

测试理论基础

什么是软件测试

答：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。

软件测试的目的是什么

答：软件测试的目的在于（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。**

（标重点）**软件测试的目的是为了找bug，并不是验证软件没有bug。**

白盒测试用例设计常用方法

答：

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：

1. **语句覆盖**，每条语句至少执行一次。
2. **判断覆盖**，每个判断的每个分支至少执行一次。
3. **条件覆盖**，每个判段的每个条件应取到的各种可能的值。
4. **判断/条件覆盖**，同时满足判断覆盖条件覆盖。
5. **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。
6. **路径覆盖**，使程序中每一条可能的路径至少执行一次。

黑盒测试用例设计常用方法

答：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。

什么是灰盒测试？

答：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。

列举出你所了解的软件测试方式

答：

按照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。

按照测试关注点划分：功能测试、性能测试、稳定性测试、易用性测试、安全性测试。

按照测试实施者划分：开发方测试（ α 测试）、用户测试（ β 测试）、第三方测试。

按照技术/测试用例设计划分：白盒测试、黑盒测试、灰盒测试。

按照分析方法划分：静态测试、动态测试。

按照测试执行方式划分：手工测试、自动化测试。

按照测试对象划分：程序测试、文档测试。

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。

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。

集成测试和系统测试的区别，以及应用场景分别是什么？

答：

区别：

- 执行顺序：先执行集成测试，待集成测试问题修复后，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。

测试开发需要哪些知识？具备哪些能力？

答：

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。

请说一下手动测试与自动化测试的优缺点

答：

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。

手工测试的优点：

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。

自动化测试的缺点：

1. 不能取代手工测试。
2. 无法运用在测试复杂的场景
3. 手工测试比自动化测试发现的缺陷更多。
4. 对测试质量的依赖性极大。
5. 自动化测试不能提高有效性。
6. 比手动测试脆弱，需要维护成本。
7. 工具本身并无想象力。

自动化测试的优点：

1. 对程序的回归测试更方便。
2. 可以运行更多更繁琐的测试。
3. 可以执行一些手工测试困难或不可能进行的测试。
4. 更好地利用资源。
5. 测试具有一致性和可重复性。
6. 测试的复用性。
7. 增加软件的信任度。

自动化测试的运用场景举例

答：

1. 线上巡检（UI+接口）
2. 简单场景监控
3. 稳定性测试（monkey+遍历测试）

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。

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

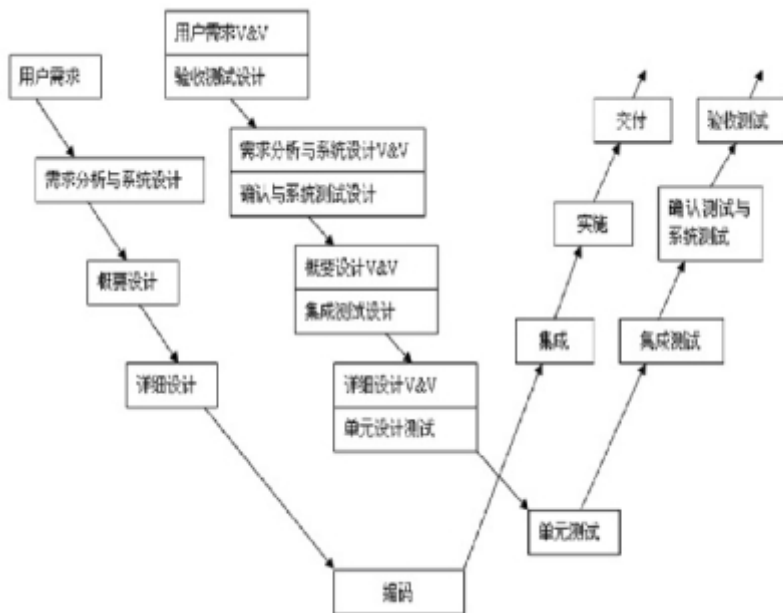
答：测试和开发可以按照V模型或W模型的方式进行结合。但应该按照W模型的方式进行结合比较合理。

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。

怎么实施自动化测试

答：

1. 首先判断项目适不适合进行自动化测试。
2. 对项目做需求分析。
3. 制定测试计划和测试方案。
4. 搭建自动化测试框架。
5. 设计或编写测试用例。
6. 执行自动化测试。
7. 评估。

测试的相关流程

答：

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

我工作中实际测试流程：

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

测试项目具体工作是什么

答：

1. 搭建测试环境
2. 撰写测试用例
3. 执行测试用例
4. 写测试计划、测试报告
5. 测试并提交BUG
6. 跟踪BUG修改情况
7. 自动化测试
8. 性能测试、压力测试、安全测试等其他测试

BUG分级

答：两个维度去划分

1. 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。
2. 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

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。

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

BUG的生命周期

答：

复杂版：

1. New (新的)
2. Assigned (已指派)
3. Open (打开的)
4. Fixed (已修复)
5. Pending Reset (待测试)
6. Reset (再测试)
7. Closed (已关闭)
8. Reopen (再次打开)
9. Pending Reject (拒绝中)
10. Rejected (被拒绝的)
11. Postponed (延期)

简单版：

1. 创建bug
2. 分配bug
3. 修复完待测试
4. 关闭
5. 重新开启
6. 无效

什么是 α 测试和 β 测试？

答：

α 测试：在受控的环境下进行，由用户在开发着的场所进行，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。

谈谈对敏捷的理解

答：

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。而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

什么是压力测试？ 压力测试需要考虑哪些因素？

答：

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。

补充问题集（无标准答案）

软件测试的目的是什么

1.为了发现程序中的缺陷，保证软件质量。 2.满足用户需要

常见的测试类型有哪些？分别说明一下？

- 1.黑盒测试，即常说的功能测试
- 2.白盒测试，即单元测试，通常由开发来完成，对程序类和方法的测试
- 3.兼容性测试，主要是浏览器的兼容测试
- 4.集成测试，即各个模块的测试
- 5.系统测试，各模块测试完成后，对整个系统的完整性测试
- 6.回归测试
- 7.验收测试

测试用例设计常用的方法有哪些？详细说明一下？

最常用的3种 等价类划分、边界值、场景法

1.等价类划分 分为有效等价类和无效等价类，将测试的范围划分成几个互不相交的子集，从每个子集选出若干个有代表性的值作为测试用例

- 2.边界值：选取正好等于、刚刚大于、刚刚小于边界的
- 3.场景法：划分不同的场景，然后逐一进行验证

解释下单元测试,集成测试,系统测试以及验收测试？

- 1.单元测试，通常由开发来完成，对程序类和方法的测试
- 2.集成测试，即各个模块的测试
- 3.系统测试，各模块测试完成后，对整个系统的完整性测试
- 4.验收测试，测试环境测试通过后，由产品或者用户进行验收测试，看看产品的实现，是不是满足了他们当初设计的需求

探索性测试是什么？应该怎么做？

在需求文档不完善或者压根没有需求文档的情况下，根据经验进行摸索尝试性进行的测试，是测试过程中形成的基本的思维性测试

什么是冒烟测试,如何有效的开展冒烟测试？

- 1.软件最基本的功能测试，通常由开发完成，只有冒烟点都通过的产品，交由测试，才会比较有意义
- 2.冒烟测试贯穿于测试的各个阶段，比如集成测试，系统测试等

一条高质量的缺陷记录(Bug)应该具有哪些内容？

- 1.记录bug产生的前提条件
- 2.产生bug的详细操作步骤
- 3.截图，直观的展示问题，有效帮助开发快速定位问题

你觉得软件测试工程师在一个团队中，都需要做什么？有什么价值？

- 1.在产品需求会上 可以站在用户的角度 提一些贴合用户需求的建议
- 2.测试阶段，根据自己对项目的了解和专业的测试技能 尽可能在项目上线前测出更多问题 保证产品质量

- 3.对测试过程中出现的问题 合理有效的进行风险评估。
- 4.对质量严格把关 测试不通过的 一定不能发布生产

你对软件测试最大的兴趣是什么？

- 1.行业前景比较好，而且近些年 越来越多企业更注重测试
- 2.测试时间越久 面临的困难和挑战也越多，解决问题的同时 也提高了自身的能力
- 3.自己的性格比较外向开朗 很容易跟产品和开发沟通 做起事情可以事半功倍

在你以往的工作中，发现的影响大或印象深刻的Bug是什么？为什么？

在没有任何文档的情况下,你如何开展测试？

- 1.尽可能的去获取其他的文档吧，比如开发的一些设计文档---概要设计、功能设计、详细设计
- 2.尝试从网上找类似的项目文档，如果幸运，可以从公司的资料库找到类似的文档
- 3.按照开发的接口或者设计文档，按照功能模块划分，大致写一些用例，然后进行摸索测试，在测试过程中逐渐完善自己的用例

测试用例是什么？如何设计有效的测试用例？

为了测试某个产品，编制的一组测试输入、执行条件以及预期结果
设计有效的测试用例：

- 1.明确需求，清晰的知道需求要实现哪些功能
- 2.根据需求文档，拆分出功能点和测试测试要点
- 3.详细的梳理业务需求，设计不同的业务场景，尽可能多的覆盖，尤其重要的逻辑，颗粒度要精细

- 4.具体逻辑的设计方法，遵循边界分析法，出问题最多的就在边界值，然后用等价类划分方法补充一些测试用例
- 5.UI测试，界面元素测试+样式+操作控件设计+浏览器兼容性相关的用例
- 6.时间充足的情况下，设计接口的测试用例，从而保证接口数据的用等价类划分方法补充一些测试用例完整性和正确性

一个水杯你会怎么测试

需求测试：查看杯子使用说明书

界面测试：查看杯子外观

功能度：用水杯装水看漏不漏；水能不能被喝到

安全性：杯子有没有毒或细菌

可靠性：杯子从不同高度落下的损坏程度

可移植性：杯子在不同的地方、温度等环境下是否都可以正常使用

兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等

易用性：杯子是否烫手、是否有防滑措施、是否方便饮用

疲劳测试：将杯子盛上水(案例一)放24 小时检查泄漏时间和情况；盛上汽油(案例二)放24 小时检查泄漏时间和情况等

压力测试：用根针并在针上面不断加重量，看压强多大时会穿透

自动化测试问题

- 你认为适合做自动化测试的标准是什么？
- 你认为什么类型的测试不适合做自动化测试？
- UI自动化测试的优点和缺点分别是什么？
- 在一个项目中目前还没有进行自动化，如果我想开展自动化测试，我应该怎么做(一般步骤)？
- 你认为该如何选择最适合的自动化测试工具？
- 什么是自动化测试框架？一个好的自动化测试框架应该具备什么元素？
- 说一下你写过的测试框架的代码目录结构是怎么样的？
- 自动化测试框架的类型有哪些？
- 说一下你在实施自动化测试过程中好的代码实践？

- 自动化测试是否仅仅可以是实施在UI层?为什么?
- 你是否熟悉Selenium工具?说一下它是什么?
- 你是否还熟悉其它的自动化测试工具?各自简单的介绍一下?

性能测试问题

- 什么是性能测试?为什么要进行性能测试?
- 性能测试的类型有哪些?
- 列举下用户会面对的性能问题和性能瓶颈?
- 列举下性能测试中常涉及的性能计数?
- 性能测试中并发用户点击量是什么?如何实现?
- 性能测试进入和结束的标准是什么?
- 说一下在选择性能测试工具之前需要考虑哪些东西?
- 在性能测试中,如何识别性能瓶颈?
- 在对应用程序进行性能测试期间一般会执行哪些活动?
- 性能测试中吞吐量是什么?
- 解释下什么是耐力测试和尖峰测试?