

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

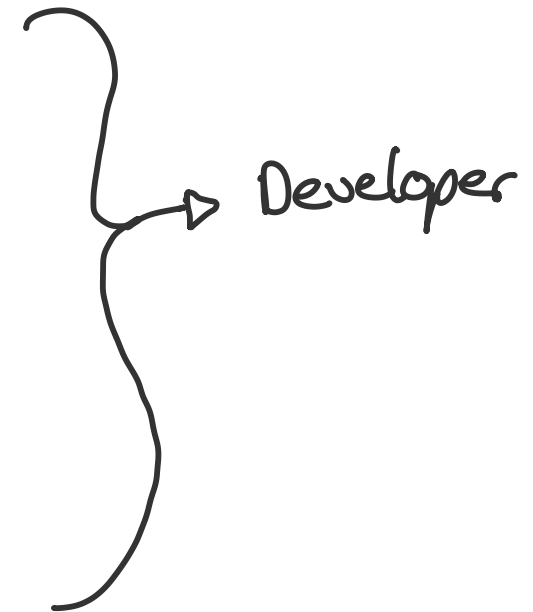
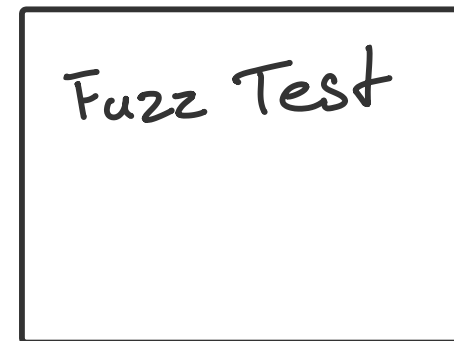
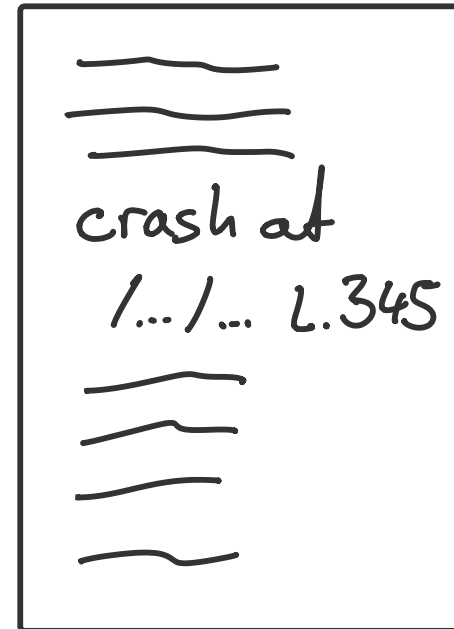
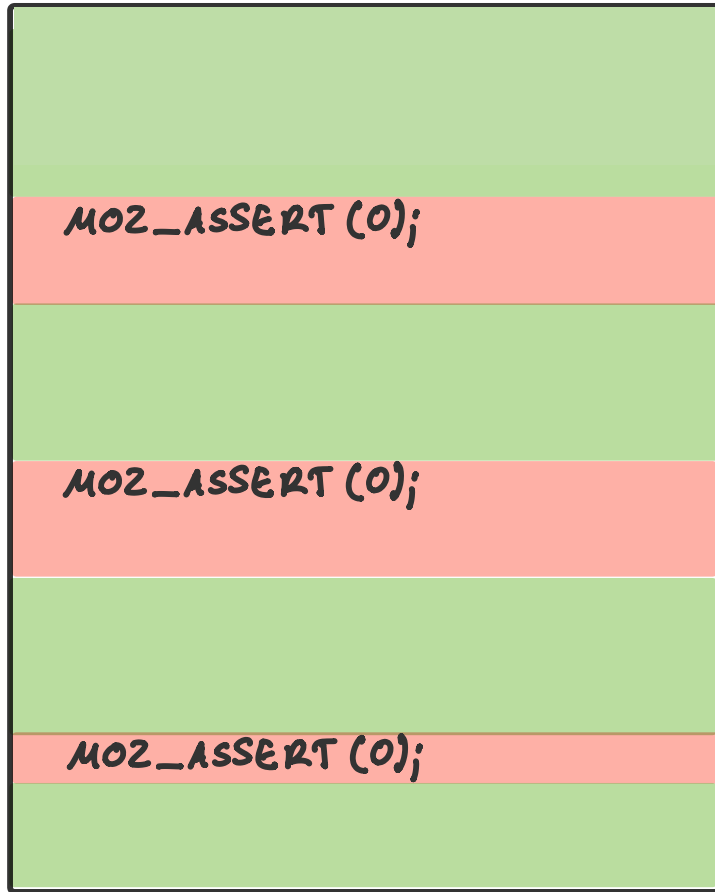
Carolin Brandt, Marco Castelluccio, Christian Holler, Jason Kratzer, Andy Zaidman, Alberto Bacchelli

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>
```

```

```

```
<div class="spacer"></div>
```

```

```

```
<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();
```

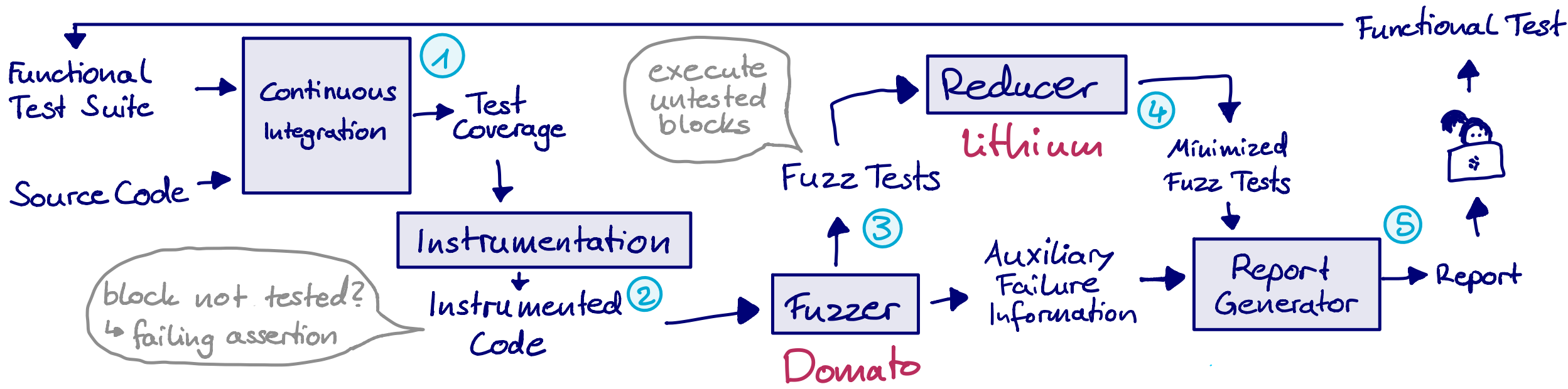
```
  }, 0);
```

```
} </script>
```

Functional Assertions

Test Framework
Boilerplate

Functional Test



First Round

Running 30 minutes

97 tests generated
(excluding duplicates)

Submit 1 Bugzilla report
per component
→ 13 reports

The screenshot shows a Bugzilla bug report interface. At the top, it indicates the bug is 'Closed' with ID 1817235, opened 7 months ago and closed 6 months ago. The title is 'Partial Test For `nsBlockFrame::IsLastLine(BlockReflowState& aState, Lineliterator aLine)`'. The report is categorized under 'Core :: Layout: Block and Inline, enhancement'. The reporter is Carolin Brandt. The bug is assigned to TYLin. The description includes an attached file named 'test.html'. A code snippet is visible, showing HTML and CSS code used for testing. The code includes a script to request an idle callback, a style rule for '#htmlvar00001', and a class rule for '.class0' with a specific 'aside:nth-last-child(2)' selector. The code also shows the start of an HTML table structure.

We created a test case that executes the not-yet-tested code block at <https://searchfox.org/mozilla-central/source/layout/generic/nsBlockFrame.cpp#5002>.

However, the test is missing a functional check (`is(..)` or `ok(..)`) to check that the behavior of the code block is correct.

Please complete the test and add it to the test suite, if you think it is worth to do so.

In the attachments we provide the generated test (`test.html`). It reaches the targeted code block through the stacktrace in `stacktrace.txt` .

We also provide some additional generated tests that target the same location (`alternative_test_N.html`).

Developers' reactions on Bugzilla Reports



Is it worth testing this code?

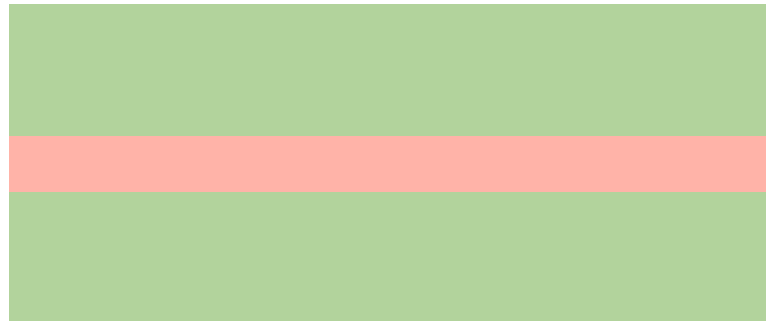
Addressed in a variety of ways:

- 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?**

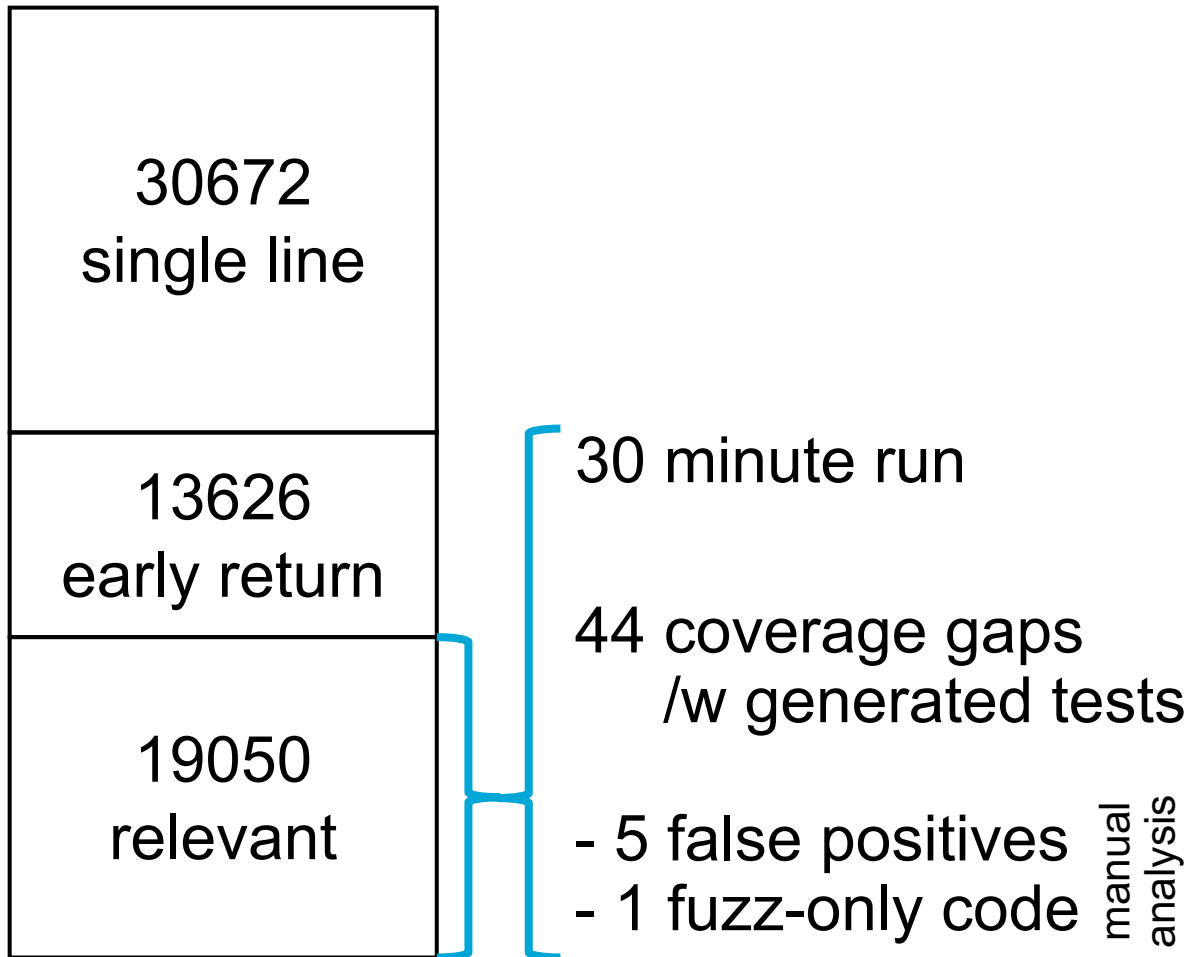
Exclude: A) 1-line coverage gaps



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

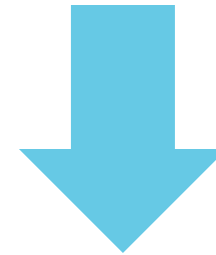
Functional tests not only way
to "cover" code



Refined, developer-driven
coverage metric

Fuzz tests can help

But need to look more like
final tests



Extend /w boilerplate
+ generated assertions

Future
Work

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

Carolyn 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