

# When to let the Developer Guide: Trade-offs between Open and Guided Test Amplification

Carolin Brandt  
Danyao Wang  
Andy Zaidman



# Test Amplification

Generate xUnit tests by  
mutating existing tests

Existing  
Test →

→ Amplified  
Tests

```
@Test
void text_escape () {
    A a = new A ("text");
    B b = a.escape()
    assertEquals (b.toString (), "text");
}
```



# Test Amplification

Generate xUnit tests by mutating existing tests

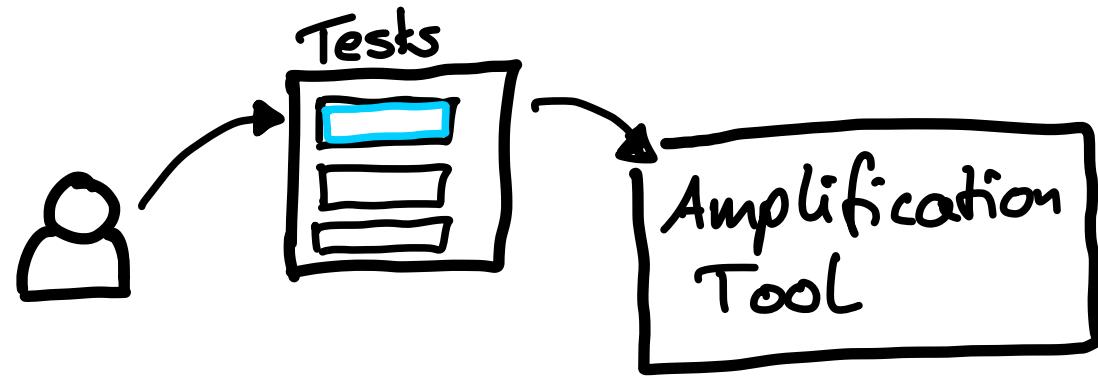


@Test

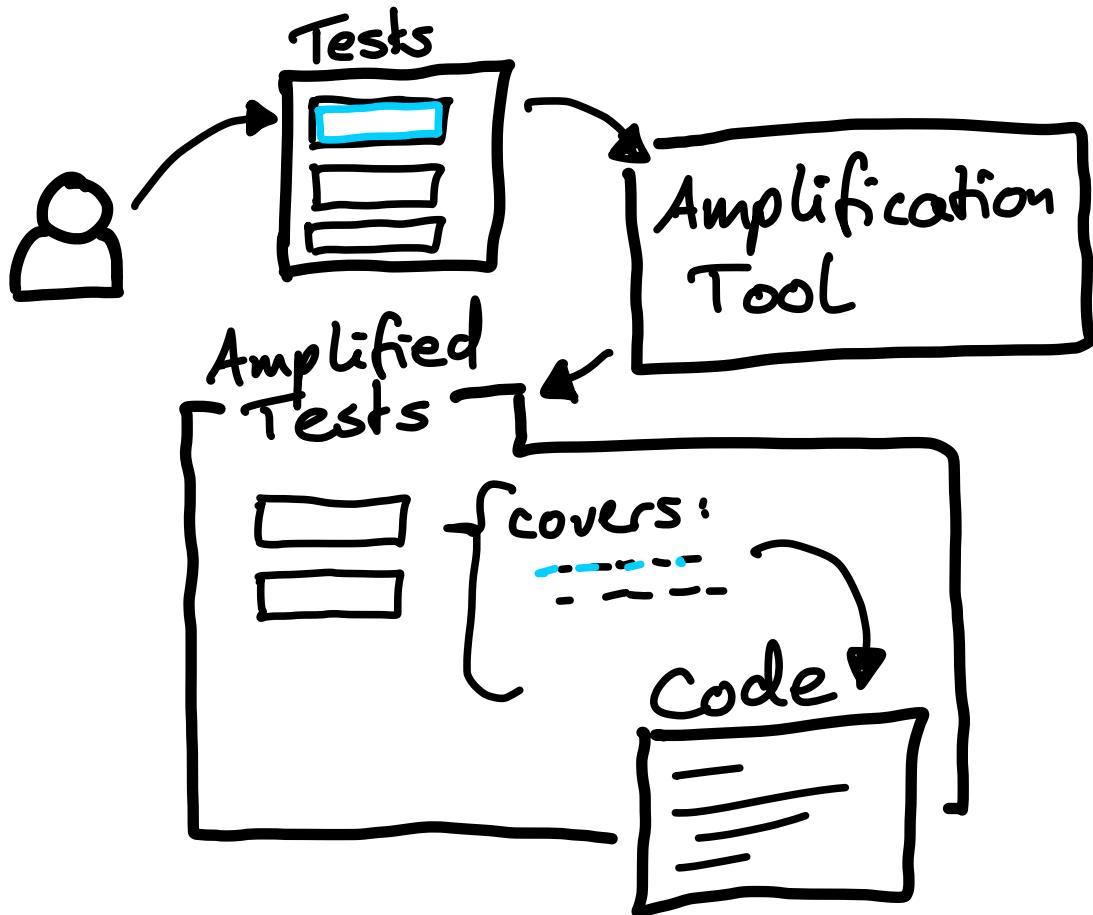
```
void text_escape () {  
    A a = new A("text"); → "text $//"  
    B b = a.escape()  
    assertEquals(b.toString(), "text"); → "text \$\\$\\\""  
}
```



# User Interaction

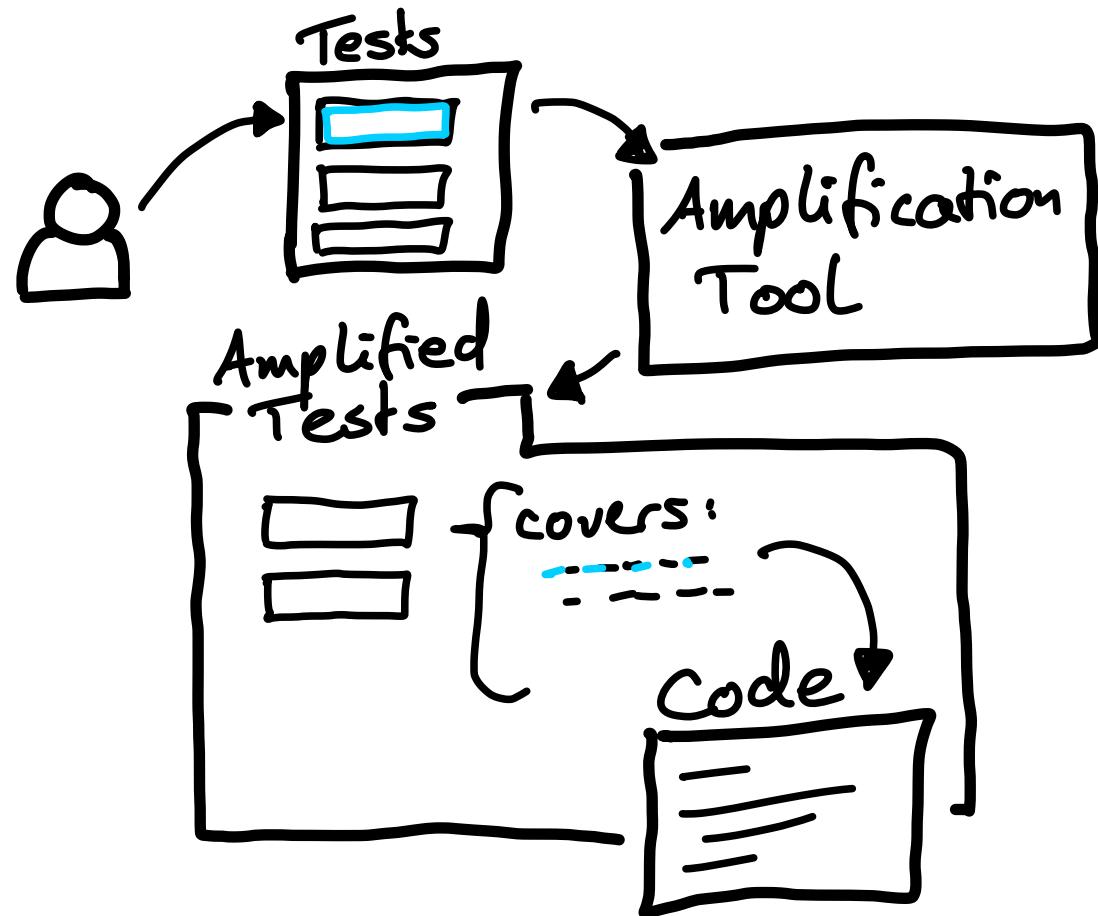


# User Interaction



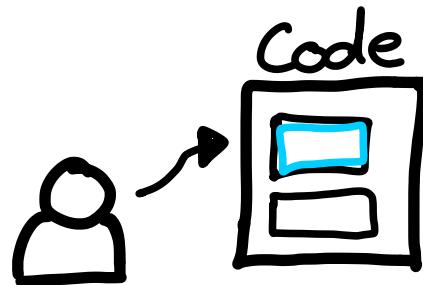
Open Test Amplification:  
Looking in all Directions for Tests

# User Interaction

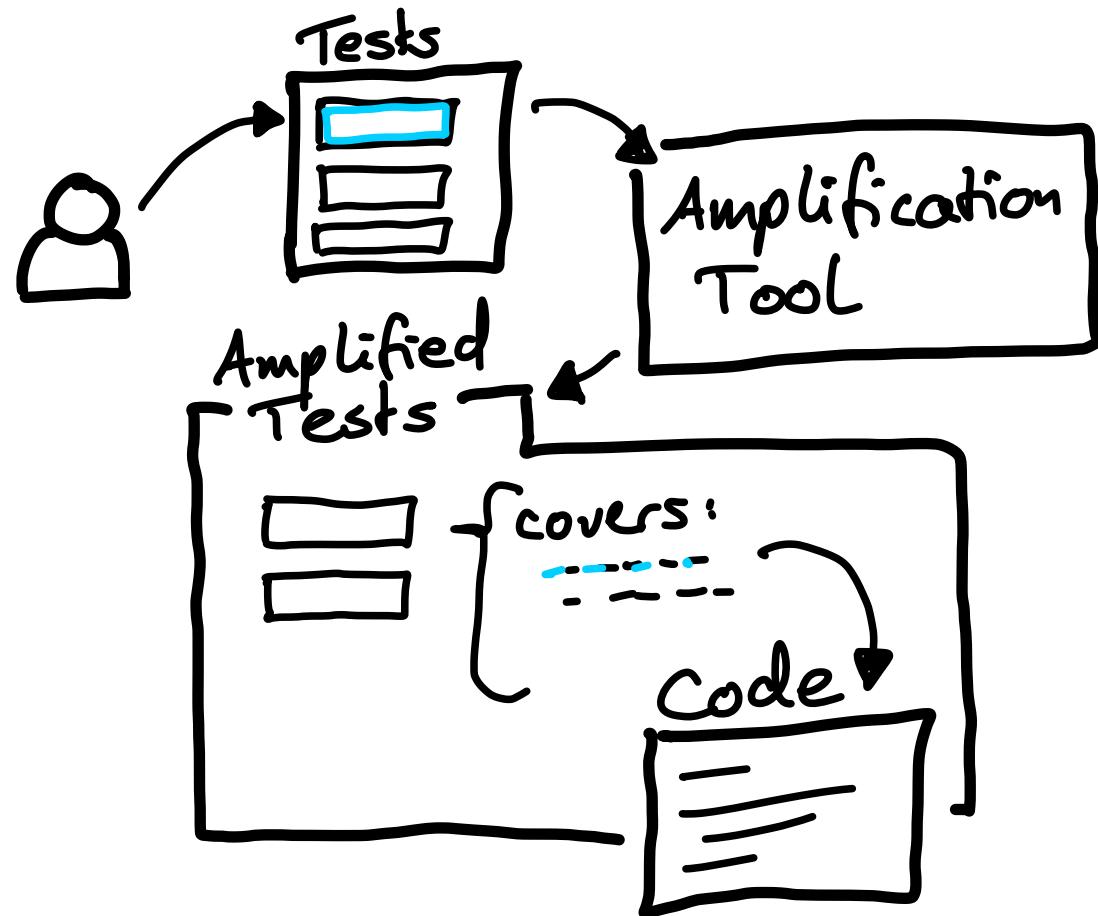


Open Test Amplification:  
Looking in all Directions for Tests

What if we let the developer  
point us to what they want to test?

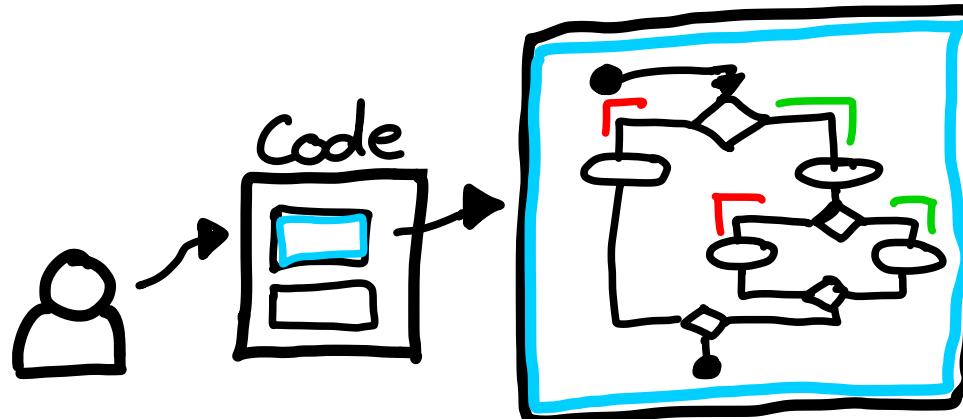


# User Interaction

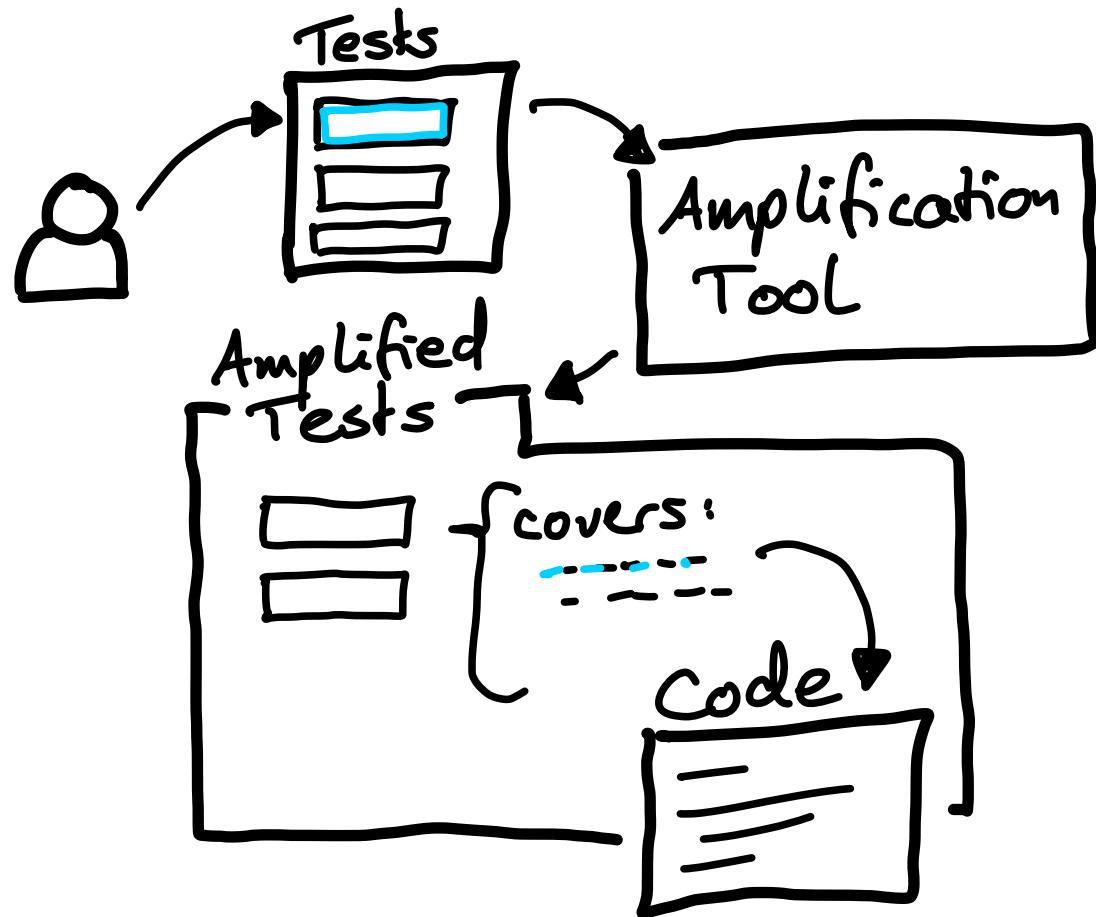


Open Test Amplification:  
Looking in all Directions for Tests

What if we let the developer point us to what they want to test?

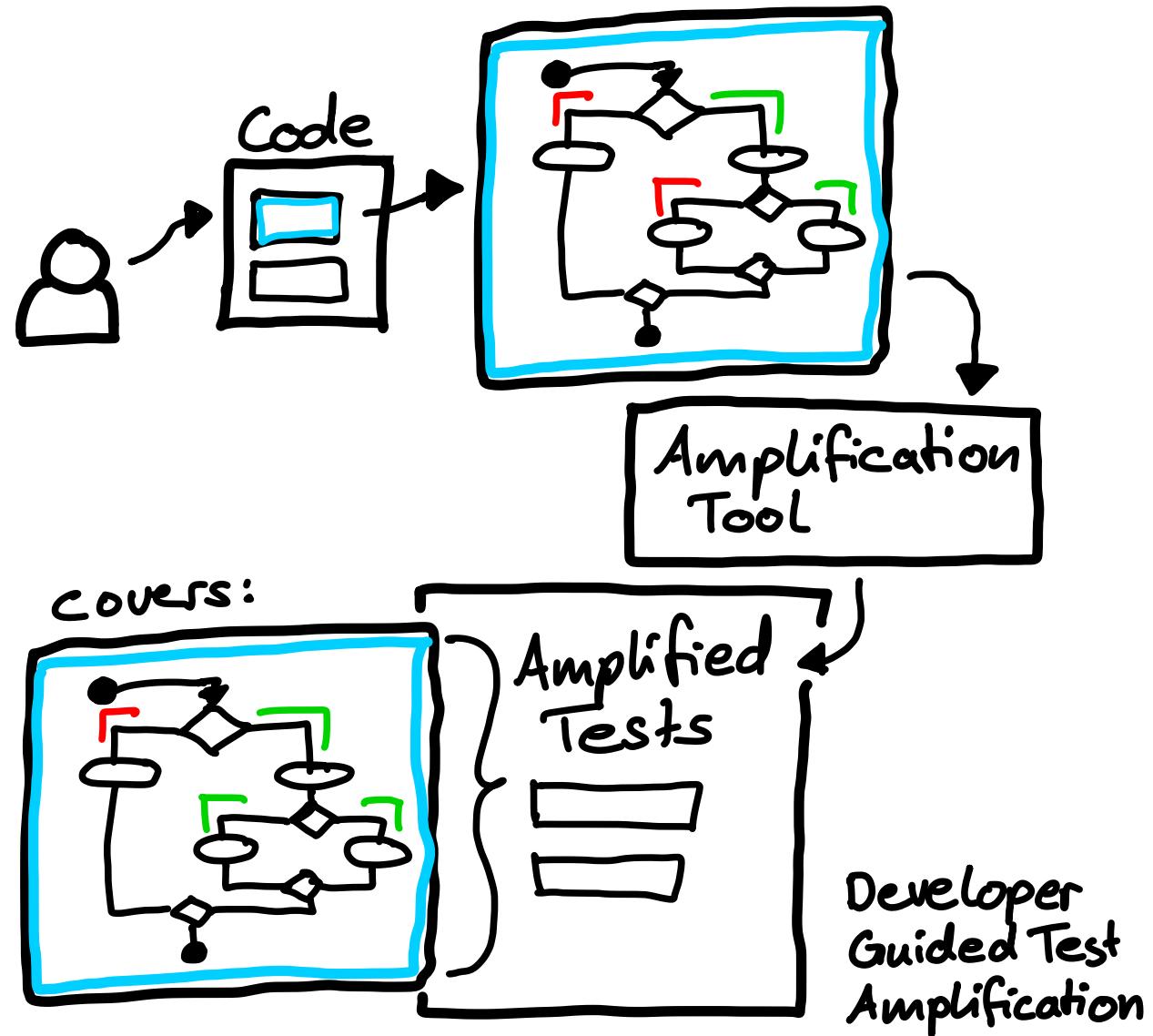


# User Interaction



Open Test Amplification:  
Looking in all Directions for Tests

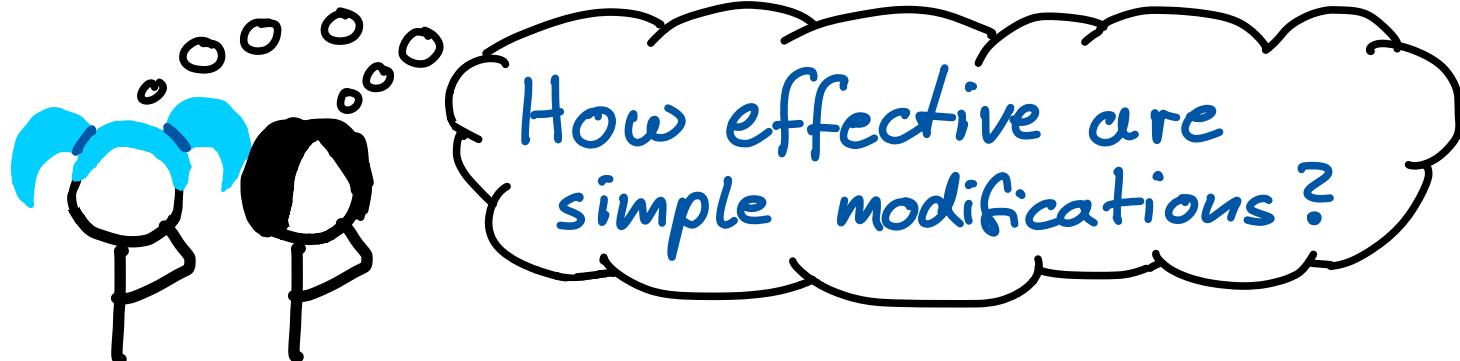
What if we let the developer  
point us to what they want to test?



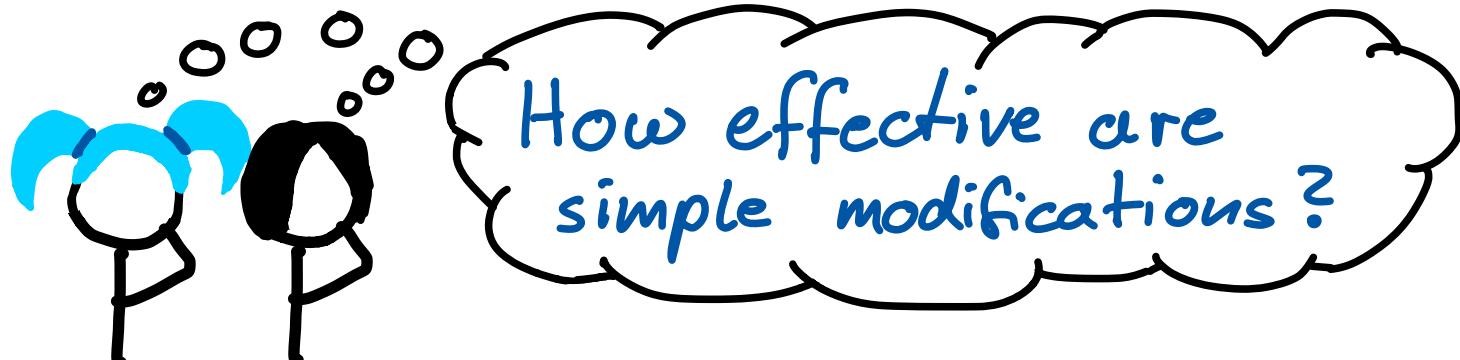
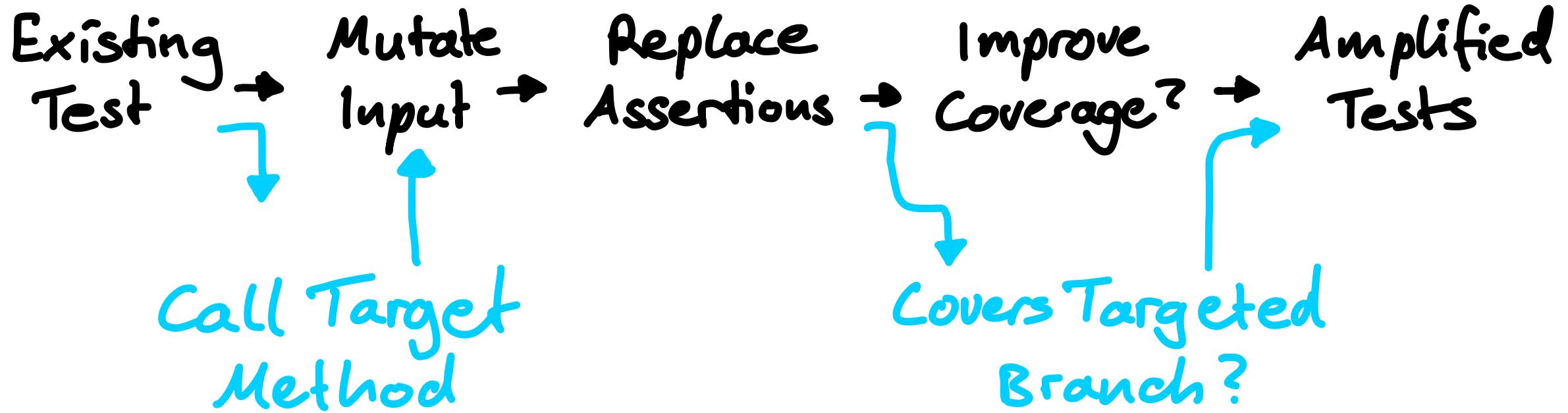
Developer  
Guided Test  
Amplification

# Guided Test Amplification

Existing Test → Mutate Input → Replace Assertions → Improve Coverage? → Amplified Tests



# Guided Test Amplification



# Technical Evaluation

sample 100 branches each from 2 projects

# Technical Evaluation

sample 100 branches each from 2 projects

Sampled branches covered in total

	JavaPoet	Stream-lib
Open	23%	35%
Guided	32%	41%

Main culprit:  
correct initialization of objects under test

to hit targeted branch

# Technical Evaluation

sample 100 branches each from 2 projects

Sampled branches covered in total

	JavaPoet	Stream-lib
Open	23%	35%
Guided	32%	41%

Tests covering target branch per run

	JavaPoet	Stream-lib
Open	24%	45%
Guided	70%	70%

Main culprit:

correct initialization of objects under test

to hit targeted branch

# User Study Evaluation

Generate Tests with Open + Guided Test Amplification with IntelliJ Plugin

12 participants in 4 groups, 2 classes, crossover

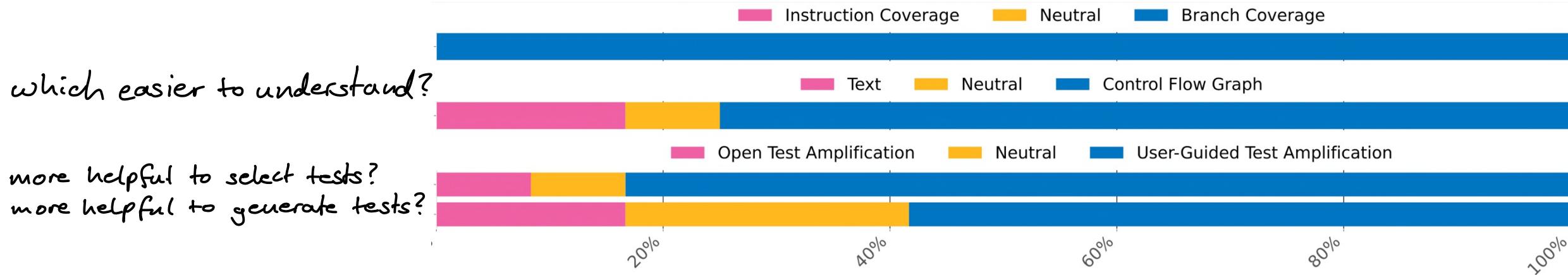
Semi-structured interviews

# User Study Evaluation

Generate Tests with Open + Guided Test Amplification with IntelliJ Plugin

12 participants in 4 groups, 2 classes, crossover

Semi-structured interviews



Guided is not necessarily better...

Guided is not necessarily better...

Trade-offs: Guided vs. Open Test Amplification

Guided is not necessarily better...

## Trade-offs: Guided vs. Open Test Amplification

Fits use case	Writing code & wanting tests for it	Improving test suite
---------------	-------------------------------------	----------------------

Guided is not necessarily better...

## Trade-offs: Guided vs. Open Test Amplification

Fits use case	Writing code & wanting tests for it	Improving test suite
Understand Coverage	In target method in detail	Across whole project

Guided is not necessarily better...

## Trade-offs: Guided vs. Open Test Amplification

Fits use case	Writing code & wanting tests for it	Improving test suite
Understand Coverage	In target method in detail	Across whole project
Expectation of receiving tests	Might disappoint if target branch cannot be covered	Only proposes tests it can provide

# Guided is not necessarily better...

## Trade-offs: Guided vs. Open Test Amplification

Fits use case	Writing code & wanting tests for it	Improving test suite
Understand Coverage	In target method in detail	Across whole project
Expectation of receiving tests	Might disappoint if target branch cannot be covered	Only proposes tests it can provide
Runtime efficiency	More effective at providing tests for targeted method	Larger variety of tests for whole class

# Trade-offs: Guided vs. Open Test Amplification

Fits use case	Writing code & wanting tests for it	Improving test suite
Understand Coverage	In target method in detail	Across whole project
Expectation of receiving tests	Might disappoint if target branch cannot be covered	Only proposes tests it can provide
Runtime efficiency	More effective at providing tests for targeted method	Larger variety of tests for whole class

When to let the Developer Guide: Trade-offs  
Between Open and Guided Test Amplification

Carolin Brandt, Danyao Wang, Andy Zaidman 



Calculator.java x CalculatorTest.java x

Test Cube Test Generation for range()

1 @Test(timeout = 10000)  
 2     public void testSub\_mg12\_assSep300() throws Exception {  
 3         Assert.assertEquals("X is less than 10", new Calculator(3, 2).other());  
 4     }

Add Test To Test Suite Ignore Test Case Next Test Case Previous Test Case Close Amplification Result

1 public String range() {  
 2     if (x < 10) {  
 3         return "X is less than 10";  
 4     } else if (x > 15) {  
 5         return "X is more than 15";  
 6     } else {  
 7         return "X is more than 10 and less than 15";  
 8     }  
 9 }  
 10  
 11 public String SingleIf(){  
 12     if (y > x) {  
 13         return "Y is larger than X";  
 14     }  
 15     return "X is larger than Y";  
 16 }  
 17  
 18 public int uncovered() {  
 19     for (int i = 1; i < x; ++i) {  
 20         --x;  
 21     }  
 22 }

34: True 34: False  
 35: return "X is less than 10"  
 36: True 36: False  
 37: return "X is more than 15"  
 38: 39: return "X is more than 10 and less than 15"

34: True 34: False  
 35: return "X is less than 10"  
 36: True 36: False  
 37: return "X is more than 15"  
 38: 39: return "X is more than 10 and less than 15"

Test Cube found 3 amplified test cases.

Actions ▾

Generate test to cover the selected branch Close

Inspect amplification results

Inspect DSpot terminal output

Event Log

```

graph TD
    1((1)) --> 34[x < 10]
    34 --> 35[35: return "X is less than 10"]
    34 --> 36[x > 15]
    36 --> 37[37: return "X is more than 15"]
    36 --> 39[39: return "X is more than 10 and less than 15"]
    39 --> 2((2))
    39 --> 3((3))
  
```

AttributeTest.java

```
1 package org.jsoup.nodes;
2
3 import ...
4
5 public class AttributeTest {
6     @Test
7     public void html() {
8         Attribute attr = new Attribute("key", "value &");
9         assertEquals(expected: "key=\"value &\"", attr.html());
10    assertEquals(attr.html(), attr.toString());
11 }
12
13     @Test public void testWithSupplementaryCharacterInAttributeValue() {
14     Run... String s = new String(Character.toChars( codePointAt(35361)));
15     Debug... Attribute attr = new Attribute(s, "A" + s + "B");
16     Run with Coverage... assertEquals(expected: s + "\\" + "A" + s + "B\\\"", attr.html());
17     Profile... assertEquals(attr.html(), attr.toString());
18     Modify Run Configuration...
19
20     Amplify 'testWithSupplementaryCharacterInAttributeValue()'
21     @Test public void validatesKeysNotEmpty() {
22     assertThrows(IllegalArgumentException.class, () -> new Attribut
23
24     @Test public void validatesKeysNotEmptyViaSet() {
25     assertThrows(IllegalArgumentException.class, () -> {
26         Attribute attr = new Attribute("One", "Check");
27         attr.setKey(" ");
28     });
29
30
31
32
33 }
```

Test Cube Amplification of 'html()'

Amplified test case 'html\_assSep5'

Input modifications: 0 4

Assert statements added: 1

This test case improves the coverage in these classes/methods/lines:  
(Click on the green links to see these lines within the class)

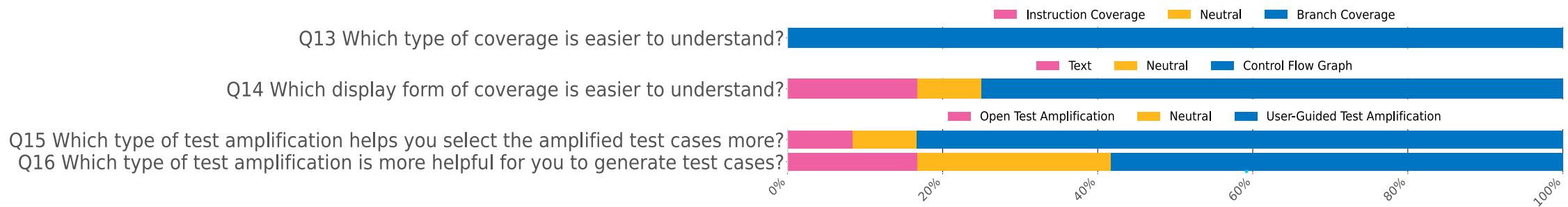
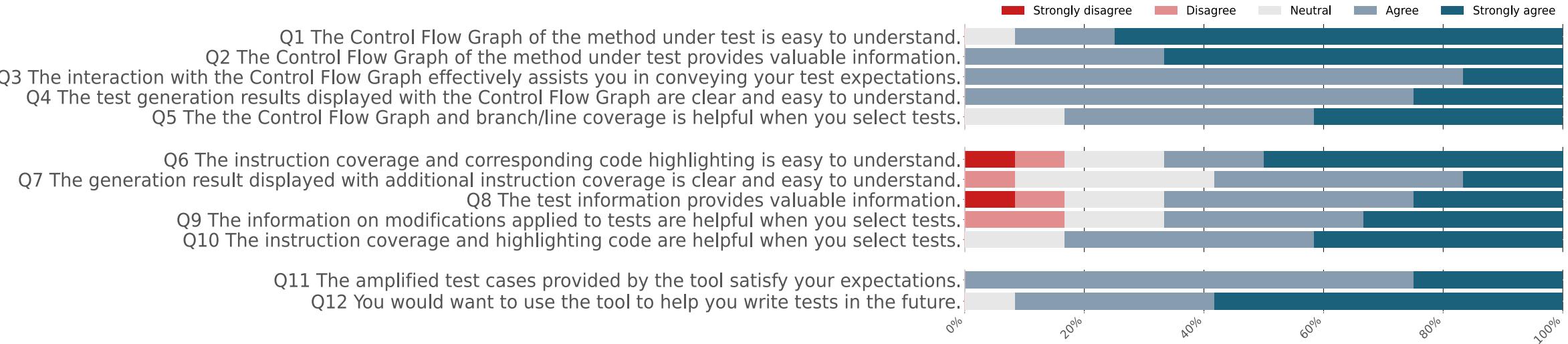
[org.jsoup.nodes.Attribute](#):  
[hashCode](#)  
L. 198 +8 instr.  
L. 199 +12 instr.  
L. 200 +2 instr.

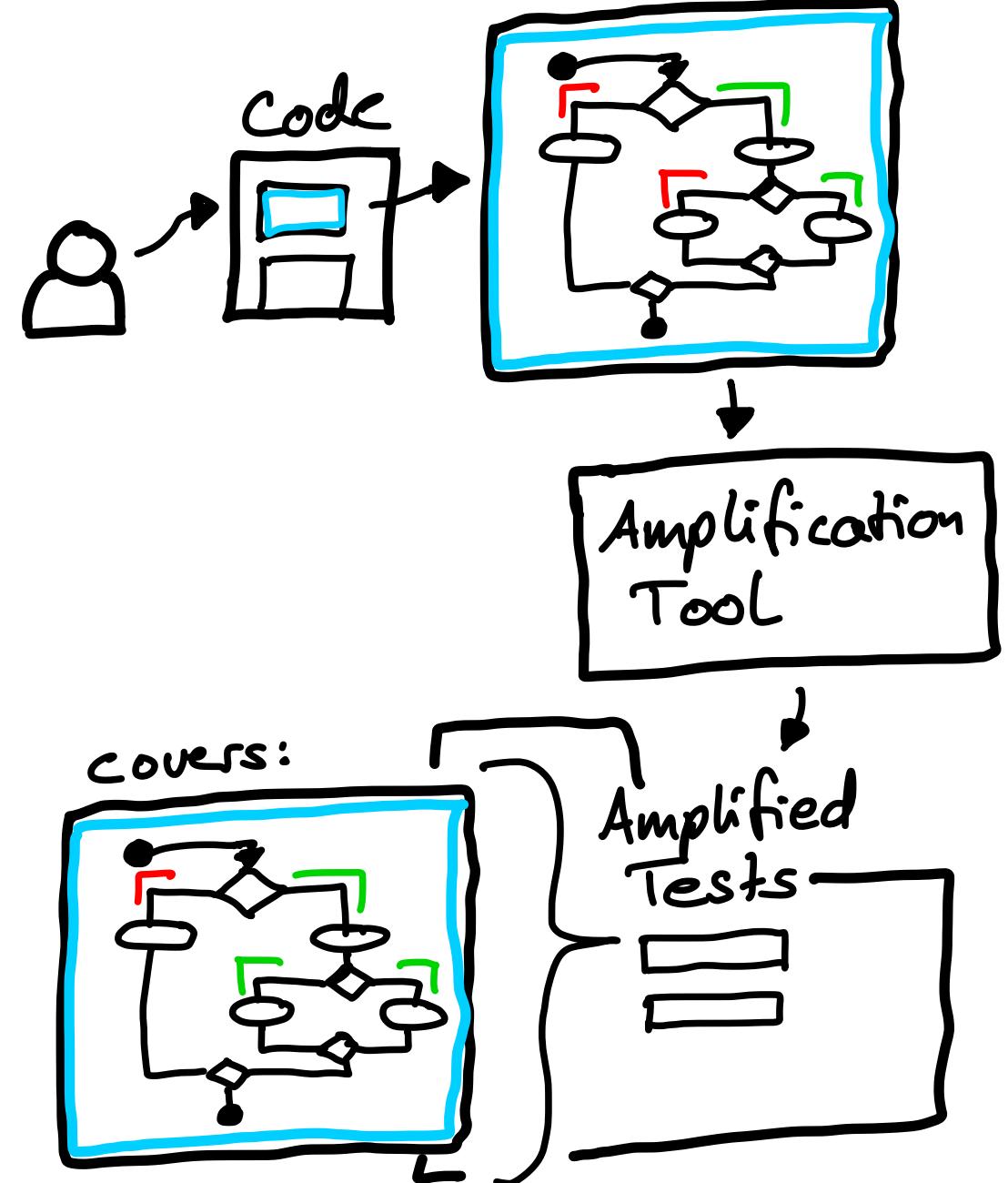
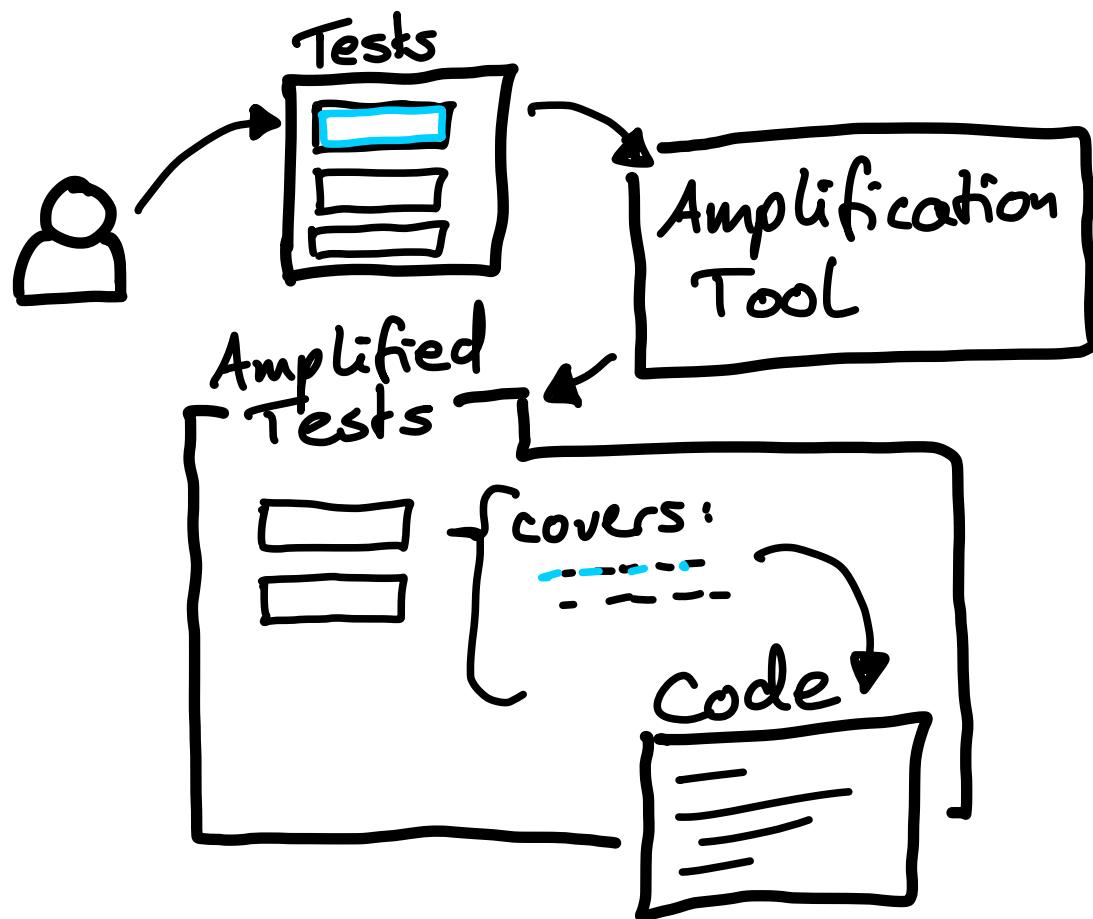
Add Test To Test Suite Ignore Test Case Next Test Case Previous Test Case Close

```
6
7     public void html_assSep5() throws Exception {
8         Attribute attr = new Attribute("key", "value &");
9         Assertions.assertEquals(234891960, ((int) ((Attribute) (attr)).hashCode()));
10
11     @Test
12     public void html_literalMutationString19_assSep92() throws Exception {
13         Attribute attr = new Attribute("key", " Hello\nthere \u00a0 ");
14         Assertions.assertEquals("key=\" Hello\nthere \u00a0 \"", ((Attribute) (attr)).toS
15
16     @Test
17     public void html_mg22_assSep1() throws Exception {
18 }
```

Test Cube found 4 amplified test cases.

Inspect amplification results 3 Inspect DSpot terminal





- 1) To be actually helpful for users, we need to consciously design how users interact with generative tools like GPT, Copilot, ...
- 2) Too much research blindly applies LLMs to inappropriate tasks (tasks that require other skills than (natural) language composition.