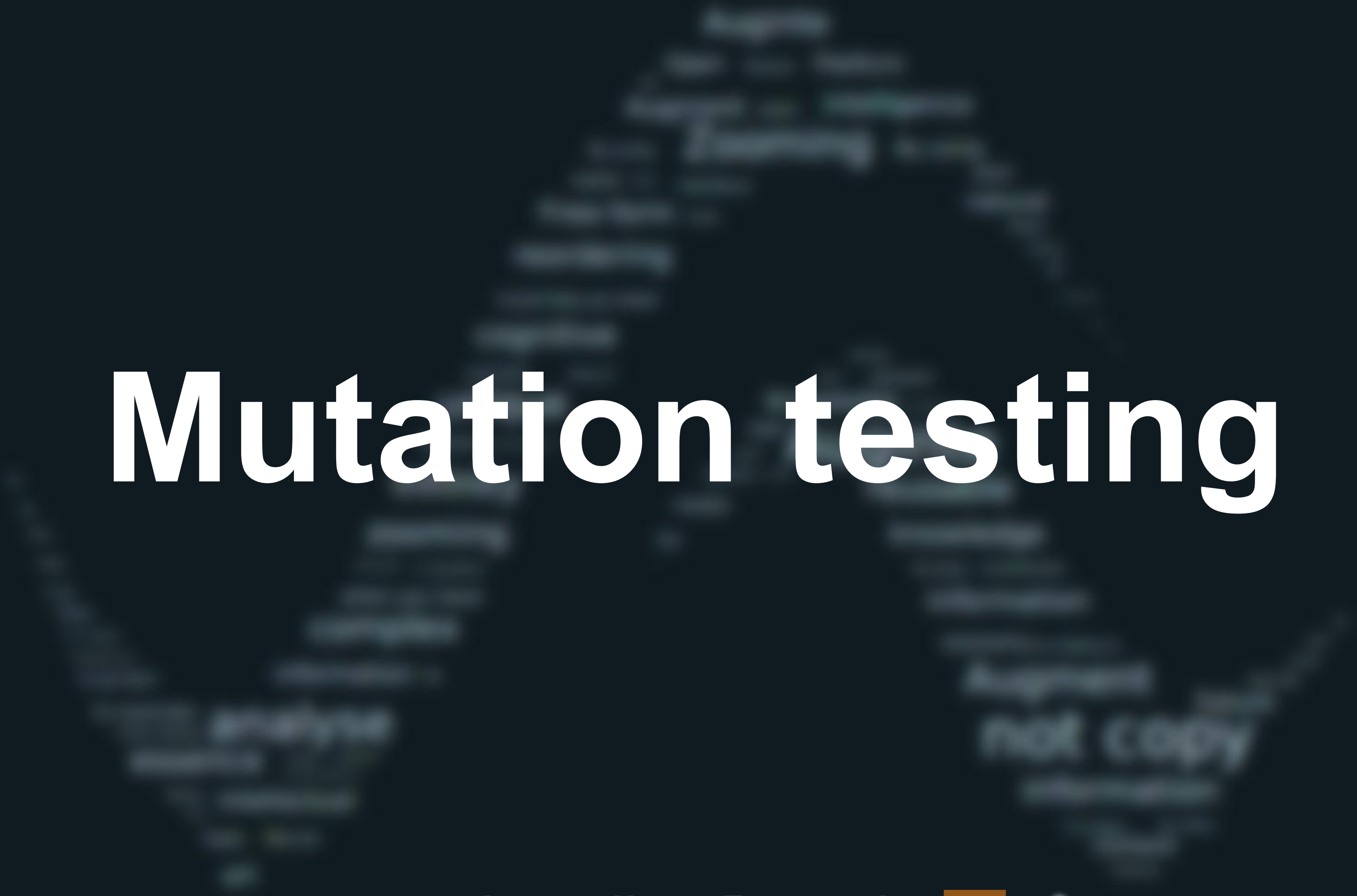


Mutation testing

Aurelijus Banelis

Aurelijus Banelis

Software developer
aurelijus.banelis.lt
aurelijus@banelis.lt

Mutation testing

with PHP

```
359 mutations were generated:
     280 mutants were killed
      46 mutants were not covered by tests
      25 covered mutants were not detected
       4 fatal errors were encountered
       4 time outs were encountered

Metrics:
    Mutation Score Indicator (MSI): 80%
    Mutation Code Coverage: 87%
    Covered Code MSI: 92%
```

```
Mutation Testing is commencing on 48 files...
(.: killed, M: escaped, S: uncovered, E: fatal error, T: timed out)

.S.S.........E..................................M...M....... |   60 (17/48)
M.......M.........M................................SS....... |  120 (24/48)
....S............M.....M.................................M.. |  180 (25/48)
.........................S......EE.M........E.T.SSSSSSSSSSSS |  240 (35/48)
SSSSSSSSSSSSSSSSSSSSSSSSSS..S....M.MMMM...M.MMM............. |  300 (41/48)
.M...M.M..S.MMM..M.........T..TT...........................
```

```
    - 100.00% src/Symfony/Component/CssSelector/Parser/Tokenizer/TokenizerEscaping.php
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/FunctionExtension.php
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/HtmlExtension.php
    -  96.77% src/Symfony/Component/CssSelector/XPath/Extension/NodeExtension.php
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/PseudoClassExtension.php
    -  88.33% src/Symfony/Component/CssSelector/XPath/Translator.php
```

COVERAGE		LINES
60.0	...ent/CssSelector/Exception/SyntaxErrorException.php	73
85.71	...nent/CssSelector/Parser/Handler/CommentHandler.php	48
88.33	...Symfony/Component/CssSelector/XPath/Translator.php	298
88.89	...onent/CssSelector/Parser/Handler/StringHandler.php	88

Can You Trust Your Tests?

2015 Vaidas Pilkauskas & Tadas Ščerbinskas

Inspired by Vaidas’ presentation

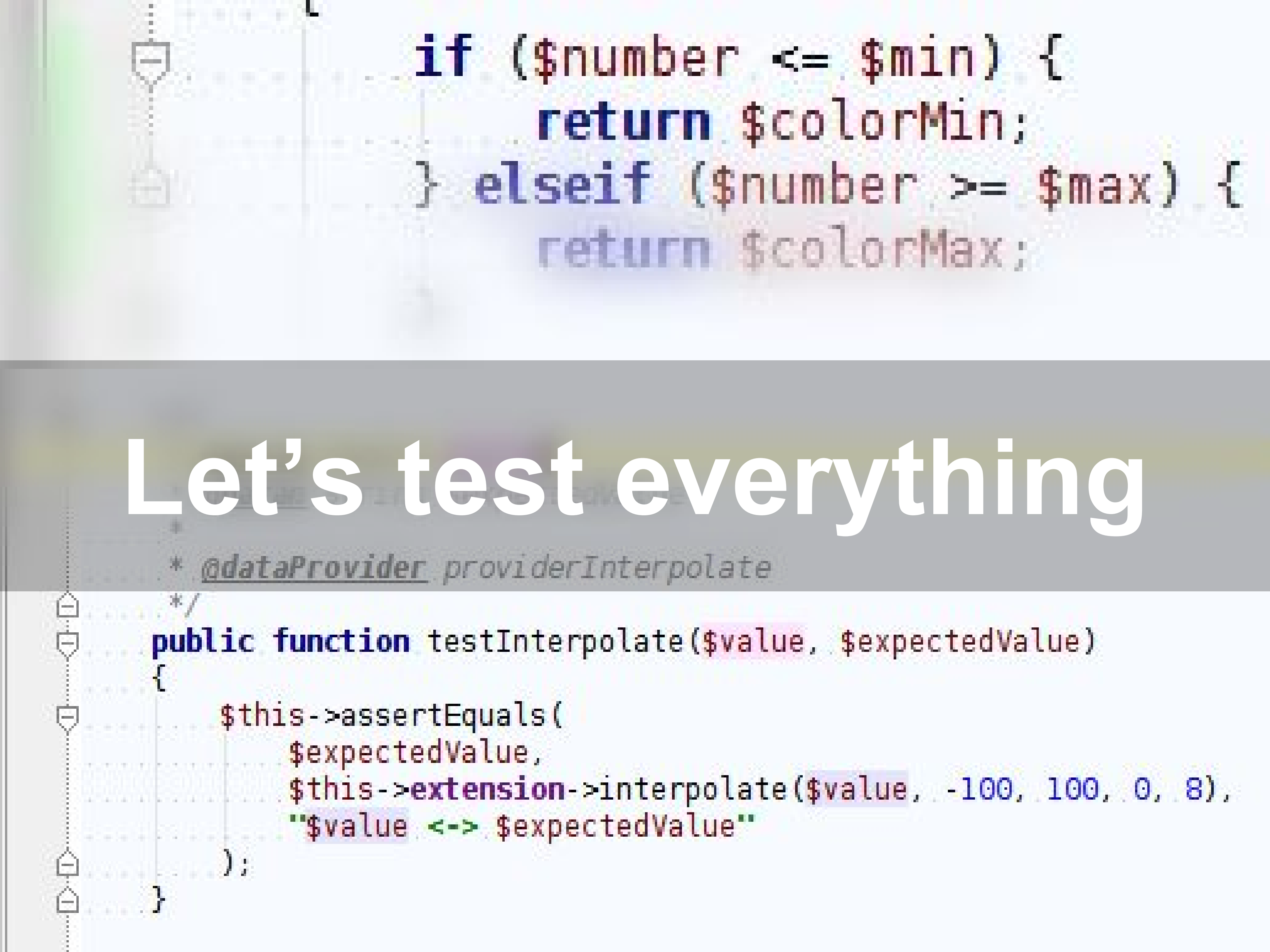
WHY	**Why we need mutation testing**
WHAT	**What it is and how it works**
HOW	**How it can be used in practice**

WHY	**Why we need mutation testing**
WHAT	**What it is and how it works**
HOW	**How it can be used in practice**

It is my code
It should be good

Small mistakes
But huge impact

Illiustration: https://www.youtube.com/watch?v=_BgJEXQkjNQ



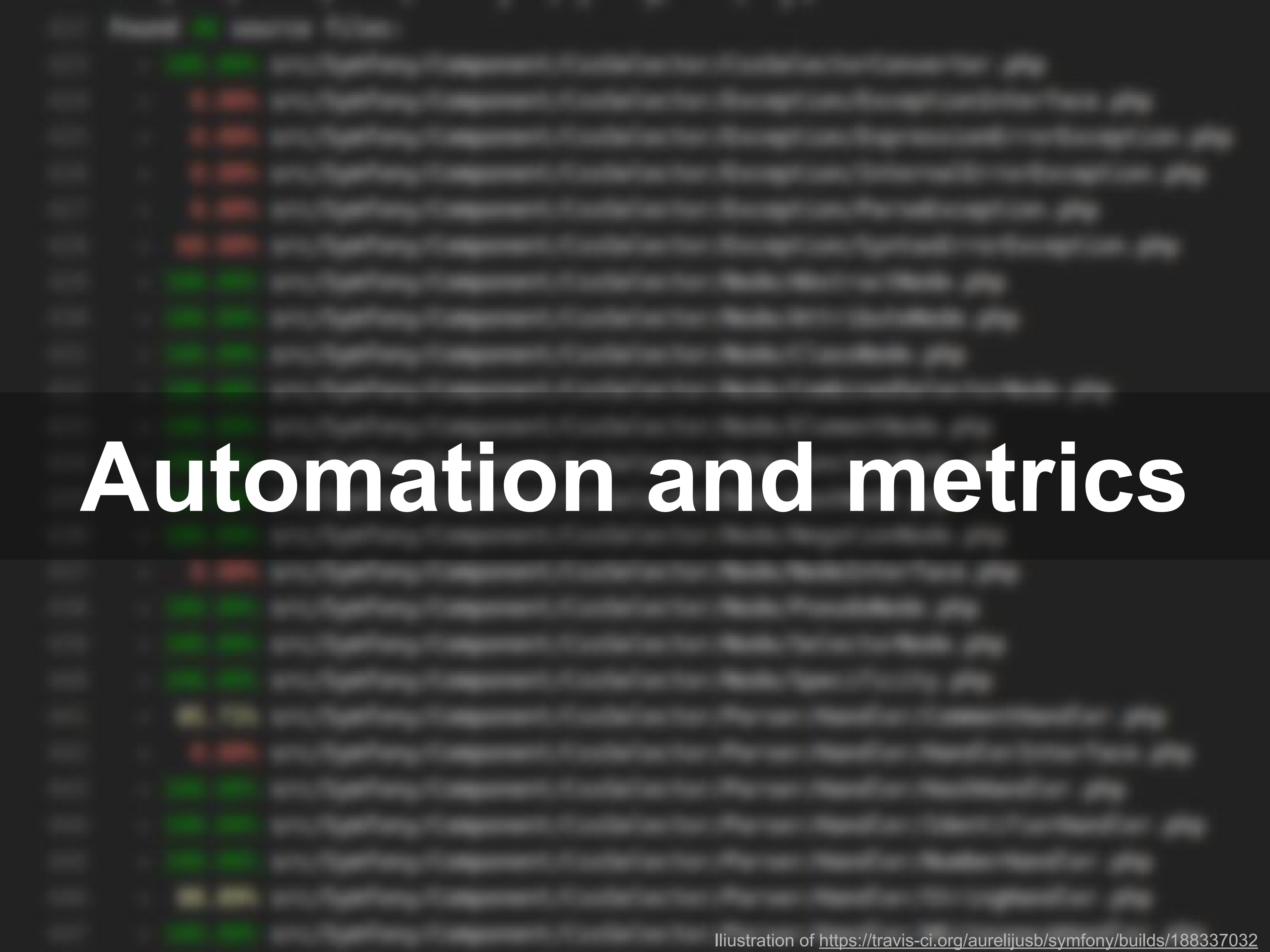
Let’s test everything

```
        if ($number <= $min) {
            return $colorMin;
        } elseif ($number >= $max) {
            return $colorMax;
```

```
     *
     * @dataProvider providerInterpolate
     */
    public function testInterpolate($value, $expectedValue)
    {
        $this->assertEquals(
            $expectedValue,
            $this->extension->interpolate($value, -100, 100, 0, 8),
            "$value <-> $expectedValue"
        );
    }
```

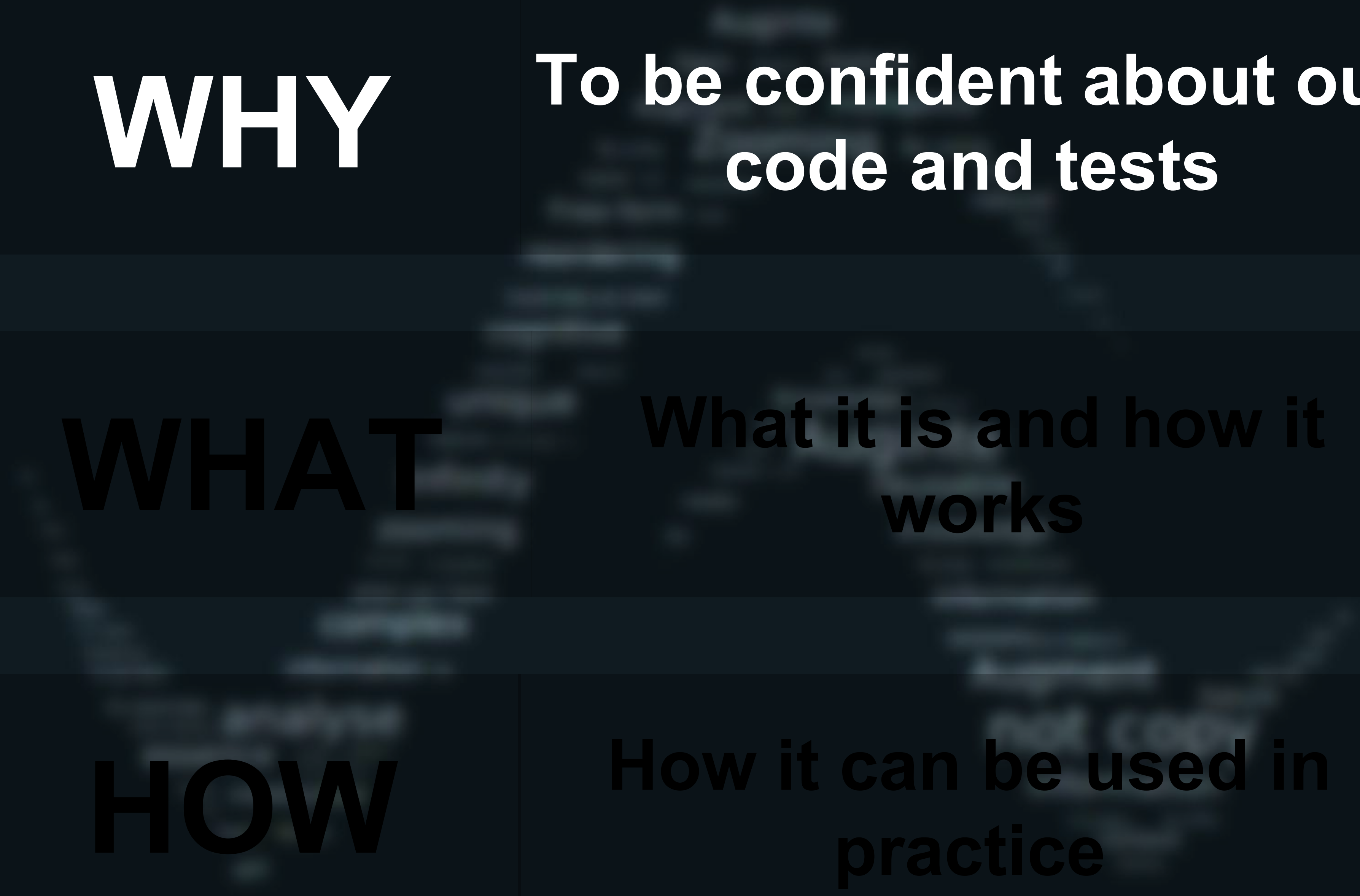
Who tests the tester?

```
umber <= $min)
urn $colorMin
($number >=

 * @dataProvider providerInterpolate
 */
public function testInterpolate($value,
```



Automation and metrics

Iliustration of https://travis-ci.org/aurelijusb/symfony/builds/188337032

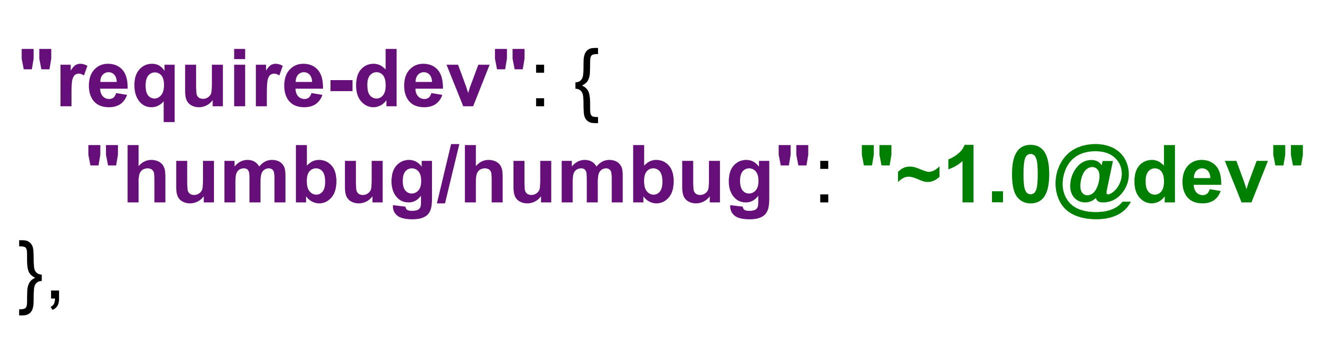


WHY	To be confident about our code and tests
WHAT	What it is and how it works
HOW	How it can be used in practice

WHY	Why we need mutation testing
WHAT	What it is and how it works
HOW	How it can be used in practice

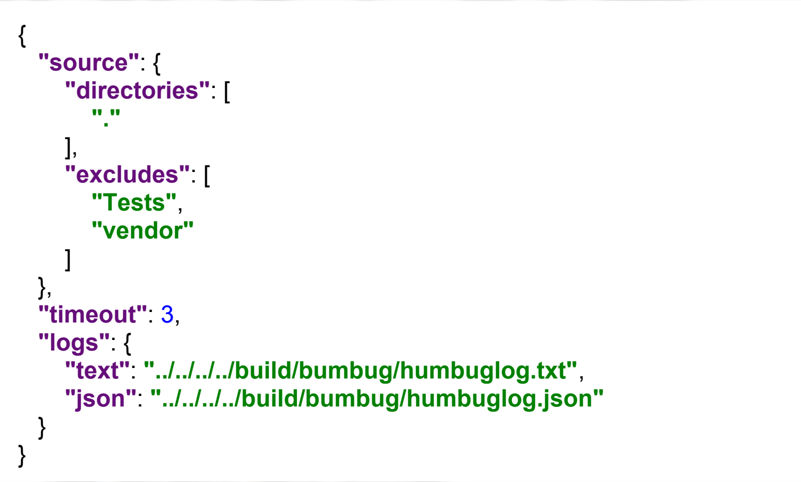
Hambug

One of PHP mutation testing tools



```
"require-dev": {
   "humbug/humbug": "~1.0@dev"
},
```

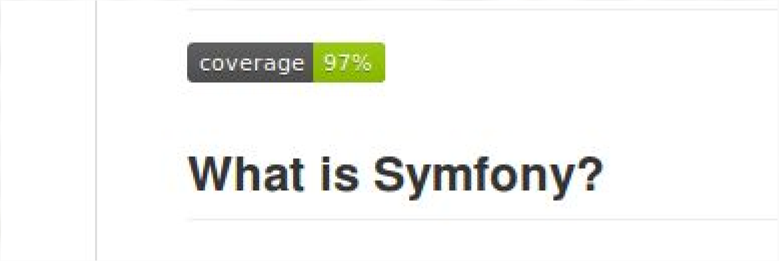
Getting started



```
{
   "source": {
      "directories": [
         "."
      ],
      "excludes": [
         "Tests",
         "vendor"
      ]
   },
   "timeout": 3,
   "logs": {
      "text": "../../../../build/bumbug/humbuglog.txt",
      "json": "../../../../build/bumbug/humbuglog.json"
   }
}
```

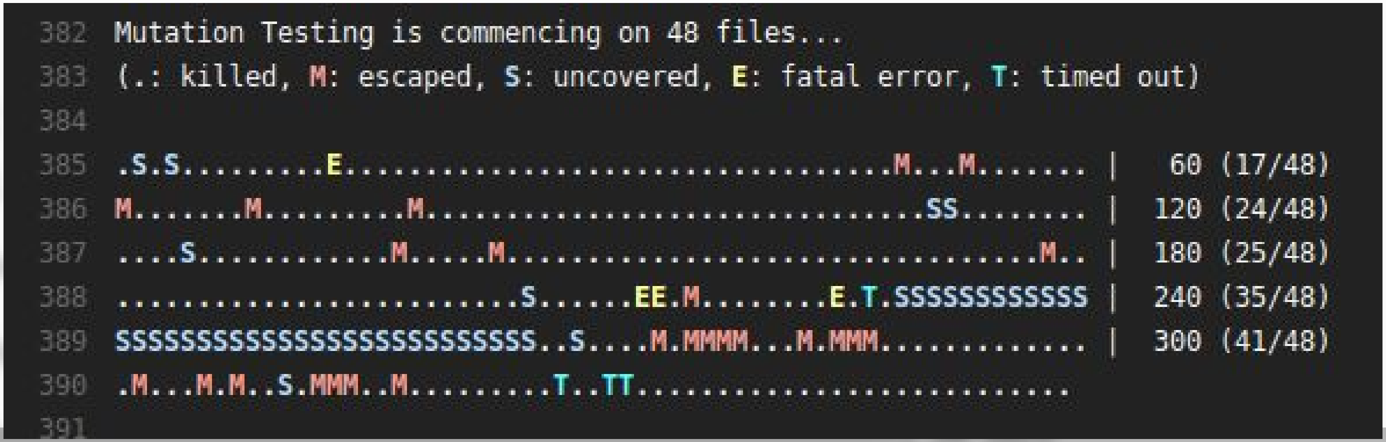
https://github.com/aurelijusb/symfony/tree/mutation-testing

Let’s test on symfony



coverage 97%

What is Symfony?

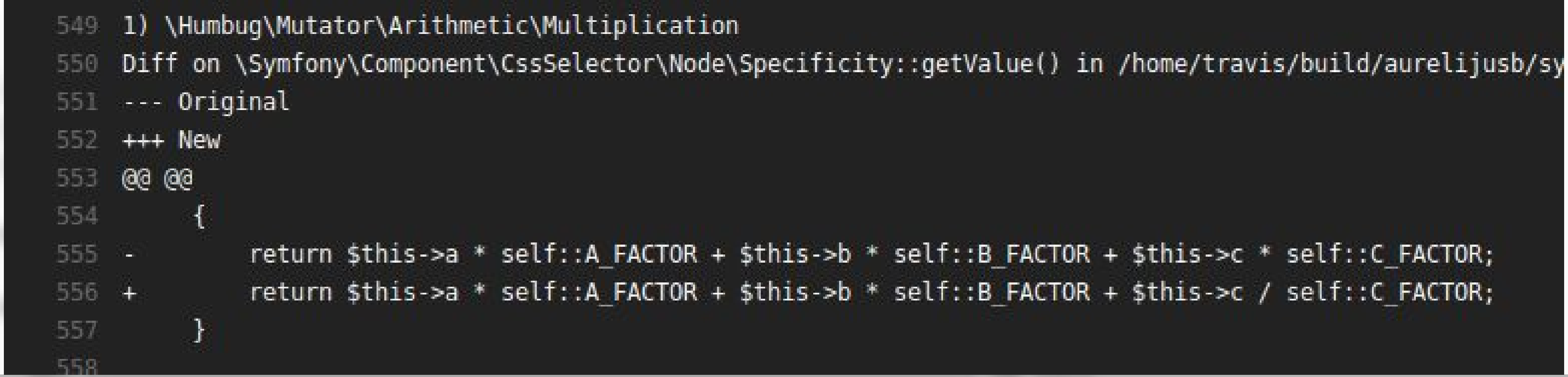


```
Mutation Testing is commencing on 48 files...
(.: killed, M: escaped, S: uncovered, E: fatal error, T: timed out)

.S.S.........E..................................M...M....... |   60 (17/48)
M.......M.........M...............................SS........ |  120 (24/48)
....S............M.....M.................................M.. |  180 (25/48)
.........................S......EE.M........E.T.SSSSSSSSSSSS |  240 (35/48)
SSSSSSSSSSSSSSSSSSSSSSSSSS..S....M.MMMM...M.MMM............. |  300 (41/48)
.M...M.M..S.MMM..M.........T..TT...........................
```

Mutation statistics

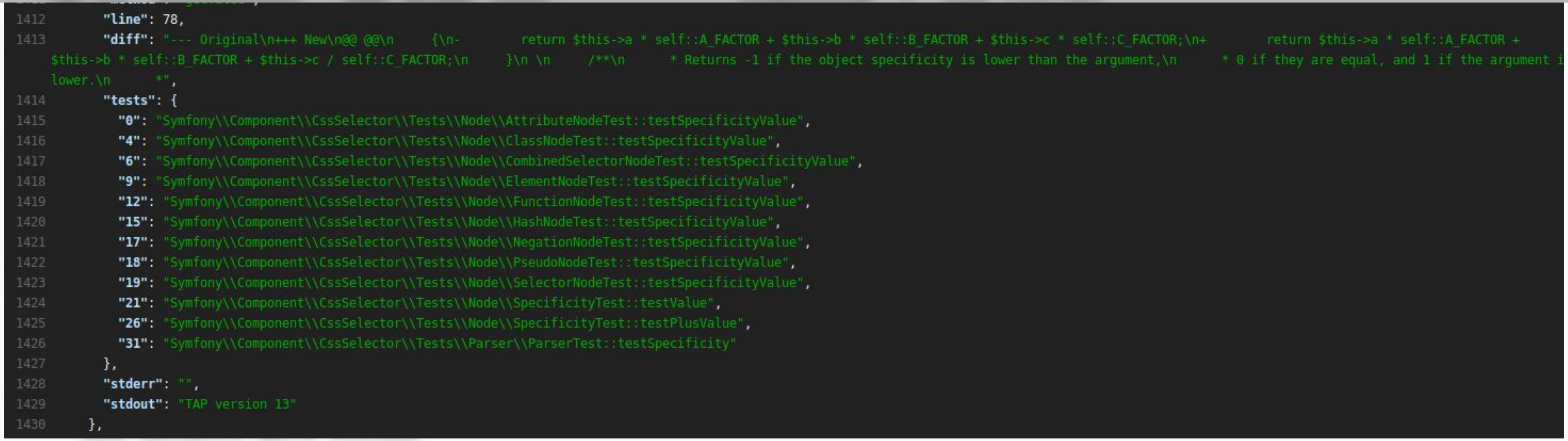
- **Killed Mutation (.):** A mutation that caused unit tests to fail which is a positive outcome.
- **Escaped Mutation (M):** A mutation where the unit tests still passed which is not what we want! Our unit tests should detect any behaviour changes.
- **Uncovered Mutation (S)**: A mutation which occurs on a line not covered by any unit test.
- **Fatal Error (E):** A mutation created a fatal error.
- **Timeout (T):** This is where unit tests exceed the allowed timeout

https://github.com/padraic/humbug
https://travis-ci.org/aurelijusb/symfony/builds/188651125

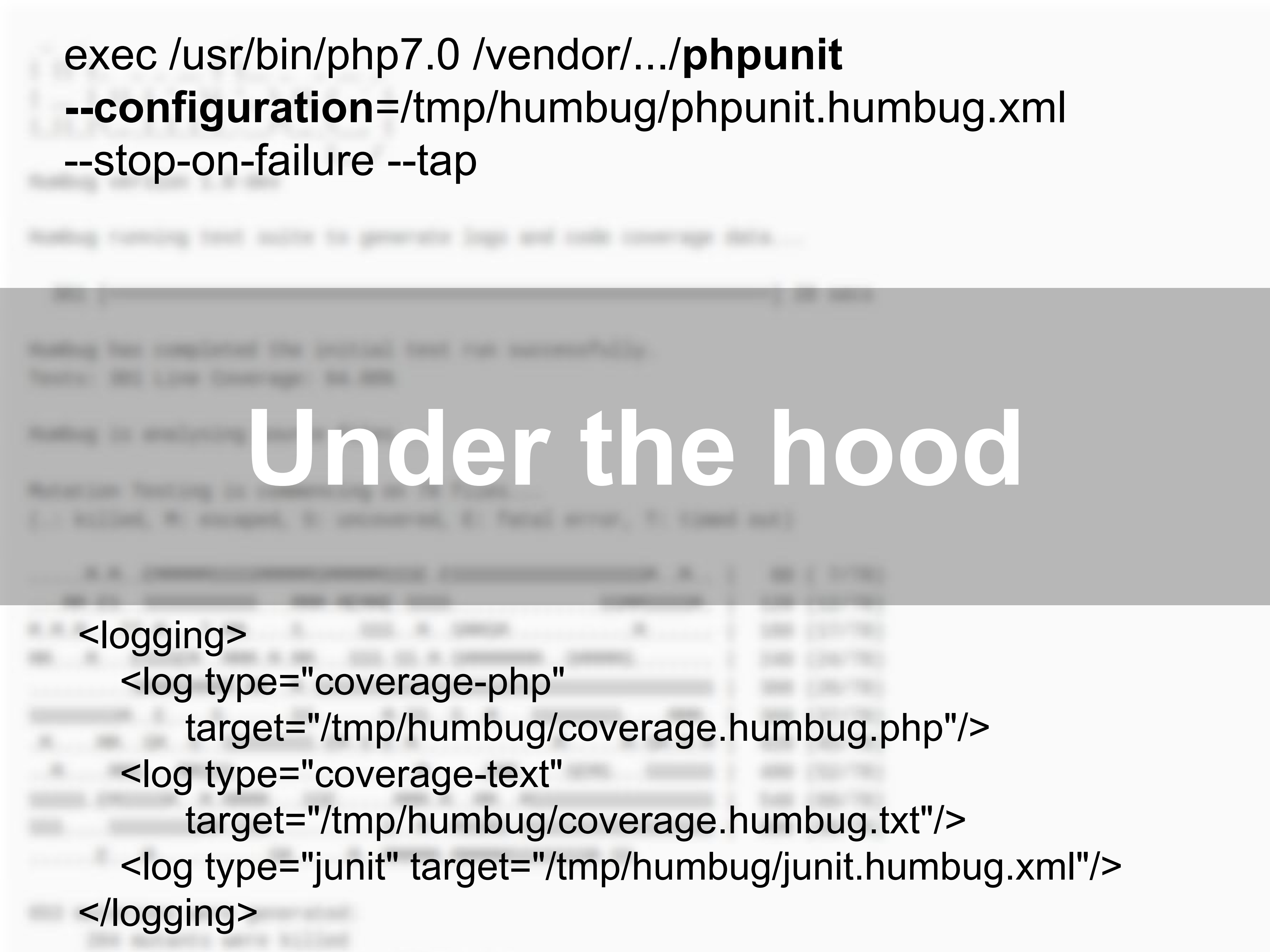


```
1) \Humbug\Mutator\Arithmetic\Multiplication
Diff on \Symfony\Component\CssSelector\Node\Specificity::getValue() in /home/travis/build/aurelijusb/sy
--- Original
+++ New
@@ @@
     {
-        return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;
+        return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c / self::C_FACTOR;
     }
```

Mutation Details



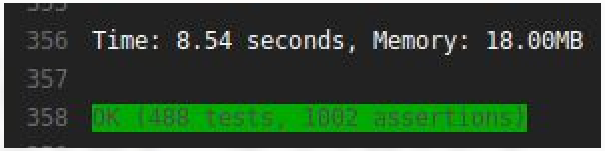
```
        "line": 78,
        "diff": "--- Original\n+++ New\n@@ @@\n     {\n-        return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;\n+        return $this->a * self::A_FACTOR +
    $this->b * self::B_FACTOR + $this->c / self::C_FACTOR;\n     }\n \n     /**\n      * Returns -1 if the object specificity is lower than the argument,\n      * 0 if they are equal, and 1 if the argument i
    lower.\n      *",
        "tests": {
          "0": "Symfony\\Component\\CssSelector\\Tests\\Node\\AttributeNodeTest::testSpecificityValue",
          "4": "Symfony\\Component\\CssSelector\\Tests\\Node\\ClassNodeTest::testSpecificityValue",
          "6": "Symfony\\Component\\CssSelector\\Tests\\Node\\CombinedSelectorNodeTest::testSpecificityValue",
          "9": "Symfony\\Component\\CssSelector\\Tests\\Node\\ElementNodeTest::testSpecificityValue",
          "12": "Symfony\\Component\\CssSelector\\Tests\\Node\\FunctionNodeTest::testSpecificityValue",
          "15": "Symfony\\Component\\CssSelector\\Tests\\Node\\HashNodeTest::testSpecificityValue",
          "17": "Symfony\\Component\\CssSelector\\Tests\\Node\\NegationNodeTest::testSpecificityValue",
          "18": "Symfony\\Component\\CssSelector\\Tests\\Node\\PseudoNodeTest::testSpecificityValue",
          "19": "Symfony\\Component\\CssSelector\\Tests\\Node\\SelectorNodeTest::testSpecificityValue",
          "21": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testValue",
          "26": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testPlusValue",
          "31": "Symfony\\Component\\CssSelector\\Tests\\Parser\\ParserTest::testSpecificity"
        },
        "stderr": "",
        "stdout": "TAP version 13"
      },
```



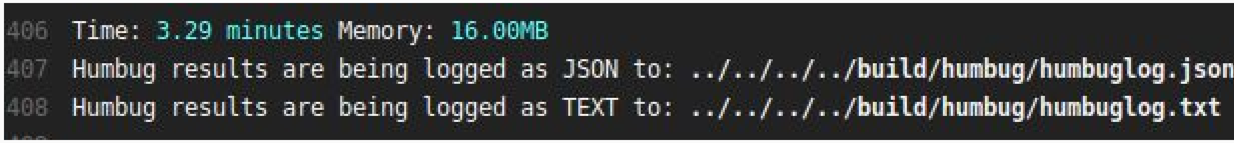
```
exec /usr/bin/php7.0 /vendor/.../phpunit
--configuration=/tmp/humbug/phpunit.humbug.xml
--stop-on-failure --tap
```

Under the hood

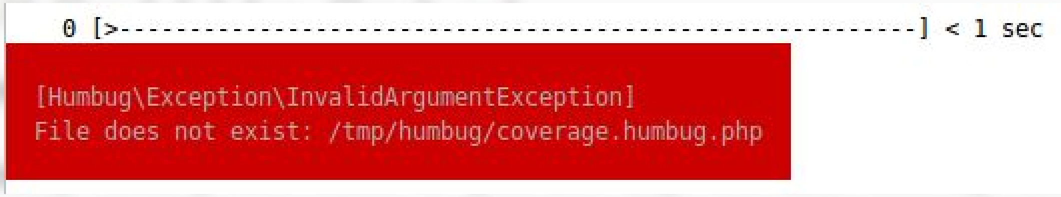
```
<logging>
   <log type="coverage-php"
        target="/tmp/humbug/coverage.humbug.php"/>
   <log type="coverage-text"
        target="/tmp/humbug/coverage.humbug.txt"/>
   <log type="junit" target="/tmp/humbug/junit.humbug.xml"/>
</logging>
```



```
Time: 8.54 seconds, Memory: 18.00MB

OK (488 tests, 1002 assertions)
```



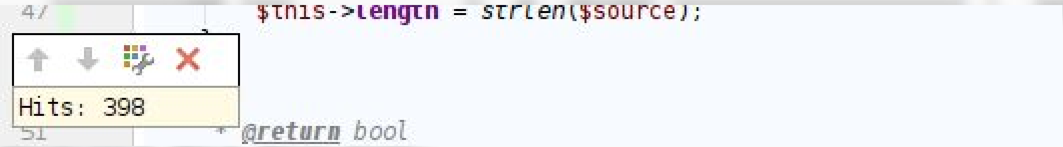
```
Time: 3.29 minutes Memory: 16.00MB
Humbug results are being logged as JSON to: ../../../../build/humbug/humbuglog.json
Humbug results are being logged as TEXT to: ../../../../build/humbug/humbuglog.txt
```



```
  0 [>----------------------------------------------------------] < 1 sec

  [Humbug\Exception\InvalidArgumentException]
  File does not exist: /tmp/humbug/coverage.humbug.php
```

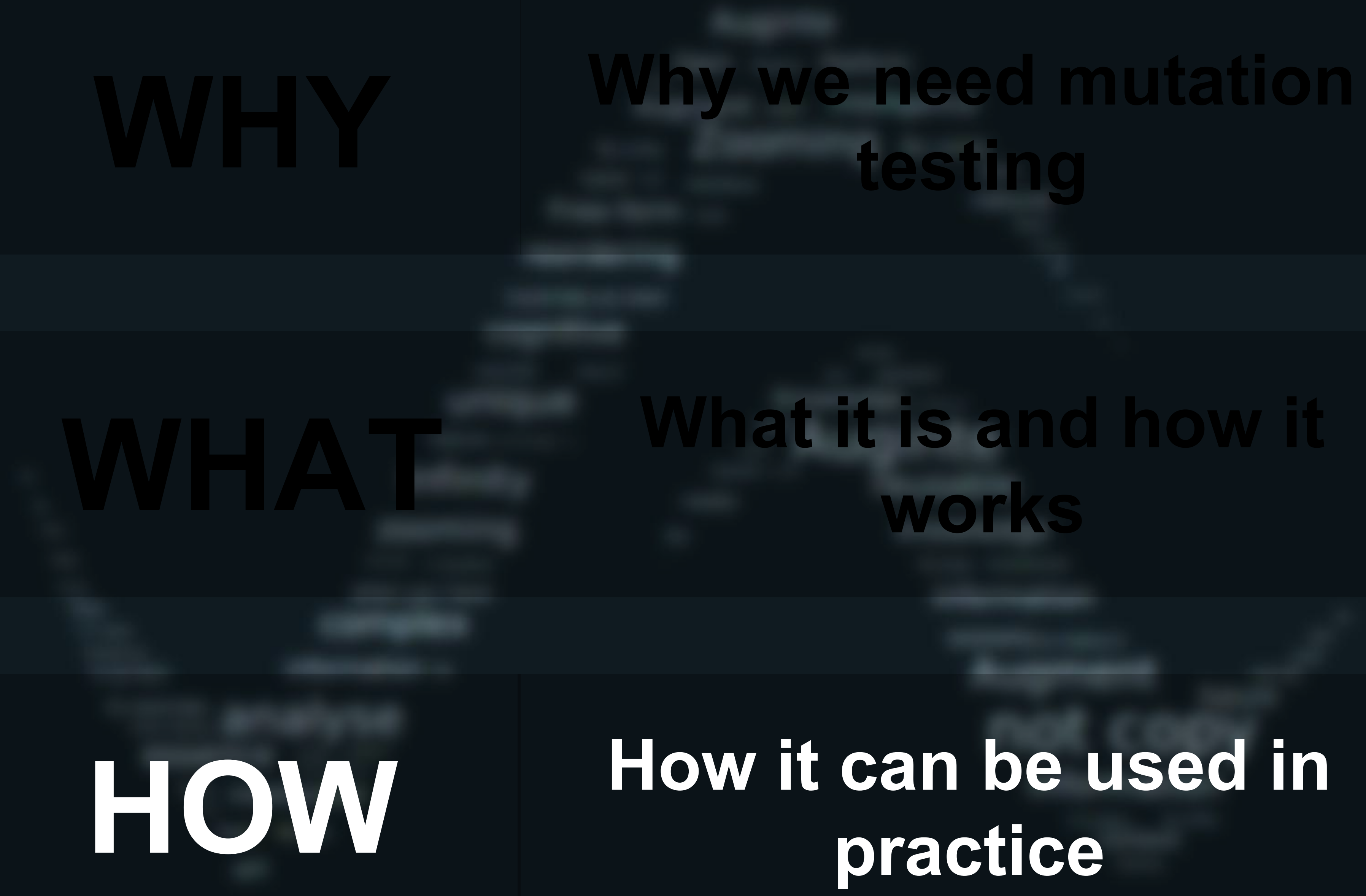
Drawbacks

- humbug/humbug 1.0.x-dev requires **phpunit/phpunit** ^4.5|^5.0
- phpunit/phpunit 5.7.5 requires php ^5.6 || ^7.0 -> **your PHP version** (5.5.9) does not satisfy that requirement.



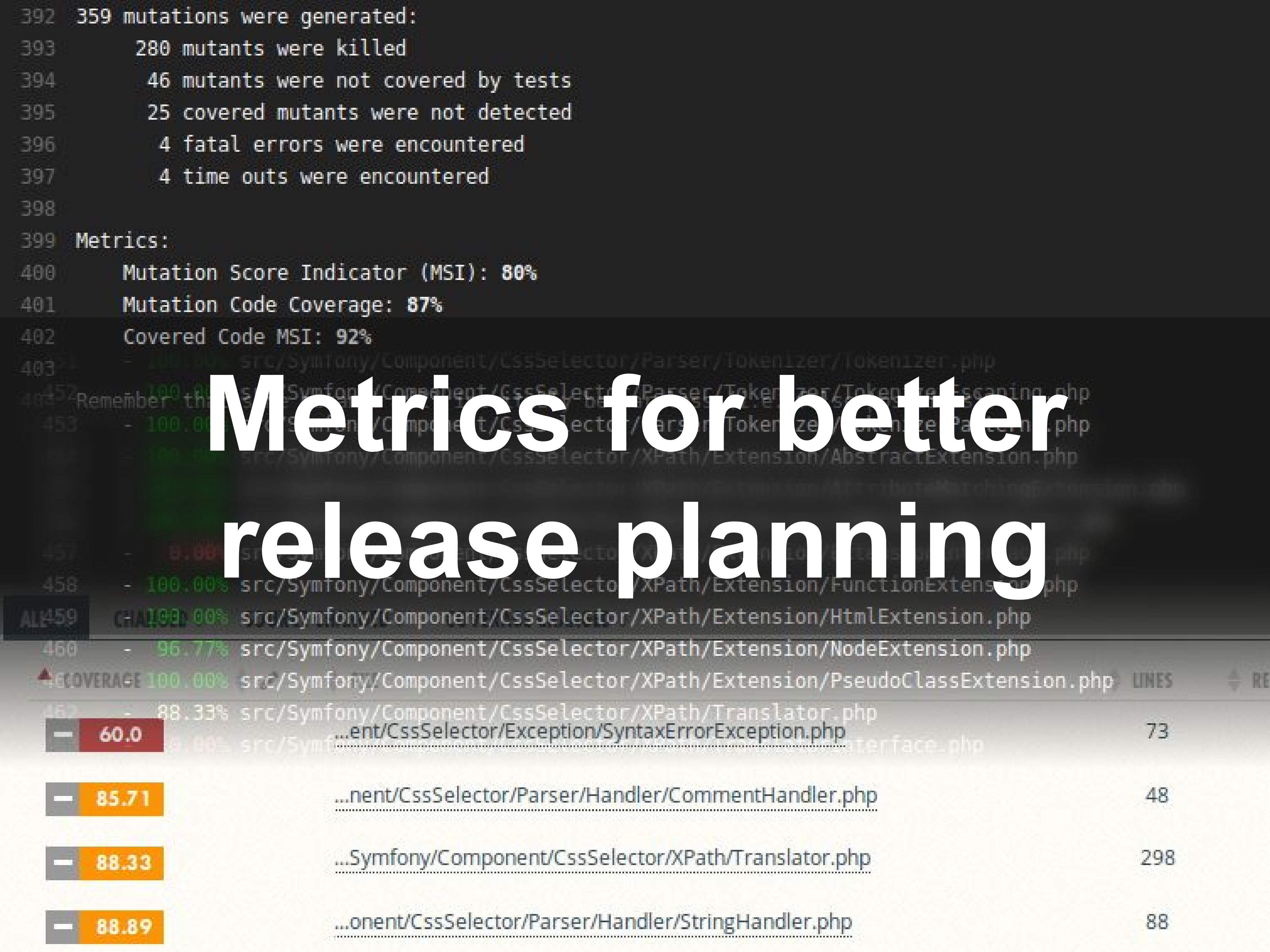
```
        $this->length = strlen($source);

Hits: 398

         * @return bool
```

WHY	Why we need mutation testing
WHAT	Testing framework, change code, run PHPUnit
HOW	How it can be used in practice



WHY	Why we need mutation testing
WHAT	What it is and how it works
HOW	How it can be used in practice

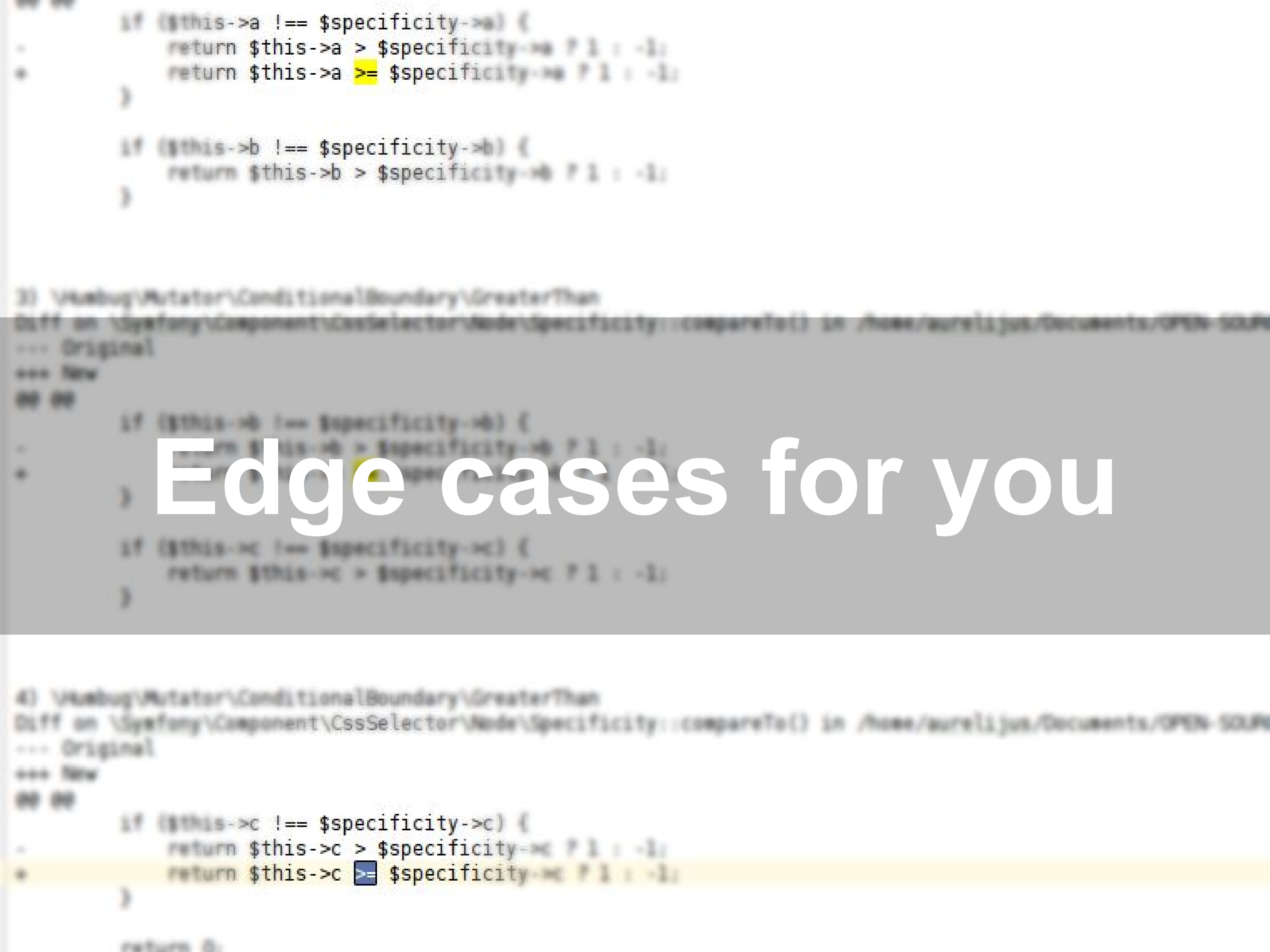
Metrics for better release planning



```
359 mutations were generated:
     280 mutants were killed
      46 mutants were not covered by tests
      25 covered mutants were not detected
       4 fatal errors were encountered
       4 time outs were encountered

Metrics:
    Mutation Score Indicator (MSI): 80%
    Mutation Code Coverage: 87%
    Covered Code MSI: 92%
```

```
    - 100.00% src/Symfony/Component/CssSelector/Parser/Tokenizer/Tokenizer.php
    - 100.00% src/Symfony/Component/CssSelector/Parser/Tokenizer/TokenizerEscaping.php
    - 100.00% src/Symfony/Component/CssSelector/Parser/Tokenizer/TokenizerPatterns.php
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/AbstractExtension.php
    -   0.00% src/Symfony/Component/CssSelector/XPath/Extension/
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/FunctionExtension.php
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/HtmlExtension.php
    -  96.77% src/Symfony/Component/CssSelector/XPath/Extension/NodeExtension.php
    - 100.00% src/Symfony/Component/CssSelector/XPath/Extension/PseudoClassExtension.php
    -  88.33% src/Symfony/Component/CssSelector/XPath/Translator.php
```

COVERAGE		LINES
60.0	...ent/CssSelector/Exception/SyntaxErrorException.php	73
85.71	...nent/CssSelector/Parser/Handler/CommentHandler.php	48
88.33	...Symfony/Component/CssSelector/XPath/Translator.php	298
88.89	...onent/CssSelector/Parser/Handler/StringHandler.php	88



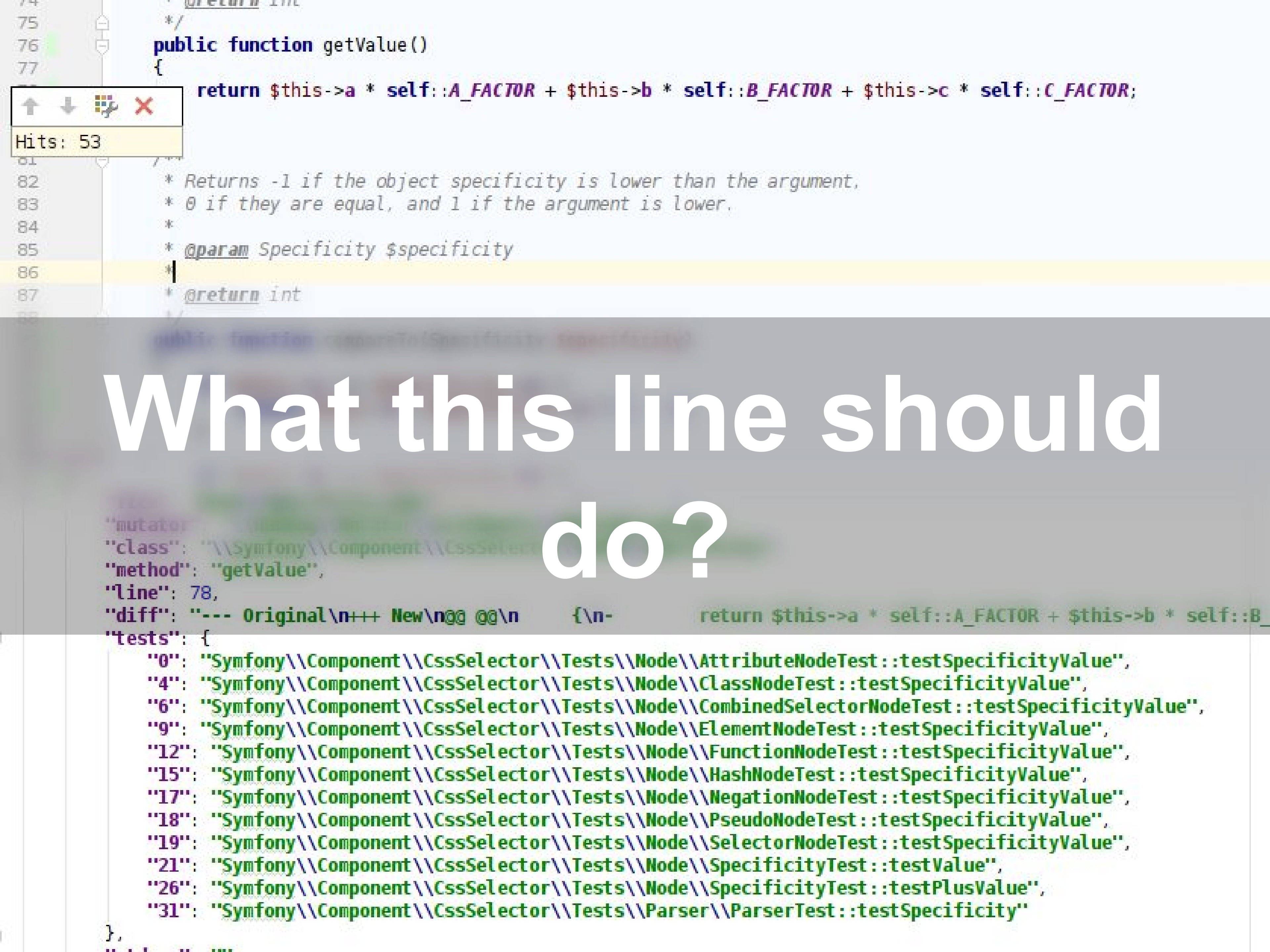
```
@@ @@
         if ($this->a !== $specificity->a) {
-            return $this->a > $specificity->a ? 1 : -1;
+            return $this->a >= $specificity->a ? 1 : -1;
         }

         if ($this->b !== $specificity->b) {
             return $this->b > $specificity->b ? 1 : -1;
         }


3) \Humbug\Mutator\ConditionalBoundary\GreaterThan
Diff on \Symfony\Component\CssSelector\Node\Specificity::compareTo() in /home/aurelijus/Documents/OPEN-SOURC
--- Original
+++ New
@@ @@
         if ($this->b !== $specificity->b) {
-            return $this->b > $specificity->b ? 1 : -1;
+            return $this->b >= $specificity->b ? 1 : -1;
         }

         if ($this->c !== $specificity->c) {
             return $this->c > $specificity->c ? 1 : -1;
         }


4) \Humbug\Mutator\ConditionalBoundary\GreaterThan
Diff on \Symfony\Component\CssSelector\Node\Specificity::compareTo() in /home/aurelijus/Documents/OPEN-SOURC
--- Original
+++ New
@@ @@
         if ($this->c !== $specificity->c) {
-            return $this->c > $specificity->c ? 1 : -1;
+            return $this->c >= $specificity->c ? 1 : -1;
         }

         return 0;
```

Edge cases for you

Quality for regression testing

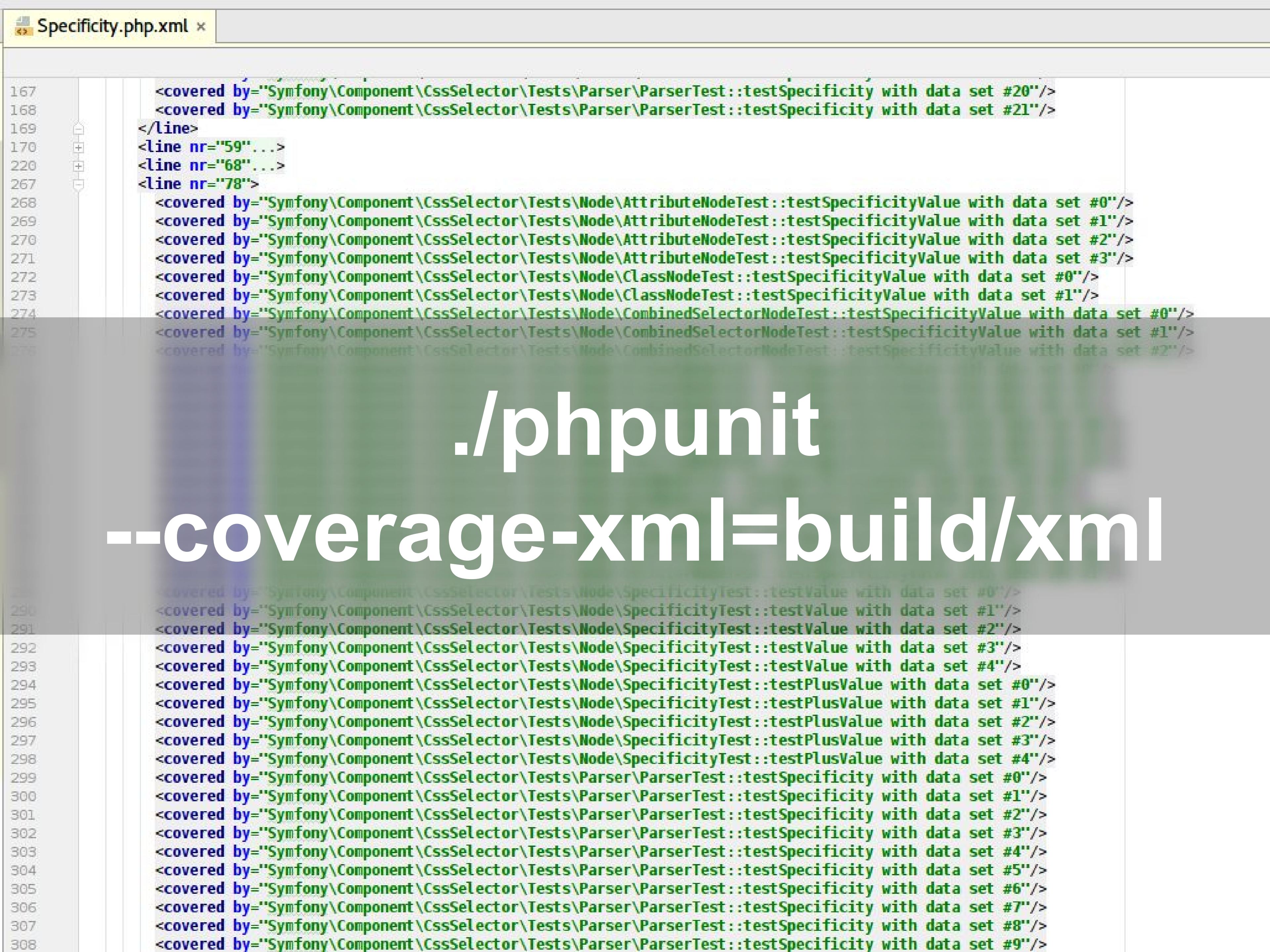
```
Diff on \Symfony\Component\CssSelector\Parser\Parser::parseSeries() in /home/aurelijus/Documents/OPEN-SOURCE/symfony/src
--- Original
+++ New
@@ @@
        $split = explode('n', $joined);
-       $first = isset($split[0]) ? $split[0] : null;
+       $first = isset($split[1]) ? $split[0] : null;

        return array(
            $first ? ('-' === $first || '+' === $first ? $int($first.'1') : $int($first)) : 1,
            isset($split[1]) && $split[1] ? $int($split[1]) : 0,
        );
    }


6) \Humbug\Mutator\Number\IntegerValue
Diff on \Symfony\Component\CssSelector\Parser\Parser::parseSeries() in /home/aurelijus/Documents/OPEN-SOURCE/symfony/src
--- Original
+++ New
@@ @@
            $first ? ('-' === $first || '+' === $first ? $int($first.'1') : $int($first)) : 1,
-           isset($split[1]) && $split[1] ? $int($split[1]) : 0,
+           isset($split[0]) && $split[1] ? $int($split[1]) : 0,
        );
    }

    /**
     * Parses selector nodes.
     *


7) \Humbug\Mutator\Boolean\LogicalAnd
Diff on \Symfony\Component\CssSelector\Parser\Parser::parseAttributeNode() in /home/aurelijus/Documents/OPEN-SOURCE/symf
--- Original
+++ New
@@ @@
            } elseif ($next->isDelimiter(array('^', '$', '*', '~', '|', '!'))
-               && $stream->getPeek()->isDelimiter(array('='))
+               || $stream->getPeek()->isDelimiter(array('='))
            ) {
                $operator = $next->getValue().'=';
                $stream->getNext();
```

What this line should do?



```
     * @return int
     */
    public function getValue()
    {
        return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;


    /**
     * Returns -1 if the object specificity is lower than the argument,
     * 0 if they are equal, and 1 if the argument is lower.
     *
     * @param Specificity $specificity
     *
     * @return int
     */
```

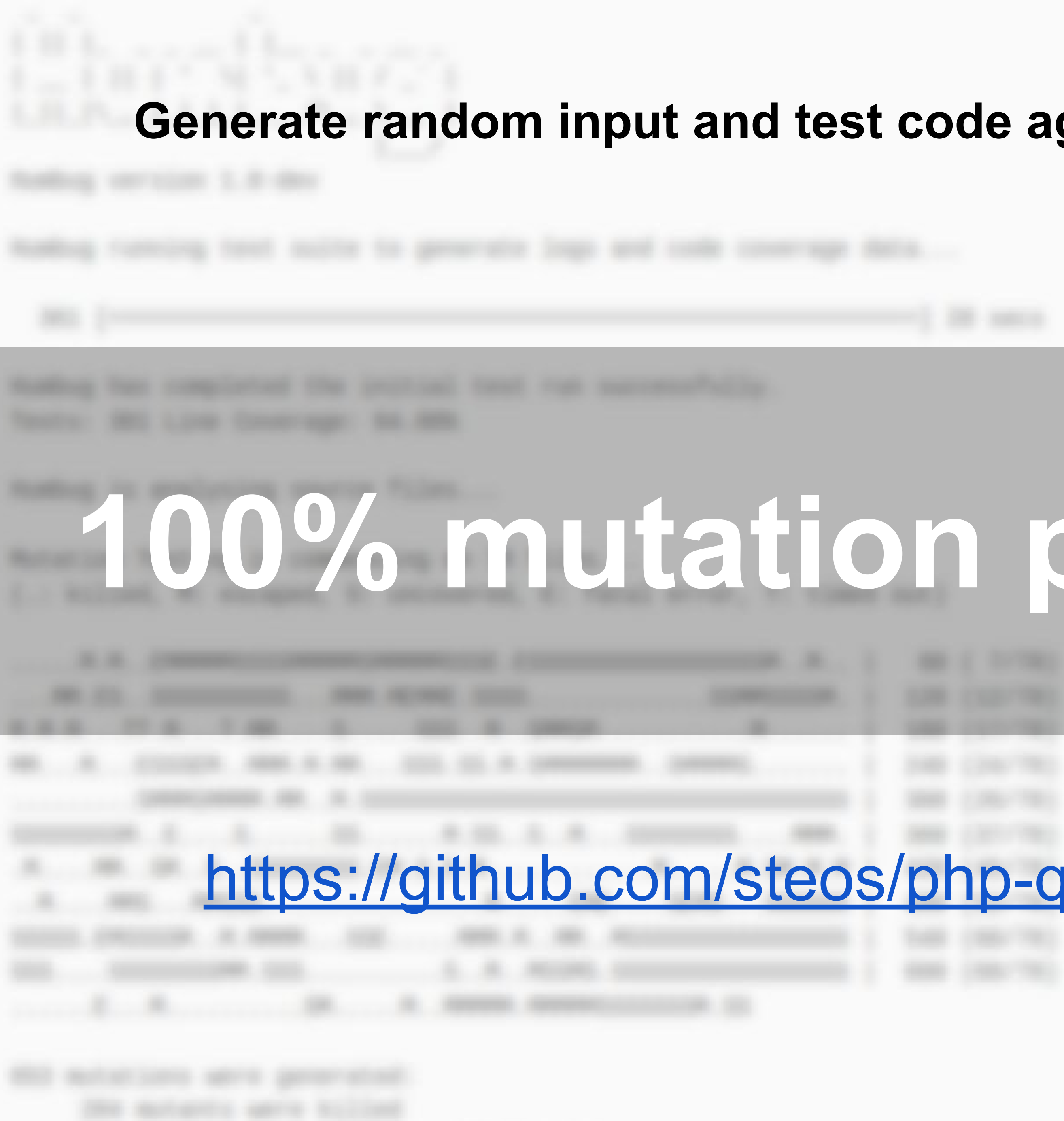
```
"mutator"
"class": "\\Symfony\\Component\\CssSelector
"method": "getValue",
"line": 78,
"diff": "--- Original\n+++ New\n@@ @@\n     {\n-        return $this->a * self::A_FACTOR + $this->b * self::B_
"tests": {
    "0": "Symfony\\Component\\CssSelector\\Tests\\Node\\AttributeNodeTest::testSpecificityValue",
    "4": "Symfony\\Component\\CssSelector\\Tests\\Node\\ClassNodeTest::testSpecificityValue",
    "6": "Symfony\\Component\\CssSelector\\Tests\\Node\\CombinedSelectorNodeTest::testSpecificityValue",
    "9": "Symfony\\Component\\CssSelector\\Tests\\Node\\ElementNodeTest::testSpecificityValue",
    "12": "Symfony\\Component\\CssSelector\\Tests\\Node\\FunctionNodeTest::testSpecificityValue",
    "15": "Symfony\\Component\\CssSelector\\Tests\\Node\\HashNodeTest::testSpecificityValue",
    "17": "Symfony\\Component\\CssSelector\\Tests\\Node\\NegationNodeTest::testSpecificityValue",
    "18": "Symfony\\Component\\CssSelector\\Tests\\Node\\PseudoNodeTest::testSpecificityValue",
    "19": "Symfony\\Component\\CssSelector\\Tests\\Node\\SelectorNodeTest::testSpecificityValue",
    "21": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testValue",
    "26": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testPlusValue",
    "31": "Symfony\\Component\\CssSelector\\Tests\\Parser\\ParserTest::testSpecificity"
},
```



```
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #20"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #21"/>
  </line>
  <line nr="59"...>
  <line nr="68"...>
  <line nr="78">
    <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #0"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #1"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #2"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #3"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\ClassNodeTest::testSpecificityValue with data set #0"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\ClassNodeTest::testSpecificityValue with data set #1"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\CombinedSelectorNodeTest::testSpecificityValue with data set #0"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\CombinedSelectorNodeTest::testSpecificityValue with data set #1"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\CombinedSelectorNodeTest::testSpecificityValue with data set #2"/>
```

./phpunit --coverage-xml=build/xml

```
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #0"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #1"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #2"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #3"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #4"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #0"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #1"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #2"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #3"/>
    <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #4"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #0"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #1"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #2"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #3"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #4"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #5"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #6"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #7"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #8"/>
    <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #9"/>
```



Generate random input and test code against predicate

100% mutation proof?

https://github.com/steos/php-quickcheck

WHY	Why we need mutation testing
WHAT	What it is and how it works
HOW	**Planning, test cases, documentation by tests**

References

- http://www.slideshare.net/AgileLietuva/vaidas-pilkauskas-and-tadas-erbinskas-can-you-trust-your-tests
- http://mnapoli.fr/code-coverage-reports-with-github-travis-and/
- https://coveralls.io
- https://phpunit.de/manual/current/en/code-coverage-analysis.html#code-coverage-analysis.edge-cases
- https://akrabat.com/global-installation-of-php-tools-with-composer/
- https://travis-ci.org/aurelijusb/symfony/branches
- https://github.com/steos/php-quickcheck

Questions?

Mutation testing

with PHP

2017-01-05 Aurelijus Banelis VilniusPHP 0x32