Mobile Testing

Vu Lam

T6/2022

What is Mobile Application Testing?

- Mobile Application Testing is a process by which application software developed for handheld devices is tested for its functionality, consistency and usability.

Difference between **mobile testing** and **mobile application testing**

- Mobile Testing or Mobile Device Testing:
  - Mobile Testing is testing of Mobile Handsets or devices.
  - Testing all the core like SMS ,Voice calls, connectivity (Bluetooth), Battery(Charging),Signal receiving, Network are working correctly.
  - Testing is conducted on both hardware and software.
- Mobile Application Testing:
  - Mobile Application Testing is the testing of mobile applications which we are making as third party for the targeted mobile handset.
  - Some core feature of the mobile are tested just to see that your application has not created any side effects on your device functionality.

What are the Challenges in Mobile Application Testing?

- Mobile Application Testing is quite different thing which involves so many things like:
  - Wide varieties of mobile devices.
  - Different mobile network operators
  - Input Methods
  - Different mobile operating systems
  - Hardware Compatibility and so on….

Why focus on Mobile Testing?

- Phones getting truly smarter
- People getting more mobile
- Faster Networks (2G,3G,4G,5G,6G …)
- More apps, features and usages
- HTML5 making it very easier
- Firmware change regularly
- …

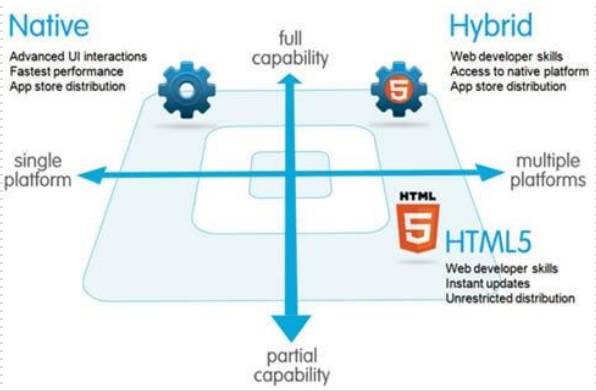
Why is quality critical?

- **Competition:** With so many applications vying for users' attention, high quality is essential for the success of an app
- **The first Impact:** The consumers should have a good experience each time they use the app starting from the very first time
- **Retaining Users:** If there are flaws that are detected by the consumers the chances are high that they never come back
- **Money Speaks:** Poor-quality applications not only hamper user adoption but also can cause revenue loss and irreparable brand damage

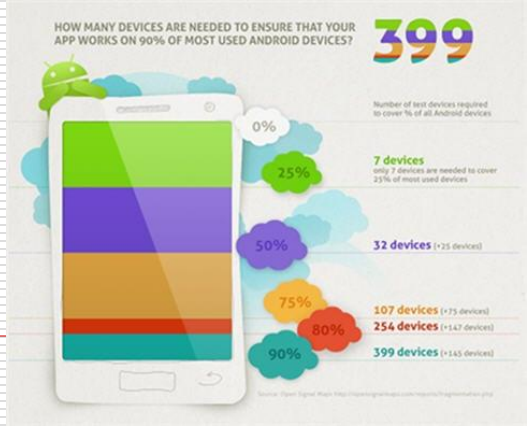
Challenges – mobile apps testing

- Testing mobile applications is more complex and time consuming compared to traditional desktop and web applications due to
  - Device diversity
  - Different OS and different versions
  - Wide variety of device form factors (phones & tablets of different screen sizes and resolution)
  - Network connectivity options
  - Unique environment conditions
  - Frequent changes in OS

Challenges



Tool	Paid/ Open Source	Native Apps	Web	Hybrid Apps	Android	IOS	Windows	Black-berry	Library/Tool
Robotium	Open Source	Y	-	Y	Y	-	-	-	Library
Sikuli	Open Source	Image Based	Image Based	Image Based	Y	Y	Y	Y	Both
Selenium WebDriver	Open Source	-	Y	-	Y	Y (but obsolete)	-	-	Library
NativeDriver	Open Source	Y	-	-	Y	Y	-	-	Library
Appium	Open Source	Y	-	Y	Y	Y	-	-	Tool
MonkeyTalk	Open Source	Y	Y	Y	Y	Y	-	-	Tool
SeeTest	Paid	Y	Y	Y	Y	Y	Y	Y	Tool
M-eux (JamoSolutions)	Paid	Y	-	Y	Y	Y	Y	Y	Tool
EggPlant	Paid	Image Based	Image Based	Image Based	Y	Y	Y	Y	Tool
mAutomate	Paid	Y	Y	Y	Y	Y	-	-	Web Based
Ranorex	Paid	Y	Y	Y	Y	Y	-	-	Tool



Types of Mobile Application Testing:

- Testing involves each and everything about the app like spellings to its functionality. So testing can be of many types like
  - Functional Testing
  - Laboratory Testing
  - Performance Testing
  - Security Testing Load & Performance Testing
  - Localization Testing
  - Usability Testing …

Functional Testing

- The Functional Testing of Mobile Application is a process of testing functionalities of mobile applications like user interactions as well as testing the transactions that users might perform.
- The main purpose of mobile application functional testing is to ensure the quality, meeting the specified expectations, reducing the risk or errors and customer satisfaction.

The various factors which are relevant in functional testing:

- Type of application based upon the business functionality usages (banking, gaming, social or business)
- Target audience type (consumer, enterprise, education)
- Distribution channel which is used to spread the application (e.g. Apple App Store, Google play, direct distribution)

Test scenarios in the functional testing

1. To validate whether all the required mandatory fields are working as required.
2. To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields.
3. To validate whether the application works as per as requirement whenever the application starts/stops.
4. To validate whether the application goes into minimized mode whenever there is an incoming phone call. In order to validate the same we need to use a second phone, to call the device.
5. To validate whether the phone is able to store, process and receive SMS whenever the app is running. In order to validate the same we need to use a second phone to send sms to the device which is being tested and where the application under test is currently running.
6. To validate that the device is able to perform required multitasking requirements whenever it is necessary to do so.
7. To validate that the application allows necessary social network options such as sharing, posting and navigation etc.
8. To validate that the application supports any payment gateway transaction such as Visa, Mastercard, Paypal etc as required by the application.
9. To validate that the page scrolling scenarios are being enabled in the application as necessary.

Test scenarios in the functional testing (cont.)

10. To validate that the navigation between relevant modules in the application are as per the requirement.
11. To validate that the truncation errors are absolutely to an affordable limit.
12. To validate that the user receives an appropriate error message like "Network error. Please try after some time" whenever there is any network error.
13. To validate that the installed application enables other applications to perform satisfactorily, and it does not eat into the memory of the other applications.
14. To validate that the application resumes at the last operation in case of a hard reboot or system crash.
15. To validate whether the installation of the application can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors.
16. To validate that the application performs auto start facility according to the requirements.
17. To validate whether the application performs according to the requirement in all versions of Mobile that is 2g, 3g and 4g.
18. To perform Regression Testing to uncover new software bugs in existing areas of a system after changes have been made to them. Also rerun previously performed tests to determine that the program behavior has not changed due to the changes.
19. To validate whether the application provides an available user guide for those who are not familiar to the app

Performance Testing

- This type of testing's fundamental objective is to ensure that the application performs acceptably under certain performance requirements such as
  - access by a huge number of users
  - the removal of a key infrastructure part like a database server.

Test scenarios for Performance Testing

1. To determine whether the application performs as per the requirement under different load conditions.
2. To determine whether the current network coverage is able to support the application at peak, average and minimum user levels.
3. To determine whether the existing client-server configuration setup provides the required optimum performance level.
4. To identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels.
5. To validate whether the response time of the application is as per as the requirements.
6. To evaluate product and/or hardware to determine if it can handle projected load volumes.

Test scenarios for Performance Testing (cont.)

7. To evaluate whether the battery life can support the application to perform under projected load volumes.
8. To validate application performance when network is changed to WIFI from 2G/3G/4G/5G or vice versa.
9. To validate each of the required the CPU cycle is optimization
10. To validate that the battery consumption, memory leaks, resources like GPS, Camera performance is well within required guidelines.
11. To validate the application longevity whenever the user load is rigorous.
12. To validate the network performance while moving around with the device.
13. To validate the application performance when only intermittent phases of connectivity is required.

Security Testing

- The fundamental objective of security testing is to ensure that the application's data and networking security requirements are met as per guidelines.

Most crucial areas for checking the security of Mobile applications

1. To validate that the application is able to withstand any brute force attack which is an automated process of trial and error used to guess a person's username, password or credit-card number.
2. To validate whether an application is not permitting an attacker to access sensitive content or functionality without proper authentication.
3. To validate that the application has a strong password protection system and it does not permit an attacker to obtain, change or recover another user's password.
4. To validate that the application does not suffer from insufficient session expiration.
5. To identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities.
6. To prevent from SQL injection related attacks.
7. To identify and recover from any unmanaged code scenarios.
8. To ensure whether the certificates are validated, does the application implement Certificate Pinning or not.

Most crucial areas for checking the security of Mobile applications (cont.)

9. To protect the application and the network from the denial of service attacks.
10. To analyze the data storage and data validation requirements.
11. To enable the session management for preventing unauthorized users to access unsolicited information.
12. To check if any cryptography code is broken and ensure that it is repaired.
13. To validate whether the business logic implementation is secured and not vulnerable to any attack from outside.
14. To analyze file system interactions, determine any vulnerability and correct these problems.
15. To validate the protocol handlers for example trying to reconfigure the default landing page for the application using a malicious iframe.
16. To protect against malicious client side injections.

Most crucial areas for checking the security of Mobile applications (cont.)

17. To protect against malicious runtime injections.
18. To investigate file caching and prevent any malicious possibilities from the same.
19. To prevent from insecure data storage in the keyboard cache of the applications.
20. To investigate cookies and preventing any malicious deeds from the cookies.
21. To provide regular audits for data protection analysis.
22. Investigate custom created files and preventing any malicious deeds from the custom created files.
23. To prevent from buffer overflows and memory corruption cases.
24. To analyze different data streams and preventing any vulnerabilities from these.

Usability Testing

- The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult application with many features.
- The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used.
- Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

Test scenarios

1. To ensure that the buttons should have the required size and be suitable to big fingers.
2. To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.
3. To ensure that the icons are natural and consistent with the application.
4. To ensure that the buttons, which have the same function should also have the same color.
5. To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled.
6. To ensure that the keyboard input can be minimized in an appropriate manner.
7. To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.
8. To ensure that the contextual menus are not overloaded because it has to be used quickly.
9. To ensure that the text is kept simple and clear to be visible to the users.

Test scenarios (cont.)

10. To ensure that the short sentences and paragraphs are readable to the end users.
11. To ensure that the font size is big enough to be readable and not too big or too small.
12. To validate the application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance.
13. To validate that the closing of the application is performed from different states and verify if it re-opens in the same state.
14. To ensure that all strings are converted into appropriate languages whenever a language translation facility is available.
15. To ensure that the application items are always synchronized according to the user actions.
16. To ensure that the end user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application's proceedings

Compatibility Testing

- Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired.
- The following are the most prominent areas for compatibility testing.
  - To validate that the user Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible.
  - To ensure that the text is readable for all users for the application.
  - To ensure that the call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed.

Recoverability Testing

- Crash recovery and transaction interruptions
- Validation of the effective application recovery situation post unexpected interruption/crash scenarios.
- Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
- The validation of the process where the connection is suspended, the system needs to re-establish for recovering the data directly affected by the suspended connection.

Important Checklist

1. Installation testing (whether the application can be installed in a reasonable amount of time and with required criterion)
2. Uninstallation testing (whether the application can be uninstalled in a reasonable amount of time and with required criterion)
3. Network test cases (validation of whether the network is performing under required load or not, whether the network is able to support all the necessary applications during the testing procedures)
4. Check Unmapped keys
5. Check application splash screen
6. Continued keypad entry during interrupts and other times like network issues
7. Methods which deal with exiting the application
8. Charger effect while an application is running in the background
9. Low battery and high performance demand
10. Removal of battery while an application is being performed
11. Consumption of battery by application
12. Check Application side effects