Carolin Brandt PhD Student @ Software Engineering Research Group, Delft University of Technology **Advisor: Andy Zaidman**

Developer-Centric Test Amplification



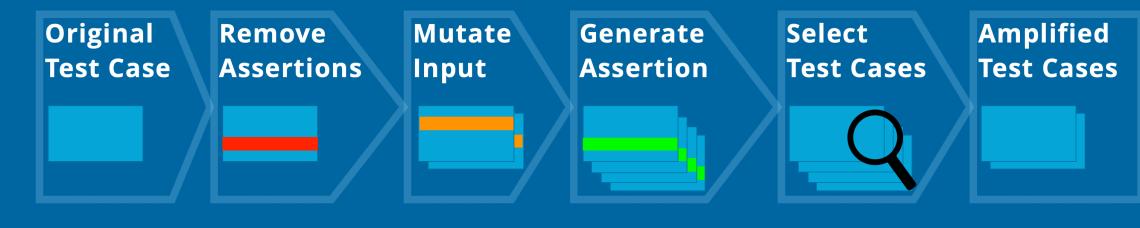
c.e.brandt@tudelft.nl carolin-brandt.de 🔰 @laci_noire

Test Amplification

Generate new tests...

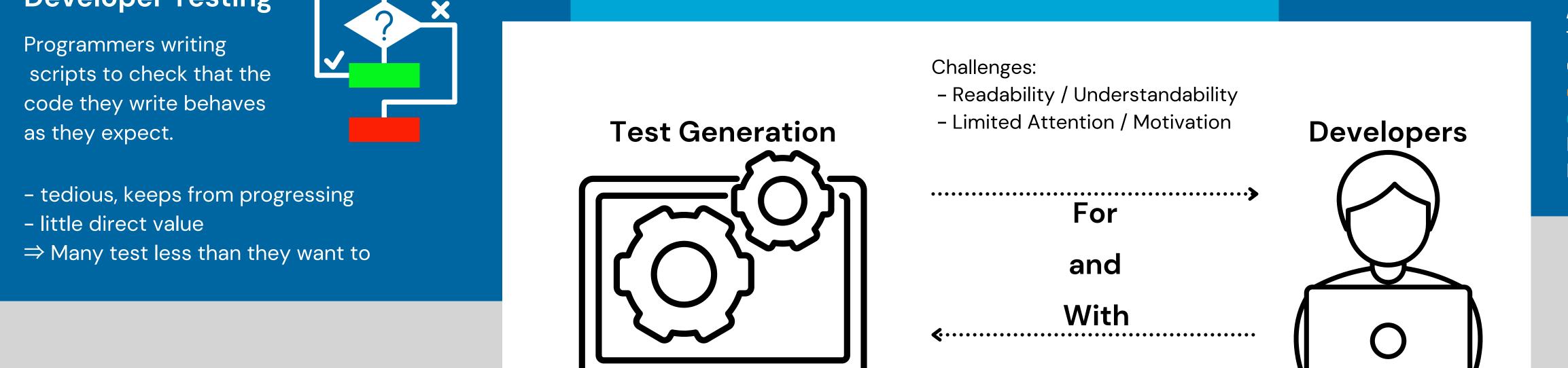
- based on existing tests
- that improve the test suite

@Test public void html() { Attribute attr = new Attribute("key", "value &"); assertEquals("key=\"value &\"", attr.html());



@Test public void html_nbsp() {

Attribute attr = new Attribute("key", "Hello\nthere\$NBSP\$"); assertEquals("key=\"Hello\nthere \"", attr.html());



Amplified test 'html_nbsp' This test improves coverage in these classes/methods/lines: org.jsoup.nodes.Entities: escape L. 197 +3 instr.

L. 198 +5 instr.

Interaction is Central

of test generation

- Prioritization

- Oracle Problem

Chances: Tackle the hard aspects

We have to generate powerful, high-quality tests,

but it is crucial to convey this power and quality to the developer to convince them to include the tests.

Communicate Impact on Test Suite Quality

TestImpactGraph

| Test | com.squareup.javapoet.CodeWriter | com.squareup.javapoet |
|--|---|-----------------------|
| public void modifyAnnotations_5() throws Exception { | public CodeWriter emit(CodeBlock codeBlock, boolean ensureTrailingNewline) throws IOException { | private void emitLi |
| ParameterSpec.Builder builder = ParameterSpec.builder(int.class, "foo").addAnnotation(Override.class).addAnnotation(SuppressWarnings.class); 📀 | int a = 0; | if (o instanceof |
| AnnotationSpec = builder.annotations.remove(1); | ClassName deferredTypeName = null; // used by "import static" logic | TypeSpec typeSp |
| Truth.assertThat(((AnnotationSpec) (annotationSpec)).toString()).isEqualTo("%java.lang.SuppressWarnings"); | ListIterator | typeSpec.emit(t |
| | while (partIterator.hasNext()) { | } else if (o inst |
| | String part = partIterator.next(); | AnnotationSpec |
| com.squareup.javapoet.AnnotationSpec | switch (part) { | annotationSpec. |
| <pre>@Override public String() {</pre> | case "\$L": | } else if (o inst |
| StringBuilder out = new StringBuilder(); | <pre>emitLiteral(codeBlock.args.get(a++));</pre> | CodeBlock codeB |
| try (| break; | emit(codeBlock) |
| CodeWriter codeWriter = new CodeWriter(out); | | } else { |
| <pre>codeWriter.emit("\$L", this);</pre> | case "\$N": | emitAndIndent(S |
| return out.toString(); | <pre>emitAndIndent((String) codeBlock.args.get(a++));</pre> | } |
| <pre>> catch (IOException e) {</pre> | break; | |
| throw new AssertionError(); 🕑 | | |
| , | case "\$S": | com.squareup.javapoet |
| | <pre>String string = (String) codeBlock.args.get(a++);</pre> | @Override CodeWrite: |
| | | boolean charsEmit |

Impact Descriptions

This test checks that equals() returns false for an Attribute and null. This test checks that addElement() throws an IllegalArgumentException if the given element is null.

Developer-Involvement Information to Start Amplifying **Direct Amplification** Can you give us an example on Generate a test how you would initialize Node? [trey for this branch Node node = new Node("?");

Edits to Amplified Tests

assertEquals("What should come out here?", attr.html());

Vision: Automatic & Integrated Test Recommendation

Developer-Driven Test Adequacy Criteria

Project- & Developer-Adapted Generation

Hey! 🙄 We generated a new test for you that checks that your new method proccess() returns null if null is passed. Inspect new test Include new test This is not important to test

This poster is based on the first study of my PhD [1], the TestImpactGraph is described in [2].

[1] Brandt, C., & Zaidman, A. (2021). Developer-Centric Test Amplification: The Interplay Between Automatic Generation and Human Exploration. arXiv:2108.12249. Link -> [2] Brandt, C., & Zaidman, A. (2021). How Does This New Developer Test Fit In?: A Visualization to Understand Amplified Test Cases. Under Submission.

