

Discussion-7 What are the different types of testing available for Android apps and when should each be used?

Discussion Topic:

Discussion-7 What are the different types of testing available for Android apps and when should each be used

My Post:

Hello Class,

Before releasing your Android app to the public, it is important to test your app's consistency by verifying its correctness, functional behavior, and usability before you release it publicly (Android Developers n.d.). It is two major types of testing, functional and non-functional. This post focuses on functional testing types by providing an overview of them and briefly explores non-functional types.

Functional Testing Types

Unit Testing

Unit testing is one of the prevalent functional testing types, they should make up approximately 70% of your total number of tests. Unit testing involves testing individual units or components of your app in isolation (GAT Staff Writers, 2024). In other words, they verify small portions of the code app like individual methods or classes. They do not require an Android emulator or device as they are executed on the developer's PC or workstation using Java Virtual Machine (JVM) (BrowserStack, 2025a). This makes the test ideal for testing business logic. The most commonly used tool for unit testing is JUnit, other tools are Mockito and Truth.

Integration Testing

Integration testing is implemented on components or modules that have been unit tested (Singh, 2024). The goal of the test is to observe data flow and identify defects in it. In other words, it examines how unit tested components work together to identify defects (BrowserStack, 2025a). The most common tools used for integration testing are Robolectric, JUnit, Mockito, and Truth.

UI Testing

User Interface (UI) testing is used to verify both the appearance and functionality of the UI by evaluating the visual elements, layout, responsiveness, and overall user experience (BrowserStack, 2025a). This involves behavior testing which examines how the UI responds to user actions and screenshot testing which is used to validate the visual accuracy of the UI across different devices (screen sizes and resolutions). The most commonly used tools used for UI testing are Espresso and UIAutomator.

End-to-End Testing

End-to-end (E2E) is used to verify an application's entire workflow from start to finish by simulating real-world scenarios. It verifies that all components of the application, including the user interface, business logic, data persistence, and external connection (API), function correctly together (BrowserStack, 2024b).

The most common tools used for E2E Espresso and UIAutomator. Other tools are Appium which has cross-platform capabilities; and BrowserStack and LambdaTest which are cloud based testing platforms that offer a wide range of real device testing.

Non-Functional Testing Types

Non-functional testing is used to verify aspects of a software application that are not directly related to its specific functions or features, but rather how well the system operates overall (GAT Staff Writers, 2024). In other words, it focuses on the quality of the software, such as performance, security, usability, and reliability. Non-functional testing is composed of a set of tests that includes performance testing that evaluates the speed, responsiveness, and stability of the app under different conditions; usability testing which is used to assess how easy is the app to use; compatibility testing which is used to verify consistent functionality of the app across different devices and OS versions; accessibility testing which is used to access if the app is usable for users with disabilities; regression testing is used to test updates for bugs; and smoke testing is used to quickly check of the app's functionalities after a build.

Summary

Before successfully releasing an Android app, rigorous testing is required. Testing such as functional testing (unit, integration, UI, end-to-end), which checks what the app does, ensuring correct functionality at all levels; non-functional testing (performance, security, usability, etc.), which focuses on how well the app performs. Combining both approaches ensures that the app is free of bugs, that it is user-friendly and secure.

-Alex

References:

Android developer (n.d.). *Fundamentals of testing Android apps*. Android.
<https://developer.android.com/training/testing/fundamentals>

BowserStack (2025a, March 18). *What is Android unit testing?* BowserStack.
<https://www.browserstack.com/guide/android-unit-testing>

BowserStack (2024b, June 28). *End To End testing: Tools, types, & best practices*. BowserStack.
<https://www.browserstack.com/guide/end-to-end-testing>

GAT Staff Writers (2024, May). *What is Android testing - Types, tools and best practices*. Global App testing. <https://www.globalapptesting.com/blog/what-is-android-testing>

Singh, R. (2024, June 13). *12 mobile app testing types: A thorough exploration for QA professionals*. Headspine. <https://www.headspin.io/blog/types-of-mobile-app-testing>