Automated Support for Mobile Application Testing and Maintenance

Mattia Fazzini
Advisor: Dr. Alessandro Orso
Mobile Applications
Mobile Applications
Bugs
Testing
Failures
Maintenance
Goals

Find more bugs early with novel testing techniques

Quickly resolve bugs with novel maintenance techniques
Research Outline

**Test Generation**
- BARISTA
  - ICST’17

**GUI Inconsistency Identification**
- DIFFDROID
  - ASE’17

**Bug Report Analysis**
- YAKUSU
  - ISSTA’18

**API-Usage Update**
- APP EVOLVE
  - Work in Progress

**Enhanced Bug Reporting**
- REPORT ASSIST
  - Work in Progress

**API Inconsistency Identification**
- DIFFDROID API
  - Work in Progress
Research Outline

- **Test Generation**
  - BARISTA
    - ICST’17

- **GUI Inconsistency Identification**
  - DIFFDROID
    - ASE’17

- **Bug Report Analysis**
  - YAKUSU
    - ISSTA’18

- **API-Usage Update**
  - APP EVOLVE
    - Work in Progress

- **Enhanced Bug Reporting**
  - REPORT ASSIST
    - Work in Progress

- **API Inconsistency Identification**
  - DIFFDROID API
    - Work in Progress
Mobile App Testing

Challenges

- Not enough time to test: 52%
- Do not have the right testing process/method: 47%
- Do not have the right tools to test: 46%

Practice: 64%
Manual Testing: 36%
Mobile App Testing

Challenges

- Not enough time to test: 52%
- Do not have the right testing process/method: 47%
- Do not have the right tools to test: 46%
Motivating Example: Manual Testing

**Division by zero:** nine divided by zero results in error message

(1) Press nine
(2) Press divide
(3) Press zero
(4) Press equal
(5) Check message

Simple

Inefficient, Tedious, Error prone
Motivating Example: Manual Testing

**Division by zero:** nine divided by zero results in error message

1. Press nine
2. Press divide
3. Press zero
4. Press equal
5. Check message

---

Simple

Inefficient, Tedious, Error prone
Motivating Example: Manual Testing

Division by zero: nine divided by zero results in error message

(1) Press nine
(2) Press divide
(3) Press zero
(4) Press equal
(5) Check message

Simple  Inefficient, Tedious, Error prone
Motivating Example: Automated Testing

Division by zero: nine divided by zero results in error message

(1) Press nine
(2) Press divide
(3) Press zero
(4) Press equal
(5) Check message

1 public void divisionByZero() {
    2     onView(withText("9")) .perform(click());
    3     onView(withId(R.id.div)) .perform(click());
    4     onView(withText("0")) .perform(click());
    5     onView(withId(R.id.eq)) .perform(click());
    6     onView(withId(R.id.dsp))
    7     .check(matches(withText("Error")));
    8 }
Motivating Example: Automated Testing

Division by zero: nine divided by zero results in error message

1. Press nine
2. Press divide
3. Press zero
4. Press equal
5. Check message

```java
public void divisionByZero() {
    onView(withText("9")).perform(click());
    onView(withId(R.id.div)).perform(click());
    onView(withText("0")).perform(click());
    onView(withId(R.id.eq)).perform(click());
    onView(withId(R.id.dsp)).check(matches(withText("Error")));
}
```
Motivating Example: Automated Testing

Division by zero: nine divided by zero results in error message

(1) Press nine
(2) Press divide
(3) Press zero
(4) Press equal
(5) Check message

1 public void divisionByZero() {
2     onView(withText("9")).perform(click());
3     onView(withId(R.id.div)).perform(click());
4     onView(withText("0")).perform(click());
5     onView(withId(R.id.eq)).perform(click());
6     onView(withId(R.id.dsp))
7         .check(matches(withText("Error")));
8 }
Test Case Recording

Test Case Encoding

Test Case Execution

Application Under Test

Test Devices

Record Device

Recorded Trace

Test Case

Test Report

ICST’17
Test Case Recording

Application Under Test

Record Device

Recorded Trace

Test Case Encoding

Test Case

Test Case Devices

Test Case Execution

Test Report
Test cases successfully encoded and executed

User efficiency in generating test cases

46%

32%
Due to the wide array of available Android devices, we are targeting our support to a select number of Android devices to continue improving our overall Salesforce1 for Android user experience.
Due to the wide array of available Android devices, we are targeting our support to a select number of Android devices to continue improving our overall Salesforce1 for Android user experience.
Due to the wide array of available Android devices, we are targeting our support to a select number of Android devices to continue improving our overall Salesforce1 for Android user experience.
“Due to the wide array of available Android devices, we are targeting our support to a select number of Android devices to continue improving our overall Salesforce1 for Android user experience”
“Due to the wide array of available Android devices, we are targeting our support to a select number of Android devices to continue improving our overall Salesforce1 for Android user experience”
Motivating Example

LG G3

LG Optimus L70
GUI Inconsistencies

96
False Positives
14%
Bug Reports

Arkonos commented on Jul 5, 2017

FC upon selecting weekly backup frequency.

Steps to reproduce:
Settings -> Backup -> Backup frequency 100% for me, no other frequency settings.
Once such crashes, the app would reproduce this.
Pretty sure the same thing happened.

Version:
Phone: Nexus 5
Android 6.0.1
LinageOS 13.1
Kernel: 3.10.100-rc7

Onko commented on Mar 24, 2017

Steps to reproduce the behavior:
1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

afwang commented on Mar 25, 2017

I found this crash on a redreader crash log file:
redreader_crash_log_file

Steps to reproduce:
1. Launch redreader
2. Subscribe to a subscribed subreddit
3. Tap the refresh button in the app background
4. Swipe down while the subscribed subreddit is open

I think it might help to reproduce if you speed is throttled.

Trolldemorted commented on May 19, 2017

I have:
open and closed issues for:
https://github.com/WhisperSystem...

Description:

Steps to reproduce:

Actual behavior:
Twidere crashes

Steps to reproduce:
Longpress retweet icon underneath teh...
Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>

<Logcat information>

Steps to reproduce the behavior

1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

WordPress App

Developer
Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>
Logcat information>

Steps to reproduce the behavior

1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes
<Text paragraph>

WORDPRESS App

Developer
Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>
(Logcat information>

Steps to reproduce the behavior

1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes
<Text paragraph>

WordPress App

Developer
Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>
<br>Logcat information>

Steps to reproduce the behavior

1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

WORDPRESS App

Developer

GrbcVK91

Content (tap to add text and media)

POST SETTINGS
Motivating Example

**Bug Report**

Onko commented on Mar 24, 2017

<Text paragraph>

Logcat information>

Steps to reproduce the behavior

1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

**WORDPRESS App**

Developer

Unfortunately, WordPress has stopped.

OK
Motivating Example

**WORDPRESS App**

**Test Case**

```java
public void test() {
    // start a new post
    onView(
       (withId(R.id.fab_button))
    ).perform(click());

    // type something
    onView(
       (withId(R.id.post_title))
    ).perform(typeText("GrbcVK9I"));

    // tap on the publish button
    onView(
       (withId(R.id.menu_save_post))
    ).perform(click());
}
```
Motivating Example

**WORDPRESS App**

**Test Case**

```java
public void test() {
    // start a new post
    onView(
       (withId(R.id.fab_button))
        .perform(click()));

    // type something
    onView(
       (withId(R.id.post_title))
        .perform(typeText("GrbcVK9I")));

    // tap on the publish button
    onView(
       (withId(R.id.menu_save_post))
        .perform(click()));
}
```
Motivating Example

**WORDPRESS App**

**Test Case**

```java
public void test() {
    // start a new post
    onView(
       (withId(R.id.fab_button))
        .perform(click()));

    // type something
    onView(
       (withId(R.id.post_title))
        .perform(typeText("GrbcVK9I")));

    // tap on the publish button
    onView(
       (withId(R.id.menu_save_post))
        .perform(click()));
}
```
Motivating Example

**Bug Report**

Onko commented on Mar 24, 2017

- Text paragraph
- Logcat information

**Steps to reproduce the behavior**

1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

- Text paragraph

**Test Case**

```java
public void test() {
    // start a new post
    onView(
        withId(R.id.fab_button))
        .perform(click());

    // type something
    onView(
        withId(R.id.post_title))
        .perform(typeText("GrbcVK9I"));

    // tap on the publish button
    onView(
        withId(R.id.menu_save_post))
        .perform(click());
}
```
Ontology Extraction

Bug Report Analysis

UI Actions Search

Relevant App

Ontology

Bug Report

Abstract Steps

Test Case
Ontology Extraction

Bug Report Analysis

UI Actions Search

Ontology

Bug Report

Abstract Steps

Test Device

Test Case
Ontology Extraction → Ontology → Bug Report Analysis → Abstract Steps

Relevant App → Ontology Extraction → Bug Report → UI Actions Search → Test Case

Test Case
YAKUSU takes only a few minutes to process a bug report
App Evolution

Lollipop
version 22

Marshmallow
version 23

API-Usage Update

public NetworkInfo[] getAllNetworkInfo ()

⚠️ This method was deprecated in API level 23.
This method does not support multiple connected networks of the same type. Use `getAllNetworks()` and `getNetworkInfo(android.net.Network)` instead.
App Evolution

Lollipop
version 22

Marshmallow
version 23

API-Usage Update

```java
public NetworkInfo[] getAllNetworkInfo ()
```

⚠️ This method was deprecated in API level 23.
This method does not support multiple connected networks of the same type. Use `getAllNetworks()` and `getNetworkInfo(android.net.Network)` instead.
Motivating Example

Update Example

Old Version

```java
1 public boolean isNetworkAvailable(Context ctx) {
2     ConnectivityManager ctv = (ConnectivityManager)
3         ctx.getSystemService(Context.CONNECTIVITY_SERVICE);
4     NetworkInfo[] info = ctv.getAllNetworkInfo();
5     if (info != null) {
6         for (int i = 0; i < info.length; i++) {
7             if (info[i].getState() ==
8                 NetworkInfo.State.CONNECTED) {
9                 return true;
10             }
11         }
12     } return false;
13 }
```

New Version

```java
1 public boolean isNetworkAvailable(Context ctx) {
2     ConnectivityManager ctv = (ConnectivityManager)
3         ctx.getSystemService(Context.CONNECTIVITY_SERVICE);
4     if (Build.VERSION.SDK_INT >=
5         Build.VERSION_CODES.M) {
6         Network[] networks = ctv.getAllNetworks();
7         NetworkInfo networkInfo;
8         for (Network mNetwork : networks) {
9             networkInfo = ctv.getNetworkInfo(mNetwork);
10             if (networkInfo.getState().equals(
11                 NetworkInfo.State.CONNECTED)) {
12                 return true;
13             }
14         }
15         else {
16             NetworkInfo[] info = ctv.getAllNetworkInfo();
17             if (info != null) {
18                 ... return false;
19             }
20         }
21     } return false;
22 }
```
Motivating Example

Update Example

Old Version

1 public boolean isNetworkAvailable(Context ctx) {  
2   ConnectivityManager ctv = (ConnectivityManager)  
3   ctx.getSystemService(Context.CONNECTIVITY_SERVICE);  
4 -   NetworkInfo[] info = ctv.getAllNetworkInfo();  
5 -   if (info != null) {  
6 -     for (int i = 0; i < info.length; i++) {  
7 -       if (info[i].getState() == NetworkInfo.State.CONNECTED) {  
8 -         return true;  
9 -       }  
10    }  
11   }  
12   return false;  
13 }

New Version

1 public boolean isNetworkAvailable(Context ctx) {  
2   ConnectivityManager ctv = (ConnectivityManager)  
3   ctx.getSystemService(Context.CONNECTIVITY_SERVICE);  
4 +   if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
5 +     Network[] networks = ctv.getAllNetworks();  
6 +     NetworkInfo networkInfo;  
7 +     for (Network mNetwork : networks) {  
8 +       networkInfo = ctv.getNetworkInfo(mNetwork);  
9 +       if (networkInfo.getState().equals(NetworkInfo.State.CONNECTED)) {  
10 +         return true;  
11 +       }  
12 +     }  
13 +   }  
14 + } else {  
15 +   NetworkInfo[] info = ctv.getAllNetworkInfo();  
16 +   if (info != null) {  
17 +     ...  
18    }  
19 +   return false;  
20 + }
21 + return false;  
22 + }
23 + return false;  
24 + }
APP EVOLVE

API-Usage Analysis

API-Usage Change Specification

Target App

App

Evolved Target App

App

API-Usage Update

API-Usage Change Specification

API-Usage Report

Search

Update Example Search

Update Example Analysis

API-Usage Report

Generic Update Patches

Update Examples

API-Usage Update Report

Generic Update Patches

Update Analysis
APP-EVOLVE

API-Usage Change Specification

API-Usage Analysis

API-Usage Change Specification

Update Example Search

API-Usage Report

Update Examples

Target App

App

Evolved Target App

API-Usage Update

Generic Update Patches

API-Usage Update Report

API-Usage Update Analysis

Update Example Analysis
API usages successfully updated

85%

Cost of running APP EVOLVE

APP EVOLVE runs overnight
Research Directions

Testing
- Fuzzing based on recorded tests
- Differential testing for detecting API inconsistencies

Maintenance
- Enhanced text-based bug reporting
- Automatically fixing inconsistencies
Research Directions

Testing

- Differential testing for detecting API inconsistencies
- Fuzzing based on recorded tests

Maintenance

- Enhanced text-based bug reporting
- Automatically fixing inconsistencies
Research Directions

Testing

- Fuzzing based on recorded tests
- Differential testing for detecting API inconsistencies

Maintenance

- Enhanced text-based bug reporting
- Automatically fixing inconsistencies
Research Directions

**Testing**
- Fuzzing based on recorded tests
- Differential testing for detecting API inconsistencies

**Maintenance**
- Enhanced text-based bug reporting
- Automatically fixing inconsistencies
Research Directions

Testing
- Fuzzing based on recorded tests
- Differential testing for detecting API inconsistencies

Maintenance
- Enhanced text-based bug reporting
- Automatically fixing inconsistencies
Research Directions

Testing

- Fuzzing based on recorded tests
- Differential testing for detecting API inconsistencies

Maintenance

- Enhanced text-based bug reporting
- Automatically fixing inconsistencies
Plan

Today

Present at Doctoral Symposium

Nov 2018

Ph.D. Proposal

2019

Complete work on differential API testing
Complete work on enhancing text-based bug reporting

Summer 2019

Ph.D. Defense
Plan

Today

Present at Doctoral Symposium

Nov 2018

Ph.D. Proposal

2019

Complete work on differential API testing
Complete work on enhancing text-based bug reporting

Summer 2019

Ph.D. Defense
Plan

Today
Present at Doctoral Symposium

Nov 2018
Ph.D. Proposal

2019
Complete work on differential API testing
Complete work on enhancing text-based bug reporting

Summer 2019
Ph.D. Defense
Plan

Today

Present at Doctoral Symposium

Nov 2018

Ph.D. Proposal

2019

Complete work on differential API testing
Complete work on enhancing text-based bug reporting

Summer 2019

Ph.D. Defense
Summary

Goals

- Find more bugs early with novel testing techniques
- Quickly resolve bugs with novel maintenance techniques

Research Outline

- Test Generation
  - BARISTA
  - ICST’17
- GUI Inconsistency Identification
  - DIFFDROID
  - ASE’17
- Bug Report Analysis
  - YAKUSU
  - ISSTA’18

- API-Usage Update
  - APP EVOLVE
  - Work in Progress
- Enhanced Bug Reporting
  - REPORT ASSIST
  - Work in Progress
- API Inconsistency Identification
  - DIFFDROID API
  - Work in Progress

Research Directions

Testing
- Fuzzing based on recorded tests
- Differential testing for API inconsistencies

Maintenance
- Enhanced text-based reporting
- Automatically fixing inconsistencies

Plan

Today
- Present at Doctoral Symposium

Nov 2018
- Ph.D. Proposal

2019
- Complete work on enhancing text-based bug reporting
- Complete work on differential API testing

Summer 2019
- Ph.D. Defense
Summary

**Goals**
- Find more bugs early with novel testing techniques
- Quickly resolve bugs with novel maintenance techniques

**Research Outline**

<table>
<thead>
<tr>
<th>Test Generation</th>
<th>GUI Inconsistency Identification</th>
<th>Bug Report Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>BARISTA ICST’17</td>
<td>DIFFDROID ASE’17</td>
<td>YAKUSU ISSTA’18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security Research</th>
<th>API Usage Update</th>
<th>Enhanced Bug Reporting</th>
<th>API Inconsistency Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISSTA’12 ICSE’15</td>
<td>APP EVOLVE Work in Progress</td>
<td>REPORT ASSIST Work in Progress</td>
<td>DIFFDROID API Work in Progress</td>
</tr>
</tbody>
</table>

**Research Directions**

<table>
<thead>
<tr>
<th>Testing</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzing based on recorded tests</td>
<td>Enhanced text-based reporting</td>
</tr>
<tr>
<td>Differential testing for API inconsistencies</td>
<td>Automatically fixing inconsistencies</td>
</tr>
</tbody>
</table>

**Plan**

- **Today**
  - Present at Doctoral Symposium

- **Nov 2018**
  - Ph.D. Proposal

- **2019**
  - Complete work on enhancing text-based bug reporting
  - Complete work on differential API testing

- **Summer 2019**
  - Ph.D. Defense
Summary

Goals

- Find more bugs early with novel testing techniques
- Quickly resolve bugs with novel maintenance techniques

Research Outline

<table>
<thead>
<tr>
<th>Test Generation</th>
<th>GUI Inconsistency Identification</th>
<th>Bug Report Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>BARISTA ICST’17</td>
<td>DIFFDROID ASE’17</td>
<td>YAKUSU ISSTA’18</td>
</tr>
<tr>
<td>API-Usage Update</td>
<td>Enhanced Bug Reporting</td>
<td>API Inconsistency Identification</td>
</tr>
<tr>
<td>APP EVOLVE Work in Progress</td>
<td>REPORT ASSIST Work in Progress</td>
<td>DIFFDROID API Work in Progress</td>
</tr>
</tbody>
</table>

Research Directions

Testing
- Fuzzing based on recorded tests
- Differential testing for API inconsistencies

Maintenance
- Enhanced text-based reporting
- Automatically fixing inconsistencies

Plan

Today
- Present at Doctoral Symposium

Nov 2018
- Ph.D. Proposal

2019
- Complete work on enhancing text-based bug reporting
- Complete work on differential API testing

Summer 2019
- Ph.D. Defense
Summary

Goals

- Find more bugs early with novel testing techniques
- Quickly resolve bugs with novel maintenance techniques

Research Outline

- Test Generation
  - BARISTA
  - ICST’17
- GUI Inconsistency Identification
  - DIFFDROID
  - ASE’17
- Bug Report Analysis
  - YAKUSU
  - ISSTA’18

Security Research

- ICCS’17
- CCS’17
- SEC’18

API-Usage Update
- APP EVOLVE
- Work in Progress

Enhanced Bug Reporting
- REPORT ASSIST
- Work in Progress

API Inconsistency Identification
- DIFFDROID API
- Work in Progress

Research Directions

Testing
- Fuzzing based on recorded tests
- Differential testing for API inconsistencies

Maintenance
- Enhanced text-based reporting
- Automatically fixing inconsistencies

Plan

Today
- Present at Doctoral Symposium

Nov 2018
- Ph.D. Proposal

2019
- Complete work on enhancing text-based bug reporting
- Complete work on differential API testing

Summer 2019
- Ph.D. Defense
Summary

Goals

- Find more bugs early with novel testing techniques
- Quickly resolve bugs with novel maintenance techniques

Research Outline

- Test Generation
  - BARISTA
  - ICST'17
- GUI Inconsistency Identification
  - DIFFDROID
  - ASE'17
- Bug Report Analysis
  - YAKUSU
  - ISSTA'18

- API-Usage Update
  - APP EVOLVE
  - Work in Progress
- Enhanced Bug Reporting
  - REPORT ASSIST
  - Work in Progress
- API Inconsistency Identification
  - DIFFDROID API
  - Work in Progress

Research Directions

Testing

- Fuzzing based on recorded tests
- Differential testing for API inconsistencies

Maintenance

- Enhanced text-based reporting
- Automatically fixing inconsistencies

Plan

Today

- Present at Doctoral Symposium

Nov 2018

- Ph.D. Proposal

2019

- Complete work on enhancing text-based bug reporting
- Complete work on differential API testing

Summer 2019

- Ph.D. Defense