**MOBILE APPLICATION TESTING**

***Abstract: Mobile app testing is the process of testing the applications developed for hand held devices for usability, functionality and performance issues. Mobile app testing is a little different when compared to the testing done for desktop applications as many things need to be considered like the user interface , hardware , platform connectivity issues and many more.***

**1. Introduction**

Mobile app testing may be time consuming and expensive, but it is critical and ensures that the customers do not find any difficulty using the application. If the app developer fails at doing a good job at testing, it may lead the customers to do the testing which might cause them trouble as they do not have the same tools as the developer does.

A comprehensive mobile app testing strategy includes device and network infrastructure, optimized selection of the target devices and a combination of different tools to cover both the functional and non-functional aspects of the application is important to ensure that the application gets launched on time.

**2. Mobile App Testing**

Mobile apps either come pre installed or can be installed from the application providers. In the last decade or two has seen immense increase in the number of mobile devices and so there has been an increase in the number of applications as well.

*Variety of mobile devices***:** Hand devices come with many variations and differences like different screen sizes , UI and different input methods and hardware

*Different OS/Platforms:* There are different OS in the market now like Android, IOS, BREW, Windows7.5, Mango. Every platform has its own challenges and ways to deal with.

*Different network providers:* There are many network providers present today. Few are CDMA, GSM and few are FOMA and TD-SCDMA. Every network provider has a unique infrastructure and therefore may limit the flow of information in their own way

*Scripting:* As devices vary so much in their hardware and UI, therefore, the code for the application must be written differently and may be a major problem if an app is being created for multiple platforms.

**3. Types of Mobile App Testing**

*Usability Testing***:** This is to see if the app is easy to use and if the user interface is working according to the customers’ expectations or not.

*Services testing*: Testing for the applications’ behaviour both online and offline.

*Compatibility testing***:** Testing to see how the application works on different platforms, how it looks on different screens and resolutions and how it responds to different user interfaces.

*Interface testing***:** Testing the various components that the user interacts with in the app like menus, buttons, navigation tools and many more.

*Performance testing***:** This is especially done for the apps that run online. Here testing is done by changing the connections from 2G to 3G and other networks as well, and the performance is evaluated in each network. Also , it sees how fast the app responds for both online and offline applications and accordingly the apps can be optimized.

*Low Level Resource testing***:** This tests for the memory usage of the apps and the auto deletion of files and so on.

*Operational testing***:** This testing is done to see how the app works if the battery goes down and sees how the data is recovered if the device gets switched off while using the application. Thus it looks for the backup plan for the app.

*Security testing***:** Checks if the data set in the app is secure or not.

*Interrupt testing*: This sees how the apps reverts after an interrupt like phone calls, messages ,etc.

*Installation testing*: Many apps come predefined while some have to be downloaded from the provider, This testing checks if the installation process has any interrupts or problems in it.

*Certification testing:* In order to get the certificate of compliance, the app has to be tested against a set of guidelines. This testing is done to see how well the app abides by the required guidelines.

**4. Mobile App Testing Strategy**

Even before testing, we need to have a testing strategy in order to meet the customers’ requirement, specifications and to avoid negative feedback. Testing is an important step for quality assurance.



Fig 4.1 Mobile app testing strategy plan

**4.1 Selecting target devices**

The target devices need to be selected in a way that they represent maximum of the devices present in the current market. Which means that the test bed must contain most of the devices such that testing on those devices is equivalent to testing the app on all the devices in the market so that we understand the various problems that might turn up when the app is run and launched on the device.

**4.2 Simulators vs Physical Devices**

Testing the app on system simulators is useful in the early stages of the development, as it allows the test team to get used to the various features of the app.



Fig 4.2: Conditions for testing on devices/simulators and different network connectivity options.

**4.3 Device Model Selection:**

Factors to be considered while selecting a device for testing:

*OS Version*: The device being selected must have the version which is available on most population of devices.

*Form Factor*: devices behave differently on smart-phones and tablets. So if the application is being developed for both tablets and mobiles, we need to test them separately and analyze the behaviour.

*Display Density*: The screen resolution and size effect the application user interface, therefore, the device used fir testing must include a mix of different display densities.

When it comes to IOS devices, selecting a test devices is easy because of lesser models available, where for Android we need to see the most popular device in the market at that particular time and then adjust the features of the app accordingly.

*Connectivity options:* In order to test the apps that require internet to work , it’s better to set up a Wifi and then test the app as it is easy and inexpensive when compared to the cellular devices that are very unpredictable when it comes to providing internet. On location testing of the application must also be considered and done for a few specific applications.

*Manual vs Automated Testing:* Automated testing provides us to do repeated testing over the same code or app and gives the results and verifies them to the expected end result. It is very helpful when it comes to regression testing, but the main drawback when it comes to automated testing is that it requires very high initial investment and sometimes the return on investment is quite poor. The ROI here depends on the test cases that are selected for testing the app, so in order to get high ROI , we need to properly select the test/use cases.

To get the maximum output from the automation, all the automatable test cases of the release need to be automated before the next release.

**5. Mobile App Testing Cycle**

Mobile apps are being upgraded frequently for one or more of the following reasons:

1. The mobile users need recent updates and bug fixes
2. The upgrades are done to take advantage of the newer versions of operating systems and platforms.

Every time an application is upgraded an additional testing cycle is required to assure that the application is compatible with the newer platform and operating system .

Important things that need to be considered during the testing cycle are:

*Test Environment:* This includes selecting devices for testing. We need to make plans for DUTs( Device under test), that is, we e need to consider the fact that there will be frequent OS updates and that should not alter the behaviour of the app.

*Application under testing*: Depending on the needs of the organisation, they need to make a decision on whether to create web app, native or hybrid application. Any changes must go through Quality Assurance and test cases must be created for each approach.

*Automation:* Most of the tools used for testing are OS specific and cannot be reused for testing the app for multiple systems or OSs and many tools favour web based applications over others. Therefore proper tools must be selected in advance for testing the applications.

Testing plan

Testing design

Understand customer requirements

Testing report

Testing execution

Fig 6.1 Mobile app testing cycle

**6. Sample Mobile App Testing using Python.**

**1. Overview : Tools in device simulators**

Outline what the tools are in Android and BlackBerry simulators that will help users interact with the simulators. E.g. For BlackBerry, there is fledgecontroller. For Android, there is adb.

**2. Introduce bblib.py and androidlib.py** Show how we can abstract these tools using Python and provide common interaction interface with both devices. Demo examples of commonly used commands in both Python libraries.

def enter(string=None):

""" If string is None, simulate Enter key in simulator.

def backspaces(count=1):

""" Press backspaces specified by count."""

def touch(xcoord, ycoord):

""" Touch screen at (xcoord, ycoord)."""

def thumbwheel(direction='up', count=1):

def trackball(direction='left', count=1):

**3. Writing device independent tests**

#1. Use python name typing to decide the device to use

def getDevice(self):

mobileDevice = testenv.getDevice()

if mobileDevice == 'android':

deviceClass = android.AndroidImpl()

else:

deviceClass = bb.BlackBerryImpl()

log.debug('Device is ' + mobileDevice)

return deviceClass

#2. Usecase implementation

# On BlackBerry

def login(username, password):

enter(username)

thumbwheel('down',1)

enter(password)

thumbwheel('down',1)

enter()

#3. Testcases (device independent) for Login

class LoginBAT(unittest.TestCase):

device = testlib.testenv.getDevice()

def testIncorrectUsername():

self.device.login(incorrectUser, correctPin)

self.assertTrue(imagelib.compare(self.device,\

self.\_testMethodName, '100%x40%+0+35%'))

def testSuccessfulLogin():

self.device.login(correctUser,correctPin)

self.assertTrue(imagelib.compare(self.device,\

self.\_testMethodName, '100%x40%+0+35%'))

Test assertion use imagelib.py - takes screenshot of current

image and compare with expected image with crop settings and

tolerance level using ImageMagick.

def compare(device, image, cropSettings=None, tolerance=500):

#4. Run test, showing assertion results.

**4. Summary : Test Framework**

Walkthrough test framework architecture (diagram will be included). Lowest level is test utilities - imagelib.py, logger.py testlib.py (exceptions handling etc.).

Next up is device simulator libraries - bblib.py, androidlib.py containing device controls.

Above this is application use case implementation, followed by end-to-end device independent tests.

Finally, at the top layer is test runner and reporting framework.

**7. Challenges in Mobile App Testing**

The current market is about the mobile devices and all, experts even predict that in the near time these mobile devices might take over desktop devices and PCs, but with every new thing comes its own set of challenges which become very difficult to tackle. Following are the few challenges faced:

***7.1 Mobile App Testing is different***

Mobile app testing is quite different from Desktop web applications. In case of web applications, they need to be tested on a single platform and the testing is done, but when it comes to mobile applications, there is one specific dormant platform to test, therefore we need up creating multiple versions of the app for different operating systems and also different versions of the same operating system. Thus, it is more complex and time consuming

For desktop systems, there is standardized hardware implementation, but for mobile devices the hardware changes from system to system and sometimes it changes for two different releases of the same device causing one of the main problems when it comes to mob app testing. The android gallery itself has 60 different device configurations with different screens resolutions and many more.



Fig 7.1.1 Android OS Version Fragmentation

All these different factors make mobile application testing complexity, time consuming and expensive.

***7.2 Aspects of Testing Mobile App***

In order to properly understand the complexity of testing mobile applications, we need to understand the various aspects of it. Some of these aspects may be general to any software testing while others are more specific to mobile app testing.

The kind of testing that is needed for an application depends on many factors like:

* The target customers
* The kind of application being developed
* Distribution channel(the store that’ll be selling the app)

**8 . Conclusion**

Even though mobile app testing has its own set of challenges, the right choice of target devices, connectivity options and tools that maximize automation ensure efficient , effective and cost free testing.

We can design and develop better quality applications by testing all the aspects of the app. These trends show how different mobile apps are when compared to the web and desktop applications and prove that different testing methodologies and strategies need to be implemented, which takes the current scenario and the mobile world into consideration and adjustments need to be made to the conventional testing methods and strategies.

**9. References**

1. <http://www.optimusinfo.com/cloud-based-mobile-application-testing-tools/>

2. <https://us.pycon.org/2011/schedule/presentations/40/>

3. <http://www.fromdev.com/2013/09/webdriver-selenium-code-snippets.html>

4. Mobile Application Testing Matrix and Challenges , Bakhtiar M. Amen, Sardasht M. Mahmood and Joan Lu.

5. Mobile Application Testing and Challenges by Ravi Ramchandra Nimbalkar , International Journal of Science and Research , Volume 2 Issue 7, July 2013.

6.http://www.tcs.com/SiteCollectionDocuments/White%20Papers/Mobility\_Whitepaper\_Mobile-Application-Testing\_1012-1.pdf