



PRAGMATIC TDD

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.

ROUGH SCHEDULE

09:30	Workshop part 1
11:00	Refreshments
11:15	Workshop part 2
12:45	Lunch

WORKSHOP RULES

ASK QUESTIONS

IF YOU STILL DO NOT UNDERSTAND, ASK QUESTIONS AGAIN

DISCUSS RIGHT NOW, DO NOT WAIT FOR THE “RIGHT MOMENT”

WHY WE NEED TESTING?

**WHY AM I
HERE?**

**ARE WE
SOFTWARE TESTERS?**

**AND YET
WE KEEP ON SAYING
WE WRITE TESTS...**

**WHERE CAN I GET
MORE OF THOSE
TESTS?**



EXCUSES, EXCUSES

TESTS SLOW US DOWN

WE WILL NEVER GET TO 100% CODE COVERAGE

WE DO NOT HAVE TIME TO LEARN TESTING, WE'RE TOO BUSY SHIPPING CODE

EXCUSES, EXCUSES

TESTS SLOW US DOWN?

STOP TESTFILEING

DO YOU OFTEN DO THIS?

~ `php test.php`

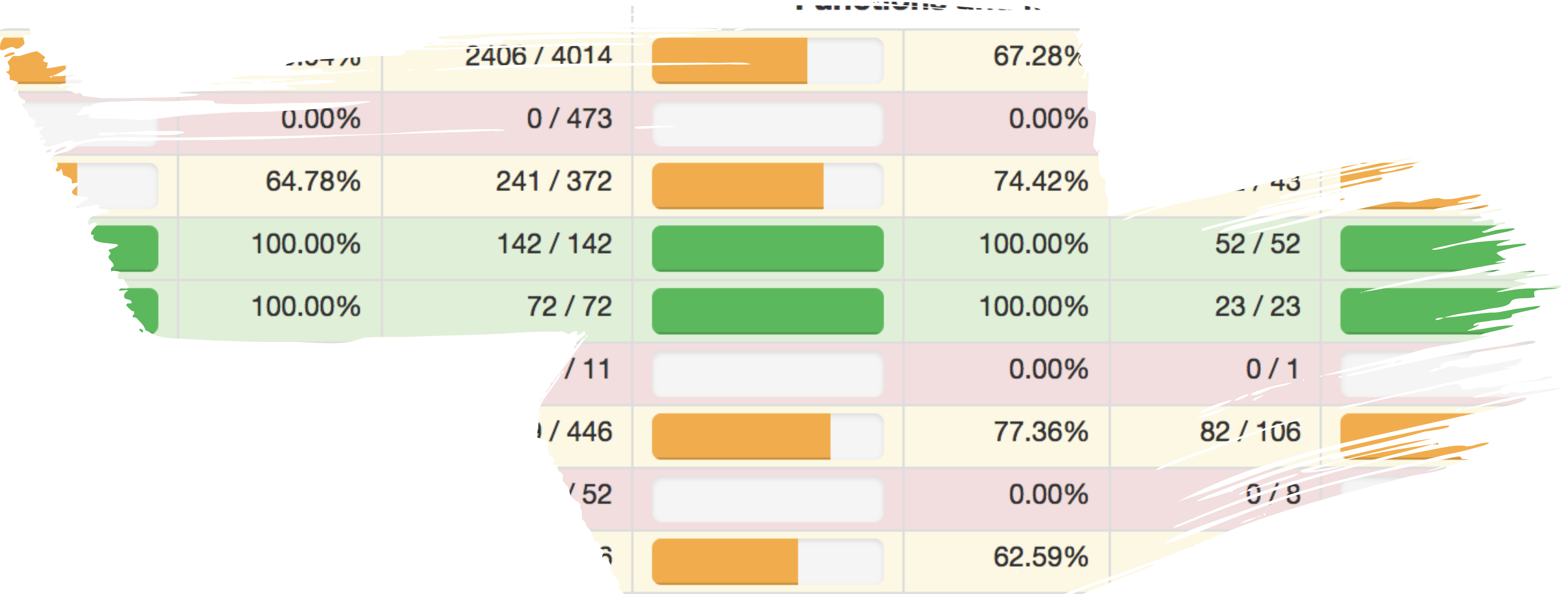
OR THIS

`http://localhost/test_problem.php`

EXCUSES, EXCUSES

**WE WILL NEVER GET TO 100%
CODE COVERAGE**

CODE COVERAGE



CODE COVERAGE

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








CODE COVERAGE

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

CODE COVERAGE

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

20 > 80

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

EXCUSES, EXCUSES

**WE DO NOT HAVE TIME TO LEARN TESTING,
WE'RE TOO BUSY SHIPPING CODE**

//@TODO: STANDSTILL

DON'T BE SCARED OF PHPUNIT*

IT IS JUST A CODE RUNNER

UNIT, INTEGRATION OR ACCEPTANCE TESTS

SMOKE TESTS

WEBSITE SCRAPER

...

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

DISCUSSION

IMPLEMENT A FOOTBALL LEAGUE MANAGEMENT APP

– We want to display standings table on our website

WHAT IS YOUR BIGGEST CONCERN?

HOW TO WRITE TESTS?

GIVEN WHEN THEN

SETUP EXERCISE VERIFY TEARDOWN

ANATOMY OF PHPUNIT TEST CASE

```
class TeamTest extends PHPUnit\Framework\TestCase;
{
    private $team;

    public function setUp()
    {
        $this->team = Team::create('York City');
    }

    public function testGetName()
    {
        $this->assertEquals('York City', $this->team->getName());
    }
}
```

WORKSHOP RULES

ASK QUESTIONS

IF YOU STILL DO NOT UNDERSTAND, ASK QUESTIONS AGAIN

DISCUSS RIGHT NOW, DO NOT WAIT FOR THE “RIGHT MOMENT”



EXERCISE 0

Checkout and do composer install

<https://github.com/lmuzinic/phpyorkshire-pragmatic-tdd>

```
git clone git@github.com:lmuzinic/phpyorkshire-pragmatic-tdd.git
cd phpyorkshire-pragmatic-tdd
composer install
vendor/bin/phpunit
```

OK (1 test, 1 assertion)

EXERCISE 1

WRITE A FUNCTIONAL TEST FOR OUR APP

- main goal of the test is to show how our code is supposed to work**
- let us play with with the design, try out how it feels**
- decide how the output is going to look**

EXERCISE 2

RUN TEST

- understand why it is failing, look at error messages

DECIDE WHAT IS CRITICAL TO WORK

- estimate if you have time to write tests now

WRITE A TEST FOR THAT

- create a separate test file as our focus is shifting

GOTO RUN TEST :)

EXERCISE 3

TIE IT ALL TOGETHER

- make all tests green

QUESTIONS? /r/AMA?



Luka Muzinic
[@lmuzinic](#)

luka.muzinic.net/talks

HOMEWORK

STUFF THAT WE MENTIONED IN WORKSHOP AND AFTER IN Q&A

Reading list

<https://www.devmynd.com/blog/five-factor-testing/>

<https://martinfowler.com/articles/practical-test-pyramid.html>

<https://dev.to/theobendixson/the-problem-that-unit-tests-solve-b2l>

<https://blog.liplex.de/testing-private-and-protected-methods-with-phpunit/>

Libraries

<https://github.com/sebastianbergmann/phpunit>

<https://github.com/phpspec/phpspec>

<https://github.com/Codeception/Codeception>

<https://github.com/phpstan/phpstan>

<https://github.com/infection/infection>

KTHXBAI

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