SO
YOU THINK
YOU CAN TEST?

HELLO

Luka Muzinic
@lmuzinic

Working in a remote team of three software engineers, able to offer outsourcing and consulting services, leadership of development teams and code reviews. Managing everything from application architecture to infrastructure. Delivering projects that are documented, covered with tests, with automated provisioning of local development virtual machines and production servers.

WHY WE NEED TESTING?

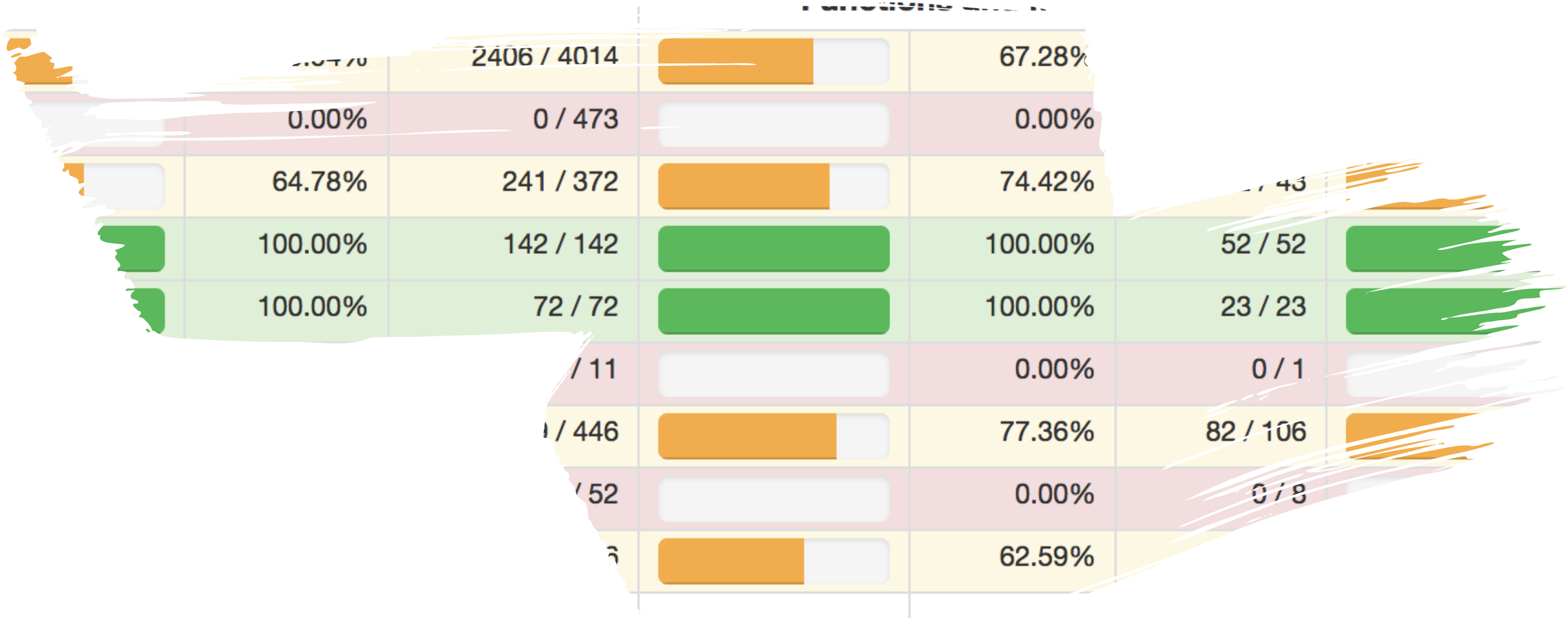
ARE WE SOFTWARE TESTERS?

AND YET
WE KEEP ON SAYING
WE WRITE TESTS…

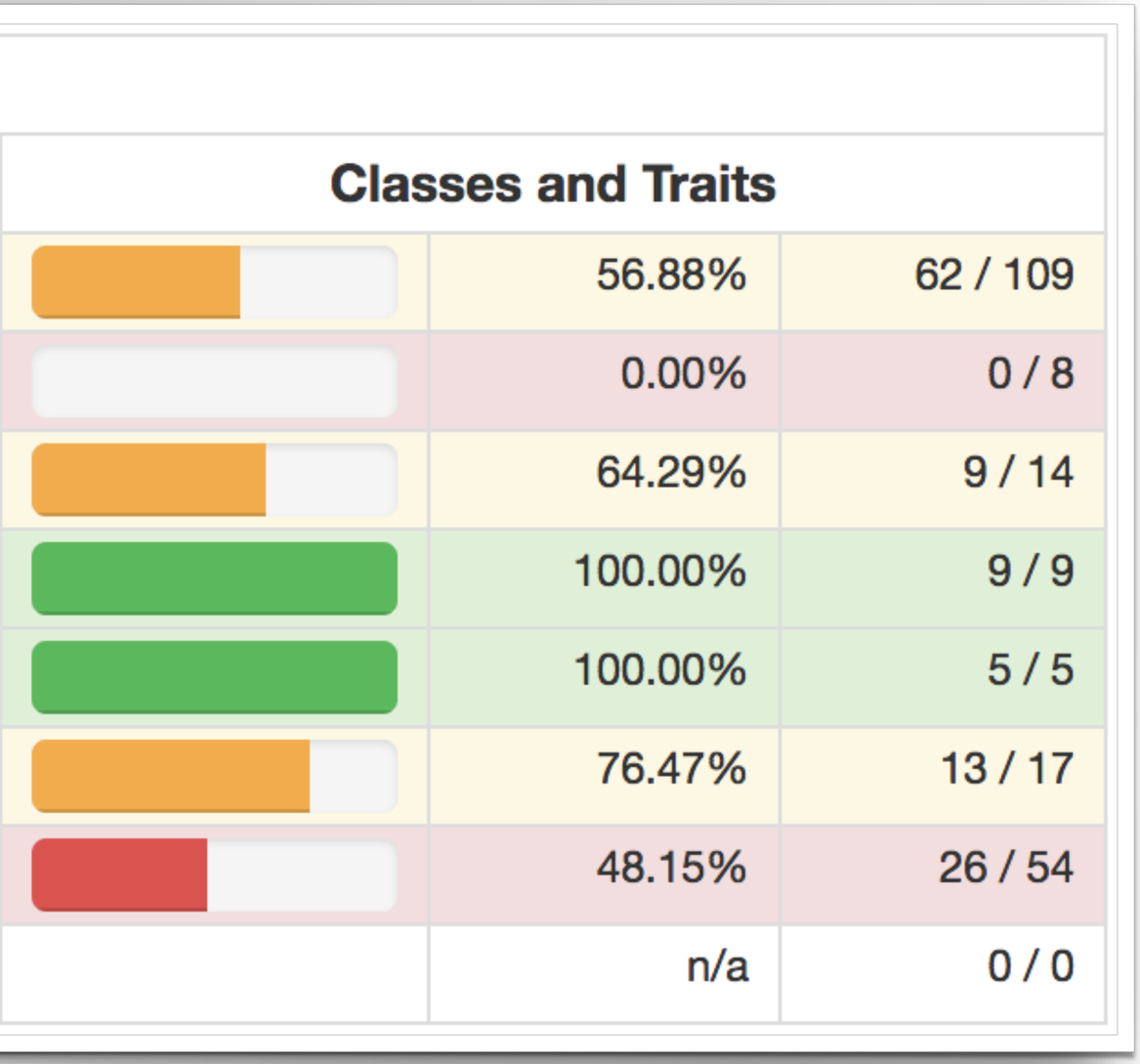
WHERE CAN I GET MORE OF THOSE TESTS?

HOW DO I KNOW HOW GOOD ARE MY TESTS?

CODE COVERAGE



CODE COVERAGE



Classes and Traits		
	56.88%	62 / 109
	0.00%	0 / 8
	64.29%	9 / 14
	100.00%	9 / 9
	100.00%	5 / 5
	76.47%	13 / 17
	48.15%	26 / 54
	n/a	0 / 0

CODE COVERAGE

Code Coverage					
Functions and Methods			**Classes and Traits**		
	67.28%	368 / 547		56.88%	62 / 109
	0.00%	0 / 28		0.00%	0 / 8
	74.42%	32 / 43		64.29%	9 / 14
	100.00%	52 / 52		100.00%	9 / 9
	100.00%	23 / 23		100.00%	5 / 5
	77.36%	82 / 106		76.47%	13 / 17
	62.59%	179 / 286		48.15%	26 / 54
	n/a	0 / 0		n/a	0 / 0

CODE COVERAGE

Code Coverage					
Lines			Functions and Methods		
59.94%	2406 / 4014		67.28%	368 / 547	
0.00%	0 / 473		0.00%	0 / 28	
64.78%	241 / 372		74.42%	32 / 43	
100.00%	142 / 142		100.00%	52 / 52	
100.00%	72 / 72		100.00%	23 / 23	
67.04%	299 / 446		77.36%	82 / 106	
67.54%	1652 / 2446		62.59%	179 / 286	
n/a	0 / 0		n/a	0 / 0	

CODE COVERAGE

	Lines			Cod / Functi
Total		59.94%	2406 / 4014	
Command		0.00%	0 / 473	
Controller		64.78%	241 / 372	
Entity		100.00%	142 / 142	
Model		100.00%	72 / 72	
Repository		67.04%	299 / 446	
Service		67.54%	1652 / 2446	
AppBundle.php		n/a	0 / 0	

20 > 80

MUTATION TESTING

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
public function
protected function

protected function
private function
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
$a + $b
$a - $b

$a * $b
$a / $b
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
true
false

&&
||
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
>
>=

<=
<
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
==
!=

>
<=
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
++
--

--
++
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
return 42;
return -42;

return $this;
return null;

return foo();
foo(); return null;
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

…

```
break;
continue;

foreach ($array as $element)
foreach ([] as $element)
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
$a <=> $b
$b <=> $a
```

MUTATORS

METHOD SIGNATURES

BINARY ARITHMETICS

BOOLEAN SUBSTITUTION

CONDITIONAL BOUNDARIES

NEGATED CONDITIONALS

INCREMENTS

RETURN VALUES

LOOPS

SORTING

LITERAL NUMBERS

...

```
0
1

1.0
0.0
```

MUTATIONS

User.php

Raw

```php
<?php

class User
{
    // ...

    public function isAdmin()
    {
        if ($this->status === self::ADMIN) {
           return true;
        }

        return false;
    }
}
```

MUTATIONS

User.php

```php
<?php

class User
{
    // ...

    public function isAdmin()
    {
        if ($this->status === self::ADMIN) {
            return true;
        }

        return false;
    }
}
```

UserFunction.php.diff

```
<?php

class User
{
    // ...

-   public function isAdmin()
+   protected function isAdmin()
    {
        if ($this->status === self::ADMIN) {
            return true;
        }

        return false;
    }
}
```

MUTATIONS

User.php

```php
<?php

class User
{
    // ...

    public function isAdmin()
    {
        if ($this->status === self::ADMI
            return true;
        }

        return false;
    }
}
```

UserFunction.php.diff

```
<?php

class User
{
    // ...

-   public function isAdmin()
+   protected function isAdmin()
    {
        if ($this->status === self::ADMIN) {
            return true;
        }
```

UserNegated.php.diff

```
<?php

class User
{
    // ...

    public function isAdmin()
    {
-       if ($this->status === self::ADMIN) {
+       if ($this->status !== self::ADMIN) {
            return true;
        }

```

MUTATIONS

User.php

```php
<?php

class User
{
    // ...

    public function isAdmin()
    {
        if ($this->status === self::ADMI
            return true;
        }

        return false;
    }
}
```

UserFunction.php.diff

```
<?php

class User
{
    // ...

-   public function isAdmin()
+   protected function isAdmin()
    {
        if ($this->status === self::ADMIN) {
            retur
        }
```

UserNegated.php.diff

```
<?php

class User
{
    // ...

    public function isAdmin()
    {
-       if ($this->status === self::ADMIN) {
+       if ($this->status !== self::ADMIN) {
            return true;
        }
```

UserReturn.php.diff

```
<?php

class User
{
    // ...

    public function isAdmin()
    {
        if ($this->status === self::ADMIN) {
-           return true;
+           return false;
        }

        return false;
    }
```

WHAT NOW?

RUN TESTS

KILLED

ESCAPED

UNCOVERED

FATAL ERROR

TIMEOUT

METRICS

MUTATION SCORE INDICATOR

```
$defeatedMutants / $totalMutants;
```

MUTATION CODE COVERAGE

```
$coveredMutants / $totalMutants;
```

COVERED CODE MUTATION SCORE INDICATOR

```
$defeatedMutants / $coveredMutants;
```

PROS & CONS

DAILY USAGE

CI

NEW INSIGHTS

FORGET 100%

SLOW

CAN I HAS?

```
Running initial test suite...

PHPUnit version: 5.7.2

   25 [============================] 2 secs

Generate mutants...

Processing source code files: 15/15Creating mutated files and processes:  0/71
Creating mutated files and processes: 71/71
.: killed, M: escaped, S: uncovered, E: fatal error, T: timed out

.......................ES....E.....EE....E..S...E   (50 / 71)
..SSEE..EEE..SS..EE..                               (71 / 71)

71 mutations were generated:
      52 mutants were killed
       6 mutants were not covered by tests
       0 covered mutants were not detected
      13 errors were encountered
       0 time outs were encountered

Metrics:
        Mutation Score Indicator (MSI): 92%
        Mutation Code Coverage: 92%
        Covered Code MSI: 100%

Please note that some mutants will inevitably be harmless (i.e. false positives).
```

PROPERTY BASED TESTING

RANDOMNESS EVERYWHERE

GENERATORS

SCALAR

COLLECTION

COMPOSITE

DOMAIN-BASED

GENERATORS

SCALAR

COLLECTION

COMPOSITE

DOMAIN-BASED

```php
public function testAllGoodNumbers()
{
    $this->forAll(
        Generator\nat(),
        Generator\pos()
    )
        ->then(function ($a, $b) {
            $this->assertTrue();
        });
}
```

GENERATORS

SCALAR

COLLECTION

COMPOSITE

DOMAIN-BASED

```
public function testAllGoodArrays()
{
    $this->forAll(
        Generator\associative([
            'natural' => Generator\nat(),
            'positive' => Generator\pos()
        ])
    )
        ->then(function (array $array) {
            $this->assertTrue();
        });
}
```

GENERATORS

SCALAR

COLLECTION

COMPOSITE

DOMAIN-BASED

```
public function testAllGoodVectors()
{
    $this->forAll(
        Generator\vector(
            2,
            Generator\suchThat(
                function ($n) {
                    return $n > 42;
                },
                Generator\choose(0, 50)
            )
        )
    ) // ...
}
```

GENERATORS

SCALAR

COLLECTION

COMPOSITE

DOMAIN-BASED

```php
public function testAllGoodDates()
{
    $this->forAll(
        Generator\date("1994-05-30", "now")
    )
        ->then(function (\DateTime $date) {
            $this->assertTrue();
        });
}
```

IT SHRINKS?

REPRODUCE RANDOMNESS

PROS & CONS

FORCES SPECIFICATION

DOCUMENTS HOW YOU TESTED

CI

HARD

AGAIN CI

SIMPLE SPECS LOOK LIKE COPY&PASTE

```
PHPUnit 5.7.21 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.2.2 with Xdebug 2.6.0
Configuration: /Users/lmuzinic/Development/freshleaves/hnbex/phpunit.xml.dist

............................                                      28 / 28 (100%)

Time: 6.05 seconds, Memory: 12.00MB

OK (28 tests, 31726 assertions)
```

QUESTIONS?

Luka Muzinic
@lmuzinic

luka.muzinic.net/talks

KTHXBAI

Photos by Les Anderson, Joshua Earle, Ian Espinosa and Tom Roberts on Unsplash