Andriod Testing

* there are tons of devices and Android versions that your app must be compatible with.
* There is failure of android system when
  + Application installation failure
  + App crash during execution
  + Scaling/layout problem
  + Application hangs if some resource is not available
  + Problem in landscape/portrait mode
* Android testing strategy
  + Unit Test: min source code
  + Integration test: test inter operation between module
  + Operational tests: test the correctness of application requirement
  + System tests: test app in real environment
* Android platform comes pre-integrated[Junit](https://www.guru99.com/junit-tutorial.html)3.0 framework. It's open source framework for automating [Unit Testing](https://www.guru99.com/unit-testing-guide.html).
* An addition to Unit Testing is User Interface (UI) tests. These tests relate to UI components of your target application. UI tests ensure that your application return the correct UI output in response to sequence of user actions on device.
* The common way to performance UI tests on device is Android [Instrumentation](http://developer.android.com/reference/android/test/InstrumentationTestCase.html). But this has performance issues. One of the best tools to conduct UI testing on Android is [Robotium](http://code.google.com/p/robotium/) . Robotium makes it easy to write powerful and robust automatic black-box UI tests for Android applications.
* In [Integration Testing](https://www.guru99.com/integration-testing.html), all unit tested modules, are combined and verified.it involves the following testing
  + Activity testing: activity have lifecycle with states: [onCreate()](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)), [onStart()](https://developer.android.com/reference/android/app/Activity.html" \l "onStart()), [onResume()](https://developer.android.com/reference/android/app/Activity.html#onResume()), [onPause()](https://developer.android.com/reference/android/app/Activity.html" \l "onPause()), [onStop()](https://developer.android.com/reference/android/app/Activity.html#onStop()), and [onDestroy()](https://developer.android.com/reference/android/app/Activity.html#onDestroy())provided by activitymanager.It also checks the key and touch event
  + Service testing: tests the states in lifecycle of services(callback methods are onStartCommand(),onBind(),onUnbind(),onRebind(),onCreate(),onDestroy() .
  + Along with this it tests ,interaction between application and services. Use Servicetestcase class to test service
  + Content Provider testing : Store and retrieve data scross applications . Use ProviderTestcase2 class to test
* There's many testing frameworks are used to conduct integration test for Android such as Troyd, Robolectric, Robotium.
* ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Operational are also called Functional Tests or Acceptation Tests. They are high level tests designed to check the completeness and correctness of application.
* In Android, FitNesse is open-source framework that makes it easy to conduct operational tests for target application.
* In **System Testing** the system is tested as a whole and the interaction between the components, software and hardware is checked.
  + In Android, System Testing normally includes:
  + GUI testing : GUI testing is the process of testing the system's Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc.
  + **Usability testing** : Usability testing, a **non-functional** testing technique that is a measure of **how easily the system** can be used by end users. It is difficult to evaluate and measure but can be evaluated based on the below parameters: Level of Skill required to learn/use the software.
  + **Performance testing** : Performance testing, a **non-functional testing** technique performed to determine the system parameters in terms of responsiveness and **stability under various workload**. Performance testing measures the **quality attributes** of the system, such as **scalability, reliability** and resource usage.
  + **Stress Tests** : Stress testing a **Non-Functional testing** technique that is performed as **part of performance testing**. During stress testing, the system is **monitored after subjecting the system to overload** to ensure that the system can sustain the stress.
  + use tools like Traceview to conduct performance test on Android
* **MonkeyRunner:** A tool that provides APIs for writing program which control an Android device or emulator outside of Android code.