# 1 移动互联网概述

移动互联网，就是将移动通信和互联网二者结合起来，成为一体。是指互联网的技术、平台、商业模式和应用与移动通信技术结合并实践的活动的总称。而与此相对的就是传统互联网，传统的互联网与移动互联网都属于互联网的范畴，如果要说移动互联网与传统互联网的区别，那就在于“移动”。现在的“移动”特性体现在“随时”和“随地”的利用网络获取所需要的信息。

在移动互联网发展的初期，大部分产品的界面、交互都是根据产痛互联网时期获得的经验来设计的。但是在慢慢的体验过程中，发现这些经验在移动平台上的体验越来越不受用户所喜爱，因此需要根据移动互联网的特点来针对性的进行满足移动互联网特性的设计。

# 2 移动互联与传统互联体验的区别

作为一名移动互联网测试工程师，需要对产品有着良好的使用经验，而这些经验也会引导你对产品进行测试，只有充分了解了移动互联网的特点后才有可能把测试做好，把产品做到极致。

## 2.1用户关注点

早期的互联网，用户通过web网页，因为电脑的屏幕都比较大，所以需要获取的信息要相对较全，而不是要用户经过大量的点击、翻页才能获取到足够的信息。

而移动互联网的终端，最大的也就是Pad电脑，也不过区区10寸的屏幕大，大部分还是4寸~5.5寸的屏幕，这时候如果仍旧在这么大的页面上显示密密麻麻的信息，最终会导致用户会直接将你公司的产品扔进垃圾筐！移动互联网一个巨大的挑战就是将视线从电脑屏幕转移到小屏幕，能够显示的信息非常有限和珍贵，因此就需要设计人员要将最优价值的信息放在应用的显眼位置，否则必将被淘汰。

## 2.2场合复杂化

传统互联网人们一般在家里或者办公室等固定场所使用电脑，相信几乎不会有人坐公交车还开这个电脑看网页。但是对于移动互联网，更多的用户会选择在公交、地铁、上厕所、排队的时候使用手机。因此作为测试人员，需要将各种不同的场景下去测试产品，包括网络2G、3G、4G；网络状况不佳；电量不足等场景。

## 2.3时间碎片化

用户使用移动应用的时间大多数都是碎片化的，看微博、看新闻、玩小游戏等等。那么应用在设计时就需要考虑到要符合用户碎片化时间中的使用习惯，能够在碎片化的时间内阐述表明自己应用的独特之处，因为用户没有那么多时间浪费在此。

## 2.4输入困难化

用户在使用电脑的时候，鼠标、键盘都是非常方便的输入设备。但是在移动互联网上，应用的输入变的尤为复杂，移动互联网的输入一般分为实体键盘和触屏。输入方式的改变会让人一时不适应，尤其是在复杂的交互场景中。

# 3 功能测试要点

## 3.1多分辨率测试

IOS目前机型相对较少，得益于Apple公司专心致力于为用户提供极致的产品。而Android不一样了，从240\*320 到2560\*1600多大几十款。对于多分辨率测试可以考虑使用模拟器和真机结合的方式进行测试。Android可以根据目前市场上的分辨率统计数据有针对性的进行真机测试。

## 3.2 多系统测试

继分辨率测试之后紧接着的就是多系统支持的测试。Android的大部分使用的系统都为2.3以上，相对分辨率测试而言，多系统测试比其更为复杂，因为除了原生的Android系统外，市面上很多厂家都对其进行了深度定制，像华为的EMUI、小米的MUI、魅族的Flyme、乐视的EUI、联想的VIBEUI等还有其他的系统。这些都给测试工作增加了一定的复杂度。而apple的用户群大部分都有保持升级到最新系统的习惯，并且IOS只能单向进行升级，不能降级，使得IOS系统的测试工作相对较为简单，不过也需要尽可能的覆盖5.0以上的系统。

## 3.3 用户不同的使用习惯

**①Android权限问题**

作为测试工程师，需要了解被测试对象使用了哪些权限，为什么使用，是否会对用户造成困扰等问题。

**②Android硬件问题**

Android智能机系统层出不穷，同时硬件差异也很大。硬件上的差异同样会造成软件商的缺陷，如实体键盘和虚拟键盘。

**③Android操作习惯**

主要有以下几个按键：菜单键(Menu)、Home键，返回键，Home键长按，显示当前进程列表、调整音量、待机。

需要考虑以下问题：在各个功能界面点击Back键会有什么反馈；用户在各个界面点击Home键之后，再次打开应用的反馈，应用是默认处于后台的状态还是结束生命周期；应用是否使用到菜单键，切不可遗漏这个案件与应用的交互；系统在待机下，系统被重新唤醒后，应用处于什么状态，是否有正确的反馈；应用进程被用户手动关闭后，是否可以再次正确启动。

**④IOS操作习惯**

主要有以下几个按键：单击Home键、双击Home键、关闭当前某应用程序、打开或关闭音量、调整音量、待机、

**⑤IOS越狱问题**

在相同的IOS环境下使用不同的越狱工具，一些应用的部分功能所表现出的现象可能不一样。因为越狱之后，最常见的一个问题就是系统变得不稳定，容易导致应用程序冲突，如输入法、美化主题插件、手机管理软件。

## 3.4 网络的不稳定性

移动互联网之所以比传统互联网更具有传播力，流行得更快，正是因为人们可以随时随地的拿出移动设备来接入互联网，像发微博、发朋友圈、发空间等。从2G到4G网络的快速发展，可看到人们使用互联网对于网络的依赖性越来越高。

而目前3G、4G信号未全覆盖，部分地方会出现网络状况不好的情况，这时候的应用会如何响应用户的操作呢？这时候需要测试工程师去测试应用在此类场景下的反应，像地铁、电梯、洗手间、公交车等场景。而目前一些应用在此类情况下的提示皆为：“网络错误，请稍后再试”，这种通篇提示对于用户来说，明显是不负责任的做法。

对于测试工程师需要完完全全站在用户角度去思考问题。针对网络问题，根据不同的状况给予不同的提示，如网络不稳定，请稍后再试、网络目前未连接，请重新连接网络之后再尝试、目前无法接收相关数据，请稍后再试···

还有就是对于网络切换之间的场景测试、网络状态由好变差的场景测试。

## 3.5 安装/卸载测试

Android和IOS安装应用的方式有很多种。像Android的各种第三方的手机管家、命令行安装；IOS的APPStore及越狱后的PP助手等第三方工具。虽然这些安装方式跟手动命令行安装没什么区别，但还是需要测试工程师进行实际的操作验证之后，才能保证该测试点的逻辑正常。对于Android支持扩展卡的情况，两种情况下都需要进行测试，同时数据相互转移的场景也需要进行测试。

以下场景还需要关注：

* 不联网的情况下是否能正常运行，会不会崩溃？
* 在线下载数据不完整，能否启动？
* 是否支持断点续传？
* 安装成功后，检查版本号等信息

## 3.6 升级测试

应用升级包括增量升级和全量升级。所谓的增量升级就是，旧版APK只有6M，新版有8M，那增量升级就可能只需要下载2M即可；而全量升级则是8M全下载。

应用升级的几个必要条件：新旧版本需要相同的签名；新旧版本需要相同的文件名；新旧版本需要有标识符来进行区分。

另外一些手机ROM在出厂时会包括一些内置应用，这时候升级的话要跟系统的.so文件进行对比。

## 3.7 并发测试

APP测试不同于传统测试的区别在于，移动设备在使用过程中并发情况较多，如你在使用一个应用的过程中，会出现如下情形：来电、来短信、闹铃、通知、电量低警告等情况。

而测试工程师需要测试的是，在出现上述情形时，应用是否能继续使用而不产生崩溃情形。

如下几种情形需要注意：应用正在播放视频、应用正在向服务器连接发出或者接受请求、应用正在下载数据或升级、用户正在进行输入、关机或待机、功能冲突等。

## 3.8 推送

推送功能是企业推广自己产品和保持用户粘性的不二选择，因此推送功能也是测试的重点之一。

在测试时需要关注以下几点：智能机在关机、待机和打开等状态下执行推送的功能、消息显示以及推送跳转等是否正确；应用在打开或关闭情况，应用启动在后台运行等情况下的推送功能测试；以及多次推送和推送的成功率也需要进行测试。

## 3.9 分享跳转

移动互联网的一大特性就是传播速度非常的迅速，一则新闻在很短的时间内就能被众人所知，像微博、微信、QQ空间等。需要模拟分享成功和失败的场景。

# 4 测试工具介绍

## 4.1 Tcpdump/WireShark介绍

对于需要定位网络问题--丢包率的测试，我们可以使用Tcpdump和wireshark这两款工具进行测试。

使用方法如下：

*#将Tcpdump放入到Android系统的目录中去*

adb push <tcpdump path> /data/local/tcpdump

*#赋予相应的权限给Tcpdump文件目录*

adb shell chmod 6755 /data/local/tcpdump

*#保存抓包文件*

/data/local/tcpdump -p -vv -s 0 -w /data/local/tcpdump/capture.pcap

*#导出抓包文件*

adb pull /data/local/capture.pcap .

# 5 测试用例设计

场景：智能机一个很常用的功能就是打电话，但是有时候会接到许多莫名其妙的骚扰电话，因此某家公司抓住该点设计了一款具有“黑名单”功能的APP，该功能可以阻止加入其列表中的号码或联系人再次与智能机主人通话。那该如何设计测试用例呢？

首先想到的是最基本的用例：

**①添加一个或多个联系人或号码进入黑名单，阻止通话**

**②从黑名单中删除一个或多个联系人或号码，阻止通话**

上述两个用例应该做过测试的都会想到，但是有经验的测试人员又会想出新的测试用例：

**①当黑名单未添加任何一个号码的时候，界面显示是否正常**

**②当黑名单添加了一个号码之后，界面显示是否正常**

**③当黑名单正好满一个屏幕的时候，界面显示是否正常**

**④当黑名单列表超过一个屏幕的时候，界面显示是否正常**

貌似一看也就只有这些用例，但是我们想象下使用该功能的场景，使用黑名单功能原因有以下几种：被某个陌生号码一直骚扰、有意识的不想和某号码联系、恶作剧等

因此，又会联想到一些测试用例：

**①直接添加一个号码到黑名单**

**②从联系人中选一个联系人到黑名单**

**③从通话记录中选一个号码添加到黑名单中**

但是单从添加号码的这个行为上挖掘还不够，应该深挖联系人和号码两个因素：

**①号码的长度（固话、移动电话、短号码、小灵通等）**

**②号码的不同表现方式（如13800138000和+861380013800是一个号码，包括漫游、甚至跨国号码等）**

是不是只有以上这些用例呢？给你的回答：不！还有。

像现在出现的双卡单待、双卡双待机器，此时添加黑名单又会产生新的用例：

双卡双待情况（AB都有同一个号码C）：

**①黑名单从A卡中添加C，用C给A卡、B卡分别拨打电话**

双卡单待情况（AB都有同一个号码C）：

**②黑名单从A卡中添加C，用C给A卡、B卡分别拨打电话**