

# What is the Vocabulary of

## Flaky Tests?





Gustavo Pinto



Breno Miranda



Supun Dissanayake



Marcelo d'Amorim



Christoph Treude



Antonia Bertolino











# Mhat is a flaky test



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```

Runs: 1/1

x Errors: 0

x Failures: 0



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```

Runs: 1/1

x Errors: 1





```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
```



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
```



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
```



## What do we know about flaky tests?

### An Empirical Analysis of Flaty Tests

Complete Complete Control And Compared Control States Making Control C

### 407110

Destination of Arthur States and Arthur States a

### NAME AND ADDRESS.

The beautiful property of the property of the

Transference of the second

The control of a column control of the column control of the column colu

The second second projects or provided assessment in the contract of the contr

The second section of the section of the second section of the second section of the second section of the second section of the section of the second section of the second section of the section

C DESCRIPTION THE TANK WHAT

.

**Luo @ FSE 2014** 

### An Empirical Study of Flaky Tows in Andreid Apps

Important Annie Berleite De Berg Representa de Compositorio Maria del Baltimo Palas DA

man the soft of the 1997 We have an a "solid bethere for the property of the solid besolid by the solid beside of the solid besolid below the solid besolid below the solid bethere is a s

### COLUMN SERVICE

In the contraction of the contra

may appear. Note the same the black deviation system in the property makes in terms and proper street, and the contract makes forming the street, and the contract makes the contract of the contract. Let us a resoluted an explored only on the contract or analysis property makes in 1 and 1 and 1 and 1 and the analysis property makes in 1 and 1 and 1 and 1 and the analysis of the contract and the contract of the cont

Retirus Espanyarishin is in equit ordani of its Retirus Espanyarishin is in equit ordani of its its of its annual ordani of its annual is obtained. It contains the matter is a contain in the matter is annual in earlier of its annual in equit or its ordani ordan

Final Systematics, Grazuli cité deltre con a administration de la cardina de la cardina de la cardina de la cardina de la descripción de la cardina de la cardina de la cardina del la cardina de la Cardina de administration de la cardina del la cardina del cardina del la cardina del la cardina del del cardina del cardina del la cardina del Entreto de composito del la cardina del Entreto del cardina del la cardina del la cardina del Entreto del cardina del la cardina del la cardina del Entreto del cardina del la cardina del la cardina del Cardina del cardina del la cardina del Entreto del cardina del la cardina del Entreto del cardina del la cardina del Cardina del la cardina del la cardina del la cardina del Cardina del la cardina del la cardina del la cardina del Cardina del la cardina del la cardina del la cardina del Cardina del la cardina

The second secon

Thorve @ ICSME 2018



**Empirical-based** 



## What do we know about flaky tests?

### An Empirical Analysis of Flaty Tests

### An Empirical Study of Flaky Tows in Andreid Appa

Luo @ FSE 2014

### Thorve @ ICSME 2018

**Bell @ ICSE 2018** 

### DFlakies: A Framework for Detecting and Partially Classifying Flacy Tests

Lam @ ICST 2019



**Empirical-based** 

Dynamic detection of flakiness



### What do we know about flaky tests?

### An Empirical Analysis of Flaty Tests

### An Empirical Study of Flaky Tows in Andreid Appa

**Empirical-based** 

Dynamic detection of flakiness

Static detection of flakiness

Luo @ FSE 2014

### Thorve @ ICSME 2018

DFlakies: A Framework for Detecting and Partially Classifying Flacy Tests

Lam @ ICST 2019

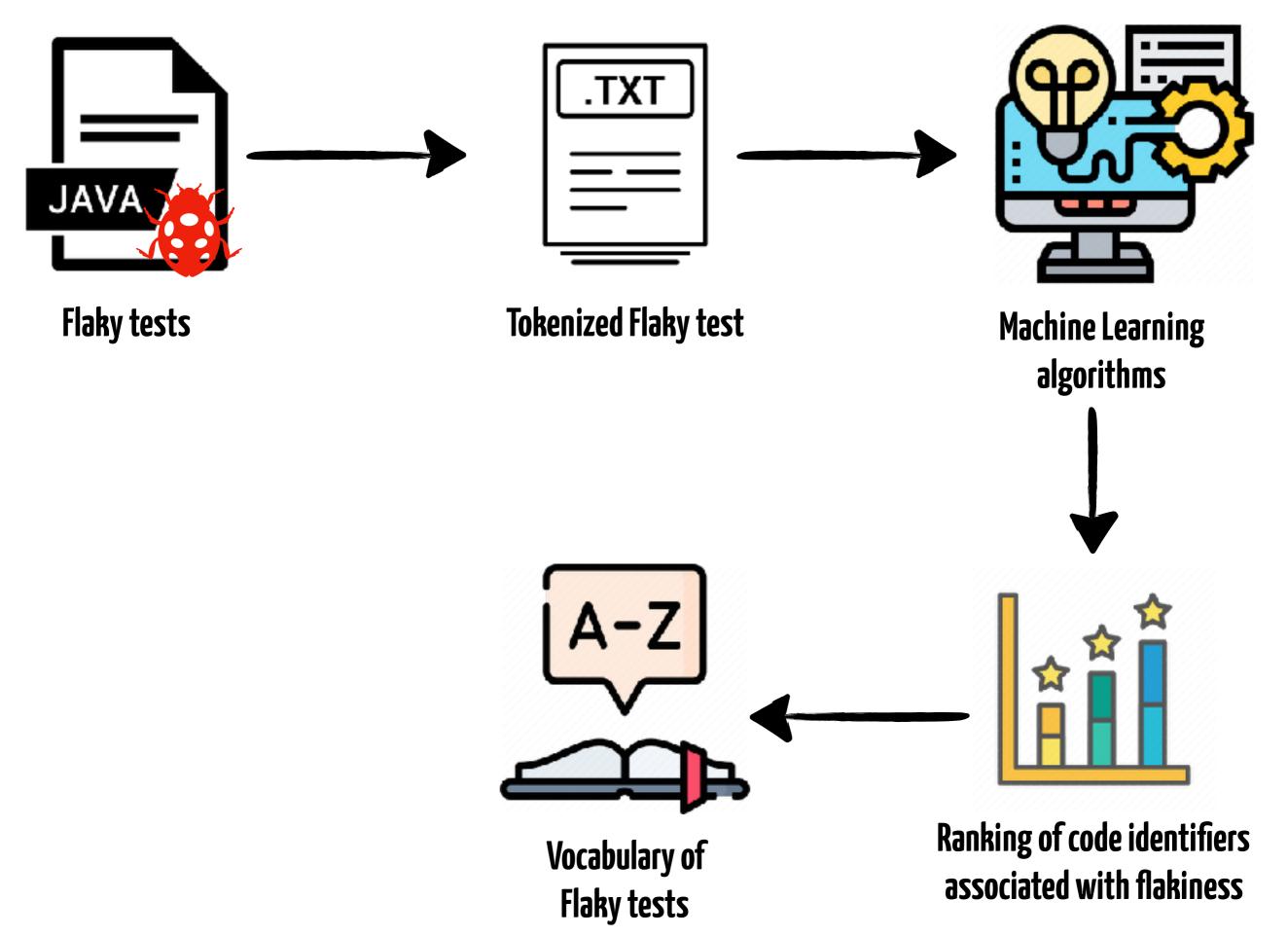


Bell @ ICSE 2018



# How to statically detect if a test code is









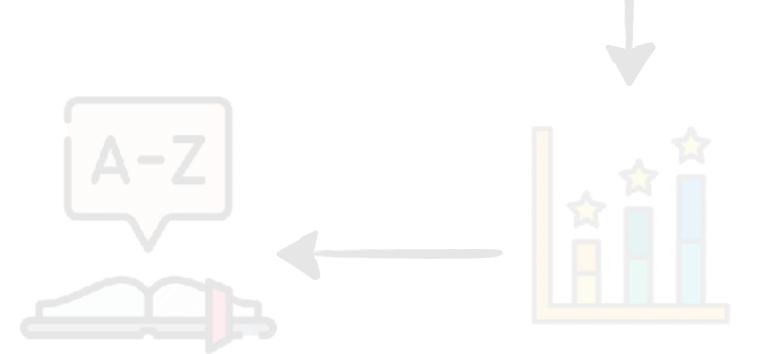
Flaky tests



Tokenized Flaky test



**Machine Learning** algorithms



**Vocabulary of** Flaky tests

Ranking of code identifiers associated with flakiness





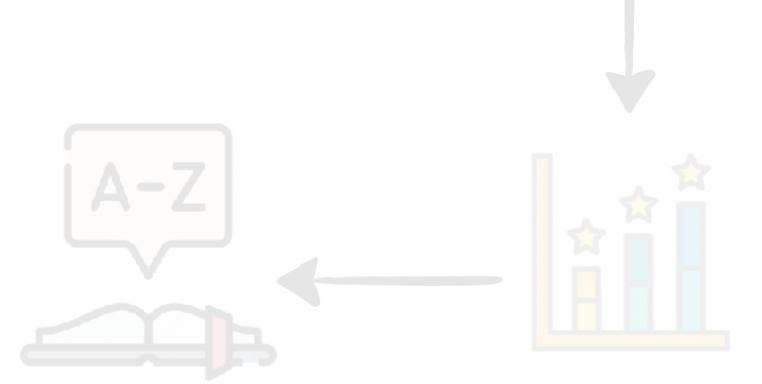
Flaky tests



**Tokenized Flaky test** 



Machine Learning algorithms



Vocabulary of Flaky tests

Ranking of code identifiers associated with flakiness





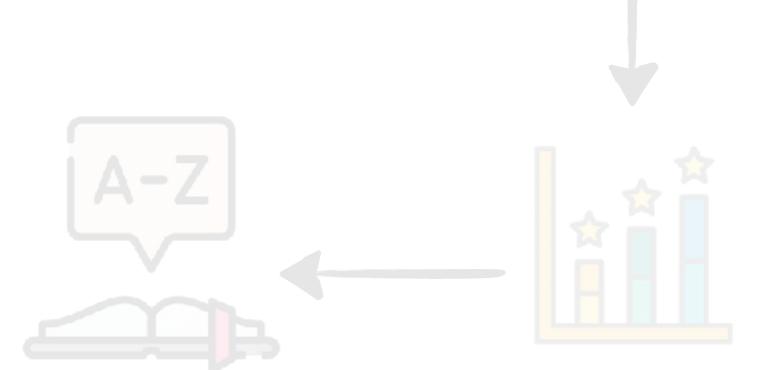
Flaky tests



Tokenized Flaky test



**Machine Learning** algorithms



**Vocabulary of** Flaky tests

Ranking of code identifiers associated with flakiness







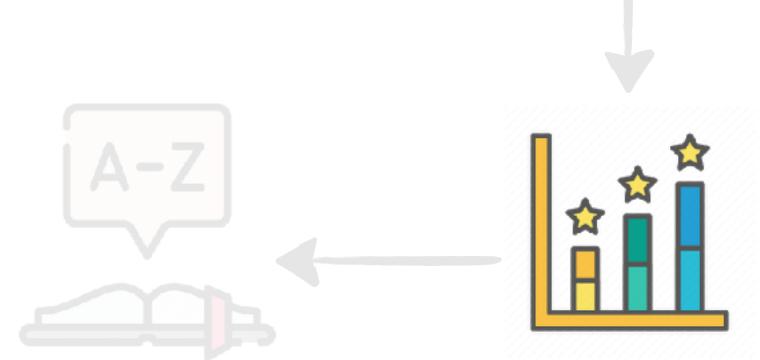
Flaky tests



Tokenized Flaky test



Machine Learning algorithms



Vocabulary of Flaky tests

Ranking of code identifiers associated with flakiness





Flaky tests



Tokenized Flaky test



Machine Learning algorithms



Vocabulary of Flaky tests

Ranking of code identifiers associated with flakiness



## Is this test flaky?

```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```



## Is this test flaky?

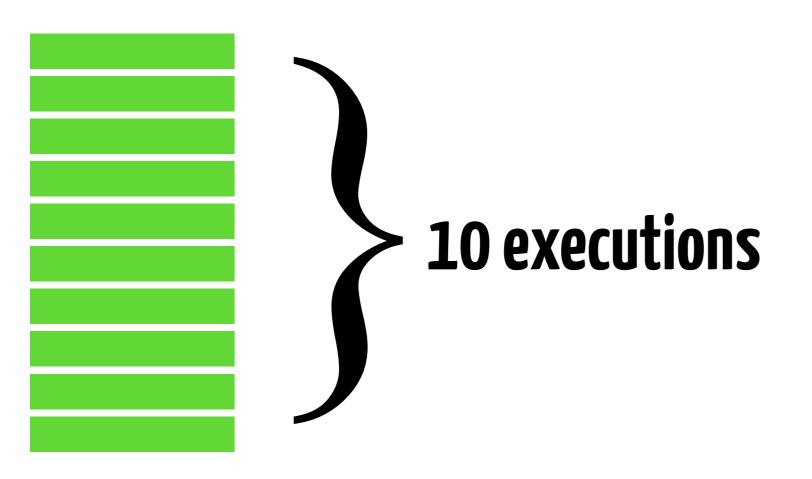
```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder (channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```

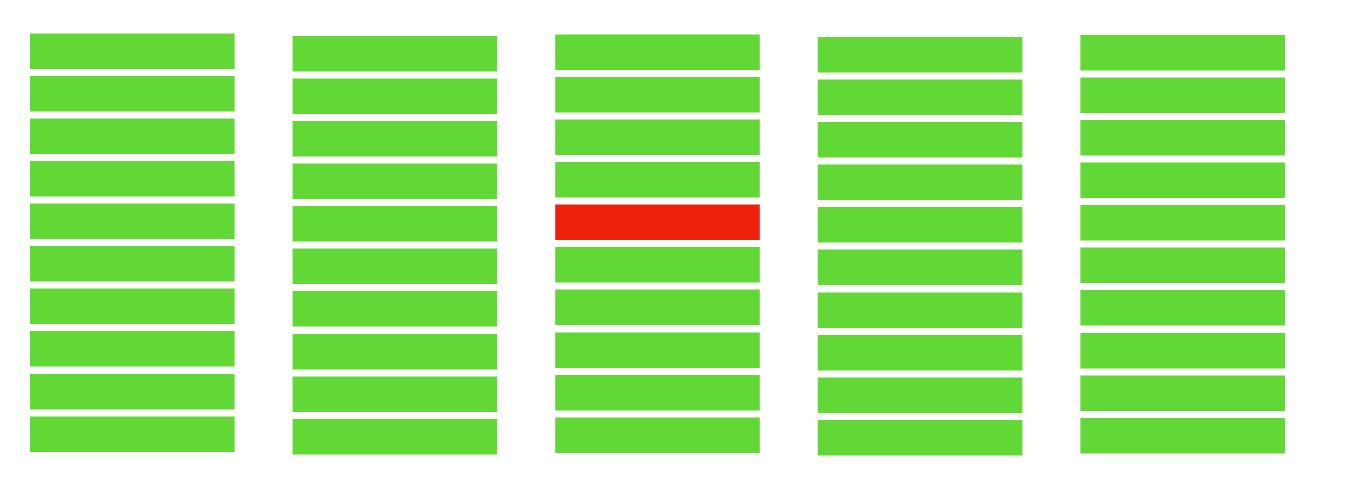
Runs: 1/1

x Errors: 0

x Failures: 0











### **GitHub Project** achilles alluxio ambari assertj-core checkstyle commons-exec dropwizard hadoop handlebars hbase hector httpcore jackrabbit-oak jimfs logback ninja okhttp oozie orbit oryx spring-boot togglz undertow wro4j zxing



### **GitHub Project** achilles alluxio ambari assertj-core checkstyle commons-exec dropwizard hadoop handlebars hbase hector httpcore jackrabbit-oak jimfs logback ninja okhttp oozie orbit oryx spring-boot togglz undertow wro4j zxing

### The same studied here

### **DEFLAKER: Automatically Detecting Flaky Tests**

}onathan Bell', Owolabi Legamen<sup>2</sup>, Michael Hilton<sup>4</sup>, Lamyee Eleussi', Tiliary Yung', and Darko Marinov \*George Mason University, Paintics, VA, USA.
\*Observety of fluorous Octoory Change ign, University of Fluorous University. Carnegie Visilan University, Pittidungh, PA, USA bolj@gma.etujlegmeni.eloc.ed2.yungi.murimvejtilinele.etu,mhlvenijema.etu.

### ARRESTMANT

Developers efter our tests to check that their latest changes to a endo repetitory die not bench any proviously working functionality. Little, any new test rationer would be test our regressions are need by the latest changes. However, come test failures may not be due to the keer altenges hit die tor needsterminism in De tests populaty relikal gebyere in The typical sary to deker listig verbalationer en Diting kom reportedly. Dekemberly, removing reliking tests met he certiy and can dow down the development cycle.

We present the first extensive evaluation of renuming failing tests and propose a new technique, called Delitesters, that denotes in a test follows is the tall of by rest without remaining one with very low reating overfixed. California mentions the coverage of locus ones counges and maximum linky any mode y talling test that dal not execute any of the changes. We deployed Definition live. In the heliciprocess of M large polyets on Twelst 1, and found 27 proviously unknown their tests in the 13 of three projects. We also can experiments on particul distorters, where Bullianum detected t, and fisher tests from 4, the follows, with a low fishe alternative (the d) ITP carry had a legior wealth to see as 200) of conferred field years than blaver's definite field; see desector.

### OCS CONCEPTS.

Software and its engineering 

Suftware testing and de-

### KEYWORDS

Software testing, flaky tests, cade coverage

### ACABA komunikansak

Journal of the Treath regions, Addisord Hiller, Langue Berner, Plany Sung, and De Lee Marker. 2005. Colfession Administrative Detection, Euley studied in Proceedings of 2007. Mr. 6th Aller without Confession and Administration Algorithm. An Administration Algorithm. 2007. See 5. 2007. See 5. 2007. 300 Fit pages 18 pp. 7 Lee app. 6 (Lee Fit 1980). His pages 18 pp. 7 Lee app. 6 (Lee Fit 1980).

Formulas as miles digital enhant age as afted on past ad this mark has record out a service as as a great of interesting protect of the expect of most of a distinct of interesting protect of the expect of most of the relation of the distinct of the miles as the distinct of the expect of the expect of the order of the control of the distinct on the distinct of the distinct of the expect of th

### 1 INTRODUCTION

Amenatol regression terring is widely rank in modern software development. We never a serveloper pushes some changes to a repeatury tests are removed on the other threatment products some functionally, likely, every new test failure would be due to the Constituting that the developer and a set the level oper could do not set the specific of the set the set of t tions or find when run on the capes service of the code

Thisy term are frequent in most large software, and create professor in development, as described to many assemblers and practitoman (1.2), S. S. T. (6.44.4), S. S. S. S. (6.5), S. A. (7.8), S. O. Provides dispersion in that half of their built fall controlled the built fall controlled the built fall controlled the fall controlled the fall controlled the fall described the fall of reported that Paleproven accounted the VE of the 1 California) deby terrificians in the Google TAP system for regression testing

When a test tidle, slove open next automated techniques that can hely determine whether the failure is due to a flaky test or to a repently introduced regression [48, 56]. The most widely-used technique to identify fishly test factories. States; is to sterain each is ling test and it ple times that of incoding the interest flames were. passes, the perior definitely flaky, but it all recent fig. the status in unknown. Here y is supported by several testing from execting  $a_{ij}$ , stations, [xi],  $[a_{ij}]$  and [xi], is the  $a_{ij}$  [xi],  $[a_{ij}]$ , and the Cough TVP seaton (43,4%) Developed do not proachedly seath. Sufficiently test maintenance articly insteads uply using Return

to kirchity that a given treat for our britishy.

There is little suspined guidance describing how to record falling. in is order to marintar the blothhood of seturning the test pass. Berma might need to be delayed to allow the cause of the following, a network orthody in he remised. He by instructions. deterministic by defination, so there is no granutize that necessing a tinky issistil dramps in origons. The performance overhead of Surrous scales with the number of finding tests — Re each failed test, Tenues, so ill seems it is consisted number of times, potentially a senjectny a John Between such sorum becoming a my failed text in externely entity when expeniations are bundrels to million of test fallanes per day. Deca theogle, with its that compete measures, does not remain that high tests on every commit (4), 20) but one or only force supported to be folio, and only comside of peak test. execution times (64)

### **Bell @ ICSE 2018**



### **GitHub Project** achilles alluxio ambari assertj-core checkstyle commons-exec dropwizard hadoop handlebars hbase hector httpcore jackrabbit-oak jimfs logback ninja okhttp oozie orbit oryx spring-boot togglz undertow wro4j zxing

### The same studied here

### **DEFLAKER: Automatically Detecting Flaky Tests**

)onathan Bell', Owolabi Legameen', Michael Hilton', Lamyee Eleussi', Tiliary Yung', and Darko Marinov \*George Mason University, Paintics, VA, USA.
\*Observety of fluorous Octoory Change ign, University of Fluorous University. Carnigle Vision University, Pittourgh, PA, USA bel jägna etu jägmen alonasityungi mainryejtilinek etu midrendena ett.

### ARRESTMANT

Developers efter our tests to check that their latest changes to a rade repetitory die nat beach any proviously working functionality. Meeting to real test rations would test out a growing a most by the latest changes I lowerer, some test failures may not be due to the keer altenges hit die tor needsterminism in De tests populaty relikal gebyere in The typical sary to deker listig verbalationer en Diting kom reportedly. Dekemberly, removing reliking tests met he certiy and can dow down the development cycle.

We present the first extensive evaluation of renuming failing tests and propose a new technique, called Delitesters, that denotes in a test follows is the tall of by rest without remaining one with very low reating overfixed. California mentions the coverage of locus ones counges and maximum linky any mode y talling test that dal not execute any of the changes. We deployed Definition two in the heliciprocess of M June projects on The 4x121, and found 27 proviously unknown their tests in the 13 of three projects. We also can experiments on particul distorters, where Bullianum detected t, and fisher tests from 4, the follows, with a low fishe alternative (the d) ITP carry had a legior wealth to see as 200) of conferred field years than blaver's definite field; see desector.

### OCS CONCEPTS

Software and its engineering 

Suftware testing and de-

### KEYWORDS

Software testing, flaky tests, cade coverage

### ACABA komunikansak

reconstitution of the control of the

Formulas as miles digital enhant age as afted on past ad this mark has record out a service as as a great of interesting protect of the expect of most of a distinct of interesting protect of the expect of most of the relation of the distinct of the miles as the distinct of the expect of the expect of the order of the control of the distinct on the distinct of the distinct of the expect of th

### 1 INTRODUCTION

Assembled regression terring is widely upolite modern software development. We cover a developer pushes some changes to a repeating best-zero monetock who has the changes backs some functionally. Blodly, every new test failure would be due to the Treat that yet that the Astroper as the artific here open could do not enoughly those believes Trainformatily some till research and the color and the color of the color of the color believes the color of the colo tions or find when run on the capes service of the code

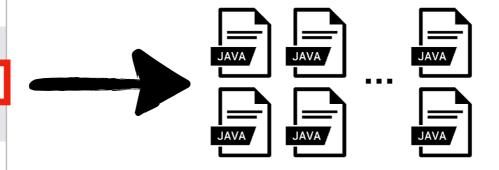
Thisy term are frequent in most large software, and create professor in development, as described to many assemblers and practi-Provides dispersion in that half of their built fall controlled the built fall controlled the built fall controlled the fall controlled the fall controlled the fall described the fall of reported that Paleproven accounted the VE of the 1 California) deby terrificians in the Google TAP system for regression testing

When a test tidle, slove open next automated techniques that can help determine whether the failure is due to a flaky test or to a repently introduced regression [48, 56]. The most widely-used technique to identify fishly test factories. States; is to sterain each is ling test and it ple times that of incoding the interest flames were. passes, the perior definitely flaky, but it all recent fig. the status in unknown. Here y is supported by several testing from execting  $a_{ij}$ , stations, [xi],  $[a_{ij}]$  and [xi], is the  $[a_{ij}]$ ,  $[a_{ij}]$ , and the Cough TVP seaton (43,4%) Developed do not proachedly seath. Sufficiently test maintenance articly insteads uply using Return

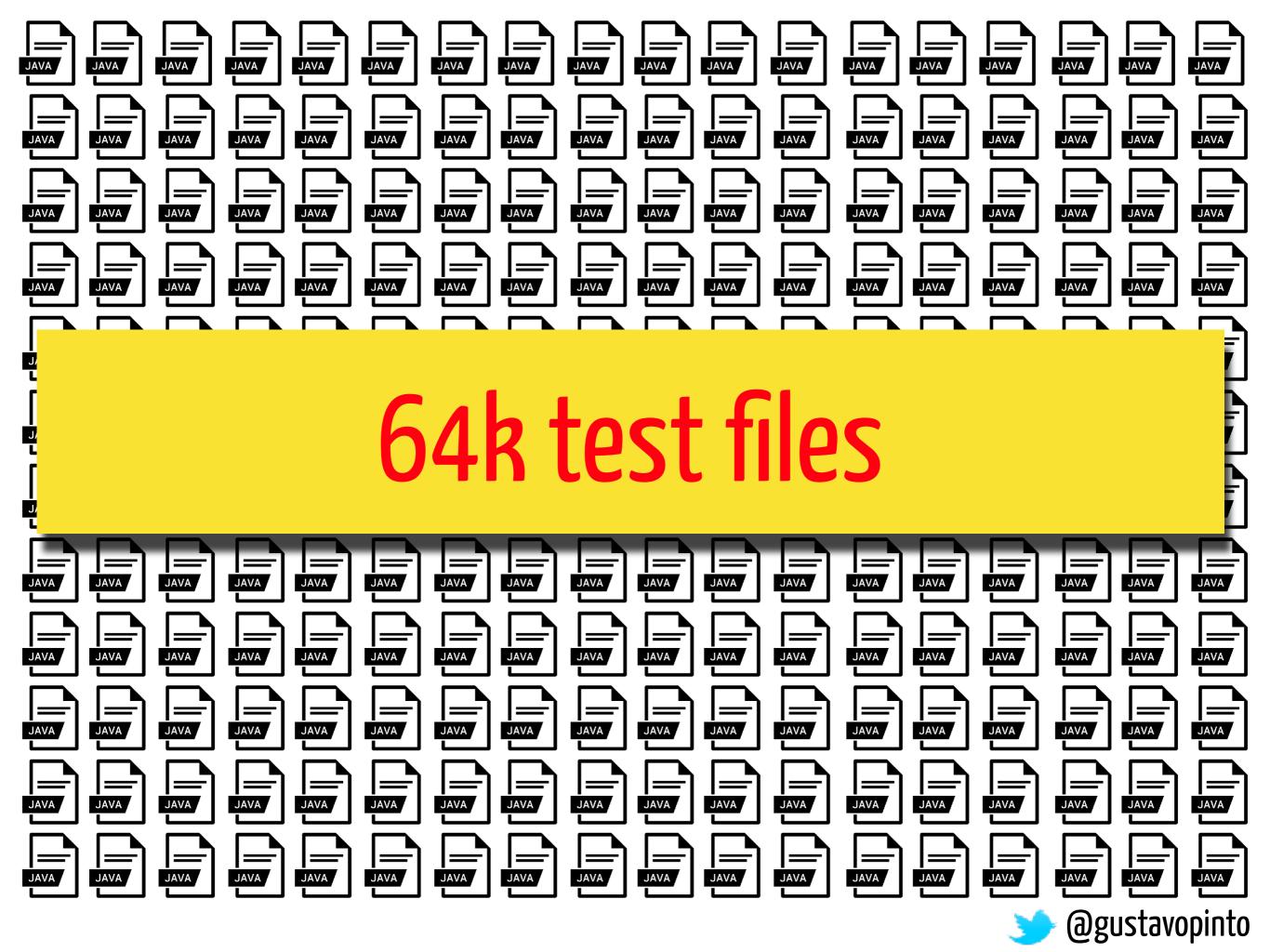
to kired by that a given test follow believe.

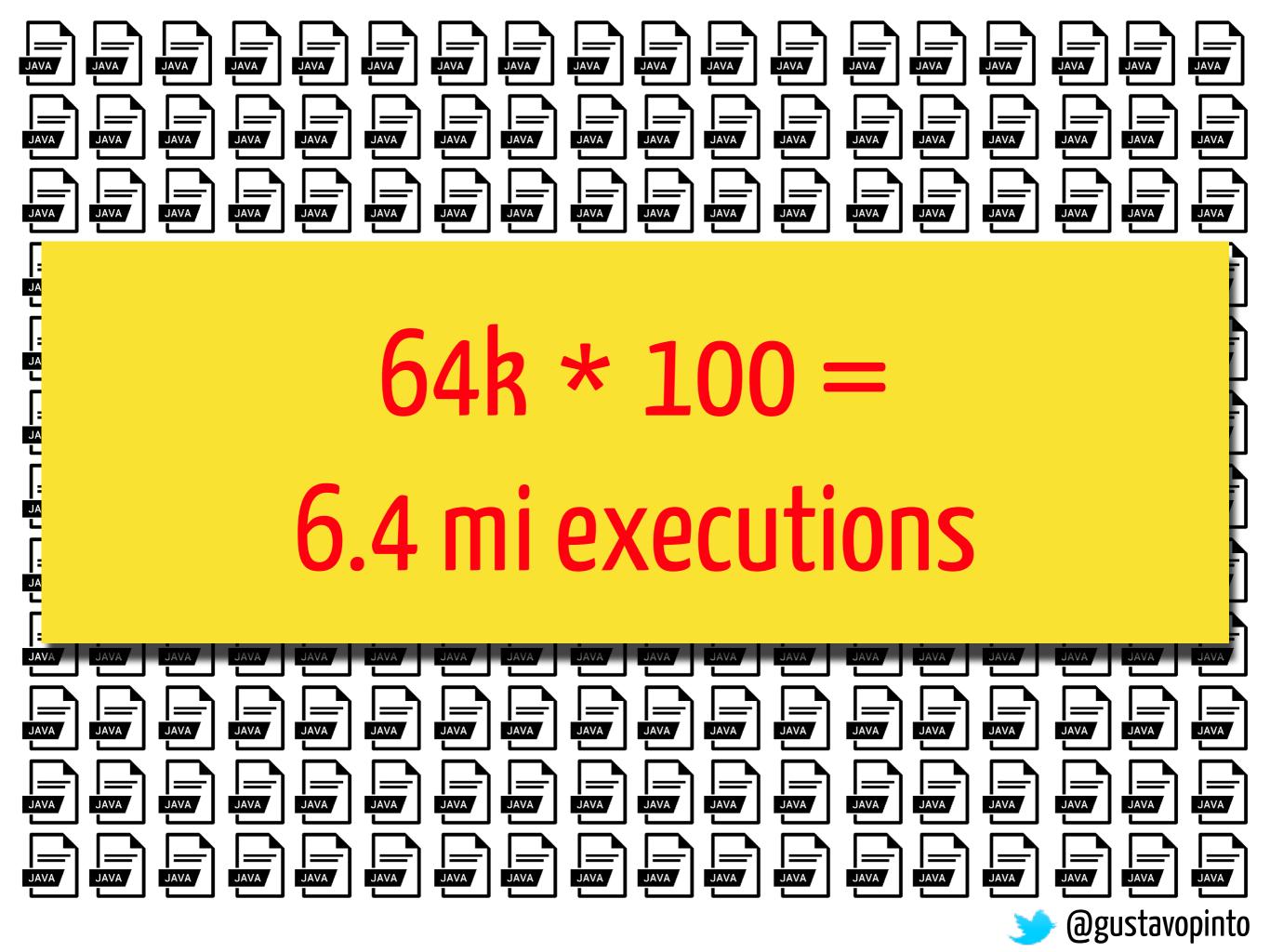
There is little empirical guidance describing how to ration falling. in is order to marintar the blothhood of seturning the test pass. Berma might need to be delayed to allow the cause of the following, a network orthody in he remised. He by instructions. deterministic by defination, so there is no granutize that necessing a tinky issistil dramps in origons. The performance overhead of Surrous scales with the number of finding tests — Re each failed test, Tenues, so ill seems it is contable number of times, potentially a senjectny a John Between such sorum becoming a my failed text in externely entity when expeniations are bundrels to million of test fallanes per day. Deca theogle, with its that compete measures, does not remain that high tests on every commit (4), 20) but one or only force supported to be folio, and only comside of peak test. execution times (64)

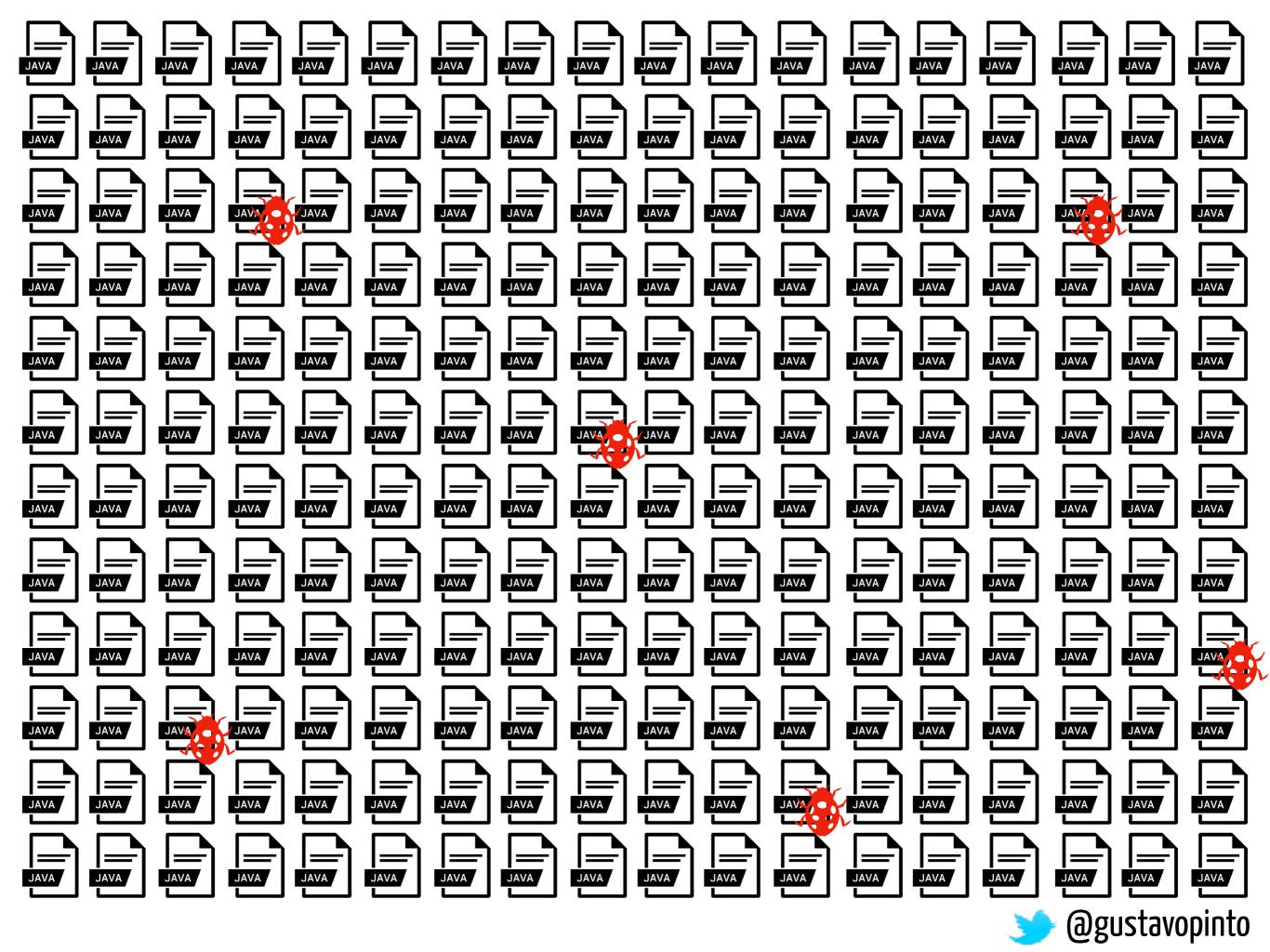
### **Bell @ ICSE 2018**

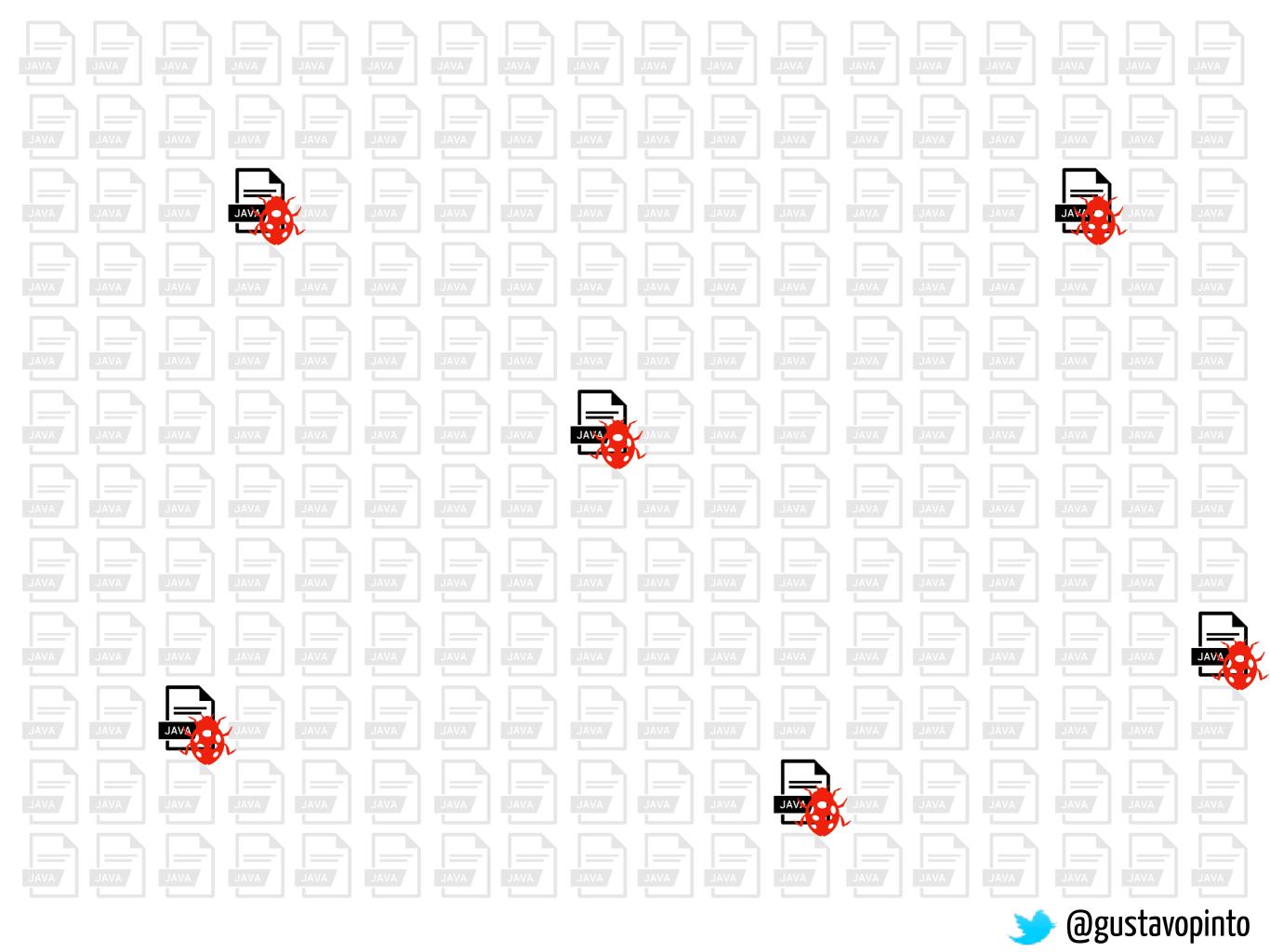


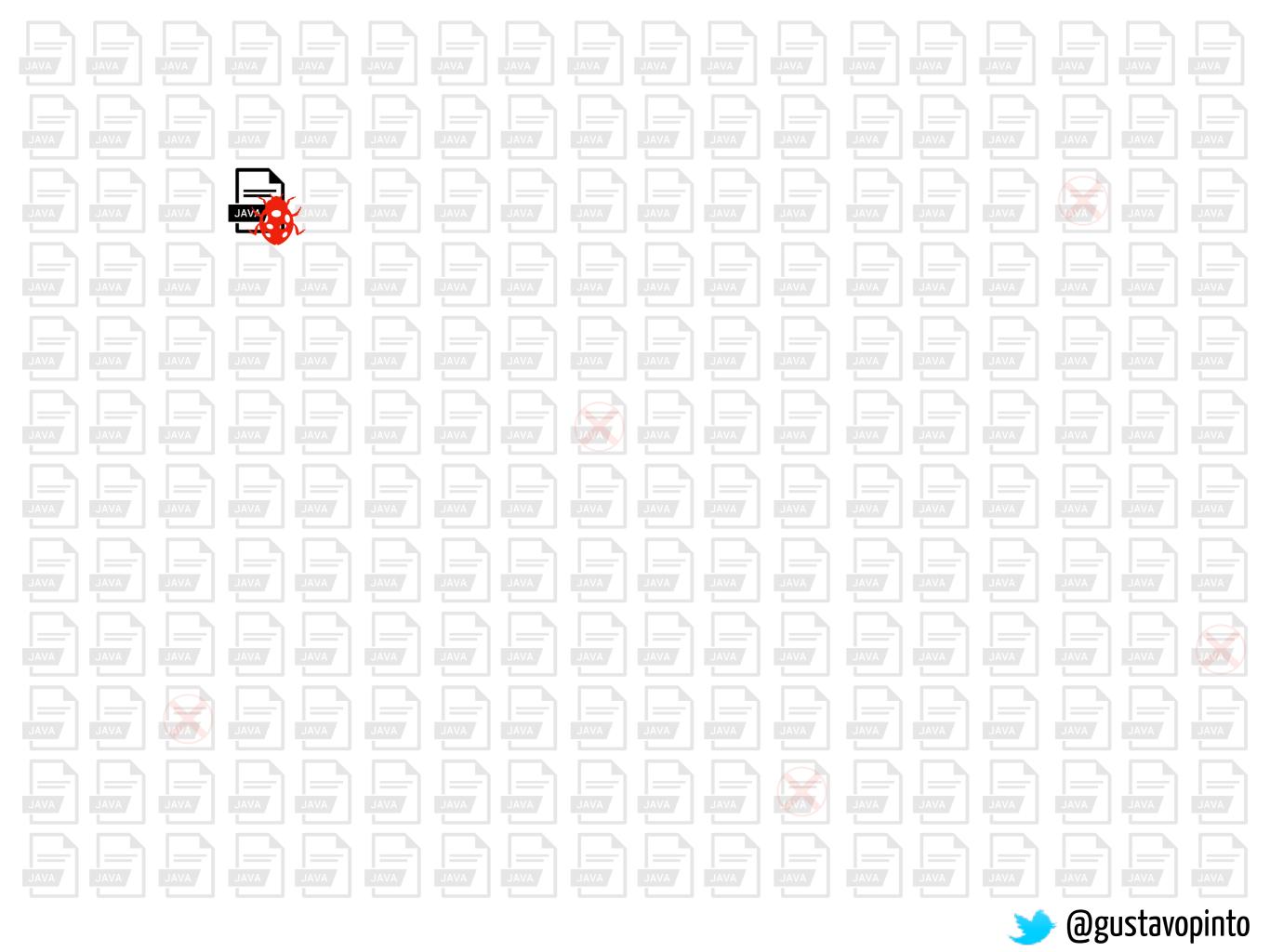


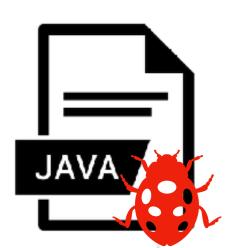












```
public class TestIdentifyEncoder {
    @Test
   public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
        encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
        Assert.assertTrue(encoder.isCompleted());
        Assert.assertEquals("stuff", s);
```



```
public class TestIdentifyEncoder {
   public void testCodingEmptySrcBuffer()
Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer.allocate(100);
        empty.flip();
        encoder.write(empty);
       encoder.write(null);
        encoder.complete();
        outbuf.flush(channel);
        final String s = channel.dump(Consts.ASCII);
       Assert.assertTrue(encoder.isCompleted());
       Assert.assertEquals("stuff", s);
```



```
public class TestIdentifyEncoder
   public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock (64):
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl metrics = new HttpTransportMetricsImpl();
        final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
        encoder.write(CodecTestUtils.wrap("stuff"));
        final ByteBuffer empty = ByteBuffer allocate(100);
        empty flip();
        encoder write(empty);
        encoder.write(null);
        encoder complete();
        outbuf flush (channel):
        final String s = channel.dump(Consts.ASCII);
       Assert assertTrue (encoder isCompleted());
       Assert.assertEquals("stuff", s);
```



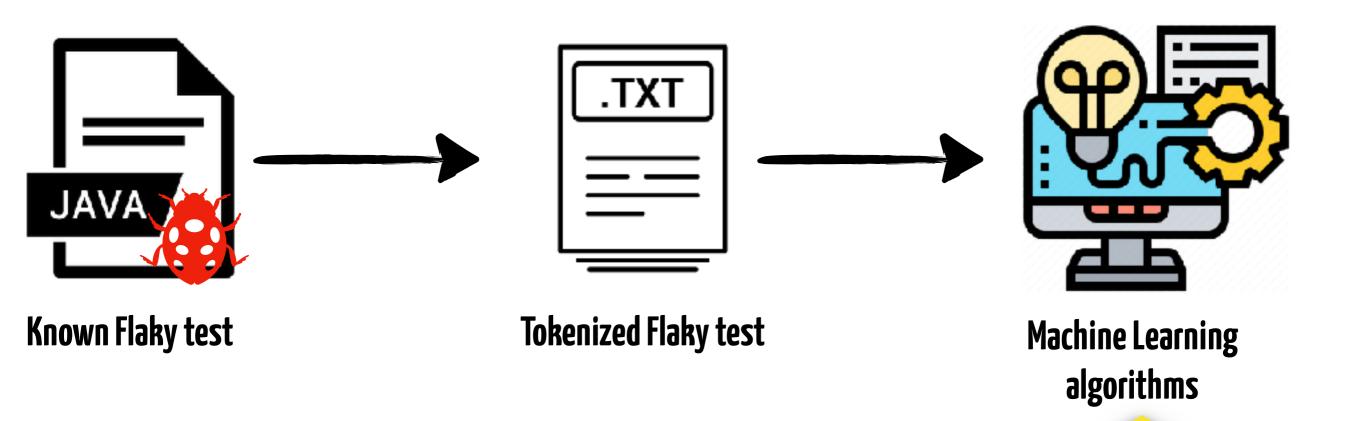
```
public class test identify encoder
   public void test coding empty src buffer() throws exception
        final writable byte channel mock channel = new writable byte channel mock (64);
        final session output buffer outbuf = new session output buffer impl(1024, 128);
        final http transport metrics impl metrics = new http transport metrics impl();
        final identity encoder encoder = new identity encoder channel outbuf metrics
        encoder.write(codec test utils wrap("stuf"));
        final byte buffer empty = byte buffer allocate(1);
        empty flip();
        encoder write(empty);
        encoder write(null);
        encoder complete();
        outbuf flush (channel)
        final string s = channel.dump(consts.ascii);
        assert assert true(encoder is completed());
        assert assert equals ("stuff",
```





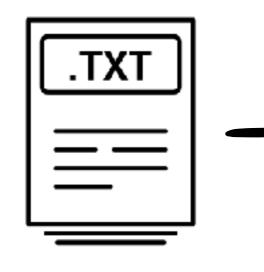
test identify encoder coding empty src buffer byte channel mock writable session output outbuf session http transport metrics impl identity codec utils wrap buffer allocate flip write complete flush dump consts ascii is completed assert equals







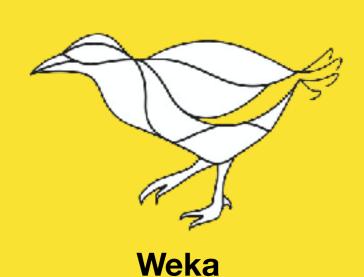
**Known Flaky test** 





**Tokenized Flaky test** 

Machine Learning algorithms



Nearest Neighbour

Support Vector Machine

**Decision Tree** 

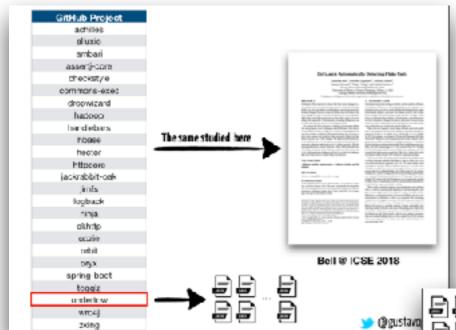
**Naive Bayes** 

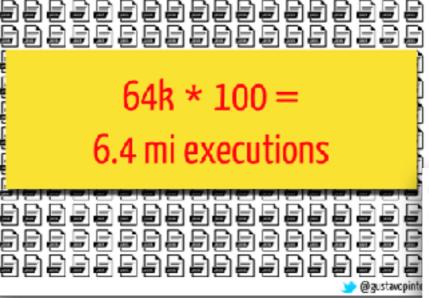
**Random Forest** 

Implementation available at: <a href="https://github.com/damorimRG/msr4flakiness/">https://github.com/damorimRG/msr4flakiness/</a>









🍅 @gustavopinto



Project	# Test	# Flaky	% Flaky
alluxio	3,034	12	0.4
hector	322	40	12.4
jackrabbit-oak	13,193	2	2
okhttp	1,682	19	19
undertow	609	2	2
wro4j	1,158	11	11

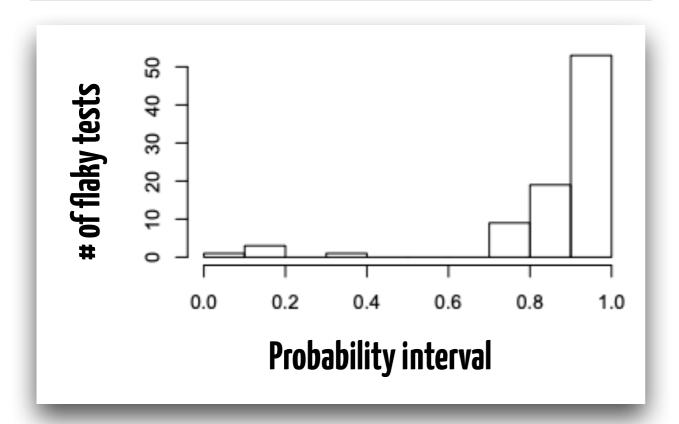
25% of the projects

have at least one flaky test

We found 86 flaky tests



Project	# Test	# Flaky	% Flaky
alluxio	3,034	12	0.4
hector	322	40	12.4
jackrabbit-oak	13,193	2	2
okhttp	1,682	19	19
undertow	609	2	2
wro4j	1,158	11	11



25% of the projects

have at least one flaky test

We found 86 flaky tests

70% (61 out of the 86)
passed more than 90%



## RQ2: How accurately can we predict test flakines?



### RQ2: Can we predict flakiness?

ML	Precision	Recall	F1	MCC	AUC
Random Forest	0.99	0.91	0.95	0.90	0.98
Decision Tree	0.98	0.88	0.89	0.77	0.91
Naive Bayes	0.93	0.80	0.86	0.74	0.93
Support Vector	0.93	0.92	0.93	0.85	0.92
Nearest Neighbour	0.97	0.88	0.92	0.85	0.93

Random Forest achieved best precision

Tuning (e.g., # of trees)

Had no performance

impact



## RQ3: What value do different features add to the classifier?



### RQ3: What's the value of different features

#### **Random Forest**

Features	Precision	Recall	F1	мсс	AUC
All features	0.99	0.91	0.95	0.90	0.98
No stemming	0.99	0.91	0.95	0.90	0.98
No Stop W. removal	0.99	0.91	0.95	0.90	0.98
No Lowercasing	0.98	0.91	0.94	0.89	0.98
No Java Keywords	0.99	0.90	0.94	0.89	0.98



### RQ3: What's the value of different features

#### **Random Forest**

Features	Precision	Recall	F1	МСС	AUC
All features	0.99	0.91	0.95	0.90	0.98
No stemming	0.99	0.91	0.95	0.90	0.98
No Stop W. removal	0.99	0.91	0.95	0.90	0.98
No Lowercasing	0.98	0.91	0.94	0.89	0.98
No Java Keywords	0.99	0.90	0.94	0.89	0.98

#### **Support Vector Machine**

Features	Precision	Recall	F1	MCC	AUC
All features	0.93	0.92	0.93	0.85	0.93
No stemming	0.93	0.92	0.93	0.85	0.93
No Stop W. removal	0.93	0.92	0.93	0.85	0.93
No Lowercasing	0.91	0.93	0.92	0.84	0.92
No Java Keywords	0.93	0.92	0.93	0.85	0.93



### RQ3: What's the value of different features

#### **Random Forest**

Features	Precision	Recall	F1	мсс	AUC
All features	0.99	0.91	0.95	0.90	0.98
No stemming	0.99	0.91	0.95	0.90	0.98
No Stop W. removal	0.99	0.91	0.95	0.90	0.98
No Lowercasing	0.98	0.91	0.94	0.89	0.98
No Java Keywords	0.99	0.90	0.94	0.89	0.98

### impact No performance

#### **Support Vector Machine**

Features	Precision	Recall	F1	MCC	AUC
All features	0.93	0.92	0.93	0.85	0.93
No stemming	0.93	0.92	0.93	0.85	0.93
No Stop W. removal	0.93	0.92	0.93	0.85	0.93
No Lowercasing	0.91	0.93	0.92	0.84	0.92
No Java Keywords	0.93	0.92	0.93	0.85	0.93



## RQ4: Which test code identifiers are strongly associated with test flakiness?

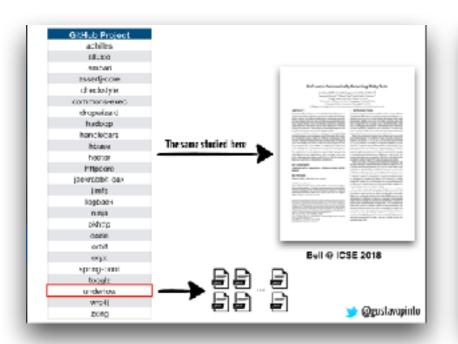


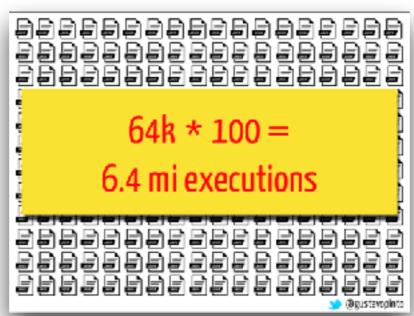
### RQ4: What's the vocabulary?

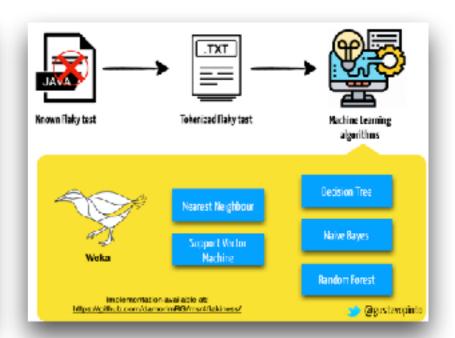
		_		า-Flaky
inf. gain	# test	# projects	# test	# projects
0.20	524	2	4	1
0.14	406	4	8	2
0.14	552	9	52	4
0.13	387	3	8	2
0.13	274	1	0	0
0.13	371	2	7	1
0.11	307	1	0	0
0.11	287	4	1	1
0.10	258	1	0	0
0.10	147	2	6	2
0.09	207	1	0	0
0.08	246	2	2	2
0.08	296	2	18	1
0.07	207	2	0	0
0.07	200	1	0	0
0.07	367	4	67	3
	0.20 0.14 0.13 0.13 0.13 0.11 0.11 0.10 0.10 0.09 0.08 0.08 0.07 0.07	nf. gain# test0.205240.144060.145520.133870.132740.133710.112870.102580.101470.092070.082460.082960.072070.07200	0.14       406       4         0.14       552       9         0.13       387       3         0.13       274       1         0.13       371       2         0.11       307       1         0.11       287       4         0.10       258       1         0.10       147       2         0.09       207       1         0.08       246       2         0.08       296       2         0.07       207       2         0.07       200       1	Inf. gain         # test         # projects         # test           0.20         524         2         4           0.14         406         4         8           0.14         552         9         52           0.13         387         3         8           0.13         274         1         0           0.13         371         2         7           0.11         307         1         0           0.11         287         4         1           0.10         258         1         0           0.10         147         2         6           0.09         207         1         0           0.08         246         2         2           0.08         296         2         18           0.07         207         2         0           0.07         200         1         0

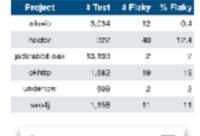
Many of them associated with remote tasks and/or queue events

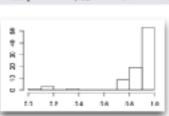












25% of the projects have at least one flatly test	

We found 86 flally tests

70% (alexandes) passed more than 90%

#### RQ3: What's the value of different features

Random Forest					
Features	Precision	Recall	Ff	MCC	AUG
All features	0.99	0.91	0.95	0.90	0.98
No stemming	0.99	0.91	0.95	0.90	0.98
No Stop W. removal	0.99	0.91	0.95	0.90	0.98
No Lowercasing	0.98	0.91	0.94	0.29	0.98
No Java Keyworch	0.99	0.90	0.94	0.29	0.98
Support Vector Mach	ine				
Feetures	Precision	Rectil	Ft	MCC	AUG
All fratures	0.50	(1.92	0.93	0.05	0.83
No stemming	0.501	(1.102	0.93	0.75	0.93
No Stop W. removal	0.93	0.92	0.33	0.85	0.33
No Lowersasing	0.571	(1.50)	0.92	0.34	0.522

0.83

0.93

No Java Keywords

No performance impact

#### RQ2: Can we predict flakiness?

ML	Precision	Recoil	Pt	MCG	AUG	
Random Forest	0.99	0.91	0.95	0.90	0.98	Bondon Forest
Dycision Tree	0.98	0.88	0.89	0.77	0.91	Random Forest
Naive Bayes	0.93	0.80	0.86	0.74	0.93	ach eved bast precision
Support Vector	0.33	98.0	0.33	0.35	0.32	
Veorest Naighbour	0.97	58.D	9.92	Q:35	0.93	Tuning (a.g. = oftress) Had no performance impact
						🍑 Øgustavopin

#### RQ4: What's the vocabulary?

	Non-Flaky		Flaky			
	# projects	# test	# projects	# test	inf. gein	Features.
Job, table, and actio	1	4	2	624	0.20	job
	52	8	4	400	0.14	Hibbs
	4	92	9	352	0.14	id
	2	8	3	387	0.13	action
	0	0	1	274	0.18	ocule
	1	7	2	371	0.18	ventors
Many of them associated wi remote tooks and/or queue events	0	0	1	307	0.11	coord
	1	- 1	4	387	0.11	getic
	0	0	1	258	0.10	coordinator
	2	6	2	147	0.10	XIII
	0	0	1	207	0.09	workflow
	2	2	2	246	0.08	gelstatus
	1	18	2	296	0.08	record
	0	0	2	207	0.07	jpa -
	0	0	1	200	0.07	passervice
	3	67	4	367	0.07	gervice

🟏 @gustavopinto

