

DAVE LIDDAMENT

CODE REVIEW AND CONTINUOUS INTEGRATION



MORAL OF THE STORY

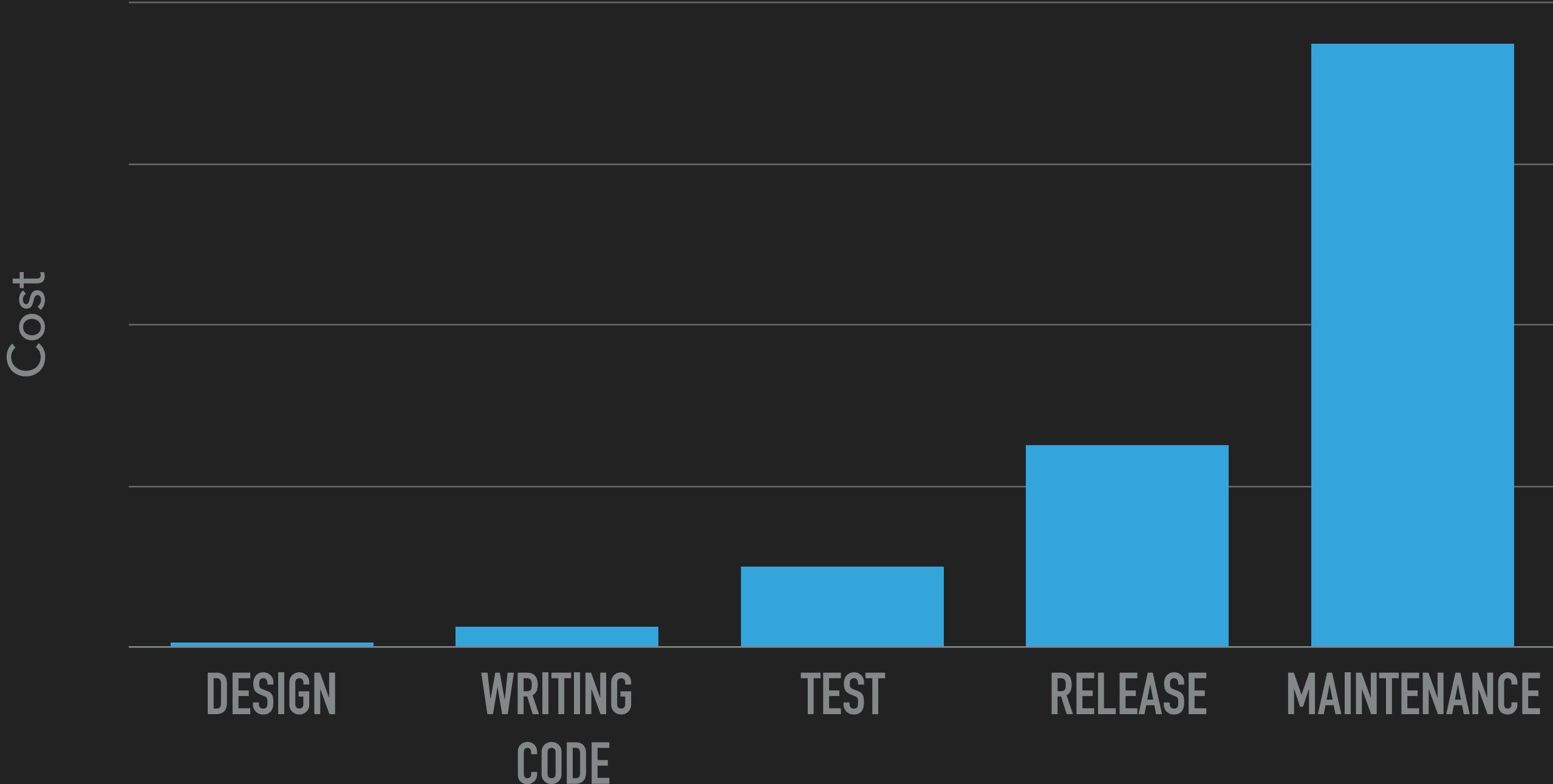
- ▶ We all make mistakes.
- ▶ If the mistakes are spotted and rectified quickly the consequences of the mistakes are minimal.
- ▶ We want to make mistakes easy to spot.

QUESTION

**HOW CAN WE REDUCE THE COST
OF SOFTWARE DEVELOPMENT?**

COST OF A BUG

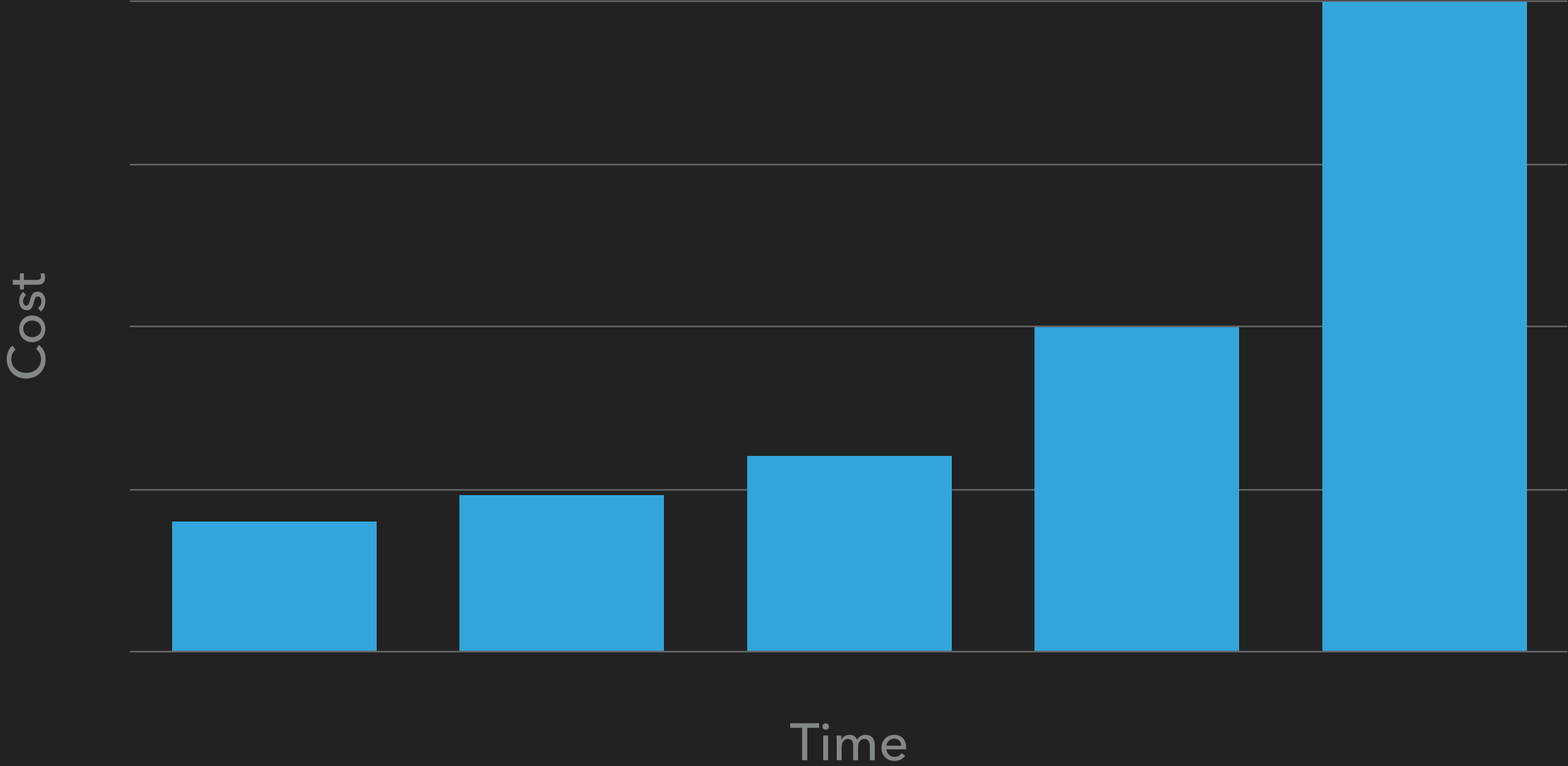
FIND BUGS SOONER



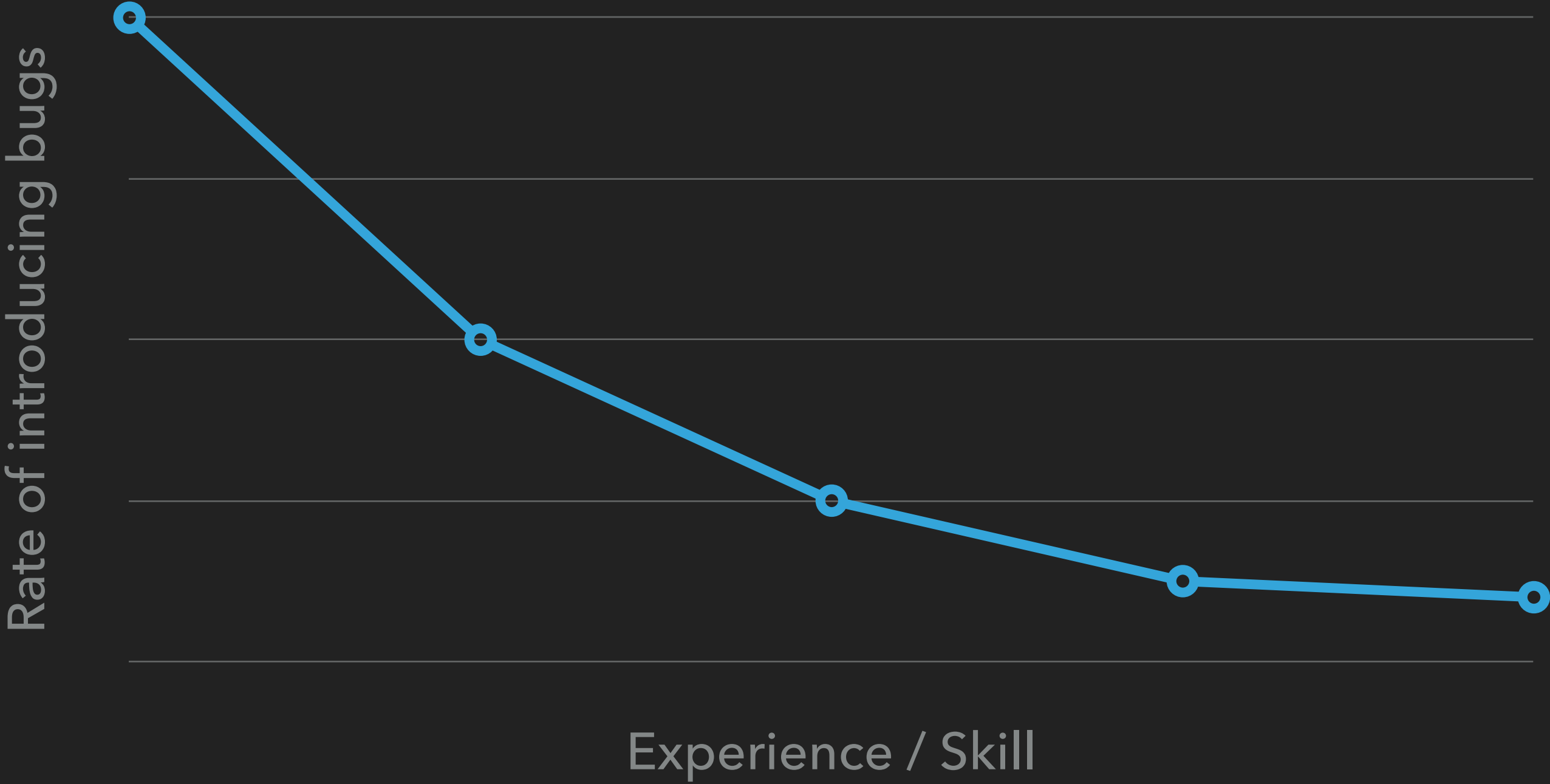
TYPES OF BUGS

- ▶ Code doesn't work
- ▶ Code works but does the wrong thing
- ▶ Poor architecture / design

COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



SKILL AND EXPERIENCE VS BUGS



CODE REVIEW AND CI DONE WELL WILL REDUCE COST OF SOFTWARE DEVELOPMENT

- ▶ Fewer bugs
- ▶ Cheaper bugs
- ▶ Write code quicker (by making code easier to read)
- ▶ Providing opportunities to quickly up skill team members

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 review
- ▶ CI - Continuous Integration
- ▶ Branching strategies
- ▶ Github / Bitbucket setup
- ▶ CircleCI example
- ▶ Wrap up

HOW

HOW ARE WE GOING TO REDUCE COSTS?

HOW ARE WE GOING TO REDUCE COSTS?

- ▶ Code Review
 - ▶ github
 - ▶ bitbucket
 - ▶ gerrit

HOW ARE WE GOING TO REDUCE COSTS?

- ▶ Code Review
 - ▶ github
 - ▶ bitbucket
 - ▶ gerrit
- ▶ Continuous Integration (CI)
 - ▶ Jenkins
 - ▶ CircleCI
 - ▶ Travis CI

CODE REVIEW

**CODE REVIEW SHOULD FOCUS ON
THINGS THAT CAN NOT BE AUTOMATED**

**DO THE TESTS TEST THE REQUIRED
FUNCTIONALITY?**

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.

HOW MANY TESTS DO WE NEED?

```
class Person
{
    /**
     * Returns true if the person is 18 or over
     */
    public function isAdult(): 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



“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

Reading

Writing



COST OF WRITING CODE

Reading

Writing



HOW CAN WE REDUCE COST OF WRITING CODE

COST OF WRITING CODE

Reading

Writing



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;
    }
}
```

WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

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 ...  
  
}
```

WILL I UNDERSTAND THE CODE IN 6 MONTHS TIME?

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);
}
```


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)

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

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
{
    /**
     * 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?**

Documentation for addEmailAddress

← → ↑ 📄 plotfinder-symfony

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->

m addEmailAddress (emailAddress : \EmailAddress) void

Press ^Space again to see more variants >>

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

```
/**
 * 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;
}
```


**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?

(E.G. SOLID)

BENEFITS OF CODE REVIEW

SOMEONE IS WATCHING YOU

LEARNING AND UPSKILLING

COST OF WRITING CODE

Reading

Writing



COST OF WRITING CODE

Reading

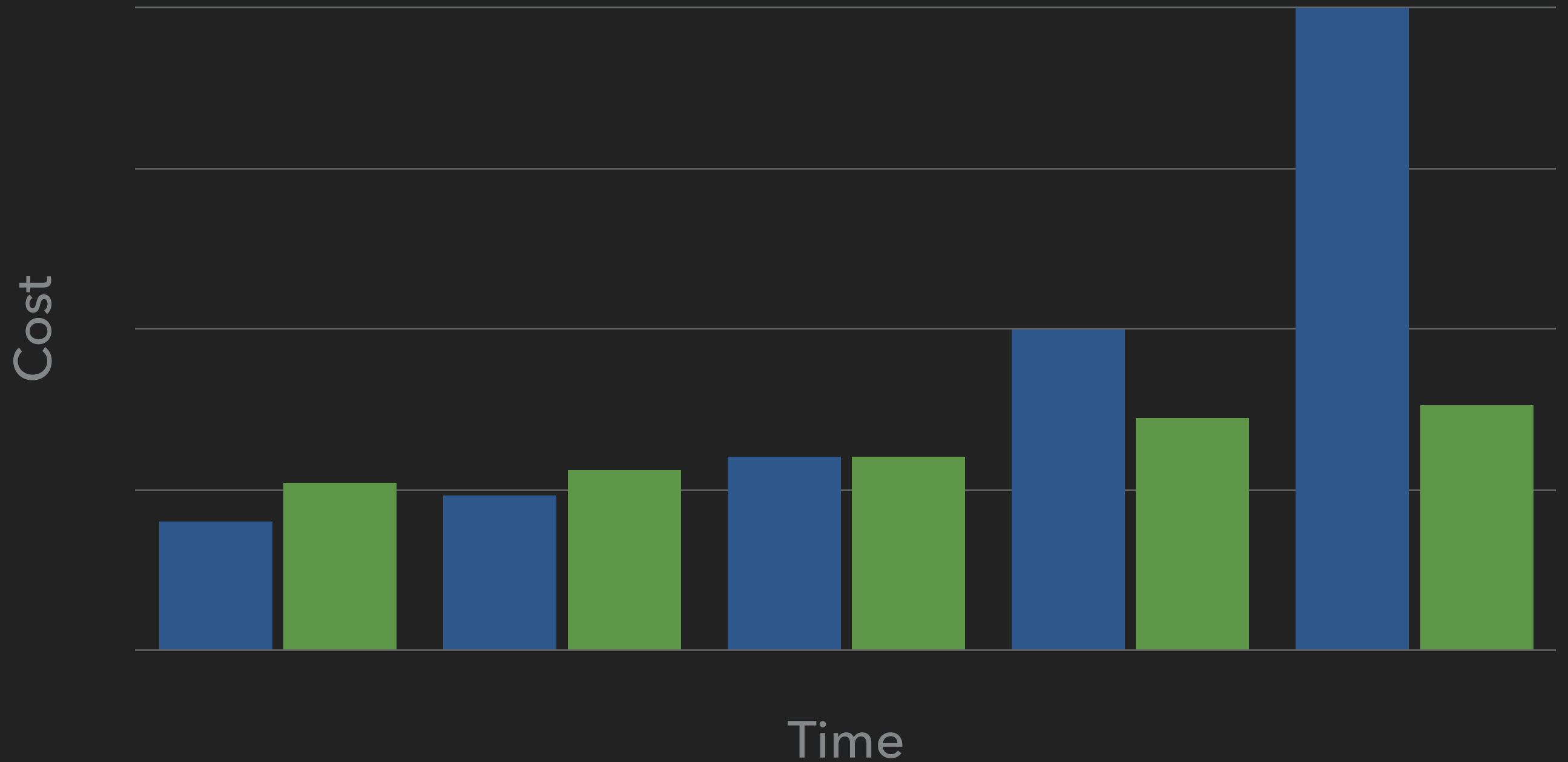
Writing



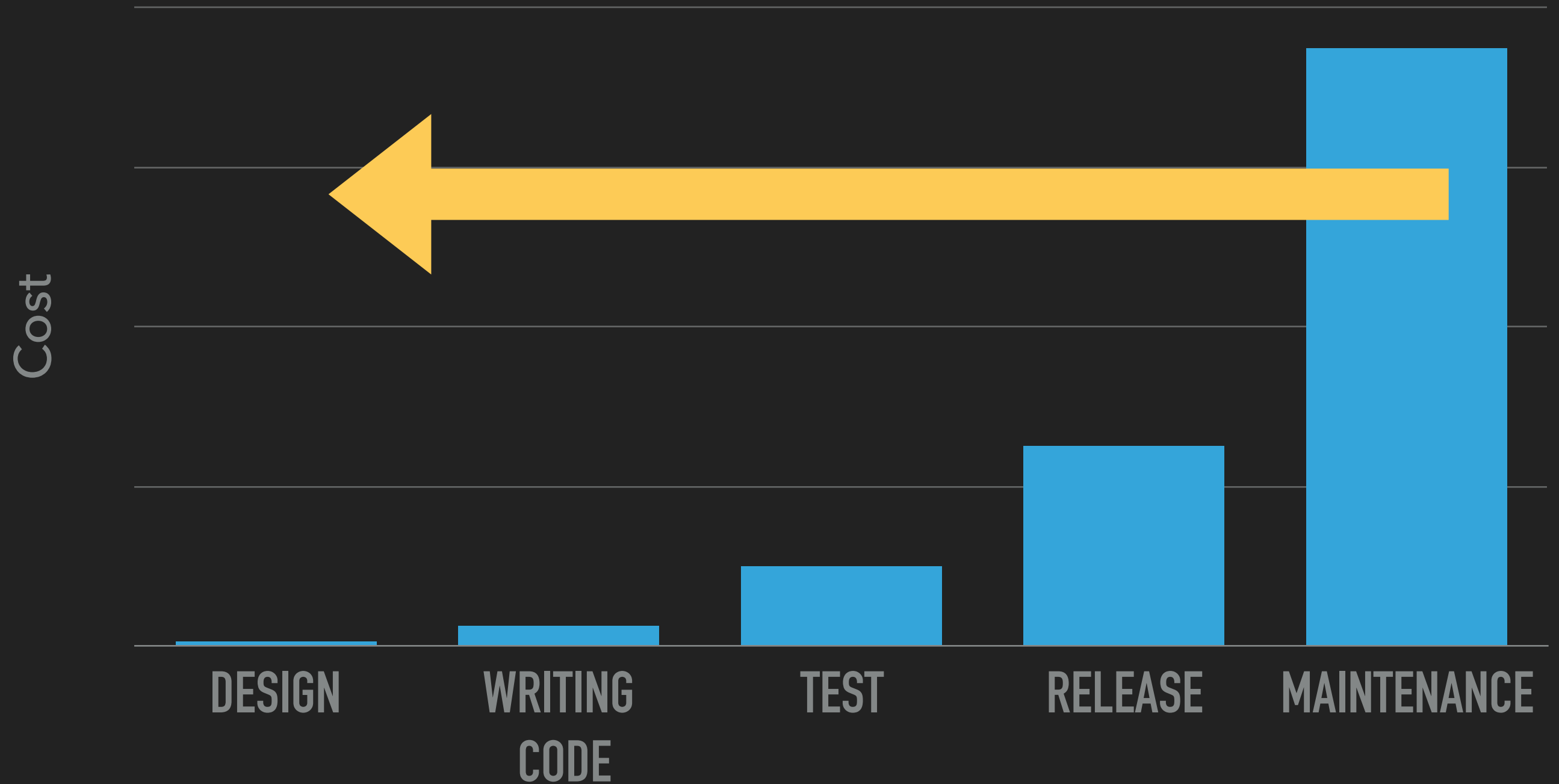
Code review + refactor



COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME



FIND BUGS SOONER



QUICK FEEDBACK LOOP

WHO SHOULD CONDUCT CODE REVIEWS

EVERYONE SHOULD CODE REVIEW

**WHAT MAKES CODE
EASY TO 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

**GOOD REVIEW
COMMENTS**

DON'T BE A *****

NOT CRITICAL OF THE CODE AUTHOR

**STATE PROBLEM AND
SUGGEST IMPROVEMENT**

GOOD REVIEW COMMENTS

LINK TO STACK OVERFLOW, BLOG, ETC

USE: 'LET'S CHAT'

USE 'QUESTION:'

RECEIVING REVIEW COMMENTS

RECEIVING REVIEW COMMENTS

DON'T BE A *****

CONTINUOUS INTEGRATION

RUNNING TESTS

LINT (OR BETTER STILL PARALLEL LINT)

```
composer require --dev jakub-ondrka/php-parallel-lint
```

```
vendor/bin/parallel-lint src
```

CODE STYLE CHECKER

```
composer require --dev friendsofphp/php-cs-fixer
```

```
vendor/bin/php-cs-fixer fix --dry-run src
```

Or locally to fix problems

```
vendor/bin/php-cs-fixer fix src
```

VAR DUMP CHECKER

```
composer require --dev jakub-onderka/php-var-dump-check
```

```
vendor/bin/php-var-dump-check --no-colors src
```

SYMFONY

```
bin/console lint:twig app
```

```
bin/console lint:yaml app
```

STATIC ANALYSIS TOOLS?

QUICK FEEDBACK LOOP

CODE REVIEW & CONTINUOUS INTEGRATION RECAP

BRANCHING STRATEGIES

ESSENTIAL - CODE CAN ONLY BE DEPLOYED IF:

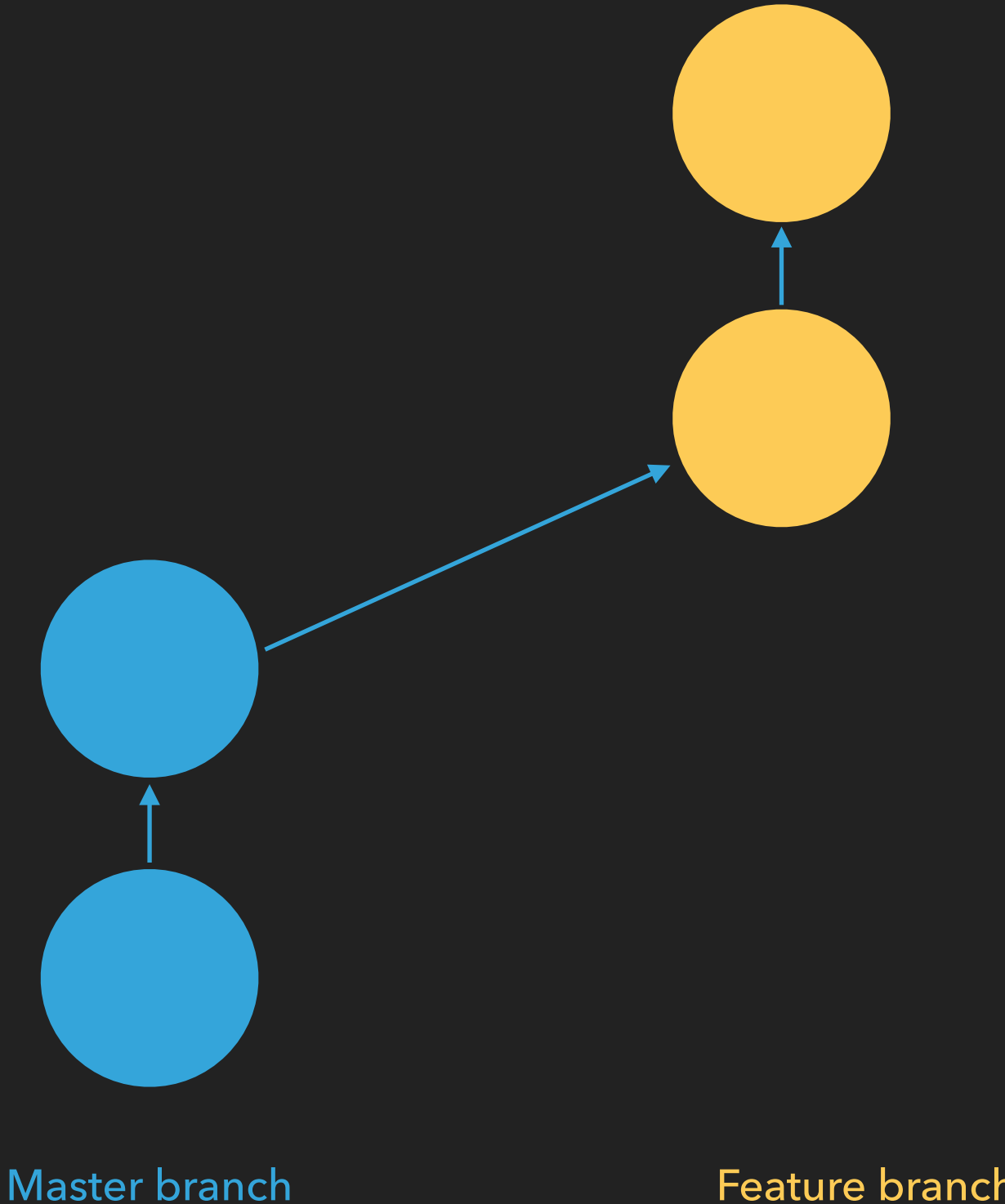
- ▶ CI passes
- ▶ Code review passes

IDEALLY - OTHERS ONLY DEVELOP WITH CODE THAT:

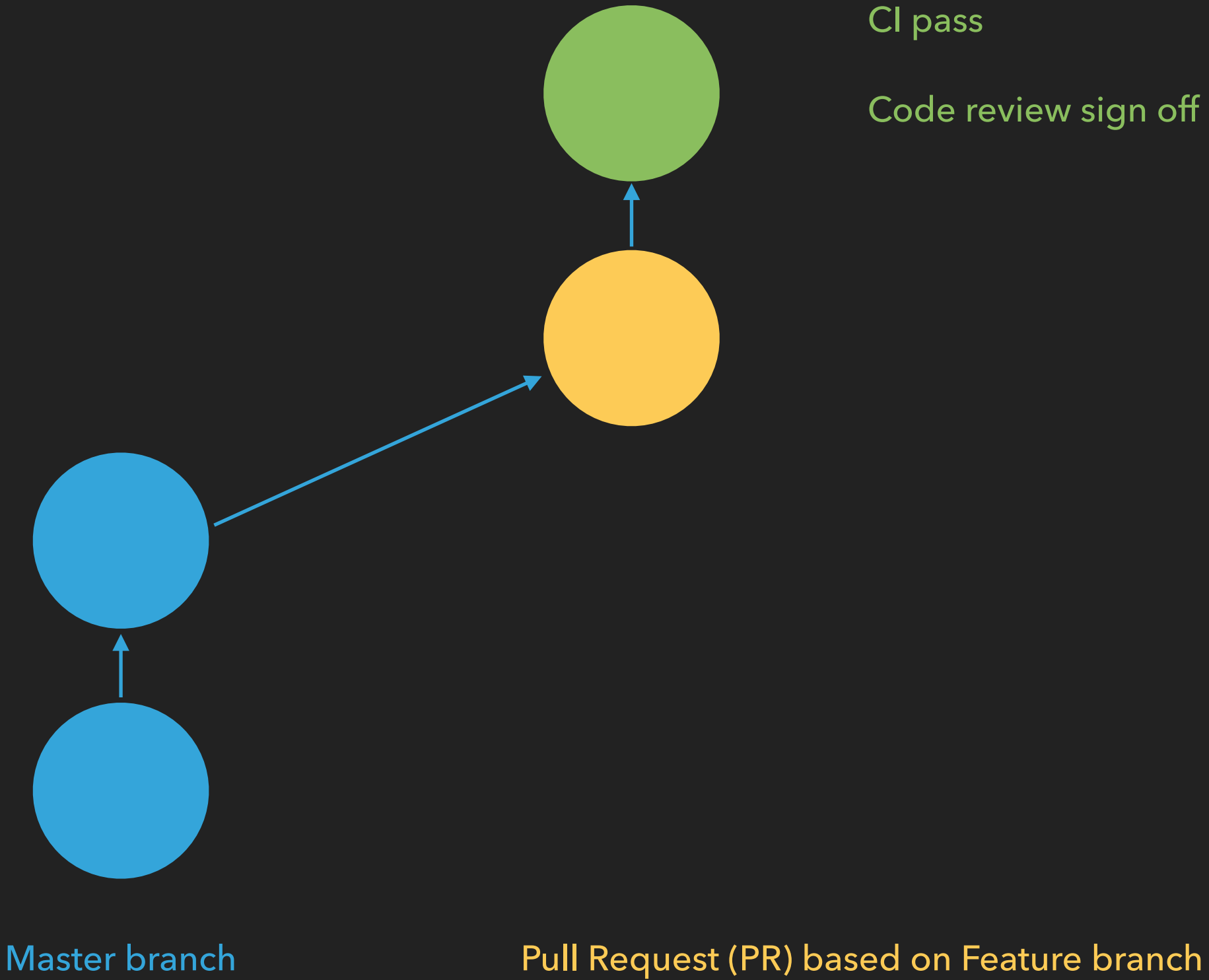
- ▶ CI passes
- ▶ Code review passes

PULL REQUESTS

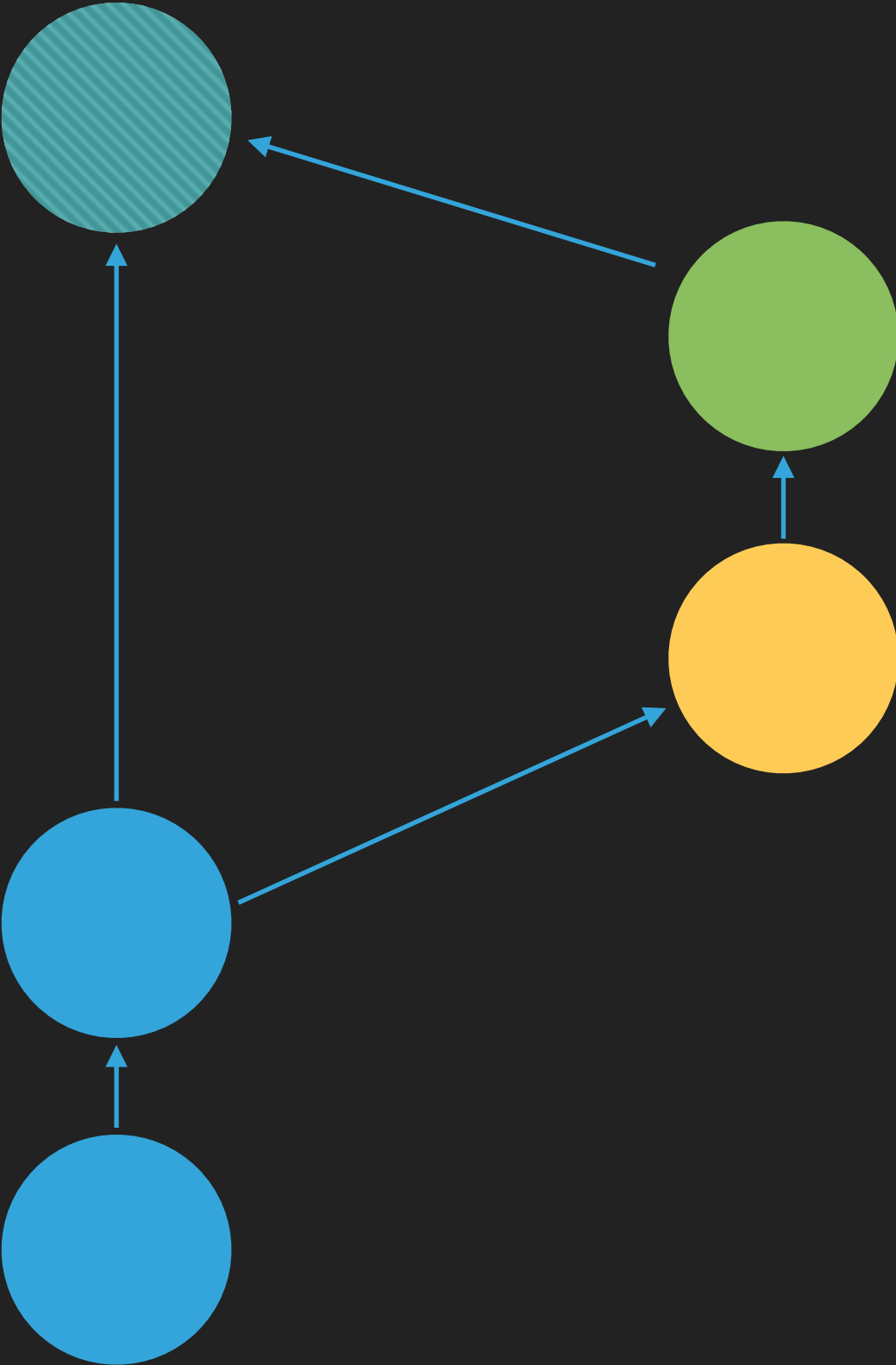
PULL REQUEST METHOD



PULL REQUEST METHOD



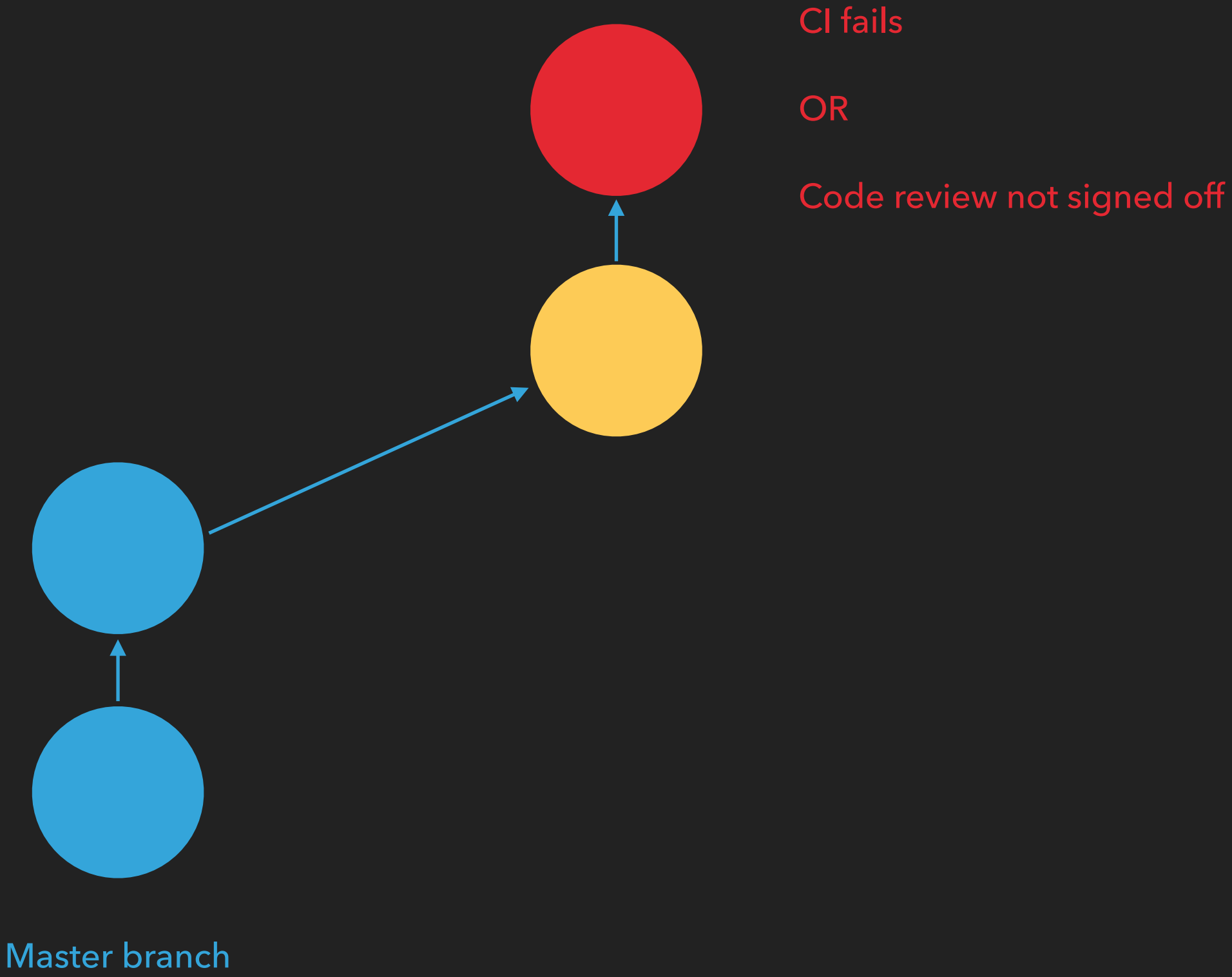
PULL REQUEST METHOD



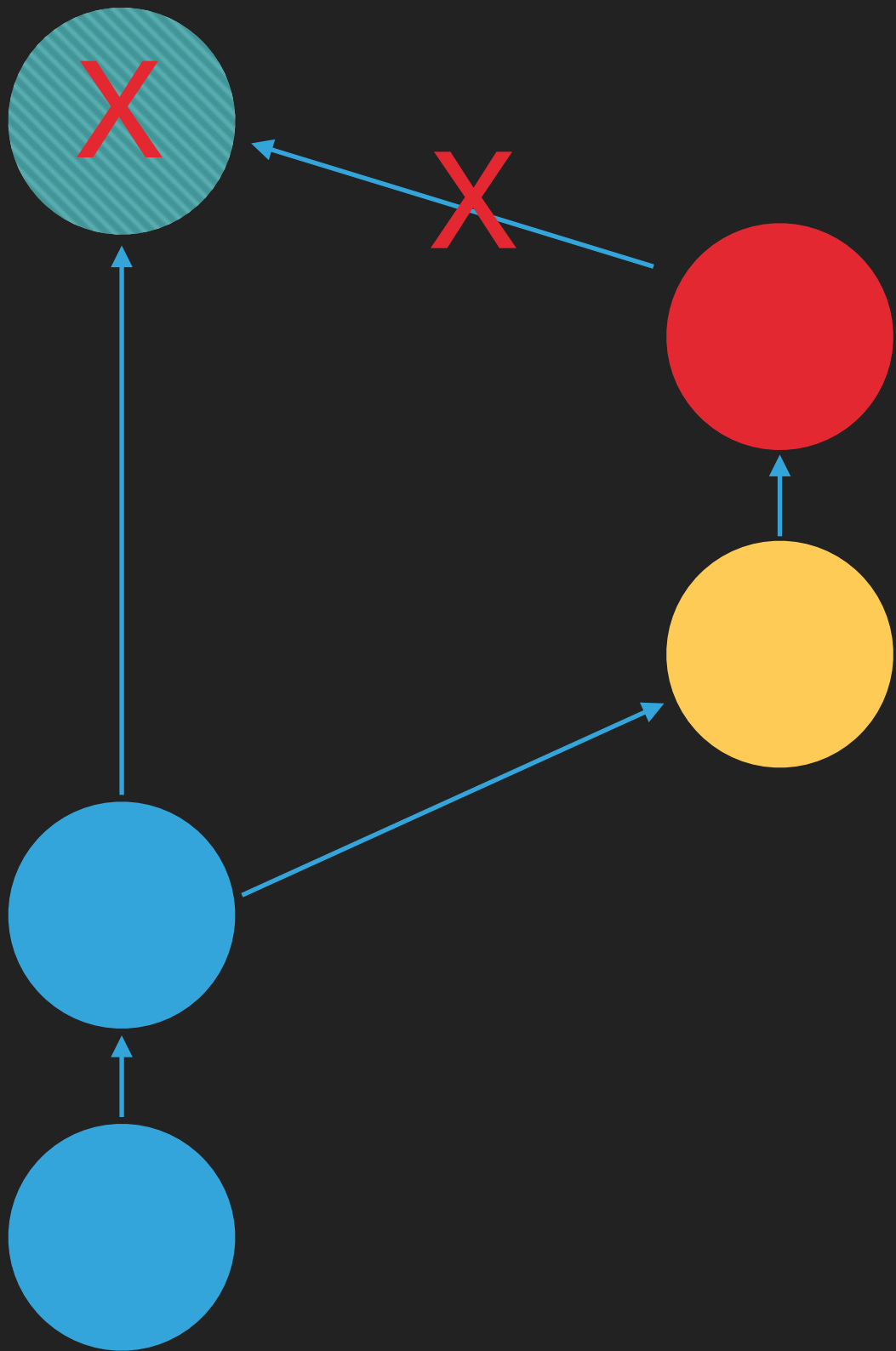
Master branch

CI pass
Code review sign off

PULL REQUEST METHOD



PULL REQUEST METHOD



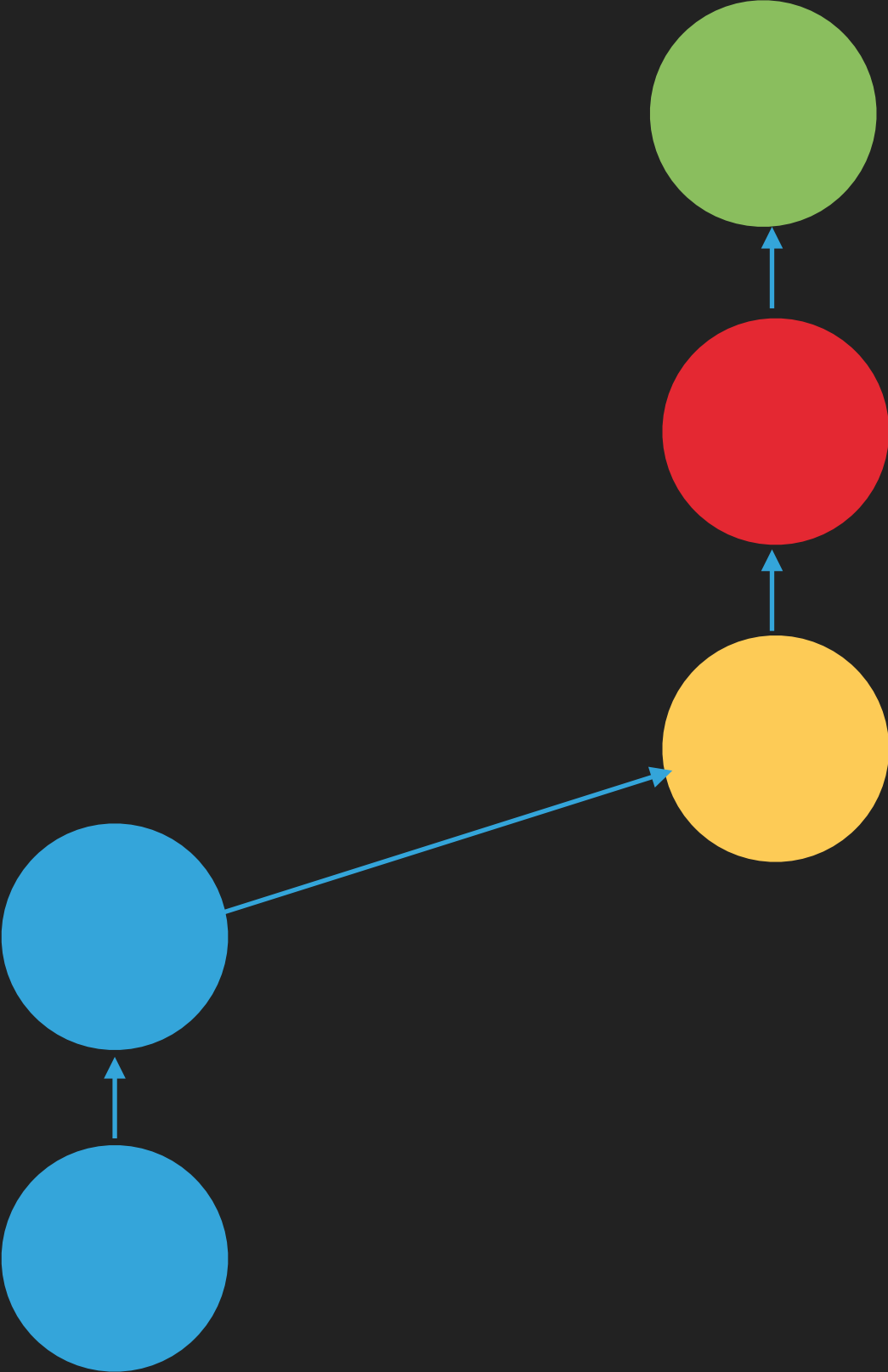
CI fails

OR

Code review not signed off

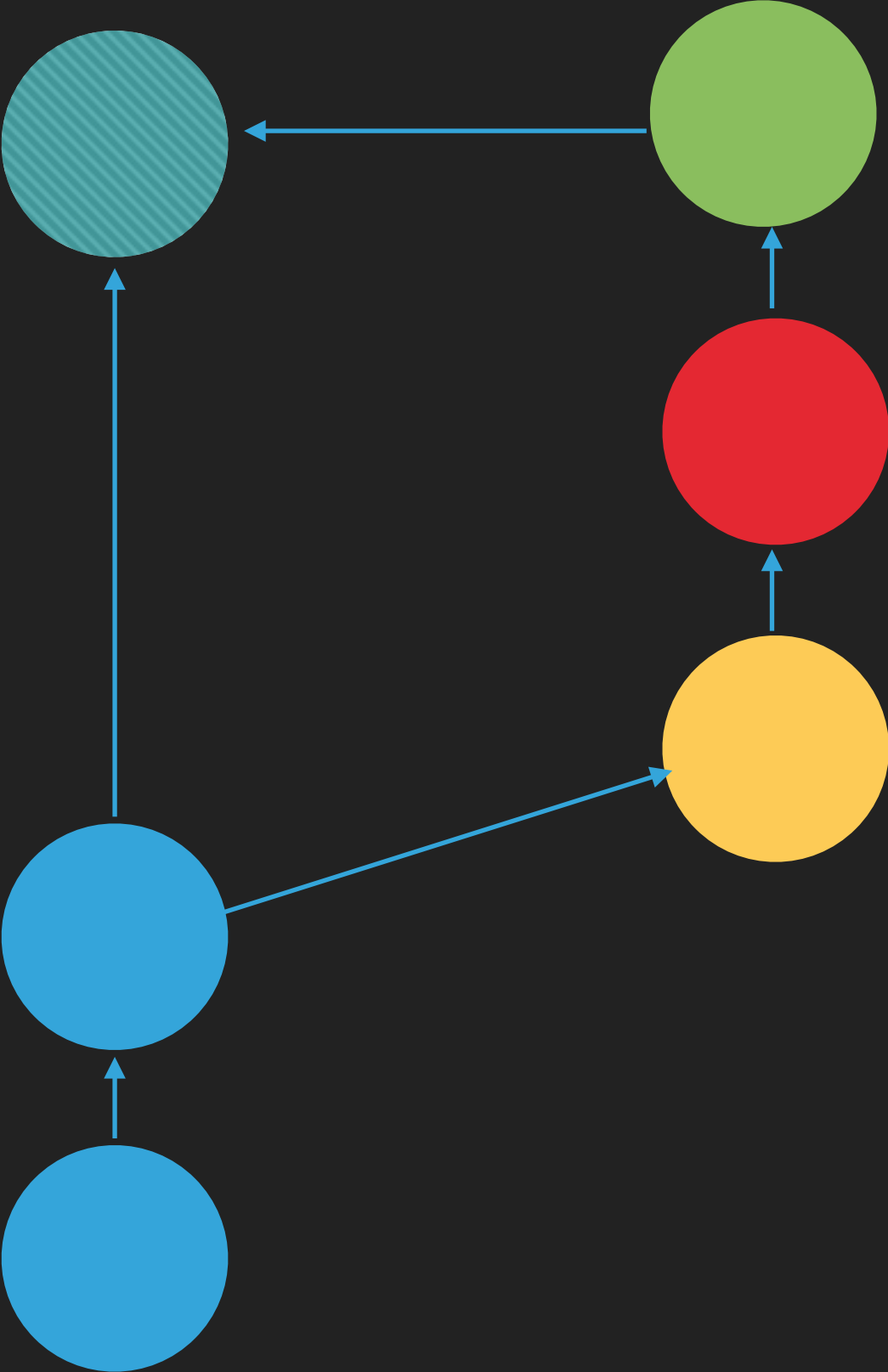
Master branch

PULL REQUEST METHOD

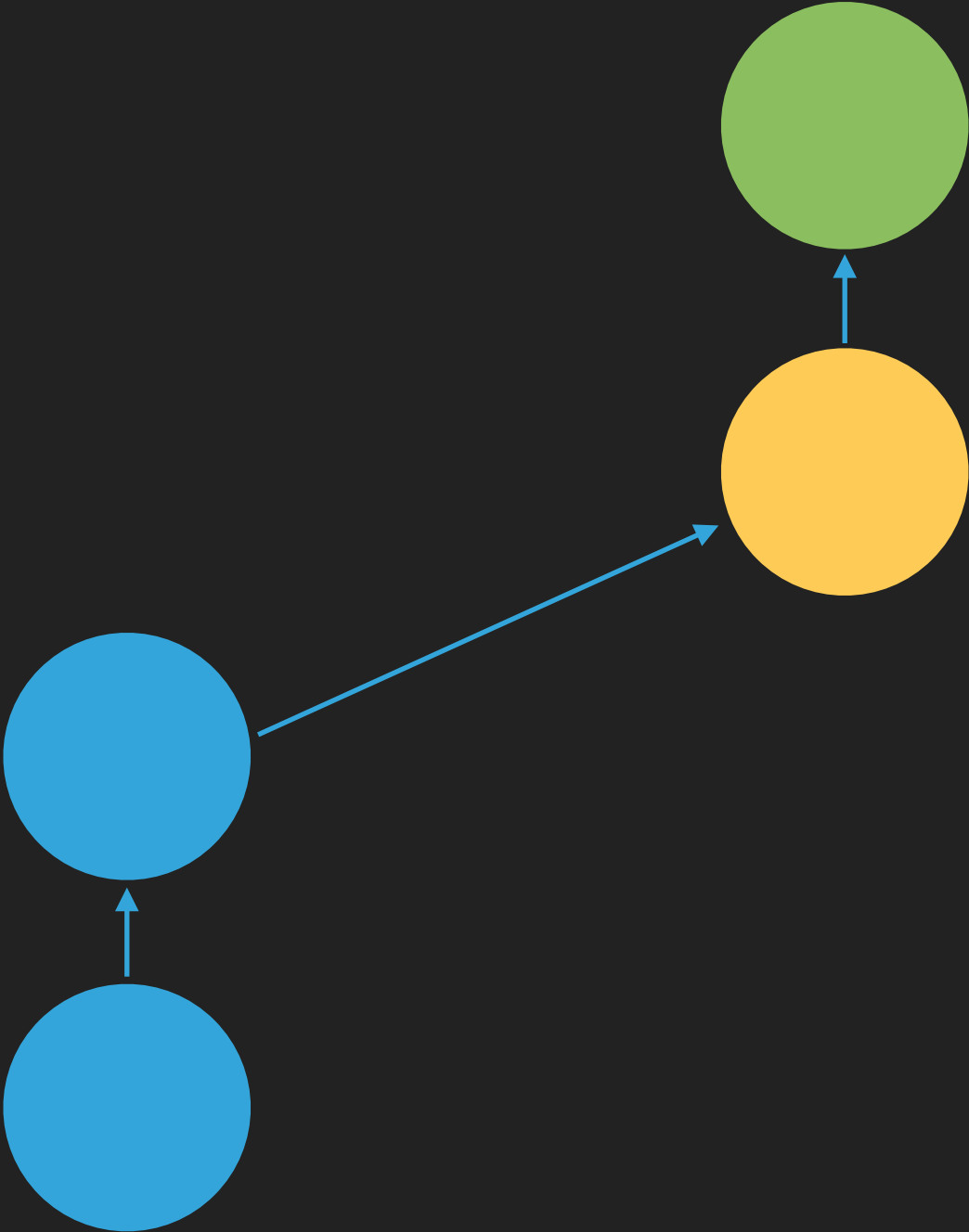


Master branch

PULL REQUEST METHOD

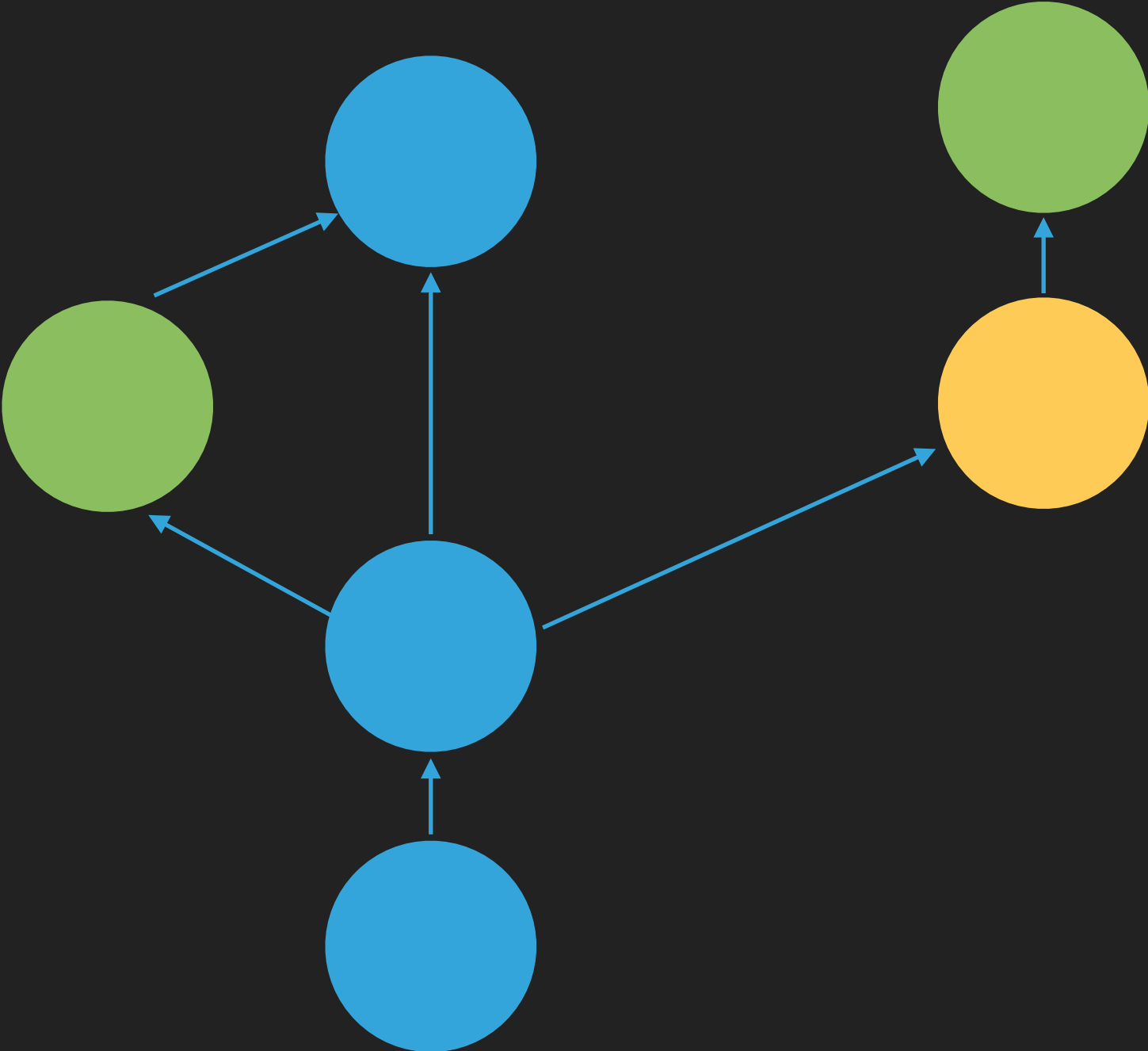


PULL REQUEST METHOD



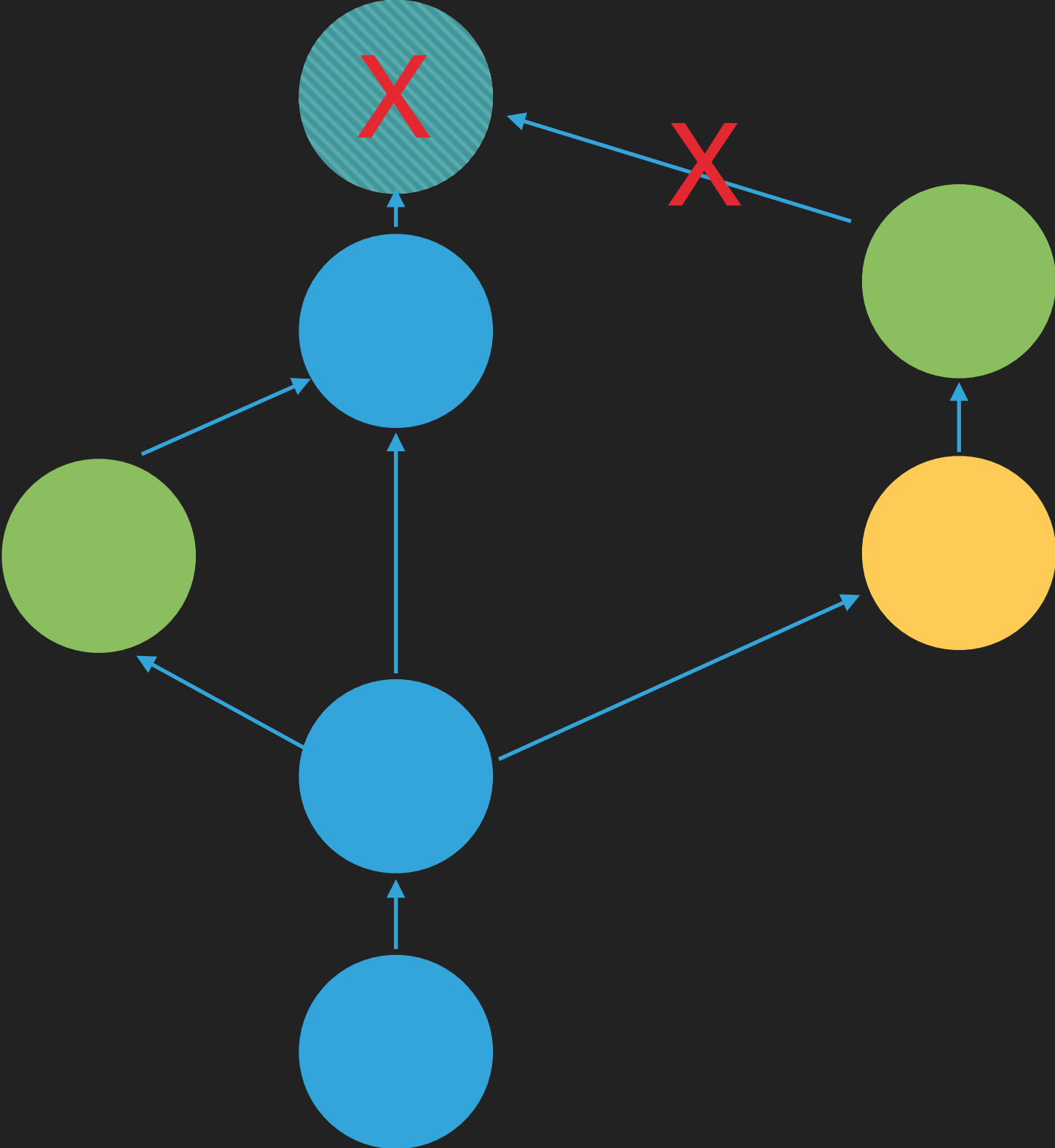
Master branch

PULL REQUEST METHOD



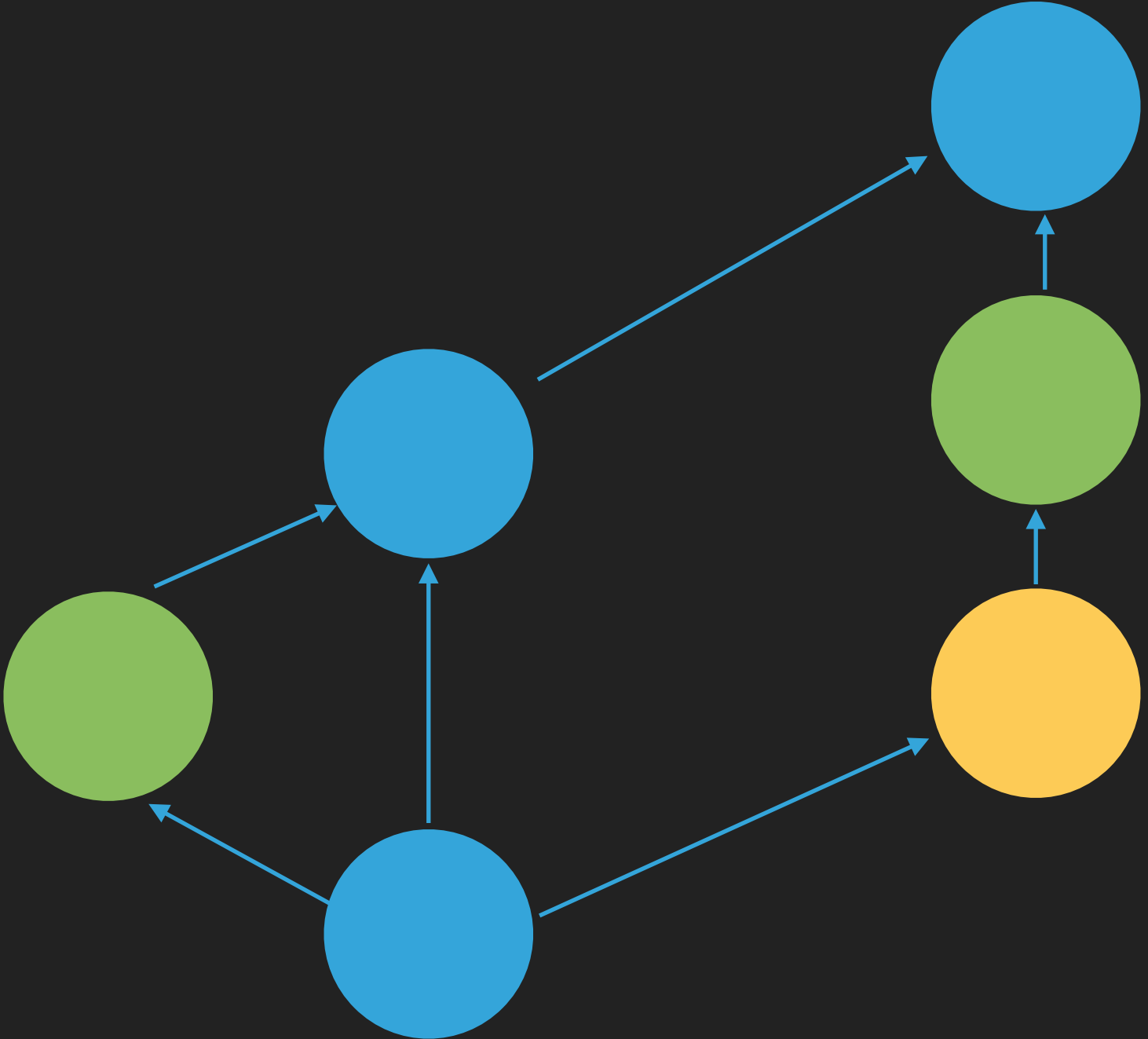
Master branch

PULL REQUEST METHOD



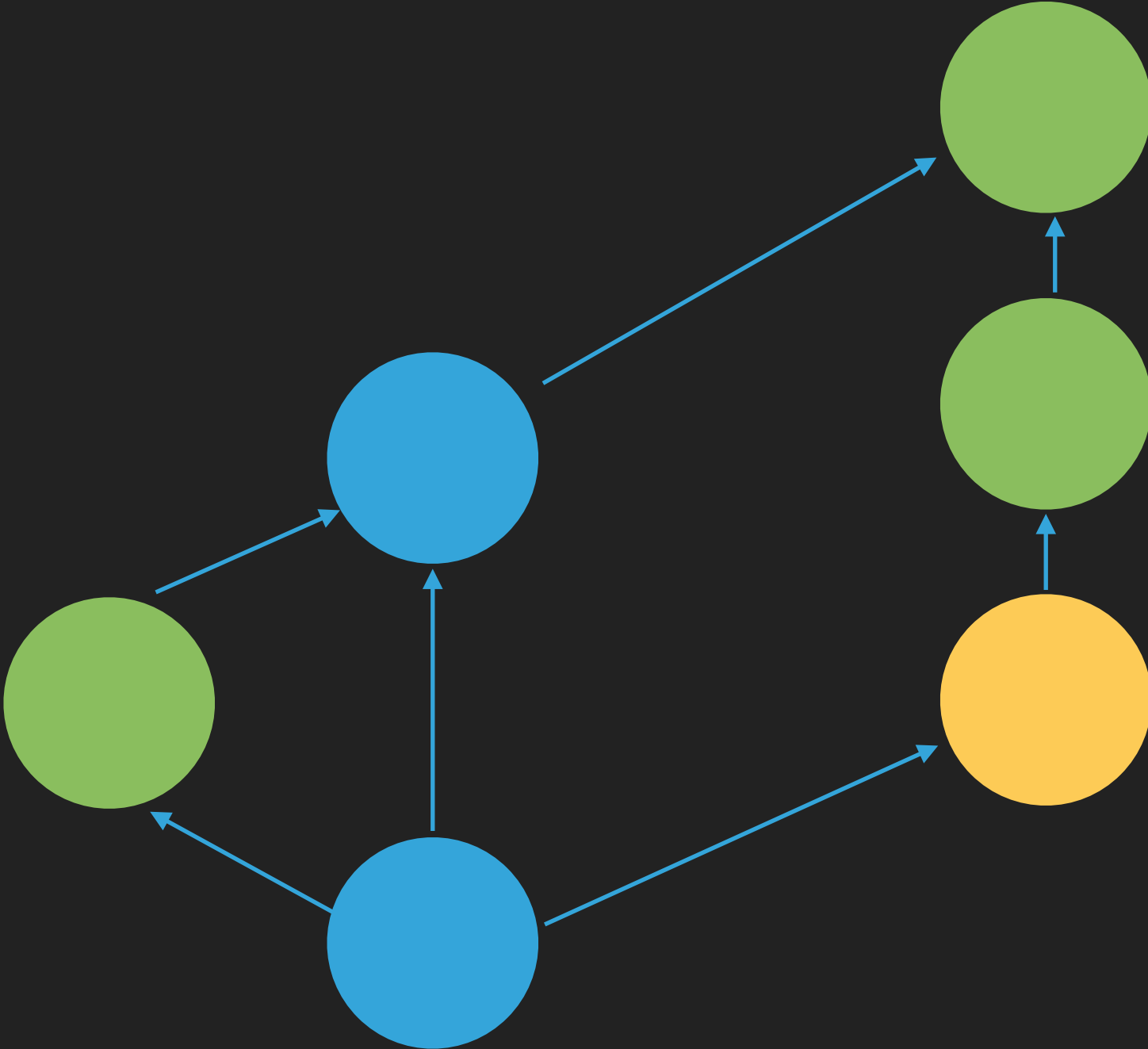
Master branch

PULL REQUEST METHOD



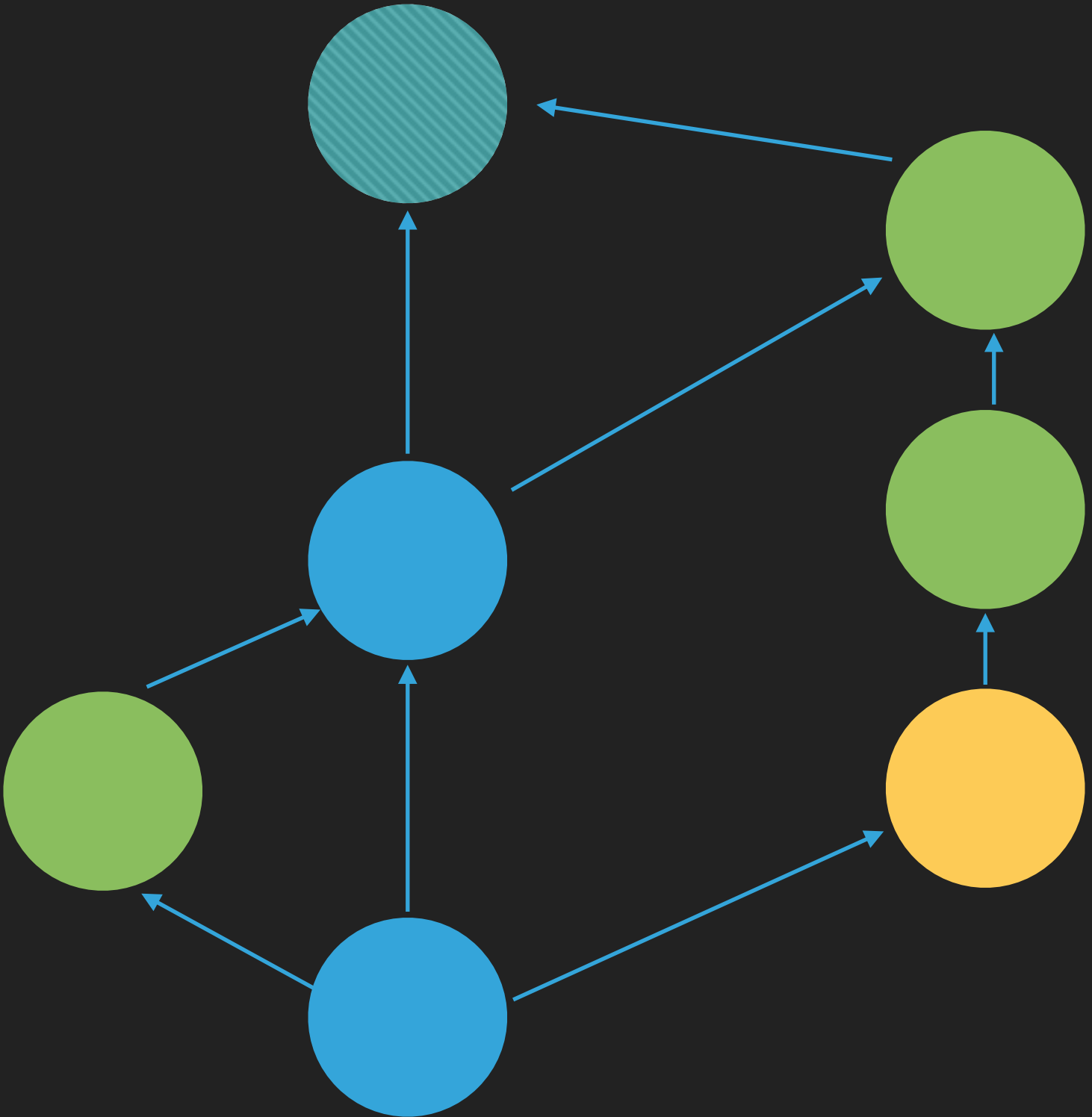
Master branch

PULL REQUEST METHOD



Master branch

PULL REQUEST METHOD



Master branch

LET'S SET THIS UP

LET'S SET THIS UP

PS It's really easy

**THE PROCESS IS IMPORTANT
NOT THE TOOLS**

GITHUB

SETUP GITHUB

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

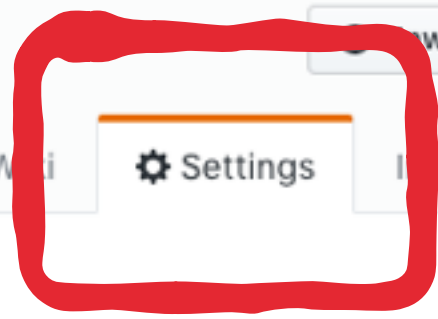
Protected branches

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



- 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

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

- 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.

Protected branches

For more information on how to enable force pushing, prevent branches from being deleted, and optionally require status checks before merging. [View to protected branches? Learn more.](#)

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 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.](#)

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.](#)

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.

BITBUCKET

SETUP BITBUCKET

The screenshot shows the Bitbucket web interface. On the left is a blue sidebar with navigation icons and labels: Demo project, Overview, Source, Commits, Branches, Pull requests, Pipelines, Issues (with a 'TRY' badge), Downloads, and Settings (highlighted). Below the sidebar is a 'SHARE YOUR THOUGHTS' section with 'Give feedback' and 'Turn off new nav' options. The main content area has a breadcrumb 'Lamp Devs / Lamp Bristol / Demo project' and a 'Settings' header. The settings are categorized into: GENERAL (Repository details, User and group access, Access keys, Username aliases), WORKFLOW (Branch permissions, Default reviewers, Webhooks, Links), FEATURES (Git LFS, Wiki, Issue tracker), BITBUCKET FOR HIPCHAT (HipChat integration), and PIPELINES (Settings, Environment variables, SSH keys, Integrations). The 'Branch permissions' section is expanded, showing a sub-header 'Branch permissions', an illustration of a terminal window with a branch icon, and the text 'Manage who can write or merge to specific branches' followed by 'This helps prevent accidents, like somebody deleting master.' At the bottom of this section are two buttons: 'Add a branch permission' and 'Learn more'.

Lamp Devs / Lamp Bristol / Demo project

Settings

GENERAL

- Repository details
- User and group access
- Access keys
- Username aliases

WORKFLOW

- Branch permissions**
- Default reviewers
- Webhooks
- Links

FEATURES

- Git LFS
- Wiki
- Issue tracker

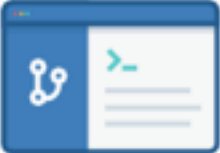
BITBUCKET FOR HIPCHAT

- HipChat integration

PIPELINES

- Settings
- Environment variables
- SSH keys
- Integrations

Branch permissions



Manage who can write or merge to specific branches
This helps prevent accidents, like somebody deleting master.

[Add a branch permission](#) [Learn more](#)

SETUP BITBUCKET

The screenshot shows the Bitbucket interface for a project named 'Demo project'. The left sidebar contains navigation options: Overview, Source, Commits, Branches, Pull requests, Pipelines, and Issues. The 'Settings' option is highlighted with a red circle. The main content area is titled 'Settings' and is divided into several sections: GENERAL (Repository details, User and group access, Access keys, Username aliases), WORKFLOW (Branch permissions, Default reviewers, Webhooks, Links), FEATURES (Git LFS, Wiki, Issue tracker), BITBUCKET FOR HIPCHAT (HipChat integration), and PIPELINES (Settings, Environment variables, SSH keys, Integrations). The 'Branch permissions' section is expanded, showing a sub-header 'Branch permissions' and a description: 'Manage who can write or merge to specific branches. This helps prevent accidents, like somebody deleting master.' Below this description are two buttons: 'Add a branch permission' and 'Learn more'. An icon of a terminal window with a branch symbol is also visible.

Lamp Devs / Lamp Bristol / Demo project

Settings

GENERAL

- Repository details
- User and group access
- Access keys
- Username aliases

WORKFLOW

- Branch permissions**
- Default reviewers
- Webhooks
- Links

FEATURES

- Git LFS
- Wiki
- Issue tracker

BITBUCKET FOR HIPCHAT

- HipChat integration

PIPELINES

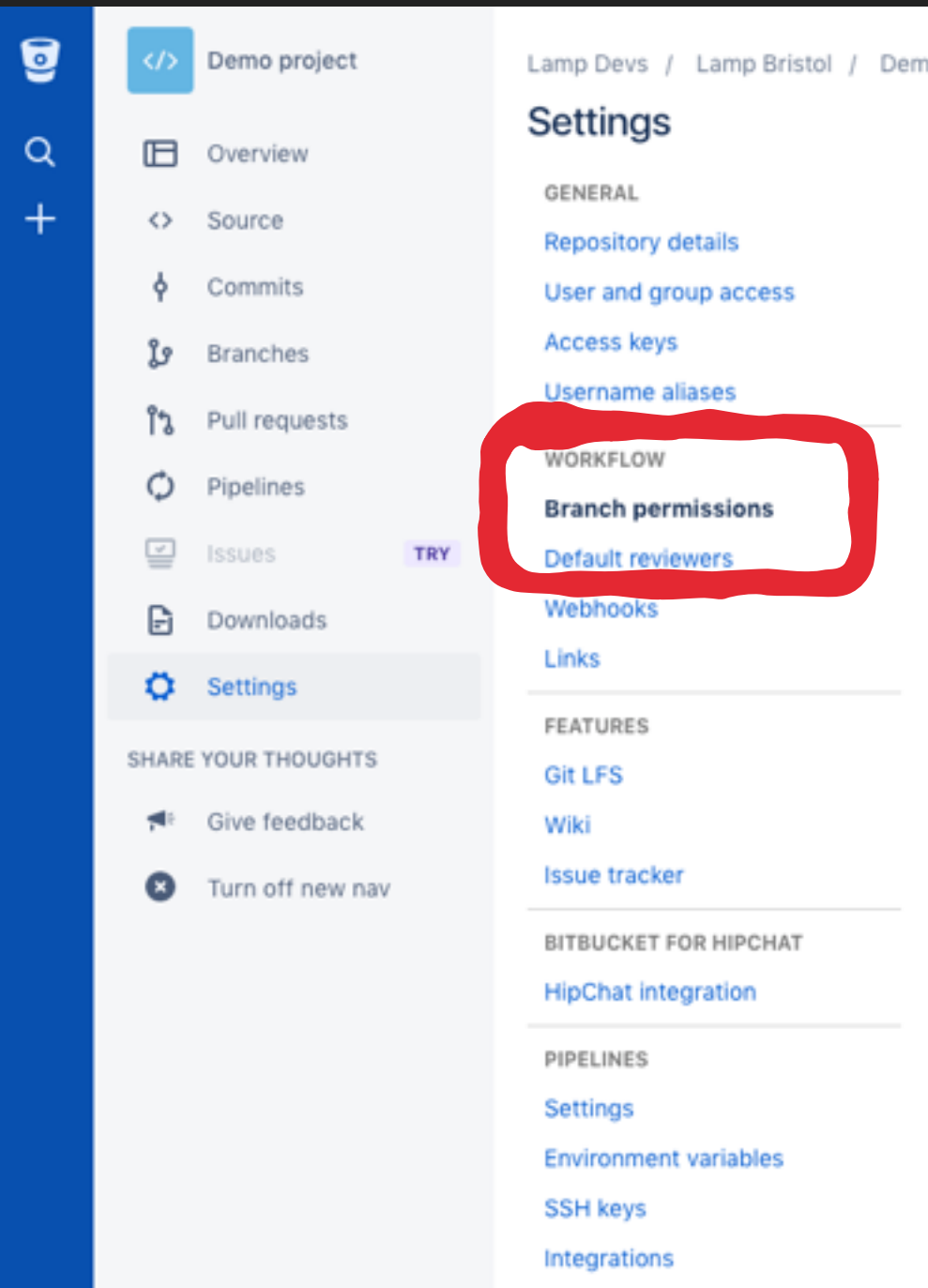
- Settings
- Environment variables
- SSH keys
- Integrations

Branch permissions

Manage who can write or merge to specific branches
This helps prevent accidents, like somebody deleting master.

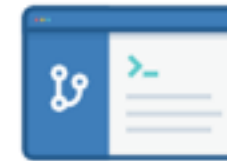
[Add a branch permission](#) [Learn more](#)

SETUP BITBUCKET



The screenshot shows the Bitbucket interface for a project named 'Demo project'. The left sidebar contains navigation options: Overview, Source, Commits, Branches, Pull requests, Pipelines, Issues (with a 'TRY' badge), Downloads, and Settings (highlighted). The main content area is titled 'Settings' and is divided into several sections: GENERAL (Repository details, User and group access, Access keys, Username aliases), WORKFLOW (Branch permissions, Default reviewers), FEATURES (Git LFS, Wiki, Issue tracker), BITBUCKET FOR HIPCHAT (HipChat integration), and PIPELINES (Settings, Environment variables, SSH keys, Integrations). The 'Branch permissions' link under the WORKFLOW section is highlighted with a red rectangle.

Branch permissions



Manage who can write or merge to specific branches

This helps prevent accidents, like somebody deleting master.

[Add a branch permission](#)

[Learn more](#)

SETUP BITBUCKET

The screenshot shows the Bitbucket web interface. On the left is a navigation sidebar with icons for Overview, Source, Commits, Branches, Pull requests, Pipelines, Issues, Downloads, and Settings (which is highlighted). Below the sidebar are options to 'SHARE YOUR THOUGHTS' such as 'Give feedback' and 'Turn off new nav'. The main content area is titled 'Settings' and includes a breadcrumb 'Lamp Devs / Lamp Bristol / Demo project'. Under the 'Settings' heading, there are several categories: 'GENERAL' (Repository details, User and group access, Access keys, Username aliases), 'WORKFLOW' (Branch permissions, Default reviewers, Webhooks, Links), 'FEATURES' (Git LFS, Wiki, Issue tracker), 'BITBUCKET FOR HIPCHAT' (HipChat integration), and 'PIPELINES' (Settings, Environment variables, SSH keys, Integrations). The 'Branch permissions' section is expanded, showing a sub-header 'Branch permissions' and a description: 'Manage who can write or merge to specific branches. This helps prevent accidents, like somebody deleting master.' Below this is a blue button labeled 'Add a branch permission' and a link 'Learn more'. The button and link are enclosed in a red hand-drawn rectangle.

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check for successful build on the last commit

Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

Save

Cancel

SETUP BITBUCKET

Edit branch permission

Branch or pattern

Write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check for successful build on the last commit
Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

SETUP BITBUCKET

Edit branch permission

Branch or pattern

Write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check for successful build on the last commit
Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check for successful build on the last commit

Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

Save

Cancel

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least 1 approval
- Check for unresolved pull request tasks
- Check for 1 successful build on the last commit
Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

Save

Cancel

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

Check for at least 1 approval

Check for unresolved pull request tasks

Check for 1 successful build on the last commit

Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

Prevent a merge with unresolved merge checks

Reset approvals when the source branch is modified

Save

Cancel

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

Check for at least 1 approval

Check for unresolved pull request checks

Check for 1 successful build on the last commit

Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

Prevent a merge with unresolved merge checks

Reset approvals when the source branch is modified

Save

Cancel

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check for successful build on the last commit

Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks

Save

Cancel

SETUP BITBUCKET

Edit branch permission

Branch or pattern

master

Write access

Nobody has write access

Users and groups who can push or merge any changes to this branch directly

Merge via pull request

Everybody x

Users and groups who can merge to this branch via pull request

▼ Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check for successful build on the last commit

Also notifies users of failed builds before merging

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

Save

Cancel

CIRCLE CI

You're Almost There.

Signing up with CircleCI is free. Next, you'll be taken to GitHub, Bitbucket, or Google to authenticate so you can start shipping faster.



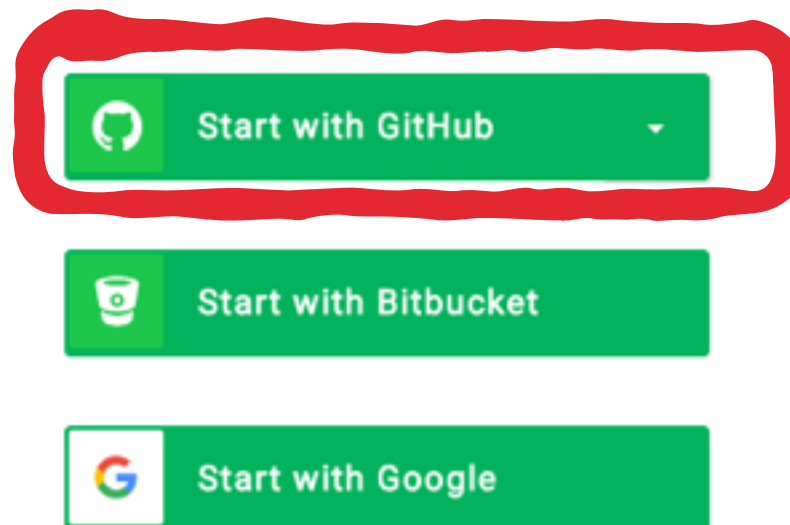
By clicking any start button above, you are agreeing to our [Terms of Service](#) and [Privacy Policy](#).



Your information is safe with us! We ask for read/write access to your repos to make your experience seamless on CircleCI. [Not ready to add private projects yet?](#)

You're Almost There.

Signing up with CircleCI is free. Next, you'll be taken to GitHub, Bitbucket, or Google to authenticate so you can start shipping faster.



By clicking any start button above, you are agreeing to our [Terms of Service](#) and [Privacy Policy](#).



Your information is safe with us! We ask for read/write access to your repos to make your experience seamless on CircleCI. [Not ready to add private projects yet?](#)

You're Almost There.

Signing up with CircleCI is free. Next, you'll be taken to GitHub, Bitbucket, or Google to authenticate so you can start shipping faster.

 Start with GitHub

 Start with Bitbucket

 Start with Google

By clicking any start button above, you are agreeing to our [Terms of Service](#) and [Privacy Policy](#).



Your information is safe with us! We ask for read/write access to your repos to make your experience seamless on CircleCI. [Not ready to add private projects yet?](#)

SETUP CIRCLE CI



Authorize CircleCI



CircleCI by [circleci](#)

wants to access your [lampbristolteam](#) account



Personal user data

Email addresses (read-only)



Repositories

Public and private



Organization access

 [lampbristol](#) ✓

Authorizing will redirect to
<https://circleci.com>

Authorize circleci



Not owned or
operated by GitHub



Created
6 years ago



More than 1K
GitHub users

[Learn more about OAuth](#)

SETUP CIRCLE CI



Authorize CircleCI



CircleCI by [circleci](#)

wants to access your [lampbristolteam](#) account



Personal user data

Email addresses (read-only)



Repositories

Public and private



Organization access

 lampbristol ✓

Authorizing will redirect to
<https://circleci.com>

Authorize circleci



Not owned or
operated by GitHub



Created
6 years ago



More than 1K
GitHub users

[Learn more about OAuth](#)

SETUP CIRCLE CI

Add Projects

CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

To kick things off, you'll need to choose a project to build. We'll start a new build for you each time someone pushes a new commit.

- Choose an organization that you are a member of.
- Choose a repo to add to CircleCI. We'll start a new build for you each time someone pushes a new commit. You can also follow a project that's already been added to CircleCI. You'll see your followed projects in [Builds](#) and [Insights](#).

GitHub Bitbucket

DaveLiddament

lampbristol

phpsw

Linux OS X

Filter projects... Show Forks

circle_demo_project

Demo

AliceBundle

Unfollow all projects

Setup project

Setup project

Setup project

SETUP CIRCLE CI

Add Projects Check Out Our New Look Reload Organizations

CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

To kick things off, you'll need to choose a project to build. We'll start a new build for you each time someone pushes a new commit.

- 1 Choose an organization that you are a member of.
- 2 Choose a repo to add to CircleCI. We'll start a new build for you each time someone pushes a new commit. You can also follow a project that's already been added to CircleCI. You'll see your followed projects in [Builds](#) and [Insights](#).

GitHub Bitbucket

Linux OS X

Filter projects... Show Forks

DaveLiddament	circle_demo_project	Unfollow all projects	Setup project
lampbristol			
phpsw	AliceBundle		Setup project

SETUP CIRCLE CI

The screenshot displays the CircleCI interface for a project named 'DaveLiddament/circle_demo_project'. The main area shows a list of build jobs with their status, names, and completion times. The jobs are sorted by project and filtered by 'All branches'. The jobs are as follows:

Job Name	Status	Step	Time
feature/circle_ci_2	Failed	1 day ago	
feature/foo	Failed	2 months ago	
master	Failed	1 day ago	
step-01/add-circle-ci-config	Success	2 months ago	
step-02/cache-composer-files	Success	2 months ago	
step-03/add-database-test	Failed	2 months ago	
step-04/fix-database-test	Success	2 months ago	
step-05/add-parallel-lint	Failed	2 months ago	
step-06/more-lint	Success	2 months ago	
step-07/add-var_dump-checker	Success	2 months ago	
step-08/add-cs-fixer	Failed	2 months ago	

The detailed view of the build jobs is as follows:

Job ID	Status	Job Name	Step	Time
6559f82	Success	DaveLiddament / ciricle_demo_project / step-08/add-cs-fixer #18	REFACTOR apply CS fixes	02:02
e159277	Success	DaveLiddament / ciricle_demo_project / step-07/add-var_dump-checker #17	ADD vardump checker	02:14
8823852	Success	DaveLiddament / ciricle_demo_project / step-06/more-lint #16	ADD more linting (symfony projects only)	02:12
f63c695	Fixed	DaveLiddament / ciricle_demo_project / step-05/add-parallel-lint #15	ADD lint	01:49
87b1273	Failed	DaveLiddament / ciricle_demo_project / step-05/add-parallel-lint #14	ADD lint	01:45
8112ec6	Failed	DaveLiddament / ciricle_demo_project / step-05/add-parallel-lint #13	ADD lint	01:41
f3c64fd	Failed	DaveLiddament / ciricle_demo_project / step-03/add-database-test #12		01:43
6559f82	Success	DaveLiddament / ciricle_demo_project / step-04/fix-database-test #10	FIX database test	02:04
de3e25b	Success	DaveLiddament / ciricle_demo_project / step-02/cache-composer-files #11		01:43
c24c918	Failed	DaveLiddament / ciricle_demo_project / step-3/add-database-test #8	ADD repository test	01:57

SETUP CIRCLE CI

The screenshot displays the CircleCI interface for a project named 'DaveLiddament/circle_demo'. The 'Builds' section is active, showing a list of build jobs. A red circle highlights the job 'step-08/add-cs-fixer #18', which has a 'SUCCESS' status. The interface includes a sidebar with navigation options like 'BUILDS', 'WORKFLOWS', 'INSIGHTS', 'PROJECTS', 'TEAM', 'USER SETTINGS', 'DOCS', 'SUPPORT', and 'WHAT'S NEW'. The main content area shows a table of build jobs with columns for job name, status, and completion time.

Job Name	Status	Completion Time
feature/circle_ci_2	SUCCESS	2 months ago
feature/foo	SUCCESS	2 months ago
master	SUCCESS	1 day ago
step-01/add-circle-ci-config	SUCCESS	2 months ago
step-02/cache-composer-files	FIXED	2 months ago
step-03/add-database-test	FAILED	2 months ago
step-04/fix-database-test	FAILED	2 months ago
step-05/add-parallel-lint	FAILED	2 months ago
step-06/more-lint	SUCCESS	2 months ago
step-07/add-var_dump-checker	SUCCESS	2 months ago
step-08/add-cs-fixer	SUCCESS	2 months ago

CIRCLE C1 V1 OR V2

CIRCLE.YML

CIRCLE BUILD STAGES

- ▶ machine
- ▶ dependencies
- ▶ database
- ▶ compile
- ▶ test
- ▶ deploy

STEPS WITHIN STAGE

- ▶ pre
- ▶ override
- ▶ post

MACHINE

machine:

timezone: Europe/London

php:

version: 5.6.22

environment:

COMPOSER_CACHE_DIR: ~/.cache/composer

MACHINE

machine:

timezone: Europe/London

 php:

 version: 5.6.22

 environment:

 COMPOSER_CACHE_DIR: ~/.cache/composer

MACHINE

machine:

 timezone: Europe/London

 php:

 version: 5.6.22

 environment:

 COMPOSER_CACHE_DIR: ~/.cache/composer

MACHINE

```
machine:  
  timezone: Europe/London  
  php:  
    version: 5.6.22  
environment:  
  COMPOSER_CACHE_DIR: ~/.cache/composer
```

DEPENDENCIES

dependencies:

cache_directories:

- "~/.cache/composer"

pre:

- echo "date.timezone = Europe/London" >> \
/opt/circleci/php/\$(phpenv global)/etc/php.ini

DEPENDENCIES

```
dependencies:
```

```
  cache_directories:
```

- "~/.cache/composer"

```
pre:
```

- echo "date.timezone = Europe/London" >> \
 /opt/circleci/php/\$(phpenv global)/etc/php.ini

DEPENDENCIES

```
dependencies:  
  cache_directories:  
    - "~/.cache/composer"  
  pre:  
    - echo "date.timezone = Europe/London" >> \  
      /opt/circleci/php/$(phpenv global)/etc/php.ini
```

DEPENDENCIES

```
dependencies:  
  cache_directories:  
    - "~/.cache/composer"  
  pre:  
    - echo "date.timezone = Europe/London" >> \  
      /opt/circleci/php/$(phpenv global)/etc/php.ini  
    - cp app/config/parameters.yml.circle \  
      app/config/parameters.yml
```

DATABASE

database:

override:

- mysql -u ubuntu -e "CREATE DATABASE testdb;"
- createdb testdb2
- psql -c "CREATE EXTENSION postgis;" testdb2
- app/console doctrine:schema:create --env=test

DATABASE

database:

 override:

- `mysql -u ubuntu -e "CREATE DATABASE testdb;"`
- `createdb testdb2`
- `psql -c "CREATE EXTENSION postgis;" testdb2`
- `app/console doctrine:schema:create --env=test`

DATABASE

```
database:  
  override:  
    - mysql -u ubuntu -e "CREATE DATABASE testdb;"  
    - createdb testdb2  
    - psql -c "CREATE EXTENSION postgis;" testdb2  
    - app/console doctrine:schema:create --env=test
```

DATABASE

```
database:  
  override:  
    - mysql -u ubuntu -e "CREATE DATABASE testdb;"  
    - createdb testdb2  
    - psql -c "CREATE EXTENSION postgis;" testdb2  
    - app/console doctrine:schema:create --env=test
```

DATABASE

```
database:  
  override:  
    - mysql -u ubuntu -e "CREATE DATABASE testdb;"  
    - createdb testdb2  
    - psql -c "CREATE EXTENSION postgis;" testdb2  
    - app/console doctrine:schema:create --env=test
```

COMPILE

`compile:`

`override:`

- `- bin/parallel-lint src`
- `- app/console lint:twig app`
- `- app/console lint:yaml app`
- `- bin/var-dump-check --no-colors src`
- `- bin/php-cs-fixer fix --dry-run src`

COMPILE

```
compile:
```

```
  override:
```

- bin/parallel-lint src
- app/console lint:twig app
- app/console lint:yaml app
- bin/var-dump-check --no-colors src
- bin/php-cs-fixer fix --dry-run src

TEST

test:

pre:

```
- mkdir -p $CIRCLE_TEST_REPORTS/phpunit
```

override:

```
- bin/phpunit -c app/phpunit.xml.dist \  
  --log-junit $CIRCLE_TEST_REPORTS/phpunit/junit.xml \  
  -d memory_limit=512M
```

TEST

test:

pre:

- mkdir -p \$CIRCLE_TEST_REPORTS/phpunit

override:

- bin/phpunit -c app/phpunit.xml.dist \
--log-junit \$CIRCLE_TEST_REPORTS/phpunit/junit.xml \
-d memory_limit=512M

TEST

test:

pre:

- mkdir -p \$CIRCLE_TEST_REPORTS/phpunit

override:

- bin/phpunit -c app/phpunit.xml.dist \
--log-junit \$CIRCLE_TEST_REPORTS/phpunit/junit.xml \
-d memory_limit=512M

TEST

test:

pre:

- mkdir -p \$CIRCLE_TEST_REPORTS/phpunit

override:

- bin/phpunit -c app/phpunit.xml.dist \
--log-junit \$CIRCLE_TEST_REPORTS/phpunit/junit.xml \
-d memory_limit=512M

DEPLOY

```
deployment:  
  live:  
    branch: master  
    commands:  
    - ssh user@mywebsite.com \  
      "bash --login -c 'deploy "$CIRCLE_SHA1"'"
```

DEPLOY

```
deployment:  
  live:  
    branch: master  
    commands:  
    - ssh user@mywebsite.com \  
      "bash --login -c 'deploy "$CIRCLE_SHA1"'"
```

DEPLOY

```
deployment:  
  live:  
    branch: master  
    commands:  
    - ssh user@mywebsite.com \  
      "bash --login -c 'deploy "$CIRCLE_SHA1"'"
```

DEPLOY

```
deployment:  
  live:  
    branch: master  
    commands:  
    - ssh user@mywebsite.com \  
      "bash --login -c 'deploy "$CIRCLE_SHA1"'"
```

DEPLOY

```
deployment:  
  live:  
    branch: master  
    commands:  
    - ssh user@mywebsite.com \  
      "bash --login -c 'deploy "$CIRCLE_SHA1"'"
```

DEPLOY

```
deployment:  
  live:  
    branch: master  
    commands:  
    - ssh user@mywebsite.com \  
      "bash --login -c 'deploy "$CIRCLE_SHA1"'"
```

WHEN YOU TRY THIS AT HOME

TIPS

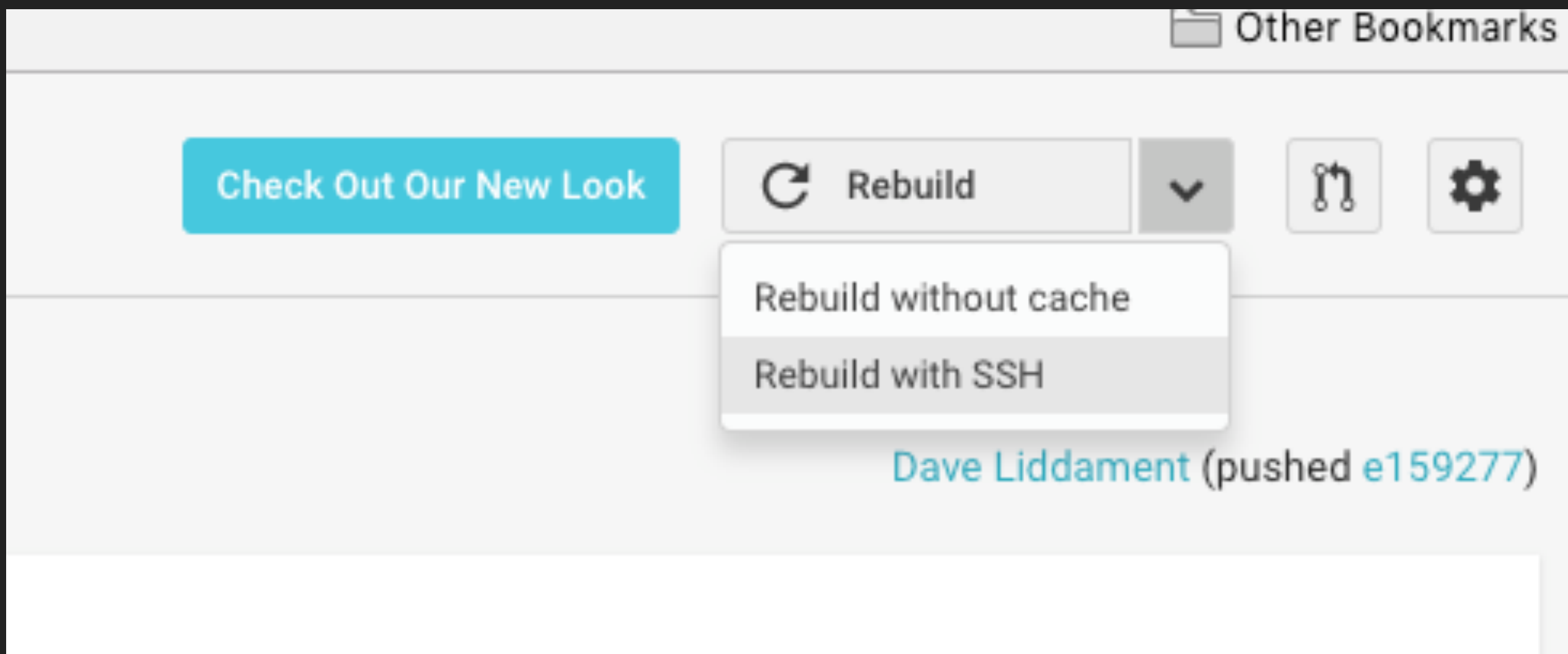
TIPS

- ▶ https://github.com/DaveLiddament/circle_demo_project
 - ▶ Branches called step-01, step-02, etc

TIPS

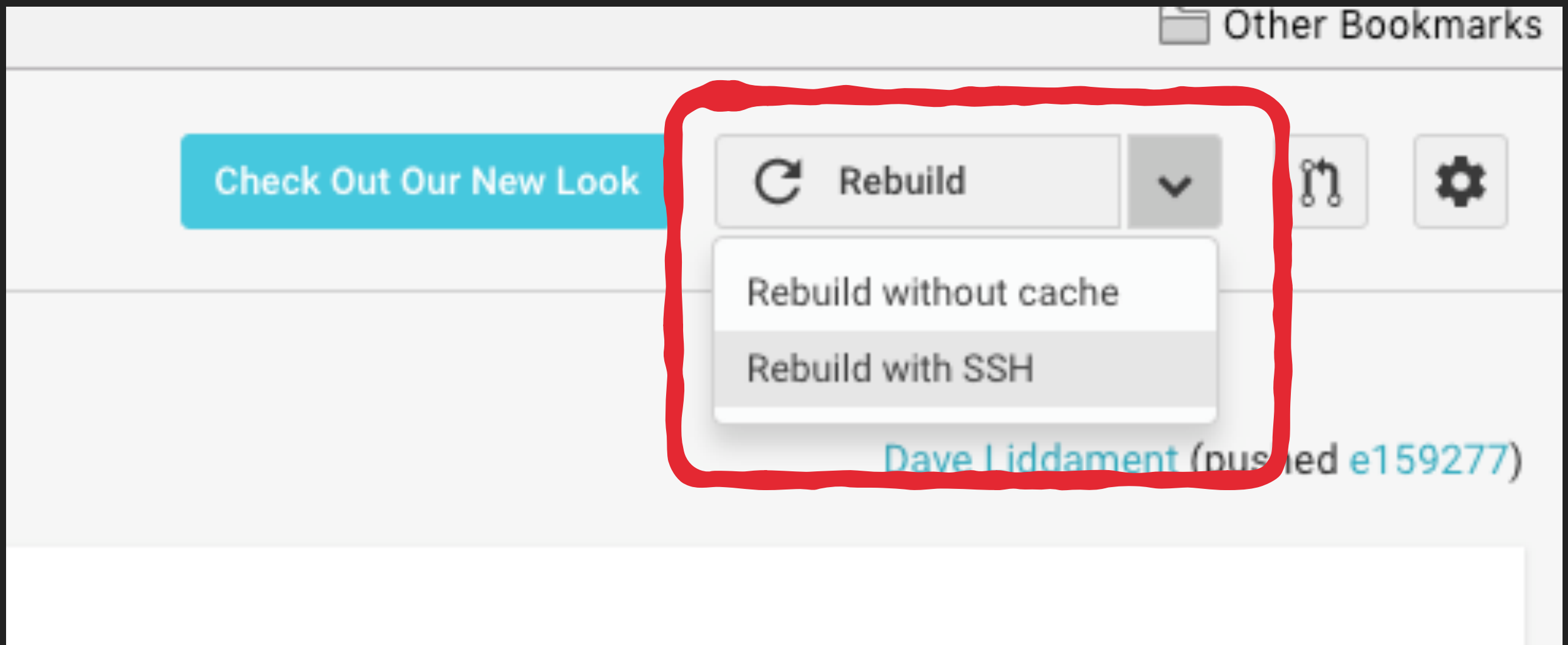
- ▶ https://github.com/DaveLiddament/circle_demo_project
 - ▶ Branches called step-01, step-02, etc
- ▶ Build things up in steps.
 - ▶ Run only unit tests first
 - ▶ Then get integration tests running

TIPS



WHEN YOU TRY THIS AT HOME

TIPS



SUMMARY

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

QUESTIONS