

PHPUnit Best Practices

Sebastian Bergmann





Sebastian Bergmann

Helps teams build testable software and implement efficient tests.

sharing experience

thePHP.cc

Interessengemeinschaft PHP e.V.

Als gemeinnütziger Verein möchten wir die Akzeptanz, Verbreitung und Weiterentwicklung der Programmiersprache PHP und das Wissen um die Programmiersprache PHP in Wissenschaft, Forschung und Ausbildung fördern.

 <https://igphp.de/>

 @igphp

Life is too short for old software.



At the risk of sounding like a broken record ...





Prepare for tomorrow, today

Sebastian Bergmann

October 18, 2018 | International PHP Conference | Munich

<https://thephp.cc/dates/2018/10/international-php-conference/prepare-for-tomorrow-today>



Use a supported version of PHP!

Version	Initial Release	Active Support	Security Support
PHP 7.3	December 6, 2018	December 6, 2020	December 6, 2021
PHP 7.2	November 30, 2017	November 30, 2019	November 30, 2020
PHP 7.1	December 1, 2016	December 1, 2018	December 1, 2019
PHP 7.0	December 3, 2015	December 3, 2017	January 10, 2019
PHP 5.6	August 28, 2014	January 19, 2017	January 10, 2019

<https://php.net/supported-versions.php>



<https://thephp.cc/news/2016/12/php-5-active-support-ends-now-what>

Use a supported version of PHPUnit!

Major Version	PHP Compatibility	Initial Release	Support
PHPUnit 8	PHP 7.2, PHP 7.3, PHP 7.4	February 1, 2019	Support ends on February 5, 2021
PHPUnit 7	PHP 7.1, PHP 7.2, PHP 7.3	February 2, 2018	Support ends on February 7, 2020
PHPUnit 6	PHP 7.0, PHP 7.1, PHP 7.2	February 3, 2017	Support ended on February 1, 2019
PHPUnit 5	PHP 5.6, PHP 7.0, PHP 7.1	October 2, 2015	Support ended on February 2, 2018
PHPUnit 4	PHP 5.3, PHP 5.4, PHP 5.5, PHP 5.6	March 7, 2014	Support ended on February 3, 2017

<https://phpunit.de/supported-versions.html>



Do not let new versions surprise you!

This is not a version constraint:

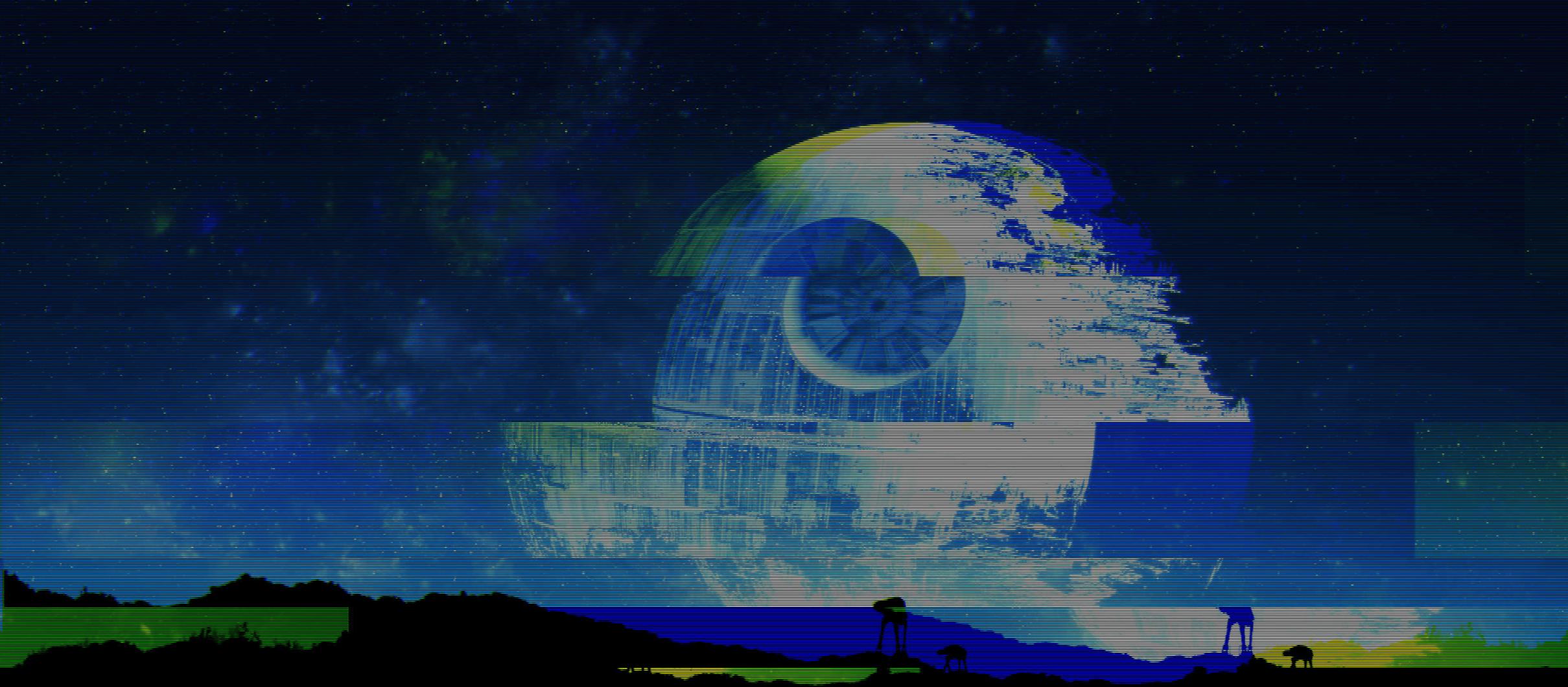
```
{  
    "require-dev": {  
        "phpunit/phpunit": "*"  
    }  
}
```

This is the "Death Star" version constraint:

```
{  
    "require-dev": {  
        "phpunit/phpunit": "*"  
    }  
}
```



<https://thephp.cc/news/2017/02/death-star-version-constraint>



Don't be too proud of the technological terror you've constructed.
The ability to destroy a planet is insignificant next to the power of testing.

This is not a version constraint:

```
{  
    "require-dev": {  
        "phpunit/phpunit": ">=8.0"  
    }  
}
```

This is not a version constraint:

```
{  
    "require-dev": {  
        "phpunit/phpunit": "dev-master"  
    }  
}
```

This does not lead to surprises:

```
{  
    "require-dev": {  
        "phpunit/phpunit": "^8.0"  
    }  
}
```

The same applies to PHPUnit's PHAR:

Please use

```
$ wget https://phar.phpunit.de/phpunit-8.1.phar
```

or

```
$ wget https://phar.phpunit.de/phpunit-8.phar
```

instead of

```
$ wget https://phar.phpunit.de/phpunit.phar
```

While we're on subject of the PHAR ...

I do not use Composer to install a tool but instead use a PHP Archive (PHAR). The self-contained PHAR of a tool ensures that its dependencies cannot conflict with the actual software's dependencies.

https://twitter.com/s_bergmann/status/999635212723212288

Do not install PHPUnit globally!

Read the ChangeLog
when you update to a new major version!

<https://phpunit.de/announcements/phpunit-8.html>





<https://thephp.cc/news/2019/02/help-my-tests-stopped-working>



PHP Configuration

php.ini

```
error_reporting=-1
```

We want to see all PHP errors, warnings, notices, etc. when we run our tests.

PHP Configuration

php.ini

```
log_errors_max_len=0
```

When something goes really wrong then we want to see the entire error message (which is truncated to 1024 characters by default).

PHP Configuration

php.ini

```
xdebug.show_exception_trace=0
```

We do not want Xdebug to print its exception traces while our tests are being executed.

PHP Configuration

`php.ini`

```
zend.assertions=1  
assert.exception=1
```

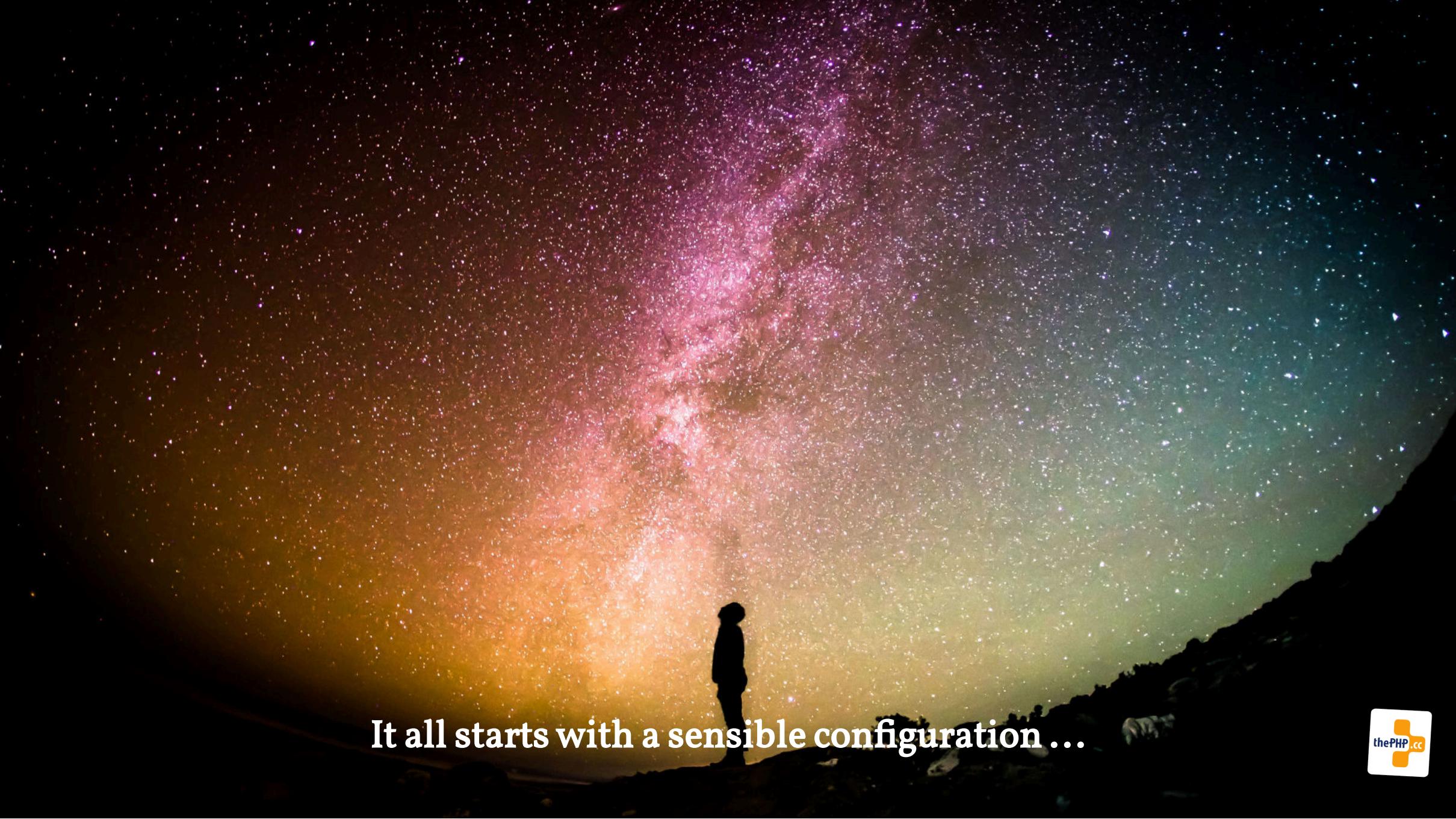
We want `assert()` statements to be evaluated and to raise exceptions.

PHP Configuration

php.ini

```
memory_limit=-1
```

The collection of code coverage data and the generation of a code coverage report sometimes requires more memory than PHP is allowed to use by default.



It all starts with a sensible configuration ...



phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd">
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
          verbose="true">
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    forceCoversAnnotation="true"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>

    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    forceCoversAnnotation="true"
    beStrictAboutCoversAnnotation="true"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    forceCoversAnnotation="true"
    beStrictAboutCoversAnnotation="true"
    beStrictAboutOutputDuringTests="true"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
</phpunit>
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    forceCoversAnnotation="true"
    beStrictAboutCoversAnnotation="true"
    beStrictAboutOutputDuringTests="true"
    beStrictAboutTodoAnnotatedTests="true"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>
    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
</phpunit>
```

```
$ ./tools/phpunit --generate-configuration
PHPUnit 8.1.6 by Sebastian Bergmann and contributors.

Generating phpunit.xml in /home/sb/project

Bootstrap script (relative to path shown above; default: vendor/autoload.php): src/autoload.php
Tests directory (relative to path shown above; default: tests):
Source directory (relative to path shown above; default: src):

Generated phpunit.xml in /home/sb/project
```

```
$ tree
.
├── phive.xml
└── src
    ├── autoload.php
    ├── BillingAddress.php
    ├── CreditCard.php
    ├── Email.php
    ├── ShippingAddress.php
    ├── ShoppingCartItem.php
    ├── ShoppingCart.php
    └── ShoppingCartRepository.php
├── tests
    ├── acceptance
    │   └── CheckoutProcessTest.php
    ├── integration
    │   └── ShoppingCartRepositoryTest.php
    └── unit
        └── ShoppingCartTest.php
└── tools
    └── phpunit

6 directories, 13 files
```

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    forceCoversAnnotation="true"
    beStrictAboutCoversAnnotation="true"
    beStrictAboutOutputDuringTests="true"
    beStrictAboutTodoAnnotatedTests="true"
    verbose="true">
    <testsuites>
        <testsuite name="unit">
            <directory suffix="Test.php">tests/unit</directory>
        </testsuite>
        <testsuite name="integration">
            <directory suffix="Test.php">tests/integration</directory>
        </testsuite>
        <testsuite name="acceptance">
            <directory suffix="Test.php">tests/acceptance</directory>
        </testsuite>
    </testsuites>
    <!-- ... -->
</phpunit>
```

```
$ ./tools/phpunit  
PHPUnit 8.1.6 by Sebastian Bergmann and contributors.
```

```
Runtime:      PHP 7.3.6 with Xdebug 2.7.2  
Configuration: /home/sb/project/phpunit.xml
```

```
..... 11 / 11 (100%)
```

```
Time: 83 ms, Memory: 12.00MB
```

```
OK (11 tests, 11 assertions)
```

```
$ ./tools/phpunit --testsuite unit
PHPUnit 8.1.6 by Sebastian Bergmann and contributors.
```

```
Runtime:      PHP 7.3.6 with Xdebug 2.7.2
Configuration: /home/sb/project/phpunit.xml
```

```
...
```

```
3 / 3 (100%)
```

```
Time: 69 ms, Memory: 10.00MB
```

```
OK (3 tests, 3 assertions)
```

```
$ ./tools/phpunit --testdox
PHPUnit 8.1.6 by Sebastian Bergmann and contributors.
```

```
Runtime:      PHP 7.3.6 with Xdebug 2.7.2
Configuration: /home/sb/project/phpunit.xml
```

ShoppingCart

- ✓ Is initially empty [0.15 ms]
- ✓ Total for empty shopping cart is 0 [0.16 ms]
- ✓ Item can be added [0.15 ms]

ShoppingCartRepository

- ✓ Can save shopping cart [0.17 ms]
- ✓ Can find shopping cart by id [0.15 ms]

CheckoutProcess

- ✓ Item can be added to shopping cart [0.32 ms]
- ✓ Shopping cart page can be displayed [0.16 ms]
- ✓ Shipping address can be entered [0.16 ms]
- ✓ Billing address can be entered [0.16 ms]
- ✓ Payment information can be entered [0.16 ms]
- ✓ Purchase can be completed [0.16 ms]

Time: 83 ms, Memory: 12.00MB

OK (11 tests, 11 assertions)

```
<?php declare(strict_types=1);
use PHPUnit\Framework\TestCase;

final class RouterTest extends TestCase
{
    /**
     * @dataProvider routesProvider
     */
    public function testRoutesRequest(string $url, string $handler): void
    {
        // ...
    }

    public function routesProvider(): array
    {
        return [
            '/foo/bar' => [
                '/foo/bar',
                FooBarHandler::class,
                // ...
            ]
        ];
    }
}
```

```
<?php declare(strict_types=1);
use PHPUnit\Framework\TestCase;

final class RouterTest extends TestCase
{
    /**
     * @dataProvider routesProvider
     */
    public function testRoutesRequest(string $url, string $handler): void
    {
        // ...
    }

    public function routesProvider(): array
    {
        return [
            '/foo/bar' => [
                // ...
            ]
        ];
    }
}
```

Router

- ✓ Routes request data set "/foo/bar"

```
<?php declare(strict_types=1);
use PHPUnit\Framework\TestCase;

final class RouterTest extends TestCase
{
    /**
     * @dataProvider routesProvider
     * @testdox      Routes $url to $handler
     */
    public function testRoutesRequest(string $url, string $handler): void
    {
        // ...
    }

    public function routesProvider(): array
    {
        return [
            '/foo/bar' => [
                // ...
            ]
        ];
    }
}
```

Router

✓ Routes /foo/bar to FooBarHandler



This is not readable:

```
public function testAvailabilityIsDisplayedCorrectly(): void
{
    // ...

    $response = file_get_contents(
        'http://localhost/search/2204/20171025-20171026'
    );

    $document = new \DOMDocument;
    $document->loadXML($response);

    $xpath = new \DOMXPath($document);
    $xpath->registerNamespace('html', 'http://www.w3.org/1999/xhtml');

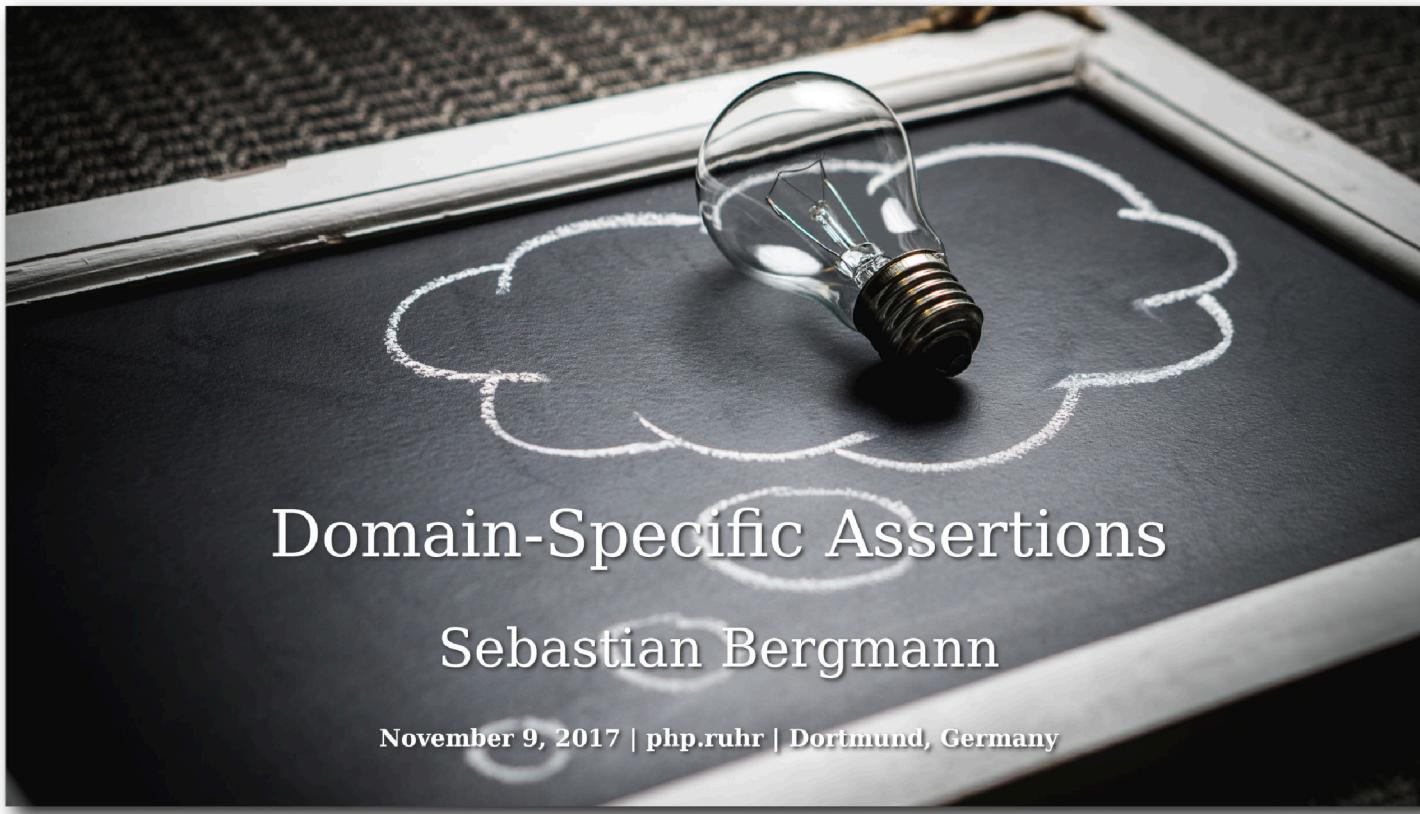
    $nodeList = $xpath->query(
        '//html:input[@id="price_radio_' . $roomId . '_' . $rateId . '"}'
    );

    $this->assertGreaterThan(0, $nodeList->length);
}
```

Test code should be readable!

```
public function testAvailabilityIsDisplayedCorrectly(): void
{
    // ...

    $this->requestAvailability($hotelId, $startDate, $endDate);
    $this->assertRateOfferedForRoom($rateId, $roomId);
}
```



Domain-Specific Assertions

Sebastian Bergmann

November 9, 2017 | php.ruhr | Dortmund, Germany

<https://thephp.cc/dates/2017/11/php-ruhr/domain-specific-assertions>

<https://www.youtube.com/watch?v=FC7Sz8gwbuc>

"The beauty of testing is found not in the effort but in the efficiency.
Knowing what should be tested is beautiful, and knowing what is
being tested is beautiful."

– Murali Nandigama

Code Coverage is a great tool.

Please use the `@covers` annotation.

Do not just take my word for it:

**Code Coverage marks as executed
not only subject under test, but *all*
related code.**

Use @covers annotation

https://slides.com/born_free/mutation_testing_infection



Good news!

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.1/phpunit.xsd"
    bootstrap="vendor/autoload.php"
    executionOrder="depends,defects"
    forceCoversAnnotation="true"
    beStrictAboutCoversAnnotation="true"
    beStrictAboutOutputDuringTests="true"
    beStrictAboutTodoAnnotatedTests="true"
    verbose="true">
    <testsuites>
        <testsuite name="default">
            <directory suffix="Test.php">tests</directory>
        </testsuite>
    </testsuites>

    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
</phpunit>
```



The configuration can "remind" you to use the @covers annotation.

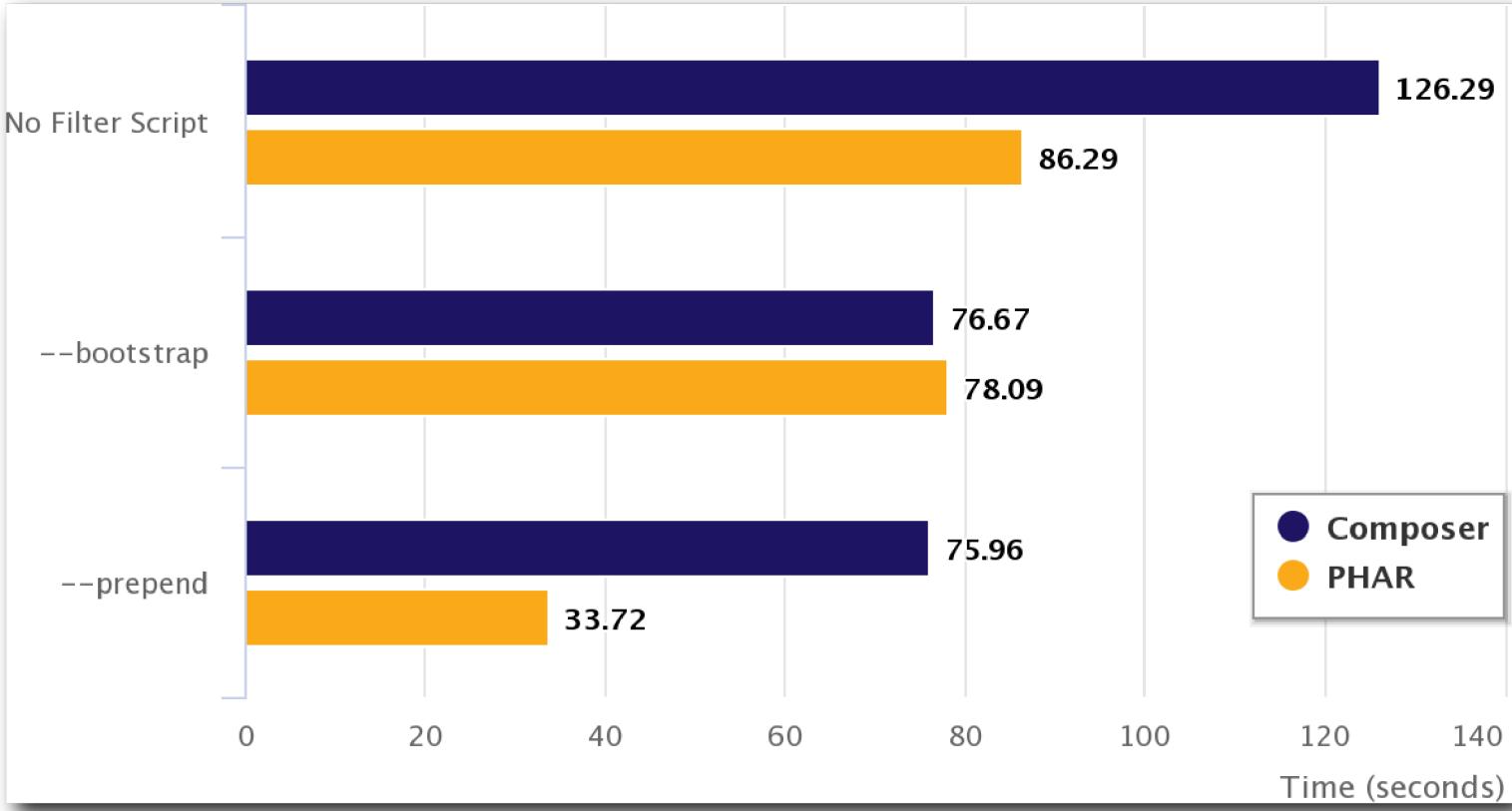


But it's slow ...

A wide-angle photograph of a carnival or fair at night. The most prominent feature is a large Ferris wheel on the left, its lights creating a massive, blurred circular pattern of red, orange, and yellow streaks against a dark blue sky. To the right, another attraction is visible, featuring a large purple and yellow dome-shaped structure. In the foreground, there's a sign with the word "SPACESHIP" in red neon lights, and a checkered racing track graphic. Various flags, including a red one with a white number "2", are flying from poles. The overall atmosphere is one of motion and vibrant nighttime entertainment.

Make it faster!





<https://thephp.cc/news/2019/01/faster-code-coverage>

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6			
PHP 7.3.6 + Xdebug 2.7.2			
PHP 7.3.6 + Xdebug 2.7.2	✓		
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓		
PHPDBG 7.3.6	✓		
PHP 7.3.6 + PCOV 1.0.3	✓		

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6		25.26 seconds	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2			
PHP 7.3.6 + Xdebug 2.7.2	✓		
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓		
PHPDBG 7.3.6	✓		
PHP 7.3.6 + PCOV 1.0.3	✓		

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6		25.26 seconds	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2		1.05 minutes	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2	✓		
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓		
PHPDBG 7.3.6	✓		
PHP 7.3.6 + PCOV 1.0.3	✓		

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6		25.26 seconds	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2		1.05 minutes	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2	✓	13.82 minutes	97.10 MB *
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓		
PHPDBG 7.3.6	✓		
PHP 7.3.6 + PCOV 1.0.3	✓		

* Xdebug does not use PHP's memory manager, memory usage shown by PHPUnit is not correct

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6		25.26 seconds	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2		1.05 minutes	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2	✓	13.82 minutes	97.10 MB *
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓	13.25 minutes	97.10 MB *
PHPDBG 7.3.6	✓		
PHP 7.3.6 + PCOV 1.0.3	✓		

* Xdebug does not use PHP's memory manager, memory usage shown by PHPUnit is not correct

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6		25.26 seconds	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2		1.05 minutes	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2	✓	13.82 minutes	97.10 MB *
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓	13.25 minutes	97.10 MB *
PHPDBG 7.3.6	✓	4.83 minutes	7.35 GB
PHP 7.3.6 + PCOV 1.0.3	✓		

* Xdebug does not use PHP's memory manager, memory usage shown by PHPUnit is not correct

Benchmark: <https://github.com/sebastianbergmann/diff>

	Code Coverage	Time	Memory
PHP 7.3.6		25.26 seconds	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2		1.05 minutes	81.10 MB
PHP 7.3.6 + Xdebug 2.7.2	✓	13.82 minutes	97.10 MB *
PHP 7.3.6 + Xdebug 2.7.2 with native filtering	✓	13.25 minutes	97.10 MB *
PHPDBG 7.3.6	✓	4.83 minutes	7.35 GB
PHP 7.3.6 + PCOV 1.0.3	✓	2.04 minutes	3.69 GB

* Xdebug does not use PHP's memory manager, memory usage shown by PHPUnit is not correct

<https://blog.krakjoe.ninja/2019/01/running-for-coverage.html>

WHAT ARE YOU WORKING ON?

TRYING TO FIX THE PROBLEMS I
CREATED WHEN I TRIED TO FIX
THE PROBLEMS I CREATED WHEN
I TRIED TO FIX THE PROBLEMS
I CREATED WHEN...



<https://xkcd.com/1739/>

Contact

 <https://thephp.cc>

 <https://talks.thephp.cc>

 sebastian@thephp.cc

 [@s_bergmann](https://twitter.com/s_bergmann)

sharing experience

thePHP.cc

Image Credits

- "Vinyl Turntable" by Giovanni Randisi
- "Skeletons" by Danielle MacInnes
- "Death Star" by Douglas Gavelin
- "Clap Board" by Jacob Owens
- "Spiral Jetty" by Greg Rakozy
- "Snail" by Patrick Pellegrini
- "Amusement Park Ride" by Jason Chen