Effective Code Review

Dave Liddament

WHAT DID WE LEARN?

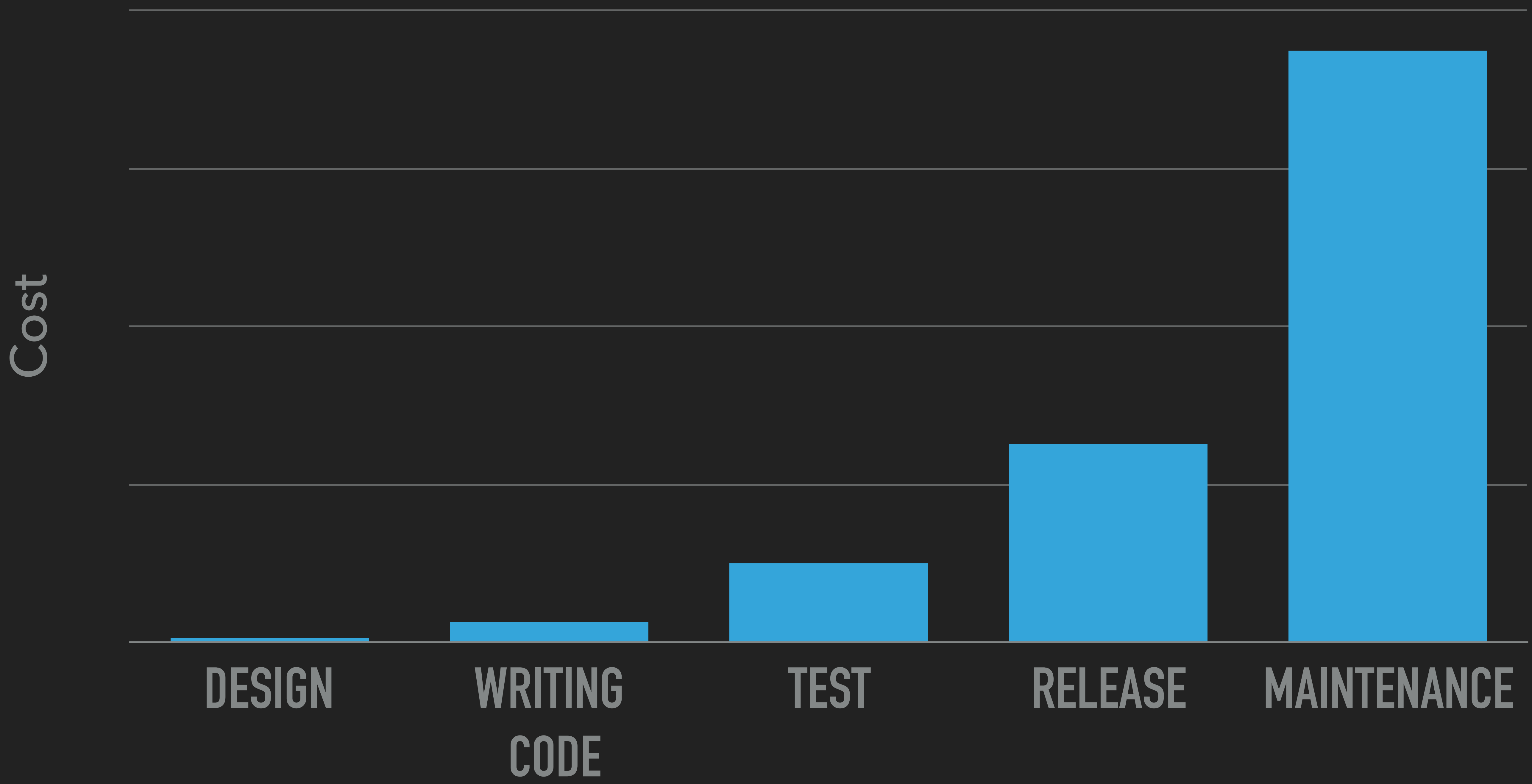
- We all make mistakes.
- If the mistakes are spotted and rectified quickly the consequences of the mistakes are minimal.
- We want to make mistakes easy to spot.

WHAT ABOUT SOFTWARE DEVELOPMENT?

WHAT ABOUT SOFTWARE DEVELOPMENT?

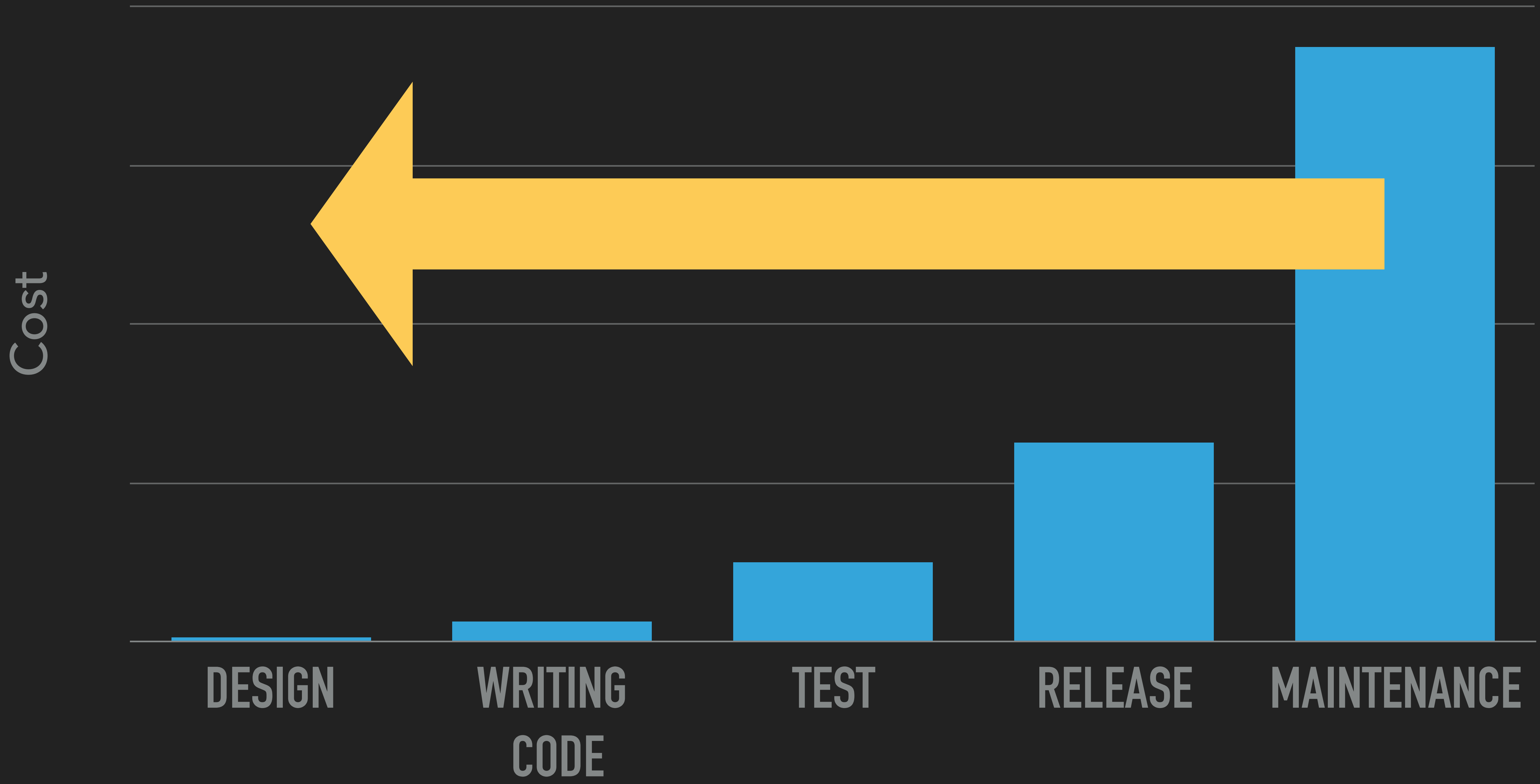
- We all make mistakes.
- If the mistakes are spotted and rectified quickly the consequences of the mistakes are minimal.
- We want to make mistakes easy to spot.

FIND BUGS SOONER



COST OF A BUG

FIND BUGS SOONER



EFFECTIVE CODE REVIEW REDUCES OVERALL COST OF SOFTWARE DEVELOPMENT

Dave Liddament

@daveliddament

Lamp Bristol

15+ years software development (PHP, Java, Python, C)

Organise PHP-SW user group and Bristol PHP Training

AGENDA

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

WHY

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

INCREASING CODE QUALITY

- Code Review
- Tests
- Static code analysis
- Continuous Integration (e.g. Jenkins, Travis, CircleCI)
- Modern IDE

AGENDA

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

WHAT IS CODE REVIEW

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

Wikipedia

WHAT IS CODE REVIEW

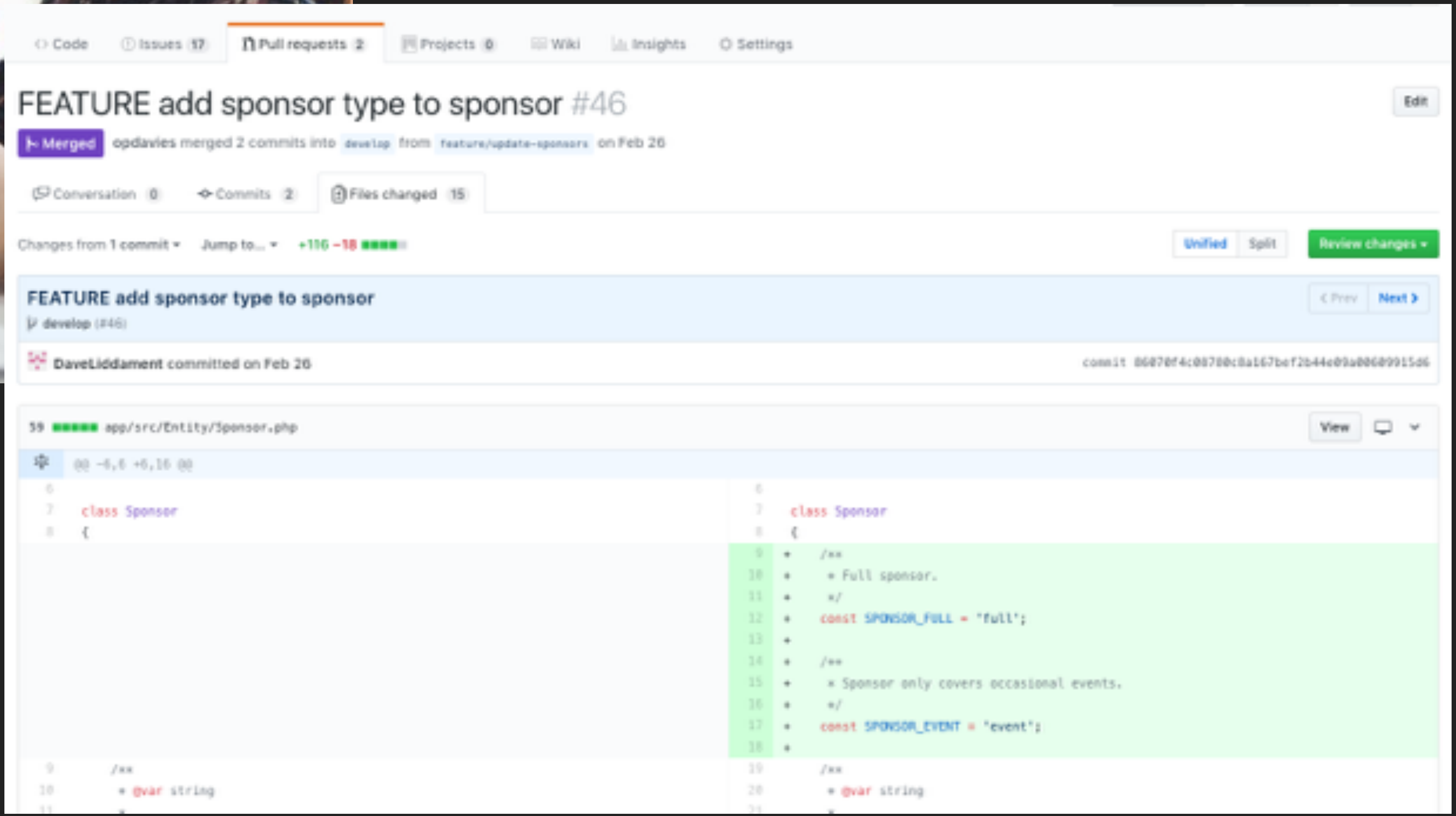
> IT IS INTENDED TO FIND MISTAKES OVERLOOKED IN SOFTWARE DEVELOPMENT, IMPROVING THE OVERALL QUALITY OF SOFTWARE.

Wikipedia

HOW IS IT DONE

HOW IS IT DONE

HOW IS IT DONE



FEATURE add sponsor type to sponsor #46

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

+116 −18

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 @@

class Sponsor                         class Sponsor
{                                     {
                                      +    /**
                                      +     * Full sponsor.
                                      +     */
                                      +    const SPONSOR_FULL = 'full';
                                      +
                                      +    /**
                                      +     * Sponsor only covers occasional events.
                                      +     */
                                      +    const SPONSOR_EVENT = 'event';
                                      +
    /**                                   /**
     * @var string                         * @var string
```

AGENDA

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

BENEFITS

- Reduce defects
- Find security vulnerabilities
- Spread knowledge
- Mentoring
- Peer pressure improves code quality

WHAT ARE DEFECTS?

[1, 2]

WHAT ARE DEFECTS?

Bugs

[1, 2]

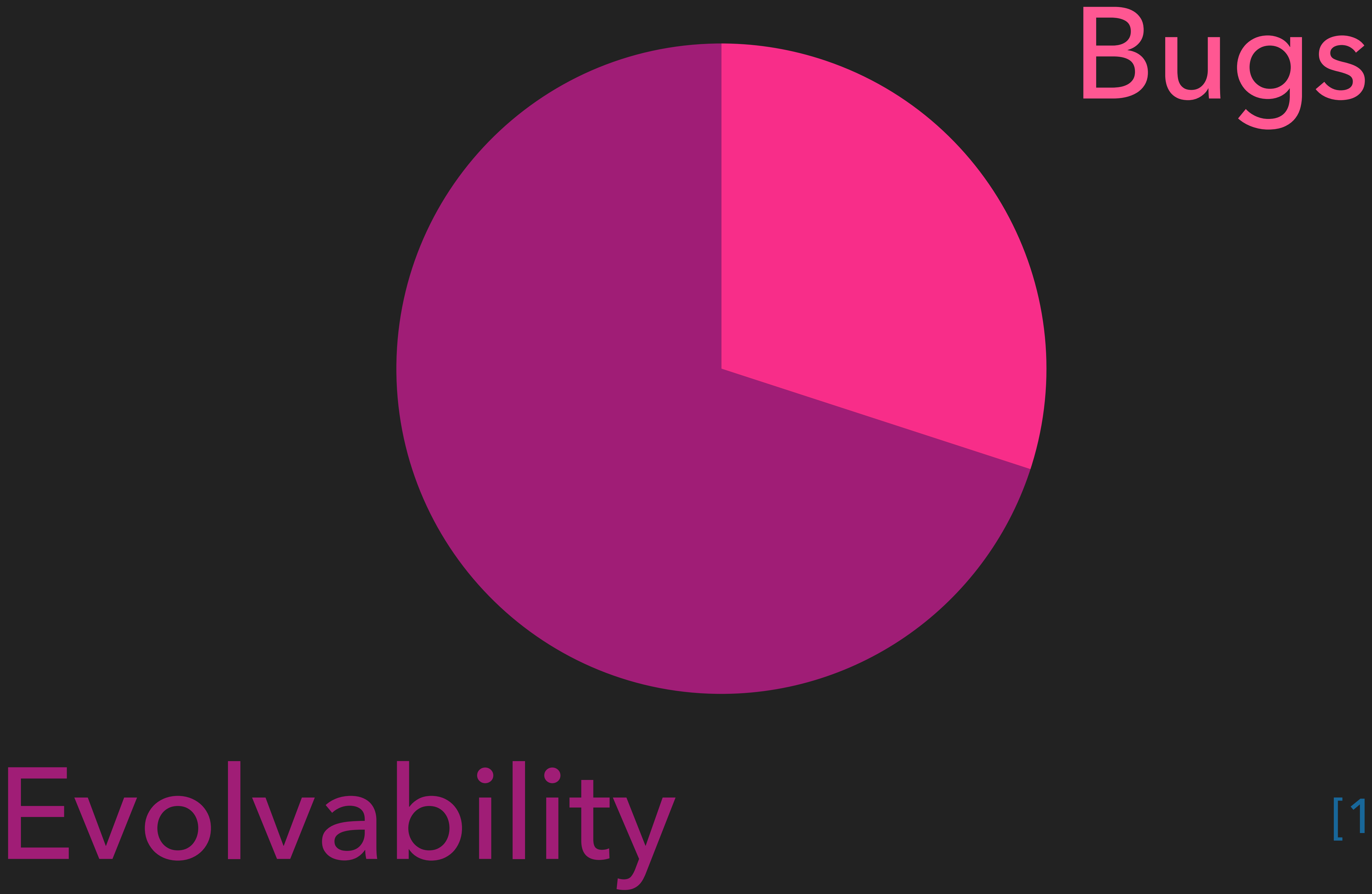
WHAT ARE DEFECTS?

Bugs

Evolvability

[1, 2]

WHAT ARE DEFECTS?



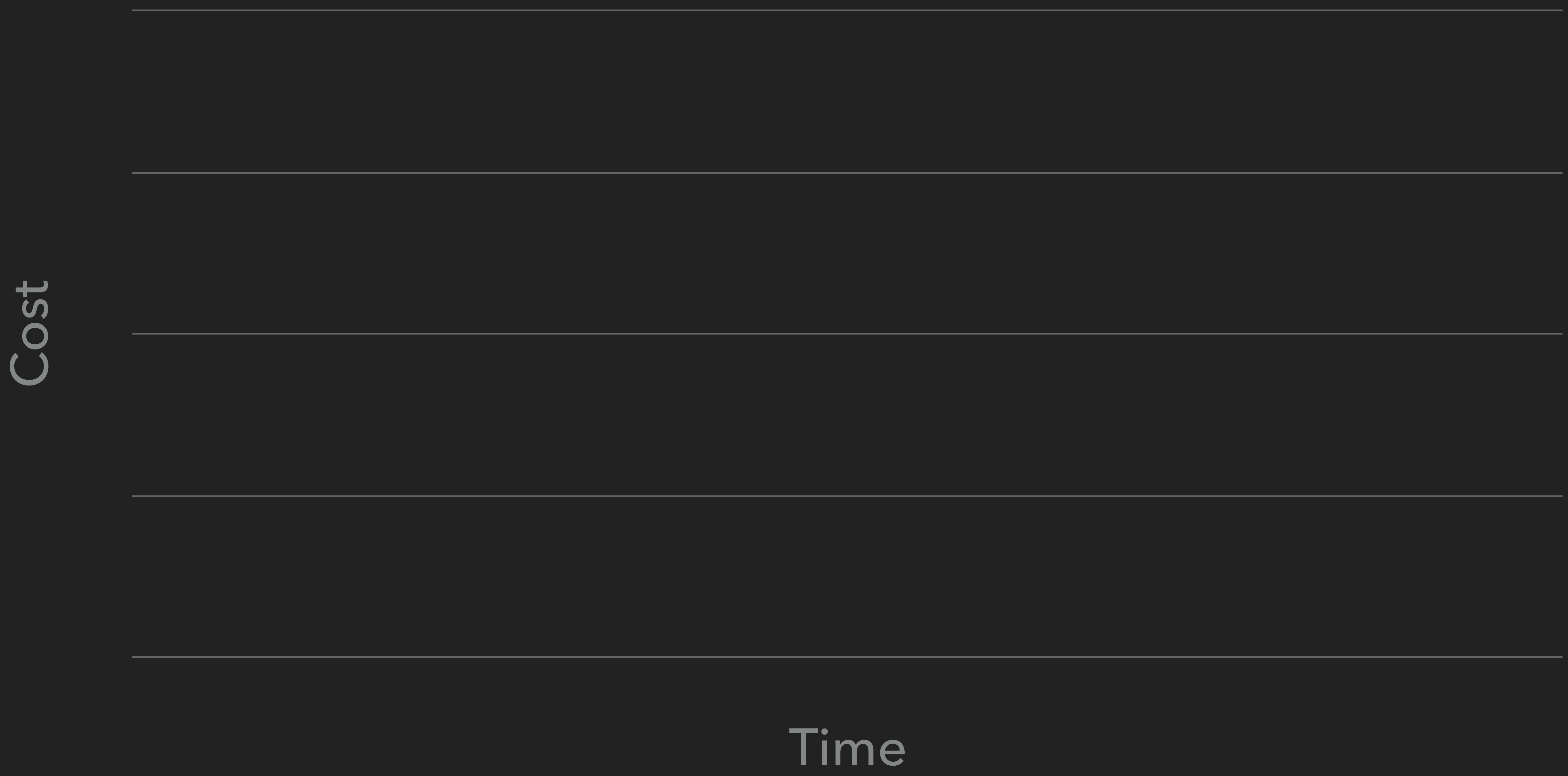
[1, 2]

EVOLVABILITY IS IMPORTANT

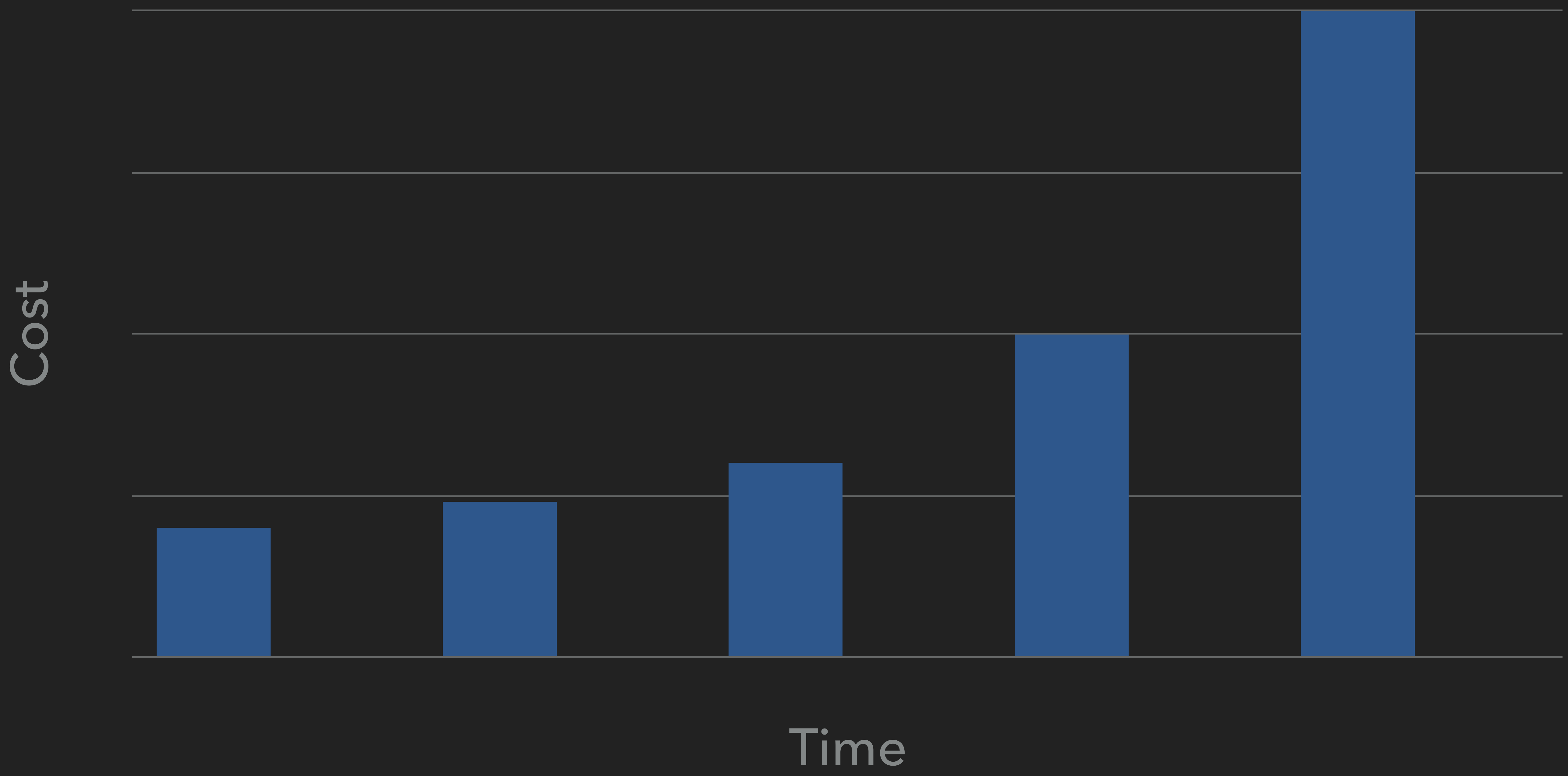
- Low evolvability costs money:
  - New features took 28% longer to implement [3]
  - Fixing bugs took 36% longer [3]
- Software structure may account for 25% of total maintenance costs [4]

COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME

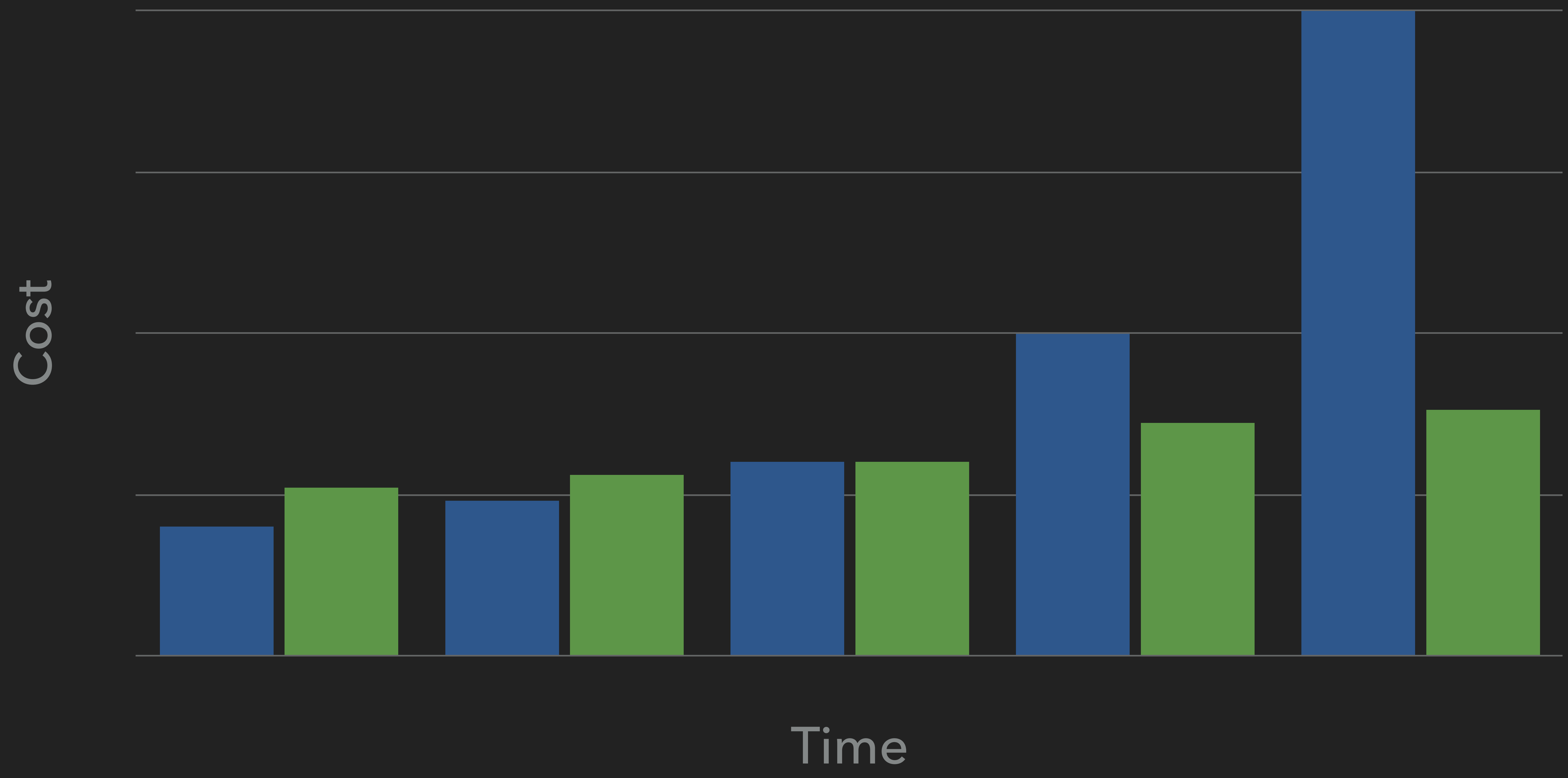
COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



TESTS WILL FIND DEFECTS

```
Scenario: Navigation at T junction in a cave
  Given: I am coming up to a T
  When: Before I pass the T junction
  Then: I should drop a cookie
```

TESTS WILL FIND DEFECTS

```
Scenario: Navigation at T junction in a cave
  Given: I am coming up to a T
  When: Before I pass the T junction
  Then: I should drop a cookie on the exit side.
```

TAKE AWAY

- The majority of code review comments will not be “bugs”.
- Most comments will be code improvements.
  - Lower technical debt. Lower overall cost.
- Remember to sell the right metric to management.

SECURITY REVIEW

SECURITY REVIEW

- Files that shouldn't be there? (e.g. malware)

SECURITY REVIEW

- Files that shouldn’t be there? (e.g. malware)
- Writing sensitive data to logs (e.g. password)

SECURITY REVIEW

- Files that shouldn’t be there? (e.g. malware)
- Writing sensitive data to logs (e.g. password)
- OWASP top 10

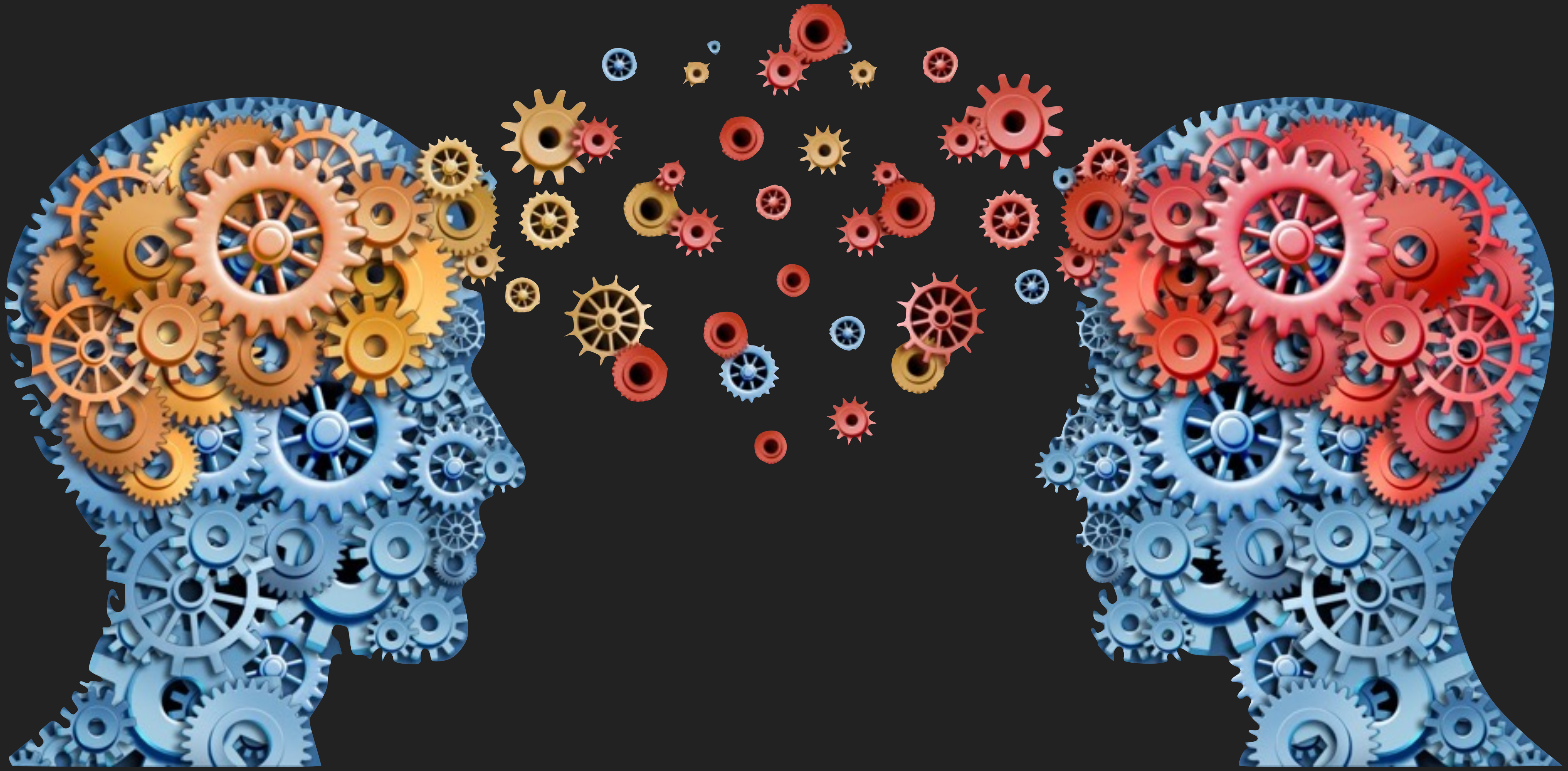
SECURITY REVIEW

- Files that shouldn’t be there? (e.g. malware)
- Writing sensitive data to logs (e.g. password)
- OWASP top 10
- Not / weakly hashing passwords

SECURITY REVIEW

- Files that shouldn’t be there? (e.g. malware)
- Writing sensitive data to logs (e.g. password)
- OWASP top 10
- Not / weakly hashing passwords
- Rolled your own authentication / hashing / encryption algorithms

SPREAD THE KNOWLEDGE



NO MORE SILOS

NO MORE SILOS

SPREAD THE KNOWLEDGE

NO MORE SILOS

SPREAD THE KNOWLEDGE

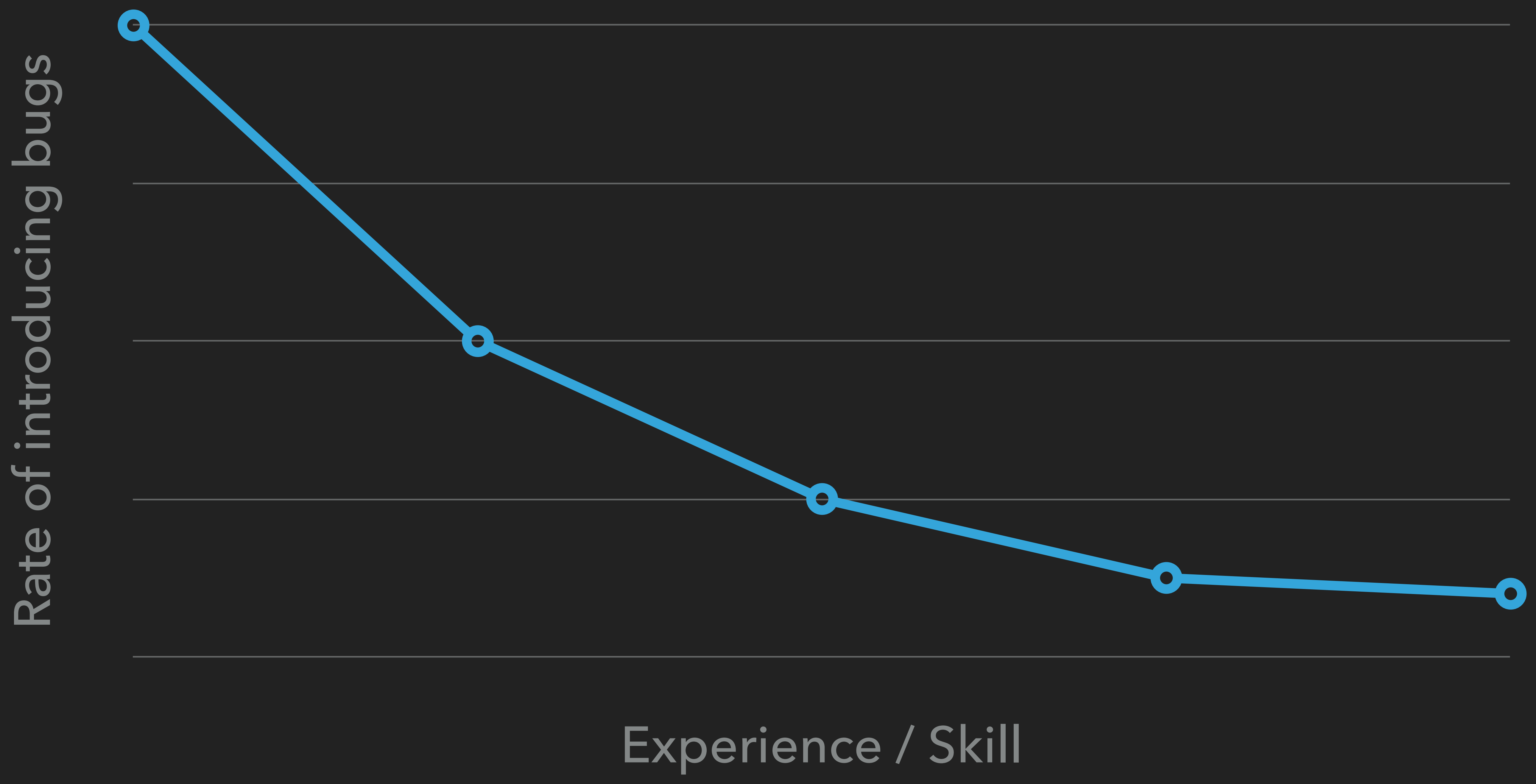
NO MORE SILOS

SPREAD THE KNOWLEDGE

NO MORE SILOS

MENTORING

SKILL AND EXPERIENCE VS BUGS



YOU'RE BEING WATCHED!

YOU'RE BEING WATCHED!

[5]

BENEFITS

- Reduce defects
- Find security vulnerabilities
- Spread knowledge
- Mentoring
- Peer pressure improves code quality

AGENDA

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

ONLY CHECK WHAT WE CAN NOT AUTOMATE

WE CAN AUTOMATE

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

WHAT ARE WE LOOKING FOR IN CODE REVIEW

DO THE TESTS TEST THE REQUIRED FUNCTIONALITY?

QUESTIONS TO ASK IN CODE REVIEW

ARE THE TESTS ADEQUATE?

IS THIS ENOUGH TESTING?

```
Scenario: Navigation at T junction in a cave
  Given: I am coming up to a T.
  When: Before I pass the T junction.
  Then: I should drop a cookie on the exit side.
```

HOW MANY TESTS DO WE NEED?

```
class Person
{

  /**
   * Returns true if the person is 18 or over
   */
  public function isAdult(int $age): bool
  {
    .. some implementation ..
  }

}
```

HOW MANY TESTS DO WE NEED (2)?

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

WILL I UNDERSTAND THIS CODE IN 6 MONTHS TIME?

COST OF WRITING CODE CODE

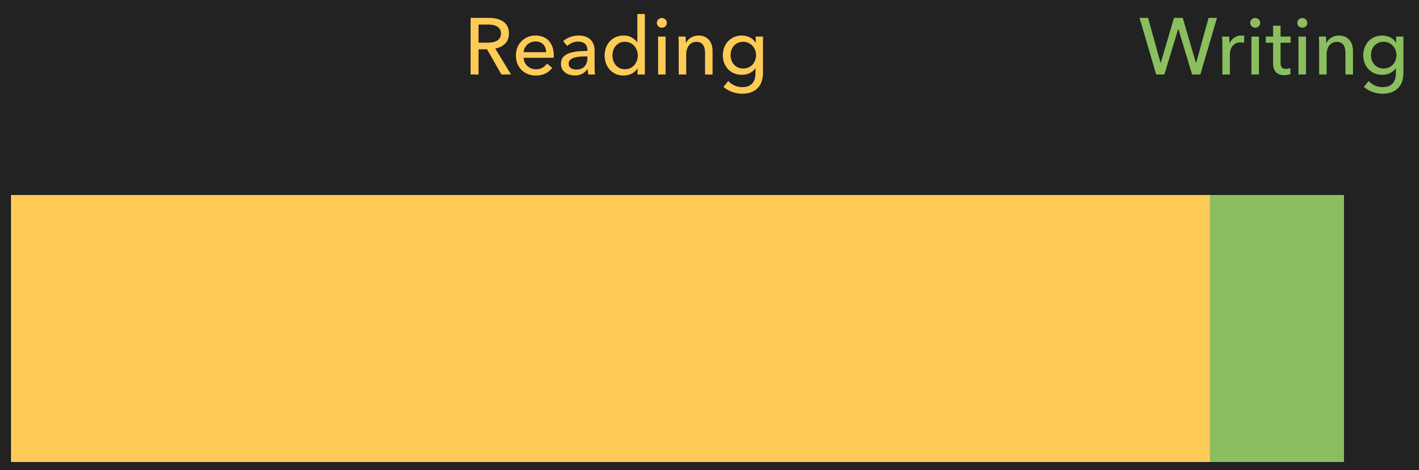
COST OF WRITING CODE CODE

> “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.”

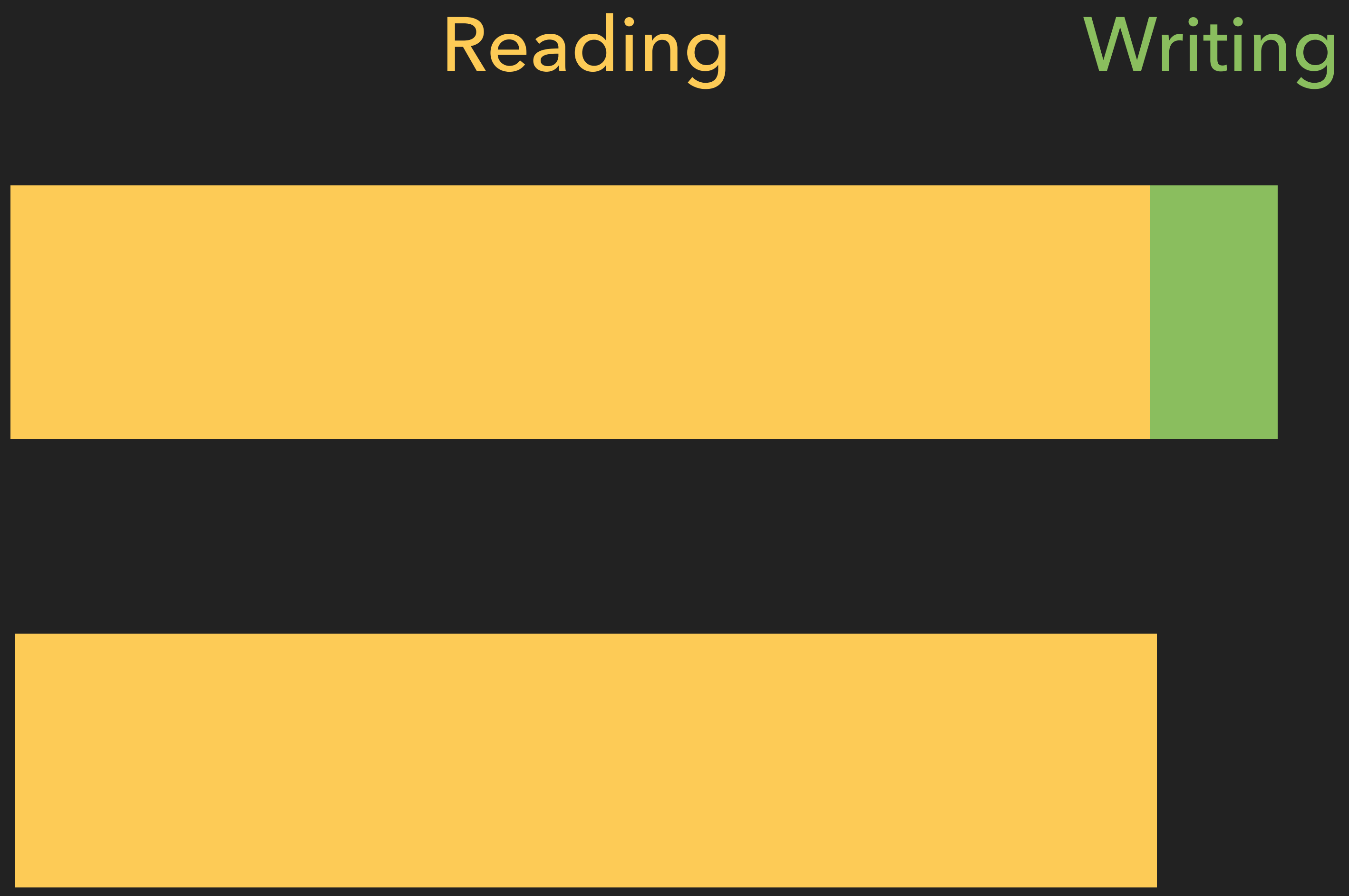
Robert C. Martin (Clean Code)

HOW CAN WE REDUCE COST OF WRITING CODE

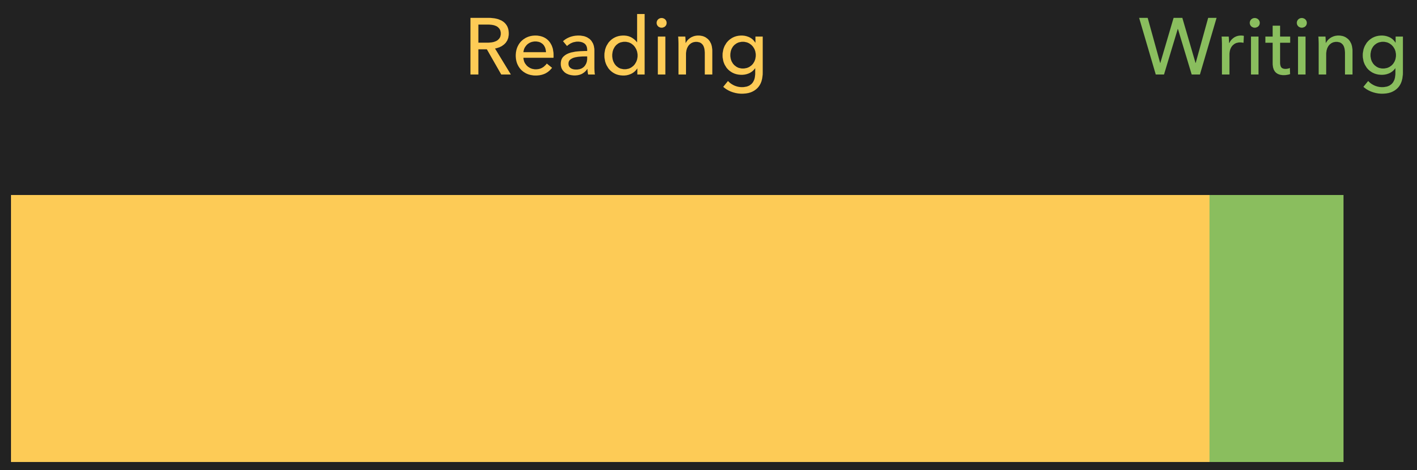
COST OF WRITING CODE



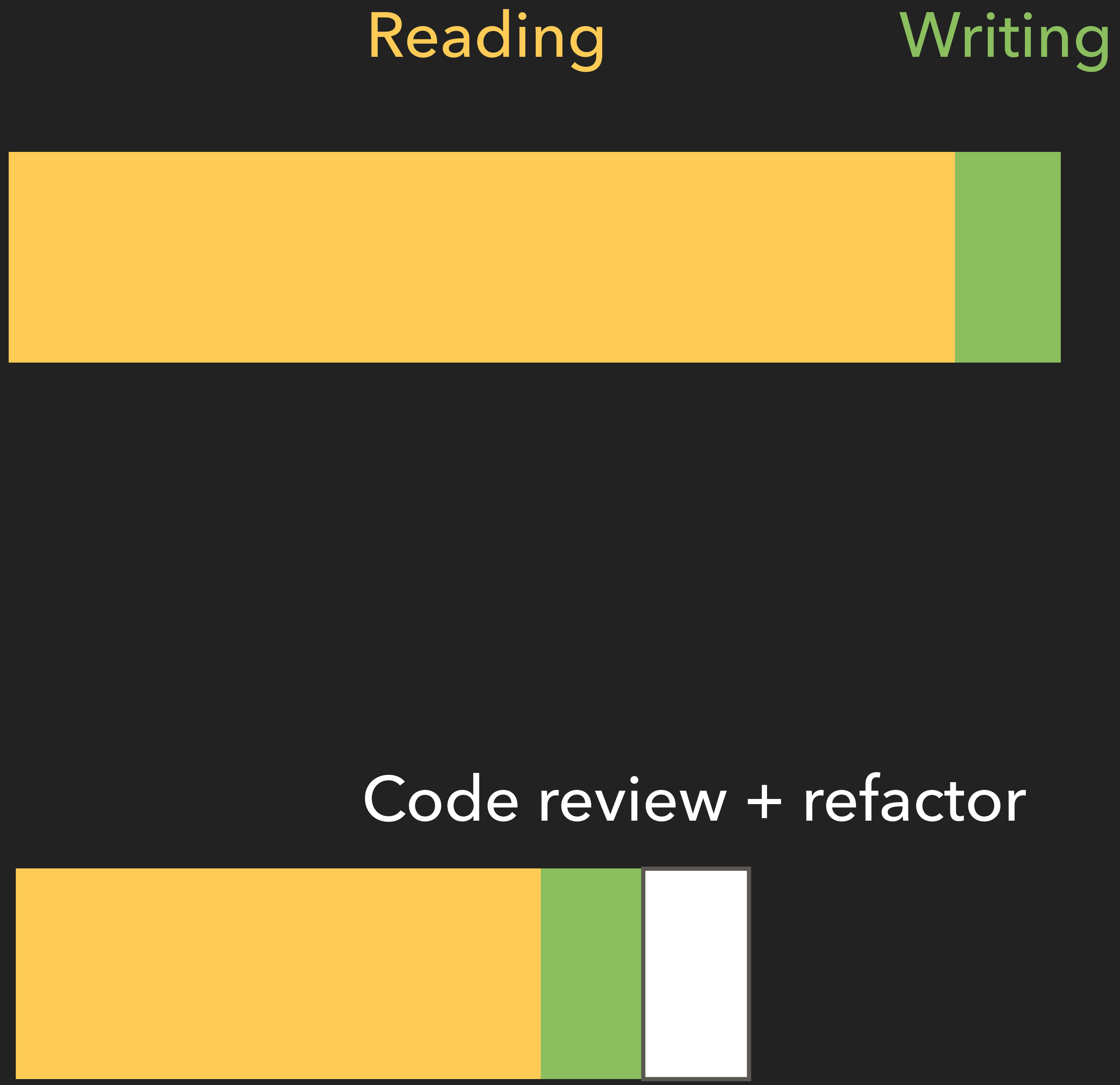
COST OF WRITING CODE



COST OF WRITING CODE



COST OF WRITING CODE



Reading

Writing

Code review + refactor

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}());
    }
}
```

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
{

  … other methods …

  private $url;

  /**
   * @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
{

  … other methods …

  private $url;

  /**
   * @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
{

  … other methods …

  private $url;

  /**
   * @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 …
}
```

WHAT DOES 1 MEAN?

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

ARE WE FOLLOWING PROJECT CONVENTIONS?

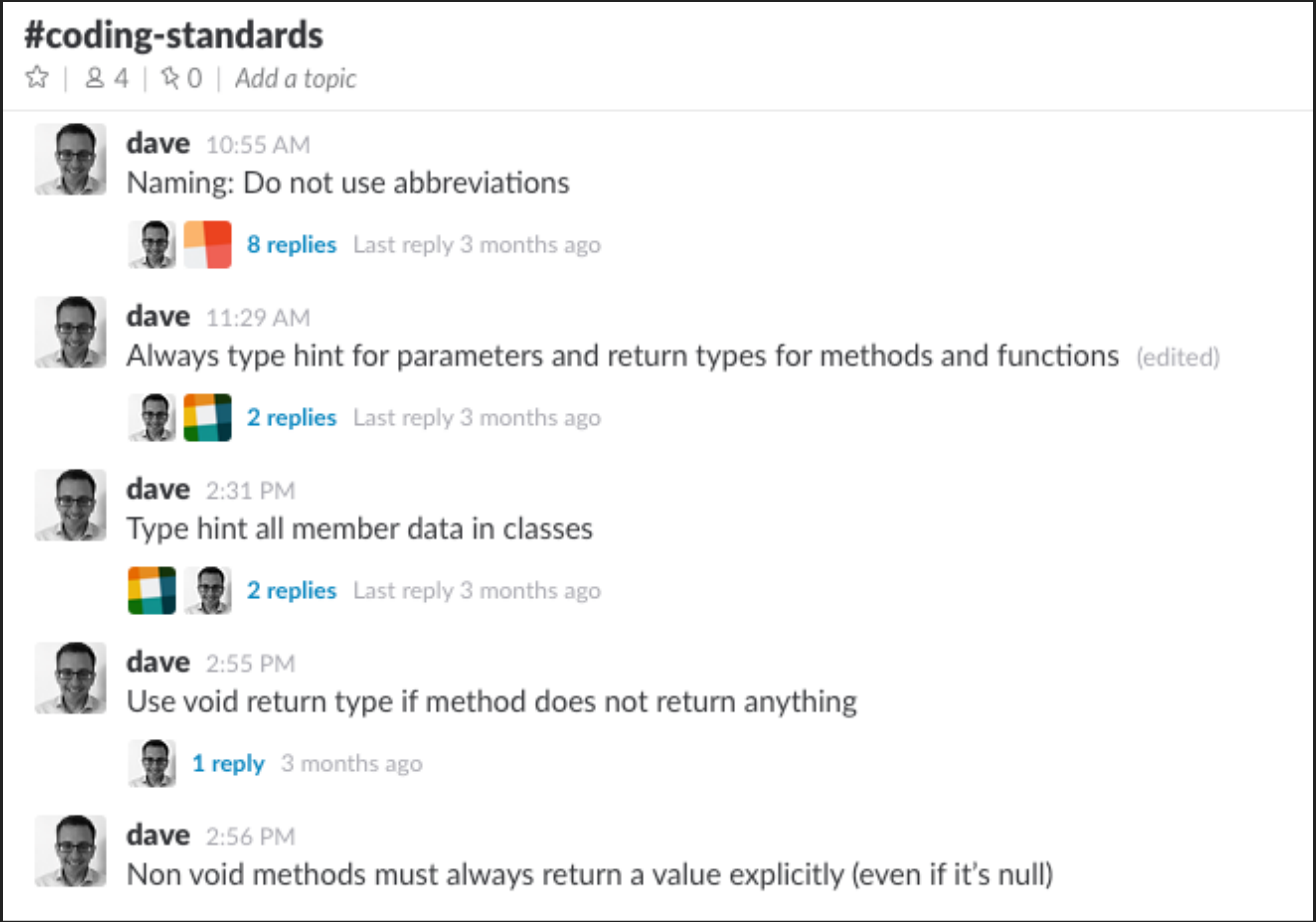
ARE WE FOLLOWING PROJECT CONVENTIONS

```
interface LocationRepository
{

  public function findClosestTo($point);

  public function findByName($name);

  public function findBySlug($slug);

  public function searchForLocation($name, $type);

  public function findAllByType($type);

}
```

ARE WE FOLLOWING PROJECT CONVENTIONS

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)

QUESTIONS TO ASK IN CODE REVIEW

IS 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 ..
  }

}
```

HOW DO WE MAKE THIS MORE OBVIOUS (2)

```
class MarketingCampaign
{

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

}
```

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 ..
  }

}
```

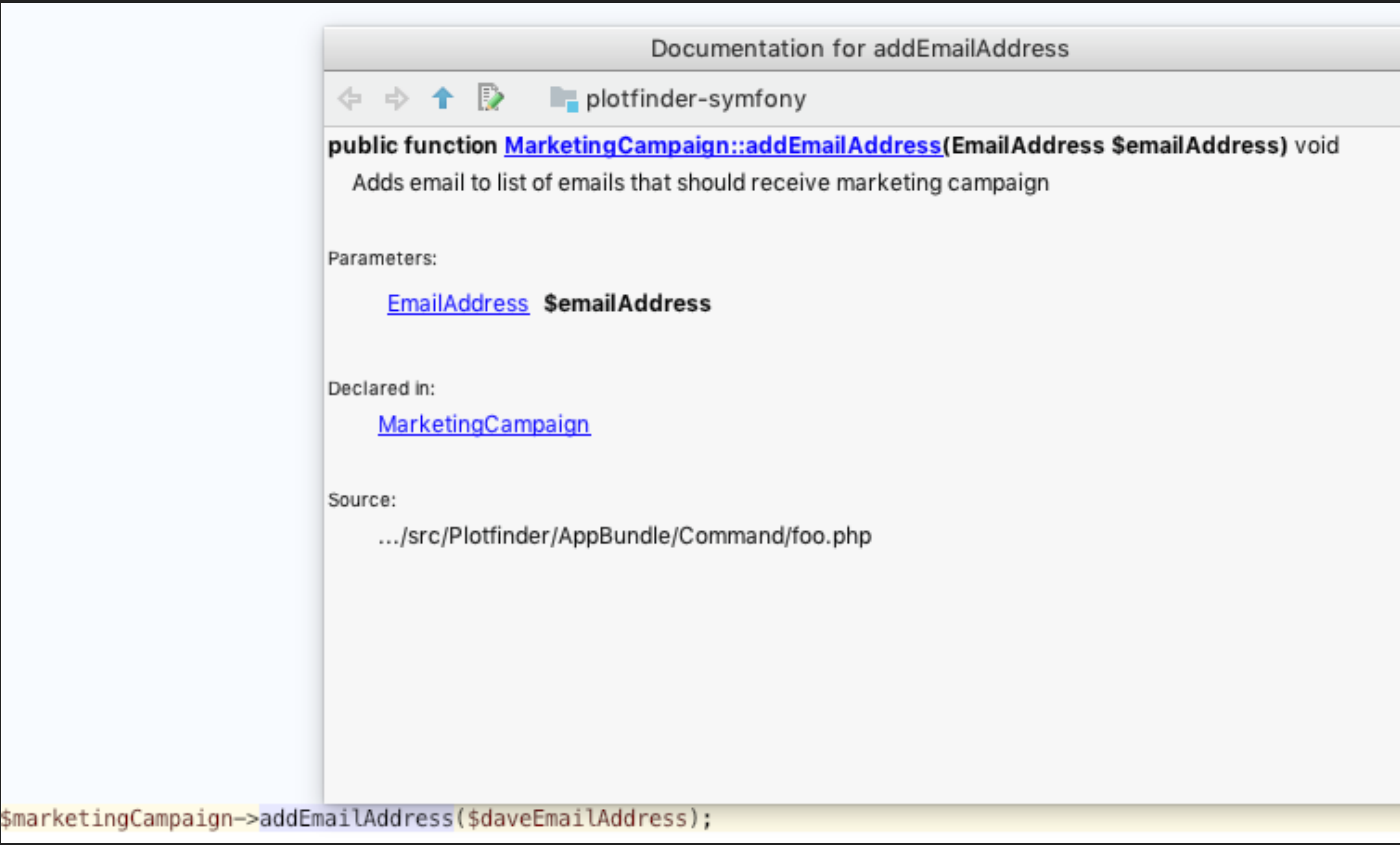
HOW DO WE MAKE THIS MORE OBVIOUS (4)

```
class MarketingCampaign
{
  /**
   * Add email address that should received campaign
   * messages.
   */
  public function addEmailAddress(string $emailAddress
  ): void {
    .. some implementation ..
  }

}
```

HOW DO WE MAKE THIS MORE OBVIOUS (5)

```
class MarketingCampaign
{
  /**
   * Add email address that should received campaign
   * messages.
   */
  public function addEmailAddress(
    EmailAddress $emailAddress
  ): void {
    .. some implementation ..
  }

}
```

CAN I UNDERSTAND THE FUNCTIONALITY OF THE CODE WITHOUT READING IT?



public function [MarketingCampaign::addEmailAddress]()**(EmailAddress $emailAddress)** void

Adds email to list of emails that should receive marketing campaign

Parameters:

[EmailAddress]() **$emailAddress**

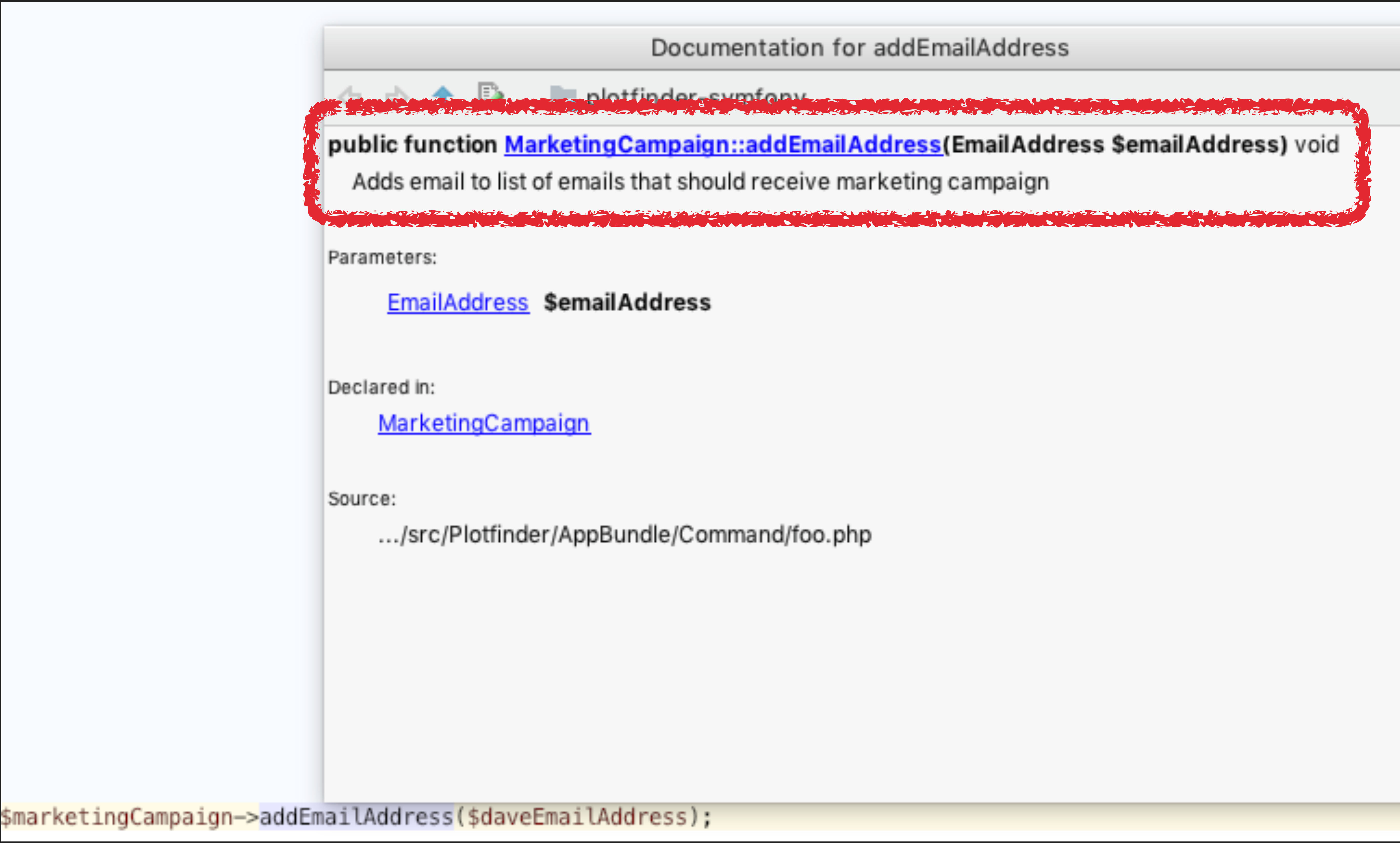
Declared in:

[MarketingCampaign]()

Source:

.../src/Plotfinder/AppBundle/Command/foo.php

```
$marketingCampaign->addEmailAddress($daveEmailAddress);
```



Documentation for addEmailAddress

public function MarketingCampaign::addEmailAddress(EmailAddress $emailAddress) void

Adds email to list of emails that should receive marketing campaign

Parameters:

EmailAddress **$emailAddress**

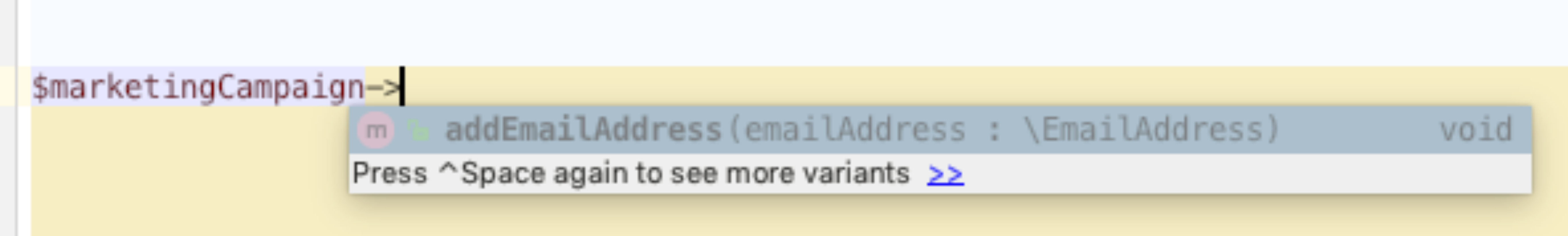
Declared in:

MarketingCampaign

Source:

.../src/Plotfinder/AppBundle/Command/foo.php

```
$marketingCampaign->addEmailAddress($daveEmailAddress);
```



```
$marketingCampaign->
```

addEmailAddress(emailAddress : \EmailAddress) void

Press ^Space again to see more variants >>

QUESTIONS TO ASK IN CODE REVIEW

DO THE COMMENTS MATCH THE CODE?

HAS DEFENSIVE CODING BEEN USED?

HAS DEFENSIVE CODING BEEN USED

```
switch($status) {

  case ‘started’:
   … do something …
   break;

  case ‘finished’:
   … do something …
   break;

  case ‘quit’:
   … do something …
   break;
}
```

MISSING DEFAULT

```
switch($status) {

  case ‘started’:
   … do something …
   break;

  case ‘finished’:
   … do something …
   break;

  case ‘quit’:
   … do something …
   break;
}
```

HAS DEFENSIVE CODING BEEN USED

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

HAS DEFENSIVE CODING BEEN USED

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;
}
```

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 …
```

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

CAN ARCHITECTURE BE IMPROVED? (E.G. SOLID)

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
- Can I understand functionality without reading the code
- 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

Tips

EVERYONE SHOULD CODE REVIEW

WHAT MAKES CODE EASY TO TO REVIEW

> ASK PROGRAMMERS TO REVIEW 10 LINES OF CODE THEY'LL FIND 10 ISSUES…

Anyone who's done code review

WHAT MAKES CODE EASY TO TO REVIEW

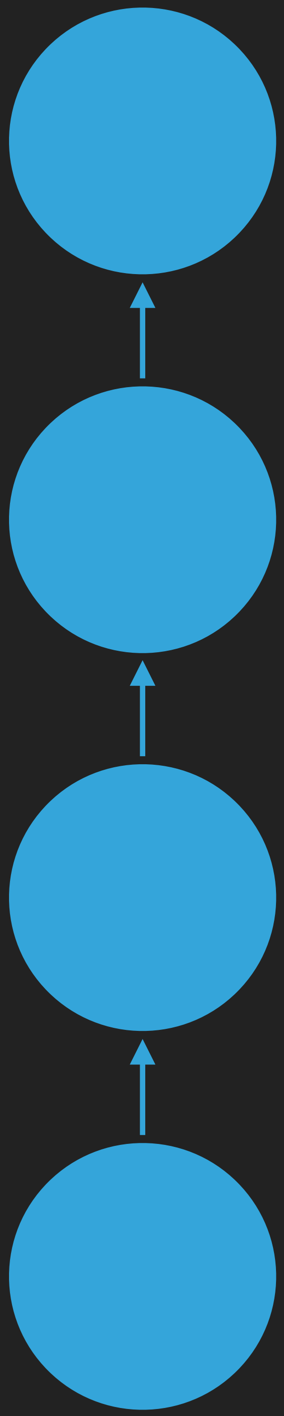
> ASK THEM TO DO 500 LINES THEY'LL SAY IT'S GOOD TO GO

Anyone who's done code review

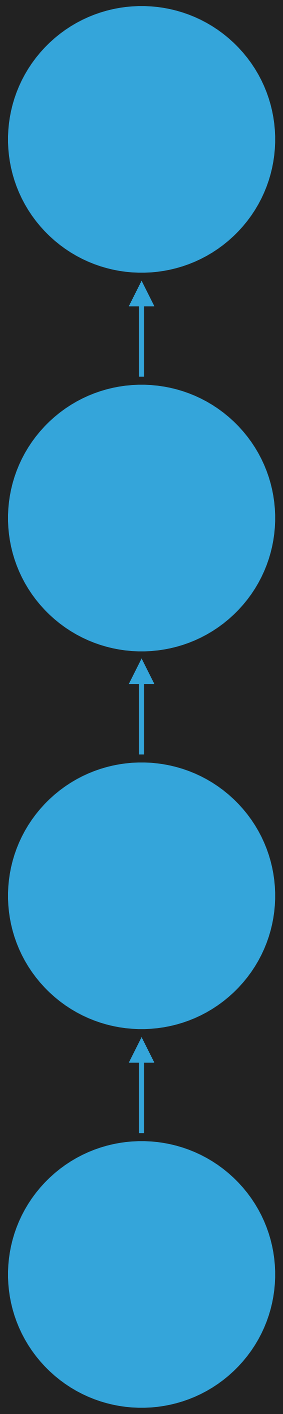
WHAT MAKES CODE EASY TO TO REVIEW

SMALL COMMITS

SMALL COMMITS

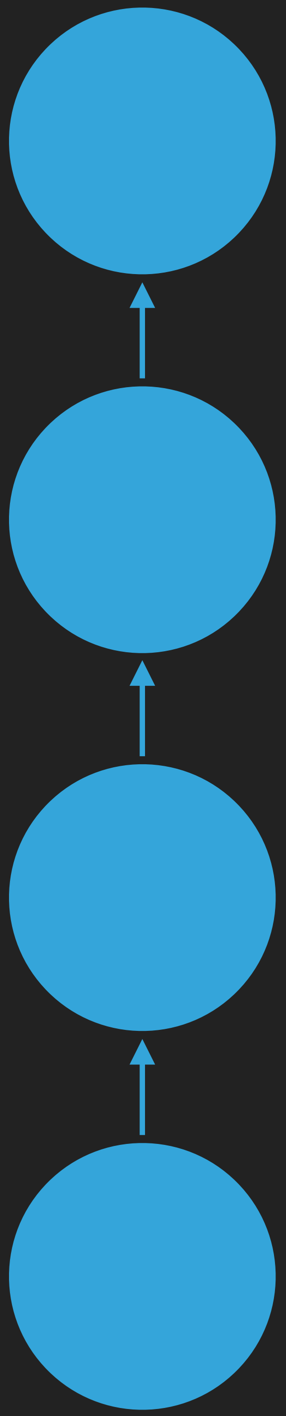


SMALL COMMITS



DEPRECATE: The price calculation service

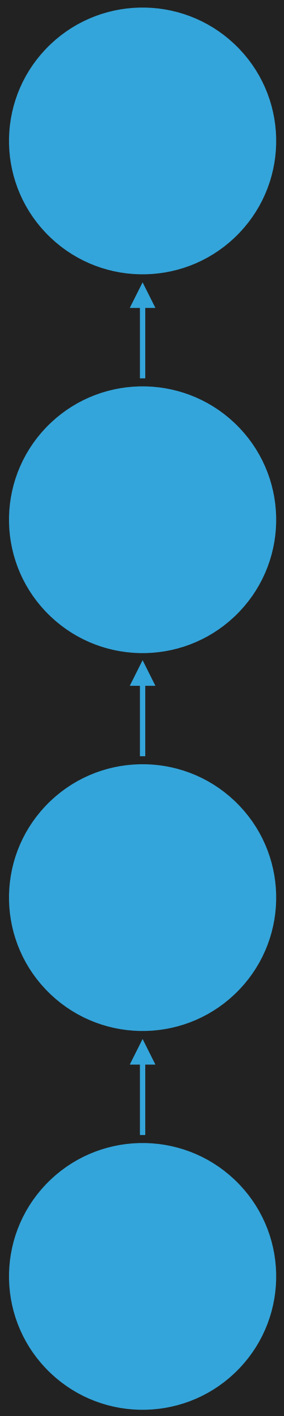
SMALL COMMITS



ADD: Facade to 3rd party price calculation service

DEPRECATE: The price calculation service

SMALL COMMITS

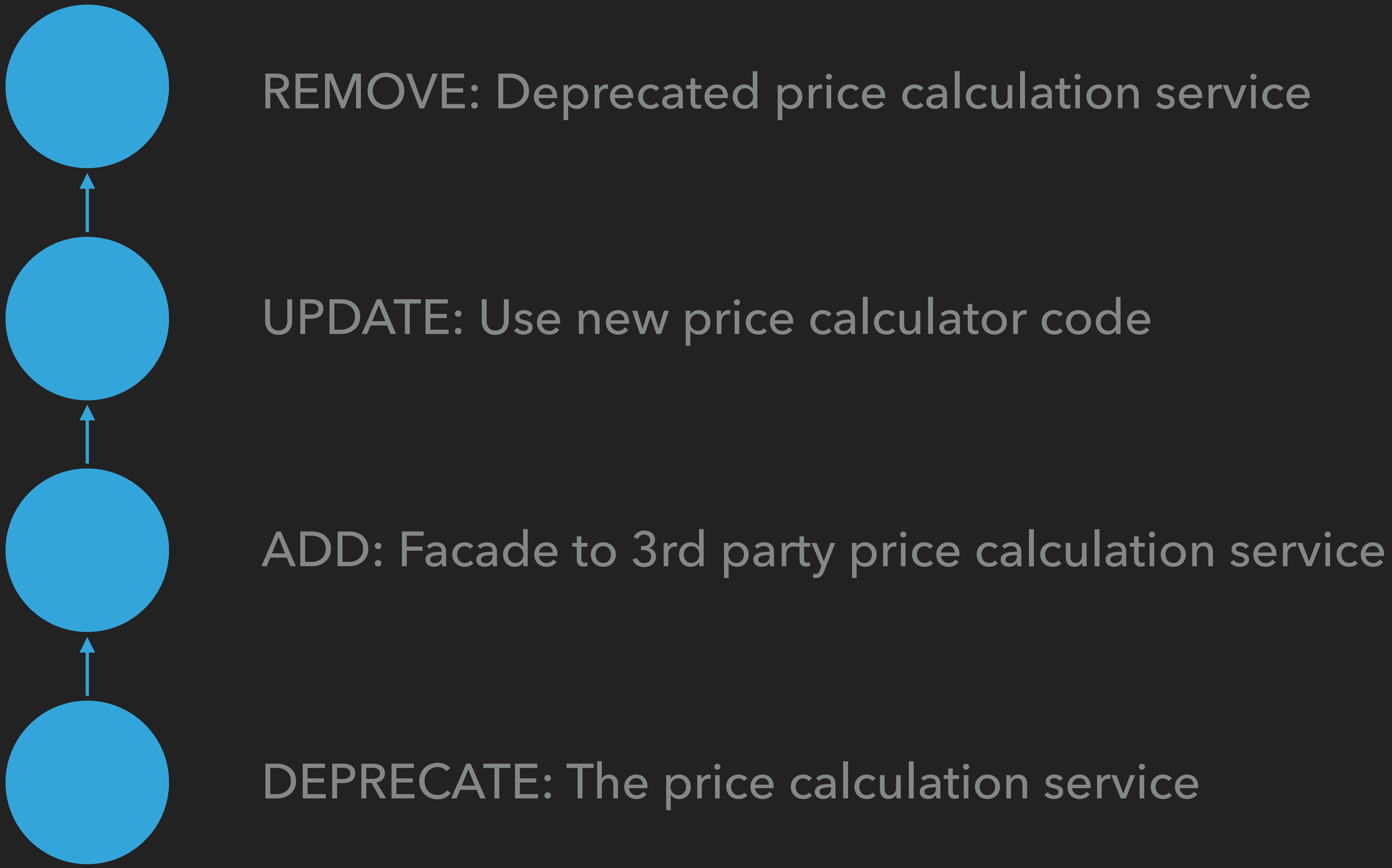


UPDATE: Use new price calculator code

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
- Composer update in own commit

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

AGENDA

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

BRANCHING STRATEGIES

ESSENTIAL – CODE CAN ONLY BE DEPLOYED IF:

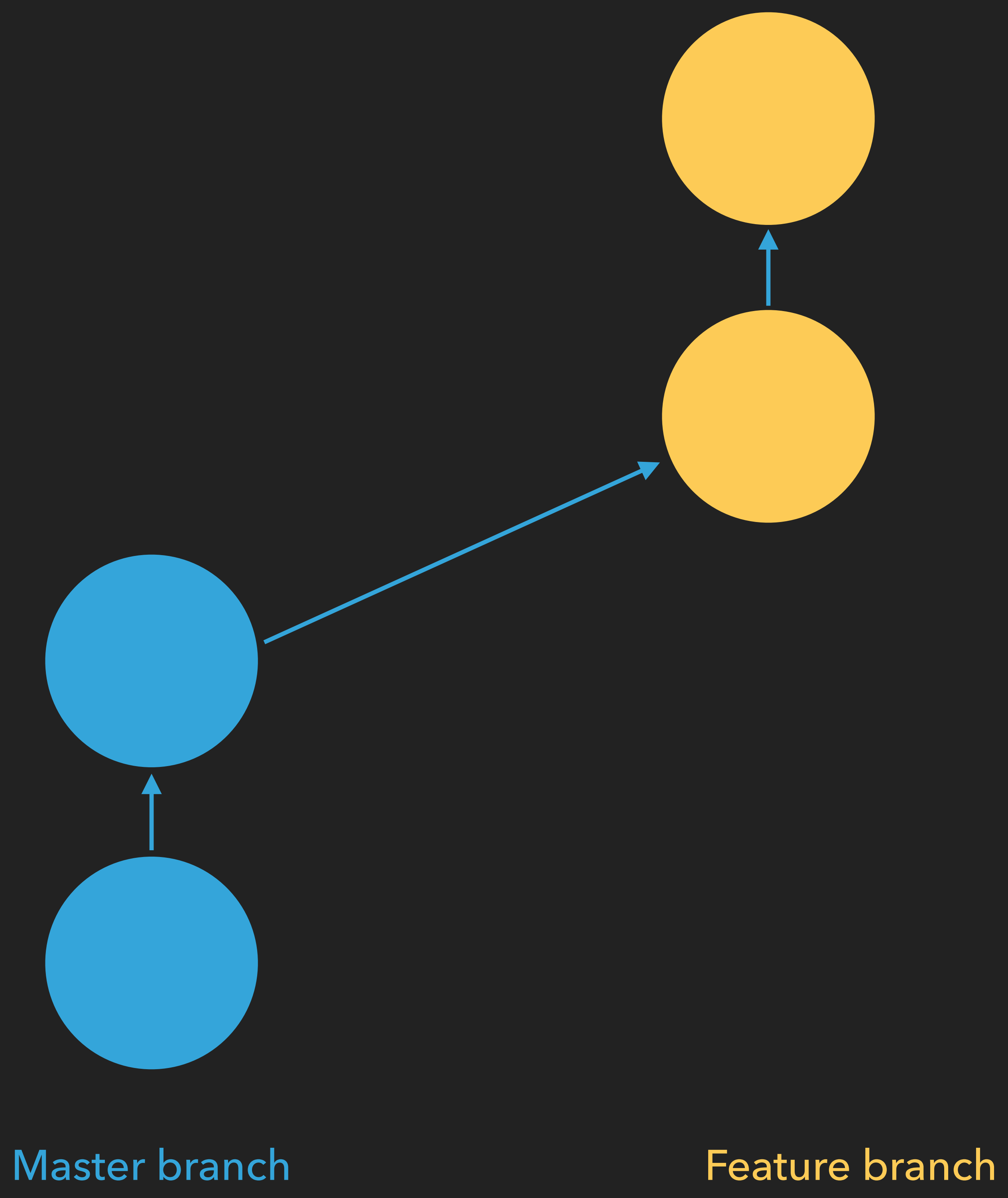
- CI passes
- Code review passes

IDEALLY - OTHERS ONLY DEVELOP WITH CODE THAT:

- CI passes
- Code review passes

PULL REQUESTS

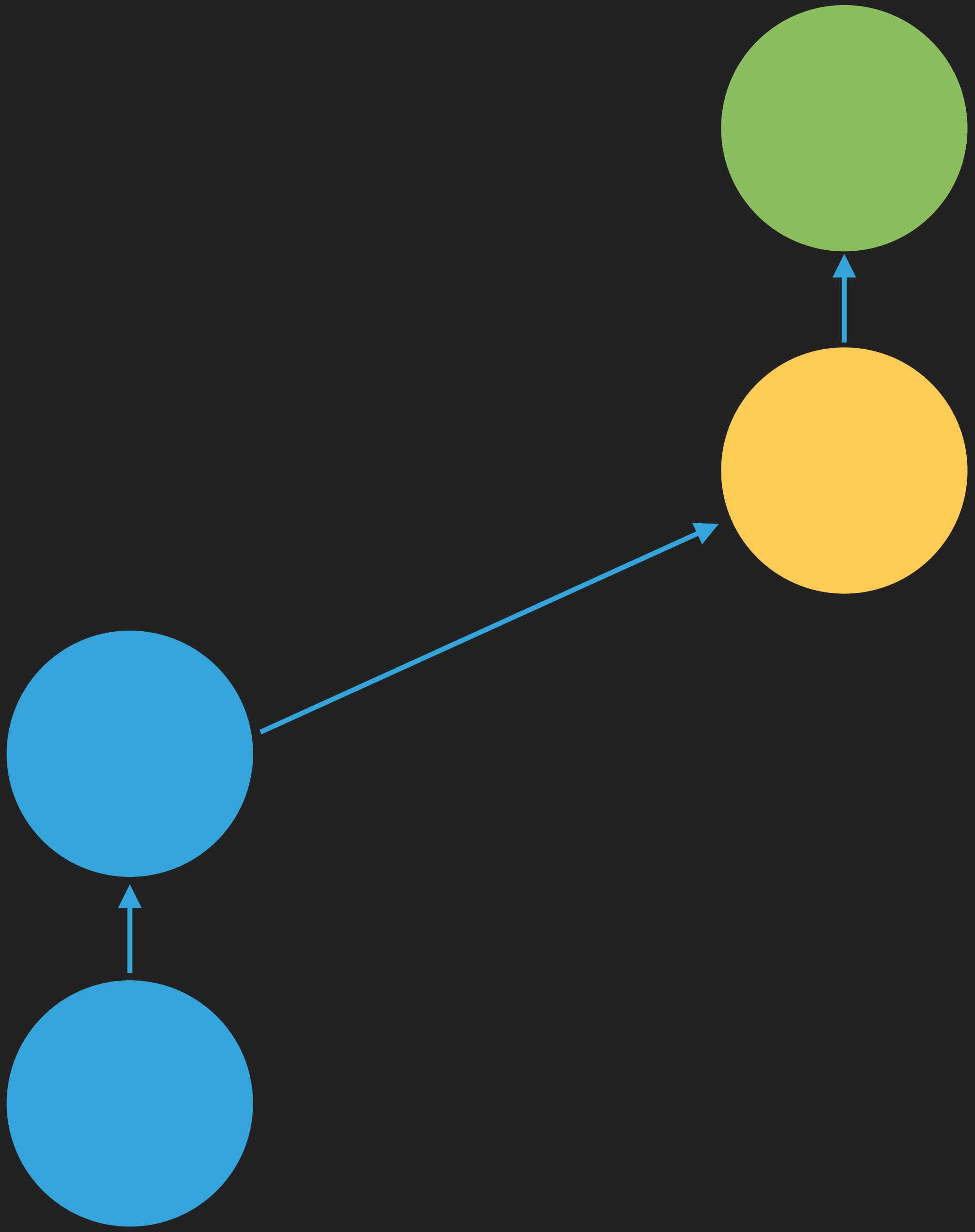
PULL REQUEST METHOD



Master branch

Feature branch

PULL REQUEST METHOD



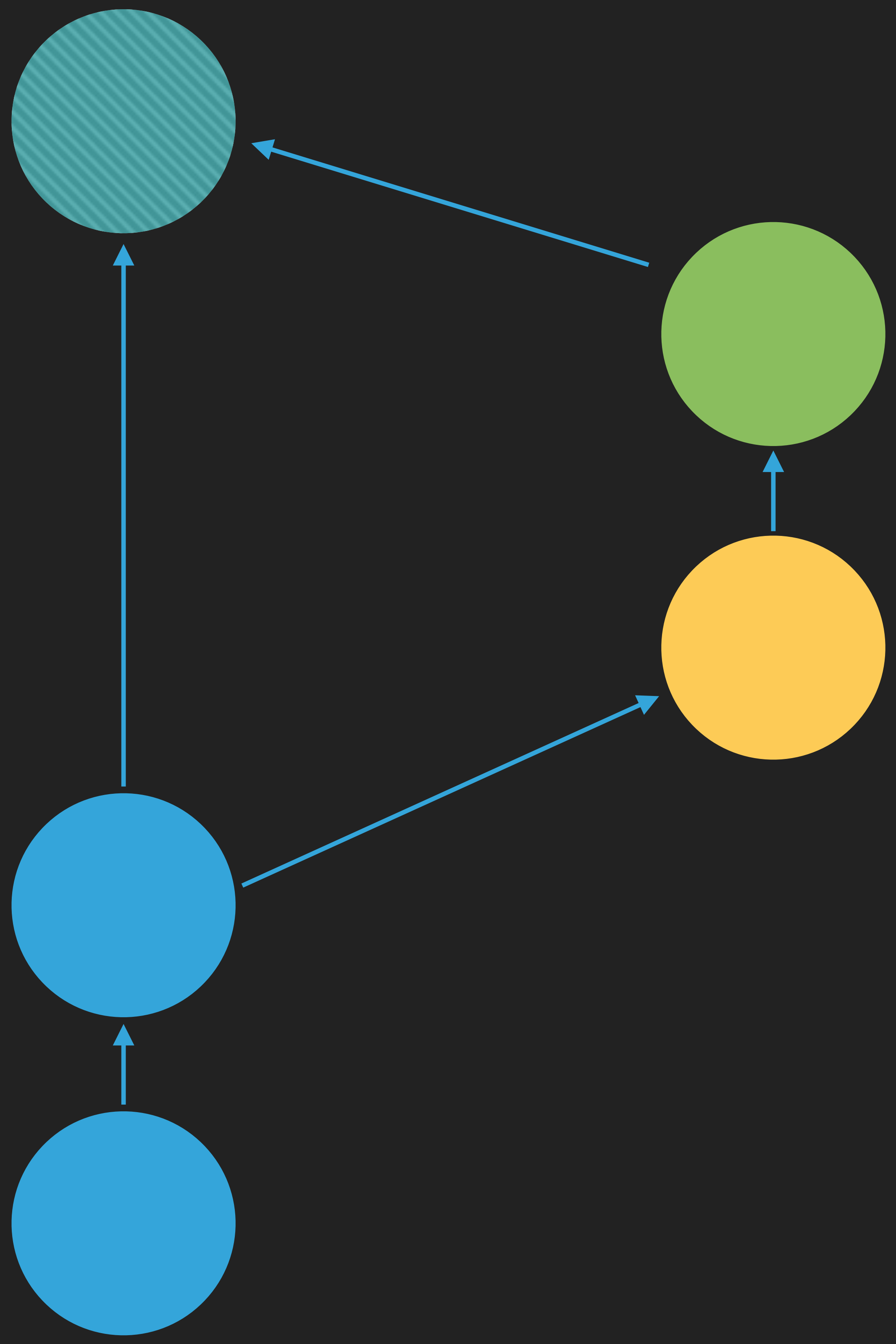
CI pass

Code review sign off

Master branch

Pull Request (PR) based on Feature branch

PULL REQUEST METHOD

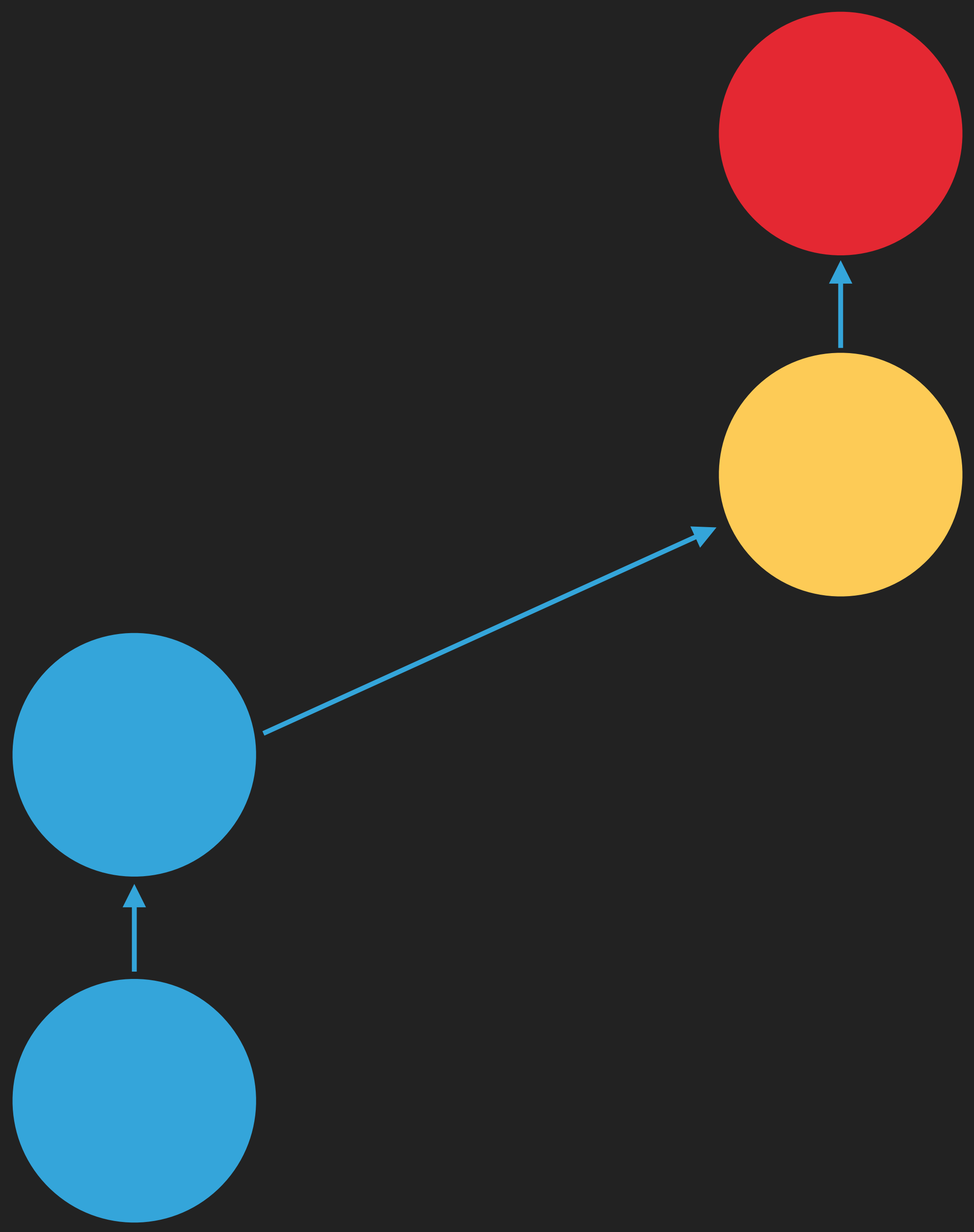


CI pass

Code review sign off

Master branch

PULL REQUEST METHOD



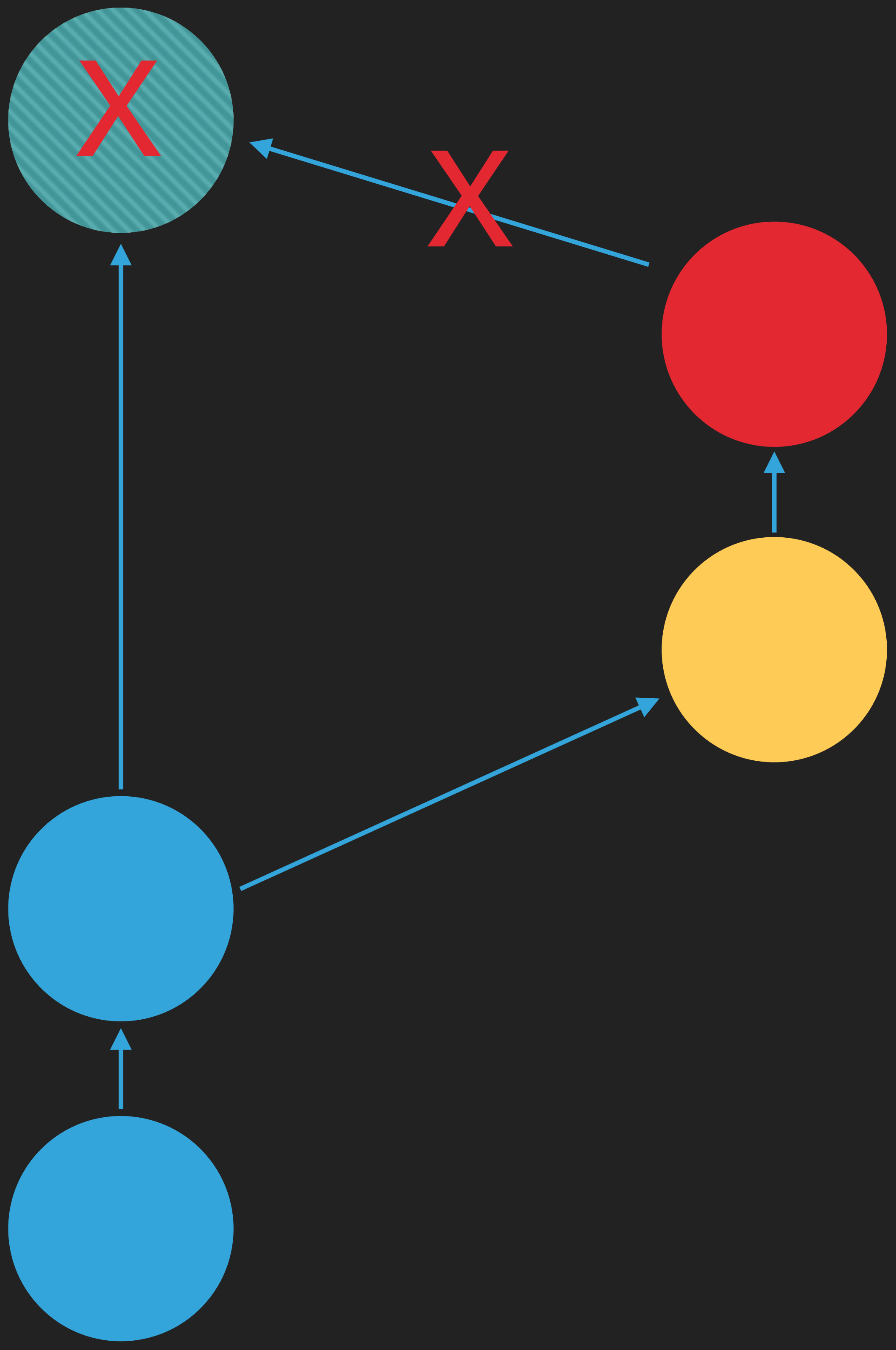
CI fails

OR

Code review not signed off

Master branch

PULL REQUEST METHOD



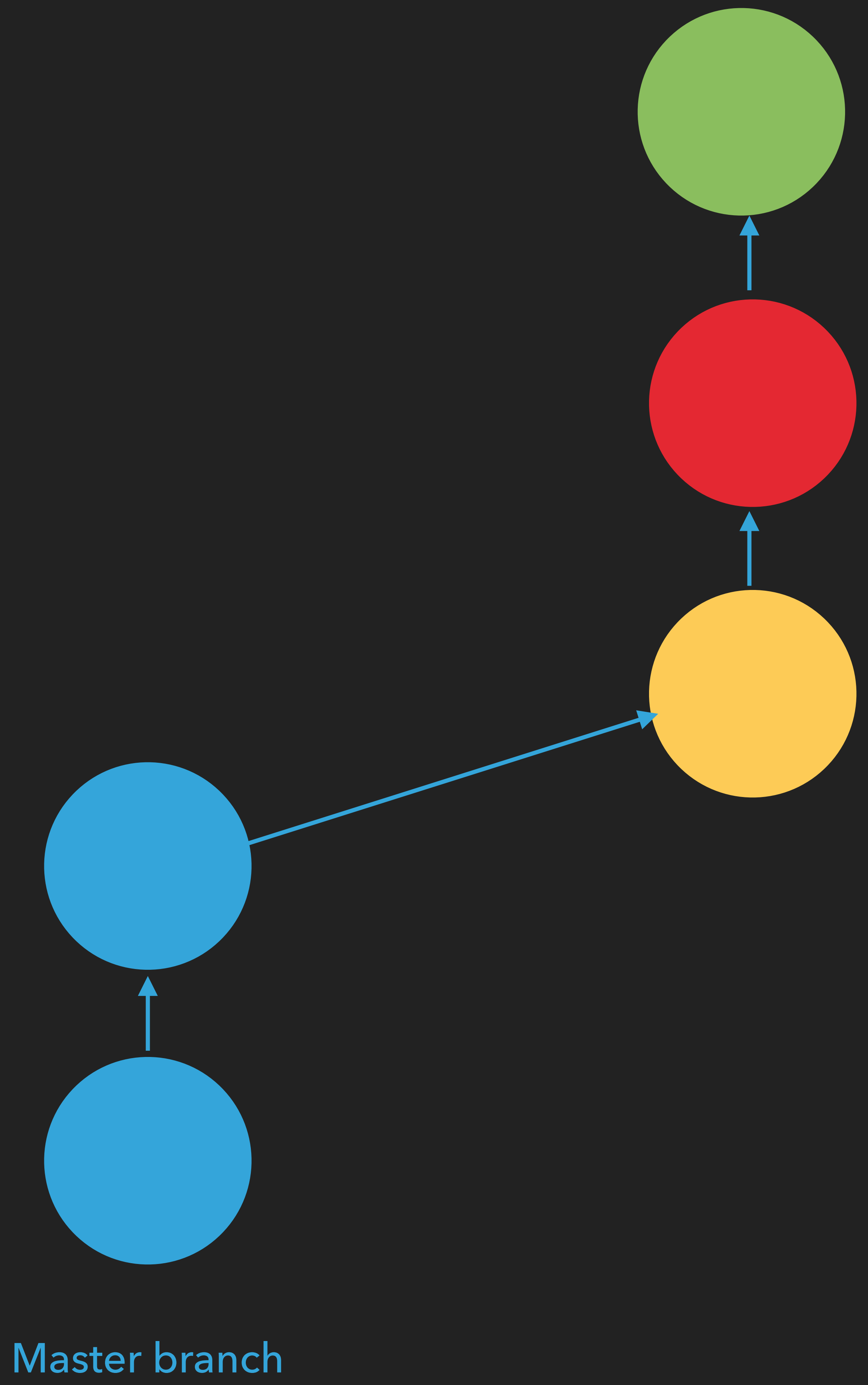
CI fails

OR

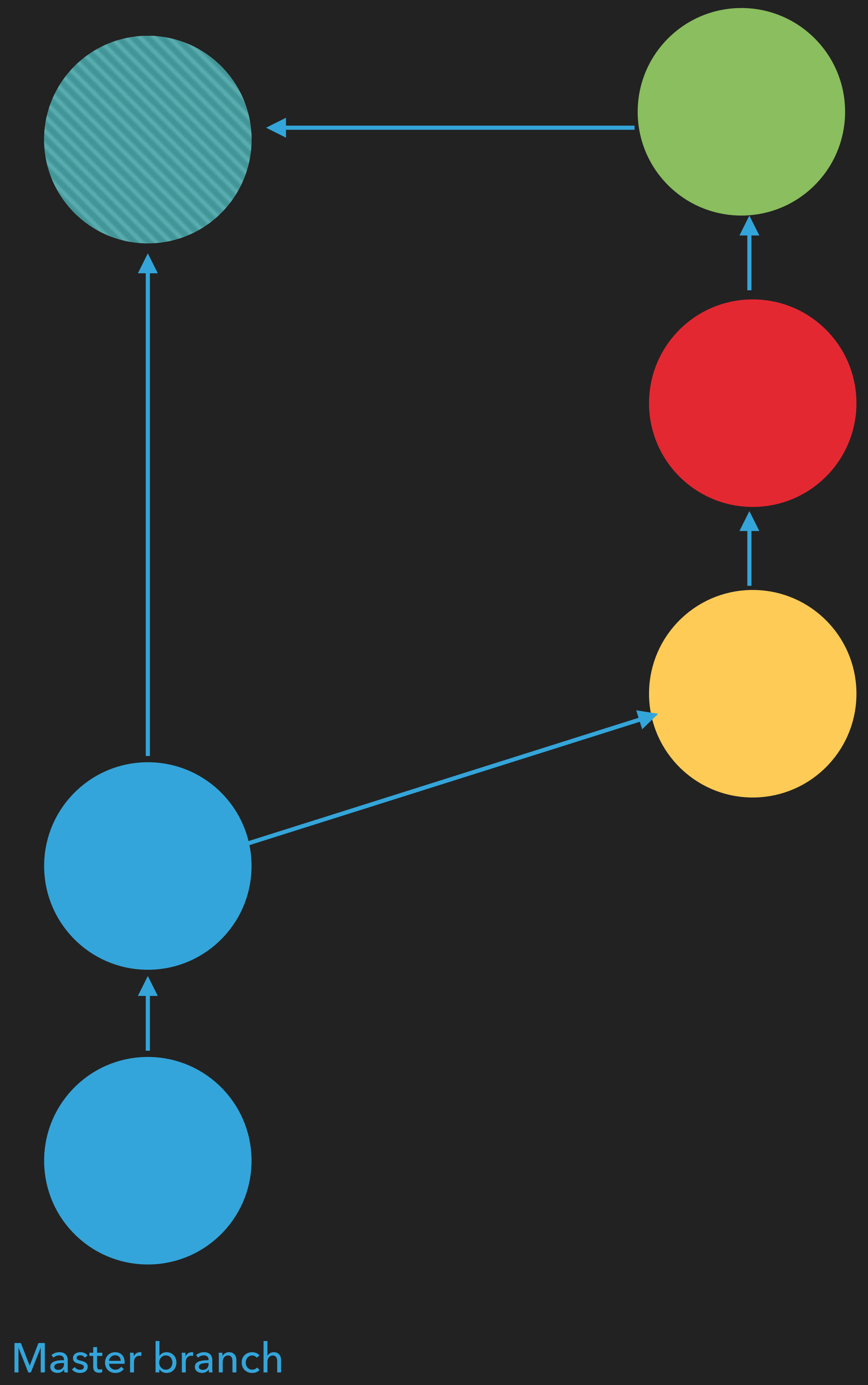
Code review not signed off

Master branch

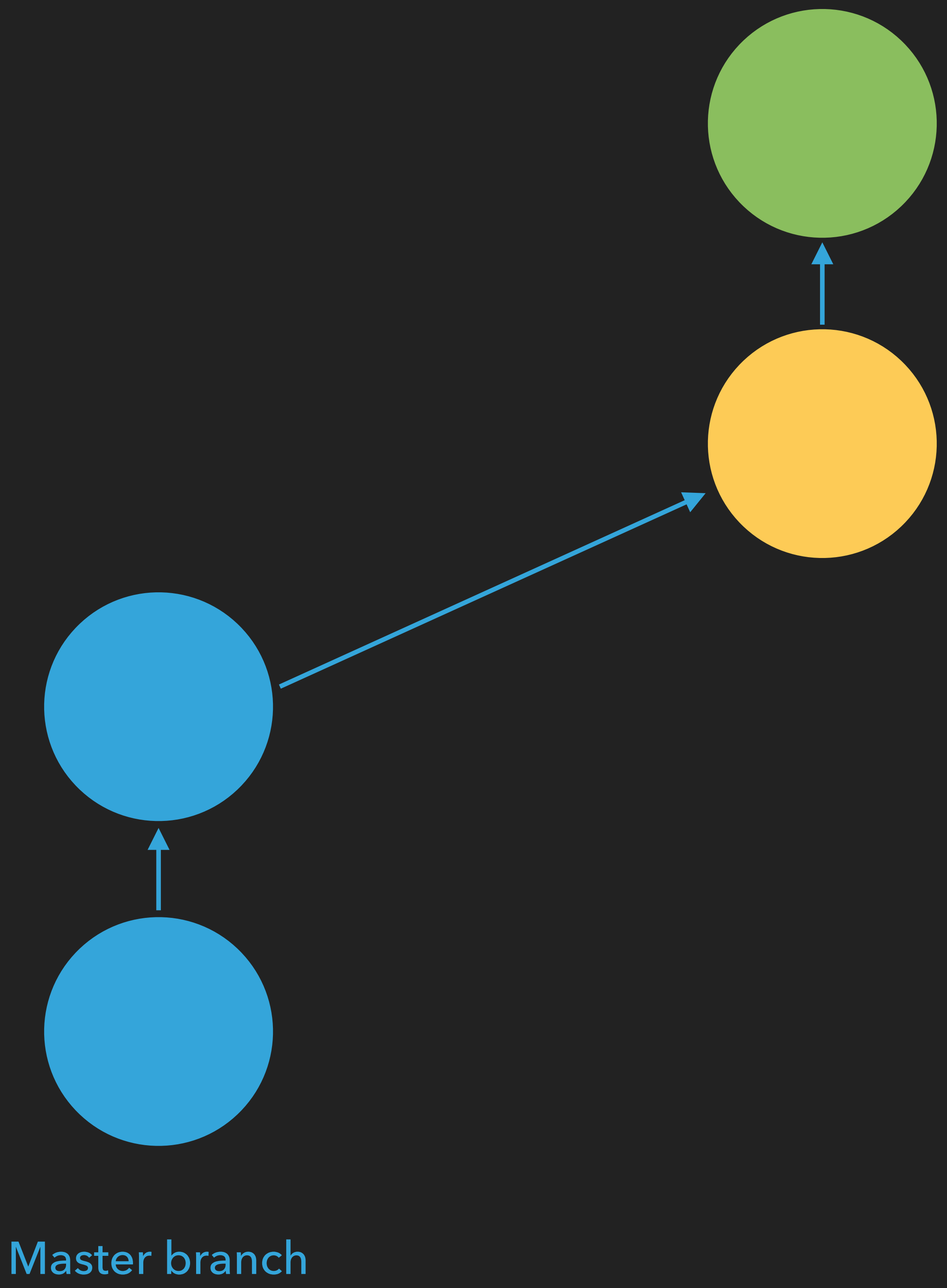
PULL REQUEST METHOD



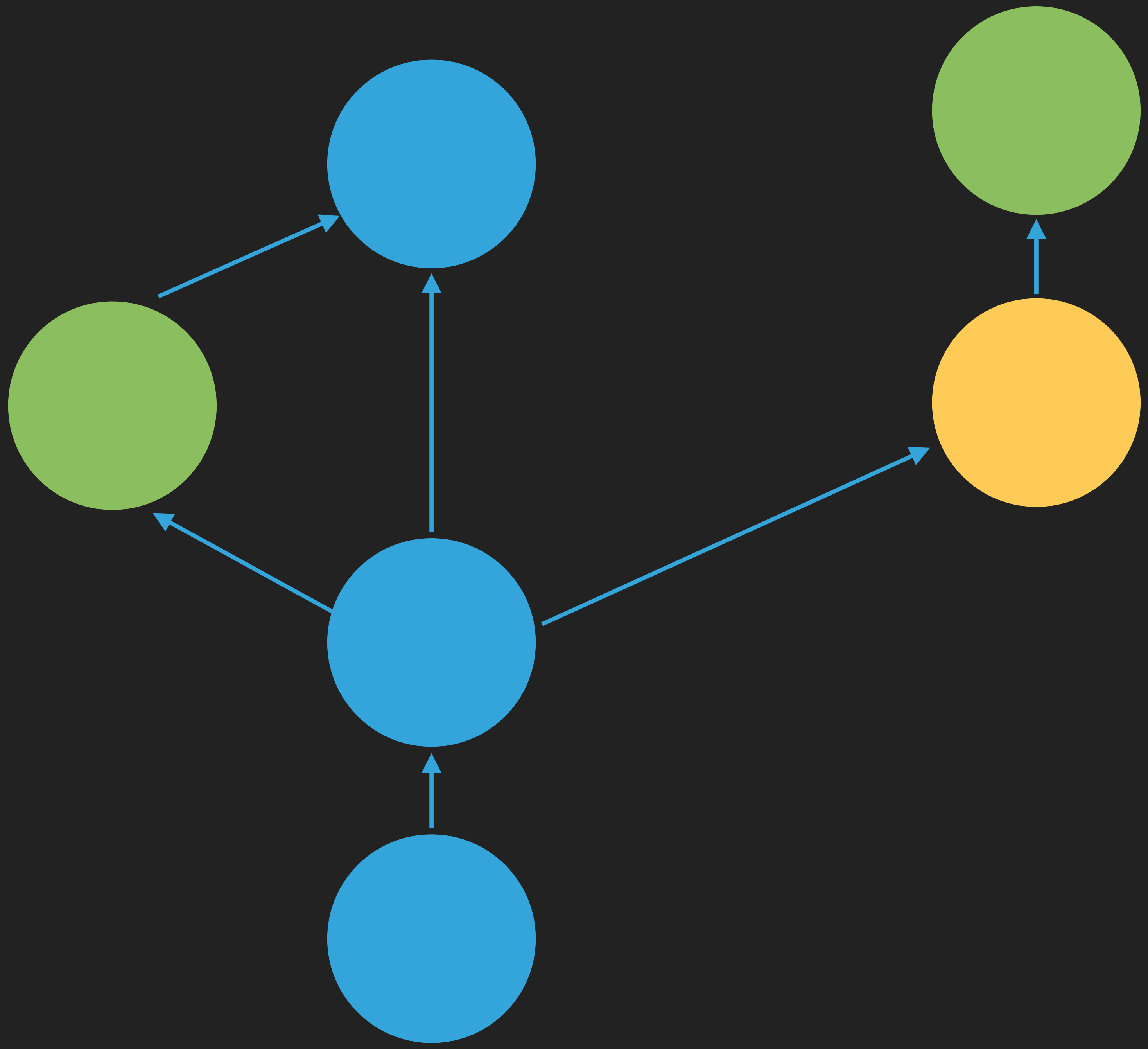
PULL REQUEST METHOD



PULL REQUEST METHOD

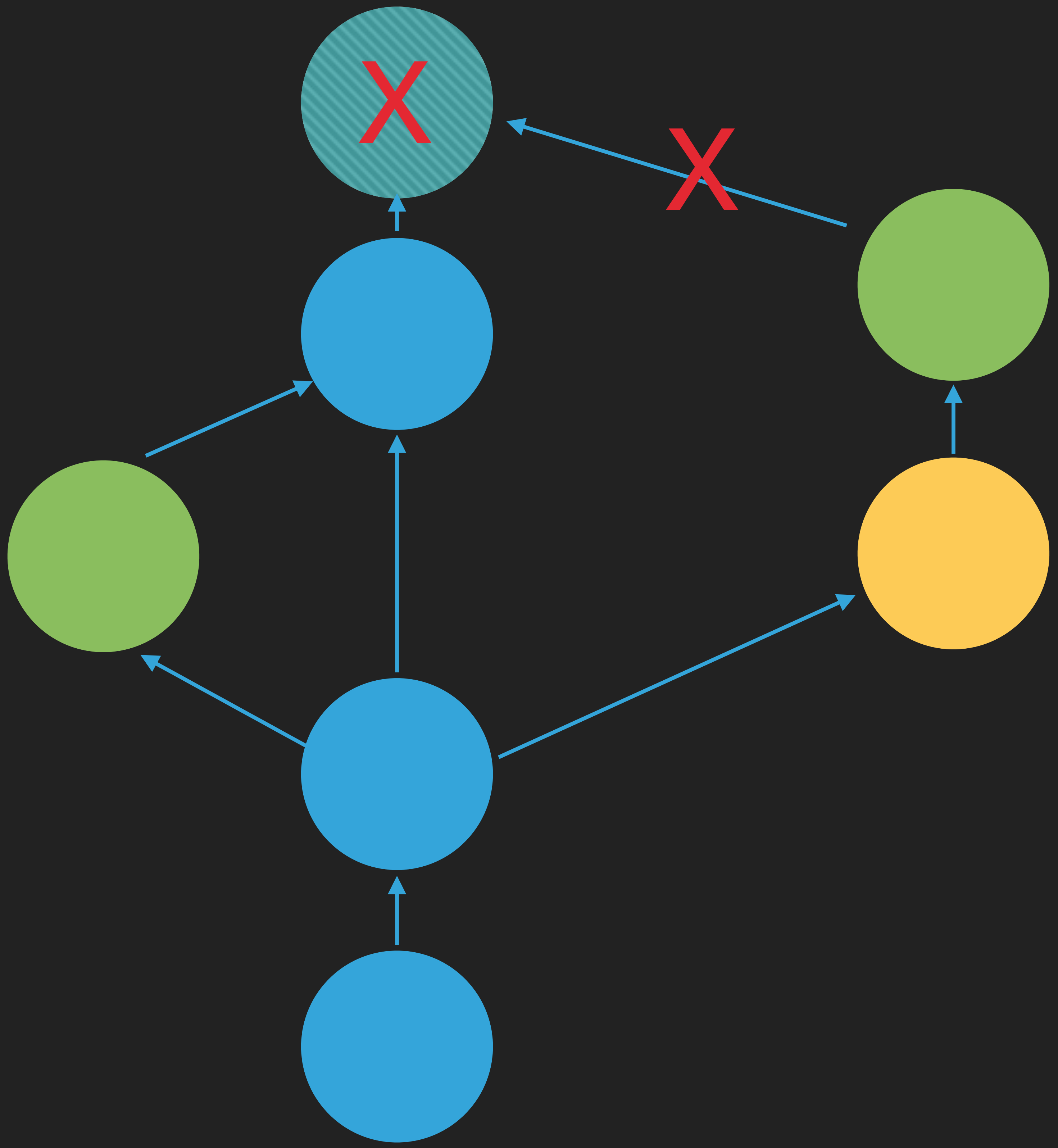


PULL REQUEST METHOD



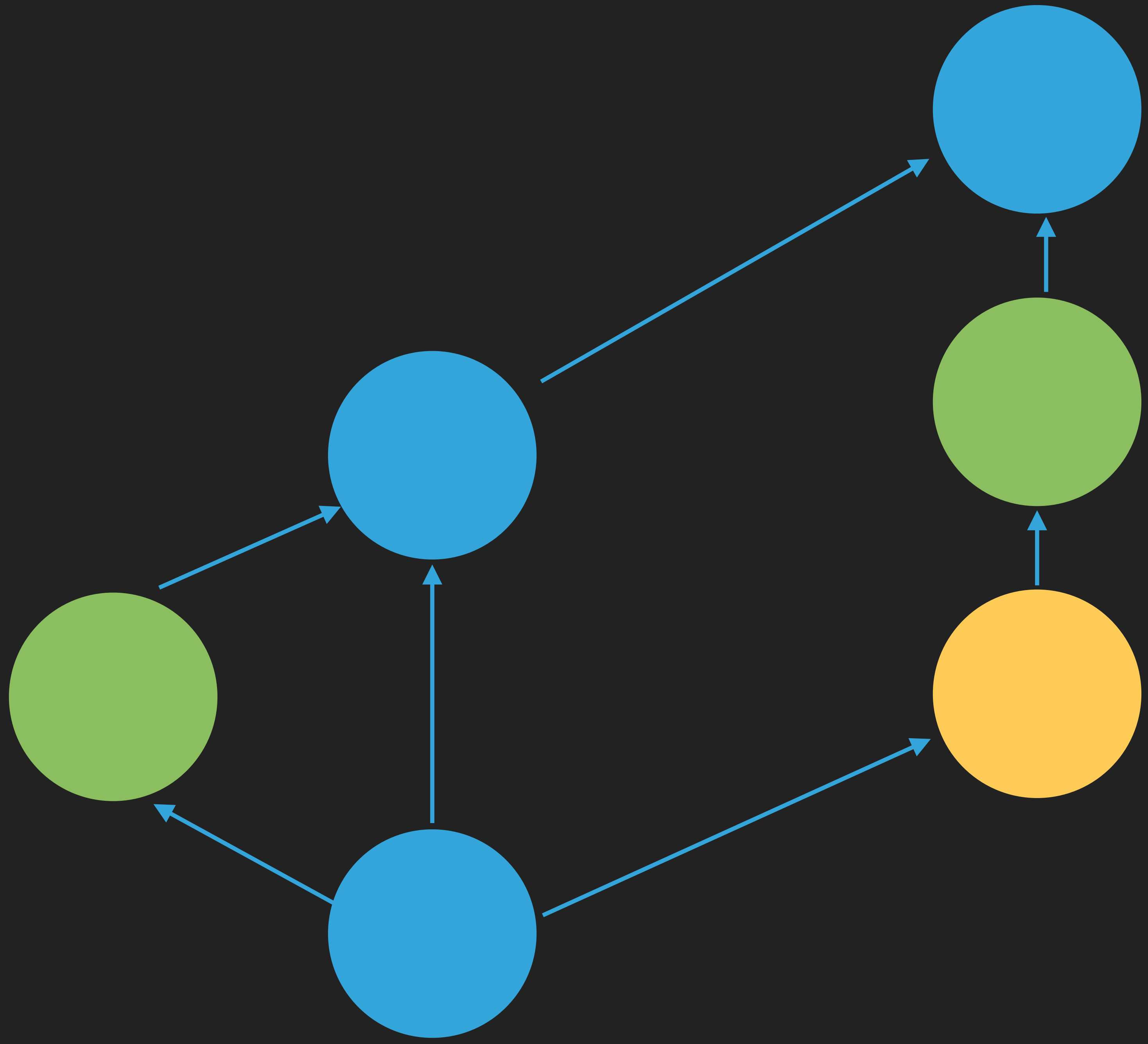
Master branch

PULL REQUEST METHOD



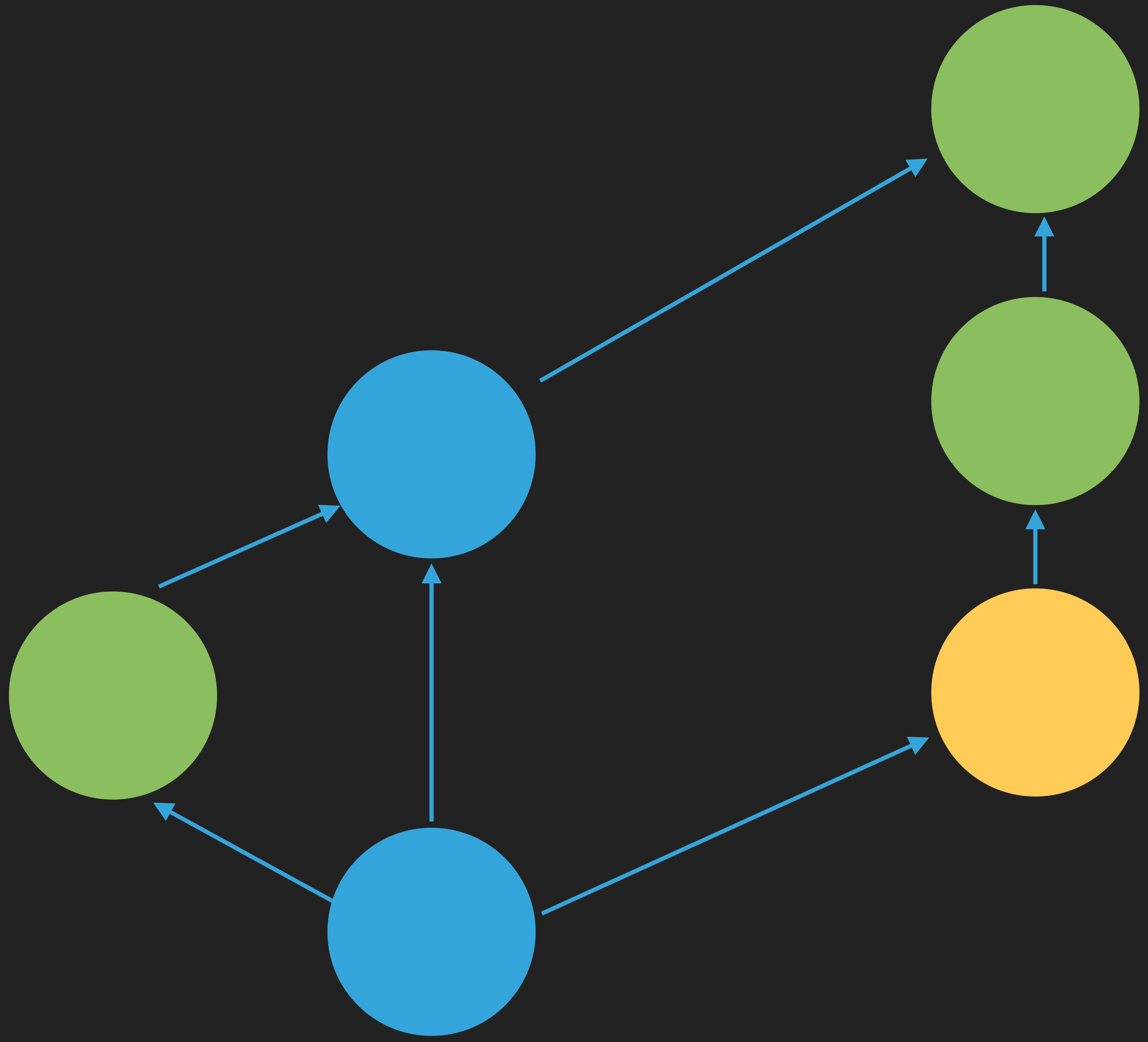
Master branch

PULL REQUEST METHOD



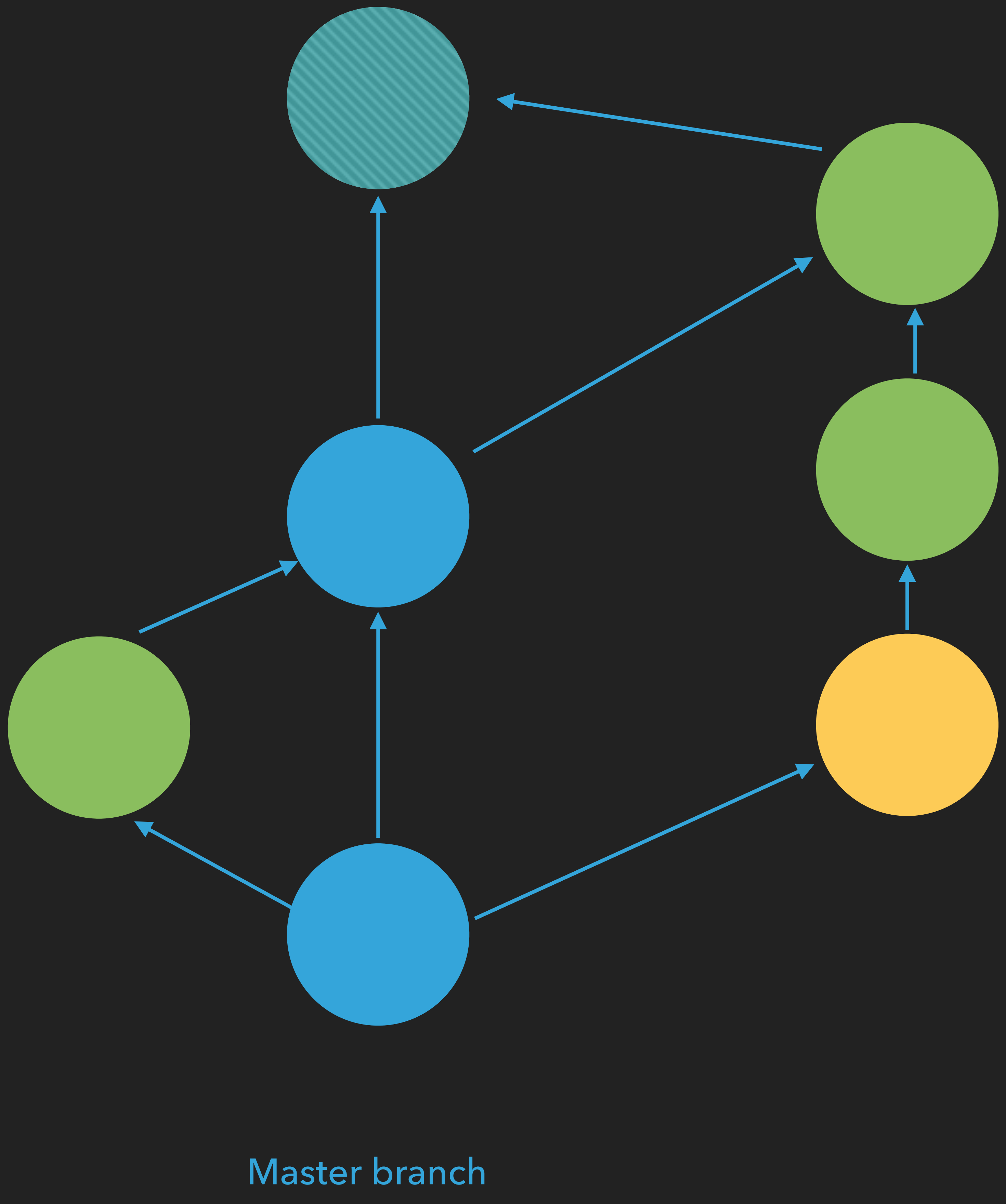
Master branch

PULL REQUEST METHOD



Master branch

PULL REQUEST METHOD

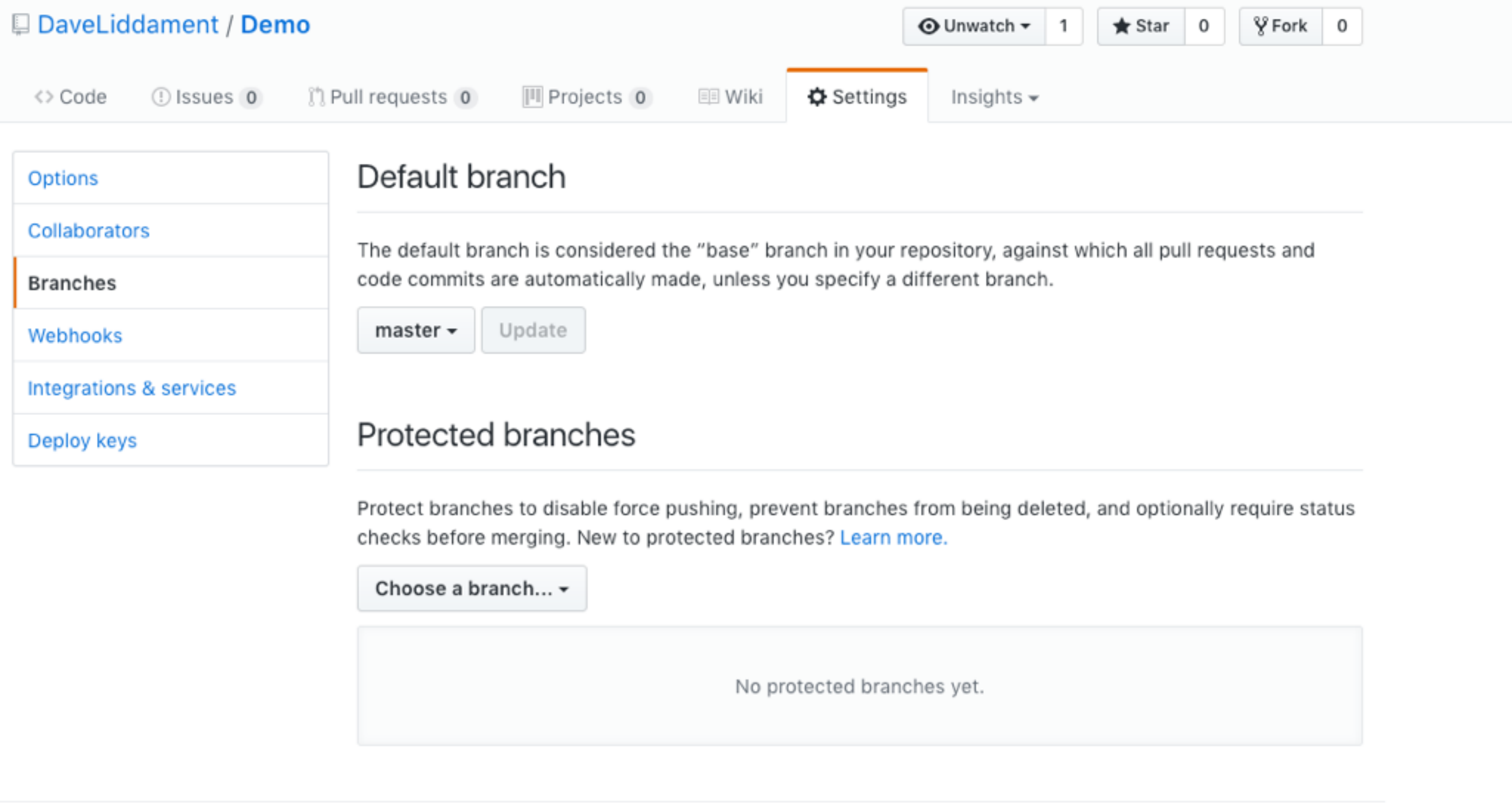


LET’S SET THIS UP

LET'S SET THIS UP

PS It's really easy

SETUP GITHUB



Default branch

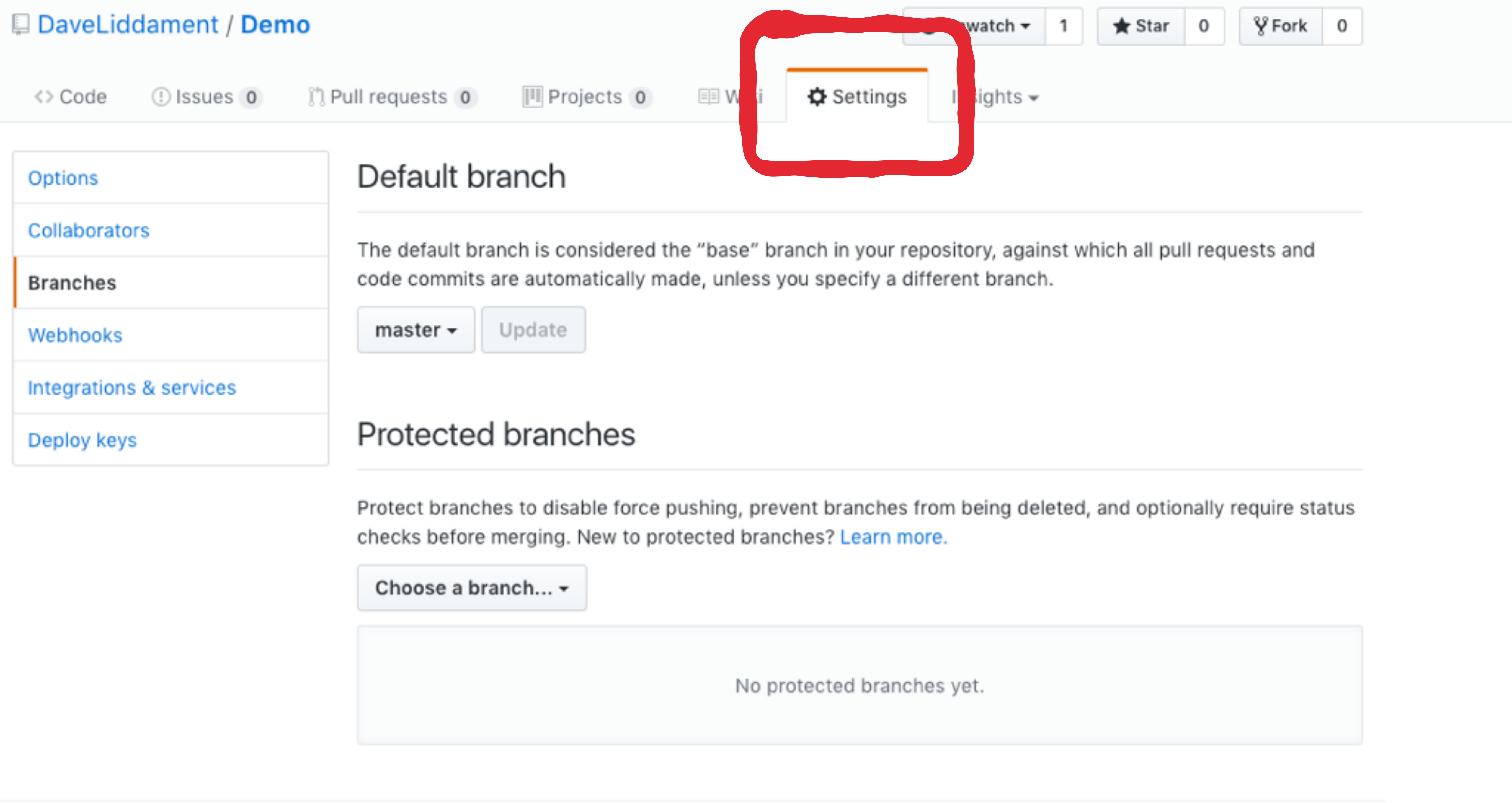
The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

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.

No protected branches yet.

SETUP GITHUB



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

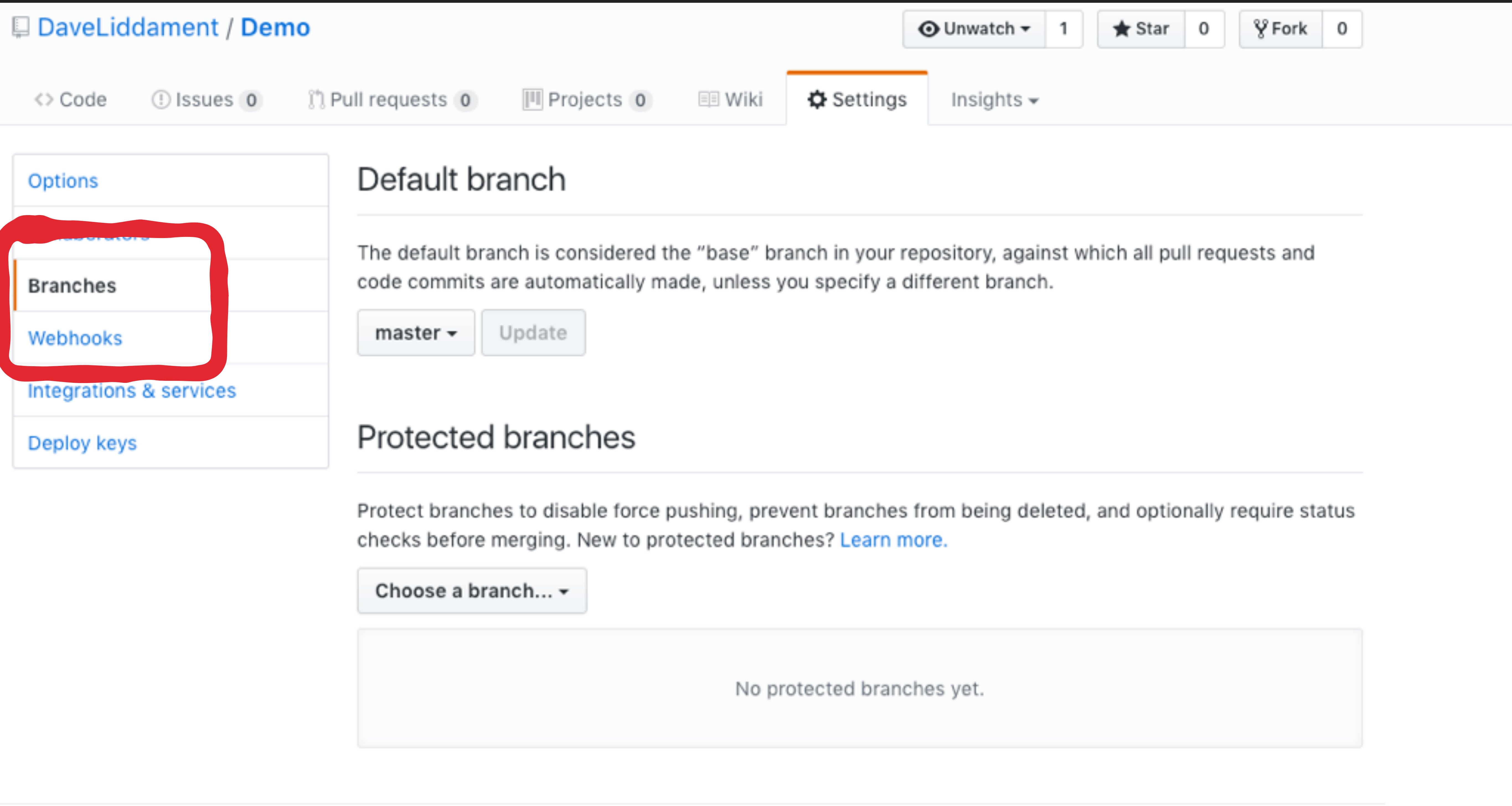
Protected branches

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



Default branch

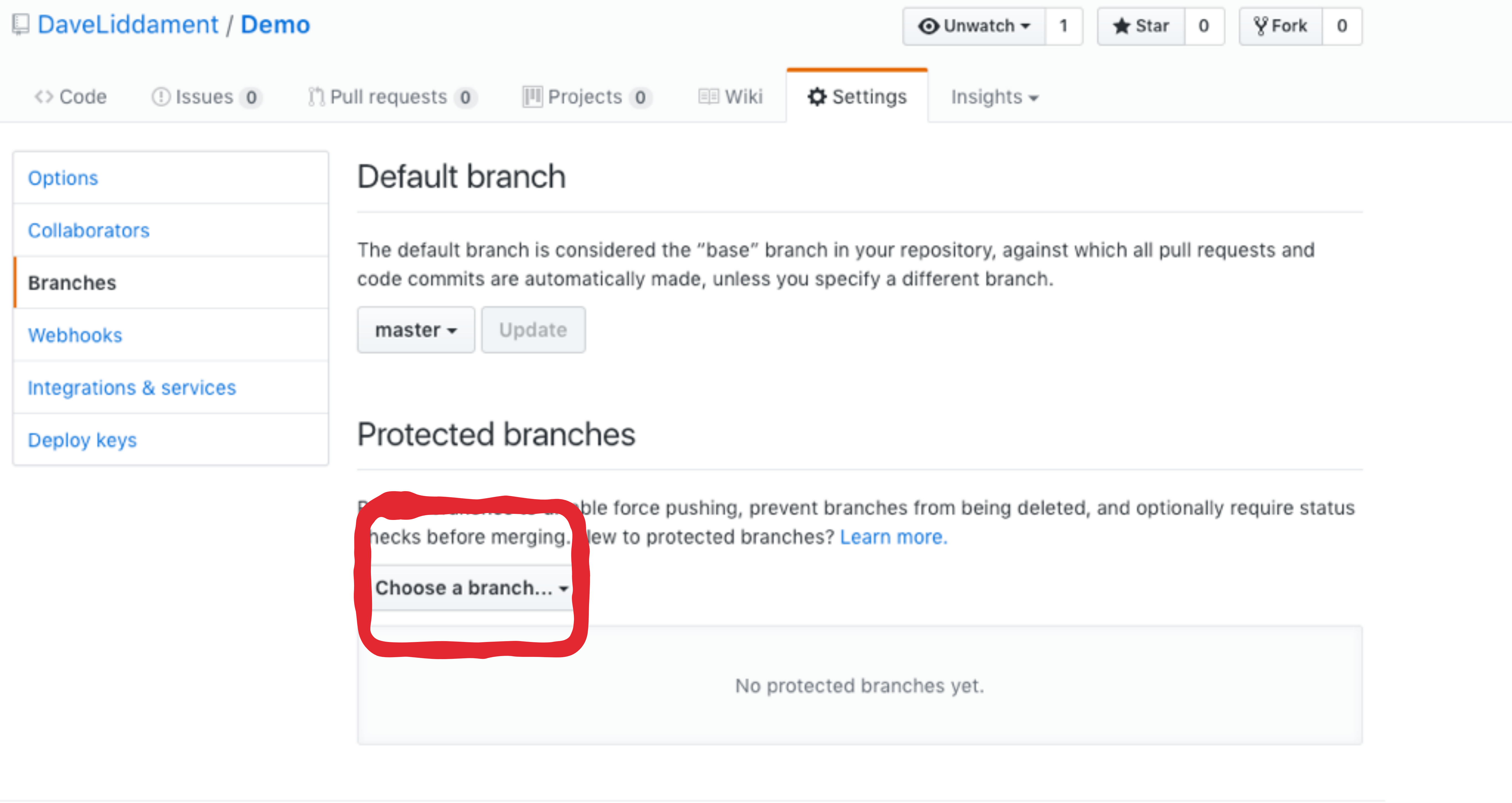
The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

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.

No protected branches yet.

SETUP GITHUB



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

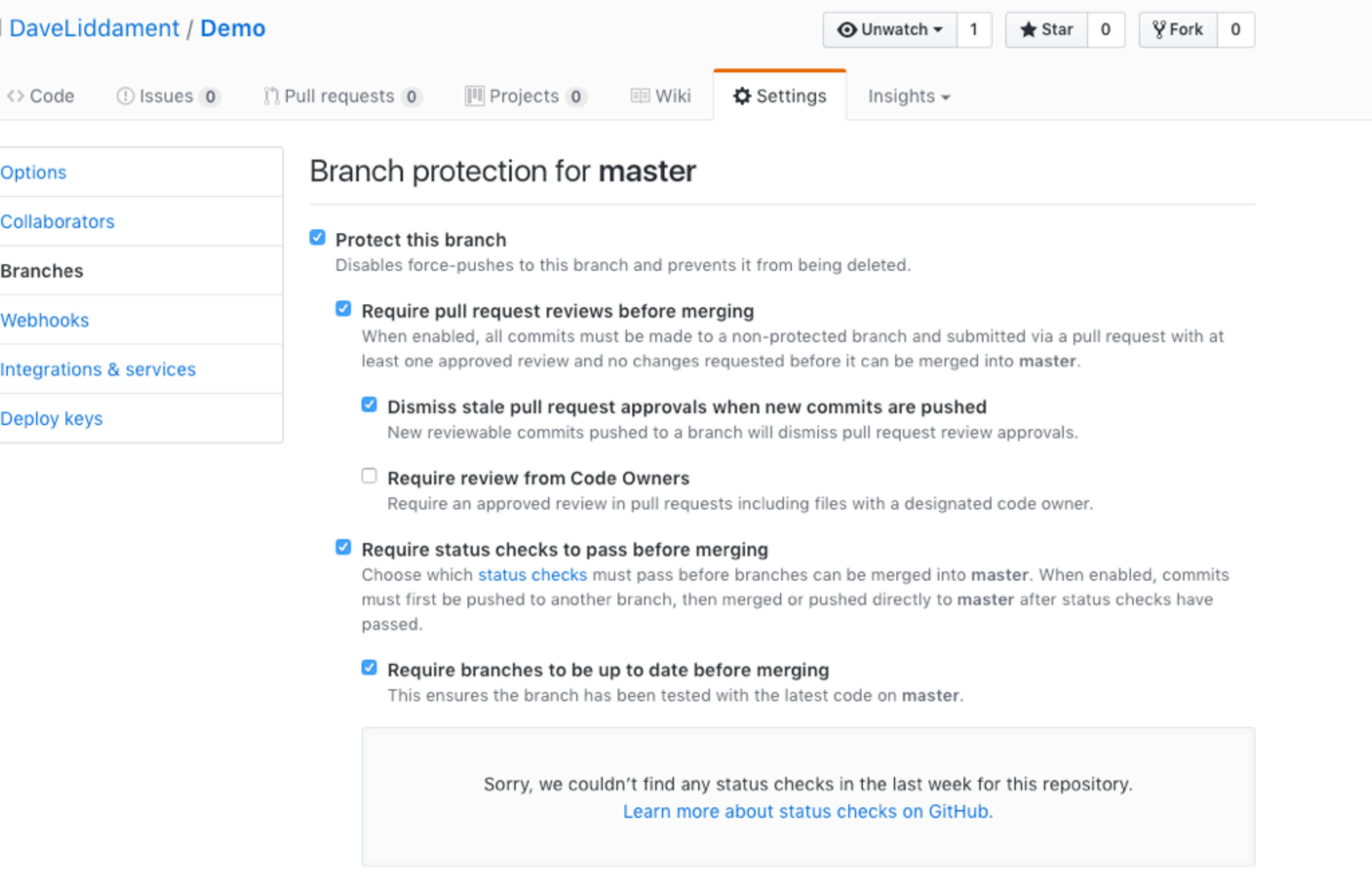
Protected branches

P[illegible]ble force pushing, prevent branches from being deleted, and optionally require status hecks before merging. lew to protected branches? Learn more.

Choose a branch... ▾

No protected branches yet.

SETUP GITHUB



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.

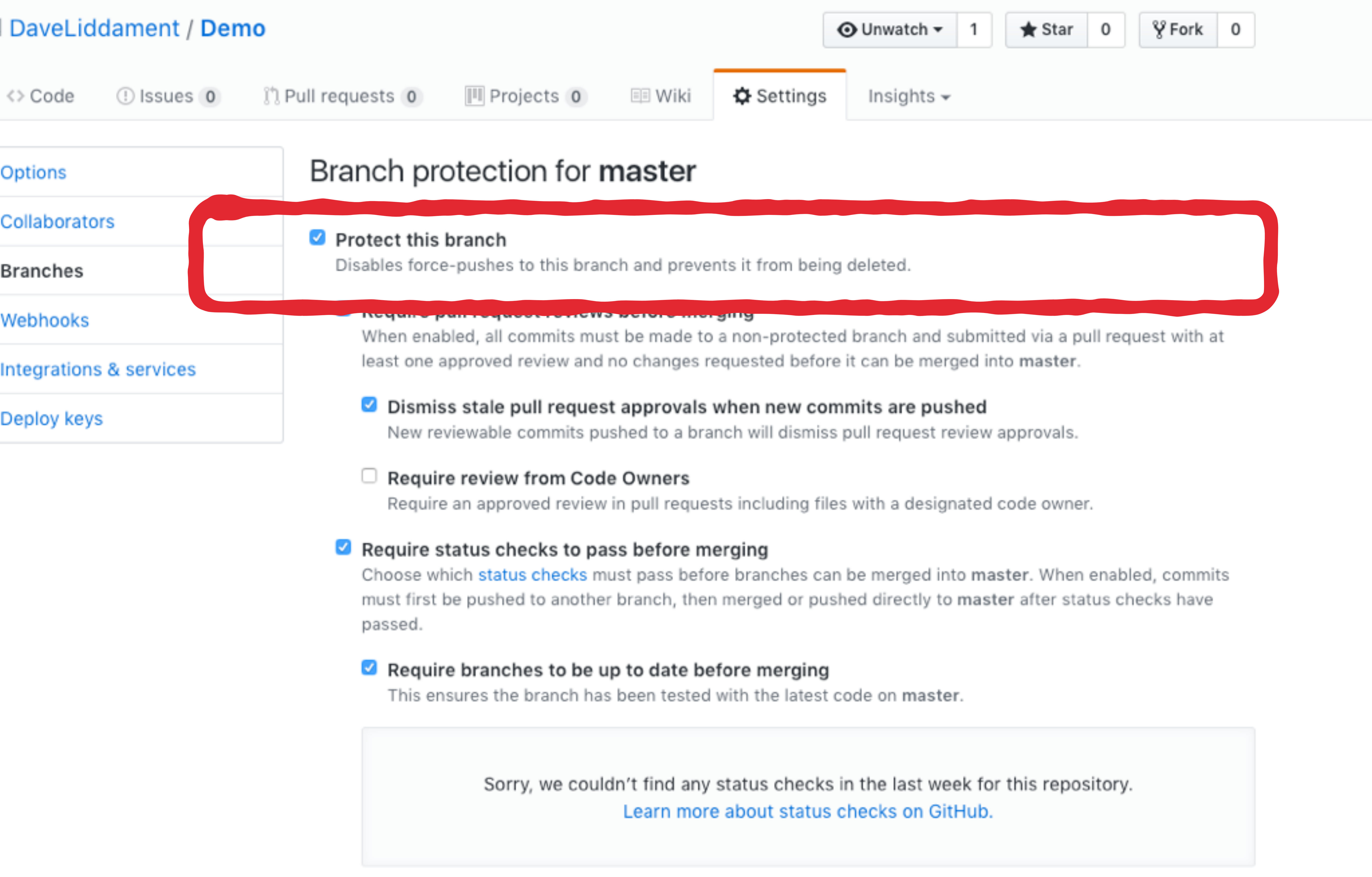
☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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



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.

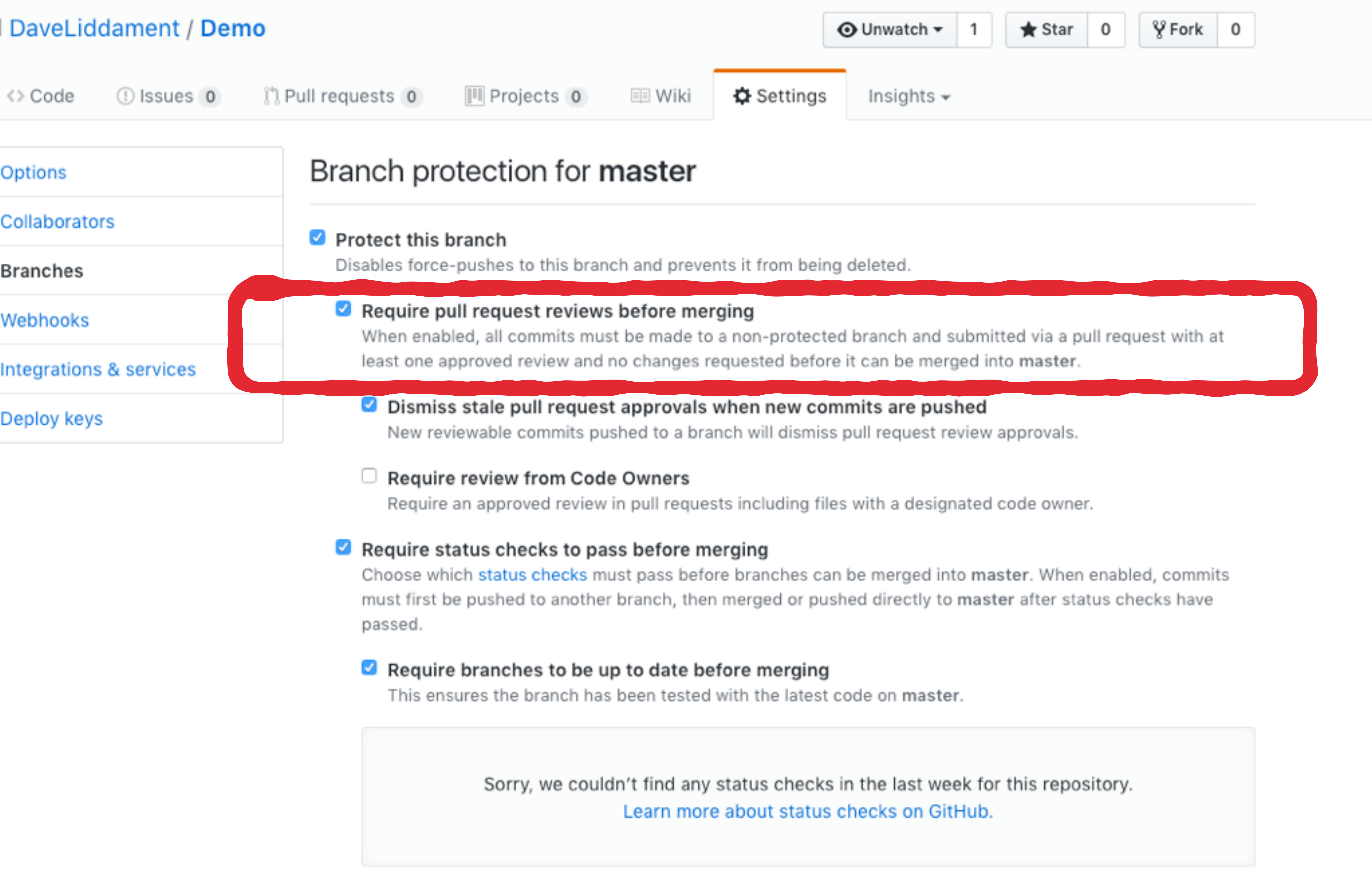
☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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



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.

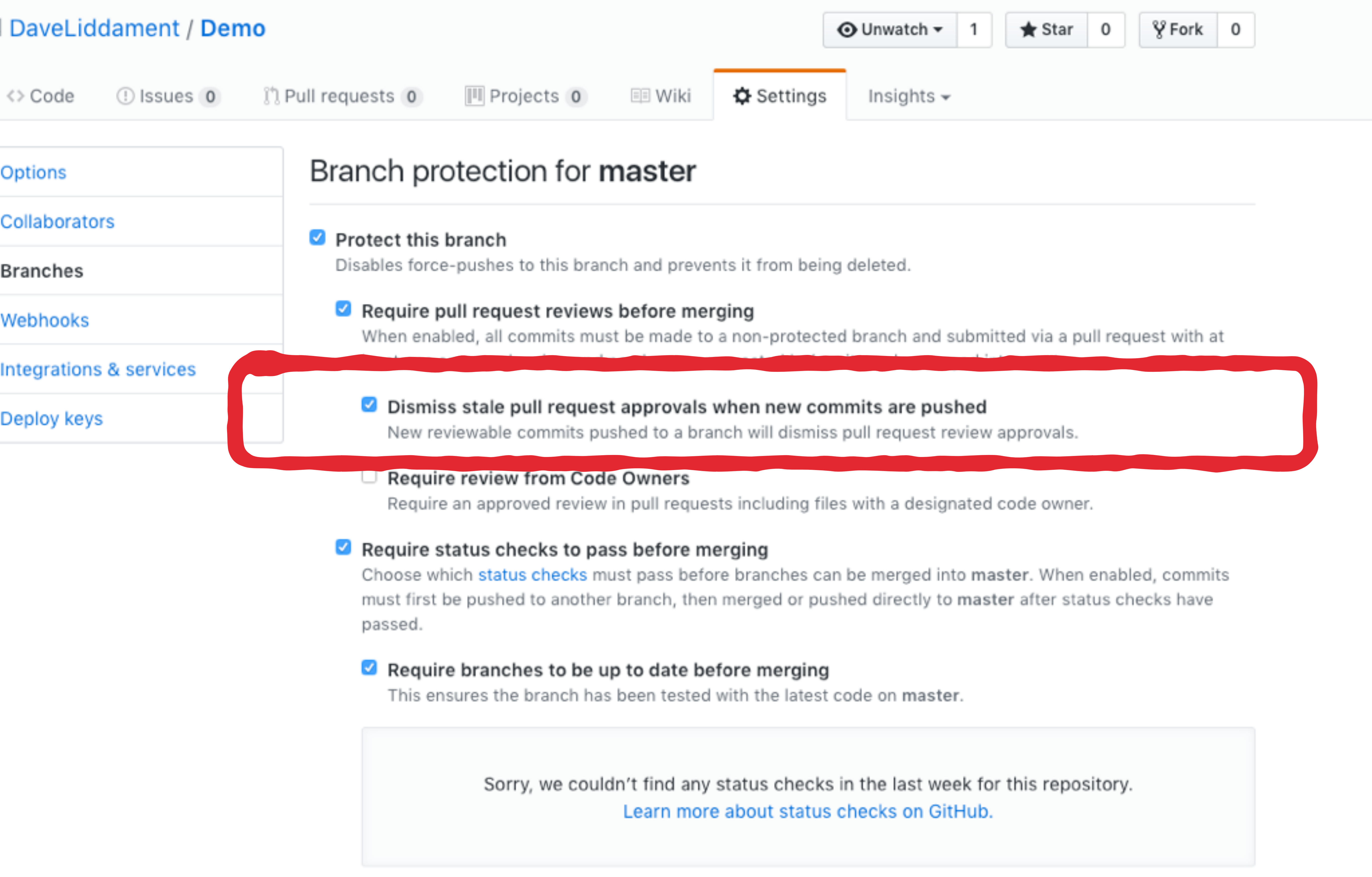
☐ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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



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.

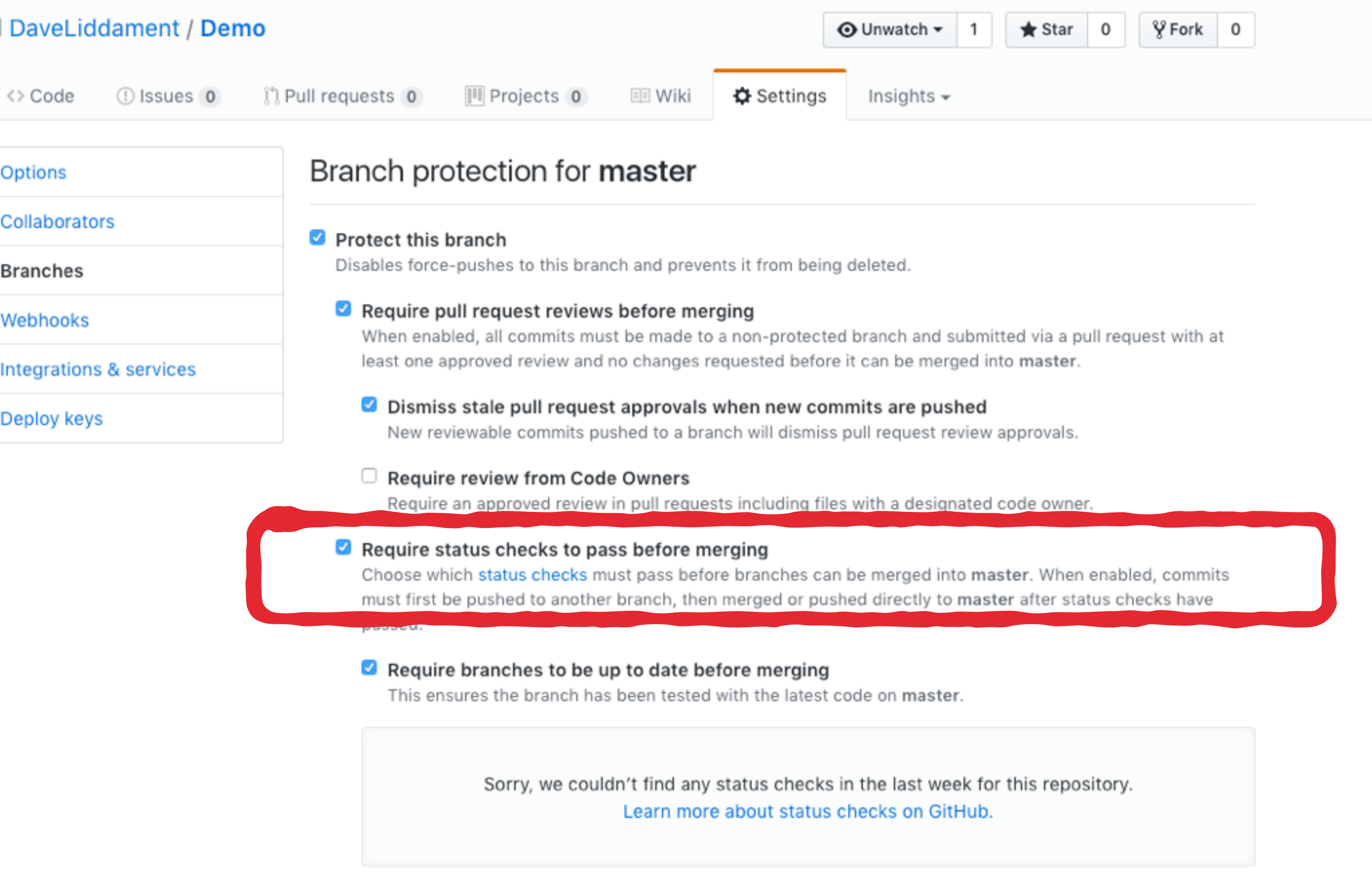
□ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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

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.

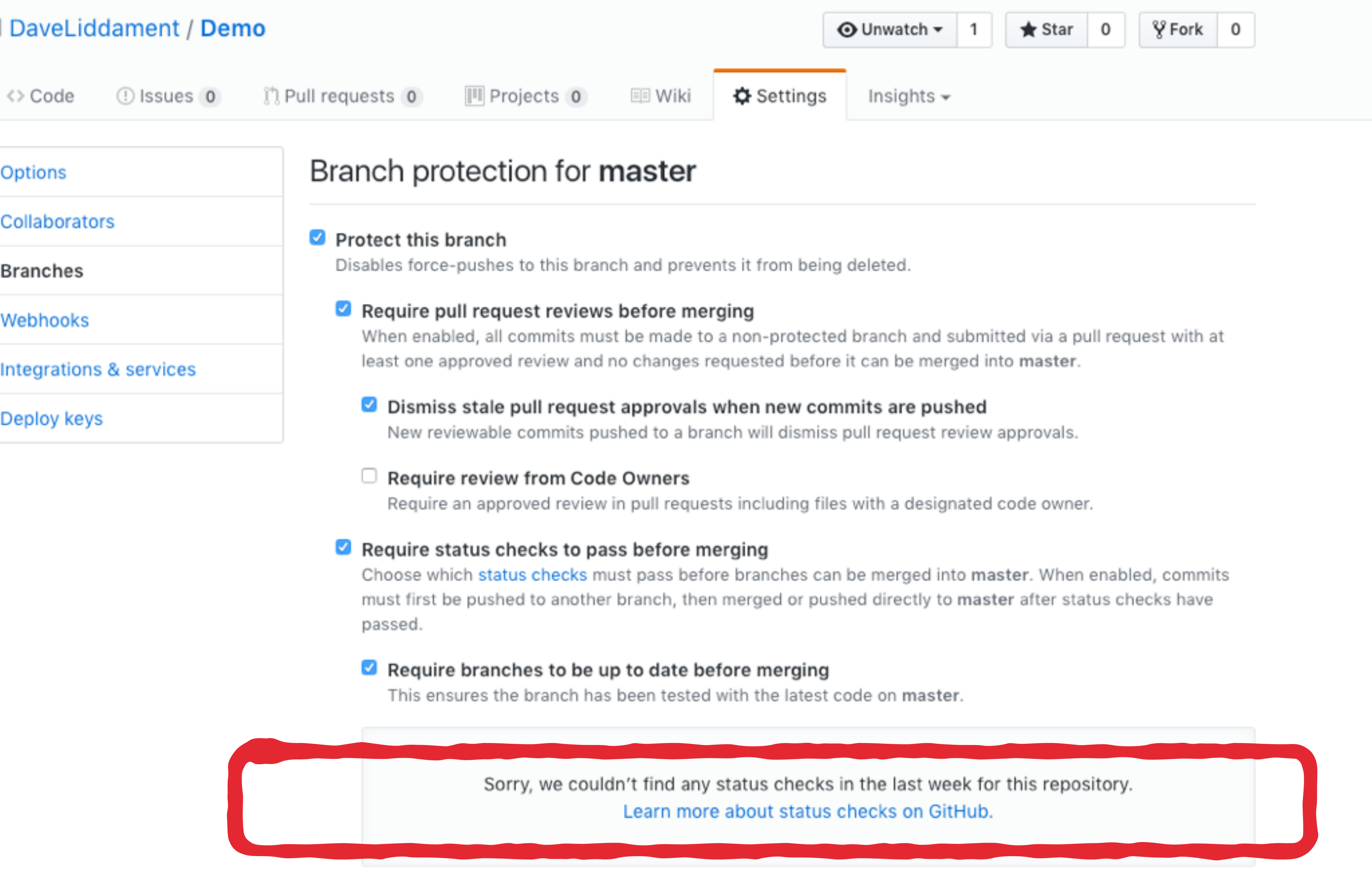
□ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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



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.

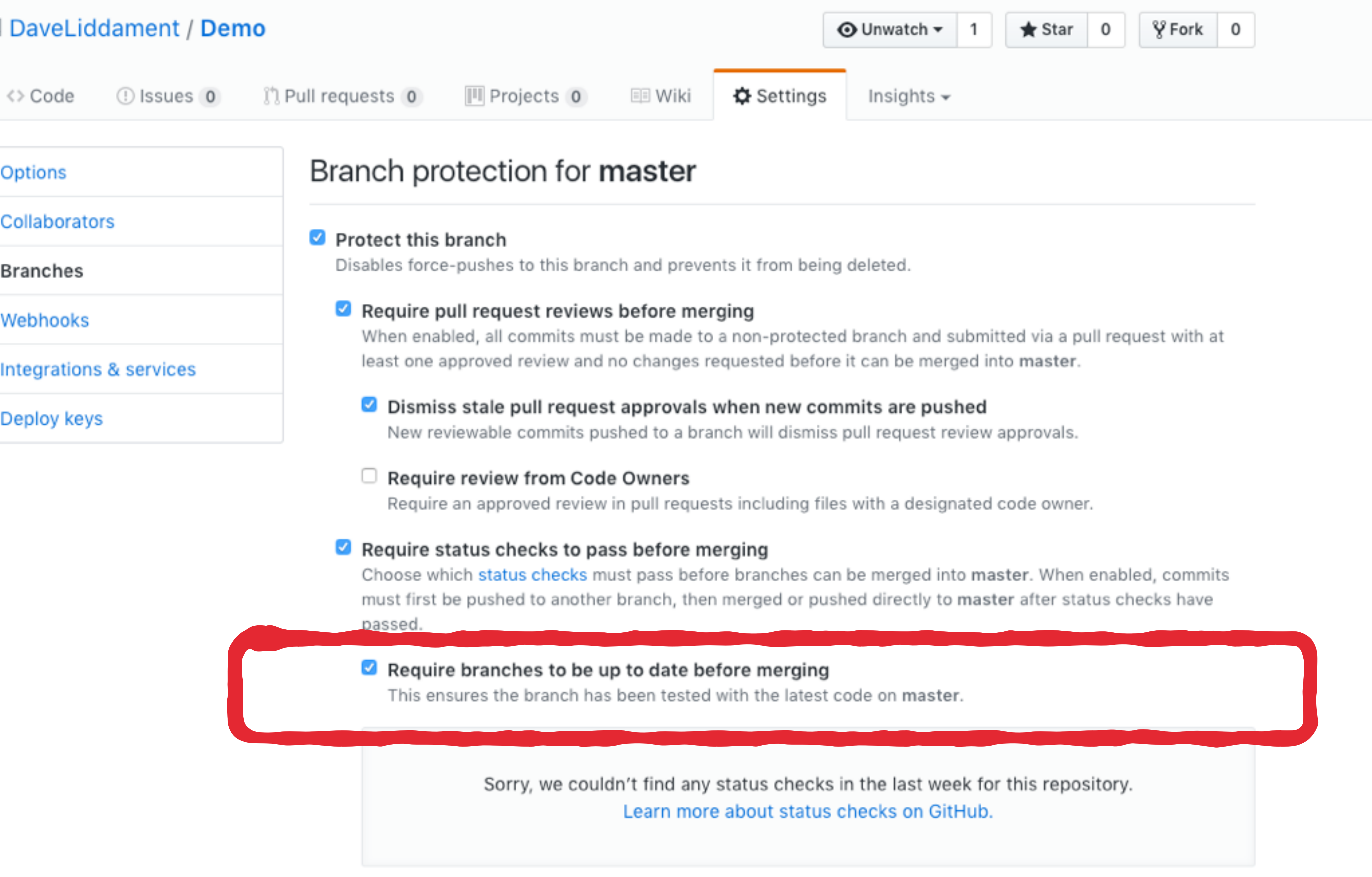
□ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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



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.

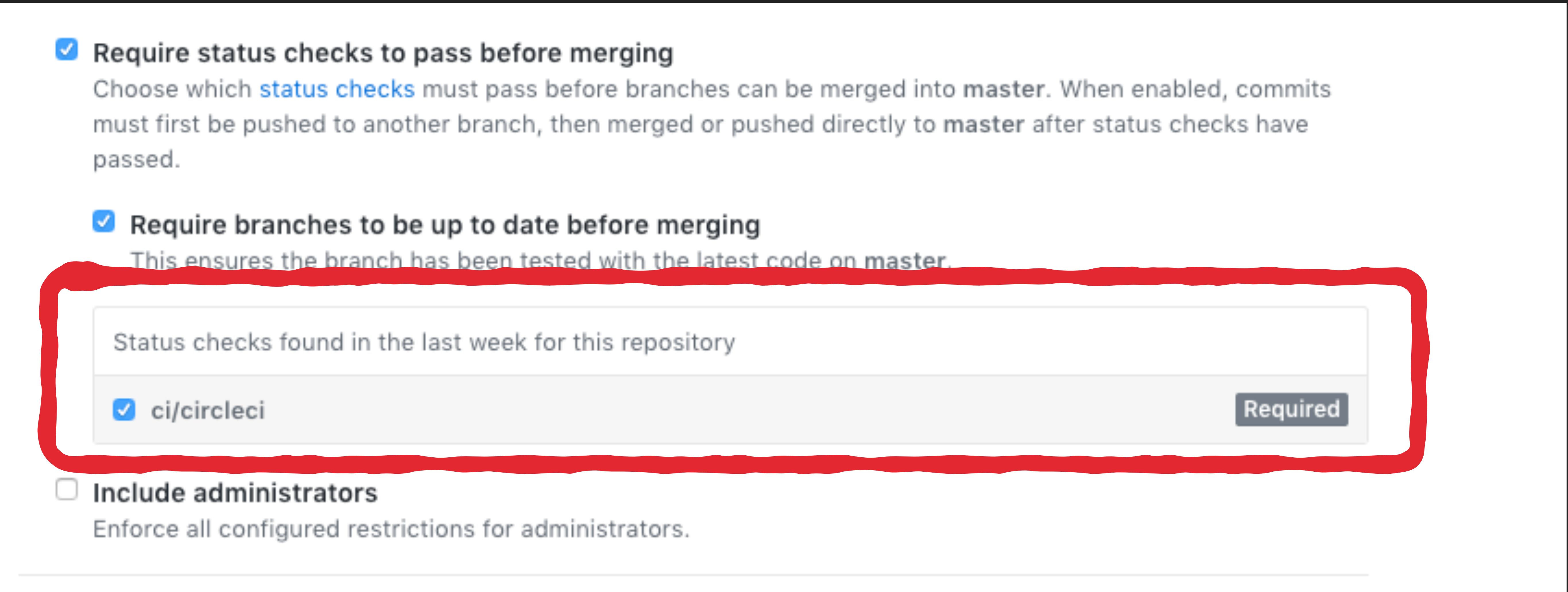
□ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.

☑ **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

□ **Include administrators**
Enforce all configured restrictions for administrators.

INTEGRATING CODE REVIEW INTO PROJECT WORKFLOW

- Branching strategy
- 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

SUMMARY

- Code review reduce costs
  - Fewer bugs
  - Cleaner code (which means faster development)
  - Quickly up skill new people to the project
- Tools today mean you can set up quickly

SUMMARY: REVIEW CHECKLIST

- Security problems
- Correct tests
- Clean code
- Bugs

QUESTIONS

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/

FEEDBACK

https://joind.in/talk/93a63