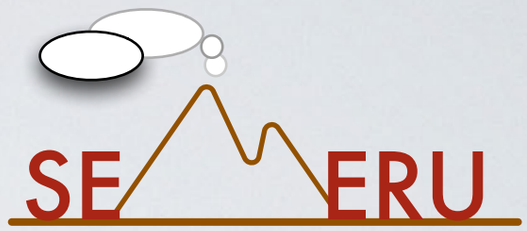


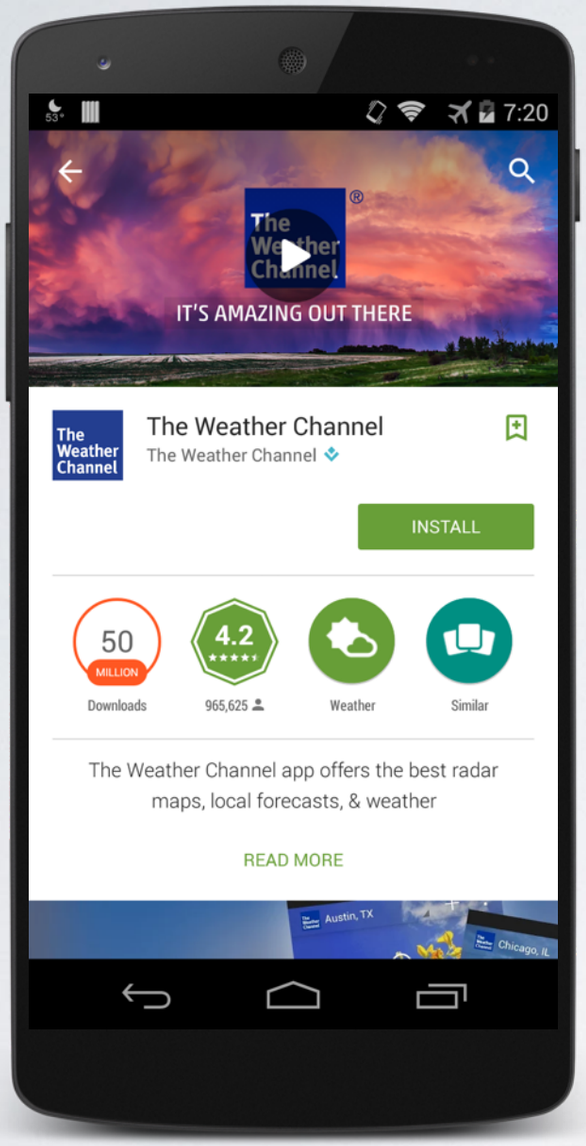
&

College of William & Mary - SEMERU - Department of Computer Science

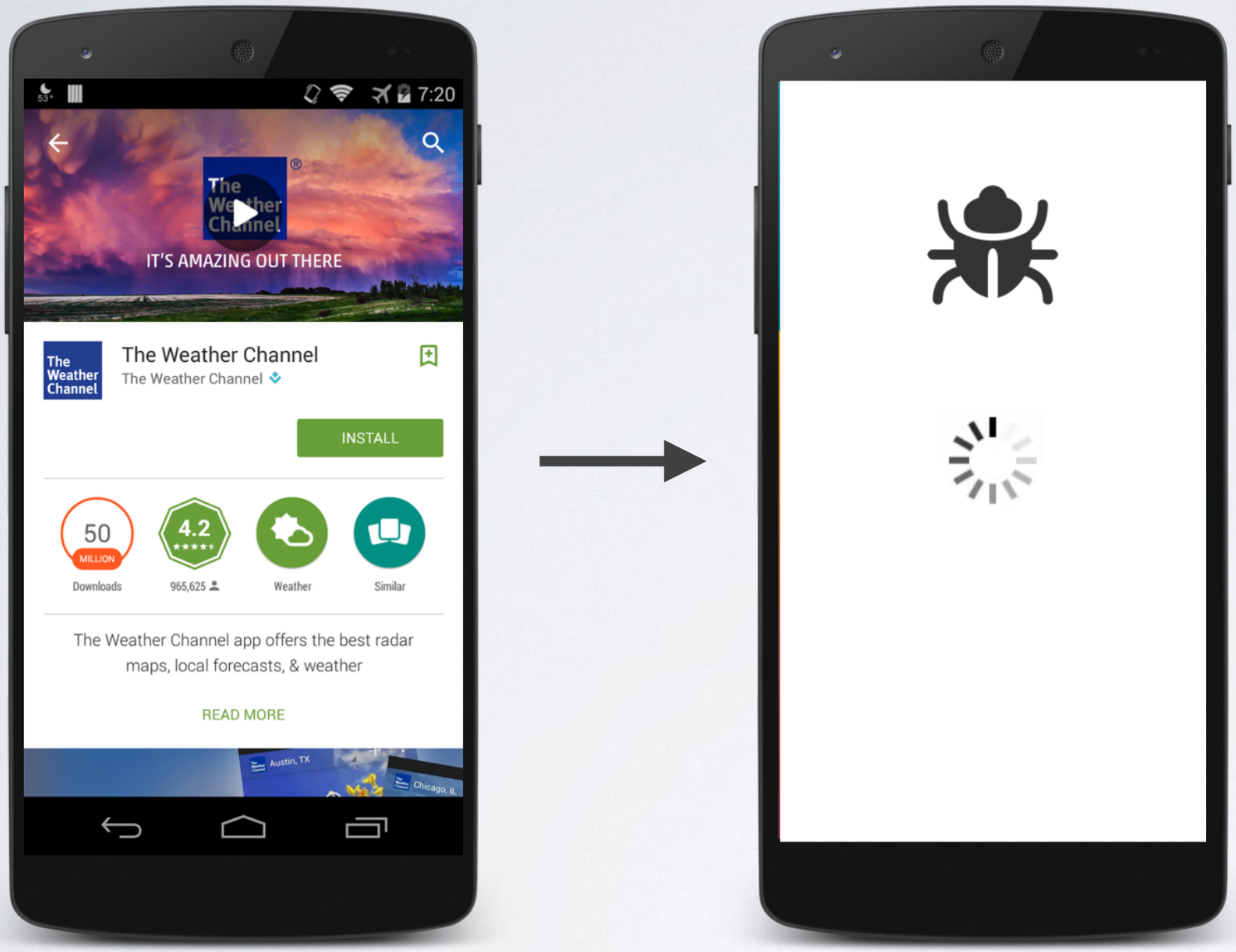
FUSION: A Tool for Facilitating and Augmenting Android Bug Reporting

Kevin Moran,
Mario Linares-Vásquez,
Carlos Bernal-Cárdenas,
and Denys Poshyvanyk

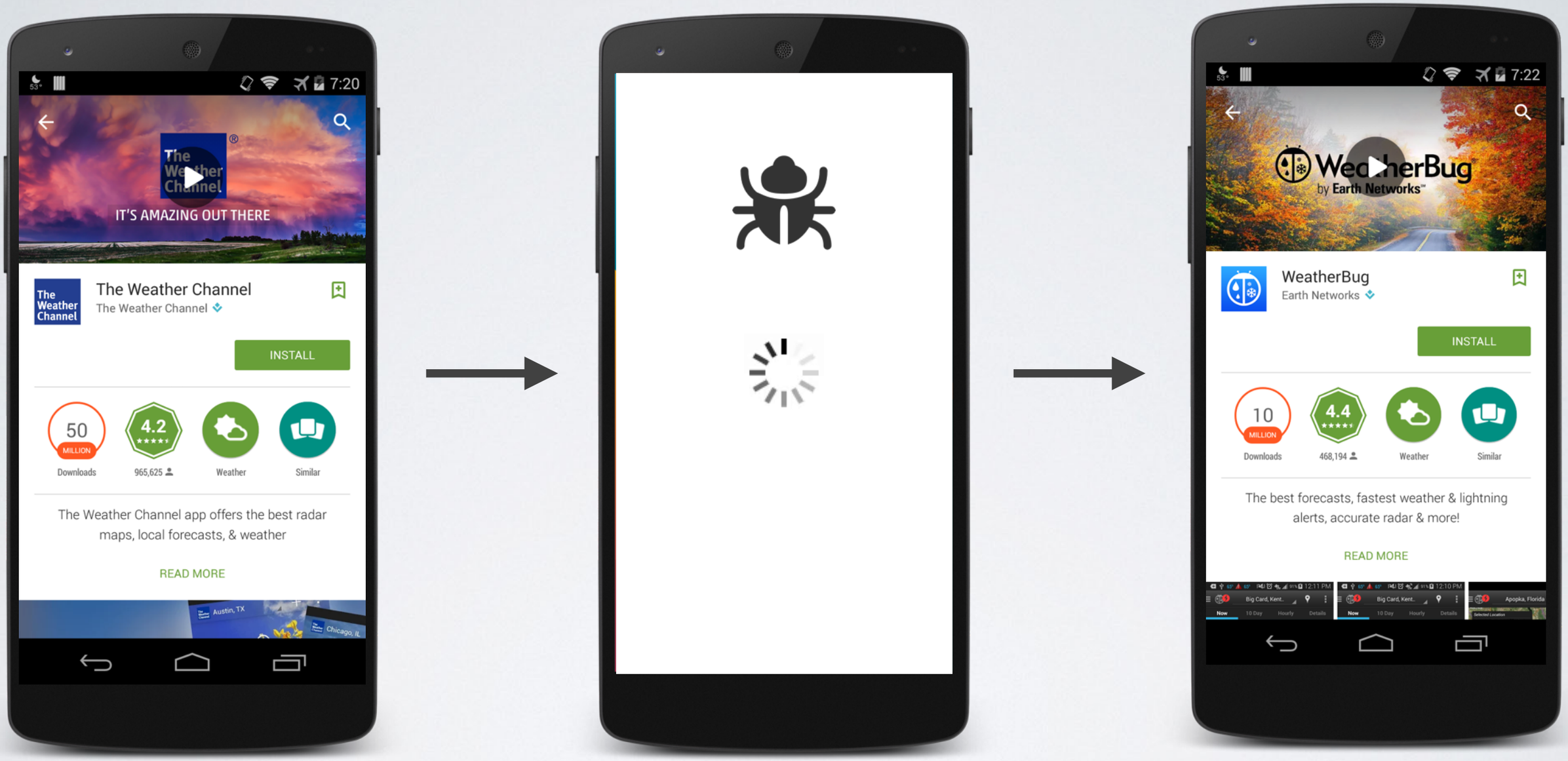
ICSE16
Austin, TX
Doctoral Symposium
Wednesday, May 18th, 2016



Images Courtesy of Google



Images Courtesy of Google



Images Courtesy of Google

"If dissatisfied with the performance of a mobile app, 48 percent of users would be less likely to use the app again."

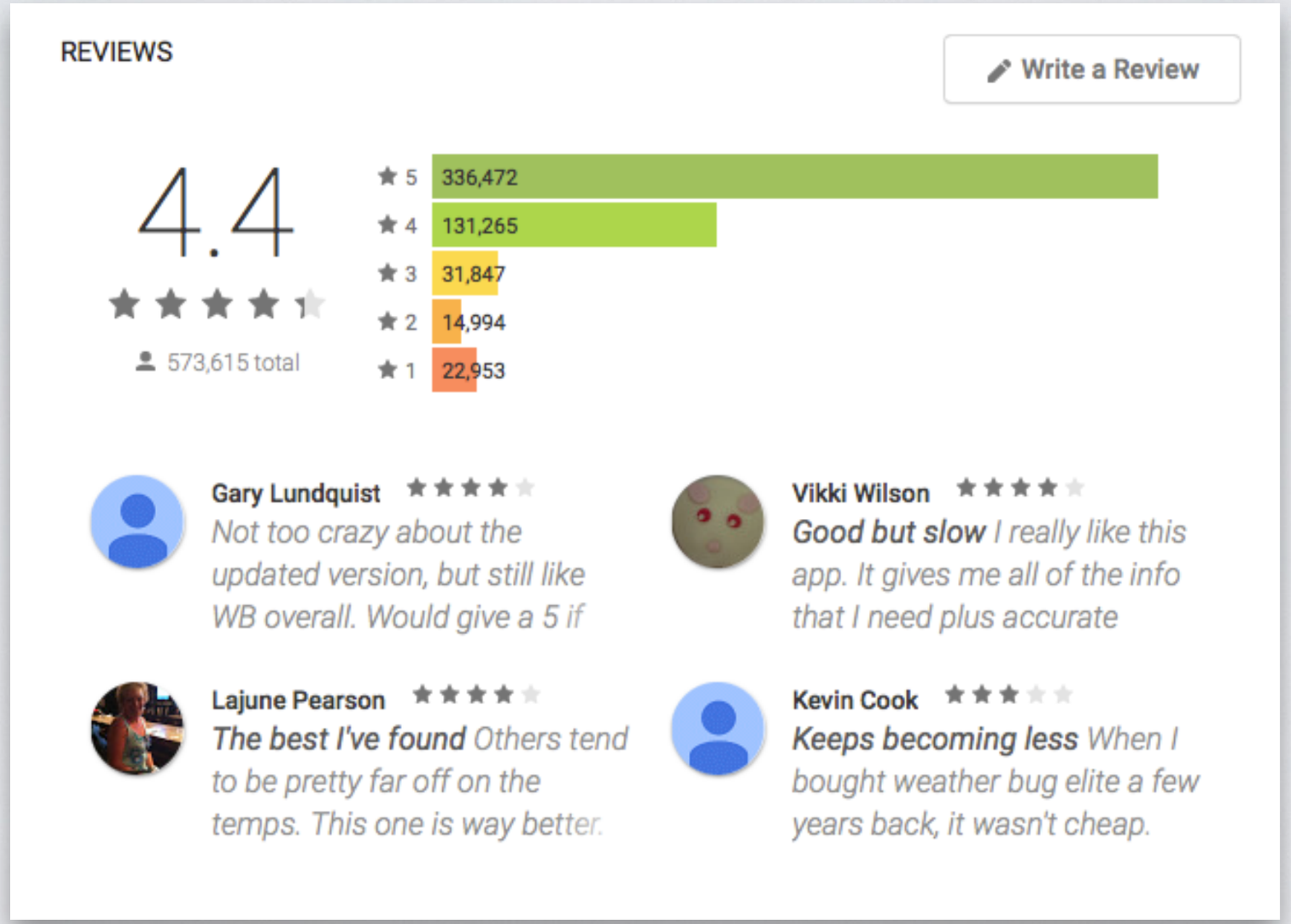
"Dynatrace Mobile App Survey Report" - https://info.dynatrace.com/rs/compuware/images/Mobile_App_Survey_Report.pdf

Images Courtesy of Google

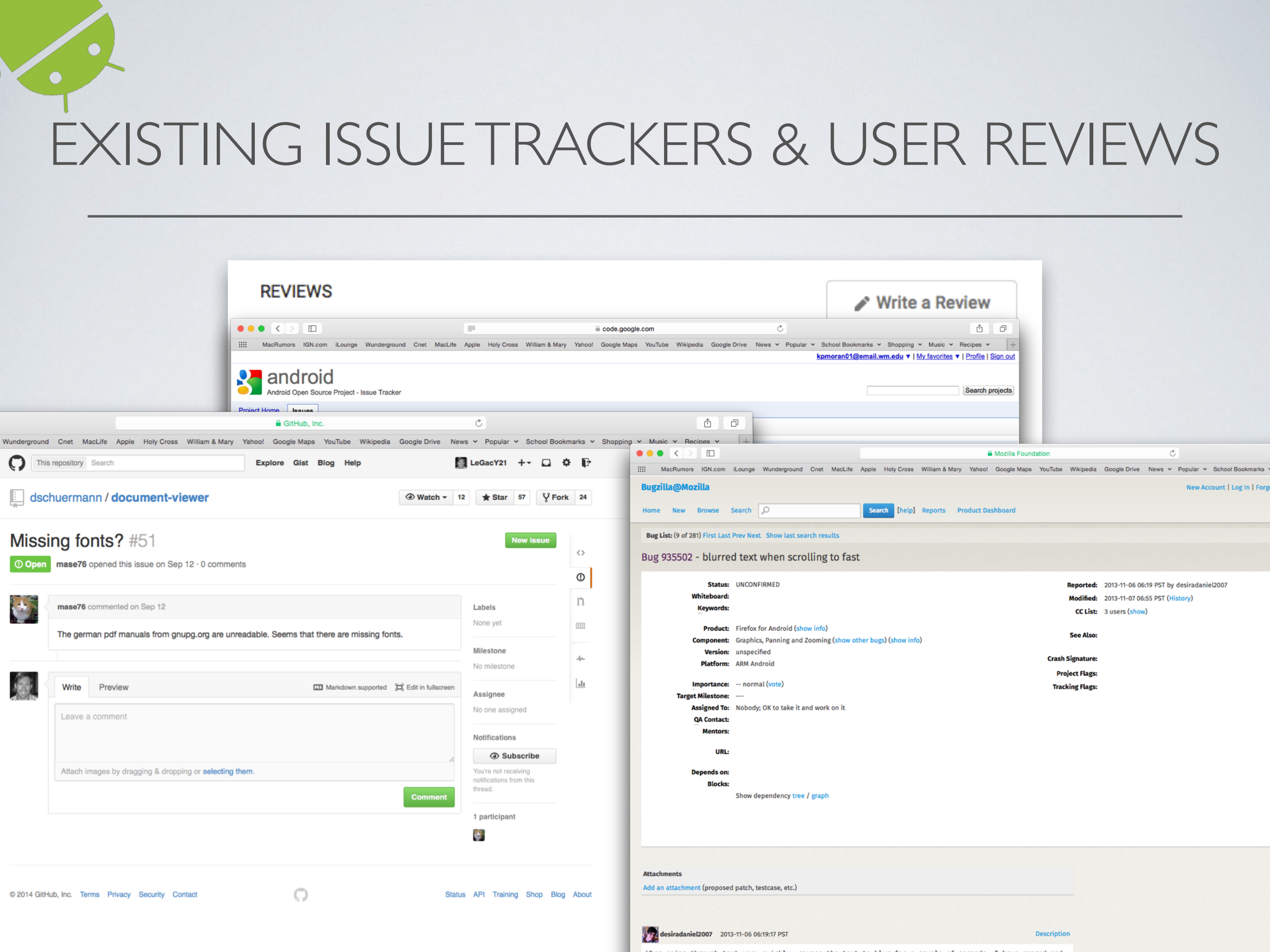
Software maintenance, specifically the prompt resolution of bug reports, is extremely important to an application's success.

Images Courtesy of Google

EXISTING ISSUE TRACKERS & USER REVIEWS

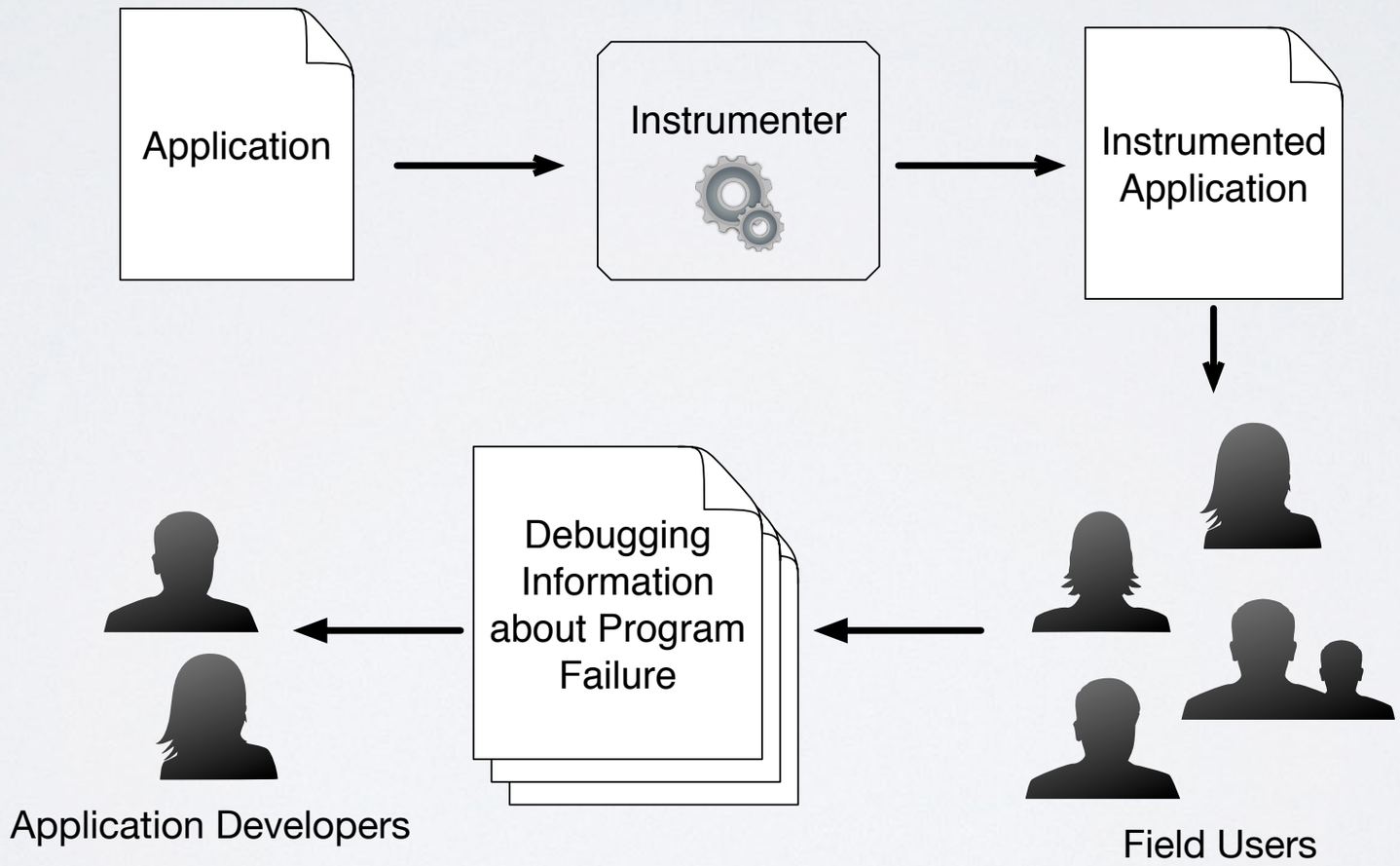


EXISTING ISSUE TRACKERS & USER REVIEWS



IN-FIELD FAILURE REPRODUCTION

- Allows for in-house debugging of field failures[1].



James Clause and Alessandro Orso. 2007. A Technique for Enabling and Supporting Debugging of Field Failures. In Proceedings of the 29th international conference on Software Engineering (ICSE '07)

IN-FIELD FAILURE REPRODUCTION

IN-FIELD FAILURE REPRODUCTION

- Requires potentially expensive program instrumentation, not suitable for a mobile environment.
- Requires oracles in order to capture and reproduce failures in the field.
- May not be easily adaptable to the event-driven and fragmented nature of mobile apps.

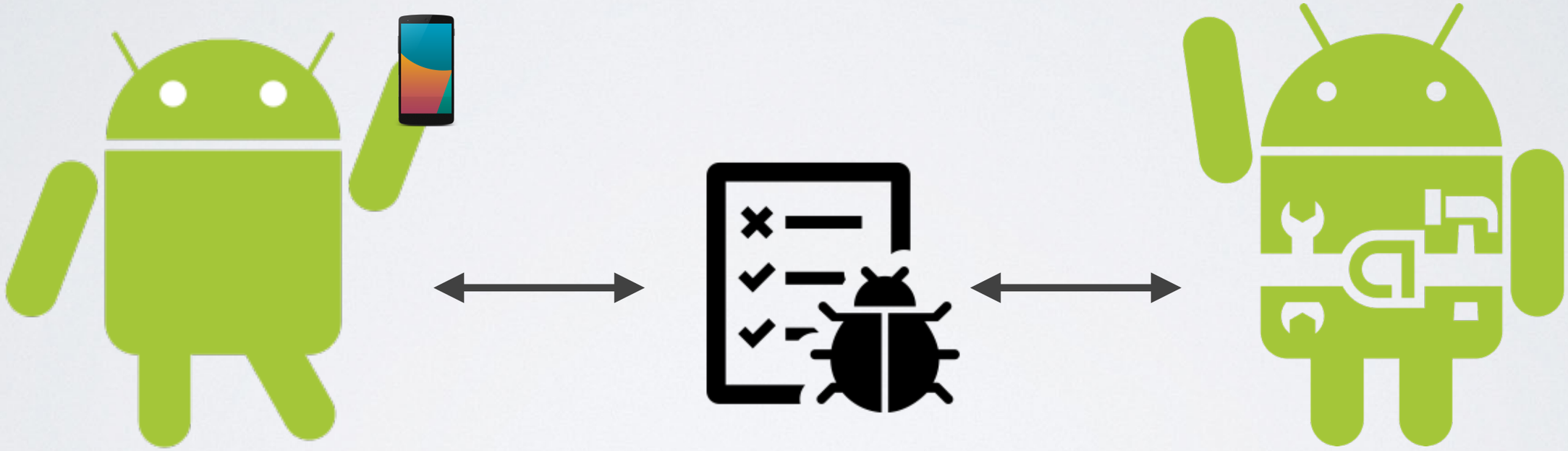
THE LEXICAL GAP IN BUG REPORTING

Reporters:

-Functional Knowledge of a Software Bug.

Developers:

-Intimate Code Level Knowledge of Application



Images Courtesy of Google

THE LEXICAL GAP IN BUG REPORTING

Reporters:

-Functional Knowledge of a Software Bug.

Developers:

-Intimate Code Level Knowledge of Application

Inherent Lexical Gap

Images Courtesy of Google

WHAT MAKES A GOOD BUG REPORT ?

- ***Insufficient information*** in bug reports is one of the leading causes of non-reproducible reports[1]
- Developers consider (i) ***steps to reproduce***, (ii) ***stack traces***, and (iii) ***test cases/scenarios*** as the most helpful sources of information in bug reports[2]
- Information needs are ***greatest*** earliest in a bug's lifecycle[3]

---

[1] *M. Erfani Joorabchi, M. Mirzaaghaei, and A. Mesbah. Works for me! characterizing non-reproducible bug reports. MSR 2014,*

[2] *N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann. What makes a good bug report? (SIGSOFT '08/FSE-16),*

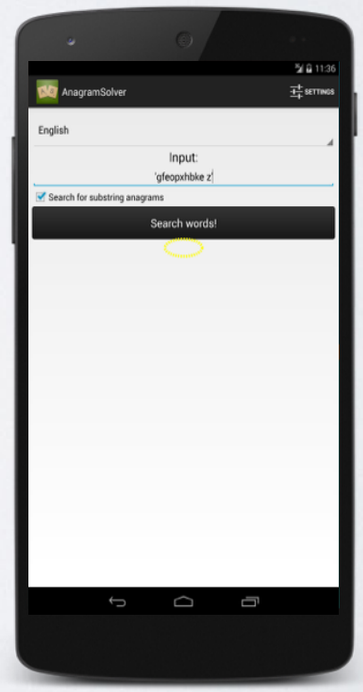
[3] *S. Breu, R. Premraj, J. Sillito, and T. Zimmermann. Information needs in bug reports: Improving cooperation between developers and users. (CSCW)*

FUSION: THE KEY IDEA

Static Code Analysis

```
private void createNote() {
    Intent i = new Intent(this, QuoteEdit.class);
     startActivityForResult(i, ACTIVITY_CREATE);
}

@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    super.onListItemClick(l, v, position, id);
    Cursor c = mNotesCursor;
    c.moveToPosition(position);
    Intent i = new Intent(this, QuoteEdit.class);
    i.putExtra(QuotesDBAdapter.KEY_ROWID, id);
    i.putExtra(QuotesDBAdapter.KEY_QUOTES, c.getString(
            c.getColumnIndexOrThrow(QuotesDBAdapter.KEY_QUOTES)));
    startActivityForResult(i, ACTIVITY_EDIT);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    Bundle extras = intent.getExtras();
    switch(requestCode) {
    case ACTIVITY_CREATE:
        String title = extras.getString(QuotesDBAdapter.KEY_QUOTES);
        mDbHelper.createQuote(title);
        fillData();
        break;
    case ACTIVITY_EDIT:
        Long rowId = extras.getLong(QuotesDBAdapter.KEY_ROWID);
        if (rowId != null) {
            String editTitle = extras.getString(QuotesDBAdapter.KEY_QUOTES);
            mDbHelper.updateQuote(rowId, editTitle);
        }
        fillData();
        break;
    }
}
```

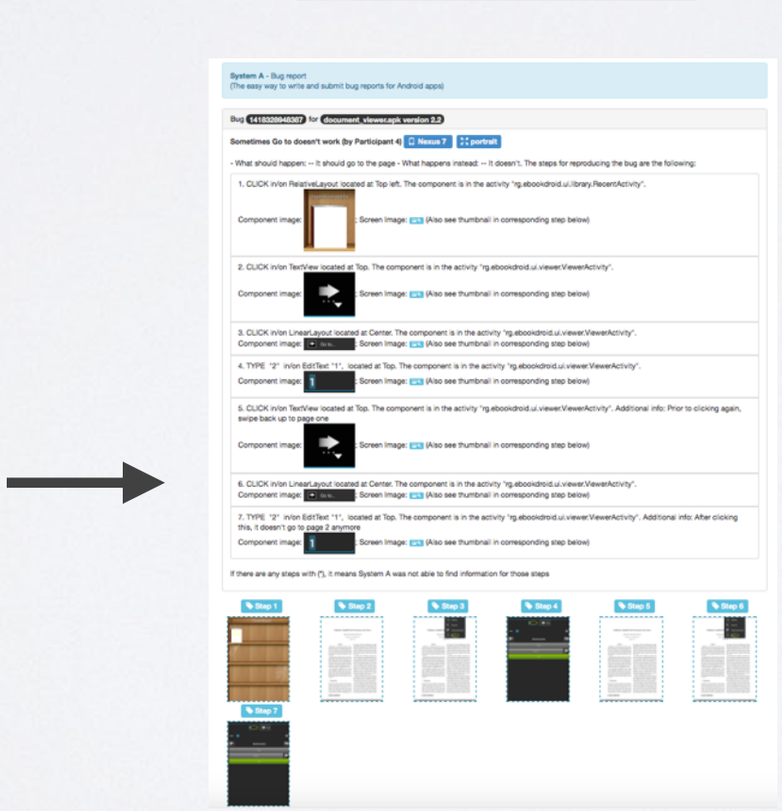
Dynamic Analysis



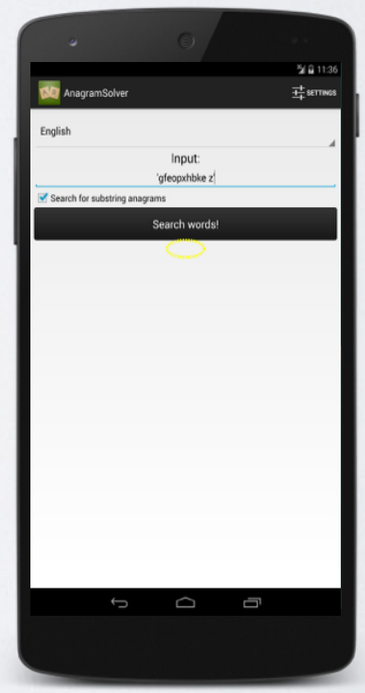
FUSION: THE KEY IDEA

Static Code Analysis

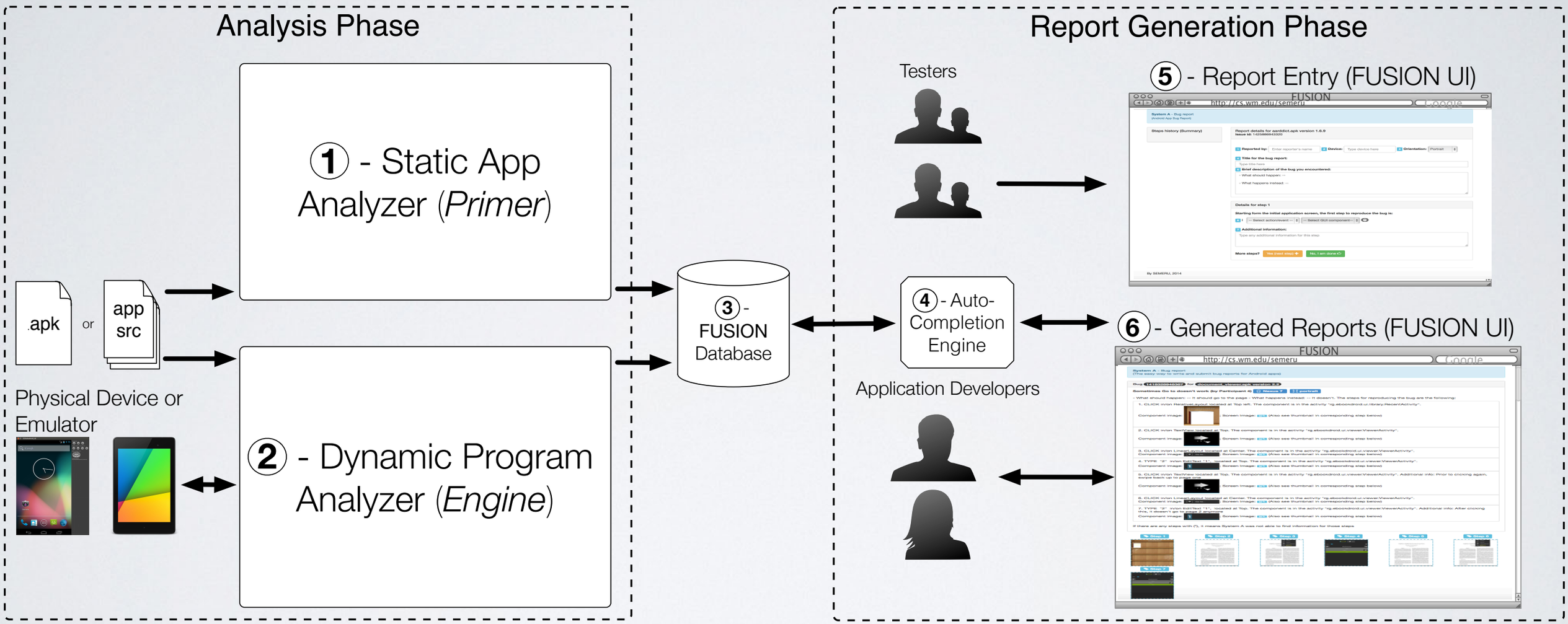
```
private void createNote() {
    Intent i = new Intent(this, QuoteEdit.class);
     startActivityForResult(i, ACTIVITY_CREATE);
}

@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    super.onListItemClick(l, v, position, id);
    Cursor c = mNotesCursor;
    c.moveToPosition(position);
    Intent i = new Intent(this, QuoteEdit.class);
    i.putExtra(QuotesDBAdapter.KEY_ROWID, id);
    i.putExtra(QuotesDBAdapter.KEY_QUOTES, c.getString(
            c.getColumnIndexOrThrow(QuotesDBAdapter.KEY_QUOTES)));
    startActivityForResult(i, ACTIVITY_EDIT);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    Bundle extras = intent.getExtras();
    switch(requestCode) {
    case ACTIVITY_CREATE:
        String title = extras.getString(QuotesDBAdapter.KEY_QUOTES);
        mDbHelper.createQuote(title);
        fillData();
        break;
    case ACTIVITY_EDIT:
        Long rowId = extras.getLong(QuotesDBAdapter.KEY_ROWID);
        if (rowId != null) {
            String editTitle = extras.getString(QuotesDBAdapter.KEY_QUOTES);
            mDbHelper.updateQuote(rowId, editTitle);
        }
        fillData();
        break;
    }
}
```



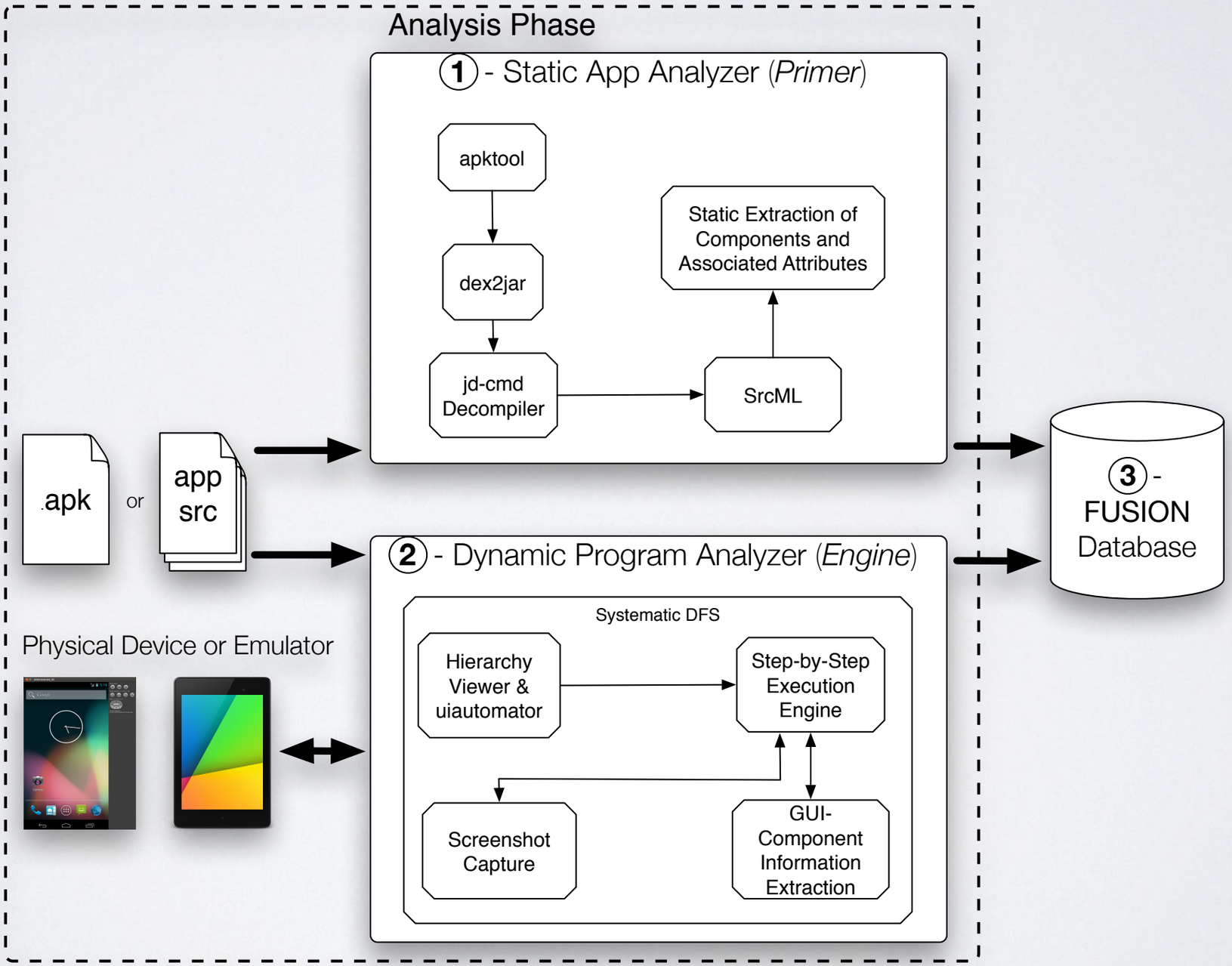
Dynamic Analysis



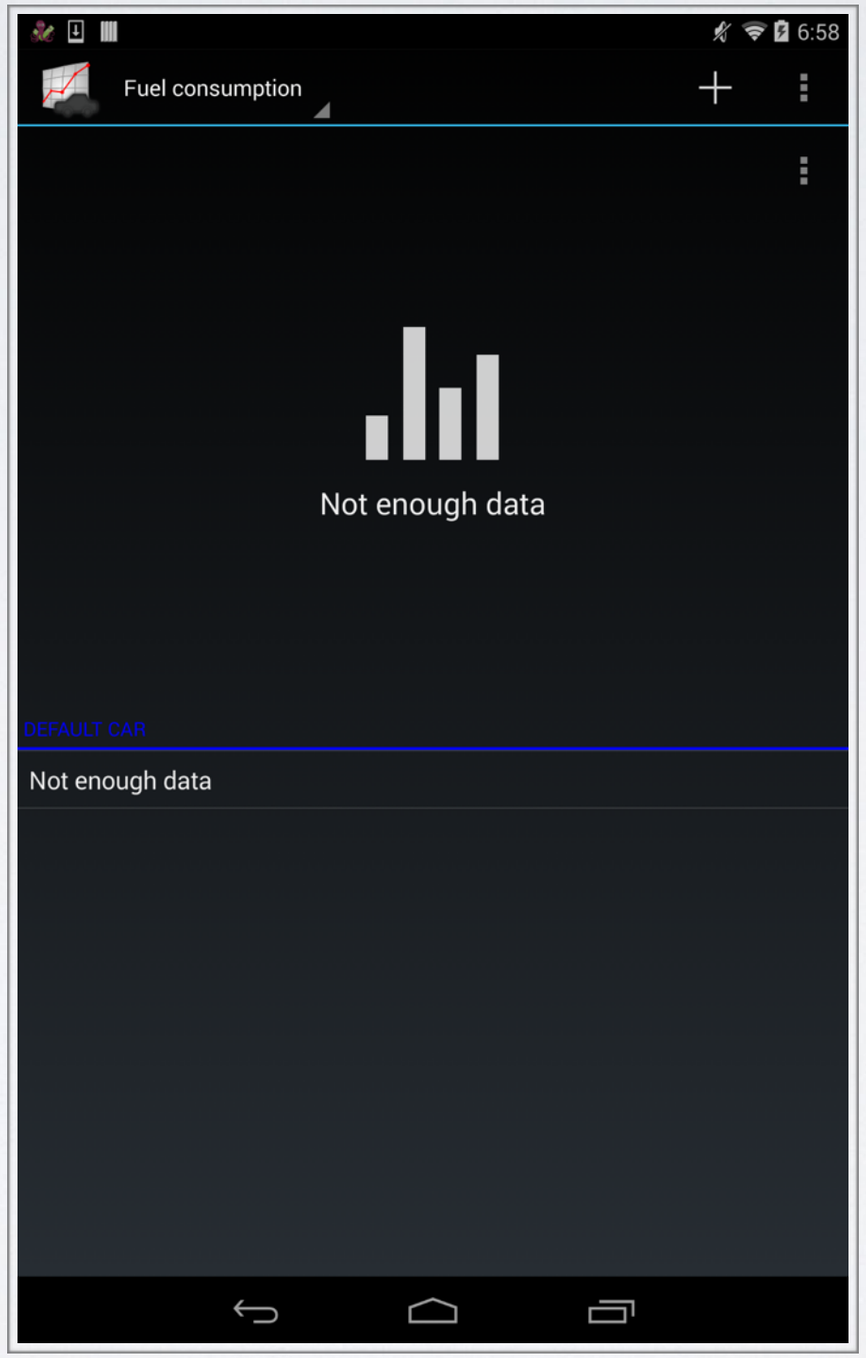
FUSION: OVERVIEW



FUSION: ANALYSIS PHASE

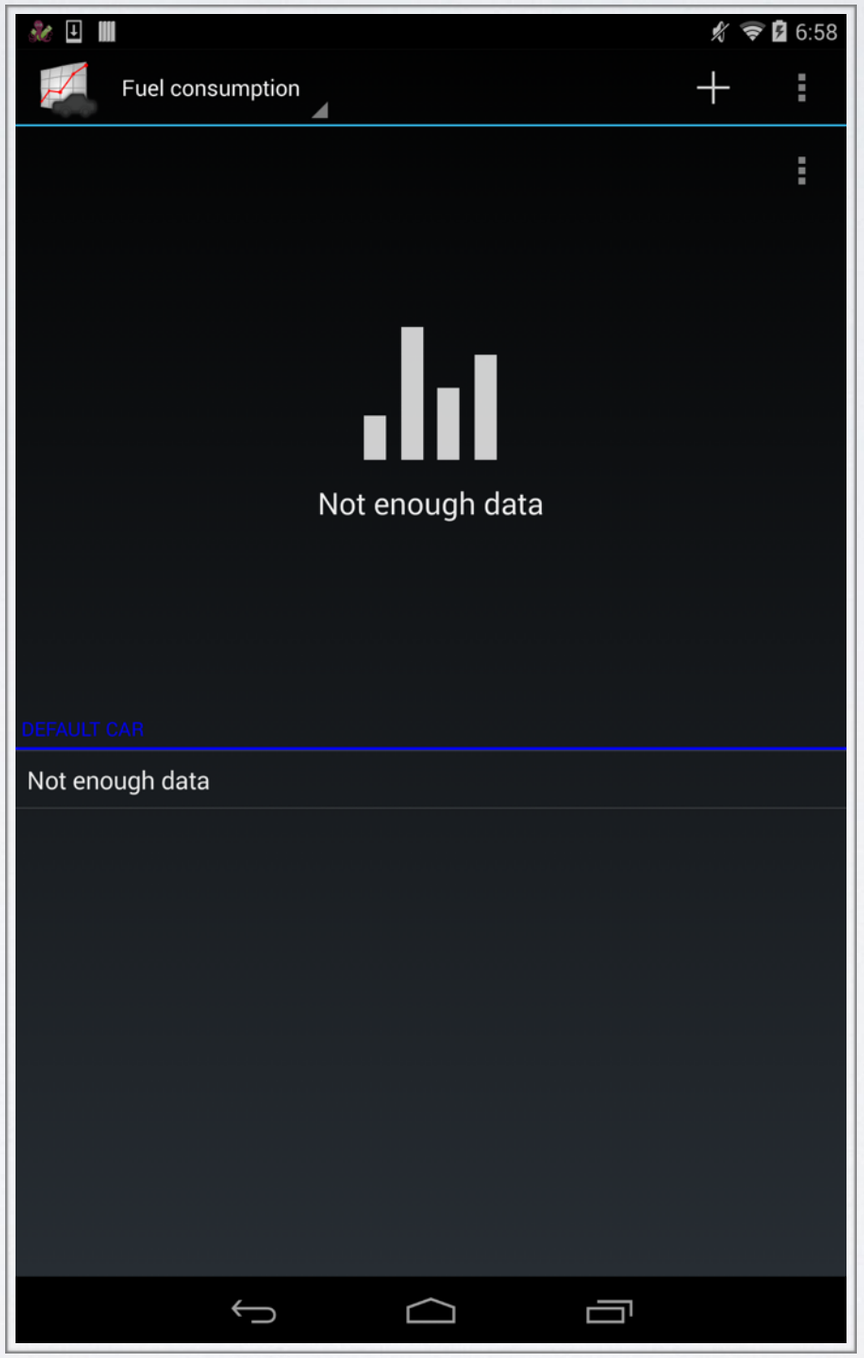


FUSION: OVERVIEW



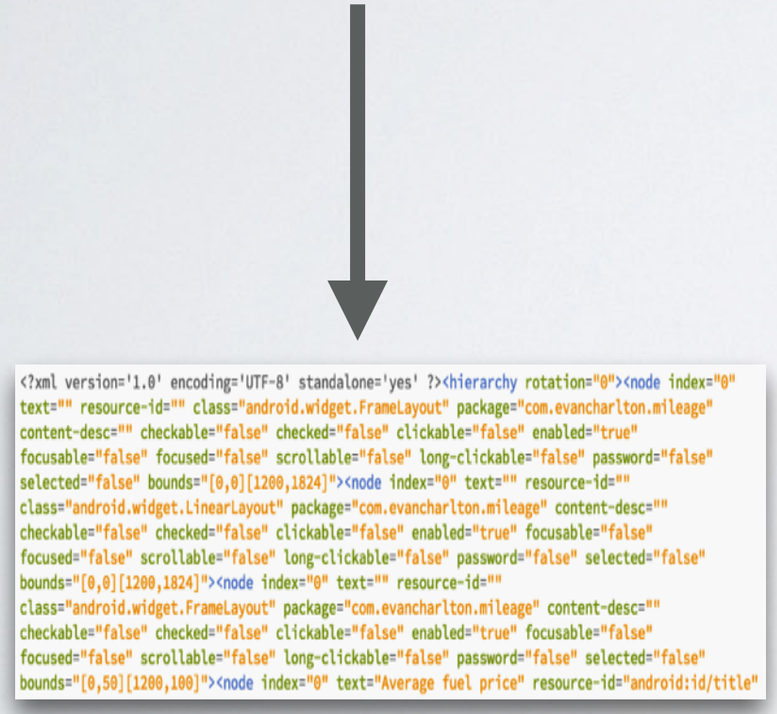
FUSION: OVERVIEW

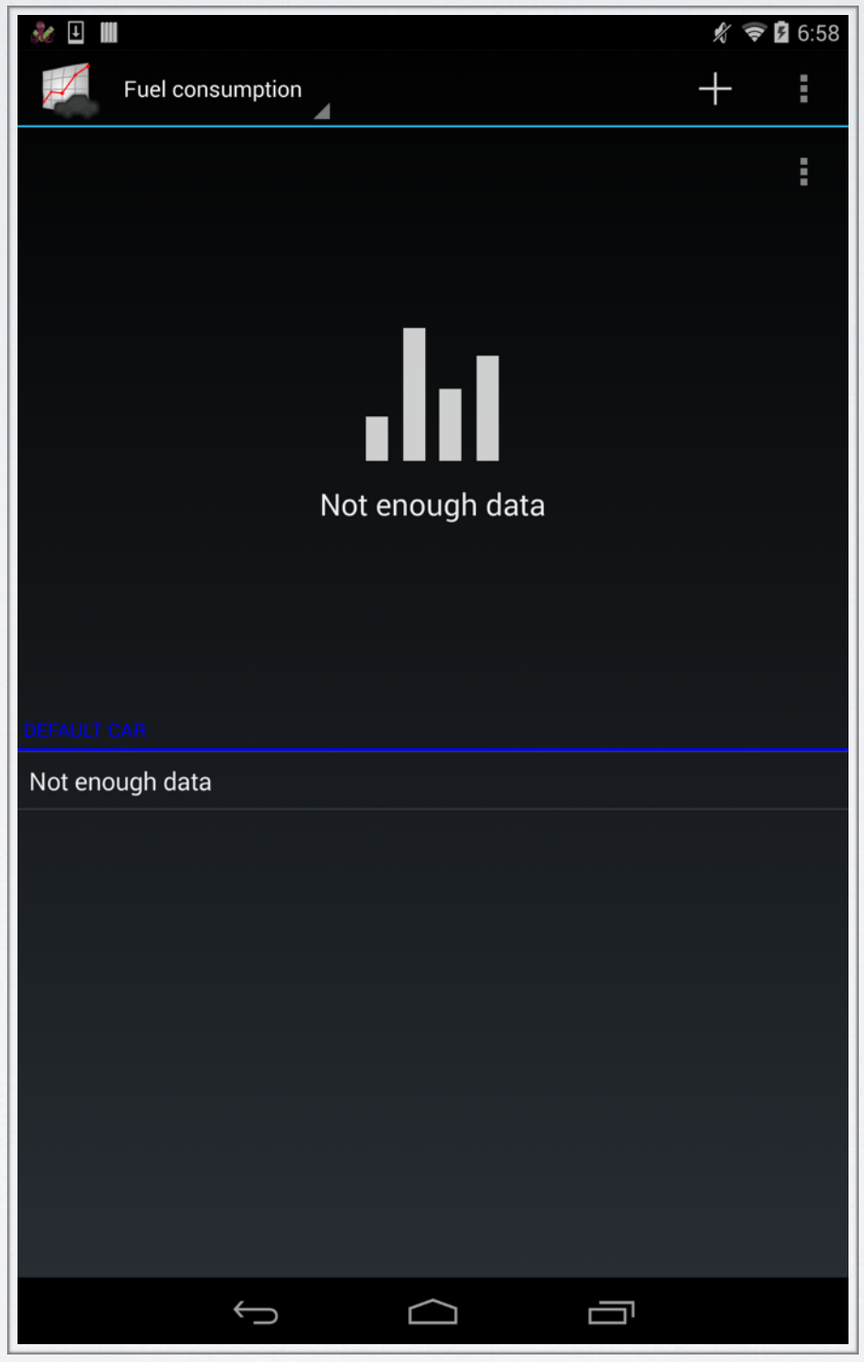
uiautomator



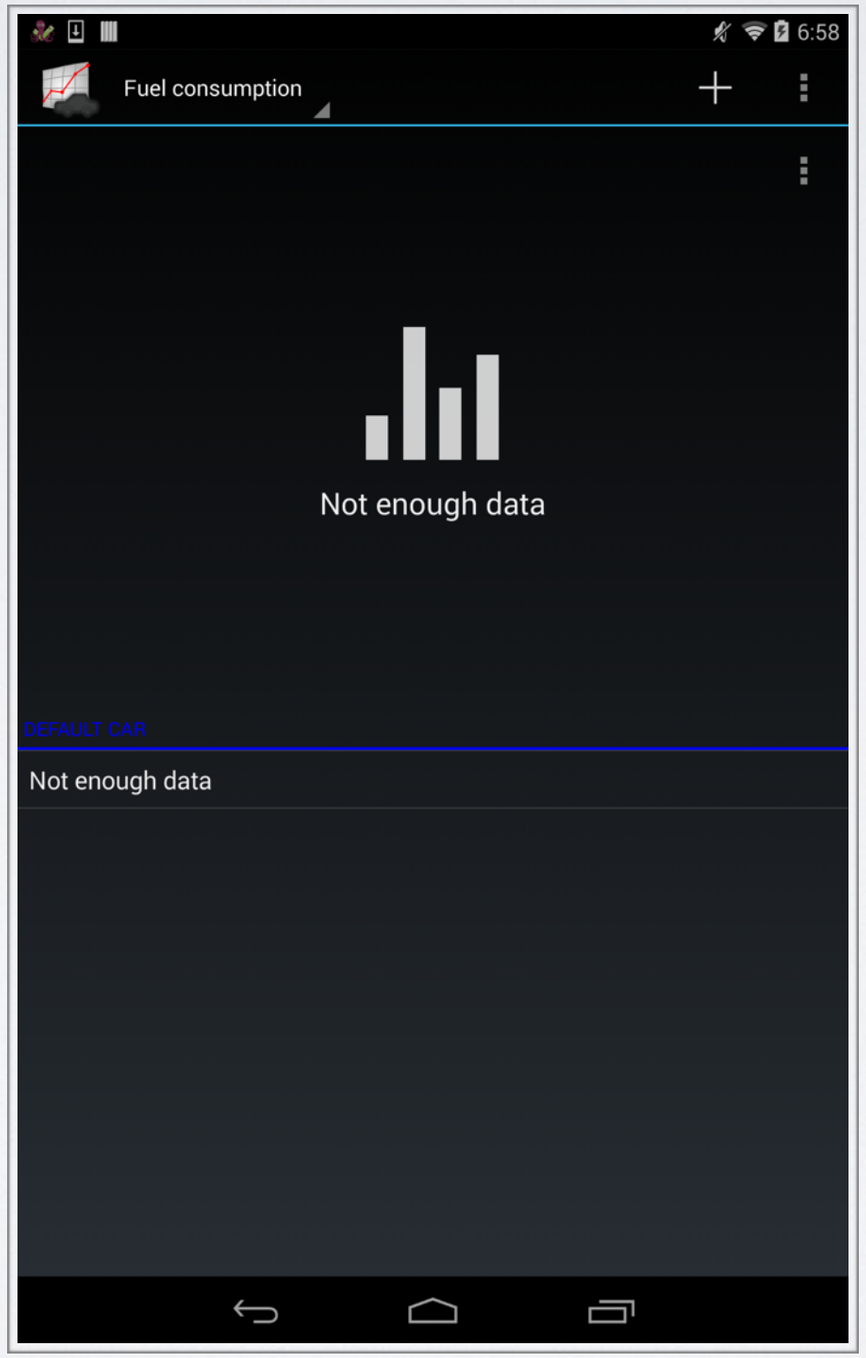
FUSION: OVERVIEW

uiautomator

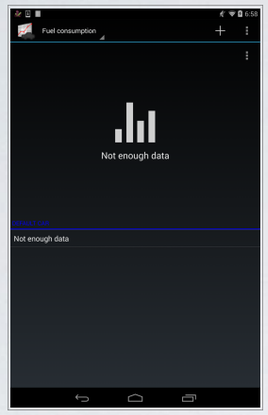




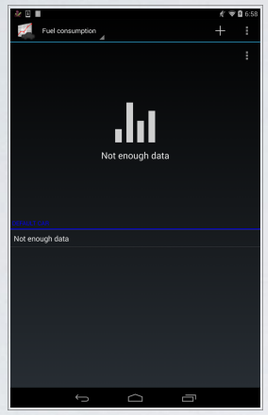
FUSION: OVERVIEW



FUSION: OVERVIEW

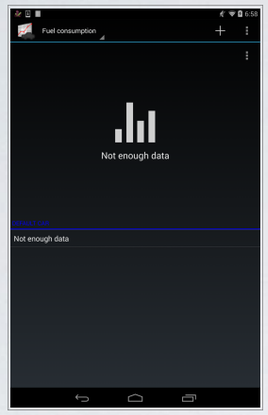


FUSION: OVERVIEW

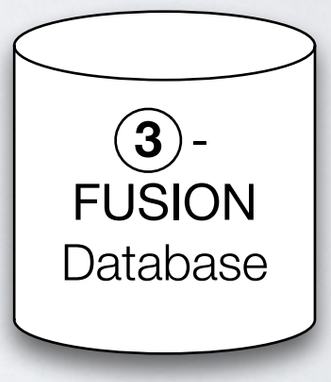


- Activity
- Checkable, Checked, Clickable, Long Clickable?
- Component Index
- Current Window
- Enabled?
- XML_ID
- Component Type
- Position (Absolute and Relative)
- Text
- Screenshot →

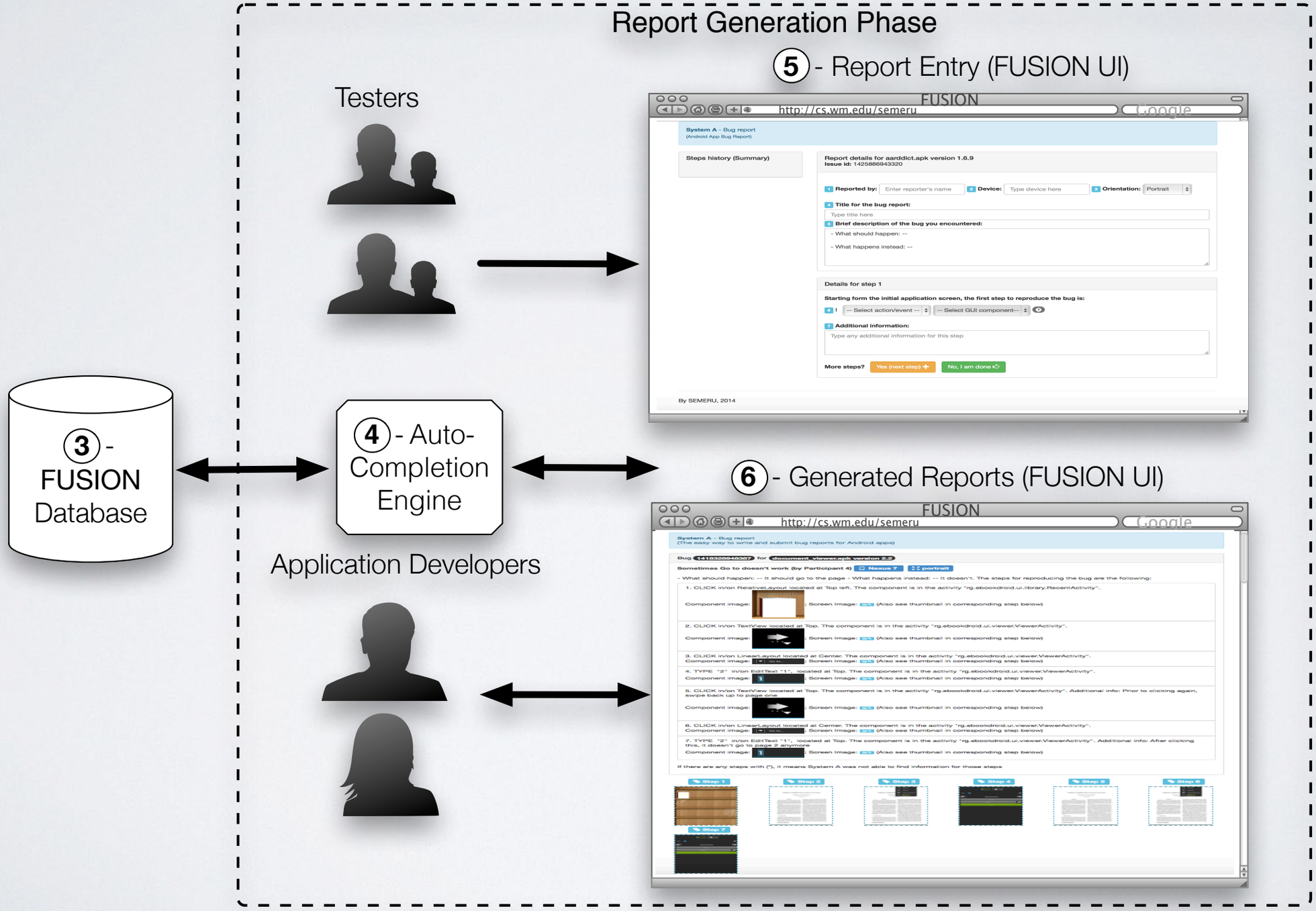
FUSION: OVERVIEW



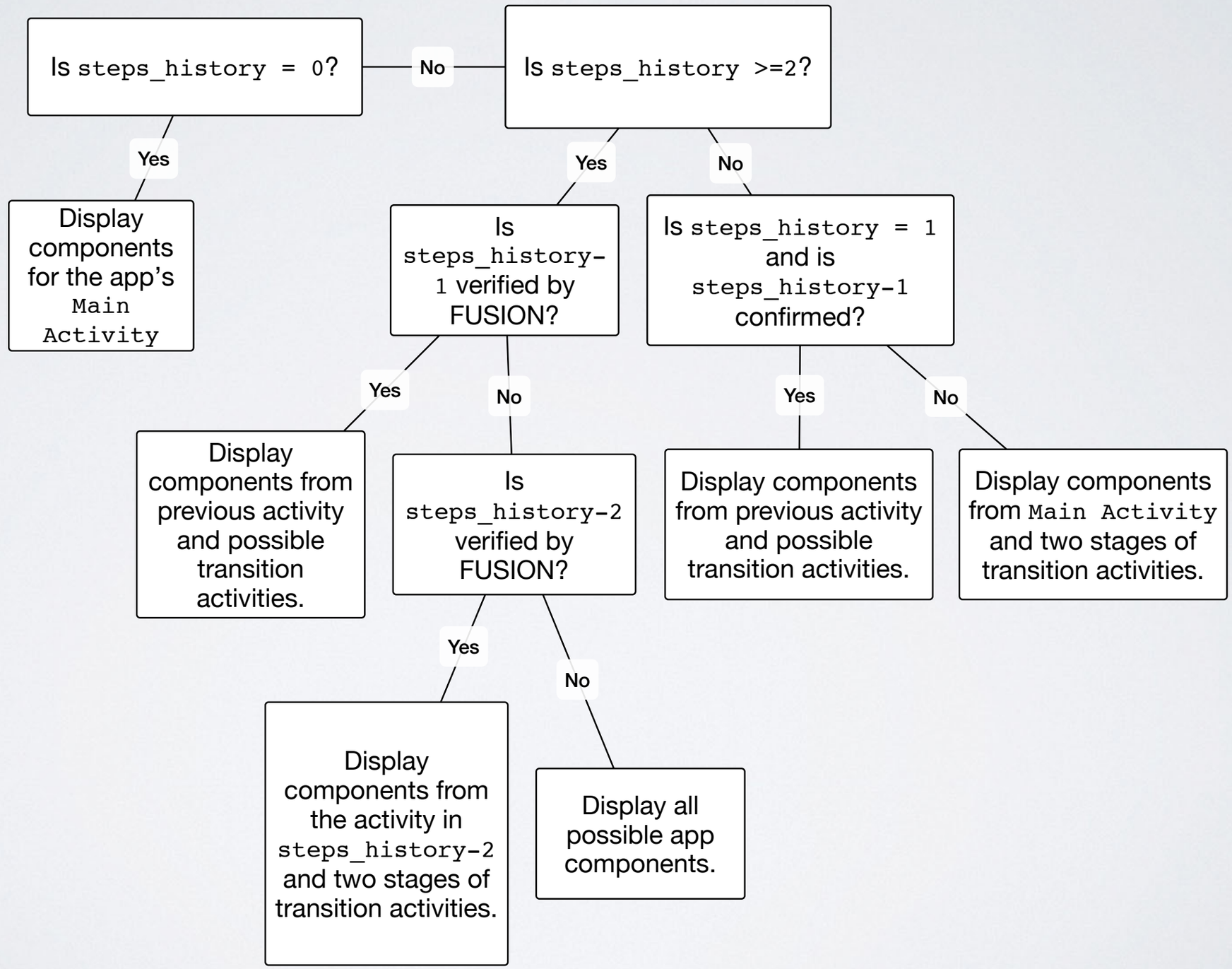

```
'73', 'me.kuehle.carreport/.gui.ReportActivity', '0', '0', '1', '1',
'me.kuehle.carreport/.gui.ReportActivity#null.FrameLayout.FrameLayout.LinearLayout.FrameLayout
.Spinner.CheckedTextView.TextView.ImageButton.FrameLayout.FrameLayout.ImageButton.ListView.Lin
earLayout', '1', '1', '0', '64', 'me.kuehle.carreport:id/btnMenu', '0',
'android.widget.ImageButton', '0', '1104', '194', 'TopRight', '0', '0', '', '64',
'me.kuehle.carreport__1_gui.png'
```



FUSION: REPORT GENERATION PHASE

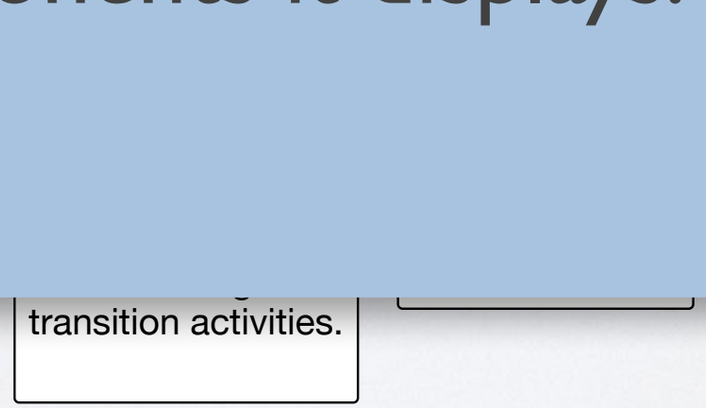


FUSION: AUTO-COMPLETION ENGINE



FUSION: AUTO-COMPLETION ENGINE

- FUSION tracks the users location in the app's event-flow.
- Suggests only components from the current screen, and possible transition screens, based on the last action.
- If steps cannot be autocompleted, FUSION expands the number of components it displays.



Report details for aarddict.apk version 1.6.9
Issue id: 1441125984977

1 **Reported by:** Kevin Moran

2 **Device:** Type device here

3 **Orientation:** Portrait

4 **Title for the bug report:**

Type title here

5 **Brief description of the bug you encountered:**

- What should happen: --

- What happens instead: --

Report details for aarddict.apk version 1.6.9
Issue id: 1441125984977

1 **Reported by:** Kevin Moran

2 **Device:** Type device here

3 **Orientation:** Portrait

4 **Title for the bug report:**

Type title here

5 **Brief description of the bug you encountered:**

- What should happen: --

- What happens instead: --

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

Details for step 1

Starting form the initial application screen, the first step to reproduce the bug is:

6 I Clicked -- Select GUI component--

7 Additional information:

Type any additional information for this step

More steps? Yes (next step) + No, I am done

Details for step 1

Starting form the initial application screen, the first step to reproduce the bug is:

6 I Clicked -- Select GUI component--

7 Additional information:

Type any additional information for this step

More steps? Yes (next step) + No, I am done

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

2. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".

Component image: 
; Screen Image: (Also see thumbnail in corresponding step below)

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

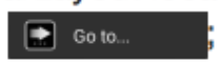
1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

2. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".

Component image: 
; Screen Image: (Also see thumbnail in corresponding step below)

3. CLICK in/on LinearLayout located at Center. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

Details for step 4

6 I -- Select action/event -- -- Select GUI component-- ?

7 Additional information:

Type any additional information for this step

More steps? Yes (next step) + No, I am done

Details for step 4

6 I -- Select action/event -- -- Select GUI component-- ?

7 Additional information:

Type any additional information for this step

More steps? Yes (next step) + No, I am done

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

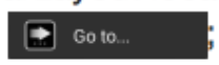
1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

2. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".

Component image: 
; Screen Image: (Also see thumbnail in corresponding step below)

3. CLICK in/on LinearLayout located at Center. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

4. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) **Nexus 7** **portrait**

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

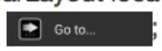
1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

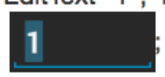
2. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".

Component image: 
; Screen Image: (Also see thumbnail in corresponding step below)

3. CLICK in/on LinearLayout located at Center. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

4. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

5. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity". Additional info: Prior to clicking again, swipe back up to page one

Component image: 
; Screen Image: (Also see thumbnail in corresponding step below)

7. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity". Additional info: After clicking this, it doesn't go to page 2 anymore

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

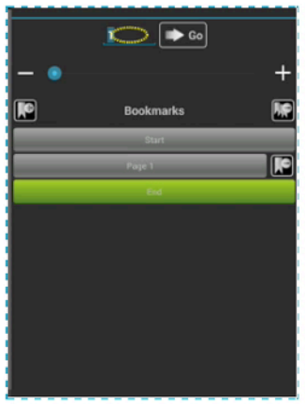
If there are any steps with (*), it means System A was not able to find information for those steps

Step 1

Step 2

Step 3

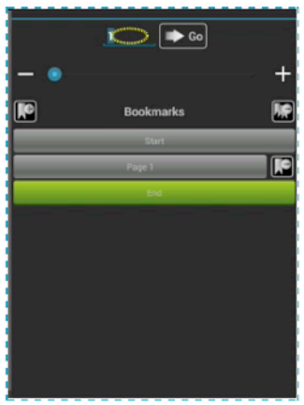
Step 4



Step 5

Step 6

Step 7



Back to Fusion | Print report | Go to viewer

By SEMERU, 2014

7. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity". Additional info: After clicking this, it doesn't go to page 2 anymore

Component image: 

; Screen Image: (Also see thumbnail in corresponding step below)

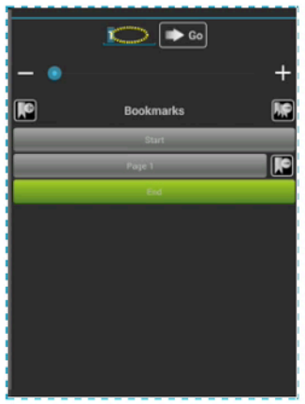
If there are any steps with (*), it means System A was not able to find information for those steps

Step 1

Step 2

Step 3

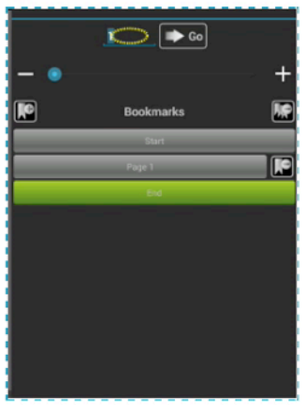
Step 4



Step 5

Step 6

Step 7



Back to Fusion | Print report | Go to viewer

By SEMERU, 2014

FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

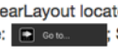
Bug **1418328948387** for **document_viewer.apk version 2.2**

The "Go To Page" Function does not work properly (by Kevin Moran) Nexus 7 portrait

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

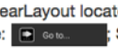
1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

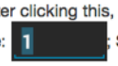
2. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

3. CLICK in/on LinearLayout located at Center. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

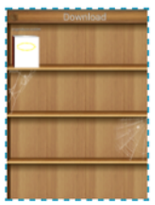
4. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

5. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity". Additional info: Prior to clicking again, swipe back up to page one
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

6. CLICK in/on LinearLayout located at Center. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

7. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity". Additional info: After clicking this, it doesn't go to page 2 anymore
Component image:  ; Screen Image: (Also see thumbnail in corresponding step below)

If there are any steps with (*), it means System A was not able to find information for those steps

Step 1


Step 2


Step 3


Step 4


Step 5


Step 6


Step 7


Back to Fusion | Print report | Go to viewer

EMPIRICAL EVALUATION

- Empirical study involving two software maintenance tasks and 28 users:

- Creating a bug report for a real app issue.

- Reproducing the bug on a device from a report.

- We used 15 real-world Android application bugs and compare FUSION to the Google Code Issue Tracker (GCIT) as well as the Original Bug Reports.

CONTEXT: BUG REPORTS USED IN THE STUDY

Summary of the bug reports used for the empirical studies: GDE = GUI Display Error, C = Crash, DIC = Data Input/Calculation Error, NE = Navigation Error

App (Bug Index)	Bug ID	Min # of Steps	Bug Type	DFS Activity Coverage
1) A Time Tracker	24	3	GDE	1/5
2) Aarddict	106	4-5	GDE	3/6
3) ACV	11	5	C	3/11
4) Car report	43	10	DIC	5/6
5) Document	48	4	NE	4/8
6) DroidWeight	38	7	GDE	3/8
7) Eshotroid	2	10	GDE/NE	6/6
8) GnuCash	256	10	DIC	3/4
9) GnuCash	247	10	DIC	3/4
10) Mileage	31	5	GDE/DIC	2/27
11) NetMBuddy	3	4	GDE/NE	5/13
12) Notepad	23	6	C	4/7
13) OI Notepad	187	10	GDE/DIC	3/9
14) Olam	2	3	C	1/1
15) QuickDic	85	5	GDE	3/6

TASK 1: BUG REPORT CREATION

- Goal: To assess whether FUSION's features are useful when reporting bugs for Android apps.

- Eight students from W&M, 4 CS graduate students, 4 undergraduate students.

- Users were exposed to the bugs through titled videos.

- All participants reproduced bugs on Google Nexus 7 Tablets with Android v4.4.3 KitKat installed.

TASK 2: BUG REPRODUCTION

- Goal: Evaluate the ability of FUSION to improve the reproducibility of bug reports

- 20 participants, all CS graduate students

- 135 bug reports were evaluated (120 from Study 1, plus the 15 original bug reports), each by two participants

- All participants reproduced bugs on Google Nexus 7 Tablets with Android v4.4.3 KitKat installed

RESEARCH QUESTIONS

- ***RQ1:*** Ease of Use?
- ***RQ2:*** Information Preferences?
- ***RQ3:*** Reproducibility of Reports?
- ***RQ4:*** Speed of Reproduction?

RESEARCH QUESTIONS

- *RQ1:* FUSION is about as *easy for developers to use* as traditional bug-tracking systems

- *RQ2:* Extra Information *increased quality* of reports

- *RQ3:* FUSION reports are *more reproducible* than traditional bug reports

- *RQ4:* Developers take *slightly longer* to reproduce FUSION Reports than traditional reports

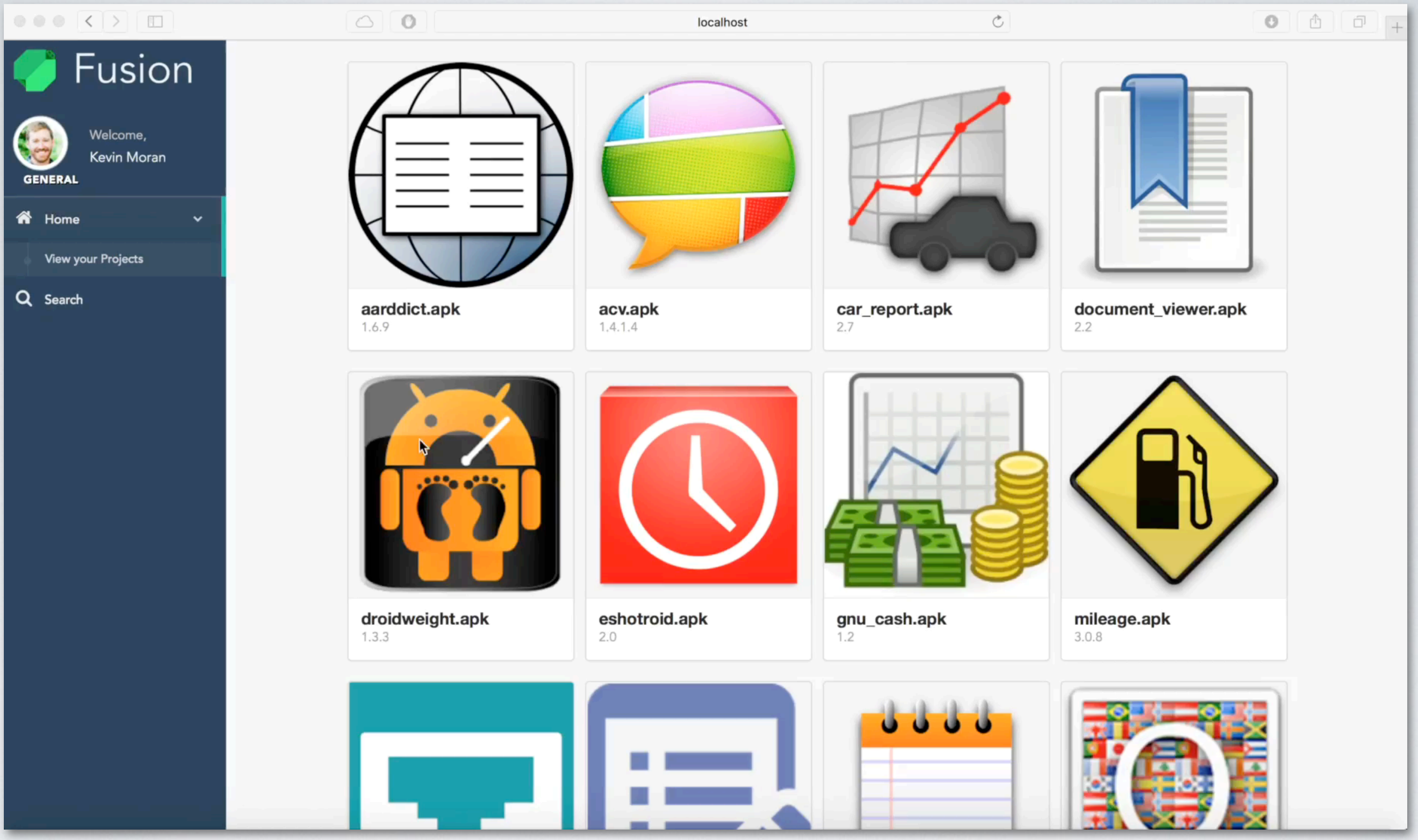
FUSION: Improving Mobile Bug Reporting

Team Members: Kevin Moran, Mario Linares-Vásquez, Carlos Bernal-Cárdenas, & Denys Poshyvanyk

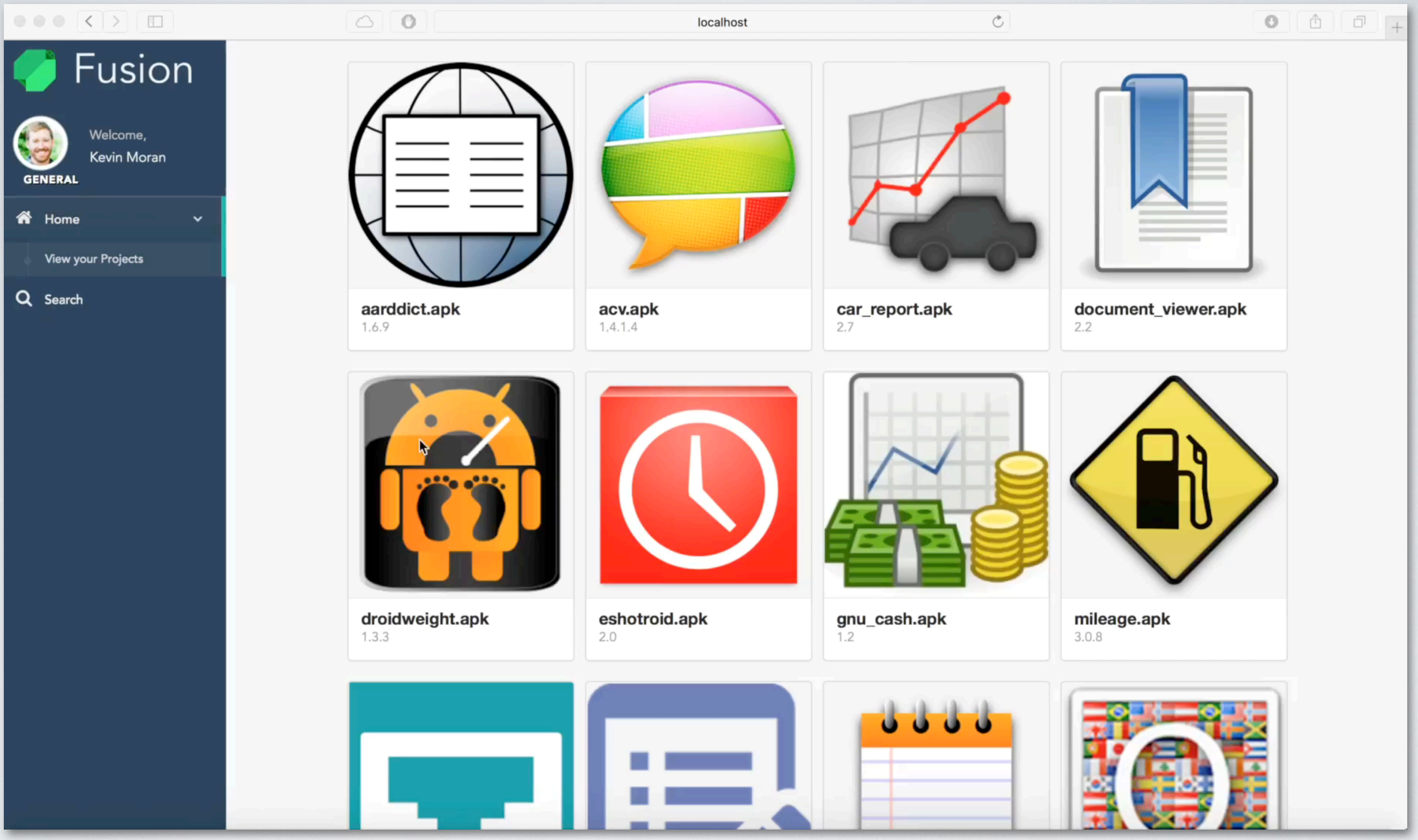
College of William & Mary --- SEMERU

Purpose

This project was created by the Software Engineering Maintenance and Evolution Research Unit (SEMERU) at the College of William & Mary, under the supervision of Dr. Denys Poshyvanyk. The major goal of the FUSION project is provide a more effective means of off-device bug reporting for Android applications that facilitates reporting through auto-completion, and provides detailed information to developers to aid in bug reproduction. In the future we hope to build out the tool to provide fault location capabilities and bug fixing suggestions.



Credits: Jacob Lisi, Ulises Giacoman, Sarah Melvin, Jiangnan Fu



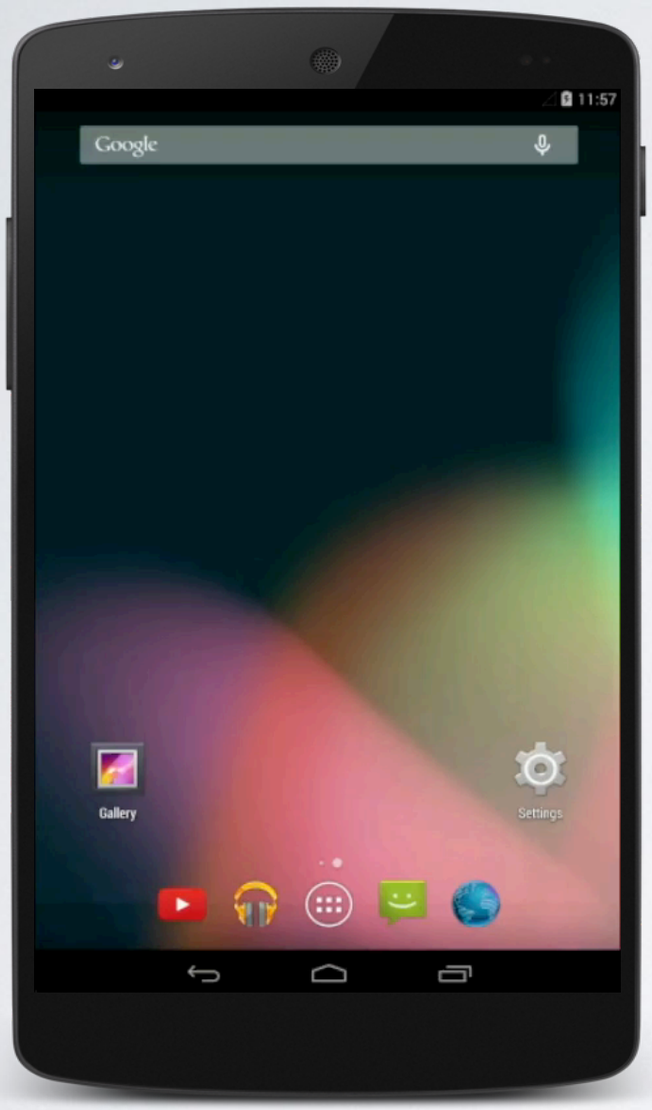
Credits: Jacob Lisi, Ulises Giacoman, Sarah Melvin, Jiangnan Fu

Thank you!

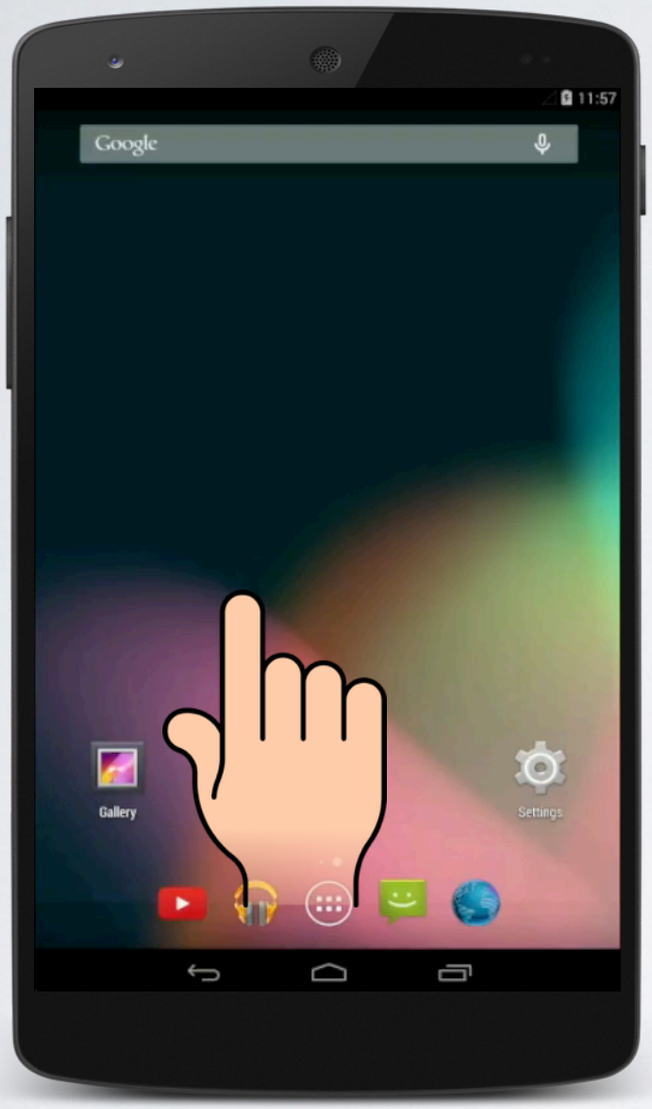
Questions?

www.fusion-android.com

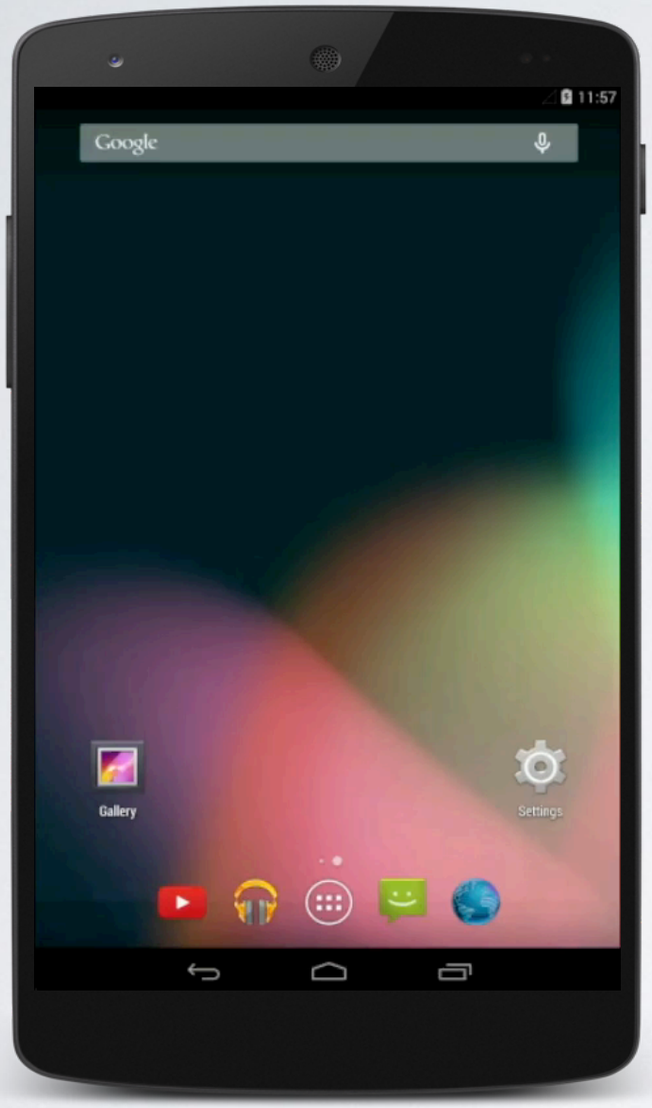
ADDITIONAL SLIDES


11:57
Google
Gallery
Settings

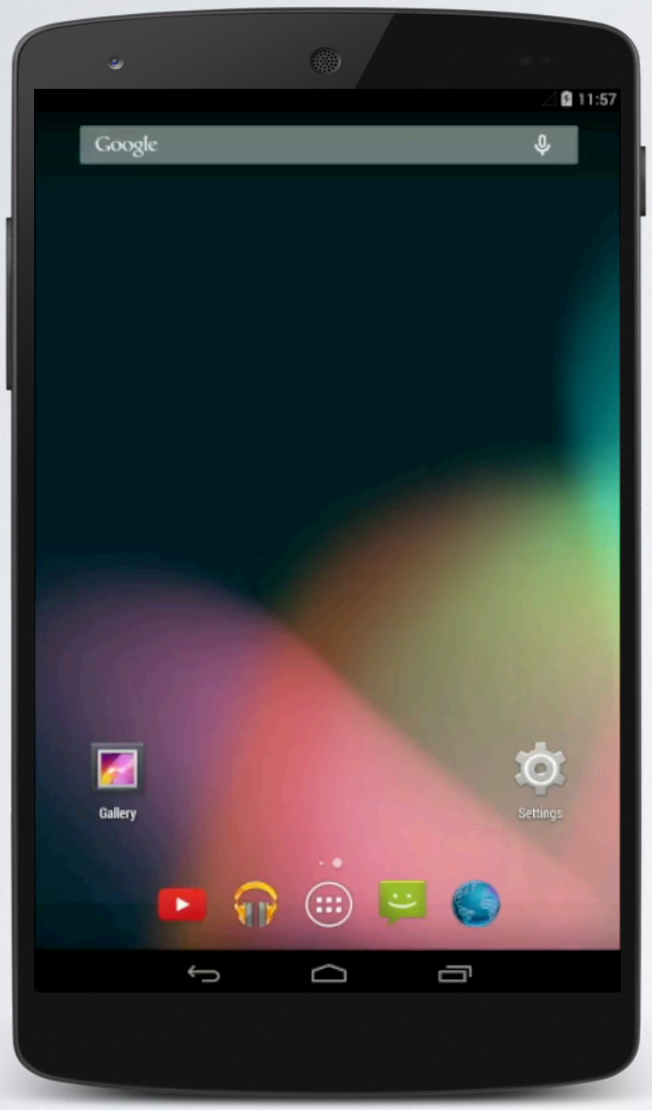
MANUAL TESTING

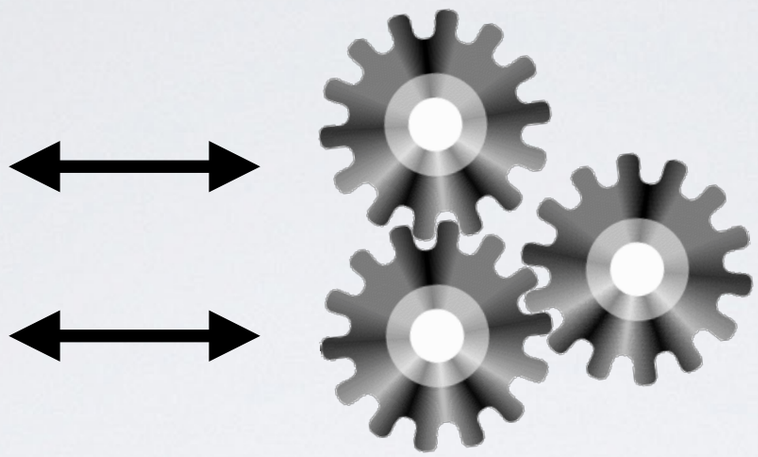


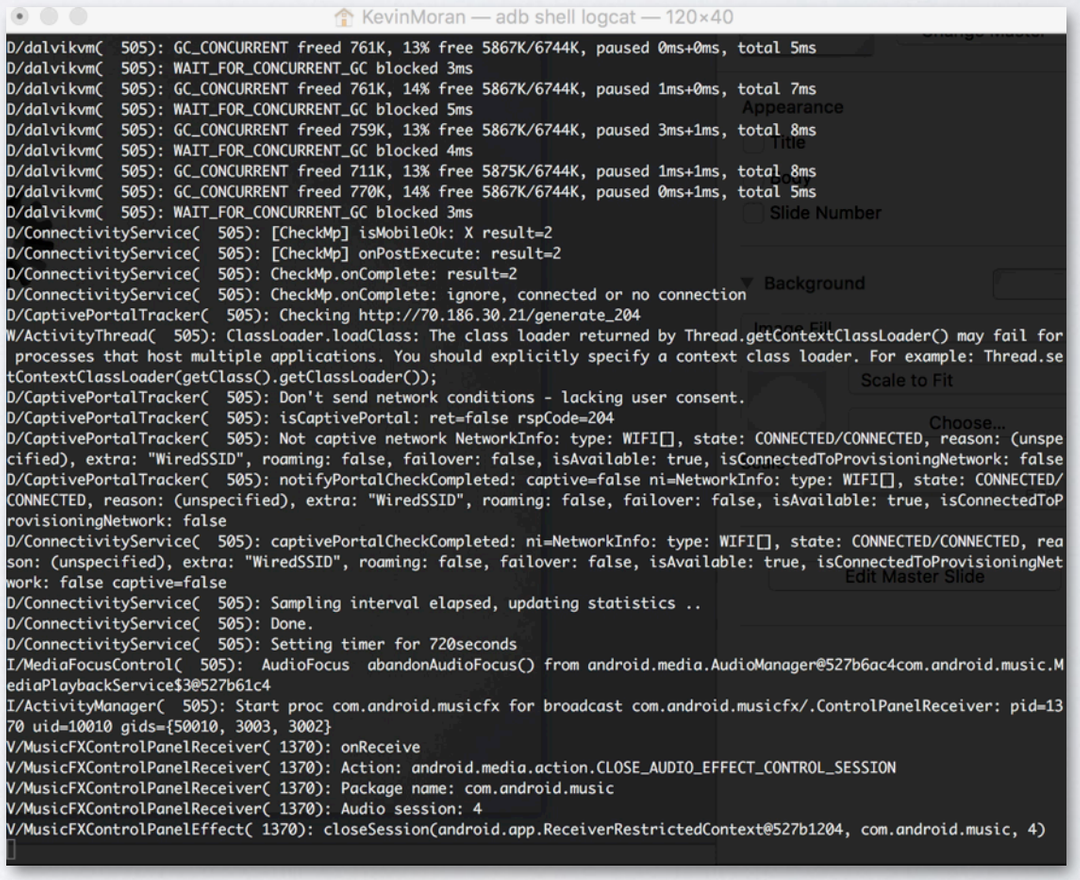
MANUAL TESTING



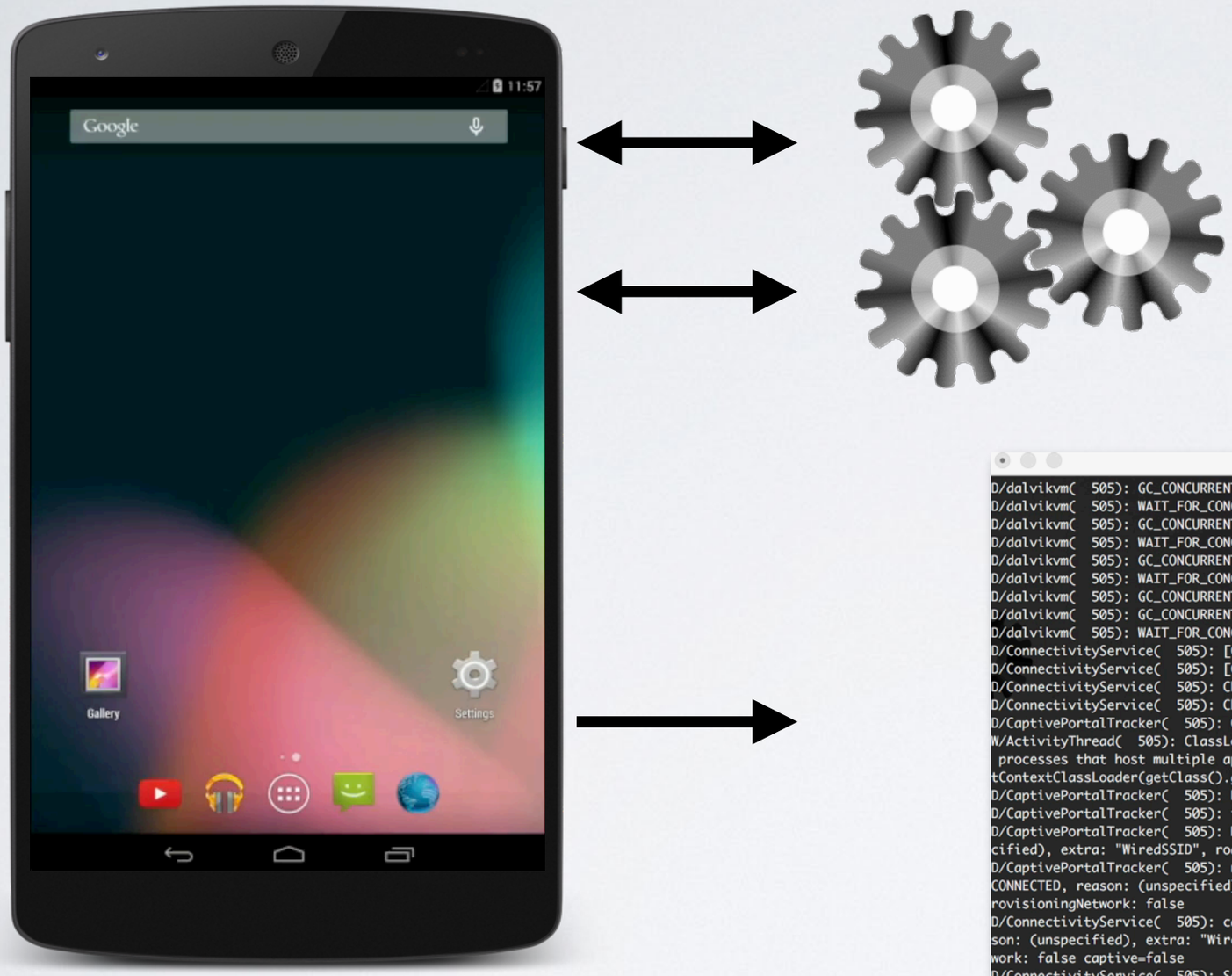
AUTOMATED TESTING

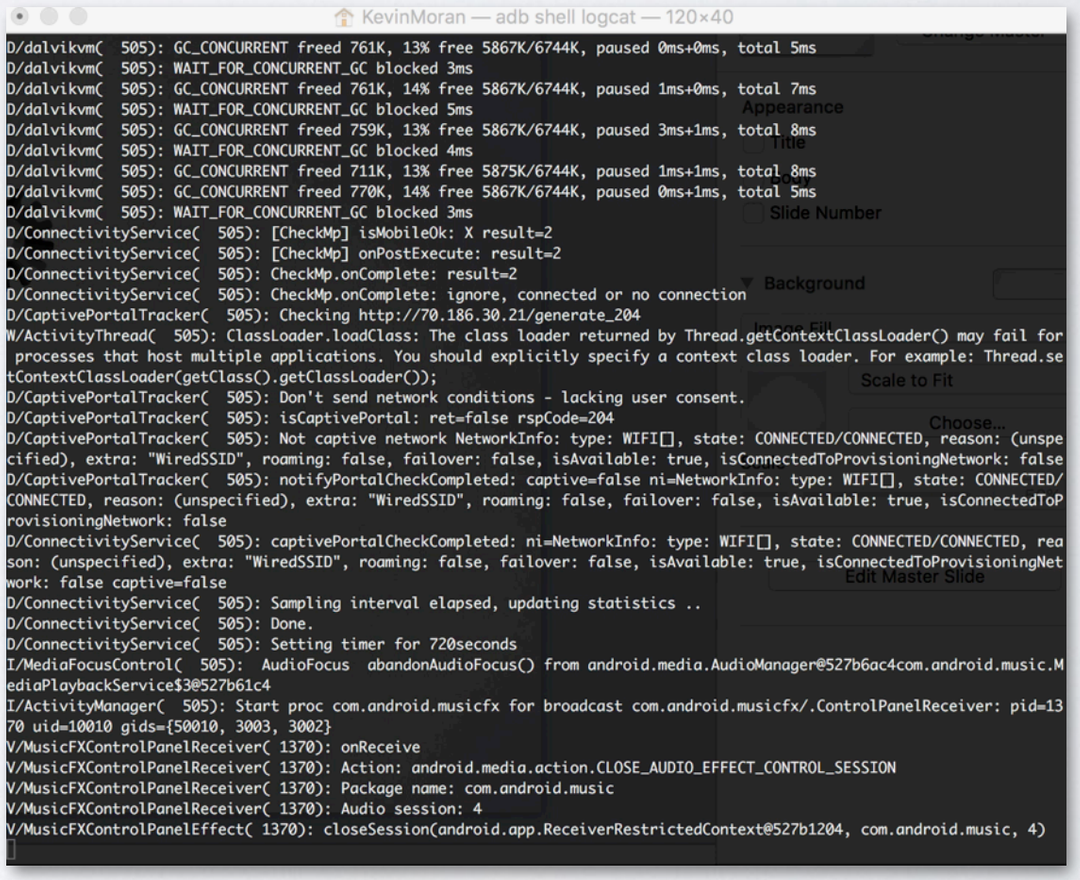




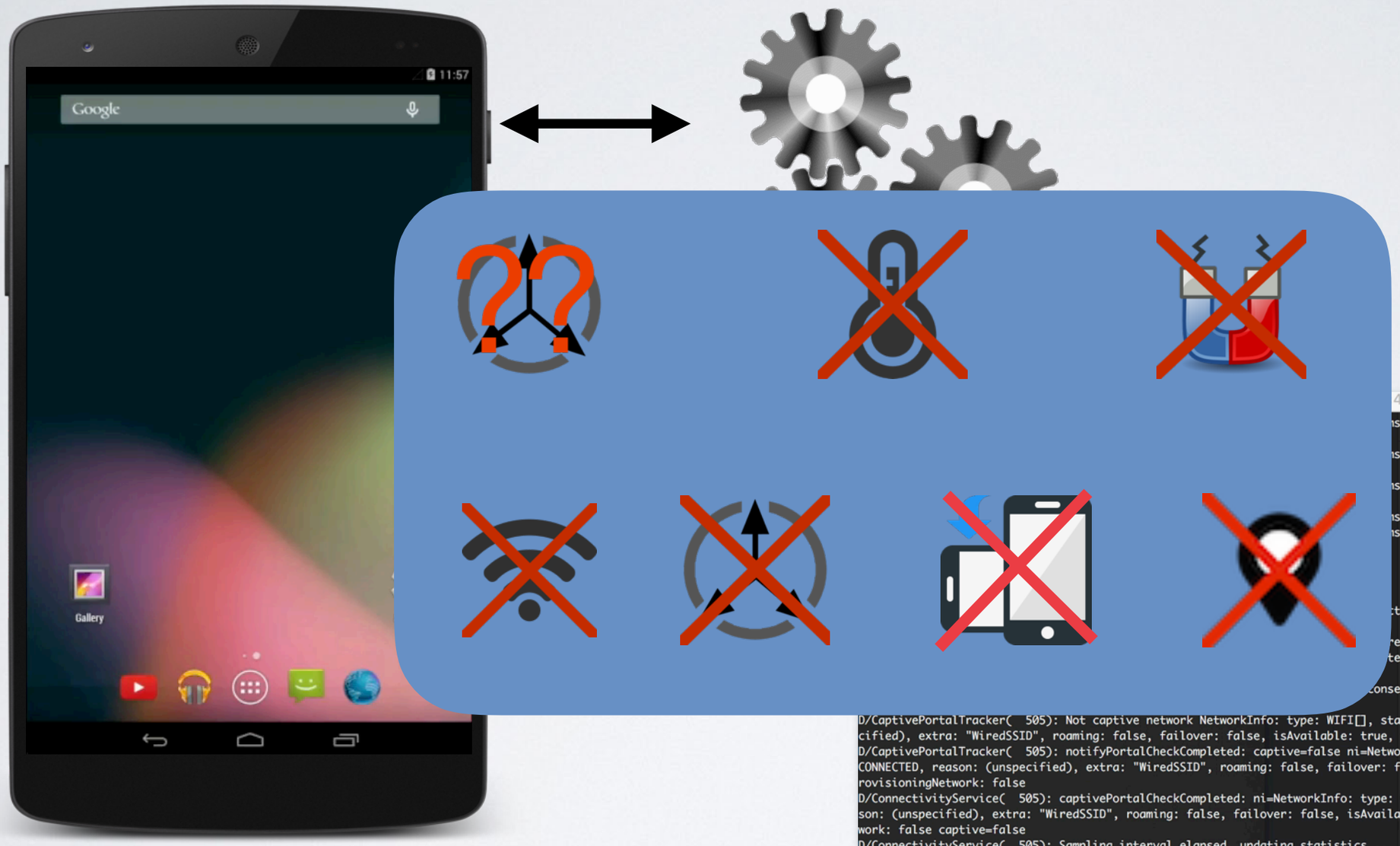


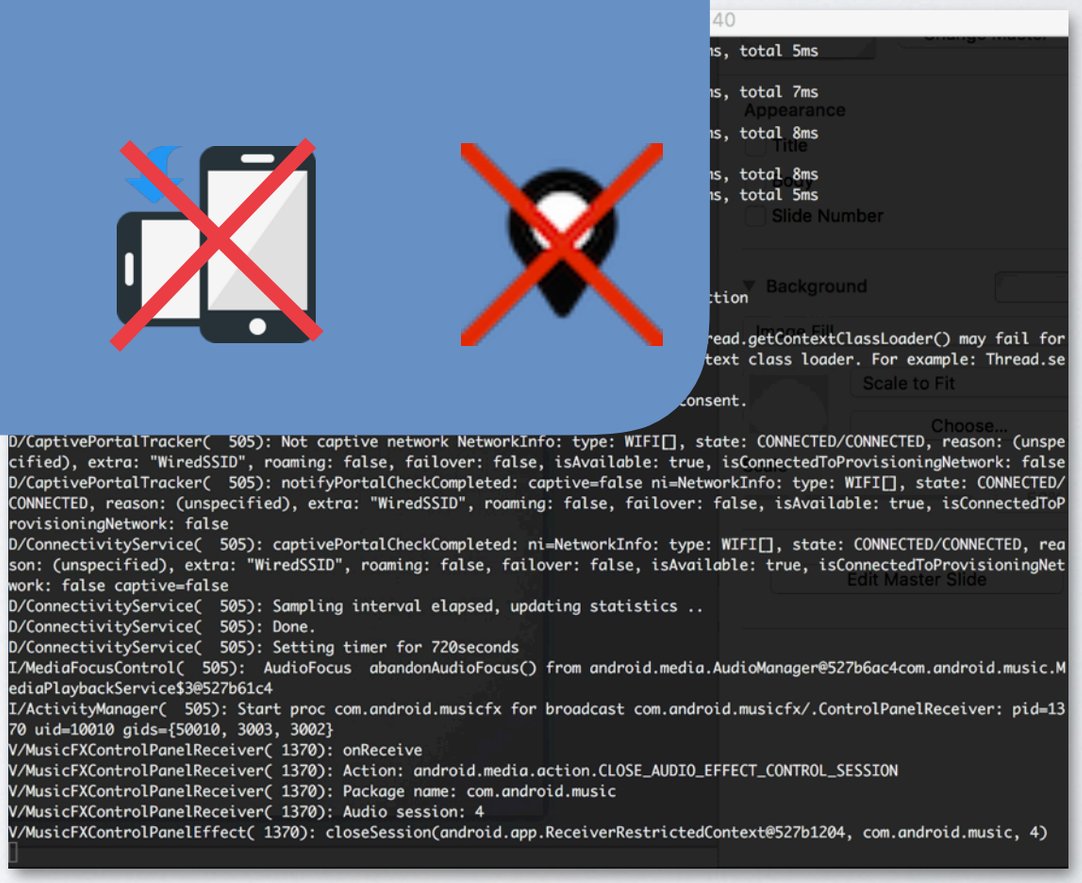
AUTOMATED TESTING





AUTOMATED TESTING





THE CURRENT STATE OF AUTOMATED MOBILE APPLICATION TESTING

Tool Name	Instr.	GUI Exploration	Types of Events	Crash Resilient	Replayable Test Cases	NL Crash Reports	Emulators, Devices
Dynodroid	Yes	Guided/Random	System, GUI, Text	Yes	No	No	No
EvoDroid	No	System/Evo	GUI	No	No	No	N/A
AndroidRipper	Yes	Systematic	GUI, Text	No	No	No	N/A
MobiGUItar	Yes	Model-Based	GUI, Text	No	Yes	No	N/A
A3E DFS	Yes	Systematic	GUI	No	No	No	Yes
A3E Targeted	Yes	Model-Based	GUI	No	No	No	Yes
Swifthand	Yes	Model-Based	GUI, Text	N/A	No	No	Yes
PUMA	Yes	Programmable	System, GUI, Text	N/A	No	No	Yes
ACTEve	Yes	Systematic	GUI	N/A	No	No	Yes
VANARSena	Yes	Random	System, GUI, Text	Yes	Yes	No	N/A
Thor	Yes	Test Cases	Test Case Events	N/A	N/A	No	No
QUANTUM	Yes	Model-Based	System, GUI	N/A	Yes	No	N/A
AppDoctor	Yes	Multiple	System, GUI, Text	Yes	Yes	No	N/A
ORBIT	No	Model-Based	GUI	N/A	No	No	N/A
SPAG-C	No	Record/Replay	GUI	N/A	N/A	No	No
JPF-Android	No	Scripting	GUI	N/A	Yes	No	N/A
MonkeyLab	No	Model-based	GUI, Text	No	Yes	No	Yes
CrashDroid	No	Manual Rec/Replay	GUI, Text	Manual	Yes	Yes	Yes
SIG-Droid	No	Symbolic	GUI, Text	N/A	Yes	No	N/A
CrashScope	**No**	**Systematic**	**GUI, Text, System**	**Yes**	**Yes**	**Yes**	**Yes**

THE CURRENT STATE OF AUTOMATED MOBILE APPLICATION TESTING

Tool Name	Instr.	GUI Exploration	Types of Events	Crash Resilient	Replayable Test Cases	NL Crash Reports	Emulators, Devices
Dy							
Ar							
I							
A							
V							
J							
M							
Cr							
SIG-Dr							N/A
CrashScope	**No**	**Systematic**	**GUI, Text, System**	**Yes**	**Yes**	**Yes**	**Yes**

What are the limitations of current automated approaches?

LIMITATIONS OF AUTOMATED MOBILE TESTING AND DEBUGGING

- Lack of detailed, easy to understand testing results for faults/ crashes[1]
- No easy way to reproduce test scenarios[1]
- Not practical from a developers viewpoint
- Few approaches enable different strategies capable of generating text and testing contextual features

[1]S. R. Choudhary, A. Gorla, and A. Orso. Automated Test Input Generation for Android: Are we there yet? In 30th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015), 2015

PAST STUDIES OF MOBILE CRASHES AND BUGS

- Many crashes can be mapped to ***well-defined***, ***externally inducible*** faults[1]

- ***Contextual features***, such as network connectivity and screen rotation, account for many of these externally inducible faults[1,2]

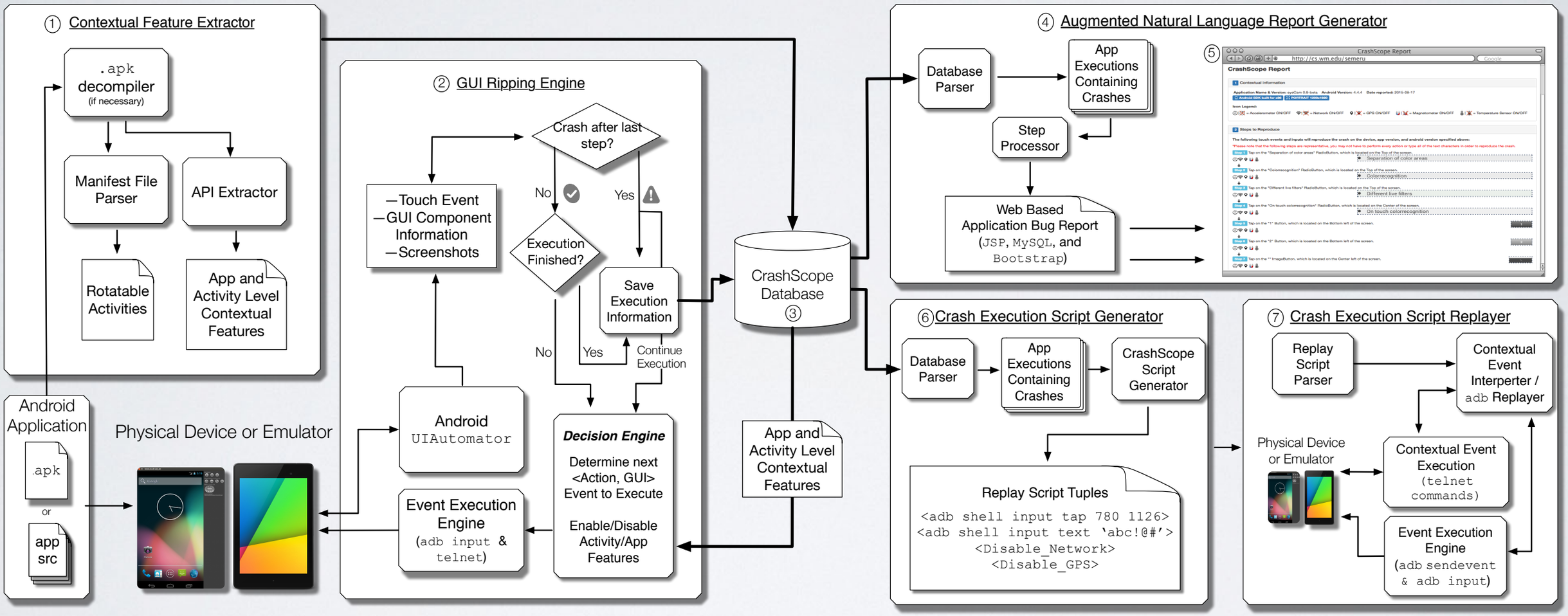
- These dominant root causes can affect ***many different*** user execution paths[1]

---

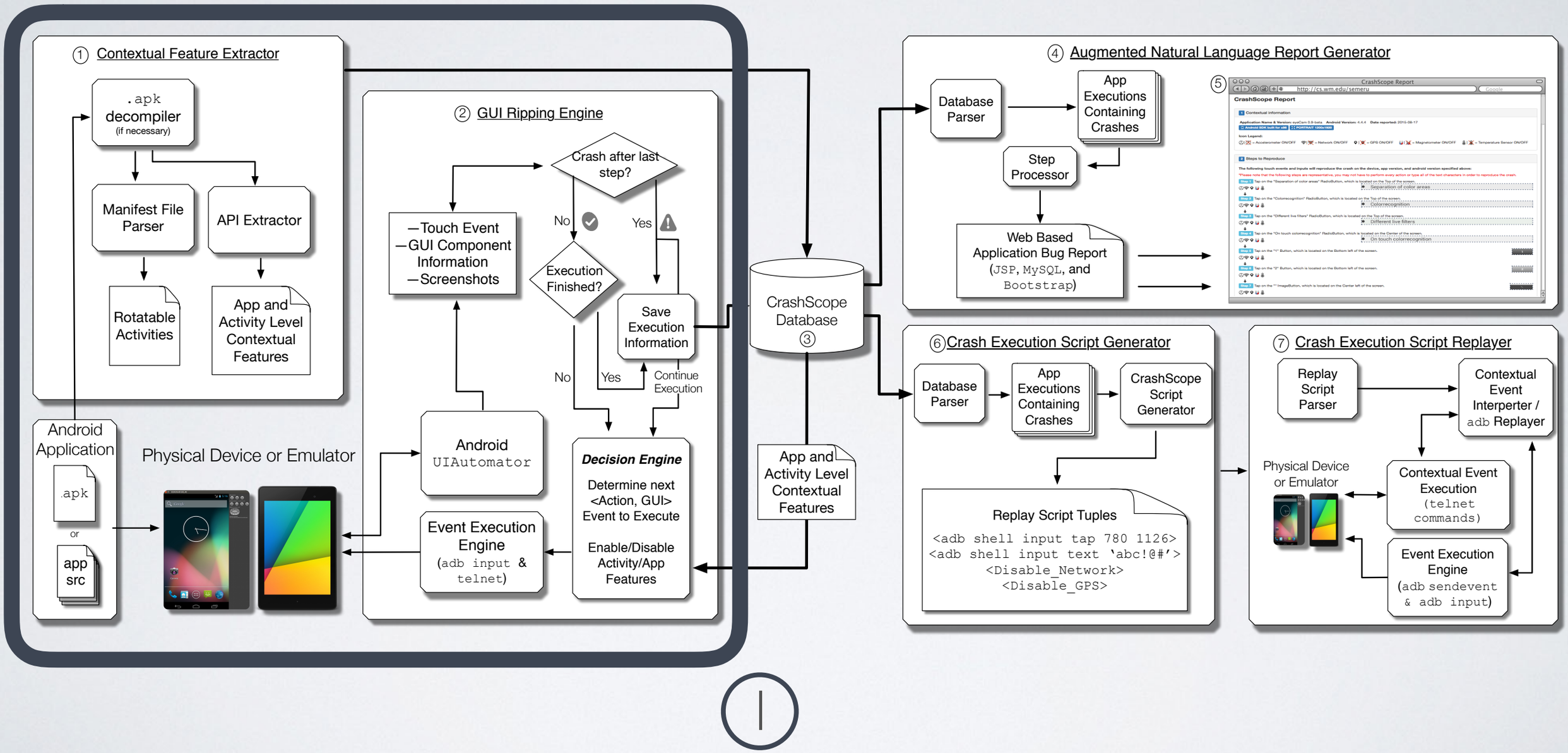
[1] L. Ravindranath, S. Nath, J. Padhye, and H. Balakrishnan. Automatic and scalable fault detection for mobile applications. MobiSys '14

[2] R. N. Zaeem, M. R. Prasad, and S. Khurshid. Automated generation of oracles for testing user-interaction features of mobile apps, ICST '14

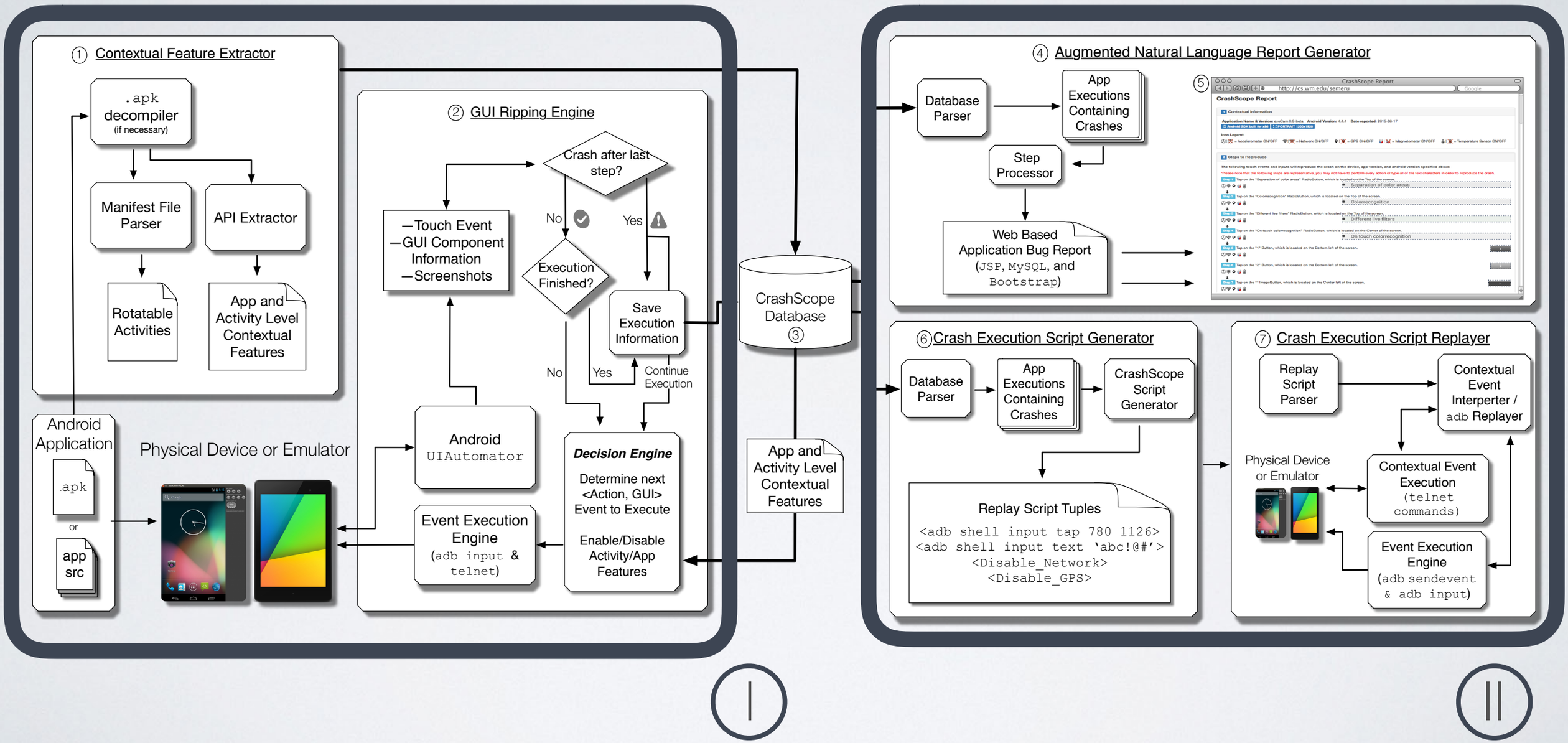
CRASHSCOPE DESIGN



CRASHSCOPE DESIGN



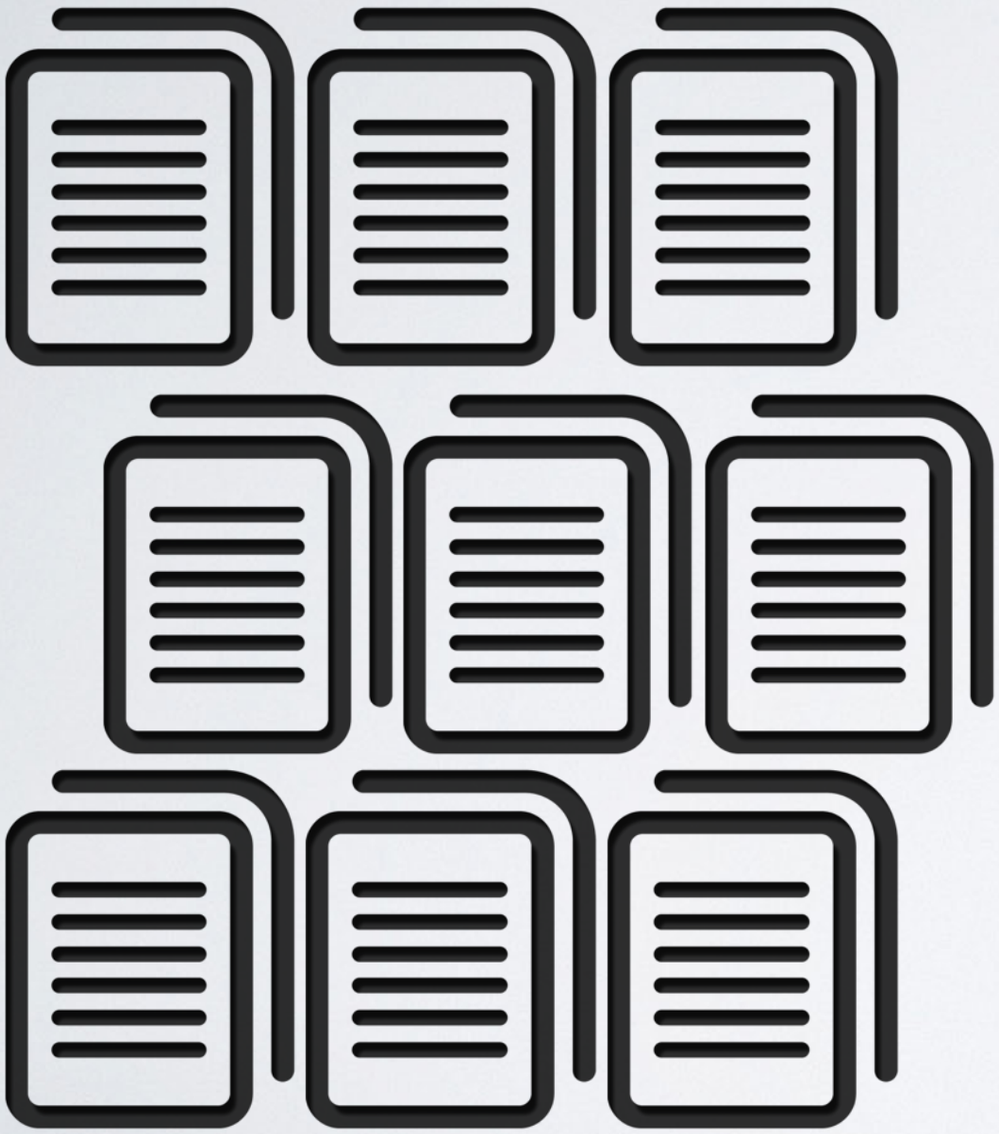
CRASHSCOPE DESIGN



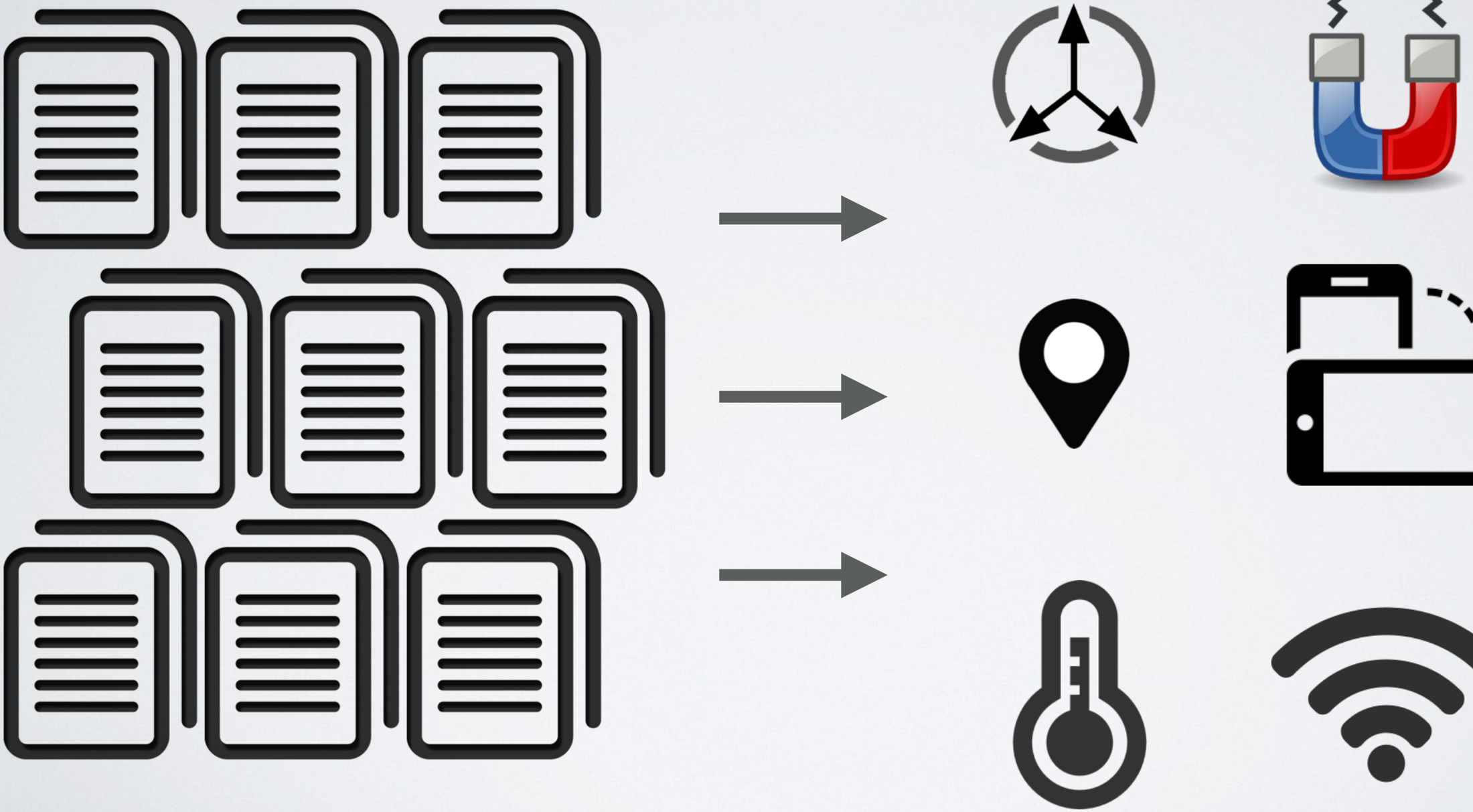
I

II

CRASHSCOPE: EXPLORATION



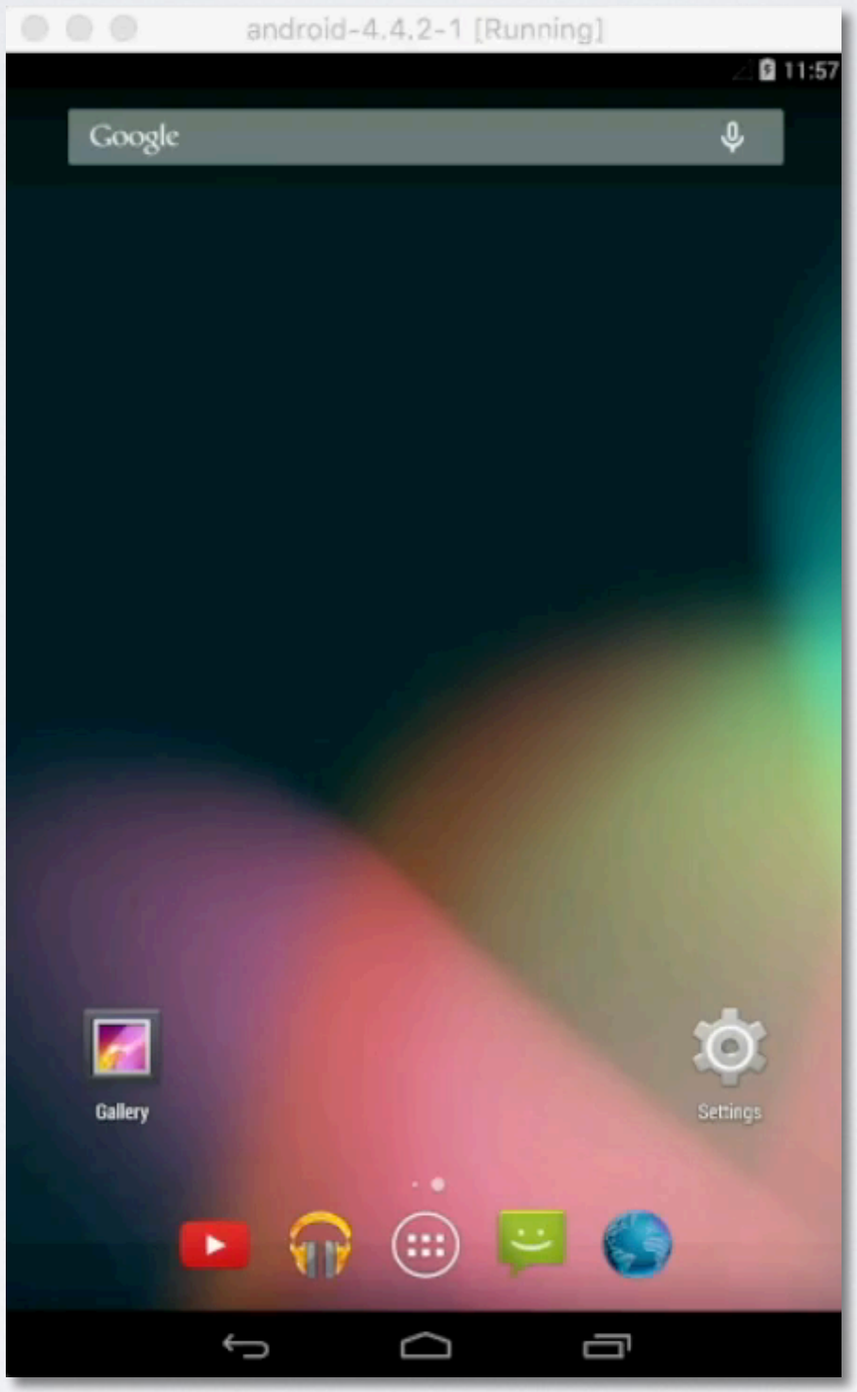
CRASHSCOPE: EXPLORATION



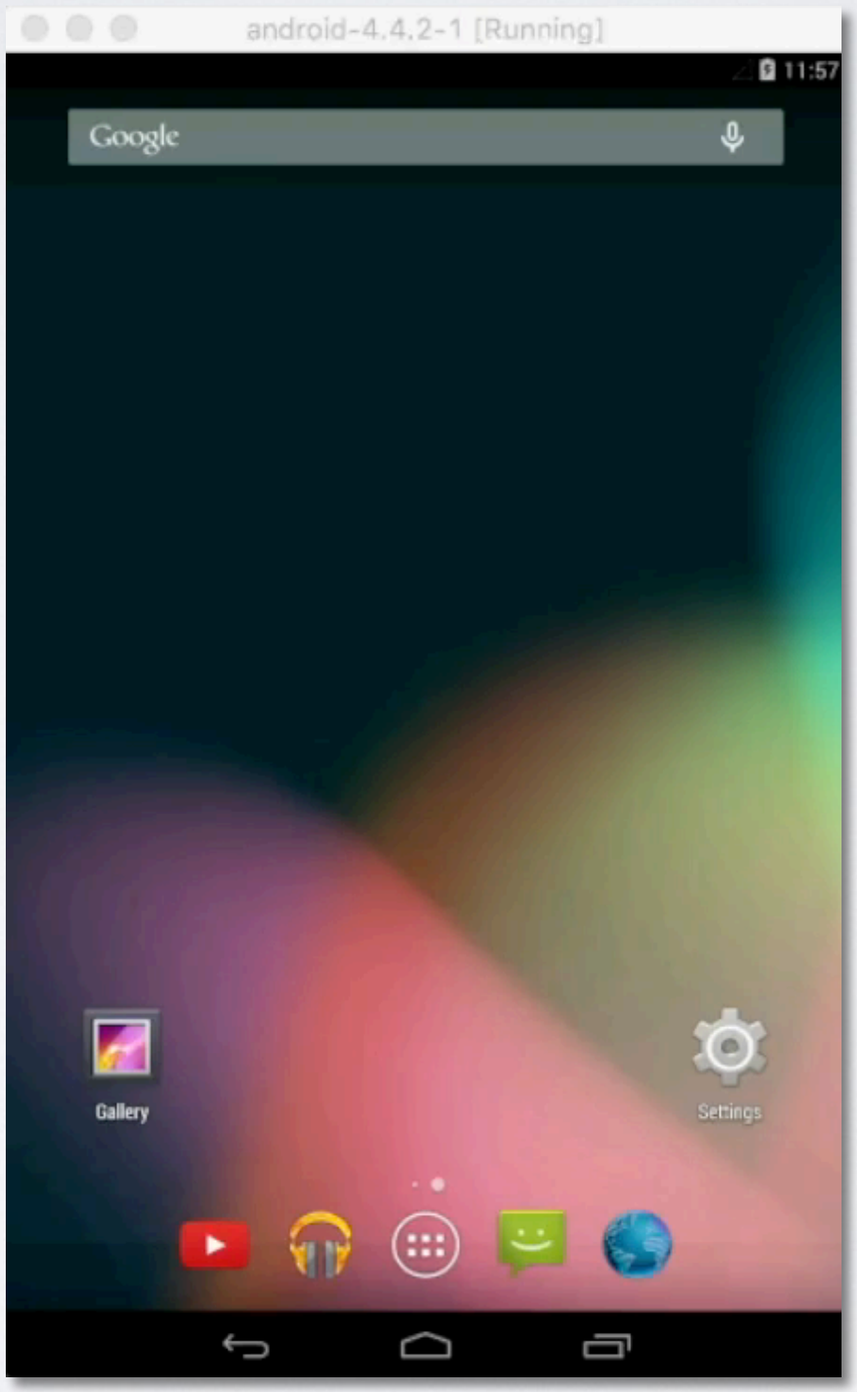
CRASHSCOPE STRATEGIES

- ***GUI-Traversal:*** Top-Down & Bottom Up
- ***Text Entry:*** Expected, Unexpected, No Text
- ***Contextual Features:*** Enabled or Disabled

CRASHSCOPE DEMO



CRASHSCOPE DEMO



CRASHSCOPE: REPORTS

CrashScope Report

1 Contextual information

Application Name & Version: GnuCash 1.0.3 **Android Version:** 4.4.4 **Date reported:** 2015-08-17

Android SDK built for x86 | PORTRAIT 1200x1920

Icon Legend:

= Accelerometer ON/OFF = Network ON/OFF = GPS ON/OFF = Magnetometer ON/OFF = Temperature Sensor ON/OFF

2 Steps to Reproduce

The following touch events and inputs will reproduce the crash on the device, app version, and android version specified above:

*Please note that the following steps are representative, you may not have to perform every action or type all of the text characters in order to reproduce the crash.

Step 1 Tap on the "Expenses" CheckedTextView, which is located on the Center of the screen.

Expenses

Step 2 Tap on the "Income" CheckedTextView, which is located on the Center of the screen.

Income

Step 3 Tap on the "Assets" CheckedTextView, which is located on the Center of the screen.

Assets

Step 4 Tap on the "Entertainment" CheckedTextView, which is located on the Center of the screen.

Entertainment

Step 5 Tap on the "Insurance" CheckedTextView, which is located on the Center of the screen.

Insurance

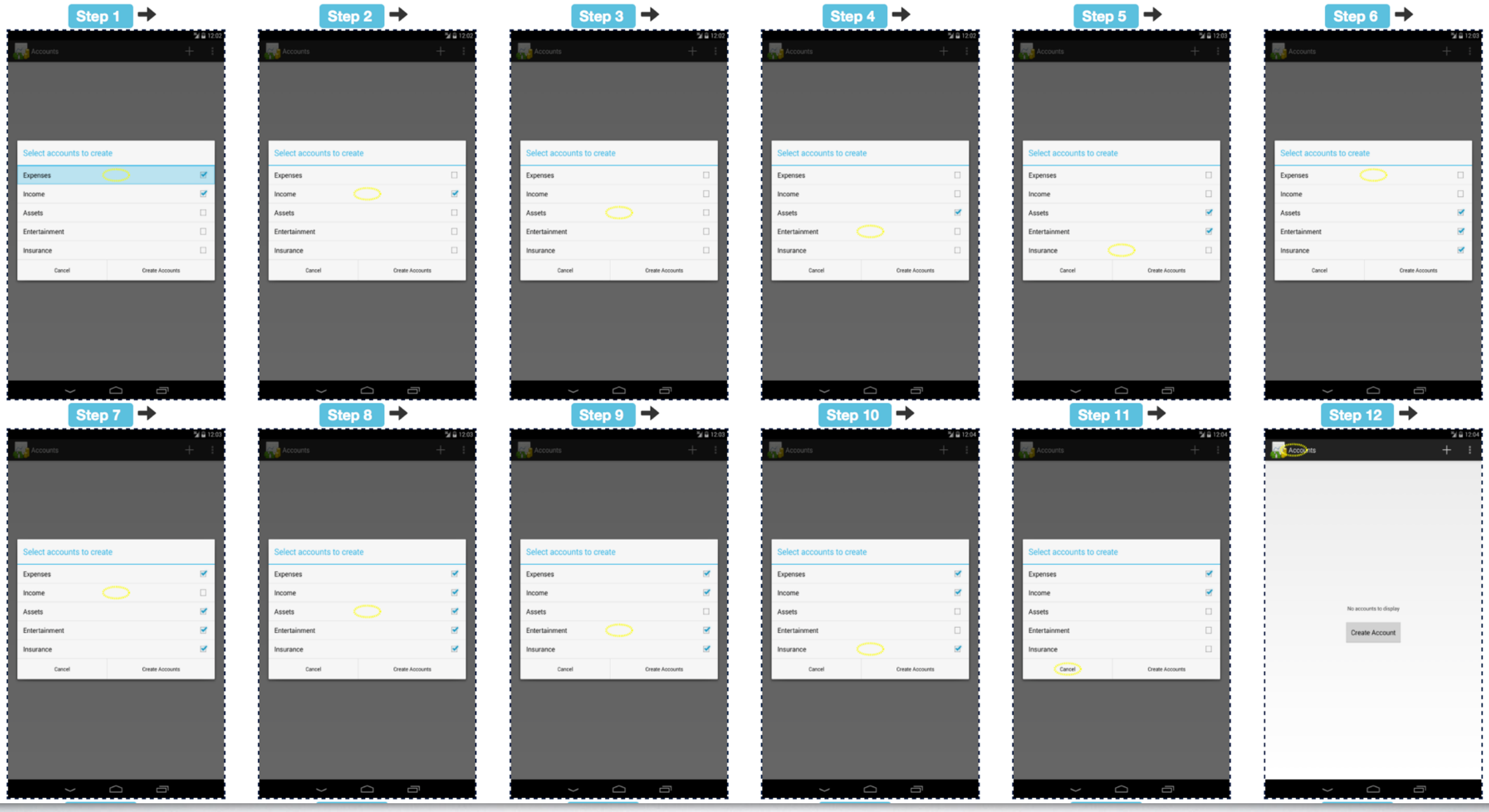
Step 6 Tap on the "Expenses" CheckedTextView, which is located on the Center of the screen.

Expenses

CRASHSCOPE: REPORTS

3 Crash Application Screen-Flow

(Go top)



CRASHSCOPE: REPORTS

4 Crash Pruned Stack Trace (Go top)

```
E/SQLiteLog(17653): (1) near "inhphsjgf": syntax error
E/AndroidRuntime(17653): FATAL EXCEPTION: main
E/AndroidRuntime(17653): Process: org.gnucash.android, PID: 17653
E/AndroidRuntime(17653): android.database.sqlite.SQLiteException: near "inhphsjgf": syntax error (code 1): , while compiling: SELECT _i
d, uid FROM accounts WHERE uid = ''inhphsjgf-d724114f522e'
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteConnection.nativePrepareStatement(Native Method)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteConnection.acquirePreparedStatement(SQLiteConnection.java:889)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteConnection.prepare(SQLiteConnection.java:500)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteSession.prepare(SQLiteSession.java:588)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteProgram.(SQLiteProgram.java:58)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteQuery.(SQLiteQuery.java:37)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteDirectCursorDriver.query(SQLiteDirectCursorDriver.java:44)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteDatabase.rawQueryWithFactory(SQLiteDatabase.java:1314)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteDatabase.queryWithFactory(SQLiteDatabase.java:1161)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteDatabase.query(SQLiteDatabase.java:1032)
E/AndroidRuntime(17653):        at android.database.sqlite.SQLiteDatabase.query(SQLiteDatabase.java:1200)
E/AndroidRuntime(17653):        at org.gnucash.android.db.AccountsDbAdapter.getAccountID(AccountsDbAdapter.java:166)
E/AndroidRuntime(17653):        at org.gnucash.android.db.AccountsDbAdapter.addAccount(AccountsDbAdapter.java:76)
E/AndroidRuntime(17653):        at org.gnucash.android.ui.accounts.NewAccountDialogFragment$1.onClick(NewAccountDialogFragment.java:156
)
E/AndroidRuntime(17653):        at android.view.View.performClick(View.java:4438)
E/AndroidRuntime(17653):        at android.view.View$PerformClick.run(View.java:18422)
E/AndroidRuntime(17653):        at android.os.Handler.handleCallback(Handler.java:733)
E/AndroidRuntime(17653):        at android.os.Handler.dispatchMessage(Handler.java:95)
E/AndroidRuntime(17653):        at android.os.Looper.loop(Looper.java:136)
E/AndroidRuntime(17653):        at android.app.ActivityThread.main(ActivityThread.java:5001)
E/AndroidRuntime(17653):        at java.lang.reflect.Method.invokeNative(Native Method)
E/AndroidRuntime(17653):        at java.lang.reflect.Method.invoke(Method.java:515)
E/AndroidRuntime(17653):        at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:785)
E/AndroidRuntime(17653):        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:601)
E/AndroidRuntime(17653):        at dalvik.system.NativeStart.main(Native Method)
```

EVALUATION

- Two Empirical Studies
- Study 1: Crash Detection Capabilities
- Study 2: Crash Report Reproducibility and Readability

STUDY 1: CRASH RESULTS

Unique Crashes Discovered With Instrumented Crashes in Parentheses

App	A3E	GUI- Ripper	Dynodroid	PUMA	Monkey (All)	CrashScope
A2DP Vol	1	0	0	0	0	0
aagtl	0	0	1	0	1	0
Amazed	0	0	0	0	1	0
HNDroid	1	1	1	2	1	1
BatteryDog	0	0	1	0	1	0
Soundboard	0	1	0	0	0	0
AKA	0	0	0	0	1	0
Bites	0	0	0	0	1	0
Yahtzee	1	0	0	0	0	1
ADSDroid	1	1	1	1	1	1
PassMaker	1	0	0	0	1	1
BlinkBattery	0	0	0	0	1	0
D&C	0	0	0	0	1	0
Photostream	1	1	1	1	1	0
AlarmKlock	0	0	1	0	0	0
Sanity	1	1	0	0	0	0
MyExpenses	0	0	1	0	0	0
Zooborns	0	0	0	0	0	2
ACal	1	2	2	0	1	1
Hotdeath	0	2	0	0	0	1
Total	**8 (21)**	**9 (5)**	**9 (6)**	**4 (0)**	**12 (1)**	**8 (0)**

STUDY 1: CRASH RESULTS

Unique Crashes Discovered With Instrumented Crashes in Parentheses

App	A3E	GUI- Ripper	Dynodroid	PUMA	Monkey (All)	CrashScope
A2DP Vol	1	0	0	0	0	0
aagtl	0	0	1	0	1	0
Amazed	0	0	0	0	1	0
HNDroid	1	1	1	2	1	1
BatteryDog	0	0	1	0	1	0
Soundboard	0	1	0	0	0	0
AKA	0	0	0	0	1	0
Bites	0	0	0	0	1	0
Yahtzee	1	0	0	0	0	1
ADSDroid	1	1	1	1	1	1
PassMaker	1	0	0	0	1	1
BlinkBattery	0	0	0	0	1	0
D&C	0	0	0	0	1	0
Photostream	1	1	1	1	1	0
AlarmKlock	0	0	1	0	0	0
Sanity	1	1	0	0	0	0
MyExpenses	0	0	1	0	0	0
Zooborns	0	0	0	0	0	2
ACal	1	2	2	0	1	1
Hotdeath	0	2	0	0	0	1
Total	8 (21)	9 (5)	9 (6)	4 (0)	12 (1)	8 (0)

STUDY 1: CRASH RESULTS

-CrashScope is about as effective as other techniques with regard to uncovering crashes.

-CrashScope is able to uncover orthogonal crashes

STUDY 2: READABILITY RESULTS

Question	CrashScope Mean	CrashScope StdDev	Original Mean	Original StdDev
UX1: I think I would like to have this type of bug report frequently.	4.00	0.89	3.06	0.77
UX2: I found this type of bug report unnecessarily complex.	2.81	1.04	2.125	0.96
UX3: I thought this type of bug report was easy to read/understand.	4.00	0.82	3.00	0.97
UX4: I found this type of bug report very cumbersome to read.	2.50	1.10	2.44	0.81
UX5: I thought the bug report was very useful for reproducing the crash.	4.13	0.62	3.44	0.89

STUDY 2: READABILITY RESULTS

Question	CrashScope Mean	CrashScope StdDev	Original Mean	Original StdDev
UX1: I think I would like to have this type of bug report frequently.	4.00	0.89	3.06	0.77
UX2: I found this type of bug report unnecessarily complex.	2.81	1.04	2.125	0.96
UX3: I thought this type of bug report was easy to read/understand.	4.00	0.82	3.00	0.97
UX4: I found this type of bug report very cumbersome to read.	2.50	1.10	2.44	0.81
UX5: I thought the bug report was very useful for reproducing the crash.	4.13	0.62	3.44	0.89

STUDY 2: READABILITY RESULTS

Question	CrashScope Mean	CrashScope StdDev	Original Mean	Original StdDev
UX1: I think I would like to have this type of bug report frequently.	4.00	0.89	3.06	0.77
UX2: I found this type of bug report unnecessarily complex.	2.81	1.04	2.125	0.96
UX3: I thought this type of bug report was easy to read/understand.	4.00	0.82	3.00	0.97
UX4: I found this type of bug report very cumbersome to read.	2.50	1.10	2.44	0.81
UX5: I thought the bug report was very useful for reproducing the crash.	4.13	0.62	3.44	0.89

STUDY 2: READABILITY RESULTS

- Reports generated by CrashScope are more readable and reproducible

CRASHSCOPE: A PRACTICAL TOOL

HUAWEI

CrashScope | Dashboard

localhost:8080/CrashScope/app/report_results.xhtml

Kevin

Requests/Tasks | New CrashScope Task | Kevin

Detected Crashes

Link to Bug Report	Crash ID	# Steps	Strategies
View Report	1	29	
View Report	2	12	
View Report	3	9	

Strategy Icon Legend
Top-Down Exploration:
Bottom-Up Exploration:
Unexpected Text Input:
Expected Text input:
No Text Input:
Contextual Features Enabled/Disabled: ✓/ ✗

Report Viewer

1 Contextual information

Application Name & Version: GnuCash 1.0.3 **Android Version:** 4.4.4 **Date reported:** Mon Aug 17 00:00:00 EDT 2015

Android SDK built for x86 | PORTRAIT 1200x1920

Icon Legend:

= Accelerometer ON/OFF = Network ON/OFF = GPS ON/OFF = Magnetometer ON/OFF = Temperature Sensor ON/OFF

2 Steps to Reproduce

The following touch events and inputs will reproduce the crash on the device, app version, and android version specified above:

*Please note that the following steps are representative, you may not have to perform every action or type all of the text characters in order to reproduce the crash.

Step 1 Tap on the "android.widget.CheckedTextView" Expenses, which is located on the Center of the screen.

Expenses

Step 2 Tap on the "android.widget.CheckedTextView" Income, which is located on the Center of the screen.

Income

Step 3 Tap on the "android.widget.CheckedTextView" Assets, which is located on the Center of the screen.

Assets

Step 4 Tap on the "android.widget.CheckedTextView" Entertainment, which is located on the Center of the screen.

CRASHSCOPE: A PRACTICAL TOOL

HUAWEI

CrashScope | Dashboard

localhost:8080/CrashScope/app/report_results.xhtml

Kevin

Requests/Tasks | New CrashScope Task | Kevin

Detected Crashes

Link to Bug Report	Crash ID	# Steps	Strategies
View Report	1	29	
View Report	2	12	
View Report	3	9	

Strategy Icon Legend
Top-Down Exploration:
Bottom-Up Exploration:
Unexpected Text Input:
Expected Text input:
No Text Input:
Contextual Features Enabled/Disabled: ✓/ ✗

Report Viewer

1 Contextual information

Application Name & Version: GnuCash 1.0.3 **Android Version:** 4.4.4 **Date reported:** Mon Aug 17 00:00:00 EDT 2015

Android SDK built for x86 | PORTRAIT 1200x1920

Icon Legend:

= Accelerometer ON/OFF = Network ON/OFF = GPS ON/OFF = Magnetometer ON/OFF = Temperature Sensor ON/OFF

2 Steps to Reproduce

The following touch events and inputs will reproduce the crash on the device, app version, and android version specified above:

*Please note that the following steps are representative, you may not have to perform every action or type all of the text characters in order to reproduce the crash.

Step 1 Tap on the "android.widget.CheckedTextView" Expenses, which is located on the Center of the screen.

Expenses

Step 2 Tap on the "android.widget.CheckedTextView" Income, which is located on the Center of the screen.

Income

Step 3 Tap on the "android.widget.CheckedTextView" Assets, which is located on the Center of the screen.

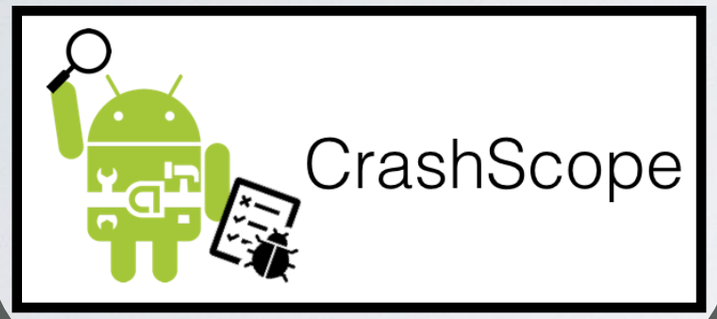
Assets

Step 4 Tap on the "android.widget.CheckedTextView" Entertainment, which is located on the Center of the screen.

LOOKING FORWARD:
POTENTIAL RESEARCH MAP

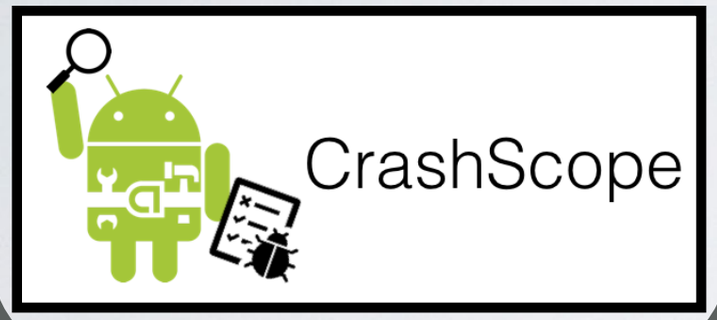
The Starting Point

&



The Starting Point

&



On-Device Bug Reporting

FUSION: ON-DEVICE BUG REPORTING

Advisees:
Richard Bonnet,
Brendan Otten,
Daniel Park

FUSION: ON-DEVICE BUG REPORTING

Advisees:
Richard Bonnet,
Brendan Otten,
Daniel Park

The Starting Point

FUSION

&

CrashScope

On-Device Bug Reporting

The Starting Point

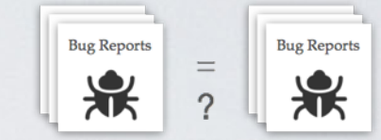
FUSION

&

CrashScope

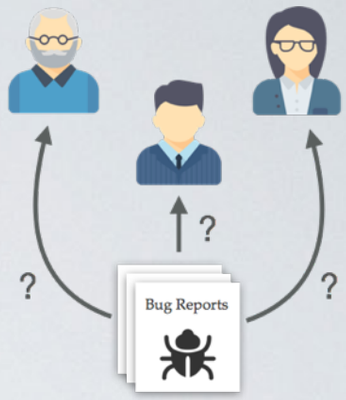
On-Device Bug Reporting

Improving Related Maintenance Tasks



Duplicate Bug Report Detection

Fault-Localization



Developer Triaging

The Starting Point

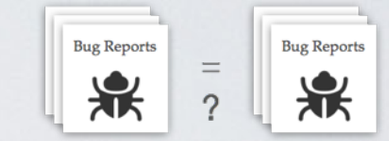
FUSION

&

CrashScope

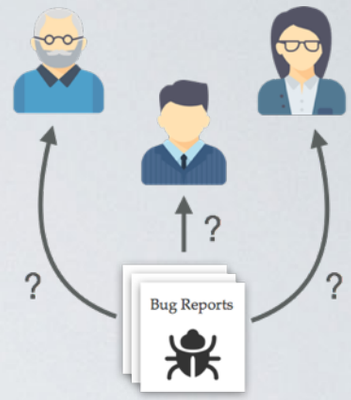
On-Device Bug Reporting

Improving Related Maintenance Tasks



Duplicate Bug Report Detection

Fault-Localization



Developer Triaging

Large-Scale Dynamic Analysis

The Starting Point

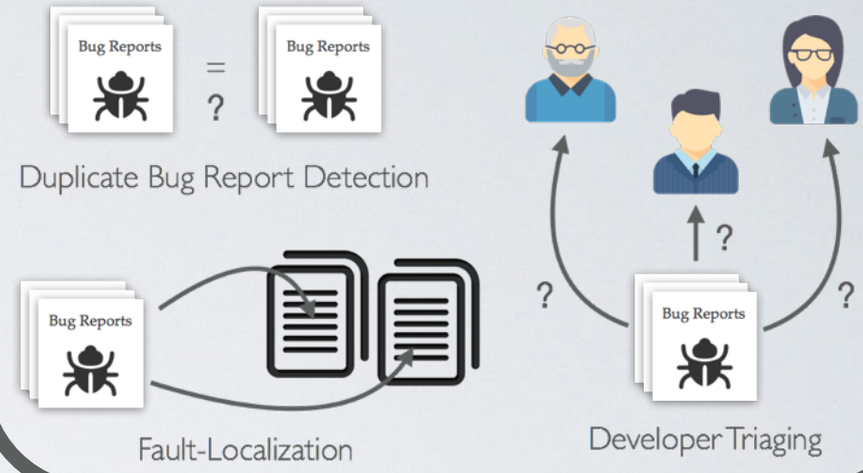
FUSION

&

CrashScope

On-Device Bug Reporting

Improving Related Maintenance Tasks



Large-Scale Dynamic Analysis

FUSION for Web-Apps

The Starting Point

FUSION

&

CrashScope

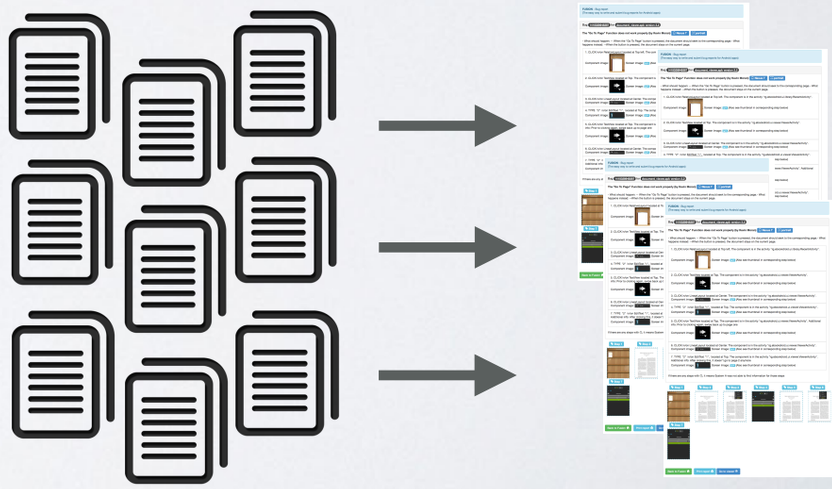
On-Device Bug Reporting

Improving Related Maintenance Tasks

Large-Scale Dynamic Analysis

FUSION for Web-Apps

Inferring Bug Reports from User Reviews



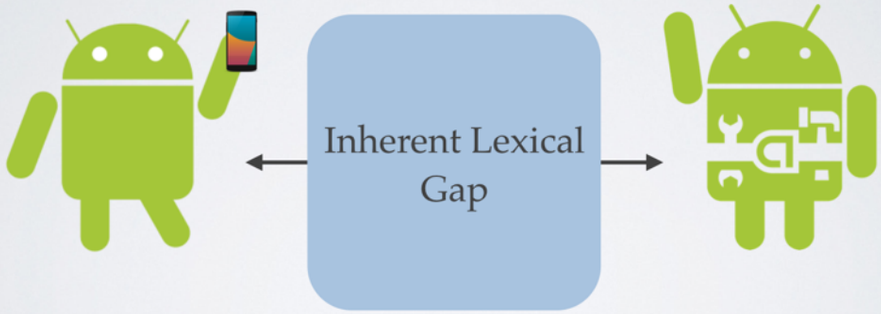
User Reviews

Bug Reports

THE LEXICAL GAP IN BUG REPORTING

Reporters:
-Functional Knowledge of a Software Bug.

Developers:
-Intimate Code Level Knowledge of Application

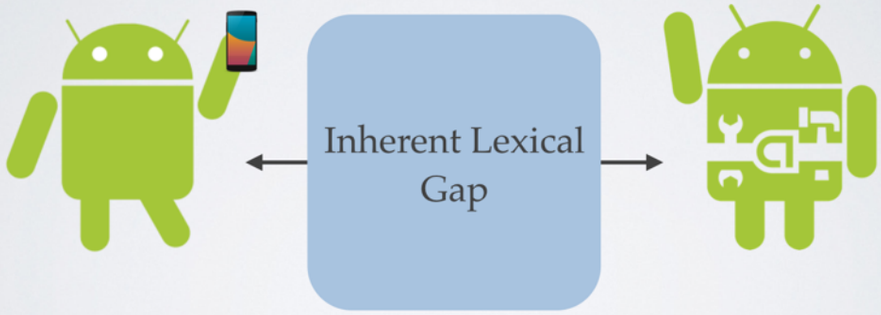


Images Courtesy of Google

THE LEXICAL GAP IN BUG REPORTING

Reporters:
-Functional Knowledge of a Software Bug.

Developers:
-Intimate Code Level Knowledge of Application

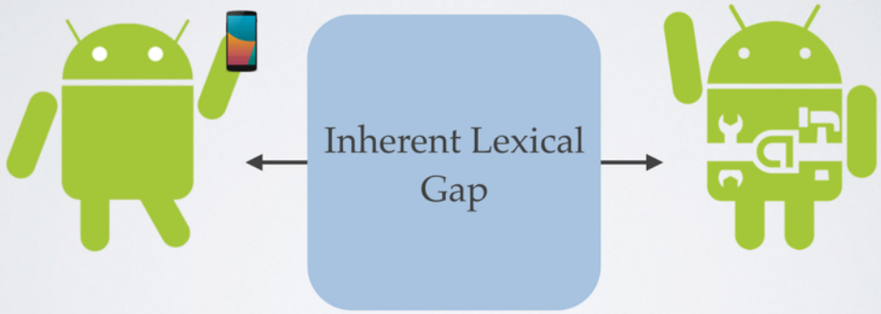


Images Courtesy of Google

THE LEXICAL GAP IN BUG REPORTING

Reporters:
-Functional Knowledge of a Software Bug.

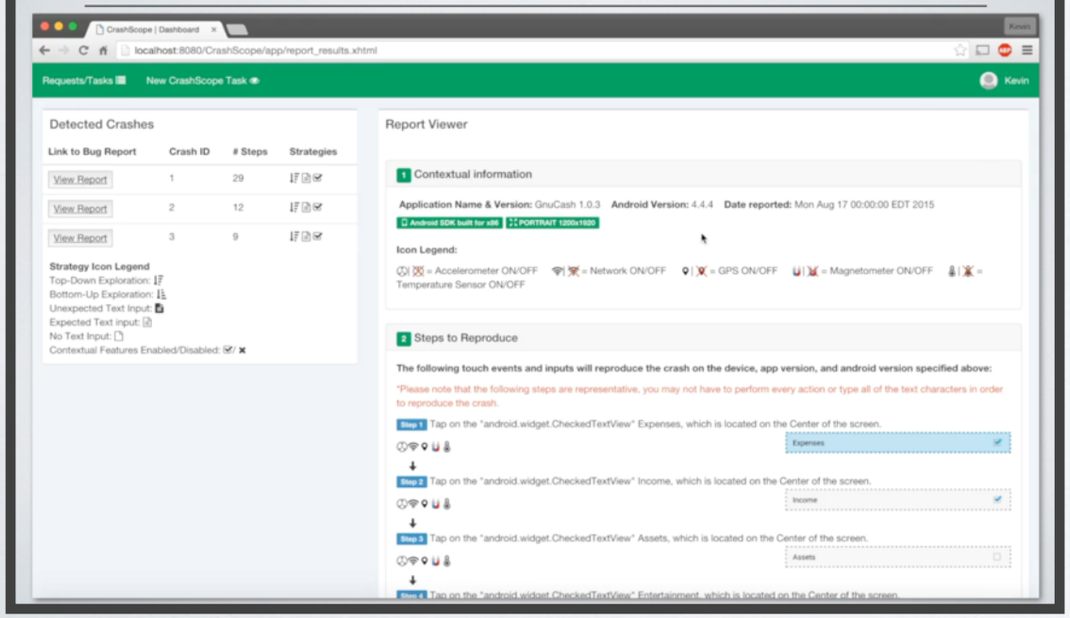
Developers:
-Intimate Code Level Knowledge of Application



Images Courtesy of Google

CRASHSCOPE: A PRACTICAL TOOL

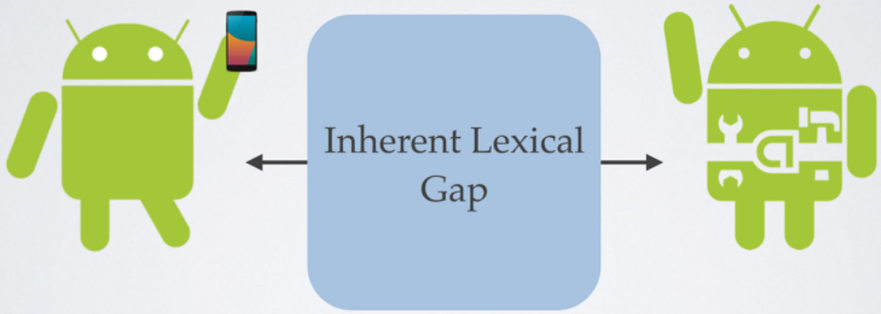
HUAWEI



THE LEXICAL GAP IN BUG REPORTING

Reporters:
-Functional Knowledge of a Software Bug.

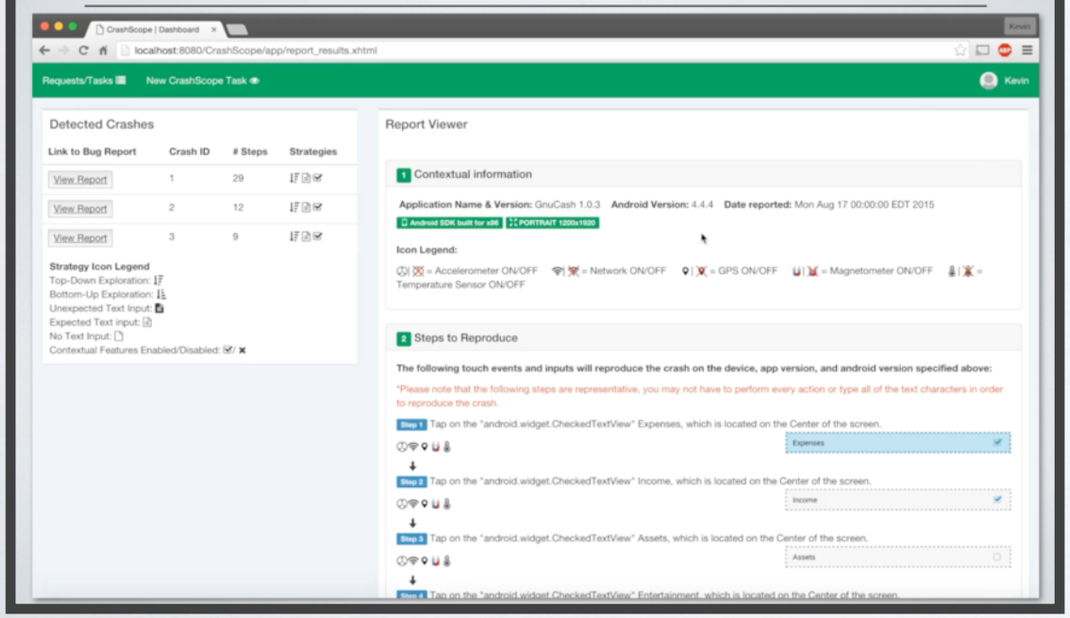
Developers:
-Intimate Code Level Knowledge of Application

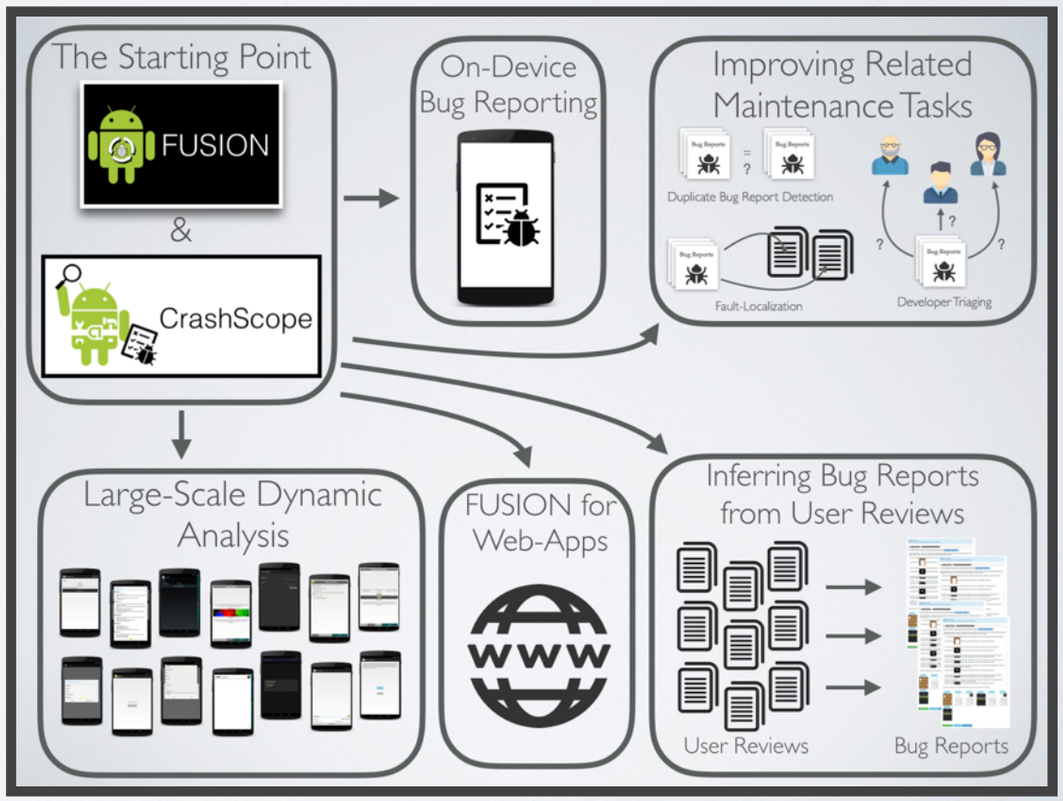


Images Courtesy of Google

CRASHSCOPE: A PRACTICAL TOOL

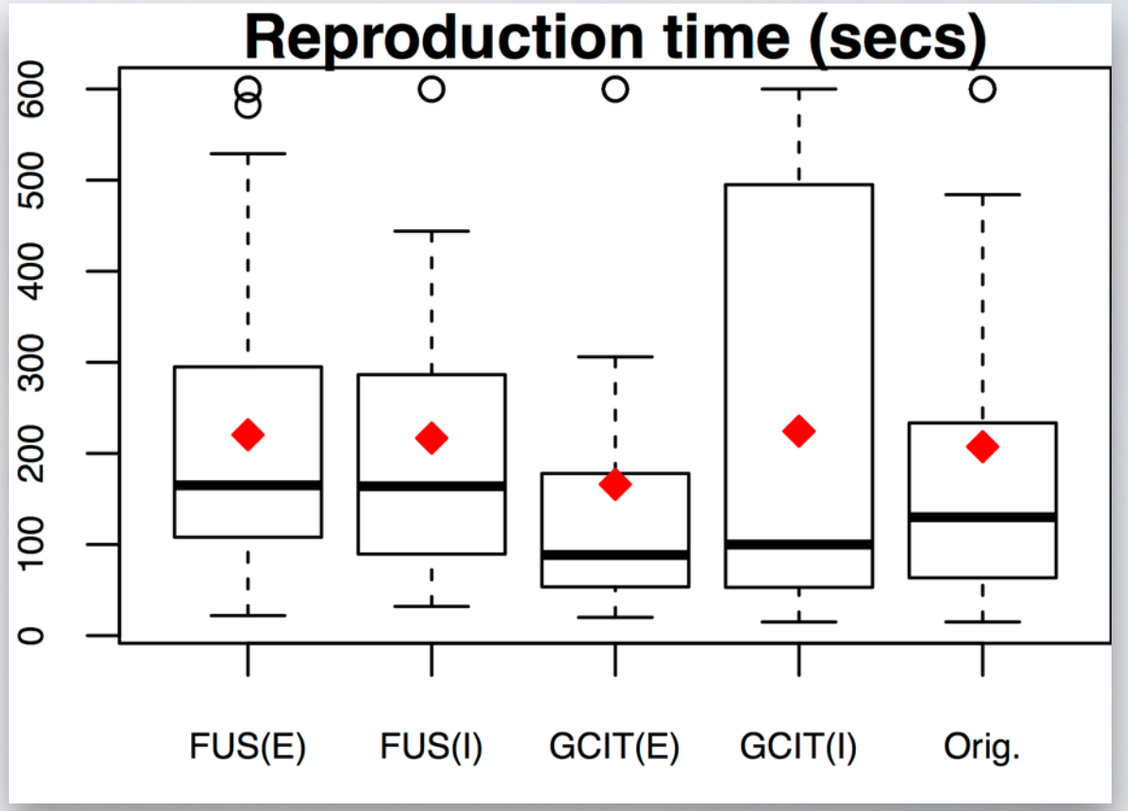
HUAWEI





RESULTS: REPRODUCTION TIME

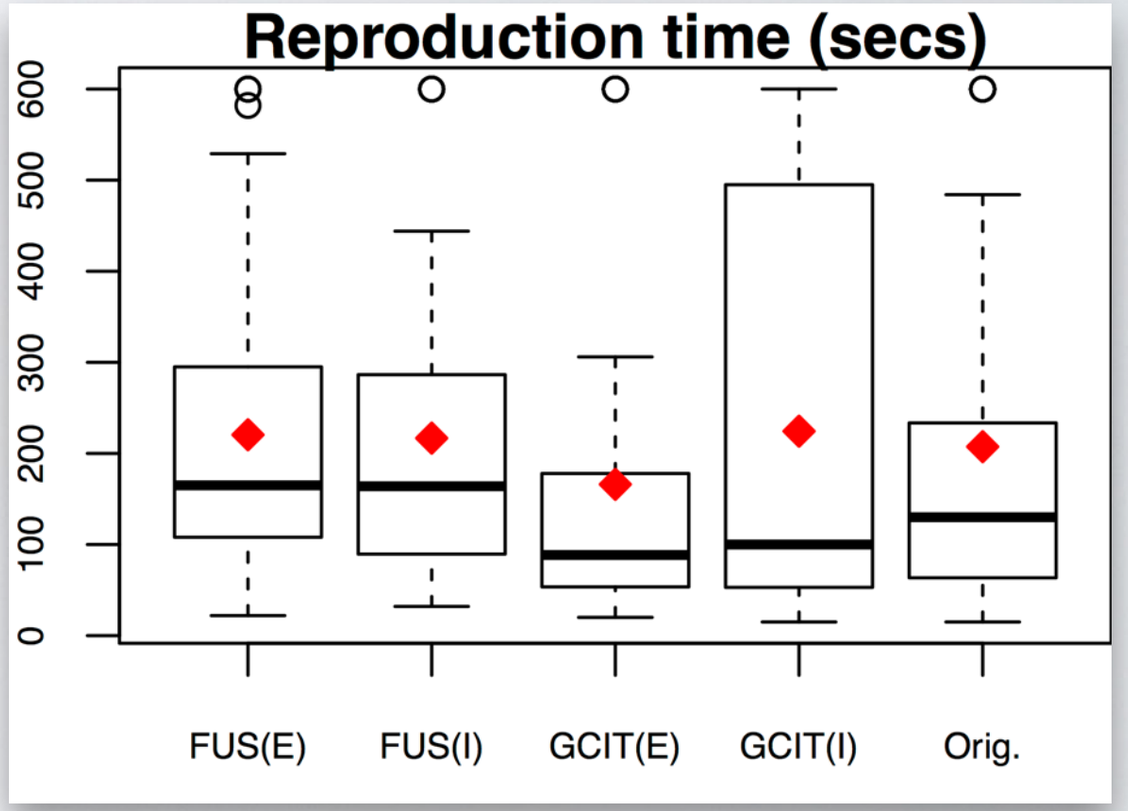
Bug Report Type	Avg Time to Reproduce
FUSION (E)	3:15
FUSION(I)	2:35
Google Code (E)	1:46
Google Code (I)	1:46
Original	1:59
FUSION Average	2:55
Google Code Average	1:46



Average Time to Reproduce Bug by Bug Report Type

RESULTS: REPRODUCTION TIME

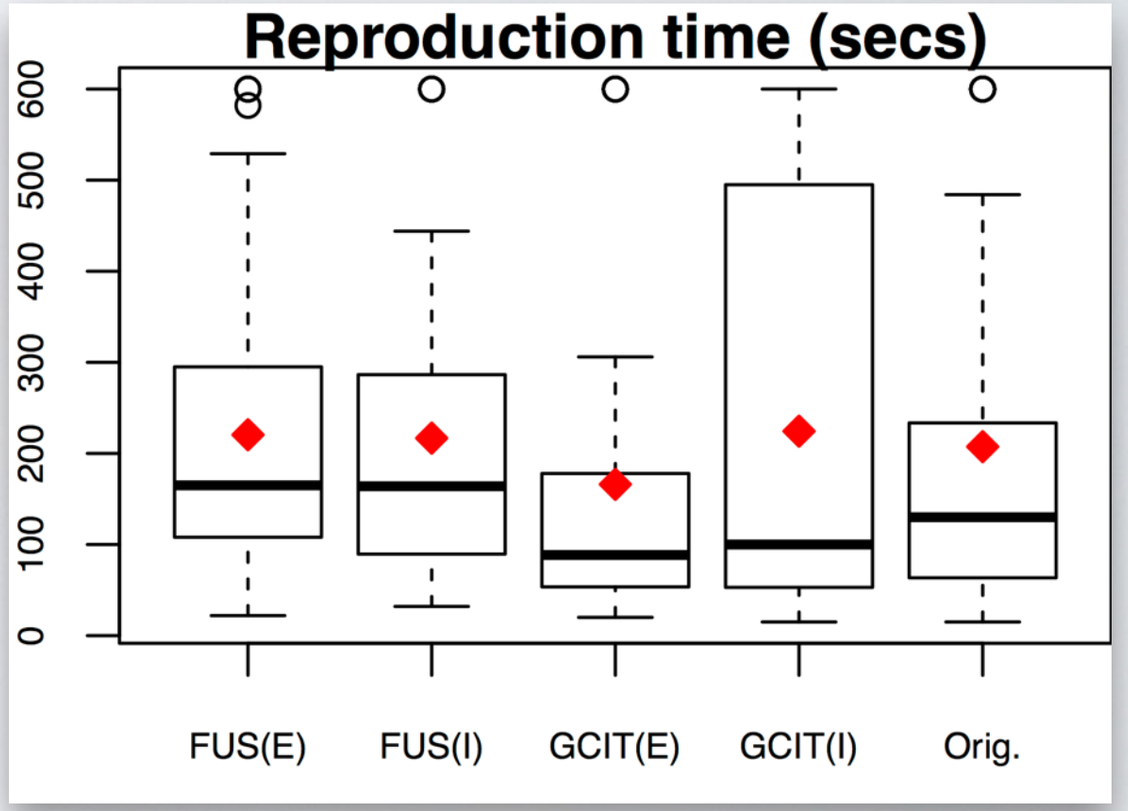
Bug Report Type	Avg Time to Reproduce
FUSION (E)	3:15
FUSION(I)	2:35
Google Code (E)	1:46
Google Code (I)	1:46
Original	1:59
FUSION Average	2:55
Google Code Average	1:46



Average Time to Reproduce Bug by Bug Report Type

RESULTS: REPRODUCTION TIME

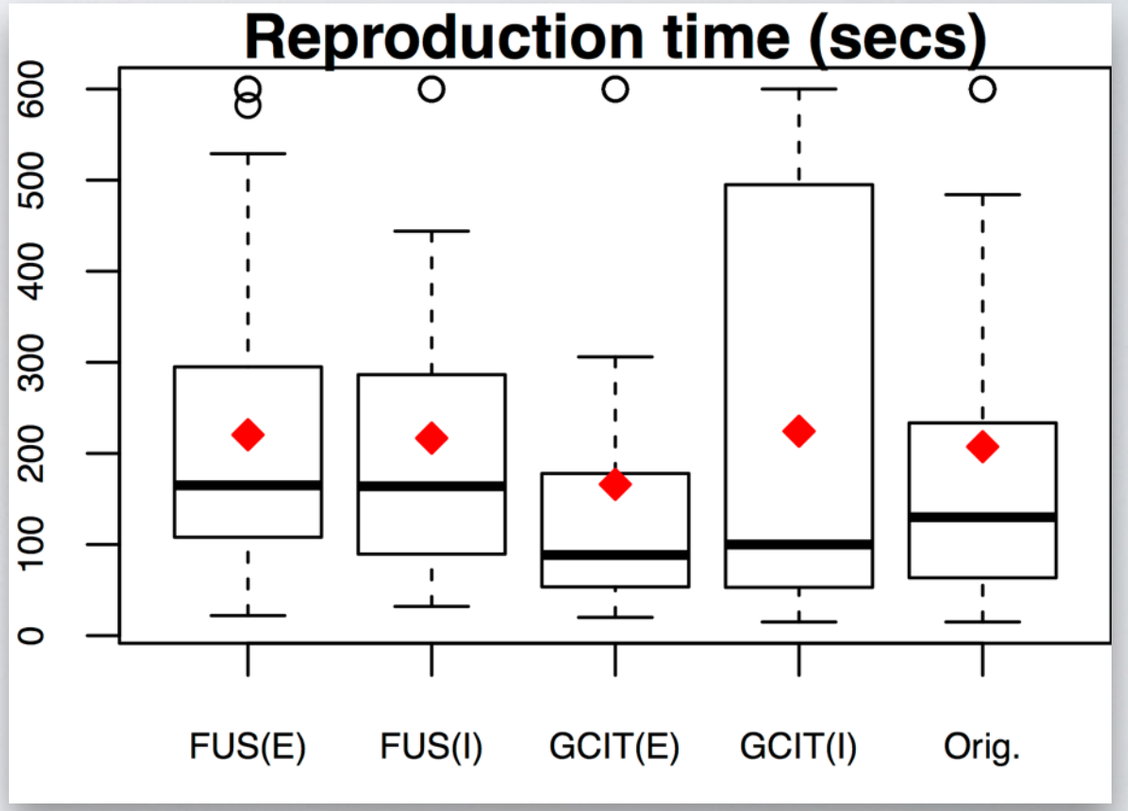
Bug Report Type	Avg Time to Reproduce
FUSION (E)	3:15
FUSION(I)	2:35
Google Code (E)	1:46
Google Code (I)	1:46
Original	1:59
FUSION Average	2:55
Google Code Average	1:46



Average Time to Reproduce Bug by Bug Report Type

RESULTS: REPRODUCTION TIME

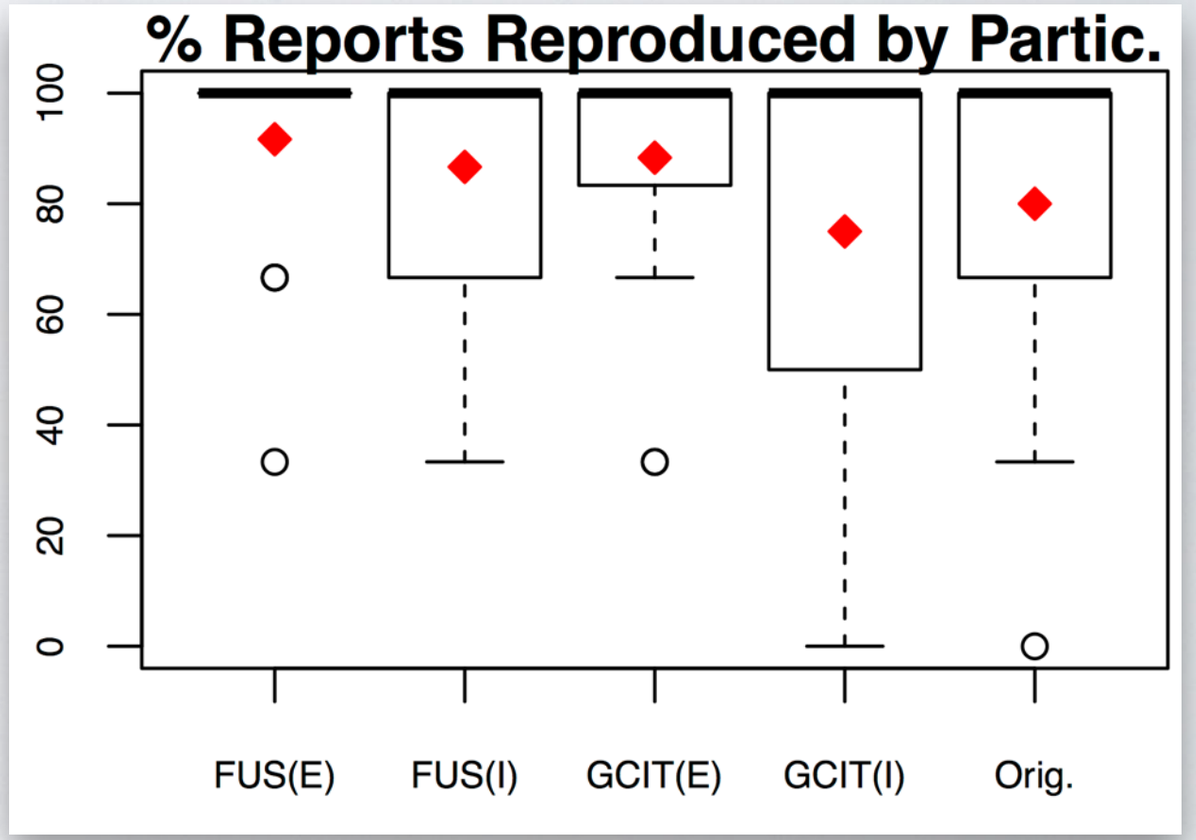
Bug Report Type	Avg Time to Reproduce
FUSION (E)	3:15
FUSION(I)	2:35
Google Code (E)	1:46
Google Code (I)	1:46
Original	1:59
FUSION Average	2:55
Google Code Average	1:46



Average Time to Reproduce Bug by Bug Report Type

RESULTS: REPRODUCTION

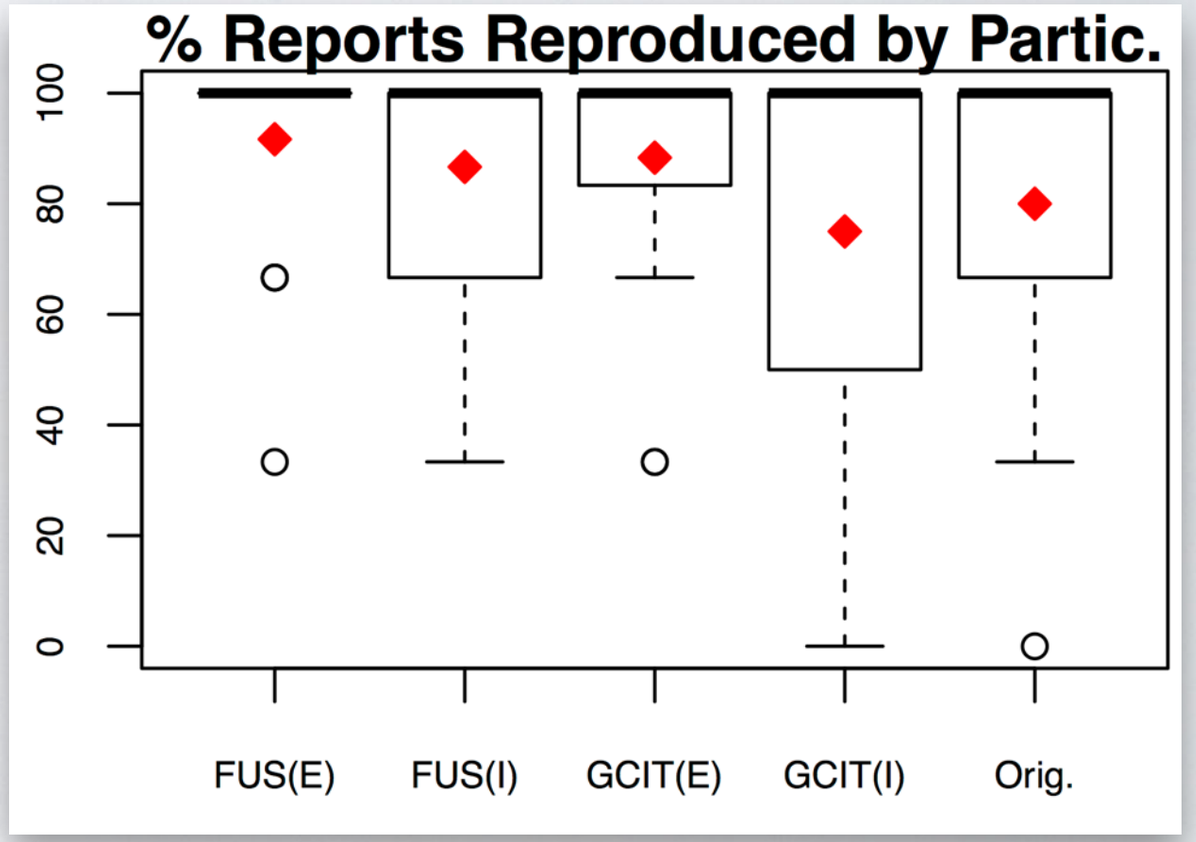
Bug Report Type	# of Bugs that were not reproduced
FUSION (E)	5
FUSION(I)	8
Google Code (E)	8
Google Code (I)	15
Original	11
FUSION Total	13
Google Code Total	23



% of Bugs reproduced by Bug Report Type

RESULTS: REPRODUCTION

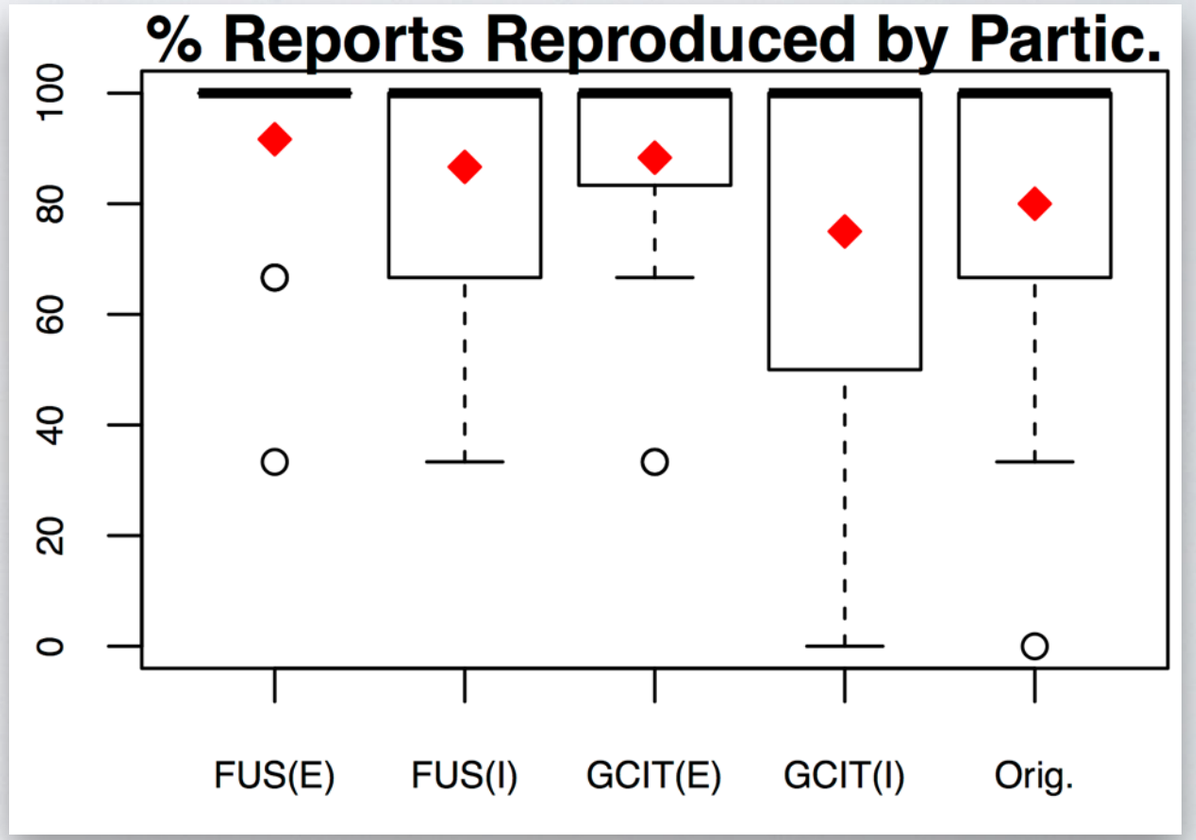
Bug Report Type	# of Bugs that were not reproduced
FUSION (E)	5
FUSION(I)	8
Google Code (E)	8
Google Code (I)	15
Original	11
FUSION Total	13
Google Code Total	23



% of Bugs reproduced by Bug Report Type

RESULTS: REPRODUCTION

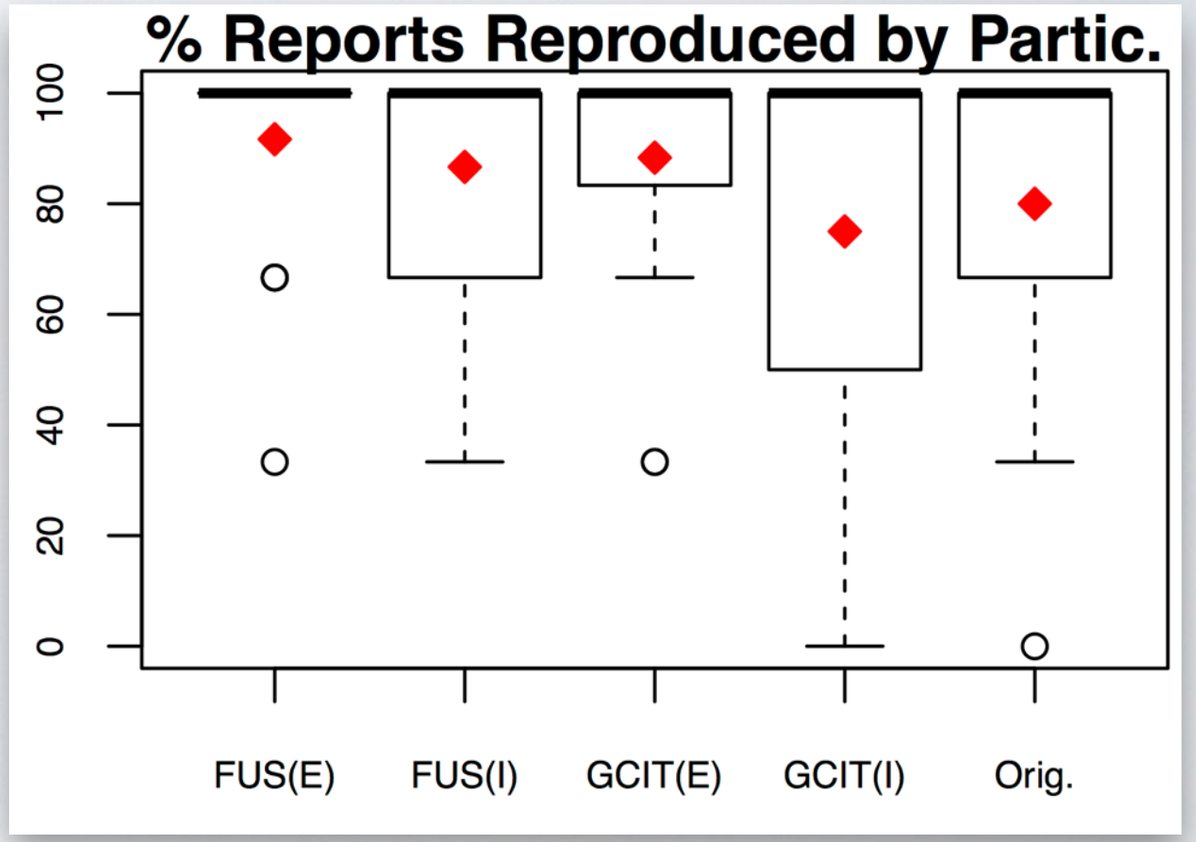
Bug Report Type	# of Bugs that were not reproduced
FUSION (E)	5
FUSION(I)	8
Google Code (E)	8
Google Code (I)	15
Original	11
FUSION Total	13
Google Code Total	23



% of Bugs reproduced by Bug Report Type

RESULTS: REPRODUCTION

Bug Report Type	# of Bugs that were not reproduced
FUSION (E)	5
FUSION(I)	8
Google Code (E)	8
Google Code (I)	15
Original	11
FUSION Total	13
Google Code Total	23



% of Bugs reproduced by Bug Report Type

DYNAMIC PROGRAM ANALYZER (ENGINE)

- Extracts run-time information of components exercised.

- Extracts the XML GUI Hierarchy using UIAutomator subroutines.

- Able to detect when execution leaves the subject app, and re-launch the app.

Details for step 4

6 I -- Select action/event -- -- Select GUI component-- ?

7 Additional information:

Type any additional information for this step

More steps? Yes (next step) + No, I am done

Details for step 4

6 I -- Select action/event -- -- Select GUI component-- ?

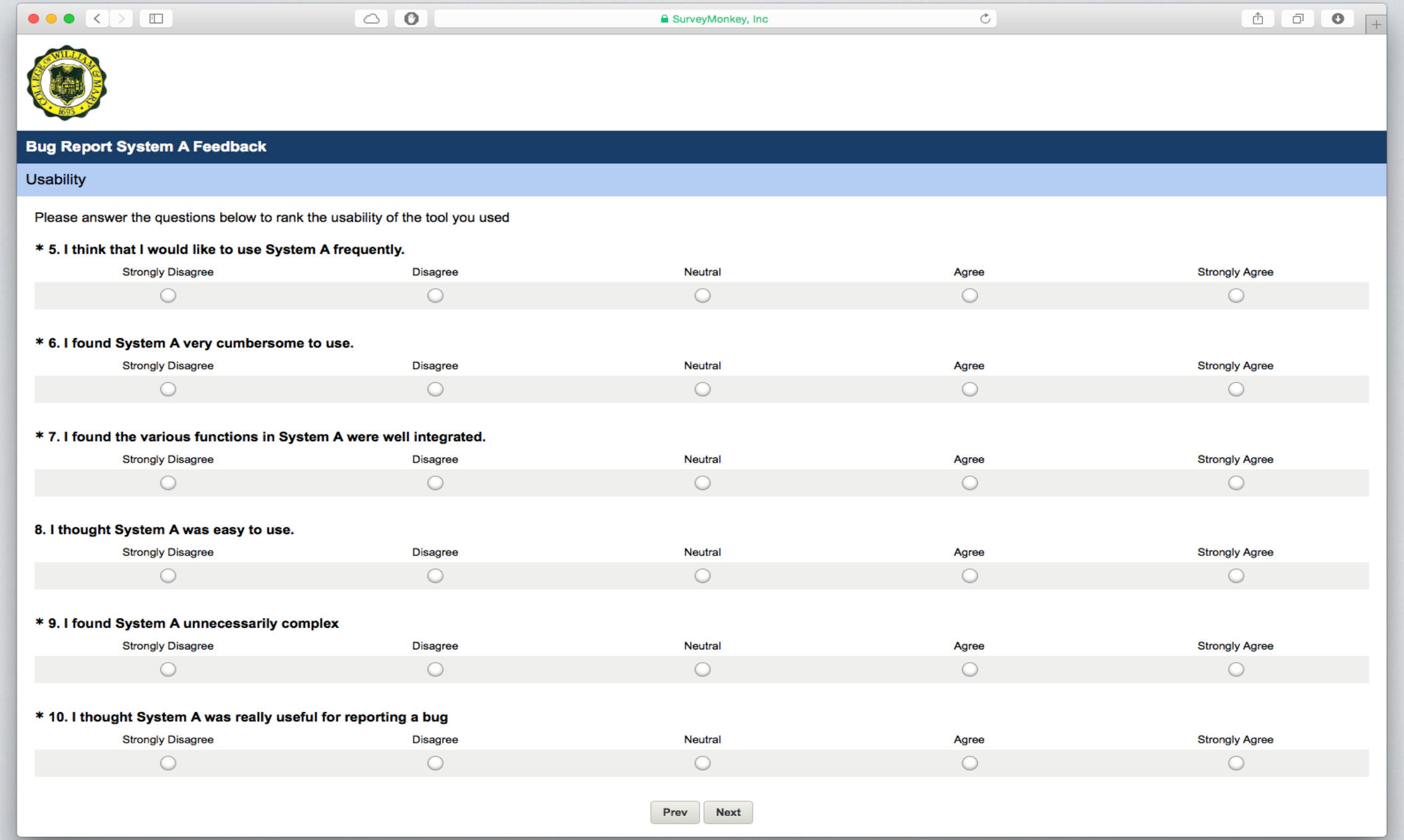
7 Additional information:

Type any additional information for this step

More steps? Yes (next step) + No, I am done

CONTEXT: BUG REPORTS USED IN THE STUDY

CONTEXT: BUG REPORTS USED IN THE STUDY



USER EXPERIENCE (UX) QUESTIONS

Question Identifier	Question
UX1	I think that I would like to have this type of bug report/system frequently.
UX2	I found this type of bug report/system unnecessarily complex.
UX3	I thought this type of bug report/system was easy to read/use.
UX4	I found this type of bug report/system very cumbersome to read/use.
UX5	I thought the bug report/system was really useful for reporting/reproducing the bug

USER PREFERENCE (UP) QUESTIONS

Question Identifier	Question
UP1	What information from this <system> did you find useful for reporting/reproducing the bug?
UP2	What other information (if any) would you like to see in this <system>?
UP3	What elements do you like the most from this <system>?
UP4	What elements do you like the least from this <system>?

BUG REPORTING TIME RESULTS: FUSION

Bug Index	App	Participant #1 (Experienced)	Participant #2 (Experienced)	Participant #3 (Inexperienced)	Participant #4 (Inexperienced)
1	A Time Tracker	7:48	11:30	24:30	2:01
2	Aarddict	4:12	4:10	3:30	4:51
3	ACV	2:27	5:30	8:18	05:14
4	Car Report	12:21	**4:50***	15:45	**8:00***
5	Document	**4:03***	5:10	**16:32***	**6:38***
6	Droid Weight	**3:10***	**2:10***	**7:43***	6:09
7	Eshotroid	7:30	6:30	10:29	6:21
8	GnuCash	9:45	**7:10***	18:45	08:23
9	GnuCash	9:23	7:30	20:03	9:27
10	Mileage	**2:22***	5:10	7:07	**3:04***
11	NetMBuddy	2:02	3:15	4:00	1:27
12	Notepad	3:53	3:20	4:45	3:14
13	OI Notepad	5:15	9:20	13:30	6:17
14	Olam	1:23	2:20	2:30	1:40
15	QuickDic	2:58	2:10	2:40	2:01
	Average	**5:14**	**5:20**	**10:40**	**4:59**

BUG REPORTING TIME RESULTS: FUSION

Bug Index	App	Participant #1 (Experienced)	Participant #2 (Experienced)	Participant #3 (Inexperienced)	Participant #4 (Inexperienced)
1	A Time Tracker	7:48	11:30	24:30	2:01
2	Aarddict	4:12	4:10	3:30	4:51
3	ACV	2:27	5:30	8:18	05:14
4	Car Report	12:21	**4:50***	15:45	**8:00***
5	Document	**4:03***	5:10	**16:32***	**6:38***
6	Droid Weight	**3:10***	**2:10***	**7:43***	6:09
7	Eshotroid	7:30	6:30	10:29	6:21
8	GnuCash	9:45	**7:10***	18:45	08:23
9	GnuCash	9:23	7:30	20:03	9:27
10	Mileage	**2:22***	5:10	7:07	**3:04***
11	NetMBuddy	2:02	3:15	4:00	1:27
12	Notepad	3:53	3:20	4:45	3:14
13	OI Notepad	5:15	9:20	13:30	6:17
14	Olam	1:23	2:20	2:30	1:40
15	QuickDic	2:58	2:10	2:40	2:01
	Average	**5:14**	**5:20**	**10:40**	**4:59**

BUG REPORTING TIME RESULTS: FUSION

Bug Index	App	Participant #1 (Experienced)	Participant #2 (Experienced)	Participant #3 (Inexperienced)	Participant #4 (Inexperienced)
1	A Time Tracker	7:48	11:30	24:30	2:01
2	Aarddict	4:12	4:10	3:30	4:51
3	ACV	2:27	5:30	8:18	05:14
4	Car Report	12:21	**4:50***	15:45	**8:00***
5	Document	**4:03***	5:10	**16:32***	**6:38***
6	Droid Weight	**3:10***	**2:10***	**7:43***	6:09
7	Eshotroid	7:30	6:30	10:29	6:21
8	GnuCash	9:45	**7:10***	18:45	08:23
9	GnuCash	9:23	7:30	20:03	9:27
10	Mileage	**2:22***	5:10	7:07	**3:04***
11	NetMBuddy	2:02	3:15	4:00	1:27
12	Notepad	3:53	3:20	4:45	3:14
13	OI Notepad	5:15	9:20	13:30	6:17
14	Olam	1:23	2:20	2:30	1:40
15	QuickDic	2:58	2:10	2:40	2:01
	Average	**5:14**	**5:20**	**10:40**	**4:59**

BUG REPORTING TIME RESULTS: GCIT

Bug Index	App	Participant #1 (Experienced)	Participant #2 (Experienced)	Participant #3 (Inexperienced)	Participant #4 (Inexperienced)
1	A Time Tracker	4:16	7:30	1:51	1:56
2	Aarddict	3:33	8:25	2:13	2:22
3	ACV	2:37	11:10	0:51	1:42
4	Car Report	2:52	12:23	0:40	2:39
5	Document	3:15	9:31	0:45	1:46
6	Droid Weight	2:33	7:13	1:03	1:45
7	Eshotroid	2:08	5:27	1:47	1:03
8	GnuCash	2:40	6:48	1:15	2:30
9	GnuCash	6:20	5:12	1:40	2:22
10	Mileage	3:53	5:25	1:00	1:16
11	NetMBuddy	3:52	3:13	1:20	1:48
12	Notepad	2:02	4:32	1:01	1:23
13	OI Notepad	3:16	6:25	0:58	1:12
14	Olam	4:26	3:13	1:16	1:49
15	QuickDic	1:37	03:17	0:55	0:59
	Average	**3:17**	**6:39**	**1:14**	**1:46**

BUG REPORTING TIME RESULTS: GCIT

Bug Index	App	Participant #1 (Experienced)	Participant #2 (Experienced)	Participant #3 (Inexperienced)	Participant #4 (Inexperienced)
1	A Time Tracker	4:16	7:30	1:51	1:56
2	Aarddict	3:33	8:25	2:13	2:22
3	ACV	2:37	11:10	0:51	1:42
4	Car Report	2:52	12:23	0:40	2:39
5	Document	3:15	9:31	0:45	1:46
6	Droid Weight	2:33	7:13	1:03	1:45
7	Eshotroid	2:08	5:27	1:47	1:03
8	GnuCash	2:40	6:48	1:15	2:30
9	GnuCash	6:20	5:12	1:40	2:22
10	Mileage	3:53	5:25	1:00	1:16
11	NetMBuddy	3:52	3:13	1:20	1:48
12	Notepad	2:02	4:32	1:01	1:23
13	OI Notepad	3:16	6:25	0:58	1:12
14	Olam	4:26	3:13	1:16	1:49
15	QuickDic	1:37	03:17	0:55	0:59
	Average	**3:17**	**6:39**	**1:14**	**1:46**

BUG REPORTING TIME RESULTS: GCIT

Bug Index	App	Participant #1 (Experienced)	Participant #2 (Experienced)	Participant #3 (Inexperienced)	Participant #4 (Inexperienced)
1	A Time Tracker	4:16	7:30	1:51	1:56
2	Aarddict	3:33	8:25	2:13	2:22
3	ACV	2:37	11:10	0:51	1:42
4	Car Report	2:52	12:23	0:40	2:39
5	Document	3:15	9:31	0:45	1:46
6	Droid Weight	2:33	7:13	1:03	1:45
7	Eshotroid	2:08	5:27	1:47	1:03
8	GnuCash	2:40	6:48	1:15	2:30
9	GnuCash	6:20	5:12	1:40	2:22
10	Mileage	3:53	5:25	1:00	1:16
11	NetMBuddy	3:52	3:13	1:20	1:48
12	Notepad	2:02	4:32	1:01	1:23
13	OI Notepad	3:16	6:25	0:58	1:12
14	Olam	4:26	3:13	1:16	1:49
15	QuickDic	1:37	03:17	0:55	0:59
	Average	**3:17**	**6:39**	**1:14**	**1:46**

EASE OF USE: WHAT DID WE LEARN?

- **RQ_1:** Is FUSION easier to use for reporting/reproducing bugs than traditional bug tracking systems?

  - FUSION is about as easy for developers to use as a traditional bug tracking system
  - FUSION is more difficult for inexperienced users to use than traditional bug tracking systems

BUG REPORTING UX: WHAT DID WE LEARN?

- **RQ_2:** What types of information fields do developers/testers consider important when reporting and reproducing bugs in Android?

- While reporters generally felt that the opportunity to enter extra information in a bug report using FUSION increased the quality of their reports, inexperienced users would have preferred a simpler web UI.

REPRODUCTION TIME: WHAT DID WE LEARN?

- ***RQ4:*** Do bug reports generated with FUSION allow for faster bug reproduction compared to reports submitted using traditional bug tracking systems?

- Bug reports generated with FUSION ***do not allow for faster reproduction*** of bugs compared bug reports generated using traditional bug tracking systems such as the GCIT.

REPRODUCTION UX: WHAT DID WE LEARN?

- ***RQ$_2$:*** Is FUSION easier to use for reporting/reproducing bugs than traditional bug tracking systems?

- Participants preferred FUSION over the original bug reports and GCIT over FUSION
- Some participants thought the FUSION steps were overly detailed.

BUG REPORTING UX: WHAT DID WE LEARN?

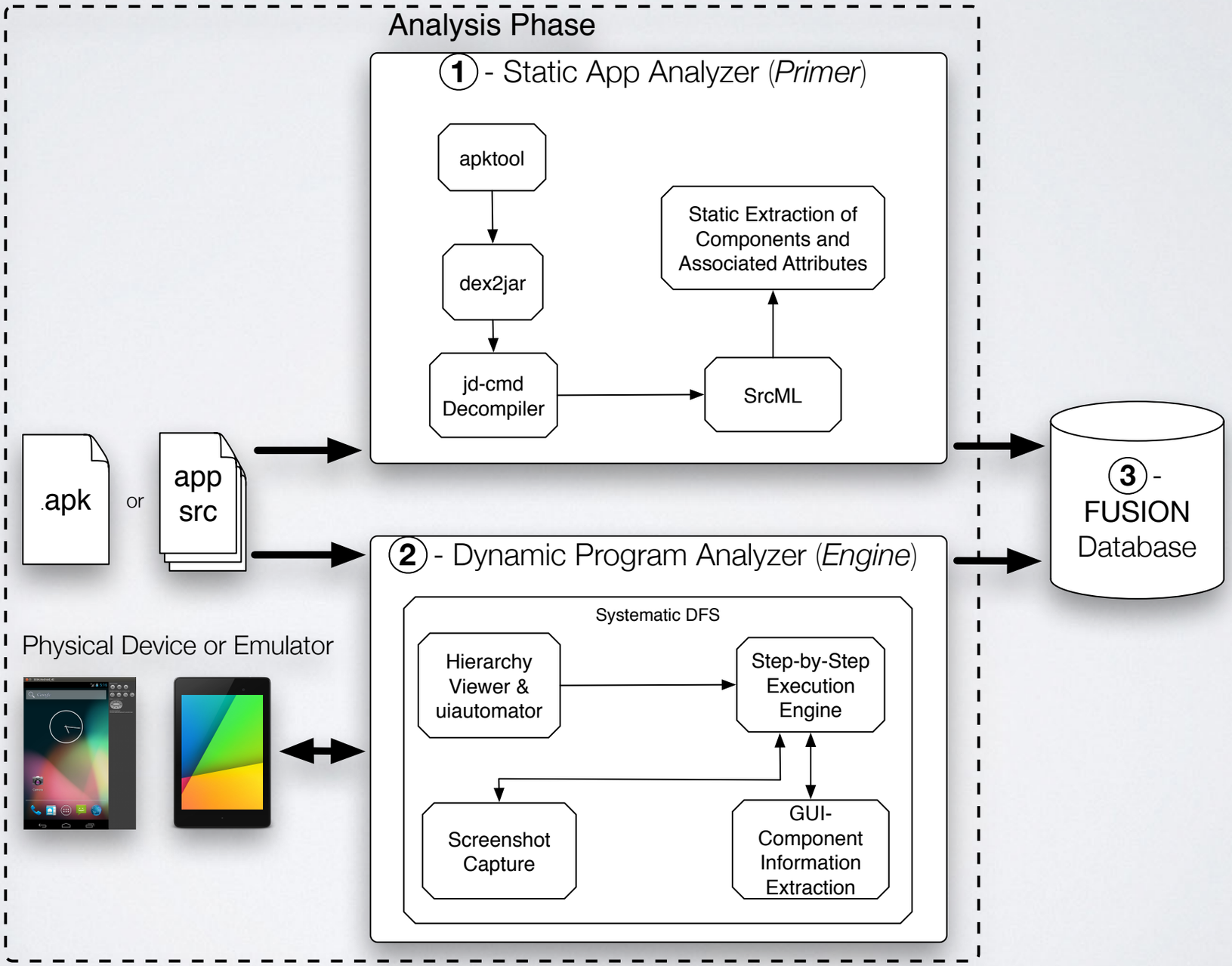
- ***What elements do you like most from the system?***
- Experienced User: *"The GUI component form and the action/event. They provide an easy way to report the steps."*
- Inexperienced User: *"The parts where you could simply type out the issue"*

BUG REPORTING UX: WHAT DID WE LEARN?

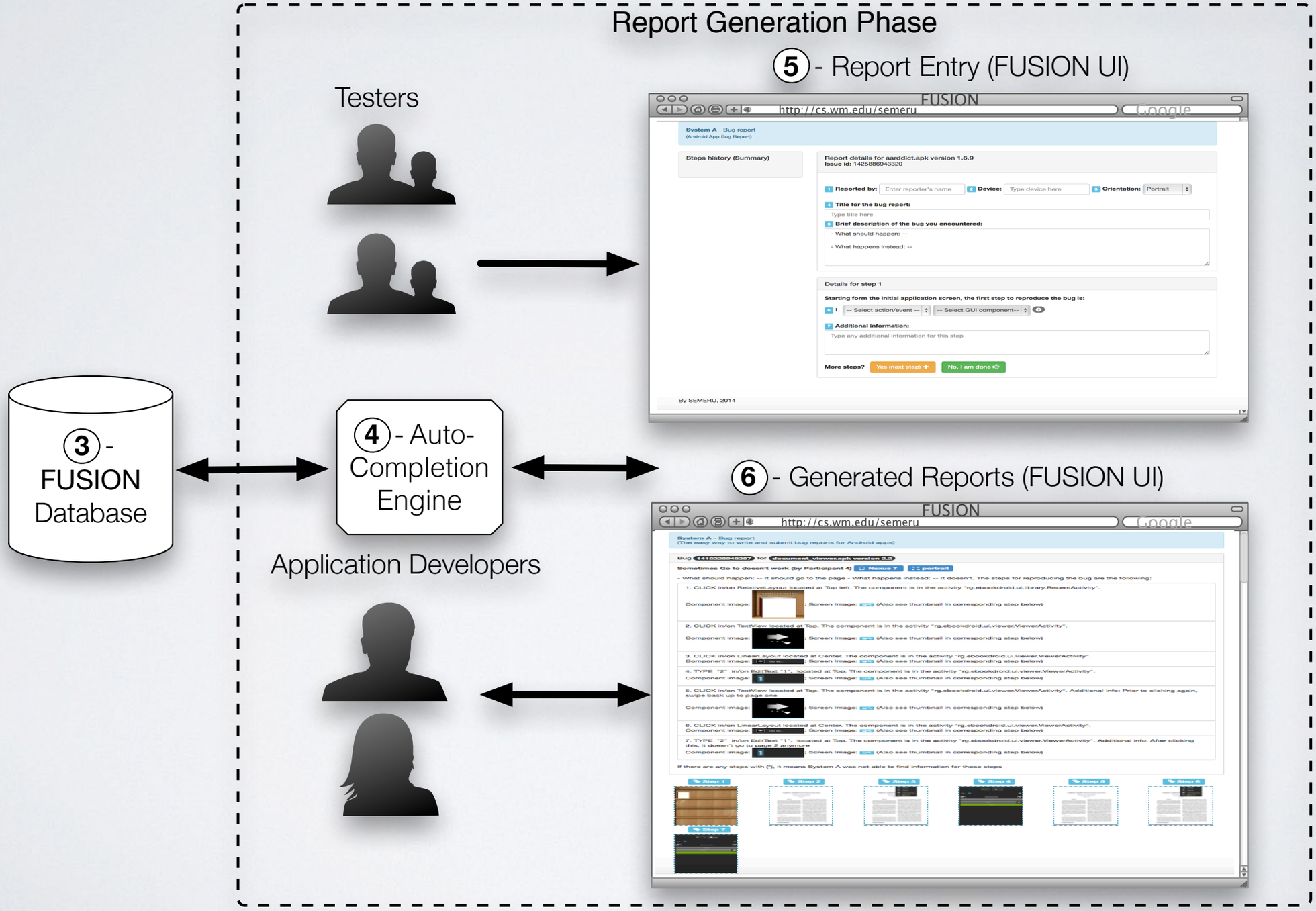
- ***RQ$_3$:*** *Do developers/testers using FUSION reproduce more bugs compared to traditional bug tracking systems?*

- Developers using FUSION ***are able to reproduce more bugs*** compared to traditional bug tracking systems such as the GCIT.

FUSION: ANALYSIS PHASE

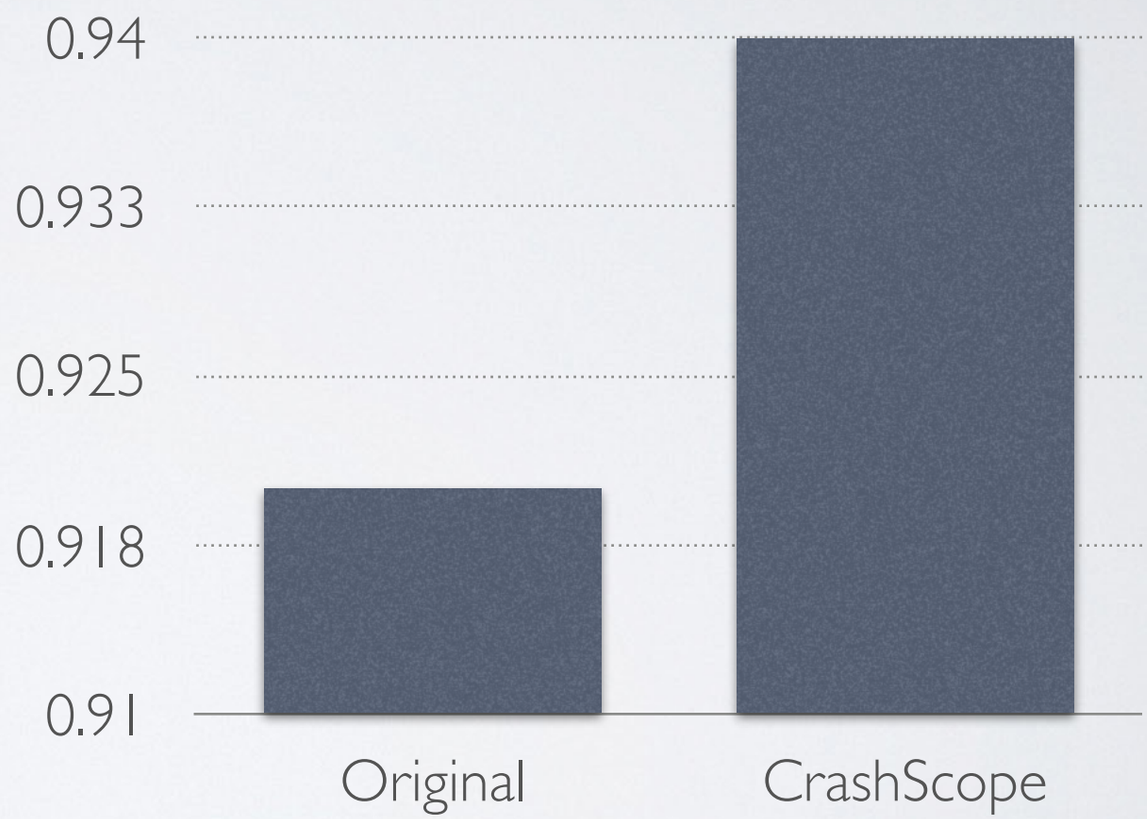


FUSION: REPORT GENERATION PHASE



STUDY 2: REPRODUCIBILITY RESULTS

Type of Crash Report	# of Total/Non-Reproducible Reports
Original Bug Reports	59/64
CrashScope Bug Reports	60/64



% of Bug Reports Reproduced by Type

STUDY 2: REPRODUCIBILITY RESULTS

-CrashScope reports are about as reproducible as other reports

STUDY 1: SUMMARY OF FINDINGS

- **RQ1:** CrashScope is nearly as effective at discovering crashes as the other tools, without reporting crashes caused by instrumentation

- **RQ2&3:** CrashScope's differing strategies led to the discovery of unique crashes

- **RQ4:** Higher statement coverage does not necessarily correspond with crash detection capabilities

STUDY 1: EXPERIMENTAL SETUP

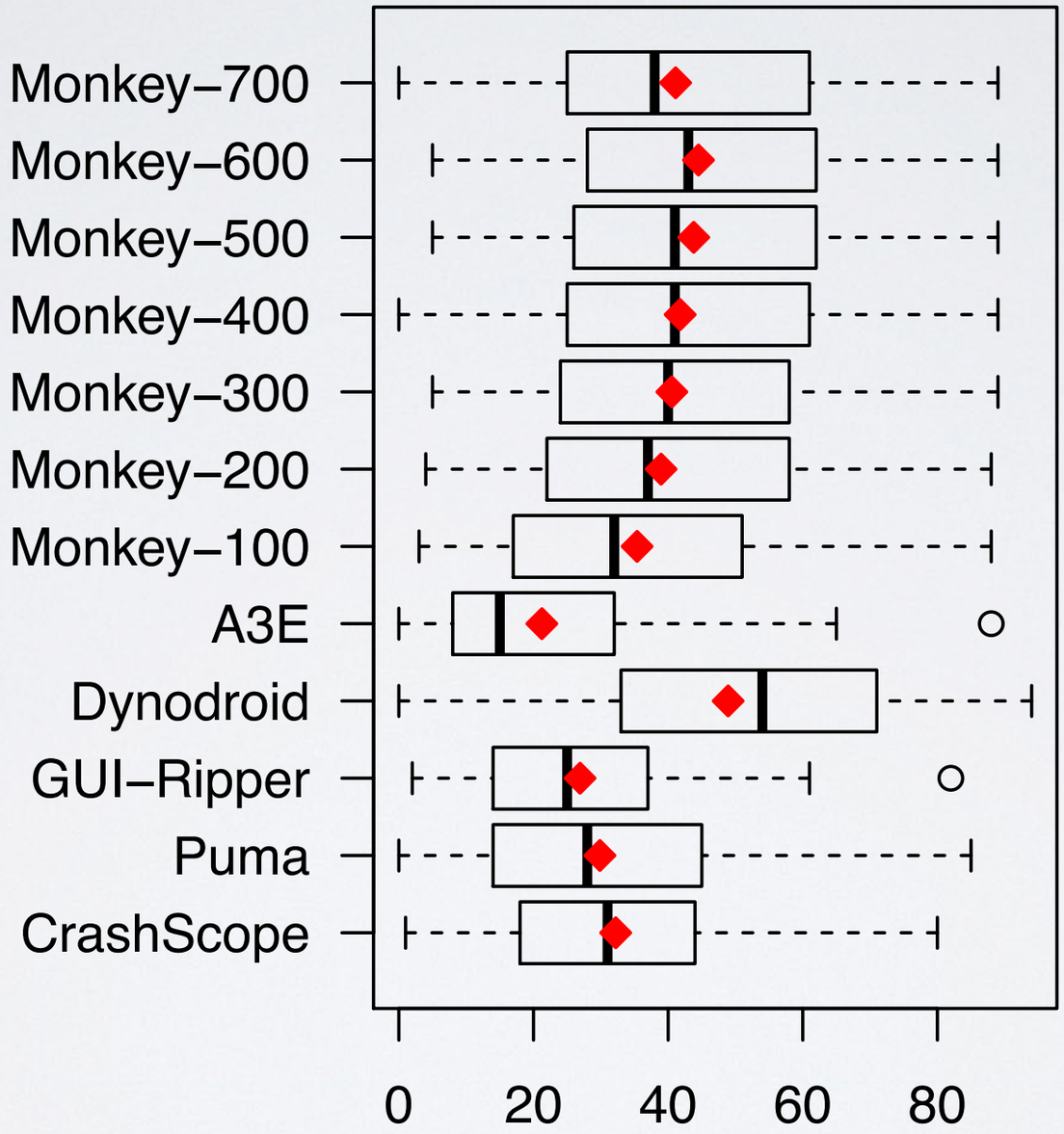
TOOLS USED IN THE COMPARATIVE FAULT FINDING STUDY

Tool Name	Android Version	Tool Type
Monkey	Any	Random
A3E Depth-First	Any	Systematic
GUI-Ripper	Any	Model-Based
Dynodroid	v2.3	Random-Based
PUMA	v4.1+	Random-Based

- 61 subject applications from the ***Androtest***[1] toolset
- Each testing tool was run 5 separate times for 1 hour, whereas CrashScope ran through all strategies
- Monkey was limited by the number of events

[1]S. R. Choudhary, A. Gorla, and A. Orso. Automated Test Input Generation for Android: Are we there yet? In 30th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015), 2015

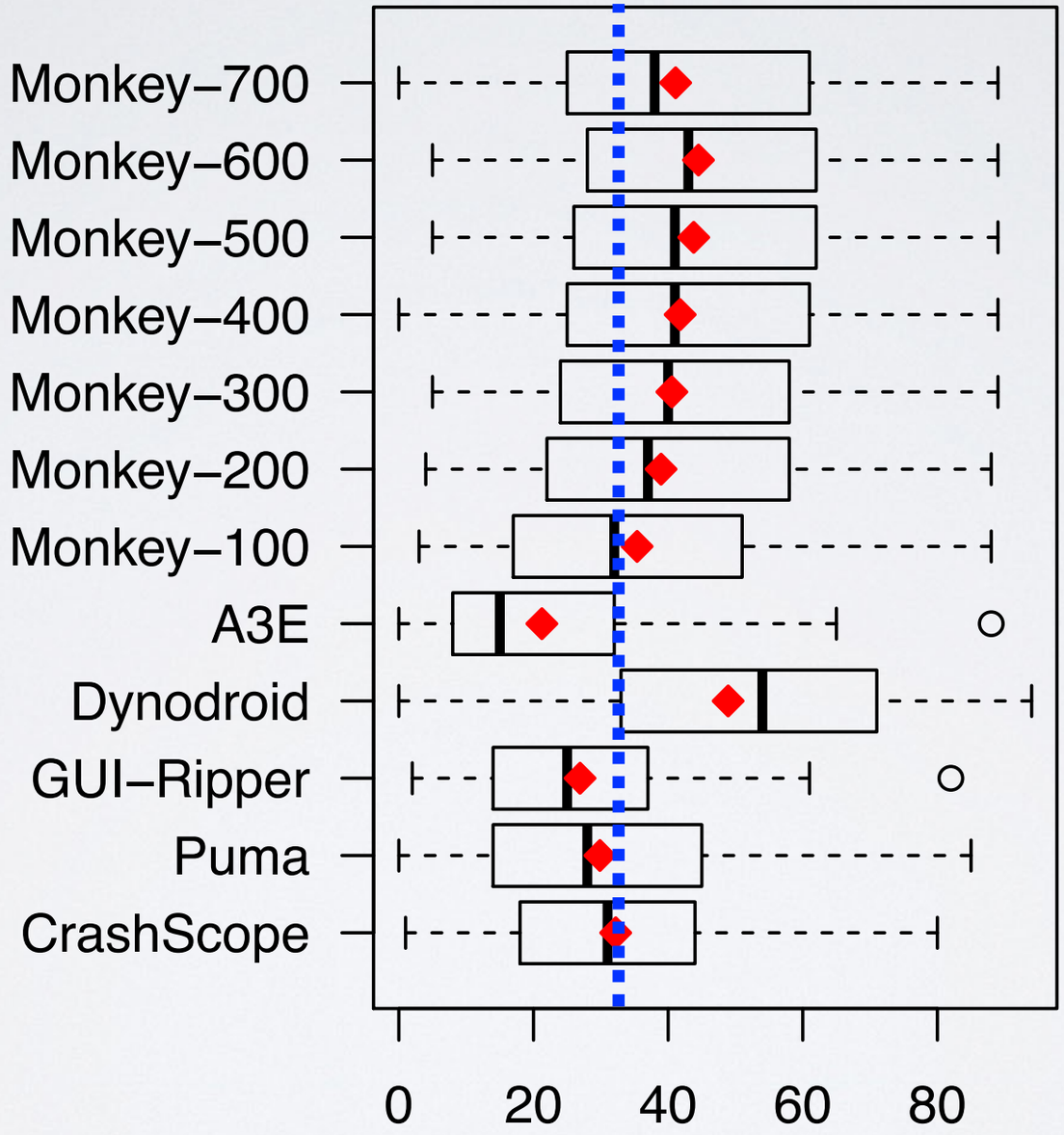
STUDY 1: STATEMENT COVERAGE RESULTS



Average Statement Coverage Results for the Comparative Study

Reported in Average %

STUDY 1: STATEMENT COVERAGE RESULTS



Average Statement Coverage Results for the Comparative Study

Reported in Average %

STUDY 2: EXPERIMENTAL SETUP

- 8 Real-World Crash Reports from Open Source Apps
- 16 Graduate Students from the College of William & Mary

Application Name	# of Reproduction Steps
BMI	4
Schedule	7
adsdroid	2
Anagram-solver	7
Eyecam	14
GNU Cash	29
Olam	2
CardGame Scores	23

- Each student attempted to reproduce 8 bugs: 4 from the original reports, 4 from CrashScope Reports

- Participants used a Nexus 7 tablet for reproduction

STUDY 2: SUMMARY OF FINDINGS

- **RQ_5:** Reports generated by CrashScope are about as reproducible as human written reports extracted from open-source issue trackers

- **RQ_6**: Reports generated by CrashScope are more readable and useful from a developers' perspective compared to human-written reports.

RESEARCH QUESTIONS

- ***RQ1:*** Ease of Use?
- ***RQ2:*** Information Preferences?
- ***RQ3:*** Reproducibility of Reports?
- ***RQ4:*** Speed of Reproduction?

RESEARCH QUESTIONS

- *RQ1:* FUSION is about as *easy for developers to use* as traditional bug-tracking systems

- *RQ2:* Extra Information *increased quality* of reports

- *RQ3:* FUSION reports are *more reproducible* than traditional bug reports

- *RQ4:* Developers take *slightly longer* to reproduce FUSION Reports than traditional reports