QA MADNESS

MOBILE APP TESTING: FROM A TO Z

QA MADNESS

Types of Mobile App Testing

FUNCTIONAL TESTING
Every function of an application is working as intended and acting by predetermined requirements.

USER INTERFACE (UI)
It's about testing the graphical elements to verify that they are functioning according to requirements.

USABILITY TESTING (UX)
Allows us to measure evaluation of the compliance of the application design with its functionality specified by the customer.

COMPATIBILITY TESTING
Ensures the consistent functioning of an app across devices and software ecosystems.

ACCESSIBILITY TESTING
We run mobile accessibility testing to verify that an app has been adjusted for accessibility needs.

PERFORMANCE TESTING
We can learn about an app's stability, resource consumption, and ability to withstand high loads.

LOCALIZATION TESTING
Ensures that an application has been adjusted to a target market in terms of language, cultural characteristics, and specifics of a particular country or region.

INTERNATIONALIZATION TESTING
Verifies that an app has been adjusted to a particular region.

SECURITY TESTING
Allows uncovering existing and potential vulnerabilities, threats, risks in a mobile application, and loopholes in the processes.

CHANGE-RELATED TESTING
Once developers fix bugs, QA engineers need to verify the changes and make sure the rest of the functionality is still working.

API TESTING
Helps to determine software strengths at the early stages of development. Therefore, API tests detect small errors that might become more serious while running GUI.

END-TO-END (E2E) TESTING
A QA engineer checks a user flow from the beginning to the end. It gives a clear perspective of a user journey through some part of a mobile app.

ACCEPTANCE TESTING
It is a final verification of the business functionality. The QA team goes through a product's functionality to see if the software system meets specification requirements.

QA MADNESS

Mobile App Testing Approaches

The approaches to mobile app testing are often based on ***timing.***

PROACTIVE APPROACH	REACTIVE APPROACH
A test design process starts early and aims to prevent issues before even code is ready. The QA team starts with review of requirements to prevent bugs and shape software quality from the very beginning.	Implies working with an end-product, when its design and functionality are supposedly ready for use. Utilizing the reactive approach, a testing team waits to run tests at the late stage of an SDLC.

QA MADNESS

Areas to Focus on in Mobile Testing App Testing

- Installation / de-installation / update process.
- Location & VPN.
- Data synchronization.
- Interruptions of the application by cells, messages, etc.
- An ability to run the app in the background.
- Payment gateway.
- Communication (chat, audio, video).
- Touchscreen (tap, swipe, pull, shake, pinch).
- Voice commands.

QA MADNESS

Native Apps vs Hybrid Apps

NATIVE MOBILE APPS	HYBRID MOBILE APPS
Developed for a specific platform – Android or iOS. So if a service is present in both stores, these are two separate applications.	Universal, so one app can be installed on both platforms. It is possible thanks to mixing native and web elements in the development.

The testing approach will be the same for both types of apps, but…

1. The ***functional*** checklists or test cases will focus on the common areas.

2. Performance issues are the legacy of hybrid apps.

3. Hybrid applications require spending more time on ***UI testing***

4. Testing of hybrid apps tends to require more real devices for ***compatibility testing*** than native app testing.

QA MADNESS

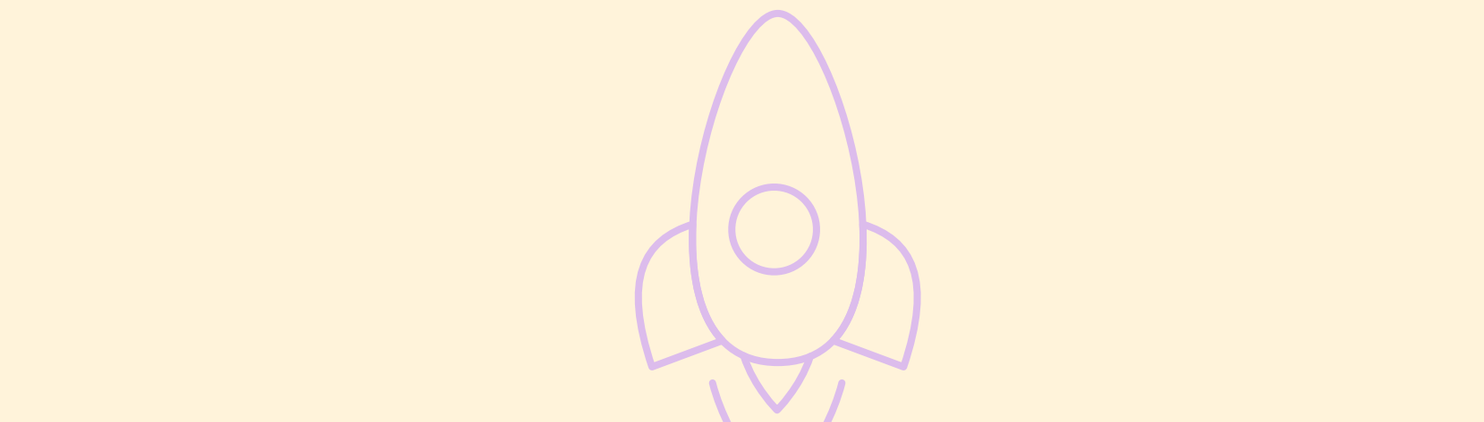
Android App Testing vs iOS App Testing

Android is represented by a vast variety of devices and OS versions. Their combinations are more numerous than in iOS. Thus, there is a bigger scope of work for Android device testing when it comes to ***cross-platform*** checks.

To be more specific:

- Since Android devices come with more screen sizes and resolutions, there is more room for layout issues and interface bugs. Thus, the scope of tasks for ***UI testing*** is also bigger compared to iOS.

- Android offers lots of software and hardware configurations, so again, ***performance and compatibility testing*** would require more time.



QA MADNESS

Real-Device Testing

Testing or real devices allows a QA team to assure that an application works smoothly on a variety of smartphones. For example, only using a real device, you can properly check:

- Colors on screens with different resolutions;
- Touchscreen with all the swipes, scrolls, and taps;
- Location, camera, audio, and other device-specific features;
- Different software and hardware configurations;
- Different software and hardware configurations;