Driving Development with PhpSpec

with Ciaran McNulty

PHPLondon November 2014

My experiences

- Unit testing since 2004
- Test Driven Development since 2005(ish)
- Behaviour Driven Development since 2012

TDD vs BDD

(or are they the same?)

BDD is a second-generation, outside-in, pull-based, multiple-stakeholder...

Dan North

...multiple-scale, high-automation, agile methodology.

★

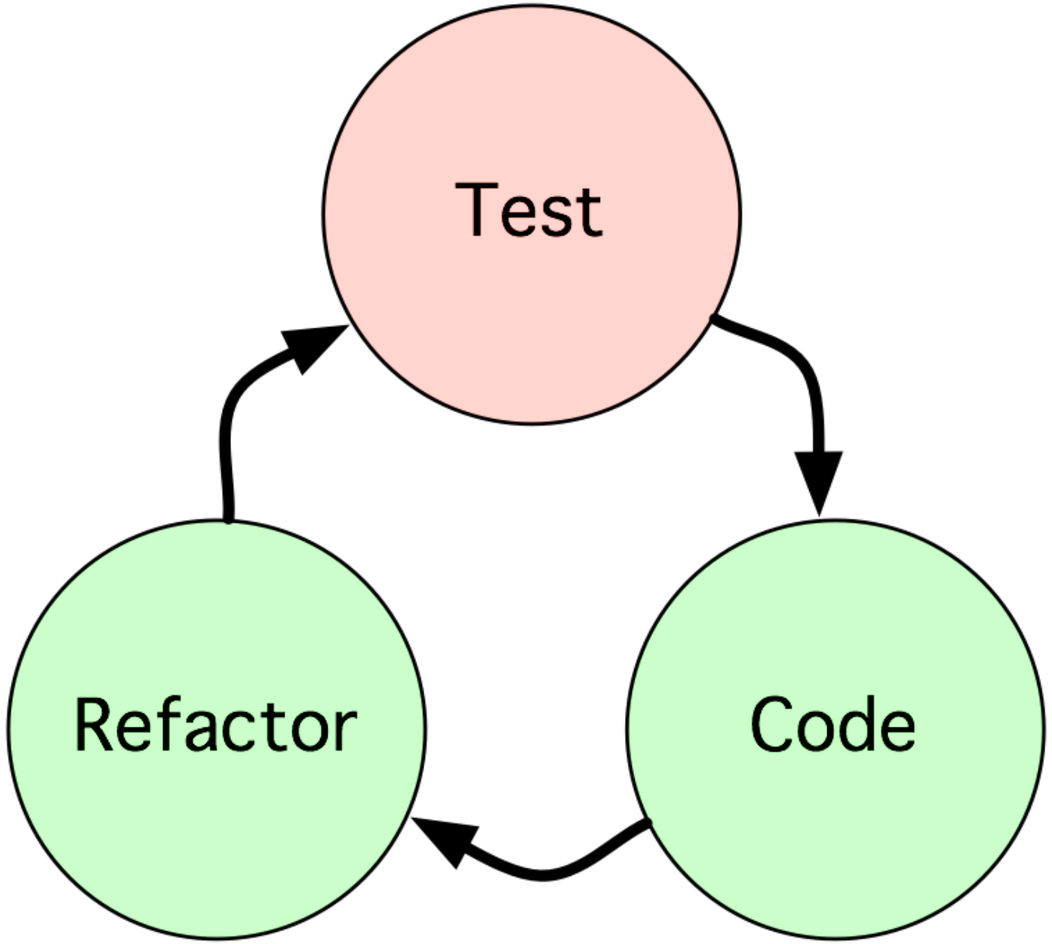
Dan North

BDD is the art of using examples in conversation to illustrate behaviour

Liz Keogh

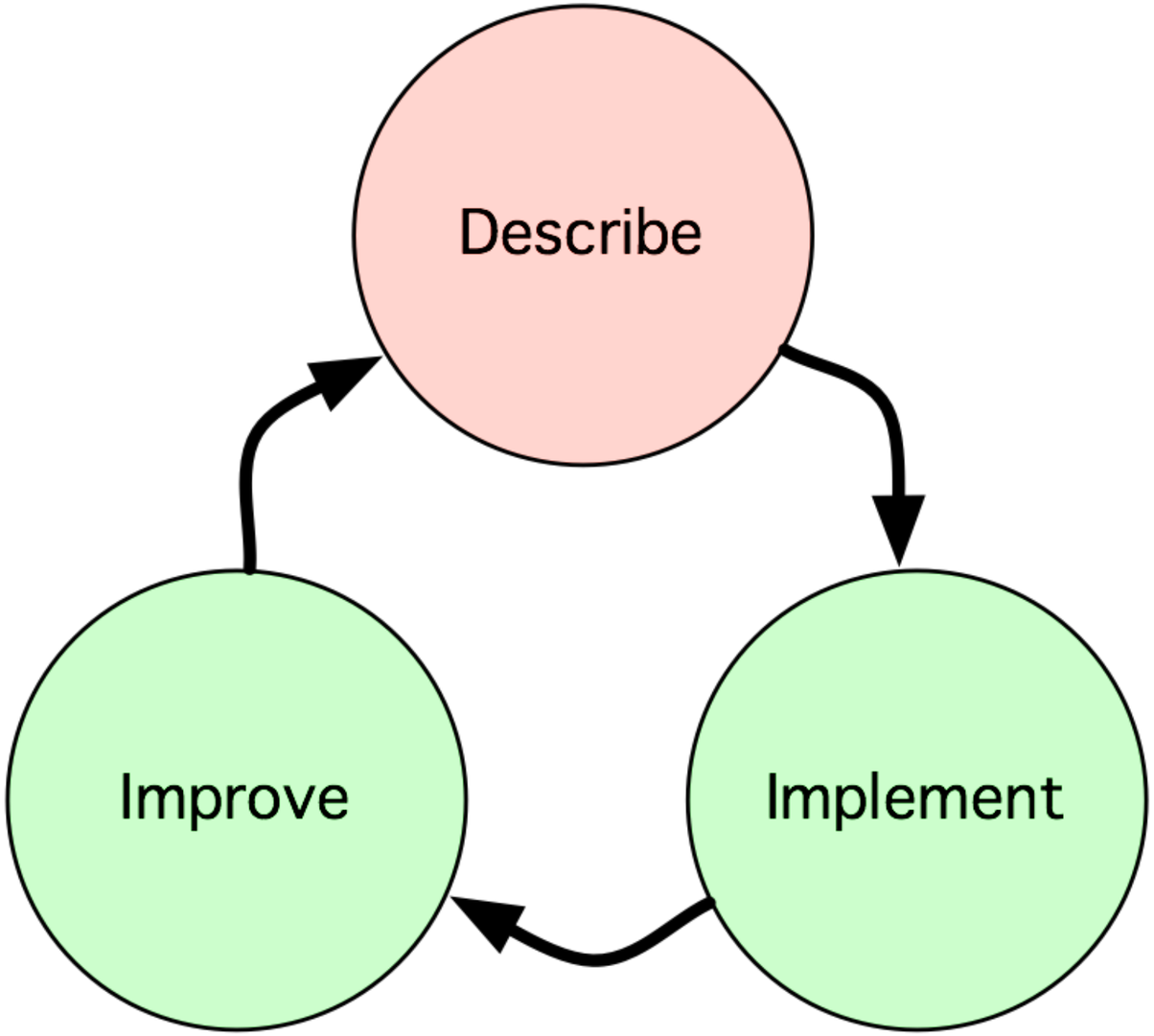
Test Driven Development

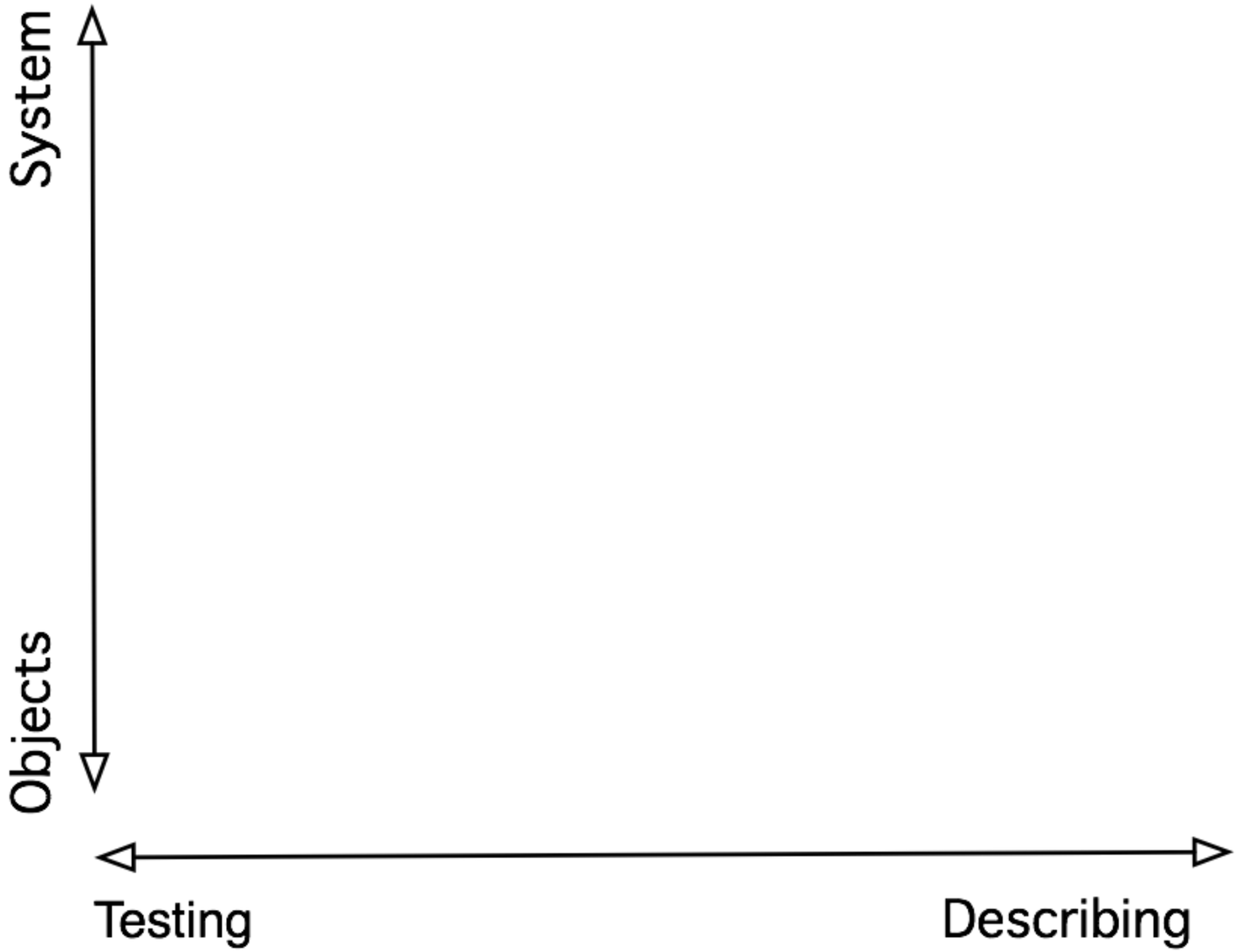
- Before you write your code, **write a test that validates how it should behave**
- After you have written the code, **see if it passes the test**

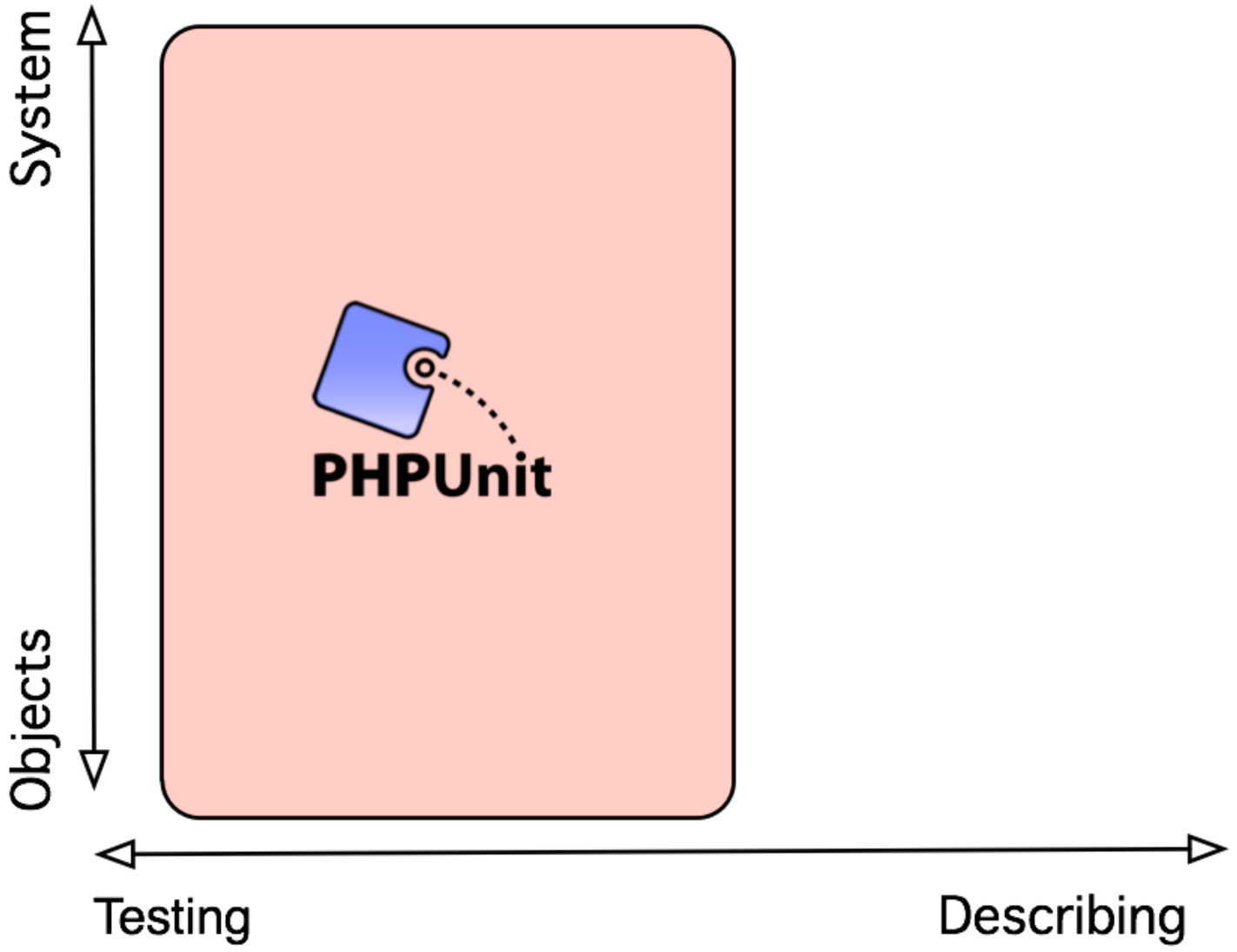


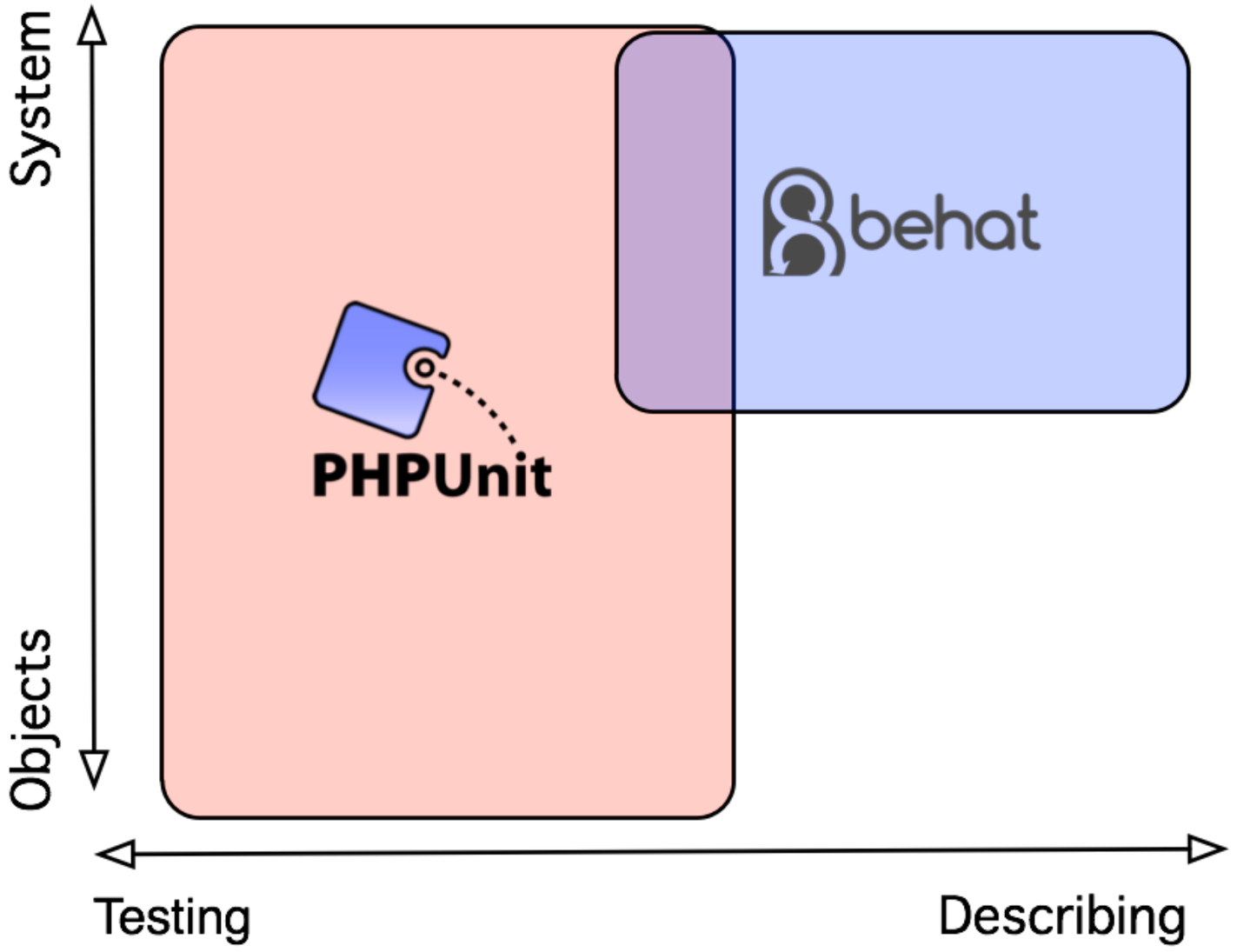
Behaviour Driven Development

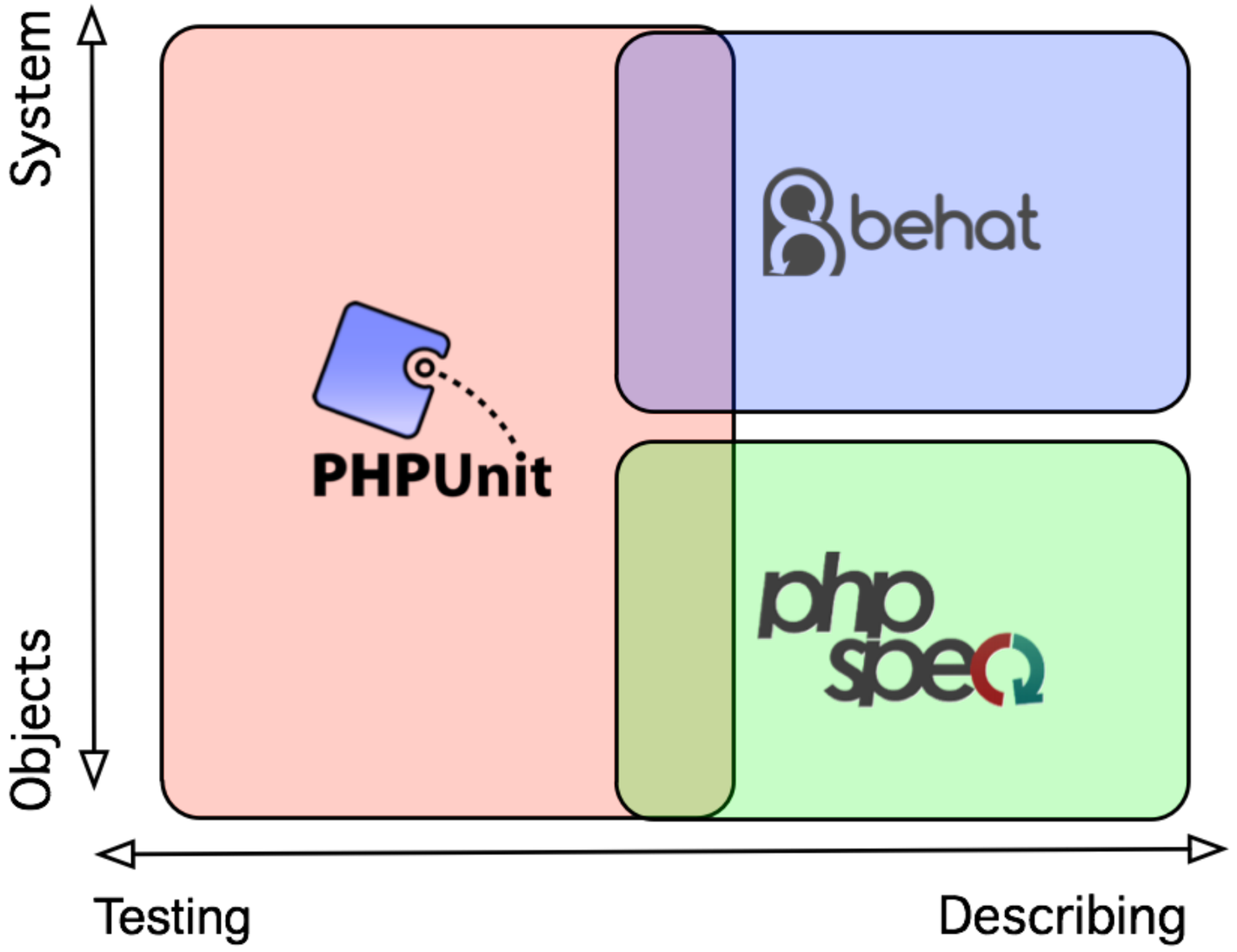
- Before you write your code, **describe how it should behave using examples**
- Then, **Implement the behaviour you you described**




System
Objects
Testing
Describing


System
Objects
PHPUnit
Testing
Describing


System
Objects
PHPUnit
behat
Testing
Describing


System
Objects
PHPUnit
behat
phpspec
Testing
Describing

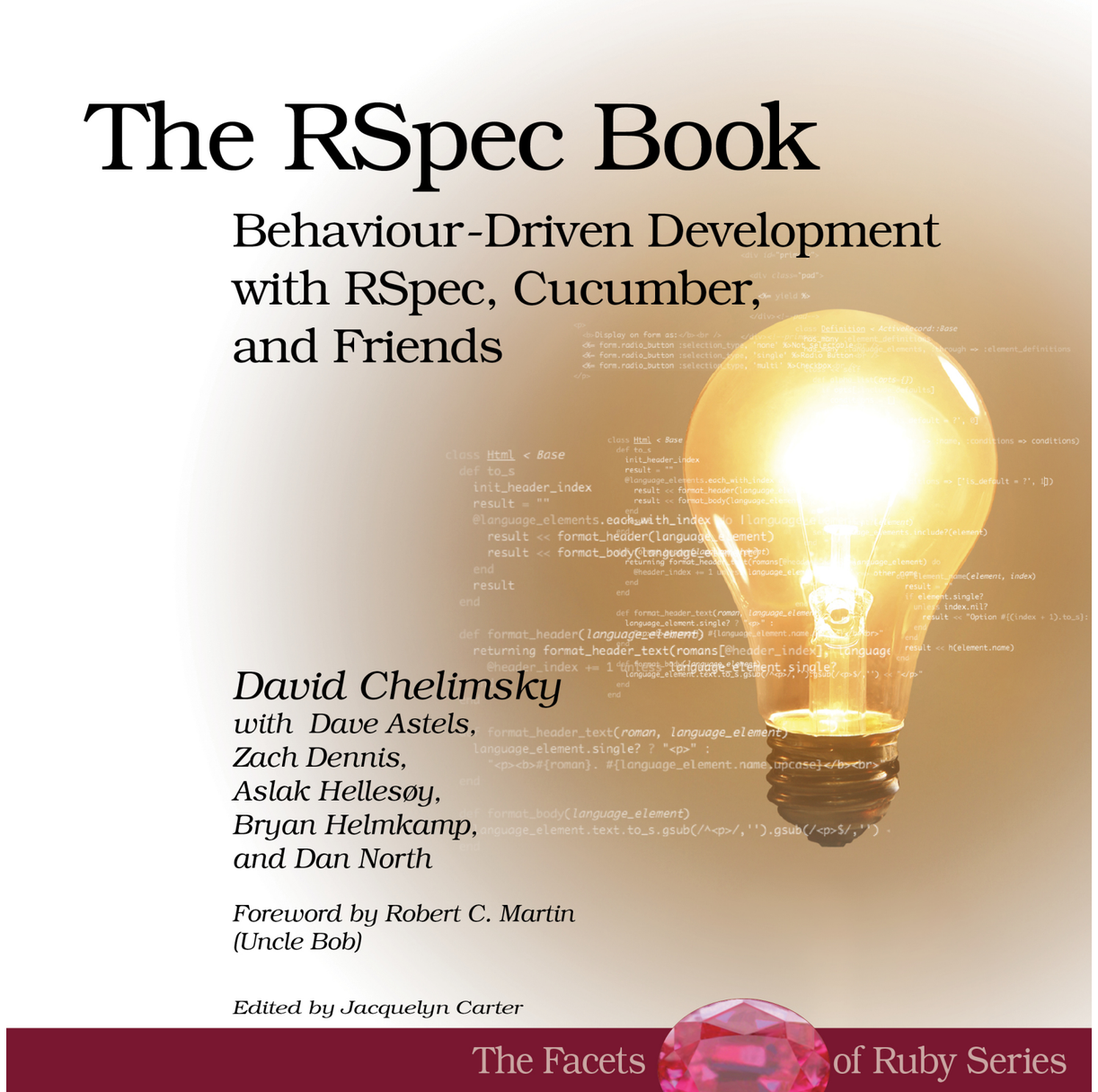
SpecBDD with PhpSpec

Describing individual classes

History

1.0 - Inspired by RSpec

- Pádraic Brady and Travis Swicegood



History

2.0beta - Inspired by 1.0

- Marcello Duarte and Konstantin Kudryashov (Everzet)
- Ground-up rewrite
- No BC in specs

History

2.0 stable - The boring bits

- Me
- Christophe Coevoet
- Jakub Zalas
- Richard Miller
- Gildas Quéméner

Installation via Composer

```
{
    "require-dev": {
        "phpspec/phpspec": "~2.1-RC1"
    },
    "config": {
        "bin-dir": "bin"
    },
    "autoload": {"psr-0": {"": "src"}}
}
```

```
  - Installing phpdocumentor/reflection-docblock (2.0.3)
    Loading from cache

  - Installing doctrine/instantiator (1.0.4)
    Loading from cache

  - Installing phpspec/prophecy (v1.2.1)
    Loading from cache

  - Installing phpspec/php-diff (v1.0.2)
    Loading from cache

  - Installing phpspec/phpspec (2.1.0-RC1)
    Loading from cache

symfony/event-dispatcher suggests installing symfony/dependency-injection ()
symfony/event-dispatcher suggests installing symfony/http-kernel ()
symfony/console suggests installing psr/log (For using the console logger)
phpdocumentor/reflection-docblock suggests installing dflydev/markdown (1.0.*)
phpdocumentor/reflection-docblock suggests installing erusev/parsedown (~0.7)
phpspec/phpspec suggests installing phpspec/nyan-formatters (~1.0 - Adds Nyan fo
rmatters)
Generating autoload files
PhpLondon>
```

A requirement:

We need something that says hello to people

Describing object behaviour

- We **describe an object** using a **Specification**
- A specification is made up of **Examples** illustrating different scenarios

Usage:

```
phpspec describe [Class]
```

```
PhpLondon> bin/phpspec describe PhpLondon/HelloWorld/Greeter
Specification for PhpLondon\HelloWorld\Greeter created in /Users/ciaranmcnulty/D
ocuments/phpspec-talk/spec/PhpLondon/HelloWorld/GreeterSpec.php.

PhpLondon>
```

/spec/PhpLondon/HelloWorld/GreeterSpec.php

```php
namespace spec\PhpLondon\HelloWorld;

use PhpSpec\ObjectBehavior;
use Prophecy\Argument;

class GreeterSpec extends ObjectBehavior
{
    function it_is_initializable()
    {
        $this->shouldHaveType('PhpLondon\HelloWorld\Greeter');
    }
}
```

Verifying object behaviour

- Compare the real objects' behaviours with the examples

Usage:

```
phpspec run
```

```
PhpLondon> bin/phpspec describe PhpLondon/HelloWorld/Greeter
Specification for PhpLondon\HelloWorld\Greeter created in /Users/ciaranmcnulty/D
ocuments/phpspec-talk/spec/PhpLondon/HelloWorld/GreeterSpec.php.

PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Greeter
    ! it is initializable
      class PhpLondon\HelloWorld\Greeter does not exist.

                          100%                          1
1 specs
1 example (1 broken)
38ms

  Do you want me to create `PhpLondon\HelloWorld\Greeter` for you?
                                                        [Y/n]
```

```
Specification for PhpLondon\HelloWorld\Greeter created in /Users/ciaranmcnulty/D
ocuments/phpspec-talk/spec/PhpLondon/HelloWorld/GreeterSpec.php.

PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Greeter
      ! it is initializable
        class PhpLondon\HelloWorld\Greeter does not exist.

                            100%                            1
1 specs
1 example (1 broken)
38ms

  Do you want me to create `PhpLondon\HelloWorld\Greeter` for you?
                                                            [Y/n] Y
Class PhpLondon\HelloWorld\Greeter created in /Users/ciaranmcnulty/Documents/php
spec-talk/src/PhpLondon/HelloWorld/Greeter.php.

                            100%                            1
1 specs
1 example (1 passed)
26ms
PhpLondon>
```

/src/PhpLondon/HelloWorld/Greeter.php

```php
namespace PhpLondon\HelloWorld;

class Greeter
{
}
```

An example for Greeter:

When this greets, it should return "Hello"

/spec/PhpLondon/HelloWorld/GreeterSpec.php

```
class GreeterSpec extends ObjectBehavior
{
    function it_greets_by_saying_hello()
    {
        $this->greet()->shouldReturn('Hello');
    }
}
```

```
PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Greeter
      ! it greets by saying hello
        method PhpLondon\HelloWorld\Greeter::greet not found.

                                100%                                1
1 specs
1 example (1 broken)
41ms

  Do you want me to create `PhpLondon\HelloWorld\Greeter::greet()` for you?
                                                                [Y/n]
```


PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Greeter
! it greets by saying hello
method PhpLondon\HelloWorld\Greeter::greet not found.
100% 1
1 specs
1 example (1 broken)
41ms
Do you want me to create `PhpLondon\HelloWorld\Greeter::greet()` for you?
[Y/n]
Method PhpLondon\HelloWorld\Greeter::greet() has been created.
PhpLondon/HelloWorld/Greeter
✘ it greets by saying hello
expected "Hello", but got null.
100% 1
1 specs
1 example (1 failed)
44ms
PhpLondon>

/src/PhpLondon/HelloWorld/Greeter.php

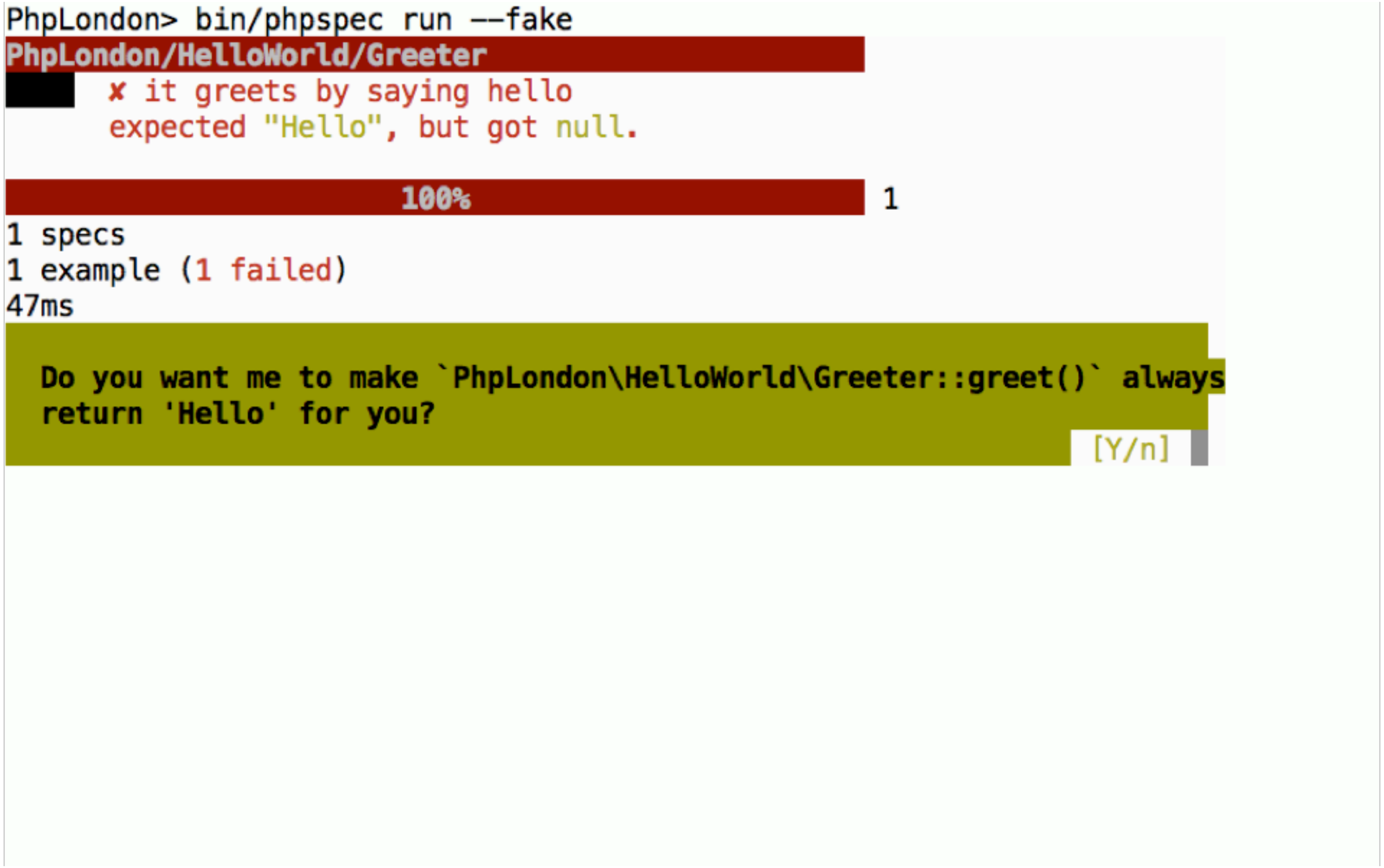
```
class Greeter
{
    public function greet()
    {
        // TODO: write logic here
    }
}
```

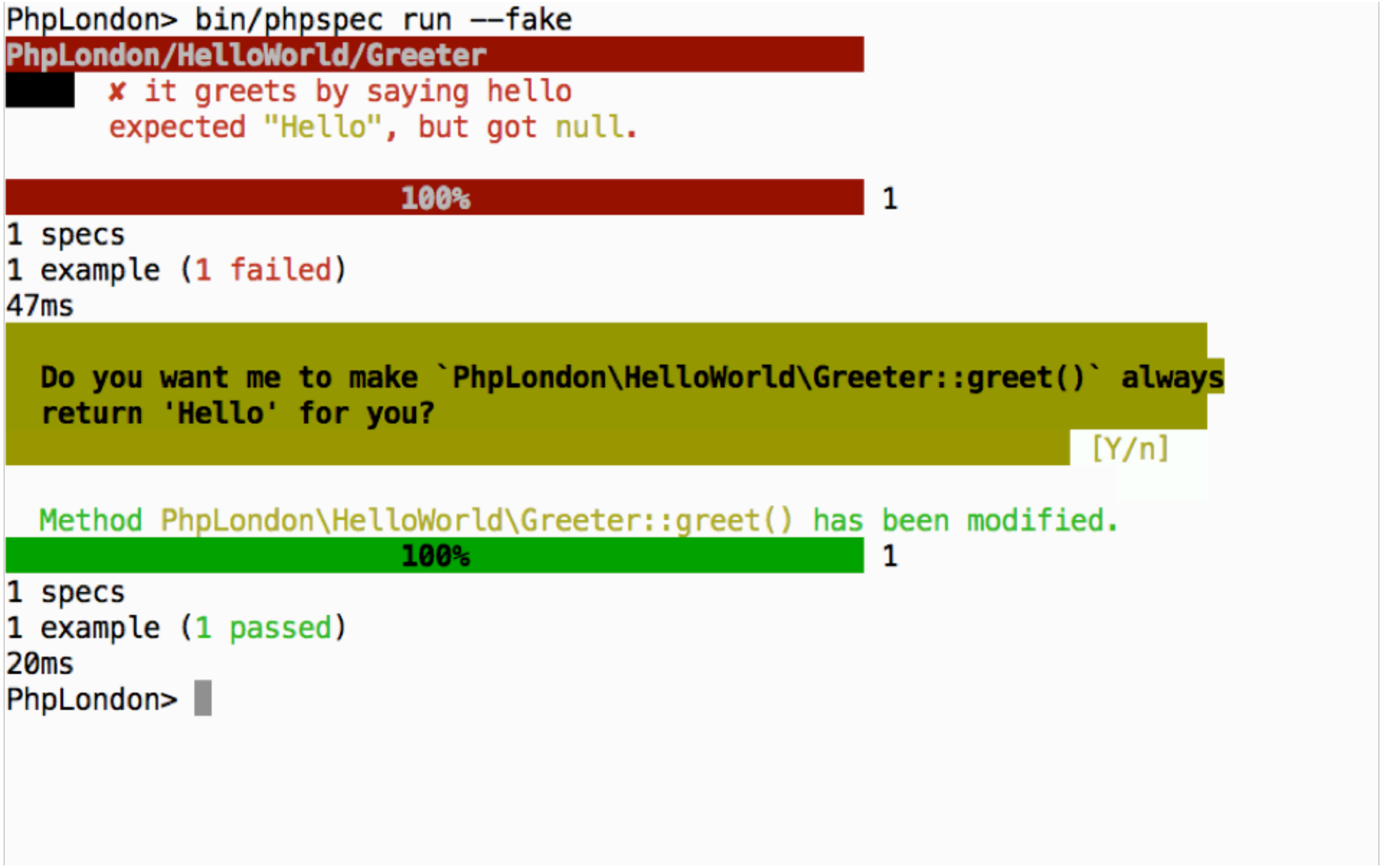
So now I write some code?

Fake it till you make it

- Do the simplest thing that works
- Only add complexity later when more examples drive it

```
phpspec run --fake
```


PhpLondon> bin/phpspec run --fake
PhpLondon/HelloWorld/Greeter
✘ it greets by saying hello
expected "Hello", but got null.
100% 1
1 specs
1 example (1 failed)
47ms
Do you want me to make `PhpLondon\HelloWorld\Greeter::greet()` always return 'Hello' for you?
[Y/n]


PhpLondon> bin/phpspec run --fake
PhpLondon/HelloWorld/Greeter
✘ it greets by saying hello
expected "Hello", but got null.
100% 1
1 specs
1 example (1 failed)
47ms
Do you want me to make `PhpLondon\HelloWorld\Greeter::greet()` always return 'Hello' for you?
[Y/n]
Method PhpLondon\HelloWorld\Greeter::greet() has been modified.
100% 1
1 specs
1 example (1 passed)
20ms
PhpLondon>

/src/PhpLondon/HelloWorld/Greeter.php

```
class Greeter
{
    public function greet()
    {
        return 'Hello';
    }
}
```

Describing values

Matchers

Describing values - Equality

```
$this->greet()->shouldReturn('Hello');

$this->sum(3,3)->shouldEqual(6);

$user = $this->findById(1234);
$user->shouldBe($expectedUser);

$this->numberList()
     ->shouldBeLike(new ArrayObject([1,2,3]));
```

Describing values - Type

```
$this->address()->shouldHaveType('EmailAddress');

$this->getTime()->shouldReturnAnInstanceOf('DateTime');

$user = $this->findById(1234);
$user->shouldBeAnInstanceOf('User');

$this->shouldImplement('Countable');
```

Describing values - Strings

```
$this->getStory()->shouldStartWith('A long time ago');
$this->getStory()->shouldEndWith('happily ever after');

$this->getSlug()->shouldMatch('/^[0-9a-z]+$/');
```

Describing values - Arrays

```
$this->getNames()->shouldContain('Tom');

$this->getNames()->shouldHaveKey(0);

$this->getNames()->shouldHaveCount(1);
```

Describing values - object state

```
// calls isAdmin()
$this->getUser()->shouldBeAdmin();

// calls hasLoggedInUser()
$this->shouldHaveLoggedInUser();
```

Describing custom values

```
function it_gets_json_with_user_details()
{
    $this->getResponseData()->shouldHaveJsonKey('username');
}

public function getMatchers()
{
    return [
        'haveJsonKey' => function ($subject, $key) {
            return array_key_exists($key, json_decode($subject));
        }
    ];
}
```

Another example for Greeter:

When this greets Bob, it should return "Hello, Bob"

Wait, what is Bob?

Bob is a Person

What is a Person?

```
PhpLondon> bin/phpspec describe PhpLondon/HelloWorld/Person
Specification for PhpLondon\HelloWorld\Person created in /Users/ciaranmcnulty/Do
cuments/phpspec-talk/spec/PhpLondon/HelloWorld/PersonSpec.php.

PhpLondon>
```

An example for a Person:

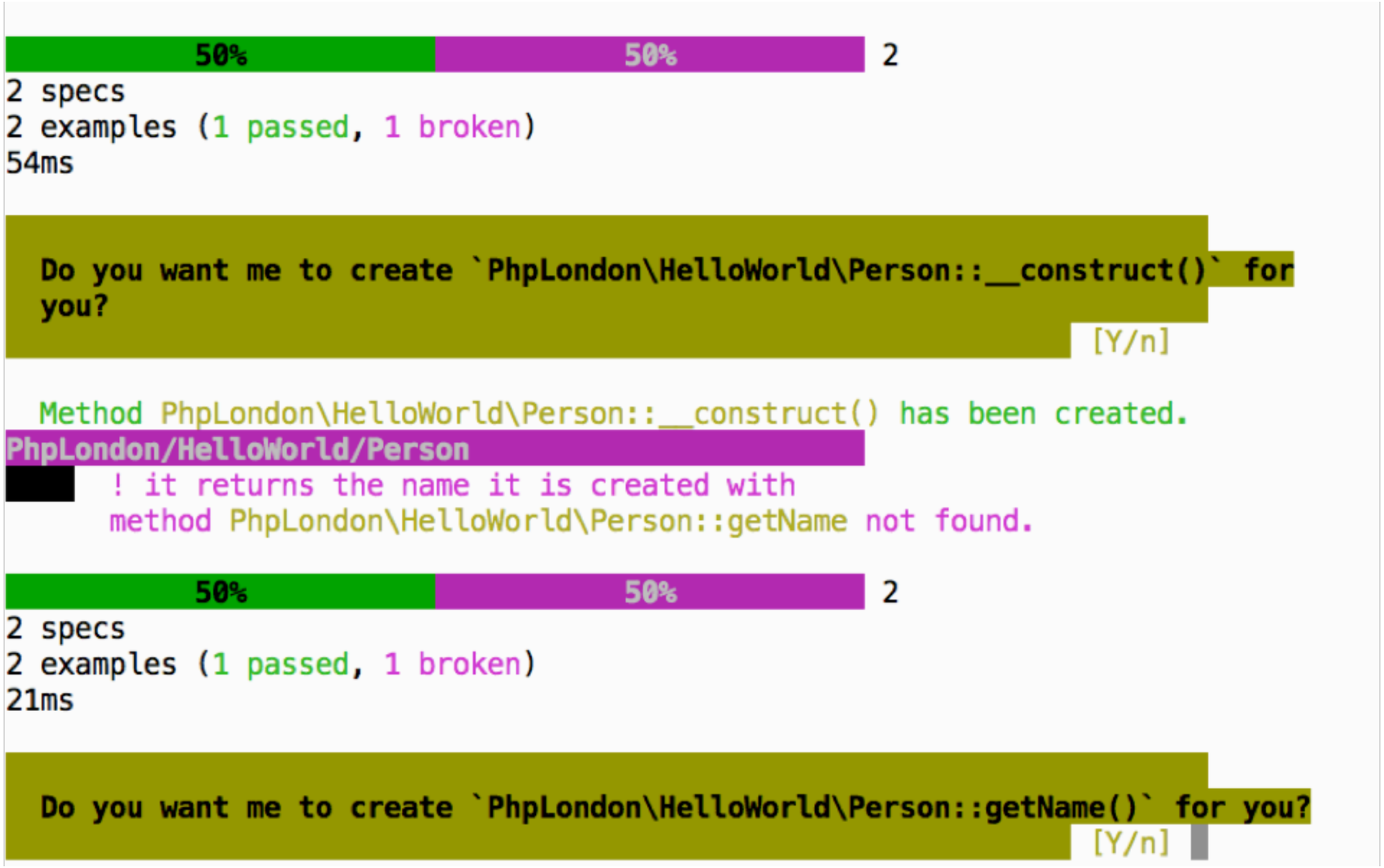
When you ask a person named "Alice" for their name, they return "Alice"

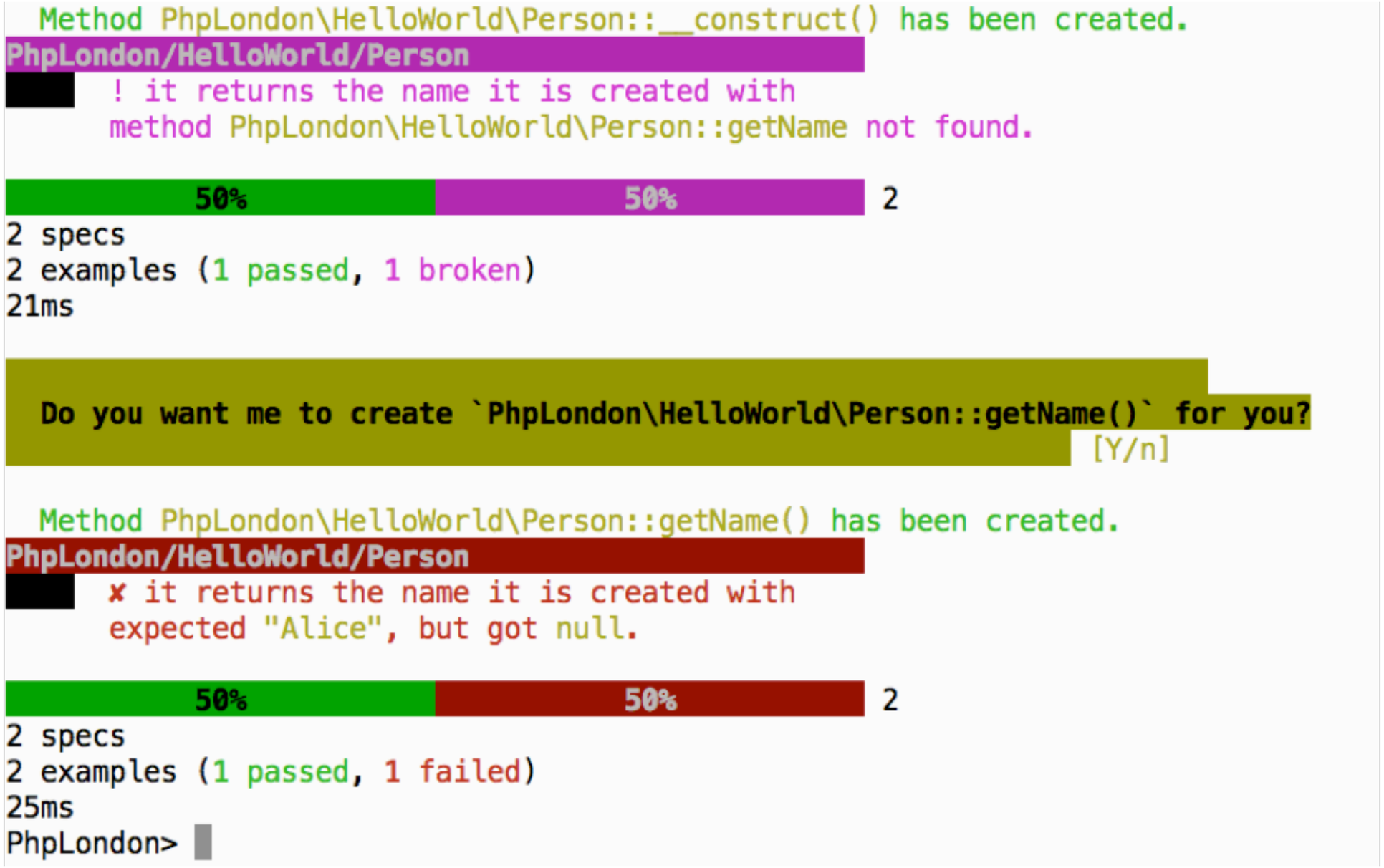
/spec/PhpLondon/HelloWorld/PersonSpec.php

```
class PersonSpec extends ObjectBehavior
{
    function it_returns_the_name_it_is_created_with()
    {
        $this->beConstructedWith('Alice');

        $this->getName()->shouldReturn('Alice');
    }
}
```

```
PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Person
      ! it returns the name it is created with
        class PhpLondon\HelloWorld\Person does not exist.

           50%                  50%                  2
2 specs
2 examples (1 passed, 1 broken)
48ms

  Do you want me to create `PhpLondon\HelloWorld\Person` for you?
                                                      [Y/n]
```

```
50%                    50%                    2
2 specs
2 examples (1 passed, 1 broken)
48ms

  Do you want me to create `PhpLondon\HelloWorld\Person` for you?
                                                                [Y/n]
Class PhpLondon\HelloWorld\Person created in /Users/ciaranmcnulty/Documents/phps
pec-talk/src/PhpLondon/HelloWorld/Person.php.

PhpLondon/HelloWorld/Person
  ! it returns the name it is created with
    method PhpLondon\HelloWorld\Person::__construct not found.

50%                    50%                    2
2 specs
2 examples (1 passed, 1 broken)
54ms

  Do you want me to create `PhpLondon\HelloWorld\Person::__construct()` for
  you?
                                                                [Y/n]
```


50% 50% 2
2 specs
2 examples (1 passed, 1 broken)
54ms
Do you want me to create `PhpLondon\HelloWorld\Person::__construct()` for you?
[Y/n]
Method PhpLondon\HelloWorld\Person::__construct() has been created.
PhpLondon/HelloWorld/Person
! it returns the name it is created with
method PhpLondon\HelloWorld\Person::getName not found.
50% 50% 2
2 specs
2 examples (1 passed, 1 broken)
21ms
Do you want me to create `PhpLondon\HelloWorld\Person::getName()` for you?
[Y/n]


Method PhpLondon\HelloWorld\Person::__construct() has been created.
PhpLondon/HelloWorld/Person
! it returns the name it is created with
method PhpLondon\HelloWorld\Person::getName not found.
50%
50%
2
2 specs
2 examples (1 passed, 1 broken)
21ms
Do you want me to create `PhpLondon\HelloWorld\Person::getName()` for you?
[Y/n]
Method PhpLondon\HelloWorld\Person::getName() has been created.
PhpLondon/HelloWorld/Person
✘ it returns the name it is created with
expected "Alice", but got null.
50%
50%
2
2 specs
2 examples (1 passed, 1 failed)
25ms
PhpLondon>

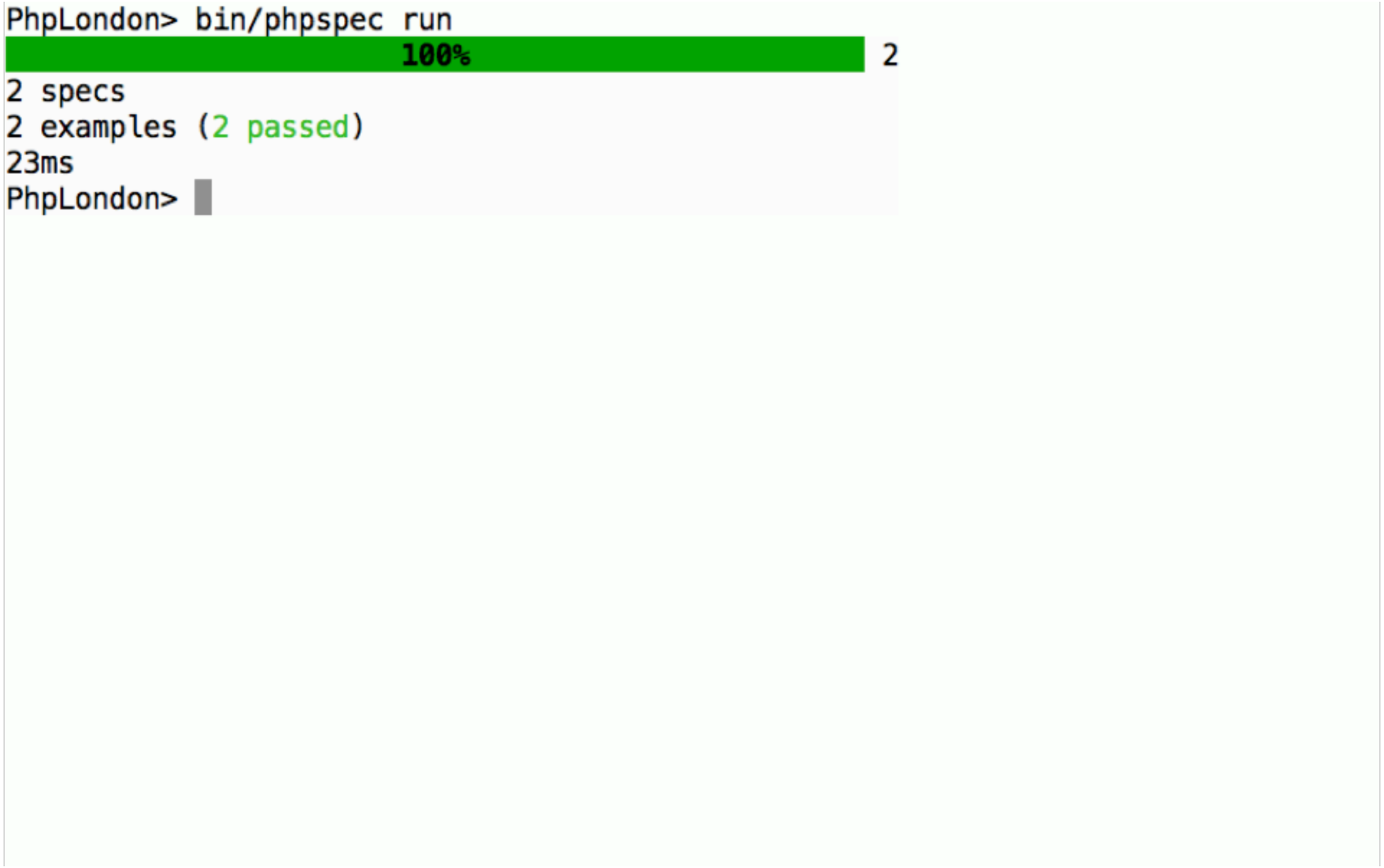
/src/PhpLondon/HelloWorld/Person.php

```php
class Person
{
    public function __construct($argument1)
    {
        // TODO: write logic here
    }

    public function getName()
    {
        // TODO: write logic here
    }
}
```

So now I write some code!

/src/PhpLondon/HelloWorld/Person.php

```php
class Person
{
    private $name;

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

    public function getName()
    {
        return $this->name;
    }
}
```


PhpLondon> bin/phpspec run
100%
2
2 specs
2 examples (2 passed)
23ms
PhpLondon>

Another example for a Person:

When a person named "Alice" changes their name to "Bob", when you ask their name they return "Bob"

/spec/PhpLondon/HelloWorld/PersonSpec.php

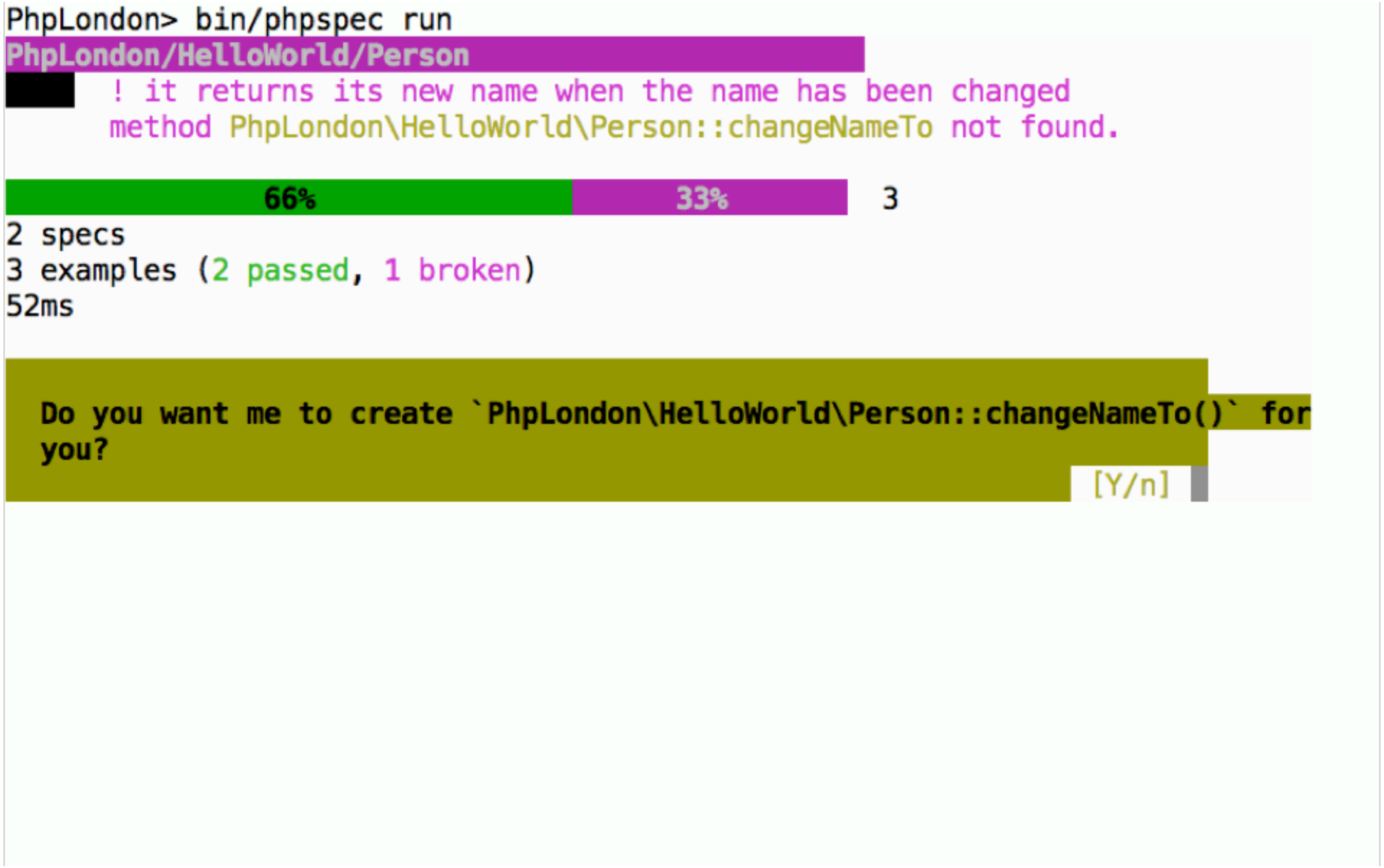
```
class PersonSpec extends ObjectBehavior
{
    function it_returns_the_name_it_is_created_with()
    {
        $this->beConstructedWith('Alice');
        $this->getName()->shouldReturn('Alice');
    }
}
```

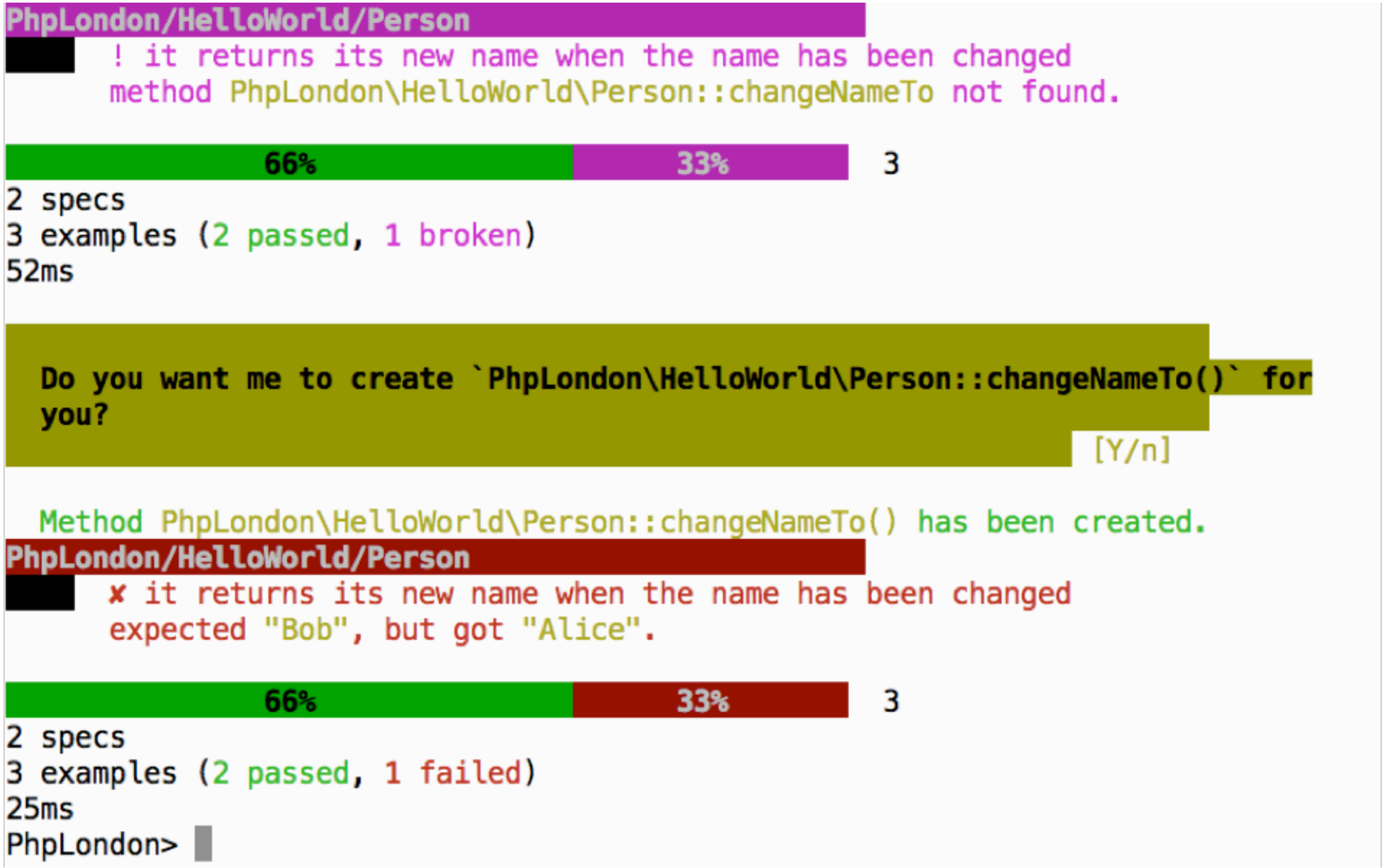
/spec/PhpLondon/HelloWorld/PersonSpec.php

```
class PersonSpec extends ObjectBehavior
{
    function let()
    {
        $this->beConstructedWith('Alice');
    }

    function it_returns_the_name_it_is_created_with()
    {
        $this->getName()->shouldReturn('Alice');
    }
}
```

/spec/PhpLondon/HelloWorld/PersonSpec.php

```
class PersonSpec extends ObjectBehavior
{
    function let()
    {
        $this->beConstructedWith('Alice');
    }

    // …

    function it_returns_its_new_name_when_the_name_has_been_changed()
    {
        $this->changeNameTo('Bob');

        $this->getName()->shouldReturn('Bob');
    }
}
```


PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Person
! it returns its new name when the name has been changed
method PhpLondon\HelloWorld\Person::changeNameTo not found.
66%
33%
3
2 specs
3 examples (2 passed, 1 broken)
52ms
Do you want me to create `PhpLondon\HelloWorld\Person::changeNameTo()` for you?
[Y/n]

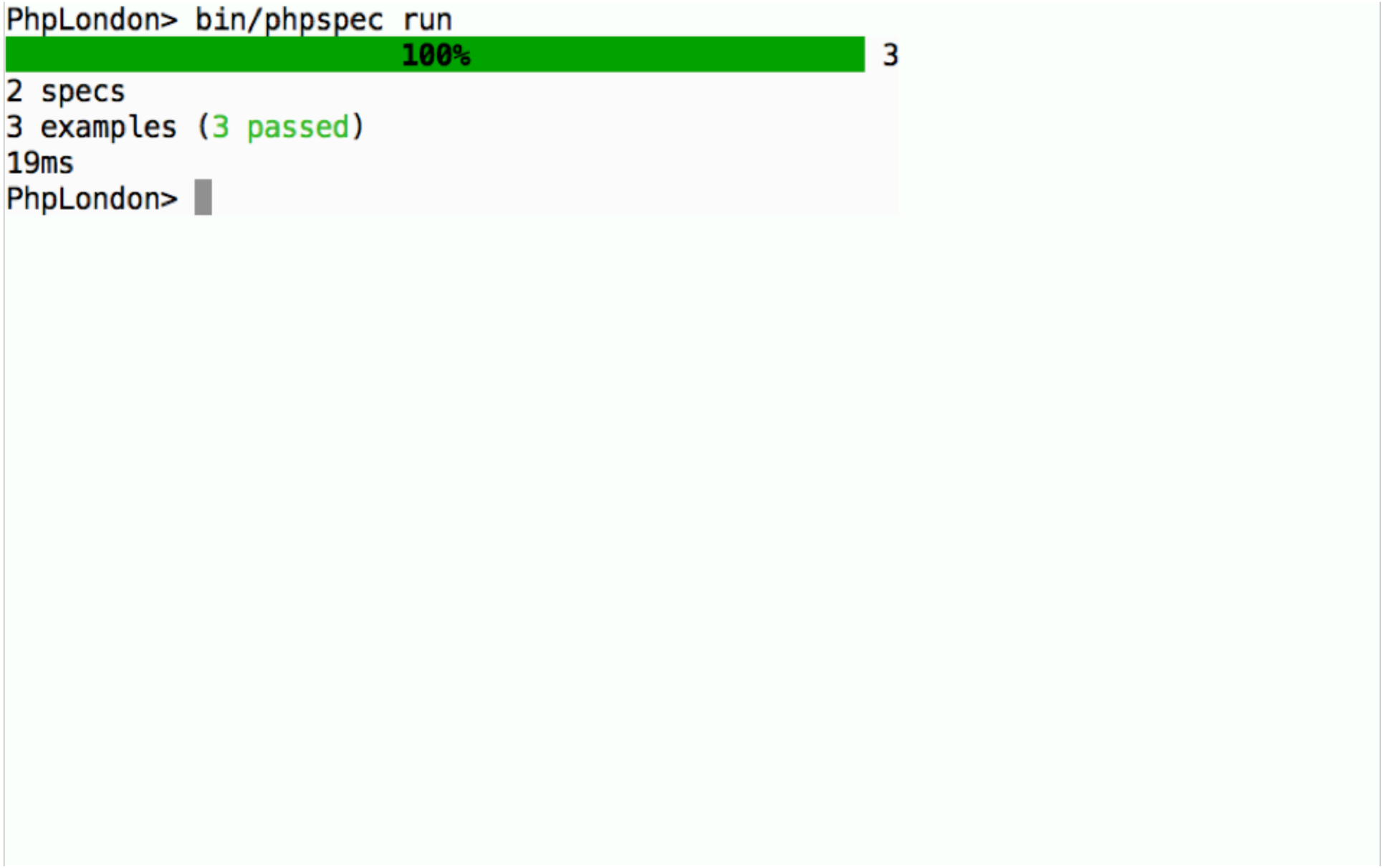

PhpLondon/HelloWorld/Person
! it returns its new name when the name has been changed
method PhpLondon\HelloWorld\Person::changeNameTo not found.
66%
33%
3
2 specs
3 examples (2 passed, 1 broken)
52ms
Do you want me to create `PhpLondon\HelloWorld\Person::changeNameTo()` for you?
[Y/n]
Method PhpLondon\HelloWorld\Person::changeNameTo() has been created.
PhpLondon/HelloWorld/Person
✘ it returns its new name when the name has been changed
expected "Bob", but got "Alice".
66%
33%
3
2 specs
3 examples (2 passed, 1 failed)
25ms
PhpLondon>

/src/PhpLondon/HelloWorld/Person.php

```php
class Person
{
    private $name;

    // …

    public function changeNameTo($argument1)
    {
        // TODO: write logic here
    }
}
```

/src/PhpLondon/HelloWorld/Person.php

```php
class Person
{
    private $name;

    // …

    public function changeNameTo($name)
    {
        $this->name = $name;
    }
}
```


PhpLondon> bin/phpspec run
100%
3
2 specs
3 examples (3 passed)
19ms
PhpLondon>

Another example for Greeter:

When this greets Bob, it should return "Hello, Bob"

Describing collaboration - Stubs

Stubs are used to describe how we interact with objects we **query**

- Maybe it is hard to get the real collaborator to return the value we want
- Maybe using the real collaborator is expensive

/spec/PhpLondon/HelloWorld/GreeterSpec.php

```
class GreeterSpec extends ObjectBehavior
{
    //…

    function it_greets_people_by_name(Person $bob)
    {
        $bob->getName()->willReturn('Bob');

        $this->greet($bob)->shouldReturn('Hello, Bob');
    }
}
```


PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Greeter
✘ it greets people by name
expected "Hello, Bob", but got "Hello".
75%
25%
4
2 specs
4 examples (3 passed, 1 failed)
66ms
PhpLondon>

/src/PhpLondon/HelloWorld/Greeter.php

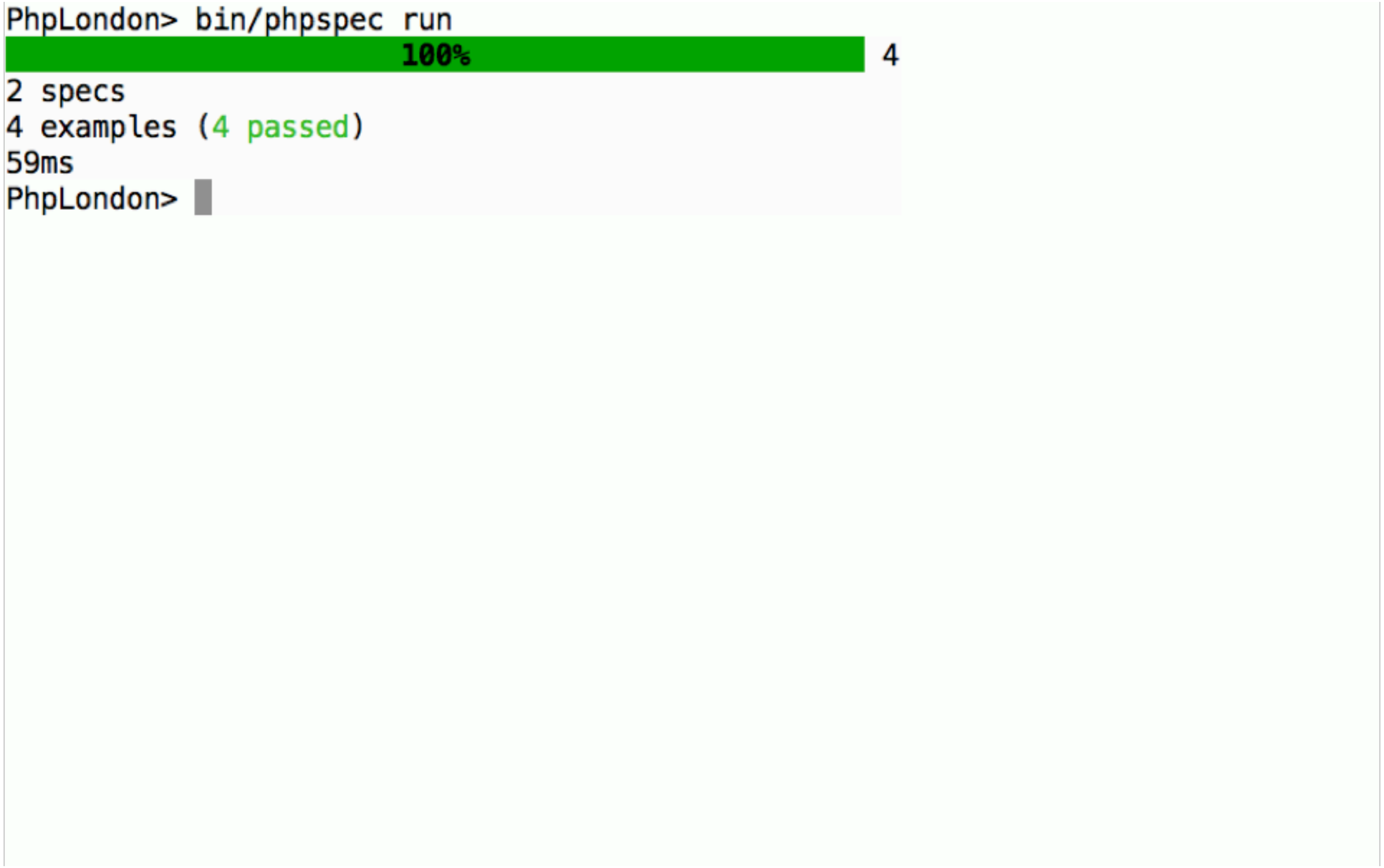
```
class Greeter
{
    public function greet()
    {
        return 'Hello';
    }
}
```

/src/PhpLondon/HelloWorld/Greeter.php

```php
class Greeter
{
    public function greet()
    {
        $greeting = 'Hello';

        return $greeting;
    }
}
```

/src/PhpLondon/HelloWorld/Greeter.php

```php
class Greeter
{
    public function greet(Person $person = null)
    {
        $greeting = 'Hello';

        if ($person) {
            $greeting .= ', ' . $person->getName();
        }

        return $greeting;
    }
}
```


PhpLondon> bin/phpspec run
100%
4
2 specs
4 examples (4 passed)
59ms
PhpLondon>

Final example for Greeter:

When it greets Bob, the message "Hello Bob" should be logged

What's a log?

Let's not worry yet

/src/PhpLondon/HelloWorld/Logger.php

```php
interface Logger
{
    public function log($message);
}
```

Describing collaboration - Mocks and Spies

Mocks or **Spies** are used to describe how we interact with objects we **command**

- → Maybe the real command is has side effects
- → Maybe using the real collaborator is expensive

/spec/PhpLondon/HelloWorld/GreeterSpec.php

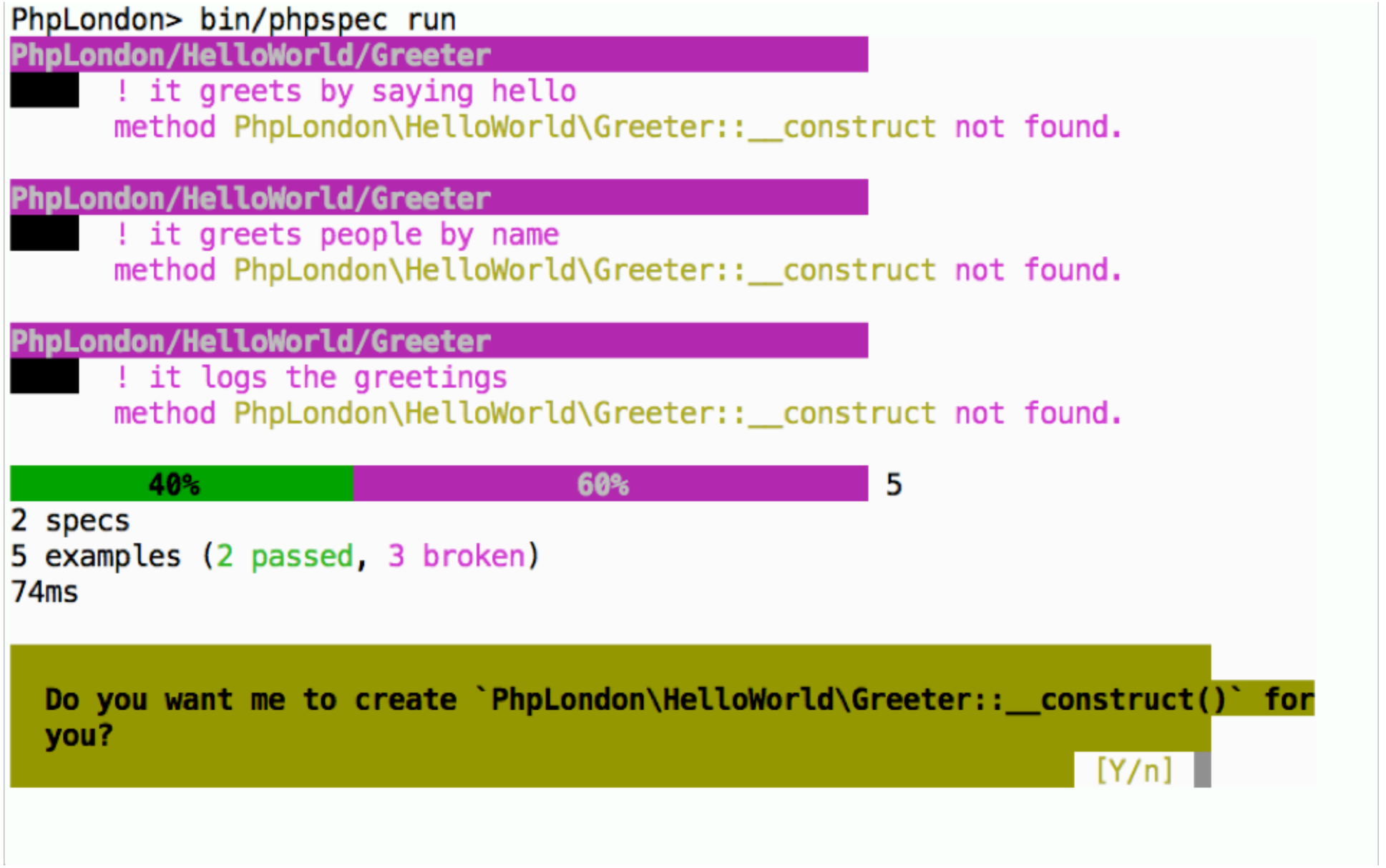
```
class GreeterSpec extends ObjectBehavior
{
    //…

    function it_greets_people_by_name(Person $bob)
    {
        $bob->getName()->willReturn('Bob');
        $this->greet($bob)->shouldReturn('Hello, Bob');
    }
}
```

/spec/PhpLondon/HelloWorld/GreeterSpec.php

```
class GreeterSpec extends ObjectBehavior
{
    function let(Person $bob)
    {
        $bob->getName()->willReturn('Bob');
    }

    //…

    function it_greets_people_by_name(Person $bob)
    {
        $this->greet($bob)->shouldReturn('Hello, Bob');
    }
}
```

/spec/PhpLondon/HelloWorld/GreeterSpec.php

```
class GreeterSpec extends ObjectBehavior
{
    function let(Person $bob, Logger $logger)
    {
        $this->beConstructedWith($logger);
        $bob->getName()->willReturn('Bob');
    }

    //…

    function it_logs_the_greetings(Person $bob, Logger $logger)
    {
        $this->greet($bob);
        $logger->log('Hello, Bob')->shouldHaveBeenCalled();
    }
}
```


PhpLondon> bin/phpspec run
PhpLondon/HelloWorld/Greeter
! it greets by saying hello
method PhpLondon\HelloWorld\Greeter::__construct not found.
PhpLondon/HelloWorld/Greeter
! it greets people by name
method PhpLondon\HelloWorld\Greeter::__construct not found.
PhpLondon/HelloWorld/Greeter
! it logs the greetings
method PhpLondon\HelloWorld\Greeter::__construct not found.
40%
60%
5
2 specs
5 examples (2 passed, 3 broken)
74ms
Do you want me to create `PhpLondon\HelloWorld\Greeter::__construct()` for you?
[Y/n]

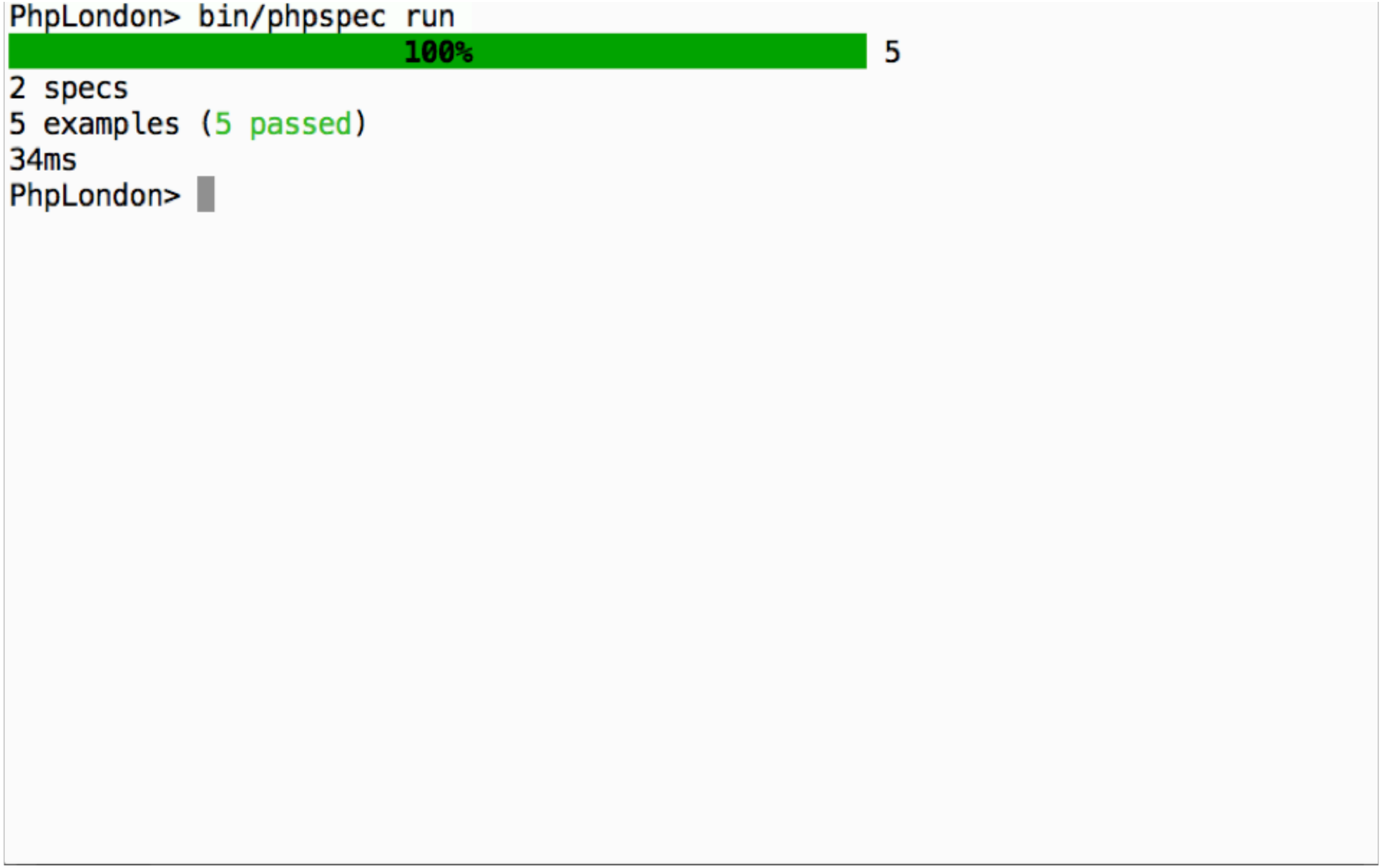
```
      method PhpLondon\HelloWorld\Greeter::__construct not found.

        40%                    60%                    5
2 specs
5 examples (2 passed, 3 broken)
74ms

  Do you want me to create `PhpLondon\HelloWorld\Greeter::__construct()` for
  you?
                                                      [Y/n]

  Method PhpLondon\HelloWorld\Greeter::__construct() has been created.
PhpLondon/HelloWorld/Greeter
      ✘ it logs the greetings
      no calls been made that match:
        Double\Logger\Logger\P5->log(exact("Hello, Bob"))
      but expected at least one.

                    80%                    20%          5
2 specs
5 examples (4 passed, 1 failed)
41ms
PhpLondon>
```

/src/PhpLondon/HelloWorld/Greeter.php

```php
class Greeter
{
    public function __construct($argument1)
    {
        // TODO: write logic here
    }

    public function greet(Person $person = null)
    {
        $greeting = 'Hello';
        if ($person) { $greeting .= ', ' . $person->getName(); }

        return $greeting;
    }
}
```

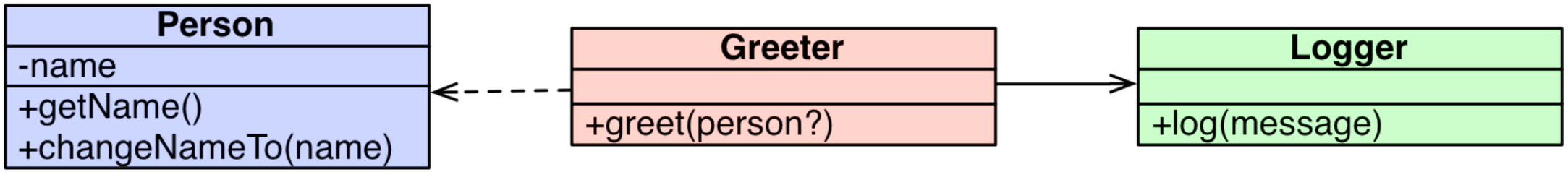
/src/PhpLondon/HelloWorld/Greeter.php

```php
class Greeter
{
    private $logger;

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

    public function greet(Person $person = null)
    {
        $greeting = 'Hello';
        if ($person) { $greeting .= ', ' . $person->getName(); }

        $this->logger->log($greeting);

        return $greeting;
    }
}
```

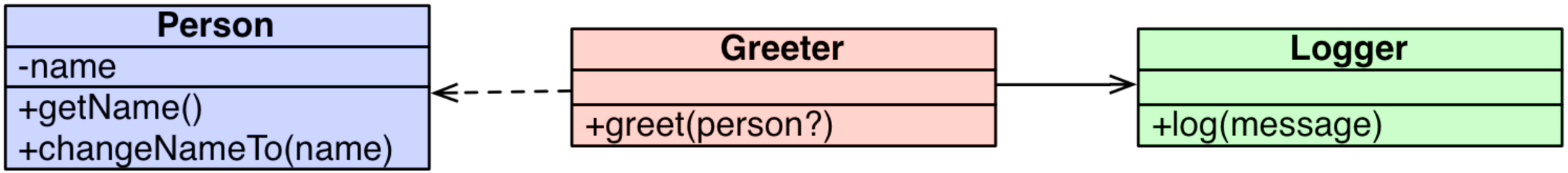

PhpLondon> bin/phpspec run
100%
5
2 specs
5 examples (5 passed)
34ms
PhpLondon>

What have we built?

The domain model



Specs as documentation




```
PhpLondon> bin/phpspec run -f pretty

      PhpLondon\HelloWorld\Greeter

  ✔ greets by saying hello
  ✔ greets people by name
  ✔ logs the greetings

      PhpLondon\HelloWorld\Person

  ✔ returns the name it is created with
  ✔ returns its new name when the name has been changed
```

PhpSpec

- Focuses on being **descriptive**
- Makes common dev activities **easier or automated**
- **Drives** your design

2.1 release - soon!

- Rerun after failure
- `--fake` option
- Named constructors: User::named('Bob')
- PSR-4 support (+ other autoloaders)
- + lots of small improvements

Me

- **Senior Trainer** at Inviqa / Sensio Labs UK / Session Digital
- Contributor to **PhpSpec**
- **@ciaranmcnulty**
- https://github.com/ciaranmcnulty/**phplondon-phpspec-talk**

Questions?