
Afprøvning



Kvalitetssikring - overblik

- Afprøvning
 - Enhedstest
 - JUnit, Android JUnit, Robotium, Roboelectric, Cucumber
 - Integrationstest
 - Accepttest
 - Performancetest
 - Abetest
- Kodedækning
 - Kan gøres med EMMA (<http://emma.sourceforge.net/>)
- Fejlrapportering
 - Stakspor - Google Play, ACRA, BugSense
 - Direkte brugerkontakt

Afprøvning - testmonkey

- adb shell monkey 100
 - laver 100 tilfældige tryk på skærm og knapper
- adb shell monkey -p dk.dr.radio -v 500
 - laver 500 tilfældige tryk på skærm og knapper på DR Radio og skriver ud hvad den gør
- Mere info

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

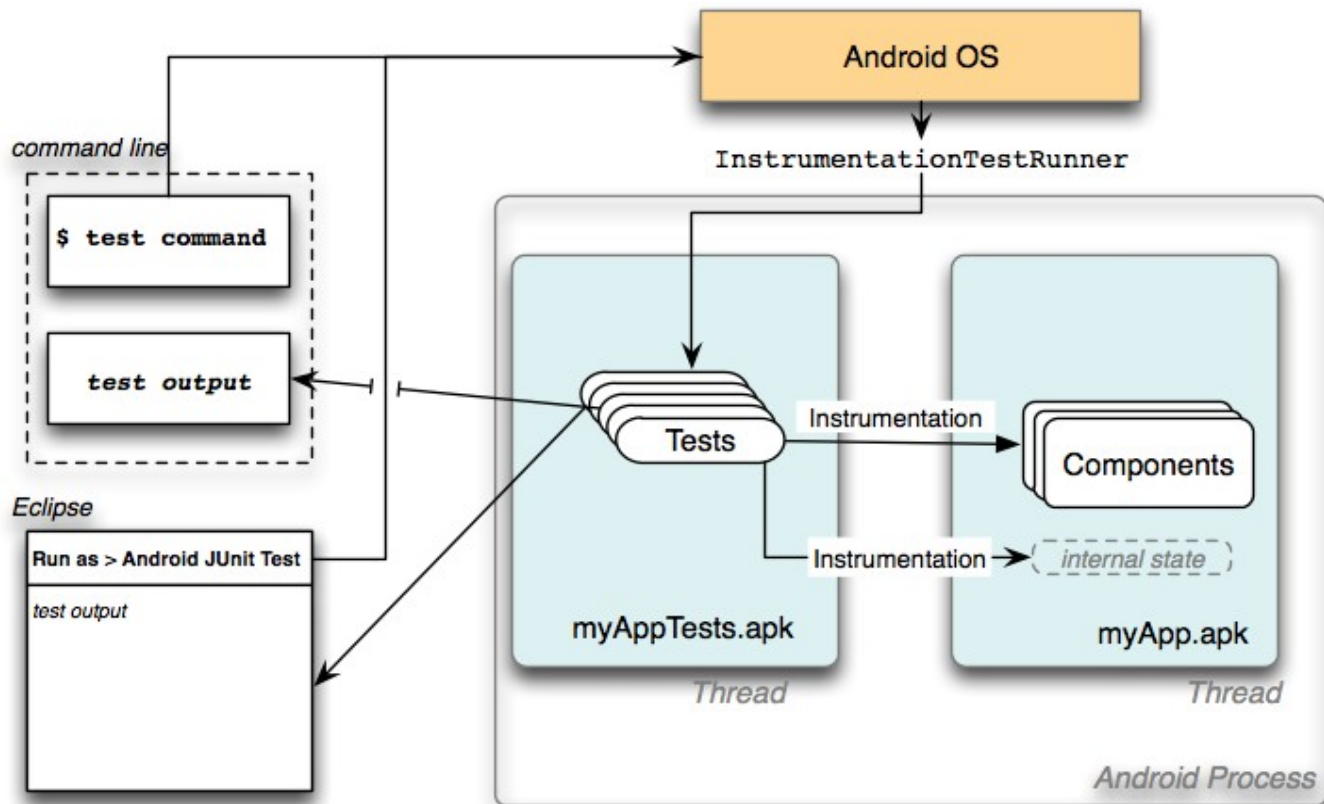
Afprøvning - Android JUnit

- Opret test-projekt der peger på projekt
- I test-projekts manifest henvises til hovedprojekt:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.example.spinner.test"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">

    <uses-library android:name="android.test.runner" />
    </application>
    <uses-sdk android:minSdkVersion="3" />
    <instrumentation
        android:targetPackage="com.android.example.spinner"
        android:name="android.test.InstrumentationTestRunner" />
</manifest>
```

Afprøvning - Android JUnit



Afprøvning - Android JUnit

Eksempel på test-kode

```
import com.android.example.spinner.R;

public class SpinnerActivityTest extends
    ActivityInstrumentationTestCase2<SpinnerActivity>
{
    private SpinnerActivity mActivity;
    private Spinner mSpinner;
    private SpinnerAdapter mPlanetData;

    protected void setUp() throws Exception {
        super.setUp();
        setActivityInitialTouchMode(false);
        mActivity = getActivity();
        mSpinner = (Spinner) mActivity.findViewById(R.id.Spinner01);
        mPlanetData = mSpinner.getAdapter();
    }

    public void testSpinnerUI() {
        // get the position of the selected item
        mPos = mSpinner.getSelectedItemPosition();
        mSelection = (String)mSpinner.getItemAtPosition(mPos);
        TextView resultView = (TextView) mActivity.findViewById(R.id.SpinnerResult);
        String resultText = (String) resultView.getText();
        assertEquals(resultText,mSelection);
    }
}
```

Oprette Android JUnit testprojekt

- Pakke der skal testes: org.apertium.android
- Testpakke: org.apertium.android.test

Steps

1. Choose Project
- 2. Test Project Data**

Name and Location

Project Name:

Project Location:

Project Folder:

Set as Main Project

Tested project:

Package Name:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.apertium.android.test">
  <application>
    <uses-library android:name="android.test.runner" />
  </application>
  <instrumentation android:name="android.test.InstrumentationTestRunner"
    android:targetPackage="org.apertium.android"
    android:label="Tests for org.apertium.android"/>
```

Android JUnit i praksis

```
import org.apertium.android.App;
import org.apertium.android.TranslatorActivity;

public class TranslatorActivityTest extends
    android.test.ActivityInstrumentationTestCase2<TranslatorActivity> {
    private TranslatorActivity a;
    private Button submitButton;

    public TranslatorActivityTest() { super(TranslatorActivity.class); }

    protected void setUp() throws Exception {
        super.setUp();
        a = this.getActivity();
        submitButton = (Button) a.findViewById(R.id.translateButton);
    }

    public void testPreconditions() {
        assertNotNull(submitButton);
    }

    public void testObserversOk() {
        assertTrue(App.apertiumInstallation.observers.contains(getActivity()));
    }

    @UiThreadTest
    public void testNoPackagesStartInstallActivity() {
        Set<String> set = new HashSet<String>(App.apertiumInstallation.modeToPackage.values());
        for (String s : set) App.apertiumInstallation.uninstallPackage(s);
        App.apertiumInstallation.rescanForPackages();

        ActivityMonitor am = new Instrumentation.ActivityMonitor(
            InstallActivity.class.getName(), null, true);
        getInstrumentation().addMonitor(am);
        a.onClick(submitButton);
        assertEquals("Expected InstallActivity to be invoked", 1, am.getHits());
    }
}
```


Udfordringer i Android JUnit

- Det er langsommeligt at afprøve en app
 - Compile og pakke APK
 - Overføre og installere APK på emulator/telefon
 - Starte test
 - Føre data tilbage
- Det er kompliceret
 - Svært at teste et forløb gennem flere aktiviteter
- UI test har de sædvanlige vedligeholdelsesproblemer
- UI test overbygninger
 - Android UI Automator
 - Robotium

Afprøvning - Robotium

- Hjælpebibliotek til Android JUnit
- Solo-objekt gør instrumentering en del nemmere
- Inspireret af Firefox-testværktøjet Selenium

Afprøvning - Robotium

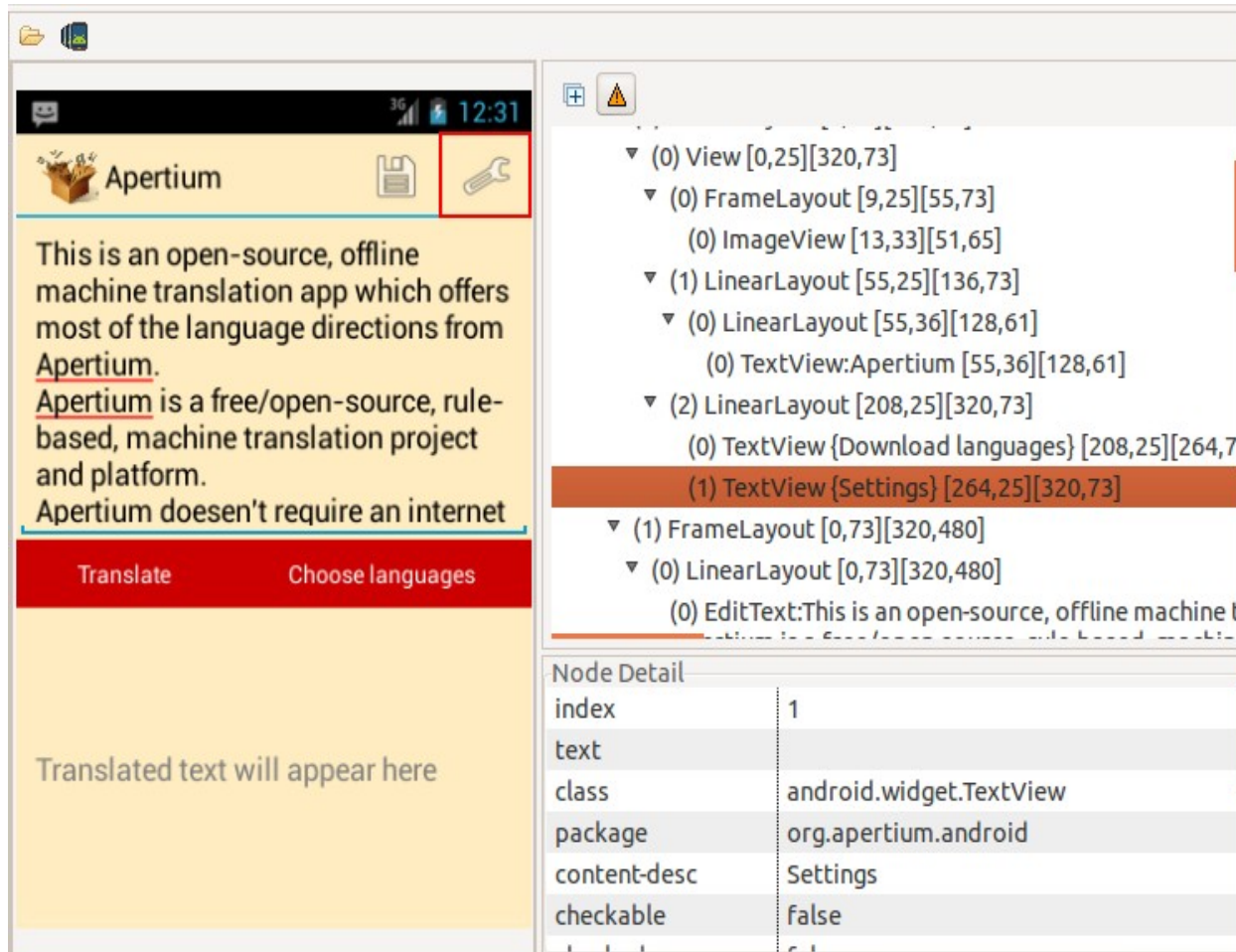
```
import android.test.ActivityInstrumentationTestCase2;
import android.widget.EditText;
import android.widget.TextView;
import com.calculator.Main;
import com.calculator.R;
import com.jayway.android.robotium.solo.Solo;
public class TestMain extends ActivityInstrumentationTestCase2<Main> {
    private Solo solo;
    public TestMain() {
        super("com.calculator", Main.class);
    }
    @Override
    protected void setUp() throws Exception {
        super.setUp();
        solo = new Solo(getInstrumentation(), getActivity());
    }

    public void testDisplayBlackBox() {
        //Enter 10 in first editfield
        solo.enterText(0, "10");
        //Enter 20 in first editfield
        solo.enterText(1, "20");
        //Click on Multiply button
        solo.clickOnButton("Multiply");
        //Verify that resultant of 10 x 20
        assertTrue(solo.searchText("200"));
    }
}
```

Android JUnit UI Automator

- Beregnet på UI afprøvning af flere aktiviteter
- Søger efter view ud fra android:hint og android:contentDescription
 - `UiObject okButton = new UiObject(new UiSelector().text("OK"));`
- Kræver enhed med Android 4.2
- Hvordan
 - Inspicér UI via 'uiautomatorviewer'
 - Skriv `UiAutomatorTestCase` testklasse
 - Kør 'uiautomator'

UI Automator Viewer



```
new UiObject(new UiSelector().text("Settings"));
```

Afprøvning - anbefalinger

- Faktorisér programlogik ud i separate pakker
- Flyt logik til *standard* Java projekt
 - Brug standard JUnit
- Udfordringer
 - Afhængighed af Android-klasser
 - Kopier android-sdk/platforms/android-1x/android.jar ind i libs/
 - android.jar har kun stubs
 - Brug Roboelectric hvis din logik afhænger af Android-biblioteker
 - Hvordan bruges standard Java projekter fra Android-projekter ?
 - Script til at kopiere jar-fil over
 - Symbolske links
 - Rette i project source paths

Afprøvning - Roboelectric

Kørsel sker i normal JVM, ikke på telefon!

```
// Test class for MyActivity
@RunWith(RobolectricTestRunner.class)
public class MyActivityTest {
    private Activity activity;
    private Button pressMeButton;
    private TextView results;

    @Before
    public void setUp() throws Exception {
        activity = new MyActivity();
        activity.onCreate(null);
        pressMeButton = (Button) activity.findViewById(R.id.press_me_button);
        results = (TextView) activity.findViewById(R.id.results_text_view);
    }

    @Test
    public void shouldUpdateResultsWhenButtonIsClicked() throws Exception {
        pressMeButton.performClick();
        String resultsText = results.getText().toString();
        assertEquals("Testing Android Rocks!", resultsText);
    }
}
```

Afprøvning - Cucumber

- Cucumber is a tool that executes plain-text functional descriptions as automated tests.
 - Ruby-baseret
- Calabash er Cucumber til Android
 - <https://github.com/calabash/calabash-android>

```
Feature: Settings feature
```

```
Scenario: As a user I can edit the settings
```

```
When I press "Settings"
```

```
Then I see "Mark the unknown words"
```


Afprøvning - Cucumber

- Opsætning
 - Windows: RubyInstaller.org
 - Linux: `sudo apt-get install rubygem`
 - `gem install calabash-android`
 - `calabash-android gen`
- Rediger .feature-fil
- Køre testen
 - `calabash-android run <apk>`

Test på 100+ fysiske Android-enheder

- Kan optage et brugsscenarium og afspille det igen
 - <http://testdroid.com/>
- Cucumber-baseret afprøvning
 - <https://www.lesspainful.com/>
 - <https://appthwack.com/>
- Alle har gratis prøveperioder