

软件测试面试题汇总

测试技术面试题

.....	5
1、什么是兼容性测试？兼容性测试侧重哪些方面？	5
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？	5
3、测试的策略有哪些？	5
4、正交表测试用例设计方法的特点是什么？	5
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？	5
6、你觉得 bugzilla 在使用的过程中，有什么问题？	5
7、描述测试用例设计的完整过程？	6
8、单元测试的策略有哪些？	6
9、LoadRunner 分哪三部分？	6
10、LoadRunner 进行测试的流程？.....	6
什么是并发？在 lordrunner 中，如何进行并发的测试？集合点失败了会怎么样？	6
12、使用 QTP 做功能测试，录制脚本的时候，要验证多个用户的登录情况/查询情况，如何操作？	6
13、QTP 中的 Action 有什么作用？有几种？	6
14、TestDirector 有些什么功能，如何对软件测试过程进行管理？	7
15、你所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）？	7
16、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？	8
17、Beta 测试与 Alpha 测试有什么区别？	8
18、软件的评审一般由哪些人参加？其目的是什么？	8
19、测试活动中，如果发现需求文档不完善或者不准确，怎么处理？	8
20、阶段评审与项目评审有什么区别？	8
21、阐述工作版本的定义？	8
22、什么是桩模块？什么是驱动模块？	8
23、什么是扇入？什么是扇出？	8
24、你认为做好测试计划工作的关键是什么？	8
25、你认为做好测试用例工作的关键是什么？	9
26、简述一下缺陷的生命周期？	9
27、软件的安全性应从哪几个方面去测试？	9
28、软件配置管理工作开展的情况和认识？	9
29、你觉得软件测试通过的标准应该是什么样的？	10
30、引入测试管理的含义？	10
31、一套完整的测试应该由哪些阶段组成？	10
32、单元测试的主要内容？	10
33、集成测试也叫组装测试或者联合测试，请简述集成测试的主要内容？	10
34、简述集成测试与系统测试关系？	10
35、软件测试的文档测试应当贯穿于软件生命周期的全过程，其中用户文档是文档测试的重点。那么软件系统的用户文档包括哪些？	10
36、软件系统中除用户文档之外，文档测试还应该关注哪些文档？	10
37、简述软件系统中用户文档的测试要点？	11
38、单元测试主要内容是什么？	11
39、如何理解强度测试？	13
40、如何理解压力、负载、性能测试测试？	13
41、什么是系统瓶颈？	13
42、文档测试主要包含什么内容？	13
43、功能测试用例需要详细到什么程度才是合格的？	14

44、配置和兼容性测试的区别是什么？	14
45、软件文档测试主要包含什么？	15
46、没有产品说明书和需求文档地情况下能够进行黑盒测试吗？	15
47、测试中的“杀虫剂怪事”是指什么？	15
48、在配置测试中，如何判断发现的缺陷是普通问题还是特定的配置问题？	15
49、为什么尽量不要让时间有富裕的员工去做一些测试？	16
50、完全测试程序是可能的吗？	16
51、 软件测试的风险主要体现在哪里？	16
52、发现的缺陷越多，说明软件缺陷越多吗？	16
53、所有的软件缺陷都能修复吗？所有的软件缺陷都要修复吗？	17
54、软件测试人员就是 QA 吗？	17
55、 如何减少测试人员跳槽带来的损失？	17
56、测试产品与测试项目的区别是什么？	17
57、和用户共同测试（UAT 测试）的注意点有哪些？	18
58、如何编写提交给用户的测试报告？	18
59、测试工具在测试工作中是什么地位？	18
60、什么是软件测试，软件测试的目的？	18
61、简述负载测试与压力测试的区别。	19
62、写出 bug 报告流转的步骤，每步的责任人及主要完成的工作。	19
63、写出 bug 报告当中一些必备的内容。	19
64、开发人员老是犯一些低级错误怎么解决？	20
65、画出软件测试的 V 模型图。	20
66、为什么要在一个团队中开展软件测试工作？	20
67、您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？	20
68、您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）	20
69、您认为做好测试用例设计工作的关键是什么？	21
70、 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。	21
71、测试计划工作的目的是什么？测试计划工作的内容都包括什么？其中哪些是最重要的？	22
72、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。	22
73、请以您以往的实际工作为例，详细的描述一次测试用例设计的完整的过程。	23
74、您以往是否曾经从事过性能测试工作？如果有，请尽可能的详细描述您以往的性能测试工作的完整过程。	23
75、你对测试最大的兴趣在哪里？为什么？	23
76、你以前工作时的测试流程是什么？	24
77、当开发人员说不是 BUG 时，你如何应付？	24
78、软件的构造号与版本号之间的区别？BVT(BuildVerificationTest)	24
79、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？	25
80、您以往所从事的软件测试工作中，是否使用了一些工具来进行软件缺陷（ Bug ）的管理？如果有，请结合该工具描述软件缺陷（ Bug ）跟踪管理的流程。	25
81、您认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？	25
82、单元测试、集成测试、系统测试的侧重点是什么？	25
83、集成测试通常都有那些策略？	25
84、一个缺陷测试报告的组成	25
85、基于 WEB 信息管理系统测试时应考虑的因素有哪些？	25
86、软件测试项目从什么时候开始，？为什么？	26
87、需求测试注意事项有哪些？	26

88、简述一下缺陷的生命周期.....	26
89、你在你所在的公司是怎么开展测试工作的？是如何组织的？	26
90、你认为理想的测试流程是什么样子的？	26
91、您在从事性能测试工作时，是否使用过一些测试工具？如果有，请试述该工具的工作原理，并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。	26
92、软件测试活动的生命周期是什么？	26
93、请画出软件测试活动的流程图？	26
94、针对缺陷采取怎样管理措施？	26
95、什么是测试评估？测试评估的范围是什么？	26
96、如果能够执行完美的黑盒测试，还需要进行白盒测试吗？为什么？	26
97、测试结束的标准是什么？	26
98、软件验收测试除了 alpha ,beta 测试以外,还有哪一种?.....	26
99、 做测试多久了？以前做过哪些项目？你们以前测试的流程是怎样的？用过哪些测试工具？	27
100、请就如何在开发中进行软件质量控制说说你的看法.....	27
101、一套完整的测试应该由哪些阶段组成？分别阐述一下各个阶段。	27
102、软件测试的类型有那些？分别比较这些不同的测试类型的区别与联系。	27
103、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.....	27
104、在分别测试 winform 的 C/S 结构与测试 WEB 结构的软件是，应该采取什么样的方法分别测试？他们存在什么样的区别与联系？	27
105、在测试 winform 的 C/S 结构软件时，发现这个软件运行速度很慢，您会认为是什么原因？您会采取哪些方法去检查这个原因？	27
106、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.....	27
107、你都用什么测试方法 针对不同的产品或者系统或者模块，有不同的测试方法。总体而言有白盒测试和黑盒测试。	27
108、怎么编写案例 案例的编写与测试阶段的定义有很大的关系。系统测试和 unit 测试的案例可能不同。总体而言测试案例根据系统的需求而定。	27
109、怎么才能全面的测试到每一个点 测试的全面性主要需要在设计测试计划的时候考虑，从测试策略，产品需求等等多个角度考虑从而定义全部的测试点。	27
110、谈谈软件测试技术，以及如何提高.....	27
111、谈谈软件测试职业发展，以及个人的打算.....	27
112、谈谈软件测试在企业的地位，也可以结合软件生命周期来谈.....	27
113、一般公司里实际的软件测试流程是什么样的？你们公司又是怎样的？	27
114、软件工程师要具有那些素质？	27
115、你会哪些测试工具？怎么操作？	27
116、你能不能说下你的 3 到 5 年的职业计划（规划）	27
117、你觉得你来应聘有那些优势？	27
其他问题：（有可能清晰的思路比确切的答案更重要）	27

开发及环境搭建类面试题

.....	28
1、描述软件产生内存泄露的原因以及检查方式。（可以结合一种开发语言进行描述）	28
2、简述什么是值传递，什么是地址传递，两者区别是什么？	28
3、结构化程序设计和面向对象程序设计各自的特点及优缺点是什么？	28
4、简述什么是存储过程和触发器？	28
5、使用 C 语言编写一个函数，用于交换两个变量的值（地址传递）。	29
6、请简述 DNS、活动目录、域的概念。	29
7、 描述 TCP/IP 协议的层次结构，以及每一层中重要协议。	29
8、 简述子网掩码的用途。	29
9、说出 4 种以上常用的操作系统及其主要的应用范围（微软的操作系统除外）。	29

10、在 Linux 系统中，一个文件的访问权限是 755，其含义是什么？	29
11、Windows 操作系统中 PATH 环境变量的作用是什么？	30
12、Ghost 的主要用途和常用方法？	30
13、在 RedHat 中，从 root 用户切到 user1 用户，一般用什么命令？	30
14、Linux 中，一般怎么隐藏文件？	30
15、如何将自己的本地磁盘（D）做成 FTP 供远端主机使用？	30
16、对 RUP, CMM, CMMI, XP, PSP, TSP 的认识？	30
17、DNS 是什么，它是如何工作的？	31
18、防火墙如何保证安全的？主要有哪些？	31
19、目前流行的操作的系统有哪些？请举例说明安装操作系统的注意事项？	33
20、简述一下 c/s 模式或者 b/s 模式？	33
21、TCP/UDP 有哪些区别？	34
22、ISO 模型？HUB、tch、Router 是 ISO 的第几层设备？	34
23、内存有哪几种存储组织结构，请分别加以说明？	34

人力资源面试题

.....	34
1、你的测试职业发展是什么？你自认为做测试的优势在哪里？	34
2、你为什么想离开目前的职务？	34
3、你对我们公司了解有多少？	34
4、你找工作时，最重要的考虑因素为何？	34
5、为什么我们应该录取你？	34
6、请谈谈你个人的最大特色。	34
7、一个测试工程师应具备那些素质和技能？	35
8、您认为在测试人员同开发人员的沟通过程中，如何提高沟通的效率和改善沟通的效果？维持测试人员同开发团队中其他成员良好的人际关系的关键是什么？	35
9、在您以往的测试工作中，最让您感到不满意或者不堪回首的事情是什么？您是如何来对待这些事情的？	35
10、在即将完成这次笔试前，您是否愿意谈一些自己在以往的学习和工作中获得的工作经验和心得体会？ （可以包括软件测试、过程改进、软件开发或者与此无关的其他方面）	35
11、为什么选择测试这行？	35
12、你的工作通常能在时限内完成吗。（我想问一下就是她问这个问题的动机是什么）	35
13、通常你对于别人批评你会有什么样的反应	35
14、如果明知这样做不对，你还会依主管的指过去做吗？	35
15、如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理？	35
16、请就软件测试人员应该具备什么样的基本素质说说你的看法。	36
17、你在五年内的个人目标和职业目标分别是什么？	36
18、你怎样做出自己的职业选择？	36

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、测试测试

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。

什么是并发？在 lordrunner 中，如何进行并发的测试？集合点失败了会怎么样？

参考答案：

在同一时间点，支持多个不同的操作。

LoadRunner 中提供 IP 伪装，集合点，配合虚拟用户的设计，以及在多台电脑上设置，可以比较好的模拟真实的并发。

集合点，即是多个用户在某个时刻，某个特定的环境下同时进行虚拟用户的操作的。集合点失败，则集合点的才操作就会取消，测试就不能进行。

12、使用 QTP 做功能测试，录制脚本的时候，要验证多个用户的登录情况/查询情况，如何操作？

参考答案：

分析用户登录的基本情况，得出一组数据，通过性测试/失败性测试的都有（根据 TC 来设计这些数据），然后录制登录的脚本，将关键的数据参数化，修改脚本，对代码进行加强，调试脚本。

13、QTP 中的 Action 有什么作用？有几种？

参考答案：

Action 的作用

- 用 Action 可以对步骤集进行分组

-
- 步骤重组，然后被整体调用
 - 拥有自己的 **sheet**
 - 组合有相同需求的步骤，整体操作
 - 具有独立的对象仓库

Action 的种类

- 可复用 **Action**
- 不可复用 **Action**
- 外部 **Action**

14、TestDirector 有些什么功能，如何对软件测试过程进行管理？

参考答案：

需求管理

- 定义测试范围
- 定义需求树
- 描述需求树的功能点

测试计划

- 定义测试目标和测试策略。
- 分解应用程序，建立测试计划树。
- 确定每个功能点的测试方法。
- 将每个功能点连接到需求上，使测试计划覆盖全部的测试需求。
- 描述手工测试的测试步骤
- 指明需要进行自动测试的功能点

测试执行

- 定义测试集合。
- 为每个测试人员制定测试任务和测试日程安排。
- 运行自动测试。

缺陷跟踪

- 记录缺陷
- 查看新增缺陷，并确定哪些是需要修正的
- 相关技术人员修改缺陷
- 回归测试
- 分析缺陷统计图表，分析应用程序的开发质量。

15、你所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）？

参考答案：**Compatibility Testing**（兼容性测试），也称“**Configuration testing**（配置测试）”，测试软件是否和系统的其它与之交互的元素之间兼容，如：浏览器、操作系统、硬件等。验证测试对象在不同的软件和硬件配置中的运行情况。

Functional testing（功能测试），也称为 **behavioral testing**（行为测试），根据产品特征、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。

Performance testing（性能测试），评价一个产品或组件与性能需求是否符合的测试。包括负载测试、强度测试、数据库容量测试、基准测试等类型。

16、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？

参考答案：5C 标准

17、Beta 测试与 Alpha 测试有什么区别？

参考答案：**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

18、软件的评审一般由哪些人参加？其目的是什么？

参考答案：

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

19、测试活动中，如果发现需求文档不完善或者不准确，怎么处理？

参考答案：

测试需求分析 发现需求文档不完善或者不准确，应该立即和相关人员进行协调交流。

20、阶段评审与项目评审有什么区别？

参考答案：

阶段评审 对项目各阶段评审：对阶段成果和工作

项目评审 对项目总体评审：对工作和产品

21、阐述工作版本的定义？

参考答案：

构造号：**BUILD**

22、什么是桩模块？什么是驱动模块？

参考答案：

桩模块：被测模块调用模块

驱动模块 调用被测模块

23、什么是扇入？什么是扇出？

参考答案：

扇入：被调次数，扇出：调其它模块数目

24、你认为做好测试计划工作的关键是什么？

参考答案：

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。

25、你认为做好测试用例工作的关键是什么？

参考答案：

需求和设计文档的理解程度，对系统的熟悉程度

26、简述一下缺陷的生命周期？

参考答案：提交->确认->分配->修复->验证->关闭

27、软件的安全性应从哪几个方面去测试？

参考答案：

- (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

28、软件配置管理工作开展的情况和认识？

参考答案：

软件配置管理贯穿于软件开发、测试活动的始终，覆盖了开发、测试活动的各个环节，它的重要作用之一就是全面的管理保存各个配置项，监控各配置项的状态，并向项目经理及相关的人员报告，从而实现软件过程的控制。

软件测试配置管理包括 4 个最基本的活动：

- 配置项标识
- 配置项控制
- 配置项状态报告
- 配置审计

软件配置管理通常借助工具来辅助，主要有 MS SourceSafe、Rational ClearCase 等

29、你觉得软件测试通过的标准应该是什么样的？

参考答案：

缺陷密度值达到客户的要求

30、引入测试管理的含义？

参考答案：风险分析，进度控制、角色分配、质量控制

31、一套完整的测试应该由哪些阶段组成？

参考答案：测试计划、测试设计与开发、测试实施、测试评审与测试结论

32、单元测试的主要内容？

参考答案：

模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试

33、集成测试也叫组装测试或者联合测试，请简述集成测试的主要内容？

参考答案：

- (1) 在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
- (2) 一个模块的功能是否会对另一个模块的功能产生不利的影响；
- (3) 各个子功能组合起来，能否达到预期要求的父功能；
- (4) 全局数据结构是否有问题；
- (5) 单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

34、简述集成测试与系统测试关系？

参考答案：

- (1) 集成测试的主要依据概要设计说明书，系统测试的主要依据是需求设计说明书；
- (2) 集成测试是系统模块的测试，系统测试是对整个系统的测试，包括相关的软硬件平台、网络以及相关外设的测试。

35、软件测试的文档测试应当贯穿于软件生命周期的全过程，其中用户文档是文档测试的重点。那么软件系统的用户文档包括哪些？

参考答案：

用户手册
安装和设置指导
联机帮助
指南、向导
样例、示例和模板
授权/注册登记表
最终用户许可协议

36、软件系统中除用户文档之外，文档测试还应该关注哪些文档？

参考答案：

开发文档
软件需求说明书

数据库设计说明书
概要设计说明书
详细设计说明书
可行性研究报告

管理文档

项目开发计划
测试计划
测试报告
开发进度月报
开发总结报告

37、简述软件系统中用户文档的测试要点？

参考答案：

- (1) 读者群。文档面向的读者定位要明确。对于初级用户、中级用户以及高级用户应该有不同的定位
- (2) 术语。文档中用到的术语要适用与定位的读者群，用法一致，标准定义与业界规范相吻合。
- (3) 正确性。测试中需检查所有信息是否真实正确，查找由于过期产品说明书和销售人员夸大事实而导致的错误。检查所有的目录、索引和章节引用是否已更新，尝试链接是否准确，产品支持电话、地址和邮政编码是否正确。
- (4) 完整性。对照软件界面检查是否有重要的分支没有描述到，甚至是否有整个大模块没有描述到。
- (5) 一致性。按照文档描述的操作执行后，检查软件返回的结果是否与文档描述的不同。
- (6) 易用性。对关键步骤以粗体或背景色给用户以提示，合理的页面布局、适量的图表都可以给用户更高的易用性。需要注意的是文档要有助于用户排除错误。不但描述正确操作，也要描述错误处理方法。文档对于用户看到的错误信息应当有更详细的文档解释。
- (7) 图表与界面截图。检查所有图表与界面截图是否与发行版本相同。
- (8) 样例与示例。像用户一样载入和使用样例。如果是一段程序，就输入数据并执行它。以每一个模块制作文件，确认它们的正确性。
- (9) 语言。不出现错别字，不要出现有二义性的说法。特别要注意的是屏幕截图或绘制图形中的文字。
- (10) 印刷与包装。检查印刷质量；手册厚度与开本是否合适；包装盒的大小是否合适；有没有零碎易丢失的小部件等等。

38、单元测试主要内容是什么？

参考答案：

单元测试大多数由开发人员来完成，测试人员技术背景较好或者开发系统软件时可能会安排测试人员进行单元测试，大多数进行的单元测试都是开发人员调试程序或者开发组系统联合调试的过程。讨论这个问题主要是扩充一下读者的视野。

单元测试一般包括五个方面的测试：

(1) 模块接口测试：模块接口测试是单元测试的基础。只有在数据能正确流入、流出模块的前提下，其他测试才有意义。模块接口测试也是集成测试的重点，这里进行的测试主要是为后面打好基础。测试接口正确与否应该考虑下列因素：

- 输入的实际参数与形式参数的个数是否相同；
- 输入的实际参数与形式参数的属性是否匹配；
- 输入的实际参数与形式参数的量纲是否一致；
- 调用其他模块时所给实际参数的个数是否与被调模块的形参个数相同；
- 调用其他模块时所给实际参数的属性是否与被调模块的形参属性匹配；
- 调用其他模块时所给实际参数的量纲是否与被调模块的形参量纲一致；
- 调用预定义函数时所用参数的个数、属性和次序是否正确；
- 是否存在与当前入口点无关的参数引用；

-
- 是否修改了只读型参数;
 - 对全程变量的定义各模块是否一致;
 - 是否把某些约束作为参数传递。

如果模块功能包括外部输入输出，还应该考虑下列因素：

- 文件属性是否正确;
- OPEN/CLOSE 语句是否正确;
- 格式说明与输入输出语句是否匹配;
- 缓冲区大小与记录长度是否匹配;
- 文件使用前是否已经打开;
- 是否处理了文件尾;
- 是否处理了输入/输出错误;
- 输出信息中是否有文字性错误。

- 局部数据结构测试;
- 边界条件测试;
- 模块中所有独立执行通路测试;

(2) 局部数据结构测试：检查局部数据结构是为了保证临时存储在模块内的数据在程序执行过程中完整、正确，局部功能是整个功能运行的基础。重点是一些函数是否正确执行，内部是否运行正确。局部数据结构往往是错误的根源，应仔细设计测试用例，力求发现下面几类错误：

- 不合适或不相容的类型说明;
- 变量无初值;
- 变量初始化或省缺值有错;
- 不正确的变量名（拼错或不正确地截断）;
- 出现上溢、下溢和地址异常。

(3) 边界条件测试：边界条件测试是单元测试中最重要的一项任务。众所周知，软件经常在边界上失效，采用边界值分析技术，针对边界值及其左、右设计测试用例，很有可能发现新的错误。边界条件测试是一项基础测试，也是后面系统测试中的功能测试的重点，边界测试执行的较好，可以大大提高程序健壮性。

(4) 模块中所有独立路径测试：在模块中应对每一条独立执行路径进行测试，单元测试的基本任务是保证模块中每条语句至少执行一次。测试目的主要是为了发现因错误计算、不正确的比较和不适当的控制流造成的错误。具体做法就是程序员逐条调试语句。常见的错误包括：

- 误解或用错了算符优先级;
- 混合类型运算;
- 变量初值错;
- 精度不够;
- 表达式符号错。

比较判断与控制流常常紧密相关，测试时注意下列错误：

- 不同数据类型的对象之间进行比较;
- 错误地使用逻辑运算符或优先级;
- 因计算机表示的局限性，期望理论上相等而实际上不相等的两个量相等;
- 比较运算或变量出错;
- 循环终止条件或不可能出现;
- 迭代发散时不能退出;
- 错误地修改了循环变量。

模块的各条错误处理通路测试：程序在遇到异常情况时不应该退出，好的程序应能预见各种出错条件，并预设各种出错处理通路。如果用户不按照正常操作，程序就退出或者停止工作，实际上也是一种缺陷，因此单元测试要测试各种错误处理路径。一般这种测试着重检查下列问题：

- 输出的出错信息难以理解;
- 记录的错误与实际遇到的错误不相符;
- 在程序自定义的出错处理段运行之前，系统已介入;
- 异常处理不当;

-错误陈述中未能提供足够的定位出错信息。

39、如何理解强度测试？

参考答案：

强度测试是为了确定系统在最差工作环境的工作能力，也可能是用于验证在标准工作压力下的各种资源的最下限指标。

它和压力测试的目标是不同的，压力测试是在标准工作环境下，不断增加系统负荷，最终测试出该系统能力达到的最大负荷（稳定和峰值），而强度测试则是在非标准工作环境下，甚至不断人为降低系统工作环境所需要的资源，如网络带宽，系统内存，数据锁等等，以测试系统在资源不足的情况下的工作状态，通过强度测试，可以确定本系统正常工作的最差环境。

强度测试和压力测试的测试指标相近，大多都是与时间相关的指标，如并发量（吞吐量），延迟（最大\最小\平均）以及顺序指标等

强度测试需要对系统的结构熟悉，针对系统的特征设计强度测试的方法

40、如何理解压力、负载、性能测试测试？

参考答案：

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。

41、什么是系统瓶颈？

参考答案：

瓶颈主要是指整个软硬件构成的软件系统某一方面或者几个方面能力不能满足用户的特定业务要求，“特定”是指瓶颈会在某些条件下会出现，因为毕竟大多数系统在投入前。

严格的从技术角度讲，所有的系统都会有瓶颈，因为大多数系统的资源配置不是协调的，例如CPU使用率刚好达到100%时，内存也正好耗尽的系统不是很多见。因此我们讨论系统瓶颈要从应用的角度讨论：关键是看系统能否满足用户需求。在用户极限使用系统的情况下，系统的响应仍然正常，我们可以认为改系统没有瓶颈或者瓶颈不会影响用户工作。

因此我们测试系统瓶颈主要是实现下面两个目的：

- 发现“表面”的瓶颈。主要是模拟用户的操作，找出用户极限使用系统时的瓶颈，然后解决瓶颈，这是性能测试的基本目标。
- 发现潜在的瓶颈并解决，保证系统的长期稳定性。主要是考虑用户在将来扩展系统或者业务发生变化时，系统能够适应变化。满足用户目前需求的系统不是最好的，我们设计系统的目标是在保证系统整个软件生命周期能够不断适应用户的变化，或者通过简单扩展系统就可以适应新的变化。

42、文档测试主要包含什么内容？

参考答案：

在国内软件开发管理中，文档管理几乎是最弱的一项，因而在测试工作中特别容易忽略文档测试也就不足为奇了。要想给用户提供的产品，文档测试是必不可少的。文档测试一般注重下面几个方面：

文档的完整性：主要是测试文档内容的全面性与完整性，从总体上把握文档的质量。例如用户手册应

该包括软件的所有功能模块。

描述与软件实际情况的一致性：主要测试软件文档与软件实际的一致程度。例如用户手册基本完整后，我们还要注意用户手册与实际功能描述是否一致。因为文档往往跟不上软件版本的更新速度。

易理解性：主要是检查文档对关键、重要的操作有无图文说明，文字、图表是否易于理解。对于关键、重要的操作仅仅只有文字说明肯定是不够的，应该附有图表使说明更为直观和明了。

文档中提供操作的实例：这项检查内容主要针对用户手册。对主要功能和关键操作提供的应用实例是否丰富，提供的实例描述是否详细。只有简单的图文说明，而无实例的用户手册看起来就像是软件界面的简单拷贝，对于用户来说，实际上没有什么帮助。

印刷与包装质量：主要是检查软件文档的商品化程度。有些用户手册是简单打印、装订而成，过于粗糙，不易于用户保存。优秀的文档例如用户手册和技术白皮书，应提供商品化包装，并且印刷精美。

43、功能测试用例需要详细到什么程度才是合格的？

参考答案：

这个问题也是测试工程师经常问的问题。有人主张测试用例详细到每个步骤执行什么都要写出来，目的是即使一个不了解系统的新手都可以按照测试用例来执行工作。主张这类写法的人还可以举出例子：欧美、日本等软件外包文档都是这样做的。

另外一种观点就是主张写的粗些，类似于编写测试大纲。主张这种观点的人是因为软件开发需求管理不规范，变动十分频繁，因而不能按照欧美的高标准来编写测试用例。这样的测试用例容易维护，可以让测试执行人员有更大的发挥空间。

实际上，软件测试用例的详细程度首先要以**覆盖到测试点**为基本要求。举个例子：“用户登陆系统”的测试用例可以不写出具体的执行数据，但是至少要写出五种以上情况（），如果只用一句话覆盖了这个功能是不合格的测试用例。覆盖功能点不是指列出功能点，而是要写出功能点的各个方面（如果组合情况较多时可以采用等价划分）。

另一个影响测试用例的就是组织的开发能力和测试对象特点。如果开发力量比较落后，编写较详细的测试用例是不现实的，因为根本没有那么大的资源投入，当然这种情况很随着团队的发展而逐渐有所改善。测试对象特点重点是指测试对象在进度、成本等方面的要求，如果进度较紧张的情况下，是根本没有时间写出高质量的测试用例的，甚至有些时候测试工作只是一种辅助工作，因而不编写测试用例。

因此，测试用例的编写要根据测试对象特点、团队的执行能力等各个方面综合起来决定编写策略。最后要注意的是测试人员一定不能抱怨，力争在不断提高测试用例编写水平的同时，不断地提高自身能力。

44、配置和兼容性测试的区别是什么？

参考答案：

配置测试的目的是保证软件在其相关的硬件上能够正常运行，而兼容性测试主要是测试软件能否与不同的软件正确协作。

配置测试的核心内容就是使用各种硬件来测试软件的运行情况，一般包括：

- (1) 软件在不同的主机上的运行情况，例如 Dell 和 Apple；
- (2) 软件在不同的组件上的运行情况，例如开发的拨号程序要测试在不同厂商生产的 Modem 上的运行情况；
- (3) 不同的外设；
- (4) 不同的接口；
- (5) 不同的可选项，例如不同的内存大小；

兼容性测试的核心内容：

- (1) 测试软件是否能在不同的操作系统平台上兼容；
- (2) 测试软件是否能在同一操作系统平台的不同版本上兼容；
- (3) 软件本身能否向前或者向后兼容；
- (4) 测试软件能否与其它相关的软件兼容；
- (5) 数据兼容性测试，主要是指数据能否共享；

配置和兼容性测试通称对开发系统类软件比较重要，例如驱动程序、操作系统、数据库管理系统等。具体进行时仍然按照测试用例来执行。

45、软件文档测试主要包含什么？

参考答案：

随着软件文档系统日益庞大，文档测试已经成为软件测试的重要内容。文档测试对象主要如下：

- 包装文字和图形；
- 市场宣传材料、广告以及其它插页；
- 授权、注册登记表；
- 最终用户许可协议；
- 安装和设置向导；
- 用户手册；
- 联机帮助；
- 样例、示范例子和模板；
-

文档测试的目的是提高易用性和可靠性，降低支持费用，因为用户通过文档就可以自己解决问题。因文档测试的检查内容主要如下：

- 读者对象——主要是文档的内容是否能让该级别的读者理解；
- 术语——主要是检查术语是否适合读者；
- 内容和主题——检查主题是否合适、是否丢失、格式是否规范等；
- 图标和屏幕抓图——检查图表的准确度和精确度；
- 样例和示例——是否与软件功能一致；
- 拼写和语法；
- 文档的关联性——是否与其它相关文档的内容一致，例如与广告信息是否一致；

文档测试是相当重要的一项测试工作，不但要给予充分的重视，更要认真的完成，象做功能测试一样来对待文档测试。

46、没有产品说明书和需求文档地情况下能够进行黑盒测试吗？

参考答案：

这个问题是国内测试工程师经常遇到的问题，根源就是国内软件开发文档管理不规范，对变更的管理方法就更不合理了。实际上没有任何文档的时候，测试人员是能够进行黑盒测试的，这种测试方式我们可以称之为探索测试，具体做法就是测试工程师根据自己的专业技能、领域知识等不断的深入了解测试对象、理解软件功能，进而发现缺陷。

在这种做法基本上把软件当成了产品说明书，测试过程中要和开发人员不断的进行交流。尤其在作项目的时候，进度压力比较大，可以作为加急测试方案。最大的风险是不知道有些特性是否被遗漏。

47、测试中的“杀虫剂怪事”是指什么？

参考答案：

“杀虫剂怪事”一词由 Boris Beizer 在其编著的《软件测试技术》第二版中提出。用于描述测试人员对同一测试对象进行的测试次数越多，发现的缺陷就会越来越少的现象。就像老用一种农药，害虫就会有免疫力，农药发挥不了效力。这种现象的根本原因就是测试人员对测试软件过于熟悉，形成思维定势。

为了克服这种现象，测试人员需要不断编写新的测试程序或者测试用例，对程序的不同部分进行测试，以发现更多的缺陷。也可以引用新人来测试软件，刚刚进来的新手往往能发现一些意想不到的问题。

48、在配置测试中，如何判断发现的缺陷是普通问题还是特定的配置问题？

参考答案：

在进行配置测试时，测试工程师仍然会发现一些普通的缺陷，也就是与配置环境无关的缺陷。因此判断新发现的问题，需要在不同的配置中重新执行发现软件缺陷的步骤，如果软件缺陷不出现了，就可能是配置缺陷；如果在所有的配置中都出现，就可能是普通缺陷。

需要注意的是，配置问题可以在一大类配置中出现。例如，拨号程序可能在所有的外置 Modem 中都存在问题，而内置的 Modem 不会有任何问题。

49、为什么尽量不要让时间有富裕的员工去做一些测试？

参考答案：

表面上看这体现了管理的效率和灵活性，但实际上也体现了管理者对测试的轻视。测试和测试的人有很大关系。测试工作人员应该是勤奋并富有耐心，善于学习、思考和发现问题，细心有条理，总结问题，如果具备这样的优点，做其它工作同样也会很出色，因此这里还有一个要求，就是要喜欢测试这项工作。如果他是专职的，那么肯定更有经验和信心。国内的小伙子好象都喜欢做程序员，两者工作性质不同，待遇不同，地位不同，对自我实现的价值认识也不同，这是行业的一个需要改善的问题。如果只是为了完成任务而完成任务，或者发现了几个问题就觉得满意了，这在任何其它工作中都是不行的。

50、完全测试程序是可能的吗？

参考答案：

软件测试初学者可能认为拿到软件后需要进行完全测试，找到全部的软件缺陷，使软件“零缺陷”发布。实际上完全测试是不可能的。主要有以下一个原因：

- 完全测试比较耗时，时间上不允许；
 - 完全测试通常意味着较多资源投入，这在现实中往往是行不通的；
 - 输入量太大，不能一一进行测试；
 - 输出结果太多，只能分类进行验证；
 - 软件实现途径太多；
 - 软件产品说明书没有客观标准，从不同的角度看，软件缺陷的标准不同；
- 因此测试的程度要根据实际情况确定。

51、软件测试的风险主要体现在哪里？

参考答案：

我们没有对软件进行完全测试，实际就是选择了风险，因为缺陷极有可能存在没有进行测试的部分。举个例子，程序员为了方便，在调试程序时会弹出一些提示信息框，而这些提示只在某种条件下会弹出，碰巧程序发布前这些代码中的一些没有被注释掉。在测试时测试工程师又没有对其进行测试。如果客户碰到它，这将是代价昂贵的缺陷，因为交付后才被客户发现。

因此，我们要尽可能的选择最合适的测试量，把风险降低到最小。

52、发现的缺陷越多，说明软件缺陷越多吗？

参考答案：

这是一个比较常见的现象。测试工程师在没有找到缺陷前会绞尽脑汁的思考，但是找到一个后，会接二连三的发现很多缺陷，颇有个人成就感。其中的原因主要如下：

- 代码复用、拷贝代码导致程序员容易犯相同的错误。类的继承导致所有的子类会包含基类的错误，反复拷贝同一代码意味可能也复制了缺陷。
- 程序员比较劳累是可以导致某些连续编写的功能缺陷较多。程序员加班是一种司空见惯的现象，因此体力不支时容易编写一些缺陷较多的程序。而这些连续潜伏缺陷恰恰是测试工程师大显身手的地方。

“缺陷一个连着一个”不是一个客观规律，只是一个常见的现象。如果软件编写的比较好，这种现象就不常见了。测试人员只要严肃认真的测试程序就可以了。

53、所有的软件缺陷都能修复吗？所有的软件缺陷都要修复吗？

参考答案：

从技术上讲，所有的软件缺陷都是能够修复的，但是没有必要修复所有的软件缺陷。测试人员要做的是能够正确判断什么时候不能追求软件的完美。对于整个项目团队，要做的是对每一个软件缺陷进行取舍，根据风险决定那些缺陷要修复。发生这种现象的主要原因如下：

-没有足够的时间资源。在任何一个项目中，通常情况下开发人员和测试人员都是不够用的，而且在项目中没有预算足够的回归测试时间，再加上修改缺陷可能引入新的缺陷，因此在交付期限的强大压力下，必须放弃某些缺陷的修改。

-有些缺陷只是特殊情况下出现，这种缺陷处于商业利益考虑，可以在以后升级中进行修复。

-不是缺陷的缺陷。我们经常会碰到某些功能方面的问题被当成缺陷来处理，这类问题可以以后有时间时考虑再处理。

最后要说的是，缺陷是否修改要由软件测试人员、项目经理、程序员共同讨论来决定是否修复，不同角色的人员从不同的角度来思考，以做出正确的决定。

54、软件测试人员就是 QA 吗？

参考答案：

软件测试人员的职责是尽可能早的找出软件缺陷，确保得以修复。而质量保证人员（QA）主要职责是创建或者制定标准和方法，提高促进软件开发能力和减少软件缺陷。测试人员的主要工作是测试，质量保证人员日常工作重要内容是检查与评审，测试工作也是测试保证人员的工作对象。

软件测试和质量是相辅相成的关系，都是为了提高软件质量而工作。

55、如何减少测试人员跳槽带来的损失？

参考答案：

在 IT 行业里跳槽已经是一种司空见惯的现象，而且跳槽无论给公司还是给个人都会带来一定的损失。测试队伍也无疑会面临跳槽的威胁，作为测试经理管理者，只有从日常工作中开始做起，最能最大限度的减少损失。建议我们从以下两个方面做起：

-加强部门内员工之间的互相学习，互相学习是建立学习型组织的基本要求，是知识互相转移的过程。在此基础上，可以把个人拥有的技术以知识的形式沉积下来，也就完成了隐性知识到显性知识的转化。

-通常情况下，企业能为员工提供足够大的发展空间时，如果不是待遇特别低，员工都不会主动离开企业。因此我们要想留住员工，管理者就应该把员工的个人成长和企业的发展联系起来，为员工设定合理发展规划并付诸实现。不过这项要求做起来比较，要有比较好的企业文化为依托。

56、测试产品与测试项目的区别是什么？

参考答案：

习惯上把开发完成后进行商业化、几乎不进行代码修改就可以售给用户使用的软件成为软件产品，也就是可以买“卖拷贝”的软件，例如 Windows2000。而通常把针对一个或者几个特定的用户而开发的软件成为软件项目，软件项目是一种个性化的产品，可以是按照用户要求全部重新开发，也可以修改已有的软件产品来满足特定的用户需求。项目和产品的不同特点，决定我们测试产品和测试项目仍然会有很多不同的地方：

-质量要求不同。通常产品的质量要高一些，修复发布后产品的缺陷成本较高，甚至会带来很多负面的影响。而做项目通常面向某一用户，虽然质量越高越好，但是一般只要满足用户要求就可以了。

-测试资源投入多少不同。做软件产品通常是研发中心来开发，进度压力要小些。同时由于质量要求高，因此会投入较多的人力、物力资源。

-项目最后要和用户共同验收测试，这是产品测试不具有的特点。

此外，测试产品与测试项目在缺陷管理方面、测试策略制定都会有很大不同，测试管理者应该结合具

体的环境，恰如其分的完成工作。

57、和用户共同测试（UAT 测试）的注意点有哪些？

参考答案：

软件产品在投产前，通常都会进行用户验收测试。如果用户验收测试没有通过，直接结果就是那不到“Money”，间接影响是损害了公司的形象，而后者的影响往往更严重。根据作者的经验，用户验收测试一定要让用户满意。

实际上用户现场测试更趋于是一种演示。在不欺骗用户的前提下，我们向用户展示我们软件的优点，最后让“上帝”满意并欣然掏出“银子”才是我们的目标。因此用户测试要注意下面的事项：

(1) 用户现场测试不可能测试全部功能，因此要测试核心功能。这需要提前做好准备，这些核心功能一定要预先经过测试，证明没有问题才可以和用户共同进行测试。测试核心模块的目的是建立用户对软件的信心。当然如果这些模块如果问题较多，不应该进行演示。

(2) 如果某些模块确实有问题，我们可以演示其它重要的业务功能模块，必要时要向用户做成合理的解释。争得时间后，及时修改缺陷来弥补。

(3) 永远不能欺骗用户，蒙混过关。道理很简单，因为软件是要给用户用的，问题早晚会暴露出来，除非你可以马上修改。

和用户进行测试还要注意各种交流技巧，争取不但短期利益得到了满足，还要为后面得合作打好基础。

58、如何编写提交给用户的测试报告？

参考答案：

随着测试工作越来越受重视，开发团队向客户提供测试文档是不可避免的事情。很多人会问：“我们可以把工作中的测试报告提供给客户吗？”答案是否定的。因为提供内部测试报告，可能会让客户失去信心，甚至否定项目。

测试报告一般分为内部测试报告和外部测试报告。内部报告是我们在测试工作中的项目文档，反映了测试工作的实施情况，这里不过多讨论，读者可以参考相关教材。这里主要讨论一下外部测试报告的写法，一般外部测试报告要满足下面几个要求：

- 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。
- 总之，外部测试报告要小心谨慎的编写。

59、测试工具在测试工作中是什么地位？

参考答案：

国内的很多测试工程师对测试工具相当迷恋，尤其是一些新手，甚至期望测试工具可以取代手工测试。测试工具在测试工作中起的是辅助作用，一般用来提高测试效率。自动化测试弥补了手工测试的不足，减轻一定的工作量。实际上测试工具是无法替代大多数手工测试的，而一些诸如性能测试等自动化测试也是手工所不能完成的。

对于自动测试技术，应当依据软件的不同情况来分别对待，一般自动技术会应用在引起大量重复性工作的地方、系统的压力点、以及任何适合使用程序解决大批量输入数据的地方。然后再寻找合适的自动测试工具，或者自己开发测试程序。一定不要为了使用测试工具而使用。

60、什么是软件测试，软件测试的目的？

参考答案：

61、简述负载测试与压力测试的区别。

参考答案：

压力测试（**Stress Testing**）

压力测试的主要任务就是获取系统正确运行的极限，检查系统在瞬间峰值负荷下正确执行的能力。例如，对服务器做压力测试时就可以增加并发操作的用户数量；或者不停地向服务器发送请求；或一次性向服务器发送特别大的数据等。看看服务器保持正常运行所能达到的最大状态。人们通常使用测试工具来完成压力测试，如模拟上万个用户从终端同时登录，这是压力测试中常常使用的方法。

负载测试（**Volume Testing**）

用于检查系统在使用大量数据的时候正确工作的能力，即检验系统的能力最高能达到什么程度。例如，对于信息检索系统，让它使用频率达到最大；对于多个终端的分时系统，让它所有的终端都开动。在使整个系统的全部资源达到“满负荷”的情形下，测试系统的承受能力。

62、写出 bug 报告流转的步骤，每步的责任人及主要完成的工作。

参考答案：（要结合自身实际的工作经验进行回答，不同公司略有区别）

测试人员提交新的 **Bug** 入库，错误状态为 **New**。

高级测试员/测试经理验证错误，如果确认是错误，分配给开发组。设置状态为 **Open**。如果不是错误，则拒绝，设置为 **Declined** 状态。

开发经理分配 **bug** 至对应的模块开发人员。

开发人员查询状态为 **Open** 的 **Bug**，如果不是错误，则置状态为 **Declined**；如果是 **Bug** 则修复并置状态为 **Fixed**。不能解决的 **Bug**，要留下文字说明及保持 **Bug** 为 **Open** 状态。

对于不能解决和延期解决的 **Bug**，不能由开发人员自己决定，一般要通过某种会议（评审会）通过才能认可。

测试人员查询状态为 **Fixed** 的 **Bug**，然后验证 **Bug** 是否已解决，如解决，置 **Bug** 的状态为 **Closed**，如没有解决，置 **bug** 状态为 **Reopen**。

63、写出 bug 报告当中一些必备的内容。

参考答案：

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

64、开发人员老是犯一些低级错误怎么解决？

参考答案：

这种现象在开发流程不规范的团队里特别常见，尤其是一些“作坊式”的团队里。解决这种问题一般从两个方面入手：

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。

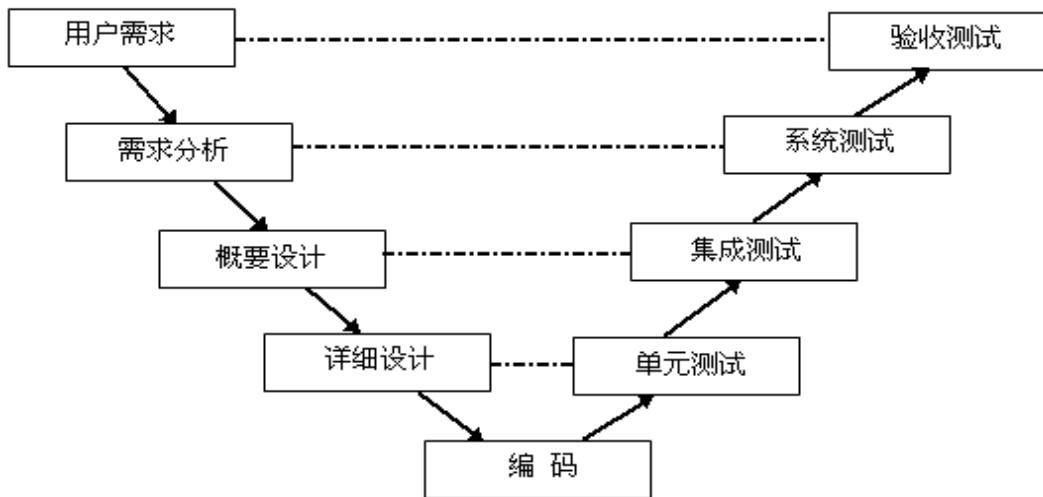
另一方面就是加强测试，具体做法就是加强开发人员的自己测试，把这些问题“消灭”在开发阶段，这是比较好的做法，读者可以参考第 13 章试案例分析的“13.1.2 缺陷反复出现，谁的责任”小节，13.1.2 专门讨论了这类问题的方法。

此外，还可以通过规范的缺陷管理来对开发人员进行控制，比如测试部门整理出常见的缺陷，让开发人员自己对照进行检查，以减少这类低级错误的发生。

开发人员犯错误是正常的现象，作为测试人员一定不能抱怨，要认认真真的解决问题才是上策。

65、画出软件测试的 V 模型图。

参考答案：



66、为什么要在一个团队中开展软件测试工作？

参考答案：

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

67、您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

参考答案：（根据项目经验不同，灵活回答即可）

我曾经做过 web 测试，后台测试，客户端软件，其中包括功能测试，性能测试，用户体验测试。最擅长的是功能测试

68、您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）

参考答案：

测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。是把测试对象看作一个黑盒子。利用黑盒

测试法进行动态测试时，需要测试软件产品的功能，不需测试软件产品的内部结构和处理过程。采用黑盒技术设计测试用例的方法有：等价类划分、边界值分析、错误推测、因果图和综合策略。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范（快捷键之类的），是否美观（能否吸引用户的注意力），是否安全（尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告）？做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的，然后再考虑该功能点的性能测试

69、您认为做好测试用例设计工作的关键是什么？

参考答案：

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

70、请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

参考答案：

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。

单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的

一致。

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起测试。

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。（常见的联调测试）

系统测试的目的是对最终软件系统全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

71、测试计划工作的目的是什么？测试计划工作的内容都包括什么？其中哪些是最重要的？

参考答案：

软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

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。所以其中最重要的是测试策略和测试方法（最好是能先评审）

72、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。

参考答案：

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.

73、请以您以往的实际工作为例,详细的描述一次测试用例设计的完整的过程。

参考答案:

就说最近的这次网站功能的测试吧

首先:得到相关文档(需求文档和设计文档),理解需求和设计思想后,想好测试策略(测试计划简单点就OK了),考虑到测试环境,测试用例,测试时间等问题。

第二步:设计测试用例,测试策略是:把网站部分的功能点测试完,然后在进行系统测试(另外个模块呢有另一个测试人员负责,可以进行联调测试),网站模块的测试基本是功能测试和界面测试(用户并发的可能性很小,所以不考虑):这次的网站的输入数据呢是使用数据库中的某张表记录,如果表中某一数据记录中新加进来的(还没有被处理的,有个标志位),网站启动后会立刻去刷那张表,得到多条数据,然后在进行处理.处理过程中,会经历3个步骤,网站才算完成了它的任务.有3个步骤呢,就可以分别对这3个步骤进行测试用例的设计,尽量覆盖到各种输入情况(包括数据库中的数据,用户的输入等),得出了差不多50个用例.界面测试,也就是用户看的到的地方,包括发送的邮件和用户填写资料的页面展示。

第三步:搭建测试环境(为什么这个时候考虑测试环境呢?因为我对网站环境已经很熟了,只有有机器能空于下来做该功能测试就可以做了),因为网站本身的环境搭建和其他的系统有点不同,它需要的测试环境比较麻烦,需要web服务器(Apache,tomcat),不过这次需求呢,网站部分只用到了tomcat,所以只要有tomcat即可

第四步:执行测试

74、您以往是否曾经从事过性能测试工作?如果有,请尽可能的详细描述您以往的性能测试工作的完整过程。

参考答案:(以自己最熟悉的性能测试项目为例)

是的,曾经做过网站方面的性能测试,虽然做的时间并不久(2个月吧),当时呢,是有位网站性能测试经验非常丰富的前辈带着我一起做。

性能测试类型包括负载测试,强度测试,容量测试等

负载测试:负载测试是一种性能测试指数据在超负荷环境中运行,程序是否能够承担。

强度测试:强度测试是一种性能测试,他在系统资源特别低的情况下软件系统运行情况

容量测试:确定系统可处理同时在线的最大用户数

在网站流量逐渐加大的情况下,开始考虑做性能测试了,首先要写好性能测试计划,根据运营数据得出流量最大的页面(如果是第一次的话,一般是首页,下载页,个人帐户页流量最大,而且以某种百分比),

Web 服务器指标指标:

* Avg Rps: 平均每秒钟响应次数=总请求时间/秒数;

* Successful Rounds: 成功的请求;

* Failed Rounds : 失败的请求;

* Successful Hits : 成功的点击次数;

* Failed Hits : 失败的点击次数;

* Hits Per Second : 每秒点击次数;

* Successful Hits Per Second : 每秒成功的点击次数;

* Failed Hits Per Second : 每秒失败的点击次数;

* Attempted Connections : 尝试链接数;

75、你对测试最大的兴趣在哪里?为什么?

参考答案:

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.曾经在无忧测试网上看到一篇文章,是关于如何做好一名测试工程师.一共罗列了11,12点,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 1, 2 点我没有把握, 其他点我都很有信心做好它。

刚开始进入测试行业时, 对测试的认识是从无忧测试网上了解到的一些资料, 当时是冲着做测试需要很多技能才能做的好, 虽然入门容易, 但做好很难, 比开发更难, 虽然当时我很想做开发(学校专业课我基本上不缺席, 因为我喜欢我的专业), 但看到测试比开发更难更有挑战性, 想做好测试的意志就更坚定了。

不到一年半的测试工作中, 当时的感动和热情没有减退一点(即使环境问题以及自身经验, 技术的不足, 做测试的你一定能理解)。

我觉得做测试整个过程中有 2 点让我觉得很有难度(对我来说, 有难度的东西我就非常感兴趣), 第一是测试用例的设计, 因为测试的精华就在测试用例的设计上了, 要在版本出来之前, 把用例写好, 用什么测试方法写?(也就是测试计划或测试策略), 如果你刚测试一个新任务时, 你得花一定的时间去消化业务需求和技术基础, 业务需求很好理解(多和产品经理和开发人员沟通就能达到目的), 而技术基础可就没那么简单了, 这需要你自觉的学习能力, 比如说网站吧, 最基本的技术知识你要知道网站内部是怎么运作的, 后台是怎么响应用户请求的? 测试环境如何搭建? 这些都需要最早的学好。至少在开始测试之前能做好基本的准备, 可能会遇到什么难题? 需求细节是不是没有确定好? 这些问题都能在设计用例的时候发现。

第二是发现 BUG 的时候了, 这应该是测试人员最基本的任务了, 一般按测试用例开始测试就能发现大部分的 bug, 还有一部分 bug 需要测试的过程中更了解所测版本的情况获得更多信息, 补充测试用例, 测试出 bug。还有如何发现 bug? 这就需要在测试用例有效的情况下, 通过细心和耐心去发现 bug 了, 每个用例都有可能发现 bug, 每个地方都有可能出错, 所以测试过程中思维要清晰(测试过程数据流及结果都得看仔细了, bug 都在里面发现的)。如何描述 bug 也很有讲究, bug 在什么情况下会产生, 如果条件变化一点点, 就不会有这个 bug, 以哪些最少的操作步骤就能重现这个 bug, 这个 bug 产生的规律是什么? 如果你够厉害的话, 可以帮开发人员初步定位问题。

76、你以前工作时的测试流程是什么?

参考答案:(灵活回答)

公司对测试流程没有规定如何做, 但每个测试人员都有自己的一套测试流程。我说下我 1 年来不断改正(自己总结, 吸取同行的方法)后的流程吧。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)。

77、当开发人员说不是 BUG 时, 你如何应付?

参考答案:

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? 如果被用户发现或出了问题, 会有什么不良结果? 程序员可能会给你很多理由, 你可以对他的解释进行反驳。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。

78、软件的构造号与版本号之间的区别? BVT(BuildVerificationTest)

参考答案: 版本控制命名格式: 主版本号.子版本号[.修正版本号[.编译版本号]]

Major.Minor [.Revision[.Build]]

应根据下面的约定使用这些部分:

Major: 具有相同名称但不同主版本号的程序集不可互换。例如, 这适用于对产品的大量重写, 这些重写使得无法实现向后兼容性。

Minor：如果两个程序集的名称和主版本号相同，而次版本号不同，这指示显著增强，但照顾到了向后兼容性。例如，这适用于产品的修正版或完全向后兼容的新版本。

Build：内部版本号的不同表示对相同源所作的重新编译。这适合于更改处理器、平台或编译器的情况。

Revision：名称、主版本号和次版本号都相同但修订号不同的程序集应是完全可互换的。这适用于修复以前发布的程序集中的安全漏洞。

BVT(Build Verification Test):

作为 **Build** 的一部分，主要是通过对基本功能、特别是关键功能的测试，保证新增代码没有导致功能失效，保证版本的持续稳定。实现 **BVT** 方式是有以下几种：**1**、测试人员手工验证关键功能实现的正确性。特点：这是传统开发方法中，通常采用的方式。无需维护测试脚本的成本，在测试人力资源充足，测试人员熟悉业务、并对系统操作熟练情况下效率很高，比较灵活快速。缺点：人力成本较高；对测试人员能力有一定要求；测试人员面对重复的工作，容易产生疲倦懈怠，从而影响测试质量。**2**、借助基于 **GUI** 的自动化功能测试工具来完成，将各基本功能操作录制成测试脚本，每次回放测试脚本验证功能实现的正确性。特点：能够模拟用户操作完成自动的测试，从 **UI** 入口到业务实现，每一层的代码实现都经过验证；节约人力成本；降低测试人员重复劳动的工作量，机器不会疲倦；缺点：对于 **UI** 变动比较频繁的系统来说，这种方式的维护成本很高，实施起来非常困难。另外，在项目周期较短且后续无延续性或继承的情况下，也不推荐使用此方式。**3**、由开发人员通过自动化测试工具完成业务层的 **BVT** 测试。特点：通过对业务层关键功能的持续集成测试，保证系统功能的持续稳定。可以结合 **DailyBuild**，做为 **Build** 的一部分，自动实现并输入 **BVT** 报告。缺点：仅对业务规则实现的正确性进行了测试，对表现层无法测试到，对于诸如：前台页面控件各种事件响应、页面元素变化等方面的问题无法保证。

79、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？

参考答案：

80、您以往所从事的软件测试工作中，是否使用了一些工具来进行软件缺陷（**Bug**）的管理？如果有，请结合该工具描述软件缺陷（**Bug**）跟踪管理的流程。

参考答案：

81、您认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？

参考答案：

82、单元测试、集成测试、系统测试的侧重点是什么？

参考答案：

83、集成测试通常都有那些策略？

参考答案：

84、一个缺陷测试报告的组成

参考答案：

85、基于 WEB 信息管理系统测试时应考虑的因素有哪些？

参考答案：

86、软件测试项目从什么时候开始，？为什么？

参考答案：

87、需求测试注意事项有哪些？

参考答案：

88、简述一下缺陷的生命周期

参考答案：

89、你在你所在的公司是怎么开展测试工作的？是如何组织的？

参考答案：

90、你认为理想的测试流程是什么样子？

参考答案：

91、您在从事性能测试工作时，是否使用过一些测试工具？如果有，请试述该工具的工作原理，并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。

参考答案：

92、软件测试活动的生命周期是什么？

参考答案：

93、请画出软件测试活动的流程图？

参考答案：

94、针对缺陷采取怎样管理措施？

参考答案：

95、什么是测试评估？测试评估的范围是什么？

参考答案：

96、如果能够执行完美的黑盒测试，还需要进行白盒测试吗？为什么？

参考答案：

97、测试结束的标准是什么？

参考答案：

98、软件验收测试除了 alpha , beta 测试以外, 还有哪一种？

参考答案：

99、做测试多久了？以前做过哪些项目？你们以前测试的流程是怎样的？用过哪些测试工具？

参考答案：

100、请就如何在开发中进行软件质量控制说说你的看法

参考答案：

101、一套完整的测试应该由哪些阶段组成？分别阐述一下各个阶段。

102、软件测试的类型有那些？分别比较这些不同的测试类型的区别与联系。

103、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

104、在分别测试 winform 的 C/S 结构与测试 WEB 结构的软件是，应该采取什么样的方法分别测试？他们存在什么样的区别与联系？

105、在测试 winform 的 C/S 结构软件时，发现这个软件的运行速度很慢，您会认为是什么原因？您会采取哪些方法去检查这个原因？

106、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

107、你都用什么测试方法

针对不同的产品或者系统或者模块，有不同的测试方法。总体而言有白盒测试和黑盒测试。

108、怎么编写案例

案例的编写与测试阶段的定义有很大的关系。系统测试和 unit 测试的案例可能不同。总体而言测试案例根据系统的需求而定。

109、怎么才能够全面的测试到每一个点

测试的全面性主要需要在设计测试计划的时候考虑，从测试策略，产品需求等等多个角度考虑从而定义全部的测试点。

110、谈谈软件测试技术，以及如何提高

111、谈谈软件测试职业发展，以及个人的打算

112、谈谈软件测试在企业的地位，也可以结合软件生命周期来谈

113、一般公司里实际的软件测试流程是什么样的？你们公司又是怎样的？

114、软件工程师要具有那些素质？

115、你会哪些测试工具？怎么操作？

116、你能不能说下你的 3 到 5 年的职业计划（规划）

117、你觉得你来应聘有那些优势？

其他问题：（有可能清晰的思路比确切的答案更重要）

对测试的理解——**考查点：**基本的测试知识，对测试是否认可

谈一谈过去自己的工作——**考查点**：了解经历、提供进一步提问的素材，表达能力、测试技能
测试设计的方法并举例说明——**考查点**：测试技术的使用
测试工具——**考查点**：熟悉程度，能否与当前工作匹配？
如何做计划？如何跟踪计划？——**考查点**：日常工作能力
如果开发人员提供的版本不满足测试的条件，如何做？——**考查点**：与开发人员协作的能力
熟悉 unix 系统、oracle 数据库吗？——**考查点**：是否具备系统知识
做过开发吗？写过哪些代码？——**考查点**：开发技能
阅读英语文章，给出理解说明？——**考查点**：部分英语能力
文档的意义——**考查点**：是否善于思考？（最简单的概念，不同层次的理解）
假如进入我们公司，对我们哪些方面会有帮助？——**考查点**：讲讲自己的特长
随便找一件物品，让其测试——**考查点**：测试的实际操作能力
有一个新的软件，假如你是测试工程师，该如何做——**考查点**：实际项目经验、是否有带领测试团队的经验和潜力

开发及环境搭建类面试题

1、描述软件产生内存泄露的原因以及检查方式。（可以结合一种开发语言进行描述）

参考答案：

内存泄露的原因，主要是由于开发过程当中申请了计算机资源（例如对象、内存等），但是使用资源完成以后没有及时释放资源导致的。例如在 C 语言当中使用了 malloc 申请了内存，但是未使用 free 来释放内存。

2、简述什么是值传递，什么是地址传递，两者区别是什么？

参考答案：

值传递主调函数传递给被调函数的是值的拷贝，不是原值；地址传递主调函数传递给被调函数的是值的地址。区别是值传递被调函数中的操作不改变主调函数的值，而地址传递则不同。

3、结构化程序设计和面向对象程序设计各自的特点及优缺点是什么？

参考答案：（不需要回答如此复杂）

结构化程序设计思想采用了模块分解与功能抽象和自顶向下、分而治之的方法，从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。它的重点在于把功能进行分解。但是由于在实际开发过程当中需求会经常发生变化，因此，它不能很好的适应需求变化的开发过程。结构化程序设计是面向过程的。

面向对象程序设计以需求当中的数据作为中心，来进行设计，具有良好的代码重用性。

封装性：也叫数据隐藏，用户无需知道内部工作流程，只要知道接口和操作就可以的，C++中一般用类来实现封装。

继承性：一种支持重用的思想，在现有的类型派生出新的子类，例如新型电视机在原有型号的电视机上增加若干中功能而得到，新型电视机是原有电视机的派生，继承了原有电视机的属性，并增加了新的功能。

多态性：指在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。

动态联编：指一个计算机程序自身彼此关联的过程，按照联编所进行的阶段不同，可分为两种不同的联编方法：静态联编和动态联编。

4、简述什么是存储过程和触发器？

参考答案：

存储过程：是数据库中的一个对象，**Transact-SQL** 语句的预编译集合，这些语句在一个名称下存储并作为一个单元进行处理。（可以理解为 C 语言中的函数，有参数、返回值等函数特性）

触发器是一种特殊类型的存储过程，当使用下面的一种或多种数据修改操作在指定表中对数据进行修改时，触发器会生效：**UPDATE**、**INSERT** 或 **DELETE**。

5、使用 C 语言编写一个函数，用于交换两个变量的值（地址传递）。

参考答案：

```
void Swap(int *a,int *b)
{
    int temp;
    int temp=*a;
    int *a=*b;
    int *b=temp;
}
```

6、请简述 DNS、活动目录、域的概念。

参考答案：

DNS：域名服务，作用是将网络域名解析成 **IP** 地址；

活动目录：微软提供的目录服务的一种，它存储有关网络上的对象信息，并使管理员和用户更方便的查找和使用这类信息；

域：网络系统的一个安全边界，在一个域当中，计算机和用户共享一些列的安全信息。

7、描述 TCP/IP 协议的层次结构，以及每一层中重要协议。

参考答案：（可以回答五层结构）

TCP/IP	协议
应用层/Application	HTTP 、 SMTP 、 FTP
传输层/Transport	TCP、UDP
网络层/Network	IP
链路层/Link	ARP、RARP

8、简述子网掩码的用途。

参考答案：

子网掩码主要用来判断两个 **IP** 地址是否处在同一个局域网当中；子网掩码是由连续的 **2** 进制 **1** 组成的。子网掩码和 **IP** 地址进行按位与运算后，结果一致，表示处于一个局域网当中，如果不一致，表示不再一个局域网当中，需要寻找路由。

9、说出 4 种以上常用的操作系统及其主要的应用范围（微软的操作系统除外）。

参考答案：

Linux (Red Hat、SUSE、Debian、Trubo Linux)：主要用于搭建各类服务器

MAC OS：苹果机的操作系统，用于图像处理

Unix (AIX)：IBM 服务器的专用操作系统；

Solaris：Sun 操作系统；**FreeBSD、NetBSD**)

10、在 Linux 系统中，一个文件的访问权限是 **755**，其含义是什么？

参考答案：

755 表示该文件所有者对该文件具有读、写、执行权限，该文件所有者所在组用户及其他用户对该文件

具有读和执行权限。

11、Windows 操作系统中 PATH 环境变量的作用是什么？

参考答案：

PATH 是 **Windows** 操作系统环境变量，**PATH** 作用是用户在命令行窗口执行一个命令，则在 **PATH** 变量设置的目录下依次寻找该命令或对应的执行文件，若找到，则执行，若没有找到，则命令行窗口返回无效命令。

12、Ghost 的主要用途和常用方法？

参考答案：

Ghost 是一个非常著名的硬盘克隆工具。该工具的主要作用是可以将一个硬盘或硬盘中的某个分区原封不动的复制到另一个硬盘或其他的分区中。如果你需要备份启动分区或者是需要在多台机器上安装相应的系统和应用程序，都可以通过 **Ghost** 来实现，相信通过这个工具备份，恢复速度和硬盘安装速度会成倍的提高。

Norton Ghost 有一个很大的特点，就是在克隆硬盘时不会改变任何文件信息，程序可以很好的支持 **FAT16**、**FAT32** 以及 **NTFS** 格式的文件分配结构（其中包括 **Windows 2000** 的文件分配格式），虽然是 **DOS** 环境下运行的程序，但工具可支持 **Win 9x** 的长文件名特性。

常用方法包括：硬盘克隆、分区克隆、硬盘或分区克隆成镜像文件等。

13、在 RedHat 中，从 root 用户切到 user1 用户，一般用什么命令？

参考答案：su

su user1 切换到 user1，但切换后的当前目录还是 root 访问的目录

su - user1 切换到 user1，并且当前目录切换到 user1 的根目录下（/home/user1/）

14、Linux 中，一般怎么隐藏文件？

参考答案：文件名以一个.开头

15、如何将自己的本地磁盘（D）做成 FTP 供远端主机使用？

参考答案：**Windows** 下安装 **FTP** 服务，并将 **FTP** 的根目录指向 **D** 盘即可。

16、对 RUP, CMM, CMMI, XP, PSP, TSP 的认识？

参考答案：软件过程标准：**CMMI**、**PSP**、**TSP**、**RUP**、软件工程规范国家标准；（**AP**、**XP**、**ASD** 等开发过程思想好像还不能称其为标准）

RUP (Rational Unified Process) 是 **Rational** 公司提出的一套开发过程模型，它是一个面向对象软件工程的通用业务流程。它描述了一系列相关的软件工程流程，它们具有相同的结构，即相同的流程构架。**RUP** 为在开发组织中分配任务和职责提供了一种规范方法，其目标是确保在可预计的时间安排和预算内开发出满足最终用户需求的高品质的软件。**RUP** 具有两个轴，一个轴是时间轴，这是动态的。另一个轴是 workflow 轴，这是静态的。在时间轴上，**RUP** 划分了四个阶段：初始阶段、细化阶段、构造阶段和发布阶段。每个阶段都使用了迭代的概念。在 workflow 轴上，**RUP** 设计了六个核心 workflow 和三个核心支撑 workflow，核心 workflow 轴包括：业务建模 workflow、需求 workflow、分析设计 workflow、实现 workflow、测试 workflow 和发布 workflow。核心支撑 workflow 包括：环境 workflow、项目管理工作流和配置与变更管理工作流。**RUP** 汇集现代软件开发中多方面的最佳经验，并为适应各种项目及组织的需要提供了灵活的形式。作为一个商业模型，它具有非常详细的过程指导和模板。但是由于该模型比较复杂，因此在模型的掌握上需要花费比较大的成本。尤其对项目管理者提出了比较高的要求。

CMM (Capability Maturity Model 能力成熟度模型) 由美国卡内基-梅隆大学的软件工程研究所（简称 **SEI**）受美国国防部委托，于 1991 年研究制定，初始的主要目的是为了评价美国国防部的软件合同承包组织的

能力，后因为在软件企业应用 **CMM** 模型实施过程改进取得较大的成功，所以在全世界范围内被广泛使用，**SEI** 同时建立了主任评估师评估制度，**CMM** 的评估方法为 **CBA-IPI**。**CMM** 的本质是软件管理工程的一个部分。它是对于软件组织在定义，实现，度量，控制和改善其软件过程的进程中各个发展阶段的描述。他通过 5 个不断进化的层次来评定软件生产的历史与现状：初始层是混沌的过程；可重复层是经过训练的软件过程；定义层是标准一致的软件过程；管理层是可预测的软件过程；优化层是能持续改善的软件过程。

CMM/PSP/TSP 即软件能力成熟度模型/个体软件过程/群组软件过程，是 1987 年美国 **Carnegie Mellon** 大学软件工程研究所(**CMU/SEI**)以 **W.S.Humphrey** 为首的研究组发表的研究成果“承制方软件工程能力的评估方法”。

CMMI 是 **SEI** 于 2000 年发布的 **CMM** 的新版本。**CMMI** 不但包括了软件开发过程改进，还包含系统集成、软硬件采购等方面的过程改进内容。

CMMI 纠正了 **CMM** 存在的一些缺点，使其更加适用企业的过程改进实施。**CMMI** 适用 **SCAMPI** 评估方法。需要注意的是，**SEI** 没有废除 **CMM** 模型，只是停止了 **CMM** 评估方法：**CBA-IPI**。现在如要进行 **CMM** 评估，需使用 **SCAMPI** 方法。但 **CMMI** 模型最终代替 **CMM** 模型的趋势不可避免。

XP（极限编程）规定了一组核心价值和方法，可以让软件开发人员发挥他们的专长：编写代码。**XP** 消除了大多数重量型过程的不必要产物，通过减慢开发速度、耗费开发人员的精力（例如干特图、状态报告，以及多卷需求文档）从目标偏离。

XP 的核心价值：交流、简单、反馈、勇气。

17、DNS 是什么,它是如何工作的?

参考答案：域名解析服务。用于将域名解析为 **IP**，或反和将 **IP** 解析为域名。

客户机可指定 **DNS** 服务器来解析，或用本机 **hosts** 文件进行解析。

Windows 下配置 **DNS** 服务器在《搭建 **Windows** 测试环境》中有。

18、防火墙如何保证安全的? 主要有哪些?

参考答案：防火墙分类 1

从防火墙的软、硬件形式来分的话，防火墙可以分为软件防火墙和硬件防火墙以及芯片级防火墙。

第一种：软件防火墙

软件防火墙运行于特定的计算机上，它需要客户预先安装好的计算机操作系统的支持，一般来说这台计算机就是整个网络的网关。俗称“个人防火墙”。软件防火墙就像其它的软件产品一样需要先在计算机上安装并做好配置才可以使用。防火墙厂商中做网络版软件防火墙最出名的莫过于 **Checkpoint**。使用这类防火墙，需要网管对所工作的操作系统平台比较熟悉。

第二种：硬件防火墙

这里说的硬件防火墙是指“所谓的硬件防火墙”。之所以加上“所谓”二字是针对芯片级防火墙说的了。它们最大的差别在于是否基于专用的硬件平台。目前市场上大多数防火墙都是这种所谓的硬件防火墙，他们都基于 **PC** 架构，就是说，它们和普通的家庭用的 **PC** 没有太大区别。在这些 **PC** 架构计算机上运行一些经过裁剪和简化的操作系统，最常用的有老版本的 **Unix**、**Linux** 和 **FreeBSD** 系统。值得注意的是，由于此类防火墙采用的依然是别人的内核，因此依然会受到 **OS**（操作系统）本身的安全性影响。

传统硬件防火墙一般至少应具备三个端口，分别接内网，外网和 **DMZ** 区（非军事化区），现在一些新的硬件防火墙往往扩展了端口，常见四端口防火墙一般将第四个端口做为配置口、管理端口。很多防火墙还可以进一步扩展端口数目。

第三种：芯片级防火墙

芯片级防火墙基于专门的硬件平台，没有操作系统。专有的 **ASIC** 芯片促使它们比其他种类的防火墙速度更快，处理能力更强，性能更高。做这类防火墙最出名的厂商有 **NetScreen**、**FortiNet**、**Cisco** 等。这类防火墙由于是专用 **OS**（操作系统），因此防火墙本身的漏洞比较少，不过价格相对比较高昂。

防火墙技术虽然出现了许多，但总体来讲可分为“包过滤型”和“应用代理型”两大类。前者以以色列的 **Checkpoint** 防火墙和美国 **Cisco** 公司的 **PIX** 防火墙为代表，后者以美国 **NAI** 公司的 **Gauntlet** 防火墙为代表。

(1). 包过滤(Packet filtering)型

包过滤型防火墙工作在 **OSI** 网络参考模型的网络层和传输层，它根据数据包源地址，目的地址、端口号和协议类型等标志确定是否允许通过。只有满足过滤条件的数据包才被转发到相应的目的地，其余数据包则被从数据流中丢弃。

包过滤方式是一种通用、廉价和有效的安全手段。之所以通用，是因为它不是针对各个具体的网络服务采取特殊的处理方式，适用于所有网络服务；之所以廉价，是因为大多数路由器都提供数据包过滤功能，所以这类防火墙多数是由路由器集成的；之所以有效，是因为它能很大程度上满足了绝大多数企业安全要求。

在整个防火墙技术的发展过程中，包过滤技术出现了两种不同版本，称为“第一代静态包过滤”和“第二代动态包过滤”。

●第一代静态包过滤类型防火墙

这类防火墙几乎是与路由器同时产生的，它是根据定义好的过滤规则审查每个数据包，以便确定其是否与某一条包过滤规则匹配。过滤规则基于数据包的报头信息进行制订。报头信息中包括 **IP** 源地址、**IP** 目标地址、传输协议(**TCP**、**UDP**、**ICMP** 等等)、**TCP/UDP** 目标端口、**ICMP** 消息类型等。

●第二代动态包过滤类型防火墙

这类防火墙采用动态设置包过滤规则的方法，避免了静态包过滤所具有的问题。这种技术后来发展成为包状态监测(**Stateful Inspection**)技术。采用这种技术的防火墙对通过其建立的每一个连接都进行跟踪，并且根据需要可动态地在过滤规则中增加或更新条目。

包过滤方式的优点是不用改动客户机和主机上的应用程序，因为它工作在网络层和传输层，与应用层无关。但其弱点也是明显的：过滤判别的依据只是网络层和传输层的有限信息，因而各种安全要求不可能充分满足；在许多过滤器中，过滤规则的数目是有限制的，且随着规则数目的增加，性能会受到很大地影响；由于缺少上下文关联信息，不能有效地过滤如 **UDP**、**RPC**（远程过程调用）一类的协议；另外，大多数过滤器中缺少审计和报警机制，它只能依据包头信息，而不能对用户身份进行验证，很容易受到“地址欺骗型”攻击。对安全管理人员素质要求高，建立安全规则时，必须对协议本身及其在不同应用程序中的作用有较深入的理解。因此，过滤器通常是和应用网关配合使用，共同组成防火墙系统。

(2). 应用代理(**Application Proxy**)型

应用代理型防火墙是工作在 **OSI** 的最高层，即应用层。其特点是完全“阻隔”了网络通信流，通过对每种应用服务编制专门的代理程序，实现监视和控制应用层通信流的作用。其典型网络结构如图所示。

在代理型防火墙技术的发展过程中，它也经历了两个不同的版本，即：第一代应用网关型代理防火和第二代自适应代理防火墙。

第一代应用网关(**Application Gateway**)型防火墙

这类防火墙是通过一种代理(**Proxy**)技术参与到一个 **TCP** 连接的全过程。从内部发出的数据包经过这样的防火墙处理后，就好像是源于防火墙外部网卡一样，从而达到隐藏内部网结构的作用。这种类型的防火墙被网络安全专家和媒体公认为是最安全的防火墙。它的核心技术就是代理服务器技术。

第二代自适应代理(**Adaptive proxy**)型防火墙

它是近几年才得到广泛应用的一种新防火墙类型。它可以结合代理类型防火墙的安全性和包过滤防火墙的高速度等优点，在毫不损失安全性的基础之上将代理型防火墙的性能提高 **10** 倍以上。组成这种类型防火墙的基本要素有两个：自适应代理服务器(**Adaptive Proxy Server**)与动态包过滤器(**Dynamic Packet filter**)。

在“自适应代理服务器”与“动态包过滤器”之间存在一个控制通道。在对防火墙进行配置时，用户仅仅将所需要的服务类型、安全级别等信息通过相应 **Proxy** 的管理界面进行设置就可以了。然后，自适应代理可以根据用户的配置信息，决定是使用代理服务从应用层代理请求还是从网络层转发包。如果是后者，它将动态地通知包过滤器增减过滤规则，满足用户对速度和安全性的重要要求。

代理类型防火墙的最突出的优点就是安全。由于它工作于最高层，所以它可以对网络中任何一层数据通信进行筛选保护，而不是像包过滤那样，只是对网络层的数据进行过滤。

另外代理型防火墙采取是一种代理机制，它可以为每一种应用服务建立一个专门的代理，所以内外部网络之间的通信不是直接的，而都需先经过代理服务器审核，通过后再由代理服务器代为连接，根本没有给内、外部网络计算机任何直接会话的机会，从而避免了入侵者使用数据驱动类型的攻击方式入侵内部网。

代理防火墙的最大缺点就是速度相对比较慢，当用户对内外部网络网关的吞吐量要求比较高时，代理防火墙就会成为内外部网络之间的瓶颈。那因为防火墙需要为不同的网络服务建立专门的代理服务，在自己的代理程序为内、外部网络用户建立连接时需要时间，所以给系统性能带来了一些负面影响，但通常不会很明显。

防火墙分类 3

从防火墙结构上分，防火墙主要有：单一主机防火墙、路由器集成式防火墙和分布式防火墙三种。

单一主机防火墙是最为传统的防火墙，独立于其它网络设备，它位于网络边界。

这种防火墙其实与一台计算机结构差不多（如下图），同样包括 CPU、内存、硬盘等基本组件，当然主板更是不能少了，且主板上也有南、北桥芯片。它与一般计算机最主要的区别就是一般防火墙都集成了两个以上的以太网卡，因为它需要连接一个以上的内、外部网络。其中的硬盘就是用来存储防火墙所用的基本程序，如包过滤程序和代理服务程序等，有的防火墙还把日志记录也记录在此硬盘上。虽然如此，但我们不能说它就与我们平常的 PC 机一样，因为它的工作性质，决定了它要具备非常高的稳定性、实用性，具备非常高的系统吞吐性能。正因如此，看似与 PC 机差不多的配置，价格甚远。

随着防火墙技术的发展及应用需求的提高，原来作为单一主机的防火墙现在已发生了许多变化。最明显的变化就是现在许多中、高档的路由器中已集成了防火墙功能，还有的防火墙已不再是一个独立的硬件实体，而是由多个软、硬件组成的系统，这种防火墙，俗称“分布式防火墙”。

原来单一主机的防火墙由于价格非常昂贵，仅有少数大型企业才能承受得起，为了降低企业网络投资，现在许多中、高档路由器中集成了防火墙功能。如 Cisco IOS 防火墙系列。但这种防火墙通常是较低级的包过滤型。这样企业就不用再同时购买路由器和防火墙，大大降低了网络设备购买成本。

分布式防火墙再也不是只是位于网络边界，而是渗透于网络的每一台主机，对整个内部网络的主机实施保护。在网络服务器中，通常会安装一个用于防火墙系统管理软件，在服务器及各主机上安装有集成网卡功能的 PCI 防火墙卡，这样一块防火墙卡同时兼有网卡和防火墙的双重功能。这样一个防火墙系统就可以彻底保护内部网络。各主机把任何其它主机发送的通信连接都视为“不可信”的，都需要严格过滤。而不是传统边界防火墙那样，仅对外部网络发出的通信请求“不信任”。

防火墙分类 4

如果按防火墙的应用部署位置分，可以分为边界防火墙、个人防火墙和混合防火墙三大类。

边界防火墙是最为传统的那种，它们于内、外部网络的边界，所起的作用的对内、外部网络实施隔离，保护边界内部网络。这类防火墙一般都是硬件类型的，价格较贵，性能较好。

个人防火墙安装于单台主机中，防护的也只是单台主机。这类防火墙应用于广大的个人用户，通常为软件防火墙，价格最便宜，性能也最差。

混合式防火墙可以说就是“分布式防火墙”或者“嵌入式防火墙”，它是一整套防火墙系统，由若干个软、硬件组件组成，分布于内、外部网络边界和内部各主机之间，既对内、外部网络之间通信进行过滤，又对网络内部各主机间的通信进行过滤。它属于最新的防火墙技术之一，性能最好，价格也最贵。

防火墙分类 5

如果按防火墙的性能来分可以分为百兆级防火墙和千兆级防火墙两类。

因为防火墙通常位于网络边界，所以不可能只是十兆级的。这主要是指防火的通道带宽 (Bandwidth)，或者说吞吐率。当然通道带宽越宽，性能越高，这样的防火墙因包过滤或应用代理所产生的延时也越小，对整个网络通信性能的影响也就越小。

19、目前流行的操作的系统有哪些？请举例说明安装操作系统的注意事项？

参考答案：MS Windows 系列：win 98、windows 2000 系列、win XP、win 2003 Server、win Vista 等等。

UNIX 类：SVR_x、FreeBSD、OpenBSD、NetBSD、Solaris、各种 Linux 等等。Mac OS……

多重引导时，一般先安装 win 操作系统，从低版本到高，再安装 Linux

20、简述一下 c/s 模式或者 b/s 模式？

参考答案：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，是随着 Internet 技术的兴起，对 C/S 结构的一种变化或者改进的结构。在这种结构下，用户界面完全通过 WWW 浏览器实现，一部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现，形成所谓 3-tier 结构。B/S 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。

21、TCP/UDP 有哪些区别？

参考答案：TCP-有连接,所以握手过程会消耗资源,过程为可靠连接,不会丢失数据,适合大数据量交换
UDP-非可靠连接,会丢包,没有校验,速度快,无须握手过程

	TCP	UDP
是否连接	面向连接	面向非连接
传输可靠性	可靠的	不可靠的
应用场合	传输大量数据	少量数据
速度	慢	快

22、ISO 模型？HUB、tch、Router 是 ISO 的第几层设备？

参考答案：从底向上：物理层、数据链路层、网络层、传输层、会话层、表示层和应用层
HUB：1层（物理层）；Switch：2层（数据链路层）；Router：3层（网络层）

23、内存有哪几种存储组织结构.请分别加以说明？

参考答案：

人力资源面试题

1、你的测试职业发展是什么？你自认为做测试的优势在哪里？

参考答案：

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累积的，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年累积测试经验，按如何做好测试工程师的要求自己，不断的更新自己改正自己，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、你为什么想离开目前的职务？

参考答案：

3、你对我们公司了解有多少？

参考答案：

4、你找工作时，最重要的考虑因素为何？

参考答案：工作的性质和内容是否能让我发挥所长，并不断成长。

5、为什么我们应该录取你？

参考答案：您可以由我过去的工作表现所呈现的客观数据，明显地看出我全力以赴的工作态度。

6、请谈谈你个人的最大特色。

参考答案：我的坚持度很高，事情没有做到一个令人满意的结果，绝不罢手。

7、一个测试工程师应具备那些素质和技能？

参考答案：

8、您认为在测试人员同开发人员的沟通过程中，如何提高沟通的效率和改善沟通的效果？维持测试人员同开发团队中其他成员良好的人际关系的关键是什么？

参考答案：

9、在您以往的测试工作中，最让您感到不满意或者不堪回首的事情是什么？您是如何来对待这些事情的？

参考答案：

10、在即将完成这次笔试前，您是否愿意谈一些自己在以往的学习和工作中获得的工作经验和心得体会？（可以包括软件测试、过程改进、软件开发或者与此无关的其他方面）

参考答案：

11、为什么选择测试这行？

参考答案：

它是一个新兴的行业，有发展潜力，而且很锻炼人，需要掌握更多的技能，比做开发要更难

为什么值得他们公司雇用？如果我雇用你，你能给部门带来什么贡献？

如果明知这样做不对，你还会依主管的指过去做吗

如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理

你觉得什么样的人最难相处

为什么值得他们公司雇用？

帮助公司提高软件质量和测试部门的技术水平

如果我雇用你，你能给部门带来什么贡献？

分享我的测试经验和测试技能，提高测试部门技术水平

如何从工作中看出你是个自动自觉的人

自动自觉范围太广

1. 工作成果

2. 工作质量

12、你的工作通常能在时限内完成吗。（我想问一下就是她问这个问题的动机是什么）

参考答案：

在有足够的资源和合理的工作量的情况下，完全可以按时完成，并能比一般人做的更好

13、通常你对于别人批评你会有什么样的反应

参考答案：有错即改，无错勉之

14、如果明知这样做不对，你还会依主管的指过去做吗？

参考答案：

15、如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理？

参考答案：

弄清楚客户为什么抱怨？是怎么样的问题？

如果是客服问题，提交客服部门解决
如果是质量问题，分析原因，下一版本改进

16、请就软件测试人员应该具备什么样的基本素质说说你的看法。

参考答案：

17、你在五年内的个人目标和职业目标分别是什么？

参考答案：

分析这个问题是用来了解你的计划能力的，通过这个问题，面试官同时还可以知道你的目标是否符合企业对你的安排。

错误回答我想在将来的某个时候考虑这个问题。如今企业的领导者更换频繁，我认为做太多的个人计划是荒谬可笑的，不是吗？

评论这种回答属于令人反感的一类。首先，当有人想了解你的目标时，“将来的某个时候”这种通俗说法并不奏效。其次，认为企业很脆弱，领导者更换频繁，这种说法毫无疑问会令人反感，而且也是不合理的。最后，认为做计划可笑，看不起这个问题，而且反问面试官，这些都注定了这样的求职者最终会失败。

正确回答从现在起的五年之内，我希望能够在很好的职位上待几年，而且最好有一次晋升，然后就期待着下一步。不管是向上提升，还是在企业内横向调动，对我个人来说，我希望找到一家企业——一家愿意做相互投入的企业——待上一段时间。

评论这个问题没有回答得过分具体（那样可能会产生漏洞），而且它表明你有雄心，并且思考过在企业中的成长方式。通过表达横向调动和向上提升的愿望，表明你是一个有灵活性的人。

18、你怎样做出自己的职业选择？

参考答案：

分析 面试官提出这个问题是为了了解求职者的动机，看看他（她）应聘这份工作是否有什么历史渊源，是否有职业规划，是不是仅仅在漫无目的地申请很多工作。

错误回答 我一直都想在企业界工作。自孩提时代起，我就梦想自己至少也要成为大企业的副总裁。

评论 除了难以令人相信之外，这种回答还存在一个问题：它表明求职者会对副总裁以下的职位不感兴趣。

正确回答 在上大学四年级前的那个夏天，我决定集中精力在某一领域谋求发展。尽管我是学商业的，但是我不知道自己最终会从事哪一行行业的工作。我花了一定的时间考虑自己的目标，想清楚了自己擅长做的事情以及想从工作中得到的东西，最后我得出了一个坚定的结论，那就是这个行业是最适合我的。

评论 这种回答表明，求职者认真地做过一些计划，缩小了自己的关注点，而且也认准了前进的方向。这种回答还表明，求职者理解个人职业规划的重要性，并且有能力做出认真的个人决策。