"Selecting Mobile Application Automation Tools"

Presented by:

Pradeep Kumar Govindasamy
Cognizant Technology Solutions
Pradeep Govindasamy
Cognizant Technology Solutions

With more than twelve years of experience in the information technology industry, Pradeep Govindasamy currently holds the lead role for the Automation and Mobile Testing Center of Excellence at Cognizant. Pradeep started—and is now the research and development unit lead for—the automation and mobile practices which has more than 1500 experts servicing global customers. Pradeep has spoken on topics of automation, test data management, and tools at conferences worldwide including STARWEST, STAREAST 2011, Swiss Testing Day, HP Universe, and EuroStar.
Mobile Testing
Selecting Mobile Application Automation Tools

Thursday, October 06, 2011 3:00 PM

Agenda

Demystifying Mobile Platforms
Introduction

Understanding & Challenges
Mobile Automation Testing
Tool selection Criteria
Understanding tools

Determining the Right tool
Cognizant framework

Best in class frameworks
Part – 1
Demystifying the Mobile World

Mobile Platform

<table>
<thead>
<tr>
<th>Platform</th>
<th>Current Version</th>
<th>Source</th>
<th>Multi-tasking</th>
<th>Web Browser</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone OS</td>
<td>4.3</td>
<td>Closed</td>
<td>Restricted</td>
<td>Safari</td>
<td>Sync/Patch</td>
</tr>
<tr>
<td>Android</td>
<td>3</td>
<td>Open Source</td>
<td>Multi-tasking</td>
<td>Chrome/WebKit</td>
<td>OTA</td>
</tr>
<tr>
<td>Windows Mobile</td>
<td>7</td>
<td>Open Source</td>
<td>Multi-touch</td>
<td>I.E/Trident</td>
<td>Sync/Patch</td>
</tr>
<tr>
<td>Symbian OS</td>
<td>Symbian^4</td>
<td>Open Source</td>
<td>Multi-touch</td>
<td>Mozilla Browser</td>
<td>Sync/Patch</td>
</tr>
<tr>
<td>Blackberry OS</td>
<td>6.0</td>
<td>Closed</td>
<td>Multi-tasking</td>
<td>Webkit Browser</td>
<td>Sync/Patch</td>
</tr>
</tbody>
</table>
World Wide OS Market Share Trend

Worldwide Smartphone operating system (OS) market share in 2009-2015, according to Gartner

<table>
<thead>
<tr>
<th>Operating Systems</th>
<th>2009 market share</th>
<th>2010 market share</th>
<th>2011 market share</th>
<th>2015 market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>3.9%</td>
<td>22.7%</td>
<td>38.5%</td>
<td>48.8%</td>
</tr>
<tr>
<td>BlackBerry</td>
<td>19.9%</td>
<td>16.0%</td>
<td>13.4%</td>
<td>11.1%</td>
</tr>
<tr>
<td>iOS</td>
<td>14.4%</td>
<td>15.7%</td>
<td>19.4%</td>
<td>17.2%</td>
</tr>
<tr>
<td>Symbian</td>
<td>46.9%</td>
<td>37.6%</td>
<td>19.2%</td>
<td>0.1%</td>
</tr>
<tr>
<td>Windows Phone/Mobile</td>
<td>8.7%</td>
<td>4.2%</td>
<td>5.6%</td>
<td>19.5%</td>
</tr>
<tr>
<td>Others</td>
<td>6.1%</td>
<td>3.8%</td>
<td>3.9%</td>
<td>3.3%</td>
</tr>
<tr>
<td>Total smartphones sold</td>
<td>172 million</td>
<td>297 million</td>
<td>468 million</td>
<td>631 million</td>
</tr>
</tbody>
</table>

What is Mobile App

Mobile Applications are a rapidly developing segment of the global Mobile Market. They consist of software that runs on a mobile device and performs certain tasks before the user of the Mobile Phone. They can be downloaded physically through USB / WiFi from a desktop or can be downloaded by a web server over internet.
Types of Mobile Applications

Native Applications
- Applications that reside on the device and are accessible at any point of time
- May require network connectivity
- Messaging, Telephony, Multimedia, PIM, etc...

Hybrid Applications
- Third party applications that are installed in the device
- Requires network connectivity
- Google Chat / Messenger, shopping application on iPhone

Web Based Applications
- Accessible through mobile browsers
- Interfaces with external systems/servers
- Wireless Application Protocol (WAP) or Hyper Text Transfer Protocol (HTTP)
- Mobile Banking, Reservation system

Deciding the type
Depending on the requirements. Some apps are a better fit with web technologies than others. Knowing the pros and cons of each approach will help you make the right decision about which path is appropriate for your situation.

Stop! – Quality check for Mobile Application

Increase of Mobile Applications
Hurry-to-market pressures
Being the first increases greater usage of Applications

- Failure and Security losses destroys the brand reputation
- Test coverage
- Unhappy Customer

Remedy
End to End testing is required for thorough validation of Mobile Applications. Mobile Automation testing is vital to ensure a high application quality and faster time to market in a highly competitive market

Mobile Automation Testing becomes Paramount
Part 2- Understanding the Components of Mobile Automation Testing

Mobile Testing?

- Hardware/Device Testing
  - Factory testing
- Software Testing
  - Certificate Testing
  - Application testing
  - Content testing
Mobile Testing?

Mobile Testing intends to enable greater mobile experience
Mobile Application testing tests application on various parameters to enable defect free mobile applications
Automating Mobile Application Testing can deliver greater benefits in-terms of cost and quality

Hardware or Device Testing
- Factory Testing
- Certificate Testing

Software Testing
- Application Testing
- Content Testing

Mobile Testing

Mobile App Testing Challenges

DEVICE DIVERSITY
- Multiple Platforms
- Multiple Browsers
- Rendering differences
- Mobile devices have different application runtimes.

HARDWARE CHALLENGES
- Limitations in processing speed
- Limitations of Memory size of mobile
- Differences in Communication Protocols of devices WAP/ HTTP.

NETWORK CHALLENGES
- Multiple type of Networks(GSM / GPRS / Wi-Fi / Wi-Max etc)
- Unpredictable time taken for data transfer
- Different speed of connectivity across geographies
- Multiple Network Operators with customized Network features
Mobile Application Testing Eco System

Mobile Application Testing

Automation Tools
Emulators
Virtual Devices
Test Tool
Simulators
Real Devices

Mobile Automation Testing?

Why?
Usual
Rare

Quicker
Highly reliable

Cheaper
Robust

Net - Net

Though there are challenges the benefits of Automation supersede the demerits

CHALLENGES

Innumerable variations
OS/Platforms
Hardware devices/ OEM

Limited Resources
Processing speed
Small memory

Numerous Connectivity modes
Wi-Fi, Wi-Max,
Cellular networks

Varying features
Accelerometer
Radio transmitter
A Typical Test Automation setup for mobile

- Real Device
- Emulator
- Automation tool

Connectivity
- Bluetooth
- Wi-Fi
- USB
- Remote

Typical Mobile Automation

- Mobile applications under Test (AUT)
- App 1, App 2, App N

- Test Data

- Automation Engine starts and automates test cases on the application working on device and/or emulator

- Test Management Tool
  - Scheduler
  - Defects

- Access Interface

- Reporting Defects

Schedule and track test execution
Mobile Automation Approach

Understand Requirements and Budgets

Type of test automation

Traditional: e.g. Regression etc
- Classify Application as Native, Hybrid, Web Based
- Determine the test tool for maximum automation coverage
- Perform manual testing and create Scripts for Automation
- Execute Automation test cycles

Non Traditional: e.g. Compatibility, Location based etc
- Determine feasibility of Automation
- Scan markets to identify tools for Automation
- Evaluate CBA of patch/tool creation for Automation
- Execute manual testing and create test scripts
- Execute Automation test cycles

Perform manual testing and create Scripts for Automation

Types of Test Automation

Classify Application as Native, Hybrid, Web Based

Determine the test tool for maximum automation coverage

Manual execution, creation and execution of Automation test cycles
Candidates for Mobile Automation

Typical Candidates for Mobile Automation

<table>
<thead>
<tr>
<th>Business</th>
<th>Testing Types</th>
<th>Testing Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Complex business logic</td>
<td>• Smoke and build acceptance testing</td>
<td>• Less Dynamic UI components</td>
</tr>
<tr>
<td>• Lengthy business flow</td>
<td>• Regression testing</td>
<td>• Across platform testing</td>
</tr>
<tr>
<td>• Data Driven Test scenarios</td>
<td>• Synchronization testing</td>
<td>• Testing with different carriers across various location</td>
</tr>
<tr>
<td>• Frequently changing Application under test</td>
<td>• Compatibility testing</td>
<td>• User’s interaction with the application</td>
</tr>
<tr>
<td></td>
<td>• GUI Testing</td>
<td>• Test steps spanning across multiple test cases</td>
</tr>
</tbody>
</table>

Part 3: Tools and Best Practices
Part 3.1: Mobile Web Automation
Mobile Web Automation

Web Pages render differently on various mobile phone making Automation of Mobile Testing of Web Page Key to its success

Automation Testing Approach

- User Agent Based Automation Testing
- Emulator / Device based Automation Testing
- Remote based Automation Testing

Mobile Web Automation Steps

1. Gather Requirement for Testing
2. Classify the testing types
   - Compliance
   - End to End testing
3. Compliance Testing
   - Automated Testing tool to validate the content at every step of development
4. Decide on Automation tool
   - Browser Simulation tool
   - Emulator / Device Automation tools
   - Cloud based automation tools
5. Conduct Final verification testing on multiple Devices
6. Conduct testing on emulators

User Agent Based Automation Testing

A User agent is a utility which comes as an add-ons with any desktop browsers that makes the browser request and display mobile web pages. Test Automation tools like QTP or Selenium can be leverage for validating Mobile Web page displayed on Web Browser

- **Pros**
  - Exact look and feel of the web application is not realized
  - No additional software required to be installed
  - Easy to incorporate in-house frameworks for automation.
  - Does not need mobile devices to run the tests
- **Cons**
  - Issues that occur due to device browsers may not get captured
  - Automation is possible with existing automation tools like QTP, Selenium and so on

<table>
<thead>
<tr>
<th>Automation Tools</th>
<th>Independent Addons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fidler</td>
<td>Selenium</td>
</tr>
<tr>
<td>BaydenUAPick</td>
<td>QTP</td>
</tr>
<tr>
<td>Plugins in Firefox</td>
<td>RFT</td>
</tr>
<tr>
<td>UA Switcher</td>
<td>Silk Test</td>
</tr>
<tr>
<td>Custom Addon's</td>
<td></td>
</tr>
</tbody>
</table>
The functional validation of any mobile application would normally be done on emulators/mobile devices. While it is highly recommended to test the end to end testing on mobile devices, emulators can be leveraged for initial sanity checks and also to a certain extent for functional validation. There are various automation tools available for validating the mobile application on both devices and emulators.

Pros:
- It is possible to do an end to end testing of the application.
- Test coverage can be highly increased.
- Highly reliable
- It is possible to automate lot of live scenarios.
- A good mix of emulators and devices is recommended if testing is exhaustive.

Cons:
- The initial environment set up needs to be done and it involves cost.
- Device specific issues cannot be captured using emulator based automation.

Tools:
- Selenium
- Robot Selenium

Supported Platforms:
- iOS and Android

Part 3.2: Mobile Native / Hybrid Application Automation
Native/Hybrid Application Automation

Native and hybrid Applications are more rich applications and highly interactive applications. Automation Testing Approach and Process are different for Native Application.

Automation Testing Approach
- Emulator / Device based Automation Testing
- Remote based Automation Testing

Native App Automation Steps
- Gather Requirement for Testing
- Classify the testing types
- Perform POC for Application Automation
- Conduct testing on emulators
- Conduct Final verification testing on multiple Devices
- Decide on Automation tool
  - Emulator / Device Automation tools
  - Cloud based automation tools

Emulator/ Device Based Automation Testing Approach

Native Application Automation would demand possible download from App server and installation of Mobile Applications. The Device Should have Automation agent Running in it.

Pros
- It is possible to do end to end testing of the Application
- Test coverage to a very high percentage
- Highly reliable
- It is possible to automate – many automation tools are available.
- A good mix of emulators and devices is recommended if testing is exhaustive

Cons
- The initial environment set up needs to be done and it involves cost.
- Testing under various carrier networks is cumbersome.

Automation Tools
- Robotium
- UI Automation
- Test Quest
- Jamo Solutions
- See Test
Cloud Based Agent’s Automation Testing Approach

Cloud based testing offers testing on real devices hosted in the cloud. It facilitates in validating a mobile application if it has to be done on a various mobile devices on a live carrier network. They also support automation and test management.

Pros:
- Access to large pool of devices
- Facilitates testing on live devices under live network
- Credible testing results
- Possible to test all modes of connectivity
- Highly increased test coverage
- Incorporates automation.

Cons:
- Highly dependent on network performance bottlenecks creep up
- Not possible to trouble shoot issues on devices and on the test environment

Client Side IDE / Test Tool

Test Engine  Test Data Manager  Report Manager

Automation Tools

Device Anywhere  Perfecto Mobile

Part 3.3: Tool Selection and Best Practices
Tool Selection Criteria

- **Scope based decisions**
  - Platforms support
  - Supported Versions of platforms
  - Types of automation supported.
  - Device variants within a platforms (form factors)

- **Feature based decisions**
  - Working on both simulator/device
  - Integration to test management
  - Ease of use and scripting support

- **Infrastructural decisions**
  - Stability
  - Dependency on mac/desktops
  - Cloud based/stand alone

Best practices for Selecting the ideal Mobile Automation Tool

1. Always perform tool feasibility since mobile technologies and platforms are varied
2. Select tools that support both platform simulators and device as one could mix and match to optimizing runs on devices.
3. Also look for automation in non functional areas like interruption, hardware scenarios like battery state changes etc
4. Always optimize on the platform support, in some cases there could be a need for one or more tools to perform automation
5. Look for multiple devices support and versions support
6. Look for utility and reusable functions that would add value to automation
7. Always make sure the ROIs per platform and overall time
8. Look for avenues for early automation since most of mobile projects are also agile
9. Integrated execution with test management tool is going to be important for tool success
10. Look for data driven automation support as iterations in execution is going to increase coverage and ROI
11. Also look for failover and error handling mechanisms in the automation tools, since there is a lot of external interactions that could cause unexpected errors
Candidates for Automation

1. Identify regression test cases
2. Classify test cases according to Automation tool ability
3. Identify complex applications business logic
4. Identify test cases that are complex in creation and maintenance
5. Look for test cases that are redundant in nature
6. Identify test case flow that can be accommodated across multiple mobile platforms.
7. Time consuming manual test cases that can be effective when automated
8. Test cases that can run unattended with little / no manual intervention
9. Identify test cases based on the automation technique of mobile automation tools

Best Practices for Automation

- Organize the test assets i.e. test objects, test data, procedures / functions and test scripts.
- Use Re-Usable components within various scripts.
- Identify Re-usable components within organized objects, procedures and functions.
- Use Re-usable components across platforms.
- Data drive the test scripts to reduce script creation and execution time.
### Best Practices for Automation

<table>
<thead>
<tr>
<th>Best Practices</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform POC on critical platforms to narrow down to a tool/tool set</td>
<td>One tool may not solve cross platform automation goals be open for more than one tool</td>
</tr>
<tr>
<td>One tool may not solve cross platform automation goals be open for more than one tool</td>
<td>While evaluating tools for Automation, also evaluate them for integration with Test management tools</td>
</tr>
<tr>
<td>While evaluating tools for Automation, also evaluate them for integration with Test management tools</td>
<td>Identify reusable scenarios across platforms and group them. This facilitates creation of appropriate framework to save time</td>
</tr>
<tr>
<td>Identify reusable scenarios across platforms and group them. This facilitates creation of appropriate framework to save time</td>
<td>For Native Apps, choose critical test cases that are going to be run across releases as candidates for automation</td>
</tr>
<tr>
<td>For Native Apps, choose critical test cases that are going to be run across releases as candidates for automation</td>
<td>Always keep in mind different form factors while determining test coverage, especially for Mobile Web.</td>
</tr>
<tr>
<td>Always keep in mind different form factors while determining test coverage, especially for Mobile Web.</td>
<td>Factor the different connectivity aspects for automation execution runs</td>
</tr>
</tbody>
</table>

---

**Part 4: Best in Class Frameworks**

©2011, Cognizant
One of the more useful (from a developer standpoint) features coming in iOS 4 is the UI Automation tool. This enhances to run an automated set of tests against an application, and test to see if they had the expected results.

**Script can be easily executed on real device and simulator**

**Ease of Script using JavaScript**

**Ability to perform UI level automation on iPhone applications**

---

### Key Benefits

- Collection of JavaScript library functions
- Eases the test script creation
- Easy to install and use
- Improves reusability

---

### iPhone Automation Framework Approach

**Architecture**

- **Tune up**
  - UI Automation Assertion
  - Pre-defined Reusable Functions

**Features**

- **Instrument**
  - UI Automation

**Pre-defined Reusable Functions**

---

### Android Framework Approach

One of the more useful (from a developer standpoint) features coming is the UI Automation tool. This enhances to run an automated set of tests against an application, and test to see if they had the expected results.

**Robotium Implement**

- Android Instrumentation Testing

**Features**

- Robotium implements Activity Instrumentation of Android Testing Framework in a simple manner similar to Selenium
- The test script is a java file which implements robotium’s class ‘solo’
- The test scripts can be run on actual device or emulator using Android Junit Framework in eclipse platforms
- It can also be run using the Instrumentation test runner through ‘adb’

**Key Benefits**

- Offers additional functions which are not present in Activity testing
- It is open source and easy to develop scripts.
- Improves test productivity, and can be executed on device and Emulators.
# Part 4.1: Industry Tools

## Mobile Test Tools in Market

<table>
<thead>
<tr>
<th>Tools/Parameters</th>
<th>Test Quest Countdown</th>
<th>Device Anywhere Pro</th>
<th>Squish</th>
<th>Jamo Solutions</th>
<th>Perfecto Mobile</th>
<th>Robotium</th>
<th>Eggplant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support of devices/platforms</td>
<td>Android, BlackBerry, Brew MP, Symbian and Windows Phone</td>
<td>Android, BlackBerry, Brew MP, Symbian, Windows Phone and Apple iPhone and iPad.</td>
<td>Squish is a general GUI testing tool</td>
<td>Android, Windows Mobile, Windows CE.</td>
<td>iPhone, IOS, BlackBerry, Android, Windows Mobile and Symbian.</td>
<td>Android, iPhone, Symbian, Windows Mobile, Windows CE</td>
<td>Android, iPhone, Symbian, Windows Mobile, BlackBerry</td>
</tr>
<tr>
<td>Licensing</td>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
<td>OpenSource</td>
<td>Cost</td>
</tr>
<tr>
<td>Ease Of Usage</td>
<td>Easy to use</td>
<td>Easy to use</td>
<td>Needs programming</td>
<td>Easy to use.</td>
<td>Easy to use.</td>
<td>Needs programming</td>
<td>Easy to use.</td>
</tr>
<tr>
<td>Scripting</td>
<td>No Scripting</td>
<td>Scripting Possible</td>
<td>Uses JavaScript, Python, Perl or TCL.</td>
<td>Uses VB script.</td>
<td>Uses no programming language for scripts, script is very simple and can be easily understood</td>
<td>Uses java.</td>
<td>Uses 'Sensetalk'</td>
</tr>
<tr>
<td>Test Data Parameterization</td>
<td>Not available</td>
<td>Possible</td>
<td>Available</td>
<td>Available</td>
<td>Not Possible</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Remote Connectivity</td>
<td>Possible</td>
<td>Possible</td>
<td>Possible</td>
<td>Possible</td>
<td>Not Possible</td>
<td>Not Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Integration with test management</td>
<td>In-built test management tool test manager</td>
<td>Quality Centre</td>
<td>Quality Centre, Cruise control</td>
<td>Quality Centre and any other test management tool supported by QTP</td>
<td>Possible</td>
<td>Not Possible</td>
<td>Not Possible</td>
</tr>
</tbody>
</table>
Device Anywhere


- Device Anywhere provides testers access to more than 3,500 Mobile handsets via the Internet for remote testing.
- Virtual Test Lab with real devices hosted
- Available for usage 24x7x365 in live network
- Shared across distributed teams
- Real-time access to Mobile networks
- Screen-by-screen phone activity capture
- Automation testing feasibility
- Test Case Manager tool helps QA teams to define test plans and create customized test reports

Device Anywhere Advantages
- User Friendly
- User gets access to many devices
- Users can share a device
- We can test real mobiles

Device Anywhere Disadvantages:
- Cost involved

---

Perfecto Mobile

Perfecto Mobile is provider of Remote Access and Automated testing solutions for Mobile devices.

- Currently it support the following platforms: iPhone, Android, Windows Mobile, Symbian, Blackberry, Droid
- The Perfecto Mobile Handset Cloud service enables you to test your Mobile applications, websites or services on a multitude of REAL handsets.
- The service enables you to access real Mobile devices via the web and control them as if you were holding them in your hands.
- Test Automation enables you to extend your testing coverage, shorten your testing cycles and replace repetitive and time consuming manual work.
Part 4.2: Cognizant’s Mobile Automation Testing Frameworks

Benefits

1. Reusability of scripts is maintained.
2. Speed-up Test execution which reduces time.
3. Higher Reproducibility for errors through automated testing.
4. Ease to maintain the portability of scripts across device.
5. Less manual interception, hence less error prone which gives higher ROI.
Typical Mobile Application Testing Framework

**Application layers**
- Backend
- Middleware
- Device UI

**Types of Testing**
- Integration testing
- SOA Testing
- Device Mgmt testing
- Installation Testing
- Functional Testing
- Security testing
- Compatibility testing
- Device profiling
- Authentication testing
- UI Testing
- Interruption testing

**Tools to accelerate testing process**
- Fully Automatable
- Partially Automatable
- Not Automatable

**Framework Integrated Testing for Mobile Web APPs**

**Testing of Mobile Web Applications**
- The browser simulation tools like BaydenUA Pick, Fiddler, and User Agent Switcher etc can be used to simulate the mobile browser experience on IE, Mozilla Firefox, to render web sites as it would appear on mobile devices.
- The Selenium IDE can be used to record scripts and export the code to java, ruby, python or c#.
- The exported code can now be used with Selenium RC, where the scripts can be modified according to the needs and can be run against any browser.
- The approach makes use of the in-house CRAFT framework where the Business flows are driven through excel sheets, using keywords.

**Automation approach for testing mobile web apps using Selenium**

**Key Benefits**
- This is a very effective method for testing mobile web apps, since it doesn’t use any simulators or real devices.
- It just uses the desktop browser and the open source tool Selenium, using which we can easily test the functionality of web applications.
- Makes efficient use of the home-grown CRAFT framework.
Cognizant Mobile Testing Tools

CSMartPro
- CSMartPro is an Cross Platform Automation tool for Graphical User Interface based functional testing.

M-Jammer
- Cognizant’s Mobile Interrupt Simulating Tool, can be used for testing the AUT when the mobile application is stressed with multiple interrupts.

M-TestLocale
- M-TestLocale overrides the GPS on User Location, thus enabling Location Based Service Applications to be tested in a Smart fashion.

M-WebChecker
- A desktop tool which would check the compatibility of web pages for different mobile devices in accordance with W3C standards.