

测试知识点

答疑篇

***面试必问题：

- 1.你为什么需要做软件测试？（要讲案例） 测开和研发的区别？
- 2.什么是软件测试：当你拿到一个软件时，你对软件进行操作，看测试的结果和规格需求说明书或用户的期望是否是一样的。
- 3.测试和研发的区别：

难易程度：开发要求的专业度更高，测试更着重广度，要从各方面考虑问题

工作环境：基本类似

发展前景：自动化测试、安全测试等领域发展前景和研发基本一致

繁忙程度：一般测试比研发轻松，但敏捷模式（什么是敏捷模式？：最大的特点：快速迭代，半个月或一周一个迭代版本）下差距不大，产品发布前压力比较大

技能要求：测试要求更广泛，业务能力，设计和架构分析能力，测试手段和工具使用，用户的模型分析和理解，有一定的编程能力

测试和调试的区别：

目的不同：测试的任务是发现程序中的缺陷；调试的任务是定位并解决程序中问题

参与角色不同：测试主要由测试人员和开发人员来执行，黑盒测试主要由测试人员完成，单元/集成测试主要由开发人员执行。调试主要由开发人员完成

执行的阶段不同：测试贯穿整个开发生命周期，开发一般在开发阶段。

概念篇

一、测试目的和原则（不常考）

测试目的

测试软件有没有问题

测试原则：

要以客户为中心，遵循软件测试的规范、流程、标准和要求

（测试并不是仅仅为了找出错误，测试人员可以通过产生错误的原因、错误发展的趋势帮助项目管理者

了解当前软件开发过程中的缺陷，及时纠错、改进。同时帮助测试人员设计出有针对性的测试方法，从而

可以改善测试的效率和有效性)

从测试的目的出发，测试可分为两类：为了验证程序能正常工作的测试；为了验证程序不能正确运行的测试

二、需求

需求分为：用户需求和软件需求

用户需求：

使用产品的用户提出的需求，即用户使用产品时必须完成的任务，需求一般比较简略

软件需求：

也叫功能需求，该需求会详细描述开发人员必须实现的软件功能

如何将用户需求转换成软件需求？最核心的就是通过不断的沟通

三 什么是bug（重点）

理解性解答：对软件进行测试，测试出来的实际结果和测试用例的预期结果不一致就是缺陷当且仅当规格说明是存在的并正确，程序与规格说明之间不匹配是错误。当没有规格说明书时，判定标准以最终用户为准：

如果软件没有达到最终用户合理预期的功能要求时，就是软件错误。

四、测试用例（重点）

向被测试的系统提供的一组集合，这组集合包含：测试环境、操作步骤、测试数据、预期结果等要素

五种开发模型：瀑布模型、螺旋模型、迭代模型、增量模型、敏捷模型（重点）

软件的生命周期分为六个阶段：需求分析、计划、设计、编码、测试、运行维护

具体说：首先测试人员了解需求、对需求进行分解，从而得出测试需求。然后根据需求进行编写测试计划或方案。编写测试计划之后测试人员要搭建测试用例框架，根据需求和设计编写测试用例，在编码阶段，专业的白盒测试人员可以计划执行单元测试，完善细化测试用例以及调整测试计划和方案。测试阶段，是测试人员最为重要的阶段，根据测试用例和计划执行测试，在测试的过程中记录、管理缺陷，测试完成后编写测试报告。最后测试人员参与项目的实施工作，给用户使用软件的培训，在时运项目时收集问题并反馈给相关负责人

1.瀑布模型：

需求分析--->计划--->设计--->编码--->测试

优缺点：强调的是开发的阶段性，强调早期计划和需求调查，也强调产品测试。瀑布模型各个阶段是串行的，有前后的关联关系，

因此适合需求变更比较小的项目。但依赖于早期唯一一次的需求调查，不能适应需求的变化，风险较高。

2.螺旋模型

螺旋模型是一种渐进式的开发模型，这对于那些规模庞大、复杂度高、风险大的项目比较适合。

优点：强调的是风险，每个阶段都有风险分析。但缺点就是：一旦强调风险，控制风险，项目的成本较高，时间就花费比较久。

3.增量、迭代模型

增量模型就是鼓励用户反馈，在每个迭代过程中，促使开发小组以一种循环的、可预测的方式驱动产品的开发。增量和迭代还是有区别的：

以一个画人物图的例子来说明：增量就是逐块建造，先画人的头部，再画身体，最后画手脚。而迭代是反复的精益求精，我们可以采用先

画整体轮廓，再勾勒出基本的雏形，再逐个细化、着色等。

因此，增量中新增的功能对前面的功能没有任何影响，它们之间没有关联关系。迭代是在前面的功能上进行细化，前后有关系。

增量可以降低项目的风险，因为它把一个大型项目分成一小块一小块去做。

4.敏捷模型

敏捷模型强调人与人之间的沟通，轻文档，即对文档的依赖度降低了，同时也强调客户参与，根据用户的需求进行及时变更。

它与瀑布模型存在很大的区别：瀑布模型中需求不好变更，一般不欢迎变更。而敏捷模型随时欢迎变更需求

敏捷开发模型中有三个角色：product owner (po) 产品负责人，他负责整理user story (用户故事，即产品需求规格说明书分成几个小的功能)，

是一个不可拆分的单位

敏捷开发模型的流程：

scrum master负责召开会议、team团队成员

1.产品负责人负责整理user story，形成product backlog。

2.发布计划会议：po负责讲解user story，对其进行估算和排序，发布会议的产出就是制定出这一期迭代要完成的story列表。

3.迭代计划会议：项目团队对每一个story进行任务分解（一般一个user story不能超过一周），分解的标准是完成该story的所有任务，

每个任务都有明确的负责人，并完成工时初步估计。

4.每日例会：每天scrum master召集站立会议，团队成员回答昨天做了什么有什么问题，今天计划做什么。

5.演示会议：迭代结束之后，召开演示会议，相关人员都要参加，团队负责向大家展示本次迭代取得的成果。

在这期间大家的反馈记录下来，有po整理，形成新的story。

6.回顾会议：项目团队对本期迭代进行总结，发现不足，指定改进计划，下一次迭代继续改进，已达到持续改进的效果。

敏捷中的测试：

挑战1：轻文档

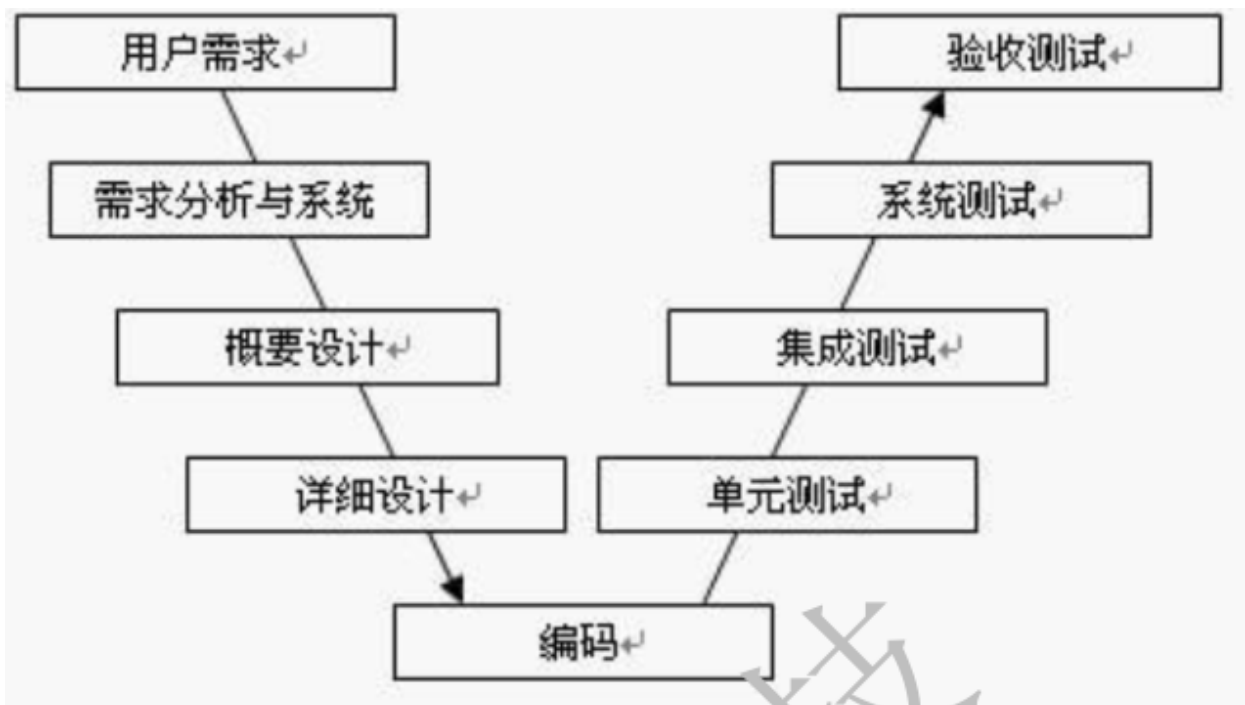
挑战2：快速迭代

1.测试工作的核心内容没有变，就是不断的找bug。

2.测试人员不能依赖文档，测试用例作用减弱，更多的采用思维导图、探索性测试（强调自由度，设计和执行同时开展，根据测试结果不断调整测试计划）、自动化测试。

软件测试的V模型和W模型（重点）

V模型



用户需求：分析用户需求（测试人员参与）

需求分析和系统：进行架构设计

概要设计

详细设计：模块里面具体应该怎么做，接口设计、数据库的设计

编码：研发人员进行编写代码，测试人员编写测试用例（测试人员参与）

单元测试：开发人员或白盒测试工程师参与

集成测试：开发人员或白盒测试工程师参与，用灰盒测试方法（灰盒=白盒+黑盒（接口）），关注接口和功能

系统测试：黑盒测试方法（测试人员参与，花费的时间最长）包含五个阶段：

1.搭建环境 2.数据准备 3.测试执行 4.缺陷管理 5.测试报告编写

验收测试：用户参与。黑盒测试方法

局限性：仅仅把测试作为在编码之后的一个阶段，未在需求阶段就进入测试

W模型（双V模型）



W模型测试人员全程参与

W模型特点：测试的对象不仅是程序，需求、设计等同样要测试，测试与开发是同步进行的

W模型优点：有利于尽早地全面的发现问题。例如，需求分析完成后，测试人员就应该参与到对需求的验证和确认活动中，以尽早地找出缺陷所在。同时，对需求的测试也有利于及时了解项目难度和测试风险，及早制定应对措施，显著减少总体测试时间，加快项目进度。

局限性：需求、设计、编码等活动被视为串行的；测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一个阶段工作。无法支持敏捷开发模式。对于当前软件开发复杂多变的情况，W模型并不能解除测试管理面临着困惑。

基础篇

软件测试的生命周期（区分软件的生命周期）

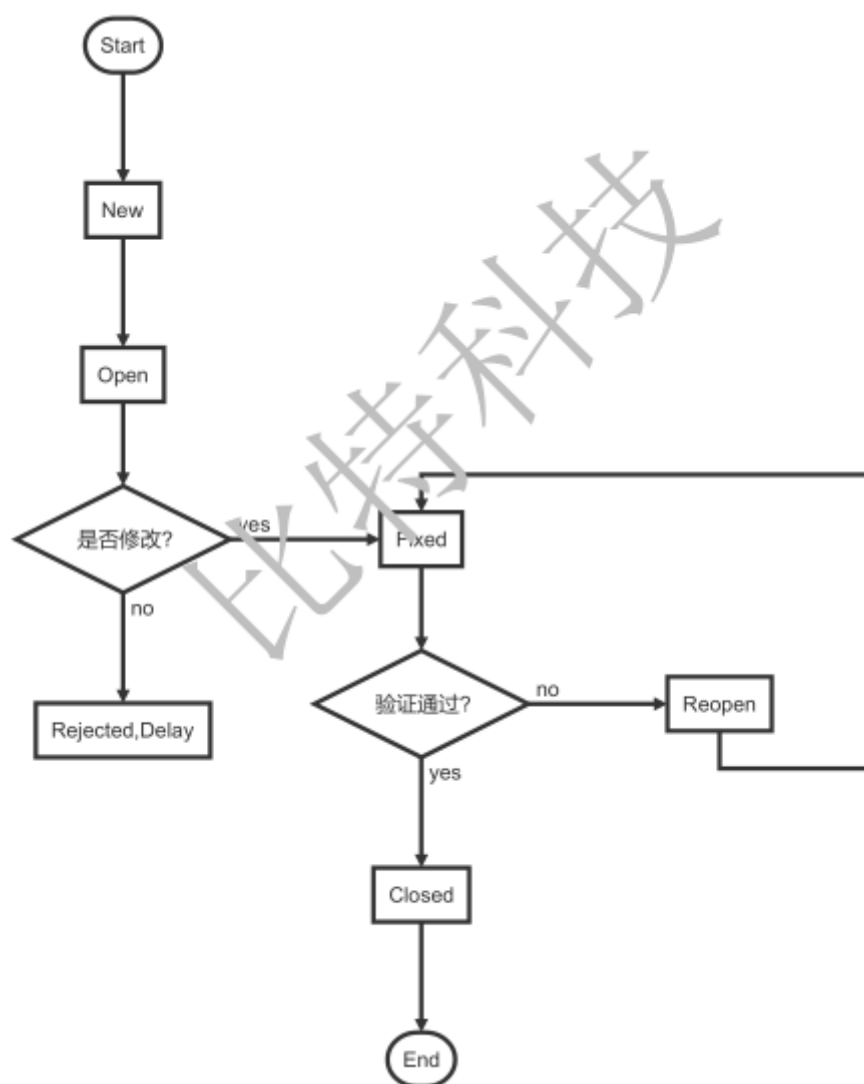
需求分析--->测试计划--->测试设计、测试开发--->测试执行--->测试评估

具体流程是：首先对需求进行分析，分析了解需求，确定测试的范围。根据我们对需求的了解以及确定的需求范围来制定计划。根据我们的测试计划的设计每个阶段的时间点来编写测试用例，测试用例编写完之后我们需要对测试用例进行评审，如果评审不通过，我们要继续修改测试用例，直到评审通过完之后，根据我们测试计划的时间点进行测试用例的执行。在这个阶段中，一旦发现缺陷，我们要对缺陷进行确认、记录、沟通、验证。本轮测试完成之后，我们需要对本次测试进行总结，看看哪些地方做得好，哪些地方做的不好，写进测试报告中。

Bug的级别

崩溃 严重 一般 次要

Bug的生命周期



- New:新发现的Bug，未经评审决定是否指派给开发人员进行修改。
- Open：确认是Bug，并且认为需要进行修改，指派给相应的开发人员。

- Fixed:开发人员进行修改后标识成修改状态，有待测试人员的回归测试验证。
- Rejected: 如果认为不是Bug，则拒绝修改。
- Delay:如果认为暂时不需要修改或暂时不能修改，则延后修改。
- Closed: 修改状态的Bug经测试人员的回归测试验证通过，则关闭Bug。
- Reopen: 如果经验证Bug仍然存在，则需要重新打开Bug，开发人员重新修改。

无效的bug: open->closed open-rejected-closed

缺陷状态变更流程每个项目团队的实际做法可能不大一样。并且需要结合实际的开发流程和协作流程来使用。

例如，测试人员新发现的Bug，必须由测试组长评审后才决定是否Open并分派给开发人员。测试人员Open的Bug可以直接分派给Bug对应的程序模块的负责人，也可以要求都先统一提交给开发主管，由开发主管审核后再决定是否分派给开发人员进行修改。Bug的跟踪以及状态变更应该遵循一些基本原则：

- 测试人员对每一个缺陷的修改必须重新取一个包含更改后的代码的新版本进行回归测试，确保相同的问题不再出现，才能关闭缺陷。
- 对于拒绝修改和延迟修改的Bug，需要经过包含测试人员代表和开发人员代表、用户方面的代表（或代表用户角度的人）的评审。

开始第一次测试

- 1、阅读所有项目有关的文档，包括：需求文档、设计文档、用户手册
- 2、尽可能参加各种项目会议，了解项目的背景、人员组成、尽可能的了解需求和业务。特别针对业务专业性较强的项目，例如银行业务，需要了解各种业务知识，如高低柜、一二三类账户等、存款、贷款等。
- 3、熟悉项目所使用的测试管理工具、配置管理工具，获取对应的地址和登录方式
- 4、阅读已有的测试方案和测试案例
- 5、阅读旧有的bug库，了解系统功能。尤其重要的是和现有的测试团队保持一致的故障定级原则
- 6、了解公司的规范要求，特别是用例编写规范、用例执行规范、bug提交规范、测试工具工具使用规范等

在进行了以上的准备工作之后，第一次测试工作到来了，我们需要与测试组长确认具体的工作内容：

- 1、测试的计划是什么？
- 2、测试的内容是什么？test case有多少？安排了几几天执行？有没有自由测试的时间？
- 3、我要测试的内容开发人员是谁？需求人员是谁？
- 4、分配给我的测试内容是否需要特殊的测试资源？资源是否满足需要？

测试的执行和BUG管理

现在我们开始进行测试了：

1. 打开待测试的系统
2. 打开测试管理工具用例模块，开始执行用例
3. 发现bug！进行复现并确认bug的复现步骤
4. 记录bug
5. 沟通bug
6. 验证以前提交的bug
7. 确认本次测试完成
8. 编写测试报告

执行测试时处理要做到测试用例和需求的覆盖外，还要有临时发挥的能力。根据自己的经验、对测试的感悟以及随机测试可以发现很多根据测试用例无法发现的缺陷。

不能拘泥于测试用例或者已经有的测试方法，在测试执行过程中要不断总结测试方法和测试故障模型。真正优秀的测试人员在执行测试时是想着做，做着想，这样的测试效果才好，尤其是在测试过程中，对程序的处理相当了解的情况下，测试的思路会更加清晰和全面。

用例篇

编写测试用例的设计方法

- 基于需求的设计方法
- 等价类
- 边界值
- 因果图
- 正交排列
- 场景设计法
- 错误猜测法

面试中编写测试用例常用的是：等价类、边界值、场景设计法、错误猜测
具体内容见课件

进阶篇



重点掌握：按开发阶段分、按是否查看代码、按测试对象划分