

@[toc]

1、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

开发人员说不是bug, 有2种情况,

1. 需求没有确定。可以找来产品经理进行确认需不需要改动, 三方商量确定好后再看要不要改。
2. 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, 那我可以把这个问题提出来, 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。

其实参考答案已经很完整了, 但是可以看到上面的答案明显是偏向测试人员的, 但有时开发说的并没错, 测试要站在对方的角度换位思考。所以回答这个问题还可以从开发人员的角度延伸。

分析什么Bug会让开发认为不是bug

1. 测试人员描述不清晰 工作中也有测试人员把某些“Bug操作步骤”描述的只有自己看得懂, 开发人员按照步骤复现Bug不知所云, 搞错了问题所在。解决方法: 修改Bug操作步骤: 清晰描述、无歧义、无冗余步骤, 要达到即使给一个不懂的人去重现这个Bug, 也能按照你的操作步骤复现。
2. 难以复现的Bug 不是所有的问题都能用同样的操作步骤来复现的, 有的Bug概率出现甚至偶现, 或者是在测试环境里出现。解决办法: 针对难以复现的Bug, 需要保存截图或者记录log保留证据; 对于只在测试环境下才会出现的, 找研发在测试环境进行确认。这类Bug要慎重对待, 规避风险。
3. 有争议的Bug 有争议的Bug多发生于建议类型的Bug: 与同类软件不符、易用性、美观性等类型的Bug。解决办法: 这种问题是否要修改需要根据公司的项目类型进行讨论。开Bug评审会, 在开发能实现的情况下说出自己的理由, 改善产品。
4. 功能性Bug 与需求不符、与原型设计不符。有时候研发对需求没有深入了解可能会忽略或者搞错个别功能。解决办法: 拿证据(需求、设计说明书)给他看, 这种bug自然合情合理。

2、给你一个网站, 你如何测试

1. 首先, 查找需求说明、网站设计等相关文档, 分析测试需求。
2. 制定测试计划, 确定测试范围和测试策略, 一般包括以下几个部分: 功能性测试; 界面测试; 性能测试; 数据库测试; 安全性测试; 兼容性测试
3. 设计测试用例:
 - 功能性测试可以包括, 但不限于以下几个方面:
 - 链接测试。链接是否正确跳转, 是否存在空页面和无效页面, 是否有不正确的出错信息返回。
 - 提交功能的测试。

- 多媒体元素是否可以正确加载和显示。
- 多语言支持是否能够正确显示选择的语言等。
- 界面测试可以包括但不限于一下几个方面：
 - 页面是否风格统一，美观
 - 页面布局是否合理，重点内容和热点内容是否突出
 - 控件是否正常使用
 - 对于必须但未安装的控件，是否提供自动下载并安装的功能
 - 文字检查
- 性能测试一般从以下三个方面考虑：

压力测试；负载测试；强度测试
- 数据库测试要具体决定是否需要开展。数据库一般需要考虑连结性，对数据的存取操作，数据内容的验证等方面。
- 安全性测试：

基本的登录功能的检查

是否存在溢出错误，导致系统崩溃或者权限泄露

相关开发语言的常见安全性问题检查，例如SQL注入等

如果需要高级的安全性测试，确定获得专业安全公司的帮助，外包测试，或者获取支持
- 兼容性测试，根据需求说明的内容，确定支持的平台组合：

浏览器的兼容性；

操作系统的兼容性；

软件平台的兼容性；

数据库的兼容性
- 4. 开展测试，并记录缺陷。合理的安排调整测试进度，提前获取测试所需的资源，建立管理体系（例如，需求变更、风险、配置、测试文档、缺陷报告、人力资源等内容）。
- 5. 定期评审，对测试进行评估和总结，调整测试的内容。

4、一台客户端有三百个客户与三百个客户端有三百个客户对服务器施压，有什么区别？

我目前主要做功能测试，后面希望可以在自动化测试方面有所进步；但是现在对性能测试没有太多研究，不过，根据您的问题，我觉得重点是一个是带宽的问题，一个是单ip连接数的限制，我觉得应该从这两方面来考虑。

5、试述软件的概念和特点?软件复用的含义?构件包括哪些?

1. 软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据、文档的完整集合。
2. 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。
3. 可以被复用的软件成分一般称作可复用构件

6、软件生存周期及其模型是什么？

1. 软件生存周期是软件开发全部过程、活动和任务的结构框架，是从可行性研究到需求分析、软件设计、编码、测试、软件发布维护的过程。
2. 在经历需求、分析、设计、实现、部署后，软件将被使用并进入维护阶段。这样的一个过程，称为"生命周期模型"（Life Cycle Model）。

7、什么是软件测试?软件测试的目的与原则

1. 软件测试是使用人工或自动手段，来运行或测试某个系统的过程。其目的在于检验它是否满足规定的需或弄清预期结果与实际结果之间的差别。
2. 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
3. 测试的原则：教材的说法：软件测试应尽早执行，并贯穿于整个软件生命周期软件测试应追溯需求测试应由第三方来构造穷举测试是不可能的,要遵循 Good-enough 原则必须确定预期输出（或结果）必须彻底检查每个测试结果充分注意测试中的群集现象缺陷的二八定理严格执行测试计划，排除测试的随意性注意合法合理的输入，也要注意非法的非预期的输入检查程序是否做了不该做的测试应从“小规模”开始，逐步转向“大规模”反复使用同样的测试会使软件具有抵抗力关注缺陷的修复

8、软件配置管理的作用?软件配置包括什么？

1. 软件配置管理作为软件开发过程的必要环节和软件开发管理的基础，贯穿整个软件生命周期，同时对软件开发过程的宏观管理即项目管理也有重要的支持作用。一个软件开发组织真正有效的实施软件配置管理，将会使软件开发过程有更好的可预测性，使系统具有可重复性，大大提高软件组织的竞争力。
2. 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

9、什么是软件质量？

1. 软件质量，是指软件系统或系统中的软件部分的质量，即满足用户需求，包括功能需求和性能需求的程度。直白些说，就是软件在功能性，易用性，界面，性能，安全，可移植，可维护等方面满足用户需求，叫用户用着爽。

10、目前主要的测试用例设计方法是什么？

11、软件的安全性应从哪几个方面去测试？

- 用户认证安全的测试要考虑问题：

1. 明确区分系统中不同用户权限

2. 系统中会不会出现用户冲突
 3. 系统会不会因用户的权限的改变造成混乱
 4. 用户登陆密码是否是可见、可复制
 5. 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）
 6. 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
 1. 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
 2. 模拟非授权攻击，看防护系统是否坚固
 3. 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 IPhacker IP）
 4. 采用各种木马检查工具检查系统木马情况
 5. 采用各种防外挂工具检查系统各组程序的客外挂漏洞
 - 数据库安全考虑问题：
 1. 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）
 2. 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）
 3. 系统数据可管理性
 4. 系统数据的独立性
 5. 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）

原文 <http://blog.csdn.net/flm2003/article/details/4359703>

12、什么是测试用例?什么是测试脚本?两者的关系是什么?

- 测试用例是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。
- 两者的关系：测试脚本的编写必须对应相应的测试用例。

13、简述什么是静态测试、动态测试、黑盒测试、白盒测试、 α 测试 β 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。

- α 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。
- β 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。

14、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?

- 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被所有项目所采用。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，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
- 国家标准：
 - i. GB/T 8567-2006 计算机软件文档编制规范
 - ii. GB/T 11457-2006 信息技术 软件工程术语
 - iii. GB/T 16260.1-2006 软件工程 产品质量 第1部分：质量模型
 - iv. GB/T 16260.2-2006 软件工程 产品质量 第2部分：外部度量
 - v. GB/T 16260.3-2006 软件工程 产品质量 第3部分：内部度量
 - vi. GB/T 16260.4-2006 软件工程 产品质量 第4部分：使用质量的度量
 - vii. GB/Z 20156-2006 软件工程 软件生成周期过程 用于项目管理的指南
 - viii. GB/T 20157-2006 信息技术 软件维护
 - ix. GB/T 20158-2006 信息技术 软件生成周期过程 配置管理

15、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

16、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。

17、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集成起来进行测试，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。
- 验收测试的测试策略：

使用自动化测试工具

18、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功能覆盖。该阶段需要提交测试总结和缺陷报告。

19、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。

20、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

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，可选择是否进行图像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。

21、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，而不能知道设计的正确与否，可能会漏掉一些功能需求；系统庞大时，测试开销会非常大。

22、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

22、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）。

23、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。

24、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。

25、BUC管理工具的跟踪过程(用BuzZilla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的问题，则转发出去，由下一个开发人员来进行此行为；如果认为不是问题，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题。

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。

26、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。

27、你对测试最大的兴趣在那里?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。

28、你自认为测试的优势在哪里?

- 该面试也没有固定不变的答案，但可参考以下几点，并结合自身特点：

有韧性、有耐心、做事有条理性、喜欢面对挑战、有信心做好每一件事情、较强的沟通能力、从以前的经理处都得到了很好的评价表明我做的很好

29、简述你在以前的工作中做过哪些事情，比较熟悉什么。

参考答案如下。

- 我过去的主要工作是系统测试和自动化测试。
 - 在系统测试中，主要是对ERP系统的单据审批，以及建筑管理系统的项目管理和财务核算功能进行测试。
 - 性能测试中，主要是进行的压力测试，在各个不同数量请求的情况下，获取系统响应时间以及系统资源消耗情况。
 - 自动化测试主要是通过自己写脚本以及一些第三方工具的结合来测试软交换的特性测试。
- 在测试中，我感觉对用户需求的完全准确的理解非常重要。另外，就是对BUG的管理，要以需求为依据，并不是所有BUG均需要修改。
- 测试工作需要耐心和细致，因为在新版本中，虽然多数原来发现的BUG得到了修复，但原来正确的功能也可能变得不正确。因此要注重迭代测试和回归测试。

30、你的离职原因是什么？

31、通过画因果图来写测试用例的步骤为：

(1)根据程序规格说明书描述，分析并确定因(输入条件)和果(输出结果或程序状态的改变)，画出因果图。

(2)将得到的因果图转换为判定表。

(3)为判定表中每一列所表示的情况设计一个测试用例。

32、APP测试与web测试有什么区别？

单纯从功能测试的层面上来讲的话，APP 测试、web测试 在流程和功能测试上是没有区别的。

根据两者载体不一样，则区别如下：

一、系统结构方面

web项目，b/s架构，基于浏览器的；web测试只要更新了服务器端，客户端就会同步会更新。

app项目，c/s结构的，必须要有客户端；app 修改了服务端，则客户端用户所有核心版本都需要进行回归测试一遍。

二、性能方面

web项目 需监测 响应时间、CPU、Memory

app项目 除了监测 响应时间、CPU、Memory外，还需监测 流量、电量等。

三、兼容方面

1、web项目：

浏览器（火狐、谷歌、IE等）

操作系统（Windows7、Windows10、Linux等）

2、app项目：

设备系统:iOS（ipad、iphone）、Android（三星、华为、联想等）、Windows（Win7、Win8）、OSX（Mac）

手机设备可根据 手机型号、分辨率不同

四、相对于 Web 项目，APP有专项测试

干扰测试：中断，来电，短信，关机，重启等

弱网络测试（模拟2g、3g、4g，wifi网络状态以及丢包情况）；网络切换测试（网络断开后重连、3g切换到4g/wifi 等）

安装、更新、卸载

安装：需考虑安装时的中断、弱网、安装后删除安装文件等情况

卸载：需考虑 卸载后是否删除app相关的文件

更新：分强制更新、非强制更新、增量包更新、断点续传、弱网状态下更新

界面操作：关于手机端测试，需注意手势，横竖屏切换，多点触控，前后台切换

安全测试：安装包是否可反编译代码、安装包是否签名、权限设置，例如访问通讯录等

边界测试：可用存储空间少、没有SD卡/双SD卡、飞行模式、系统时间有误、第三方依赖（QQ、微信登录）等

权限测试：设置某个App是否可以获取该权限，例如是否可访问通讯录、相册、照相机等

五、测试工具方面

1、自动化工具：APP 一般使用 Appium; Web 一般使用Selenium

2、性能测试工具：APP 一般使用 JMeter; Web 一般使用 LR、JMeter

33、说说你对集成测试中自顶向下集成和自底向上集成两个策略的理解，要谈出它们各自的优缺点和主要适用于哪种类型测试？

- 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。

34、软件验收测试包括哪几种？

正式验收测试；

非正式验收测试其中包括 α 测试（由用户、测试人员、开发人员共同参与的内部测试。）和 β 测试（内测后的公测，即完全交给最终用户测试）

35、系统测试的策略有哪些？

- 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。

36、设计系统测试计划需要参考的项目文档有哪些？

软件测试计划，软件需求规范和迭代计划

37、通过画因果图来写测试用例的步骤为____、____、____及把因果图特换为状态图共五个步骤。

- 分析程序规格说明的描述中，哪些是原因，哪些是结果；
- 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；
- 在因果图上使用若干个特殊的符号标明特定的约束条件；
- 把判定表中每一列表示的情况写成测试用例

38、测试的种类很多，比如：代码、函数级测试模块、组件级测试系统测试请说出这些测试最好由那些人员完成，测试的是什么？

代码、函数级测试一般由白盒测试人员完成，他们针对每段代码或函数进行正确性检验，检查其是否正确的实现了规定的功能。

模块、组件级测试主要依据是程序结构设计测试模块间的集成和调用关系，一般由测试人员完成。

系统测试在于模块测试与单元测试的基础上进行测试。了解系统功能与性能，根据测试用例进行全面的测试。

39、设计测试用例时应该考虑那些方面，即不同的测试用例针对那些方面进行测试？

设计测试用例时需要注意的是，除了对整体流程及功能注意外，还要注意强度测试、性能测试、压力测试、边界值测试、稳定性测试、安全性测试等多方面。

测试用例需要考虑的四个基本要素是输入、输出、操作和测试环境；另外，测试用例需要考虑的是测试类型（功能、性能、安全...），这部分可以参照 TP 作答。此外，还需要考虑用例的重要性和优先级

40、在winindows下保存一个文本文件时会弹出保存对话框，如果为文件名建立测试用例，等价类应该怎样划分？

- 单字节，如 A；双字节，AA、我我；

- 特殊字符 /'。'; 、=-等;
- 保留字，如 com;
- 文件格式为 8.3 格式的;
- 文件名格式为非 8.3 格式的;
- /,,等九个特殊字符。

41、假设有一个文本框要求输入10个字符的邮政编码，对于该文本框应该怎样划分等价类？

特殊字符，如 10 个或¥； 英文字母，如 ABCDefghik； 小于十个字符，如 123； 大于十个字符，如 11111111111； 数字和其他混合，如 123AAAAAAA； 空字符； 保留字符

42、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。

43、什么是白盒测试？什么是黑盒测试?? 什么是回归测试？

参考课程内容

数据库实操题目，计算库存量。

表一 AA种类

T	库存总量
SA	997
B	1234

表二 BB种类

T	出库数量
SA	105
B	116
B	303
A	213
B	211

```
select distinct AA.T kind ,AA.S-(select sum(BB.S) sumnum from BB groupby T having AA.T=BB.T) stock from AA,BB where AA.T=BB.T
```


44、单元测试、集成测试、系统测试的侧重点是什么？

白盒测试是测试人员要了解程序结构和处理过程,按照程序内部逻辑测试程序,检查程序中的每条通路是否按照预定要求正确工作。它主要的针对被测程序的源代码,测试者可以完全不考虑程序的功能。

白盒测试流程:详细设计-->源程序-->分析程序内部逻辑结构-->流程图-->制定测试用例-->被测程序-->执行路径-->覆盖情况分析。

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

回归测试: (regression testing): 回归测试有两类: 用例回归和错误回归; 用例回归是过一段时间以后再回头对以前使用过的用例在重新进行测试,看看会重新发现问题。错误回归,就是在新版本中,对以前版本中出现并修复的缺陷进行再次验证,并以缺陷为核心,对相关修改的部分进行测试的方法。

2. 单元测试、集成测试、系统测试的侧重点是什么？

单元测试针对的是软件设计的最小单元--程序模块(面向过程中是函数、过程;面向对象中是类),进行正确性检验的测试工作,在于发现每个程序模块内部可能存在的差错。

一般有两个步骤:人工静态检查\动态执行跟踪集成测试针对的是通过了单元测试的各个模块所集成起来的组件进行检验,其主要内容是各个单元模块之间的接口,以及各个模块集成后所实现的功能。系统测试针对的是集成好的软件系统,作为整个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,要在实际的运行环境中,对计算机系统进行一系列的集成测试和确认测试。

45、一个测试工程师应具备那些素质？

1、责任心 2、沟通能力 3、团队合作精神 4、耐心、细心、信心 5、时时保持怀疑态度,并且有缺陷预防的意识 6、具备一定的编程经验

46、你所了解的的软件测试类型都有哪些,简单介绍一下

按测试策略分类: 1、静态与动态测试 2、黑盒与白盒测试 3、手工和自动测试 4、冒烟测试 5、回归测试; 按测试阶段分类: 单元测试、集成测试、系统测试; 其他常见测试方法: 1、功能测试 2、性能测试 3、压力测试 4、负载测试 5、易用性测试 6、安装测试 7、界面测试 8、配置测试 9、文档测试 10、兼容性测试 11、安全性测试 12、恢复测试

47、你认为做好测试工作的关键是什么？

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)。

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。

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。

48、你认为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。

问题六：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？

性能测试的目的主要是发现在并发多用户和大数据量操作时是否会出现与需求有差异的地方。

性能测试工作的关键是做好系统分析和功能分析，确定系统瓶颈所在（这里参看 ATT第十章 LoadRunner 的 PPT）。

问题七：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呢？

49、你的测试职业发展展目标是什么？

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 x 年累积测试经验，不断的更新自己改正自己，做好测试任务。

问题九：你对我们公司了解有多少？

建议从招聘广告上多了解信息，同时到应聘公司的网站上去尽可能多的了解这个公司的情况，以便回答好这类问题。

50、测试结束的标准是什么？

- 从微观上来说，在测试计划中定义，比如系统在一定性能下平稳运行 72 小时，目前 BugTracking System 中，本版本中没有一般严重的 BUG，普通 BUG 的数量在 3 以下，BUG 修复率 90%以上等等参数，然后由开发经理，测试经理，项目经理共同签字认同版本 Release。
- 如果说宏观的，则是当这个软件彻底的消失以后，测试就结束了。

软件测试分为黑盒和白盒，分别适合什么情况？

软件测试方法一般分为两种：白盒测试与黑盒测试。

白盒测试又称为结构测试、逻辑驱动测试或基于程序本身的测试，它着重于程序的内部结构及算法，通常不关心功能与性能指标；

黑盒测试又被称为功能测试、数据驱动测试或基于规格说明的测试，它实际上是站在最终用户的立场，检验输入输出信息及系统性能指标是否符合规格说明书中有关功能需求及性能需求的规定。

51、一套完整的项目流程应该由哪些阶段组成？

可行性分析、需求分析、概要设计、详细设计、编码、单元测试、集成测试、系统测试、验收测试

4、测试用例通常包括那些内容？

不同结构的用例包括的不一样。（版本、编号、项目、设计人员、设计日期、输入、预期输出,,,）

软件测试用例的基本要素包括测试用例编号、测试标题、重要级别、测试输入、操作步骤、预期结果。

- 用例编号：测试用例的编号有一定的规则，比如系统测试用例的编号这样定义规则：PROJECT1-ST-001，命名规则是项目名称+测试阶段类型（系统测试阶段）+编号。定义测试用例编号，便于查找测试用例，便于测试用例的跟踪。
- 测试标题：对测试用例的描述，测试用例标题应该清楚表达测试用例的用途。比如“测试用户登录时输入错误密码时，软件的响应情况”。
- 重要级别：定义测试用例的优先级别，可以笼统的分为“高”和“低”两个级别。一般来说，如果软件需求的优先级为“高”，那么针对该需求的测试用例优先级也为“高”；反之亦然，一般而言，是5级划分。
- 测试输入：提供测试执行中的各种输入条件。根据需求中的输入条件，确定测试用例的输入。测试用例的输入对软件需求当中的输入有很大的依赖性，如果软件需求中没有很好的定义需求的输入，那么测试用例设计中会遇到很大的障碍。
- 操作步骤：提供测试执行过程的步骤。对于复杂的测试用例，测试用例的输入需要分为几个步骤完成，这部分内容在操作步骤中详细列出。
- 预期结果：提供测试执行的预期结果，预期结果应该根据软件需求中的输出得出。如果在实际测试过程中，得到的实际测试结果与预期结果不符，那么测试不通过；反之则测试通过。

52、你是否了解以往所工作的企业的软件开发过程？如果了解，请试述一个完整的开发过程需要完成哪些工作？分别由那些不同的角色来完成这些工作？你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？

开发过程---需求调研（需求人员）、需求分析（需求人员）、概要设计（设计人员）、详细设计(设计人员)、编码（开发人员）

测试过程---需求评审、系统测试设计、概要设计评审、集成测试设计、详细设计评审、单元测试设计、测试执行测试工作的整个过程都做过，擅长做测试设计过程决定质量，软件的过程改进正是为了提高软件的质量，将过往的种种经验和教训积累起来。

在您所经历的测试活动中，参与人员有哪些？您所担任的角色是什么？

- 有项目管理员、开发管理员、系统分析员、设计员、开发员、质量管理员、测试管理员、测试设计员、测试员
- 担任过测试管理员、测试设计员、测试员

53、测试用例设计的原則是什么?目前主要的测试用例设计方法有哪些?

代表性：能够代表并覆盖各种合理的和不合理、合法和非法的、边界的和越界的、以及极限的输入数据、操作和环境设置等。

可判定性：即测试执行结果的正确性是可判定的，每一个测试用例都应有相应的期望结果。

可再现性：即对同样的测试用例，系统的执行结果应当是相同的。

方法有等价类、边界值、因果图、状态图、正交实验法、判定表等。

54、面向对象的测试用例设计有几种方法?如何实现?

给类中的每个构造函数设计一组测试用例组合类中的类变量、实例变量组合类中的各种方法根据前置条件和后置条件设计测试用例根据代码设计测试用例

55、LoadRunner分为哪三个模块?请简述各模块的主要功能。

Virtual User Generator：用于录制脚步 Mercury LoadRunner Controller：用于创建、运行和监控场景 Mercury LoadRunner Analysis：用于分析测试结果

56、你所熟悉的软件测试类型都有哪些?请試着分別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

57、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。

58、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。

59、一份测试计划应该包括哪些内容？

背景、项目简介、目的、测试范围、测试策略、人员分工、资源要求、进度计划、参考文档、常用术语、提交文档、风险分析。

如果有机会转成开发人员，你会去做开发工作吗？

如果公司确实需要我可以从事开发，但我还是喜欢做测试，我认为我更适合做测试。

七、软件测试分哪些阶段？各阶段的含义？

分为单元测试、集成测试、确认测试、系统测试、验收测试。

(下面的内容是连在一起的，为了方便你们阅读，我把它按逻辑分成了几段)

单元测试是最小单位的测试，测试独立模块；

集成测试主要测试模块之间的接口是否正常，

确认测试类似于冒烟测试通常在大规模系统测试之前验证版本主要功能是否实现，版本的稳定性是否可以进入系统测试，

系统测试是全面测试验证系统是否满足用户需求包括功能、性能、兼容性等等。

验收测试是用户参与的测试。

60、针对于软件的行业背景，你如何理解软件的业务？

阅读用户手册了解软件的功能和操作流程；看一些业务的专业书籍补充业务知识；如果有用户实际的数据，可以拿实际的数据进行参考；参考以前的用例和 BUG 报告；在使用软件的过程中多思考；多与产品经理交流。

61、如何定位测试用例的作用？

组织性：编写、组织性、功能覆盖、重复性、跟踪、测试确认

测试用例应包括哪些内容？

编号、模块名称、编写人、日期、操作说明、输入数据、预期结果等。

测试过程中什么是最重要的？需求、计划。

62、什么是兼容性测试？请举例说明如何利用兼容性测试列表进行测试。

主要验证软件产品在不同版本之间的兼容性。包括向下兼容和交错兼容，向下兼容是测试软件新版本保留它早期版本功能的情况，交错兼容是验证共同存在的两个相关但不相同的产品之间的兼容性。

63、对某软件进行测试，发现在WinXP上运行得很慢，怎么判别是该软件存在问题还是其软硬件运行环境存在问题？

看软件的运行环境要求。如果符合要求则是程序存在问题，若不符合要求则是硬件系统存在问题

64、需求测试的注意事项有哪些？

是否使用了公司的模板文档内容是否符合规范所有的需求是分级是否清晰适当？

所有的需求是否具有 consistency 需求是否可行（即，该需求组合有解决方案）需求可否用已知的约束来实现需求是否足够（即，可以把它送到一个规范的开发组织，并有一个生产出所需要产品的合理的可能性）所有的其它需求是交叉引用是否正确用户描述是否清楚是否用客户的语言来描述需求每个需求描述是否清楚没有歧义，可以移交给一个独立的组去实现时也能理解是否所有的需求都是可验证的是否每条需求都具有独立性，即使发生了变化也不会影响其它需求性能指标是否明确非功能性需求是否得到充分表现是否完整列出适用的标准或协议标准和协议之间是否存在冲突请简述一下缺陷的生命周期。提交□分配□处理□返测□关闭（返测和处理为循环）

65、主键、外键的作用，索引的优点与不足？

- **主键：** 是表中的唯一标示键。作用： 保证实体的完整性;加快数据库的操作速度;增加新的表记录时，数据库会自动检索新记录的主键值，不允许该值与其他表中记录的主键重复;数据库会按主键值的顺序显示记录，如果没有设定主键，则按输入的顺序显示记录。
- **外键：** 是主键的从属，表示了两个表之间的联系。作用： 使用外键可以避免冗余。

索引的优点：

1. 通过创建唯一性的索引，可以保证表中数据的唯一性;
2. 加速数据的检索速度;
3. 加快表与表之间的连接;
4. 在使用分组与排序数据检索时，可以显著检索分组与排序的时间;
5. 在查询的过程中使用优化隐藏器，提供系统性能。

缺点：

1. 创建索引需要时间，且随着数据量的增加而增加;
2. 索引需要占用物理空间;
3. 当对表中数据进行修改时，索引也要动态维护，降低了数据的维护速度。

66、性能测试的流程？

测试前装备（进行需求分析，建立性能目标，制定项目计划和方案等），搭建环境，编写脚本，准备测试数据，执行测试，结果分析，调优，测试报告。

67、简述bug的生命周期?

- 测试人员在执行测试用例的过程中发现bug，将其上报到缺陷管理系统，并指派给开发，开发初步判断缺陷，
- 如果是缺陷并需要修复的，开发修改代码后将代码提交到服务器上，并将bug解决为 已解决，同时回指给测试。
- 测试环境通过最新代码更新后，测试人员验证bug，如果确定已经修复，测试关闭bug；否则，测试重新激活bug。
- 如果开发认为缺陷是重复缺陷，开发会将缺陷解决为重复缺陷，并回指给测试，测试要确定是否确实重复，
- 如果是，测试关闭bug，否则，测试重新 激活 bug。如果在开发环境不能重现bug，开发会将缺陷解决为 不能重现，
- 并回指给测试，测试要确定是否确实不能重现，如果是，测试关闭bug，否则，测试重新 激活 bug。
- 如果开发认为是无效bug，开发会将缺陷解决为 无效【外部原因，设计如此】，并回指给测试，测试要确定是否确实无效，
- 如果是，测试关闭bug，否则，测试重新 激活 bug。如果开发认为缺陷不需要修复，开发会召集产品经理，开发组长，
- 测试人员及测试组长一起讨论，如果大家一致同意该缺陷不需要修复，开发会将缺陷解决为 不予修复，并回指给测试，测试关闭bug。
- 如果讨论之后确定该缺陷需要修复，开发会修改缺陷如果缺陷需要推迟到下一个版本修改，在产品经
- 理，
- 项目经理及其他相关人员同意之后，开发可以将缺陷解决为推迟修改，并回指给测试。

68、缺陷记录应包含的内容?

1. 创建者 reporter
2. 创建时间 date
3. 缺陷版本 version
4. 缺陷类型 type
5. 缺陷所属的产品/项目/模块 product, project, feature
6. 指派给 assign to
7. 缺陷编号 no
8. 标题 title
9. 缺陷严重程度 severity
10. 缺陷优先级 priority
11. 详细描述description
12. 重现率 reproduce rate
13. 步骤 steps
14. 实际结果 actual result
15. 期望结果 expected result
16. 附件 attachment （截图，视频，文件-log日志）

69、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
2. 坚持“5W”规则，明确内容与过程

3. 采用评审和更新机制，保证测试计划满足实际需求

4. 分别创建测试计划与测试详细规格、测试用例

70、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

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。

黑盒测试与白盒测试的区别及他们主要测试的方向？

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作是一个黑盒子，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者数据驱动测试。

- 黑盒测试主要是为了发现以下几类错误：
 - 是否有不正确或遗漏的功能
 - 在接口上，输入是否能正确的接受？能否输出正确的结果
 - 是否有数据结构错误或外部信息(例如数据文件)访问错误
 - 性能上是否能够满足要求
 - 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作一个打开的盒子，它允许测试人员利用程序内部的逻辑结构和有关信息，设计或者选择测试用例，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合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：
 - 对程序模块的所有独立的执行路径至少测试一遍
 - 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
 - 在循环的边界和运行的界限内执行循环体
 - 测试内部数据结构的有效性等等

71、您以往的工作中是否曾开展过测试用例的评审工作?如果有,请描述测试用例评审的过程和评审的内容。

评审计划->预审->评审;

评审内容主要是测试用例对软件需求的覆盖程度，对于相关边界是否考虑，是否针对复杂流程准备多套测试数据，是否有专门针对非功能性需求的测试。

72、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？

关键是测试脚本的录制，测试时候测试环境的干净。

73、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。

QC，也可以使用BugFree等免费工具。

74、您如何看待软件过程改进?在您曾经工作过的企业中，是否有一些需要改进的东西呢?您期望的理想的测试人员的工作环境是怎样的?

将先进的经验或思想固化到过程中，通过过程改进和能力提高来改进软件质量。