



PHPUnit 10-12: What to know & what to forget

Sebastian Bergmann



Why was last year's PHPUnit 10 important?



What improvements did PHPUnit II bring?



Looking to the future: PHPUnit 12

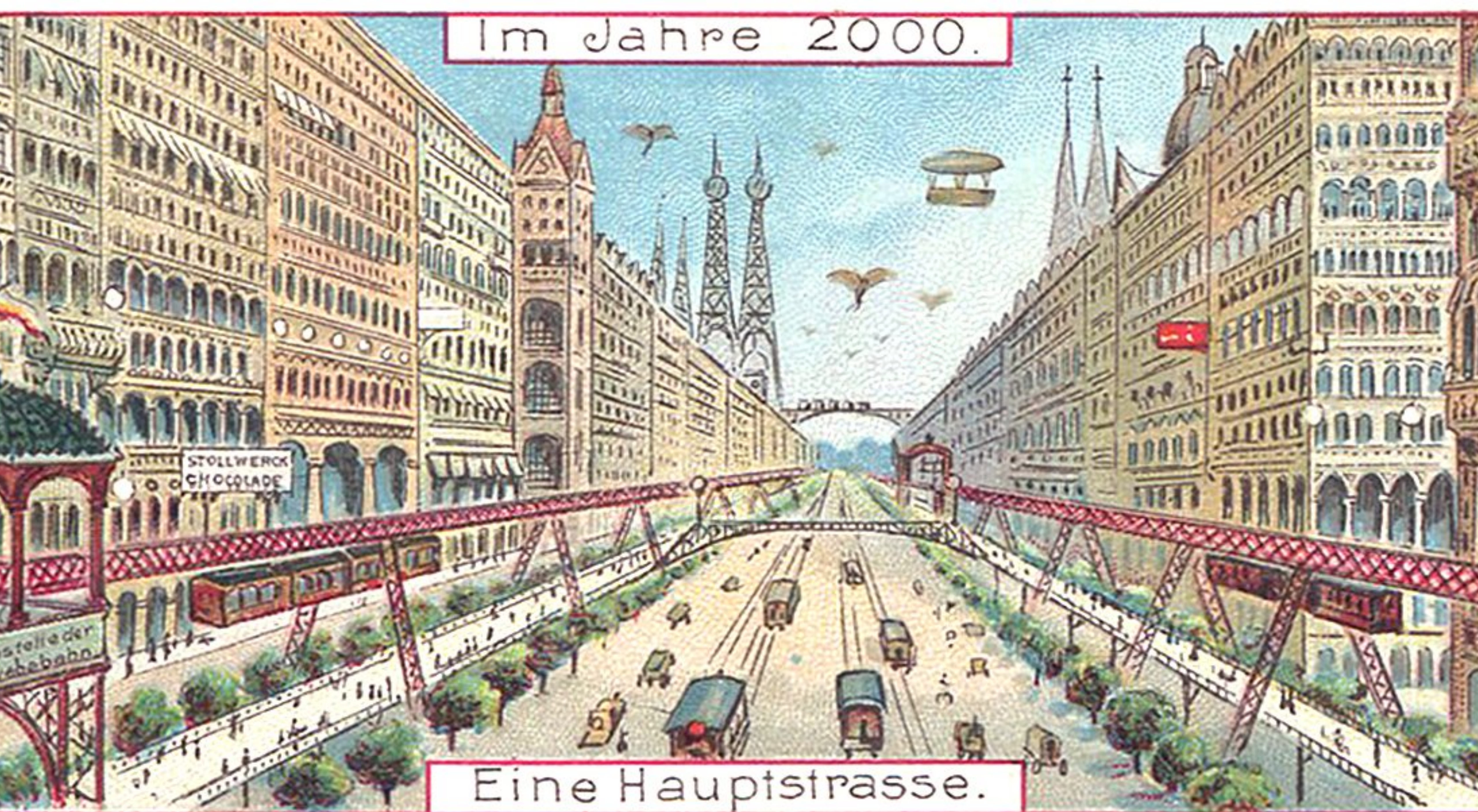
- **Sebastian** since 1978
- **Computers** since 1990
- **PHP** since 1998
- **Open Source** since 1998
- **PHPUnit** since 2000
- **thePHP.cc** since 2009



sharing experience

thePHP.cc

Im Jahre 2000.



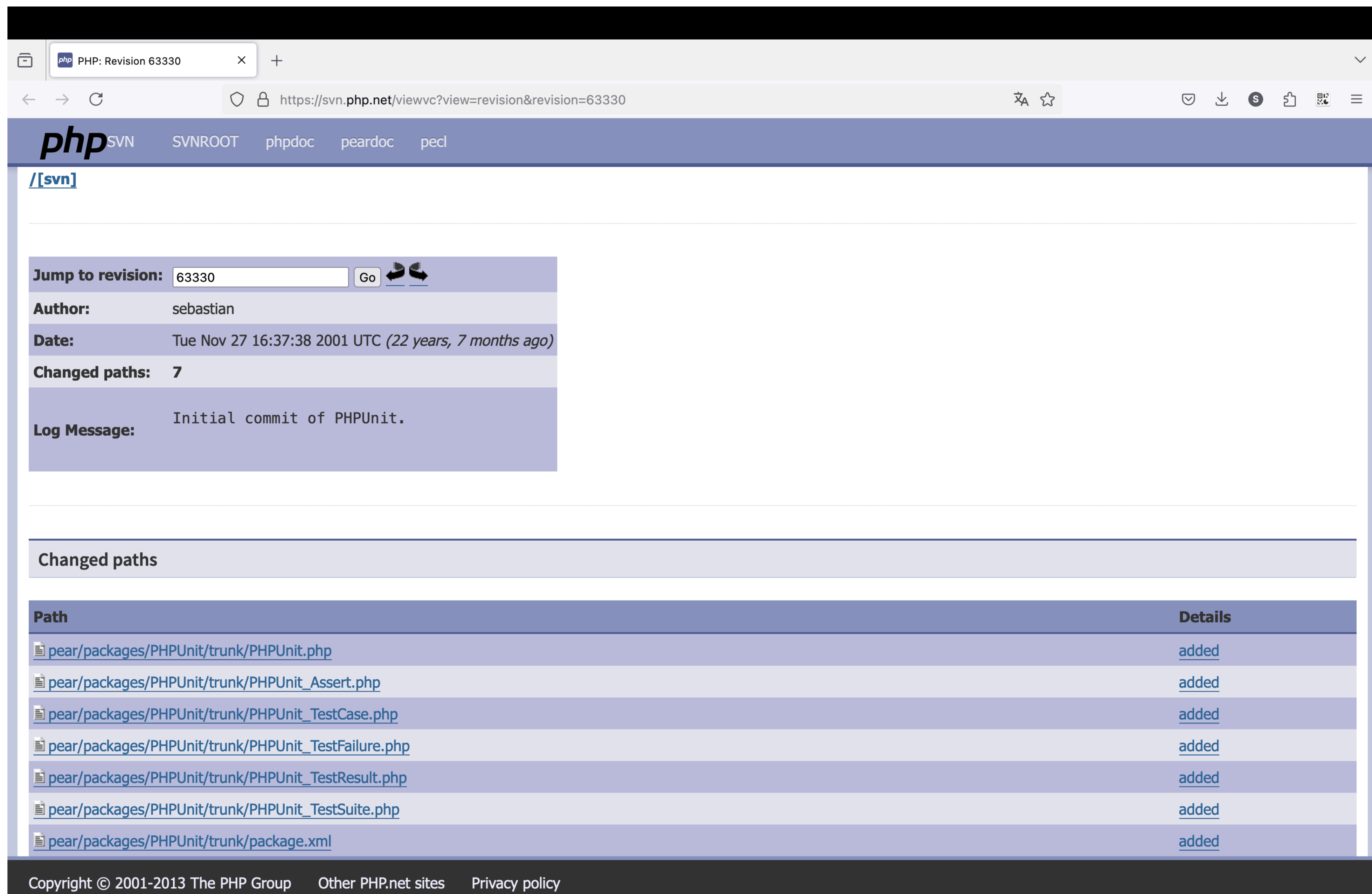
Eine Hauptstrasse.



JAGI


```
> git log --author="Sebastian Bergmann" --full-history --no-merges --date=relative --pretty="%cd %s"
.
.
7 years ago Use "Mac" instead of "OSX" to identify macOS in PHP_OS_FAMILY
.
.
12 years ago Invoke re2c with --no-generation-date to prevent unintentional / unnecessary changes in generated
            files
.
.
13 years ago Add optional argument to debug_backtrace() and debug_print_backtrace() to limit the amount of
            stack frames returned
.
.
15 years ago Add ReflectionMethod::setAccessible() for invoking non-public methods through the Reflection API
.
.
19 years ago Add an additional field $frame['object'] to the result array of debug_backtrace() that contains
            a reference to the respective object when the frame was called from an object
.
.
20 years ago Make ReflectionClass::getMethod() and ReflectionClass::getProperty() raise an ReflectionException
            instead of returning NULL on failure
24 years ago Fixed windows build
24 years ago Removed files from Win32 workspace that were removed in libmysql 3.23
24 years ago Removed obsolete entries from php.ini
.
.
```


**I started working on PHPUnit in 2000,
but did not publish anything about it until 2001.**



<https://svn.php.net/viewvc?view=revision&revision=63330>



I do not want to fish for compliments here.



I just want to provide context.

Major Version	PHP Compatibility	Release	End of Bugfix Support
PHPUnit 12	$\geq$ PHP 8.3	February 7, 2025	February 5, 2027
PHPUnit 11	$\geq$ PHP 8.2	February 2, 2024	February 6, 2026
PHPUnit 10	$\geq$ PHP 8.1	February 3, 2023	February 7, 2025
PHPUnit 9	$\geq$ PHP 7.3	February 7, 2020	February 2, 2024
PHPUnit 8	$\geq$ PHP 7.2	February 1, 2019	February 3, 2023

Bugfix Support
Life Support
In development



Feature Sunsetting

- **Soft Deprecation**

- `@deprecated` annotation (if applicable)
 - Maybe `# [Deprecated]` attribute in the future?
- Documentation

- **Hard Deprecation**

- Deprecations are emitted (if possible)

- **Removal**

- **Examples**

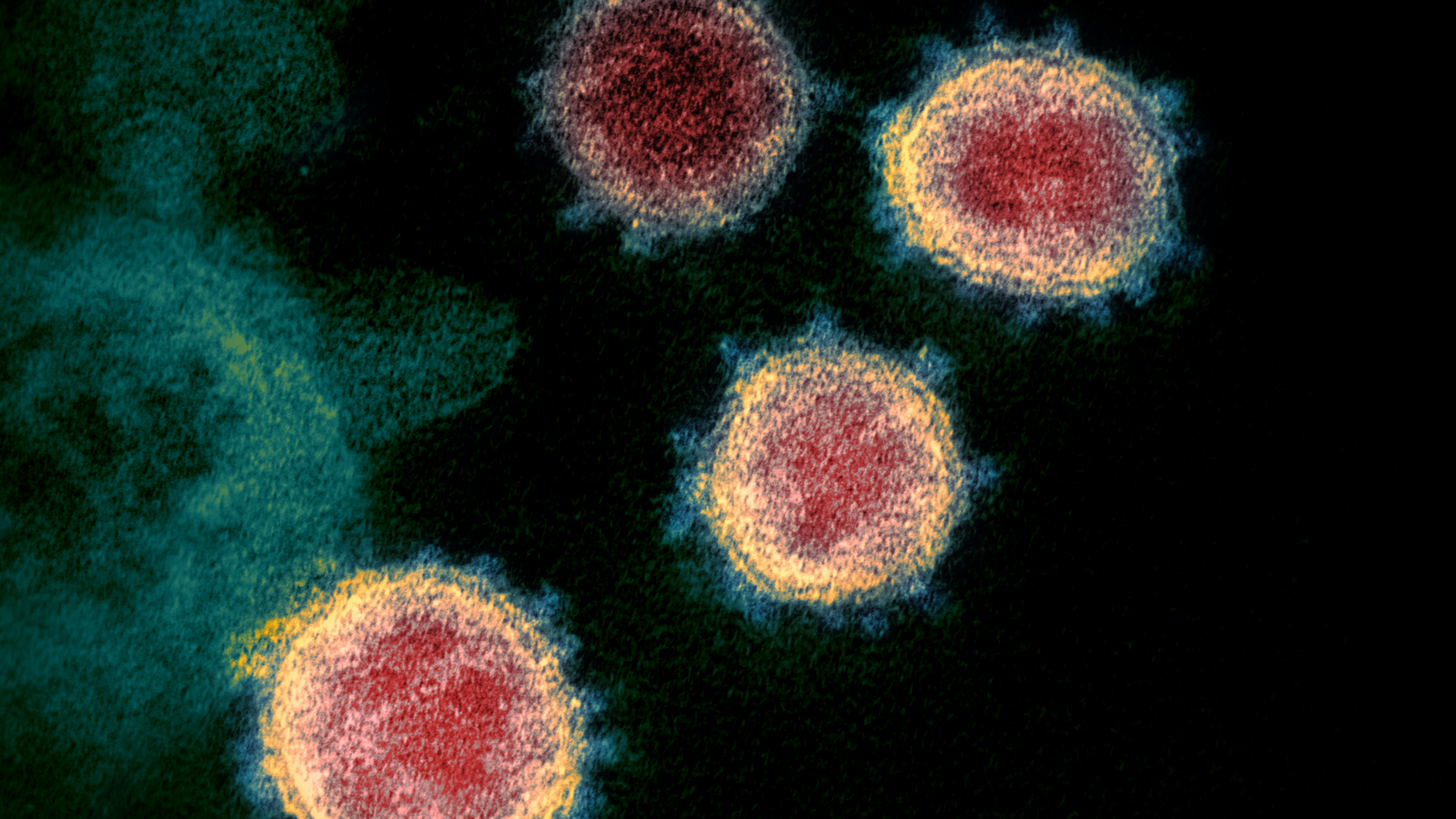
- <https://github.com/sebastianbergmann/phpunit/blob/10.5/DEPRECATIONS.md>
- <https://github.com/sebastianbergmann/phpunit/blob/11.4/DEPRECATIONS.md>

10

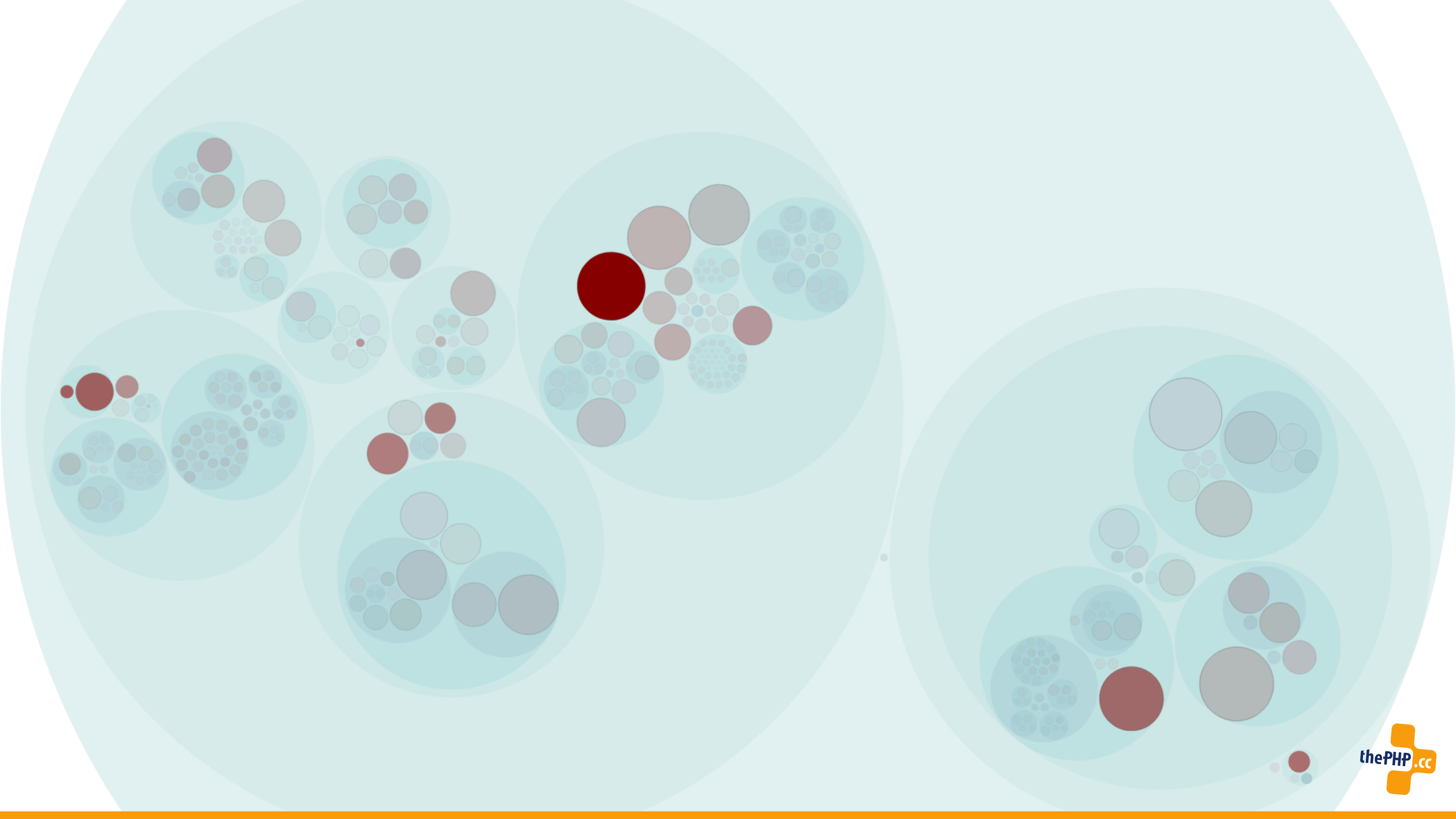
11

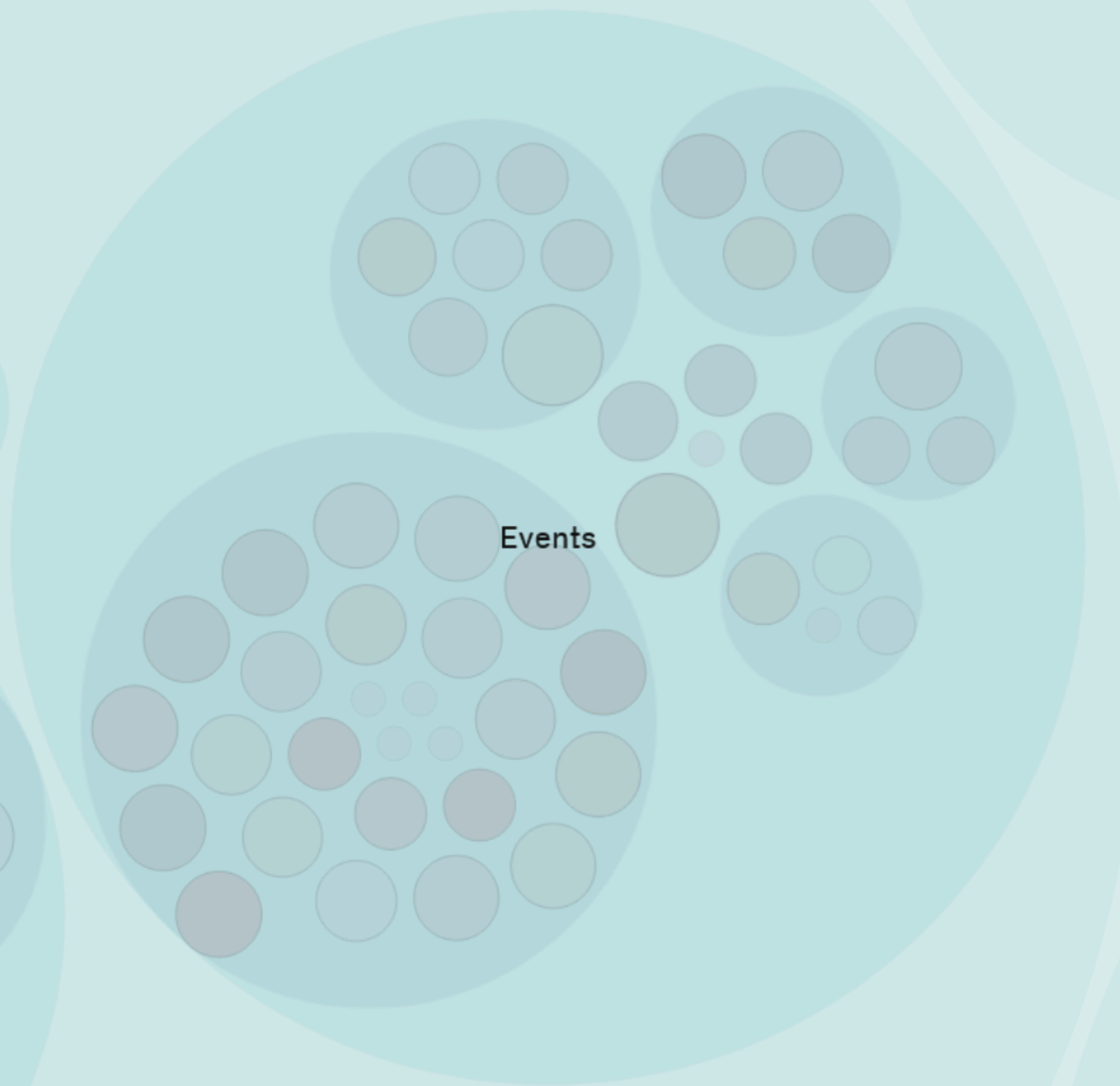
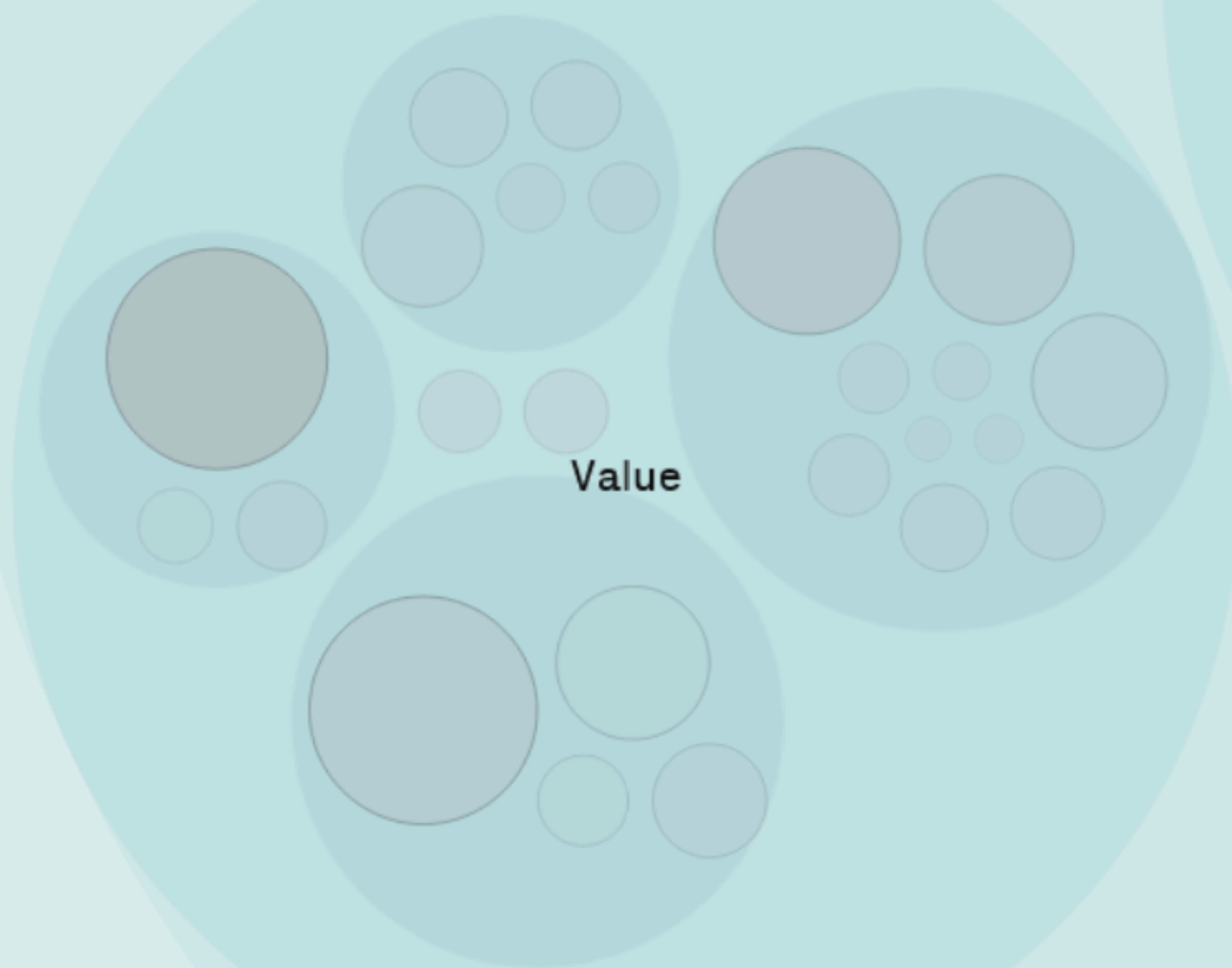
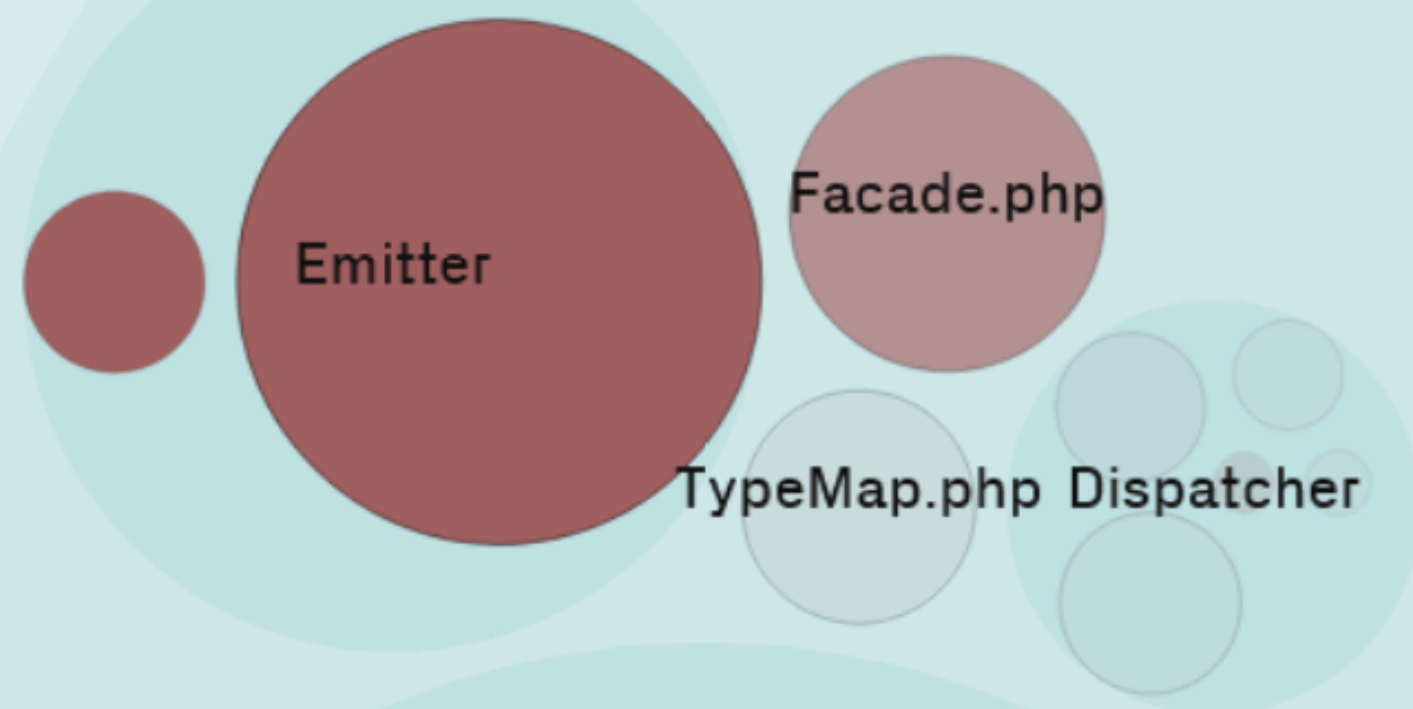
12

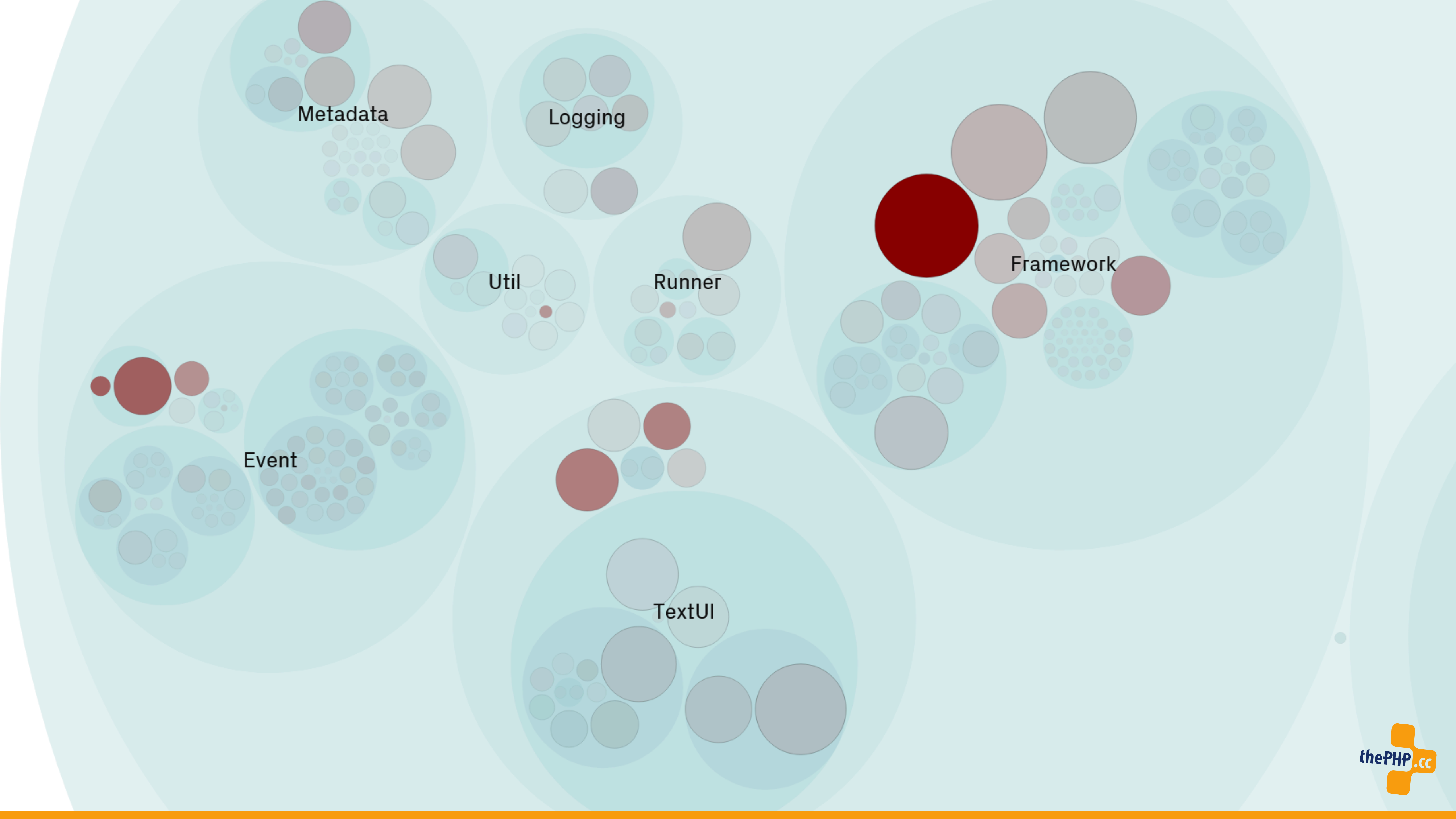


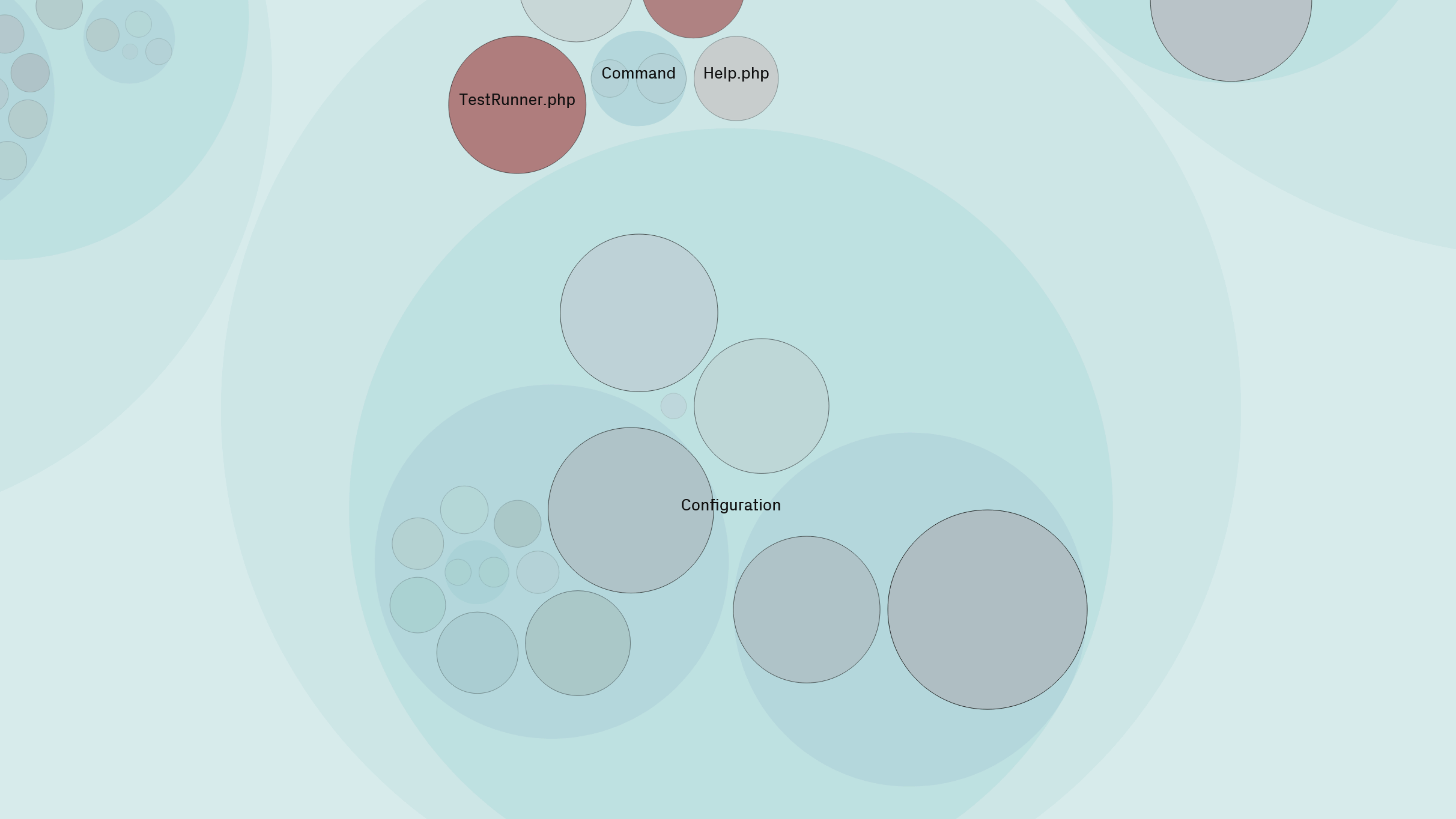


PHP 8









PHPUnit 10

- Event System
 - New system for extending PHPUnit
 - Rewrite of all code related to output and logging
- Outcome and Issues
 - Reporting of issues can be limited to your code
 - Baseline for deprecations, notices, and warnings
- Support for PHP 8 Attributes
- Annotations in code comments soft-deprecated
- Confusing and no longer needed test double features soft-deprecated

PHPUnit <= 9

Outcome

Passed

Errored

Failed

Warning

Considered Risky

Incomplete

Skipped

PHPUnit >= 10

Outcome	Issues
Passed	Triggered PHPUnit Error
Errored	Triggered PHPUnit Warning
Failed	Triggered PHPUnit Deprecation
Incomplete	Triggered E_USER_ERROR
Skipped	Triggered E_USER_WARNING
	Triggered E_USER_DEPRECATED
	Triggered E_USER_NOTICE
	Triggered E_WARNING
	Triggered E_DEPRECATED
	Triggered E_NOTICE or E_STRICT
	Considered Risky

Annotations in Code Comments

```
<?php declare(strict_types=1);  
namespace SebastianBergmann\Raytracer;  
  
use PHPUnit\Framework\TestCase;  
  
/**  
 * @covers \SebastianBergmann\Raytracer\Color  
 * @small  
 */  
final class ColorTest extends TestCase  
{  
    // ...  
}
```


Attributes

```
<?php declare(strict_types=1);
namespace SebastianBergmann\Raytracer;

use PHPUnit\Framework\Attributes\CoversClass;
use PHPUnit\Framework\Attributes\Small;
use PHPUnit\Framework\TestCase;

#[CoversClass(Color::class)]
#[Small]
final class ColorTest extends TestCase
{
    // ...
}
```


10

ONE DAY
7/3/2014
ONE DAY
8/13/2014

11

12

PHPUnit II

- Outcome and Issues
 - Distinguish between self, direct, and indirect deprecations
- Support for doubling read-only classes and property hooks
- Annotations in code comments hard-deprecated
- Confusing and no longer needed test double features hard-deprecated


```
<?php declare(strict_types=1);  
namespace example;  
  
use vendor\ThirdPartyClass;  
  
final class FirstPartyClass  
{  
    public function method(): true  
    {  
        (new ThirdPartyClass)->method();  
  
        @trigger_error(  
            'deprecation in first-party code',  
            E_USER_DEPRECATED  
        );  
  
        return true;  
    }  
}
```



```
<?php declare(strict_types=1);
namespace vendor;

final class ThirdPartyClass
{
    public function method(): void
    {
        @trigger_error(
            'deprecation in third-party code',
            E_USER_DEPRECATED
        );
    }

    public function anotherMethod(): true
    {
        return (new FirstPartyClass)->method();
    }
}
```



```
<?php declare(strict_types=1);
namespace example;

use PHPUnit\Framework\TestCase;
use vendor\ThirdPartyClass;

final class FirstPartyClassTest extends TestCase
{
    public function testOne(): void
    {
        $this->assertTrue((new FirstPartyClass)->method());
    }

    public function testTwo(): void
    {
        $this->assertTrue((new ThirdPartyClass)->anotherMethod());
    }
}
```



```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "https://schema.phpunit.de/11.4.xsd"
  bootstrap="vendor/autoload.php"
>
  <testsuites>
    <testsuite name="default">
      <directory>tests</directory>
    </testsuite>
  </testsuites>

  <source ignoreSuppressionOfDeprecations="true"
    ignoreDirectDeprecations="true"
    ignoreIndirectDeprecations="true">
    <include>
      <directory>src</directory>
    </include>
  </source>
</phpunit>
```


> ./phpunit
PHPUnit 11.4.3 by Sebastian Bergmann and contributors.

Runtime: PHP 8.4.1
Configuration: /Users/sb/example/phpunit.xml

D. 2 / 2 (100%)

Time: 00:00.014, Memory: 12.00 MB

1 test triggered 1 deprecation:

1) /Users/sb/example/src/FirstPartyClass.php:12
deprecation in first-party code

Triggered by:

* example\FirstPartyClassTest::testOne
/Users/sb/example/tests/FirstPartyClassTest.php:9

OK, but there were issues!

Tests: 2, Assertions: 2, Deprecations: 1.


```
<?php declare(strict_types=1);  
interface InterfaceWithHookedProperty  
{  
    public string $property { get; set; }  
}
```



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

final class ExampleTest extends TestCase
{
    public function testExample(): void
    {
        $stub = $this->createStub(
            InterfaceWithHookedProperty::class
        );

        $stub
            ->method(PropertyHook::get('property'))
            ->willReturn('value');

        $this->assertSame('value', $stub->property);
    }
}
```



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

final class ExampleTest extends TestCase
{
    public function testExample(): void
    {
        $mock = $this->createMock(
            InterfaceWithHookedProperty::class
        );

        $mock
            ->expects($this->once())
            ->method(PropertyHook::set('property'))
            ->with('value');

        $mock->property = 'value';
    }
}
```


10

ONE DAY
7/3/2014

ONE DAY
8/13/2014

11

12

PHPUnit 12

- Support for annotations in code comments of test code will be removed (hard-deprecated in PHPUnit 11)
- A lot of confusing and/or no longer needed test double functionality will be removed (hard-deprecated in PHPUnit 11)
- The above allows for a lot of cleanup under the hood and will ease the implementation of new functionality in the future
- Apart from that: it's too early to tell what PHPUnit 12 will bring

**You cannot buy PHPUnit,
but you can pay for PHPUnit.**

PHPUnit

<https://phpunit.de/sponsors.html>

**Good for you, good for me:
Consulting, Coaching, Training from**



<https://thephp.cc/>

Thank you!



`https://thePHP.cc/`



`sebastian@thephp.cc`



`@sebastian@phpc.social`

Image Credits

- <https://unsplash.com/de/fotos/orange-weisses-betongebaude--FZsyDjPbS4>
- <https://www.onverticality.com/blog/im-jahre-2000>
- https://de.academic.ru/pictures/dewiki/97/avz_iii.jpg
- <https://www.pexels.com/de-de/foto/fischen-angeln-schwarz-und-weiss-schwarzweiss-5887837>
- <https://www.pexels.com/de-de/foto/weisses-reihenboot-auf-gewasser-127160>