

软件[测试工程师](#)，和开发工程师相比起来，虽然前期可能不会太深，但是涉及的面还是比较广的。涉及的知识主要有 MySQL 数据库的使用、Linux 操作系统的使用、软件测试框架性的问题，测试环境搭建问题、当然还有一些自动化测试和性能测试的问题。[测试工程师](#)的面试题，基本上都是大同小异的，面试的核心主要在于框架模块

1、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积累的，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 年积累测试经验，按如何做好[测试工程师](#)的要点去要求自己，不断更新自己改正自己，做好测试任务。

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。

2、你认为测试人员需要具备哪些素质

做测试应该要有一定的协调能力，因为测试人员经常要与开发接触处理一些问题，如果处理不好的话会引起一些冲突，这样的话工作上就会不好做。还有测试人员要有一定的耐心，有的时候做测试很枯燥乏味。除了耐心，测试人员不能放过每一个可能的错误。

3、你为什么能够做测试这一行

虽然我的测试技术还不是很成熟，但是我觉得我还是可以胜任软件测试这个工作的，因为做软件测试不仅是要求技术好，还有有一定的沟通能力，耐心、细心等外在因素。综合起来看我认为我是胜任这个工作的。

4、测试的目的是什么？

测试的目的是找出软件产品中的错误，是软件尽可能的符合用户的要求。当然软件测试是不可能找出全部错误的。

5、测试分为哪几个阶段？

一般来说分为 5 个阶段：单元测试、集成测试、确认测试、系统测试、验收测试

6、单元测试的测试对象、目的、测试依据、测试方法？

测试对象是模块内部的程序错误，目的是消除局部模块逻辑和功能上的错误和缺陷。测试依据是模块的详细设计，测试方法是采用白盒测试。

7、怎样看待加班问题

加班的话我没有太多意见，但是我还是觉得如果能够合理安排时间的话，不会有太多时候加班的。

8、结合你以前的学习和工作经验，你认为如何做好测试。

根据我以前的工作和学习经验，我认为做好工作首先要有一个良好的沟通，只有沟通无障碍了，才会有好的协作，才会有更好的效率，再一个就是技术一定要过关，做测试要有足够的耐心，和一个良好的工作习惯，不懂的就要问，实时与同事沟通这样的话才能做好测试工作。

9、你为什么选择软件测试行业

因为之前了解软件测试这个行业，觉得他的发展前景很好。

10、根据你以前的工作或学习经验描述一下软件开发、测试过程，由哪些角色负责，你做什么

要有架构师、开发经理、测试经理、程序员、测试员。我在里面主要是负责所分到的模块执行测试用例。

11、根据你的经验说说你对软件测试/质量保证的理解

软件质量保证与测试是根据软件开发阶段的规格说明和程序的内部结构而精心设计的一批测试用例(即输入数据和预期的输出结果)，并根据这些测试用例去运行程序，以发现错误的过程。它是对应用程序的各个方面进行测试以检查其功能、语言有效性及其外观排布。

12、软件测试的流程是什么？

需求调查：全面了解系统概况、应用领域、软件开发周期、软件开发环境、开发组织、时间安排、功能需求、性能需求、质量需求及测试要求等。根据系统概况进行项目所需的人员、时间和工作量估计以及项目报价。

制定初步的项目计划。

-

测试准备：组织测试团队、培训、建立测试和管理环境等。

-

-

测试设计：按照测试要求进行每个测试项的测试设计，包括测试用例的设计和测试脚本的开发等。

-

-

测试实施：按照测试计划实施测试。

-

-

测试评估：根据测试的结果，出具测试评估报告。

-

13、你对 SQA 的职责和工作活动(如软件度量)的理解?

SQA 就是独立于软件开发的项目组，通过对软件开发过程的监控，来保证软件的开发流程按照指定的 CMM 规程(如果有相应的 CMM 规程),对于不符合项及时提出建议和改进方案，必要时可以向高层经理汇报以求问题的解决。通过这样的途径来预防缺陷的引入，从而减少后期软件的维护成本。SQA 主要的工作活动包括制定 SQA 工作计划，参与阶段产物的评审，进行过程质量、功能配置及物理配置的[审查](#)等；对项目开发过程中产生的数据进行度量等等。

14、说说你对软件配置管理的理解

项目在开发过程中要用相应的配置管理工具对配置项(包括各个阶段的产物)进行变更控制，配置管理的使用取决于项目规模和复杂性及风险的水平。软件的规模越大，配置管理就越显得重要。还有在配置管理中，有一个很重要的概念，那就是基线，是在一定阶段各个配置项的组合，一个基线就提供了一个正式的标准，

随后的工作便基于此标准，并只有经过授权后才能变更这个标准。配置管理工具主要有 CC，VSS,CVS,SVN 等。

15、怎样写测试计划和测试用例

简单点，测试计划里应有详细的测试策略和测试方法，合理详尽的资源安排等，至于测试用例，那是依赖于需求(包括功能与非功能需求)是否细化到功能点，是否可测试等。

16、什么是兼容性测试？兼容性测试侧重哪些方面？

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。

17、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？

- 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。

18、测试的策略有哪些？

黑盒/白盒，静态/动态## 标题，手工/自动，冒烟测试，回归测试，公测（Beta 测试的策略）

19、你觉得 bugzilla 在使用的过程中，有什么问题？

- 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。

20、描述测试用例设计的完整过程？

- 1、需求分析 + 需求变更的维护工作；
- 2、根据需求得出测试需求；
- 3、设计测试方案，评审测试方案；
- 4、方案评审通过后，设计测试用例，再对测试用例进行评审；

21、单元测试的策略有哪些？

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

22、LoadRunner 分哪三部分？

用户动作设计；场景设计； 测试数据分析；

23、LoadRunner 进行测试的流程？

- 1、 熟悉业务流程，测试规划
- 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。

24、软件的评审一般由哪些人参加？其目的是什么？

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

25、Beta 测试与 Alpha 测试有什么区别？

-Beta testing(β 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 (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

26、你认为做好测试计划工作的关键是什么？

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。

27、你认为做好测试用例工作的关键是什么？

需求和设计文档的理解程度，对系统的熟悉程度

28、简述一下缺陷的生命周期？

提交->确认->分配->修复->验证->关闭

29、软件的安全性应从哪几个方面去测试？

(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

30、你觉得软件测试通过的标准应该是什么样的？

缺陷密度值达到客户的要求

31、一套完整的测试应该由哪些阶段组成？

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）

32、如何理解压力、负载、性能测试测试？

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。

33、如何编写提交给用户的测试报告？

----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。

34、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.

35、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。做测试，有部分是和人的性格有关，有部分需要后天的努力。但除了性格有关的我没有把握，其他点我都很有信心做好它。

36、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。

37、写出 bug 报告当中一些必备的内容。

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

38、开发人员老是犯一些低级错误怎么解决？

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。

39、简述一下 c/s 模式或者 b/s 模式？

C/S 模式：客户端/服务器模式。工作原理：Client 向 Server 提交一个请求；Server 则使用一些方法处理这个请求，并将效果返回给 Client。

B/S 结构，即 Browser/Server(浏览器/服务器)结构，主要是利用了不断成熟的 WWW 浏览器技术，结合浏览器的多种 Script 语言(VBScript、JavaScript...)和 ActiveX 技术，用通用浏览器就实现了原来需要复杂专用软件才能实现的强大功能，并节约了开发成本，是一种全新的软件系统构造技术。

Part2

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、 计划负载测试
- 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。

part3

1. 软件的生命周期（prdctrm）

计划阶段（planning）-> 需求分析（requirement）-> 设计阶段（design）-> 编码（coding）->测试（testing）->运行与维护（running maintrnacne）

2、问：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？

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、原型图、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，

1) 可以根据同行或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；

2) 根据用户的一般使用习惯，来确认是否是缺陷；

3) 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，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，并有上级做出决定。

3、给你一个网站，你如何测试？

首先，查找需求说明、网站设计等相关文档，分析测试需求。

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

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两个方面考虑：

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

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）。

定期评审，对测试进行评估和总结，调整测试的内容。

4、问：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？

300 个用户在一个客户端上，会占用客户机更多的资源，而影响测试的结果。线程之间可能发生干扰，而产生一些异常。

300 个用户在一个客户端上，需要更大的带宽。

IP 地址的问题，可能需要使用 IP Spoof 来绕过服务器对于单一 IP 地址最大连接数的限制。

所有用户在一个客户端上，不必考虑分布式管理的问题；

而用户分布在不同的客户端上，需要考虑使用控制器来整体调配不同客户机上的用户。同时，还需要给予相应的权限配置和***设置。

5、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，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：问题的定义及规划、需求分析/评审、软件设计、软件编码、测试阶段、运行维护 六个时期，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）：

1) 瀑布模型

2) 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。

3) 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

6、什么是软件测试？软件测试的目的与原则

定义：

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。

目的：

测试是程序的执行过程，目的在于发现错误

软件测试为了发现程序中存在的代码或业务逻辑错误

软件测试为了检验产品是否符合用户的需求

软件测试为了提高用户体验

软件测试的原则：

测试应尽早启动、介入（需求分析阶段），所有的测试应追溯到用户需求，测试证明软件存在缺陷，不可能执行穷尽测试，完全测试是不可能的，测试需要终止。

二八原则，测试发现的错误中 **80%** 很可能的起源于 **20%** 的模块中。（缺陷存在群集现象）

对错误结果要进行一个确认的过程（测试的详细数据，截图，前置条件等），制定严格的测试计划；妥善保管测试过程中的所有文档；程序员尽量避免自己的检查程序；设计测试用例是应该考虑到合法的输入和不合法的输入

7、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義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）。

8、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

9、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：

- 1) 明确区分系统中不同用户权限、系统中会不会出现用户冲突、系统会不会因用户的权限的改变造成混乱
- 2) 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）
- 3) 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
- 2) 有关系统的补丁是否打上
- 3) 模拟非授权***
- 4) 看防护系统是否坚固
- 5) 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*****工具***试一下，现在最常用的是 NBSI 系列和 IPhacker IP ）
- 6) 采用各种***检查工具检查系统***情况
- 7) 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：

- 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）、系

4) 统数据可管理性 、

5) 系统数据的独立性 、

6) 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)

10、什么是测试用例 什么是测试脚本 两者的关系是什么?

用例:

未实施测试而编制的一组测试输入、执行条件、各种环境设置以及预期结果以及期望结果的一个特定的集合。

脚本:

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

11、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：由软件的一个或多个用户在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。

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求是什么？

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：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：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：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：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：

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。

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。

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。

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。

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。

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

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。

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。

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。

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。

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。

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。

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。

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。

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、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。

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。

part4

探讨测试用例设计的六大思路

有这样一个面试题：在一个 Web 测试页面上，有一个输入框，一个计数器(count)按钮，用于计算一个文本字符串中字母 a 出现的个数。请设计一系列测试用例用以测试这个 Web 页面。

有经验的测试人员可能会问面试官，字母 a 区分大小写吗？只统计英文字母的 a 吗？最长输入字符是多少，最少输入字符是多少？对输入的字符类型是否有限制，是否会自动清除不符合要求的字符？

所以第一步应该是明确需求，然后我们才开始进行思考如何设计测试用例。

通常说来，我们考虑一个测试对象的时候至少从以下六方面来考虑：

1.功能性

2.兼容性

3.易用性

4.可靠性

5.性能

6.安全

1.从功能方面考虑:

输入: "" (思路: 什么都不输入)

输入: "null" (思路: 特殊值)

输入: "Aa" (思路: 输入字符既含大写字符也有小写)

输入: "abc" (思路: 以 a 开头)

输入: "cac" (思路: a 在中间)

输入: "aba" (思路: 以 a 开头, 以 a 结尾)

输入: " ba" (思路: 以空格开头含 a)

输入: "中 ba" (思路: 以中文或者其他字符开头含 a)

输入: "AAaa" (思路: 输入字符仅仅只有大写 A 和小写 a)

输入: "全角和半角 a" (思路: 考虑半角和全角符号)

2.从兼容性方面考虑:

1.各个浏览器 显示是否正确, 点击按钮是否有效;

2.浏览器各个版本显示是否正确, 点击按钮是否有效;

3.是否支持手机端和平板端。

3.从易用性方面考虑:

1.web 界面外观, 风格是否合适;

2.文本输入框长度是否合适, 是否应该默认提示如何输入;

3.输入错误时提示是否友好;

4.考虑该应用是否支持其他语言。

4.从可靠性和性能方面考虑:

1.输入 HTML 和 JavaScript 相关标签字符, 计算是否正确, 是否会破坏页面;

2.这个应用能否在同一台服务器上运行多个实例，多个用户同时使用是否会有问题；

3.在大并发下使用，计算速度是否满足要求。

5.从安全性方面考虑：

1.输入的数据是否会被保存，输入字符串可能包含敏感信息；

2.尝试复制/粘贴字符串；

3.尝试快速点击多次计算按钮；

4.考虑是否有安全漏洞，点击计算按钮，请求是否会被截取，导致返回失败。

part5

金融软件测试面试题有哪些

网上银行转账是怎么测的，设计一下测试用例。

回答思路：

宏观上可以从质量模型（万能公式）来考虑，重点需要测试转账的功能、性能与安全性。设计测试用例可以使用场景法为主，先列出转账的基本流和备选流。然后设计场景，最后根据场景设计数据。实际面试中需要举出具体的例子。

先检查界面。

再测试功能：

验证同行转账，跨行转账。

验证转账限额。

验证非法账户（挂失，冻结，锁定的账户）的转账。

再测试性能方面的。

测试工作的流程？缺陷状态有什么？设计测试用例有几种方法？

[测试工程师](#)的实际工作流程（以 P2P 中型版本为例，一个月一个版本）：

产品经理或者 SR 把需求书发下来给开发和测试

测试先看一遍，进行需求分析。测试组长编写测试计划，并且分配测试任务给测试人员（2 天时间）（此时开发也在进行需求分析）

过了 2 天，产品经理再把测试和开发召集在一起，进行需求讲解（或者说需求评审），有问题可以直接问，如果发现需求有问题，也可以提出来，SR 回去会修改。（需求讲解时间 0.5 天）

讲完需求后，测试同事要进行测试场景的梳理和案例的编写了（xmind 和 Excel 就要用上了），一共 5 个工作日。（此时开发在编写代码）

之后就要进行案例评审了，评审时候有 SR、测试同事、开发同事，评审时候一般 SR、测试组长、对应模块的开发同事会提出一点意见，评审完之后，回去修改、补充一下案例。（案例评审 0.5 天）

修改完以后，有两种处理情况：

对大项目有时候要进行案例的第二次评审。

对小项目，在时间紧的时候，一般不会二审，但是要以邮件的形式把修改或者新增后的案例发出来，给领导看，并抄送给其他同事。（案例评审 0.5 天，修改案例 0.5 天，案例二审 0.5 天）

案例评审完就要开始测试了，一般测试环境开发搭建好（要说自己也会搭建，搭建流程背老师总结的）：

中型版本的测试一般分 2 轮：第一轮：5 天；第二轮：3 天；回归测试 2 天；（共 10 个工作日）。

回归测试完后，达到了上线标准，就会如期上线，一般当天晚上 12 点上线

缺陷状态：缺陷管理的流程图

在项目中找到的经典 BUG 是什么？

兼容性问题，在 ie 浏览器，提交订单按钮可以点击，到了谷歌，火狐就不能了。

查询订单页面，根据条件筛选的结果不是想要的结果，还有某些字段的值没有显示出来，或者显示错误。（因为开发从库表取值有误）

付款成功后，订单状态一直不翻转为交易成功。（因为代码没有正确获取库表中付款成功记录的状态码）

修改支付密码，新密码和原密码一致，也通过了，系统没有做新旧密码的校验。

付款时候的手机验证码，可以一直使用，没有成功做有效期控制。

手机 app 断开网络后，再去点击，没有友好的错误页面提示网络已断开，只有 undefined 返回

定期存款到期自动转存该怎么测？

回答思路：到期肯定会有边界，所以设计里面可以考虑边界值法。自动转存（首先要搞清楚什么是自动转存。）

存钱该怎么测，用什么测试方法？

准备思路：存钱要分类：活期、零存整取等（具体规则[百度](#)下），然后根据每类的业务规则选择合适的用例设计方法。譬如一次最少存入多少？最多一次能存入多少等。

你发现 Bug 后，应该怎么办？

首先咨询一下开发是不是 bug，让他初步判断一下。

如果不是 bug，开发给到理由也比较充分，确实自己也搞错了，也就算了。

如果开发也认为是 bug，那就直接提了。

如果我怀疑开发的解答，我觉得是 bug，开发坚持不是 bug，我就要咨询我们组长或者开发组长，让他们判断一下。

假如发现了一个 BUG，跟开发本身没什么关系，涉及到理念，需求问题，如何解决？

把问题暴露给测试组长和开发组长，咨询他们意见，组长们再知会开发分组经理和[项目经理](#)，然后大家和产品经理一起探讨解决，需要改需求的地方就要改了。

测试非常紧急过程中，遇到阻塞性问题，对应的开发没有时间解决，你如何推动问题解决？

首先判断问题的严重性，向对应的开发了解问题的原因。

然后再汇报给自己的测试组长和开发组长，让组长知情，咨询他们的意见，再把问题汇报给开发分组经理，让他们统一协调处理。安排经验丰富的其他高级开发人员来协助此开发解决问题，然后通过加班来完成问题解决和测试。

功能测试的 BUG 级别你们怎么划分？

bug 严重程度：一般提 L4 和 L3，L2 很少提，除非影响流程。L1 这个是非常致命的 bug，基本上不会提。

执行别人的用例，如果发现用例有错怎么处理？

首先咨询一下案例作者或者询问测试组长，确认一下，如果确实有误就要修正用例。

你们做过冒烟测试吗？冒烟测试是什么（理论）？

冒烟测试也叫预测试，就是正式测试之前的一种测试，为了确保主流程能走通。

可以回答没有冒烟测试，就说测试之前一般会要求开发自测，开发自测后（自测大概就是一天左右的时间），确保没有大的问题，再通知测试开始测试。

你们项目做了多久，共写了多少用例？项目多少人？

项目做了多久：（两种回答，建议选择第一种）

我进去的时候项目已经上线了，一直存在，然后就是版本的微小更新，小修改的话，大概半个月一个版本，中修改的话，大概一个月一个版本。每次版本更新，针对新的功能点或者修改点大概写了 60 条案例左右（一个月一个版本的例子）。

我进去的时候，一开始就参与这个项目（也就是需求分析开始），项目从零到有进行了半年左右，六个月内大概整个项目组写了 900 条案例左右。自己写了 200 条左右(共 5 个测试，包括组长)。

PS:如果大家说自己是从零到有参与的项目，那么 6 个月时间是从需求分析开始。需求书编写完成前，产品经理他们是要做很多前期准备工作，可能要花费 3 个月左右的时间。

那么测试 6 个月的实际工作时间内：

前期 2 个月：刚开始需求书的漏洞比较多，需求评审比较多，基本上每个星期一次评审。开发和测试都会参与，此时开发在进行代码设计，测试就在分析需求，看参考文档，用 xmind 梳理测试场景，提取测试点，开发经常和产品经理讨论需求，测试经常问开发和产品经理有关需求的疑问。大家一直碰撞，一步一步得出比较完美的逻辑。

中间 2 个月：开发设计完后，进行编码，我们测试就根据之前梳理的测试场景来编写案例，进一步优化。这个期间，需求书基本稳定，不会再改了。要改也就是把细化需求，把笼统的地方，描述的更详细，更让人易懂，功能点的大方向不会

改。开发和测试在此期间有疑问，都会邮件或者电话联系产品经理。测试也会经常去问开发有关功能点的逻辑问题。

后面 2 个月： 执行案例工作开始进行，一般分为两轮 st 测试，第一轮 1 个月，第二轮半个月，回归测试半个月。Uat 测试组在 st 测试第二轮时候，并行开始。Uat 测试组有专门人负责，一般需要 st 测试组派一个人左右去支持，uat 测试也有第一轮（半个月），第二轮（半个月）。

项目多少人：一个公司往往有很多项目，自己只是其中一个项目组的，我的 P2P 项目组大概 20 人，开发 15 个，测试 5 个。（大家把自己当成外包人员，在甲方工作，也叫驻场工作）

假如要你测试 6 个月期限的 p2p 借款产品，你应该怎么设计案例，说出测试点

（回答思路：1 站在用户的角度测试，用户怎么用，你就怎么测试。2 一个人扮演多种角色测试。 3 多想出一些异常场景。）

借款产品投标结束日 T+7 时，满标和不满标的情况。

借款产品投标结束日 T+7 前，产品提前满标情况

产品成立后，每个月还款日前，检查系统有没有发出邮件，短信，站内信通知借款人充值到平台账户。

在每月还款日，借款人充值用来还款时，充值资金足够、不足够、不充值情况，查看系统如何处理。充值资金不足或者没有充值时，系统应该有罚息。

借款人提前还清余款场景，有些产品不支持提前还款，有些产品要满一定期限才可以提前还款（提前还款有一定手续费）。这些都是要关注的测试点。（自己要扮演借款用户去操作提前还清余款，然后扮演后台管理员去审核，然后又扮演投资人用户去检查虚拟账户的资金到账情况）

最后一期借款人还清资金时，去后台页面查看借款产品状态，应该已正常结束。再去前台页面搜索，应该无该借款产品了。（或者补充说：去数据库里查看此借款产品的状态）

你们这个 P2P 上线了吗？能查吗？项目花了多久时间，预计多久完成？

回答：两种方案：

还没上线，查不了，这个是新项目，计划半年时间完成，但是因为中途有出现一些问题没有解决完毕，所以现在还没有在预计时间内完成。

大家写的项目名在网上确实能查出来，就说上线了，能查到的。（面试官其实不一定会去查）

实名认证你们是怎么测得？调取什么平台的资料？

实名认证接口：

银行卡实名认证（调用银行接口，验证卡号，姓名，身份证号码，手机号码。需要利用到手机接收到的验证码）

身份证实名认证（全国公民身份证号码查询服务中心，或者说公安接口）

注册需要实名认证吗？

注册不需要实名认证：当购物时候需要实名认证。

P2P 你们也测试后台管理吗？个人芝麻信用积分是调取哪里的资料？

测试后台管理：

后台也测，但是我主要测试前台，我的关注点是前台，后台只是拿来用，能配合前台正常走完流程就行。

后台主要对前台进行管理，主要有贷款管理，资金管理。

贷款管理：可以查看投资人的投资情况，也可以查看借款人的借款产品，对借款产品进行管理。比如审批，每期的还款提醒，预警等。

资金管理：管理查看用户的充值，审批用户的提现过程。

芝麻信用积分：调用的是支付宝的接口，芝麻信用：调用的是支付宝那边的接口（支付宝提供这样的芝麻信用服务，每查一次收取大概 0.1 元）

如果要测试后台删除用户，就是用户名后面一个删除按钮的情况，能写出哪些测试用例

删除一个用户的场景：点击删除按钮，页面自动刷新，此用户在该页面已查询不到。再去打开另外一个浏览器，在前台登录已删除的用户，页面提示该用户不存在。

同时删除多个用户的场景：利用复选框，测试多选，反选，全选删除用户的情况。删除后，被删用户在该页面已查询不到，同样要去前台登录已删除的用户，页面应该提示该用户不存在。

如果[京东](#)有一个购物网页给你，你要怎么进行测试？ 测试哪些主要功能？

首先进行需求分析，用 xmind 梳理测试点，再编写案例，之后就行案例评审，寻求他人意见。之后再完善案例，发出来给其他人检查。

测试点，首先是 UI 方面：美观度，和易操作型，易理解性型方面进行测试。

然后再考虑他的功能点，注册登录，添加购物车，下单，付款，发货，确认收货，评价。还有支付时候的绑定银行卡，实名认证

性能方面：打开网页，确认订单、付款的响应时间等等。

兼容性：支持各种主流浏览器，ie，360，火狐，谷歌等。

针对添加购物车这个测试点说一下你要怎么测试“添加购物车”

（增删改查的角度）

能否加入购物车，同一件商品能否再次添加到购物车。

购物车商品件数的上限限制（淘宝限制 100 件）

购物车是否可以正常移除商品，移除商品后，能否再添加回来。

添加的每种商品是否可以正常增减数量，数量大于 0

退出购物车，再去查询购物车，商品正常。

购物车的商品可以全选，取消全选，可以复选，选中的商品和数量可以正常下单。

商品添加到购物车以后，已下架。购物车会提示此宝贝已失效。

商品添加到购物车以后，降价了，购物车会有降价提示。

商品添加到购物车以后，库存不足了。

P2P 功能测试你们一般做几轮？

答：

中型版本（大修改，一个月上线一次）：测试一般分 2 轮：第一轮：5 天；第二轮：3 天；回归测试 2 天；（共 10 个工作日）。（一个月工作日 22 天，需求分析评审，编写测试用例等等一般占用整个版本时间的一半，或者少个几天）

小型版本（小修改，两个星期一次）：一轮测试 3 天，回归测试 2 天。

你们每次开会讨论的时候十几个开发都去开会了吗？

案例评审会：一般开发和测试、产品经理都会到场。（开发分组经理可能也会去）
需求评审会：[项目经理](#)、开发分组经理、产品经理、测试、开发一般都会到。

如果是我们测试小组开会，一般都要到，各位测试同事报告自己的心得体会，汇报自己的进度和问题。

数据库查找两个表

回答思路：

多表查询，后面具体会学到：`select 列 1, 列 2 from 表 1,表 2 where 表 1.列=表 2.列` 这样的格式要能说出来。

熟悉数据库吗？平时数据库用的多吗？

熟悉数据库吗：比较熟，比如 DML 语句有增删改查：（有序思维说出来）

1 `insert into 表名 values (值 1, 值 2, 值 3, ...)`

2 `delete from 表名 where 条件`

3 `update 表名 set 列名 = 新值`

4 `select * from 表名`

查询语句最长的是 `select * from 表名 where 条件 group by 分组列名 having 分组后的条件 order by 列名`。

平时数据库用的多吗（大概测试过程的 1/4 时间在查数据库）：还行，一般出现问题，遇到 bug，就要去查询数据库，初步定为问题。开发会给到我们一个库表设计的 excel（数据字典），里面有描述表名和表中的字段，我把交易过程的一些唯一标识，把他作为 where 条件去查询数据。初步分析后，再把问题暴露给开发。（比如淘宝支付时，输入支付密码后，已经返回了支付成功的提示信息，然后界面上的订单查询还是待付款，这个时候就要去查询订单表的数据，找到自己刚才做的交易的那一笔订单，去分析一下错误，再暴露给开发）

linux 查看文件用什么命令，查看进程用什么命令

回答：

查看文件内容的命令有 `more less head tail cat tac`

查看进程：`ps -ef | grep` 进程号

查看日志文件常用：`less`、`view`

查看日志常用什么命令，主要查看什么内容

查看日志常用 `less` 命令或者 `view` 命令。

主要查看程序运行的记录，比如支付失败，后台就有报错信息打印到`.log`日志文件中，就可以通过分析日志信息来初步定为问题。（补充：同时也去查询数据库，分析订单数据，查看支付状态等等）

PS:日志就是`.log`的文本文件，和`.txt`一样属于文本文件。`vi`或者`vim`编辑器属于记事本软件，一般不会用来查看日志。

如何查找 `a.log` 日志文件的 `error` 字符串

第一种方式：（建议说第一种方式）

`cat a.log | grep error;`

第二种方式：

1 `less a.log;`

2 `/error;`

你所熟悉的 linux 命令

linux: `cat,more,less,head -n, tail -n, find ,| grep,ps -ef,tar,gzip,mv,cp,touch,mkdir,vi,top`

也可以结合搭建环境的过程说用到的命令。

你们测试用的测试环境是谁给的？linux 怎么搭建测试环境？

一般开发搭建，但是我也会，我之前自己搭建过一个小项目（松勤学员参考考试系统的搭建流程）

流程大概是：

首次搭建：

通过 winscp 上传 tomcat, MySQL 安装包, JDK (Java 开发环境工具包) 到 linux 下

利用 tar -zxvf 解压缩包命令对 jdk,tomcat, mysql 进行解包、安装, 再配置 jdk 环境变量。

把 war 包 (web 程序) 放到 tomcate 指定目录 webapps 下, 再启动服务器即可。
(输入 startup.sh 的路径, 直接回车即可运行)

非首次搭建:

把 war 包 (web 程序) 放到 tomcate 指定目录 webapps 下 (已经存在 web 服务器和数据库服务器的前提下), 启动服务器即可。(输入 startup.sh 的路径, 直接回车即可运行)

抓包工具使用:

就是打开 fiddler 工具后, 再去浏览器打开网页, fiddler 会自动抓包, 抓取请求响应数据。他会自动设置为本地代理, 还可以设置抓取 https 协议的包。

如果要抓取手机访问互联网数据包, 就要在手机上的网络设置里, 设置代理服务器。就是把 fiddler 作为代理服务器 (fiddler 自身要设置为支持远程连接), 手机连接 fiddler 工具, 所以手机代理服务器设置页面要输入打开 fiddler 工具的电脑的 ip 地址和 fiddler 的端口号 8888, 好让手机能连接 fiddler, 通过 fiddler 来访问互联网。

PS: 浏览器都自带抓包工具, F12 快捷键可以调用此工具, 开发经常利用此工具来分析页面数据, 通过分析页面数据来定位程序问题。

金融行业知识你了解多少

把以下老师整理的理解记忆一下:

如果领导分配你的任务超出负荷, 领导高估了你的能力, 怎么办

回答思路:

首先表达态度, 态度上愿意通过加班来完成, 还可以请求测试同事支援, 让组长协调。

高估了能力, 能力可以在工作中通过自己的努力来达到领导的要求

总而言之基本的思路是态度要端正。

不能直接拒绝任务。但也同时表达万一做不好还请领导包容。

假设你是组长，团队中有一个员工无法按时完成交付的任务，你如何处理；

回答思路：

首先先检讨自己是否任务安排超过了这个员工的能力。

如果没有超过，首先表示关心身体和状态，了解未及时完成任务的原因，如果原因是客观原因则一起加班跟员工来完成任务。

如果是态度原因，则指出利害关系，责令其通过加班来完成。

如果因为你的错误导致工作发生问题，你怎么办？

回答思路：

首先要表达在过去的工作中从未发生过类似事情，因为自己工作态度还是很端正的。

万一因为自己的错误导致工作发生问题，首先应该把问题上报给领导，争取把问题的影响降到最低程度。

给你一个模块测试，只有一个星期的时间你如何有效率地完成？

答：在有限的时间里，明确需求的情况下，制定工作计划，把每天任务细分，先保证重要功能，跟进修复情况，及时验证 bug。每天发工作日报，汇报进度，如果遇到风险，及时汇报领导。

如果给你一个没有需求的 app 测试项目，你应该怎么测

老师建议：根据 APP 的 11 大测试点：

权限测试

安装、运行、卸载测试

UI 测试

功能测试

性能测试

中断测试

兼容测试

安全测试

回归测试

升级更新测试

用户体验测试

补充：根据自己的经验，制定测试计划，每天汇报自己的进度，发出测试日报。

测试过程有问题，及时上报，及时跟进 bug，多和开发交流沟通，明确需求。

如果你和开发的意见产生分歧，你怎么处理？

回答思路：

大的原则是对事不对人。

另外我会首先尝试站在开发的角度接受对方的意见和建议，同时控制好自己的情绪，在对方情绪可控的情况下表达自己的意见。

如果你组长的用例写错了，但他认为是对的，你怎么处理？

回答：

通常情况下，领导看问题的角度会比我们更全面，所以我首先得确保领导的用例是否真的有考虑不到的地方。

我不会坚持自己的是对的，但会在合理的情况下表达自己的观点。

你同时负责功能和性能，你怎么做

先测成功能，保证功能的完成，再做性能，在提交 bug 后，开发还没改好时，可以准备性能测试，在工作时间很紧的情况下会主动加班

我们公司自动化测试用的语言是 Java，Java 你不会，该怎么办？

回答思路

问到不会的标准思路：要么说会一点相关的内容，要么表达自己有不错的学习能力和很好的学习意愿和态度。

我们学了 Java 了就说会，知道面向对象的封装，继承，多态，知道多线程的两种创建方式（自定义子类继承 Thread 类，或者自定义子类实现 Runnable 接口），还知道异常 Throwable，Exception 的格式，try catch finally。知道 List，Set，Map 集合。我可以很快的学会用 Java 做自动化。

以前的项目是怎么管理的？

回答思路：

我们以前的项目是用禅道来做测试的需求管理、用例管理、缺陷管理的。另外版本管理工具使用的是 SVN。

以前的项目每天需要执行多少用例

回答思路：

正常情况一般每天执行 20 个左右的用例，刚开始测试的时候，bug 比较多，需要很多时间和开发交流沟通

案例执行会比较慢。越到后面就越快了。

你们做回归测试的时候是否全部都做呢？

看时间，如果时间比较充足，会全部回归，回归时候因为自己操作比较熟练，然后系统基本上也没有 bug。所以执行案例的速度会比较快。

如果时间比较紧，就会挑选重要模块来回归测试了。

PS:自己组织好语言。

你们怎么确保用例覆盖率？确保不重复

利用判定表法的思想，先穷举，再挑代表。

然后，案例评审时候产品经理、开发组长、测试组长，还有对应模块的开发负责人也会把关，可以咨询他们意见，确保案例即覆盖完全，又没有多余的重复案例。

你们案例是怎么评审的

评审时候有产品经理（SR）、测试同事、开发同事，评审时候一般产品经理（SR）、测试组长、对应模块的开发同事会提出一点意见，评审完之后，回去修改、补充一下案例。

修改完以后，有两种处理情况：

对大项目有时候要进行案例的第二次评审。

对小项目，在时间紧的时候，一般不会二审，但是要以邮件的形式把修改或者新增后的案例发出来，给领导看，并抄送给其他同事。（案例评审 0.5 天，修改案例 0.5 天，案例二审 0.5 天）。

视图是什么？

视图记录了一条 SQL 语句，当查询时才有数据返回。表就是一张具体的表。视图只能查询数据，表可以增删改查。

工作非常努力了，还是没有完成上级交代的任务，怎么办？

回答思路：

其实领导最喜欢的员工是：能力强、态度好的。领导招聘我们的目的是帮助他解决问题。

你工作非常努力，还是没有完成上级的任务，要分析原因，如果是能力不够的原因，则表示愿意且一直在提高能力，希望领导能谅解。

如果是因为可能的领导安排的任务过多，则要委婉地表示自己的能力有限，不希望自己的能力影响项目的进度，另外也请领导多给点提高效率的建议。

你的职业规划是什么？

首先快速熟悉业务，熟悉环境，再主动研究，转组长，经理（突出自己的努力和稳定）

（切忌在功能测试的面试说自己要往自动化，性能发展。

因为他怕你不稳定，以后会嫌弃他公司的功能测试。

除非该公司以后会考虑使用自动化或者性能测试技术）

平时周末不上班都做些什么呢？

有空就会学习巩固技术知识，比如自动化，性能，还自学 python 和 selenium

从上家公司学到了些什么？

从大家一起努力认真而有序的项目过程中，虽然辛苦，但是收获良多。我获得了测试的经验，业务的熟悉，技能的提升，以及团队配合协作的精神、坚持不懈的精神。

为什么从上家工资离职

面试官可能会说：你就实在和我说吧，不要说什么套话。

（还是选择说套话吧）首先感谢上家公司提供的提升自我工作经验的机会，之所以想离职是因为想积累不一样的经验，更进一步的学习，来提升自己。我觉得贵公司非常符合自己的要求。

你住哪里？

因为很多人离职时候，往往会以住的地方太远为借口来申请离职，所以面试官可能会问你住哪里，防止你以后入职不稳定。

回答：

住的比较远的同学就说住哪里哪里，上班比较近。（住的地方建议说成和上班的地方在 1 个小时路程以内）

离职时候工资多少？

说比现在期望薪资少 500 元。

人力面试：

1、为什么转做测试

回答思路：

大学就通过互联网了解软件测试，了解 IT，自己也比较喜欢，然后也选修了 C 语言或者 Java 语言来学。

在大四之前的暑假，在松勤培训过软件测试。

2、加班出差能接受吗，加班能接受吗？

回答思路：

通常如果这个问题被问题，是绝对不能直接说不接受的，能接受出差，还没有男/女朋友。

搞 IT 一般都要加班，我以前也是这么加的，没问题。

站在自己的角度说：还年轻，希望能在短时间内提高自己的能力和积累更丰富的经验，加班是没有问题的。

3、说说你自己与众不同的地方和性格上的缺陷以及你准备如何改善

回答思路：

其实这个问题就是回答优缺点。

性格本身是一种习惯，说以你应该表示通过优化自己的行为习惯来改变自己的缺点。

向身边的榜样学习，就是学最好的别人，做最完美的自己。

4、在学校时参加过社团吗、当过最高的职位，会协调吗？

回答：

如果有就更好，这个能够体现自己的协调能力、组织能力、沟通能力。这些对于工作很重要。要讲一两件具体的事情，把能力通过事情体现出来。

5、领导和追随者你认为自己适合哪个？

回答：

领导是带领和指导，一般通用的回答要是领导，因为自己可以以身作则，技术上也能对下属有一定的指导能力。

6、以往工作经验；

回答：

在忙碌的工作当中，既充实，又有成就感。通过不断的测试，我的沟通能力、协调能力得到了提高，同时还收获了行业知识经验等，深刻感受到了团队精神的重要性。

8、为什么要从事软件测试；

回答：

自己非常喜欢互联网，喜欢 it，我觉得这一行非常有前景，马云说现在已经世界已经进入第三次工业革命了，就是信息技术革命。计算机发展速度很快，互联网公司可以利用短短几年时间到达传统行业过去要几十年才能达到的境界。

9、过去工作中最有成就的事情是什么；

回答思路：

基本原则是要谦卑，谈不上最有成就的事情。

如果非得要说有的话从某一件事情上收获颇多，克服了什么样的困难等。

如果对软件测试有兴趣，想了解更多测试知识，解决测试问题,以及入门指导，帮你解决测试中遇到的困惑，我们这里有技术高手。如果你正在找工作或者刚刚学校出来，又或者已经工作但是经常觉得难点很多，觉得自己测试方面学的不够精想要继续学习的，想转行怕学不会的，都可以加入我们 810119819，群内可领取最新软件测试大厂面试资料和 Python 自动化、接口、框架搭建学习资料！

10、试用期、转正期望工资多少；

回答思路：

首先要说其实工资不是最关键的，然后给一个 500 元范围浮动的值

一线城市工资应届生最低 6000，毕业一年 7000，毕业两年 8000，毕业三年 9000 以上。小编给的是最低标准，大家看根据自己学习情况，适当调整，比如学的不错的同学，两年工作经验提 10000 没有问题的。

如果问你上一家公司工资多少，就说出比你现在期望工资少个 500 元的值。