**一 网络状态**

**1. 存在哪些网络状态？**

* 网络畅通
* WIFI与移动网络切换
* 弱网模拟
* 真实弱网场景
* 断网状态

**2. 不同网络状态下的测试点**

* 加载中的状态
* 加载失败的状态
* 加载失败的提示文案
* 断网重连后的网络请求
* 弱网重复点击
* 弱网丢包
* 网络切换后功能是否正常
* 断网启动
* 断网点击

**二 机型适配**

**1. 存在哪些机型适配问题？**

* 不同屏幕大小
* 不同分辨率
* 不同系统版本
* 不同机型

**2. 如何选择测试机？**

* 需求方要求的最低兼容版本
* 市面上比较热的机型和系统版本
* APP用户统计机型top30
* APP crash统计机型top10

**3. 不同机型适配下的测试点**

* UI展示是否完整
* UI展示是否重叠
* 不同机型对第三方库的兼容性
* 不同机型本身的特性
* 不同机型的手势冲突
* 不同机型的权限控制
* 不同语言环境的功能和UI展示
* 不同系统时间下的功能和UI展示
* 不同字体大小的功能和UI展示
* 不同主题下的功能和UI展示
* 不同输入法对输入框的功能和UI展示

**三 安装卸载升级**

**1. 安装**  
各大应用市场安装，不同机型，命令行（adb install <packagename>）  
**2. 卸载**  
是否卸载干净，卸载后重新安装能够正常使用  
**3. 升级**  
更新提醒，选择更新，强制更新  
覆盖升级  
手动更新，自动更新

**四 并发测试**

**1. 弹出框提示**  
闹钟，低电量，短信，微信，通知  
**2.另一个应用启动**  
正在播放视频，正在发送或接受请求，正在下载数据或升级，正在进行输入  
**3.关机或待机**  
推送，数据刷新，数据同步  
**4.功能冲突**  
音乐，语音，相机  
**5.可存储设备**  
图片，音乐

**五 内存泄漏**

**1. 为什么会产生内存泄漏？**  
当一个对象已经不需要再使用了，本该被回收时，而有另外一个正在使用的对象持有它的引用从而导致它不能被回收，这导致本该被回收的对象不能被回收而停留在堆内存中，这就产生了内存泄漏。  
**2. 内存泄漏对程序的影响？**  
内存泄漏是造成应用程序OOM的主要原因之一！  
我们知道android系统为每个应用程序分配的内存有限，而当一个应用中产生的内存泄漏比较多时，这就难免会导致应用所需要的内存超过这个系统分配的内存限额，这就造成了内存溢出而导致应用Crash。  
**3. 检测内存泄漏常用工具？**

* LeakCannary
* debugBottle
* Memory Leaks

**六 抓包工具**

**1. 常用抓包工具？**

* Fiddler
* Charles

**2. 抓包工具常用测试点？**

* 接口返回状态是否正常
* 接口返回数据是否符合规范
* APP传参是否正确
* APP接口请求是否合理
* 获取token,版本号，userId，clientId等信息
* 修改request
* 修改response
* 模拟弱网

**七 Crash收集**

**1. 常用收集平台？**

* Fabric
* Bugly
* Umeng

**2. 常见Crash类型？**

* 内存溢出（OutOfMemory）
* 空指针（NullPointerException）
* 数组越界（OutOfBounds）
* 图片渲染
* 传参问题
* 数据未判空
* 数据类型不匹配
* NSAttributedString相关
* 网络问题
* 数据结构变化

**八 提测包管理**

**常用提测包第三方管理平台？**

* fir
* 蒲公英
* TestFlight

**九 云测**

**常用云测平台？**

* Testin
* MTC
* MQC
* WeTest