Thanks to Our Sponsors

AUTOMATTIC

Woo

RingCentral®

twilio

nexmo®

The Vonage® API Platform

OSMI

mailchimp

wayfair®

Stay Connected

PHP[TEK]
2019

Wifi

Network: LCC

PW: iluvmeeting

#phptek

WELCOME TO THE TESTING TRACK

EGO-CENTRIC INTRODUCION GOES HERE

TOOL AND TECHNIQUE COMPARISON

TOOLS

- I USE PHPUNIT

TOOLS

- BUT THAT IS NOT THE ONLY CHOICE YOU HAVE

TOOLS

GRUMPY CRITERIA

TOOLS

GRUMPY CRITERIA

1. MUST HAVE CLEAR DOCUMENTATION

TOOLS

GRUMPY CRITERIA

1. MUST HAVE CLEAR DOCUMENTATION
2. MUST RUN FROM THE COMMAND LINE

TOOLS

GRUMPY CRITERIA

1. MUST HAVE CLEAR DOCUMENTATION
2. MUST RUN FROM THE COMMAND LINE
3. MUST GENERATE CLEAR INDICATORS OF PASSES AND FAILURES

TOOLS

PHPUNIT

- THE STANDARD TO WHICH ALL PHP TESTING FRAMEWORKS ARE COMPARED TO

TOOLS

PHPUNIT

```php
function testAreWeDoneYet()
{
    $this->assertTrue($sessionCount == 7);
}
```

TOOLS

PHPUNIT

- NO PREFERENCE FOR DIRECTORY STRUCTURE
- EXTREMELY CONFIGURABLE

TOOLS

PHPUNIT

- USES OWN TEST DOUBLE FRAMEWORK
- SUPPORTS USE OF OTHER TEST DOUBLE FRAMEWORKS

TOOLS

PHPUNIT

- INTEGRATION WITH MANY OTHER TOOLS

TOOLS

ATOUM

TOOLS

ATOUM

```
$this->given($this->newTestedInstance)
    ->then
    ->string($this->testedInstance->getHiAtoum())
    ->isEqualTo('Hi atoum !')
```

TOOLS

ATOUM

- ASSERTION BASED
- DIRECTORY STRUCTURE ENFORCEMENT

TOOLS

ATOUM

- USES IT'S OWN TEST DOUBLING TOOLS

TOOLS

PHPSPEC

TOOLS

PHPSPEC

```php
namespace spec;

use PhpSpec\ObjectBehavior;

class MarkdownSpec extends ObjectBehavior
{
    function it_converts_plain_text_to_html_paragraphs()
    {
        $this->toHtml("Hi, there")->shouldReturn("<p>Hi, there</p>");
    }
}
```

TOOLS

PHPSPEC

- USES SPECIFICATIONS AND CODE GENERATION
- ASSERTION BASED

TOOLS

PHPSPEC

- USES PROPHECY LIBRARY FOR CREATING DOUBLES

TOOLS

KAHLAN

TOOLS

KAHLAN

```
describe("MVT Monkey Patching", function () {
    it('should patch PDO', function() {
        $pdo = Stub::classname();
        Monkey::patch('PDO', $pdo);
        Stub::on($pdo)->method('prepare', function() {
            $statement = Stub::create();
            Stub::on($statement)->method('fetchAll', function() {
                return [['name' => 'bob']];
            });
            return $statement;
        });
        $user = new User();
        expect($user->getAll())->toBe([
            ['name' => 'bob']
        ]);
    });
});
```

TOOLS

KAHLAN

- `describe-it` SYNTAX
- REQUIRES A SPECIFIC DIRECTORY STRUCTURE

TOOLS

KAHLAN

- USES IT'S OWN TEST DOUBLE FRAMEWORK
  - ALLOWS "MONKEYPATCHING" OF CODE

TOOLS

MONKEYPATCHING

```
it("shows some examples of function stubbing", function() {
    allow('PDO')->toReceive('prepare->fetchAll')->andReturn([['name' => 'bob']]);
    allow('PDO')->toReceive('prepare->fetchAll')->andRun(function() {
        return [['name' => 'bob']];
    });
});
```

TOOLS

MONKEYPATCHING

"DYNAMIC MODIFICATIONS OF A CLASS OR MODULE AT RUNTIME"

TOOLS

MONKEYPATCHING

"CREATE DOUBLES OF CORE PHP FUNCTIONS AND/OR OBJECTS"

TOOLS

MONKEYPATCHING

- DATE AND TIME FUNCTIONALITY
- DATABASE CONNECTION FUNCTIONS

TOOLS

CODECEPTION

TOOLS

CODECEPTION

```php
class FirstCest
{
    public function frontpageWorks(AcceptanceTester $I)
    {
        $I->amOnPage('/');
        $I->see('Home');
    }
}
```

TOOLS

CODECEPTION

- EMPHASIS ON BROWSER-BASED TESTING

TOOLS

BEHAT

TOOLS

BEHAT

```
Feature: Product basket
  In order to buy products
  As a customer
  I need to be able to put interesting products into a basket

  Rules:
  - VAT is 20%
  - Delivery for basket under £10 is £3
  - Delivery for basket over £10 is £2

  Scenario: Buying a single product under £10
    Given there is a "Sith Lord Lightsaber", which costs £5
    When I add the "Sith Lord Lightsaber" to the basket
    Then I should have 1 product in the basket
    And the overall basket price should be £9
```

TOOLS

BEHAT

```
/**
 * @Given /there is an? \"([^\"]+)\", which costs £([\d\.]+)/
 */
public function thereIsAWhichCostsPs($arg1, $arg2)
{
    throw new PendingException();
}
```

TOOLS

BEHAT

- COMBINATION OF SPECIFICATIONS AND CODE GENERATION
- CREATES SKELETONS FOR TESTS

TOOLS

BEHAT

- FOCUSED ON USING BUSINESS LOGIC AND A DSL TO DRIVE TESTS
- UNDERLYING PRINCIPLE OF BEHAVIOUR DRIVEN DESIGN

TECHNIQUES

TECHNIQUES

1. MANUAL

TECHNIQUES

1. MANUAL
2. SCRIPT RUN BY A PERSON

TECHNIQUES

1. MANUAL
2. SCRIPT RUN BY A PERSON
3. SCRIPT RUN BY A COMPUTER

TECHNIQUES

1. MANUAL
2. SCRIPT RUN BY A PERSON
3. SCRIPT RUN BY A COMPUTER ON BEHALF OF A HUMAN
4. SCRIPT RUN BY A COMPUTER ON BEHALF OF A COMPUTER

SLIDES

HTTPS://GRUMPY-LEARNING.COM/PRESENTATIONS/TEK2019