Unit Testing, Integration Testing and Continuous Builds

Manfred Moser

simpligility technologies inc.
http://www.simpligility.com
@simpligility

Agenda

Get an overview about testing and continuous integration for Android app development

Why testing?

What can we test?

How can we do it?

Apache Maven

See previous presentation

Maven used to control build and more

Good library reuse and dependency use – makes testing easier out of the box

Strong tool support

But its all possible without Maven too...

Why (automated) testing?

Find problem early and you

- Can fix it quickly
- Safe money on QA testing
- Do not get negative customer feedback
- Deal with feature requests instead of bugs
- Avoid production problems
- Can refactor (and change) without breaking old stuff

What are we testing?

Plain java code

Android dependent code

Configuration

User interface

Look and feel

JVM vs Dalvik/Android stack

JVM based:

- Faster
- More tools available
- More mature tooling

Dalvik based:

- Necessary for integration tests
- Reproduce actual behaviour
- Full stack testing (beyond VM, to native..)

JVM testing tools

- JUnit
- TestNG
- EasyMock
- Unitils
- Cobertura
- Emma
- and many more

Android SDK Test Tools

- Integrated Junit
  - use on emulator/device though
- Instrumentation Test Tools
  - rich set of classes for testing
  - now well documented
- MonkeyRunner
  - control device/emulator running tests
  - take screenshots
  - jython

Dalvik/Android test tools

- Robotium
- Robolectric
- AndroidMock
- Calculon

Unit tests

Just like with normal Java development

Running on JVM

With TestNG or Junit

Lots more other testing tools (Cobertura, Emma, EasyMock, Unitils...)

MorseFlash example

Android SDK Instrumentation tests

Separate module with instrumentation test

Wide array of helper classes

Including mock classes

MorseFlash example

Others

Unit tests with Android dependencies need to run on Dalvik/Android in most cases

Android dependencies mocked can run on JVM

Robotium

Robotium

Like Selenium for Android

Extends SDK instrumentation testing

Add dependency to pom.xml and start coding tests

Robotium sample

Robolectric

Runs in JVM

Shadows Android SDK classes

No emulator necessary for run

High performance

Robolectric sample

Calculon

Runs on Dalvik

More focus to unit testing than Robotium

Project in infancy, but promising

Calculon sample

Other tools

- Android Mock
  - Runs on Dalvik
  - EasyMock on Android
- Android Junit Report
  - Allows download of junit report off emulator
- Vogar/Caliper
  - Google sponsored (test) code execution and (micro) benchmarking tool
  - Targets JVM, Harmony or Dalvik

Why Continuous Integration

- Avoid “works on my machine” problems - Reproducibility
- Free up developer machine/time – I don't have time to run all tests before each commit
- No IDE dependency (less setup problems)
- Rapid feedback in team
- Improved communication

Continuous Integration

- Run build and tests for each check in
- Setup for various development branches
- Run release build as one click action
- Create website with project details as well as analysis of build history

Continuous integration servers

- Hudson/Jenkins
- Cruise Control
- Bamboo
- TeamCity
- and lots more

Example Hudson

- Easy to install
- Large community
- Android plugin
- Commercial offering as hosted
- Open source

Options for Install

- On demand on development machine
- Local networked server
- Virtual machine in cloud
- Commercial offering

simpligility technologies inc.

Installation of CI

- Headless install of Android SDK and build tools
- Install
- Configure
- Watch it run and be notified

Beyond testing

Static analysis, test coverage, trending – Sonar

Site build and more ...

Resources Testing

•Android SDK Test Tools http://developer.android.com/
•JUnit http://www.junit.org/
•TestNG http://www.testng.org/

•Robotium http://code.google.com/p/robotium/
•Robotium Samples https://github.com/jayway/robotium-samples

•Robolectric http://pivotal.github.com/robolectric/
•Robolectric Sample https://github.com/pivotal/RobolectricSample

•Calculon https://github.com/kaeppler/calculon
•Android Mock http://code.google.com/p/android-mock/
•Android Junit Report https://github.com/jsankey/android-junit-report

•Vogar http://code.google.com/p/vogar
•Caliper http://code.google.com/p/caliper

Resources Continuous Integration

Hudson
http://hudson-ci.org/

Jenkins
http://jenkins-ci.org/

CruiseControl
http://cruisecontrol.sourceforge.net/

AtlassianBamboo
http://www.atlassian.com/software/bamboo/

JetBrains TeamCity
http://www.jetbrains.com/teamcity/

and many more

Summary

Testing makes things easier

Find problems before your customers find it

Implement new features confidently without breaking existing functionality

The End

Thank you for your attention.

@simpligility

manfred@simpiligility.com