**[移动应用自动化测试](http://wiki.primeton.com/pages/viewpage.action?pageId=10848259)**

**一.移动应用开发特点**

1、团队结构比较精简，测试人员配备较少；

2、并且团队以技术导向型为主，需求频繁变动导致测试成本很高；

3、产品发布周期较短，其中经历多次迭代测试，测试时间不够充分；

4、移动电子产品百家争鸣，应用对设备的依赖使得兼容性测试成本大大增加。

在以上特点的局限下，移动应用出现如下“囧境”，如数量增长过快、质量难以保证、难以脱颖而出、盈利困难等等。

**二.移动应用测试的问题与难点**

1、移动应用从数量上和逻辑复杂程度上的增长，以及产品发布周期的紧缩，使得快速回归测试迫在眉睫；

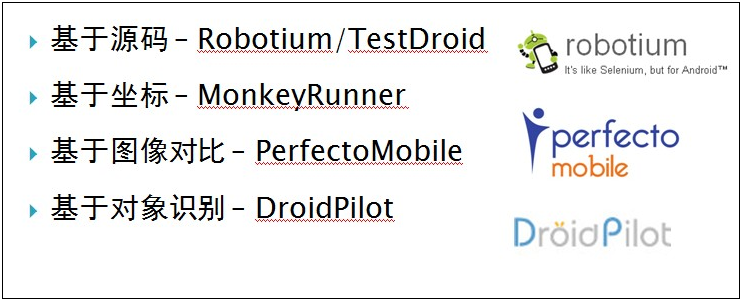
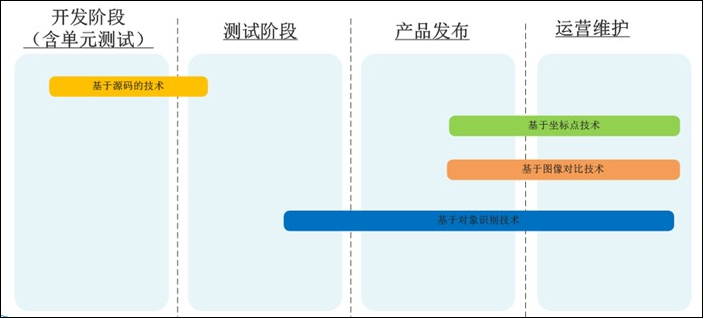
2、安卓系统的开放性造成硬件厂商百家争鸣的局面，设备款式之多，在不同平台/批量设备上的兼容性测试比较困难；

3、安卓碎片化问题，安卓设备厂商很多，测试无法兼顾

4、刚起步，尚处于探讨与研究阶段，没有像传统pc自动化测试一样具有HP QTP、Rational Robot、Rational Functional Tester、Borland SilkTest 、Selenium (opensource)等工具的支持。

**三.移动应用测试的技术与产品**

 1、移动应用自动化测试技术分类--主流移动应用自动化测试技术

    
2、各类技术的实现原理--各类技术的对比 (\*号表示图像在不同屏幕尺寸的设备上必须大小一致)  
  
3、各类技术在开发周期中的作用  
  
4、各类技术能解决的关键难题  
     基于源码 – 单元测试阶段及前期功能测试阶段的逻辑验证  
     基于坐标点 – 快速点击、性能验证  
     图像对比 – 跨平台、兼容性测试  
     对象识别 – 深层测试、逻辑验证、使用直观、模拟人工测试  
   
5、 DroidPilot自动化测试技术的特性和优势  
      DroidPilot是一款安卓应用自动化测试工具,用于功能回归测试和兼容性测试。它具备以下特性：  
      第一, 基于智能映射对象识别技术,而非屏幕坐标点操作或图像识别。  
      第二, 测试脚本一次编译,即可运行于不同设备。测试脚本与屏幕尺寸和Android OS版本无关。  
      第三, 多设备之间的通信完全使用Wifi连接。  
      第四, 测试脚本的设计沿用了业界流行的关键字驱动概念,语法类似VBScript    
      与源码技术相比，对象识别技术更贴近测试人员使用，而非开发人员使用；与坐标点相比，对象识别技术的脚本适应性更高；与图像对比技术相比，对象识别技术脚本制作更简单、执行效率更高  
**四.深入DroidPilot的原理**

**1、关键字驱动**

       DroidPilot是基于关键字驱动的自动化测试工具，关键字测试的主要思路是以面向对象的方式来管理被测应用的对象、对象的相关操作、测试数据以及 这些测试数据之间的组合关系。关键字驱动是自动化测试中行之有效的方式，它可以帮助测试工程师更方便的维护测试脚本、构建复杂的业务逻辑测试用例、并节省 手工测试的执行时间（尤其是在回归测试阶段）。  
      关键字驱动主要由以下三种元素构成：  
     （1）、被测对象，即被测应用界面上的元素  
     （2）、针对这些对象的操作，如点击（按钮）、填充（文字）、选择（单选框/多选框）  
     （3）、以及基于这些操作的数值

**2、对象库**

      对象库是用于储存被测应用程序界面对象（界面元素）的地方。它是关键字驱动测试工具的关键点。有了它，用户可以更容易的维护被测对象、更快速的构建测试脚 本。如果对象都保存在对象库，它们可以被集中管理。与此同时，测试脚本（一个或多个）可以根据自身也许需求随意调用这些对象来使用。这使得多位测试工程师 协同开发测试脚本成为可能。

      对象库也可以分为两层，一层面向用户视角，另一层面向程序视角。这两层之间的对象是一一对应关系。这样做有助于测试脚本（测试用例）独立于被测应用程序。 例如，应用程序的界面对象Obj 1改变了，但所有引用了该对象的测试脚本无需改动，因为它们是引用的对象Obj A，而非直接引用Obj 1。

**3、测试数据组织结构**

      一般提到测试用例，通常指的是一系列操作步骤以及针对这些步骤所用到的测试数据。测试数据，通常可以理解为填入被测表单各个字段中的数值。这些数值需要被 很好的组织和保存，以防止数据冗余。所以测试数据可以像对象那样被保存于数据库中，也可以存在表格文件，如Excel或.csv文件。这些表中的字段（表 结构）需要跟被测表单中的字段一一对应。在每次测试迭代中，测试用例从数据表中挑选其中一条数据用以执行，所以，表格中的数据需要设计为可重复利用的。

**4、DroidPilot结构**

1、代理组件(Agent)

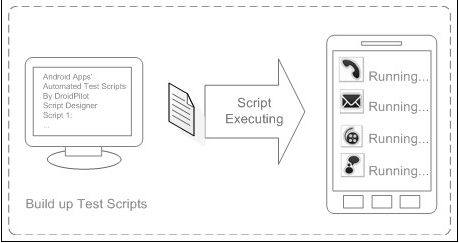
     DroidPilot代理组件(Agent)其实是一个安卓应用程序(Agent.apk)。它被安装于被测设备，用于连接被测应用与控制程序，使用者可 以在DroidPilot安装目录DroidPilot\DP-win32-v2.1.0\droidpilot-designer-Win32- RV2.1.0\windows下找到Agent.apk。

     2、DroidPilot部署工具

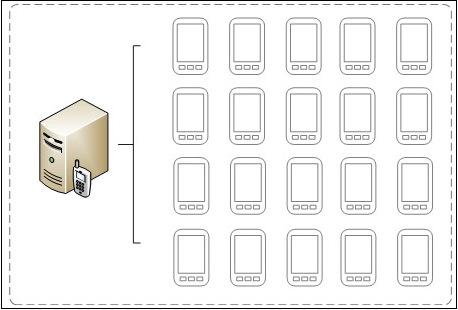
     此应用于部署一个或多个被测应用到被测设备，连接的方式有两种：  
     （1）USB连线部署：用USB连接，则默认IP地址是“127.0.0.1”  
     （2）Wi-Fi无线部署：用Wi-Fi连接，需输入设备IP地址。端口号也必须与设备端输入的号码一致，该号码默认是“8888”

     3、脚本编辑器(Script Designer)

      脚本编辑器通过对移动应用程序捕获GUI对象并使用关键字驱动功能向关键字视图中手动添加测试步骤来创建测试或组件。然后，可以在专业脚本编辑器中使用特 殊的编程语句来修改测试或组件生成的脚本。并可以通过USB连接或Wi-Fi连接，连接单台设备调试测试脚本，用于功能性回归测试。

        
     4、脚本执行器(Script AutoRunner)

      AutoRunner 提供支持多设备的测试服务。可以一套测试运行于各种Android设备；或多套测试运行于多套设备；或多套测试脚本运行于一套设备。可以通过点击 “停止”按钮随时停止批处理测试运行。

       
      **5、DroidPilot运行原理**

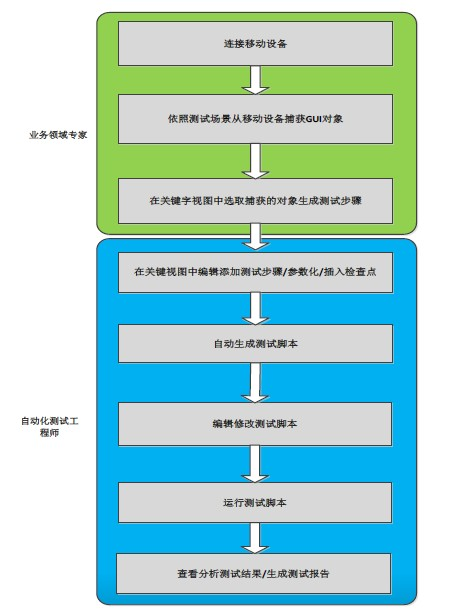
      DroidPilot代理组件(Agent)用于连接被测应用与控制程序-------脚本编辑器(Script Designer)和脚本执行器(Script AutoRunner)。这种连接是通过安卓(Android)签名机制实现的。

      如果代理组件(Agent)与被测应用(APK Under Test)具有相同的签名，那么，这两者之间就会存在信任关系。如此，代理组件与被测应用之间的通信就成为可能。再者，代理组件与控制程序之间存在衍生的 信任关系，就使得控制程序与被测应用之间可以畅通无阻的通信。如此，控制程序就可以通过代理程序，抓取被测应用的界面对象，截获被测应用的行为，甚至控制 被测应用的行为。

      正因为以上关系，才需要被测应用必须通过代理程序部署于被测设备。在部署过程中，代理程序会为被测设备增加一个临时签名。这种做法并不 会破坏被测应用本身的代码结构和运行机制，因此，被测应用通过代理程序安装，与直接安装后的行为并不会有差别，这是验证被测应用功能性是否可靠的基础。即 是说，通过代理程序执行的自动化测试，与人工操作的手工测试，两者本身都是可靠、可重复、可验证的。即便如此，代理组件也需要被测应用必须是未加扰码的应 用，即被测应用必须是测试版本，而非正式投放市场的发布版本。因为被测应用一旦加入扰码，代理组件对于被测应用界面对象与行为的识别将会收到干扰，无法正 确重现界面对象和操作步骤，这样就会导致测试脚本的回放结构具有不确定性。

      总而言之，代理组件是被测应用与控制程序之间的一座桥梁，承接这两者之间的通信之职责，是DroidPilot工具套件的重要组成部分。

**6、DroidPilot测试流程**

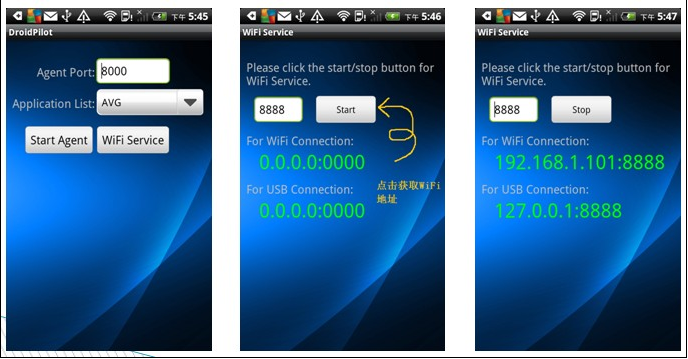
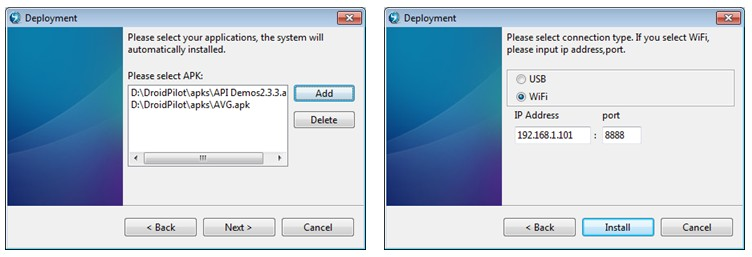
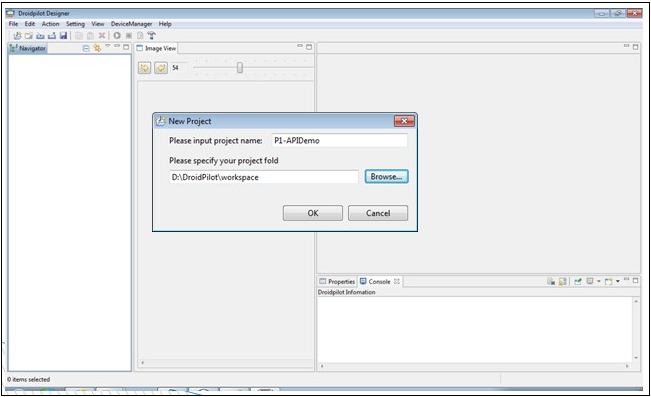
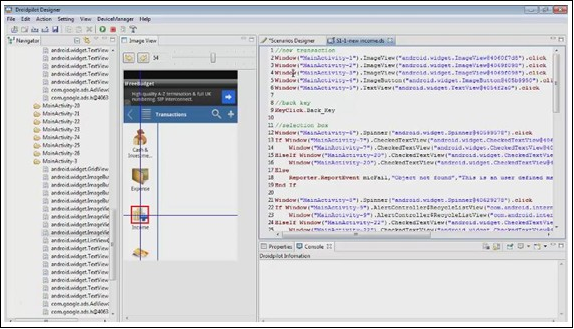
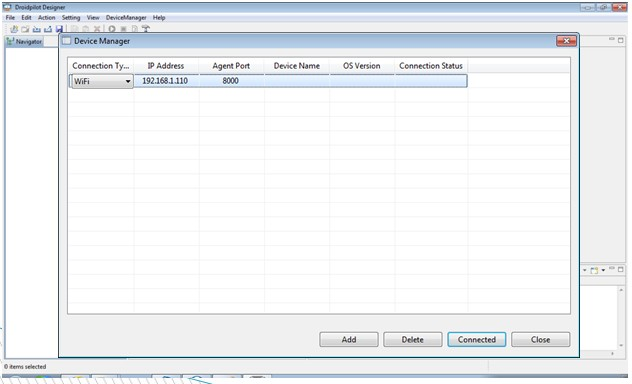
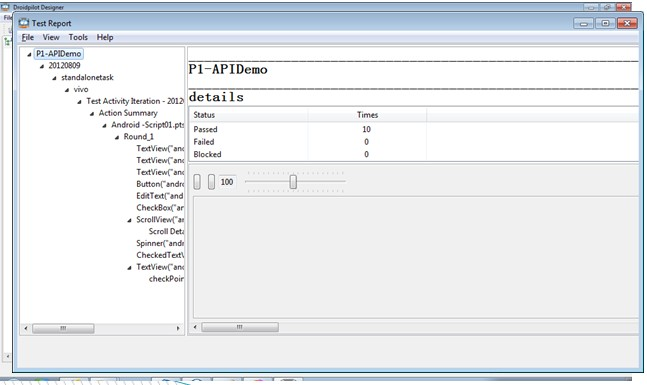
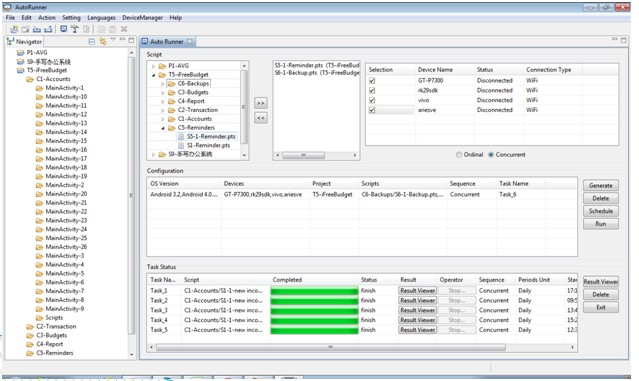


**五. DroidPilot案例开发**

**1、下载和安装**

(1)下载  
  到DroidPilot官网[www.droidpilot.cn](http://www.droidpilot.cn/)下载安装包DroidPilot\_V2.1.0\_Win32\_Setup.exe  
  (2)安装  
  a)启动DroidPilot\_V2.1.0\_Win32\_Setup.exe的安装；  
  b)在被测设备安装Agent.apk应用。

**2、Agent获取被测设备的WIFI地址**  
   被测设备中启动Agent应用，获取被测设备的WIFI地址。

  
**3、部署工具部署被测应用到被测设备**  
 在DroidPilot安装目录启动APK Deployer.exe，在部署工具中选择被部署的应用(如：iReader.apk)，然后选择WIFI连接，并配置WIFI的IP地址和端口号。  
  
**4、启动被测应用**  
   被测应用部署完成后，便可以启动被测应用。  
  
**5、启动脚本设计器和开发脚本**  
在DroidPilot安装目录启动Script Designer.exe，启动脚本编辑器。  
****  
在脚本编辑的资源管理器中右键，选择“新建项目”，完成项目的创建。  
****  
制作测试脚本  
****  
点击菜单DeviceManager--Device  Manager，连接测试设备(单台)，配置WIFI的ip和地址，点击链接操作：  
  
**5、运行&查看测试报告**  
选中一个脚本点击“运行”按钮， 或选择菜单“活动”->“运行”：  
  
当运行会话结束后，可以在“测试结果”窗口中查看测试结果。默认情况下，“测试结果”窗口在运行结束时自动打开：  


**六. 相关下载**

部署工具相关文档：[Operation Guide\_CN - APK Deployer.pdf](http://wiki.primeton.com/download/attachments/10848259/Operation+Guide_CN+-+APK+Deployer.pdf?version=1&modificationDate=1364219279000)

脚本开发相关文档：[Operation Guide\_CN - Script Designer.pdf](http://wiki.primeton.com/download/attachments/10848259/Operation+Guide_CN+-+Script+Designer.pdf?version=1&modificationDate=1364219279000)