



PHP ODESSA CONF

Effective Code Review

Dave Liddament

WHY THIS TALK?

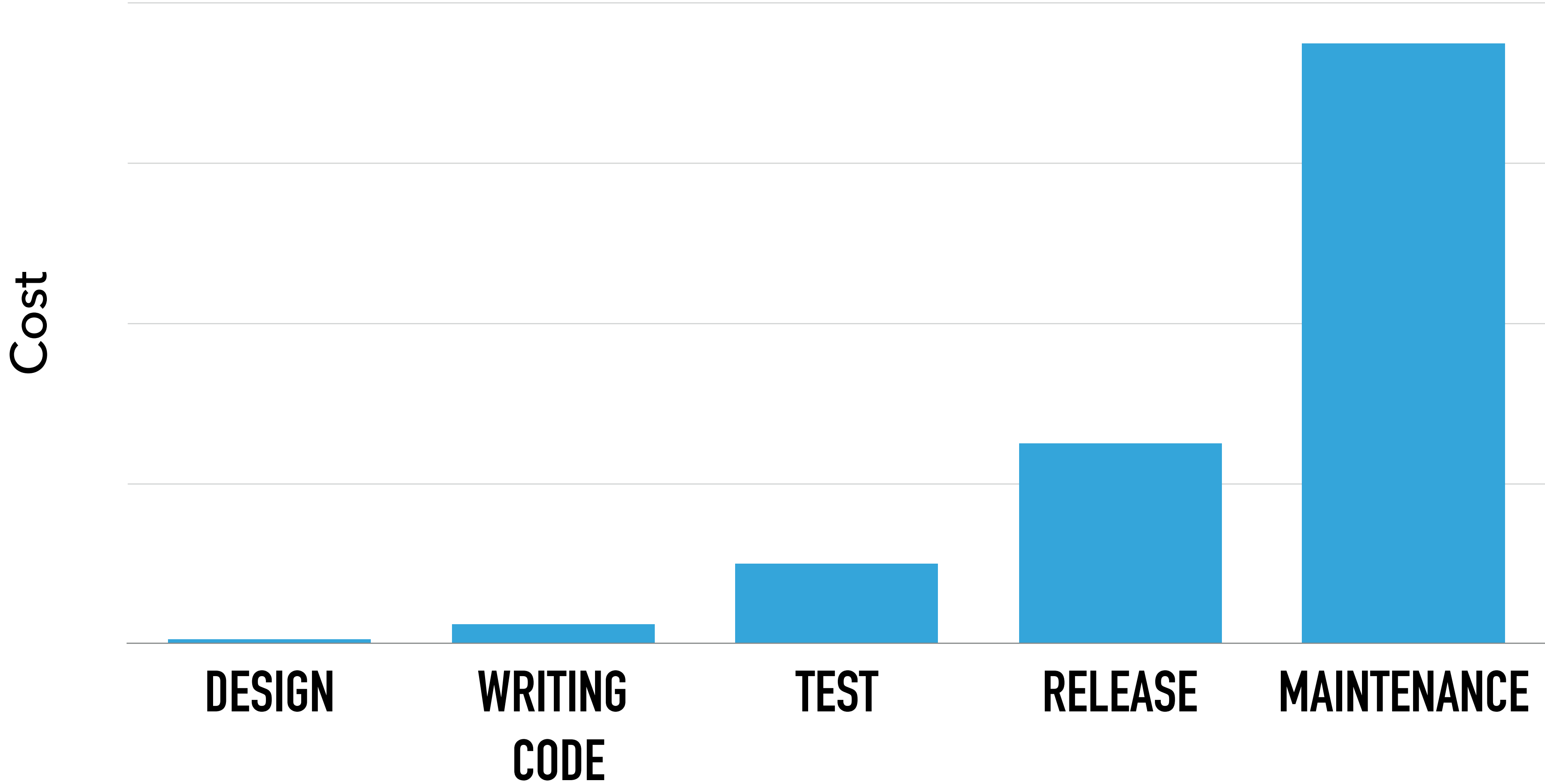
EFFECTIVE CODE REVIEW REDUCES OVERALL COST OF SOFTWARE DEVELOPMENT



@daveliddament

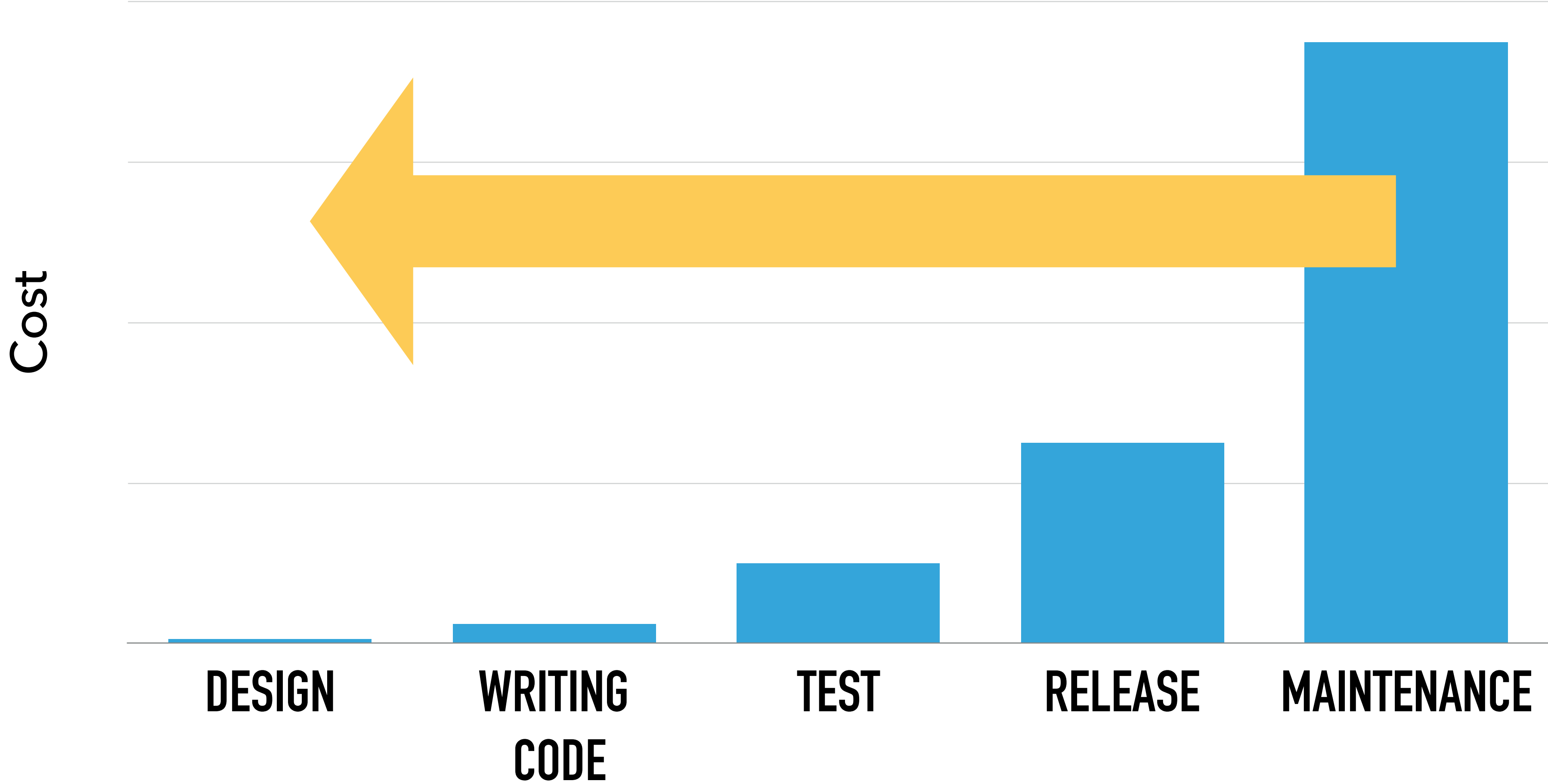
COST OF A BUG

FIND BUGS SOONER



COST OF A BUG

FIND BUGS SOONER



Dave Liddament

Lamp Bristol

Organize:

- PHP-SW user group
- Bristol PHP Training

Author of SARB

@daveliddament



AGENDA

- ▶ Why
- ▶ Code quality
- ▶ What is code review
- ▶ Benefits
- ▶ Implementation
- ▶ Tips
- ▶ Integrating code review into project workflow



EFFECTIVE CODE REVIEW REQUIRES...



EFFECTIVE CODE REVIEW REQUIRES...



EFFECTIVE CODE REVIEW REQUIRES...



**Check
Style**



Psalm



EFFECTIVE CODE REVIEW REQUIRES...



**Check
Style**



Psalm



Travis CI



**CODE REVIEW IS THE
SYSTEMATIC EXAMINATION OF
SOURCE CODE...**

Wikipedia



**IT IS INTENDED TO FIND MISTAKES
OVERLOOKED IN SOFTWARE
DEVELOPMENT, IMPROVING THE
OVERALL QUALITY OF SOFTWARE.**

Wikipedia



WHAT IS CODE REVIEW

HOW IS IT DONE



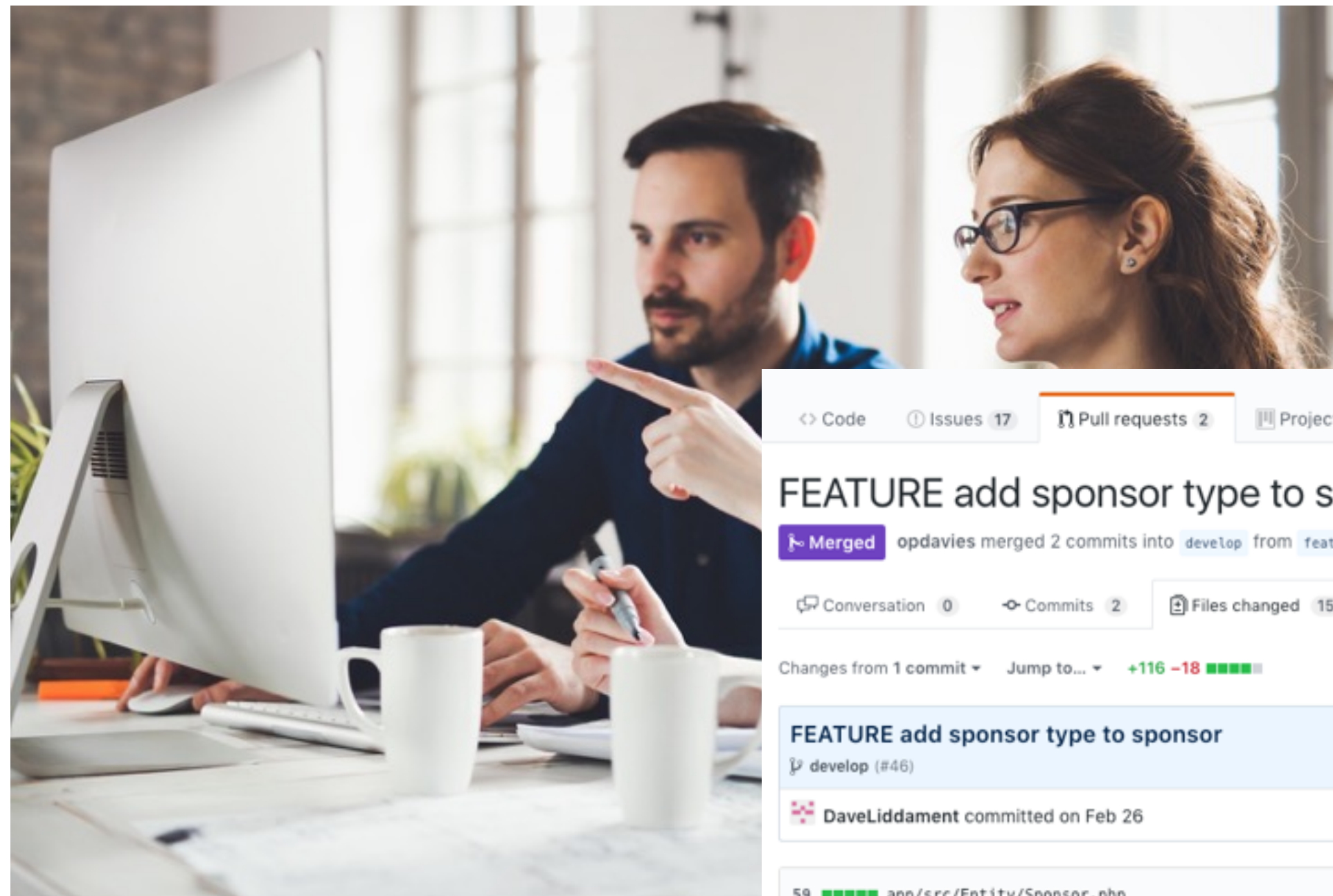
WHAT IS CODE REVIEW

HOW IS IT DONE



WHAT IS CODE REVIEW

HOW IS IT DONE



<> Code

① Issues 17

Pull requests 2

Projects 0

Wiki

Insights

Settings

FEATURE add sponsor type to sponsor #46

Merged

opdavies merged 2 commits into develop from feature/update-sponsors on Feb 26

Conversation 0

Commits 2

Files changed 15

Changes from 1 commit ▾ Jump to... ▾ +116 -18

Unified Split Review changes ▾

FEATURE add sponsor type to sponsor

develop (#46)

DaveLiddament committed on Feb 26

commit 86070f4c08780c8a167bef2b44e09a00609915d6

59 app/src/Entity/Sponsor.php

@@ -6,6 +6,16 @@

6

7 class Sponsor

8 {

6

7 class Sponsor

8 {

9 + /**

10 + * Full sponsor.

11 + */

12 + const SPONSOR_FULL = 'full';

13 +

14 + /**

15 + * Sponsor only covers occasional events.

16 + */

17 + const SPONSOR_EVENT = 'event';

18 +

19 +

20 +

21 +



@daveliddament

AGENDA

- ▶ Why
- ▶ Code quality
- ▶ What is code review
- ▶ **Benefits**
- ▶ Implementation
- ▶ Tips
- ▶ Integrating code review into project workflow



CODE REVIEW BENEFITS



WHAT ARE DEFECTS?



WHAT ARE DEFECTS?

Bugs



WHAT ARE DEFECTS?

Bugs

Evolvability

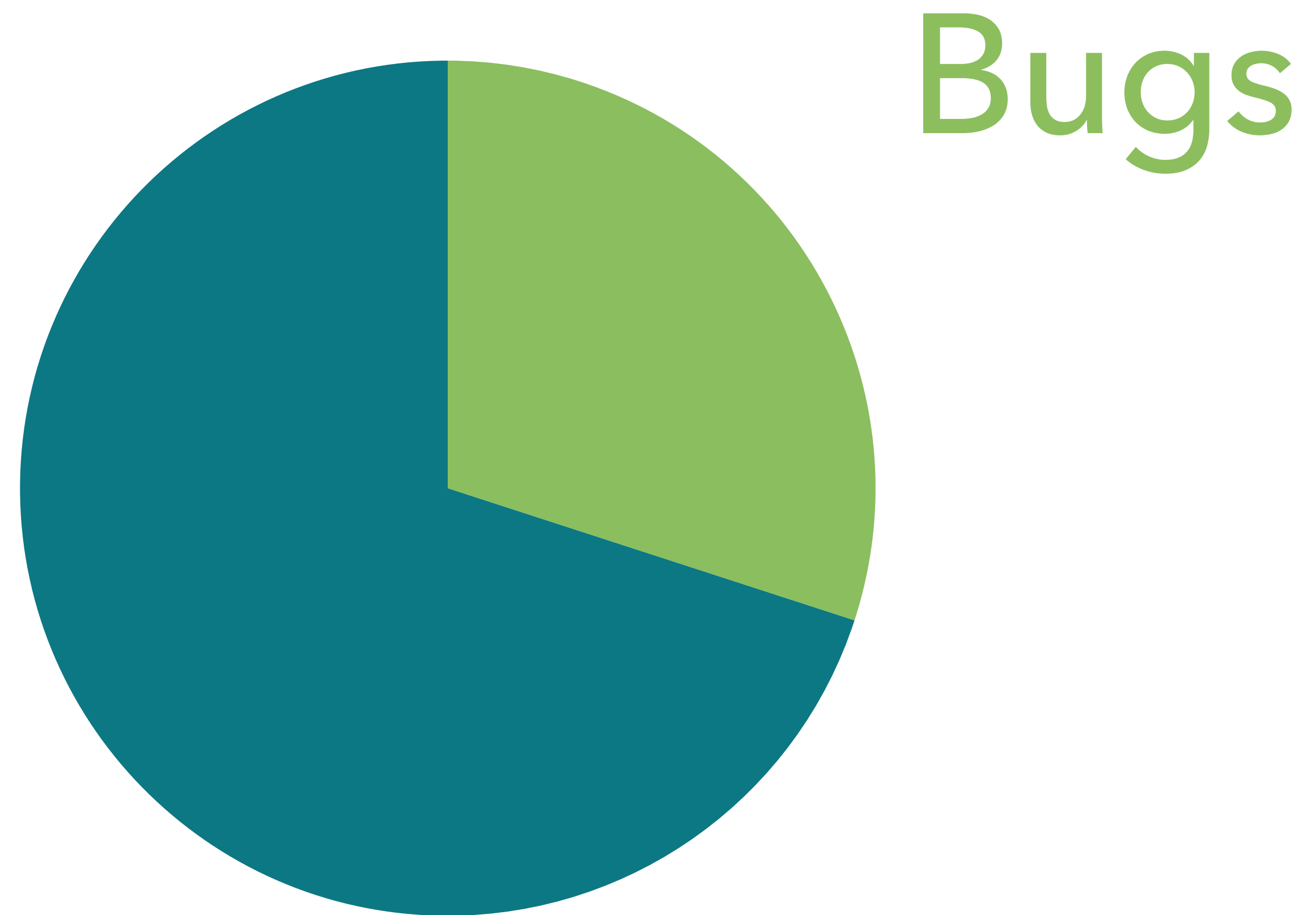


AN EVOLVABILITY DEFECT IS...

Code that makes code base less compliant with standards, more error prone, or more difficult to modify, extend or understand.



WHAT ARE DEFECTS?



Evolvability

[1, 2]



EVOLVABILITY IS IMPORTANT

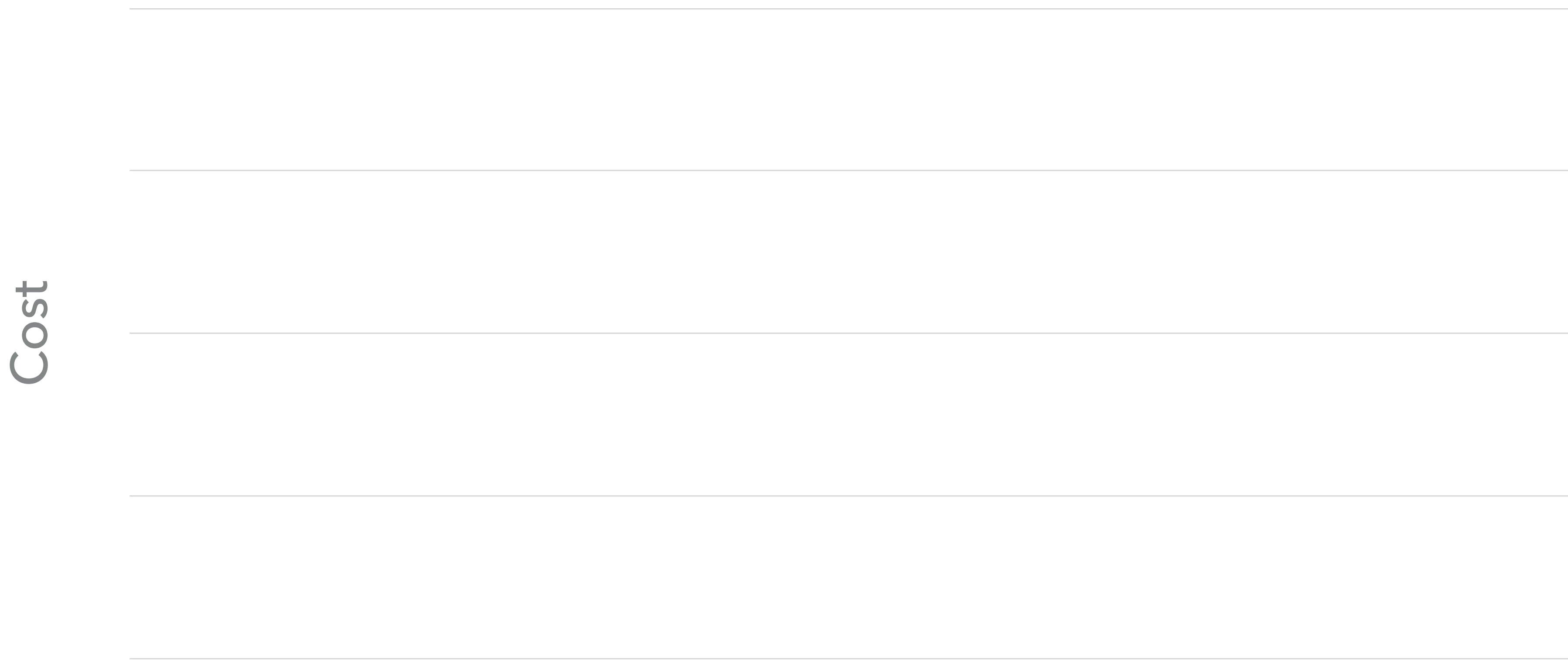
- ▶ Low evolvability costs money:
 - ▶ New features took 28% longer to implement [3]
 - ▶ Fixing bugs took 36% longer [3]
- ▶ Software structure may account for 25% of total maintenance costs [4]



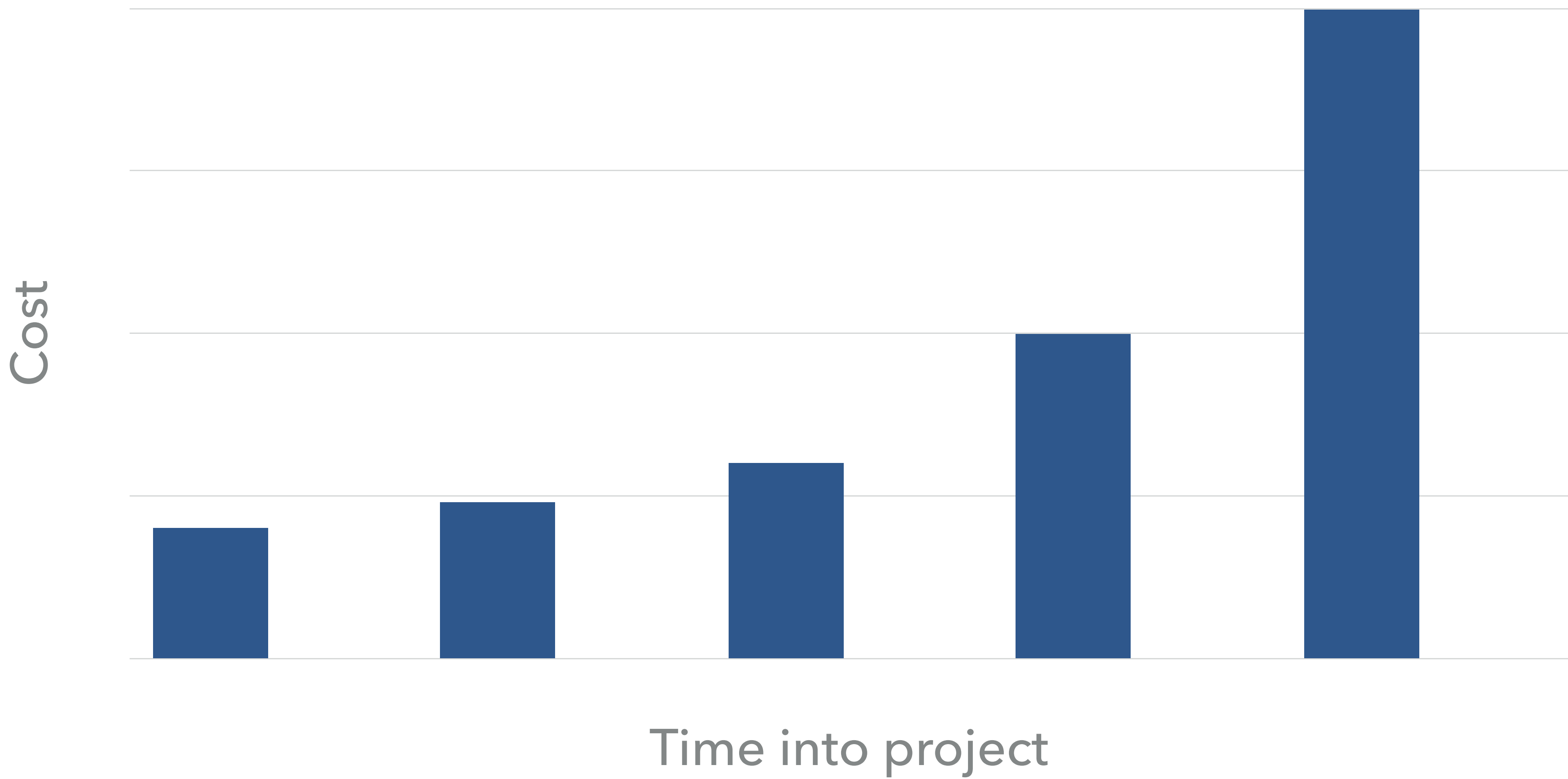
COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



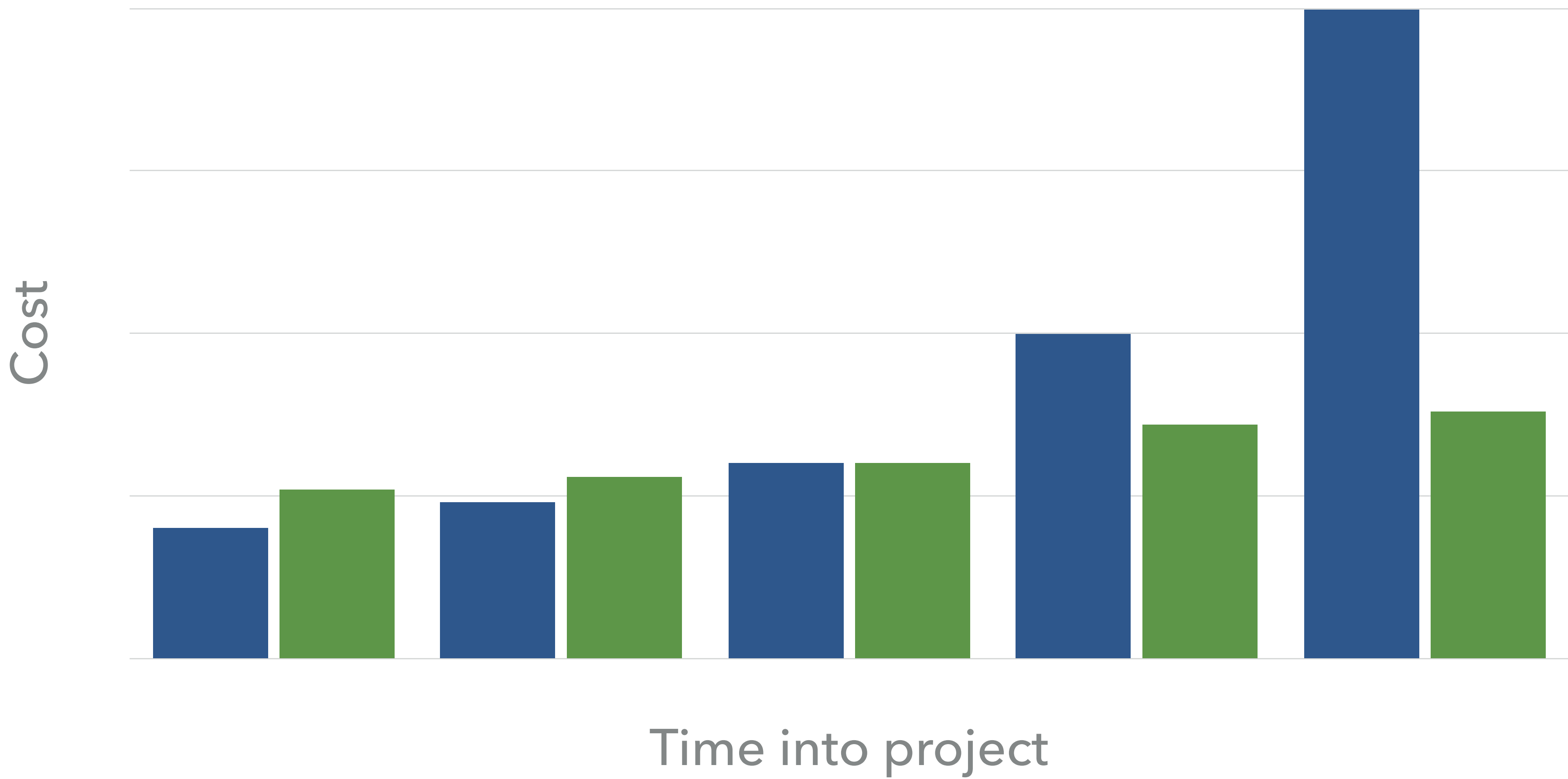
COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



TAKE AWAY

- ▶ The majority of code review comments will not be “bugs”.
 - ▶ OK not to find lots of “bugs”
- ▶ Most comments will be code improvements.
 - ▶ Lower technical debt. Lower overall cost.
 - ▶ Studies back this up.
- ▶ Remember to sell the right metric to management.



CODE REVIEW BENEFITS

SECURITY REVIEW



SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)



SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)
 - ▶ Bugsnag / error log reporting



SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)
 - ▶ Bugsnag / error log reporting
- ▶ Files that shouldn't be there? (e.g. malware)



SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)
 - ▶ Bugsnag / error log reporting
- ▶ Files that shouldn't be there? (e.g. malware)
- ▶ OWASP top 10



SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)
 - ▶ Bugsnag / error log reporting
- ▶ Files that shouldn't be there? (e.g. malware)
- ▶ OWASP top 10
 - ▶ OWASP top 10 cheat sheet



SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)
 - ▶ Bugsnag / error log reporting
- ▶ Files that shouldn't be there? (e.g. malware)
- ▶ OWASP top 10
 - ▶ OWASP top 10 cheat sheet
- ▶ Not / weakly hashing passwords

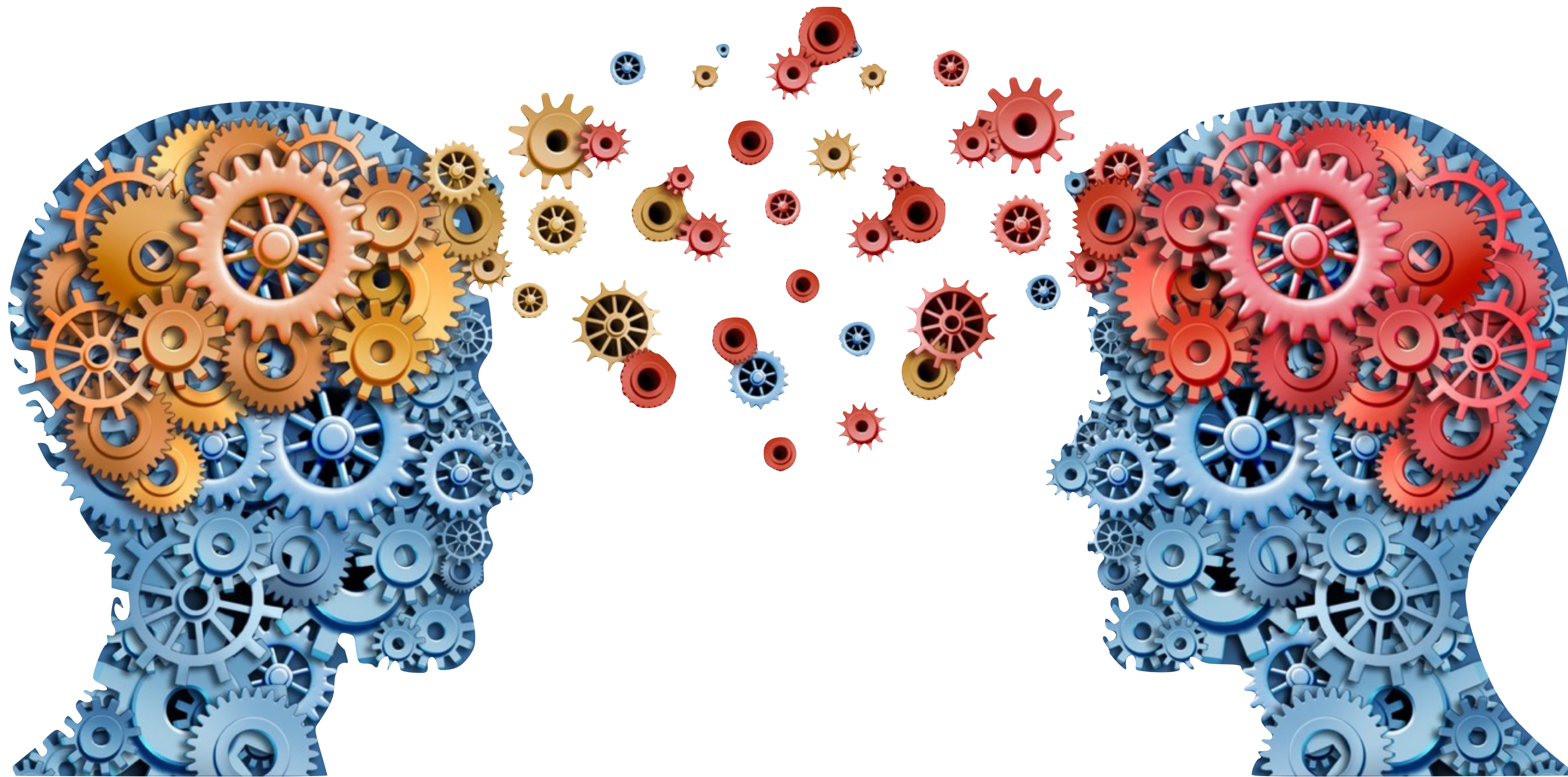


SECURITY REVIEW

- ▶ Writing sensitive data to logs (e.g. password)
 - ▶ Bugsnag / error log reporting
- ▶ Files that shouldn't be there? (e.g. malware)
- ▶ OWASP top 10
 - ▶ OWASP top 10 cheat sheet
- ▶ Not / weakly hashing passwords
- ▶ Rolled your own authentication / hashing / encryption algorithms



SPREAD THE KNOWLEDGE



NO MORE SILOS



NO MORE SILOS



NO MORE SILOS



NO MORE SILOS



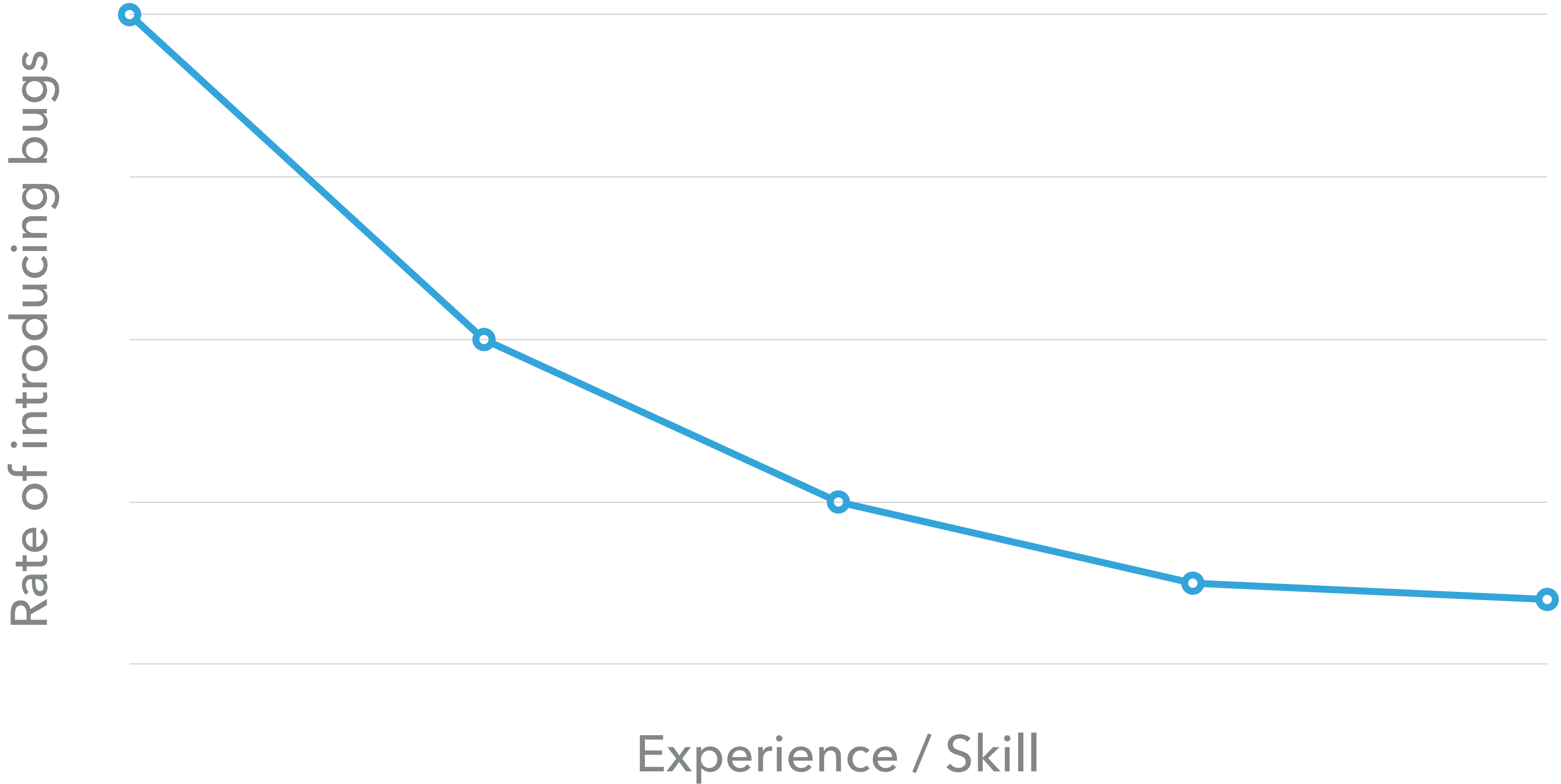
NO MORE SILOS



MENTORING



SKILL AND EXPERIENCE VS BUGS



YOU'RE BEING WATCHED!



YOU'RE BEING WATCHED!



[5]

BENEFITS

- ▶ Reduce defects
 - ▶ Studies that back this up
- ▶ Find security vulnerabilities
- ▶ Spread knowledge
- ▶ Mentoring
- ▶ Peer pressure improves code quality



EFFECTIVE CODE REVIEW REDUCES OVERALL COST OF SOFTWARE DEVELOPMENT



AGENDA

- ▶ Why
- ▶ Code quality
- ▶ What is code review
- ▶ Benefits
- ▶ **Implementation**
- ▶ Tips
- ▶ Integrating code review into project workflow



IMPLEMENTATION





IMPLEMENTATION

WE CAN AUTOMATE



WE CAN AUTOMATE

► Tests



WE CAN AUTOMATE

- ▶ Tests
- ▶ Code style (php-cs-fixer)



WE CAN AUTOMATE

- ▶ Tests
- ▶ Code style (php-cs-fixer)
- ▶ Lint (php, yaml, twig, doctrine, composer)



WE CAN AUTOMATE

- ▶ Tests
- ▶ Code style (php-cs-fixer)
- ▶ Lint (php, yaml, twig, doctrine, composer)
- ▶ Static analysis (Psalm, PHPStan, Phan)



WE CAN AUTOMATE

- ▶ Tests
- ▶ Code style (php-cs-fixer)
- ▶ Lint (php, yaml, twig, doctrine, composer)
- ▶ Static analysis (Psalm, PHPStan, Phan)
- ▶ 3rd party security checks (sensiolabs/security-checker)



DO THE TESTS TEST THE REQUIRED FUNCTIONALITY



DO THE TESTS TEST THE REQUIRED FUNCTIONALITY

Input	Expected Output
my blog	my-blog
hello Dave	hello-Dave



DO THE TESTS TEST THE REQUIRED FUNCTIONALITY

Input	Expected Output
my blog	my-blog
hello Dave	hello-Dave



IMPLEMENTATION

ARE THE TESTS ADEQUATE



ARE THE TESTS ADEQUATE

Input	Expected Output
my blog	my-blog
it's Saturday	its-saturday



HOW MANY TESTS DO WE NEED?

```
/**
 * @param int $id
 * @return bool
 * @throws NotFoundException
 */
public function isAllowed(int $id): bool
{
    .. some implementation ..
}
```



**WILL I UNDERSTAND THIS
CODE IN 6 MONTHS TIME?**



“THE RATIO OF TIME SPENT **READING** VERSUS **WRITING** IS WELL OVER **10 TO 1**. WE ARE CONSTANTLY READING OLD CODE AS PART OF THE EFFORT TO WRITE NEW CODE. . . .
[THEREFORE,] **MAKING IT EASY TO READ MAKES IT EASIER TO WRITE.**”

Robert C. Martin (Clean Code)



WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

WHAT DOES THIS CODE DO?

```
$userFields = [  
    'Username',  
    'Email',  
    'FirstName',  
    'LastName',  
    'Phone',  
];  
  
foreach ($userFields as $key) {  
    if ($userDetails->{'get' . $key}()) {  
        $user->{'set' . $key}($userDetails->{'get' . $key}());  
    }  
}
```



WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

WHAT DOES THIS CODE DO? (2)

```
if ($userDetails->getUsername()) {  
    $user->setUsername($userDetails->getUsername());  
}  
if ($userDetails->getEmail()) {  
    $user->setEmail($userDetails->getEmail());  
}  
if ($userDetails->getFirstName()) {  
    $user->setFirstName($userDetails->getFirstName());  
}  
if ($userDetails->getLastName()) {  
    $user->setLastName($userDetails->getLastName());  
}  
if ($userDetails->getPhone()) {  
    $user->setPhone($userDetails->getPhone());  
}
```



WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

```
/**
 * Represents a location in the UK.
 * (eg city, town, village)
 */
class Location
{

    /**
     * @return string URL
     */
    public function getUrl(): string
    {
        return $this->url;
    }
}
```



WHAT IS URL?

```
/**
 * Represents a location in the UK.
 * (eg city, town, village)
 */
class Location
{

    /**
     * @return string URL
     */
    public function getUrl(): string
    {
        return $this->url;
    }
}
```



WHAT IS URL?

```
/**
 * Represents a location in the UK.
 * (eg city, town, village)
 */
class Location
{
```

```
    /**
     * @return string URL
     */
    public function getUrl(): string
    {
        return $this->url;
    }
```



WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

```
if ($agent->getType() === 1) {  
    ... do something ...  
}
```



WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

WHAT DOES 1 MEAN?

```
if ($agent->getType() === 1) {  
    ... do something ...  
}
```



IS THE CODE AS OBVIOUS AND EXPLICIT AS IT POSSIBLY CAN BE

HOW DO WE MAKE THIS MORE OBVIOUS

```
class MarketingCampaign
{
    public function addAddress($address)
    {
        .. some implementation ..
    }
}
```



IS THE CODE AS OBVIOUS AND EXPLICIT AS IT POSSIBLY CAN BE

HOW DO WE MAKE THIS MORE OBVIOUS (2)

```
class MarketingCampaign
{
    public function addEmailAddress($emailAddress)
    {
        .. some implementation ..
    }
}
```



IS THE CODE AS OBVIOUS AND EXPLICIT AS IT POSSIBLY CAN BE

HOW DO WE MAKE THIS MORE OBVIOUS (3)

```
class MarketingCampaign
{
    /**
     * Adds email address, person will then be
     * messaged as part of the campaign.
     */
    public function addEmailAddress($emailAddress)
    {
        .. some implementation ..
    }
}
```



IS THE CODE AS OBVIOUS AND EXPLICIT AS IT POSSIBLY CAN BE

HOW DO WE MAKE THIS MORE OBVIOUS (4)

```
class MarketingCampaign
{

    /**
     * Adds email address, person will then be
     * messaged as part of the campaign.
     */
    public function addEmailAddress(
        string $emailAddress): void
    {
        .. some implementation ..
    }
}
```



IS THE CODE AS OBVIOUS AND EXPLICIT AS IT POSSIBLY CAN BE

HOW DO WE MAKE THIS MORE OBVIOUS (4)

```
class MarketingCampaign
{
    /**
     * Adds email address, person will then be
     * messaged as part of the campaign.
     */
    public function addEmailAddress(
        EmailAddress $emailAddress): void
    {
        .. some implementation ..
    }
}
```



**ARE WE FOLLOWING
PROJECT CONVENTIONS?**



ARE WE FOLLOWING PROJECT CONVENTIONS

```
interface LocationRepository
{
    public function findClosestTo($point) ;

    public function findByName($name) ;

    public function findBySlug($slug) ;

    public function searchForLocation($name, $type) ;

    public function findAllByType($type) ;
}
```



INCONSISTENT METHOD NAME

```
interface LocationRepository
{
    public function findClosestTo($point) ;

    public function findByName($name) ;

    public function findBySlug($slug) ;

    public function searchForLocation($name, $type) ;

    public function findAllByType($type) ;
}
```



#coding-standards

☆ | 👤 4 | 📌 0 | [Add a topic](#)



dave 10:55 AM

Naming: Do not use abbreviations



8 replies

Last reply 3 months ago



dave 11:29 AM

Always type hint for parameters and return types for methods and functions (edited)



2 replies

Last reply 3 months ago



dave 2:31 PM

Type hint all member data in classes



2 replies

Last reply 3 months ago



dave 2:55 PM

Use void return type if method does not return anything



1 reply

3 months ago



dave 2:56 PM

Non void methods must always return a value explicitly (even if it's null)



**DO THE COMMENTS MATCH
THE CODE?**



HAS DEFENSIVE CODING
BEEN USED?



HAS DEFENSIVE CODING BEEN USED

```
/**
 * Set status (one of started|finished|quit)
 *
 * @param string $status
 */
public function setStatus(string $status): void
{
    $this->status = $status;
}
```



MISSING CHECK THAT STATUS IS A VALID VALUE

```
/**
 * Set status (one of started|finished|quit)
 *
 * @param string $status
 */
public function setStatus(string $status): void
{
    $this->status = $status;
}
```



CORRECT NAMING



CORRECT NAMING

- ▶ Language for domain or project you're working on



CORRECT NAMING

- ▶ Language for domain or project you're working on
- ▶ Design patterns



HAS TECHNICAL DEBT BEEN
DOCUMENTED?



HAS TECHNICAL DEBT BEEN DOCUMENTED

ALL TODO COMMENTS MUST REFERENCE A TICKET

```
// TODO: Refactor to method https://trello.com/c/Aaa123
```

```
... some hacky code ...
```



CAN ARCHITECTURE BE IMPROVED?



ARE THERE ANY BUGS?!



CHECK LIST

- ▶ Do the tests fully test the required functionality
- ▶ Will I understand this code in 6 months
- ▶ Does the code follow project conventions
- ▶ Is the code obvious and explicit
- ▶ Do comments match the code
- ▶ Has defensive coding been used
- ▶ Has technical debt been documented
- ▶ Can architecture be improved
- ▶ Are there any bugs



CHECK LIST: TLDR

- ▶ Correct tests
- ▶ Clean code
- ▶ No bugs



AGENDA

- ▶ Why
- ▶ Code quality
- ▶ What is code review
- ▶ Benefits
- ▶ Implementation
- ▶ Tips
- ▶ Integrating code review into project workflow



CODE REVIEW TIPS



EVERYONE SHOULD CODE REVIEW



ASK PROGRAMMERS TO
REVIEW **10 LINES OF CODE**
THEY'LL FIND **10 ISSUES...**

Anyone who's done code review

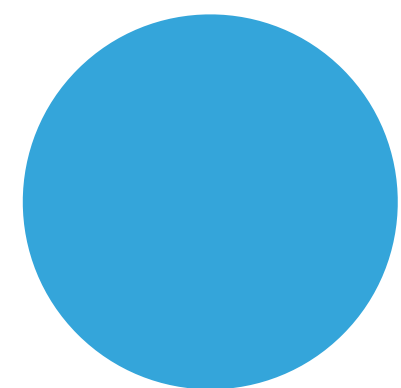


ASK THEM TO DO 500 LINES
THEY'LL SAY IT'S GOOD TO GO

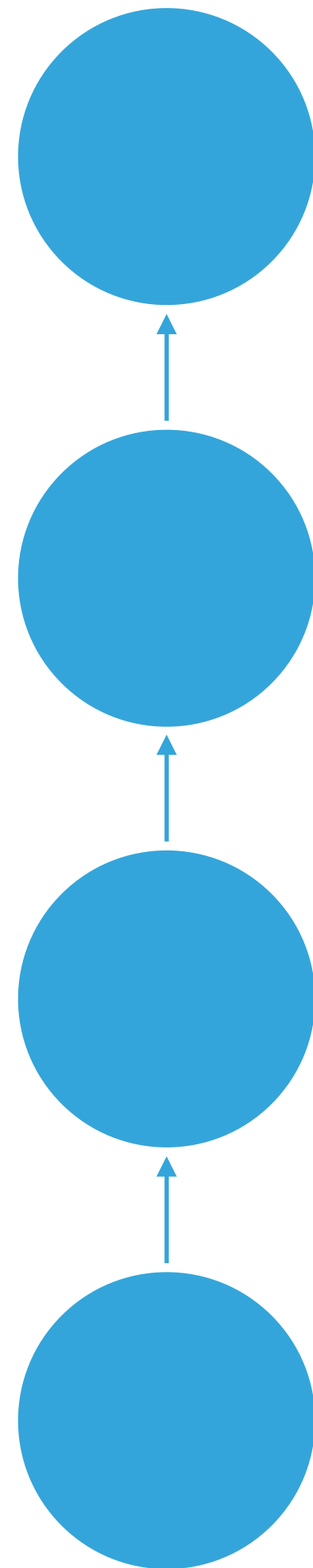
Anyone who's done code review



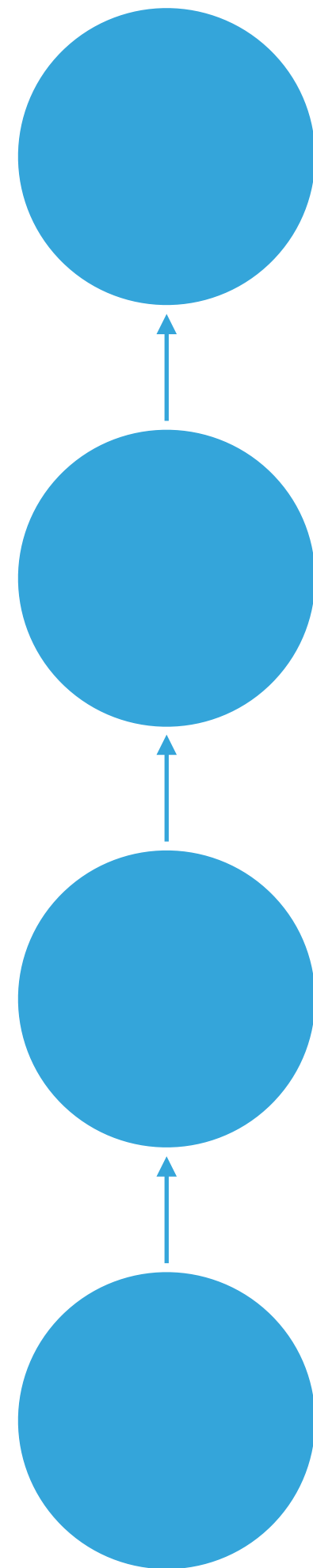
SMALL COMMITS



SMALL COMMITS



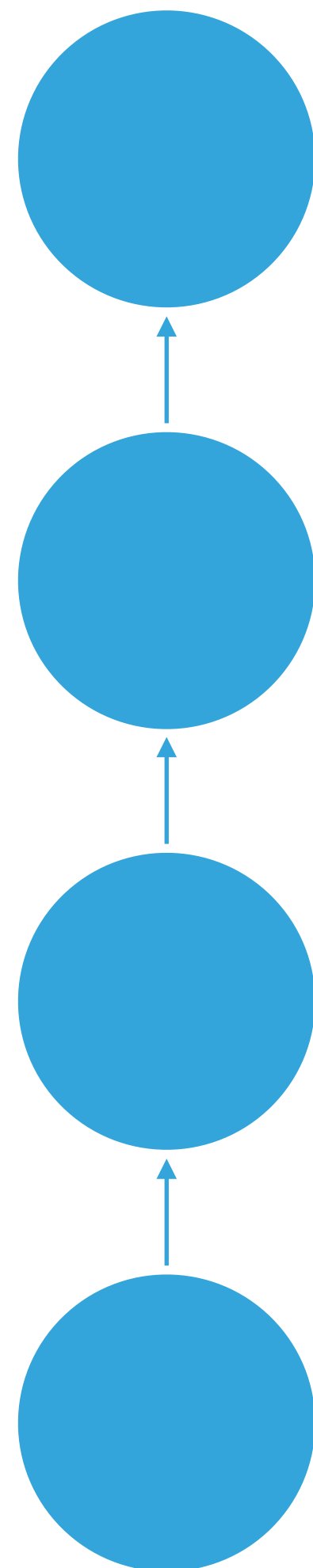
SMALL COMMITS



DEPRECATE: The price calculation service



SMALL COMMITS

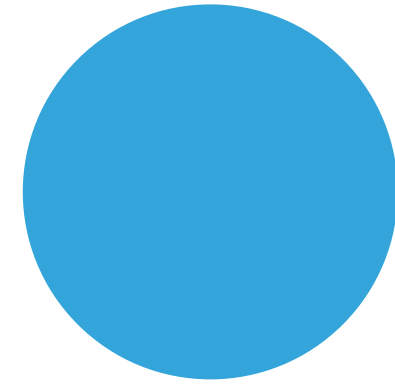
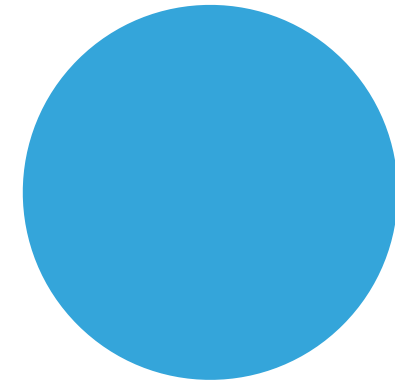
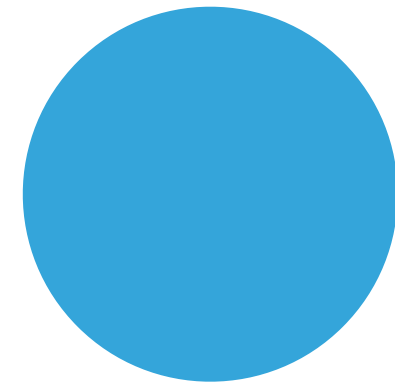
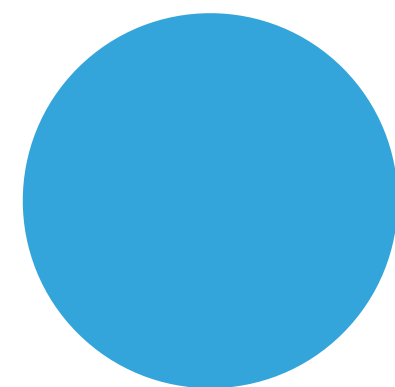


ADD: Facade to 3rd party price calculation service

DEPRECATE: The price calculation service



SMALL COMMITS



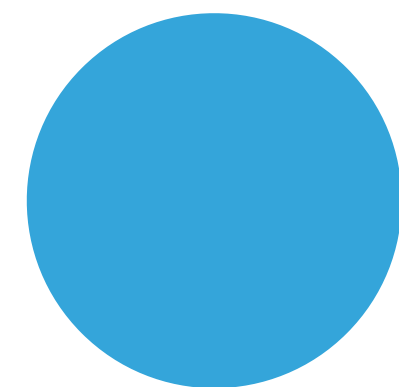
UPDATE: Use new price calculator code

ADD: Facade to 3rd party price calculation service

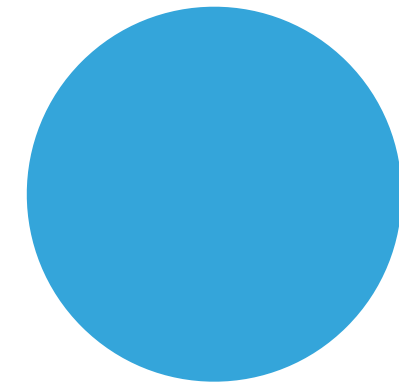
DEPRECATE: The price calculation service



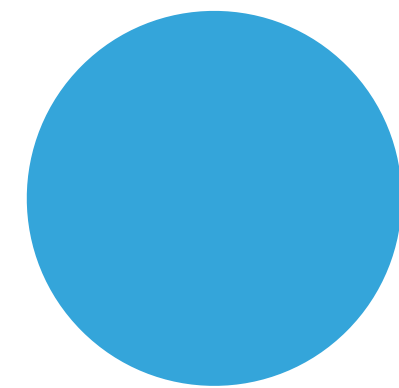
SMALL COMMITS



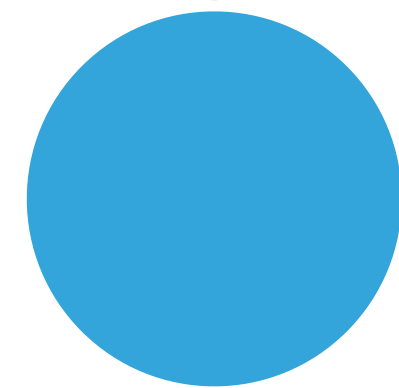
REMOVE: Deprecated price calculation service



UPDATE: Use new price calculator code



ADD: Facade to 3rd party price calculation service



DEPRECATE: The price calculation service



HOW MUCH SHOULD YOU REVIEW IN ONE GO?

- ▶ Fewer than 400 lines of code at a time [6]
- ▶ Under 500 line of code reviewed per hour [6]
- ▶ Max 1 hour review at a time [6]



GOOD COMMITS



GOOD COMMITS

- ▶ Small



GOOD COMMITS

- ▶ Small
- ▶ Focus on one thing



GOOD COMMITS

- ▶ Small
- ▶ Focus on one thing
- ▶ Whitespace changes into their own commits



GOOD COMMITS

- ▶ Small
- ▶ Focus on one thing
- ▶ Whitespace changes into their own commits
- ▶ Reference to story, bug report, Trello/Jira card



GOOD REVIEW COMMENTS



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Not critical of the author



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Not critical of the author
- ▶ State problem and solution



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Not critical of the author
- ▶ State problem and solution
- ▶ Link to Stack Overflow, blog, etc



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Not critical of the author
- ▶ State problem and solution
- ▶ Link to Stack Overflow, blog, etc
- ▶ Use: "Let's chat"



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Not critical of the author
- ▶ State problem and solution
- ▶ Link to Stack Overflow, blog, etc
- ▶ Use: "Let's chat"
- ▶ Use: "Question"



GOOD REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Not critical of the author
- ▶ State problem and solution
- ▶ Link to Stack Overflow, blog, etc
- ▶ Use: "Let's chat"
- ▶ Use: "Question"
- ▶ Compliment



RECEIVING REVIEW COMMENTS



RECEIVING REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude



RECEIVING REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Don't take offence



RECEIVING REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Don't take offence
- ▶ Do say if you disagree



RECEIVING REVIEW COMMENTS

- ▶ Don't be an idiot. Don't be rude
- ▶ Don't take offence
- ▶ Do say if you disagree
- ▶ Compliment



CODE REVIEW TIPS

- ▶ Get everyone involved
- ▶ Keep commits small
- ▶ Be constructive in code review comments
- ▶ Link to relevant bugs / stories / Trello cards



AGENDA

- ▶ Why
- ▶ Code quality
- ▶ What is code review
- ▶ Benefits
- ▶ Implementation
- ▶ Tips
- ▶ Integrating code review into project workflow



CODE CAN ONLY BE DEPLOYED IF:

- ▶ CI passes
- ▶ Code review passes



SETUP GITHUB

DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

master

Update

Protected branches

Protect branches to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to protected branches? [Learn more.](#)

Choose a branch...

No protected branches yet.



SETUP GITHUB

DaveLiddament / Demo

watch 1

★ Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

⚙ Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

master

Update

Protected branches

Protect branches to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to protected branches? [Learn more.](#)

Choose a branch...

No protected branches yet.



SETUP GITHUB

DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Repositories

Branches

Webhooks

Integrations & services

Deploy keys

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

master

Update

Protected branches

Protect branches to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to protected branches? [Learn more.](#)

Choose a branch...

No protected branches yet.



SETUP GITHUB

DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

master

Update

Protected branches


Prevent accidental force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to protected branches? [Learn more.](#)

Choose a branch...

No protected branches yet.



SETUP GITHUB

 DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Branch protection for master

- ☒ **Protect this branch**
Disables force-pushes to this branch and prevents it from being deleted.
- ☒ **Require pull request reviews before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at least one approved review and no changes requested before it can be merged into **master**.
- ☒ **Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a branch will dismiss pull request review approvals.
- ☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.
- ☒ **Require status checks to pass before merging**
Choose which [status checks](#) must pass before branches can be merged into **master**. When enabled, commits must first be pushed to another branch, then merged or pushed directly to **master** after status checks have passed.
- ☒ **Require branches to be up to date before merging**
This ensures the branch has been tested with the latest code on **master**.

Sorry, we couldn't find any status checks in the last week for this repository.
[Learn more about status checks on GitHub.](#)



SETUP GITHUB

DaveLiddament / Demo

Unwatch 1Star 0Fork 0

CodeIssues 0Pull requests 0Projects 0WikiSettingsInsights

OptionsCollaboratorsBranchesWebhooksIntegrations & servicesDeploy keys

Branch protection for master

☒ **Protect this branch**
Disables force-pushes to this branch and prevents it from being deleted.

☒ **Require pull request reviews before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at least one approved review and no changes requested before it can be merged into master.

☒ **Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a branch will dismiss pull request review approvals.

☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☒ **Require status checks to pass before merging**
Choose which [status checks](#) must pass before branches can be merged into master. When enabled, commits must first be pushed to another branch, then merged or pushed directly to master after status checks have passed.

☒ **Require branches to be up to date before merging**
This ensures the branch has been tested with the latest code on master.

Sorry, we couldn't find any status checks in the last week for this repository.

[Learn more about status checks on GitHub.](#)

SETUP GITHUB

DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Branch protection for master

- ☒ **Protect this branch**
Disables force-pushes to this branch and prevents it from being deleted.
- ☒ **Require pull request reviews before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at least one approved review and no changes requested before it can be merged into master.
- ☒ **Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a branch will dismiss pull request review approvals.
- ☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.
- ☒ **Require status checks to pass before merging**
Choose which [status checks](#) must pass before branches can be merged into master. When enabled, commits must first be pushed to another branch, then merged or pushed directly to master after status checks have passed.
- ☒ **Require branches to be up to date before merging**
This ensures the branch has been tested with the latest code on master.

Sorry, we couldn't find any status checks in the last week for this repository.
[Learn more about status checks on GitHub.](#)

SETUP GITHUB

DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Branch protection for master

☒ Protect this branch

Disables force-pushes to this branch and prevents it from being deleted.

☒ Require pull request reviews before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at

☒ Dismiss stale pull request approvals when new commits are pushed

New reviewable commits pushed to a branch will dismiss pull request review approvals.

☐ Require review from Code Owners

Require an approved review in pull requests including files with a designated code owner.

☒ Require status checks to pass before merging

Choose which [status checks](#) must pass before branches can be merged into master. When enabled, commits must first be pushed to another branch, then merged or pushed directly to master after status checks have passed.

☒ Require branches to be up to date before merging

This ensures the branch has been tested with the latest code on master.

Sorry, we couldn't find any status checks in the last week for this repository.

[Learn more about status checks on GitHub.](#)



SETUP GITHUB

DaveLiddament / Demo

Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Settings](#) [Insights](#)

Options
Collaborators
Branches
Webhooks
Integrations & services
Deploy keys

Branch protection for master

- ☒ **Protect this branch**
Disables force-pushes to this branch and prevents it from being deleted.
- ☒ **Require pull request reviews before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at least one approved review and no changes requested before it can be merged into **master**.
- ☒ **Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a branch will dismiss pull request review approvals.
- ☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.
- ☒ **Require status checks to pass before merging**
Choose which [status checks](#) must pass before branches can be merged into **master**. When enabled, commits must first be pushed to another branch, then merged or pushed directly to **master** after status checks have passed.
- ☒ **Require branches to be up to date before merging**
This ensures the branch has been tested with the latest code on **master**.

Sorry, we couldn't find any status checks in the last week for this repository.
[Learn more about status checks on GitHub.](#)

SETUP GITHUB

DaveLiddament / Demo

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Branch protection for master

☒ Protect this branch

Disables force-pushes to this branch and prevents it from being deleted.

☒ Require pull request reviews before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at least one approved review and no changes requested before it can be merged into master.

☒ Dismiss stale pull request approvals when new commits are pushed

New reviewable commits pushed to a branch will dismiss pull request review approvals.

☐ Require review from Code Owners

Require an approved review in pull requests including files with a designated code owner.

☒ Require status checks to pass before merging

Choose which [status checks](#) must pass before branches can be merged into master. When enabled, commits must first be pushed to another branch, then merged or pushed directly to master after status checks have passed.

☒ Require branches to be up to date before merging

This ensures the branch has been tested with the latest code on master.

Sorry, we couldn't find any status checks in the last week for this repository.

[Learn more about status checks on GitHub.](#)



daveliddament

SETUP GITHUB

DaveLiddament / Demo

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Options
Collaborators
Branches
Webhooks
Integrations & services
Deploy keys

Branch protection for master

- ☒ **Protect this branch**
Disables force-pushes to this branch and prevents it from being deleted.
- ☒ **Require pull request reviews before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with at least one approved review and no changes requested before it can be merged into **master**.
- ☒ **Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a branch will dismiss pull request review approvals.
- ☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.
- ☒ **Require status checks to pass before merging**
Choose which [status checks](#) must pass before branches can be merged into **master**. When enabled, commits must first be pushed to another branch, then merged or pushed directly to **master** after status checks have passed.
- ☒ **Require branches to be up to date before merging**
This ensures the branch has been tested with the latest code on **master**.

Sorry, we couldn't find any status checks in the last week for this repository.
[Learn more about status checks on GitHub.](#)

SETUP GITHUB

☒ **Require status checks to pass before merging**

Choose which [status checks](#) must pass before branches can be merged into **master**. When enabled, commits must first be pushed to another branch, then merged or pushed directly to **master** after status checks have passed.

☒ **Require branches to be up to date before merging**

This ensures the branch has been tested with the latest code on **master**.

Status checks found in the last week for this repository

☒ ci/circleci

Required

☐ **Include administrators**

Enforce all configured restrictions for administrators.



INTEGRATING CODE REVIEW INTO PROJECT WORKFLOW

- ▶ Easy with tools like Github
- ▶ No excuse not to start using today



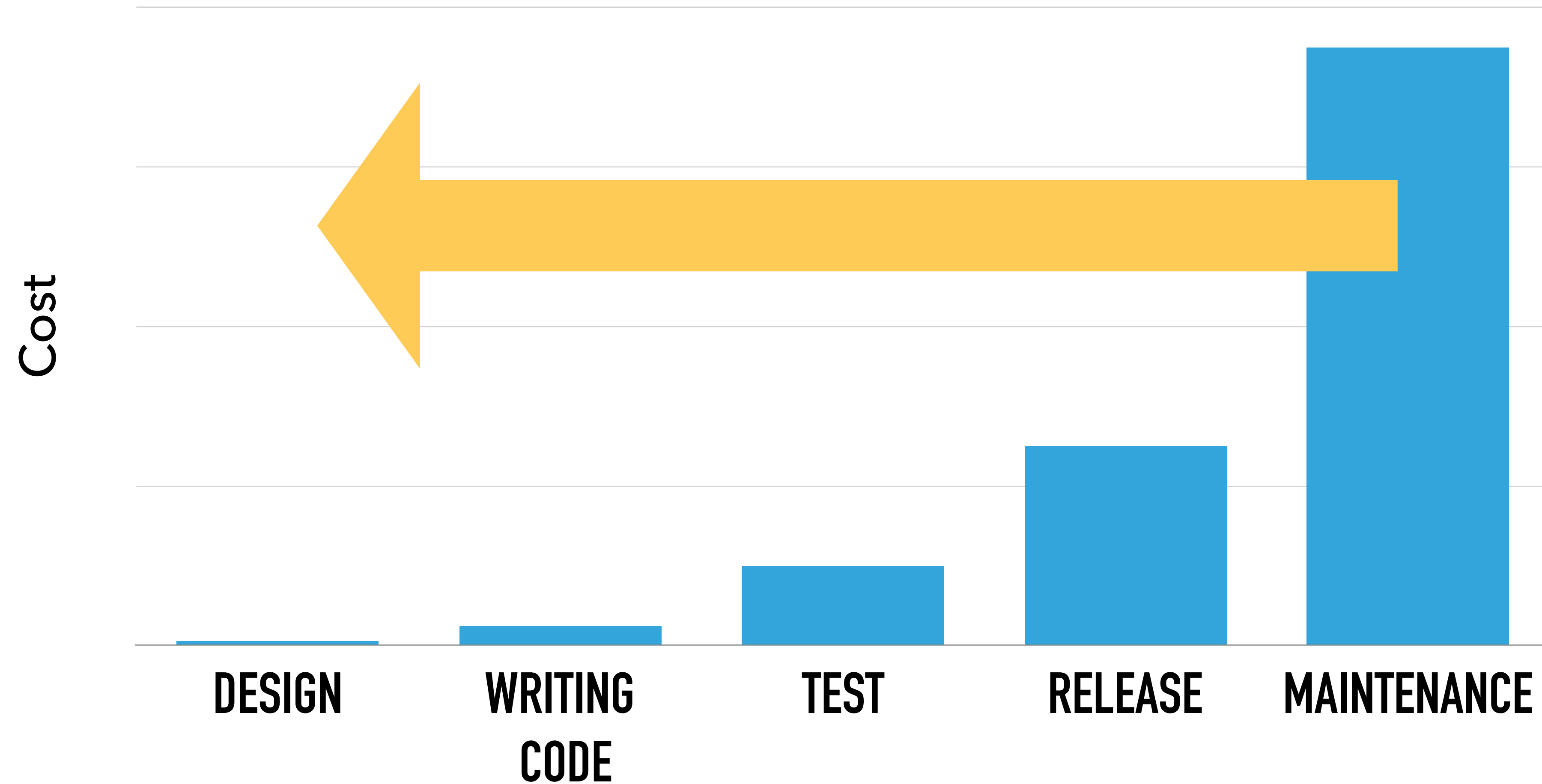
WHAT WE'VE COVERED

- ▶ Why
- ▶ Code quality
- ▶ What is code review
- ▶ Benefits
- ▶ Implementation
- ▶ Tips
- ▶ Integrating code review into project workflow

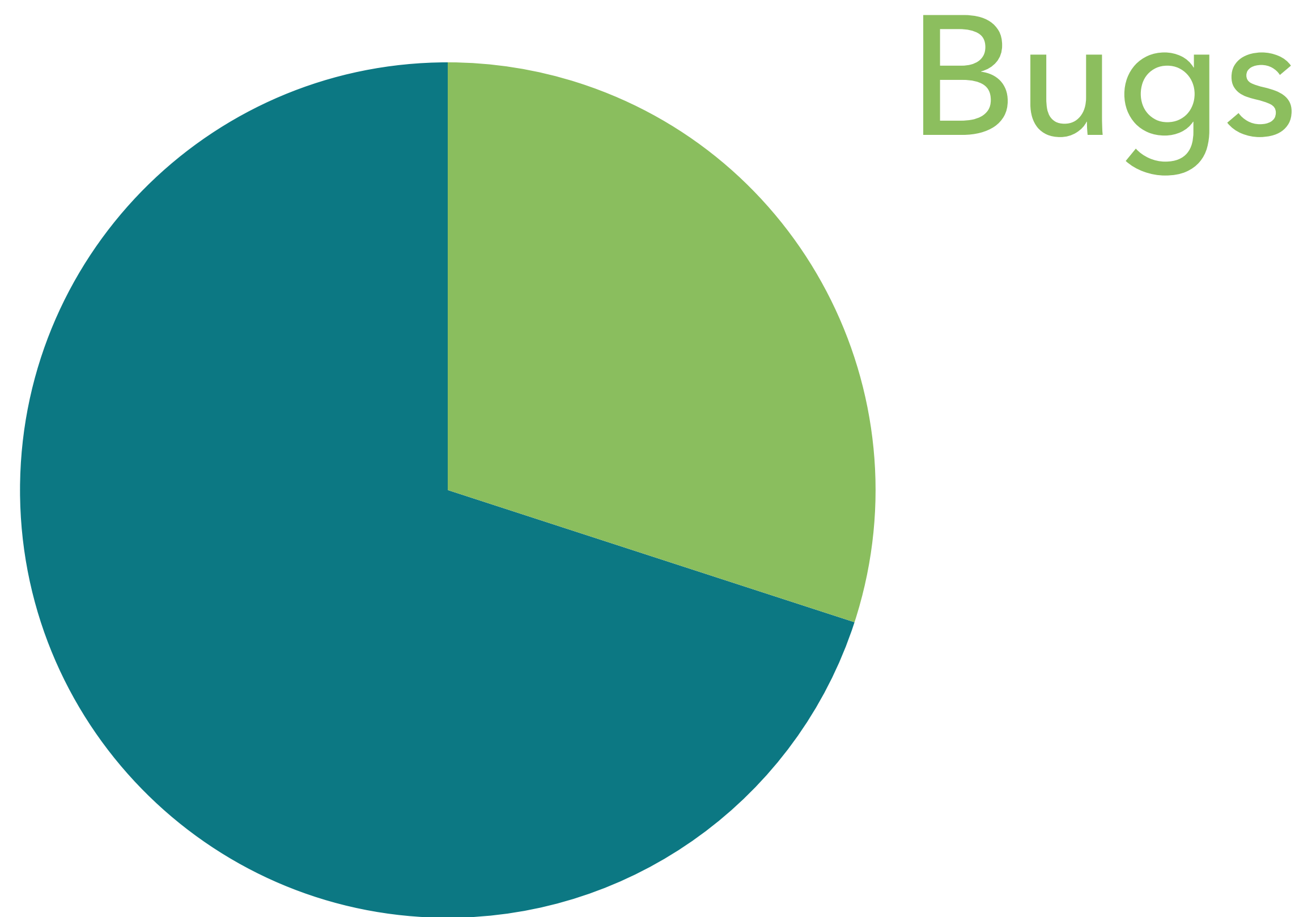


WRAP UP

FIND BUGS SOONER



WHAT ARE DEFECTS?

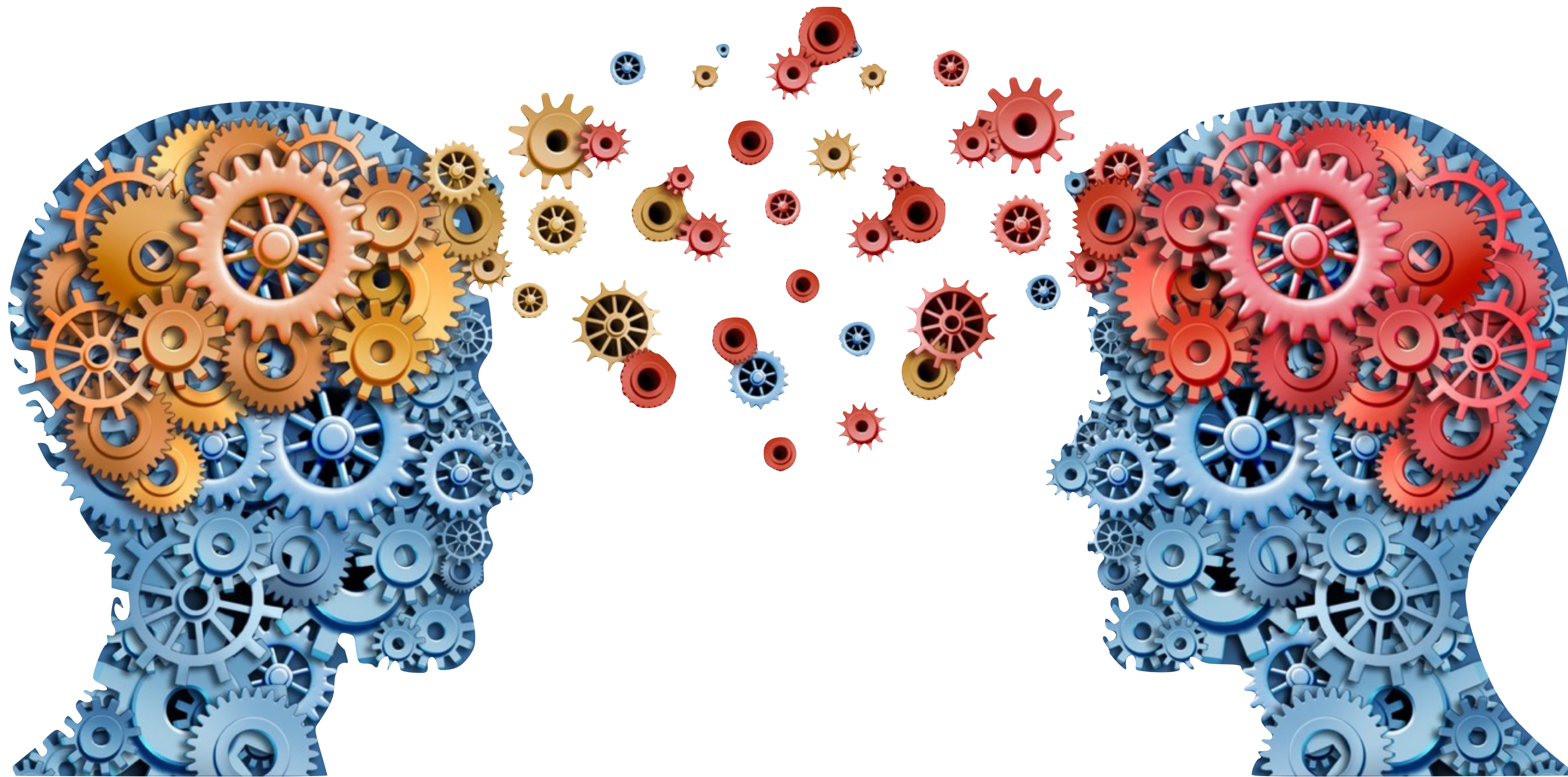


Evolvability

[1, 2]



SPREAD THE KNOWLEDGE



MENTORING



WRAP UP

YOU'RE BEING WATCHED!



WRAP UP

YOU'RE BEING WATCHED!



[5]





SUMMARY: REVIEW CHECKLIST

- ▶ Security problems
- ▶ Correct tests
- ▶ Clean code
- ▶ Bugs



#coding-standards

☆ | 👤 4 | 📌 0 | [Add a topic](#)



dave 10:55 AM

Naming: Do not use abbreviations



8 replies

Last reply 3 months ago



dave 11:29 AM

Always type hint for parameters and return types for methods and functions (edited)



2 replies

Last reply 3 months ago



dave 2:31 PM

Type hint all member data in classes



2 replies

Last reply 3 months ago



dave 2:55 PM

Use void return type if method does not return anything



1 reply

3 months ago



dave 2:56 PM

Non void methods must always return a value explicitly (even if it's null)



WRAP UP

EVERYONE SHOULD CODE REVIEW



@daveliddament



EFFECTIVE CODE REVIEW REDUCES OVERALL COST OF SOFTWARE DEVELOPMENT





REFERENCES

- ▶ [1] Mika V. Mantyla and Casper Lassenius "What Types of Defects Are Really Discovered in Code Reviews?" IEEE Transactions on Software Engineering
- ▶ [2] Harvey Siy, Lawrence Votta "Does The Modern Code Inspection Have Value?"
- ▶ [3] R.K. Bandi, V.K. Vaishnavi, and D.E. Turk, "Predicting Maintenance Performance Using Object-Orientated Design Complexity Metrics"
- ▶ [4] R.D. Banker, S.M. Datar, C.F. Kemerer, and D. Zweig, "Software Complexity and Maintenance Costs,"
- ▶ [5] <https://www.bbc.co.uk/news/uk-37502136>
- ▶ [6] <https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>



CREDITS

IMAGE CREDITS

- ▶ © Can Stock Photo / kmitu
- ▶ © Kirsty Andrews
- ▶ © Jez Kent
- ▶ © Can Stock Photo / focalpoint
- ▶ © Can Stock Photo / sellingpix
- ▶ © Can Stock Photo / robwilson39
- ▶ © Can Stock Photo / lindwa
- ▶ © Can Stock Photo / piedmont_photo
- ▶ © Can Stock Photo / pressmaster
- ▶ © Can Stock Photo / marish

