1. 什么是软件测试
2. 软件测试的目的
3. 什么是需求文档测试？

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。

1. 什么是设计文档测试？

测试设计是否符合全部需求以及设计是否合理。

1. α测试：
2. 用户在开发环境下进行的测试。
3. 不能由程序员或者测试员完成。
4. 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。
5. 可以从软件产品编码结束之后开始，或在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。
6. β测试：
7. 用户在实际使用环境下进行的测试。
8. 开发者不在现场，不能由程序员或者测试员完成。
9. 目的是测试可支持性。注重产品的支持性，包括文档，客户培训和支持产品的生产能力。
10. 在α测试达到一定的可靠程度后，才开始β测试。
11. 驱动模块

1）就是主程序（main函数）。

2）主要完成：对输入进行判断，给被测单元传入数据，接受被测单元执行结果并对结果进行判断，将判断结果作为用例执行结果输出测试报告。

1. 桩模块

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

1. 白盒测试

1）主要方法有：逻辑驱动、基路测试等，主要用于软件验证。是穷举路径测试。

2）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

3）缺点：没有关注到功能

1. 静态测试&动态测试
2. 动态测试：运行程序测试软件。通常使用白+黑从不同的角度设计测试用例，查找软件代码中的错误。
3. 静态测试：不运行程序，通过评审文档、阅读代码等检查语法、结构、过程、接口等来检查程序正确性。
4. 缺陷等级

严重性：

1. 致命错误：可能导致模块异常、死机等问题；
2. 严重错误：问题仅限于本模块，导致模块功能失常或异常退出
3. 一般错误：模块部分失效
4. 建议错误：改进建议

优先级：P1, P2, P3, P4

1. 缺陷管理

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。

1. 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。

1. 测试阶段
2. 单元测试、集成测试、系统测试、验收测试
3. 每个阶段分为5个步骤：测试计划、测试设计、用例设计、执行结果、测试报告
4. 集成测试

在软件系统集成过程中所进行的测试，目的是检验软件单位之间的接口是否正确

1. 系统测试

对集成好的软件系统进行彻底的测试，验证软件系统的正确性和性能等是否满足其规约所指定的要求

1. 验收测试

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。

1. 回归测试

软件维护阶段，对软件进行修改之后进行的测试。 目的是检验对软件进行的修改是否正确。

1. 单元测试、集成测试、系统测试的侧重点是什么

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。

1. 设计用例的方法、依据有哪些
2. 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
3. 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。
4. 单元测试、集成测试、系统测试、验收测试、回归测试中最重要的是哪一步

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。

1. 集成测试和系统测试的区别，以及应用场景

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。

1. 手动测试和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

24. 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。

1. 自动化测试的意义，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

1. 测试流程

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

1. 测试项目的具体工作

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

1. 边界值分析法

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