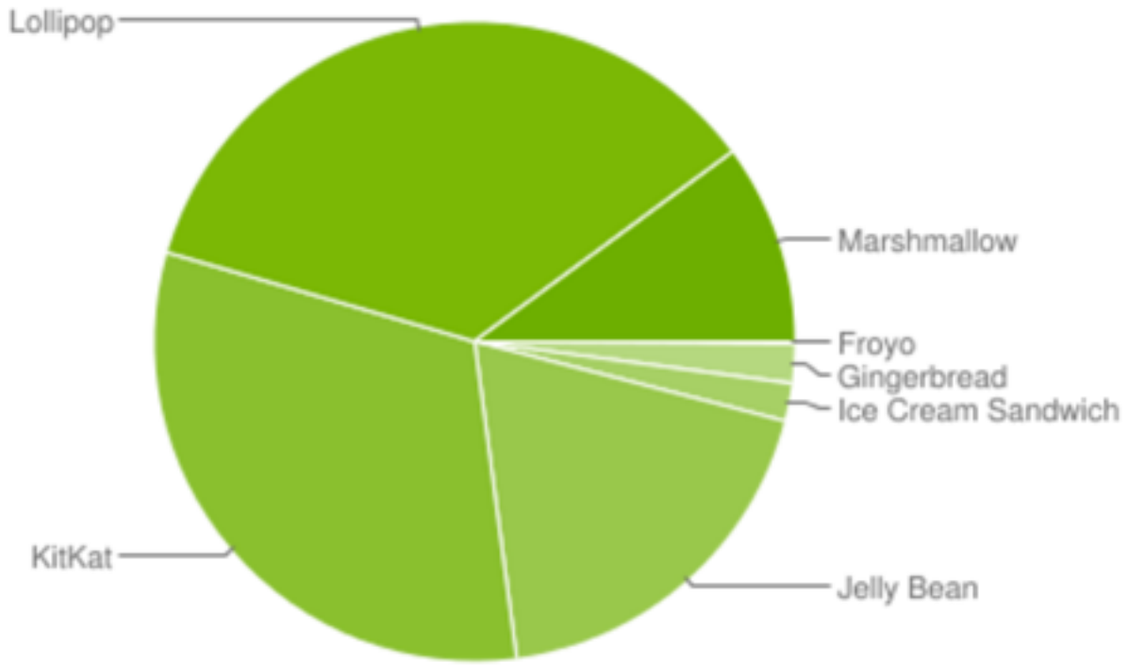
# Espresso workshop

somkiat.cc

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

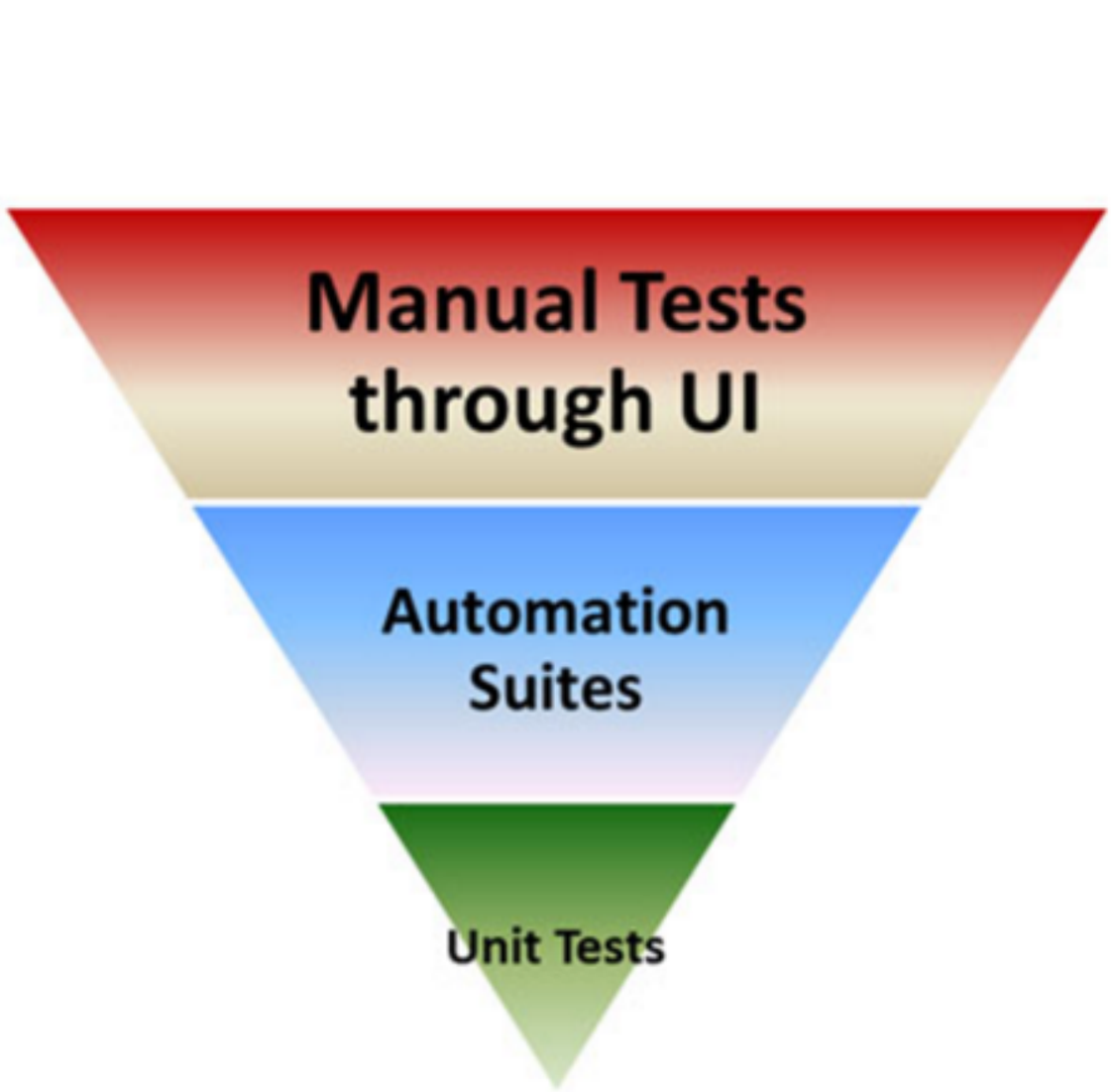
# ทดสอบไปทำไม ?

| Version | Codename | API | Distribution |
|---|---|---|---|
| 2.2 | Froyo | 8 | 0.1% |
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 2.0% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 1.9% |
| 4.1.x | Jelly Bean | 16 | 6.8% |
| 4.2.x | | 17 | 9.4% |
| 4.3 | | 18 | 2.7% |
| 4.4 | KitKat | 19 | 31.6% |
| 5.0 | Lollipop | 21 | 15.4% |
| 5.1 | | 22 | 20.0% |
| 6.0 | Marshmallow | 23 | 10.1% |

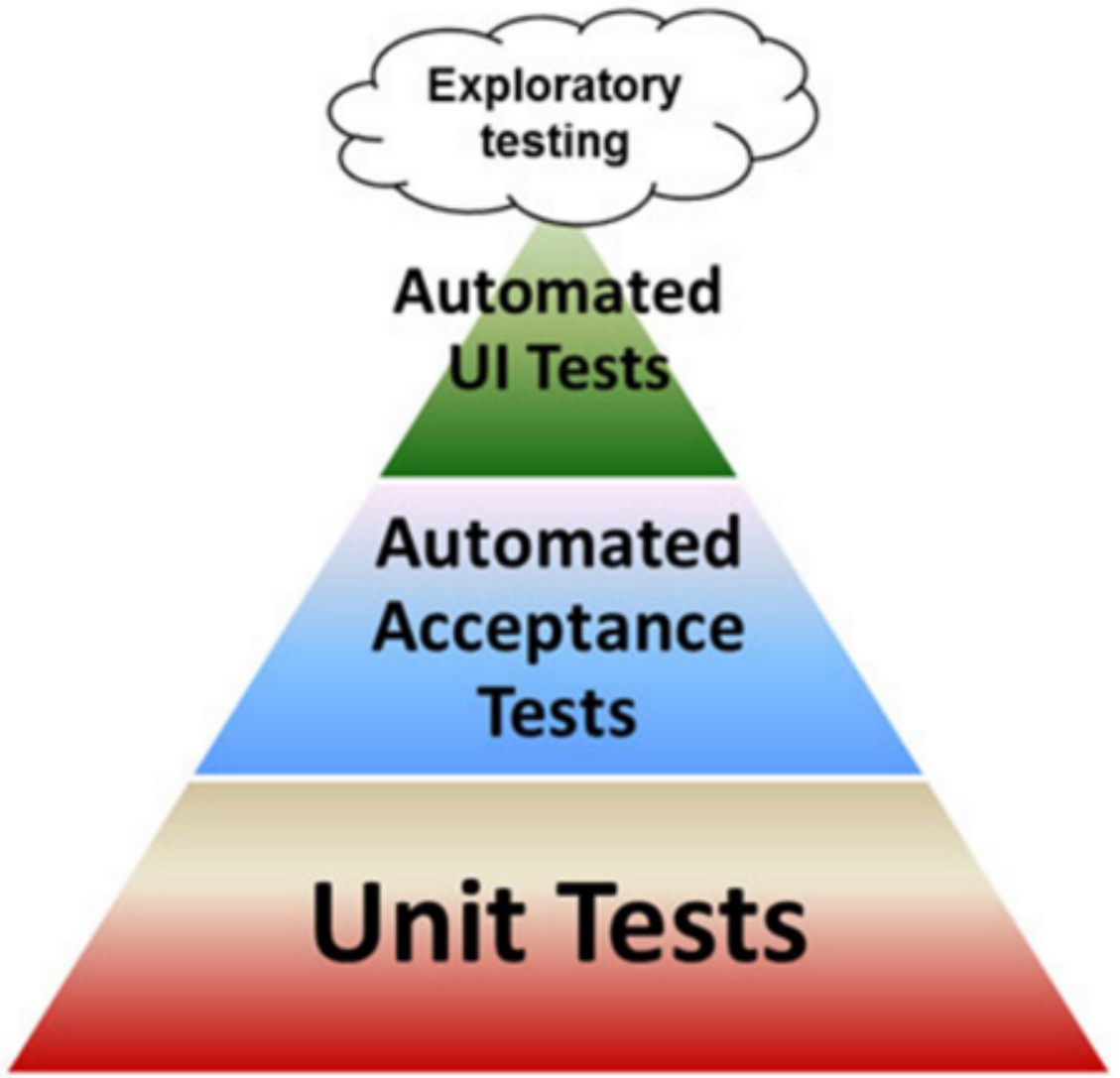



*Data collected during a 7-day period ending on June 6, 2016.*

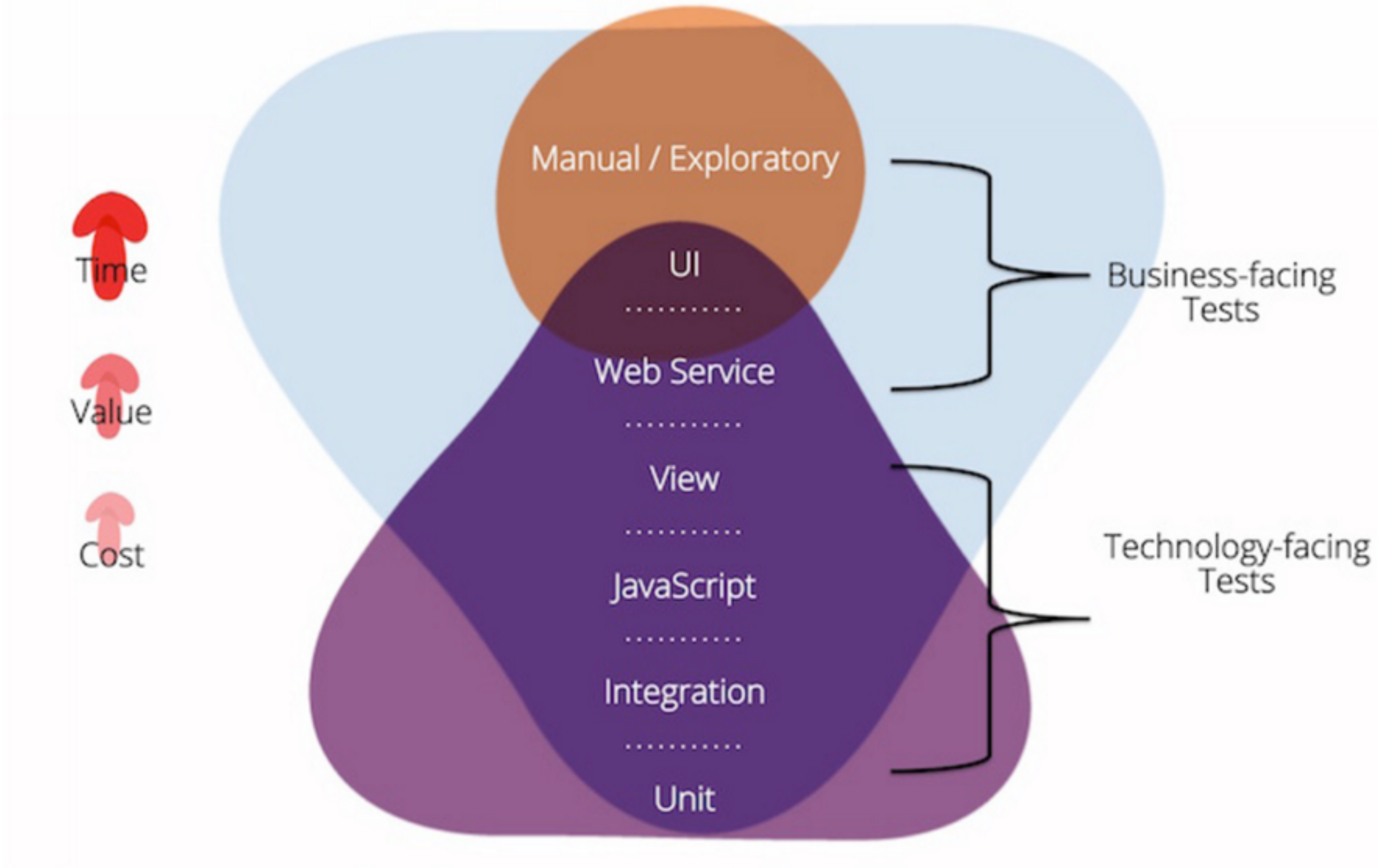
*Any versions with less than 0.1% distribution are not shown.*




Traditional
(find bugs)



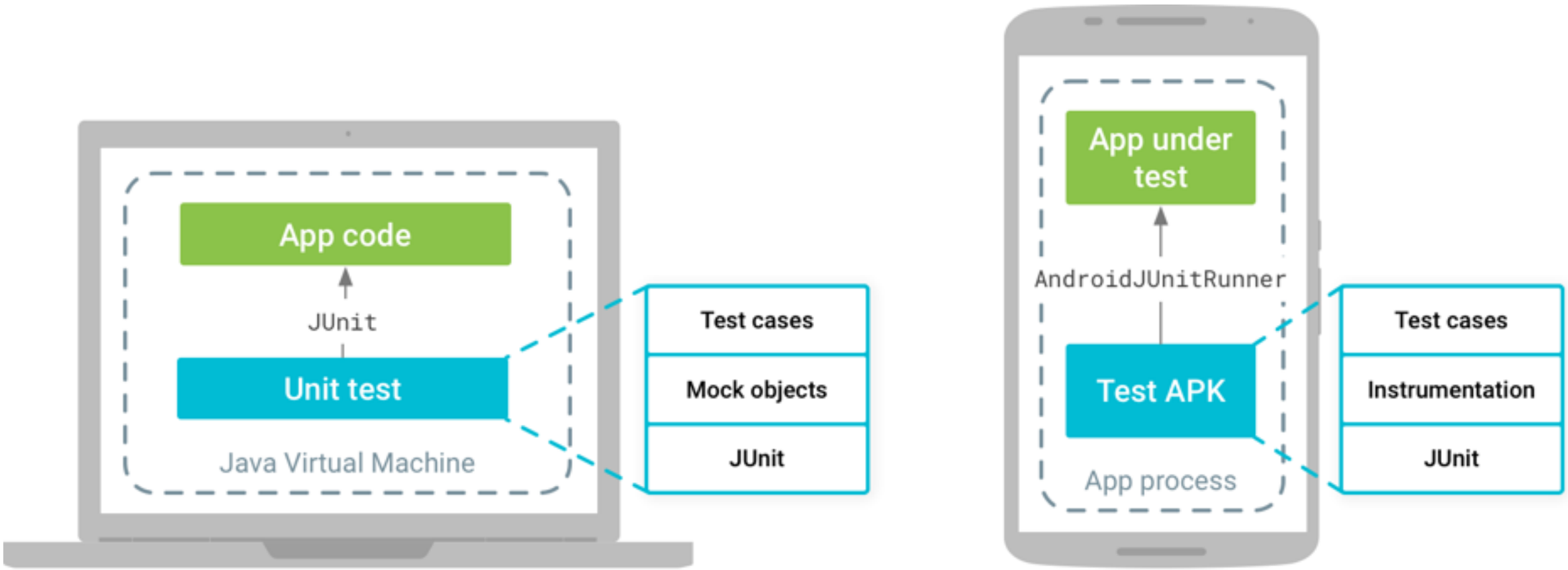

Agile
(prevent bugs)


Time
Value
Cost
Manual / Exploratory
UI
Web Service
View
JavaScript
Integration
Unit
Business-facing Tests
Technology-facing Tests

# การทดสอบสำหรับ Android

1. Unit testing
2. UI testing

# การทดสอบสำหรับ Android




Local unit test
src/test/java/

Instrumented test
src/androidTest/java/

https://developer.android.com/training/testing/start/index.html

# Unit testing

Unit

Mockito

PowerMock

# UI testing

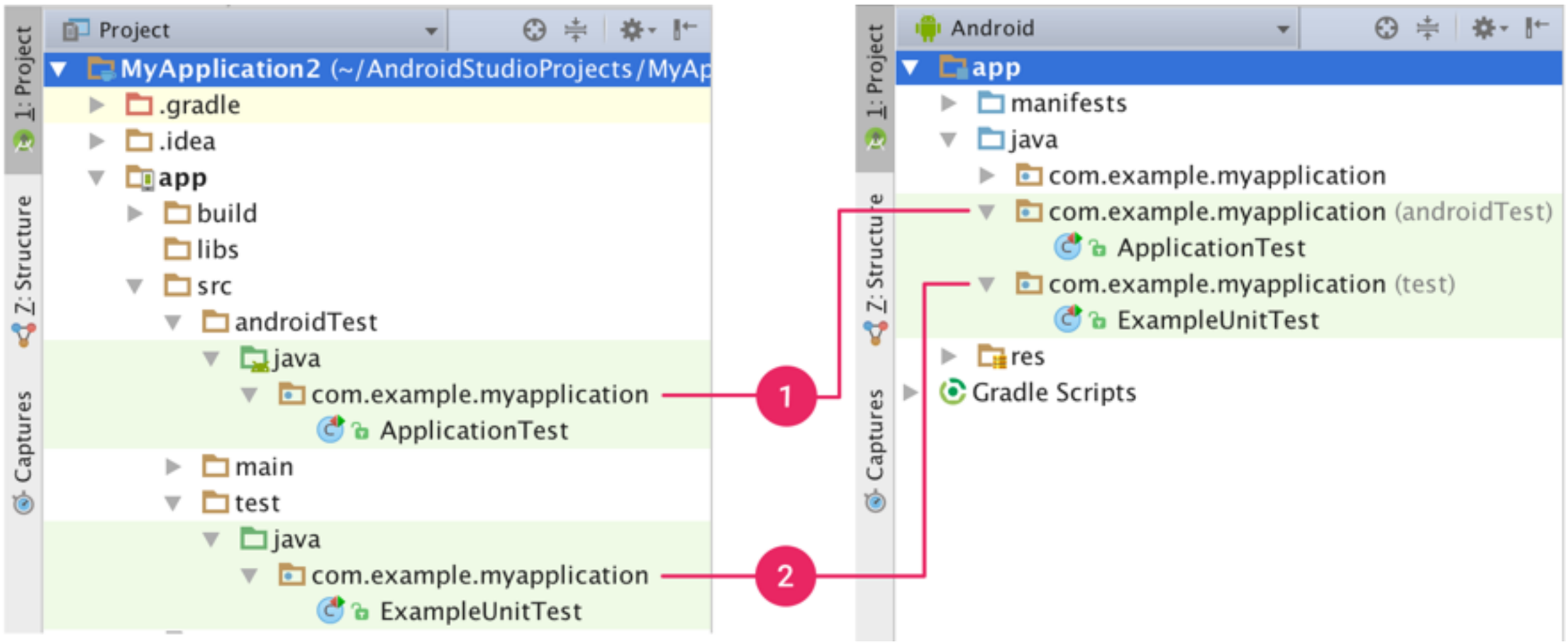
Instrumentation test

**Espresso**

UI Automator

Calabash, Appium, Robotium, Robolectrie

https://google.github.io/android-testing-support-library/




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

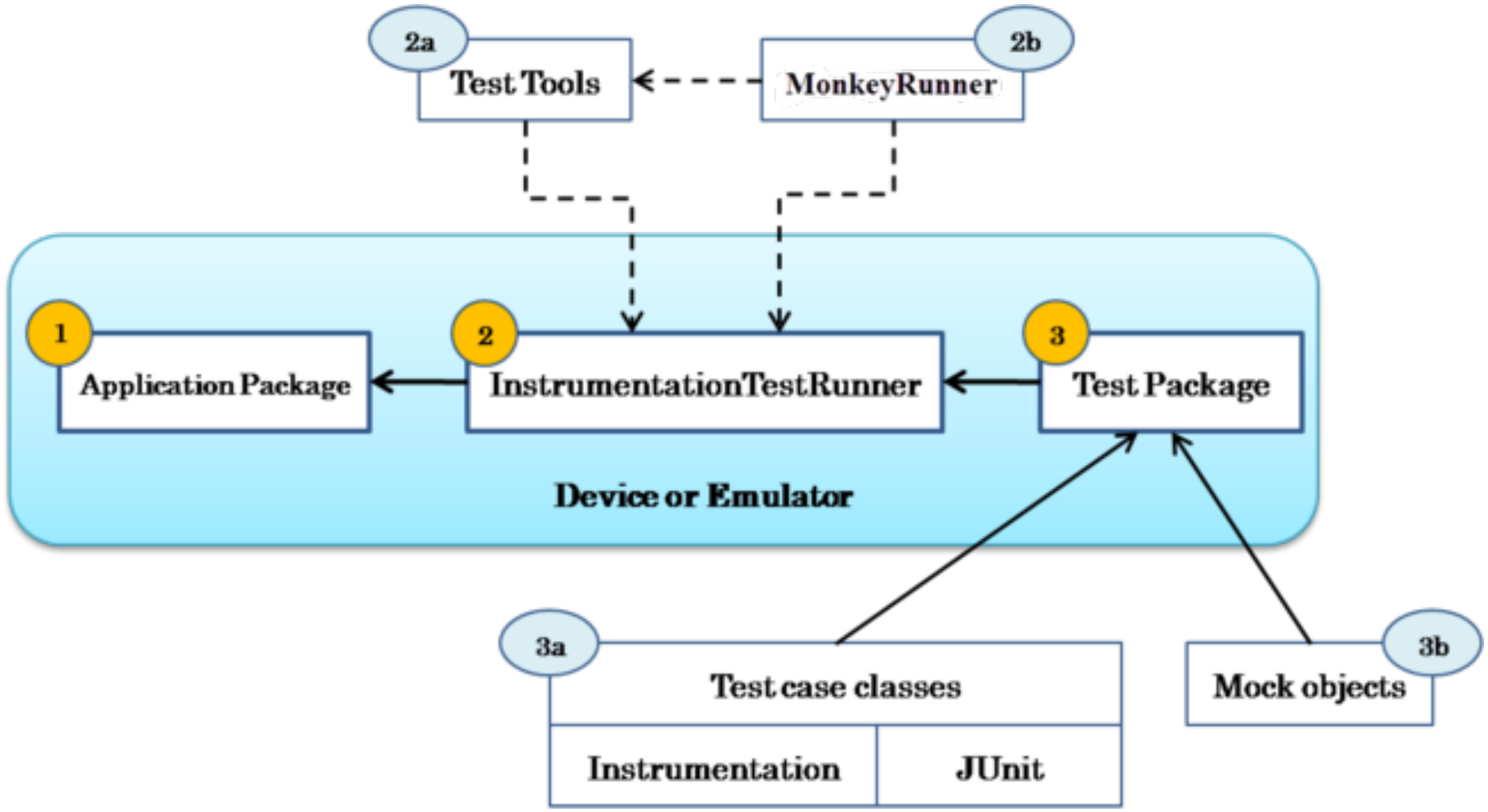
# Build High-Quality App

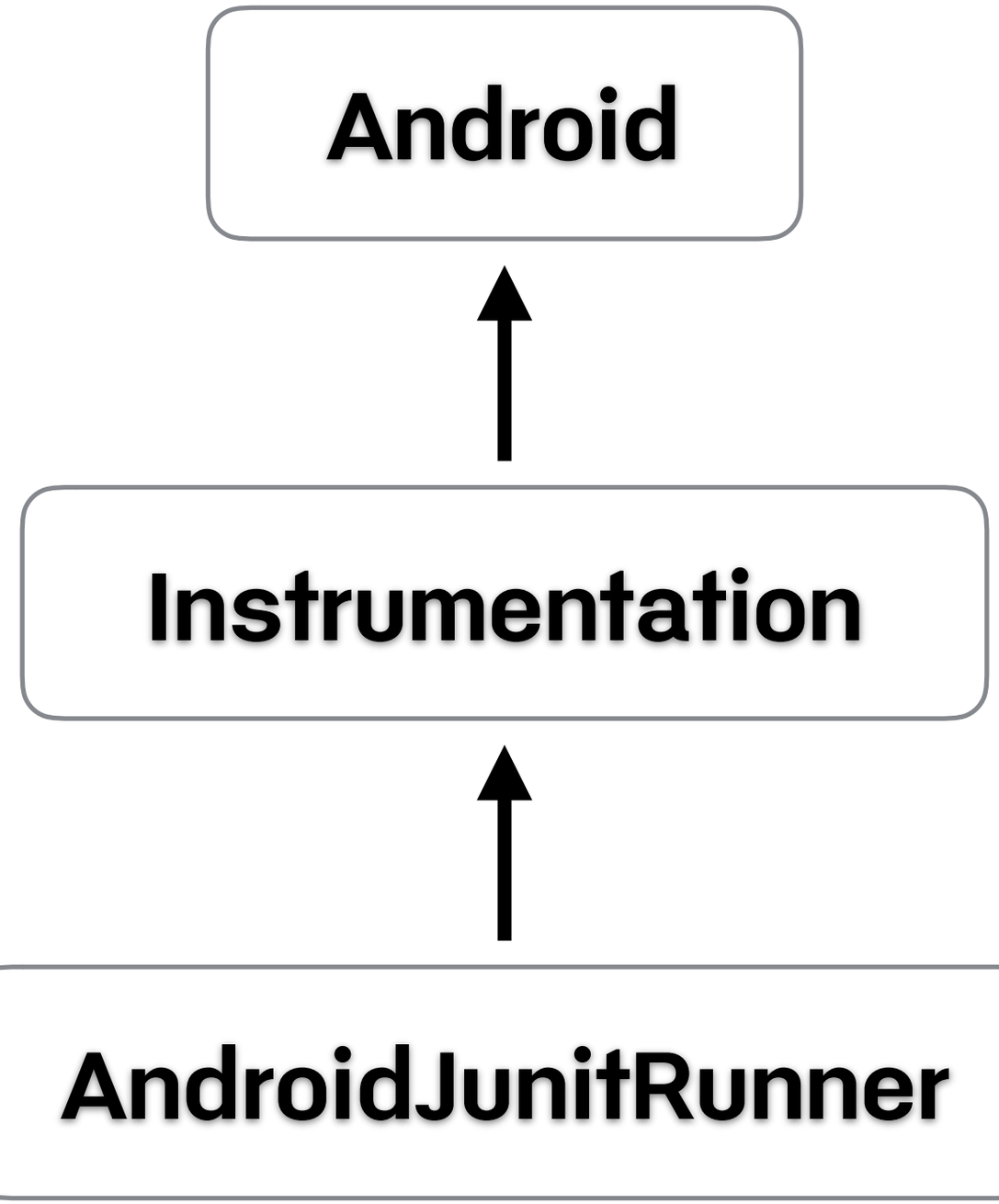
# Espresso คืออะไร ?

Automated UI testing

Simulate user interaction

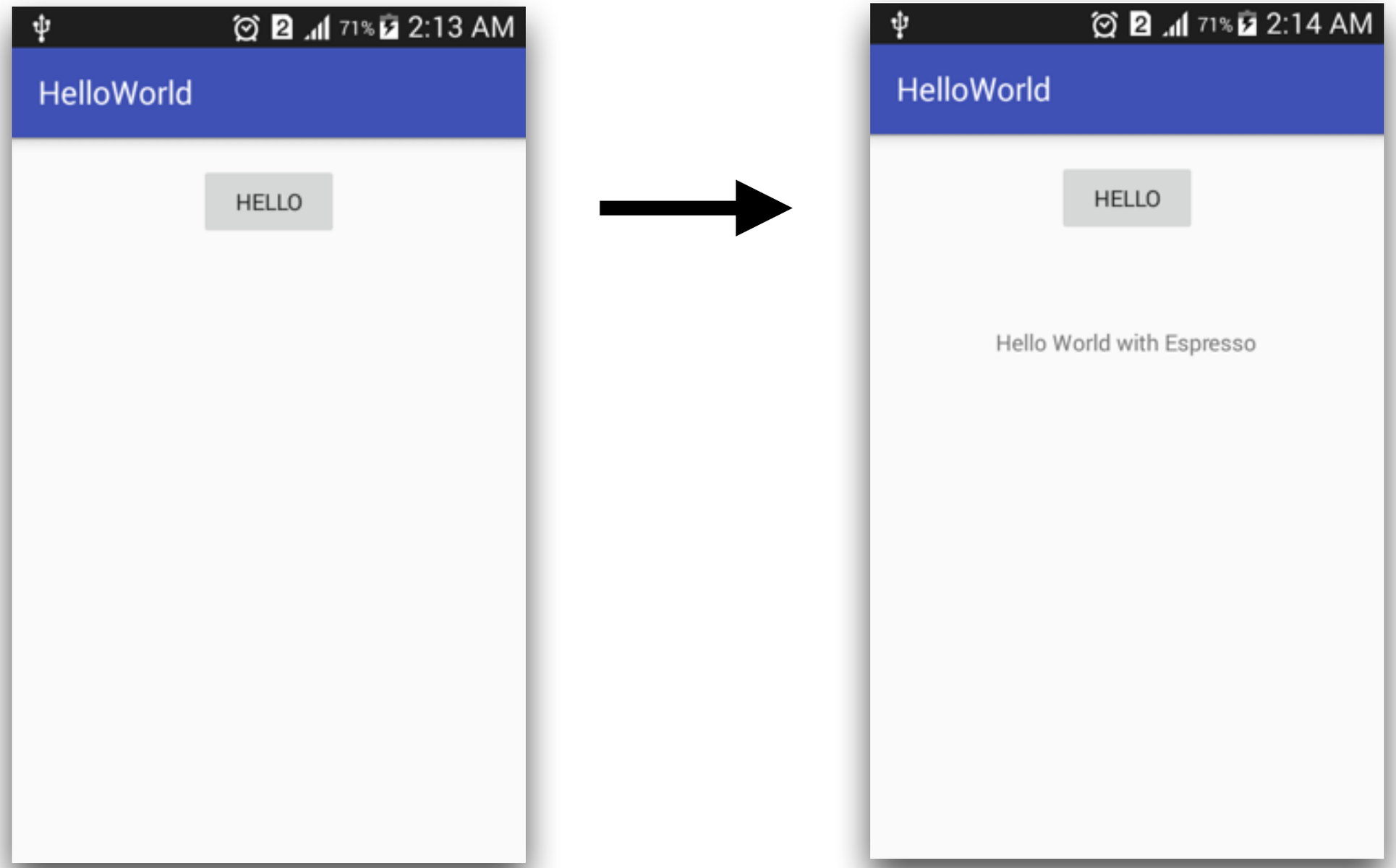
Synchronization of test actions with UI

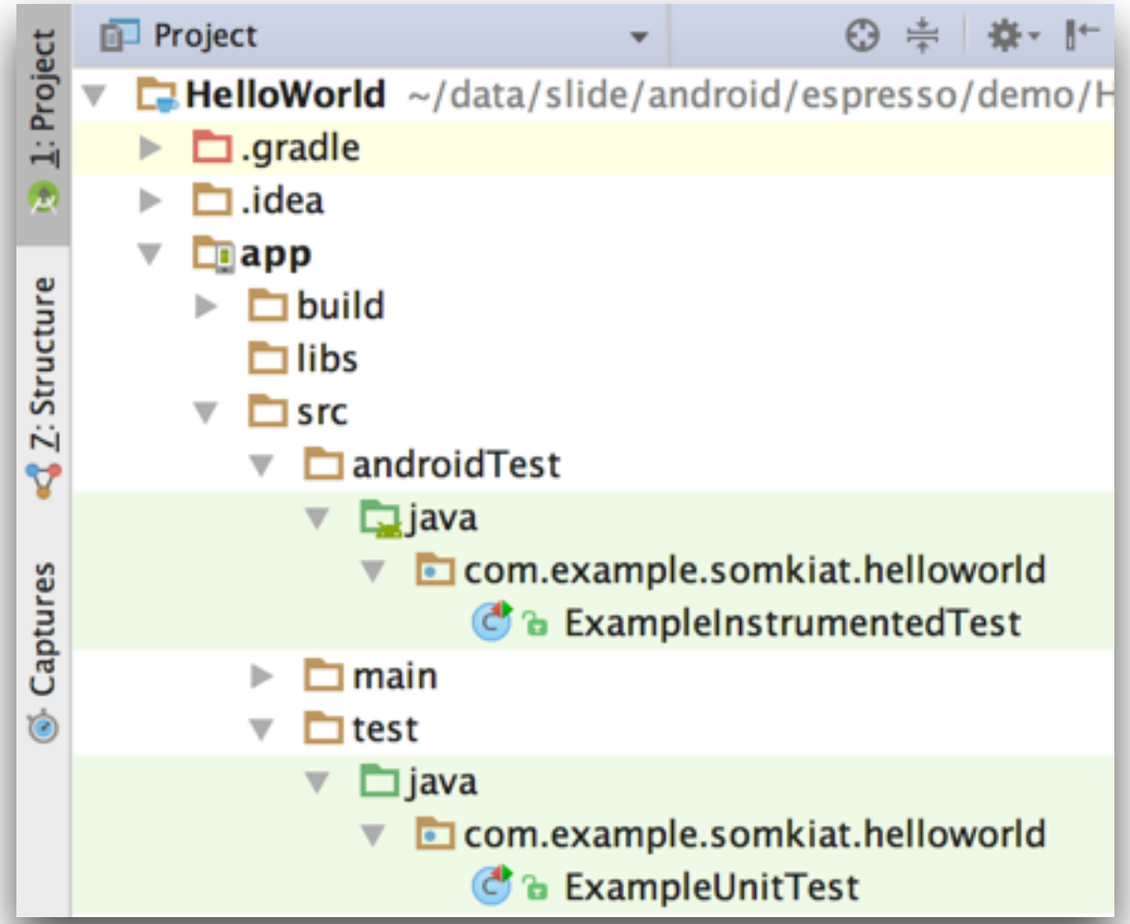

2a
Test Tools
2b
MonkeyRunner
1
Application Package
2
InstrumentationTestRunner
3
Test Package
Device or Emulator
3a
Test case classes
Instrumentation
JUnit
3b
Mock objects

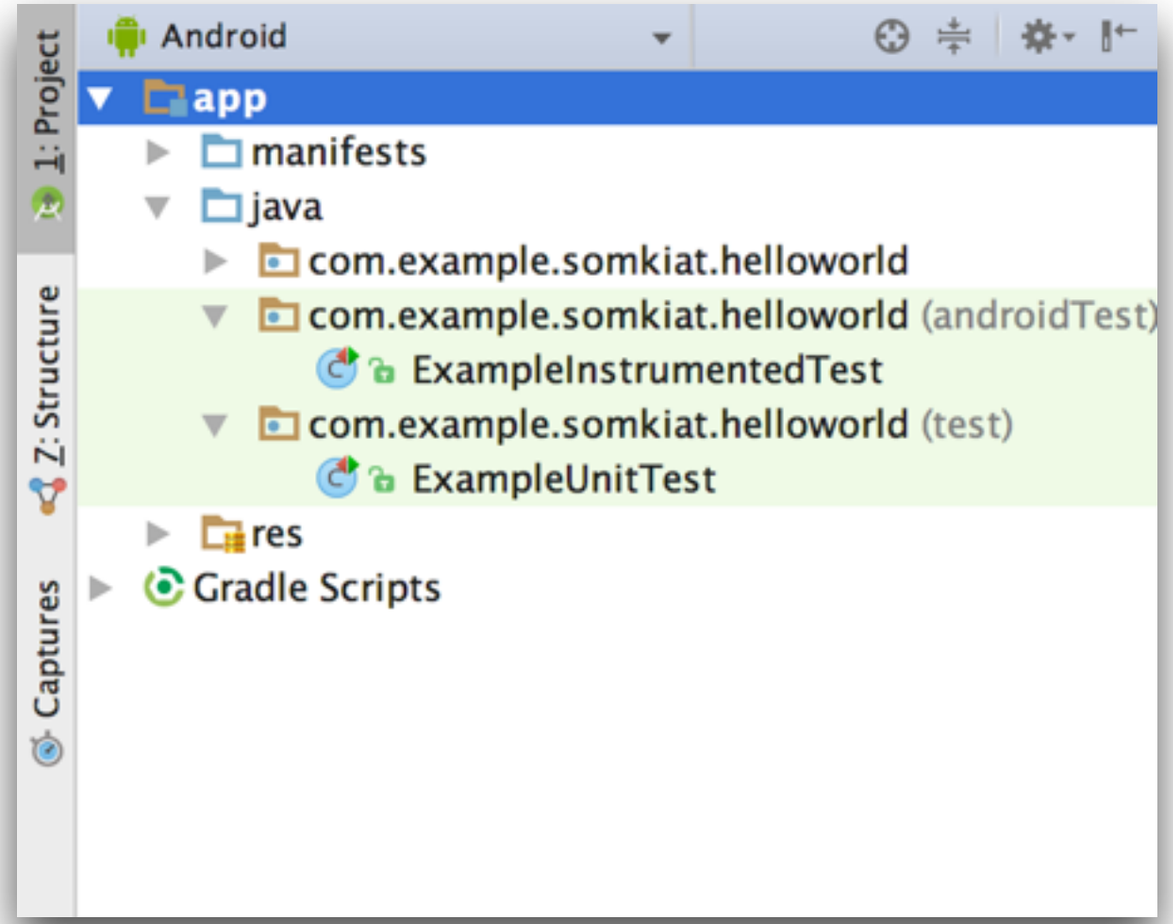

Android
Instrumentation
AndroidJunitRunner

# Hello World

# Hello World


2:13 AM
HelloWorld
HELLO
2:14 AM
HelloWorld
HELLO
Hello World with Espresso


Project
HelloWorld ~/data/slide/android/espresso/demo/H
.gradle
.idea
app
build
libs
src
androidTest
java
com.example.somkiat.helloworld
ExampleInstrumentedTest
main
test
java
com.example.somkiat.helloworld
ExampleUnitTest
1: Project
Z: Structure
Captures


Android
app
manifests
java
com.example.somkiat.helloworld
com.example.somkiat.helloworld (androidTest)
ExampleInstrumentedTest
com.example.somkiat.helloworld (test)
ExampleUnitTest
res
Gradle Scripts
1: Project
Z: Structure
Captures

# Configuration

```
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"
    defaultConfig {
        applicationId "com.example.somkiat.helloworld"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
```

# Configuration

```
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"
    defaultConfig {
        applicationId "com.example.somkiat.helloworld"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
```

Default in Android Studio 2.2

# Dependencies

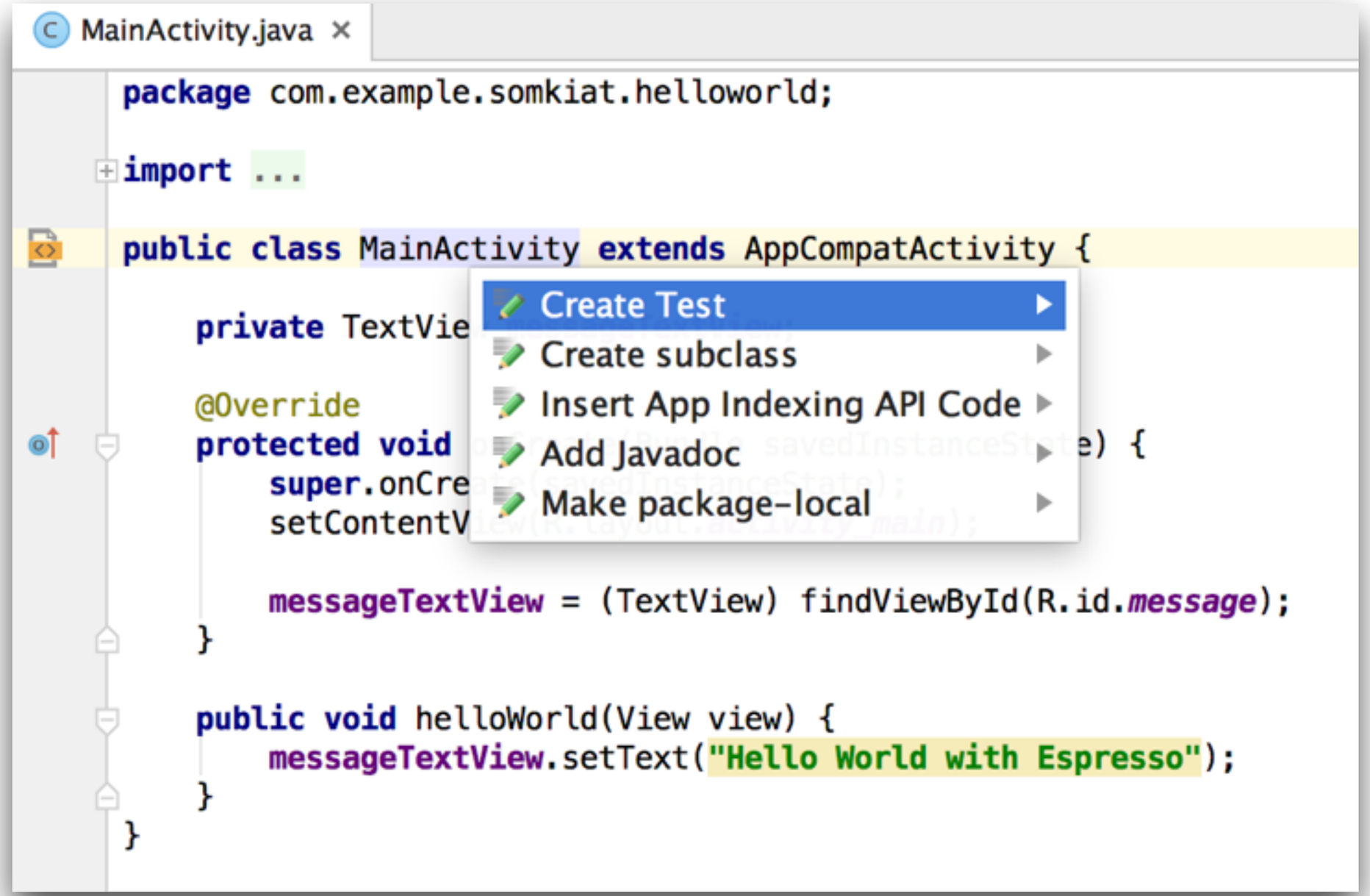
```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:23.4.0'
    compile 'com.android.support.constraint:constraint-layout:1.0.0-alpha3'
    testCompile 'junit:junit:4.12'
}
```

# Dependencies

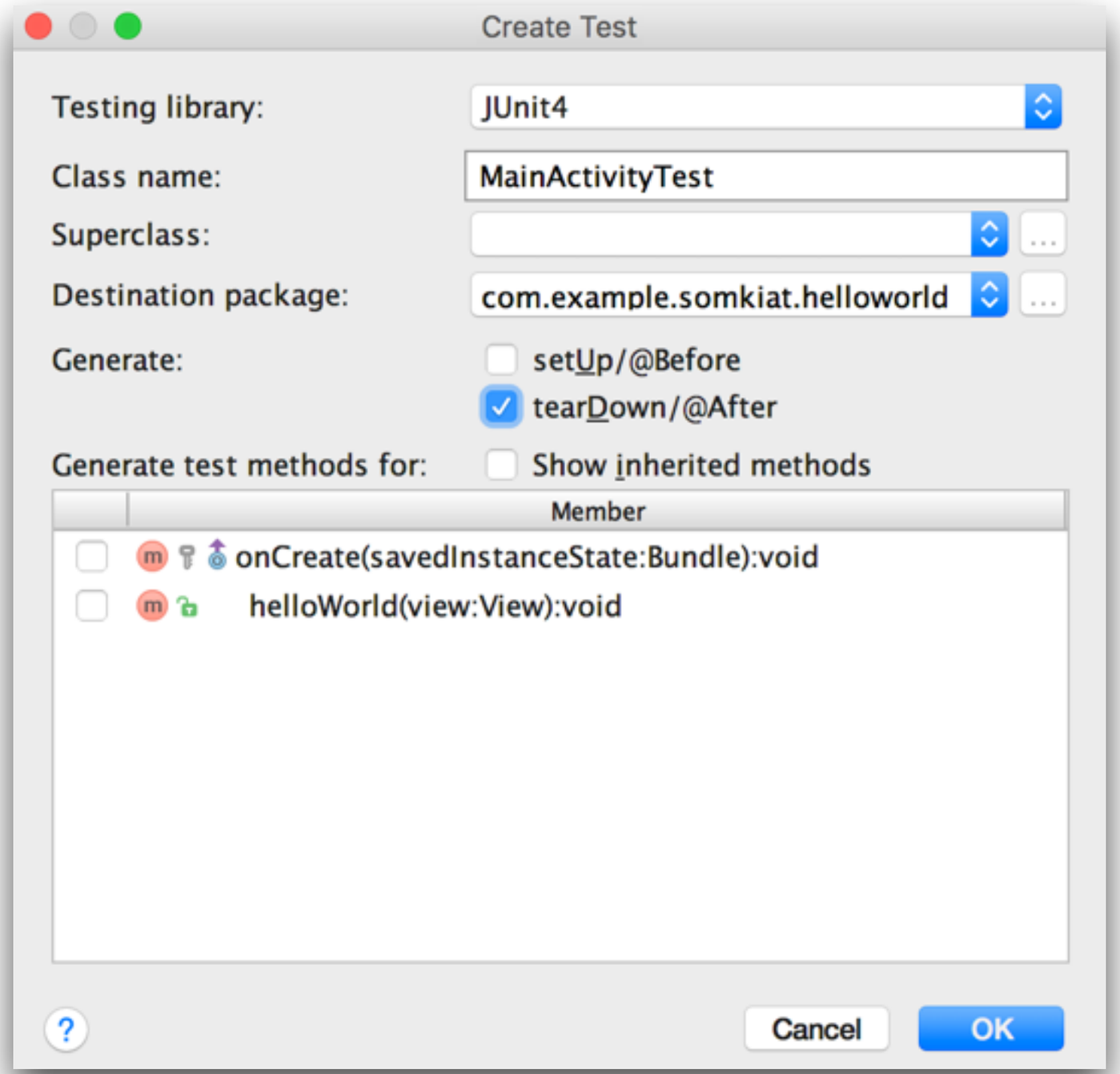
```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:23.4.0'
    compile 'com.android.support.constraint:constraint-layout:1.0.0-alpha3'
    testCompile 'junit:junit:4.12'
}
```

Default in Android Studio 2.2

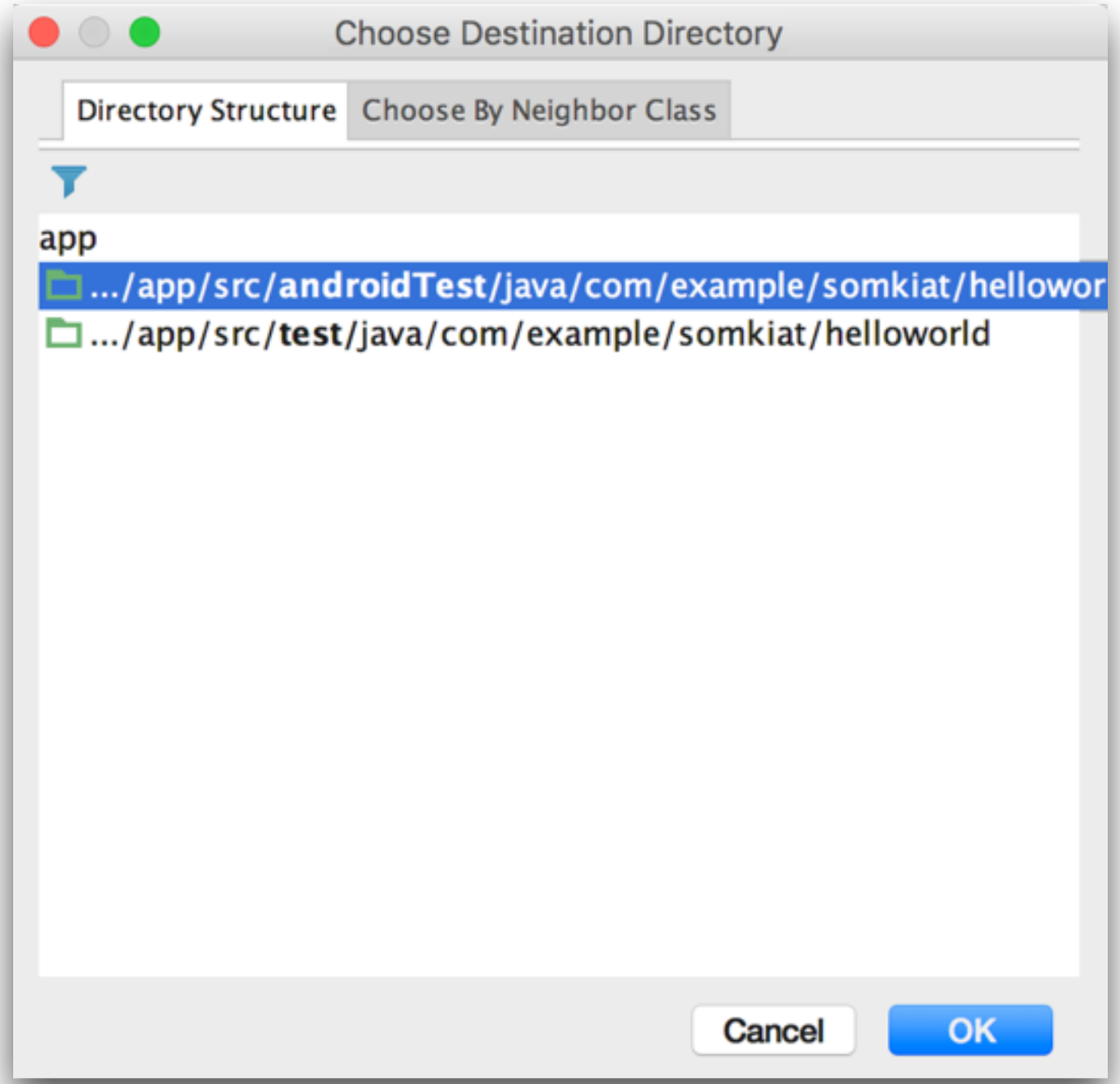
# Write first test



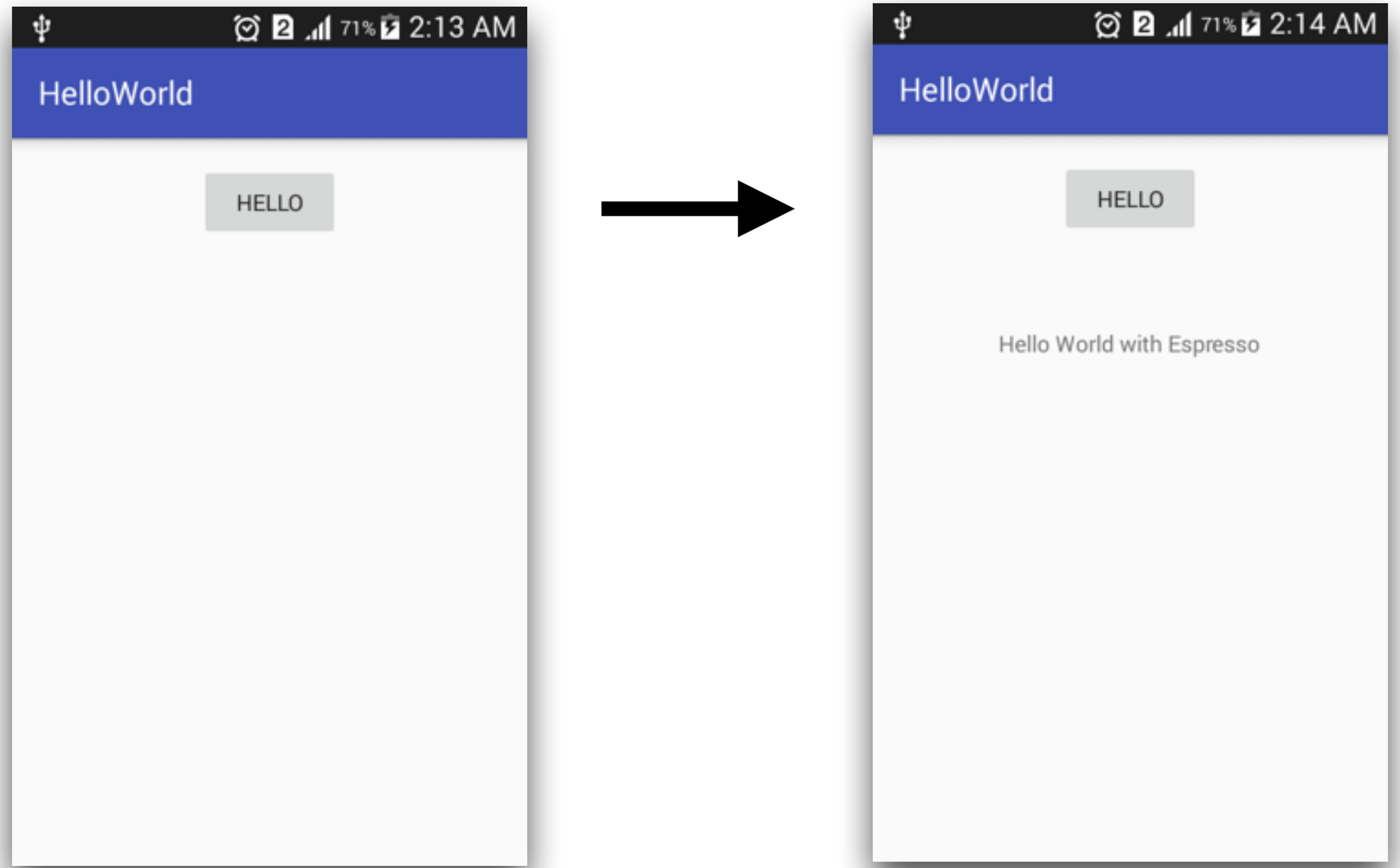
# Write first test



# Write first test



# Write first test



# Step 1 :: Find Hello Button

```
public class MainActivityTest {

    @Rule
    public ActivityTestRule activityTestRule = new ActivityTestRule(MainActivity.class);

    @Test
    public void findHelloButtonInView() {
        onView(withId(R.id.hello_button)).check(matches(isDisplayed()));
    }

}
```

# Step 2 :: Click Hello Button

```
public class MainActivityTest {


    @Rule
    public ActivityTestRule activityTestRule = new ActivityTestRule(MainActivity.class);


    @Test
    public void findHelloButtonInView() {
        onView(withId(R.id.hello_button)).check(matches(isDisplayed()));
        onView(withId(R.id.hello_button)).perform(click());
    }

}
```

# Step 2 :: Click Hello Button

```
public class MainActivityTest {


    @Rule
    public ActivityTestRule activityTestRule = new ActivityTestRule(MainActivity.class);


    @Test
    public void findHelloButtonInView() {
        onView(withId(R.id.hello_button)).check(matches(isDisplayed()));
        onView(withId(R.id.hello_button)).perform(click());
    }

}
```

# Step 3 :: Check result

```
public class MainActivityTest {


    @Rule
    public ActivityTestRule activityTestRule = new ActivityTestRule(MainActivity.class);


    @Test
    public void findHelloButtonInView() {
        onView(withId(R.id.hello_button)).check(matches(isDisplayed()));
        onView(withId(R.id.hello_button)).perform(click());
        onView(withId(R.id.message))
                .check(matches(withText("Hello World with Espresso")));
    }

}
```

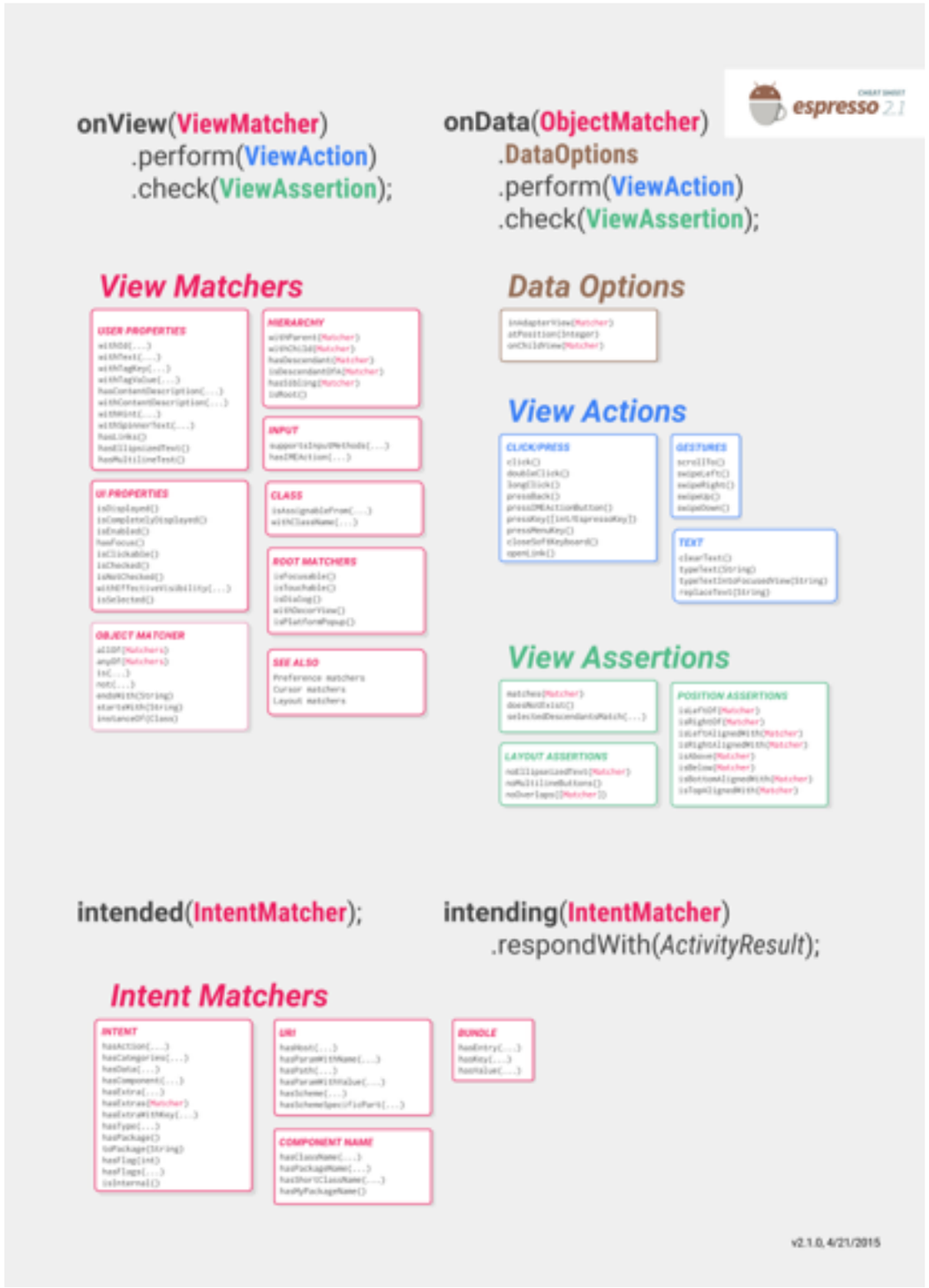
# Espresso APIs

View Matcher

View Action

View Assertion

# Espresso cheat sheet

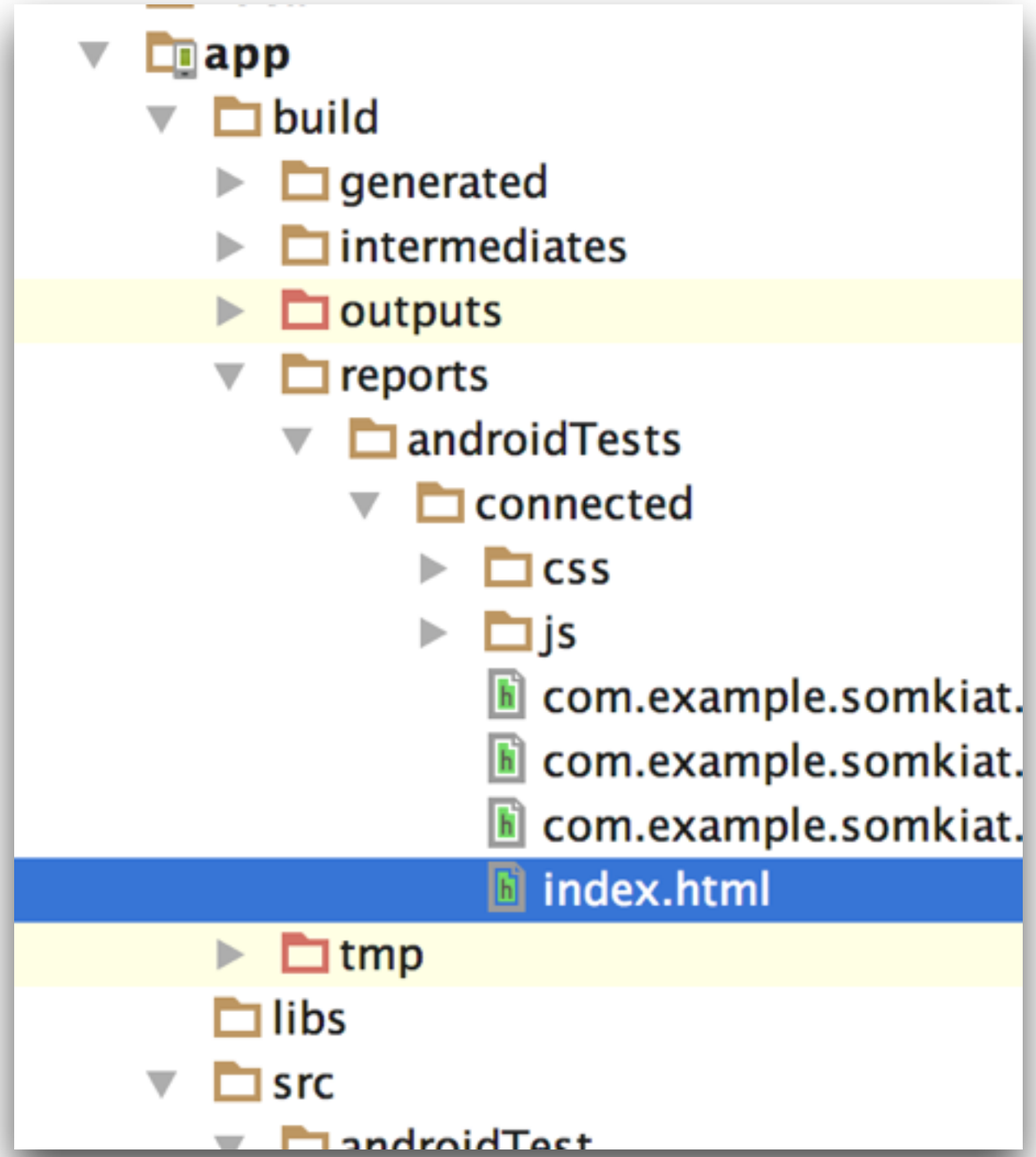


# Run test

$./gradlew clean connectedAndroidTest

$./gradlew clean cAT

# Test report


app
build
generated
intermediates
outputs
reports
androidTests
connected
css
js
com.example.somkiat.
com.example.somkiat.
com.example.somkiat.
index.html
tmp
libs
src

# Test report

## Test Summary

| 2 | 0 | 0.933s |
|---|---|---|
| tests | failures | duration |

**100%**
successful

**Packages** Classes

| Package | Tests | Failures | Duration |
|---|---|---|---|
| com.example.somkiat.helloworld | 2 | 0 | 0.933s |

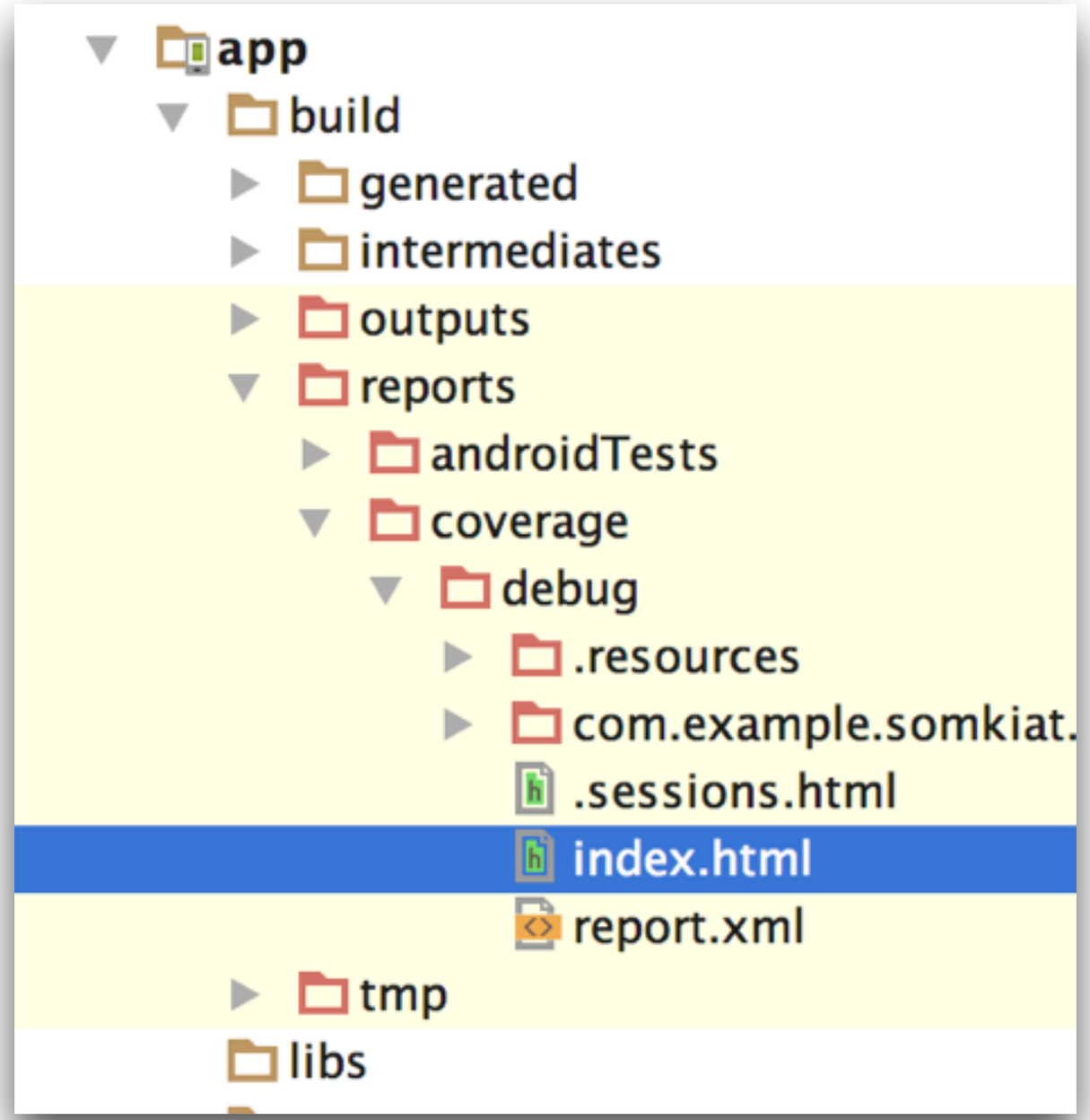
# Enable code coverage

```
buildTypes {
    debug {
        testCoverageEnabled = true
    }
```

# Run with code coverage

$./gradlew createDebugCoverageReport

# Code coverage report



# Code coverage report

debugAndroidTest > com.example.somkiat.helloworld

## com.example.somkiat.helloworld

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | M |
|---|---|---|---|---|---|---|---|
| SecondActivity | | 0% | | n/a | 2 | 2 | |
| MainActivity | | 100% | | n/a | 0 | 3 | |
| Total | 10 of 31 | 68% | 0 of 0 | n/a | 2 | 5 | |

Generated by the Android Gradle plugin 2.2.0-alpha4

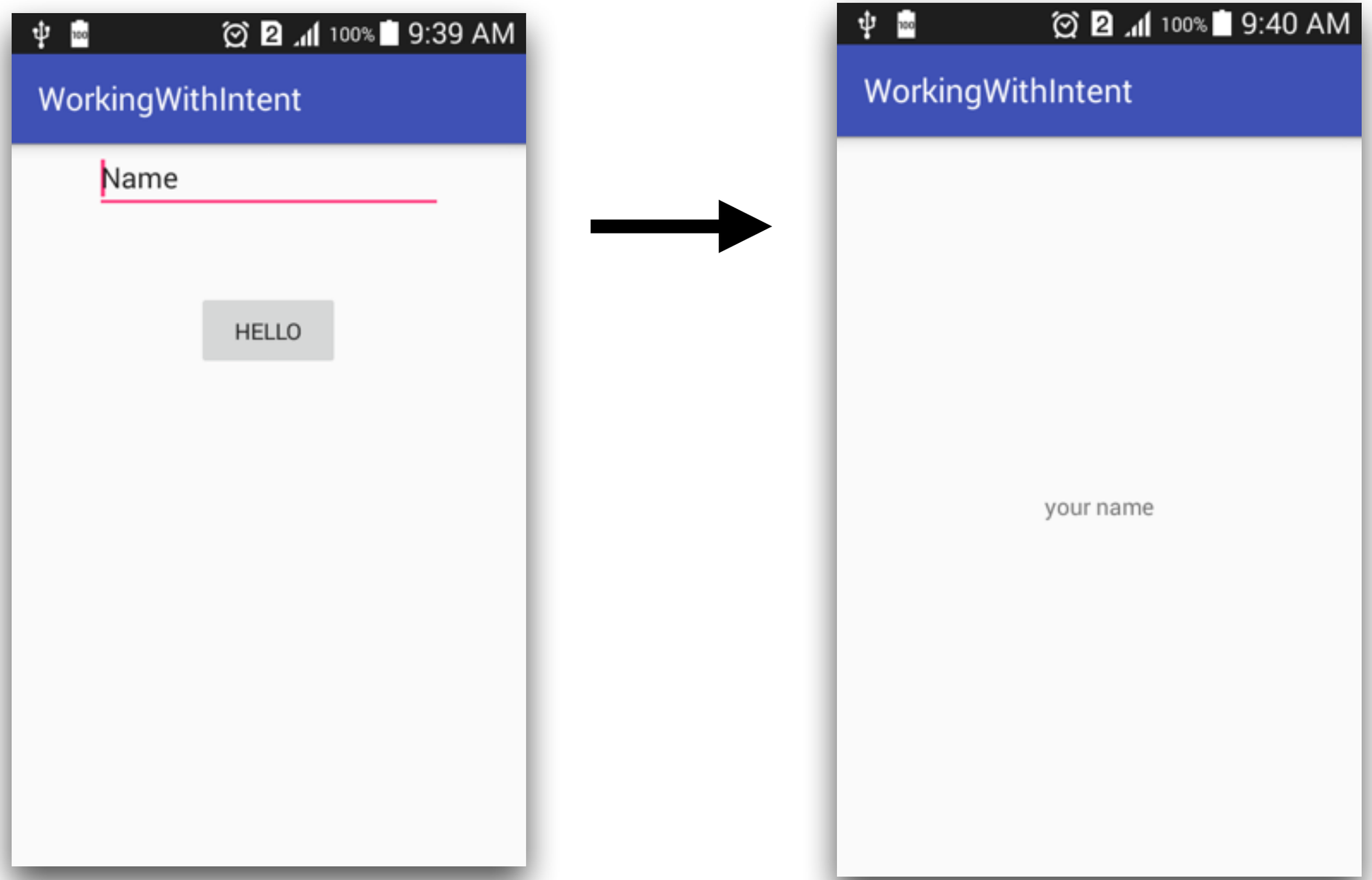
# Working with second activity

# Check result

```
public class SecondActivityTest {

    @Rule
    public ActivityTestRule activityTestRule
            = new ActivityTestRule(SecondActivity.class);


    @Test
    public void checkResultInView() {
        onView(withId(R.id.result)).check(matches(isDisplayed()));
        onView(withId(R.id.result)).check(matches(withText("Second activity")));
    }

}
```

# Working with intent

# Working with Intent



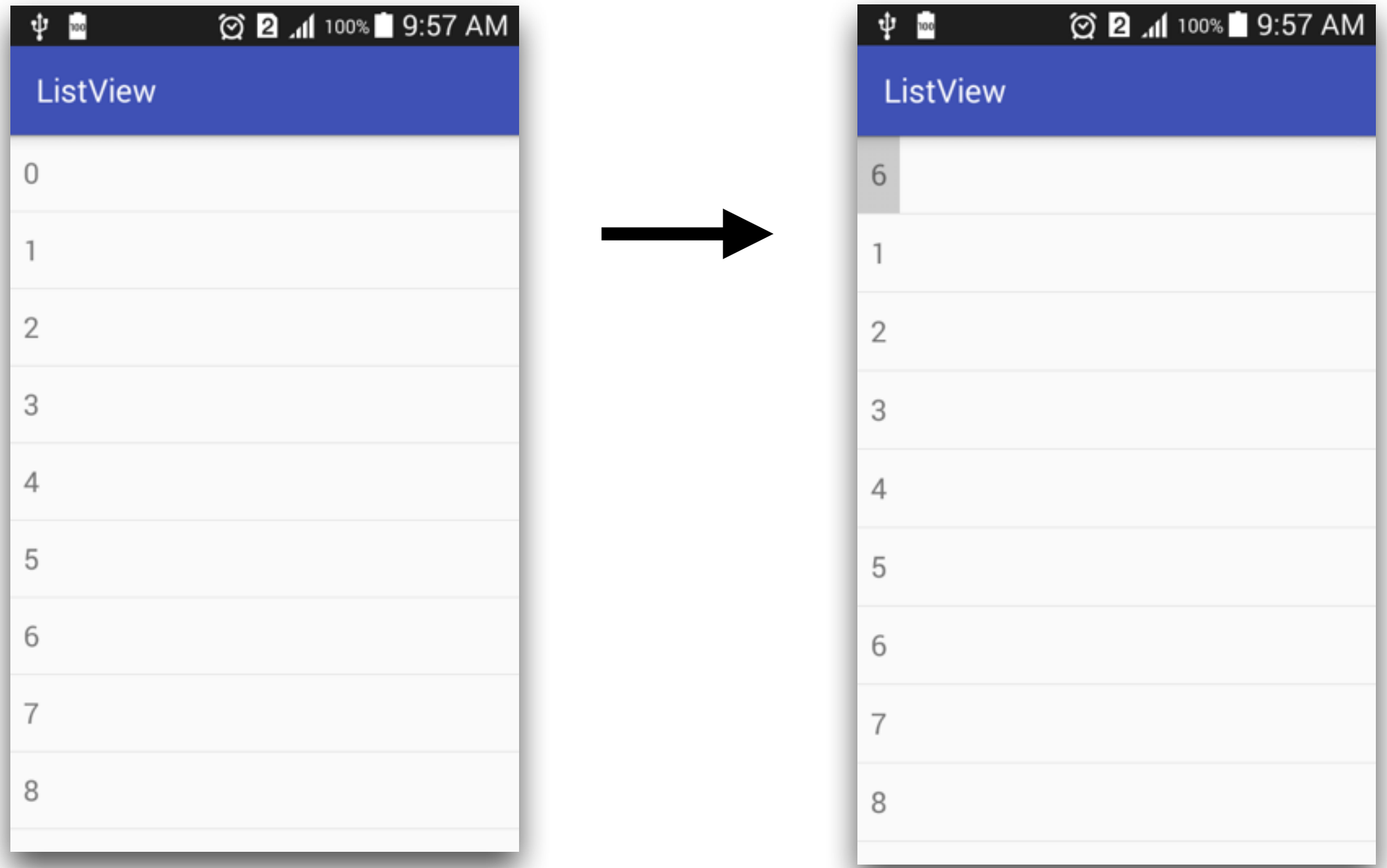
# Stub Intent

```
public class SecondActivityTest {

    @Rule
    public ActivityTestRule activityTestRule
            = new ActivityTestRule(SecondActivity.class, false, false);

    @Before
    public void initialInput() {
        Intent stubIntent = new Intent();
        stubIntent.putExtra("result", "My message");
        activityTestRule.launchActivity(stubIntent);
    }

    @Test
    public void showResult() {
        onView(withId(R.id.result)).check(matches(isDisplayed()));
        onView(withId(R.id.result)).check(matches(withText("My message")));
    }

}
```

# Workshop

# Working with Listview

# Working with Listview



# Resources

https://google.github.io/android-testing-support-library/

https://github.com/googlesamples/android-testing

https://github.com/googlesamples/android-testing-templates

https://github.com/googlesamples/android-architecture