Launchpad: a bunch of accessible good practices

COQUARD Cyrille

Who is using?

- Composer

Who is using?

- Composer
- Namespaces

Who is using?

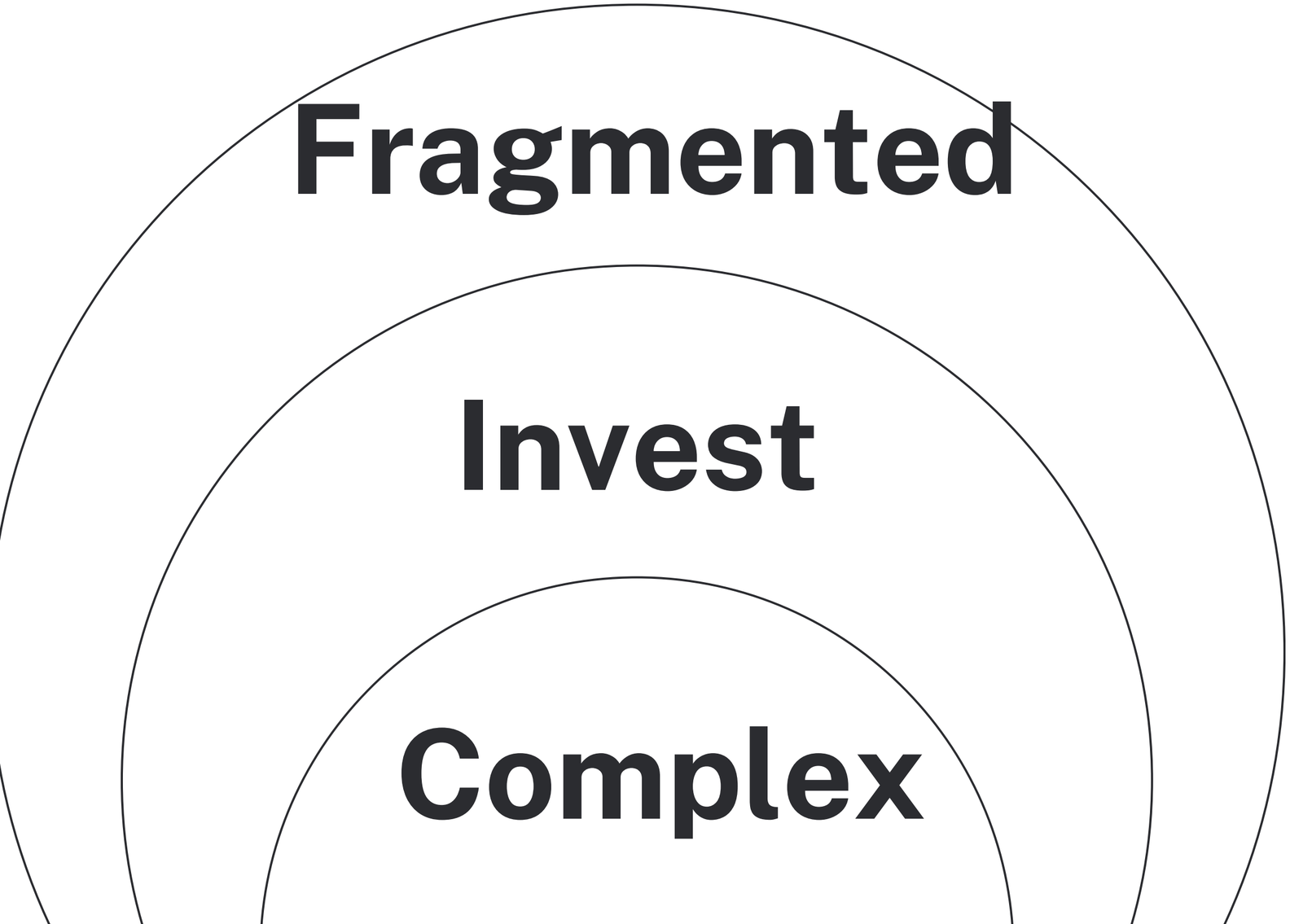
- Composer
- Namespaces
- Dependency injection

The garden is always greener

- Good practices are **the base**
- **Not the case** in WordPress

You are not bad, your tools are broken

- WordPress is end user centered
- Plugin devs are left aside



Layers of problems

Fragmented

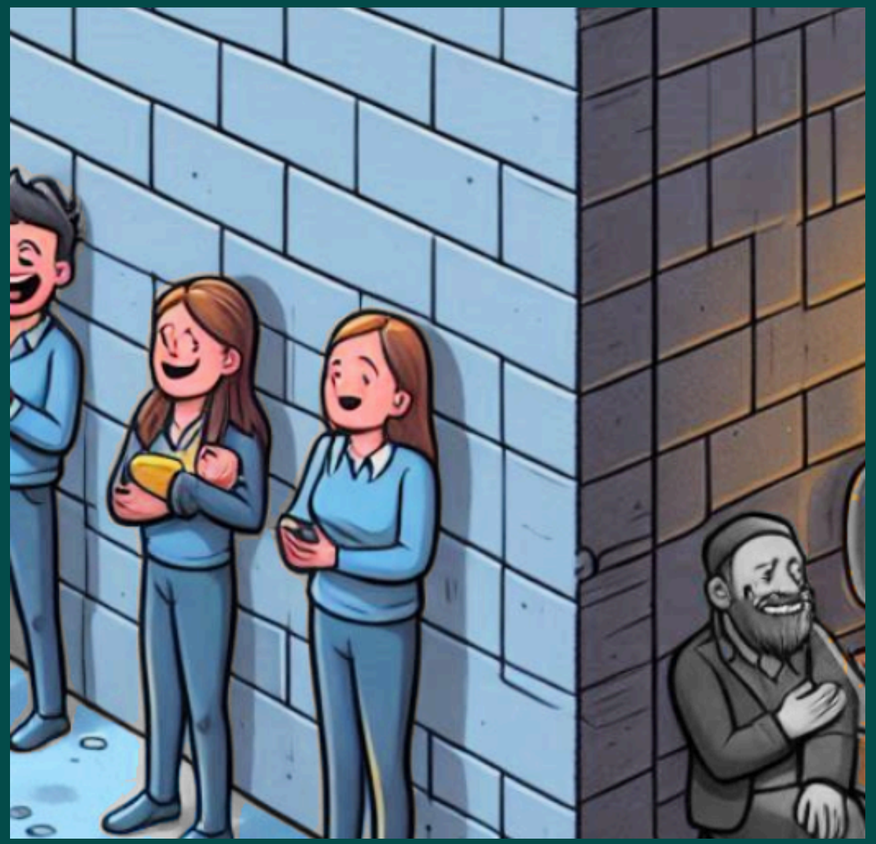
No centralized documentation

Invest

Initial investment

Complex

Knowing all is a requirement



Good practices is reserved to an elite

- Each problem **excluded developers**
- Elite is created **by excluding**

WordPress is inclusive

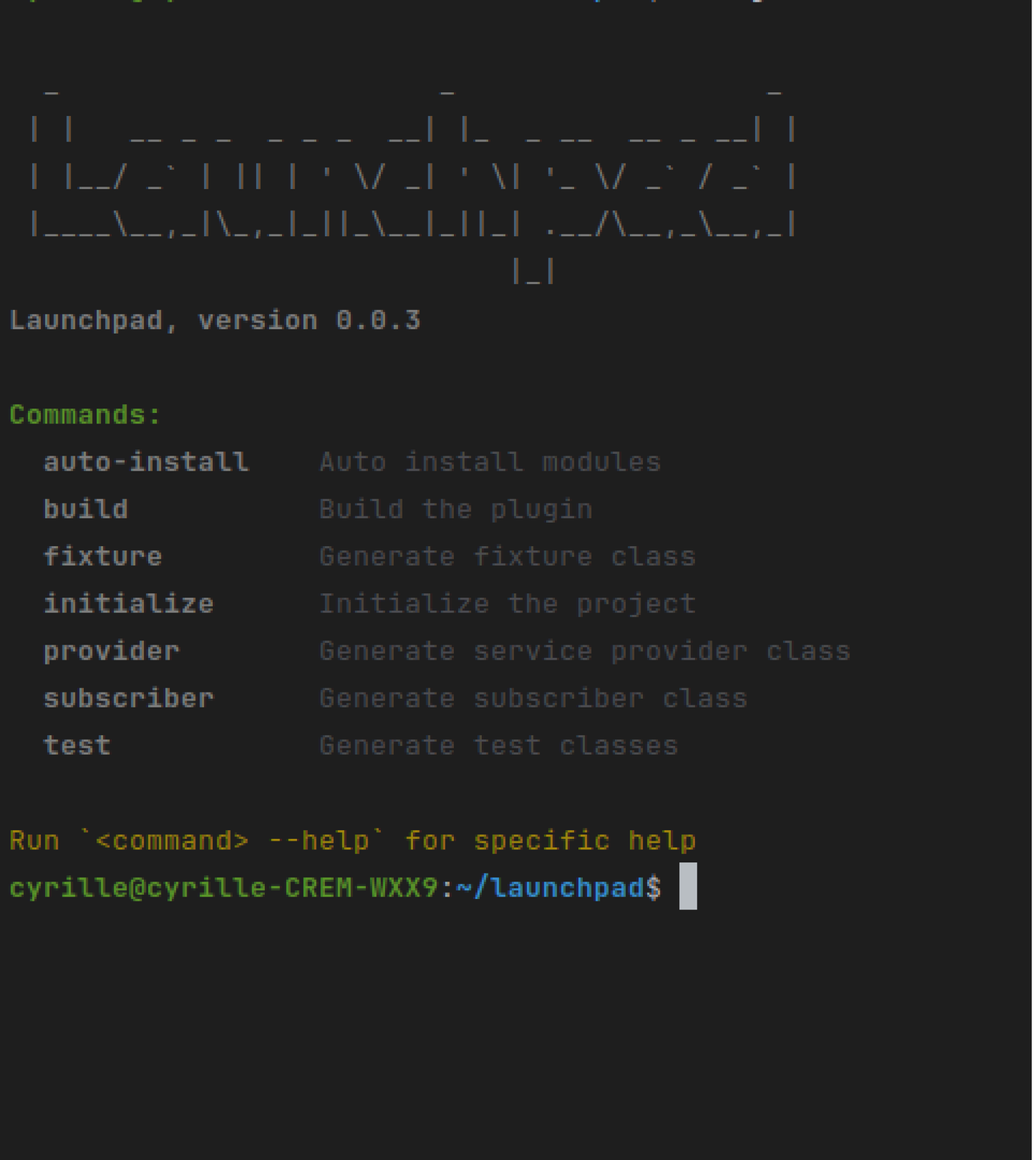
- **Excluding** is an issue
- **Inclusivity** is a main value

L A U N C H P A D

Lower the requirements to make good practices accessible.

Methods :

- **Abstract maximum of notions**
- **Offer a base**
- **Document notions**



INVEST

Provide a base:

- Up to date
- 2 commands to start

```php
namespace MonPlugin;



class ServiceProvider extends AbstractServiceProvider
{
    no usages
    public function get_subscribers(): array {
        return [
            Subscriber::class,
        ];
    }
}
```

COMPLEXITY

Minimize entrance barrier:

- Provider
- Subscriber

modification is added to that file.

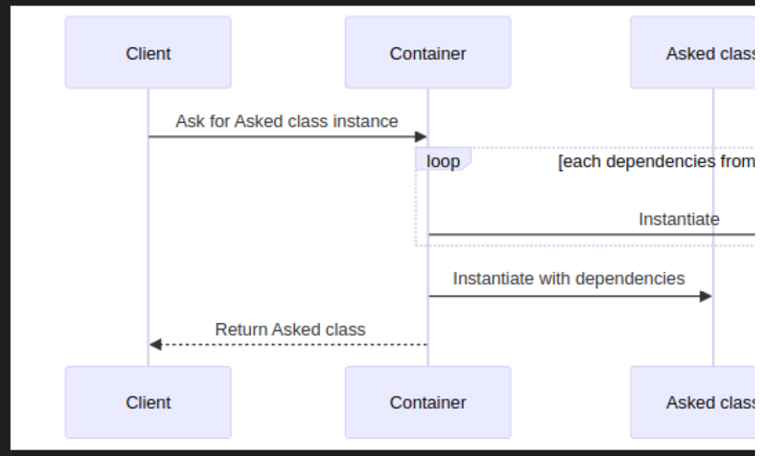
The second solution is to add classes wiring inside the constructor for t

This prevents conflicts as now classes organization is now split betwee
main drawback is that now it is really hard to mock classes as it is not a
classes.

Inversion of dependencies enters here. This solution provides us both b
drawbacks.

Solution	Can test	Not conflict prone
Wiring inside the core	Yes	No
Wiring inside constructors	No	Yes
Inversion of dependencies	Yes	Yes

The inversion of dependencies is based on wiring inside the core but in
rely on a container to make the wiring between classes.



This way the wiring logic is not anymore done by the core from the soft
To tackle this issue two solution can be picked with each one their draw

Rely on the reflection to make the wiring which makes the wiring disapp
slow and memory intensive.

Create another layer of abstraction to break down the wiring per feature

FRAGMENTED

Make progress linear:

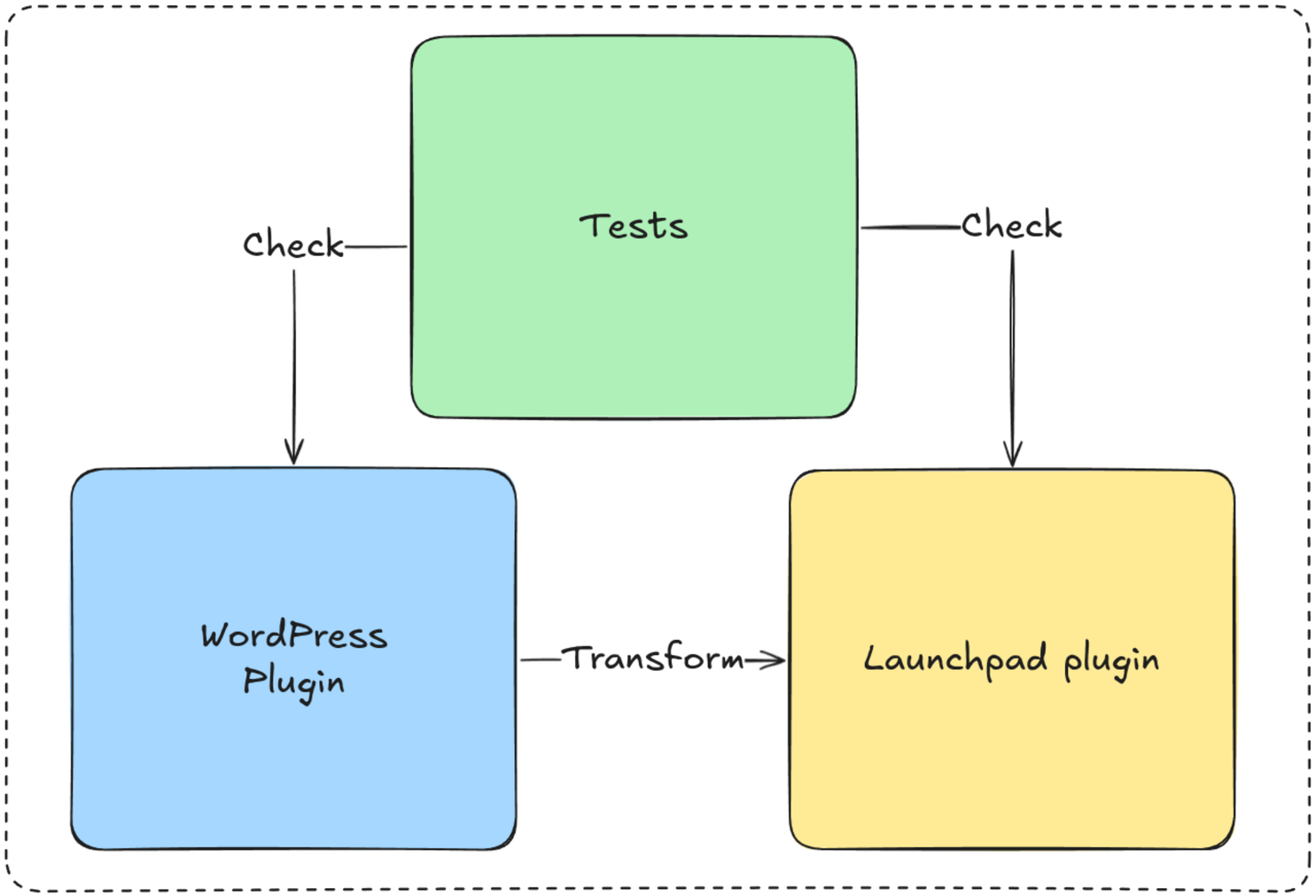
- Arrange by level
- Play with curiosity

OK, but what in ***practice*****?**

The best way to learn: Practice

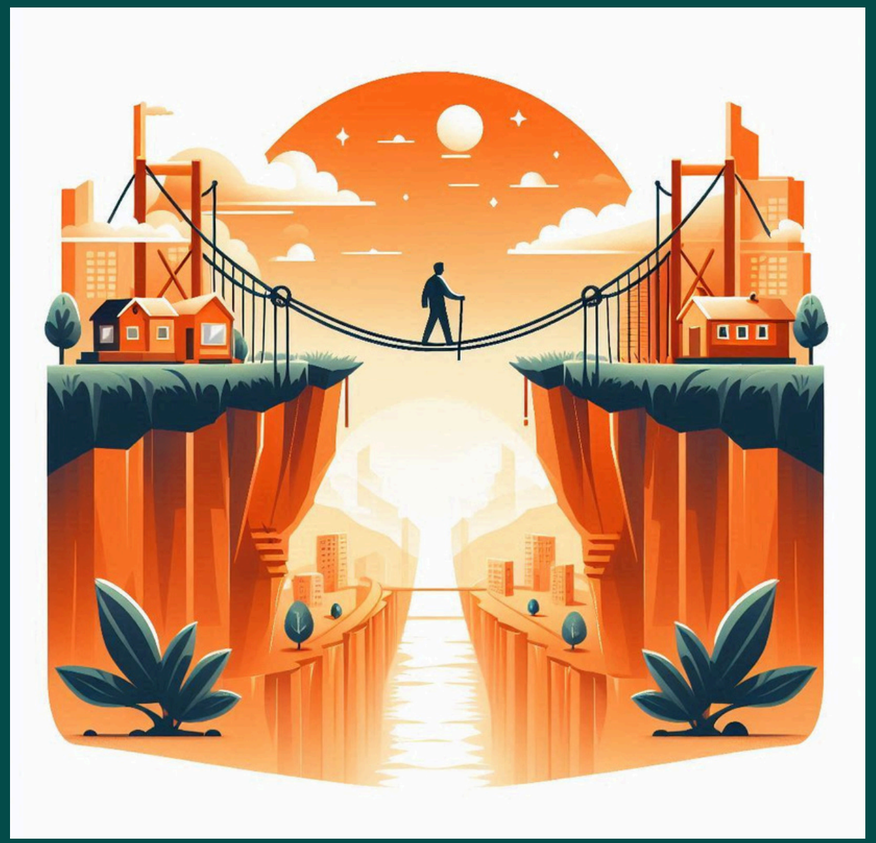
Small plugin:

- Change sitemap URL
- Get the pages wiring



Transform a plugin

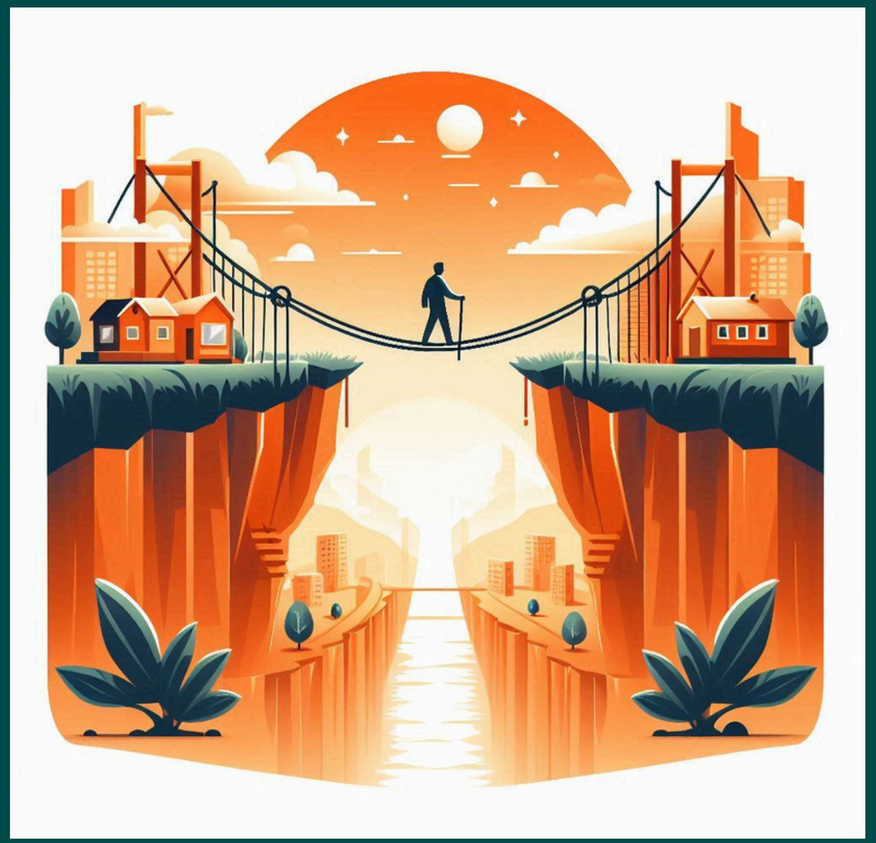
OK, so we code, ***right?***



Define what we want to achieve

Change sitemap URL

- Create a route to serve sitemap
- Change URL used by WordPress
- Save new sitemap in an option
- Admin interface for changing URL



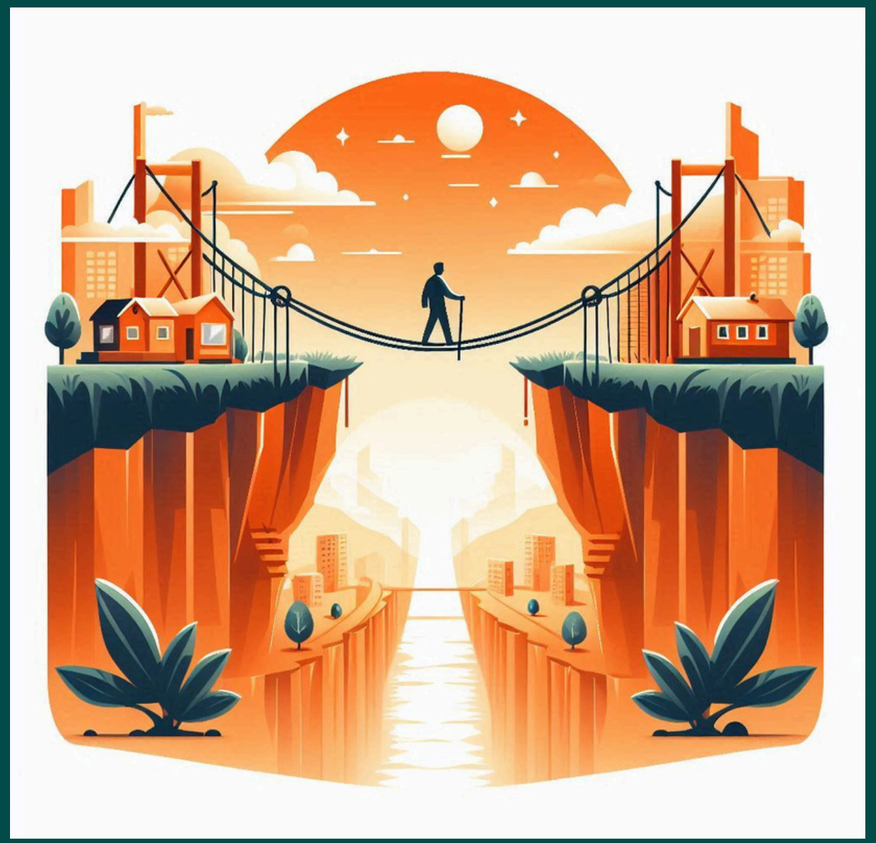
Define what we want to achieve

Change sitemap URL

- Create a route to serve sitemap
- Change URL used by WordPress
- Save new sitemap in an option
- Admin interface for changing URL

Get the pages wiring

- Fetch pages URLs from sitemap
- Enqueue pages to parse
- Parse pages



We need to keep it 50 mins

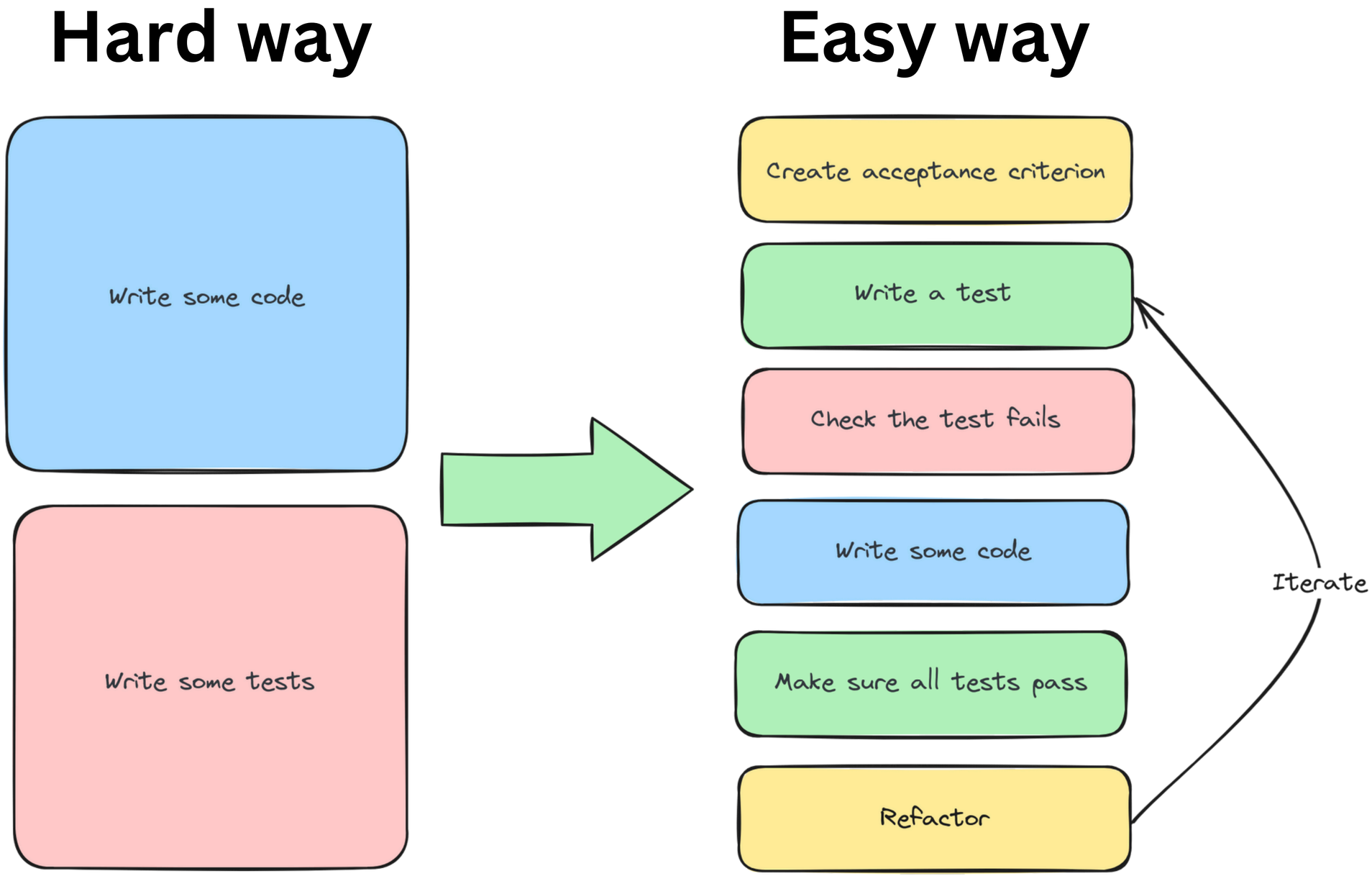
Change sitemap URL

- Create a route to serve sitemap
- Change URL used by WordPress
- ~~Save new sitemap in an option~~
- ~~Admin interface for changing URL~~

Get the pages wiring

- Fetch pages URLs from sitemap
- Enqueue pages to parse
- ~~Parse pages~~

OK, so we code, ***right?***



Developing Workflow

Better done than perfect

Gonna make some entorces:

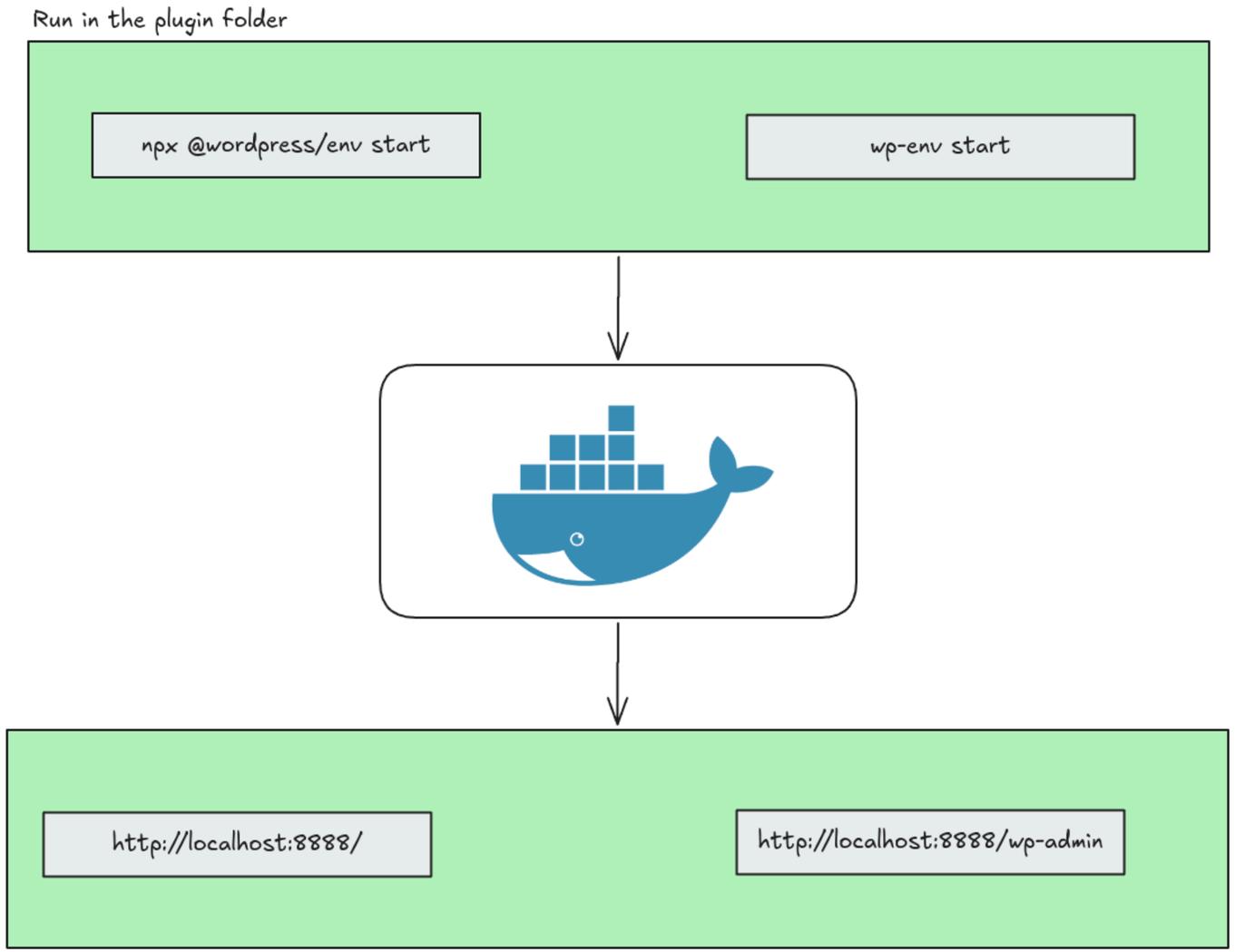
- Write all tests feature
- Refactor to Launchpad

Lets make a plan

Small plugin:

- Option to store the slug
- An action to register the route
- A filter to change sitemap URL
- An action to parse sitemap and create A.S. job

OK, so we code, ***right?***



Test env already configured

CREATE A ROUTE TO SERVE SITEMAP

```
Feature: Create a route to serve the sitemap

  Scenario:
    Given option 'sitemap_url' equals 'sitemap.xml'
    When WordPress initialize
    Then rule '^sitemap\.xml$' should be added to `index.php?sitemap=index`
```

2 ACs:

- When the option is set

CREATE A ROUTE TO SERVE SITEMAP

```
Feature: Create a route to serve the sitemap

  Scenario:
    Given option 'sitemap_url' equals 'sitemap.xml'
    When WordPress initialize
    Then rule '^sitemap\.xml$' should be added to `index.php?sitemap=index`

    Given option 'sitemap_url' equals ''
    When WordPress initialize
    Then rule '^sitemap\.xml$' should have same size
```

2 ACs:

- When the option is set
- When no option

```php
    public function set_up() {
        parent::set_up();
        add_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
    }

    no usages
    public function tear_down() {
        remove_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
        parent::tear_down();
    }

    no usages
    public function testShouldEnqueueNewRule() {
        global $wp_rewrite;

        $this->option_value = 'sitemap.xml';

        do_action( hook_name: 'init');

        $this->assertArrayHasKey("^sitemap\.xml$", $wp_rewrite->extra_rules_top);
        $this->assertSame("index.php?sitemap=index", $wp_rewrite->extra_rules_top["^sitemap\.xml$"]);
    }

    2 usages
    public function get_sitemap_option() {
        return $this->option_value;
    }
}
```

CREATE A ROUTE TO SERVE SITEMAP

2 tests:

- When the option is set

```php
    public function set_up() {
        parent::set_up();
        add_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
    }

    no usages
    public function tear_down() {
        remove_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
        parent::tear_down();
    }

    no usages
    public function testShouldEnqueueNewRule() {
        global $wp_rewrite;

        $this->option_value = 'sitemap.xml';

        do_action( hook_name: 'init');

        $this->assertArrayHasKey("^sitemap\.xml$", $wp_rewrite->extra_rules_top);
        $this->assertSame("index.php?sitemap=index", $wp_rewrite->extra_rules_top["^sitemap\.xml$"]);
    }

    no usages
    public function testEmptyOptionShouldNotEnqueue() {
        global $wp_rewrite;
        $this->option_value = '';

        $original_size = count($wp_rewrite->extra_rules_top);

        do_action( hook_name: 'init');

        $this->assertCount($original_size, $wp_rewrite->extra_rules_top);
    }

    2 usages
    public function get_sitemap_option() {
        return $this->option_value;
    }
}
```

CREATE A ROUTE TO SERVE SITEMAP

2 tests:

- When the option is set
- When no option

First step

```
no usages
function sitemap_add_redirect_rule() {
    $sitemap = get_option('sitemap_sitemap_url', '');

    add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
}


add_action( hook_name: 'init',  callback: 'sitemap_add_redirect_rule');
```

CREATE A ROUTE TO SERVE SITEMAP

2 steps:

- Pass the first test

First step

```php
no usages
function sitemap_add_redirect_rule() {
    $sitemap = get_option('sitemap_sitemap_url', '');

    add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
}


add_action( hook_name: 'init',  callback: 'sitemap_add_redirect_rule');
```

Second step

```php
no usages
function sitemap_add_redirect_rule() {
    $sitemap = get_option('sitemap_sitemap_url', '');

    if ( ! $sitemap ) {
        return;
    }

    add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
}


add_action( hook_name: 'init',  callback: 'sitemap_add_redirect_rule');
```

CREATE A ROUTE TO SERVE SITEMAP

2 steps:

- Pass the first test
- Pass both tests

CHANGE URL USED BY WORDPRESS

```
Feature: Change sitemap URL used by WordPress

  Scenario:
    Given option 'sitemap_url' equals 'sitemap.xml'
    When when retrieve the URL 'http://example.org'
    Then the URL should be 'http://example.org'
```

3 ACs:

- When URL is not sitemap

CHANGE URL USED BY WORDPRESS

```
Feature: Change sitemap URL used by WordPress

  Scenario:
    Given option 'sitemap_url' equals 'sitemap.xml'
    When when retrieve the URL 'http://example.org'
    Then the URL should be 'http://example.org'

    Given option 'sitemap_url' equals 'sitemap.xml'
    When when retrieve the URL 'http://example.org/wp-sitemap.xml'
    Then the URL should be 'http://example.org/sitemap.xml'
```

3 ACs:

- When URL is not sitemap
- When URL is sitemap

CHANGE URL USED BY WORDPRESS

```
Feature: Change sitemap URL used by WordPress

  Scenario:
    Given option 'sitemap_url' equals 'sitemap.xml'
    When when retrieve the URL 'http://example.org'
    Then the URL should be 'http://example.org'

    Given option 'sitemap_url' equals 'sitemap.xml'
    When when retrieve the URL 'http://example.org/wp-sitemap.xml'
    Then the URL should be 'http://example.org/sitemap.xml'

    Given option 'sitemap_url' equals ''
    When when retrieve the URL 'http://example.org/wp-sitemap.xml'
    Then the URL should be 'http://example.org/wp-sitemap.xml'
```

3 ACs:

- When URL is not sitemap
- When URL is sitemap
- When no option

```
usages
blic function set_up() {
  parent::set_up();
  add_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);


usages
blic function tear_down() {
  remove_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
  parent::tear_down();


usages
blic function testNotSitemapShouldReturnSame() {
  $this->option_value = 'sitemap.xml';
  $url = apply_filters( hook_name: 'home_url',  ...args: 'http://example.org/my-url', '/my-url');
  $this->assertSame('http://example.org/my-url', $url);
```

```
sages
blic function get_sitemap_option() {
  return $this->option_value;
```

CHANGE URL USED BY WORDPRESS

3 tests:

- When URL is not sitemap

```
usages
blic function set_up() {
  parent::set_up();
  add_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);


usages
blic function tear_down() {
  remove_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
  parent::tear_down();


usages
blic function testNotSitemapShouldReturnSame() {
  $this->option_value = 'sitemap.xml';
  $url = apply_filters( hook_name: 'home_url',  ...args: 'http://example.org/my-url', '/my-url');
  $this->assertSame('http://example.org/my-url', $url);


usages
blic function testSitemapShouldChange() {
  $this->option_value = 'sitemap.xml';
  $url = apply_filters( hook_name: 'home_url',  ...args: 'http://example.org/wp-sitemap.xml', '/wp-sitemap.xml');
  $this->assertSame('http://example.org/sitemap.xml', $url);




sages
blic function get_sitemap_option() {
  return $this->option_value;
```

CHANGE URL USED BY WORDPRESS

3 tests:

- When URL is not sitemap
- When URL is sitemap

```
usages
blic function set_up() {
  parent::set_up();
  add_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);


usages
blic function tear_down() {
  remove_filter( hook_name: 'pre_option_sitemap_sitemap_url', [$this, 'get_sitemap_option']);
  parent::tear_down();


usages
blic function testNotSitemapShouldReturnSame() {
  $this->option_value = 'sitemap.xml';
  $url = apply_filters( hook_name: 'home_url',  ...args: 'http://example.org/my-url', '/my-url');
  $this->assertSame('http://example.org/my-url', $url);


usages
blic function testSitemapShouldChange() {
  $this->option_value = 'sitemap.xml';
  $url = apply_filters( hook_name: 'home_url',  ...args: 'http://example.org/wp-sitemap.xml', '/wp-sitemap.xml');
  $this->assertSame('http://example.org/sitemap.xml', $url);


usages
blic function testSitemapAndEmptyOptionShouldReturnSame() {
  $this->option_value = '';
  $url = apply_filters( hook_name: 'home_url',  ...args: 'http://example.org/wp-sitemap.xml', '/wp-sitemap.xml');
  $this->assertSame('http://example.org/wp-sitemap.xml', $url);


sages
blic function get_sitemap_option() {
  return $this->option_value;
```

CHANGE URL USED BY WORDPRESS

3 tests:

- When URL is not sitemap
- When no option
- When URL is sitemap

First step

```php
no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args: 2);
```

CHANGE URL USED BY WORDPRESS

3 steps:

- When URL is sitemap

First step

```php
no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args: 2);
```

Second step

```php
no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    if('/wp-sitemap.xml' !== $path ) {
        return $url;
    }

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args: 2);
```

CHANGE URL USED BY WORDPRESS

3 steps:

- When URL is sitemap
- When URL is not sitemap

First step

```php
no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args: 2);
```

Second step

```php
no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    if('/wp-sitemap.xml' !== $path ) {
        return $url;
    }

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args: 2);
```

Third step

```php
no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    if('/wp-sitemap.xml' !== $path || ! $sitemap ) {
        return $url;
    }

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args: 2);
```

CHANGE URL USED BY WORDPRESS

3 steps:

- When URL is sitemap
- When URL is not sitemap
- When no option

FETCH PAGES URLS FROM SITEMAP

```
Feature: Fetch pages urls from sitemap
  Scenario:
    When sitemap fetch succeed
    Then action scheduler 'sitemap_scrape_links' for 'http://example.org'

  Scenario:
    When sitemap fetch fails
    Then no action scheduler 'sitemap_scrape_links' for 'http://example.org'
```

2 ACs:

- When sitemap fetched
- When sitemap failed

```
no usages
public function set_up() {
    parent::set_up();
    add_filter( hook_name: 'pre_http_request', [$this, 'mock_request']);
}

no usages
public function tear_down() {
    remove_filter( hook_name: 'pre_http_request', [$this, 'mock_request']);
    parent::tear_down();
}

no usages
public function testFetchSitemapShouldEnqueue() {
    $this->response = [...];
    do_action( hook_name: 'sitemap_parse_sitemap');
    $this->assertTrue(as_has_scheduled_action( hook: 'sitemap_scrape_links', [
        'http://example.org'
    ], group: 'sitemap'));
}

no usages
public function testFetchSitemapFailingShouldNotEnqueue() {
    $this->response = [...];
    do_action( hook_name: 'sitemap_parse_sitemap');
    $this->assertFalse(as_has_scheduled_action( hook: 'sitemap_scrape_links', [
        'http://example.org'
    ], group: 'sitemap'));
}

2 usages
public function mock_request() {
    return $this->response;
}
```

FETCH PAGES URLS FROM SITEMAP

2 tests:

- When sitemap fetched
- When sitemap failed

```php
no usages
function sitemap_parse_sitemap() {

    $sitemaps = wp_sitemaps_get_server();

    $sitemap_url = $sitemaps->index->get_index_url();

    $response = wp_safe_remote_get( $sitemap_url );

    if ( 200 !== wp_remote_retrieve_response_code( $response ) ) {
        return;
    }

    $data = wp_remote_retrieve_body( $response );

    $xml = simplexml_load_string( $data );

    $url_count = count( $xml->url );

    for ( $i = 0; $i < $url_count; $i++ ) {
        $url = (string) $xml->url[ $i ]->loc;
        if ( ! $url ) {
            continue;
        }
        as_enqueue_async_action( hook: 'sitemap_scrape_links', [$url], group: 'sitemap');
    }
}

add_action( hook_name: 'sitemap_parse_sitemap', callback: 'sitemap_parse_sitemap');
```

FETCH PAGES URLS FROM SITEMAP

2 steps:

- When sitemap fetched
- When sitemap failed

OK, but where is

***Launchpad*?**

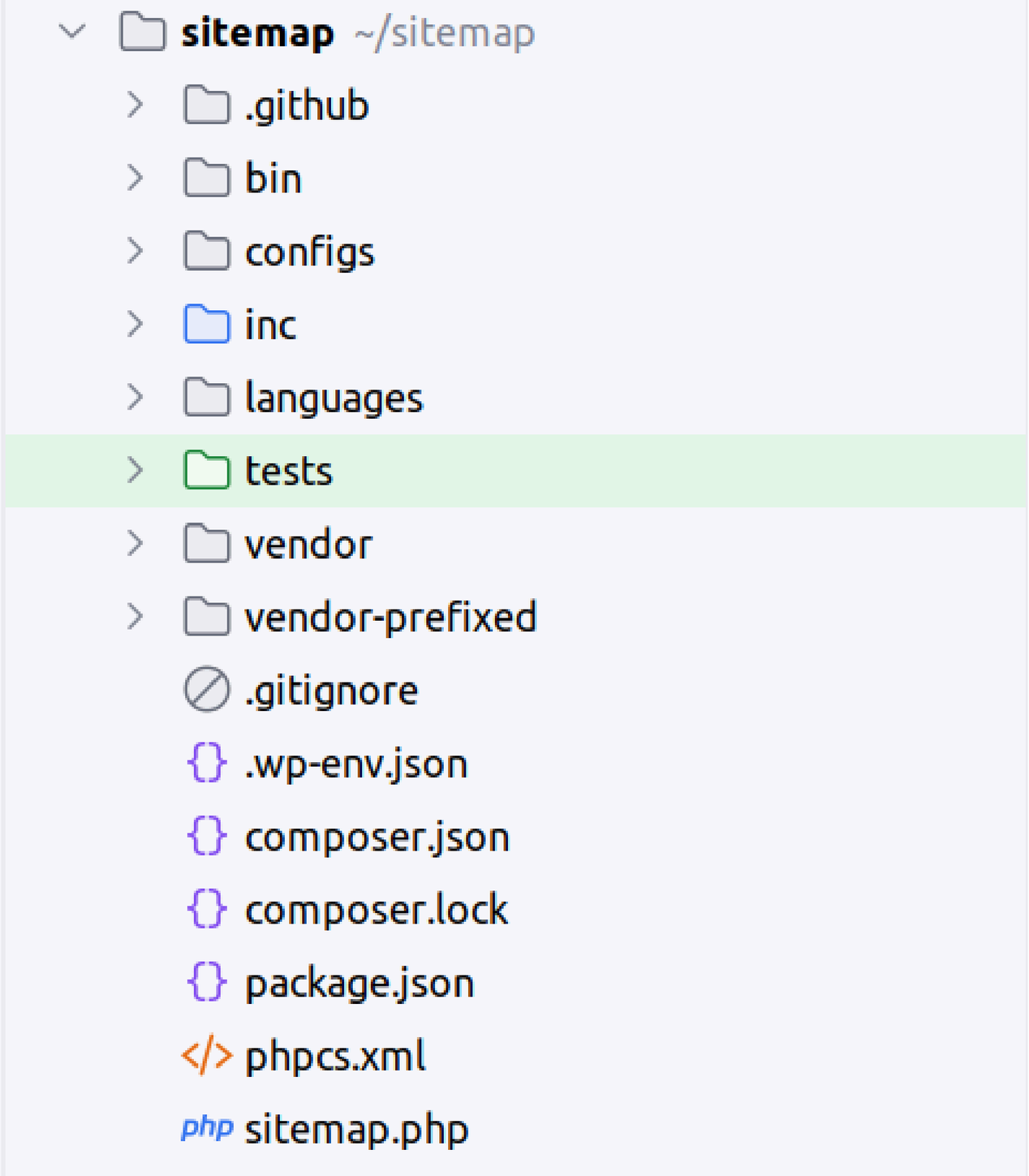
```
cyrille@cyrille-CREM-WXX9:~$ composer create-project wp-launchpad/launchpad my-plugin-test
Creating a "wp-launchpad/launchpad" project at "./my-plugin-test"
Installing wp-launchpad/launchpad (v1.0.5)
  - Installing wp-launchpad/launchpad (v1.0.5): Extracting archive
Created project in /home/cyrille/my-plugin-test
Loading composer repositories with package information
Updating dependencies
Lock file operations: 100 installs, 0 updates, 0 removals
  - Locking adhocore/cli (0.9.1)
  - Locking antecedent/patchwork (2.1.28)
  - Locking brain/monkey (2.6.1)
  - Locking brianhenryie/strauss (0.14.1)
  - Locking composer/ca-bundle (1.5.1)
  - Locking composer/class-map-generator (1.3.4)
  - Locking composer/composer (2.7.7)
  - Locking composer/installers (v2.3.0)
  - Locking composer/metadata-minifier (1.0.0)
  - Locking composer/pcre (3.1.4)
  - Locking composer/semver (3.4.3)
  - Locking composer/spdx-licenses (1.5.8)
  - Locking composer/xdebug-handler (3.0.5)
  - Locking dealerdirect/phpcodesniffer-composer-installer (v0.7.2)
  - Locking doctrine/instantiator (2.0.0)
  - Locking hamcrest/hamcrest-php (v2.0.1)
  - Locking json-mapper/json-mapper (2.22.2)
  - Locking justinrainbow/json-schema (5.3.0)
  - Locking league/container (3.4.1)
  - Locking league/event (3.0.3)
```

```
cyrille@cyrille-CREM-WXX9:~/sitemap$ bin/generator initialize
```

CREATE THE LAUNCHPAD PROJECT

2 steps:

- Create composer project
- Initialize project



LAUNCHPAD PROJECT STRUCTURE

3 important places:

- **inc:** Business logic
- **tests:** Tests
- **package.json:** local env

```php
<?php
/** Plugin Name: sitemap ...*/

use function Sitemap\Dependencies\LaunchpadCore\boot;

defined( constant_name: 'ABSPATH' ) || exit;


require __DIR__ . '/vendor-prefixed/wp-launchpad/core/inc/boot.php';

boot( plugin_launcher_file: __FILE__ );

no usages
function sitemap_add_redirect_rule() {
    $sitemap = get_option('sitemap_sitemap_url', '');

    add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
}


add_action( hook_name: 'init', callback: 'sitemap_add_redirect_rule');

no usages
function sitemap_change_url_sitemap($url, $path) {
    $sitemap = get_option('sitemap_sitemap_url', '');

    if('/wp-sitemap.xml' !== $path || ! $sitemap ) {
        return $url;
    }

    return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
}

add_filter( hook_name: 'home_url', callback: 'sitemap_change_url_sitemap', priority: 10, accepted_args

no usages
function sitemap_parse_sitemap() {
```

CREATE THE LAUNCHPAD PROJECT

Mixing old and new:

- Add logic inside **sitemap.php**
- After booting logic

OK, but where to ***start*?**

RUN THE LAUNCHPAD PHPSTAN

```
composer run-script run-stan
> vendor/bin/phpstan analyze --memory-limit=2G --no-progress -c tests/PHPStan/phpstan.neon.dist
 ------ ------------------------------------------------------------------------------------------------------------
  Line   sitemap.php
 ------ ------------------------------------------------------------------------------------------------------------
  21     Use Launchpad module to manipulate options.
         💡 composer require wp-launchpad/framework-options-take-off
  27     Adding a callback for a filter or an action should be done using the dispatcher or the @hook annotation.
  30     Use Launchpad module to manipulate options.
         💡 composer require wp-launchpad/framework-options-take-off
  39     Adding a callback for a filter or an action should be done using the dispatcher or the @hook annotation.
  68     Use Launchpad module to manipulate Action Scheduler jobs.
         💡 composer require wp-launchpad/action-scheduler-take-off
  72     Adding a callback for a filter or an action should be done using the dispatcher or the @hook annotation.
  76     Use Launchpad module to manipulate options.
         💡 composer require wp-launchpad/framework-options-take-off
  92     Adding a callback for a filter or an action should be done using the dispatcher or the @hook annotation.
  103    Use Launchpad module to manipulate options.
         💡 composer require wp-launchpad/framework-options-take-off
  108    Adding a callback for a filter or an action should be done using the dispatcher or the @hook annotation.
 ------ ------------------------------------------------------------------------------------------------------------


 [ERROR] Found 10 errors


Script vendor/bin/phpstan analyze --memory-limit=2G --no-progress -c tests/PHPStan/phpstan.neon.dist handling the run
```

3 types of alerts:

- Hook annotation
- Options
- Action Scheduler

```php
namespace Sitemap;

no usages
class Subscriber {
    /**
     * @hook init
     */
    no usages
    public function add_redirect_rule() {
        $sitemap = get_option('sitemap_sitemap_url', '');

        add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
    }

    /**
     * @hook home_url
     */
    no usages
    public function change_url_sitemap($url, $path) {
        $sitemap = get_option('sitemap_sitemap_url', '');

        if('/wp-sitemap.xml' !== $path || ! $sitemap ) {
            return $url;
        }

        return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
    }

    /**
     * @hook sitemap_parse_sitemap
     */
    no usages
    public function parse_sitemap() {

        if ( ! function_exists( function: 'as_enqueue_async_action' ) ) {
            return;
        }

        $sitemaps = wp_sitemaps_get_server();
```

CREATE A LAUNCHPAD SUBSCRIBER

2 steps:

- Create any class
- Use **hook** annotation

```php
<?php

namespace Sitemap;

use Sitemap\Dependencies\LaunchpadCore\Container\AbstractServiceProvider;

class ServiceProvider extends AbstractServiceProvider {

    /**
     * Define your services.
     *
     * @return void
     */
    1 usage
    protected function define() {
        $this->register_common_subscriber( classname: Subscriber::class);
    }
}
```

WIRING A LAUNCHPAD SUBSCRIBER

2 steps:

- Create composer project
- Initialize project

RUN THE LAUNCHPAD PHPSTAN

```
------ ------------------------------------------------------------
 Line   inc/Subscriber.php
------ ------------------------------------------------------------
 10     Use Launchpad module to manipulate options.
        💡 composer require wp-launchpad/framework-options-take-off
 19     Use Launchpad module to manipulate options.
        💡 composer require wp-launchpad/framework-options-take-off
 58     Use Launchpad module to manipulate Action Scheduler jobs.
        💡 composer require wp-launchpad/action-scheduler-take-off
 64     Use Launchpad module to manipulate options.
        💡 composer require wp-launchpad/framework-options-take-off
 97     Use Launchpad module to manipulate options.
        💡 composer require wp-launchpad/framework-options-take-off
------ ------------------------------------------------------------

[ERROR] Found 5 errors

Script vendor/bin/phpstan analyze --memory-limit=2G --no-progress -c tests/PHPStan/phpstan.neon.dist handling the run-st
```

2 types of alerts:

- Options
- Action Scheduler

USING THE LAUNCHPAD OPTION MODULE

```
cyrille@cyrille-CREM-WXX9:~/sitemap$ composer require wp-launchpad/framework-options-take-off
Composer could not detect the root package (wp-launchpad/launchpad) version, defaulting to '1.0.0'. See htt
./composer.json has been updated
Composer could not detect the root package (wp-launchpad/launchpad) version, defaulting to '1.0.0'. See htt
Running composer update wp-launchpad/framework-options-take-off
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking wp-launchpad/framework-options-take-off (v0.3.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
  - Downloading wp-launchpad/framework-options-take-off (v0.3.0)
  - Installing psr/container (2.0.2): Extracting archive
  - Installing wp-launchpad/dispatcher (v1.0.0): Extracting archive
  - Installing league/container (4.2.4): Extracting archive
  - Installing wp-launchpad/core (v0.3.2): Extracting archive
  - Installing wp-launchpad/framework-options-take-off (v0.3.0): Extracting archive
Generating autoload files
65 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> "bin/generator" auto-install
> composer update --no-interaction --no-scripts
Composer could not detect the root package (wp-launchpad/launchpad) version, defaulting to '1.0.0'. See htt
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 1 update, 0 removals
  - Upgrading squizlabs/php_codesniffer (3.11.0 => 3.11.1)
  - Locking wp-launchpad/framework-options (v0.3.0)
```

1 steps:

- Require the module
- Wait
- That’s all !

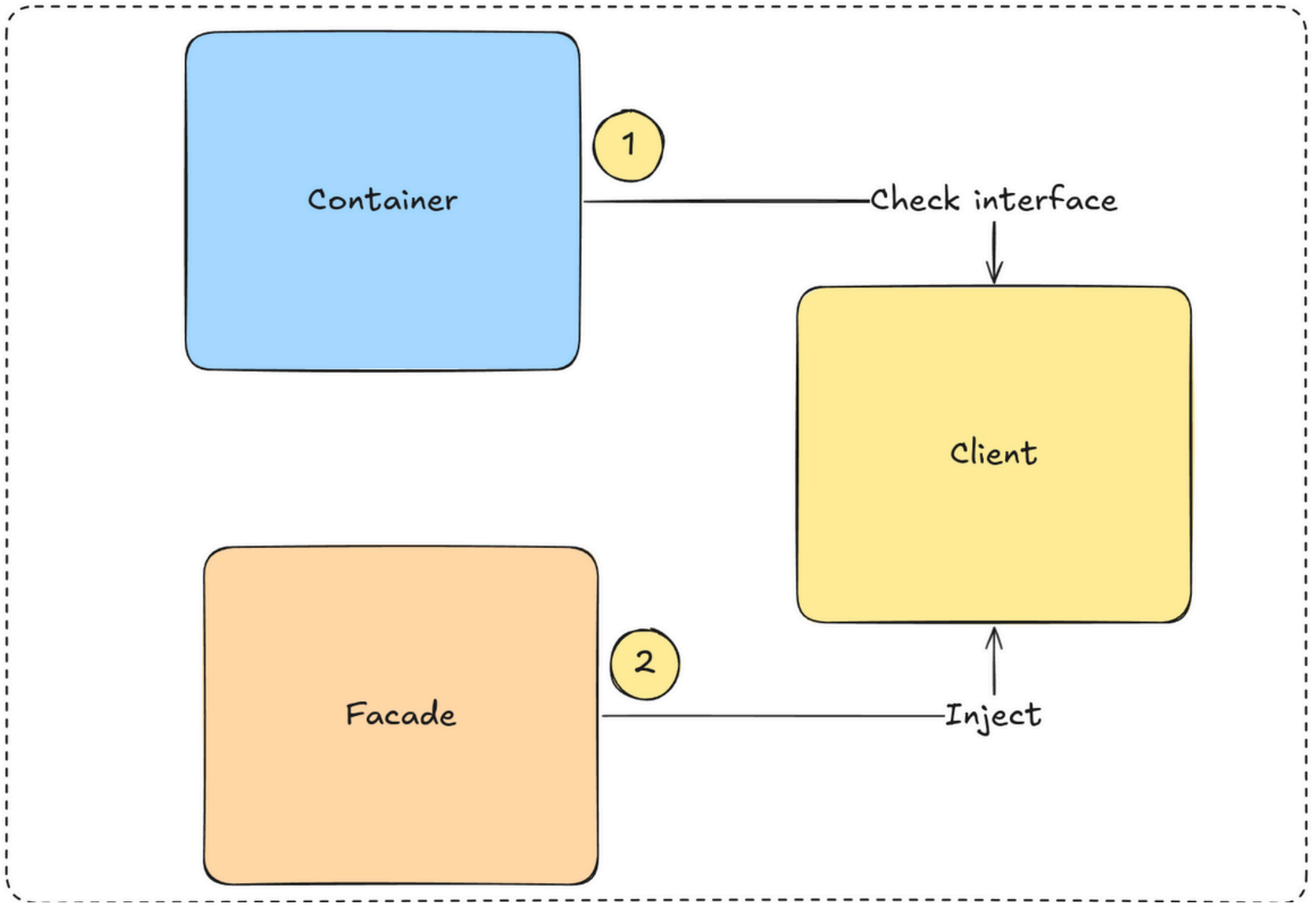
```php
<?php

namespace Sitemap;

use Sitemap\Dependencies\LaunchpadFrameworkOptions\Interfaces\SettingsAwareInterface;
use Sitemap\Dependencies\LaunchpadFrameworkOptions\Traits\SettingsAwareTrait;

1 usage
class Subscriber implements SettingsAwareInterface {
    use SettingsAwareTrait;

    /**
     * @hook init
     */
    no usages
    public function add_redirect_rule() {
        $sitemap = $this->settings->get('sitemap_sitemap_url', '');

        add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
    }

    /**
     * @hook home_url
     */
    no usages
    public function change_url_sitemap($url, $path) {
        $sitemap = $this->settings->get('sitemap_sitemap_url', '');

        if('/wp-sitemap.xml' !== $path || ! $sitemap ) {
            return $url;
        }

        return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
    }

    /**
     * @hook sitemap_parse_sitemap
     */
    no usages
    public function parse_sitemap() {
```

USING THE LAUNCHPAD OPTION MODULE

Use settings:

- Options vs Settings
- Use inflector not facade

UNDERSTAND INFLECTORS



2 steps:

- Detect interface
- Inject payload

```php
<?php

namespace Sitemap;

use Sitemap\Dependencies\LaunchpadFrameworkOptions\Interfaces\SettingsAwareInterface;
use Sitemap\Dependencies\LaunchpadFrameworkOptions\Traits\SettingsAwareTrait;

1 usage
class Subscriber implements SettingsAwareInterface {
    use SettingsAwareTrait;

    /**
     * @hook init
     */
    no usages
    public function add_redirect_rule() {
        $sitemap = $this->settings->get('sitemap_sitemap_url', '');

        add_rewrite_rule('^' . preg_quote($sitemap) . '$', 'index.php?sitemap=index', 'top');
    }

    /**
     * @hook home_url
     */
    no usages
    public function change_url_sitemap($url, $path) {
        $sitemap = $this->settings->get('sitemap_sitemap_url', '');

        if('/wp-sitemap.xml' !== $path || ! $sitemap ) {
            return $url;
        }

        return str_replace( search: '/wp-sitemap.xml', replace: '/' . $sitemap, $url);
    }

    /**
     * @hook sitemap_parse_sitemap
     */
    no usages
    public function parse_sitemap() {
```

UNDERSTAND INFLECTORS

Advantages:

- Easy to test
- Use a contract

Wait, so tests are ***wrong*****?**

```php
    2 usages
    protected $settings_value = '';

    no usages
    public function testShouldEnqueueNewRule($configs, $expected) {
        global $wp_rewrite;

        $this->settings_value = $configs['settings_value'];

        $original_size = count($wp_rewrite->extra_rules_top);

        do_action( hook_name: 'init');

        if(count($expected['rules']) === 0) {
            $this->assertCount($original_size, $wp_rewrite->extra_rules_top);
            return;
        }

        foreach ($expected['rules'] as $rule => $destination) {
            $this->assertArrayHasKey($rule, $wp_rewrite->extra_rules_top);
            $this->assertSame($destination, $wp_rewrite->extra_rules_top["^sitemap\.xml$"]);
        }
    }

    /**
     * @hook pre_get_sitemap_settings_sitemap_url
     */
    no usages
    public function get_sitemap_option() {
        return $this->settings_value;
    }
}
```

```php
<?php

return [
    'SettingsValueShouldAddRule' => [
        'configs' => [
            'settings_value' => 'sitemap.xml'
        ],
        'expected' => [
            'rules' => [
                "^sitemap\.xml$" => "index.php?sitemap=index"
            ]
        ],
    ],
    'SettingsValueNotSetShouldDoNothing' => [
        'configs' => [
            'settings_value' => ''
        ],
        'expected' => [
            'rules' => []
        ],
    ]
];
```

REFACTOR TESTS

4 steps:

- Change AC
- Change to settings
- Use fixtures
- Use hook annotation

RUN THE LAUNCHPAD PHPSTAN

```
composer run-script run-stan
> vendor/bin/phpstan analyze --memory-limit=2G --no-progress -c tests/PHPStan/phpstan.neon.dist
 ------ ------------------------------------------------------------
  Line   inc/Subscriber.php
 ------ ------------------------------------------------------------
  63     Use Launchpad module to manipulate Action Scheduler jobs.
         💡 composer require wp-launchpad/action-scheduler-take-off
 ------ ------------------------------------------------------------


 [ERROR] Found 1 error


Script vendor/bin/phpstan analyze --memory-limit=2G --no-progress -c tests/PHPStan/phpstan.neon.di
```

2 types of alerts:

- Action Scheduler

```
cyrille@cyrille-CREM-WXX9:~/sitemap$ composer require wp-launchpad/action-scheduler-take-off
Composer could not detect the root package (wp-launchpad/launchpad) version, defaulting to '1.0.0
./composer.json has been updated
Composer could not detect the root package (wp-launchpad/launchpad) version, defaulting to '1.0.0
Running composer update wp-launchpad/action-scheduler-take-off
Loading composer repositories with package information
Updating dependencies
Lock file operations: 3 installs, 0 updates, 0 removals
  - Locking crochetfeve0251/rocket-launcher-builder (v1.0.7)
  - Locking woocommerce/action-scheduler (3.9.0)
  - Locking wp-launchpad/action-scheduler-take-off (v0.0.4)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 9 installs, 0 updates, 0 removals
  - Downloading woocommerce/action-scheduler (3.9.0)
  - Downloading crochetfeve0251/rocket-launcher-builder (v1.0.7)
  - Downloading wp-launchpad/action-scheduler-take-off (v0.0.4)
  - Installing psr/container (2.0.2): Extracting archive
  - Installing league/container (4.2.4): Extracting archive
  - Installing woocommerce/action-scheduler (3.9.0): Extracting archive
  - Installing crochetfeve0251/rocket-launcher-builder (v1.0.7): Extracting archive
  - Installing wp-launchpad/action-scheduler-take-off (v0.0.4): Extracting archive
  - Installing wp-launchpad/dispatcher (v1.0.0): Extracting archive
  - Installing wp-launchpad/options (v0.1.5): Extracting archive
  - Installing wp-launchpad/core (v0.3.2): Extracting archive
  - Installing wp-launchpad/framework-options (v0.3.0): Extracting archive
Generating autoload files
65 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> "bin/generator" auto-install
> composer update --no-interaction --no-scripts
Composer could not detect the root package (wp-launchpad/launchpad) version, defaulting to '1.0.0
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 3 updates, 2 removals
  - Removing crochetfeve0251/rocket-launcher-builder (v1.0.7)
  - Removing league/pipeline (1.0.0)
  - Upgrading wp-launchpad/action-scheduler-take-off (v0.0.4 => v0.0.5)
  - Downgrading wp-launchpad/build (v0.0.10 => v0.0.8)
  - Downgrading wp-launchpad/cli (v1.1.0 => v1.0.7)
```

USING THE LAUNCHPAD ACTION SCHEDULER MODULE

1 steps:

- Require the module
- Wait
- That’s all !

USING THE LAUNCHPAD ACTION SCHEDULER MODULE

```php
namespace Sitemap;

use Sitemap\Dependencies\LaunchpadCore\Container\PrefixAware;
use Sitemap\Dependencies\LaunchpadCore\Container\PrefixAwareInterface;

3 usages
class Queue extends Dependencies\LaunchpadActionScheduler\Queue\AbstractASQueue implements PrefixAwareInterface {
    use PrefixAware;

    no usages
    public function __construct() {
        parent::__construct( action_scheduler_queue_group: 'sitemap', $this->prefix);
    }

    1 usage
    public function sitemap_scrape_links(string $url) {
        $this->queue->add_async('sitemap_scrape_links', [$url]);
    }
}
```

Create a queue:

- Extend from AbstractASQueue
- Add a method to enqueue

```php
namespace Sitemap;

use Sitemap\Dependencies\LaunchpadFrameworkOptions\Interfaces\SettingsAwareInterface;
use Sitemap\Dependencies\LaunchpadFrameworkOptions\Traits\SettingsAwareTrait;

1 usage
class Subscriber implements SettingsAwareInterface {
    use SettingsAwareTrait;

    /**
     * @var Queue
     */
    2 usages
    protected $queue;

    /**
     * @param Queue $queue
     */
    no usages
    public function __construct( Queue $queue ) {
        $this->queue = $queue;
    }

    /**
     * @hook init
     */
    no usages
    public function add_redirect_rule() {
        $sitemap = $this->settings->get('sitemap_sitemap_url', '');


        for ( $i = 0; $i < $url_count; $i++ ) {
            $url = (string) $xml->url[ $i ]->loc;
            if ( ! $url ) {
                continue;
            }
            $this->queue->sitemap_scrape_links($url);
        }
    }
```

USING THE LAUNCHPAD ACTION SCHEDULER MODULE

Use the queue in the subscriber:

- Add as dependency
- Use the queue

GOING FUTHER

Modules available

For the moment the all list of developed modules are the following:

Module	Description
Action scheduler	Handle asynchronous code or code to execute later
BerlinDB	A simple Database library
Logger	A PSR-3 compatible logger
Uninstaller	Set actions to be done on uninstall
Filesystem	Interact with the filesystem
Front-end	Create a front-end with Bud.js for your plug-in
Renderer	Use a renderer for your plug-in
Options	Use options for your plug-in
Bus	Use CRQS bus for your plug-in
Hook extractor	Extract hooks from your plugin into a yaml file

Much more modules and features:

- Command to release
- Constants, front-end, logger,...

Thanks !

COQUARD Cyrille
Software engineer
at WP Media

Launchpad repo