Beta Mobile app Testing guidelines

Quality Assurance (QA) plays an important role in the mobile applications development life cycle, but many overlook the critical nature of this piece of the app development process.

To ensure the successful development of any app, QA must be involved in all stages of development, from creating the concept, analyzing requirements and wishes, creating test specifications, testing early versions of the app, releasing the finished product, to the post-development review process.

Preparatory phase :

This phase begins after app development has been commissioned and is complete before the first assembly of a functioning app begins.

Important activities during this phase include :

01. Identifying target devices
02. Introducing functional requirements
03. Developing test documentation
04. Preparing the test environment

Guidelines to determine target devices include :

01. Figuring out what devices the application will support (phones, tablets, other devices – players, navigation).
02. Determining the earliest version of relevant operating systems to be supported.
03. Identifying the most popular models for the target audience.
04. Selecting additional devices with different screen sizes than the most popular models.

Functional requirements :

The next step is introducing functional requirements.

It is important to define whether the app is browser-based or installable, whether it interacts with a Web site or database, whether it interacts with other apps or social networks, or if it is completely self-contained (does not interact with the Internet and cellular network).

Information collected at this stage should focus on important app functions.

Test Documentation :

Then it is time to develop test documentation.

Many mobile apps only need high-level documentation, as tasks within the app are typically done with just a few clicks and do not involve complex tasks. This simplicity means that it is not necessary to create detailed instructions for testing.

For non-mobile apps as well, much depends on optimally partitioning app functionality into blocks while creating test documentation. This allows connections between different units to be tracked accurately, and requires less time to verify various functions.

Types of Mobile App Testing

To address all the above technical aspects, the following types of testing are performed on Mobile applications.

Functionality Testing : The functional testing of Mobiles normally consists in the areas of testing user interactions as well as testing the transactions. The various factors which are relevant in functional testing are:

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

The most fundamental test scenarios in the functional testing can be considered as:

01. To validate whether all the required mandatory fields are working as required.
02. To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields.
03. To validate whether the application works as per as requirement whenever the application starts/stops.
04. 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.
05. 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.
06. To validate that the device is able to perform required multitasking requirements whenever it is necessary to do so.
07. To validate that the application allows necessary social network options such as sharing, posting and navigation etc.
08. To validate that the application supports any payment gateway transaction such as Visa, Mastercard, Paypal etc as required by the application.
09. To validate that the page scrolling scenarios are being enabled in the application as necessary.
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 or the removal of a key infrastructure part like a database server.

The general test scenarios for Performance Testing in a Mobile application are:

01. To determine whether the application performs as per the requirement under different load conditions.
02. To determine whether the current network coverage is able to support the application at peak, average and minimum user levels.
03. To determine whether the existing client-server configuration setup provides the required optimum performance level.
04. To identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels.
05. To validate whether the response time of the application is as per as the requirements.
06. To evaluate product and/or hardware to determine if it can handle projected load volumes.
07. To evaluate whether the battery life can support the application to perform under projected load volumes.
08. To validate application performance when network is changed to WIFI from 2G/3G or vice versa.
09. 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.

Compatibility testing : Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. Browsers, Operating Systems, or hardware.

The following are the most prominent areas for compatibility testing.

01. 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.

02. To ensure that the text is readable for all users for the application.

03. 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.

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.

01. To ensure that the buttons should have the required size and be suitable to big fingers.

02. To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.

03. To ensure that the icons are natural and consistent with the application.

04. To ensure that the buttons, which have the same function should also have the same color.

05. To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled.

06. To ensure that the keyboard input can be minimized in an appropriate manner.

07. To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.

08. To ensure that the contextual menus are not overloaded because it has to be

used quickly.

09. To ensure that the text is kept simple and clear to be visible to the users.

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.

Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

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.

The following are the most crucial areas for checking the security of Mobile applications.

01. 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.

02. To validate whether an application is not permitting an attacker to access sensitive content or functionality without proper authentication.

03. 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.

04. To validate that the application does not suffer from insufficient session expiration.
05. To identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities.
06. To prevent from SQL injection related attacks.
07. To identify and recover from any unmanaged code scenarios.
08. To ensure whether the certificates are validated, does the application implement Certificate Pinning or not.
09. 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 i-frame.
16. To protect against malicious client side injections.
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.

Recoverability Testing : Recovery Testing is the failure which is forced into an application to check how well the recover process is performed.

01. Cash recovery and transaction interruptions
02. Validation of the effective application recovery situation post unexpected interruption/crash scenarios.
03. Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
04. 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.

Other Important Checks :

01. Installation testing (whether the application can be installed in a reasonable amount of time and with required criterion)
02. Uninstallation testing (whether the application can be uninstalled in a reasonable amount of time and with required criterion)
03. 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)
04. Continued keypad entry during interrupts and other times like network issues
05. Methods which deal with exiting the application
06. Charger effect while an application is running in the background
07. Low battery and high performance demand
08. Removal of battery while an application is being performed
09. Consumption of battery by application
10. Check Application side effects