

# Modeling Readability to Improve Unit Tests

Ermira Daka, José Campos, and Gordon Fraser

University of Sheffield

Jonathan Dorn and Westley Weimer

University of Virginia

ESEC/FSE

2015

“[Developers] read tests [...] 77% of the total time they spend in them”

Moritz Beller, Georgios Gousios, Annibale Panichella, and Andy Zaidman. When, How, and Why Developers (Do Not) Test in Their IDEs. FSE 2015

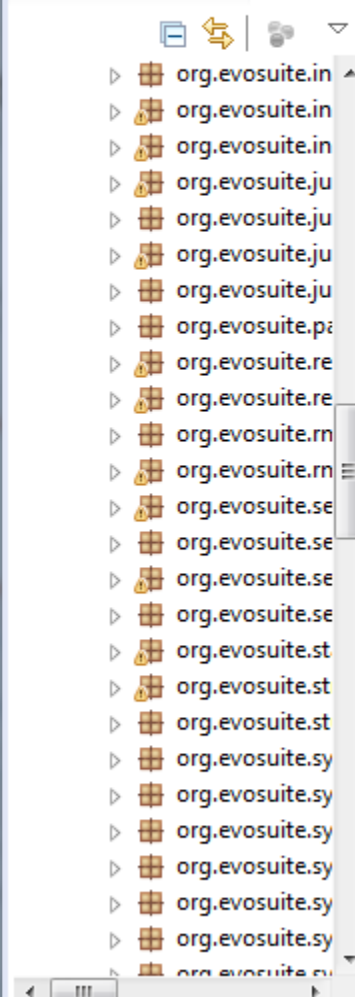


Quick Access

Java EE

Debug

Project Expl...



TestIDNaming.ja...

TestMethodTrac...

TestNameGenera...

VariableNames...

TestInputCover...

TestSuiteWrite...

```
20 * Copyright (C) 2010-2015 Gordon Fraser, Andrea Arcuri and EvoSuite
21 package org.evosuite.idenaming;
22 import com.examples.with.different.packagename.Calculator;
38
39 public class TestIDNaming extends SystemTest {
40
41     @Test
42     public void testIDNamingOn() {
43         EvoSuite evosuite = new EvoSuite();
44
45         String targetClass = SnippetInputContainer.class.getCanonicalName();
46
47         Properties.TARGET_CLASS = targetClass;
48         Properties.ID_NAMING = true;
49         Properties.JUNIT_TESTS = true;
50         StringBuilder analysisCriteria = new StringBuilder();
51         analysisCriteria.append(Properties.Criterion.METHOD); analysisCriteria.append(",");
52         analysisCriteria.append(Properties.Criterion.OUTPUT); analysisCriteria.append(",");
53         analysisCriteria.append(Properties.Criterion.BRANCH);
54         Properties.ANALYSIS_CRITERIA = analysisCriteria.toString();
55
56         Properties.CRITERION = new Properties.Criterion[3];
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit

Finished after 70.833 seconds

Runs: 2/2

Errors: 0

Failures: 1

testIDNamingOff (27.586 s)

testIDNamingOn (39.898 s)

Failure Trace

Writable

Smart Insert

68 : 1

```
public void testFactory_weeksBetween_RInstant() {  
    DateTime start = new DateTime(2006, 6, 9, 12, 0, 0, 0, PARIS);  
    DateTime end1 = new DateTime(2006, 6, 30, 12, 0, 0, 0, PARIS);  
    DateTime end2 = new DateTime(2006, 7, 21, 12, 0, 0, 0, PARIS);  
  
    assertEquals(3, Weeks.weeksBetween(start, end1).getWeeks());  
    assertEquals(0, Weeks.weeksBetween(start, start).getWeeks());  
    assertEquals(0, Weeks.weeksBetween(end1, end1).getWeeks());  
    assertEquals(-3, Weeks.weeksBetween(end1, start).getWeeks());  
    assertEquals(6, Weeks.weeksBetween(start, end2).getWeeks());  
}
```

```
public void test16() throws Throwable {  
    Weeks weeks0 = Weeks.ZERO;  
    Weeks weeks1 = weeks0.minus(3);  
    Instant instant0 = new Instant((long) 3);  
    DateTime dateTime0 = instant0.toDateTime();  
    LocalDate localDate0 = dateTime0.toLocalDate();  
    LocalDate localDate1 = localDate0.minus(weeks1);  
    Weeks weeks2 = Weeks.weeksBetween((ReadablePartial) localDate1, (ReadablePartial) localDate0);  
    assertEquals(-3, weeks2.getWeeks());  
}
```

# What is test case readability?

- “is the ease with which a written text can be understood by a reader.”
- “the state or quality of being readable.”
- “a quality of writing (print or handwriting) that can be easily read.”

```
public void test0() throws Throwable {  
    Charset charset0 = Charset.defaultCharset();  
    Locale locale0 = Locale.CHINESE;  
    String string0 = charset0.displayName(locale0);  
    MessageDigestHashFunction messageDigestHashFunction0 = null;  
    try {  
        messageDigestHashFunction0 = new MessageDigestHashFunction("UTF-8", "UTF-8");  
        fail("Expecting exception: AssertionError");  
    } catch (AssertionError e) {  
        //  
        // java.security.NoSuchAlgorithmException: UTF-8 MessageDigest not available  
        //  
    }  
}
```

# Hypothesis

*Improving readability of test cases leads to improved software quality:*


- *Less human effort*
- *Easy understandable test cases*
- *Easy maintainable test cases*
- *Easy maintainable software*

# How can we measure readability on unit tests?

```
public void testFactory_weeksBetween_RInstant() {  
    DateTime start = new DateTime(2006, 6, 9, 12, 0, 0, 0, PARIS);  
    DateTime end1 = new DateTime(2006, 6, 30, 12, 0, 0, 0, PARIS);  
    DateTime end2 = new DateTime(2006, 7, 21, 12, 0, 0, 0, PARIS);  
  
    assertEquals(3, Weeks.weeksBetween(start, end1).getWeeks());  
    assertEquals(0, Weeks.weeksBetween(start, start).getWeeks());  
    assertEquals(0, Weeks.weeksBetween(end1, end1).getWeeks());  
    assertEquals(-3, Weeks.weeksBetween(end1, start).getWeeks());  
    assertEquals(6, Weeks.weeksBetween(start, end2).getWeeks());  
}
```

?

```
public void test16() throws Throwable {  
    Weeks weeks0 = Weeks.ZERO;  
    Weeks weeks1 = weeks0.minus(3);  
    Instant instant0 = new Instant((long) 3);  
    DateTime dateTime0 = instant0.toDateTime();  
    LocalDate localDate0 = dateTime0.toLocalDate();  
    LocalDate localDate1 = localDate0.minus(weeks1);  
    Weeks weeks2 = Weeks.weeksBetween((ReadablePartial) localDate1, (ReadablePartial) localDate0);  
    assertEquals(-3, weeks2.getWeeks());  
}
```

A yellow rounded rectangle with a slight gradient and a thin black border, centered on the page.

Test Readability Model



Training Data Collection



Machine Learning Model



Automatically Generated  
Readable Unit Tests

# Test readability judgments



```
public void test3() throws Throwable {  
    LongAdder longAdder0 = new LongAdder();  
    longAdder0.reset();  
    assertEquals(0, longAdder0.shortValue());  
}
```

Snippet Pack demo: 1 of 4

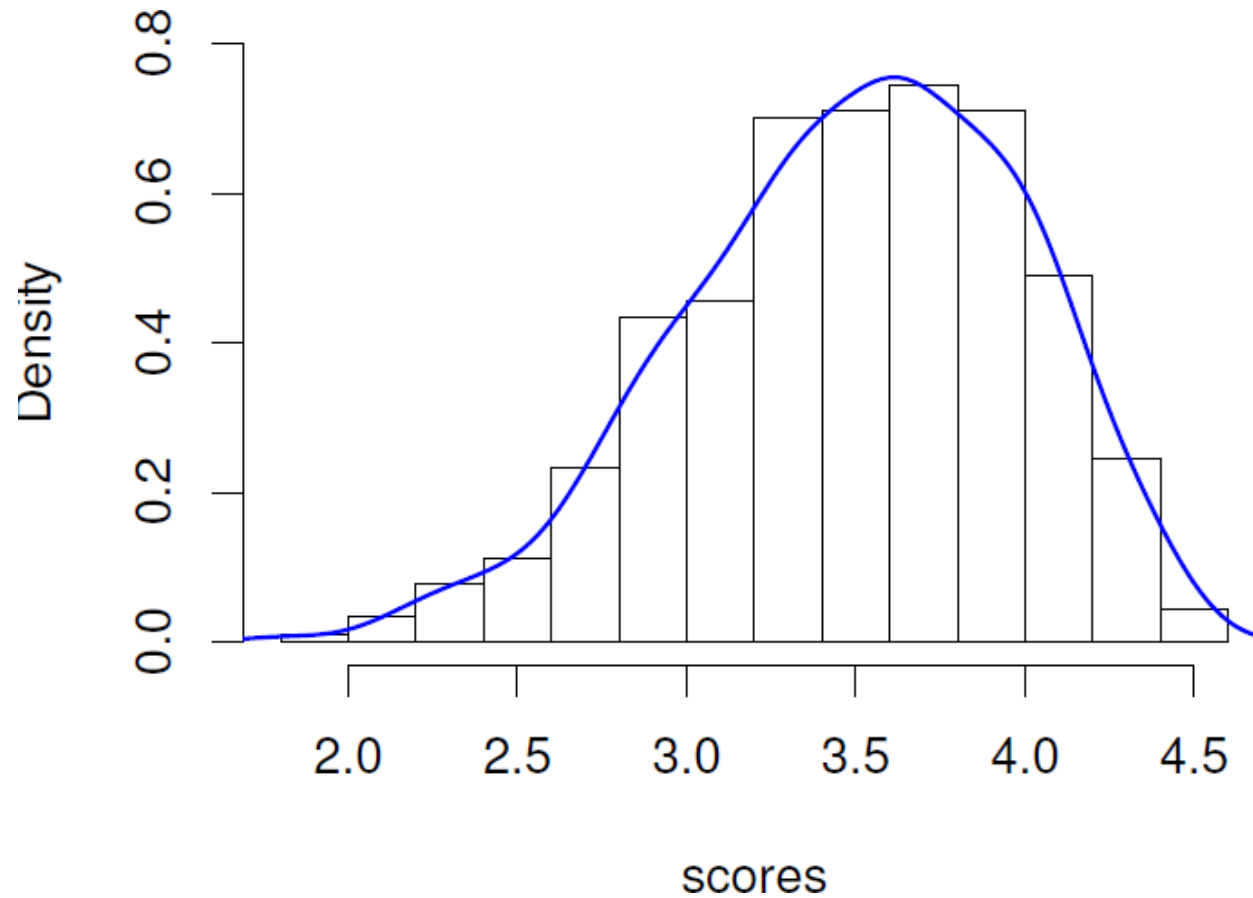
1 2 3 4 5

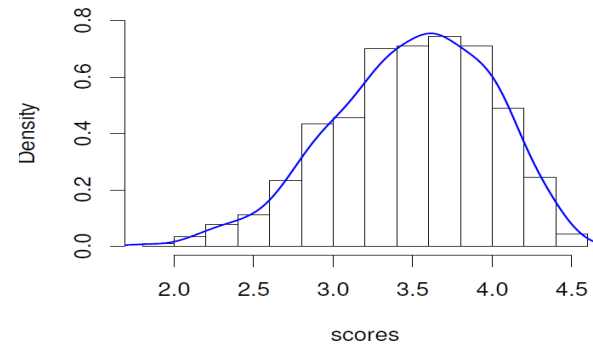


Skip

# Test readability judgments

- 15,669 human judgments of readability





Machine Learning Model



Automatically Generated  
Readable Unit Tests

```
public void test0() throws Throwable {  
    String string0 = "";  
    String string1 = "];"  
    MessageDigestHashFunction messageDigestHashFunction0 = null;  
    try {  
        messageDigestHashFunction0 = new MessageDigestHashFunction(string0, string1);  
    } catch (AssertionError e) {  
        //  
        // java.security.NoSuchAlgorithmException: MessageDigest not available  
        //  
    }  
}
```

Test Length = 10

Unique Identifiers = 3

Max Line Length = 77

Has Exception = True

Constructors = 1

Assertions = 0

String Length = 1

... 99 other features

max\_identifier\_length total\_keywords avg\_blank\_lines avg\_digits total\_casts  
avg\_loops total\_distinct\_classes avg\_class\_instances total\_string\_length avg\_assertions  
total\_strings total\_method\_invocations total\_blank\_lines max\_characters total\_floats avg\_distinct\_methods  
max\_tokens total\_loops total\_assertions avg\_unique\_identifiers total\_halstead\_effort  
total\_additional\_assertions max\_types max\_digits total\_exceptions total\_byte\_entropy  
total\_identifier\_ratio total\_has\_assertions max\_line\_length max\_casts  
total\_unique\_identifiers avg\_string\_length total\_digits avg\_distinct\_classes total\_distinct\_methods  
total\_characters max\_unique\_identifiers total\_branches max\_booleans total\_distinct\_methods  
avg\_strings total\_class\_instances total\_token\_entropy avg\_assignments avg\_characters max\_string\_length  
avg\_method\_invocations avg\_floats avg\_tokens avg\_branches total\_exceptional\_comments avg\_single\_characters  
total\_assignments avg\_comparison\_operations avg\_numbers max\_arrays total\_comparison\_operations  
max\_distinct\_methods total\_method\_ratio max\_identifiers total\_string\_score av\_parentheses  
max\_method\_invocations total\_indentation avg\_arithmetic\_operations max\_numbers  
total\_unused\_identifiers avg\_comments avg\_nulls max\_single\_characters total\_field\_accesses  
total\_booleans total\_single\_characters max\_indentation max char occ in test  
avg\_field\_accesses total\_commas total\_nulls avg\_line\_length max identifier occ in  
total\_types av\_commas max\_field\_accesses av\_spaces max\_commas  
total\_class\_ratio max\_nulls avg\_indentation total\_halstead\_difficulty avg\_exceptional\_comments  
avg\_booleans avg\_types max\_periods avg\_identifiers avg arrays total\_halstead\_volume  
total\_line\_length avg\_casts total\_arithmetic\_operations max\_floats avg\_keywords max char occ in line total\_identifiers  
total\_tokens av\_periods total\_numbers total\_periods max\_keywords max identifier occ in line  
max\_strings total\_comments total\_test\_length avg\_identifier\_length total\_identifier\_length total\_spaces

# High dimensional feature space

**IMPORTANT** – reduce noise, overfitting, efficiency.

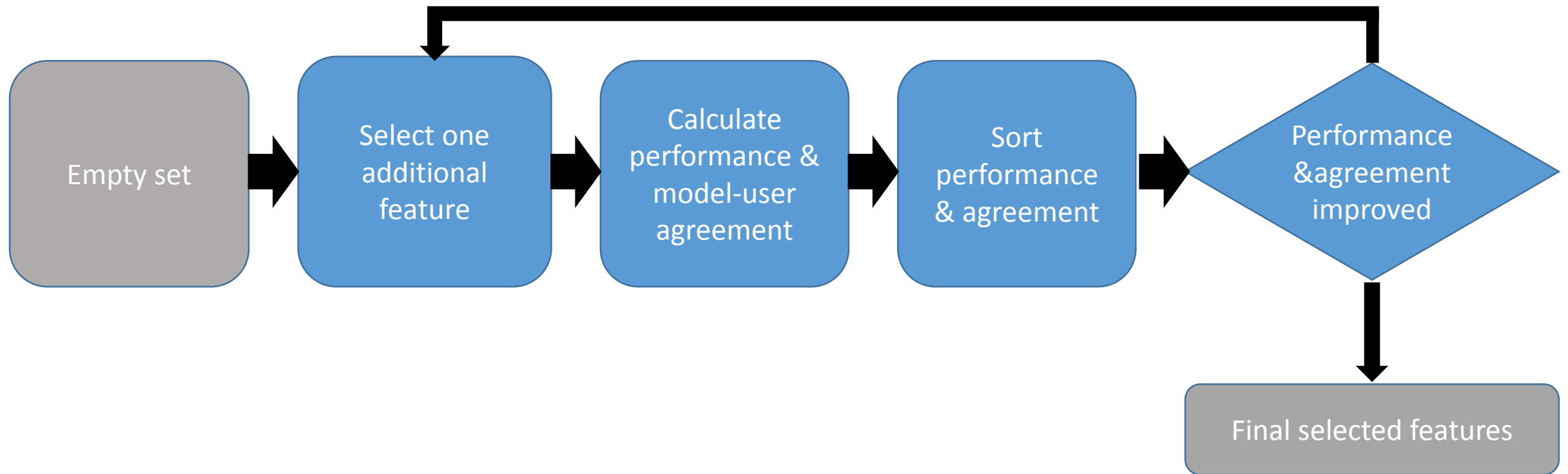
- A feature is good when it is relevant but not redundant to the other relevant features.
- Feature selection improves quality of learning algorithm.

# Feature selection

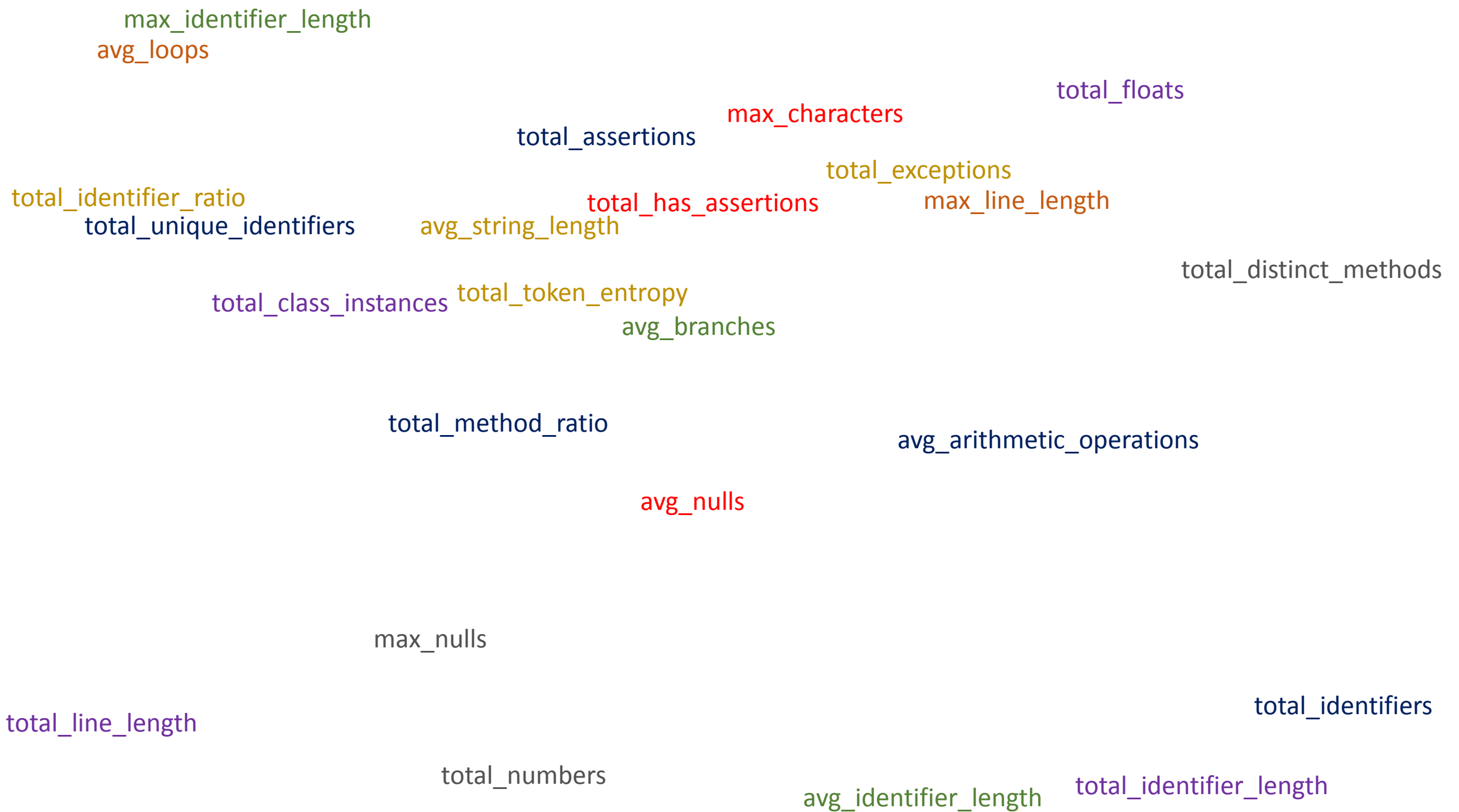
- Wrapper
  - **Forward feature selection**
  - Backward feature selection
  - Select one feature at time
  - Select by leaving one feature out
- Filter
  - Relief-F
  - Correlation



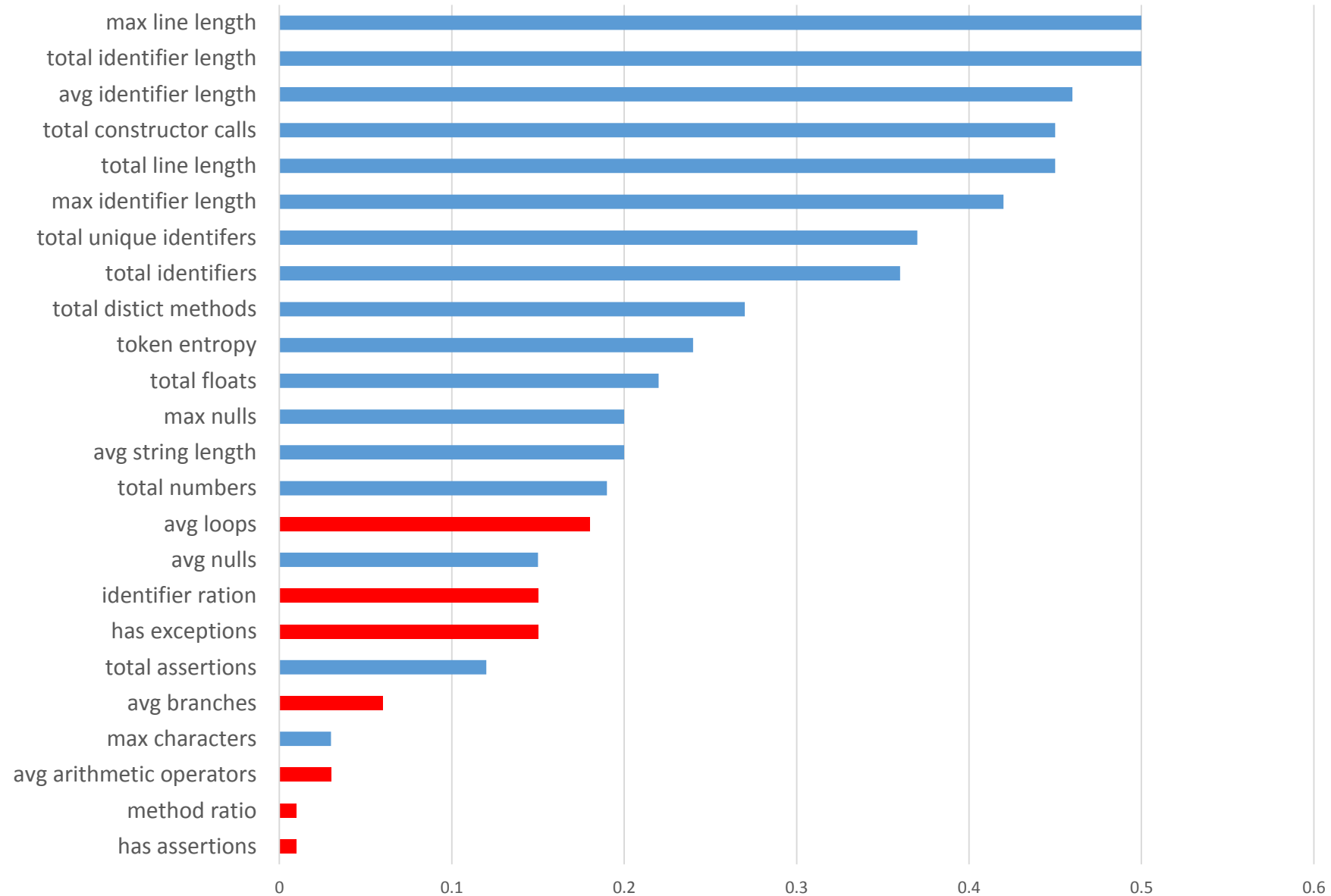
# Forward Feature Selection



max\_identifier\_length total\_keywords avg\_blank\_lines avg\_digits total\_casts  
avg\_loops total\_distinct\_classes avg\_class\_instances total\_string\_length avg\_assertions  
total\_strings total\_method\_invocations total\_blank\_lines max\_characters total\_floats avg\_distinct\_methods  
max\_tokens total\_loops total\_assertions avg\_unique\_identifiers total\_halstead\_effort  
total\_additional\_assertions max\_types max\_digits total\_exceptions total\_byte\_entropy  
total\_identifier\_ratio total\_unique\_identifiers avg\_string\_length total\_digits max\_line\_length max\_casts  
total\_characters max\_unique\_identifiers total\_branches avg\_distinct\_classes total\_distinct\_methods  
avg\_strings total\_class\_instances total\_token\_entropy avg\_assignments avg\_characters max\_string\_length  
avg\_method\_invocations avg\_floats avg\_tokens avg\_branches total\_exceptional\_comments avg\_single\_characters  
total\_assignments avg\_comparison\_operations avg\_numbers max\_arrays total\_comparison\_operations  
max\_distinct\_methods total\_method\_ratio max\_identifiers total\_string\_score av\_parentheses  
max\_method\_invocations total\_indentation avg\_arithmetic\_operations max\_numbers  
total\_unused\_identifiers avg\_comments avg\_nulls max\_single\_characters total\_field\_accesses  
total\_booleans total\_single\_characters max\_indentation max char occ in test  
avg\_field\_accesses total\_commas total\_nulls avg\_line\_length max identifier occ in  
total\_types av\_commas max\_field\_accesses av\_spaces max\_commas  
total\_class\_ratio max\_nulls avg\_indentation total\_halstead\_difficulty avg\_exceptional\_comments  
avg\_booleans avg\_types max\_periods avg\_identifiers avg arrays total\_halstead\_volume  
total\_line\_length avg\_casts total\_arithmetic\_operations max\_floats avg\_keywords max char occ in line total\_identifiers  
total\_tokens av\_periods total\_numbers total\_periods max\_keywords max identifier occ in line  
max\_strings total\_comments total\_test\_length avg\_identifier\_length total\_identifier\_length total\_spaces



# Feature predictive power



Training Data Collection



Machine Learning Model

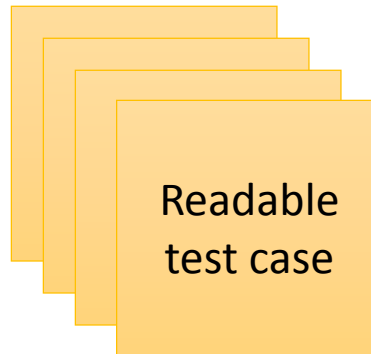


Automatically Generated  
Readable Unit Tests



EVASUITE

The logo for EVASUITE, featuring the word "EVASUITE" in a bold, black, sans-serif font. The letter "A" is replaced by a stylized Android robot head icon, which is black on top and blue on the bottom with white details.



# Readable test generation

- For a test  $t = \langle s1, s2, \dots, si \rangle$  with coverage goal  $c$  we:
  - iterate over the statements in the test from the last to the first statement
  - For each statement we determine the possible set of replacement statements
  - For each candidate replacement  $t'$  we determine if it still satisfies  $c$
  - Tests are then sorted based on their readability score

```
Foo foo = new Foo();  
Bar bar = new Bar("Some parameter", 17);  
foo.setBar(bar);  
assertTrue(foo.isBar());
```

```
Foo foo = new Foo();  
Bar bar = new Bar();  
foo.setBar(bar);  
assertTrue(foo.isBar());
```

```
Bar bar = new Bar();  
assertFalse(foo.isBar());
```

**EVALUATION ...**



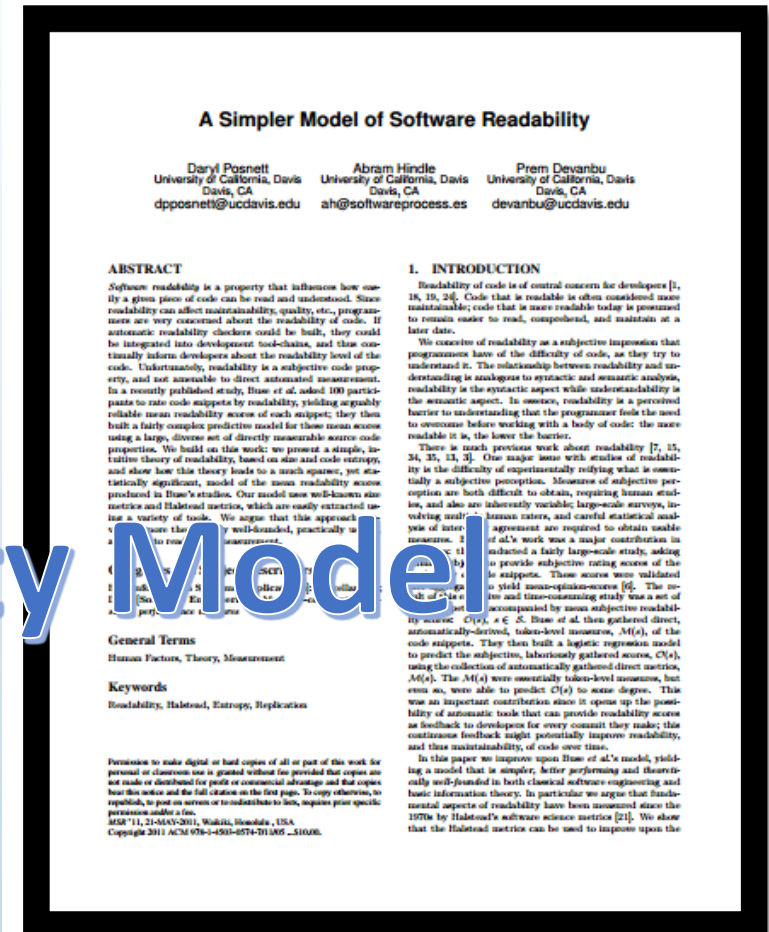
# RQ1: Code vs. Test Readability



Buse & Weimer

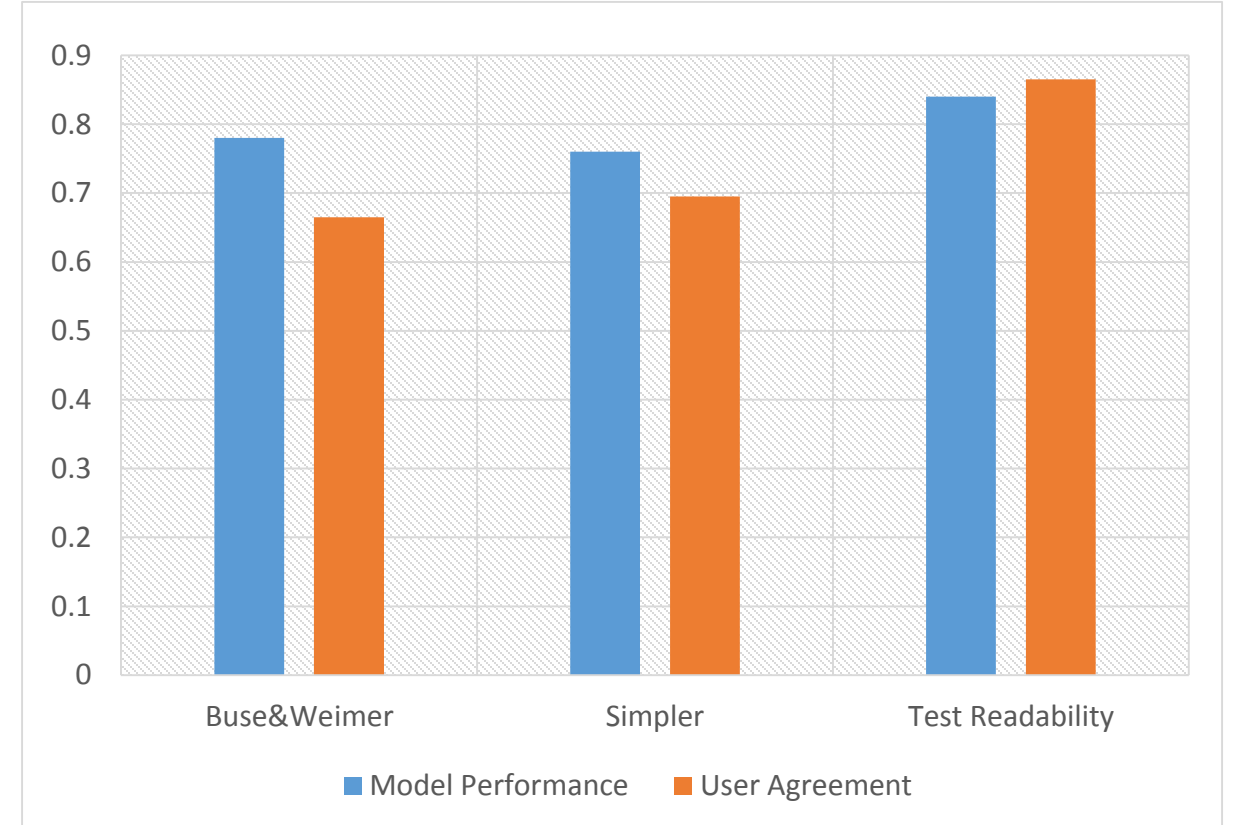
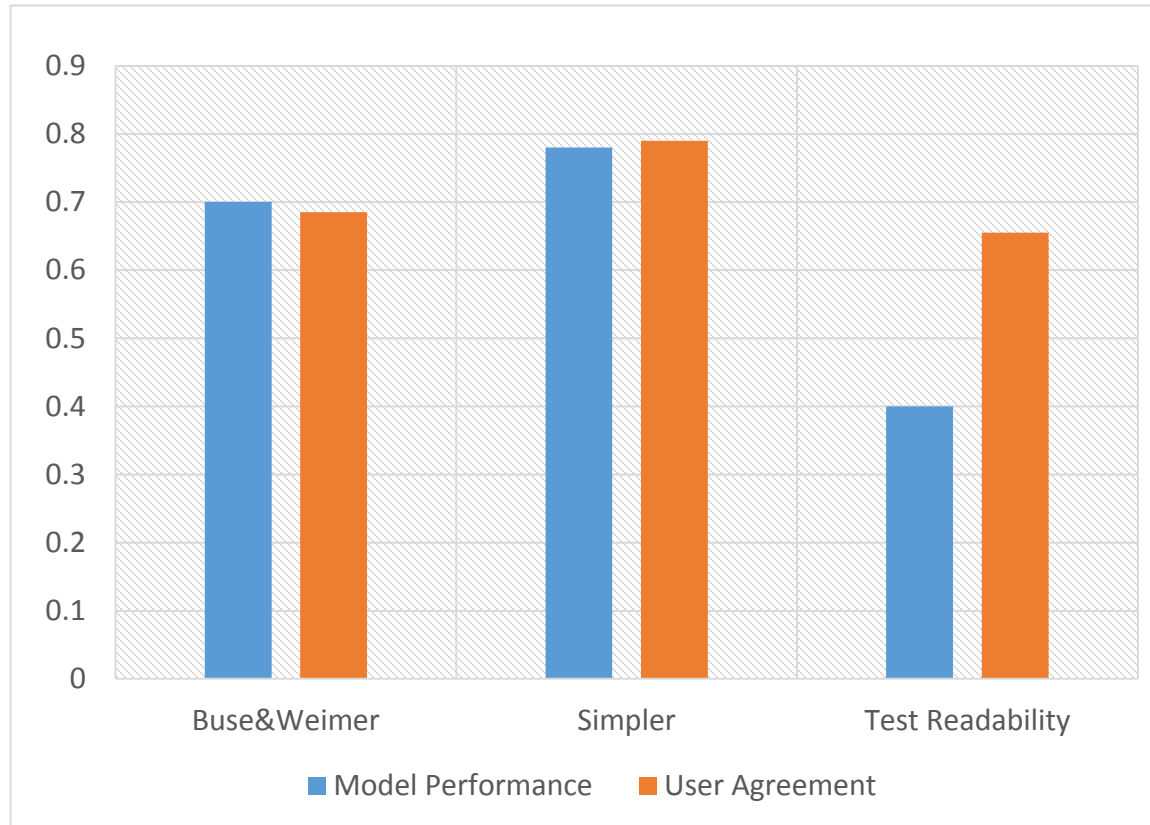
Readability prediction model  
code based on structural factors

CODE INDENTATION  
vs.  
Test Readability Model  
LENGTH OF STATEMENT



Posnett

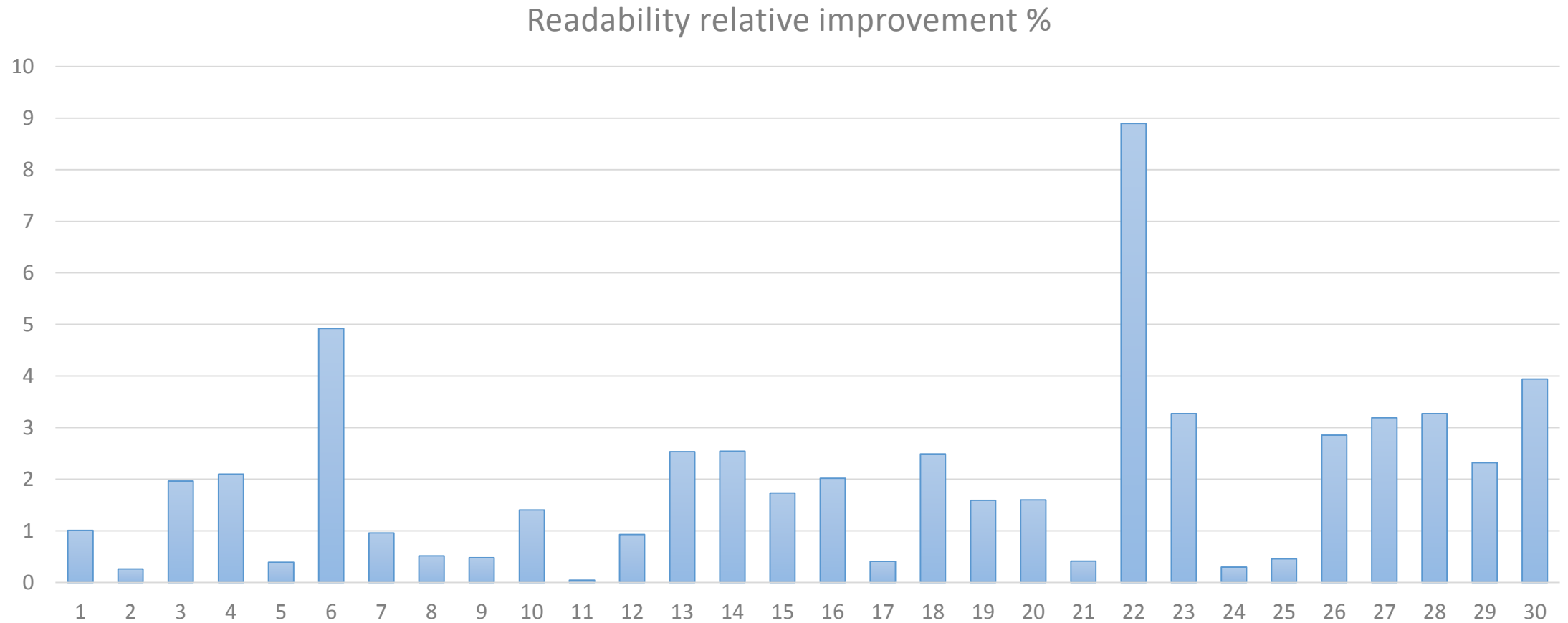
# RQ1: Code vs. Test Readability



# RQ2: Improved Test Generation

**EV****SUITE**

# RQ2: Improved Test Generation




# RQ3: Do Humans Prefer Readability Optimized Tests?

Java Unit Testing Study

Gordon

readability.evosuite.org/pair/study/question/0



0%

Test Case A

```
package org.apache.commons.cli;

import static org.junit.Assert.*;
import org.junit.Test;
import org.apache.commons.cli.CommandLine;
import org.apache.commons.cli.Option;

public class CommandLine_ESTest {

    @Test
    public void test0() throws Throwable {
        CommandLine commandLine0 = new CommandLine();
        boolean boolean0 = commandLine0.hasOption("lVW");
        String string0 = commandLine0.getOptionValue('');
        Option option0 = new Option((String) null, "lV", true, "addOption");
        commandLine0.addOption(option0);
        boolean boolean1 = commandLine0.hasOption("lVW");
        assertFalse(boolean1 == boolean0);
        assertTrue(boolean1);
    }
}
```

Test Case B

```
package org.apache.commons.cli;

import static org.junit.Assert.*;
import org.junit.Test;
import org.apache.commons.cli.CommandLine;
import org.apache.commons.cli.Option;

public class CommandLine_ESTest {

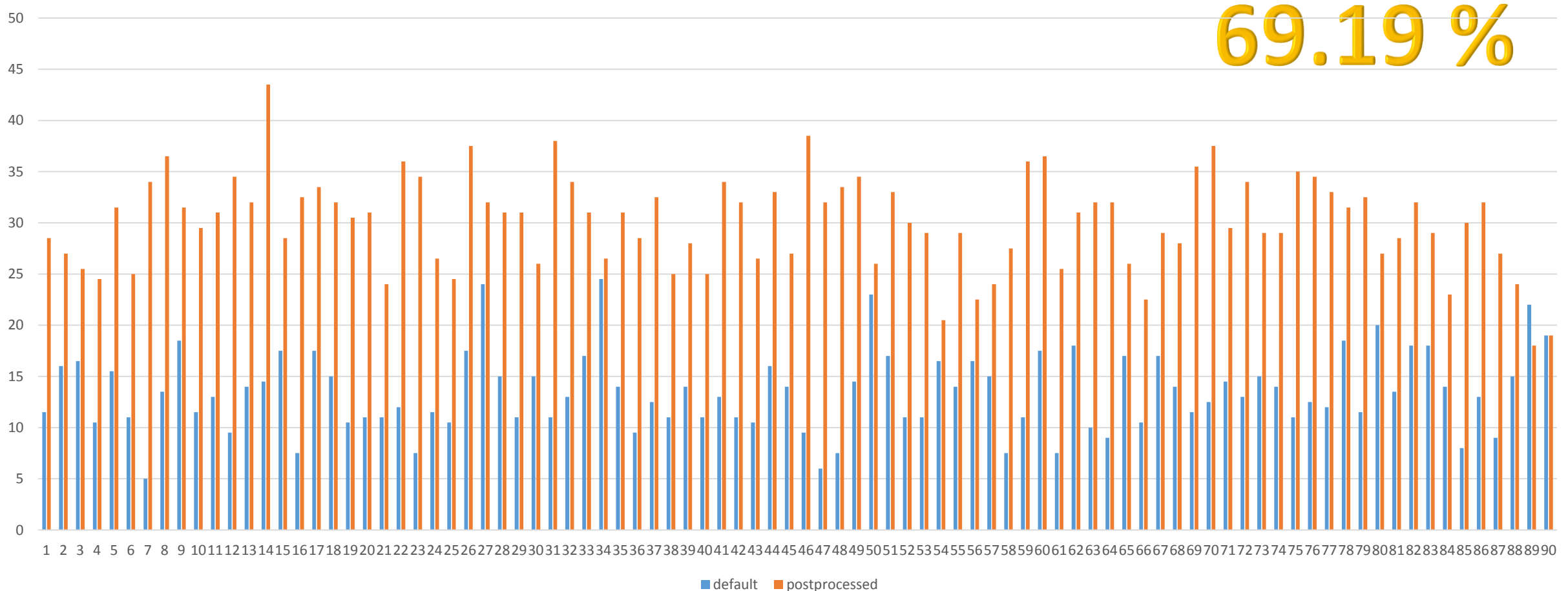
    @Test
    public void test0() throws Throwable {
        CommandLine commandLine0 = new CommandLine();
        Option option0 = new Option("", false, "");
        commandLine0.addOption(option0);
        boolean boolean0 = commandLine0.hasOption('-');
        assertTrue(boolean0);
    }
}
```

Test A

Test B

Next »

# RQ3: Do Humans Prefer Readability Optimized Tests?



# RQ4: Does Readability Optimization Improve Human Understanding of Tests?



Select either the Pass or Fail button below the source code view, and click Next or Submit your answers.

### Source Explorer

org.apache.commons.cli  
▶ Option  
▶ OptionValidator

### Test Case

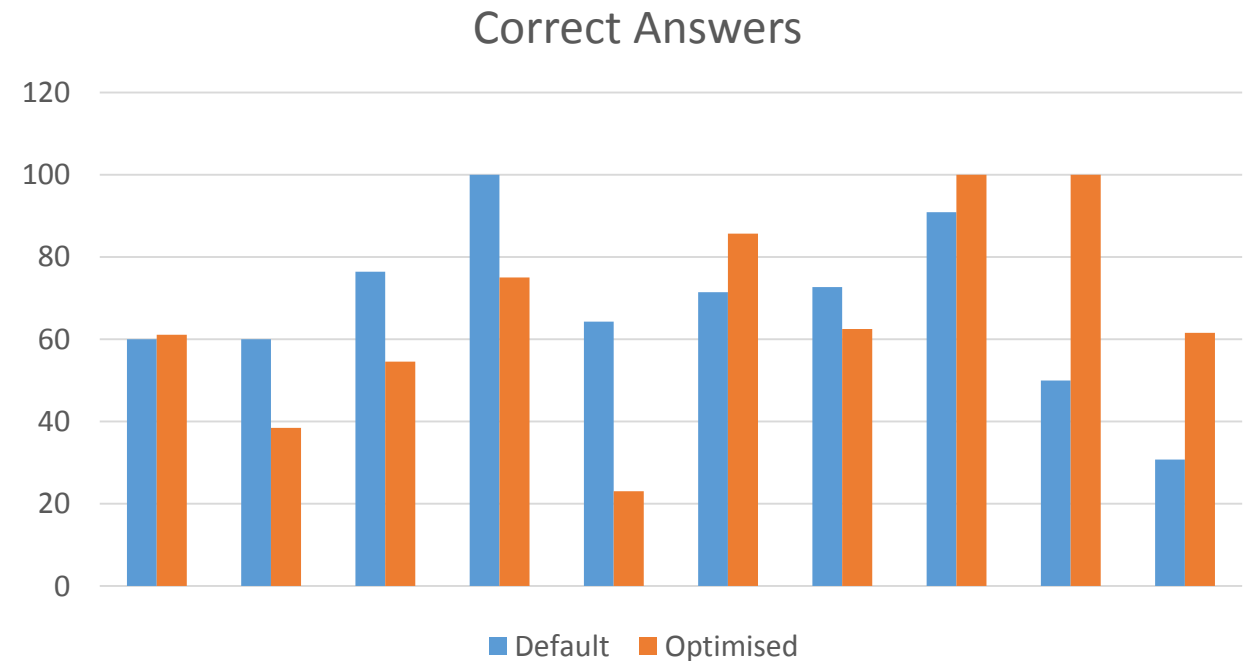
```
package org.apache.commons.cli;  
  
import static org.junit.Assert.*;  
import org.junit.Test;  
import org.apache.commons.cli.Option;  
  
public class Option_ESTest {  
  
    @Test  
    public void test0() throws Throwable {  
        Option option0 = new Option((String) null, " ");  
        // Undeclared exception!  
        try {  
            int int0 = option0.getId();  
            fail("Expecting exception: NullPointerException");  
        } catch (NullPointerException e) {  
            //  
            // no message in exception (getMessage() returned null)  
            //  
        }  
    }  
}
```

Pass

Fail

Next »

# RQ4: Does Readability Optimization Improve Human Understanding of Tests?



```

public void test0() throws Throwable {
    Option option0 = new Option("1P01Ca", "1P01Ca");
    String string0 = option0.getDescription();
    assertNull(string0);

    Option option1 = new Option((String) null, " (", false, "1P01Ca");
    String string1 = option1.getKey();
    assertNull(string1);
    assertEquals(-2, option1.getArgs());
    assertEquals(false, option1.hasArgName());
    assertEquals("[ option: null ( [ARG...] :: 1P01Ca ]", option1.toString());
}

```

Default

```

public void test0() throws Throwable {
    Option option0 = new Option((String) null, " ");
    // Undeclared exception!
    try {
        int int0 = option0.getId();
        fail("Expecting exception: NullPointerException");
    } catch (NullPointerException e) {
        //
        // no message in exception (getMessage() returned null)
        //
    }
}

```

Optimized

```

public void test0() throws Throwable {
    FixedOrderComparator.UnknownObjectBehavior[] fixedOrderComparator_UnknownObjectBehaviorArray0 =
FixedOrderComparator.UnknownObjectBehavior.values();
    LinkedList<FixedOrderComparator<Object>> linkedList0 = new LinkedList<FixedOrderComparator<Object>>();
    FixedOrderComparator<FixedOrderComparator<Object>> fixedOrderComparator0 = new
FixedOrderComparator<FixedOrderComparator<Object>>((List<FixedOrderComparator<Object>>) linkedList0);
    LinkedList<Object> linkedList1 = new LinkedList<Object>();
    FixedOrderComparator<Object> fixedOrderComparator1 = new FixedOrderComparator<Object>((List<Object>) linkedList1);
    boolean boolean0 = fixedOrderComparator0.add(fixedOrderComparator1);
    FixedOrderComparator<Object> fixedOrderComparator2 = new FixedOrderComparator<Object>((Object[]))
fixedOrderComparator_UnknownObjectBehaviorArray0);
    boolean boolean1 = fixedOrderComparator0.addAsEqual(fixedOrderComparator1, fixedOrderComparator2);
    assertEquals(FixedOrderComparator.UnknownObjectBehavior.EXCEPTION,
fixedOrderComparator2.getUnknownObjectBehavior());
    assertEquals(FixedOrderComparator.UnknownObjectBehavior.EXCEPTION,
fixedOrderComparator1.getUnknownObjectBehavior());
    assertTrue(boolean1);
}

```

```

public void test0() throws Throwable {
    String[] stringArray0 = new String[11];
    FixedOrderComparator<String> fixedOrderComparator0 = new FixedOrderComparator<String>(stringArray0);
    boolean boolean0 = fixedOrderComparator0.addAsEqual((String) null, " not known to ");
    assertEquals(FixedOrderComparator.UnknownObjectBehavior.EXCEPTION,
fixedOrderComparator0.getUnknownObjectBehavior());
    assertTrue(boolean0);
}

```

```

1 package org.apache.commons.beanutils.locale.converters;
2
3 import static org.junit.Assert.*;
4
5 public class DateLocaleConverter_ESTest {
6
7     @Test
8     public void test0() throws Throwable {
9         DateLocaleConverter dateLocaleConverter0 = new DateLocaleConverter();
10        Locale locale0 = Locale.CANADA_FRENCH;
11        DateLocaleConverter dateLocaleConverter1 = new DateLocaleConverter((Object) dateLocaleConverter0, locale0);
12        String string0 = locale0.getDisplayLanguage();
13        DateLocaleConverter dateLocaleConverter2 = (DateLocaleConverter)dateLocaleConverter1.convert((Object) dateLocaleConverter0);
14        DateLocaleConverter dateLocaleConverter3 = new DateLocaleConverter(locale0, true);
15        Class<Object> class0 = Object.class;
16        // Undeclared exception!
17        try {
18            DateLocaleConverter dateLocaleConverter4 = (DateLocaleConverter)dateLocaleConverter3.convert(class0, (Object) dateLocaleConverter2, "\u06CD\u06587");
19            fail("Expecting exception: ConversionException");
20        } catch (ConversionException e) {
21            // Error parsing date 'org.apache.commons.beanutils.locale.converters.DateLocaleConverter@1' at position=0
22            //
23        }
24    }
25 }

```

Finished after 54.453 seconds

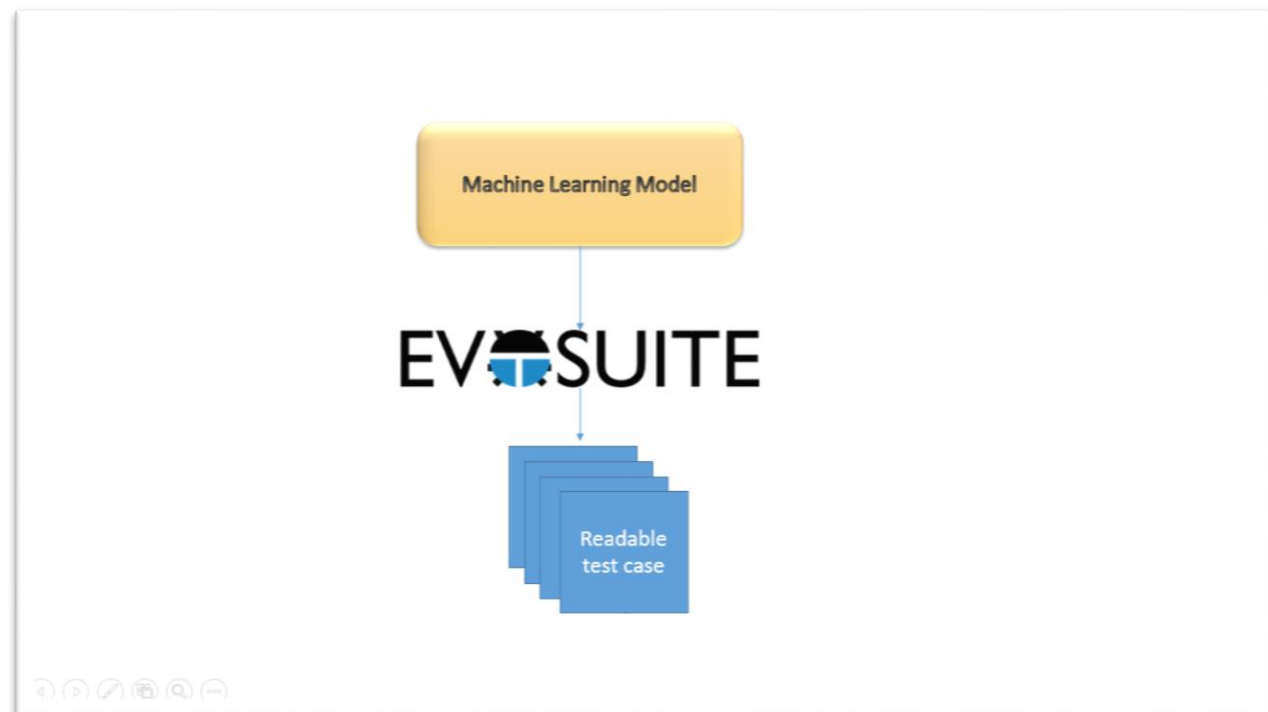
Runs: 2/2 Errors: 0 Failures: 1

Failure Trace

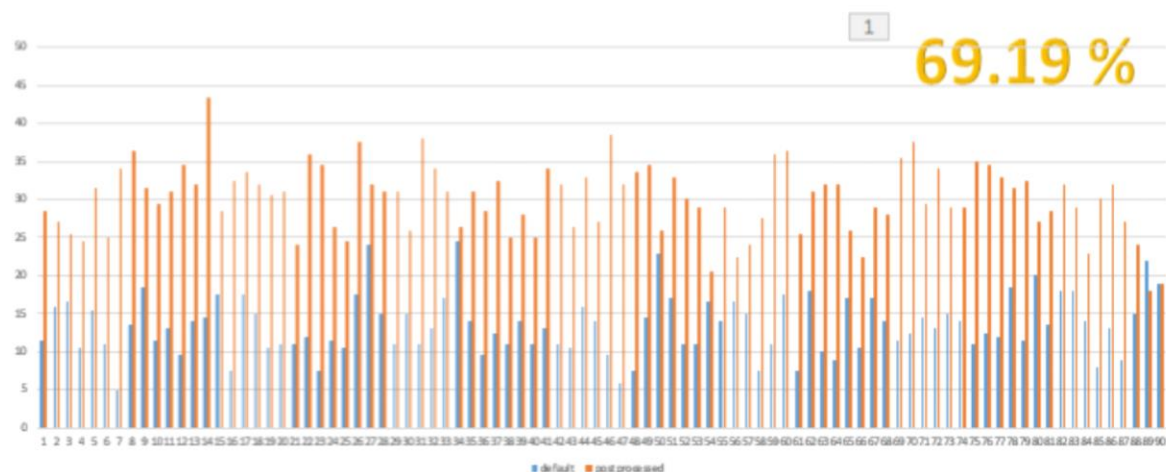
```

java.lang.AssertionError: Wrong number of goals: expected:<15> but was:<8>
    at org.evosuite.idNaming.TestIDNaming.testIDNamingOn(TestIDNaming.java:65)

```



## Do Humans Prefer Readability Optimized Tests?



## Does Readability Optimization Improve Human Understanding of Tests?

