

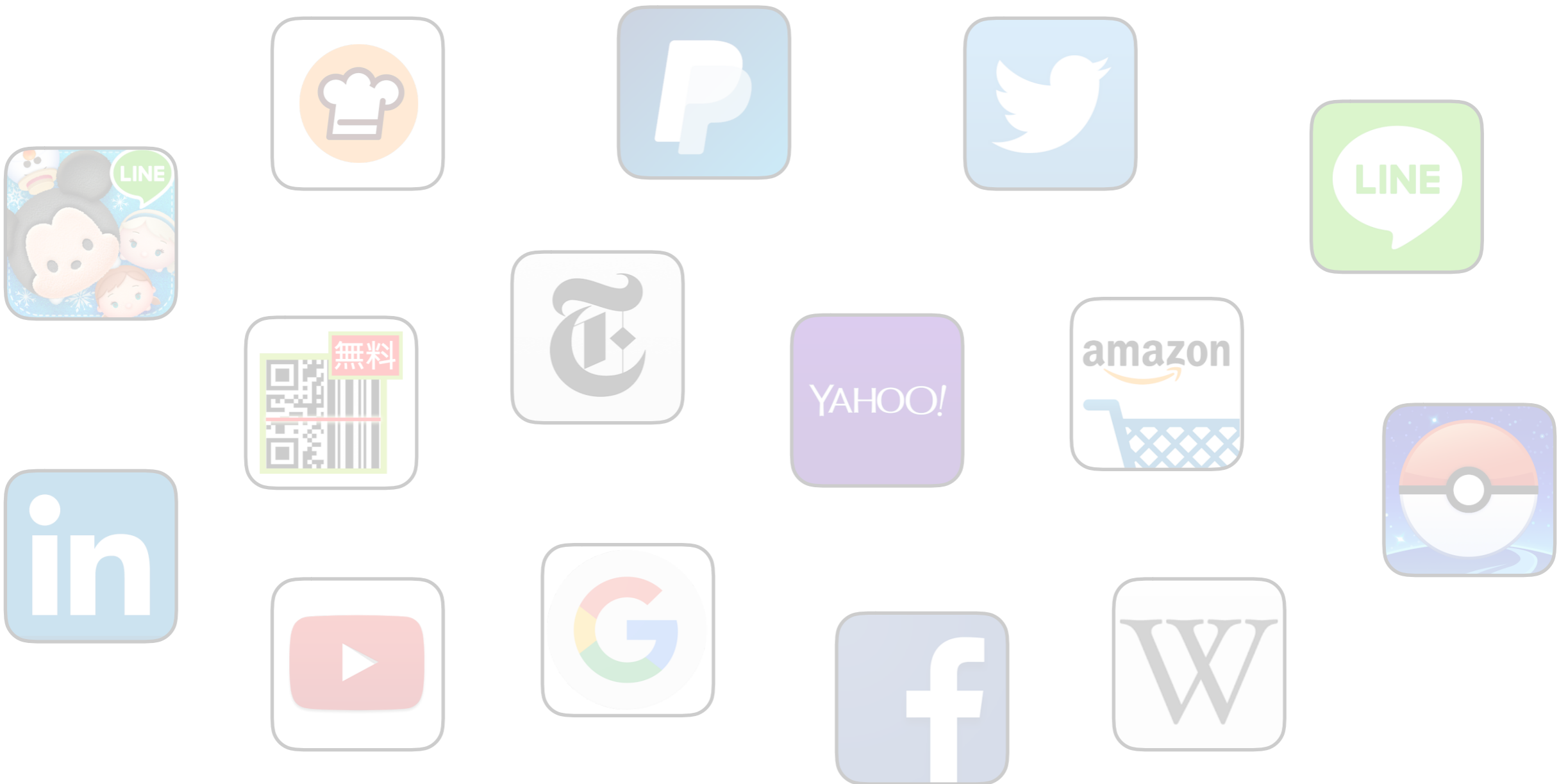
Automated Support for Mobile Application Testing and Maintenance

Mattia Fazzini

Advisor: Dr. Alessandro Orso



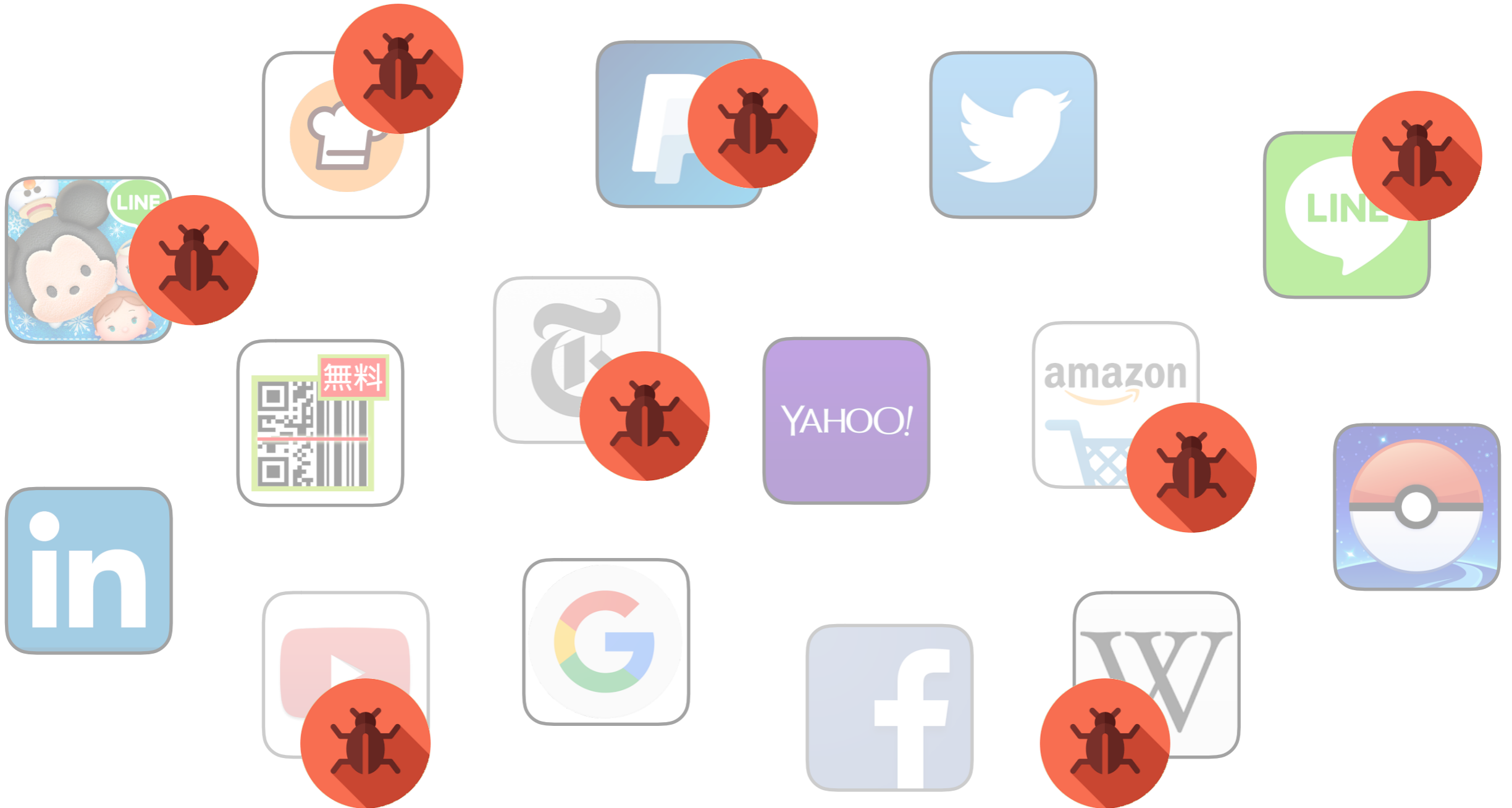
Mobile Applications



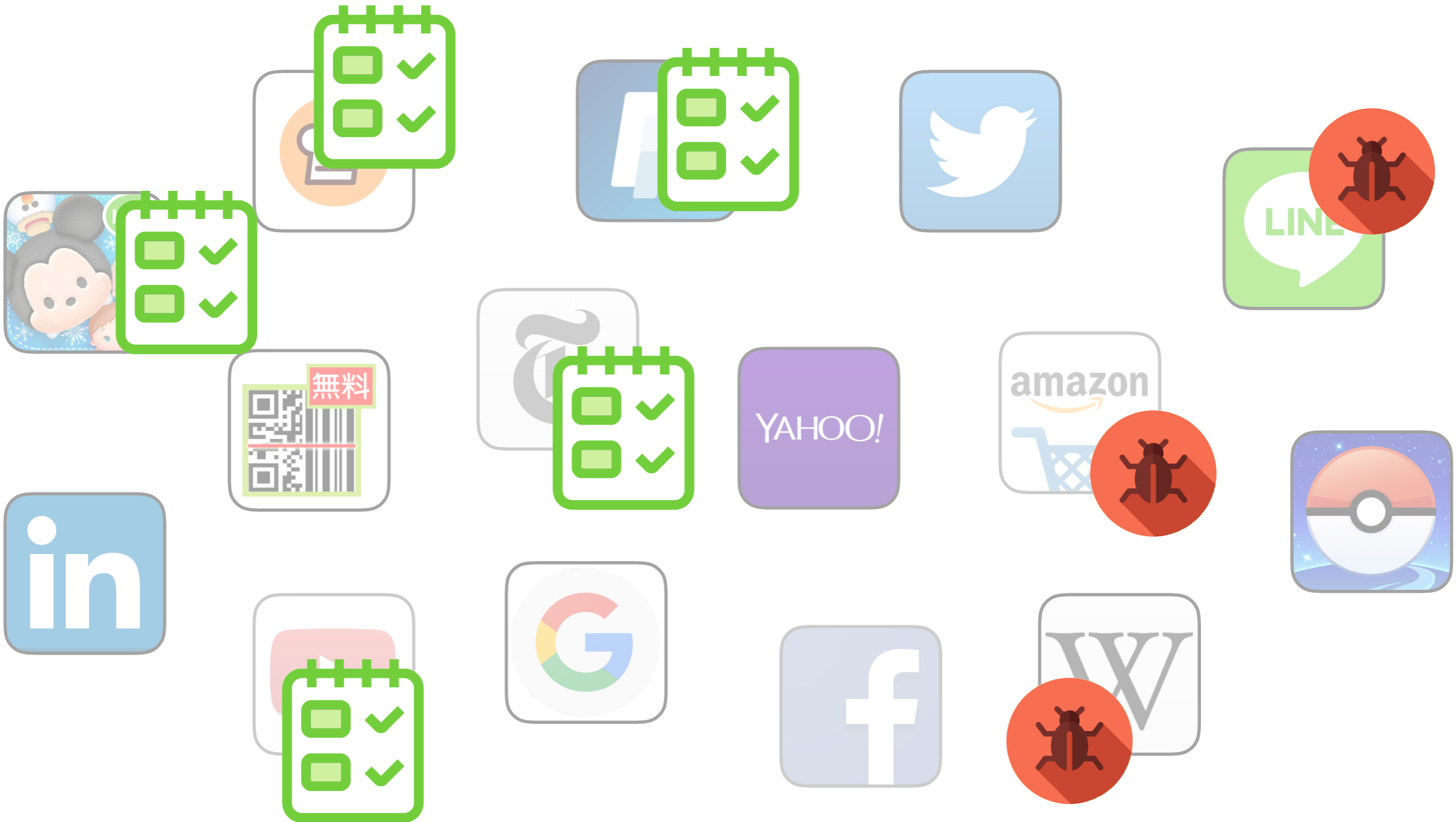
Mobile Applications



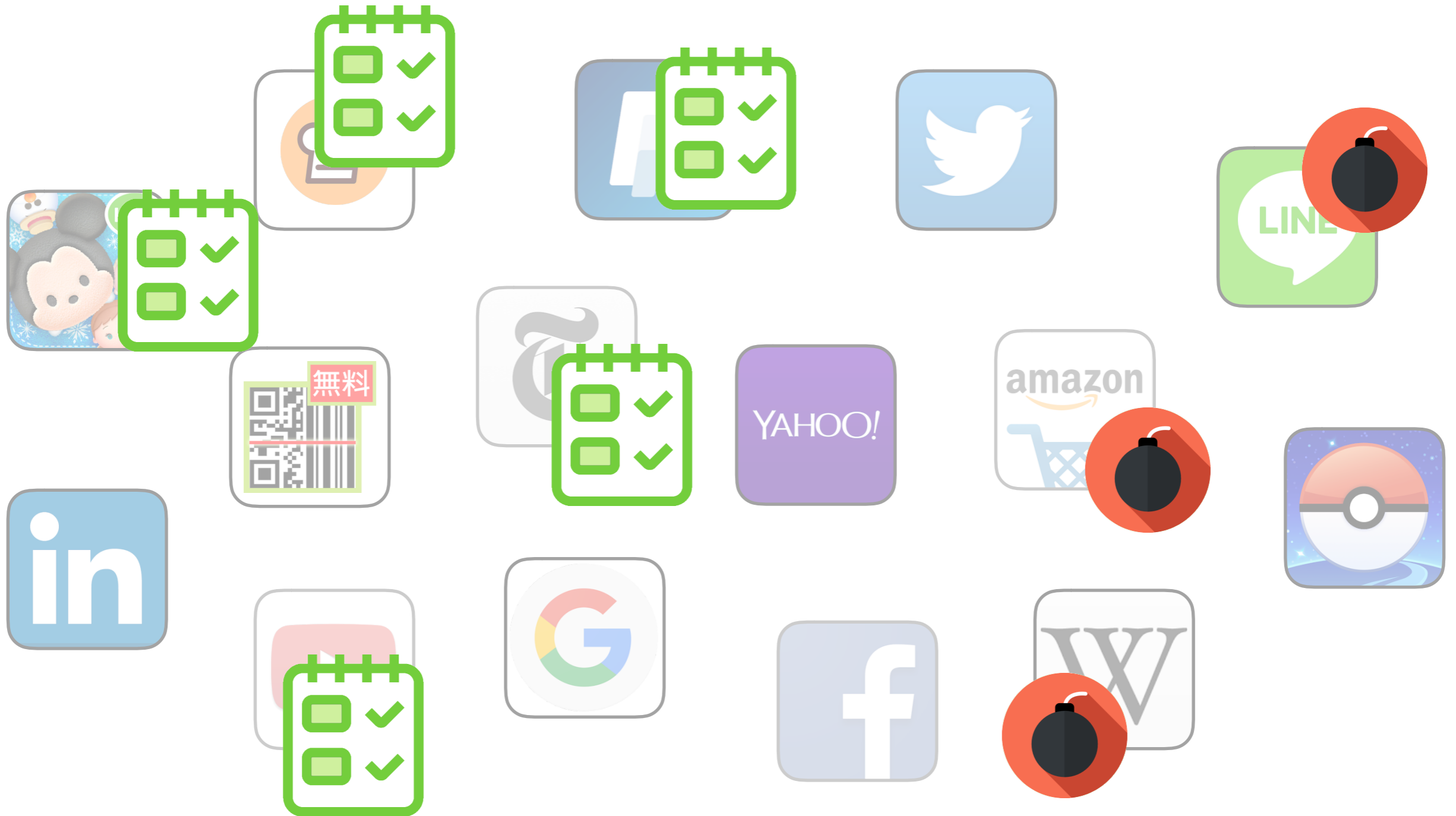
Bugs



Testing



Failures



Maintenance



Goals

Find more bugs early with novel testing techniques

Quickly resolve bugs with novel maintenance techniques

Research Outline

Test Generation



ICST'17

GUI Inconsistency Identification



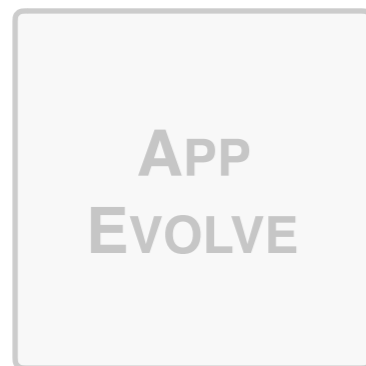
ASE'17

Bug Report Analysis



ISSTA'18

API-Usage Update



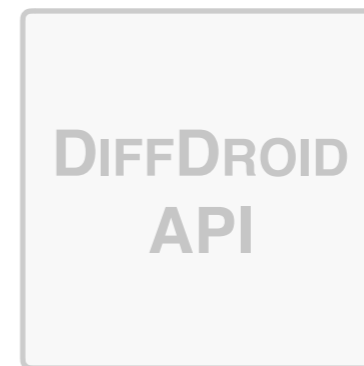
Work in Progress

Enhanced Bug Reporting



Work in Progress

API Inconsistency Identification



Work in Progress

Research Outline

Test Generation



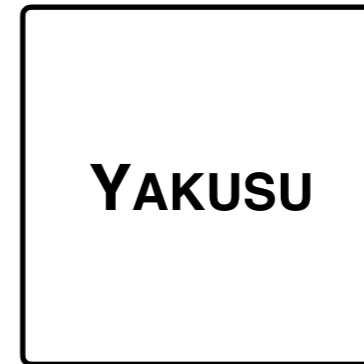
ICST'17

GUI Inconsistency Identification



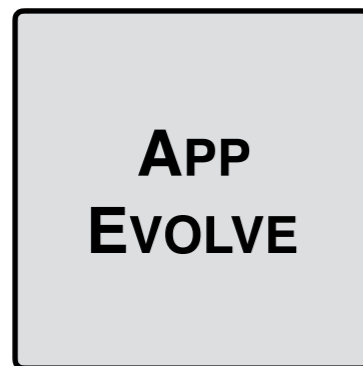
ASE'17

Bug Report Analysis



ISSTA'18

API-Usage Update



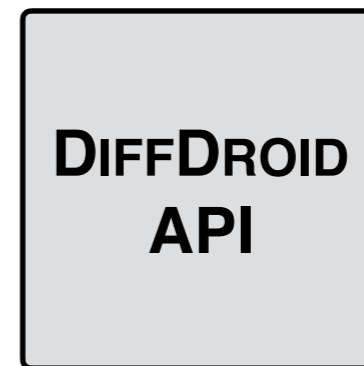
Work in Progress

Enhanced Bug Reporting



Work in Progress

API Inconsistency Identification



Work in Progress

Research Outline

Test Generation



ICST'17

GUI Inconsistency Identification



ASE'17

Bug Report Analysis



ISSTA'18

Security Research

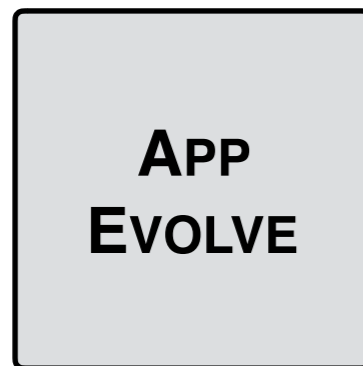
ISSTA'12

ICSE'15

CCS'17

SEC'18

API-Usage Update



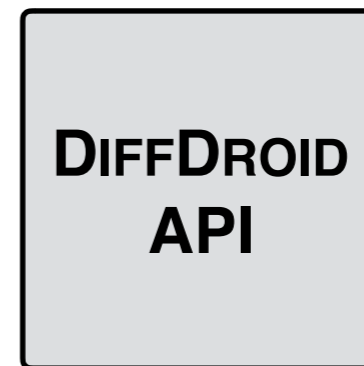
Work in Progress

Enhanced Bug Reporting



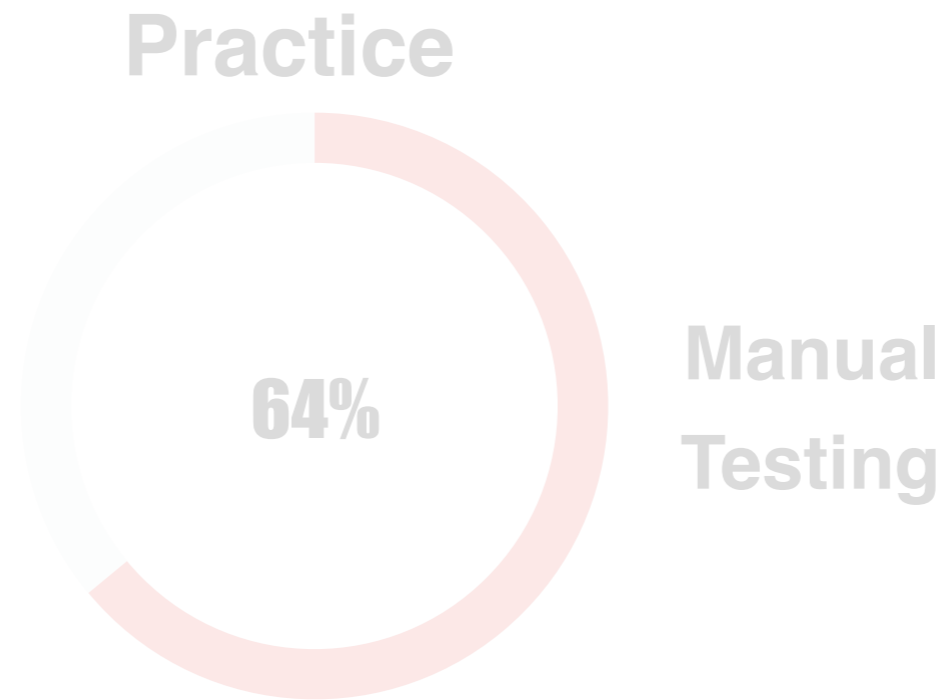
Work in Progress

API Inconsistency Identification



Work in Progress

Mobile App Testing

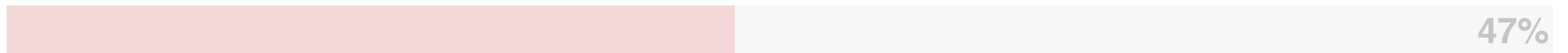


Challenges

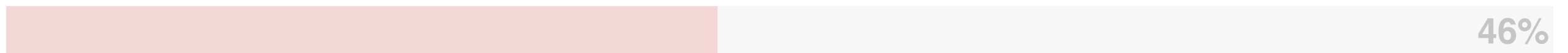
Not enough time to test



Do not have the right testing process/method

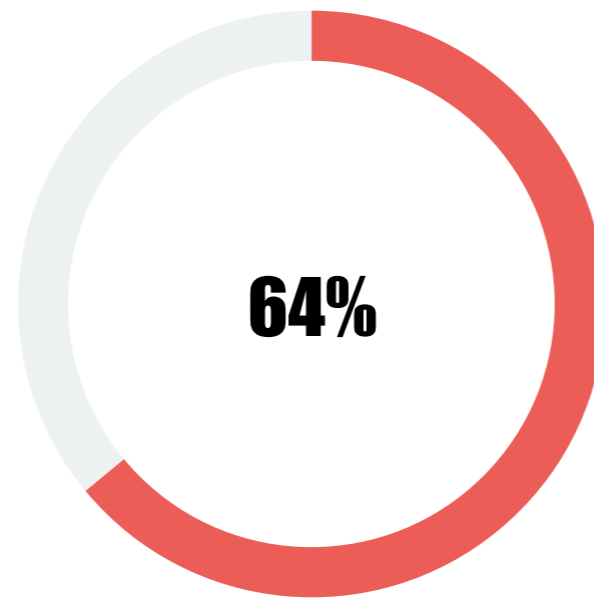


Do not have the right tools to test



Mobile App Testing

Practice



Manual
Testing

Challenges

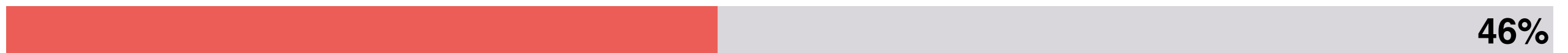
Not enough time to test



Do not have the right testing process/method



Do not have the right tools to test



Motivating Example: Manual Testing

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

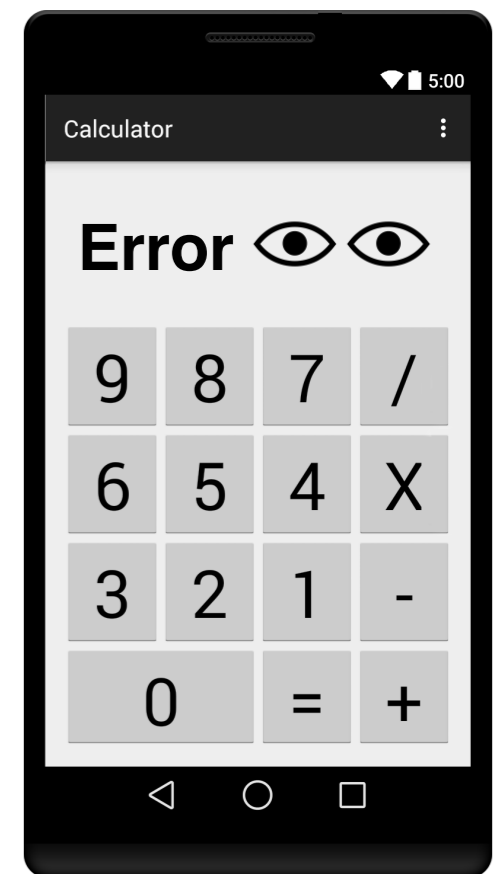
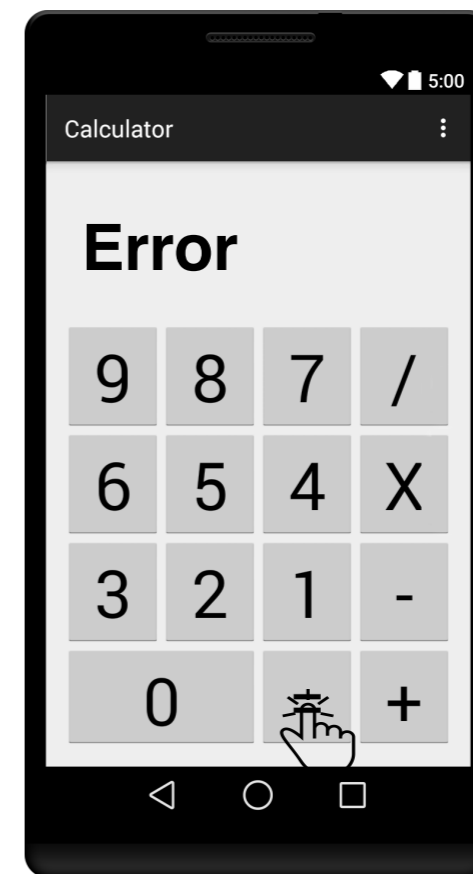
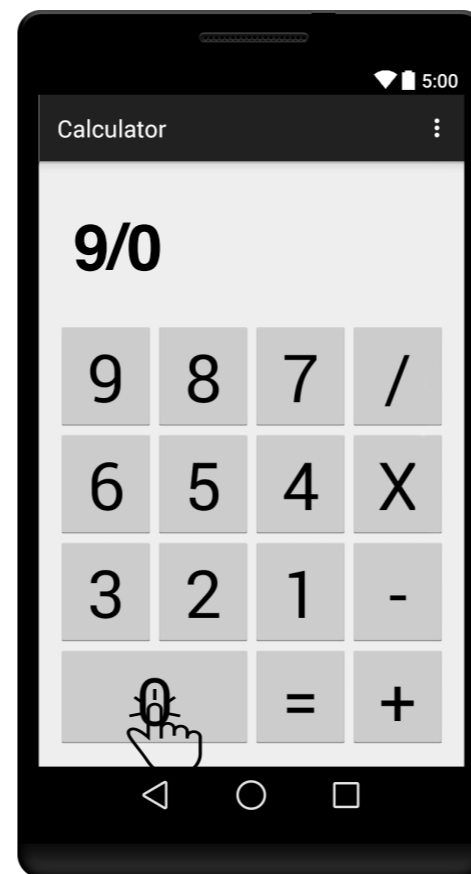
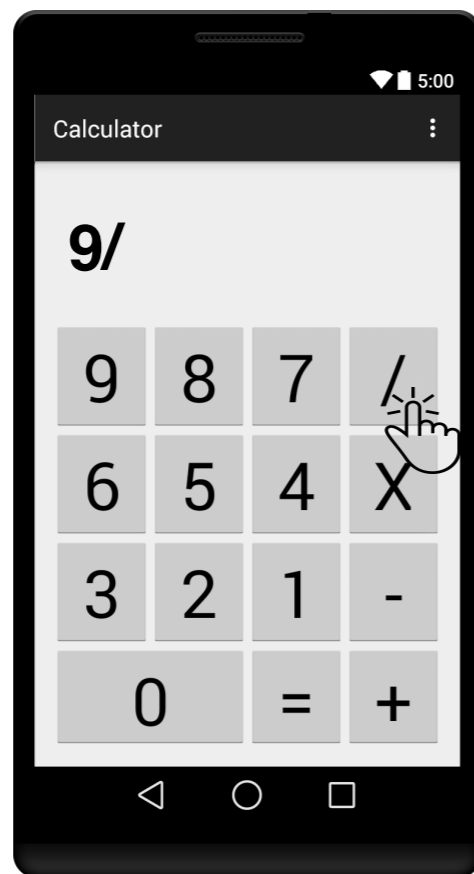
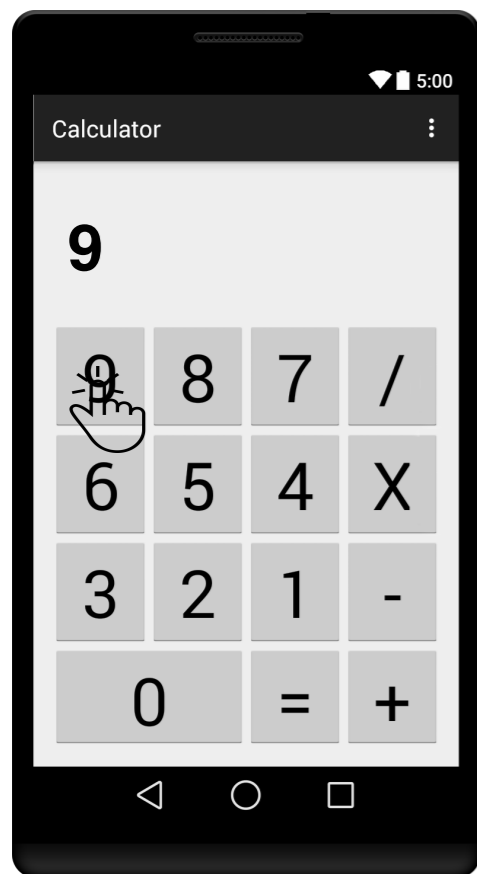
(1) Press nine

(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

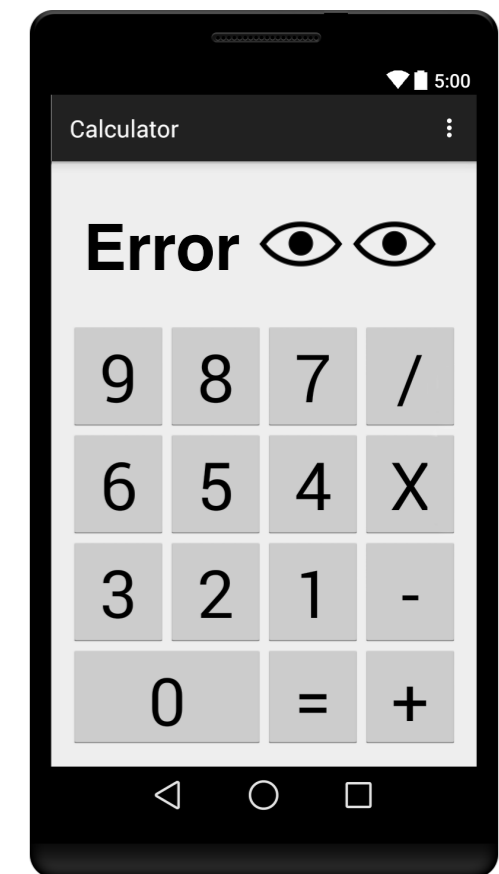
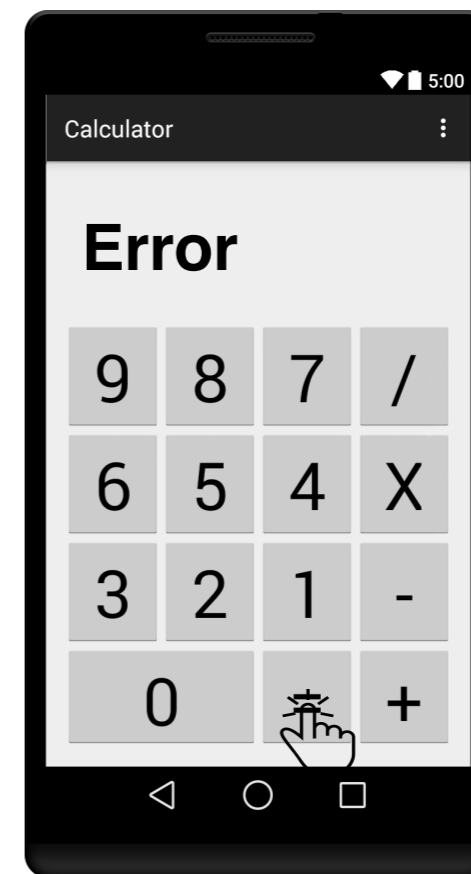
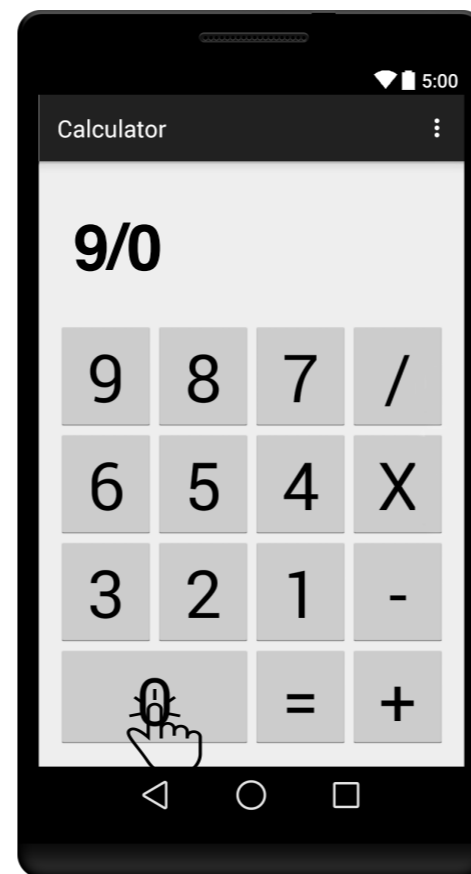
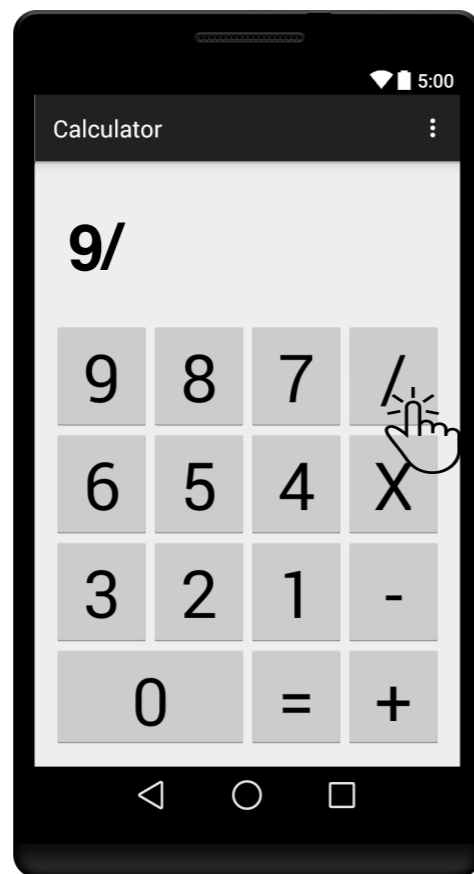
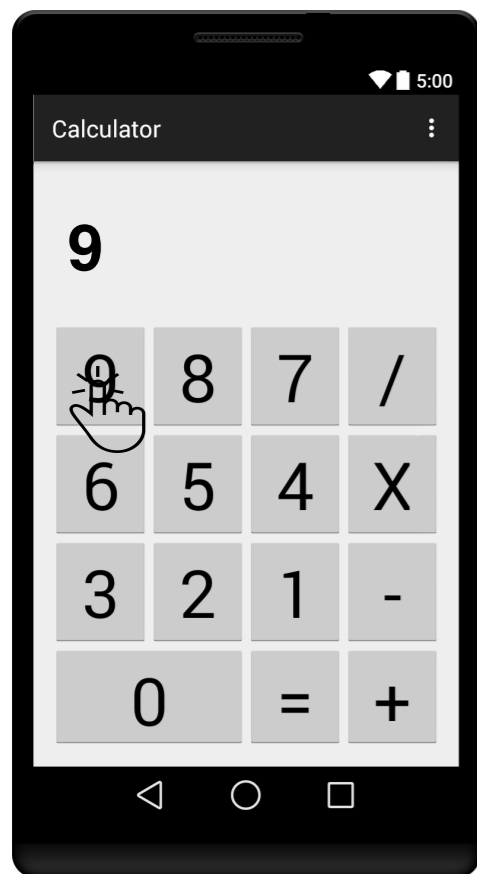
(1) Press nine

(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

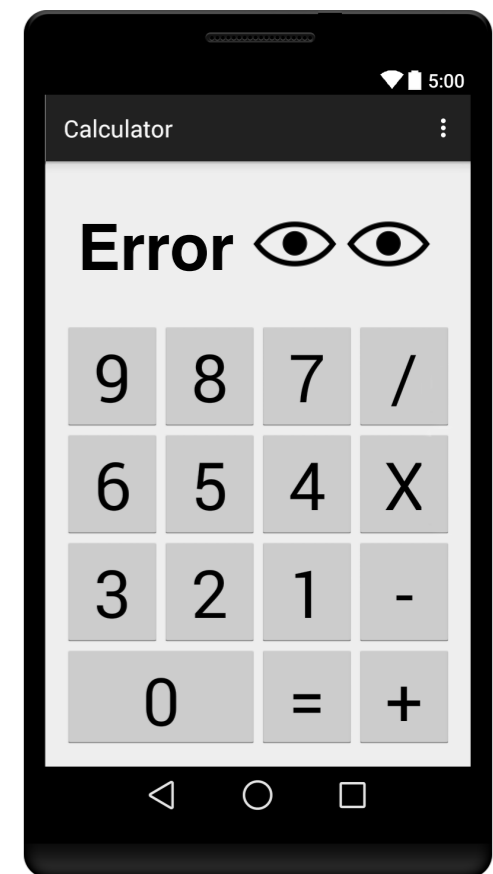
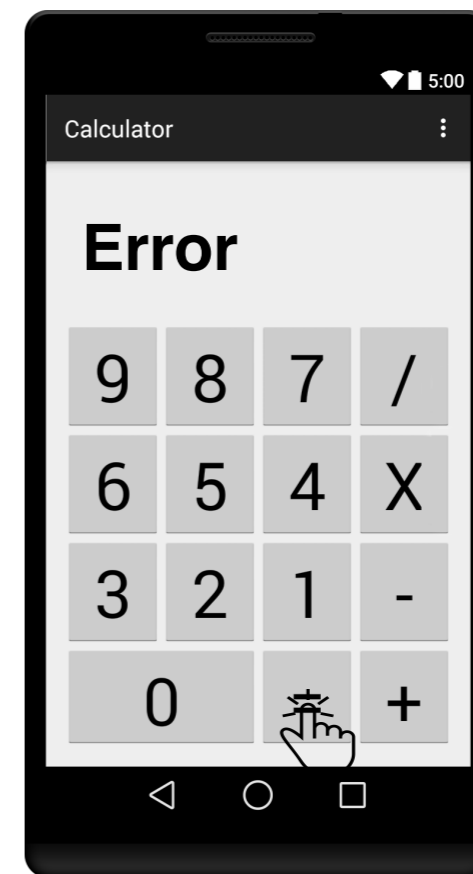
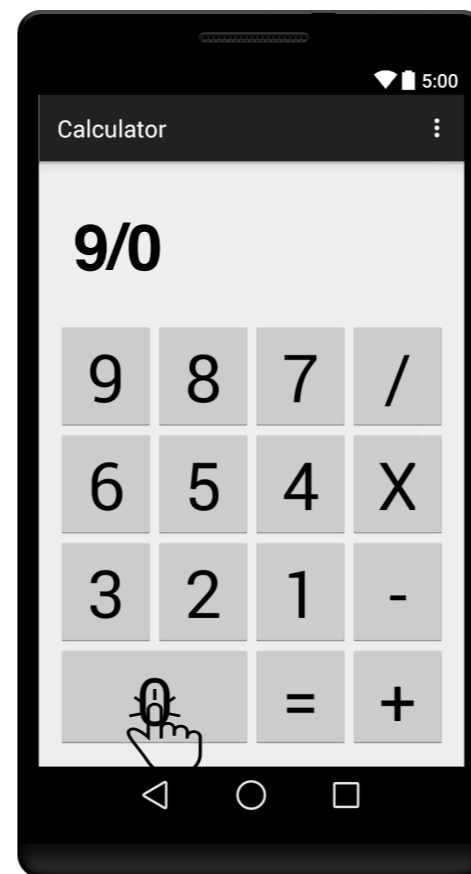
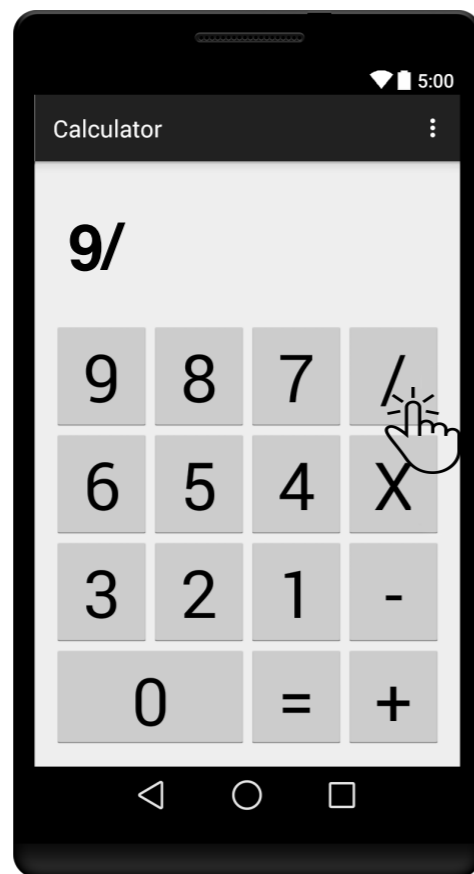
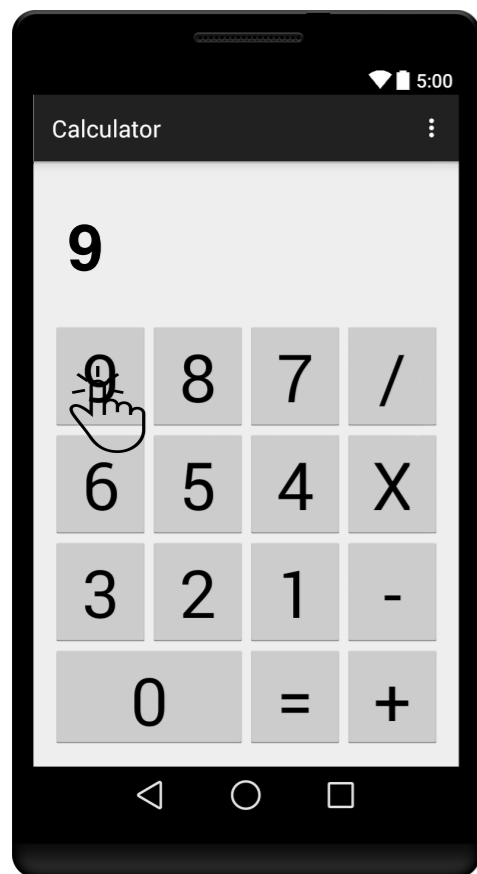
(1) Press nine

(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 }
```

Efficient

Complex

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 }
```

Efficient

Complex

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

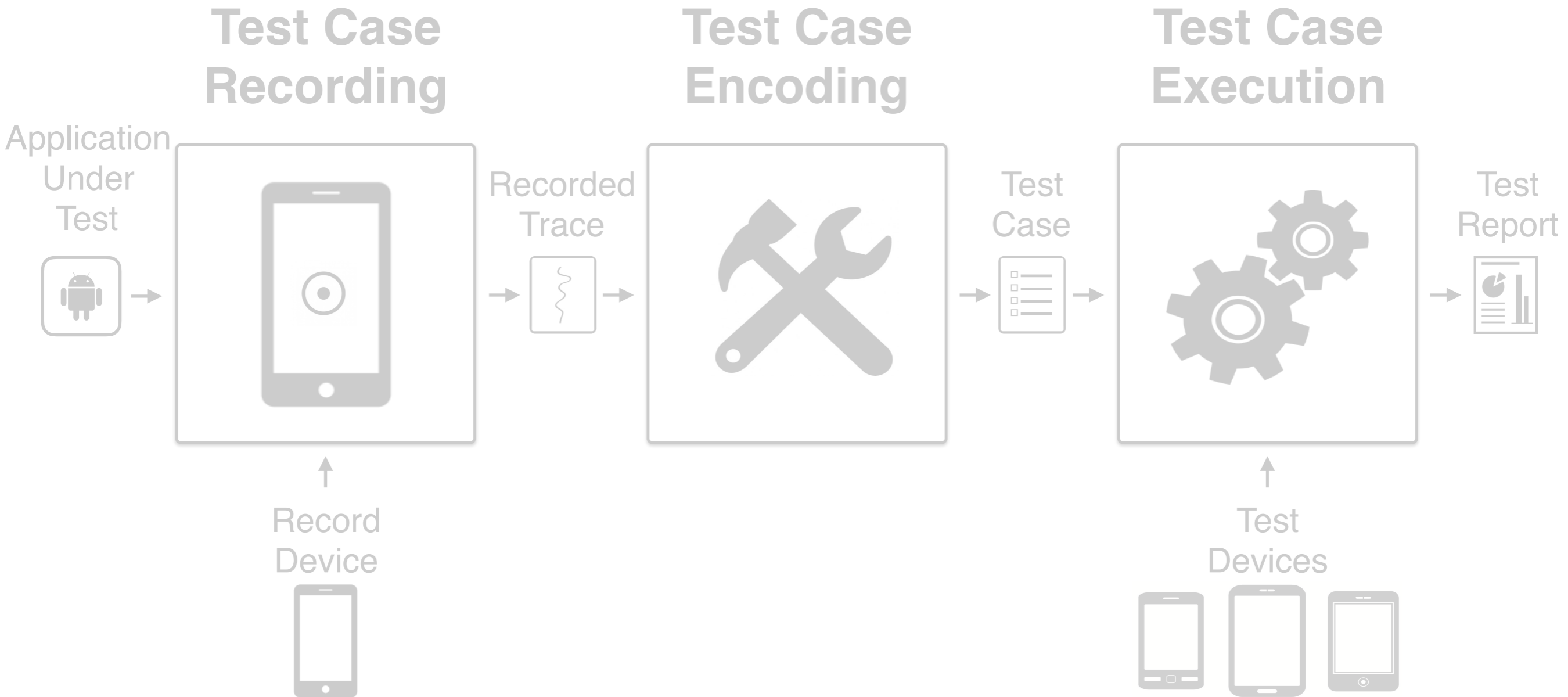
```
1 public void divisionByZero() {  
2     onView(withText("9")).perform(click());  
3     onView(withId(R.id.div)).perform(click());  
4     onView(withText("0")).perform(click());  
5     onView(withId(R.id.eq)).perform(click());  
6     onView(withId(R.id.dsp))  
7     .check(matches(withText("Error")));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

Efficient

Complex

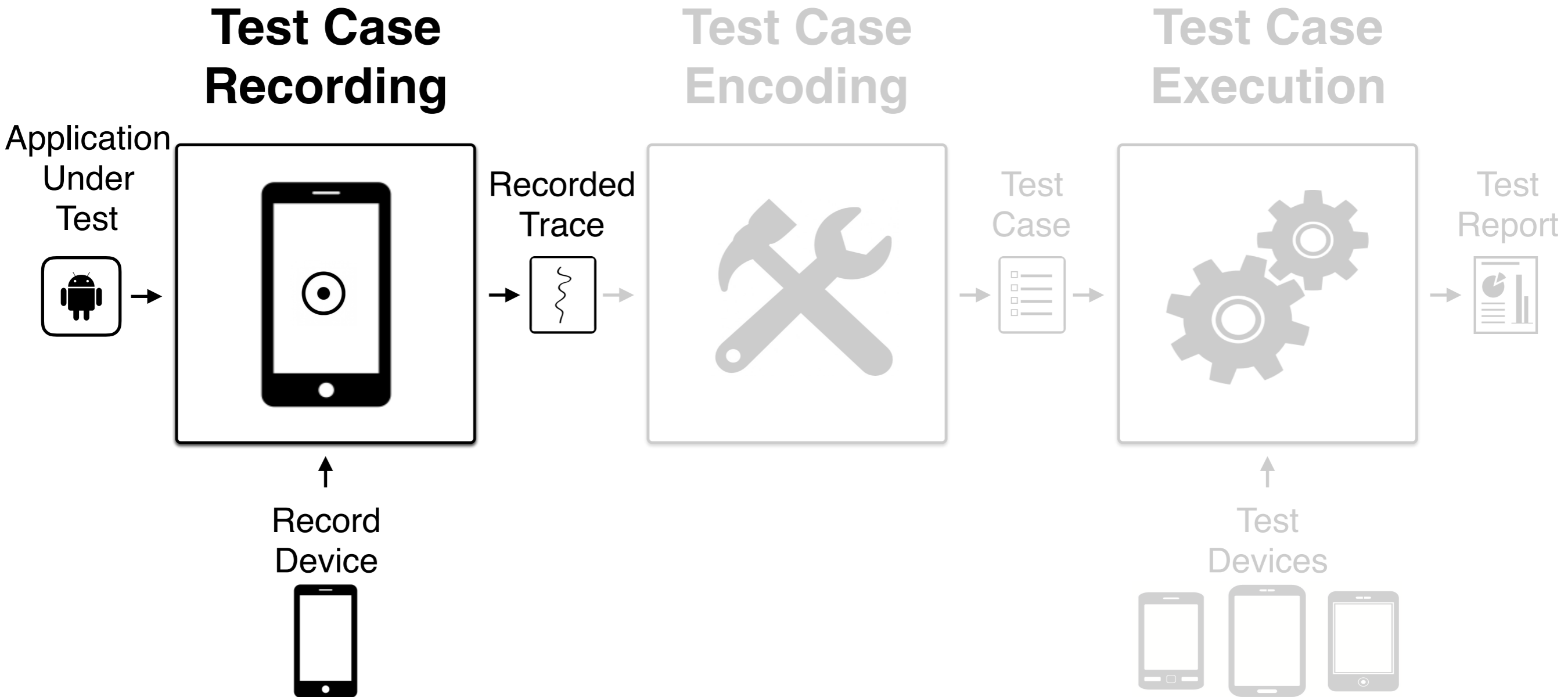
BARISTA²

ICST'17



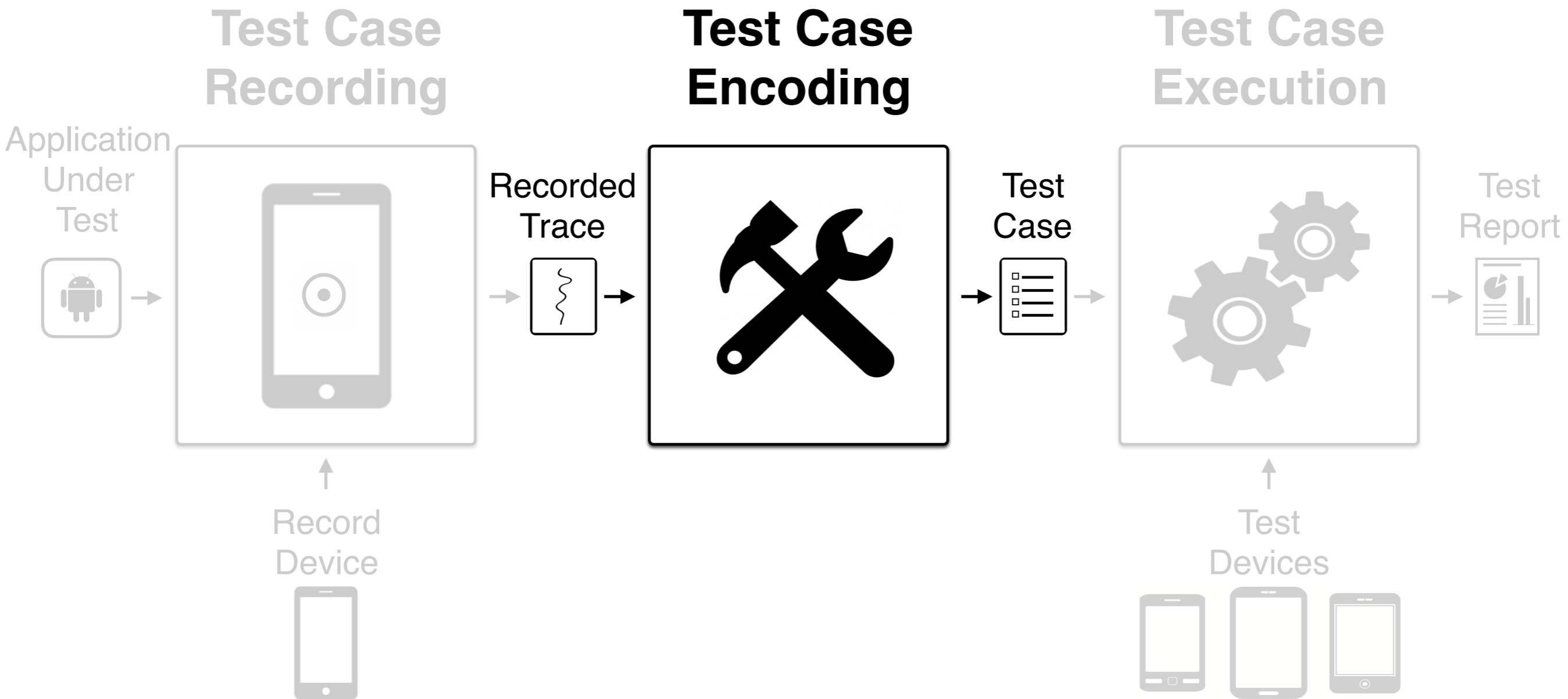
BARISTA²

ICST'17



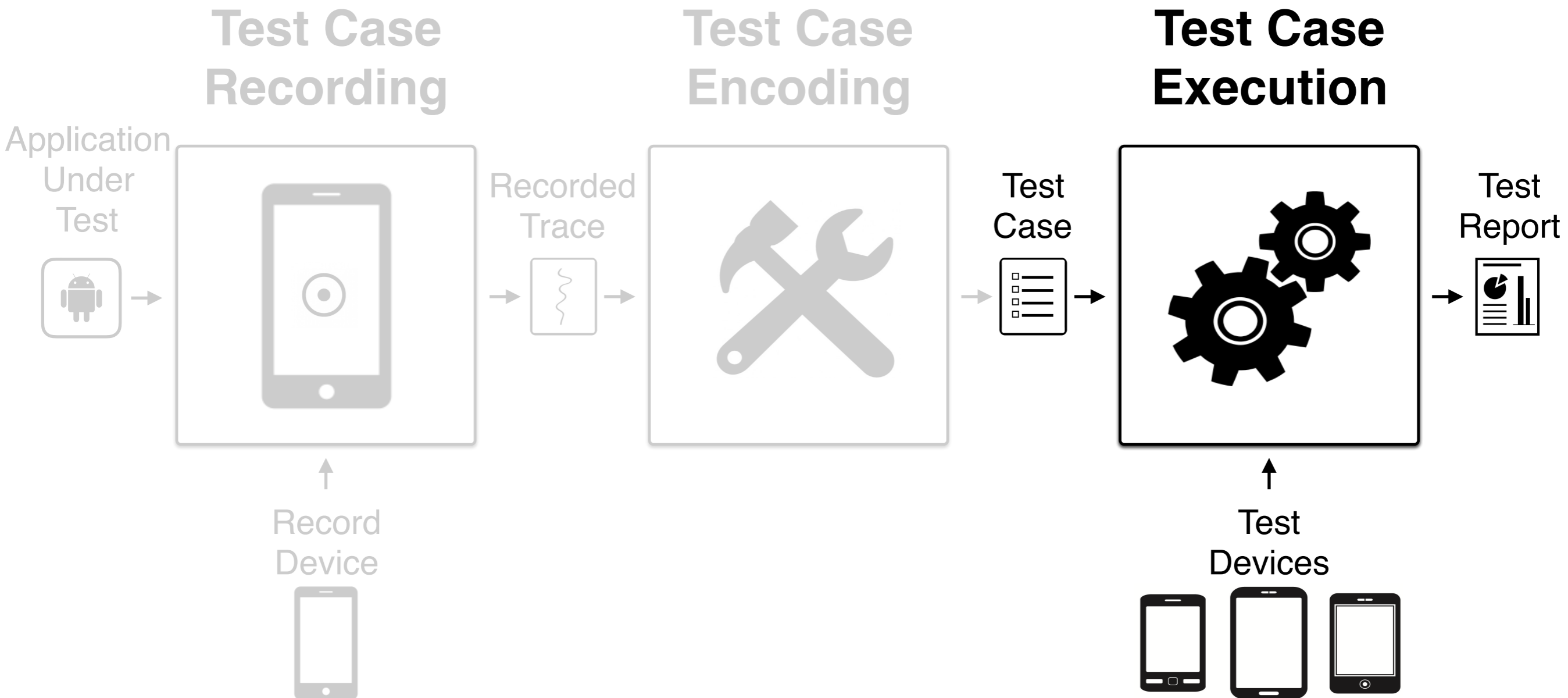
BARISTA²

ICST'17



BARISTA²

ICST'17



BARISTA²

ICST'17

**Test Case
Recording**

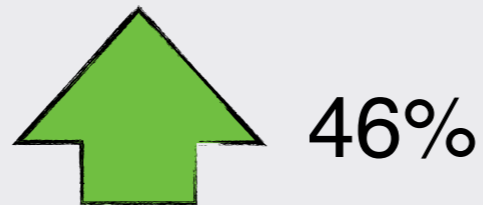
**Test Case
Encoding**

**Test Case
Execution**

Applicat
Unde
Test



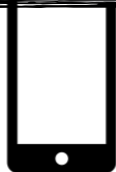
Test cases successfully encoded and executed



Test
Report

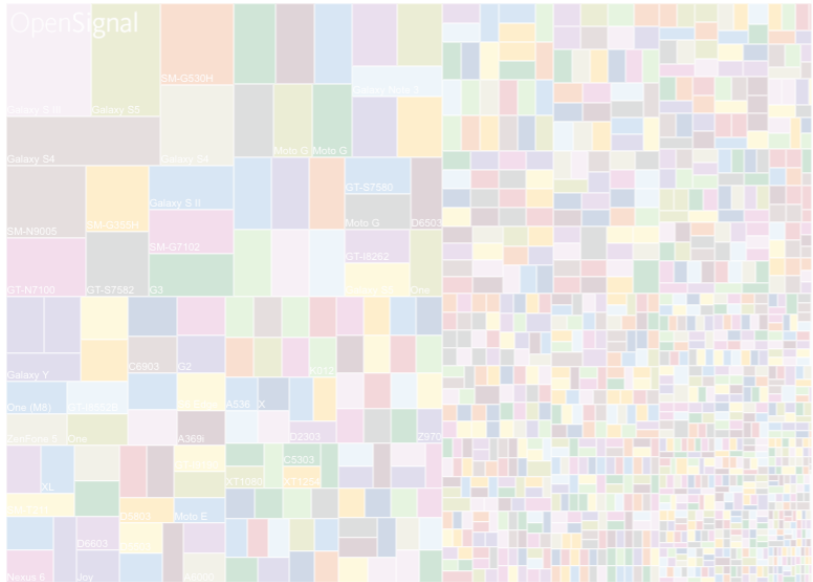


User efficiency in generating test cases



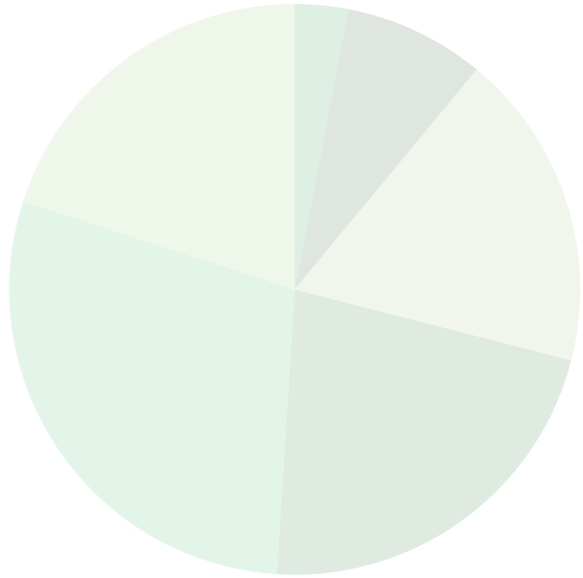
Fragmentation

Device

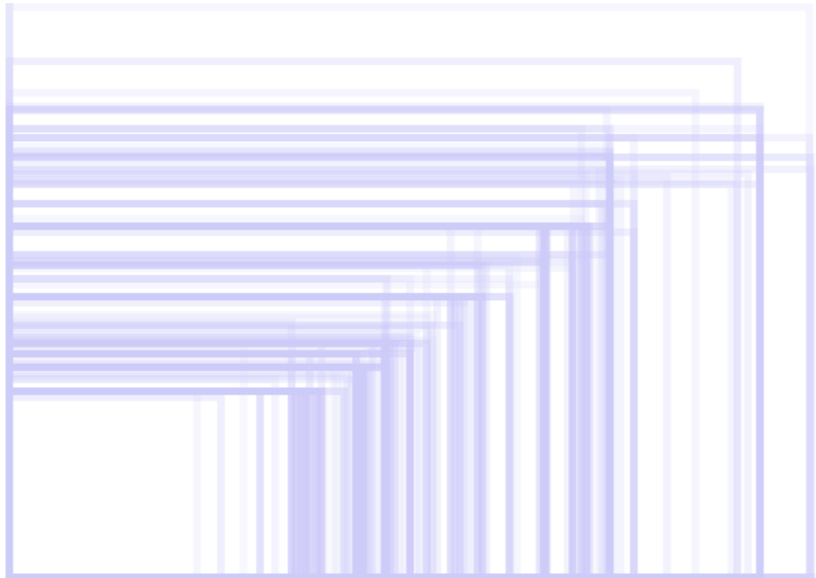


System

- Gingerbread
- IC Sandwich
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo



Screen



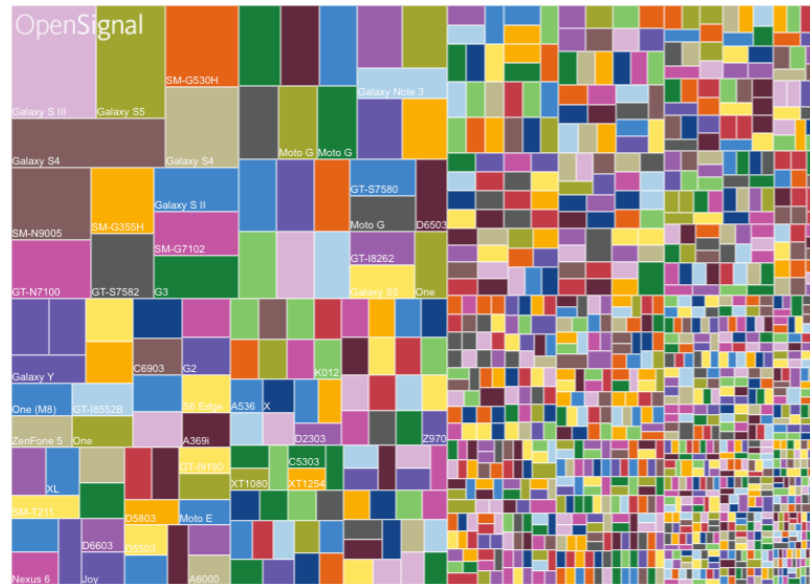
News

Jul 2016

“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”

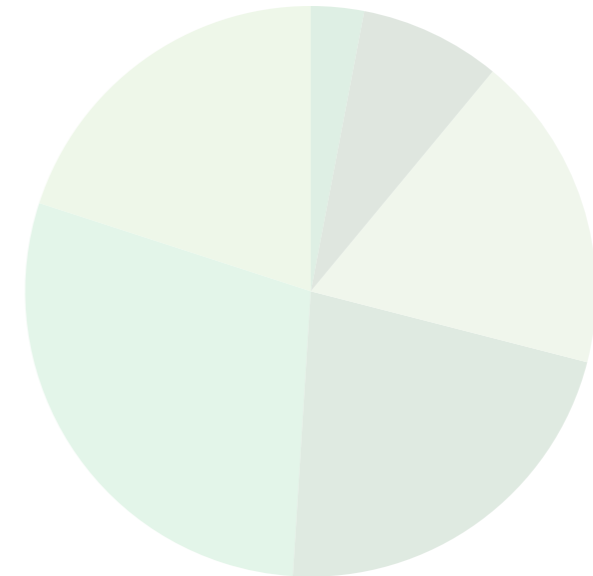
Fragmentation

Device

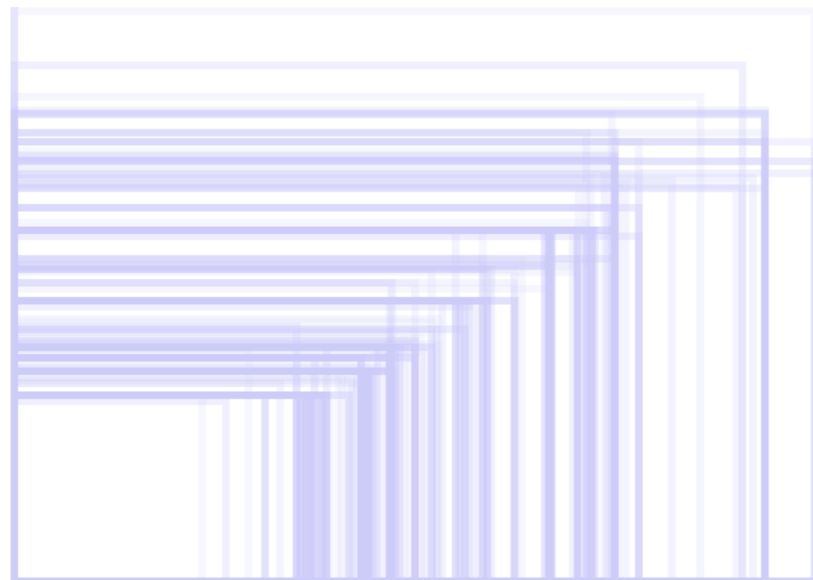


System

- Gingerbread
- IC Sandwich
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo



Screen



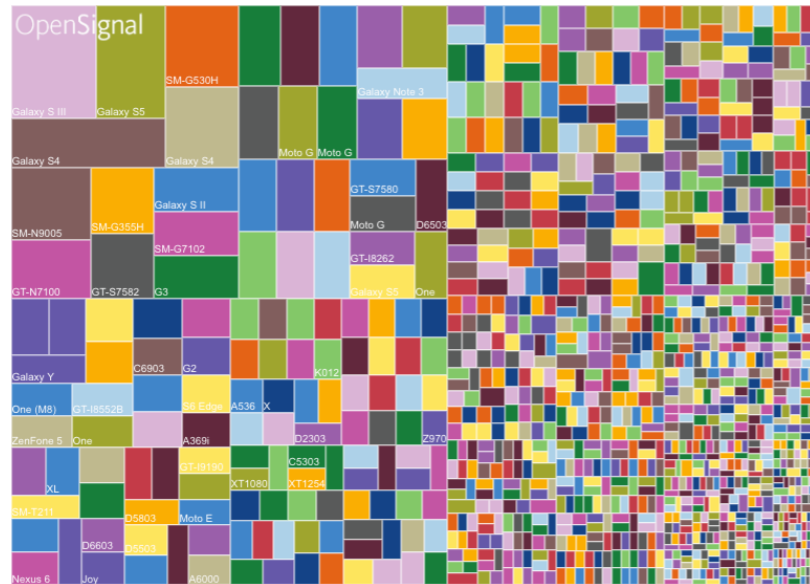
News

Jul 2016

“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”

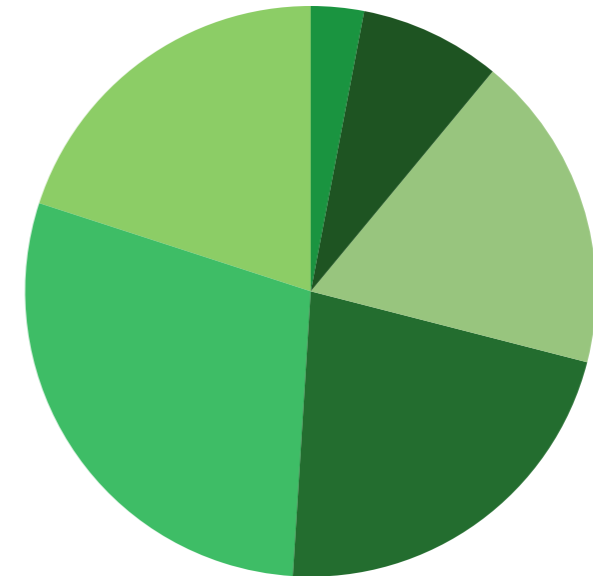
Fragmentation

Device

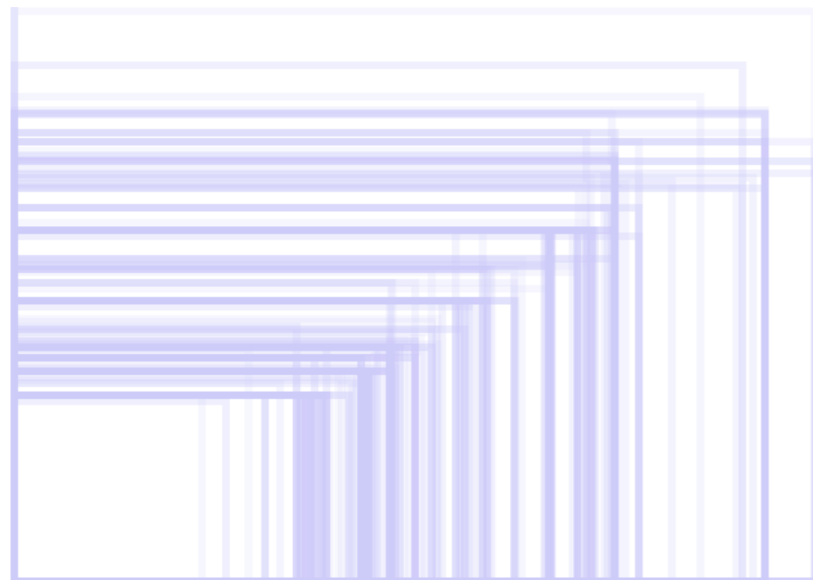


System

- Gingerbread
- IC Sandwich
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo



Screen



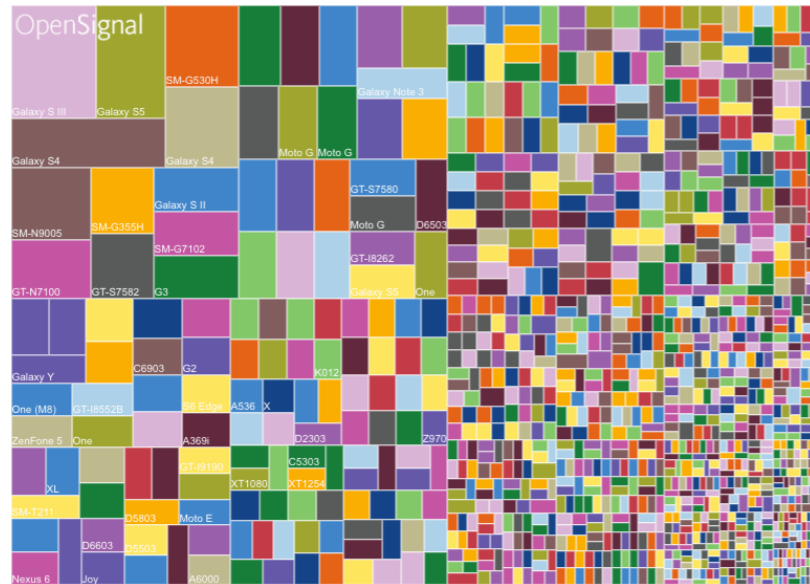
News

Jul 2016

“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”

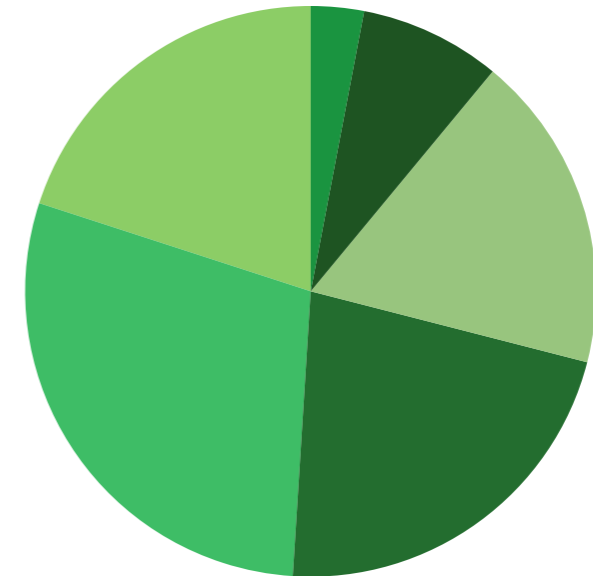
Fragmentation

Device

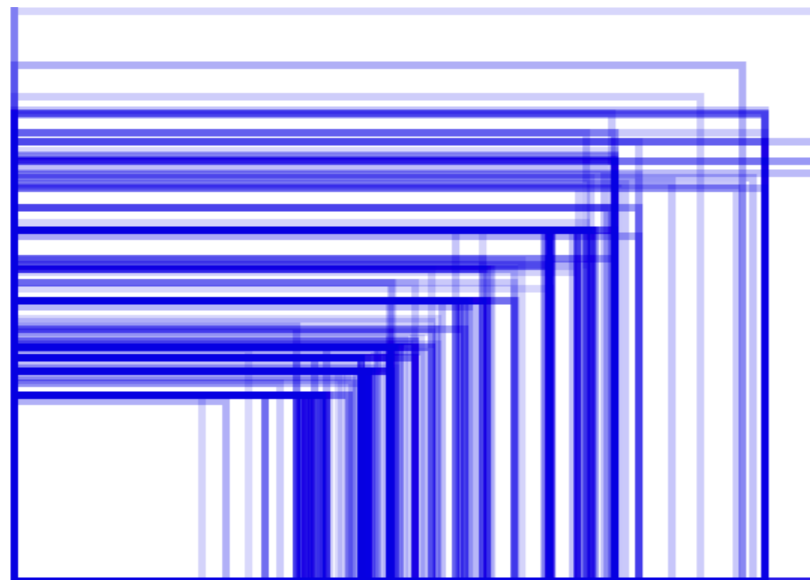


System

- Gingerbread
- IC Sandwich
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo



Screen



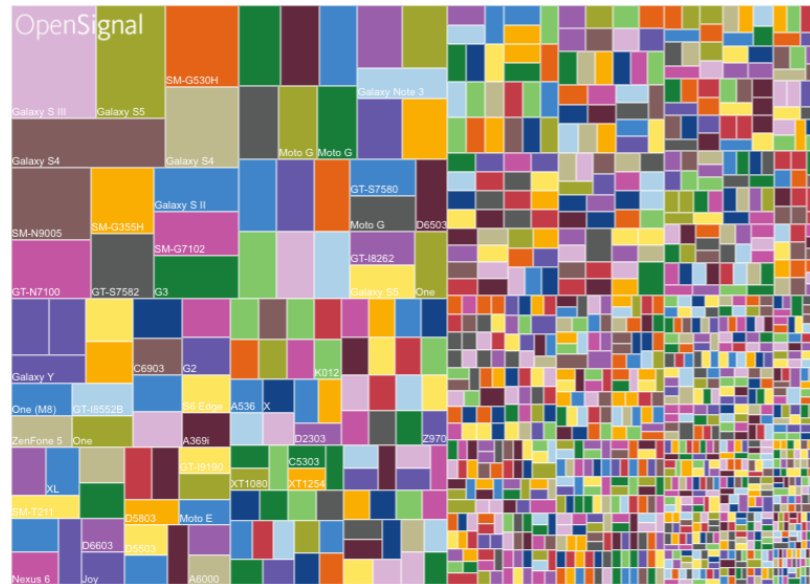
News

Jul 2016

“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”

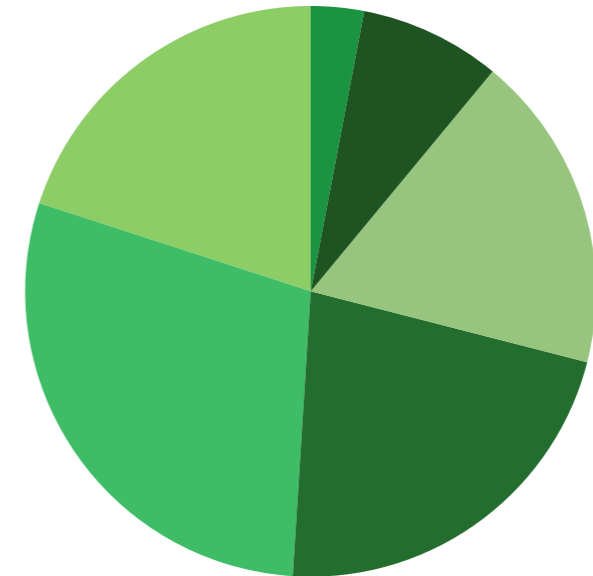
Fragmentation

Device

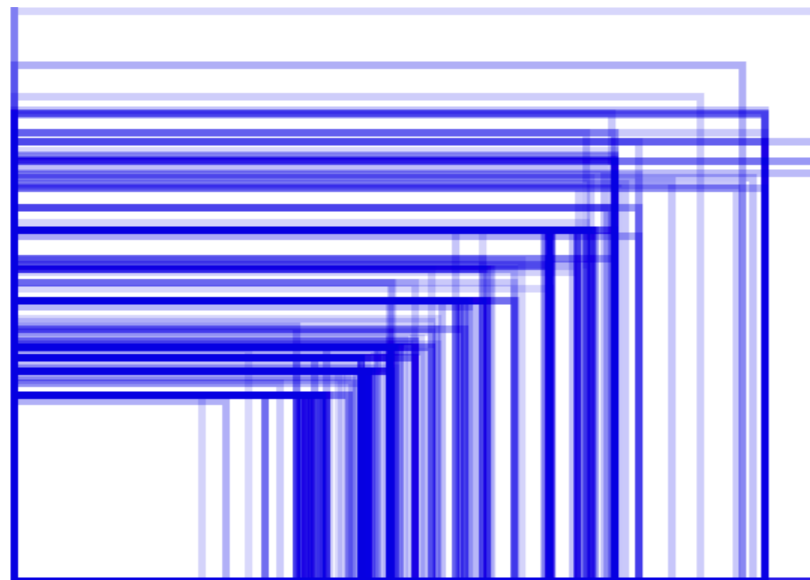


System

- Gingerbread
- IC Sandwich
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo



Screen



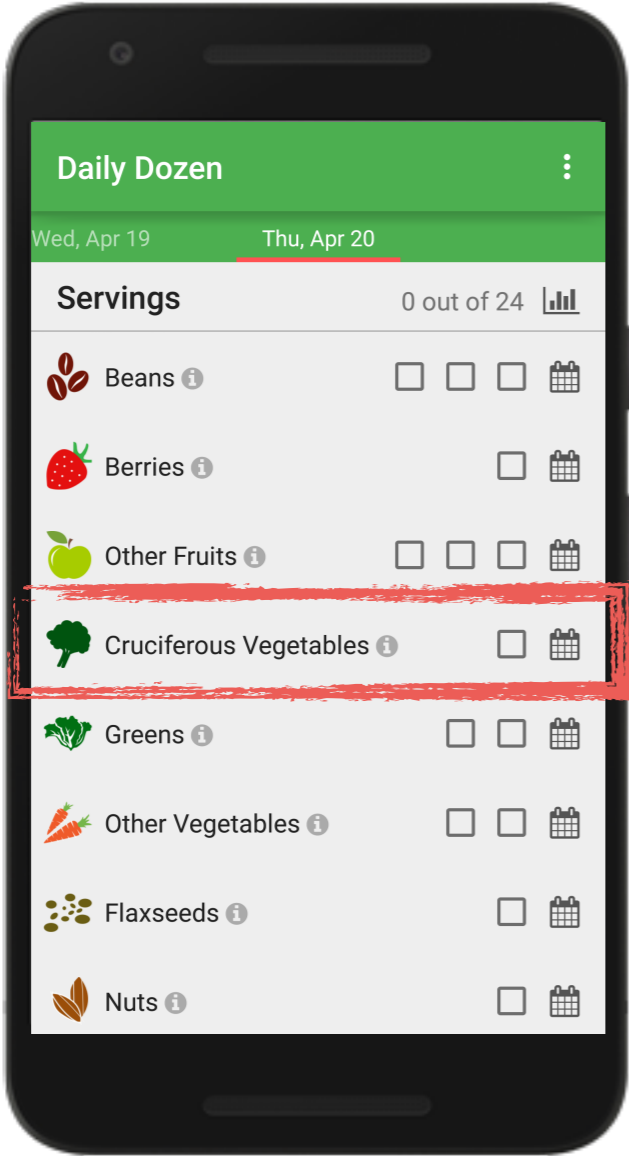
News

Jul 2016

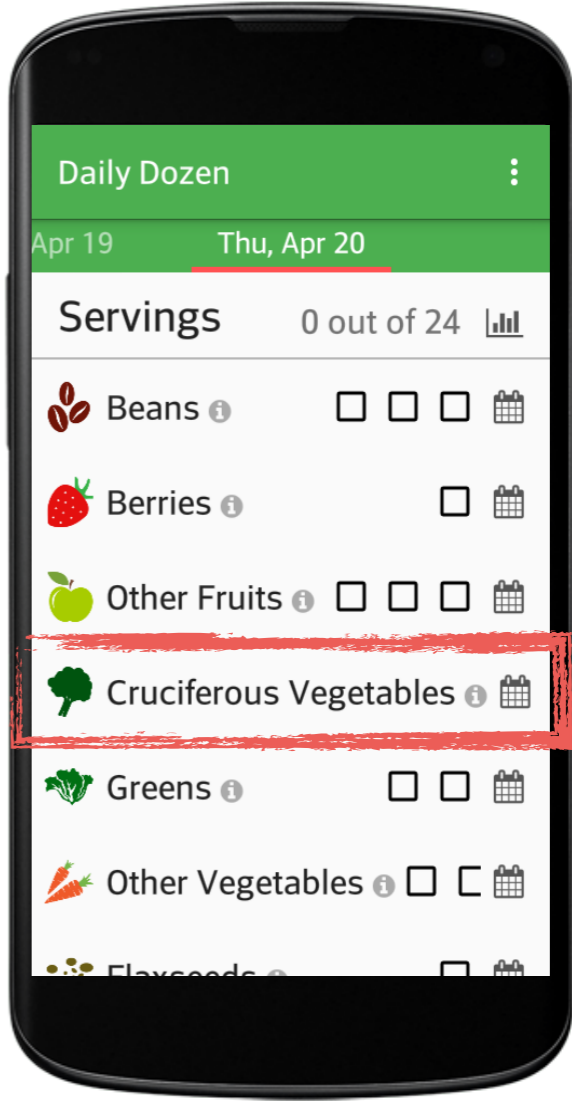
“Due to the wide array of available Android devices, **we are targeting our support to a select number of Android devices** to continue improving our overall Salesforce1 for Android user experience”

Motivating Example

LG G3

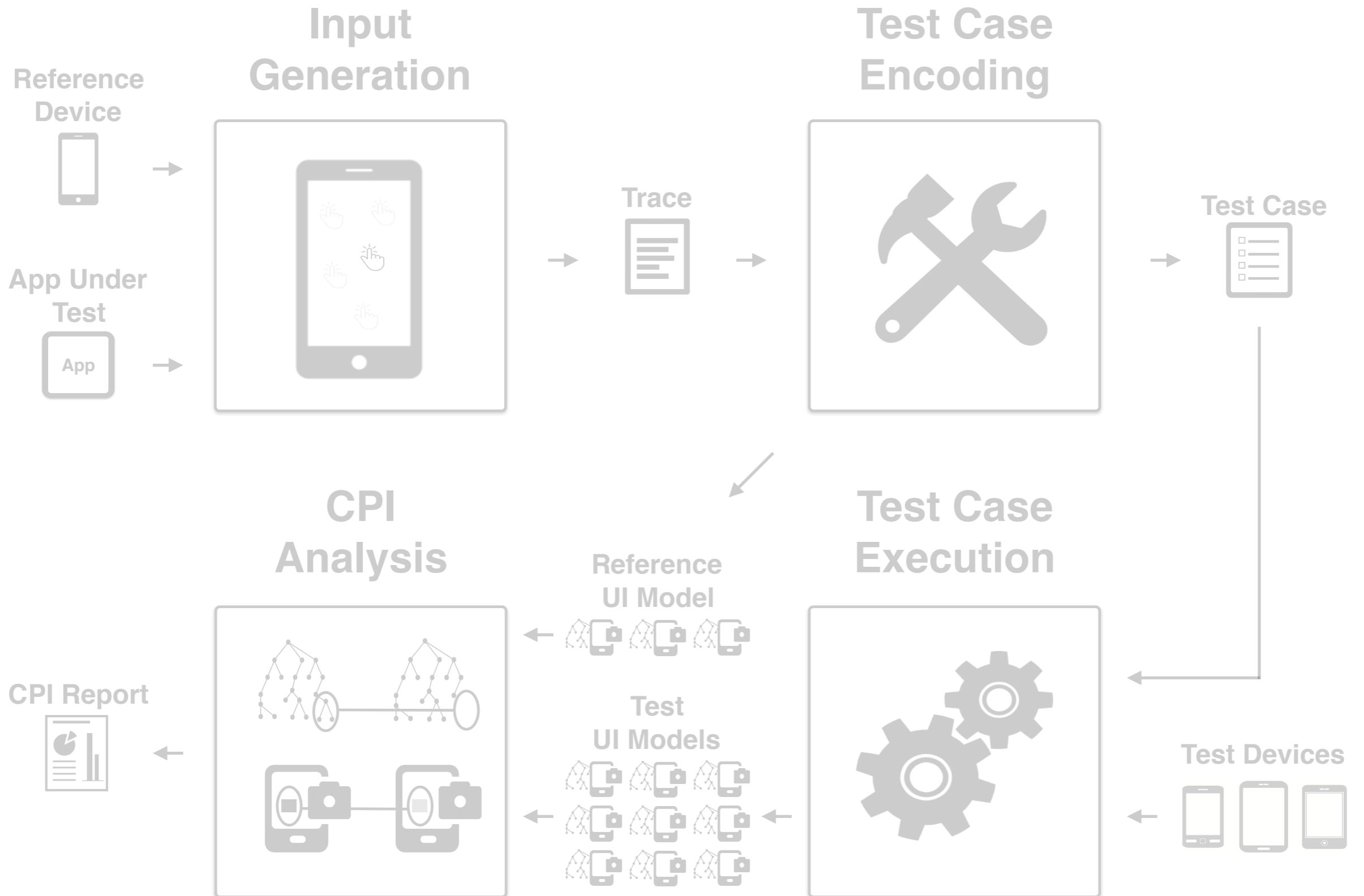


LG Optimus L70



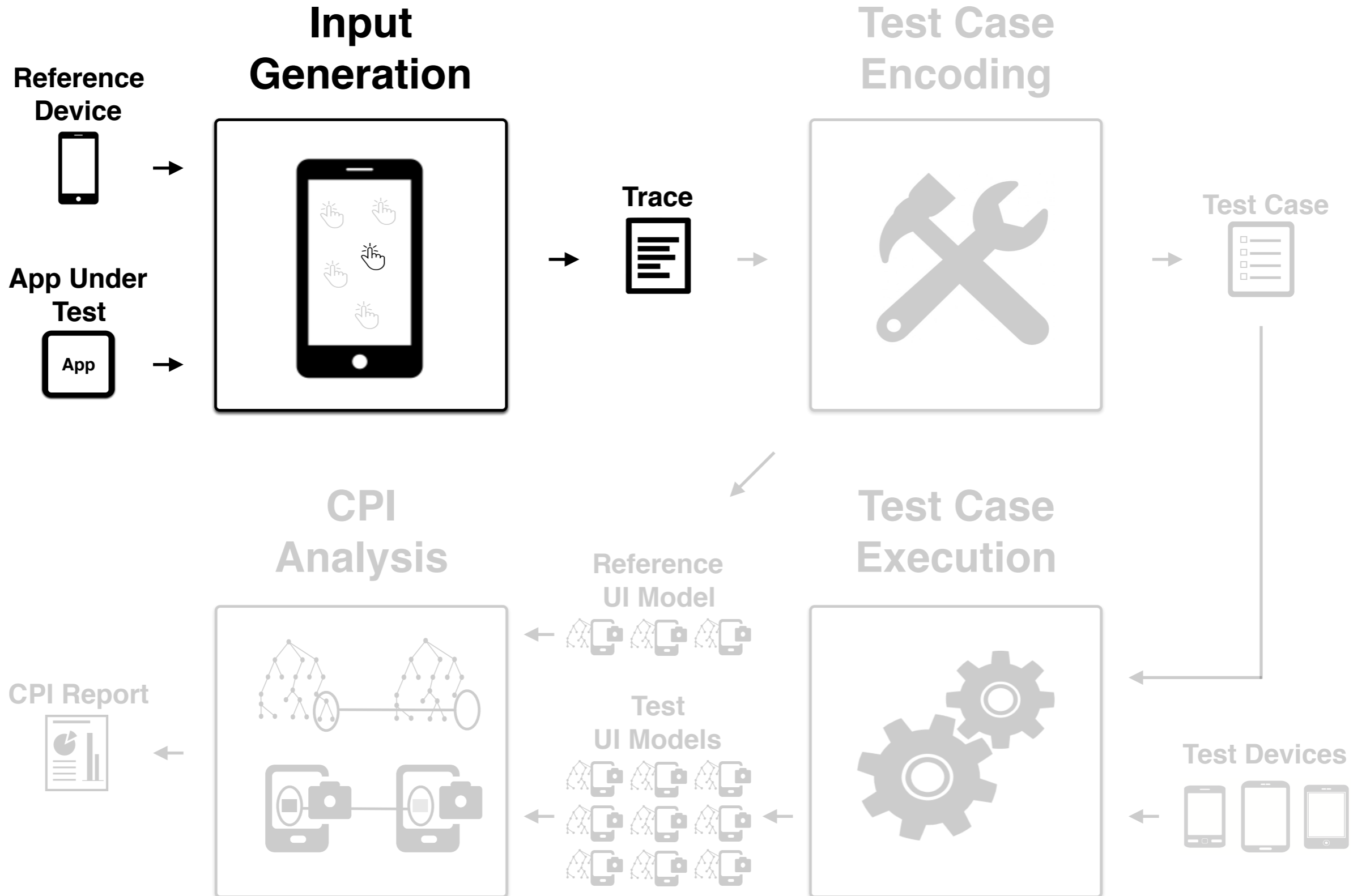
DIFFDROID

ASE'17



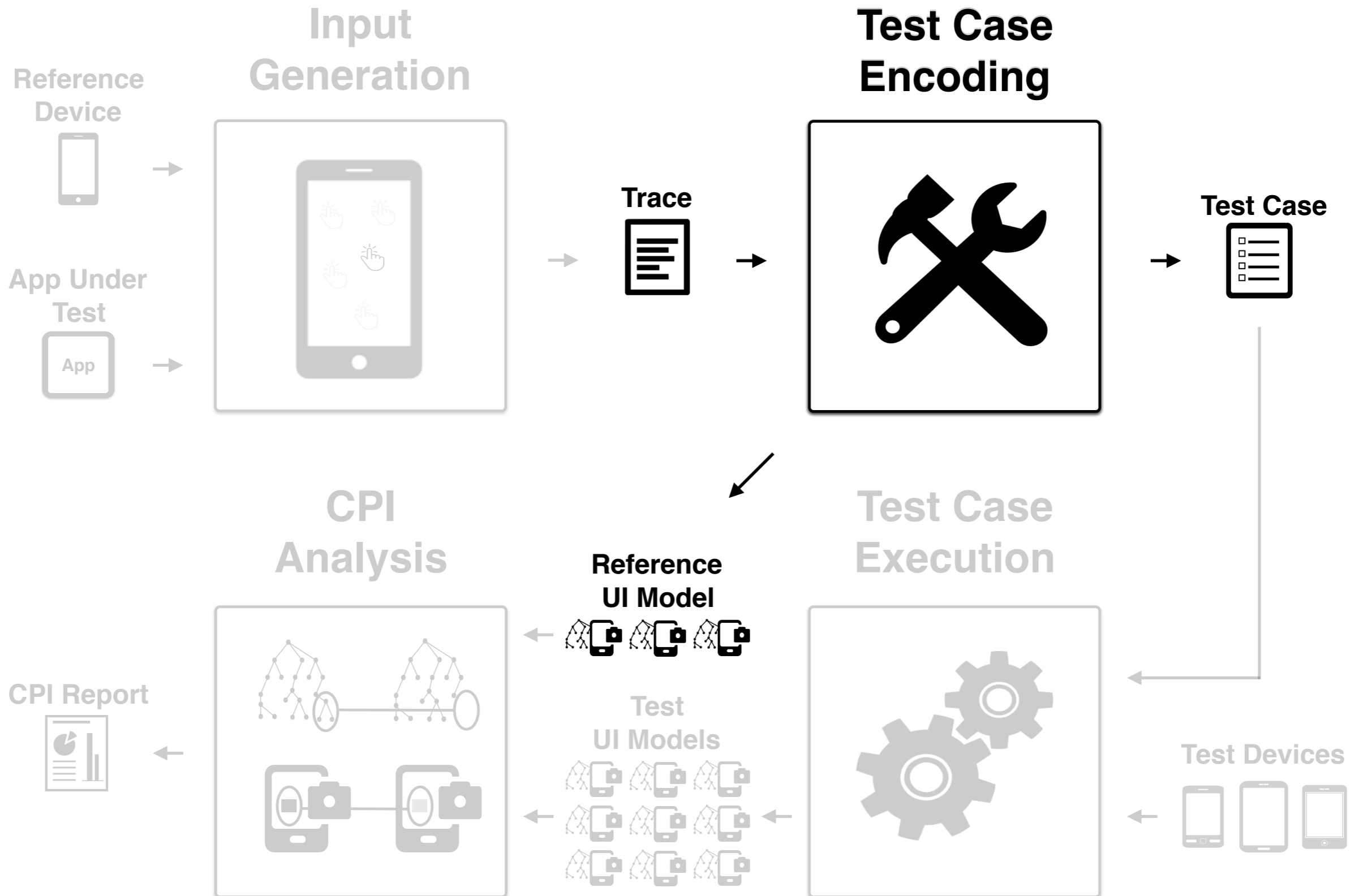
DIFFDROID

ASE'17



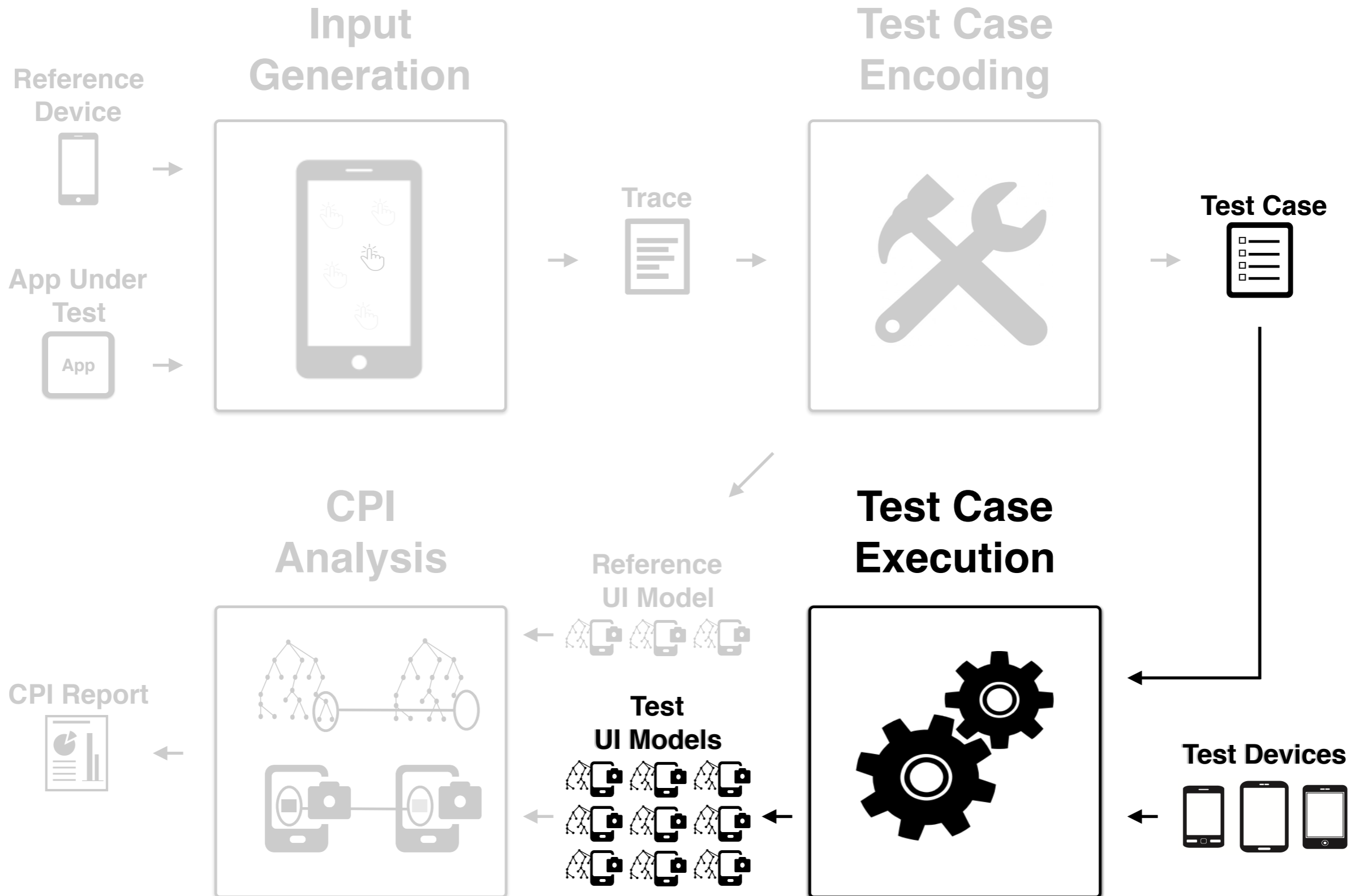
DIFFDROID

ASE'17



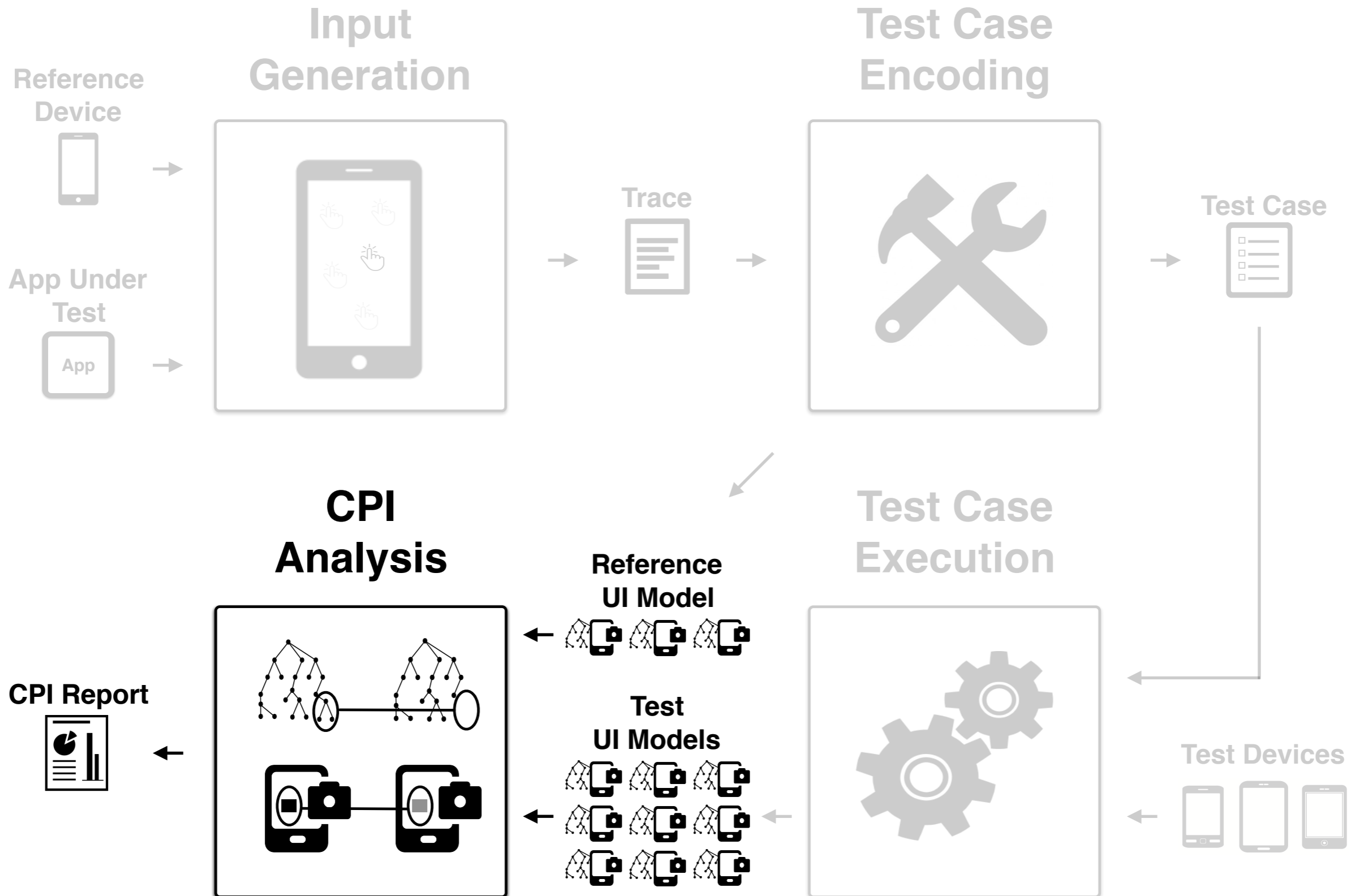
DIFFDROID

ASE'17



DIFFDROID

ASE'17



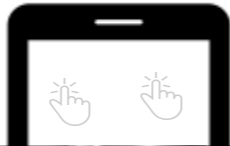
DIFFDROID

ASE'17

Input Generation

Test Case Encoding

Reference Device



Trace



Test Case

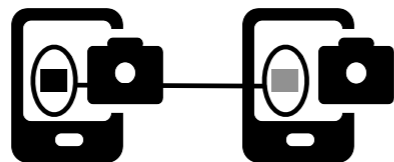


GUI Inconsistencies

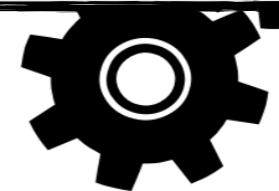
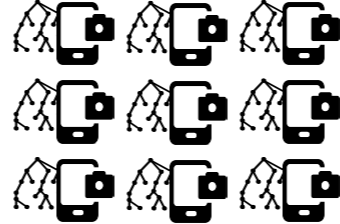
96

False Positives

14%



Or Models



Test Devices



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 after such crash, the app would not
reproduce this.
Pretty sure the same thing happened

Version:
Phone: N
Android 7
LineageOS
Kernel:3

Trolldemorted commented on May 19, 2017

I have:

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>

open and closed issues for
<https://github.com/WhisperSystems>

Description

es.

Mar 28, 2017

```
runtime(30920): Shutting down  
runtime(30920): FATAL EXCEPTION:  
runtime(30920): Process: org  
runtime(30920): java.lang.IllegalStateException:  
runtime(30920): at org
```

Tweet opens with selected account

Actual behavior

Twidere crashes

Steps to reproduce

Longpress retweet icon underneath t

afwang commented on

I found this crash on v
[redreader_crash_log_f](#)

Reproduction steps:

1. Launch redreader
subscribed subscribed
2. Tap the refresh button in the app bar
3. Swipe down while the subscribed st

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

QuantumBadger added the **Un**



Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>

<Logcat information>

Steps to reproduce the behavior

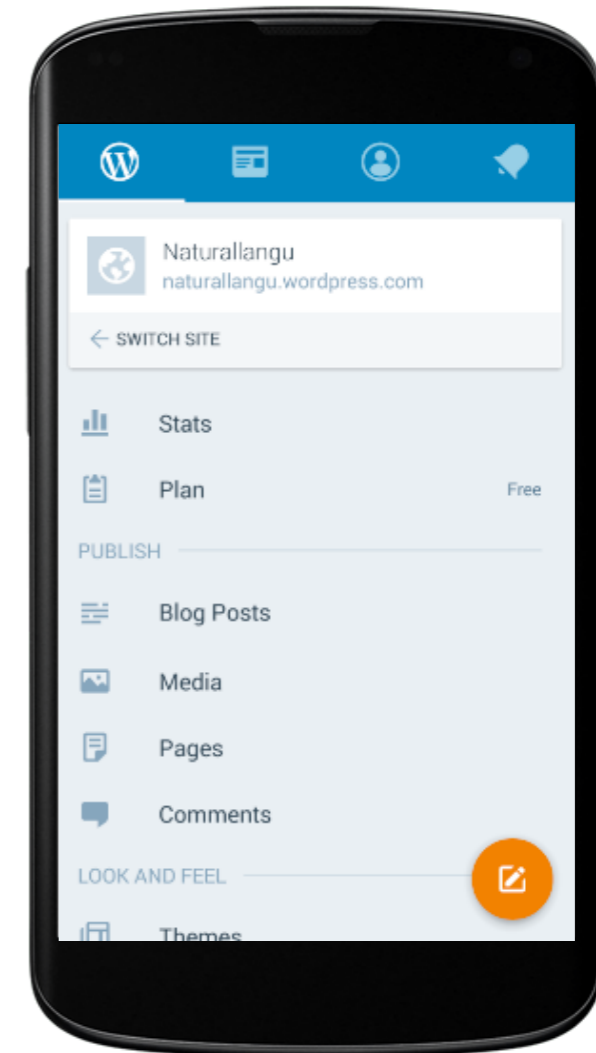
1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

Developer



WORDPRESS App



Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>

<Logcat information>

Steps to reproduce the behavior

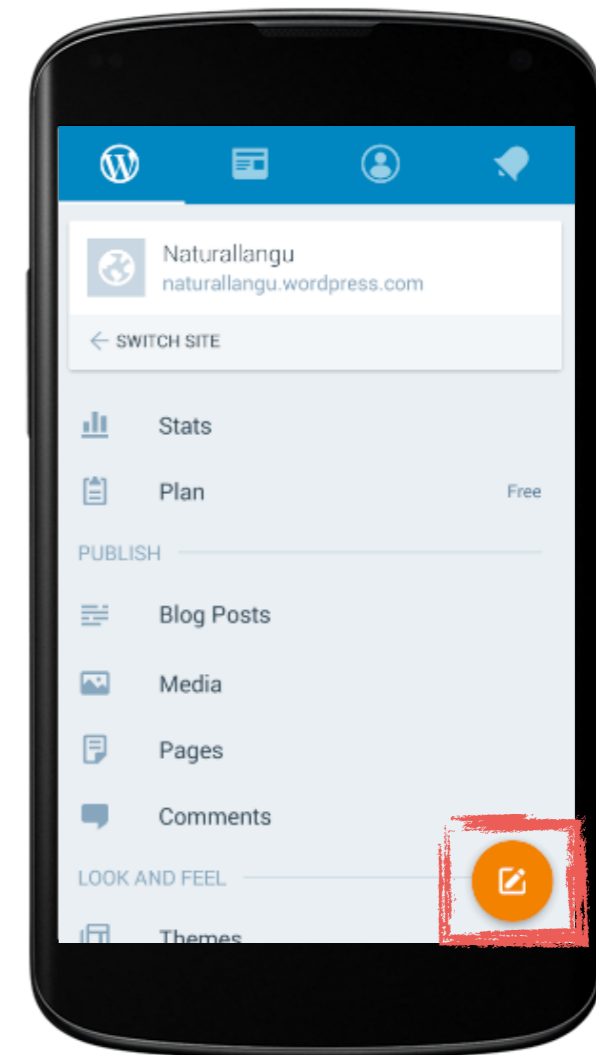
1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

Developer



WORDPRESS App



Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>

<Logcat information>

Steps to reproduce the behavior

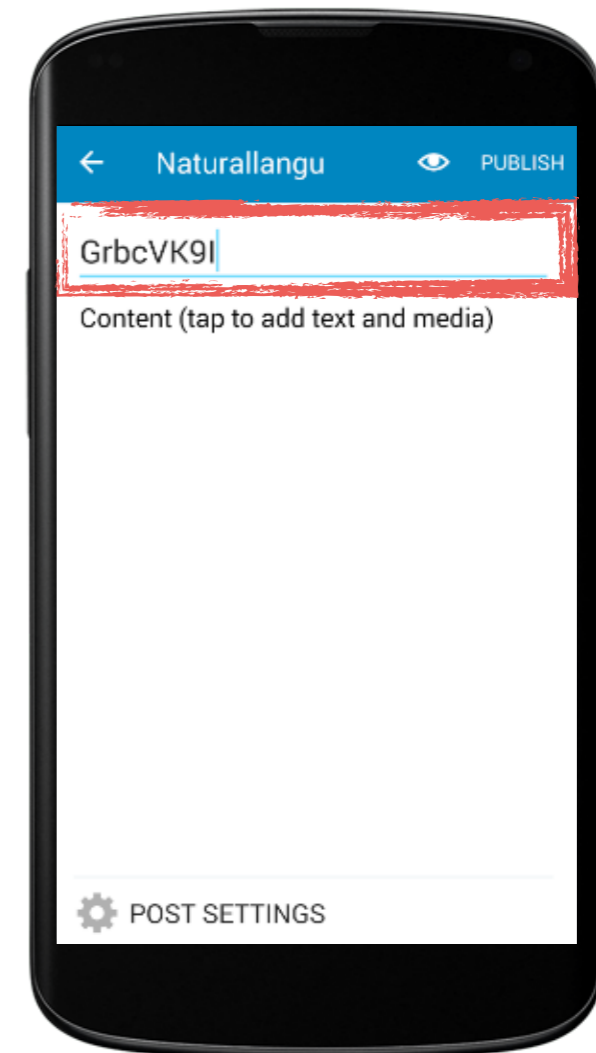
1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

Developer



WORDPRESS App



Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>

<Logcat information>

Steps to reproduce the behavior

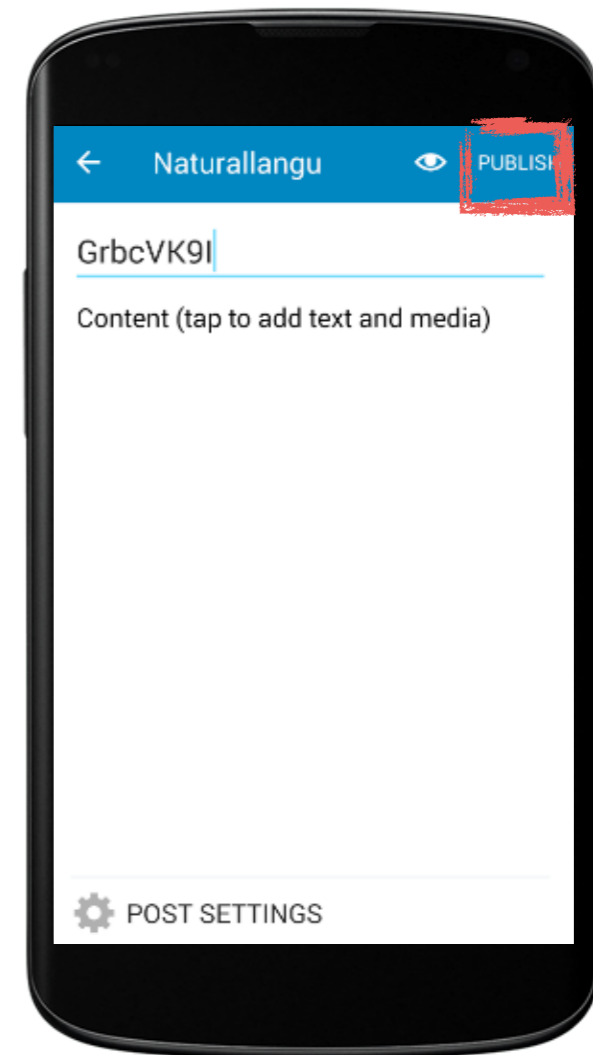
1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

Developer



WORDPRESS App



Motivating Example

Bug Report

Onko commented on Mar 24, 2017

<Text paragraph>

<Logcat information>

Steps to reproduce the behavior

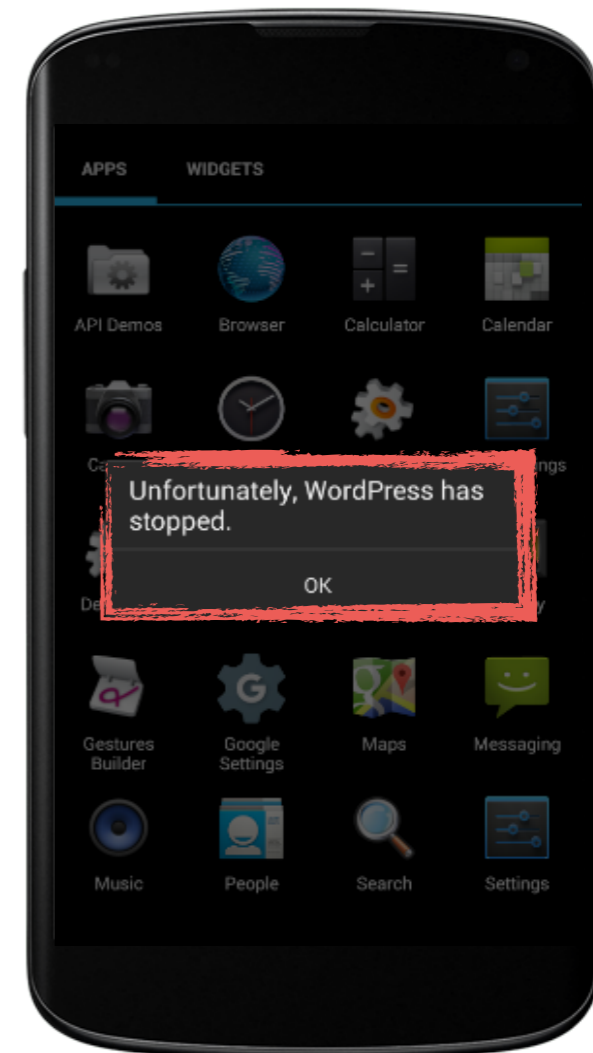
1. Start a new post
2. Type something
3. Tap on the Publish button
4. The app crashes

<Text paragraph>

Developer

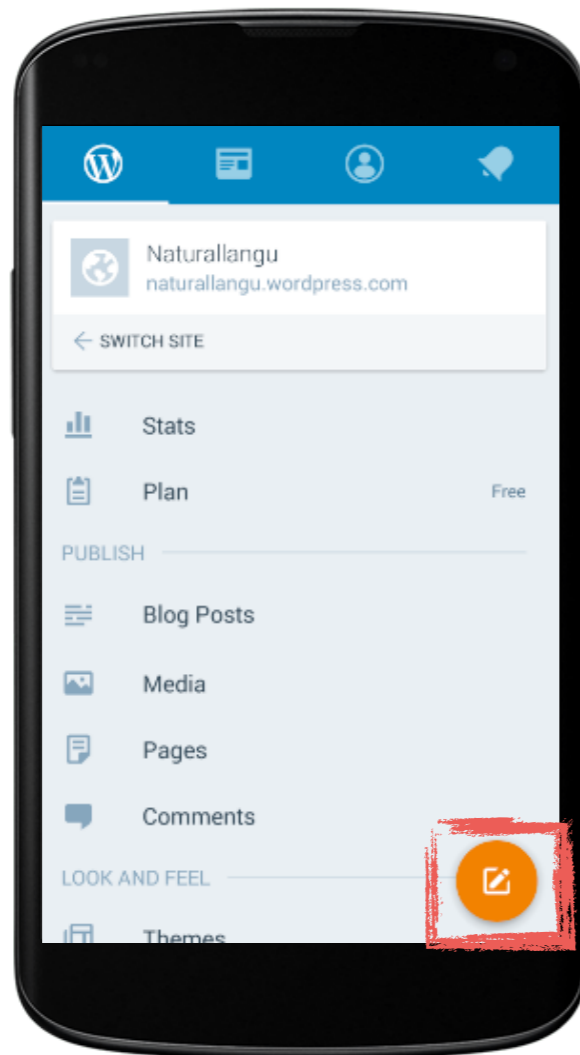


WORDPRESS App



Motivating Example

WORDPRESS App



Developer

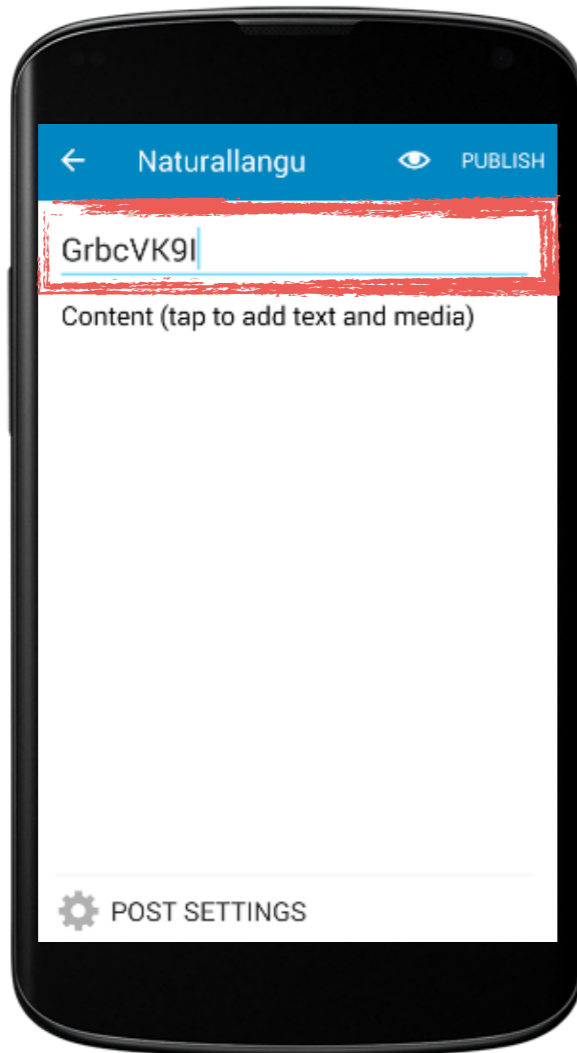


Test Case

```
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



Developer

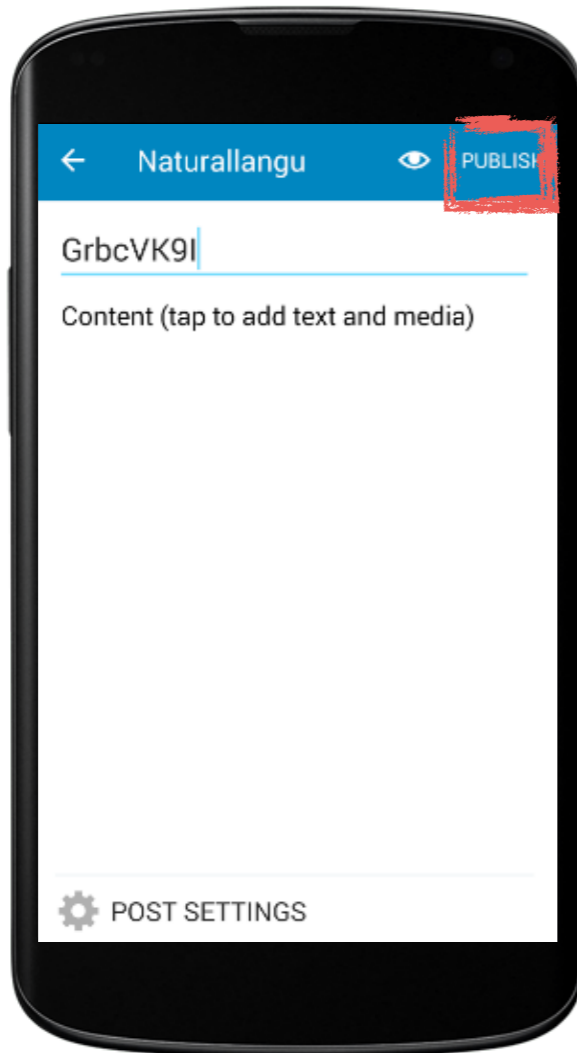


Test Case

```
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



Developer



Test Case

```
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

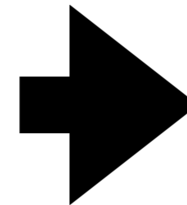
<Text paragraph>

<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>

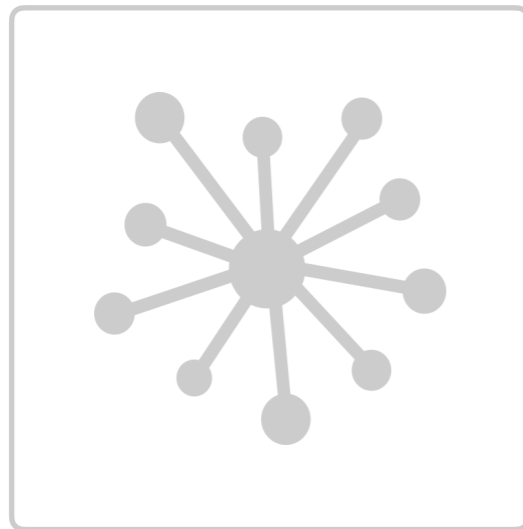


Test Case

```
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());  
}
```

YAKUSU

Ontology Extraction



Bug Report Analysis



UI Actions Search



Relevant App



Ontology



Bug Report



Abstract Steps



Test Device



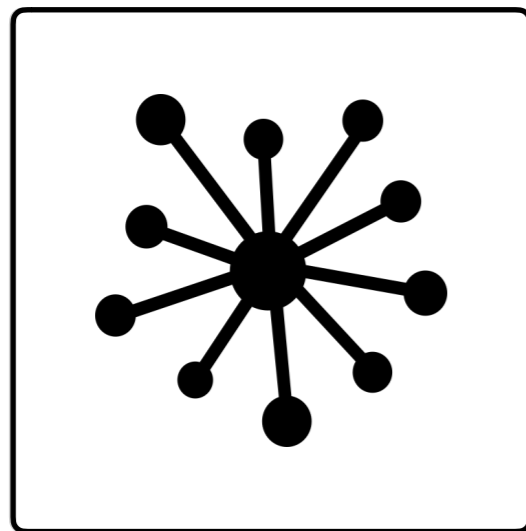
Test Case



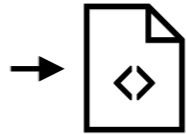
YAKUSU

Ontology Extraction

Relevant App



Ontology



Bug Report



Bug Report Analysis



Abstract Steps



Test Device



UI Actions Search

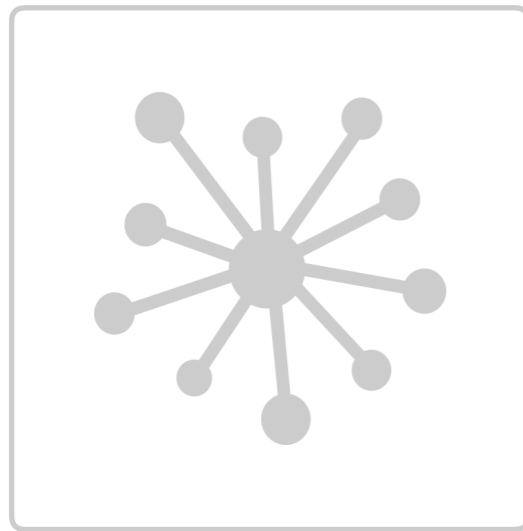


Test Case

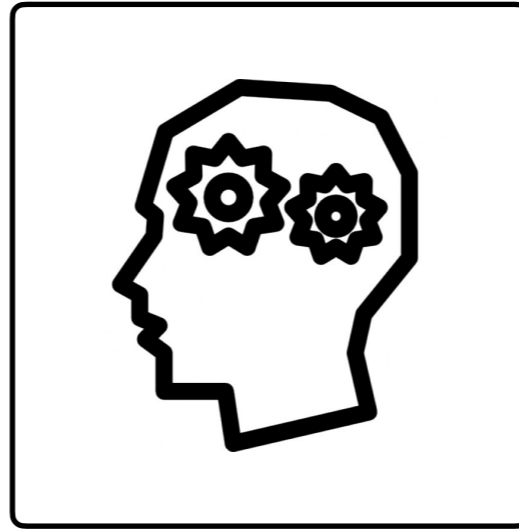


YAKUSU

Ontology Extraction



Bug Report Analysis



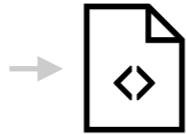
UI Actions Search



Relevant App



Ontology



Bug Report



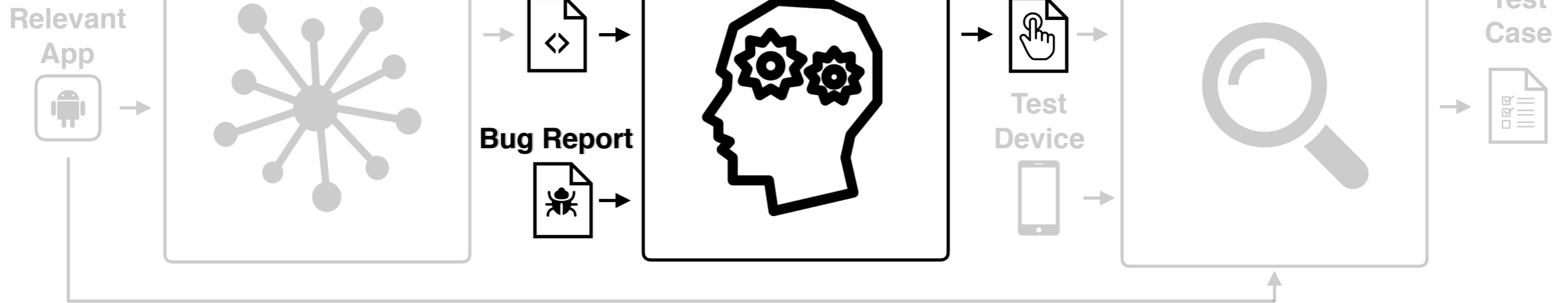
Abstract Steps



Test Device



Test Case

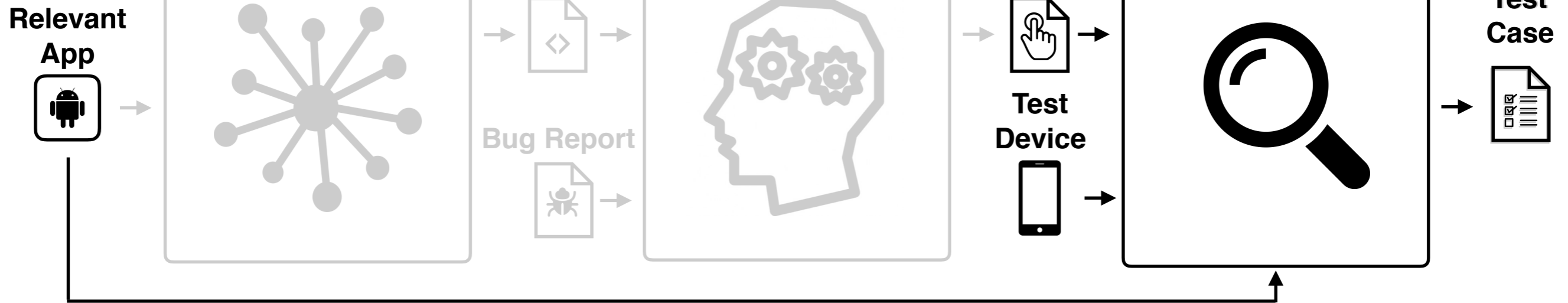


YAKUSU

Ontology Extraction

Bug Report Analysis

UI Actions Search



YAKUSU

**Ontology
Extraction**

**Bug Report
Analysis**

**UI Actions
Search**

Translated Bug Report

60%

Cost of running YAKUSU

YAKUSU takes only a few minutes to process a bug report

Releva
App



Test
Case



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



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.

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() ==
8 -       NetworkInfo.State.CONNECTED) {
9       return true;
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 >=
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 + } else {
15 +   NetworkInfo[] info = ctv.getAllNetworkInfo();
16 +   if (info != null) {
17     ...
22  }
23  return false;
24 }
```

Motivating Example

Update Example

Update Example

Update Example

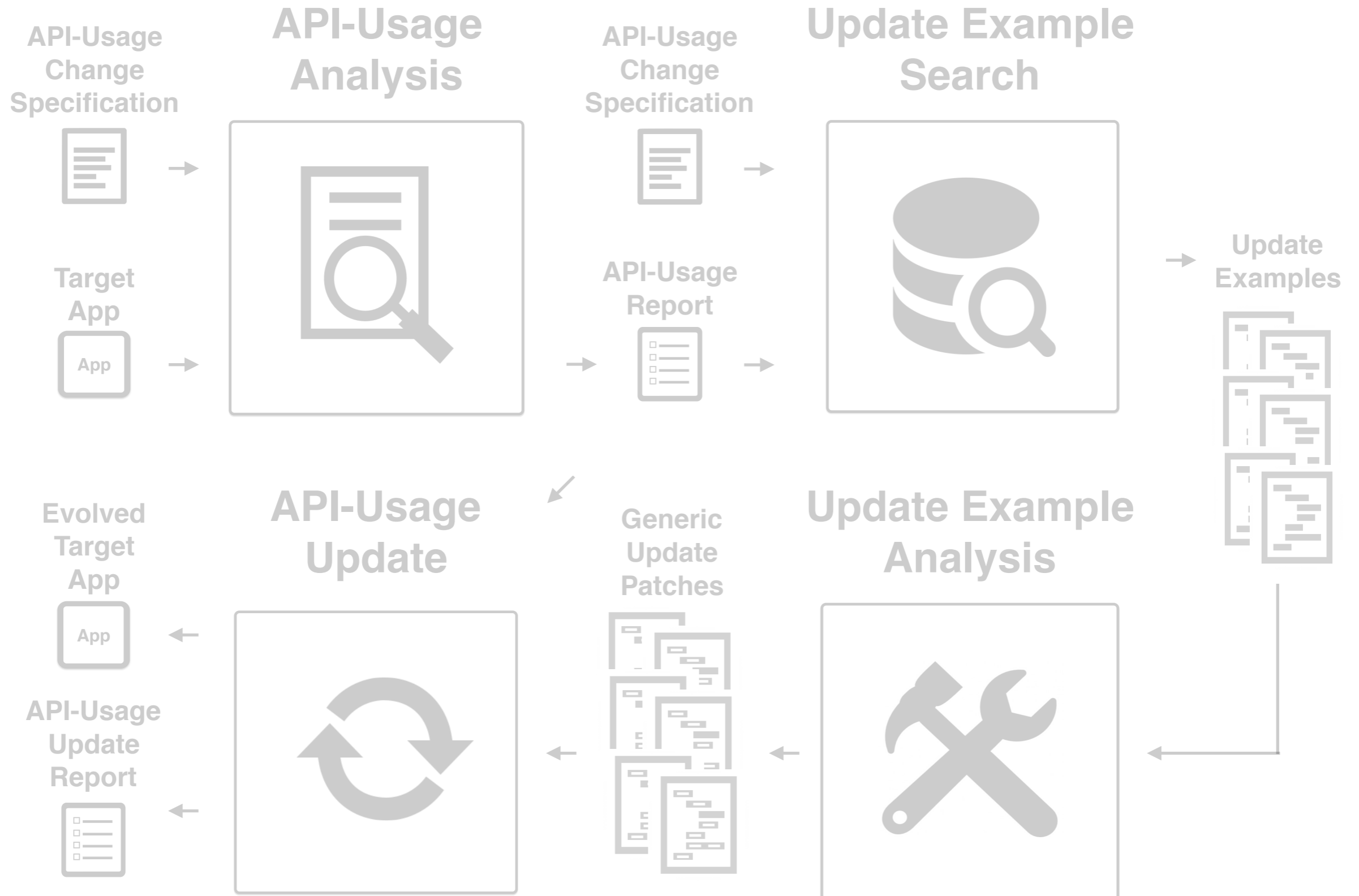
Old Version

```
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() ==
8 -       NetworkInfo.State.CONNECTED) {
9       return true;
10    } }
11  }
12  return false;
13 }
```

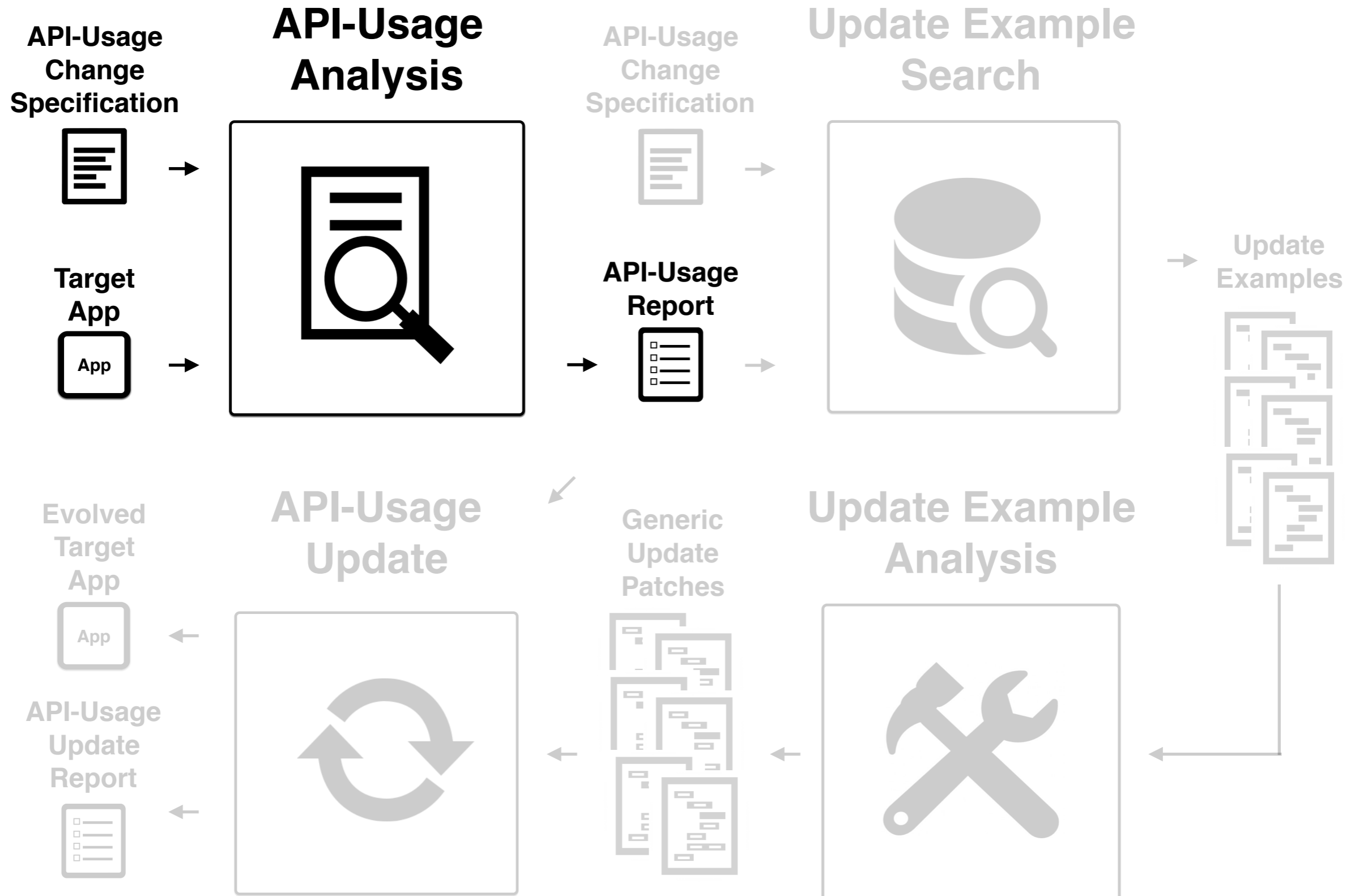
New Version

```
1 public boolean isNetworkAvailable(Context ctx) {
2   ConnectivityManager ctv = (ConnectivityManager)
3   ctx.getSystemService(Context.CONNECTIVITY_SERVICE);
4 + if (Build.VERSION.SDK_INT >=
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 + } else {
15 +   NetworkInfo[] info = ctv.getAllNetworkInfo();
16 +   if (info != null) {
17 +     ...
22   }
23   return false;
24 }
```

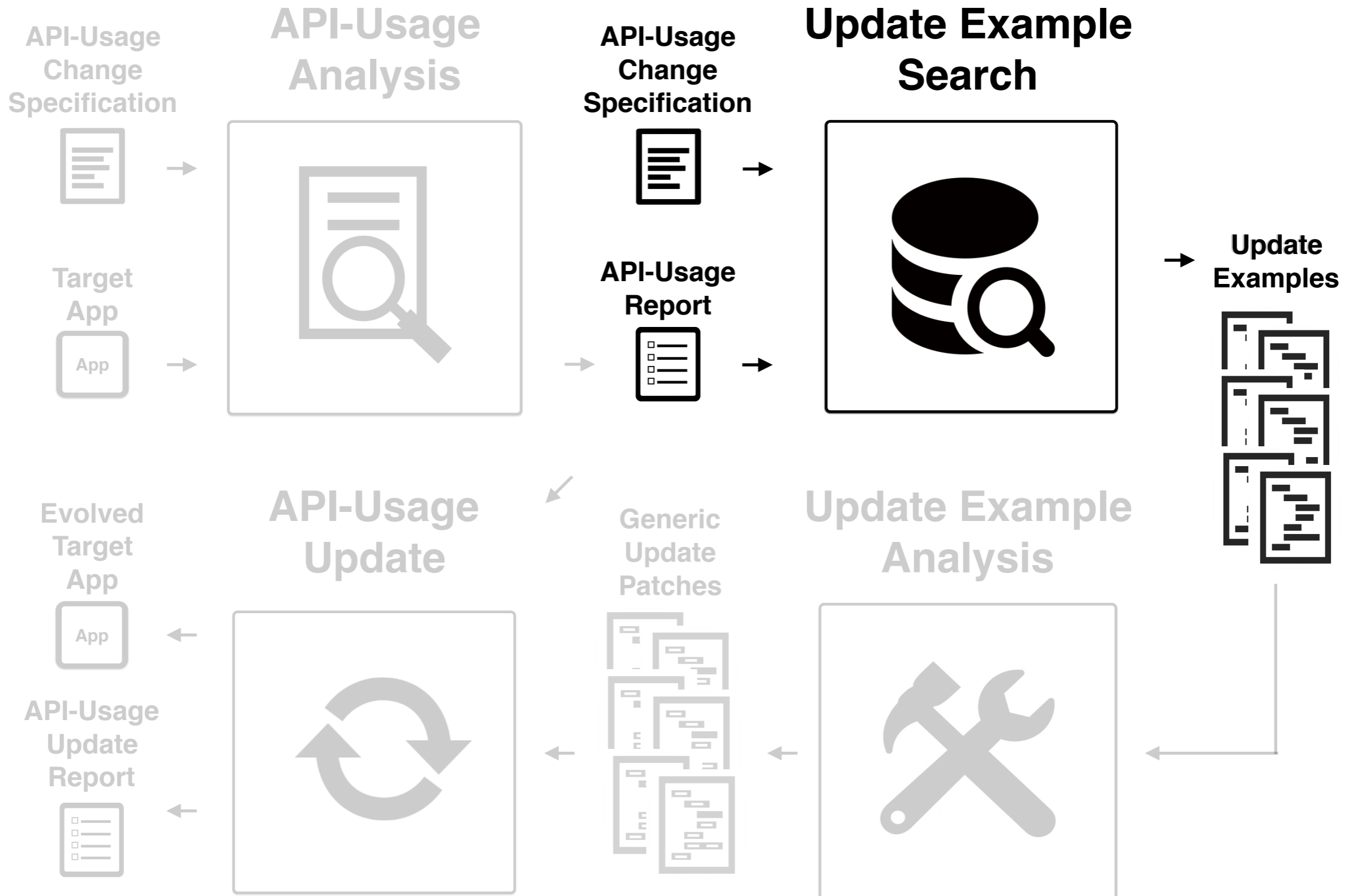
APPEVOLVE



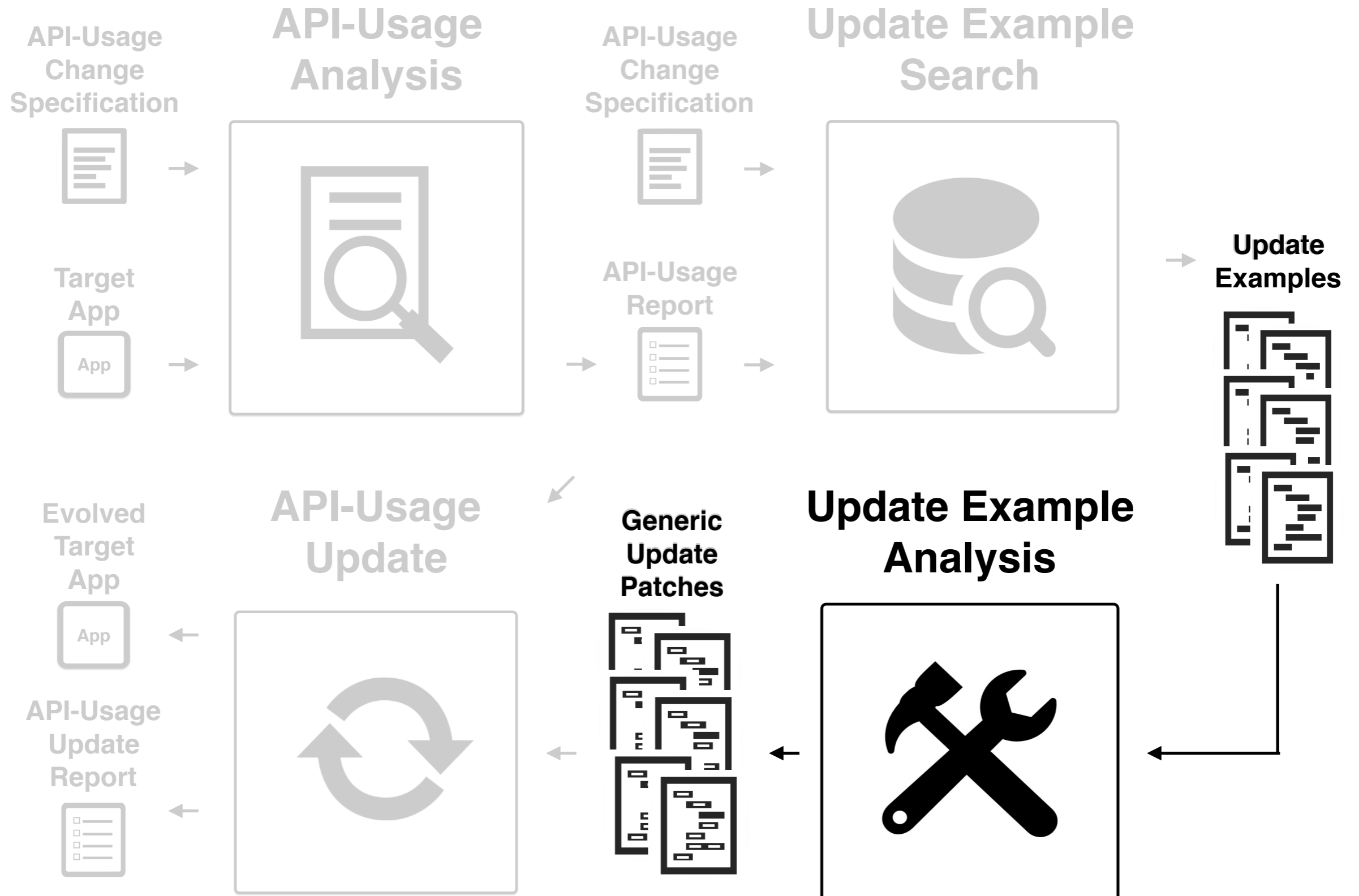
APPEVOLVE



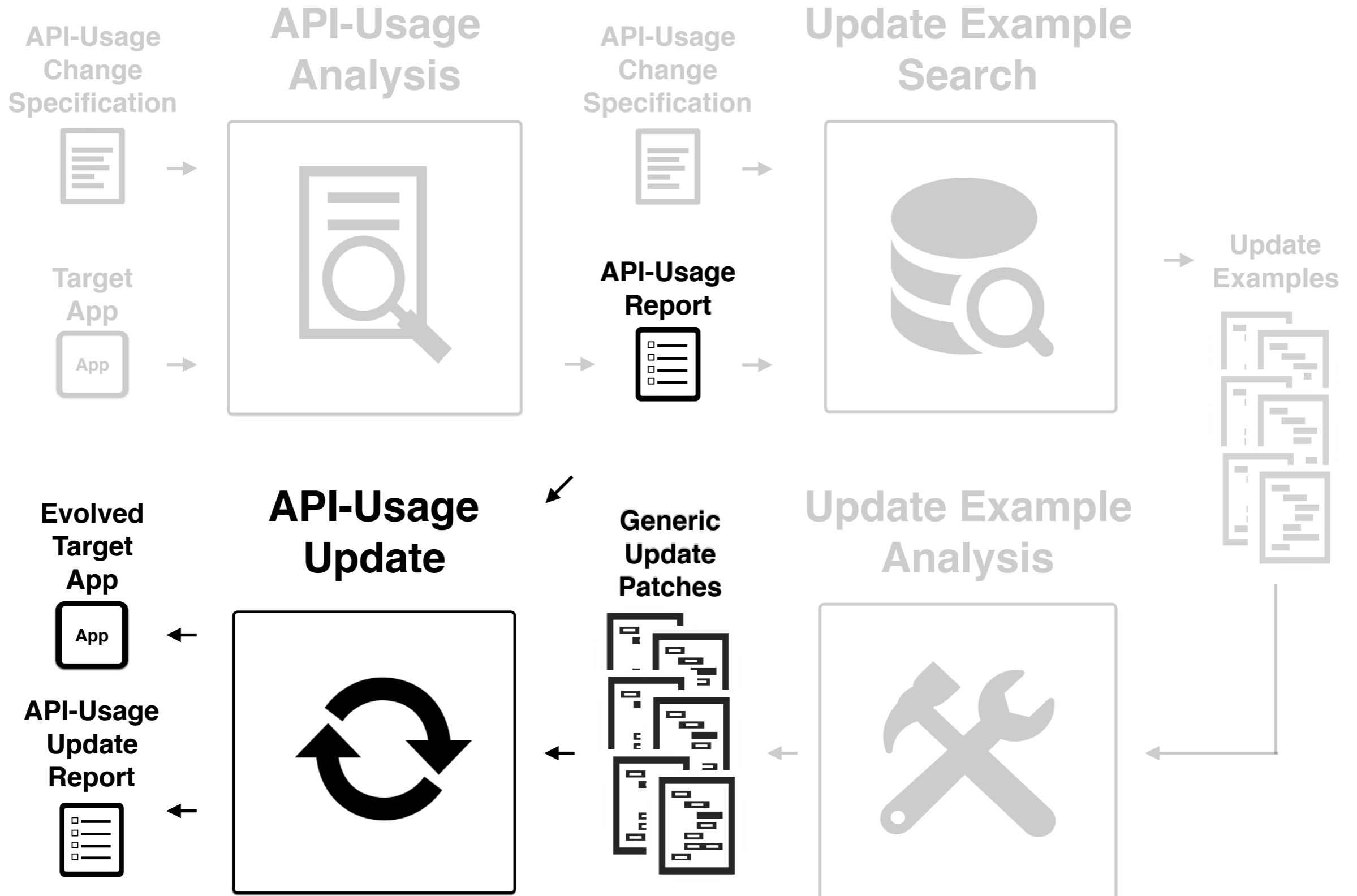
APPEVOLVE



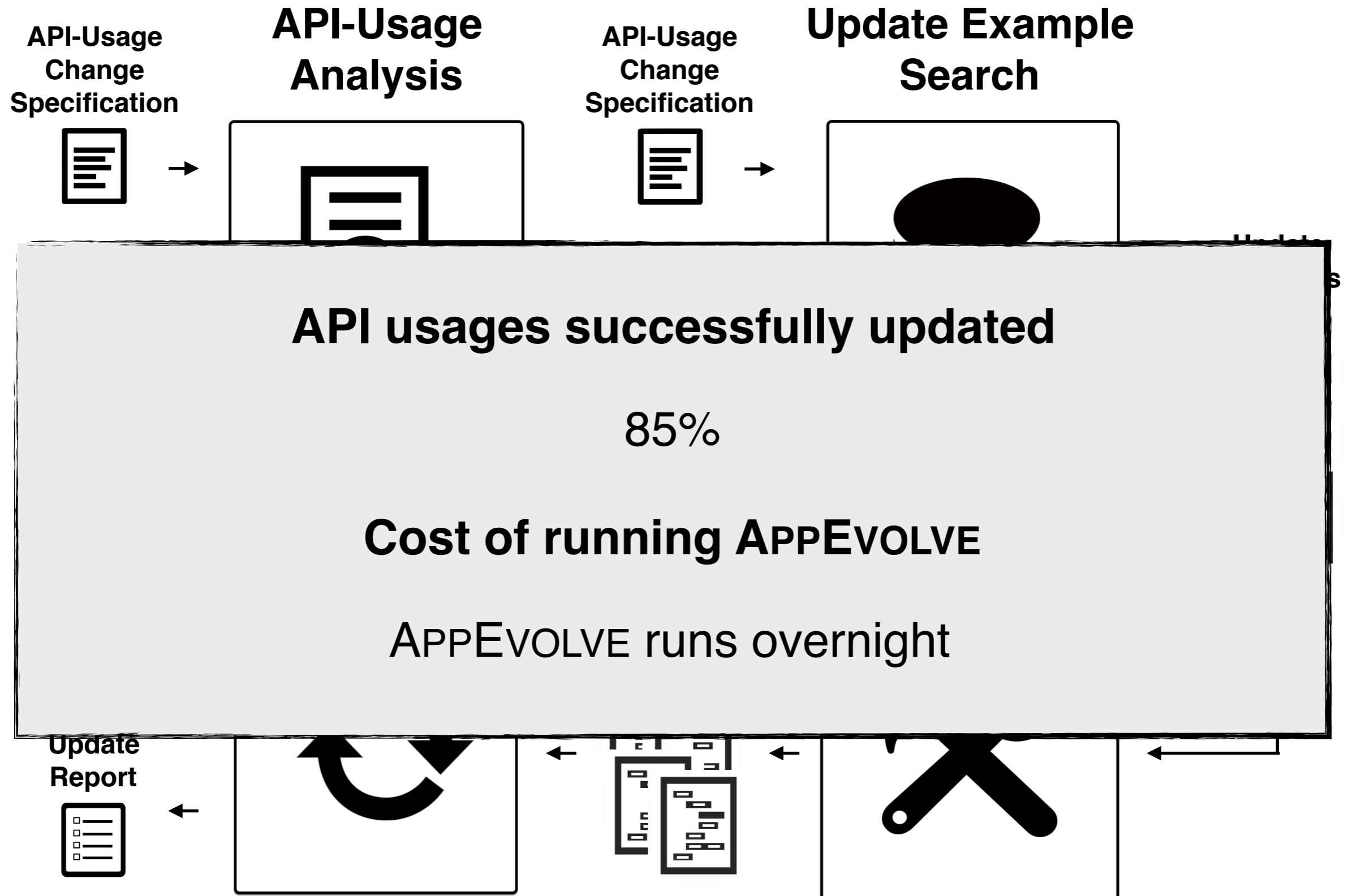
APPEVOLVE



APPEVOLVE



APPEVOLVE



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**

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

Security Research

ISSTA'12
ICSE'15
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

Security Research

ISSTA'12

ICSE'15

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

Security
Research

ISSTA'12

ICSE'15

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

Security
Research

ISSTA'12

ICSE'15

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

Security Research

ISSTA'12

ICSE'15

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