

# PHP UK CONFERENCE

15th & 16th February 2024

[www.phpconference.co.uk](http://www.phpconference.co.uk)

## Dave Liddament

Elevating Legacy: A Case Study  
on the migration from Laravel 4  
to 9



📍 The Brewery, London, UK

@daveliddament



**Dave Liddament**  
@DaveLiddament

...

Finally completed an upgrade from [#laravel](#) 4.2 to 9.

To celebrate Laravel have released version 10!

Remember your tests folks. This upgrade wouldn't have been possible, or as smooth, without tests.

Also [@rectorphp](#) is pretty handy for this.

I must write a blog/talk about it.

+50,509 -69,952



ALT

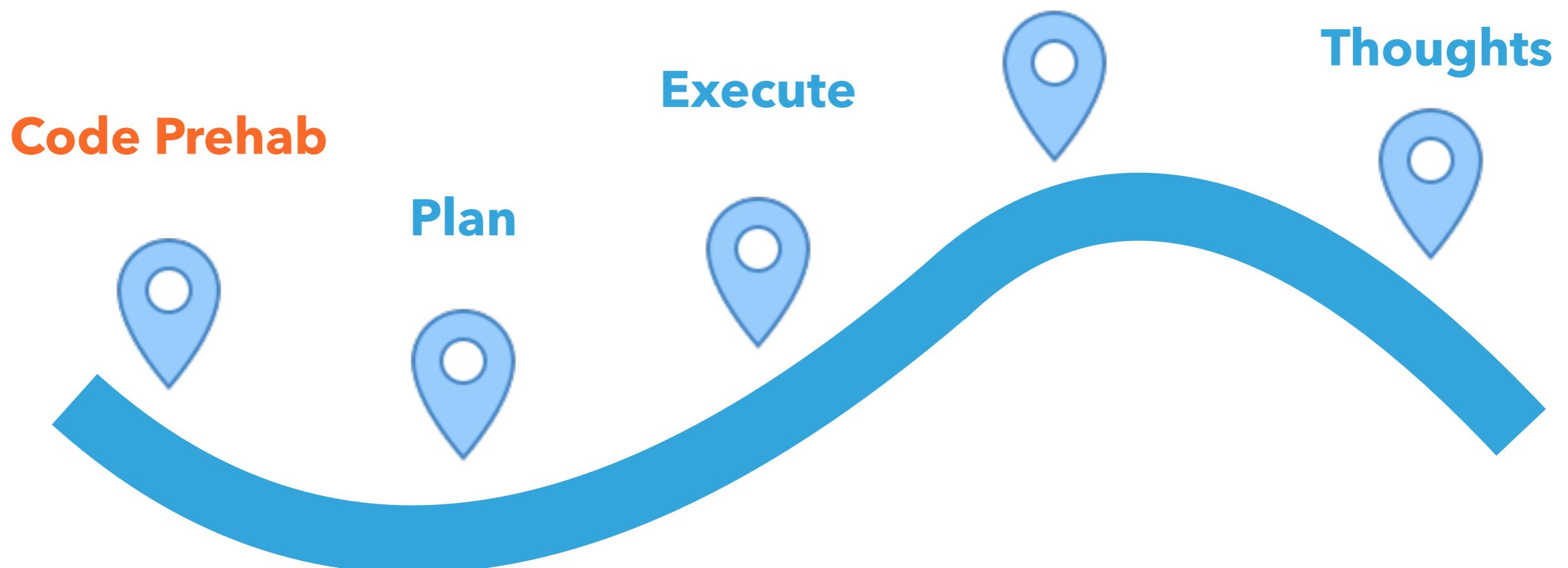
5:25 PM · Feb 24, 2023 · 3,813 Views

@daveliddament

# Writing a Custom Rector Rule



# Writing a Custom Rector Rule





# Add tombstones

```
final class PriceCalculator
{
    public function calculate(): void
    {
        TombstoneReporter::trigger(
            'PriceCalculator::calculate');

        // Rest of method's code
    }
}
```

# Trigger tombstones



`triggered_tombstones.json`

[

```
'PriceCalculator::calculate',
'PriceController::index',
'PriceRepository::getItems',
...
```

];

# Remove triggered tombstones

```
final class PriceCalculator
{
    public function calculate(): void
    {
        TombstoneReporter::trigger(
            'PriceCalculator::calculate');
        // Rest of method's code
    }
}
```

# Delete code



**"REMOVE: Product listing page"**

**"REMOVE: 2 for 1 discount code"**

# Increase type coverage

```
function getNames($users)
{
    $names = [];
    foreach($users as $user) {
        $names[] = $user->getUser();
    }
    return join(', ', $names);
}
```

0% type coverage

@daveliddament

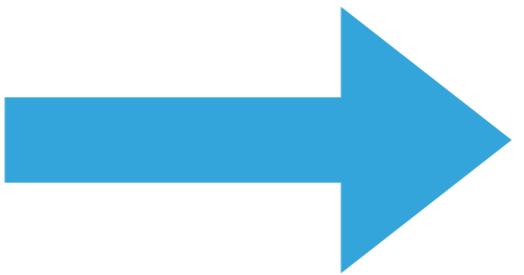
```
/** @param array<int,User> $users */
function getNames(array $users): string
{
    $names = [];
    foreach($users as $user) {
        $names[] = $user->getUser();
    }
    return join(', ', $names);
}
```

100% type coverage

@daveliddament

# User

getUser



getUsername

# 0% type coverage

```
function getNames($users)
{
    $names = [];
    foreach($users as $user) {
        $names[] = $user->getUser();
    }
    return join(', ', $names);
}
```

# 100% type coverage

```
/** @param array<int,User> $users */
function getNames(array $users): string
{
    $names = [];
    foreach($users as $user) {
        $names[] = $user->getUser();
    }
    return join(', ', $names);
}
```

# 100% type coverage

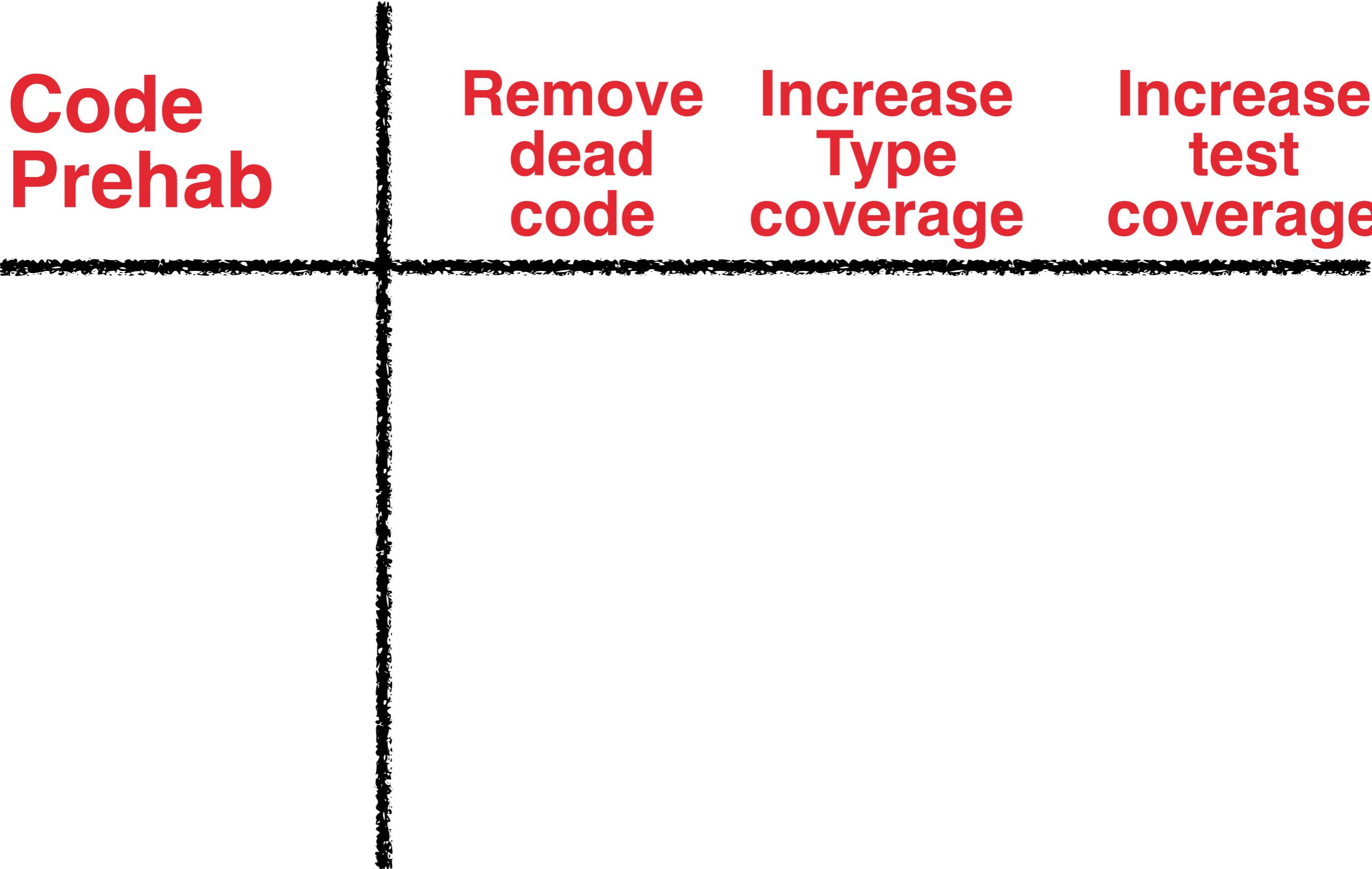
```
/** @param array<int,User> $users */
function getNames(array $users): string
{
    $names = [];
    foreach($users as $user) {
        $names[] = $user->getUsername();
    }
    return join(', ', $names);
}
```

100% automated!

# Tests



# Code Prehab

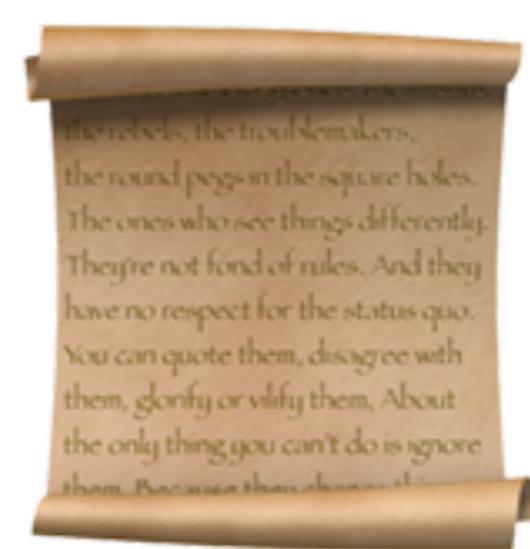


Remove dead code      Increase Type coverage      Increase test coverage

# Writing a Custom Rector Rule



# SHIFT



# SHIFT

<https://laravelshift.com/>

@daveliddament



<https://getrector.com/>



# 359 Rules (and counting)

## RenameMethodRector

Turns method names to new ones.

🔧 **configure it!**

- class: [Rector\Renaming\Rector\MethodCall\RenameMethodRector](#)

```
$someObject = new SomeExampleClass;  
-$someObject->oldMethod();  
+$someObject->newMethod();
```



## ChangeIfElseValueAssignToEarlyReturnRector

Change if/else value to early return

- class: [Rector\EarlyReturn\Rector\If\\_\ChangeIfElseValueAssignToEarlyReturnRector](#)

```
class SomeClass
{
    public function run()
    {
        if ($this->hasDocBlock($tokens, $index)) {
-            $docToken = $tokens[$this->getDocBlockIndex($tokens, $index)];
-        } else {
-            $docToken = null;
+            return $tokens[$this->getDocBlockIndex($tokens, $index)];
        }

-
-            return $docToken;
+            return null;
    }
}
```



```
composer require --dev rector/rector
```

```
vendor/bin/rector
```

```
No "rector.php" config found. Should we  
generate it for you? [yes]:
```

```
> yes
```

```
[OK] The config is added now. Re-run command  
to make Rector do the work!
```

```
<?php
```

```
use Rector\Config\RectorConfig;
use Rector\Renaming\Rector\MethodCall\RenameMethodRector;
use Rector\Renaming\ValueObject\MethodCallRename;

return RectorConfig::configure()

->withPaths([
    __DIR__ . '/src',
])

->withConfiguredRule(
    RenameMethodRector::class,
    [
        new MethodCallRename(
            App\Framework\Model::class,
            'belongsTo',
            'belongs')
    ]
);


```

@daveliddament

7/7 [██████████] 100%

2 files with changes

=====

1) src/Framework/Model.php:5

----- begin diff -----

@@ @@

```
abstract class Model
{
-    public function belongsTo(string $model): BelongsTo
+    public function belongs(string $model): BelongsTo
    {
        return new BelongsTo($model);
    }
}
```

----- end diff -----

Applied rules:

\* RenameMethodRector

2) src/Models/Car.php:10

```
----- begin diff -----
@@ @@
{
    public function user(): BelongsTo
    {
-        return $this->belongsTo('User');
+        return $this->belongs('User');
    }

    public function manufacturer(): BelongsTo
    {
-        return $this->belongsTo(Manufacturer::class);
+        return $this->belongs(Manufacturer::class);
    }
}

----- end diff -----
```

Applied rules:

- \* RenameMethodRector



# Rulesets: A set of rules

Rector: PHP Language upgrades

<https://github.com/rectorphp/rector-symfony>

<https://github.com/driftingly/rector-laravel>

```
<?php

declare(strict_types=1);

use Rector\Config\RectorConfig;
use RectorLaravel\Set\LaravelSetList;

return RectorConfig::configure()

->withSets([
    LaravelSetList::LARAVEL_100
]);
```

## Renamed "Swift" Methods

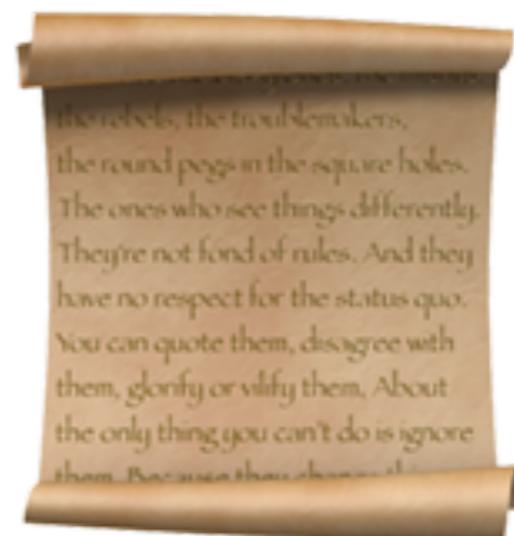
Various SwiftMailer related methods, some of which were undocumented, have been renamed to their Symfony Mailer counterparts. For example, the `withSwiftMessage` method has been renamed to `withSymfonyMessage`:

<https://laravel.com/docs/9.x/upgrade#symfony-mailer>

---

```
$rectorConfig->ruleWithConfiguration(  
    RenameMethodRector::class,  
    [  
        new MethodCallRename(  
            'Illuminate\Mail\Message',  
            'withSwiftMessage',  
            'withSymfonyMessage'),  
    ]);
```

<https://github.com/driftingly/rector-laravel/blob/main/config/sets/laravel90.php#L89>





Prologue

Release Notes

• Upgrade Guide

Contribution Guide

Q Search

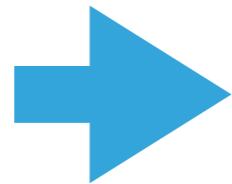
# Upgrade Guide

# Upgrading to 10.0 from 9.x

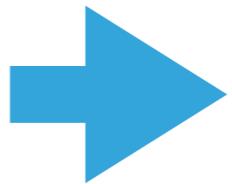


Symfony

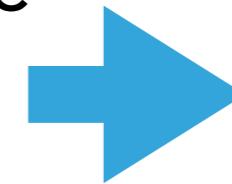
Upgrade  
to highest  
minor  
version



Fix all the  
deprecation  
warnings



Upgrade  
major  
version



Run flex  
recipes

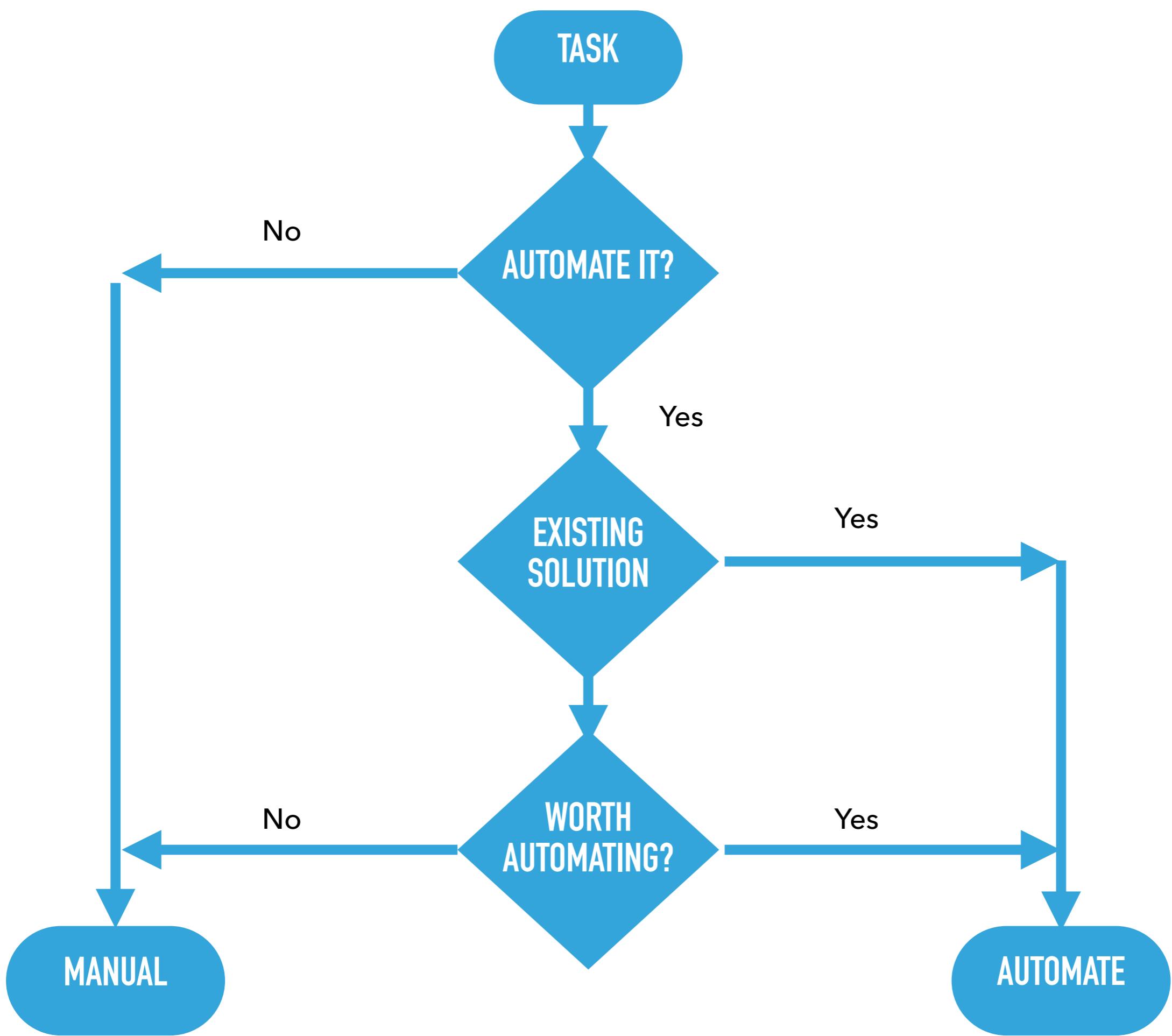
[https://symfony.com/doc/current/setup/upgrade\\_major.html](https://symfony.com/doc/current/setup/upgrade_major.html)

@daveliddament

4.2

9

- 
- Laravel validators
  - Symfony validators
  - E2E tests
  - Initial controller (probably login one)
  - Controllers and middleware
  - Remaining commands
  - Unit tests
  - Multiple environments (currently only test environment is setup)  
Includes ansible updates
  - Full testings
  - Merge in changes done since this branch started
  - PHP 8.1 upgrade



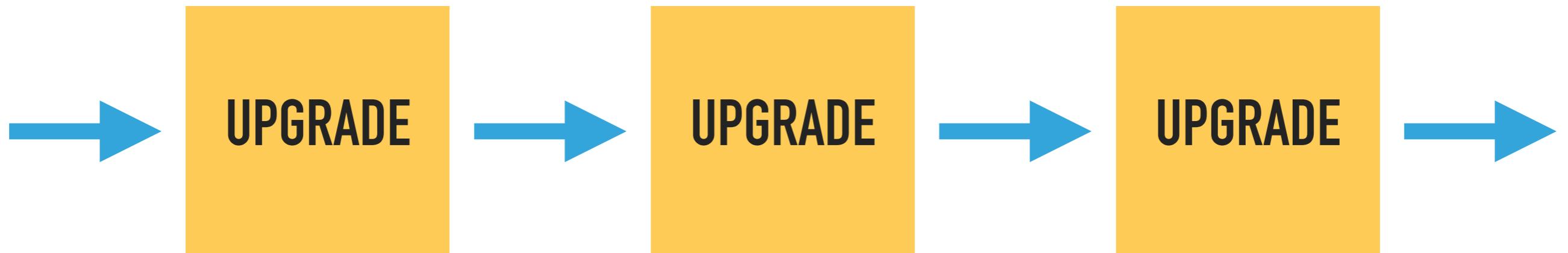
Development



Development



# Smaller deploys would be better\*



	<b>Framework</b>	<b>PHP</b>	<b>PHPUnit</b>
<b>Current deployed versions</b>	12	8.1	8
<b>Current supported versions</b>	12	8.0, 8.1, 8.2	8, 9
<b>New supported versions</b>	13	8.2, 8.3	9, 10

	Framework	PHP	PHPUnit
Current deployed versions	12	8.1	8
Current supported versions	12	8.0, 8.1, 8.2	8, 9
New supported versions	13	8.2, 8.3	9, 10



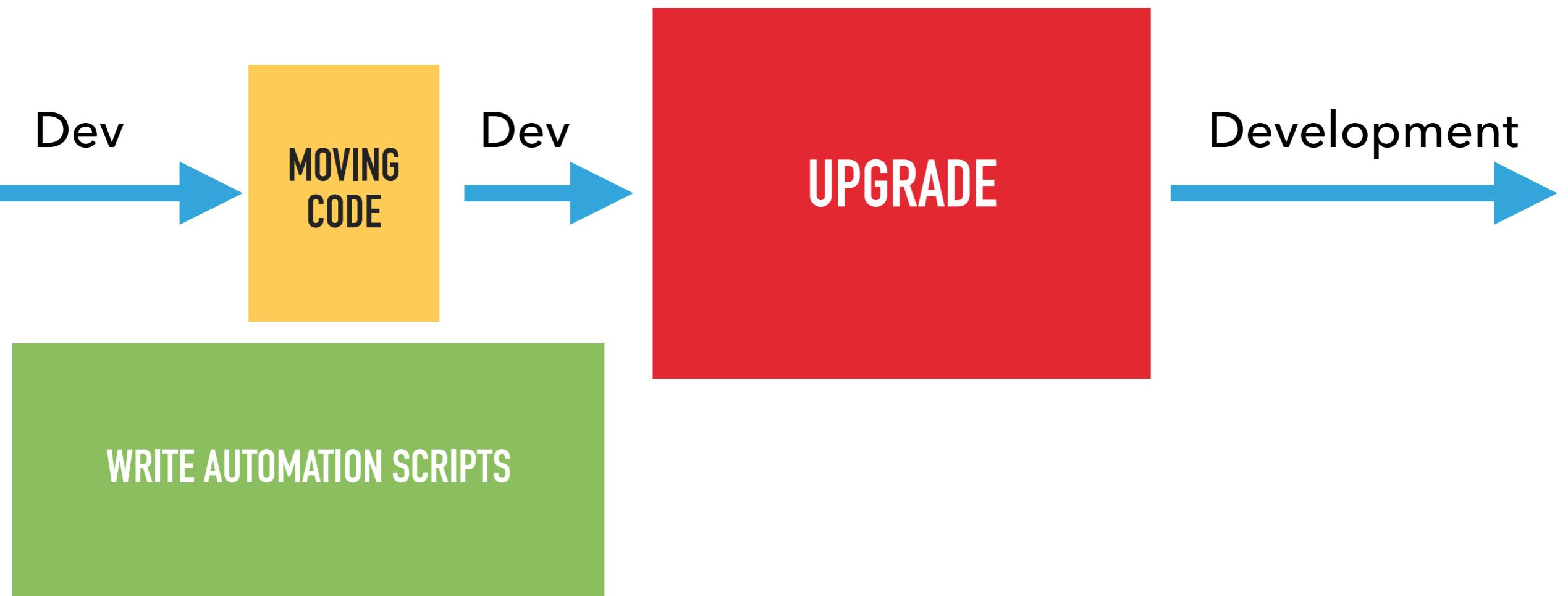
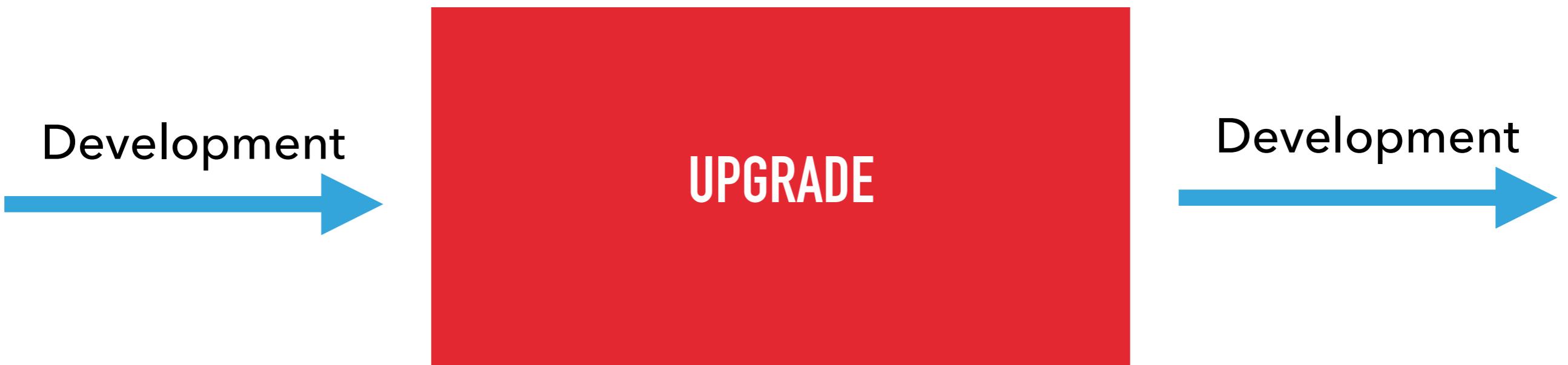
Development



UPGRADE

Development





# Code Prehab

# Plan

Remove dead code      Increase Type coverage      Increase test coverage



- Laravel validators
- Symfony validators
- E2E tests
- Initial controller (probably login one)
- Controllers and middleware
- Remaining commands
- Unit tests
- Multiple environments (currently only test environment is setup)  
Includes ansible updates
- Full testings
- Merge in changes done since this branch started
- PHP 8.1 upgrade



# Writing a Custom Rector Rule



# Tips

- ▶ Be systematic; do one type of change at once. (Both manual and automation).
  - ▶ Note other changes you need to make.
- ▶ Use git commits (even if you later squash them)
- ▶ Upgrades: fix the deprecations, you don't need to (immediately) use the new features.
- ▶ Do the minimum number of changes between possible release points

# Writing a Custom Rector Rule



# Old

```
class Car extends Model
{
    public function user()
    {
        return $this->belongsTo('User');
    }
}
```

# New

```
class Car extends Model
{
    public function user()
    {
        return $this->belongsTo(App\Models\User::class);
    }
}
```

# No Change

```
class Car extends Model
{
    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

# No Change

```
class CarService
{
    public function user()
    {
        return $this->belongsTo('User');
    }
}
```

```
$ vendor/bin/rector custom-rule
```

What is the name of the rule class (e.g. "LegacyCallToDbalMethodCall")?:

> FixModelMappingsRector

Generated files

=====

- \* utils/rector/src/Rector/FixModelMappingsRector.php
- \* utils/rector/tests/Rector/FixModelMappingsRector/Fixture/some\_class.php.inc
- \* utils/rector/tests/Rector/FixModelMappingsRector/config/configured\_rule.php
- \* utils/rector/tests/Rector/FixModelMappingsRector/FixModelMappingsRectorTest.php

[OK] Base for the "FixModelMappingsRector" rule was created.

Now you can fill the missing parts

[OK] We also update composer.json autoload-dev, to load Rector rules. Now run "composer dump-autoload" to update paths

# Old

```
<?php
namespace Utils\Rector\Tests\Rector\FixModelMappingsRector\Fixture;

use App\Framework\BelongsTo;
use App\Framework\Model;

final class Car extends Model
{
    public function user(): BelongsTo
    {
        return $this->belongsTo('User');
    }
}
----
```

# New

```
<?php
namespace Utils\Rector\Tests\Rector\FixModelMappingsRector\Fixture;

use App\Framework\BelongsTo;
use App\Framework\Model;

final class Car extends Model
{
    public function user(): BelongsTo
    {
        return $this->belongsTo(App\Models\User::class);
    }
}
```



No  
change



```
<?php
namespace Utils\Rector\Tests\Rector\FixModelMappingsRector\Fixture;

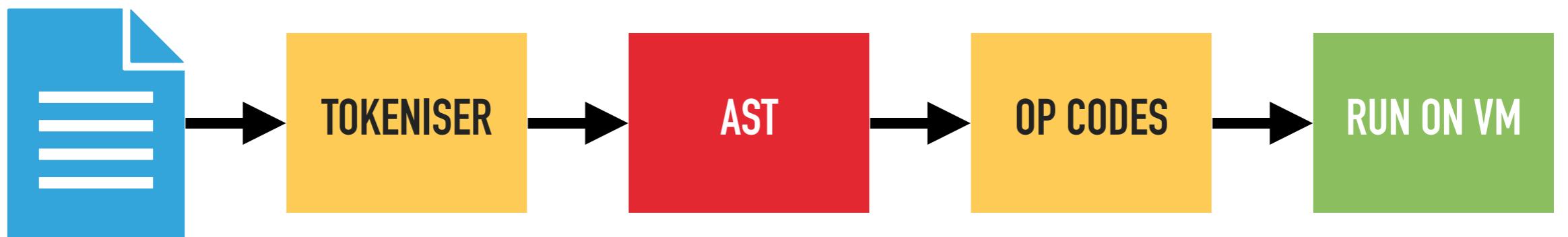
use App\Framework\BelongsTo;
use App\Framework\Model;

final class Car extends Model
{
    public function user(): BelongsTo
    {
        return $this->belongsTo(App\Models\User::class);
    }
}
```

```
FixModelMappingsRector extends AbstractRector
{
    public function getRuleDefinition(): RuleDefinition
    public function getNodeTypes(): array
    public function refactor(Node $node): ?Node
}
```

```
public function getRuleDefinition(): RuleDefinition
{
    return new RuleDefinition(
        "Fix Model mappings",
        [
            new CodeSample(
                '$this->belongsTo("User");',
                '$this->belongsTo(User::class);',
            ),
        ],
    );
}
```

```
FixModelMappingsRector extends AbstractRector
{
    public function getRuleDefinition(): RuleDefinition
    public function getNodeTypes(): array
    public function refactor(Node $node): ?Node
}
```



```
class Car
```

```
{
```

```
    public function users() {...}
```

```
    public function anotherMethod() {...}
```

```
}
```

**Name**

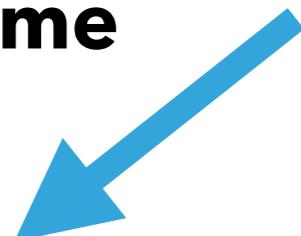
IDENTIFIER  
Name: Car

CLASS  
Flags: 0

**Statements**

CLASS METHOD  
Flags: 1

CLASS METHOD  
Flags: 1



```
public function user ()
```

```
{
```

```
    return $this->belongsTo('User');
```

```
}
```

**Name**

CLASS METHOD  
Flags: 1

**Statements**

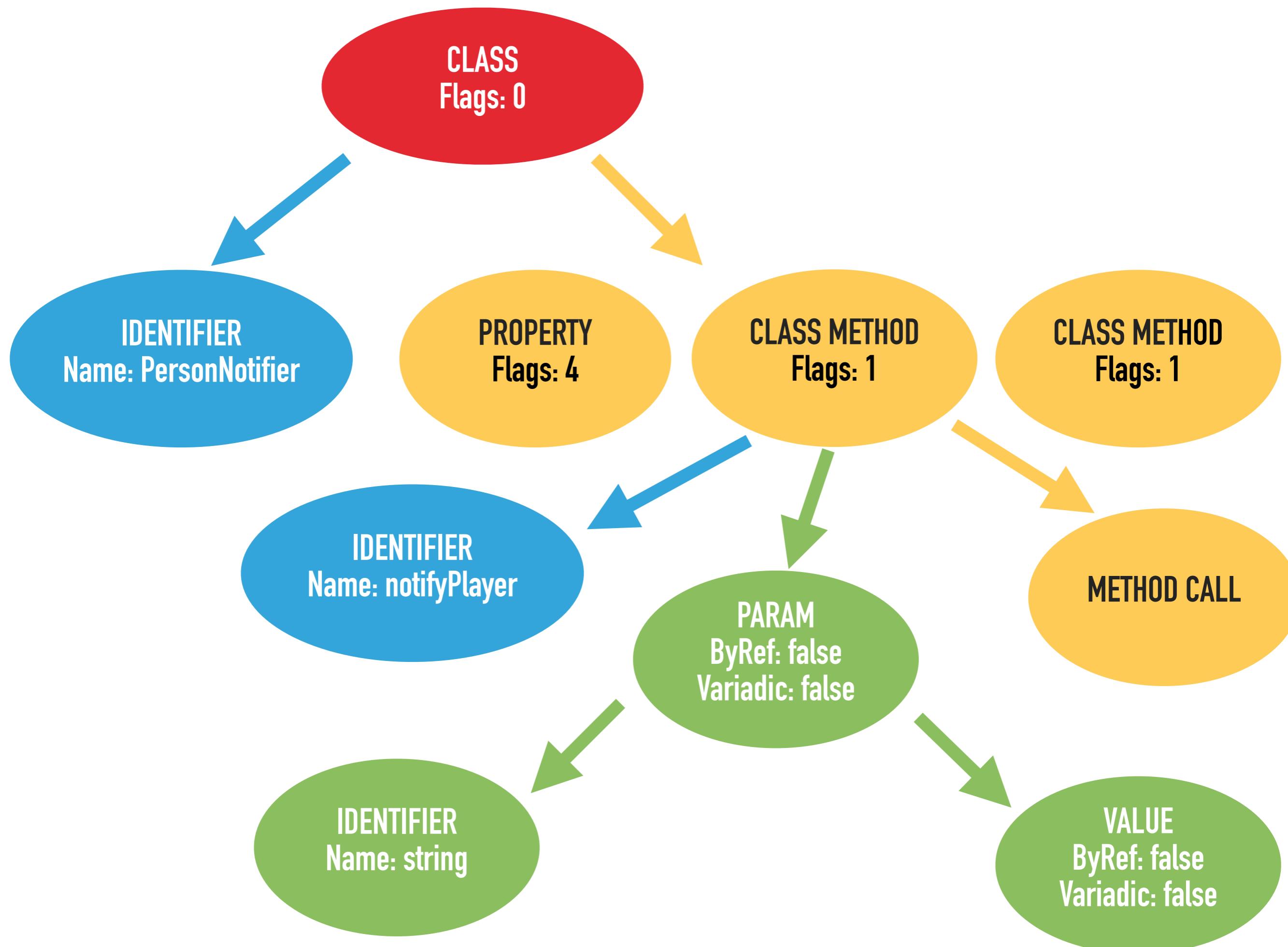
IDENTIFIER  
Name: notifyPlayer

METHOD CALL

# <https://github.com/nikic/PHP-Parser>

The screenshot shows the GitHub repository page for 'nikic/PHP-Parser'. At the top, it displays the repository name, a 'Public' badge, and statistics: 232 watchers and 891 forks. Below this is a navigation bar with links for Code, Issues (44), Pull requests (9), Actions, Wiki, Security, and Insights. The 'Code' tab is selected. In the center, there's a summary box showing 'master', '9 branches', '80 tags', and a recent commit by 'nikic' from 2 hours ago. To the right, there's an 'About' section describing it as 'A PHP parser written in PHP' with tags for 'php', 'parser', 'static-analysis', and 'ast'. At the bottom of the screenshot, there's a large red list of bullet points.

- ▶ PHP code can be represented by an AST
- ▶ Different types of Node
- ▶ Nodes contain information
- ▶ Each type of node has different information



```
$this->belongsTo('User');
```

```
$this->belongsTo(App\Models\User::class);
```



METHOD CALL

```
class MethodCall extends \PhpParser\Node\Expr\CallLike
{
    /** @var Expr Variable holding object */
    public $var;

    /** @var Identifier|Expr Method name */
    public $name;

    /** @var array<Arg|VariadicPlaceholder> Arguments */
    public $args;

    // Rest of class ...
}
```

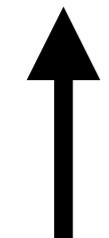
---

```
$this -> belongsTo ('User');
```

```
public function getNodes(): array  
{  
    return [MethodCall::class];  
}
```

```
public function refactor(Node $node): ?Node
{
}
```

```
$this->belongsTo( 'User' );
```



var



name



args

- ▶ var is an object that extends Model
- ▶ name is belongsTo
- ▶ 1st argument is a string

```
public function refactor(Node $node): ?Node
{
    // Is var an object that extends Model?

    $modelType = new ObjectType(Model::class);

    if (!$this->isObjectType($node->var, $modelType)) {
        return null;
    }
}
```

```
// Is name belongsTo  
  
if (!this->isName($node->name, 'belongsTo')) {  
    return null;  
}  
}
```

```
// Is 1st argument a string?  
  
$arg = $node->args[0] ?? null;  
if (! $arg instanceof Arg) {  
    return null;  
}  
  
if (! $arg->value instanceof String_) {  
    return null;  
}
```

```
// Return updated node

$fqcn = 'App\\Models\\' . $arg->value->value;

$arg->value = new ClassConstFetch(
    new Name($fqcn),
    new Identifier("class"));

return $node;
}
```

---

**User::class** is a ClassConstFetch Node in the AST

```
public function refactor(Node $node): ?Node
{
    // Is var an object that extends Model?
    $modelType = new ObjectType(Model::class);
    if (!$this->isObjectType($node->var, $modelType)) {
        return null;
    }

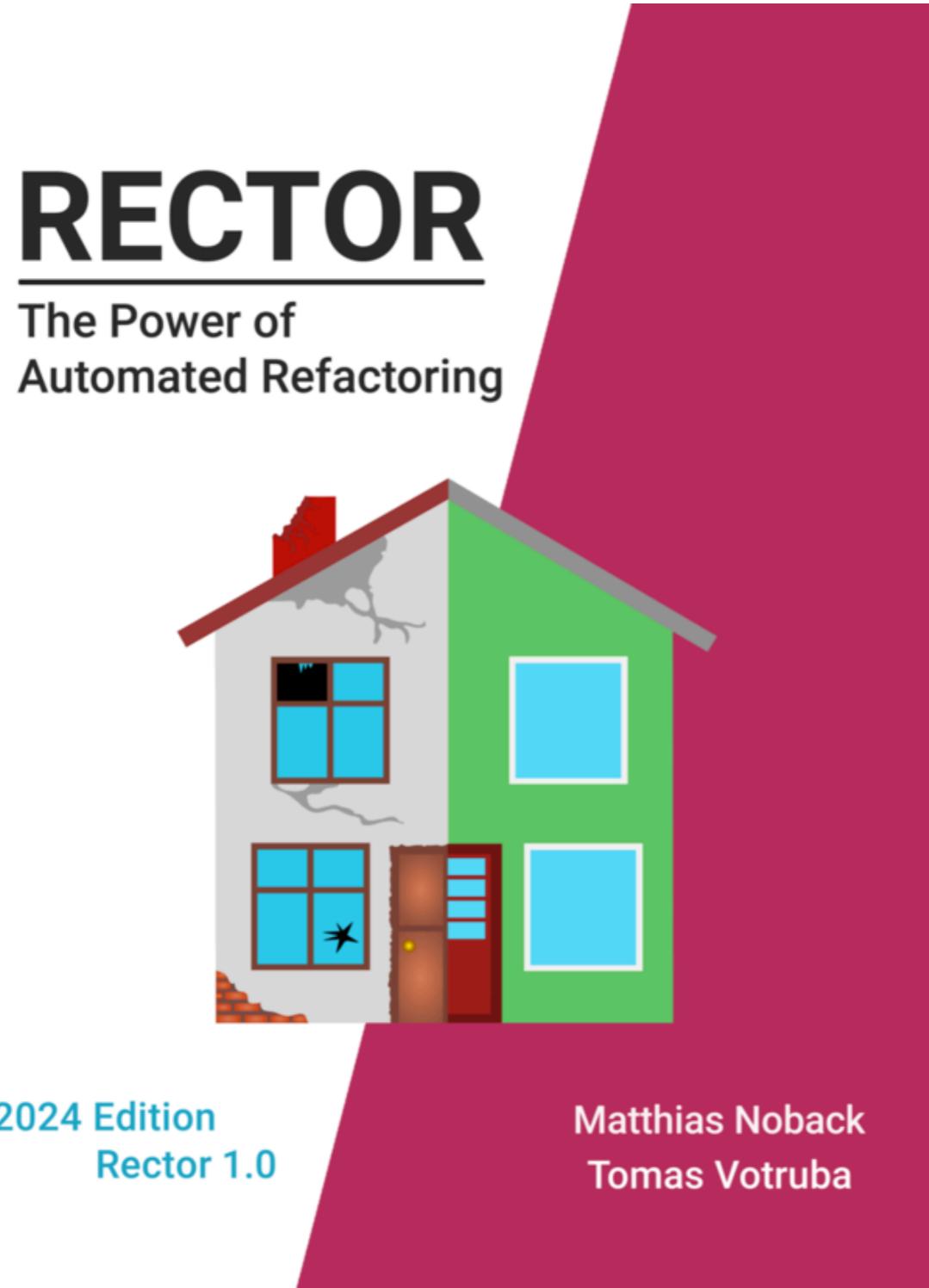
    // Is name belongsTo
    if (!$this->isName($node->name, 'belongsTo')) {
        return null;
    }

    // Is 1st argument a string?
    $arg = $node->args[0] ?? null;
    if (! $arg instanceof Arg) {
        return null;
    }
    if (! $arg->value instanceof String_) {
        return null;
    }

    // Return updated node
    $fqcn = 'App\\Models\\' . $arg->value->value;
    $arg->value = new ClassConstFetch(new Name($fqcn), new Identifier("class"));

    return $node;
}
```

<https://github.com/DaveLiddament/rector-rule-demo-update-belongs-to>



@daveliddament

# Code Prehab

Remove  
dead  
code

Increase  
Type  
coverage

Increase  
test  
coverage

# Plan

SHIFT



- Laravel validators
- Symfony validators
- E2E tests
- Initial controller (probably login one)
- Controllers and middleware
- Remaining commands
- Unit tests
- Multiple environments (currently only test environment is setup)  
Includes ansible updates
- Full testings
- Merge in changes done since this branch started
- PHP 8.1 upgrade

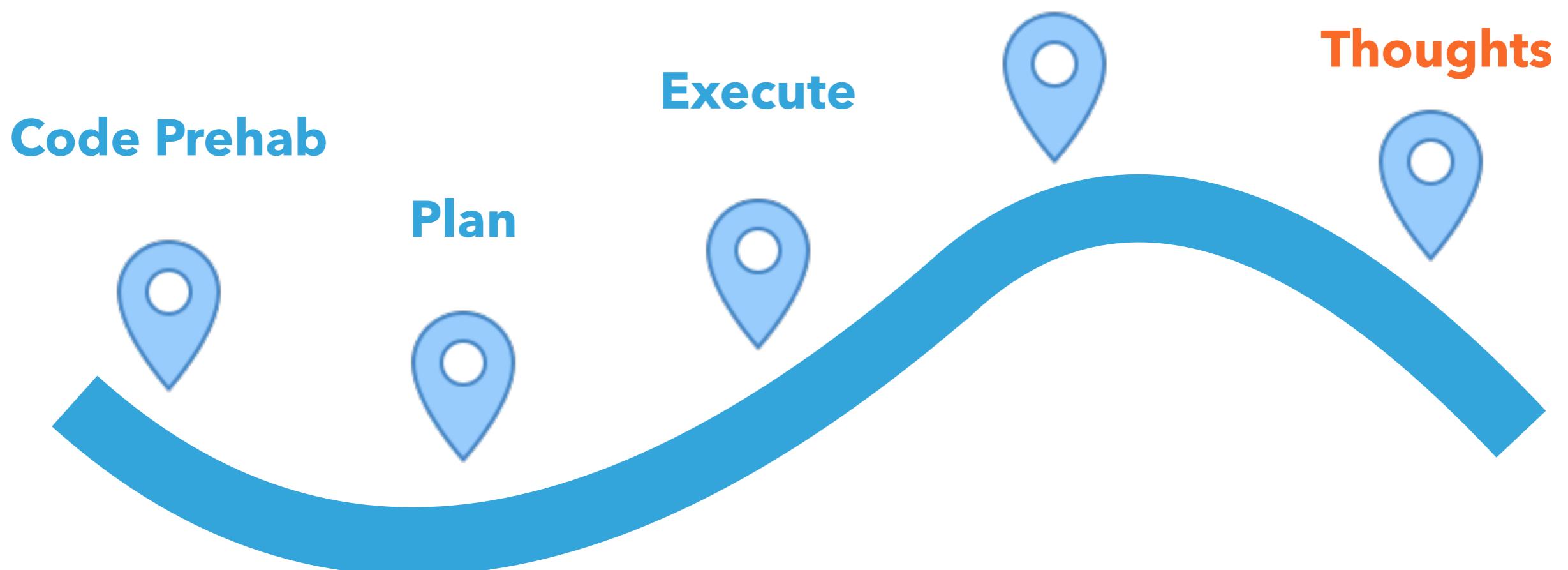


# Execute

Small  
focused  
steps

Minimum  
to  
upgrade

# Writing a Custom Rector Rule



**FRAMEWORK**



**BUSINESS LOGIC**

LIB

**PHP**



```
interface TextMessageGateway
{
    public function sendMessage(
        MobileNumber $from,
        MobileNumber $to,
        string $message,
    ) ;
}
```

**FRAMEWORK**



**BUSINESS LOGIC**

LIB

**PHP**



# What if....

**composer.json**

```
"version-upgrade" : {  
    "2" : {  
  
        "validate" : [  
            "phpstan:my-framework/v1-2-checks",  
            "script:upgrades/v1-2-checks.sh"  
        ],  
  
        "upgrade" : [  
            "script:upgrades/v1-2-update-1.sh"  
            "rector:my-framework/v1-2-rector-pre",  
            "script:upgrades/v1-2-update-2.sh"  
        ]  
    }  
}
```

# Dave Liddament

@daveliddament

