

测试相关

软件测试方法分类

白盒测试

概念：关注程序代码的具体细节，根据软件内部代码的逻辑结构分析来进行测试。主要是通过阅读程序代码或者通过使用开发工具中的单步调试来判断软件质量。**关注代码的实现细节。**

测试用例设计方法：逻辑覆盖、循环覆盖、基本路径覆盖、判定覆盖等

1. 语句覆盖 (Statement Coverage)：测试人员通过执行所有可执行的语句，以确保每个语句至少被执行一次。
2. 分支覆盖 (Branch Coverage)：测试人员通过执行所有的分支路径，以确保每个条件分支语句至少被执行一次。
3. 路径覆盖 (Path Coverage)：测试人员通过执行所有可能的路径，以确保每条程序路径至少被执行一次。
4. 条件覆盖 (Condition Coverage)：测试人员通过执行所有可能的条件，以确保每个条件语句至少被执行一次。
5. 判定覆盖 (Decision Coverage)：测试人员通过执行所有可能的条件组合，以确保每个条件语句至少被执行一次。
6. 多重条件覆盖 (Multiple Condition Coverage)：测试人员通过执行所有可能的多重条件组合，以确保每个多重条件语句至少被执行一次

黑盒测试

概念：不考虑其内部结构，即具体代码实现，检测软件的各个功能能否得以实现，确认软件功能的正确性，依靠软件说明书来判断测试用例，**关注具体的客户需求及软件功能。**

测试用例设计方法：边界值分析法、等价类划分、错误猜测法、因果图法、状态图法、测试大纲法、随机测试法、场景法

静态测试

概念：在不运行程序的情况下寻找代码中可能存在的错误，或者评估程序代码

动态测试

实际运行被测程序，输入相应的测试用例，检查运行结果与预期结果的差异，从而验证程序的**正确性、有效性和可靠性**，并且分析**系统运行的效率和健壮性**。

质量模型

重点：功能、性能、兼容、易用性、安全

结论：无论测试硬件或软件，都应该从以上几点来进行分类验证

测试流程

需求分析（评审）、测试计划、用例设计、用例执行、缺陷管理、测试报告

首先是需求分析，然后是负责人制定测试计划，之后我们按照测试计划进行用例设计，用例设计完进行用例的评审。等开发产品编译完毕，测试组进行测试，对测试过程中发现的缺陷提交到缺陷管理平台，对 bug 进行跟踪、回归测试，直至缺陷满足用户需求。测试结束后，对测试结果进行分析，编写测试报告。

测试用例

用例编号、测试项、用例名称、前置条件、操作步骤、预期结果、优先级

测试分类（测试方法）

- 阶段划分：

单元测试：针对程序源代码进行测试

集成测试：针对单元与单元之间的接口进行测试

系统测试：针对系统整体功能+兼容+文档（说明、安装文档）

验收测试：

- 代码可见度：

黑盒测试：看不见源码，主要对程序功能进行测试

灰盒测试：看见部分源码，主要对程序接口进行测试

白盒测试：看见全部代码，主要针对程序源代码进行测试

测试用例设计方法（黑盒测试）

- 等价类划分法：解决穷举场景
- 边界值分析法：解决边界限制问题
- 判定表法：解决多条件有依赖关系的测试
- 错误推荐法
- 正交实验法

接口测试

通过代码或者工具模拟 http 请求的发送与接收，postman 等工具

调用接口，验证接口的正确性可用性

自动化测试

selenium：是一个自动化测试工具，利用它可以驱动浏览器执行特定的动作，如点击、下拉等操作，同时还可以获取浏览器当前呈现的页面的源代码

测试题

如何测百度首页

1. 功能测试：测试百度首页的各项功能是否正常，例如搜索框、热搜词、各类搜索功能、登录、注册等等。
2. 性能测试：测试百度首页的**页面加载时间、响应时间、并发用户数等性能指标**，找出性能瓶颈并提出优化建议。
3. 兼容性测试：测试百度首页**在不同浏览器、操作系统和设备上的兼容性**，例如 Chrome、Firefox、Safari、IE 等主流浏览器，不同的分辨率
4. 安全测试：测试百度首页的安全性，包括 XSS、CSRF、SQL 注入等安全漏洞，禁止敏感内容检索。
5. 易用性测试：有没有在线帮助文档，对用户是否友好
6. 用户体验测试：测试用户在使用百度首页时的整体体验，例如页面布局、字体大小、颜色搭配、交互设计等。
7. 可访问性测试：测试百度首页的可访问性，包括对残障人士的支持、对多语言的支持等

如何测试微信发红包功能

遵循：成功发送红包—聊天界面看到红包消息—点开红包消息—电机“开”按钮—查看抢红包详情—账户余额

1. 功能测试：

在红包钱数，和红包个数的输入框中只能输入数字

余额不足时，红包发送失败

红包里最多和最少可以输入的钱数

输入红包钱数是不是只能输入数字

2. 性能测试：

不同网速时抢红包，发红包的时间

发红包和收红包成功后的跳转时间

收发红包的耗电量

3. 兼容性测试：

ios 和安卓是否都可以发送红包

电脑端是否可以抢微信红包

4. 安全性测试：

红包被领取以后，发送红包人的金额会减少，收红包金额会增加

发送红包失败，余额和银行卡里的钱数不会少

红包发送成功，是否会收到微信支付的通知

5. 易用性测试：

红包描述，可以通过语音输入

支付方式可以为指纹支付也可以为密码支付

设计一个登录页面的用例(套路)

1. 功能测试：正确输入、为空输入、字符类型校验、长度校验、密码是否加密显示、大写提示、跳转页面是否成功、登出后用另一个账号登录

2. UI：界面布局合理、风格统一、界面文字简洁好理解、没有错别字

3. 性能测试：打开登录页面需要几秒、点击登录跳转首页需要几秒、多次点击、多人点击

4. 安全性：用户名和密码是否加密发送给服务器、错误登录的次数限制（防止暴力破解）、一台机器登录多个用户、一个用户多方登录、检查元素能否看到密码

5. 兼容性测试：**不同浏览器、不同的操作系统、不同的移动设备，不同的分辨率界面**

6. 易用性：输入框可否 tab 键切换、回车能否登录等。

购物车测试用例

- 功能测试：从商品信息页面添加的商品能显示在购物车中、若未登录，点击购物车中的商品直接进行结算，则提示用户输入用户名和密码，或者提示用户进行注册、若没有选择任何商品，点击结算，则提示用户“请添加要结算的商品”、勾选商品后，已选商品的总价（和优惠满减活动）会显示、勾选商品，点击结算按钮后，进去确认订单信息页面、购物车页面中，可以对添加商品信息做信息的修改，并自动保存成功、可以在购物车中重新修改商品规格、购物车能添加的商品种类是有数量上限的
- 界面测试：打开淘宝购物车页面后，页面的布局是否合理，是否完整、页面的功能按钮可以正常显示、向下滑动页面，在购物车顶端展示“购物车”
- 性能测试：打开购物车时间是否在已定的用户可以棘手的时间范围内、清空失效商品需要的时间、编辑购物车：删除、添加商品需要的时间
- 兼容性测试：操作系统、浏览器、不同的移动设备，不同的分辨率界面
- 安全性测试：用户实名认证后个人信息是否会泄露、是否需要绑定手机号
- 易用性测试：是否有全选、全不选的功能、每个商品是否有店铺名称的提示、是否有各种提示。

搜索框测试用例

- 功能测试：搜索内容为空，验证系统如何处理、搜索内容为空格，查看系统如何处理、超长字符串输入，系统是否会截取允许的长度来检验结果、合法的字符串长度后，加空格验证检索结果、是否支持回车键搜索、多次输入相同的内容，查看系统的检索结果是否一致、输入链接是否正确跳转、是否可以输入数字，英文，中文、输入

拼音也可以进行检索、点击搜索，显示搜索界面、.点击清空历史记录，搜索框是否会清空历史记

- 界面测试：查看 UI 是否显示正确，布局是否合理、是否有错别字、搜索结果显示的布局是否美观、结果数量庞大时，页面的分页布局是否合理、界面的颜色搭配是否合理
- 性能测试：搜索页面的链接打开速度的时间、搜索出结果消耗时间、弱网时搜索的响应时间
- 兼容性测试：操作系统、浏览器、不同的移动设备，不同的分辨率界面
- 安全性测试：SQL 的注入、敏感内容的检索是禁止的、特殊字符的检索
- 易用性测试：支持拍照搜索，语音搜索、搜索内容与搜索结果的匹配程度

测试的类型，以及每种类型的功能

1. 功能测试 (Functional Testing)：验证软件的各种功能是否符合需求规格说明书或用户需求，检查软件是否按照要求正常运行。
2. 性能测试 (Performance Testing)：评估软件的性能指标，如响应时间、吞吐量、并发性能等，以确保软件能够在预期的负载下正常工作。
3. 安全测试 (Security Testing)：检查软件的安全性，包括防止未经授权的访问、数据保护、漏洞修复等，以确保软件不会被黑客攻击或者数据泄漏。
4. 兼容性测试 (Compatibility Testing)：验证软件是否与不同的操作系统、浏览器、硬件设备等兼容，以确保软件能够在不同的环境中正常工作。
5. 接口测试 (Integration Testing)：测试软件的各个组件之间的交互和集成，以确保它们能够协同工作。
6. 回归测试 (Regression Testing)：在软件修改后重新运行之前的测试用例，以确保之前的功能没有被影响。

7. 用户验收测试 (User Acceptance Testing) : 测试软件是否符合最终用户的需求和期望, 以确定软件是否准备好交付。
8. 压力测试 (Stress Testing) : 模拟高负载和高并发情况下的软件性能, 以评估软件的可靠性和稳定性。

测试场景题

与开发人员存在分歧，怎么做？

1. **先检查自身，检查 bug 是否描述清楚。**如果能正确地、高质量地录入一个 Bug，那么基本上已经成功地与开发人员沟通了一大半的关于 Bug 的信息。
2. **站在用户角度考虑问题。**应该让开发人员了解到 Bug 对用户可能造成的困扰，这样才能促使开发人员更加积极地、高质量地修改 Bug。
3. **BUG 定级要有理有据。**BUG 定级时，不仅要参考 BUG 级别，还要考虑 BUG 是否会影响流程，往往用户的 BUG 级别和我们的是有区别的，需站在用户的角度定考虑定位级别。
4. **提高自身的技术和业务水平。**提高我们测试人员自身的业务和技术水平，不但要做到能提出问题，还能够提出解决问题的思路。这样才能更让人信服。
5. 最后如果实在还是不行，把问题提出来，跟开发经理和测试经理进行确认，看是否修改。

怎么判断 bug 是前端的还是后端的？

1. 前端的 bug 表现通常是**页面展示异常**、交互失效等，而后端的 bug 表现通常是**请求失败、数据错误**等。因此，可以根据 **bug 的表现**来初步判断是前端还是后端的问题。
2. 如果 bug 只出现在**特定的浏览器**中，那么很可能是**前端**的问题。因为不同浏览器对前端代码的解释和支持程度有所差异。
3. 如果在**后端日志**中能够看到异常信息，那么很可能是后端的问题。因此，在定位 bug 时，可以先查看**后端的日志**，看是否有**相关异常信息**。

4. 如果前端和后端都没有异常信息，那么可以通过**接口调试工具**（如 **Postman**）来模拟请求，检查请求参数和返回结果是否符合预期，从而进一步判断是前端还是后端的问题。

所以判断一个 bug 是前端还是后端的，需要从多个方面进行综合考虑。

测试工具有哪些？

禅道、postman、Jmeter、Selenium、JUnit

接口测试工具：postman、Jmeter

网页自动化测试框架：Selenium

单元测试：JUnit

如何排查项目问题，是前端还是后端？

1. 可以通过抓包工具来进行分析：

(1) 传参内容是否正确

如果传参内容不正确，定位为前端的 bug。

(2) 响应内容是否正确

如果响应内容不正确，为后端 bug。

2. 前后端 bug 各有什么样的特殊性质

(1) 前端 bug 特性：界面相关，布局相关，兼容性相关，交互相关。

(2) 后端 bug 特性：数据相关，安全性相关，逻辑性相关，性能相关。

3. 定位 BUG 属于前端还是后端，常用的有以下 2 种方法：

(1) 抓包看一下，查看 http 请求参数和响应结果。

(2) 查看后端服务 log 日志有无错误日志信息。

为什么选择测试开发，没有选择开发？

对确保软件质量保证是比较感兴趣的，希望通过测试和自动化来提高产品的可靠性。

测试开发通常涉及更多的团队合作和与其他部门的互动，以确保软件的质量。而开发人员可能更多地专注于编码和技术挑战。

为什么选择测试开发？

第一点：对互联网这个行业比较感兴趣

第二点：对确保软件质量保证也是比较乐意做这样的事情，希望通过测试和自动化来提高产品的可靠性。

第三点：测试开发通常涉及更多的团队合作和与其他部门的互动，需要有一定的交际能力。同时自己也有一定的测试编码能力。

如何做接口测试？

单接口测试

正向数据：也就是能正常发送请求,正常获取响应的数据，一般我们从三个方面去组织:

所有必填参数

全部参数(必填参数+选填参数)

参数组合(必填参数+某些选填参数)

反向测试：用不属于规定范围的数据区发送请求检查服务器能否正常处理

异常数据：数据为空，长度过多或者过少(边界值外)，类型不符(需要数字类型传递 str 类型)，错误的数据库

异常的参数：不传参数，少传参数，多传参数，传递错误的参数

异常的业务数据：结合业务功能考虑输出的各种异常返回情况