

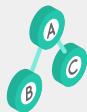


# TEST AMPLIFICATION

Enhance the quality of your test suite

By: Carl Jensen, Axel Boldt-Christmas and Alexander Volminger

# AGENDA



## 1: TEST AMPLIFICATION

What is test automation and how does it relate to DevOps?



## 3: DSPOT IN THE REAL WORLD

Evaluated against ten open-source projects



## 2: EXPLORING DSPOT

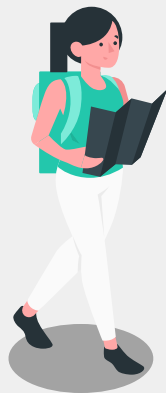
How does Dspot, a tool for test amplification work?



## 4: AUDIENCE REFLECTION

See link in zoom chat

# TEST AMPLIFICATION



“Test amplification consists of exploiting the knowledge of a large number of test methods ... in order to enhance these manually written tests with respect to an engineering goal”\*

—BENJAMIN DANGLLOT



\* Danglot, Benjamin, “Automatic Unit Test Amplification for DevOps”, PhD diss., Inria Lille–Nord Europe, 2019, HAL, (tel-02396530)

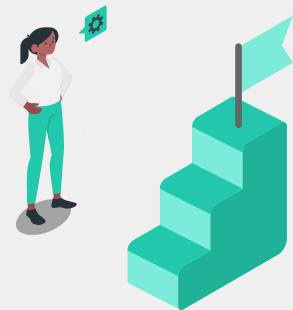
# TEST AMPLIFICATION: BETTER SUITE?



**HOW IS A TEST SUITE ENHANCED?**

# TEST AMPLIFICATION: BETTER SUITE?

## TWO COMMON APPROACHES



## CODE COVERAGE

Compare how many lines of code is executed in the code under test



## MUTATION SCORE

Compare how well the amplified test suite kills mutants



# TEST AMPLIFICATION: BETTER SUITE?

## TWO COMMON APPROACHES

### CODE COVERAGE

Compare how many lines of code is executed in the code under test

### MUTATION SCORE

Compare how well the amplified test suite kills mutants

**DISCLAIMER**  
There are many more ways to measure this!



# TEST AMPLIFICATION: HOW DOES IT WORK?



## HOW DOES IT WORK?

There are various approaches to amplify the test suite.



# TEST AMPLIFICATION: HOW DOES IT WORK?

## AMPadd

Looks at the existing test suites and generates variants of unit tests that is added

## AMPchange

Resembles AMPadd but focuses more on creating tests that targets changes on the production code.

## AMPexec

Modifies the test execution sequence in order to identify hidden dependencies

## AMPmod

Modifies and updates existing test.

# TEST AMPLIFICATION: RELEVANCE TO DEVOPS?



# TEST AMPLIFICATION: RELEVANCE TO DEVOPS?



**ENSURE QUALITY**

**+ SPEED UP THE WORKFLOW**

**= MINIMIZE COST**






# EXPLORING DSPOT

DSpot is a tool that generates missing assertions in  
JUnit tests.

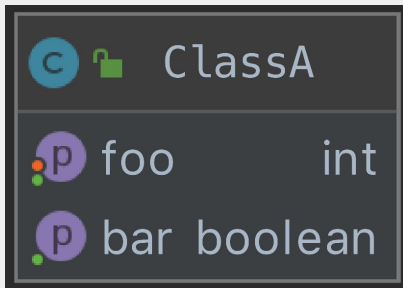
<https://github.com/STAMP-project/dspot>

# EXPLORING DSPOT

# EXPLORING DSPOT

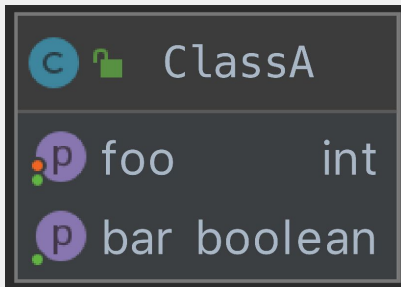
	ClassA
	foo      int
	bar    boolean

# EXPLORING DSPOT



```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

# EXPLORING DSPOT

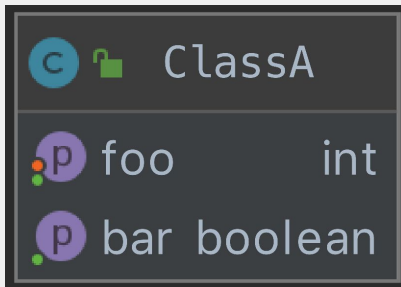


```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

```
@Test
void Test() throws Exception {
    ClassA obj = new ClassA();
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertTrue(o_Test__4);
    Assertions.assertEquals(1, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertTrue(((ClassA) (obj)).isBar());
}
```



# EXPLORING DSPOT



```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

```
@Test
void Test() throws Exception {
    ClassA obj = new ClassA();
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertTrue(o_Test__4);
    Assertions.assertEquals(1, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertTrue(((ClassA) (obj)).isBar());
}
```

# EXPLORING DSPOT

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```



```
@Test
void Test() throws Exception {
    ClassA obj = new ClassA();
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertTrue(o_Test__4);
    Assertions.assertEquals(1, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertTrue(((ClassA) (obj)).isBar());
}
```

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

Observer objects in T

Generate Assertions

Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

#### Rewrite Assertions

Observer objects in T

Generate Assertions

Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    // assertFalse(obj.isBar());
    boolean o_Test__4 = obj.isBar();
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

### Observer objects in T

Generate Assertions

Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    Instrumentation.observer(obj);
    obj.setFoo(0);
    Instrumentation.observer(obj);
    boolean o_Test__4 = obj.isBar();
    Instrumentation.observer(obj);
    Instrumentation.observer(o_Test__4);
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

Observer objects in T

**Generate Assertions**

Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertFalse(o_Test__4);
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

Observer objects in T

Generate Assertions

### Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertFalse(o_Test__4);
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

Observer objects in T

Generate Assertions

Keep improvements

### Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(1); // Apply +1 to integer
    boolean o_Test__4 = obj.isBar();
    Assertions.assertFalse(o_Test__4);
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.



# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

Observer objects in T

Generate Assertions

Keep improvements

Transform T

**Repeat N times**

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

### Observer objects in T

Generate Assertions

Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    Instrumentation.observer(obj);
    obj.setFoo(1);
    Instrumentation.observer(obj);
    boolean o_Test__4 = obj.isBar();
    Instrumentation.observer(o_Test__4);
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

## DSPOT'S MAIN LOOP

### FOR EVERY TEST T

Rewrite Assertions

Observer objects in T

**Generate Assertions**

Keep improvements

Transform T

Repeat N times

```
@Test
void Test() {
    ClassA obj = new ClassA();
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertTrue(o_Test__4);
    Assertions.assertEquals(1, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertTrue(((ClassA) (obj)).isBar());
}
```

Sources: <https://github.com/STAMP-project/dspot>

Danglot, Benjamin. (2019). Automatic Unit Test Amplification For DevOps.

# EXPLORING DSPOT

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```



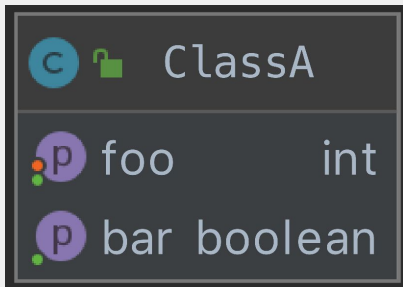
```
@Test
void Test() throws Exception {
    ClassA obj = new ClassA();
    Assertions.assertEquals(0, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertTrue(o_Test__4);
    Assertions.assertEquals(1, ((int) (((ClassA) (obj)).getFoo())));
    Assertions.assertTrue(((ClassA) (obj)).isBar());
}
```

# EXPLORING DSPOT

```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

```
@Test
void Test() throws Exception {
    ClassA obj = new ClassA();
    Assertions.assertEquals(0, (((ClassA) (obj)).getFoo()));
    Assertions.assertFalse(((ClassA) (obj)).isBar());
    obj.setFoo(1);
    boolean o_Test__4 = obj.isBar();
    Assertions.assertTrue(o_Test__4);
    Assertions.assertEquals(1, (((ClassA) (obj)).getFoo()));
    Assertions.assertTrue(((ClassA) (obj)).isBar());
}
```

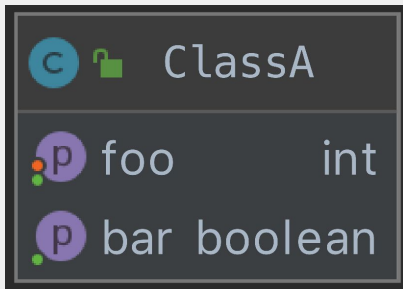
# EXPLORING DSPOT



```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

IMPROVEMENTS?

# EXPLORING DSPOT



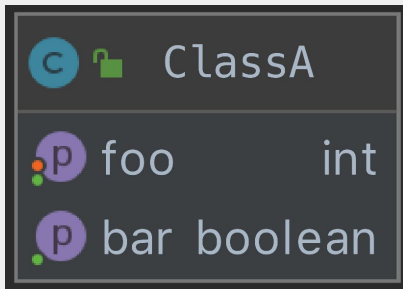
```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

## IMPROVEMENTS?

### CODE COVERAGE

70% → 100%

# EXPLORING DSPOT



```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

## IMPROVEMENTS?

### CODE COVERAGE

70% → 100%

### MUTATIONS KILLED

↑ Up by 100%



## EXPLORING DSPOT

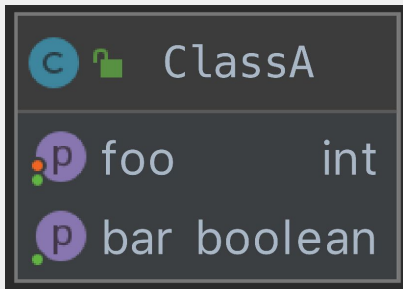
```
ClassA  
foo    int  
bar    boolean
```

```
@Test  
void Test() {  
    ClassA obj = new ClassA();  
    obj.setFoo(0);  
    assertFalse(obj.isBar());  
}
```

DISCLAIMER

IMPROVEMENT  
CODE COVERAGE  
70% → 100%  
MUTATIONS KILLED  
↑ Up by 100%

# EXPLORING DSPOT



```
@Test
void Test() {
    ClassA obj = new ClassA();
    obj.setFoo(0);
    assertFalse(obj.isBar());
}
```

## IMPROVEMENTS?

### CODE COVERAGE

12 loc → 17 loc

### MUTATIONS KILLED

1 → 2

# DSPOT IN THE REAL WORLD



# DSPOT IN THE REAL WORLD

The jsoup logo, featuring the word "jsoup" in a blue, lowercase, sans-serif font.The Twilio logo, consisting of the word "twilio" in a red, lowercase, sans-serif font.

## EVALUATED AGAINST TEN OPEN-SOURCE PROJECTS

For example Jsoup, twilio and LogBack

## 40 UNIT TESTS WERE SELECTED

4 test classes in each project

## 23/40 UNIT TESTS WERE AMPLIFIED

Added new behavior and valuable assertions on these test classes

## 13/23 WERE ACCEPTED BY THE DEVELOPERS

Amplified tests presented to the developers as pull requests

# FINAL REMARKS

1. **USEFUL FOR REAL WORLD PROJECTS**
2. **INTEGRATE WITH CI PIPELINE**

# THANKS

Feel free to check the “audience reflection poll” that will be presented during the question session. Link also in Zoom!

<https://www.menti.com/pn7d4tcfss>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

# RESOURCES

- **Link to DSpot Github:**
  - <https://github.com/STAMP-project/dspot>
- **Benjamin Danglot's Phd dissertation:**
  - Danglot, Benjamin, "Automatic Unit Test Amplification for DevOps", PhD diss., Inria Lille-Nord Europe, 2019, HAL, (tel-02396530)
- **Study, Benjamin Danglot et al:**
  - Danglot, Benjamin, et al. "Automatic test improvement with DSpot: a study with ten mature open-source projects."