



Carolyn Brandt

+

SE Forschung an der TU Delft

+



Automatische Testgenerierung für Entwickler und Vieles Mehr

Worum geht's heute?

VORSTELLUNG & KENNENLERNEN

Interaktiv

TESTS GENERIEREN FÜR & MIT ENTWICKLERN

+ Themenübersicht Software Engineering Research Group

MASTERARBEIT & PROMOTION IN DELFT

+ Was plant ihr nach dem Master?



Worum geht's heute?

VORSTELLUNG & KENNENLERNEN

Interaktiv

TESTS GENERIEREN FÜR & MIT ENTWICKLERN

Fragerunde

+ Themenübersicht Software Engineering Research Group

Fragerunde

MASTERARBEIT & PROMOTION IN DELFT

+ Was plant ihr nach dem Master?

Fragerunde



Carolyn Brandt

Meine Laufbahn



**INFORMATIK:
GAMES-
ENGINEERING**

Bachelor TUM



**ELITE GRADUATE
PROGRAM:
SOFTWARE
ENGINEERING**

12. Jahrgang
Praktikum: Oracle Labs



**MASTERARBEIT:
TU DELFT /
PRETSCHNER &
KNAPP**

Automatische Analyse von
Build Logs



**PHD @ SERG TU
DELFT**

Visualizing Test
Amplification

TestShift Project,
VICI Grant,
NWO (~ DFG in NL)



Kennenlernen

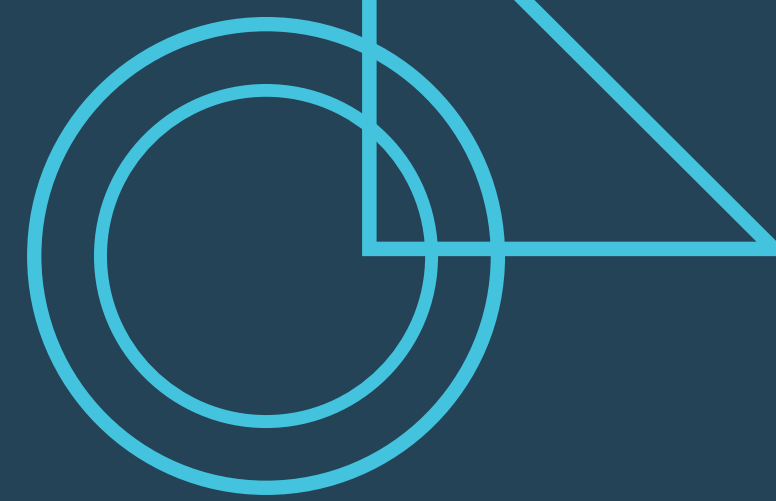
<https://ttpoll.eu/p/ringvorlesung>



The background is a solid dark blue. It is decorated with several abstract geometric elements: a teal vertical bar in the upper left, a teal horizontal bar in the lower left, a teal vertical bar in the lower center, and various light blue outlines of circles and triangles scattered across the left side. The main text is in a bold, white, sans-serif font, positioned on the right side of the image.

Test Amplification for & with Developers

Automatische Test Generierung



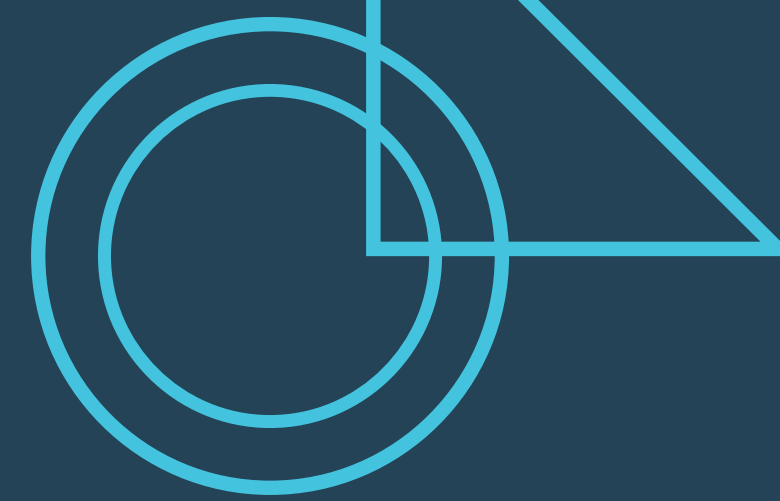
Große Hürden

Orakelproblem

Was muss getestet werden?



Automatische Test Generierung



Große Hürden

Orakelproblem

Was muss getestet werden?

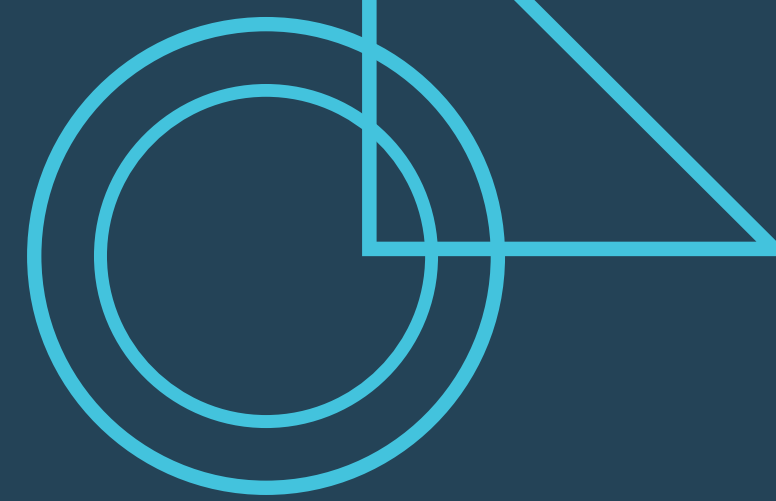
Tests sind für Entwickler

Dokumentation

Wo ist schief gelaufen?



Automatische Test Generierung



Große Hürden

Orakelproblem

Was muss getestet werden?

Tests sind für Entwickler

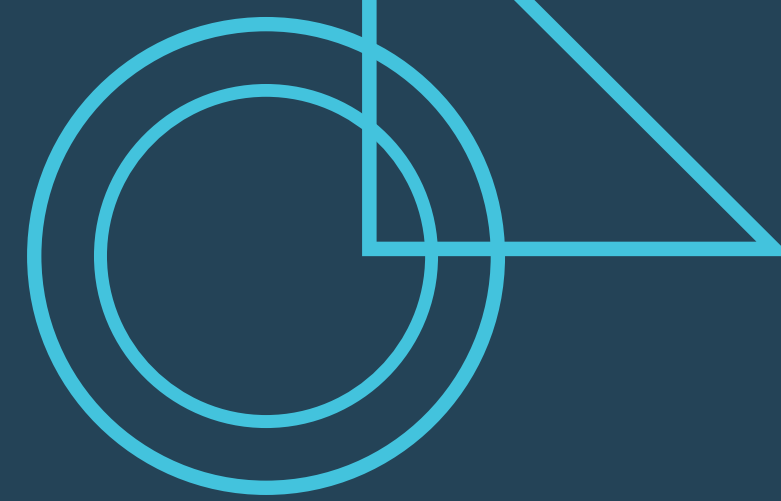
Dokumentation

Wo ists schief gelaufen?

Generieren für Entwickler



Automatische Test Generierung



Große Hürden

Orakelproblem

Was muss getestet werden?

Tests sind für Entwickler

Dokumentation

Wo ist schief gelaufen?

Generieren für und mit Entwicklern



Test Amplification



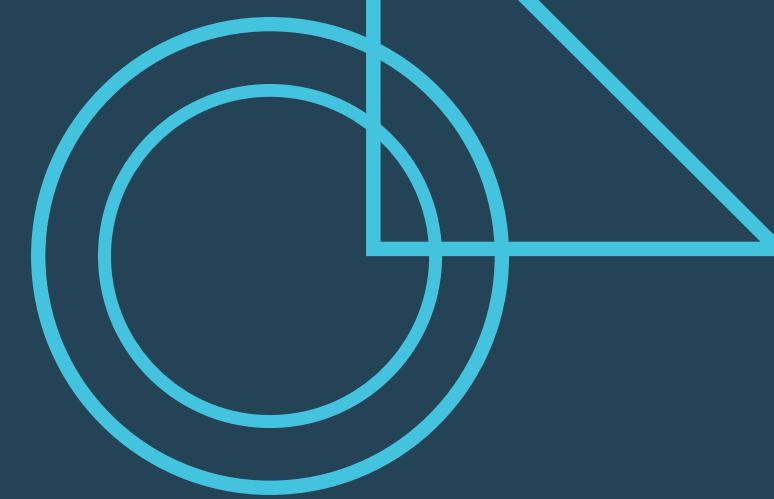
Neue Tests generieren...

- durch Mutation existierender
- die die Test Suite verbessern

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



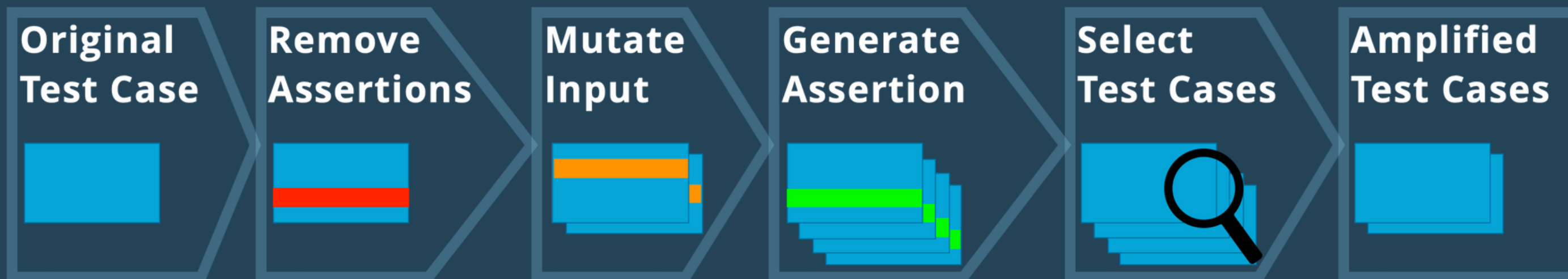
Test Amplification



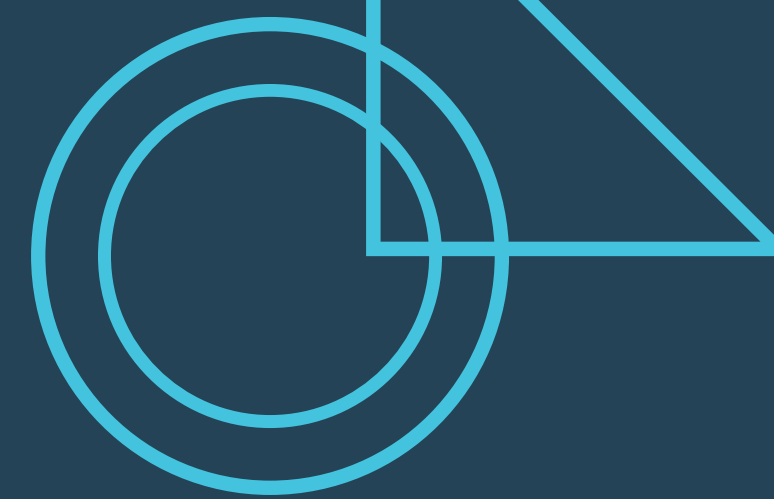
Neue Tests generieren...

- durch Mutation existierender
- die die Test Suite verbessern

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



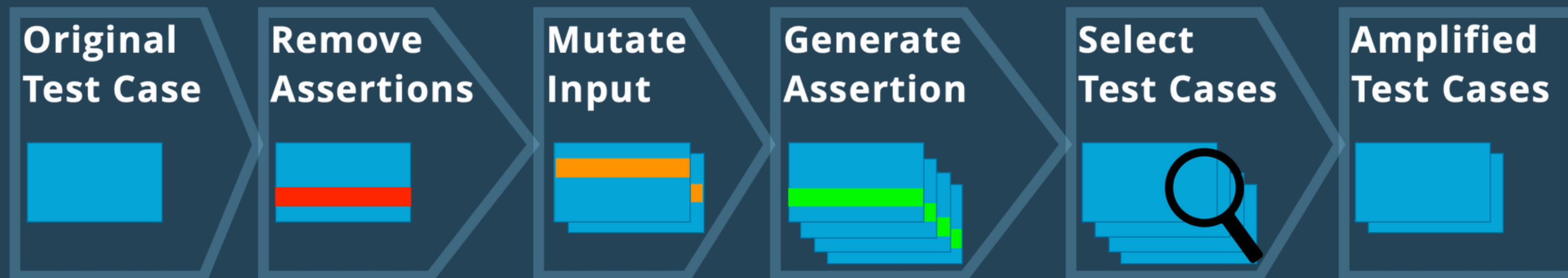
Test Amplification



Neue Tests generieren...

- durch Mutation existierender
- die die Test Suite verbessern

```
@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&nbsp;\"", attr.html());  
}
```

Amplified test 'html_ '
This test improves coverage in these
classes/methods/lines:
org.jsoup.nodes.Entities:
escape
L. 197 +3 instr.
L. 198 +5 instr.

Test Cube



```
AttributeTest.java x
1 package org.jsoup.nodes;
2
3 import ...
4
5
6
7
8
9 public class AttributeTest {
10     @Test
11     public void html() {
12         Attribute attr = new Attribute("key", "value &");
13         assertEquals( expected: "key=\"value &amp;\"", attr.html());
14         assertEquals(attr.html(), attr.toString());
15     }
16 }
```

Test Cube



```
AttributeTest.java x
1 package org.jsoup.nodes;
2
3 import ...
4
5
6
7
8
9 public class AttributeTest {
10     @Test
11     public void html() {
12         Attribute attr = new Attribute("key", "value &");
13         assertEquals( expected: "key=\"value &amp;\"", attr.html());
14         assertEquals(attr.html(), attr.toString());
15     }
16
17     @Test public void testWithSupplementaryCharacterInAttributeKeyAndValue() {
18         String s = new String(Character.toChars(0x1F600));
19         Attribute attr = new Attribute(s, "A" + s + "B");
20         assertEquals( expected: s + "\"A" + s + "B\"", attr.html());
21         assertEquals(attr.html(), attr.toString());
22     }
23
24     @Test public void validatesKeysNotEmpty() {
25         assertThrows(IllegalArgumentException.class, () -> new Attribute(" ", "A"));
26     }
27
28     @Test public void validatesKeysNotEmptyViaSet() {
29         assertThrows(IllegalArgumentException.class, () -> {
30             Attribute attr = new Attribute("One", "Check");
31             attr.setKey(" ");
32         });
33     }
34 }
```

1

2

Run...
Debug...
Run with Coverage...
Profile...
Modify Run Configuration...

Amplify 'testWithSupplementaryCharacterInAttributeKeyAndValue()'

Test Cube



```
AttributeTest.java x
1 package org.jsoup.nodes;
2
3 import ...
4
5
6
7
8
9 public class AttributeTest {
10     @Test
11     public void html() {
12         Attribute attr = new Attribute("key", "value &");
13         assertEquals( expected: "key=\"value &amp;\"", attr.html());
14         assertEquals(attr.html(), attr.toString());
15     }
16
17     @Test public void testWithSupplementaryCharacterInAttributeKeyAndValue() {
18         String s = new String(Character.toChars(0x1F600));
19         Attribute attr = new Attribute(s, "A" + s + "B");
20         assertEquals( expected: s + "\"A" + s + "B\"", attr.html());
21         assertEquals(attr.html(), attr.toString());
22     }
23
24     @Test public void validatesKeysNotEmpty() {
25         assertThrows(IllegalArgumentException.class, () -> new Attribute("", "A"));
26     }
27
28     @Test public void validatesKeysNotEmptyViaSet() {
29         assertThrows(IllegalArgumentException.class, () -> {
30             Attribute attr = new Attribute("One", "Check");
31             attr.setKey(" ");
32         });
33     }
34 }
```

1 Amplify 'testWithSupplementaryCharacterInAttributeKeyAndValue()'

Test Cube found 4 amplified test cases.

3 Inspect amplification results Inspect DSpot terminal output



A stylized logo consisting of a light blue triangle and two overlapping light blue circles, set against a dark blue background. The triangle is positioned in the upper left, and the circles are in the lower right.

Interviews



Amplified Test Cases

Exploration Tool

Information

Entwickler



INTERAKTION



Test Amplification

Kommende Projekte

TestImpactGraph

```
Test
public void modifyAnnotations_3() throws Exception {
    ParameterSpec.Builder builder = ParameterSpec.builder(int.class, "foo").addAnnotation(Override.class).addAnnotation(SuppressWarnings.class);
    AnnotationSpec annotationSpec = builder.annotations().remove();
    Truth.assertThat(((AnnotationSpec) annotationSpec).toString()).isEqualTo("java.lang.SuppressWarnings");
}

com.squareup.javapoet.AnnotationSpec
@Override public String toString() {
    ToStringHelper out = new ToStringHelper();
    try {
        CodeWriter codeWriter = new CodeWriter(out);
        codeWriter.emit("S", this);
        return out.toString();
    } catch (IOException e) {
        throw new AssertionError();
    }
}

com.squareup.javapoet.CodeWriter
public CodeWriter emit(CodeBlock codeBlock, boolean ensureTrailingNewline) throws IOException {
    int a = 0;
    ClassNode deferredTypeName = null; // used by "import static" logic
    ListIterator
    while (partIterator.hasNext()) {
        String part = partIterator.next();
        switch (part) {
            case "L":
                emitLiteral(codeBlock.args.get(a++));
                break;
            case "S":
                emitAndIndent((String) codeBlock.args.get(a++));
                break;
            case "SS":
                String string = (String) codeBlock.args.get(a++);
                // emitAndIndent(string);
                break;
        }
    }
}

com.squareup.javapoet.CodeWriter
private void emitLiteral(Object o) throws IOException {
    if (!o instanceof TypeSpec) {
        TypeSpec typeSpec = (TypeSpec) o;
        typeSpec.emit(this, null, Collections.emptySet());
    } else if (o instanceof AnnotationSpec) {
        AnnotationSpec annotationSpec = (AnnotationSpec) o;
        annotationSpec.emit(this, true);
    } else if (o instanceof CodeBlock) {
        CodeBlock codeBlock = (CodeBlock) o;
        emit(codeBlock);
    } else {
        emitAndIndent(String.valueOf(o));
    }
}

com.squareup.javapoet.ClassNode
@Override CodeWriter emit(CodeWriter out) throws IOException {
    boolean suppressed = false;
}
```

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.

Kommende Projekte

TestImpactGraph

```
Test
public void modifyAnnotations() throws Exception {
    ParameterSpec.Builder builder = ParameterSpec.builder(int.class, "foo").addAnnotation(Override.class).addAnnotation(SuppressWarnings.class);
    AnnotationSpec annotationSpec = builder.annotationSpec().addAnnotation(Override.class).addAnnotation(SuppressWarnings.class).build();
    Truth.assertThat(annotationSpec.toString()).isEqualTo("java.lang.SuppressWarnings");
}

com.squareup.javapoet.AnnotationSpec
@Override public String toString() {
    ToStringBuilder builder = new ToStringBuilder(this);
    try {
        CodeWriter codeWriter = new CodeWriter(out);
        codeWriter.emit("S", this);
        return builder.toString();
    } catch (IOException e) {
        throw new AssertionError();
    }
}

com.squareup.javapoet.CodeWriter
public CodeWriter emit(CodeBlock codeBlock, boolean ensureTrailingNewline) throws IOException {
    int a = 0;
    ClassNode deferredTypeNode = null; // used by "import static" logic
    ListIterator while (getIterator().hasNext()) {
        String part = getIterator().next();
        switch (part) {
            case "S":
                emitLiteral(codeBlock.args.get(a++));
                break;
            case "S2":
                emitAndIndent((String) codeBlock.args.get(a++));
                break;
            case "S3":
                String string = (String) codeBlock.args.get(a++);
                // ...
        }
    }
}

com.squareup.javapoet.CodeWriter
private void emitLiteral(Object o) throws IOException {
    if (o instanceof TypeSpec) {
        TypeSpec typeSpec = (TypeSpec) o;
        typeSpec.emit(this, null, Collections.emptyList());
    } else if (o instanceof AnnotationSpec) {
        AnnotationSpec annotationSpec = (AnnotationSpec) o;
        annotationSpec.emit(this, true);
    } else if (o instanceof CodeBlock) {
        CodeBlock codeBlock = (CodeBlock) o;
        emit(codeBlock);
    } else {
        emitAndAddIndent(String.valueOf(o));
    }
}

com.squareup.javapoet.ClassNode
@Override CodeWriter emit(CodeWriter out) throws IOException {
    boolean suppressed = false;
}
```

Direct Amplification



Generate a test
for this branch

Information to Start Amplifying

Can you give us an example on
how you would initialize Node?

```
Node node = new Node("?");
```

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.

Edits to Amplified Tests

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

Fragenrunde!

Wann wäre solch eine Testgenrierung für euch hilfreich?

Was wäre euch wichtig in vorgeschlagenen Tests?
Was wollt ihr über die Tests wissen?

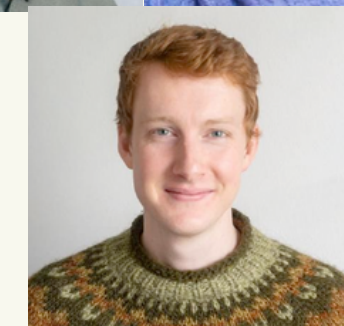
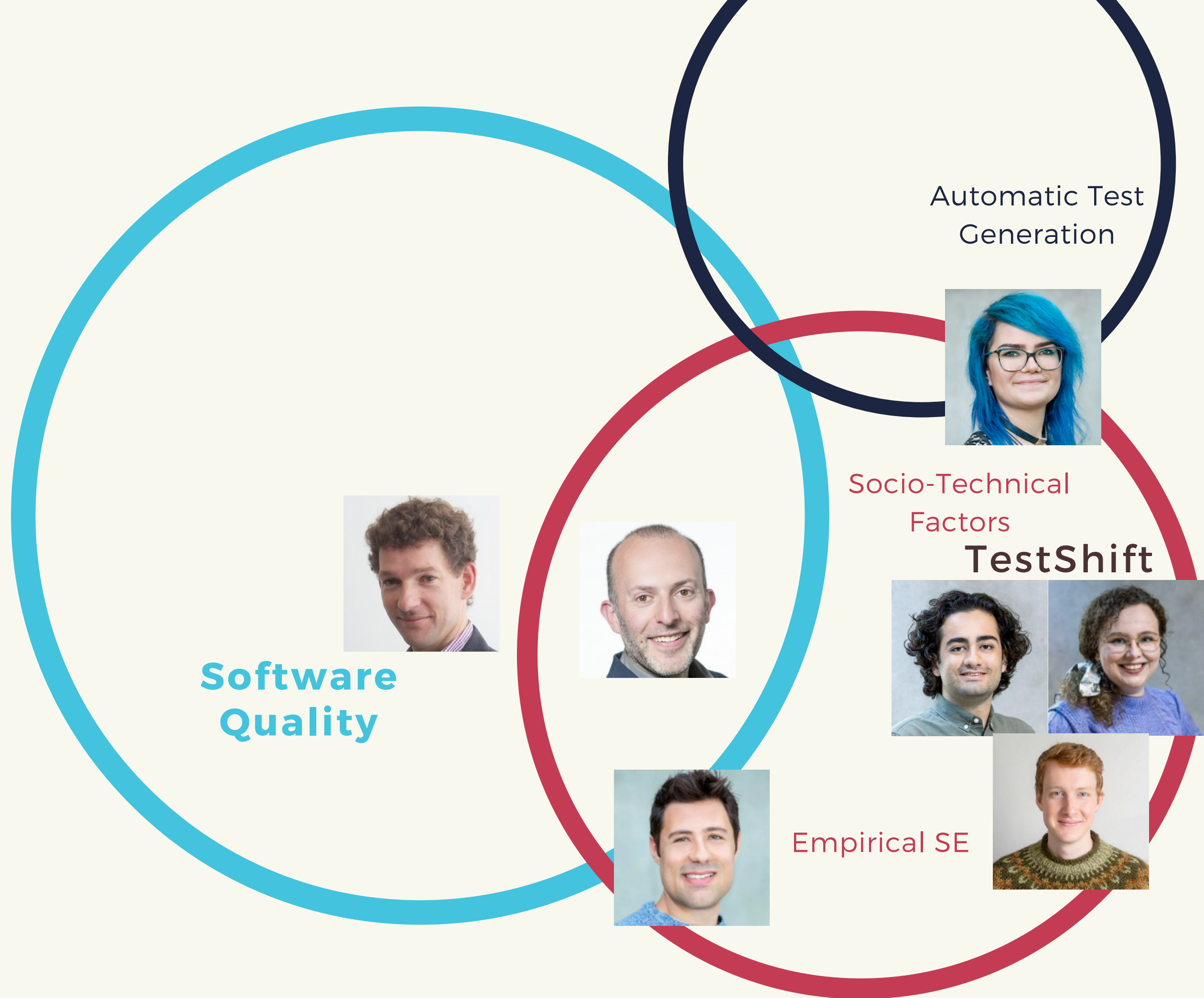
Welche anderen Teile der Software-Entwicklung sollten wir auch automatisieren?



SERG TU Delft: **Themenübersicht**

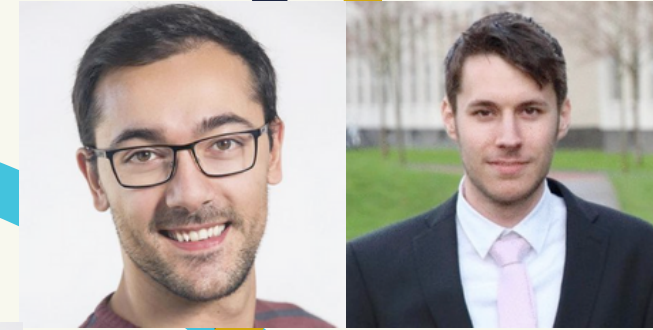


**Software
Quality**



Computational Intelligence
for SE

COSMOS



Automatic Test
Generation



Socio-Technical
Factors
TestShift



Empirical SE



**Software
Quality**



Computational Intelligence
for SE

COSMOS

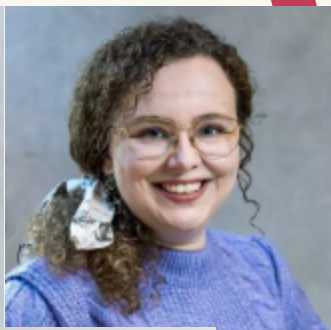
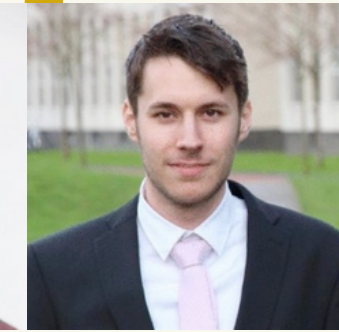
Automatic Test
Generation

Socio-Technical
Factors
TestShift

Empirical SE

**Software
Quality**

DevOps &
CI/CD



AI4Fintech



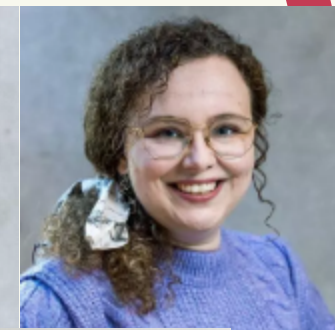
Computational Intelligence
for SE



Automatic Test
Generation



Socio-Technical
Factors
TestShift



Empirical SE



COSMOS



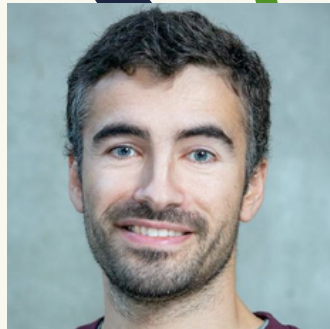
DevOps &
CI/CD



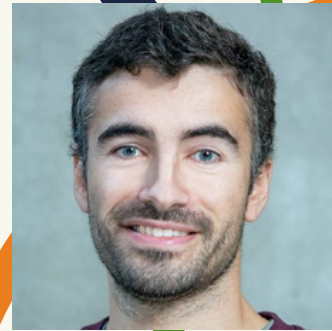
**Software
Quality**



Green
SE



AI4Fintech



SE 4 AI
AI 4 SE



Green
SE

Computational Intelligence
for SE



COSMOS



DevOps &
CI/CD



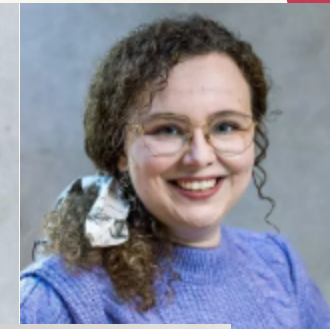
Software
Quality



Automatic Test
Generation



Socio-Technical
Factors
TestShift



Empirical SE



AI4Fintech



Computational Intelligence
for SE



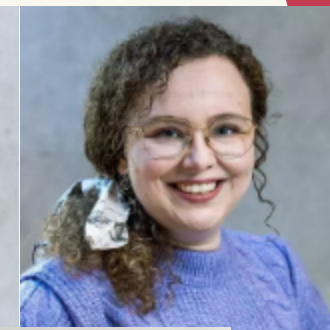
Automatic Test
Generation

COSMOS



Socio-Technical
Factors
TestShift

DevOps &
CI/CD



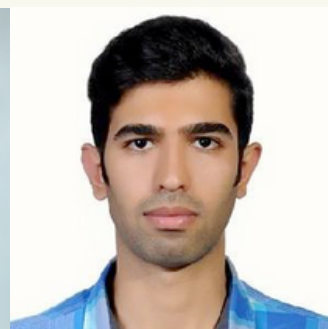
Empirical SE



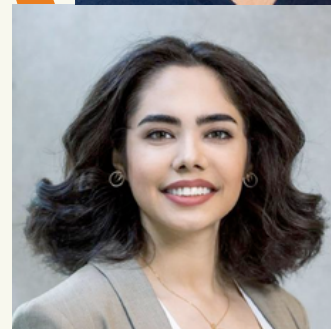
Software Quality



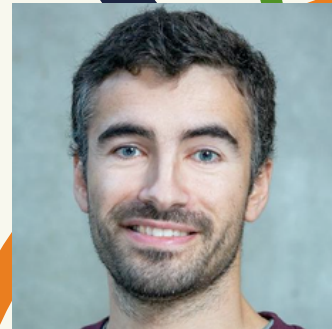
Software Analytics
FASTEN



SE 4 AI
AI 4 SE



Green
SE



AI4Fintech



Computational Intelligence
for SE



Automatic Test
Generation



Socio-Technical
Factors
TestShift



Empirical SE



COSMOS



DevOps &
CI/CD



Software Quality



Concurrent &
Distributed Systems



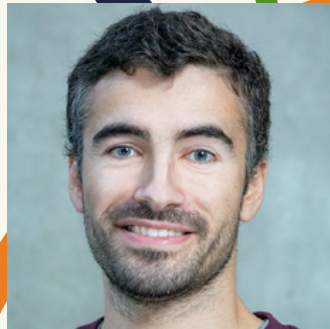
Software Analytics
FASTEN



SE 4 AI
AI 4 SE



Green
SE



Fragenrunde!

Seht ihr Überschneidungen
zu euren Bachelorarbeiten / geplanten Masterarbeiten
oder eurer Forschung?

Zu welchen Bereichen wollt ihr mehr hören?



Masterarbeit in Delft

Normal: 9 Monate / 30 Wochen

Paper möglich

Breite Themenauswahl, ständig
wechselnd
→ schreibt mich an

Nicht nach offiziellem TUD Prozess
Gast (Kaffee!!)

Alles auf Englisch

Promotion in Delft

4 Jahre

Wenig / keine Lehre
→ wenig Arbeit mit Studenten

! FUNDING

EU Projekt vs.
NWO vs.
Industrie vs.
TU Delft

Projektarbeit
Industriekontakte
Sponsor-wünsche
Themenoffenheit

akademische Ausrichtung

Softskill - Kurse

Reisen

Fragenrunde!

Was ist euer Plan nach dem Master?
Könnt ihr euch vorstellen zu Promovieren?

Möchtet ihr ins Ausland gehen?

Was ist bei Masterarbeit + Promotion in München /
Augsburg anders als in Delft?



Schreibt mir! :)

Fragen, Ideen,
Gedanken nach Delft /
NL zu kommen

<https://se.ewi.tudelft.nl/>

<https://carolin-brandt.de>

c.e.brandt@tudelft.nl

Twitter: @laci_noire