- 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。或者说，软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例(即输入数据及其预期的输出结果)，并利用这些测试用例去运行程序，以发现程序错误的过程。

- 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。

- 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;

- 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。

- 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。

- 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。

- 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。

- 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。

- 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。

- 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。

- 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。

- 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。

- 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；

- 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。

- 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。

- 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。

- 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。

- 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”。

- 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.

- 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。

- 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。

- 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。

- 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\等价类划分\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。

- 测试用例通常包括那些内容?着重阐述编制测试用例的具体做法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……)

答: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，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。

- 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。