Unit and Functional Testing for the Android Platform

Christopher M. Judd

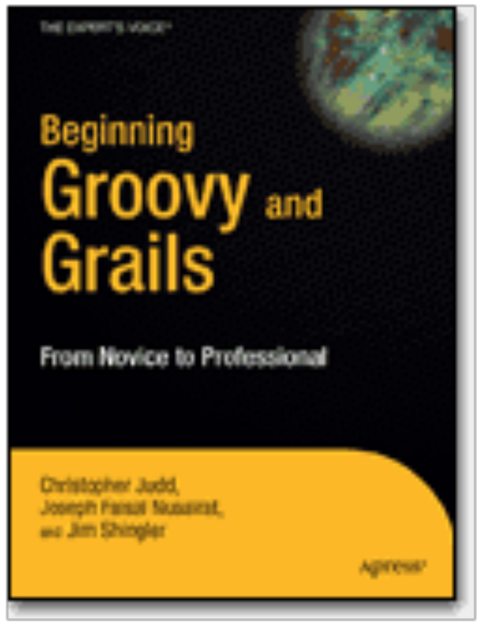
Judd Solutions

Christopher M. Judd

President/Consultant of Judd Solutions

COJUG Central Ohio Java Users Group leader

Columbus iPhone Developer User Group (CIDUG)


THE EXPERT'S VOICE

Beginning Groovy and Grails

From Novice to Professional

Christopher Judd, Joseph Faisal Nusairat, and Jim Shingler

Apress

THE EXPERT'S VOICE

Pro Eclipse JST

Plug-ins for J2EE Development

Christopher Judd and Hakeem Shittu

Apress

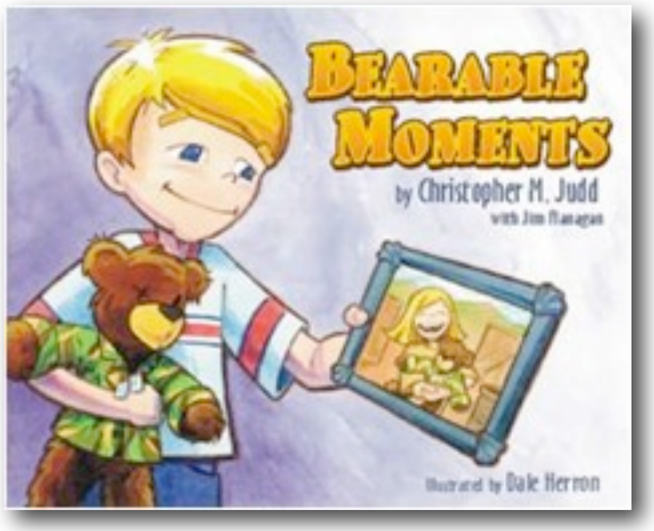
THE EXPERT'S VOICE IN JAVA

Enterprise Java Development on a Budget

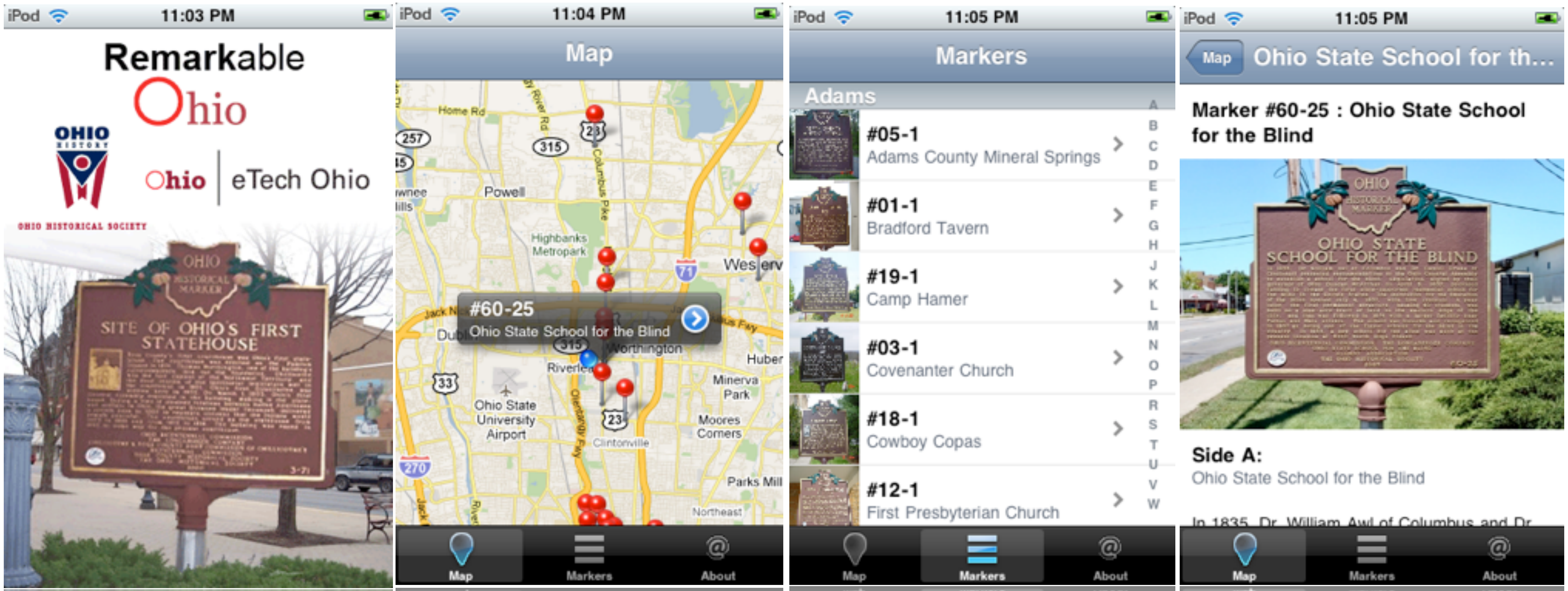
Leveraging Java Open Source Technologies

Brian Sam-Bodden and Christopher M. Judd

Apress



Remarkable Ohio



#60-25
Ohio State School for the Blind

iPod 11:05 PM

Markers

Adams

#05-1
Adams County Mineral Springs

#01-1
Bradford Tavern

#19-1
Camp Hamer

#03-1
Covenanter Church

#18-1
Cowboy Copas

#12-1
First Presbyterian Church

iPod 11:05 PM

Map Ohio State School for th...

Marker #60-25 : Ohio State School for the Blind

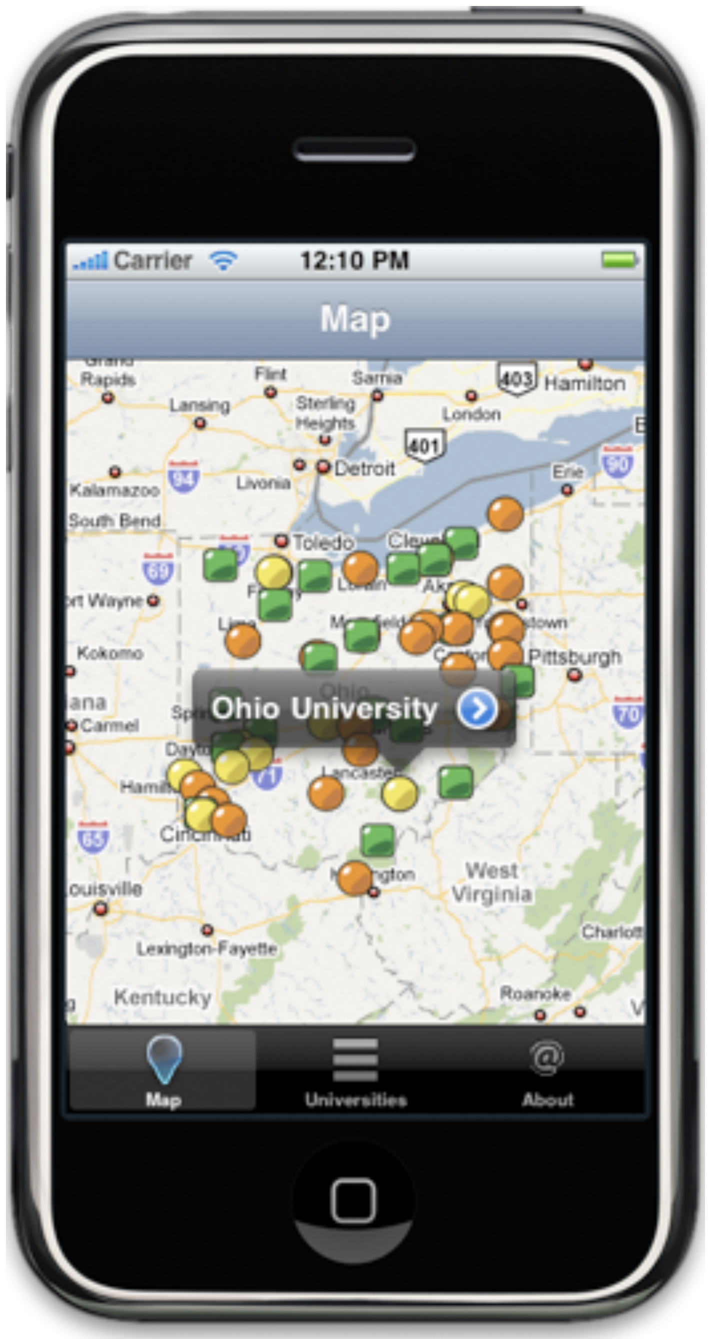
Side A:
Ohio State School for the Blind

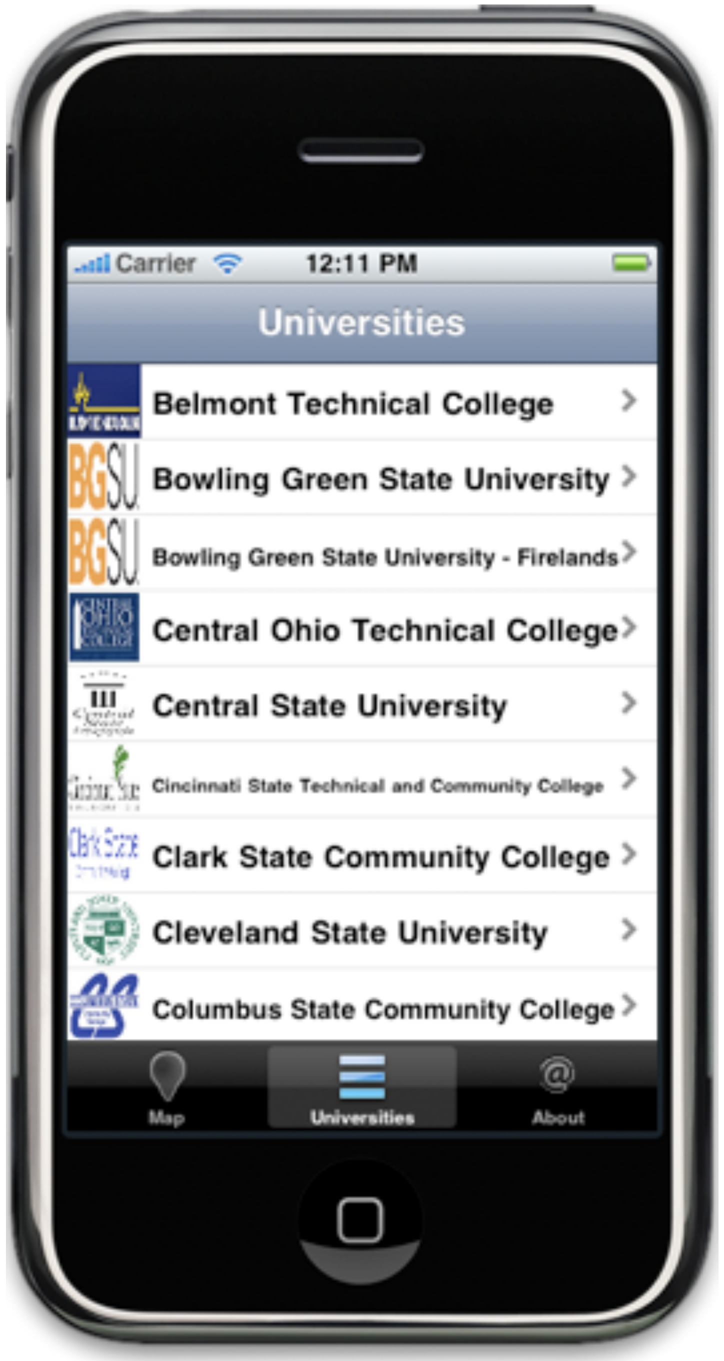
In 1835, Dr. William Awl of Columbus and Dr.

Free

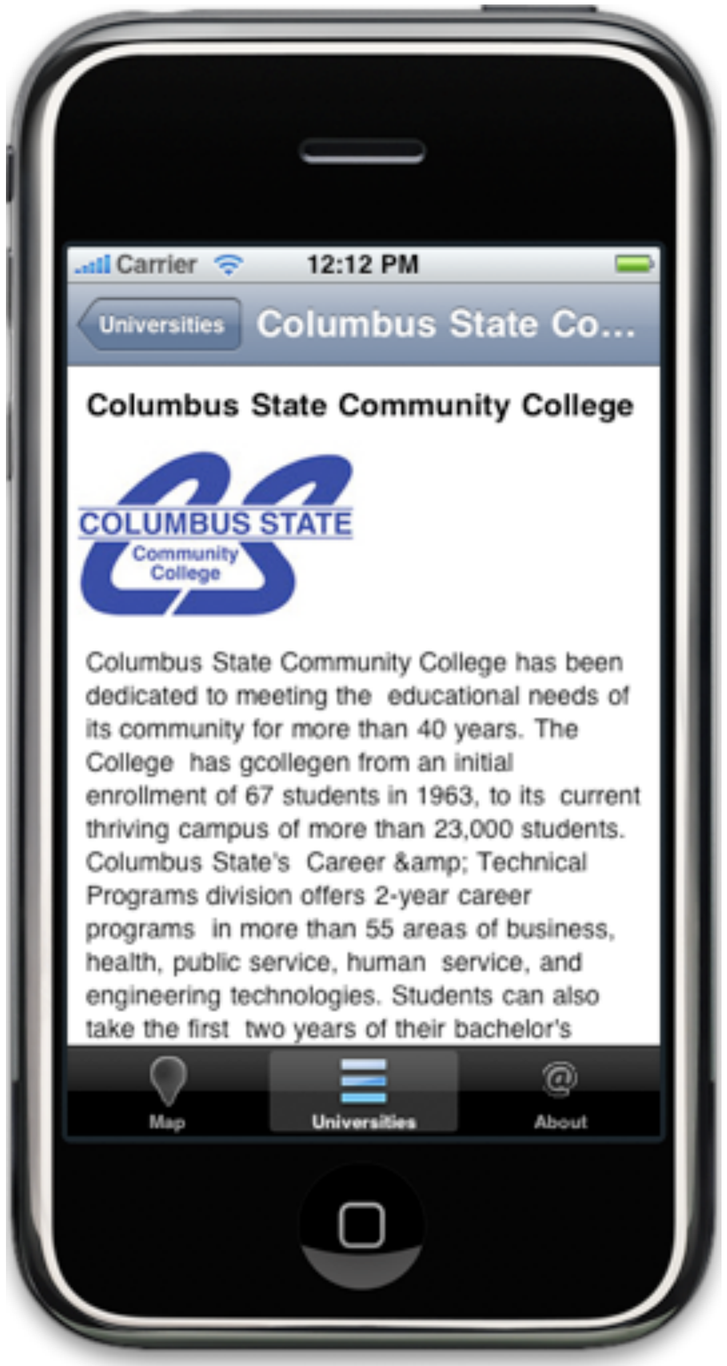
Developed for eTech Ohio and Ohio Historical Center

University System Of Ohio





- Belmont Technical College
- Bowling Green State University
- Bowling Green State University - Firelands
- Central Ohio Technical College
- Central State University
- Cincinnati State Technical and Community College
- Clark State Community College
- Cleveland State University
- Columbus State Community College



Columbus State Community College

Columbus State Community College has been dedicated to meeting the educational needs of its community for more than 40 years. The College has gcollegen from an initial enrollment of 67 students in 1963, to its current thriving campus of more than 23,000 students. Columbus State's Career &amp; Technical Programs division offers 2-year career programs in more than 55 areas of business, health, public service, human service, and engineering technologies. Students can also take the first two years of their bachelor's

Free

Developed for eTech Ohio and University System Of Ohio

How many of you are currently or have developed applications for the Android Platform?

How many of you have ever unit or functionally tested your Android application?

How many of you have ever unit tested on another platform?

WHY AREN'T YOU TESTING YOUR ANDROID APPLICATIONS?

Testing is the key to

Testing is the key to

Agility

Unit Testing

Unit Testing Basics

Why Unit Test?

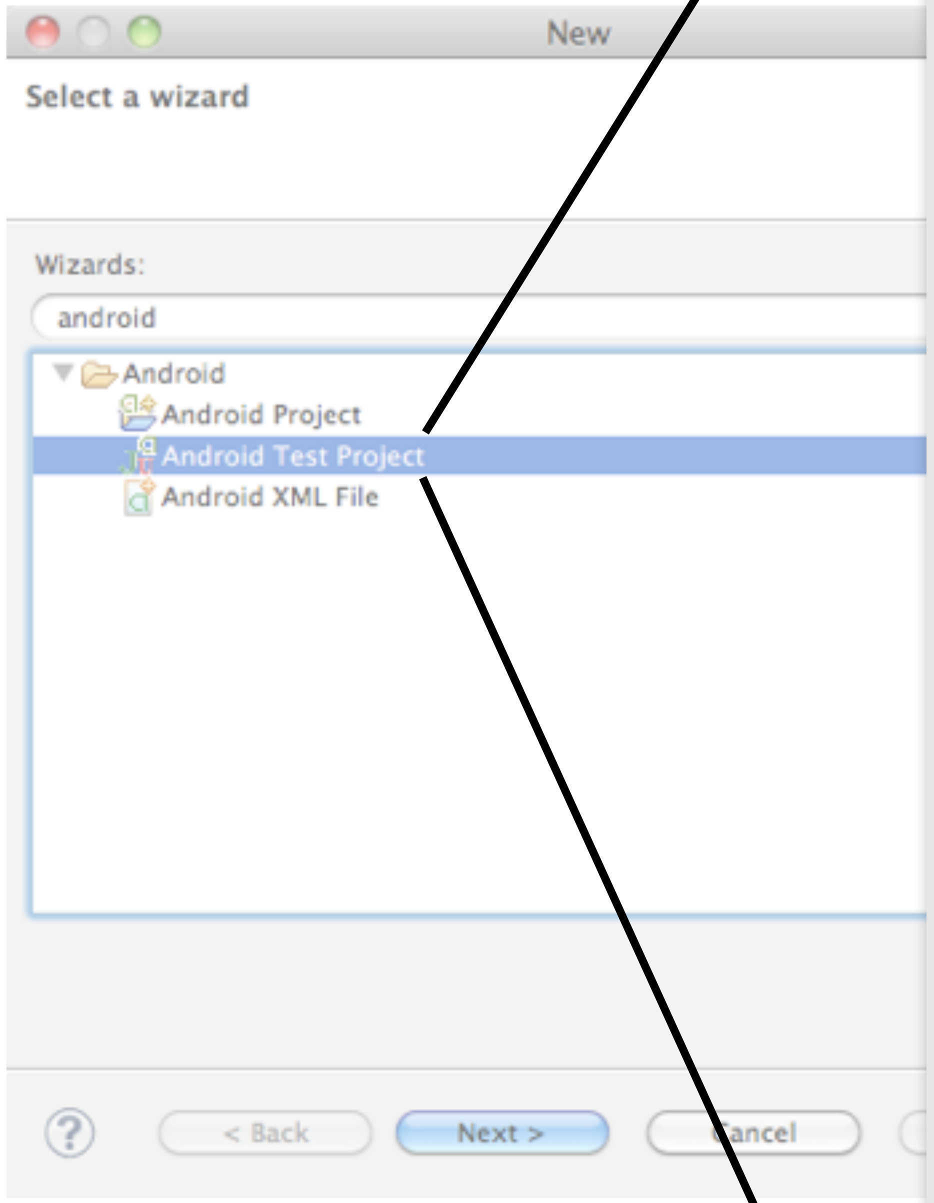
- Improves design
- Facility change and refactoring
- Simplifies integration
- Provides executable documentation

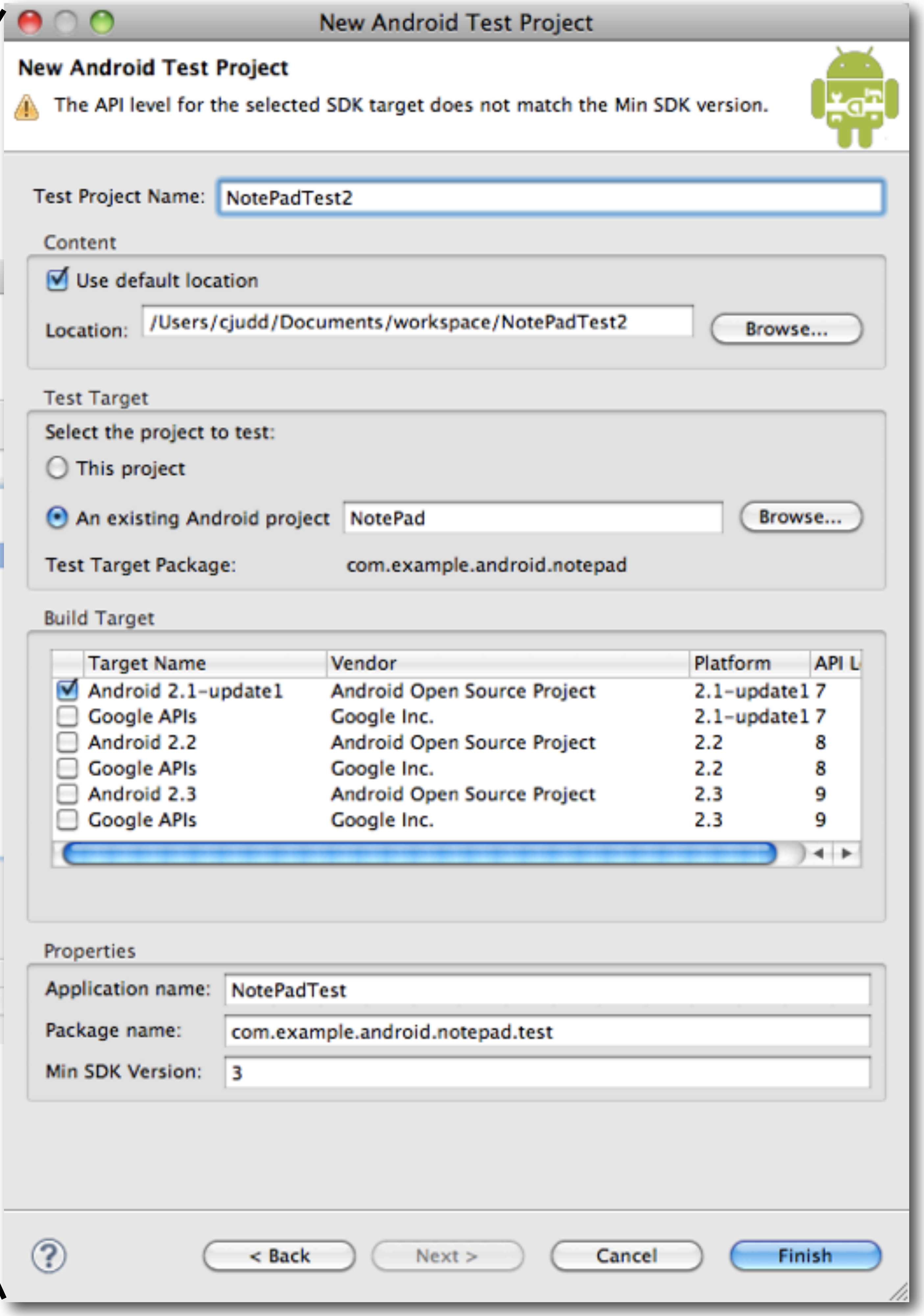
includes JUnit

Getting Started

1. Create Android Test Project

Create Android Test Project





New Android Test Project

New Android Test Project

⚠ The API level for the selected SDK target does not match the Min SDK version.

Test Project Name: NotePadTest2

Content

[✓] Use default location

Location: /Users/cjudd/Documents/workspace/NotePadTest2 Browse...

Test Target

Select the project to test:

○ This project

● An existing Android project NotePad Browse...

Test Target Package: com.example.android.notepad

Build Target

	Target Name	Vendor	Platform	API L
✓	Android 2.1-update1	Android Open Source Project	2.1-update1	7
	Google APIs	Google Inc.	2.1-update1	7
	Android 2.2	Android Open Source Project	2.2	8
	Google APIs	Google Inc.	2.2	8
	Android 2.3	Android Open Source Project	2.3	9
	Google APIs	Google Inc.	2.3	9

Properties

Application name: NotePadTest

Package name: com.example.android.notepad.test

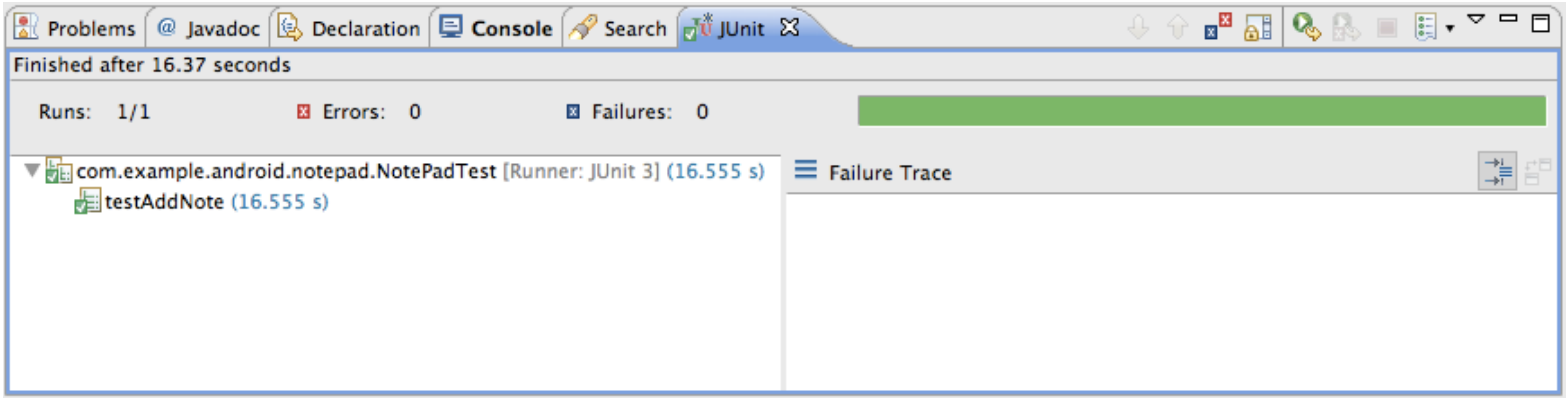
Min SDK Version: 3

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.example.android.notepad.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.example.android.notepad"
            android:name="android.test.InstrumentationTestRunner" />
    <uses-sdk android:targetSdkVersion="4" />
</manifest>
```

Running Unit Tests

Running

Run As > Android JUnit Test



Finished after 16.37 seconds

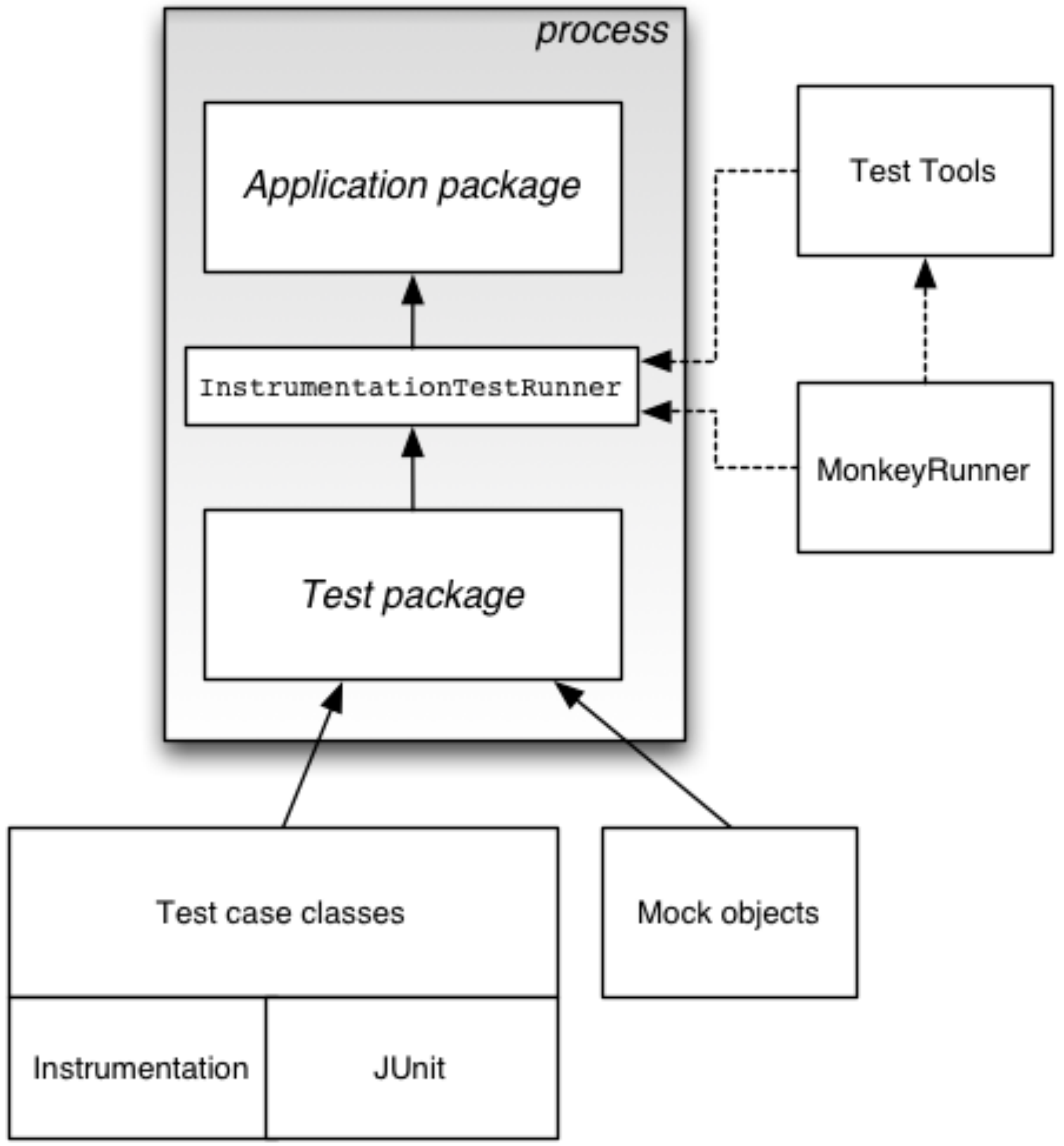
Runs: 1/1 Errors: 0 Failures: 0

com.example.android.notepad.NotePadTest [Runner: JUnit 3] (16.555 s)
testAddNote (16.555 s)

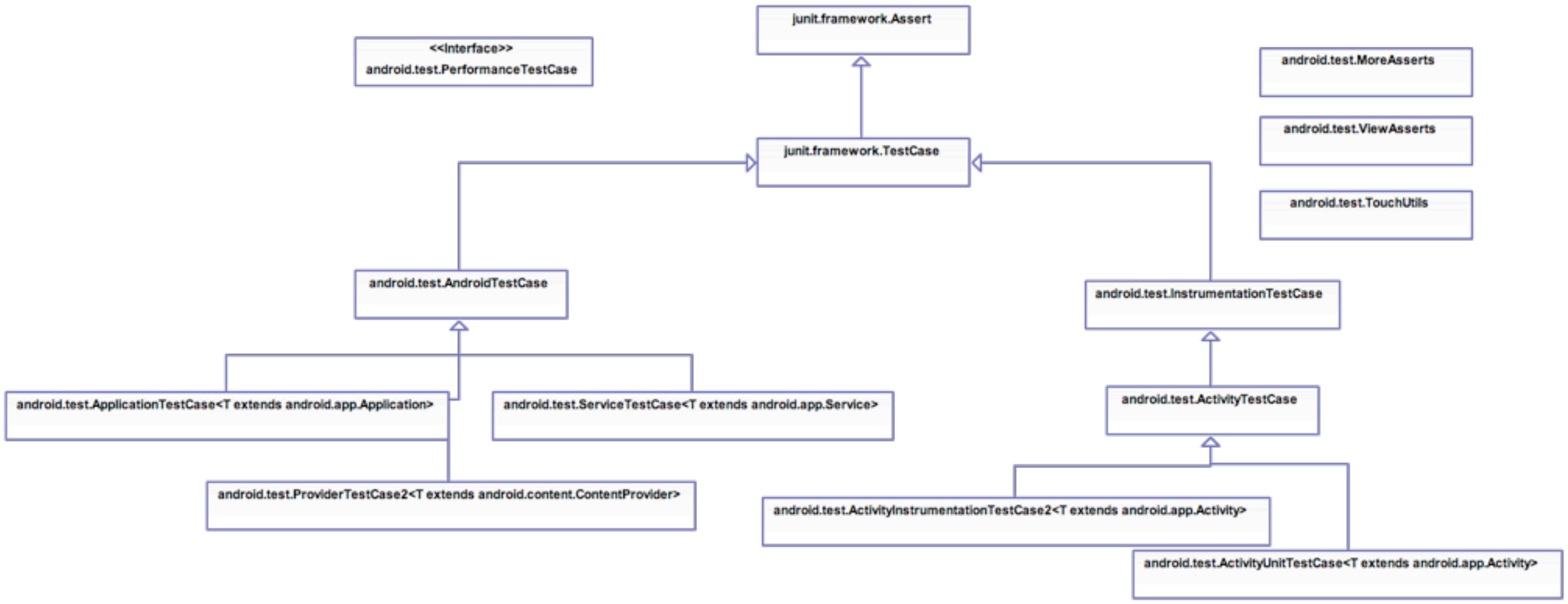
Writing Unit Tests

Test Framework

Instrumentation controls an Android component independently of its normal lifecycle.



TestCases



junit.framework.Assert
<<Interface>> android.test.PerformanceTestCase
android.test.MoreAsserts
android.test.ViewAsserts
junit.framework.TestCase
android.test.TouchUtils
android.test.AndroidTestCase
android.test.InstrumentationTestCase
android.test.ApplicationTestCase<T extends android.app.Application>
android.test.ServiceTestCase<T extends android.app.Service>
android.test.ActivityTestCase
android.test.ProviderTestCase2<T extends android.content.ContentProvider>
android.test.ActivityInstrumentationTestCase2<T extends android.app.Activity>
android.test.ActivityUnitTestCase<T extends android.app.Activity>

Mocks

android.test.mock.MockApplication

android.test.mock.MockContentProvider

android.test.mock.MockContentResolver

android.test.mock.MockContext

android.test.mock.MockCursor

android.test.mock.MockDialogInterface

android.test.mock.MockPackageManager

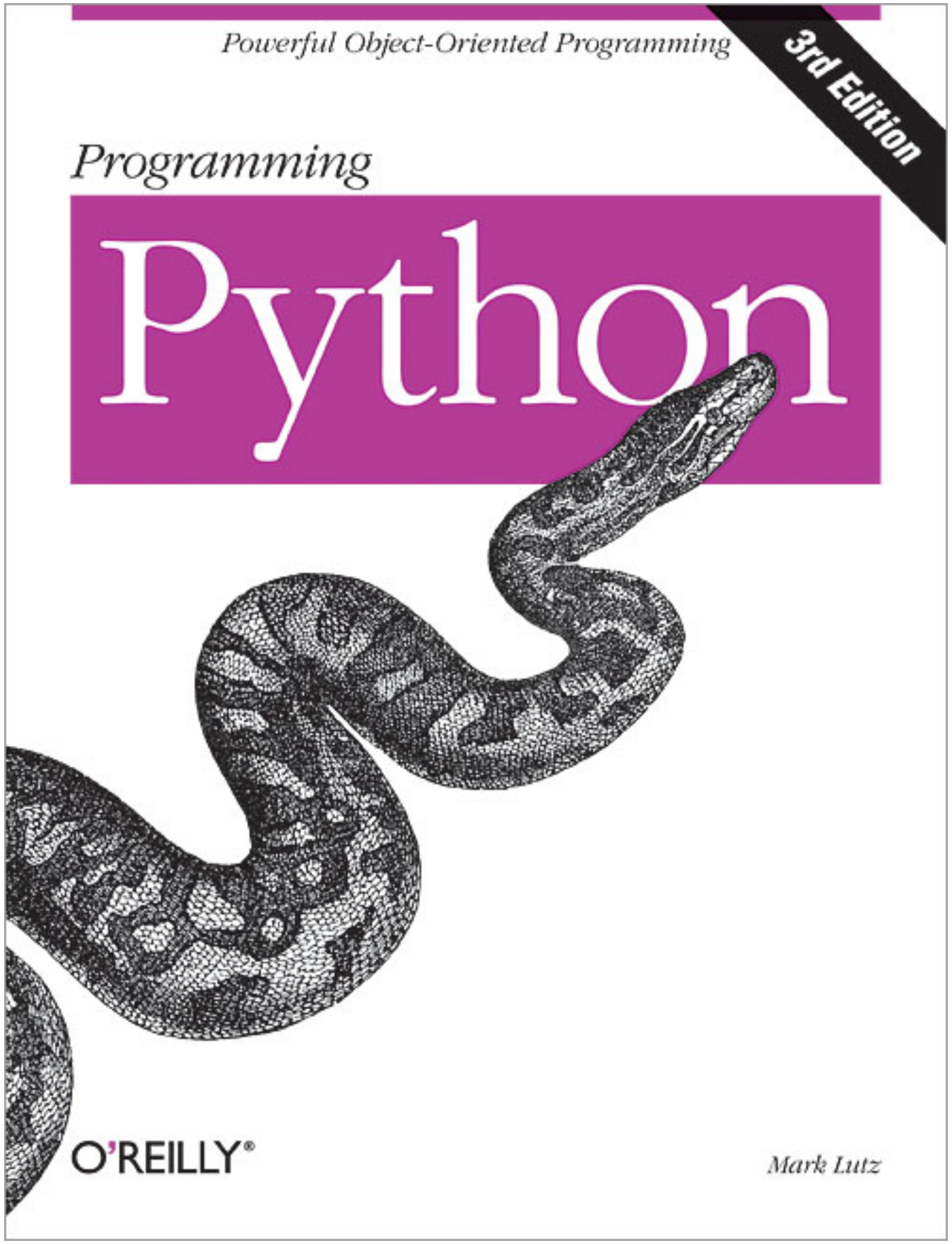
android.test.mock.MockResources

Functional Testing

- Monkey Runner
- Monkey
- Robotium

MonkeyRunner

functional testing framework for Android applications and devices


Powerful Object-Oriented Programming

3rd Edition

Programming

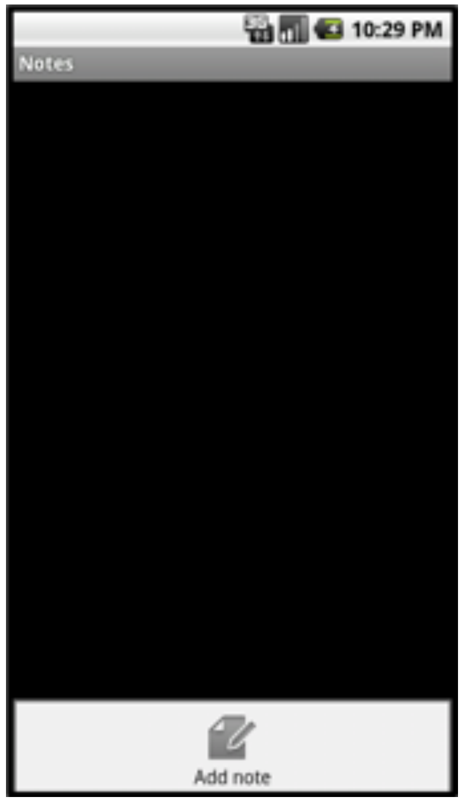
Python

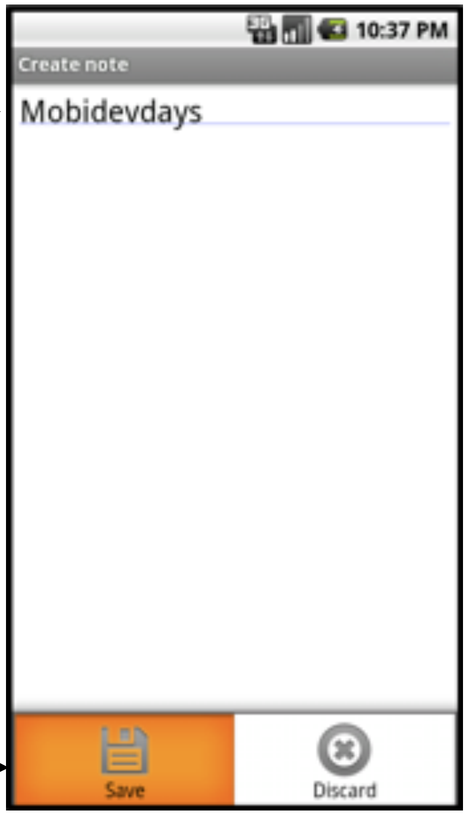
O'REILLY®

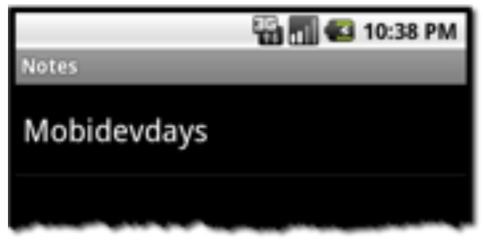
Mark Lutz

```
monkeyrunner add_note.py
```

```
# Imports the monkeyrunner modules
from com.android.monkeyrunner import MonkeyRunner, MonkeyDevice, MonkeyImage

# Connect to the current device
device = MonkeyRunner.waitForConnection()

# Install package
device.installPackage('bin/NotePad.apk')

# Run activity
device.startActivity(component='com.example.android.notepad/.NotesList')

# Press the Menu button
device.press('KEYCODE_MENU','DOWN_AND_UP')

# Press Add Note menu item
device.press('KEYCODE_A','DOWN_AND_UP')

# Type "Mobidevdays"
device.press('KEYCODE_M','DOWN_AND_UP')
device.press('KEYCODE_O','DOWN_AND_UP')
device.press('KEYCODE_B','DOWN_AND_UP')
device.press('KEYCODE_I','DOWN_AND_UP')
device.press('KEYCODE_D','DOWN_AND_UP')
device.press('KEYCODE_E','DOWN_AND_UP')
device.press('KEYCODE_V','DOWN_AND_UP')
device.press('KEYCODE_D','DOWN_AND_UP')
device.press('KEYCODE_A','DOWN_AND_UP')
device.press('KEYCODE_Y','DOWN_AND_UP')

# Press the Menu button
device.press('KEYCODE_MENU','DOWN_AND_UP')

# Press save menu item
device.press('KEYCODE_S','DOWN_AND_UP')

# Take snapshot
screenshot = device.takeSnapshot()

# Write snapshot to file system
screenshot.writeToFile('notes_list.png','png')
```







When things dont work

When things dont work

add

```
MonkeyRunner.sleep(1)
```

😀

- automates android application
- can run in the simulator or the device

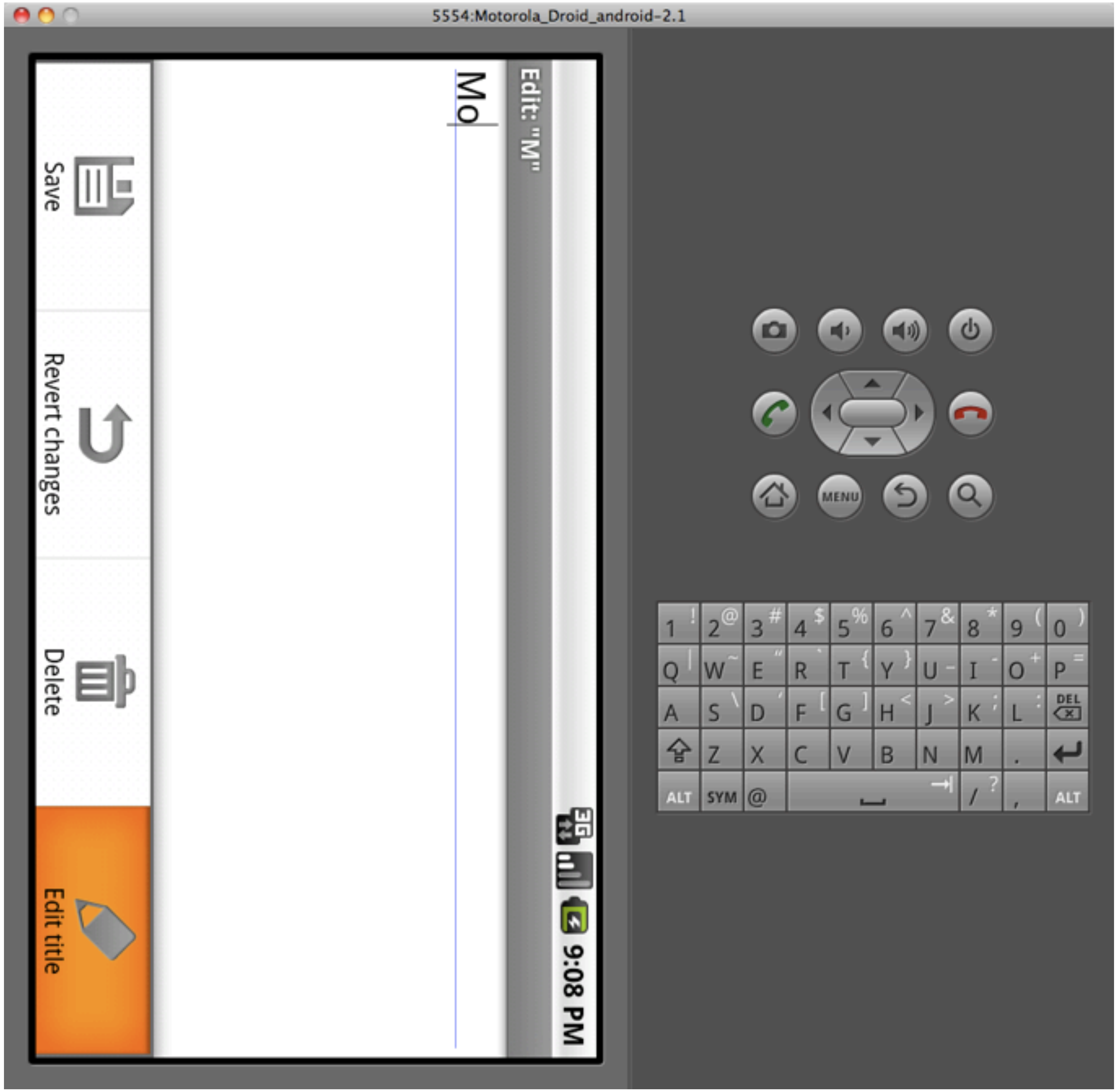
☹️

- difficult to write scripts
- no red bar/green bar
- no verification (other than screenshots)
- very little documentation

Monkey

random click stress tester

```
adb shell monkey -p com.example.android.notepad -v 500
```



😀

- child proofs our app
- looks for crashes
- identifies unresponsiveness

🙁

- not sure the real value

ROBOTIUM

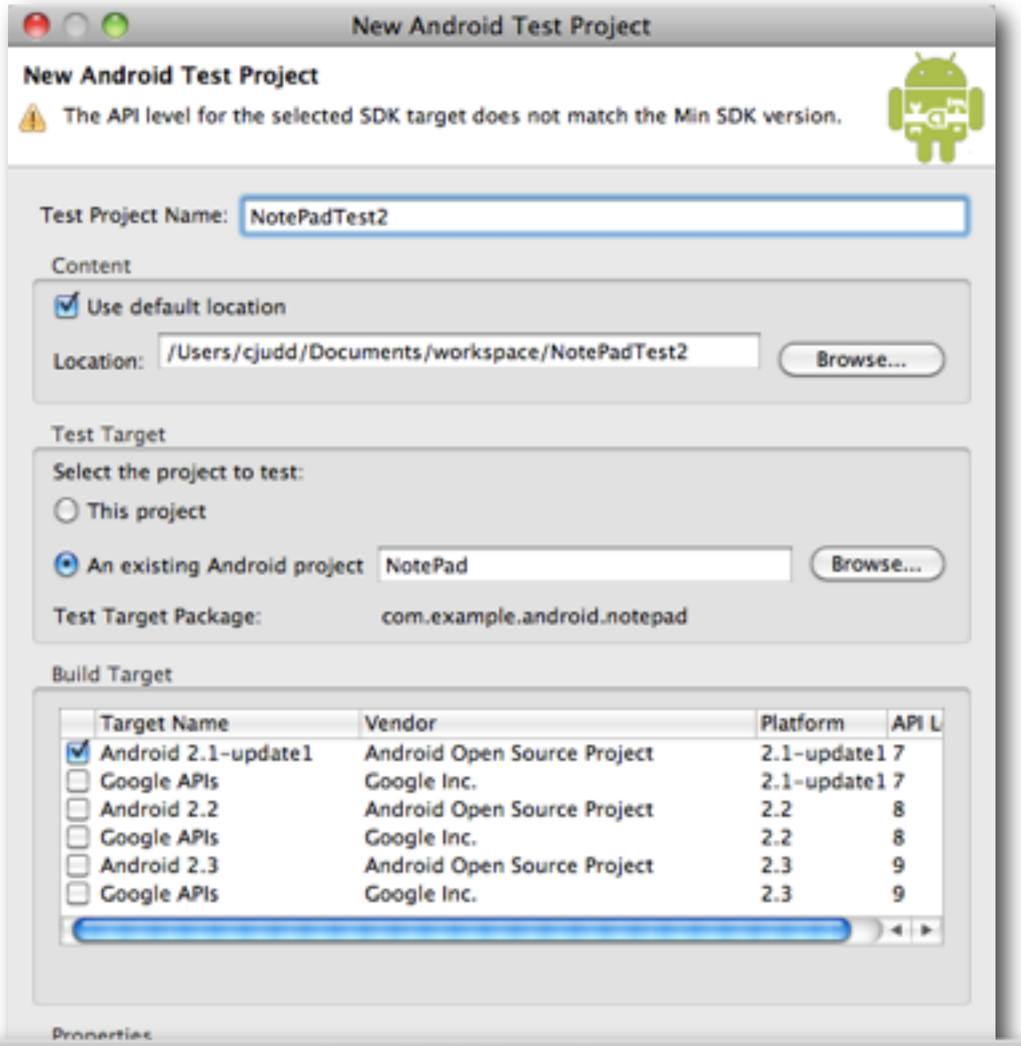
Selenium for Android

Open Source

http://code.google.com/p/robotium/

Setup

1. Create Android Test Project
2. Add robotium-solo-x.x.jar



New Android Test Project

New Android Test Project

The API level for the selected SDK target does not match the Min SDK version.

Test Project Name: NotePadTest2

Content

[✓] Use default location

Location: /Users/cjudd/Documents/workspace/NotePadTest2 Browse...

Test Target

Select the project to test:

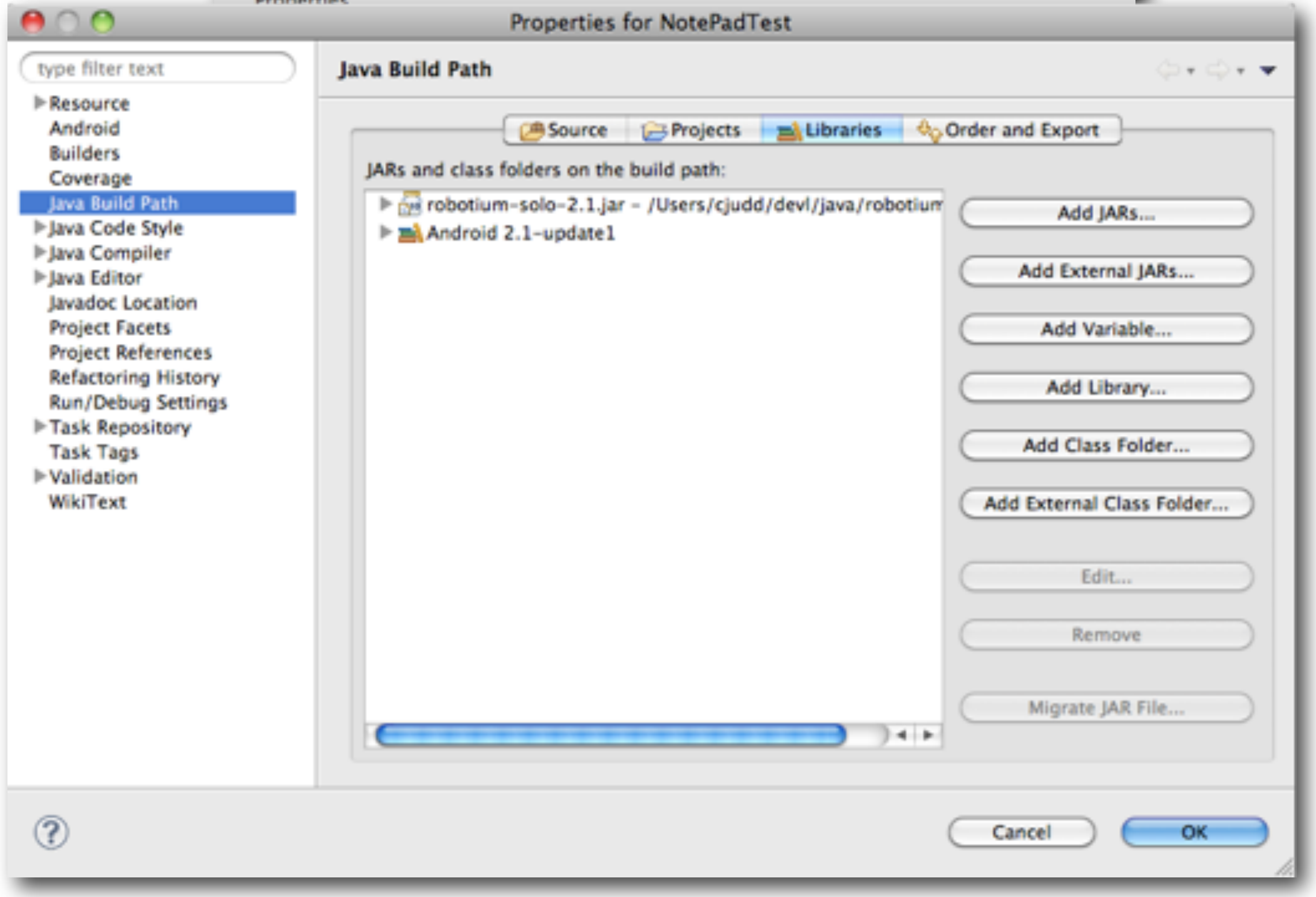
( ) This project

(●) An existing Android project NotePad Browse...

Test Target Package: com.example.android.notepad

Build Target

	Target Name	Vendor	Platform	API L
✓	Android 2.1-update1	Android Open Source Project	2.1-update1	7
	Google APIs	Google Inc.	2.1-update1	7
	Android 2.2	Android Open Source Project	2.2	8
	Google APIs	Google Inc.	2.2	8
	Android 2.3	Android Open Source Project	2.3	9
	Google APIs	Google Inc.	2.3	9



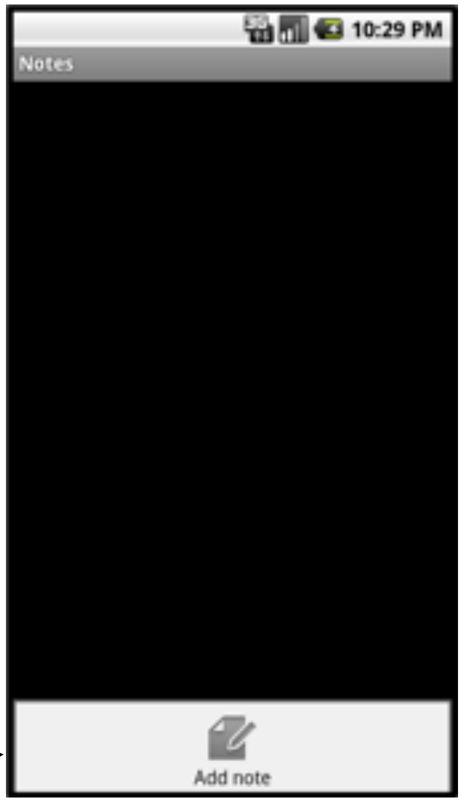
Properties for NotePadTest

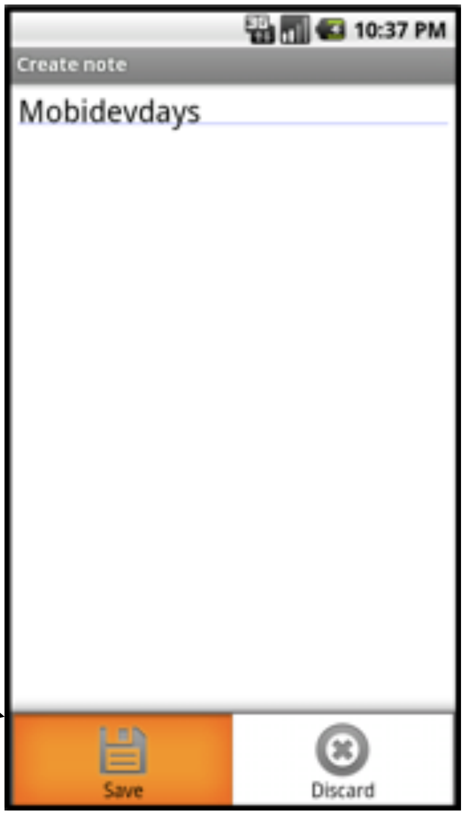
Java Build Path

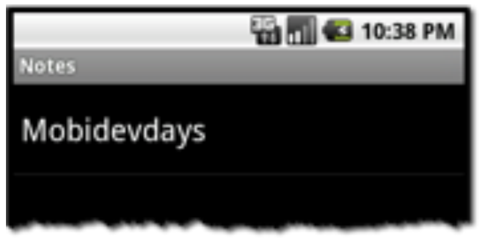
JARs and class folders on the build path:

- robotium-solo-2.1.jar - /Users/cjudd/devl/java/robotium
- Android 2.1-update1

```java
public class NotePadTest extends ActivityInstrumentationTestCase2<NotesList> {

  private static final int TWO_SECONDS = 2000;
  private Solo solo;

  public NotePadTest() {
    super("com.example.android.notepad", NotesList.class);
  }

  protected void setUp() throws Exception {
    super.setUp();
    solo = new Solo(getInstrumentation(), getActivity());
  }

  public void testAddNote() throws Exception {

    solo.sendKey(Solo.MENU);
    solo.sendKey(Solo.MENU); // issue 61

    solo.clickOnMenuItem("Add note");

    solo.sleep(TWO_SECONDS);
    EditText note = (EditText) solo.getView(R.id.note);
    solo.clickOnView(note);
    solo.enterText(note, "Mobidevdays");

    solo.sendKey(Solo.MENU);
    solo.sendKey(Solo.MENU); // issue 61

    solo.clickOnMenuItem("Save");


    assertTrue(solo.searchText("Mobidevdays"));
  }

  public void tearDown() throws Exception {
    try {
      solo.finalize();
    } catch (Throwable e) {
      e.printStackTrace();
    }
    getActivity().finish();
    super.tearDown();
  }

}
```

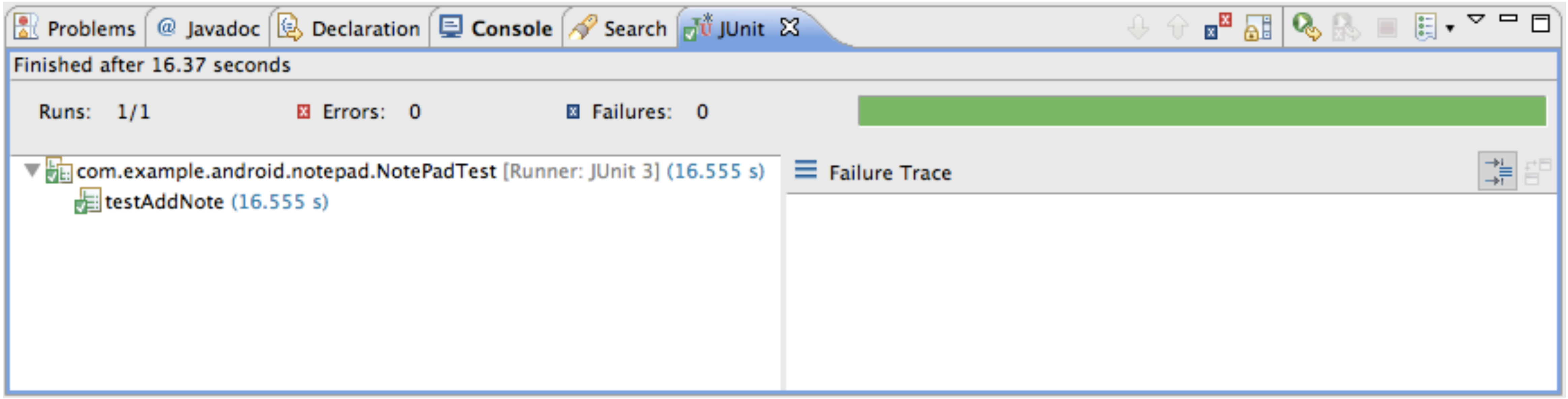






Running

Run As > Android JUnit Test



Finished after 16.37 seconds

Runs: 1/1 Errors: 0 Failures: 0

com.example.android.notepad.NotePadTest [Runner: JUnit 3] (16.555 s)
testAddNote (16.555 s)

Command-line

```
$ adb shell am instrument -w
    com.example.android.notepad.test/android.test.InstrumentationTestRunner

com.example.android.notepad.NotePadTest:.
Test results for InstrumentationTestRunner=.
Time: 14.517

OK (1 test)
```

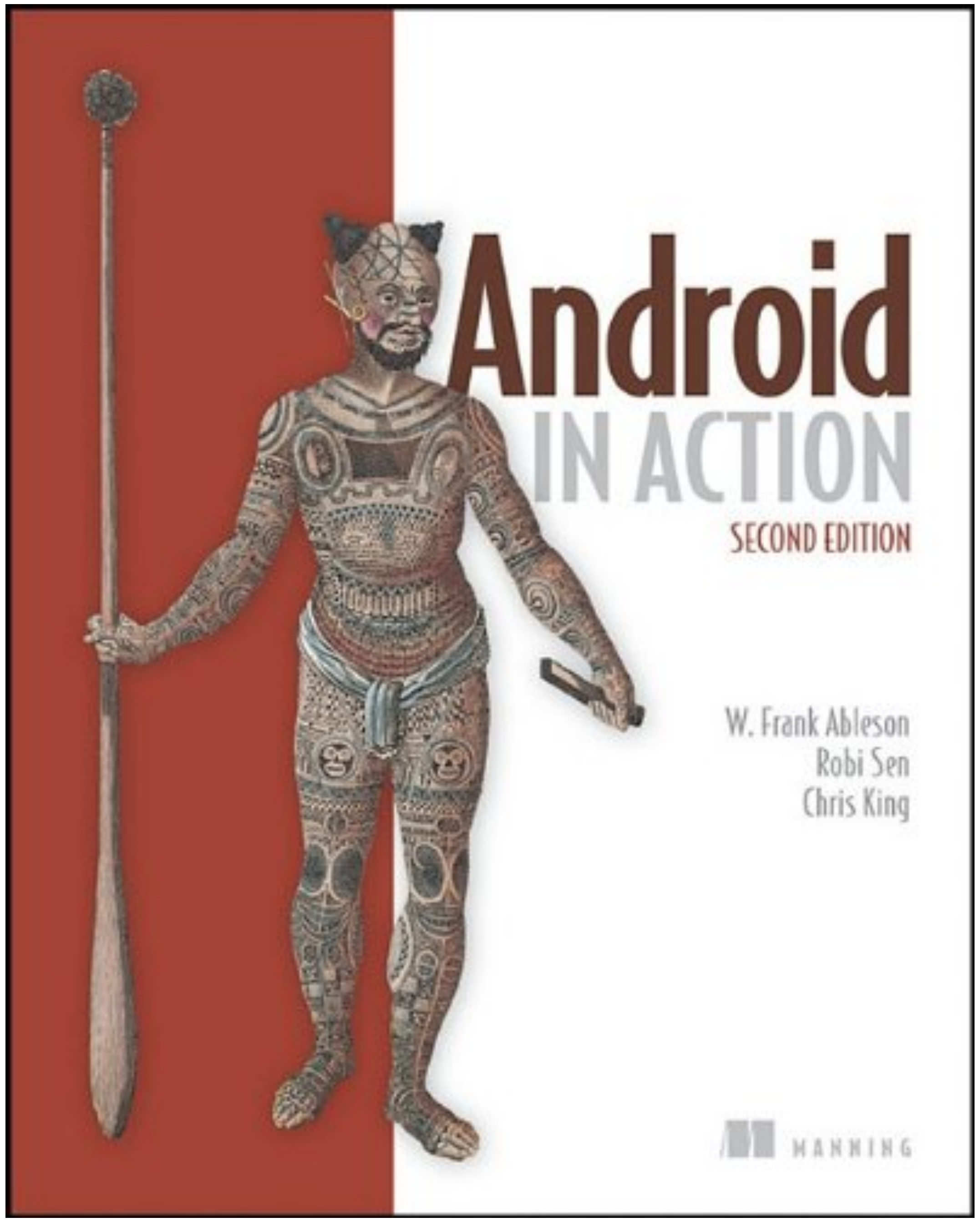
😀

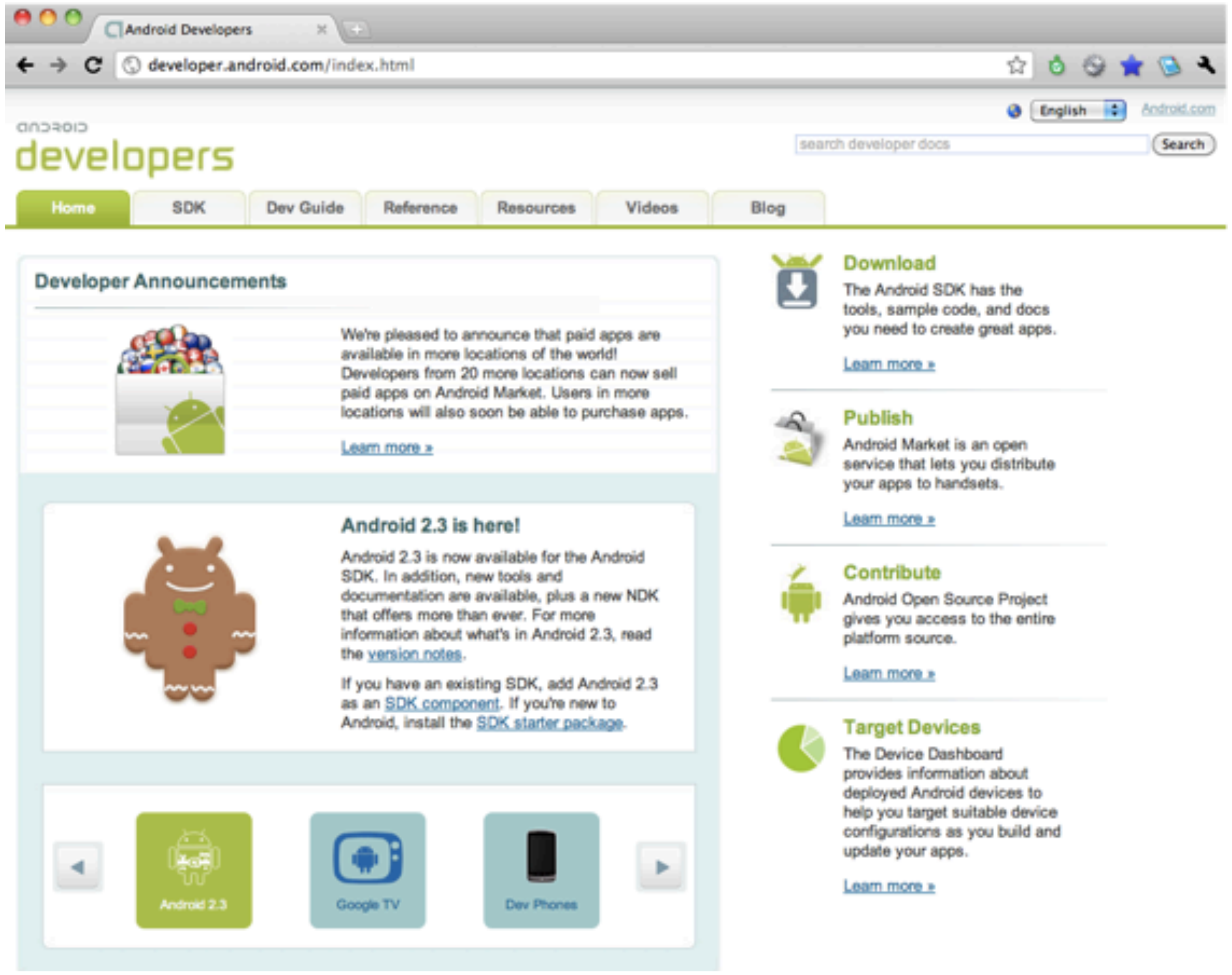
- JUnit based
  - red bar/ green bar
  - asserts
- can run in the simulator or the device
- command-line automation
- integrates with cucumber

☹

- little documentation
- not approachable by traditional testers

Android Resources





Developer Announcements

We're pleased to announce that paid apps are available in more locations of the world! Developers from 20 more locations can now sell paid apps on Android Market. Users in more locations will also soon be able to purchase apps.

Learn more »

Android 2.3 is here!

Android 2.3 is now available for the Android SDK. In addition, new tools and documentation are available, plus a new NDK that offers more than ever. For more information about what's in Android 2.3, read the version notes.

If you have an existing SDK, add Android 2.3 as an SDK component. If you're new to Android, install the SDK starter package.

Android 2.3 Google TV Dev Phones

Download

The Android SDK has the tools, sample code, and docs you need to create great apps.

Learn more »

Publish

Android Market is an open service that lets you distribute your apps to handsets.

Learn more »

Contribute

Android Open Source Project gives you access to the entire platform source.

Learn more »

Target Devices

The Device Dashboard provides information about deployed Android devices to help you target suitable device configurations as you build and update your apps.

Learn more »

http://developer.android.com

Christopher M. Judd

Judd Solutions

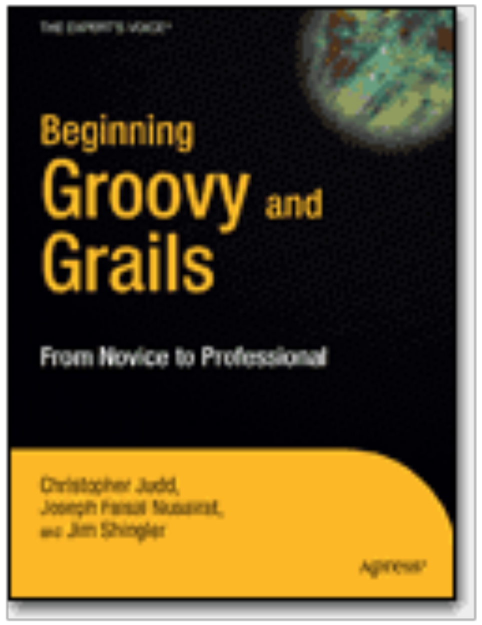
President/Consultant/Author

email: cjudd@juddsolutions.com

web: www.juddsolutions.com

blog: juddsolutions.blogspot.com

twitter: javajudd


THE EXPERT'S VOICE
Beginning Groovy and Grails
From Novice to Professional
Christopher Judd, Joseph Faisal Nusairat, and Jim Shingler
Apress

THE EXPERT'S VOICE
Pro Eclipse JST
Plug-ins for J2EE Development
Christopher Judd and Hakeem Shittu
Apress

THE EXPERT'S VOICE IN JAVA
Enterprise Java Development on a Budget
Leveraging Java Open Source Technologies
Brian Sam-Bodden and Christopher M. Judd
Apress

Attributions

http://www.organicdesign.co.nz/File:Warning.svg

http://www.flickr.com/photos/heliotrop3/4310957752/