

Testing WordPress Waters

A quick dive into integration testing

COQUARD Cyrille



WordCamp
Porto 2024

Software engineer

At WP Media since 2 years.



IMAGIFY



Mail: cyrille@wp-media.me
Medium: [@crochetfeve0251](https://medium.com/@crochetfeve0251)



WordCamp
Porto 2024



Always improve

Developer eXperience

- Machine should serve the developer
- Code doesn't have to be complex

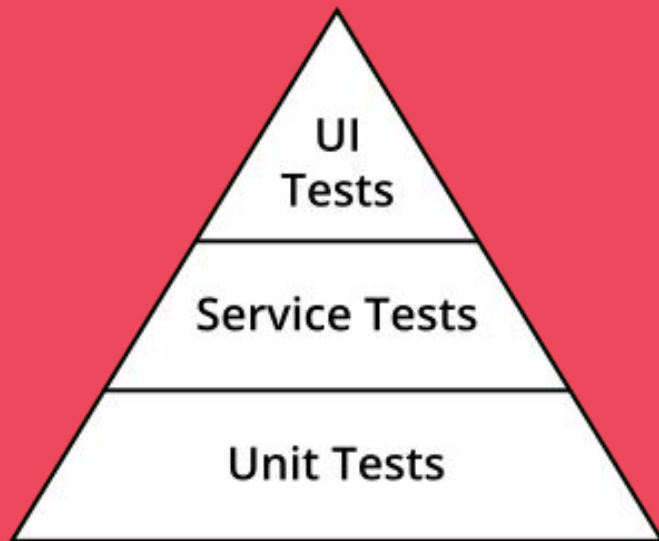
Testing

- Always interested in testing
- Started testing for WordPress 2 years ago



What is testing?

- Stands for automated tests
- Code to assert application behavior
- Multiple type of tests



Why this speak?

In one quote...



“Testing is limited.”

- J.B., WP Media CEO 2022

“Testing should be an OKR.”

- J.B., WP Media CEO 2024

Effective testing is hard

- Know what to test
- Know how to test
- Know when to test
- Know why testing

Bad test

=

Useless



WordCamp
Porto 2024



Effective testing is high reward

Tests are your copilot

- Low number of bugs
- Delivering on time

Objective:

Guide you toward effective test



WordCamp
Porto 2024



Summary

- Why testing
- Unit vs Integration
- How to test
- Usual testing scenarios

Unit vs Integration

Unit: academic way

- Class or method level
- More isolated
- Developer oriented
- Easy to set up
- First ones to learn



Integration: academic way

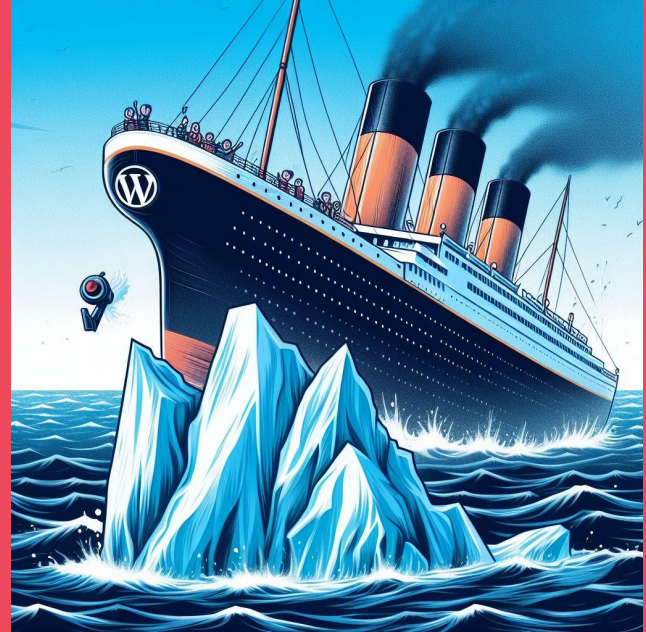
- Feature level
- More abstract
- Business oriented
- Complex to set up



What is the issue?

We ain't at school

- CEO / Manager decides
- Rentability / on schedule is key
- Need to convince



Unit: real world way

- More fragile so low productivity
- Developer language
- Lot of tests
- Need to create testcases



Integration: real world way

- More abstract so high productivity
- Business language
- Reduced number of tests
- Based on project definition



Start with integration

- Higher productivity
- Business language
- Reduced number of tests
- Ensure the feature is done



WordCamp
Porto 2024



Convince your management

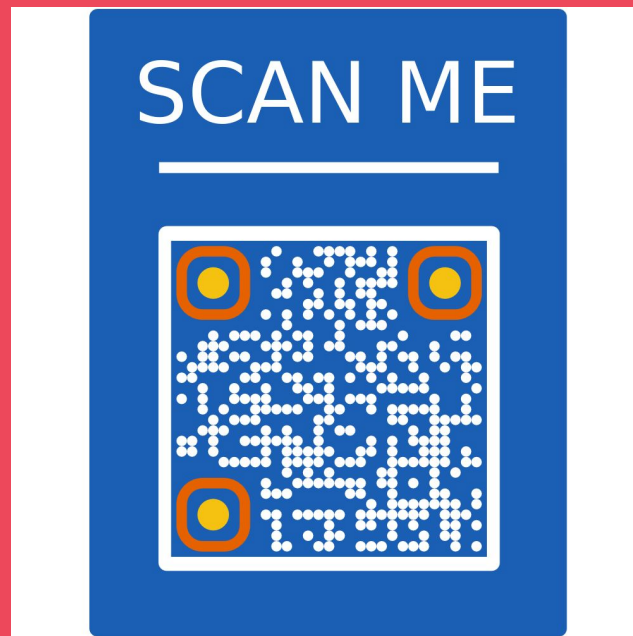
- Limit the risk
- Talk their language
- Highlight potential gains
- Learn to wait
- Make baby steps



How to test?

Setup the environment

- Setup a development env
- Setup integration tests
- Ease filter mocking
- Create the base for integration tests



Now what?



What to test?

Asset you made the task
not work done

- Testing your code: **Useless**
- Testing expectations: **Useful**



WordCamp
Porto 2024



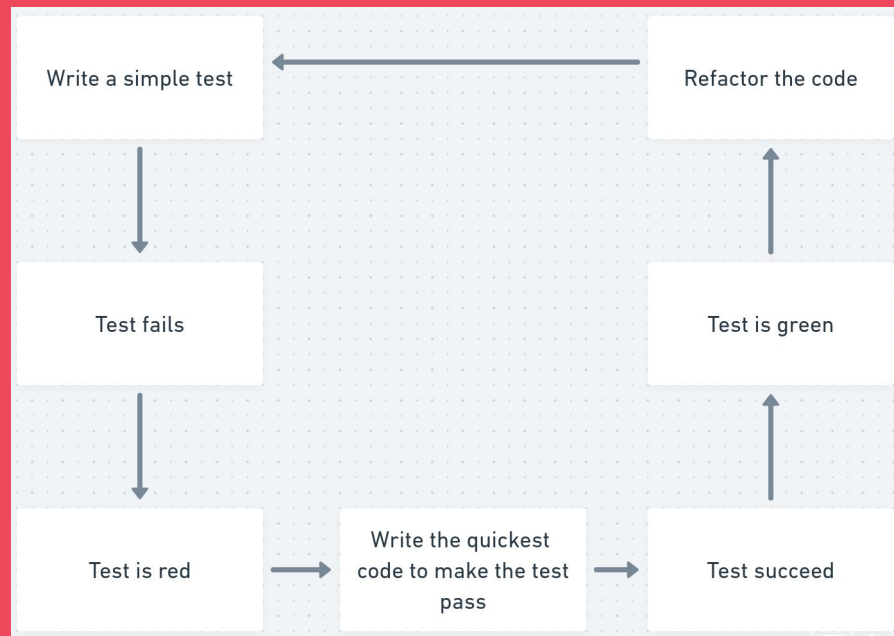
Acceptance criteria

- Simple sentence asserting a behavior
- Done by the person scoping the project
- Ensure you are not testing your code

```
1 - Image is preloaded using the image URLs
2 from srcset attribute and becomes imagesrcset
3
4 - sizes="50vw" attribute is moved to the
5 preload markup and becomes imagesizes="50vw"
6
7 - fetchpriority="high" is added to the
8 preload markup
9
10 - Image is excluded from LL feature
```


Test Driven development

- Really trendy
- Really dogmatic
- Better done than perfect



Rome didn't built in one day

- Learn gradually
- Don't follow strictly doctrines
- You need to understand



WordCamp
Porto 2024



TDD essentials

- Write test before code
- Make sure it failed before writing code
- Make sure all tests pass after writing code



AAA

- **Arrange:** put the plugin in a certain state
- **Act:** run the logic to test
- **Assert:** verify the plugin is in the expected state

```
1 <?php
2
3 class Test_Logout extends TestCase {
4     public function testLogoutShouldNotEdit()
5     {
6         do_action(
7             'grant_admin',
8             wp_current_user()
9         );
10
11        do_action('logout');
12
13        $this->assertSame(
14            false,
15            apply_filters(
16                'user_can_edit',
17                false
18            )
19        );
20
21    }
22 }
```



WordCamp
Porto 2024



Gerkhin

- **Arrange:** Given
- **Act:** When
- **Assert:** Then

```
1 GIVEN image with srcset "http://my-image.example/img.png"
2 WHEN the image is preloaded
3 THEN image have imagesrcset "http://my-image.example/img.png"
4 AND no srcset
5
6 GIVEN image with sizes "50vw"
7 WHEN the image is preloaded
8 THEN image have imagesizes "50vw"
9 AND no sizes
10
11 GIVEN an image
12 WHEN the image is preloaded
13 THEN image have fetchpriority "high"
14
15 GIVEN an image
16 WHEN the image is preloaded
17 THEN image is excluded from lazyload
```

Usual testing scenarios

Scenarios

- Unleash CQRS
- Control filter value
- Mock external API
- Isolate callbacks



WordCamp
Porto 2024



Unleash CQRS

Problem

Need to re-implement a way to interact with the elements for each test



WordCamp
Porto 2024



Unleash CQRS

Solution

Use actions and filters to cut your logic into small reusable units

```
1 <?php
2
3 class Test_Logout extends TestCase {
4     public function testLogoutShouldNotEdit()
5     {
6         do_action(
7             'grant_admin',
8             wp_current_user()
9         );
10
11        do_action('logout');
12
13        $this->assertSame(
14            false,
15            apply_filters(
16                'user_can_edit',
17                false
18            )
19        );
20
21    }
22 }
```

Control filter value

Problem

Need to control the flow to direct
the test into a certain state



WordCamp
Porto 2024



Control filter value

Solution

Mock the filter with a callback function

```
1 <?php
2
3 class Test_Logout extends TestCase {
4     public function testLogoutShouldNotEdit()
5     {
6         ...
7     }
8
9     /**
10     * @hook my_filter 10
11     */
12     public function my_callback() {
13         return true;
14     }
15 }
```

With `wp-launchpad/phpunit-wp-hooks`



WordCamp
Porto 2024



Mock external API call

Problem

The code call an external API
and there is no way to control it



WordCamp
Porto 2024



Mock external API call

Solution

Use `wp_remote_request` and mock the filter `pre_http_request` with a callback function

```
1 <?php
2
3 class Test_Logout extends TestCase {
4     public function testLogoutShouldNotEdit()
5     {
6         ...
7     }
8
9     /**
10    * @hook pre_http_request 10
11    */
12    public function my_callback($response, $args, $url) {
13
14        if (
15            strpos(
16                $url,
17                'https://app.imagify.io' ) === false
18        ) {
19            return $response;
20        }
21
22        return [
23            'body' => $message,
24            'response' => ['code' => 200 ]
25        ];
26    }
27 }
```

With `wp-launchpad/phpunit-wp-hooks`



WordCamp
Porto 2024



Isolate callback

Problem

The callback to test is linked to a hook with another callback messing with the state to assert



WordCamp
Porto 2024



Isolate callback

Solution

Isolate the callback by removing
all other callbacks

```
1 <?php
2
3 class Test_Logout extends TestCase {
4     /**
5      * @isolate-hook my-event myCallback 15
6      */
7     public function testLogoutShouldNotEdit()
8     {
9         // ...
10    }
11 }
```

With `wp-launchpad/phpunit-wp-hooks`



WordCamp
Porto 2024





Launchpad

All good practices from WP
Media in one framework.

