What is the Vocabulary of Flaky Tests?






Gustavo Pinto

Breno Miranda

Supun Dissanayake

Marcelo d'Amorim

Christoph Treude

Antonia Bertolino

@gustavopinto

What's a flaky test?

```java
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);
    }
}
```

@gustavopinto

What's 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);
    }
}
```

Runs: 1/1 x Errors: 0 x Failures: 0

@gustavopinto

What's 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);
    }
}
```

Runs: 1/1 x Errors: 1 x Failures: 0

@gustavopinto

What's a flaky test?

```java
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);
    }
}
```

@gustavopinto

What's 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);
    }
}
```

@gustavopinto

What's 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);
    }
}
```

@gustavopinto

What do we know about flaky tests?

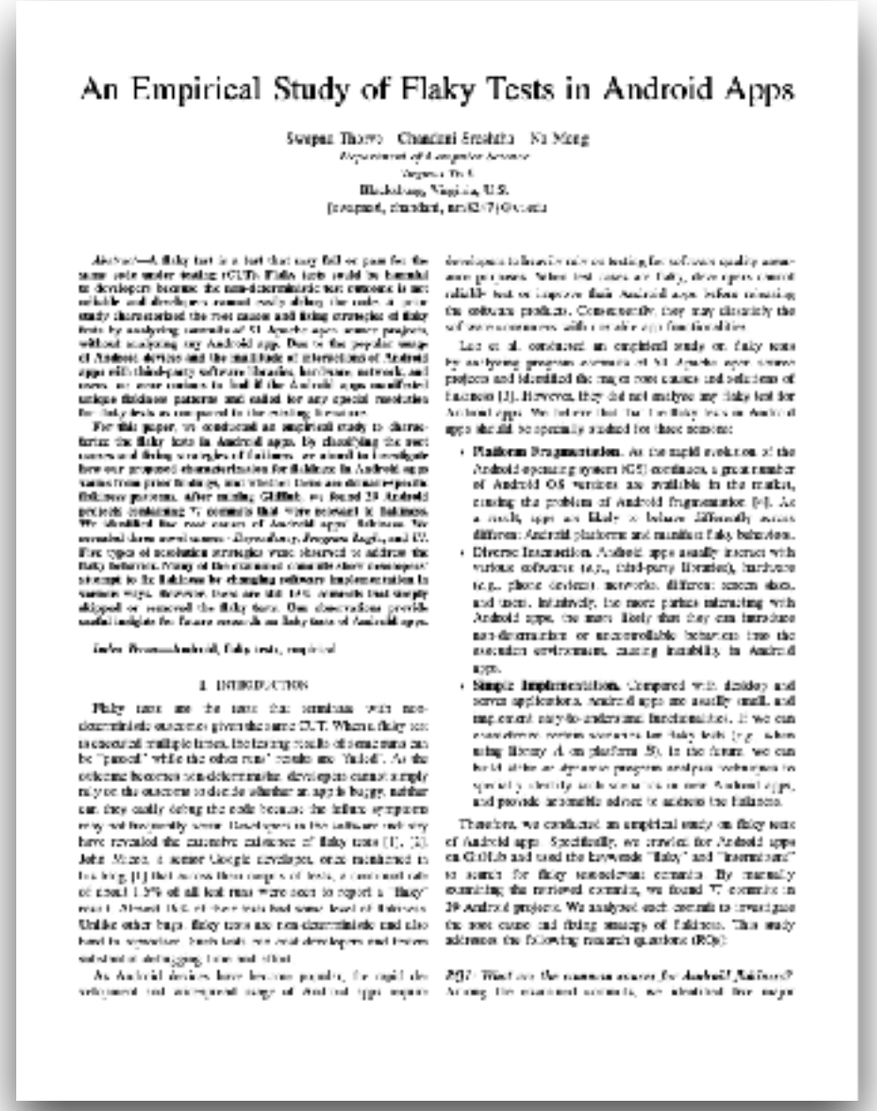
An Empirical Analysis of Flaky Tests

Qingzhou Luo, Farah Hariri, Lamyaa Eloussi, Darko Marinov
Department of Computer Science, University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
(luo8, hariri2, eloussi2, marinov)@illinois.edu

ABSTRACT

1. INTRODUCTION

Luo @ FSE 2014



An Empirical Study of Flaky Tests in Android Apps

Swapna Thorve, Chandani Sreshtha, Ni Meng
Department of Computer Science
Virginia Tech
Blacksburg, Virginia, U.S.
(swapnat, chandani, nm8@vt.edu)

I. INTRODUCTION

Thorve @ ICSME 2018

@gustavopinto

What do we know about flaky tests?

An Empirical Analysis of Flaky Tests

Qingzhou Luo, Farah Hariri, Lamyaa Eloussi, Darko Marinov
Department of Computer Science, University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{luoq2, hariri2, eloussi2, marinov}@illinois.edu

ABSTRACT

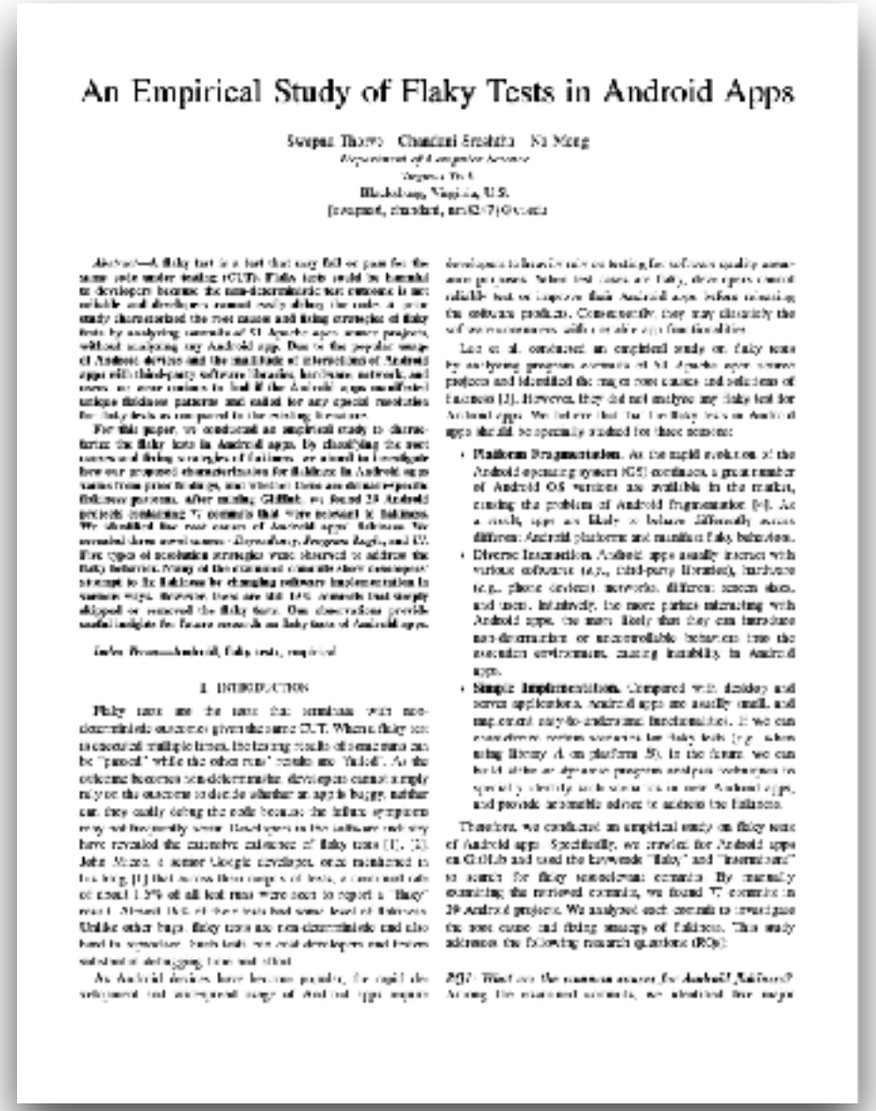
1. INTRODUCTION

Luo @ FSE 2014

- **Analyzed 201 commits that likely fix flaky tests in 51 open-source projects.**
- **Classify the root causes of flaky tests**
- **Identify approaches that could manifest flaky behavior**
- **Describe strategies to fix flaky tests**

@gustavopinto

What do we know about flaky tests?



An Empirical Study of Flaky Tests in Android Apps

Thorve @ ICSME 2018

- **Xxx**

@gustavopinto

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);
    }
}
```

@gustavopinto

Runs: 1/1 Errors: 0 Failures: 0

Runs: 1/1 Errors: 0 Failures: 0

```
@Test
public void testCodingEmptySrcBuffer() throws Exception {
    final WritableByteChannelMock channel = new WritableByteChannelMock(64);
    final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
```

Runs: 1/1 Errors: 0 Failures: 0

```
    final IdentityEncoder encoder = new IdentityEncoder(channel, outbuf, metrics);
    encoder.write(CodecTestUtils.wrap("stuff"));
```

Runs: 1/1 Errors: 0 Failures: 0

```
    final ByteBuffer empty = ByteBuffer.allocate(100);
    empty.flip();
    encoder.write(empty);
    encoder.write(null);
```

Runs: 1/1 Errors: 1 Failures: 0

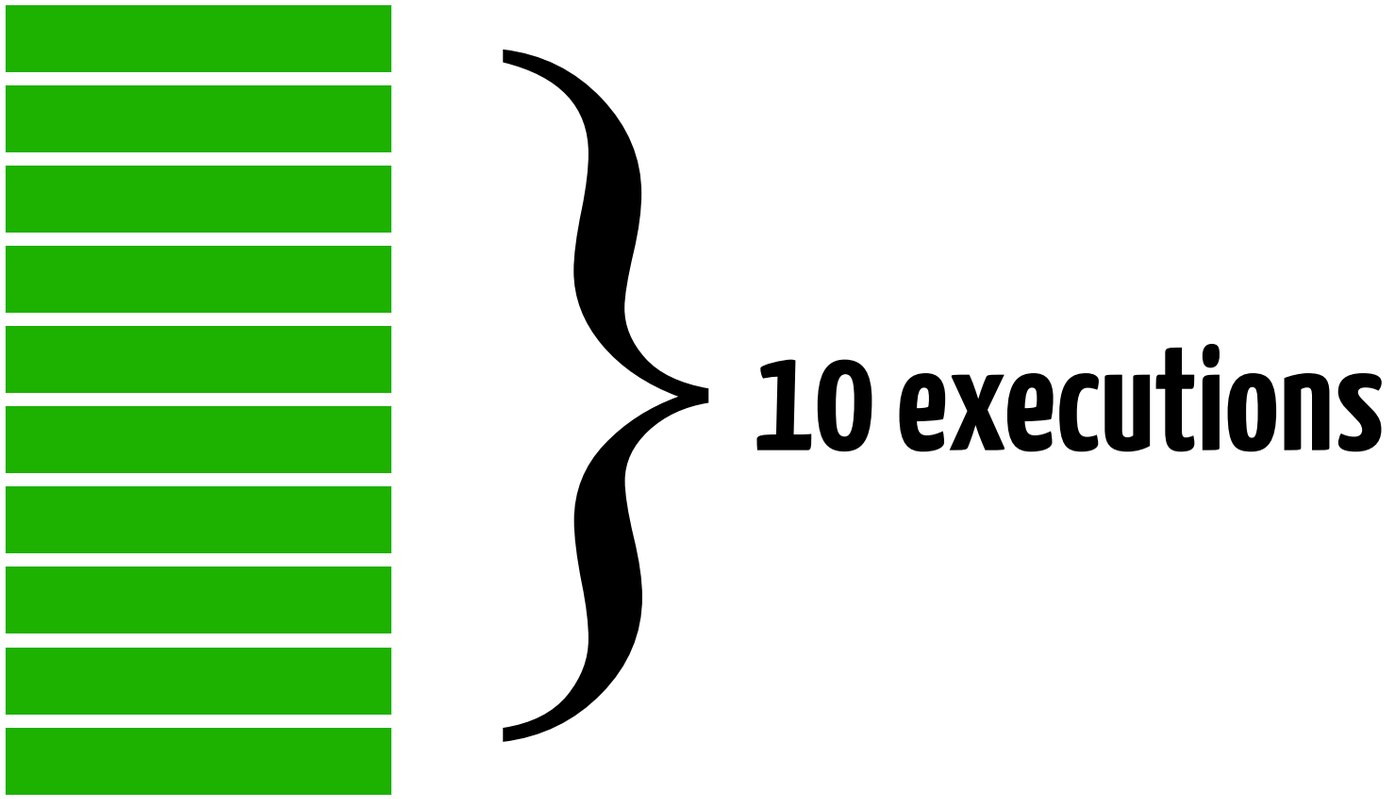
```
    outbuf.flush(channel);
    final String s = channel.dump(Consts.ASCII);
```

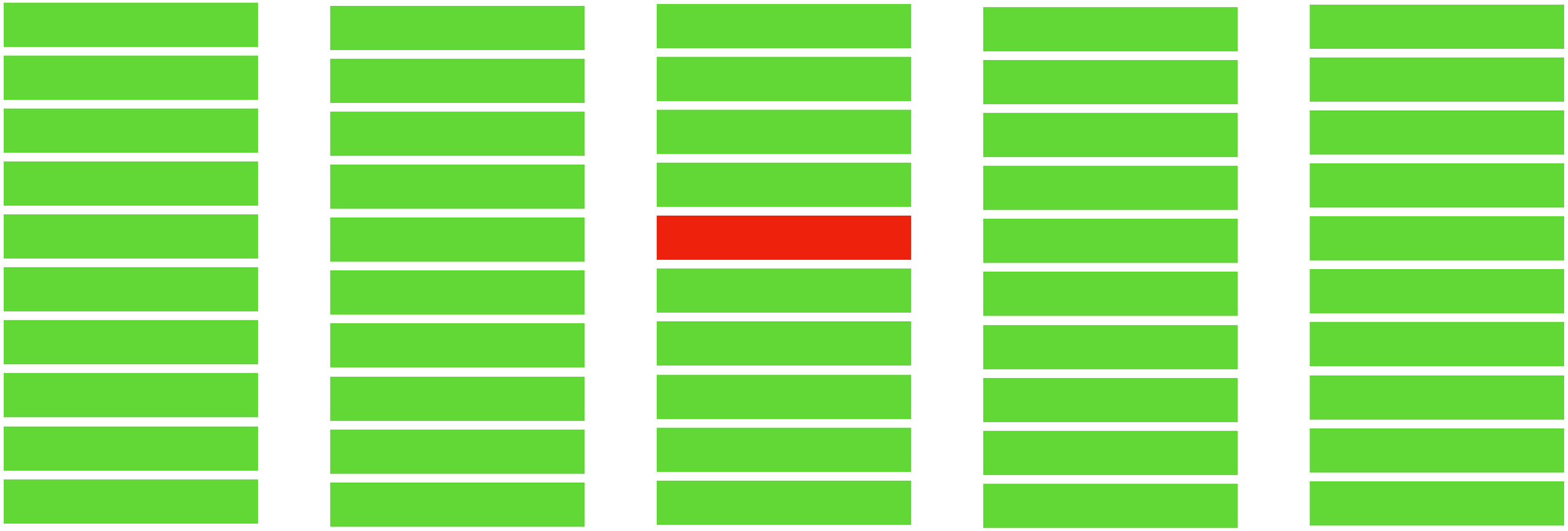
Runs: 1/1 Errors: 0 Failures: 0

```
    Assert.assertEquals("stuff", s);
}
```

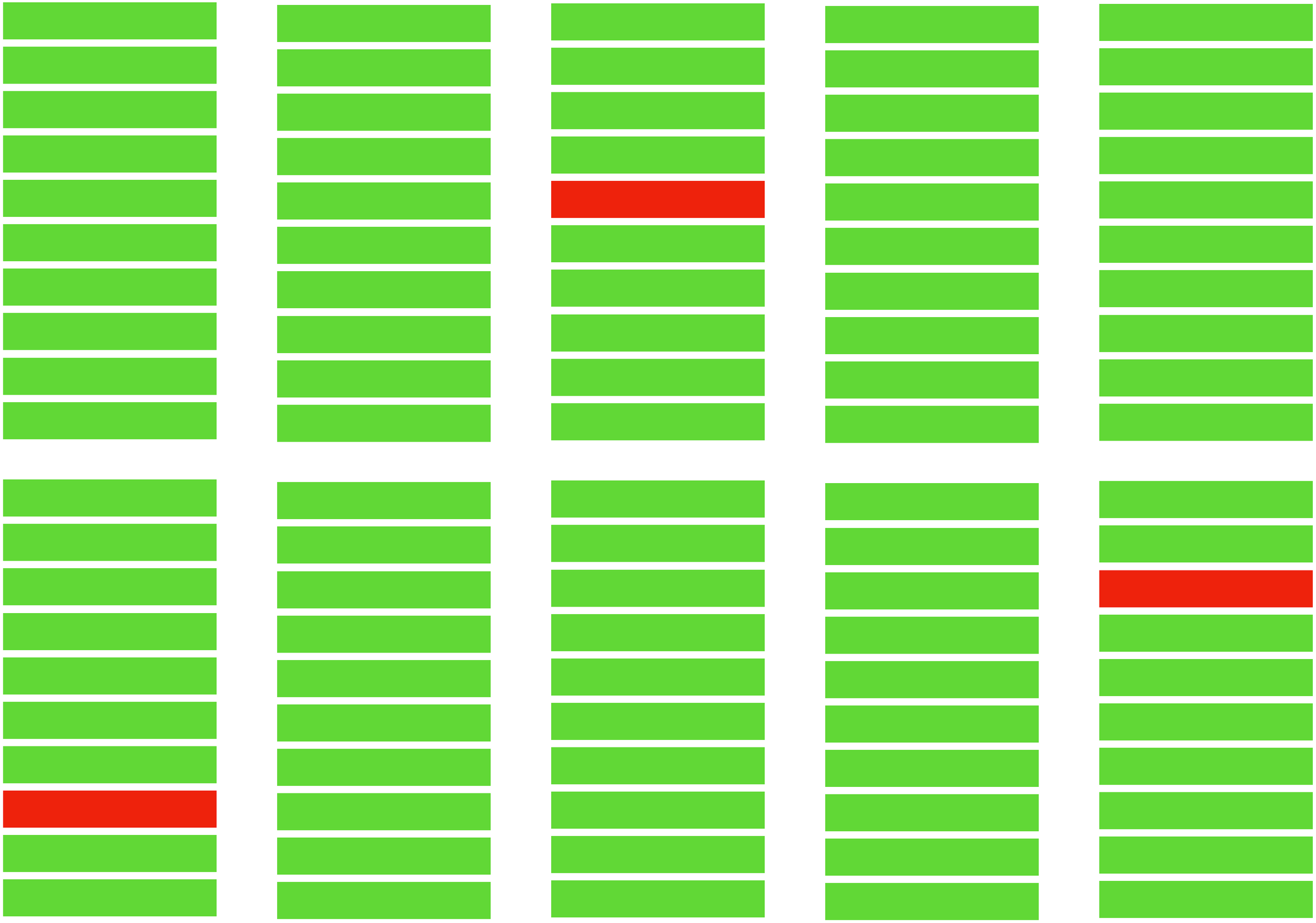
Runs: 1/1 Errors: 0 Failures: 0

@gustavopinto


10 executions



@gustavopinto



@gustavopinto

@gustavopinto

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

@gustavopinto

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

Jonathan Bell[1], Owolabi Legunsen[2], Michael Hilton[3], Lamyaa Eloussi[1], Tiffany Yung[1], and Darko Marinov[1]

[1]George Mason University, Fairfax, VA, USA

[2]University of Illinois at Urbana-Champaign, Urbana, IL, USA

[3]Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

I. INTRODUCTION

CCS CONCEPTS

KEYWORDS

Bell @ ICSE 2018

@gustavopinto

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

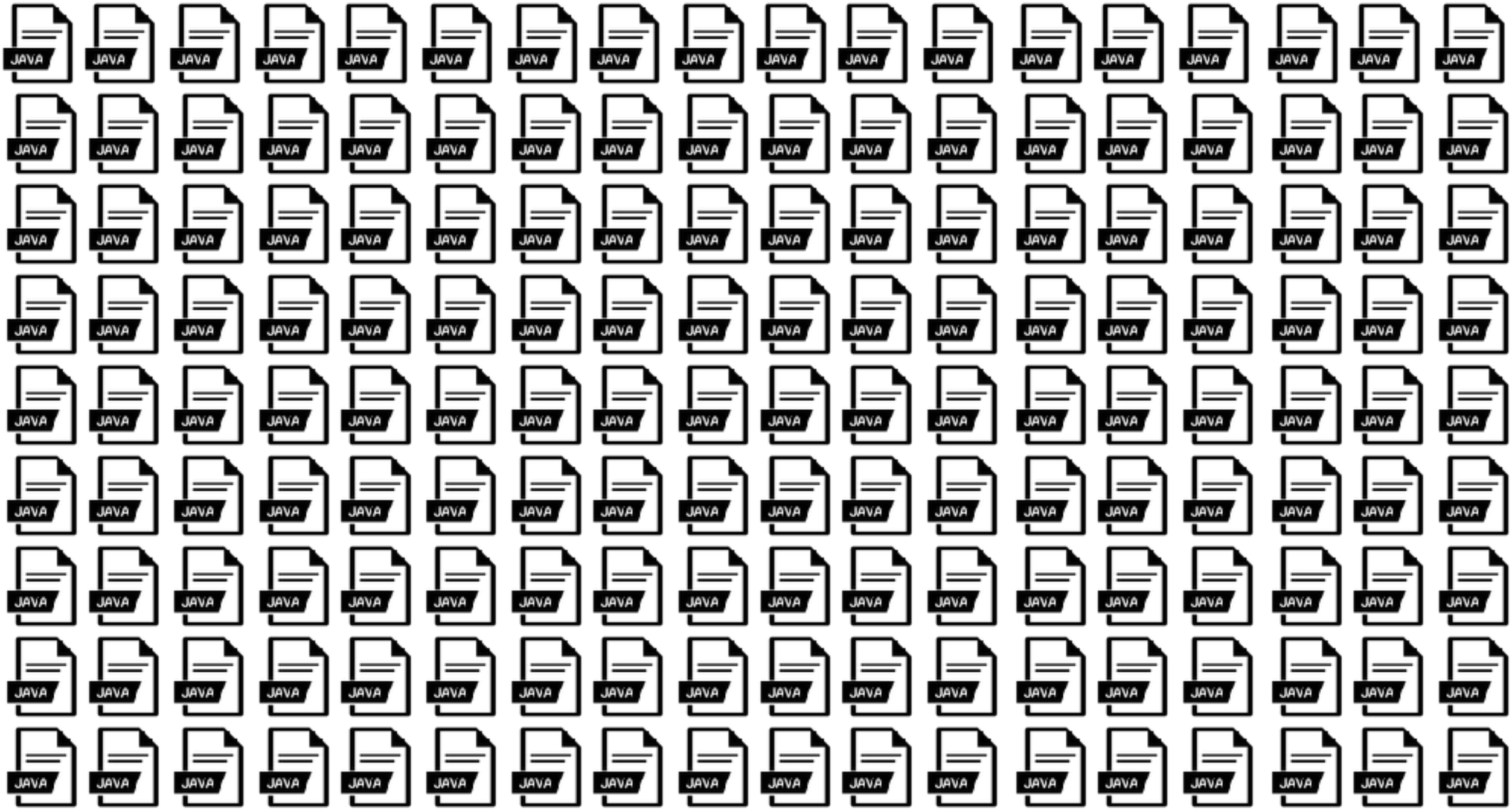
DEFLAKER: Automatically Detecting Flaky Tests

Bell @ ICSE 2018

@gustavopinto

@gustavopinto

@gustavopinto



@gustavopinto

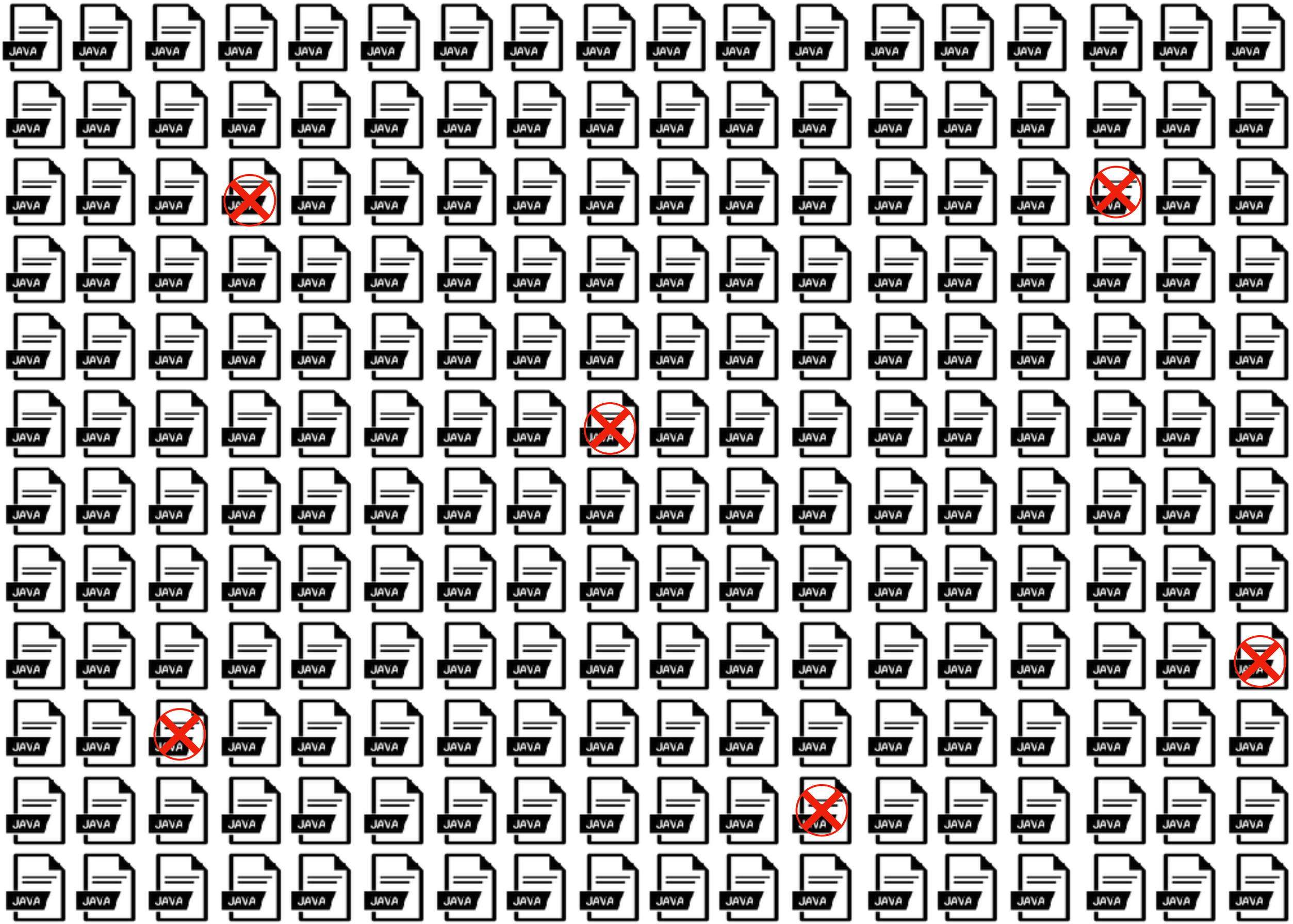
64k test files

@gustavopinto

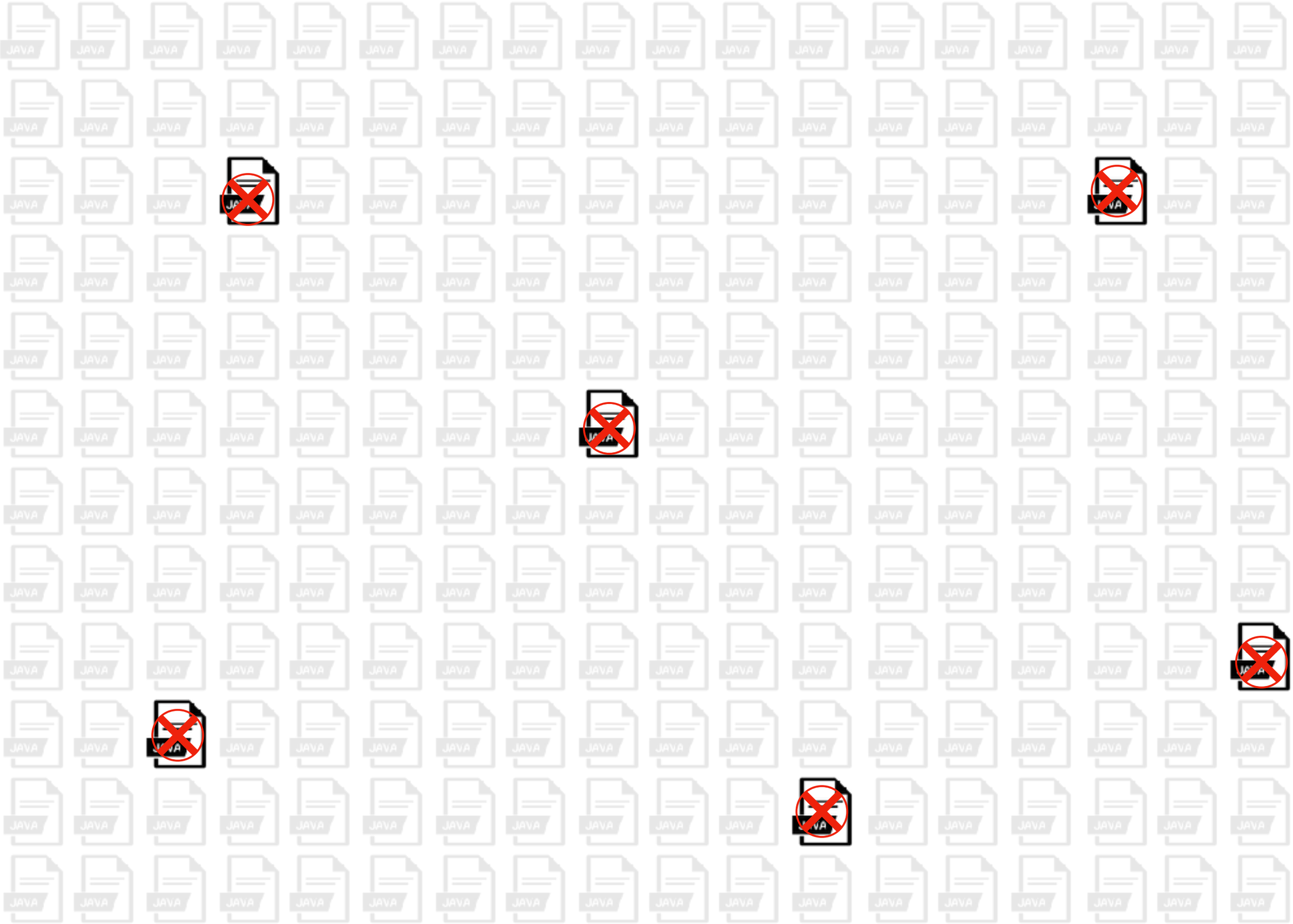
64k * 100 =

6.4 mi executions

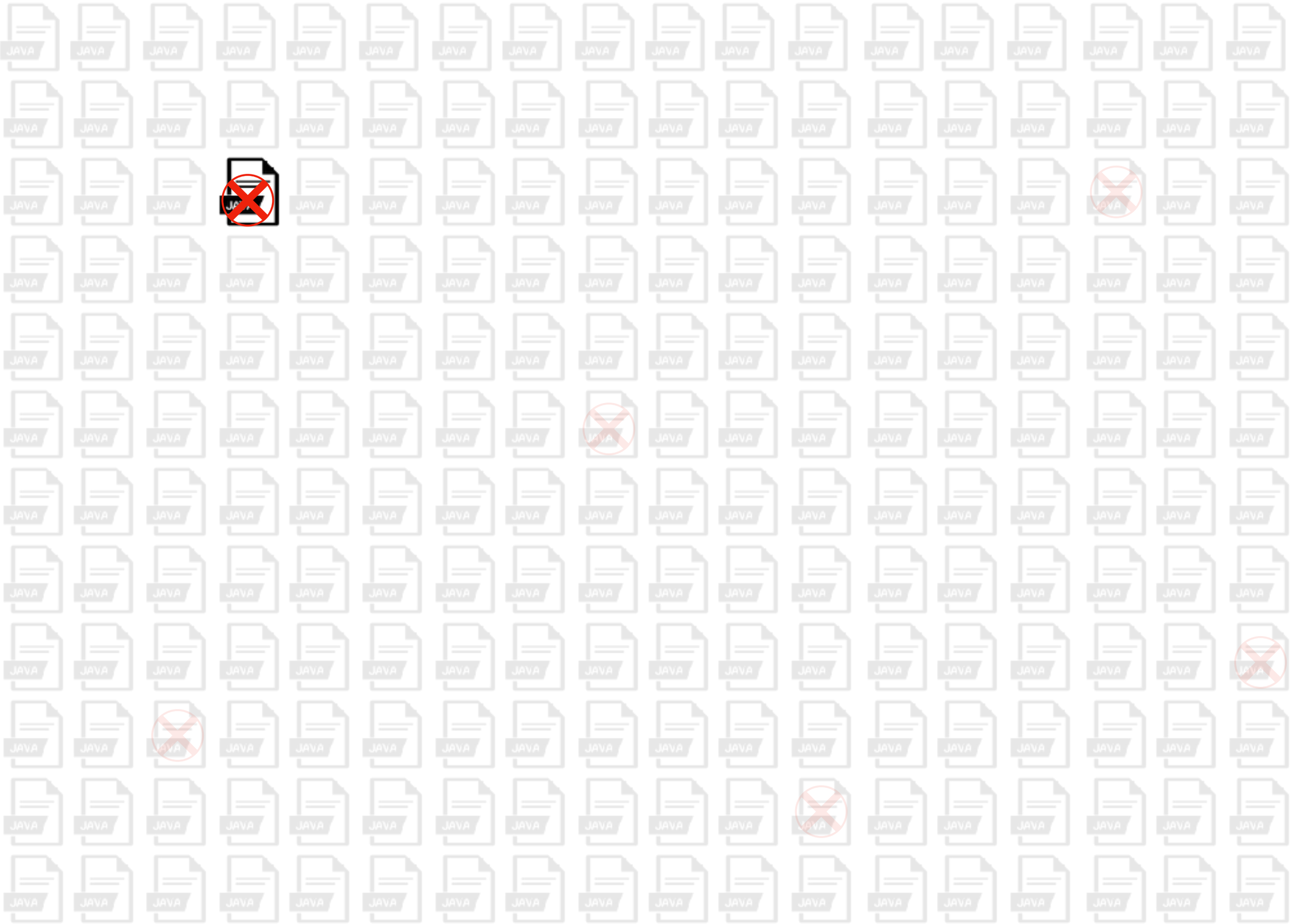
@gustavopinto



@gustavopinto



@gustavopinto



@gustavopinto


```java
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);
    }
}
```

@gustavopinto

```java
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);
    }
}
```

```java
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);
    }
}
```

@gustavopinto

```java
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);
    }
}
```

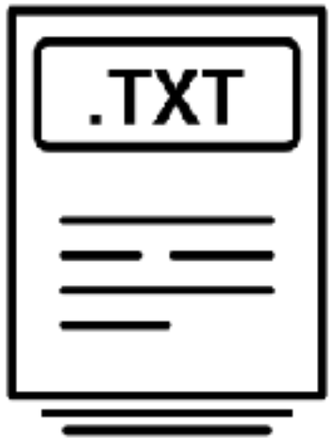
@gustavopinto

```java
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);
    }
}
```

@gustavopinto

```
public class test identify encoder {
    @Test
    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("stuff"));

        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", s);
    }
}
```

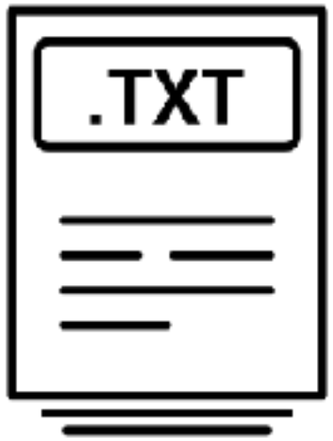
@gustavopinto




```
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
```

@gustavopinto

Known Flaky test

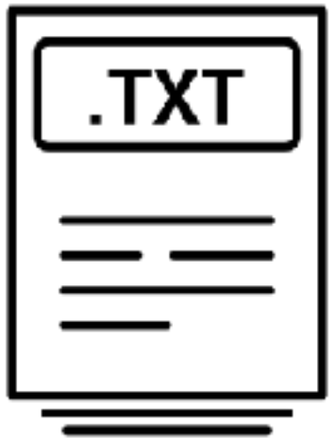


Tokenized Flaky test

Machine Learning algorithms

@gustavopinto

Known Flaky test



Tokenized Flaky test

Machine Learning algorithms

Weka

Nearest Neighbour

Support Vector Machine

Decision Tree

Naive Bayes

Random Forest

Implementation available at:
https://github.com/damorimRG/msr4flakiness/

@gustavopinto

RQ1:

How prevalent are flaky tests?

@gustavopinto

RQ2:
How accurately can we predict test flakiness?

@gustavopinto

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

@gustavopinto

RQ4:

Which test code identifiers are strongly associated with test flakiness?

@gustavopinto

RQ1: How prevalent are 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

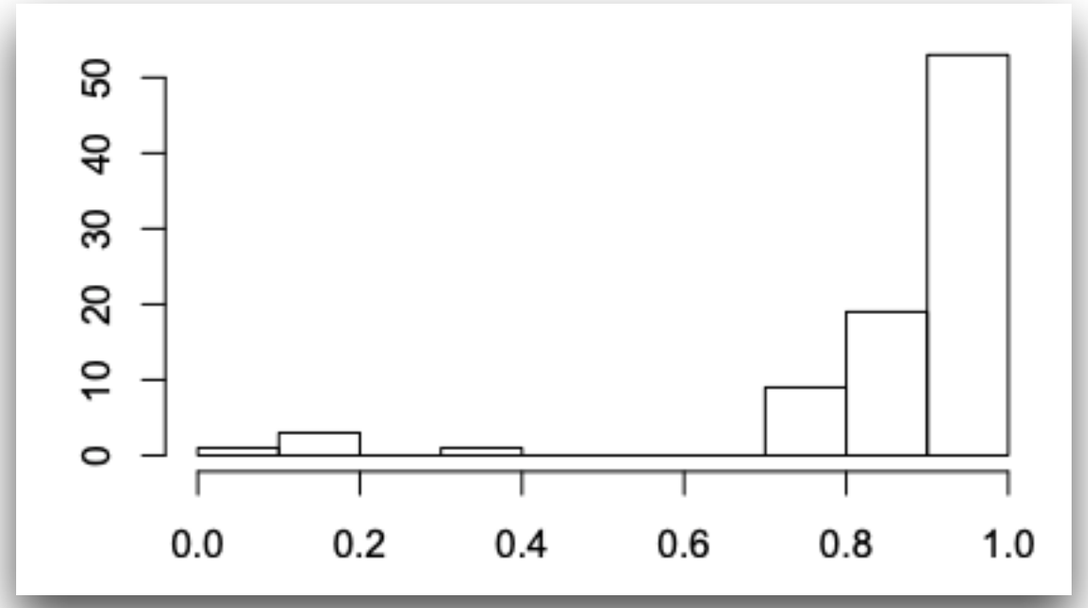
25% of the projects
have at least one flaky test

We found
86 flaky tests

@gustavopinto

RQ1: How prevalent are 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%

@gustavopinto

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

@gustavopinto

RQ3: What's the value of different features

Random Forest

Features	Precision	Recall	F1	MCC	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	MCC	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	MCC	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

No performance impact

@gustavopinto

RQ4: What's the vocabulary?

Features	inf. gain	Flaky # test	Flaky # projects	Non-Flaky # test	Non-Flaky # projects
job	0.20	524	2	4	1
table	0.14	406	4	8	2
id	0.14	552	9	52	4
action	0.13	387	3	8	2
oozie	0.13	274	1	0	0
services	0.13	371	2	7	1
coord	0.11	307	1	0	0
getid	0.11	287	4	1	1
coordinator	0.10	258	1	0	0
xml	0.10	147	2	6	2
workflow	0.09	207	1	0	0
getstatus	0.08	246	2	2	2
record	0.08	296	2	18	1
jpa	0.07	207	2	0	0
jpaservice	0.07	200	1	0	0
service	0.07	367	4	67	3

Common words:

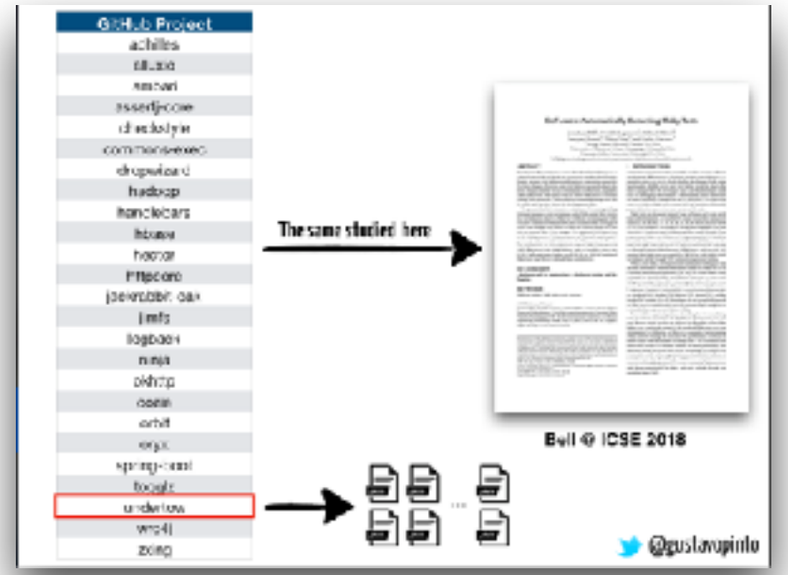
job, table, and action

Many of them associated with

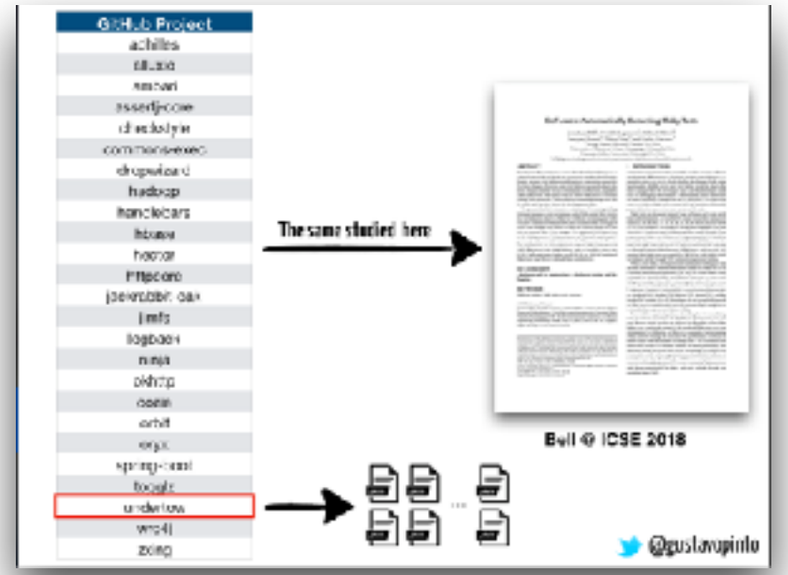
remote tasks

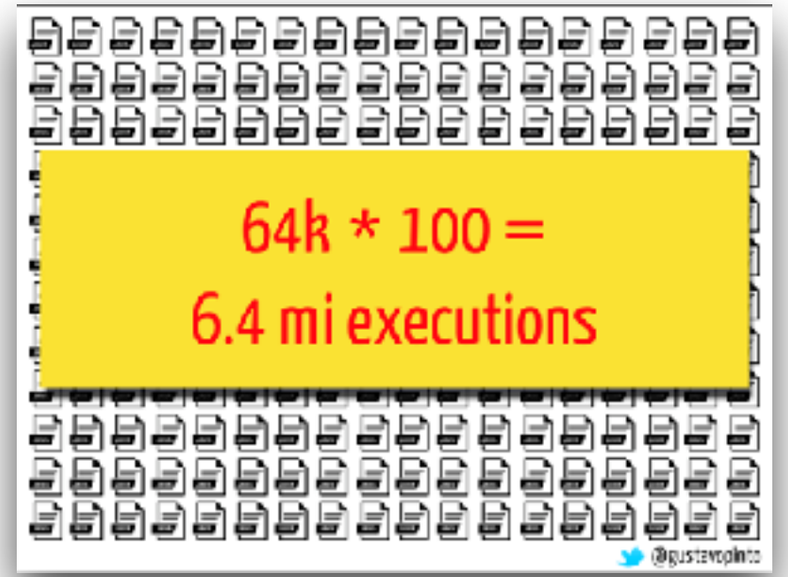
and/or

queue events

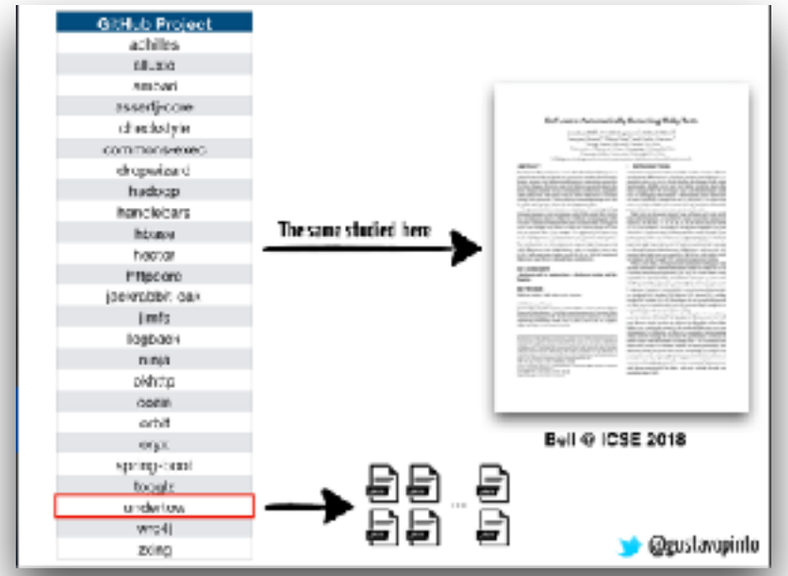


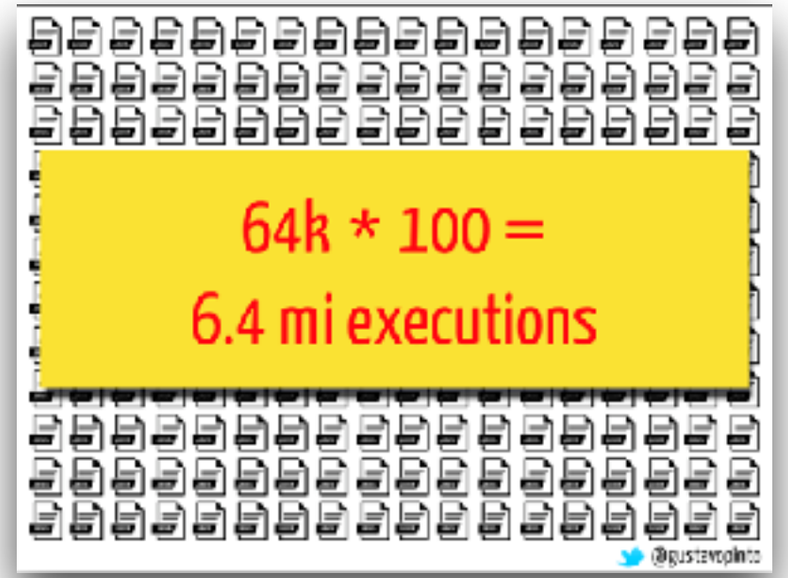
@gustavopinto

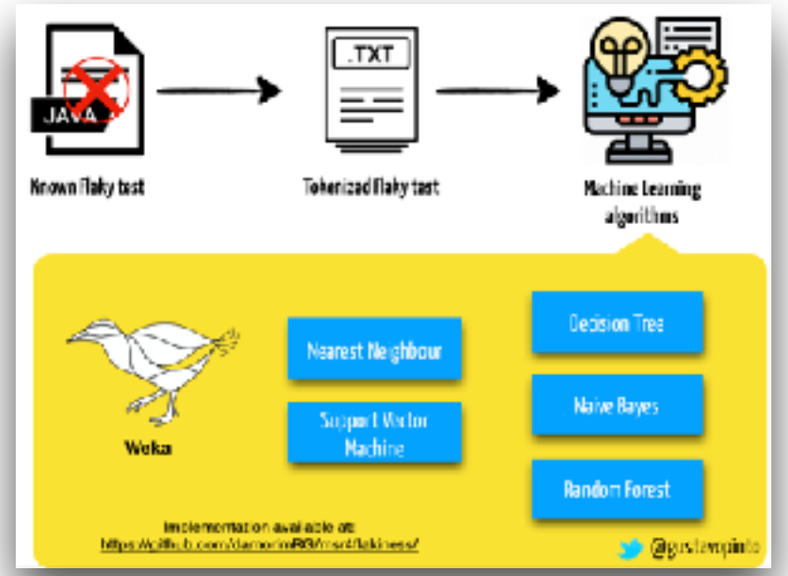




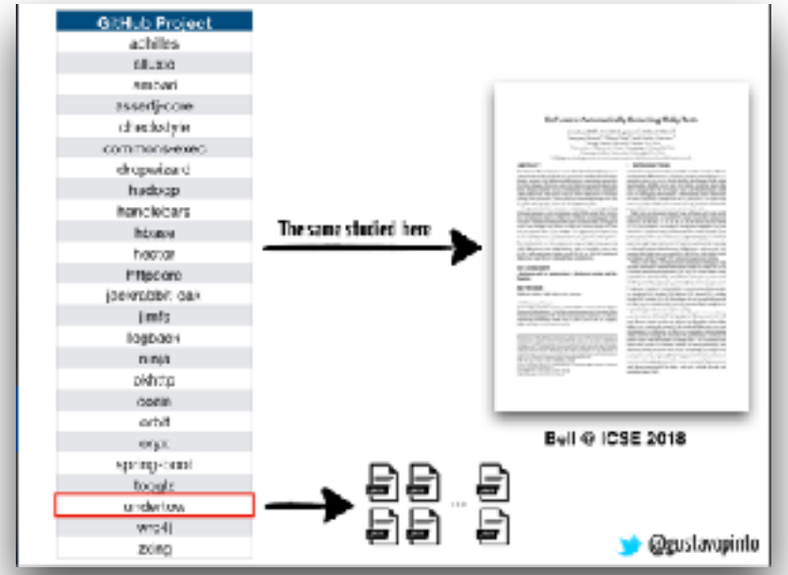
@gustavopinto

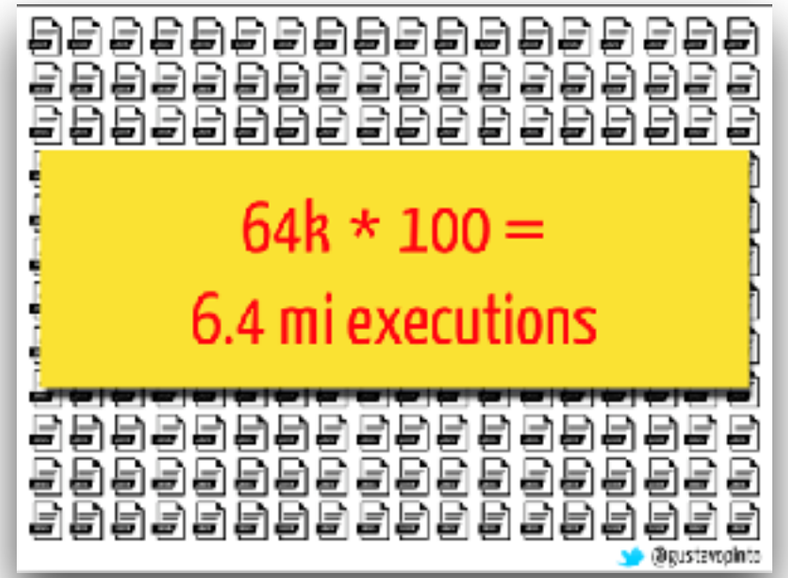


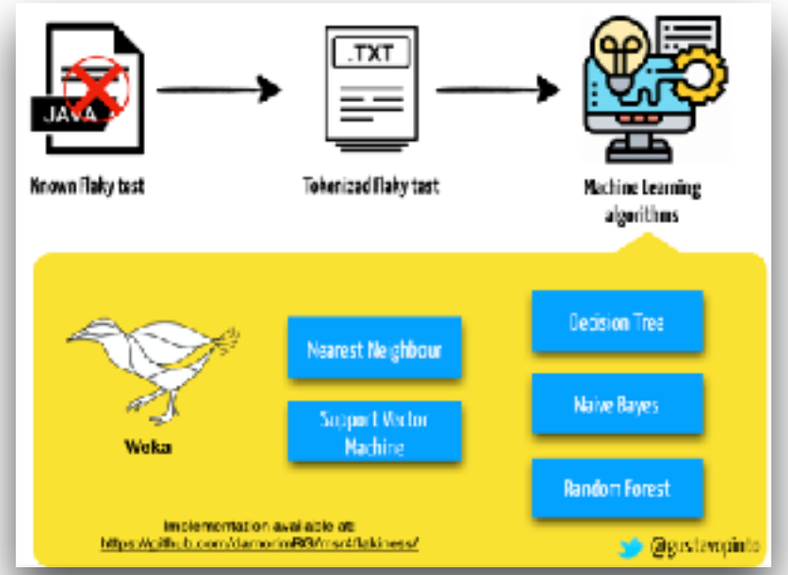


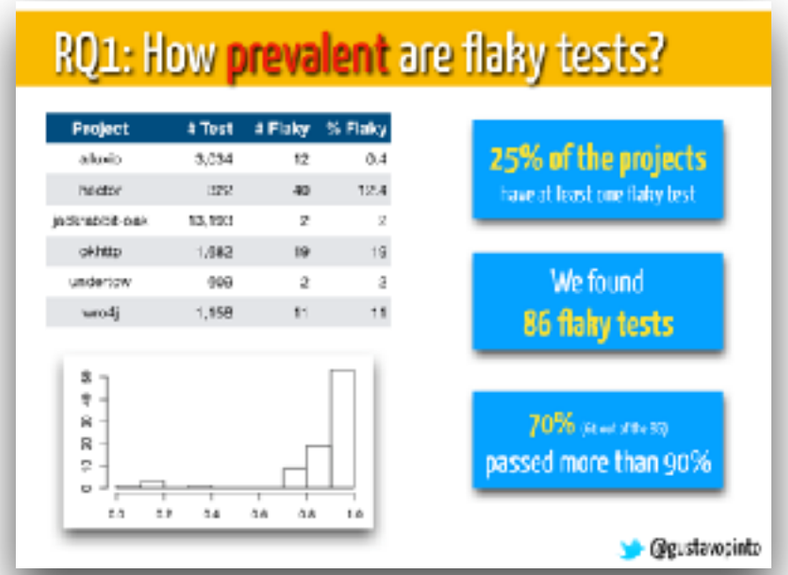


@gustavopinto

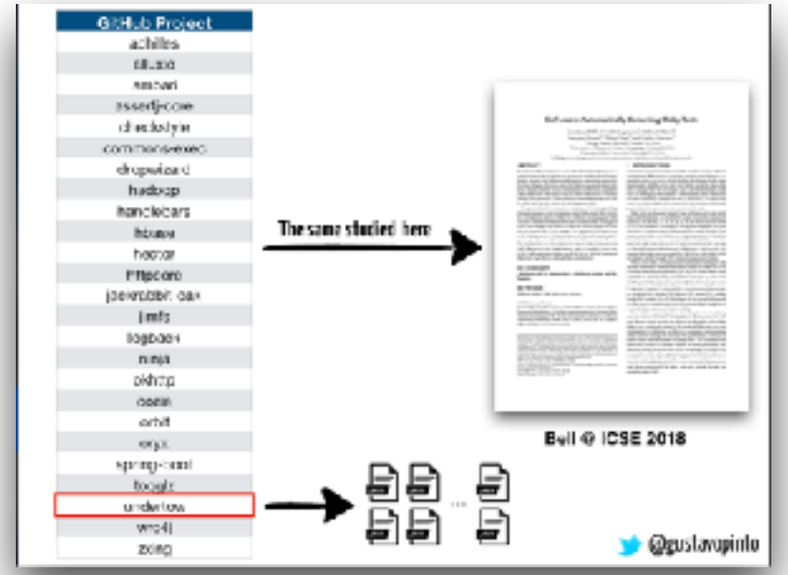


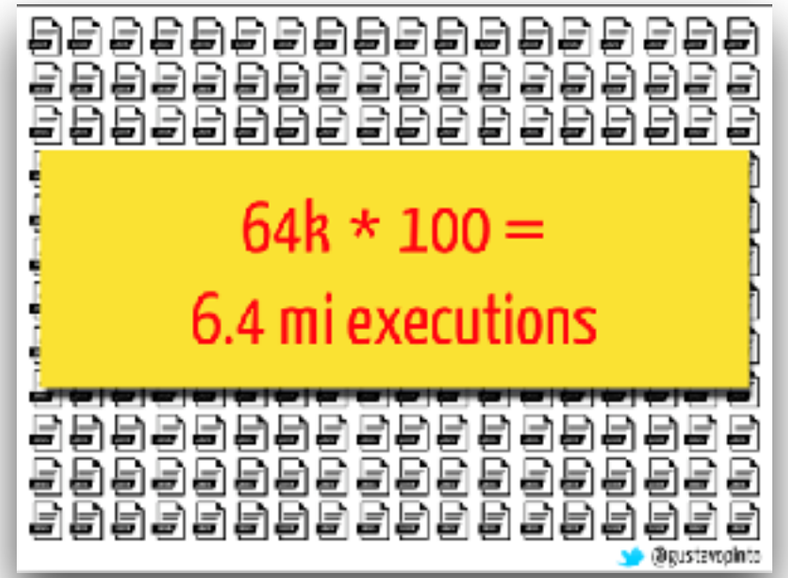


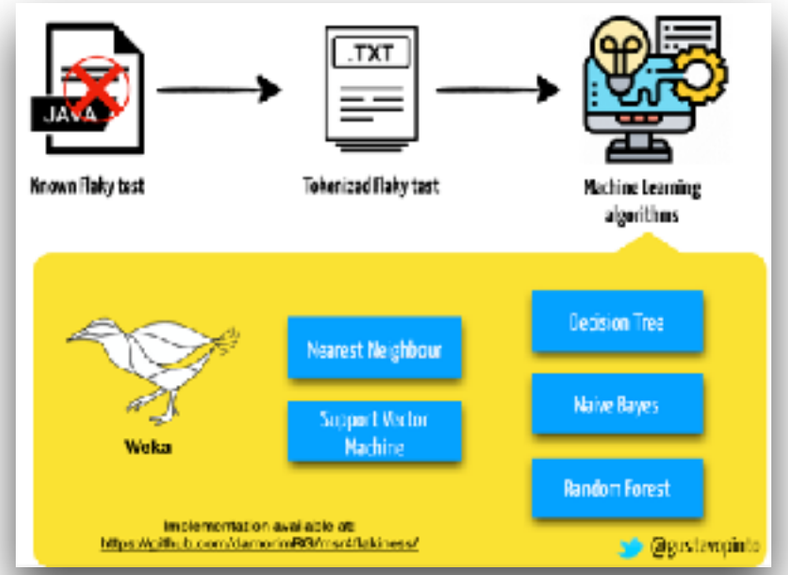




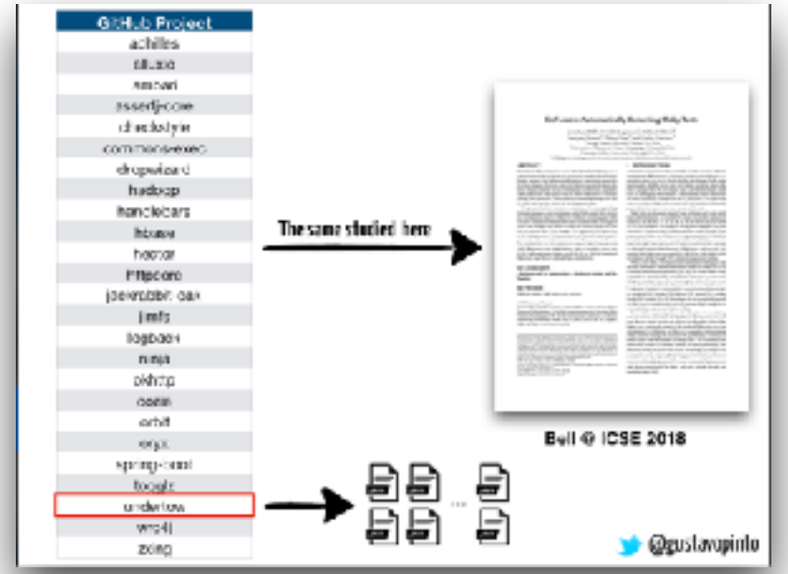
@gustavopinto

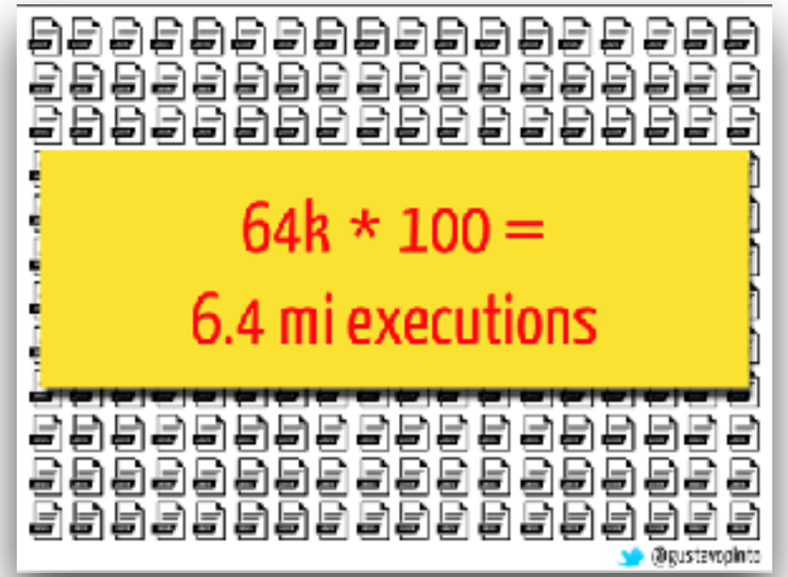


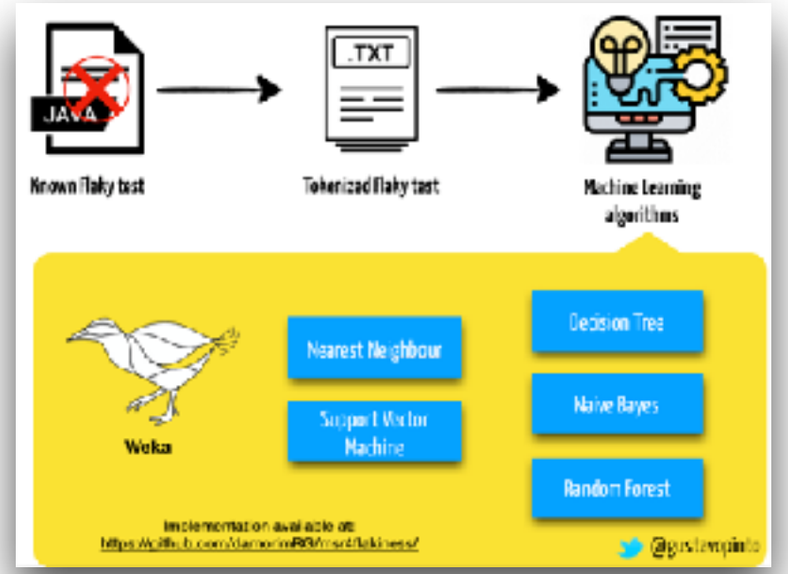




@gustavopinto

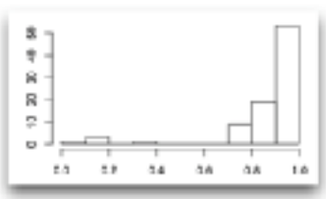






RQ1: How prevalent are flaky tests?

Project	# Test	# Flaky	% Flaky
alluxio	3,234	12	0.4
hadoop	1252	40	12.4
jackrabbit-oak	13,193	2	2
okhttp	1,082	19	19
undertow	999	2	2
wro4j	1,158	11	11



25% of the projects have at least one flaky test

We found **86 flaky tests**

70% (4 out of the 10) passed more than 90%

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	0.91	0.95	0.74	0.93
Support Vector	0.93	0.92	0.93	0.85	0.92
Nearest Neighbour	0.97	0.88	0.92	0.86	0.93

Random Forest achieved best precision

Tuning (e.g. # of trees) **Had no performance impact**

@gustavopinto

RQ3: What's the value of different features

Random Forest

Features	Precision	Recall	F1	MCC	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 Wo. 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

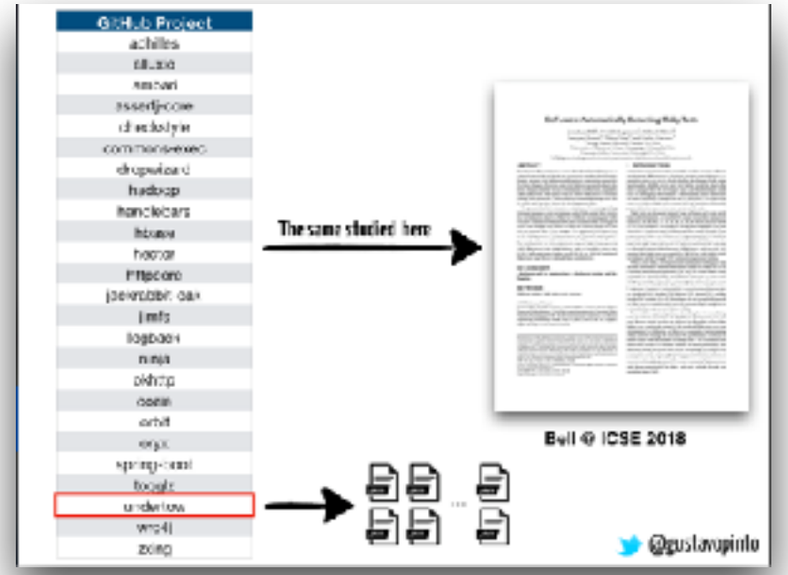
Support Vector Machine

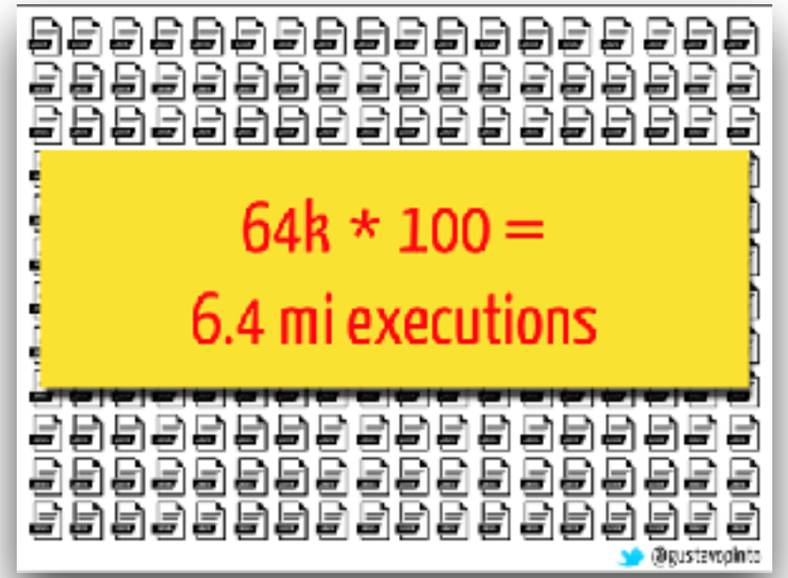
Features	Precision	Recall	F1	MCC	AUC
All features	0.93	0.92	0.93	0.85	0.93
No stemming	0.93	0.92	0.93	0.85	0.93
No Stop Wo. removal	0.93	0.92	0.93	0.85	0.93
No Lowercasing	0.91	0.93	0.92	0.84	0.92
No Java Keywords	0.93	0.92	0.93	0.85	0.93

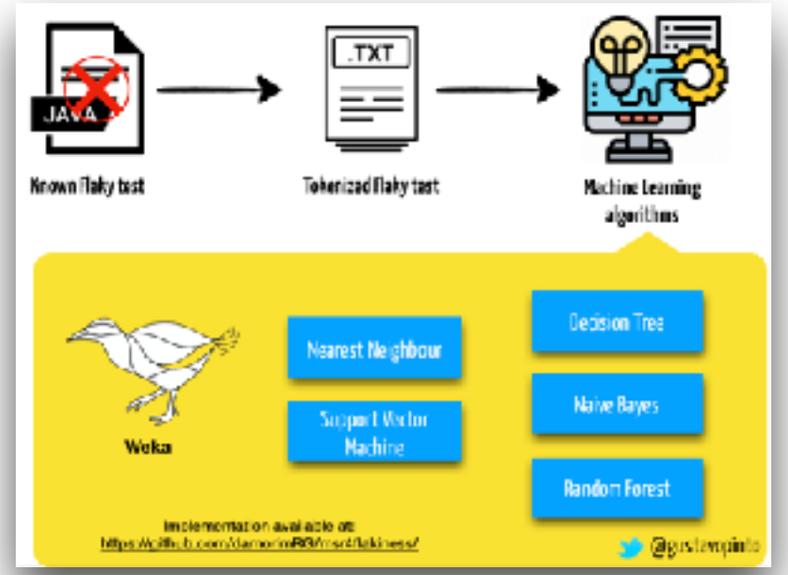
No performance impact

@gustavopinto

@gustavopinto

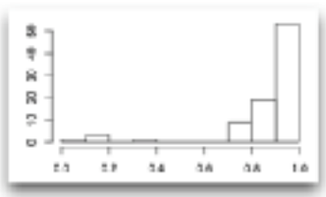






RQ1: How prevalent are flaky tests?

Project	# Test	# Flaky	% Flaky
alluxio	3,234	12	0.4
hadoop	1292	40	12.4
jackrabbit-oak	13,193	2	2
okhttp	1,082	19	19
undertow	999	2	2
wro4j	1,158	11	11



25% of the projects have at least one flaky test

We found **86 flaky tests**

70% (4 out of the 86) passed more than 90%

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	0.91	0.95	0.74	0.93
Support Vector	0.93	0.92	0.93	0.85	0.92
Nearest Neighbour	0.97	0.88	0.92	0.86	0.93

Random Forest achieved best precision

Tuning (e.g. # of trees) **Had no performance impact**

@gustavopinto

RQ3: What's the value of different features

Random Forest

Features	Precision	Recall	F1	MCC	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 Wo. 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 Wo. 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

No performance impact

RQ4: What's the vocabulary?

		Flaky		Non-Flaky	
Features	inf. gain	# test	# projects	# test	# projects
job	0.20	524	2	4	1
table	0.14	406	4	8	2
id	0.14	352	5	32	4
action	0.13	382	5	8	2
route	0.13	274	1	0	0
workers	0.13	371	2	7	1
cache	0.11	399	1	0	0
pgsql	0.11	387	1	1	1
coordinator	0.10	356	1	0	0
ssl	0.10	547	2	6	2
workflow	0.09	399	1	0	0
gatewayhttp	0.08	346	2	2	2
record	0.08	296	2	18	1
jpa	0.07	397	2	0	0
jpaservice	0.07	300	1	0	0
service	0.07	367	4	67	3

Common words: **job, table, and action**

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

@gustavopinto

@gustavopinto