

COLUMNS

A divide-and-conquer approach to mobile app testing

November 12, 2013



By **Tatyana Mahlaeva**

Timely, thoughtful, consistent testing is an important part of the quality assurance process. So what will help to ensure consistency and structure of the testing process? Well-written test documentation.

To create a strong application testing plan, you must pay attention to two important aspects. First, all application functions and device features must be tested. Second, it is desirable to expand coverage to previously detected defects and rare scenarios that lead to their reproduction.

The main point of this article is to highlight general functional mobile defects that occur in relation to application functions. Please note this article does not cover usability, security and other special types of testing.

App functions

First, let us divide all the functions of the application and device features into a few large groups. This division is not definitive. It just shows how it can be done.

Lifecycle issues

Audio/video

Graphic user interface

Network-related issues

Hardware/software specialties

Data-related issues

Let us start with lifecycle issues. This functionality includes installing, uninstalling, updating from a previous version, and the initial steps of opening an application launch and log-in processes. The most common defects found here include:

Blank or corrupted application icon or name

No registration function

Incorrect or missing application version number

Application data not saved during update process

User data remains on the device after removing the application

User data remains on the device after removing the application

When updating, a new version of the application is installed, but does not replace the previous version

Issues with updating from a previous version

The inability to log out permanently (critical for public devices)

No synchronization in accounts (when both desktop and Web versions exist)

Audio/video

Next is audio/video. There are typically not many defects in this group, but it is important not to forget about them, because defects here can confuse users to a degree that they leave your application forever.

When the application has its own sound, the music player application is not paused

The inability to turn off sounds in the application

Video does not pause when another video on the page begins to play

Video issues when the device/application has two or more screens

Sound lags behind the video or on-screen action, or video/action lags behind the sound

Graphical user interface defects exist in every application.

The reasons vary remaking the application for a different platform and static design leads to issues on devices with different screen resolutions. There are many types of such defects.

Graphic issues

Here is the list of the most-common GUI issues found in mobile applications:

Portrait/landscape mode issues

Resizing issues

Animations that do not run smoothly

Inconsistencies with action elements and their place in the design

Non-retina display for devices with retina support

Inconsistency with platform standards (e.g. iOS HIG)

Font size and type issues (cut text and elements overlapped)

Keyboard issues (e.g. standard keyboard in case of numeric-only field)

Grammar issues

When about it comes to network-specific issues, most defects are caught when there is some type of switch switching between Wi-Fi points and switching from EDGE to 3G. It is also necessary to check what happens with a poor connection (real or simulated).

Of course, if the application needs to transfer large amounts of data, crashes can be caught during normal EDGE connection, while in airplane mode, or with cellular data switched off.

All of these should be checked at all key points of the application installation, launching, registering, logging in and submitting forms. The application should react correctly every time working normally, or displaying a message asking the user to check the connection.

Potential defects

The next group includes many potential defects. Hardware and software factors are too different, especially for Android devices, to cover all of them in this article, but let us try to highlight the main areas of concern:

Low operating memory issues (memory warnings)

In-app purchase issues

Gestures reaction

Interruptions (calls, messages and low battery)

GPS (location services) issues

SD card issues

External devices issues (display and keyboard)

Browser, email client and other OS-specific services issues

OS compatibility issues

These are the main groups of issues specific to hardware or software. There may be no defects or an overwhelming number in each of them.

Please note if your application uses any of the device functions (e.g. location services) you must include deep checks of this functionality in your test plan.

Data-related issues are specific to each application type. Games will have data issues that are completely different from enterprise applications.

Points to check

Data types can also be different. Here are the main points to check:

Time settings (Does the application use server time or phone time? Is the time determined correctly?)

Correct application of check-ins/adding friends/sending gifts and other data exchange between the device and server

Ensuring that progress is saved in the application

Desktop/Web profiles, if any, are correctly synchronized

File types all necessary types are supported, and unsupported types are correctly processed with appropriate messages

Large and small files are correctly processed

Caching issues

Redirecting from the application to the Web and vice versa

Social network content

The groups outlined above are not comprehensive, but provide a high-level checklist that should be augmented with specific checks based on the type of application being tested.

For rare defects, consider conducting research on application reviews in markets or social networks particularly if this is not the first version of the application or there are similar applications already available.

These defects can be caught during ad hoc testing or with special complex scenarios based on prior experience.

Here are several examples:

1. A game crashed if there was a need to download audio files from the server larger than 25MB
2. A blank white space appeared on the Settings screen when the user quickly changed the On/Off setting (five times in a second)
3. "Post to Twitter" posted tweets not to the current account timeline, but to the timeline of the account which was the first used in the application

THIS ARTICLE IS not a complete solution for the creation of test documentation, but these are basic tips to start writing a test scenario.

Remember that the point is to include all possible sources of defects functional specification, your experience and social network reviews.

The final test documentation for each application will be different, and may vary greatly for different projects.

Your skills, experience and knowledge of your application will be key in finding your own best solution.

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

© 2020 Napean LLC. All rights reserved.

American Marketer is published each business day. Thank you for reading us. Your **feedback** is welcome.