***Testing***: Page

1- Mobile site testing strategy key points…………………………………………………………………

1-1 Devices selection …………………………………………………………………………………….

1-2 Emulators or simulators? …………………………………………………………………………

2- Mobile manual and automated testing …………………………………………………….…………

3- Mobile application testing stages ……………………………………………………………….………

3-1. Documentation Testing ……………………………………………………………………

3-2. Functional testing …………………………………………………………………………….

3-3. Usability testing …………………………………………………………………….…………

3-4. UI (User Interface) testing …………………………………………………………………

3-5. Compatibility (Configuration) testing …………………………………….…………

3-6. Performance testing …………………………………………………………………………

3-7. Security testing …………………………………………………………………………………

3-8. Recovery testing ………………………………………………………………………………

3-9. Localization testing …………………………………………………………………..………

3-10. Change related testing ……………………………………………………………………

3-11. Beta testing ……………………………………………………………………………………

3-12. Certification testing …………………………………………………………………………

4-Tips to test mobile application …………………………………………………………………………………….

5-Test mobile application with EasyQA SDK ……………………………………………………………………

**Mobile site testing strategy key points**

Now, we can think about our testing strategy. Let’s consider the main points and challenges we should face to.



**Devices selection**



There is no doubt, that the real device is the best decision if you want to test mobile application. Testing on a real device always gives you the highest accuracy of results.

In fact, this is really not easy to choose the most appropriate device. Anyway, here are some actions you should do while selecting device for the mobile testing:

* Make the analysis to define the most popular and used gadgets in the market.
* Choose devices with different OS.
* Choose devices with different screen resolutions.
* Pay attention to the next factors: compatibility, memory size, connectivity etc.

As it was mentioned before you have lots of **advantages** for testing mobile apps on the real devices:

* High accuracy of the testing result.
* Simple bug replication.
* The points like battery drainage, geolocation, push notifications, devices built-in sensors are easy for testing.
* Ability to test incoming interrupts (calls, SMS).
* Ability to test mobile application in the real environment and conditions.
* No false positives.

And also some **disadvantages**:

* A huge number of the often used devices.
* Additional expenses for the maintenance of the devices.
* Limited access to the devices often used in the foreign countries.

As you can see testing on the real devices is the good decision, but also it has some limitations. You should overcome them to make mobile apps testing process real effective.

**Emulators or simulators?**



There is no difficult to guess, that they are special tools which emulate/simulate functionality and behavior of the mobile devices.

“Emulator” and “simulator meanings are often confused. Despite theirs almost similars pronunciation, they have no equal meaning.

In fact, an emulator is the original device replacement. Though you can run soft and apps on your gadget, you have no ability to modify them.

The simulator doesn’t replicate device’s hardware, but you have an ability to set up the similar environment as the original device’s OS.

So, it is better to use mobile simulators to test mobile application. Emulators are more appropriate for the mobile site testing.

Some **advantages** of using the simulators to test mobile application:

* Easy setup.
* Fast working.
* Helps to verify and explore the behavior of your mobile app.
* Cost effective.

Some**disadvantages** of using the simulators to test mobile application:

* Device hardware is not taken into the consideration.
* False positives are possible.
* Incomplete data of the simulation results, which makes some difficulties for the complete analysis of the test results.

**Cloud-based testing of the mobile application**



Testing mobile applications with cloud-based tools seems to be the optimal choice. It can help you to overcome disadvantages of the real devices and simulators.

The main advantages of this approach:

* Easy availability.
* An ability to run mobile devices on multiple systems and networks.
* An ability not only to test, but also update and manage apps in the cloud.
* Cost effective.
* High scalability.
* The same script can be run on several devices in parallel.

Some weak points of the cloud mobile testing:

* Less of the control.
* No so high level of the security.
* Dependence of the Internet connection.

Some useful cloud-based tools, which can help you to test mobile application: [**Xamarin Test Cloud**](https://www.xamarin.com/test-cloud)**,**[**Perfecto Mobile Continuous Quality Lab**](https://www.perfectomobile.com/continuous-quality-lab)**,** [**Keynote Mobile Testing**](https://www.keynote.com/resources/videos/deviceanywhere-mobile-test-automation)**.**

**Mobile manual and automated testing**



Nowadays many specialists support the opinion that manual testing is going to die. Sure, it is not true. Of course, we can not do without test automation, but there also situations when manual testing is preferable.

Some **merits** of the manual mobile application testing:

* It is more cost effective in the short-termed period.
* Manual testing is more flexible.
* Better simulation of user actions.

Some **demerits** of the manual mobile application testing:

* Manual test cases are hard to be reused.
* Less effective of execution certain and constant task.
* Test running process is slow.
* Some kinds of test cases couldn’t be executed manually ( load testing).

Some **advantages** of the app automation testing:

* Test running process is quite faster.
* Cost effective in the long-termed period.
* Automated test cases are easy to be reused.
* The only decision for some kind of testing (perfomance testing).
* Test results are easy to be shared.

Some**disadvantages** of the app automation testing:

* Automated mobile testing tools have some limitations.
* Time-consuming process.
* Automated testing has less efficiency in the defining user-friendliness or positive customer experience.

As you can see you should make different decisions creating your strategy for the mobile testing. Of course, there are no univocal answers on them.

The combination of different approaches seems to be the optimal way. For example, you can use simulators in the earliest stages of your testing process. But is better to use real devices (physical or cloud-based) in the final stages. Automated testing is preferable for the load and regression testing. But manual mobile testing tools are better to be used for usability and exploratory testing.

**Mobile application testing stages**

So, let’s start to consider the main stages of the mobile app testing process. They more mostly similar to the website testing stages. Mostly, but not quite similar. As you have read before, there are some basic differences between mobile and desktop applications. Therefore, we need to pass some additional stages and make some additional verifications.



**1. Documentation Testing**



Documentation testing is the necessary preparatory stage of the mobile application testing process.

Actually, testing begins before software development process. Testers get navigational charts, screen layouts, other requirements invisible on the design. These requirements are analyzed for completeness and inconsistency. Contradictions in the requirements must be resolved before the start of development.

Artifacts like Requirements (Specification, PRD), Test Plan, Test Cases, Traceability Matrix are created and analyzed on this stage.

**2. Functional testing**



Functional testing is aimed to ensure that it is working as per the defined requirements. In simple terms, we check whether the application performs the expected functions, which are usually described in the specification or correspond to the logic of business processes. Pay attention to the next important factors while providing functional testing of your mobile app:

* The application type, which is defined by its business functionality (social networks, banking, education, ordering and delivery of food, tickets, the game industry etc.).
* Target audience (companies, users, educational environment etc.).
* Distribution channels (direct delivery, Google Play, App Store, etc.)

Now, let’s consider the main verifications, which should be passed to test mobile application functionality.

**Installing and running the application**

* The installation of the application should take place without significant errors, if the device meets the system requirements.
* Verify the application automatically starts correctly.
* Ensure the user manual is available.
* Ensure the application’s operation during startup/exit meets the basic requirements.

**Fields testing**

* Verify the required fields work correctly.
* Make sure that mandatory and optional fields are displayed in different ways.

**Business functionalities testing**

* Verify the declared price and content correspond to the user got information.
* Ensure the user can perform typical operations: buying, adding goods to the cart, ordering goods etc.
* Make sure the application supports payment transactions through payment systems like Visa, Mastercard, PayPal etc.
* Check the recovery of the purchase regardless of the device, but with an account binding.

**Interruptions testing**

* Incoming and outgoing calls, SMS, and MMS.
* Battery discharge/removal.
* Disconnecting and connecting the network/Wi-Fi.
* Disconnecting and connecting the SD-card.
* Charging the device.

**Constant users feedback testing**

* Downloading content messages
* Progress bar.
* The appropriate reaction of the buttons on pressing.
* Network access error messages.
* Attempt to delete important information messages.
* Availability and synchronization of sound, vibration, and visual notifications.
* The appearance of a screen (message) at the end of the process (game).

**Update testing**

* All user data is saved after updates.
* Ensure the update progress is displayed properly.
* Make sure updates are supported by older operating systems.
* Testing various ways of installing updates (Wi-Fi, Bluetooth, USB)

**Device resources testing**

* Lack of space to install or run the application.
* Memory leaks. Pay attention to windows, with a lot of information, and tasks with long workflow.
* Installing/replacement the app on the SD-card.
* The absence of some functions supported by the application (3G, SD-card, etc.).
* Ensure the installed application does not interfere with the normal operation of other apps and does not consume their memory.

**Some other verifications:**

* Games concerned verifications: correctness of connecting/disconnecting players, players connection via different networks etc.
* Make sure the information error messages are correct on time and appropriate.
* Verify connection to the analytical tools like [Google Analytics](https://www.google.com/analytics/).
* Testing the power consumption.
* Verify the necessary options correct work with social networks – **Share, Publish, Navigation.**

Some useful tools to test mobile application functionality:[**Appium**](http://appium.io/)**, Selendroid, Robotium, Ranorex.**

**3. Usability testing**



Usability testing is aimed to ensure the convenience of using the application, creates an intuitive interface that conforms to accepted standards. It is performed to create fast and easy-to-use applications. Here are 3 main basic criteria for the apps evaluation:

* Satisfaction
* Efficiency
* Effectiveness

Let’s consider the simple checklist to test mobile application usability:

* Make sure that the buttons are of the normal size and placed in one area of the screen
* Verify the app works in multitasking mode, when necessary.
* Check the navigation of the important application modules.
* Ensure the icons and pictures look natural in the app environment.
* Verify the color of the buttons that perform the same function is the same.
* The text should be simple, clear and visible to the user. Short sentences and paragraphs are possible to read.
* Define the optimal font size.
* Ensure correct operation of the Zoom-in and Zoom-out system.
* Verify the context menus are not overloaded.
* Make sure that the application can be terminated by any state and that it resumes operation in the same state.
* Ensure that the application components are synchronized with the user’s actions.
* Verify the user can return or cancel the action if he/she pressed the wrong button.
* Verify the speed of response of the element is high enough

Some useful tools to test mobile application usability: [**User Zoom**](https://www.userzoom.com/)**,**[**Reflector,**](https://reflector.en.softonic.com/) [**Loop HYPERLINK "https://www.loop11.com/"11**](https://www.loop11.com/)**.**

**4. UI (User Interface) testing**



User Interface (UI) testing is performed to ensure the graphic user interface of your app meets the specifications.

Here are some verifications to test mobile application UI:

* Ensure the compliance with the standards of UI
* Check your app’s UI with the standard screen resolutions: 640 × 480, 800 × 600, 1024 × 768, 1280 × 800, 1366 × 768, 1400 × 900, 1680 × 1050.
* Verify responsiveness of applications on different devices.
* Test the main design element: buttons, icons, colors, links, fonts, font sizes, layout, text boxes, text formatting, labels, captions, buttons, lists etc.
* Verify advertising does not overlap application control buttons.
* Ensure the advertising has an accessible closing button.
* Make sure the correct display of various elements on retina and non-retina screens.
* Verify all elements display with portrait and landscape page orientation.

Some useful tools to test mobile application interface: **FitNesse, iMacros,** [**Coded UI,**](https://msdn.microsoft.com/en-us/library/dd286726.aspx) **Jubula,** **LoadUI.**

**5. Compatibility (Configuration) testing**



Compatibility (Configuration) testing is conducted in order to ensure optimal application performance on different devices – taking into account their size, screen resolution, version, hardware, etc. You should pay attention to the next points:

* OS Configuration
* Browser Configuration
* Database Configuration
* Device Configuration
* Network Configuration

**Cross-platform testing** helps you to test mobile application in different OS: Windows, iOS, Android, and BlackBerry etc.

**Cross-browser testing** allows ensuringthe correct work of the app in different browser configurations: Mozilla Firefox, Google Chrome, Opera Mini etc.

**Database testing** is aimed to verify the correct work of your application in different database configurations: Oracle, DB2, MySQL, MSSQL Server, Sybase.

**Device Configuration testing** should take into account such parameters:

* Device type: smartphone, tablet, etc.
* Device configuration: RAM, processor type, screen resolution, battery capacity, etc.

**Network configuration testing** is performed to ensure the correct work in different network configurations (GSM, TDMA) and standards (2G, 3G, 4G).

Some tips to test your mobile application compatibility:

* Create a coverage matrix (the table in which all possible configurations are entered).
* Prioritize configurations.
* Check each configuration, step by step, in accordance with the set priorities.

Some useful tools to test mobile application performance compatibility: [**BrowserStack HYPERLINK "https://www.browserstack.com/",**](https://www.browserstack.com/) **CrossBrowserTesting by Smart Bear,**[**Litmus HYPERLINK "https://www.browsera.com/",  HYPERLINK "https://www.browsera.com/"Browsera**](https://litmus.com/)**,**[**Rational HYPERLINK "https://www-03.ibm.com/software/products/ru/clearcase"Clearcase HYPERLINK "https://www-03.ibm.com/software/products/ru/clearcase" by IBM**](https://www-03.ibm.com/software/products/ru/clearcase)**,  Ghostlab.**

**6. Performance testing**



Performance testing is a set of types of testing, the purpose of which is to determine the operability, stability, resource consumption and other attributes of application quality under different usage scenarios and loads.

The main aims of the performance testing:

* Checking the response time of the application to various types of requests, in order to make sure that the application is working according to the requirements for the normal user load. **(Load testing).**
* Testing the working capacity of the application at loads exceeding the user’s several times. **(Stress testing).**
* Examine the operability of the application for long time work, under normal load. **(Stability testing).**
* Check work in the conditions of the “expanded” database, under the normal time. **(Volume testing).**
* Determine the number of users who can simultaneously work with the application. **(Concurrency testing).**

Some verifications for performance testing your mobile app:

* Determine whether the application is running the same under different network conditions.
* Find various application and infrastructure bottlenecks that reduce application performance.
* Evaluate the ability of the app to cope with planned load volumes.
* Verify the response time of the application meets the requirements.
* Check the application stability under conditions of a hard user load.
* Ensure the performance of the application if it works under conditions of a non-permanent connection to the Internet.
* Make sure the existing client-server configuration provides optimal performance.

Some useful tools to test mobile application performance: **NeoLoad by Neotys,** [**HYPERLINK "http://www.neotys.com/neoload/overview"Apteligent HYPERLINK "http://www.neotys.com/neoload/overview" (formerly HYPERLINK "http://www.neotys.com/neoload/overview"Crittercism HYPERLINK "http://www.neotys.com/neoload/overview")**](https://www.apteligent.com/)**,**[**New Relic.**](https://newrelic.com/why-new-relic)

**7. Security testing**



Security testing is aimed to check the security of the system, as well as to analyze the risks associated with providing a holistic approach to application protection, hackers, viruses, unauthorized access to sensitive data.

Some verifications you have to pass to test mobile application security:

* Ensure the data of users of the application **(logins, passwords, bank card numbers)** are protected from network attacks of automated systems and can not be found by selection.
* Verify the application security system requires a strong password and does not allow the attacker to seize the passwords of other users.
* Make sure that the application does not give access to sensitive content or functionality without proper authentication.
* Protect the application against attacks of the SQL injection type.
* Protect the application and the network from DoS Attacks.
* Protect the application from malicious attacks on clients.
* Protect the system from malicious implementations when the program is running.
* Provide session management to protect information from unauthorized users.
* Prevent possible malicious consequences of file caching.
* Examine user files and prevent their possible harmful effects.
* Analyze the interaction of system files, identify and correct vulnerabilities.
* Prevent possible malicious actions of cookies.

Some useful tools to test mobile application security: [**Retina CS Community**](https://www.beyondtrust.com/products/retina-cs-community/)**,**[**OWASP Zed Attack Proxy**](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)**,**[**Veracode,**](https://www.veracode.com/) [**Google HYPERLINK "https://github.com/google/nogotofail"Nogotofail HYPERLINK "https://github.com/google/nogotofail",**](https://github.com/google/nogotofail) and [**SQL Map.**](http://sqlmap.org/)

**8. Recovery testing**



Recovery test verifies the app under test in terms of its ability to withstand and successfully recover from possible failures caused by software errors, hardware failures, or communication problems.

Here is the list of the verifications for the recovery testing:

* Verify the effective recovery of the application after unforeseen crash scenarios.
* Ensure the process of data recovery after a break in the connection.
* Test the recovery after a system failure and a transaction failure.
* Verify the ability of the application to process transactions in the event of a power failure (low battery, incorrect application shutdown etc.).

**9. Localization testing**



Localization testing allows you to test mobile application adaptation for a specific target audience in accordance with its cultural specifics.

Some verifications for the localization testing:

* Determine languages supported by the application.
* Ensure the correctness of the translation.
* Verify the correctness of the translation in accordance with the theme of the application
* Check the date formats.
* Check the delimiters in numbers.

Of course, the native speakers are preferred to perform localization testing of the mobile app.

[**Ubertesters**](https://ubertesters.com/)**,** [**eggPlant** HYPERLINK "https://www.testplant.com/eggplant/testing-tools/eggplant-mobile-eggon/"](https://www.testplant.com/eggplant/testing-tools/eggplant-mobile-eggon/) can be useful to test mobile application localization.

**10. Change related testing**



So, you passed all mentioned stages and found some bugs. Therefore, some changes have been made to the code of your app.

The key goals of the change related testing:

* Verify your team has successfully fixed all detected bugs  (**Re-testing or Confirmation testing**). Put it simply, the test cases that originally detected the bugs are run again. And this time they should be passed with no bugs.
* Verify the new changes did not lead to the appearance of new bugs. (**Regression testing**). Actually, providing regression testing, you should pass not only test cases with detected bugs, but also test cases checking all functionalities of your app.

Some useful tools for change related testing of your app:[**Appium**](http://appium.io/)**, Robotium, Ranorex.**

**11. Beta testing**



Finally, you have the prerelease full functionality version of your mobile app. It would be better to evaluate the possibilities and stability of the program in terms of its future users.

**Beta testing** is the stage of debugging and checking the beta version of the program. Its main purpose is identifying the maximum number of errors in its work for their subsequent elimination before the final release of the app to the market.

People who have experience with working with similar type apps, better yet, with the previous version of the application are chosen to the role of beta testers.

You should pay attention to the next factors before providing beta testing of your mobile app:

* A number of testing participants.
* Testing duration.
* Shipping
* Demographic coverage
* Testing costs.

Though you need to spend some money for beta testing, it could be a good investment in the quality of your mobile app.

Some popular platforms for beta testing of the mobile apps: [**HockeyApp**](https://hockeyapp.net/)**, Ubertesters,** [**TestFlight**](https://developer.apple.com/testflight/) .

**12. Certification testing**



There are certain rules for organizing an installation file **(.apk)** and rules for applications design for each application store. Certification testing verifies the app meets the requirements of the most popular stores like Google Play, the App Store, and Windows Phone.

Let’s consider the main criteria for application compliance with standards, licensing agreements and terms of use.

**Android:**

* The installation file for the application (.apk) matches with [**Program Policies**](http://www.android.com/us/developer-content-policy.html) **.**
* The application meets the requirements of the[**UIG**](https://developer.android.com/guide/practices/ui_guidelines/index.html)**.**
* There are no viruses in the app. Android market semi-automatically checks the application for viruses and could block you account if detect them.
* You should follow the order of version control in the case of publishing an updated version of your app.

**iOS:**

* The application meets the requirements of the [**Human Interface Guidelines**](https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf)**.**
* The application must have a unique name.
* You need to provide a link for feedback from the developer.
* The application should be put to the determined particular category.
* App Store test the app for compatibility.
* App doesn’t contains prohibited materials, unforeseen delays in work or repetition of existing functions.

**Tips to test mobile application**

Let’s systematize our knowledge, and try to determine the main tips for mobile application testing.

* Learn the app you are going to test.
* Remember the differences between desktop and mobile apps.
* Take into account the operating system and hardware specifics
* Use real devices when it is possible.
* Don’t Try to Find the “Swiss Army Knife” of Testing. Use the tools you are familiar with.
* Use the advantages of the cloud mobile testing.
* Confirm your findings with screenshots, logs and videos.
* Provide your mobile app testing both for portrait and landscape screen mode.
* Use the development menu options for  iOS and Android.
* Do not neglect (but do not abuse) emulators and simulators for testing.
* Verify performance of your app.
* Don’t automate everything
* Get real users to test your app
* Release the time to work out more complex, unconventional test scenarios (f.e. use test “monkeys”).
* Consider the human factor

**Test mobile application with EasyQA SDK**

Practice is the best way to learn something. You have an ability to use some features of the EasyQA for testing your mobile app.



EasyQA SDK is the set of features, developed by our team, to make software testing process more effective and simpler.

Here are its key features:

* Cras reporting
* In-app bug reporting
* Making screenshots
* Video recording
* It works both with Android and iOS apps.