1. 问题预测
2. 让简单介绍一下 自己

你好，我叫徐涵，来自江苏，毕业于南京晓庄学院计算机科学与技术专业，是18届的毕业生。目前有一年java开发和一年测试经验。最近的一份工作是在南京未来软件，主要负责的工作是未来投标云助手的测试工作。从项目立项开始即加入项目组，全程参与到整个系统的测试，直至上线。

1. 说下自己会的内容

熟悉软件测试基础理论和测试流程，测试方法等。有web测试、接口测试、性能测试经验。

熟悉简单的sql语句，linux基本命令

了解java基础，熟悉开发思想，更容易定位bug

了解python基本语法，结合selenium协助开发简单的爬虫demo

1. 看了哪些书籍

《图解HTTP》、《测试基础》、《软件测试第二版》

1. 了解过哪些技术博客/论坛

51testing论坛

CSDN一些博客（面试经验：面试中会问具体哪些博客），用fiddler做抓包分析详解、python+selenium

公众号（搜狗测试、软件测试资源分享），软件测试资源站（爬虫搬运互联网资源，分享给自学爱好者。主攻软件测试自动化及开发方向）

1. 是否了解软件测试需要哪些知识

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

1. 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，需要储备这些知识

1. 为什么做测试，觉得自己做测试有哪些优势

* 我觉得我个人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。
* 我平时喜欢刷微博、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

1. 知道哪些bug系统

禅道，jrra

1. 测试的基本要素是

版本号、功能模块、优先级别、前置条件、预期结果、实际结果等

1. 介绍一下公司项目
2. 技能方面
3. 数据库方面知识
4. 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：菲关系型数据库也被称为NoSQL数据库，NoSQL的本意是“Not Only SQL”。NoSql为了高性能、高并发而生
2. 其他分类

* 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

* 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，姓名和年龄就会放在一个列族中，而薪资则放在另外一个列族中。

典型：Hbase

* 面向文档数据库，数据存储的最小单位是文档

典型：Mongodb、Hive

1. linux操作

linux搭建测试环境、比如web系统服务搭建

1. 缺陷方面

描述一个你印象深刻的bug

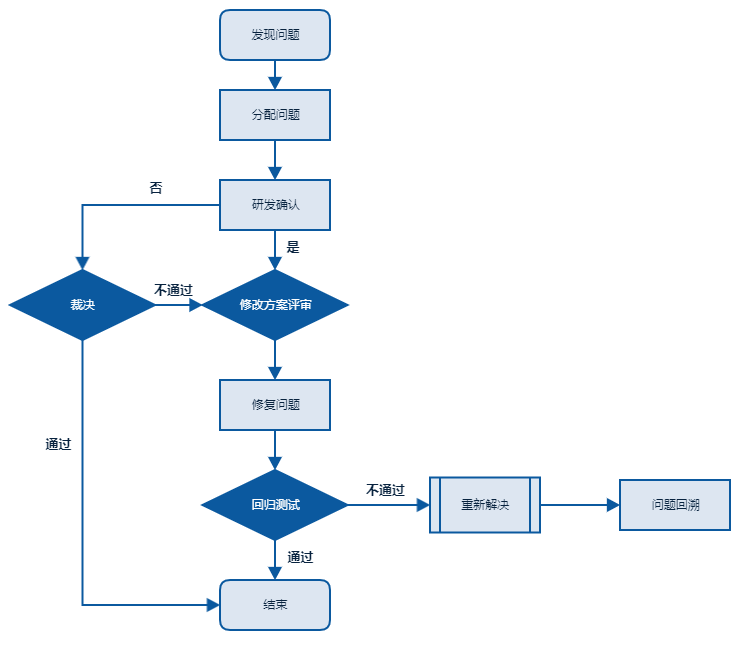
1. 用例部分

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。

1. 软件测试流程



2.1. 发现问题

测试发现问题后，将问题提交到 JIRA 或禅道上

2.2. 分配问题

由 PO 根据当前人员任务情况及问题优先级将问题分配到开发人员上，现网阻塞性的问题优先级最高，需要立即安排。

2.3. 研发确认

研发人员收到问题后，需要先分析问题，确认该 ”问题“ 是否真的是问题。

2.4. 裁决

如果研发认为不是问题，进入裁决环节，此环节需要 PO，产品共同决议这个 ”问题“ 是不是问题。如果确认是，开发人员必须去修改，如果还有异议，可以找研发经理。

如果裁决不是问题，那么直接将问题结束。

2.5. 修改方案评审

如果确认是问题或者裁决不通过进入此环节，在此环节中评审修改问题的方案，重点评审其对现有代码的冲击，是否会将通用性的代码改成特化版，无法再适配开发文档中的其它场景。

这里需要秉持一个原则：如果是架构的问题不适配，那就去改架构，千万不要在通用代码上胡乱打 Patch。如果认为是架构的问题，需要提出详细理由。

2.6. 修复问题

开发人员按照评审后的方案进行修改代码。

2.7. 回归测试

测试人员对该修复的问题进行测试，同时要关注修复的代码对其它功能的影响。

2.8. 重新解决

如果回归不通过，将问题打回，按发现问题之后的流程重新走一遍。

2.9. 问题回溯

由 PO 发起对此次问题回归不通过的讨论会，重点是如何改变，下次不会再犯同样的错误。