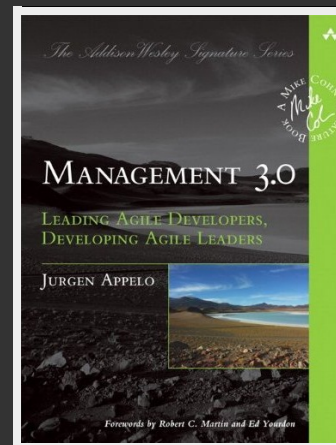
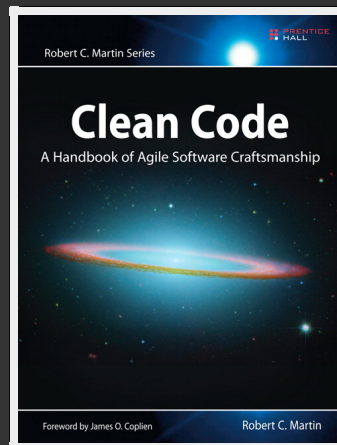


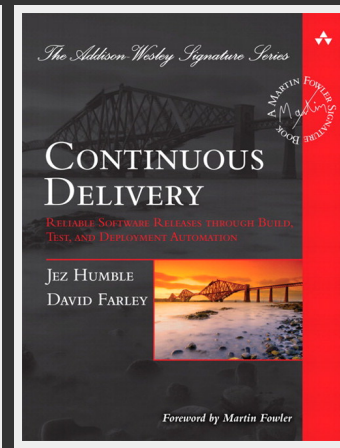
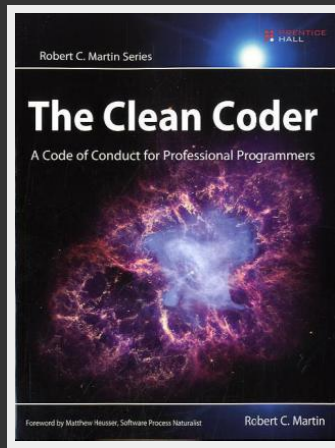
THE STATE OF PHPUNIT

Volker Dusch / @_edorian

ABOUT ME

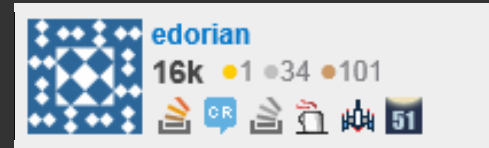
- PHP since 10 years
- CI
- CleanCode
- DevOps
- TDD
- Shipping





GET IN TOUCH

- stackoverflow:
- Twitter: @_edorian
- Xing / G+: Volker Dusch
- IRC: edorian
- Mail: php@wallbash.com



LET'S GO



TESTING?



TOO MAINSTREAM

quickmeme.com

Unit testing in PHP got mainstream

THAT'S A GOOD THING!

Unit testing got "normal" in PHP

Even people that don't really know PHP or even the CLI know that they should write unit tests.

AGENDA

- Current State
- PHPUnit.next
- The EcoSystem
- Annoying things we got rid of
- Features you might have missed
- Upcoming stuff
- Test suite organization
- Discussion

CURRENT STATE

- Version: 3.6.12
- Over **60** contributors since 3.6.0
- Deep into the 3.6 Bugfix cycle, no new features will be added. Those go into .next
- French docs thanks to @poum and still current Japanese docs thanks to @m-takagi!
- If you want to help out:
<https://github.com/sebastianbergmann/phpunit/#contributing>

PHPUNIT.NEXT

- Next up: 3.7
- 15+ features
- 10+ issues fixed
- Details later

BC

Did you like upgrading to PHPUnit 3.6?

BC FREE-ISH

We are trying really hard to not introduce any BC breaks.

- Test listeners trigger one autoloader call
- cwd gets restored after test case
- parameter cloning behavior
- Removed deprecated OutputTestCase
- and that's it

THE ECOSYSTEM

<http://phpqatools.org/> & friends

BDD

Use Behat, It's great!

We're not going to remove the BDD stuff in PHPUnit but It's not being worked on.

WEB INTERFACE?

Not sure why you'd want that but someone solved that for you

<https://github.com/NSinopoli/VisualPHPUnit>

MOCKING IS UGLY!

Real world problems and solutions:

[http://stackoverflow.com/search?tab=votes&q=\[phpunit\] mock](http://stackoverflow.com/search?tab=votes&q=[phpunit] mock)

MOCKERY

Try it! It's great.

Afraid it might be hard to set up and use?

Look into <https://github.com/etsy/phpunit-extensions/wiki/Mockery>

Works with strict mode, leaves the old API intact and is as easy as writing a comment:

```
class MyTest extends PHPUnit_Extensions_Mockery_TestCase {  
  
    /** @mockery Foo */  
    protected $foo;  
  
}
```

LOTS OF LEGACY CODE?

Or whatever "hard to test" is called these days.

<https://github.com/lapistano/proxy-object>

One of the main purpose of this library is to expose invisible (private or protected) methods and properties to the SUT.

TESTING WEB SERVICES

Unit and Integration test web services with the same code?

Try wsUnit <https://github.com/lapistano/wsunit>

IDE SUPPORT

- Netbeans support improved and they adopted the skeleton-generator
- Honestly I have no clue about eclipse
- PHPStorm has really beautiful coverage support by now
- vim user? :!phpunit --coverage-text

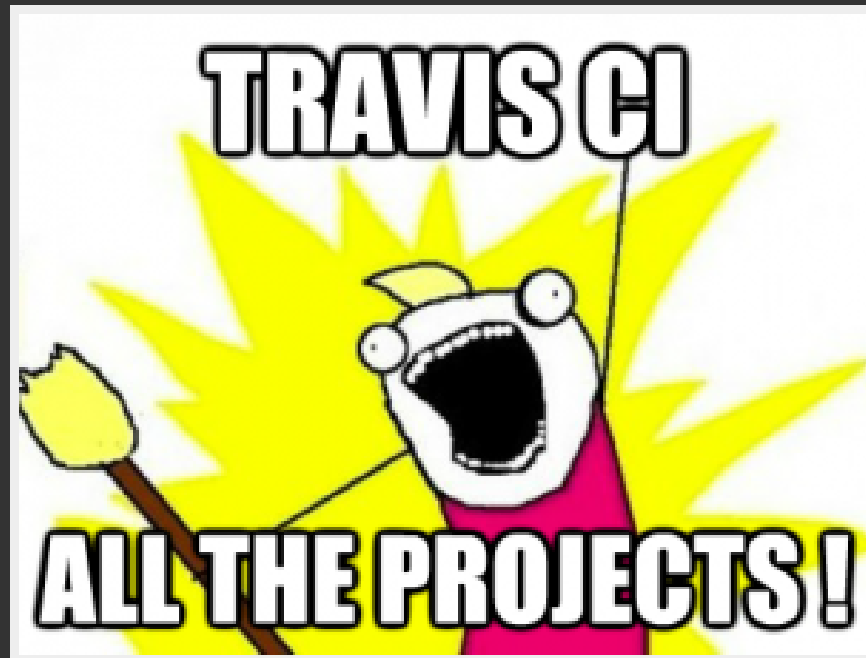
JENKINS

- Jenkins template: jenkins-php.org
- clover php plugin



TRAVIS CI

- Awesome for GitHub and collaboration
- *Everyone* has the way to run their tests documented by now
- No really CI but a very cool test runner
- Use `--coverage-text` if you want to show off :)



COMPOSER

Last week:

- When someone does the work it it will happen

This week:

- Someone did.

Current status at

<https://github.com/sebastianbergmann/phpunit/issues/522>



PHAR

The PHP tool format

We still have issues with process-isolation and figuring out the nicest way to maintain and build the phars.

Hopefully the 3.7 alpha will also feature a 3.6 & 3.7 phar.

PHPUNIT-SELENIUM

- Now maintained by @giorgiosironi
 - Code coverage support
-

If you ask me: Don't drive your web tests with phpunit.
Use Behat or something non php.

**ANNOYING THINGS WE GOT
RID OF**

@ BLOCK

When using process-isolation PHPUnit won't die silently when unserializing of the test result fails.

CLI ARGUMENT PARSING

```
phpunit TestA TestA.php TestB TestB.php
```

This only executed TestA. Now it gives you an error message that you are doing it wrong.

HEY, LISTEN!

Test listeners now trigger one autoload call instead of being silently ignored when the class was not loaded.

EXCEPTIONAL

Improved reporting of exceptions...

```
public function testNestedExceptions()
{
    $e3 = new Exception('Three');
    $e2 = new InvalidArgumentException('Two', 0, $e3);
    $e1 = new Exception('One', 0, $e2);
    throw $e1;
}
```

because it only told you...

```
ExceptionStackTest::testNestedExceptions
Exception: One

[trace]
```

but now...

EXCEPTIONAL

...we are printing out the previous exception names, messages and traces.

```
ExceptionStackTest::testNestedExceptions  
Exception: One
```

```
[trace]
```

```
Caused by  
InvalidArgumentException: Two
```

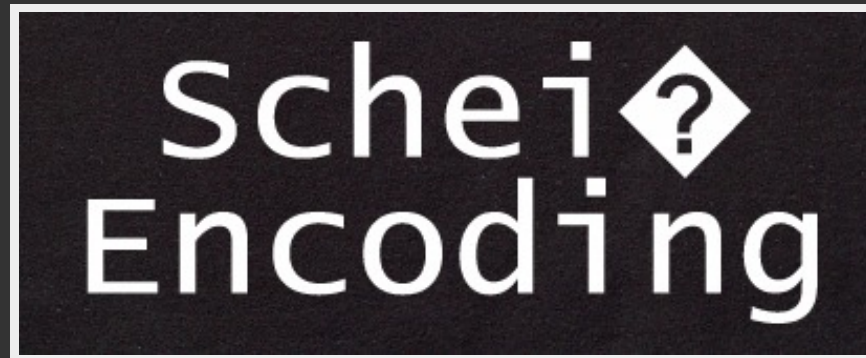
```
[trace]
```

```
Caused by  
Exception: Three
```

```
[trace]
```


????????????????

Fixed an annoying crash when using --process-isolation with PHP 5.3
and detect_unicode=on.



EVEN MORE EXCEPTIONAL

```
@expectedException \Exception
```

Works again!

ONE MORE THING

```
Fatal error: Uncaught exception 'PHPUnit_Framework_Exception' with message 'Neither "tests.php" nor "tests.php" could be opened.' in ...
```

That one is still alive. Sorry ;)

It just means that PHPUnit can't find any tests.

FEATURES YOU MIGHT HAVE MISSED

LINE ENDINGS MISMATCH

```
Issue503Test::testCompareDifferentLineEndings
Failed asserting that two strings are identical.
--- Expected
+++ Actual
@@ @@
  #Warning: Strings contain different line endings!
  foo
```

IMPROVED THE JSON LOG

If tests produce output it is now properly included in the log.

BOOTSTRAP ORDER

Bootstrap script should be loaded before trying to load
testSuiteLoaderClass.

PROCESS ISOLATION FOR EVERYONE

Process Isolation did not work when PHPUnit is invoked through Apache Ant, for instance, due to PHP binary detection issues.

@EXPECTEDEXCEPTIONCODE

Now can be 0 or a string.

Useful for PDO

BINARY MESS

Before it broke your terminal and BEEPed at you

```
1) FooTest::testFoo
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-
+ ☹️♥️♦️
+ ♂ ♀
+ 🎵☀️▶️⏪⏩!!⌨️-⏶⏷⏸⏹⏺⏻⏼⏽⏾⏿
```

Now it shows a hex dump:

```
Binary String: 0x000102030405.....
```

ONE LINE ANNOTATIONS

You can use:

```
/** @expectedException \FubarException */
```

COUNTING ON IT

Now working with Countable and Iterator Interfaces:

- `assertCount()`
- `assertCountAttributeCount()`
- `assertNotCount()`
- `assertAttributeNotCount()`
- `assertSameSize()`
- `assertNotSameSize()`

FLUENT INTERFACES

A little mock improvement:

```
->will($this->returnSelf());
```

@REQUIRES

```
/**  
 * @requires PHP 5.4  
 */  
public function testTrait()  
{  
}
```

REQUIRES

```
<file
  phpVersion="5.3.0"
  phpVersionOperator=">="
>
/path/to/MyTest.php
</file>
```

MAGIC SUPPORT

`assertEquals()` now looks for (and invokes) a `__toString()` method when an object and string are compared.

PERFORMANCE

`setUpBeforeClass()` and `tearDownAfterClass()` are no longer invoked when all tests of the class are skipped

UPCOMING STUFF

LESS MAGIC NUMBERS

@expectedException* now works with constants:

```
/**  
 * @expectedException Class  
 * @expectedExceptionCode @Class::TEAPOT_MISSING  
 * @expectedExceptionMessage @Class::TEAPOT_MISSING_MESSAGE  
 */
```

Reduce copy paste testing.

***TEST.PHP IS TOO MAINSTREAM**

Added --test-suffix that allows specifying which filename suffixes are recognised by PHPUnit.

Helps out some projects with their CI.

I don't see a general use case.

{'JSON': 'YES PLEASE'}

We added assertJson* functions that work like the existing assertXml* functions.

```
assertJsonFileEqualsJsonFile()  
assertJsonStringEqualsJsonFile()  
assertJsonStringEqualsJsonString()
```

CALLBACK MATCHER

Helps with complex assertions on mocks

```
public function send($message, EndpointConfig $config)
```

It's not pretty but it might solve that one annoying problem:

```
$foo->expects($this->once())  
->method('send')  
->with(  
    $this->equalTo('syn'),  
    $this->callback(  
        function ($config) {  
            $url = $config->getUrl();  
            if (!$url) { throw new Exception('....'); }  
            return $url->getHost() === 'http://example.com';  
        }  
    )  
);
```

OO-ARRAYS

```
assertArrayHasKey()  
assertArrayNotHasKey()
```

Now work with objects that implement `ArrayAccess`.

PHPUNIT.XSD

PHPUnit now provides a configuration.xsd schema file.

<http://schema.phpunit.de/configuration.xsd>

Useful to validate your phpunit.xml and phpunit.xml.dist configuration files.

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


MORE @REQUIRES

```
/**  
 * @requires PHP 5.4-dev  
 * @requires PHPUnit 3.7-dev  
 * @requires function someFunc  
 * @requires function Class::foo  
 * @requires extension apc  
 * @requires extension mysqli  
 */
```

- Require functions, methods and extensions
- Merges Class & Method doc blocks
- PHP & PHPUnit version get override

.PHPT IMPROVEMENTS

```
--TEST--
```

```
GH-503: assertEquals() Line Ending Differences Are Obscure
```

```
--FILE--
```

```
<?php
```

```
$_SERVER['argv'][1] = '--no-configuration';
```

```
$_SERVER['argv'][2] = 'Issue503Test';
```

```
$_SERVER['argv'][3] = __DIR__ . '/503/Issue503Test.php';
```

```
require_once __DIR__ . '/../../../../../PHPUnit/Autoload.php';
```

```
PHPUnit_TextUI_Command::main();
```

```
?>
```

.PHPT IMPROVEMENTS

```
--EXPECTF--
PHPUnit %s by Sebastian Bergmann.

F

Time: %i %s, Memory: %sMb

There was 1 failure:

1) Issue503Test::testCompareDifferentLineEndings
Failed asserting that two strings are identical.
--- Expected
+++ Actual
@@ @@
 #Warning: Strings contain different line endings!
foo

%s:%i

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

.PHPT IMPROVEMENTS

Before 3.7 diffs where really hard to read.

Now the diff will be processed line by line.

LISTS AND COLLECTIONS

```
public function testAssertContainsOnlyInstancesOf()  
{  
    $library = array(  
        new Book(),  
        new Book()  
    );  
    $this->assertContainsOnlyInstancesOf(  
        'Book',  
        $library  
    );  
}
```

Helper for checking Collection objects and arrays.

Descriptive assertion and error message.

FAILED @EXPECTEDEXCEPTION ANNOTATIONS

Before:

```
Failed asserting that exception of type "ErrorException" matches expected exception "MyException".
```

Now:

```
Failed asserting that exception of type "ErrorException" matches expected exception "MyException".  
Message from "ErrorException" was "undefined variable $htis".
```

LESS REQUIRES, LESS CRASHES

```
<whitelist addUncoveredFilesFromWhitelist="true">
```

People seem to constantly run into issues with code coverage

```
<whitelist processUncoveredFilesFromWhitelist="true">
```

When enabled, uncovered whitelisted files are processed to properly calculate the number of executable lines.

If not we take a quite ok guess to get the number of ELOC.

MORE USEFUL GROUPS

```
$ phpunit --group FlakyTests
```

```
<phpunit>  
  <groups>  
    <exclude>  
      <group>FlakyTests</group>  
    </exclude>  
  </groups>  
  <testsuites ... />  
</phpunit>
```

Now works as expected

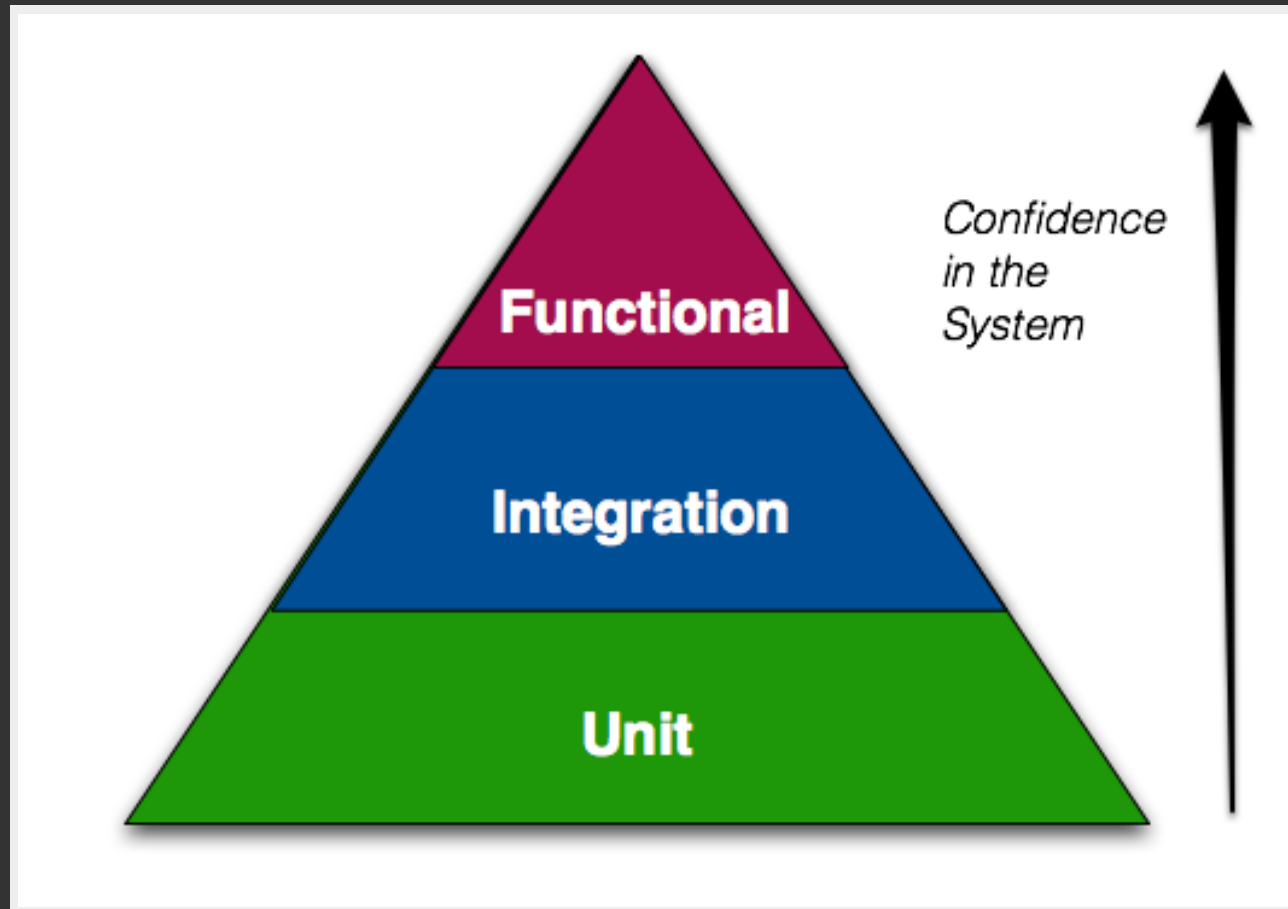
TEST SUITE ORGANIZATION

FOLDER LAYOUT?

```
project/  
  phpunit.xml.dist  
  src/  
  tests/  
    bootstrap.php  
    unit/  
    integration/  
    functional/ (or system/)  
    helpers/
```

TEST DISTRIBUTION

<http://elblinkin.info/2012/03/goldilocks-on-test-sizes/>



SMALL HINTS

- Use the xml configuration for pretty much everything
- Separate unit and integration tests
- Your unit test folders should mirror your applications folder structure
- To only execute specific tests use `phpunit --filter` or `phpunit tests/unit/module1`
- Use the strict mode from the get go and never turn it off.
- Imho: Don't have a test namespace. Nicer to read and shows production usage.

DISCUSSION

Issues?

Feature requests?

Anything I missed?

THANK YOU



**KEEP
CALM**

AND

**CONTINUE
TESTING**



