

Unit Tests with PHPUnit

What are unit tests, why and how to use them

Guillaume Renoult - SydPHP

March 2020

What are unit tests?

Anatomy of a unit test

Live example / Demo

Tips and philosophy around unit tests

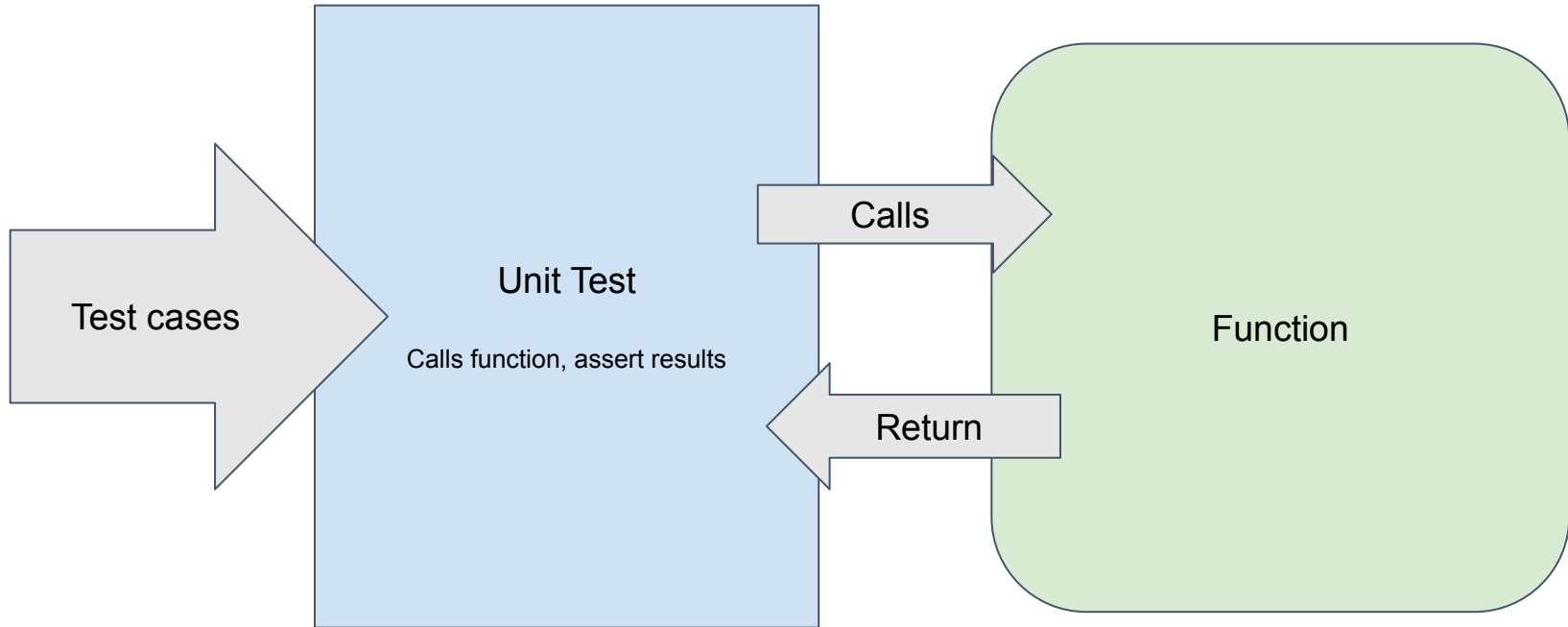
Common excuses

Going further

References / Questions

- What: It's testing individual units/components of a software.
 - A unit is the smallest testable part of any software. Usually a function: we pass an input and expect an output.
- Purpose: to validate that each unit of the software performs as designed.

```
// Typical example of a function to unit test: no dependency and we can
// easily give input and check output.
function validate($phoneNumber)
{
    return strlen($phoneNumber) == 10 && is_numeric($phoneNumber);
}
```



Test function must start with “test” to be picked up by PHPUnit

Other assertions:
assertSame(),
assertCount(),
assertGreaterThan(),
...

Checking output

```
public function testValidate()  
{  
    $validator = new \V2\PhoneNumberValidator();  
    $result = $validator->validate('0469 853 621');  
    $this->assertTrue($result);  
  
    $result = $validator->validate('0469853621');  
    $this->assertTrue($result);  
  
    $result = $validator->validate('abc');  
    $this->assertFalse($result);  
}
```

Calling function with input

Git Repository: <https://github.com/grenoult/sydneyphp-12-03-2020>

```
$ phpunit --testdox
PHPUnit 6.5.5 by Sebastian Bergmann and contributors.

PhoneNumberValidatorV1
[x] Validate

V2\PhoneNumberValidatorV2
[ ] Validate

V3\PhoneNumberValidatorV3
[x] Validate Test phone number with spaces
[x] Validate Test phone number without spaces
[x] Validate Test invalid phone number (string)
```

“Will my function still work well if I touch it? 🤔”



“Will my function still work well if I touch something else (dependencies)? 🤔”



- Add unit tests around what is complex and doesn't give you confidence.
- Do not put unit tests on simple code (like getters, setters). This is an overkill.

- Dealing with dependencies? Use [Mockery](#)!

```
class PhoneLibrary {
    public function getCountryExtensions() {
        // Do some complex stuff including remote API calls and database requests...
    }
}

class PhoneNumberValidator
{
    /** @var PhoneLibrary $phoneLibrary */
    private $phoneLibrary;

    public function __construct($phoneLibrary)
    {
        $this->phoneLibrary = $phoneLibrary;
    }

    function validate($phoneExtension, $phoneNumber)
    {
        $extensionExists = in_array($phoneExtension, $this->phoneLibrary->getCountryExtensions());
        // ...
    }
}

class PhoneNumberValidatorTest
{
    public function testValidate()
    {
        $phoneLibrary = \Mockery::mock(PhoneLibrary::class); // Mock dependencies!
        $validator = new PhoneNumberValidator($phoneLibrary);
        // ... do unit tests
    }
}
```

- Go TDD: Test Driven Development:
 - build unit test first (they will be obviously failing),
 - write enough (i.e. bad) code for the test to pass,
 - refactor your code to be maintainable and sustainable while your tests pass.

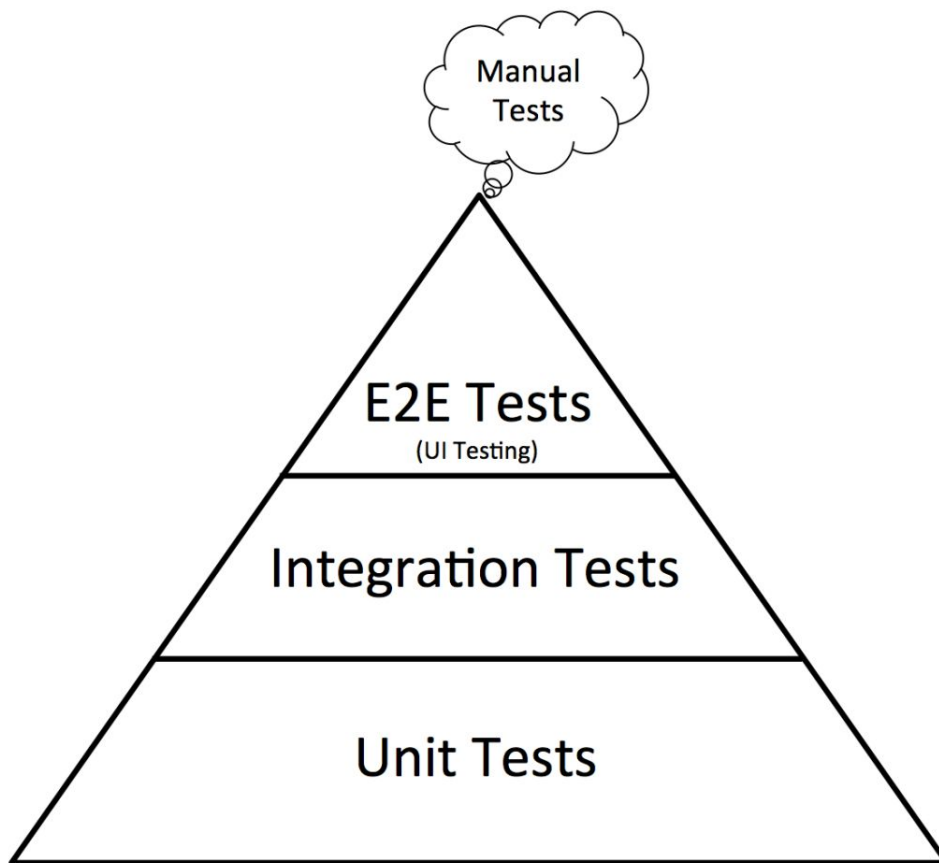
- Go FIRST!
 - Fast
 - Isolated / independant
 - Repeatable
 - Self-validating
 - Thorough

<https://hackernoon.com/test-f-i-r-s-t-65e42f3adc17>

- “I don’t have time to do unit tests!”
 - 1 day of testing now will save you 1 week in the future

- “My manager won’t allow me to spend time on unit tests!”
 - Have a conversation with your manager about the risk of **not** doing unit tests.
 - Unit tests will save your and your workmates time in the future.
 - Unit tests will save your reputation as a business.

- “I don’t do tests, I’m not a QA (Quality Assurance)!”
 - You (as a developer) are responsible of the quality you ship.
 - QAs should help you think about how you can test your code (assistance), not necessarily do the testing (assurance).



Unit tests fundamentals: <http://softwaretestingfundamentals.com/unit-testing/>

The FIRST approach: <https://hackernoon.com/test-f-i-r-s-t-65e42f3adc17>

PHPUnit with Symfony: <https://symfonycasts.com/screencast/phpunit>

TDD: <https://www.freecodecamp.org/news/test-driven-development-what-it-is-and-what-it-is-not-41fa6bca02a2/>

End to end tests with CypressJS: https://www.youtube.com/watch?v=V9KBT_H1kpk

