O'REILLY®

Android Open CONFERENCE

Static Analysis For Improved Application Performance And Quality

Eric Cloninger (ericc@motorola.com)

Product Line Manager, Development Tools

Motorola Mobility

Housekeeping

- bit.ly bundle for all content - http://bitly.com/nwJ1Jj

@ecmoto

@ericcloninger

Google+

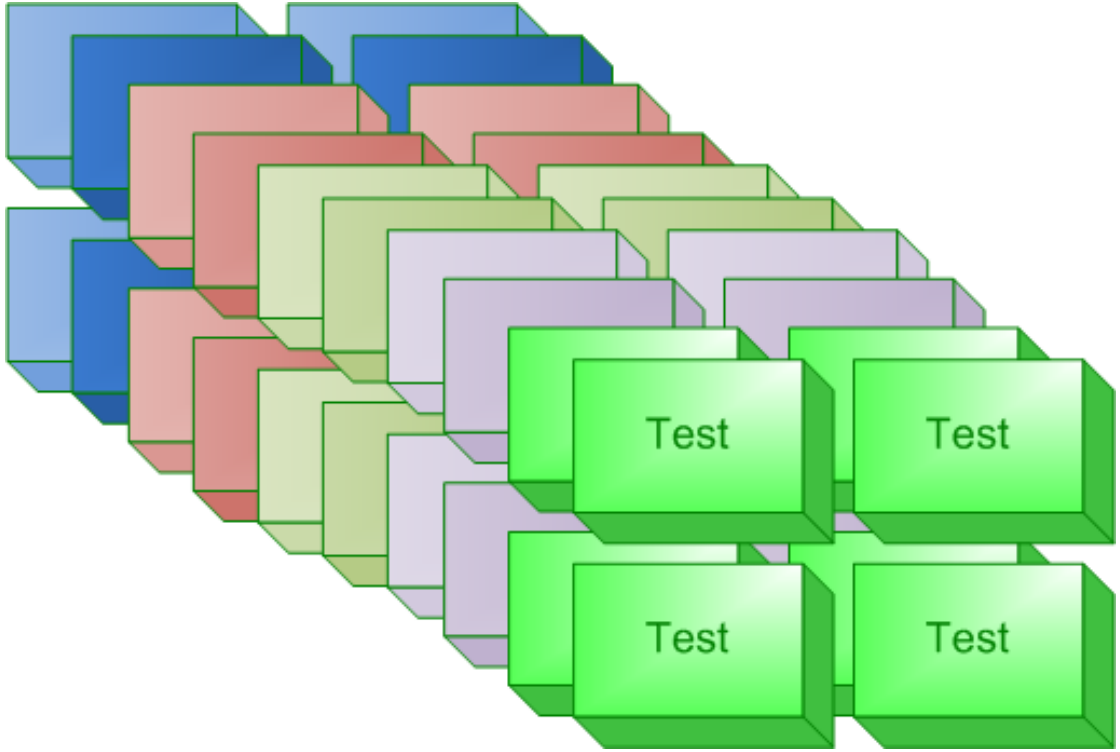
Hypothetical Example

- Existing application is to be preloaded on a new device
  - App looks great in a car dock
- Device has a new density, screen size, and a default orientation
  - *"Did we mention it has a new car dock design? You'll get a prototype before we ship."*
- Developer duplicates existing resource folders and requests designer to create images at new depth/size and with car dock
- Disaster at car dock manufacturer results in delays
- Designer returns images. No car dock images for new size.
  - Developer doesn't cross-check his initial request
- Car dock isn't in anyone's test plan
- Sales says **"SHIP IT"**

2 Months Later…

- Device sales are steady
- Engineers are working on something else
- Dock ships
- Customer buys device…
- With the car dock…
- Customer puts device in dock…
- The device fires a dock broadcast …
- Which launches the app that was never tested with the dock…
- …

Matrix of Pain



Static Analysis

“*Static Analysis is checking a story for spelling errors. Runtime Analysis is reading a story and realizing it's stupid.*”

@dougschaefer

Static Analysis

- Solves a different problem set than runtime tests
- Finds “Matrix of Pain” issues that runtime tests don’t
- Catches errors early in the development cycle
- Automatable / scriptable systems

Static Analysis Tools

- http://en.wikipedia.org/wiki/Static_program_analysis
- http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis
- Many language-specific tools and products for desktop and server platforms
- Few mobile solutions
- Fewer for our favorite platform

MOTODEV App Validator

- Free (as in beer) static analysis tool (registration req'd)
  - http://developer.motorola.com/
- Works on Mac, Windows, and Linux
- Available as an Eclipse plugin, a command line tool, and as a drag-and-drop web app
- Operates on .APK files (plugins, command line & web) as well as Eclipse projects (plugins)
- Highly configurable
- Generates output as text, CSV, or XML

MOTODEV App Validator

- Decomposes an APK into its “file system”
- Examines the app manifest for improper values
- Decompiles the Java code found in the APK using smali and deodex, and examines code patterns
- Compares parameters to API calls against the contents of the resource directory
- Compares app against a set of specs to find issues that would only occur on certain types of devices

Capabilities

- http://developer.motorola.com/docstools/library/motodev-app-validator/
- Permissions – Too few, too many, or requesting permissions not available to your app (e.g. *android.permission.BRICK)*
- Missing localized strings and drawable resources
- Market filters (missing required meta-data, certificate date, SDK level vs. screen size, permissions that imply needed features)
- Layout issues (missing IDs, duplicated IDs, improper reuse of layout IDs, missing x-large layouts for Honeycomb SDK)
- Code (opened database cursors that aren't closed)
- And more….

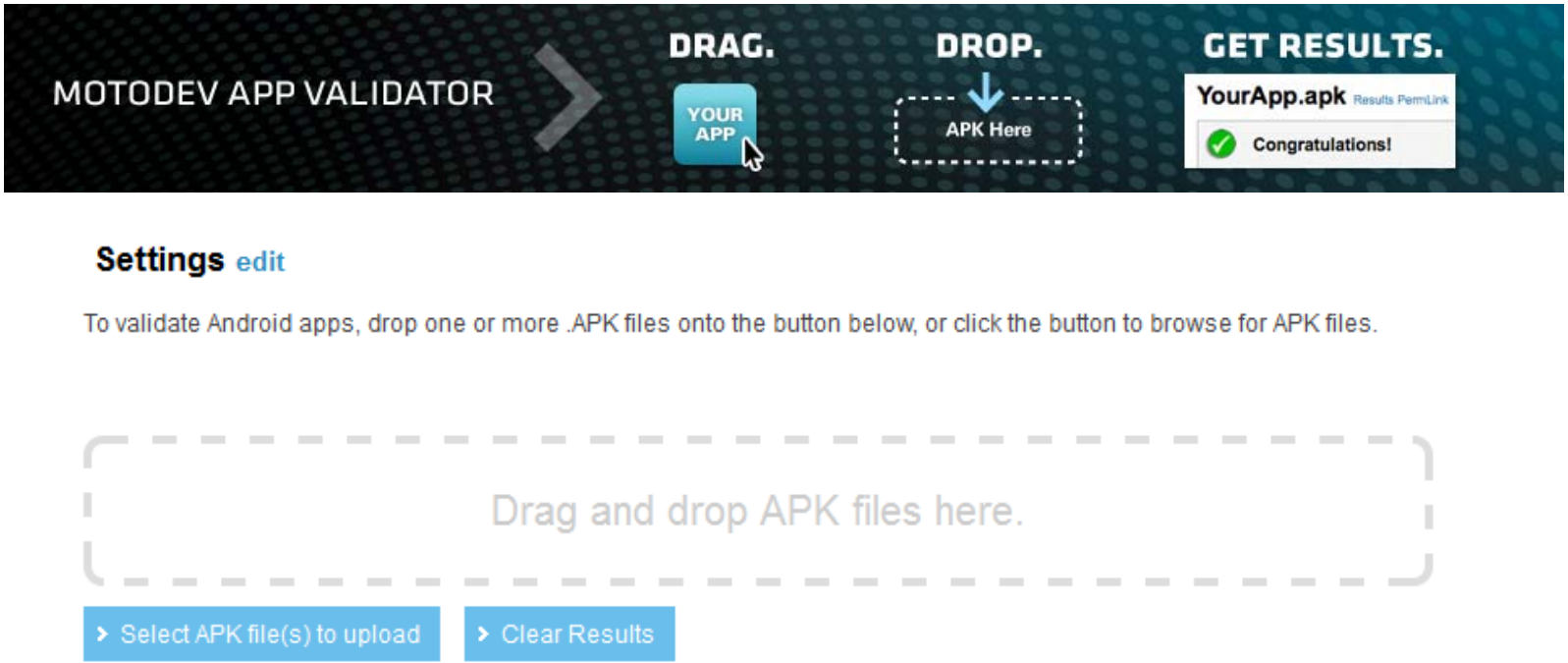
No Magic Bullet

- False positives are a very real possibility
  - Take results with a grain of salt until you can vet them
- App Validator cannot determine intent, so every result needs to be judged on its own merit
  - Increase/decrease warning levels and enable/disable checkers

DEMOS

Online App Validator

- http://developer.motorola.com/testing/app-validator/



Command Line

Usage: APPVALIDATOR [FILE][OPTION [PRM]]...

Options:

-sdk SDKPATH - Specify an Android SDK to be used during validation instead of the default one

-c CHK [PRM]... - Specify a checker to be run by supplying its ID (CHK) and specific checker parameters (PRM).

-d [DEV] - Specify a device [DEV] that will be checked against your application for incompatibilities. Use 'none' to skip device verification.

-wx [CHK[.CND]]... - Increases the warning level for all or for a particular set of checkers or conditions specified by CHK and CND IDs

-xw [CHK[.CND]]... - Decreases the warning level for all or for the particular set of checkers or conditions specified by CHK and CND IDs

-output [TYPE] - Format output to the specified type. TYPE can be [text (default) | csv | xml].

-limit [COUNT] - Limit the number of output entries.

-list-checkers - List available checkers

-list-devices - List available devices

-describe-device [DEV] - Show information about a specific device

-help [CHK] - See information about general usage or about a checker by specifying its ID (CHK)

-vN - Specify verbosity level, where N is one of: 0 (default, most succinct), 1, 2 (most verbose)

-wN - Specify warning level, where N is one of: 0 (most severe warnings only), 1, 2 (default), 3, 4 (all warnings)

Implementation Details

- “Headless” Eclipse application uses OSGi for plugin loading
- A “condition” is a single issue that can occur and be reported upon
- Groups of related conditions are implemented as an Eclipse plugin called a “checker”, which can be disabled, elevated, or decreased at will
- Source is closed ☹ (but….)

Coming attractions

- More code-based checkers
- Anything obvious from Ice Cream Sandwich
- SDK (a.k.a. Eclipse extensions) available **soon** so you can write your own checkers

Get it!

- Today, it's delivered with MOTODEV Studio or as a part of our plugins archive
  - http://developer.motorola.com/docstools/motodevstudio/
  - http://developer.motorola.com/docstools/motodevstudio/download/MOTODEV_Studio_for_Android_3_0_0_Archive.zip/
  - <install_dir>/app_validator[.bat | .sh]
- Next release, separate installer
- Ask Questions on our discussion boards
  - http://community.developer.motorola.com/mtrl/

THANK YOU!