

测试的目的是什么？

- 发现软件缺陷
- 提升软件质量
- 避免发布后存在风险

黑盒测试是什么？

- 黑盒，看不见内部的实现逻辑，只针对外观进行测试
- 主要是功能测试，测试应用程序的功能是否符合预期

白盒测试是什么？

- 白盒，能看见内部的实现逻辑，针对内部代码进行测试
- 主要是单元测试、集成测试，一般是开发来做，因为需要写代码
- 验证不同代码块之间的逻辑关系是否符合预期

黑盒测试的测试方法有哪些？

等价类、边界值、场景法、错误推断、因果图、判定表、正交法、状态迁移法

什么时候用场景法？

写测试用例的时候不仅要考虑某个功能是否正常，还要从用户的角度去思考有哪些常见使用场景以及可能会遇到的异常场景

回归测试是什么？

在应用程序的功能修改/新增后，对应用主流程或者主要功能重新测试一轮

回归测试的目的？

确保新修改/新增的功能不会影响到应用的正常使用

测试用例都包含了什么内容？

换个问法：什么样的测试用例才是好的测试用例？

- 优先级
- 模块
- 测试用例名称
- 前置条件
- 步骤
- 测试步骤
- 预期结果

- 实际结果
- 执行人

提交的缺陷都包含了什么内容？

- 标题
- 缺陷类型
- 优先级
- 影响模块
- 标签
- 严重程度
- 出现频率
- 环境配置
- 操作步骤
- 实际结果
- 预期结果
- 附件
- 也可以附上定位出问题的根本原因

如何对一个物品设计测试用例？

从多个方面回答，不一定要全，但是需要有思路的回答，不能想到啥回答啥

- 需求测试
- 功能测试
- 界面测试
- 可靠性
- 安全性
- 可移植性
- 兼容性
- 易用性
- 压力测试

如果让你来设计一个测试用例，会考虑哪些因素？

从设计思路，可以回答将从以下几个方面去设计用例

- 需求测试
- 功能测试
- 界面测试
- 可靠性

- 安全性
- 可移植性
- 兼容性
- 易用性
- 压力测试

从具体编写角度，可以参考下面的答案

- 测试用例应该100%地覆盖测试业务需求
- 参照黑盒测试用例设计的一般方法：等价类、边界值、场景分析法等等
- 利用场景法模拟核心业务流程的正确执行
- 利用场景法设计测试用例时，往往是一个业务流程需要多条数据来验证
- 利用边界值法设计测试用例，能够验证输入值的边界处理是否正确
- 还要考虑用例的可执行性、可维护性、可阅读性，确保描述语句清晰易懂，方便不同成员查看阅读

使用过什么辅助测试的工具？用来干嘛？

- Charles、Fiddler：抓包、改包，不仅查看接口请求、响应内容，还能自定义接口返回内容
- Navicat：连接数据库工具，用来查询数据库数据
- Postman：接口测试工具
- Jmeter：接口测试、压力测试工具
- Wireshark：抓取硬件层面的请求包，不仅能抓 HTTP 协议的包，还能抓 TCP、UDP 的包

你知道的测试流程是什么样的？

换个问法：你觉得最理想的测试流程是应该如何的？

1. 理解测试需求文档
2. 产品经理组织评审需求文档
3. 交互设计师组织评审交互稿
4. 视觉设计师组织评审视觉稿
5. 编写测试计划
6. 编写测试点、测试用例
7. 组内评审测试用例
8. 执行测试用例
9. 输出测试报告
10. 版本复盘会议

你的测试计划包含了哪些内容？

前期准备

- 分配测试资源
- 准备测试工具
- 准备测试数据

测试范围

- 被测对象
- 主要的测试内容
- 明确测什么，不测什么

测试策略

需要进行什么类型的测试，比如

- 功能测试
- 兼容性测试
- 拓展测试
- 交叉测试
- 自动化测试
- 接口测试
- 性能测试

送测（提测）准备

确定以下时间

- 验收时间
- 送测时间
- 发布时间
- 测试时间

确定以下内容

- 改动范围
- 影响范围，评估做那些模块的回归测试
- 正常送测范围
- 是否有延期送测的内容
- 是否有发布风险点

测试风险评估

预估整个测试过程中可能存在的潜在风险，以及风险发生时的应对策略

做测试计划的好处？

- 明确测试范围
- 制定测试策略
- 预估工作量
- 分配测试资源
- 测试进度是可控的，实时知道目前测试完成情况
- 可以提前识别潜在风险，当需求发生变化时，我们需要做出响应

你认为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？

换个问法

- 如何处理和开发的关系？
- 跟开发的关系如何？

回答两个关键点

心平静和、换位思考，然后自由发挥吧

你认为测试工程师应该具有哪些素养？

1. 测试策略设计
 2. 测试用例设计
 3. 快速学习
 4. 探索性测试思维
 5. 缺陷分析
 6. 自动化测试技术
 7. 良好的沟通
- ？

你们公司 Bug 的生命周期是怎么样的？

一般有三种情况

- 新建-处理-待测试-测试中-关闭
 - 新建-无效缺陷-关闭
 - 新建-处理-待测试-测试中-重新打开
- ？

当开发人员说这个不是一个 Bug 时，如何应对？

- 首先，详细描述自己认为这个缺陷可能产生的风险和影响
- 其次，询问他认为不是 Bug 的原因，自行初步判断开发讲的是否合情合理
- 若无法准确判断，则查看需求文档、交互稿、视觉稿，确认缺陷是否符合预期或开发讲的是否正确
- 如果开发只是认为这是优化项而不是 Bug 时，需要评估该缺陷影响范围
- 最后，可以拉上产品、交互进行讨论、评估

- 但作为测试仍然需要自己的立场，需要表明自己对这个问题的看法（严重程度、影响范围）

临近发布时间节点，但还有已知缺陷没有解决，同时版本又重要又紧急，如何应对？

将从几个维度评估已知问题是否影响发布

出现概率

- 是否必现？
- 偶现概率？
- 偶现概率偏低是否可以考虑延期修复

影响范围

- 主流程or异常场景？
- 主要功能or次要功能？
- 若是主流程，是否影响正常功能使用？
- 若是主要功能，是否影响其他功能模块的使用？
- 若是异常场景，操作步骤是否繁琐
- 该功能/场景使用频率如何？
- 出现若是界面效果不如意能否接受？

规避措施

- 技术上有没有规避措施？
- 交互上能否提醒用户执行某些操作来规避出现缺陷？
- 出现问题后能否友好提示？

项目层面

- 发布时间是否能延期？
- 发布的版本是对内还是对外发布？
- 开发能否迅速解决问题？
- 解决问题成本如何？
- 产品团队能否接受带问题发布？
- 带问题功能能否接受暂不发布，只发布正常新功能？

结论

- 若是必现，主流程/主要功能，使用频率较高，无法有效规避出现该问题的出现，解决成本较高，产品团队也不接受带问题发布的话，必须解决了再发布，即使项目再紧急，没有质量的发布也是得不到用户的认可的
- 当然在实际情况中，还是要根据不同的情况进行判断的
- 也有可能存在产品团队直接做决定，我们作为测试只能提醒风险点和建议，无法直接参与决策

你觉得软件测试人员在项目过程中除了测试工作以外，还应该肩负哪些责任？

- 项目层面：需要及时发现项目测试流程的不足点，及时沟通并改善不足的地方，推进项目组的测试规范
- 个人层面：提升个人测试能力，包括学习测试工具、测试框架等，提升测试效率，进行更多拓展测试，发现更多边界问题
- 团队层面：分享个人测试技能或项目经验，帮助团队其他成员提升测试技能，避免踩上不必要的坑

怎样保证你写的用例能覆盖所有的需求

- 需求评审、交互评审前均需要认真阅读文档，提前准备疑问，在评审会上解决疑问
- 先写测试点，进行一轮评审，补充测试点
- 将测试点转换为测试用例，再进行一轮评审，补充测试用例
- 版本发布后，复盘找到通过扩展测试发现的缺陷，若测试用例又没有的，及时补充测试用例
- 客户反馈 Bug 或内部发现 Bug，确认是否因为测试用例没写导致的漏测，若是则及时补充测试用例

如何控制测试质量与测试进度？

- 测试前，评估核心功能影响范围和上线风险
- 制定详细的测试计划，和组内成员、产品、技术负责人一起评审合理性
- 测试过程中，出现问题，快速沟通、快速解决问题
- 测试过程中，及时了解团队测试进度，提前了解可能会出现的风险点
- 测试结束后，组织版本复盘会议，回顾此次测试过程中出现的问题，制定计划防止下次再次出现类似问题

测试过程中，发生过哪些特殊情况？

- 测试工作量评估不准确
- 需要增加额外的测试类型
- 修复某些严重缺陷，导致需要全量回归
- 送测延期
- 人员变动

有发生过漏测的问题吗？

有，一般漏测问题分为两种嘛

1.旧版本漏测：缺陷修复未考虑相关联功能影响

2. 送测阶段版本漏测：

- 用例未覆盖，需求未提及的问题
- 边界场景未考虑等用例设计问题
- 缺陷触发条件特殊，暂时未想到规避的方法，增加回归测试

用例设计上有没有漏了导致漏测？

有的

- 用例未覆盖，需求未提及的问题
- 边界场景未考虑等用例设计问题

具体的栗子

- 两个账号同时添加消息模板时，后添加的消息模板会把前面添加的覆盖掉
- 账号A、账号B 登录网站
- 账号A 添加消息模板A，提交后，账号B 添加消息模板B，然后只能看到消息模板B，没有消息模板A
- 预期结果是不同账号添加的消息模板不会相互覆盖

漏测原因

边界场景比较特殊，用例没有考虑到这种场景导致漏测，后续新增此场景的测试用例

如何避免漏测？

产品经理、交互设计师角度

1. 出更详细的需求文档，除了描述需求来源，背景，需要更多的实际应用场景描述
2. 出动态的交互稿、演示视频
3. 及时更新需求文档、交互稿
4. 参加用例评审

开发角度

- 参加用例评审，了解测试检查点，协助补充测试点
- 提供完整的技术实现流程图、数据流图

测试角度

- 多看几遍需求文档、交互文档，记录疑问点，评审会上积极沟通
- 梳理用户使用场景
- 提高测试能力，掌握更多的测试方法进行多维度测试，测试用例设计方法进行精准测试
- 组织测试用例评审

工作中遇到一个技术难点你会怎么解决？

- 如果是技术难点的话，我首先会选择自己尝试解决
- 一开始会先搜百度、谷歌、stackoverflow 看看有没有对应的解决方案
- 如果没有一般就直接问一些认识的大牛朋友了
- 此时，如果还没解决，可以问下部门的开发或者是测开的同事

- 最后，如果难点解决了，会详细记录解决的方法输出到博客中

假设有一个很有挑战性的任务给到你，你之前没有接触过的，你会怎么处理？

- 首先，要挖掘这个任务的背景、需求、优先级、完成目标，详细了解这个任务的来龙去脉【挖掘需求】
- 然后要了解任务的一个整体架构，比如不同端用到的技术都有哪些，如果看不懂就要跟开发或者负责人沟通清楚【了解架构】
- 将任务根据类型或者步骤分成多个小任务，给每个小任务制定详细的测试方案、技术方案【任务分解】
- 根据方案查漏补缺自己的知识点和技术，不懂的就积极问【查漏补缺】
- 执行任务过程遇到问题记录下来，自行找解决方案或者找开发、大牛解答【沟通疑问】
- 最终输出任务结果的时候，输出每个小任务的关键结果数据以及结论【总结结果】

工作中遇到了什么困难，怎么解决的？

- 如果是说技术上的话，之前做 UI 自动化的时候就出现过脚本稳定性不好，还有执行速度比较慢的情况
- 那如果是工作上的话，就是我们的流程不够规范，比如没有交互稿、开发会延期送测
- 解决没有交互稿的问题就是直接跟交互设计部负责人沟通，后面就有专门的交互来做交互设计稿
- 解决开发延期送测的问题就是新加入了一个项目管理，负责确定每个产品的送测时间、发布时间，并要求开发无特殊原因不能随意延期送测

工作中经常有人找你打断你手头上的工作的话你会怎么处理？

一般会有以下四种因素打断手头上的工作

- 来自同事、部门的：临时需要你帮忙、临时插入了其他的需求
- 来自领导的：临时分配你一个不相关的任务
- 来自生活的：朋友、快递等等
- 来自自己的：微博、短信、微信等等

对于前三种属于客观因素，不可抗力只能选择应对，对于第四种属于主动因素，可以自己提高自制力

- 将一些杂事分为两个维度，紧急和不紧急
- 将所有紧急的事情连在一起处理，及时给到同事、部门一个答复
- 对于不紧急的事情，先给同事、部门一个预期解决事情时间点的答复，后面再进行处理

另外，自己每天的工作都要有一个详细的时间表

H5 的兼容性测试你做过哪些？

- 系统：Android（6-11）、iOS（10-14）
- 分辨率：刘海屏、挖孔屏、全面屏、曲面屏

- 手机品牌：小米、华为、三星、vivo、oppo
- 浏览器：微信内置浏览器、QQ内置浏览器、手机系统自带浏览器、第三方浏览器

H5 测试你会关注哪些测试点？

