HUST

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

LẬP TRÌNH ỨNG DỤNG DI ĐỘNG

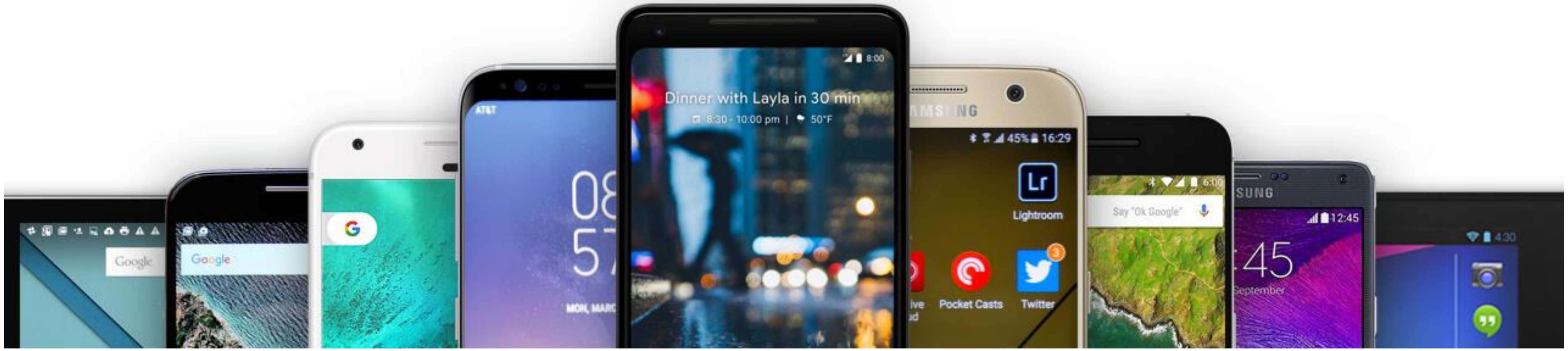
Mobile Application Programming

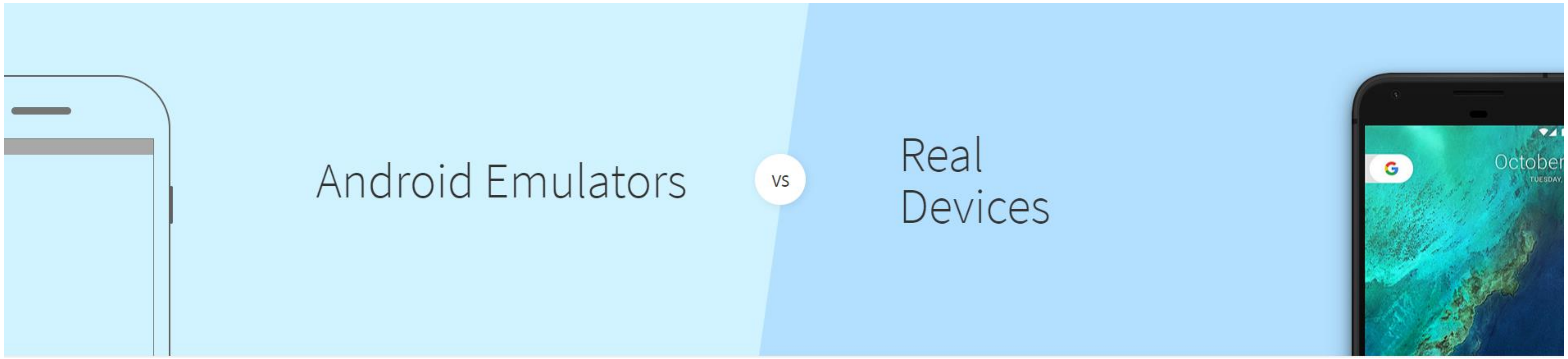
ET4710

PGS. TS. Đỗ Trọng Tuấn
Viện Điện tử Viễn thông * Đại học Bách Khoa Hà Nội

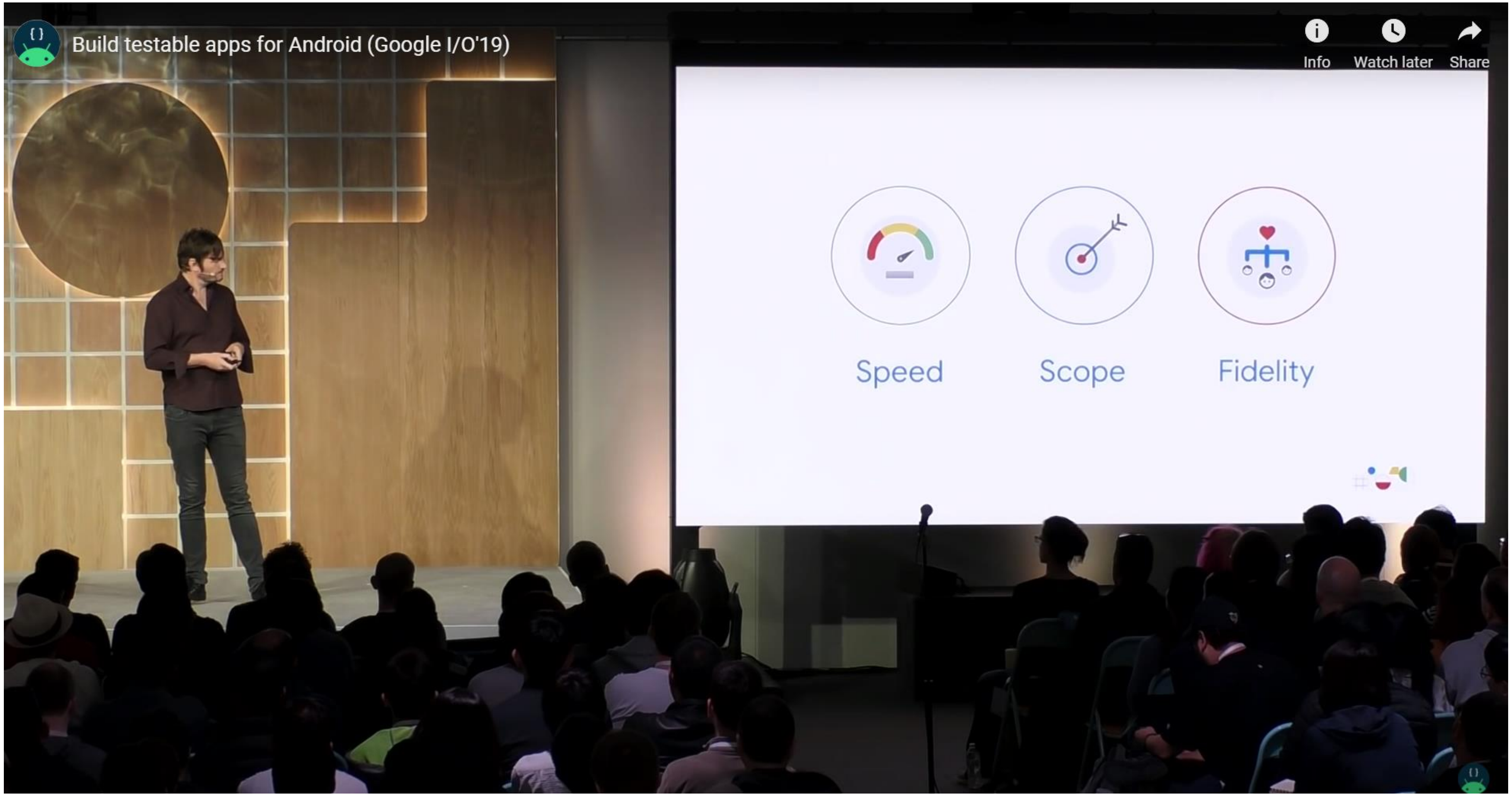
ONE LOVE. ONE FUTURE.

Mobile Application Programming



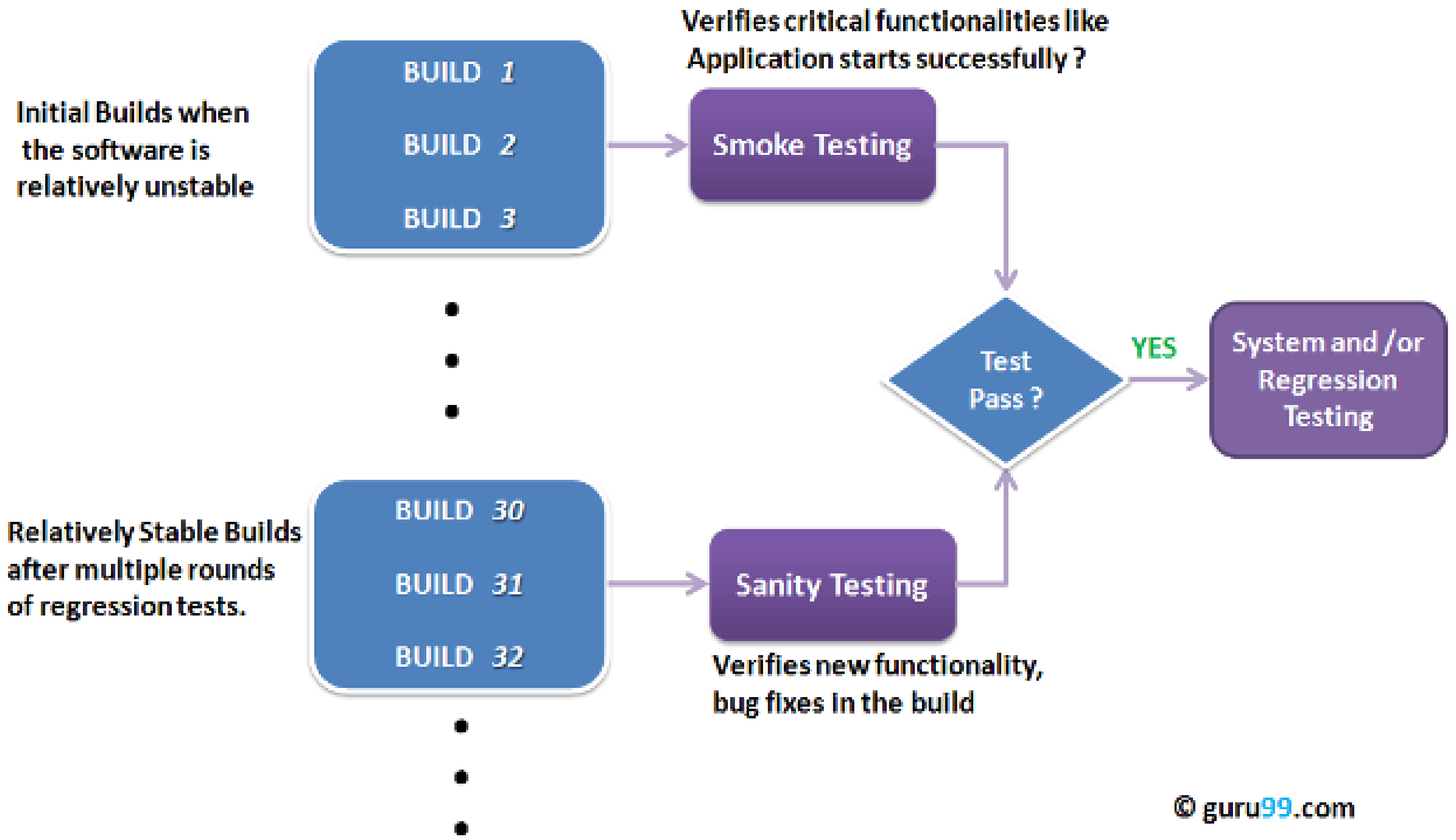


Mobile Application Testing



https://developer.android.com/training/testing

Mobile Application Testing

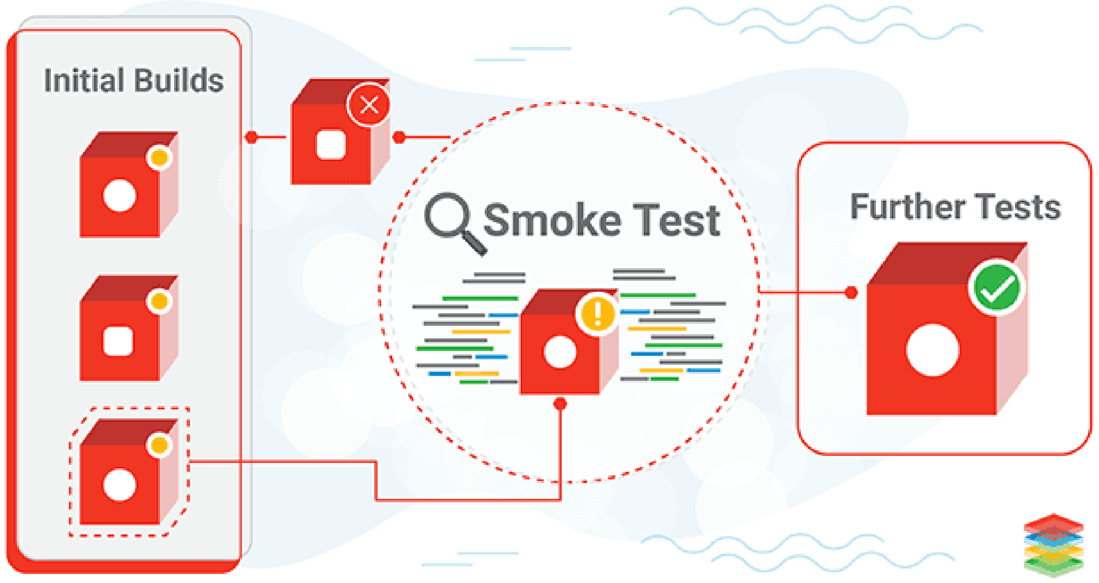

Verifies critical functionalities like Application starts successfully ?
Initial Builds when the software is relatively unstable
BUILD 1
BUILD 2
BUILD 3
Smoke Testing
Test Pass ?
YES
System and /or Regression Testing
Relatively Stable Builds after multiple rounds of regression tests.
BUILD 30
BUILD 31
BUILD 32
Sanity Testing
Verifies new functionality, bug fixes in the build
© guru99.com

https://www.guru99.com/smoke-sanity-testing.html

Smoke Testing?

Smoke Testing is a software testing technique performed post software build to verify that the critical functionalities of software are working fine. It is executed before any detailed functional or regression tests are executed. The main purpose of smoke testing is to reject a software application with defects so that QA team does not waste time testing broken software application.

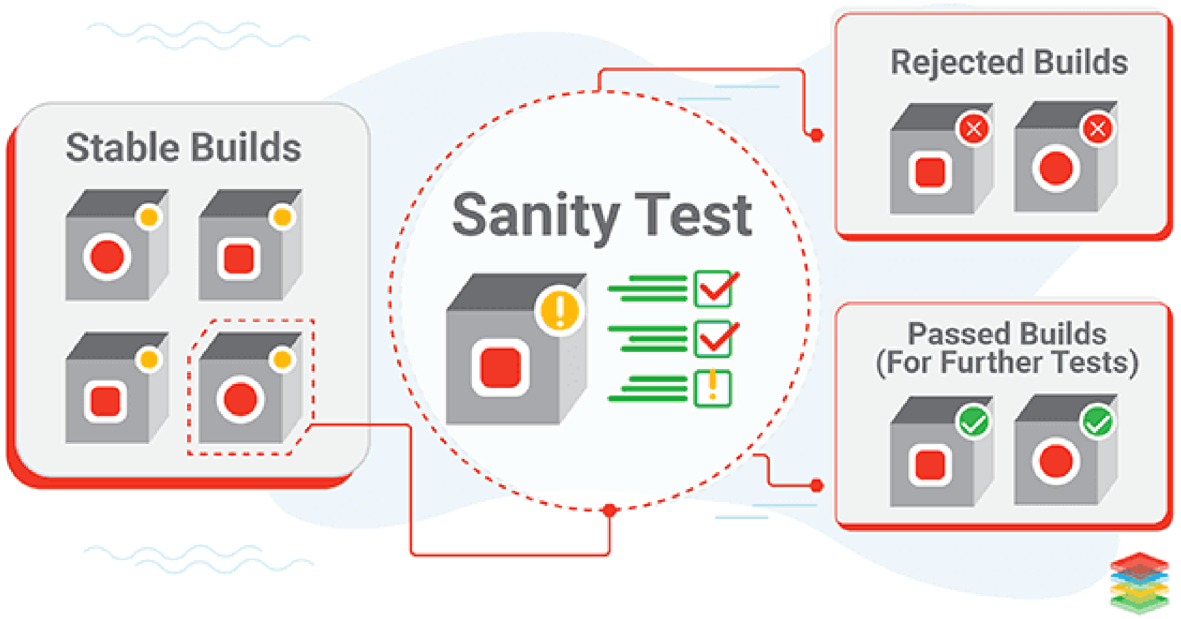
In Smoke Testing, the test cases chose to cover the most important functionality or component of the system. The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine. For Example, a typical smoke test would be - Verify that the application launches successfully, Check that the GUI is responsive ... etc.



Sanity Testing?

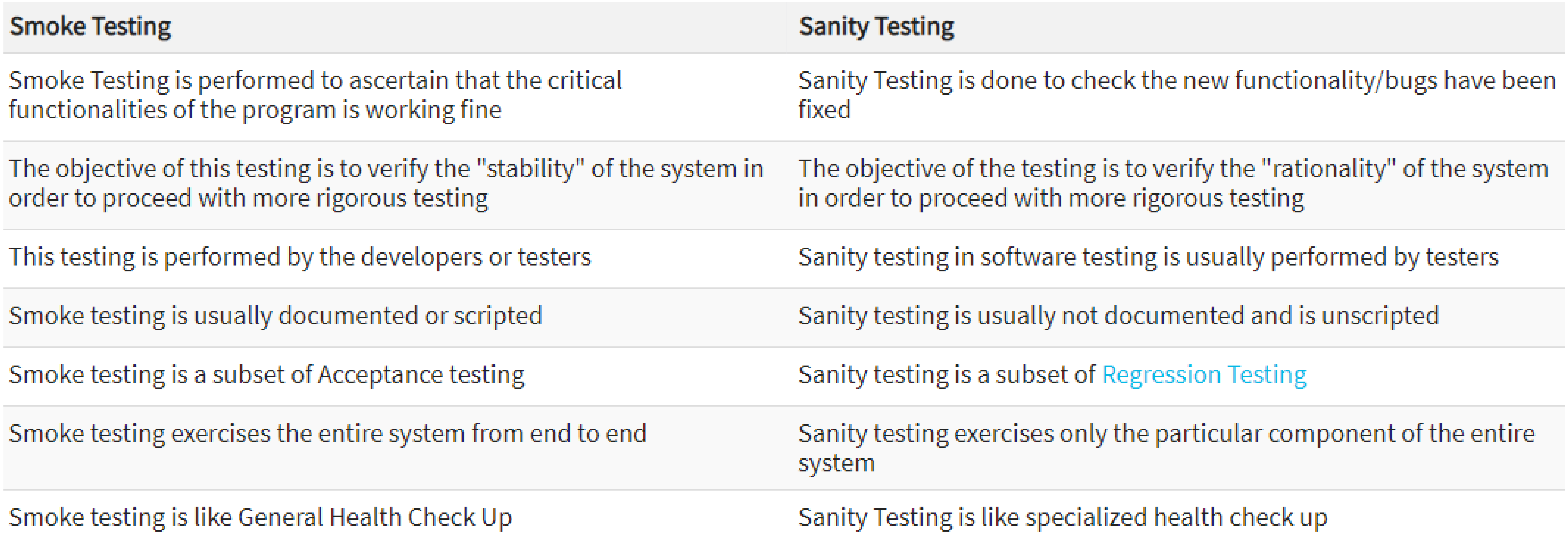
Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

The objective is "not" to verify thoroughly the new functionality but to determine that the developer has applied some rationality (sanity) while producing the software. For instance, if your scientific calculator gives the result of 2 + 2 =5! Then, there is no point testing the advanced functionalities like sin 30 + cos 50.



Mobile Application Testing

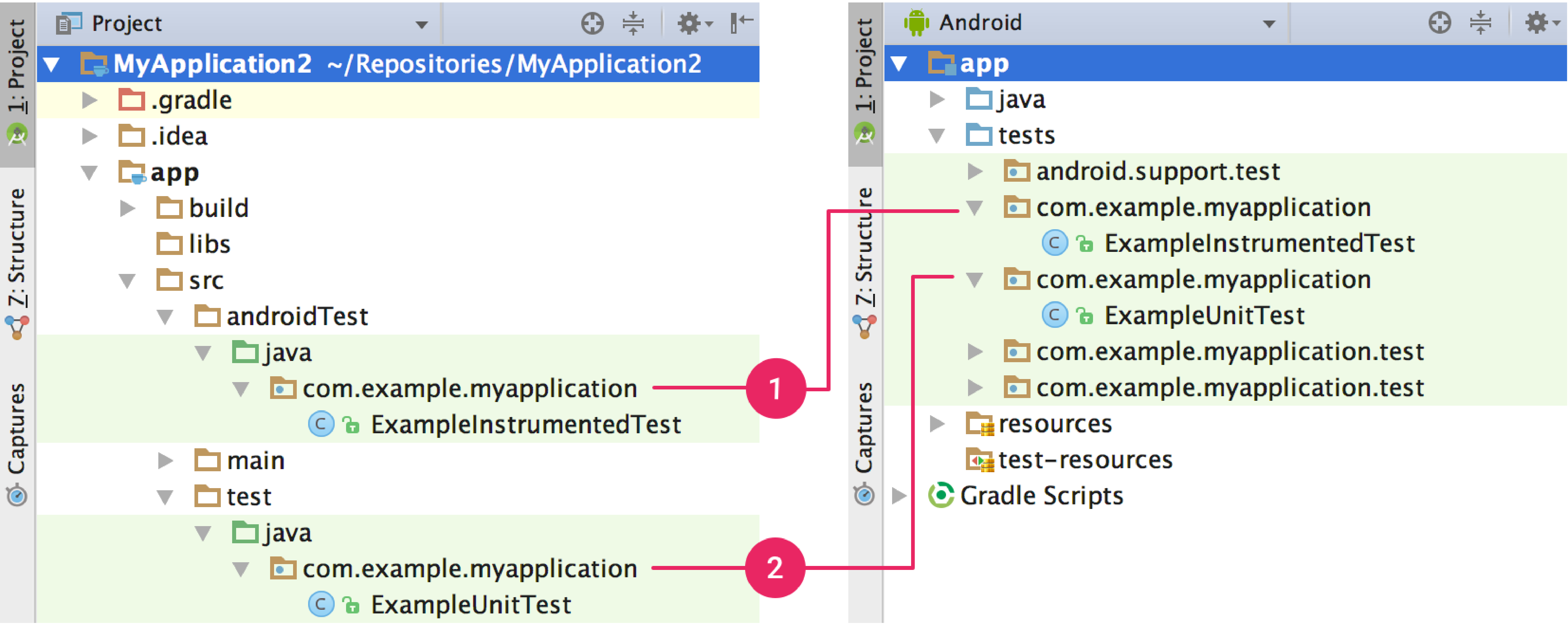
Smoke Testing Vs Sanity Testing - Key Differences



Smoke Testing	Sanity Testing
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine	Sanity Testing is done to check the new functionality/bugs have been fixed
The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing	The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing
This testing is performed by the developers or testers	Sanity testing in software testing is usually performed by testers
Smoke testing is usually documented or scripted	Sanity testing is usually not documented and is unscripted
Smoke testing is a subset of Acceptance testing	Sanity testing is a subset of Regression Testing
Smoke testing exercises the entire system from end to end	Sanity testing exercises only the particular component of the entire system
Smoke testing is like General Health Check Up	Sanity Testing is like specialized health check up

https://www.guru99.com/smoke-sanity-testing.html

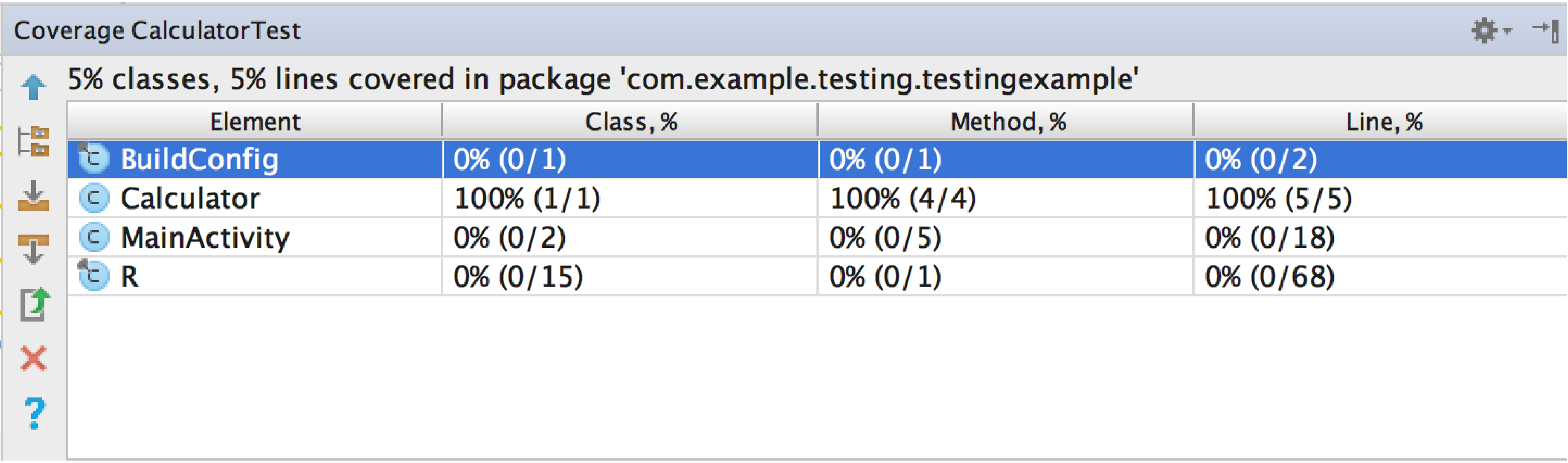
Mobile Application Testing



Android project's **(1)** instrumented tests and **(2)** local JVM tests are visible in either the **Project** view (left) or **Android** view (right).

https://developer.android.com/studio/test

Mobile Application Testing



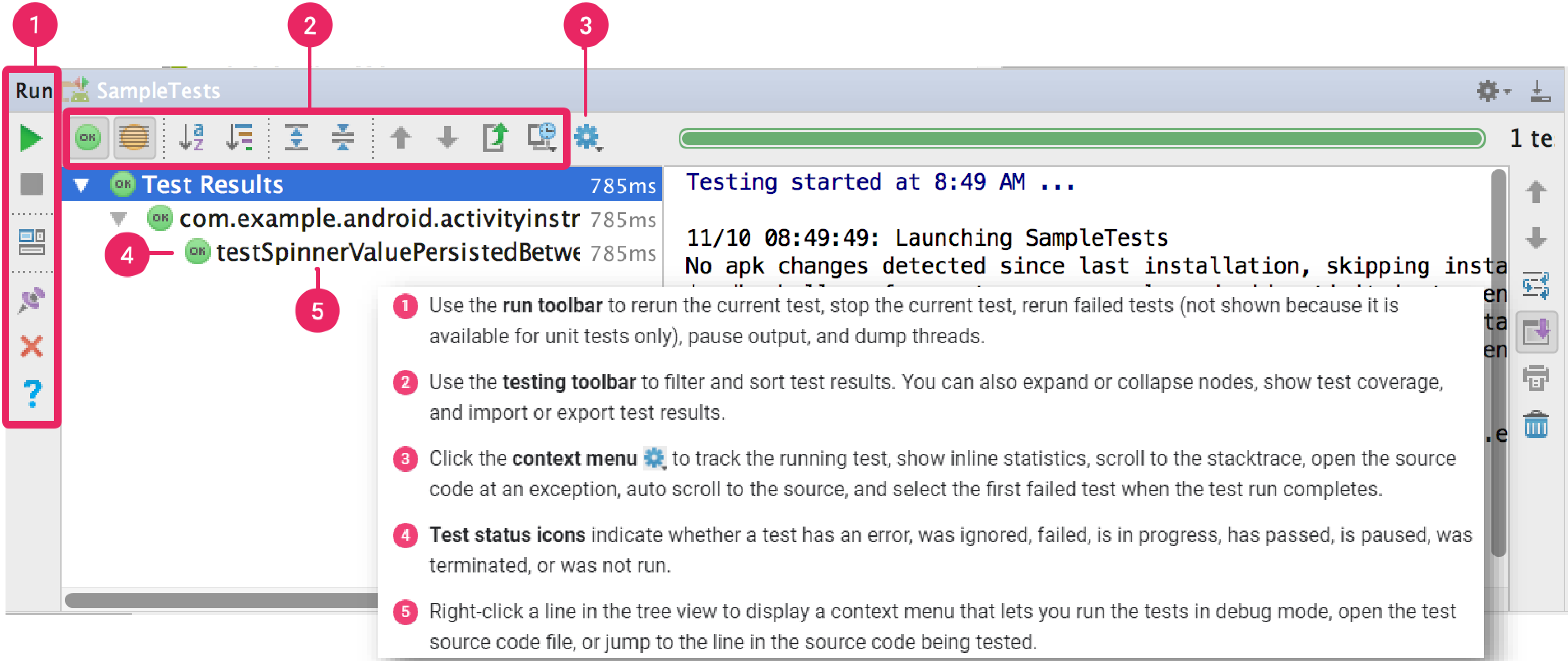
Coverage CalculatorTest

5% classes, 5% lines covered in package 'com.example.testing.testingexample'

Element	Class, %	Method, %	Line, %
BuildConfig	0% (0/1)	0% (0/1)	0% (0/2)
Calculator	100% (1/1)	100% (4/4)	100% (5/5)
MainActivity	0% (0/2)	0% (0/5)	0% (0/18)
R	0% (0/15)	0% (0/1)	0% (0/68)

See the code coverage percentages for your application.

Mobile Application Testing



```
Test Results                                785ms
  com.example.android.activityinstr        785ms
    testSpinnerValuePersistedBetwe         785ms
```

```
Testing started at 8:49 AM ...

11/10 08:49:49: Launching SampleTests
No apk changes detected since last installation, skipping insta
```

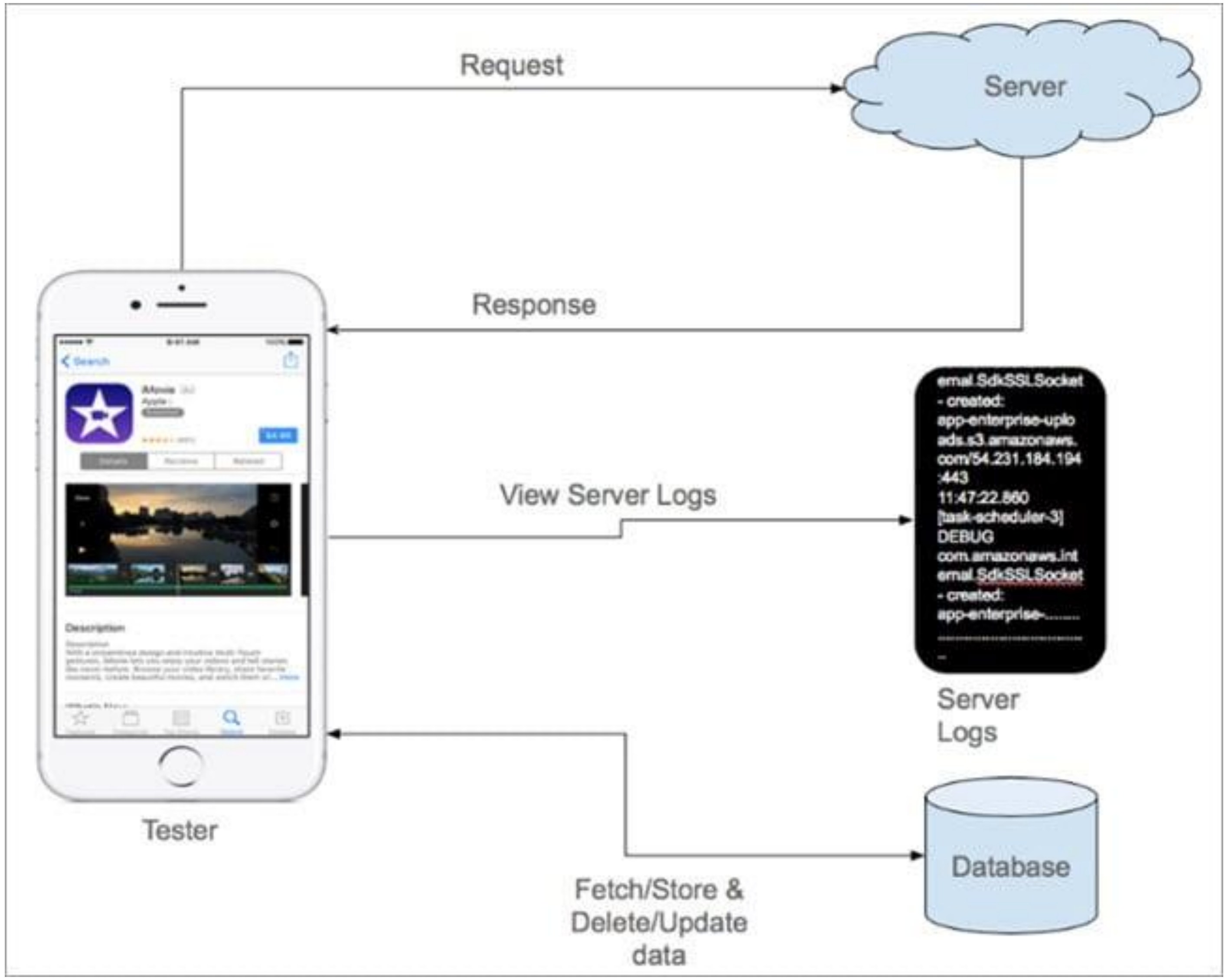
1. Use the **run toolbar** to rerun the current test, stop the current test, rerun failed tests (not shown because it is available for unit tests only), pause output, and dump threads.
2. Use the **testing toolbar** to filter and sort test results. You can also expand or collapse nodes, show test coverage, and import or export test results.
3. Click the **context menu** to track the running test, show inline statistics, scroll to the stacktrace, open the source code at an exception, auto scroll to the source, and select the first failed test when the test run completes.
4. **Test status icons** indicate whether a test has an error, was ignored, failed, is in progress, has passed, is paused, was terminated, or was not run.
5. Right-click a line in the tree view to display a context menu that lets you run the tests in debug mode, open the test source code file, or jump to the line in the source code being tested.

Test results appear in the Run window.

https://developer.android.com/studio/test

Mobile Application Testing

Testing should not just be limited to exploring the mobile app and logging bugs. We should be aware of all the request that we hit our server and the response that we get out of it. Configure to view logs not only helps in knowing the End-to-End flow of the application but also get more ideas and scenarios now.



```
emal.SdkSSLSocket
- created:
app-enterprise-uplo
ads.s3.amazonaws.
com/54.231.184.194
:443
11:47:22.860
[task-scheduler-3]
DEBUG
com.amazonaws.int
emal.SdkSSLSocket
- created:
app-enterprise-........
--------------------------------
-
```

Nothing comes into this world without any reason. Any statement should have a valid reason behind it. The reason behind analyzing the logs is that many exceptions are observed in the logs, but they don't show any impact on the UI hence we don't notice it.

****

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !

Lập trình ứng dụng di động

Mobile Application Programming

ET4710

PGS. TS. Đỗ Trọng Tuấn

*Viện Điện tử Viễn thông * Đại học Bách Khoa Hà Nội*

ONE LOVE. ONE FUTURE.