

金九银十面试怒拿 6 个 offer——测试开发面试题整理

1、软件测试的流程是什么？

分析：每当 HR 问一个问题的时候我们都可以用 1~2s 的时间去想 HR 想要从这个问题中获取什么信息，这点搞清楚之后再去回答就很好回答了。如果有工作经验，直接按照公司流程回答即可，如果是刚转行或者刚实习，那按标准回答即可，文中回答仅供参考。

回答：项目经理或者 PD 把项目需求文档提前下发给相关的研发人员，研发人员抽出一定的时间记录文档内需求不明确或者遗漏的点为后面的评审做准备；在需求评审会议上，各研发人员提出自己的疑问并解决，需求评审最终通过之后会出一份最终的需求规格说明书；（需求评审阶段）

需求规格说明书评审通过后，开发经理开始编写开发计划，测试经理开始编写测试计划，计划评审通过后开发人员开始进行程序的开发，测试人员开始测试用例的编写，等程序的第一个版本出来后，开发人员进行第二个版本的迭代，这时测试人员对程序进行测试并记录追踪管理缺陷，直到程序迭代完毕。（产品研发阶段）

程序迭代完毕并修复大部分缺陷后，测试人员开始进行工作的总结，并最终输出一份测试报告书，记录此次的测试工作中程序存在的相关问题。（产品发布阶段）

2、测试用例主要有哪些元素？

分析：每个公司因为使用的模板不一样，所以测试用例的内容也是不尽相同的，所以回答时只需要回答出基本的元素即可。

回答：测试用例主要元素有：ID、标题、模块、预置条件、操作步骤、预期结果、实际结果、是否通过、BugID 等。

3、软件测试有什么策略和阶段？

分析：软件测试的策略就是测试将按照什么样的思路 and 方式进行如采用什么技术，什么步骤等。

回答：软件测试的策略主要有：动态测试和静态测试、白盒测试和黑盒测试。

测试阶段按照研发顺序分别是：单元测试、集成测试、系统测试，有些公司还会有验收测试。（单元测试开发在调试代码时就完成，集成测试也是，但是有时测试人员也需要进行集成测试；测试人员平时主要的工作就是系统测试，验收测试是有客户参与进行的测试）

4、黑盒测试和白盒测试是什么？二者有什么区别？

分析：黑盒测试和白盒测试的概念百度百科上面都有，这里不再做太多介绍。黑盒测试和白盒测试的区别：<https://www.cnblogs.com/i16i1007/p/6624072.html> 记得

回答：黑盒测试主要是在程序界面进行测试，通过设定某种场景检验程序在这种场景下是否给出了正确的反应，验证程序正确实现了需求规格说明书中的需求，而白盒测试主要是针对程序内部结构，对程序代码进行代码走查等，但是白盒测试的成本会比较大，当程序有多个路径时，可能会产生较多的遗漏。

5、软件测试有什么类型？

回答：常见的软件测试类型有：功能测试、性能测试、兼容性测试、可靠性测试、安全性测试、压力测试、负载测试等。

6、测试用例是什么？有什么作用？

回答：测试用例就是设计一个特定场景，让软件在这种场景下运行，检验程序是否给出正确的反应，以此验证软件是否正确实现了客户需求。

作用：1、避免盲目测试并提高测试效率；在软件版本更新之后只需修正少部分用例即可开展测试工作，降低工作强度，缩短测试周期；

2、可以分清哪些是测试重点，测试用例是测试工作的见证，能知道测试了哪些功能，没测哪些模块；

3、测试用例是量化测试工作的方法之一。

7、你平时是怎么设计测试用例的？

分析：这个问题的点主要考察是否掌握测试用例设计方法，在回答之后，HR可能会继续追问某种设计方法的概念或者实例，这时举例说明即可；如：等价类划分法就是把程序的输入域划分成等价类，从每个部分中选取少数代表性数据当做测试数据。

回答：设计测试用例一般都会使用到等价类、边界值、场景/流程法、因果图还有错误推测法。

8、软件缺陷的定义是什么？

分析：什么样的问题才是一个缺陷，需要从客户需求出发。

回答：1、软件未实现需求规格说明书中的要求；

2、出现需求规格说明书中指明不应该出现的错误；

3、软件未实现需求文档中虽未明确提及但应该实现的功能；（如：账密加密）

4、软件出现难以理解、不易使用或者运行速度慢等问题都可以认为是软件缺陷。

9、缺陷中应该包含什么元素？严重等级一般有哪些？

分析：这个问题和上面测试用例一样，每个公司的要求可能都会不一样；

回答：主要元素有：标题、BugID、复现步骤、实际结果、预期结果、截图、日志等；软件缺陷等级一般有四种，致命（程序奔溃）、严重（金额计算错误、数据出错）、一般（不影响使用但是会造成一定的麻烦）、优化（字体字号不统一）

10、给你一个杯子，你会怎么测试？

分析：给你一个杯子，给你一个电梯，这种问题在前期的面试中是经常遇到也是非常经典的一道面试题，这里给出一个链接，回答时从外观、功能、性能等各个角度说起，再结合自己的一些话就可以了。

11、测试报告里面包含什么内容？

分析：测试报告，是测试工作结束后测试部门输出的一份测试结果，但每个公司的测试报告内容都会有些差别。有些公司的测试报告是有测试部门的负责人一人编写，或者是由每个测试工程师输出自己对应模块的测试报告再由测试组长整合成一份完整的测试报告。

回答：测试报告内容一般有：编写目的、系统简介、测试环境、测试方法和工具、测试执行结果与记录、缺陷汇总、遗留缺陷跟踪、测试用例执行情况、测试结论与建议等。

12、如果在测试过程中发现了 BUG，可是开发不承认这是 Bug，你会怎么办？

分析：HR 问这个问题主要还是想知道你平时是如何处理与同事之间的关系。开发和测试是两个即对立又统一的两个岗位，所以开发和测试之间关系的处理也是 HR 面试过程中需要考虑的一个点。当然，HR 问这个问题也表明该公司有氛围不是很好的风险。

回答：首先还是应该回归到客户需求上面，确认这个问题到底属不属于一个缺陷，如果确实是则要和开发同事解释清楚；如果开发还是坚持自己想法的话，则询问同事或者测试组长的意见，讨论这个问题到底属不属于缺陷问题，如果大家都觉得是则需要和开发解释清楚。

13、你们公司的需求评审是怎么进行的？

分析：需求评审，就是对客户需求，软件各个模块之间模糊的点进行审查，排除不理解或者没有考虑到的点。

回答：需求评审，在一些分工比较明确的公司，都是由 PD（产品设计师）负责，需求确认好后再下发到开发和测试部门；分工不怎么明确的公司可能就是开发测试产品等大家坐在一起共同探讨；评审形式一般分为线上和线下两种方式，负责人一般会提前把需求文档下发到大家手上供大家整理各自的疑惑点，为后续的评审会议做准备。

14、MySQL 的常用命令有哪些？

分析：数据库知识，是测试工程师必备的一个基本技能，在面试过程中也是经常会遇到的一个考点。对于刚入行的测试，对数据库知识的要求不会太高，只要求能掌握基本的增删改查语句即可。关于数据库的知识，在后续的时间里，也会慢慢的整理出来，供大家学习、参考。

回答：这里只给出几个标准的语法结构：

增：insert into 表名（列名） values （数据）；

如：在 stu 表中插入 id 为 001，姓名为张三的学生，(insert into Stu (stu_id, stu_name) values (001, '张三') ;)

删：delete from 表名 where 指定数据；

如：在 stu 表中删除 id 为 001，姓名为张三的学生:(delete from Stu where stu_id= '001' and stu_name = '张三' ;)

改：update 表名 set 改变项 where 指定数据；

如：在 stu 表中修改 id 为 001 的学生姓名为“张三”:(update Stu set stu_name = '张三' where stu_id= '001' ;)

查：select （查询项） from 表名 where 指定条件；

如：在 stu 表中查询 id 为 001，姓名为“张三”的学生信息:(select * from Stu where stu_name = '张三' and stu_id= '001' ;)

15、Linux 下的一些常用命令是什么？

分析：Linux 系统，也是软件测试工程师必须要掌握的一项基本的技能，由于 Linux 具有运行稳定等很多优点，软件的服务器大多部署在 Linux 系统上，搭建测试环境也是测试工程师需要掌握的。关于 Linux 的知识，在后续的时间里，也会慢慢的整理出来，供大家学习、参考。由于 Linux 下很多命令都是常用的，所以这里不给出答案。

16、你未来的职业规划是什么？

分析：职业规划问题，是所有面试中最常问的问题，问的人可能是 HR、部门主管、经理、甚至是董事长。同一个问题，问的人不同，想要获取的信息也肯定是不一样的。HR 更多的想看你在公司的稳定性；技术主管可能更想知道你是否真的喜欢测试这个岗位，后期是否会主动学习型新的技能等；而经理更多的是看你的职业规划是否符合公司的发展方向；软件测试工程师的发展方向主要有：测试开发、产品经理、测试转开发、测试大牛、讲师等岗

位。

回答：HR：如果是HR问的话，多从稳定性的角度回答，如：家人、朋友都在公司附近，或者喜欢贵公司的文化氛围等；

技术：回答之前可以先简单介绍一下自己为什么选择软件测试这个职业，以及自己对这个职业的看法，最后再回答自己的职业发展方向即可；

经理 or 董事长：这个回答回答起来的话还是比较难把握的，因为在面试时，面试者往往对公司的发展方向不是非常了解，所以在回答时可以再带一句，“具体的发展方向，还需要公司的发展方向去调整”。这样回答就会保险一些。

17、还有什么想要问我的吗？

分析：这个问题在每个面试的尾声都会被问到，直接说没有，会让HR觉得你不关心这个岗位，问的多了又会显得面试之前没有做好充分的准备。所以问题一般控制在两到三个比较好。

回答：1、公司的研发团队目前是什么规模？开发、测试分别有多少人？

2、公司的业务方向是什么？

3、如果我入职之后，我的工作职责是什么？

18、为什么想要离职？

分析：这个问题主要是想要了解你的近况，以及上一家公司是什么原因导致你离职，。在大部分情况下，HR都会理解你，但是在回答问题时千万不能太过于实诚，有些面试者一上来就在抱怨上一家公司如何压榨公司员工等，没有一家公司愿意接受这样的面试者，HR并不能完全感受你所遭遇到的，所以还是请控制好自己的负面情绪。

（二）

1、请你分别介绍一下单元测试、集成测试、系统测试、验收测试、回归测试。

（1）单元测试：完成最小的软件设计单元（模块）的验证工作，目标是确保模块被正确的编码，使用过程设计描述作为指南，对重要的控制路径进行测试以发现模块内的错误，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，及早的发现和解决不易显现的错误。

（2）集成测试：通过测试发现与模块接口有关的问题。目标是把通过了单元测试的模块拿来，构造一个在设计中所描述的程序结构，应当避免一次性的集成（除非软件规模很小），而采用增量集成。自顶向下集成：模块集成的顺序是首先集成主模块，然后按照控制层次结构向下进行集成，隶属于主模块的模块按照深度优先或广度优先的方式集成到整个结构中

去。自底向上集成：从原子模块开始来进行构造和测试，因为模块是自底向上集成的，进行时要要求所有隶属于某个给顶层次的模块总是存在的，也不再需要使用稳定测试桩的必要。

(3) 系统测试：是基于系统整体需求说明书的黑盒类测试，应覆盖系统所有联合的部件。系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不相符合或与之矛盾的地方。系统测试的对象不仅仅包括需要测试的产品系统的软件，还要包含软件所依赖的硬件、外设甚至包括某些数据、某些支持软件及其接口等。因此，必须将系统中的软件与各种依赖的资源结合起来，在系统实际运行环境下来进行测试。

(4) 回归测试：回归测试是指在发生修改之后重新测试先前的测试用例以保证修改的正确性。理论上，软件产生新版本，都需要进行回归测试，验证以前发现和修复的错误是否在新软件版本上再次出现。根据修复好了的缺陷再重新进行测试。回归测试的目的在于验证以前出现过但已经修复好的缺陷不再重新出现。一般指对某已知修正的缺陷再次围绕它原来出现时的步骤重新测试。

(5) 验收测试：验收测试是指系统开发生命周期方法论的一个阶段，这时相关的用户或独立测试人员根据测试计划和结果对系统进行测试和接收。它让系统用户决定是否接收系统。它是一项确定产品是否能够满足合同或用户所规定需求的测试。验收测试包括 Alpha 测试和 Beta 测试。

Alpha 测试：是由用户在开发者的场所来进行的，在一个受控的环境中进行。

Beta 测试：由软件的最终用户在一个或多个用户场所来进行的，开发者通常不在现场，用户记录测试中遇到的问题并报告给开发者，开发者对系统进行最后的修改，并开始准备发布最终的软件。

2、请你回答一下单元测试、集成测试、系统测试、验收测试、回归测试这几步中最重要的是哪一步？

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，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了需求规格的定义，因此我认为系统测试很重要。

3、请回答集成测试和系统测试的区别，以及它们的应用场景主要是什么？

(1) 区别：

1) 计划和用例编制的先后顺序：从 V 模型来讲，在需求阶段就要制定系统测试计划和用例，HLD 的时候做集成测试计划和用例，有些公司的具体实践不一样，但是顺序肯定是先做系统测试计划用例，再做集成。

2) 用例的粒度：系统测试用例相对很接近用户接受测试用例，集成测试用例比系统测试用例

更详细，而且对于接口部分要重点写，毕竟要集成各个模块或者子系统。

3) 执行测试的顺序：先执行集成测试，待集成测试出的问题修复之后，再做系统测试。

(2) 应用场景：

1) 集成测试：完成单元测试后，各模块联调测试；集中在各模块的接口是否一致、各模块间的数据流和控制流是否按照设计实现其功能、以及结果的正确性验证等等；可以是整个产品的集成测试，也可以是大模块的集成测试；集成测试主要是针对程序内部结构进行测试，特别是对程序之间的接口进行测试。集成测试对测试人员的编写脚本能力要求比较高。测试方法一般选用黑盒测试和白盒测试相结合。

2) 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（验证前两个阶段测试的正确性）和功能性（产品提交个用户的功能）测试，又包括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。系统测试测试软件《需求规格说明书》中提到的功能是否有遗漏，是否正确的实现。做系统测试要严格按照《需求规格说明书》，以它为标准。测试方法一般都使用黑盒测试法。

4、请问测试开发需要哪些知识？需要具备什么能力？

软件测试基础理论知识：黑盒测试、白盒测试等；

编程语言基础：C/C++、java、python 等；

自动化测试工具：Selenium、Appium、Robotium 等；

计算机基础知识：数据库、Linux、计算机网络等；测试框架，如 JUnit 等。

需要具备的能力：业务分析能力，分析整体业务流程、分析被测业务数据、分析被测系统架构、分析被测业务模块、分析测试所需资源、分析测试完成目标；

缺陷洞察能力：一般缺陷的发现能力、隐性问题的发现能力、发现连带问题的能力、发现问题隐患的能力、尽早发现问题的能力、发现问题根源的能力；

团队协作能力：合理进行人员分工、协助组员解决问题、配合完成测试任务、配合开发重现缺陷、督促项目整体进度、出现问题勇于承担；专业技术能力，掌握测试基础知识、掌握计算机知识、熟练运用测试工具；

逻辑思考能力：判断逻辑的正确性、对可行性逻辑分析、站在客观角度思考；

问题解决能力：技术上的问题、工作中的问题、沟通问题；沟通表达能力，和技术人员、产品人员、上下级的沟通；

宏观把控能力：有效控制测试时间、有效控制测试成本、有效制定测试计划、有效进行风险

评估、有效控制测试方向.

5、请说一说黑盒与白盒的测试方法。

(1) 黑盒测试：黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能，通过测试来检测每个功能是否都能正常使用，在测试时，把程序看作一个不能打开的黑盒子，在完全不考虑程序 内部结构和内部特性的情况下，测试者在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息，并且保持外 部信息（如数据库或文件）的完整性。

“黑盒”法着眼于程序外部结构、不考虑内部逻辑结构、针对软件界面和软件功能进行测试。“黑盒”法是穷举输入测试，只有把所有可能的输入都作为测试情况使用，才能以这种方法查出 程序中所有的错误。实际上测试情况有无穷多个，因此不仅要测试所有合法的输入，而且还要对 那些不合法但是可能的输入进行测试。

常用的黑盒测试方法有：等价类划分法；边界值分析法；因果图法；场景法；正交实验设计法；判定表驱动分析法；错误推测法；功能图分析法。

(2) 白盒测试：白盒测试也称为结构测试或逻辑驱动测试，是针对被测单元内部是如何进行工作的测试。它根据程序的控制结构设计测试用例，主要用于软件或程序验证。白盒测试法检查程序内部逻辑结构，对所有的逻辑路径进行测试，是一种穷举路径的测试方法，但即使每条路径都测试过了，但仍然有可能存在错误。因为：穷举路径测试无法检查出程序本身是否违反了设计规范，即程序是 否是一个错误的程序；穷举路径测试不可能检查出程序因为遗漏路径而出错；穷举路径测试发现 不了一些与数据相关的错误。

白盒测试需要遵循的原则有：

保证一个模块中的所有独立路径至少被测试一次；

所有逻辑值均需要测试真（true）和假（false）；两种情况；

检查程序的内部数据结构，保 证其结构的有效性；

在上下边界及可操作范围内运行所有循环。

常用白盒测试方法：

1) 静态测试：不用运行程序的测试，包括代码检查、静态结构分析、代码质量度量、文档测试等等，它可以由人工进行，充分发挥人的逻辑思维优势，也可以借助软件工具（Fxcop）自动进 行。

2) 动态测试：需要执行代码，通过运行程序找到问题，包括功能确认与接口测试、覆盖率分析、 性能分析、内存分析等。

白盒测试中的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。

六种覆盖标准发现错误的能力呈由弱到强的变化：

语句覆盖每条语句至少执行一次。

判定覆盖每个判定的每个分支至少执行一次。

条件覆盖每个判定的每个条件应取到各种可能的值。

判定/条件覆盖同时满足判定覆盖条件覆盖。

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。

6、请说一下手动测试与自动化测试的优缺点。

（1）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。

（2）手工测试优点：

- 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。

（3）自动化测试的优点：

- 1) 对程序的回归测试更方便。这可能是自动化测试最主要的任务，特别是在程序修改 比较频繁时，效果是非常明显的。由于回归测试的动作和用例是完全设计好的，测试期望的结果也是完全可以预料的，将回归测试自动运行，可以极大提高测试效率，缩短回归测试时间。
- 2) 可以运行更多更繁琐的测试。自动化的一个明显的好处是可以在较少的时间内运行更多的测试。
- 3) 可以执行一些手工测试困难或不可能进行的测试。比如，对于大量用户的测试，不可能同时让足够多的测试人员同时进行测试，但是却可以通过自动化测试模拟同时有许多用户，从而达到测试的目的。
- 4) 更好地利用资源。将繁琐的任务自动化，可以提高准确性和测试人员的积极性，将测试技术人员解脱出来投入更多精力设计更好的测试用例。有些测试不适合于自动测试，仅适合于手工测试，将可自动测试的测试自动化后，可以让测试人员专注于手工测试部分，提高手

工测试的效率。

5) 测试具有一致性和可重复性。由于测试是自动执行的，每次测试的结果和执行的内容的一致性是可以得到保障的，从而达到测试的可重复的效果。

6) 测试的复用性。由于自动测试通常采用脚本技术，这样就有可能只需要做少量的甚至不做修改，实现在不同的测试过程中使用相同的用例。

7) 增加软件信任度。由于测试是自动执行的，所以不存在执行过程中的疏忽和错误，完全取决于测试的设计质量。一旦软件通过了强有力的自动测试后，软件的信任度自然会增加。

(4) 自动化测试的缺点：

1) 不能取代手工测试

2) 手工测试比自动测试发现的缺陷更多

3) 对测试质量的依赖性极大

4) 测试自动化不能提高有效性

5) 测试自动化可能会制约软件开发。由于自动测试比手动测试更脆弱，所以维护会受到限制，从而制约软件的开发。

6) 工具本身并无想像

7、请问你怎么看待软件测试的潜力和挑战？

软件测试是正在快速发展，充满挑战的领域。尽管现在许多自动化测试软件的出现使得传统手工测试的方式被代替，但自动化测试工具的开发、安全测试、测试建模、精准测试、性能测试、可靠性测试等专项测试中仍然需要大量具有专业技能与专业素养的测试人员，并且随着云计算、物联网、大数据的发展，传统的测试技术可能不再适用，测试人员也因此面临着挑战，需要深入了解新场景并针对不同场景尝试新的测试方法，同时敏捷测试、Devops的出现也显示了软件测试的潜力。

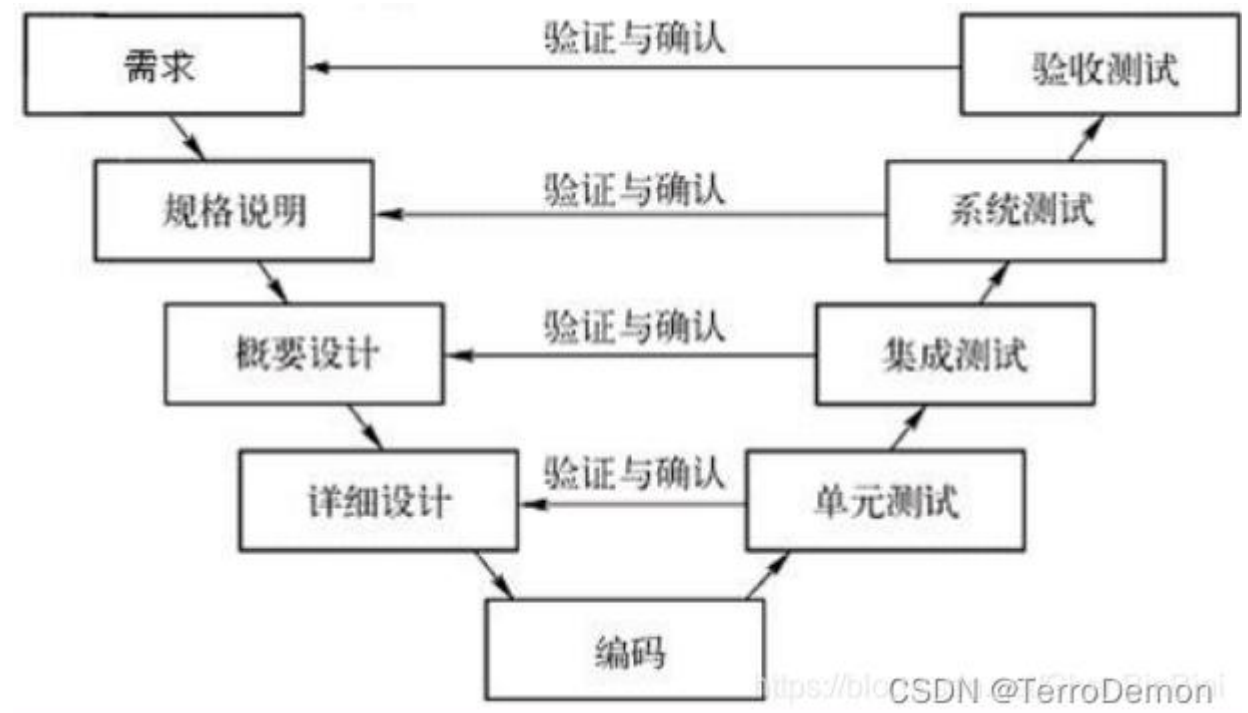
8、你觉得软件测试的核心竞争力是什么？

测试人员的核心竞争力在于提早发现问题，并能够发现别人无法发现的问题。（1）早发现问题：问题发现的越早，解决的成本越低。如果一个需求在还未实现的时候就能发现需求的漏洞，那么这种问题的价值是最高的。

（2）发现别人无法发现的问题：所有人都能发现的问题，你发现了，那就证明你是可以被替代的。别人发现不了，而你可以发现，那么你就是无法被替代。

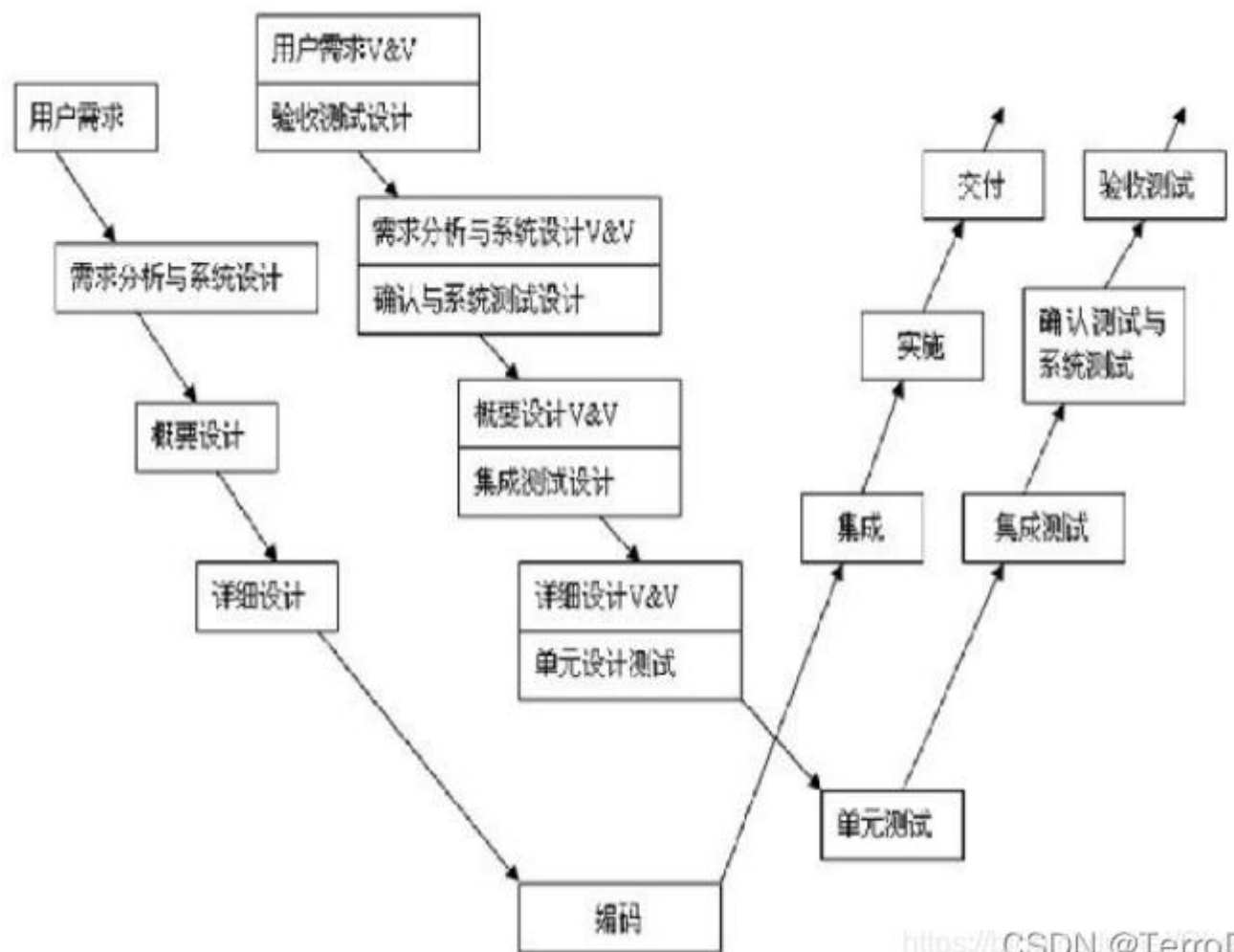
9、你觉得测试和开发需要怎么结合才能使软件的质量得到更好的保障?

测试和开发应该按照 W 模型的方式进行结合，测试和开发同步进行，能够尽早发现软件缺陷，降低软件开发的成本。



在 V 模型中,测试过程被加在开发过程的后半部分,单元测试所检测代码的开发是否符合详细设计的要求。集成测试所检测此前测试过的各组成部分是否能完好地结合到一起。系统测试所检测已集成在一起的产品是否符合系统规格说明书的要求。而验收测试则检测产品是否符合最终用户的需求。V 模型的缺陷在于仅仅把测试过程作为在需求分析、系统设计及编码之后的一个阶段,忽视了测试对需求分析、系统设计的验证,因此需求阶段的缺陷很可能一直到后期的验收测试才被发现,此时进行弥补将耗费大量人力物力资源。相对于 V 模型, W 模型增加了软件各开发阶段中应同步进行的验证和确认活动。W 模型由两个 V 字型模型组成,分别代表测试与开发过程,图中明确表示出了测试与开发的并行关系。

W 模型强调：测试伴随着整个软件开发周期，而且测试的对象不仅仅是程序，需求、设计等同样要测试，也就是说，测试与开发是同步进行的。**W** 模型有利于尽早地全面的发现问题。例如，需求分析完成后，测试人员就应该参与到对需求的验证和确认活动中，以尽早地找出缺陷所在。同时，对需求的测试也有利于及时了解项目难度和测试风险，及早制定应对措施，这将显著减少总体测试时间，加快项目进度。



W 模型中测试的活动与软件开发同步进行，测试的对象不仅仅是程序，还包括需求和设计，因此能够尽早发现软件缺陷，降低软件开发的成本。

10、你觉得单元测试可行吗？

可行，单元测试可以有效地测试某个程序模块的行为，是未来重构代码的信心保证。事前可以保证质量，事后可以快速复现问题，并在修改代码后做回归自测。可行性考虑的是要用一些可行的方法做到关键的代码可测试，如通过边界条件、等价类划分、错误、因果，设计测试用例要覆盖常用的输入组合、边界条件和异常。

11、你觉得自动化测试有什么意义，都需要做些什么？

自动化测试的意义在于:

- (1) 可以对程序的新版本自动执行回归测试
- (2) 可以执行手工测试困难或者不可能实现的测试，如压力测试，并发测试，

(3) 能够更好的利用资源，节省时间和人力执行自动化测试之前首先判断这个项目是不是和推广自动化测试，然后对项目做需求分析，指定测试计划，搭建自动化测试框架，设计测试用例，执行测试，评估。

12、请你回答一下测试的相关流程是什么。

测试最规范的过程如下

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

来自 W 模型

13、请你说一下如何写测试用例。

(1) 测试人员尽早介入，彻底理解清楚需求，这个是写好测试用例的基础

(2) 如果以前有类似的需求，可以参考类似需求的测试用例，然后还需要看类似需求的 bug 情况

(3) 清楚输入、输出的各种可能性，以及各种输入的之间的关联关系，理解清楚需求的执行逻辑，通过等价类、边界值、判定表等方法找出大部分用例

(4) 找到需求相关的一些特性，补充测试用例

(5) 根据自己的经验分析遗漏的测试场景

(6) 多总结类似功能点的测试点，才能够写出质量越来越高的测试用例

(7) 书写格式一定要清晰

14、请问你觉得测试项目具体工作是什么？

搭建测试环境

撰写测试用例

执行测试用例

写测试计划，测试报告

测试，并提交 BUG 表单

跟踪 bug 修改情况

执行自动化测试，编写脚本，执行，分析，报告

进行性能测试，压力测试等其他测试，执行，分析，调优，报告

15、请问如果想进行 bug 的测评，怎么去评测 bug？

Bug 的 priority () 和 severity () 是两个重要属性，通常人员在提交 bug 的时候，只定义 severity，而将 priority 交给 leader 定义，通常 bug 管理中，severity 分为四个等级 blocker、critical、major、minor/trivial，而 priority 分为五个等级 immediate、urgent、high、normal、low。

Severity:

(1) blocker: 即系统无法执行，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，造成系统不稳定。常见的有严重花屏、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、其它导致无法测试的错误，如服务器 500 错误。

(2) critical: 即映像系统功能或操作，主要功能存在严重缺陷，但不会映像到系统稳定性。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误字或拼写错误。

(3) major: 即界面、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，长时间操作无进度提示，系统未优化，兼容性问题。

(4) minor/trivial: 即易用性及建议性问题。

Priority

(1) immediate: 即马上解决，

(2) urgent: 急需解决

(3) high: 高度重视，有时间要马上解决

(4) low: 在系统发布前解决，或确认可以不用解决。

16、请你说一说测试用例的边界。

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

常见的边界值

- (1) 对 16-bit 的整数而言 32767 和 -32768 是边界
- (2) 屏幕上光标在最左上、最右下位置
- (3) 报表的第一行和最后一行
- (4) 数组元素的第一个和最后一个
- (5) 循环的第 0 次、第 1 次和倒数第 2 次、最后一次

17、请你说一下软件质量的六个特征。

按照软件质量国家标准 GB-T8566 - 2001G，软件质量可以用下列特征来评价：

- (1) 功能特征：与一组功能及其指定性质有关的一组属性，这里的功能是满足明确或隐含的需求的那些功能。
- (2) 可靠特征：在规定的一段时间和条件下，与软件维持其性能水平的能力有关的一组属性。
- (3) 易用特征：由一组规定或潜在的用户为使用软件所需作的努力和所作的评价有关的一组属性。
- (4) 效率特征：与在规定条件下软件的性能水平与所使用资源量之间关系有关的一组属性。
- (5) 可维护特征：与进行指定的修改所需的努力有关的一组属性。
- (6) 可移植特征：与软件从一个环境转移到另一个环境的能力有关的一组属性。

18、请你说一下设计测试用例的方法。

黑盒测试：

- (1) 等价类划分 等价类划分是将系统的输入域划分为若干部分，然后从每个部分选取少量代表性数据进行测试。等价类可以划分为有效等价类和无效等价类，设计测试用例的时候要考虑这两种等价类。
- (2) 边界值分析法 边界值分析法是对等价类划分的一种补充，因为大多数错误都在输入输出的边界上。边界值分析就是假定大多数错误出现在输入条件的边界上，如果边界附件取值不会导致程序出错，那么其他取值出错的可能性也就很小。

边界值分析法是通过优先选择不同等价类间的边界值覆盖有效等价类和无效等价类来更有效的进行测试，因此该方法要和等价类划分法结合使用。

(3) 正交试验法 正交是从大量的试验点中挑选出适量的、有代表性的点。正交试验设计是研究多因素多水平的一种设计方法，它是一种基于正交表的高效率、快速、经济的试验设计方法。

(4) 状态迁移法 状态迁移法是对一个状态在给定的条件内能够产生需要的状态变化，有没有出现不可达的状态和非法的状态，状态迁移法是设计足够的用例达到对系统状态的覆盖、状态、条件组合、状态迁移路径的覆盖。

(5) 流程分析法 流程分析法主要针对测试场景类型属于流程测试场景的测试项下的测试子项进行设计，这是从白盒测试中路径覆盖分析法借鉴过来的一种很重要的方法。

(6) 输入域测试法 输入域测试法是针对输入会有各种各样的输入值的一个测试，他主要考虑 极端测试、中间范围测试，特殊值测试 。

(7) 输出域分析法 输出域分析法是对输出域进行等价类和边界值分析，确定是要覆盖的输出域样点，反推得到应该输入的输入值，从而构造出测试用例，他的目的是为了达到输出域的等价类和边界值覆盖。

(8) 判定表分析法 判定表是分析和表达多种输入条件下系统执行不同动作的工具，他可以把复杂的逻辑关系和多种条件组合的情况表达的即具体又明确；

(9) 因果图法 因果图是用于描述系统输入输出之间的因果关系、约束关系。因果图的绘制过程是对被测系统的外部特征的建模过程，根据输入输出间的因果图可以得到判定表，从而规划出测试用例。

(10) 错误猜测法 错误猜测法主要是针对系统对于错误操作时对于操作的处理法的猜测法，从而设计测试用例

(11) 异常分析法 异常分析法是针对系统有可能存在的异常操作，软硬件缺陷引起的故障进行分析，分析发生 错误时系统对于错误的处理能力和恢复能力依此设计测试用例。

白盒测试：

白盒测试也称为结构测试或逻辑驱动测试，是针对被测单元内部是如何进行工作的测试。它根据程序的控制结构设计测试用例，主要用于软件或程序验证。白盒测试法检查程序内部逻辑结构，对所有的逻辑路径进行测试，是一种穷举路径的测试方法，但即使每条路径都测试过了，但仍然有可能存在错误。因为：穷举路径测试无法检查出程序本身是否违反了设计规范，即程序是否是一个错误的程序；穷举路径测试不可能检查出程序因为遗漏路径而出错；穷举路径测试发现 不了一些与数据相关的错误。

白盒测试需要遵循的原则有：

保证一个模块中的所有独立路径至少被测试一次；

所有逻辑值均需要测试真（true）和假（false）；两种情况；

检查程序的内部数据结构，保证其结构的有效性；

在上下边界及可操作范围内运行所有循环。常用白盒测试方法：静态测试：不用运行程序的测试，包括代码检查、静态结构分析、代码质量度量、文档测试 等等，它可以由人工进行，充分发挥人的逻辑思维优势，也可以借助软件工具（Fxcop）自动进行。动态测试：需要执行代码，通过运行程序找到问题，包括功能确认与接口测试、覆盖率分析、性能分析、内存分析等。

白盒测试中的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。

六种覆盖标准发现错误的能力呈由弱到强的变化：

语句覆盖每条语句至少执行一次。

判定覆盖每个判定的每个分支至少执行一次。

条件覆盖每个判定的每个条件应取到各种可能的值。

判定/条件覆盖同时满足判定覆盖条件覆盖。

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。

19、请你说一下 app 性能测试的指标。

（1）内存：内存消耗测试节点的设计目标是让应用不占用过多的系统资源，且及时释放内存，保障整个系统的稳定性。当然关于内存测试，在这里我们需要引入几个概念：空闲状态、中等规格、满规格。

空闲状态指打开应用后，点击 home 键让应用后台运行，此时应用处于的状态叫做空闲；

中等规格和满规格指的是对应用的操作时间的间隔长短不一，中等规格时间较长，满规格时间较短。

内存测试中存在很多测试子项，清单如下：

●空闲状态下的应用内存消耗；

●中等规格状态下的应用内存消耗； ●满规格状态下的应用内存消耗；

●应用内存峰值；

●应用内存泄露；

●应用是否常驻内存；

●压力测试后的内存使用。

(2) CPU: 使用 Android 提供的 `view plaincopy` 在 CODE 上查看代码片派生到我的代码片 `adbshell dumsys CPUinfo |grep packagename >/address/CPU.txt` 来获取；使用 `top` 命令 `view plaincopy` 在 CODE 上查看代码片派生到我的代码片 `adbshell top |grep packagename>/address/CPU.txt` 来获取。

(3) 流量: 网络流量测试是针对大部分应用而言的，可能还有部分应用会关注网速、弱网之类的测试。

流量测试包括以下测试项：

应用首次启动流量提示；

应用后台连续运行 2 小时的流量值；

应用高负荷运行的流量峰值。

(4) 电量：

●测试手机安装目标 APK 前后待机功耗无明显差异；

●常见使用场景中能够正常进入待机，待机电流在正常范围内； ●长时间连续使用应用无异常耗电现象。

(5) 启动速度：

第一类：首次启动 --应用首次启动所花费的时间；

第二类：非首次启动 --应用非首次启动所花费的时间；

第三类：应用界面切换 - 应用界面内切换所花费的时间。

(6) 滑动速度、界面切换速度

(7) 与服务器交互的网络速度

20、请你说一说 app 测试的工具。

(1) 功能测试自动化

1) 轻量接口自动化测试

jmeter,

2) APP UI 层面的自动化 android: UI Automator Viewer, Android Junit, Instrumentation, UIAutomator, iOS:基于 Instrument 的 iOS

(2) UI 自动化

(3) 性能测试

1) Web 前端性能测试

网络抓包工具: Wireshark

网页文件大小

webpagetest

pagespeed insight

chrome adb

2) APP 端性能测试

Android 内存占用分析: MAT

iOS 内存问题分析: ARC 模式

Android WebView 性能分析:

iOS WebView 性能分析

3) 后台服务性能测试负载, 压力, 耐久性

(4) 可拓展性, 基准

工具: apacheAB, Jmeter, LoadRunner

(5) 专项测试

1) 兼容性测试

手工测试: 操作系统, 分辨率, rom, 网络类型

云平台：testin，脚本编写，Android。

2) 流量测试

Android 自带的流量管理，

iOS 自带的 Network

tcpdump 抓包

WiFi 代理抓包：Fiddler

流量节省方法：压缩数据，json 优于 xml；WebP 优于传统的 JPG，PNG；控制访问的频次；只获取必要的数 据；缓存；

3) 电量测试

基于测试设备的方法，购买电量表进行测试。

GSam Battery Monitoe Pro

iOS 基于 Instrument Energy 工具

4) 弱网络测试

手机自带的网络状况模拟工具

基于代理的弱网络的模拟：

工具：windows：Network Delay Simulator

Mac：Network Link Conditioner

21、请你说一说 bug 的周期，以及描述一下不同类别的 bug。

（1）New:（新的）

当某个“bug”被第一次发现的时候，测试人员需要与项目负责人沟通以确认发现的的确是一个 bug，如果被确认是一个 bug，就将其记录下来，并将 bug 的状态设为 New

（2）Assigned（已指派的）

当一个 bug 被指认为 New 之后，将其反馈给开发人员，开发人员将确认这是否是一个 bug，如果是，开发组的负责人就将这个 bug 指定给某位开发人员处理，并将 bug 的状态设定为

“Assigned”

(3) Open (打开的)

一旦开发人员开始处理 bug 的时候，他（她）就将这个 bug 的状态设置为“Open”，这表示开发人员正在处理这个“bug”

(4) Fixed (已修复的)

当开发人员进行处理（并认为已经解决）之后，他就可以将这个 bug 的状态设置为“Fixed”并将其提交给开发组的负责人，然后开发组的负责人将这个 bug 返还给测试组

(5) Pending Reset (待测试的) 当 bug 被返还到测试组后，我们将 bug 的状态设置为 Pending Reset”

(6) Reset(再测试) 测试组的负责人将 bug 指定给某位测试人员进行再测试，并将 bug 的状态设置为“Reset”

(7) Closed (已关闭的) 如果测试人员经过再次测试之后确认 bug 已经被解决之后，就将 bug 的状态设置为“Closed”

(8) Reopen (再次打开的) 如果经过再次测试发现 bug (指 bug 本身而不是包括因修复而引发的新 bug) 仍然存在的话，测试人员将 bug 再次传递给开发组，并将 bug 的状态设置为“Reopen”

(9) Pending Reject (拒绝中) 如果测试人员传递到开发组的 bug 被开发人员认为是正常行为而不是 bug 时，这种情况下开发人员可以拒绝，并将 bug 的状态设置为“Pending Reject”

(10) Rejected(被拒绝的) 测试组的负责人接到上述 bug 的时候，如果他（她）发现这是产品说明书中定义的正常行为或者经过与开发人员的讨论之后认为这并不能算作 bug 的时候，开发组负责人就将这个 bug 的状态设置为“Rejected”

(11) Postponed (延期) 有些时候，对于一些特殊的 bug 的测试需要搁置一段时间，事实上有很多原因可能导致这种情况的发生，比如无效的测试数据，一些特殊的无效的功能等等，在这种情况下，bug 的状态就被设置为“Postponed”

不同类别的 bug:

Bug 类型

- 代码错误
- 界面优化

- 设计缺陷
- 配置相关
- 安装部署
- 安全相关
- 性能问题
- 标准规范
- 测试脚本
- 其他

22、请你说一说 PC 网络故障，以及如何排除障碍。

（1）首先是排除接触故障，即确保你的网线是可以正常使用的。然后禁用网卡后再启用，排除偶然故障。打开网络和共享中心窗口，单击窗口左上侧“更改适配器设置”右击其中的“本地连接“或”无线网络连接”，单击快捷菜单中的“禁用”命令，即可禁用所选网络。接下来重启网络，只需右击后单击启用即可。

（2）使用 `ipconfig` 查看计算机的上网参数

1) 单击“开始|所有程序|附件|命令提示符“，打开命令提示符窗口 2、输入 `ipconfig`，按 `Enter` 确认，可以看到机器的配置信息，输入 `ipconfig/all`，可以看到 IP 地址和网卡物理地址等相关网络详细信息。

（3）使用 `ping` 命令测试网络的连通性，定位故障范围在命令提示符窗口中输入” `ping 127.0.0.1` “，数据显示本机分别发送和接受了 4 个数据包， 丢包率为零，可以判断本机网络协议工作正常，如显示”请求超时“，则表明本机网卡的安装或 TCP/IP 协议有问题，接下来就应该检查网卡和 TCP/IP 协议，卸载后重装即可。

（4）`ping` 本机 IP 在确认 27.0.0.1 地址能被 `ping` 通的情况下，继续使用 `ping` 命令测试本机的 IP 地址能否被 `ping` 通，如不能，说明本机的网卡驱动程序不正确，或者网卡与网线之间连接有故障，也有可能是本地的路由表面收到了破坏，此时应检查本机网卡的状态是否为己连接，网络参数是否设置正确，如果正确可是不能 `ping` 通，就应该重新安装网卡驱动程序。丢失率为零，可以判断网卡安装配置没有问题，工作正常。

（5）`ping` 网关网关地址能被 `ping` 通的话，表明本机网络连接以及正常，如果命令不成功，可能是网关设备自身存在问题，也可能是本机上网参数设置有误，检查网络参数。

23、请你说一说测试的常用方法。(同 18)

24、请你说一说你知道的自动化测试框架。

(1) 模块化测试框架

模块化测试脚本框架 (TEST MODULARITY FRAMEWORK) 需要创建小而独立的可以描述的模块、片断以及待测应用程序的脚本。这些树状结构的小脚本组合起来, 就能组成能用于特定的测试用例的脚本。在五种框架中, 模块化框架是最容易掌握和使用的。在一个组件上方建立一个抽象层使其在余下的应用中隐藏起来, 这是众所周知的编程技巧。这样应用同组件中的修改隔离开来, 提供了程序设计的模块化特性。模块化测试脚本框架使用这一抽象或者封装的原理来提高自动测试组合的可维护性和可升级性。

(2) 测试库框架

测试库框架 (Test Library Architecture) 与模块化测试脚本框架很类似, 并且具有同样的优点。不同的是测试库框架把待测应用程序分解为过程和函数而不是脚本。这个框架需要创建描述模块、片断以及待测应用程序的功能库文件。

(3) 关键字驱动或表驱动的测试框架

对于一个独立于应用的自动化框架, 关键字驱动 (KEYWORD DRIVEN) 测试和表驱动 (TABLE DRIVEN) 测试是可以互换的术语。这个框架需要开发数据表和关键字。这些数据表和关键字独立于执行它们的测试自动化工具, 并可以用来“驱动”待测应用程序和数据的测试脚本代码, 关键字驱动测试看上去与手工测试用例很类似。在一个关键字驱动测试中, 把待测应用程序的功能和每个测试的执行步骤一起写到一个表中。这个测试框架可以通过很少的代码来产生大量的测试用例。同样的代码在用数据表来产生各个测试用例的同时被复用。

(4) 数据驱动测试框架

数据驱动 (DATA DRIVEN) 测试是一个框架。在这里测试的输入和输出数据是从数据文件中读取 (数据池, ODBC 源, CSV 文件, EXCEL 文件, ADO 对象等) 并且通过捕获工具生成或者手工生成的代码脚本被载入到变量中。在这个框架中, 变量不仅被用来存放输入值还被用来存放输出的验证值。整个程序中, 测试脚本来读取数值文件, 记载测试状态和信息。这类似于表驱动测试, 在表驱动测试中, 它的测试用例是包含在数据文件而不是在脚本中, 对于数据而言, 脚本仅仅是一个“驱动器”, 或者是一个传送机构。然而, 数据驱动测试不同于表驱动测试, 尽管导航数据并不包含在表结构中。在数据驱动测试中, 数据文件中只包含测试数据。这个框架意图减少需要执行所有测试用例所需要的总的测试脚本数。数据驱动需要很少的代码来产生大量的测试用例, 这与表驱动极其类似。

(5) 混合测试自动化 (Hybrid Test Automation) 框架

最普遍的执行框架是上面介绍的所有技术的一个结合, 取其长处, 弥补其不足。这个混合测试框架是由大部分框架随着时间并经过若干项目演化而来的。

25、请你说一说 web 测试和 app 测试的不同点。

(1) 系统架构方面:

web 项目，一般都是 b/s 架构，基于浏览器的

app 项目，则是 c/s 的，必须要有客户端，用户需要安装客户端。

web 测试只要更新了服务器端，客户端就会同步会更新。

App 项目则需要客户端和服务器都更新。

(2) 性能方面:

web 页面主要会关注响应时间

而 app 则还需要关心流量、电量、CPU、GPU、Memory 这些。它们服务端的性能没区别，都是一台服务器。

(3) 兼容方面: web 是基于浏览器的，所以更倾向于浏览器和电脑硬件，电脑系统的方向的兼容

app 测试则要看分辨率，屏幕尺寸，还要看设备系统。

web 测试是基于浏览器的所以不必考虑安装卸载。

而 app 是客户端的，则必须测试安装、更新、卸载。除了常规的安装、更新、卸载还要考虑到异常场景。包括安装时的中断、弱网、安装后删除安装文件。

此外 APP 还有一些专项测试：如网络、适配性。

26、请问黑盒测试和白盒测试有哪些方法？

黑盒测试方法有等价类划分，边界值分析，错误推测，因果图法

白盒测试方法有逻辑覆盖法，程序插桩技术，基本路径法，符号测试，错误驱动测试

27、请问你怎么看待测试，知道哪些测试的类型，有用过哪些测试方法？

测试是软件开发中不可或缺的一环，测试通过经济，高效的方法，捕捉软件中的错误，从而达到保重软件内在质量的目的。

测试分为功能测试和非功能测试，非功能测试又可以分为性能测试、压力测试、容量测试、

健壮性测试、安全性测试、可靠性测试、恢复性测试、备份测试、协议测试、兼容性测试、可用性测试、配置测试、GUI 测试。

测试方法用过等价划分法、边值分析法、错误推测法、因果图法。

28、请问你怎么测试网络协议？

协议测试包括四种类型的测试

一致性测试：检测协议实现本身与协议规范的符合程度

互操作性测试：基于某一协议检测不同协议实现间互操作互通信的能力

③性能测试：检测协议实现的性能指标，比如数据传输速度，连接时间，执行速度，吞吐量，并发度，

④健壮性测试：检测协议是现在各种恶劣环境下运行的能力，比如注入干扰报文，通信故障，信道被切断。

29、请你回答一下什么是 α 测试和 β 测试，以及什么时候用到他们？

α 测试：在受控的环境中进行，由用户在开发者的场所进行，并且在开发者对用户的指导下进行测试，开发者负责记录发现的错误和使用中遇到的问题；

β 测试：在开发者不能控制的环境中的真实应用，由软件的最终用户们在一个或多个客户场所下进行，由用户记录在测试中遇到的一系列问题，并定期报给开发者。