Mind the Gap: What Working With Developers on Fuzz Tests Taught Us About Coverage Gaps

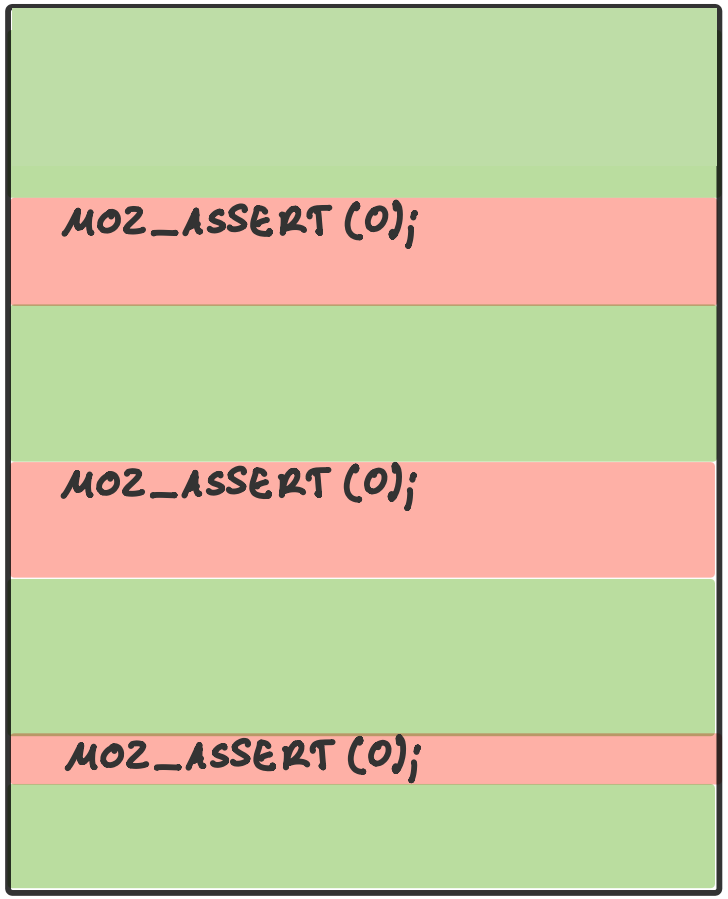
Carolin Brandt, Marco Castelluccio, Christian Holler, Jason Kratzer, Andy Zaidman, Alberto Bacchelli

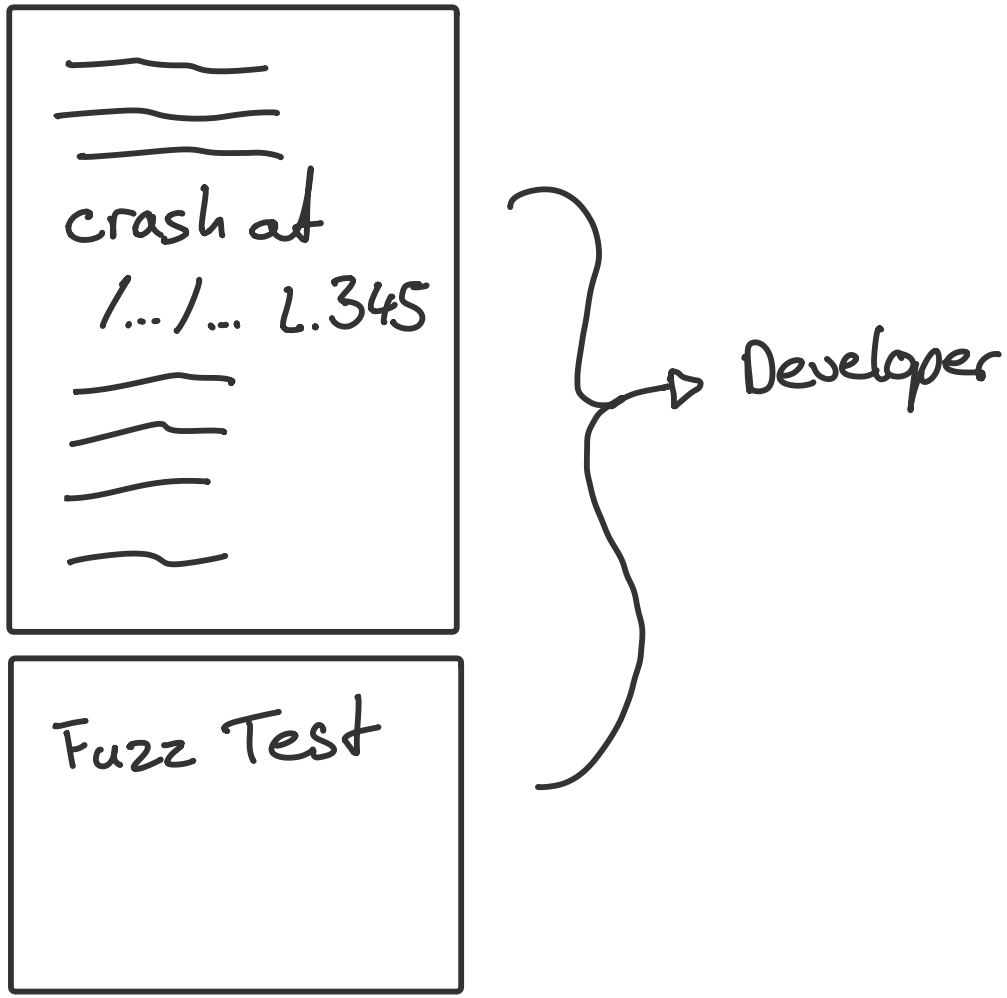
TU Delft, Mozilla, University of Zurich

ICSE-SEIP 2024

Can fuzzers generate partial tests that developers find useful enough to complete into functional tests?

Instrumentation





What does a test look like?

```
<script>
window.requestIdleCallback(window.close, {timeout: 10000})
</script>
<style>
html:last-of-type, #htmlvar00001 {
  text-align-last: start; }
.class0, aside:nth-last-child(2) {
  column-width: 1em;
</style>
<table>
<colgroup width="3" span="20">+GEE&gt;uo/c(wt6,N:1=*</colgroup>
<caption class="class0">
```

Fuzz Test

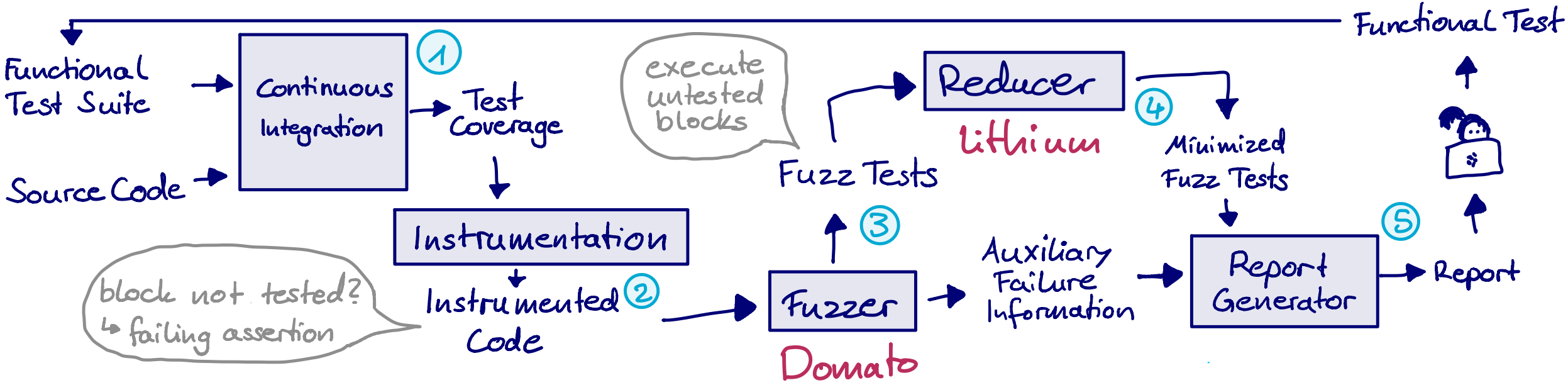
```
https://searchfox.org/mozilla-central/source/layout/generic/test/test_bug1566783.html
<!doctype html>
<title>Test for scroll anchoring adjustments during onload</title>
<script src="/tests/SimpleTest/SimpleTest.js"></script>
<script>
  SimpleTest.waitForExplicitFinish();
</script>
<link rel="stylesheet" href="/tests/SimpleTest/test.css"/>
<iframe width="300" height="300“ src="file_bug1566783.html#slow"></iframe>
```

```
https://searchfox.org/mozilla-central/source/layout/generic/test/file_bug1566783.html
<!doctype html> <style> .spacer { height: 200vh; } </style>
<script>
  function loadFailed() {
    parent.ok(false , "Image load should not fail");
  }
</script>
<div class="spacer"></div>
<img id="fast" src="file_SlowTallImage.sjs?fast" onerror="loadFailed()">
<div class="spacer"></div>
<img id="slow" src="file_SlowTallImage.sjs?slow“ onerror="loadFailed()">
<div class="spacer"></div>
<script>
onload = function () {
  setTimeout(function() {
    let rect = document.getElementById("slow").getBoundingClientRect();
    parent.is(rect.height , 1000, "#slow should take space");
    parent.is(rect.top, 0, "#slow should be at the top of the viewport");
    parent.SimpleTest.finish();
  }, 0);
} </script>
```

Test Framework Boilerplate

Functional Assertions

Functional Test

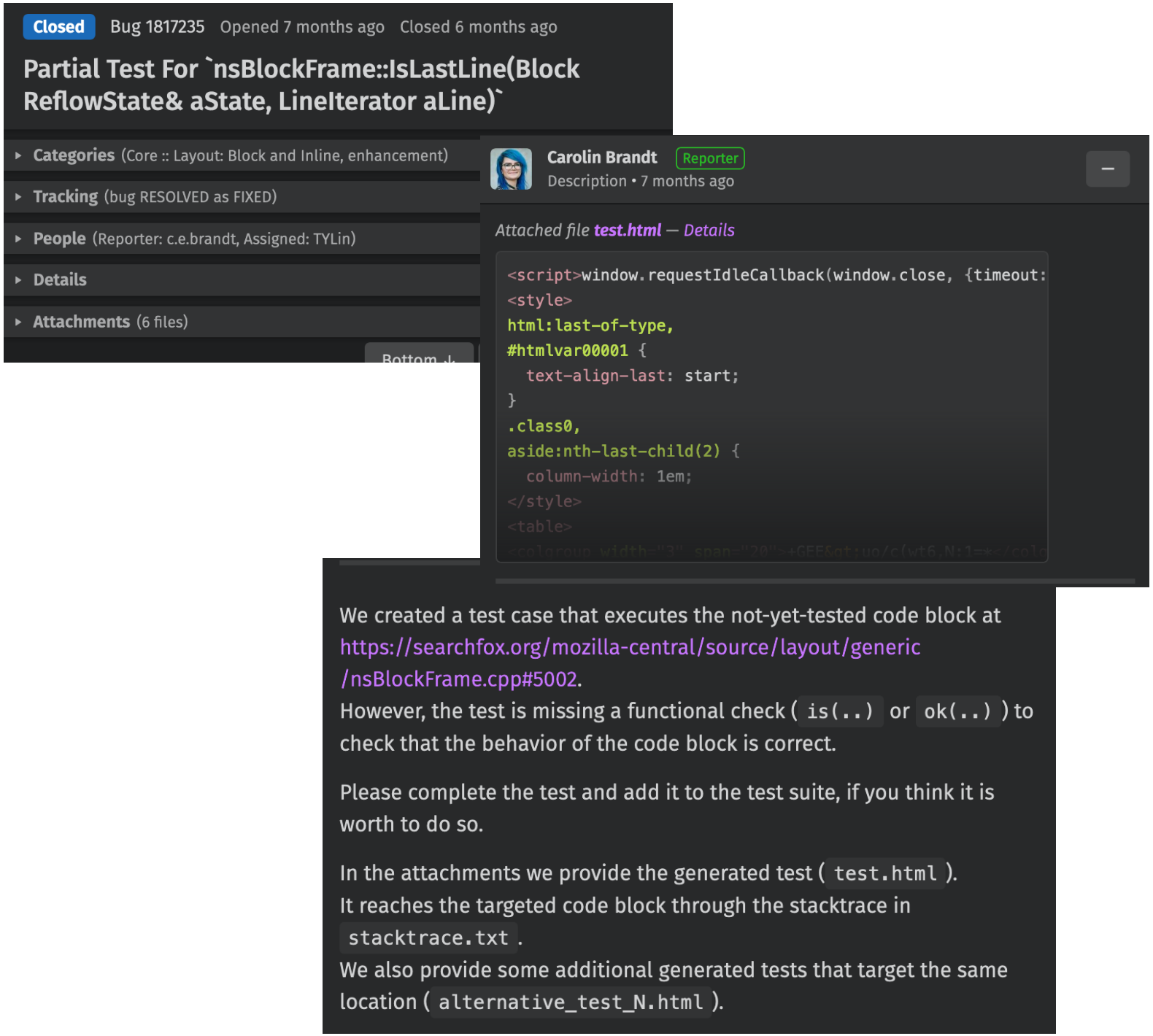

Functional Test
Functional Test Suite
Source Code
Continuous Integration
1
Test Coverage
Instrumentation
Instrumented Code
2
block not tested?
↳ failing assertion
execute untested blocks
Fuzz Tests
3
Fuzzer
Domato
Reducer
lithium
4
Minimized Fuzz Tests
Auxiliary Failure Information
Report Generator
5
Report

First Round

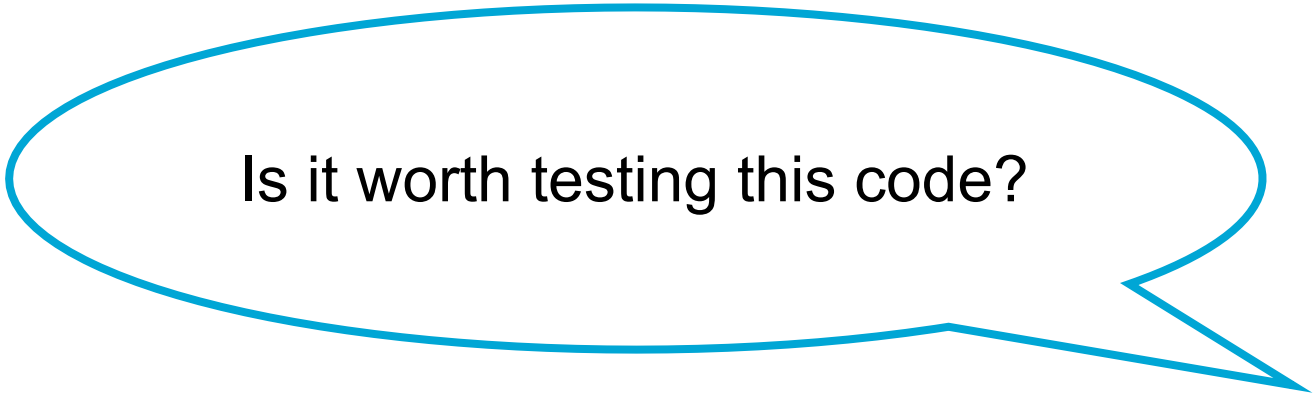
Running 30 minutes

97 tests generated
(excluding duplicates)

Submit 1 Bugzilla report per component
→ 13 reports



Developers' reactions on Bugzilla Reports



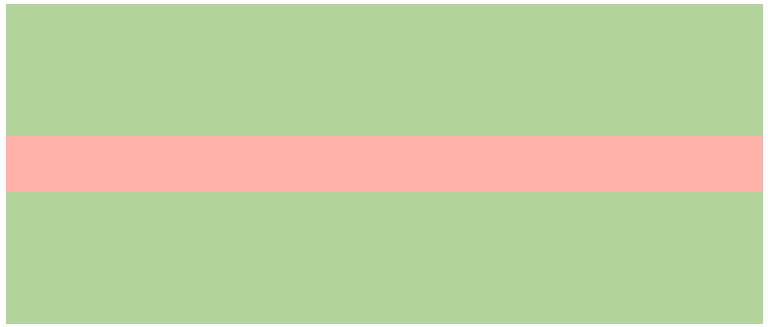
Addressed in a variety of ways:

- Write different test
- Use as inspiration for own test
- Delete code under test

Diving deeper on coverage gaps

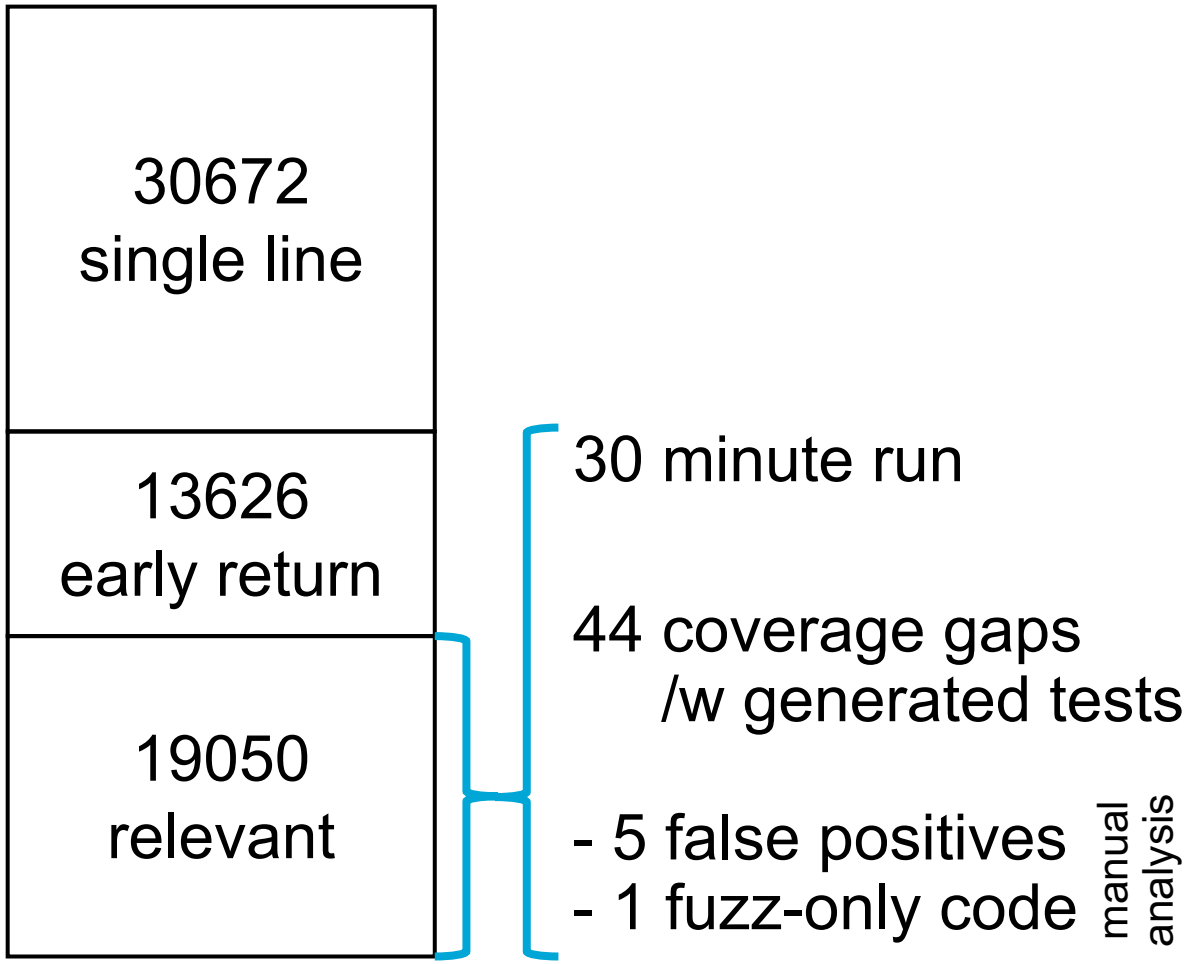
How to filter
so that we **focus on the relevant coverage gaps**?

Exclude: A) 1-line coverage gaps B) early returns



```
“NS_Warning”, “throw”, “Error”
+
“return”
```

Filtered Instrumenting + Fuzzing



Identify developers who could **judge test-worthiness:**
authors, reviewers, owners

Contact & ask:
Would this be valuable to test?
Why (not)?

13 conversations
on 1-5 coverage gaps each

Developers' opinions on filtered coverage gaps

Relevance of coverage gaps

- Better ☺
- Still reasons not worth testing
  - Unlikely bug in code
  - Unlikely reached in practice

- Enough if covered by fuzzer

Adding adapted fuzz tests

- Easier to write from scratch
- Knowledge + effort required to complete test
- Conform to test framework

Takeaways

Not all ”missing” coverage is equally worth testing

Functional tests not only way to ”cover” code

Fuzz tests can help

But need to look more like final tests

Future Work

Refined, developer-driven coverage metric

Extend /w boilerplate

+ generated assertions

Takeaways

Not all ”missing” coverage is equally worth testing

Functional tests not only way to ”cover” code

Fuzz tests can help

But need to look more like final tests

Mind the Gap: What Working With Developers on Fuzz Tests Taught Us About Coverage Gaps

Carolin Brandt, Marco Castelluccio, Christian Holler, Jason Kratzer, Andy Zaidman, Alberto Bacchelli

TU Delft, Mozilla, University of Zurich

ICSE-SEIP 2024

Mind the Gap: What Working With Developers on Fuzz Tests Taught Us About Coverage Gaps

Carolin Brandt, Marco Castelluccio, Christian Holler, Jason Kratzer, Andy Zaidman, Alberto Bacchelli

TU Delft, Mozilla, University of Zurich

ICSE-SEIP 2024

Can fuzzers generate partial tests that developers find useful enough to complete into functional tests?

Thu 18 April --- 12:00 --- Fuzzing1 Session at Fernando Pessoa

Mind the Gap: What Working With Developers on Fuzz Tests Taught Us About Coverage Gaps

Carolin Brandt, Marco Castelluccio, Christian Holler, Jason Kratzer, Andy Zaidman, Alberto Bacchelli

TU Delft, Mozilla, University of Zurich

ICSE-SEIP 2024

Developers' needs & how to improve approach

- Conform to test framework
- Receive at time of patch

- Generated test proves reachability

- Knowledge + effort required to complete test