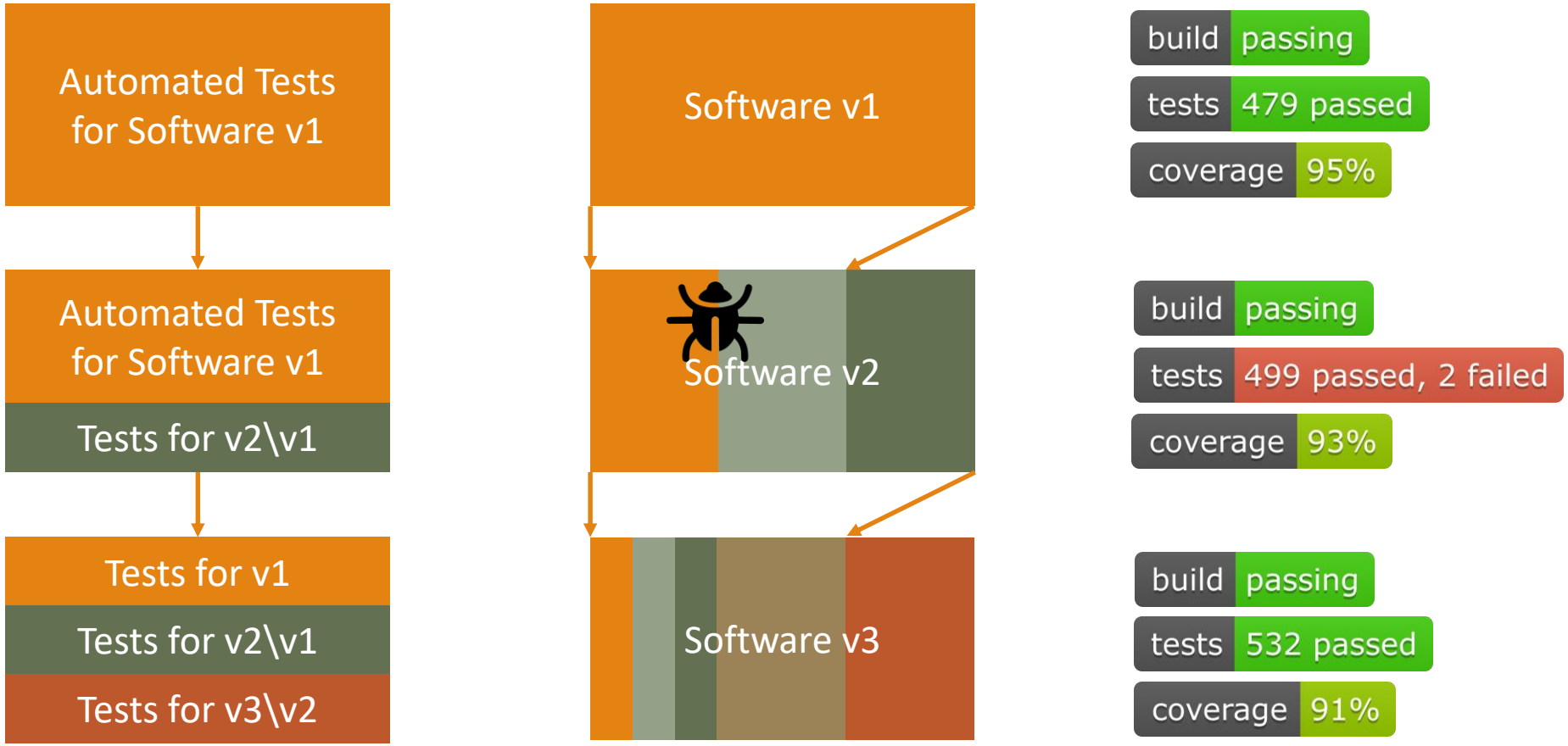
Flaky Test Detection: Earlier and Faster

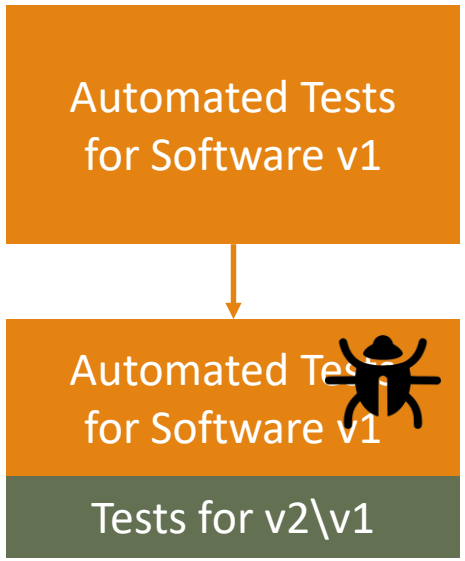
SLIDES: HTTPS://WWW.STEFAN-WINTER.NET/APPLICATION-MATERIALS/

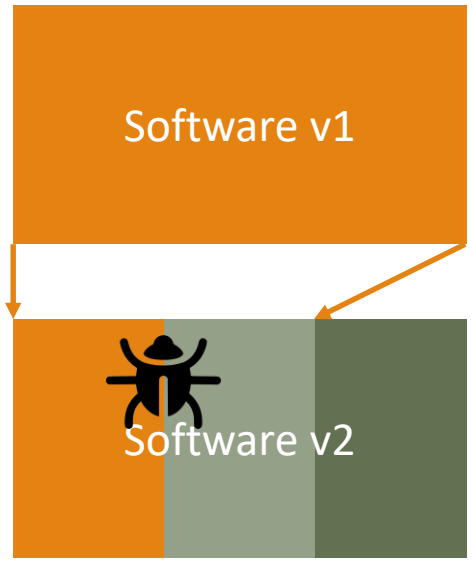
Test Automation & Regression Testing

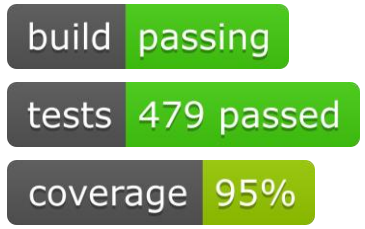


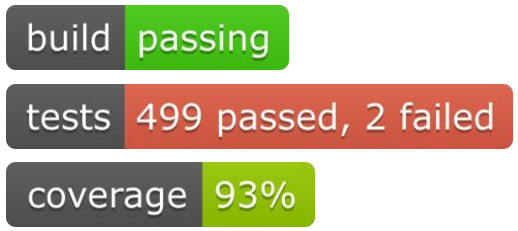
Flaky Tests

- Flaky tests can **pass *and* fail** without changes to
  - code under test
  - test code
  - runtime environment









Flaky Test Detection in 2019: NonDex

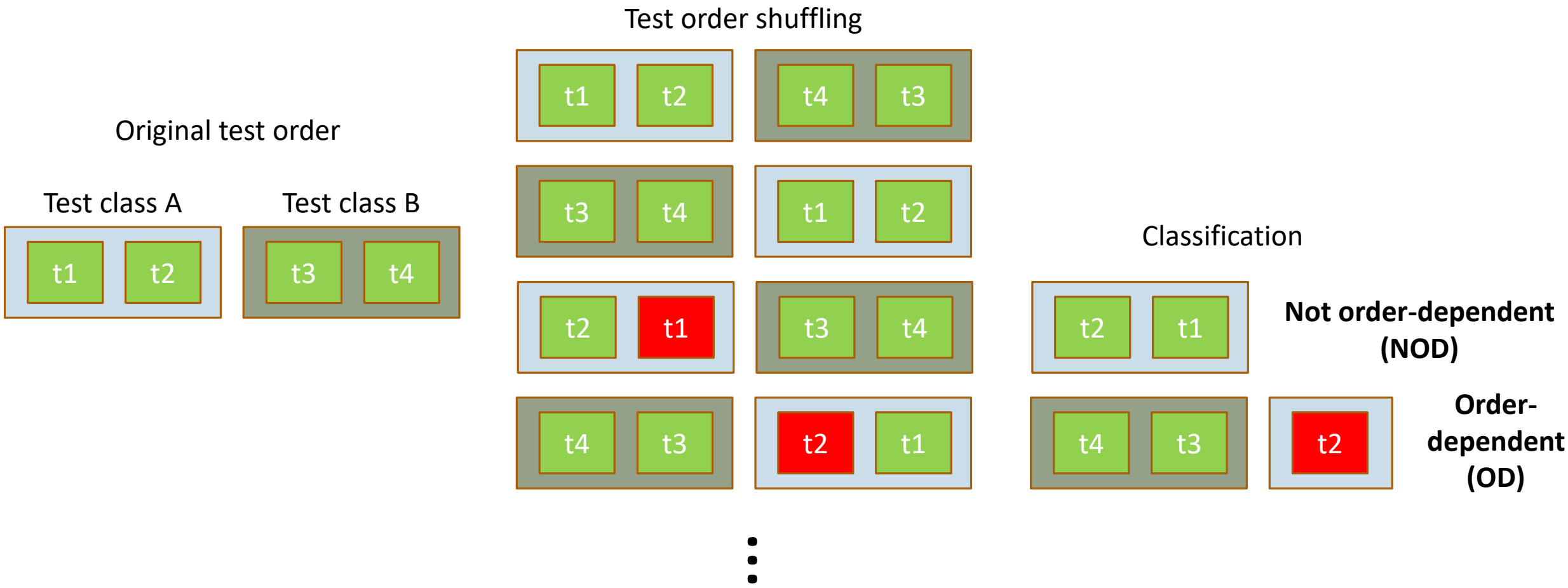
Java classes with non-deterministic specifications (examples):

Class	Methods	Category
java.lang.Object	hashCode	Random
java.util.HashMap	keySet, values, entrySet	Permute
java.util.ConcurrentHashMap	keySet, values, entrySet, keys, elements	Permute
java.io.File	list, listFiles, listRoots	Permute
java.lang.Class	getClasses, getDeclaredMethods, …	Permute

Implementation-dependent (ID) flaky test detection:

- Run test suite 10-100x with different random seeds
- For each call to identified methods:
  - return randomized result

Flaky Test Detection in 2019: iDFlakies



Problems with Flaky Test Detection 2019

Costs for detecting ID flaky tests w/ NonDex: 10-100 reruns

Costs for detecting OD flaky tests: <#test classes> x $\prod$ <#test methods per class> reruns

Range of test counts for projects in iDFlakies dataset: 4-10,503

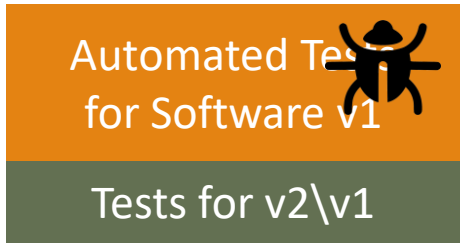
- up to 100,000 – 1,000,000 test runs with NonDex (for detecting ID flaky tests)
- up to 7.4E+37676 test suite runs with iDFlakies for detecting OD flaky tests (plus classification runs)
- an unpredictable number of runs for NOD
- for every commit to code under test or test code?

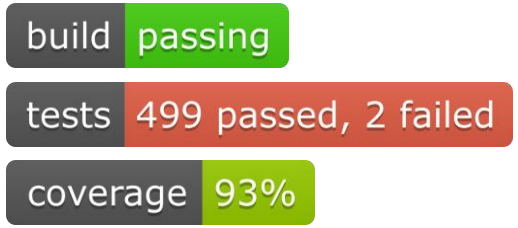
Our two contributions in 2020:

- **detecting ID & OD flaky tests earlier** by reducing the number of commits to run detectors on
- **detecting NOD flaky tests faster** by empirically determined heuristics

Detecting Flaky Tests Earlier

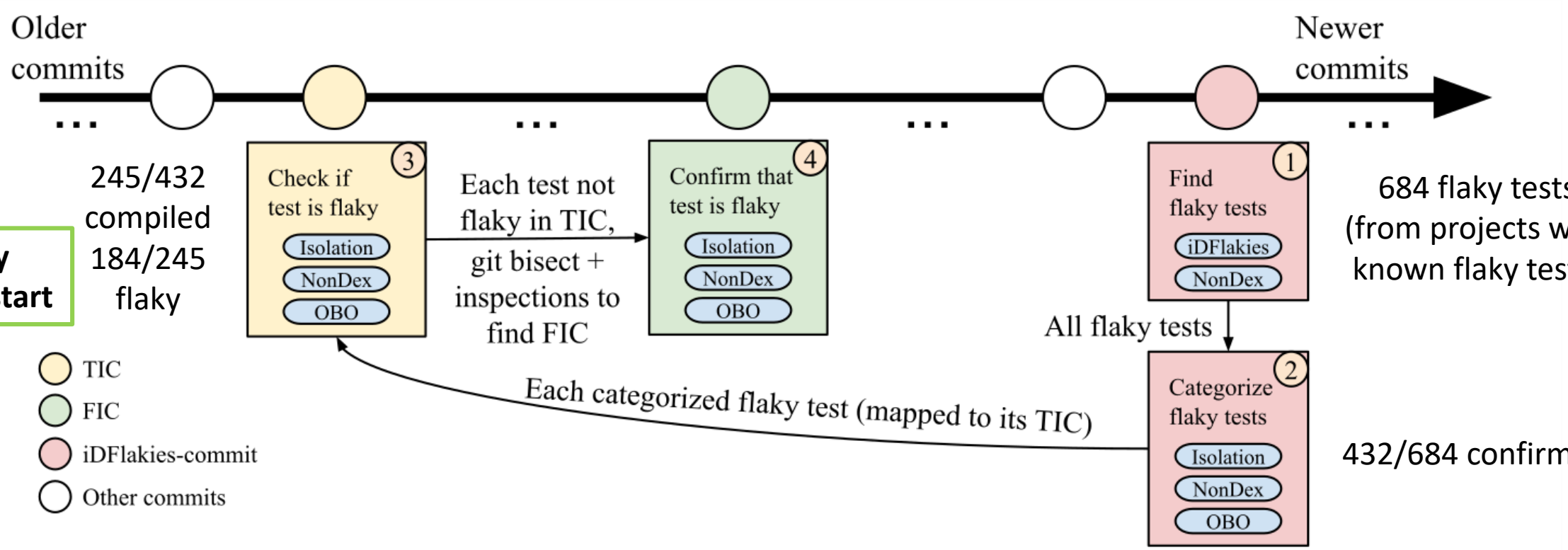
Wing Lam, Stefan Winter, Anjiang Wei, Tao Xie, Darko Marinov, Jonathan Bell:
“A Large-Scale Longitudinal Study of Flaky Tests”, OOPSLA‘20





- Are flaky tests detectable *when the test code is committed* or later?
- If later: Are there indicators on which commits we should run detectors?

Detecting Flaky Tests Earlier: Approach



>75% flaky from the start

Study Results: When to Run Detectors (1)

Are flaky tests detectable *when the test code is committed* or later?

- 184/245 (75%) flaky tests were detectable right from the start

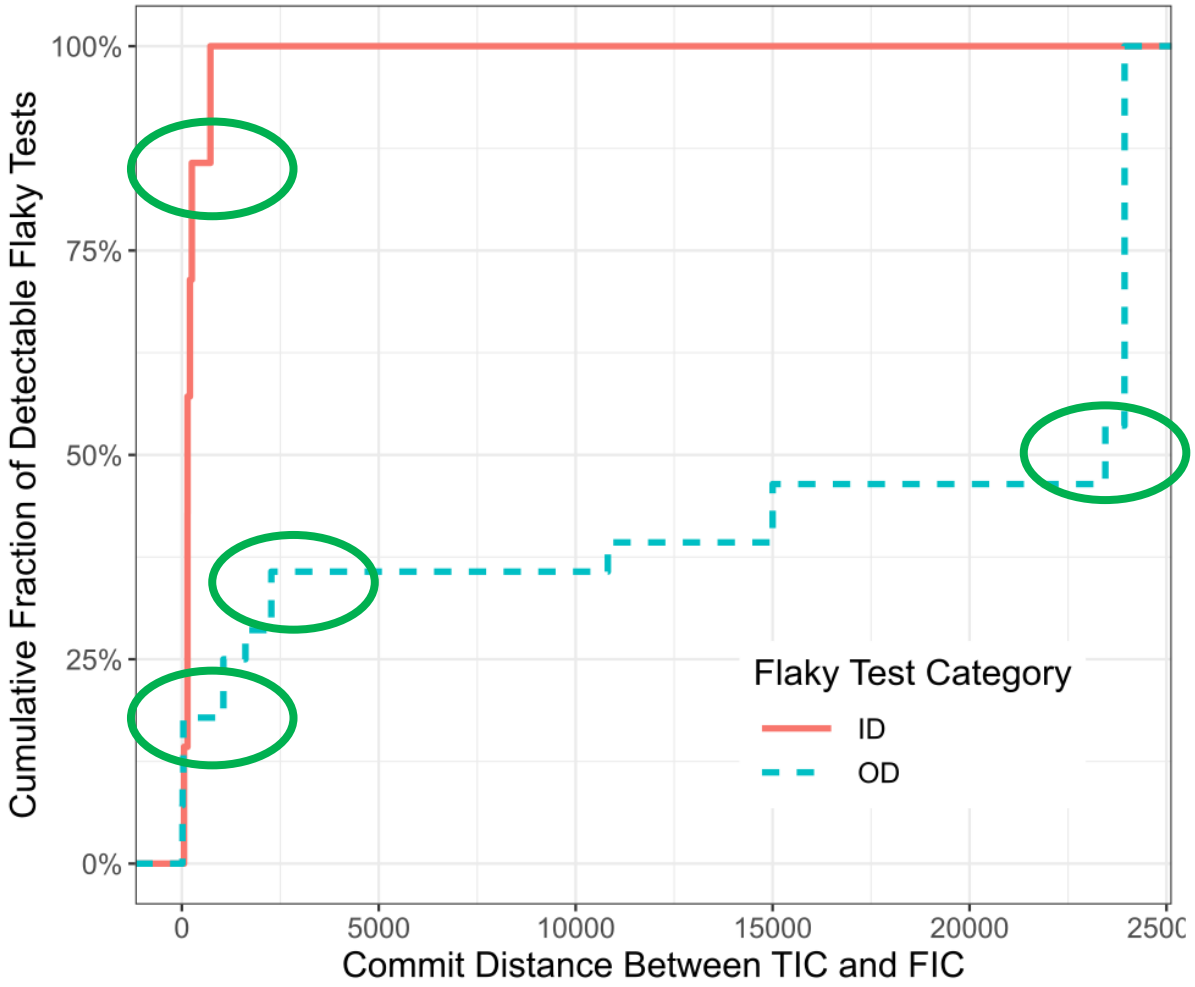
If later: Are there indicators on which commits we should run detectors?

- Of the remaining 61 tests, 24 become flaky after change to test class
- Of the remaining 37 tests
  - 8 would be detected 62 commits (median) after the FIC if detectors only run on test class changes
  - 29 could be detected 3 commits (median) after the FIC, but only when detectors run on all changes to all test code

85% of flaky tests can be detected when they are introduced or their code is changed.

To catch the remaining ones, we need to run detectors periodically on all tests.

Study Results: When to Run Detectors (2)



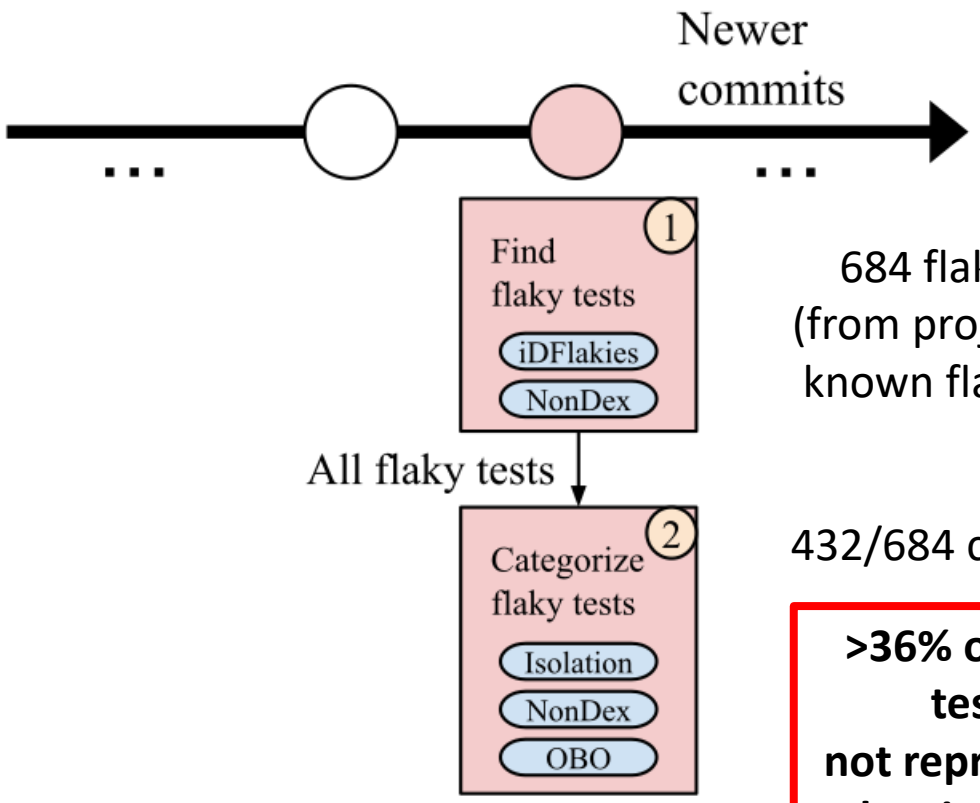
Commit distance between TIC and FIC varies for different flaky test types

- ID: 85% become flaky within 254 commits, 100% in 726 commits
- OD: 18% become flaky within 38 commits, 35% within 2273 commits, 53% within 23448 commits, 100% within 23936 commits

85% of flaky tests can be detected when they are introduced or their code is changed.

Run ID detection (NonDex) every 250 commits. For OD detection (iDFlakies), decrease frequency gradually with increasing descendant commits.

Insufficiency of Existing Detectors

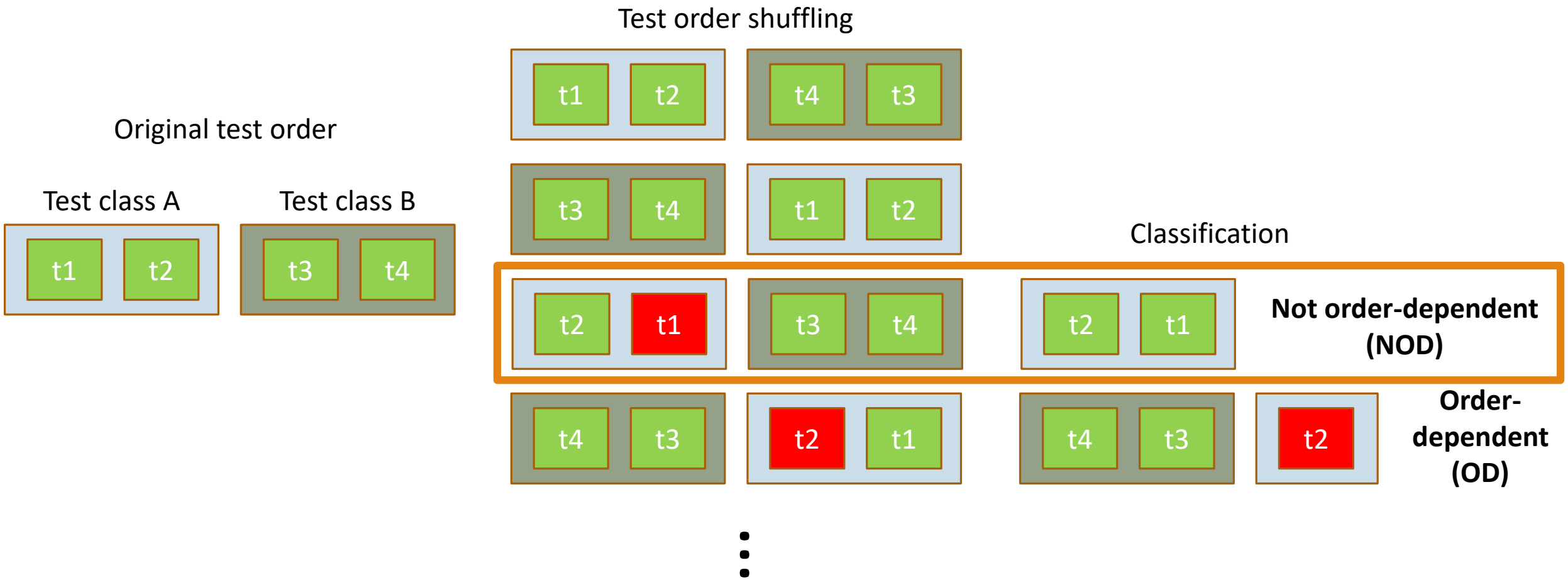


684 flaky tests
(from projects with
known flaky tests)

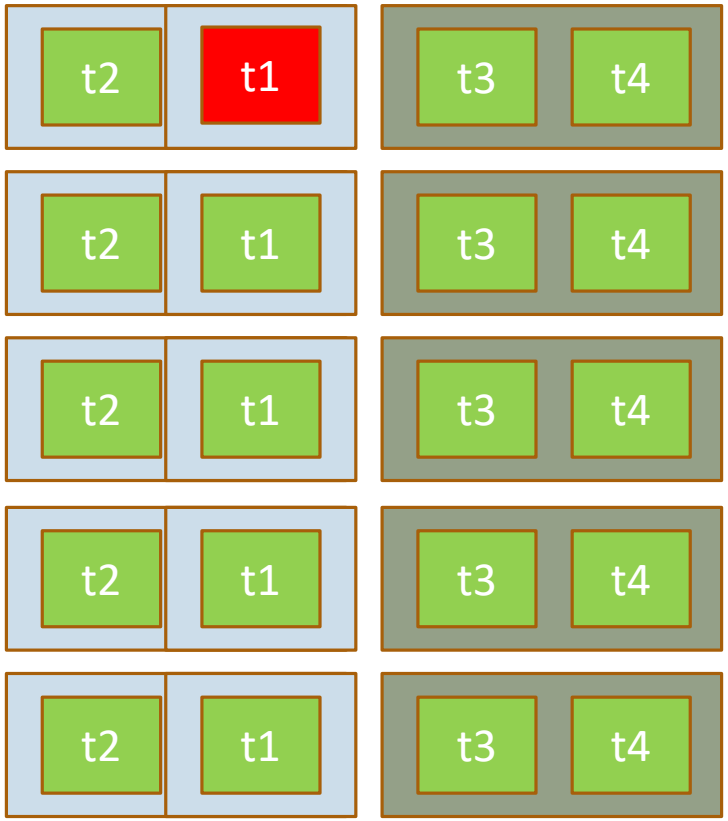
432/684 confirmed

>36% of flaky tests not reproduced despite > 200 reruns

Another Class of Flaky Tests: NOD



NOD Detection & Debugging



1. Slow detection
   - How to provoke NOD flaky test failures?
2. Slow debugging
   - Failure rates in test suite execution vs. isolation?

Detecting NOD Flaky Tests Faster

Wing Lam, Stefan Winter, Angello Astorga, Victoria Stodden, Darko Marinov: "Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects", ISSRE'20

Exploratory study: Reproducibility of NOD flaky test failures

- 62 modules with NOD flaky tests in iDFlakies dataset
- Repeat each test suite run 2,000 times and collect detailed execution logs
- Inspect failure cases

Initial dataset reduction

- 48/62 modules compiled
- 44/48 completed 2,000 runs
- 26/44 modules had 1 or more flaky test failures

Study Results

Finding 1: Test orders change and affect “not order-dependent” tests.

- Up to 20 orders observed across 2,000 runs
- Re-classify NOD as “not deterministic, order-dependent” or “not deterministic, order-independent”

1500 ms timeout

```
@Test(timeout = 1500)
public voidshouldRetryWithDynamicDelayDate() {
    ...// test setup
    atLeast(Duration.ofSeconds(1),() ->
      unit.get("/baz").dispatch(...).join());
}
```

- server-startup that may take > 1500ms
- if other tests run before, JIT compiler is triggered

Study Results: Faster Detection

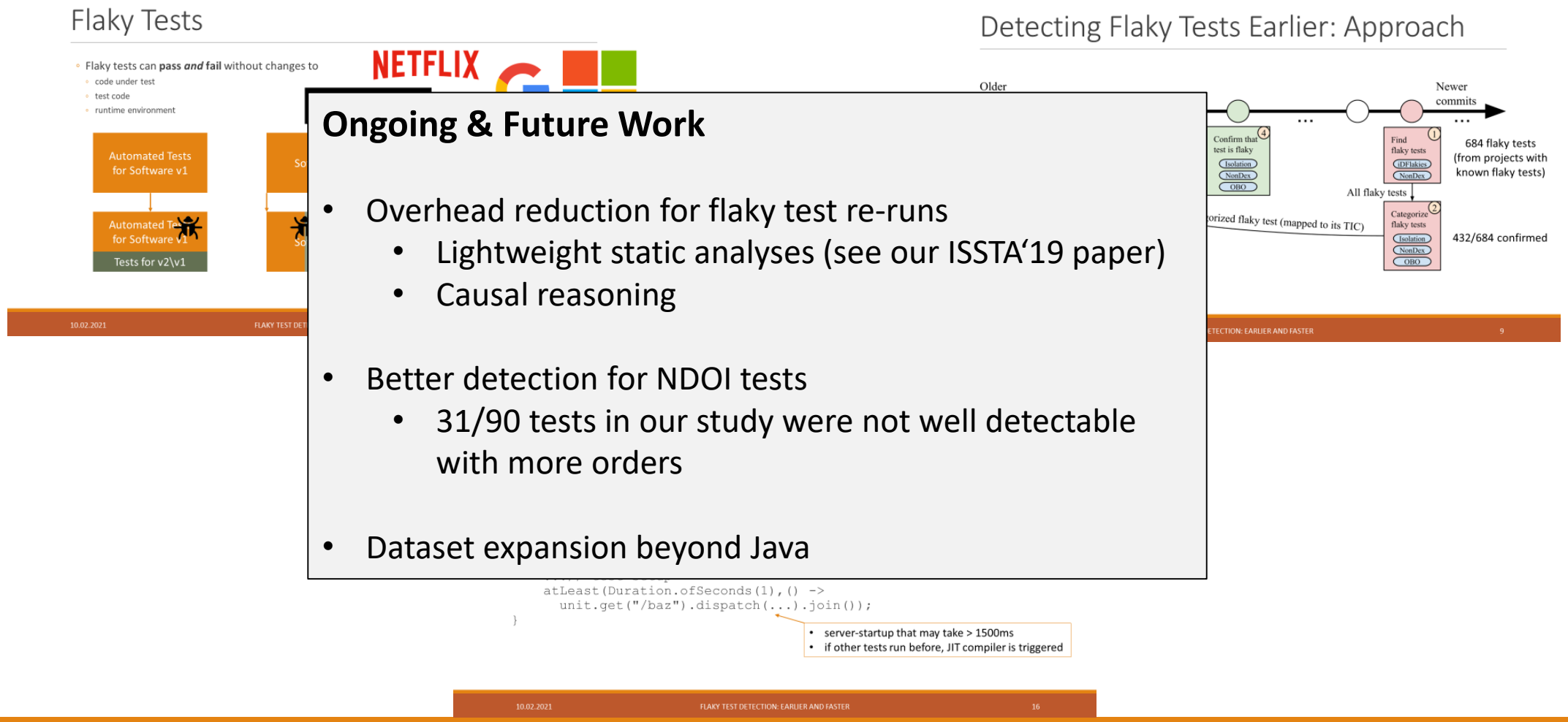
Finding 2: Running more orders fewer times yields faster detection.

- Additional 2,000 test suite runs with fixed order (“reverse alphabetical”)
- Comparison with 100 runs for 20 orders
- Higher/identical/lower rate with more orders for 55/4/31 tests

Finding 3: Isolation runs have limited value for flaky test debugging.

- Additional 4,000 test runs for each flaky test in isolation
- **Caveat:** 53% of the flaky tests we found in test suite runs cannot be found in isolation

Summary


Flaky Tests
Flaky tests can pass and fail without changes to
code under test
test code
runtime environment
NETFLIX
Automated Tests for Software v1
Tests for v2\v1
10.02.2021
Detecting Flaky Tests Earlier: Approach
Older
Newer commits
Confirm that test is flaky
Isolation
NonDex
OBO
Find flaky tests
iDFlakies
NonDex
684 flaky tests (from projects with known flaky tests)
All flaky tests
Categorize flaky tests
Isolation
NonDex
OBO
432/684 confirmed
ETECTION: EARLIER AND FASTER
9
Ongoing & Future Work
Overhead reduction for flaky test re-runs
Lightweight static analyses (see our ISSTA'19 paper)
Causal reasoning
Better detection for NDOI tests
31/90 tests in our study were not well detectable with more orders
Dataset expansion beyond Java
atLeast(Duration.ofSeconds(1),() ->
unit.get("/baz").dispatch(...).join());
}
server-startup that may take > 1500ms
if other tests run before, JIT compiler is triggered
10.02.2021
FLAKY TEST DETECTION: EARLIER AND FASTER
16