

การทดสอบโมบายแอพฯ บนระบบปฏิบัติการแอนดรอย

Android Mobile Application Testing

สถาบัน ไอเเวมซี

ANDROID

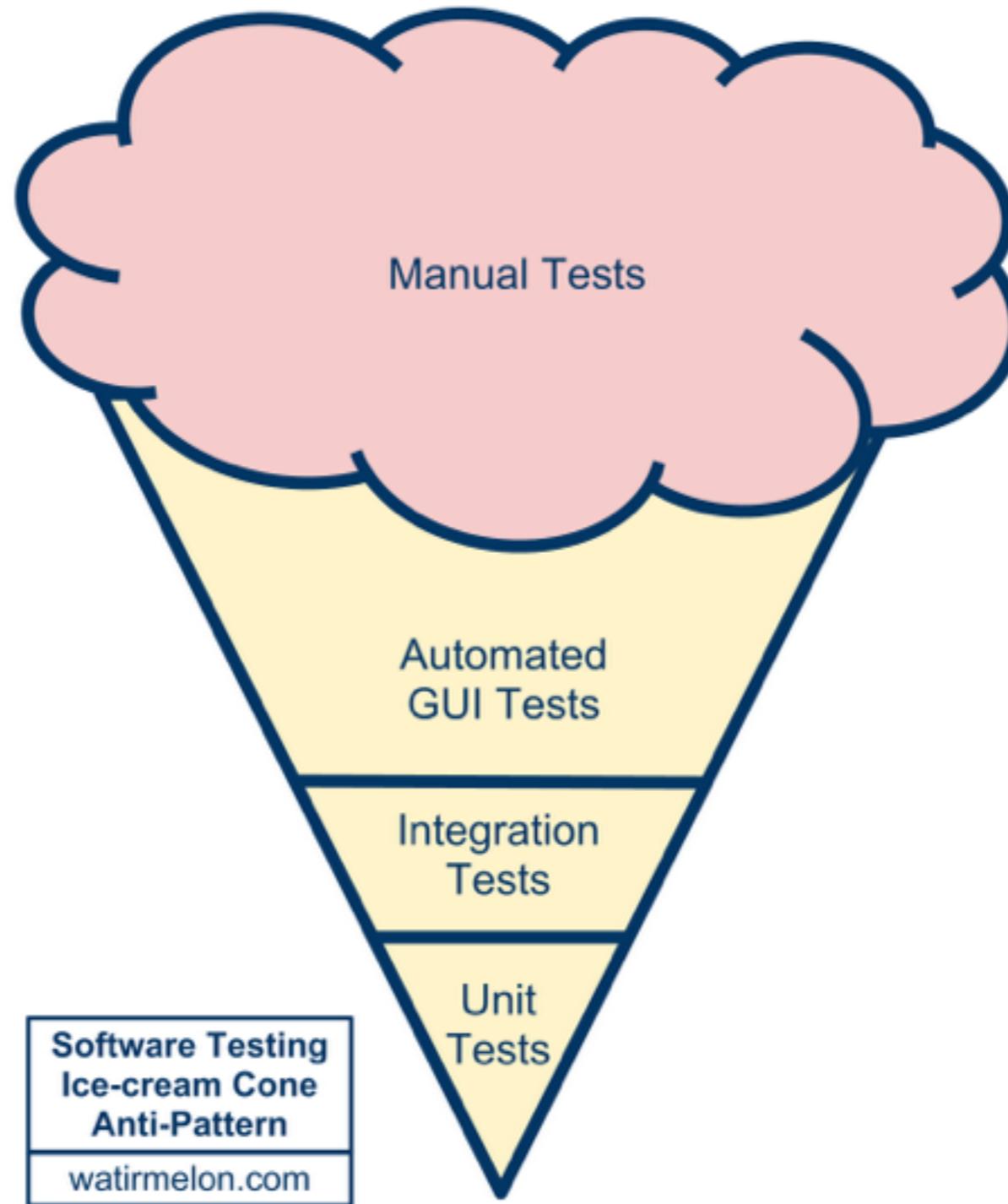
บริษัท สยามชานาญกิจ จำกัด และเพื่อนพ้องน้องพี่



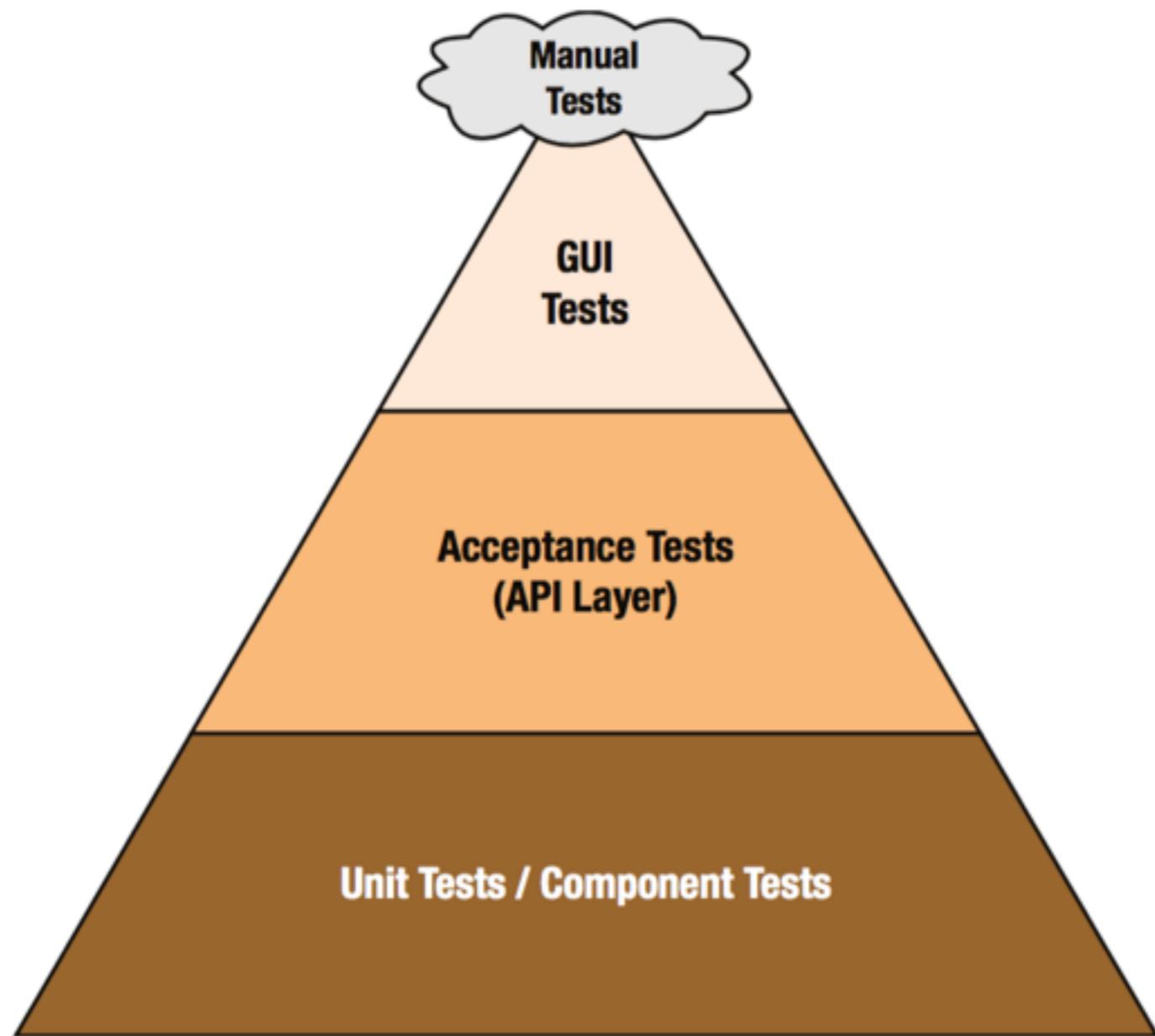
Testing



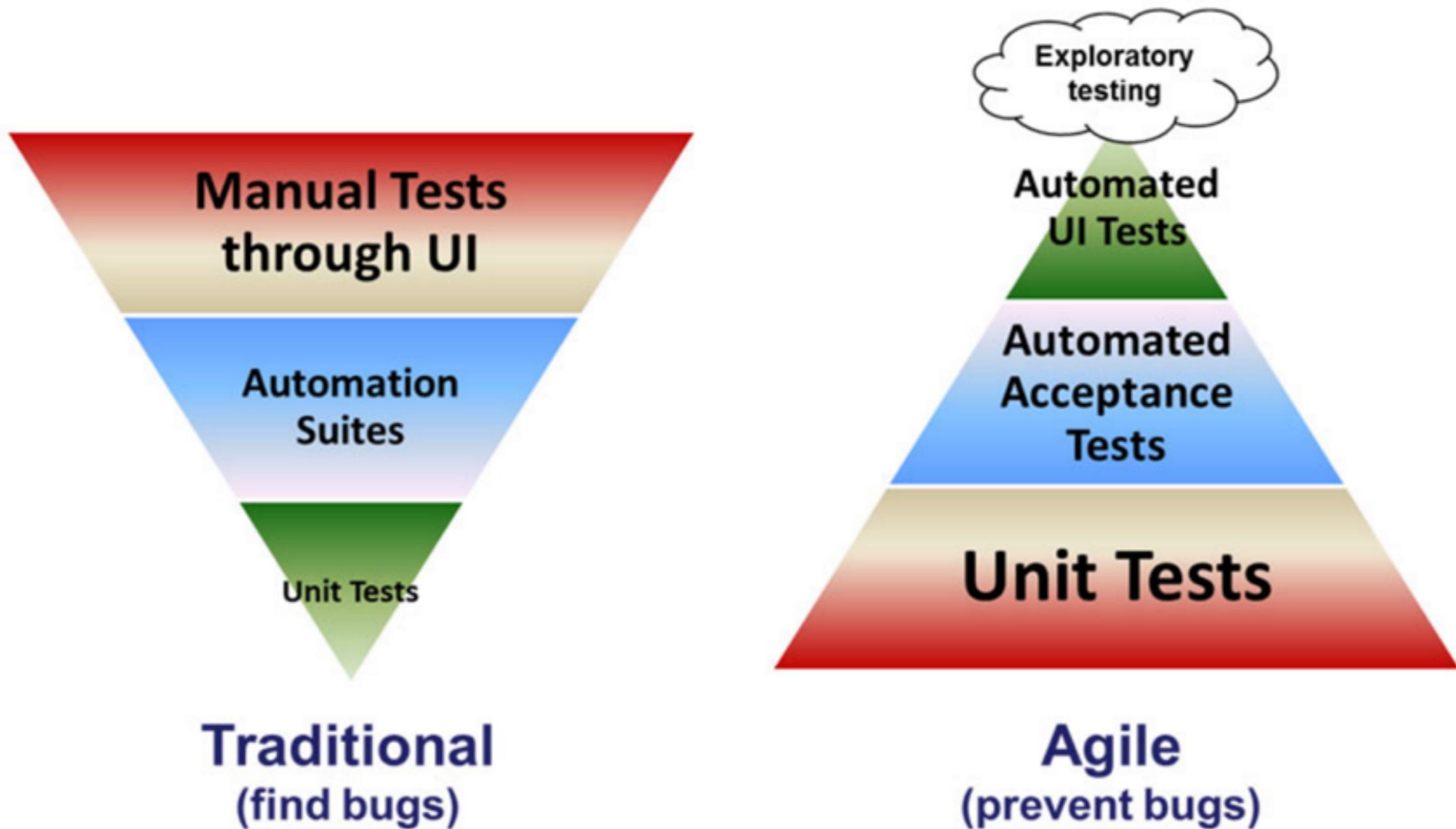
Testing



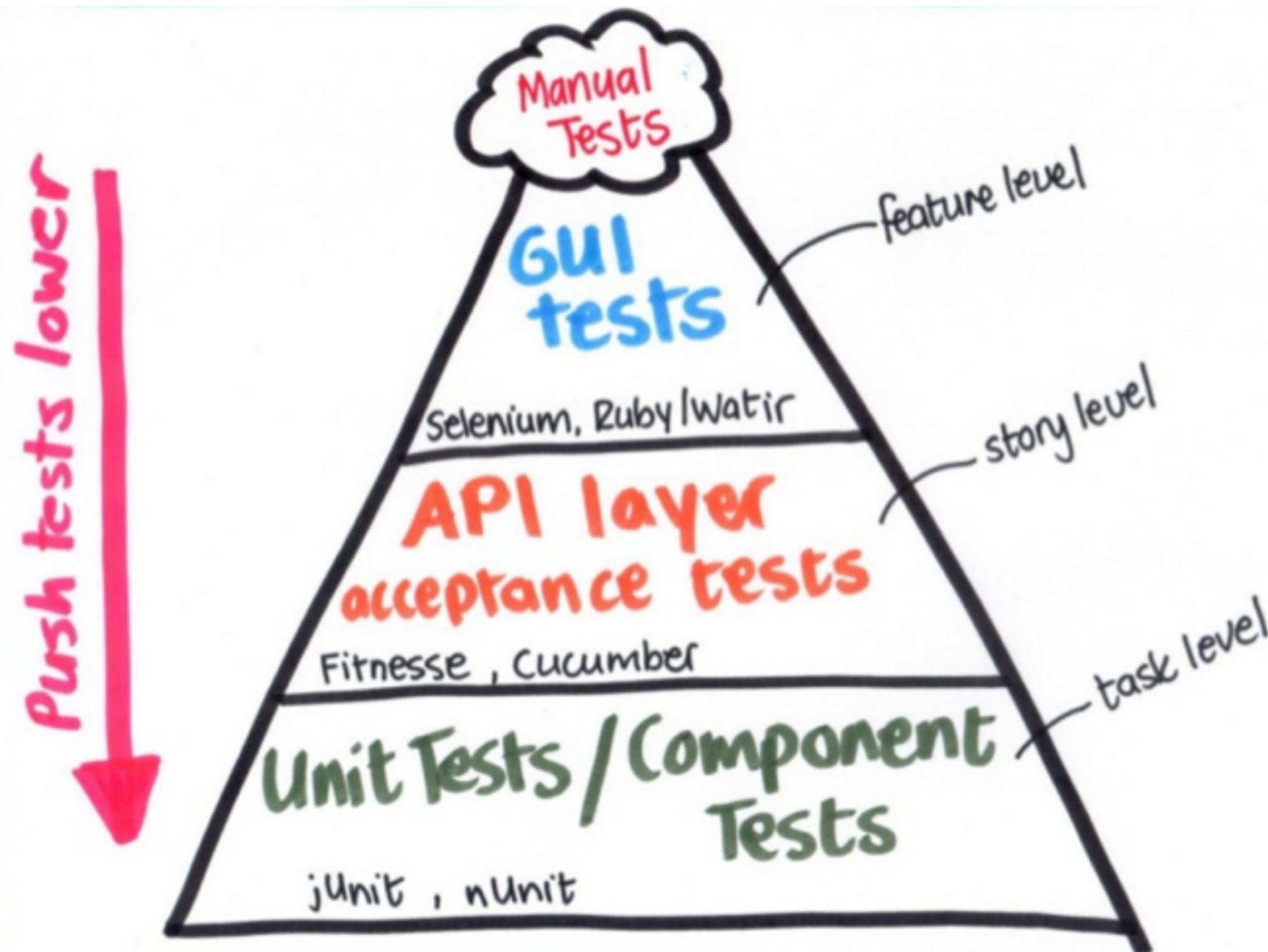
Testing pyramid



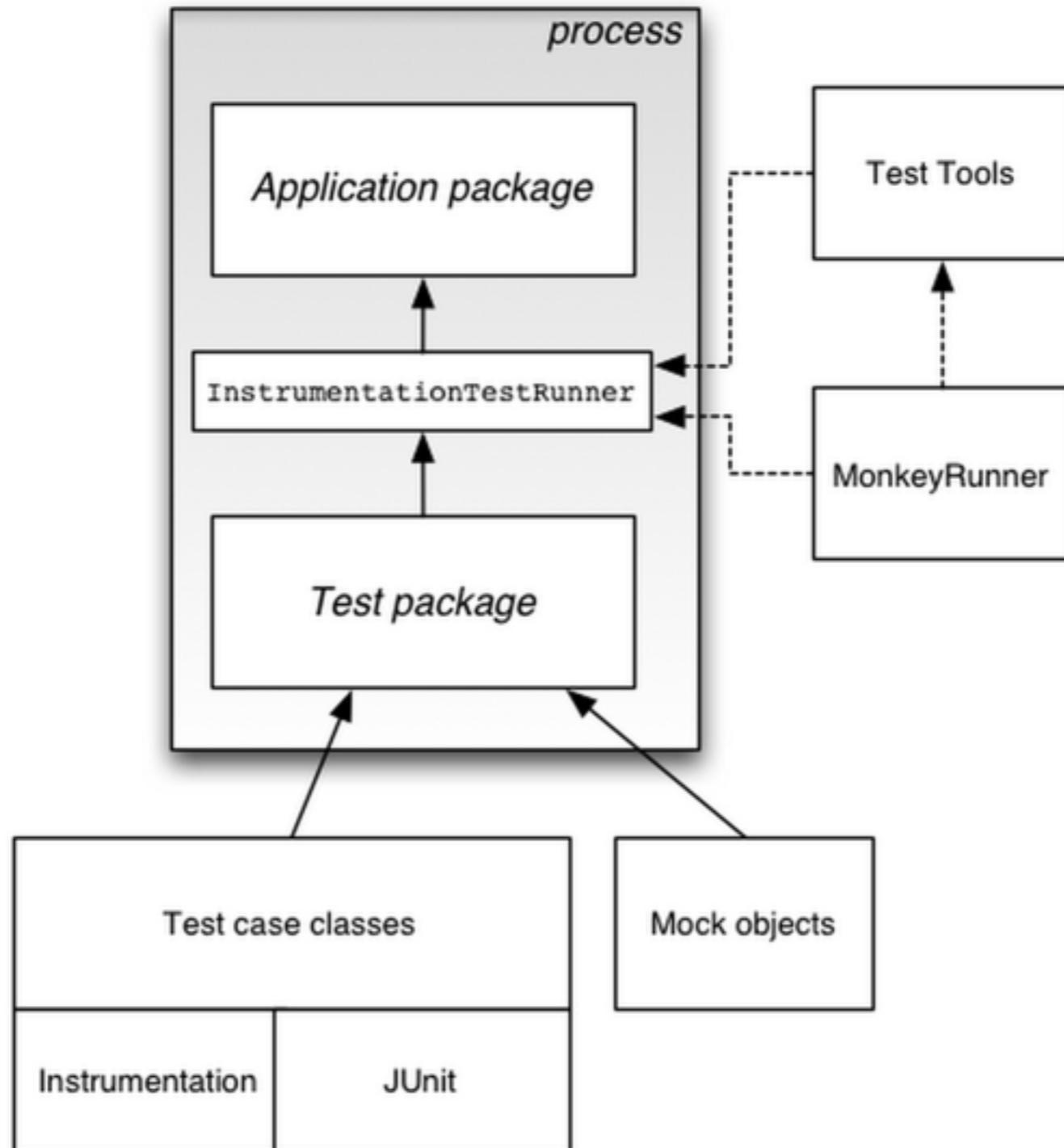
Testing pyramid



Testing pyramid



Testing for android



Testing for android

1. ต้องการ Emulator/Device
2. ไม่ต้องการ Emulator/Device



3rd-party tools



Hamcrest



Jenkins



JACOCO
Java Code Coverage



Stubby4J

sonarQube

espresso
UI TESTING FOR ANDROID

The Espresso logo features a brown Android robot head inside a grey coffee cup.

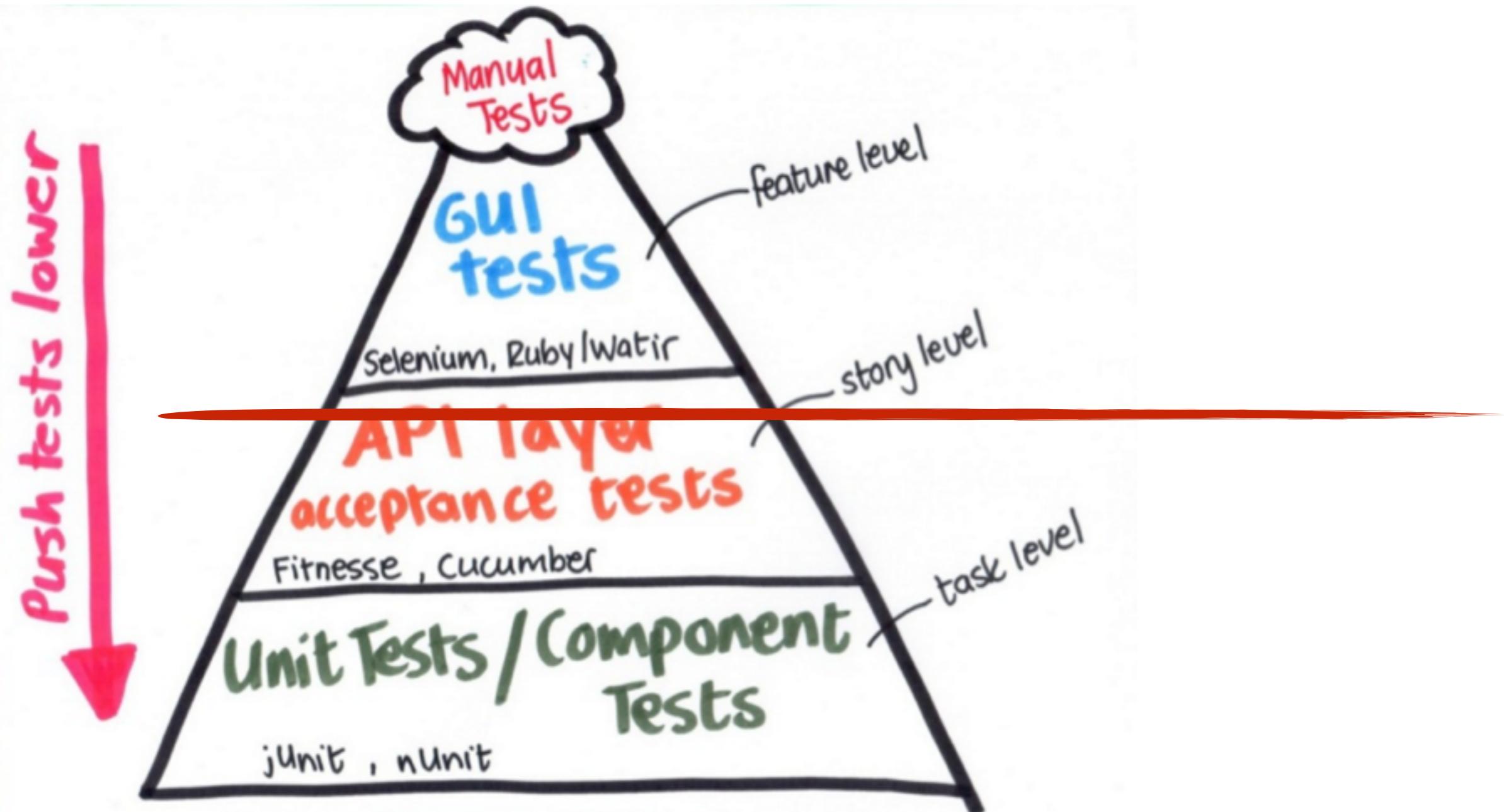
ROBOLECTRIC

The Robolectric logo features a white sheep's head inside a green scalloped circle.

JFrog Artifactory



Testing for android



Test sizes

Feature	Small	Medium	Large
Network access	No	localhost only	Yes
Database	No	Yes	Yes
File system access	No	Yes	Yes
Use external systems	No	Discouraged	Yes
Multiple threads	No	Yes	Yes
Sleep statements	No	Yes	Yes
System properties	No	Yes	Yes
Time limit (seconds)	60	300	900+

<http://googletesting.blogspot.com/2010/12/test-sizes.html>



We need small test !!

Feature	Small	Medium	Large
Network access	No	localhost only	Yes
Database	No	Yes	Yes
File system access	No	Yes	Yes
Use external systems	No	Discouraged	Yes
Multiple threads	No	Yes	Yes
Sleep statements	No	Yes	Yes
System properties	No	Yes	Yes
Time limit (seconds)	60	300	900+

<http://googletesting.blogspot.com/2010/12/test-sizes.html>



Unit testing

Code เพื่อใช้ทดสอบ Code



Unit testing

ไม่ต้องการ Emulator ในการทดสอบ
เร็วแน่นอน !!



Why Unit testing

Catch more mistakes
earlier in the development process



Why Unit testing

Confidently make more changes



Why Unit testing

Build in regression testing



Why Unit testing

Extend the life of your codebase



Good Unit Testing

Fast

Isolated

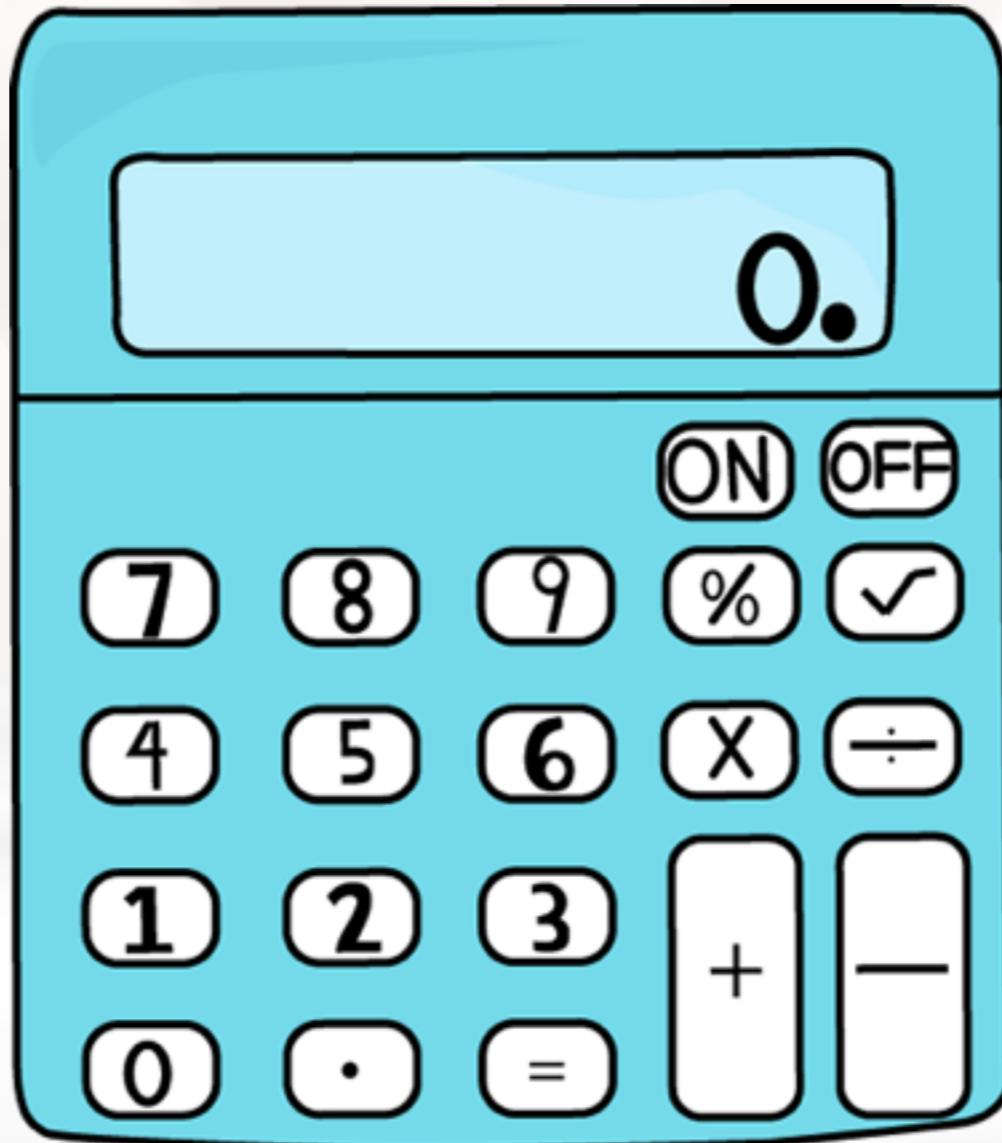
Repeatable

Self-verifying

Timely



Workshop



Workshop

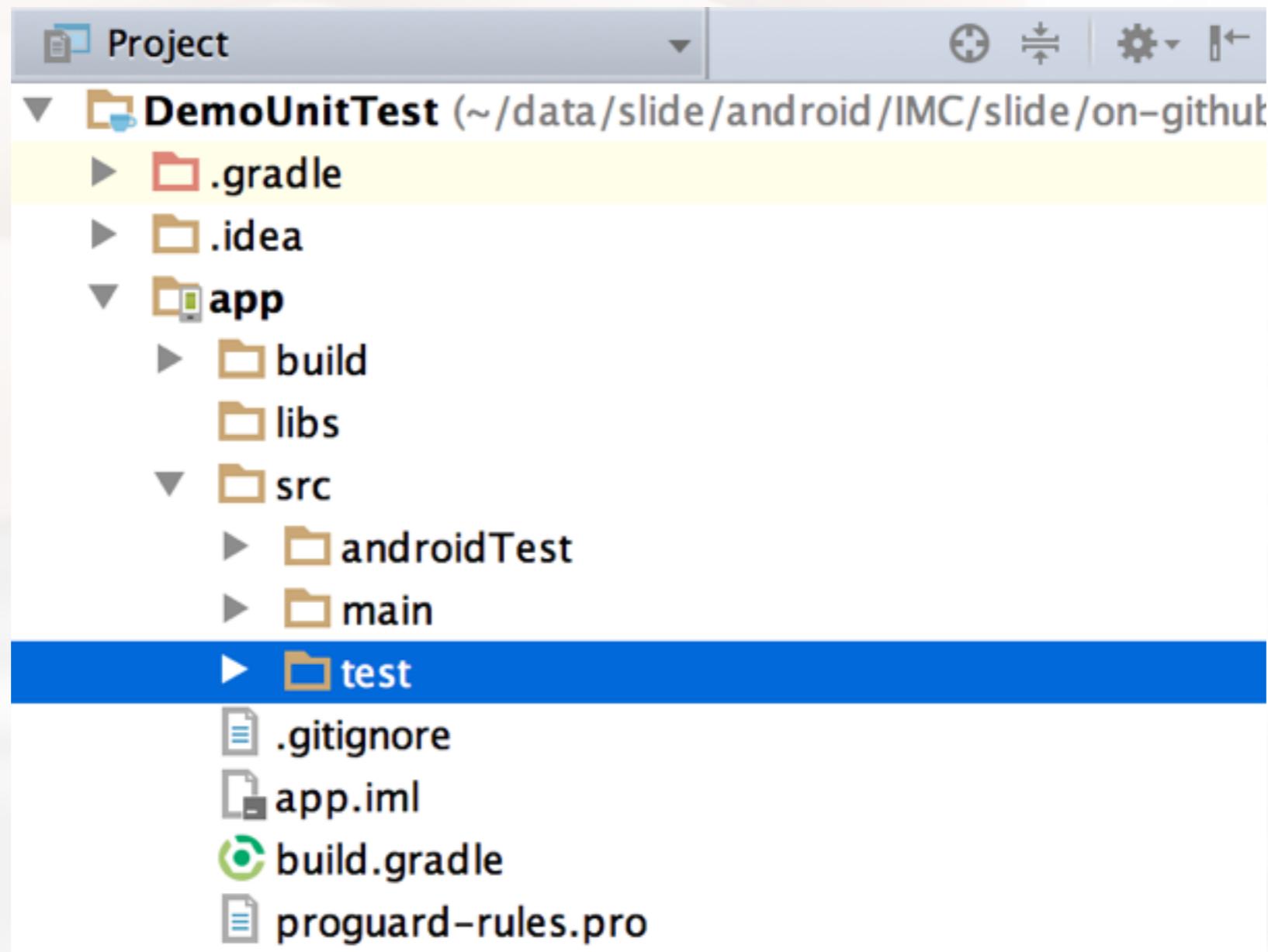
สร้าง project ใหม่ ชื่อ Calculator

เรียนรู้การสร้าง Unit test (jUnit4)

Code coverage



Project structure

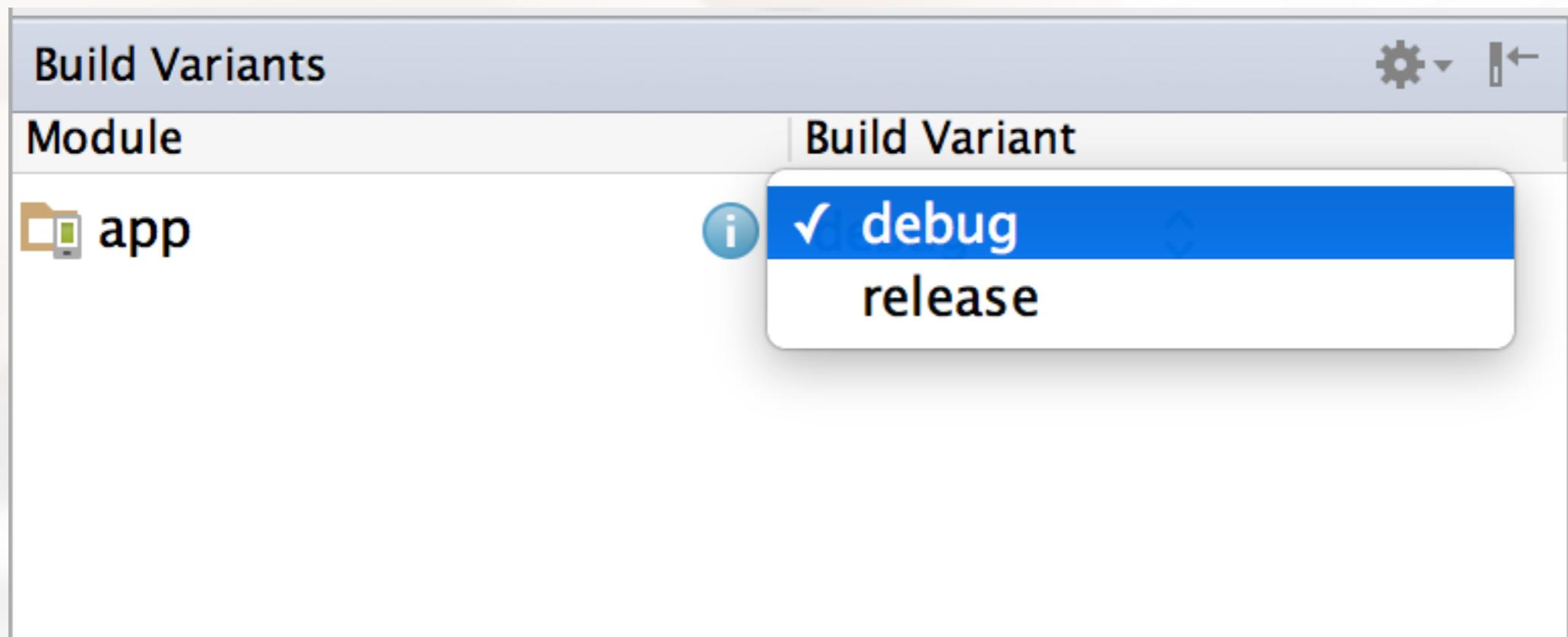


build.gradle

```
dependencies {  
    compile fileTree(dir: 'libs', include  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat_v7:23.0.1'  
}
```



Build variant

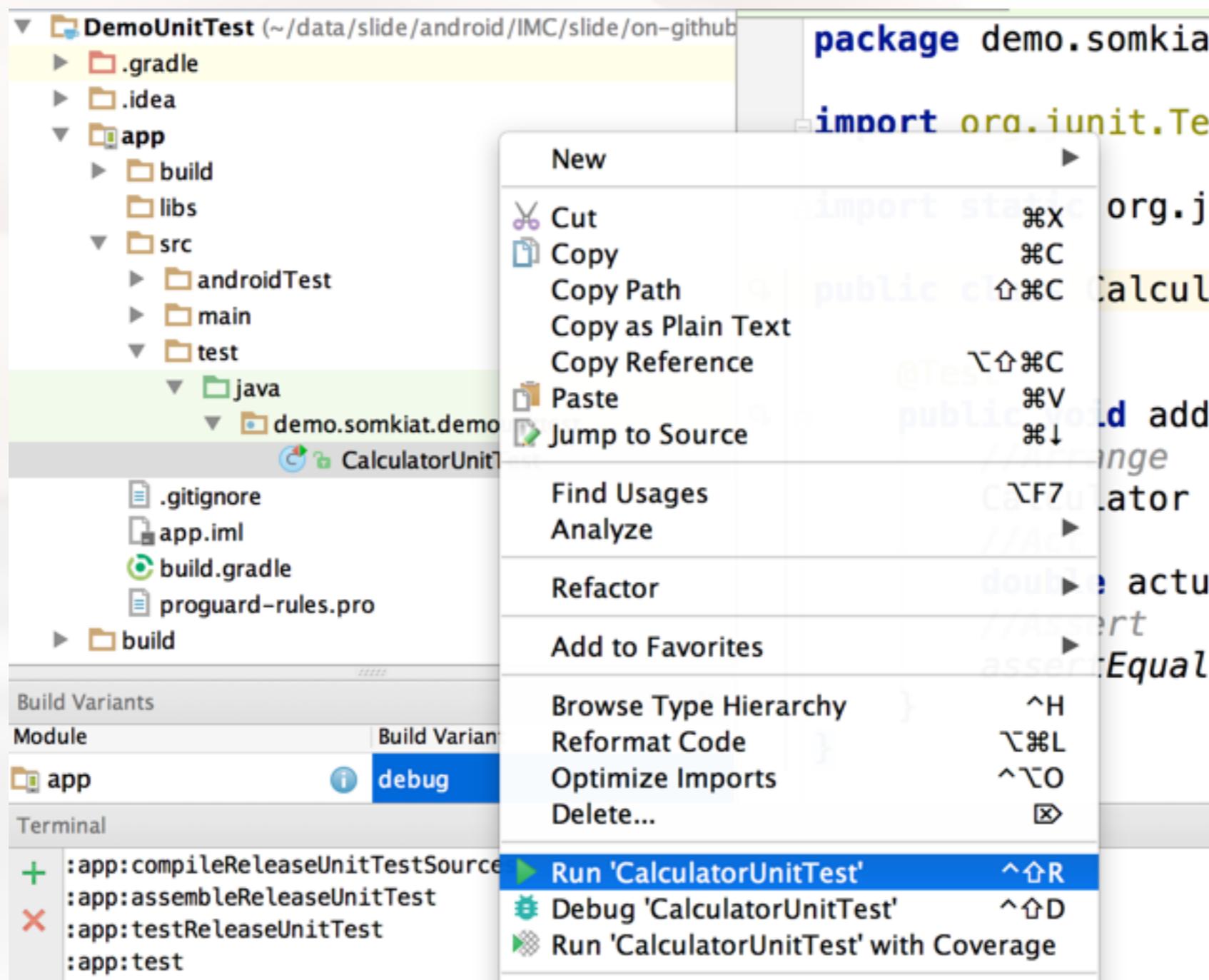


First unit test

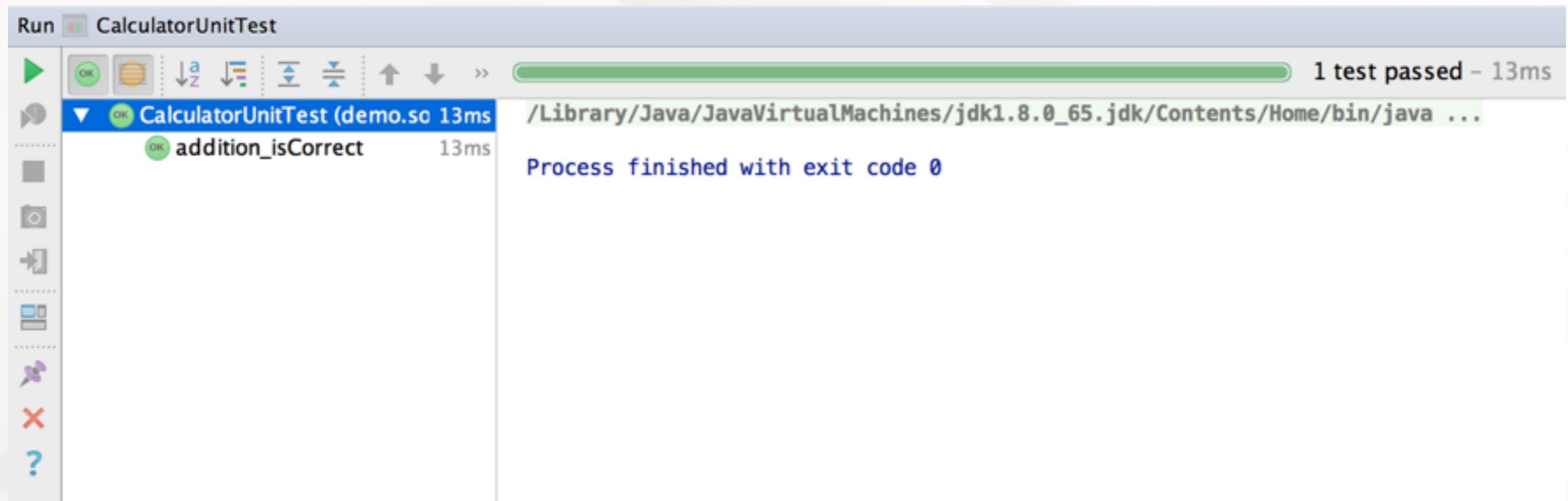
```
public class CalculatorUnitTest {  
  
    @Test  
    public void addition_isCorrect() throws Exception {  
        //Arrange  
        Calculator calculator = new Calculator();  
        //Act  
        double actualResult0fAdd = calculator.add(1, 2);  
        //Assert  
        assertEquals(3, actualResult0fAdd, 0);  
    }  
}
```



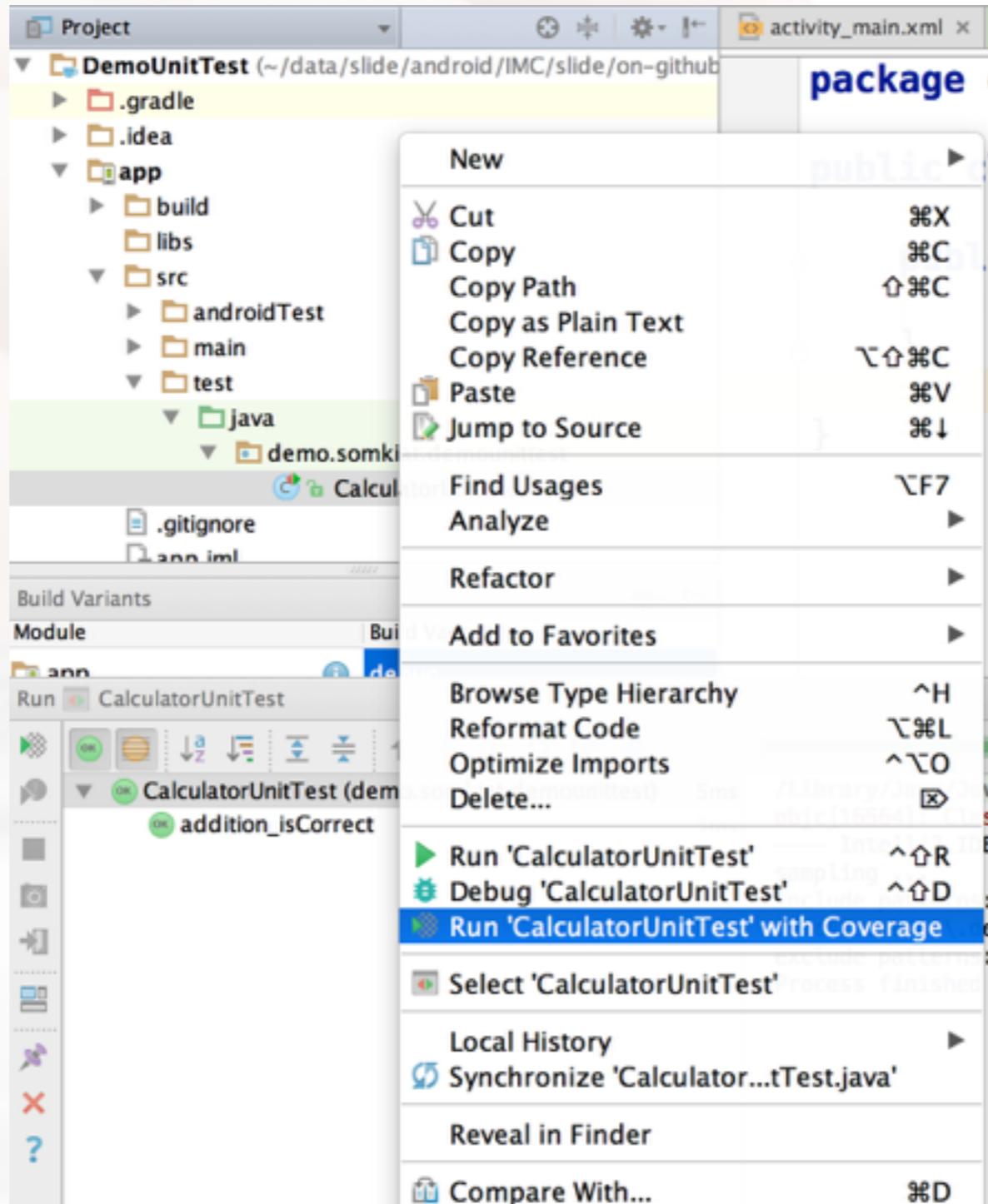
Run unit test



Run unit test



Run unit test with coverage



Run unit test with coverage

Coverage CalculatorUnitTest			
↑ 5% classes, 3% lines covered in package 'demo.somkiat.demounittest'			
Element	Class, %	Method, %	Line, %
BuildConfig	0% (0/1)	0% (0/1)	0% (0/2)
Calculator	100% (1/1)	100% (1/1)	100% (2/2)
MainActivity	0% (0/1)	0% (0/1)	0% (0/4)
R	0% (0/14)	0% (0/1)	0% (0/43)



Run unit test

`./gradlew test --continue`



Show report

Test Summary

1	0	0	0.002s
tests	failures	ignored	duration

100%
successful

Packages

Classes

Package	Tests	Failures	Ignored	Duration	Success rate
demo.somkiat.demounittest	1	0	0	0.002s	100%

Generated by [Gradle 2.10](#) at May 7, 2016, 2:50:21 PM



Run unit test with coverage

```
./gradlew createDebugCoverageReport
```



Enable code coverage

build.gradle

```
buildTypes {  
    debug {  
        testCoverageEnabled = true  
    }  
    release {  
        minifyEnabled false  
        proguardFiles getDefaultProguard...  
    }  
}
```



Show report

 [debug](#) >  [demo.somkiat.demounittest](#)

demo.somkiat.demounittest

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Mi
 MainActivity		0%		n/a	2	2	
 Calculator		0%		n/a	2	2	
Total	17 of 17	0%	0 of 0	n/a	4	4	



Run unit test with coverage

```
./gradlew createDebugCoverageReport
```

Working with instrumented test (AndroidTest) !!



Run unit test with coverage

Let's start to hack jacoco !!



Enable coverage for unit test

build.gradle

```
apply plugin: 'jacoco'

task jacocoTestReport(type: JacocoReport, dependsOn:

    reports {
        xml.enabled = true
        html.enabled = true
    }

    jacocoClasspath = configurations['androidJacocoA

    def fileFilter = ['**/R.class', '**/R*.class',
    def debugTree = fileTree(dir: "${buildDir}/inter
    def mainSrc = "${project.projectDir}/src/main/ja

    sourceDirectories = files([mainSrc])
    classDirectories = files([debugTree])
    executionData = files("${buildDir}/jacoco/testDe
```



Run unit test with coverage

```
./gradlew jacocoTestReport
```



Show report

 app >  demo.somkiat.demounittest

demo.somkiat.demounittest

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cqty	Missed	Lines	Missed
 MainActivity		0%		n/a	2	2	4	4	
 Calculator		100%		n/a	0	2	0	2	
Total	10 of 17	41%	0 of 0	n/a	2	4	4	6	



Jacoco android gradle

jacoco-android-gradle-plugin

[build](#) [passing](#) [Codecov](#) [98%](#) [Download](#) [0.1.1](#)

A Gradle plugin that adds fully configured `JacocoReport` tasks for unit tests of each Android application and library project variant.

Why

In order to generate JaCoCo unit test coverage reports for Android projects you need to create `JacocoReport` tasks and configure them by providing paths to source code, execution data and compiled classes. It can be troublesome since Android projects can have different flavors and build types thus requiring additional paths to be set. This plugin provides those tasks already configured for you.

<https://github.com/arturdm/jacoco-android-gradle-plugin>



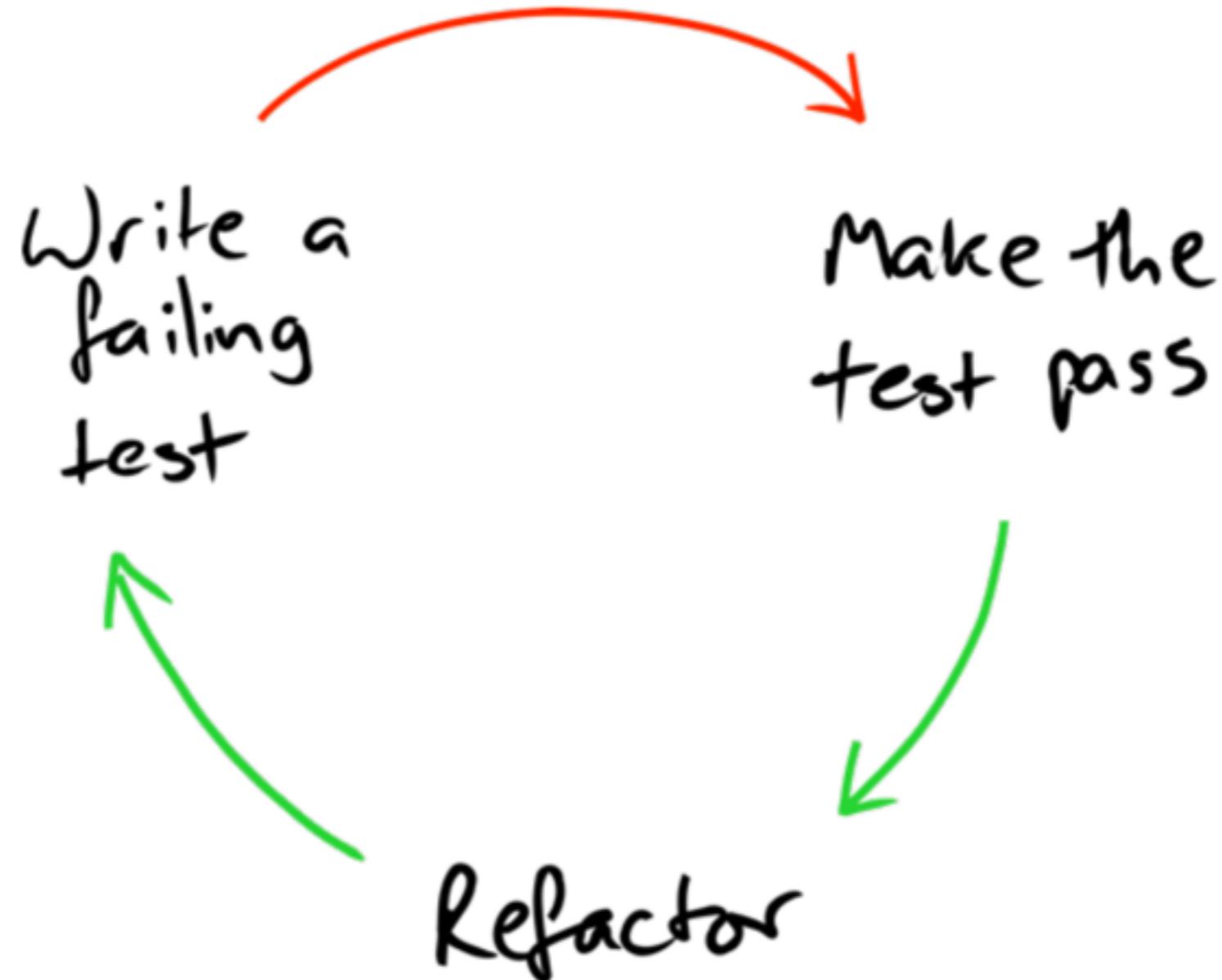
ແບບຝຶກຫັດ

ບວກ ລບ ຄູລ ຮາຮ

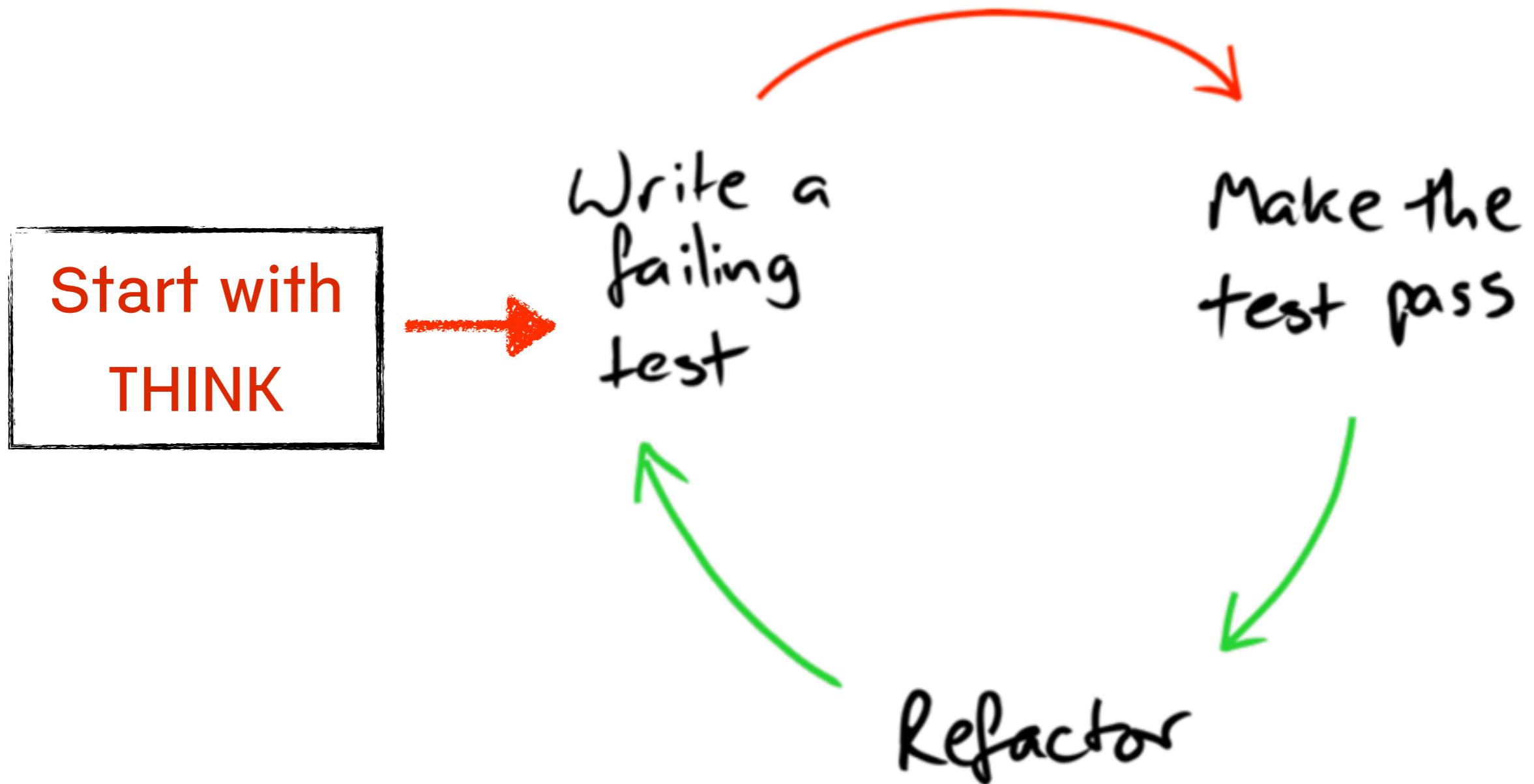
Code coverage = 100% !!



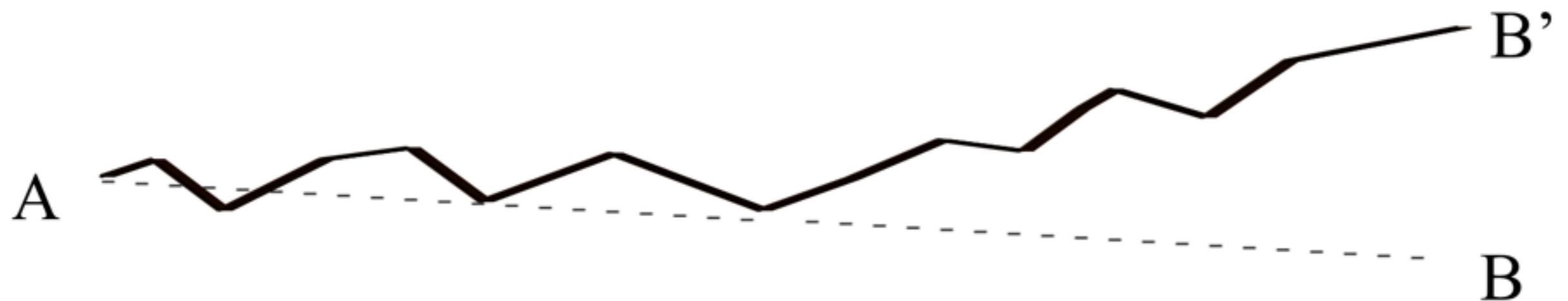
ทำดีดี



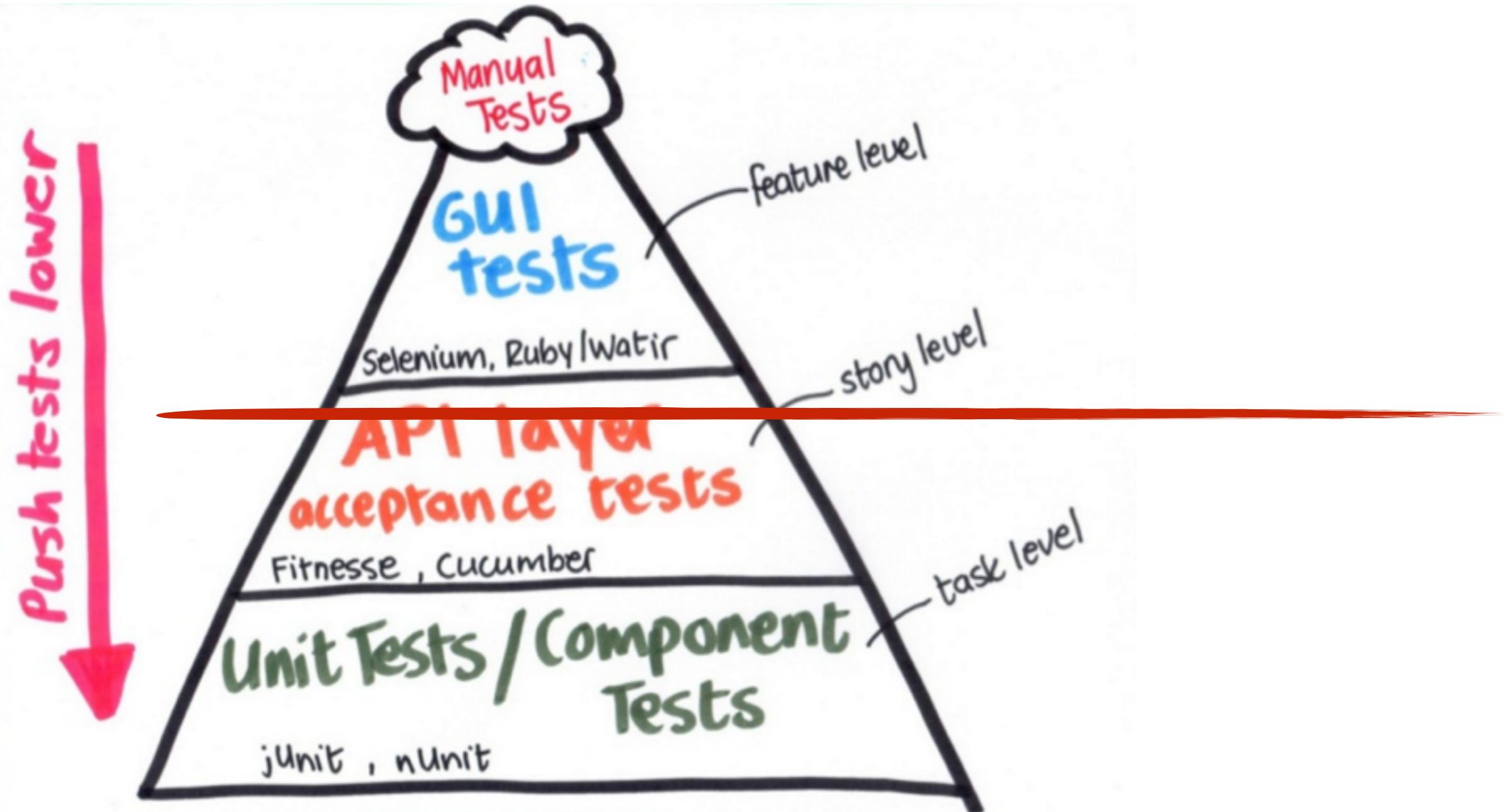
คิดก่อนทำดีดี



Small step



Testing for android



UI testing

ทำให้แน่ใจว่า App ใช้งานได้
บน device ต่าง ๆ



UI testing

Looks good but can't use !!



UI testing

ต้องการทดสอบการทำงานของ Activity

ใช้เวลาการทดสอบนาน

จำนวนการทดสอบมีเท่าที่จำเป็น



UI testing frameworks



UI TESTING FOR ANDROID
espresso

UI Automator

Calabash.sh



Robotium

 **appium**



UI testing



<https://google.github.io/android-testing-support-library/docs/espresso/index.html>



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Espresso components

ViewMatcher

ViewAction

ViewAssertion



Example



Example

```
@RunWith(AndroidJUnit4.class)
@LargeTest
public class CalculatorAddTest {

    public static final String ONE = "1";
    public static final String TWO = "2";
    public static final String RESULT = "3.0";

    @Rule
    public ActivityTestRule<MainActivity> mActivityRule =
        new ActivityTestRule<MainActivity>(MainActivity.class);

    @Test
    public void calculatorAdd() {
        onView(withId(R.id.operand_one_edit_text)).perform(typeText(ONE));
        onView(withId(R.id.operand_two_edit_text)).perform(typeText(TWO));
        onView(withId(R.id.operation_add_button)).perform(click());
        onView(withId(R.id.operation_result_text_view)).check(matches(withText(RESULT)));
    }
}
```



Workshop

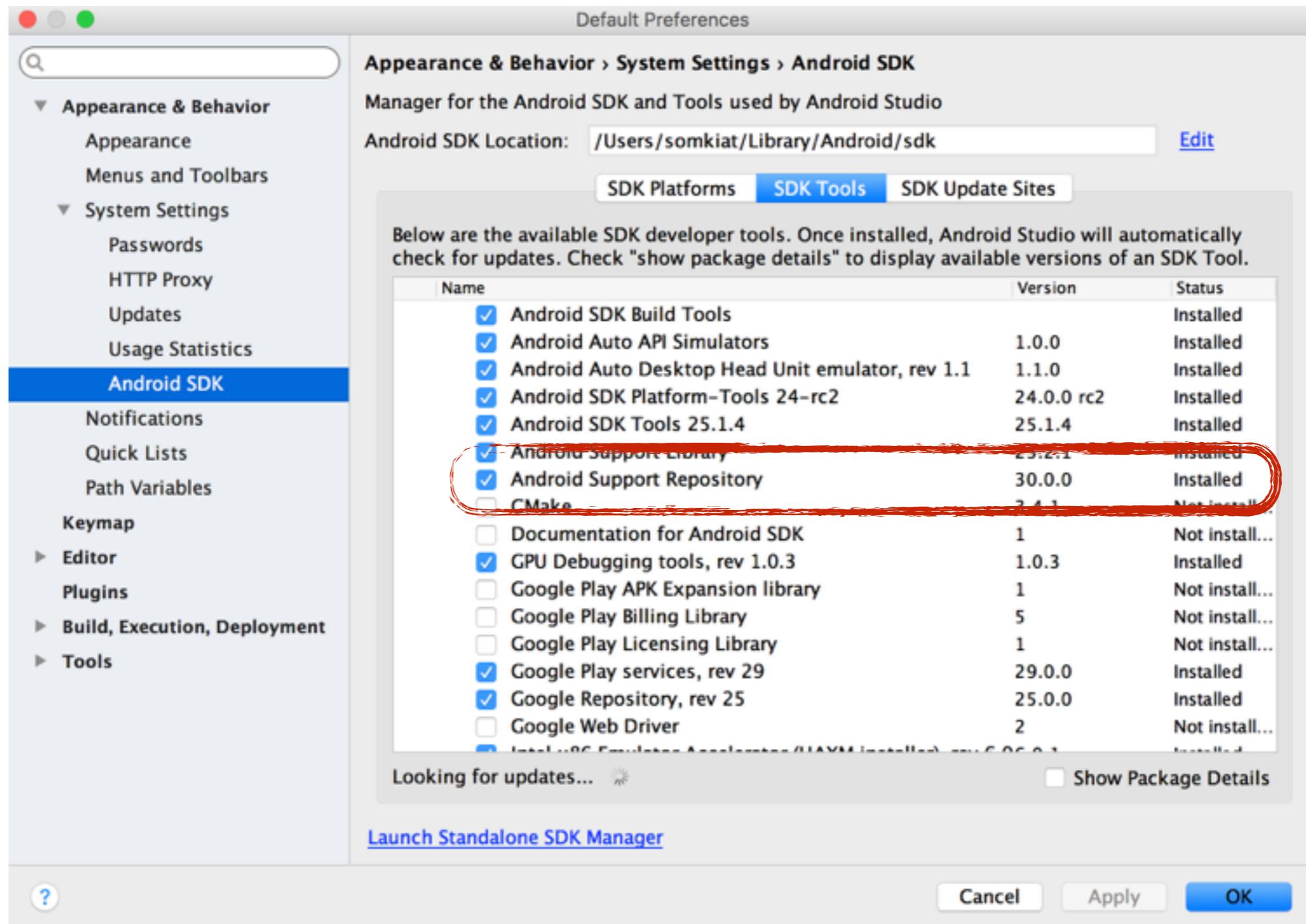
Configuration espresso in project

เรียนรู้การสร้าง UI test

Code coverage



ติดตั้ง Android support repository



เพิ่ม Espresso library

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:23.3.0'  
    compile 'com.android.support:support-annotations:23.3.0'  
    //Unit testing  
    testCompile 'junit:junit:4.12'  
    //UI testing with Espresso  
    androidTestCompile 'com.android.support:support-annotations:23.3.0'  
    androidTestCompile 'com.android.support.test:runner:0.5'  
    androidTestCompile 'com.android.support.test:rules:0.5'  
    androidTestCompile 'com.android.support.test.espresso:espresso-core:2.2.2'  
}
```



Add instrumentation runner

```
android {  
    compileSdkVersion 23  
    buildToolsVersion '24.0.0 rc3'  
  
    defaultConfig {  
        applicationId "demo.somkiat.demounittest"  
        minSdkVersion 15  
        targetSdkVersion 23  
        versionCode 1  
        versionName "1.0"  
  
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
    }  
}
```



First UI test

```
@RunWith(AndroidJUnit4.class)
@LargeTest
public class CalculatorAddTest {

    public static final String ONE = "1";
    public static final String TWO = "2";
    public static final String RESULT = "3.0";

    @Rule
    public ActivityTestRule<MainActivity> mActivityRule =
        new ActivityTestRule<MainActivity>(MainActivity.class);

    @Test
    public void calculatorAdd() {
        onView(withId(R.id.operand_one_edit_text)).perform(typeText(ONE));
        onView(withId(R.id.operand_two_edit_text)).perform(typeText(TWO));
        onView(withId(R.id.operation_add_button)).perform(click());
        onView(withId(R.id.operation_result_text_view)).check(matches(withText(RESULT)));
    }
}
```



Run UI test

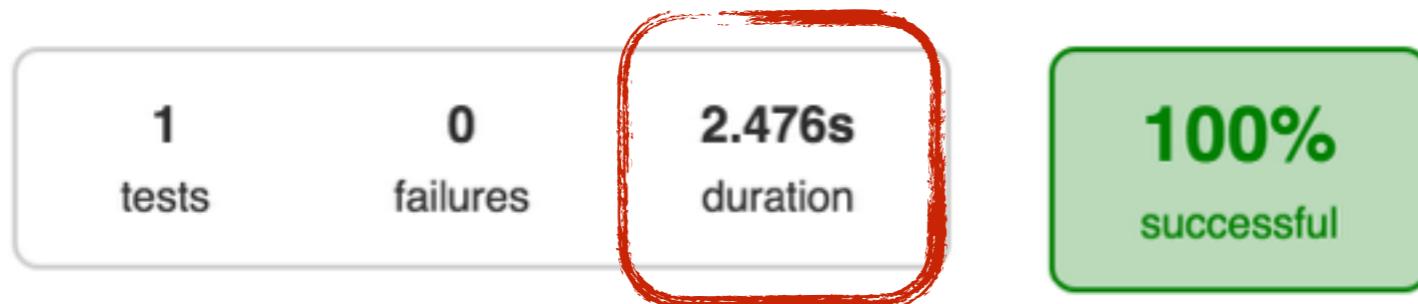
`./gradlew connectedAndroidTest`

`./gradlew cAT`



Show report

Test Summary



Packages

Classes

Package	Tests	Failures	Duration	Success rate
demo.somkiat.demounittest	1	0	2.476s	100%

Generated by [Gradle 2.10](#) at May 9, 2016, 6:22:50 AM



Show coverage report

 [debug](#) >  [demo.somkiat.demounittest](#)

demo.somkiat.demounittest

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	I
 MainActivity		85%		50%	1	6	
 Calculator		100%		n/a	0	2	
Total	13 of 91	86%	1 of 2	50%	1	8	



ແບບຝຶກຫັດ

ບວກ ລບ ຄູລ ຮາຮ

Code coverage = 100% !!



Merge coverage unit + ui

```
def fileFilter = ['**/R.class', '**/R$.class', '**/BuildConfig.*', '**/Manifest*']
def debugTree = fileTree(dir: "${buildDir}/intermediates/classes/debug", excludes)
def mainSrc = "${project.projectDir}/src/main/java"

sourceDirectories = files([mainSrc])
classDirectories = files([debugTree])
executionData = files(["${buildDir}/jacoco/testDebugUnitTest.exec",
                      "${buildDir}/outputs/code-coverage/connected/coverage.ec"])
])
```



Merge coverage unit + ui

```
./gradlew clean  
createDebugCoverageReport jacocoTestReport
```



Merge coverage unit + ui

 app >  demo.somkiat.demounittest

demo.somkiat.demounittest

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty
 MainActivity		85%		50%	1	6
 Calculator		100%		n/a	0	3
Total	13 of 95	86%	1 of 2	50%	1	9



Android unit testing

ใช้งาน JUnit 4.12

<https://github.com/junit-team/junit4>



Android assertions

Assertion	Description
assertEquals	Test that two values are the same
assertTrue	Test Boolean condition is true
assertFalse	Test Boolean condition is false
assertNull	Check that the object is null
assertNotNull	Check that the object is not null
assertSame	Test that both values refer to the same object reference
assertNotSame	Test that both values do not refer to the same object reference
assertThat	Test that the first value (object) matches the second value (or matcher)
fail	Test should always fail



Unit options

@Before

@After

@BeforeClass

@AfterClass

@Test

@Test(expected=exception)

@Test(timeout=ms)



Grouping tests

Feature	Small	Medium	Large
Network access	No	localhost only	Yes
Database	No	Yes	Yes
File system access	No	Yes	Yes
Use external systems	No	Discouraged	Yes
Multiple threads	No	Yes	Yes
Sleep statements	No	Yes	Yes
System properties	No	Yes	Yes
Time limit (seconds)	60	300	900+

<http://googletesting.blogspot.com/2010/12/test-sizes.html>



Example

```
@SmallTest  
@Test  
public void addition_isCorrect() throws Exception {  
    assertEquals(3, calculator.add(1, 2), 0);  
}  
  
@MediumTest  
@Test  
public void subtraction_isCorrect() throws Exception {  
    assertEquals(1, calculator.sub(2, 1), 0);  
}  
  
@LargeTest  
@Test  
public void divide_isCorrect() throws Exception {  
    assertEquals(2, calculator.div(2, 1), 0);  
}
```



ระดมสมอง

Unit test ?

UI test ?

<https://plus.google.com/+AndroidDevelopers/posts/TPy1EeSaSg8>



บริษัท สยามชานาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Run with instrumentation runner

```
defaultConfig {  
    applicationId "demo.somkiat.demounittest"  
    minSdkVersion 15  
    targetSdkVersion 23  
    versionCode 1  
    versionName "1.0"  
  
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
    testInstrumentationRunnerArgument "size", "medium,large"  
}  
-
```



Run test

`./gradlew clean cAT`



Parameterized tests

```
@Test  
public void addition_isCorrect() throws Exception {  
    assertEquals(3, calculator.add(1, 2), 0);  
    assertEquals(3, calculator.add(2, 1), 0);  
    assertEquals(10, calculator.add(8, 2), 0);  
    assertEquals(5, calculator.add(-1, 6), 0);  
}
```

ได้กลืนอะไรใหม่ ?



Parameterized tests

```
@Test  
public void addition_isCorrect() throws Exception {  
    assertEquals(3, calculator.add(1, 2), 0);  
    assertEquals(3, calculator.add(2, 1), 0);  
    assertEquals(10, calculator.add(8, 2), 0);  
    assertEquals(5, calculator.add(-1, 6), 0);  
}
```

มา Refactor code กัน



1. Add @RunWith(Parameterized.class)

```
@RunWith(Parameterized.class)
public class CalculatorAddUnitTest {
    Calculator calculator = new Calculator();
```



2. Add datas

```
@RunWith(Parameterized.class)
public class CalculatorAddUnitTest {
    Calculator calculator = new Calculator();

    @Parameters
    public static List<Object[]> data() {
        return Arrays.asList(new Object[][] {
            {1, 2, 3},
            {2, 1, 3},
            {8, 2, 10},
            {-1, 6, 5},
        });
    }
}
```



3. Add constructor

```
@RunWith(Parameterized.class)
public class CalculatorAddUnitTest {
    Calculator calculator = new Calculator();

    public CalculatorAddUnitTest(int mOperandOne,
                                  int mOperandTwo,
                                  int mExpectedResult) {
        this.mOperandOne = mOperandOne;
        this.mOperandTwo = mOperandTwo;
        this.mExpectedResult = mExpectedResult;
    }
}
```



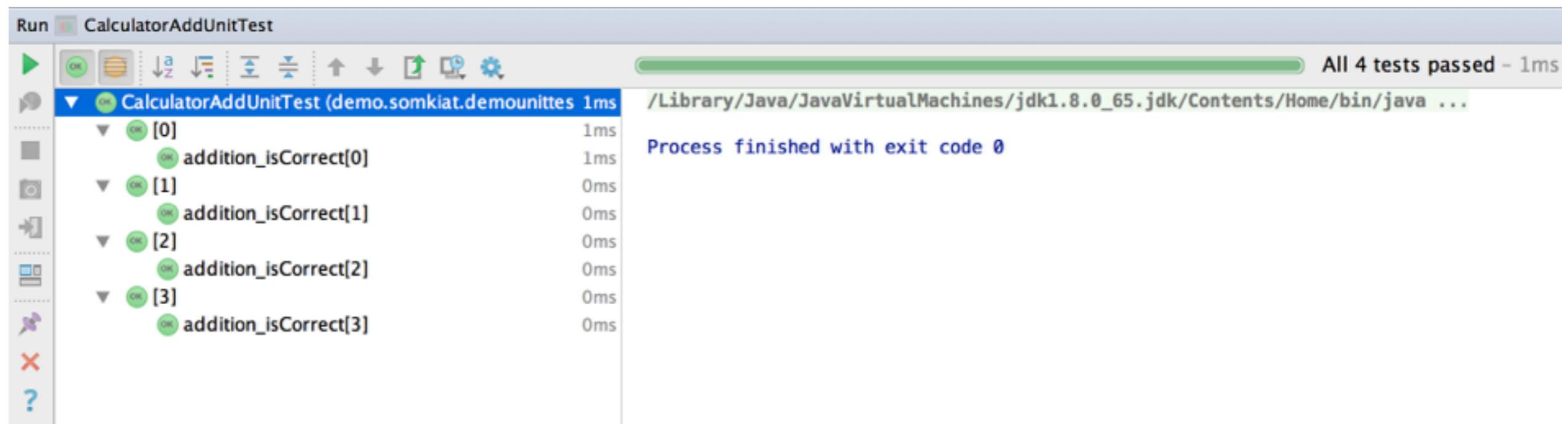
4. Use parameters in test

```
@RunWith(Parameterized.class)
public class CalculatorAddUnitTest {
    Calculator calculator = new Calculator();

    @Test
    public void addition_isCorrect(){
        assertEquals(this.mExpectedResult,
                    calculator.add(this.mOperandOne, this.mOperandTwo), 0);
    }
}
```



Run test & Show report



The screenshot shows a test runner interface with the following details:

- Run** tab selected.
- CalculatorAddUnitTest** is the current test suite.
- All 4 tests passed - 1ms** status message.
- CalculatorAddUnitTest (demo.somkiat.demounittes 1ms)** is expanded.
- Four test cases under [0] to [3] all passed (**OK**):
 - [0]: addition_isCorrect[0]
 - [1]: addition_isCorrect[1]
 - [2]: addition_isCorrect[2]
 - [3]: addition_isCorrect[3]
- Execution path: /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
- Process finished with exit code 0



Run test & Show report

Class demo.somkiat.demounittest.CalculatorAddUnitTest

[all](#) > [demo.somkiat.demounittest](#) > CalculatorAddUnitTest

4
tests

0
failures

0
ignored

0.001s
duration

100%
successful

Tests

Test	Duration	Result
addition_isCorrect[0]	0.001s	passed
addition_isCorrect[1]	0s	passed
addition_isCorrect[2]	0s	passed
addition_isCorrect[3]	0s	passed



ແບບຝຶກຫັດ

ทำการ run ຖຸກ ໃງ ກາຣທດສອບ
ທັງ unit test ແລະ ui test



**With or without TDD,
unit testing needs to become part of your development
process**



More testing

Monkey testing



<http://developer.android.com/tools/help/monkey.html>



Monkey testing

Stress testing

Command line tool

Random events



How to use ?

\$adb shell monkey -p <your package> -v <# of events>



Workshop

```
$adb shell monkey -p demo.somkiat.demounittesting  
-v 20000
```

See result and fix it !!



We need a continuous testing





Jenkins

Bamboo



TeamCity

> go™



Hudson





Jenkins

Bamboo

CI is about what people do
not about what tools they use



Visual Studio



Team Foundation Server

Hudson



travis

wercker

circleci



CI is a practice

Discipline to integrate frequently



CI is a practice

Strive to make small change

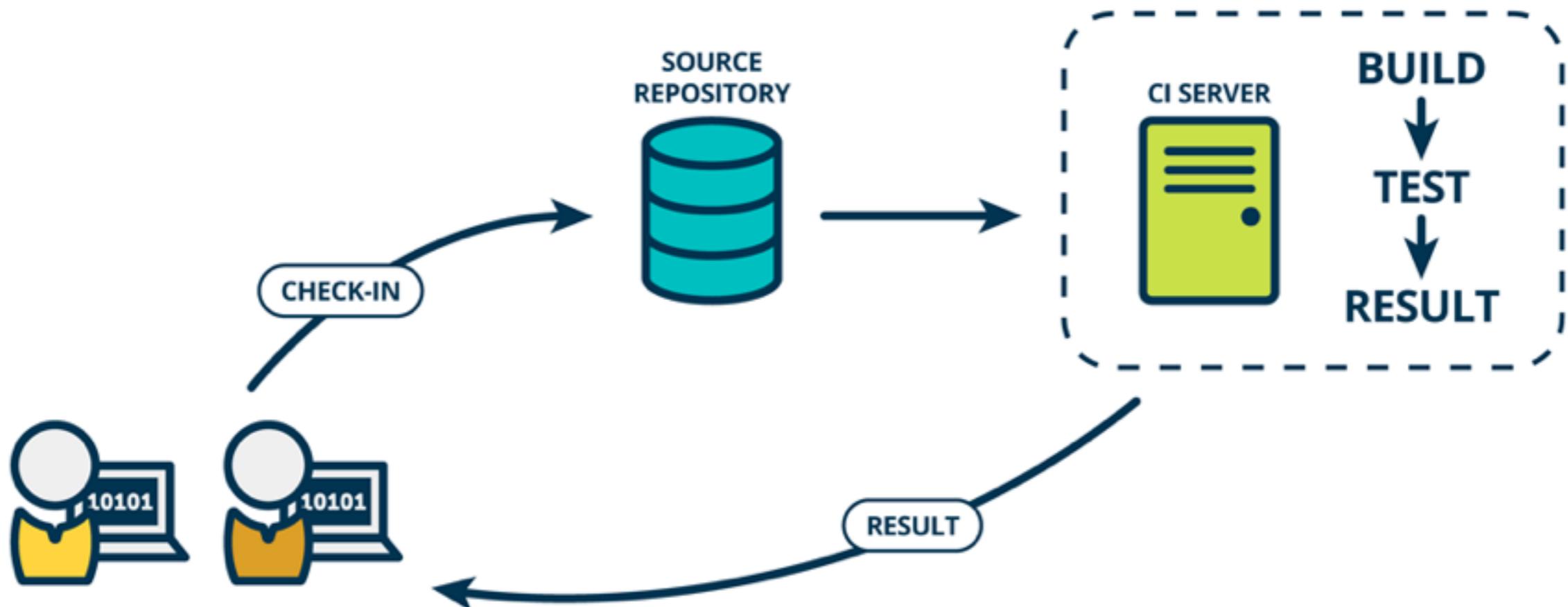


CI is a practice

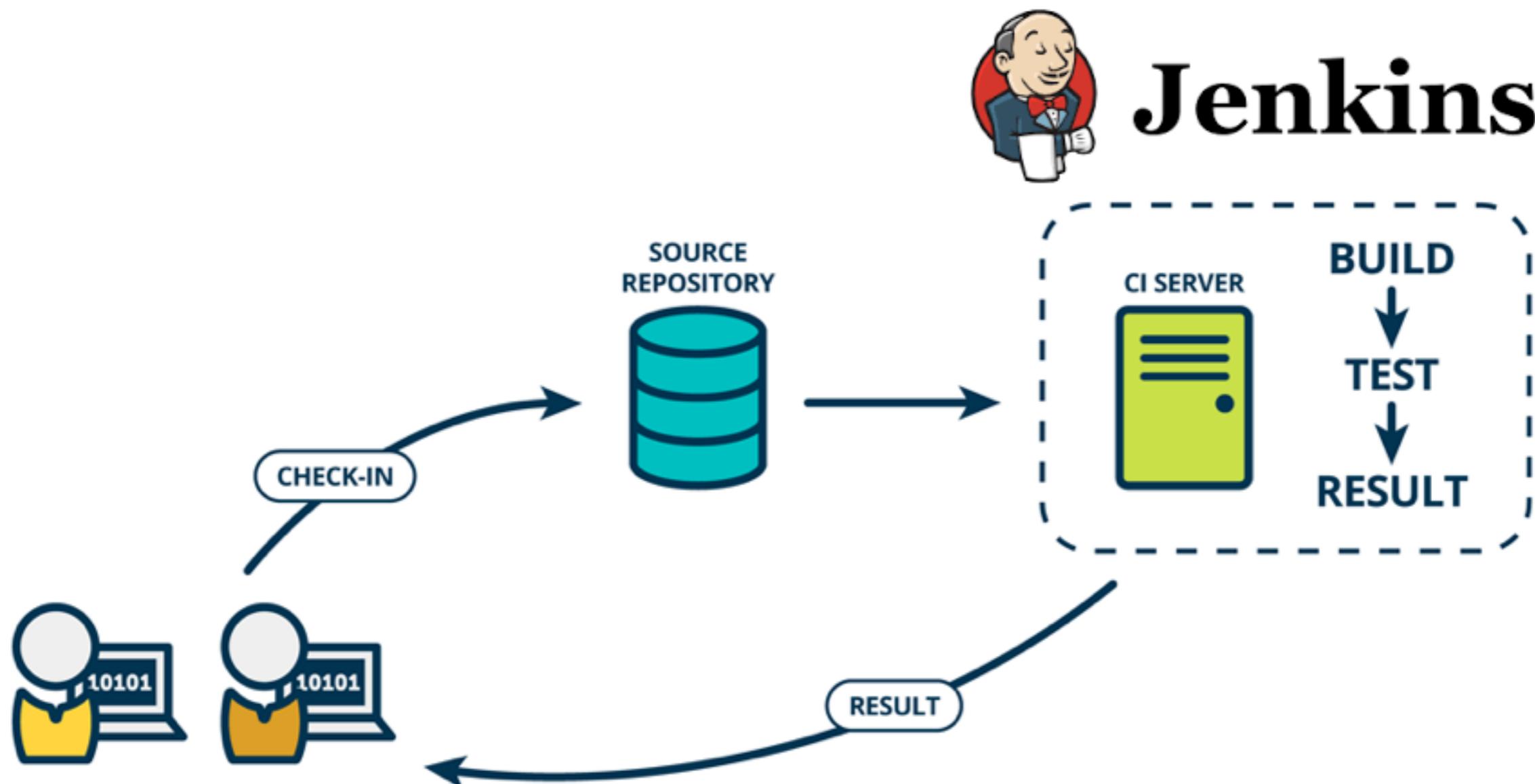
Strive for **fast feedback**



Continuous Integration



Continuous Integration Server



Workshop with Jenkins

ติดตั้ง Jenkins

ติดตั้ง plug-ins ต่าง ๆ

สร้างระบบการทดสอบแบบอัตโนมัติ



Download Jenkins

Jenkins [Downloads](#) ▾ [Participate](#) ▾ [Use-cases](#) ▾ [Blog](#) [Documentation](#) [Plugins](#) [Wiki](#) [Issues](#) [Security](#) [Press](#)

Conduct Account

Fork me on GitHub



Jenkins

Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Download Jenkins](#)

Get 1.651.1 LTS .war or the latest 2.1 weekly release

<https://jenkins-ci.org/>



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Start Jenkins

\$java -jenkins.war



Welcome to Jenkins

The screenshot shows the Jenkins dashboard. At the top left is the Jenkins logo. The main header says "Welcome to Jenkins!". Below it, a teal box contains the text "Please create new jobs to get started.". On the left, there's a sidebar with links: "New Item", "People", "Build History", "Manage Jenkins", "Credentials", and "My Views". Below the sidebar is a "Build Queue" section with the message "No builds in the queue."

- New Item
- People
- Build History
- Manage Jenkins
- Credentials
- My Views

Build Queue

No builds in the queue.



Configure Jenkins

Jenkins  Jenkins

Plugin Manager

Back to Dashboard  Manage Jenkins 

Updates Available Installed Advanced Filter: 

Enabled	Name ↓	Version	Previously installed version
<input type="checkbox"/>	Ant Plugin This plugin adds Apache Ant support to Jenkins.	1.2	
<input type="checkbox"/>	Branch API Plugin This plugin provides an API for multiple branch based projects.	1.4	
<input checked="" type="checkbox"/>	Build Monitor View Provides a highly visible view of the status of selected Jenkins jobs. It easily accommodates different computer screen sizes and is ideal as an Extreme Feedback Device to be displayed on a screen on your office wall.	1.8+build.201601112328	
<input checked="" type="checkbox"/>	Build Pipeline Plugin This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.	1.5.1	
<input checked="" type="checkbox"/>	build timeout plugin This plugin allows builds to be automatically terminated after the specified amount of time has elapsed.	1.16	
<input checked="" type="checkbox"/>	Cobertura Plugin This plugin integrates Cobertura coverage reports to Jenkins.	1.9.7	
<input checked="" type="checkbox"/>	Credentials Binding Plugin Allows credentials to be bound to environment variables for use from miscellaneous build steps.	1.7	



Create automated job

The screenshot shows the Jenkins dashboard with a sidebar on the left containing links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'Credentials', and 'My Views'. Below these are sections for 'Build Queue' (empty) and 'Build Executor Status'. The main area is titled 'Create new item' with a sub-section for 'Item name' set to '01-PULL-CODE'. A list of job types is shown with 'Freestyle project' selected, followed by 'Maven project', 'Pipeline', 'External Job', and 'Folder'.

New Item

Item name

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any kind of build step for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces configuration.

Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building complex systems or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, so you can use Jenkins as a dashboard of your existing automation system. See [the documentation](#).

Folder
Creates a container that stores nested items in it. Useful for grouping things together under a single namespace, so you can have multiple things of the same name as long as they are in different folders.



Configure job

The screenshot shows the Jenkins configuration interface for a job named "01-PULL-CODE". The "General" tab is selected. The "Project name" field contains "01-PULL-CODE". The "Description" field is empty. Below these fields is a rich text editor area with "[Plain text]" and "[Preview]" buttons. A list of checkboxes follows:

- Discard Old Builds
- GitHub project
- This build is parameterized
- Throttle builds
- Disable Build (No new builds will be executed until the project is re-enabled.)
- Execute concurrent builds if necessary
- Restrict where this project can be run

On the right side of the checkboxes, there are seven blue circular icons with question marks, each corresponding to one of the checked options. At the bottom left of the configuration panel are two buttons: "Save All" and "Apply All".



Source code management

General **Source Code Management** Build Triggers Build Environment Bindings Build Post-build Actions

Source Code Management

None
 Git

Repositories

Repository URL X ?
Please enter Git repository.

Credentials

Branches to build

Branch Specifier (blank for 'any') X ?

Repository browser

Additional Behaviours

Subversion



Build triggers

General Source Code Management **Build Triggers** Build Environment Bindings Build Post-build Actions

Repository browser (Auto) ?

Additional Behaviours Add ?

Subversion

Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- Build when a change is pushed to GitHub ?
- Poll SCM ?

Build Environment

- Delete workspace before build starts
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Use secret text(s) or file(s) ?

Save All **Apply All**



Build triggers with poll SCM

Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- Build when a change is pushed to GitHub ?
- Poll SCM ?

Schedule

*****|



⚠ Do you really mean "every minute" when you say "***"? Perhaps you meant "H * * * *" to poll once per hour**

Would last have run at Monday, May 9, 2016 at 9:57:10 PM Indochina Time; would next run at Monday, May 9, 2016 at 9:57:10 PM Indochina Time.

Ignore post-commit hooks



Build

General Source Code Management Build Triggers Build Environment Bindings **Build** Post-build Actions

Schedule ********* ?

⚠ Do you really mean "every minute" when you say "***"? Perhaps you meant "H * * * *" to poll once per hour**
Would last have run at Monday, May 9, 2016 at 9:57:10 PM Indochina Time; would next run at Monday, May 9, 2016 at 9:57:10 PM Indochina Time.

Ignore post-commit hooks ?

Build Environment

- Delete workspace before build starts
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Use secret text(s) or file(s)

Build

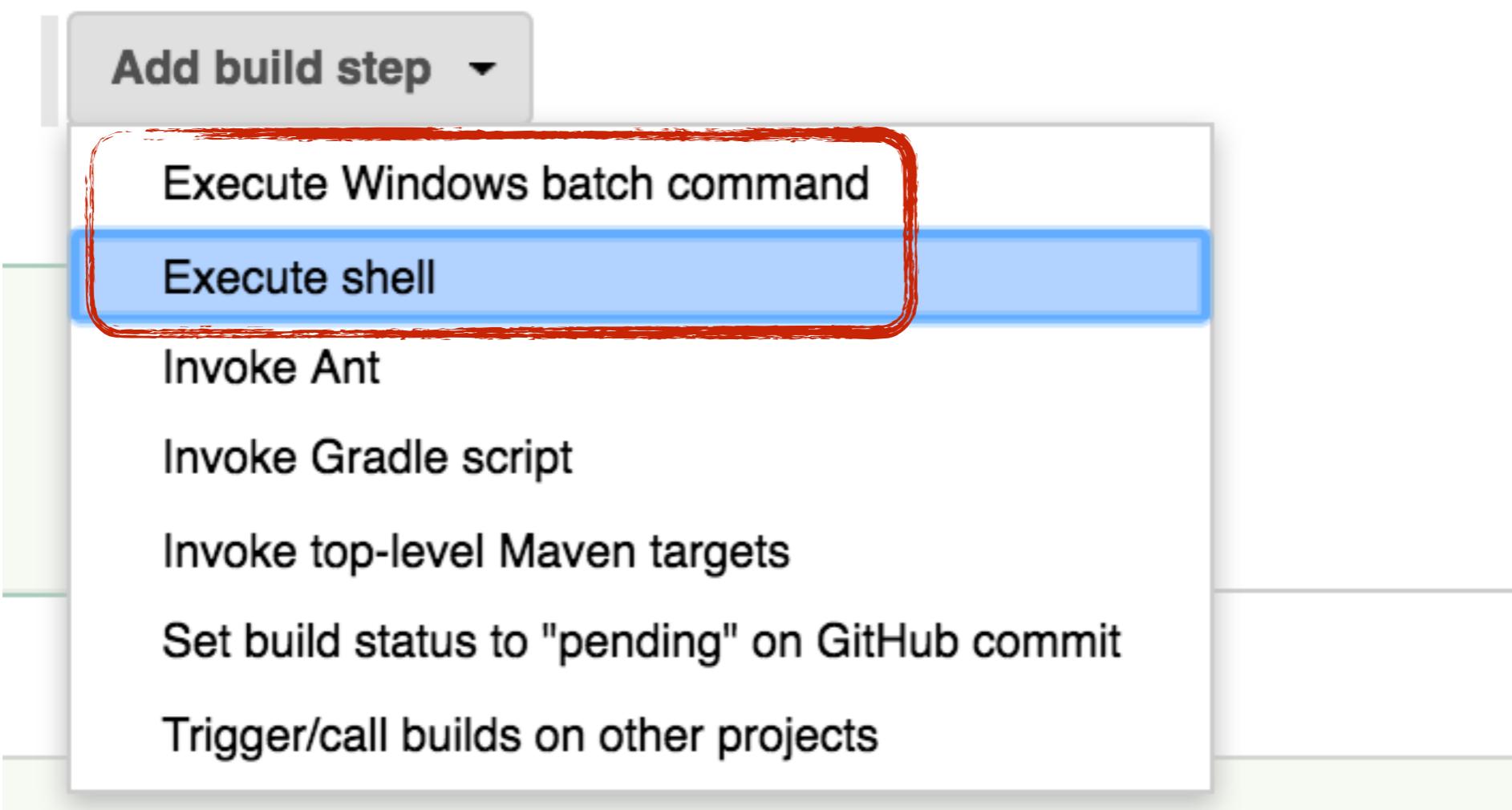
Add build step ▾

Save All **Apply All**

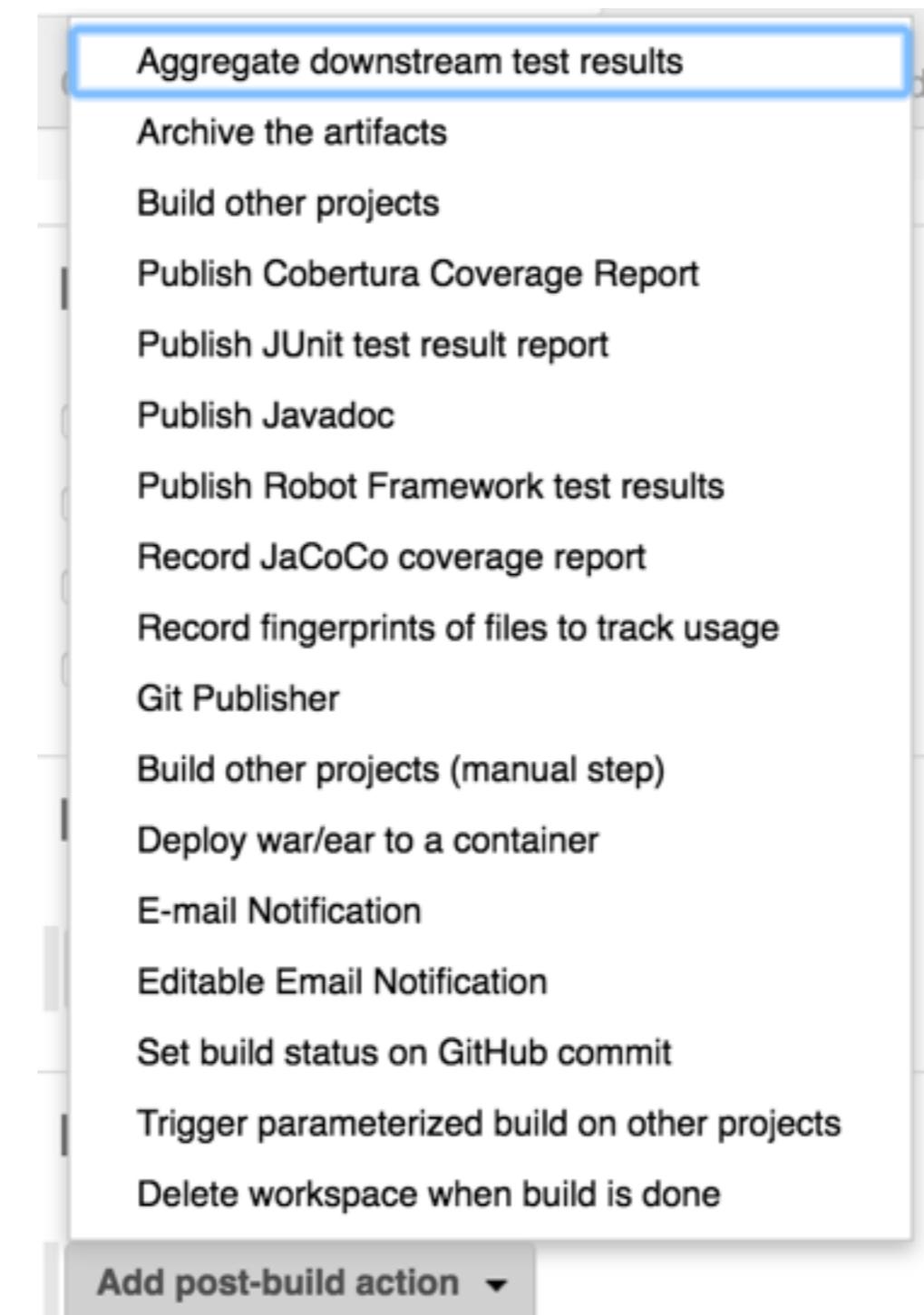


Build with command line

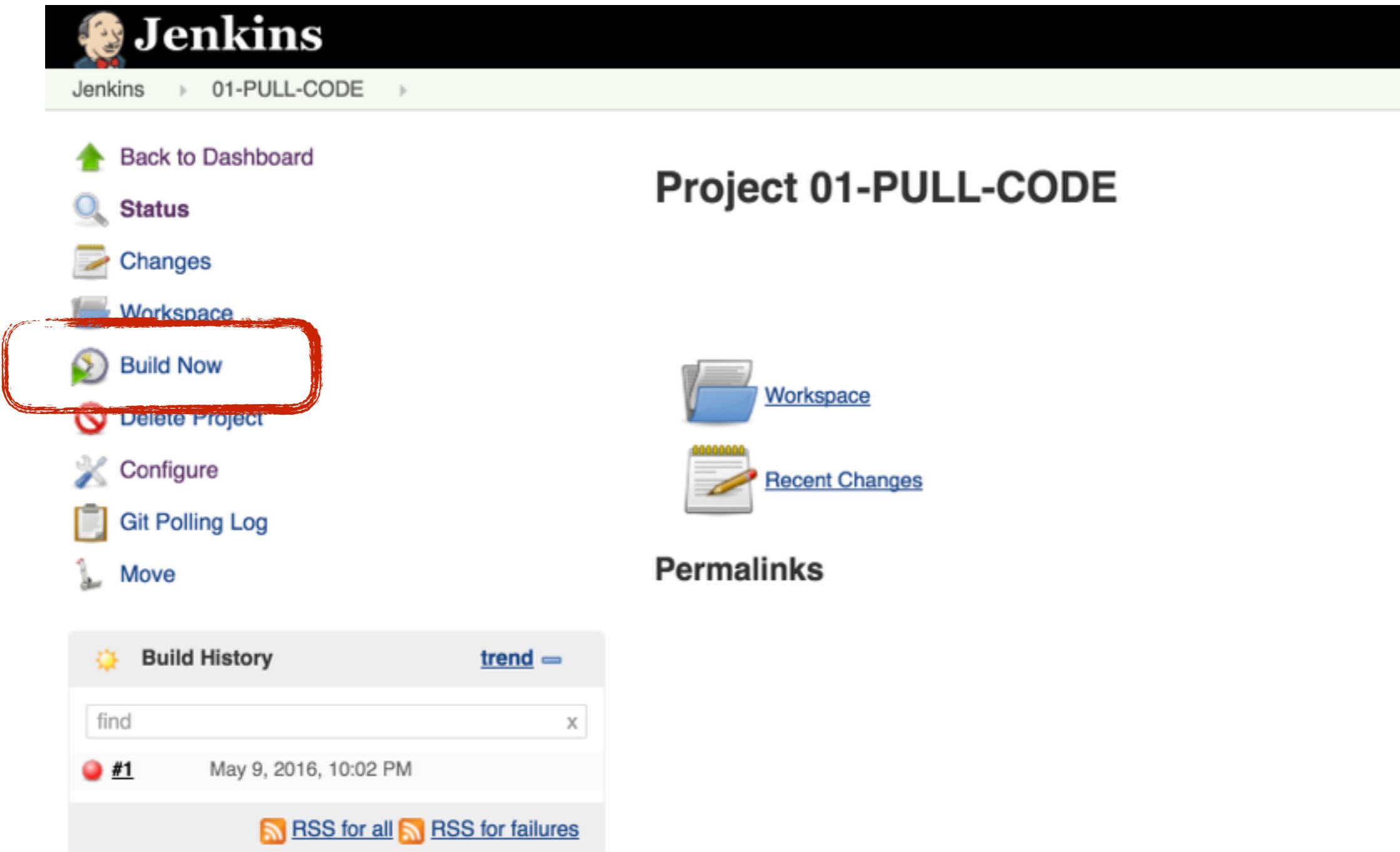
Build



Post build actions



Save and Build Now !!



Jenkins

Jenkins > 01-PULL-CODE >

- [Back to Dashboard](#)
- [Status](#)
- [Changes](#)
- [Workspace](#)
- [Build Now](#)
- [Delete Project](#)
- [Configure](#)
- [Git Polling Log](#)
- [Move](#)

Project 01-PULL-CODE

[Workspace](#)

[Recent Changes](#)

Permalinks

Build History

trend —

find

#1 May 9, 2016, 10:02 PM

RSS for all RSS for failures



ແບບຝຶກຫັດ

ອອກແບບ Build pipeline ຂອງระบบงาน



แบบฝึกหัด

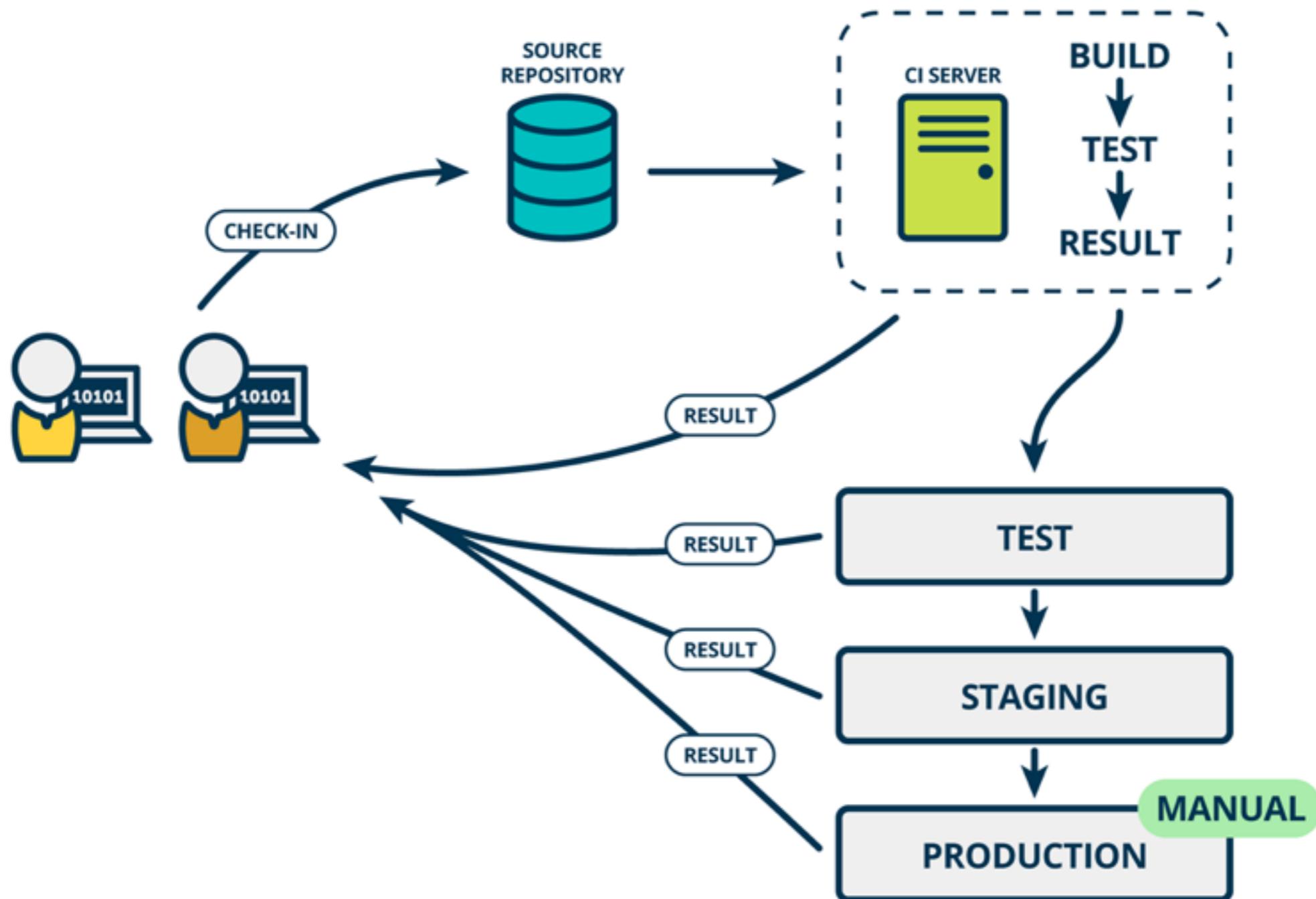
สร้าง Job ตาม build pipeline ใน Jenkins



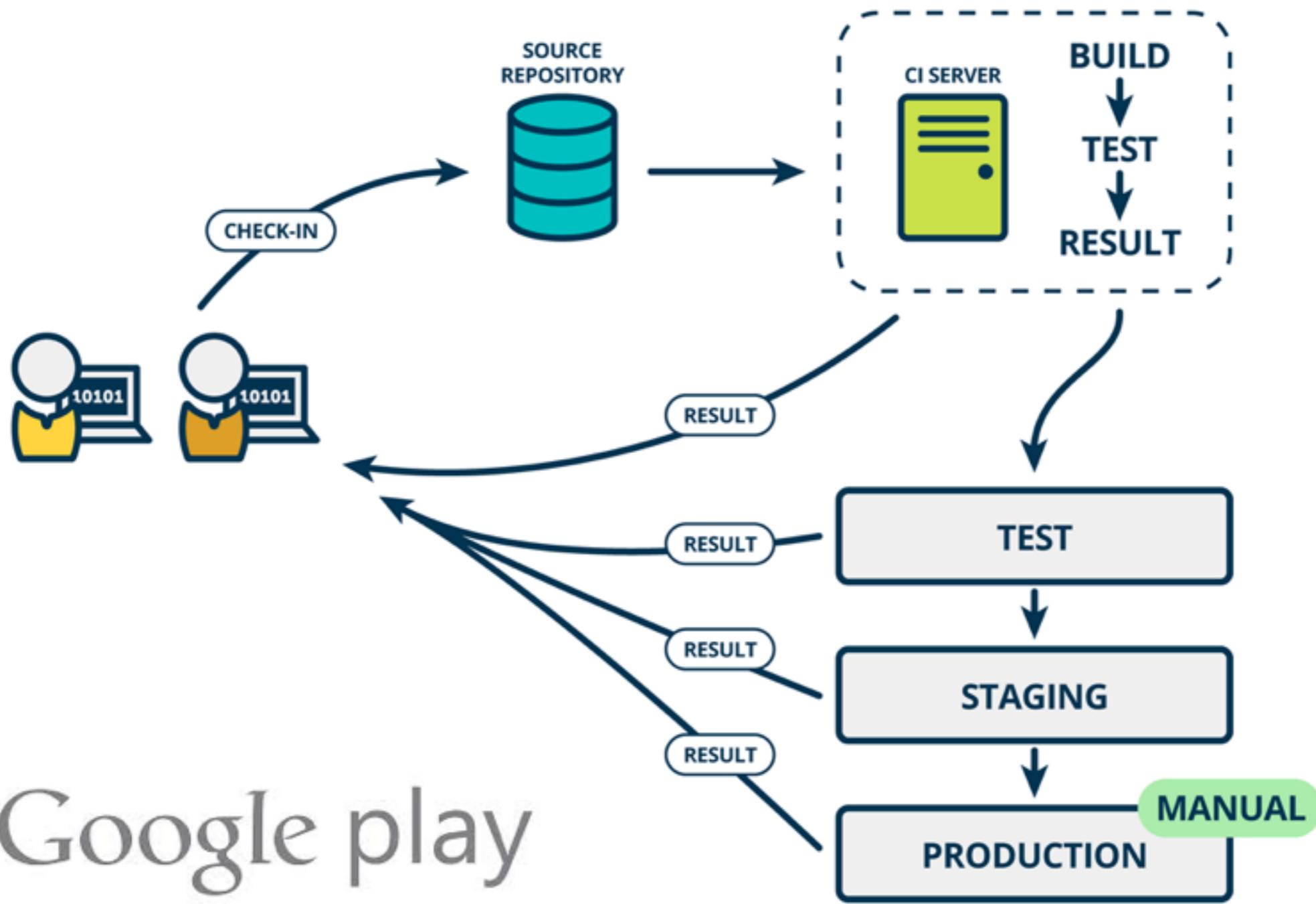
Build pipeline



Continuous Delivery



Continuous Delivery



Google play



Fastlane.tools



GitHub

Automation done right

Used by thousands of mobile developers



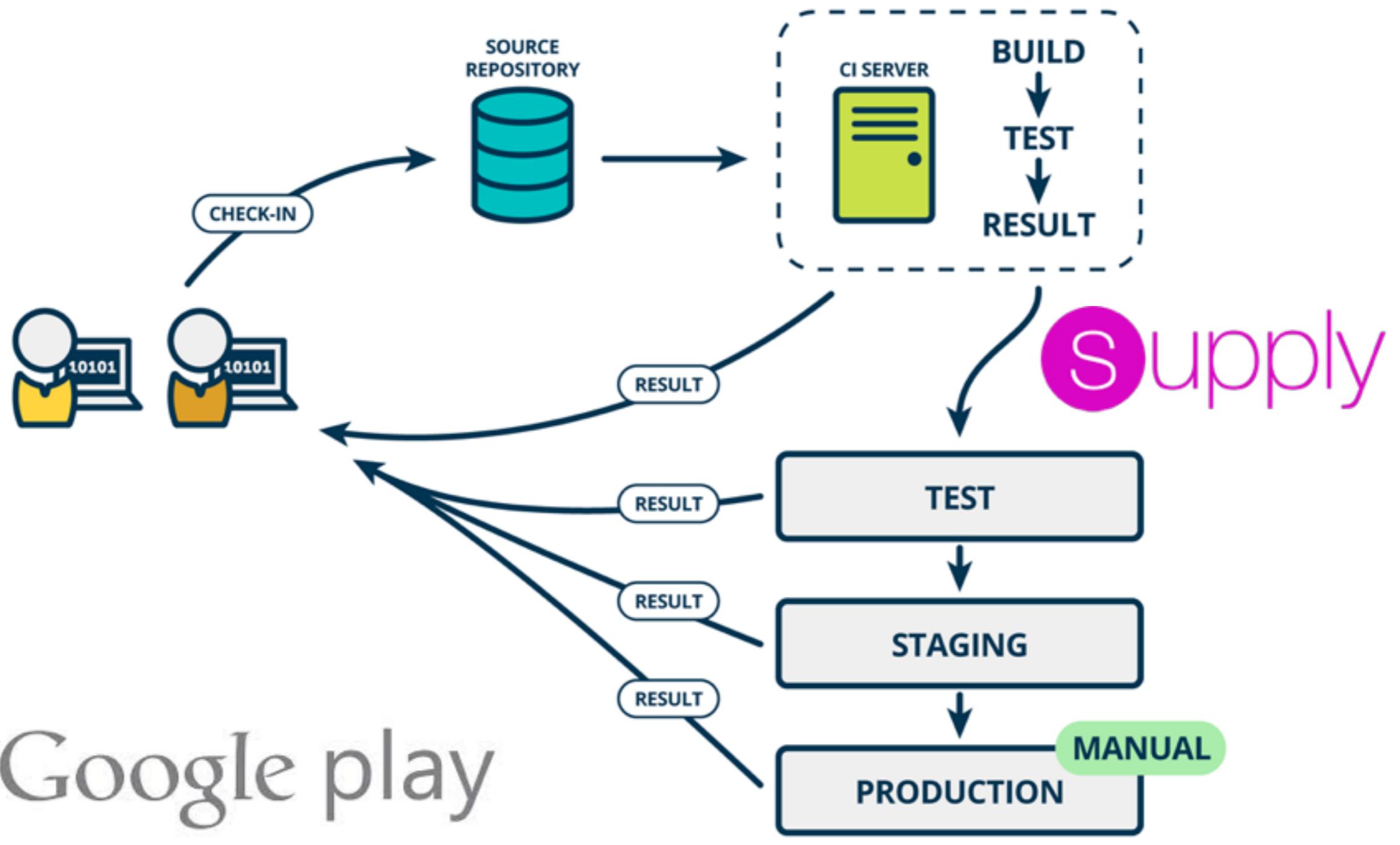
บริษัท สยามซัมนาณกิจ จำกัด และเพื่อนพ้องน้องพี่

Installation

```
$sudo gem install fastlane --verbose
```



Continuous Delivery



Workshop with Supply

สมัคร Developer account

สร้าง Service account สำหรับ publish api

สร้าง Service account key

Upload APK to Google Play



Create service account

<https://developers.google.com/android-publisher/>

The screenshot shows the 'API ACCESS' section of the Google Play Developer Console. On the left, there's a sidebar with icons for different settings: Account details, User accounts & rights, Activity log, Email notifications, API access (which is selected and highlighted in blue), Linked accounts, Developer page, and Cloud Test Lab. The main content area has a heading 'API ACCESS' with a sub-section about the Google Play Developer Publishing API. It includes a note on security regarding API users having similar access to the console. Below this is a 'LINKED PROJECT' section showing 'Google Play Android Developer' linked to the project, with 'Unlink' and 'On' buttons. Another entry for 'Games Services Publishing API' is also listed.

Google Play Developer Console

SETTINGS

- Account details
- User accounts & rights
- Activity log
- Email notifications
- API access**
- Linked accounts
- Developer page
- Cloud Test Lab

API ACCESS

The Google Play Developer Publishing API lets you publish and configure your apps from your own automated tools and processes. [Learn more](#)

Note on security: API users have access to perform actions similar to those available through the Google Play Developer Console. API users must be managed with the same care as other Google Play developer console access credentials. Rights also apply to API requests.

LINKED PROJECT

Google Play Android Developer

Games Services Publishing API

Unlink

On



Create service account

SERVICE ACCOUNTS

Service accounts allow access to the Google Play Developer Publishing API on behalf of an application rather than accessing the API from an unattended server, such as an automated build server (e.g. Jenkins). All actions made by a service account are tracked.

You can configure fine grained permissions for the service account on the 'User Accounts & Rights' page.

EMAIL	PERMISSIONS
@developer.gserviceaccount.com	<button>Grant access</button>

Create Service Account



Create service account key

API key

Identifies your project using a simple API key to check quota and access.
For APIs like Google Translate.

OAuth client ID

Requests user consent so your app can access the user's data.
For APIs like Google Calendar.

Service account key

Enables server-to-server, app-level authentication using robot accounts.
For use with Google Cloud APIs.

Help me choose

Asks a few questions to help you decide which type of credential to use.

[Create credentials ▾](#)



Create service account key

Create service account key

Service account

New service account

Service account name 

fastlane-01

Service account ID

fastlane-01 @api-5228745590174066006-637549.iam.gserviceaccou 

Key type

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.



Recommended



For backward compatibility with code using the P12 format

Create

Cancel



Create fastlane project

\$fastlane init



Build signed APK

`./gradlew assembleRelease`



Upload APK to Google play

Manual process



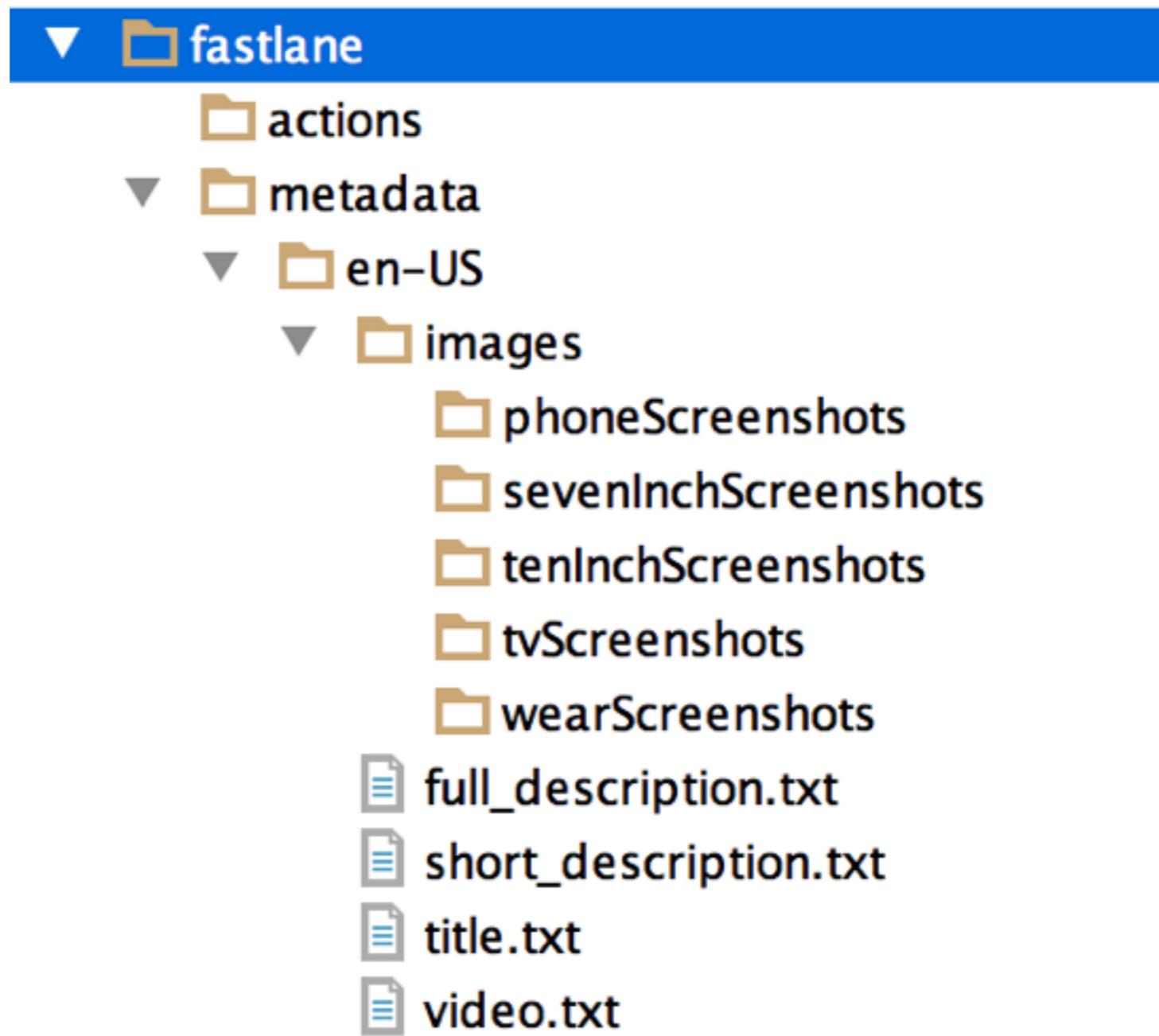
Initial supply

\$supply init

```
Writing to metadata/en-US/title.txt...
Writing to metadata/en-US/short_description.txt...
Writing to metadata/en-US/full_description.txt...
Writing to metadata/en-US/video.txt...
Due to the limit of the Google Play API `supply` can't download your existing screenshots...
Due to the limit of the Google Play API `supply` can't download your existing feature graphics...
Downloading icon for en-US...
Downloading promoGraphic for en-US...
Downloading tvBanner for en-US...
Successfully stored metadata in 'metadata'
```



Fastlane structure



Deploy to alpha

```
desc "Deploy a new version to the Google Play"
lane :deploy do
  gradle(task: "assembleRelease")
  supply(track: "alpha")
end
```



Deploy to alpha

\$fastlane deploy

Summary for supply 0.6.2	
track	alpha
package_name	demo.somkiat.demounittest
rollout	1.0
json_key	xxx-126555491470.json
apk	app/build/outputs/apk/app-release.apk
skip_upload_apk	false
skip_upload_metadata	false
skip_upload_images	false
skip_upload_screenshots	false

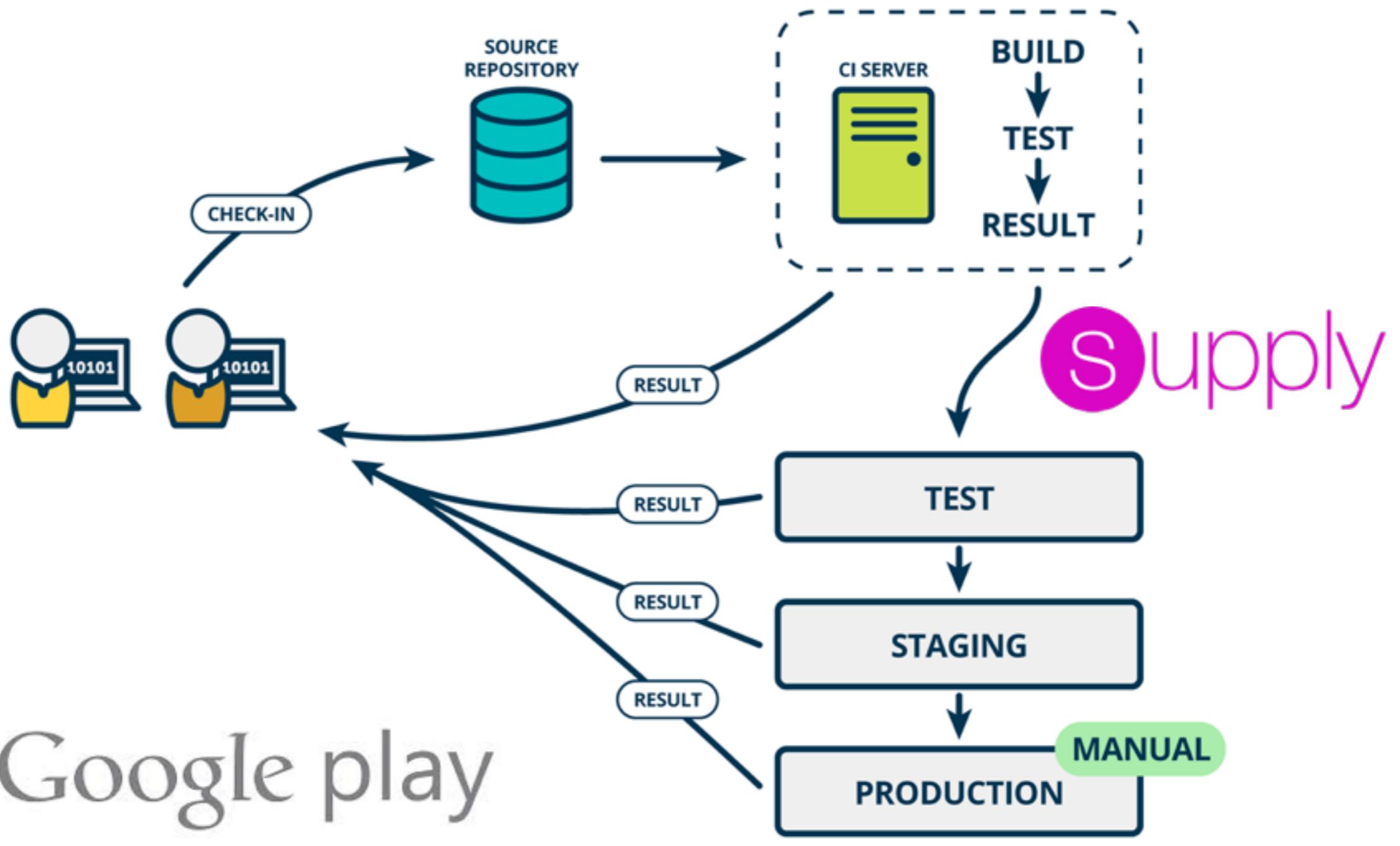
```
[12:21:33]: Preparing apk at path 'app/build/outputs/apk/app-release.apk' for upload...
[12:22:04]: Updating track 'alpha',...
[12:22:06]: Uploading all changes to Google Play...
[12:22:09]: Successfully finished the upload to Google Play
```

fastlane summary		
Step	Action	Time (in s)
1	Verifying required fastlane version	0
2	default_platform	0
3	gradle	15
4	supply	38

```
[12:22:09]: fastlane.tools finished successfully ━━
```



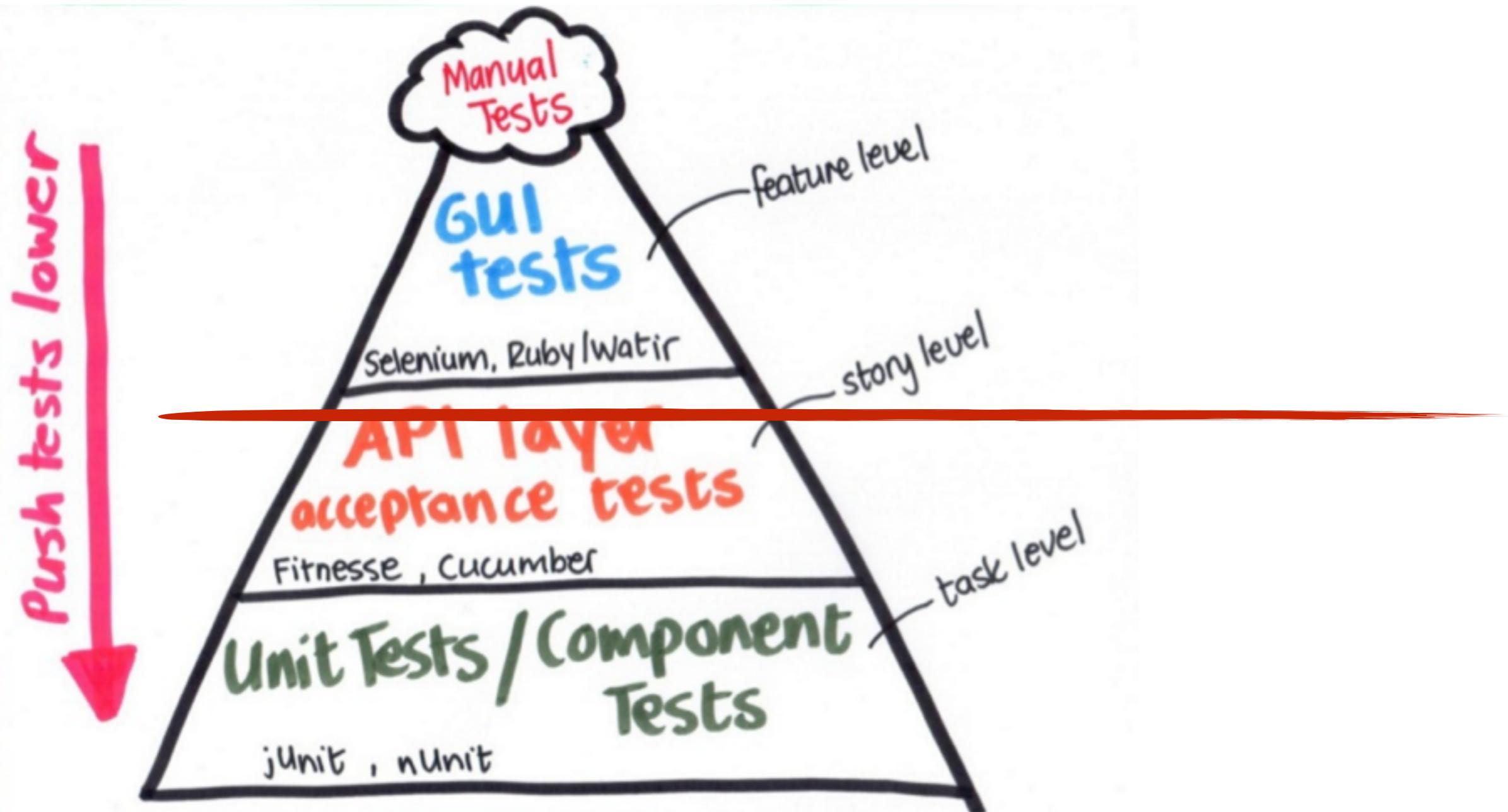
Let's fun



Mocking



Testing for android



We need small test !!

Feature	Small	Medium	Large
Network access	No	localhost only	Yes
Database	No	Yes	Yes
File system access	No	Yes	Yes
Use external systems	No	Discouraged	Yes
Multiple threads	No	Yes	Yes
Sleep statements	No	Yes	Yes
System properties	No	Yes	Yes
Time limit (seconds)	60	300	900+

<http://googletesting.blogspot.com/2010/12/test-sizes.html>



Interact with others

Classes

Shared preferences

Time

SQLite database

Network



Interact with others

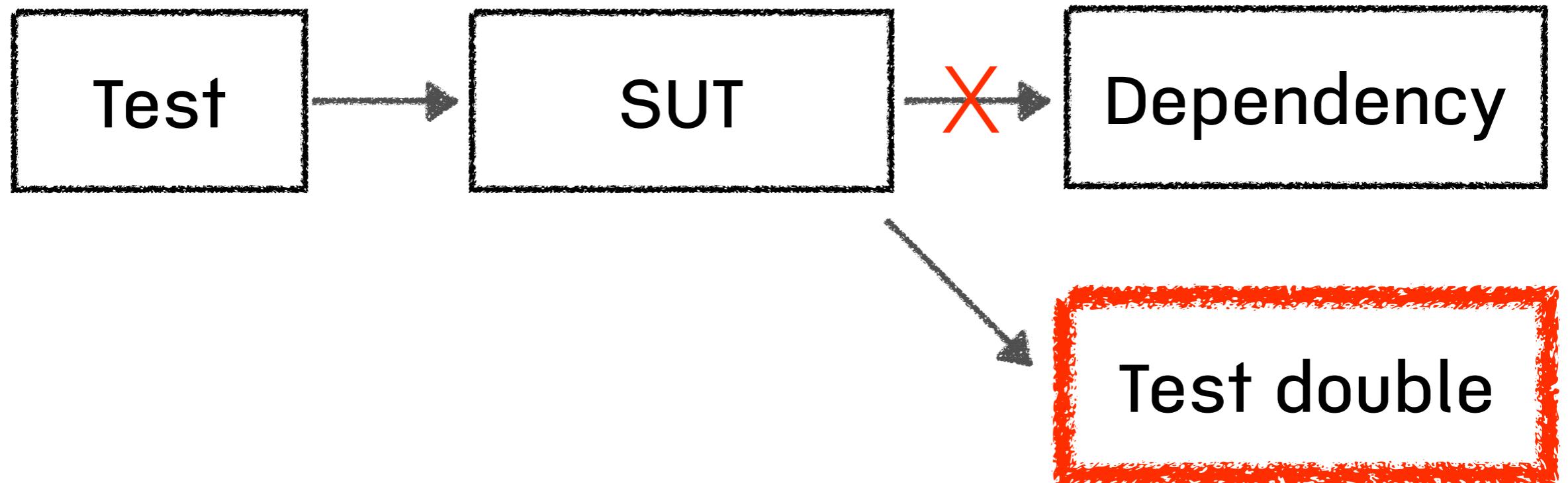
How to break dependencies ?



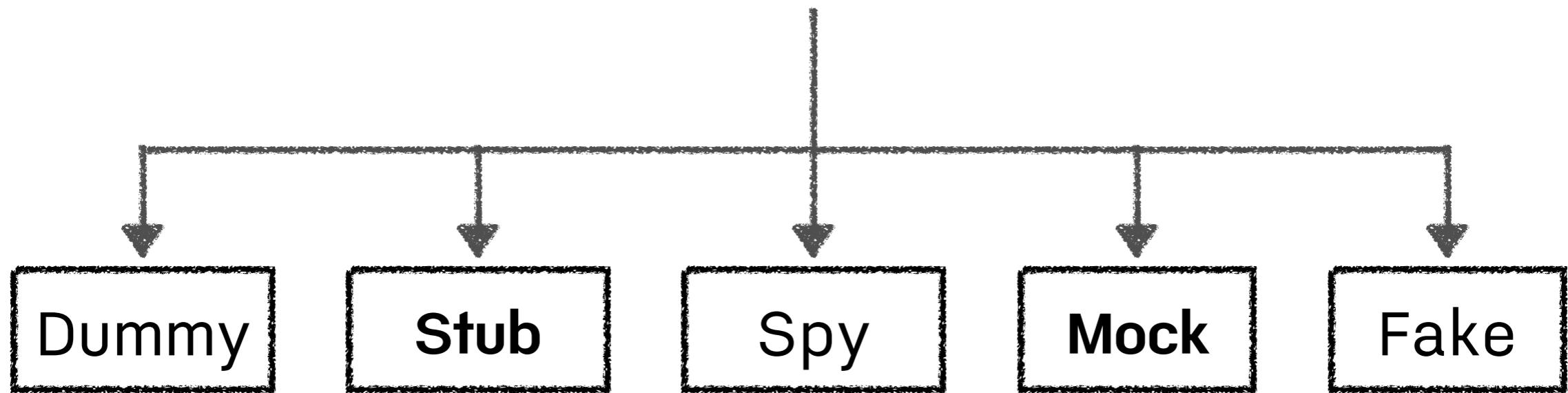
Interact with others



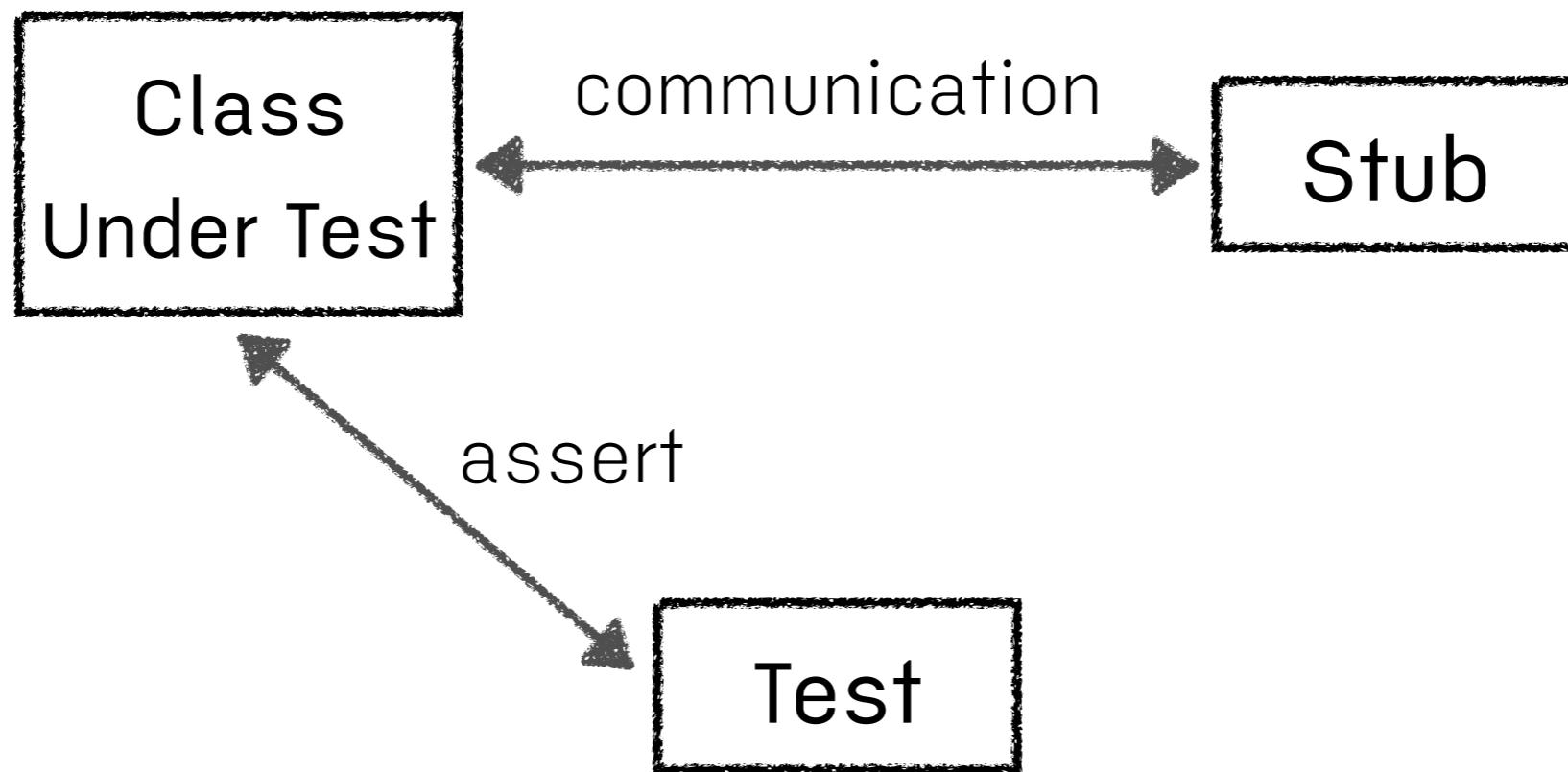
Interact with others



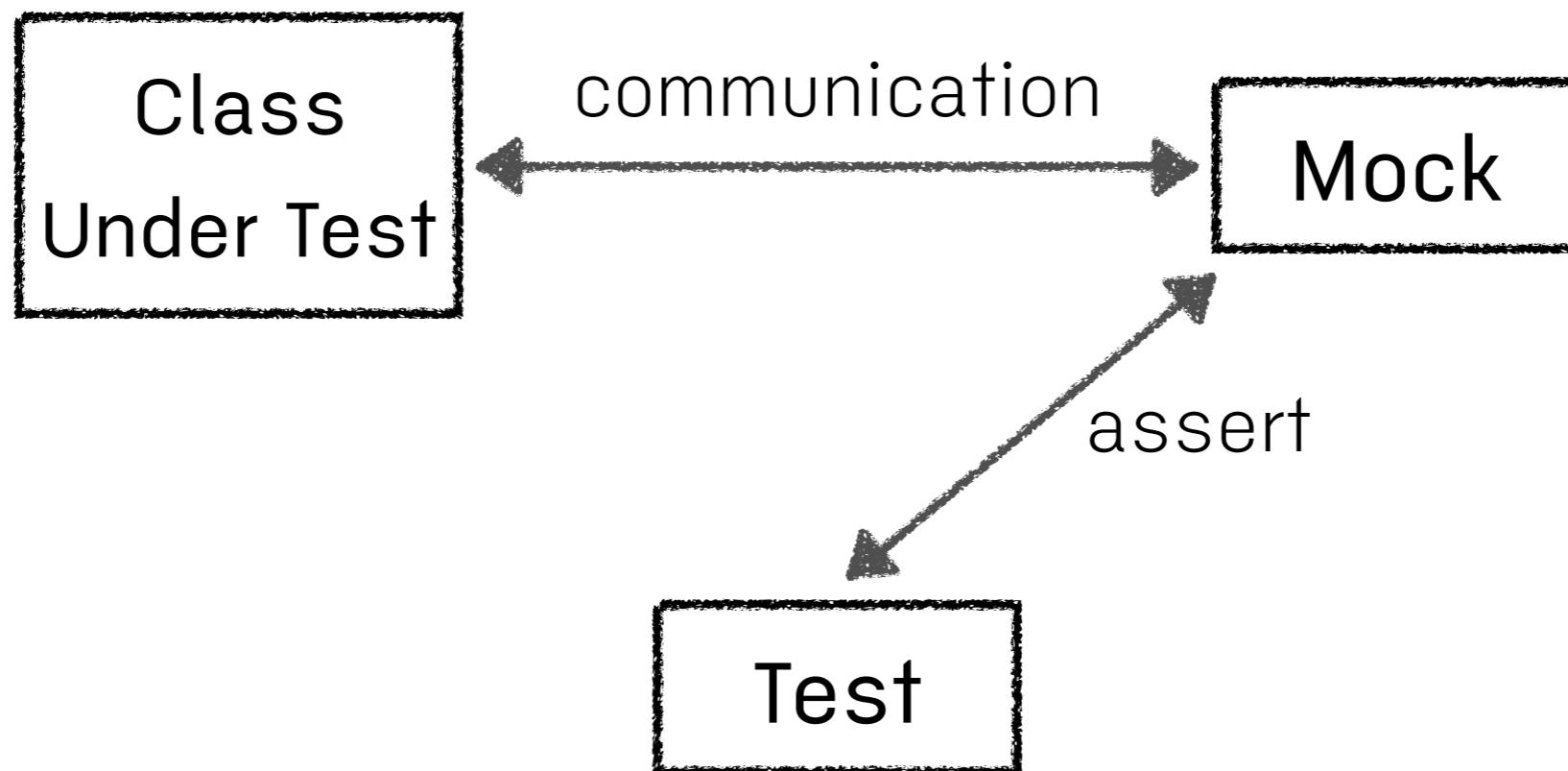
Test double



Stub



Mock



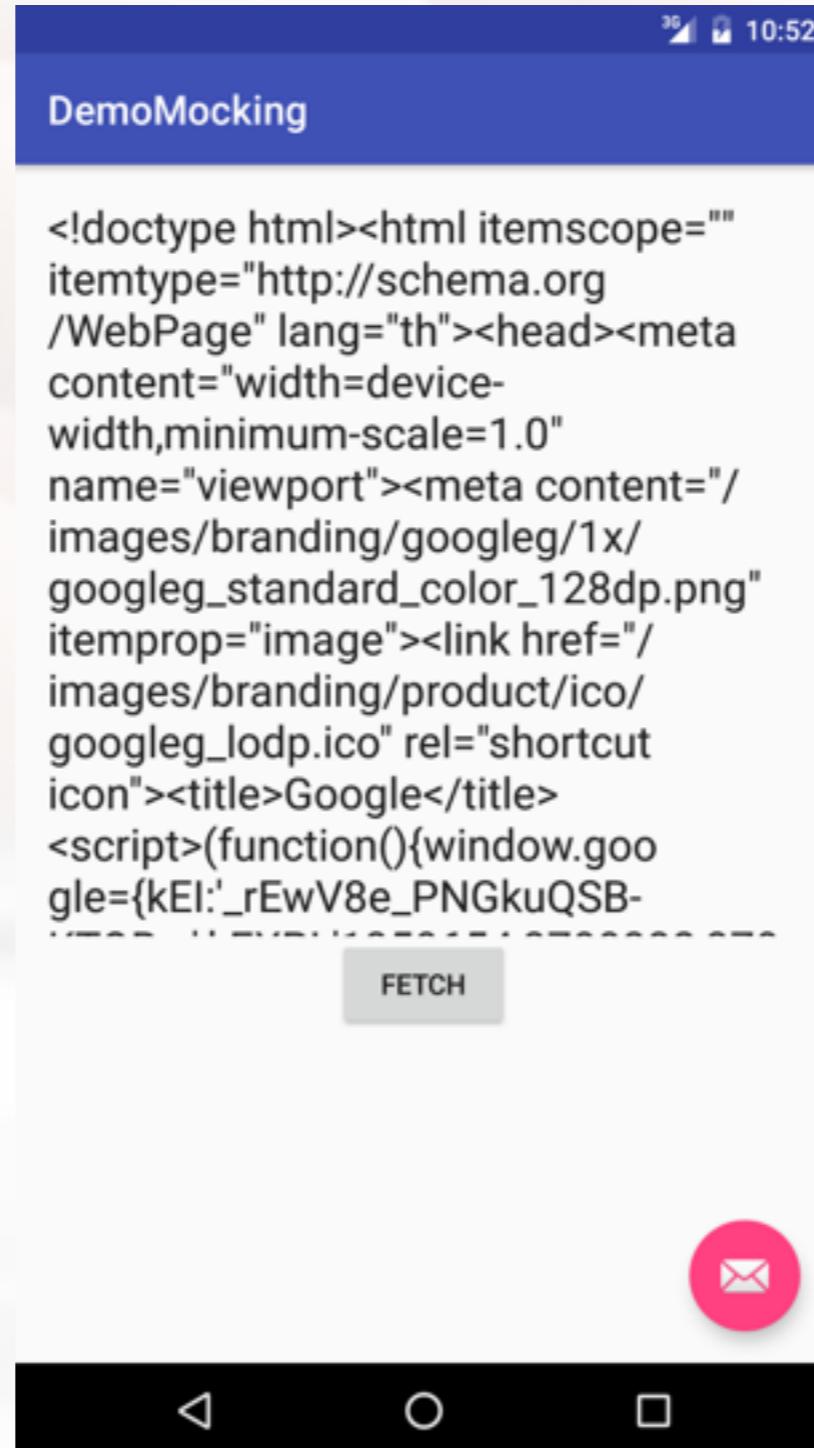
Workshop with Mocking

Network with HttpURLConnection

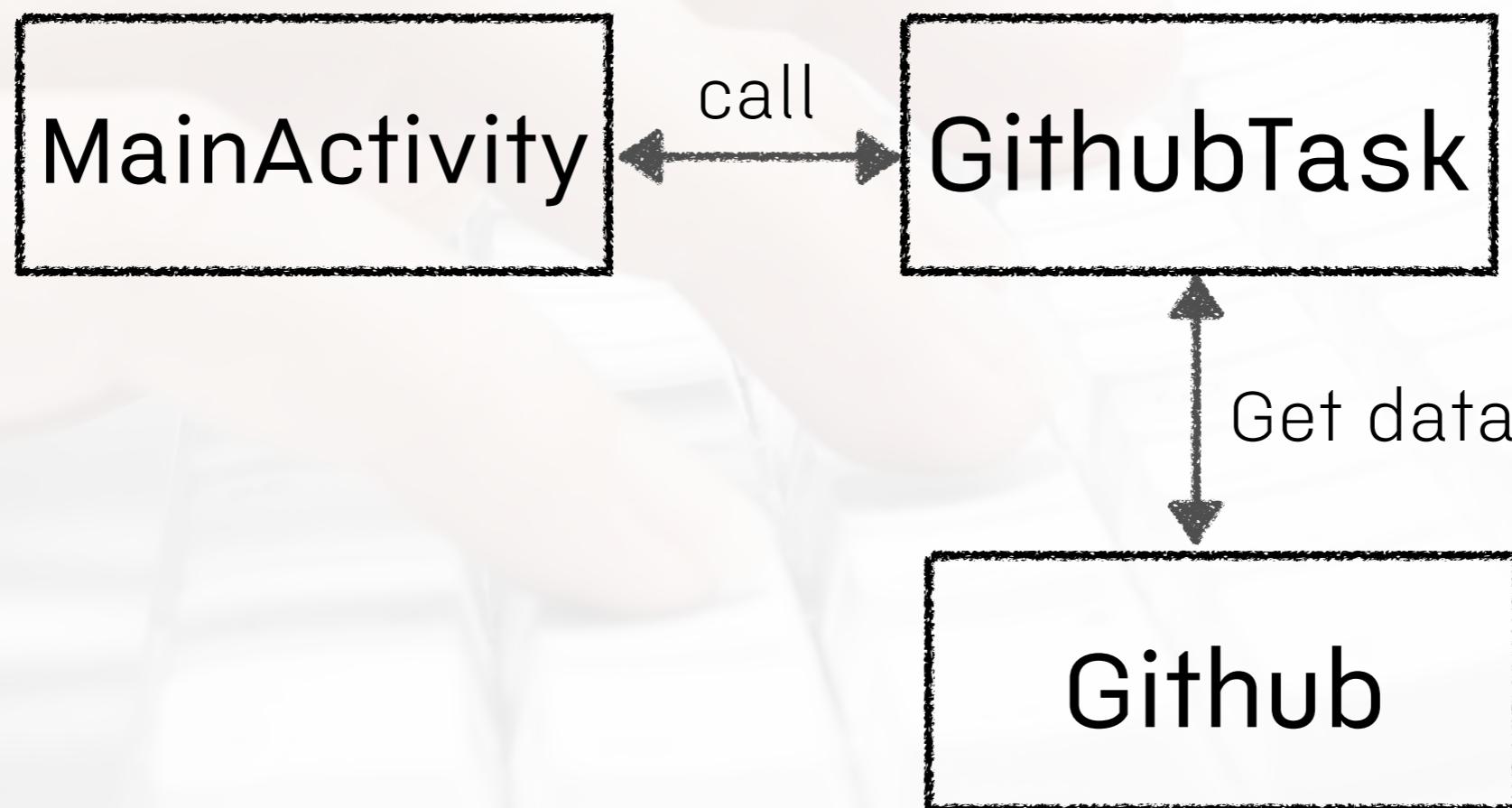
Network with Retrofit2



HttpURLConnection



HttpURLConnection



Working with Mockito

```
dependencies {  
  
    //Unit test and Mocking  
    testCompile 'junit:junit:4.12'  
    testCompile 'org.hamcrest:hamcrest-library:1.3'  
    testCompile 'org.mockito:mockito-core:1.10.19'
```



Example :: RunWith Mockito

```
@RunWith(MockitoJUnitRunner.class)
public class GithubTaskTest {

    public GitHubTask gitHubTask = Mockito.mock(GitHubTask.class);

    @Before
    public void setUp() {
        try {
            Mockito.when(gitHubTask.loadFromNetwork("http://www.google.com"))
                .thenReturn("<!doctype html><html itemscope=\"\" itemtype=\"\"");
        } catch (IOException e) {
            fail();
        }
    }

    @Test
    public void setGitHubTaskTest_ReturnsTrue() {
        try {
            assertThat(gitHubTask.loadFromNetwork("http://www.google.com"),
                containsString("doctype"));
        } catch (IOException e) {
            fail();
        }
    }
}
```



Example :: Create mock

```
@RunWith(MockitoJUnitRunner.class)
public class GithubTaskTest {

    public GitHubTask gitHubTask = Mockito.mock(GitHubTask.class);

    @Before
    public void setUp() {
        try {
            Mockito.when(gitHubTask.loadFromNetwork("http://www.google.com"))
                .thenReturn("<!doctype html><html itemscope=\"\" itemtype=\"\"");
        } catch (IOException e) {
            fail();
        }
    }

    @Test
    public void setGitHubTaskTest_ReturnsTrue() {
        try {
            assertThat(gitHubTask.loadFromNetwork("http://www.google.com"),
                containsString("doctype"));
        } catch (IOException e) {
            fail();
        }
    }
}
```



Example :: Create stub

```
@RunWith(MockitoJUnitRunner.class)
public class GithubTaskTest {

    public GitHubTask gitHubTask = Mockito.mock(GitHubTask.class);

    @Before
    public void setUp() {
        try {
            Mockito.when(gitHubTask.loadFromNetwork("http://www.google.com"))
                .thenReturn("<!doctype html><html itemscope=\"\" itemtype=\"\"");
        } catch (IOException e) {
            fail();
        }
    }

    @Test
    public void setGitHubTaskTest_ReturnsTrue() {
        try {
            assertThat(gitHubTask.loadFromNetwork("http://www.google.com"),
                containsString("doctype"));
        } catch (IOException e) {
            fail();
        }
    }
}
```



Example :: Write test

```
@RunWith(MockitoJUnitRunner.class)
public class GithubTaskTest {

    public GitHubTask gitHubTask = Mockito.mock(GitHubTask.class);

    @Before
    public void setUp() {
        try {
            Mockito.when(gitHubTask.loadFromNetwork("http://www.google.com"))
                .thenReturn("<!doctype html><html itemscope=\"\" itemtype=\"\"");
        } catch (IOException e) {
            fail();
        }
    }

    @Test
    public void setGitHubTaskTest_ReturnsTrue() {
        try {
            assertThat(gitHubTask.loadFromNetwork("http://www.google.com"),
                containsString("doctype"));
        } catch (IOException e) {
            fail();
        }
    }
}
```

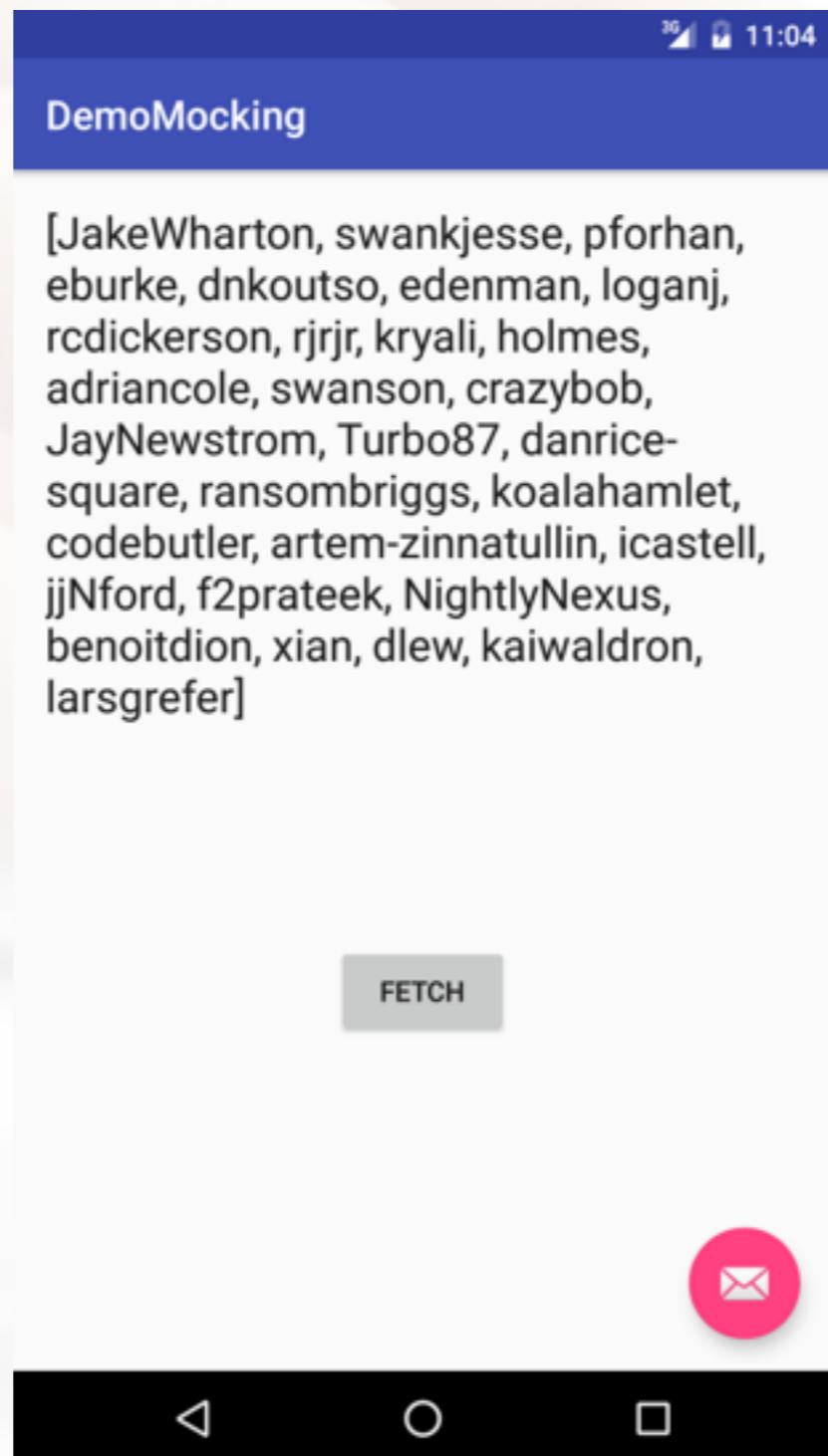


Working with hardcoded

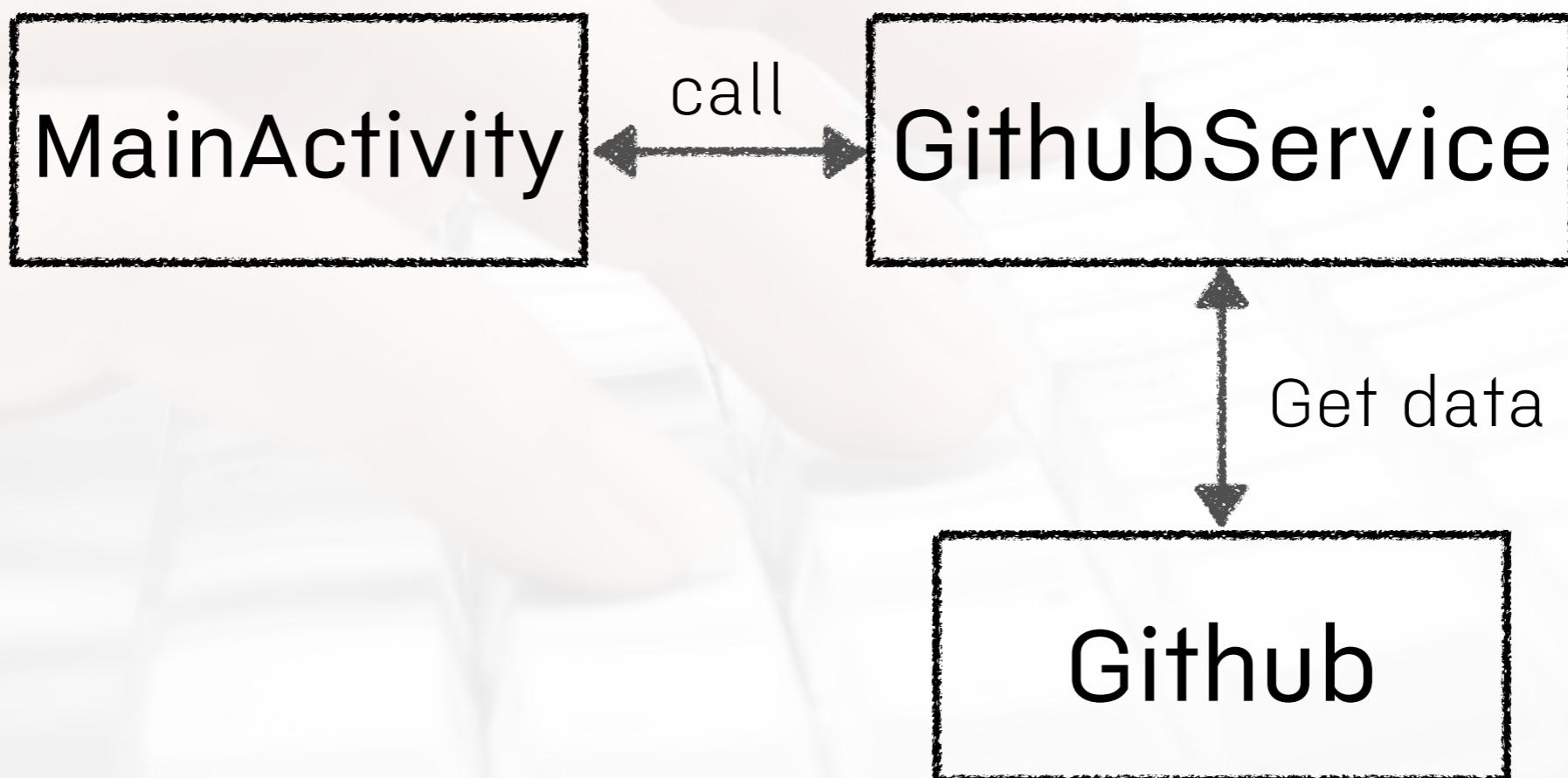
ลองทำดูสิ !!



Retrofit2



Retrofit2



Workshop with Retrofit

MockWebServer

MockRetrofit



Resources

[**https://github.com/up1/course-automated-testing-for-android-app**](https://github.com/up1/course-automated-testing-for-android-app)

