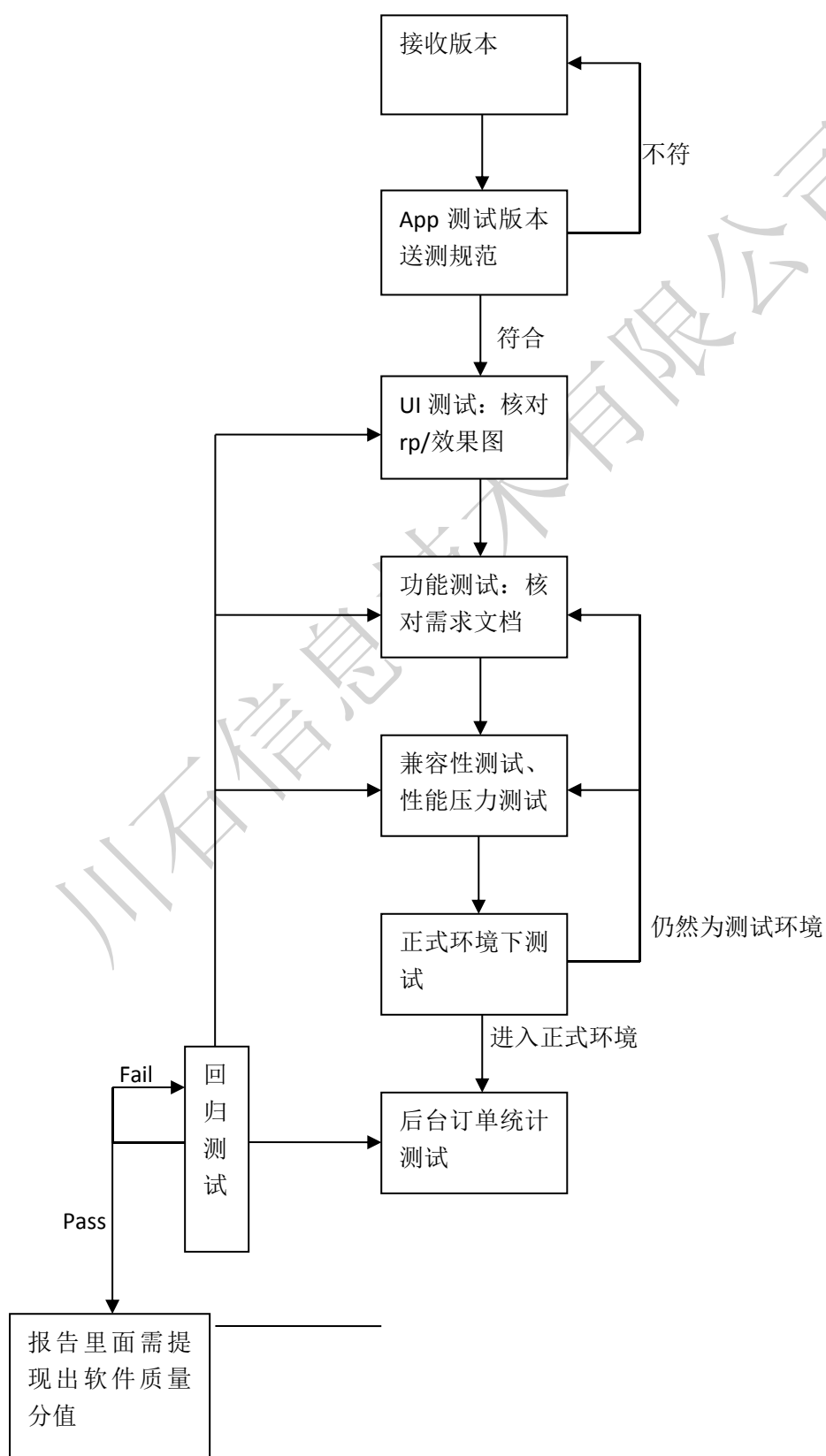


1 APP 测试基本流程

1.1 流程图



1.2 测试周期

测试周期可按项目的开发周期来确定测试时间，一般测试时间为两三周（即 15 个工作日），根据项目情况以及版本质量可适当缩短或延长测试时间。

1.3 测试资源

测试任务开始前，检查各项测试资源。

- 产品功能需求文档；
- 产品原型图；
- 产品效果图；
- 测试设备；
- 其他。

1.4 日报及产品上线报告

- 1) 测试人员每天需对所测项目发送测试日报。（也就是我这边有邮件通知测试项目的时候一般均属于输出测试日报）
- 2) 测试日报所包含的内容为：
\\Dell-server\网站软件 app 等开发\产品测试部\测试知识区域\测试文档类模板\项目测试报告邮件输出模板.doc
- 4) 不同版本测试报告输出

2 App 测试点

2.1 安全测试

2.1.1 软件权限

- 1) 扣费风险：包括发送短信、拨打电话、连接网络等
- 2) 隐私泄露风险：包括访问手机信息、访问联系人信息等
- 3) 对 App 的输入有效性校验、认证、授权、敏感数据存储、数据加密等方面进行检测
- 4) 限制/允许使用手机功能接入互联网
- 5) 限制/允许使用手机发送接受信息功能
- 6) 限制/允许应用程序来注册自动启动应用程序
- 7) 限制或使用本地连接
- 8) 限制/允许使用手机拍照或录音

- 9) 限制/允许使用手机读取用户数据
- 10) 限制/允许使用手机写入用户数据
- 11) 检测 App 的用户授权级别、数据泄漏、非法授权访问等

2.1.2 安装与卸载安全性

- 1) 应用程序应能正确安装到设备驱动程序上
- 2) 能够在安装设备驱动程序上找到应用程序的相应图标
- 3) 是否包含数字签名信息
- 4) 没有用户的允许, 应用程序不能预先设定自动启动
- 5) 卸载是否安全, 其安装进去的文件是否全部卸载
- 6) 卸载用户使用过程中产生的文件是否有提示
- 7) 其修改的配置信息是否复原
- 8) 卸载是否影响其他软件的功能
- 9) 卸载应该移除所有的文件

2.1.3 数据安全性

- 1) 当将密码或其他敏感数据输入到应用程序时, 其不会被储存在设备中, 同时密码也不会被解码
- 2) 输入的密码将不以明文形式进行显示
- 3) 密码, 信用卡明细, 或其他敏感数据将不被储存在它们预输入的位置上
- 4) 不同的应用程序的个人身份证或密码长度必需至少在 4 — 8 个数字长度之间
- 5) 当应用程序处理信用卡明细, 或其他敏感数据时, 不以明文形式将数据写到其它单独的文件或者临时文件中。以 6) 防止应用程序异常终止而又没有删除它的临时文件, 文件可能遭受人侵者的袭击, 然后读取这些数据信息。
- 7) 当将敏感数据输入到应用程序时, 其不会被储存在设备中
- 8) 备份应该加密, 恢复数据应考虑恢复过程的异常 通讯中断等, 数据恢复后再使用前应该经过校验
- 9) 应用程序应考虑系统或者虚拟机器产生的用户提示信息或安全警告
- 10) 应用程序不能忽略系统或者虚拟机器产生的用户提示信息或安全警告, 更不能在安全警告显示前, 利用显示误导信息欺骗用户, 应用程序不应该模拟进行安全警告误导用户
- 11) 在数据删除之前, 应用程序应当通知用户或者应用程序提供一个“取消”命令的操作
- 12) “取消”命令操作能够按照设计要求实现其功能
- 13) 应用程序应当能够处理当不允许应用软件连接到个人信息管理的情况
- 14) 当进行读或写用户信息操作时, 应用程序将会向用户发送一个操作错误的提示信息
- 15) 在没有用户明确许可的前提下不损坏删除个人信息管理应用程序中的任何内容 M
- 16) 应用程序读和写数据正确。
- 17) 应用程序应当有异常保护。
- 18) 如果数据库中重要的数据正要被重写, 应及时告知用户
- 19) 能合理地处理出现的错误

20) 意外情况下应提示用户

2.1.4 通讯安全性

- 1) 在运行其软件过程中, 如果有来电、SMS、EMS、MMS、蓝牙、红外等通讯或充电时, 是否能暂停程序, 优先处理通信, 并在处理完毕后能正常恢复软件, 继续其原来的功能
- 2) 当创立连接时, 应用程序能够处理因为网络连接中断, 进而告诉用户连接中断的情况
- 3) 应能处理通讯延时或中断
- 4) 应用程序将保持工作到通讯超时, 进而发送给用户一个错误信息指示有连接错误
- 5) 应能处理网络异常和及时将异常情况通报用户
- 6) 应用程序关闭或网络连接不再使用时应及时关闭) 断开
- 7) HTTP、HTTPS 覆盖测试

--App 和后台服务一般都是通过 HTTP 来交互的, 验证 HTTP 环境下是否正常;

--公共免费网络环境中(如: 麦当劳、星巴克等)都要输入用户名和密码, 通过 SSL 认证来访问网络, 需要对使用 HTTP Client 的 library 异常作捕获处理。

2.1.5 人机接口安全性

- 1) 返回菜单总保持可用
- 2) 命令有优先权顺序
- 3) 声音的设置不影响应用程序的功能
- 4) 应用程序必需利用目标设备适用的全屏尺寸来显示上述内容
- 5) 应用程序必需能够处理不可预知的用户操作, 例如错误的操作和同时按下多个键

2.2 安装、卸载测试

验证 App 是否能正确安装、运行、卸载 以及操作过程和操作前后对系统资源的使用情况

2.2.1 安装

- 1) 软件在不同操作系统下安装是否正常。
- 2) 软件安装后的是否能够正常运行, 安装后的文件夹及文件是否写到了指定的目录里。
- 3) 软件安装各个选项的组合是否符合概要设计说明
- 4) 软件安装向导的 UI 测试
- 5) 软件安装过程是否可以取消, 点击取消后, 写入的文件是否如概要设计说明处理
- 6) 软件安装过程中意外情况的处理是否符合需求(如死机, 重启, 断电)
- 7) 安装空间不足时是否有相应提示
- 8) 安装后没有生成多余的目录结构和文件
- 9) 对于需要通过网络验证之类的安装, 在断网情况下尝试一下

10) 还需要对安装手册进行测试, 依照安装手册是否能顺利安装

2.2.2 卸载

- 1) 直接删除安装文件夹卸载是否有提示信息。
- 2) 测试系统直接卸载程序是否有提示信息。
- 3) 测试卸载后文件是否全部删除所有的安装文件夹。
- 4) 卸载过程中出现的意外情况的测试(如死机、断电、重启)。
- 5) 卸载是否支持取消功能, 单击取消后软件卸载的情况。
- 6) 系统直接卸载 UI 测试, 是否有卸载状态进度条提示。

2.3 UI 测试

测试用户界面(如菜单、对话框、窗口和其它可规控件)布局、风格是否满足客户要求、文字是否正确、页面是否美观、文字、图片组合是否完美、操作是否友好等。

UI测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。确保用户界面符合公司或行业的标准。包括用户友好性、人性化、易操作性测试。

2.3.1 导航测试

- 1) 按钮、对话框、列表和窗口等; 或在不同的连接页面之间需要导航
- 2) 是否易于导航, 导航是否直观
- 3) 是否需要搜索引擎
- 4) 导航帮助是否准确直观
- 5) 导航与页面结构、菜单、连接页面的风格是否一致

2.3.2 图形测试

- 1) 横向比较。各控件操作方式统一
- 2) 自适应界面设计, 内容根据窗口大小自适应
- 3) 页面标签风格是否统一
- 4) 页面是否美观
- 5) 页面的图片应有其实际意义而要求整体有序美观
- 6) 图片质量要高且图片尺寸在设计符合要求的情况下应尽量小
- 7) 界面整体使用的颜色不宜过多

2.3.3 内容测试

- 1) 输入框说明文字的内容与系统功能是否一致
- 2) 文字长度是否加以限制
- 3) 文字内容是否表意不明
- 4) 是否有错别字
- 5) 信息是否为中文显示
- 6) 是否有敏感性词汇、关键词
- 7) 是否有敏感性图片, 如: 涉及版权、专利、隐私等图片

2.4 功能测试

根据软件说明或用户需求验证 App 的各个功能实现, 采用如下方法实现并评估功能测试过程:

- 1) 采用时间、地点、对象、行为和背景五元素或业务分析等方法分析、提炼 App 的用户使用场景, 对比说明或需求, 整理出内在、外在及非功能直接相关的需求, 构建测试点, 并明确测试标准, 若用户需求中无明确标准遵循, 则需要参考行业或相关国际标准或准则。
- 2) 根据被测功能点的特性列出相应类型的测试用例对其进行覆盖, 如: 涉及输入的地方需要考虑等价、边界、负面、异常或非法、场景回滚、关联测试等测试类型对其进行覆盖。
- 3) 在测试实现的各个阶段跟踪测试实现与需求输入的覆盖情况, 及时修正业务或需求理解错误。

2.4.1 运行

- 1) App 安装完成后的试运行, 可正常打开软件。
- 2) App 打开测试, 是否有加载状态进度提示。
- 3) App 打开速度测试, 速度是否可观。
- 4) App 页面间的切换是否流畅, 逻辑是否正确
- 5) 注册
 - 同表单编辑页面
 - 用户名密码长度
 - 注册后的提示页面
 - 前台注册页面和后台的管理页面数据是否一致
 - 注册后, 在后台管理中页面提示
- 6) 登录
 - 使用合法的用户登录系统。
 - 系统是否允许多次非法的登陆, 是否有次数限制。
 - 使用已经登陆的账号登陆系统是否正确处理。

- 使用禁用的账号登陆系统是否正确处理。
- 用户名、口令（密码）错误或漏填时能否登陆。
- 删除或修改后的用户，原用户登陆。
- 不输入用户口令和用户、重复点（确定或取消按钮）是否允许登陆。
- 登陆后，页面中登陆信息。
- 页面中有注销按钮。
- 登陆超时的处理。

7) 注销

- 注销原模块，新的模块系统能否正确处理。
- 终止注销能否返回原模块，原用户。
- 注销原用户，新用户系统能否正确处理。
- 使用错误的账号、口令、无权限的被禁用的账号进行注销

2.4.2 应用的前后台切换

- 1) APP 切换到后台，再回到 app，检查是否停留在上一次操作界面。
- 2) APP 切换到后台，再回到 app，检查功能及应用状态是否正常，IOS4 和 IOS5 的版本的处理机制有的不一样。
- 3) app 切换到后台，再回到前台时，注意程序是否崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。
- 4) 手机锁屏解屏后进入 app 注意是否会崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。
- 5) 当 App 使用过程中有电话进来中断后再切换到 app，功能状态是否正常
- 6) 当杀掉 app 进程后，再开启 app，app 能否正常启动。
- 7) 出现必须处理的提示框后，切换到后台，再切换回来，检查提示框是否还存在，有时候会出现应用自动跳过提示框的缺陷。
- 8) 对于有数据交换的页面，每个页面都必需要进行前后台切换、锁屏的测试，这种页面最容易出现崩溃。

2.4.3 免登录

很多应用提供免登录功能，当应用开启时自动以上一次登录的用户身份来使用app.

- 1) app有免登录功能时，需要考虑IOS版本差异。
- 2) 考虑无网络情况时能否正常进入免登录状态。
- 3) 切换用户登录后，要校验用户登录信息及数据内容是否相应更新，确保原用户退出。
- 4) 根据MTOP的现有规则，一个帐户只允许登录一台机器。所以，需要检查一个帐户登录多台手机的情况。原手机里的用户需要被踢出，给出友好提示。
- 5) app切换到后台，再切回前台的校验
- 6) 切换到后台，再切换回前台的测试
- 7) 密码更换后，检查有数据交换时是否进行了有效身份的校验

- 8) 支持自动登录的应用在进行数据交换时, 检查系统是否能自动登录成功并且数据操作无误。
- 9) 检查用户主动退出登录后, 下次启动app, 应停留在登录界面

2.4.4 数据更新

根据应用的业务规则, 以及数据更新量的情况, 来确定最优的数据更新方案。

- 1) 需要确定哪些地方需要提供手动刷新, 哪些地方需要自动刷新, 哪些地方需要手动+自动刷新。
- 2) 确定哪些地方从后台切换回前台时需要进行数据更新。
- 3) 根据业务、速度及流量的合理分配, 确定哪些内容需要实时更新, 哪些需要定时更新。
- 4) 确定数据展示部分的处理逻辑, 是每次从服务端请求, 还是有缓存到本地, 这样才能有针对性的进行相应测试。
- 5) 检查有数据交换的地方, 均有相应的异常处理。

2.4.5 离线浏览

很多应用会支持离线浏览, 即在本地客户端会缓存一部分数据供用户查看。

- 1) 在无网络情况可以浏览本地数据
- 2) 退出 app 再开启 app 时能正常浏览
- 3) 切换到后台再切回前台可以正常浏览
- 4) 锁屏后再解屏回到应用前台可以正常浏览
- 5) 在对服务端的数据有更新时会给予离线的相应提示

2.4.6 App 更新

- 1) 当客户端有新版本时, 有更新提示。
- 2) 当版本为非强制升级版时, 用户可以取消更新, 老版本能正常使用。用户在下次启动 app 时, 仍能出现更新提示。
- 3) 当版本为强制升级版时, 当给出强制更新后用户没有做更新时, 退出客户端。下次启动 app 时, 仍出现强制升级提示。
- 4) 当客户端有新版本时, 在本地不删除客户端的情况下, 直接更新检查是否能正常更新。
- 5) 当客户端有新版本时, 在本地不删除客户端的情况下, 检查更新后的客户端功能是否是新版本。
- 6) 当客户端有新版本时, 在本地不删除客户端的情况下, 检查资源同名文件如图片是否能正常更新成最新版本。

2.4.7 定位、照相机服务

- 1) App 有用到相机，定位服务时，需要注意系统版本差异
- 2) 有用到定位服务、照相机服务的地方，需要进行前后台的切换测试，检查应用是否正常。
- 3) 当定位服务没有开启时，使用定位服务，会友好性弹出是否允许设置定位提示。当确定允许开启定位时，能自动跳转到定位设置中开启定位服务。
- 4) 测试定位、照相机服务时，需要采用真机进行测试。

2.4.8 时间测试

客户端可以自行设置手机的时区、时间，因此需要校验该设置对 app 的影响。

--中国为东 8 区，所以当手机设置的时间非东 8 区时，查看需要显示时间的地方，时间是否展示正确，应用功能是否正常。时间一般需要根据服务器时间再转换成客户端对应的时区来展示，这样的用户体验比较好。比如发表一篇微博在服务端记录的是 10: 00，此时，华盛顿时间为 22: 00，客户端去浏览时，如果设置的是华盛顿时间,则显示的发表时间即为 22:00, 当时间设回东 8 区时间时，再查看则显示为 10: 00。

2.4.9 PUSH 测试

- 1) 检查 push 消息是否按照指定的业务规则发送
- 2) 检查不接受推送消息时，检查用户不会再接收到 push.
- 3) 如果用户设置了免打扰的时间段，检查在免打扰时间段内，用户接收不到 PUSH。
在非免打扰时间段，用户能正常收到 push。
- 4) 当 push 消息是针对登录用户的时候，需要检查收到的 push 与用户身份是否相符，没有错误地将其它人的消息推送过来。一般情况下，只对手机上最后一个登录用户进行消息推送。
- 5) 测试 push 时，需要采用真机进行测试。

2.5 性能测试

评估 App 的时间和空间特性：

- 1) 极限测试：在各种边界压力情况下，如电池、存储、网速等，验证 App 是否能正确响应。

--内存满时安装 App

--运行 App 时手机断电

--运行 App 时断掉网络

- 2) 响应能力测试：测试 App 中的各类操作是否满足用户响应时间要求。（安装包放到云测上可以测试）

--App 安装、卸载的响应时间

--App 各类功能性操作的影响时间

3) 压力测试: 反复/长期操作下、系统资源是否占用异常。(itestin)

--App 反复进行安装卸载, 查看系统资源是否正常

--其他功能反复进行操作, 查看系统资源是否正常

4) 性能评估: 评估典型用户应用场景下, 系统资源的使用情况。(Jmeter)

2.6 交叉事件测试

针对智能终端应用的服务等级划分方式及实时特性所提出的测试方法。交叉测试又叫事件或冲突测试, 是指一个功能正在执行过程中, 同时另外一个事件或操作对该过程进行干扰的测试。如: App 在前/后台运行状态时与来电、文件下载、音乐收听等关键运用的交互情况测试等。交叉事件测试非常重要, 能发现很多应用中潜在的性能问题。

- 1) 多个 App 同时运行是否影响正常功能
- 2) App 运行时前/后台切换是否影响正常功能
- 3) App 运行时拨打/接听电话
- 4) App 运行时发送/接收信息
- 5) App 运行时发送/收取邮件
- 6) App 运行时切换网络 (2G、3G、wifi)
- 7) App 运行时浏览网络
- 8) App 运行时使用蓝牙传送/接收数据
- 9) App 运行时使用相机、计算器等手机自带设备

2.7 兼容测试

主要测试内部和外部兼容性

- 1) 与本地及主流 App 是否兼容
- 2) 基于开发环境和生产环境的不同, 检验在各种网络连接下 (WiFi、GSM、GPRS、EDGE、WCDMA、CDMA1x、CDMA2000、HSPDA 等), App 的数据和运用是否正确
- 3) 与各种设备是否兼容, 若有跨系统支持则需要检验是否在各系统下, 各种行为是否一致
 - 不同操作系统的兼容性, 是否适配
 - 不同手机屏幕分辨率的兼容性
 - 不同手机品牌的兼容性

2.8 回归测试

- 1) Bug 修复后且在新版本发布后需要进行回归测试。
- 2) Bug 修复后的回归测试在交付前、要进行全量用例的回归测试。

2.9 升级、更新测试

新版发布后, 配合不同网络环境的自动更新提示及下载、安装、更新、启动、运行的验证测试。

- 1) 测试升级后的功能是否与需求说明一样

- 2) 测试与升级模块相关的模块的功能是否与需求一致
- 3) 升级安装意外情况的测试（如死机、断电、重启）
- 4) 升级界面的 UI 测试
- 5) 不同操作系统间的升级测试

2.10 用户体验测试

以主观的普通消费者的角度去感知产品或服务的舒适、有用、易用、友好亲切程度。通过不同个体、独立空间和非经验的统计复用方式去有效评价产品的体验特性 提出修改意见提升产品的潜在客户满意度。

- 1) 是否有空数据界面设计，引导用户去执行操作。
- 2) 是否滥用用户引导。
- 3) 是否有不可点击的效果，如：你的按钮此时处于不可用状态，那么一定要灰掉，或者拿掉按钮，否则会给用户误导
- 4) 菜单层次是否太深
- 5) 交互流程分支是否太多
- 6) 相关的选项是否离得很远
- 7) 一次是否载入太多的数据
- 8) 界面中按钮可点击范围是否适中
- 9) 标签页是否跟内容没有从属关系，当切换标签的时候，内容跟着切换
- 10) 操作应该有主次从属关系
- 11) 是否定义 Back 的逻辑。涉及软硬件交互时，Back 键应具体定义
- 12) 是否有横屏模式的设计，应用一般需要支持横屏模式，即自适应设计

2.11 硬件环境测试

2.11.1 手势操作测试

- 1) 手机解锁屏对运行中的 App 的影响
- 2) 切换网络对运行中的 App 的影响
- 3) 运行中的 App 前后台切换的影响
- 4) 多个运行中的 App 的切换
- 5) App 运行时关机
- 6) App 运行时重启系统
- 7) App 运行时充电
- 8) App 运行时 kill 掉进程再打开

2.11.2 网络环境

手机的网络目前主要分为2G、3G、wifi。目前2G的网络相对于比较慢，测试时尤其要注意此

块的测试。

- 1) 无网络时, 执行需要网络的操作, 给予友好提示, 确保程序不出现crash。
- 2) 内网测试时, 要注意选择到外网操作时的异常情况处理。
- 3) 在网络信号不好时, 检查功能状态是否正常, 确保不因提交数据失败而造成crash。
- 4) 在网络信号不好时, 检查数据是否会一直处于提交中的状态, 有无超时限制。如遇数据交换失败时要给予提示。
- 5) 在网络信号不好时, 执行操作后, 在回调没有完成的情况下, 退出本页面或者执行其他操作的情况, 有无异常情况。此问题也会经常出现程序crash。

2.11.3 服务器宕机或出现 404、502 等情况下的测试

后台服务牵涉到 DNS、空间服务商的情况下会影响其稳定性, 如: 当出现域名解析故障时, 你对后台 API 的请求很可能就会出现 404 错误, 抛出异常。这时需要对异常进行正确的处理, 否则可能会导致程序不能正常工作。

2.12 接口测试 (这个一般研发来做, 如下想学习的话, 可以使用 chrome 下的 postman 或者 网上其他工具进行录入查看简单的测试)

服务端一般会提供JSON格式的数据给客户端, 所以我们在服务端需要进行接口测试, 确保服务端提供的接口并转换的JSON内容正确, 对分支、异常流有相应的返回值。此块测试可以采用itest框架进行测试。最方便的是采用httpclient进行接口测试。进行服务端测试时, 需要开发提供一份接口文档。

2.13 客户端数据库测试 (mysql)

- 1) 一般的增、删、改、查测试。
- 2) 当表不存在时是否能自动创建, 当数据库表被删除后能否再自建, 数据是否还能自动从服务端中获取回来并保存。
- 3) 在业务需要从服务端取回数据保存到客户端的时候, 客户端能否将数据保存到本地。
- 4) 当业务需要从客户端取数据时, 检查客户端数据存在时, app 数据是否能自动从客户端数据中取出, 还是仍然会从服务器端获取? 检查客户端数据不存在时, app 数据能否自动从服务器端获取到并保存到客户端
- 5) 当业务对数据进行了修改、删除后, 客户端和服务端是否会有相应的更新。