## 软件测试

1. 软件测试的定义以及目的
2. 软件测试分类
3. 常见面试题
4. 什么是软件？

● 软件是计算机程序、程序所用的数据以及有关文档资料的集合

答：

● 软件是计算机的灵魂。软件分为两类：系统软件和应用软件

系统软件：系统软件是生成、准备和执行其他程序所需要的一组文件和程序。如操作系统Windows，数据库SQL-Server，驱动程序，java语言系统编译环境等。

应用软件：计算机用户为了解决某些具体问题而购买、开发或研制的各种程序或软件包。如APP、微信、QQ等。

## 应用软件

C/S与B/S架构

**C/S:**client-Server：这种是我们一定要安装一个客户端才能够用的软件。

**缺点**：每次更新都需要更新服务端和客户端，比如超市的收银系统每次更新都要重装客户端，人力财力物力都很大，非常麻烦。

**B/S**:browser-server：只需要一个浏览器就可以访问服务。

**优点**：只需要更新服务器就OK了，不需要很麻烦的去更新浏览器。用户主动性比较高，比如天猫、淘宝。

## 软件测试是什么？

软件测试的定义：

● 使用人工和自动手段来运行或测试某个系统的过程。目的在于检验它是否满足规定的需求或弄清预期结果与实际结果的差别。

我们为什么要做软件测试，它的目的是什么？

1. 为了发现程序（软件）存在的代码或业务逻辑错误
2. 为了检验产品是否符合用户的需求
3. 为了提高用户的体验 用户习惯性操作+参考比较成熟的产品+性能（响应）测试

## 软件测试的分类====XX测试的概念

按测试技术/方法划分

白盒测试、黑盒测试、灰盒测试（介于白黑之间，接口）

被测试对象是否运行划分

动态测试、静态测试（文档检查、代码走查）

按不同的测试手段划分

手工测试（点点点）、自动化测试（替代手工 工具/一些代码）

按测试包含的内容划分

功能测试、界面测试、安全测试、兼容性测试、易用性测试、性能测试

按测试执行阶段划分

单元、集成、系统、验收（正式验收、AIpha验收、Beta验收）

其他测试

冒烟测试、回归测试、探索性测试（什么需求都没有，就测试软件）/自由测试（测试思维）

## 软件测试分类的说明

**白盒测试**：基于软件内部设计和程序实现的测试方法（代码层面）。不仅仅关注输入与输出的结果是否正确，同时还关注程序是如何处理的。

**黑盒测试**：就是把所有的功能和逻辑接口都放在一个盒子里，是看不到里面的逻辑走向的，只能通过盒子的外表进行测试。黑盒测试是指在测试过程中只关注输入与输出，如果输出一个测试数据的结果是正确的，就认为这个功能是正确的。也叫数据驱动测试。

**冒烟测试**：冒烟测试的对象是每一个新编译的需要正式测试的软件版本，目的是确认软件的基本功能正常，可以进行后续测试工作。

**功能测试**：测试软件的功能是否正常，通常采用的是黑盒测试方法，一般有测试人员独立执

行。

**界面测试**：简称UI测试，测试用户界面布局是否合理，整体风格是否一致，界面文字是否

正确，命名是否统一，页面是否美观，文字、图片组合是否完美等 。

**安全性测试**：测试该系统防止非法入侵的能力。 登录鉴权

**兼容性测试**：测试该系统与其他软件硬件兼容的能力（C/S架构软件--PC客户端/APP、B/S架构软件--Web，APP与C/S架构软件）。

**易用性测试**：测试软件是否易用，主观性比较强，一般要根据很多用户的测试反馈信息，才能评价易用性，参考市面上同类型比较成熟的产品。

**回归测试**：指错误被修正后或软件功能、环境发生变化后进行（开发修改）的重新测试，确认修改部分不会对其他功能产生影响。

**Alpha测试**：一种前期的用户测试，软件产品刚研发出来前期，公司内部组织员工及部分用户，模拟实际操作环境（测试环境）下进行验收测试（内侧）。

**Beta测试**：一种后期的用户测试，此时用户已经通过内部测试，大部分错误已经被修改，即将正式发行。在一个或多个真实环境下发布版本，进行测试（公测）。

## 6、笔试面试题整理

1、什么是软件测试？软件测试的目的是什么？

答：软件测试是通过人工或自动化测试软件的一个过程。目的在于检验软件是否符合规格化需求或检验预期结果与实际结果之间的差别。

2、软件测试的分类有哪些？

答：白盒测试、黑盒测试、灰盒测试、冒烟测试、安全性测试、兼容性测试、Alpha测试、Beta测试。功能测试、界面测试、易用性测试、回归测试、

1. 什么是黑盒测试？

答：黑盒测试是把逻辑和逻辑放在一个盒子里，不清楚里面的逻辑走向，只关注输入与输出。通过盒子的外表进行测试，如果输出的结果是正确的，则说明这个功能是正确的。黑盒测试也叫数据驱动测试。

1. Alpha测试和Beta测试的区别？

答：Alpha测试是一种前期测试，在软件开发完成的前期阶段。重点关注软件本身的功能模块是否能正常运行，由公司的内部组织员工和部分员工进行的测试（内侧）。

Beta测试一种后期的用户测试，当软件开发完成并且大部分错误被纠正后，把软件版本投放在一个或多个真实环境中进行测试（公测）。

# 软件的生命周期

## 软件的生命周期

软件的生命周期是软件开始研制到最终废弃不用的各个经历。

## 2.1、瀑布型生命周期模型

在1970年人类整理了第一个软件生命周期，即瀑布型生命周期模型也叫瀑布模型规定了它们从上而下、相互衔接的固定次序，如同瀑布流水，逐级下落，具有顺序性和依耐性。每个阶段规定文档并进行评审。

**问题定义及规划<>需求分析<>设计<>编码<>测试<>运行测试**

1. **问题定义及规划**

主要确定软件的开发目的及其可行性。制定项目总体开发计划。

1. **需求分析 ->产品**

在确定软件开发可行的情况下，对软件需要实现的各个功能进行详细分析，明确客户需求，输出需求规格说明书（原型图），提交评审。召开需求评审会议（开发 测试 运维 UI 配置管理人员）

1. **设计 ->开发**

把需求分析得到的结果转换为软件结构和数据结构，形成系统架构。

概要设计：主要是架构的实现，指搭建架构。表述各模块功能、模块接口连接和数据传递的实现等项事务。

详细设计：对概要设计中表述的各模块进行深入分析等，其中需要包含数据库设计说明。

1. **编码 ->开发**

按照详细设计好的模块功能表，编程人员编写出计算机可运行的程序代码。

1. **软件测试**

在设计完成后要经过严密的测试，以发现软件在整个设计过程中存在的问题并加以纠正。测试的方法主要有白盒测试和黑盒测试两种。建立详细的测试计划并严格按照测试计划执行。

单元测试：主要是测试代码，目的是确保各单元模块被正确的编译，比如具体到模块的测试，也有具体到类、函数、方法的测试等.........一般是开发人员完成

集成测试：单元测试完成后，将各单元组合成完整的体系，测试软件各单元之间的接口是否正确、数据能否正常传递。-----比如说注册和充值这两个功能是否能够联通

系统测试：把软件系统搭建起来，按照软件规格说明书中所要求，测试软件其性能功能等是否和用户需求相符合，在系统中运行是否存在漏洞等。-----根据测试用例进行完整的测试用例，测试人员的工作。

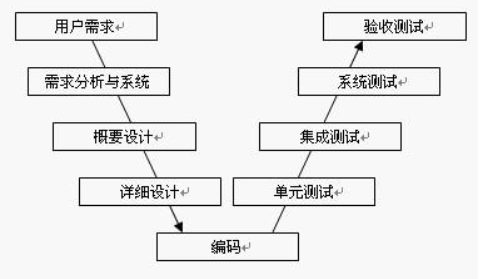
验收测试：主要是在用户拿到软件的时候，在使用现场，会根据前边所提到的需求，以及规格说明书来做相应的测试，以确定软件是否达到预期效果。------用户对软件进行验收。（正式验收测试+Alpha测试+Beta测试）

1. **运行维护**

软件维护是软件生命周期中持续最长时间的阶段。在软件开发完成后并投入使用后，由于多方面的原因不能继续适应用户的需求。需要延续软件的使用寿命，就必须对软件进行维护。软件的维护主要包括纠错性维护和改进性维护。

## 2.2、V模型

RAD（快速应用开发）是软件开发的一个重要模型，由于其模型构图形似V字母，所以又称软件开发的V模型，它通过开发和测试同时进行的方式来缩短开发周期，提高开发效率。



在开发时测试人员同时也就把测试用例给写出来了

## 2.3、敏捷开发模型

从1990年代开始组建逐渐引发广泛的关注，是一种以人为核心、迭代、循序渐进的开发方法。强调以人为本，专注于交付对客户有价值的软件。是一个用于开发和维持复杂产品的架构。就是把一个大项目分割为多个相互联系，但也可以独立完成的小项目，并分别完成，在此过程中一直处于可使用状态。

## 3.1、软件测试基本流程（重点）

测试需求分析阶段：阅读需求，理解需求，主要就是对业务的学习，分析需求点。参与需求评审会议。

测试计划阶段：主要任务是编写测试计划，参考软件需求规格说明书、项目总体计划、内容包括测试范围（来自需求文档）、进度的安排、人力物力的分配，整体测试策略的制定和风险的评估与规避措施有一个制定，一般由测试负责人编写，当然我们可能也会参与相关的评审的工作。

测试设计阶段：主要任务是编写**测试用例**，会参考需求文档（原型图）、概要设计、详细设计等文档，有不明确的也会及时和开发经理、产品经理沟通。用例测试编写完成后会进行评审。

测试执行阶段：首先搭建测试环境，执行预测（冒烟），以判定当前版本可测与否，如果预测通过，正式进入测试系统，遇到问题提交BUG进行追踪，直到被测试软件达到测试需求要求，没有重大BUG，测试结束。 ========（完善测试用例）

测试评估阶段：出测试报告，对整个测试的过程和版本质量做一个详细的评估。确认是否可以上线。

## 常见的笔试面试题

**笔试题**：

1. 生命周期模型包括哪些阶段？开发的模型是什么？

答：阶段：问题的定义及规划、需求分析、设计、编码、测试、运行维护。

模型：瀑布型模型、V模型、敏捷开发模型（以人为本）

1. 测试流程包含哪些阶段？

答：测试需求分析阶段-->测试计划阶段-->测试设计阶段-->测试执行阶段-->测试评估阶段

**面试题**： 口语化

1. 你们公司开发流程是怎么样的？

答：需求分析（详细分析了解用户的需求）-->设计（设计大模块的功能，在设计大模块中小模块的功能。从概要设计到详细设计）-->编码（在详细设计的基础上进行编码，描述实现的模块功能的数据结构、主要算法和类的层次结构等。。。。。代码层面）-->交付（软件开发完成后进行测试，测试完成后交付到客户手中）

1. 你们公司的测试流程是怎么样的？各个阶段的输出/产出物/交付物是什么？

答：单元测试（只关注底层代码，代码逻辑是否存在问题等）-->集成测试（把各单元集成一个完整的体系，测试接口是否可以正确连接）-->系统测试-->验收测试

1. 开发环境、测试环境、生产环境是什么？

**答**：开发环境：就是在开发软件的时候需要用到的辅助软件和硬件以及操作系统平台等。简称开发环境。（标准答案：指在基本硬件和宿主软件的基础上，为支持系统软件和应用软件的工程化开发和维护而使用的一组软件，简称SDE。它由软件工具和环境集成机制构成，前者用以支持软件开发的相关过程、活动和任务，后者为工具集成和软件的开发、维护及管理提供统一的支持。）

测试环境：就是在测试软件过程中所需要用到的软件、硬件、系统平台和一切与被测软件交互的软件、历史数据的总称。（标准答案：指测试运行其上的软件和硬件环境的描述，以及任何其它与被测软件交互的软件，包括驱动和桩。测试环境是指为了完成软件测试工作所必需的计算机硬件、软件、网络设备、历史数据的总称。）

生产环境：正式对外提供服务，一般会关闭错误报告而打开错误日志。可以理解为包含所有功能的环境，所有项目环境都以这个环境为基础，然后根据客户的个性化需求来做调整和修改。（标准答案：是指正式提供对外服务的，一般会关掉错误报告，打开错误日志。可以理解为包含所有的功能的环境，任何项目所使用的环境都以这个为基础，然后根据客户的个性化需求来做调整或者修改。通俗的讲，项目数据前端后台已经跑通，部署在服务器上之后，有客户使用，访问，就是网站正式运行了。）

1. 你在测试环境后台添加的数据和信息能在生产环境中看到吗？---作业，自行了解

答：不能。生产环境是用户使用的正式环境（数据不能动），而测试环境是开发之后进行测试的，数据可以随便填。生产环境的版本低于测试环境，测试环境的版本低于开发环境。（标准答案：开发环境一般指的就是本地环境，用开发工具的服务器带起来的环境测试环境就是部署到测试服务器上，让测试人员对其测试的一套环境(比开发环境版本低，数据随便填)生产环境就是客户正式使用的环境(比测试环境版本低，数据绝对不能乱动。）

## 课前思考：

1. 什么是软件测试需求？
2. 软件测试需求的必要性？
3. 如何进行软件测试需求分析？

## 测试需求是什么？

测试需求主要是解决“测什么”的问题，一般来自需求规格需求说明书中原始需求。测试需求应全部覆盖已定义的义务流程，以及功能和非功能方面的需求。

业务流程（包含一系列的操作/功能）：例（发红包）：登录-->好友-->发红包-->支付密码

2.2.1、功能测试需求

2.2.2、性能测试需求

## 2、为什么需要软件测试需求？

简而言之：**只有明确了测试需求才能知道怎么去测试、什么时候开始测试、由多少人测试、在什么环境下测试。**

## 笔试面试题

面试题

1：遇到隐形需求怎么办？需求文档没有明确说明

充分熟悉业务流程+参考同类型比较成熟的产品：跟产品/开发沟通明确

2：给你一个logo的水杯，你会如何去测试？

3：你会如何去测试朋友圈，购物车等熟知的软件产品。（支付，优惠券，二维码）

# 测试用例设计方法

## 本课程目标

## 等价类划分法

1. **等价类划分法的概念**

等价类划分法是一种典型的、重要的黑盒测试方法，是指某个输入域的子集。在该子集中所有的输入数据对揭露软件中的错误都是等效的

等价类划分**有效等价类**和**无效等价类**两种。

**例：**微信红包，在0.01~200区间内且小数点后面不超过2位小数为有效等价类；其它则为无效等价类。

1. **等价类划分法用例设计原则**
2. 划分等价及无效等价类，为每一个等价类规定一个唯一的编号
3. 设计一个新的测试用例（数据），使其尽**可能多的覆盖**尚未被覆盖的**有效等价类**，重复这一步，直到所有的有效等价类都被覆盖为止；
4. 设计一个新的测试用例（数据），使其**仅覆盖一个**尚未被覆盖的**无效等价类**，重复这一步骤，直到所有的无效等价类都被覆盖为止。

2、边界值分析法

1、定义：边界值分析法是对等价类划分法的一个补充，边界值一般都是从等价类的边缘值去寻找。边界值分析的基本思想：**正好等于、刚刚大于、刚刚小于边界的值**作为测试数据。

**注意**：0是一个特殊值，我们在考虑边界值的时候同时也要考虑这个特殊值。负数

2、边界值的作用：人们从长期的测试工作经验得知，大量的错误是发生在输入或输出范围的边界上，而不是在输入范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。

边界值的应用场景：如果需求规定了取值的范围或规定了取值的个数时，可利用边界值进行测试

## 等价类划分法/边界值分析法常见运用场景

1. 输入条件规定的取值范围或值的个数的情况（类似最小<x<最大、最小<x、最大>x）；

比如用户名长度、红包金额数值输入范围

1. 在输入条件是true和false两种情况下；比如勾选、开关设置
2. 在下拉列表包含多个选项的情况；比如城市下拉选项（第一个、中间一个、最后一个）+常用选项。
3. 在规定了输入数据必须遵守的规则情况下，可确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）；

以上除了等价类之外同时会涉及边界值的分析。边界值还包括以下：

1. 报表数据的第一行、中间一行、最后一行
2. 屏幕上光标在最左上、最右下位置

## 场景法

1. 什么是场景法？ ----基于用户操作使用立场设计

通过场景描述的业务流程（业务逻辑），也包括代码实现逻辑，设计用例来遍历场景（路径），验证软件系统功能的正确性。

1. 如何使用场景法

2.1、画出流程图

矩形：表示步骤（操作、输入和输出结果）

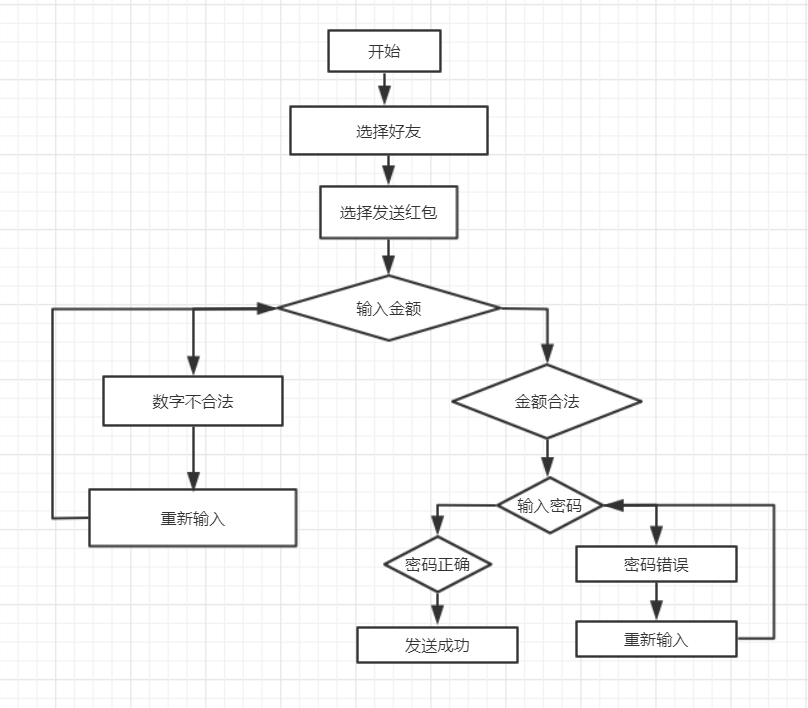
菱形：判断是否

箭头：流向

2.2、场景进行用例覆盖

注意：场景法的重点是测试流程，因此每个流程一个用例验证即可，流程测试没有问题并不能说明系统功能没有问题了，还需要针对单步的功能进行测试，只有单个功能点和测试流程，才算是充分的测试。

发红包：选择好友-->点击发红包-->输入金额是否合法-->输入密码，确定-->发送红包成功



## 4、错误推测法（反推法）

基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法。它的要素共有三点，分别为：经验、知识、直觉。

**笔试面试题整理**

1.编写测试用例会用到什么方法？

你觉得你在写用例的时候用到了吗（结合项目来回答）

答：等效划分法、边界值分析法、场景法、错误推测法

1. 笔试==用例设计大题，刷题柠檬班APP

# 测试用例编写&评审

本节课的目标：

1：什么是测试用例？

2：测试用例的重要性？

3：测试用例的八大要素（重点）

4：测试用例评审

## 什么叫软件测试用例

什么是测试用例

测试用例（TestCase）是为项目需求而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序是否满足客户需求。

可以总结为：每一个测试点的数据设计和步骤设计

1. **测试用例的重要性（了解）**
2. 测试用例是软件测试的核心。

软件测试的重要性是毋庸置疑的，测试用例是测试工作的指导，是软件测试质量稳定的根本保障。影响软件测试的因素有很多，如软件本身的复杂程度、开发质量、测试方法和技术的运用，但有些因素是客观存在不可避免的，如IT团队的流动、环境和情绪的变化等。

1. 评估测试结果的基准

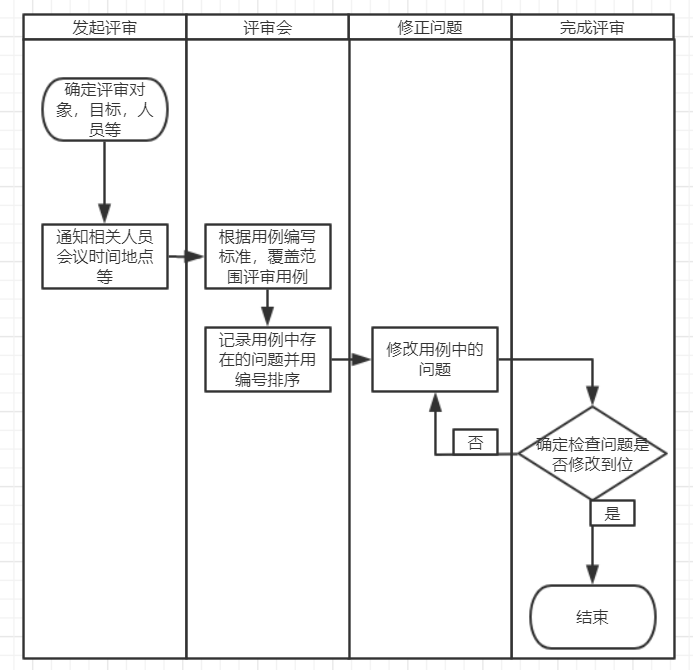
测试用例的通过率和错误率是测试结果的一个重要依据，用以判断该软件的测试结果是否通过，能否达到上线标准。

1. 保证测试的时候不遗漏一个功能点。可以在测试人员疲累的时候起到一个牵引作用。
2. 在编写测试用例的过程中，可以熟悉需求、对系统的架构和业务流程有一个整体的深入的了解。
3. 好的测试用例不仅能方便自己和他人查看，还能帮助设计的时候考虑的更周全，因此测试用例的写作和设计一样同样重要。执行性（指导性）。

## 测试用例的八大要素

1. 用例编写：产品名\_测试阶段\_测试项\_xxx 用英语来命名
2. 测试项目：对应一个功能模块（细分功能）
3. 测试标题：直接对测试点进行细化得出，输入内容+结果，同一功能模块标题不能重复（来自测试点）
4. 重要级别：高/中/低 项目包含功能模块--需求文档--功能列表
5. 预置条件：需要满足一些前提条件，否则用例无法执行
6. 测试输入（数据）：需要加工的输入信息，根据具体情况来设计（跟步骤结合起来一定要具有指导性意义）
7. 操作步骤：明确给出每个步骤的描述，执行人员可以根据该步骤完成执行工作
8. 预期结果：根据预期结果与实际结果对比，来判断被测对象是否符合需求（预期结果唯一，不能出现“是否或者”）
9. 实际结果：

## 4、测试用例评审



## 5、测试用例的变更

由于需求变更、对于业务的不断深入了解和测试评审用例，测试用例是无法一次性全部写好的，所以测试用例在全部完成后是需要不断修改的。

测试用例变更通常包括：

1. 需求变动
2. 执行完成后的用例完善
3. 评审后的用例修改

ps:一定要记得备份

## 笔试面试题整理

1：用例需要评审吗？紧急情况用例也需要评审吗？

2：如果被测项目很紧急，来不及写用例，怎么办？

3：遇到隐形需求如何写用例？（需求不明确）尽可能熟悉产品业务挖掘需求，参考同类型产品；整理成需求列表，找产品确认。

4：用例有没有优先级？如果一定要有优先级，依据什么来确定了？

5：如何编写测试用例？（以项目为基础来讲一个小模块用例设计）

正常情况，异常情况，长度；举例项目例子

# Bug的管理流程&禅道流程

课程目标

1：bug的类型

2：bug的等级

3：bug的生命周期（重点）

4：禅道的使用（重点）

5：常见的面试笔试题

6：课后安排

## bug的定义

软件的bug。狭义概念是指软件程序的漏洞和缺陷；广义概念除此之外还包括测试工程师或用户所发现和提出的软件可改进的细节、或与需求文档存在差异的功能实现等。

我们的职责是就是发现这些bug并提交给开发人员，让开发人员去修改。

## bug的类型

要确定一个bug类型，需要对项目（或产品）有比较深的理解。这个划分对于开发定义的问题逻辑影响小，但对于问题类型的统计就比较重要了。

常见的bug类型划分（禅道系统为例，可自定义）：

代码（功能）错误

界面优化

设计缺陷

## bug的等级

bug等级，这个划分有分三级和四级，也有分五级的。如果等级越高，那么修复的等级也会高一些，有些公司还会根据你提交的bug等级和bug数量来考核绩效。大多数情况下提交bug等级差不多即可，没有严格区分。

1. 致命错误：
2. 常规操作引起的系统崩溃、死机、死循环、闪退
3. 造成数据泄露的安全性问题，比如恶意攻击造成用户私密性信息泄露
4. 涉及金钱计算
5. 阻断性bug，大部分测试工作进行不下去
6. 严重错误：
7. 重要功能不能实现
8. 错误的涉及面广，影响到其它重要功能正常实现；
9. 非常规操作导致的系统死机、崩溃、死循环、闪退
10. 外观难以接受的缺陷
11. 密码明文显示；（界面+数据库）
12. 偶现的致命性bug
13. 一般错误：

不影响产品的运行、不会成为故障的起因，但对产品外观和下道工序影响较大的缺陷

1. 次要功能不能正常实现
2. 操作界面错误（包括数据窗口内列明定义、含义不一致）
3. 查询错误、数据错误显示
4. 简单的输入限制未放在前端进行控制
5. 删除操作未给出提示
6. 偶现的严重性bug
7. 细微错误：

程序在一些显示上不美观，不符合用户习惯，或者是一些文字的错误

1. 界面不规范
2. 辅助说明描述不清楚
3. 提示窗口文字未采用行业术语
4. 界面存在文字错误：

改进建议：可以提高产品质量的建议，包括新需求和对需求的改进

## 4、bug的生命周期（管理流程）

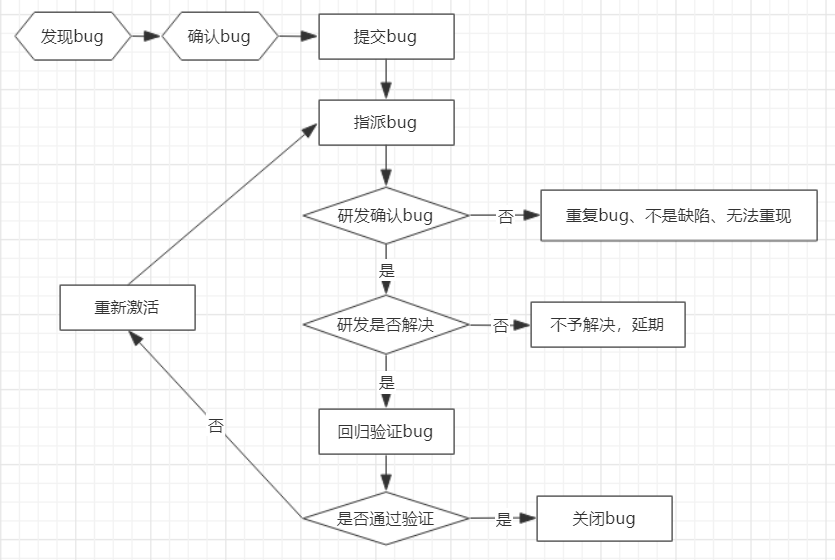
正式面试/笔试过程中经常会被问的问题。Bug的生命周期就是一个bug从被发现到关闭的一个过程（状态）。

生命周期中一般缺陷状态：新建（提bug）-->指派-->已解决（bug修复、不予解决、延期、不是缺陷、重复bug）-->待验-->关闭

如果待验的bug在验证时没有解决好，我们需要重新打开（激活）-->指派-->已解决-->待验，循环这个过程

中间其他的状态：拒绝、延期等

## 5、bug的跟踪管理流程



## 6、bug的跟踪管理-状态处理

1、**已经指派的bug**------已经指派给开发，请大家注意自己的bug走向，随时关注并进行追踪！如果一直未修复，提醒开发修改，以免开发忘记；如果已经修复等待测试环境更新后进行验证。催着修bug。

2、**已解决的bug-**-----等待测试环境更新后进行验证，验证通过则关闭，未通过则重新指派给开发。

3、**重复bug**-----先去查看下是否跟开发指定的bug重复。如果是则关闭，不是则说明原因，重新激活打开指派给开发。

4、**不是缺陷**--------在此确认需求，查看是否是bug，如果仍然觉得是则跟开发沟通，列举出觉得是的点，沟通未达一致则找产品确认，是bug注明情况重新指派给开发，确认不是则关闭bug，但是记录下来等产品开发完成后在研究。

5、**无法重现**----确认开发环境和测试环境是否一致。包括操作步骤、浏览器、环境、特定账户、输入数据等，如果多个测试版本验证后，如开发所说重现不了，依据bug的严重程度跟产品、开发一起确认关闭；如果找到重现原因，注明清楚并在此指派给开发。

6、**不予解决**------找产品经理进行确认。确认不予解决进行关闭；确认需要解决请备注原因并打开指派给开发。

7、**设计如此**-------找产品经理进行确认。确认设计如此进行关闭；确认是问题，备注原因重新指派给开发。

8、**延期修改**------查看bug的严重程度是否会影响产品的发布。与产品经理进行确认，严重则注明备注提交给开发，不是则记录好笔记。

## 7、bug的跟踪管理-缺陷管理工具

常见的缺陷管理平台：

禅道（zentao），柠檬班用的就是禅道，项目管理工具--敏捷开发流程

**禅道的密码：**

Bugzilla、jira：都挺不错的。但是搭建起来很困难

Bugfree

Readmine

Easybug

Mantis

QC（QualityCenter）、TD

不管是开源还是其它的商业的缺陷管理工具，它们本质都是一样的，用于管理bug的生命周期，只要掌握其中一种工具自然而然就会使用其它的，稍微有些许的区别只要别人稍加指点就会用了。

## 8、bug的管理流程-如何提交bug

发现bug后，接下来你提交到bug管理平台，提交一个bug包含哪些内容？

Bug标题——标题要清晰简洁，**写明bug描述**；如果没有选择功能模块，最好在标题中标注功能模块。让查看bug的工作人员能清楚的指导你表达的意思，**bug的功能模块+bug的操作+bug的结果。**

重现步骤——详细记录发现bug的过程。能指导开发人员重现这个bug。附上测试数据

实际结果——出现bug的结果，粘贴bug截图、日志截图

预期结果——记得写预期结果

Bug类型和严重程度——便于后续测试结果分析，bug的统计

Bug测试环境——例如：什么系统、哪个版本。兼容性问题、难以重现问题

附件——日志文件，文件**测试数据**。出现bug视频。图片、崩溃日志文件等

## 常见笔试面试题

1：有没有你印象深刻的bug？Bug的原因/bug当时是怎么解决的？ 作业：提交一个bug

2：bug的生命周期？

3：当你发现了一个bug，但是开发人员不认为是bug，如何处理？

4：你在发现bug并确认bug的过程中，对于复现率不高的bug怎么处理？ 提交+追踪

# 测试计划编写

## 课程目标

1. 软件测试计划介绍
2. 测试计划编写

## 软件测试计划简介

为什么编写软件测试计划？一般是由主管来写，测试计划是在做完需求分析后，整个测试工作开始之前做的一些准备计划工作，“5W+1H”去记忆，一般包括以下内容：

**目的（why）、测试范围（what）、测试进度安排（when）、测试人员（who）、测试环境（where）、测试方法+测试工具（how），风险评估**

# 测试报告编写

## 软件测试报告简介

软件测试的五个阶段：评估阶段（测试人员必须会的）

公司一般都有自己的软件测试报告模板

在进行完整的软件测试工作后，需要对测试产品做一个总结：

其中需要包括**测试范围、测试环境、测试用例覆盖率有多少，bug的统计与分析，遗留的bug有哪些，版本测试评估，发布的建议。**

## SVN的功能使用介绍

1. 安装好tortoise SVN。重启电脑
2. SVN是什么？

一个版本管理工具，常用于软件开发项目中，实现代码、文档等的历史版本保存、共享和权限管理，这里我们今天要说的是文档的管理。

1. 如果公司要你用SVN，第一件事应该做什么？

其实很简单，就是要有一个账号以及一个相关的URL地址权限！账号从哪里来？问老大，发消息/邮件给配置/项目管理员申请账号及相关权限。然后才能从SVN服务器拉取文档信息。

1. 如何拉取文档数据？

在你存的目标位置，创建一个文件夹，然后点击SVN CHECKOUT

输入SVN的URL实时更新

SVN://120.79.176.157/lemon67

用户名：lemon67 密码：lemon2017!@

# 浏览器兼容性测试知识

● 网站一把都需要做兼容，用户使用量+内核来看，选取主流浏览器，IE、Chrome、Firefox、Safari、QQ

一般的兼容性测试是怎么做的？

主要是页面的格式、字体、输入框、下拉框、复选框、按钮等的检查；页面显示正常的功能进行检查。

# Web项目实战

1. 需求评审 ----第一节

------>SVN获取目录需求文档，提前研读需求，整理疑点

------>参与需求评审会议，排除疑点

# APP测试要点提取与分析

## 课程目标（注：与web测试基本相同，除了标红的点）

功能性测试

1. 安装与卸载
2. 软件更新升级
3. 登录测试
4. 离线测试
5. 消息推送

UI界面测试

兼容性（操作系统、屏幕尺寸、分辨率）

安全性测试

接口测试

中断测试

网络测试

性能测试

## 功能测试

根据产品需求文档编写测试用例，执行测试

APP客户端的单个功能模块

需要使用等价类、边界值，考虑正常和异常情况

依据功能业务逻辑考虑功能交互

## 安装与卸载测试（**正常+异常**）

**软件安装前：**

空间不足是否有相应的提醒

**软件安装中：**

安装过程是否可以取消

安装是否可以正常运行

**软件安装后：**

是否可以卸载应用：1）通过桌面卸载2）通过软件设置卸载

常见bug：发现在IOS手机上有个应用安装时没有完全安装，终止安装后，未完成安装的应用图标一直显示在手机桌面上，并且无法成功删除。

安装后是否自动删除安装包

卸载之后，重新安装

重复安装：1）提示版本已经存在2）直接覆盖安装，但是能安装使用

## APP升级测试

**升级之前：**

当客户端有新版本时，是否有更新提示；

**升级之后：**

更新后各个功能（新/旧功能）是否能正常使用

更新后老数据存在并正常 --版本兼容性

**非强制升级版** --开发沟通确认

用户可以取消更新，老版本正常使用，用户在下次启动APP时仍然正常出现更新提示；

**强制升级版：**

用户没有做更新时，退出客户端，下次启动APP时，仍出现强制升级提示。

**在线跨版本升级：**

升级后正常使用

选择的版本--用户使用量最多的版本--主流版本

## APP登录测试---安全

登录方式：

用户名、指纹、二维码、面部识别、手势、语音、短信验证码、第三方登录（微信、QQ、微博）、一键登录是否支持。

未登录用户：

一些页面的操作，是否做了控制；点击某些资源，提示登录或者跳转登录页面。

用户主动退出登录后，下次启动APP后应该进入的是登录界面。--参考成熟产品

切换账户登录：检验登录的信息是否做到了及时更新

登录控制：通过功能限制只能在单个客户端或者指定个数的客户端进行登录操作

不允许多点的登录时是否将原用户踢下线，并且给出提示信息

允许多个客户端登录时，提示信息；且确保数据库操作无误，每个客户端可以及时看到数据库的更新。

**用户登录时间过长，账户信息会过期-----开发确认超时时间**

虽然是登录状态，系统会提示“用户没有登录”

强制退出并提示：账户信息过期，请重新登录

**单点登录：**

一个系统登录了账号，在跳转到别的页面就不需要重新登录了，比如淘宝里面有天猫，登录了淘宝就不需要在登录天猫而直接跳转到天猫。----系统内信息是互通的

## 触屏及操作测试---可以不支持，但不能有异常

● 触屏快捷手势：两指、三指滑动

● 长按、短按屏幕

● 手机横屏、竖屏测试

● 同时触摸不同的位置，同时进行不同的操作---异常、闪退

查看客户端的处理情况，是否会crash（崩溃、闪退），ANR（无响应，Application Not Responding）

## APP消息推送测试

**开关测试：**

默认全部打开状态，客户端可以接收到推送信息

设置开关可以打开、关闭；设置开关关闭时，客户端接收不到消息推送。

**手机客户端未锁屏时：**

APP应用后台运行，消息推送是否可以正常接收，并且是否可以正常点开查看。

APP应用前台使用，消息正常接收且可以点击查看

**手机客户端锁屏时：**消息是否可以正常接收

**登录状态：**

**退出登录时**是否可以接收push推送（根据客户需求）

**重新登录时**是否可以接受多条消息推送 红点/条数

**切换用户时**检查收到的推送信息是否与用户信息相符，没有错误的将别人的推送消息推送过来

**消息栏（通知中心）：**是否可以接收到消息提醒，且点击可查看。点击后消息栏消失

## UI界面测试

● 确保产品UI符合产品经理制定的原型图与UI设计效果图/切图一致

● 依据经验、用户使用习惯、参考其他成熟的产品，界面可优化的bug

一般设计界面（如菜单、对话框、窗口和其他可视化控件）布局、风格、文字是否正确，页面是否美观，操作是否友好。

如：安装APP后的加载页/动态视频显示，分享页面的产品logo显示。

**注意：APP测试和web测试的思路基本一致**

## 兼容性测试---适配



## APP中断测试

● APP被手机行为打扰的情况下：APP能否正常处理，保证数据的正确性。----运行、数据

● 主要对于核心功能存在实时数据交换的页面进行中断测试，除了保证中断过程中有合理的处理；还需确保中断结束后恢复正常。

来电、来短信、锁屏解锁、断网重连、断电、低电量提醒、前后天切换、APP切换

手机硬件上：待机、插拔数据线、耳机线、闹铃弹出框提示等。

● 常见bug场景

爱奇艺视频正常播放过程中，微信通话/电话中断：

**正常：**APP暂停状态；通话结束后，恢复播放（直接恢复或者手动点击恢复）

**异常：**APP卡死、音视频不同步

微信视频聊天，低电量提醒中断：（电话中断）

**正常：**聊天不中断，关闭提示信息后继续通讯

**异常：**APP卡死/崩溃，微信聊天被强行断开

## 网络测试

● 测试2G/3G/4G/5G/WIFI/热点，网络的切换

例如从WiFi环境切换到4G环境中是否有提示启用4G网络，会产生扣费，是否有提醒

● 测试有网无网切换下应用的运行

有网到无网在到有网环境时，数据是否可以自行恢复，正常加载（网络中断重连）

无网络时，各种提示信息是否友好，数据本地化是否正确（比如提示当前已断开网络，请检查网络设置）

● 弱网测试（延时+丢包），关注弱网场景下超时是否有合理提示，且是否有重发机制

提交数据是否一直处理提交中，是否会有延迟，数据交换失败是否会有提醒

数据多次提交（支付宝），是否仅执行一次

最大尝试次数，APP是否能正常工作

## 安全性测试--权限测试



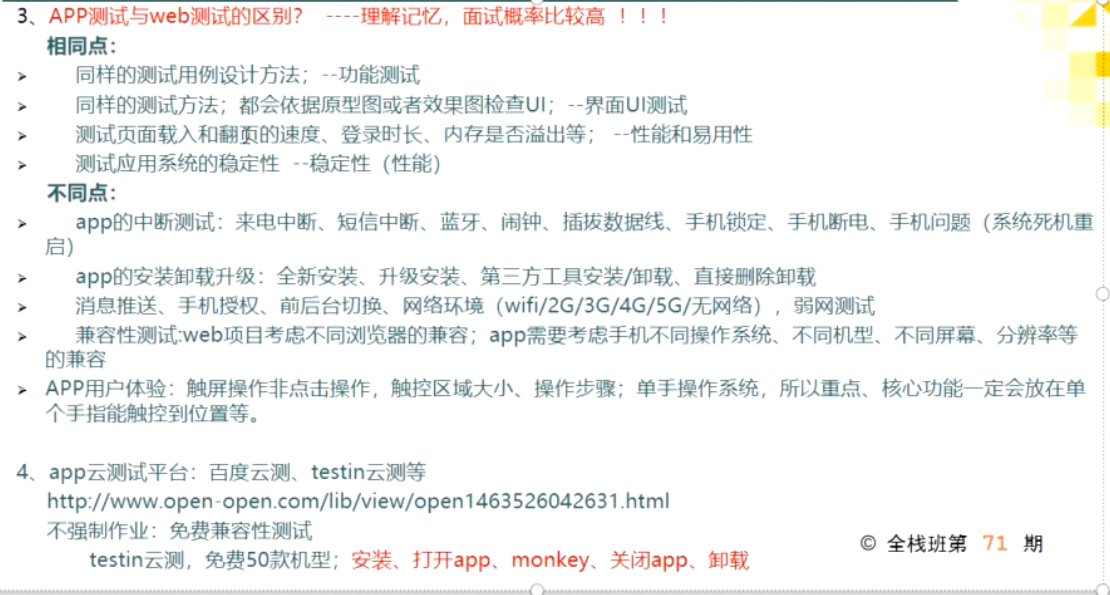
## 性能测试



## 一些关于APP测试的问题



## APP测试与web测试的区别



# 微信小程序测试要点

## 本节课目标



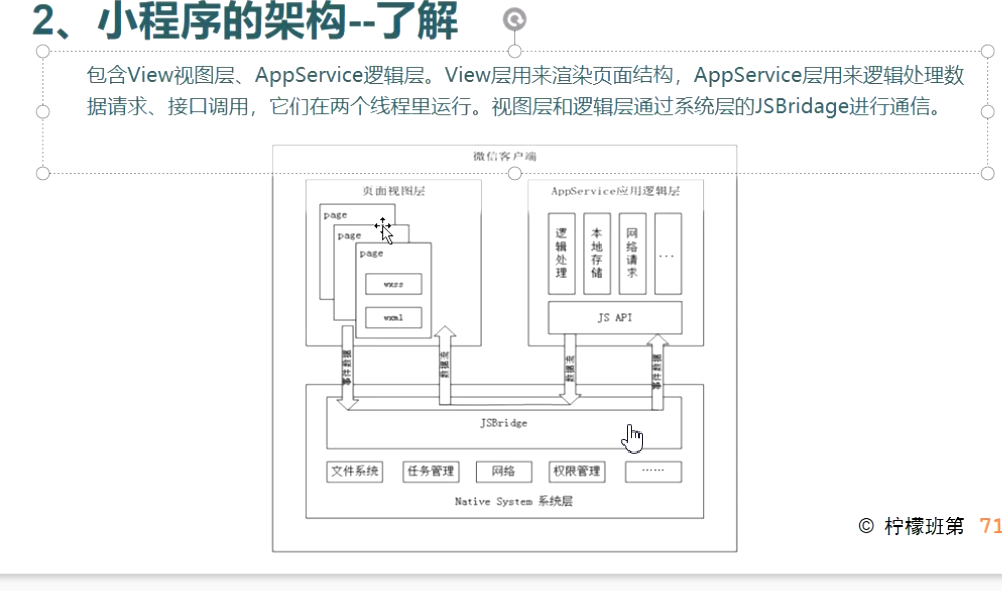
## 小程序是什么？

英文名Mini Program，是一种不需要下载安装即可使用的应用，是内嵌在微信APP里面的一款新形态软件。

小程序产品的版本类型分为：开发版、体验版、正式版

正式版是需要微信审核的。开发版和体验版是不需要审核的，只需要给指定的微信号权限，然后扫小程序二维码就能访问。

## 小程序的架构---了解



## 小程序的测试要点

**功能测试、权限测试、UI页面测试、兼容性测试、接口测试、网络测试、性能测试**

## 3.1、功能测试

功能测试和APP的功能测试一样。-----测试点，测试思维 测试方法一样

根据设计好的各大类功能模块划分，然后在逐级细化，覆盖到每个功能尽可能全面的测试点。包括业务流程、数据流向、功能入口有效性检查、交互性检查测试。

**注意：小程序除了本身功能的交互、它还和微信客户端有着交集。**

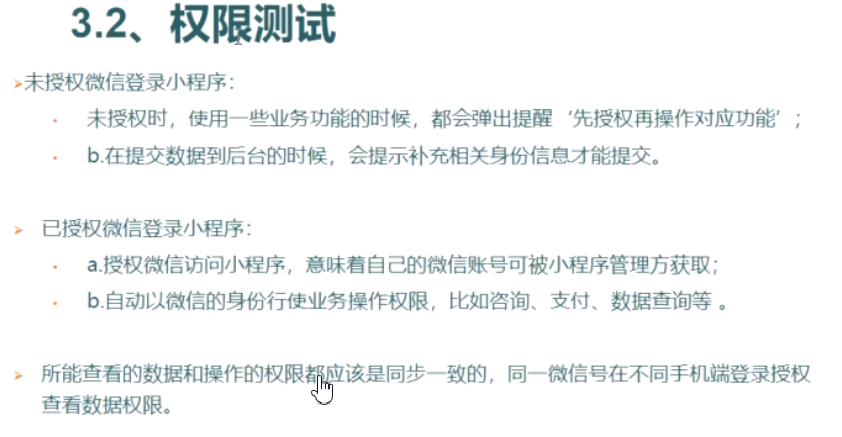
A:入口 ----主流入口 -产品、开发

B：关联交互：比如小程序支持交易，那么它与微信的钱包，卡包，是有联系的，需要测试

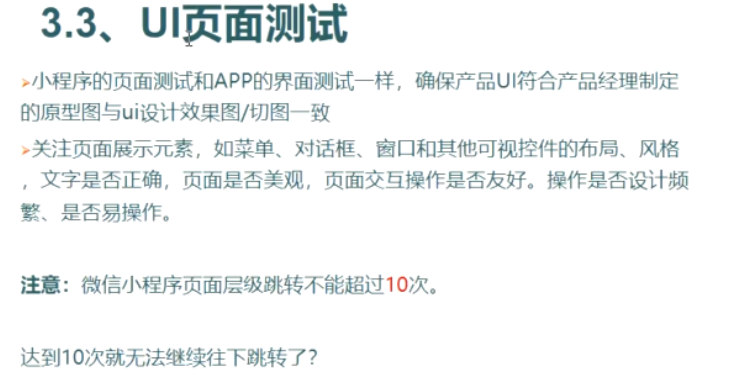
## 小程序的入口



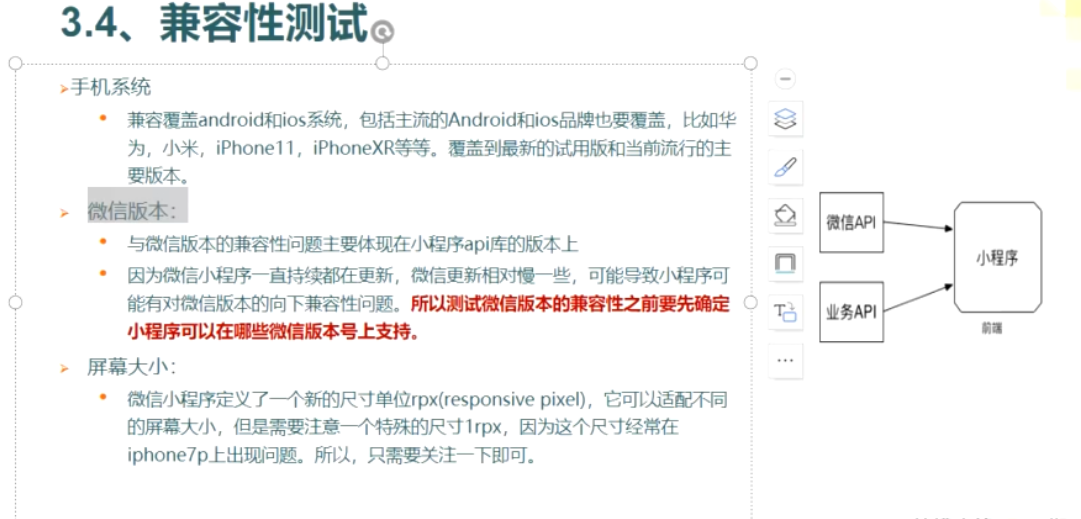
## 3.2、权限测试



## 3.3、UI页面测试



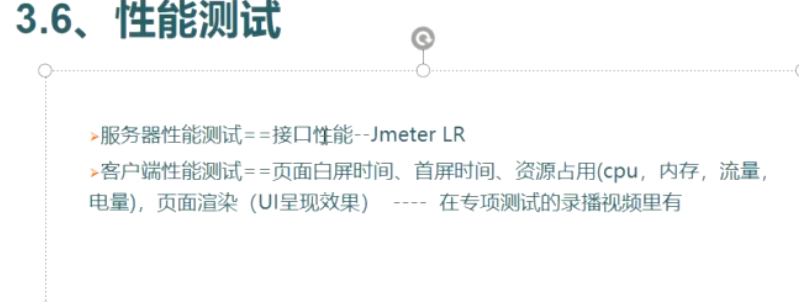
## 3.4、兼容性测试



## 3.5、网络测试



## 3.6、性能测试

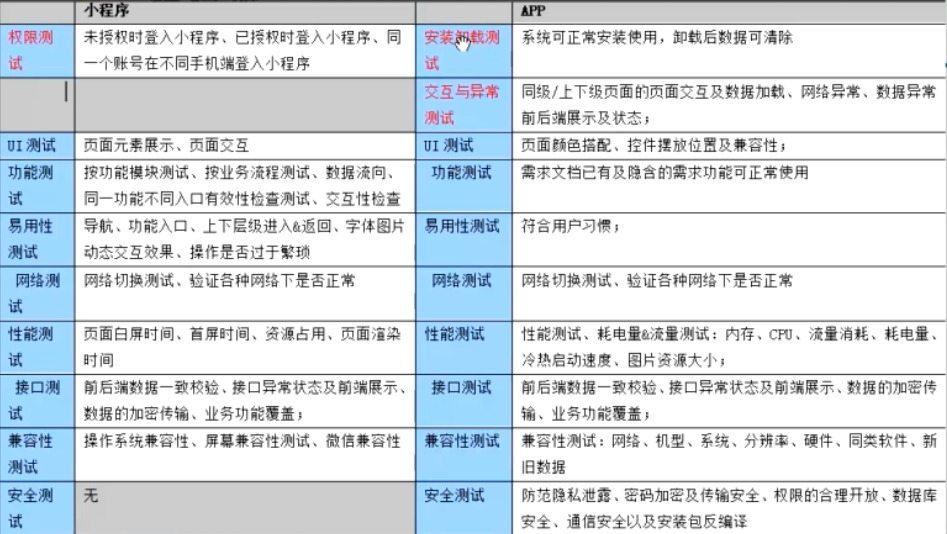


## 常见的笔试面试题

1. **小程序怎么做兼容性测试**

答：从微信版本和系统版本出发，因为小程序随时都在更新，但是微信更新相对较慢，所以有可能导致小程序向下兼容，在测试小程序兼容性问题之前要先确定小程序在哪些微信版本上可用。

1. **小程序测试和APP测试的区别 ---中断也要测**



# H5测试要点

## H5是什么？



### H5、Native APP、混合APP（区别和形态）-----重要

**H5:**

● 开发快，实现的功能也很炫；

● 仅编写一次，可以在多个设备上到处运行，易用性；--无需安装

● 具有局限性，如：会话管理、安全离线存储、访问原生设备功能（摄像头、日历和地理位置） --比较依赖于网络

**Native APP：**

**●** 原生的应用程序看起来（界面）和运行起来（性能）都是最佳的；能更好的适应各种功能的实现。----缓存

● 开发和测试的周期较长，开发比较耗时耗力

● 必须登录到应用商店、下载新版本，才能获得最新修正版

**H5+原生应用混合 --目前市面上主流的APP形态**

●对某些固定格式和控件、页面反应速度要求高的模块就采用Native开发

● 对于新闻、大段文字、资讯的用H5标准实现页面来加载，将其嵌套进原生的框架中

● 混合应用程序可以将HTML5嵌入到一个细薄的原生容器中，集原生应用程序和HTML5应用程序的优点于一身。

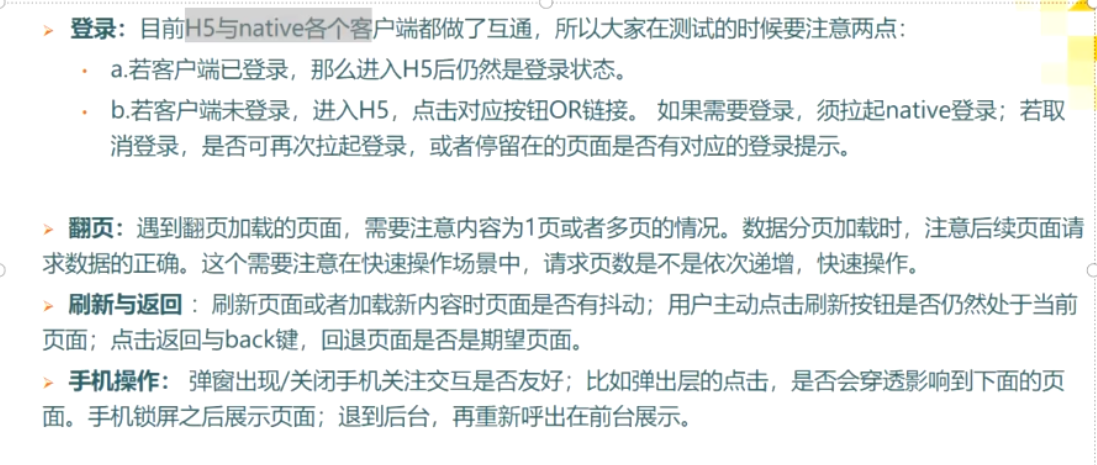
## H5的测试点

**功能测试、UI测试、兼容性测试、安全测试、接口测试、网络测试、性能测试**

## 2.1、功能测试

基本的功能测试和APP测试一样

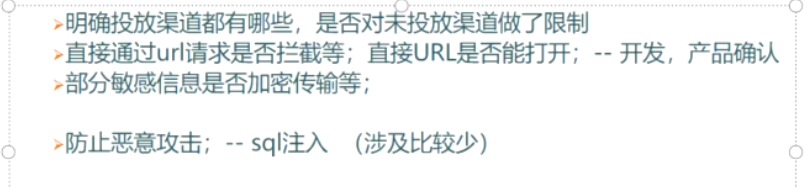
另外关注：登录、翻页、刷新、返回、手机操作



## 2.3、兼容性测试



## 2.5、安全性测试



## 2.7、性能测试----肉眼可见的性能体验

服务器性能测试==接口性能

客户端性能测试

加载速度：页面的首屏时间、白屏时间、有缓存/无缓存的加载时间

资源占用：CPU、流量、电量、内存等的占用

关注页面中有图片的话，尽量缩小图片；

资源是否压缩；

对于一些不会变化的图片，不需要每次都请求的东西，做本地缓存；

数据较多时是否做了分页加载处理；

反复访问，检查是否占用了大量内存。

## 常见的笔试面试题

1. **H5测试点有哪些？**

**答：**功能测试、UI界面测试、兼容性测试、安全测试、性能测试、网络测试、接口测试。

1. **H5和native的区别？**

**答：H5**易开发，仅开发一次，功能界面炫酷，易用性强；缺点在于局限性很大，如离线缓存，会话管理，访问原生设备（摄像头），比较依赖于网络。**Native**的界面和性能都是最佳的，能适应更多类的应用，原生软件的开发周期都比较长，比较耗时耗力，而且必须到应用商店去下载最新版本才能获得最新的修正版。

# APP系列课程

## 1、adb连接真机



## 2、adb常见命令





Android7.0以下：adb shell dumpsys activity | find “mFocusedActivity” --查看前台应用

Android8.0上需要使用： adb shell dumpsys activity | find “mResumedActivity”

## logcat日志文件

Android日志系统提供了记录和查看系统调试信息的功能，日志都是从各种软件和一些系统的缓冲区中记录下来的，缓冲区可以通过logcat来查看和使用。

使用方法：

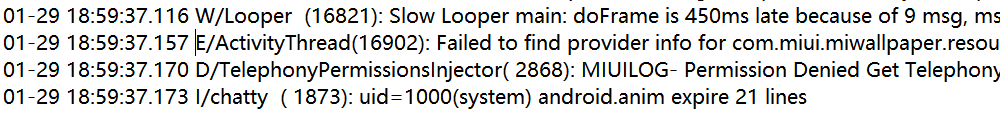
[adb] logcat [<option>] ... [<fiter-spec>]..

logcat是以如下开头的：

1. -----------beginning of XXX

开发者选项，有个选项叫做“日志记录器缓冲区大小”，默认是256K，日志是循环写入环形缓冲区的。一般情况下，写满是最旧的日志会被删除以给新输出的日志留内存空间——真机设置，模拟器一般没有。

## logcat日志文件分析



由下面5个部分组成：

1. 写下日志的时间，如上图中的01-29 18:59:37.116。
2. 优先级，在Android中，日志的优先级从低到高分以下几种：

**V**——Verbose（最低级别，开发调试中的一些详细信息，仅在开发中使用，不可在发布产品中输出）

**D**——Debug（调试，可以在发布产品中关闭，比较常见）

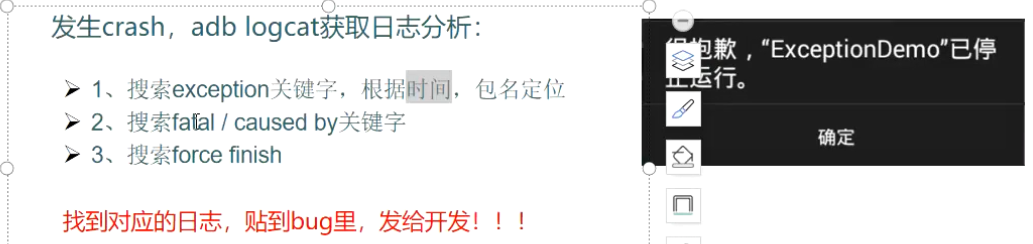
**I**——Info（信息，一般提示性信息）

**W——**Warning（警告）

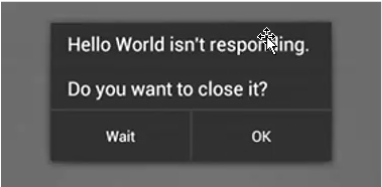
**E**——Error（错误，已经发生可影响运行的错误，比如应用crash时输出的日志）

1. 标签（tag）表明日志的发起者和方便日志的筛选，如上图中的ActivityThread，类和模块
2. PID（进程ID），如上图中的16902
3. 正文：本日志的主体内容

## 5、发生crash如何分析



## 何为ANR？

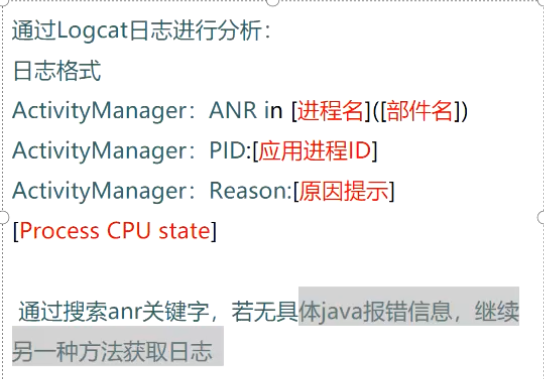
ANR全名Application Not Responding，也就是应用无响应，当操作一段时间内系统无法处理时，系统会弹出右图式的ANR对话框。

为用户在主线程长时间被阻塞时提供处理交互，提高用户体验

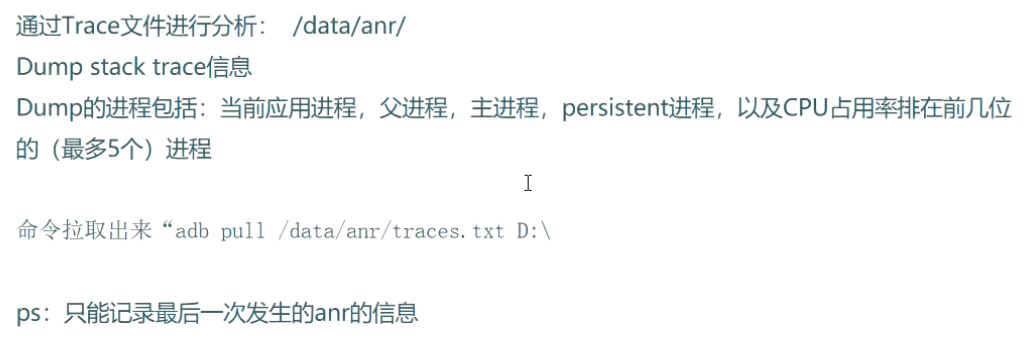
Android系统的一种自身检测机制

## 6.1、发生ANR如何分析

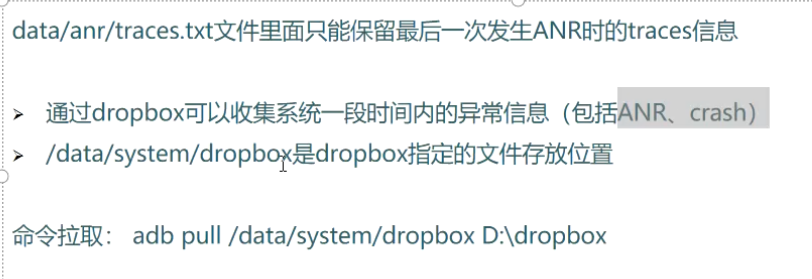
**ANR分析01：**



**ANR分析02：**



**ANR分析03：**

 adb问题补充讲解（重）

**真机：---adb驱动**

1. error：cannot connect to daemon
2. 、查看手机是否有手机助手、豌豆荚之类的软件，会占用5037端口---adb默认端口
3. 、使用**netstat -ano | find 5037**查看5037端口对应的进程号（因为5037是adb 默认的端口号），然后用**taskkill -f -pid 11100** -----进程号即可被杀死

**adb connect ， out of data -模拟器**

1. 、替换文件---adb.exe，夜神

进入目录D:\Android\_ADT\adt-bundle-windows-x86\_64-20140702\sdk\platform-tools下，把adb.exe拷贝一份覆盖夜神nox/Nox/bin目录下的adb.exe，另外D:\Android\_ADT\adt-bundle-windows-x86\_64-20140702\sdk\platform-tools下的adb.exe拷贝一份，重命名为nox\_adb.exe，替换夜神bin目录下的nox\_adb.exe。

1. offline：
2. 、adb kill-server，杀掉进程，重新连接 --adb connect
3. 、模拟器必须启动
4. 、替换文件
5. 黑屏问题，安装最新版本的夜神：<https://www.yeshen.com/>
6. 真机测试，保证步骤跟PPT上的一致以外，https://www.cnblogs.com/zymnstlm/p/12381619.html，安装驱动
7. 、权限：/data---模拟器权限低，高#--root,$--p普通用户 ====sdcard目录是有权限的，cd sdcard == 注意：新建文件；
8. 、电脑C盘：adb logcat -v time >D:\ ==拒绝访问

## OA--敏捷开发流程-实战流程

**1：需求评审会议**

获取需求文档 ===提前研读需求，整理疑点

提炼测试点，编写测试用例

任务认领，先到先得

**2：测试设计---重中之重**

按照计划进行测试用例设计

xmind提炼测试点即可

**3:测试执行**

开发发布提测通知时，会附上测试环境地址

在测试环境上，执行测试用例；先冒烟，后正式系统测试

提交bug并跟踪；测试3-4-5-6轮测试结束

**4：评估阶段**

测试结束输出测试报告，并发布邮件通知整个测试组

开发/运维上线

# 接口测试系列课程

## 目标：

1. 接口的概念 --HTTP HTTPS的封装
2. http协议的接口内容分析

● 请求方法：get,post

● request，response

● 状态码

● cookie/session/token

1. 接口相关问题

## 接口的概念

1. **接口是什么？======传输数据的通道**

**接口测试**：是测试系统组件间接口的一种测试

1.前端晚于后端开发，后端已经完成，没有前端提前介入测试，利用工具模拟前端

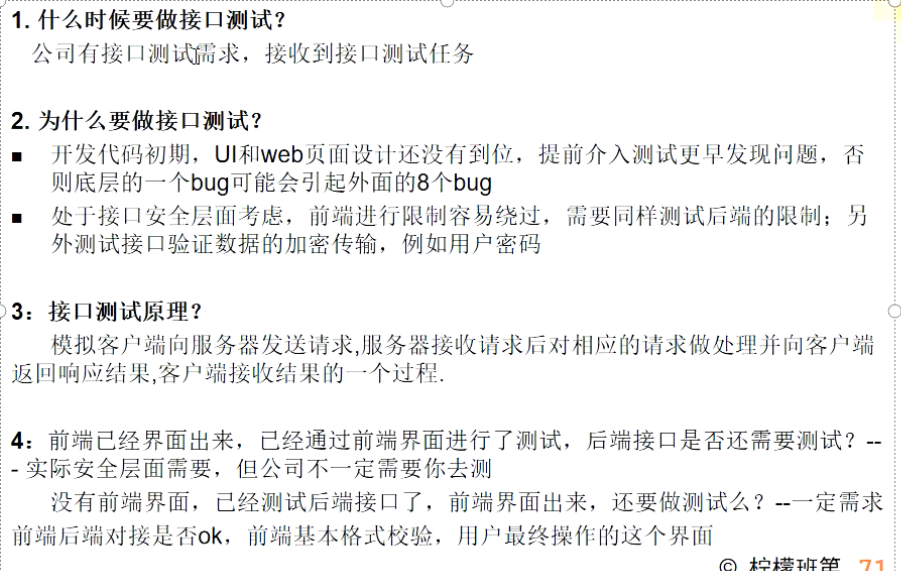
接口主要包括：

1. 同一个系统中模块与模块间的接口/**前端后端的接口** ====**内部接口**
2. 另一个是跨系统平台的接口 ===**外部接口**
3. **常见接口类型/协议**

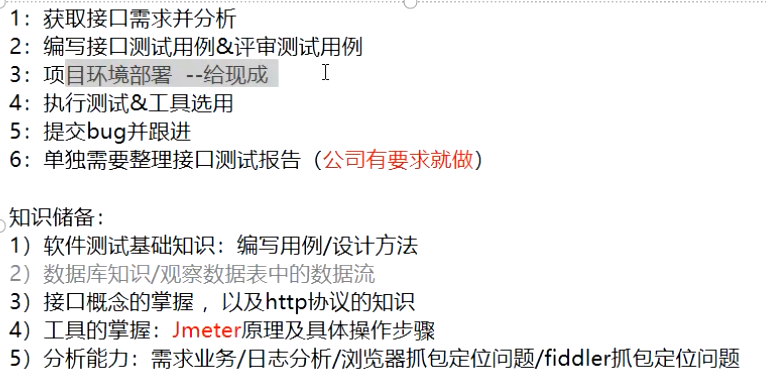
**● HTTP类型/协议：通过get/post等方法来获取数据，在数据处理上效率比较高**

● Webservice类型/协议：通过soap协议来获取数据，在数据处理上比HTTP能处理更加复杂的数据，本质上也是HTTP协议的封装。 -----了解就行

# ● 接口相关的问题



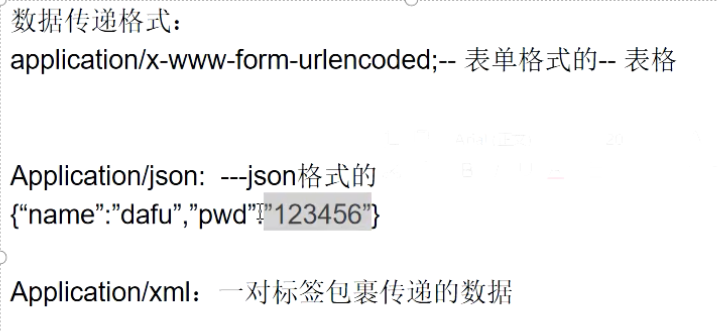
# 接口测试流程/步骤=接口测试怎么做？



# 获取接口需求并分析



# Content-Type



## 接口工具介绍

## 接口测试工具介绍



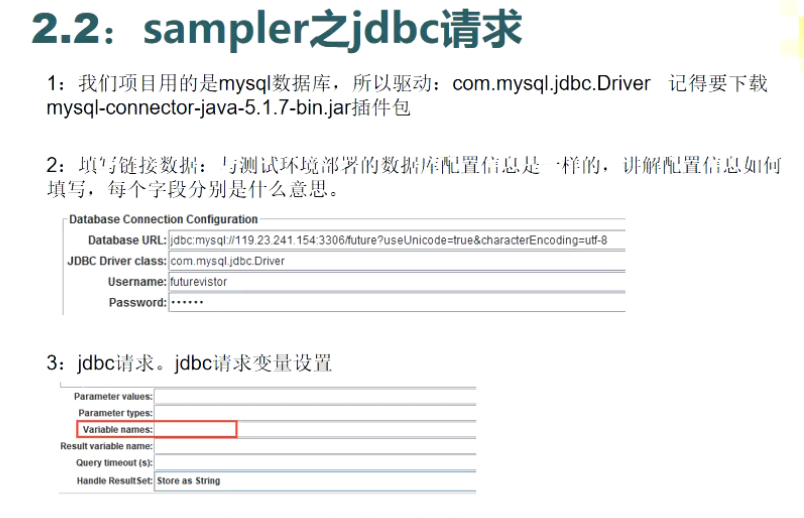
## Jmeter工具使用详解

## 

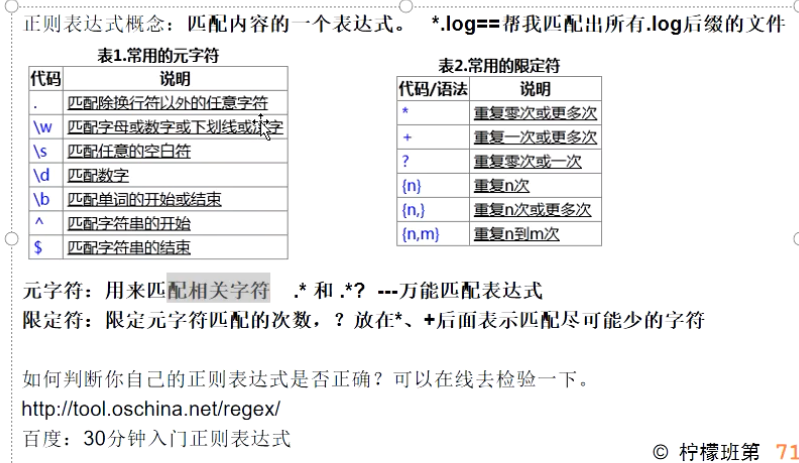
# 2.1、sampler之HTTP请求

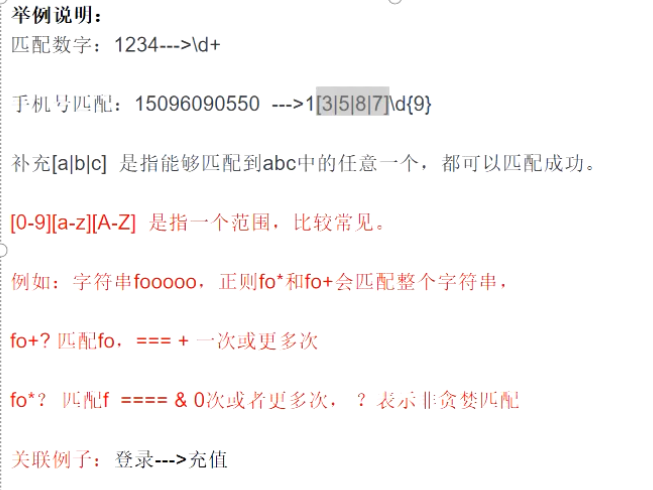


## 2.2、sampler之jdbc请求

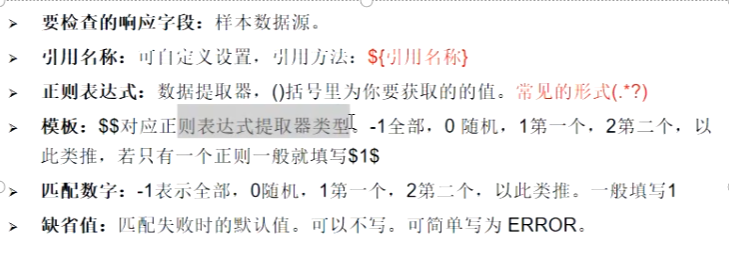


# 正则表达式





## 2.1、正则表达式提取器



## Jmeter断言



**Jmeter执行接口测试问题定位步骤：**

1. 查看结果树-红色 绿色 =====绿色-服务器正常响应（200 OK），红色：URL地址错了（空格，接口路径），网络是否有问题（ping 服务器IP地址）
2. 响应消息：响应体---code码，msg：描述这个问题：token过期====初步分析
3. 查看结果树里的请求---真正传到服务器那边的数据 请求头（），请求体

# 什么是自动化？

敏捷开发--快速迭代：回归测试--繁重 重复，枯燥==必须要做！自动化测试机器人--测试工具--Jmeter--接口：接口参数加密？？----beashell+java代码-二次开发代码自动化--灵活，满足自己特定需求

**什么阶段做自动化？**

回归测试（自动化测试用例覆盖率高-----80%+）

冒烟测试（自动化测试用例覆盖率低------30%）

**脚本实现在什么时候完成？-----一般在项目发布上线之后，下一个项目开始之前=手工测试用例转化为自动化测试脚本===写代码-----用在下一次迭代项目中**

自动化测试方向：

接口自动化：---稳定，最好实现！---使用普遍，效率最高

Web自动化：UI维护成本特别高

APP自动化：UI维护成本特别高

## 什么是python

