



WiFi:

Sheraton Conference



Pass: phptek2018



Twitter:

#phptek

Rate the Talks

<https://joind.in/event/phptek-2018>



PHP[TEK] 2018

EFFECTIVE CODE REVIEW



DAVE LIDDAMENT

EFFECTIVE CODE REVIEW REDUCES OVERALL COST OF SOFTWARE DEVELOPMENT



Thanks to Our Sponsors

AUTOMATIC



RED HAT®
OPENSHIFT

RingCentral®

MailChimp®

Square



Magento®

AGENDA

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

Dave Liddament

Lamp Bristol

Organize:

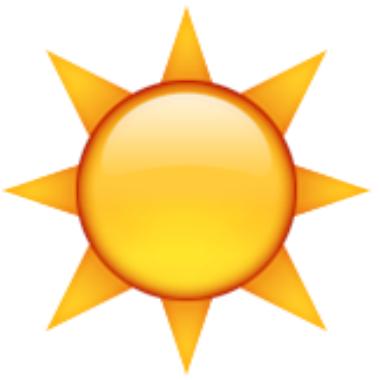
- PHP-SW user group
- Bristol PHP Training

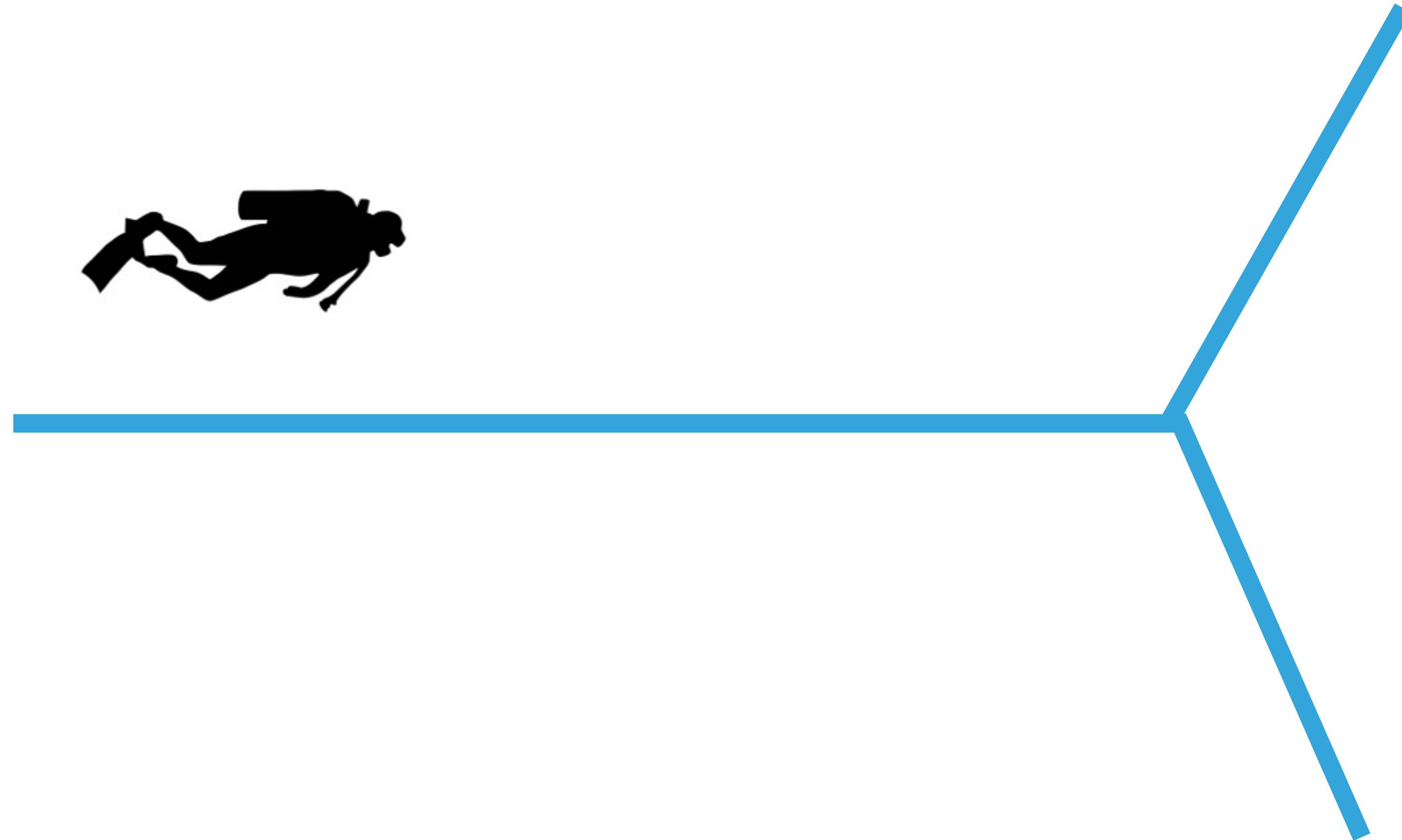
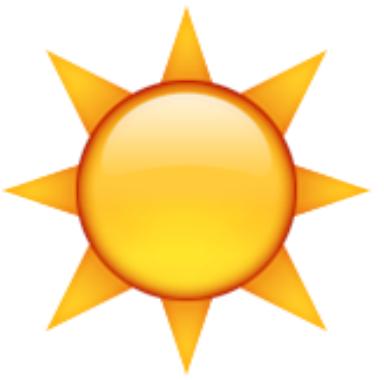
@daveliddament

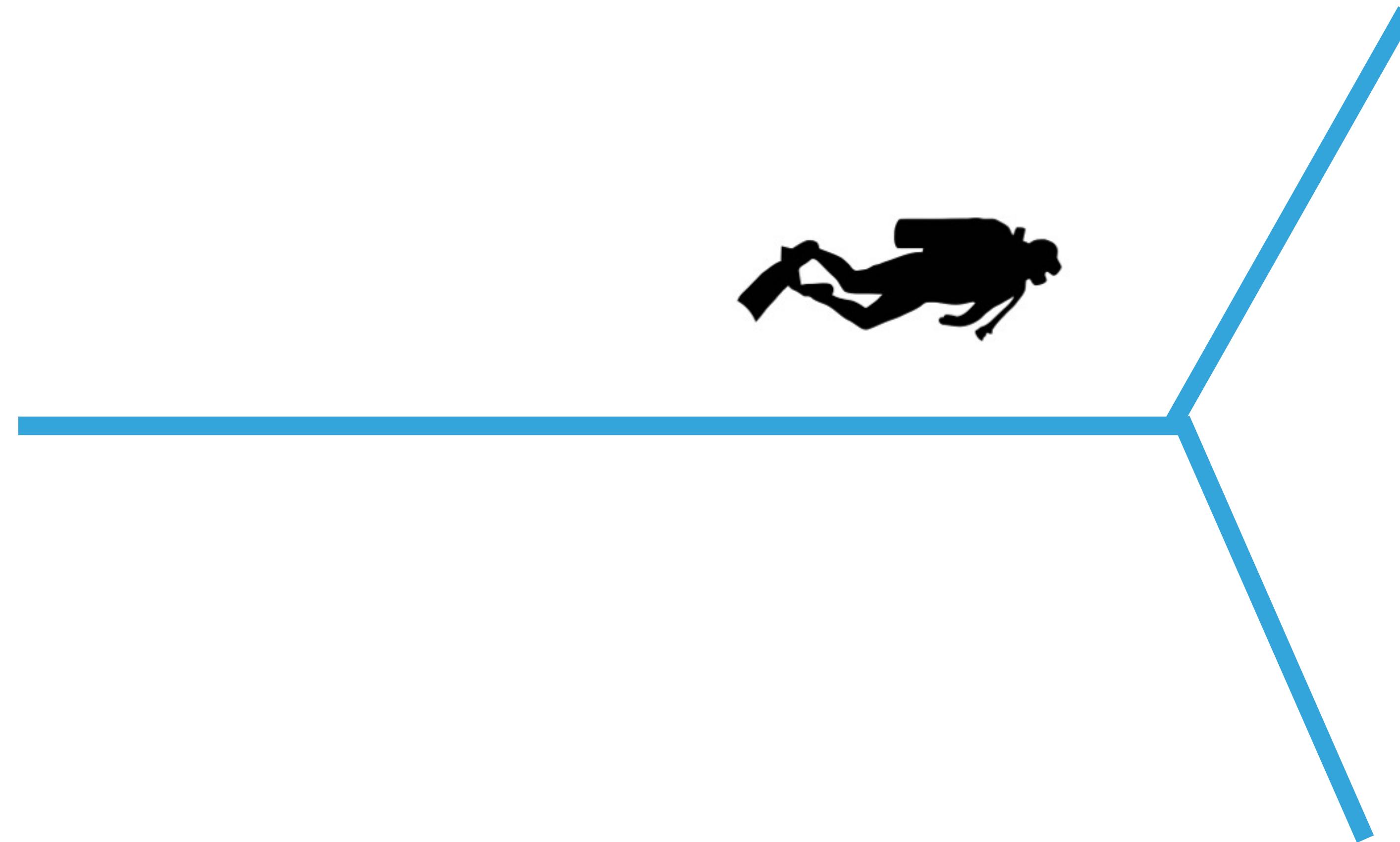
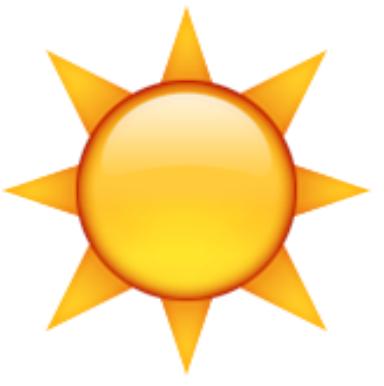


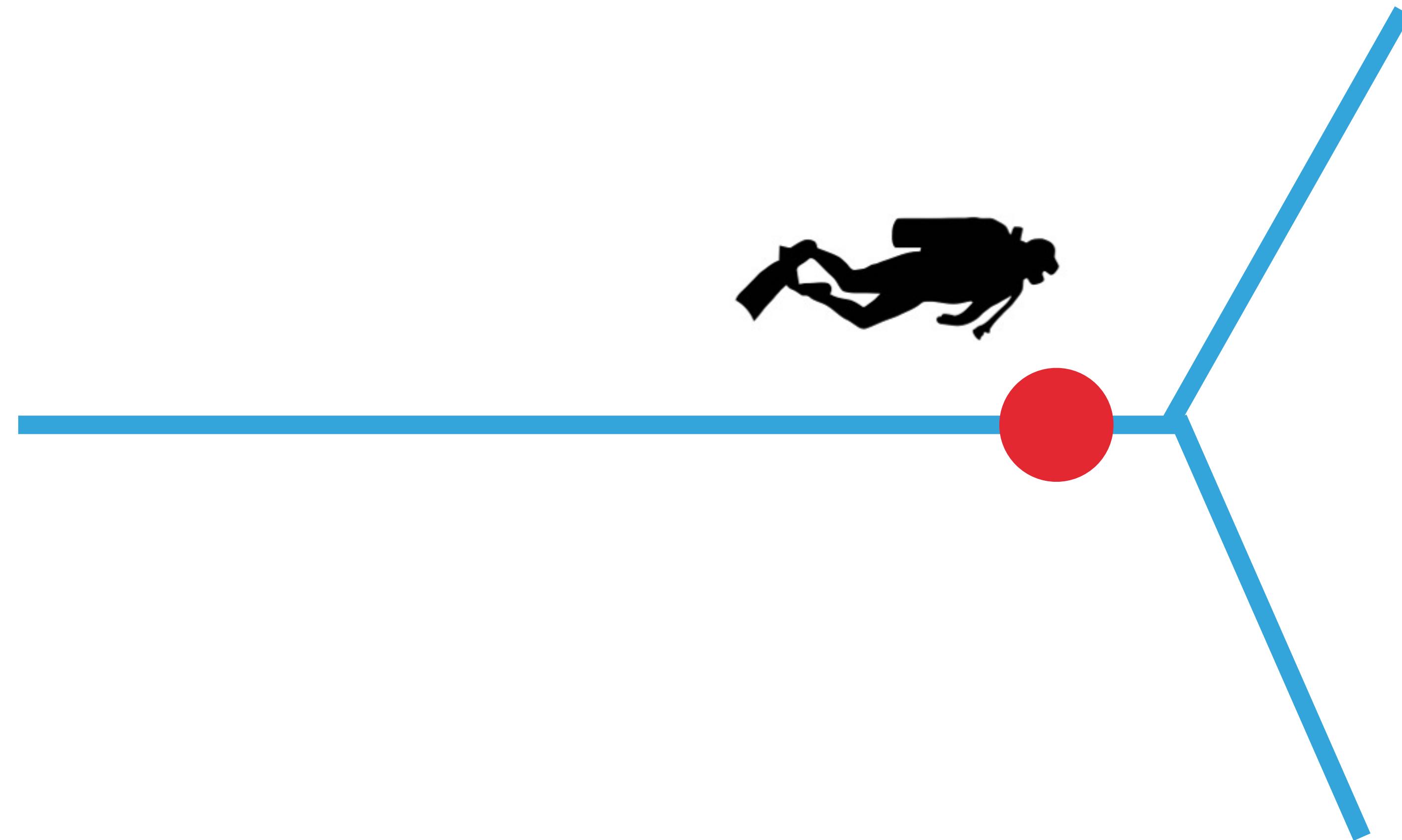
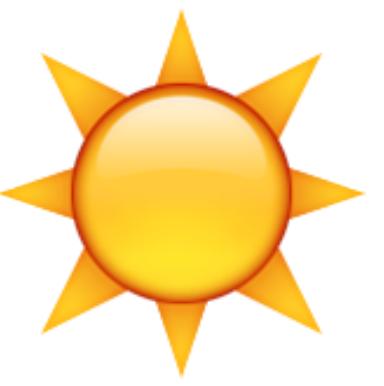


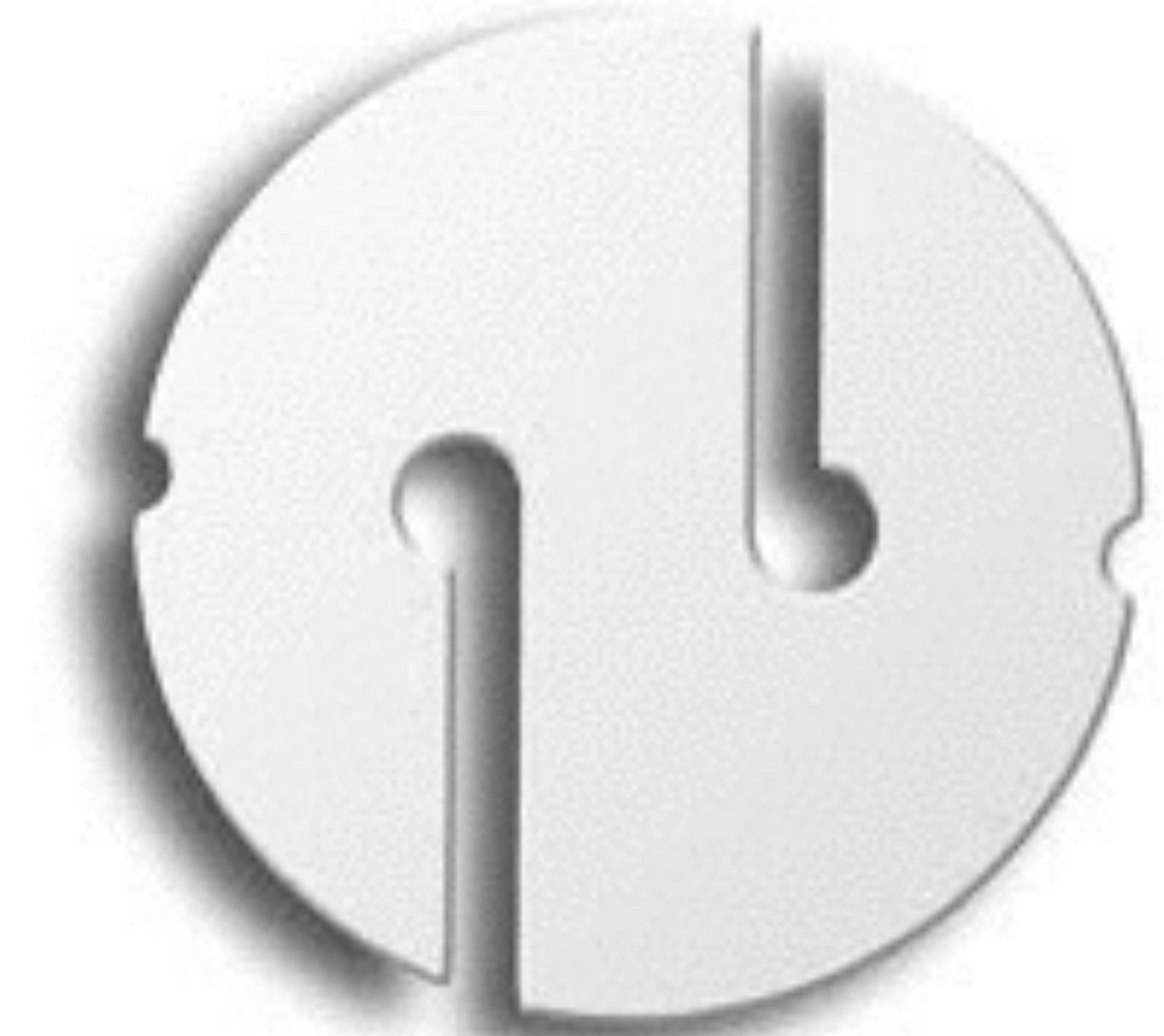


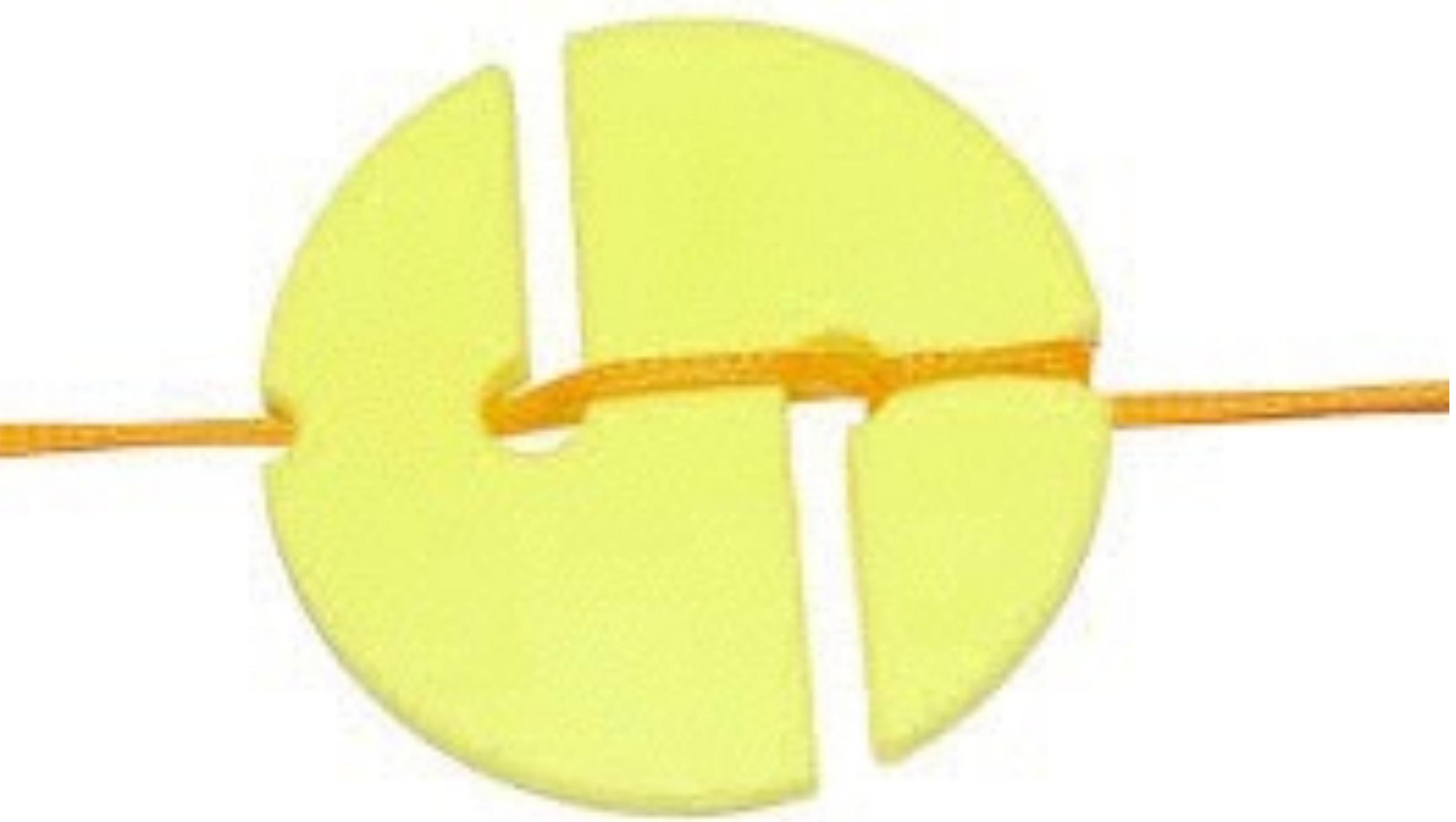
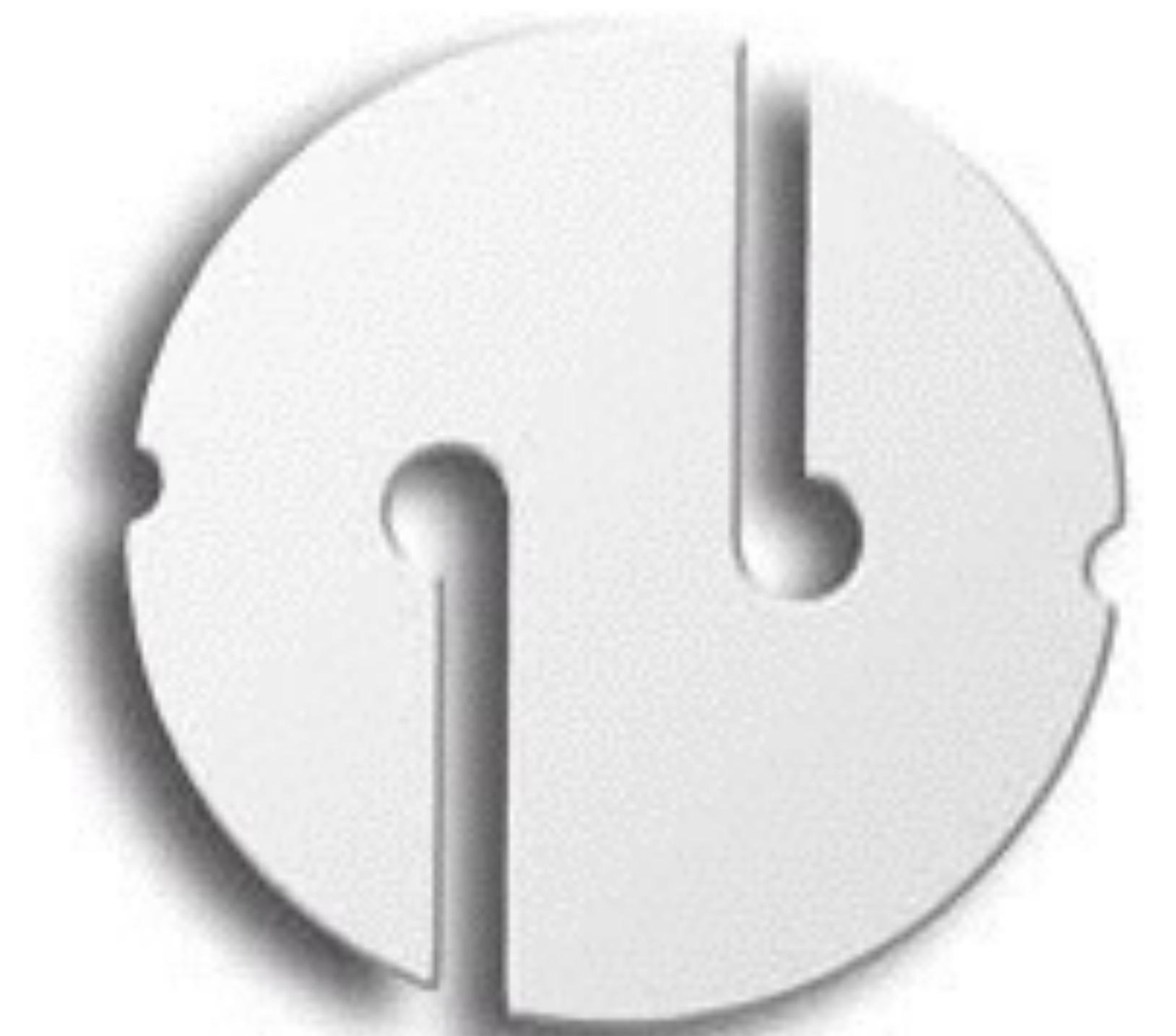


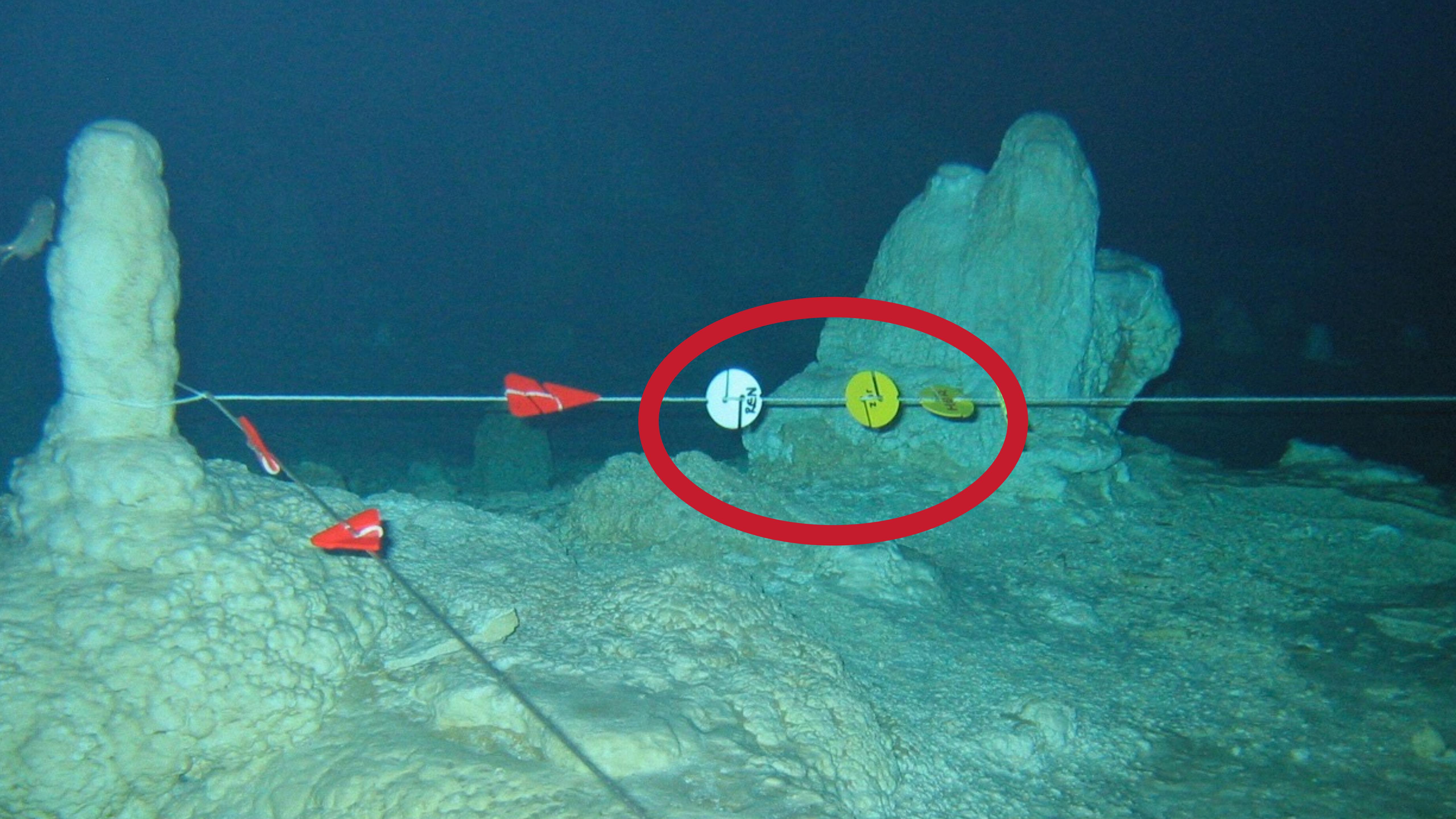


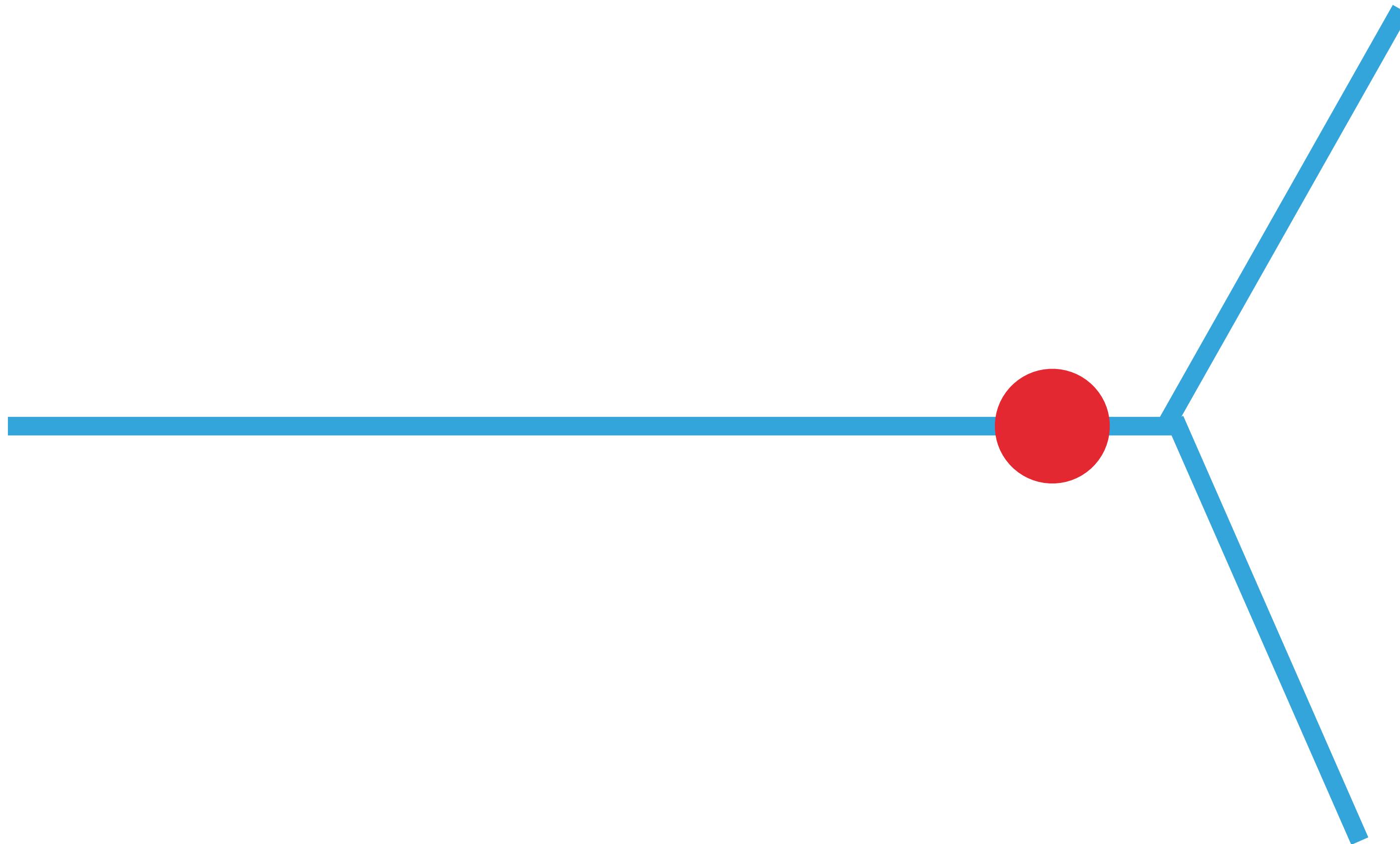


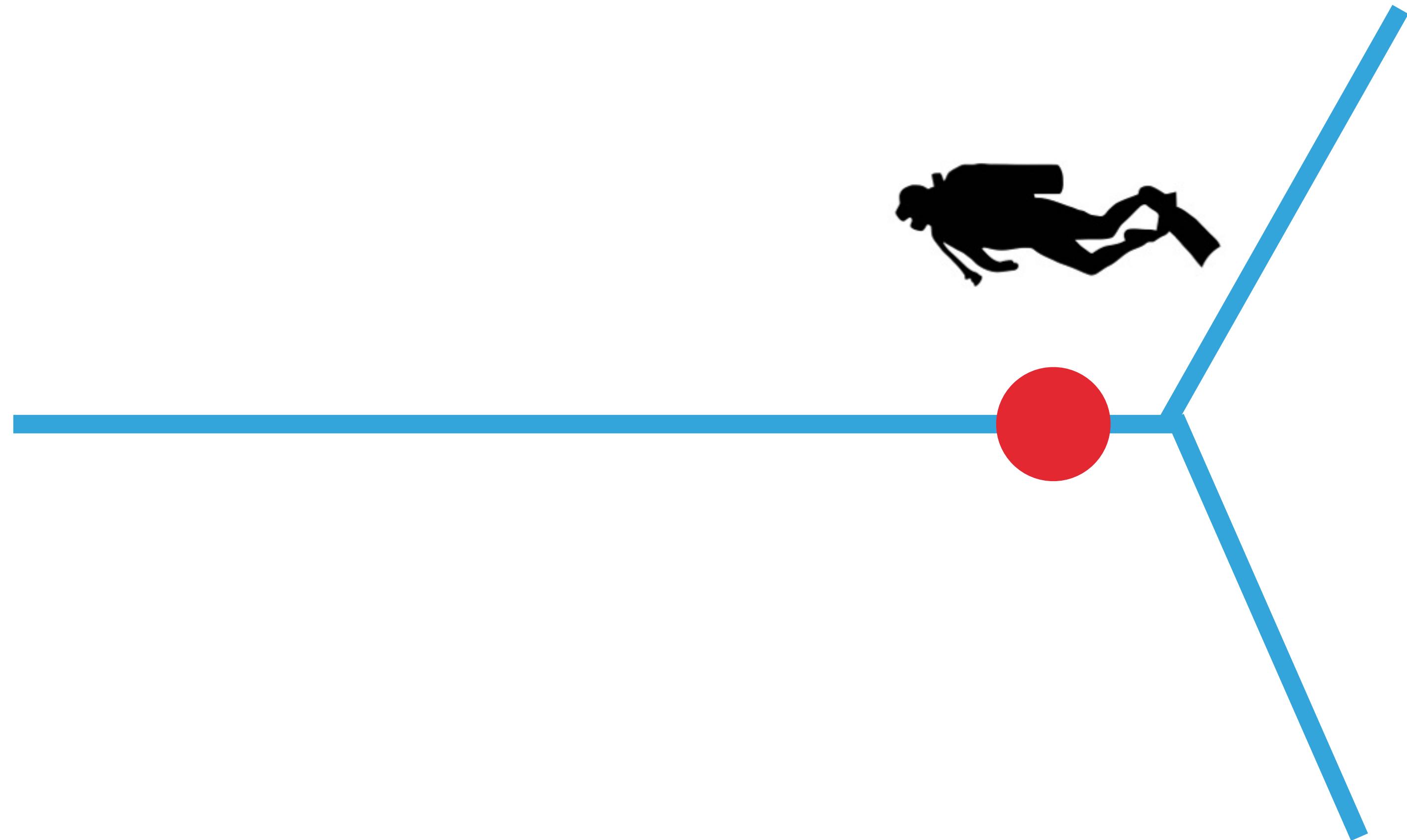


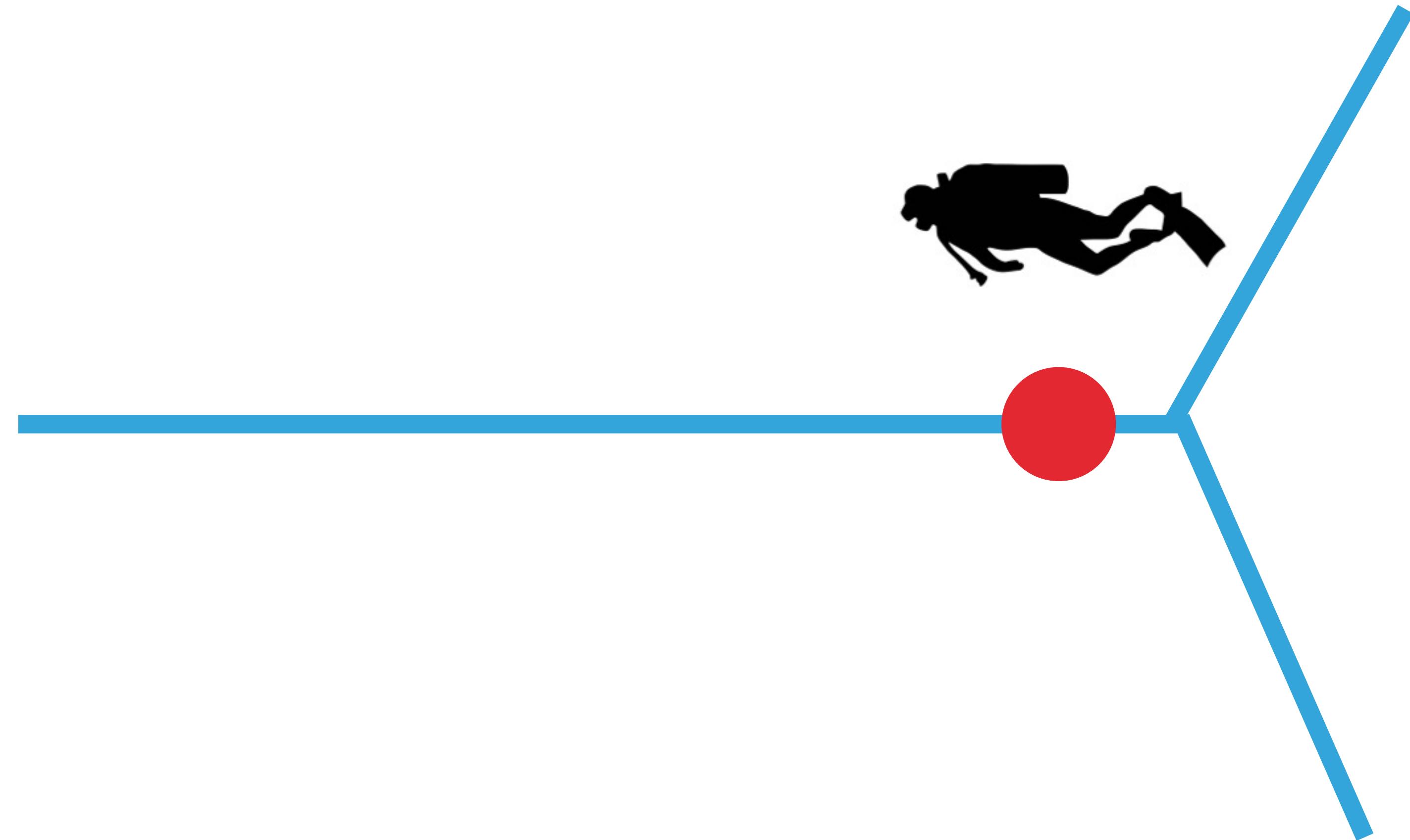
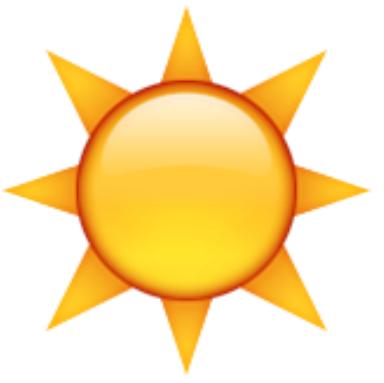


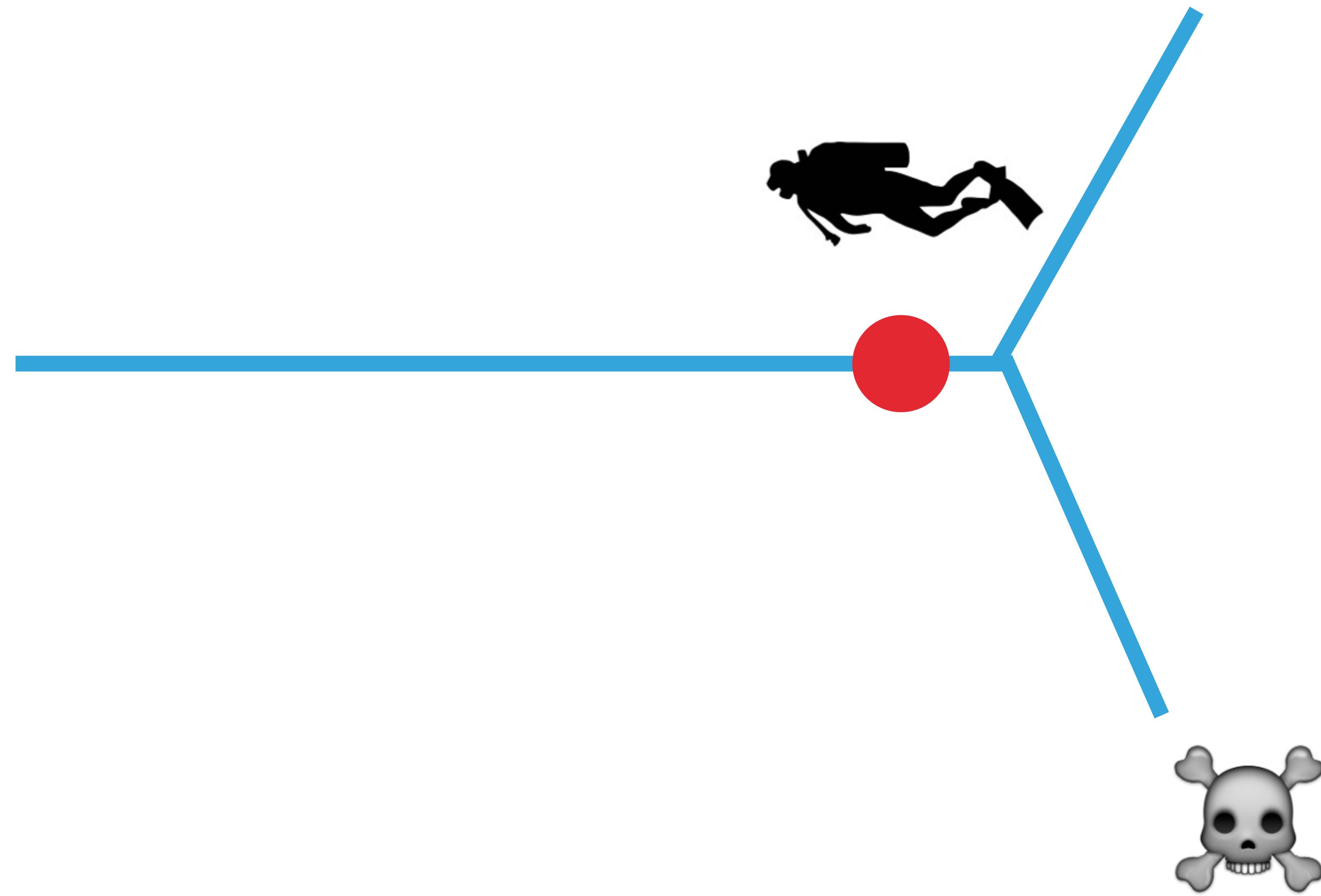
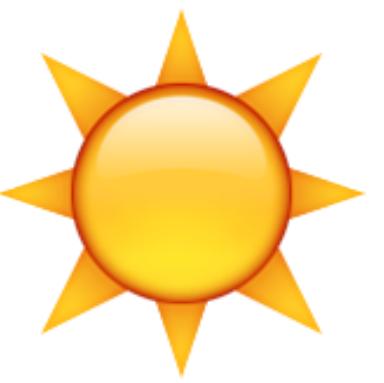


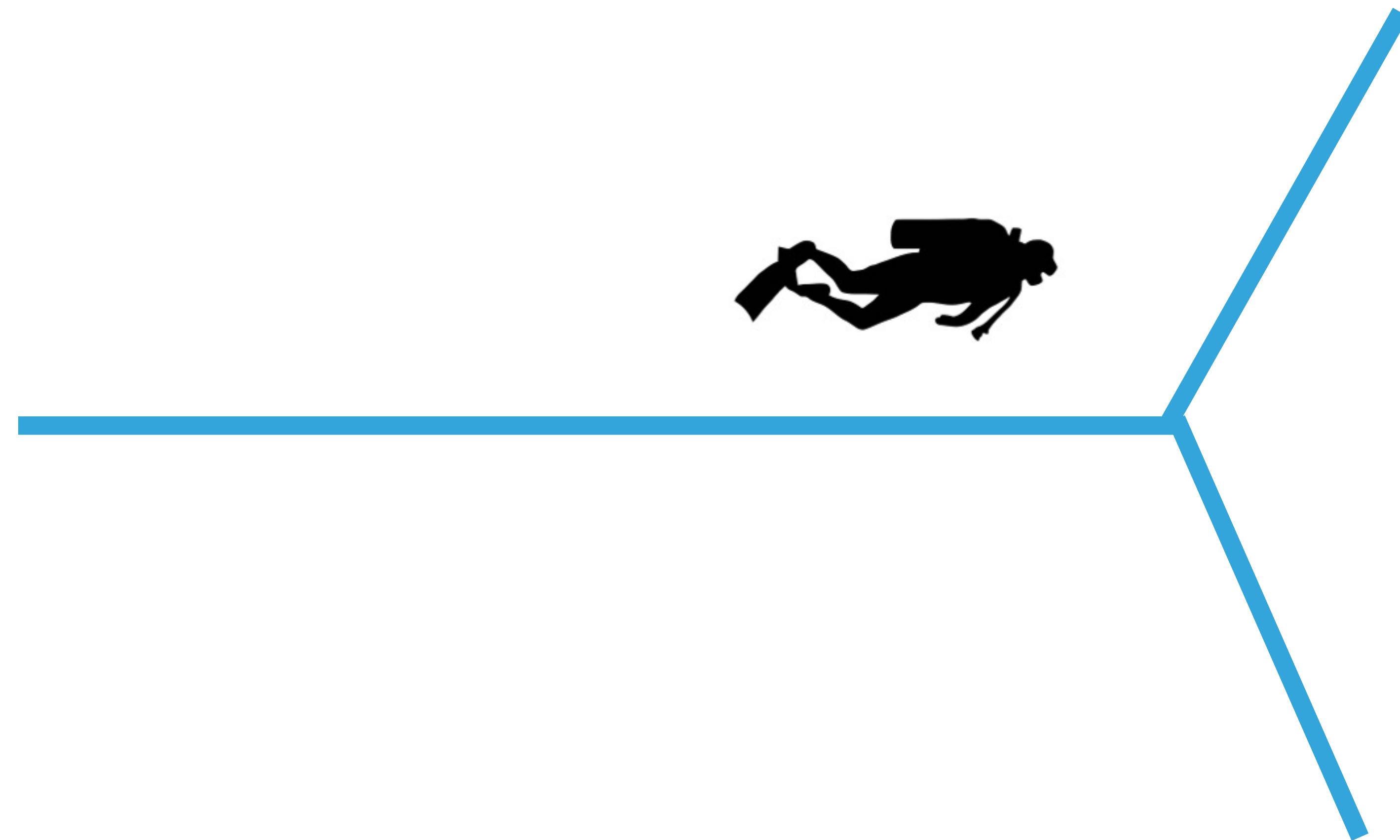
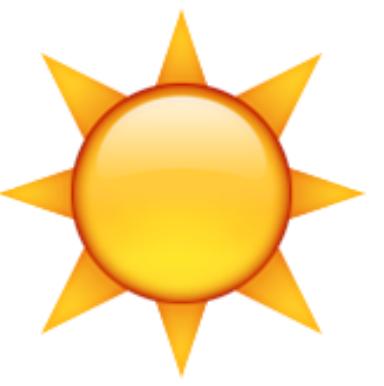


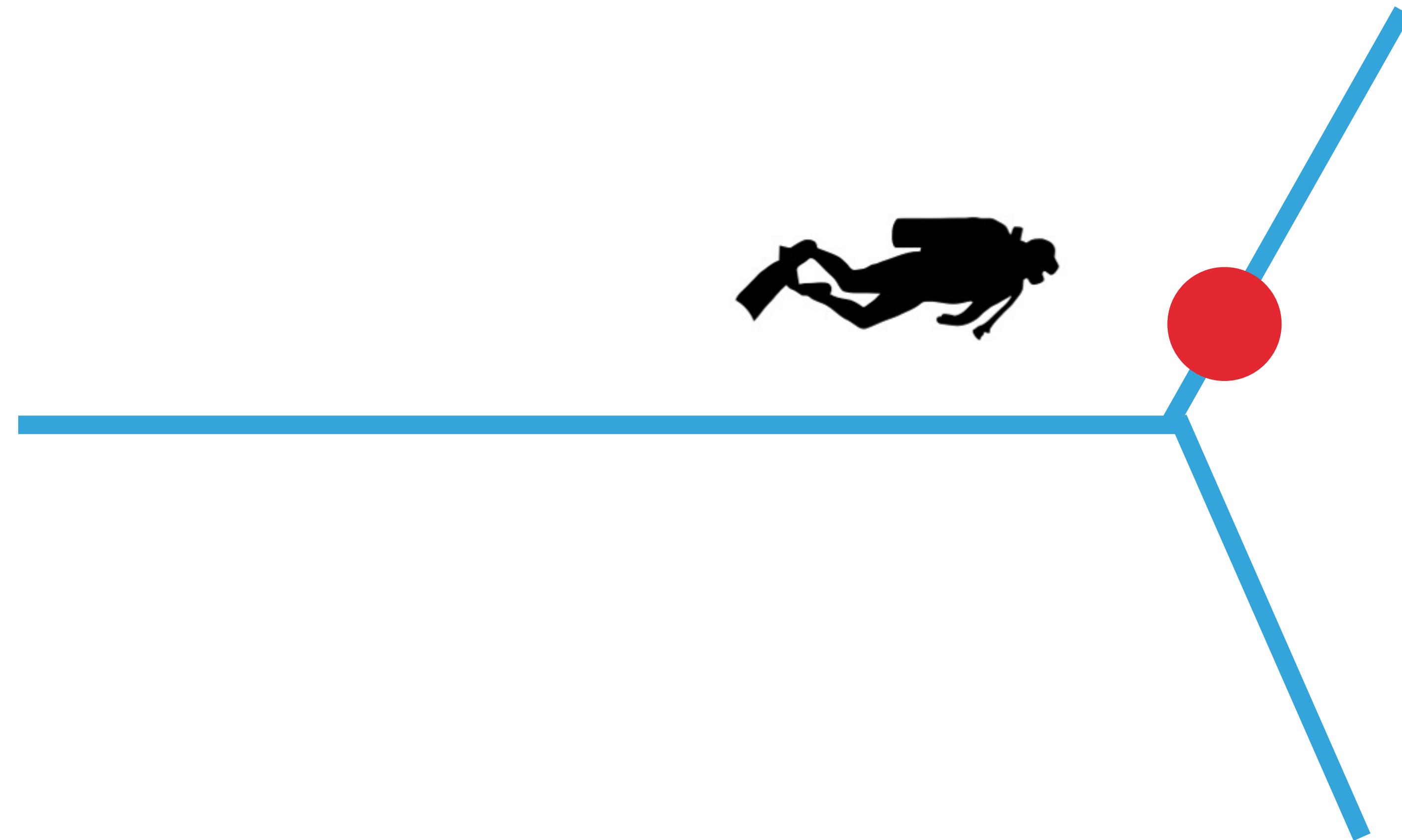
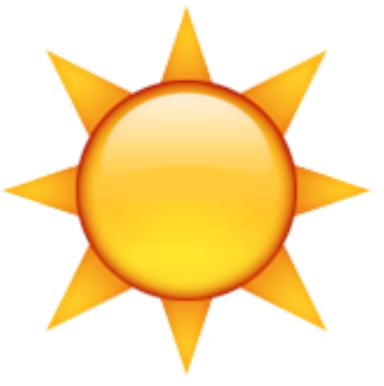










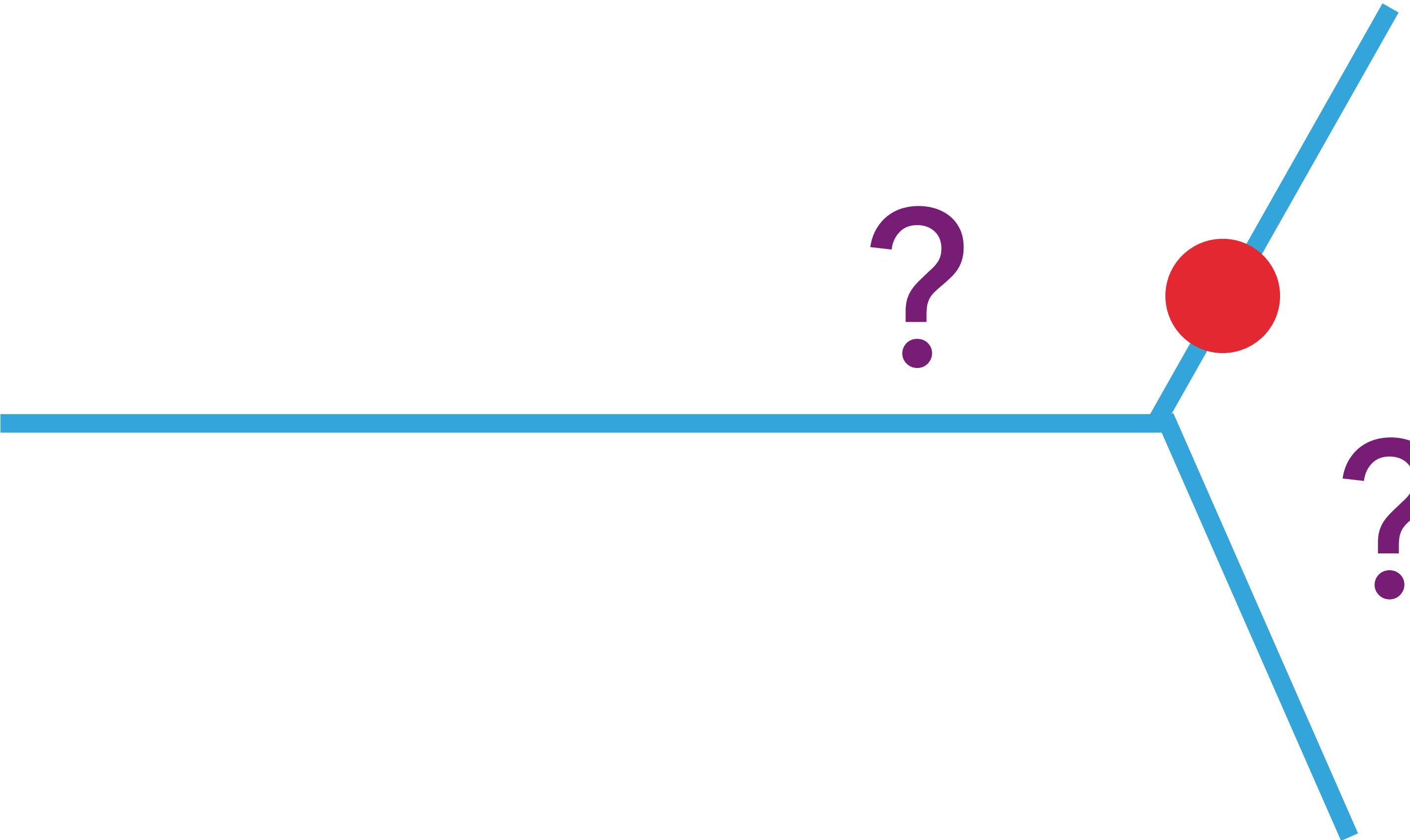


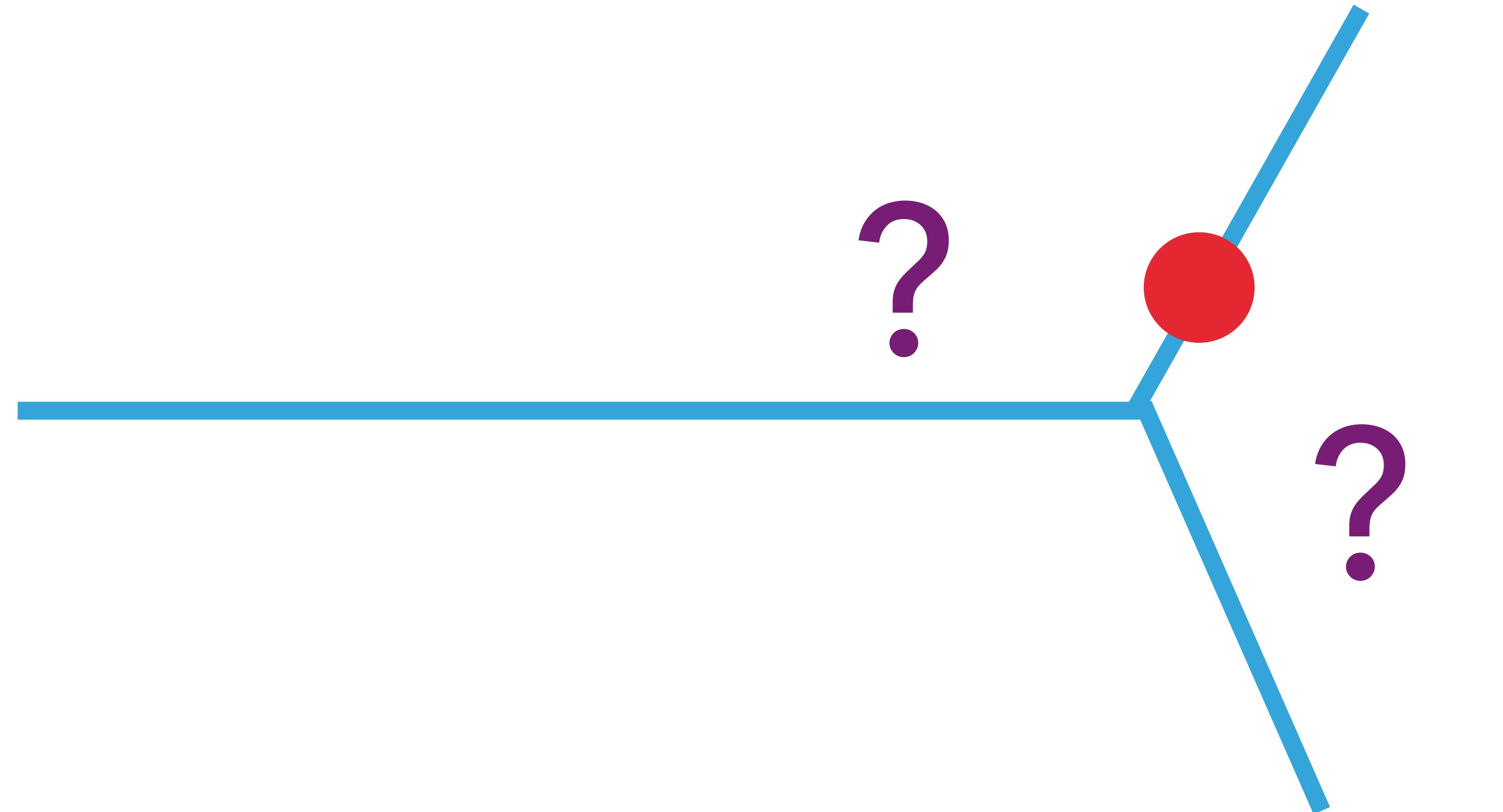
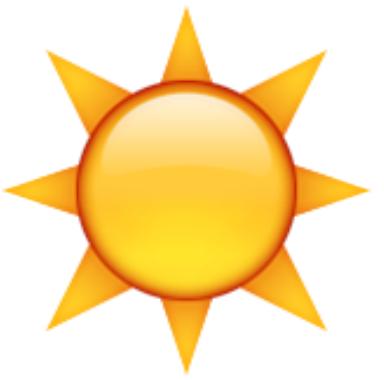


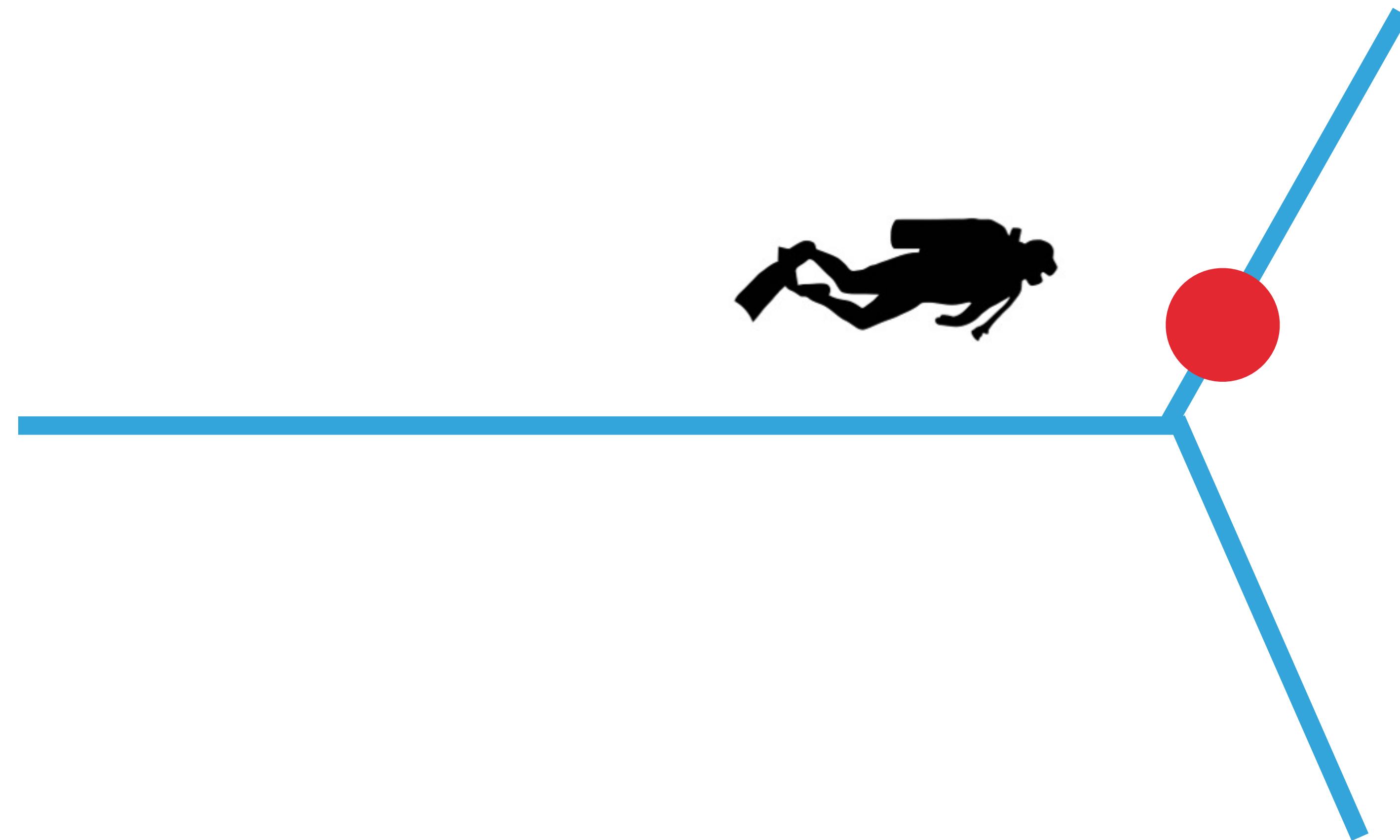
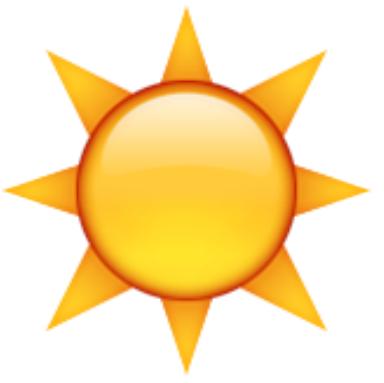


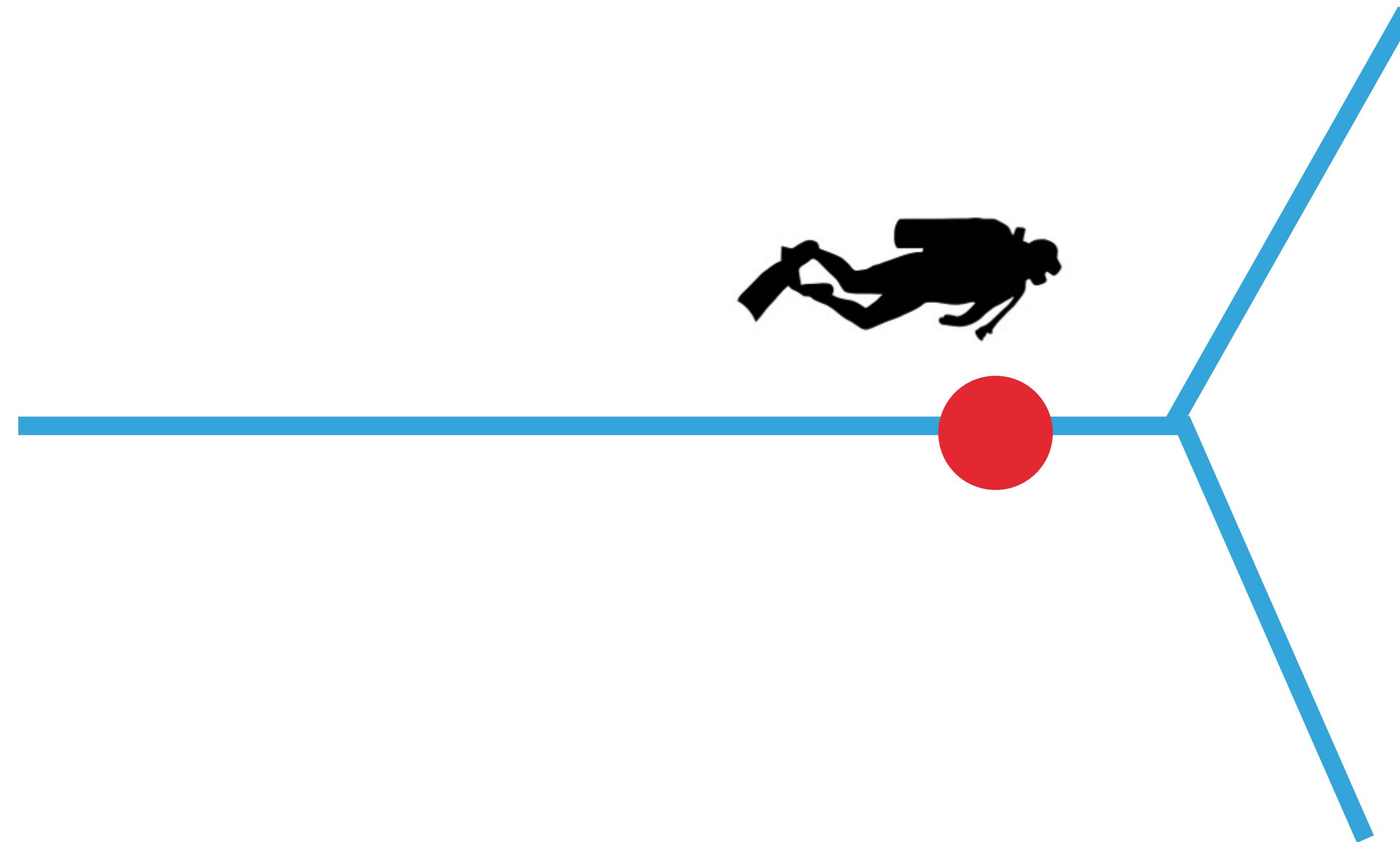
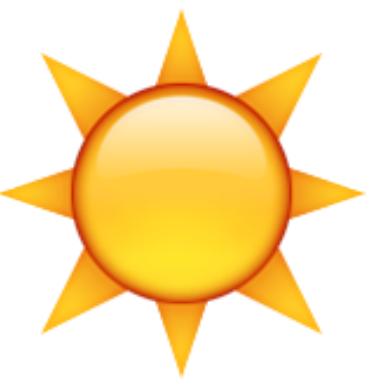
?

?









MORAL OF THE STORY

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.

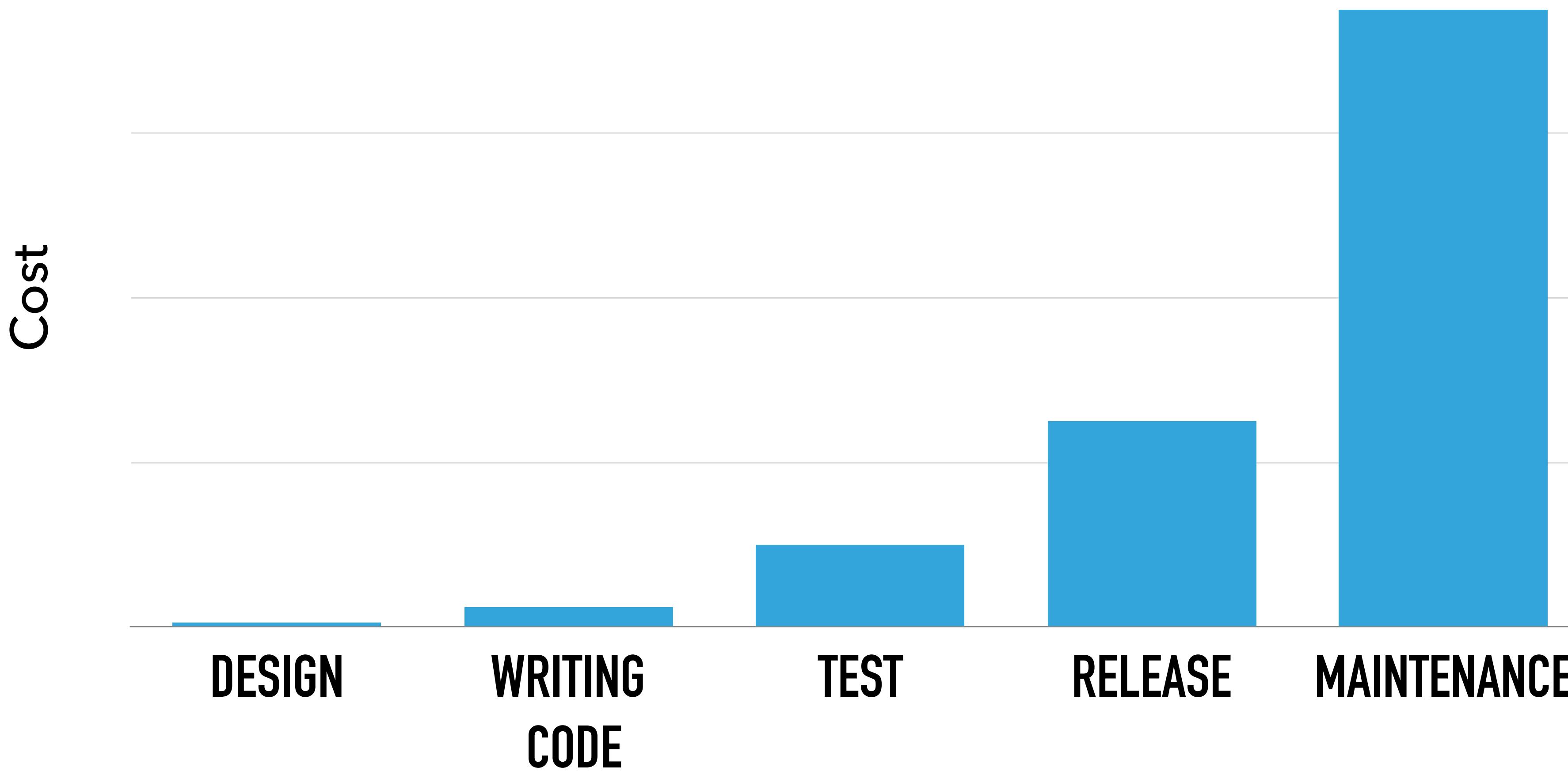
WHAT ABOUT SOFTWARE DEVELOPMENT?

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.

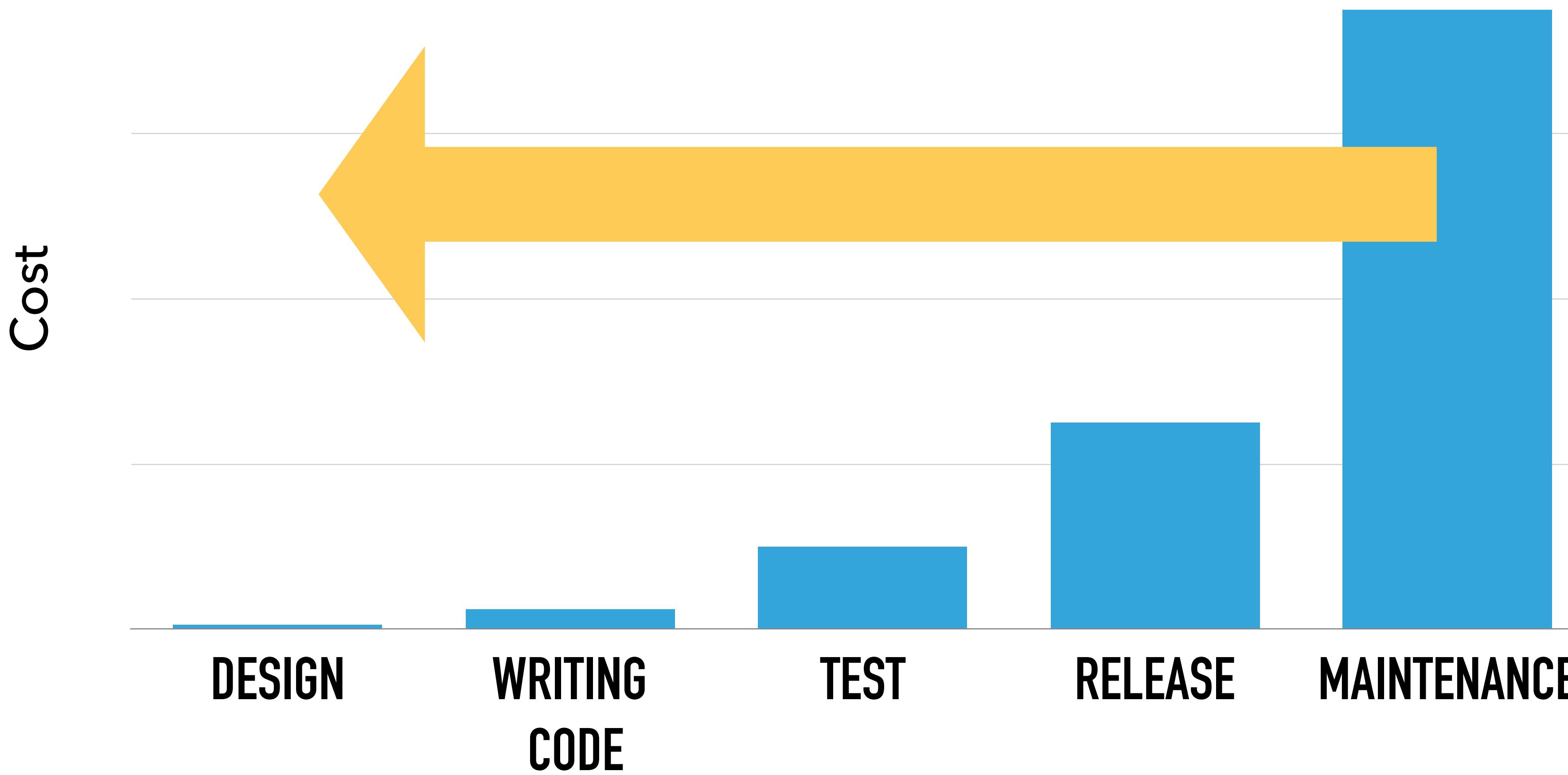
COST OF A BUG

FIND BUGS SOONER



COST OF A BUG

FIND BUGS SOONER



**EFFECTIVE CODE REVIEW
REDUCES OVERALL COST OF
SOFTWARE DEVELOPMENT**

EFFECTIVE CODE REVIEW REQUIRES...

EFFECTIVE CODE REVIEW REQUIRES...



EFFECTIVE CODE REVIEW REQUIRES...



PHPUnit



**Check
Style**



Psalm

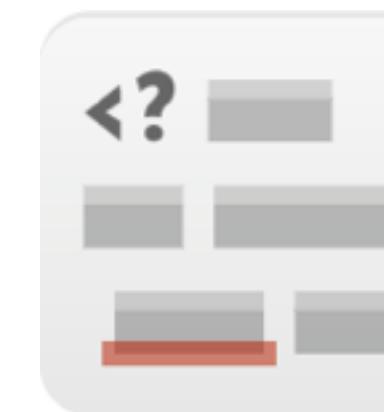
EFFECTIVE CODE REVIEW REQUIRES...



PHPUnit



**Check
Style**



Psalm



 **circleci**



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

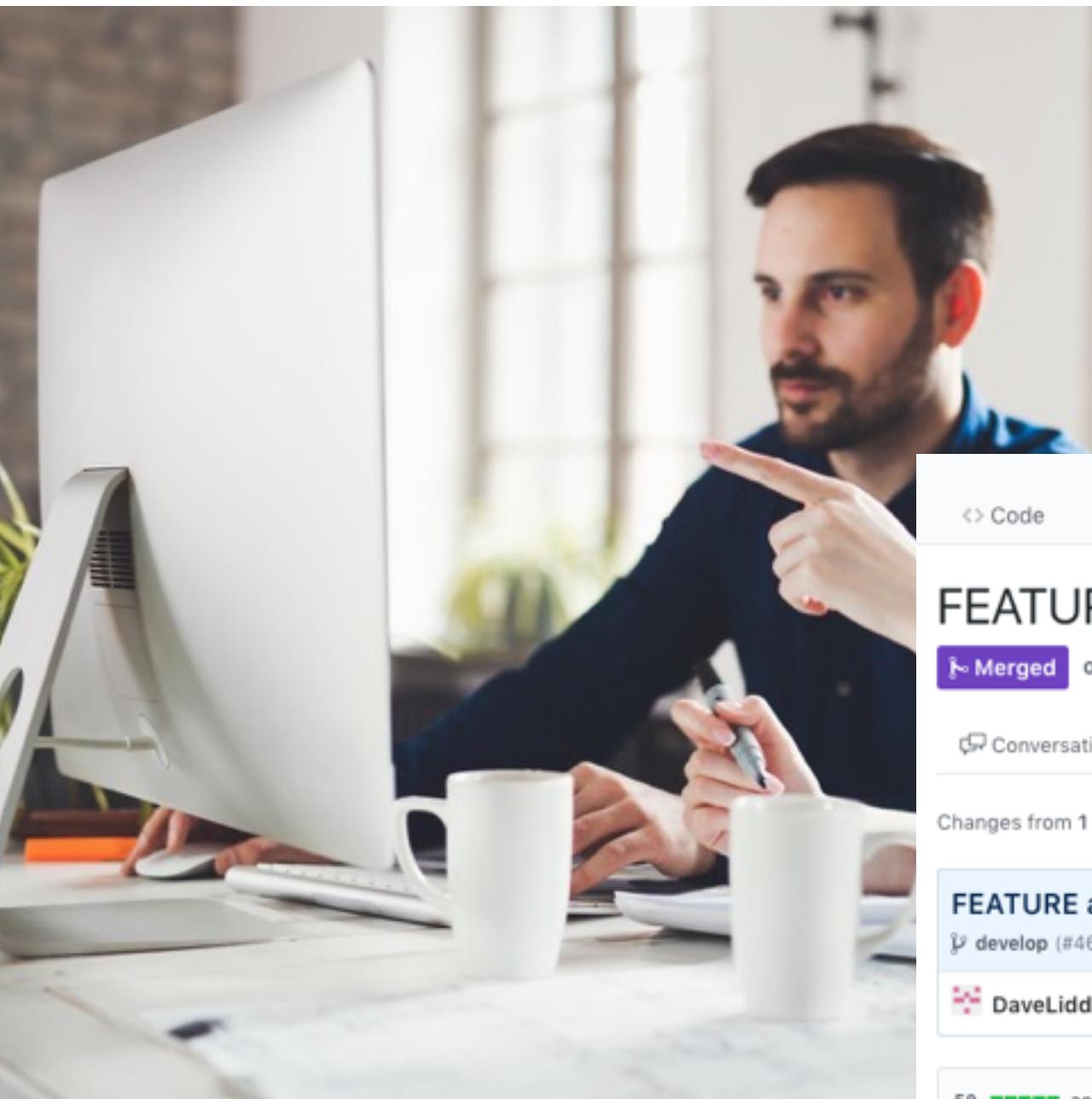
WHAT IS CODE REVIEW

HOW IS IT DONE



WHAT IS CODE REVIEW

HOW IS IT DONE



A photograph of two people, a man and a woman, sitting at a desk and looking at a computer monitor. The woman is pointing at the screen, which displays a GitHub pull request interface.

The GitHub pull request interface shows:

- Title: FEATURE add sponsor type to sponsor #46
- Merged by opdavies on Feb 26
- Conversation: 0
- Commits: 2
- Files changed: 15
- Changes from 1 commit
- Jump to... +116 -18
- Unified, Split, Review changes buttons
- Commit details:
 - FEATURE add sponsor type to sponsor (develop #46)
 - DaveLiddament committed on Feb 26
 - commit 86070f4c08780c8a167bef2b44e09a00609915d6
- Code diff view showing changes in Sponsor.php

```
59 @@ -6,6 +6,16 @@
6
7 class Sponsor
8 {
9 /**
10 * Full sponsor.
11 */
12 const SPONSOR_FULL = 'full';
13 /**
14 * Sponsor only covers occasional events.
15 */
16 const SPONSOR_EVENT = 'event';
17
18 /**
19 * @var string
20 */
21 *
```

CODE REVIEW BENEFITS

WHAT ARE DEFECTS?

WHAT ARE DEFECTS?

Bugs

WHAT ARE DEFECTS?

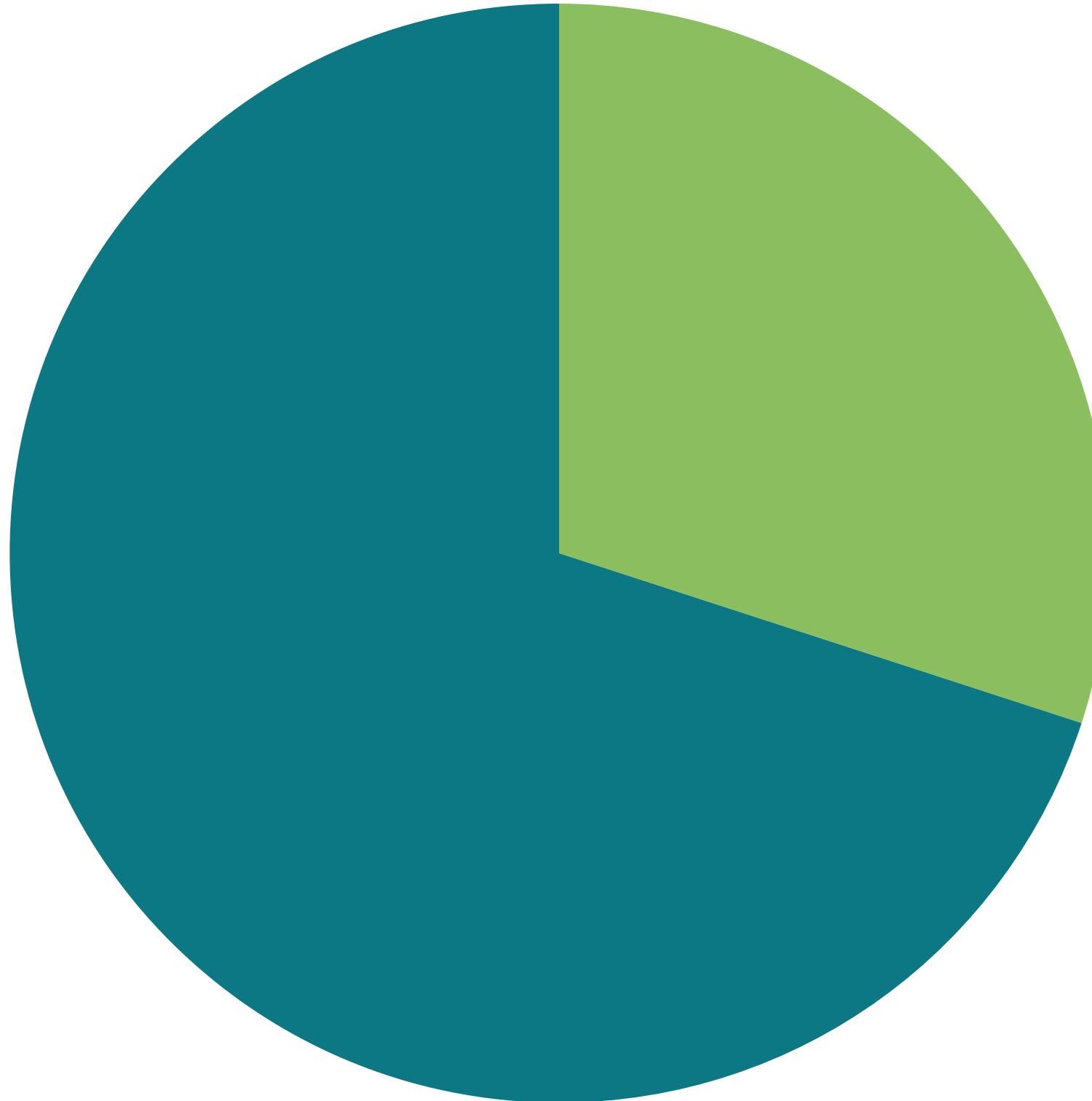
Bugs

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

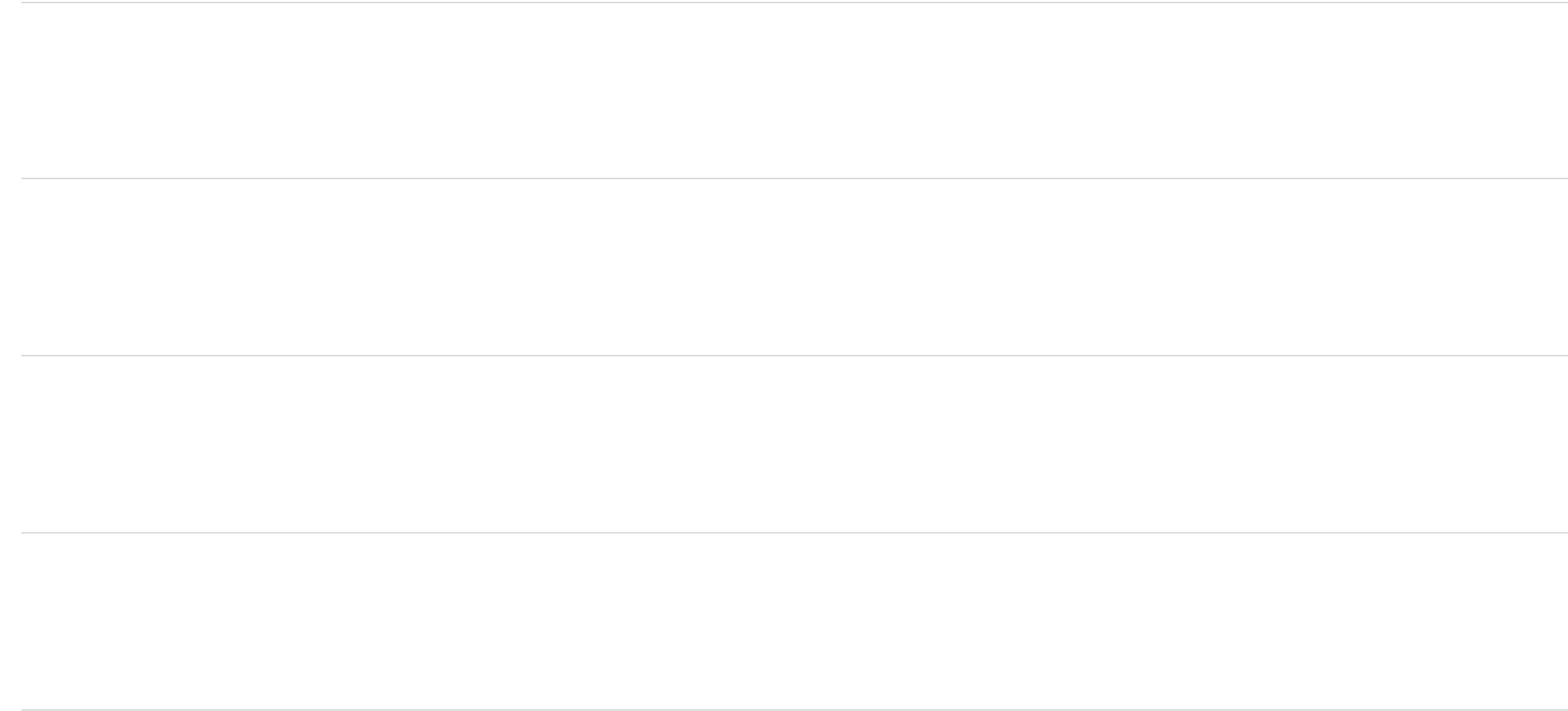
- ▶ Low evolvability costs money:
 - ▶ New features took 28% longer to implement [3]
 - ▶ Fixing bugs took 36% longer [3]
- ▶ Software structure may account for 25% of total maintenance costs [4]

COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME

CODE REVIEW BENEFITS

COST TO DEVELOP SIMILAR SIZED FEATURE OVER TIME

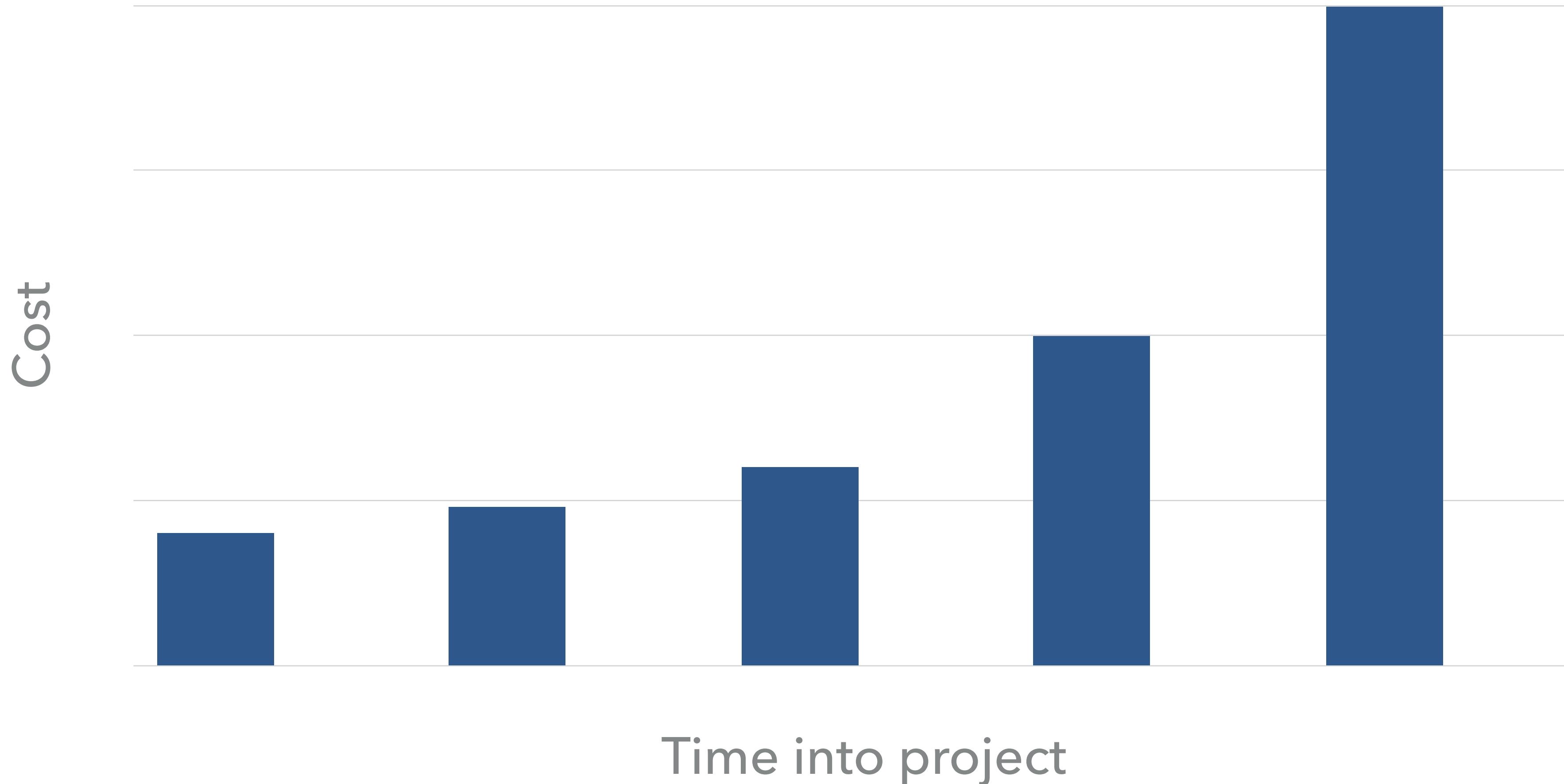
Cost



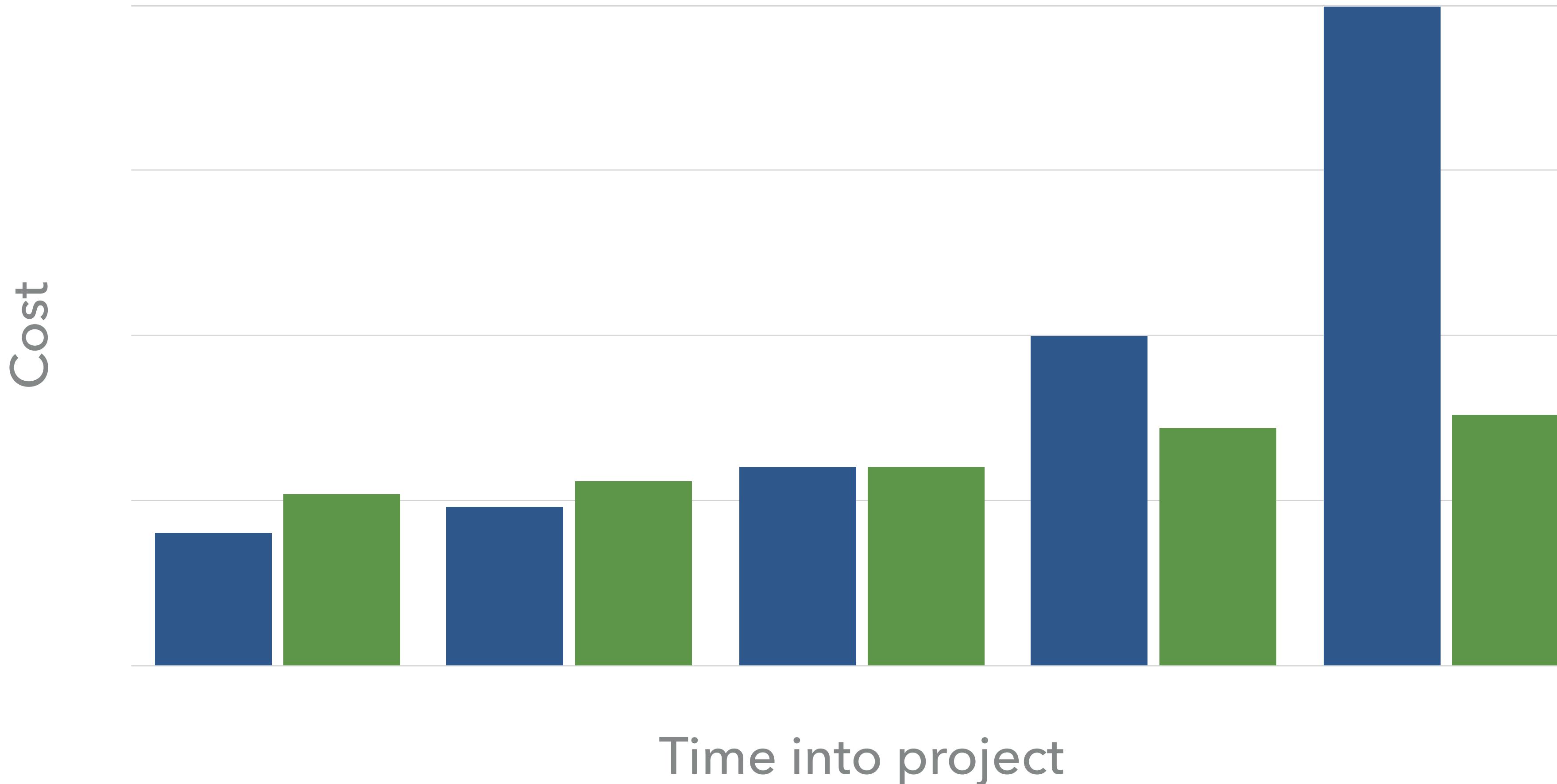
A blank graph illustrating the cost of developing similar-sized features over time. The vertical axis is labeled 'Cost' and the horizontal axis is labeled 'Time into project'. Four horizontal lines are drawn across the graph, representing different projects or feature sizes.

Time into project

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”.
- ▶ 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

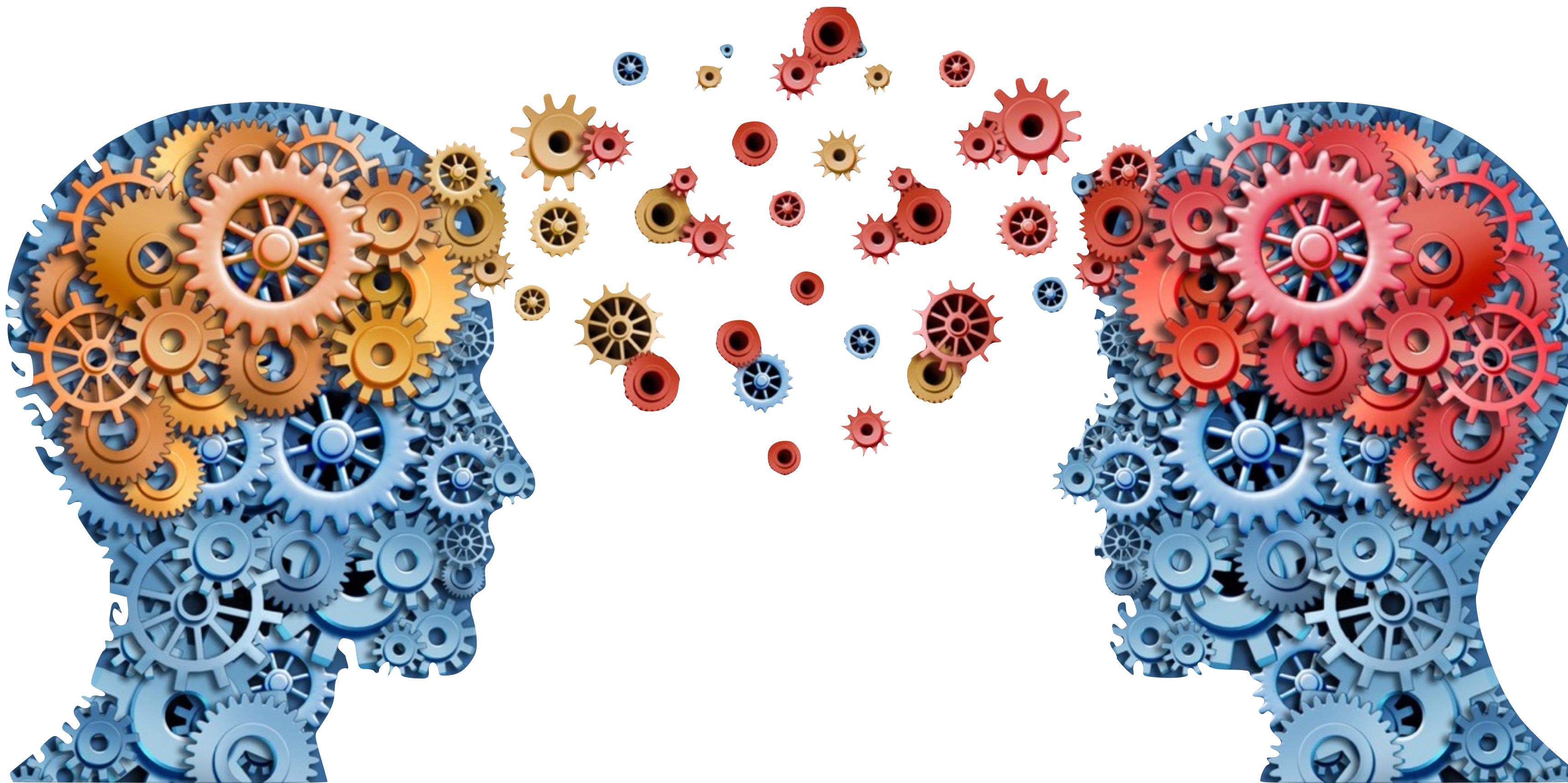
SECURITY REVIEW

- ▶ 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



CODE REVIEW BENEFITS

NO MORE SILOS



CODE REVIEW BENEFITS

NO MORE SILOS



CODE REVIEW BENEFITS

NO MORE SILOS



CODE REVIEW BENEFITS

NO MORE SILOS



CODE REVIEW BENEFITS

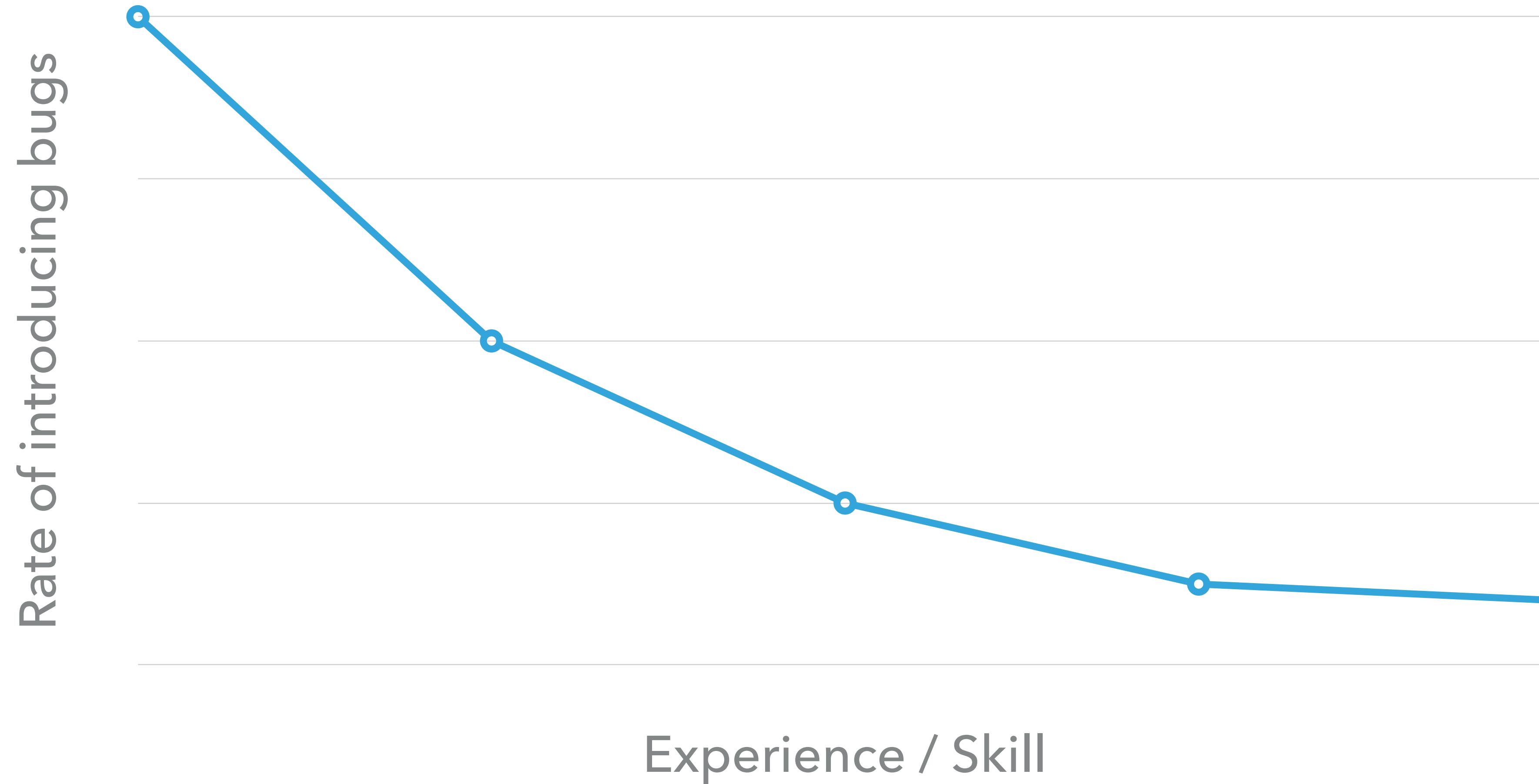
NO MORE SILOS



MENTORING



SKILL AND EXPERIENCE VS BUGS



CODE REVIEW BENEFITS

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**

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

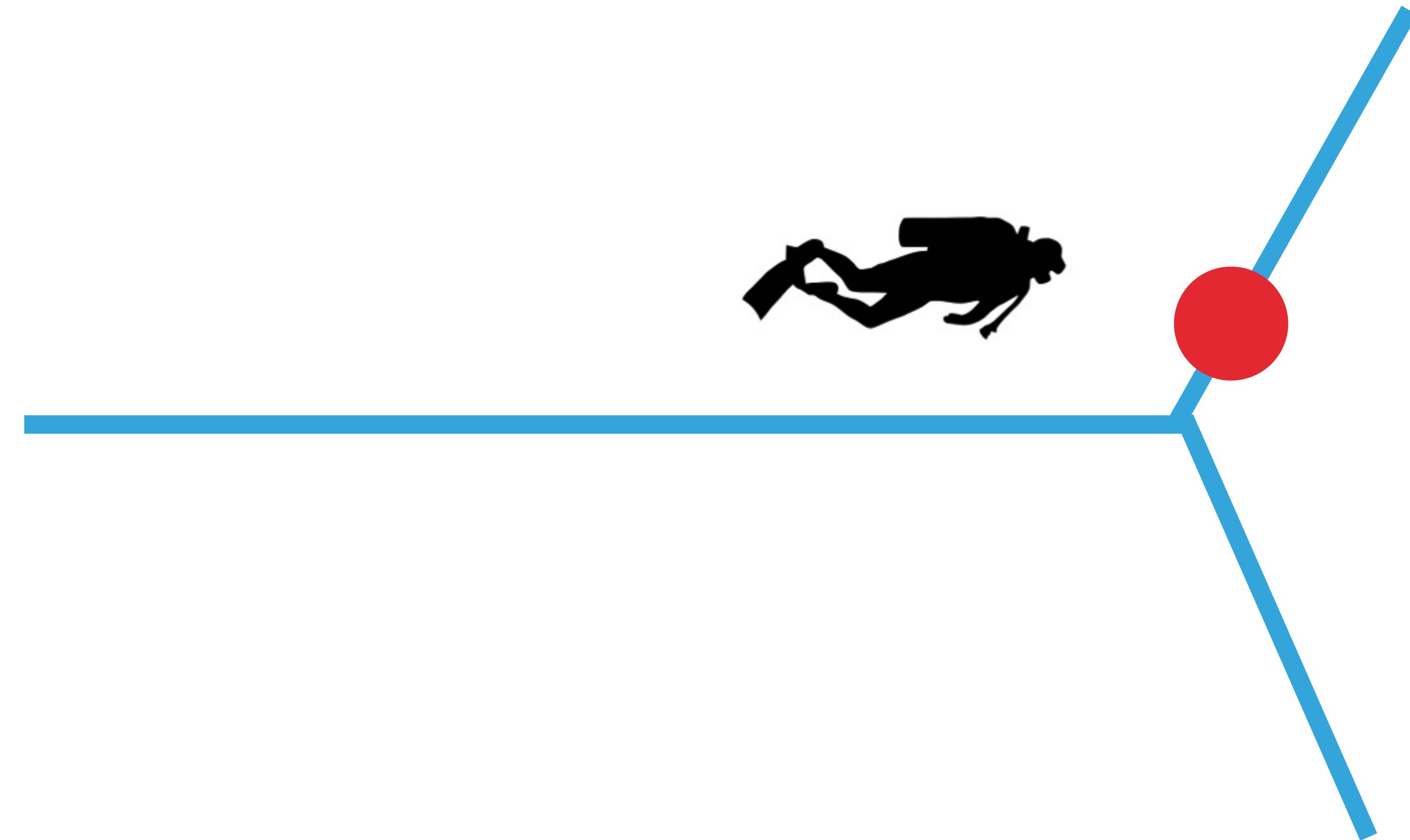
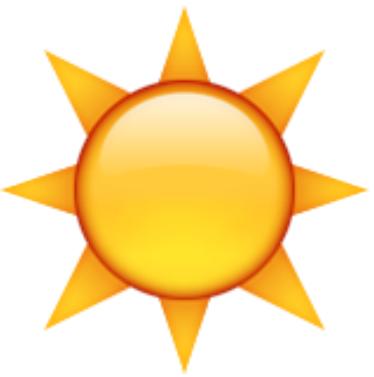
- ▶ Tests
- ▶ Code style (php-cs-fixer)
- ▶ Lint (php, yaml, twig, doctrine, composer)
- ▶ Static analysis (Psalm, PHPStan, Phan)

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?**

REMEMBER THE NAVIGATION COOKIE?



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

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?

```
/**  
 * @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

☆ | 84 | ⚡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?!

QUESTIONS TO ASK IN CODE REVIEW

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

CODE REVIEW TIPS

EVERYONE SHOULD CODE REVIEW



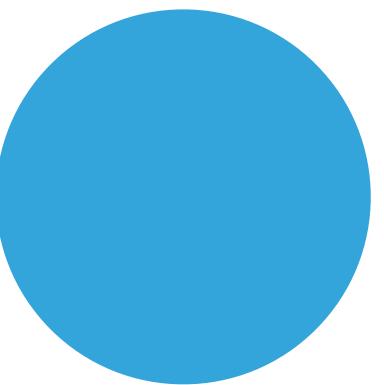
ASK PROGRAMMERS TO
REVIEW 10 LINES OF CODE
THEY'LL FIND 10 ISSUES...

Anyone who's done code review

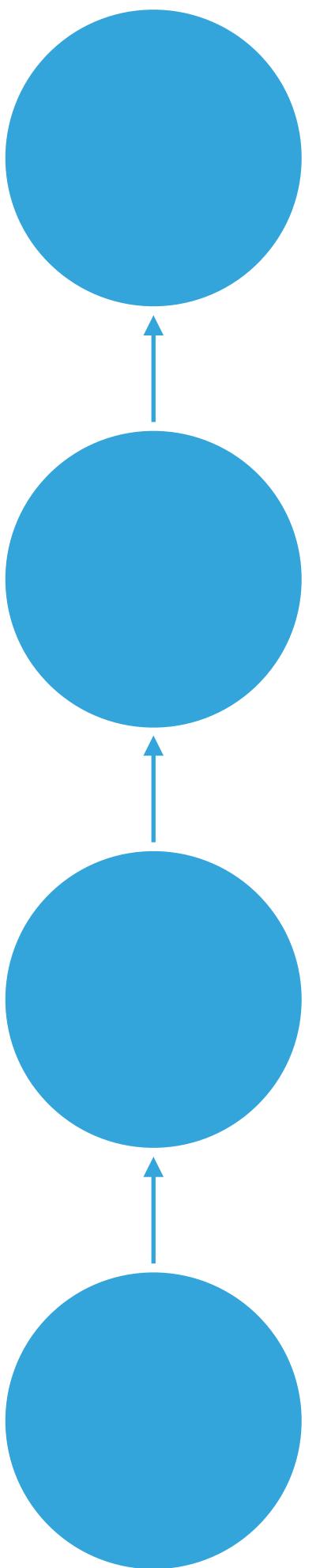
**ASK THEM TO DO 500 LINES
THEY'LL SAY IT'S GOOD TO GO**

Anyone who's done code review

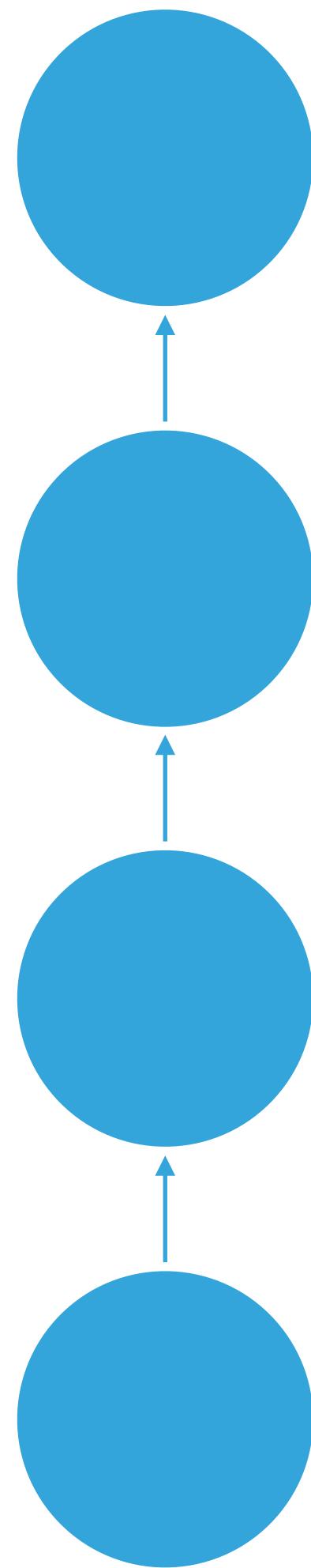
SMALL COMMITS



SMALL COMMITS

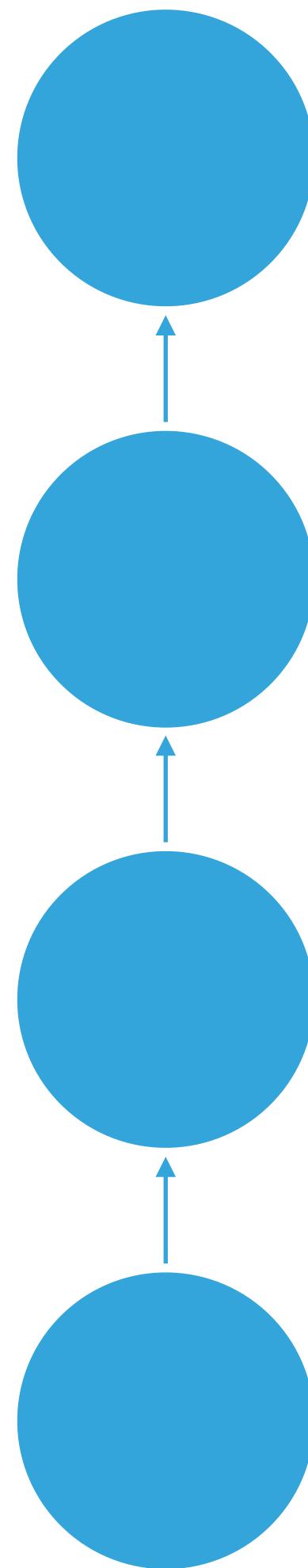


SMALL COMMITS



DEPRECATE: The price calculation service

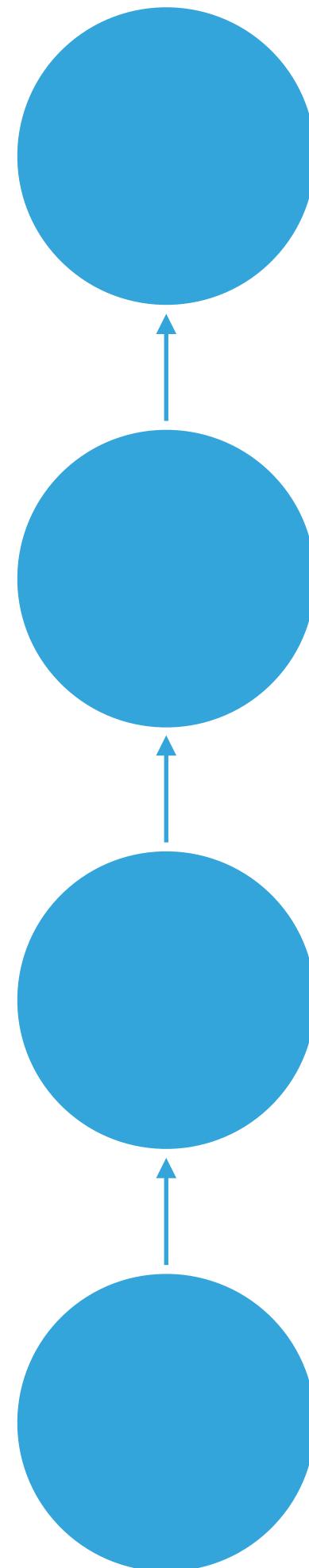
SMALL COMMITS



ADD: Facade to 3rd party price calculation service

DEPRECATE: The price calculation service

SMALL COMMITS

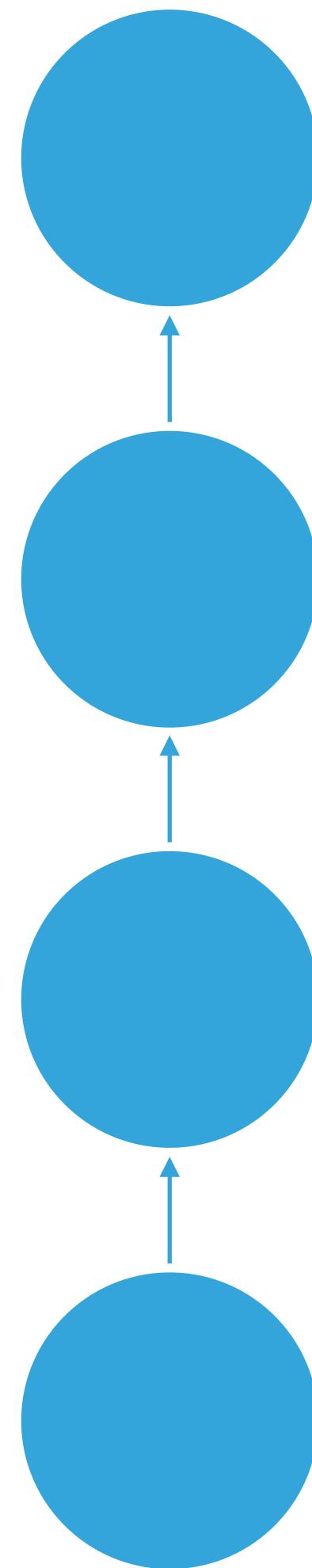


UPDATE: Use new price calculator code

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

CODE CAN ONLY BE DEPLOYED IF:

- ▶ CI passes
- ▶ Code review passes

SETUP GITHUB

The screenshot shows the GitHub repository settings page for 'DaveLiddament / Demo'. The left sidebar has a vertical list of options: Options, Collaborators, **Branches**, Webhooks, Integrations & services, and Deploy keys. The 'Branches' option is currently selected, indicated by an orange border. The main content area is titled 'Default branch'. It contains a description: '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.' Below this is a dropdown menu set to 'master' with an 'Update' button next to it. The next section is titled 'Protected branches'. It contains a description: 'Protect branches to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to protected branches? [Learn more](#).' Below this is a 'Choose a branch...' dropdown menu. A large gray box at the bottom states 'No protected branches yet.'

DaveLiddament / Demo

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights ▾

Unwatch 1 Star 0 Fork 0

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

The screenshot shows the GitHub repository settings page for 'DaveLiddament / Demo'. The top navigation bar includes links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Settings (which is highlighted with a red box), and Insights (0). The sidebar on the left lists Options, Collaborators, Branches (selected), Webhooks, Integrations & services, and Deploy keys. The main content area starts with a 'Default branch' section, which describes the default branch as the 'base' branch for pull requests and code commits. It shows 'master' selected and an 'Update' button. Below this is a 'Protected branches' section, which explains how to protect branches to prevent force pushing and deletion. It features a 'Choose a branch...' dropdown and a message stating 'No protected branches yet.'

DaveLiddament / Demo

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights 0

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

The screenshot shows the GitHub repository settings page for 'DaveLiddament / Demo'. The 'Settings' tab is selected. On the left, a sidebar lists 'Options', 'Branches' (which is highlighted with a red box), and 'Webhooks'. The main content area is titled 'Default branch' and describes it as the 'base' branch for pull requests and commits. It shows a dropdown set to 'master' and an 'Update' button. Below this is the 'Protected branches' section, which is currently empty, displaying a message 'No protected branches yet.' A 'Choose a branch...' dropdown is present.

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

The screenshot shows the GitHub repository settings for 'DaveLiddament / Demo'. The 'Branches' tab is selected in the sidebar. The main content area is titled 'Default branch' and describes it as the 'base' branch. A dropdown menu shows 'master' with a 'Update' button. Below this, the 'Protected branches' section is shown, featuring a red box around the 'Choose a branch...' dropdown menu. A message indicates no protected branches have been created yet.

DaveLiddament / Demo

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

Protected branches enable 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](#)

[!\[\]\(edb3028f087d2c2084b9188254674d23_img.jpg\) Unwatch](#) 1 [!\[\]\(8a3cffc06ebd740480151105decfe01a_img.jpg\) Star](#) 0 [!\[\]\(0e5e53e7d1f38a10c57eaab9f379e3eb_img.jpg\) Fork](#) 0

[!\[\]\(45357e9c0f5331c65ab6071394840be1_img.jpg\) Code](#) [!\[\]\(bce3c2b3d0421f08becfc04cb323e84e_img.jpg\) Issues 0](#) [!\[\]\(e9d867eec0f20763f1b8a8cbbbfa54b0_img.jpg\) Pull requests 0](#) [!\[\]\(f9a52c32fae7f8178aa70f827e4cb90d_img.jpg\) Projects 0](#) [!\[\]\(34a7880ce85817f3cce2a014d5d72715_img.jpg\) Wiki](#) [!\[\]\(aaa1eff60dfb3957b513f313fdfce834_img.jpg\) Settings](#) [!\[\]\(ba984a5b599ba3d2836ed82aed6addc0_img.jpg\) 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](#)

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   

[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 review approval.

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](#)

[!\[\]\(b8a623f0063fa7bbff8017994fa2a19c_img.jpg\) Unwatch](#) 1 [!\[\]\(123f37fc4307f228f68f2836fe85df6e_img.jpg\) Star](#) 0 [!\[\]\(1859868586e82e2dfbaaf2e31ffec252_img.jpg\) Fork](#) 0

[!\[\]\(815f840f5645cc23c174a83658db6adf_img.jpg\) Code](#) [!\[\]\(d3cdeeafe891f50067c1c69860b7347e_img.jpg\) Issues 0](#) [!\[\]\(01f304effa76d36b498a8cca5b7df8bc_img.jpg\) Pull requests 0](#) [!\[\]\(b36f43a69f0b3a4d91179669efbd9b7c_img.jpg\) Projects 0](#) [!\[\]\(41a0b70a9fed5e965cc468b98750eb3c_img.jpg\) Wiki](#) [!\[\]\(ca69990963eb7eefe6a7af3f7e0e03f2_img.jpg\) Settings](#) [!\[\]\(9b4d7d66f35462ab383ea2509cebedd6_img.jpg\) 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

The screenshot shows the GitHub repository settings page for the 'Demo' repository owned by 'DaveLiddament'. The 'Branch protection for master' section is displayed, featuring several configuration options:

- Protect this branch**: A checked checkbox with a descriptive subtitle about preventing force-pushes and deletion.
- Require pull request reviews before merging**: A checked checkbox with a detailed subtitle explaining the requirement for approved reviews and no changes requested before merging.
- Dismiss stale pull request approvals when new commits are pushed**: A checked checkbox with a subtitle about dismissing approvals for new commits.
- Require review from Code Owners**: An unchecked checkbox with a subtitle about requiring reviews from designated code owners.
- Require status checks to pass before merging**: A checked checkbox with a subtitle about requiring status checks to pass before merging, including a note about pushing to another branch first.
- Require branches to be up to date before merging**: A checked checkbox with a subtitle about ensuring the branch is tested with the latest code on master.

At the bottom, a message states: "Sorry, we couldn't find any status checks in the last week for this repository." followed by a link: "Learn more about status checks on GitHub."

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

DaveLiddament / Demo

Unwatch 1 **Star** 0 **Fork** 0

Code **Issues 0** **Pull requests 0** **Projects 0** **Wiki** **Settings** **Insights**

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.

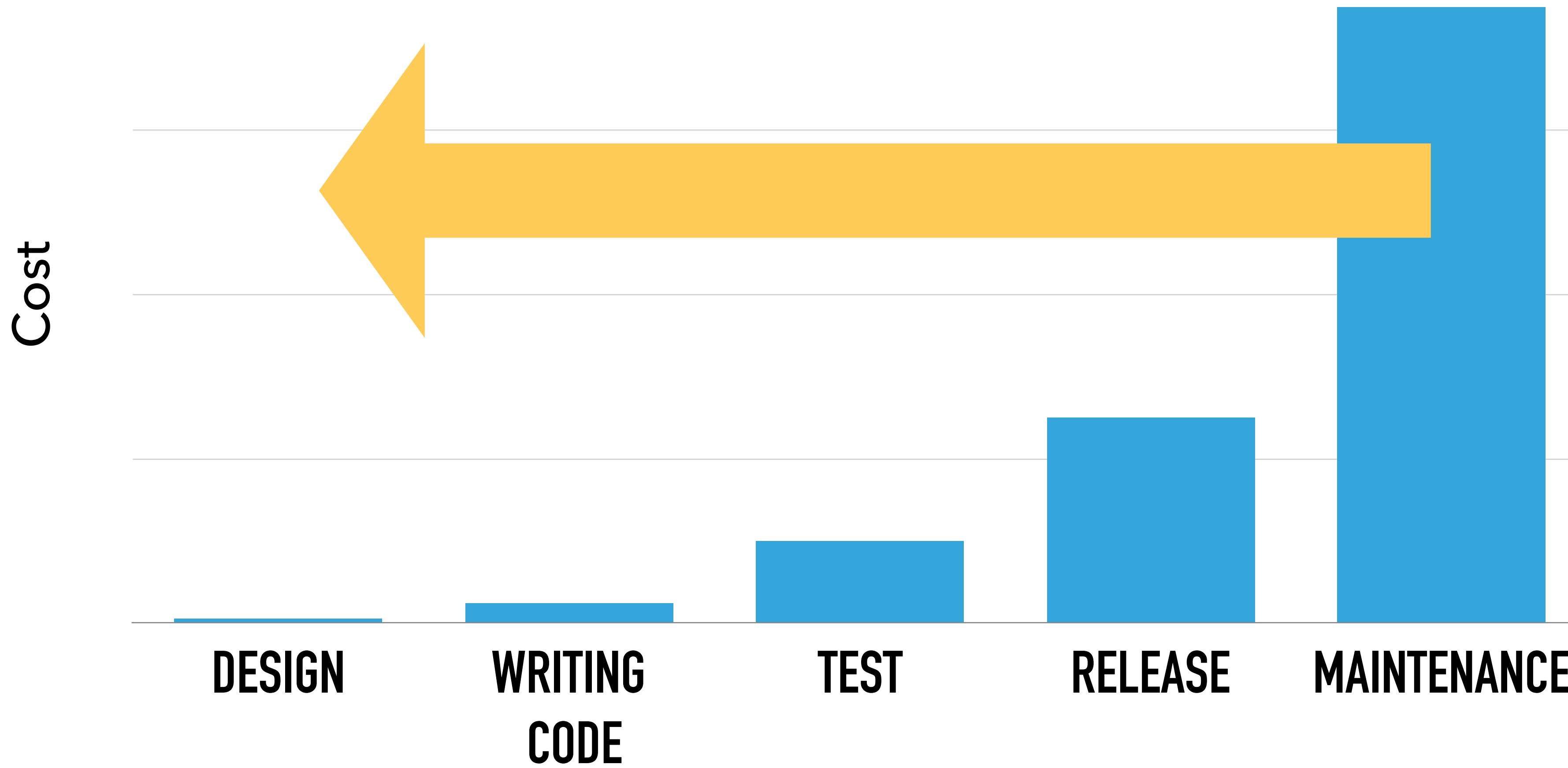
INTEGRATING CODE REVIEW INTO PROJECT WORKFLOW

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

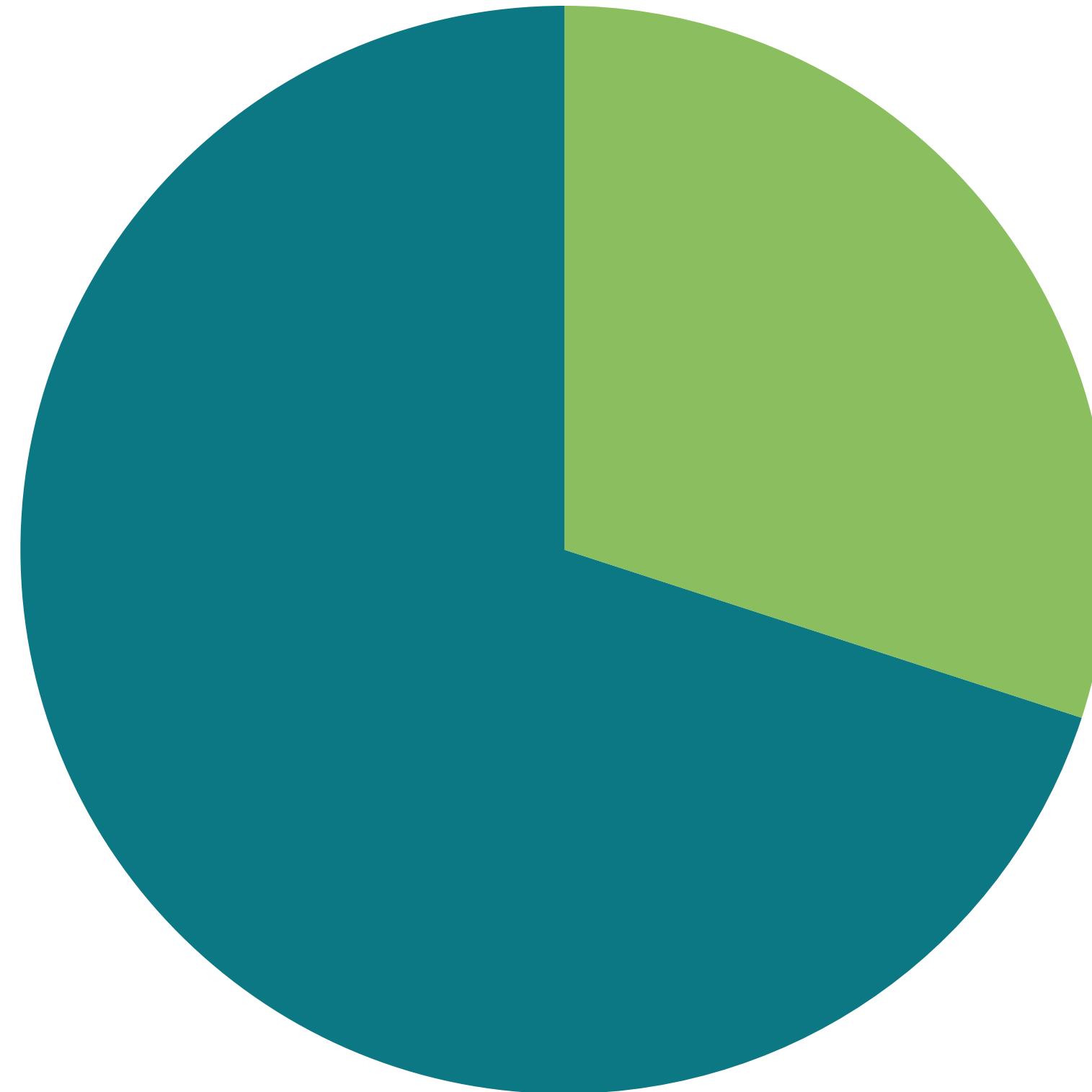
WHAT WE'VE COVERED

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

FIND BUGS SOONER



WHAT ARE DEFECTS?

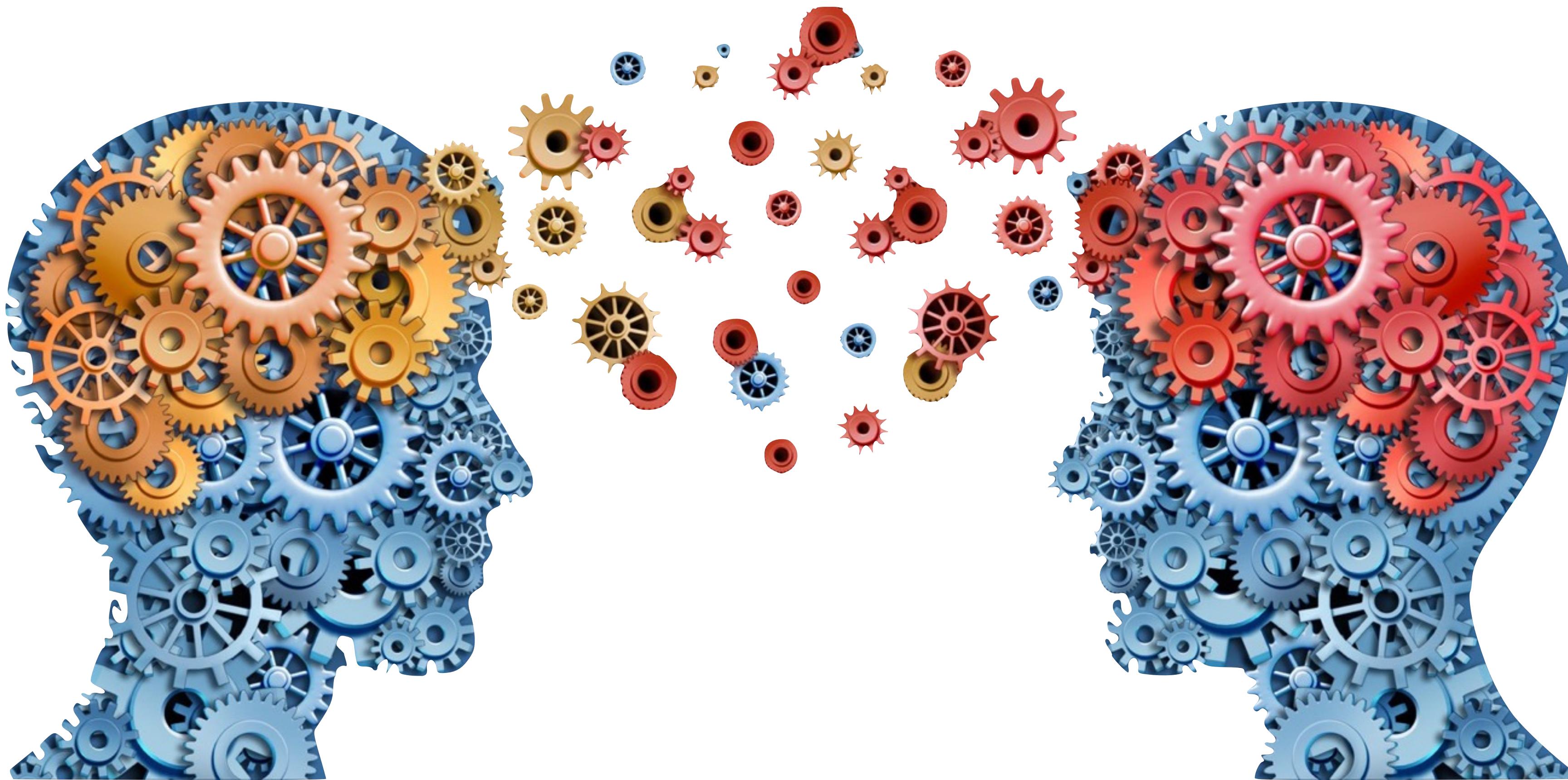


Evolvability

[1, 2]

WRAP UP

SPREAD THE KNOWLEDGE



WRAP UP

MENTORING



WRAP UP

YOU'RE BEING WATCHED!



YOU'RE BEING WATCHED!





SUMMARY: REVIEW CHECKLIST

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

#coding-standards

☆ | 84 | ⚡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



**EFFECTIVE CODE REVIEW
REDUCES OVERALL COST OF
SOFTWARE DEVELOPMENT**



FEEDBACK



<https://joind.in/talk/ccc25>

@daveliddament

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. Bunker, 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