软件测试复习题

##### 1.在软件测试过程中，应该遵循的原则

完全测试程序是不可能的

软件测试是有风险的

测试无法显示隐藏的软件缺陷

存在的故障数量与发现的故障数成正比

杀虫剂现象

并非所有软件故障都能修复

一般不要丢弃测试用例

应避免测试自己编写的程序

软件测试是一项复杂且具有创造性和需要高度智慧的挑战性任务

##### 2.测试用例的设计

编写测试用例的唯一标准是用户需求，参考资料是需求规格说明书

设计原则

正确性、全面性、整体连贯性、可维护性、测试结果可判定性和可再现性

设计方法

等价类划分法、边界值分析法、基本路径分析法、因果图法

##### 3.测试用例的原则

测试用例最小化原则

测试用例替代产品文档功能的原则

单次投入成本和多次投入成本原则

测试结果分析和调试最简单化原则

##### 4.常用到的软件质量模型

V模型

W模型

H模型

X模型

前置测试模型

##### 5.软件测试计划

软件测试计划是描述了要进行的测试活动的范围、方法、资源和进度的文档，它确定了测试项、被测特性、测试任务、谁执行任务、各种可能的风险，通常作为关于质量的重要文档呈现给管理层。

确定测试范围

制定测试策略

测试资源安排

进度安排

风险和对策

##### 6.制定软件测试的计划的原则

（只找到目标）

（1）为测试各项活动制定一个现实可行的、综合的计划、包括每项测试活动的对象、范围、方法、进度和预期结果。

（2）为项目实施建立一个组织模型，并定义测试项目中每个角色的职责和工作内容。

（3）开发有效的测试模型，能正确地验证正在开发的软件系统。

（4）确定测试所需要的时间和资源，以确保其可获得性、有效性。

（5）确立每个测试阶段测试完成及测试成功的标准、要实现的目标。

（6）识别出测试活动中各种风险，并消除可能存在的风险，降低由不可能消除的风险所带来的损失。

##### 7.制定软件测试的技术的步骤

测试需求的分析和确定

测试计划

测试设计

测试执行

测试记录和缺陷跟踪

回归测试

测试总结报告

##### 8.静态测试、动态测试

静态测试是指不运行被测程序本身，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。其被测对象是各种与软件相关的有必要进行测试的产物，是对需求规格说明书、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。

动态测试实际运行被测程序，输入相应的测试数据，检查实际输出结果和预期结果是否相符的过程

##### 9.白盒测试、黑盒测试以及二者的关系

黑盒测试是一种从软件外部对软件实施的测试，

白盒测试是路径测试，将盒子打开观察程序内部的测试。

##### 10.软件测试与软件开发的过程的关系

软件测试和软件开发都是软件过程中的重要活动，是软件生命周期中重要的组成部分，应该是贯穿于整个软件开发生命周期的，测试的尽早介入是软件测试的一个基本准则，而不应仅仅将其看作软件开发中的一个阶段。

软件开发是一个自顶向下、逐步细化的过程，而测试过程是依相反顺序自底向上、逐步集成的过程。

开发过程中，通过持续的测试可以对产品质量提供持续的、快速的反馈，从而在整个开发过程中不断地、及时地改进产品的质量，并减少各种返工，降低软件开发的成本。

##### 11.白盒测试的覆盖准则

ESTCA

A rel B ，AB为变量，<>=都要出现一次

A rel C， A为变量，C为常量，当rel为＞时，需要适当选择A，使A=C+M（M是最小单位，当A为整型时，M=1）；当rel为<时，需要适当选择A，使A=C+M

对外部输入变量赋值，使其再每一测试用例中均有不同的值与符号，并与同一组测试用例中其他变量的值与符号不一致

LCSAJ

语句覆盖，分支覆盖，LCSAJ覆盖(程序中每个LCSAJ都至少在测试中经历过一次)，两两LCSAJ覆盖，…，所有首尾相连的LCSAJ覆盖都需要经历一次

##### 12. 白盒测试的常用工具，各适用于什么

Jtest 集成的，适合单元测试

Jcontract 独立的，适合系统级测试

C++ test 动态的自动化测试

CodeWizard 静态的代码规范测试

Insure++ 内存错误和内存泄漏的检测工具

##### 13.单元测试

**单元测试的原则**

**单元测试的重要性及目的**

**单元测试主要测试问题**

单元测试又称模块测试，针对软件设计中的最小单位—程序模块，进行正确性检查的测试工作。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

原则：自动化，独立性，可重复

##### 14.插桩程序设计

一种基本的动态测试方法，向源程序中添加一些语句实现对程序代码的执行、变量的变化等情况的检查，可以获得程序的控制流和数据流信息。如果我们想要了解一个程序在某次运行中可执行语句被覆盖的情况，或是每个语句的实际执行次数，最好的办法就是利用插装技术，它在软件测试技术上占有非常高的地位。最简单的插装：在程序中插入打印语句printf(“ ...”)语句。

##### 15.集成测试：

**集成测试的主要任务**

**集成测试与单元测试，系统测试的区别**

**集成测试的内容**

**集成测试的方法**

集成测试又叫组装测试，通常在单元测试的基础上，将所有程序模块进行有序地、递增的测试。重点测试不同模块的接口部分

##### 16.系统测试

**系统测试与用户测试的区别**

**系统测试的主要内容**

**常见的系统测试方法**

系统测试指的是将整个软件系统看作一个整体进行测试，包括对功能、性能、以及软件所运行的软硬件环境进行测试。

##### 17.容量测试与压力测试的区别

容量测试的目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值，压力测试则是在强负载下查看应用系统在峰值使用情况下操作行为，从而有效地发现系统的某项功能隐患、系统是否具有良好的容错能力和恢复能力。

压力测试是在给系统不断加压，增加并发量，直到崩溃，找到系统所能承受的极限值；

容量测试是在预先分析的极限值下，系统能否正常运行。

一般容量测试大多是正向测试，即是合法条件下的，而压力测试有时候是极端异常的，会导致错误的，倾向于反向测试，或者找到临界点。

##### 18.验收测试

**验收测试的主要内容**

按照项目任务书或合同、供需双方约定的验收依据文档进行的对整个系统的测试和评审，决定是否接收系统。在系统测试的后期，以用户测试为主或有测试人员等质量保证人员共同参与的测试

##### 19.α 测试和β测试的不同

1. α测试

就是把用户请到公司内部进行测试使用。

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测试；

目的：是评价软件产品的FLURPS(即功能、局域化、可使用性、可靠性、性能和支持)。

注意！α测试不能由程序员或测试员完成。

2. β测试

用户在不同场所进行测试。

β测试是一种验收测试。β测试由软件的最终用户们在一个或多个场所进行。

3.区别

它们都是验收测试！

α测试是指把用户请到开发方的场所来测试

β测试是指在一个或多个用户的场所进行的测试。

α测试的环境是受开发方控制的,用户的数量相对比较少,时间比较集中。

β测试的环境是不受开发方控制的, 用户数量相对比较多,时间不集中。

α测试先于β测试执行。通用的软件产品需要较大规模的β测试,测试周期比较长

##### 20.如何组织软件测试团队

##### 21.软件测试人员的培养方法

职业素质

责任心、沟通能力、团队合作精神、耐心，细心和信心、时时刻刻保持怀疑态度，并且有缺陷预防意识、不断学习能力

职业技能

业务知识、产品设计知识、软件架构知识、统一建模语言UML、测试工具使用、开发工具使用、用户心理学、界面设计的三种模型（设计者模型、实现者模型、用户模型）、人机交互认知心理学、编程能力、脚本语言、文档能力

##### 22.文档测试主要测试内容

主要测试用户手册和需求说明是否真正符合用户的实际需求。

##### 23.软件测试

软件测试是通过考虑软件的所有属性(可靠性，可伸缩性，可移植性，可重用性，可用性)和评估软件组件的执行来查找软件错误或缺陷来识别软件正确性的过程

##### 24.常见的黑盒测试用例的设计方法?并分别简单介绍一下各自的思想。

等价类方法

边界值方法

因果图法

正交表法

##### 25.恢复测试

恢复测试主要检查系统的容错能力。当系统出错时，能否在指定时间间隔内修正错误并重新启动系统。恢复测试首先要采用各种办法强迫系统失败，然后验证系统是否能尽快恢复。对于自动恢复需验证重新初始化（reinitialization）、检查点(checkpointing mechanisms)、数据恢复(data recovery)和重新启动 (restart)等机制的正确性；对于人工干预的恢复系统，还需估测平均修复时间，确定其是否在可接受的范围内。

##### 26.强度测试

使软件在其设计能力的极限状态下，以及超过此极限下运行，检验软件对异常情况的抵抗能力

##### 27.正确性测试

正确性测试检查软件的功能是否符合规格说明。

##### 28.Web测试

Web测试是对web系统进行的全面的测试

大致分为用户界面测试、功能测试、性能测试、兼容性测试、安全性测试

##### 29.条件组合覆盖

条件组合覆盖是通过设计足够多的测试用例，使得运行这些测试用例后，要使每个判断中各种条件的可能组合情况取值至少满足一次

##### 30.软件可靠性

特定环境下，在给定时间内，无障碍运行的概率

度量

软件中初始故障的数量

* 软件经过测试后，通过查错，改错，在软件中剩余故障的数量
* 平均无故障时间
* 故障间隔的时间长度
* 故障发生率
* 经过预测下次故障的发生时间

##### 31.软件缺陷

软件缺陷常常被叫bug。所谓软件缺陷，即计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或隐藏的功能缺陷。

一般看来，满足以下的任意一种情况都可以称为软件缺陷：

（1）软件未达到产品说明书中标明的功能；

（2）软件出现了产品说明书中指明不会出现的功能；

（3）软件功能超出了产品说明书中指明的范围；

（4）软件未达到产品说明书中指明应达到的目的；

（5）软件难以理解和使用、运行速度慢，或最终用户认为不好。

软件缺陷产生的原因：

需求不明确

软件结构复杂

编码问题

项目期限短

使用新技术

软件缺陷生命周期：见教材p9

##### 32. 测试用例

测试用例是为某个特殊目标编制的一组测试输入、执行条件，以及预期结果，以便于测试某个程序路径或核实是否满足特定需求。

##### 33. 变异测试

变异测试也称为“变异分析”，是一种对测试数据集的有效性、充分性进行评估的技术，能为研发人员开展需求设计、单元测试、集成测试提供有效的帮助

变异测试通过对比源程序与变异程序在执行同一测试用例时差异来评价测试用例集的错误检测能力

在变异测试过程中，一般利用与源程序差异极小的简单变异体来模拟程序中可能存在的各种缺陷

##### 34. 回归测试

是指软件被修改后重新进行的测试，如重复执行上一个版本测试时的用例，是为了保证对软件所做的修改没有引入新的错误而重复进行的测试

##### 35. 兼容性测试

兼容性测试是指要测试的软件在不同的硬件平台上、不同的应用软件之间、不同的操作系统中、不同的网络环境中是否可以正常的运行、有无异常的测试过程。

##### 36. 第三方测试

第三方软件测试概念

模式一：以客户为主导，为确认软件质量，寻求第三方测试团队来检验质量，开发团队和第三方联系不紧密

模式二：以软件开发团队为主导，开发团队和第三方联系紧密

定义 ：由开发者和用户以外的第三方进行的软件测试，其目的是为了保证测试的客观性

狭义上：独立的第三方测试机构，如国家级软件测试中心，各省软件评测中心，有资质的软件评测中心

广义上：非本软件的开发人员，QA部门人员测试，公司内部交叉测试

##### 37. 冒烟测试

是指在对一个新版本进行系统大规模的测试之前，先验证一下软件的基本功能是否实现，是否具备可测试性

##### 38. 确认测试

确认测试又称有效性测试。有效性测试是在模拟的环境下，运用黑盒测试的方法，验证被测软件是否满足需求规格说明书列出的需求。任务是验证软件的功能和性能及其他特性是否与用户的要求一致。对软件的功能和性能要求在软件需求规格说明书中已经明确规定，它包含的信息就是软件确认测试的基础。

##### 39. 性能测试

定义：测试软件是否达到需求规格说明书中规定的各项性能指标，并满足相关的约束和限制。

是软件测试的高端领域，性能测试工程师的待遇和白盒测试工程师不相上下，通常我们所说的高级软件工程师一般就是指性能测试或是白盒测试工程师

时间性能

空间性能

一般性能测试

可靠性测试

负载测试

压力测试

##### 40. 压力测试

定义：对系统不断施加压力的测试，通过确定一个系统的瓶颈或者不能接收的性能点，获得系统能提供的最大服务级别的测试。

给软件不断加压，强制其在极限的情况下运行，观察它可以运行到何种程度，从而发现性能缺陷，是通过搭建与实际环境相似的测试环境，通过测试程序在同一时间内或某一段时间内，向系统发送预期数量的交易请求、测试系统在不同压力情况下的效率状况，以及系统可以承受的压力情况。然后做针对性的测试与分析，找到影响系统性能的瓶颈，评估系统在实际使用环境下的效率情况，评价系统性能以及判断是否需要对应用系统进行优化处理或结构调整。并对系统资源进行优化。

##### 41. 负载测试

负载测试（Load testing），不限制软件的运行资源，测试软件的数据吞吐量上限，以发现设计上的错误或验证系统的负载能力。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。

负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。此外，负载测试还要评估性能特征。例如，响应时间、事务处理速率和其他与时间相关的方面。

##### 42. 安全测试

安全性测试是指有关验证应用程序的安全等级和识别潜在安全性缺陷的过程。应用程序级安全测试的主要目的是查找软件自身程序设计中存在的安全隐患，并检查应用程序对非法侵入的防范能力，根据安全指标不同测试策略也不同。

目前有许多种的测试手段可以进行安全性测试，安全测试方法分主要为三种：

　　①静态的代码安全测试：主要通过对源代码进行安全扫描，根据程序中数据流、控制流、语义等信息与其特有软件安全规则库进行匹对，从中找出代码中潜在的安 全漏洞。静态的源代码安全测试是非常有用的方法，它可以在编码阶段找出所有可能存在安全风险的代码，这样开发人员可以在早期解决潜在的安全问题。而正因为如此，静态代码测试比较适用于早期的代码开发阶段，而不是测试阶段。

　　②动态的渗透测试：渗透测试也是常用的安全测试方法。是使用自动化工具或者人工的方法模拟黑客的输入，对应用系统进行攻击性测试，从中找出运行时刻所存在的安全漏洞。这种测试的特点是真实有效，一般找出来的问题都是正确的，也是较为严重的。但渗透测试一个致命的缺点是模拟的测试数据只能到达有限的测试点，覆盖率很低。

③程序数据扫描。一个有高安全性需求的软件， 在运行过程中数据是不能遭到破坏的，否则会导致缓冲区溢出类型的攻击。数据扫描的手段通常是进行内存测试，内存测试可以发现许多诸如缓冲区溢出之类的漏洞，而这类漏洞使用除此之外的测试手段都难以发现。例如，对软件运行时的内存信息进行扫描，看是否存在一些导致隐患的信息，当然这需要专门的工具来进行验证（比如：HP WebInspect、IBM Appscan和Acunetix Web Vulnerability Scanner）。

##### 43. 自动化测试

自动化测试是通过测试工具或其他手段，按照测试工程师的预定计划对软件产品进行自动化的测试，通俗地说也就是用程序测程序，用脚本的运行代替手工测试。从广义上来说，自动化测试包括一切通过工具（程序）的方式代替或辅助手工测试的行为都可以看作自动化，包括性能测试工具，或自己写的一段程序。

优缺点：

优点：

1.对程序回归测试更方便

2.建立可靠、重复的测试，减少人为失误，更好地利用资源

3.增强测试质量和覆盖率

4.执行手工测试不可能完成的任务

缺点：

1.不能取代手工测试，自动化测试没有思维，设计的好坏决定了测试质量

2.发现的问题和缺陷比手工测试要少

3.不能用于测试周期很短的项目，不能保证100%的测试覆盖率，不能测试不稳定的软件和软件易用性

##### 44. 软件质量保证

软件质量保证是贯穿软件项目整个生命周期的有计划的系统活动，经常针对整个项目质量计划执行情况进行评估、检查和改进，确保项目质量和计划保持一致。

##### 45. 逻辑覆盖

语句覆盖

判定覆盖

条件覆盖

判定/条件覆盖

条件组合覆盖

路径覆盖

##### 46. SDL

安全开发生命周期（SDL）即Security Development Lifecycle，是一个帮助开发人员构建更安全的软件和解决安全合规要求的同时降低开发成本的软件开发过程。 自2004年起，微软将SDL作为全公司的计划和强制政策，SDL的核心理念就是将安全考虑集成在软件开发的每一个阶段:需求分析、设计、编码、测试和维护。从需求、设计到发布产品的每一个阶段每都增加了相应的安全活动，以减少软件中漏洞的数量并将安全缺陷降低到最小程度。安全开发生命周期 (SDL)是侧重于软件开发的安全保证过程，旨在开发出安全的软件应用。

##### 48. 模糊测试

模糊测试 (Fuzzing)，简而言之，就是为了触发新的或不可预见的代码执行路径或bug而在程序中插入异常的、非预期的、甚至是随机的输入。 因为模糊测试涉及到为目标提供大量的测试样例，因此至少也会实现部分自动化。 模糊测试可以也应当用于测试每个需要接受某种形式输入的接口。 实际上，模糊测试最起码就应该拿来用于测试每个从潜在恶意来源 (比如互联网或用户提供的文件)获取输入的接口。 模糊测试是对其他测试技术的补充。 由模糊测试揭露出的问题往往是开发人员不太可能构建的输入（例如，在处理一些边界情况，数据正确性和错误处理例程时的输入）触发的。 在常规自动化测试过程中，模糊测试扩大了代码覆盖范围，提高了代码覆盖率测试程度。 通过模糊测试使用的非预期输入通常会触发一些平时不会触发的执行流。

##### 49. 渗透测试

渗透测试通常指模拟黑客采用的漏洞发掘技术及攻击方法，测试工程师对被测试单位的网络、主机、应用及数据是否存在安全问题进行检测的过程，这种活动主要是发现系统的脆弱性，评估信息系统是否安全，从攻击者角度发现分析系统的缺陷及漏洞，并利用这些漏洞实现主动攻击。 渗透测试是一个逐渐深入信息系统内部的过程，有可能影响业务信息系统的正常运行，所以一般选择对业务最少影响的通用方法测试。 通常被称作“踩点”，是进行攻击入侵的首要任务。 收集信息系统的网络信息，主要是为了制定有目的性和针对性的渗透测试计划和方案，来减小被发现的概率、提高测试的成功率、减小漏报率。 “踩点”的主要内容有扫描主机域名和IP地址、判断操作系统类型、发现系统开放的端口及该端口对应的应用程序、扫描系统账号列表、收集系统配置信息等。

##### 50. SBSE

定义：基于搜索的软件工程是传统软件工程和智能计算交叉的领域，它采用智能计算领域的现代启发式搜索优化算法解决软件工程的相关问题。

核心：实现智能化和自动化的软件工程相关问题求解。

常用算法：遗传算法、爬山算法、模拟退火算法、蚁群算法、粒子群算法、遗传编程

应用：该技术可以运用在软件工程的各个阶段中，比如需求分析、项目管理、软件测试等等阶段。