

# Android Testing Agenda

- **What is testing?**
  - Benefits of software tests
  - Unit tests compared with Functional and integration tests
  - Test driven development and Continuous integration
- **The Gradle Build System**
  - Product Flavors and there usage for automated testing
- **Unit Testing with JUnit**
  - JUnit 4.x
  - Using JUnit test runner
  - Unit and instrumented unit tests in Android
  - Using JUnit Rules and parameterized tests
- **Developing unit tests for Android**
  - Overview of the Android instrumentation API
  - Developing unit tests for Android running on the JVM
  - Developing instrumented unit tests running on the Android runtime
- **Using Mockito for replacing object dependencies**
  - Using Mockito to create test fixtures
  - Mocking method call and parameter access
  - Mocking method calss and parameter access
- **Using advanced Assert frameworks**
  - Using and extending Hamcrest Matcher
  - Using and extending AssertJ Matcher
- **User interface testing within one application**
  - Using Espresso for single activity testing
  - Cross component testing within one application
  - Monitoring custom background activities

- Developing custom matchers and actions with Espresso
  - Mocking and testing intents
  - Testing Webview with Espresso
- Using Robolectric for unit testing**
    - Configuring Robolectric
    - Developing units tests with Robolectric
- Cross component and stress testing**
    - Cross application testing interface testing with Ui automator
    - Stress testing with Monkey
    - Scripting test cases with Monkeyrunner
- Android tools for application optimization**
    - Using the on-device Developer Options
    - Using tools as StrictMode, Lint, TraceView, HierarchyViewer and Systrace
    - Simulate device sizes and densities
    - Profile GPU rendering
    - Analyzing Overdraw
    - Analyzing memory allocation with heap dumps
- Build and test automation and continuous integration**
    - Outlook: Using code review system for test automatization
    - Outlook: Continuous integration with the Jenkins build server
- Software architecture for testing Android applications**
    - Using Dependency injection for Android application design
    - Modularization of the application components
    - Outlook: Using event systems and RxJava to design lightly coupled applications