处

自以为是的人

什么叫单元测试？

请就软件测试人员应该具备什么样的基本素质说说你的看法。

请就如何在开发中进行软件质量控制说说你的看法

简述软件测试的意义，以及软件测试的分类

1、功能测试，性能测试，界面测试，安全测试（可以简单点，比如只涉及到 COOKIES 里的

内容），压力测试（商业性质的网站）等等，B/S 软件也要根据其具体功能采用不同的测试

策略。

2、态度、责任心、自信、敏锐的观察力、良好的发散思维

3、先设计后开发模式，加强单元测试，加强代码走查，有一套完整的白盒测试方法。关键



是加强开发人员的质量意识，增进程序员向工程师水平发展。

4、意义嘛，就自己想吧。软件测试的分类，这个很多人都按各种方法去分。无明确答案给你。

对测试的理解——基本的测试知识，对测试是否认可？ 75。

3、谈一谈过去自己的工作——了解经历、提供进一步提问的素材，表达能力

测试技能

测试设计的方法并举例说明——测试技术的使用

测试工具——熟悉程度，能否与当前工作匹配？

如何做计划？如何跟踪计划？——日常工作能力

如果开发人员提供的版本不满足测试的条件，如何做？——与开发人员协作的能力

熟悉 unix 系统、[oracle](#) 数据库吗？——是否具备系统知识

做过开发吗？写过哪些代码？——开发技能

阅读[英语](#)文章，给出理解说明？——部分英语能力 文档的意义——是否善于思考？（最简单的概念，不同层次的理解）

假如进入我们公司，对我们哪些方面会有帮助？——讲讲自己的特长

随便找一件物品，让其测试——测试的实际操作能力

软件测试的方法有？

软件测试的过程？

有一个新的软件，假如你是测试工程师，该如何做？

软件测试分哪两种方法？分别适合什么情况？

2。一套完整的测试应该由哪些阶段组成？分别阐述一下各个阶段。

3。软件测试的类型有那些？分别比较这些不同的测试类型的区别与联系。

4。测试用例通常包括那些内容？着重阐述编制测试用例的具体做法

5。在分别测试 winform 的 C/S 结构与测试 WEB 结构的软件是，应该采取什么样的方法分别测试？他们存在什么样的区别与联系？

6。在测试 winform 的 C/S 结构软件时，发现这个软件的运行速度很慢，您会认为是什么原因？

您会采取哪些方法去检查这个原因？

7。描述使用 bugzilla 缺陷管理工具对软件缺陷（BUG）跟踪的管理的流程

你在五年内的个人目标和职业目标分别是什么？

分析这个问题是用来了解你的计划能力的，通过这个问题，[面试](#)人同时还可以知道你的目标是否符合企业对你的安排。

错误回答我想在将来的某个时候考虑这个问题。如今企业的领导者更换频繁，我认为做

太多的个人计划是荒谬可笑的，不是吗？

评论这种回答属于令人反感的一类。首先，当有人想了解你的目标时，"将来的某个时候"这种通俗说法并不奏效。其次，认为企业很脆弱，领导者更换频繁，这种说法毫无疑问会令人反感，而且也是不合理的。最后，认为做计划可笑，看不起这个问题，而且反问面试官，这些都注定了这样的求职者最终会失败。

正确回答从现在起的五年之内，我希望能够在在一个很好的职位上待几年，而且最好有一次晋升，然后就期待着下一步。不管是向上提升，还是在企业内横向调动，对我个人来说，我希望找到一家企业——一家愿意做相互投入的企业——待上一段时间。

评论这个问题没有回答得过分具体（那样可能会产生漏洞），而且它表明你有雄心，并

且思考过在企业中的成长方式。通过表达横向调动和向上提升的愿望，表明你是一个有灵活性的人。

问题

23 你怎样做出自己的职业选择？

分析

面试官提出这个问题是为了了解求职者的动机，看看他（她）应聘这份工作是否有什么历史渊源，是否有职业规划，是不是仅仅在漫无目的地申请很多工作。

错误回答 我一直都想在企业界工作。自孩提时代起，我就梦想自己至少也要成为大企业的副总裁。

评论 除了难以令人相信之外，这种回答还存在一个问题：它表明求职者会对副总裁以下的职位不感兴趣。

正确回答 在上大学四年级前的那个夏天，我决定集中精力在某一领域谋求发展。尽管我是学商业的，但是我不知道自己最终会从事哪一行行业的工作。我花了一定的时间考虑自己的目标，想清楚了自己擅长做的事情以及想从工作中得到的东西，最后我得出了一个坚定的结论，那就是这个行业是最适合我的。

评论 这种回答表明，求职者认真地做过一些计划，缩小了自己的关注点，而且也认准了前进的方向。这种回答还表明，求职者理解个人职业规划的重要性，并且有能力做出认真的个人决策。

1. 你都用什么测试方法

2. 怎么编写案例

3. 怎么才能够全面的测试到每一个点

1. 你都用什么测试方法

针对不同的产品或者系统或者模块，有不同的测试方法。总体而言有白盒测试和黑盒测试。

2. 怎么编写案例

案例的编写与测试阶段的定义有很大的关系。系统测试和 unit 测试的案例可能不同。总体而言

言测试案例根据系统的需求而定。

3. 怎么才能够全面的测试到每一个点

测试的全面性主要需要在设计测试计划的时候考虑，从测试策略，产品需求等等多个角度考虑从而定义全部的测试点。

1、谈谈软件测试技术，以及如何提高

2、谈谈软件测试职业发展，以及个人的打算

3、谈谈软件测试在企业的地位，也可以结合软件生命周期来谈

有可能清晰的思路比确切的答案更重要

在这里，主要说下笔试和面试的问题，希望大家共同参考。

1，一般公司里实际的软件测试流程是什么样的？你们公司又是怎样的？

2，软件工程师要具有那些素质？

3，你会哪些测试工具？怎么操作？

4，你能不能说下你的 3 到 5 年的职业计划（规划）

5，你觉得你来应聘有那些优势？

其余的还好说，但就第 4 个问题，我感到不好说哦！希望大家给个意见

第一关：首先要自我介绍，自己的性格怎么样，目前的工作经历积累了一些什么经验取得了一些什么值得一说的成果。然后要说说对软件测试怎么看？还有对于软件测试有什么自己的想法。

法。为什么会想到要做这行（因为我的简历上的工作经历没有关于测试方面的）。哦，还有

期望薪资。

第二关：认为软件测试人员所要具备的基本素质，如果遇到问题会怎样处理，如果得不到研发人员的配合（就是研发说这个不是问题）你又会怎么处理？然后就是一些基本概念，比如软件测试的流程有哪些？如果我上任了，首先会怎么开始自己的工作计划。

（前两关通过了后面这个就好过多了）

第三关：像我介绍了一下公司的情况，告诉我主要针对什么内容的测试，会不会使用数据库。告诉我大概要做哪些内容，详细的可以上岗以后慢慢熟悉。

大概就这么多了，这对没有经过这一关的不知道有没有帮助，仅供参考吧

我觉得就像李波说的，关键是要给对方留下好印象:)

面试官最后会问你有什么问题要问吗。作为应聘者的你一般不要说没问题问，这会给面试官留下你不太重视这份工作的坏印象。所以如果你想得到这份工作的话应该抓住这最后的表现自己的机会：

你可以问：

1.

贵公司近期和远期的发展目标是什么？

2.

贵公司的主要竞争对手有哪些？

3.

贵公司有多少开发人员有多少测试人员？

4.

贵公司又进一步扩充测试人员的计划吗？

5.

如果我有幸能进入贵公司的话，我有怎么样的发展？

6.

测试人员的沟通能力很重要，贵公司有规范的沟通渠道吗？

7.

请介绍一下贵公司的福利情况。

8.

请问我什么时候能知道结果？

## 用友自动化测试面试题和部分答案

用友自动化测试面试提问：

1、英文测试用例和测试计划的编写（用英语回答）

2、自动化测试工具各自的用途（用英语回答）

3、QTP 实现功能测试的时候，当新版本的页面都改变了，应该如何解决？  
去更改对象仓库的属性和更改对象仓库

4、QTP 这个工具的优缺点？

QTP 工具的优点：可以实现数据批量录入，回归测试，数据初始化

缺点：对于页面变更太大，对象仓库的更新将会更大一些

5、测试计划所包含的内容？以及测试计划中的测试进度表示如何设计的？

优点：项目中的测试范围，质量定义，工期安排，资源安排

缺点：将测试范围和测试周期用表格形式显示出来



6、如何设计测试计划与测试用例？

根据需求，及项目的成本预算

7、整个项目共有多少测试用例，其中有多少用例使用了 QTP，都是什么类型的用例，使用

使用自动化测试工具的用例所占所有测试用例的比例

400 多个，都是功能方面的用例。50 多个使用自动化测试工具

8、对于你来说，你认为是技术更重要一些，还是业务更重要一些？（业务搞不好，技术再强也没有用）

9、编过程序吗？用什么语言？

10、测试数据是存放在脚本中，还是文档中？文档的形式有几种？

## 软件测试一些基本面试问题

### 1、黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分

是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不

考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因

此黑盒测试又叫**功能测试**或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

1) 是否有不正确或遗漏的功能？

2) 在接口上，输入是否能正确的接受？能否输出正确的结果？

3) 是否有数据结构错误或外部信息（例如数据文件）访问错误？

4) 性能上是否能够满足要求？

5) 是否有初始化或终止性错误？

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测

试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点

检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。

白盒测试主

要是想对程序模块进行如下检查：

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”与取“假”的两种情况都能至少测一遍。

3) 在循环的边界和运行的界限内执行循环体。

4) 测试内部数据结构的有效性，等等。

单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同

时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元

组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，

许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模

块与**其他**组的模块一起测试。最后，将构成进程的所有模块一起测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中

指定功能的有效方法。（常见的联调测试）

系统测试的目的是对最终软件系统全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将

其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装

成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任

务，即软件的功能和性能如同用户所合理期待的那样。

- 单元测试的主要目的是针对编码过程中可能存在的各种错误，例如用户输入验证过程中的边界值的错误。

- 集成测试主要目的是针对详细设计中可能存在的问题，尤其是检查各单元与**其它**程序部分之间的接口上可能存在的错误。

- 系统测试主要针对概要设计，检查了系统作为一个整体是否有效地得到运行，例如在产品设置中是否达到了

预期的高性能

- 验收测试通常由业务专家或用户进行，以确认产品能真正符合用户业务上的需要(需求)。

## 2、您认为做好测试计划工作的关键是什么？

### 1) 明确测试的目标，增强测试计划的实用性

编写**软件测试**计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能

需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

### 2) 坚持“5W”规则，明确内容与过程

“5W”规则指的是“**What**（做什么）”、“**Why**（为什么做）”、“**When**（何时做）”、“**Where**（在哪

里）”、“**How**（如何做）”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的

（**Why**），明确测试的范围和内容（**What**），确定测试的开始和结束日期（**When**），指出测试的方法和工具

（**How**），给出测试文档和软件的存放位置（**Where**）。

### 3) 采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内

容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

### 4) 分别创建测试计划与测试详细规格、测试用例

应把详细的**测试技术**指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例

放到独立创建的测试用例文档或测试用例管理**数据库**中。测试计划和测试详细规格、测试用例之间是战略和战术的

关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务

的具体战术。

## 3、你认为公司的 **BUG** 测试流程是什么？

1) 当测试工程师发现了一个 **bug** 而且在 **bug tracking tool** 里面没有相同的 **bug**，他需要填写所

有需要的 **bug** 信

息并且把这个 *bug* 分配给 *test leader*

2) 如果这个 *bug* 不是一个真正的 *bug*, *test leader* 需要 *close* 这个 *bug*

3) *test leader* 需要审查 *bug* 的各种信息都完备, 如果有信息不完整, 他需要把状态改

成“*feedback*”并重新 *assign* 给提交者

4) 如果这个

*bug* 是一个真正存在的 *bug*, *test leader* 需要把这个 *bug* 分配给相关的开发团队的

*PM*, 并且把

*bug*

状态改成 *Assigned*

5) 如果这个

*bug* 属于另外一个开发团队, *PM* 需要把这个 *bug* 重新分配给那个开发团队的 *PM*

6) *PM* 审查

*bug*, 并且分配给相应的开发人员去改正。

7) 开发人员收到 *bug* 以后, 对相关的缺陷进行改正, 并且重新分配给提交 *bug* 的测试人员并且

把状态改

成“*Fixed*”

8) 测试人员需要对这个 *bug* 进行重新测试, 保证相关的缺陷已经改正, 测试人员可以 *reopen* 这

个 *bug* 如果缺

陷依然存在并且重新分配给相关的开发人员或 *close* 这个 *bug* 如果缺陷已经改正。

# 软件测试面试题（软通动力、博彦科技、奇虎、瑞星、中软）

## 一、软通动力面试笔答

1.白箱测试和黑箱测试是什么?什么是回归测试?

2.单元测试、集成测试、系统测试的侧重点是什么?

单元测试的重点是系统的模块，包括子程序的正确性验证等。

集成测试的重点是模块间的衔接以及参数的传递等。

系统测试的重点是整个系统的运行以及与其他软件的兼容性。

3.设计用例的方法、依据有那些?

白盒测试用例设计有如下方法：基本路径测试\等价类划分\边界值分析\覆盖测试\循环测试\数据流

测试\程序插桩测试\变异测试.这时候依据就是详细设计说明书及其代码结构吧；

黑盒测试用例设计方法：基于用户需求的测试\功能图分析方法\等价类划分方法\边界值分析方法\

错误推测方法\因

果图方法\判定表驱动分析方法\正交实验设计方法.依据是用户需求规格说明书，详细设计说明书

4.一个测试工程师应具备那些素质和技能?

掌握基本的测试基础理论

本着找出软件存在的问题的态度进行测试,即客观吧,不要以挑刺形象出现

可熟练阅读需求规格说明书等文档

以用户的观点看待问题

有着强烈的质量意识

细心和责任心

良好的有效的沟通方式（与开发人员及客户）

具有以往的测试经验

能够及时准确地判断出高风险区在何处.

5.集成测试通常都有那些策略?

大爆炸集成；自顶向下集成；自底向上集成；三明治集成；分层集成；基干集成；基于功能的集成；

基于消息

的集成；基于风险的集成；基于进度的集成。

6.你用过的测试工具的主要功能、性能及其他？

7.一个缺陷测试报告的组成？

缺陷跟踪报告：

编号，如：*ut-dt00016*

标题，如：文字排版功能.字间距.*MarchCalculator* 计算错误

版本号，如：*V1.3*

执行状态，如：空白/草稿/提交/审批/分发/正在修改/修改完毕/正在确认/关闭...

修改记录，如：*2003* 年 *7* 月 *2* 日；肖睿编制/修改；原因

测试环境和版本号码、程序编写人员

错误严重程度和优先级别

错误详细描述

重现步骤和方式、对应的测试记录编码

## 阿里巴巴和华为笔试和面试题

阿里巴巴旗下各公司测试笔试和面试如下：

支付宝：

●

面试题：

●

讲述一下自己最熟悉的一个项目是怎么做的？具体用什么方法和测试工具？

●

简述一下整个项目的测试流程和 BUG 管理流程？

● BUG

描述中都包括那些内容？

●

讲述一个

BUG 的生命周期？

●

讲述自己在项目中发现最有意义的一个 BUG，是什么导致出现这个问题。

● LAODRUNERR

**B2B:**

●

笔试题:

●

根据下图，写出所有的备选流和输出流各个路径

●

给出需求设计测试用例:

●

给出一段需求文档画出流程图，并设计出测试用例，例子是留言板功能，从用户登陆到发送消息这样一些流程。

● BUG 描述中都包括哪些内容

●

数据库语句查询，主要包括三张表进行关联查询，表 1：学生表，表 2：班级 淘宝网

●

笔试题:

●

计算机网络和硬件知识。

●

计算机相关的文件存储和备份等知识。

●

数据库 SQL 查询

●

设计测试用例

●

逻辑思维题

**阿里软件**

面试试题:

谈一下项目整体架构。

●

自己在项目中处于角色，工作流程。

●

如何保证测试的整体覆盖率。

●

回归测试都要考虑哪些因素，如何设计相关测试用例。

●

编写一段程序，实现 0-100 之间递归。

●

数据库设计，链接查询。

**华为科技面试题（华为浙江总公司，华为杭州分公司）**

●

简单介绍一下项目结构，模块功能，主业务流程，并且用笔将项目内容描述出来。



谈一下对测试工具的认识，自己在项目中比较常用的工具是那种，具体怎么操作。

●

如何进行 BUG 分配和管理。

●

自己对编程语言的了解程度。

●

对通信电子信息的了解程度。

以上都是自己面试后，总结的一些内容，大家可以参考一下。

公众号：程序员一凡

## 哔哩哔哩软测工程师面试题

软件测试面试题整理

01. 为什么要在一个团队中开展软件测试工作？

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

02. 您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

我曾经做过 web 测试，后台测试，客户端软件，其中包括功能测试，性能测试，用户体验测试。最擅长的是功能测试

03. 您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同 04. 的测试类型的区别与联系（如功能测试、性能测试……）

测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。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。采用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。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范（快捷键之类的），

是否美观（能否吸引用户的注意力），是否安全（尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告）？做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的，然后再考虑该功能点的性能测试

04. 您认为做好测试用例设计工作的关键是什么？

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

05. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

- 1、是否有不正确或遗漏的功能？
- 2、在接口上，输入是否能正确的接受？能否输出正确的结果？
- 3、是否有数据结构错误或外部信息（例如数据文件）访问错误？
- 4、性能上是否能够满足要求？
- 5、是否有初始化或终止性错误？

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行如下检查：

- 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”与取“假”的两种情况都能至少测一遍。
- 3、在循环的边界和运行的界限内执行循环体。
- 4、测试内部数据结构的有效性，等等。

单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：

两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能

提供系统方案说明书中指定功能的有效方法。（常见的联调测试）

系统测试的目的是对最终软件系统全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

06. 测试计划工作的目的是什么？测试计划工作的内容都包括什么？其中哪些是最重要的？

软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。所以其中最重要的是测试策略和测试方法（最好是能先评审）

07. 您认为做好测试计划工作的关键是什么？

1. 明确测试的目标，增强测试计划的实用性

编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

2. 坚持“5W”规则，明确内容与过程

“5W”规则指的是“**What**（做什么）”、“**Why**（为什么做）”、“**When**（何时做）”、“**Where**（在哪里）”、“**How**（如何做）”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解

解测试的目的（**Why**），明确测试的范围和内容（**What**），确定测试的开始和结束日期（**When**），指出测试的方法和工具（**How**），给出测试文档和软件的存放位置（**Where**）。

3. 采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

4. 分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

08. 您所熟悉的测试用例设计方法都有哪些？请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

1. 等价类划分

划分等价类：等价类是指某个输入域的子集合。在该子集中，各个输入数据对于揭露程序中的错误都是等效的，并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试。因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，就可以用少量代表性的测试数据，取得较好的测试结果。等价类划分可有两种不同的情况：有效等价类和无效等价类。

## 2. 边界值分析法

边界值分析方法是等价类划分方法的补充。测试工作经验告诉我，大量的错误是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计

测试用例，可以查出更多的错误。

使用边界值分析方法设计测试用例，首先应确定边界情况。通常输入和输出等价类的边界，就是应着重测试的边界情况。应当选取正好等于，刚刚大于或刚刚小于边界的值作为测试数据，

而不是选取等价类中的典型值或任意值作为测试数据。

## 3. 错误推测法

基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法。

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。

## 4. 因果图方法

前面介绍的等价类划分方法和边界值分析方法，都是着重考虑输入条件，但未考虑输入条件之间的联系，相互组合等。考虑输入条件之间的相互组合，可能会产生一些新的情况。但要检查输入条件的组合不是一件容易的事情，即使把所有输入条件划分成等价类，他们之间的组合情况也相当多。因此必须考虑采用一种适合于描述对于多种条件的组合，相应产生多个动作的形式来考虑设计测试用例。这就需要利用因果图（逻辑模型）。因果图方法最终生成的就是判定表。它适合于检查程序输入条件的各种组合情况。

09. 请以您以往的实际工作为例，10. 详细的描述一次测试用例设计的完整的过程。

就说最近的这次网站功能的测试吧

首先：得到相关文档（需求文档和设计文档），理解需求和设计思想后，想好测试策略（测试计划简单点就 OK 了），考虑到测试环境，测试用例，测试时间等问题。

第二步：设计测试用例，测试策略是：把网站部分的功能点测试完，然后在进行系统测试（另外个模块呢有另一个测试人员负责，可以进行联调测试），网站模块的测试基本是功能测试和界面测试（用户并发的可能性很小，所以不考虑）：这次的网站的输入数据呢是使用数据库中的某张表记录，如果表中某一数据记录中新加进来的（还没有被处理的，有个标志位），网站启动后会立刻去刷那张表，得到多条数据，然后在进行处理。处理过程中，会经历 3 个步骤，网站才算完成了它的任务。有 3 个步骤呢，就可以分别对这 3 个步骤进

行测试用例的设计,尽量覆盖到各种输入情况（包括数据库中的数据，用户的输入等），得出

了差不多 50 个用例。界面测试，也就是用户看的到的地方，包括发送的邮件和用户填写资料的页面展示。

第三步：搭建测试环境（为什么这个时候考虑测试环境呢？因为我对网站环境已经很熟了，只有有机器能空下来做该功能测试就可以做了），因为网站本身的环境搭建和其他的系统有点不同，它需要的测试环境比较麻烦，需要 web 服务器（Apache,tomcat），不过这次

需求呢，网站部分只用到了 tomcat，所以只要有 tomcat 即可

第四步：执行测试

11. 您以往是否曾经从事过性能测试工作？如果有，12. 请尽可能的详细描述您以往的性能测试工作的完整过程。

是的，曾经做过网站方面的性能测试，虽然做的时间并不久（2 个月吧），当时呢，是有位网站性能测试经验非常丰富的前辈带着我一起做。

性能测试类型包括负载测试，强度测试，容量测试等

负载测试：负载测试是一种性能测试指数在超负荷环境中运行，程序是否能够承担。

强度测试：强度测试是一种性能测试，他在系统资源特别低的情况下软件系统运行情况

容量测试：确定系统可处理同时在线的最大用户数

在网站流量逐渐加大的情况下，开始考虑做性能测试了，首先要写好性能测试计划，根据运营数据得出流量最大的页面（如果是第一次的话，一般是首页，下载页，个人帐户页流量最大，而且以某种百分比），

Web 服务器指标指标：

\* Avg Rps: 平均每秒钟响应次数=总请求时间 / 秒数；

\* Successful Rounds: 成功的请求；

\* Failed Rounds : 失败的请求；

\* Successful Hits : 成功的点击次数；

\* Failed Hits : 失败的点击次数；

\* Hits Per Second : 每秒点击次数；

\* Successful Hits Per Second : 每秒成功的点击次数；

\* Failed Hits Per Second : 每秒失败的点击次数；

\* Attempted Connections : 尝试链接数；

13. 您在从事性能测试工作时，14. 是否使用过一些测试工具？如果有，15. 请试述该工具的工作原理，16. 并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。

17.

您认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？

18.

在您以往的工作中，19. 一条软件缺陷（或者叫 Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？

20. 您以往所从事的软件测试工作中，21. 是否使用了一些工具来进行软件缺陷（Bug）的管理？如果有，22. 请结合该工具描述软件缺陷（Bug）跟踪管理的流程。

23. 您认为在测试人员同

24. 开发人员的沟通过程中，25. 如何提高沟通的效率和改善沟通



的效果？维持测试人员同

26. 开发团队中其他成员良好的人际关系的关键是什么？

27. 在您以往的测试工作中，28.

最让您感到不 29. 满意或者不 30. 堪回首的事情是什么？

您是如何来对待这些事情的？

31. 在即将完成这次笔试前，32.

您是否愿意谈一些自己在以往的学习和工作中获得的工作

经验和心得体会？（可以包括软件测试、过程改进、软件开发或者与此无关的其他方面）

33. 你对测试最大的兴趣在哪里？为什么？

最大的兴趣就是测试有难度，有挑战性！做测试越久越能感觉到做好测试有多难。曾经在无忧测试网上看到一篇文章，是关于如何做好一名测试工程师。一共罗列了 11，12 点，有部分是和人的性格有关，有部分需要后天的努力。但除了性格有关的 1，2 点我没有把握，其他点我都很有信心做好它。

刚开始进入测试行业时，对测试的认识是从无忧测试网上了解到的一些资料，当时是冲着做测试需要很多技能才能做的好，虽然入门容易，但做好很难，比开发更难，虽然当时我很想做开发（学校专业课我基本上不缺席，因为我喜欢我的专业），但看到测试比开发更难

更有挑战性，想做好测试的意志就更坚定了。

不到一年半的测试工作中，当时的感动和热情没有减退一点（即使环境问题以及自身经验，技术的不足，做测试的你一定也能理解）。

我觉得做测试整个过程中有 2 点让我觉得很有难度（对我来说，有难度的东西我就非常感兴趣），第一是测试用例的设计，因为测试的精华就在测试用例的设计上了，要在版本出来之前，把用例写好，用什么测试方法写？（也就是测试计划或测试策略），如果你刚测试一个新任务时，你得花一定的时间去消化业务需求和技术基础，业务需求很好理解（多和产品经理和开发人员沟通就能达到目的），而技术基础可就没那么简单了，这需要你自觉的学习能力，比如说网站吧，最基本的技术知识你要知道网站内部是怎么运作的，后台是怎么响应用户请求的？测试环境如何搭建？这些都需要最早的学好。至少在开始测试之前能做好基本的准备，可能会遇到什么难题？需求细节是不是没有确定好？这些问题都能在设计用例的时候发现。

第二是发现 BUG 的时候了，这应该是测试人员最基本的任务了，一般按测试用例开始测试就能发现大部分的 bug，还有一部分 bug 需要测试的过程中更了解所测版本的情况获得

更多信息，补充测试用例，测试出

bug。还有如何发现 bug？这就需要在测试用例有效的情

况下，通过细心和耐心去发现 bug

了，每个用例都有可能发现 bug，每个地方都有可能出错，

所以测试过程中思维要清晰（测试过程数据流及结果都得看仔细了，bug 都在里面发现的）。如何描述 bug 也很有讲究，bug 在什么情况下会产生，如果条件变化一点点，就不会有这个

bug，以哪些最少的操作步骤就能重现这个 bug，这个 bug 产生的规律是什么？如果你够厉害的话，可以帮开发人员初步定位问题。

34. 你的测试职业发展是什么？

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级

测试工程师奔去。而且我也有初步的职业规划，前 3 年累积测试经验，按如何做好测试工程

师的 11, 12 点要求自己，不断的更新自己改正自己，做好测试任务。

35. 你自认为测试的优势在哪里？

优势在于我对测试坚定不移的信心和热情，虽然经验还不够，但测试需要的基本技能我有信心在工作中得以发挥。

36. 你以前工作时的测试流程是什么？

公司对测试流程没有规定如何做，但每个测试人员都有自己的一套测试流程。我说下我 1 年来不断改正（自己总结，吸取同行的方法）后的流程吧。需求评审（有开发人员，产品经理，测试人员，项目经理）—>需求确定(出一份确定的需求文档)—>开发设计文档（开发人员在开始写代码前就能输出设计文档）—>想好测试策略，写出测试用例—>发给开发人员和测试经理看看（非正式的评审用例）—>接到测试版本—>执行测试用例（中间可能会补充用例）—>提交 bug（有些 bug 需要开发人员的确定（严重级别的，或突然发现的在测试用例范围之外的，难以重现的），有些可以直接录制进 TD）—>开发人员修改（可以在测试过程中快速的修改）—>回归测试（可能又会发现新问题，再按流程开始跑）。

37. 当开发人员说不 38. 是 BUG 时，39. 你如何应付？

开发人员说不是 bug，有 2 种情况，一是需求没有确定，所以我可以这么做，这个时候可以找来产品经理进行确认，需不需要改动，3 方商量确定后再看要不要改。二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是 BUG 的依据是什么？

如果被用户发现或出了问题，会有什么不良结果？程序员可能会给你很多理由，你可以对他的解释进行反驳。如果还是不行，那我可以给这个问题提出来,跟开发经理和测试经理进行确认,如果要修改就改,如果不要修改就不改。其实有些真的不是

bug，我也只是建议的方式

写进 TD 中，如果开发人员不修改也没有大问题。如果确定是 bug 的话，一定要坚持自己的

立场，让问题得到最后的确认。

23. 你为什么想离开目前的职务？

因为公司运作情况并不理想，公司需要调整部门体系，公司考虑到缩减部门人员，所以大批量的裁员（有 6, 7 个），这是我的第一份工作，对公司也有较深的感情，因为在这里我

找到了职业理想（就是测试），所以公司需要精简人员，我自愿退出。虽然很舍不得，但我将会有新的发挥能力的舞台。

24: 你对我们公司了解有多少？

25: 你找工作时，最重要的考虑因素为何？

工作的性质和内容是否能让我发挥所长，并不断成长。

26: 为什么我们应该录取你？

您可以由我过去的工作表现所呈现的客观数据，明显地看出我全力以赴的工作态度。

27: 请谈谈你个人的最大特色。

我的坚持度很高，事情没有做到一个令人满意的结果，绝不罢手。



28.白箱测试和黑箱测试是什么?什么是回归测试?

29.单元测试、集成测试、系统测试的侧重点是什么?

30.设计用例的方法、依据有那些?

31.一个测试工程师应具备那些素质和技能?

32.集成测试通常都有那些策略?

33.你用过的测试工具的主要功能、性能及其他?

34.一个缺陷测试报告的组成

35.基于 WEB 信息管理系统测试时应考虑的因素有哪些?

36.软件测试项目从什么时候开始,? 为什么?

37.需求测试注意事项有哪些?

38.简述一下缺陷的生命周期

39.测试分析测试用例注意(事项)?

你在你所在的公司是怎么开展测试工作的?是如何组织的?

你认为理想的测试流程是什么样子?

你是怎样工作的?

软件测试活动的生命周期是什么?

请画出软件测试活动的流程图?

针对缺陷采取怎样管理措施?

什么是测试评估?测试评估的范围是什么?

如果能够执行完美的黑盒测试,还需要进行白盒测试吗?为什么?

测试结束的标准是什么?

软件验收测试除了 alpha,beta 测试以外,还有哪一种?

做测试多久了?

以前做过哪些项目?

你们以前测试的流程是怎样的?

<答:测试计划—测试用例设计—测试执行—测试分析报告>

用过哪些测试工具?

为什么选择测试这行?

<答:它是一个新兴的行业,有发展潜力,而且很锻炼人,需要掌握更多的技能,比做开发要更难>

为什么值得他们公司雇用?

如果我雇用你,你能给部门带来什么贡献?

如何从工作中看出你是个自动自觉的人

你的工作通常能在时限内完成吗.(我想问一下就是她问这个问题的动机是什么)

通常你对于别人批评你会有什么样的反应

如果明知这样做不对,你还会依主管的指过去做吗

如果你接到一个客户抱怨的电话,你确知无法解决他的问题,你会怎么处理

你觉得什么样的人最难相处

为什么值得他们公司雇用?

帮助公司提高软件质量和测试部门的技术水平

如果我雇用你,你能给部门带来什么贡献?

分享我的测试经验和测试技能,提高测试部门技术水平

如何从工作中看出你是个自动自觉的人

自动自觉范围太广

1.

工作成果

2.

工作质量

你的工作通常能在时限内完成吗。（我想问一下就是她问这个问题的动机是什么）

在有足够的资源和合理的工作量的情况下，完全可以按时完成，并能比一般人做的更好

通常你对于别人批评你会有什么样的反应

有错即改，无措勉之

如果明知这样做不对，你还会依主管的指过去做吗

在公司内部下级是否有申诉渠道？

如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理

为什么抱怨？是怎么样问题？

如果是客服问题，提交客服部门解决

如果是质量问题，分析原因，下一版本改进

你觉得什么样的人最难相处

自以为是的人

什么叫单元测试？

请就软件测试人员应该具备什么样的基本素质说说你的看法。

请就如何在开发中进行软件质量控制说说你的看法

简述软件测试的意义，以及软件测试的分类

1、功能测试，性能测试，界面测试，安全测试（可以简单点，比如只涉及到 COOKIES 里

的内容），压力测试（商业性质的网站）等等，B/S 软件也要根据其具体功能采用不同的测

试策略。

2、态度、责任心、自信、敏锐的观察力、良好的发散思维

3、先设计后开发模式，加强单元测试，加强代码走查，有一套完整的白盒测试方法。关键是加强开发人员的质量意识，增进程序员向工程师水平发展。

4、意义嘛，就自己想吧。软件测试的分类，这个很多人都按各种方法去分。无明确答案给你。

对测试的理解——基本的测试知识，对测试是否认可？ 75。

3、谈一谈过去自己的工作——了解经历、提供进一步提问的素材，表达能力

测试技能

测试设计的方法并举例说明——测试技术的使用

测试工具——熟悉程度，能否与当前工作匹配？

如何做计划？如何跟踪计划？——日常工作能力

如果开发人员提供的版本不满足测试的条件，如何做？——与开发人员协作的能力

熟悉 unix 系统、oracle 数据库吗？——是否具备系统知识

做过开发吗？写过哪些代码？——开发技能

阅读英语文章，给出理解说明？——部分英语能力

文档的意义——是否善于思考？（最简单的概念，不同层次的理解）

假如进入我们公司，对我们哪些方面会有帮助？——讲讲自己的特长

随便找一件物品，让其测试——测试的实际操作能力

软件测试的方法有？

软件测试的过程？

有一个新的软件，假如你是测试工程师，该如何做？

软件测试分哪两种方法？分别适合什么情况？

2. 一套完整的测试应该由哪些阶段组成？分别阐述一下各个阶段。

3. 软件测试的类型有那些？分别比较这些不同的测试类型的区别与联系。

4. 测试用例通常包括那些内容？着重阐述编制测试用例的具体做法

5. 在分别测试 winform 的 C/S 结构与测试 WEB 结构的软件是，应该采取什么样的方法分

别测试？他们存在什么样的区别与联系？

6. 在测试 winform 的 C/S 结构软件时，发现这个软件的运行速度很慢，您会认为是什么原因

因？您会采取哪些方法去检查这个原因？

7. 描述使用 bugzilla 缺陷管理工具对软件缺陷（BUG）跟踪的管理的流程

你在五年内的个人目标和职业目标分别是什么？

分析这个问题是用来了解你的计划能力的，通过这个问题，面试官同时还可以知道你的目标是否符合企业对你的安排。

错误回答我想在将来的某个时候考虑这个问题。如今企业的领导者更换频繁，我认为做太多的个人计划是荒谬可笑的，不是吗？

评论这种回答属于令人反感的一类。首先，当有人想了解你的目标时，"将来的某个时候"这种通俗说法并不奏效。其次，认为企业很脆弱，领导者更换频繁，这种说法毫无疑问会令人反感，而且也是不合理的。最后，认为做计划可笑，看不起这个问题，而且反问面试官，这些都注定了这样的求职者最终会失败。

正确回答从现在起的五年之内，我希望能够在很好的职位上待几年，而且最好有一次晋升，然后就期待着下一步。不管是向上提升，还是在企业内横向调动，对我个人来说，我希望找到一家企业——一家愿意做相互投入的企业——待上一段时间。

评论这个问题没有回答得过分具体（那样可能会产生漏洞），而且它表明你有雄心，并且思考过在企业中的成长方式。通过表达横向调动和向上提升的愿望，表明你是一个有灵活性的人。

问题

23 你怎样做出自己的职业选择？

分析

面试官提出这个问题是为了了解求职者的动机，看看他（她）应聘这份工作是否有什么历史渊源，是否有职业规划，是不是仅仅在漫无目的地申请很多工作。

错误回答 我一直都想在企业界工作。自孩提时代起，我就梦想自己至少也要成为大企业的副总裁。

评论 除了难以令人相信之外，这种回答还存在一个问题：它表明求职者会对副总裁以下的职位不感兴趣。

正确回答 在上大学四年级前的那个夏天，我决定集中精力在某一领域谋求发展。尽管我是学商业的，但是我不知道自己最终会从事哪一行行业的工作。我花了一定的时间考虑自己的目标，想清楚了自己擅长做的事情以及想从工作中得到的东西，最后我得出了一个坚定的

结论，那就是这个行业是最适合我的。

评论 这种回答表明，求职者认真地做过一些计划，缩小了自己的关注点，而且也认准了前进的方向。这种回答还表明，求职者理解个人职业规划的重要性，并且有能力做出认真的个人决策。

1. 你都用什么测试方法
2. 怎么编写案例
3. 怎么才能够全面的测试到每一个点

1. 你都用什么测试方法

针对不同的产品或者系统或者模块，有不同的测试方法。总体而言有白盒测试和黑盒测试。

2. 怎么编写案例

案例的编写与测试阶段的定义有很大的关系。系统测试和 unit 测试的案例可能不同。总体而言测试案例根据系统的需求而定。

3. 怎么才能够全面的测试到每一个点

测试的全面性主要需要在设计测试计划的时候考虑，从测试策略，产品需求等等多个角度考虑从而定义全部的测试点。

- 1、谈谈软件测试技术，以及如何提高
- 2、谈谈软件测试职业发展，以及个人的打算
- 3、谈谈软件测试在企业的地位，也可以结合软件生命周期来谈

有可能清晰的思路比确切的答案更重要

在这里，主要说下笔试和面试的问题，希望大家共同参考。

- 1，一般公司里实际的软件测试流程是什么样的？你们公司又是怎样的？
- 2，软件工程师要具有那些素质？
- 3，你会哪些测试工具？怎么操作？
- 4，你能不能说下你的 3 到 5 年的职业计划（规划）
- 5，你觉得你来应聘有那些优势？

其余的还好说，但就第 4 个问题，我感到不好说哦！希望大家给个意见

第一关：首先要自我介绍，自己的性格怎么样，目前的工作经历积累了一些什么经验取得了一些什么值得一说的成果。然后要说说对软件测试怎么看？还有对于软件测试有什么自己的想法。为什么会想到要做这行（因为我的简历上的工作经历没有关于测试方面的）。哦，还有期望薪资。

第二关：认为软件测试人员所要具备的基本素质，如果遇到问题会怎样处理，如果得不到研发人员的配合（就是研发说这个不是问题）你又会怎么处理？然后就是一些基本概念，比如软件测试的流程有哪些？如果我上任了，首先会怎么开始自己的工作计划。

（前两关通过了后面这个就好过多了）

第三关：像我介绍了一下公司的情况，告诉我主要针对什么内容的测试，会不会使用数据库。告诉我大概要做哪些内容，详细的可以上岗以后慢慢熟悉。

大概就这么多了，这对没有经过这一关的不知道有没有帮助，仅供参考吧

我觉得就像李波说的，关键是要给对方留下好印象:)

面试官最后会问你有什么问题要问吗。作为应聘者的你一般不要说没问题问，这会给面试官留下你不太重视这份工作的坏印象。所以如果你想得到这份工作的话应该抓住这最后的表现自己的机会：

你可以问：

- 1.

贵公司近期和远期的发展目标是什么？

2.

贵公司的主要竞争对手有哪些？

3. 贵公司有多少开发人员有多少测试人员？

4.

贵公司又进一步扩充测试人员的计划吗？

5.

如果我有幸能进入贵公司的话，我有怎么样的发展？

6.

测试人员的沟通能力很重要，贵公司有规范的沟通渠道吗？

7.

请介绍一下贵公司的福利情况。

8.

请问我什么时候能知道结果？

用友面试：

1. 内联接和外联接，自联接有什么区别？

内联接通常是

2

个表存在主外键关系时使用的，

内联接查询有

2

种方式实现，

1

是在

WHERE

子句中指定联接条件

2

是在

FROM 子句中使用

join...on

内联接查询通常不仅仅联接

2 表，可以 3 表甚至更多的表

参与内联接的表的地位是平等的

而外联接中参与联接的表有主从之分。以主表的每行数据去匹配从表的数据列，符合条件的数据将直接返回到结果集中，不符合的用 NULL（空值）填充后再返回到结果集中。

2. SQL 中 \group by\和\order by\有什么不同呢

一个是对处理的数据进行分组，一个是对处理的数据进行排序

自动测试的好处：

如果你需要反复运行一组测试，那么自动测试将会对你非常有用。

自动测试使你能够应对频繁改变的代码从而跟上周期性回归测试的脚步。

自动测试可以使你能够自动运行主流业务场景从而跟上周期性回归测试的脚步。（原文：It gives you the ability to run automation in main stream scenarios to catch regressions in a timely manner）

自动测试可以帮助你测试大量测试矩阵（在不同操作系统上的不同语言）。自动测试可以使你的测试同时运行在不同的机器上，而手动测试必须不断地继续执行。

自动测试的限制：

花费大。编写测试用例，编写和配置自动化测试框架将会在测试开始时花费比手动测试更多的费用。

无法自动测试一些可视的场景。例如，如果你无法通过代码告诉自动测试工具字体颜色，那么只好使用手动测试。

手动测试的好处：

如果一个测试用例在编码阶段只运行两次，那最好使用手动测试，它将比自动测试花费少得多的费用。

手动测试允许测试员进行更多的随机测试。以我的经验来看，更多的 bug 将会由随机测试发

现，而不是自动测试。并且，一个测试员花费越多的时间进行随机测试，发现真正的用户 bug 的几率就越大。

手动测试的限制：

手动进行测试将花费大量的时间。

每次有了新的 build，测试员必须重新运行测试-经过一段时间以后将会非常繁琐和疲惫。

其他的因素：

你将哪些部分进行自动测试也由你使用的工具决定。如果该工具有很多限制，那么这些部分还是手动测试吧。

是否投资的回报值得运行自动测试？是否你自动化测试的产出值得建立和支持测试用例，自动框架和运行测试用例的系统？

自动测试的标准

有两个问题可以用来判断是否应该为你的测试用例进行自动化。

Q1:是否测试场景可以自动化？

A1:是的，并且花费很少。

A2:是的，但是花费很多。

A3:不，不可能进行自动化。

Q2:该测试场景有多么重要？

A1:我必须任何可能的时候都对其进行测试。

A2:我需要有规律地对该场景进行测试。

A3:我只需要测试该场景一次。

如果这两个问题你的答案都是#1，那么你肯定需要自动化该测试。

如果这两个问题你的答案是一个#1 和一个#2，那么你最好自动化该测试。

如果这两个问题你的答案都是#2，那么你应该好好考虑一下是否你值得为自动化测试投资。

如果你无法自动测试，会有什么结果

让我们假设如果你有一个测试必须在任何可能的时间运行，但是却无法自动化它，你的选择是：

再评估 - 是否我真的需要如此频繁地运行它？

如果手动测试它会有多大的花费？

寻找新的测试工具。

考虑使用 test hooks.

四款主流测试工具的测试流程



主流测试工具的测试流程

=====winrunner

1

启动时选择要加载的插件

2

进行一些设置（如录制模式等）

3

识别应用程序的 GUI，即创建 map(就是学习被测试软件的界面)

4

建立测试脚本（录制及编写）

5

对脚本除错及调试（保证能够运行完）

6

插入各种检查点（图片，文字，控件等）

7

在新版应用程序中执行测试脚本

8

分析结果，回报缺陷

=====quicktestpro=====

1 准备录制

打开你要对其进行测试的应用程序，并检查 QuickTest 中的各项设置是否适合当前的要求。

2 进行录制

打开 QuickTest 的录制功能，按测试用例中的描述，操作被测试应用程序。

3 编辑测试脚本

通过加入检测点、参数化测试，以及添加分支、循环等控制语句，来增强测试脚本的功能，使将来的回归测试真正能够自动化。

4 调试脚本

调试脚本，检查脚本是否存在错误。

5 在回归测试中运行测试

在对应用程序的回归测试中，通过 QuickTest 回放对应用程序的操作，检验软件正确性，实

现测试的自动化进行。

6 分析结果，报告问题

查看 QuickTest 记录的运行结果，记录问题，报告测试结果。

====TestDirect=====

安装好后，先进入站点管理

1 创建域及工程

2

添加用户

3

编辑 licenses 及本服务器

4

编辑数据库



——TD

1

选择新建的工程进行定制(列表, 用户, 组, 版本等)

2

在 require 中增加需求

3

把需求转化为 plan

4

在 testlab 中由计划新建测试具体用例与执行

5

发现 bug, 在 defect 中提交 bug

(每一部分都可以相对独立地使用)

=====loadrunner

1 制定负载测试计划

(分析应用程序, 确定测试目标, 计划怎样执行 LoadRunner)

2 开发测试脚本

(录制基本的用户脚本, 完善测试脚本)

3 创建运行场景

(选择场景类型为 Manual Scenario, 选择场景类型, 理解各种类型, 场景的类型转化)

4

运行测试

5

监视场景

(MEMORY 相关, PROCESSOR 相关, 网络吞吐量以及带宽, 磁盘相关, WEB 应用程序

,

IIS5.0, SQL SERVER, NETWORK DELAY 等)

6 分析测试结果

(分析实时监视图表,分析事务的响应时间,分解页面,确定 WEBSERVER 的问题,其他有用的功能)

软件测试面试题

2007-02-28 17:17

软件测试的目的?

测试的目的是想以最少的人力、物力和时间找出软件中潜在的各种错误和缺陷,通过修正种错误和缺陷提高软件质量,回避软件发布后由于潜在的软件缺陷和错误造成的隐患带来的商业风险。

**Beta 测试:** 在客户场地,由客户进行的对产品预发布版本的测试。

软件验收测试合格通过准则: 1 软件需求分析说明书中定义的所有功能已全部实现,性能指标全部达到要求。2 所有测试项没有残余的一级二级三级的错误。3 立项审批表、需求分析文档、设计文档和编码实现一致。4 验收测试工件齐全(测试计划,测试用例,测试日志,测试通知单,测试分析报告)

软件验收测试包括正式验收测试、alpha 测试、beta 测试三种测试。

系统测试的策略: 功能测试,性能测试,外部接口测试,界面测试,强度测试,冗余测试,可靠性测试,恢复测试等

设计系统测试计划需要参考的项目文档有软件测试计划、软件需求工件、和迭代计划。

利用因果图导出测试用例需要经过的一般步骤

- 1.分析程序规格说明的描述中，哪些是原因，哪些是结果。
- 2.分析程序规格说明的描述中语义的内容，并将其表示成连接各个原因与各个结果的因果图
- 3.在因果图上使用若干个特殊的符号标明特定的约束条件
- 4.把因果图转换成判定表
- 5.把判定表中每一列表示的情况写成测试用例

阶段评审与同行评审的区别

同行评审目的:发现小规模工作产品的错误,只要是找错误;

阶段评审目的:评审模块阶段作品的正确性可行性及完整性

同行评审人数:3-7 人人员必须经过同行评审会议的培训,由 SQA 指导

阶段评审人数:5 人左右评审人必须是专家具有系统评审资格

同行评审内容:内容小一般文档 <40 页,代码 <500 行

阶段评审内容: 内容多,主要看重点

同行评审时间:一小部分工作产品完成

阶段评审时间: 通常是设置在关键路径的时间点上!

什么是软件测试?

使用人工或自动手段来运行或测定某个系统的过程,其目的在于检验它是否满足规定的要求或是弄清预期结果与实际结果之间的差别。

软件测试就是在软件投入运行前,对软件需求分析、设计规格说明和编码的最终复审,是软件质量保证的关键步骤。软件测试是为了发现错误而执行程序的过程。

简述集成测试的过程

根据 IEEE 标准 集成测试划分为 4 个阶段:计划阶段,设计阶段,实现阶段,执行阶段 (实

施阶段)

计划阶段

1) 时间安排 概要设计完成评审后大约一个星期

2) 输入 需求规格说明书 概要设计文档 产品开发计划路标

3) 入口条件

概要设计文档已经通过评审

4) 活动步骤

1.定被测对象和测试范围 2.评估集成测试被测对象的数量及难度,

即工作量 3.确定角色分工和作任务 4.标识出测试各阶段的时间,任务,约束等条件 5.考虑一定的风险分析及应急计划 6.考虑和准备集成测试需要的测试工具,测试仪器,环境等资源

7.考虑外部技术支援的力度和深度,以及相关培训安排

8.定义测试完成标准

5) 输出 集成测试计划

6) 出口条件

集成测试计划通过概要设计阶段基线评审

设计阶段

1) 时间安排

详细设计阶段开始

2) 输入 需求规格说明书 概要设计 集成测试计划

### 3) 入口条件

概要设计基线通过评审

### 4) 活动步骤

1.被测对象结构分析 2.集成测试模块分析

3.集成测试接口分析 4.集成测试策

略分析

5.集成测试工具分析 6.集成测试环境分析 7.集成测试工作量估计和安排。

### 5) 输出 集成测试设计（方案）

6.出口条件 集成测试设计通过详细设计基线评审。

实现阶段

1) 时间安排 在编码阶段开始后进行

2) 输入 需求规格说明书 概要设计 集成测试计划 集成测试设计

3) 入口条件

详细设计阶段

4) 活动步骤

集成测试用例设计 集成测试程序设计 集成测试代码设计（如果需要）

集

成测试脚本（如果需要） 集成测试工具（如果需要）

5) 输出 集成测试用例 集成测试规程 集成测试代码 集成测试脚本 集成测试工具

6) 出口条件

测试用例和测试规程通过编码阶段基线评审

执行阶段

1) 时间安排

单元测试已经完成后就可以开始执行集成测试了

2) 输入 需求规格说明书 概要设计 集成测试计划 集成高度设计 集成测试例

集成测试规程 集成测试代码（如果有） 集成测试脚本 集成测试工具 详细设计 代

码 单元测试报告

3) 入口条件 单元测试阶段已经通过基线化评审

4) 活动步 骤 执行集成测试用例 回归集成测试用例 撰写集成测试报告

5) 输出 集成测试报告

6) 出口条件 集成测试报告通过集成测试阶段基线评审

文档测试？

文档审核测试目前越来越引起人们的重视，软件质量不是检查出来的，而是融进软件开发中来。文档审核测试主要包括需求文档测试，设计文档测试，为前置软件测试中的一部分。

需求文档测试：主要测试需求中是否存在逻辑矛盾以及需求在技术上是否可以实现；

设计文档测试：测试设计是否符合全部需求以及设计是否合理。

白盒测试有哪几种方法？

白盒测试也称结构测试或逻辑驱动测试，它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。“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。

1. 软通动力面试笔答

### 1.白箱测试和黑箱测试是什么?什么是回归测试?

白箱测试是在看懂程序代码和设计方案的前提下，进行软件的测试。这种测试注重于源代码的覆盖率，同时需要测试者具备较高的技术水平。白箱测试的优点是可以对代码有详细的审查，能找出隐藏在代码中的错误，从而确保高质量的代码；缺点是很多时候不能看完所有的代码，不能找出欠缺的代码，同时白箱测试和用户如何使用软件无关。

黑箱测试的优点是测试者无需熟悉软件内部结构，并且根据蓝图在早期就可以制定测试方案，并不依赖于开发者的工作进展，而且黑箱测试简单易行，对测试者的技术要求不高；但是，黑箱测试主要是功能上的测试，只能覆盖只有一小部分的输入，不能保证程序的所有部分都被测试到。

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。自动回归测试将大幅降低系统测试、维护升级等阶段的成本。

回归测试包括两部分：函数本身的测试、其他代码的测试。

在对被修改的函数重新测试。如果函数的设计功能没有变化，直接运行函数测试就可以了。如果修改了设计功能，则要根据增减的功能点，增加或删除测试用例。另外，还要完成白盒覆盖。

函数代码的修改可能导致调用该函数的代码产生错误，所以需要测试其他代码。如果函数是私有函数并且未涉及到全局变量，应运行类测试，否则应运行工程测试。在函数列表中选择类测试或工程测试，编译运行测试工程，即可执行对其他代码的回归测试。

### 2.单元测试、集成测试、系统测试的侧重点是什么?

单元测试：以代码检查、逻辑覆盖

集成测试：增加静态结构分析、静态质量度量

系统测试：根据黑盒测试结果，采用白盒测试

单元测试是在软件开发过程中要进行的最低级别的测试活动，在单元测试活动中，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

集成测试，也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求，组装成为子系统或系统，进行集成测试。实践表明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。程序在某些局部反映不出来的问题，在全局上很可能暴露出来，影响功能的实现。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。

### 3.设计用例的方法、依据有那些?

白盒测试用例设计有如下方法:基本路径测试\等价类划分\边界值分析\覆盖测试\循环测试\数据流测试\程序插桩测试\变异测试.这时候依据就是详细设计说明书及其代码结构吧,恩,这个真不确定

黑盒测试用例设计方法:基于用户需求的测试\功能图分析方法\等价类划分方法\边界值分析方法\错误推测方法\因果图方法\判定表驱动分析方法\正交实验设计方法.依据是用户需求规格说明书,详细设计说明书

### 4.一个测试工程师应具备那些素质和技能?

掌握基本的测试基础理论

本着找出软件存在的问题的态度进行测试,即客观吧,不要以挑刺形象出现

可熟练阅读需求规格说明书等文档

以用户的观点看待问题

有着强烈的质量意识

细心和责任心

良好的有效的沟通方式(与开发人员及客户)

具有以往的测试经验

能够及时准确地判断出高危险区在何处

#### ①、沟通能力

一名理想的测试者必须能够同测试涉及到的所有人进行沟通，具有与技术（开发者）和非技术人员（客户，管理人员）的交流能力。既要可以和用户谈得来，又能同开发人员说得上话，不幸的是这两类人没有共同语言。和用户谈话的重点必须放在系统可以正确地处理什么和不可以处理什么上。而和开发者谈相同的信息时，就必须将这些活重新组织以另一种方式表达出来，测试小组的成员必须能够同等地同用户和开发者沟通。

#### ②、移情能力

和系统开发有关的所有人员都处在一种既关心又担心的状态之中。用户担心将来使用一个不符合自己要求的系统，开发者则担心由于系统要求不正确而使他不重新开发整个系统，管理部门则担心这个系统突然崩溃而使它的声誉受损。测试者必须和每一类人打交道，因此需要测试小组的成员对他们每个人都具有足够的理解和同情，具备了这种能力可以将测试人员与相关人员之间的冲突和对抗减少到最低程度。

#### ③、技术能力

就总体言，开发人员对那些不懂技术的人持一种轻视的态度。一旦测试小组的某个成员作出了一个错误的断定，那么他们的可信度就会立刻被传扬了出去。一个测试者必须既明白被测软件系统的概念又要会使用工程中的那些工具。要做到这一点需要有几年以上的编程经验，前期的开发经验可以帮助对软件开发过程有较深入的理解，从开发人员的角度正确的评价测试者，简化自动测试工具编程的学习曲线。

#### ④、自信心

开发者指责测试者出了错是常有的事，测试者必须对自己的观点有足够的自信心。如果容许别人对自己指东指西，就不能完成什么更多的事情了。

#### ⑤、外交能力

当你告诉某人他出了错时，就必须使用一些外交方法。机智老练和外交手法有助于维护与开发人员的协作关系，测试者在告诉开发者他的软件有错误时，也同样需要一定的外交手腕。如果采取的方法过于强硬，对测试者来说，在以后和开发部门的合作方面就相当于“赢了战争却输了战役”。

#### ⑥、幽默感

在遇到狡辩的情况下，一个幽默的批评将是很有帮助的。

#### ⑦、很强的记忆力

一个理想的测试者应该有能力将以前曾经遇到过的类似的错误从记忆深处挖掘出来，这一能力在测试过程中的价值是无法衡量的。因为许多新出现的问题和我们已经发现的问题相差无几。

#### ⑧、耐心

一些质量保证工作需要难以置信的耐心。有时你需要花费惊人的时间去分离、识别和分派一个错误。这个工作是那些坐不住的人无法完成的。

#### ⑨、怀疑精神

可以预料，开发者会尽他们最大的努力将所有的错误解释过去。测试者必须听每个人的说明，但他必须保持怀疑直到他自己看过以后。

#### ⑩、自我督促



干测试工作很容易使你变得懒散。只有那些具有自我督促能力的人才能够使自己每天正常地工作。

## 11、洞察力

一个好的测试工程师具有“测试是为了破坏”的观点，捕获用户观点的能力，强烈的质量追求，对细节的关注能力。应用的高风险区的判断能力以便将有限的测试针对重点环节。

### 5.集成测试通常都有那些策略？

- 1、 在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
- 2、 各个子功能组合起来，能否达到预期要求的父功能；
- 3、 一个模块的功能是否会对另一个模块的功能产生不利的影响；
- 4、 全局数据结构是否有问题；
- 5、 单个模块的误差积累起来，是否会放大，从而达到不可接受的程度。
- 6.你用过的测试工具的主要功能、性能及其他？

WinRunner (WR) 是一个基于 Windows 的企业级功能测试工具，它在业务应用正式部署之前，通过自动捕获、检测和重放用户对应用系统的交互操作，来发现系统缺陷，确保那些跨越多个应用程序和数据库的业务流程在初次发布就能避免故障的出现，保证系统对所有关键业务处理功能、处理流程的正确，保障应用的质量和准备工作的最优化

主要功能:

- 1) 轻松创建测试：用 WinRunner 创建一个测试，只需在应用软件中操作记录下一个标准的业务流程，例如下一张订单或建立一个新的商家账户，WinRunner 将直观地记录该流程。

即

使技术知识有限的用户，也能通过在 GUI 上单击鼠标而生成完整的测试。用户还可以直接编辑测试指令来满足各种复杂测试的需求

- 2) 插入检查点：在建立一个测试的过程中可以插入检查点，以在查找潜在错误的同时，将预想的结果和实际测试结果进行比较。在插入检查点后，WinRunner 会收集相应的性能指标，在测试运行时对其一一验证。WinRunner 允许使用几种不同类型的检查点，包括文本、GUI、位图 and 数据库等。例如用一个位图图检查点，可以确认一个位图图像是否出现在指定的位置上。

WinRunner 的数据库检验功能能够自动标示出被修改的数据

- 3) 检验数据：除了创建并运行测试，WinRunner 还能验证数据库的数值，从而确保交易的准确性。例如，在测试创建时，可以设定哪些数据库表格和记录资料需要检测。在重放时，测试程序就会核对数据库内的实际数值与预想的数值。WinRunner 能自动显示检测结果，在

有更新/修改、删除或插入的记录上会用突出标识引起注意

- 4) 增强测试：为了彻底全面地测试一个应用程序，用户需要了解对于不同类型的数据它是如何运行的。WinRunner 的 DataDriver Wizard 使用户只需单击几下鼠标，就能简单地将一个

记录下的业务流程转化为一个数据驱动的测试，来反映多个用户各自独特且真实的操作行为

- 5) 运行测试：在建立测试，并插入检查点和做一些必要的功能添加后，就可以开始运行测试。当 WinRunner 执行测试时，它会自动操作应用程序，正如一个真实用户根据记录流程执行着每一步的操作，而且它的意外处理功能为测试排除干扰，包括消息和警报

- 6) 分析结果：一旦测试运行后，就需要分析测试结果。WinRunner 的互动式的报告工具通过提供详尽的、易读的报告，其中会列出在测试中发现的差错和出错的位置，来帮助用户解释所得到的结果。这些报告对在测试运行中发生的重要事件进行描述，如出错内容和检查点等。单击按钮，还能进一步获取任何未被包括在此测试范围内的错误的详尽资料。这些结果

都可以通过 MI 的测试管理工具 TestDirector 来查阅

7) 维护测试：随着时间推移，开发人员会对应用程序做进一步的修改，这时，需要增加额外的测试。WinRunner 会帮助用户创建可重复使用的测试，以大大节省时间和资源，充分利

用测试投资

7.一个缺陷测试报告的组成

缺陷的标题，缺陷的基本信息，复现缺陷的操作步骤，缺陷的实际结果描述，期望的正确结果描述，注释文字和截取的缺陷图象。

缺陷的标题；

缺陷的基本信息；

测试的软件和硬件环境；

测试的软件版本；

缺陷的类型；

缺陷的严重程度；

缺陷的处理优先级。

复现缺陷的操作步骤；

缺陷的实际结果描述；

期望的正确结果描述；

注释文字和截取的缺陷图像。

8.基于 WEB 信息管理系统测试时应考虑的因素有哪些？

一、功能测试

1、链接测试

2、表单测试

3、Cookies 测试

4、设计语言测试

5、数据库测试

二、性能测试

1、连接速度测试

2、负载测试

3、压力测试

三、可用性测试

1、导航测试

2、图形测试

3、内容测试

4、整体界面测试

四、客户端兼容性测试

1、平台测试

2、浏览器测试

五、安全性测试

9.软件本地化测试比功能测试都有哪些方面需要注意？

软件本地化测试的目的：

软件本地化测试的测试策略：1.本地化软件要在各种本地化操作系统上安装并测试。2.源语言软件安装在另一台相同源语言操作系统上，作为对比测试。3.重点测试因本地化引起的软



件的功能和软件界面的错误。4.测试本地化软件的翻译质量。5.手工测试和自动测试相结合。

10.软件测试项目从什么时候开始，为什么？

软件测试应该在需求分析阶段就介入,因为测试的对象不仅仅是程序编码,应该对软件开发过程中产生的所有产品都测试,并且软件缺陷存在放大趋势.缺陷发现的越晚,修复它所花费的成本就越大.

11.需求测试注意事项有哪些？

一个良好的需求应当具有一下特点：

完整性：每一项需求都必须将所要实现的功能描述清楚，以使开发人员获得设计和实现这些功能所需的所有必要信息。

正确性：每一项需求都必须准确地陈述其要开发的功能。

一致性：一致性是指与其它软件需求或高层（系统，业务）需求不相矛盾。

可行性：每一项需求都必须是在已知系统和环境的权能和限制范围内可以实施的。

无二义性：对所有需求说明的读者都只能有一个明确统一的解释，由于自然语言极易导致二义性，所以尽量把每项需求用简洁明了的用户性的语言表达出来。

健壮性：需求的说明中是否对可能出现的异常进行了分析，并且对这些异常进行了容错处理。

必要性：“必要性”可以理解为每项需求都是用来授权你编写文档的“根源”。要使每项需求

都能回溯至某项客户的输入，如 Use Case 或别的来源。

可测试性：每项需求都能通过设计测试用例或其它的验证方法来进行测试。

可修改性：每项需求只应在 SRS 中出现一次。这样更改时易于保持一致性。另外，使用目

录表、索引和相互参照列表方法将使软件需求规格说明书更容易修改。

可跟踪性：应能在每项软件需求与它的根源和设计元素、源代码、测试用例之间建立起链接链，这种可跟踪性要求每项需求以一种结构化的，粒度好（fine-grained）的方式编写并单独标明，而不是大段大段的叙述。

12.简述一下缺陷的生命周期

• 软件缺陷的生命周期指的是一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程。

简单的软件缺陷生命周期：

1、发现——打开：测试人员找到软件缺陷并将软件缺陷提交给开发人员；

2、打开——修复：开发人员再现、修复缺陷，然后提交测试人员去验证；

3、修复——关闭：测试人员验证修复过的软件，关闭已不存在的缺陷。

但是这是一种理想的状态，在实际的工作中是很难有这样的顺利的，需要考虑的各种情况都还是非常多的。

复杂的软件缺陷生命周期：

1、新建一个软件缺陷，这个软件缺陷是（open）状态，进行 bug

审查，不是代码问题，就

是设计需要修改；

2、新建一个软件缺陷，这个软件缺陷是（open）状态，进行 bug

审查，以后修改的，就可

以延期；

3、新建一个软件缺陷，这个软件缺陷是（open）状态，进行

bug

审查，实际没有这个 bug，

可以将其关闭；

4、新建一个软件缺陷，这个软件缺陷是（open）状态，看是否清楚可重现，如果不能重现，就是缺少信息，需要返回到（open）状态；如果能够重现，就进行修正，修正后关闭，进行

回归测试。

### 13.测试分析测试用例注意（事项）？

#### 1.为什么要写用例：

我们编写测试用例，有如下的好处：

便于团队交流：假如说一个测试团队有 10 个成员，大家测试的时候都各自为政，没有统一的标准，测试的效率无疑会大打折扣；如果大家都遵循统一的用例规范去写，就会解决这一问题。

便于重复测试：大家知道，软件在实际开发过程中是会有不同版本的，比如会从 1.0 升级到 10.0，那么如果不写测试用例的话，在测试 10.0 版本的时候，你能完全记得 1.0 版本时你

做过哪些测试吗？测试用例就像一个备忘录一样，便于重复测试。

便于跟踪统计：这一点是针对测试经理或是项目经理来说的，项目负责人通过看测试用例的执行情况，就能了解到项目目前的概况，比如已经执行了哪些测试，还有哪些测试没有执行，测试没有通过的地方主要集中在哪些模块等。

便于用户自测：尤其是项目软件，有的时候用户希望自己测试一下软件产品，但是用户大都是非专业人士，他需要根据你写好的用例来更好的检验产品的质量

说了这么多编写测试用例的优点，那它有没有缺点呢？有一个明显的缺点就是需要花费大量的时间，通常编写测试用例的时间比实际执行测试的时间还要长，这一点大家会在实际工作中有深刻的体会

#### 2.什么时候写用例：

什么时候写用例？这个问题没有统一的标准答案，但有一点可以肯定，就是测试用例要尽早编写。大家认为在哪个阶段开始写用例比较好呢？

通常，我们都会在测试设计阶段来写用例，即《需求规格说明书》和《测试计划》都已完成之后

#### 3.由谁来写测试用例

有的读者会说，当然是测试人员来写用例了！

可是测试人员又会有不同的角色，一般分为测试经理，测试设计人员，测试执行人员和测试工具开发人员等，一般测试用例是由测试设计人员来编写，由测试执行人员来执行，这就要求测试设计人员有一定的用例设计经验，并对被测试的系统有深入的了解。

但是在很多小公司里面，区分的不是这么明显，一个测试人员往往会身兼数职，既是测试组长，又是测试设计人员，又是测试执行人员。项目组里就你一个测试工程师，你不写用例谁写啊！

#### 4.根据什么写测试用例

我们编写测试用例的唯一标准就是用户需求，具体的参考资料就是《系统需求规格说明书》和软件原型，其中软件原型指的是没有嵌入全部源代码的软件界面，比如我做一个电子商务网站，为了尽快能给用户演示，我只是用 html 语言作一些静态页面，并没有编写动态的程序，这就是一个软件原型，它也看作是需求的一部分。

## 二. 瑞星笔试题(15道)

### 1.一台计算机的 IP

是 192.168.10.71 子网掩码 255.255.255.64 与 192.168.10.201 是同一局域网吗?

你的子网掩码不对。

不可能出现 255.255.255.64 的子网掩码。

另外，这个题也不能说成“同一局域网”，局域网是针对物理的拓扑结构而言。

事实上，我们研究的是否在同一子网的一些 IP，往往都是同一个局域网内。

针对此题：

.....

子网掩码为

255.255.128.0

时，是同一子网。

子网掩码为

255.255.255.0

时，是同一子网。

子网掩码为

255.255.255.128

时，不是同一子网。

子网掩码为

255.255.255.192

时，不是同一子网。

.....

### 2.internet 中 e-mail 协议,IE

的协议,NAT 是什么,有什么好处,能带来什么问题?DNS

是什

么,它

是如何工作的?

NAT 全称 Network Address Translation，中文解释为“网络地址转换”。NAT 是一种 IETF

(Internet Engineering Task Force) 的标准，简单描述其功能就是让处于内网的计算机能够通过

过 NAT 的作用透明的访问外网的互联网资源。NAT

的功能一般集成在路由器、防火墙或

者

单独的 NAT

设备中。

DNS

全名是

Domain Name System, 透过 DNS 系统,

我们可以由一部机器的 domain name

查其

IP, 也可以由机器的 IP 反查它的 domain name,

除此之外

DNS 还与 Mail System

结合, 提供

Mail routing 的功能.

DNS 分为

Client 和

Server, Client

扮演发问的角色, 也就是问 Server 一个 Domain Name,

而

Server 必须要回答此

Domain Name

的真正 IP 地址。而当地的

DNS 先会查自己的资料

库。

如果自己的资料库没有, 则会往该 DNS 上所设的 DNS 询问, 依此得到答案之后, 将收到

的答案存起来, 并回答客户。

3.PROXY 是如何工作的?

Proxy 是什么呢, 是代理。普通的因特网访问是一个典型的客户机与服务器结构: 用户利用计算机上的客户端程序, 如浏览器发出请求, 远端 WWW 服务器程序响应请求并提供相应的数据。而 Proxy 处于客户机与服务器之间, 对于服务器来说, Proxy 是客户机, Proxy 提出请求, 服务器响应; 对于客户机来说, Proxy 是服务器, 它接受客户机的请求, 并将服务器上传来的数据转给客户机。它的作用很象现实生活中的代理商。因此 Proxy Server 的中文名称就是代理服务器。

Proxy Server

的工作原理是: 当客户在浏览器中设置好 Proxy Server 后, 你使用浏览器访问

所有 WWW

站点的请求都不会直接发给目的主机, 而是先发给代理服务器, 代理服务器接

受了客户的请求以后, 由代理服务器向目的主机发出请求, 并接受目的主机的数据, 存于代理服务器的硬盘中, 然后再由代理服务器将客户要求的数据发给客户。

4.win2k 系统内 AT 命令完成什么功能,Messenger 服务是做什么,怎么使用?

AT 命令可在指定时间和日期、在指定计算机上运行命令和程序。

Messenger 服务:发送和接收系统管理员或者“警报器”服务传递的消息。

5 进程,线程的定义及区别

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位.

线程是进程的一个实体,是 CPU 调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源.

一个线程可以创建和撤销另一个线程;同一个进程中的多个线程之间可以并发执行 6,32 位操作系统内,1 进程地址空间多大,进程空间与物理内存有什么关系?

7

网络攻击常用的手段,防火墙如何保证安全.

8

如何配静态 IP,如何测网络内 2 台计算机通不通,PING 一次返几个数据包?

9WIN9X

与 WINNT 以上操作系统有"服务"吗,服务是什么,如何停止服务?

10AD 在

WIN2KSERVER 上建需什么文件格式,AD 是什么?XP 多用户下"注销"与"切换"的区

别.

11UDP 可以跨网段发送吗?

12

最简单的确认远程计算机(win2K 以上)某个监听端口是正常建立的?

13

软件测试的定义,测试工作是枯燥反复的,你是如何理解的?黑盒,白盒,回归,压力测试的定义.

14winrunner,loadrunner 是什么,区别

15

磁盘分区如何分类,请举例说明安装操作系统的注意事项.

(1

小时答题)

三. 中软的面试题

一. 简答题.

1. 避免死锁的方法有哪些?

产生死锁的四个必要条件:

(1)

互斥条件: 一个资源每次只能被一个进程使用。

(2)

请求与保持条件: 一个进程因请求资源而阻塞时, 对已获得的资源保持不放。

(3)

不剥夺条件:进程已获得的资源, 在未使用完之前, 不能强行剥夺。

(4)

循环等待条件:若干进程之间形成一种头尾相接的循环等待资源关系。

这四个条件是死锁的必要条件, 只要系统发生死锁, 这些条件必然成立, 而只要上述条件之一不满足, 就不会发生死锁。

2.

在

Sybase 数据库中注册用户与数据库用户有什么区别?

3.

在

MS SQL\_Server 数据库中通过什么约束保证数据库的实体完整性

可以通过建立唯一的索引、PRIMARY KEY 约束、UNIQUE 约束或 IDENTITY 约束来实现

实体完整性

4. 内存有哪几种存储组织结构.请分别加以说明

5. JAVA 中的 Wait() 和 notify()方法使用时应注意些什么?

6.

用户输入一个整数.系统判断,并输出是负数还是非负数,请设计测试用例.

7.

操作系统中的同步和互斥解决了什么问题

8. UNIX 中

init

二. 编写类

String 的构造函数,析构函数和赋值函数

已知类 String 的原型为

```
class string
{
public:
string(const char *str=null);//普通构造函数
string(const string &other);//拷贝构造函数
---string(void);
string &operate=(const string &other);//赋值函数
private:
char * m-data;//用于保存字符串
};
```

请编写 string 的上述 4 个函数

三. 有关内存的思考题

1. void getmemory(char \*p)

```
{ p=(char*)malloc(100);
}
```

void test(void)

```
{
char * str =null;
getmemory(str);
strcpy(str,"hello,world");
printf(str);
}
```

请问运行 Test 函数会有什么样的结果

2. char\*getmemory(void)

```
{ char p[]="hello world";
return p;
}
```

void test(void)

```
{
char *str=null;
str=Getmemory();
printf(str);
} 请问运行 Test 函数会有什么样的结果.
```

三. 奇虎面试题

### 前三道程序题

（下面的题不排序，有笔试题，也有面试题）

4、怎么划分缺陷的等级？

5、怎么评价软件工程师？

6、软件工程师的素质是什么？

7、怎么看待软件测试？

8、软件测试是一个什么样的行业？

9、图书（图书号，图书名，作者编号，出版社，出版日期）

作者（作者姓名，作者编号，年龄，性别）

用 SQL 语句查询年龄小于平均年龄的作者姓名、图书名，出版社。

10、你的职业生涯规划

11、测一个三角形是普通三角形、等腰三角形、等边三角形的流程图，测试用例。

12、写出你常用的测试工具。

13、Irdrunner 分哪三部分？

14、希望以后的软件测试是怎么样的一个行业？

15、软件测试项目从什么时候开始？

我答：从软件项目的需要分析开始。

问：为什么从需求分析开始？有什么作用？

### 四. 北京博彦科技笔试+面试

#### 笔试题

1. 文件格式系统有哪几种类型？分别说说 win95、win98、winMe、w2k、winNT、winXP 分别支持那些文件系统。

FAT(File Allocation Table)是“文件分配表”的意思。

对我们来说，它的意义在于对硬盘分区的管理。FAT16、FAT32、NTFS 是目前最常见的三种文件系统。

Win95: FAT16 和 FAT32

Win98: FAT16,FAT32

winMe:FAT16,FAT32

w2k: FAT(FAT16),FAT32,NTFS

winNT: FAT16/FAT32/NTFS

winXP:FAT16,FAT32,NTFS

2. 分别填入一个语句，完成下面的函数，通过递归计算数组 a[100]的前 n 个数之和。

Int sum ( int a[],int n )

```
{
if (n>0) return ____ sum(a[], n--) + a[n] ____;
else return ____ a[n]____; // 其实就是 a[0]
}
```

//一直递归到 0，然后逐级返回，实现累加

3. 写出你所知道的 3 种常用的排序方法，并用其中一种方法设计出程序为数组 a[100]排序。

常用的排序算法有很多：

冒泡，快速排序，直接插入，希尔排序，选择排序，堆排序，归并排序！

就举冒泡排序(c++)：



```
void bubblesort()
{
    for (i = 1; i < max; i++)
    {
        for (j = max - 1; j >= i; j --)
            if (a[j + 1] < a[j]) //小则交换
            {
                a[0] = a[j + 1];
                a[j + 1] = a[j];
                a[j] = a[0];
            }
    }
}
```

4. 什么是兼容性测试？兼容性测试侧重哪些方面，请按照优先级用矩阵图表列出。

（这题的第二问我不会答，所以原题目记得不是很清楚，大家能看明白问什么就好）

兼容性是指协调性，

1. 硬件上就是说你的电脑的各个部件，CPU，显卡等等组装到一起以后的情况，会不会相互有影响，不能很好的运作。

2. 软件上就是说你的电脑的软件之间能否很好的运做，会不会有影响啊？还有软件和硬件之间能否发挥很好的效率工作，会不会影响导致系统的崩溃。

#### 1、 平台测试

市场上有很多不同的操作系统类型，最常见的有 Windows、Unix、Macintosh、Linux 等。

Web 应用系统的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。这样，就可能会发生兼容性问题，同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。

因此，在 Web

系统发布之前，需要在各种操作系统下对 Web 系统进行兼容性测试。

#### 2、浏览器测试

浏览器是 Web

客户端最核心的构件，来自不同厂商的浏览器对 Java、JavaScript、

ActiveX、

plug-ins

或不同的 HTML 规格有不同的支持。例如，ActiveX 是

Microsoft 的产品，是为

Internet

Explorer

而设计的，JavaScript 是 Netscape 的产品，Java 是 Sun

的产品等等。另外，框

架和

层次结构风格在不同的浏览器中也有不同的显示，甚至根本不显示。不同的浏览器对安全性和 Java 的设置也不一样。

测试浏览器兼容性的一个方法是创建一个兼容性矩阵。在这个矩阵中，测试不同厂商、不同版本的浏览器对某些构件和设置的适应性。

5. 我现在有个程序，发现在 WIN98 上运行得很慢，怎么判别是程序存在问题还是软硬件系

统存在问题？

可能是病毒或者恶意程序啊,如果是程序运行慢那肯定是软件问题,硬件问题主要表现在点不亮机子和报警.

解决方法,安装杀毒软件和优化软件.

6. 翻译，中——英，有关 P2P 点对点文件传输的原理。

7. 翻译，英——中，有关互联网的发展对商务、学习、交流的影响。

笔试完了是初步的面试

先问了个问题：FAT16/FAT32/NTFS 哪个的安全性最好，为什么？（不会答）

NTFS 文件系统是一个基于安全性的文件系统，是 Windows NT 所采用的独特的文件系统结

构，它是建立在保护文件和目录数据基础上，同时照顾节省存储资源、减少磁盘占用量的一种先进的文件系统。使用非常广泛的 Windows NT 4.0 采用的就是 NTFS 4.0 文件系统，相信

它所带来的强大的系统安全性一定给广大用户留下了深刻的印象。Win 2000 采用了更新版

本的 NTFS 文件系统——NTFS 5.0，它的推出使得用户不但可以像 Win 9X 那样方便快捷

地

操作和管理计算机，同时也可享受到 NTFS 所带来的系统安全性。

又做了两道题，

一题是关于 C++类的继承，看程序写出输出结果，A 是虚类，B 继承 A，跟一般 C++的书上

的习题差不多。

一题是写出在 32 位机器下，计算几个变量的 size，

最后用英文介绍一下自己

黑盒测试的测试用例设计方法

目前黑盒测试的测试用例设计方法有 5 种：

1.

等价类划分

2.

边界值分析

3.

错误推测法

4.

因果图

5.

功能图

一、等价类划分

等价类划分设计方法是把所有可能的输入数据，即程序的输入域划分成若干部分（子集），然后从每一个子集中选取少量具有代表性的数据作为测试用例。

等价类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是

等效的。并合理地假定：测试某等价类的代表值就等于对这一类其他值的测试。

等价类划分有两种不同的情况：有效等价类和无效等价类。设计时要同时考虑这两种等价类。

下面给出 6 条确定等价类的原则：

1. 在输入条件规定了取值范围或值的个数的情况下，则可以确立一个有效等价类和两个无效等价类。

2. 在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情况下，则可以确立一个有效等价类和一个无效等价类。

3.

在输入条件是一个布尔量的情况下，可以确立一个有效等价类和一个无效等价类。

4.

在规定了输入数据的一组值（假定  $n$  个），并且程序要对每一个输入值分别处理的情况下，可以确立  $n$  个有效等价类和一个无效等价类。

5. 在规定了输入数据必须遵守的规则的情况下，可以确立一个有效等价类（符合规则）

和若干个无效等价类（从不同角度违反规则）。

6. 在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价类进一步的划分为更小的等价类。

在确立了等价类后，可建立等价类表，列出所有划分出的等价类。然后从划分出的等价类中按以下的 3 个原则设计测试用例：

？

为每一个等价类规定一个唯一的编号

？

设计一个新的测试用例，使其尽可能多的覆盖尚未被覆盖的有效等价类，重复这一步，直到所有的有效等价类都被覆盖为止。

？设计一个新的测试用例，使其仅覆盖一个尚未被覆盖的无效等价类，重复这一步，直到所有的无效等价类都被覆盖为止。

例：程序规定：输入三个整数作为三边的边长构成三角形。当此三角形为一般三角形、等腰三角形、等边三角形时，分别作计算。用等价类划分方法为该程序进行测试用例设计。

解：设  $a$ 、 $b$ 、 $c$  代表三角形的三条边。

1) 分析题目中给出的和隐含的对输入条件的要求：

a)

整数

b)

3 个数

c)

非零数

d)

正数

e)

两边之和大于第三边

f)

等腰

g)

等边

2) 列出等价类表并编号

3) 列出覆盖上述等价类的测试用例，如下表：

## 二、边界值分析法

使用边界值分析方法设计测试用例，首先：应确定边界情况。通常输入和输出等价类的边界，就是应着重测试的边界情况。其次，应选取正好等于、刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值作为测试数据。

基于边界值分析方法选择测试用例的原则：

1. 如果输入条件规定了值的范围，应取刚达到这个范围的边界值，以及刚刚超过这个范围边界的值作为测试输入的数据。

2. 如果输入条件规定了值的个数，应用最大个数、最小个数、比最小个数少一、比最大个数多一的数作为测试输入的数据。

3.

根据规格说明的每个输出条件，使用前面的原则

1。

4.

根据规格说明的每个输出条件，使用前面的原则

2。

5.

如果程序的规格说明给出的输入域或输出域是有序集合，则应选取集合的第一个元素和最后一个元素作为测试用例数据。

6. 如果程序中使用了一个内部数据结构，应当选择这个内部数据结构边界上的值作为测试用例。

7. 分析规格说明，找出其他可能的边界条件。

## 三、错误推测法

错误推测法就是根据经验和直觉推测程序中所有可能存在的各种错误，从而有针对性地设计测试用例的方法。

基本思路：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：输入数据和输出数据为 0 的情况。

例：现有一个学生标准化考试批阅试卷，产生成绩报告的程序。其规格说明如下：程序的输入文件由一些有 80 个字符的记录组成，所有记录分为 3 组，如图：

1、标题：改组只有一个记录，其内容是成绩报告的名字。

2、各题的标准答案：每个记录均在第 80 个字符处标以数字 2。该组的记录：

第一个记录：第 1~3 个字符为试题数（1~999）。第 10~59 个字符是 1~50 题的标准答案（每

个合法字符表示一个答案）。

第二个记录：是第 51~100 题的标准答案。

.....

3、学生的答案：每个记录均在第 80 个字符处标以数字 3。每个学生的答卷在若干个记录中

给出。

学号：1~9 个字符

1~50 题的答案：10~59。当大于 50 题时，在第二、三、.....个记录中给出。

学生人数不超过

200，试题数不超过 999。

程序的输出有 4

个报告：

- a)按学号排列的成绩单，列出每个学生的成绩、名次。
- b)按学生成绩排序的成绩单。
- c)平均分数及标准偏差的报告
- d)试题分析报告。按试题号排序，列出各题学生答对的百分比。

解答一：采用边界值分析方法，分析和设计测试用例。分别考虑输入条件和输出条件，以及边界条件。下表列出了输入条件及相应的测试用例。

下表为输出条件及相应的测试用例表。

解答二：采用错误推测法还可补充设计一些测试用例：

1.

程序是否把空格作为回答

2.

在回答记录中混有标准答案记录

3.

除了标题记录外，还有一些的记录最后一个字符即不是 2 也不是 3

4.

有两个学生的学号相同

5.

试题数是负数。

#### 四、因果图法

因果图法是一种适合于描述对于多种条件的组合、相应产生多个动作的形式的测试用例设计方法。

利用因果图生成测试用例的基本步骤：

- 1. 分析软件规格说明描述中那些是原因，那些是结果，并给每个原因和结果赋予一个标识符。
- 2. 分析软件规格说明描述的语义。找出原因和结果之间、原因和原因之间的关系，根据这些关系，画出因果图。
- 3. 在因果图上用一些记号表明约束或限制条件。

4.

把因果图转换为判定表。

5.

把判定表的每一列拿出来作为依据，设计测试用例。

例：第一列字符必须是 A 或 B，第二列字符必须是一个数字，在此情况下进行文件的修改，但如果第一列字符不正确，则给出信息 L；如果第二列字符不是数字，则给出信息 M。

解 1、画出因果关系表和因果图。

2、根据因果图建立判定表。

按条件的各种组合情况产生对应的动作。原因 1 和原因 2 不能同时成立，故可排除这种情况。

从判定表可设计出测试用例：6 个测试用例是所需的数据。

## 美团网软测面试题回忆版本

1. 您在从事性能测试工作时，是否使用过一些测试工具？如果有，请试述该工具的工作原理，并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。

2.

您认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？

3.

在您以往的工作中，一条软件缺陷（或者叫 Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？

4. 您以往所从事的软件测试工作中，是否使用了一些工具来进行软件缺陷（Bug）的管理？如果有，请结合该工具描述软件缺陷（Bug）跟踪管理的流程。

5. 您认为在测试人员同开发人员的沟通过程中，如何提高沟通的效率和改善沟通的效果？维持测试人员同开发团队中其他成员良好的人际关系的关键是什么？

6. 在您以往的测试工作中，最让您感到不满意或者不堪回首的事情是什么？您是如何来对待这些事情的？

7. 在即将完成这次笔试前，您是否愿意谈一些自己在以往的学习和工作中获得的工作经验 and 心得体会？（可以包括软件测试、过程改进、软件开发或者与此无关的其他方面）

8. 你对测试最大的兴趣在哪里？为什么？

最大的兴趣就是测试有难度，有挑战性！做测试越久越能感觉到做好测试有多难。曾经

在无忧测试网上看到一篇文章，是关于如何做好一名测试工程师。一共罗列了十一二点，有部分是和人的性格有关，有部分需要后天的努力。但除了性格有关的 1、2 点我没有

把握，其他点我都很有信心做好它。

刚开始进入测试行业时，对测试的认识是从无忧测试网上了解到的一些资料，当时是冲着做测试需要很多技能才能做的好，虽然入门容易，但做好很难，比开发更难，虽然当时我很想做开发（学校专业课我基本上不缺席，因为我喜欢我的专业），但看到测试比开发更难更有挑战性，想做好测试的意志就更坚定了。

不到一年半的测试工作中，当时的感动和热情没有减退一点（即使环境问题以及自身经验，技术的不足，做测试的你一定也能理解）。

我觉得做测试整个过程中有 2 点让我觉得很有难度（对我来说，有难度的东西我就非常

感兴趣），第一是测试用例的设计，因为测试的精华就在测试用例的设计上了，要在版本出来之前，把用例写好，用什么测试方法写？（也就是测试计划或测试策略），如果你

刚测试一个新任务时，你得花一定的时间去消化业务需求和技术基础，业务需求很好理解（多和产品经理和开发人员沟通就能达到目的），而技术基础可就没那么简单了，这需要你自觉的学习能力，比如说网站吧，最基本的技术知识你要知道网站内部是怎么运作的，后台是怎么响应用户请求的？测试环境如何搭建？这些都需要最早的学好。至少在开始测试之前能做好基本的准备，可能会遇到什么难题？需求细节是不是没有确定好？这些问题都能在设计用例的时候发现。

第二是发现 BUG 的时候了，这应该是测试人员最基本的任务了，一般按测试用例开始测试就能发现大部分的 bug，还有一部分 bug 需要测试的过程中更了解所测版本的情



况获得更多信息，补充测试用例，测试出

bug。还有如何发现 bug?这就需要在测试

用

例有效的情况下，通过细心和耐心去发现

bug 了，每个用例都有可能发现 bug，每

个

地方都有可能出错，所以测试过程中思维要清晰（测试过程数据流及结果都得看仔细了，

bug 都在里面发现的）。如何描述 bug 也很有讲究，bug 在什么情况下会产生，如

果条

件变化一点点，就不会有这个 bug，以哪些最少的操作步骤就能重现这个 bug，这个

bug 产生的规律是什么?如果你够厉害的话，可以帮开发人员初步定位问题。

9. 你的测试职业发展是什么?

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级

测试工程师奔去。而且我也有初步的职业规划，前 3 年累积测试经验，按如何做好测

试

工程师的 11, 12 点要求自己，不断的更新自己改正自己，做好测试任务。

10. 你自认为测试的优势在哪里

优势在于我对测试坚定不移的信心和热情，虽然经验还不够，但测试需要的基本技能我

有信心在工作中得以发挥。

11. 你以前工作时的测试流程是什么?

公司对测试流程没有规定如何做，但每个测试人员都有自己的一套测试流程。我说下我

1 年来不断改正（自己总结，吸取同行的方法）后的流程吧。需求评审（有开发人员，产品经理，测试人员，项目经理） -> 需求确定(出一份确定的需求文档) -> 开发设计文档（开发人员在开始写代码前就能输出设计文档） -> 想好测试策略，写出测试用例 -> 发给开发人员和测试经理看看（非正式的评审用例） -> 接到测试版本 -> 执行测试用例（中间可能会补充用例） -> 提交 bug（有些 bug 需要开发人员的确定（严重级别的，或突然发现的在测试用例范围之外的，难以重现的），有些可以直接录制进 TD） ->

开发人员修改（可以在测试过程中快速的修改） -> 回归测试（可能又会发现新问题，再按流程开始跑）。

## 12. 当开发人员说不是 BUG

时，你如何应付？

开发人员说不是 bug，有 2

种情况，一是需求没有确定，所以我可以这么做，这个时候

可以找来产品经理进行确认，需不需要改动，3 方商量确定好后再看要不要改。二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是 BUG 的依据是什么？如果被用户发现或出了问题，会有什么不良结果？程序员可能会给你很多理由，

你可以对他的解释进行反驳。如果还是不行，那我可以给这个问题提出来，跟开发经理和测试经理进行确认，如果要修改就改，如果不要修改就不改。其实有些真的不是 bug，我

也只是建议的方式写进 TD 中，如果开发人员不修改也没有大问题。如果确定是 bug

的话，一定要坚持自己的立场，让问题得到最后的确认。

13. 你为什么想离开目前的职务？

因为公司运作情况并不理想，公司需要调整部门体系，公司考虑到缩减部门人员，所以大批量的裁员（有 6, 7 个），这是我的第一份工作，对公司也有较深的感情，因为在这

里我找到了职业理想（就是测试），所以公司需要精简人员，我自愿退出。虽然很舍不得，但我将会有新的发挥能力的舞台。

14: 你对我们公司了解有多少？

15: 你找工作时，最重要的考虑因素为何？

工作的性质和内容是否能让我发挥所长，并不断成长。

16: 为什么我们应该录取你？

您可以由我过去的工作表现所呈现的客观数据，明显地看出我全力以赴的工作态度。

17: 请谈谈你个人的最大特色。

我的坚持度很高，事情没有做到一个令人满意的结果，绝不罢手。

18.白箱测试和黑箱测试是什么?什么是回归测试?

19. 单元测试、集成测试、系统测试的侧重点是什么？

20. 设计用例的方法、依据有那些？

21. 一个测试工程师应具备那些素质和技能？

22.集成测试通常都有那些策略？

23.你用过的测试工具的主要功能、性能及其他？

24.一个缺陷测试报告的组成

25.基于 WEB 信息管理系统测试时应考虑的因素有哪些？

26.软件测试项目从什么时候开始，？为什么？

27.需求测试注意事项有哪些？

28.测试分析测试用例注意（事项）？

## 拼多多软件测试校招面试题

01. 为什么要在一个团队中开展软件测试工作？

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

02. 您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

我曾经做过 web 测试，后台测试，客户端软件，其中包括功能测试，性能测试，用户体验测试。最擅长的是功能测试

03. 您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试.....）

测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。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。采用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。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范（快捷键之类的），是否美观（能否吸引用户的注意力），是否安全（尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告）?做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的，然后再考虑该功能点的性能测试

04. 您认为做好测试用例设计工作的关键是什么？

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

05. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

- 1、是否有不正确或遗漏的功能？
- 2、在接口上，输入是否能正确的接受？能否输出正确的结果？
- 3、是否有数据结构错误或外部信息（例如数据文件）访问错误？
- 4、性能上是否能够满足要求？
- 5、是否有初始化或终止性错误？

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对

程序模块进行如下检查：

- 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”与取“假”的两种情况都能至少测一遍。
- 3、在循环的边界和运行的界限内执行循环体。
- 4、测试内部数据结构的有效性，等等。

单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。（常见的联调测试）

系统测试的目的是对最终软件系统进行全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。



验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

## 瑞星软通测试工程师笔试题

### 软通动力面试

1. 白箱测试和黑箱测试是什么?什么是回归测试?
2. 单元测试、集成测试、系统测试的侧重点是什么?
3. 设计用例的方法、依据有那些?
4. 一个测试工程师应具备那些素质和技能?
5. 集成测试通常都有那些策略?
6. 你用过的测试工具的主要功能、性能及其他?
7. 一个缺陷测试报告的组成
8. 基于 WEB 信息管理系统测试时应考虑的因素有哪些?
9. 软件本地化测试比功能测试都有哪些方面需要注意?
10. 软件测试项目从什么时候开始, ? 为什么?
11. 需求测试注意事项有哪些?
12. 简述一下缺陷的生命周期
13. 测试分析测试用例注意(事项)?

### 瑞星笔试题

1. 一台计算机的 IP 是 192.168.10.71 子网掩码 255.255.255.64 与 192.168.10.201 是同一局域网吗?
2. internet 中 e-mail 协议, IE 的协议, NAT 是什么, 有什么好处, 能带来什么问题? DNS 是什么, 它是如何工作的?
3. PROXY 是如何工作的?
4. win2k 系统内 AT 命令完成什么功能, Messenger 服务是做什么, 怎么使用?
- 5 进程, 线程的定义及区别
- 6, 32 位操作系统内, 1 进程地址空间多大, 进程空间与物理内存有什么关系?

7

网络攻击常用的手段, 防火墙如何保证安全.

8

如何配静态 IP, 如何测网络内 2 台计算机通不通, PING 一次返几个数据包?

9WIN9X 与 WINNT 以上操作系统有“服务”吗, 服务是什么, 如何停止服务?

10AD 在 WIN2KSERVER 上建需什么文件格式, AD 是什么?XP 多用户下“注销”与“切

换”的区别.

11UDP 可以跨网段发送吗?

12

最简单的确认远程计算机(win2K 以上)某个监听端口是正常建立的?

13

软件测试的定义, 测试工作是枯燥反复的, 你是如何理解的?黑盒, 白盒, 回归, 压力测试的定义.

14winrunner, loadrunner 是什么, 区别

15 磁盘分区如何分类, 请举例说明安装操作系统的注意事项.

中软的面试题

一. 简答题.

1.

避免死锁的方法有哪些?

2.

在

Sybase 数据库中注册用户与数据库用户有什么区别?

3.

在

MS SQL\_Server 数据库中通过什么约束保证数据库的实体完整性

4.

内存有哪几种存储组织结构.请分别加以说明 <https://space.bilibili.com/592291739> 软件测试视频教学

5. JAVA 中的 Wait() 和 notify()方法使用时应注意些什么?

6.

用户输入一个整数.系统判断,并输出是负数还是非负数,请设计测试用例.

7.

操作系统中的同步和互斥解决了什么问题

8. UNIX 中

init

二. 编写类

String 的构造函数,析构函数和赋值函数

已知类 String 的原型为

class string

{

public:

string(const char \*str=null);//普通构造函数

string(const string &other);//拷贝构造函数

```
---string(void);
string &operate=(const string &other);//赋值函数
private:
char * m-data;//用于保存字符串
};
```

请编写 `string` 的上述 4 个函数

三. 有关内存的思考题

1. `void getmemory(char *p)`

```
{ p=(char*)malloc(100);
}
```

`void test(void)`

```
{
char * str =null;
getmemory(str);
strcpy(str,"hello,world");
printf(str);
}
```

请问运行 `Test` 函数会有什么样的结果

2. `char*getmemory(void)`

```
{ char p[]="hello world";
return p;
}
```

`void test(void)`

```
{
char *str=null;
str=Getmemory();
printf(str);
}
```

请问运行 `Test` 函数会有什么样的结果.

奇虎面试题

前三道程序题

(下面的题不排序, 有笔试题, 也有面试题)

4、怎么划分缺陷的等级?

5、怎么评价软件工程师?

6、软件工程师的素质是什么?

7、怎么看待软件测试?

8、软件测试是一个什么样的行业?

9、图书(图书号, 图书名, 作者编号, 出版社, 出版日期)

作者(作者姓名, 作者编号, 年龄, 性别)

用 `SQL` 语句查询年龄小于平均年龄的作者姓名、图书名, 出版社。

10、你的职业生涯规划

11、测一个三角形是普通三角形、等腰三角形、等边三角形的流程图, 测试用例。

12、写出你常用的测试工具。

13、`lordrunner` 分哪三部分?

14、希望以后的软件测试是怎么样一个行业？

15、.软件测试项目从什么时候开始？

我答：从软件项目的需要分析开始。

问：为什么从需求分析开始？有什么作用？

北京博彦科技笔试+面试

笔试题

1. 文件格式系统有哪几种类型？分别说说 win95、win98、winMe、w2k、winNT、winXP

分别支持

那些文件系统。

2. 分别填入一个语句，完成下面的函数，通过递归计算数组 a[100]的前 n 个数之和。

Int sum ( int a[],int n )

```
{
if (n>0) return _____;
else return _____;
}
```

3. 写出你所知道的 3 种常用的排序方法，并用其中一种方法设计出程序为数组 a[100]排序。

4. 什么是兼容性测试？兼容性测试侧重哪些方面，请按照优先级用矩阵图表列出。

（这题的第二问我不会答，所以原题目记得不是很清楚，大家能看明白问什么就好）

5. 我现在有个程序，发现在 WIN98 上运行得很慢，怎么判别是程序存在问题还是软硬件系统存在问题？

6. 翻译，中——英，有关 P2P 点对点文件传输的原理。

7. 翻译，英——中，有关互联网的发展对商务、学习、交流的影响。

笔试完了是初步的面试

先问了个问题：FAT16/FAT32/NTFS 哪个的安全性最好，为什么？（不会答）

又做了两道题，

一题是关于 C++类的继承，看程序写出输出结果，A 是虚类，B 继承 A，跟一般 C++的书上的习题差不多。

一题是写出在 32 位机器下，计算几个变量的 size，

\*\*\*\*\*

Question 53. (问答)

请根据以下程序片段，设计最少的测试用例实现条件覆盖

If((A>1)AND(B=0))Then X=X/A65c712c49df56926c2fa

If((A=2)OR(X>1))Then X=X+1

Printf("X=%d

",x)

Question 54. (问答)

一套完整的测试应该由哪些阶段组成？分别阐述一下各个阶段。e5bc528021116d4aac36

Question 55. (问答)

请列出关于一个印有文字的水杯，你能想到的测试用例

1.白箱测试和黑箱测试是什么？什么是回归测试？

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。自动回归测试将大幅降低系统测试、维护升级等阶段的成本。回归测试包括两部分：函数本身的测试、其他代码的测试。

### 2.单元测试、集成测试、系统测试的侧重点是什么？

单元测试是在软件开发过程中要进行的最低级别的测试活动，在单元测试活动中，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

集成测试，也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求，组装成为子系统或系统，进行集成测试。实践表明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。程序在某些局部反映不出来的问题，在全局上很可能暴露出来，影响功能的实现。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。

### 3.设计用例的方法、依据有那些？

白盒测试：逻辑覆盖法，主要包括语句覆盖，判断覆盖，条件覆盖，判断-条件覆盖，路径覆盖

黑盒测试：等价划分类，边界值分析，错误推测法。

### 5.集成测试通常都有那些策略？

- 1、在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
- 2、各个子功能组合起来，能否达到预期要求的父功能；
- 3、一个模块的功能是否会对另一个模块的功能产生不利的影响；
- 4、全局数据结构是否有问题；
- 5、单个模块的误差积累起来，是否会放大，从而达到不可接受的程度。

学网,收藏本篇文章 [1] [2] [3]

\$False\$

### pan>7.一个缺陷测试报告的组成

缺陷的标题，缺陷的基本信息，复现缺陷的操作步骤，缺陷的实际结果描述，期望的正确结果描述，注释文字和截取的缺陷图象。

### 8.基于 WEB 信息管理系统测试时应考虑的因素有哪些？

### 9.软件本地化测试比功能测试都有哪些方面需要注意？

软件本地化测试的目的：

软件本地化测试的测试策略：1.本地化软件要在各种本地化操作系统上安装并测试。2.源语言软件安装在另一台相同源语言操作系统上，作为对比测试。3.重点测试因本地化引起的软件的功能和软件界面的错误。4.

测试本地化软件的翻译质量。5.手工测试和自动测试相结合。

### 11.需求测试注意事项有哪些？

一个良好的需求应当具有一下特点：

完整性：每一项需求都必须将所要实现的功能描述清楚，以使开发人员获得设计和实现这些功能所需的所有必要信息。

正确性：每一项需求都必须准确地陈述其要开发的功能。

一致性：一致性是指与其它软件需求或高层（系统，业务）需求不相矛盾。

可行性：每一项需求都必须是在已知系统和环境的权能和限制范围内可以实施的。

无二义性：对所有需求说明的读者都只能有一个明确统一的解释，由于自然语言极易导致二义性，

所以尽量把每项需求用简洁明了的用户性的语言表达出来。

健壮性：需求的说明中是否对可能出现的异常进行了分析，并且对这些异常进行了容错处理。

必要性：“必要性”可以理解为每项需求都是用来授权你编写文档的“根源”。要使每项需求都能回溯至某项客户的输入，如 **Use Case** 或别的来源。

可测试性：每项需求都能通过设计测试用例或其它的验证方法来进行测试。

可修改性：每项需求只应在 **SRS** 中出现一次。这样更改时易于保持一致性。欢迎光临学网,点击这里查看更多文章教程 [1] [2] [3]

另外，使用目录表、索引和相互参照列表方法将使软件需求规格说明书更容易修改。

可跟踪性：应能在每项软件需求与它的根源和设计元素、源代码、测试用例之间建立起链接链，这种可跟踪性要求每项需求以一种结构化的，粒度好（**fine-grained**）的方式编写并单独标明，而不是大段大段的叙述。

## 12.测试分析测试用例注意（事项）？

\*\*\*\*\*

有关内存的思考题

```
void GetMemory(char *p)
```

```
{
```

```
p = (char *)malloc(100);
```

```
}
```

```
void Test(void)
```

```
{
```

```
char *str = NULL;
```

```
GetMemory(str);
```

```
strcpy(str, "hello world");
```

```
printf(str);
```

```
}
```

请问运行 **Test** 函数会有什么样的结果？

答：程序崩溃。

因为

**GetMemory** 并不能传递动态内存，

**Test**

函数中的 **str** 一直都是 **NULL**。

**strcpy(str, "hello world");**将使程序崩溃。

```
void GetMemory2(char **p, int num)
```

```
{
```

```
*p = (char *)malloc(num);
```

```
}
```

```
void Test(void)
```

```
{
```

```
char *str = NULL;
```

```
GetMemory(&str, 100);
```

```
strcpy(str, "hello");
```

```
printf(str);
```

```
}
```

请问运行 `Test` 函数会有什么样的结果？

答：

(1)

能够输出 `hello`

(2)

内存泄漏

```
char *GetMemory(void)
```

```
{
```

```
char p[] = "hello world";
```

```
return p;
```

```
}
```

```
void Test(void)
```

```
{
```

```
char *str = NULL;
```

```
str = GetMemory();
```

```
printf(str);
```

```
}
```

请问运行 `Test` 函数会有什么样的结果？

答：

可能是乱码。

因为

`GetMemory` 返回的是指向“栈内存”

的指针，该指针的地址不是 `NULL`，但其原

现的内容已经被清除，新内容不可知。

```
void Test(void)
```

```
{
```

```
char *str = (char *) malloc(100);
```

```
strcpy(str, "hello");
```

```
free(str);
```

```
if(str != NULL)
```

```
{
```

```
strcpy(str, "world");
```

```
printf(str);
```

```
}
```

```
}
```

请问运行 `Test` 函数会有什么样的结果？

答：篡改动态内存区的内容，后果难以预

料，非常危险。

因为 `free(str);` 之后，`str` 成为野指针，

`if(str != NULL)` 语句不起作用。

一. 主观判断题（10 分）

1. 软件测试的目的是尽可能多的找出软件的缺陷。（Y）

2. 只要我们做了充分的测试。就能保证软件没有 `BUG`（n）



3. 验收测试是由最终用户来实施的。(Y)

4. 项目立项前测试人员不需要提交任何工件。(n)

5. 单元测试能发现约 80%的软件缺陷。(Y)

6. 代码评审是检查源代码是否达到模块设计的要求。(Y)

7. 负载测试是验证要检验的系统的能力最高能达到什么程度。(n)

8. 测试人员要坚持原则，缺陷未修复完坚决不予通过。(Y)

9. 我们可以人为的使得软件不存在配置问题。(n)

10. 集成测试计划在需求分析阶段末提交 (n)

二. 基础理论题：(35 分)

1. 已建立二张个学生基本信息表，表的结构如下：

Test1

No Name Sex Age Department Place

2002001

张三

男

20

计算机系 北京

2002002

李四

女

20

数学系 山东

2002003

王五

男

21

计算机系 北京

2002004

小红

女

21

数学系

河北

2002005

小李

男

20

数学系

辽宁

2002006

小王

男

22

计算机系 浙江

Test2

No Grade(N

型) Courses

2002001 67

英语

2002002 81

数学

2002003 75

操作系统

2002004 91

网络

2002005 70

数据库

2002006 85 C 语言

1) 用

SQL 语句找出是计算机系并籍贯是北京的所有记录

`select * from test where department='计算机系' and place='北京'`

2) 用

SQL 语句把计算机系的学生的 Department 改为信息学院

`UPDATE test SET department='信息学院' WHERE department='计算机系';`

3) 用 SQL 连接查询找出成绩大于等于 75 分的学生的 No,Nname,Grade,Courses 字段的记录

`select distinct test.name1,test.number,test1.grade,test1.courses from test,test1`

`where test.number=test1.number and test1.grade>75`

2. 在数据库中如何实现并发控制，有哪两种方式？

封锁和封锁协议

3. 简述软件生命周期有那些阶段

软件生命周期——需求分析——软件设计——程序编码——软件测试——运行维护

4. CMM 软件成熟度模型有几个级别，分别是什么。

5 个级别，由低到高分别是：

CMM1

初始级

CMM2

可重复级

CMM3

已定义级

CMM4

已管理级

CMM5

优化级

5. 一个典型的 B/S 结构有哪些组件构成

所谓 B/S 结构，就是只安装维护一个服务器（Server），而客户端采用浏览器（Browse）运行软件，即浏览器/服务器结构。

三. 测试技能题：（30 分）

1. 黑盒测试中设计测试用例一般有哪几种方法  
边界值、等价类划分、穷举法、因果图、场景法

2. Bug 管理和版本控制一般用到哪些工具。

BUG 管理：TD 或 QC 或 bugzilla

版本控制：CVS、SVN、CSS

3. 简述性能测试的一般步骤和目的？

步骤：

步骤一：性能测试与数据收集

步骤二：性能瓶颈分析

步骤三：性能调优解决方案

目的：

λ 最终的目的是最大限度地满足用户需求

评价系统当前性能

λ

预测系统未来性能

λ

λ 寻找瓶颈，优化性能

4. 在 LR 中使用 IP 欺骗的作用是什么？

贴近用户实际使用环境。仿真效果好

有些系统不兼容相同 IP

的访问

5. 说出软件测试过程 V

模型，并说明它存在的缺点，可以画图。

§

测试是开发之后的一个阶段。

§

测试的对象就是程序本身。

§

实际应用中容易导致需求阶段的错误一直到最后系统测试阶段才被发现。

综合题：（15

分）

一个程序读入

3 个整数，a:输出最大值或最小值

A: 最大值：（最小值把“>”替换为“<”，“max”替换为“min”）

```
#include <stdio.h>
```

```
#definr max(x,y) (((x) > (y)) ? (x) : (y))
```

```
int
```

```
main()
```

```
{
```

```
int
```

```
a, b, c, d;
```

```
scanf( "%d,%d,%d" .&a,&b,&c);
d=max(a,max(b,c));
printf( "max=%d\n",d)
}
```

有两个小组对同一个软件进行测试（测试的时间不清楚，软件的规模不清楚），

A 组测试出 50 个 Bug；B 组测试出 55 个 Bug，提交汇总后发现其中有 25 个

是相

同的；我的问题是：请你估算一下这个软件还有多少个 Bug 没被发现？

听一个同事说有次面试的时候主考官给他出了这样一道题，正好在很久以前看到过类似的资料，这里给大家共享出来，看看这种算法合理不。

先说这个问题的答案是 30，怎么算出来的呢？可以按照下面的公式：

可以估计出的软件的缺陷共有： $50 \times 55 / 25 = 110$  个

目前已经发现的有： $50 + 55 - 25 = 80$  个

没有发现的 bug 有： $110 - 80 = 30$  个

这个公式又是怎么得出来的呢，可以看看下面的推导过程：

B——组 A 和组 B 都发现的缺陷数

N1——组

A

发现的缺陷数

N2——组

B

发现的缺陷数

T——软件所有的缺陷数

根据原理：组 A 发现的缺陷数占总缺陷数的比例等于组 A 和组 B 都发现的缺陷

数占组 B 发现的缺陷数的比例，即  $N1/T = B/N2$

上面的公式改变形式即： $T = N1 \times N2 / B$ （软件总 bug 数）

有了上面的公式再计算其它问题就容易了。

## 搜狗软件测试工程师笔试题

全套软件测试面试笔试题（附答案）

一、判断题

1.软件测试的目的是尽可能多的找出软件的缺陷。(Y)

2.Beta 测试是验收测试的一种。(Y)

- 3.验收测试是由最终用户来实施的。(N)
- 4.项目立项前测试人员不需要提交任何工件。(Y)
- 5.单元测试能发现约 80%的软件缺陷。(Y)
- 6.代码评审是检查源代码是否达到模块设计的要求。(N)
- 7.自底向上集成需要测试员编写驱动程序。(Y)
- 8.负载测试是验证要检验的系统的能力最高能达到什么程度。(N)
- 9.测试人员要坚持原则，缺陷未修复完坚决不予通过。(N)
- 10.代码评审员一般由测试员担任。(N)
- 11.我们可以人为的使得软件不存在配置问题。(N)
- 12.集成测试计划在需求分析阶段未提交。(N)

## 二、选折

- 1.软件验收测试的合格通过准则是：(ABCD)
  - A.软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求。
  - B.所有测试项没有残余一级、二级和三级错误。
  - C.立项审批表、需求分析文档、设计文档和编码实现一致。
  - D.验收测试工件齐全。
- 2.软件测试计划评审会需要哪些人员参加?(ABCD)
  - A.项目经理
  - B.SQA 负责人
  - C.配置负责人
  - D.测试组

3.下列关于 alpha 测试的描述中正确的是：(AD)

A.alpha

测试需要用户代表参加

B.alpha

测试不需要用户代表参加

C.alpha

测试是系统测试的一种

D.alpha

测试是验收测试的一种

4.测试设计员的职责有：(BC)

A.制定测试计划

B.设计测试用例

C.设计测试过程、脚本

D.评估测试活动

5.软件实施活动的进入准则是：(ABC)

A.需求工件已经被基线化

B.详细设计工件已经被基线化

C.构架工件已经被基线化

D.项目阶段成果已经被基线化

三、添空

1.软件验收测试包括：正式验收测试，alpha 测试，beta 测试。

2.系统测试的策略有：功能测试，性能测试，可靠性测试，负载测试，易用性测试，强度测试，安全测试，配置测试，安装测试，卸载测试，文档测试，故障恢复测试，界面测试，容量测试，兼容性测试，分布测试，可用性测试，(有的可以合在一起，分开写只要写出 15 就满分哦)

3.设计系统测试计划需要参考的项目文档有：软件测试计划，软件需求工件和迭代计划。

4.对面向过程的系统采用的集成策略有：自顶向下，自底向上两种。

5.(这题出的有问题哦，详细的 5 步骤为~~)通过画因果图来写测试用例的步骤为：

(1)分析软件规格说明描述中，哪些是原因(即输入条件或输入条件的等价类)，哪些是结果(即输出条件)，并给每个原因和结果赋予一个标识符。

(2)分析软件规格说明描述中的语义，找出原因与结果之间，原因与原因之间对应的是什么关系?根据这些关系，画出因果图。

(3)由于语法或环境限制，有些原因与原因之间，原因与结果之间的组合情况不可能出现。为表明这些特殊情况，在因果图上用一些记号标明约束或限制条件。

(4)把因果图转换成判定表。

(5)把判定表的每一列拿出来作为依据，设计测试用例。

四、简答(资料是搜集整理的，感谢前辈的解题)无

1.区别阶段评审的与同行评审

同行评审目的:发现小规模工作产品的错误,只要是找错误;

阶段评审目的:评审模块阶段作品的正确性可行性及完整性



同行评审人数:3-7 人人员必须经过同行评审会议的培训,由 SQA 指导

阶段评审人数:5 人左右评审人必须是专家具有系统评审资格

同行评审内容:内容小一般文档 < 40 页,代码 < 500 行

阶段评审内容:内容多,主要看重点

同行评审时间:一小部分工作产品完成

阶段评审时间:通常是设置在关键路径的时间点上!

## 2.什么是软件测试

为了发现程序中的错误而执行程序的过程

## 3 简述集成测试的过程

系统集成测试主要包括以下过程:

- 1.构建的确认过程。
- 2.补丁的确认过程。
- 3.系统集成测试测试组提交过程。
- 4.测试用例设计过程。
- 5.测试代码编写过程。
6. Bug 的报告过程。
- 7.每周/每两周的构建过程。
- 8.点对点的测试过程。
- 9.组内培训过程。

## 4 怎么做好文档测试

仔细阅读, 跟随每个步骤, 检查每个图形, 尝试每个示例。P142

检查文档的编写是否满足文档编写的目的

内容是否齐全，正确

内容是否完善

标记是否正确

## 5 白盒测试有几种方法

总体上分为静态方法和动态方法两大类。

静态：关键功能是检查软件的表示和描述是否一致,没有冲突或者没有歧义

动态：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。

## 6 系统测试计划是否需要同行审批，为什么

需要，系统测试计划属于项目阶段性关键文档，因此需要评审。

## 7 Alpha 测试与 beta 的区别

Alpha 测试在系统开发接近完成时对应用系统的测试;测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成，不能由程序或测试员完成。由一个或多个用户在开发环境下进行测试。

Beta 测试当开发和测试根本完成时所做的测试，最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成，不能由程序员或测试员完成。由一个或多个用户在用户实际环境下进行而是。

8 比较负载测试,容量测试和强度测试的区别。性能测试从使用方便的角度来说,可以分为以下六种：负载测试、压力测试、容量测试、配置测试、基准测试和并发测试。

负载测试：在一定的工作负荷下，系统的负荷及响应时间。在一定的软件、硬件及网络环境下，运行一种或多种业务，在不同虚拟用户数量下，测试服务器指标

是否在用户的要求范围内，以此确定系统所能承受的最大用户数，系统响应时间及资源利用率等。

压力测试：在一定的软件、硬件及网络环境下，模拟大量的虚拟用户向服务器产生负载，使服务器的资源处于极限状态下并长时间连续运行，以测试服务器在高负载情况下是否能够稳定工作。

强度测试：在一定的负荷条件下，在较长时间跨度内的系统连续运行给系统性能所造成的影响。

容量测试：容量测试目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值(如最大并发用户数、数据库记录数等)，系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。在一定的软件、硬件及网络下，在数据库中构造不同数量级的数据记录，在一定虚拟用户下运行一种或多种业务，以获取不同数量级别的服务器性能指标，以确定数据库的最大容量及最佳容量。

配置测试：在一定软件、硬件及网络环境下，运行一种或多种业务，在一定数量的虚拟用户下，获得不能配置的性能指标。

基准测试：在一定软件、硬件及网络环境下，模拟一定数量的虚拟用户运行一种或多种业务，将测试结果作为基线数据，在系统调优或系统评测的过程中，通过运行相同业务场景比较测试结果，确定调优的结果是否达到预期效果或者为系统

的选择提供决策数据。基准测试一般基于配置测试，通过配置测试得到数据，并将这个这个数据作为基准来比较每次调优后性能是否有所改善。

并发测试：通过模拟多个用户并发访问同一个应用、存储过程、数据记录或其他并发操作，测试是否存在死锁、数据错误等故障。

## 9 测试结束的标准是什么？

用例全部测试。

覆盖率达到标准。

缺陷率达到标准。

其他指标达到质量标准

## 10 描述软件测试活动的生命周期？

测试周期分为计划、设计、实现、执行、总结。其中：

计划：对整个测试周期中所有活动进行规划，估计工作量、风险，安排人力物力资源，安排进度等;从宏观上规划整个测试活动。测试包括文档目的，项目背景介绍，输入文档，系统运行环境(网络拓扑图、软硬件配置)、测试内容、资源安排(角色和职责，测试工具)、进度安排、交付件及风险等。

设计：完成测试方案，从技术层面上对测试进行规划;其目的是为整个测试阶段的执行内容及侧罗进行详细说明。测试方案包括文档目的，测试策略等。

实现：进行测试用例和测试规程设计;

执行：根据前期完成的计划、方案、用例、规程等文档，执行测试用例。

总结：记录测试结果，进行测试分析，完成测试报告。

## 11 软件的缺陷等级应如何划分？

A 类—严重错误，包括以下各种错误： 1.由于程序所引起的死机,非法退出 2.

死循环 3.数据库发生死锁 4.因错误操作导致的程序中断 5.功能错误

6.与数据

库连接错误 7.数据通讯错误

B 类—较严重错误，包括以下几种错误： 1.程序错误 2.程序接口错误 3.数据库

的表、业务规则、缺省值未加完整性等约束条件

C 类—一般性错误，包括以下几种错误： 1.操作界面错误(包括数据窗口内列名定义、含义是否一致) 2.打印内容、格式错误 3.简单的输入限制未放在前台进行控制 4.删除操作未给出提示 5.数据库表中有过多的空字段

D 类—较小错误，包括以下几种错误： 1.界面不规范 2.辅助说明描述不清楚 3.输入输出不规范 4.长操作未给用户提示 5.提示窗口文字未采用行业术语 6.可输入区域和只读区域没有明显的区分标志

E 类—测试建议

## 腾讯近三年 78 道软件测试面试题(包含答案)

1、什么是兼容性测试?兼容性测试侧重哪些方面?

参考答案:

兼容测试主要是检查软件在不同的硬件平台、软件平台上是否可以正常的运行，即是通常说的软件的可移植性。

兼容的类型，如果细分的话，有平台的兼容，网络兼容，数据库兼容，以及数据

格式的兼容。

兼容测试的重点是，对兼容环境的分析。通常，是在运行软件的环境不是很确定的情况下，才需要做兼容。根据软件运行的需要，或者根据需求文档，一般都能够得出用户会在什么环境下使用该软件，把这些环境整理成表单，就得出做兼容测试的兼容环境了。

兼容和配置测试的区别在于，做配置测试通常不是 Clean OS 下做测试，而兼容

测试多是在 Clean OS 的环境下做的。

2、我现在有个程序，发现在 Windows 上运行得很慢，怎么判别是程序存在问题还是软硬件系统存在问题？

参考答案：

- 1、检查系统是否有中毒的特征;
- 2、检查软件/硬件的配置是否符合软件的推荐标准;
- 3、确认当前的系统是否是独立，即没有对外提供什么消耗 CPU 资源的服务;
- 4、如果是 C/S 或者 B/S 结构的软件，需要检查是不是因为与服务器的连接有问题，或者访问有问题造成的;
- 5、在系统没有任何负载的情况下，查看性能监视器，确认应用程序对 CPU/内存的访问情况。

3、测试的策略有哪些？

参考答案：

黑盒/白盒，静态/动态，手工/自动，冒烟测试，回归测试，公测(Beta 测试的

策

略)

4、正交表测试用例设计方法的特点是什么？

参考答案：

用最少的实验覆盖最多的操作，测试用例设计很少，效率高，但是很复杂；

对于基本的验证功能，以及二次集成引起的缺陷，一般都能找出来；但是更深的缺陷，更复杂的缺陷，还是无能为力的；

具体的环境下，正交表一般都很难做的。大多数，只在系统测试的时候使用此方法。

5、描述使用 bugzilla 缺陷管理工具对软件缺陷(BUG)跟踪的管理的流程？

参考答案：

就是 Bugzilla 的状态转换图。

6、你觉得 bugzilla 在使用的过程中，有什么问题？

参考答案：

界面不稳定；

根据需要配置它的不同的部分，过程很烦琐。

流程控制上，安全性不好界定，很容易对他人的 Bug 进行误操作；

没有综合的评分指标，不好确认修复的优先级别。

7、描述测试用例设计的完整过程？



参考答案：

需求分析

+ 需求变更的维护工作;

根据需求

得出测试需求;

设计测试方案，评审测试方案;

方案评审通过后，设计测试用例，再对测试用例进行评审;

8、单元测试的策略有哪些？

参考答案：

逻辑覆盖、循环覆盖、同行评审、桌前检查、代码走查、代码评审、景泰数据流  
分析

9、LoadRunner 分哪三部分？

参考答案：

用户动作设计;

场景设计;

测试数据分析;

10、LoadRunner 进行测试的流程？

参考答案：

1、

测试测试

2、

创建虚拟用户脚本

3、

创建运行场景

4、

运行测试脚本

5、

监视场景

6、

分析测试的结果

以上，最好是结合一个案例，根据以上流程来介绍。

什么是并发?在 lordrunner 中，如何进行并发的测试?集合点失败了会怎么样?

参考答案：

在同一时间点，支持多个不同的操作。

LoadRunner 中提供 IP 伪装，集合点，配合虚拟用户的设计，以及在多台电脑

上设置，可以比较好的模拟真实的并发。

集合点，即是多个用户在某个时刻，某个特定的环境下同时进行虚拟用户的操作的。集合点失败，则集合点的才操作就会取消，测试就不能进行。

12、使用 QTP 做功能测试，录制脚本的时候，要验证多个用户的登录情况/查询

情况，如何操作?

参考答案：

分析用户登录的基本情况，得出一组数据，通过性测试/失败性测试的都有(根据

TC 来设计这些数据)，然后录制登录的脚本，将关键的数据参数化，修改脚本，对代码进行加强，调试脚本。

13、QTP 中的 Action 有什么作用?有几种?

参考答案：

Action 的作用

n

用 Action 可以对步骤集进行分组

n

步骤重组，然后被整体调用

n

拥有自己的 sheet

n

组合有相同需求的步骤，整体操作

n

具有独立的对象仓库

Action 的种类

n

可复用 Action

n

不可复用 Action

n

外部 Action

14、TestDirector 有些什么功能，如何对软件测试过程进行管理？

参考答案：

需求管理

n

定义测试范围

n

定义需求树

n

描述需求树的功能点

测试计划

n

定义测试目标和测试策略。

n

分解应用程序，建立测试计划树。

n

确定每个功能点的测试方法。

n

将每个功能点连接到需求上，使测试计划覆盖全部的测试需求。

n

描述手工测试的测试步骤

n

指明需要进行自动测试的功能点

测试执行

n

定义测试集合。

n

为每个测试人员制定测试任务和测试日程安排。

n

运行自动测试。

缺陷跟踪

n

记录缺陷

n

查看新增缺陷，并确定哪些是需要修正的

n

相关技术人员修改缺陷

n

回归测试

n

分析缺陷统计图表，分析应用程序的开发质量。

15、你所熟悉的软件测试类型都有哪些?请试着分别比较这些不同的测试类型的区别与联系(如功能测试、性能测试.....)?

参考答案：Compatibility Testing(兼容性测试)，也称“Configuration testing(配置测试)”，测试软件是否和系统的其它与之交互的元素之间兼容，如：

浏览器、操作系统、硬件等。验证测试对象在不同的软件和硬件配置中的运行情况。

Functional testing (功能测试), 也称为 behavioral testing(行为测试), 根据产品特征、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。

Performance testing(性能测试), 评价一个产品或组件与性能需求是否符合的测试。包括负载测试、强度测试、数据库容量测试、基准测试等类型。

16、软件缺陷(或者叫 Bug)记录都包含了哪些内容?如何提交高质量的软件缺陷(Bug)记录?

参考答案: 5C 标准

17、Beta 测试与 Alpha 测试有什么区别?

参考答案: Beta testing( $\beta$ 测试),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

Alpha testing ( $\alpha$ 测试),是由一个用户在开发环境下进行的测试, 也可以是公司内部的用户在模拟实际操作环境下进行的受控测试

18、软件的评审一般由哪些人参加?其目的是什么?

参考答案:

在正式的会议上将软件项目的成果(包括各阶段的文档、产生的代码等)提交给用户、客户或有关部门人员对软件产品进行评审和批准。其目的是找出可能影响软

件产品质量、开发过程、维护工作的适用性和环境方面的设计缺陷，并采取补救

措施，以及找出在性能、安全性和经济方面的可能的改进。

人员：用户、客户或有关部门开发人员，测试人员，需求分析师都可以，就看处于评审那个阶段

19、测试活动中，如果发现需求文档不完善或者不准确，怎么处理？

参考答案：

测试需求分析发现需求文档不完善或者不准确，应该立即和相关人员进行协调交流。

20、阶段评审与项目评审有什么区别？

参考答案：

阶段评审对项目各阶段评审：对阶段成果和工作

项目评审对项目总体评审：对工作和产品

21、阐述工作版本的定义？

参考答案：

构造号： BUILD

22、什么是桩模块？什么是驱动模块？

参考答案：

桩模块：被测模块调用模块

驱动模块调用被测模块

23、什么是扇入？什么是扇出？

参考答案：



扇入：被调次数，扇出：调其它模块数目

24、你认为做好测试计划工作的关键是什么？

参考答案：

软件测试计划就是在软件测试工作正式实施之前明确测试的对象，并且通过对资源、时间、风险、测试范围和预算等方面的综合分析和规划，保证有效的实施软件测试；

做好测试计划工作的关键：目的，管理，规范

1. 明确测试的目标，增强测试计划的实用性

编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

2.坚持“5W”规则，明确内容与过程

“5W”规则指的是“What(做什么)”、“Why(为什么做)”、“When(何时做)”、

“Where(在哪里)”、“How(如何做)”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的(Why)，明确测试的范围和内容(What)，确定测试的开始和结束日期(When)，指出测试的方法和工具(How)，给出测试文档和软件的存放位置(Where)。

3.采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

#### 4. 分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

#### 25、你认为做好测试用例工作的关键是什么？

参考答案：

需求和设计文档的理解程度，对系统的熟悉程度

#### 26、简述一下缺陷的生命周期？

参考答案：提交->确认->分配->修复->验证->关闭

#### 27、软件的安全性应从哪几个方面去测试？

参考答案：(1)用户认证机制：如数据证书、智能卡、双重认证、安全电子交易协议

(2)加密机制

(3)安全防护策略：如安全日志、入侵检测、隔离防护、漏洞扫描

(4)数据备份与恢复手段：存储设备、存储优化、存储保护、存储管理

(5)防病毒系统

28、软件配置管理工作开展的情况和认识？

参考答案：

软件配置管理贯穿于软件开发、测试活动的始终，覆盖了开发、测试活动的各个环节，它的重要作用之一就是要全面的管理保存各个配置项，监控各配置项的状态，并向项目经理及相关的人员报告，从而实现对软件过程的控制。

软件测试配置管理包括 4 个最基本的活动：

配置项标识

配置项控制

配置项状态报告

配置审计

软件配置管理通常借助工具来辅助，主要有 MS SourceSafe、Rational ClearCase 等

29、你觉得软件测试通过的标准应该是什么样的？

参考答案：

缺陷密度值达到客户的要求

30、引入测试管理的含义？

参考答案：风险分析，进度控制、角色分配、质量控制

31、一套完整的测试应该由哪些阶段组成？

参考答案：测试计划、测试设计与开发、测试实施、测试评审与测试结论

32、单元测试的主要内容？

参考答案：

模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试

33、集成测试也叫组装测试或者联合测试，请简述集成测试的主要内容？

参考答案：

(1)在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；

(2)一个模块的功能是否会对另一个模块的功能产生不利的影响；

(3)各个子功能组合起来，能否达到预期要求的父功能；

(4)全局数据结构是否有问题；

(5)单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

34、简述集成测试与系统测试关系？

参考答案：

(1)集成测试的主要依据概要设计说明书，系统测试的主要依据是需求设计说明书；

(2)集成测试是系统模块的测试，系统测试是对整个系统的测试，包括相关的软硬件平台、网络以及相关外设的测试。

35、软件测试的文档测试应当贯穿于软件生命周期的全过程，其中用户文档是文档测试的重点。那么软件系统的用户文档包括哪些？

参考答案：

用户手册

安装和设置指导

联机帮助

指南、向导

样例、示例和模板

授权/注册登记表

最终用户许可协议

36、软件系统中除用户文档之外，文档测试还应该关注哪些文档？

参考答案：

开发文档

软件需求说明书

数据库设计说明书

概要设计说明书

详细设计说明书

可行性研究报告

管理文档

项目开发计划

测试计划

测试报告

开发进度月报

开发总结报告

37、简述软件系统中用户文档的测试要点？

参考答案：(1)读者群。文档面向的读者定位要明确。对于初级用户、中级用户以及高级用

户应该有不同定位

(2)术语。文档中用到的术语要适用与定位的读者群，用法一致，标准定义与业界规范相吻合。

(3)正确性。测试中需检查所有信息是否真实正确，查找由于过期产品说明书和

销售人员夸大事实而导致的错误。检查所有的目录、索引和章节引用是否已更新，尝试链接是否准确，产品支持电话、地址和邮政编码是否正确。

(4)完整性。对照软件界面检查是否有重要的分支没有描述到，甚至是否有整个大模块没有描述到。

(5)一致性。按照文档描述的操作执行后，检查软件返回的结果是否与文档描述的相同。

(6)易用性。对关键步骤以粗体或背景色给用户以提示，合理的页面布局、适量的图表都可以给用户更高的易用性。需要注意的是文档要有助于用户排除错误。不但描述正确操作，也要描述错误处理办法。文档对于用户看到的错误信息应当有更详细的文档解释。

(7)图表与界面截图。检查所有图表与界面截图是否与发行版本相同。

(8)样例与示例。像用户一样载入和使用样例。如果是一段程序，就输入数据并执行它。以每一个模块制作文件，确认它们的正确性。

(9)语言。不出现错别字，不要出现有二义性的说法。特别要注意的是屏幕截图或绘制图形中的文字。

(10)印刷与包装。检查印刷质量;手册厚度与开本是否合适;包装盒的大小是否合适;有没有零碎易丢失的小部件等等。

38、单元测试主要内容是什么？

参考答案：

单元测试大多数由开发人员来完成，测试人员技术背景较好或者开发系统软件时可能会安排测试人员进行单元测试，大多数进行的单元测试都是开发人员调试程序或者开发组系统联合调试的过程。讨论这个问题主要是扩充一下读者的视野。

单元测试一般包括五个方面的测试：

(1)模块接口测试：模块接口测试是单元测试的基础。只有在数据能正确流入、流出模块的前提下，其他测试才有意义。模块接口测试也是集成测试的重点，这里进行的测试主要是为后面打好基础。测试接口正确与否应该考虑下列因素：

- 输入的实际参数与形式参数的个数是否相同;
- 输入的实际参数与形式参数的属性是否匹配;
- 输入的实际参数与形式参数的量纲是否一致;
- 调用其他模块时所给实际参数的个数是否与被调模块的形参个数相同;
- 调用其他模块时所给实际参数的属性是否与被调模块的形参属性匹配;
- 调用其他模块时所给实际参数的量纲是否与被调模块的形参量纲一致;
- 调用预定义函数时所用参数的个数、属性和次序是否正确;
- 是否存在与当前入口点无关的参数引用;
- 是否修改了只读型参数;
- 对全程变量的定义各模块是否一致;
- 是否把某些约束作为参数传递。

如果模块功能包括外部输入输出，还应该考虑下列因素：

- 文件属性是否正确;
- OPEN/CLOSE 语句是否正确;
- 格式说明与输入输出语句是否匹配;
- 缓冲区大小与记录长度是否匹配;
- 文件使用前是否已经打开;
- 是否处理了文件尾;



- 是否处理了输入/输出错误;
- 输出信息中是否有文字性错误。
- 局部数据结构测试;
- 边界条件测试;
- 模块中所有独立执行通路测试;

(2)局部数据结构测试：检查局部数据结构是为了保证临时存储在模块内的数据在程序执行过程中完整、正确，局部功能是整个功能运行的基础。重点是一些函数是否正确执行，内部是否运行正确。局部数据结构往往是错误的根源，应仔细设计测试用例，力求发现下面几类错误：

- 不合适或不相容的类型说明;
- 变量无初值;
- 变量初始化或省缺值有错;
- 不正确的变量名(拼错或不正确地截断);
- 出现上溢、下溢和地址异常。

(3)边界条件测试：边界条件测试是单元测试中最重要的一项任务。众所周知，软件经常在边界上失效，采用边界值分析技术，针对边界值及其左、右设计测试用例，很有可能发现新的错误。边界条件测试是一项基础测试，也是后面系统测试中的功能测试的重点，边界测试执行的较好，可以大大提高程序健壮性。

(4)模块中所有独立路径测试：在模块中应对每一条独立执行路径进行测试，单元测试的基本任务是保证模块中每条语句至少执行一次。测试目的主要是为了发现因错误计算、不正确的比较和不适当的控制流造成的错误。具体做法就是程序员逐条调试语句。常见的错误包括：

- 误解或用错了算符优先级;
- 混合类型运算;
- 变量初值错;
- 精度不够;
- 表达式符号错。

比较判断与控制流常常紧密相关，测试时注意下列错误：

- 不同数据类型的对象之间进行比较;
- 错误地使用逻辑运算符或优先级;
- 因计算机表示的局限性，期望理论上相等而实际上不相等的两个量相等;
- 比较运算或变量出错;
- 循环终止条件或不可能出现;
- 迭代发散时不能退出;
- 错误地修改了循环变量。

模块的各条错误处理通路测试：程序在遇到异常情况时不应该退出，好的程序应能预见各种出错条件，并预设各种出错处理通路。如果用户不按照正常操作，程序就退出或者停止工作，实际上也是一种缺陷，因此单元测试要测试各种错误处理路径。一般这种测试着重检查下列问题：

- 输出的出错信息难以理解;
- 记录的错误与实际遇到的错误不相符;
- 在程序自定义的出错处理段运行之前，系统已介入;
- 异常处理不当;
- 错误陈述中未能提供足够的定位出错信息。

### 39、如何理解强度测试？

参考答案：

强度测试是为了确定系统在最差工作环境的工作能力,也可能是用于验证在标准工作压力下的各种资源的最下限指标。

它和压力测试的目标是不同的,压力测试是在标准工作环境下,不断增加系统负荷,

最终测试出该系统能力达到的最大负荷(稳定和峰值),而强度测试则是在非标准工作环境下,甚至不断人为降低系统工作环境所需要的资源,如网络带宽,系统内存,数据锁等等,以测试系统在资源不足的情况下的工作状态,通过强度测试,可以确定本系统正常工作的最差环境.

强度测试和压力测试的测试指标相近,大多都是与时间相关的指标,如并发量(吞吐量),延迟(最大\最小\平均)以及顺序指标等

强度测试需要对系统的结构熟悉,针对系统的特征设计强度测试的方法

### 40、如何理解压力、负载、性能测试测试？

参考答案： 性能测试是一个较大的范围，实际上性能测试本身包含了性能、强度、压力、负

载等多方面的测试内容。

压力测试是对服务器的稳定性以及负载能力等方面的测试，是一种很平常的测试。增大访问系统的用户数量、或者几个用户进行大数据量操作都是压力测试。

而负载测试是压力相对较大的测试,主要是测试系统在一种或者集中极限条件下的相应能力，是性能测试的重要部分。100 个用户对系统进行连续半个小时的访

访

问可以看作压力测试，那么连续访问 8 个小时就可以认为负载测试，1000 个用

户连续访问系统 1 个小时也可以看作是负载测试。

实际上压力测试和负载测试没有明显的区分。测试人员应该站在关注整体性能的高度上来对系统进行测试。

#### 41、什么是系统瓶颈？

参考答案：

瓶颈主要是指整个软硬件构成的软件系统某一方面或者几个方面能力不能满足用户的特定业务要求，“特定”是指瓶颈会在某些条件下会出现，因为毕竟大多数系统在投入前。

严格的从技术角度讲，所有的系统都会有瓶颈，因为大多数系统的资源配置不是协调的，例如 CPU 使用率刚好达到 100%时，内存也正好耗尽的系统不是很多

见。因此我们讨论系统瓶颈要从应用的角度讨论：关键是看系统能否满足用户需求。在用户极限使用系统的情况下，系统的响应仍然正常，我们可以认为改系统没有瓶颈或者瓶颈不会影响用户工作。

因此我们测试系统瓶颈主要是实现下面两个目的：-发现“表面”的瓶颈。主要是模拟用户的操作，找出用户极限使用系统时的瓶颈，然后解决瓶颈，这是性能测试的基本目标。

-发现潜在的瓶颈并解决，保证系统的长期稳定性。主要是考虑用户在将来扩展系统或者业务发生变化时，系统能够适应变化。满足用户目前需求的系统不是最好的，我们设计系统的目标是在保证系统整个软件生命周期能够不断适应用户的

变化，或者通过简单扩展系统就可以适应新的变化。

#### 42、文档测试主要包含什么内容？

参考答案：

在国内软件开发管理中，文档管理几乎是最弱的一项，因而在测试工作中特别容易忽略文档测试也就不足为奇了。要想给用户提供完整的产品，文档测试是必不可少的。文档测试一般注重下面几个方面：

文档的完整性：主要是测试文档内容的全面性与完整性，从总体上把握文档的质量。例如用户手册应该包括软件的所有功能模块。

描述与软件实际情况的一致性：主要测试软件文档与软件实际的一致程度。例如用户手册基本完整后，我们还要注意用户手册与实际功能描述是否一致。因为文档往往跟不上软件版本的更新速度。

易理解性：主要是检查文档对关键、重要的操作有无图文说明，文字、图表是否易于理解。对于关键、重要的操作仅仅只有文字说明肯定是不够的，应该附有图表使说明更为直观和明了。

文档中提供操作的实例：这项检查内容主要针对用户手册。对主要功能和关键操作提供的实例是否丰富，提供的实例描述是否详细。只有简单的图文说明，而无实例的用户手册看起来就像是软件界面的简单拷贝，对于用户来说，实际上没有什么帮助。

印刷与包装质量：主要是检查软件文档的商品化程度。有些用户手册是简单打印、装订而成，过于粗糙，不易于用户保存。优秀的文档例如用户手册和技术白皮书，应提供商品化包装，并且印刷精美。

#### 43、功能测试用例需要详细到什么程度才是合格的？

参考答案：

这个问题也是测试工程师经常问的问题。有人主张测试用例详细到每个步骤执行什么都要写出来，目的是即使一个不了解系统的新手都可以按照测试用例来执行工作。主张这类写法的人还可以举出例子：欧美、日本等软件外包文档都是这样做的。

另外一种观点就是主张写的粗些，类似于编写测试大纲。主张这种观点的人是因为软件开发需求管理不规范，变动十分频繁，因而不能按照欧美的高标准来编写测试用例。这样的测试用例容易维护，可以让测试执行人员有更大的发挥空间。实际上，软件测试用例的详细程度首先要以覆盖到测试点为基本要求。举个例子：

“用户登陆系统”的测试用例可以不写出具体的执行数据，但是至少要写出五种以上情况()，如果只用一句话覆盖了这个功能是不合格的测试用例。覆盖功能点不是指列出功能点，而是要写出功能点的各个方面(如果组合情况较多时可以采用等价划分)。

另一个影响测试用例的就是组织的开发能力和测试对象特点。如果开发力量比较落后，编写较详细的测试用例是不现实的，因为根本没有那么大的资源投入，当然这种情况很随着团队的发展而逐渐有所改善。测试对象特点重点是指测试对象在进度、成本等方面的要求，如果进度较紧张的情况下，是根本没有时间写出高质量的测试用例的，甚至有些时候测试工作只是一种辅助工作，因而不编写测试用例。

因此，测试用例的编写要根据测试对象特点、团队的执行能力等各个方面综合起

来决定编写策略。最后要注意的是测试人员一定不能抱怨，力争在不断提高测试

用例编写水平的同时，不断地提高自身能力。

#### 44、配置和兼容性测试的区别是什么？

参考答案：

配置测试的目的是保证软件在其相关的硬件上能够正常运行，而兼容性测试主要是测试软件能否与不同的软件正确协作。

配置测试的核心内容就是使用各种硬件来测试软件的运行情况，一般包括：

- (1)软件在不同的主机上的运行情况，例如 Dell 和 Apple;
- (2)软件在不同的组件上的运行情况，例如开发的拨号程序要测试在不同厂商生产的 Modem 上的运行情况;
- (3)不同的外设;
- (4)不同的接口;
- (5)不同的可选项，例如不同的内存大小;

兼容性测试的核心内容：

- (1)测试软件是否能在不同的操作系统平台上兼容;
- (2)测试软件是否能在同一操作系统平台的不同版本上兼容;
- (3)软件本身能否向前或者向后兼容;
- (4)测试软件能否与其它相关的软件兼容;
- (5)数据兼容性测试，主要是指数据能否共享;

配置和兼容性测试通称对开发系统类软件比较重要，例如驱动程序、操作系统、数据库管理系统等。具体进行时仍然按照测试用例来执行。

#### 45、软件文档测试主要包含什么？

参考答案：



随着软件文档系统日益庞大，文档测试已经成为软件测试的重要内容。文档测试

对象主要如下：

- 包装文字和图形;
- 市场宣传材料、广告以及其它插页;
- 授权、注册登记表;
- 最终用户许可协议;
- 安装和设置向导;
- 用户手册;
- 联机帮助;
- 样例、示范例子和模板;
- .....

文档测试的目的是提高易用性和可靠性，降低支持费用，因为用户通过文档就可以自己解决问题。因文档测试的检查内容主要如下：

- 读者对象——主要是文档的内容是否能让该级别的读者理解;
- 术语——主要是检查术语是否适合读者;
- 内容和主题——检查主题是否合适、是否丢失、格式是否规范等;
- 图标和屏幕抓图——检查图表的准确度和精确度;
- 样例和示例——是否与软件功能一致;
- 拼写和语法;
- 文档的关联性——是否与其它相关文档的内容一致，例如与广告信息是否一致;

文档测试是相当重要的一项测试工作，不但要给予充分的重视，更要认真的完成，象做功能测试一样来对待文档测试。

46、没有产品说明书和需求文档地情况下能够进行黑盒测试吗？

参考答案：

这个问题是国内测试工程师经常遇到的问题，根源就是国内软件开发文档管理不规范，对变更的管理方法就更不合理了。实际上没有任何文档的时候，测试人员是能够进行黑盒测试的，这种测试方式我们可以称之为探索测试，具体做法就是测试工程师根据自己的专业技能、领域知识等不断的深入了解测试对象、理解软件功能，进而发现缺陷。

在这种做法基本上把软件当成了产品说明书，测试过程中要和开发人员不断的进行交流。尤其在作项目的时候，进度压力比较大，可以作为加急测试方案。最大的风险是不知道有些特性是否被遗漏。

47、测试中的“杀虫剂怪事”是指什么？

参考答案：

“杀虫剂怪事”一词由 BorisBeizer 在其编著的《软件测试技术》第二版中提出。

用于描述测试人员对同一测试对象进行的测试次数越多，发现的缺陷就会越来越少的现象。就像老用一种农药，害虫就会有免疫力，农药发挥不了效力。这种现象的根本原因就是测试人员对测试软件过于熟悉，形成思维定势。

为了克服这种现象，测试人员需要不断编写新的测试程序或者测试用例，对程序的不同部分进行测试，以发现更多的缺陷。也可以引用新人来测试软件，刚刚进来的新手往往能发现一些意想不到的问题。

48、在配置测试中，如何判断发现的缺陷是普通问题还是特定的配置问题？

参考答案：

在进行配置测试时，测试工程师仍然会发现一些普通的缺陷，也就是与配置环境无关的缺陷。因此判断新发现的问题，需要在不同的配置中重新执行发现软件缺陷的步骤，如果软件缺陷不出现了，就可能是配置缺陷；如果在所有的配置中都出现，就可能是普通缺陷。

需要注意的是，配置问题可以在一大类配置中出现。例如，拨号程序可能在所有的外置 Modem 中都存在问题，而内置的 Modem 不会有任何问题。

49、为什么尽量不要让时间有富裕的员工去做一些测试？

参考答案：

表面上看这体现了管理的效率和灵活性，但实际上也体现了管理者对测试的轻视。测试和测试的人有很大关系。测试工作人员应该是勤奋并富有耐心，善于学习、思考和发现问题，细心有条理，总结问题，如果具备这样的优点，做其它工作同样也会很出色，因此这里还有一个要求，就是要喜欢测试这项工作。如果他是专职的，那么肯定更有经验和信心。国内的小伙子好象都喜欢做程序员，两者工作性质不同，待遇不同，地位不同，对自我实现的价值认识也不同，这是行业的一个需要改善的问题。如果只是为了完成任务而完成任务，或者发现了几个问题就觉得满意了，这在任何其它工作中都是不行的。

50、完全测试程序是可能的吗？

参考答案：

软件测试初学者可能认为拿到软件后需要进行完全测试，找到全部的软件缺陷，使软件“零缺陷”发布。实际上完全测试是不可能的。主要有以下一个原因：

- 完全测试比较耗时，时间上不允许；
- 完全测试通常意味着较多资源投入，这在现实中往往是行不通的；

- 输入量太大，不能一一进行测试;
  - 输出结果太多，只能分类进行验证;
  - 软件实现途径太多;
  - 软件产品说明书没有客观标准，从不同的角度看，软件缺陷的标准不同;
- 因此测试的程度要根据实际情况确定。

51、软件测试的风险主要体现在哪里?

参考答案:

我们没有对软件进行完全测试，实际就是选择了风险，因为缺陷极有可能存在没有进行测试的部分。举个例子，程序员为了方便，在调试程序时会弹出一些提示信息框，而这些提示只在某种条件下会弹出，碰巧程序发布前这些代码中的一些没有被注释掉。在测试时测试工程师又没有对其进行测试。如果客户碰到它，这将是代价昂贵的缺陷，因为交付后才被客户发现。

因此，我们要尽可能的选择最合适的测试量，把风险降低到最小。

52、发现的缺陷越多，说明软件缺陷越多吗?

参考答案： 这是一个比较常见的现象。测试工程师在没有找到缺陷前会绞尽脑汁的思考，但

是找到一个后，会接二连三的发现很多缺陷，颇有个人成就感。其中的原因主要如下：

- 代码复用、拷贝代码导致程序员容易犯相同的错误。类的继承导致所有的子类会包含基类的错误，反复拷贝同一代码意味可能也复制了缺陷。
- 程序员比较劳累是可以导致某些连续编写的功能缺陷较多。程序员加班是一种

司空见惯的现象，因此体力不支时容易编写一些缺陷较多的程序。而这些连续潜伏缺陷恰恰是测试工程师大显身手的地方。

“缺陷一个连着一个”不是一个客观规律，只是一个常见的现象。如果软件编写的比较好，这种现象就不常见了。测试人员只要严肃认真的测试程序就可以了。

53、所有的软件缺陷都能修复吗？所有的软件缺陷都要修复吗？

参考答案：

从技术上讲，所有的软件缺陷都是能够修复的，但是没有必要修复所有的软件缺陷。测试人员要做的是能够正确判断什么时候不能追求软件的完美。对于整个项目团队，要做的是对每一个软件缺陷进行取舍，根据风险决定那些缺陷要修复。发生这种现象的主要原因如下：

- 没有足够的时间资源。在任何一个项目中，通常情况下开发人员和测试人员都是不够用的，而且在项目中没有预算足够的回归测试时间，再加上修改缺陷可能引入新的缺陷，因此在交付期限的强大压力下，必须放弃某些缺陷的修改。

- 有些缺陷只是特殊情况下出现，这种缺陷处于商业利益考虑，可以在以后升级中进行修复。

- 不是缺陷的缺陷。我们经常会碰到某些功能方面的问题被当成缺陷来处理，这类问题可以以后有时间时考虑再处理。

最后要说的是，缺陷是否修改要由软件测试人员、项目经理、程序员共同讨论来决定是否修复，不同角色的人员从不同的角度来思考，以做出正确的决定。

54、软件测试人员就是 QA 吗？

参考答案：

软件测试人员的职责是尽可能早的找出软件缺陷，确保得以修复。而质量保证人

员(QA)主要职责是创建或者制定标准和方法，提高促进软件开发能力和减少软件缺陷。测试人员的主要工作是测试，质量保证人员日常工作重要内容是检查与评审，测试工作也是测试保证人员的工作对象。

软件测试和质量是相辅相成的关系，都是为了提高软件质量而工作。

55、如何减少测试人员跳槽带来的损失？

参考答案：

在 IT 行业里跳槽已经是一种司空见惯的现象，而且跳槽无论给公司还是给个人都会带来一定的损失。测试队伍也无疑会面临跳槽的威胁，作为测试经理管理者，只有从日常工作中开始做起，最能最大限度的减少损失。建议我们从以下两个方面做起：

-加强部门内员工之间的互相学习，互相学习是建立学习型组织的基本要求，是知识互相转移的过程。在此基础上，可以把个人拥有的技术以知识的形式沉积下来，也就完成了隐性知识到显性知识的转化。

-通常情况下，企业能为员工提供足够大的发展空间时，如果不是待遇特别低，员工都不会主动离开企业。因此我们要想留住员工，管理者就应该把员工的个人成长和企业发展联系起来，为员工设定合理发展规划并付诸实现。不过这项要求做起来比较，要有比较好的企业文化为依托。

56、测试产品与测试项目的区别是什么？

参考答案：

习惯上把开发完成后进行商业化、几乎不进行代码修改就可以售给用户使用的软件成为软件产品，也就是可以买“卖拷贝”的软件，例如 Windows2000。而通

常把针对一个或者几个特定的用户而开发的软件成为软件项目，软件项目是一种个性化的产品，可以是按照用户要求全部重新开发，也可以修改已有的软件产品来满足特定的用户需求。项目和产品的不同特点，决定我们测试产品和测试项目仍然会有很多不同的地方：

- 质量要求不同。通常产品的质量要高一些，修复发布后产品的缺陷成本较高，甚至会带来很多负面的影响。而做项目通常面向某一用户，虽然质量越高越好，但是一般只要满足用户要求就可以了。

- 测试资源投入多少不同。做软件产品通常是研发中心来开发，进度压力要小些。同时由于质量要求高，因此会投入较多的人力、物力资源。

- 项目最后要和用户共同验收测试，这是产品测试不具有的特点。

此外，测试产品与测试项目在缺陷管理方面、测试策略制定都会有很大不同，测试管理者应该结合具体的环境，恰如其分的完成工作。

57、和用户共同测试(UAT 测试)的注意点有哪些？

参考答案： 软件产品在投产前，通常都会进行用户验收测试。如果用户验收测试没有通过，

直接结果就是那不到“Money”，间接影响是损害了公司的形象，而后者的影响往往更严重。根据作者的经验，用户验收测试一定要让用户满意。

实际上用户现场测试更趋于是一种演示。在不欺骗用户的前提下，我们向用户展示我们软件的优点，最后让“上帝”满意并欣然掏出“银子”才是我们的目标。

因此用户测试要注意下面的事项：

(1)用户现场测试不可能测试全部功能，因此要测试核心功能。这需要提前做好准备，这些核心功能一定要预先经过测试，证明没有问题才可以和用户共同进行



测试。测试核心模块的目的是建立用户对软件的信心。当然如果这些模块如果问题较多，不应该进行演示。

(2)如果某些模块确实有问题，我们可以演示其它重要的业务功能模块，必要时要向用户做成合理的解释。争得时间后，及时修改缺陷来弥补。

(3)永远不能欺骗用户，蒙混过关。道理很简单，因为软件是要给用户用的，问题早晚会暴露出来，除非你可以马上修改。

和用户进行测试还要注意各种交流技巧，争取不但短期利益得到了满足，还要为后面得合作打好基础。

#### 58、如何编写提交给用户的测试报告？

参考答案：

随着测试工作越来越受重视，开发团队向客户提供测试文档是不可避免的事情。

很多人会问：“我们可以把工作中的测试报告提供给客户吗？”答案是否定的。

因为提供内部测试报告，可能会让客户失去信心，甚至否定项目。

测试报告一般分为内部测试报告和外部测试报告。内部报告是我们在测试工作中的项目文档，反映了测试工作的实施情况，这里不过多讨论，读者可以参考相关教材。这里主要讨论一下外部测试报告的写法，一般外部测试报告要满足下面几个要求：

- 根据内部测试报告进行编写，一般可以摘录;
- 不可以向客户报告严重缺陷，即使是已经修改的缺陷，开发中的缺陷也没有必要让客户知道;
- 报告上可以列出一些缺陷，但必须是中级的缺陷，而且这些缺陷必须是修复的;
- 报告上面的内容尽量要真实可靠;

-整个测试报告要仔细审阅，力争不给项目带来负面作用，尤其是性能测试报告。

总之，外部测试报告要小心谨慎的编写。

59、测试工具在测试工作中是什么地位？

参考答案：

国内的很多测试工程师对测试工具相当迷恋，尤其是一些新手，甚至期望测试工具可以取代手工测试。测试工具在测试工作中起的是辅助作用，一般用来提高测试效率。自动化测试弥补了手工测试的不足，减轻一定的工作量。实际上测试工具是无法替代大多数手工测试的，而一些诸如性能测试等自动化测试也是手工所不能完成的。

对于自动测试技术，应当依据软件的不同情况来分别对待，一般自动技术会应用在引起大量重复性工作的地方、系统的压力点、以及任何适合使用程序解决大批量输入数据的地方。然后再寻找合适的自动测试工具，或者自己开发测试程序。一定不要为了使用测试工具而使用。

60、什么是软件测试，软件测试的目的？

参考答案：

61、简述负载测试与压力测试的区别。

参考答案：

压力测试(Stress Testing)

压力测试的主要任务就是获取系统正确运行的极限，检查系统在瞬间峰值负荷下正确执行的能力。例如，对服务器做压力测试时就可以增加并发操作的用户数量；或者不停地向服务器发送请求；或一次性向服务器发送特别大的数据等。看看服务器保持正常运行所能达到的最大状态。人们通常使用测试工具来完成压力测

试，如模拟上万个用户从终端同时登录，这是压力测试中常常使用的方法。

### 负载测试(Volume Testing)

用于检查系统在使用大量数据的时候正确工作的能力，即检验系统的能力最高能达到什么程度。例如，对于信息检索系统，让它使用频率达到最大;对于多个终端的分时系统，让它所有的终端都开动。在使整个系统的全部资源达到“满负荷”的情形下，测试系统的承受能力。

62、写出 bug 报告流转的步骤，每步的责任人及主要完成的工作。

参考答案：(要结合自身实际的工作经验进行回答，不同公司略有区别)

测试人员提交新的 Bug 入库，错误状态为 New。

高级测试员/测试经理验证错误，如果确认是错误，分配给开发组。设置状态为 Open。如果不是错误，则拒绝，设置为 Declined 状态。

开发经理分配 bug 至对应的模块开发人员。

开发人员查询状态为 Open

的 Bug，如果不是错误，则置状态为 Declined;

如果

是 Bug 则修复并置状态为 Fixed。不能解决的 Bug，要留下文字说明及保持

Bug

为 Open 状态。

对于不能解决和延期解决的

Bug，不能由开发人员自己决定，一般要通过某种

会议(评审会)通过才能认可。

测试人员查询状态为 Fixed 的 Bug，然后验证 Bug 是否已解决，如解决，置

Bug

的状态为

Closed, 如没有解决,置 bug 状态为 Reopen。

63、写出

bug 报告当中一些必备的内容。

参考答案:

硬件平台和操作系统

测试应用的硬件平台(Platform), 通常选择 “PC” 。

测试应用的操作系统平台(OS)。

a) 版本

提交缺陷报告时通过该字段标识此缺陷存在于被测试软件的哪个版本。

b) Bug

报告优先级

c) Bug

状态

d) Bug

的编号

e)

发现人

f)

提交人

g)

指定处理人

h)

概述

i)

从属关系

j)

详细描述

k)

严重程度

l)

所属模块

m) 附件

n) 提交日期

64、开发人员老是犯一些低级错误怎么解决？

参考答案：

这种现象在开发流程不规范的团队里特别常见，尤其是一些“作坊式”的团队里。

解决这种问题一般从两个方面入手：

一方面从开发管理入手，也就是从根源来解决问题。可以制定规范的开发流程，甚至可以制定惩罚制度，还有就是软件开发前做好规划设计。

另一方面就是加强测试，具体做法就是加强开发人员的自己测试，把这些问题“消灭”在开发阶段，这是比较好的做法，读者可以参考第 13 章试案例分析的

“13.1.2

缺陷反复出现，谁的责任”小节，13.1.2 专门讨论了这类问题的方法。

此外，还可以通过规范的缺陷管理来对开发人员进行控制，比如测试部门整理出常见的缺陷，让开发人员自己对照进行检查，以减少这类低级错误的发生。

开发人员犯错误是正常的现象，作为测试人员一定不能抱怨，要认认真真的解决问题才是上策。

65、画出软件测试的 V 模型图。

参考答案：

66、为什么要在一个团队中开展软件测试工作？

参考答案： 因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量

认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

67、您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

参考答案：(根据项目经验不同，灵活回答即可)

我曾经做过 web 测试，后台测试，客户端软件，其中包括功能测试，性能测试，用户体验测试。最擅长的是功能测试

68、您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系(如功能测试、性能测试.....)

参考答案：

测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。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。采用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。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶

颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范(快捷键之类的)，是否美观(能否吸引用户的注意力)，是否安全(尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告)?

做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的



的，然后再考虑该功能点的性能测试

69、您认为做好测试用例设计工作的关键是什么？

参考答案：

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

70、请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、

验收测试的区别与联系。

参考答案：

黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

- 1、是否有不正确或遗漏的功能？
- 2、在接口上，输入是否能正确的接受？能否输出正确的结果？
- 3、是否有数据结构错误或外部信息(例如数据文件)访问错误？
- 4、性能上是否能够满足要求？

## 5、是否有初始化或终止性错误？

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行如下检查：

- 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”与取“假”的两种情况都能至少测一遍。
- 3、在循环的边界和运行的界限内执行循环体。
- 4、测试内部数据结构的有效性，等等。

单元测试(模块测试)是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件(或者场景)下某个特定函数的行为。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

集成测试(也叫组装测试，联合测试)是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起

测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。(常见的联调测试)

系统测试的目的是对最终软件系统进行全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备

就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

71、测试计划工作的目的是什么?测试计划工作的内容都包括什么?其中哪些是最重要的?

参考答案：

软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完

成测试任务的具体战术。所以其中最重要的是测试策略和测试方法(最好是能先评审)

72、您所熟悉的测试用例设计方法都有哪些?请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

参考答案:

### 1.等价类划分

划分等价类: 等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的.并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试.因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据.取得较好的测试结果.等价类划分可有两种不同的情况:有效等价类和无效等价类.

### 2.边界值分析法

边界值分析方法是对等价类划分方法的补充。测试工作经验告诉我,大量的错误是发生在输入或输出范围的边界上,而不是发生在输入输出范围的内部.因此针对各种边界情况设计测试用例,可以查出更多的错误.

使用边界值分析方法设计测试用例,首先应确定边界情况.通常输入和输出等价类的边界,就是应着重测试的边界情况.应当选取正好等于,刚刚大于或刚刚小于边界的值作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据.

### 3.错误推测法

基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法.

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.

#### 4.因果图方法

前面介绍的等价类划分方法和边界值分析方法,都是着重考虑输入条件,但未考虑输入条件之间的联系, 相互组合等. 考虑输入条件之间的相互组合,可能会产生一些新的情况. 但要检查输入条件的组合不是一件容易的事情, 即使把所有输入条件划分成等价类,他们之间的组合情况也相当多. 因此必须考虑采用一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例. 这就需要利用因果图(逻辑模型). 因果图方法最终生成的就是判定表. 它适合于检查程序输入条件的各种组合情况.

73、请以您以往的实际工作为例，详细的描述一次测试用例设计的完整的过程。

参考答案：

就说最近的这次网站功能的测试吧

首先：得到相关文档(需求文档和设计文档)，理解需求和设计思想后，想好测试策略(测试计划简单点就 OK 了)，考虑到测试环境，测试用例，测试时间等问题。

第二步：设计测试用例，测试策略是：把网站部分的功能点测试完，然后在进行系统测试(另外个模块呢有另一个测试人员负责，可以进行联调测试)，网站模块

的测试基本是功能测试和界面测试(用户并发的可能性很小, 所以不考虑): 这次的网站的输入数据呢是使用数据库中的某张表记录, 如果表中某一数据记录中新加进来的(还没有被处理的, 有个标志位), 网站启动后会立刻去刷那张表, 得到多条数据, 然后在进行处理。处理过程中, 会经历 3 个步骤, 网站才算完成了它的任务。有 3 个步骤呢, 就可以分别对这 3 个步骤进行测试用例的设计, 尽量覆盖到各种输入情况(包括数据库中的数据, 用户的输入等), 得出了差不多 50 个用例。界面测试, 也就是用户看的到的地方, 包括发送的邮件和用户填写资料的页面展示。

第三步: 搭建测试环境(为什么这个时候考虑测试环境呢?因为我对网站环境已经很熟了, 只有有机器能空于下来做该功能测试就可以做了), 因为网站本身的环境搭建和其他的系统有点不同, 它需要的测试环境比较麻烦, 需要 web 服务器 (Apache,tomcat), 不过这次需求呢, 网站部分只用到了 tomcat, 所以只要有 tomcat 即可

第四步: 执行测试

74、您以往是否曾经从事过性能测试工作?如果有, 请尽可能的详细描述您以往的性能测试工作的完整过程。

参考答案: (以自己最熟悉的性能测试项目为例)

是的, 曾经做过网站方面的性能测试, 虽然做的时间并不久(2 个月吧), 当时呢, 是有位网站性能测试经验非常丰富的前辈带着我一起做。

性能测试类型包括负载测试, 强度测试, 容量测试等

负载测试：负载测试是一种性能测试指数数据在超负荷环境中运行，程序是否能够承担。

强度测试：强度测试是一种性能测试，他在系统资源特别低的情况下软件系统运行情况

容量测试：确定系统可处理同时在线的最大用户数

在网站流量逐渐加大的情况下，开始考虑做性能测试了，首先要写好性能测试计划，根据运营数据得出流量最大的页面(如果是第一次的话，一般是首页，下载页，个人帐户页流量最大，而且以某种百分比)，

Web 服务器指标指标：

- \* Avg Rps: 平均每秒钟响应次数=总请求时间 / 秒数;
- \* Successful Rounds: 成功的请求;
- \* Failed Rounds : 失败的请求;
- \* Successful Hits : 成功的点击次数;
- \* Failed Hits : 失败的点击次数;
- \* Hits Per Second : 每秒点击次数;
- \* Successful Hits Per Second : 每秒成功的点击次数;
- \* Failed Hits Per Second : 每秒失败的点击次数;
- \* Attempted Connections : 尝试链接数;

75、你对测试最大的兴趣在哪里?为什么?

参考答案：

最大的兴趣就是测试有难度，有挑战性!做测试越久越能感觉到做好测试有多难。

曾经在无忧测试网上看到一篇文章，是关于如何做好一名测试工程师。一共罗列



了 11, 12 点, 有部分是和人的性格有关, 有部分需要后天的努力。但除了性格

有关的 1, 2 点我没有把握, 其他点我都很有信心做好它。

刚开始进入测试行业时, 对测试的认识是从无忧测试网上了解到的一些资料, 当时是冲着做测试需要很多技能才能做的好, 虽然入门容易, 但做好很难, 比开发更难, 虽然当时我很想做开发(学校专业课我基本上不缺席, 因为我喜欢我的专业), 但看到测试比开发更难更有挑战性, 想做好测试的意志就更坚定了。

不到一年半的测试工作中, 当时的感动和热情没有减退一点(即使环境问题以及自身经验, 技术的不足, 做测试的你一定也能理解)。

我觉得做测试整个过程中有 2 点让我觉得很有难度(对我来说, 有难度的东西我就非常感兴趣), 第一是测试用例的设计, 因为测试的精华就在测试用例的设计上了, 要在版本出来之前, 把用例写好, 用什么测试方法写?(也就是测试计划或测试策略), 如果你刚测试一个新任务时, 你得花一定的时间去消化业务需求和技术基础, 业务需求很好理解(多和产品经理和开发人员沟通就能达到目的), 而技术基础可就没那么简单了, 这需要你自觉的学习能力, 比如说网站吧, 最基本的技术知识你要知道网站内部是怎么运作的, 后台是怎么响应用户请求的?测试环境如何搭建?这些都需要最早的学好。至少在开始测试之前能做好基本的准备, 可能会遇到什么难题?需求细节是不是没有确定好?这些问题都能在设计用例的时候发现。

第二是发现 BUG 的时候了, 这应该是测试人员最基本的任务了, 一般按测试用

例开始测试就能发现大部分的 bug, 还有一部分 bug

需要测试的过程中更了

解

所测版本的情况获得更多信息，补充测试用例，测试出

bug。还有如何发现

bug?

这就需要在测试用例有效的情况下，通过细心和耐心去发现 bug 了，每个用例都有可能发现 bug，每个地方都有可能出错，所以测试过程中思维要清晰(测试过程数据流及结果都得看仔细了，bug 都在里面发现的)。如何描述 bug 也很有

讲究，bug 在什么情况下会产生，如果条件变化一点点，就不会有这个 bug，以哪些最少的操作步骤就能重现这个 bug，这个 bug 产生的规律是什么?如果你

够厉害的话，可以帮开发人员初步定位问题。

76、你以前工作时的测试流程是什么?

参考答案：(灵活回答)

公司对测试流程没有规定如何做，但每个测试人员都有自己的一套测试流程。我说下我 1 年来不断改正(自己总结，吸取同行的方法)后的流程吧。需求评审(有开

发人员，产品经理，测试人员，项目经理)->需求确定(出一份确定的需求文档)->开发设计文档(开发人员在开始写代码前就能输出设计文档)->想好测试策略，写出测试用例->发给开发人员和测试经理看看(非正式的评审用例)->接到测试版本->执行测试用例(中间可能会补充用例)->提交 bug(有些 bug 需要开发人

员的

确定(严重级别的, 或突然发现的在测试用例范围之外的, 难以重现的), 有些可以直接录制进 TD)->开发人员修改(可以在测试过程中快速的修改)->回归测试(可能又会发现新问题, 再按流程开始跑)。

77、当开发人员说不是 BUG 时, 你如何应付?

参考答案:

开发人员说不是 bug, 有 2 种情况, 一是需求没有确定, 所以我可以这么做, 这个时候可以找来产品经理进行确认, 需不需要改动, 3 方商量确定好后再看要不要改。二是这种情况不可能发生, 所以不需要修改, 这个时候, 我可以先尽可能的说出是 BUG 的依据是什么?如果被用户发现或出了问题, 会有什么不良结果?

程序员可能会给你很多理由, 你可以对他的解释进行反驳。如果还是不行, 那我可以给这个问题提出来,跟开发经理和测试经理进行确认,如果要修改就改,如果不要修改就不改。其实有些真的不是 bug, 我也只是建议的方式写进 TD 中, 如

果开发人员不修改也没有大问题。如果确定是 bug 的话, 一定要坚持自己的立场, 让问题得到最后的确认。

78、软件的构造号与版本号之间的区别?BVT(BuildVerificationTest)

参考答案: 版本控制命名格式: 主版本号.子版本号[.修正版本号[.编译版本号]]

Major.Minor [.Revision[.Build]]

应根据下面的约定使用这些部分:

Major：具有相同名称但不同主版本号的程序集不可互换。例如，这适用于对产品的大量重写，这些重写使得无法实现向后兼容性。

Minor：如果两个程序集的名称和主版本号相同，而次版本号不同，这指示显著增强，但照顾到了向后兼容性。例如，这适用于产品的修正版或完全向后兼容的新版本。

Build：内部版本号的不同表示对相同源所作的重新编译。这适合于更改处理器、平台或编译器的情况。

Revision：名称、主版本号和次版本号都相同但修订号不同的程序集应是完全可互换的。这适用于修复以前发布的程序集中的安全漏洞。

BVT(BuildVerificationTest):

作为 Build 的一部分，主要是通过对基本功能、特别是关键功能的测试，保证新

增代码没有导致功能失效，保证版本的持续稳定。实现 BVT 方式是有以下几种：

1、测试人员手工验证关键功能实现的正确性。特点：这是传统开发方法中，通常采用的方式。无需维护测试脚本的成本，在测试人力资源充足，测试人员熟悉业务、并对系统操作熟练情况下效率很高，比较灵活快速。缺点：人力成本较高；对测试人员能力有一定要求；测试人员面对重复的工作，容易产生疲倦懈怠，从而影响测试质量。2、借助基于 GUI 的自动化功能测试工具来完成，将各基本功能

能操作录制成测试脚本，每次回放测试脚本验证功能实现的正确性。特点：能够模拟用户操作完成自动的测试，从 UI 入口到业务实现，每一层的代码实现都经过验证；节约人力成本；降低测试人员重复劳动的工作量，机器不会疲倦；缺点：对

于 UI 变动比较频繁的系统来说, 这种方式的维护成本很高, 实施起来非常困难。

另外, 在项目周期较短且后续无延续性或继承的情况下, 也不推荐使用此方式。

3、由开发人员通过自动化测试工具完成业务层的 BVT 测试。特点: 通过对业务

层关键功能的持续集成测试, 保证系统功能的持续稳定。可以结合 DailyBuild, 做为 Build 的一部分, 自动实现并输入 BVT 报告。缺点: 仅对业务规则实现的

正确性进行了测试, 对表现层无法测试到, 对于诸如: 前台页面控件各种事件响应、页面元素变化等方面的问题无法保证。

## 字节跳动测试工程师笔试题

1、阶段评审与同行评审的区别?

参考答案:

同行评审目的: 发现小规模工作产品的错误, 只要是找错误;

阶段评审目的: 评审模块 阶段作品的正确性 可行性 及完整性

同行评审人数: 3-7 人 人员必须经过同行评审会议的培训, 由 SQA 指导

阶段评审人数: 5 人左右 评审人必须是专家

具有系统评审资格

同行评审内容: 内容小 一般文档 < 40 页,

代码 < 500 行

阶段评审内容: 内容多, 主要看重点

同行评审时间：一小部分工作产品完成

阶段评审时间：通常是设置在关键路径的时间点上

## 2、什么是软件测试及其目的？

参考答案：

软件测试是使用人工或自动化手段来运行或测定某个系统的过程。其目的是：在于检验它是否能满足规定的需求或是弄清楚预期结果与实际结果之间的差别。

## 3、简述集成测试的过程？

参考答案：

集成测试流程：

在完成软件的概要设计后，即开始制定集成测试计划-》设计集成测试用例和测试过程-》实施集成测试，设计所需驱动和桩-》执行集成测试，记录测试结果-》评估集成测试，根据测试结果评估此次测试，生成评估报告文档。(驱动或桩函数是做单元测试时要用到的。驱动函数是所测 4 函数的主程序，它接收测试

数据，并把数据传送给所测试单元，最后再输出实测结果。当被测单元能完成相关功能时，也可以不要驱动单元。桩，是用来代替所测试单元调用的子单元。)

## 4、白盒测试有哪几种方法？

参考答案：代码审查，语句覆盖，判定覆盖，条件覆盖，组合覆盖，基本路径，形式化方法，符号执行

## 5、简述测试目标有哪些类型？

参考答案：

功能测试，负载测试，性能测试，安全性测试，恢复测试，安装测试，兼容

性测试，可用性测试，可靠性测试，国际化测试，本地化测试。

6、怎么样做好文档测试？

参考答案：

文档的测试主要采用静态测试即走查的方法，可以依据的是同行评审，列出一个检查表，然后大家一起坐下来对着被测试的文档进行阅读排错。通常文档都很长，而一般的建议是——

7、测试结束的标准是什么？

参考答案：

从项目周期看： 1、超出了所分配的测试时间；

2、用尽了分配的测试资源；

3、到达了某一个固定的里程碑（如合同规定的交付日期）。

从测试角度看：

1、测试需求覆盖率；

2、测试代码覆盖率；

3、测试用例度量；

4、缺陷检查度量

8、Alpha 测试与 Beta 测试的区别？

参考答案：

Alpha 测试是由一个用户在开发环境下进行的测试，也可以是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。Alpha 测试发现的错误，可以在测试现场立刻反馈给开发人员，由开发人员及时分析和处理。目的是评价软件产品的功能、可使用性、可靠性、性能和



支持。尤其注重产品的界面和特色。Alpha 测试可以从软件产品编码结束之后开始，或在模块（子系统）测试完成后开始，也可以在确认测试过程中产品达到一定的稳定和可靠程度之后再开始。有关的手册（草稿）等应该在 Alpha 测试前准备好。

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。因而，Beta 测试是在开发者无法控制的环境下进行的软件现场应用。在 Beta 测试中，由用户记下遇到的所有问题，包括真实的以及主管认定的，定期向开发者报告，开发者在综合用户的报告后，做出修改，最后将软件产品交付给全体用户使用。Beta 测试着重于产品的支持性，包括文档、客户培训和支持产品的生产能力。只有当 Alpha 测试达到一定的可靠程度后，才能开始 Beta 测试。由于 Beta 测试的主要目标是测试可支持性，所以 Beta 测试应该尽可能由主持产品发行的人员来管理。

9、系统测试计划是否需要同行审批，为什么？

参考答案：

需要，系统测试计划属于项目阶段性关键文档，因此需要评审。 10、比较负载测试，容量测试和强度测试的区别？

参考答案：

负载测试：在一定的工作负荷下，系统的负荷及响应时间。

强度测试：在一定的负荷条件下，在较长时间跨度内的系统连续运行给系统性能所造成的影响。

容量测试：容量测试目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值（如最大并发用户数、数据库记录数等），系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。次同行评审步的时间不能超过两个小时，因此可以对被测试文档执行测试时，列个计划，将总的文档分解，按照计划多次对被测试的文档进行走查。

## 5 个重要的面试题（含金量高，分水岭能 pass 掉很多人）

1. 结合你以前的工作经验和学习到的测试技术，说说你对质量保证的理解？

这是个综合性基础知识问题，考验你在测试行业工作几年后的心得，是否有测试思维，明确自己的定位。可以谈谈你认为如何做好测试？如何衡量质量？你又如何专注软件质量提升，帮助团队快速迭代高质量交付

2. 非关系型数据库和关系型数据库的区别，谈谈优势比较？

数据库在测试工作中的重要性相信我就不用在这多累赘了，需要你对常见的关系型数据库和非关系型数据库的原理理解，而且总结出它们之前的区别

3. 在自动化方面有什么成熟的方案，有没有做过二次开发？

测试工程师离不开一个关键字，“自动化”，在任何类型的项目或多或少会用到自动化测试技术。关于前后端自动化技术，是否有一套完整的测试框架，技术体系。在开源自动化测试框架的理解上，有一定的开发思维，能进行封装，开源框架上二次开发。能根据项目灵活打造适合团队的测试框架，是测试工程师的能力分水岭

4.dubbo 接口了解吗，它的测试核心是什么？

这明显是个阿里系的问题，dubbo 是阿里开源的一个高性能优秀的服务框架。考验的不仅是对测试技术，做为一个资深测试工程师，你还得了解公司产品的开发基础服务架构

5.质量部如果需要定制化一个质量体系平台，你会考虑哪些问题？

说实话，面试官这么问测试平台的问题，我快扛不住了。干过测试开发的应该都知道，测试平台的构造，除了技术难点。你要考虑如何去推进这个工作？测试平台研发出来后，在整个测试部门推广运营这个平台是大难题  
站在功能测试人员的角度来想，去使用不成熟的测试平台来测试，但凡加大我的工作量，而对我自身工作和技术没有什么太大价值，一般都不是非常积极的去配合，测试平台如何推广运营是个难题。你得让所有人亲身感受到用与不用这个测试平台有什么区别。

在工作多年之后，当你有一定的技术积累和资源，总有一些猎头或者朋友内推，推荐你去大公司。

根据我的面试经验，首先重点在基础知识的考查，二是在技术的深度和原理的理解，再者看你的测试思维和开发思维。每个问题不仅需要你对这个测试技术的掌握，还得从其他维度去考虑，看你在之前的工作和学习中，是否能总结自己的经验和心得  
结合自身面试官的丰富经验，给大家分析从简历到面试的全过程优化

## 接口测试面试题汇总

1，目前市面上流行的接口大多有哪几种协议的接口？

答：http, https, dubbo, rpc 等即可。

2，接口的请求方式有哪几种？

答：get, post, put, delete, head, Trace, options 等，大多以 get 和 post 请求为主

### 3、get 和 post 区别是什么？

答：POST 和 GET 都是向服务器提交数据，并且都会从服务器获取数据。

区别：

(1) 传送方式：get

通过地址栏传输，post 通过报文传输，故而 post 更相对来说私密性一点

(2) 传送长度：get

参数有长度限制（受限于 url 长度），而 post 无限制

(3) get

请求参数会被完整保留在浏览历史记录里，而 post 中的参数不会被保留

(4) get

方式大多用作查询接口，获取响应数据；而 post 方式更多做数据添加、修改或删除等操作

### 4、post 请求的请求类型有哪几种？

- 

application/json

json 字符串

- 

- 

application/x-www-form-urlencoded

表单传递

- 

- 

multipart/form-data

主要用于上传文件

- 

### 5、cookie 和 session 的区别

- 

cookie

数据存放在客户的浏览器上，session 数据放在服务器上

- 

- 

cookie

不是很安全，别人可以分析存放在本地的 cookie 并进行 cookie 欺骗，考虑到安全应当使用 session

- 

- 

session 会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑到减轻服务器性能方面应当使用

cookie

-

- 单个 cookie 保存的数据不能超过 4K，很多浏览器都限制一个站点最多保存 20 个 cookie

- 
- 可以将登陆信息等重要信息存放为 session；其他信息需要保存，可以放在 cookie

## 6、请求接口中常见的返回状态码

答：

**a.**

1xx

— 信息提示（表示临时的响应。客户端在收到常规响应之前，准备接收一个或多个

1xx

响应）

**b.**

**c.**

2xx

—

成功（表明服务器成功地接受了客户端请求）

**d.**

**e.**

3xx

—

重定向（客户端浏览器必须采取更多操作来实现请求，例如用户未登录就操作了修改的功能）

**f.**

**g.**

4xx

—

客户端错误（发送错误，客户端有问题）

**h.**

**i.**

5xx

—

服务器错误（服务器由于遇到错误而不能完成该请求）

**j.**

常见的有：

- 200 OK：服务器成功返回用户请求的数据
- 
- 201：用户新建或修改数据成功

- 
- 
- 202：表示一个请求已经进入后台排队（异步任务）

- 
- 
- 301：删除请求数据

- 
- 
- 302：在其他地址发现了请求数据

- 
- 
- 303：建议客户访问其他 URL 或访问方式

- 
- 
- 304：客户端已经执行了 GET，但文件未变化

- 
- 
- 400：用户发出的请求有错误，服务器没有进行新建或修改数据的操作

- 
- 
- 401：表示用户没有权限（令牌、用户名、密码错误）

- 
- 
- 403：表示用户得到授权（与 401 错误相对），但是访问被禁止

- 
- 
- 404：用户发出的请求针对得到是不存在的记录，服务器没有进行操作，该操作是幂等的

- 
- 
- 500：服务器发生错误，用户将无法判断发出的请求是否成功。

- 
- 
- 502：服务器返回超时

## 7、接口测试用例如何进行设计

- 
- 针对输入，可按照参数类型进行设计，参数是否必填，参数之间是否存在关联，参数数据类型限制，参数数据类型自身的数据范围值限制；

- 
-

针对接口处理，可按照逻辑进行用例设计；

- 
- 

针对输出，可根据结果进行分析设计。

- 

#### 8、如何分析是前端还是后端的问题

答：

- 

检查接口，前端和后台之间是通过接口文件相互联系的，需要查看接口文件

- 
- 

检查请求的数据是什么，反馈的数据又是什么

- 
- 

页面可以直接 F12，或者抓包查看。如果发送的数据是正确的，但是后台反馈的数据是不符合需求的，那就是后台的问题；如果前端没有请求接口或请求的时候发送数据与需求不符，那这个时候就是前端的问题了。

- 

#### 9、接口测试中，下游接口需要依赖上游接口的数据，该如何处理？

答：在工具中可以使用全局变量等方式将需要的数据进行传送，或者使用对响应数据进行提取，传给下游接口。

#### 10、依赖第三方数据的接口如何进行测试？

答：可以使用 fiddler 进行调用接口时预设期望响应，mock 返回自己设置的响应数据，最大限度的降低对第三方数据接口的依赖

#### 11、若请求的接口需要先登录后方可请求，如何进行接口测试？

答：请求登录口获取返回的响应头，或者响应信息中的数据，cookie，token，session

等，传递给依赖登录接口的请求头中，发起请求即可。

## App/Web 自动化的面试题（内含参考答案）

十七、

App/Web 自动化的面试题（详细答案自己百度哈,前加粗的表示

app

自动化专有问题、倾斜的 web 自动化专有的问题，无格式的表达共有



的问题)

备注：一般面试的时候是结合简历上的项目一步一步深入进行问的问题，只要你实际做过一个项目的，加上自己面试准备下，问题不会太大

### 1、目前主流的 APP 自动化测试框架，各个自动化适合的语言

参考点：appium macaca、robotium、UiAutomator

### 2、你对单元测试框架了解多少

参考点：unittest、testng、nose

### 3、app 自动化，简述一下这几种自动化框架的异同

### 4、定位方式有哪几种？一般如何选择

参考点：

ID/className/Name/LinkText/PartialLinkText/Xpath/CSS  
selector

### 5、UI 自动化能发现多少 Bug

参考点：UI 自动化的目的不是为了发现多少 Bug，主要是为了减轻重复的基础操作和线上监控的作用

### 6、monkey 属于自动化吗？

参考点：monkey 不属于严格意义上的自动化，monkey 是生成用户或系统的伪随机事件，在屏幕上触发随机点击事件

### 7、谈谈你们的自动化是怎么搭建的？

### 8、你们在自动化过程中遇到了哪些问题

### 9、你们一般对什么 case 会进行自动化，自动化一般在哪个阶段进行

参考点：主要是主流程中比较容易实现的进行自动化，一般在集成阶段进行该版本的自动化监控，平常的话会一直跑线上监控的

#### **10、robotium 可以跨进程吗?你能简述一下吗?**

#### **11、app 自动化你们一般用什么工具定位元素?**

参考点：UiAutomatorview 和 appium 的客户端

#### **12、您需要一台服务器机器来运行 Appium 上的测试吗?**

参考点：不，您不需要服务器机器在 Appium 上运行测试。Appium 促进了一个 2 层架构，其中测试机连接到运行 Appium 的测试服务器并自动化整个事情。您可以在运行测试的同一台机器上运行 Appium。

#### **13、使用 Appium 可能遇到的错误是什么?**

参考点：

错误 1：需要以下所需的功能，但不提供：设备名称，platformName

错误 2：找不到 adb。请使用 Android SDK 根目录路径设置

ANDROID\_HOME 环境变量

错误 3：openqa.selenium.SessionNotCreatedException：无法创建新的会话

错误 4：如何在移动应用程序中查找 DOM 元素或 XPath?

#### **14、简述 Appium 的原理?**

参考点：Appium 是使用 Node.js 平台编写的“HTTP Server”，并使用 Webdriver JSON

线协议驱动 iOS 和 Android

会话。因此，在初

始化 Appium Server

之前，必须在系统上预先安装

Node.js 当

Appium 被下载并安装时，在我们的机器上设置一个暴露 REST API 的服务器

它从客户端接收连接和命令请求，并在移动设备（Android / iOS）上执行该命令，

它响应 HTTP 响应。再次，为了执行此请求，它使用移动测试自动化

框架来驱动应用程序的用户界面。框架像 Apple Instruments for iOS

（仅适用于 Xcode 3.0 或更高版本的 OS X v10.5 及更高版本）适用于

Android API 的 Google UIAutomator 16 级或更高版本

Selendroid for Android API 等级在 15 以下。

15、目前你们用的自动化工具版本号是多少？

16、元素等待的 3 种方式简述一下，并且你平常用的是哪一种？

17、断言是什么

18、测试报告你们用的是什么框架？

**19、app 自动化 如何控制在多台设备上执行？**

20、*selenium*

的原理？

21、*selenium*

如何控制多个浏览器执行？

22、*robotframework* 是什么？和 *selenium* 的关系是什么

23、你们目前用的 py 版本是多少的？2.X 和 3.X 的异同？

24、*selenium* 中如何保证操作元素的成功率？

参考点：- 添加元素智能等待时间 `driver.implicitly_wait(30)`

- try 方式进行 id,name,clas,x path, css selector 不同方式进行定位，如果第一种失败可以自动尝试第二种

-Selenium 保证元素成功率是通过元素的定位，当然它的定位方法很多，一定能有合适的。但是在自动化工程的实施过程中，高质量的自动化测试不是只有测试人员保证的。需要开发人员规范开发习惯，如给页面元素加上唯一的 name,id 等，这样就能大大地提高元素定位的准确性。当然如果开发人员开发不规范，我们在定位元素的时候尽量使用相对地址定位，这样能减少元素定位受页面变化的影响。只要我们元素定位准确，就能保证我的每一个操作符合我的预期

25、如何提高 *selenium* 脚本的执行速度？

参考点：如网速，操作步骤的繁琐程度，页面加载的速度，以及我们在脚本中设置的等待时间，运行脚本的线程数等。所以不能单方面追求运行速度的，要确保稳定性，能稳定地实现回归测试才是关键。

我们可以从以下几个方面来提高速度：

一，减少操作步骤，如经过三四步才能打开我们要测试的页面的话，我们就可以直接通过网址来打开，减少不必要的操作。

二，中断页面加载，如果页面加载的内容过多，我们可以查看一下加载

慢的原因，如果加载的内容不影响我们测试，就设置超时时间，中断页面加载。

三，在设置等待时间的时候，可以 sleep 固定的时间，也可以检测某个元素出现后中断等待也可以提高速度。

四，配置 testNG 实现多线程。在编写测试用例的时候，一定要实现松耦合，然后在服务器允许的情况下，尽量设置多线程运行，提高执行速度。

26、selenium2.X 和 3.X 的异同？

27、什么是持续集成？

- 持续集成源于极限编程(XP)，是一种软件实践，软件开发过程中集成步骤是一个漫长并且无法预测的过程。集成过程中可能会爆发大量的问题，因此集成过程需要尽可能小而多，实际上持续集成讲的是不断的去做软件的集成工作。持续集成，最简单的形式是包括一个监控版本控制 (SVN 等等) 变化的工具。当变化被发觉时，这个工具可以自动的编

译并测试你的应用

28、什么是 page object 设计模式？

通过分离测试对象和测试脚本的抽象来实现的

29、selenium 中下拉框怎么定位？

30、你觉得自动化测试最大的缺陷是什么？

参考点：-不稳定

-可靠性

-不易维护

-成本与收益

31、*Selenium* 是否支持桌面应用软件的自动化测试。

Selenium 不支持桌面软件的自动化测试，Selenium 是根据网页元素的属性才定位元素，而其他桌面软件自动化测试工具是根据桌面元素的位置来定位元素，当然现在也有根据桌面元素的属性来定位的。

32、*BDD* 是什么？你了解多少？*TDD* 是什么？

参考点：BDD：行为驱动开发（Behavior Driven Development）

TDD：测试驱动开发（Test-Driven Development）

33、*selenium* 是否可以直接读取 *Excel* 表中测试用例，来执行相关测试

试

参考点：可以的，需要借助第三方库

34、*Selenium* 有哪些组件？

最早的有 Selenium IDE,IDE 只支持安装在 firefox 上一个插件，支持录制自动化脚本。还有

remote RC,和 Grid 和 webdriver。我们一般最重要的就是使用 webdriver。

**最后：**

欢迎大家关注我的公众号**程序员一凡**获取更多软件测试大厂面试、技术进阶资料。如果对你有帮助，我也会很开心，软件测试，与你同行！

## 软件测试面试题解析

1、有一个考核流程的功能需求如下，请运用系统测试用例设计方法，设系统测试用例：

**答案解析：**

1、审计员页面提交功能验证：输入 20 个汉字、金额 8 位正整数、备注随意输入

选择一个审计主管提交，流程进入下一环节

2、审计员页面提交功能验证：输入 1 个汉字、金额 1 位正整数、备注不填选

择多个(2 个、最大支持审核人数)审计主管提交，流程进入下一环节

3、以上 1、2

中，任一主管审核通过，选择一个事务所人员进行审批流程流转  
到下一环节

4、以上 1、2

中，任一主管审核通过，选择多个（2 个、最大支持审核人数）  
事务所人员进行审批流程流转下一环节

5、以上 1 中，主管审核不通过，流程返回上一环节

6、以上 1 中，多个审核主管，全部不同意，流程返回上一环节

7、针对以上

3

中，事务所人员同意，流程结束。

8、针对以上

3

中，事务所人员不同意，流程退回到审计员

9、针对以上



4

中，多个事务所人员全部同意，流程才结束。

10、针对以上 4 中，多个事务所人员有存在 1 个不同意，流程退回到审计员。

11、审计员提交页面异常输入验证：1) 必填项输入为空或空格、2) 标题输入 21 个汉字 3) 金额输入为 0、负数、正小数、1 位金额数、最大 8 位正整数 +1

12、审计员提交页面输入部分数据后，点击保存，返回上级界面再进入能否支持数据保存

13、审计员已提单，状态为审核中/同意/不同意，该单是否支持修改再提交？

(不同意/不同意/同意)

14、审计员保存业务单后，可进行删除。

15、审计员提单后，状态为审核中/同意/不同意，该单是否支持删除？

16、审计员、主管、事务所人员不同角色页面权限验证

2、会不会项目测试环境搭建？如果会请列出详细步骤？

答案：由于公司项目性质不一样，具体测试环境搭建步骤就会不一样，这个问题学员如果具体回答了，收集回答，由老师查阅是否 ok

3、简述一下

`ps -ef | grep tomcat` 的作用

答案：`ps -ef`

是查找所有进程的命令。

该命令的作用是去判断 `tomcat` 进程是否有启动，同时也可以判断如果

`tomcat`

进程启动了，那么进程号是多少

4、简述一下 `netstat -apn | grep 8080` 的作用

答案：`netstat -apn` 是查找所有端口的命令。

该命令的作用是去判断 8080 这个端口是否已经被占用了，如果有进程在占用这个端口，那么可以判断是哪个进程在使用这个端口

5、简述一下 `kill -9` 的作用

答案：强制杀死某个进程

6、MySQL 数据库中主键的特征是什么？主键的语法关键字是什么？

答案：

a.主键字段存储的值不能重复，因此主键可以唯一标识一行数据。

b.一个表只能有一个主键。

c.主键是非必需的，也就是说一个表可以不声明主键字段。

关键字：`primary key`

7、

用一条 SQL 语句查询出每门课都大于 80 分的学生姓名（表名 `TestScores`）  
表结构如下：

Name

Course

Score

答案：

`select Name from TestScores where Name not in (select distinct Name`

`from TestScores where Score<=80);`

或者

`select Name from TestScores group by Name having min(Score) > 80;`

8、

style(Style\_id,Style\_number,Style\_description) ---样式 id, 样式编码, 样式详情描述

color(Color\_id,Style\_id,Color\_code,Color\_name) ---颜色 id, 样式 id, 颜色编码, 颜色名称

Size(Size\_id,Color\_id,Size\_code,Size\_name) ---尺码 id, 颜色

id, 尺码编码,

尺码名称

Sales(Style\_id,Color\_id,Size\_id,Year, Month, Day,

Sales\_unit,Sales\_cost,Sales\_retail) ---样式 id, 颜色 id, 尺码

id, 年,月, 日,

销售数量, 销售成本, 销售金额

a. 查出 style 表里, style\_number 里包含了"SS"字段的所有 style 信息

答案: select \* from style where style\_number like '%SS%';

b. 查出 style\_number 为 S001 的 color 信息

答案: select color.color\_id, color.style\_id, color.color\_code,color.

color\_name from style,color where style.style\_id=color.style\_id and

style\_number=S001;

c. 用一个 select 语句查询出三个字段 style\_number,color\_code,size\_code

答案: select t3.style\_number,t1.color\_code, t2.size\_code from color t1,  
size t2, style t3 where t2.color\_id=t1.color\_id and t1.style\_id=t3.style\_id;

d. 查出 2015 年 12 份的销售记录, 只需要显示 style\_number,color\_code,

sales\_cost,sales\_retail 这些字段

答案: select t1.style\_number, t3.color\_code, t2.sales\_cost,sales\_retail from

style t1,sales t2, color t3 where t2.style\_id=t1.style\_id and

t2.color\_id=t3.color\_id and t2.Year=2015 and t2.Month=12;

9、请简述一下以 HTTP 接口为例（可以是 get 请求或者 post 请求），详细描述接口测试的一般流程。

答案：收集回答，由老师查阅是否 ok

10、列出常用的接口测试工具，并且挑一个你熟悉的工具，写出详细的使用步骤。

答案：收集回答，由老师查阅是否 ok

11、现有一个字典：dict = {'name':'小明','age':18,'occup':'students','teacher':{'语文':'李老师','数学':'王老师','英语':'张老师'}}，请获取到小明同学的名字；然后再获取到小明的数学老师。

答案：这题考的是字典数据类型的取值。

```
name= dict.get('name')
```

```
teacher= dict['teacher']['数学']
```

```
print(name)
```

```
print(teacher)
```

12、现在有一个列表，list = [12, 'python','nmb',[1,'b',5]]，判断列表长度否大于 5，如果大于 5，就输出 True,否则 False.

答案：这题考的是列表数据类型以及 if 判断。

```
if len(list) > 5:
```

```
print(True)
```

```
else:
```

```
print(False)
```

13、设计一个函数，获取一个 100

以内偶数的纯数字序列，并存到列表里，然后求这些偶数数字的和。

答案： 这题考的是 for 循环以及

range 函数的使用

```
def list_func(num):
```

```
list = []
```

```
sum = 0
```

```
for i in range(0,num,2):
```

```
list.append(i)
```

```
print(list)
```

```
for info in list:
```

```
sum = sum + info
```

```
return sum
```

```
sum = list_func(100)
```

```
print(sum)
```

## 软件测试真题试卷

单项选择题：共 20 小题，每小题 1 分，满分 20 分；请将答案填入题后括号中。

1. 在软件生命周期的哪一个阶段，软件缺陷修复费用最低（**A**）

A.需求分析（编制产品说明书） B.设计

C.编码

D.产品发布

2. 单元测试用来模拟被测模块调用者的模块是（**C**）

A.父模块

B.子模块

C.驱动模块 D.桩模块

3. 为了提高测试的效率，应该（D）

A.随机地选取测试数据；

B.取一切可能的输入数据作为测试数据；

C.在完成编码以后制定软件的测试计划；

D.选择发现错误可能性大的数据作为测试数据。

4. 侧重于观察资源耗尽情况下的软件表现的系统测试被称为（B）

A.强度测试 B.压力测试

C.容量测试

D.性能测试

5. 必须要求用户参与的测试阶段是（D）

A.单元测试

B.集成测试

C.确认测试 D.验收测试

6. 软件测试员究竟做些什么。（C）

A.软件测试员的目的是发现软件缺陷

B.软件测试员的目的是发现软件缺陷，尽可能早一些

C.软件测试员的目的是发现软件缺陷，尽可能早一些，并确保其得以修复

D.软件测试员的目的是发现软件缺陷，尽可能早一些，并将其得以修复

7. 下面四种说法中正确的是（C）

- A.因果图法是建立在决策表法基础上的一种白盒测试方法；
- B.等价类划分法是边界值分析法的基础；
- C.健壮性等价类测试的测试用例要求在有效等价类中取值；
- D.在任何情况下做黑盒测试皆应首先考虑使用错误推断法。

8. 不属于单元测试内容的是（A）

- A.模块接口测试 B.局部数据结构测试
- C.路径测试
- D.用户界面测试

9. 划分软件测试属于白盒测试还是黑盒测试的依据是（C）

- A.是否执行程序代码
- B.是否能看到软件设计文档
- C.是否能看到被测源程序 D.运行结果是否确定

10. 下列项目中不属于测试文档的是（C）

- A.测试计划
- B.测试用例
- C.程序流程图 D.测试报告

11. 几乎没有产品计划、进度安排和正规的开发过程的软件开发模式是（B）

- A.大棒模式 B.边写边改模式
- C.瀑布模式
- D.快速原型开发模式

12. 如果某测试用例集实现了某软件的路径覆盖，那么它一定同时实现了该软件



的（A）

A.判定覆盖 B.条件覆盖

C.判定/条件覆盖

D.组合覆盖

13. 下列说法不正确的是（D）

A.测试不能证明软件的正确性；

B.测试员需要良好的沟通技巧；

C.QA 与 testing 属于一个层次的概念；

D.成功的测试是发现了错误的测试。

14. 对 Web 网站进行的测试中，属于功能测试的是（B）

A.连接速度测试 B.链接测试

C.平台测试

D.安全性测试

15. 在进行单元测试时，常用的方法是（B）

A.采用黑盒测试，辅之以白盒测试；

B.采用白盒测试，辅之以黑盒测试；

C.只使用黑盒测试；

D.只使用白盒测试。

16. 使用白盒测试方法时，确定测试数据的依据是指定的覆盖标准和（B）

A.程序的注释 B.程序的内部逻辑

C.用户使用说明书

D.程序的需求说明

17. 下列\_\_不是软件自动化测试的优点 (D)

- A.速度快、效率高
- B.准确度和精确度高
- C.能提高测试的质量 D.能充分测试软件

18. 配置测试 (C)

- A.是指检查软件之间是否正确交互和共享信息
- B.是交互适应性、实用性和有效性的集中体现
- C.是指使用各种硬件来测试软件操作的过程
- D.检查缺陷是否有效改正

19. 下列各项中\_\_不是一个测试计划所应包含的内容 (B)

- A.测试资源、进度安排 B.测试预期输出
- C.测试范围
- D.测试策略

20. 下列不属于正式审查的方式是 (D)

- A.同事审查
- B.公开陈述
- C.检验 D.编码标准和规范

二、判断题：共 10 小题，每小题 1 分，满分 10 分；请将答案以“√”、“×”形式填入题后括号中。

1. 好的测试员不懈追求完美。

(×)

2.

软件测试是有效的排除软件缺陷的手段。（√）

3.

程序员与测试工作无关。（×）

4.

程序员兼任测试员可以提高工作效率。（×）

5.

产品说明书（需求文档）的变更应当受到控制。（√）

6.

白盒测试的“条件覆盖”标准强于“判定覆盖”。（×）

7.

软件开发全过程的测试工作都可以实现自动化。（×）

8.

找出的软件缺陷越多，说明剩下的软件缺陷越少。（×）

9.

采用自动化测试有可能延误项目进度。（√）

10. 测试应从“大规模”开始，逐步转向“小规模”。（×）

三、填空题：共 10 小题 20 个空格，每个空格 1 分，满分 20 分；请将答案填入指定的空白处。

1.

软件缺陷产生的原因包括软件说明书、设计、编写代码以及其他原因。

2.

软件开发模式包括大棒模式、边写边改模式、流水模式、螺旋模式。

3.

黑盒测试用例设计方法包括等价类划分法，边界值法，以及因果图法，错误推测法等。

4. 测试过程中，\_测试计划\_描述用于描述测试的整体方案，\_缺陷报告\_描述依据测试案例找出的问题。

5. 传统的等价类划分测试的实现分两步进行，一是\_划分等价类表\_，二是\_边界值分析法\_。

6. 在兼容性测试中，\_向前兼容\_是指可以使用软件的以前版本，\_向后兼容\_是指可以使用软件的未来版本。

7. 测试工作就是进行输入、接受输出、检验结果，不深入代码细节，这样的测试

方法称为\_动态黑盒测试\_，在不执行代码的条件下有条理地仔细审查软件设计、体系结构和代码，从而找出软件缺陷的测试方法称为\_静态白盒测试\_。

8. 不需要修复软件缺陷的原因包括\_没有时间\_、不能算真正的软件缺陷、\_风险太大\_、不值得修复。

9. 白盒测试又称为\_结构测试或逻辑结构测试\_，黑盒测试又称为\_功能测试或基于规格说明书的测试\_。

10. 动态测试的两个基本要素是\_被测试程序\_、\_测试数据\_。

四、名词解释：共 5 小题，每小题 3 分，满分 15 分

1. 软件缺陷

1. 软件未达到产品说明书的功能
2. 软件出现了产品说明书指明不会出现的错误
3. 软件功能超出产品说明书指明范围

4. 软件未达到产品说明书虽未指出但应达到的目标

5. 软件测试员认为难以理解、不易使用、运行速度缓慢、或者最终用户认为不好

## 2. $\beta$ 测试

$\beta$  测试是指软件开发公司组织各方面的典型用户在日常工作中实际使用  $\beta$  版本，并要求用户报告异常情况，提出批评意见。

## 3. 测试的配置管理

配置管理的目的是建立和维护在软件生命周期中软件产品的完整性和一致性。一般来说，软件测试配置管理包括 4 个最基本的活动：①配置标识；②变更控制；③配置状态报告；④配置审计。

## 4. 测试用例

测试用例就是将软件测试的行为活动，做一个科学化的组织归纳。

## 5. 黑盒测试

黑盒测试也称功能测试或数据驱动测试，前提是已知产品所具有的功能，通过测试来检测每个功能是否都正常使用。

五、简答题：共 3 小题，每小题 5 分，满分 15 分。

### 1. 如何划分等价类？

1) .在输入条件规定了取值范围或值的个数的情况下，则可以确立一个有效等价类和两个无效等价类。

2) .在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情况下，则可以确立一个有效等价类和一个无效等价类。

3) .在输入条件是一个布尔量的情况下，可以确立一个有效等价类和一个无效等价类。

4) .在规定了输入数据的一组值（假定  $n$  个），并且程序要对每一个输入值分别处理的情况下，可以确立  $n$  个有效等价类和一个无效等价类。

5) .在规定了输入数据必须遵守的规则的情况下，可以确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。

6) .在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价类进一步的划分为更小的等价类。

2. 软件验收测试应完成哪些主要测试工作？

- 1) 文档资料的审查验收
- 2) 功能测试
- 3) 性能测试
- 4) 强化测试
- 5) 性能降级执行方式测试
- 6) 检查系统的余量要求
- 7) 安装测试
- 8) 用户操作测试

3. 简述软件开发-软件测试的对应关系。

相辅相成，大家都是站在公司的立场上做事。

六、综合题：共 1 小题，每小题 20 分，满分 20 分。

1. 某城市的电话号码由三部分组成。这三部分的名称和内容分别是

地区码：空白或三位数字；

前

缀：非'0'或'1'开头的三位数；

后

缀：四位数字。

假定被调试的程序能接受一切符合上述规定的电话号码，拒绝所有不符合规定的号码，试用等价分类法来设计它的调试用例。

解：第一步：划分等价类，包括 4 个有效等价类，11 个无效等价类。

第二步：确定调试用例。

对 11 个无效等价类，要选择 11 个调试用例，如下所示：

选取的调试数据可以不同，关键是与调试内容相符。

软件测试技术（A 卷）答案及评分标准

一、

单选题（ $20 \times 1 = 20$

分）

二、

判断题（ $10 \times 1 = 10$

分）

三、

填空题（ $20 \times 1 = 20$

分）

1.编写说明书; 编写代码

2.大棒模式; 流水模式

3.等价类划分; 边界值分析法

4.测试计划; 缺陷报告

5.划分等价类表; 设计相应的测试用例

6.向前兼容; 向后兼容

7.动态黑盒测试; 静态白盒测试

8.没有时间; 风险太大

9.结构测试或逻辑驱动测试;

功能测试或数据驱动测试或基于规格说明书的测试

10.被测试程序; 测试数据（测试用例）

四、 名词解释（ $5 \times 3 = 15$  分）

1. 软件缺陷：（符合下列规则的叫软件缺陷）：

- 1) . 软件未达到产品说明书的功能
- 2) . 软件出现了产品说明书指明不会出现的错误
- 3) . 软件功能超出产品说明书指明范围
- 4) . 软件未达到产品说明书虽未指出但应达到的目标
- 5) . 软件测试员认为难以理解、不易使用、运行速度缓慢、或者最终用户认为不好

2.  $\beta$  测试是指软件开发公司组织各方面的典型用户在日常工作中实际使用  $\beta$  版



本，并要求用户报告异常情况，提出批评意见。

3. 配置管理的目的是建立和维护在软件生命周期中软件产品的完整性和一致性。

一般来说，软件测试配置管理包括 4 个最基本的活动：（1）配置标识；（2）变更控制；（3）配置状态报告；（4）配置审计。

4. 测试用例就是将软件测试的行为活动，做一个科学化的组织归纳。

5. 黑盒测试也称功能测试或数据驱动测试，前提是已知产品所具有的功能，通过测试来检测每个功能是否都正常使用。

## 五、简答题（3×5=15 分）

1.

1) .在输入条件规定了取值范围或值的个数的情况下，则可以确立一个有效等价类和两个无效等价类。

2) .在输入条件规定了输入值的集合或者规定了“必须如何”的条件情况下，则可以确立一个有效等价类和一个无效等价类。

3) .在输入条件是一个布尔量的情况下，可以确立一个有效等价类和一个无效等价类。

4) .在规定了输入数据的一组值（假定  $n$  个），并且程序要对每一个输入值分别处理的情况下，可以确立  $n$  个有效等价类和一个无效等价类。

5) .在规定了输入数据必须遵守的规则的情况下，可以确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。

6) .在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价类进一步的划分为更小的等价类。

2. 软件验收测试应完成哪些主要测试工作？

1) 文档资料的审查验收

2) 功能测试

3) 性能测试

- 4) 强化测试
  - 5) 性能降级执行方式测试
  - 6) 检查系统的余量要求
  - 7) 安装测试
  - 8) 用户操作测试
3. 简述软件开发-软件测试的对用关系

六、 综合题（1×20=20 分）

解：第一步：划分等价类，包括 4 个有效等价类，11 个无效等价类。

第二步：确定调试用例。

对 11 个无效等价类，要选择 11 个调试用例，如下所示：

选取的调试数据可以不同，关键是与调试内容相符。



## 目录

## 一、开场白

Q: 简单自我介绍一下吧

Q: 项目和实习经历？（注意使用 [STAR 法则表述](#)）



## 二、软件测试基础

工作内容

Q: 按测试内容划分，测试有哪些种类？

Q: 软件开发（测试）的流程是怎么样的？

Q: 你印象最深刻的

bug 是？

Q: 谈谈你对

CI/CD

的理解

Q: 谈谈你对

DevOps 的理解

Q: 什么是 BDD？什么是 TDD？

Q: APP 的一个页面，你怎么区分是原生 Native 页面，还是 H5?

#### 测试方法

Q: 黑盒测试的方法有哪些?

Q: 白盒测试的方法有哪些?

Q: 什么是单元测试?

Q: 什么是集成测试?

Q: 测试用例怎么编写与设计?

Q: 什么是灰盒测试?

#### 测试文档

Q: 测试用例都包含哪些要素?

Q: 测试报告需要展示哪些要素?

Q: 测试排期应该怎么估算?

Q: 谈谈你构造数据的经历?

#### 自动化测试

Q: 举例说明，都有哪些功能可以用自动化来进行?

Q: 如何判断一个功能能否进行自动化测试?

Q: 定位 xpath 路径都有哪些方法?

Q: 如何定位一个动态的元素?

Q: 如何定位类似于悬浮在 web 页面上的元素（类似于 web 上飞来飞去的广告）

Q: 列举你知道的自动化测试工具

#### 测试工具

Q: 聊聊 fiddler 的抓包原理

Q: fiddler 怎么抓取 https 请求?

Q: wireshark 抓包的头部有什么?

## 性能测试

Q：性能测试如何做的？

Q：性能测试需要关注哪些方面？

## 游戏测试

Q：印象最深的游戏是哪个？哪些方面最吸引你？

Q：印象最深的游戏里有过哪些 bug，分析下原因

Q：如何对一个大地图游戏进行测试？该有哪些测试点？

Q：测试程序设计的不合理 与 测试程序完善但是测试人员出现失误，哪一项更严重？

Q：游戏测试的定位和职责你觉得是什么？

Q：游戏采用客户端和服务端架构的必要性是什么？

Q：LOL 脚本的原理是什么？

Q：为什么 RPG 游戏脚本没有 LOL 这种竞技类脚本开发的多？

Q：有一个触发概率极低但是非常致命的 BUG，明天项目一定要上线，你如何处理，不用非得解决，就是问我怎么处理这种事。

Q：为什么想做游戏测试？

Q：让你做一个游戏，你更想做哪个类型的（MMORPG？MOBA？

FPS？RTS？）

Q：说说游戏史，除了游戏还有什么爱好？

Q：列举你知道的游戏测试的测试工具

## 三、软件测试实战

### 排查问题的思路

Q：网页崩溃的原因是什么？

Q：有个用户反馈上传头像失败，分析原因？

Q: app 闪退的原因?

Q: 偶然闪退的排查?

Q: 网页卡顿的原因是什么?

Q: 10%的用户反馈用不了功能，你讲如何排查?

Q: 登录的按钮不能点击，如何排查问题?

Q: 压测的时候，QPS 一直上不去，你会怎么排查?

Q: APP 提示无法连接网络，你会如何排查?

Q: 怎么判断一个 BUG 到底是前端的 BUG 还是后端的 BUG?

### 实战案例

Q: 微博发动态，设计一下测试点

Q: 对一台自动售货机进行测试用例设计

Q: 设计微信发红包测试用例

Q: 设计抖音直播功能测试用例

Q: 设计微信扫码支付的测试用例

Q: 设计百度首页的测试用例

Q: 微信的点赞功能怎么测试?

Q: 微信红包是先计算每个人能获得的钱还是当这个人点了再计算。

Q: 微信朋友圈评论功能怎么测试?

Q: 微信上线一个新的好友推荐功能功能如何测试?

Q: 测试微信换头像功能，设计测试用例

Q: 抖音视频的安全性测试，测试点有哪些?

Q: 如果手机浏览器输入 <http://baidu.com> 打不开页面，你会怎么

排查？

Q：设计输入框测试用例？

Q：编写一个登录界面的测试用例？

Q：对一个接口编写测试用例

Q：搜索功能怎么测试？

#### 四、语言基础

##### Java

Q：什么是多态？

Q：什么是 GC？

Q：set 与 hashset 的区别

Q：map 与 hashmap 的区别

Q：hashmap 的查找，删除，添加的时间复杂度

Q：查找所有 java 进程

Q：杀死 java 进程

Q：介绍一下 java 的拷贝、四种引用

Q：java 容器有哪些？简要介绍一下 java 的数组和链表

Q：谈谈你对 java 集合了解多少？

Q：Java 三大特性是什么？

Q：List 和 Map 有什么区别？

Q：谈谈 List、Set、Map 的区别？

Q：重写与重载的区别是什么？

Q：HashMap 和 Hashtable 的区别是什么？

Q：ArrayList 和 LinkedList 的区别是什么？



Q: Map 的 put 方法

Q: java 的封装, java 的访问权限关键字的作用域?

Q: 为什么 java 要面向接口编程?

Q: 什么是反射?

Q: interface 和

abstract 的区别是什么?

Q: 熟悉 maven

么? 谈谈你对 maven 的了解

## Python

Q: 全局变量和局部变量变量名能否一样?

Q: Python

里 is 和 == 的区别?

Q: Python

变量的创建与消亡过程

Q: Python 的垃圾回收的机制

Q: dict 的底层结构, tuple 和 list 的底层结构的区别

Q: 深拷贝浅拷贝的区别是什么?

Q: 什么是协程?

Q: 什么是装饰器? 举一个你用过装饰器的例子

Q: @classmethod 和 @staticmethod 的区别, 以及分别运用在哪些使用场景?

Q: 什么是鸭子类型?

Q: python

的可变类型有哪些？

Q: python

常见的数据类型有哪些？

Q: 列举 python2 和 python3 的区别

Q: 什么是 lambda 函数？怎么用？

Q: 别的编程语言都有三目运算符，三目运算符在 python 中怎么表达？

Q: try...except...else 和 try...except...finally 的区别是什么？

Q: 什么是可迭代对象？可迭代对象的原理是什么？

Q: with...as 的原理是什么？

Q: 解释一下 python 的 GIL 锁

Q: python

是单继承还是多继承？

Q: python

继承的顺序是什么？

Q: 什么是元类？

Q: 为什么都说 python 慢？

## Shell

Q: 什么场景下，适合编写 Shell 脚本来处理？

## C++

Q: 指针和引用的区别是什么？

## 五、数据结构与算法

### 链表

Q: 一个有序链表，怎么求第 K 大个节点？

Q: 怎么找出这两个链表是否有相交的点

Q: 数组与链表的区别

Q: 链表逆序

### 数组

Q: 怎么对俩有序数组合并？

Q: 求数组中和为  $n$  的两个数，时间复杂度是多少？

Q: `int` 型数组，怎么排序？

Q: `int` 型数组，怎么去重？

### 复杂度

Q: 时间复杂度是什么？

Q: 怎么计算时间复杂度？

### 排序

Q: 八大排序都有哪些？

Q: 快排的原理

Q: 归并排序的原理

Q: 冒泡排序的原理

Q: 插入排序的原理

Q: 选择排序的原理

### 树

Q: 数的定义，代码实现

Q: 什么是树的高度？怎么求？

Q: 计算二叉树节点的个数

Q: 什么是根结点？什么是叶子节点？

Q: 打印二叉树

递归

Q: 什么是递归?

Q: 递归的时间复杂度是什么?

字符串

Q: 字符串长度可以改变么?

堆与栈

Q: 堆和栈有什么区别 Q: 代码实现栈

动态规划

Q: 什么是动态规划?

高级算法

Q: 什么是深度遍历? 什么是广度遍历?

查找

Q: 能够实现二分查找的必要条件是什么?

哈希

待施工(善用搜索引擎)

图

待施工(善用搜索引擎)

## 六、框架

开发框架

**Django**

Q: 简单介绍一下 Django 的目录结构

Q: 了解什么是 MVC 么? 为什么说 Django 是 MTV 模型?

Q: 一个网络请求在 Django 中的实现过程

Q: uwsgi 是什么? 用 uwsgi 和用 Django 的原生启动方式, 有什么区别?

Q: 了解 ORM 么? 简单介绍一下 Django 的 ORM 映射

Q: Django 的 CBV

和

FBV 的区别？实际项目中你会使用哪种方式？为什么？

Q: 怎么把 Django

的

module 同步到数据库中？同步过程中会遇到什么坑么？

**Spring**

Q: Spring

框架介绍

Q: 什么是

IOC？

Q: 什么是

DI？

Q: 什么是 AOP？

测试框架

**selenium**

Q: selenium 框架的运行原理

Q: selenium 定位元素的方法都有哪些

**Appium**

Q: Appium 用过吗？原理是什么？

**Unittest**

待施工

**Pytest**

待施工

**TestNG**

待施工

**Junit**

待施工

七、计算机基础

## 计算机网络

### TCP/UDP

Q: 简单介绍 TCP 三次握手（为什么不是两次、四次）

Q: 四次分手是什么？

Q: TCP 拥塞是什么？

Q: TCP 怎么保证安全的，UDP 能否也像 TCP 那样安全，怎么做？

Q: 你知道 tcp 的控制可靠性的策略和重传机制么？

Q: TCP 协议属于哪一层？

### HTTP/HTTPS

Q: cookie 和 session 机制、区别

Q: 输入 url 到网页显示出来中间的过程

Q: 列举你知道的网页状态码

Q: 3

开头的网络状态码的含义是什么？302 和 304 的区别是什么？

Q: 4

开头的网络状态码含义是什么？

Q: 5

开头的网络状态码含义是什么？

Q: 什么是

HTTPS? 原理是什么？

Q: GET 和

POST 的区别是什么？

Q: 网络请求 method 有哪几种？

Q: 简单介绍一下什么是 RESTful API

Q: PUT 和 POST 的区别

Q: 列举常见的请求 Header 头

Q: 一个 HTTP 请求报文是什么样的？（GET 举例）

Q: GET 的长度限制了解么？

其他

Q: 最大连接数和 QPS 区别

Q: 网络一共分为几层？

Q: 客户端向服务器请求图片和动态资源的区别

Q: 图片渲染的过程 前端

Q: 网络七层有哪些？tcp, udp, arp 都在哪一层？

Q: QPS 和 TPS 的区别是什么？

Q: 解释一下 DNS

Q: 什么是反向代理？

Q: 什么是 socket？

Q: 【手撕】用 socket 实现一个聊天室功能？

操作系统

进程/线程

Q: 进程与线程的区别？

Q: 何时 cpu 处理进程最慢？

Q: 为什么会出现死锁？

Q: 进程间通信的方式？

Q: 线程间通信的方式？

Q: 如何做到线程同步？

## Linux

Q: linux 切换目录

Q: linux 命令，统计一个文本中关键字出现的次数

Q: linux 查找当前目录下所有后缀为 .py 文件

Q: 知道的 linux 常用命令：查看指定端口进程

Q: cd - 和 cd ~

Q: linux 查看某个进程命令怎么写

Q: 如何查看日志？怎么查看后 500 条日志？

Q: awk 有什么用？如何用（举个例子）？

Q: 如何查看系统性能？

Q: 如何查看剩余磁盘空间大小？

Q: 如何查看目录占空间大小？

Q: 你知道 xarg 的用法吗？

Q: 怎么从本地计算机与服务器中进行文件传输？

Q: 测试服务器之间怎么进行文件拷贝？

Q: sed 有什么用？如何用（举个例子）？

Q: 怎么杀死一个进程？

Q: 怎么递归删除一个目录下的所有文件？

Q: 怎么查看内存大小？

Q: 怎么查看

CPU 使用情况？

Q: 怎么重启

Linux 服务器？



Q: 怎么打印出一个文件的第 500-1000 行?

文件存储

Q: 静态存储和动态存储的区别

Q: 视频在服务器的存储几种方式

Q: CDN 有什么用?

Q: 为什么前端静态资源要上传到 CDN 上?

其他

Q: 系统资源包括哪些?

Q: 什么是 IO 操作?

Q: 什么是内存?

Q: 什么是硬盘?

Q: 什么是 CPU?

Q: 什么操作比较消耗 CPU?

Q: 什么是 UTF-8? 什么是 Unicode?

Q: 什么是 IO 多路复用? 以及怎么实现?

Q: 谈谈什么是分布式? 为什么要用分布式?

## 数据库

非关系型数据库

Q: 非关系型数据库有哪些?

Q: Redis 的数据结构有哪些?

Q: Redis

和 Memcached 的区别?

Q: Redis

的用途？

Q: Redis 一秒能写入多少数据？

Q: Redis 为什么快？

Q: Redis 的过期时间怎么设置？哪些场景适合缓存更长时间？

Q: 什么是缓存雪崩？

Q: 什么是缓存击穿？缓存击穿和缓存穿透的区别是什么？

Q: 什么时候适合用 MongoDB？

Q: Redis

适合做消息队列吗？为什么？

Q: Redis

的数据是存储在内存当中的，假如断电之后就会造成数据丢失，那怎么对 Redis 进行数据固化？

关系型数据库

数据库基本理论

Q: 有哪些数据库优化的方式？

Q: 关系型数据库和非关系型数据库的区别？

Q: 数据库的事务有什么用？什么时候应该使用事务，什么时候不应该使用事务？

Q: 数据库索引有什么用？什么是联合索引？

Q: 主键适合用自定义 ID 还是自动 ID？

Q: 外键是什么？为什么大公司有时候不建议使用外键？

Q: 索引设置得越多越好吗？索引得优缺点是什么？

Q: InnoDB 是基于什么实现的？

Q: 为什么会造成数据库死锁？怎么解决？

Q: 事务都有哪些特点？

Q: 数据库设计的三大范式是什么？

Q: 一张数据表最多不建议超过多少行？

Q: 为什么要进行分表分库？

Q: 分表分库有哪几种方式？举例说明

## SQL

Q: 【手撕】数据库，查找一个学生两门功课都大于 80 分的姓名

Q: 【手撕】联表查询 2 个表中工号为“123”的人的所有信息

Q: 【手撕】一个人员表，一个部门表，人员表中存了部门 id，查人员表各部门表所有数据

Q: 【手撕】查询一个城市列表里面重复的城市名，并且统计重复次数

Q: 【手撕】查找一个学生成绩表中平均分数大于 90 分的学生名单

Q: 【手撕】查找学生成绩表中平均成绩最高的同学

Q: 数据库怎么拷贝数据？

Q: 删除数据的方式有哪些？说说 drop table 和 truncate table 的区别？

## 八、智力题

Q: 跳台阶问题

Q: 4 分钟沙漏和 7 分钟沙漏怎么漏出 9 分钟

Q: 两个粗细不同的香，燃尽时间都是 1 个小时，怎么用这个 2 根香计算 15 分钟的时间

Q: 赛马

Q: 10 堆苹果，每堆 10 个，9 堆每个 50g，1 堆每个 40g，有一个称，求只称一次，找出这个轻的一堆

Q: 飞机加油问题

Q: 逻辑：四个开关四个灯泡

Q: 地球弧形

## 九、编程题

Q: 求最大回文个数

Q: 一个数组中有正数有负数（没有 0），请将它排成正负相间的数组（多余的全部放后面），时间复杂度不超过  $O(n)$ ；

Q: 一道编程题，输入一串由 ABCD 四个字随机组成的字符串和一个整数 k，返回字符串种前 k 个字的顺序重复了几次。

Q: 编程：判断一个字符串是否符合 ipv4 格式

Q: 代码题，给一个句子，只把单词翻转然后输入

Q: 【手撕代码】字符串中只出现一次的字符、找出数组中最小的四个数字

Q: 代码：一串字符串中最小的整数

Q: 写代码，类似高考成绩，一个表中有很多数据（无序的），给你一个成绩，查出在表中的排名

Q: 编程题，鸡兔同笼，一半的兔子伸起一半的脚，输入地上有几只脚，列出所有的可能性（兔子是基数则整除 2）

Q: 判断一个字符串是否是点分十进制的 ipv4 格式

Q: 100 万个学生 按照成绩 及对应排名录入 分数查找排名

(hashmap)

Q: python 的编程题，输入一个字符串然后空格切割在统计每个字母出现的次数

## 十、HR 常问

Q: 为什么想做测试

Q: 对测开的理解

Q: 测试过程中有没有出现问题，是如何解决的

Q: 最近看了什么书？学了什么？为什么学？有看什么技术书籍吗？

Q: 个人优缺点，举例

Q: 测试看重什么能力

Q: 项目问题细挖

Q: 为什么选择 xx 公司？

Q: 你对我们公司有什么了解吗？

Q: 之前实习收获了什么

Q: 介绍下自己的优缺点

Q: 抗压能力如何，描述一件自己如何抗压的经历

Q: 反问环节：你有什么问题想问我么？

Q: 项目中收获了什么？

Q: 平时怎么学习的

Q: 为什么要离职？

Q: 你的期望薪资是多少？

## 软件测试面试问题

本文收录软件测试面试过程中常见的面试题.一些问题是从网上搜罗而来,剔除了不合时宜的;一些则是自己总结的面试题.很多的问题是开放性的,并没有确切的标准答案.

### 常见问题

- 软件测试的目的是什么?
- 软件测试的一般流程是怎么样的?
- 常见的测试类型有哪些? 分别说明一下?
- 测试用例设计常用的方法有哪些?详细说明一下?
- 解释下单元测试,集成测试,系统测试以及验收测试?
- 探索性测试是什么? 应该怎么做?
- 什么是冒烟测试,如何有效的开展冒烟测试?
- 一条高质量的缺陷记录(Bug)应该具有哪些内容?
- 缺陷的生命周期是怎样的?
- Alpha 测试与 Beta 测试的区别?
- 你认为做好软件测试应该具备哪些素质?
- 作为测试人员,在与开发人员沟通过程中,如何有效的提高沟通效率和效果?
- 你觉得软件测试工程师在一个团队中,都需要做什么? 有什么价值?
- 你对软件测试最大的兴趣是什么?
- 你对自己的职业规划是什么?
- 在你以往的工作中,发现的影响大或印象深刻的 Bug 是什么? 为什么?
-

在你以往的经历中,解决过的最困难的问题是什么?

•

在你以往的工作或学习中,你最大的收获是什么?学到了什么?

•

你认为做好软件测试应该具备哪些素质?

•

在没有任何文档的情况下,你如何开展测试?

## 测试用例设计问题

•

测试用例是什么? 如何设计有效的测试用例?

•

输入三个整数,判断是否构成有效的三角形,针对这个设计测试用例

•

针对文件上传功能, 设计下测试用例

•

针对网上购物中订单提交的过程, 设计测试用例

## 测试管理问题

•

你认为测试经理的工作职责和内容是什么?

•

如果你作为测试 Leader,你应该怎么建立公司的测试体系并实施它?

•

说明你作为测试团队的负责人, 如何提高测试团队的技术能力?

•

列举你以往项目测试中遇到的风险以及你如何处理的?

•

如果当时间不充裕时,该如何安排测试?

•

列举你曾经做过的测试(你认为有技术含量的或者提高了测试管理能力的),并说下你从中如何受益?

•

在开发和测试存在不合作甚至对立的情况下, 你如何平衡和协调工作?

## 自动化测试问题

•

你认为适合做自动化测试的标准是什么?

•

你认为什么类型的测试不适合做自动化测试?

•

UI 自动化测试的优点和缺点分别是什么?

- 
- 在一个项目中目前还没有进行自动化，如果我想开展自动化测试，我应该怎么做(一般步骤)?
- 
- 你认为该如何选择最适合的自动化测试工具?
- 
- 什么是自动化测试框架?一个好的自动化测试框架应该具备什么元素?
- 
- 说一下你写过的测试框架的代码目录结构是怎么样的?
- 
- 自动化测试框架的类型有哪些?
- 
- 说一下你在实施自动化测试过程中好的代码实践?
- 
- 自动化测试是否仅仅可以是实施在 UI 层?为什么?
- 
- 你是否熟悉 Selenium 工具?说一下它是什么?
- 
- 你是否还熟悉其它的自动化测试工具?各自简单的介绍一下?

## 性能测试问题

- 
- 什么是性能测试?为什么要进行性能测试?
- 
- 性能测试的类型有哪些?
- 
- 列举下用户会面对的性能问题和性能瓶颈?
- 
- 列举下性能测试中常涉及的性能计数?
- 
- 性能测试中并发用户点击量是什么?如何实现?
- 
- 性能测试进入和结束的标准是什么?
- 
- 说一下在选择性能测试工具之前需要考虑哪些东西?
- 
- 在性能测试中,如何识别性能瓶颈?
- 
- 在对应用程序进行性能测试期间一般会执行哪些活动?
- 
- 性能测试中吞吐量是什么?



- 解释下什么是耐力测试和尖峰测试?

## 数据库问题

### MySql

- 一张表,里面有 ID 自增主键,当 insert 了 17 条记录之后,删除了第 15,16,17 条记录,再把 Mysql 重启,再 insert 一条记录,这条记录的 ID 是 18 还是 15

## 操作系统问题

- 列出超过 10 个 Linux 常用的命令以及其作用?
- 进程和线程是什么?它们有什么区别和联系?

这里收集了软件测试面试过程中的经典题目，欢迎投稿！  
读者请根据自己的实际情况对列出的题目进行取舍，不一定要全部掌握。  
大部分的题目没有完全标准的答案，合理即可。

## 面试题-one

01、您所熟悉的测试用例设计方法都有哪些？请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

答：有黑盒和白盒两种测试种类，黑盒有等价类划分方法

边界值分析方法

错误推测方法

因果图方法

判定表驱动分析方法

正交实验设计方法

功能图分析方法

场景设计方法。

白盒有逻辑覆盖法，

循环测试路径选择，

基本路径测试。

例子：在一次输入多个条件的完整性查询中。利用等价类划分法则和边界分析法则，首先利用等价划分法，可以一个或多个结果是 OK 的测试用例，然后确认多个 NG 的测试用例，然后利用边界值分析法，可以对结果分别是 OK 和 NG 的测试用例进行扩展和补充。

## 02、您认为做好测试用例设计工作的关键是什么？

答：测试用例设计工作的关键是对可行的和不可行的都要考虑。

1，输入 2，详细的操作步骤 3，预期输出 4，实际输出。

## 03、您在从事性能测试工作时，是否使用过一些测试工具？如果有，请试述该工具的工作原理，并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。

答：有使用过 LoadRunner，该工具能够录制测试人员的操作步骤，然后对这个操作步骤模拟出多个用户来播放出来。

1、Visual User Genertor 创建脚本，选择协议，录制操作，编辑操作。

2、中央控制器（Controller）调度虚拟用户，创建场景，选择脚本，建立虚拟用户，设计 shedual，设置 ip spoofer。

3、运行脚本。分析 shedual。

4、分析测试结果。

## 04、您认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？

答：性能测试工作的目的是检查系统是否满足在需求说明书中规定的性能，性能测试常常需要和强度测试结合起来，并常常要求同时进行软件和硬件的检测。

性能测试主要的关注对象是响应时间，吞吐量，占用内存大小（辅助存储区），处理精度等。

## 05、在您以往的工作中，一条软件缺陷（或者叫 Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？

答：检测时间，系统环境，硬件环境，严重程度，程式版本，确认人，功能模板，问题描述，详细操作步骤，是否会重现。

问题描述和详细操作步骤要尽可能详细。Bug 应该尽量用书面语，对于严重程度比较高的缺陷要在相同环境下测试一遍。

在 C\S 模式下，如果条件满足可以使用替换法来确认是 client 端的问题还是 server 端的问题。

## 06、你对测试最大的兴趣在哪里？为什么？

答：最大的兴趣就是具有挑战性。

因为我并不知道哪里会出现 bug，在找到一个 bug 后会很高兴。并且测试需要很强的耐心和细心。我可以很容易的找到一些细节问题。

## 07、测试活动中，如果发现需要文档不完善或者不准确，怎么处理？

答：要及时的与项目经理进行沟通协调。要在邮件中详细的把不完善不准确的地方描述出来，并提出自己的意见。

## 08、你认为做好测试计划工作的关键是什么？

答：首先，要有一个明确的目标，详细的阅读需求文档说明。

其次，要对整个测试人员、测试时间、测试进度进行一个预估，并预先进行管理。

最后，要对整个测试流程设定一个规范，所有测试人员都按着规范做事，不能随心所欲的测试。

#### 09、软件配置管理工作开展的情况和认识？

拿到一台裸机过后要安装客户需要的操作系统，并且安装一些所必须的软件。

#### 10、你觉得软件测试通过的标准应该是什么样的？

答：测试用例完全执行，测试用例覆盖到所有的测试点，并且缺陷的密度达到客户的需求。

#### 11、软件测试的文档测试应当贯穿于软件生命周期的全过程，其中用户文档是文档测试的重点。那么软件系统的用户文档包括哪些？

答：用户安装文档、用户配置文档、用户使用手册、联机指导等。

#### 12、简述软件系统中用户文档的测试要点？

完整性：用户文档中功能的描述要完整的。不能让用户产生疑问。

一致性：用户文档中的功能描述要与实际软件中的功能一致。不能描述过盛。

易使用性：用户文档描述的内容要方便用户阅读并且能够让用户很清楚的知道如何操作。

图表：有的时候用图表描述会很明了。

#### 13、什么是系统瓶颈？

系统瓶颈就是软件在一定的并发量、访问量下无法达到用户的需求。

比如说用户需要在 10s 内完成一个访问，但是每一次都要 12s 才能完成，这个就是性能瓶颈，有可能是程序本身的问题，也有可能和操作系统、软件相关。

#### 14、没有产品说明书和需求文档地情况下能够进行黑盒测试吗？

可以。

这个情况下我们就要进行探索性测试，把软件当成用户需求，一步步进行测试。凭借经验判断功能正确与否，有的时候还可以与项目经理、开发人员一起进行交流沟通，从而进行更好的测试。

#### 15、为什么尽量不要让时间富裕的员工去做一些测试？

首先，专业的测试人员是有一定的技能和耐心对软件一步一步进行测试。如果让时间充裕的员工去测试的话，他可能心思并不在测试上面。会很随意的、没有目标的进行测试。这样的话测试并不完整，有的时候甚至很重要的 bug 都没法找出。所以还是需要专业的测试人员来进行测试的。

#### 16、完全测试程序是可能的吗？

不可能

测试人员对程序进行测试，只能找出程序中的 bug，但是并不能保证程序是没有 bug 的。

完全的测试要花费很多的人力财力，并且测试的数据量过大，很浪费时间。测试的结果还很多，有的都是类似的，没有必要进行相同的测试。所以完全测试

是不可能的。

#### 18、软件测试的风险主要体现在哪里？

主要体现在没法完全测试。有些问题可能隐藏在没有测到的地方。这样子就被忽略了。客户使用的时候并不熟悉软件是如何操作的。可能有的时候会误点点出问题。这样的话我们就要承担很大的风险了。

发现的缺陷越多，说明软件缺陷越多吗？

是的，通常如果发现一个缺陷的话，有的时候会发现很多类似的缺陷，因为由于开发人员的习惯，可能一个地方有错误，另外一个地方就会有相同的错误。

#### 19、所有的软件缺陷都能修复吗？所有的软件缺陷都要修复吗？

从理论上来说所有的缺陷都是可以修复的，但是并不是所有的缺陷都

要修复。

一些对于软件没有影响的、不影响使用的缺陷我们可以不用修复。因为修复些细小的缺陷也是需要花费很多时间。项目上面可能会因为时间问题而先忽略这些小缺陷。

#### 20、开发人员老是犯一些低级错误怎么解决？

要在开发的前期就制定好一些编码规范，这样子可以减少很多因为个人习惯引起的错误。同时，测试人员在发现开发人员犯一些低级错误的时候不可以指责他们，要耐心的给他们指出错误所在。然后可以有开发人员自己进行测试，找出一些一眼看得出来是错误的地方。

#### 21、您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

我一般都是做的 Web 测试，搭建测试环境，对于一个程序进行集成测试，系统测试，回归测试等。还要编写测试用例以及一些文档，用户使用手册，功能测试文档等等。最擅长的是功能测试。

#### 22、开发人员说不是 bug 时，你如何应付？

首先把自己的理由告诉开发人员。在同开发人员沟通到底是不是 bug，但是如果开发人员还是认为不是 bug 的话，就把这个问题提到项目经理处，同时附上自己的理由。有项目经理决定是否为 bug。

#### 23、软件测试项目从什么时候开始为什么？

一般软件测试越早展开越好，一般是从需要阶段就要进行软件测试。软件测试不仅是测试功能，对于需求文档一类的也要进行测试。越早的找出 bug，就会减少后续开发人员修改程序的次数，并且可以降低成本，如果等整个软件开发的差不多了发现一个致命的错误的话，是需要花费很多时间和人力来重新修改的。如果在一开始就发现的话就不会出现这种情况了。

#### 24、你能不能说下你的 3-5 年的职业规划？

首先，要巩固自己的测试基础知识，在基本知识扎实的情况下提高理解需求文档地能力。

其次，学习自动化测试工具，并将它运用到测试中。

然后，在测试技术达到一定程度后，要学会如何带领一个测试团队。

最后，争取在最快的时间内达到测试经理的水平。

**25、功能测试用例需要详细到什么程度才是合格的？**

测试用例覆盖到所有的测试点。

**26、一个缺陷测试报告的组成？**

缺陷编号、缺陷标题、缺陷描述、缺陷的优先级、缺陷的重要程度、缺陷所述的模块、缺陷所属的版本、缺陷所属的开发人员、输入数据、输出结果、缺陷分析等。

**27、测试用例通常包括哪些内容？**

用例编号、测试环境、用例标题、输入数据、预期结果等

**28、你都用什么测试方法？**

根据不同的系统和模块有不同的方法。主要是黑盒测试和白盒测试。

**29、软件的评审一般由哪些人员参加？其目的是什么？**

参加人员：客户、项目经理、开发人员、测试人员

目的：查看软件在未正式投入运行前是否还存在问题。对于不同软硬件平台能否正常运行，是否有与客户理解不一致的地方，同时可以对一些可以改进的地方再多加改进。

**30、什么是软件测试，软件测试的目的？**

软件测试是通过人工或者自动化的操作进行还没有商业化用途的程序，查看他们的功能是否满足客户需求。

目的：在最短时间内找出尽可能多的软件缺陷。

**31、什么是兼容性测试？**

兼容性测试是检查软件在不同软件平台，硬件平台上是否可以正常运行的测试。主要查看软件在不同操作系统、浏览器、数据库中是否运行正常。

**32、什么是软件测试？**

答：为了发现程序中的错误而执行程序的过程

**33、软件测试的对象有哪些？**

答：软件测试并不等于程序测试。软件测试应贯穿于软件定义与开发的整个期间。

需求分析、概要设计、详细设计以及程序编码等各阶段所得到的文档，包括需求规格说明、概要设计规格说明、详细设计规格说明以及源程序，都应成为软件测试的对象。

**34、当测试过程发生错误时，有哪几种解决办法？**

答：1) 跳转到别的测试过程

2) 调用一个能够清除错误的过程

3) 退出过程，启用另一个

4) 退出过程 and 应用程序，重新启动 Windows，在失败的地方重新开始测试

**35、怎么才能够全面的测试到每一个点？**

答：测试的全面性主要需要在设计测试计划的时候考虑，从测试策略，产品需求等等多个角度考虑从而定义全部的测试点。

**36、开发与测试的关系？**

答：开发和测试是一个有机的整体。在产品发布之前，开发和测试是循环进行的，测出的缺陷要经开发人员修改后继续测试。在开发的同时测试经理开始编写测试用例，测试文档要参考开发文档，所以开发和测试是不可分割的，少了



任何一个都不能开发出产品。

### 37、测试活动中统计了哪些数据？

答：工作量

bug 数量

### 38、进行测试时产生了哪些文档或记录？

答：测试的整个过程有系统测试计划、系统测试用例、系统测试报告、缺陷报告、产品发布说明

在执行测试的过程中只有缺陷报告，这个还是用在缺陷管理工具中进行的，最后在工具中导出缺陷报告

### 39、怎样做好测试计划？

答：1) 理解系统。从整个系统的高度了解被测系统必须满足的功能和非功能性需求。利用涉及整个系统的文档，形成对系统的整体了解。

2) 及早介入。为了深入了解项目，测试人员应该在系统的开始阶段介入，可以增加对客户需求，客户问题，潜在风险以及最重要的功能方面的理解

3) 测试期望。程序员的期望是什么？客户的期望是什么？销售对测试的期望又是什么？测试目标必须是绝对的，以免说不清是否达到目标。

4) 吸取教训。把以前工作中学习到的经验教训运用过来，对确定测试策略很有作用。

5) 工作量太小。完成测试需要多少工作量？需要多少人员？

6) 技术选择。系统会采取什么技术？系统会采用什么架构？这些信息有助于确定测试策略和测试工具。

7) 时间表。系统开发和测试分配的时间有多长？截止日期是什么时候？

### 40、测试用例如何设计的？

答：在测试用例的设计之前首先要仔细阅读开发的详细设计文档，充分了解产品的详细功能，不清楚的地方与开发人员进行沟通，搞懂每个功能，尽量详细到输入框、按钮等小功能，功能点清楚之后按照功能模块分类进行用例编写。在具体的用例设计中会运用到等价类边界值等黑盒测试方法

### 41、简单概述缺陷报告，并说明包括哪些项？

答：现在缺陷报告一般不再使用纸质档文档编写，而是专用测试管理工具（如 TestDirector），这样便于缺陷管理。在这些工具中，每个缺陷作为一条记录输入指定的缺陷管理系统中。

缺陷报告包括：软件名称、版本号、功能模板、缺陷编号、对应的用例编号、编写时间、编写人、测试员、预期结果、实际结果、缺陷描述、严重级别、优先级别

### 42、什么是 bug？

答：软件的 bug 指的是软件中（包括程序和文档）不符合用户需求的问题。

常见的软件 bug 分为以下三类：

没有实现的功能

完成了用户需求的功能，但是运行时会出现一些功能或性能上的问题

实现了用户不需求的多余功能

#### 43、开发人员修复缺陷后，如何保证不影响其他功能？

答：重新执行用例、看是否出现错误结果。并对周围的一些相关功能点追加新的测试用例。

#### 44、什么时候功能测试？

答：功能测试是在规定的一段时间内运行软件系统的所有功能，以验证这个软件系统有无严重错误。

#### 45、请问功能测试和性能测试的区别是什么？

答：1) 测试目的：

功能测试：检测实际软件的功能是否符合用户需求，测功能是不是全部实现，某个实现是不是有 BUG。主要为了发现以下几类错误：A、是否有不正确或遗漏的功能？B、功能实现是否满足用户需求和系统设计的隐藏需求？C、能否正确接收输入？能否正确输出结果？

性能测试：验证软件质量的三个质量特性，可靠性，正确性和效率。主要是测试产品的健壮性

2) 测试方式：

功能测试按照系用例，按照系统需求说明书和测试用例，对产品的功能一步步进行测试。找出产品功能是否全部实现

性能测试：一般都使用性能工具对产品的健壮性进行评估。通过创建场景和虚拟用户模拟真实环境，进行压力测试和负载测试。

#### 46、为什么选择测试这行？

答：它是一个新兴的行业，有发展潜力，而且很锻炼人，需要掌握更多的技能，比做开发要更全面。

## 面试题-two

### 一、问题预测

\1.

让简单介绍下自己（每次面试开场）

\2.

让说下自己会的内容

\3.

看了哪些书籍（有问到）

\4.

了解过哪些技术博客/论坛（有问到）

\5.

是否了解软件测试需要掌握哪些知识（问到类似问题）

\6.

之前面试过，觉得自己需要补充哪些？做了哪些行动？

\7.

为什么做测试，觉得自己做测试有哪些优势？（有问到）

\8.

知道哪些 Bug 系统

9. 测试用例的基本要素是？
- 二、介绍一下公司项目
- 三、技能方面
  - 1、数据库方面常识
  - 2、linux 操作
  - 3、缺陷方面（有问到）
  - 4、用例部分
  - 5、软件测试流程
  - 6、网络相关
  - 7、测试工具
  - 8、其他概念问题
- 四、你还有什么想问的吗（必答）
- 五、简历模板

## 一、问题预测

### 1. 让简单介绍下自己（这个不用说了每次面试开场）

你好，我叫 xx，来自 xx，毕业于 xx。目前有两年的功能测试经验。最近的一份工作是 xx 公司，主要参与 app 系统测试，负责 xxapp, 一款类似抖音的短视频 app 功能测试，负责过的功能模块有拍摄、上传、搜索、推荐引擎等。主要运用边界值，等价类，错误推测等常见黑盒测试方法。

### 2. 让说下自己会的内容

我熟悉软件测试基础理论和测试流程，测试方法等，有 app 测试、web 测试、接口测试经验。熟悉数据库增删改查操作，熟悉使用测试管理工具。

### 3. 看了哪些书籍（有问到）

软件测试，软件测试的艺术、软件测试实用教程，在我负责短视频的推荐引擎测试期间看完了项亮的《推荐系统实战》主要是推荐系统的评测部分。

### 4. 了解过哪些技术博客/论坛（有问到）

51testing 论坛，CSDN 一些博客（面试经验：面试中会问具体哪些博客），和公众号（搜狗测试、软件测试资源分享）

### 5. 是否了解软件测试需要掌握哪些知识（有问到类似问题）

软件测试基础知识，流程，测试用例方法，数据库相关知识，抓包分析，接口测试、测试工具、性能测试等。

### 6. 之前面试过，觉得自己需要补充哪些？做了哪些行动？

很多公司对性能测试和自动化测试工具有要求，由于之前的工作主要涉及的是功能测试，所以这方面的知识储备不够。不过最近我在学习这方面的知识，希望以后在工作中能深入学习。

### 7. 为什么做测试, 觉得自己做测试有哪些优势？（有问到）

我觉得我个人的性格比较适合做测试。我比较细心耐心，考虑事情比较全面，



这样对于我在设计测试用例时很有帮助，而且我能够很好的与人协调沟通，当我们测试和开发发生沟通上的矛盾时我也能很好的解决，我平常喜欢刷微博、知乎看热门评论，喜欢考究大众心理，这有助于我站在用户角度设计测试点。

#### 8. 知道哪些 Bug 系统

禅道/bugzilla 等

#### 9. 测试用例的基本要素是？

版本号，功能模块，优先级，前置条件，步骤，预期结果，实际结果等。

## 二、介绍一下公司项目

xxapp，是一款集短视频、游戏、直播、社交互动于一体的内容娱乐 APP。公司大约一个月发布一个较大的版本，需求数 20 几个-40 几个不等(用例数 xx+)，每个版本包括的需求 www\wap、后台以及客户端的需求。项目分客户端版本负责人、后台版本负责人、H5 版本负责人等，负责人牵头及落实整个测试流程。我当过的角色有 H5 活动负责人、推荐引擎版本负责人、客户端和后台系统测试人员。负责过的模块用例数大概是 500 左右。

## 三、技能方面

### 1、数据库方面常识

1 关系型数据库：把复杂的数据结构归结为简单的二元关系（即二维表格形式），通过 SQL 结构化查询语句存储数据

典型产品：

Mysql：互联网领域、大中小型网站，游戏公司，电商平台等等。体积小、速度快、成本低、开放源代码

Oracle：传统大企业、大公司、政府、金融、证券等。安全性、成本高、

1 非关系型数据库：非关系型数据库也被成为 NoSQL 数据库，

NOSQL 的本意是 “Not

Only SQL”。NOSQL 为了高性能、高并发而生

其他分类

1) 键值 (Key-Value) 存储数据库：主要是使用一个哈希表，这个表中有一个特定的键和一个指针指向特定的数据。简单、易部署、高并发

典型：Redis、Memcached

2) 列存储 (Column-oriented) 数据库：应对分布式存储的海量数据。如果我们有一个 Person 类，我们通常会一起查询他们的姓名和年龄，而不是薪资。这种情况下，姓名和年龄就会被放入一个列族中，而薪资则在另外一个列族中。

典型：Hbase

3) 面向文档数据库：数据存储的最小单位是文档

典型：Mongodb、Hive

Mongodb 一个介于关系型数据库和非关系型数据库之间的产品。高性能、易部

署、易使用，存储数据非常方便。

Hive 可以用来进行统计查询，HBase 可以用来进行实时查询

一些增删改查笔试题准备

（另起一篇）

## 2、linux 操作

linux 搭建测试环境，比如 web 系统服务搭建。

一些常见命令准备

（另起一篇）

## 3、缺陷方面（有问到）

描述一个你印象最深刻的 bug

在做上传视频的测试时，发现华为荣耀 V10 上传手机自带相机专业模式录制的视频会闪退。而 ios 上传同个视频提示合成失败。

我将手机自带相机录制的专业模式和普通模式录制的同样时长的视频发到电脑上，用格式工厂软件查看视频的不同之处，之后发现视频编码是不同的。

我继续网上查阅了视频编码方面的知识，发现 mp4 视频有几种编码，而继续测试验证发现我们的 app 上传的视频只支持 mp4 视频中的 H.264 编码格式。于是

提交了视频上传不支持非 H.264 格式的视频。并补充完善了相关用例。

（因为在公司没有查日志权限，所以其实应该先查日志）

## 4、用例部分

现场让你设计个用例，比如水杯、凳子怎么测试？

首先说明的是，遇到这样的测试题目，首先应该反问面试官，需求是什么样的，比如是测什么样的杯子。

因为设计测试用例的规则应该是根据需求分析文档设计用例，客户需求什么，就测试什么。

但是在没有需求分析文档的前提下，来设计测试用例，可以考查一个测试人员的基本功，比如考虑问题是否全面，设计测试用例的方法是否合理等。

一般是根据自己的日常经验和测试的思维来设计测试用例。在设计测试用例时一般从以下几个方面进行分析：功能测试，性能测试，界面测试，安全性测试，兼容性测试，可用性测试，可靠性测试，本地化/国际化测试。

例子（另起一篇）

## 5、软件测试流程

公司严格规范测试流程和测试文档，首先是参与需求评审，编写测试计划、测试方案、测试用例，进行测试方案及用例的测试组内部评审，外部评审。

提取部分一级用例提交研发自测，研发自测通过后开始执行一轮系统测试。

测试过程中发现并提交、跟踪问题。

问题修复后进行回归测试。

一轮测试完成后对修复包进行冒烟测试，测试通过则进行二轮测试。

二轮测试完成后会进行需求交叉测试。

完成测试编写系统测试报告提交验收测试。验收测试通过输出验收测试报告。

## 6、网络相关

网络协议，如 TCP/UDP 的区别？

(<https://www.cnblogs.com/steven520213/p/8005258.html>)

1、TCP 面向连接（如打电话要先拨号建立连接）；UDP 是无连接的，即发送数据之前不需要建立连接

2、TCP 提供可靠的服务。也就是说，通过 TCP 连接传送的数据，无差错，不丢失，不重复，且按序到达；UDP 尽最大努力交付，即不保证可靠交付

3、TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流；UDP 是面向报文的

UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）

4、每一条 TCP 连接只能是点到点的；UDP 支持一对一，一对多，多对一和多对多的交互通信

5、TCP 首部开销 20 字节；UDP 的首部开销小，只有 8 个字节

6、TCP 的逻辑通信信道是全双工的可靠信道，UDP 则是不可靠信道

三次握手与四次挥手

**三次握手通俗版：**

第一次握手：客户端要和服务端进行通信，首先要告知服务端一声，遂发出一个 SYN=1 的连接请求信号，“服务端哥哥，我想给你说说话”。

第二次握手：当服务端接收到客户端的连接请求，此时要给客户端一个确认信息，“我知道了（ACK），我这边已经准备好了，你现在能连吗（SYN）”。

第三次握手：当客户端收到了服务端的确认连接信息后，要礼貌的告知一下服务端，“好的，咱们开始联通吧（ACK）”。

到此整个建立连接的过程已经结束，接下来就是双方你一句我一句甚至同时交流传递信息的过程了。

**四次挥手断开连接通俗版：**

第一次挥手：双方交流的差不多了，此时客户端也已经结尾了，接下来要断开通信连接，所以告诉服务端“我说完了（FIN）”，此时自身形成等待结束连接的状态。

第二次挥手：服务端知道客户端已经没话说了，服务端此时还有两句话要给客户端说“我知道你说完了（ACK），我再说两句&\*...%¥”...

第三次挥手：此时客户端洗耳恭听继续处于等待结束的状态，服务器端也说完了，自身此时处于等待关闭连接的状态，并对告诉客户端，“我说完了，咱们断了吧（FIN）”。

第四次挥手：客户端收到服务端也说完了，也要告诉服务端一声（ACK），因为连接和断开要双方都按下关闭操作才能断开，客户端同时又为自己定义一个定时器，因为不知道刚才说的这句话能不能准确到达服务端（网络不稳定或者其他因素引起的网络原因）。

所以默认时间定为两个通信的最大时间之和，超出这个时间就默认服务器端已经接收到了自己的确认信息，此时客户端就关闭自身连接，服务器端一旦接收到客户端发来的确定通知就立刻关闭服务器端的连接。

到此为止双方整个通信过程就此终结。

这里要声明一下：断开链接不一定是客户端，谁都可以先发起断开指令，另

外客户端和服务端是没有固定标准的，谁先发起请求谁就是客户端。

三次握手阐述：

在第一次消息发送中，A 随机选取一个序列号作为自己的初始序号发送给 B；第二次消息 B 使用 ack 对 A 的数据包进行确认，因为已经收到了序列号为 x 的数据包，准备接收序列号为 x+1 的包，所以  $ack=x+1$ ，同时 B 告诉 A 自己的初始序列号，就是  $seq=y$ ；第三条消息 A 告诉 B 收到了 B 的确认消息并准备建立连接，A 自己此条消息的序列号是 x+1，所以  $seq=x+1$ ，而  $ack=y+1$  是表示 A 正准备接收 B 序列号为 y+1 的数据包。

四次挥手阐述：

由于 TCP 连接时全双工的，因此，每个方向都必须单独进行关闭，这一原则是当一方完成数据发送任务后，发送一个 FIN 来终止这一方向的连接，收到一个 FIN 只是意味着这一方向上没有数据流动了，即不会再收到数据了，但是在这个 TCP 连接上仍然能够发送数据，直到这一方向也发送了 FIN。首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭，上图描述的即是如此。

(1) 第一次挥手：Client 发送一个 FIN，用来关闭 Client 到 Server 的数据传  
送，Client 进入 FIN\_WAIT\_1 状态。

(2) 第二次挥手：Server 收到 FIN 后，发送一个 ACK 给 Client，确认序号为  
收到序号+1(与 SYN 相同，一个 FIN 占用一个序号)，Server 进入 CLOSE\_WAIT  
状态。

(3) 第三次挥手：Server 发送一个 FIN，用来关闭 Server 到 Client 的数据传  
送，Server 进入 LAST\_ACK 状态。

(4) 第四次挥手：Client 收到 FIN 后，Client 进入 TIME\_WAIT 状态，接着发  
送一个 ACK 给 Server，确认序号为收到序号+1，Server 进入 CLOSED 状态，  
完  
成四次挥手。

## 7、测试工具

测试工具，无非这几类：

自动化测试工具（如 QTP）

性能测试工具（如 loadrunner）

测试管理类（如 jira）

安全测试工具

渗透测试工具

## 8、其他概念问题

Beta 测试与 Alpha 测试有什么区别

### 1、Alpha 测试

Alpha 测试是由用户在开发环境下进行的测试，也可以是开发机构内部的用户在模拟实际操作环境下进行的测试。开发者坐在用户旁边，这是在开发者受控的环境下进行的测试。由开发者随时记录下错误情况和使用中的问题。

### 2、Beta 测试

Beta 测试是由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，这是在开发者无法控制的环境下进行的测试。由用户记录下遇到的所有问题，定期向开发者报告。beta 测试是一模拟真实的使用环境从而发现缺陷的一种测试

### 3、验收测试

验收测试是以用户为主的测试，软件开发和 QA 人员也应该参加，测试一般在用户所在地进行，由用户验证软件产品是否满足了所有的需求的一系列的验收测试工作。

仅限于做项目的公司，部门内部测试稳定后，根据合同中需求由发包商进行验收测试。验收测试的目的是为了以发现”未实现的需求”为目的，以评估”适合使用”为目标，该类测试的不是以发现缺陷为主要目的。

区别：两者的主要区别是测试的场所不同。

Alpha 测试是指把用户请到开发方的场所来测试，beta 测试是指在一个或多个用户的场所进行的测试。Alpha 测试的环境是受开发方控制的，用户的数量相对比较少，时间比较集中。

而 beta 测试的环境是不受开发方控制的，谁也不知道用户如何折磨软件，用户数量相对比较多，时间不集中。

一般地，alpha 测试先于 beta 测试执行。通用的软件产品需要较大规模的 beta 测试，测试周期比较长。如果产品通过了 beta 测试，那么就可以正式发行了。

Alpha 测试在系统开发接近完成时对应用系统的测试；测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成，不能由程序或测试员完成。

Beta 测试 当开发和测试根本完成时所做的测试，最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成，不能由程序员或测试员完成。

### 四、你还有什么想问的吗（必答）

我非常希望能够加入公司，所以想请问您觉得我还有哪些地方比较不足，能否给一些建议？以及是否有复试时间呢？

## 面试题-three

首先，我觉得你在软件测试面试的过程中，逻辑比较混乱的最大一个原因是，说明你没有形成一个整体的体系。

导致你说的时候很多东西都杂乱无章。

我个人认为软件测试，其实开始首先进行的是一些需求的分析工作，之后呢，



进行需求的评审，需求评审完成之后，当需求确定之后我们下来开始着手去写一些测试计划相关的内容。

等测试计划编写完成之后，针对每个人分配好的固定模块，各自去编写一些测试用例。

然后编写完成测试用例之后。

进行测试用例的评审，评审完成之后大家形成一个统一的测试用例。

之后，在这个时候开发的软件已经开发的差不多了，我们对开发的软件输出一部分冒烟测试用例，要求开发对着冒烟的测试用例全部通过之后，进行提测。

之后就是我们下来就执行相关的已经编写并且评审好的测试用例。

对于发现软件的一些 bug，督促开发去修改完成相应的 bug。

整理所提出的 bug，进行分析，对 bug 的严重程度，类型以及模块进行分析和整理，形成一份软件的测试报告。

其实，软件测试来回就这么多东西，所以，你只有理清楚整个测试的过程，你的逻辑才不会混乱，下来，面试官会问你具体的一些东西包括一些技术细节的东西。

我个人总结，面试前的准备就是一个整体的主线加细化的一个过程。

## 面试题- four

### 1、在搜索引擎中输入汉字就可以解析到对应的域名，请问如何用 LoadRunner 进行测试？

建立测试计划，确定测试标准和测试范围

•

设计典型场景的测试用例，覆盖常用业务流程和不常用的业务流程等。

•

根据测试用例，开发自动测试脚本和场景：

•

录制测试脚本：新建一个脚本（Web/HTML 协议）；点击录制按钮，在弹出的对话框的 URL 中输入”about:blank”；在打开的浏览器中进行正常操作流程后，结束录制；调试脚本并保存，可能要注意到字符集的关联。

•

设置测试场景：针对性能设置测试场景，主要判断在正常情况下，系统的平均事务响应时间是否达标；针对压力负载设置测试场景，主要判断在长时间处于满负荷或者超出系统承载能力的条件下，系统是否会崩溃；执行测试，获取测试结果，分析测试结果。

### 2、试述软件的概念和特点？软件复用的含义？构件包括哪些？

软件是计算机系统中与硬件相互依存的另一部分，与计算机系统操作有关的计算机程序、规程、规则，以及可能有的文件、文档及数据。

软件复用(SoftWare Reuse)是将已有软件的各种有关知识用于建立新的软件，以缩减软件开发和维护的花费。软件复用是提高软件生产力和质量的一种重要技术。早期的软件复用主要是代码级复用，被复用的知识专指程序，后来扩大到包括领域知识、开发经验、设计决定、体系结构、需求、设计、代码和文档

等一切有关方面。

可以被复用的软件成分一般称作可复用构件。

### 3、软件生存周期及其模型是什么？

软件生存周期（Software life cycle）又称为软件生命期，生存期。是指从形成开发软件概念起，所开发的软件使用以后，知道失去使用价值消亡为止的整个过程。一般来说，整个生存周期包括计划（定义）、开发、运行（维护）三个时期，每个时期又划分为若干个阶段。每个阶段有明确的任务。

**周期模型（典型的几种）：**

•

**瀑布模型：**

**快速原型模型：**快速原型模型允许在需求分析阶段对软件的需求进行初步而非完全的分析和定义，快速设计开发出软件系统的原型，该原型向用户展示待开发软件的全部或部分功能和性能；用户对该原型进行测试评定，给出具体改进意见以丰富细化软件需求；开发人员据此对软件进行修改完善，直至用户满意认可之后，进行软件的完整实现及测试、维护。

•

**迭代模型：**

迭代包括产生产品发布（稳定、可执行的产品版本）的全部开发活动和要使用该发布必需的所有其他外围元素。在某种程度上，开发迭代是一次完整地经过所有工作流程的过程：需求分析、设计、实施和测试工作流程。实质上，它类似小型的瀑布式项目。RUP 认为，所有的阶段都可以细分为迭代。每一次的迭代都会产生一个可以发布的产品，这个产品是最终产品的一个子集。

**生命周期阶段：**

软件计划与可行性分析

需求分析

软件设计

编码

软件测试

运行与维护

### 4、什么是软件测试？软件测试的目的与原则

在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

**软件测试的目的：**

测试是程序的执行过程，目的在于发现错误

一个成功的测试用例在于发现至今未发现的错误

一个成功的测试是发现了至今未发现的错误的测试

确保产品完成了它所承诺或公布的功能，并且用户可以访问到的功能都有明确的书面说明。

确保产品满足性能和效率的要求

确保产品是健壮的和适应用户环境的

**软件测试的原则：**

测试用例中一个必须部分是对预期输出或接过进行定义

程序员应避免测试自己编写的程序

编写软件的组织不应当测试自己编写的软件

应当彻底检查每个测试的执行结果

测试用例的编写不仅应当根据有效和预料到的输入情况，而且也应当根据无效和未预料到的输入情况

检查程序是否“未做其应该做的”仅是测试的一半，测试的另一半是检查程序是否“做了其不应该做的”

应避免测试用例用后即弃，除非软件本身就是个一次性的软件

计划测试工作时不应默许假定不会发现错误

程序某部分存在更多错误的可能性，与该部分已经发现错误的数量成正比

软件测试是一项极富创造性，极具智力的挑战性的工作。

### 5、软件配置管理的作用？软件配置包括什么？

软件配置管理（Software Configuration Management, SCM）是一种标识、组织和控制修改的技术。软件配置管理应用于整个软件工程过程。在软件建立时变更是不可避免的，而变更加剧了项目中软件开发之间的混乱。SCM 活动的目标就是为了标识变更、控制变更、确保变更正确实现并向其他有关人员报告变更。从某种角度讲，SCM 是一种标识、组织和控制修改的技术，目的是使错误降为最小并最有效地提高生产效率。

软件配置包括如下内容：配置项识别、工作空间管理、版本控制、变更控制、状态报告、配置审计

### 6、什么是软件质量？

概括地说，软件质量就是“软件与明确的和隐含的定义的需求相一致的程度”。具体地说，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准、以及所有专业开发的软件都应具有的隐含特征的程度。影响软件质量的主要因素，这些因素是从管理角度对软件质量的度量。可划分为三组，分别反应用户在使用软件产品时的三种观点。正确性、健壮性、效率、完整性、可用性、风险（产品运行）；可理解性、可维修性、灵活性、可测试性（产品修改）；可移植性、可再用性、互运行性（产品转移）。

### 7、目前主要的测试用例设计方法是什么？

**白盒测试：**逻辑覆盖、循环覆盖、基本路径覆盖

**黑盒测试：**边界值分析法、等价类划分、错误猜测法、因果图法、状态图法、测试大纲法、随机测试、场景法

### 8、软件的安全性应从哪几个方面去测试？

软件安全性测试包括程序、数据库安全性测试。根据系统安全指标不同测试策略也不同。

**用户认证安全的测试要考虑问题：**明确区分系统中不同用户权限、系统中不会出现用户冲突、系统会不会因用户的权限的改变造成混乱、用户登陆密码是否是可见、可复制、是否可以通过绝对途径登陆系统（拷贝用户登陆后的链接直接进入系统）、用户退出系统后是否删除了所有鉴权标记，是否可以使用后退键而不通过输入口令进入系统、系统网络安全的测试要考虑问题、测试采取的防护措施是否正确装配好，有关系统的补丁是否打上、模拟非授权攻击，看防护系统是否坚固、采用成熟的网络漏洞检查工具检查系统相关漏洞（即用最专业的黑客攻击工具攻击试一下，现在最常用的是 NBSI 系列和 IPacker IP）、采用各种木马检查工具检查系统木马情况、采用各种防外挂工具检查系统各组程序的外挂漏洞。



**数据库安全考虑问题：** 系统数据是否机密（比如对银行系统，这一点就特别重要，一般的网站就没有太高要求）、系统数据的完整性（我刚刚结束的企业实名核查服务系统中就曾存在数据的不完整，对于这个系统的功能实现有了障碍）、系统数据可管理性、系统数据的独立性、系统数据可备份和恢复能力

（数据备份是否完整，可否恢复，恢复是否可以完整）

## 9、什么是测试用例 什么是测试脚本 两者的关系是什么？

为实施测试而向被测试系统提供的输入数据、操作或各种环境设置以及期望结果的一个特定的集合。

测试脚本是为了进行自动化测试而编写的脚本。

测试脚本的编写必须对应相应的测试用例。

## 10、简述什么是静态测试、动态测试、黑盒测试、白盒测试、 $\alpha$ 测试 $\beta$ 测试

- 静态测试是不运行程序本身而寻找程序代码中可能存在的错误或评估程序代码的过程。

- 动态测试是实际运行被测程序，输入相应的测试实例，检查运行结果与预期结果的差异，判定执行结果是否符合要求，从而检验程序的正确性、可靠性和有效性，并分析系统运行效率和健壮性等性能。

- 黑盒测试一般用来确认软件功能的正确性和可操作性，目的是检测软件的各个功能是否能得以实现，把被测试的程序当作一个黑盒，不考虑其内部结构，在知道该程序的输入和输出之间的关系或程序功能的情况下，依靠软件规格说明书来确定测试用例和推断测试结果的正确性。

- 白盒测试根据软件内部的逻辑结构分析来进行测试，是基于代码的测试，测试人员通过阅读程序代码或者通过使用开发工具中的单步调试来判断软件的质量，一般黑盒测试由项目经理在程序员开发中来实现。

- $\alpha$  测试是由一个用户在开发环境下进行的测试，也可以是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。

## 11、软件质量保证体系是什么 国家标准中与质量保证管理相关的几个标准是什么？他们的编号和全称是什么？

SQA 由一套软件工程过程和方法组成，以保证（软件的）质量。SQA 贯穿整个软件开发过程，（它）应包括需求文档评审、代码控制、代码评审、变更管理、配置管理、版本管理和软件测试。

软件质量保证（SQA-Software Quality Assurance）是建立一套有计划，有系统的方法，来向管理层保证拟定出的标准、步骤、实践和方法能够正确地所有项目所采用。软件质量保证的目的是使软件过程对于管理人员来说是可见的。

它通过对软件产品和活动进行评审和审计来验证软件是合乎标准的。软件质量

保证组在项目开始时就一起参与建立计划、标准和过程。这些将使软件项目满足机构方针的要求。

### 12、软件产品质量特性是什么？

功能性：适应性、准确性、互操作性、依从性、安全性。

可靠性：成熟性、容错性、易恢复性。

可使用性：易理解性、易学习性、易操作性。

效率：时间特性、资源特性。

可维护性：易分析性、易变更性、稳定性、易测试性。

可移植性：适应性、易安装性、遵循性、易替换性

### 13、软件测试的策略是什么？

软件测试策略：在一定的软件测试标准、测试规范的指导下，依据测试项目的特定环境约束而规定的软件测试的原则、方式、方法的集合。

### 14、软件测试分为几个阶段 各阶段的测试策略和要求是什么？

和开发过程相对应，测试过程会依次经历单元测试、集成测试、系统测试、验收测试

#### 四个主要阶段：

•

单元测试：单元测试是针对软件设计的最小单位——程序模块甚至代码段进行正确性检验的测试工作，通常由开发人员进行。

•

集成测试：集成测试是将模块按照设计要求组装起来进行测试，主要目的是发现与接口有关的问题。由于在产品提交到测试部门前，产品开发小组都要进行联合调试，因此在大部分企业中集成测试是由开发人员来完成的。

•

系统测试：系统测试是在集成测试通过后进行的，目的是充分运行系统，验证各子系统是否都能正常工作并完成设计的要求。它主要由测试部门进行，是测试部门最大最重要的一个测试，对产品的质量有重大的影响。

•

验收测试：验收测试以需求阶段的《需求规格说明书》为验收标准，测试时要求模拟实际用户的运行环境。对于实际项目可以和客户共同进行，对于产品来说就是最后一次的系统测试。测试内容为对功能模块的全面测试，尤其要进行文档测试。

单元测试测试策略：

自顶向下的单元测试策略：比孤立单元测试的成本高很多，不是单元测试的一个好的选择。

自底向上的单元测试策略：比较合理的单元测试策略，但测试周期较长。

孤立单元测试策略：最好的单元测试策略。

集成测试的测试策略：

大爆炸集成：适应于一个维护型项目或被测试系统较小

自顶向下集成：适应于产品控制结构比较清晰和稳定；高层接口变化较小；底层接口未定义或经常可能被修改；产口控制组件具有较大的技术风险，需要尽早被验证；希望尽早能看到产品的系统功能行为。

自底向上集成：适应于底层接口比较稳定；高层接口变化比较频繁；底层组件较早被完成。

基于进度的集成

优点：具有较高的并行度；能够有效缩短项目的开发进度。

缺点：桩和驱动工作量较大；有些接口测试不充分；有些测试重复和浪费。

系统测试的测试策略：

数据和数据库完整性测试；功能测试；用户界面测试；性能评测；负载测试；强度测试；容量测试；安全性和访问控制测试；故障转移和恢复测试；配置测试；安装测试；加密测试；可用性测试；版本验证测试；文档测试

**15、软件测试各个阶段通常完成什么工作？各个阶段的结果文件是什么？包括什么内容？**

**单元测试阶段：**各独立单元模块在与系统地其他部分相隔离的情况下进行测试，单元测试针对每一个程序模块进行正确性校验，检查各个程序模块是否正确地实现了规定的功能。生成单元测试报告，提交缺陷报告。

**集成测试阶段：**集成测试是在单元测试的基础上，测试在将所有的软件单元按照概要设计规格说明的要求组装成模块、子系统或系统的过程中各部分工作是否达到或实现相应技术指标及要求的活动。该阶段生成集成测试报告，提交缺陷报告。

**系统测试阶段：**将通过确认测试的软件，作为整个给予计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起，在实际运行环境下，对计算机系统进行全面的覆盖。该阶段需要提交测试总结和缺陷报告。

**16、测试人员在软件开发过程中的任务是什么？**

- 1、尽可能早的找出系统中的 Bug；
- 2、避免软件开发过程中缺陷的出现；
- 3、衡量软件的品质，保证系统的质量；
- 4、关注用户的需求，并保证系统符合用户需求。

总的目标是：确保软件的质量。

**17、在您以往的工作中，一条软件缺陷（或者叫 Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？**

一条 Bug 记录最基本应包含：

- bug 编号；
- bug 严重级别，优先级；
- bug 产生的模块；
- 首先要有 bug 摘要，阐述 bug 大体的内容；
-

bug 对应的版本；

•

bug 详细现象描述，包括一些截图、录像... 等等；

•

bug 出现时的测试环境，产生的条件即对应操作步骤；

### 高质量的 Bug 记录：

1) 通用 UI 要统一、准确

缺陷报告的 UI 要与测试的软件 UI 保持一致，便于查找定位。

2) 尽量使用业界惯用的表达术语和表达方法

使用业界惯用的表达术语和表达方法，保证表达准确，体现专业化。

3) 每条缺陷报告只包括一个缺陷

每条缺陷报告只包括一个缺陷，可以使缺陷修正者迅速定位一个缺陷，集中精力每次只修正一个缺陷。校验者每次只校验一个缺陷是否已经正确修正。

4) 不可重现的缺陷也要报告

首先缺陷报告必须展示重现缺陷的能力。不可重现的缺陷要尽力重现，若尽力之后仍不能重现，仍然要报告此缺陷，但在报告中要注明无法再现，缺陷出现的频率。

5) 明确指明缺陷类型

根据缺陷的现象，总结判断缺陷的类型。例如，即功能缺陷、界面缺陷、数据缺陷，合理化建议这是最常见的缺陷或缺陷类型，其他形式的缺陷或缺陷也从属于其中某种形式。

6) 明确指明缺陷严重等级和优先等级

时刻明确严重等级和优先等级之间的差别。高严重问题可能不值得解决，小装饰性问题可能被当作高优先级。

7) 描述 (Description)，简洁、准确，完整，揭示缺陷实质，记录缺陷或缺陷出现的位置

描述要准确反映缺陷的本质内容，简短明了。为了便于在软件缺陷管理数据库中寻找制定的测试缺陷，包含缺陷发生时的用户界面 (UI) 是个良好的习惯。例如记录对话框的标题、菜单、按钮等控件的名称。

8) 短行之间使用自动数字序号，使用相同的字体、字号、行间距

短行之间使用自动数字序号，使用相同的字体、字号、行间距，可以保证各条记录格式一致，做到规范专业。

9) 每一个步骤尽量只记录一个操作

保证简洁、条理井然，容易重复操作步骤。

10) 确认步骤完整，准确，简短

保证快速准确的重复缺陷，“完整”即没有缺漏，“准确”即步骤正确，“简短”即没有多余的步骤。

11) 根据缺陷，可选择是否进行图象捕捉

为了直观的观察缺陷或缺陷现象，通常需要附加缺陷或缺陷出现的界面，

以图片的形式作为附件附着在记录的“附件”部分。为了节省空间，又能真实反映缺陷或缺陷本质，可以捕捉缺陷或缺陷产生时的全屏幕，活动窗口和局部区域。为了迅速定位、修正缺陷或缺陷位置，通常要求附加中文对照图。

附加必要的特殊文档和个人建议和注解

如果打开某个特殊的文档而产生的缺陷或缺陷，则必须附加该文档，从而可以迅速再现缺陷或缺陷。有时，为了使缺陷或缺陷修正者进一步明确缺陷或缺陷的表现，可以附加个人的修改建议或注解。

#### 12) 检查拼写和语法缺陷

在提交每条缺陷或缺陷之前，检查拼写和语法，确保内容正确，正确的描述缺陷。

#### 13) 尽量使用短语和短句，避免复杂句型句式

软件缺陷管理数据库的目的是便于定位缺陷，因此，要求客观的描述操作步骤，不需要修饰性的词汇和复杂的句型，增强可读性。

以上概括了报告测试缺陷的规范要求，随着软件的测试要求不同，测试者经过长期测试，积累了相应的测试经验，将会逐渐养成良好的专业习惯，不断补充新的规范书写要求。此外，经常阅读、学习其他测试工程师的测试缺陷报告，结合自己以前的测试缺陷报告进行对比和思考，可以不断提高技巧。

#### 14) 缺陷描述内容

缺陷描述的内容可以包含缺陷操作步骤，实际结果和期望结果。操作步骤可以方便开发人员再现缺陷进行修正，有些开发的再现缺陷能力很差，虽然他明白你所指的缺陷，但就是无法再现特别是对系统不熟悉的新加入开发人员，介绍步骤可以方便他们再现。实际结果可以让开发明白错误是什么，期望结果可以让开发了解正确的结果应该是如何。

### 18、黑盒测试和白盒测试是软件测试的两种基本方法，请分别说明各自的优点和缺点！

黑盒测试的优点有：比较简单，不需要了解程序内部的代码及实现；与软件的内部实现无关；

从用户角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题；基于软件开发文档，所以也能知道软件实现了文档中的哪些功能；在做软件自动化测试时较为方便。

黑盒测试的缺点有：不可能覆盖所有的代码，覆盖率较低，大概只能达到总代码量的 30%；自动化测试的复用性较低。

白盒测试的优点有：帮助软件测试人员增大代码的覆盖率，提高代码的质量，发现代码中隐藏的问题。

白盒测试的缺点有：程序运行会有很多不同的路径，不可能测试所有的运行路径；测试基于代码，只能测试开发人员做的对不对，而不能知道设计是否正确，可能会漏掉一些功能需求；系统庞大时，测试开销会非常大。

### 19、如何测试一个纸杯？

功能度：用水杯装水看漏不漏；水能不能被喝到

安全性：杯子有没有毒或细菌

可靠性：杯子从不同高度落下的损坏程度

可移植性：杯子在不同的地方、温度等环境下是否都可以正常使用

兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等

易用性：杯子是否烫手、是否有防滑措施、是否方便饮用

用户文档：使用手册是否对杯子的用法、限制、使用条件等有详细描述

疲劳测试：将杯子盛上水（案例一）放 24 小时检查泄漏时间和情况；盛上汽油



（案例二）放 24 小时检查泄漏时间和情况等

压力测试：用根针并在针上面不断加重量，看压强多大时会穿透

**20、测试计划工作的目的是什么？测试计划文档的内容应该包括什么？其中哪些是最重要的？**

软件测试计划是指导测试过程的纲领性文件：

领导能够根据测试计划进行宏观调控，进行相应资源配置等

测试人员能够了解整个项目测试情况以及项目测试不同阶段的所要进行的工作等

便于其他人员了解测试人员的工作内容，进行有关配合工作

包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

**测试计划编写 6 要素（5W1H）：**

why——为什么要进行这些测试；

what——测试哪些方面，不同阶段的工作内容；

when——测试不同阶段的起止时间；

where——相应文档，缺陷的存放位置，测试环境等；

who——项目有关人员组成，安排哪些测试人员进行测试；

how——如何去做，使用哪些测试工具以及测试方法进行测试

测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。所以其中最重要的是测试策略和测试方法（最好是能先评审）。

**21、黑盒测试的测试用例常见设计方法都有哪些？请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。**

1) 等价类划分：等价类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是等效的，并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试。因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，就可以用少量代表性的测试数据，取得较好的测试结果。等价类划分可有两种不同的情况：有效等价类和无效等价类。

2) 边界值分析法：是对等价类划分方法的补充。测试工作经验告诉我，大量的错误是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。使用边界值分析方法设计测试用例，首先应确定边界情况。通常输入和输出等价类的边界，就是应着重测试的边界情况。应当选取正好等于，刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值作为测试数据。

3) 错误猜测法：基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法。

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。

4) 因果图方法：前面介绍的等价类划分方法和边界值分析方法，都是着重考虑输入条件，但未考虑输入条件之间的联系，相互组合等。考虑输入条件之间的相互组合，可能会产生一些新的情况。但要检查输入条件的组合不是一件容易的事情，即使把所有输入条件划分成等价类，他们之间的组合情况也相当多。因此必须考虑采用一种适合于描述对于多种条件的组合，相应产生多个动作的形式来考虑设计测试用例。这就需要通过因果图（逻辑模型）。因果图方法最终生成的就是判定表。它适合于检查程序输入条件的各种组合情况。

5) 正交表分析法：可能因为大量的参数的组合而引起测试用例数量上的激增，同时，这些测试用例并没有明显的优先级上的差距，而测试人员又无法完成这么多数量的测试，就可以通过正交表来进行缩减一些用例，从而达到尽量少的用例覆盖尽量大的范围的可能性。

6) 场景分析方法：指根据用户场景来模拟用户的操作步骤，这个比较类似因果图，但是可能执行的深度和可行性更好。

7) 状态图法：通过输入条件和系统需求说明得到被测系统的所有状态，通过输入条件和状态得出输出条件；通过输入条件、输出条件和状态得出被测系统的测试用例。

8) 大纲法：大纲法是一种着眼于需求的方法，为了列出各种测试条件，就将需求转换为大纲的形式。大纲表示为树状结构，在根和每个叶子结点之间存在唯一的路径。大纲中的每条路径定义了一个特定的输入条件集合，用于定义测试用例。树中叶子的数目或大纲中的路径给出了测试所有功能所需测试用例的大致数量。

## 22、详细的描述一个测试活动完整的过程。（供参考，本答案主要是瀑布模型的做法）

项目经理通过和客户的交流，完成需求文档，由开发人员和测试人员共同完成需求文档的评审，评审的内容包括：需求描述不清楚的地方和可能有明显冲突或者无法实现的功能的地方。项目经理通过综合开发人员，测试人员以及客户的意见，完成项目计划。然后 SQA 进入项目，开始进行统计和跟踪。

开发人员根据需求文档完成需求分析文档，测试人员进行评审，评审的主要内容包括是否有遗漏或双方理解不同的地方。测试人员完成测试计划文档，测试计划包括的内容上面有描述。

测试人员根据修改好的需求分析文档开始写测试用例，同时开发人员完成概要设计文档，详细设计文档。此两份文档成为测试人员撰写测试用例的补充材料。测试用例完成后，测试和开发需要进行评审。

测试人员搭建环境

开发人员提交第一个版本，可能存在未完成功能，需要说明。测试人员进行测试，发现 BUG 后提交给 BugZilla。

开发提交第二个版本，包括 Bug Fix 以及增加了部分功能，测试人员进行测试。

重复上面的工作，一般是 3-4 个版本后 BUG 数量减少，达到出货的要求。

如果有客户反馈的问题，需要测试人员协助重现并重新测试。

## 23、BUG 管理工具的跟踪过程（用 BugZilla 为例子）

测试人员发现了 BUG，提交到 Bugzilla 中，状态为 new，BUG 的接受者为开发接口人员。

开发接口将 BUG 分配给相关的模块的开发人员，状态修改为已分配，开发人员和测试确认 BUG，如果是本人的 BUG，则设置为接收；如果是别的开发人员的问題，则转发出去，由下一个开发人员来进行此行为；如果认为不是问題，则需要大家讨论并确认后，拒绝这个 BUG，然后测试人员关闭此问題。

如果开发人员接受了 BUG，并修改好以后，将 BUG 状态修改为已修复，并告知测试在哪个版本中可以测试。

测试人员在新版本中测试，如果发现问题依然存在，则拒绝验证；如果已经修复，则关闭 BUG。

**24、您认为在测试人员同开发人员的沟通过程中，如何提高沟通的效率和改善沟通的效果？维持测试人员同开发团队中其他成员良好的人际关系的关键是什么？**

尽量面对面的沟通，其次是能直接通过电话沟通，如果只能通过 Email 等非及时沟通工具的话，强调必须对特性的理解深刻以及能表达清楚。

运用一些测试管理工具如 TestDirector 进行管理也是较有效的方法，同时要注意在 TestDirector 中对 BUG 有准确的描述。

在团队中建立测试人员与开发人员良好沟通中注意以下几点：

一真诚、二是团队精神、三是在专业上有共同语言、四是要对事不对人，工作至上

当然也可以通过直接指出一些小问题，而不是进入 BUG Tracking System 来增加对方的好感。

**25、你对测试最大的兴趣在哪里？为什么？**

回答这个面试题，没有固定统一的答案，但可能是许多企业都会问到的。提供以下答案供考：

最大的兴趣，感觉这是一个有挑战性的工作；

测试是一个经验行业，工作越久越能感觉到做好测试的难度和乐趣

通过自己的工作，能使软件产品越来越完善，从中体会到乐趣

**回答此类问题注意以下几个方面：**

尽可能的切合招聘企业的技术路线来表达你的兴趣，例如该企业是数据库应用的企业，那么表示你的兴趣在数据库的测试，并且希望通过测试提升自己的数据库掌握能力。

表明你做测试的目的是为了提升能力，也是为了更好的做好测试；提升能力不是为了以后转开发或其他，除非用人企业有这样的安排。

不要过多的表达你的兴趣在招聘企业的范畴这外。比如招聘企业是做财务软件的，可是你表现出来的是对游戏软件的兴趣；或招聘是做 JAVA 开发的，而你的兴趣是在 C 类语言程序的开发。

## 面试题 -five

本系列文章总结归纳了一些软件测试工程师常见的面试题，主要来源于个人面试遇到的、网络搜集（完善）、工作日常讨论等，分为以下十个部分，供大家



参考。如有错误的地方，欢迎指正。有更多的面试题或面试中遇到的坑，也欢迎补充分享。希望大家都能找到满意的工作，共勉之！~

### 自动化测试相关

包含 Selenium、Appium 和接口测试。

\1. 自动化代码中，用到了哪些设计模式？

单例模式

- 

工厂模式

- 

P0 模式

- 

数据驱动模式

\2.

什么是断言？

- 

检查一个条件，如果它为真，就不做任何事，用例通过。如果它为假，则会抛出 AssertionError 并且包含错误信息。

\3.

UI 自动化测试中，如何做集群？

- 

Selenium Grid, 分布式执行用例

- 

Appium 使用 STF 管理多设备

- 

Docker+K8S 管理集群

\4.

怎么对含有验证码的功能进行自动化测试？

- 

万能验证码

- 

测试环境屏蔽验证

- 

其他操作不推荐

\5.

如何优化和提高 Selenium 脚本的执行速度？

- 

尽量使用 by\_css\_selector() 方法

- 

by\_css\_selector() 方法的执行速度比 by\_id() 方法的更快，因为源码

中 by\_id() 方法会被自动转成 by\_css\_selector() 方法处理；

- 使用等待时，尽量使用显示等待，少用 `sleep()`，尽量不用隐式等待；
- 尽量减少不必要的操作：可以直接访问页面的，不要通过点击操作访问；
- 并发执行测试用例：同时执行多条测试用例，降低用例间的耦合；
- 有些页面加载时间长，可以中断加载；

\6.

接口测试能发现哪些问题？

- 可以发现很多在页面上操作发现不了的 bug；
- 检查系统的异常处理能力；
- 检查系统的安全性、稳定性；
- 前端随便变，接口测好了，后端不用变；
- 可以测试并发情况，一个账号，同时（大于 2 个请求）对最后一个商品下单，或不同账号，对最后一个商品下单；
- 可以修改请求参数，突破前端页面输入限制（如金额）；

\7.

Selenium 中隐藏元素如何定位？

- 如果单纯的定位的话，隐藏元素和普通不隐藏元素定位没啥区别，用正常定位方法就行了（这个很多面试官也搞不清楚）；
- 元素的属性隐藏和显示，主要是 `type="hidden"` 和属性来控制的，  
接下来在元素属性里面让它隐藏，隐藏元素可以正常定位到，只是不能操作（定位元素和操作元素是两码事，很多初学者傻傻分不清楚），操作元素是 `click`, `clear`, `send_keys` 这些方法；

- JS 操作隐藏元素；

\8.

如何判断一个页面上元素是否存在？

- 方法一：用 `try...except...`

- 方法二：用 `elements` 定义一组元素方法，判断元素是否存在，存在返回 `True`，不存返回 `False`

- 
- 方法三：结合 WebDriverWait 和 expected\_conditions 判断（推荐）
- \9.
- 如何提高脚本的稳定性？
- 
- 不要右键复制 xpath(十万八千里那种路径，肯定不稳定)，自己写相对路径，多用 id 为节点查找；
- 
- 定位没问题，第二个影响因素那就是等待了，sleep 等待尽量少用（影响执行时间）；
- 
- 定位元素方法重新封装，结合 WebDriverWait 和 expected\_conditions 判断元素方法，自己封装一套定位元素方法；
- \10.
- 如何定位动态元素？
- 
- 动态元素有 2 种情况，一个是属性动态，比如 id 是动态的，定位时候，那就不要用 id 定位就是了；
- 
- 还有一种情况动态的，那就是这个元素一会在页面上方，一会在下方，飘忽不定的动态元素，定位方法也是一样，按 f12，根据元素属性定位（元素的 tag、name 的步伐属性是不会变的，动的只是 class 属性和 styles 属性）；
- \11.
- 如何通过子元素定位父元素
- 
- 使用 element.parent 方法
- \12.
- 平常遇到过哪些问题？如何解决的
- 
- 可以把平常遇到的元素定位的一些坑说下，然后说下为什么没定位到，比如动态 id、有 iframe、没加等待等因素；
- \13. 一个元素明明定位到了，点击无效（也没报错），如果解决？
- 
- 使用 JS 点击，Selenium 有时候点击元素是会失效；
- \14.
- 测试的数据你放在哪？
- 
- 对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，写的配置文件里（如 ini）；
- 
- 对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成；
-

对于一个接口有多组测试的参数，可以参数化，数据放 YAML, Text, JSON, Excel 都可以；

•

对于可以反复使用的数据，比如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理；

•

对于邮箱配置的一些参数，可以用 ini 配置文件；

•

对于全部是独立的接口项目，可以用数据驱动方式，用 excel/csv 管理测试的接口数据；

•

对于少量的静态数据，比如一个接口的测试数据，也就 2-3 组，可以写到

py 脚本的开头，十年八年都不会变更的；

\15.

什么是数据驱动，如何参数化？

•

参数化的思想是代码用例写好了后，不需要改代码，只需维护测试数据就可以了，并且根据不同的测试数据生成多个用例；

\16. 其他接口都需要登录接口的信息，怎么去让这个登录的接口只在其他接口调用一次？

•

使用单例模式

•

使用自定义缓存机制

•

使用测试框架中的 setup 机制

•

pytest 中 fixture 机制

\17.

接口产生的垃圾数据如何清理？

•

造数据和数据清理，需用 python 连数据库了，做增删改查的操作测试用例前置操作，setUp 做数据准备后置操作，tearDown 做数据清理

\18.

怎么用接口案例去覆盖业务逻辑？

•

考虑不同的业务场景，一个接口走过的流程是什么样的，流程的逻辑是什么样的，什么样的参数会有什么样的结果，多场景覆盖；

**性能篇**

\1.

性能测试指标包括哪些

- 最大并发用户数，HPS（点击率）、事务响应时间、每秒事务数、每秒点击量、吞吐量、CPU 使用率、物理内存使用、网络流量使用等。

- 前端需主要关注的点是：

- 响应时间：用户从客户端发出请求，并得到响应，以及展示出来的整个过程的时间。

- 加载速度：通俗的理解为页面内容显示的快慢。

流量：所消耗的网络流量。

- 后端需主要关注的是：

- 响应时间：接口从请求到响应、返回的时间。

- 并发用户数：同一时间点请求服务器的用户数，支持的最大并发数。

- 内存占用：也就是内存开销。

- 吞吐量（TPS）：Transaction Per Second，每秒事务数。在没有遇到性能瓶颈时： $TPS = \text{并发用户数} * \text{事务数} / \text{响应时间}$ 。

- 错误率：失败的事务数/事务总数。

- 资源使用率：CPU 占用率、内存使用率、磁盘 I/O、网络 I/O。

- 从性能测试分析度量的角度来看，主要可以从如下几个大的维度来收集考察性能指标：

- 系统性能指标、资源性能指标、稳定性指标

\2.

如果一个需求没有明确的性能指标，要如何开始进行性能测试？

- 先输出业务数据，如 pv、pu、时间段等，计算出大概的值，然后不断加压测到峰值

\3.

介绍 JMeter 聚合报告包括哪些内容？

- 请求名、线程数、响应时间（50 95 99 最小 最大）错误率、吞吐量

\4.

如果有一个页面特别卡顿，设想一下可能的原因？

- 
- 后台：接口返回数据慢，查询性能等各种问题
- 
- 前端：使用 Chrome 工具调试，判断 JS 执行久或是其他问题
- 
- 网络问题

\5.

说一说项目中的实际测试内容

- 
- 根据自己项目中的经验实话实说，有没有经验很容易露馅。

\6.

介绍一下 JMeter 进行性能测试的过程

- 
- 结合自己的项目经验聊。大家也可以自行搜索。

\7.

介绍一下 JMeter 和 LoadRunner 的区别

- 
- 详细的不展开了，最重要的是相对来说 LoadRunner 的笨重、昂贵、闭源，理念和生态都落后，而 JMeter 是开源、可定制化开发，功能强大易用，并且在互联网大厂都已经有了非常成熟的落地方案（主流的互联网公司基本都在使用 JMeter+ELK+Grafana+Influxdb 这套架构），可以说是进 BAT 大厂必备技能。还不会 JMeter 的同学建议抓紧补起来。

### 测试用例设计经典面试题——电梯，杯子，笔，桌子，洗衣机

首先说明的是，遇到这样的测试题目，首先应该反问面试官，需求是什么样的，比如是测什么样的杯子。

因为设计测试用例的规则应该是根据需求分析文档设计用例，客户需求什么，就测试什么。但是在没有需求分析文档的前提下，来设计测试用例，可以考查一个测试人员的基本功，比如考虑问题是否全面，设计测试用例的方法是否合理等。一般是根据自己的日常经验和测试的思维来设计测试用例。在设计测试用例时一般从以下几个方面进行分析：功能测试，性能测试，界面测试，安全性测试，兼容性测试，可用性测试，可靠性测试，本地化/国际化测试。

#### \*1、测试项目——电梯\*

**需求测试：**查看电梯使用说明书，安全说明书等。

**功能测试：**

- 1、上升键和下降键，电梯的楼层按钮是否正常；
- 2、开关键是否正常，报警装置是否安装，报警电话是否可用；
- 3、通风状况如何，是否有手机信号；
- 4、在电梯上升过程中的测试，比如电梯在 1 楼，有人按了 18 楼，在上升到 5 楼的时候，有人按了 10 楼，电梯会不会停；
- 5、在电梯下降过程中的测试，比如电梯下降到 10 层时显示满员，若有人在 5 楼等待，此时还会不会停。

#### 压力测试：

看电梯的最大承重重量，在电梯超重时，报警装置是否启用，在一定时间内让电梯连续的上升和下降，看在最大负载条件下平稳运行的时间。

#### 界面测试：

查看电梯的外观，电梯的按钮是否好用(开和关按钮设计的图标不容易区分)，电梯的说明书是否有错别字。

#### 可用性测试：

电梯的按钮是否符合人的使用习惯。

#### 用户文档：

使用手册是否对电梯的使用，限制等有描述。

### 2、测试项目——杯子

**需求测试：**查看杯子的使用说明书，安全说明书等。

#### 功能测试：

- 1、杯子能否装水；
- 2、可以装多少 L 的水；
- 3、杯子是否可以放冰箱；
- 4、水可不可以被喝到。

#### 安全性测试：

- 1、杯子有没有毒和细菌；
- 2、杯子从高处坠落，是否已破；
- 3、杯子是否有缺口，容易滑倒嘴巴；
- 4、将杯子放入微波炉中，是否爆炸或融化；

#### 性能测试：

- 1、看杯子能够容纳的最大体积和最高温度；
- 2、将杯子盛上水，经过 24 小时后查看杯子的泄露情况和时间(可分别使用水和汽油做测试)；
- 3、将杯子装上填充物，看不会摔破的最高度；
- 4、用根针并在针上面不断加重量，看压强多大时会穿透；

**可用性测试：**杯子是否好拿，是否烫手，是否防滑，是否方便饮用。

**兼容性测试：**除了装水，是否还可以装其它的液体，比如果汁，汽油等。

**界面测试：**查看杯子的外观：杯子是什么材质的，颜色，外形，重量，图案是否合理，是否有异味。

**用户文档：**使用手册是否对杯子的用法、限制、使用条件等有详细描述。

### \*3、测试项目——笔\*

- 1、需求测试：查看使用说明书。
- 2、功能测试：能不能写字。
- 3、界面测试：查看笔的外观。
- 4、可靠性：笔从不同高度落下摔坏的程度。
- 5、安全性：笔有没有毒或细菌。
- 6、可移植性：笔在不同的纸质、温度等环境下是否可以使用。
- 7、兼容性：笔是否可以装不同颜色、大小的笔芯。
- 8、易用性：是否方便使用、方便携带。
- 9、压力测试：给笔不断的增加重力，观察压力多大时压坏。



10、震动测试：笔在包装时，各面震动，检查是否能应对恶劣的公路、铁路、航空运输。

11、跌落测试：笔包装时，在多高的情况下摔不坏。

#### \*4、测试项目——桌子\*

需求测试：查看桌子相关的使用说明书。

功能测试：桌子是办公用的还是防治东西用的，桌子的面积大小是否适合；

界面测试：桌子的桌面是否平滑，有没有凹凸不平的地方；

安全性测试：桌子的支撑点是否可靠；将桌子推倒后，它的损坏情况；

压力测试：桌子可以承受的重量；

可用性测试：桌子是否好移动；它的重量是否合适；

#### \*5、测试项目——洗衣机\*

需求测试：查看洗衣机的使用说明书。

功能测试：洗衣机是否正常的洗衣服；

安全性测试：洗衣机是否漏电；

兼容性测试：除了洗衣服是否还可以洗其它的；

性能测试：使用时电量如何，是否满足客户需求；加到一定量的衣服后，过一段时间，看洗衣机是否正常洗；通过逐步增加系统负载，最终确定在什么负

载条件下系统性能将处于崩溃状态，以此获得系统能提供的最大服务

界面测试：洗衣机的外观是否符合用户的需求；

可用性测试：洗衣机的操作是否简单已操作；

## 面试题-six

### 01 性能测试包含了哪些软件测试（至少举出 3 种）？

参考答案：负载测试；压力测试；容量测试

**负载测试（Load Testing）**：负载测试是一种主要为了测试软件系统是否达到需求文档设计的目标，譬如软件在一定时期内，最大支持多少并发用户数，软件请求出错率等，测试的主要是软件系统的性能。

**压力测试（Stress Testing）**：强度测试也就是压力测试，压力测试主要是为了测试硬件系统是否达到需求文档设计的性能目标，譬如在一定时期内，系统的 cpu 利用率，内存使用率，磁盘 I/O 吞吐率，网络吞吐量等，压力测试和负载测试最大的差别在于测试目的不同。

**容量测试（Volume Testing）**：确定系统最大承受量，譬如系统最大用户数，最大存储量，最多处理的数据流量等。

或者在下面选择几项：

并发测试



—  
测试多用户并发访问同一个应用、模块、数据时是否产生隐藏的并发问题  
基准测试  
—

比较新的或未知测试对象与已知参照标准（如现有软件或评测标准）的性能。

争用测试：- 核实测试对象对于多个主角对相同资源（数据记录、内存等）的请求的处理是否可以接受。

性能配置 - 核实在操作条件保持不变的情况下，测试对象在使用不同配置时其性能行为的可接受性。

负载测试- 核实在保持配置不变的情况下，测试对象在不同操作条件（如不同用户数、事务数等）下性能行为的可接受性。

强度测试- 核实测试对象性能行为在异常或极端条件（如资源减少或用户数过多）之下的可接受性。

容量测试- 核实测试用户同时使用软件程序的最大数量

## 02 请问什么是性能测试、负载测试、压力测试？

答：性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试、压力测试参考答案如上题。

## 03 在给定的测试环境下进行，考虑被测系统的业务压力量和典型场景？

答：负载测试。

负载测试是用来测定系统饱和状态、确定阈值。其特点有：

（1） 这种方法的目的是找到系统处理能力的极限；通过“检测、加压、阈值”手段找到如“响应时间不超过 10 秒”，“平均 CPU 利用率低于 65%”等指标。

（2） 这种性能测试方法需要在给定的测试环境下进行，通常也需要考虑被测系统的业务压力量和典型场景、另外 HP Mercury LoadRuner 在使用该方法进行“加压”的时候必须选择典型场景。

（3） 这种性能测试方法一般用来了解系统的性能容量，或者是配合性能调优的时候来使用。特别是该的 Weblogic 和库的性能调优。

## 04 什么时候可以开始执行性能测试？

答：在产品相对比较稳定，功能测试结束后。灵活性比较强。

## 05 简述性能测试的步骤。

答：熟悉应用

了解应用的架构、功能逻辑

测试需求：

1、需要将开发给定的需求转为吞吐量和响应时间。

2、根据测试目的，细化需求

测试准备：

测试准备包括测试客户端机器准备、测试数据准备、测试脚本准备。

测试执行：



测试的执行中，需要监控测试客户端和服务端性能，监控服务器端应用情况：

- 客户端的系统资源（cpu、io、memory）情况
- 服务端的系统资源（cpu、io、memory）情况
- 服务器的 jvm 运行情况
- 服务端的应用情况，看是否有异常
- 响应时间、吞吐量等指标
- 系统资源监控，linux 下可以采用的工具有：vmstat、top、meminfo 等。
- JVM 的监控，可以用 jprofiler 工具，linux 下面的 jmap、jhat 等。
- 响应时间、吞吐量等，由 grinder 提供。

上述这些信息，一般在测试结束后，均需要归档整理，以备后续详细分析。

我们自己开发一套脚本，用于以固定的频率获取测试客户端和服务器的 vmstat 和 top 输出、grinder 的 log，并从中截取有用信息保存，用于事后分析。每次测试运行完以后，肯定会增加很多数据，需要考虑本次执行对数据量的影响，如果数据量的变化对后续测试会有影响，则需要清理数据。

#### 06 你如何识别性能瓶颈？

答：RBI 方法：重点测试“吞吐量”指标，因为 RBI 认定 80%的系统性能瓶颈由吞吐量造成。按照网络、硬件、数据库、应用服务器、代码的顺序自上而下分析性能工具：

IBM、  
HP、

OpenSource 工具都支持。需使用分析模块、根据 Weblogic、Oracle 区别有专门的工具实现 RBI。

#### 07 性能测试时，是不是必须进行参数化？为什么要创建参数？LoadRunner 中如何创建参数？

答：是。模拟用户真实的业务操作。创建参数列表，用参数替换固定的文本。

#### 08 你如何设计负载？标准是什么？

答：负载测试计划多少用户数量、使用什么类型的机器、以及在什么环境下进行。主要基于两个重要的文档，任务分布图和事务信息，任务分布图告诉我们在负载时间段内，某一个事务使用的用户数，高峰使用率及低峰使用率均来自该文档；事务信息告诉我们事务名及优先级，在设计场景时可以参考。

#### 09 解释 5 个常用的性能指标的名称与具体含义。

答：响应时间、并发用户数，吞吐量，性能计数器，TPS，HPS

**响应时间：**指的是“系统响应时间”定义为应用系统从发出请求开始到客户端接收到响应所消耗的时间。把它作为用户视角的软件性能的主要体现。

**最大并发用户数：**有两种理解方式，一种是从业务的角度来模拟真实的用户访问，体现的是业务并发用户数，指在同一时间段内访问系统的用户数量。另一种是从服务器端承受的压力来考虑，这里的“并发用户数”指的是同时向服务器端发出请求的客户数，该概念一般结合并发测试（Concurrency Testing）使用，体现的是服务端承受的最大并发访问数。

**吞吐量：**是指“单位时间内系统处理的客户请求的数量”，直接体现软件系统的性能承载能力。

**性能计数器（Counter）**是描述服务器或操作系统性能的一些数据指标。例如，对 Windows 系统来说，使用内存数（Memory In Usage），进程时间（Total Process Time）等都是常见的计数器。

**思考时间（Think Time）**，也被称为“休眠时间”，从业务的角度来说，这个时间指的是用户在进行操作时，每个请求之间的间隔时间。从自动化测试实现的角度来说，要真实地模拟用户操作，就必须在测试脚本中让各个操作之间等待一段时间，体现在脚本中，具体而言，就是在操作之间放置一个 Think 的函数，使得脚本在执行两个操作之间等待一段时间。

**TPS：Transaction per second**，每秒钟系统能够处理的交易或者事务的数量。它是衡量系统处理能力的重要指标。

**点击率：HPS**，每秒钟用户向 WEB 服务器提交的 HTTP 请求数。

**10 描述不同的角色（用户、产品开发人员、系统管理员）各自关注的软件性能要点。**

**答：用户：**重点关注打开速度及响应时间

**开发：**重点关注响应时间和数据库交互

**管理员：**重点关注用户感受到的软件性能；如何利用管理功能进行性能调优；如何利用其他软硬件手段进行性能调优

**11 请分别针对性能测试、负载测试和压力测试试举一个简单的例子？**

**答：性能测试例子：**公司开发了一个小型项目管理系统，上线前需要做负载、压力、大数据量、强度测试等。

**负载测试：**逐步加压，从而得到“响应时间不超过 10 秒”，“服务器平均 CPU 利用率低于 85%”等指标阈值。

**压力测试：**逐步加压，从而使“响应时间超过 10 秒”，“服务器平均 CPU 利用率高于 90%”等指标来确定系统能承受的最大负载量。

**12 请问您是如何得到性能测试需求？怎样针对需求设计、分析是否达到需求？**

**答：**在查看需求文档，从中提取性能测试需求，与用户交流，了解实际使用情况。结合业务信息设计操作场景总结出需测试的性能关键指标。执行用例后根据提取关键性能指标来分析是否满足性能需求。

## 面试题- seven

HTTPS 和 HTTP 的区别主要如下：

- 1、https 协议需要到 ca 申请证书，一般免费证书较少，因而需要一定费用。
- 2、http 是超文本传输协议，信息是明文传输，https 则是具有安全性的 ssl 加密传输协议。
- 3、http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，后者是 443。
- 4、http 的连接很简单，是无状态的；HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，比 http 协议安全。

### HTTPS 的工作原理

我们都知道 HTTPS 能够加密信息，以免敏感信息被第三方获取，所以很多银行网站或电子邮箱等等安全级别较高的服务都会采用 HTTPS 协议。

客户端在使用 HTTPS 方式与 Web 服务器通信时有以下几个步骤，如图所示。

- (1) 客户使用 https 的 URL 访问 Web 服务器，要求与 Web 服务器建立 SSL 连接。
- (2) Web 服务器收到客户端请求后，会将网站的证书信息（证书中包含公钥）传送一份给客户端。
- (3) 客户端的浏览器与 Web 服务器开始协商 SSL 连接的安全等级，也就是信息加密的等级。
- (4) 客户端的浏览器根据双方同意的安全等级，建立会话密钥，然后利用网站的公钥将会话密钥加密，并传送给网站。
- (5) Web 服务器利用自己的私钥解密出会话密钥。
- (6) Web 服务器利用会话密钥加密与客户端之间的通信。

### 四、HTTPS 的优点

尽管 HTTPS 并非绝对安全，掌握根证书的机构、掌握加密算法的组织同样可以进行中间人形式的攻击，但 HTTPS 仍是现行架构下最安全的解决方案，主要有以下几个好处：

- (1) 使用 HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户机和服务器；
- (2) HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 http 协议安全，可防止数据在传输过程中不被窃取、改变，确保数据的完整性。
- (3) HTTPS 是现行架构下最安全的解决方案，虽然不是绝对安全，但它大幅增加了中间人攻击的成本。
- (4) 谷歌曾在 2014 年 8 月份调整搜索引擎算法，并称“比起同等 HTTP 网站，采用 HTTPS 加密的网站在搜索结果中的排名将会更高”。

### 五、HTTPS 的缺点

虽然说 HTTPS 有很大的优势，但其相对来说，还是存在不足之处的：

- (1) HTTPS 协议握手阶段比较费时，会使页面的加载时间延长近 50%，增加 10% 到 20% 的耗电；
- (2) HTTPS 连接缓存不如 HTTP 高效，会增加数据开销和功耗，甚至已有的安

全措施也会因此而受到影响；

(3) SSL 证书需要钱，功能越强大的证书费用越高，个人网站、小网站没有必要一般不会用。

(4) SSL 证书通常需要绑定 IP，不能在同一 IP 上绑定多个域名，IPv4 资源不可能支撑这个消耗。

(5) HTTPS 协议的加密范围也比较有限，在黑客攻击、拒绝服务攻击、服务器劫持等方面几乎起不到什么作用。最关键的，SSL 证书的信用链体系并不安全，特别是在某些国家可以控制 CA 根证书的情况下，中间人攻击一样可行。

## 六、http 切换到 HTTPS

如果需要将网站从 http 切换到 https 到底该如何实现呢？

这里需要将页面中所有的链接，例如 js, css, 图片等等链接都由 http 改为 https。例如：<http://www.baidu.com> 改为 <https://www.baidu.com>

BTW, 这里虽然将 http 切换为了 https, 还是建议保留 http。所以我们在切换的时候可以做 http 和 https 的兼容，具体实现方式是，去掉页面链接中的 http

头部，这样可以自动匹配 http 头和 https 头。例如：将 <http://www.baidu.com> 改为 <http://www.baidu.com>。然后当用户从 http 的入口进入访问页面时，页面就是 http，如果用户是从 https 的入口进入访问页面，页面即使 https 的。

### 提到 UI 级别测试和 API 测试之间的关键区别？

UI（用户界面）是指测试图形界面，如用户如何与应用程序交互，测试应用程序元素，如字体，图像，布局等。UI 测试基本上侧重于应用程序的外观和感觉。而 API 可以实现两个独立的软件系统之间的通信。实现 API 的软件系统包含可由另一软件系统执行的功能或子例程

请详细阐述接口测试和 UI 测试在测试活动中是如何协同测试的？

### 接口测试用例的编写要点有哪些？

- 1) 必填字段：请求参数必填项、可选项
- 2) 合法性：输入输出合法、非法参数
- 3) 边界：请求参数边界值等
- 4) 容错能力：大容量数据、频繁请求、重复请求（如：订单）、异常网络等的处理
- 5) 响应数据校验：断言、数据提取传递到下一级接口...
- 6) 逻辑校验：如两个请求的接口有严格的先后顺序，需要测试调转顺序的情况
- 7) 性能：对接口模拟并发测试，逐步加压，分析瓶颈点
- 8) 安全性：构造恶意的字符请求，如：SQL 注入、XSS、敏感信息、业务逻辑（如：跳过某些关键步骤；未经验证操纵敏感数据）

\* 测试每个参数类型不合法的情况(类型不合法容易遗漏 NULL 型)

\*

测试每个参数取值范围不合法的情况

\*



测试参数为空的情况

\*

测试参数前后台定义的一致性

\*

测试每个参数的上下限(这里容易出致命的 BUG，如果程序处理不当，可能导致崩溃)

\* 如果两个请求有严格的先后顺序，需要测试调转顺序的情况

**接口自动化测试的流程？**

基本的接口功能自动化测试流程为：需求分析-->用例设计-->脚本开发-->测试执行-->结果分析

POST 和 GET 有什么区别？

答案

1.

a.

GET 在浏览器回退时是无害的，而 POST 会再次提交请求。

a.

GET 产生的 URL 地址可以被保存为书签，而 POST 不可以。

a.

GET 请求会被浏览器主动 cache，而 POST 不会，除非手动设置。

a.

GET 请求只能进行 url 编码，而 POST 支持多种编码方式。

a.

GET 请求参数会被完整保留在浏览器历史记录里，而 POST 中的参数不会被保留。

a.

GET 请求在 URL 中传送的参数是有长度限制的，而 POST 没有。

a.

GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。

a.

GET 参数通过 URL 传递，POST 放在请求 Body 中。

a.

GET 产生一个 TCP 数据包，POST 产生两个 TCP 数据包。

a.

GET 请求的 URL 传参有长度限制，而 POST 请求没有长度限制

a.

GET 请求的参数只能是 ASCII 码，所以中文需要 URL 编码，而 POST 请求传参没有这个限制；

a.

GET 产生一个 TCP 数据包；POST 产生两个 TCP 数据包

常见的 HTTP

Header 及其作用

答案

常规 HTTP 头  
 Request  
 URL: <https://www.jintix.com/>  
 Request  
 Method: GET  
 Status  
 Code: 200 OK  
 Remote  
 Address: 14.215.177.38:443  
 相应头:  
 Cache-Control: private  
 Connection: Keep-Alive  
 Content-Encoding: gzip  
 Content-Type:  
 text/html; charset=utf-8  
 Date: Wed, 27  
 Mar 2019 09:18:39 GMT  
 Expires: Wed,  
 27 Mar 2019 09:18:39 GMT  
 Server: BWS/1.1  
 Set-Cookie:  
 H\_PS\_PSSID=26524\_1449\_21099\_28721\_28558\_28697\_28585\_28640\_26350\_2870  
 2; path=/  
 Strict-Transport-Security: max-age=172800  
 Transfer-Encoding: chunked  
 X-Ua-Compatible: IE=Edge, chrome=1  
 请求头:  
 Accept:  
 text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, ima  
 ge/apng, /;q=0.8  
 Accept-Encoding:  
 gzip, deflate, br  
 Accept-Language:  
 zh-CN, zh;q=0.9, en;q=0.8  
 Cache-Control: max-age=0  
 Connection: keep-alive  
 Cookie: BIDUPSID=8B429E47FD1DA3ED78C8389452B0E8C2; PSTM=1529386822  
 Host: <http://www.baidu.com>  
 User-Agent:  
 Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML,  
 like  
 Gecko)  
 Chrome/68.0.3440.84 Safari/537.36

HTTP  
与  
HTTPS 的区别  
答案

1.

a.

HTTP

明文传输，数据都是未加密的，安全性较差，HTTPS  
(SSL+HTTP)

数据传输过程是加密的，安全性较好。

a.

使

用

HTTPS

协议

需要到

CA

(Certificate

Authority, 数字证书认证机构)

申请证书，一般免

费证书较少，因而需要一定费用。证书颁发机构如：Symantec、Comodo、  
GoDaddy

和

GlobalSign 等。

a.

HTTP

页面响应速度比

HTTPS 快，主要是因

为

HTTP 使用 TCP 三次握手建立连接，客户端和服务端需要交  
换

3 个包，而 HTTPS 除了

TCP 的三个包，还要加上 ssl 握  
手需要的

9

个包，所以一共是

12 个包。

a.

http

和 https 使用的是完全不同的连接方式，用的端  
口也不一样，前者是

80，后者是 443。

a.

HTTPS



其实就是建构在

SSL/TLS 之上的

HTTP

协议，所以，要比较

HTTPS 比

HTTP

要更耗费服务器资源。

Session 与 Cookie 有什么区别？

答案

1.

a.

**保存位置。**SESSION

数据保存在**服务器端**，Cookie 数据保存在客

户端浏览器

a.

**保存方式。**SESSION 默认被存在在服务器的一个文件里，可以手动设置放在文件、数据库、或内存中；Cookie

默认保存在客户端**内存中**，如果

设置了过期时间就保存在硬盘中。

a.

**依赖关系。**SESSION 依赖

Cookie 来识别 session\_id，

如果浏览器禁用了 Cookie，SESSION

也会失效，此时可以通

过

url 传递 session\_id。

a.

**安全性。**因为 SESSION 数据保存在服务端，所以

SESSION 安全

性比

Cookie 高。

a.

**尺寸大小。**SESSION 基本上没有大小限制，COOKIE

保存的内容比

较小，具体由浏览器决定。

a.

**服务器性能。**

SESSION 对**服务器的压力会更大一些**，而 Cookie 放在客户端，所以对服务器基本没影响

HTTP 通讯过程中，是客户端还是服务端主动断开连接？

答案

不考虑 keepalive 的情况下，

http1.0

- 
- 

带 content-length, body 长度可知, 客户端在接收 body 时, 就可以依据这个长度来接受数据。接受完毕后, 就表示这个请求完毕了。**客户端主动调用 close 进入四次挥手。**

- 

不带 content-length, body 长度不可知, 客户端一直接受数据, 直到**服务端主动断开**  
http1.1

- 
- 

带 content-length body 长度可知 **客户端主动断开**

- 

带 Transfer-encoding: chunked body 会被分成多个块, 每块的开始会标识出当前块的长度, body 就不需要通过 content-length 来指定了。但依然可以知道 body 的长度 **客户端主动断开**

- 

不带 Transfer-encoding: chunked 且不带 content-length 客户端接收数据, 直到**服务端主动断开连接**。  
即：如果能够有办法知道服务器传来的长度, 都是客户端首先断开。如果不知道就一直接收数据。知道服务端断开。

**HTTP 有哪些请求方法?**

HTTP

共有如下 7 种请求方式, 每种都可以发送 Header 和 Body:

1.
  - a. GET
  - a. POST
  - a. PUT
  - a. DELETE
  - a. OPTIONS
  - a. HEAD
  - a. PATCH

**Cookie**

**保存在哪里?**

答案

-

○  
如果设置了过期时间，Cookie 保存在硬盘中。  
○

如果没有设置过期时间，Cookie 保存在内存中。

简述 TCP/IP 的三次握手和四次挥手

答案

三次握手

起初，服务器和客户端都为 CLOSED 状态。

在通信开始前，双方都得创建各自的传输控制块（TCB）。

服务器创建完 TCB 后进入 LISTEN 状态，此时准备接收客户端发来的连接请求。

**第一次握手**

客户端向服务端发送连接请求报文段。该报文段的头部中 SYN=1, ACK=0, seq=x。请求发送后，客户端便进入 SYN-SENT 状态。

•  
○  
PS1: SYN=1, ACK=0 表示该报文段为连接请求报文。

○  
PS2: x 为本次 TCP 通信的字节流的初始序号。

○  
TCP 规定：SYN=1 的报文段不能有数据部分，但要消耗掉一个序号。

**第二次握手**

服务端收到连接请求报文段后，如果同意连接，则会发送一个应答：

SYN=1, ACK=1, seq=y, ack=x+1。该应答发送完成后便进入 SYN-RCVD 状态。

•  
○  
PS1: SYN=1, ACK=1 表示该报文段为连接同意的应答报文。

○  
PS2: seq=y 表示服务端作为发送者时，发送字节流的初始序号。

○  
PS3: ack=x+1 表示服务端希望下一个数据报发送序号从 x+1 开始

的字节。

**第三次握手**

当客户端收到连接同意的应答后，还要向服务端发送一个确认报文段，表示：服务端发来的连接同意应答已经成功收到。该报文段的头部为：

ACK=1,

seq=x+1,

ack=y+1。

客户端发完这个报文段后便进入 ESTABLISHED 状态，服务端收到这个应答后也进入 ESTABLISHED 状态，此时连接的建立完成！

四次挥手

TCP 连接的释放一共需要四步，因此称为『四次挥手』。

我们知道，TCP 连接是双向的，因此在四次挥手中，前两次挥手用于断开一个方向的连接，后两次挥手用于断开另一方向的连接。

### 第一次挥手

若 A 认为数据发送完成，则它需要向 B 发送连接释放请求。该请求只有报文头，头中携带的主要参数为：FIN=1，seq=u。此时，A 将进入 FIN-WAIT-1 状态。

•

○

PS1: FIN=1 表示该报文段是一个连接释放请求。

○

PS2: seq=u，u-1 是 A 向 B 发送的最后一个字节的序号。

### 第二次挥手

B 收到连接释放请求后，会通知相应的应用程序，告诉它 A 向 B 这个方向的连接已经释放。此时 B 进入 CLOSE-WAIT 状态，并向 A 发送连接释放的应答，其报

文头包含：ACK=1，seq=v，ack=u+1。

○

PS1: ACK=1: 除 TCP 连接请求报文段以外，TCP 通信过程中所有数据报的 ACK 都为 1，表示应答。

○

PS2: seq=v，v-1 是 B 向 A 发送的最后一个字节的序号。

○

PS3: ack=u+1 表示希望收到从第 u+1 个字节开始的报文段，并且已经成功接收了前 u 个字节。

A 收到该应答，进入 FIN-WAIT-2 状态，等待 B 发送连接释放请求。

第二次挥手完成后，A 到 B 方向的连接已经释放，B 不会再接收数据，A 也不会

再发送数据。但 B 到 A 方向的连接仍然存在，B 可以继续向 A 发送数据。

### 第三次挥手

当 B 向 A 发完所有数据后，向 A 发送连接释放请求，请求头：FIN=1，ACK=1，seq=w，ack=u+1。B 便进入 LAST-ACK 状态。

### 第四次挥手

A 收到释放请求后，向 B 发送确认应答，此时 A 进入 TIME-WAIT 状态。

该状态会持续 2MSL 时间，若该时间段内没有 B 的重发请求的话，就进入 CLOSED

状态，撤销 TCB。

当 B 收到确认应答后，也便进入 CLOSED 状态，撤销 TCB。

为什么 A 要先进入 TIME-WAIT 状态，等待 2MSL 时间后才进入 CLOSED 状态？

为了保证 B 能收到 A 的确认应答。若 A 发完确认应答后直接进入 CLOSED 状态，那么如果该应答丢失，B 等待超时后就会重新发送连接释放请求，但此时 A 已经关闭了，不会作出任何响应，因此 B 永远无法正常关闭。

TCP 和 UDP 有什么区别

答案

1.

a.

TCP 面向连接（如打电话要先拨号建立连接）；UDP 是无连接的，即发送数据之前不需要建立连接

a.

TCP 提供可靠的服务。也就是说，通过 TCP 连接传送的数据，无差错，不丢失，不重复，且按序到达；UDP 尽最大努力交付，即不保证可靠交付

a.

TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流；UDP 是面向报文的 UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）

a.

每一条 TCP 连接只能是点到点的；UDP 支持一对一，一对多，多对一和多对多的交互通信

a.

TCP 首部开销 20 字节；UDP 的首部开销小，只有 8 个字节

a.

TCP 的逻辑通信信道是全双工的可靠信道，UDP 则是不可靠信道

什么是 TCP/IP？

答案

**TCP/IP**，也就是互联网协议套件（英语：Internet Protocol Suite，缩写 IPS）是一个**网络通信模型**，以及一整个**网络传输协议家族**，为网际网络的基础通信架构。

因为该协议家族的两个核心协议：TCP（传输控制协议）和 IP（网际协议）为该家族中最早通过的标准，所以它常被通称为 **TCP/IP 协议族**，简称 TCP/IP。由于在网络通讯协议普遍采用分层的结构，当多个层次的协议共同工作时，类似计算机科学中的堆栈，因此又被称为 **TCP/IP 协议栈**。

TCP/IP 提供点对点的链接机制，将数据应该如何封装、定址、传输、路由以及在目的地如何接收，都加以标准化。

它将软件通信过程抽象化为**四个抽象层**，采取协议堆栈的方式，分别实现出不同通信协议。

协议族下的各种协议，依其功能不同，被分别归属到这四个层次结构之中，常被视为是简化的**七层 OSI 模型**。

1. json 和字典的区别？-对基础数据类型的考察

2. 测试的数据你放在哪？-数据与脚本分离

3. 参数化 - 数据驱动模式

4. 下个接口请求参数依赖上个接口的返回数据 - 参数关联

5. 依赖于登录的接口如何处理

-token 和 session 的管理

6. 依赖第三方的接口如何处理

-mock 模拟数据返回

7. 不可逆的操作，如何处理，比如删除一个订单这种接口如何测试 -造数据

8. 接口产生的垃圾数据如何清理 - 数据清理

9. 一个订单的几种状态如何全部测到，如：未处理，处理中，处理失败，处理成功 - 造数据，改数据库订单状态

10. python 如何连接数据库操作？

11. 其它的就是运行出报告、代码管理（git）、运行策略和持续集成 jenkins 相关了

### 1. json 和字典 dict 的区别？

现在自动化培训烂大街，是个人都能说的上几个框架，面试如果问框架相关问题，求职者只需一瓶 82 年的雪碧，会吹的让你怀疑人生！

所以面试官为了更清楚的知道你是停留在表面上的花拳绣腿还是有扎实的基础，就不会问框架这种东西了。基本上问几个数据类型的基础就知道有没货了。那么 json 和字典到底有什么区别呢？初学者连 python 的基础数据类型都没搞清楚，直接撸框架，有的人学了几个月可能都迷迷糊糊的，以为 json 就是字典。这个是肯定不对的。

首先 python 里面的基础数据类型有：int、str、float、list、bool、tuple、dict、set 这几种类型，里面没 json 这种数据类型。

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的 js 规范) 的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

由于你的代码是 python 写的（也有可能是 php, java, c, ruby 等语言），但是后

端接口是 java 写的（也有可能是其它语言），不同的语言数据类型是不一样的（就好比中国的语言和美国语言数据类型也不一样，中国的一般说一只羊，一头牛，美国都是 a /an 这种单位），所以就导致你提交的数据，别的开发语言无法识别，这就需要规范传输的数据（传输的数据都是一个字符串），大家都遵循一个规范，按一个标准的格式去传输，于是就有 json 这种国际化规范的数据类型。

json 本质上还是字符串，只是按 key: value 这种键值对的格式来的字符串

```
import json
#
a 是字典 dict
a
=
{"a": 1, "b": 2, "c": True}
#
b 是 json
b
=
'{"a": 1, "b": 2,
"c": true}'
print(type(a))
print(json.dumps(a))
```

```
# a 转 json
运行结果
<class 'dict'>
{'a': 1, 'b':
2,
'c':
true}
<class 'str'>
{'a': 1, 'b':
2,
'c':
True}
```

## 2. 测试的数据你放在哪？

测试数据到底该怎么放，这个是面试官最喜欢问的一个题了，似乎仁者见仁智者见智，没有标准的答案，有的人说放 excel，也有的说放.py 脚本，也有的说放 ini 配置文件，还有放到 json, yaml 文件, txt 文件，甚至有的放数据库，五花八门，一百个做自动化的小伙伴有 100 个放的地方。

这里总结下测试的数据到底该怎么放？

首先测试的数据是分很多种的，有登录的账户数据，也有注册的账户数据，还有接口的参数，还有邮箱配置的数据等等等等，所以这个题不能一概而论给答死了。要不然就是给自己挖坑。

以下两个大忌不能回答：

- -
- 测试的数据是不能写死到代码里面的，这个是原则问题，也是写代码的大忌（你要是回答写在代码里面，估计就是回去等通知了）

◦

测试数据放到.py 的开头，这种其实很方便，对于少量的，固定不变的数据其实是可以放的，但是面试时候，千万不能这样说，面试官喜欢装逼的方法

测试数据存放总结：

1. 对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，写的配置文件里（如 ini）
2. 对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成
3. 对于一个接口有多组测试的参数，可以参数化，数据放 yaml, text, json, excel 都可以
4. 对于可以反复使用的数据，比如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理
5. 对于邮箱配置的一些参数，可以用 ini 配置文件
6. 对于全部是独立的接口项目，可以用数据驱动方式，用 excel/csv 管理测试的接口数据



7. 对于少量的静态数据，比如一个接口的测试数据，也就 2-3 组，可以写到 py 脚本的开头，十年八年都不会变更的  
总之不同的测试数据，可以用不同的文件管理

### 3. 什么是数据驱动，如何参数化？

参数化和数据驱动的概念这个肯定要知道的，参数化的思想是代码用例写好了后，不需要改代码，只需维护测试数据就可以了，并且根据不同的测试数据生成多个用例

python 里面用 unittest 框架

```
import
unittest
import
ddt
# 测试数据
datas = [ {"user":
"admin", "psw": "123", "result": "true"},
{"user":
"admin1",
"psw":
"1234",
"result":
"true"},
{"user":
"admin2",
"psw":
"1234",
"result":
"true"},
{"user":
"admin3",
"psw":
"1234",
"result":
"true"},
{"user":
"admin4",
"psw":
"1234",
"result":
"true"},
{"user":
"admin5",
"psw":
"1234",
```



```
"result":
"true"},
{"user":
"admin6",
"psw":
"1234",
"result":
"true"},
{"user":
"admin7",
"psw":
"1234",
"result":
"true"},
{"user":
"admin8",
"psw":
"1234",
"result":
"true"},
{"user":
"admin9",
"psw":
"1234",
"result":
"true"},
{"user":
"admin10",
"psw":
"1234",
"result":
"true"},
{"user":
"admin11",
"psw":
"1234",
"result":
"true"}]
```

[@ddt.ddt](#)

```
class Test(unittest.TestCase):
@ddt.data(*datas)
def test_(self, d):
"""上海一凡: {0}"""
print("测试数据: %s" % d)
```

```

if name == "main":
    unittest.main()
unittest 框架还有一个 paramunittest 也可以实现
import
unittest
import
paramunittest
import
time
#
python3.6
#

```

作者：上海一凡

```

@paramunittest.parametrize(
    {"user":
    "admin", "psw": "123", "result": "true"},
    {"user":
    "admin1",
    "psw":
    "1234",
    "result":
    "true"},
    {"user":
    "admin2",
    "psw":
    "1234",
    "result":
    "true"},
    {"user":
    "admin3",
    "psw":
    "1234",
    "result":
    "true"},
    {"user":
    "admin4",
    "psw":
    "1234",
    "result":
    "true"},
    {"user":
    "admin5",
    "psw":

```

```

"1234",
"result":
"true"},
{"user":
"admin6",
"psw":
"1234",
"result":
"true"},
{"user":
"admin7",
"psw":
"1234",
"result":
"true"},
{"user":
"admin8",
"psw":
"1234",
"result":
"true"},
{"user":
"admin9",
"psw":
"1234",
"result":
"true"},
{"user":
"admin10",
"psw":
"1234",
"result":
"true"},
{"user":
"admin11",
"psw":
"1234",
"result":
"true"},
)
class TestDemo(unittest.TestCase):
def setParameters(self, user, psw,
result):
'''这里注意了，user, psw, result 三个参数和前面定义的字典一一对应'''

```

```

self.user
=
user
self.user
=
psw
self.result
= result
def testcase(self):
print("开始执行用例：-----")
time.sleep(0.5)
print("输入用户名： %s"
% self.user)
print("输入密码： %s" %
self.user)
print("期望结果： %s " % self.result)
time.sleep(0.5)
self.assertTrue(self.result == "true")
if name == "main":
unittest.main(verbosity=2)
如果用的是 pytest 框架，也能实现参数化
#
content of test_canshu1.py
#
coding:utf-8
import pytest

@pytest.mark.parametrize("test_input, expected",
[ ("3+5", 8),
("2+4", 6),
("6
* 9", 42),
])
def
test_eval(test_input, expected):
assert eval(test_input) == expected
if name == "main":
pytest.main(["-s", "test_canshu1.py"])
pytest 里面还有一个更加强大的功能，获得多个参数化参数的所有组合，可以
堆叠参数化装饰器
import pytest
@pytest.mark.parametrize("x",
[0,
1])

```

```
@pytest.mark.parametrize("y",
[2,
3])
def test_foo(x, y):
print("测试数据组合: x->%s, y->%s" % (x, y))
if name == "main":
pytest.main(["-s", "test_canshul.py"])
```

#### 4. 下个接口请求参数依赖上个接口的返回数据

这个很容易，不同的接口封装成不同的函数或方法，需要的数据 return 出来，用一个中间变量 a 去接受，后面的接口传 a 就可以了

参考这篇【python 接口自动化 26-参数关联和 JSESSIONID(上个接口返回数据作为下个接口请求参数)】

#### 5. 依赖于登录的接口如何处理

登录接口依赖 token 的，可以先登录后，token 存到一个 yaml 或者 json，或者

ini 的配置文件里面，后面所有的请求去拿这个数据就可以全局使用了

参考之前分享的一篇 python 接口自动化 24-有 token 的接口项目使用

unittest

框架设计

如果是 cookies 的参数，可以用 session 自动关联

```
s=requests.session()
```

后面请求用 s.get() 和 s.post() 就可以自动关联 cookies 了

#### 6. 依赖第三方的接口如何处理

这个需要自己去搭建一个 mock 服务，模拟接口返回数据，

moco 是一个开源的框架，在 github 上可以下载到

<https://github.com/dreamhead/moco>

moco 服务搭建需要自己能够熟练掌握，面试会问你具体如何搭建，如何模拟返回的数据，是用的什么格式，如何请求的

#### 7. 不可逆的操作，如何处理，比如删除一个订单这种接口如何测试

此题考的是造数据的能力，接口的请求数据，很多都是需要依赖前面一个状态的

比如工作流这种，流向不同的人状态不一样，操作权限不一样，测试的时候，每种状态都要测到，就需要自己会造数据了。

平常手工测试造数据，直接在数据库改字段状态。那么自动化也是一样，造数据可以用 python 连数据库了，做增删改查的操作

测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

#### 8. 接口产生的垃圾数据如何清理

跟上面一样，造数据和数据清理，需用 python 连数据库了，做增删改查的操作测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

#### 9. 一个订单的几种状态如何全部测到，如：未处理，处理中，处理失败，处理成功

跟上面一样，也是考察造数据，修改数据的状态

## 10. python 如何连接数据库操作？

这个就是详细的考察你是如何用 python 连数据库的，并且最好能现场写代码那种（有的笔试题就是 python 连数据库）

具体问你用到哪个模块，查询的数据是什么类型？如何删除数据？如何新增数据？如何修改数据？

PyMySQL 是在 Python3.x 版本中用于连接 MySQL 服务器的一个库，Python2 中则使用 mysqlldb。

详情参考教程 <http://www.runoob.com/python3/python3-mysql.html>

```
#!/usr/bin/python3
```

```
# 查询 EMPLOYEE 表中 salary（工资）字段大于 1000 的所有数据：
```

```
import pymysql
```

```
# 打开数据库连接
```

```
db = pymysql.connect("localhost","testuser","test123","TESTDB" )
```

```
# 使用 cursor() 方法获取操作游标
```

```
cursor = db.cursor()
```

```
# SQL
```

```
查询语句
```

```
sql =
```

```
"SELECT * FROM
```

```
EMPLOYEE
```

```
WHERE
```

```
INCOME > %s" %
```

```
(1000)
```

```
try:
```

```
# 执行 SQL 语句
```

```
cursor.execute(sql)
```

```
# 获取所有记录列表
```

```
results
```

```
= cursor.fetchall()
```

```
for row
```

```
in results:
```

```
fname
```

```
=
```

```
row[0]
```

```
lname
```

```
=
```

```
row[1]
```

```
age
```

```
=
```

```
row[2]
```

```
sex
```

```
=
```

```
row[3]
income = row[4]
# 打印结果
print ("fname=%s,lname=%s,age=%s,sex=%s,income=%s" %
      (fname, lname,
       age, sex,
       income ))
except:
print ("Error:
unable to

fetch data")
```

```
# 关闭数据库连接
db.close()
```

其它的就是运行出报告、代码管理（git）、运行策略和持续集成 jenkins 相关了，这个所以的自动化但是一样的，后面会单独讲一篇 jenkins 持续集成相关在 API 测试中测试的常用协议是什么？

HTTP

JMS

REST

SOAP

UDDI

接口测试的步骤有哪些？

1) 发送接口请求

2) 测试接口获取返回值

3) 断言：判断实际结果是否符合预期

接口测试中依赖登录状态的接口如何测试？

依赖登录状态的接口，本质上是在每次发送请求时需要带上存储有账户有效信息的 Session 或 Cookie 才能发送成功，在构建 POST 请求时添加必要的

Session

或 Cookie

依赖于第三方数据的接口如何进行测试？

可以利用一些 MOCK 工具（如：JSON Server、Easy Mock）来模拟第三方的数据返回，最大限度的降低对第三方数据接口的依赖

解释什么是 SOAP？

SOAP 代表简单对象访问控制，它是一种基于 XML 的协议，用于在计算机之间交换信息。

解释什么是 REST API？

这是开发人员执行请求并接收响应的一组功能。在 REST API 中，通过 HTTP 协议进行交互

REST - 代表状态转移，它正快速成为 API 创建的标准。

API

Builder 如何工作？

API

Builder 是一个由四个 SQL 文件组成的 PLSQL 程序

要设置 API 参数并启动该过程，一个文件是负责的

为临时表和主包创建两个文件以创建输出的代码

第四个文件将代码的“假脱机”输出创建为一个名为“output\_script\_.sql”的文件

解释什么是 TestApi?

TestApi 是一个实用程序和测试 API 库，可让测试人员和开发人员为 .NET 和 Win32 应用程序创建测试工具和自动测试。它提供了一套常见的测试构建块，类型，数据结构和算法。

什么是使用 runscope 的 API 测试?

Runscope 是一个 Web 应用程序，它提供后端服务和易于使用的界面来测试 API。

解释 API 测试设计的原理是什么?

API 测试设计的原理是

安装：创建对象，启动服务，初始化数据等

执行：执行 API 或场景的步骤，也记录日志

验证：比较预期结果和实际结果

报告：查看执行情况，通过，失败或阻止

清理：回到最初的测试状态

API 测试发现的 Bug 类型是什么?

缺少或重复的功能

无法正常处理错误条件

强调

可靠性

安全

未使用的标志

未实现错误

错误处理不一致

性能

多线程问题

错误不正确

我们测试的接口属于哪一类?

服务器接口（基于 HTTP 协议的接口）

大多数人常说的接口测试，通常是 B/S 架构，由客户端（浏览器）调用，或模拟客户端（浏览器）调用服务器提供的请求接口，由服务器完成处理并返回一个应答的过程。

例如：Webservice 接口，http 接口，jms 接口，hessian 接口。

**TCP 长连接和短连接的区别**

**TCP 长连接和短连接的区别**

当网络通信时采用 TCP 协议时，在真正的读写操作之前，server 与 client 之间必须建立一个连接，当读写操作完成后，双方不再需要这个连接时它们可以释放这个连接，连接的建立是需要三次握手的，而释放则需要 4 次挥手，所以



说每个连接的建立都是需要资源消耗和时间消耗的

### 长连接：

所谓长连接，指在一个 TCP 连接上可以连续发送多个数据包，在 TCP 连接保持期间，如果没有数据包发送，需要双方发检测包以维持此连接，一般需要自己做在线维持（不发生 RST 包和四次挥手）。

连接→数据传输→保持连接(心跳)→数据传输→保持连接(心跳)→……→关闭连接（一个 TCP 连接通道多个读写通信）；

这就要求长连接在没有数据通信时，定时发送数据包(心跳)，以维持连接状态；TCP 保活功能，保活功能主要为服务器应用提供，服务器应用希望知道客户主机是否崩溃，从而可以代表客户使用资源。如果客户已经消失，使得服务器上保留一个半开放连接，而服务器又在等待来自客户端的数据，则服务器将应远等待客户端的数据，保活功能就是试图在服务器端检测到这种半开放连接。如果一个给定的连接在两小时内没有任何的动作，则服务器就向客户发一个探测报文段，客户主机必须处于以下 4 个状态之一：

1.

a.

客户主机依然正常运行，并从服务器可达。客户的 TCP 响应正常，而服务器也知道对方是正常的，服务器在两小时后将保活定时器复位。

a.

客户主机已经崩溃，并且关闭或者正在重新启动。在任何一种情况下，客户的 TCP 都没有响应。服务端将不能收到对探测的响应，并在 75 秒后

超时。服务器总共发送 10 个这样的探测

，每个间隔 75 秒。如果服务

器没有收到一个响应，它就认为客户主机已经关闭并终止连接。

a.

客户主机崩溃并已经重新启动。服务器将收到一个对其保活探测的响应，这个响应是一个复位，使得服务器终止这个连接。

a.

客户机正常运行，但是服务器不可达，这种情况与 2 类似，TCP 能发现的就是没有收到探查的响应。

### 短连接：

短连接是指通信双方有数据交互时，就建立一个 TCP 连接，数据发送完成后，则断开此 TCP 连接（管理起来比较简单，存在的连接都是有用的连接，不需要

额外的控制手段）；

连接→数据传输→关闭连接；

应用场景：

长连接多用于操作频繁（读写），点对点的通讯，而且连接数不能太多情况，。每个 TCP 连接都需要三步握手，这需要时间，如果每个操作都是先连接，再操作的话那么处理速度会降低很多，所以每个操作完后都不断开，次处理时直接发送数据包就 OK 了，不用建立 TCP 连接。例如：数据库的连接用长连接，如果用短连接频繁的通信会造成 socket 错误，而且频繁的 socket 创建也是对资

源的浪费。

而像 WEB 网站的 http 服务一般都用短链接（http1.0 只支持短连接，1.1keep alive 带时间，操作次数限制的长连接），因为长连接对于服务端来说会耗费一定的资源，而像 WEB 网站这么频繁的成千上万甚至上亿客户端的连接用短连接会更省一些资源，如果用长连接，而且同时有成千上万的用户，如果每个用户都占用一个连接的话，那可想而知吧。所以并发量大，但每个用户无需频繁操作情况下需用短连好；

在长连接中一般是没有条件能够判断读写什么时候结束，所以必须要加长度报文头。读函数先是读取报文头的长度，再根据这个长度去读相应长度的报文。

心跳 Demo

```
package socket.keepalive.test;
import
java.io.Serializable;
import
java.text.SimpleDateFormat;
import
java.util.Date;
/**
 *
 * 维持连接的消息对象（心跳对象）
 */
public class KeepAlive implements Serializable{
private static final long serialVersionUID = -2813120366138988480L;
/* 覆盖该方法，仅用于测试使用。
 * @see java.lang.Object#toString()
 */

@Override

public
String toString() {
return
new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new
Date())+"\t 维持连接包";
}
}
```

Client Demo

```
package socket.keepalive.test;
import
java.io.IOException;
import
java.io.InputStream;
import
java.io.ObjectInputStream;
import
```

```

java.io.ObjectOutputStream;
import
java.net.Socket;
import
java.net.UnknownHostException;
import
java.util.Collections;
import
java.util.concurrent.ConcurrentHashMap;
/**
 *
 * C/S 架构的客户端对象，持有该对象，可以随时向服务端发送消息。
 *
 * 创建时间：2010-7-18 上午 12:17:25
 *
 * @author yifan
 *
 * @since 1.0
 */
public class Client {
/**
 * 处理服务端发回的对象，可实现该接口。
 */
public static interface ObjectAction{
void doAction(Object obj,Client client);
}
public
static final class DefaultObjectAction implements ObjectAction{
public
void doAction(Object obj,Client client) {
System.out.println("处理: \t"+obj.toString());
}
}

public static void main(String[] args) throws UnknownHostException,
IOException {
String serverIp = "127.0.0.1";
int port = 65432;
Client client = new Client(serverIp,port);
client.start();
}
private
String serverIp;

```

```

private
int port;
private
Socket socket;
private
boolean running=false; //连接状态
private
long lastSendTime; //最后一次发送数据的时间
//用于保存接收消息对象类型及该类型消息处理的对象
private ConcurrentHashMap<Class, ObjectAction> actionMapping
= new
ConcurrentHashMap<Class, ObjectAction>();
public Client(String serverIp, int port) {
this.serverIp=serverIp;
this.port=port;
}
public void start() throws UnknownHostException, IOException
{
if(running)return;
socket = new Socket(serverIp, port);
System.out.println("本地端口: "+socket.getLocalPort());
lastSendTime=System.currentTimeMillis();
running=true;
new Thread(new KeepAliveWatchDog()).start();
//保持长连接的线程, 每
隔 2 秒向服务器发一个一个保持连接的心跳消息
new Thread(new ReceiveWatchDog()).start();
//接受消息的线程, 处理消
息
}
public void stop() {
if(running)running=false;
}

/**
 *
 * 添加接收对象的处理对象。
 *
 * @param
 * cls 待处理的对象, 其所属的类。
 *
 * @param
 * action 处理过程对象。
 */

```

```

public
void
addActionMap(Class
public
void
sendObject(Object obj) throws IOException {
    ObjectOutputStream oos = new
    ObjectOutputStream(socket.getOutputStream());
    oos.writeObject(obj);
    System.out.println("发送: \t"+obj);
    oos.flush();
}
class KeepAliveWatchDog implements Runnable{
    long
    checkDelay = 10;
    long
    keepAliveDelay = 2000;
    public void run() {
        while(running){
            if(System.currentTimeMillis()-lastSendTime>keepAliveDelay){
                try {
                    Client.this.sendObject(new KeepAlive());
                } catch (IOException e) {
                    e.printStackTrace();
                    Client.this.stop();
                }
                lastSendTime = System.currentTimeMillis();
            }else{
                try {
                    Thread.sleep(checkDelay);
                } catch (InterruptedException e)
                {
                    e.printStackTrace();
                    Client.this.stop();
                }
            }
        }
    }
}
class ReceiveWatchDog implements
Runnable{

    public void run() {

```

```
while(running) {
    try {
        InputStream in = socket.getInputStream();
        if(in.available()>0) {
            ObjectInputStream ois = new ObjectInputStream(in);
            Object obj = ois.readObject();
            System.out.println("接收: \t"+obj);
            ObjectAction oa =
            actionMapping.get(obj.getClass());
            oa = oa==null?new
            DefaultObjectAction():oa;
            oa.doAction(obj, Client.this);
        }else{
            Thread.sleep(10);
        }
    } catch (Exception e) {
        e.printStackTrace();
        Client.this.stop();
    }
}
}
```

服务端 Demo

```
package socket.keepalive.test;
import
java.io.IOException;
import
java.io.InputStream;
import
java.io.ObjectInputStream;
import
java.io.ObjectOutputStream;
import
java.net.ServerSocket;
import
java.net.Socket;
import
java.util.concurrent.ConcurrentHashMap;
/**
 *
 * C/S 架构的服务端对象。
 *
 *
 * 创建时间：2010-7-18 上午 12:17:37
```

```

*
@author yifan
*
@since 1.0
*/

public class Server {

    /**
     * 要处理客户端发来的对象，并返回一个对象，可实现该接口。
     */
    public
    interface ObjectAction{
        Object
        doAction(Object rev, Server server);
    }
    public
    static
    final class DefaultObjectAction implements ObjectAction{
        public
        Object
        doAction(Object rev, Server server) {
            System.out.println("处理并返回: "+rev);
            return
            rev;
        }
    }
    public
    static void main(String[] args) {
        int port = 65432;
        Server server = new Server(port);
        server.start();
    }
    private
    int port;
    private
    volatile boolean running=false;
    private
    long receiveTimeDelay=3000;
    private
    ConcurrentHashMap<Class, ObjectAction> actionMapping = new
    ConcurrentHashMap<Class, ObjectAction>();
    private Thread connWatchDog;
    public Server(int port) {
        this.port =

```

```

port;
}

public void
start() {
    if(running)return;
    running=true;
    connWatchDog = new Thread(new ConnWatchDog());
    connWatchDog.start();
}

@SuppressWarnings("deprecation")
public void stop() {

    if(running)running=false;

    if(connWatchDog!=null)connWatchDog.stop();
}

public void addActionMap(Class
class ConnWatchDog implements Runnable{
    public void run() {
        try {
            ServerSocket ss = new ServerSocket(port,5);
            while(running){
                Socket s = ss.accept();
                new Thread(new SocketAction(s)).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
            Server.this.stop();
        }
    }
}

class SocketAction implements Runnable{
    Socket s;
    boolean run=true;
    long lastReceiveTime = System.currentTimeMillis();
    public
    SocketAction(Socket s) {
        this.s
        = s;
    }

    public
    void run() {
        while(running && run){
            if(System.currentTimeMillis()-lastReceiveTime>receiveTimeDelay){
                overThis();
            }
        }
    }
}

```



```

    }else{
    try {
    InputStream in = s.getInputStream();
    if(in.available()>0){
    ObjectInputStream ois = new ObjectInputStream(in);
    Object obj = ois.readObject();
    lastReceiveTime = System.currentTimeMillis();
    System.out.println("接收: \t"+obj);
    ObjectAction oa =
    actionMapping.get(obj.getClass());
    oa = oa==null?new
    DefaultObjectAction():oa;

    Object out = oa.doAction(obj, Server.this);
    if(out!=null){
    ObjectOutputStream oos =
    new ObjectOutputStream(s.getOutputStream());
    oos.writeObject(out);
    oos.flush();
    }
    }else{
    Thread.sleep(10);
    }
    } catch (Exception e) {
    e.printStackTrace();
    overThis();
    }
    }
    }
    private void overThis()
    {
    if(run)run=false;
    if(s!=null){
    try {
    s.close();
    } catch (IOException e)
    {
    e.printStackTrace();
    }
    }
    System.out.println("关闭: "+s.getRemoteSocketAddress());
    }
    }

```

}

代码参考自：<http://blog.csdn.net/zmx729618/article/details/54378259>

Session 和 cookie 的区别

cookie

cookie 的原理和作用

cookie 的原理

什么是 cookie

分类两种

内存 cookie

相关的 cookie 信息存放在浏览器的缓存中

如果浏览器关闭后，cookie 信息被释放

硬盘 cookie

cookie 信息是保存在客户端机器的硬盘中

永久保存，可以反复读取，除非用户执行删除操作

易用性好

安全性差

为什么要有 cookie，它的作用是什么？

cookie 有什么作用？

cookie 可以解决 http 的无状态的问题，与服务器进行交互，作为 http 规范存在。它具有极高的简便性、可扩展性和可用性，也可以通过加密和 SSL 技术来提高其安全性。因此推荐使用 cookie 作为标识而不是身份验证的工具。其中 cookie 的作用就是为了解决 HTTP 协议无状态的缺陷所作出的努力

cookie 测试点

Cookie 测试的测试点

1. 禁止使用 Cookie

设置浏览器禁止使用 Cookie，访问网页后，检查存放 Cookie 文件中未生成相关文件；

2.Cookie 存储路径

按照操作系统和浏览器对 Cookie 存放路径的设置，检查存放路径是否与设置一致；

3.Cookie 过期检查

按照 Cookie 过期时间，检查存放文件该 Cookie 是否被自动删除

4. 检查浏览器中 Cookie 选项

通过不同浏览器，设置是否接受 Cookie 文件，如同意接受 Cookie，检查存放路径中是否存在 Cookie 文件

5. 浏览器删除 Cookie

通过浏览器的设置，删除 Cookie 文件

6.Cookie 加密

提交敏感信息时，数据应加密

7.Cookie 保存信息

验证 Cookie 能正常工作

8. 篡改 Cookie

修改 Cookie 内容，查看系统功能是否出现异常，或数据错乱

#### 9.Cookie 的兼容性

使用不同类型，或同一类型不同版本的浏览器，检查 cookie 文件的兼容性

#### 10.刷新操作对 cookie 的影响

进行刷新操作后，是否重新生成 cookie 文件或是对 cookie 文件进行修改

#### 11.检查 cookie 内容存储是否完整正确

若 cookie 进行了加密，先对 cookie 文件内容进行解密，然后检查是否按照设计要求存储了相关所有的 cookie 记录信息。

#### 12.对应硬盘存储空间没有空闲时，是否能进行 cookie 内容的有效存储

#### 13.多次做相同的操作或设置，检查是否更新或添加了新的 cookie 文件

按照设计要求进行判断

#### 14.如果使用 cookie 来统计次数，则要检测是否统计正确

例如通过用户登录次数进行统计

1, 访问网站，每次都需要输入用户名和密码

分第一次登陆还是非第一次登录

#### cookie 的缺点

(1) 大小和数目受限制。浏览器对一个域 cookie 的条目数有上限要求，且每个 cookie 的大小不得超过 4kb。

(2) 存在安全性问题，易被人拦截。

(3) 需要指定域，不可以跨域

(4) 浪费带宽，因为我每次请求一个新的页面，cookie 都会被自动发送过去。

(5) 有的移动端浏览器不支持 cookie 或浏览器禁用 cookie

(6) 有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

#### session

#### session 的作用

1, 为什么要是用 cookie

a:方便用户进行登录

b: 收集用户的相关登录时间，时长，使用习惯

2, 为什么要是使用 session

s1: 登录网站后

s2: 跳转到其他页面

其他协议

无状态协议

不记录用户身份信息

产生的问题

1, 只要

2, 任意人都

s3: 产生了 session 技术处理

3, session 的原理

1: 当用户注册后

2, 在本地生成 cookie 文件, 保留用户信息

3, 用户登录

1, 系统检查本地是否存在 cookie 文件

若不存在, 等待登录后,

在本地创建 cookie 文件

2, 若已存在, 系统在用户登录的同时, 由服务器端口自动随机分配一个 sessionId 传入 cookie 文件

4, 若用户打开某个需要权限设置的页面, 系统回去检查对应的 sessionId 是否存在有效正确

session 机制则是又一种在客户端与服务器之间保持状态的解决方案。

session 的测试点

用户登录系统时, 服务器给发送随机的 sessionId

页面跳转时进行身份的验证

测试前沟通准备工作

cookie 与 session 的区别:

1、cookie 数据存放在客户的浏览器上, session 数据放在服务器上。

2、cookie 不是很安全, 别人可以分析存放在本地的 cookie 并进行 cookie 欺骗考虑到安全应当使用 session。

3、session 会在一定时间内保存在服务器上。当访问增多, 会比较占用你服务器的性能考虑到减轻服务器性能方面, 应当使用 cookie。

4、单个 cookie 保存的数据不能超过 4K, 很多浏览器都限制一个站点最多保存 20 个 cookie。

5、所以个人建议:

将登陆信息等重要信息存放为 session

其他信息如果需要保留, 可以放在 cookie 中

作业

知道 cookie 和 session 的关系

知道 cookie 是如何产生的

知道 sessionId 的产生和作用

知道 cookie 上的重要字段和作用

**问题一:**

你在测试中发现了一个 bug, 但是开发经理认为这不是一个 bug, 你应该怎样解决?

**答案:**

首先, 将问题提交到缺陷管理库里面进行备案;

然后, 要获取判断的依据和标准:

根据需求说明书、产品说明、设计文档等, 确认实际结果是否与计划有不一致的地方, 提供缺陷是否确认的直接依据;

如果没有文档依据, 可以根据类似软件的一般特性来说明是否存在不一致的地方, 来确认是否是缺陷;

根据用户的一般使用习惯，来确认是否是缺陷；与设计人员、开发人员和客户代表等相关人员探讨，确认是否是缺陷；合理的论述，向测试经理说明自己的判断的理由，注意客观、严谨，不参杂个人情绪。

等待测试经理做出最终决定，如果仍然存在争议，可以通过公司政策所提供的渠道，向上级反映，并由上级做出决定。

#### 问题二：

软件生命周期及其模型是什么？

#### 答案：

软件生命周期是指从形成开发软件概念起，所开发的软件使用以后，直到失去使用价值消亡为止的整个过程。一般来说，整个生存周期包括计划(定义)、开发、运行(维护)三个时期，每个时期又划分为若干个阶段。每个阶段有明确的任务。

#### 周期模型(典型的几种)：

**瀑布模型 快速原型模型：**快速原型模型允许在需求分析阶段对软件的需求进行初步而非完全的分析和定义，快速设计开发出软件系统的原型，该原型向用户展示待开发软件的全部或部分功能和性能；用户对该原型进行测试评定，给出具体改进意见以丰富细化软件需求；开发人员据此对软件进行修改完善，直至用户满意认可之后，进行软件的完整实现及测试、维护。

**迭代模型：**迭代包括产生产品发布(稳定、可执行的产品版本)的全部开发活动和要使用该发布必需的所有其他外围元素。在某种程度上，开发迭代是一次

完整地经过所有工作流程的过程：需求分析、设计、实施和测试工作流程。

实质上，它类似小型的瀑布式项目。RUP 认为，所有的阶段都可以细分为迭代。每一次

的迭代都会产生一个可以发布的产品，这个产品是最终产品的一个子集。

#### 生命周期阶段：

软件计划与可行性分析

需求分析

软件设计

编码

软件测试

运行与维护

#### 问题三：

什么是软件测试？软件测试的目的与原则？

#### 答案：

在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

#### 软件测试的目的：

测试是程序的执

行过程，目的在于发现错误；

一个成功的测试用例在于发现至今未发现的错误；

一个成功的测试是发现了至今未发现的错误的测试；

确保产品

完成了它所承诺或公布的功能，并且用户可以访问到的功能都有明确的书面说明；

确保产品满足性能和效率的要求；

确保产品是健壮的和适应用户环

境的；

**软件测试的原则：**

测试用例中一个必须部分是对预期输出或结果进行定义；

程序员应避免测试自己编写的程序；

编写软件的组织不应

当测试自己编写的软件；

应当彻底检查每个测试的执行结果；

测试用

例的编写不仅应当根据有效和预料到的输入情况，而且也应当根据无效和未预料到的输入情况；

检查程序是否“未做其应该做的”仅是测试的一半，测

试的另一半是检查程序是否“做了其不应该做的”；

应避免测试用例用后即

弃，除非软件本身就是个一次性的软件；

计划测试工作时不应默许假定不

会发现错误；

程序某部分存在更多错误的可能性，与该部分已经发现错误的

数量成正比；

软件测试是一项极富创造性，极具智力的挑战性的工作；

**问题四：**

目前主要的测试用例设计方法是什么？

**答案：**

白盒测试：逻辑覆盖、循环覆盖、基本路径覆盖(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)

黑盒测试：边

界值分析法、等价类划分、错误猜测法、因果图法、状态图法、测试大纲法、随机测试、场景法

**问题五：**

什么是测试用例

，什么是测试脚本？两者的关系是什么？

**答案：**

为实施测试而向被测试系统提供的输入数据、操作或各种环境设置以及期望结果的一个特定的集合；

测试脚本是为了进行自动化测试而编写的脚本； 测

试脚本的编写必须对应相应的测试用例。

**问题六：**

软件产品质量特性是什么？

**答案：**

功能性：适应性、准确性、互操作性、依从性、安全性。

可靠性：成熟性、

容错性、易恢复性。 可使用性：易理解性、易学习性、易操作性。 效率：时间特性、资源特性。 可维护性：易分析性、易变更性、稳定性、易测试性。

可

移植性：适应性、易安装性、遵循性、易替换性

**问题七：**

软件测试分为几个阶段

，各阶段的测试策略和要求是什么？

**答案：**

和开发过程相对应，测试过程会依次经历单元测试、集成测试、系统测试、验收测试四个主要阶段：

**单元测试：**单元测试是针对软件设计的最小单位——

程序模块甚至代码段进行正确性检验的测试工作，通常由开发人员进行。

**集**

**成测试：**集成测试是将模块按照设计要求组装起来进行测试，主要目的是发现与接口有关的问题。由于在产品提交到测试部门前，产品开发小组都要进行联合调试，因此在大部分企业中集成测试是由开发人员来完成的。

**系统测试：**

系统测试是在集成测试通过后进行的，目的是充分运行系统，验证各子系统是否都能正常工作并完成设计的要求。它主要由测试部门进行，是测试部门最大最重要的一个测试，对产品的质量有重大的影响。

**验收测试：**验收测试以需

求阶段的《需求规格说明书》为验收标准，测试时要求模拟实际用户的运行环境。对于实际项目可以和客户共同进行，对于产品来说就是最后一次的系统测试。测试内容为对功能模块的全面测试，尤其要进行文档测试。

**单元测试测**

**试策略：**

自顶向下的单元测试策略：比孤立单元测试的成本高很多，不是单

元测试的一个好的选择。自底向上的单元测试策略：比较合理的单元测试策略，但测试周期较长。

孤立单元测试策略：最好的单元测试策略。 **集成测试的**

**测试策略：**

大爆炸集成：适应于一个维护型项目或被测试系统较小项目

自

顶向下集成：适应于产品控制结构比较清晰和稳定；高层接口变化较小；底层接口未定义或经常可能被修改；产品控制组件具有较大的技术风险，需要尽早被验证；希望尽早能看到产品的系统功能行为。

自底向上集成：适应于底层接口

比较稳定；高层接口变化比较频繁；底层组件较早被完成。

基于进度的集成优点：具有较高的并行度；能够有效缩短项目的开发进度。

缺点：桩和驱动工作量较大；有些接口测试不充分；有些测试重复和浪费。



系统测试的测试策略：

数据和数据库完整性测试；功能测试；用户界面测试；

性能评测；负载测试；强度测试；容量测试；安全性和访问控制测试；故障转移和恢复测试；配置测试；安装测试；加密测试；可用性测试；版本验证测试；文档测试

**问题八：**

您认为做好测试用例设计工作的关键是什么？

**答案：**

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果；

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题；

**问题九：**

您认为做好测试计划工作的关键是什么？

**答案：**

1. 明确测试的目标，增强测试计划的实用性

编写软件测试计划重要目的就是

使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

2. 坚持“5W”规则，明确内

容与过程

“5W”规则指的是“What(做什么)”、“Why(为什么做)”、“When(何时做)”、“Where(在哪里)”、“How(如何做)”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的(Why)，明确测试的范围和内容(What)，确定测试的开始和结束日期(When)，指出测试的方法和工具(How)，给出测试文档和软件的存放位置(Where)。

3. 采用评审和更新机制，保证测试

计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测

试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

4.

分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含

到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

**问题十：**

您所熟悉的测试用例设计方法都有哪些？

**答案：**

1. 等价类划分

等价类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是等效的。并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试。因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，就可以用少量代表性的测试数据。取得较好的测试结果。等价类划分可有两种不同的情况：有效等价类和无效等价类。

## 2. 边界值分析法

边界值分析方法是等价类划分方法的补充。测试工作经验告诉我，大量的错误是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。

使用边界值法首先应确定边界情况。通常输入和输出等价类的边界，就是应着重测试的边界情况。应当选取正好等于，刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值作为测试数据。

## 3. 比较法

不同的软件版本比较

**4. 错误推测法** 基于经验和直觉推测程序中所有可能存在的各种错误，从而

有针对性的设计测试用例的方法。

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。

## 5. 因果图方法

前面介绍的等价类划分方法和边界值分析方法，都是着重考虑

输入条件，但未考虑输入条件之间的联系，

相互组合等。考虑输入条件之间

的相互组合，可能会产生一些新的情况。但要检查输入条件的组合不是一件容易的事情，

即使把所有输入条件划分成等价类，他们之间的组合情况也相当

多。因此必须考虑采用一种适合于描述对于多种条件的组合，相应产生多个动作的形式来考虑设计测试用例。这就需要利用因果图(逻辑模型)。因果图方法最终生成的就是判定表。它适合于检查程序输入条件的各种组合情况。

## 最后：

有一些面试题来源于网络，感谢前辈的解题。欢迎大家关注我的公众号程序员一凡获取更多软件测试大厂面试、技术进阶资

料。如果对你有帮助，我也会很开心，软件测试，与你同行！



发现网上有关测试的学习资料还是太分散了，不像开发的学习资料那么系统化。希望能帮到刚刚入行的测试新手们，希望能够给大家带来帮助！

(善用 Ctrl+F，快速找到你要的面试题以及答案) 祝大家顺利拿到心仪的 offer。