

# L'instabilité de nos tests nous empêche de délivrer

**Sofía LESCANO CARROLL**



@SofLesc





**Levez la main si vos tests vous mettent en colère ou  
leur écriture est pénible**



**J'ai la solution !**



**Pas de tests, pas de problème !**  
(Évidemment c'est une blague)

# Sofía Lescano Carroll

Développeuse Senior



Madrid

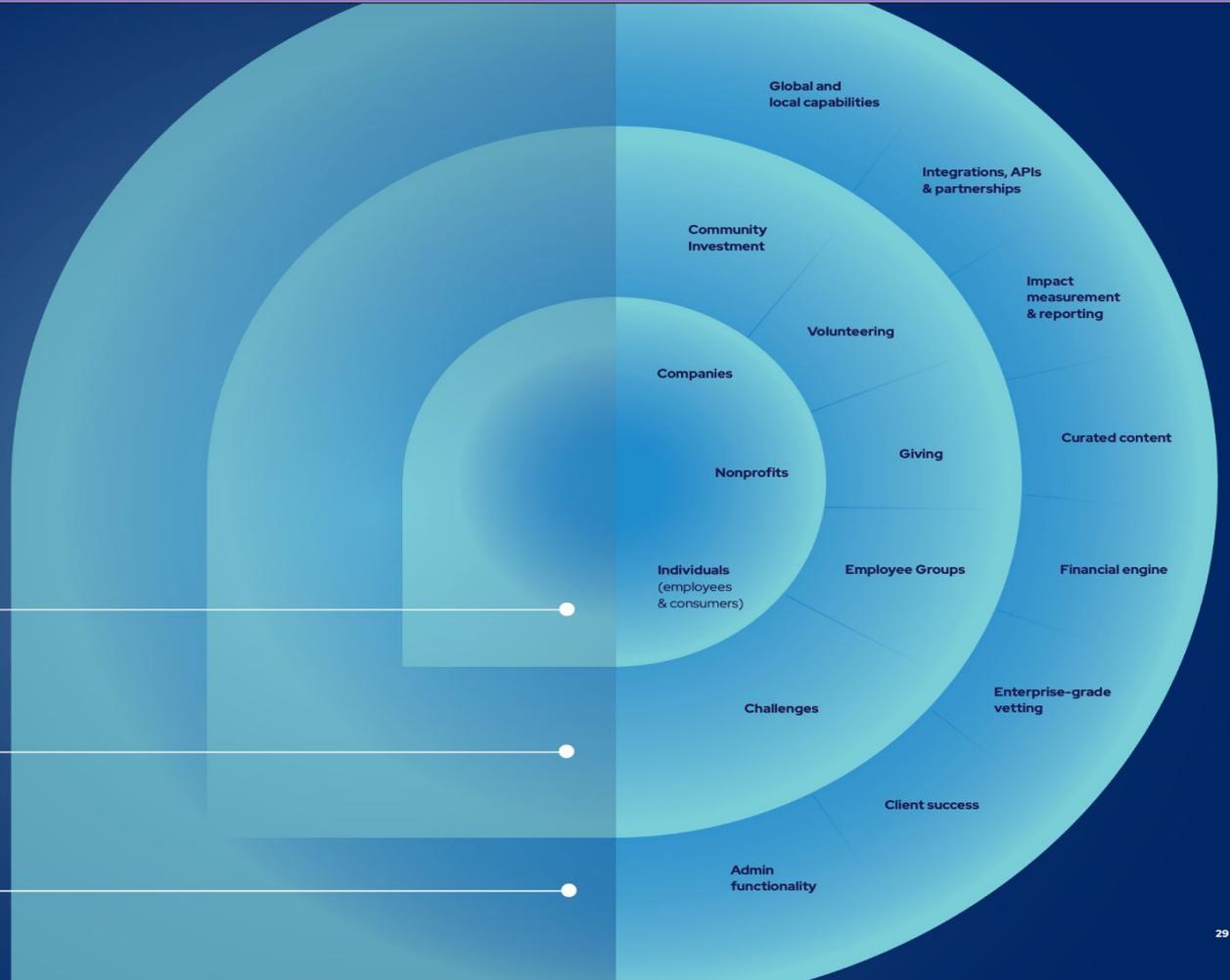


# Meet the Benevity platform

Who we serve

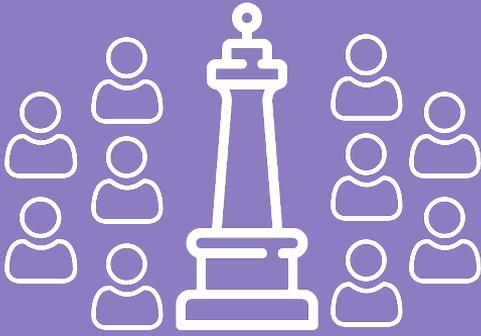
Solutions

Capabilities



Pourquoi écrire des tests ?

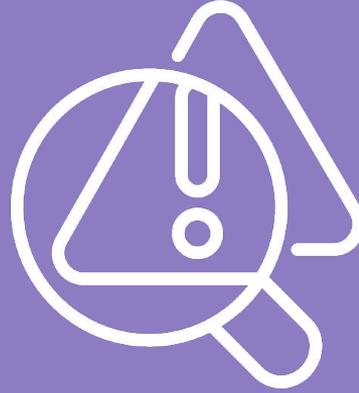
# Pourquoi écrire des tests ?



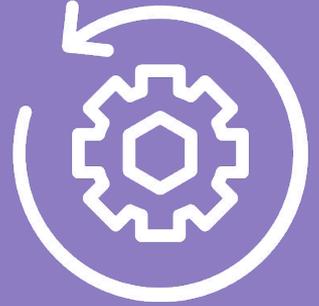
Difficulté à connaître  
toute l'application



Vérifier le bon  
fonctionnement

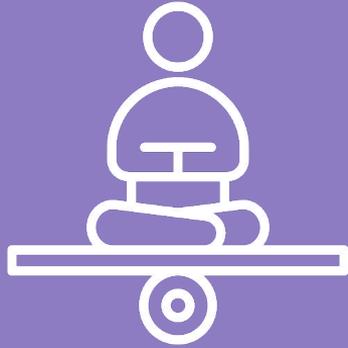


Éviter les  
regressions



Automatique

# Pourquoi écrire des tests ?

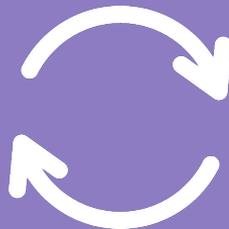


Tranquilité lors du développement et  
confiance sur le code

# Si les tests sont bien écrits



Test qui  
ne passe pas



Mettre à  
jour le test



On vient de détecter  
un problème



Et si je n'ai pas  
confiance en mes  
tests ?

C'est quoi un test peu fiable ou 'flaky' ?



# Flaky test

Un test est dit *flaky* ou peut fiable lorsque il donne à la fois des résultats positifs et négatifs malgré aucune modification du code ou du test.

# Flaky test

Le résultat du *flaky* test étant non-déterministe, reproduire et corriger le cas d'erreur peut s'avérer très complexe.

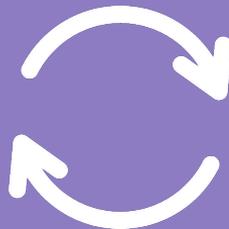
# Flaky test

Un test flaky peut passer  
 $\frac{1}{2}$  ,  $\frac{3}{4}$  ,  $\frac{199}{200}$  ...

# Si les tests sont bien écrits



Test qui  
ne passe pas

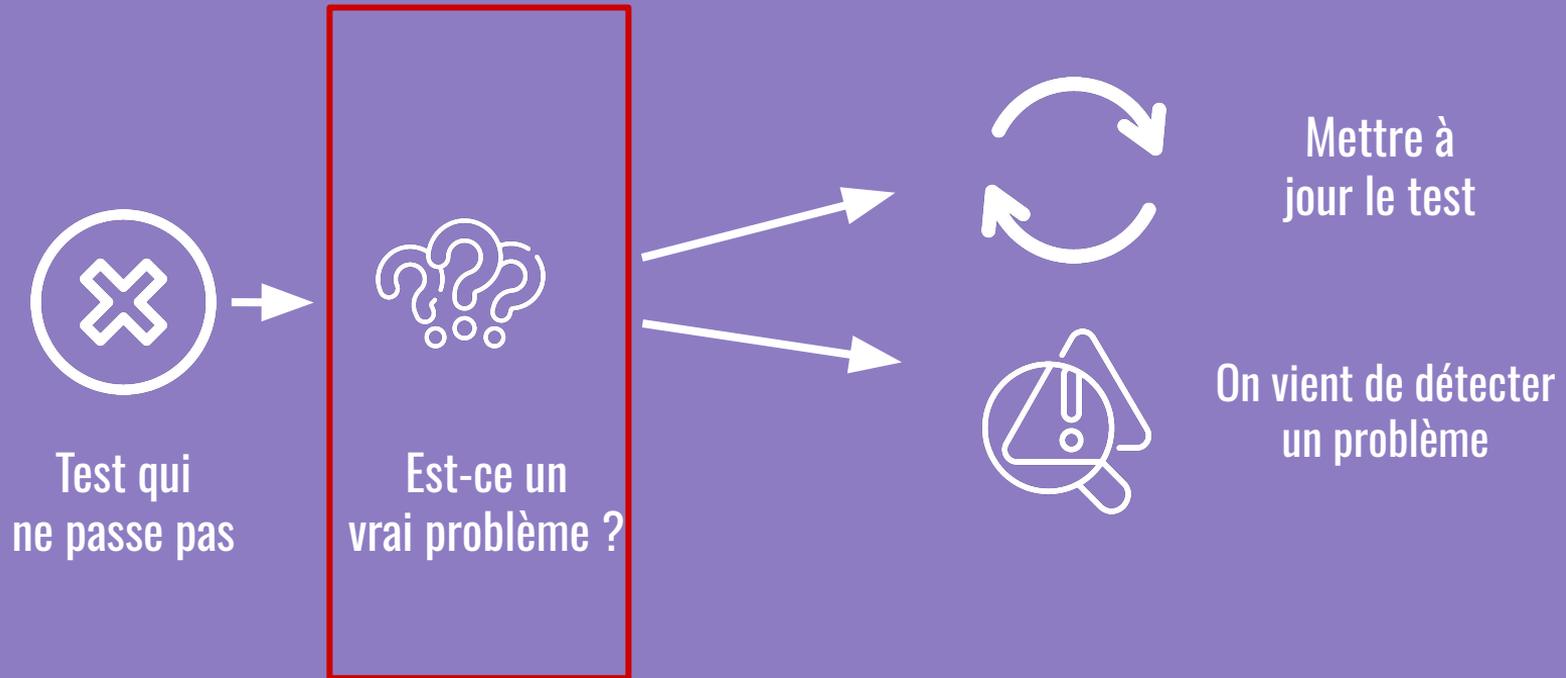


Mettre à  
jour le test



On vient de détecter  
un problème

# Et si je n'ai pas confiance en mes tests ?

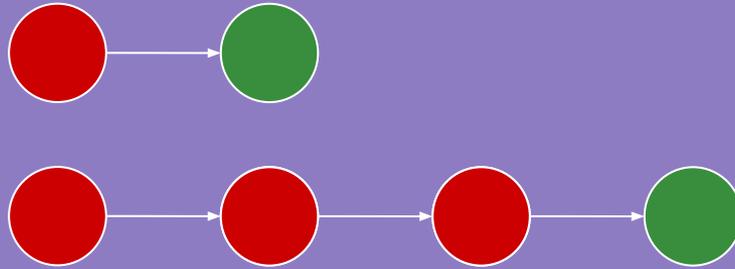


# Y a-t-il un vrai problème dans mon application ?



Plusieurs exécutions ne  
permettent pas toujours de  
conclure

# Y a-t-il un vrai problème dans mon application ?



Plusieurs exécutions ne permettent pas toujours de conclure

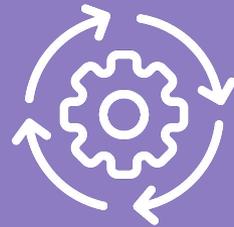
# Y a-t-il un vrai problème dans mon application ?



Plusieurs exécutions ne permettent pas toujours de conclure

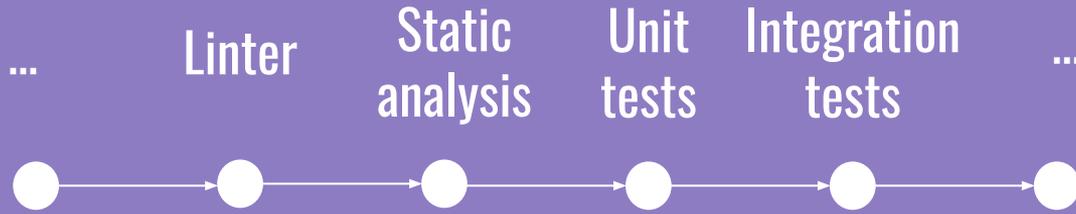


Comprendre l'objectif d'un test qu'on a pas écrit



Temps et capacité à exécuter le tests

# Cycle de développement



On ne merge et déploie que quand les tests passent

# L'impact des tests instables



Incertitude sur l'état  
de l'application



Dégradation  
capacité à délivrer  
de la valeur



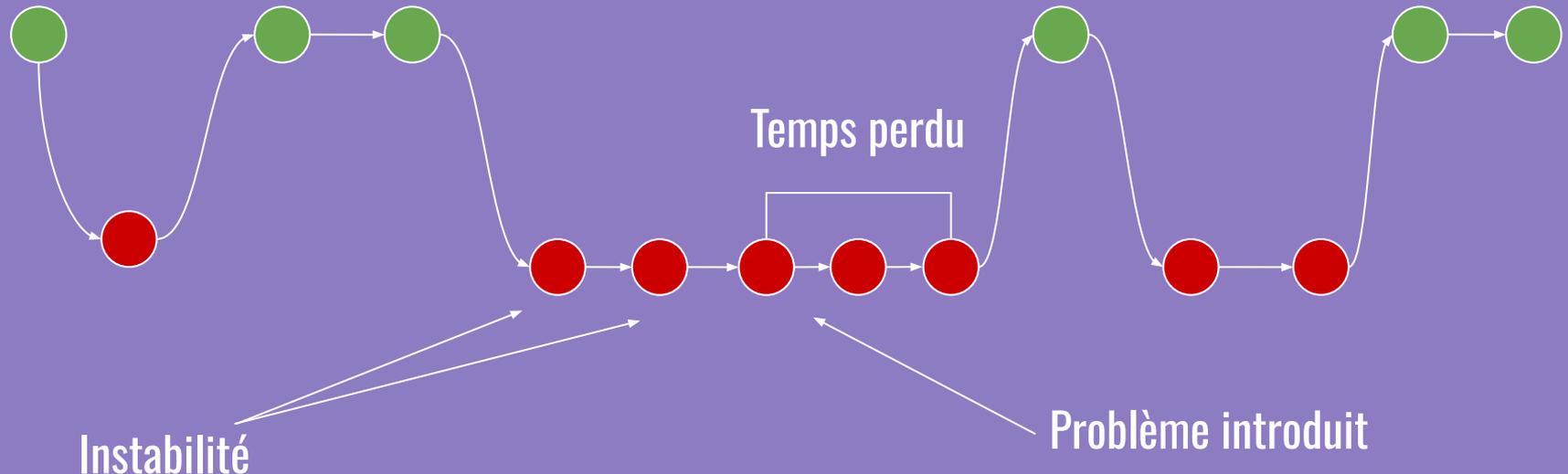
Frustration des  
développeurs

# L'impact des tests instables



**Il y a tellement de bruit que on ne prend plus  
au sérieux les vrai problèmes**

# L'impact des tests instables

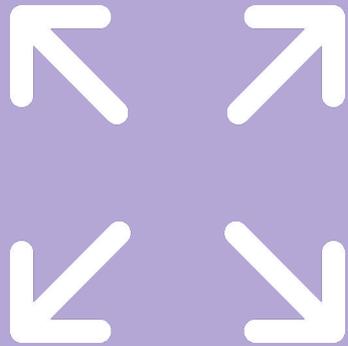


Historique des builds

# L'impact des tests instables



- Le test ne passe pas
- C'est normal ... il doit juste être 'flaky'



Un problème connu  
qui peut se répandre  
très vite



**Levez la si de temps en temps quand vous rencontrez un problème vous faites semblant de ne pas l'avoir vu**

# tech-backend-chapter – Jan 6th

damn you flaky test!

# tech – Feb 23rd

2:21 PM  
Oh, flaky tests strike again...

For those who's watching

- We disabled the Manta test because the behaviour is actually flaky

... now more

👍 1 🗨️

11:43 AM

These flaky tests are getting out of hands :

< 1 minute ago

the flaky test monster is a hydra, you cut of one head and two grow in its place!

Sep 7th, 2023 at 2:33 PM

yeah, it's very flaky, now it went the other way around

# tech – Feb 23rd

I restarted the workflow on CircleCI, apparently it was just \*another\* flaky test

😄 1 reaction

Sorry for the delay, I hate flaky tests, it's the 4th time in a row I launch tests execution and they are failing due to flaky tests !!

what a waste of time 🙄 (edited)

This seems to be a flaky test

when flaky test will have a good mood for my PR ><

- Build failed.
- The test has previously been reported as flaky. - <https://pm.>

We will go in production in 15 minutes, we just fix a flaky test

# Un problème connu: persévérance et patience

14 minutes ago

after 6 try, test are ok now





**Levez la main si vous avez déjà copié du code qui existait dans votre projet pour en créer du nouveau**

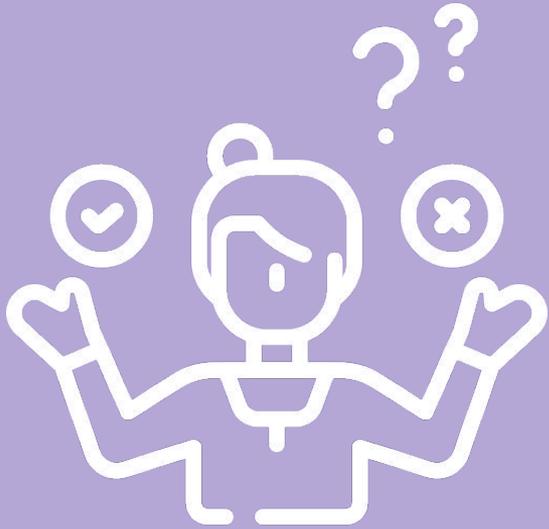
# Un problème qui s'aggrave



Créé de nouveaux tests  
basés sur les anciens

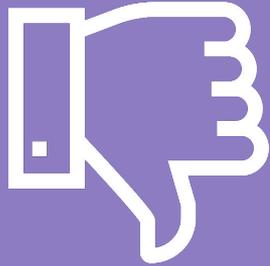


Nouvelles fonctionnalités >  
stabilité de la stack technique

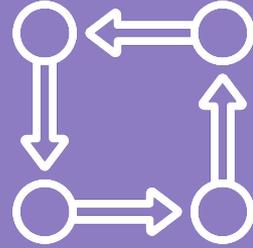


Comment écrire  
un bon test *flaky*

# Principales causes de *flaky* test



Des tests mal écrits (dates,  
règles métier ... )



dépendances et  
problèmes d'ordre

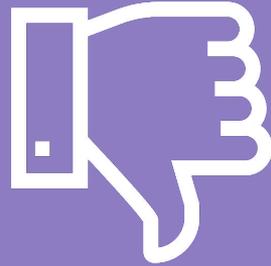


Problèmes de timing



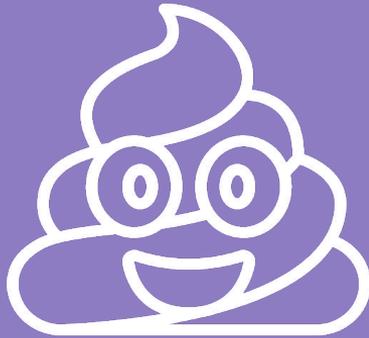
Dépendances externes

# Principales causes de *flaky* test

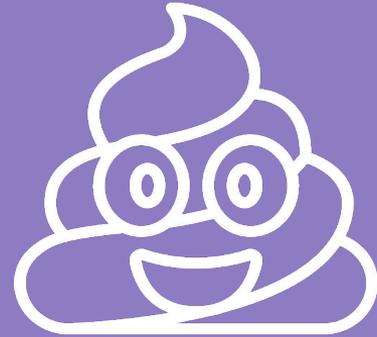


Des tests mal écrits (dates,  
règles métier ... )

# Principales causes de *flaky* test



Des tests pas  
terrible



Des résultats  
pas terrible

# Principales causes de *flaky* test: des test mal écrits

```
public function testRenewedBudgetEnd()
{
    // Arrange.
    $budget = self::budgetWithNoLimits();

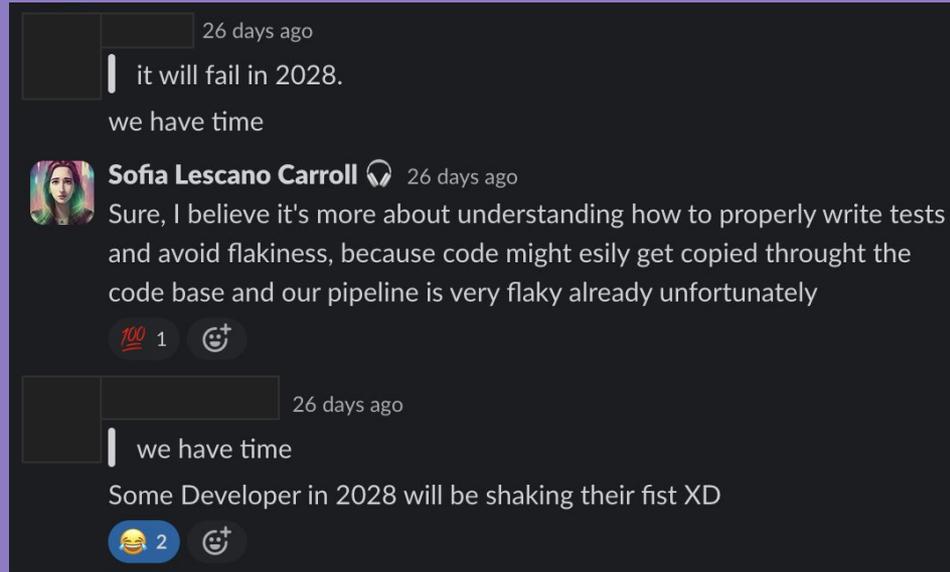
    // Act.
    $newStart = $budget->getRenewedBudgetEnd();
    $dateDiff = $budget->getDuration()->getEndTimeStamp()->diff($newStart);
    $daysInterval = $dateDiff->days;

    // Assert.
    $this->assertEquals(366, $daysInterval, 'Renewed budget end date should 1 year more than original budget end date');
}
```



Le test qui arrête de fonctionner le 1er mars

# Principales causes de *flaky* test: des test mal écrits



Un problème pour le développeur du futur

# Principales causes de *flaky* test: des test mal écrits

```
public function testExportInvoices(){
```

```
...
```

```
$firstInvoice = Invoice::factory()->for($brand)->create(['created_at' => Carbon::now()->subMonths(2)]);
```

```
$secondInvoice = Invoice::factory()->for($brand)->create(['created_at' => Carbon::now()->subMonth());
```

```
$thirdInvoice = Invoice::factory()->for($brand)->create(['created_at' => Carbon::now()]);
```

```
$subject->exportInvoices($brand);
```

```
...
```

```
}
```

**31 mars -> 31 février ?!**

**->SubMonthNoOverflow**

"Non mais Sofia, tu ne sais pas qu'on a pas le droit de tester le 31 de chaque mois ?"

# Principales causes de *flaky* test: des test mal écrits



Piste de résolution: utilisation d'une clock interface

# Principales causes de *flaky* test: des test mal écrits

```
// Des règles business à prendre en compte : pas de vente entre France et Angleterre
public function testAddItemToCart(){
    $client = Client::factory()->create(); // Le pays est aléatoire
    $brand = Brand::factory()->create(); // Le pays est aléatoire
    $product = Product::factory()->for($brand)->create();
    $subject->addItemToCart($client, $product);
    self::assertCount(1, $client->cart);
    ...
}
```

Des données aléatoires oui, mais pas toujours

# Principales causes de *flaky* test: des test mal écrits

```
public function myTest() {  
    ...  
    $firstVariable = uniqid(); // 65691234abcde  
    $secondVariable = uniqid(); // 6569123456789  
    $subject->functionToTest($firstVariable, $secondVariable);  
    ...  
}                               uniqid('first_variable') -> first_variable_6569123456789
```

Données aléatoires avec `uniqid()` : string pouvant être convertie en int

# Principales causes de *flaky* test: des test mal écrits

```
// Les ids de mysql ne sont pas remis à zéro entre les tests

public function testCreateBrand(){
    $brand = $subject->createBrand('mybrand@mycompany.com', 'FR')
    $logger->shouldReceive('info')->with('New brand created with id 1')->once();
    ...
}
```



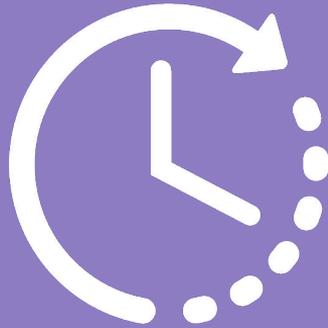
Id des entités sqlite -> mysql

1 test à corriger ok, 234 tests à corriger ... pas fun

# Dépendance avec une Feature Flag

- 1/ Introduit une fonctionnalité derrière une Feature Flag
- 2/ Le test n'est pas explicite sur la valeur de la FF
- 3/ Si on change la FF sur l'environnement le test ne passe plus

# Principales causes de *flaky* test



Problèmes de timing

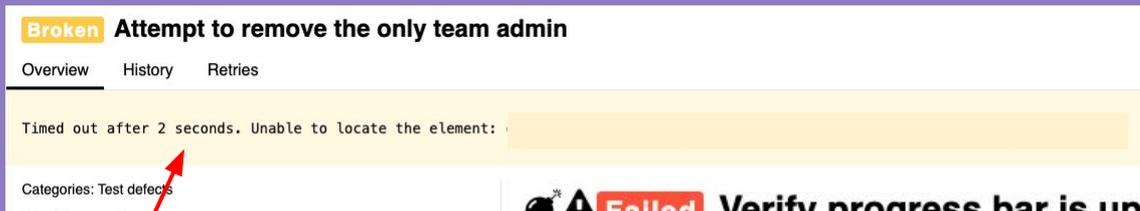
# Principales causes de *flaky* test: temps d'attente dépassé

**Broken** Attempt to remove the only team admin

Overview History Retries

Timed out after 2 seconds. Unable to locate the element: ...

Categories: Test defects



**Failed** Verify progress bar is updated with giving account donation

Overview History Retries

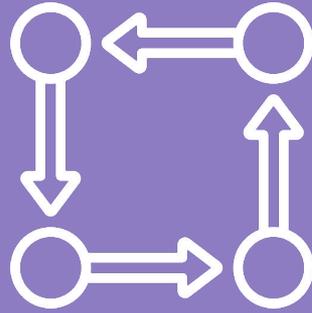
The following 4 assertions failed:

- 1) [the friendraiser total] expected:<"\$[23]6.11"> but was:<"\$[18]6.11">
- 2) [the friendraiser donation percentage] expected:<"[7]4"> but was:<"[9]4">
- 3) [the friendraiser matching percentage] expected:<"[2]5"> but was:<"[ ]5">
- 4) [the friendraiser funded message] expected:<"1[5]% Funded"> but was:<"1[2]% Funded">



Temps d'attente dépassé ou lecture anticipée

# Principales causes de *flaky* test



Dépendances et  
problèmes d'ordre

# Principales causes de *flaky* test : tests dépendants

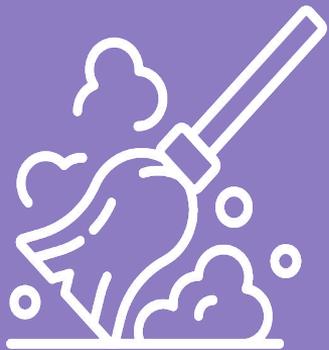
```
// Une base de données partagée par vos tests

public function testCreateBrand(){
    $brand = $subject->createBrand(self::BRAND_EMAIL, self::BRAND_COUNTRY);
    ...
}

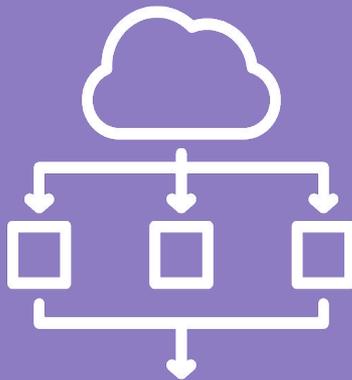
public function testEditBrand(){
    $brandToEdit = $repository->findByEmail(self::BRAND_EMAIL);
    $editedBrand = $subject->editBrand($brandToEdit, ['country' => 'ES']);
    ...
}

public function testListBrands(){
    $brands = $subject->listBrands();
    self::assertEquals('ES', $brands[0]->country);
    ...
}
```

# Principales causes de *flaky* test : tests dépendants



Manque de nettoyage  
(données, configurations)



Concurrence

# Principales causes de *flaky* test

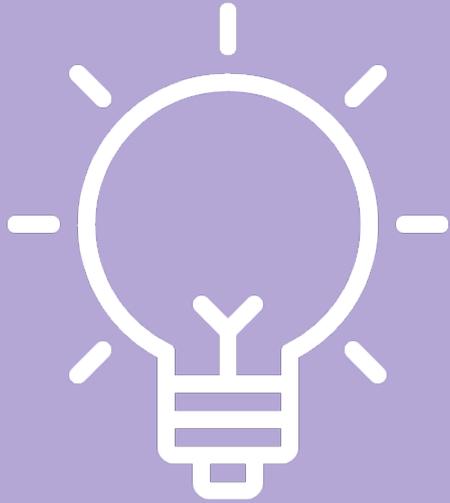


Dépendances externes

# Principales causes de *flaky* test : tests dépendants



Appel à une sandbox de paiement qui peut être instable

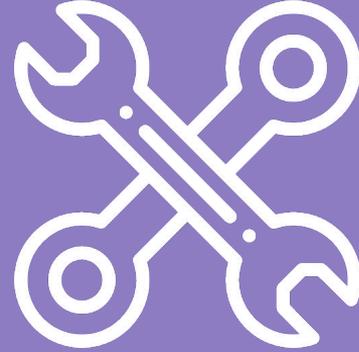


Comment  
s'attaquer à ce  
problème

# Combattre cela



Un problème  
humain



Un problème  
technique

# Combattre cela



Avec une grande base de code et beaucoup de développeurs, cela va arriver tôt ou tard

# Combattre cela



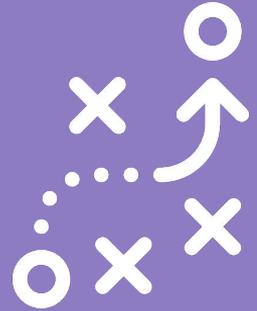
Comprendre les impacts



Améliorer les pratiques de test



Mesurer



Établir une stratégie



Avoir les bons outils

# Combattre cela



Aux développeurs

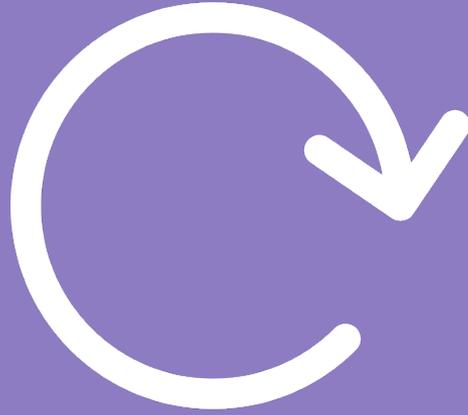


Faire comprendre les impacts



Au produit & management

A l'échelle du développeur



C'est plus simple de rejouer le test et espérer qu'il passe

# A l'échelle de l'entreprise



Un test *flaky* peut potentiellement échouer sur chaque workflow en cours (PR, déploiement ...) jusqu'à ce qu'il soit fixé ou ignoré

## A l'échelle de l'entreprise

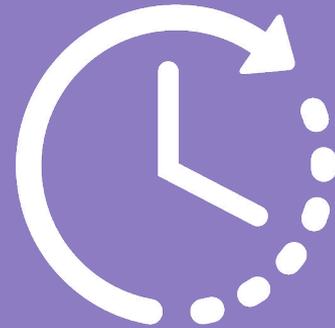


Perte de temps et coût humain et logiciel

# A l'échelle de l'entreprise



**Un effort collectif**



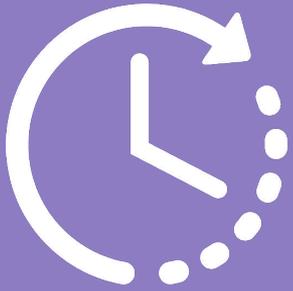
**Allouer du temps**

# Quantifier le problème

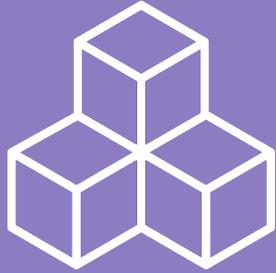


"If you can't measure it, you can't manage it" - Peter Drucker

# Quantifier le problème: le rendre visible



Temps passé



Nombre de  
flaky tests



Releases  
affectées



Mesurer  
évolution

Combattre cela



Améliorer les pratiques de test

# Améliorer les pratiques de test



Se former



Partager les erreurs  
rencontrées



Relire les tests et  
pas juste le code

**Combattre cela**



**Avoir les bons outils & apprendre à  
les utiliser**

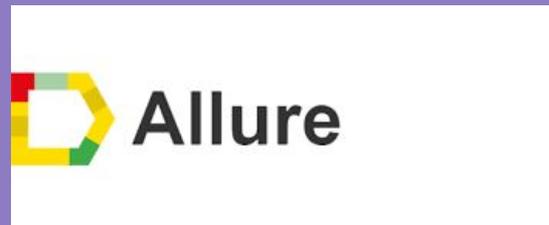
# Avoir les bons outils et apprendre à les utiliser



Intégration et  
déploiement continu



Monitoring  
d'infrastructure et  
d'application



Outil reporting tests  
d'automatisation

# Détecter les tests *flaky* avec Allure et CircleCI

## Détection de résultats différents avec le même code

**Broken** Verifies that Giving Opportun  
See bug DEV-32802

Overview History Retries

Success rate 99.28% (558 of 562)

broken	3/5/2024 at 10:46:43
passed	3/4/2024 at 18:43:27
passed	3/4/2024 at 17:13:07
passed	3/4/2024 at 15:12:14
broken	3/4/2024 at 13:13:47
passed	3/4/2024 at 11:14:42
passed	3/1/2024 at 21:02:22
passed	3/1/2024 at 18:55:44
skipped	3/1/2024 at 17:11:18
passed	3/1/2024 at 15:10:51
passed	3/1/2024 at 14:46:52
passed	3/1/2024 at 11:55:02
passed	3/1/2024 at 9:08:58
passed	2/29/2024 at 16:28:31
passed	2/28/2024 at 18:35:43
passed	2/28/2024 at 18:02:26
passed	2/28/2024 at 15:58:53
passed	2/28/2024 at 15:36:32
passed	2/28/2024 at 14:33:39
passed	2/28/2024 at 10:15:39

! 1 test failed out of 16481

▼ testBrandCanViewOrder with data set "status-brand-confirmed"  **FLAKY**

# Détecter les tests *flaky* avec CircleCI

ⓘ Tests metrics are calculated based on 100 most recent workflow runs

Filtering by date range does not apply to tests.

Avg. tests per run

24,611

[See more ↓](#)

Flaky tests detected

346

[See more ↓](#)

Test failures

9

[See more ↓](#)

Slowest test P95 Duration

53s

[See more ↓](#)

## Most Failed Tests

Among 100 most recent workflow runs

Test	Job	Duration (p95)	Runs	Success Rate
testAlertLowStripeBalanceWithInsufficientBalance with data set "GBP" <b>FLAKY</b>	test-php-sta...	6s	98	93%
testAlertLowStripeBalanceWithSufficientBalance with data set "GBP" <b>FLAKY</b>	test-php-sta...	6s	98	94%
testCreate <b>FLAKY</b>	test-php-sta...	1s	97	99%
testDirtyDiscountRate <b>FLAKY</b>	test-php-sta...	15s	97	99%

# Monitoring de l'état global de l'application

# Détecter les tests *flaky* avec Datadog

Commits

Search by commit SHA

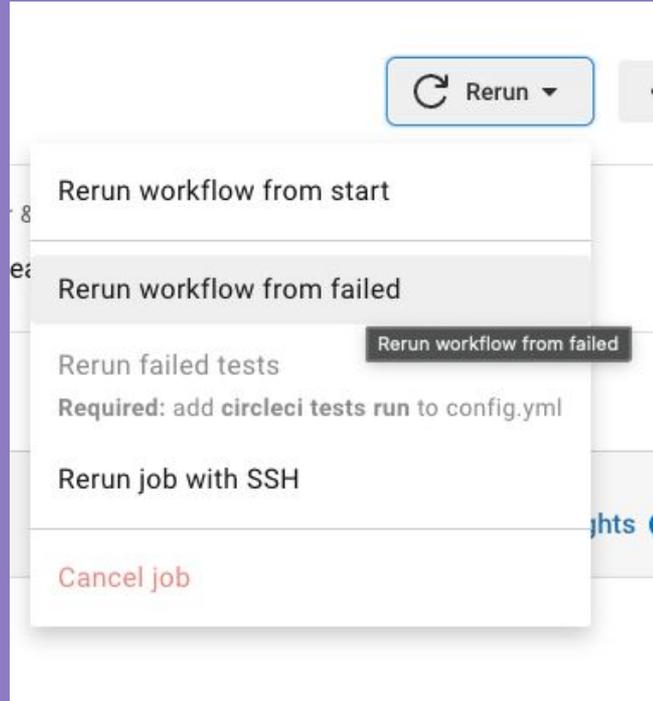
Commits per page: 5

COMMIT	FAILED	PASSED	SKIPPED	NEW FLAKY	REGRESSIONS	FAILURE %	TOTAL TESTS	TOTAL TIME	LAST UPDATED
Revert: Add new payment method with b...  7cdbdac <span>LATEST</span>	3	159	0	1	0	1.85%	162	24 min 12 s	1 hour ago
Improve performance on integration tests  4041638	3	159	0	0	0	1.85%	162	23 min 7 s	1 hour ago
New feature: order products by available stock  56f304f	3	159	0	0	0	1.85%	162	24 min 26 s	1 hour ago
Update dependency used to minify css assets  ceaafb4	3	159	0	1	0	1.85%	162	23 min 24 s	6 hours ago
Improve performance on integration tests  94ec285	3	159	0	1	0	1.85%	162	23 min 40 s	6 hours ago

Ignore flaky tests

## Ignorer un test détecté par erreur

# Relancer les *falky* test avec CircleCI



Exécuter un sous-ensemble de tests ou l'étape en erreur pour gagner du temps

Éviter certains comportements à travers le code

Utiliser un analyseur statique  
(PHPStan)

Implémenter nos propres règles

# Éviter certains comportements à travers le code

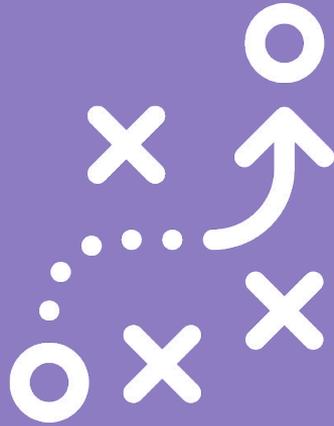
```
public function myTest {  
  ...  
  
  $date = Carbon::now();  
  $previousMonth = $date->subMonth(); // This will trigger the rule.  
  $safePreviousMonth = $date->subMonthNoOverflow(); // This will not trigger the rule.  
  
  ...  
}
```

----- line xxx -----

Avoid using `subMonth()`. Use `subMonthNoOverflow()` instead to prevent unexpected date changes.

💡 Use `subMonthNoOverflow()` instead.

# Combattre cela

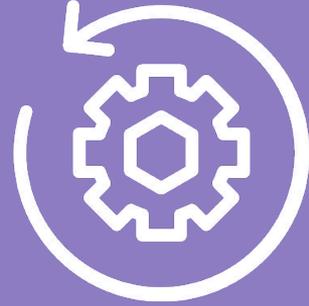


## Stratégies de contournement

# Combattre cela: comment est fait le déploiement ?

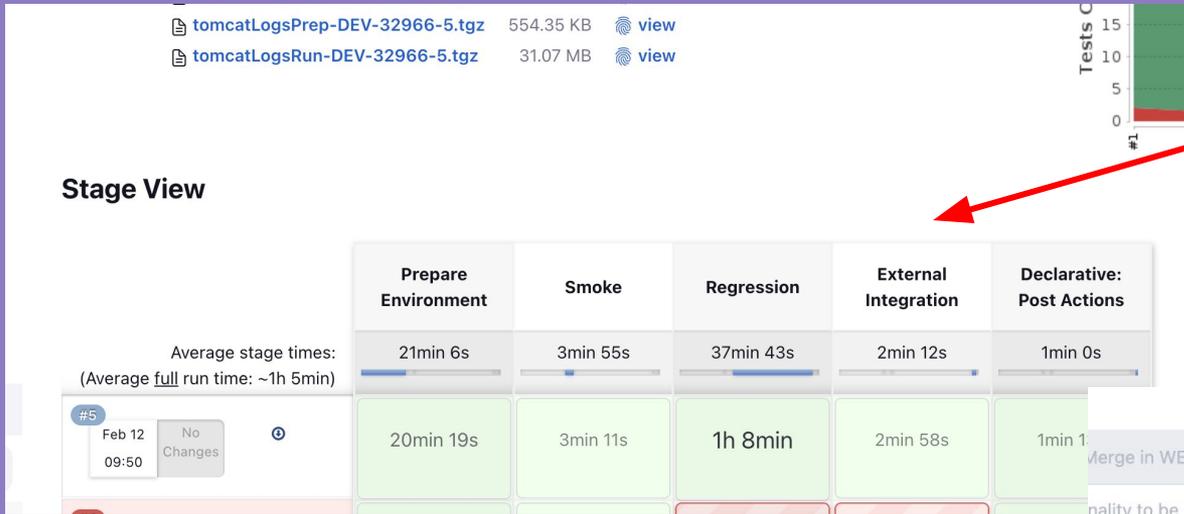


**Manuel**



**Automatique**

# Déploiement manuel Séparer les suites de tests

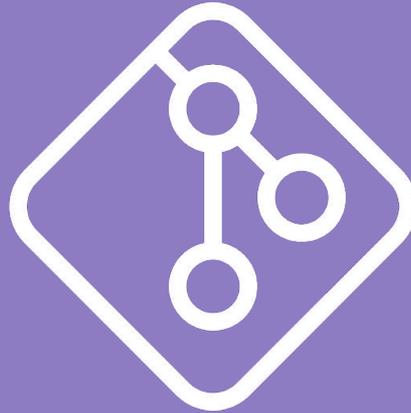


Séparer les tests à risque

	Commit date	Issues	Builds
Merge in WEB/wpg from DEV-3560...	30 Jan 2024	DEV-35605	⬇
2024.01.30.761 2024.01.30.761_FailedExternalInte gration	30 Jan 2024	DEV-31165	⬇
e day. Merge in WEB/wpg from DEV-35943 to	30 Jan 2024	DEV-35943	⬇
e Emails in Budget Notifications Merge in WEB/	30 Jan 2024	DEV-35950	⬇

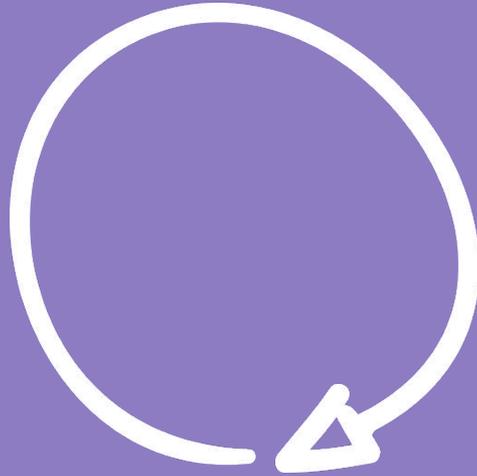
Un tag spécifique

# Déploiement manuel forcer le déploiement



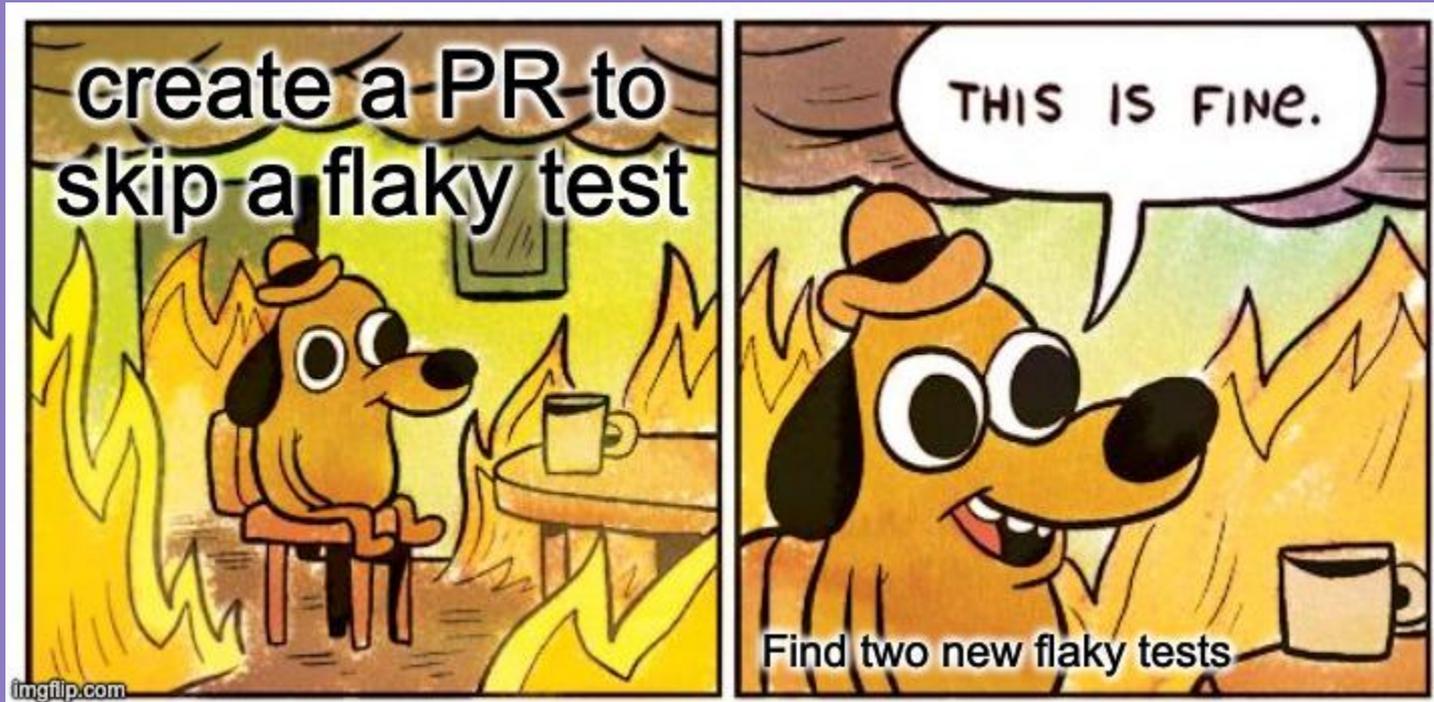
Forcer un  
merge

# Déploiement manuel forcer le déploiement



**Le comble du test instable : avoir un autre test instable qui empêche de le fixer**

# Déploiement manuel forcer le déploiement



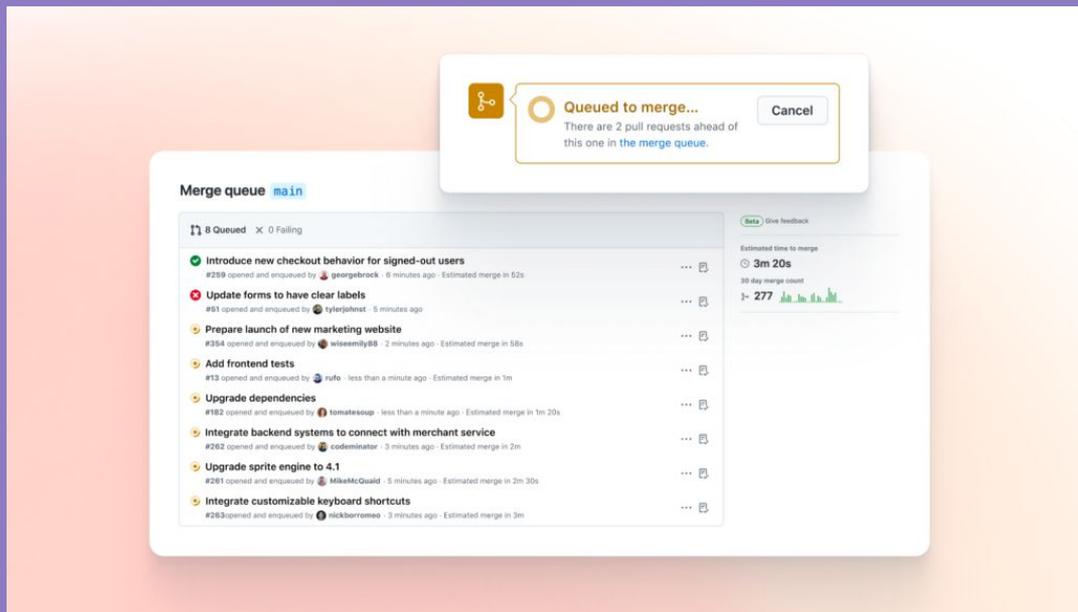
# Déploiement manuel forcer le déploiement



Vérification manuelle

'With great power comes great responsibility'

# Déploiement automatique : merge queue

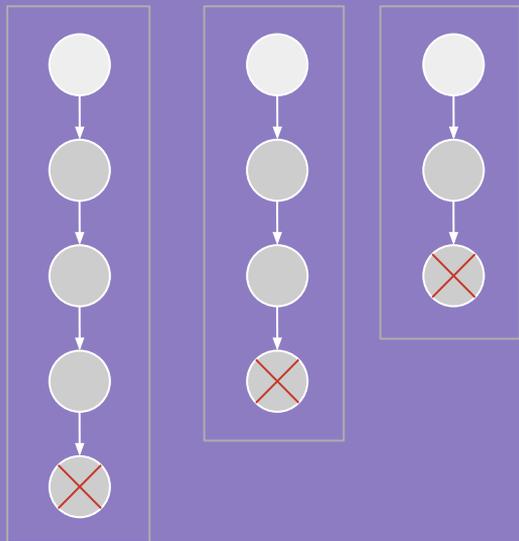


Automatisation du processus de merge et de création des releases

S'appuie sur la pipeline et les tests

On ne peut plus régler le problème manuellement

# Déploiement automatique : merge queue



Trouver des PR qui fonctionnent ensemble

Supprime un PR si la suite de tests échoue.

Les PR sont compatibles, mais un test instable perturbe les résultats.

Un seul test flaky  
peut bloquer la  
merge queue pendant  
des heures

## Éviter de bloquer la merge queue avec une stratégie de réessai

```
//mergequeue/retry-circleci-workflow.sh
```

```
TRIGGER_WORKFLOW="$(curl -s --retry 3 --retry-all-errors  
"https://circleci.com/api/v2/workflow/${WORKFLOW_ID}/rerun" \  
  --header "Circle-Token: $CIRCLE_TOKEN" \  
  --header 'Content-Type: application/json' \  
  --data '{"from_failed": true}')"  
echo "$TRIGGER_WORKFLOW"
```

Donner une nouvelle chance aux tests qui ne  
sont pas passés

# Combattre cela



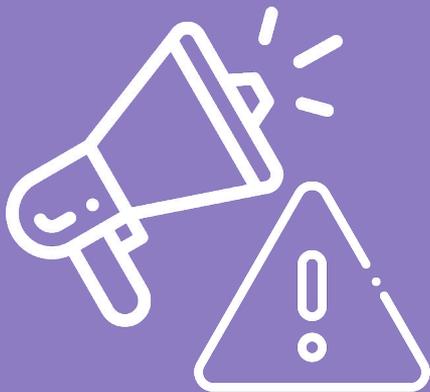
**Une stratégie de réessai augmente le temps d'exécution de la pipeline et cache le problème**

Combattre cela

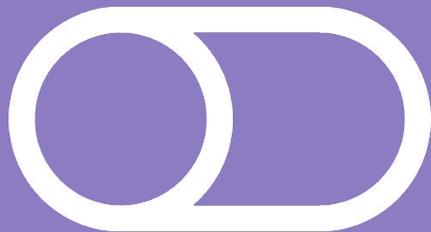


Définir un processus

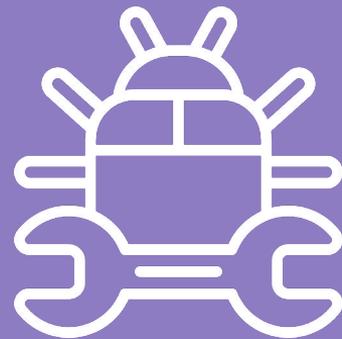
## Un processus clair



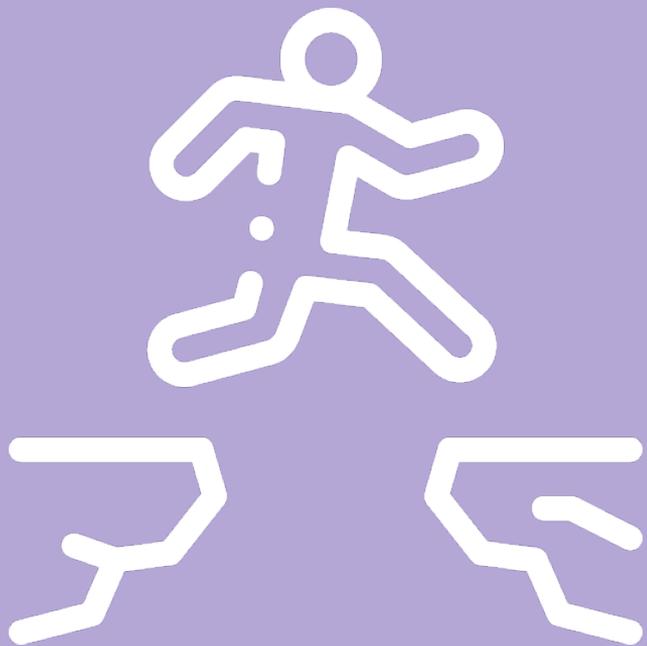
Reporter un  
nouveau flaky



Désactiver le test  
/!\ couverture code

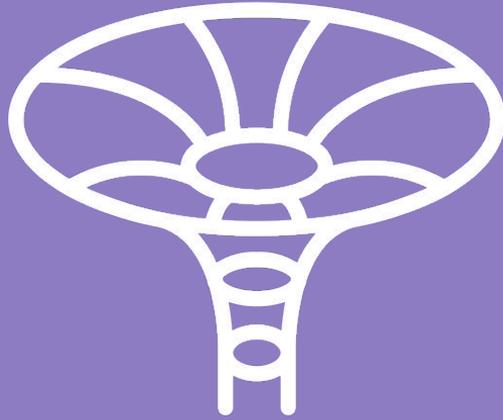


Dette technique  
urgente



Défis et limites

**Prendre contrôle de la situation au plus tôt**



**Plus le problème avance, plus il est difficile  
de le remédier**

**Tout le monde ne suit pas les règles**



**Nous avons un process mais il est plus simple de rejouer le test**

**Tout le monde ne suit pas les règles**



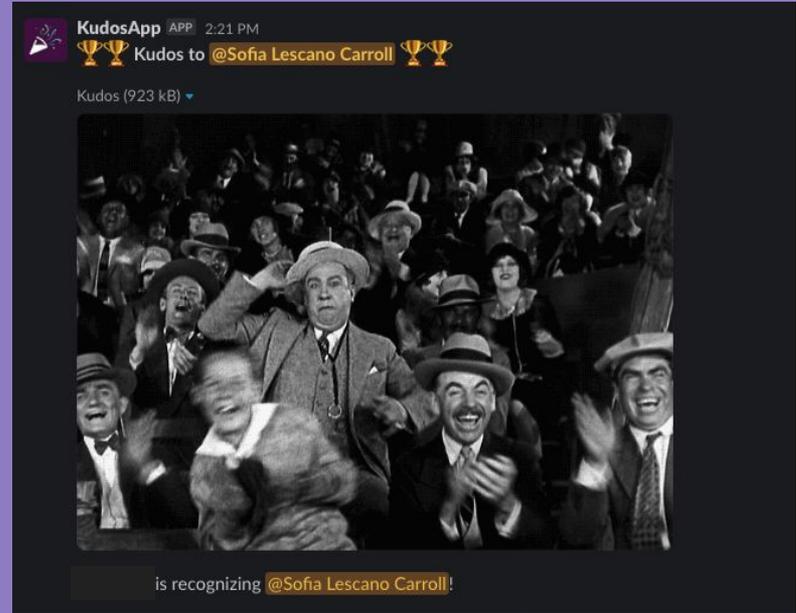
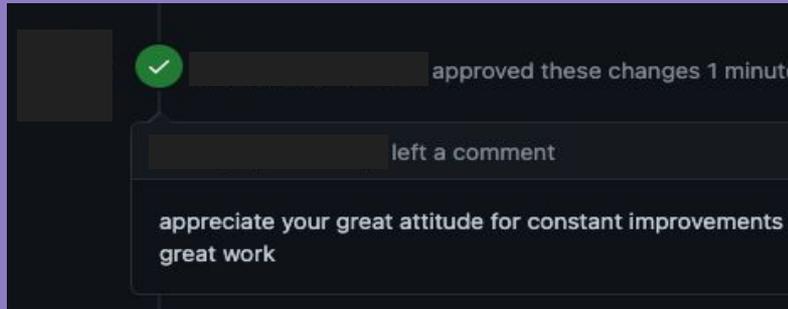
**Inspirer en donnant l'exemple plutôt que  
dénoncer**

**Tout le monde ne suit pas les règles**



**Valoriser le travail des personnes impliquées**

# Tout le monde ne suit pas les règles



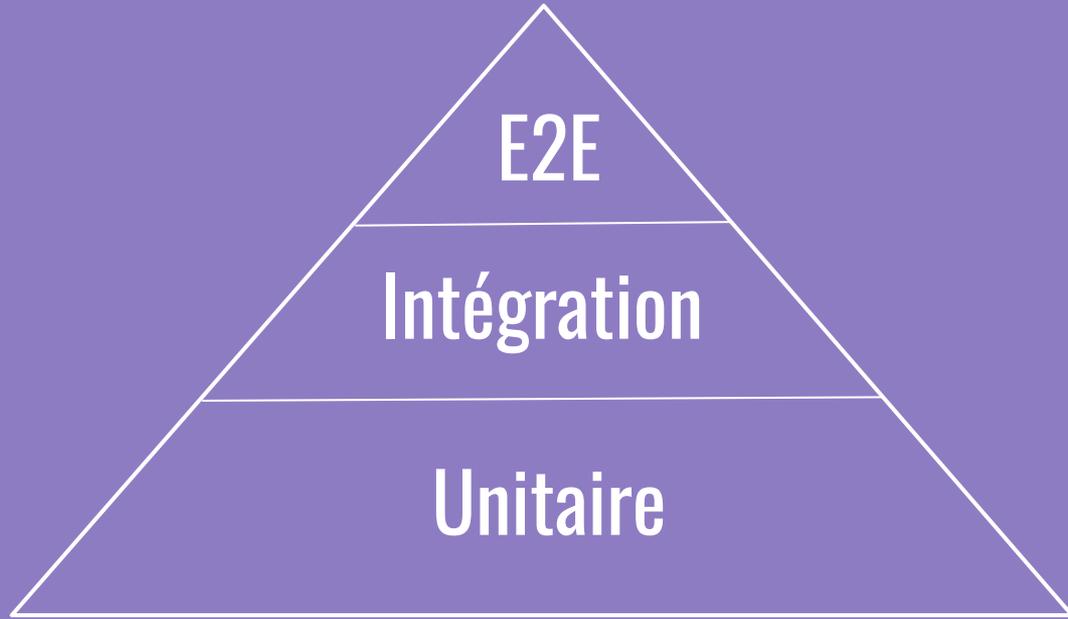
## Valoriser le travail des personnes impliquées

**Des instabilités difficiles à comprendre**



**Peut être que la réponse est finalement supprimer le test  
ou tester autrement**

# Des instabilités difficiles à comprendre



Peut être que la réponse est finalement supprimer le test  
ou tester autrement

Quand rien ne  
fonctionne, on peut  
au moins en rire

# Quand rien ne fonctionne : utiliser humour

FLAKY TESTS



1

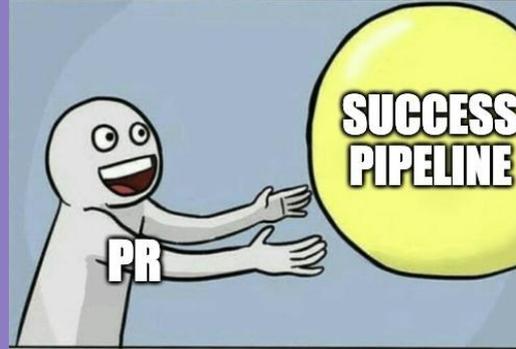
Sofia



0



Where are the tests ?



PR

SUCCESS PIPELINE



ME TRYING TO RELEASE

FIRST INCIDENT

E2E FAILS

SECOND INCIDENT



FLAKINESS

FLAKINESS EVERYWHERE



FLAKY TESTS

PR

SUCCESS PIPELINE

Une image vaut plus que mille mots

**juicy\_fish** **Handicon** **Berkahicon**  
**Freepik**  
**Parzival' 1997** **Iutfix**  
**Pixel perfect** **Iconixar** **Rukanicon**  
**Thank you !**  **Wahyu Adam**  
**Geotatah** **nangicon** **srip**  
**Monkik** **Smashicons** **M Karruly** **mynamepong**

Icons made by the artists mentioned, from [www.flaticon.com](http://www.flaticon.com)



**Thank you !**

# Nos tests nous empêchent de délivrer

**Sofía LESCANO CARROLL**



@SofLesc



**#fullRemote #PHP #Laravel #lifeInSpain #startup #doGood #quality**