1. App端的基本功能，和其他C/S架构的程序的client是一样的。（安装/卸载/升级测试）
2. APP内部功能测试
3. 兼容和适配的问题。

   兼容和适配包含了几个方面：

     - 硬件的适配。 比如硬件的性能，屏幕的大小，一些依赖的设备比如GPS等。

     - OS版本的兼容，ios和android都有一样的问题，比如如果用了一些新的API在老的系统上不支持会导致crash。

     - 屏幕的分辨率适配。移动设备的分辨率多种多样，如果app没有做比较合适的处理就可能会显示不好，甚至影响功能的操作。

     要做到比较好的覆盖，这些都是很耗费时间的。现在想到的办法主要有4种：

* 自行购买或者借用设备来实际验证。但是受限于财力人力不可能做得很全面。
* 考虑使用第三方云测试的解决方法，比如testin.cn这种，可以提供基本的运行情况和一些截图，有助于扩大测试的范围。
* 比较白盒的方法。将不兼容的地方整理出来，然后去分析我们的app中可能不兼容的地方。但是对团队的技术能力的要求比较高，前期也需要花费不少的时间。
* 最后是收集用户的反馈，亡羊补牢。

4.网络制式.

Gprs/E-gprs/3G/4G/wifi/热点

5.用户操作习惯

单击/双击/快速连击/左右滑动/上拉/下拉/

6.系统后台关闭（进程结束、再次进入APP重新加载界面。）

**其他测试方法：**

1. app crash的问题

     crash，或称为闪退，原因有很多，针对这一部分出来很重要的是能收集到crash的问题，做事后的修补。所以需要确认我们的app有crash上报的能力，无论是公司内部的还是第三方的平台，我们需、要定期的知道外网的app crash的次数和crash的基本信息，帮助我们定位和修复。

2. App端本身的性能分析，内存泄漏的分析。

3. 代码覆盖率分析的方法也是很好的参考，无论是App端还是后台服务端。

4. 灰度发布的方案（大型系统）。保护app端发布和提交app store的灰度，也可以是自动更新的提示的灰度。后台服务端也可以做灰度，类似于网站的做法，不过要考虑和app的兼容性。