**关于自动化测试的说明**

# 1.测试目的

由于目前测试人员较少，为了提高功能测试的效率，针对项目当前实际情况，先做app的IU自动化。主要目的是发布新版本时，通过自动化脚本，遍历app功能，确保版本的修改不会对原有的功能造成影响。

这样做的好处是：一提高工作效率，不可能每次迭代新版本就将所有功能都收工测试，但可以通过自动化脚本把功能遍历。二避免手工测试时出现漏测，手工测试时，有些场景可能遗漏。

总之，自动化测试的目的是提高效率的同时保证产品的质量。

# 2.测试框架

目前，主流的自动化框架综合对比情况如下：



Appium越来越成熟也逐渐成为app自动化测试的最佳选择，其优点是：

跨框架，native hybrid webview

跨设备，Android IOS firefoxOS

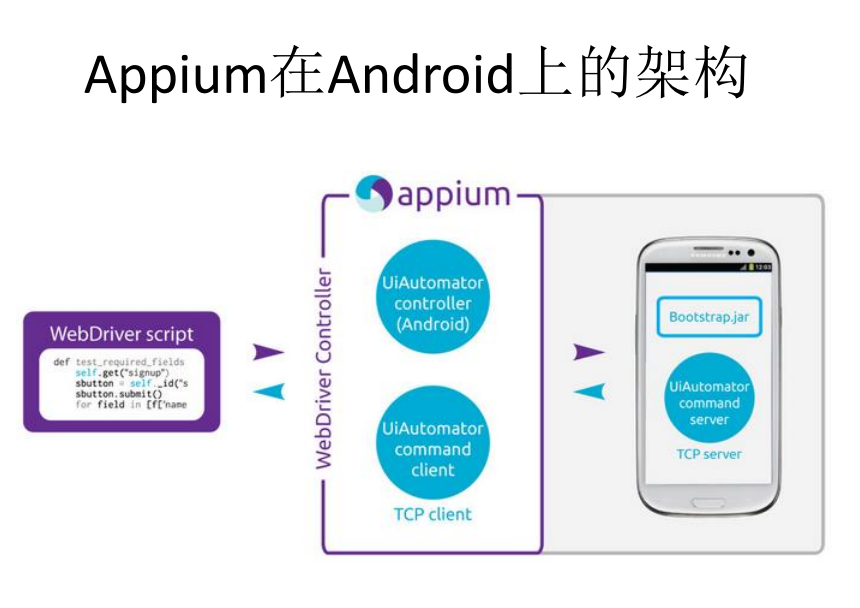
跨语言,java，python ruby nodejs php

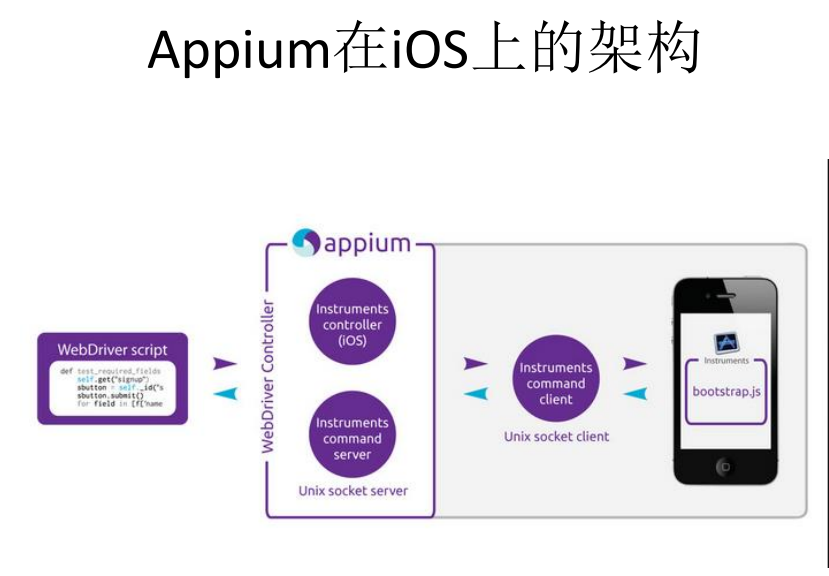
跨app，可以支持多个app之间的交互（如跳转到微信分享）

总之不依赖源码，不限制测试平台和框架，appium已经越来越成熟。框架毫无疑问选择APPium。

在脚本语言的选择上，目前基本上集中在java和Python上。我选择python，相对来说，语法更加简单，好写一点，而且Python作为脚本语言的库更加丰富。

在appium的设备端即可以连接真机测试也可以连接模拟器，以达到模拟多种机型的目的。





# 3.测试环境和工具

|  |  |  |
| --- | --- | --- |
| **平台工具** | **本版号** | **备注** |
| java | 1.7.0\_67 |  |
| appium | 1.4.16 |  |
| python | 3.5.2 |  |
| 模拟器 | 夜神Android模拟器 |  |
| 真机 | 华为，iphon5 |  |

# 4.测试脚本和测试用例

测试用例遍历七彩芯，设置，我，以及导航栏的所欲模块。主要遍历所有正常功能操作，所有正常提交，页面信息检查等。暂时不做对外接口的UI自动化，如缴费，指纹录入，只查看到对应页面，不做对应提交。

目前已经完成的Android app的初版测试脚本，如下：

daohanglan.py --主要是导航栏业主审核相关模块，呼叫中心和通知模块的操作。

qicaixin.py --主要是七彩芯页面各个功能模块的操作

wo\_and\_shezhi.py --主要是我和设置模块的相关功能操作

# 自动化测试流程

开发完成脚本迭代开发-->monkey测试--->新功能手工测试—>已有功能自动化测试-->版本发布-->将新开发功能补充到自动化测试脚本中。

# 6.后续优化和工作

1. IOS自动化脚本编写。
2. 将脚本进行封装，加入更多检查和判断等进行优化。

# 7.总结

这是我工作中首次实施app自动化测试，一切是零学起。目前也主要是IU做起，后续看情况是否还需要更多的深入。IU测试只是这个框架的一小部分，但也是主要应用的部分。随着产品的推进和成熟，也可以对自动化的框架进行更好的封装和分离，满足变更的需求。