

Barista: A Technique for Recording, Encoding, and Running Platform Independent Android Tests

Mattia Fazzini

マッティア
ファッズィーニ



Eduardo Freitas

エドアルド
フレイタス



Shauvik Roy Choudhary

ショーヴィック
ロイ チャドリー



Alessandro Orso

アレッサンドロ
オルソ



Mobile Applications

Mobile Applications



Mobile Applications



Mobile Applications



Mobile Applications



Mobile Applications



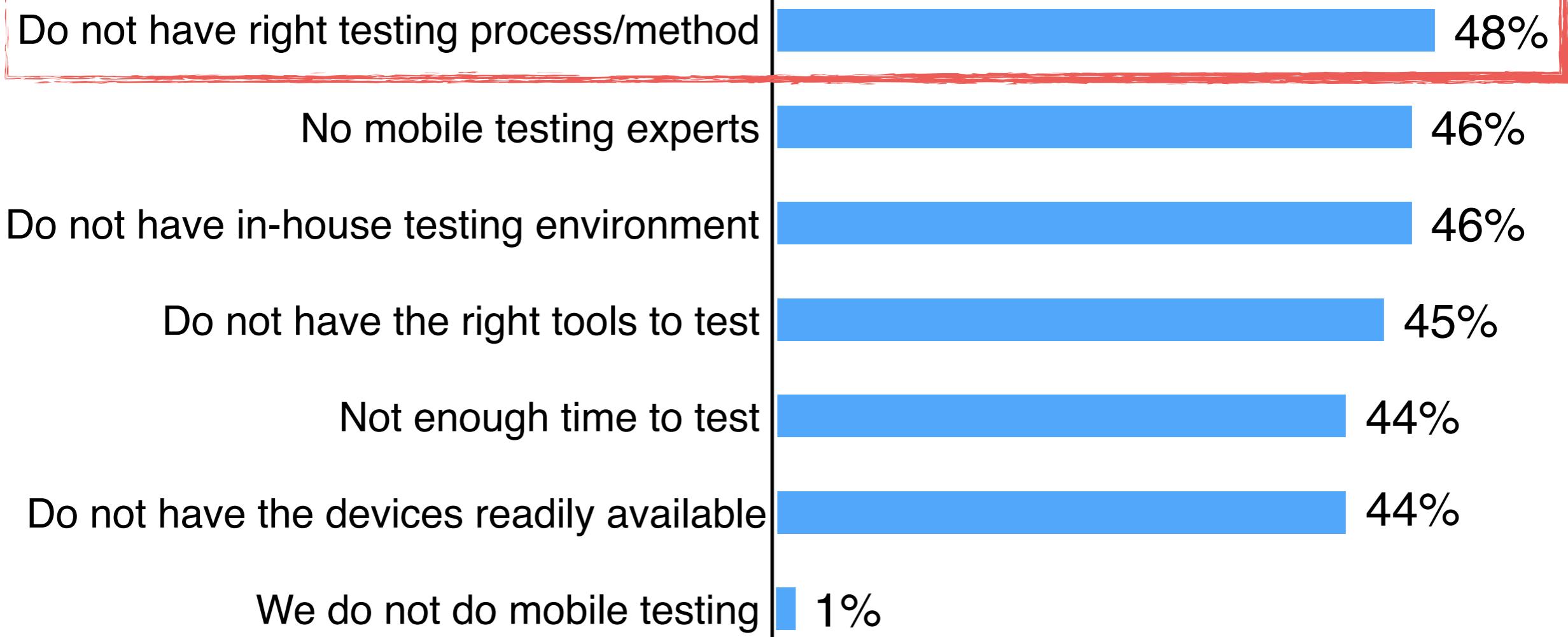
Testing of Mobile Applications

Testing Challenges



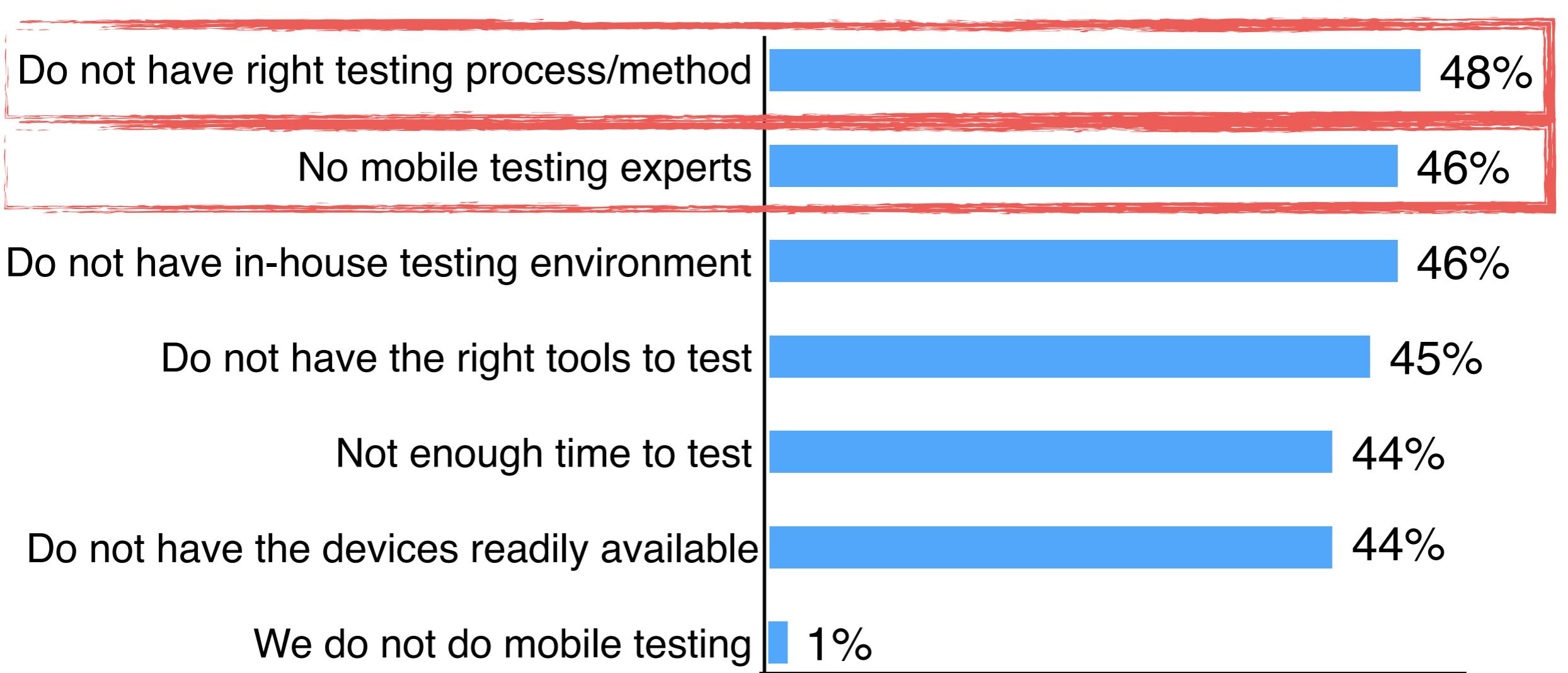
Testing of Mobile Applications

Testing Challenges



Testing of Mobile Applications

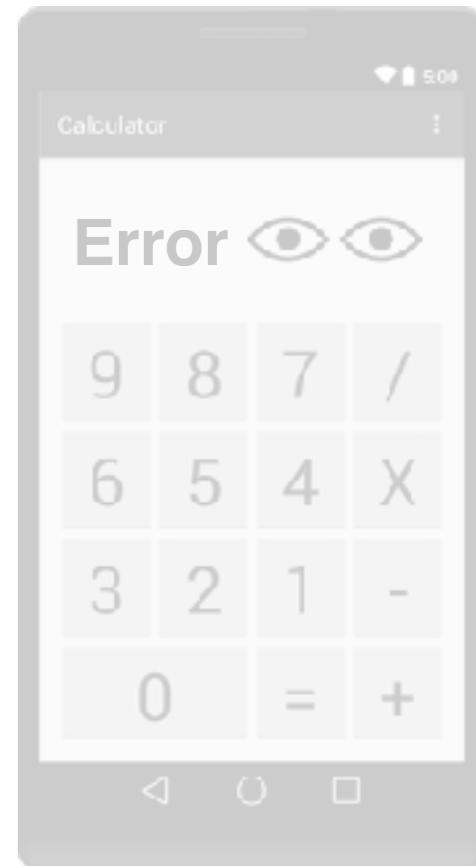
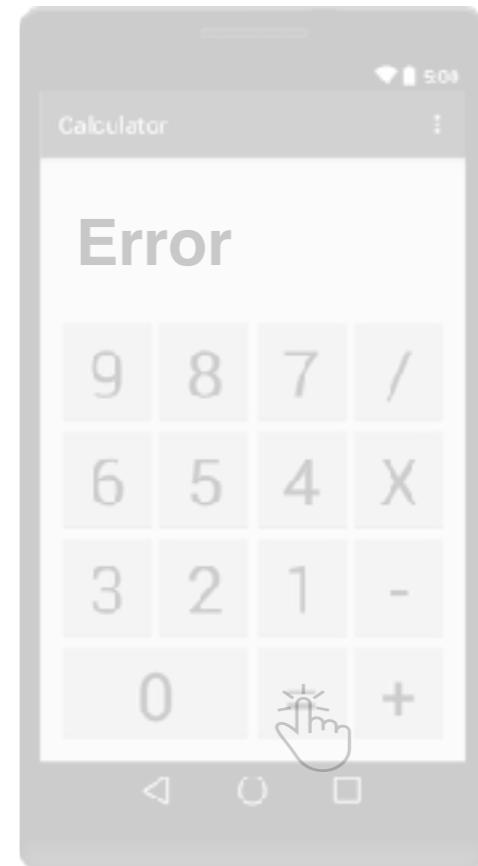
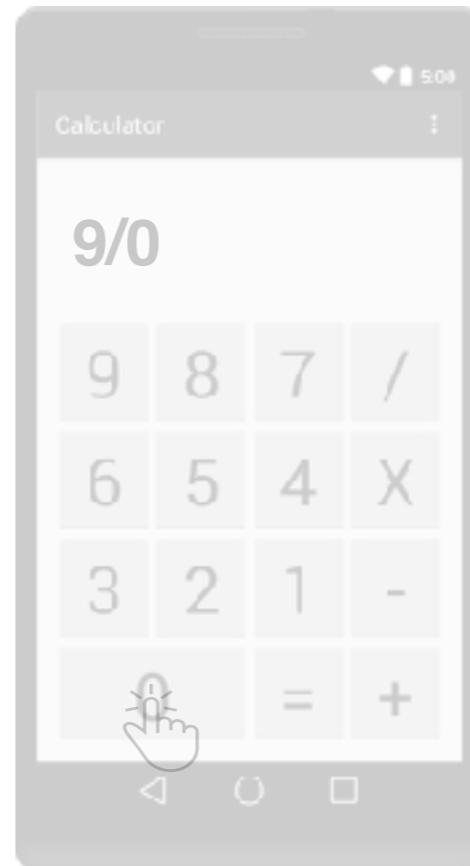
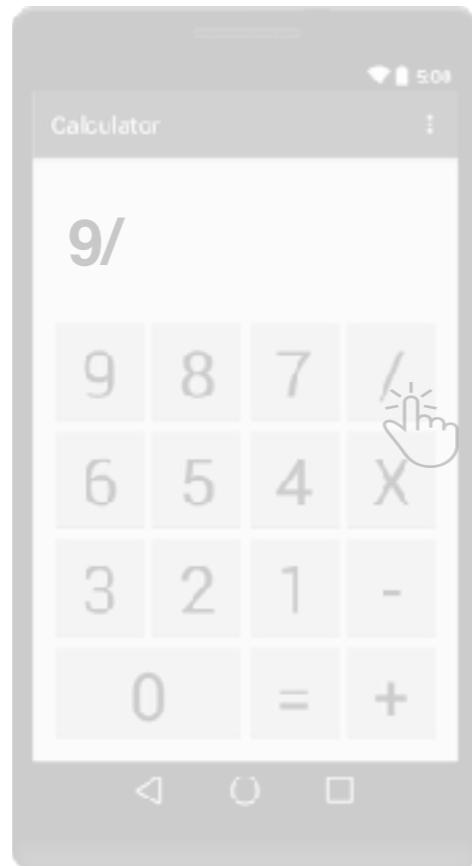
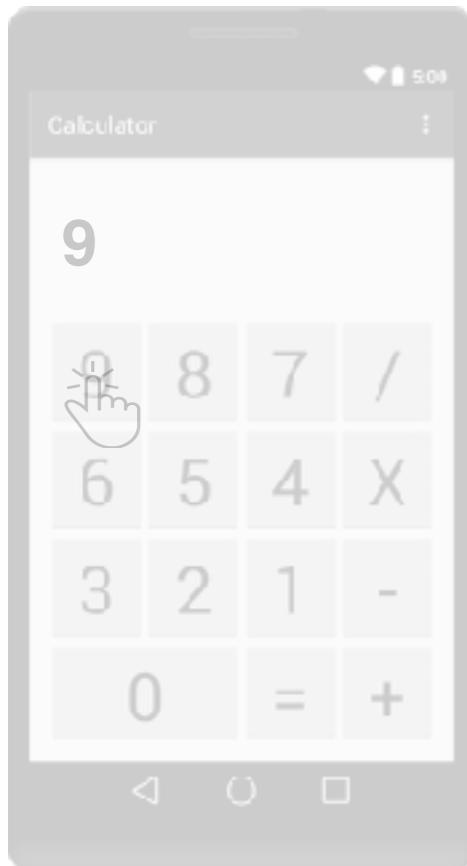
Testing Challenges



Running Example: Manual Testing

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

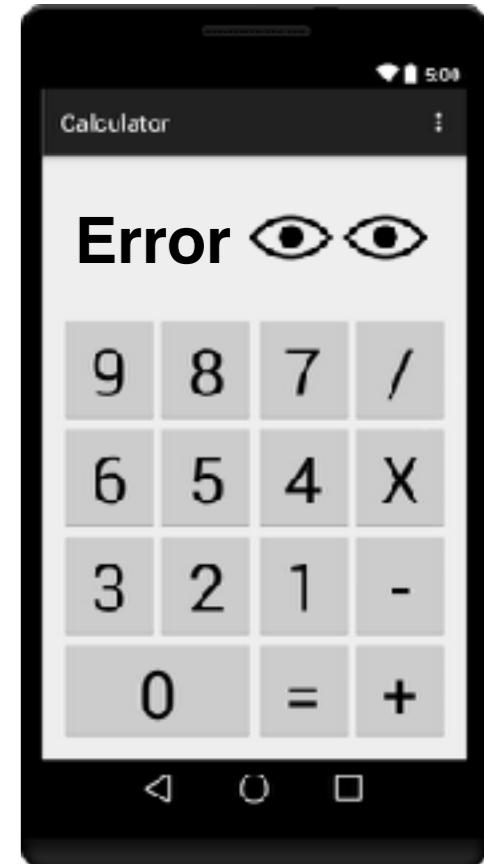
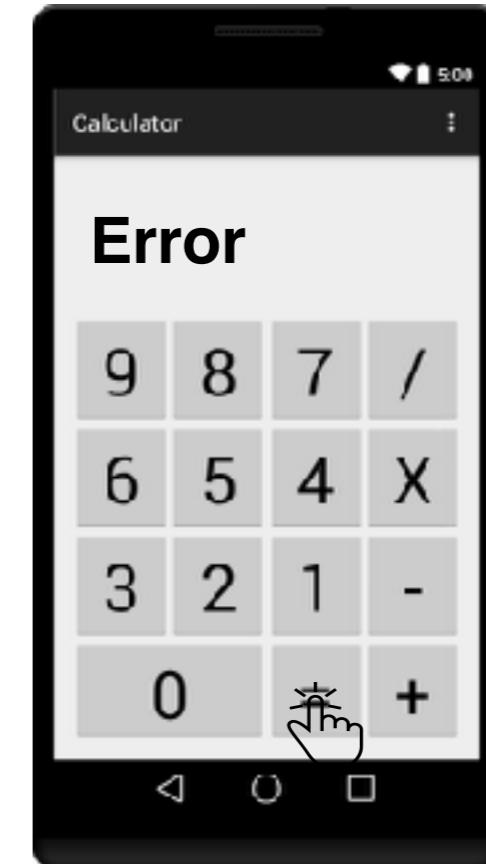
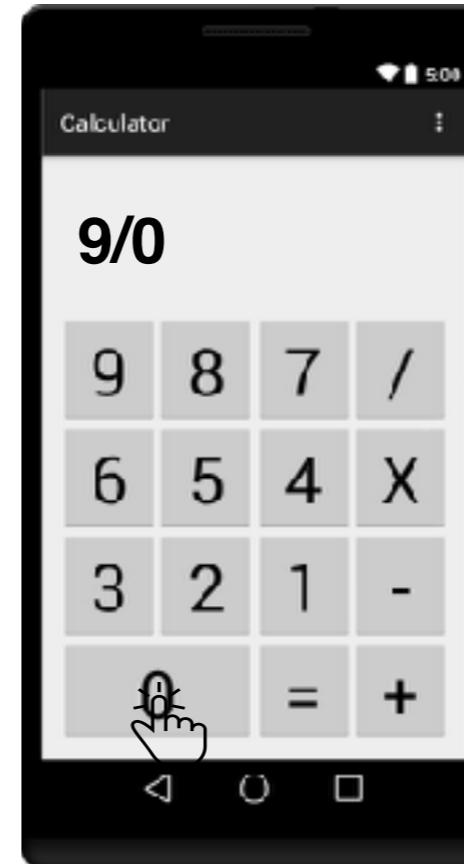
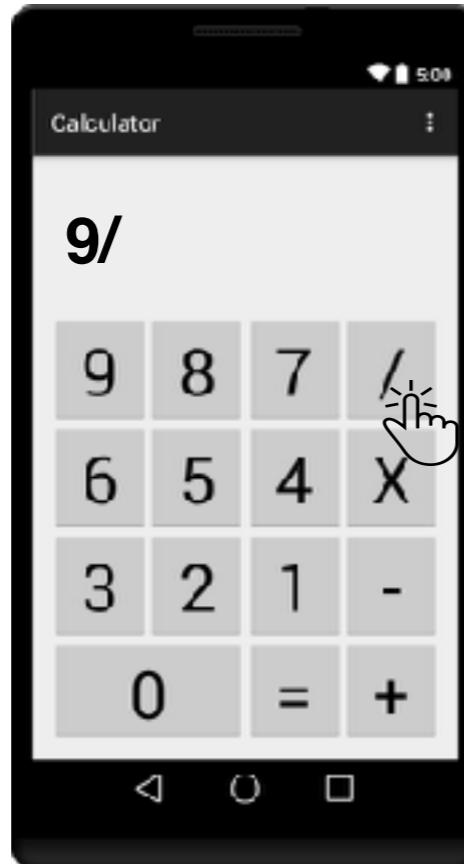
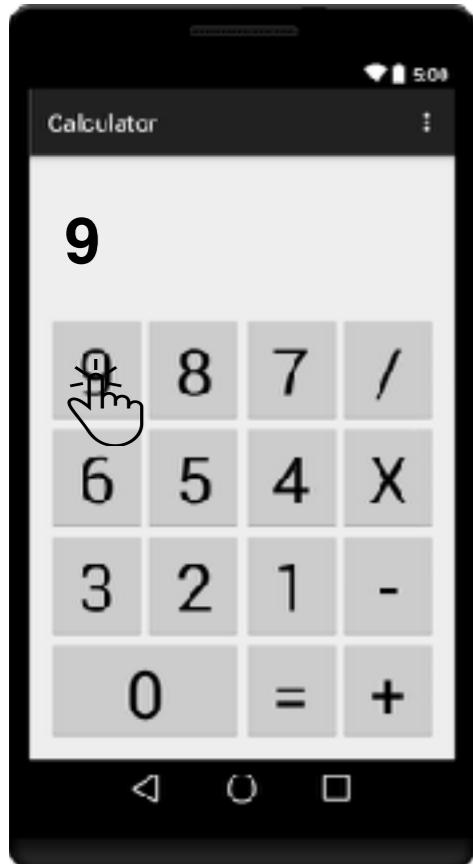
- (1) Press nine
- (2) Press divide
- (3) Press zero
- (4) Press equal
- (5) Check message



Running Example: Manual Testing

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

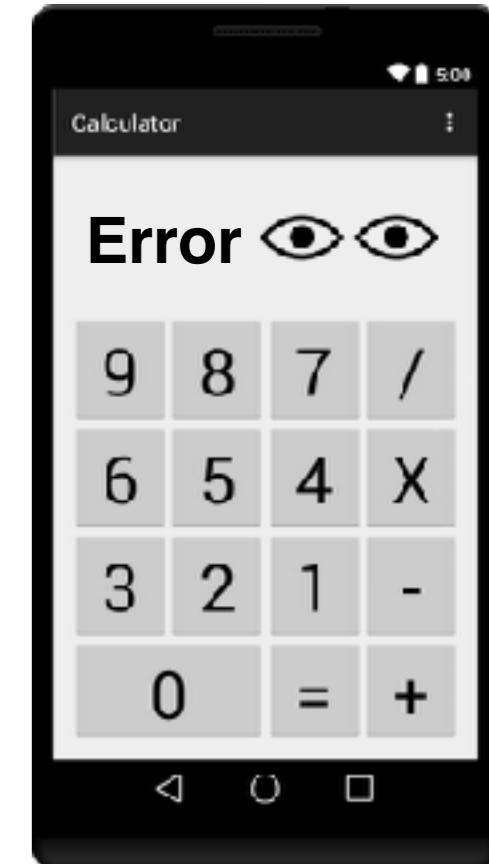
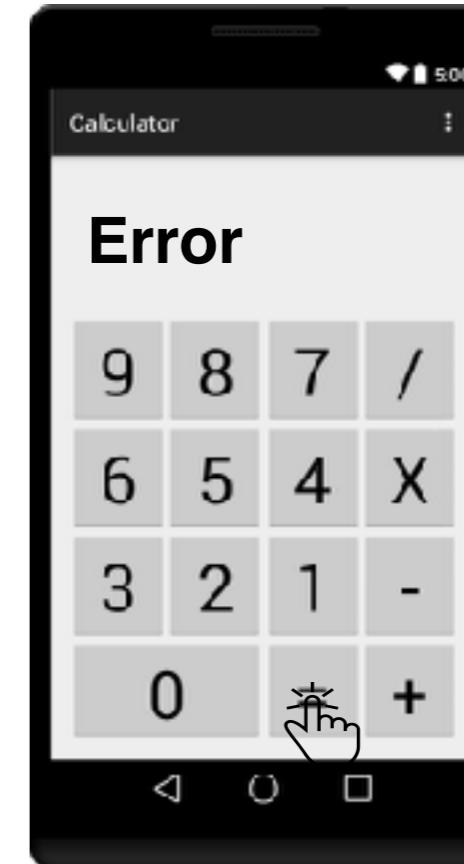
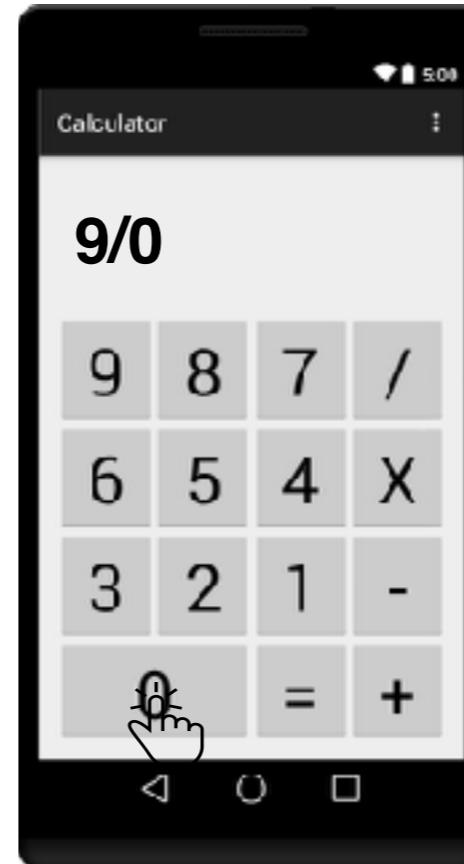
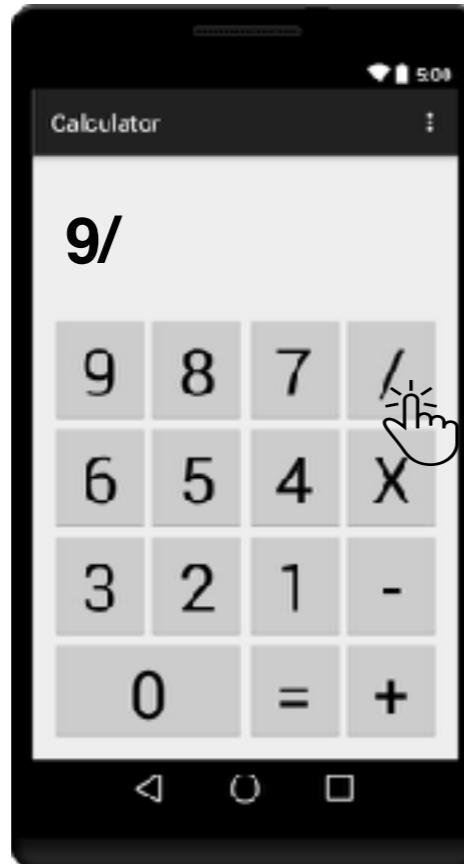
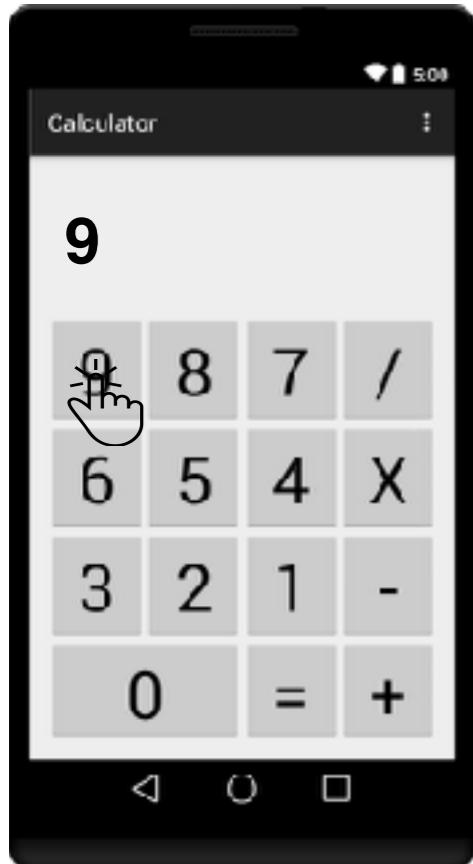
- (1) Press nine
- (2) Press divide
- (3) Press zero
- (4) Press equal
- (5) Check message



Running 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

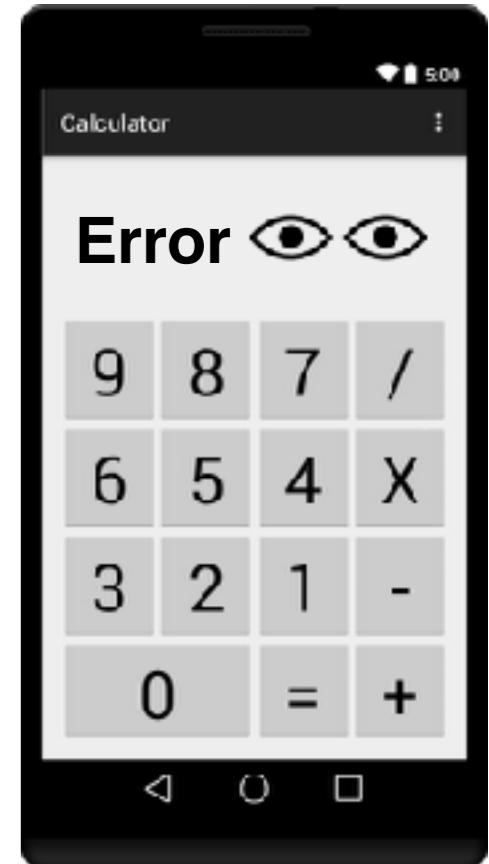
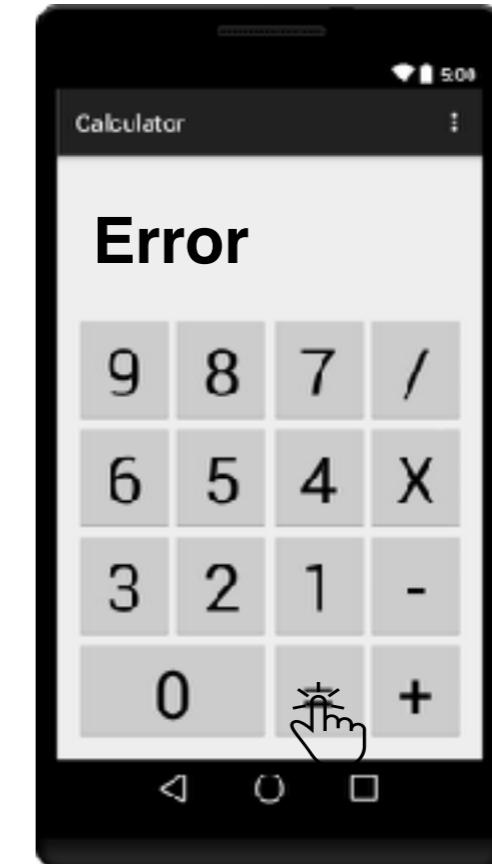
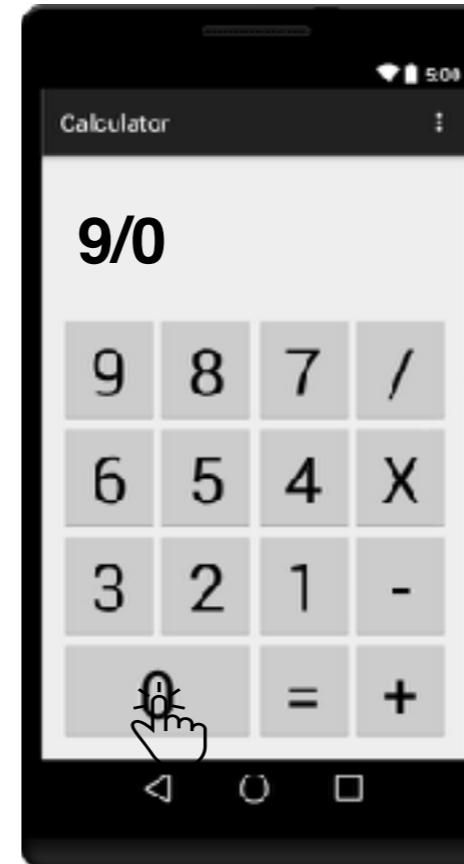
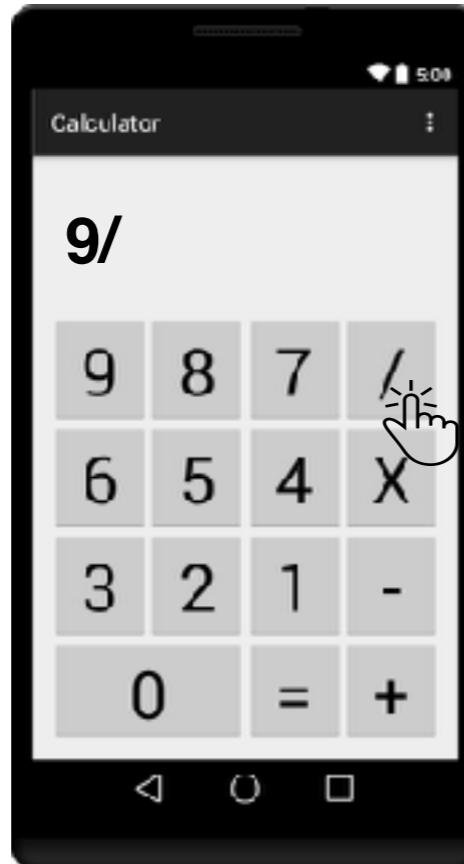
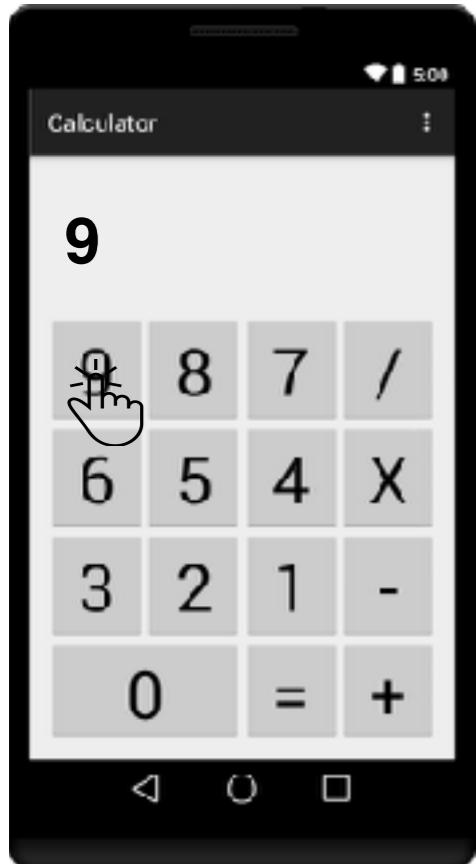


Simple

Running 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

Running 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 }
```

Running 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 }
```

Running 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 }
```

Running 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

Running Example: Automated Testing

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

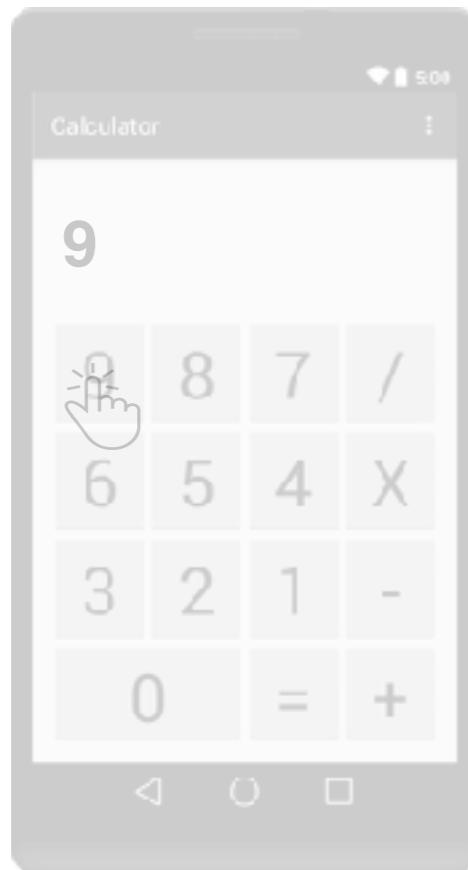
- (1) Press nine
- (2) Press divide
- (3) Press zero
- (4) Press equal
- (5) Check message

```
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

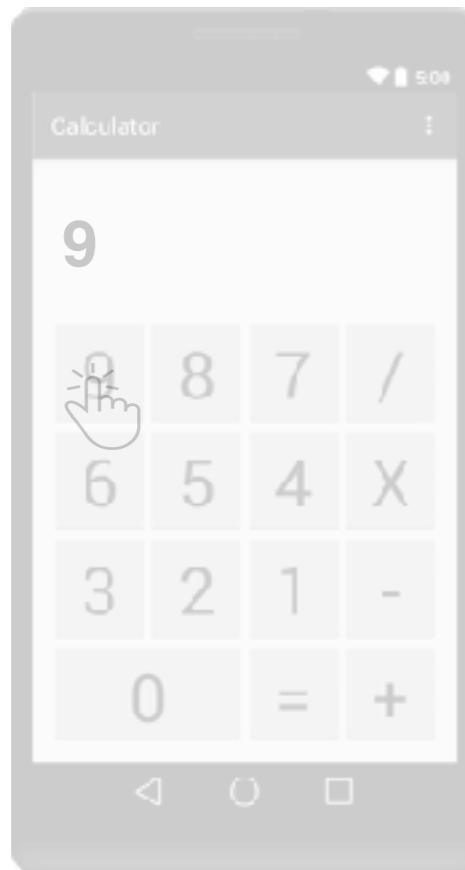
Complex

Running Example: Barista



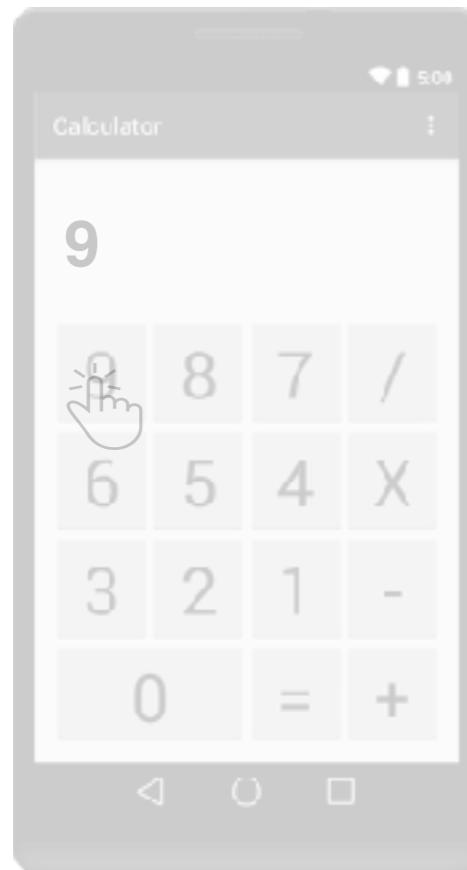
```
1 public void divisionByZero() {  
    8 }
```

Running Example: Barista



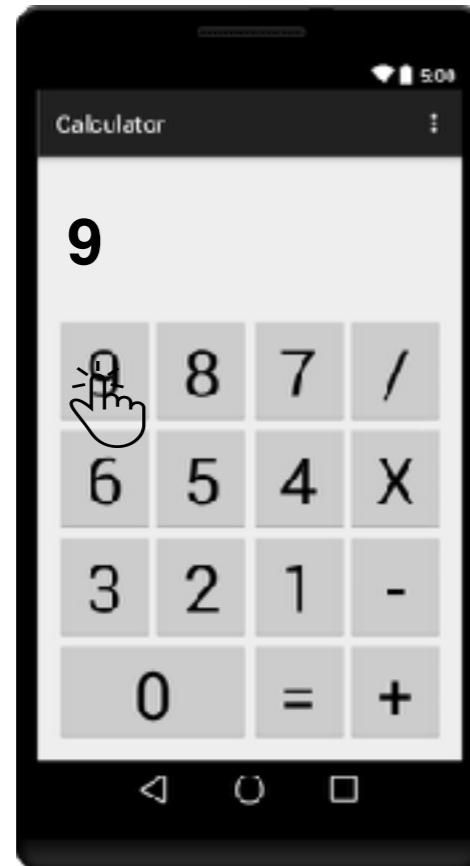
```
1 public void divisionByZero() {  
    8 }
```

Running Example: Barista



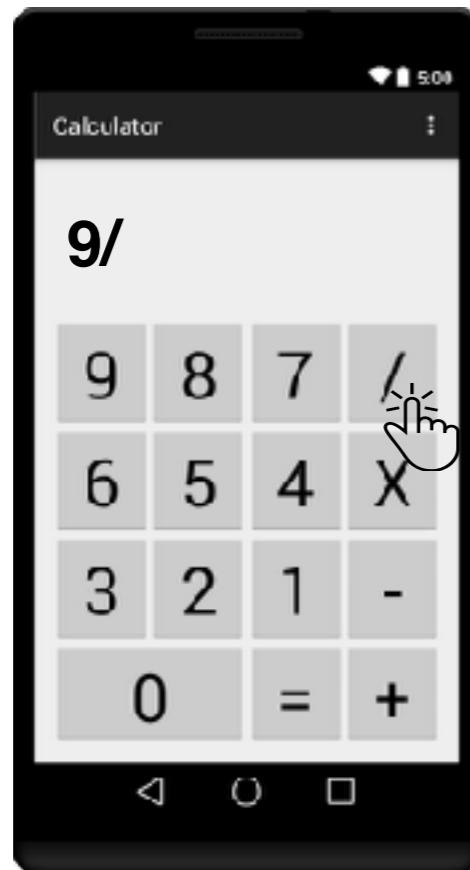
```
1 public void divisionByZero() {  
    8 }
```

Running Example: Barista



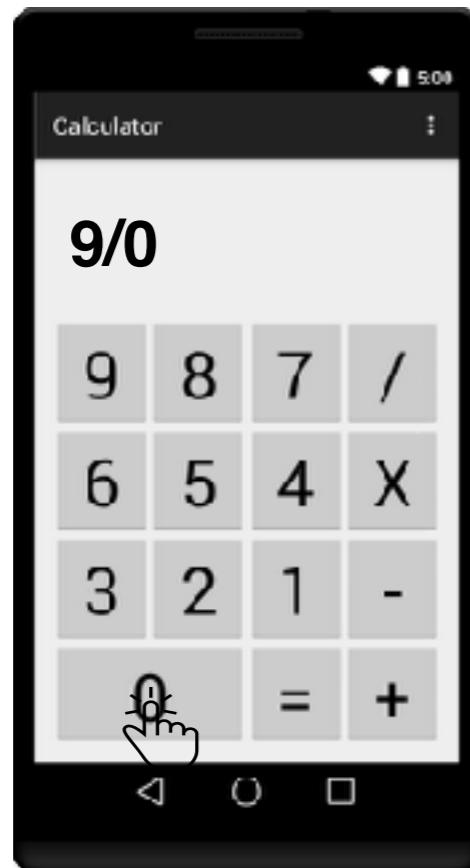
```
1 public void divisionByZero() {  
2     onView(withXPath("//...")).perform(click());  
8 }
```

Running Example: Barista



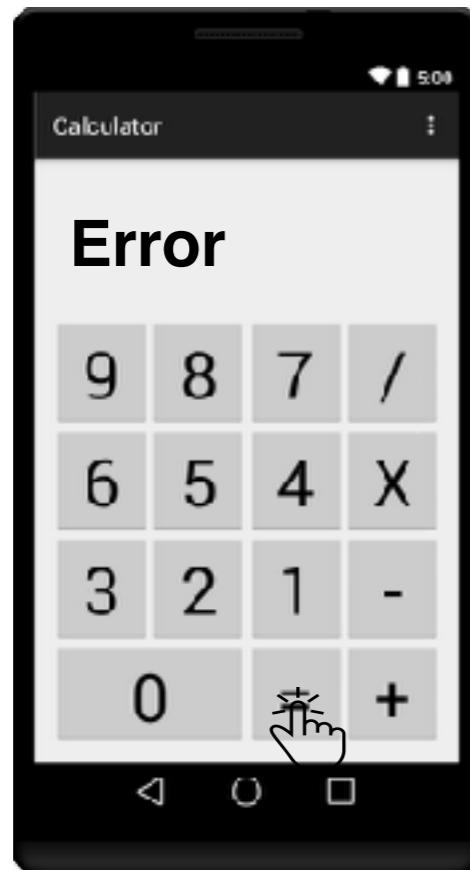
```
1 public void divisionByZero() {  
2     onView(withXPath("//...")).perform(click());  
3     onView(withId(R.id.div)).perform(click());  
8 }
```

Running Example: Barista



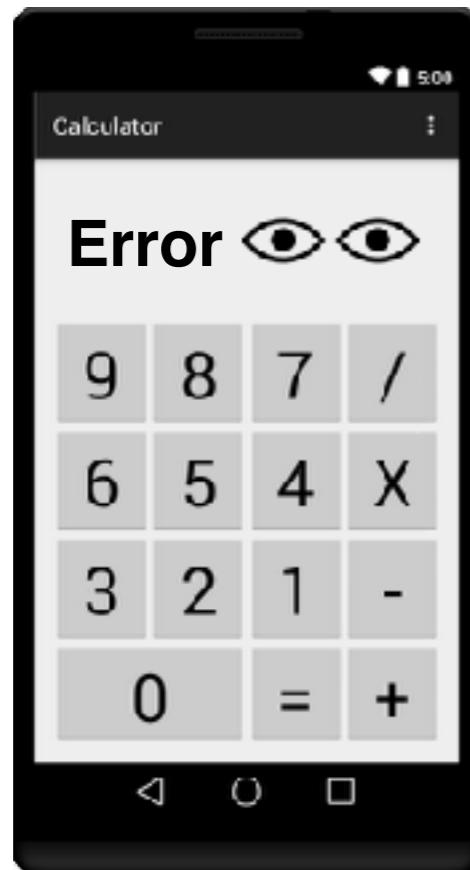
```
1 public void divisionByZero() {  
2     onView(withXPath("//...")).perform(click());  
3     onView(withId(R.id.div)).perform(click());  
4     onView(withXPath("//...")).perform(click());  
8 }
```

Running Example: Barista



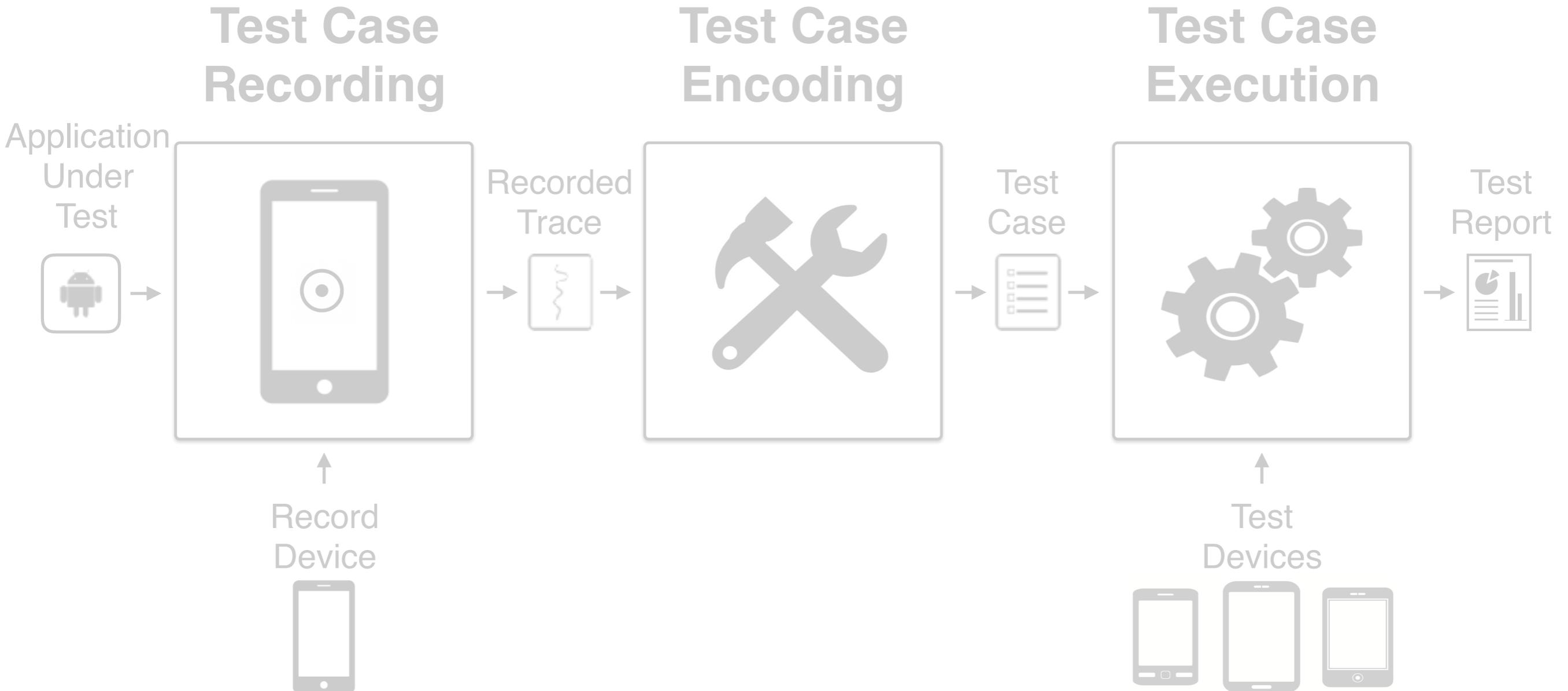
```
1 public void divisionByZero() {  
2     onView(withXPath("//...")).perform(click());  
3     onView(withId(R.id.div)).perform(click());  
4     onView(withXPath("//...")).perform(click());  
5     onView(withId(R.id.eq)).perform(click());  
8 }
```

Running Example: Barista

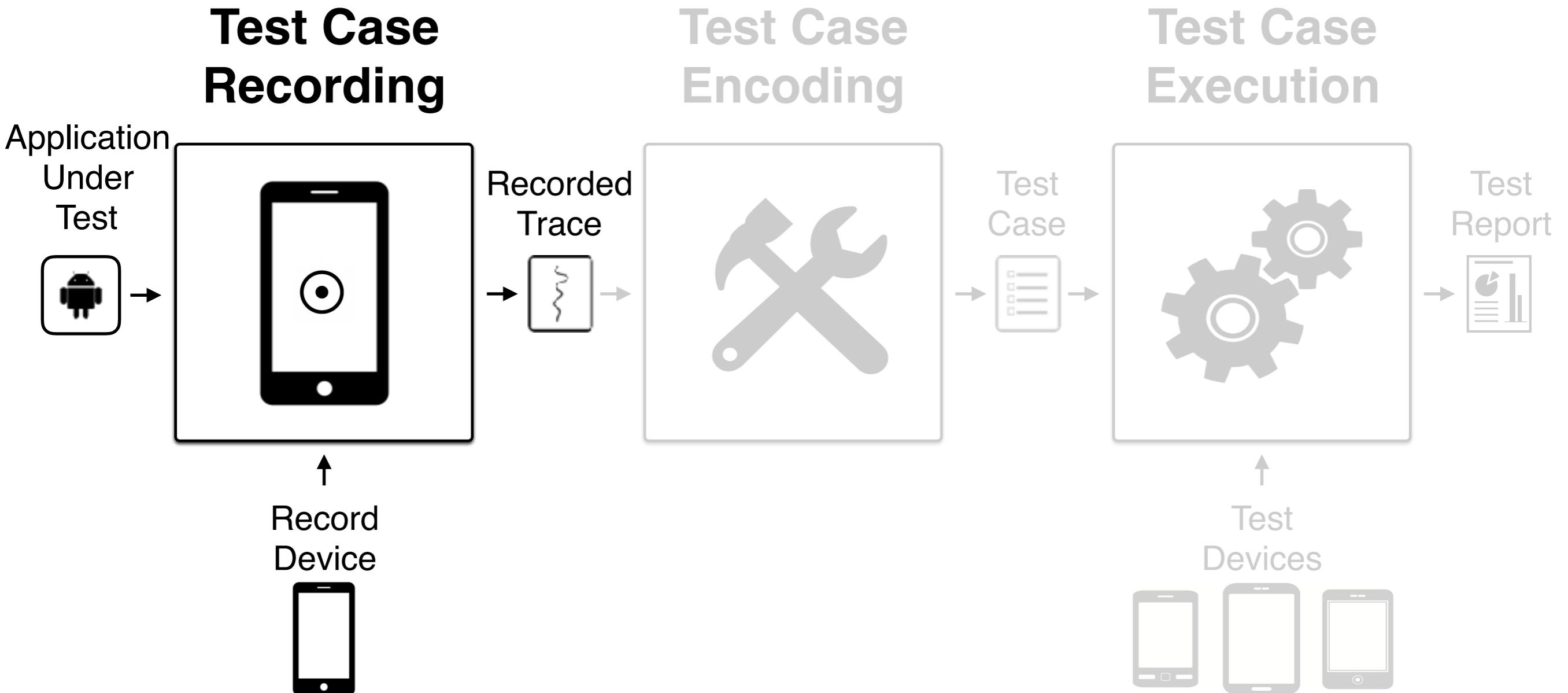


```
1 public void divisionByZero() {  
2     onView(withXPath("//...")).perform(click());  
3     onView(withId(R.id.div)).perform(click());  
4     onView(withXPath("//...")).perform(click());  
5     onView(withId(R.id.eq)).perform(click());  
6     onView(withId(R.id.dsp))  
7         .check(matches(withText("Error"))));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

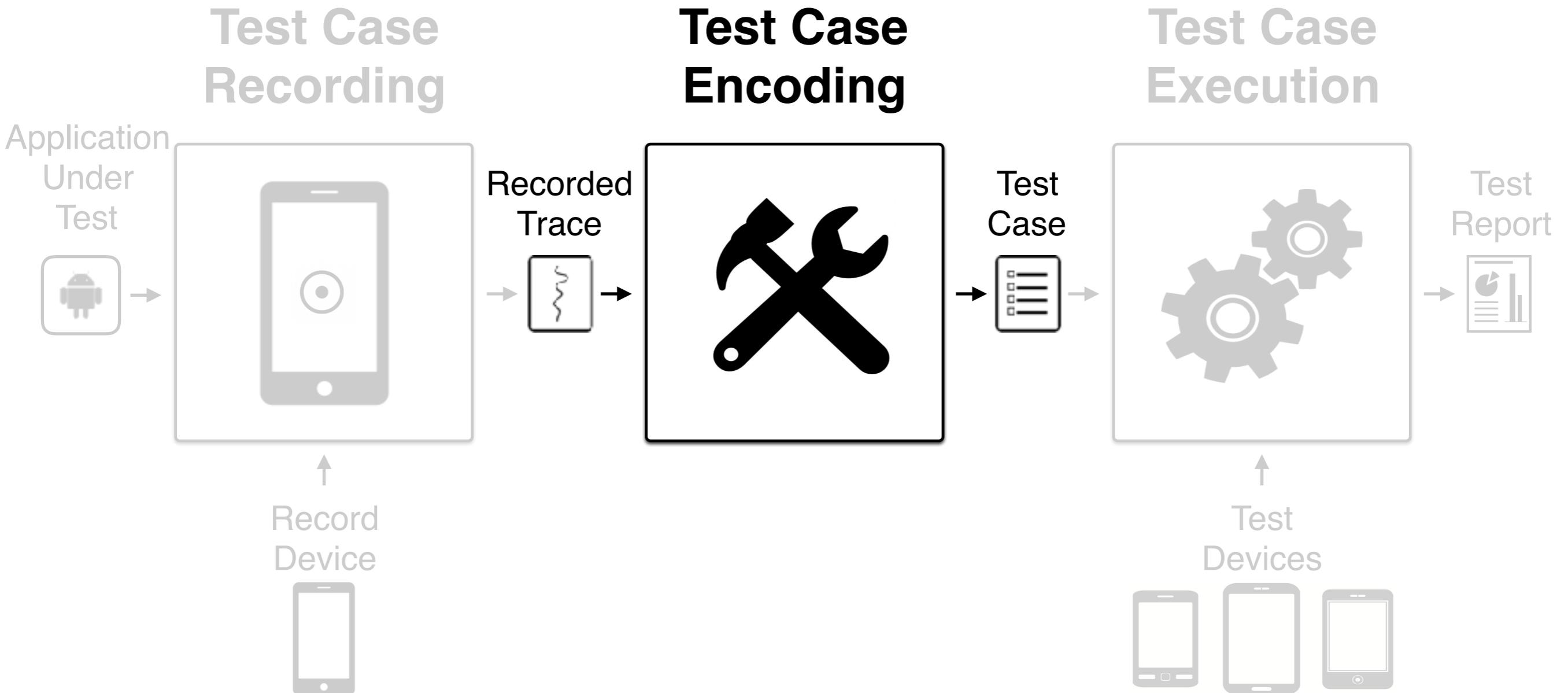
Barista Overview



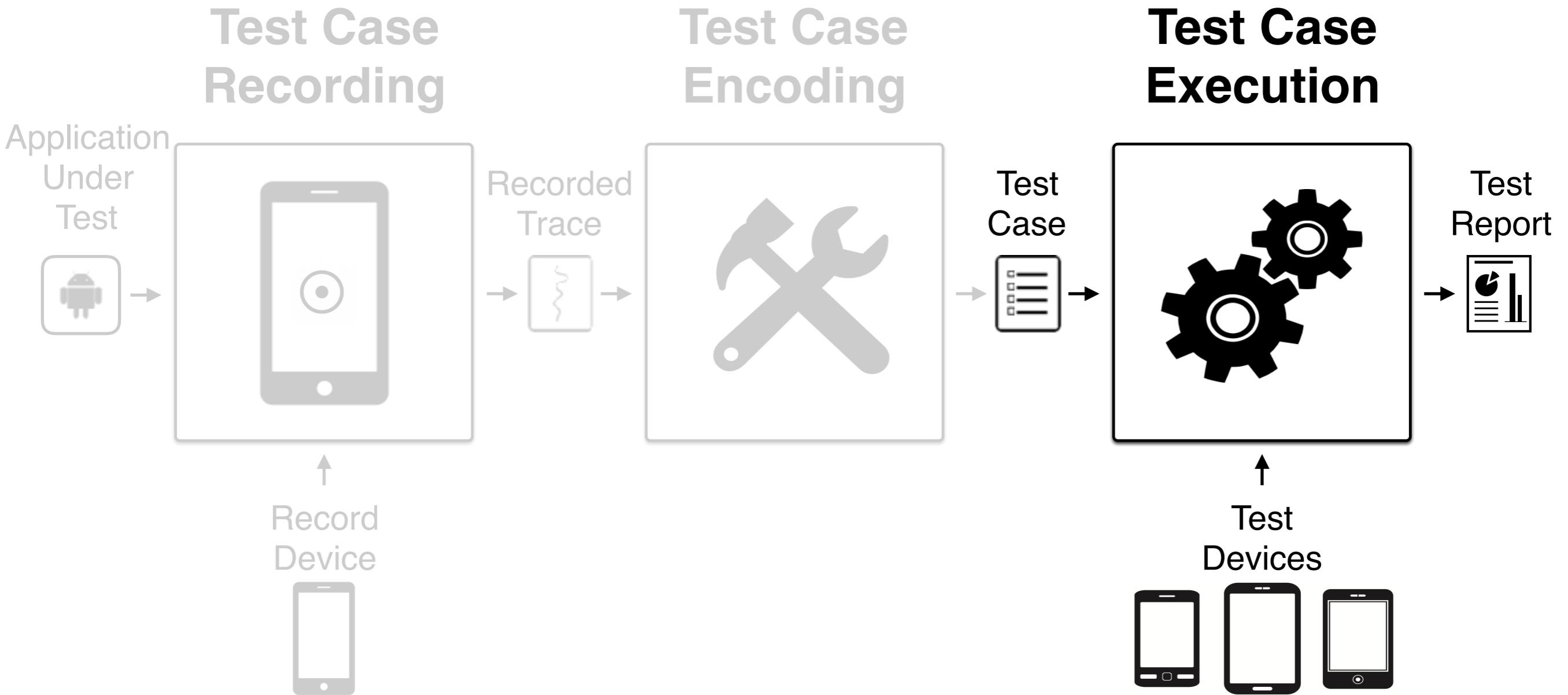
Barista Overview



Barista Overview



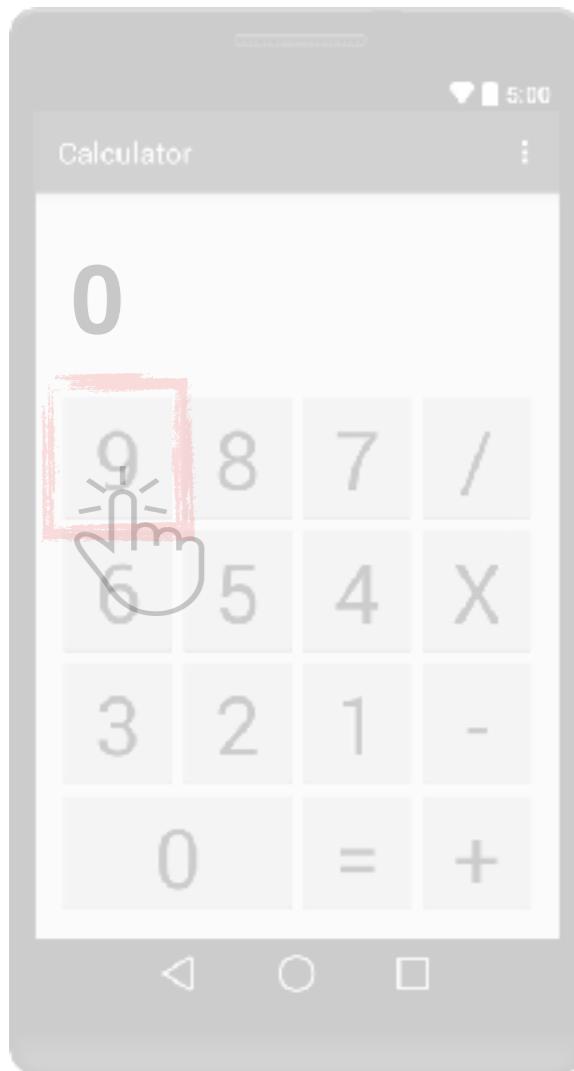
Barista Overview



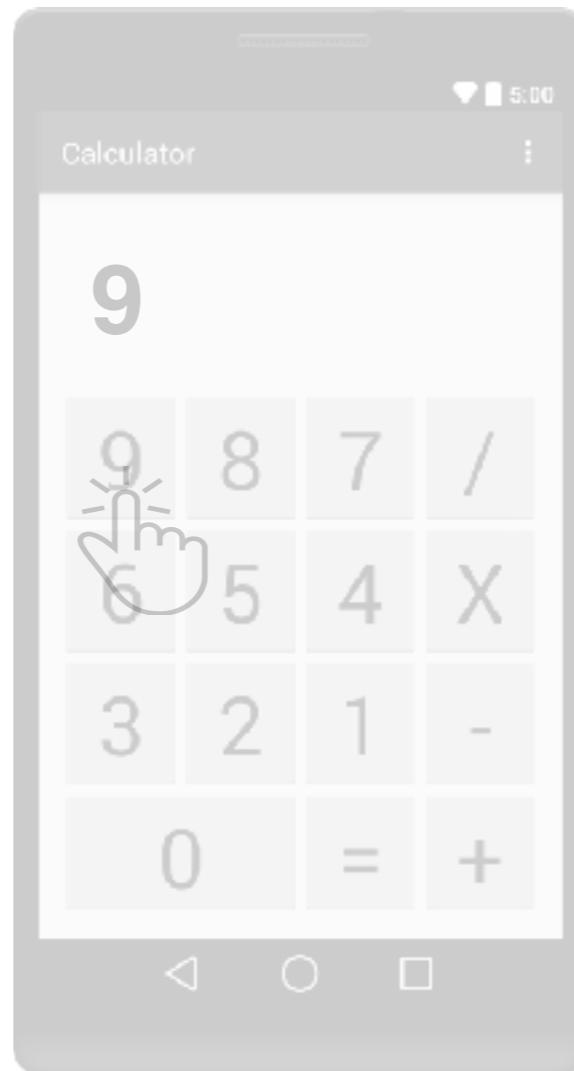
Test Case Recording

Objective: record actions performed on the AUT

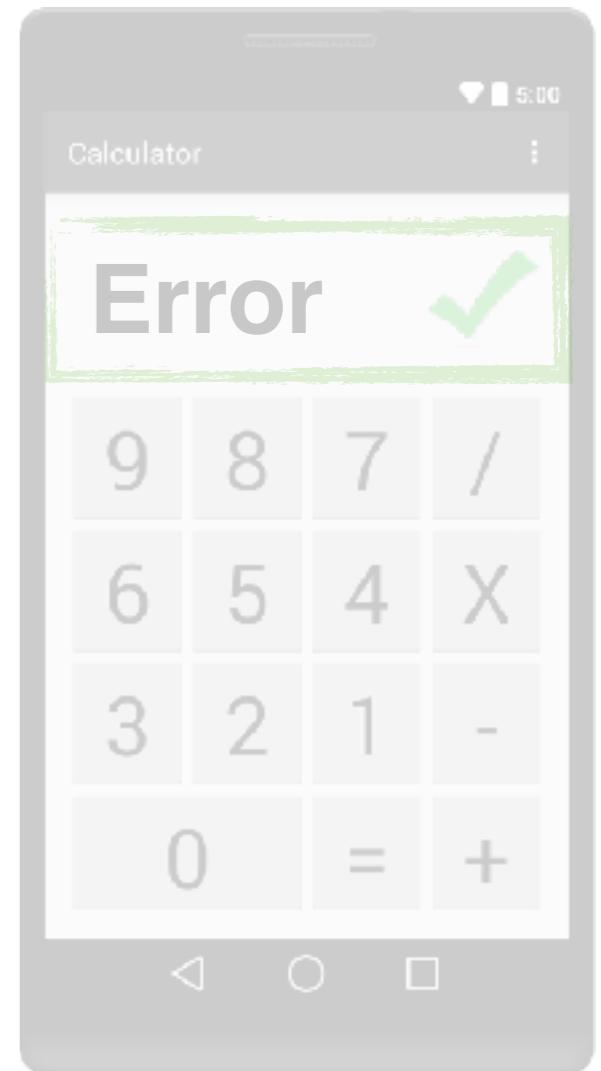
Elements



Actions



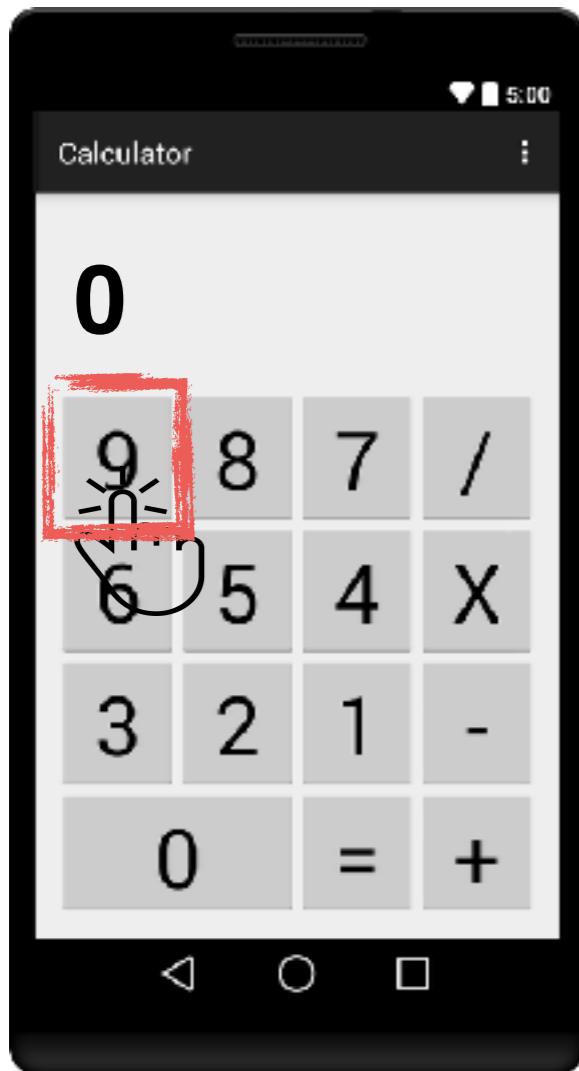
Oracles



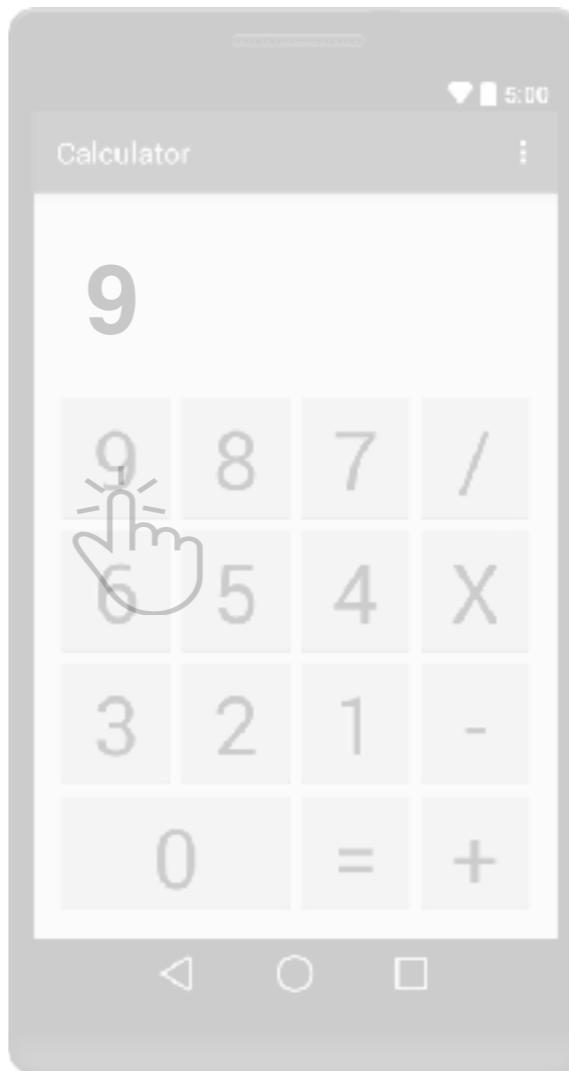
Test Case Recording

Objective: record actions performed on the AUT

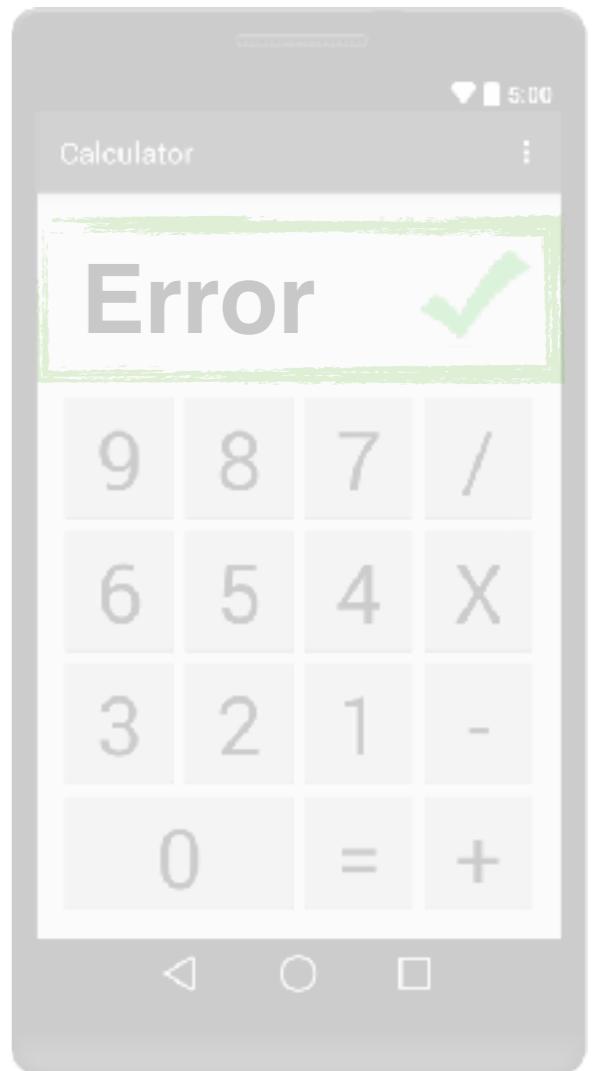
Elements



Actions



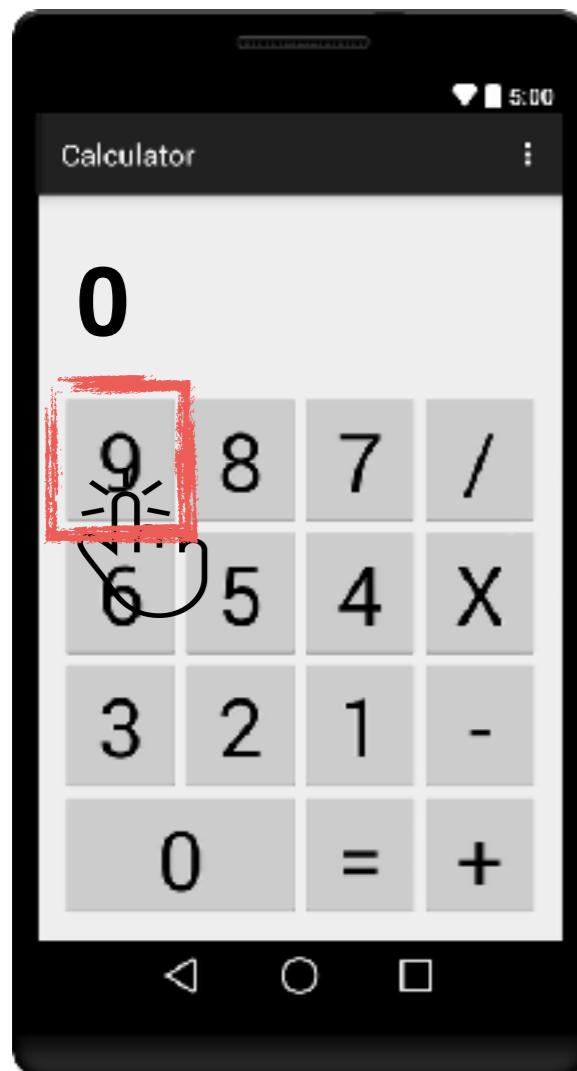
Oracles



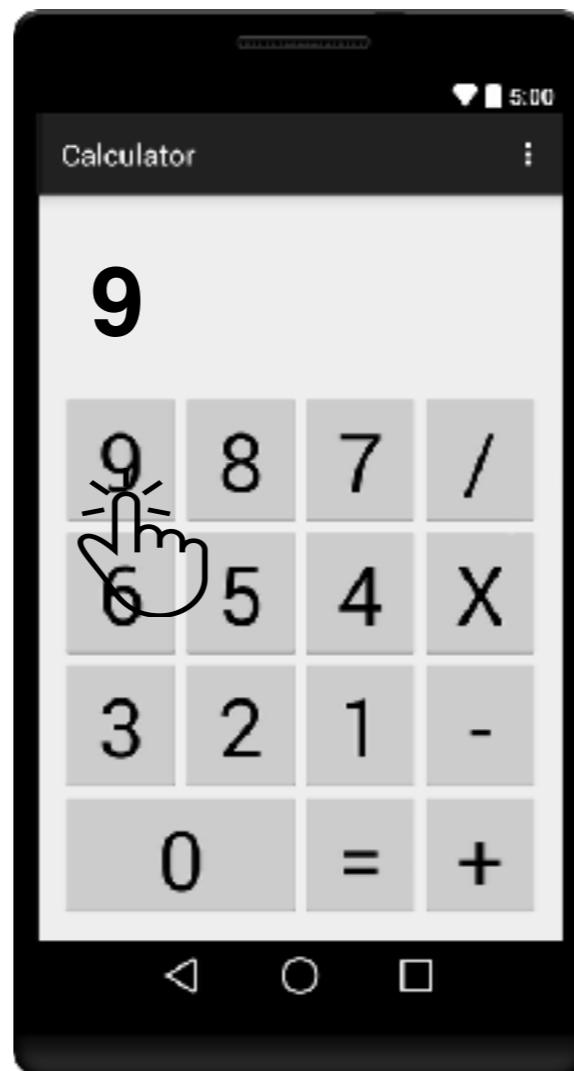
Test Case Recording

Objective: record actions performed on the AUT

Elements



Actions



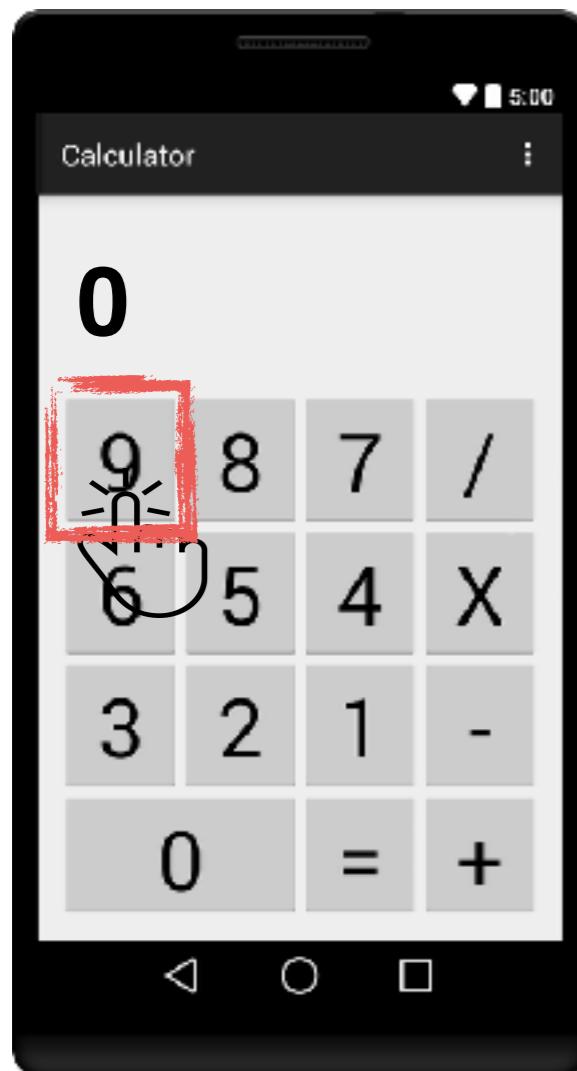
Oracles



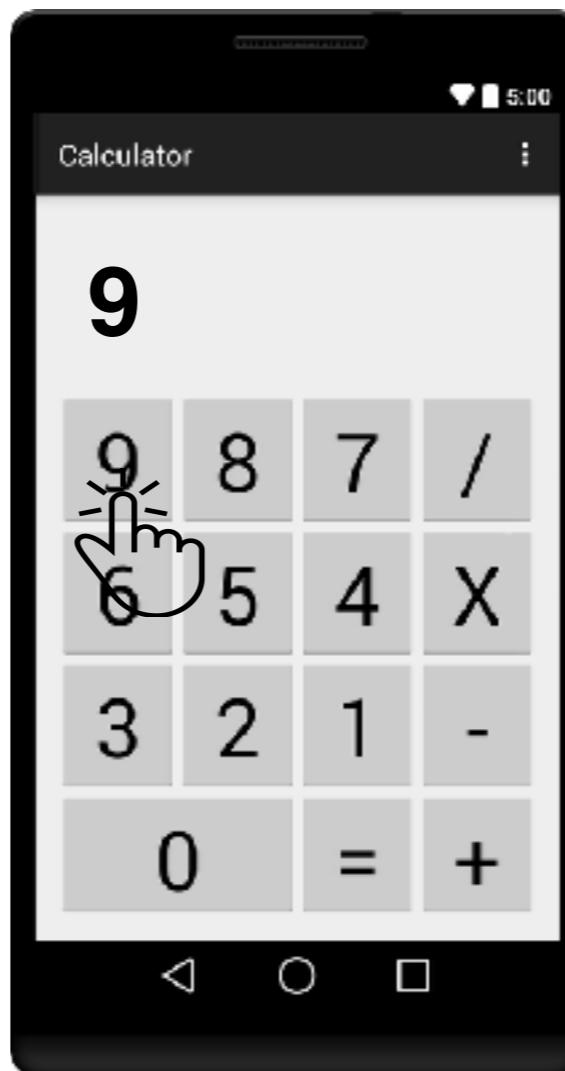
Test Case Recording

Objective: record actions performed on the AUT

Elements



Actions

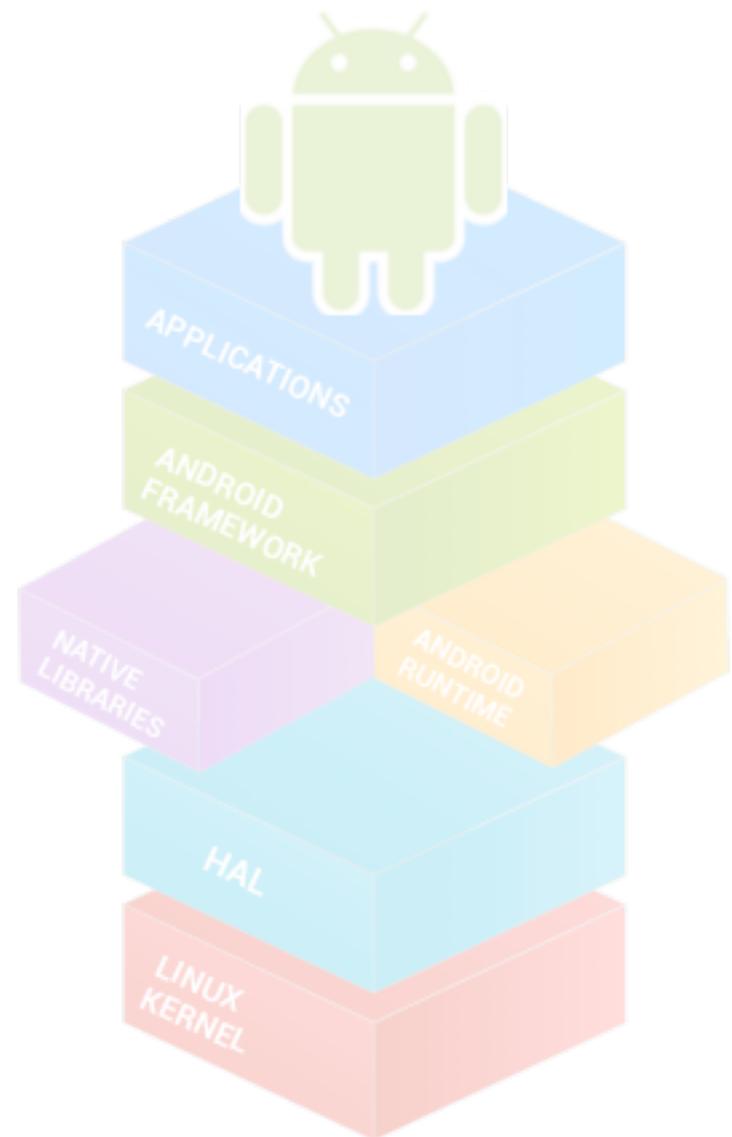


Oracles

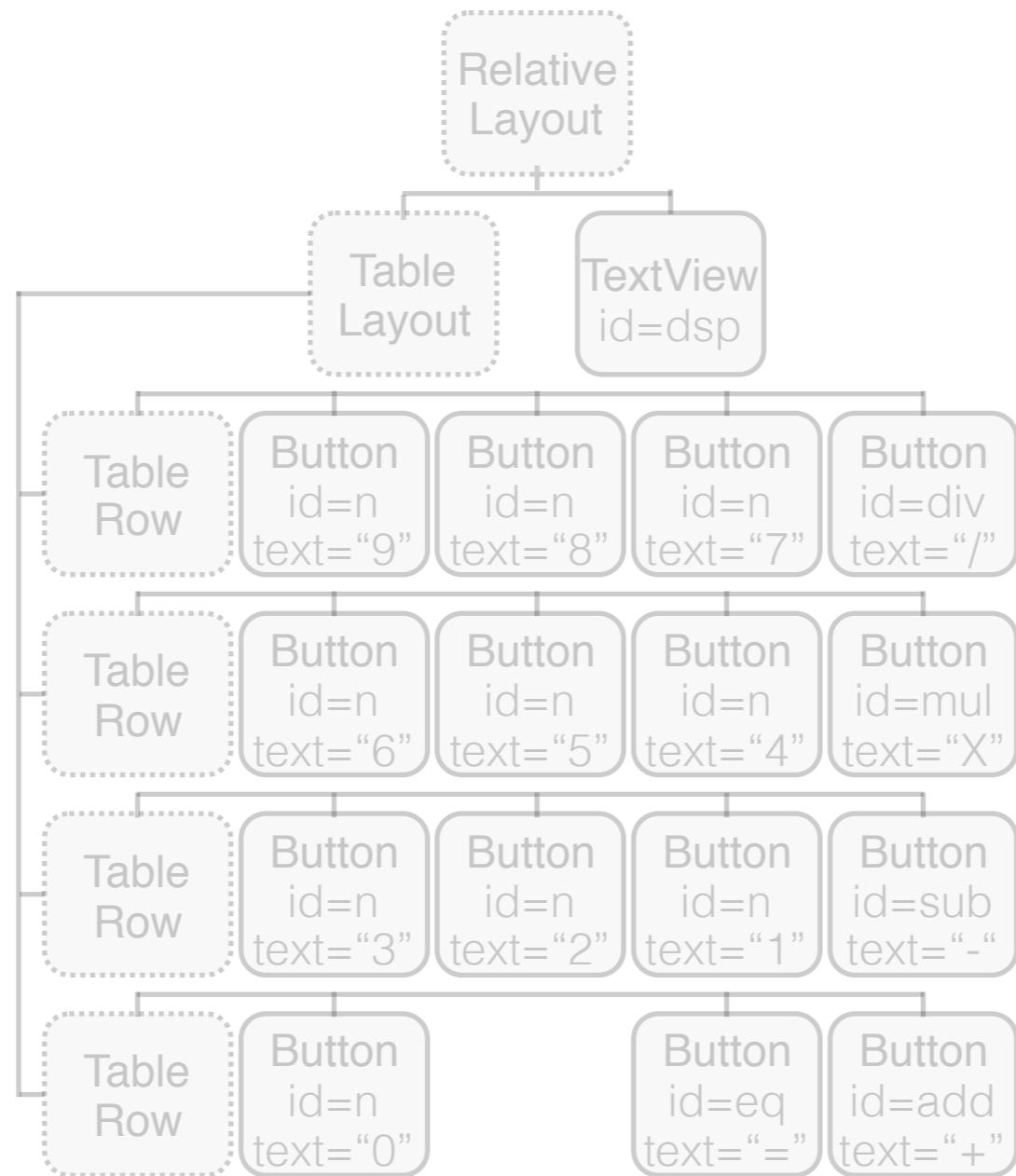


Test Case Recording: Elements

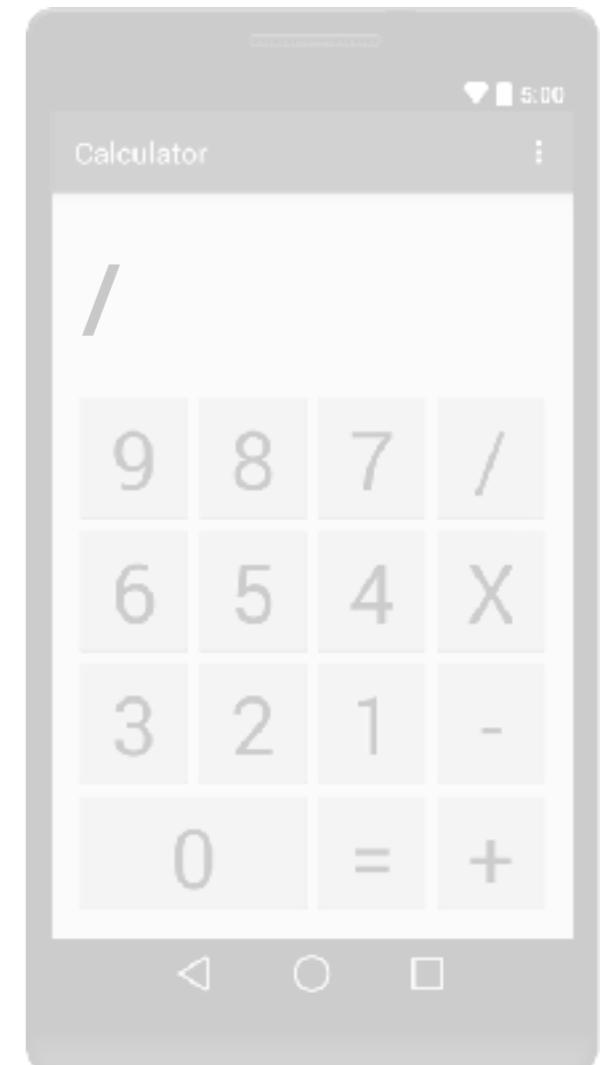
Accessibility



UI Hierarchy



UI

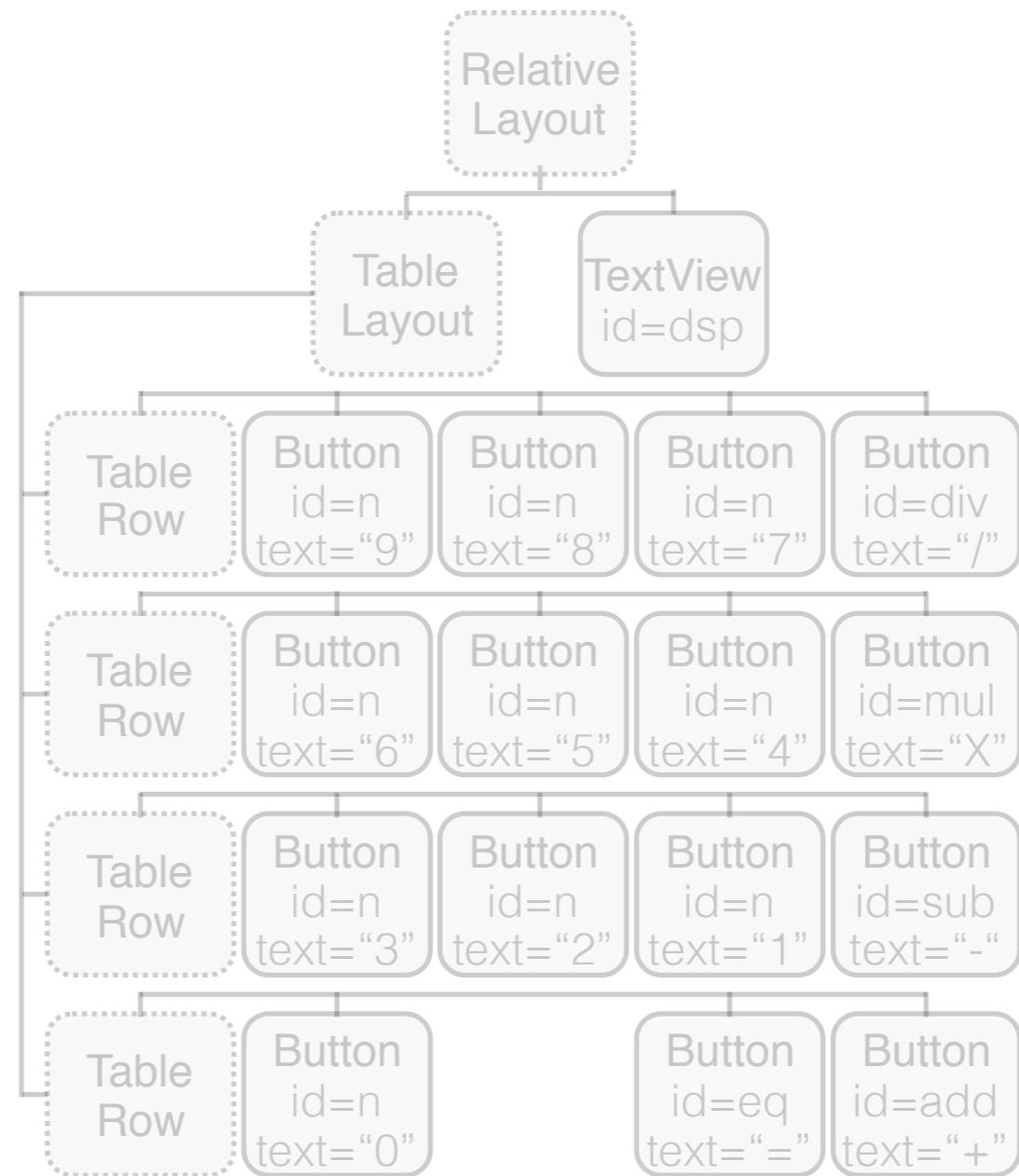


Test Case Recording: Elements

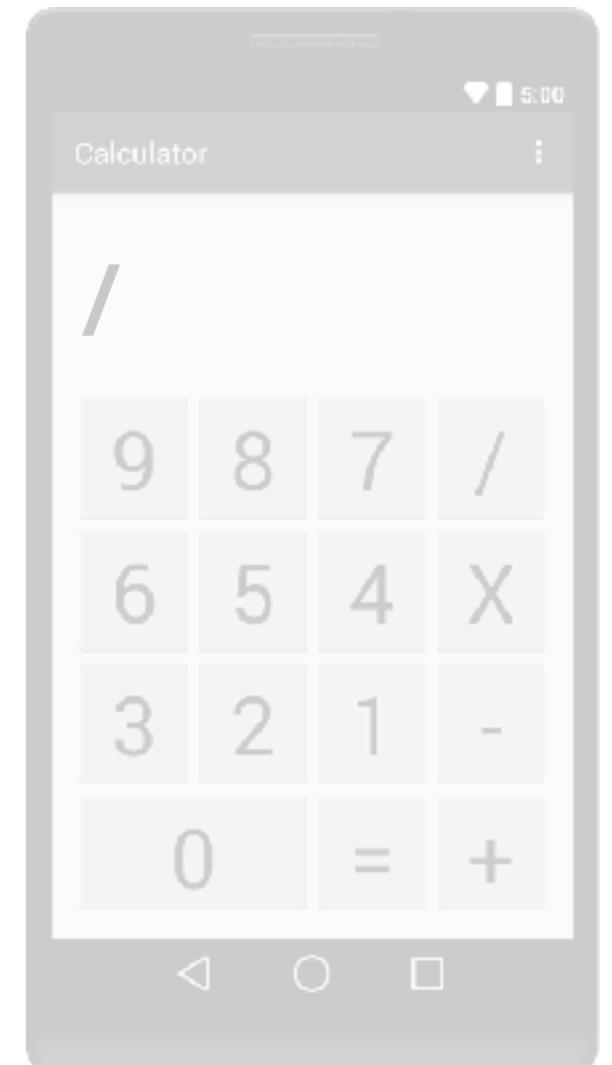
Accessibility



UI Hierarchy



UI

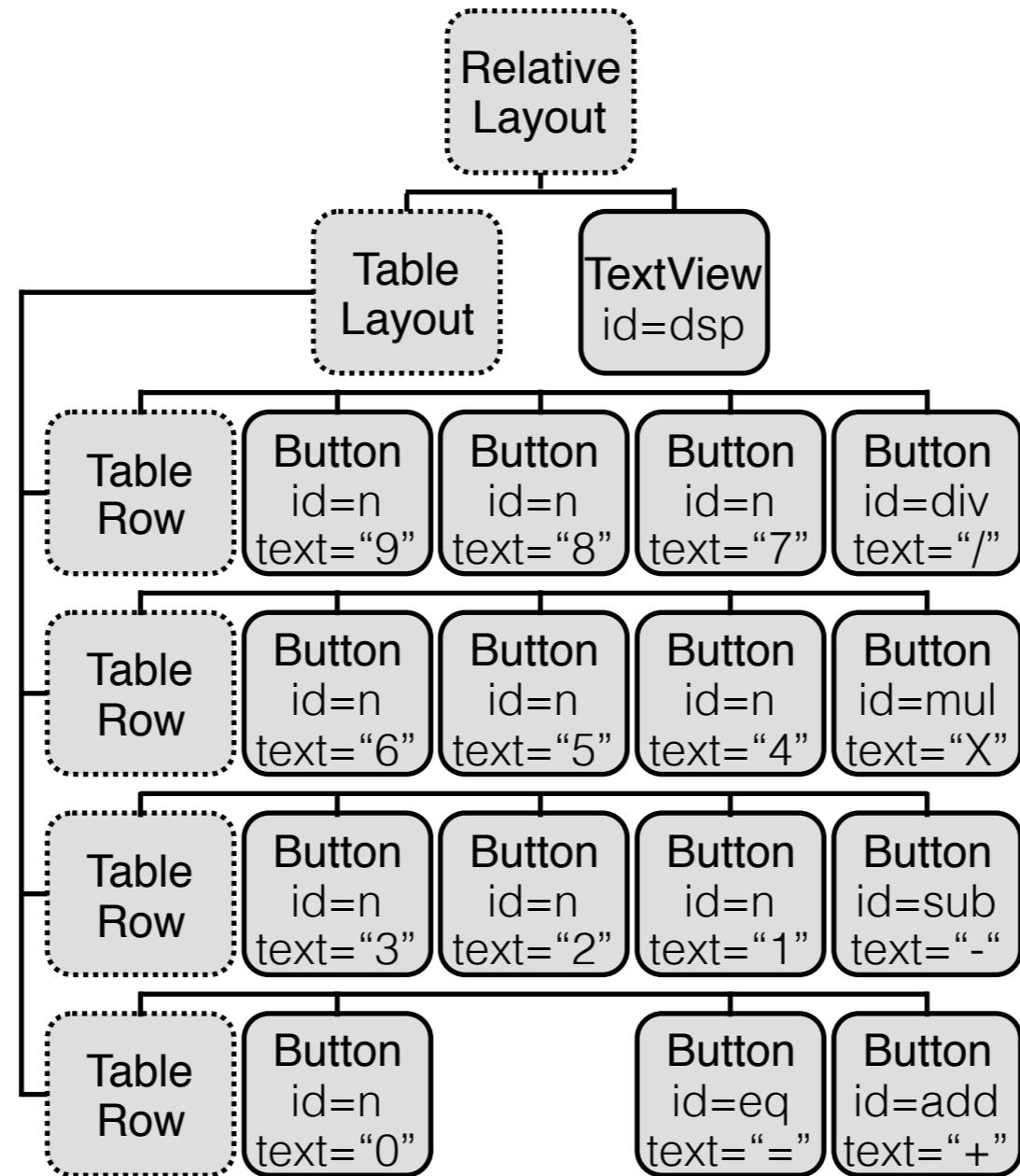


Test Case Recording: Elements

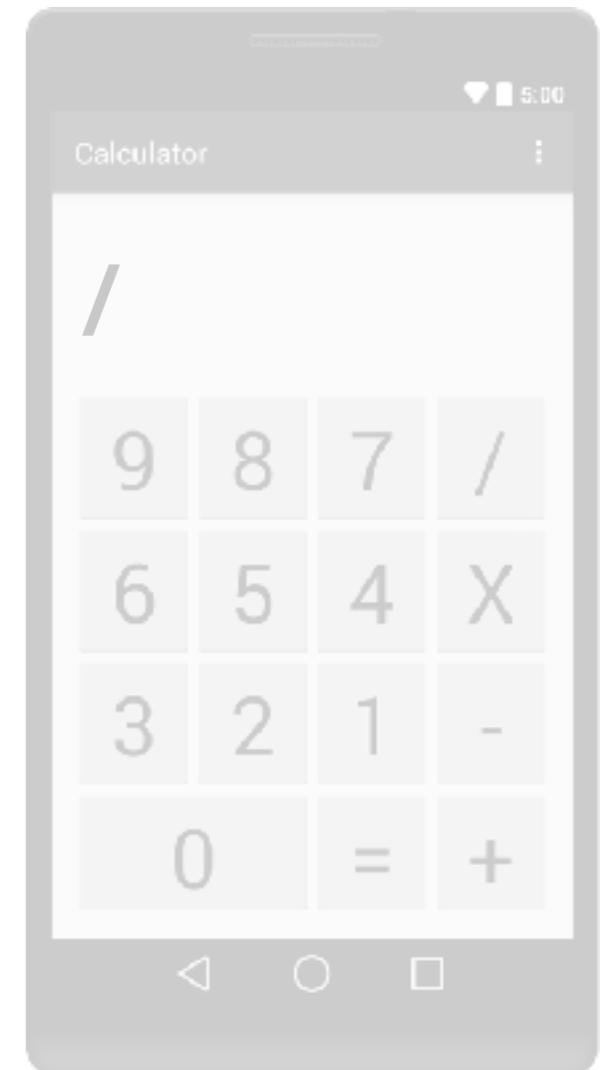
Accessibility



UI Hierarchy



UI

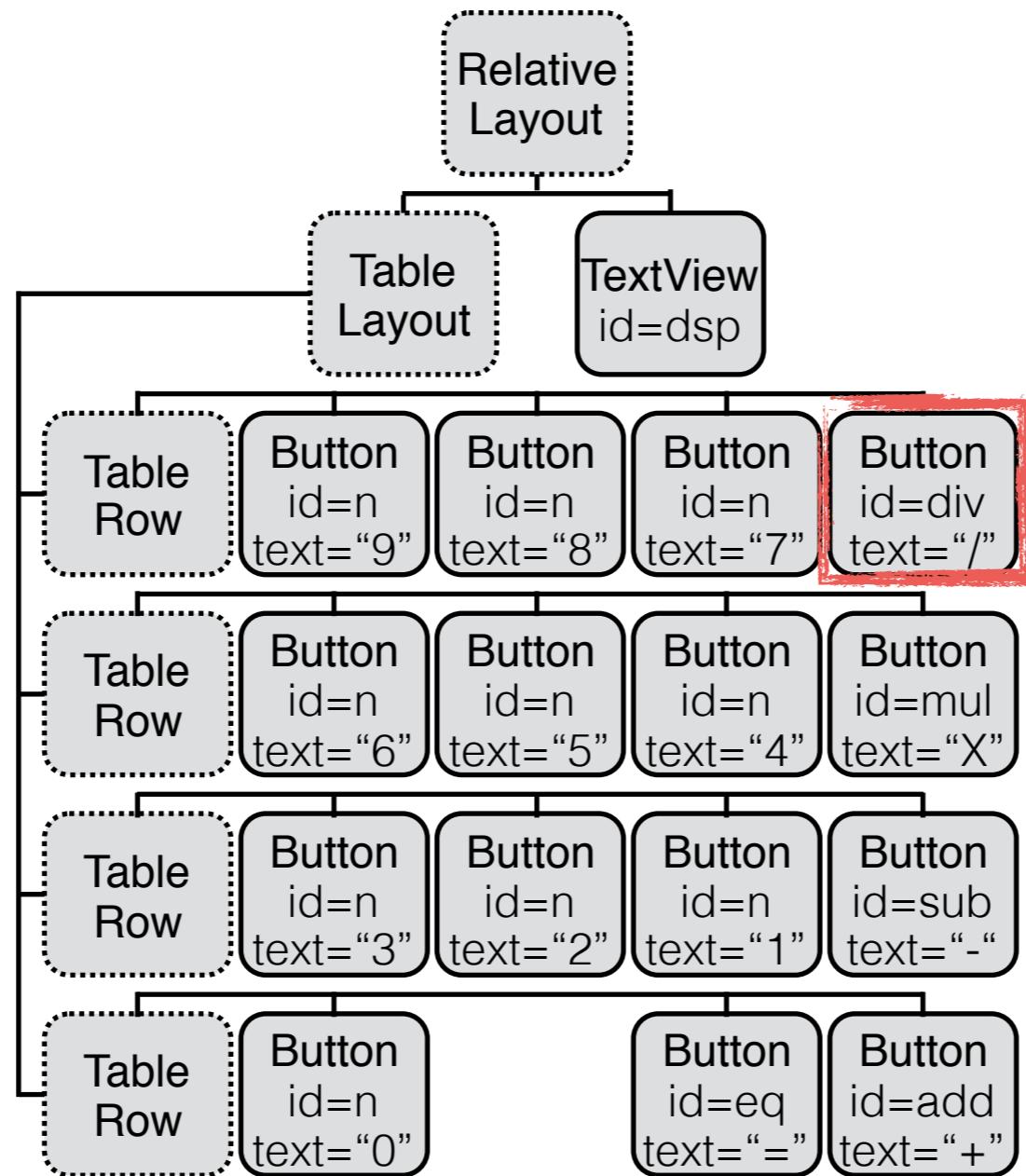


Test Case Recording: Elements

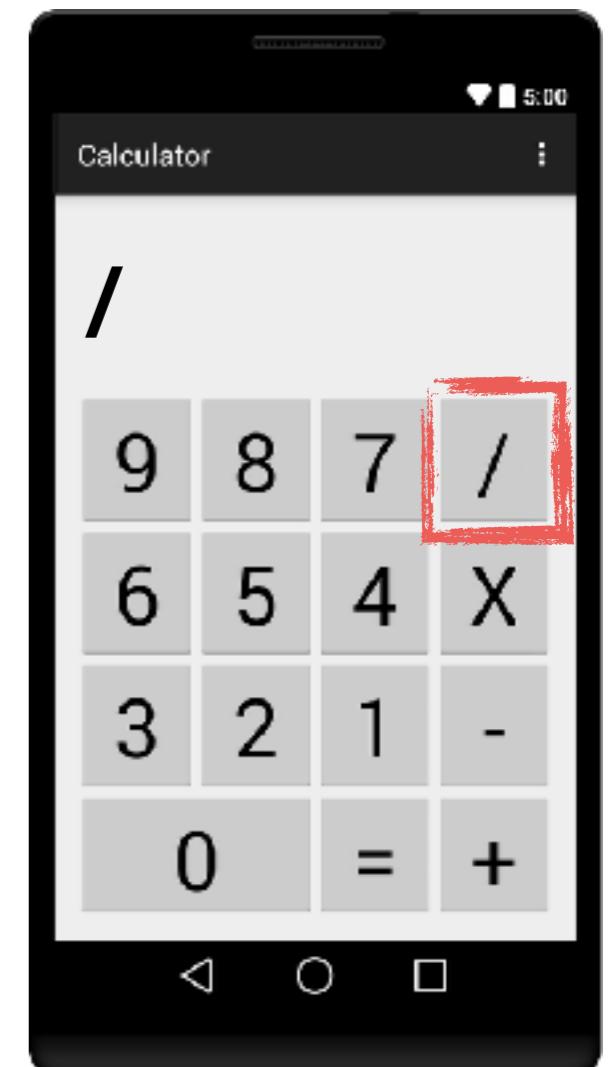
Accessibility



UI Hierarchy



UI



Test Case Recording: Elements

Types

Resource ID

div

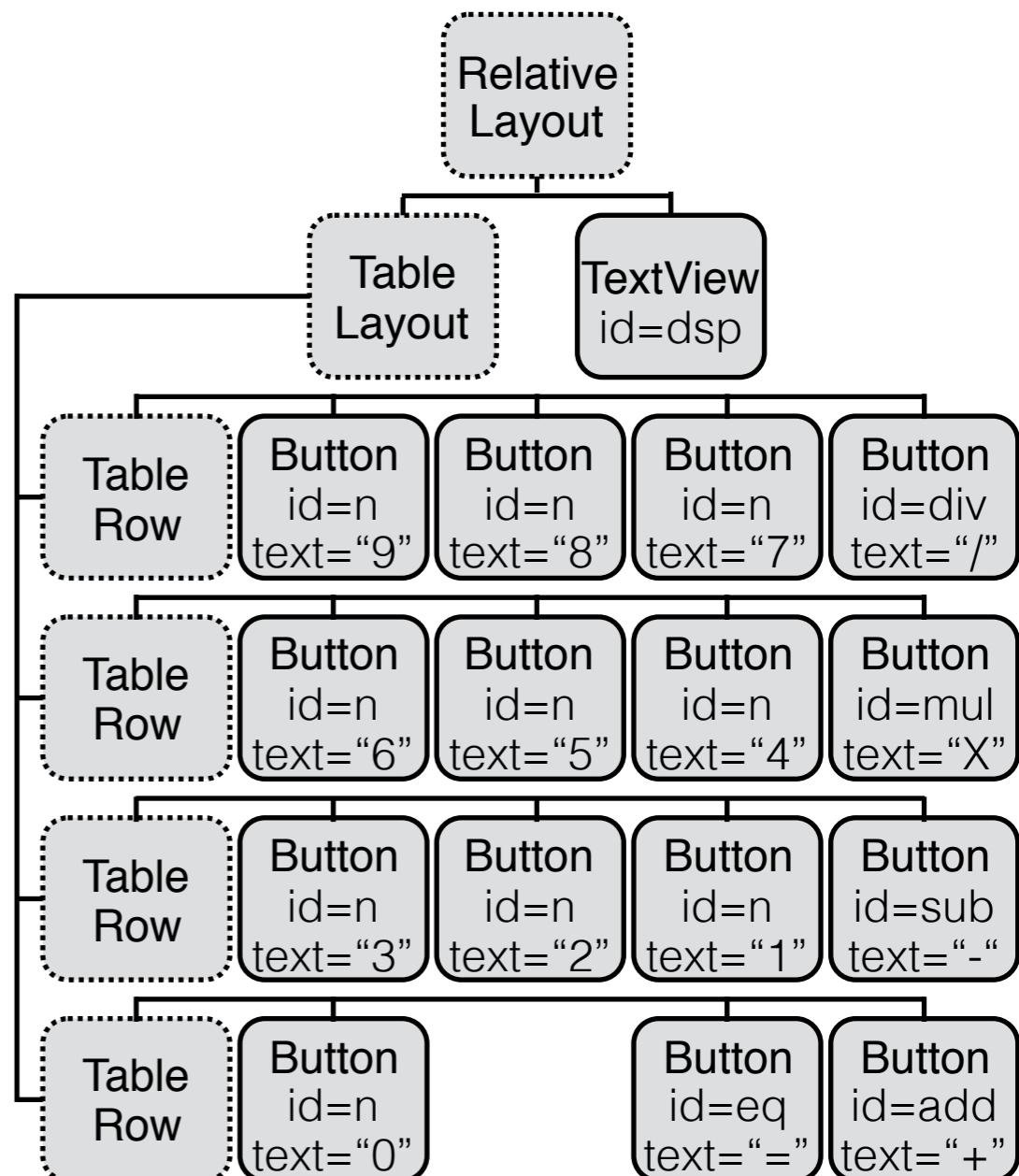
XPath

/RelativeLayout
/TableLayout[1]
/TableRow[4]
/Button[1]

Property-based

Class, Text

UI Hierarchy



Test Case Recording: Elements

Types

Resource ID

div

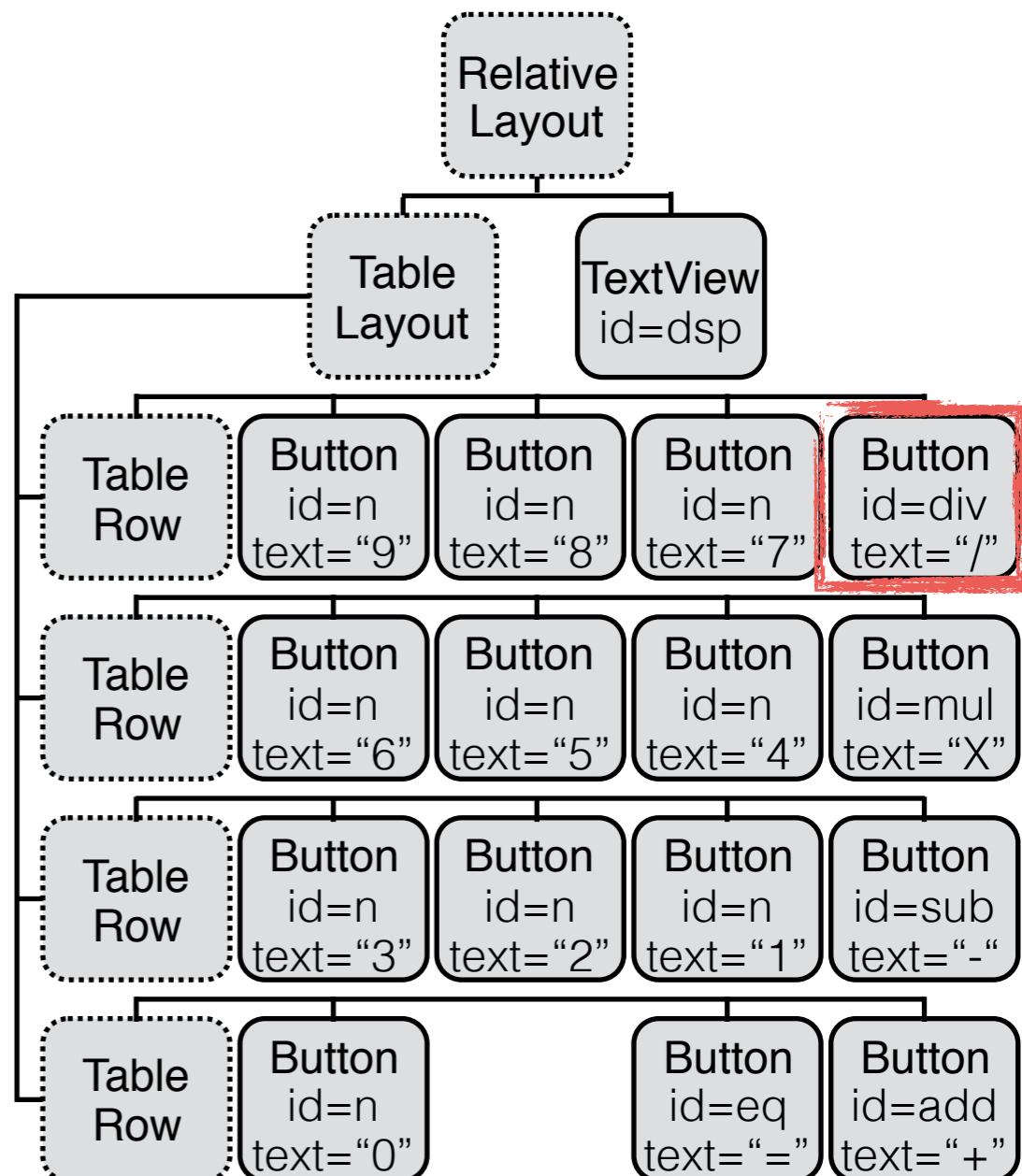
XPath

/RelativeLayout
/TableLayout[1]
/TableRow[4]
/Button[1]

Property-based

Class, Text

UI Hierarchy



Test Case Recording: Elements

Types

Resource ID

div

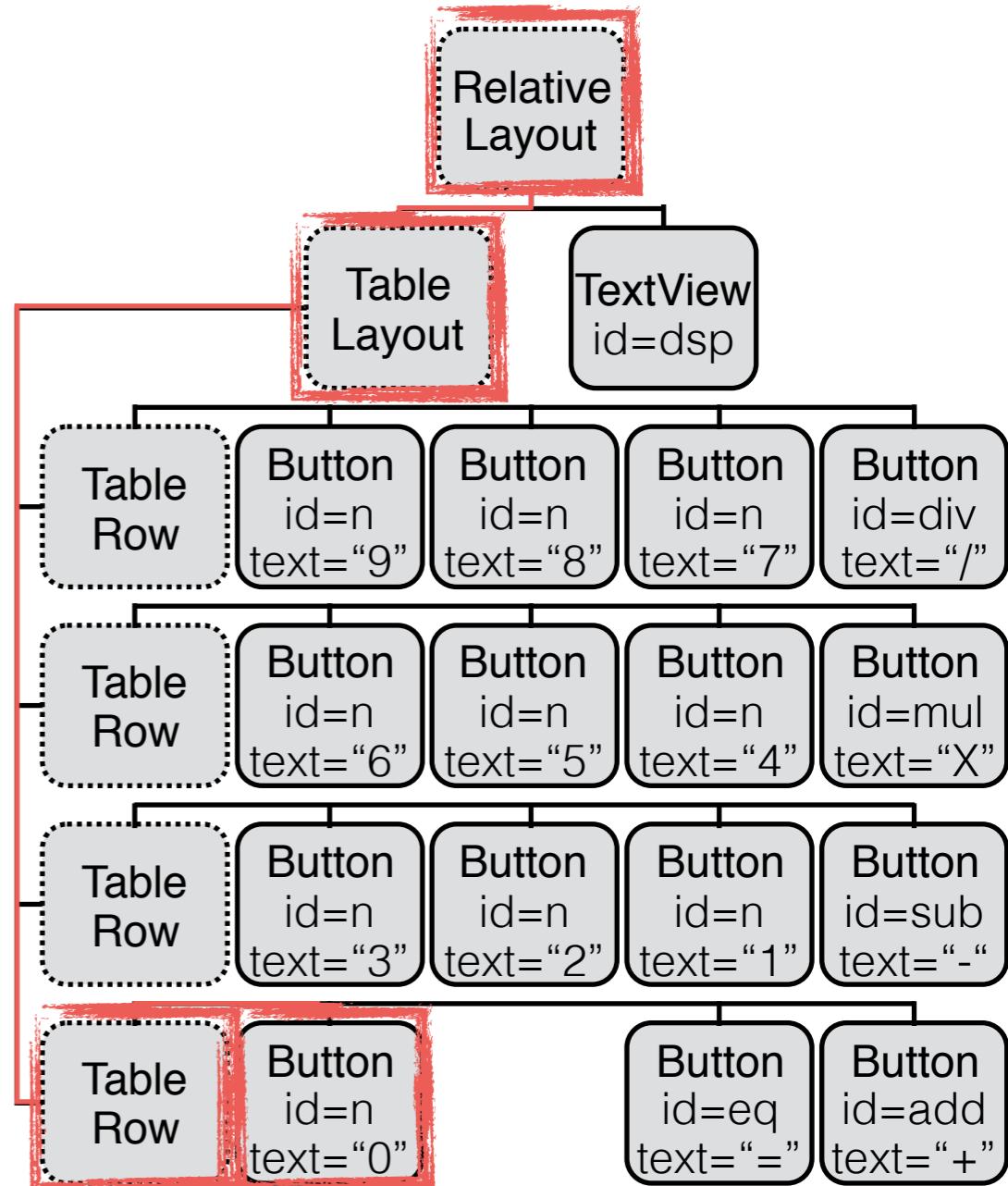
XPath

/RelativeLayout
/TableLayout[1]
/TableRow[4]
/Button[1]

Class, Text

Property-based

UI Hierarchy



Test Case Recording: Elements

Types

Resource ID

div

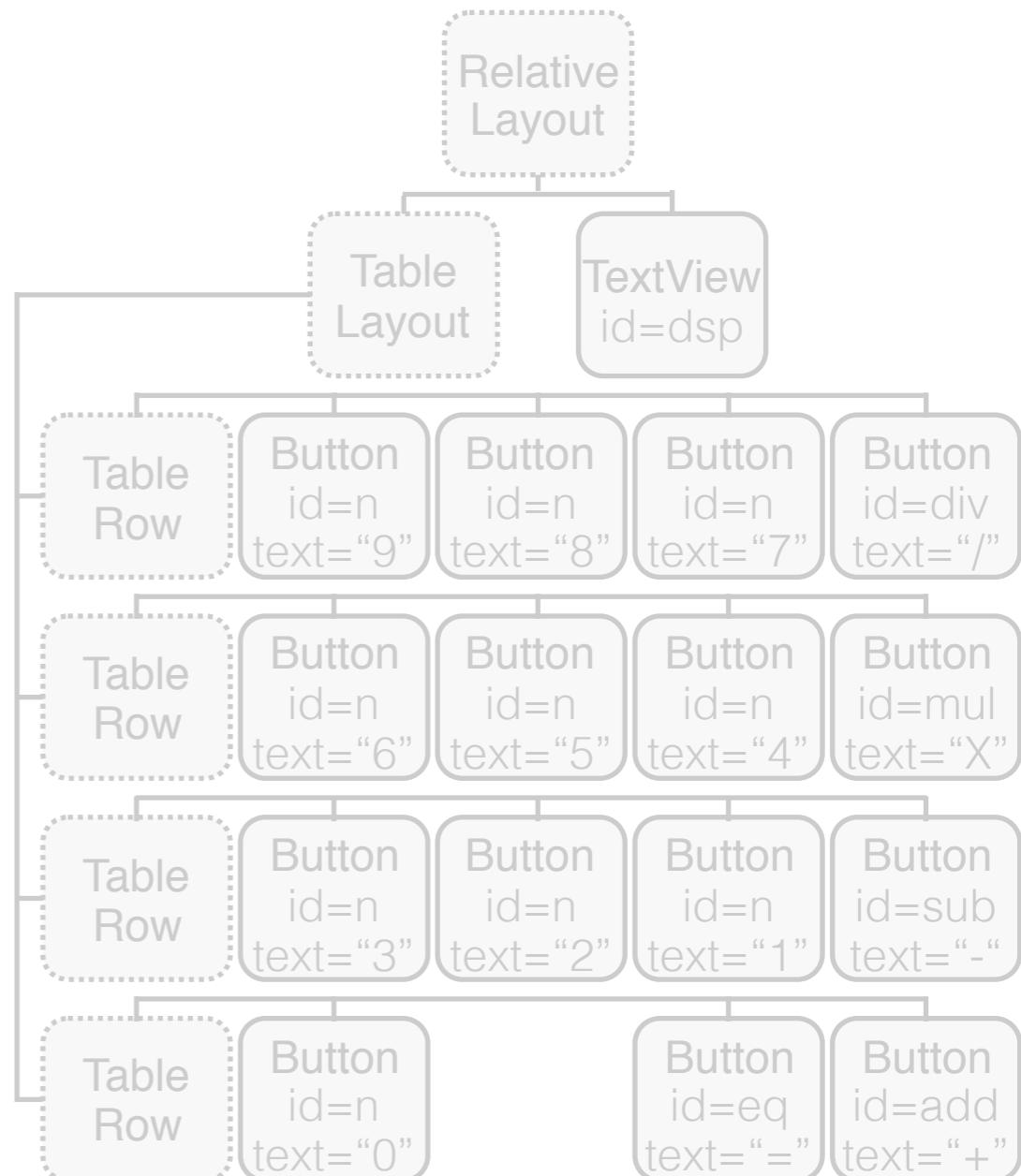
XPath

/RelativeLayout
/TableLayout[1]
/TableRow[4]
/Button[1]

Property-based

Class, Text

UI Hierarchy



Test Case Recording: Elements

Types

Resource ID

div

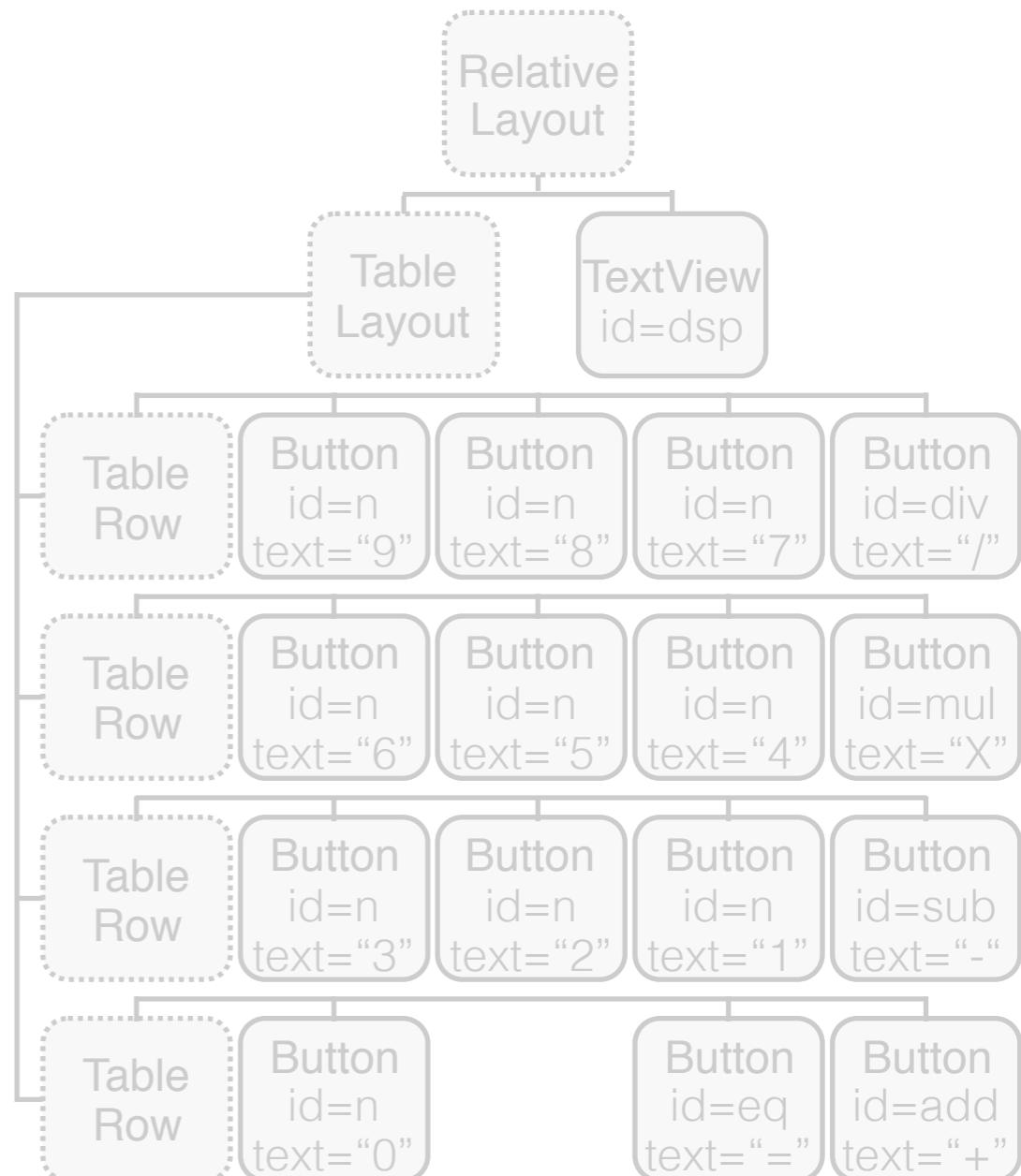
XPath

/RelativeLayout
/TableLayout[1]
/TableRow[4]
/Button[1]

Property-based

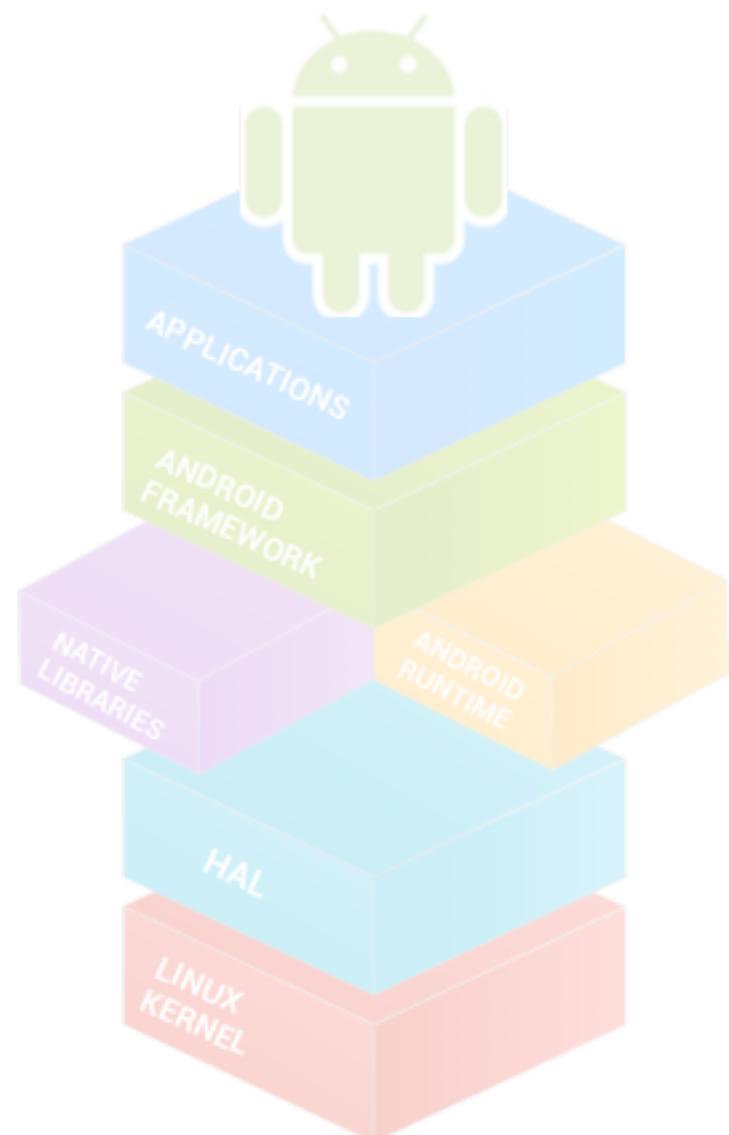
Class, Text

UI Hierarchy



Test Case Recording: Actions

Accessibility



Events

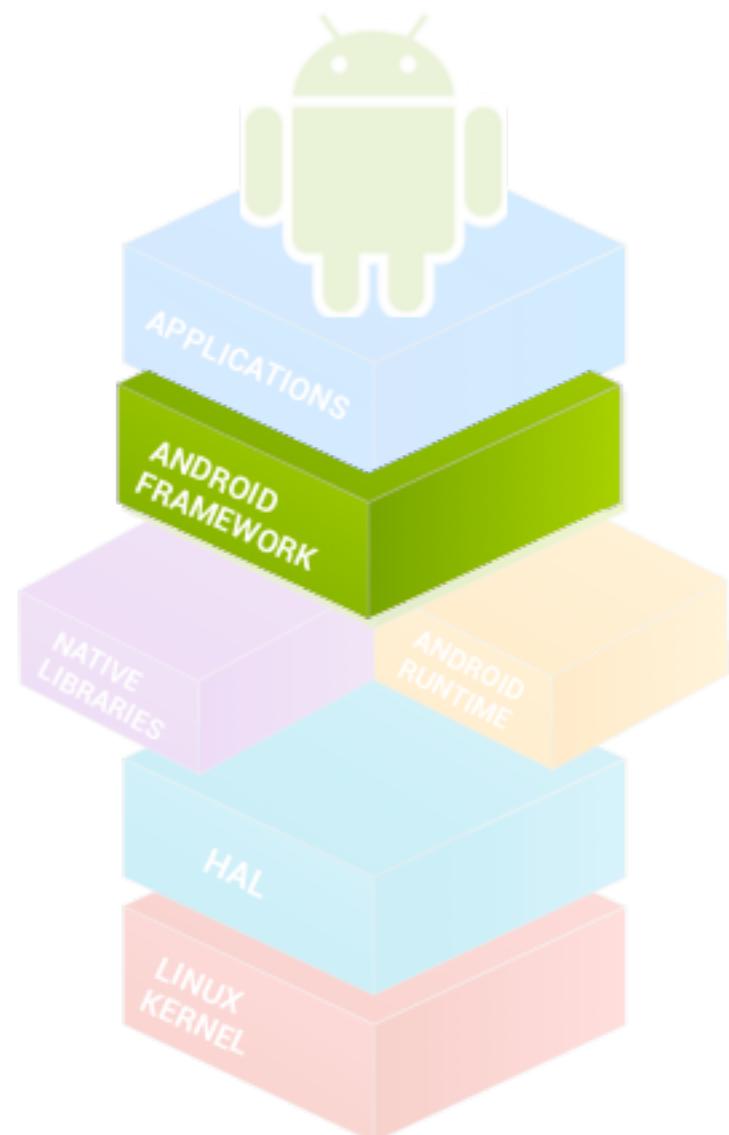


Recorded
Trace

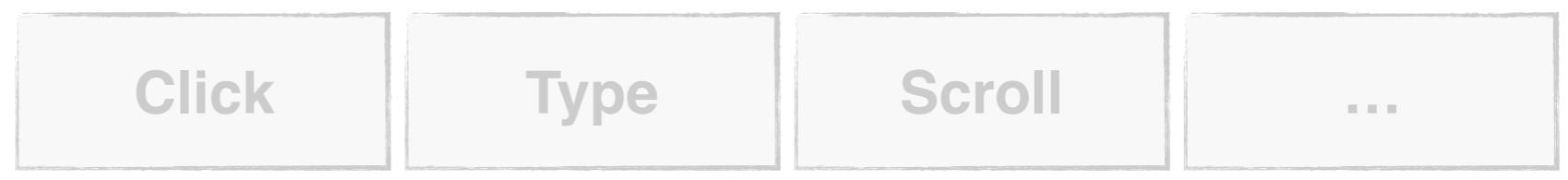


Test Case Recording: Actions

Accessibility



Events

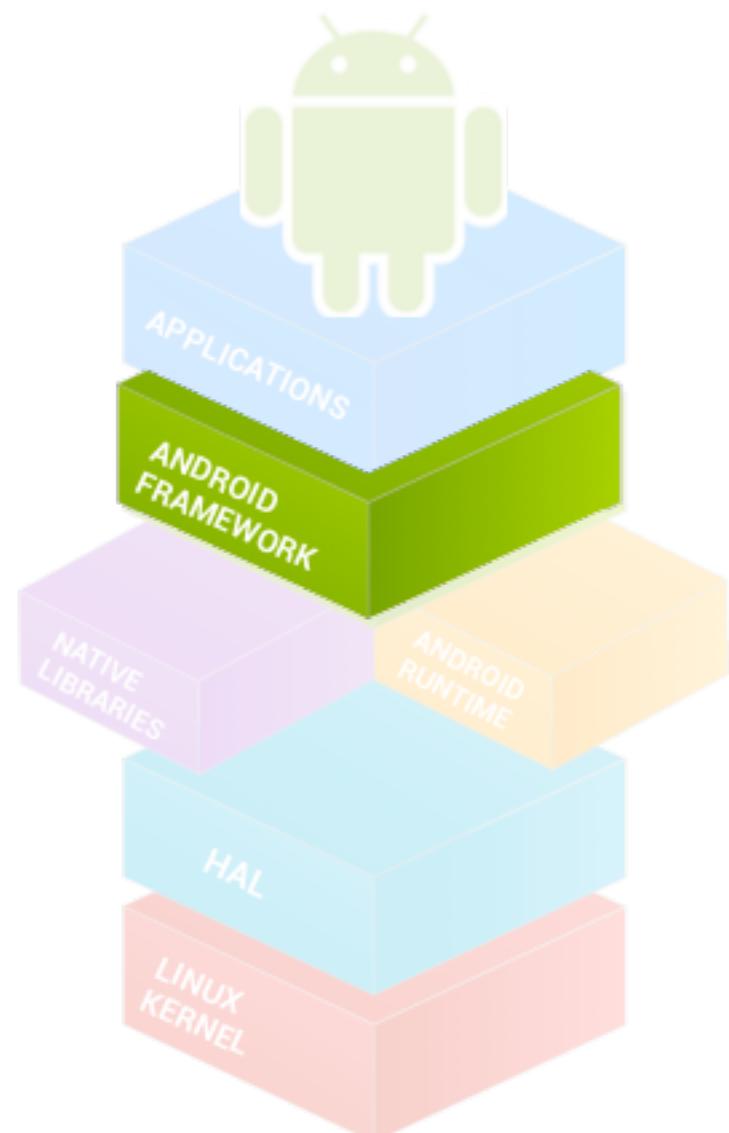


Recorded
Trace

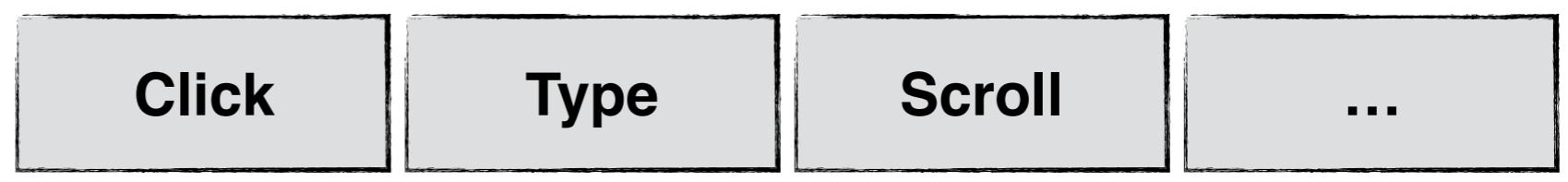


Test Case Recording: Actions

Accessibility



Events

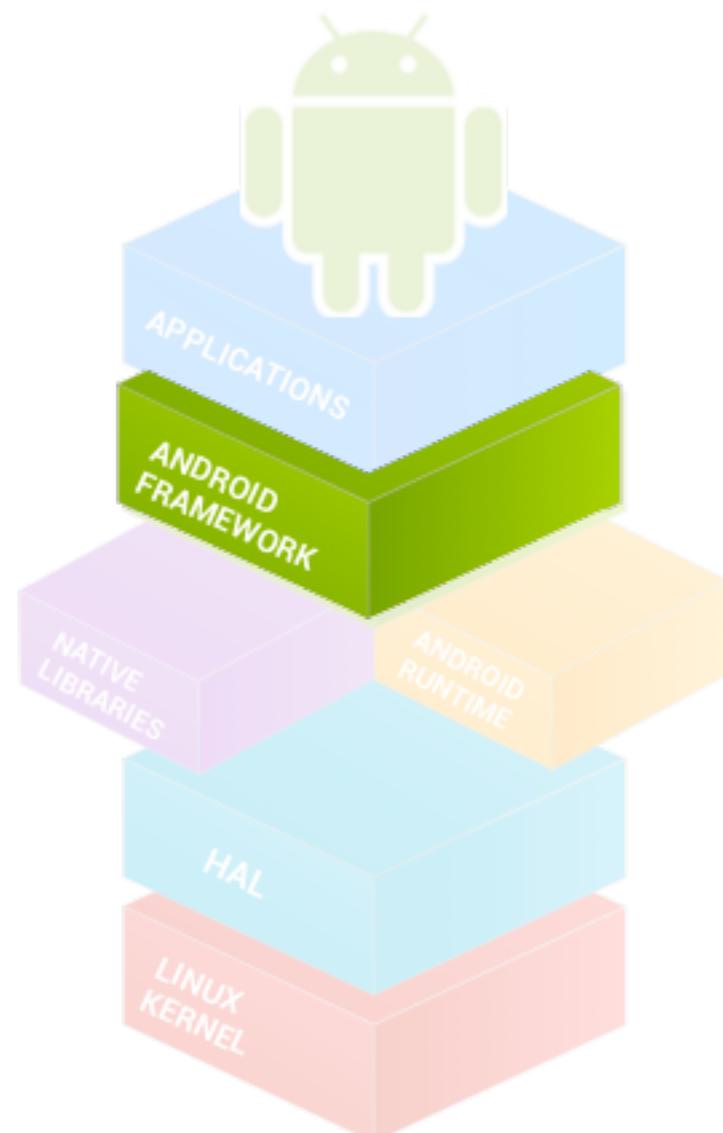


Recorded
Trace

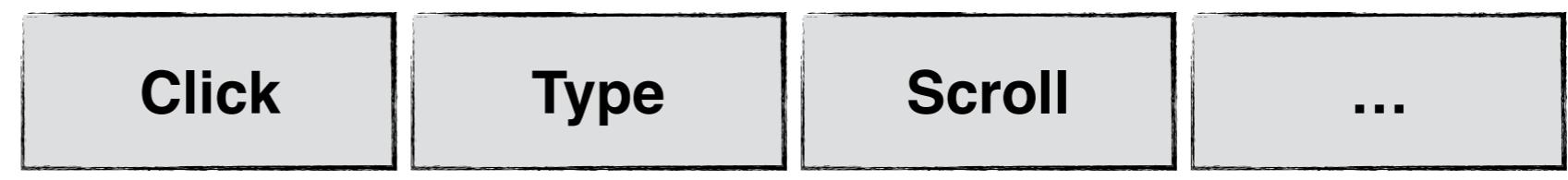


Test Case Recording: Actions

Accessibility



Events

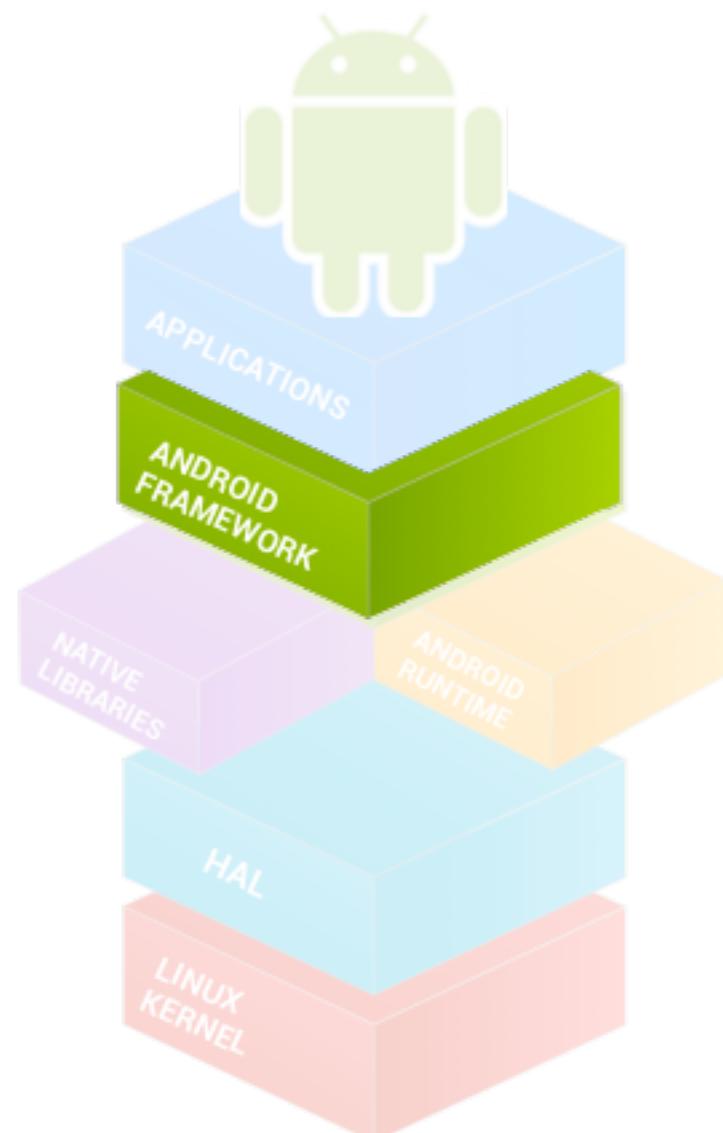


Recorded
Trace

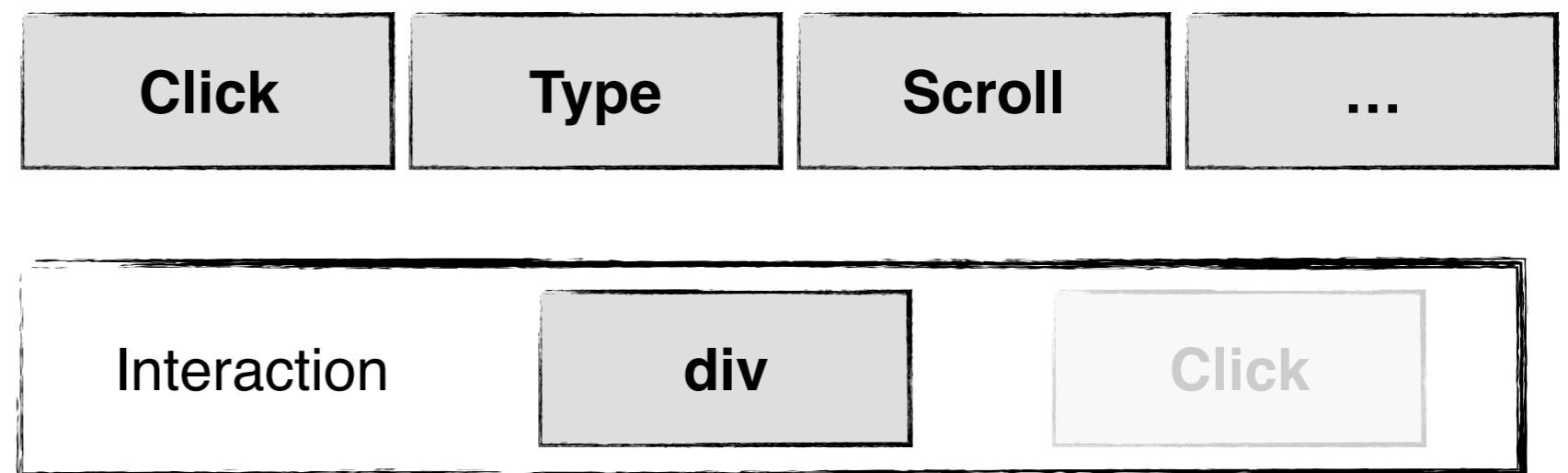


Test Case Recording: Actions

Accessibility



Events



Recorded
Trace

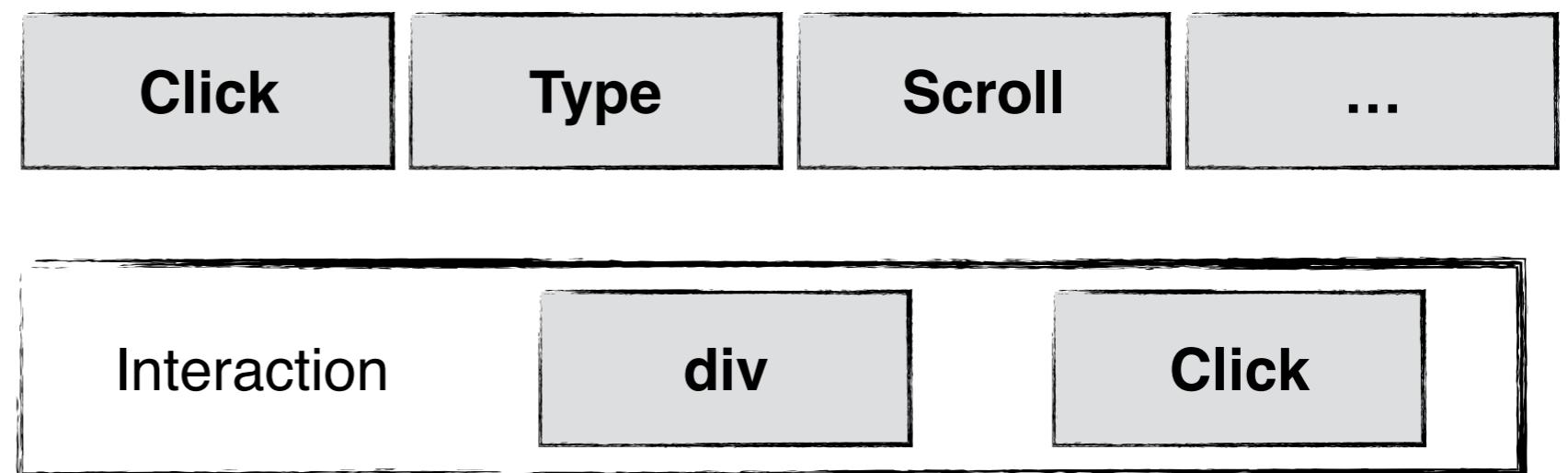


Test Case Recording: Actions

Accessibility

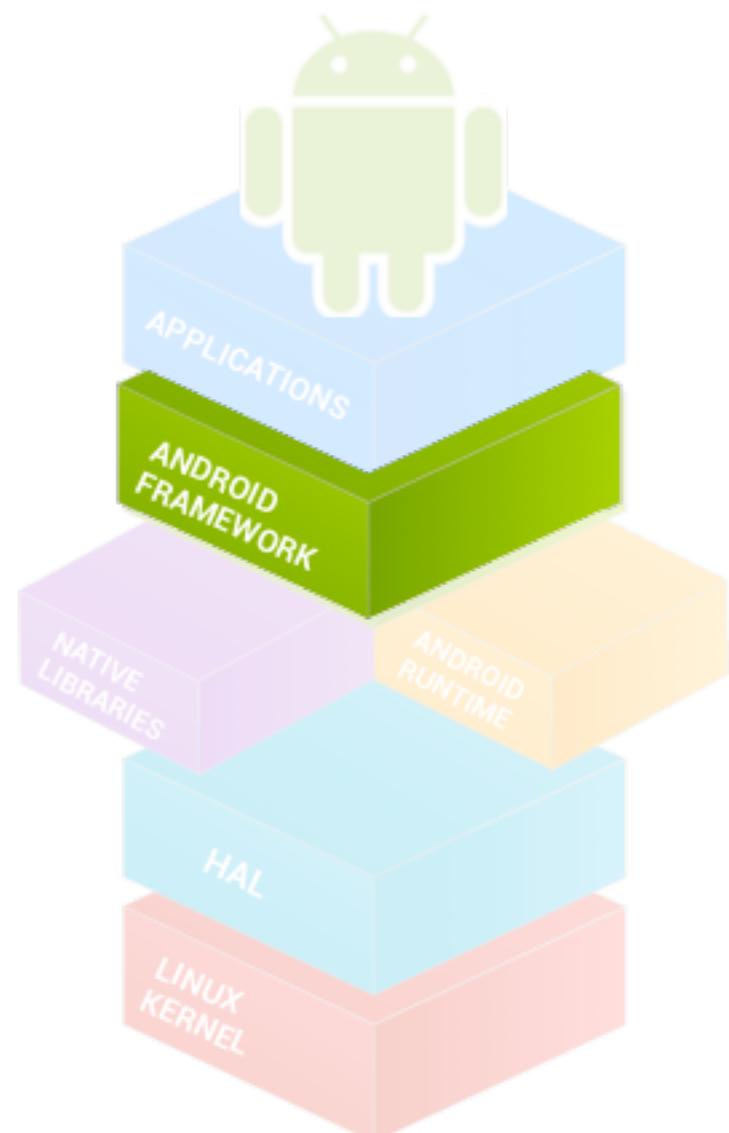


Events

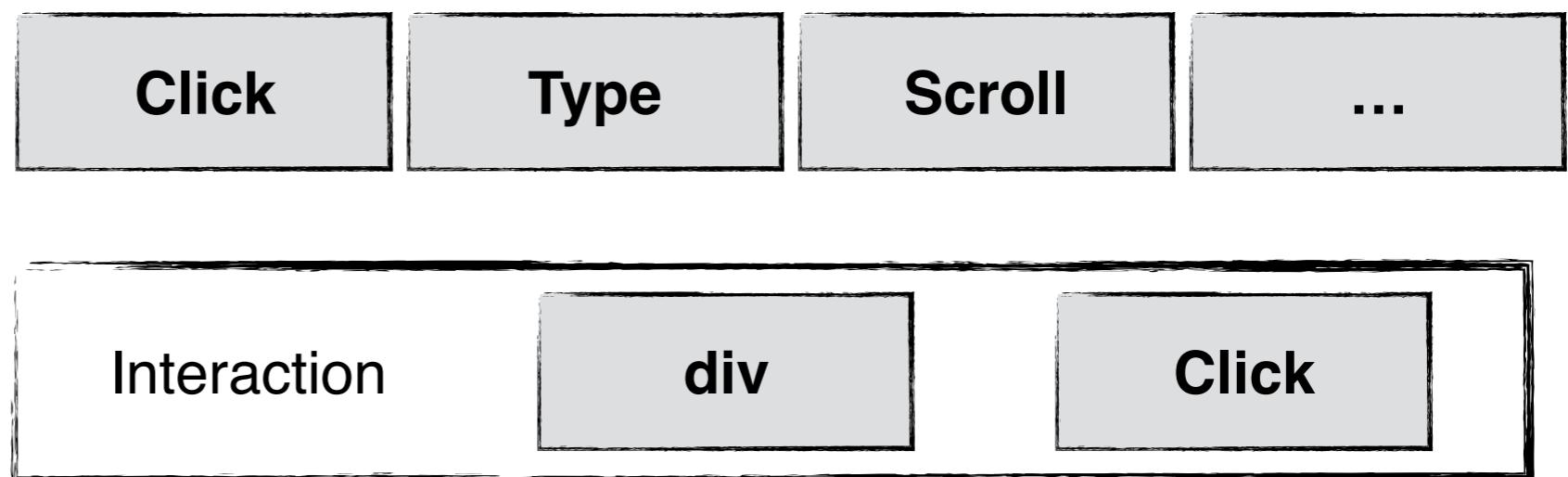


Test Case Recording: Actions

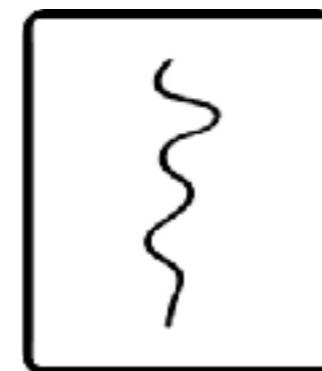
Accessibility



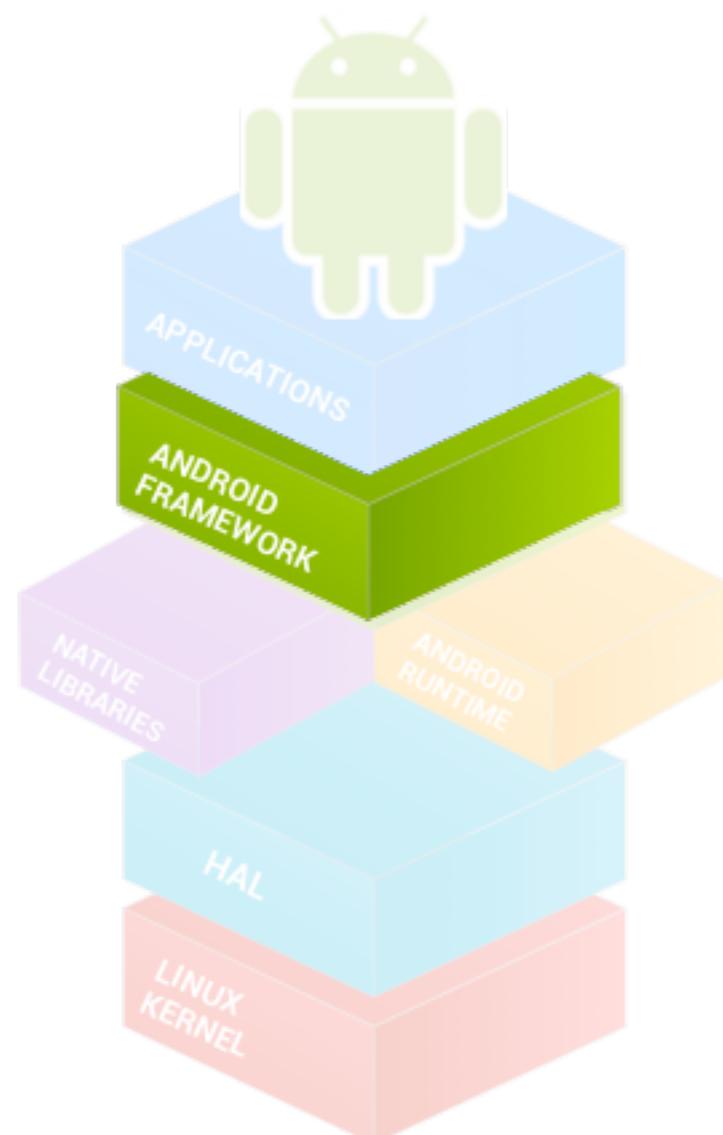
Events



**Recorded
Trace**



Test Case Recording: System Actions



User-induced System Events

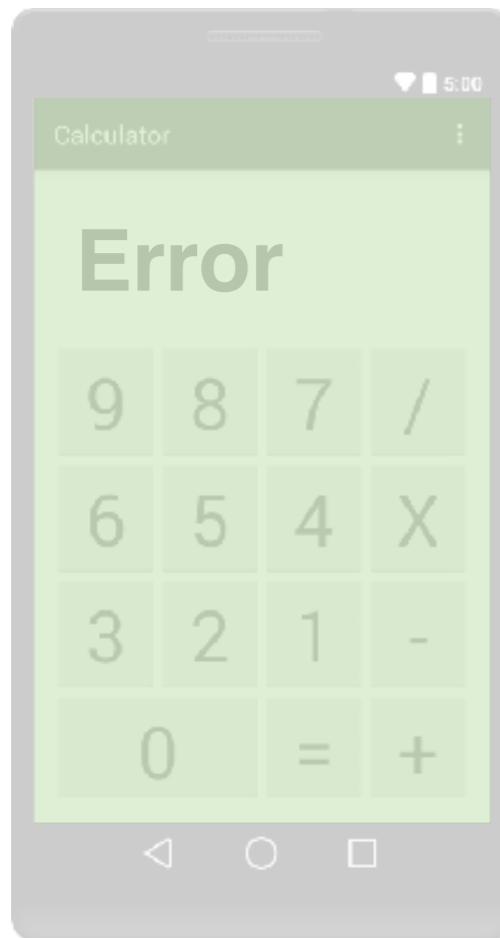
Intents

Activity

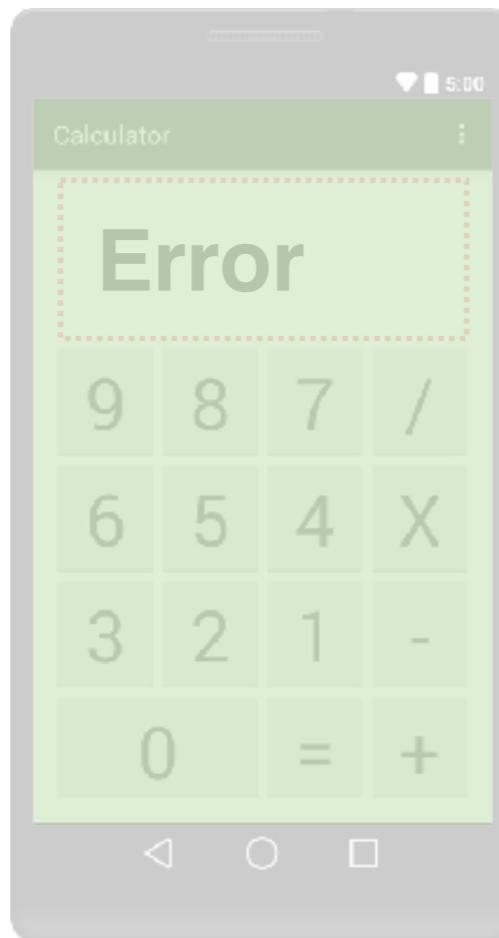
Config

Test Case Recording: Oracles

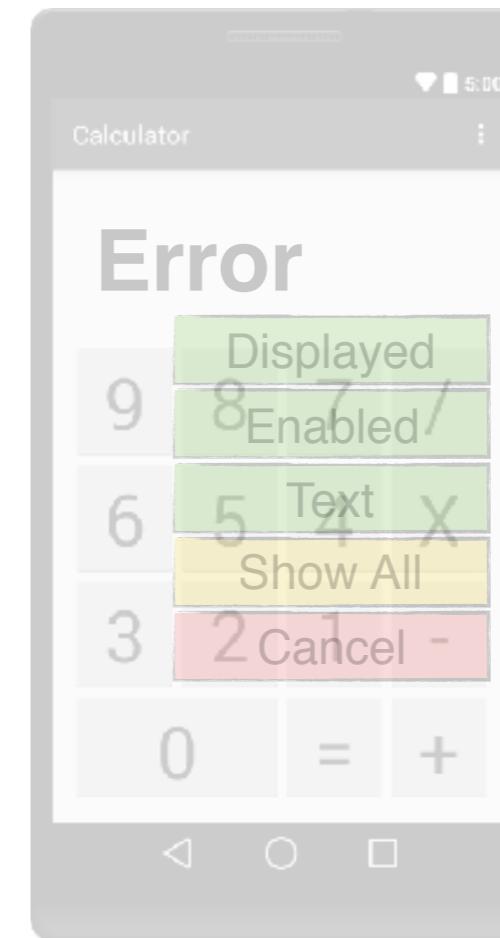
Overlay



Selection



Assertion



Recorded
Trace

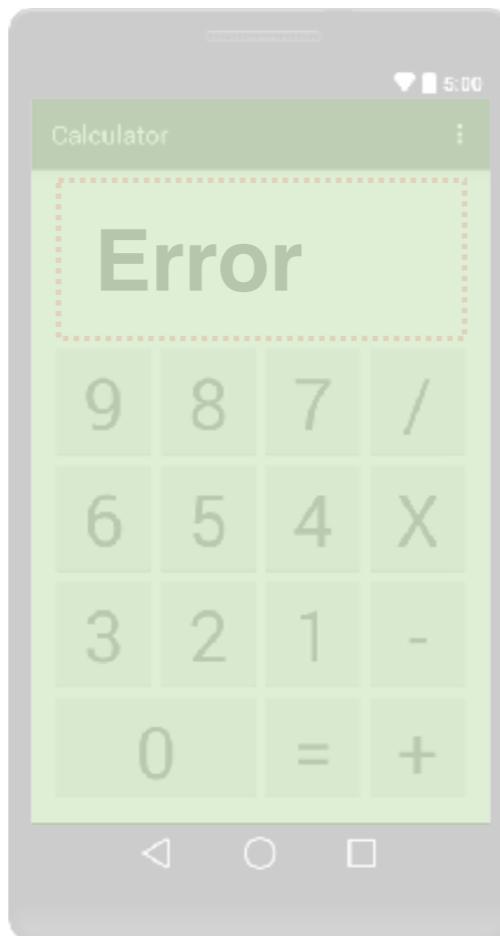


Test Case Recording: Oracles

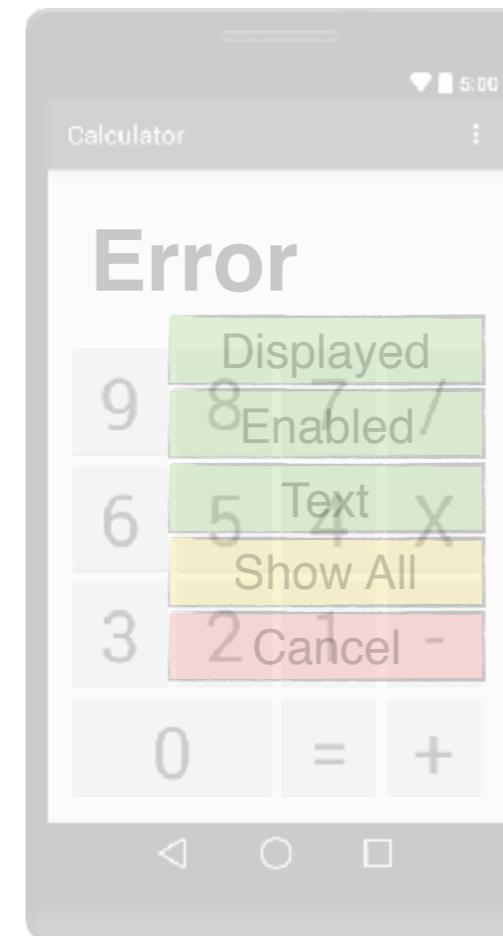
Overlay



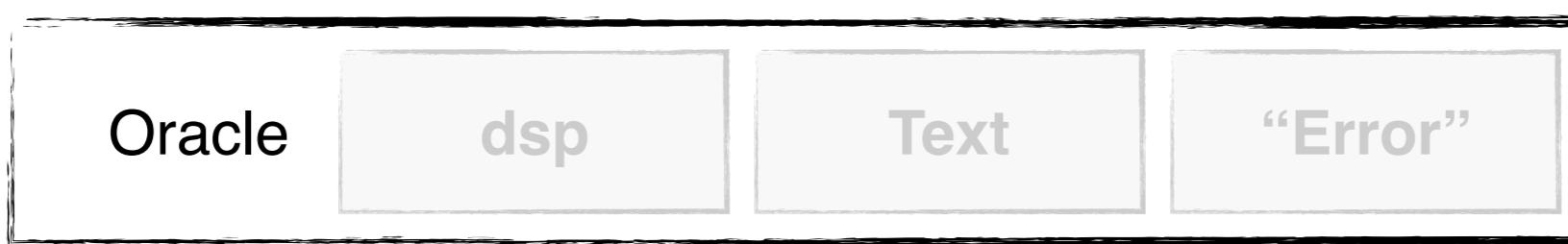
Selection



Assertion



Recorded
Trace



Test Case Recording: Oracles

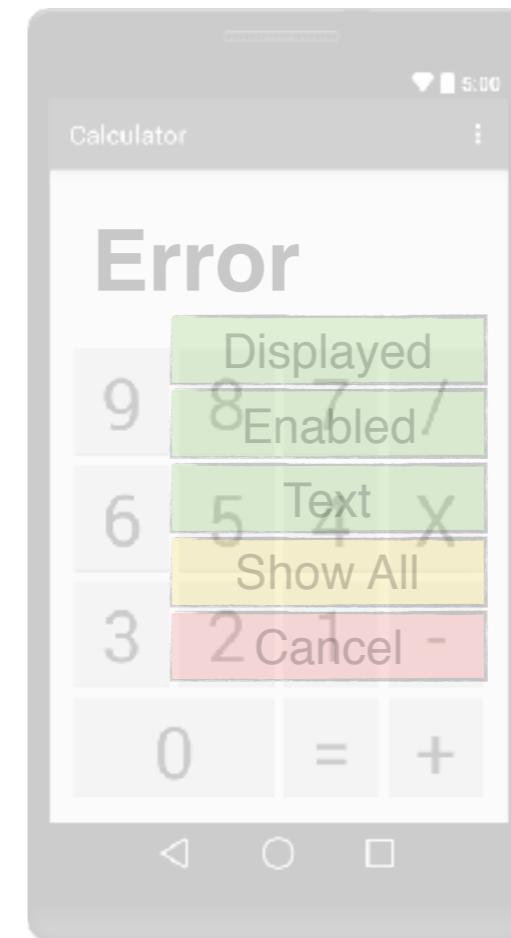
Overlay



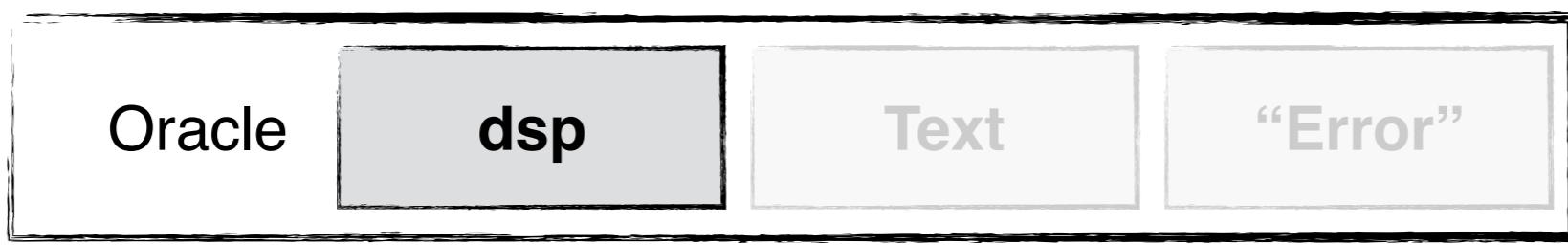
Selection



Assertion



Recorded
Trace



Test Case Recording: Oracles

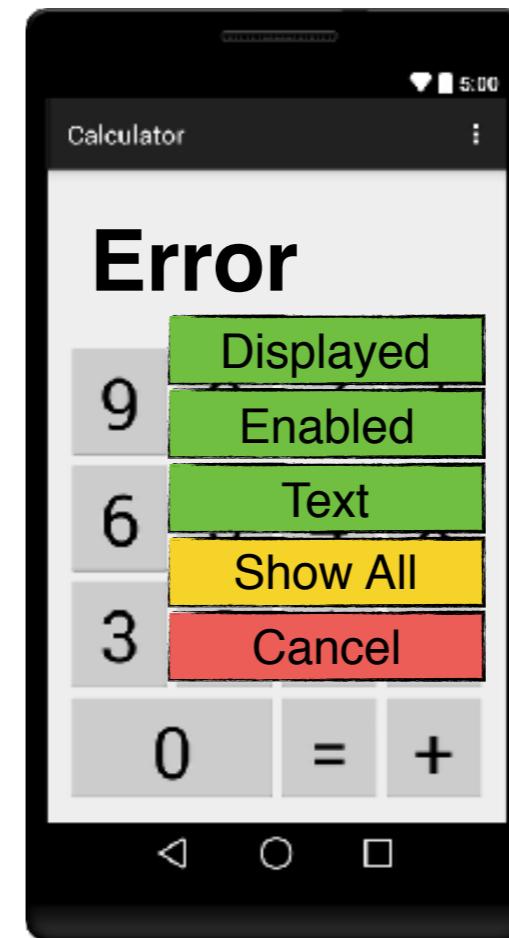
Overlay



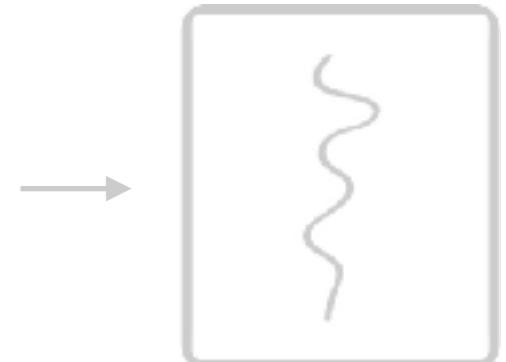
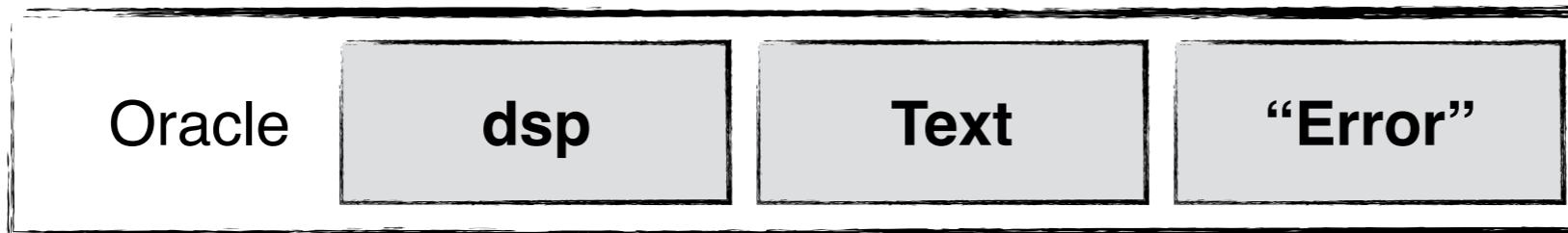
Selection



Assertion



Recorded
Trace



Test Case Recording: Oracles

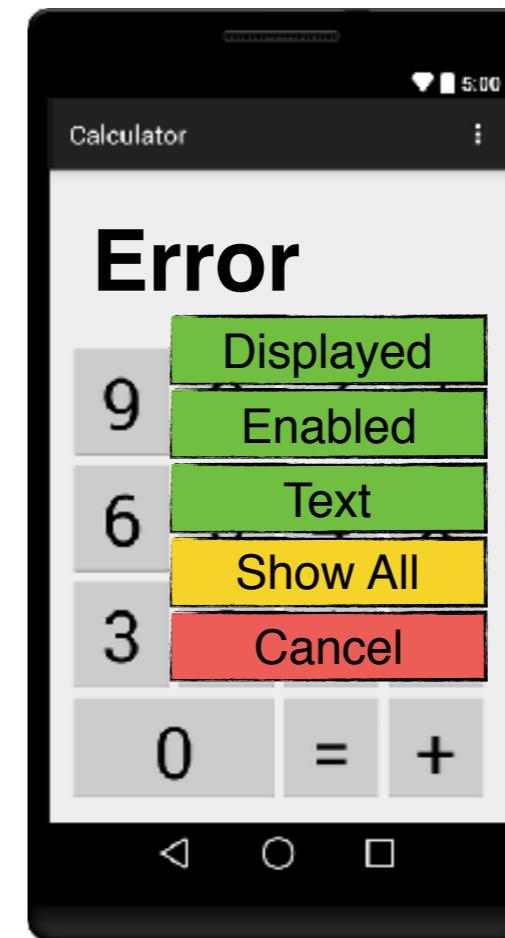
Overlay



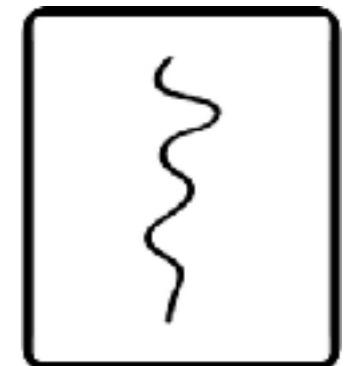
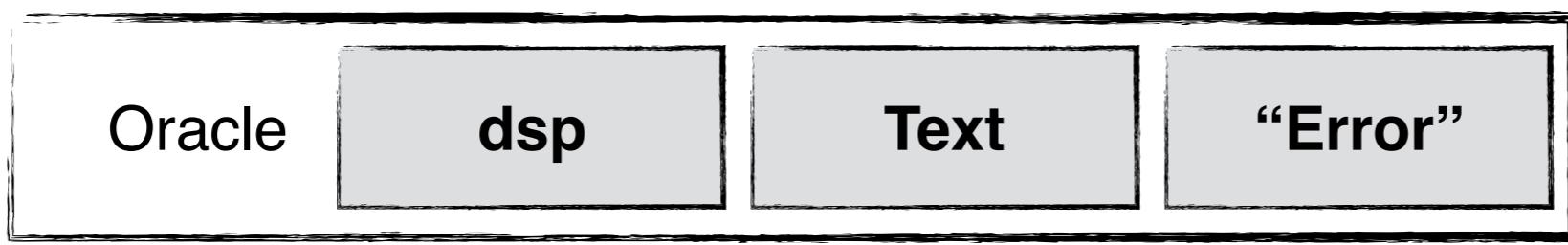
Selection



Assertion



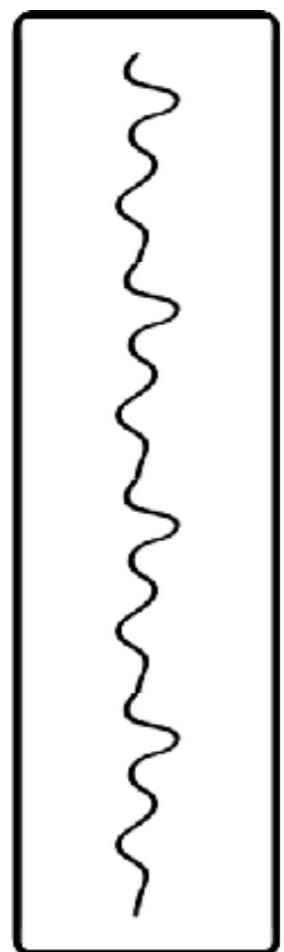
Recorded
Trace



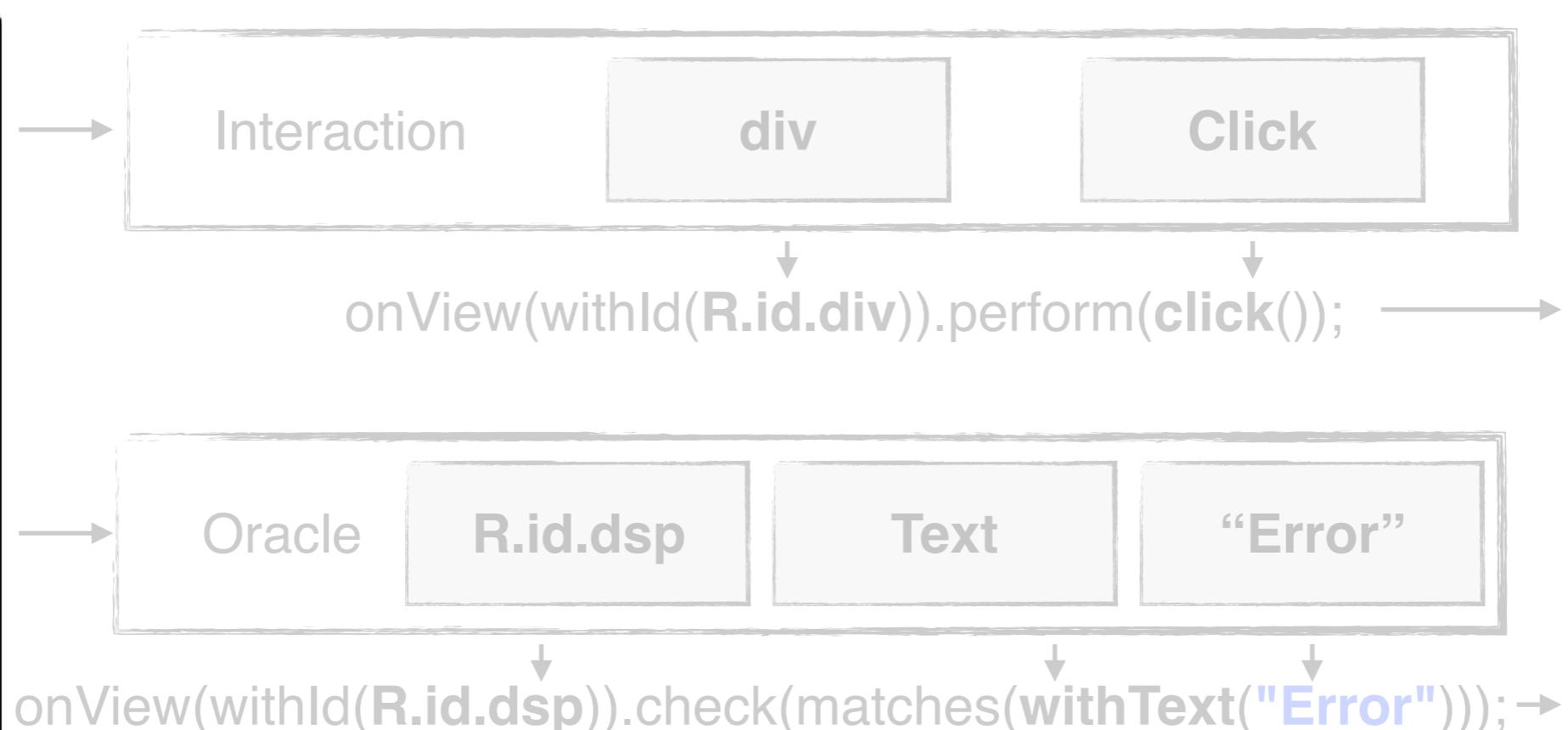
Test Case Encoding

Objective: translate recorded actions into a test case

**Recorded
Trace**



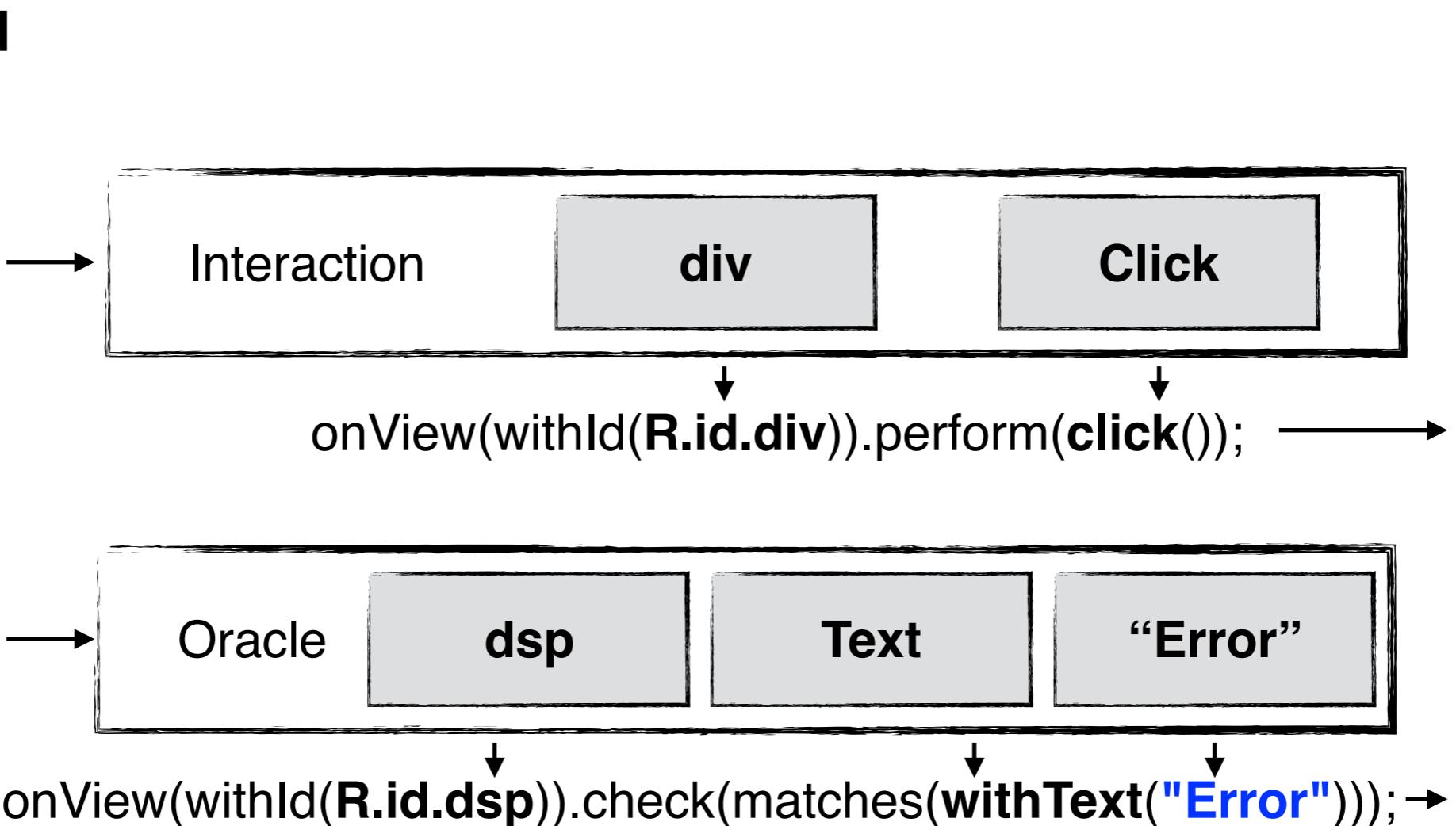
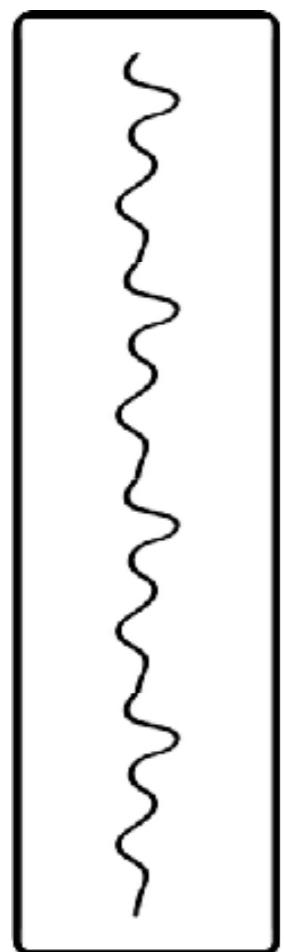
**Test
Case**



Test Case Encoding

Objective: translate recorded actions into a test case

Recorded
Trace



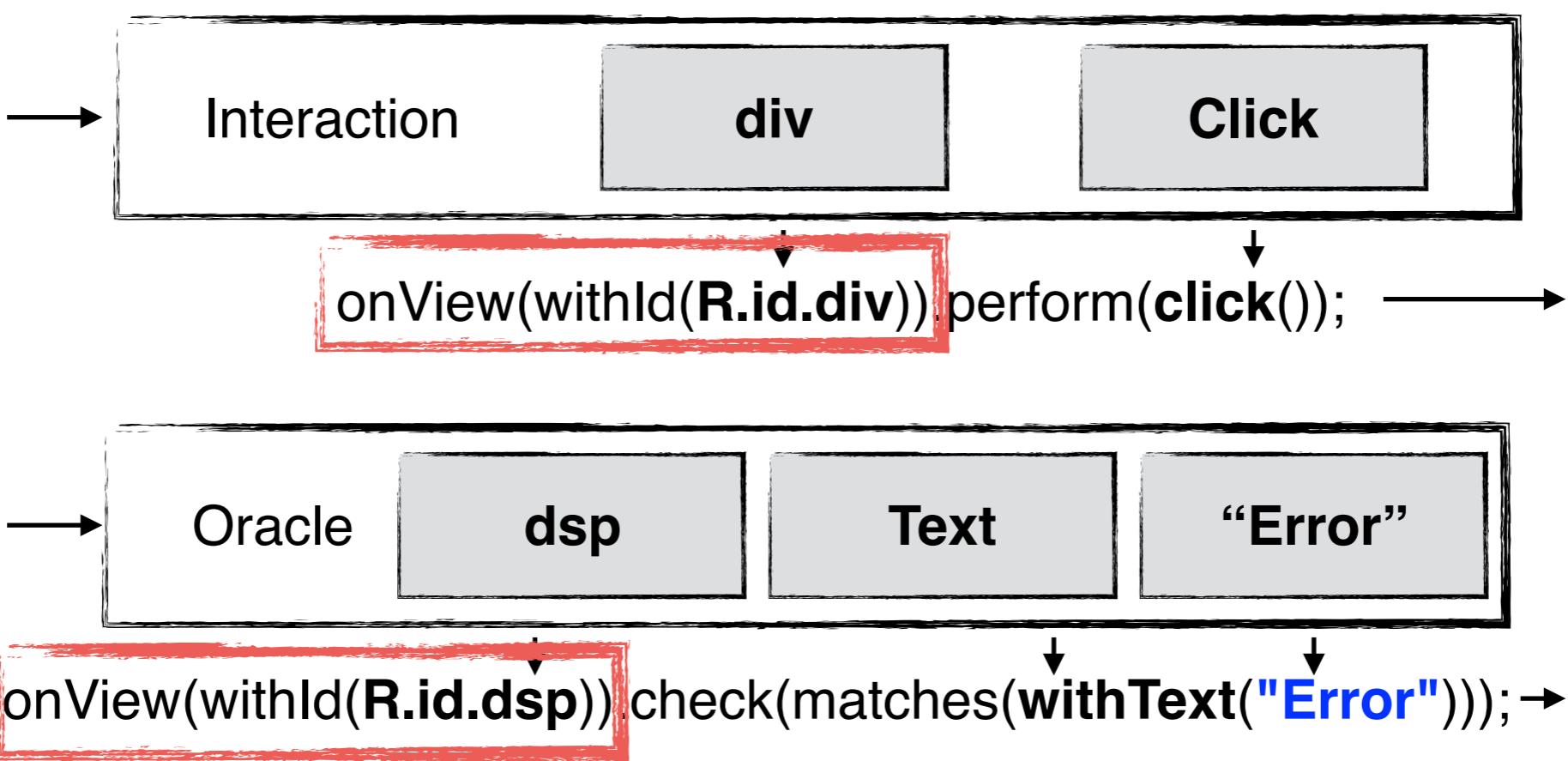
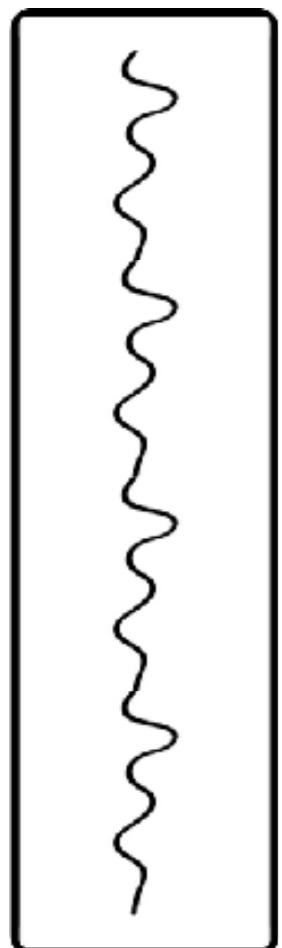
Test
Case



Test Case Encoding

Objective: translate recorded actions into a test case

Recorded
Trace



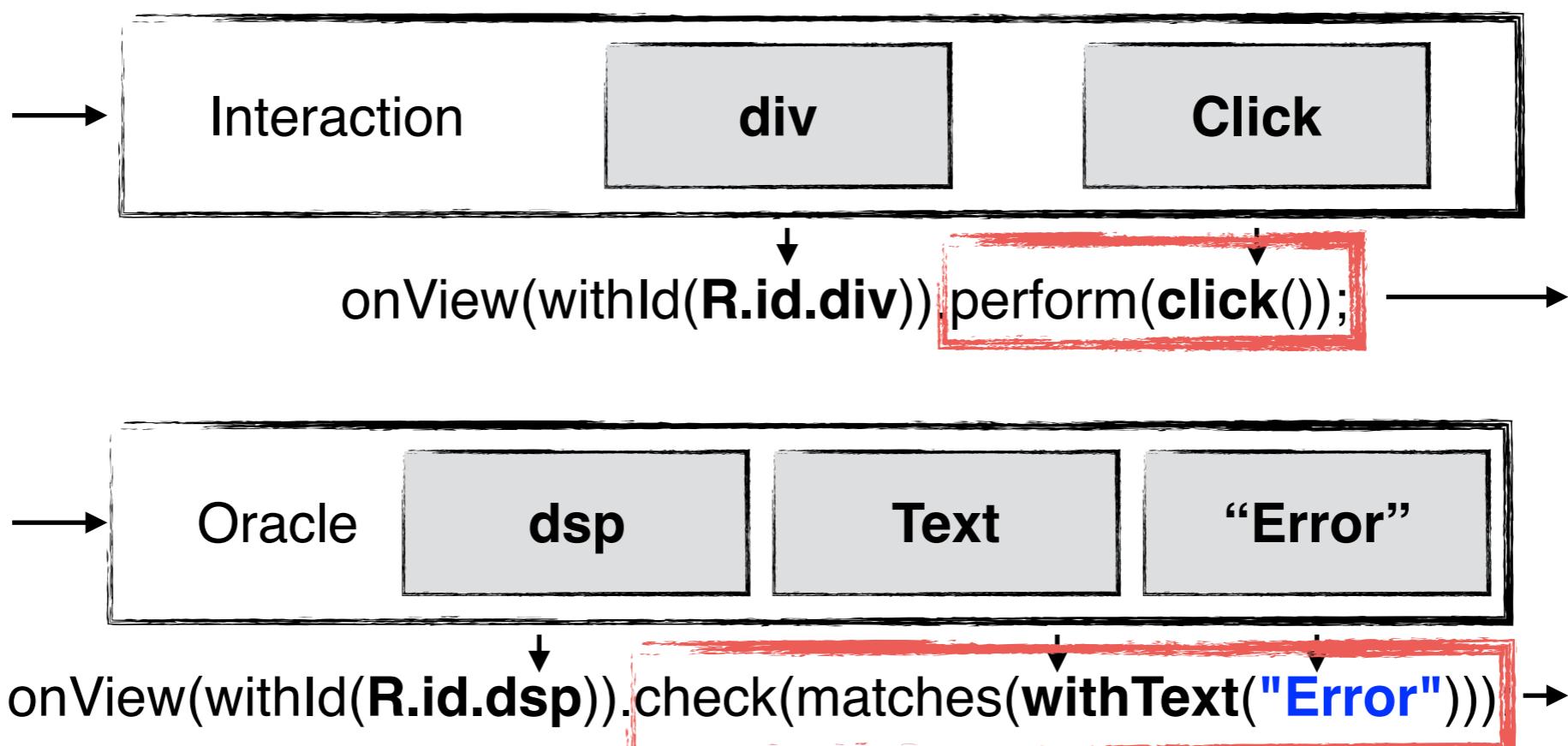
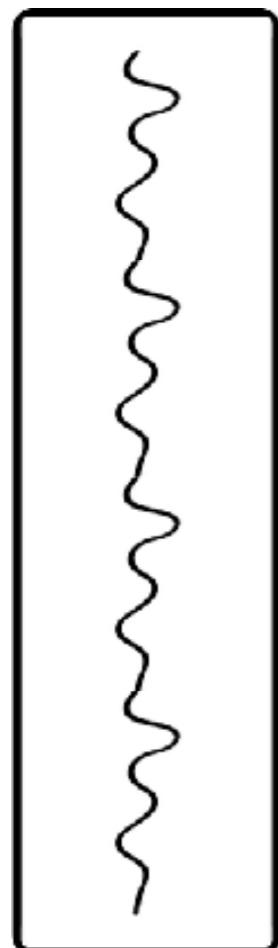
Test
Case



Test Case Encoding

Objective: translate recorded actions into a test case

Recorded
Trace



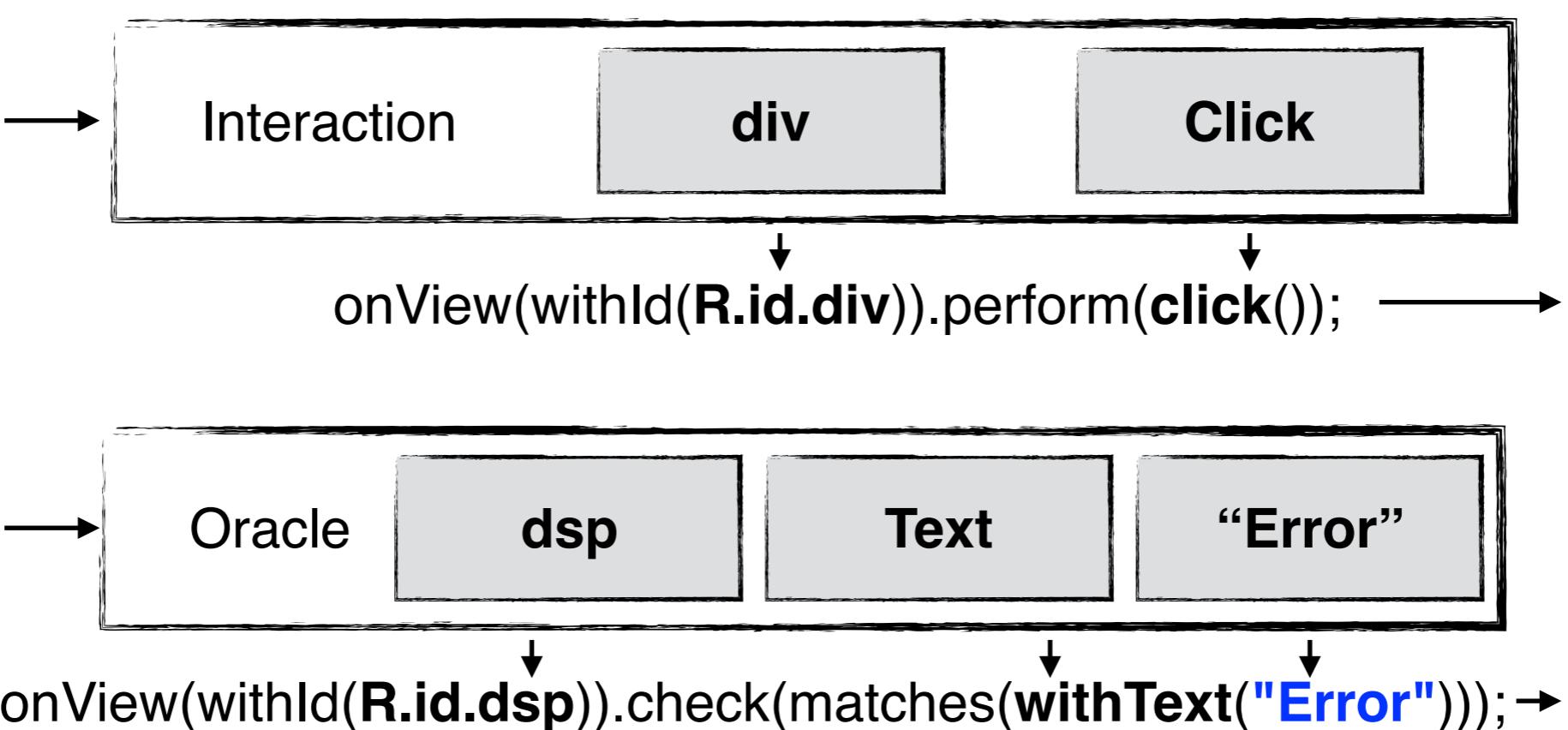
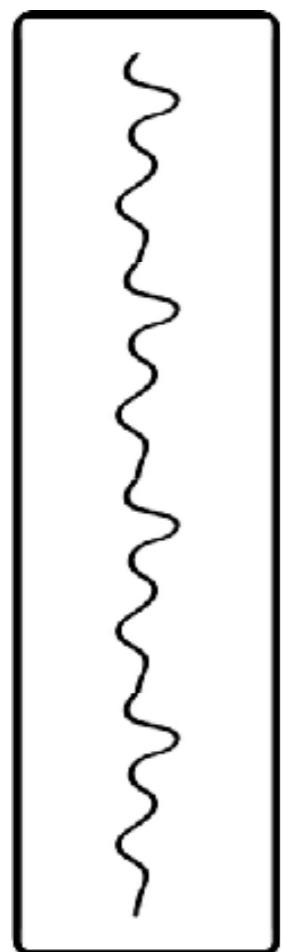
Test
Case



Test Case Encoding

Objective: translate recorded actions into a test case

Recorded
Trace



Test
Case



Test Case Execution

Objective: execute test cases and generate a report

**Test
Case**



Test Devices



Success

Error

Failure

**Test
Report**



Test Case Execution

Objective: execute test cases and generate a report

Test Devices

**Test
Case**



Success

Error

Failure

Test
Report



Test Case Execution

Objective: execute test cases and generate a report

Test Devices

**Test
Case**



Success

Error

Failure

Test
Report



Test Case Execution

Objective: execute test cases and generate a report

Test Devices

**Test
Case**



Success

Error

Failure

**Test
Report**



Test Case Execution

Objective: execute test cases and generate a report

Test Devices

**Test
Case**



Success

Error

Failure

**Test
Report**



Test Case Execution

Objective: execute test cases and generate a report

Test Devices

**Test
Case**



Success

Error

Failure

**Test
Report**



Implementation

Test Case
Recording



Test Case
Encoding



Test Case
Execution



+

Java Poet



Source Code
Generator

+

Spoon



Test Report
Generator

Implementation

Test Case Recording



Test Case Encoding



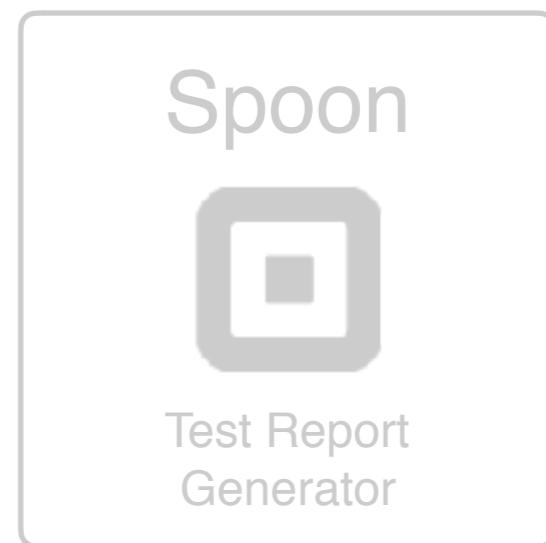
Test Case Execution



+



+



Implementation

Test Case
Recording



Test Case
Encoding



Test Case
Execution



+

Java Poet



Source Code
Generator

+

Spoon



Test Report
Generator

Implementation

Test Case
Recording



Test Case
Encoding



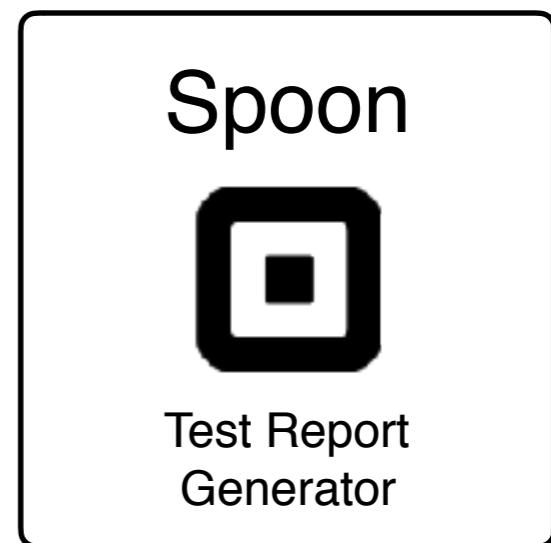
Test Case
Execution



+



+



Empirical Evaluation

Research Questions:

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2 (EFFICIENCY): Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3 (REPRODUCIBILITY): Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?

Empirical Evaluation

Research Questions:

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2 (EFFICIENCY): Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3 (REPRODUCIBILITY): Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?

Empirical Evaluation

Research Questions:

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2 (EFFICIENCY): Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3 (REPRODUCIBILITY): Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?

Empirical Evaluation

Research Questions:

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2 (EFFICIENCY): Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3 (REPRODUCIBILITY): Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?

Empirical Evaluation

Research Questions:

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2 (EFFICIENCY): Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3 (REPRODUCIBILITY): Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?

Empirical Evaluation

Research Questions:

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2 (EFFICIENCY): Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3 (REPRODUCIBILITY): Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?

Benchmarks

<i>ID</i>	<i>Name</i>	<i>Category</i>	<i>Installations (#K)</i>
A1	Daily Money	Finance	500 - 1000
A2	Alarm Clock	Tools	500 - 1000
A3	QuickDic	Books	1000 - 5000
A4	Simple C25K	Health	50 - 100
A5	Comics Reader	Comics	100 - 500
A6	ConnectBot	Communication	1000 - 5000
A7	Weather Notification	Weather	100 - 500
A8	Barcode Scanner	Shopping	100000 - 500000
A9	MicDroid	Media	1000 - 5000
A10	EP Mobile	Medical	50 - 100
A11	BeeCount	Productivity	10 - 50
A12	Bodhi Timer	Lifestyle	10 - 50
A13	andFHEM	Personalization	10 - 50
A14	Xmp Mod Player	Music & Audio	10 - 50
A15	World Clock	Travel & Local	50 - 100

Study Protocol

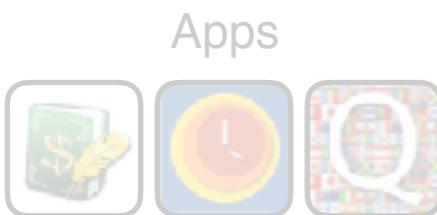


x 15

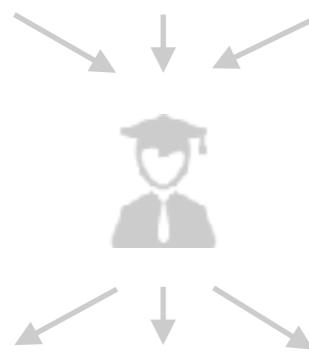


x 15

First Task



Apps

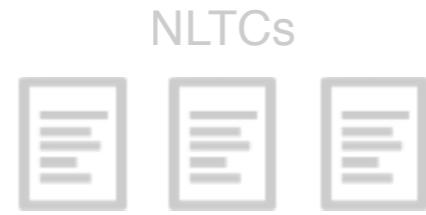


NLTCs

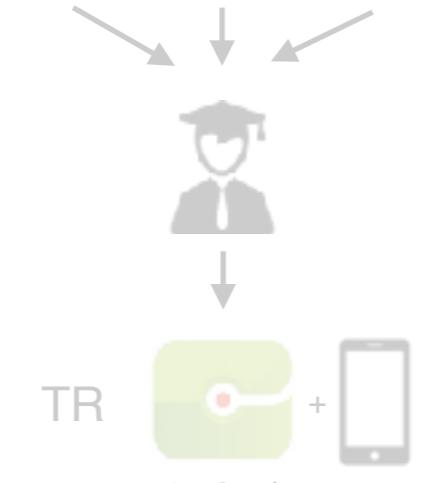


x5 x5 x5

Second Task



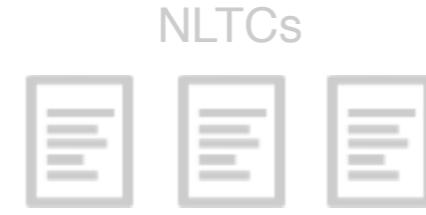
NLTCs



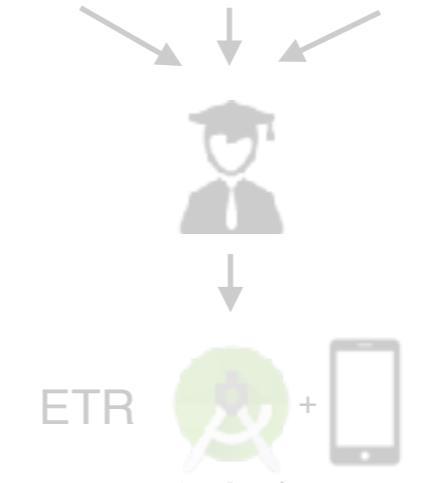
Test Cases



Third Task



NLTCs



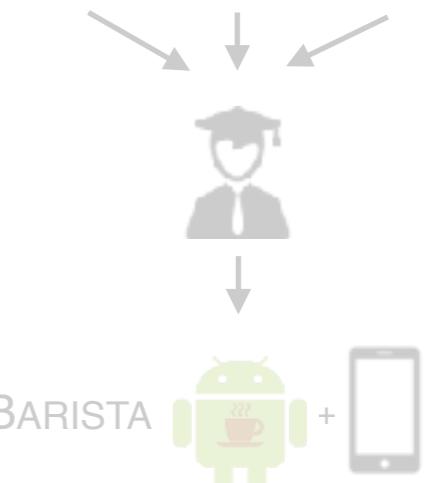
Test Cases



Fourth Task



NLTCs



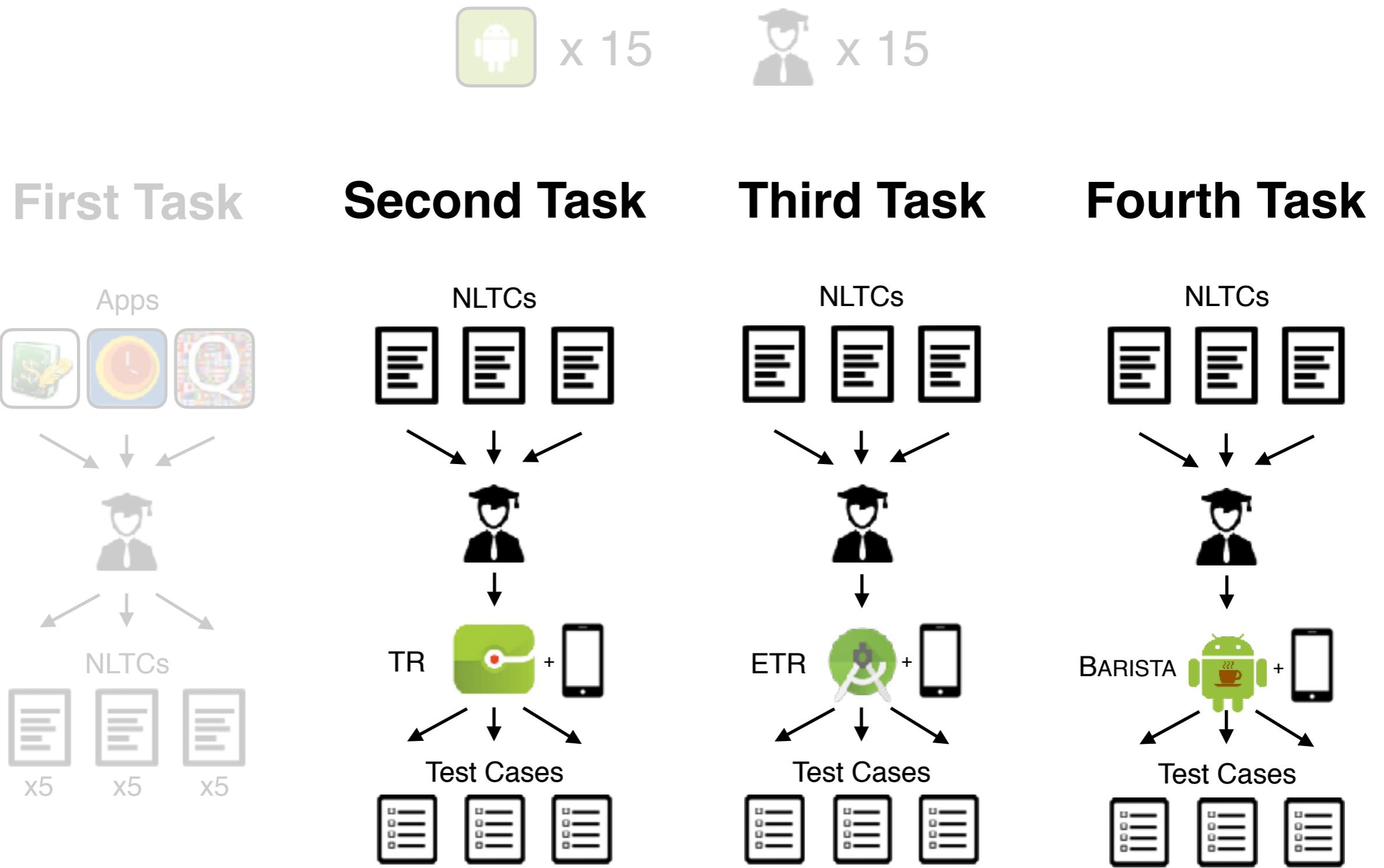
Test Cases



Study Protocol



Study Protocol



RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)
A1	15	0	9	20	15	0	0	36	15	0	2	0
A2	4	11	0	2	15	0	14	3	15	0	0	0
A3	9	5	5	1	13	1	3	9	14	0	2	0
A4	9	5	8	7	14	0	26	13	14	0	3	0
A5	12	2	2	2	14	0	1	29	14	0	0	0
A6	6	7	0	1	13	0	0	13	13	0	0	0
A7	11	3	13	5	13	1	5	21	14	0	0	0
A8	11	3	5	9	14	0	0	17	14	0	0	0
A9	11	1	3	3	12	0	0	12	12	0	0	0
A10	13	2	10	2	14	1	17	8	15	0	0	0
A11	12	3	10	0	15	0	1	13	15	0	0	0
A12	15	0	5	0	15	0	4	14	15	0	0	0
A13	13	2	14	3	15	0	0	39	15	0	0	0
A14	15	0	7	2	11	4	1	14	15	0	2	0
A15	15	0	17	0	14	1	1	35	15	0	2	0
Tot	171	44	108	48	208	7	74	277	215	0	11	0

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)
A1	15	0	9	20	15	0	0	36	15	0	2	0
A2	4	11	0	2	15	0	14	3	15	0	0	0
A3	9	5	5	1	13	1	3	9	14	0	2	0
A4	9	5	8	7	14	0	26	13	14	0	3	0
A5	12	2	2	2	14	0	1	29	14	0	0	0
A6	6	7	0	1	13	0	0	13	13	0	0	0
A7	11	3	13	5	13	1	5	21	14	0	0	0
A8	11	3	5	9	14	0	0	17	14	0	0	0
A9	11	1	3	3	12	0	0	12	12	0	0	0
A10	13	2	10	2	14	1	17	8	15	0	0	0
A11	12	3	10	0	15	0	1	13	15	0	0	0
A12	15	0	5	0	15	0	4	14	15	0	0	0
A13	13	2	14	3	15	0	0	39	15	0	0	0
A14	15	0	7	2	11	4	1	14	15	0	2	0
A15	15	0	17	0	14	1	1	35	15	0	2	0
Tot	171	44	108	48	208	7	74	277	215	0	11	0

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)
A1	15	0	9	20	15	0	0	36	15	0	2	0
A2	4	11	0	2	15	0	14	3	15	0	0	0
A3	9	5	5	1	13	1	3	9	14	0	2	0
A4	9	5	8	7	14	0	26	13	14	0	3	0
A5	12	2	2	2	14	0	1	29	14	0	0	0
A6	6	7	0	1	13	0	0	13	13	0	0	0
A7	11	3	13	5	13	1	5	21	14	0	0	0
A8	11	3	5	9	14	0	0	17	14	0	0	0
A9	11	1	3	3	12	0	0	12	12	0	0	0
A10	13	2	10	2	14	1	17	8	15	0	0	0
A11	12	3	10	0	15	0	1	13	15	0	0	0
A12	15	0	5	0	15	0	4	14	15	0	0	0
A13	13	2	14	3	15	0	0	39	15	0	0	0
A14	15	0	7	2	11	4	1	14	15	0	2	0
A15	15	0	17	0	14	1	1	35	15	0	2	0
Tot	171	44	108	48	208	7	74	277	215	0	11	0

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)
A1	15	0	9	20	15	0	0	36	15	0	2	0
A2	4	11	0	2	15	0	14	3	15	0	0	0
A3	9	5	5	1	13	1	3	9	14	0	2	0
A4	9	5	8	7	14	0	26	13	14	0	3	0
A5	12	2	2	2	14	0	1	29	14	0	0	0
A6	6	7	0	1	13	0	0	13	13	0	0	0
A7	11	3	13	5	13	1	5	21	14	0	0	0
A8	11	3	5	9	14	0	0	17	14	0	0	0
A9	11	1	3	3	12	0	0	12	12	0	0	0
A10	13	2	10	2	14	1	17	8	15	0	0	0
A11	12	3	10	0	15	0	1	13	15	0	0	0
A12	15	0	5	0	15	0	4	14	15	0	0	0
A13	13	2	14	3	15	0	0	39	15	0	0	0
A14	15	0	7	2	11	4	1	14	15	0	2	0
A15	15	0	17	0	14	1	1	35	15	0	2	0
Tot	171	44	108	48	208	7	74	277	215	0	11	0

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)
A1	15	0	9	20	15	0	0	36	15	0	2	0
A2	4	11	0	2	15	0	14	3	15	0	0	0
A3	9	5	5	1	13	1	3	9	14	0	2	0
A4	9	5	8	7	14	0	26	13	14	0	3	0
A5	12	2	2	2	14	0	1	29	14	0	0	0
A6	6	7	0	1	13	0	0	13	13	0	0	0
A7	11	3	13	5	13	1	5	21	14	0	0	0
A8	11	3	5	9	14	0	0	17	14	0	0	0
A9	11	1	3	3	12	0	0	12	12	0	0	0
A10	13	2	10	2	14	1	17	8	15	0	0	0
A11	12	3	10	0	15	0	1	13	15	0	0	0
A12	15	0	5	0	15	0	4	14	15	0	0	0
A13	13	2	14	3	15	0	0	39	15	0	0	0
A14	15	0	7	2	11	4	1	14	15	0	2	0
A15	15	0	17	0	14	1	1	35	15	0	2	0
Tot	171	44	108	48	208	7	74	277	215	0	11	0

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)

BARISTA



Test Cases Completed

100%

Tot	171	44	108	48	208	7	74	277	215	0	11	0
-----	-----	----	-----	----	-----	---	----	-----	-----	---	----	---

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)

BARISTA



Assertions Skipped

2%

Tot	171	44	108	48	208	7	74	277	215	0	11	0
-----	-----	----	-----	----	-----	---	----	-----	-----	---	----	---

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)

BARISTA



TR



vs.

Test Cases Completed

+20%

Tot	171	44	108	48	208	7	74	277	215	0	11	0
-----	-----	----	-----	----	-----	---	----	-----	-----	---	----	---

RQ1

RQ1 (COMPLETENESS): Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

ID	TR				ETR				Barista			
	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)	C(#)	NC(#)	AS(#)	AA(#)

Assertions Skipped

TR



108

Assertions Altered

48

ETR



74

277

Tot	171	44	108	48	208	7	74	277	215	0	11	0
-----	-----	----	-----	----	-----	---	----	-----	-----	---	----	---

RQ4

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?



RQ4

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?



RQ4

RQ4 (COMPATIBILITY): Can test cases generated by BARISTA run on different devices? How does it compare to TR and ETR?



Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

“I have been looking for something like BARISTA to help me get into automation for a while”

Feedback #2

“Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us”

Feedback #3

“There are a few more assertions I’d like to see. For example, testing the number of items in a ListView”

Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

“I have been looking for something like BARISTA to help me get into automation for a while”

Feedback #2

“Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us”

Feedback #3

“There are a few more assertions I’d like to see. For example, testing the number of items in a ListView”

Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

“I have been looking for something like BARISTA to help me get into automation for a while”

Feedback #2

“Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us”

Feedback #3

“There are a few more assertions I’d like to see. For example, testing the number of items in a ListView”

Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

“I have been looking for something like BARISTA to help me get into automation for a while”

Feedback #2

“Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us”

Feedback #3

“There are a few more assertions I’d like to see. For example, testing the number of items in a ListView”

Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

"I have been looking for something like BARISTA to help me get into automation for a while"

Feedback #2

"Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us"

Feedback #3

"There are a few more assertions I'd like to see. For example, testing the number of items in a ListView"

Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

"I have been looking for something like BARISTA to help me get into automation for a while"

Feedback #2

"Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us"

Feedback #3

"There are a few more assertions I'd like to see. For example, testing the number of items in a ListView"

Evaluation Summary

RQ1: BARISTA can record test cases and is more expressive than TR and ETR

RQ2: BARISTA is more efficient in recording test cases than TR and ETR

RQ3: BARISTA generates test cases that work correctly ad it outperforms TR and ETR

RQ4: BARISTA generates test cases that can run on different devices and it generates a greater number of cross-device-compatible tests than TR and ETR

Feedback #1

“I have been looking for something like BARISTA to help me get into automation for a while”

Feedback #2

“Overall, a very interesting tool! This could save us quite some time by generating some of the tests for us”

Feedback #3

“There are a few more assertions I’d like to see. For example, testing the number of items in a ListView”

Limitations

Complex Multi-touch Gestures

Pinch in and out, custom gestures

Bitmapped Apps

Games

Sandboxing

Sensors data, networking data, camera data

Limitations

Complex Multi-touch Gestures

Pinch in and out, custom gestures

Bitmapped Apps

Games

Sandboxing

Sensors data, networking data, camera data

Limitations

Complex Multi-touch Gestures

Pinch in and out, custom gestures

Bitmapped Apps

Games

Sandboxing

Sensors data, networking data, camera data

Limitations

Complex Multi-touch Gestures

Pinch in and out, custom gestures

Bitmapped Apps

Games

Sandboxing

Sensors data, networking data, camera data

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

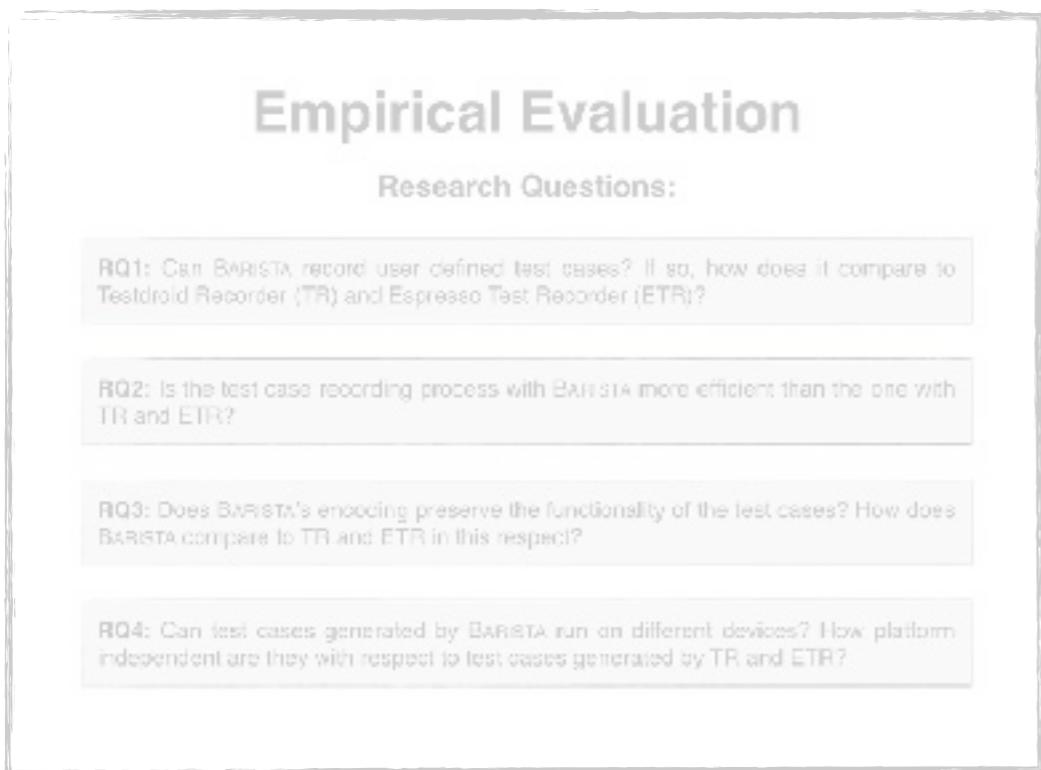
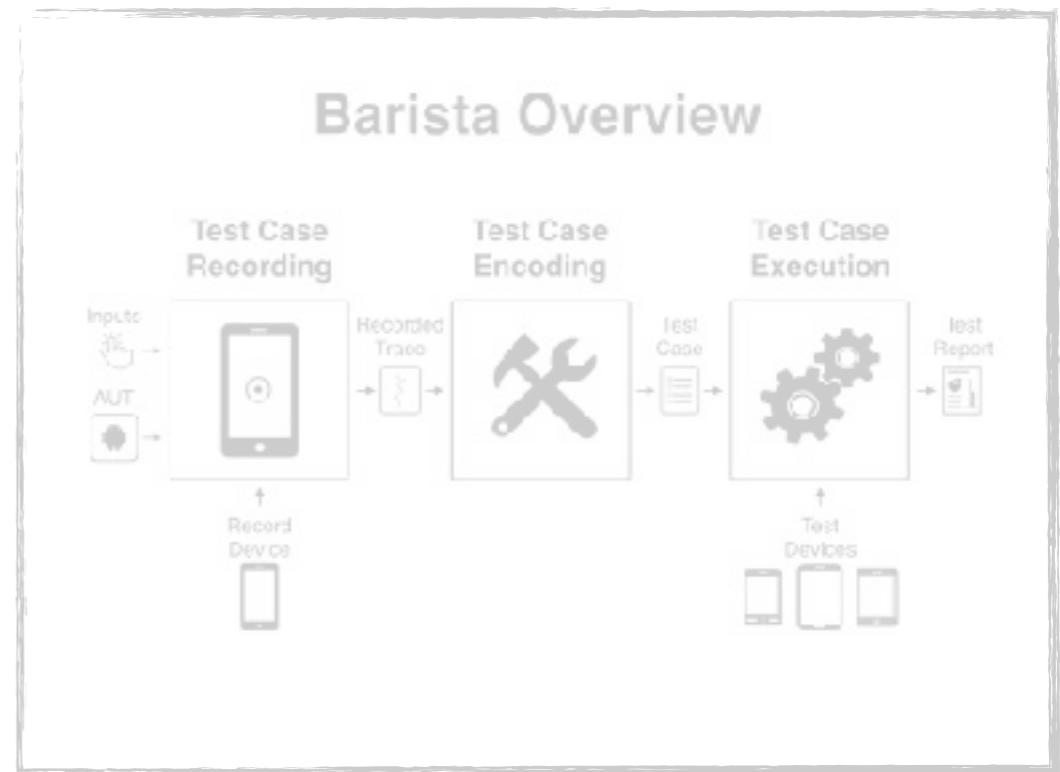
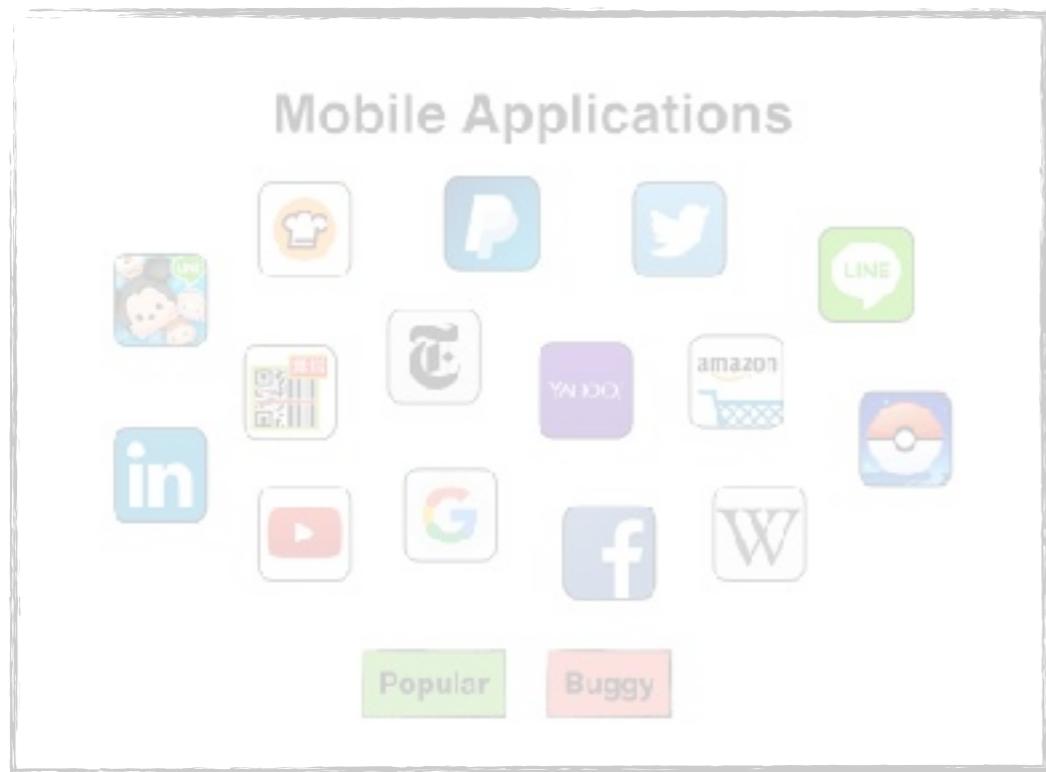
Extend assertable properties

Fuzzing

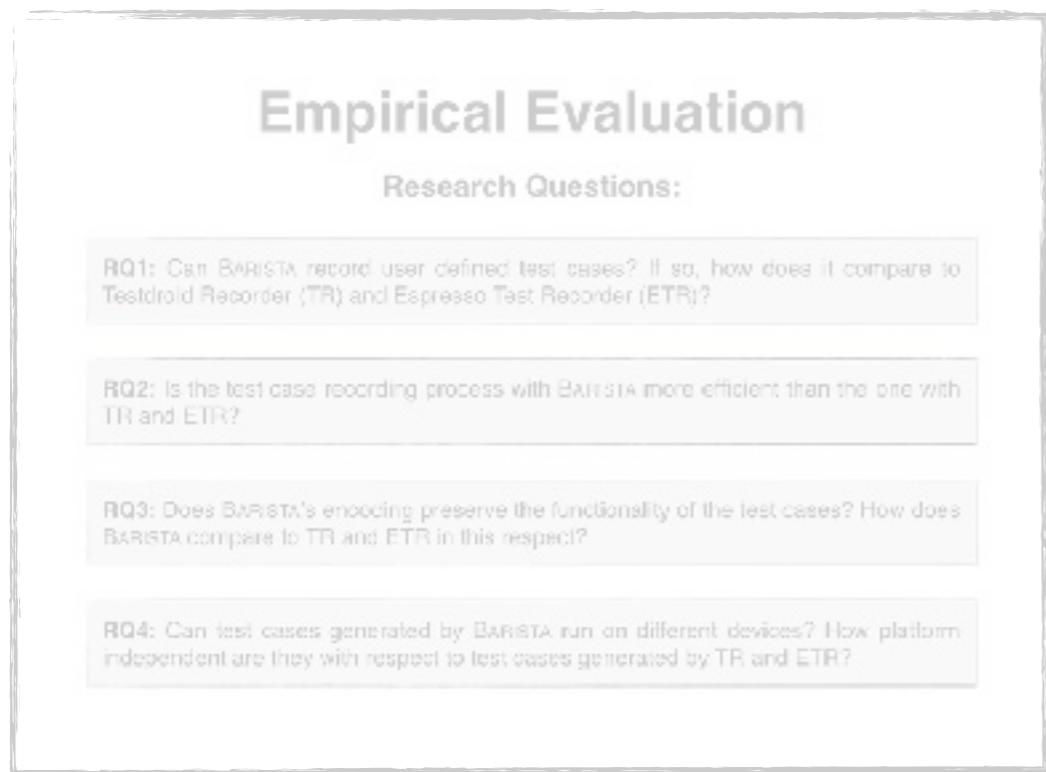
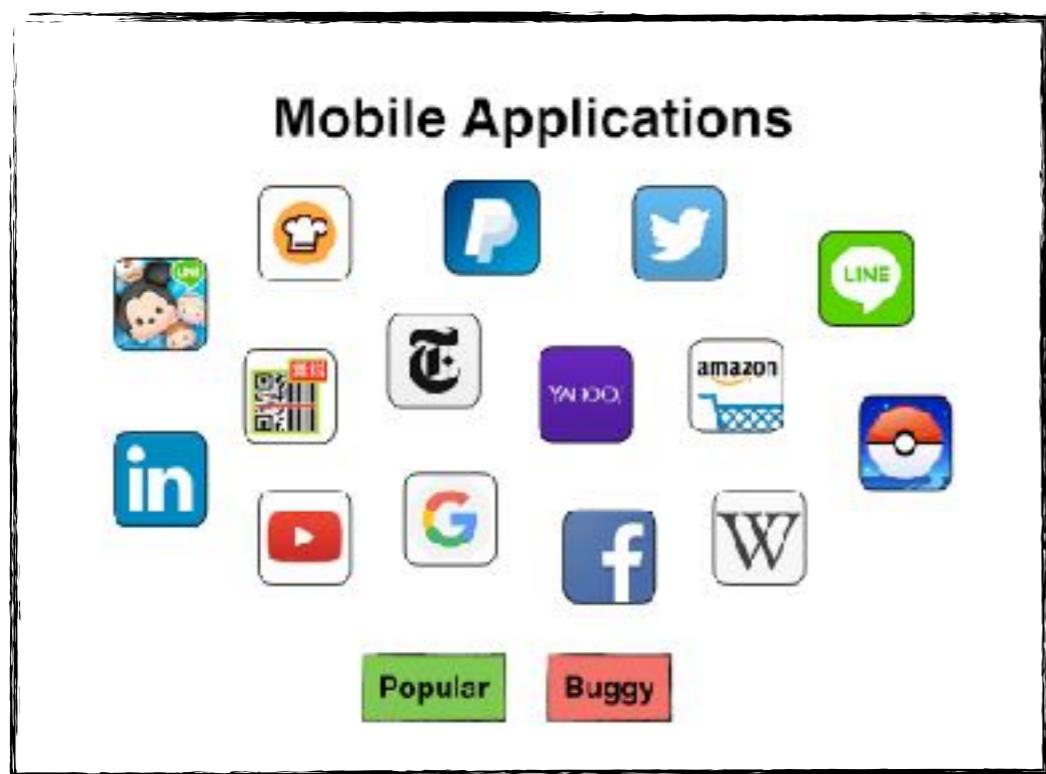
Fix broken test case during evolution

Failure diagnosis

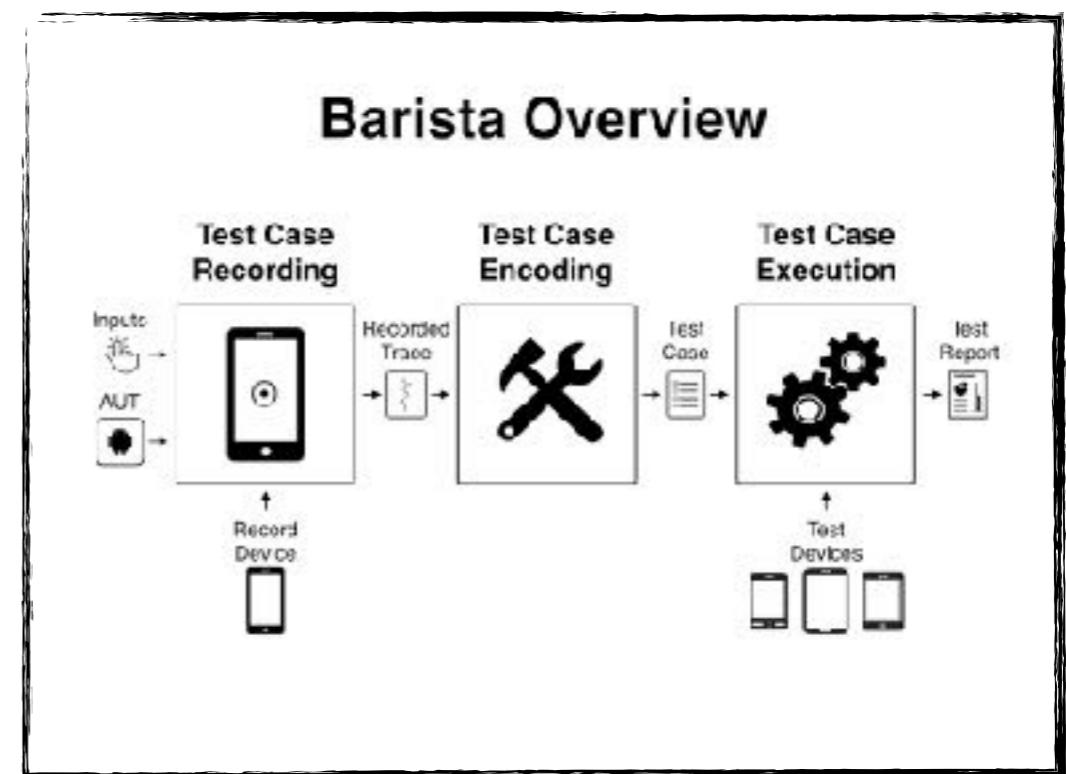
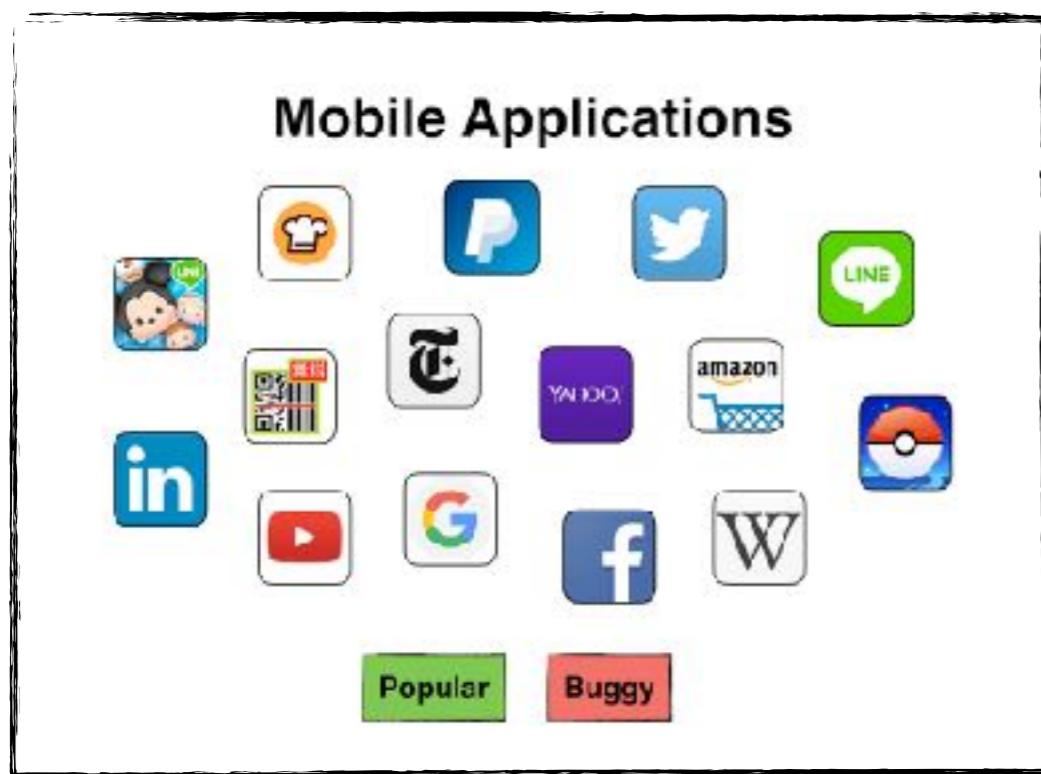
Summary



Summary



Summary



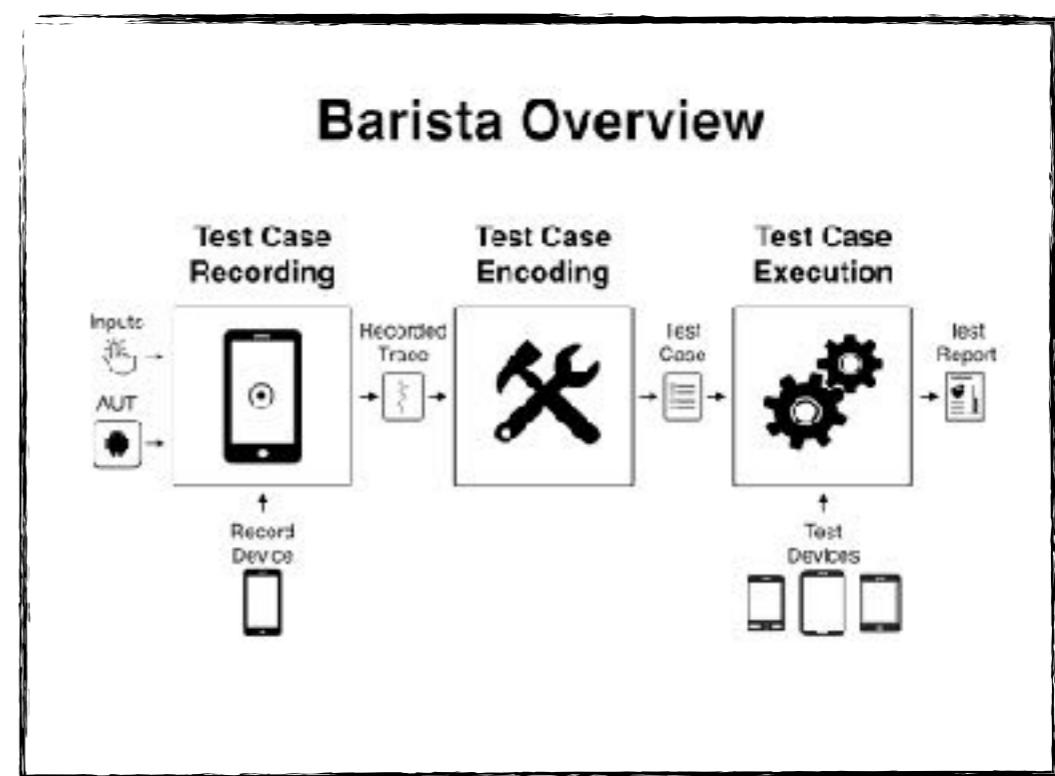
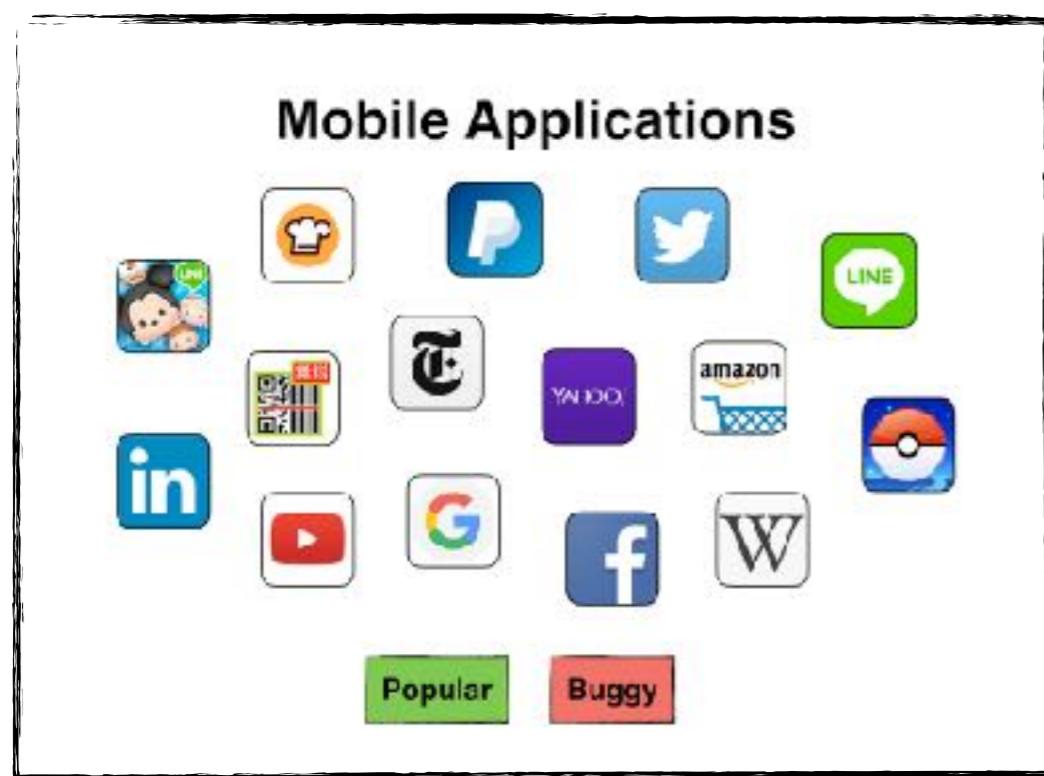
Empirical Evaluation

Research Questions:

Future Work

Fix broken test cases during evolution

Summary



Empirical Evaluation

Research Questions:

RQ1: Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2: Is the test case recording process with BAHRIA more efficient than the one with TR and ETR?

RQ3: Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TD and ETB in this respect?

RQ4: Can test cases generated by BARETA run on different devices? How platform-independent are they with respect to test cases generated by TB and ETB?

Future Work

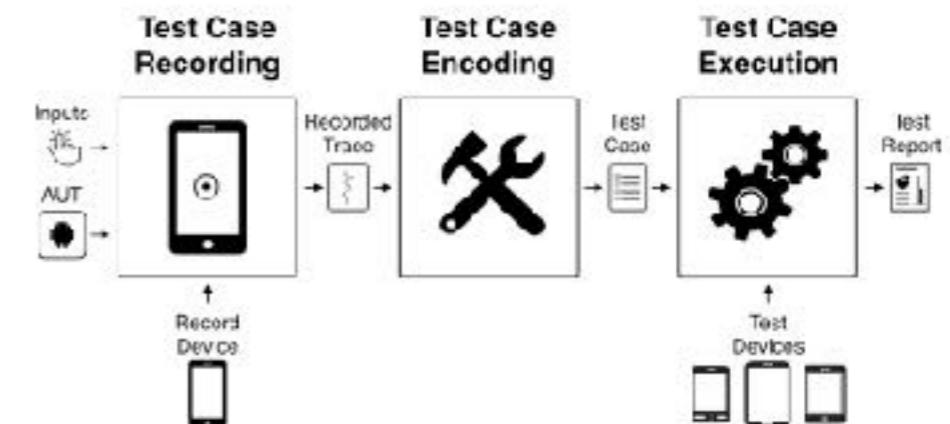
Fix broken test cases during evolution

Summary

Mobile Applications



Barista Overview



Empirical Evaluation

Research Questions:

RQ1: Can BARISTA record user defined test cases? If so, how does it compare to Testdroid Recorder (TR) and Espresso Test Recorder (ETR)?

RQ2: Is the test case recording process with BARISTA more efficient than the one with TR and ETR?

RQ3: Does BARISTA's encoding preserve the functionality of the test cases? How does BARISTA compare to TR and ETR in this respect?

RQ4: Can test cases generated by BARISTA run on different devices? How platform independent are they with respect to test cases generated by TR and ETR?

Future Work

Additional user study

Reuse repetitive action sequences

Sandboxing

Extend assertable properties

Fuzzing

Fix broken test case during evolution

Failure diagnosis

Summary

