软件测试面试题

## **软件测试面试题100道整理**

###### **1.什么是软件测试？**

答：软件测试是为了发现错误而执行程序的过程。

###### **2.软件测试的目的?**

答;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。

###### **3.什么是需求文档测试？**

答：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;

###### **4.什么是设计文档测试?**

答:测试设计是否符合全部需求以及设计是否合理。

###### **5.什么是α测试?**

答:Alpha测试(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。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。

###### **6.什么是β测试?**

答：Beta测试(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。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。

###### **7.什么是驱动模块？**

答:驱动模块在大多数场合称为"主程序"，它接收测试数据并将这些数据传递到被测试模块.单元测试一个函数单元时，被测单元本身是不能独立运行的，需要为其传送数据，为此写驱动  
驱动模块主要完成以下事情：  
1、接受测试输入;  
2、对输入进行判断;  
3、将输入传给被测单元，驱动被测单元执行;  
4、接受被测单元执行结果，并对结果进行判断;  
5、将判断结果作为用例执行结果输出测试报告。

###### **8.什么是桩模块?**

答:比如对函数A做单元测试时，被测的函数单元下还包括了一个函数B，为了更好的错误，定位错误，就要为函数B写桩，来模拟函数B的功能，保证其正确。

###### **9.什么是白盒测试?**

答:白盒测试(White-box Testing，又称逻辑驱动测试,结构测试),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有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。  
对开发语言的支持：白盒测试工具是对源代码进行的测试，测试的主要内容包括词法分析与语法分析、静态错误分析、动态检测等。目前测试工具主要支持的开发语言包括：标准C、C++、Visual C++、Java、Visual J++等。

###### **10.什么是静态测试?**

答:通过运行程序测试软件称为动态测试.通过评审文档、阅读代码等方式测试软件称为静态测试,在动态测试中,通常使用白盒测试和黑盒测试从不同的角度设计测试用例,查找软件代码中的错误.  
静态测试方法是指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。对需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行来找错。静态方法通过程序静态特性的分析，找出欠缺和可疑之处，例如不匹配的参数、不适当的循环嵌套和分支嵌套、不允许的递归、未使用过的变量、空指针的引用和可疑的计算等。静态测试结果可用于进一步的查错，并为测试用例选取提供指导。

###### **11.什么是回归测试?**

答:回归测试的目的是在程序有修改的情况下，保证原有功能正常的一种测试策略和方法。  
说白了就是，我们测试人员在对程序进行测试时发现bug，然后返还程序员修改，程序员修改后发布新的软件包或新的软件补丁包给我们测试人员，我们就要重新对这个程序测试，已保证程序在修正了以前bug的情况下，正常运行，且不会带来新的错误的这样一个过程。 一般情况下是不需要全面测试的，而是根据修改的情况进行有效的测试。

###### **12.白盒测试有哪几种方法?**

答: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有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。

###### **13.软件的缺陷等级应如何划分?**

软件缺陷的等级可以用严重性和优先级来描述；  
严重性：衡量缺陷对客户满意度影响的满意程度，分为  
1,致命错误，可能导致本模块以及其他相关的模块异常，死机等问题；  
2.严重错误，问题局限在本模块，导致模块功能失常或异常退出；  
3.一般错误，模块功能部分失效；  
4.建议模块，有问题提出人对测试模块的改进建议；  
优先级：缺陷被修复的紧急程度；  
1.立即解决（P1级）：缺陷导致系统功能几乎不能使用或者测试不能继续，需立即修复；  
2.高优先级（P2级）：缺陷严重，影响测试，需优先考虑；  
3.正常排队（P3级）：缺陷需要正常排队等待修复；  
4.低优先级（P4级）：缺陷可以在有时间的时候被纠正；

###### **14.如果能够执行完美的黑盒测试，还需要进行白盒测试吗?(白盒与黑盒的区别)**

答:任何工程产品(注意是任何工程产品)都可以使用以下两种方法之一进行测试。  
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。  
以上事实说明，软件测试有一个致命的缺陷，即测试的不完全、不彻底性。由于任何程序只能进行少量(相对于穷举的巨大数量而言)的有限的测试，在未发现错误时，不能说明程序中没有错误。

###### **15.软件测试应该划分几个阶段?简述各个阶段应重点测试的点?各个阶段的含义?**

答:大体上来说可分为单元测试,集成测试,系统测试,验收测试,每个阶段又分为以下五个步骤: 测试计划，测试设计，用例设计，执行结果，测试报告  
初始测试集中在每个模块上，保证源代码的正确性，该阶段成为单元测试，主要用白盒测试方法。 接下来是模块集成和集成以便组成完整的软件包。集成测试集中在证实和程序构成问题上。主要采用黑盒测试方法，辅之以白盒测试方法。  
软件集成后，需要完成确认和系统测试。确认测试提供软件满足所有功能、性能需求的最后保证。确认测试仅仅应用黑盒测试方法。

###### **16.什么是单元测试？**

答:单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。

###### **17.什么是集成测试**

答:集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。

###### **18.系统测试？**

答:系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确并非一项简单的任务，它被称为测试的“先知者问题”。

###### **19.验收测试**

答:验收测试旨在向软件的购买者展示该软件系统满足其用户的需求。它的测试数据通常是系统测试的测试数据的子集.

###### **20.回归测试**

答:回归测试是在软件维护阶段，对软件进行修改之后进行的测试。其目的是检验对软件进行的修改是否正确。

###### **21.针对缺陷采取怎样的管理措施?**

答:1. 要更好的管理缺陷，必须引入缺陷管理工具，商用的或者开源的都可。  
2. 根据缺陷的生命周期，考虑缺陷提交的管理、缺陷状态的管理和缺陷分析的管理。  
3. 所有发现的缺陷(不管是测试发现的还是走读代码发现的)都必须全部即时的、准确的提交到缺陷管理工具中，这是缺陷提交的管理。  
4. 缺陷提交后，需要即时的指派给相应的开发人员，提交缺陷的人需要密切注意缺陷的状态， 帮助缺陷的尽快解决。缺陷解决后需要即时对缺陷的修复进行验证。这样的目的有两个：一个是让缺陷尽快解决;二是方便后面缺陷的分析(保证缺陷相关的信息准确，如龄期等)，这是缺陷状态的管理。  
5. 为了更好的改进开发过程和测试过程，需要对缺陷进行分析，总结如缺陷的类别、缺陷的龄期分布等信息，这是缺陷分析的管理。

###### **22.单元测试、集成测试、系统测试的侧重点是什么?**

答: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，测试重点是系统的模块，包括子程序的正确性验证等。 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。测试重点是模块间的衔接以及参数的传递等。  
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。测试重点是整个系统的运行以及与其他软件的兼容性。

###### **23.设计用例的方法、依据有那些?**

答:白盒测试用例设计有如下方法:基本路径测试\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  
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。

##### **24.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**

答:1) 测试人员或开发人员发现bug后，判断属于哪个模块的问题，填写bug报告后，系统会自动通过Email通知项目组长或直接通知开发者。  
2) 经验证无误后，修改状态为VERIFIED.待整个产品发布后，修改为CLOSED.  
3) 还有问题，REOPENED，状态重新变为“New"，并发邮件通知。  
4) 项目组长根据具体情况，重新reassigned分配给bug所属的开发者。  
5) 若是，进行处理，resolved并给出解决方法。(可创建补丁附件及补充说明)  
6) 开发者收到Email信息后，判断是否为自己的修改范围。  
7) 若不是，重新reassigned分配给项目组长或应该分配的开发者。  
8) 测试人员查询开发者已修改的bug，进行重新测试。

###### **25.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**

答;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，进行时要求所有隶属于某个给顶层次的模块总是存在的，也不再有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。

###### **26.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**

答：这些测试步骤分别在软件开发的不同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。

###### **27.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？**

参考回答：

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。

应用场景：

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。

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。

###### **28.请问测试开发需要哪些知识？需要具备什么能力？**

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；

考编程语言基础，如C/C++、java、python等；

自动化测试工具，如Selenium、Appium、Robotium等；

计算机基础知识，如数据库、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。

###### **29.请说一说黑盒与白盒的测试方法**

参考回答：

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盆子，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锯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。

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：1. 保证一个模块中的所有独立路径至少被测试一次；2. 所有逻辑值均需要测试真（true）和假（false）；两种情况；3. 检查程序的内部数据结构，保证其结构的有效性；4. 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：

1.语句覆盖每条语句至少执行一次。

2.判定覆盖每个判定的每个分支至少执行一次。

3.条件覆盖每个判定的每个条件应取到各种可能的值。

4.判定/条件覆盖同时满足判定覆盖条件覆盖。

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。

###### **30.请说一下手动测试与自动化测试的优缺点**

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。

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、工具本身并无想像力

###### **31.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。

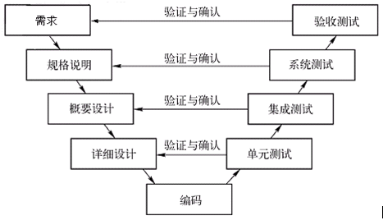
###### **32.你觉得软件测试的核心竞争力是什么**

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。

###### **33.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**

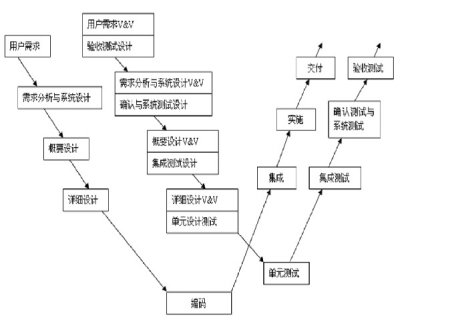
参考回答：

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。

###### **34.你觉得单元测试可行吗**

参考回答：

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。

###### **35.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

###### **36.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

###### **37.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

###### **38.请问你觉得测试项目具体工作是什么？**

答;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

###### **39.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。

###### **40.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

###### **41.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：  
a.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。

b.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。

c.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。

d.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。

e.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。

f.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。

###### **42.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，他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：1. 保证一个模块中的所有独立路径至少被测试一次；2. 所有逻辑值均需要测试真（true）和假（false）；两种情况；3. 检查程序的内部数据结构，保证其结构的有效性；4. 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：

1.语句覆盖每条语句至少执行一次。

2.判定覆盖每个判定的每个分支至少执行一次。

3.条件覆盖每个判定的每个条件应取到各种可能的值。

4.判定/条件覆盖同时满足判定覆盖条件覆盖。

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。

###### **43.请你说一说测试工程师的必备技能**

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、Robotium等；

• 计算机基础知识，如数据库、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。

###### **44.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为了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 dumpsys 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

###### **45.请你说一说app测试的工具**

参考回答：

功能测试自动化  
a) 轻量接口自动化测试

jmeter，

b) APP UI层面的自动化

android：UI Automator Viewer，Android Junit，Instrumentation，UIAutomator，

iOS:基于Instrument的iOS UI自动化，

性能测试

a) 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

b) 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

c) 后台服务性能测试

负载，压力，耐久性

可拓展性，基准

工具：apacheAB，Jmeter，LoadRunner，

专项测试

a) 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。

b) 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；

c) 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

d) 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

###### **46.请你说一说bug的周期，以及描述一下不同类别的bug**

参考回答：

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（待在测试的）

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

###### **47.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127.0.0.1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已连接，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。

###### **48.请你说一说测试的常用方法**

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盆子，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锯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。

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：1. 保证一个模块中的所有独立路径至少被测试一次；2. 所有逻辑值均需要测试真（true）和假（false）；两种情况；3. 检查程序的内部数据结构，保证其结构的有效性；4. 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：

1.语句覆盖每条语句至少执行一次。

2.判定覆盖每个判定的每个分支至少执行一次。

3.条件覆盖每个判定的每个条件应取到各种可能的值。

4.判定/条件覆盖同时满足判定覆盖条件覆盖。

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。

###### **49.请你说一说你知道的自动化测试框架**

参考回答：

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）I9LJJ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，关键宇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），LJ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

###### **50.请你说一说web测试和app测试的不同点**

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。

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。

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。

###### **51.请问你了解什么测试方法**

参考回答：

等价类划分，边界值分析，错误推测，因果图法，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

###### **52.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

###### **53.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。

###### **54.请问你怎么测试网络协议?**

协议测试包括四种类型的测试  
1、一致性测试：检测协议实现本身与协议规范的符合程度

2、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