Luxury Daily®

The News and Intelligence You Need on Luxury

COLUMNS

Testing mobile apps: QA can make or break project's success

May 16, 2013

By Tatyana Mahlaeva

Sign up now
Luxury Daily

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

To ensure the success of an 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:

1. Identifying target devices

2. Introducing functional requirements

3. Developing test documentation

4. Preparing the test environment

Guidelines to determine target devices include:

1. Figuring out what devices the application will support (phones, tablets, other devices - players, navigation);

2. Determining the earliest version of relevant operating systems to be supported.

There are two options: a special restriction, which will be further reflected in the requirements when installing the app, or selecting a lower limit operating system by device popularity among the target audience.

For example, for iOS devices it should be the first iPad with iOS 5.1.1 (if the application has no performance problems on it, there should be no problems in later versions of the device); the iPhone 3GS only runs iOS 4.3.5, so it is the least-productive platform to develop for, but the device is still popular globally.

3. Identifying the most popular models for the target audience.

4. 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.

User feedback
If the app is an updated version of an existing AppStore or Google Play offering, it may be useful to analyze user reviews and feedback that has been posted on social networks and the app marketplace.

Problems are often found and documented in these environments by end users, and this information will help identify and focus on the app’s most important shortcomings.

When new app versions are introduced, users expect to find specific problems have been fixed.

If they are not, it often causes them to react with anger or frustration, especially if they and others have taken the time to document or report errors.

It can be useful to add these end-user concerns to the test checklist as a separate category of problems, and pay special attention to them during testing.

Test environment

Preparing a test environment typically requires the installation of required apps on the mobile device, as well as the installation and configuration of required apps on the QA engineer's computer.

This might include apps such as the iPhone Configuration Utility and Android SDK.

From here, the process becomes an iterative cycle. It starts after the integration of the first assembly and ends with the completion of product development. It includes testing multiple assemblies at regular intervals (once or twice a week, for example).

Development and testing are carried out simultaneously with each new build of the app.

The main idea of testing is for the QA engineers to put themselves in the shoes of the user, but with a more profound knowledge of the settings and the principles of a particular device and the features of the app being tested.

Here are the characteristics of different types of apps and operating platforms, along with areas that require special attention when it comes to testing for various mobile operating systems.

Apple iOS testing

1. Verify compliance with UI Guidelines from Apple

2. Apps must be backward-compatible with the OS (a new version of the OS is expected to maintain full functionality of the app)

3. When multitasking, all settings and current progress of the app must be preserved when the app is minimized or when the user switches between apps

4. You must be able to debug the app and collect logs via a USB cable connection without additional operations being performed on the device

5. There is no need to reboot the device while testing

Google's Android testing

1. The app must support running multiple apps in the background

2. For debugging via USB, you must first select "Enable USB Debugging" in the device settings, and then connect the device to the computer

3. When using the hardware "Back" button, the app should be minimized, not closed.

Otherwise, the device may require rebooting to run the app again. You can also use an app such as Task Killer as an alternative to rebooting.

Windows mobile testing
1. The app must support multiple apps running in the background

2. You must be able to close the app through the task manager

3. You must connect the device to a computer via USB to debug the app.

Check these, regardless of the device/OS

1. Input methods, devices and form factors (number pad, qwerty-keyboard, touch screen, side panel, external devices)

2. Different networks (WiFi, Edge, 3G, 4G, GSM / CDMA), including unstable networks

3. GPS functionality

4. Energy consumption

5. Accelerometer

6. Standards Compliance mobile development (AppleHIG, Android principles)

7. External interrupts (calls, SMS, stability in the event of a shortage of disk space, installation and restoration in the background)

8. Memory cards

9. Stability after a crash

10. Using data on the device (address book, calendar)

11. Access rights apps

12. Social networks and other third-party apps

13. Interface (animations, icons, portrait/landscape orientation, Retina (for iOS), the visibility of labels, size, ease of use)

14. Time (server time/phone time, time zones)

15. Redirections (Web → app, app → Web)

Control phase
The control phase is where you prepare the draft for release after product development ends.

This phase includes detailed and complete testing (during early iterative phases, a full test might not have been performed, or more regression tests may be required) to stabilize the app and uncover minor defects.

This also tests for loss of auxiliary modules (testing in the app can present opportunities to check subsidiary functions such as Log Collector and artificial recharge of game balance, which were added to facilitate or reduce testing time, but that should not appear in the final product)

The approach typically used for testing in this control phase is somewhat like using scaffolding when constructing a building.

We can add features and change settings for the sake of development and testing that will not be in the final version of the app.

For example, an app undergoing control phase testing might be set to provide endless lives or infinite ammunition for the tester.

Once the building is complete, the scaffolding is taken away – much like how settings and features used to test and develop the final version of the app are then taken away before it is made available to the public.

Acceptance testing
The final phase is acceptance testing. The focus here is to check if the app matches its acceptance criteria (which was defined at the start of the development cycle) or not.

Acceptance testing is performed based on a set of typical test cases and scenarios that are created based on the app requirements. It may specify "no major defects exist in the application," or "all application labels and functions fully correspond to the rules of the target language."

This phase of testing is also typically the most important, especially when further support of the app is not provided once it is made public.

Acceptance testing is used to ensure the quality of the app is perfect – or near perfect – and to ensure that the customer is happy with the results of the development process.

At this step, the customer decides if the app is excellent enough to accept it as final, and that decision is primarily based on the test results.

AFTER CAREFULLY tailoring these test phases to the app being developed and meticulously carrying them out, you are certain to end up with functional product.

Tatyana Mahlaeva is mobile applications QA manager for A1QA, an Austin, TX-based global software testing and quality assurance company. Reach her at t.mahlaeva@a1qa.com.

© Napean LLC. All rights reserved.

Luxury Daily is published each business day. Thank you for reading us. Your feedback is welcome.