

2016-3-17

# APP 测试

从入门到精通

测试群：248249896

# 第一章 测试基础培训

## 1.1 测试概念

### 1.1.1 经典定义：

- 1、在规定的条件下对程序进行操作，以发现错误和对软件质量进行评估；
- 2、使用人工或自动化手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是发现预期结果与实际结果之间的差别；

### 1.1.2 扩展定义：

- 1、软件测试就是在软件投入运行前，对软件需求分析、设计规格说明和编码的最终复审，是软件质量保证的关键步骤；
- 2、软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例（包括输入数据与预期输出结果），并利用这些测试用例运行软件，以发现软件错误的过程；

## 1.2 测试流程

- 1、需求分析：阅读需求，理解需求，与客户、开发、架构多方交流，深入了解需求；
- 2、测试计划：根据需求估算测试所需资源（人力、设备等）、所需时间、功能点划分、如何合理分配安排资源等；
- 3、用例设计：根据测试计划、任务分配、功能点划分，设计合理的测试用例；
- 4、测试环境搭建；

5、执行用例：根据测试用例的详细步骤，执行测试用例；

6、执行结果记录和 bug 记录：对每个 case 记录测试的结果，有 bug 的在测试管理工具中编写 bug 记录；

7、回归测试：测试人员查询开发者已修改的 bug，进行回归测试；

8、测试报告：对软件功能的结论，说明为满足此项功能而设计的软件能力以及经过一项或多项测试已证实的能力；

9、用户体验、软件发布等；

## 1.3 Bug 的分类

### 1.3.1 功能

- 1、产品功能是否已实现；
- 2、产品功能是否符合设计要求；
- 3、产品功能是否有重复；
- 4、产品是否有多余的功能；

### 1.3.2 易用性

- 1、界面是否美观；
- 2、操作是否负责；
- 3、是否有相关的帮助文档；

### 1.3.3 安全性

- 1、软件是否有正规的数字签名；
- 2、软件程序是否有加密；
- 3、数据传输时是否有加密；

4、安全性漏洞；

5、系统漏洞；

### 1.3.4 可靠性

1、验证程序中影响软件可靠性的故障，并排除故障实现软件可靠性增长；

2、验证软件当前的可靠性水平是否满足用户的要求；

3、验证软件的数据备份、恢复；

### 1.3.5 性能

#### 1.3.5.1 响应时间

响应时间指的是“系统响应时间”，定义为应用系统从发出请求开始到客户端接收到响应所消耗的时间。

#### 1.3.5.2 最大并发用户数

模拟真实的用户访问，体现的是业务并发用户数，指在同一时间段内访问系统的用户数量，同时向服务器端发出请求的客户数，体现的是服务端承受的最大并发访问数。

#### 1.3.5.3 吞吐量

“单位时间内系统处理的客户请求的数量”，直接体现软件系统的性能承载能力。一般来说，吞吐量用请求数/秒或是页面数/秒来衡量，从业务的角度，吞吐量也可以用访问人数/天或是处理的业务数/小时等单位来衡量。当然，从网络的角度来说，也可以用字节数/天来考察网络流量。对于交互式应用来说，吞吐量指标反映的是服务器承受的压力。

#### 1.3.5.4 性能计数器

描述服务器或操作系统性能的一些数据指标。

### 1.3.6 兼容性

- 1、系统兼容性；
- 2、软件兼容性；
- 3、硬件兼容性；
- 4、web 系统架构，测试浏览器兼容性；

### 1.3.7 可维护性

可维护性指的是软件产品可被修改的能力，这里的修改包括纠正、改进或者软件对环境、需求和功能规格说明变化的适应能力。因此，测试人员在面临各种变更的时候，都需要进行相应的可维护性测试。

- 1、软件产品在用户使用现场发现了缺陷，需要开发团队快速的解决或者提供可替换的解决方案；
- 2、软件产品需要在新的操作环境中使用，当出现新的系统环境，需要对该软件产品进行一定的修改；
- 3、随着用户新需求的不断出现，需要对软件产品根据用户的新需求进行不断的完善；

## 1.4 Bug 的等级

bug 缺陷等级一般划分为四个等级，紧急、严重、一般、低。

### 1.4.1 紧急（一级 bug）

通常表现为:系统无法运行，崩溃或严重资源不足，应用模块无法启动或异常退出，主要功能模块无法使用。

比如：

- 1、系统崩溃（蓝屏）；

- 2、功能设计与需求严重不符；
- 3、系统无法登录；
- 4、产品无法运行；
- 5、内存泄漏；
- 6、错误操作导致的程序中断；

### 1.4.2 严重（二级 bug）

通常表现为：影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统稳定性。

比如：

- 1、功能未实现；
- 2、功能出现异常；
- 3、数据错误；

### 1.4.3 一般（三级 bug）

通常表现为：界面、性能缺陷。

比如：

- 1、操作界面错误；
- 2、提示类错误；
- 3、边界值错误；
- 4、大数据操作时，没有提供进度条；

### 1.4.4 低（四级 bug）

通常表现为：易用性及建议性问题

比如：

- 1、产品的易用性；
- 2、界面的美观度；
- 3、产品说明不明确；
- 4、功能重复；
- 5、提示信息错误；
- 6、新增功能；

## 1.5 Bug 的状态

### 1.5.1 新建缺陷

Bug 创建后的初始状态。

### 1.5.2 确认缺陷

经过确认软件问题后分配给开发人员的状态。

### 1.5.3 测试验证

开发部门对软件问题进行处理或修改后的状态。

### 1.5.4 缺陷退回

开发人员判断软件问题后，将 BUG 状态修改后退回给测试人员

### 1.5.5 缺陷关闭

经测试部门对修改后的软件问题进行验证并确认修改正确后的状态。

## 1.6 Bug 的定位

- 1、产品功能不符合软件需求；
- 2、产品功能未实现；

- 3、产品无法运行操作；
- 4、运行产品导致系统异常；
- 5、产品易用性差；

## 1.7 测试行业术语

单元测试（unit testing）：是指对软件中的最小可测试单元进行检查和验证。单元测试是在软件开发过程中要进行的最低级别的测试活动，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

集成测试：也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求（如根据结构图）组装成为子系统或系统，进行集成测试。

系统测试（System Testing）：针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。系统测试发现问题之后要经过调试找出错误原因和位置，然后进行改正。

验收测试：部署软件之前的最后一个测试操作。在软件产品完成了单元测试、集成测试和系统测试之后，产品发布之前所进行的软件测试活动。

回归测试：指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

白盒测试：又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。

黑盒测试：也称功能测试，它是通过测试来检测每个功能是否都能正常使用。

灰盒测试：介于白盒测试与黑盒测试之间的，可以这样理解，灰盒测试关注输出对于输入的正确性，同时也关注内部表现，但这种关注不象白盒那样详细、完整，只是通过一些表征性的现象、事件、标志来判断内部的运行状态，有时候输出是正确的，但内部其实已



经错误了，这种情况非常多，如果每次都通过白盒测试来操作，效率会很低，因此需要采取这样的一种灰盒的方法。

## 1.8 测试岗位的晋升&发展

# 第二章 测试用例的编写

## 2.1 测试用例的规范

### 2.1.1 目的

统一测试用例编写的规范，以保证使用最有效的测试用例，保证测试质量。

### 2.1.2 范围

适用于公司对产品的业务流程、功能测试测试用例的编写。

### 2.1.3 术语解释

#### 2.1.3.1 测试分析

对重要业务、重要流程进行测试前的分析。

#### 2.1.3.2 业务流程测试用例

关于产品业务、重要流程的测试用例。

### 2.1.4 业务流程测试用例编写原则

#### 2.1.4.1 系统性

1、对于系统业务流程要能够完整说明整个系统的业务需求、系统由几个子系统组成以及它们之间的关系；

2、对于模块业务流程要能够说明清楚子系统内部功能、重要功能点以及它们之间的关系；

#### 2.1.4.2 连贯性

1、对于系统业务流程来说，各个子系统之间是如何连接在一起，如果需要接口，各个子系统之间是否有正确的接口；如果是依靠页面链接，页面链接是否正确；

2、对于模块业务流程来说，同级模块以及上下级模块是如何构成一个子系统，其内部功能接口是否连贯；

## 2.2 测试用例的编写

对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并形成文档。

# 第三章 App 测试纲要

## 3.1 APP 概要

App 是 application 的缩写，通常专指手机上的应用软件，或称手机客户端。手机 APP 就是手机的应用程序。苹果公司的 Appstore 开创了手机软件业发展的新篇章，使得第三方软件的提供者参与其中的积极性空前高涨。随着智能手机越发普及、用户越发依赖手机软件商店，App 开发的市场需求与发展前景也逐渐蓬勃。现除了苹果的 AppStore，还有谷歌的

Google Play Store，诺基亚的 Ovi store，还有黑莓用户的 BlackBerry App World，微软的 Marketplace 等应用商店

## 3.2 操作系统

苹果系统版本 iOS、安卓 Android、微软 windowsphone、塞班系统版本 Symbian、黑莓 BlackBerry

## 3.3 用途

随着智能手机和 iPad 等移动终端设备的普及，人们逐渐习惯了使用应用客户端上网的方式，而目前国内各大电商，均拥有了自己的应用客户端，这标志着，应用客户端的商业使用，已经开始初露锋芒。应用已经不仅仅只是移动设备上的一个客户端那么简单，如今，在很多设备上已经可以下载厂商官方的应用软件对不同的产品进行无线控制。

## 3.4 分类

APP 应用分类：社交应用、地图导航、网购支付、通话通讯、生活消费类、查询工具、拍摄美化、影音播放、图书阅读、浏览器、新闻资讯；

社交应用包括：微信、QQ 空间、人人网等；

地图导航包括：Google 地图、百度地图等；

网购支付包括：淘宝、京东、支付宝等；

通话通讯包括：手机 QQ、阿里旺旺等；

生活消费类包括：58 同城、赶集网、旅游网等；

查询工具包括：墨迹天气、铁路 12306；

拍摄美化包括：美图等；

影音播放包括：多米音乐、百度音乐、QQ 音乐等；

图书阅读包括：91 熊猫看书、QQ 阅读、Adobe 阅读器等；

浏览器包括：UC 浏览器、QQ 浏览器等；

新闻资讯包括：头条新闻、凤凰新闻等；

## 3.5 APP 测试

手机 APP 测试，主要针对的是安卓（Android）和苹果 IOS 两大主流操作系统，主要考虑的就是功能性、兼容性、稳定性、易用性、性能等测试

### 3.5.1 App 安装、卸载

#### 3.5.1.1 App 安装

- 1、点击运行 APP 安装包，检测安装包是否正常；
- 2、进入【安装向导】界面，检测界面是否正常，内容是否有误；
- 3、大部分 APP 默认安装在手机内部存储中，因此要判断内部存储空间是否足够；
- 4、当内部存储空间不足时，是否有相应的提示；
- 5、在运行 APP 安装包时，内存不足是否有提示；
- 6、安装过程中，点击取消安装，是否会退出安装进程；
- 7、安装过程中，接听电话或查看短信等操作时，安装是否会中断；
- 8、安装过程中，接听完电话或者查看完短信等操作时，安装是否会继续；
- 9、安装过程中，前后台切换，是否会影响安装；
- 10、安装完成后查看，默认安装路径是否正确；
- 11、安装成功后，是否会有安装完成的提示；
- 12、安装失败后，是否会有安装失败的提示；
- 13、安装完成后，是否会退出安装界面；

- 14、安装完成后，是否会创建手机快捷方式；
- 15、安装完成后，运行 APP 是否可以运行起来；
- 16、在不同手机系统下安装，是否可以安装正常（ 安卓 Android、苹果 IOS、微软 windowsphone、塞班 Symbian、黑莓 blackberry ）；
- 17、在除 IOS 其他的系统下，当存储空间不足时，是否会检测手机是否有 SIM 卡；
- 18、当有 SIM 卡是，手机存储空间不足时，是否会默认安装在 SIM 卡中；
- 19、在不同的网络环境下安装，是否可以正常安装（ 2G、3G、4G、WiFi ）；
- 20、在没有网络的环境下，安装是否可以正常安装；
- 21、是否支持覆盖安装；
- 22、是否支持第三方工具安装；
- 23、在用户不允许的情况下，是否可以安装；

### 3.5.1.2 App 卸载

- 1、手机卸载有 2 种方式，1 种 APP 自带的卸载，2 种第三方工具的卸载；
- 2、点击运行 APP 卸载，是否可以正常运行；
- 3、当 APP 安装在 SIM 卡中时，运行卸载，是否可以正常卸载；
- 4、卸载时，是否会有确认卸载的提示信息；
- 5、卸载中，取消卸载是否正常；
- 6、卸载中，取消卸载后，APP 是否可以正常使用；
- 7、卸载中，如接听电话或短信等操作时，卸载是否会中断；
- 8、卸载中，如接听完电话或短信等操作时，卸载是否会继续；

- 9、卸载中，前后台切换，是否会影响卸载；
- 10、卸载完成后，查看卸载界面是否退出；
- 11、卸载完成后，手机桌面的快捷方式是否存在；
- 12、卸载完成后，查看 APP 是否卸载干净；
- 13、当用第三方工具卸载时，APP 是否卸载干净；
- 14、在不同手机系统下卸载，是否可以卸载成功（ 安卓 Android、苹果 IOS、微软 windowsphone、塞班 Symbian、黑莓 blackberry ）；
- 15、在不同网络下卸载，卸载是否成功（ 2G、3G、4G、WiFi ）；
- 16、卸载完成后，是否有提示信息；

### 3.5.2 App 运行

- 1、安装完成后运行 APP，确认运行是否正常；
- 2、当 APP 安装在 SIM 卡中时，运行 APP 是否正常；
- 3、网络断开后，app 是否运行正常；
- 4、是否支持多个相同 app 运行；
- 5、在不同网络环境下，运行 app 是否正常；
- 6、在不同系统环境下，运行 app 是否正常；
- 7、在不同硬件环境下，运行 app 是否正常；
- 8、强行关闭 app 后，再次运行 app 是否正常；
- 9、在运行此 app 时，如果有来电、短信等通讯或者充电时，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，继续其功能；
- 10、当 app 运行时间过久时，是否重新进入 app 或者刷新 app 功能；

11、当 APP 运行时，涉及到隐私设置时，是否提示用户，并教用户怎么开启此设置；

### 3.5.3 UI 测试

UI 即 User Interface ( 用户界面 ) 的简称。UI 设计则是指对软件的人机交互、操作逻辑、界面美观的整体设计。好的 UI 设计不仅是让软件变得有个性有品味，还要让软件的操作变得舒适、简单、自由、充分体现软件的定位和特点。手机 APP 从启动界面开始，到运行过程，直至退出，UI 测试都有着自己的规范和要求

#### 3.5.3.1 界面显示

- 1、不同手机系统下，APP 界面显示是否正常；
- 2、不同的手机系统分辨率下，APP 界面显示是否正常；
- 3、APP 界面中，前景色和背景色的搭配是否协调；

#### 3.5.3.2 图形测试

- 1、在不同手机系统下，如 APP 正在运行时，无关的按钮是否有屏蔽（避免操作不方便）；
- 2、控件的焦点与非焦点状态的边框要有明显的区分；
- 3、当 APP 操作时，长时间无响应，是否有对应的界面提示，表明操作正在进行中（如：用户登录、上传、APP 更新、APP 下载等）；
- 4、APP 按钮位置显示是否符合要求；
- 5、APP 各种弹框提示界面是否符合要求，文字内容是否正确；
- 6、APP 运行后，界面中是否出现敏感图片；
- 7、APP 运行后，用户是否可以自定义修改界面风格；

- 8、APP logo 快捷图标是否美观；
- 9、APP 点击进入关于界面中，logo 图标显示是否正常；

### 3.5.3.3 内容测试

- 1、APP 软件名称是否正确；
- 2、APP 整个软件的文字描述是否准确；
- 3、有无错别字；
- 4、是否有敏感词汇出现；
- 5、字体长度、大小是否符合规格；
- 6、APP 界面的易用性是否符合用户要求；

## 3.5.4 App 功能测试

根据软件需求以及用户要求针对 APP 功能进行测试

### 3.5.4.1 登录、注销、退出

- 1、点击登录按钮是否正常；
- 2、使输入正确的用户名和密码，登录是否正常；
- 3、输入错误的用户名登录，登录是否正常；
- 4、输入错误的密码登录，登录是否正常；
- 5、不输入用户名或者密码时，点击登录是否正常；
- 6、被禁用的用户名登录，是否仍然可以登录；
- 7、不存在的用户名登录，是否可以登录；
- 8、不存在的用户名登录，是否会提示“此用户不存在，请重新登录”；
- 9、首次登录后，是否会记录登录用户名；



- 10、登录过的用户名可否删除；
- 11、查看是否有“记住密码”功能；
- 12、勾选“记住密码”，下次登录时，密码是否显示；
- 13、是否有“自动登录”功能；
- 14、勾选“自动登录”，下次 APP 运行时，是否还会显示登录界面；
- 15、登录方式是否与第三方账号绑定；
- 16、选择第三方绑定方式登录，是否正常；
- 17、第三方登录后，检测数据同步是否正常；
- 18、无网络情况下登录，是否可以正常登录；
- 19、登录超时，是否有弹框提示；
- 20、相同用户名，是否支持多个手机登录；
- 21、不同系统环境下登录，是否正常；
- 22、不同网络环境下登录，是否正常；
- 23、当登录成功后，点击注销，是否会退出游戏界面，返回登录界面；
- 24、点击注销时，是否给用户弹框提示；
- 25、点击取消注销时，是否会返回 APP 界面，继续 APP 运行；
- 26、退出登录时，是否会退出当前用户，返回登录界面；
- 27、点击退出时，是否给用户弹框提示；
- 28、点击取消退出时，是否会返回 APP 界面，继续对 APP 运行；

### 3.5.4.2 前后台切换

- 1、是否可以正常切换 APP 前后台操作；

- 2、当 APP 切换到后台时，APP 界面是否仍然显示；
- 3、当 APP 切换到后台时，APP 是否会正常运行；
- 4、当 APP 切换到后台时，APP 是否退出运行；
- 5、当 APP 切换到后台时，APP 是否出现异常现象；
- 6、当不同系统手机下切换，APP 是否可以正常切换；
- 7、当 APP 又切换到前台时，APP 界面是否可以正常显示；
- 8、APP 频繁前后台切换，是否会导致 APP 异常；
- 9、当 APP 正在使用时，来短信、微信时，是否将 APP 切换到后台；
- 10、当 APP 正在使用时，来电话并挂断时，APP 功能是否正常；
- 11、当手机锁屏并解锁时，APP 界面是否正常显示；
- 12、当手机锁屏并解锁时，APP 功能是否正常；
- 13、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；
- 14、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；
- 15、强行退出 app 进程后，再开启 app，app 能否正常启动；

### 3.5.4.3 界面功能

- 1、界面工呢过是否符合用户要求；
- 2、不同分辨率下，界面显示是否正常；
- 3、触摸点是否正常；
- 4、界面动画是否符合要求；

- 5、背景音效是否符合要求；
- 6、界面功能是否符合用户要求；
- 7、当没有网络时，界面功能是否仍然可是使用；

更详细的 APP 功能测试根据实际情况来进行测试

### 3.5.5 App 性能测试

#### 3.5.5.1 响应时间

- 1、APP 安装、卸载的响应时间；
- 2、APP 各种功能操作的响应时间；
- 3、APP 前后台切换的响应时间；
- 4、APP 登录后获取个人信息的响应时间；
- 5、Web 服务器处理的响应时间；
- 6、APP 服务器的响应时间；

#### 3.5.5.2 最大并发用户数

- 1、同一时间段内访问系统的用户数量；
- 2、同时向服务器端发出请求的客户数；
- 3、同时在线用户数；

#### 3.5.5.3 压力测试

- 1、不同系统环境下，做压力测试；
- 2、不同的网络环境下，做压力测试；
- 3、不同的硬件环境下，做压力测试；
- 4、电量不足时，运行 APP 会挺多长时间；

- 5、手机存储不足时，APP 是否可以安装；
- 6、手机 SIM 卡存储不足时，APP 是否可以安装；
- 7、APP 安装时，手机存储不足，是否会出现异常；
- 8、APP 安装时，手机 SIM 存储不足，是否会出现异常；
- 9、内存不足时，APP 是否能运行起来；
- 10、APP 运行时，超出了内存范围，是否会出现异常；
- 11、多次安装卸载 APP，测试 APP 是否出现异常情况；
- 12、多次安装卸载 APP，测试是否有卡顿的现象；
- 13、频繁点击 APP 功能，测试 APP 是否出现异常情况；
- 14、频繁点击 APP 功能，测试是否有卡顿现象；
- 15、长时间 APP 不退出，测试是否有出现异常情况；
- 16、长时间 APP 不退出，测试是否有卡顿现象；
- 17、长时间 APP 不退出，是否会占用网络流量；
- 18、长时间 APP 不退出，是否会导致内存溢出；
- 19、多次切换前后台，是否有卡顿现象或异常情况；
- 20、多个用户同时像服务器发送请求，服务器是否运行正常；
- 21、APP 的 CPU 检测；

### 3.5.6 交叉事件测试

交叉事件测试又叫事件或冲突测试，对于正在运行的应用，若进入短信、电话等其他软件响应的情况，不会影响所测试应用，且会保证应用都能正确运行；

- 1、前后台切换是否影响功能使用；

- 2、当有短信、电话等其他软件进入时，是否会影响功能正常使用；
- 3、当发短信、打电话等其他软件输出时，是否会影响功能正常使用；
- 4、当切换不同的网络环境时，是否会影响功能使用（2G\3G\4G\WIFI）；
- 5、当 APP 正在运行时，出现电量不足提示时，是否会影响 APP 使用；
- 6、当手机没电正在充电时，是否会影响 APP 正常使用；
- 7、当 APP 支持其他语言的操作时，切换其他的语言，是否会出现乱码等现象；
- 8、检测 APP 是否依据各事件的优先级别依次进行处理。；

### 3.5.7 App 更新测试

- 1、当有新版本更新时，是否会有提示；
- 2、不同手机系统环境下，APP 升级是否正常；
- 3、不同的网络环境下，APP 升级是否正常；
- 4、APP 更新方式有三种：自动更新、手动更新、定时更新；
- 5、自动更新，登录 APP 界面，是否会校验是否有更新；
- 6、登录 APP 界面后，弹框提示更新，当点击取消时，是否还继续更新；
- 7、当取消升级时，是否有时间限制（例如“2 小时一提示”）；
- 8、登录 APP 界面后，提示更新，点击更新（或确定）后，是否正常更新；
- 9、APP 升级过程中，手机断电，再次运行 APP 时，功能是否可用；
- 10、APP 升级过程中，如来电或短信等操作时，升级是否会中断；
- 11、APP 升级时，切换到后台，升级是否会中断；
- 12、登录 APP 界面后，更新，网络断开，是否会提示更新失败；
- 13、当 APP 升级成功后，APP 功能是否正常使用；

- 14、当 APP 升级成功后，APP 界面是否正常；
- 15、升级后的 APP 功能是否符合软需或者用户的要求；
- 16、升级后的 APP 是否解决了之前所提交的 BUG；
- 17、当 APP 升级失败时，APP 功能是否正常使用；

### 3.5.8 App 兼容性测试

- 1、不同网络环境兼容性测试；
- 2、不同手机操作系统兼容性测试；
- 3、不同应用软件的兼容性测试；
- 4、不同的容量大小的 SIM 卡之间的互相兼容测试；
- 5、当安装杀毒软件时，应用是否可以正常使用；
- 6、不同手机品牌兼容性测试；
- 7、手机大小屏幕分辨率兼容性测试；

### 3.5.9 App 回归测试

- 1、针对提出的所有 BUG 进行重新验证；
- 2、APP 安装、卸载验证；
- 3、APP 功能验证；
- 4、APP 兼容性验证；
- 5、APP 易用性验证；

## 第四章 App 安全性测试

安全性测试的目的是发现危害手机中数据的安全和完整性的错误和缺陷。发现安全错误通常是比较困难的，软件通常功能运行正常但却不安全。

## 4.1 软件权限

APP 软件权限包括：网络通信、信息发送、自动启动、媒体录制、读取用户信息、写入用户数据等权限，因关系到用户个人信息和隐私的保护，需对软件权限和访问安全机制进行测试。

### 4.1.1 网络通信

- 1、网络访问权限：限制/允许手机使用网络；
- 2、网络传输时，确认通信机制是明文传输还是密文传输；
- 3、在手机 APP 需要网络连接时，弹框提示用户，用户确认后方可连接；

### 4.1.2 信息发送

- 1、限制/允许手机使用发送接收信息功能；
- 2、发送或接收的信息是否是加密过的；
- 3、在手机 APP 使用接发信息功能时，弹框提示，用户确认后才可使用；

### 4.1.3 自动启动权限

- 1、限制/允许应用程序来注册自启动应用程序；
- 2、是否自启动应用程序，需弹框提示用户，方可进行操作；

### 4.1.4 媒体录制权限

- 1、限制/允许 APP 软件使用手机拍照或录音；
- 2、运行 APP，如需要使用手机拍照或录音，需弹框提示用户方可进行操作；

### 4.1.5 读取用户信息

- 1、限制/允许 APP 软件使用手机读取用户信息；

#### 4.1.6 写入用户数据

- 1、限制/允许使用手机写入用户数据；

### 4.2 安装&卸载

#### 4.2.1 安装

- 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、JAD 文件和 JAR 包中包含的所有托管属性及其值必需是正确的；
- 5、JAD 文件显示的资料内容与应用程序显示的资料内容应一致；
- 6、安装路径应能指定；
- 7、没有用户的允许， 应用程序不能预先设定自动启动；
- 8、卸载是否安全， 其安装进去的文件是否全部卸载；

#### 4.2.2 卸载

- 1、卸载是否安全，安装的文件是否卸载干净；
- 2、卸载时，用户在使用过程中产生的文件是否有提示确认；
- 3、其修改的配置信息是否复原；
- 4、卸载是否影响其他软件的功能；
- 5、卸载应该移除所有的文件；
- 6、卸载后快捷方式是否删除；

### 4.3 数据安全性



数据安全性测试是最重要的测试内容，数据安全是安全的核心内容。数据安全测试应从密码显示、存储、数据备份和恢复、安全提示等方面进行有效的测试，以确保用户使用的安全性

- 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、输入人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，文件可能遭受入侵者的袭击，然后读取这些数据信息；
- 6、当将敏感数据输入到应用程序时，其不会被储存在设备中；
- 7、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；
- 8、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；
- 9、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；
- 10、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；
- 11、“取消”命令操作能够按照设计要求实现其功能；
- 12、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；

13、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；

14、在没有用户明确许可的前提下不损坏或删除个人信息管理应用程序中的任何内容；

15、应用程序读和写数据正确；

16、应用程序应当有异常保护；

17、如果数据库中重要的数据正要被重写， 应及时告知用户；

18、能合理地处理出现的错误；

19、意外情况下应提示用户；

## 4.4 通讯安全

1、在运行其软件过程中， 如果有来电、短信等通讯或充电时， 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 HTTPClient 的 library 异常作捕获处理。

## 4.5 人机接口安全

人机接口包括交互界面菜单、接口命令、声音等，是负责与用户相连接的接口，为避免用户无意输入的可干扰程序正常运作的错误数据，因此要对所有的输入之前进行检查并确认有效。

- 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；

## 4.6 测试通信处理能力

- 1、多连接，小数据量测试

客户端采用多线程与服务连接，采用多台终端同时与服务连接，每台终端同时建立不少于 1000 个连接，观察服务通信是否正常

- 2、常连接、大数据量测试

每个客户端与服务建立一条长连接，同时发送大数据量数据（如每次 1M 数据），观察服务能否正确接收

## 4.7 数据处理能力

1、每次连接时启动计时，在一定的时间内观察服务是否正确返回，统计交易成功率

2、碎片包测试：客户端把一个完整交易包分割成多个碎片包发送，观察服务能否正确组合和响应

## 4.8 可靠性测试

多台客户端同时与服务连接并自动发送交易数据，交易数据有单包，多包，碎片等模式，连接通信 7X24 小时，统计服务的交易成功率。

# 第五章 常用 App 测试工具

## 5.1 监控工具

DDMS 的全称是 Dalvik Debug Monitor Service，是 Android 开发环境中的 Dalvik 虚拟机调试监控服务。

## 5.2 压力工具

Monkey 介绍

Monkey 测试是 Android 自动化测试的一种手段，Monkey 测试本身非常简单，就是模拟用户的按键输入，触摸屏输入，手势输入等，看设备多长时间会出现异常。

Monkey 是一个命令行工具，可以运行在模拟器里或实际设备中。他向系统发送伪随机的用户事件流，实现对正在开发的应用程序进行压力测试。

Monkey 基本用法

可以通过开发机器上的命令行或脚本来启动 Monkey。由于 Monkey 运行在模拟器/设备环境中，所以必须用其环境中的 shell 来进行启动。可以通过在每条命令前加上 adb shell 来达到目的，也可以进入 Shell 后直接输入 Monkey 命令。

基本语法如下：

```
$ adb shell monkey [options]
```

如果不指定 options，Monkey 将以无反馈模式启动，并把事件任意发送到安装在目标环境中的全部包。下面是一个更为典型的命令行示例，它启动指定的应用程序，并向其发送 500 个伪随机事件：

```
$ adb shell monkey -p your.package.name -v 500
```

可以尝试的去操作一下

## 第六章 App 测试实战讲解

### 6.1 管理平台的使用

常用的管理平台有：QC ( Quality Center )、BugFree、.Bugzilla

#### 6.1.1 QC ( Quality Center )

是原 Mercury Interactive 公司(现已被 HP 收购)生产的企业级基于 WEB 测试管理工具，需要安装配置 IIS 和数据库，系统资源消耗比较大；功能很强大，结合有 BUG 管理，需求管理及用例管理等功能；和其它的测试工具，比如 Loadrunner 测试工具的接口做得比较好，数据可以在它们中共享；英文版的且易用性不是很好，最重要的是收费且价格不菲；XX 版的网上也很多但找起来和 XX 也比较费事，且性能就不那么稳定。

## 6.1.2 BugFree

基于 WEB 的，配置安装简单，只需到网上获取安装包，再配下 PHP 通用的环境即可；纯功能型的界面就无所谓美观；没有直接的截图功能但是可以以附件的形式存在；也有简单的报表统计功能；整体使用还是比较容易上手，而且是开源免费中文版的 BUG 管理系统。

## 6.1.3 Bugzilla

Bugzilla 是由 Mozilla 公司提供的基于 Web 方式，免费的开源的一款功能强大的 Bug 管理系统，比如强大的检索功能，强大的后端数据库支持，丰富多样的配置设定等；安装需要 Perl 和配置 MYSQL 数据库，过程比较繁琐，修改配置文件比较麻烦；英文版的，能汉化但是汉化后容易出现乱码；

# 6.2 测试文档的准备

## 6.2.1 软件需求文档

一般软件测试的话，前期一般都是用户需求规格说明书的了解，只有了解了产品的功能和性能，才能开展工作。

## 6.2.2 测试计划

根据需求规格说明书，编写测试计划，当中包括测试用例的编写。

## 6.2.3 执行用例

执行测试用例，对产品的功能性能进行测试，发现 bug，提交 bug，跟踪 bug，记录下 bug。

# 6.3 测试环境的搭建

测试环境是测试实施的一个重要阶段，测试环境适合与否会严重影响测试结果的真实性和正确性。现针对苹果 IOS 和安卓 Android 两大系统环境进行测试环境的搭建

### 6.3.1 苹果 IOS

- 1、针对有苹果机的测试人员，只需要在 MAC 系统下，安装 x-code 软件即可
- 2、非苹果机的测试人员，首先需要在 Windows 下安装虚拟机，再安装苹果的 mac 系统，最后安装 x-code 软件即可

### 6.3.2 安卓 Android

#### 6.3.2.1 测试软件

- 1、JAVA JDK
- 2、ECLIPSE
- 3、ANDROID SDK
- 4、Python ( 可选 )

#### 6.3.2.2 准备工作

##### 6.3.2.2.1 JAVA JDK 安装

- 1、下载并安装 Java JDK

下载链接：<http://java.sun.com/javase/downloads/index.jsp>

- 2、JDK 环境变量配置有 2 种方式：我的电脑--属性--高级--环境变量

(1)新建用户环境变量 JAVA\_HOME ，设置为 java 安装地址

" C:\Program Files\Java\jdk1.6.0\_17 " ；

在系统变量 PATH 增加

";%JAVA\_HOME%\bin" ；

在用户变量 CLASSPATH 增加

“.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar”。

(2)直接系统变量 PATH 增加

“;C:\Program Files\Java\jdk1.6.0\_17\bin ” ;

直接用户变量 CLASSPATH 增加

;C:\ProgramFiles\Java\jdk1.6.0\_17\lib;C:\ProgramFiles\Java\jdk1.6.0\_17\lib\tools.jar

3、验证环境变量

开始=>运行，输入 cmd，进入 DOS 界面

如果 Java 安装在 c:盘，直接运行 java -version

注：Java 安装在什么目录下，就在什么目录下运行

#### **6.3.2.2.2 ECLIPSE**

1、下载并安装 ECLIPSE

下载链接：<http://www.eclipse.org/downloads/>

#### **6.3.2.2.3 Android SDK**

1、下载并安装 Android SDK

下载链接：<http://developer.android.com/sdk/index.html>

2、Android SDK 环境配置

我的电脑--属性--高级--环境变量

直接在系统变量 PATH 增加



;D:\soft\T\_TOOLS\android-sdk-windows\tools;D:\soft\T\_TOOLS\android-sdk-windows\platform-tools"

注：均已分号（；）隔开

#### 6.3.2.2.4 Python

1、下载并安装 Python

下载链接：<http://www.python.org/getit/>

2、Python 环境配置

我的电脑--属性--高级--环境变量

直接系统变量 PATH 增加 ";E:\python"

### 6.3.2.3 环境搭建

#### 6.3.2.3.1 安装 Android 开发插件

1、打开 Eclipse.exe，help=>Install New Software

2、点击 add

Name: Android (可自定义)

Location:[http\(https\)://dl-ssl.google.com/android/eclipse/](http(https)://dl-ssl.google.com/android/eclipse/)

输入完后，点击 OK

3、进行下一步，选择 I accept the terms of the license agreements

点击 Next，进入安装插件界面

4、完成插件安装后，重启 Eclipse

#### 6.3.2.3.2 配置 Android SDK

- 1、点击菜单 windows=>preferences，选择 Android SDK 的安装目录，点击 OK
  - 2、Android SDK 升级，点击菜单 windows=>Android SDK Manage，勾选需要安装的 SDK 版本，点击可下载更新
  - 3、AVD(Android Vitural Device)创建，点击菜单 windows=>Android Vitural Device Manage，点击 new，创建一个 AVD
- AVD Name：自定义
- Device: 选择分辨率
- Target: 选择 SDK 版本
- CPU/ABT: 电脑芯片
- SD Card: SD 大小（一般不要超过 900M）
- Android 测试环境搭建完成。

## 6.4 测试流程开始

- 1、测试需求：根据用户需求来评定测试员需要测试的内容；
- 2、测试计划：根据测试需求来制定测试计划，即时间安排，人员安排以及硬件安排等；
- 3、测试用例设计：设计测试用例，用以指导测试并可直观看出你测试的覆盖率；
- 4、测试环境搭建；
- 5、用例执行；
- 6、提交 BUG；
- 7、回归测试；

8、测试总结：即完成一轮测试后，需要根据 BUG 分布来分析软件薄弱点在何处，  
以方便后续测试计划的制定；