**UI用户界面**

需求——效果图——风格设计——高保证效果——html

**Ui设计师的工作：**

互联网平面设计师，banner，活动页面，logo，吉祥物。

线下平面，广告设计，以及传单，

页面制作，页面标注，切图，

每个公司或者每个团队都有自己的工作流。

所以，我这里提及到的只是囊括了自己认知的一部分。

**1.技术选型**

**2.搭建开发环境**

**3.项目部署**

**4.代码编写**

**5.代码质量管控**

**6.联调**

**7.跟踪测试**

**8.代码优化，性能优化**

**测试工程师**

自动化测试工程师

测试开发工程师

移动端测试工程师

接口测试工程师

软件测试的基本流程：

1.需求分析

2.编写测试用例

3.评审测试用例

4.搭建测试环境

5.等待开发提交测试包

6.部署测试包

7.冒烟测试（对软件主题基本功能进行基本测试），比如QQ，最基本的聊天要先实现。

8.执行测试用例

9.BUG跟踪处理

10.测试结束

**测试要点，测试点：**

为什么不让开发自己搞测试：

当前行业，有许多测试行业从业人员。而且他们之前本来就是开发岗。

专业度：软件测试和软件开发分别属于软件行业两个不同的技术方向。

所以尽量让专人做专事。

**思维定式：**

在软件的开发周期中，对于程序员而言，他们大多数时间是来思考如何实现 软件的功能。你的思想里面是如何提高软件的质量。而不会从用户的角度去考虑使用。

测试力度：相对于开发而言。产品就是他们的孩子，所以下手就不会那么狠。

**测试原则：**——任何软件都不可能是完美的。

1.测试证明软件存在缺陷。无论进行什么样的操作都不能证明软件是有缺陷的。

2.不能执行穷尽测试。——没有办法将所有的测试情况都罗列出来。所以任何的测试操作都是有结束的时候

3.缺陷存在群集现象。对于软件功能而言。二八理论，任何软件都的核心功能其实只占用20%，但是大部分测试都应该放在上面。

4.某些测试需要依赖特殊的环境

5.测试应尽早介入，测试人员不仅仅只是测试功能的。

6.杀虫剂现象。同样的一个测试用例不能重复使用多次。因为软件会对她产品免疫。

7.不存在缺陷谬论。你给开发提出一个这是一个bug，但是开发给你说这不是一个bug,哪你怎么回答

**软件质量**

描述当前软件是否好用，在当前的软件行业里，我们所采用的一套标准是基于ISO组织制订的。

需要我们记忆的就是软件质量的溜达特性。

1.功能性。 软件需要满足用户显式和隐式的功能。

2.易用性。 软件易于学习和使用。

3.可靠性。 软件必须实现需求中的功能。

4.效率性。 类似于软件的性能。

5.可维护性 后期如果想添加其他功能了，那么就应该能修改

6.可移植性。 当前软件可以从一个平台移植到另一个平台。

Demo:

1.业务逻辑正确性测试：产品文档。

2.兼容性测试：系统版本，分辨率，网络情况。

3.安卓和iOS。。

4.安卓4.0.5.0 等等。

5.原生系统和封装系统。。

6.苹果系统，版本之间是否兼容。。

7.手机屏幕之间的适配。屏幕分辨率。。

8.网络切换，中断恢复。

9.是否支持断点续传。。

10.电话&信息中断恢复、

11.升级&安装卸载测试。

13.手机资源消耗。 CPU的消耗和内存的消耗。。

14.流量消耗。 加载个图片瞬间消耗了20M，哪用户谁敢用。。

**常见的测试方法：**

一。按照测试对象来进行分类：

1.白盒测试            在意的就是底层代码的逻辑和底层功能实现，同时逻辑正确。

2.黑盒测试。        看不见的，这种就是测试的软件的外在功能是否正常。比如，登陆，注册

3.灰盒测试、            介于两者之间。

盒。指的就是我们的软件。

二。按照测试对象是否执行分类。

1.静态测试。        指的就是测试不执行。

2。动态测试。        将软件放在真实的环境中，测试功能。

三。按照测试手段进行分类。

手工测试，            由测试人员手动的对被测对象进行验证。优点就是能够灵活的改变测试操作。

自动化测试。            所谓的自动化有两种测试，一种就是自己写脚本，一种就是通过第三方的工具进行测试。

但是有时候第三方软件有可能会收费，优点：人不累，测试效率高。

在当前测试中，系统测试分类主要分为以下几种：

1.功能测试。            验证当前的软件主题功能是否可用。

2.兼容性测试。        验证当前软件在不同的软件下是否还能使用。

3.安全测试。            对授权用户是否提供使用。

4.性能测试                相当于当前软件消耗的资源。

**单元测试**：[uT]   —UNIT  TEST

单元测试，又称为模块测试，针对软件设计中的最小单位，程序模块。

进行正确性检查的测试工作，单元测试需要从程序的内部结构出发设计测试用例。

多个模块可以平行的独立进行单元测试。单元测试就是对源代码测试。

找到你当前项目中的某一个模块是否正常，不用管页面做的好看，看的只是底层代码结构。

一般就是类，函数，组件，

每一个小东西都是独立的模块，就叫单元测试。。测试程序中最小的模块。

**集成测试**：【it test】  system  ingertaion  test

又叫组装测试，通常在单元测试的基础上，将所有程序模块进行有序的，递增的测试。 组合起来，看他功能是否正常。不同的单元模块组合起来，

程序内部之间都是有相关性的。。把单元模块放在一起进行测试。。比如说接口测试。

**系统测试：【st】 system test**

将整个软件系统看做一个整体进行测试，包括对功能，性能，以及软件所运行的软硬件环境进行测试。

系统测试在系统集成完毕后进行测试，前期主要测试系统的功能是否满足需求，后期主要测试系统运行的性能是否满足需求，以及系统在不同软硬件环境中的兼容性问题。

测试人员吧自己当做普通用户进行测试。

黑盒测试：数据驱动，不考虑软件的内部结构和特性，只考虑软件的功能需求、不管代码

白盒测试：相当于吧盒子打开，去分析里面的结构和数据。

灰盒测试：介于两者之间的测试，

一般是采用白盒和黑盒测试，。。

电脑发生故障，文件没有备份而丢失了。

由于人员离职，导致某些资料丢失了。

我怎么知道手头的公共资料是不是最新版本的。

想要追溯几个月前的某个状态，却发现那个版本的文件以及被当做垃圾删除了。

每天要花费很多时间来向别人提供需要共享的资料、

相似的应用系统。每次都要重复开发，难以服用。

人员分布在两地开发，版本如何同步。

甲乙两人为不同目的修改了同一份文件，乙的提交在甲提交之后，导致甲修改的内容丢失了。