**软件测试**

**1.什么是软件测试**

软件测试是在规定的条件下对程序进行操作，以发现错误，对软件质量进行评估。

**2.软件测试的目的是什么？**

（1）发现软件的缺陷和错误（2）保证软件的质量，确保能够满足用户以及产品的需求。

软件测试的目的是为了找bug，并不是为了验证软件没有bug。

**3.白盒测试用例设计常用方法**

静态测试：不用运行程序的测试，如文档测试、代码检查等

动态测试：需要执行代码，接口测试、覆盖率分析、性能分析、内存分析等。

逻辑覆盖法：主要包括语句覆盖，判断覆盖，条件覆盖，判断/条件覆盖，条件组合覆盖，路径覆盖等。

六种覆盖标准发现错误的能力由弱到强的变化：

语句覆盖，每条语句至少执行一次。

判断覆盖，每个判断的每个分支至少执行一次。

条件覆盖，每个判断的每个条件应取到的各种可能的值。

判断/条件覆盖，同时满足判断覆盖条件覆盖。

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。

路径覆盖，使程序中每一条可能的路径至少执行一次。

**4.黑盒测试用例设计常用方法**

等价划分类，边界值分析，错误推测法、因果图法、场景法、正交试验设计法、判定表驱动分析法、功能图分析法等。

**5.什么是灰盒测试？**

灰盒测试，是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段。目前互联网的测试大多数都是灰盒测试。

**6.列举出你所了解的软件测试方式**

软件的生命周期划分：单元测试、集成测试、系统测试、回归测试、验收测试。

测试测试对象划分：功能测试、性能测试、稳定性测试、易用性测试、安全性测试。

测试实施者划分：开发方测试（α测试）、用户测试（β测试）、第三方测试。

测试用例设计划分：白盒测试、黑盒测试、灰盒测试。

按照测试执行方式划分：静态测试、动态测试。

是否手工划分：手工测试、自动化测试。

测试对象划分：程序测试、文档测试。

**7.什么是单元测试**

答：完成最小的软件设计单元（模块）的验证工作，确保模块被正确编码。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，及早发现和解决不易显现的错误。

**8.单元测试、集成测试、系统测试、验收测试、回归测试这几步最重要的是哪一步？**

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，系统测试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需求规格的定义，因此，我认为系统测试很重要。

**9.集成测试和系统测试的区别，以及应用场景分别是什么？**

执行顺序：先执行集成测试，再做系统测试。

用例粒度：集成测试比系统测试用例更详细，集成测试对于接口部分也要重点写，而系统测试的用例更接近用户接受的测试用例。

应用场景：

集成测试：一般包含接口测试，对程序的提测部分进行测试。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和功能性测试，又包含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。测试方法一般采用黑盒测试法。

**10.测试开发需要哪些知识？具备哪些能力？**

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；

编程语言基础，如C/C++、java、python等；

自动化测试工具，如Selenium、Appium等；

计算机基础知识，如数据库、Linux、计算机网络等；

测试卡框架，如JUnit、Pytest、Unittest等。

具备的能力：

业务分析能力、缺陷洞察能力、团队协作能力、专业技术能力、逻辑思考能力、问题解决能力、沟通表达能力和宏观把控能力。

1. **请说一下手动测试与自动化测试的优缺点**

手工测试的优点：

自动化测试无法替代探索性测试，发散思维的测试

手工测试缺点：

重复的手工回归测试，代价昂贵、容易出错、效率慢

自动化测试的缺点：

无法运用在测试复杂的场景，手工测试比自动化测试发现的缺陷更多。

对测试质量的依赖性极大，自动化测试不能提高有效性。

自动化测试的优点：

对程序的回归测试更方便。

可以运行更多更繁琐的测试。

可以执行一些手工测试困难或不可能进行的测试。

更好地利用资源。

测试具有一致性和可重复性。

测试的复用性。

增加软件的信任度。

**12.自动化测试的运用场景举例**

线上回归（UI+接口）

手工测试难以执行的操作

简单场景监控

稳定性测试（monkey+遍历测试）

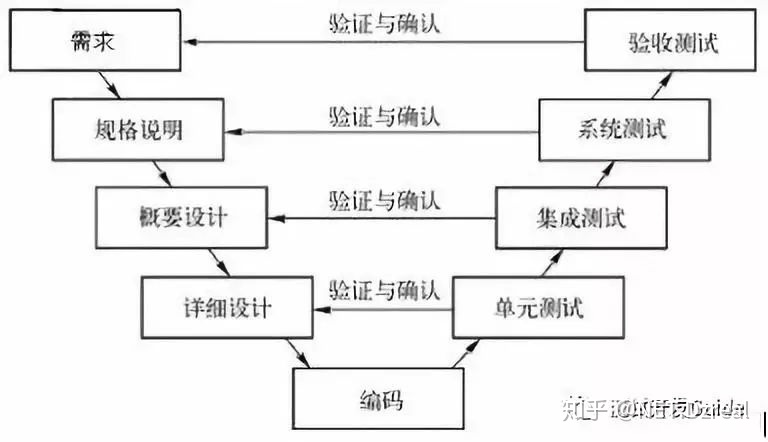
**13、软件测试的核心竞争力是什么？**

答：早发现问题和发现别人无法发现的问题。

**14.测试和开发要怎么结合才能使软件的质量得到更好的保障**

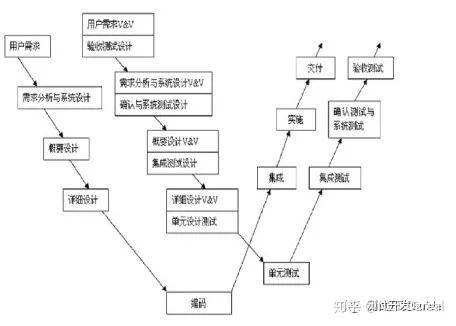
测试和开发可以按照V模型或W模型的方式进行结合。但应该按照W模型的方式进行结合比较合理。

V模型：



测试过程加在开发过程的后半段，比较被动。

W模型：



测试提前，甚至和开发是同步进行，测试不仅是程序，还包括需求和设计。W模型有利于尽早地全面的发现问题，降低软件开发的成本，风险前置。

**15.怎么实施自动化测试**

明确测试目的。

判断项目适不适合进行自动化测试。

对项目做测试分析。

制定测试计划和测试方案。

搭建自动化测试框架。

设计或编写测试用例。

执行自动化测试。

评估。

**16.测试的相关流程**

按W模型：

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

实际工作中的测试流程：

需求评审 -> 技术评审 -> case评审 -> 开发自测以及冒烟测试 -> 整体提测（集成测试） -> 回归测试 -> 系统测试 -> 验收测试

**17.测试项目具体工作是什么**

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

跟踪BUG修改情况

自动化测试

性能测试、压力测试、安全测试等其他测试

**18.BUG分级**

两个维度去划分

按BUG严重程度划分等级：

**blocker：**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，系统不稳定。常见的：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、服务器500等。

**critical：**影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统的稳定性。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别字或拼写错误。

**major：**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，长时间操作无法提示、系统未优化、兼容性问题。

**minor/trivial**：易用性及建议性的问题。

按BUG处理优先级划分：

immediate：马上解决

urgent：急需解决

high：高度重视，有时间马上解决

low：在系统发布前解决或确认可以不用解决

**19.APP性能指标有哪些？**

答：内存、CPU、流量、电量、启动速度、滑动速度、界面切换速度、与服务器交互的网络速度。

**20.APP测试工具有哪些？**

接口测试：postman

性能测试：jmeter

抓包工具：chales、fiddler

UI自动化：uiautomator2、appium、atx

稳定性测试：monkey、maxim、uicrawler、appcrawler

兼容性测试：wetest、testin

内存、cpu、电量测试：GT、soloPi

弱网测试：chales

**21.BUG的生命周期**

复杂版：

New（新的）

Assigned（已指派）

Open（打开的）

Fixed（已修复）

Pending Reset（待测试）

Reset（再测试）

Closed（已关闭）

Reopen（再次打开）

Pending Reject（拒绝中）

Rejected（被拒绝的）

Postponed（延期）

简单版：

创建bug

分配bug

修复完待测试

关闭

重新开启

无效

**22.什么是α测试和β 测试？**

α测试：在受控的环境下进行，由用户在开发者的场所进行，开发者指导用户测试，开发者负责记录发现的错误和使用中遇到的问题。

β 测试：在开发者不可控的环境下进行，由软件最终用户在一个或多个客户场所下进行，用户记录测试中遇到的问题，并定期上报给开发者。

**23.谈谈对敏捷的理解**

提到敏捷，不得不联想到瀑布开发。

瀑布开发按项目为核心，一般都会有一个相对较长的项目周期，一开始把项目设计得大而全，完成项目并交付后，工作重心就会转移到另一个项目去。

敏捷开发是以需求为核心，一开始不会把产品设计得大而全，而是通过快速迭代的方式，不断采集需求，不断更新迭代。敏捷开发的开发周期更短，能够快速试错，快速迭代，敏捷开发比瀑布开发更顺应目前的软件开发趋势。

敏捷开发也对应着有敏捷测试，测试环节贯穿整个迭代周期，从需求评审到发布上线，都离不开测试快速跟进。

测试左移：需求评审、用例设计、自测工具、静态代码扫描等；

测试中：业务测试，接口测试，性能测试等；

测试右移：稳定性测试，回归测试，灰度测试等

**24.什么是压力测试？压力测试需要考虑哪些因素？**

压力测试是在高负载情况下，对系统稳定性进行测试。在高负载的情况下，系统出现异常的概率要比正常负载时要高。高负载包含长时间运行、大数据、高并发等情况。

在做压力测试时，一般要考虑环境因素、性能指标、运行时间等要素。

压测环境最好和生产环境一致，假如要在生产环境进行压测，需要在凌晨等在线用户量极少的情况下进行。在生产环境测试时要做好数据隔离，生产环境需提供虚拟数据，采用虚拟账号，避免对真实线上用户造成影响。

性能指标包括，内存、CPU、TPS、QPS、网络流量、错误统计等，这些指标需要监控。

压测一般需要运行长时间，最好能够通过长时间的压测，绘制出曲线图，这样更容易观察到性能瓶颈。