# What is the Vocabulary of Flaky Tests?



<u>Gustavo</u> <u>Pinto</u>



Breno Miranda



Supun Dissanayake



Marcelo d'Amorim



Christoph Treude



Antonia Bertolino















```
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

x Failures: 0



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



```
public class TestIdentifyEncoder {
    @Test
    public void testCodingEmptvSrcBuffer() 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 testCodingEmptvSrcBuffer() 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 Flaky Tests

Cängginou Lub, Farah Hariri, Lamyaa Elbusci, Darko Marinov Becament of Dono, per Science, University of Ilinois at Ilidena-Champag Brance, B. 68(6), 1886 (quod. haririd. elbuscid. marinov)@filinois.edu

Degression besting or a market part of software development. It checks that software changes do not break estation functime by An important commutation of regression to fing is that that substitute are departmental in unmodelfied that is expected to either always pare at always full for the exten-cade under test. Deformabily in greation, some some offer railed Jelly and —have manufactured for an increase —and tests materials of the region testing to they make it. difficult to only on test smaller.

We present the first extension study of falsy turn. We study in decidin total of 301 common that Hole decides totals a span source projects. We arrive the resolution may not make of following a literately approaches that could manifest hally behavior, and describe common strategies that developers use to the finite tests. We believe that our imights and implications can belong to the encrease research to the important topic of involving fairy to to.

Categories and Subject Descriptors: D.25 |Software Engineering's 15sting and 3shinging

Coneral Torona Measurement, Religibility

Koywords: Empirical study, flatry tasts, non-determinism

Regression variety is a exactal part of software development. The elopars was regression less settles to elect that software changes of each brook sweeting functionality. The mail of maning a regionize, but mile is a set of that extreme for the tests in the sairs. The cultimase are important in bank proximate at the action. If it is tests uses therefore it is not better to be tests. If any cast facilit director era reason about the sauce of britary to endersond whether the arrest changes introduced a facility the real; under test (CUT) at whether the test code it. will ment to be the god [0]. The my we imposite with the process is that a test follows indicate that the recent change into cover a pulsary in the (0.01) or the test cover cover.

Presenting at males along an arbon to give at all important into well the presenting statement are a general converted integers and their angles are not made in a distribution of the state of the stat

Photod Mercanous Revol 2004, titing temp, titing Contracts 2014, ACM 515-3-400, 373-546-11, d 15, in http://dx.doi.org/10.1140/250/844.1016020

Halls for adely, test or there exist not reliable for tests that, an intermeditional source hall even for the same code version. Editiving partitioners [11, 12, 24], S. M., are rail such as as that, them entours during determinate with property of them reduced residue. Daily bett conference second problems during regions in section. Plane, and full two crossed by finity hestware behand to reproduce of, excitled an audition of the Council, thisy tests were transposed top to I seem a softward by the recent changes: the developes can spend substantial time cobegging only to find out that the failure is not due to the project changes but due to a finity part. [30]. Third, thickinstrume standars and large site table test trace importing devolutions and to ignore its failures and, thus, could make

Plairy tests one may only publishadic but also relatively common to large residence. Many providingers and as-conduct have pointed our that finity tests can be a highest large problem in general 7,10,10,2,26,59,32,30,37, but 

Google, a half og test is rever. We van agstad the same onder renties on which is rever made follook and file second in an of these films on, it is takked as a filely test [15, 37]. Several open armon testing frameworks the laws according to g., Andread has Official Fort GL Jankins has Office Confee [17] and forms has this cost [31] to be of heavy term that require . For remaining we fall  $\mu_{\rm t}$ 

Another approach would be to remove linky mass from the best outs, or to mercula improvidud; results most of the the distribution of the state of the distribution of the state of the Take a name of the problem on a carrogs and to be being that supported between the carron's proposed on and to be with this begins, they are more throton ratio fasher than advanced. They do not advanced to the carron's author than advanced. these the not remained Balls, below our new probes, all, waster that the not remained Balls, below our new probes, all, waster that all maximum remainess (with test remain) are reclaimed affectiveness of the test and to (with flaby test marketon).

<sup>147</sup> sky" (somet met spelled" lakep") meens " med islie". Personal communication with John Micro

543

### **Luo @ FSE 2014**

### An Empirical Study of Flaky Tests in Android Apps.

Swepen Thorro Chandeni Sreshtha Na Mong Dispursiment of Aurografia between Vegeta To 6. Blacksburg, Vegeta, U.S. (evaporé, riundaré, pr./2/7/6/v. etta

Advance—A filler that is a but that may full or pair for the developers because the num-described the tendence is not promote the num-described that remove is not promote the num-described that remove is not such the number of any developers because the num-described that the number of the part of the number of the number

terior the flaky lests in Andread ages, by classifying the sect tennes and fixing strategies of flations, or aloned in investigate here our proposed characterization for flakings in Andread ages ters our groupes of entrancement for manner in America, and trains from prior findings, and whether Deen are defined expected. findings parterns, after mixing Gillink, w.; Sound 33 Andreid problems containing 'V' commits that were necessal it balances. We shoulded have come of Americal agest findings. Vecorrected these served scenes - Depresalizary, Program Legit, and UV. Fire type of accolution strategies were observed to address the strongs to its Lakiness by changing reference implementation in various ways, determine, there are that 19% aromatic treat steeping diaposed or removed the filthy force. Our observations provide containing the frame research on lady tone of featurality and

Jacks Wessenshaderick Fields tests, requiried

Phily term are the term that terminate with mondeterministic outcomes given the same OUT. When a flaky test is encoured multiple times, the testing results of some suns can be "passed" while the other rens' results are "failed". As the otherne becomes sen-determinate, developes cannot simply rely on the outcome to don do whether an app is buggy, notifier can they easily debug the pode because the influes symptoms. remodure, bush tests one cost developers and travers - addresses the following research quantities (ROp)

satisfactur delarging time and effect.

In Audicid devices have become popular, the capit dec. 2017 What we the common sources for Audicid finitions? reformed and wirequest stage of Android againspare. Among the examined commute, we identified from region

Abstract—A flight test is a test that any field or pass for the — developments beautifully not testing for collection quality assur-

unique dichase patterns and called for any operal medicals.

Facilities [1]. However, they did not make my disky test for histy leak as one proved in the civiling functions.

For this paper, we conducted an amplical make to district the first best first to all the least in Academia ages the sixt to experimely studied for these sensess:

- · Mailorm Pragmentation. As the mold evolution of the Android opening system (CS) confiners, a great number of Android OS versions are available in the makes, coming the problem of Android fragmentation [4]. As a readt, uppe are Blady to behave differently across different Android platforms and manifest flaky behavious.
- Diverse Inscruction, Ansheld, apps usually inscruct with (e.g., phone (evices), notworks, different sentem sixts. and users, intuitively, the more purious interacting with Android appe, the more likely that they can introduce non-designately are uncontrollable behaviors into the execution environment, making instability in Ambrid
- Simple Implementation, Compared with desidey and server applications, Andreid apps are usually small, and me exert pay to undersume functionalities. If we can using filterry A on platform B), in the future, we can hertd at the or agreemin program analysis techniques to specially identity such sometos in new Android agps, and provide accomplify advice to appears the finlances.

Therefore, we conflicted an empirical entity on fixing tests have revealed the nationive missence of flutsy train  $(U_i, (\lambda))$  of Audioid app. Specifically, we crawled for Audioid apps. Specifically, we crawled for Audioid apps. Specifically, we crawled for Audioid apps. Specifically, we converted the Specific for Audioid apps. Specifically, we crawled for Audioid apps. this long [1] that across their complex of tests, a combinant rate for sourch for fixing tennestreant commits. By manually of exact 1.5% of all test runs were seen to report a "thany" examining the reviewed examining, we found  $\nabla$  examine in and Almost 163, of their tests had some level of Batisses. 29 Andreid projects. We analyzed each commit to investigate Unlike other buys, they trans are non-deterministic and also the root cause and fring studges of failiness. This study

### Thorve @ ICSME 2018

# What do we know about flaky tests?

### An Empirical Analysis of Flaky Tests

Cängginou Lub, Farah Hariri, Lamyaa Elbusci, Darko Marinov Becament of Dono, per Science, University of Ilinois at Ilidena-Champag Brance, B. 68(6), 1886 (quod. haririd. elbuscid. marinov)@filinois.edu

Degression besting or a market part of software development. It checks that software changes do not break estation functimelity. An experient correspins of regressive testing is that test outcomes are departmental in unmodified test in expected to either always pare at always full for the exten-cade under test. Deformabily in greation, some some offer railed Jelly and —have manufactured for an increase —and tests materials of the region testing to they make it. difficult to only on test smaller.

We present the first extension study of fisky mats. We study in decidin total of 301 common that Hole decides totals a span source projects. We arrive the resolution may not make of following a literately approaches that could martiest baily behavior, and describe common stealogies that developers are to find tests. We believe that our neights and implication some religious effects resources on the important topic of severing) failty tests.

Categories and Subject Descriptors: D.25 |Software Engineering's Driving and Johnson General Torons Measurement, Religibility

Keywords: Empirical study, flatey pace, non-determinion

Regression naming to a crucial part of software development. Discharges may repression test entires to all est that software changes of each brook sweeting functionality. The mail of raining a regionize, but rails is a set of that out-most for the tests in the sale. The outcome are importan-te back part to take a riches. The distinct have level-opsis bypostly at inches quark the test is as limited. If they that falls, directory reason about the same of billiars to endersond whether the arrest changes introduced a facility the real; under test (CUT) at whether the test code it. will ment to be the god [0]. The my we imposite with the process is that a test follows indicate that the recent change into cover a pulsary in the (0.01) or the test cover cover.

Per metter sa male algors as host as given al all se para al site next de presentant summent un se parame entrere l'impressant des regions au mensaite en attendit de de male avenuement solventes per entre aspect tour de l'action de solvent de dell'indicate en les literappe. Coppeliphe les components de des male comme qu'une des action and en la monte. L'indicate qu'en la relle se personne l'ampe de l'action de se les parameters de la composite de la parameter l'amperent de la parameter de la parameter de la comme de la composite de la consideration de la

/ North Mercanics Forci (2004) Bing Bong China. Contratts 2014 (ACP) \$15.3 (ACC) 2025 \$16.11...d (Clin. http://doi.org/10.1145/25/468.1518620

Halls for adely, test or there exist not reliable for tests that, an intermeditional source hall even for the same code version. Collecting practitioners [11, 22, 21, 25], 32], are rail such as an Shan, there exists no minimize the manufacture of the collection of th bedance behand to reproduce of excitoring audition of the Council, they tests were transport to plot as an authorist by the remaindance; the developes can upon distancial time cobegging only to find out that the failure is not due to the secret changes but due to a finite test [20]. Third, thick tests may also due to all high that both help improving devolutions and to ignore its failures and, thus, could make

and hugs.

Flair tests are not only problematic but also relatively. common to hope real-diseas. Many providitioners and to-consider have pointed our that finity tests can be a highest loopered problem in general 7,12,25,25,36,59,32,30,37, but

begand problem is given of 7,17,27,27,36,39,3,3,3,3,5, better the city appelle numbers we could obtain? we then the TMP option is the girk half this but follows a management of the H part if a making and TM and of 180° (4 at 16) that there are manufoly follows better the problem is resolved. But of 180° (4 at 16) that the content approaches to resolved Help there are resolved and the first the second common approach to our a find that the first the second of the second of the first the second of the Cough, a half up test is every 10 or was against the name or of creation on which, is ever testly foliate and fit is secure in an of these from m, it is taked as a fitty test [15, 27]. Several open armon testing frameworks the laws according to a context into the context of the country of t Andread has OfficialTest D.L. Sanking has Office Confee [17]. and forms has Blacost [11] to lacel flairy best flat routes a few routes asso. Blacost flairy best flat routes a few routes ago, Blacost approach would be to many links sates from

the best suits, or to more alle ignore that: results most of the the distribution of the state of the distribution of the state of the But a manage of I provide some coverage and on lettings had reprotent bears. Although the current toursection and to level with filling terraining allocates their largest, they are more "professormati" author than solutions. They is not adthese the not because the state of the state

Terr sky" (somet mer spelled" listen") means " medialis". Terroral communication with John Micro

 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

543

Luo @ FSE 2014



# What do we know about flaky tests?

### An Empirical Study of Flaky Tests in Android Apps.

Swepen Thorro Chandeni Sreshtha Na Mong Department of Language between Vergreen The C. Blacksburg, Virginia, U.S. (evaporé, chardani, am/2/7/9y:seta

Altebrach—A flight test in a test that may full or pass for the same ratio under studing CCTS. Thick test wide to be made to develope a term of the same the number of the same ratio of the sam

propose and storated to the Qui fill the Amirold approximation of temperature and make the any openin moderate. For this paper, we conducted an amplitudicately by district the filler both is accordant as amplitudicately to district the filler both is Amirold approximation to the conducted and approximation of the conducted and approximation of the foreign the series are made in the interligible for every proposed characterisation for fillalities in Amirold approximation are made in the amplitudicate and the Amirold approximation. As the middle excellent of the Amirold approximation is a proximation of the Amirold approximation and the amirold approximation of the fill the conduction of t ters our groupes commerciance for manner in charge of pro-terior from price to deg., and whether these are delined expected finitions parterns. After mixing Gilflish, we found 28 Archeld projects containing 7% commits that where territorial is finitional. We detailed the new course of Ancheld again Scheme. We consisted these commitments of Approximate, Program Logic, and C. Fire types of accolution strategies were observed to address the strengt to the Habiton by changing retireary implementation to content ways, detector, bard are that 19% contents that strengt diapped or removed the fidely force. Our observations provide containingly for frame consents on help that of featurable appro-

Jacks Westernhaltenid, fishe tests, requirited.

Phily taxe are the taxe that terminate with nondeterminists outcomes given the same CUT. When a flaky test is encoured multiple times, the testing results of some suns can be "cassed" while the other rens' results are "fulfol". As the ottome becomes sen-deterministic, developers cannot simply rely on the outcome to don do whether an app to buggy, notifier can they easily debug the code because the influe symptoms. hard in separative, bush tests our cost developers and testors. Addresses the following research quantities (ROp)

satisfactor delarging true and effect.

As Audicid devices have become popular, the rapid dec. 2007: What we the common sources for Audicid Palifornia?

- of Android OS versions are available in the maket, cousing the problem of Androld fragmentation [4]. As a really upper an likely to behave differently across different Andreid platforms and manifest flaky behaviors.
- Diverse Interaction, Andreid approximally interact with various softwares (e.g., third-pury libraries), hardware (e.g., phone devices), notworks, different senson show, and uses, intuitively, the more parties interacting with Android apps, the more likely that they can introduce aso-desiration or approprofible behaviors into the escular environment, causing instability in Android
- · Simple Implementation, Compared with decidey and server applications, Acadreid upps are usually small, and implement only to understand functional ties. If we can consistency posterio according to their tests (r,g) when using Biology A on playform B), in the fature, we can he ld sittle or dynamic progress analysis techniques to specially identify each scenarios in new Android apps, and provide promettle advice to actions the fintances.

Therefore, we conducted an empirical study on fixing tests have revealed the extractive extracted of links trots (II., 3). of Audreid app. Specifically, we convict for Audreid app. John Micro, a senior Clough developes, once membered in ... on Guillab and used the knyweeds "linky" and "intermittent this long [1] that across their complex of leads, a combinant rate of short 1.5% of all leaf rate were seen to report a "thing" examining the numbered exempts, we found  $\nabla$  exempts in result. Almost 16.5 of their leads had some lead of their control of their leafs and the complex of their leafs and th Unlike other bugs, fixey tree are non-deterministic and also the root cause and thing strategy of finitions. This study

refigured and wirequest stage of Android agus require. Among the examined commits, we identified flow region

Xxx

### Thorve @ ICSME 2018



# 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);
```



```
x Errors: 0
                                 x Failures: 0
Runs: 1/1
  @Test
Rupsil/d void texception {
      final WritableByteChannelMock channel = new WritableByteChannelMock(64);
      final SessionOutput Ruffer outbuf = new SessionOutput Ruffer Impl (1024 128).
Runs: 1/1hal Iden xtErrors: 0 encod x Failures: 0ntityEncoder (channel, outbuf, metrics);
      encoder.write(CodecTestUtils.wrap("stuff"));
Runs: 17 Pty.flip
                                   Failures: 0
      encoder.write(null);
Runs: 1/4tbuf.flux (Errors:11;
                                 x Failures: 0
      final String s = channel.dump(Consts.ASCII);
```

x Failures: 0

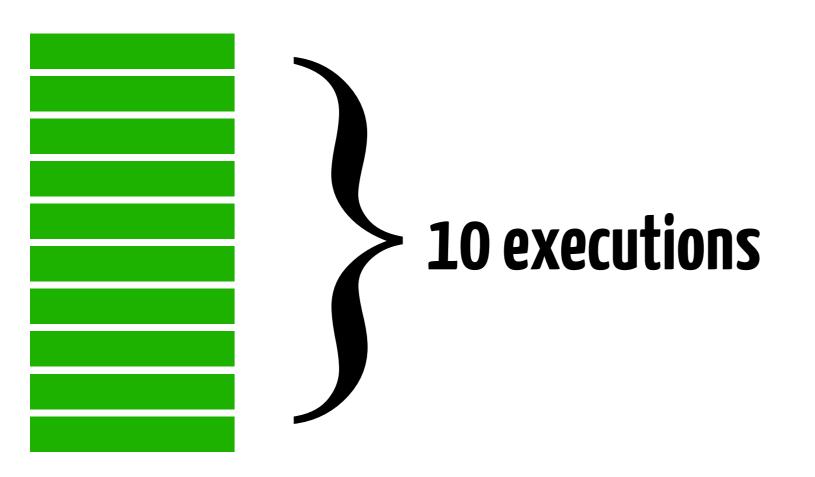
```
Runs: 1/1 x Errors: 0 x Failures: 0
```

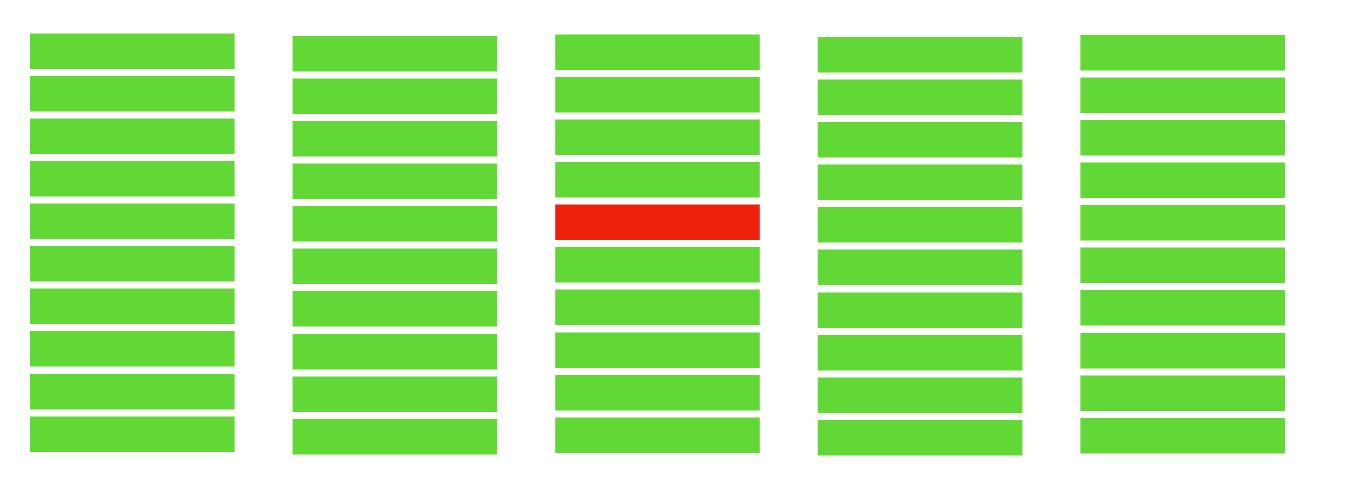
Errors: 0

Assert.assertEquals("stuff"

Runs: 1/1















### **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. Mently, any new test rationer would be test our regressions are need by the latest changes. However, come test full loss 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 kein 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 to the test fields well without remaining one with very low reating overfield. (In these mentions the coverage of lover once changes and maximum linky anymore yielling 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) (triProgram had a legic or well (#150 or or 200) of conferred field years than blaver's definite field; test desected.

### 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 page.

Formulas as miles digital enhant age as affel on pass of this mark has record out a service as a supposed in the order to be present in the order and the service as a service as a service of the order of the order to be a fine order to be a fill dischool as 10 to 10 to

### 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 as who then the energy backs 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 recovered by the constitution on changing those believes the following place to the constitution marries, some of the continuous changing those follows to find a surface and the continuous changes to the level place to an experience work, we define a takey but as a tast that can use obstructioning the continuous continuous changes are some continuous changes. 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 practice. toman (1.2), S. M. P. (6.44.4), S. M. M.-S. M. S. Al. (5.8), S. Ol. For example, according to Foreign of Magney milet; the Macrosoft Velocitor and Dynamic product to sear examines from properties of fickness of discussion by agreemently M. Stechnig, 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],  $[a_{ij}]$ ,  $[a_{ij}]$ ,  $[a_{ij}]$ , and the Cough TVP seaton (43,4%) Developed do not proachedly seath. Sufficiently test maintenance articly insteads uply using Return

to 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 orthograph to resolve d. He by tests are nondeterministic 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 some. Strenging view failed text in extensity entity when experiention so; 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<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. 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 kein 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 to the test fields well without remaining one with very low reating overfield. (In these mentions the coverage of lover once changes and maximum linky anymore yielling 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) (triProgram had a legic or well (#150 or or 200) of conferred field years than blaver's definite field; test desected.

### OCS CONCEPTS

Software and its engineering 

Suftware testing and de-

### KEYWORDS

Software testing, flaky tests, cade coverage

### ACABA komunikansak

reconstitution for the continuous photos of the continuous for the continuous section of the continuous section for the continuous forther continuous for the continuous forther continuous forther continuous forther continuous forther continuous forther continuous forther continuous for the continuous forther continu

Formulas as miles digital enhant age as affel on pass of this mark has record out a service as a supposed in the order to be present in the order and the service as a service as a service of the order of the order to be a fine order to be a fill dischool as 10 to 10 to

### 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 have the changes backs some functionally. Bloody, every new test failure would be due to the Treat that yet that the Astroper as the artific here open could do not one original that the follows to the artific here open could do not not original three follows. To the harvesty, since it thereto and the notific here or charged into the top fish, then Artific products would, we define a tidaly but as a but that our non-visit remaindently. tions or find when run on the capes service of the code

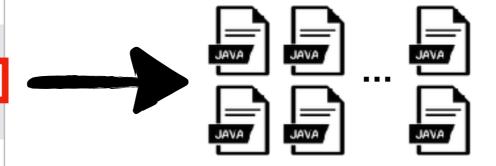
Thisy term are frequent in most large software, and create professor in development, as described to many assemblers and practice. Provides dispersion in that half of their built fall controlled the built fall controlled the built fall controlled the fall controlled the fall controlled the fall described 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. proved the vertic delibities; thelp, but it all recent full, the states in unknown. Thus is in apported by according to the growth, e.g., Analosid, [24], [enhant [22], it them [27], is ping [23], 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.

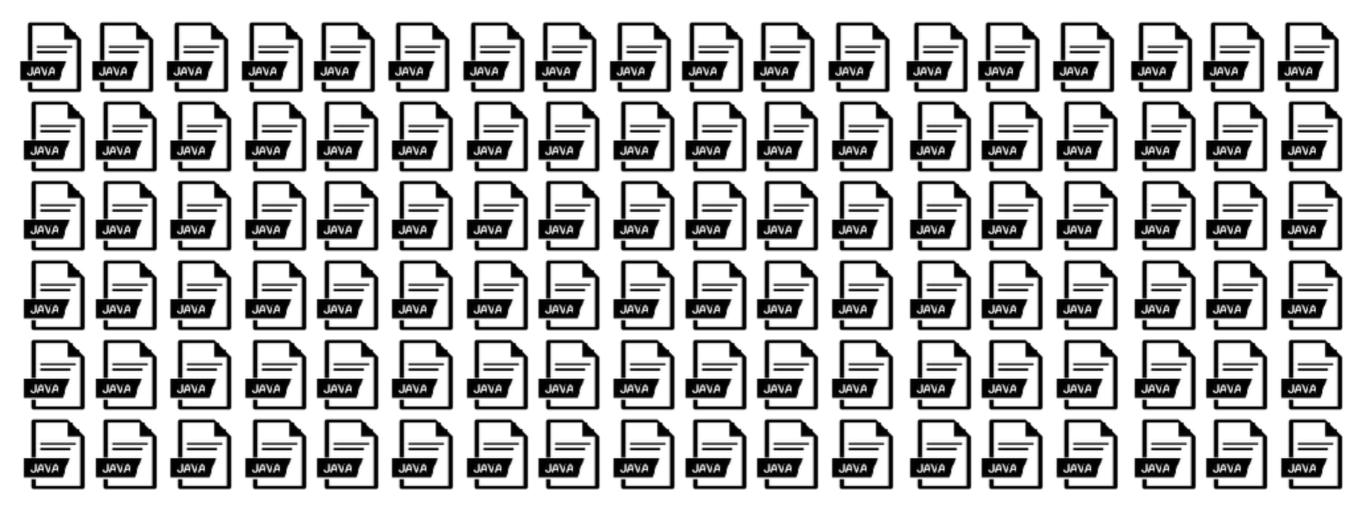
in is order to marintar the blothhood of seturning the test pass. Berma might need to be delayed to allow the cause of the following, a network orthograph to resolve d. He by tests are nondeterministic by defination, so there is no granutize that necessing a tinky issistil dramps in origons. The performance 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)

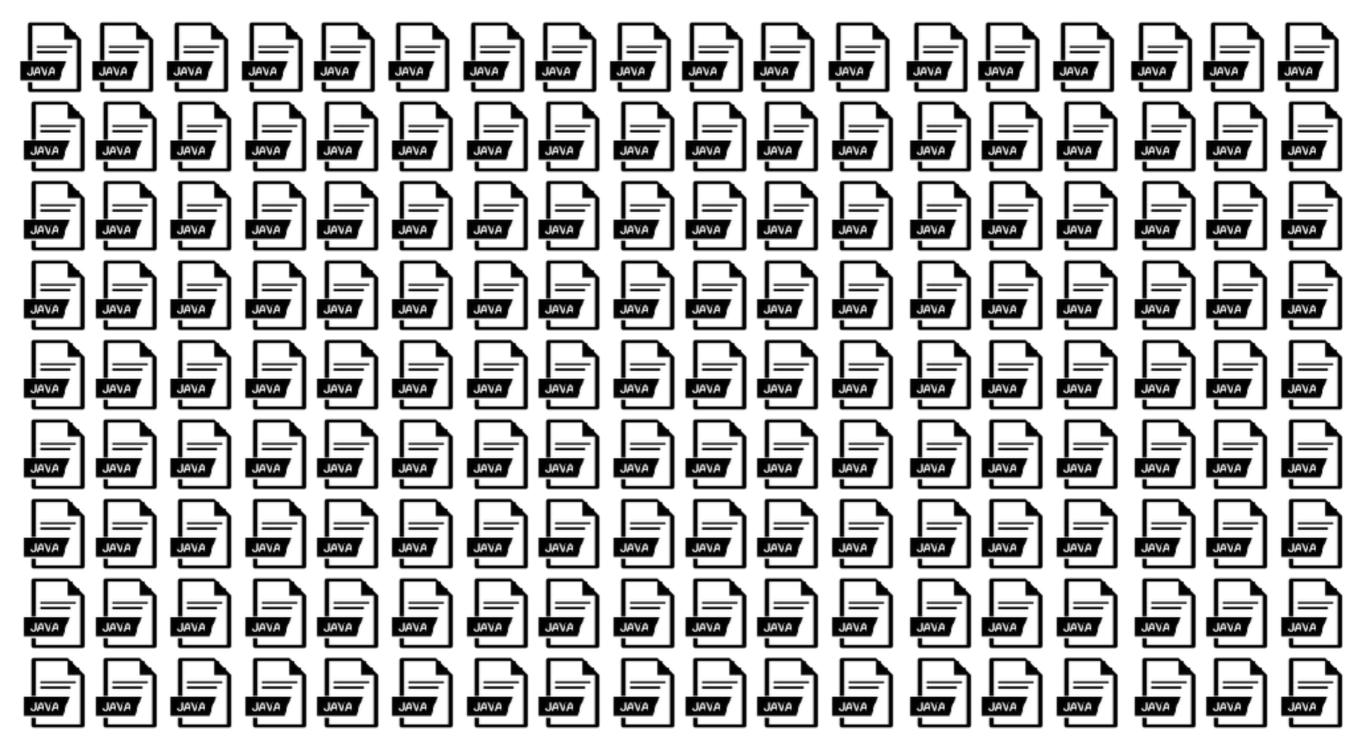
### **Bell @ ICSE 2018**

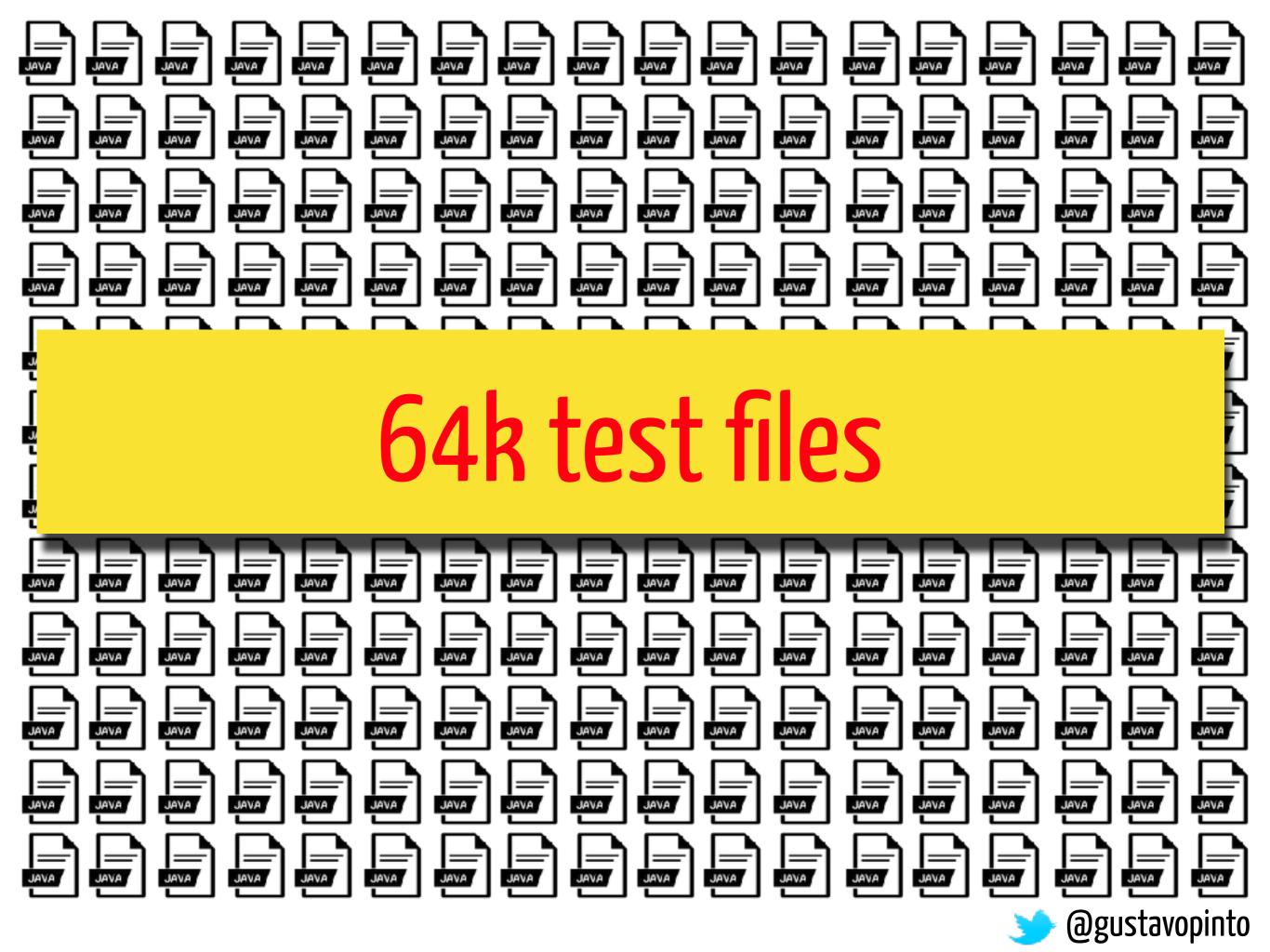


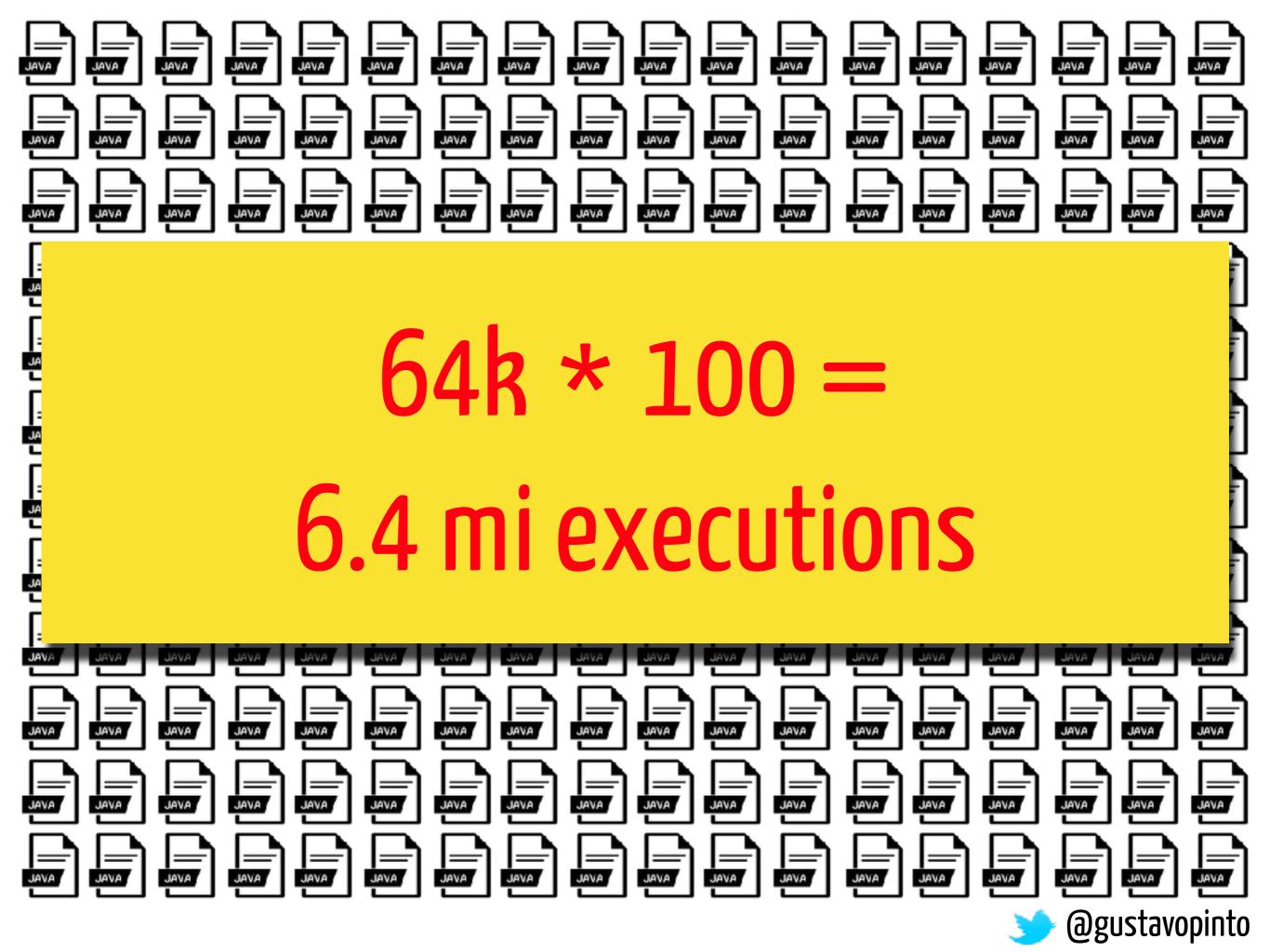


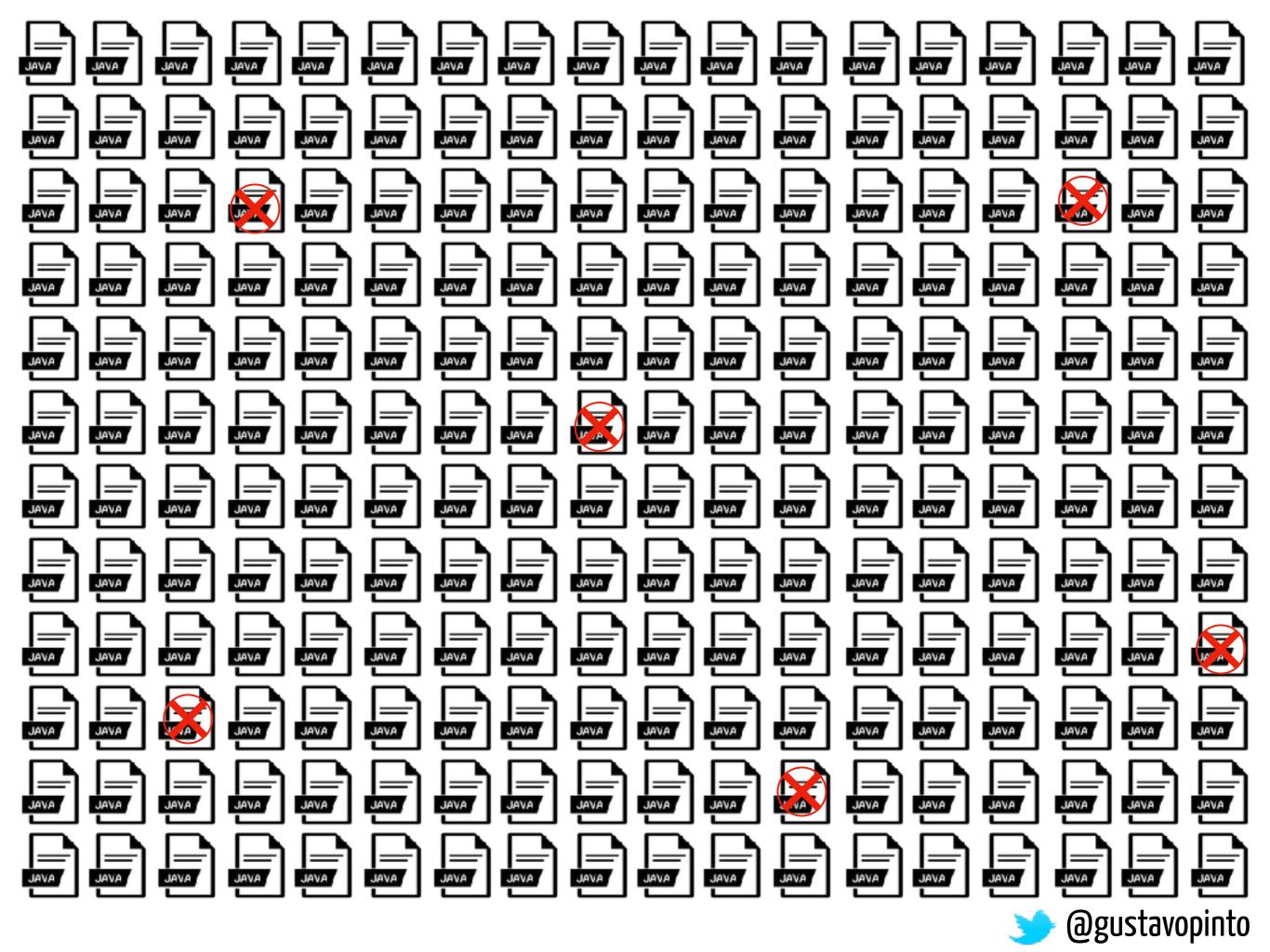


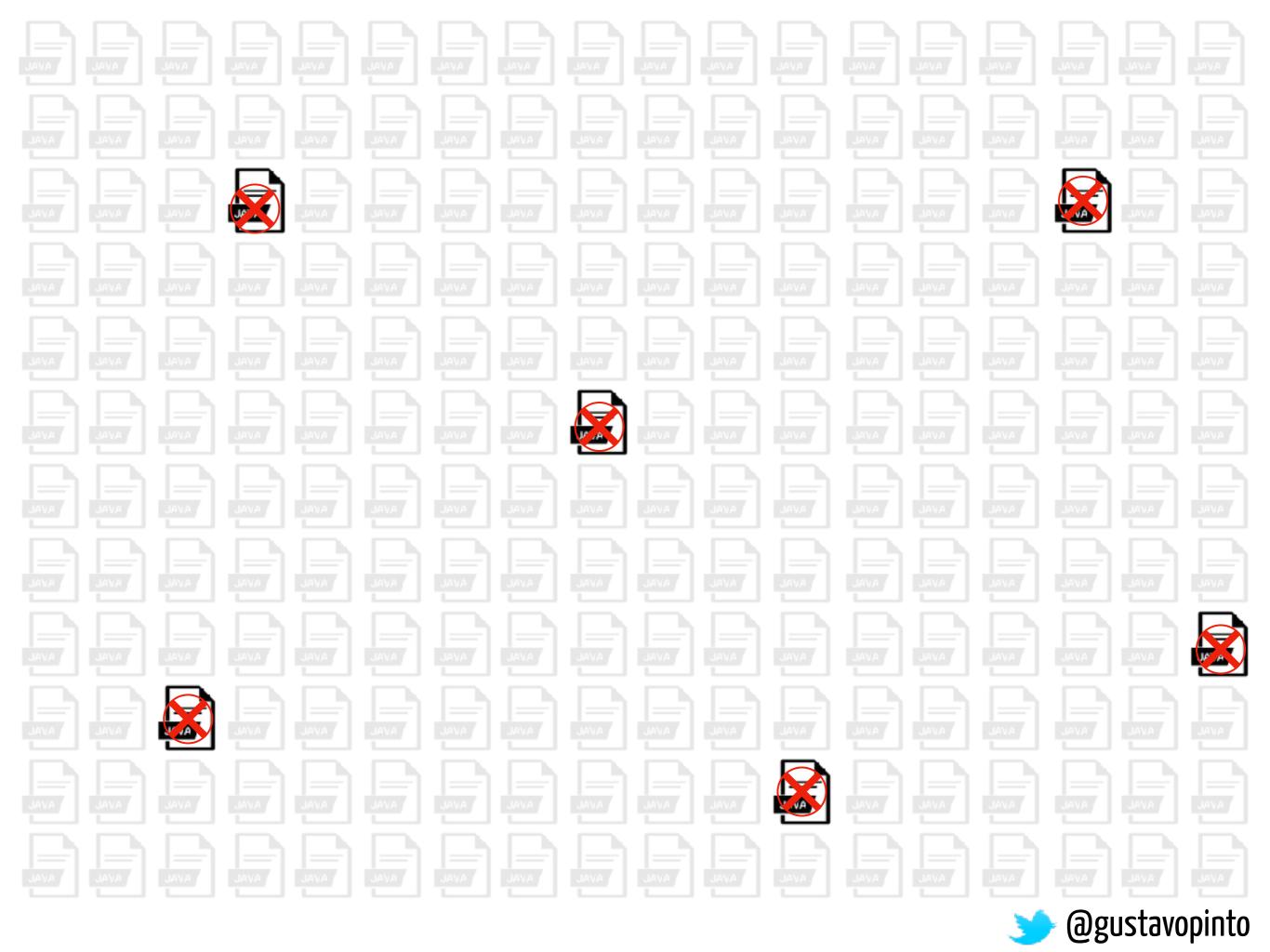


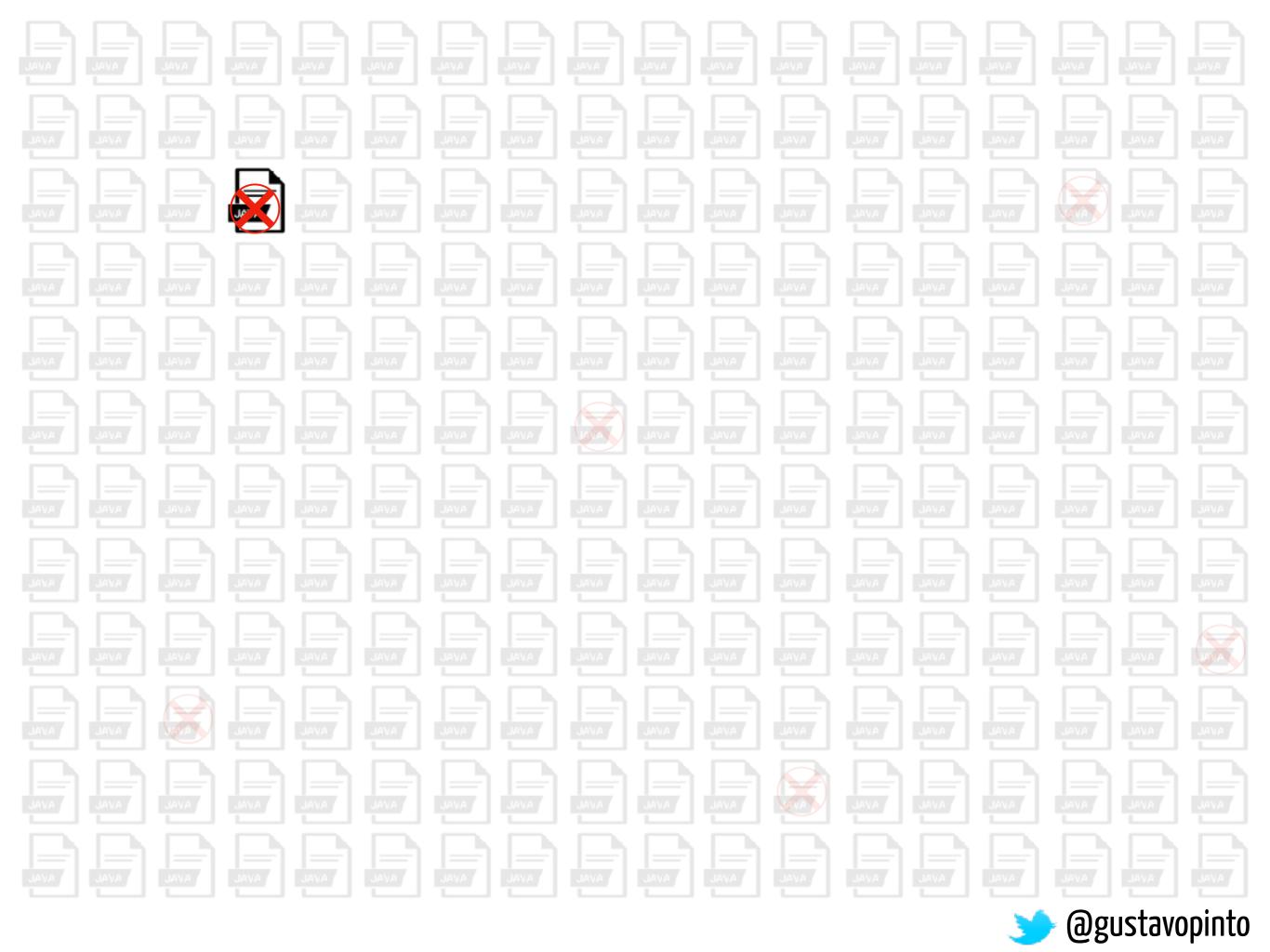














```
public class TestIdentifyEncoder {
    @Test
   public void testCodingEmptySrcBuffer() throws Exception {
        final WritableByteChannelMock channel = new WritableByteChannelMock(64);
        final SessionOutputBuffer outbuf = new SessionOutputBufferImpl(1024, 128);
        final HttpTransportMetricsImpl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



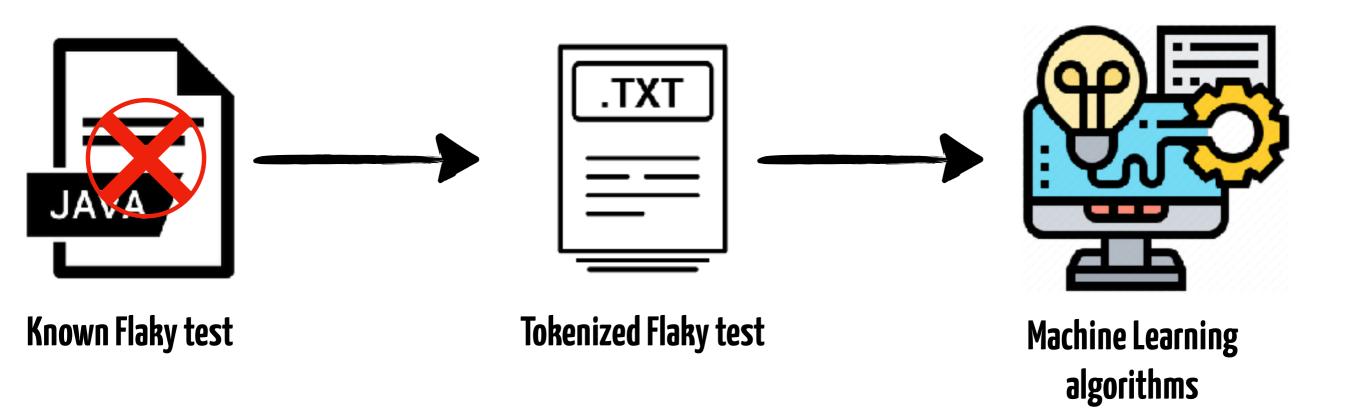
```
public class test identify encoder
   public void test coding empty src buffer() throws exception
        final writable byte channel mock channel = new writable byte channel mock (64);
        final session output buffer outbuf = new session output buffer impl(1024, 128);
        final http transport 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",
```

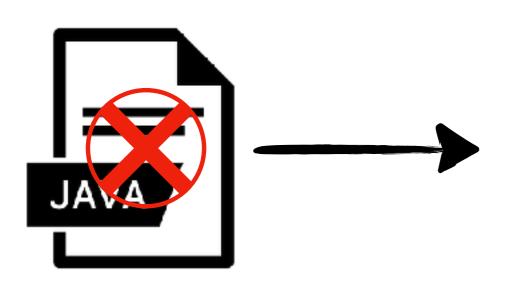




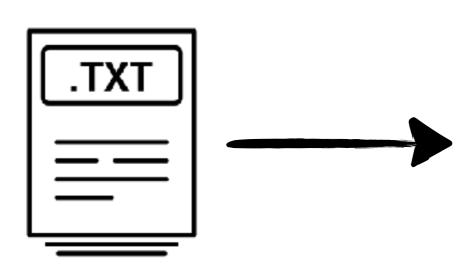
test identify encoder coding empty src buffer byte channel mock writable session output outbuf session http transport metrics impl identity codec utils wrap buffer allocate flip write complete flush dump consts ascii is completed assert equals







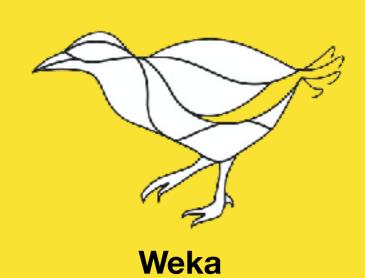




**Tokenized Flaky test** 



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>





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



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



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



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

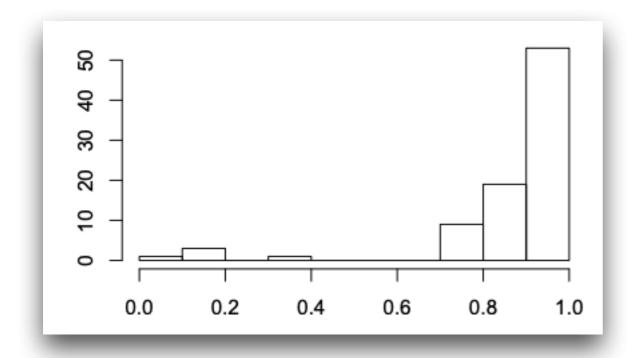
25% of the projects

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: 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'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



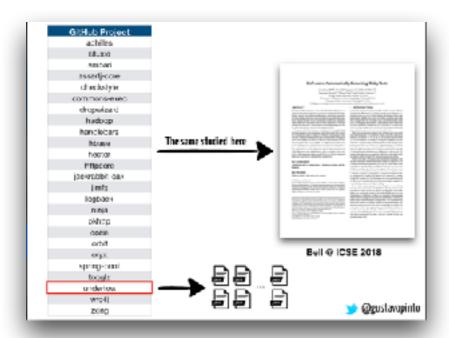
## RQ4: What's the vocabulary?

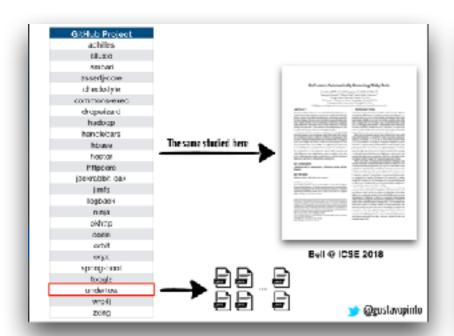
		F	laky	Nor	า-Flaky
Features	inf. gain	# test	# projects	# test	# projects
job	0.20	524	2	4	1
table	0.14	406	4	8	2
id	0.14	552	9	52	4
action	0.13	387	3	8	2
oozie	0.13	274	1	0	0
services	0.13	371	2	7	1
coord	0.11	307	1	0	0
getid	0.11	287	4	1	1
coordinator	0.10	258	1	0	0
xml	0.10	147	2	6	2
workflow	0.09	207	1	0	0
getstatus	0.08	246	2	2	2
record	80.0	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

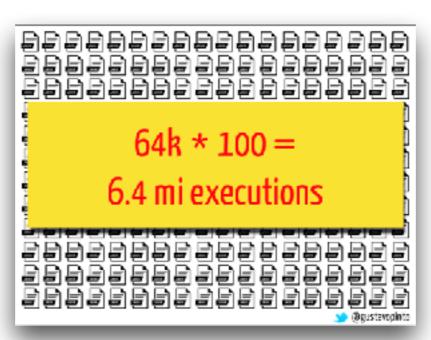
Common words: job, table, and action

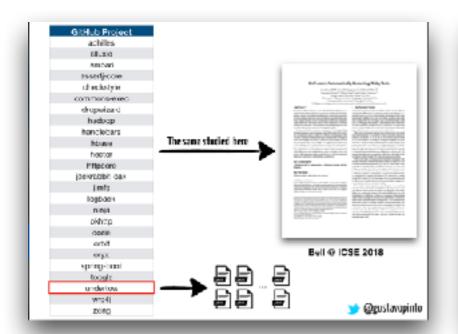
Many of them associated with remote tasks and/or queue events

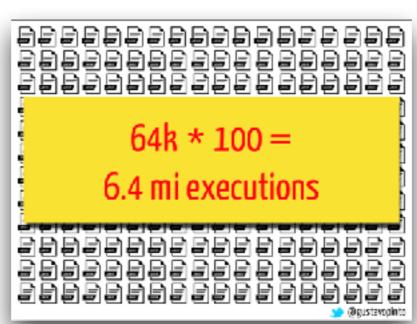


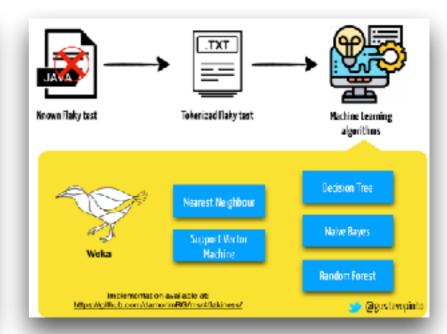


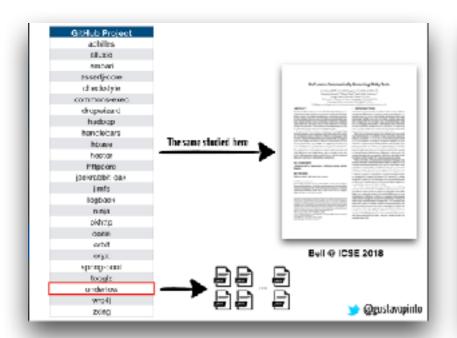


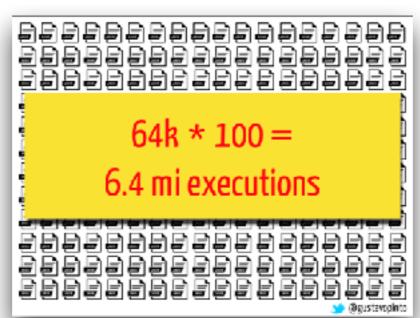


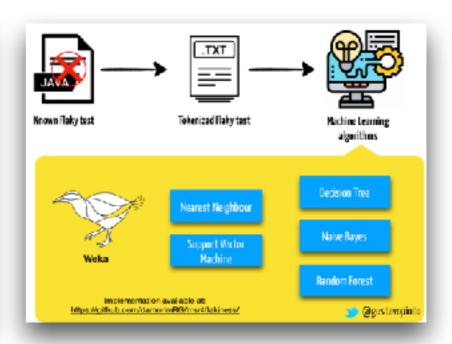


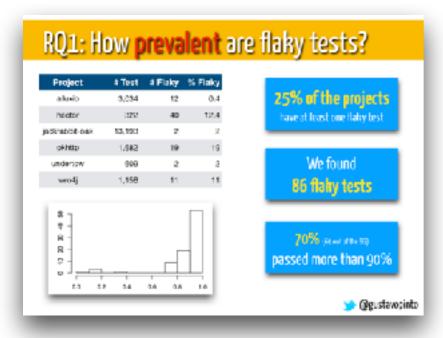




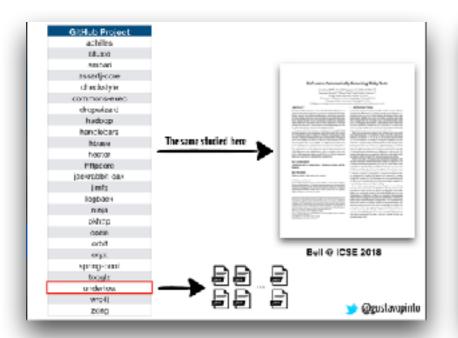


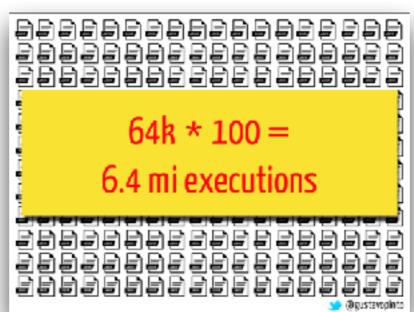


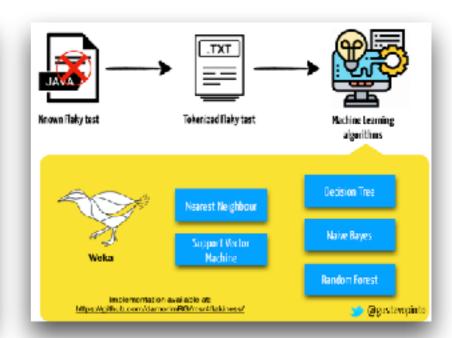














23 22 34 36 38 16

0.0



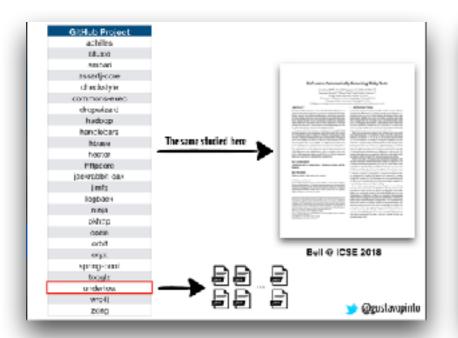
We found 86 flaky tests

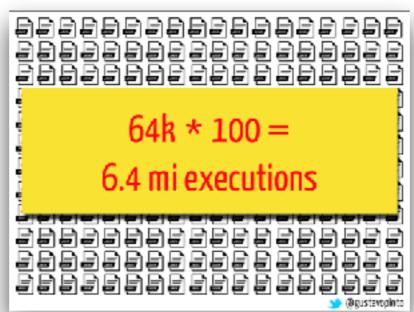
70% (Automotion 90%) passed more than 90%

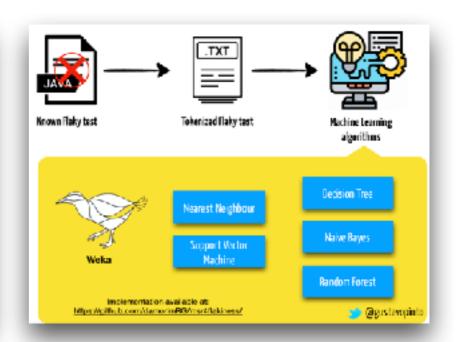
20	_	- to	- 0. 1	
R02:	Can we	predic	ct flak	riness?

ML	Precision	Receil	Pt	MCG	AUG
Random Forest	0.99	0.91	0.95	0.90	0.98
Dycision Tiew	0.98	0.88	0.29	0.77	0.91
Naive Bayes	0.93	0.80	0.36	0.74	0.33
Support Vector	0.33	98.0	0.33	0.35	0.32
vecrest Naighbour	0.97	G.83	9.52	0.35	0.53





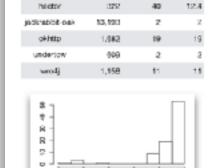






0.4

4 Tost 4 Flaky % Flaky



12 14 16 16 16

3,034

aluvio



We found 86 flahy tests

70% parameter passed more than 90%

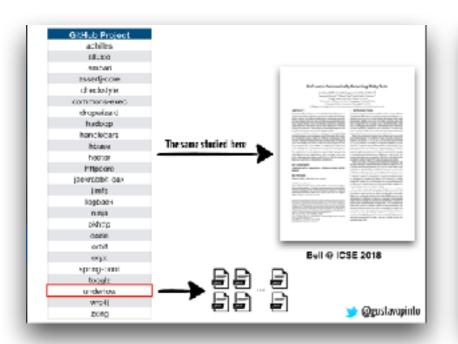
#### RQ3: What's the value of different features

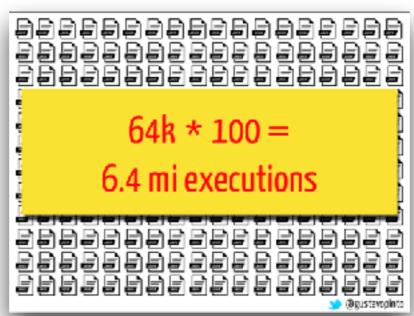
Random Forest					
Features	Precision	Recall	Ff	Mcc	AUC
All features	0.99	0.91	0.95	0.90	0.98
No sterring	0.99	0.91	0.95	0.90	0.98
No Stop W. removal	0.99	0.91	0.95	0.90	0.98
No Lowercasing	0.98	0.91	0.94	0.29	0.98
No Java Keywords	0.99	0.90	0.94	0.89	0.98
Support Vector Mach	ine				
Features	Precision	Recall	Ft	MCC	AUC
All fratures	0.90	0.82	0.93	0.05	0.33
No stemming	0.50	(1.162	0.90	0.75	0.93
No Stop W. removal	0.93	0.92	0.33	0.85	0.33
No Lowerbasing	0.571	11311	0.92	D354	0.522
No Java Keywords	0.93	0.02	0.93	0.35	0.33

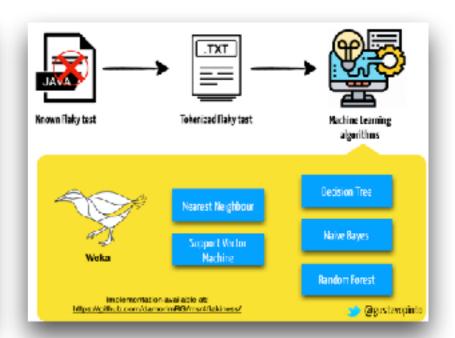
#### RQ2: Can we predict flakiness?

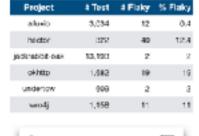


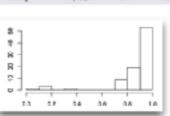












25% of the projects	
have at least one flaky test	

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 Keywords	0.99	0.90	0.94	0.29	0.98	
Support Vector Mach	ine					
Feetures	Precision	Rectil	Ft	MCC	AUG	
All fratures	0.50	0.82	0.93	0.05	0.83	
No stemming	0.501	(1.102	0.93	0.75	0.93	
No Stop W. removal	0.93	0.82	0.33	0.85	0.33	

0.83

0.93

0.35 0.33

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	Random Forest ach exed best precision
Dycision 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	0.33	98.0	0.33	0.35	0.32	
Veorest Naighbour	0.97	58.D	9.92	0.36	0.53	Tuning (a.g. = oftress) Had no performance impact
						🍑 Ø/gustavopim

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

	1-Flaky	Non-Flaky		Flaky		
	# projects	# test	# projects	# test	inf. gein	Features.
Common words:	1	4	2	624	0.20	job
job, table, and acti	52	8	4	400	0.14	mble
	4	92	9	352	0.14	id
300, 100.0, 0.00 00.00	2	8	5	387	0.13	action
	0	0	1	274	0.18	coale
	1	7	2	371	0.18	versions
	0	0	1	307	0,11	coord
Many of them associated with	1	- 1	4	387	0,11	9390
	0	0	1	256	0.10	coordinator
remote tasks	2	6	5	147	0.10	XIII
	0 2	0	1	207	0.09	workflow
and/or	2	2	2	246	0.08	gelstatus
manus mande	1	18	2	296	0.08	record
queue events	0	0	2	207	0.07	[D1
		0	1	200	0.07	passervice
	0	67	4	367	0.07	service

