



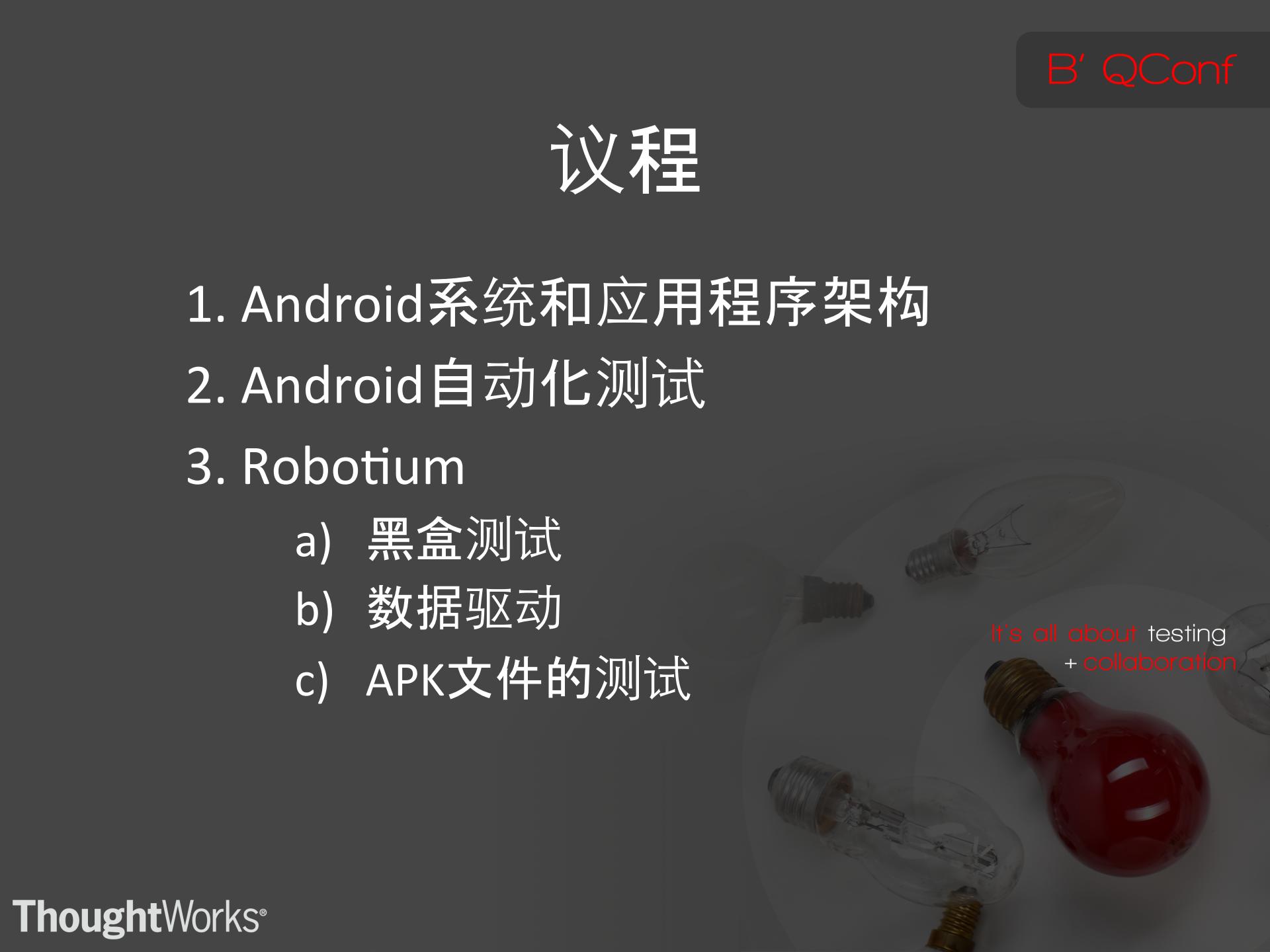
It's all about testing  
+ collaboration

# Robotium Android的黑盒自动化测试

ThoughtWorks 黄勇

# 议程

1. Android系统和应用程序架构
2. Android自动化测试
3. Robotium
  - a) 黑盒测试
  - b) 数据驱动
  - c) APK文件的测试



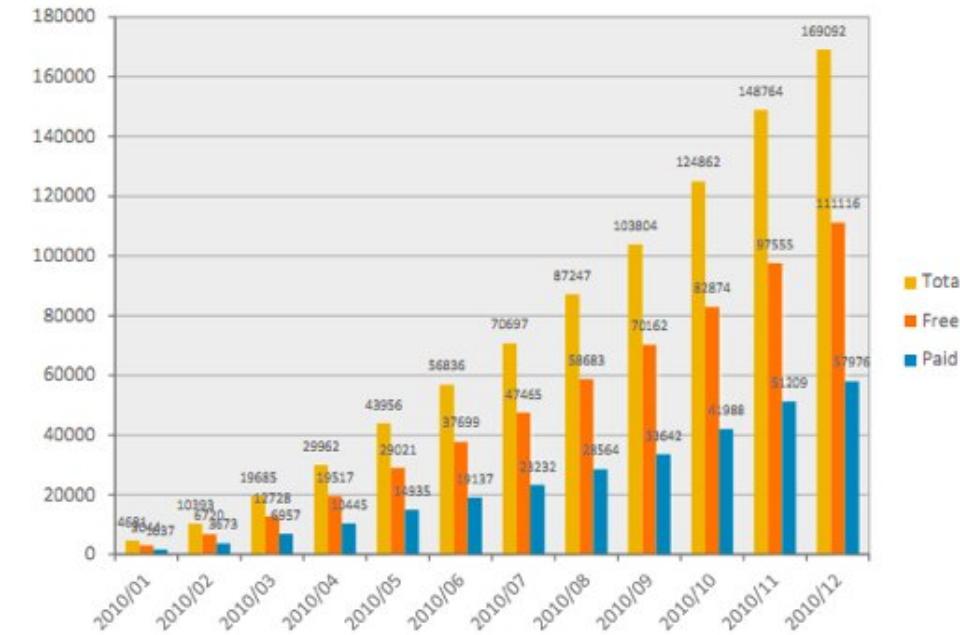
It's all about testing  
+ collaboration



ANDROID

### Applications Evolution

#### Applications Evolution, 2010



# Android系统架构

应用程序

应用程序框架

类库

Dalvik虚拟机

Android  
核心类库

Linux内核

out testing  
collaboration

# Android应用程序的4个组件

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider



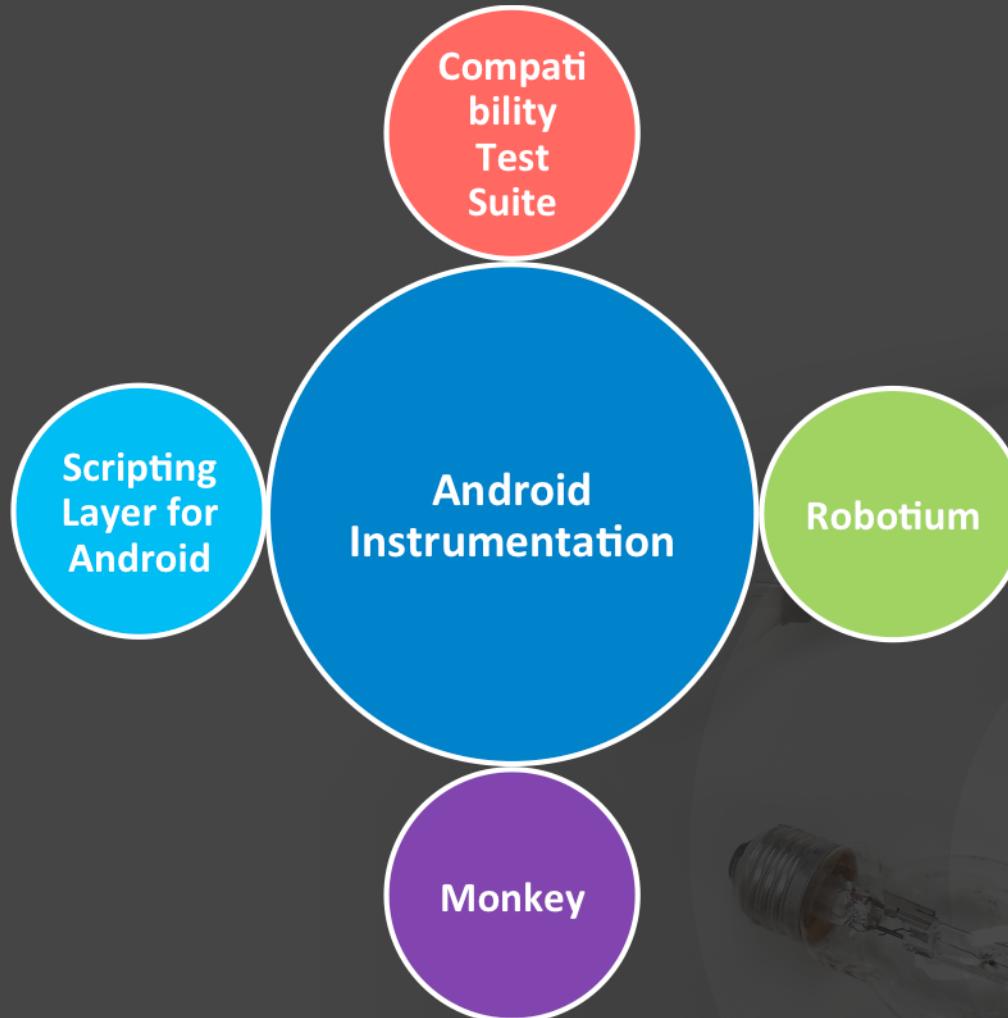
It's all about testing  
+ collaboration

# Android自动化测试



It's all about testing  
+ collaboration

# Android自动化测试方法



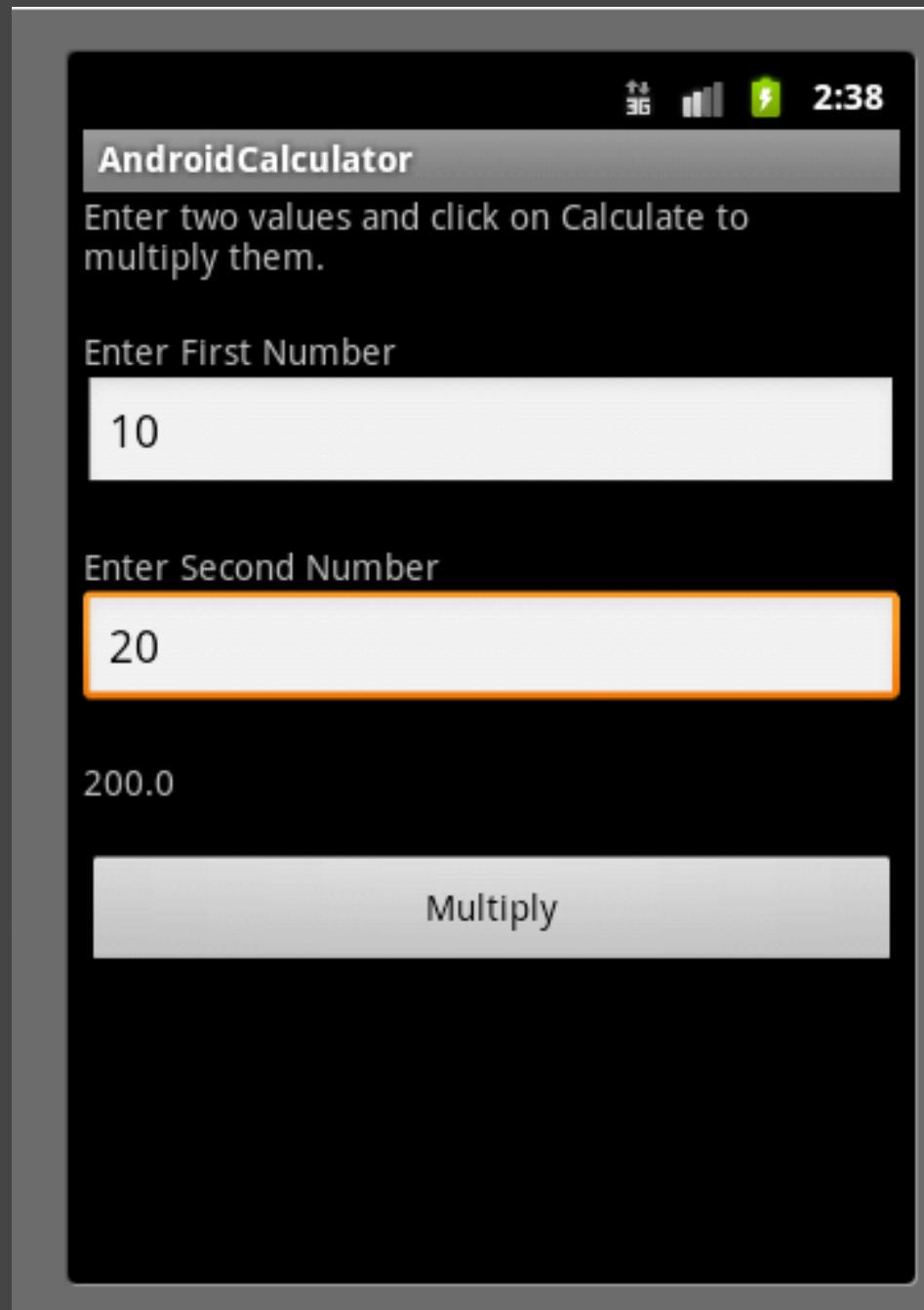
It's all about testing  
+ collaboration

# Robotium可测试的组件

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider



It's all about testing  
+ collaboration



It's all about testing  
+ collaboration



The screenshot shows an Android studio interface with two tabs: "TestMain.java" and "main.xml". The "main.xml" tab is active, displaying the XML code for a layout. The code includes two EditText fields, one TextView, and one Button, each with specific attributes like input type, id, and layout parameters.

```
</>

<EditText
    android:inputType="numberDecimal"
    android:id="@+id/EditText01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</EditText>

<EditText
    android:inputType="numberDecimal"
    android:id="@+id/EditText02"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</EditText>

<TextView
    android:id="@+id/TextView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
/>

<Button
    android:text="Multiply"
    android:id="@+id/Button01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</Button>
```

Graphical Layout main.xml

It's all about testing  
+ collaboration

# Instrumentation

The screenshot shows an IDE interface with two main windows. On the left, the code editor displays `InstrumentationTest.java`. The code implements an `InstrumentationTest` class that extends `ActivityInstrumentationTestCase2<Main>`. It includes a constructor that takes the package name and class name of the activity to be tested. The `testMultiplyResult()` method is highlighted with a red border and contains the logic for setting up input values, performing a button click, and asserting the expected result. On the right, the `JUnit` window shows the test results: it finished after 3.035 seconds with 1 run, 0 errors, and 0 failures. The test class listed is `com.calculator.test.InstrumentationTest`.

```
InstrumentationTest.java
package com.calculator.test;

import android.test.ActivityInstrumentationTestCase2;

public class InstrumentationTest extends ActivityInstrumentationTestCase2<Main> {

    public InstrumentationTest() {
        super("com.calculator", Main.class);
    }

    public void testMultiplyResult() {
        final Activity mActivity = this.getActivity();
        final EditText mFirstText=(EditText)mActivity.findViewById(R.id.EditText01);
        final EditText mSecondText=(EditText)mActivity.findViewById(R.id.EditText02);
        final Button mButton=(Button)mActivity.findViewById(R.id.Button01);
        final TextView mView=(TextView)mActivity.findViewById(R.id.TextView01);
        mActivity.runOnUiThread(new Runnable() {
            public void run() {
                mFirstText.setText("10");
                mSecondText.setText("20");
                mButton.performClick();
                assertEquals("200.0",mView.getText().toString());
            }
        });
    }

    @Override
    protected void setUp() throws Exception {
        super.setUp();
    }
}

JUnit
Finished after 3.035 seconds
Runs: 1 | Errors: 0 | Failures: 0
com.calculator.test.InstrumentationTest
```

# Robotium

The screenshot shows an IDE interface with two main panes. The left pane displays the Java code for a test class named `RobotiumTest.java`. The code uses the Robotium library to interact with an Android application. A red box highlights the `testMultiplyResult()` method, which performs the following steps:

```
    solo.enterText(0, "2");
    solo.enterText(1, "5");
    solo.clickOnButton("Multiply");
    assertTrue(solo.searchText("10"));
```

The right pane shows the JUnit results for the run. It indicates that the test was completed successfully after 8.933 seconds, with 1 run, 0 errors, and 0 failures.

```
RobotiumTest.java
package com.calculator.test;

import com.jayway.android.robotium.solo.Solo;

public class RobotiumTest extends ActivityInstrumentationTestCase2<Main> {

    private Solo solo;

    public RobotiumTest() {
        super("com.calculator", Main.class);
    }

    public void testMultiplyResult() {
        solo.enterText(0, "2");
        solo.enterText(1, "5");
        solo.clickOnButton("Multiply");
        assertTrue(solo.searchText("10"));
    }

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

    @Override
    protected void tearDown() throws Exception {
        try {
            solo.finalize();
        }
    }
}
```

JUnit X

Finished after 8.933 seconds

Runs: 1/1 Errors: 0 Failures: 0

com.calculator.test.RobotiumTest [Run]

Failure Trace

WhiteboxRobotiumTest.java

```
public void testMultiplyResult()
{
    float firstNumber=10;
    float secondNumber=2;
    float result=firstNumber*secondNumber;

    EditText FirsteditText=(EditText)solo.getView(R.id.EditText01);
    solo.enterText(FirsteditText, String.valueOf(firstNumber));

    EditText SecondeditText=(EditText)solo.getView(R.id.EditText02);
    solo.enterText(SecondeditText, String.valueOf(secondNumber));

    solo.clickOnButton("Multiply");

    assertTrue(solo.searchText(String.valueOf(result)));

    TextView outputField=(TextView)solo.getView(R.id.TextView01);

    ArrayList currentTextViews=solo.getCurrentTextViews(outputField);
    assertFalse(currentTextViews.isEmpty());

    TextView output=(TextView)currentTextViews.get(0);

    assertEquals(String.valueOf(result),output.getText().toString());
}

@Override
protected void setUp() throws Exception {
    super.setUp();
}
```

JUnit

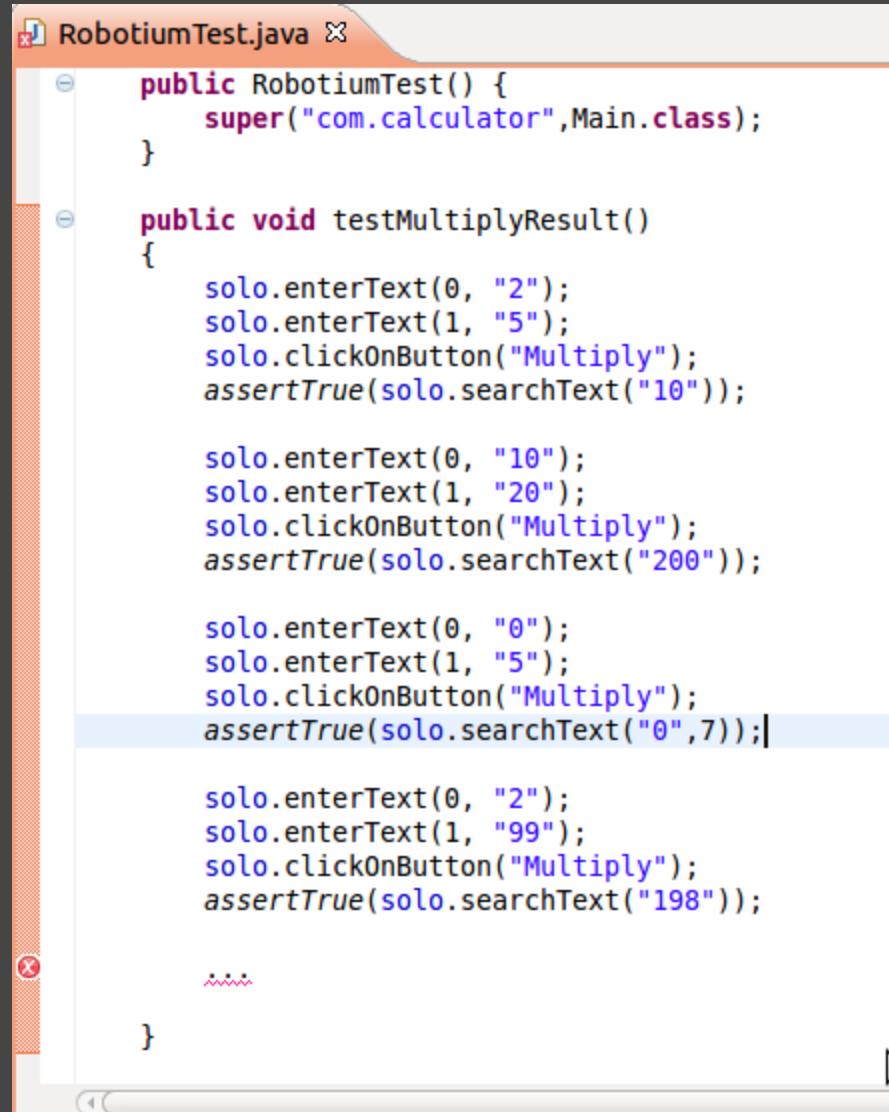
Finished after 11.826 seconds

Runs: 1/1 Errors: 0 Failures: 0

com.calculator.test.WhiteboxRobotium

Failure Trace

# 数据驱动



```
RobotiumTest.java
public RobotiumTest() {
    super("com.calculator", Main.class);
}

public void testMultiplyResult()
{
    solo.enterText(0, "2");
    solo.enterText(1, "5");
    solo.clickOnButton("Multiply");
    assertTrue(solo.searchText("10"));

    solo.enterText(0, "10");
    solo.enterText(1, "20");
    solo.clickOnButton("Multiply");
    assertTrue(solo.searchText("200"));

    solo.enterText(0, "0");
    solo.enterText(1, "5");
    solo.clickOnButton("Multiply");
    assertTrue(solo.searchText("0", 7));

    solo.enterText(0, "2");
    solo.enterText(1, "99");
    solo.clickOnButton("Multiply");
    assertTrue(solo.searchText("198"));

    ...
}
```

It's all about testing  
+ collaboration

# DDMS

**Devices**

Name	131	8604
com.android.systemui	131	8604
com.android.launcher	135	8605
com.svox.pico	350	8724
com.android.quickstep	365	8727
com.android.email	425	8756
com.calculator	443	8763 / 8700

**Emulator Control**

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

Voice  
 SMS

**File Explorer**

Name	Size	Date	Time	Permissions	Info
com.android.systemui		2011-01-26	03:14	drwxr-x-x	
com.android.term		2011-01-26	03:14	drwxr-x-x	
com.android.wallpaper.livepreviews		2011-01-26	03:14	drwxr-x-x	
com.calculator		2011-02-28	06:50	drwxr-x-x	
files		2011-02-28	06:32	drwxrwx-x	
TestData.csv	9728	2011-02-28	06:28	-rw-rw-rw-	
TestResultData.txt	24	2011-02-28	06:51	-rw-rw-r-	
lib		2011-02-28	06:50	drwxr-xr-x	
com.calculator.test		2011-02-28	06:50	drwxr-x-x	
com.example.android.notepad					/data/data/\$PACKAGE_NAME/files
com.svox.pico					
com.testcalculator					
jp.co.omronsoft.openwnn					
dontpanic					
local					
lost+found					

**LogCat**

Log

Filter:

DataDrivenRobotiumTest.java

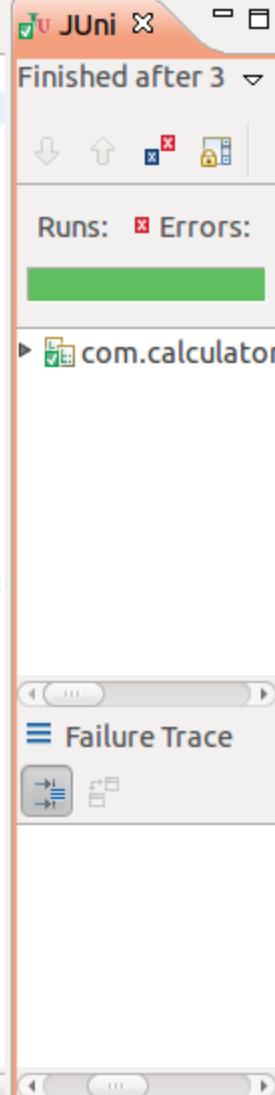
```
public void testMultiplyResultDataDriven()
{
    getTestDataFile();
    TestResults = new String[rowCount];

    for (int i = 1; i < rowCount; i++)
    {
        solo.enterText(0, FirstValue[i].toString());
        solo.enterText(1, SecondValue[i].toString());
        solo.clickOnButton("Multiply");
        float First = Float.parseFloat(FirstValue[i].toString());
        float Second = Float.parseFloat(SecondValue[i].toString());
        float expectedResult = First*Second;
        float actualResult = Float.parseFloat(solo.getText(9).getText().toString());

        System.out.println("Value in Resultant Field "+solo.getText(9).getText().toString());
        System.out.println("Expected Value "+expectedResult);

        assertNotNull(solo.getText(9));
        assertEquals("Values are not same", expectedResult, actualResult);
        TestResults[i] = "Pass;";
    }
    writeResults(TestResults);
}

public void readTestData(InputStream fileInputStream) throws Exception {
    ws = null;
```



LogCat X

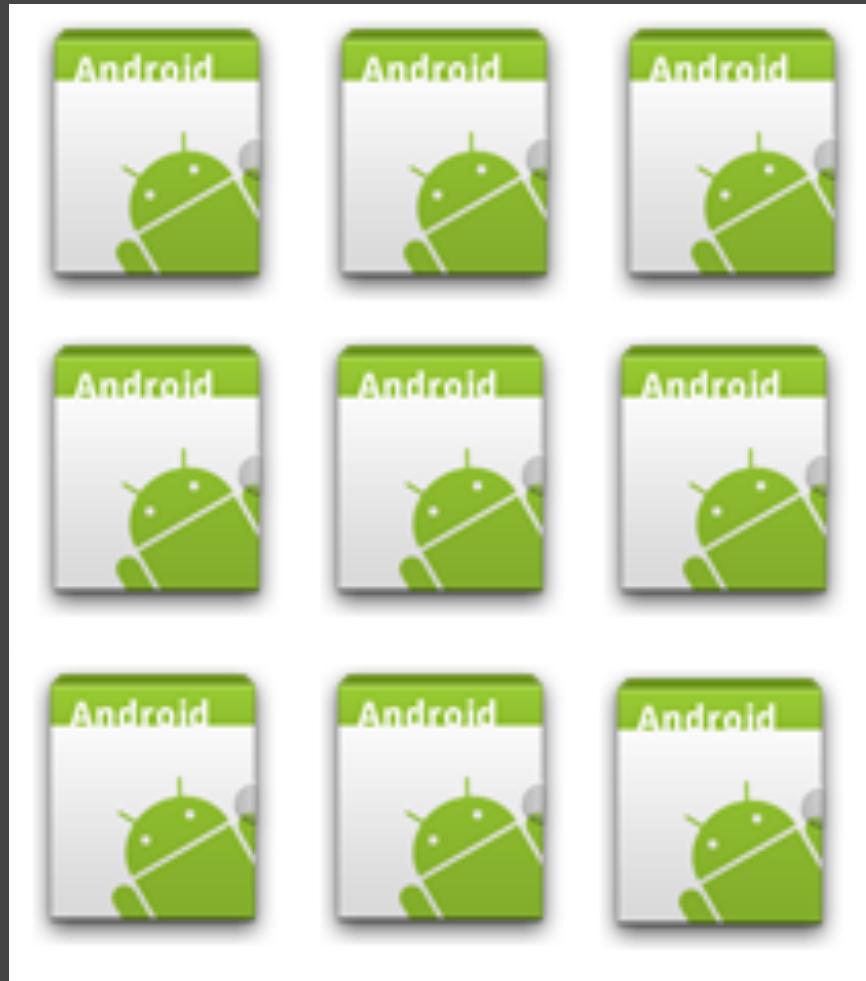
Log

Time		pid	tag	Message
02-28 07:02:06	D	2328	dalvikvm	GC_EXPLICIT freed 12K, 27% free 7837K/10631K, external 716K/1038K, paused 166ms
02-28 07:02:06	I	2328	System.out	Executed: TS1
02-28 07:02:06	I	2328	System.out	Executed: TS2
02-28 07:02:06	I	2328	System.out	Executed: TS3
02-28 07:02:06	I	2328	System.out	Executed: TS4
02-28 07:02:06	I	2328	System.out	Executed: TS5
02-28 07:02:06	I	2328	System.out	Success
02-28 07:02:06	D	2328	dalvikvm	GC_EXPLICIT freed 8K, 27% free 7833K/10631K, external 716K/1038K, paused 182ms
02-28 07:02:06	D	2328	dalvikvm	GC_EXPLICIT freed 5K, 27% free 7827K/10631K, external 716K/1038K, paused 198ms
02-28 07:02:08	D	135	dalvikvm	GC_EXPLICIT freed 52K, 52% free 2851K/5895K, external 2535K/3143K, paused 122ms
02-28 07:02:10	I	2328	System.out	Value in Resultant Field 15.0
02-28 07:02:10	I	2328	System.out	Expected Value 15.0
02-28 07:02:17	I	2328	System.out	Value in Resultant Field 60.0
02-28 07:02:17	I	2328	System.out	Expected Value 60.0
02-28 07:02:22	I	2328	System.out	Value in Resultant Field 135.0

Filter:



# 只有APK文件应用程序的测试



It's all about testing  
+ collaboration

TestAPK.java

```
package com.calculator.test;

import com.jayway.android.robotium.solo.Solo;

@SuppressWarnings("unchecked")
public class TestAPK extends ActivityInstrumentationTestCase2
{

    private static final String TARGET_PACKAGE_ID="com.calculator";
    private static final String LAUNCHER_ACTIVITY_FULL_CLASSNAME="com.calculator.Main";

    private static Class<?> launcherActivityClass;

    static{
        try
        {
            launcherActivityClass=Class.forName(LAUNCHER_ACTIVITY_FULL_CLASSNAME);
        } catch (ClassNotFoundException e) {
            throw new RuntimeException("Class not found" +e);
        }
    }

    public TestAPK() throws ClassNotFoundException
    {
        super(TARGET_PACKAGE_ID,launcherActivityClass);
    }

    private Solo solo;

    @Override
```

JUnit

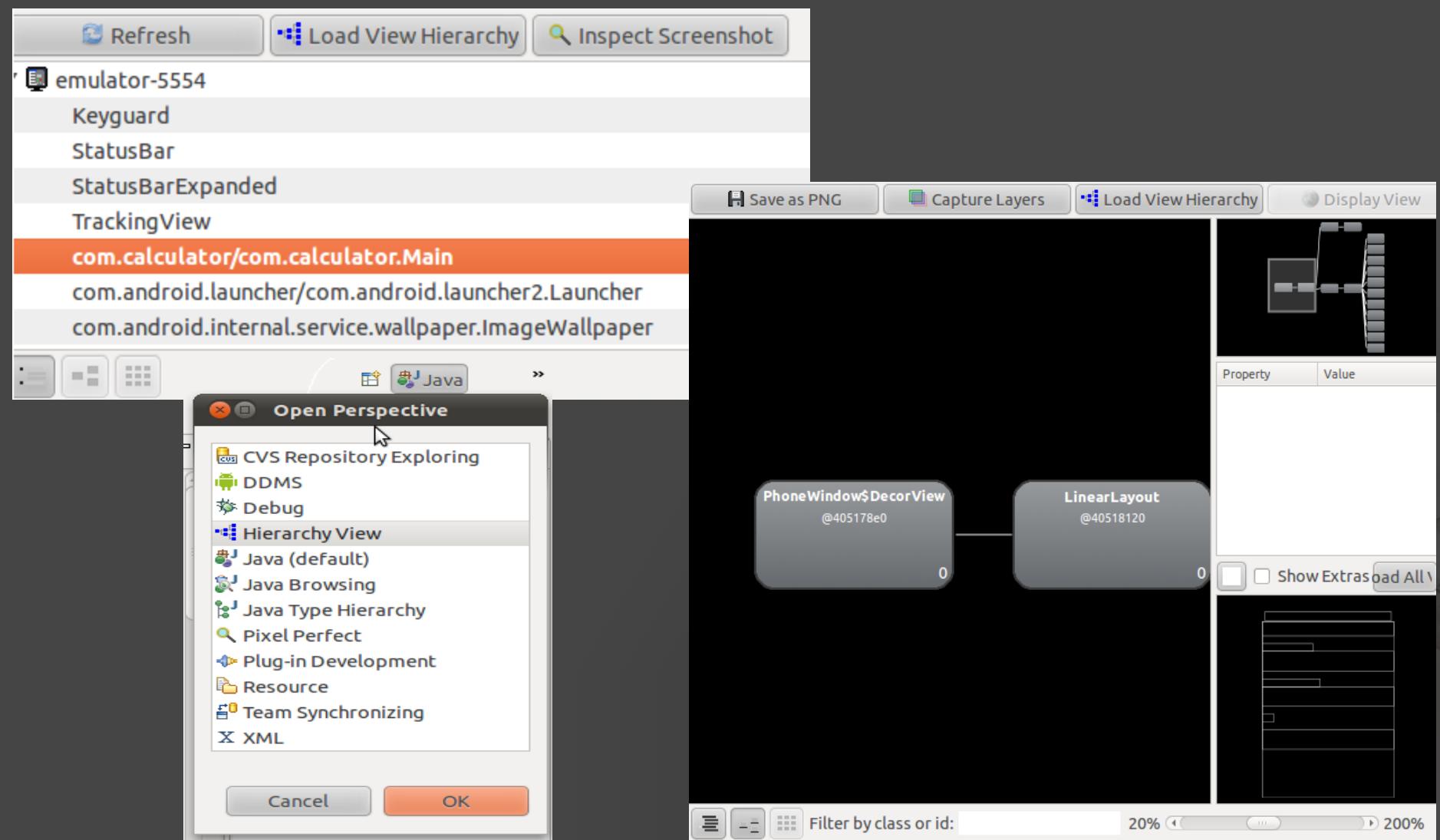
Finished after 36.288 seconds

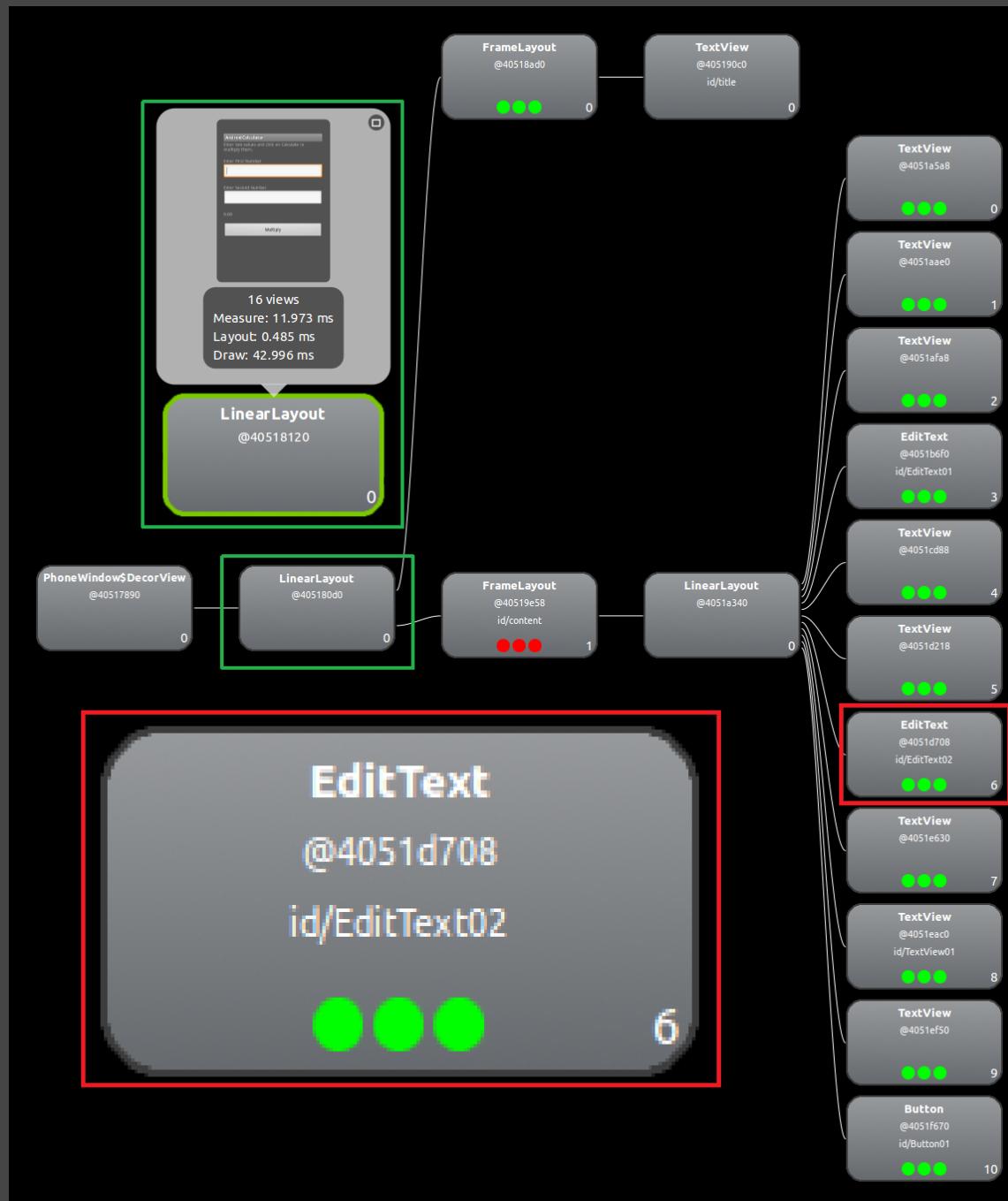
Runs: 1/1 Errors: 0 Failures: 0

com.calculator.test.TestAPK [R]

Failure Trace

# Hierarchy Viewer



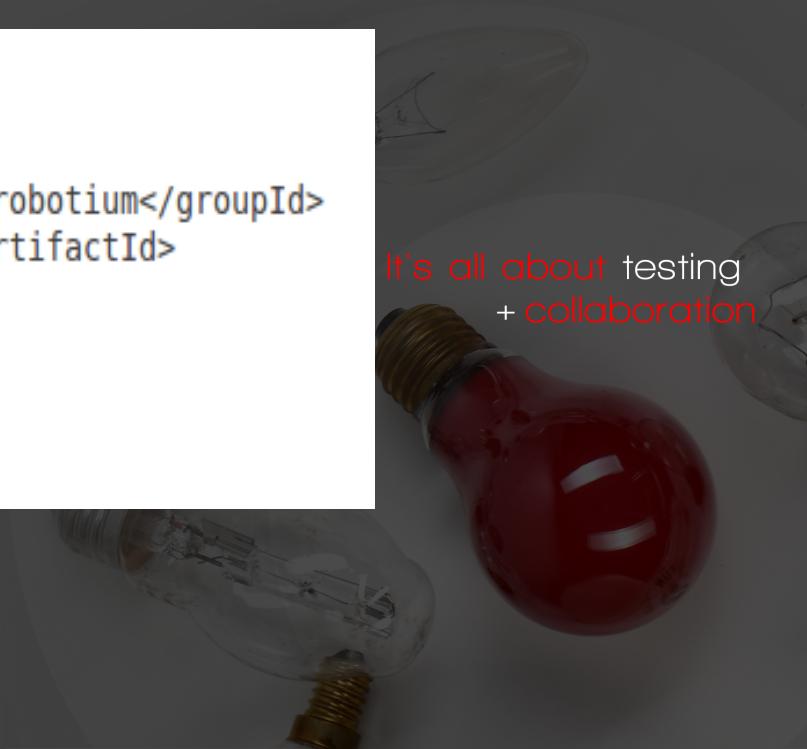


ll about testing  
+ collaboration

# 与Maven的集成

1. 按照通常步骤设置Maven
2. 在测试的工程里添加对Robotium的依赖

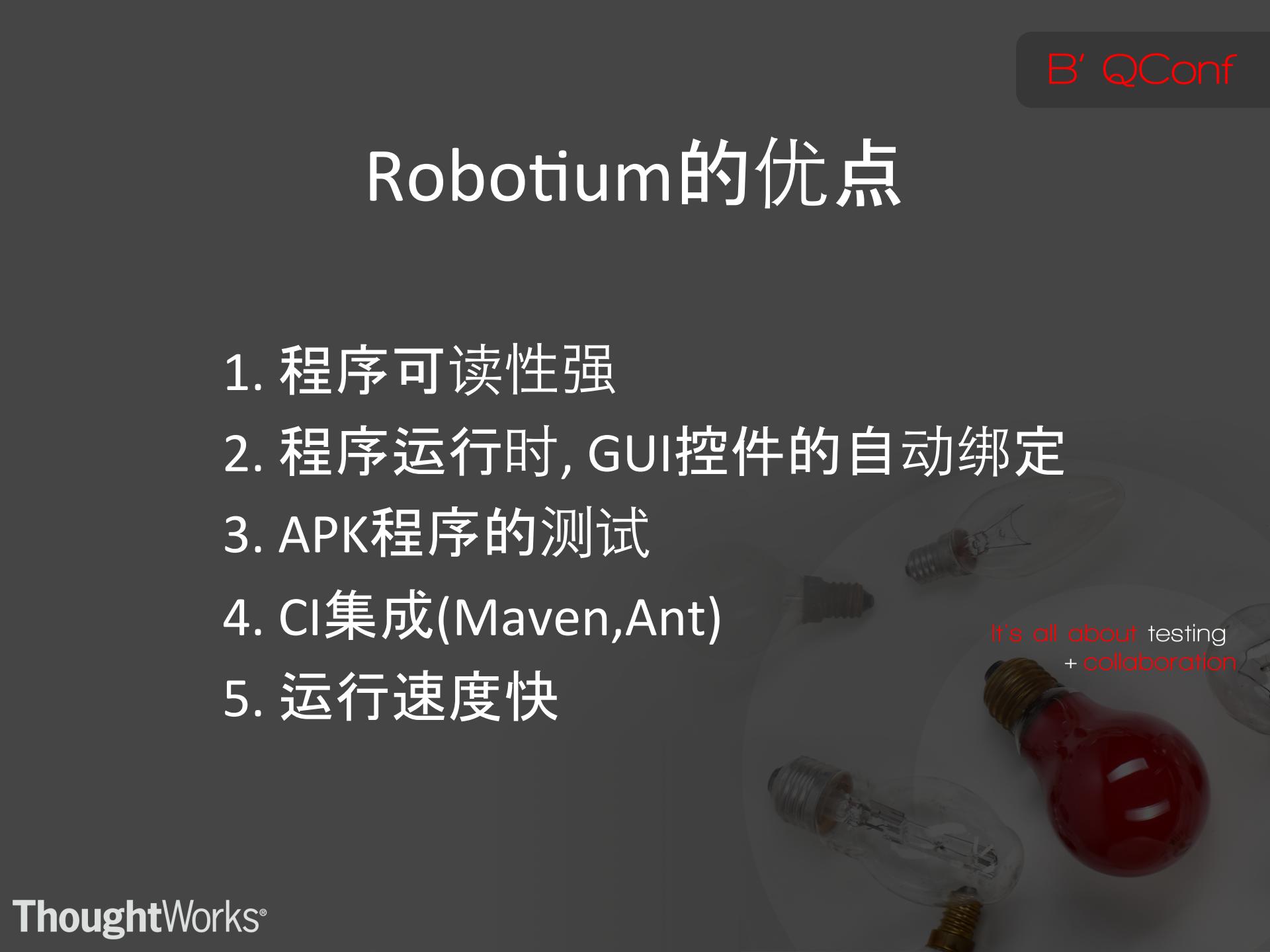
```
<dependencies>
    <dependency>
        <groupId>com.jayway.android.robotium</groupId>
        <artifactId>robotium-solo</artifactId>
        <version>1.8.0</version>
    </dependency>
    ...
</dependencies>
```

A dark background featuring several glowing lightbulbs of different sizes and colors (red, yellow, blue) arranged in a cluster, symbolizing ideas or innovation.

It's all about testing  
+ collaboration

# Robotium的优点

1. 程序可读性强
2. 程序运行时, GUI控件的自动绑定
3. APK程序的测试
4. CI集成(Maven, Ant)
5. 运行速度快



It's all about testing  
+ collaboration

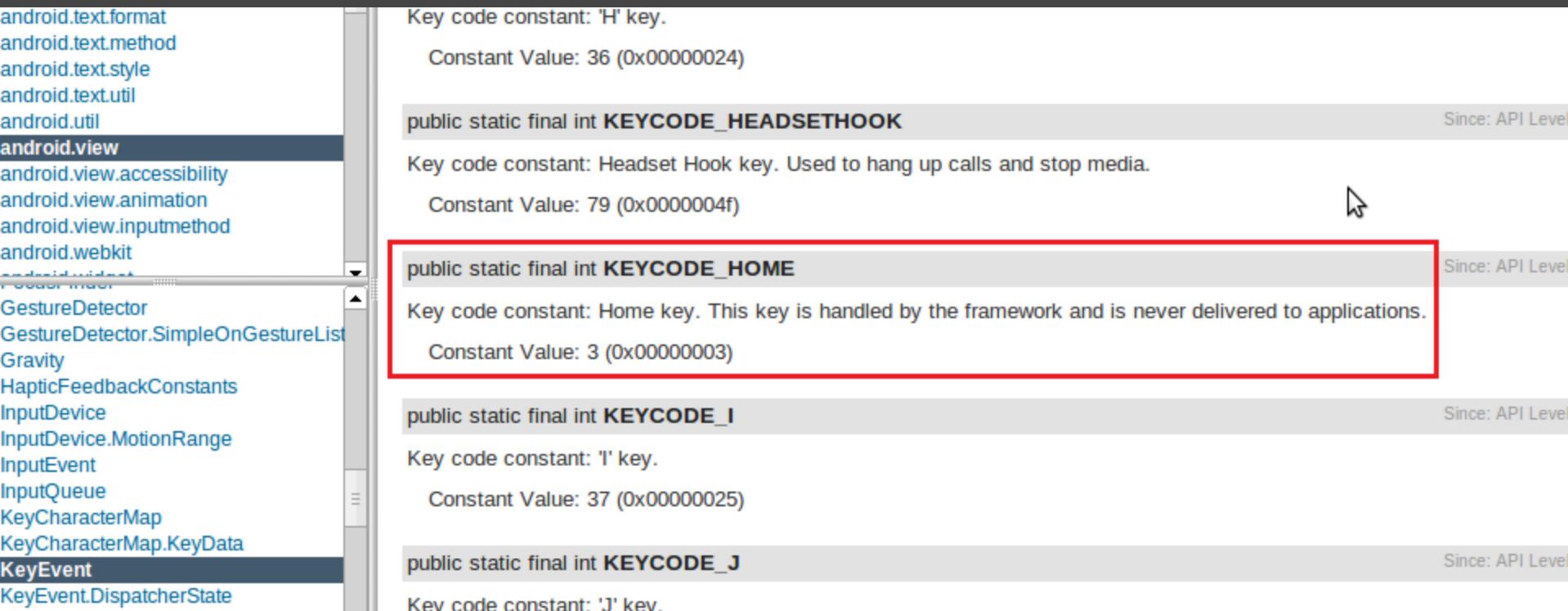
# Robotium的局限

1. 只能针对可见UI进行测试
2. 依赖于JUnit3的Instrumentation
3. 只能以第三方软件的形式安装在终端上测试
4. 没有数据驱动相关方法的封装



It's all about testing  
+ collaboration

# solo.sendKey(KeyEvent.KEYCODE\_HOME)



android.text.format  
android.text.method  
android.text.style  
android.text.util  
android.util  
**android.view**  
android.view.accessibility  
android.view.animation  
android.view.inputmethod  
android.webkit  
android.widget  
GestureDetector  
GestureDetector.SimpleOnGestureList  
Gravity  
HapticFeedbackConstants  
InputDevice  
InputDevice.MotionRange  
InputEvent  
InputQueue  
KeyCharacterMap  
KeyCharacterMap.KeyData  
**KeyEvent**  
KeyEvent.DispatcherState  
LayoutInflater

Key code constant: 'H' key.  
Constant Value: 36 (0x00000024)

**public static final int KEYCODE\_HEADSETHOOK** Since: API Level  
Key code constant: Headset Hook key. Used to hang up calls and stop media.  
Constant Value: 79 (0x0000004f)

**public static final int KEYCODE\_HOME** Since: API Level  
Key code constant: Home key. This key is handled by the framework and is never delivered to applications.  
Constant Value: 3 (0x00000003)

**public static final int KEYCODE\_I** Since: API Level  
Key code constant: 'I' key.  
Constant Value: 37 (0x00000025)

**public static final int KEYCODE\_J** Since: API Level  
Key code constant: 'J' key.

# Robotium的后续发展

1. Remote control(类似于Selenium RC)
2. 与Cucumber的集成
3. UI测试覆盖率的度量
4. 测试失败时的截图功能
5. 多种设备支持和交互



It's all about testing  
+ collaboration

# 截图功能的实现

```
public static String SCREEN_SHOTS_LOCATION="/sdcard/";

public static void takeScreenShot(View view) throws Exception {
    takeScreenShot(view, "default");
}

public static void takeScreenShot(View view, String name) throws
Exception {
    view.setDrawingCacheEnabled(true);
    view.buildDrawingCache();
    Bitmap b = view.getDrawingCache();
    FileOutputStream fos = null;
    try {
        File sddir = new File(SCREEN_SHOTS_LOCATION);
        if (!sddir.exists()) {
            sddir.mkdirs();
        }
        fos = new FileOutputStream(SCREEN_SHOTS_LOCATION+name+
                + System.currentTimeMillis() + ".jpg");
        if (fos != null) {
            b.compress(Bitmap.CompressFormat.JPEG, 90, fos);
            fos.close();
        }
    } catch (Exception e) {
    }
}

//call
takeScreenShot(solo.getViews().get(0), "testname");
```

about testing  
collaboration

# 机遇和挑战



out testing  
collaboration

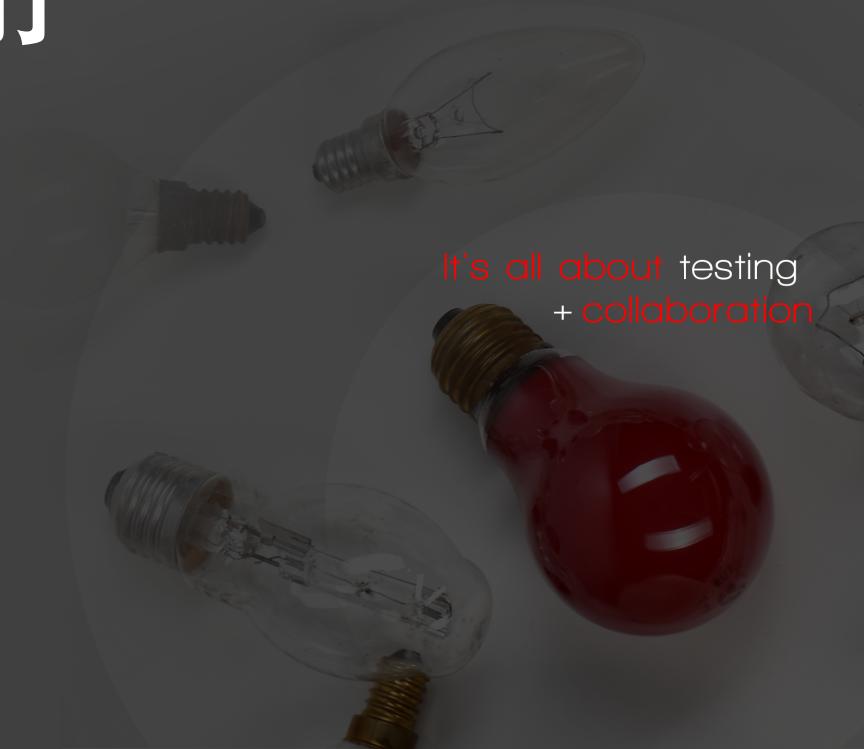
# 参考文献

- ◆ [developer.android.com/](http://developer.android.com/)
- ◆ [code.google.com/p/robotium/](http://code.google.com/p/robotium/)
- ◆ [www.squeed.com/Javaforum\\_2011Q1\\_files/Robotium.pdf](http://www.squeed.com/Javaforum_2011Q1_files/Robotium.pdf)
- ◆ [controlingquality.blogspot.com/2011/02/design-data-driven-framework-around.html](http://controlingquality.blogspot.com/2011/02/design-data-driven-framework-around.html)
- ◆ [blog.csdn.net/roger\\_ge/category/680331.aspx](http://blog.csdn.net/roger_ge/category/680331.aspx)
- ◆ [fanfq.javaeye.com/blog/781665](http://fanfq.javaeye.com/blog/781665)



It's all about testing  
+ collaboration

# 谢谢



It's all about testing  
+ collaboration

# Q&A



It's all about testing  
+ collaboration