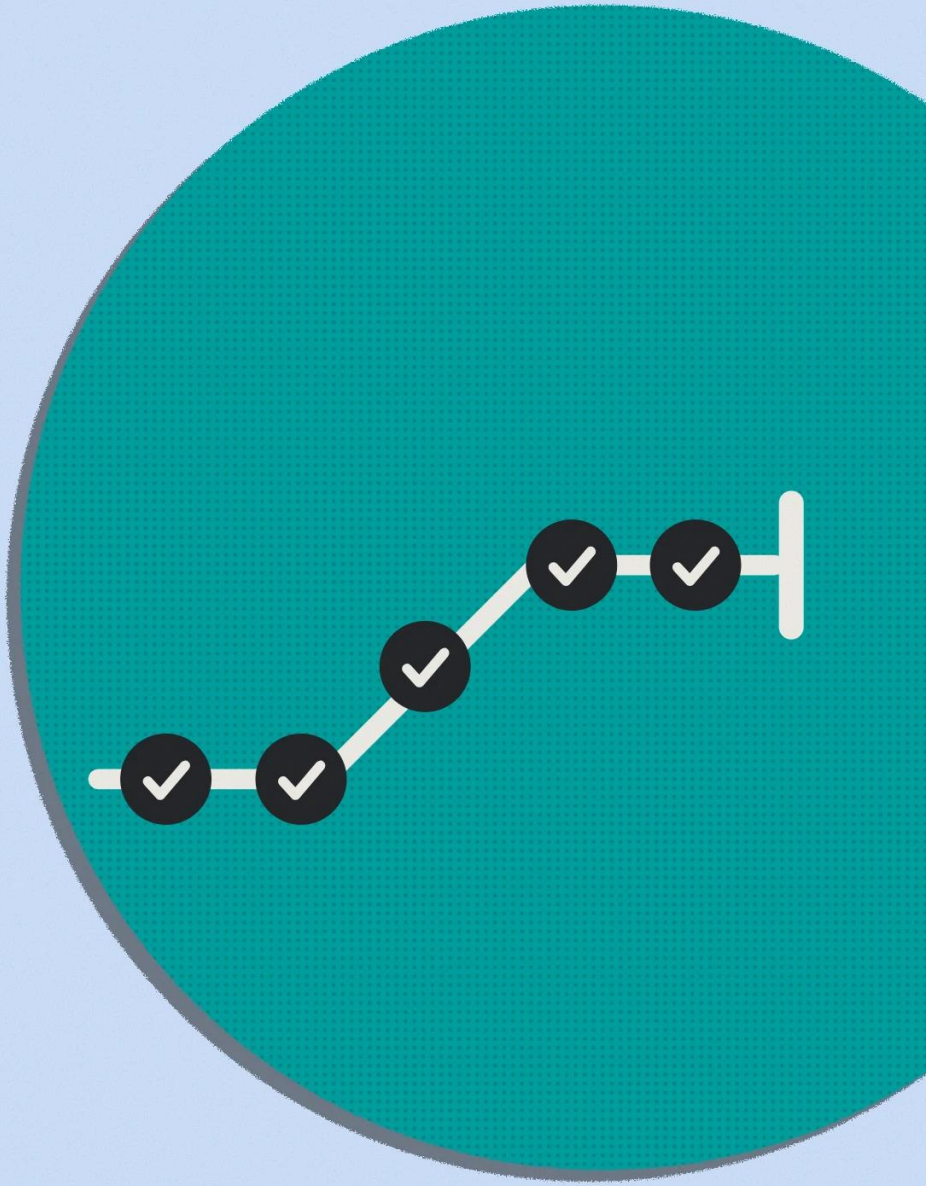
2.12

TEMPS 2 · LE TAMAGOTCHI TROUÉ · SÉANCE 2

Le bilan

Thibault Six · IFOSUP Wavre · 2026-2027



? Au temps 2, rien ne se remet. Comment savez-vous que c'est réussi ?

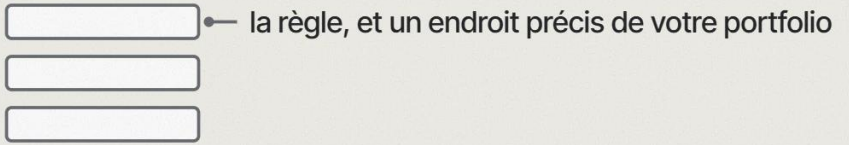
`composer test` vert prouve que les failles sont corrigées, le carnet que vous les reconnaissez.

1. Les tests — sur votre machine

```
$ composer test
```

vert de bout en bout

2. Le carnet des failles — pour vous



IL PROUVE

que vous avez corrigé les failles des chapitres 2.02 à 2.10 sans casser le jeu

IL PROUVE

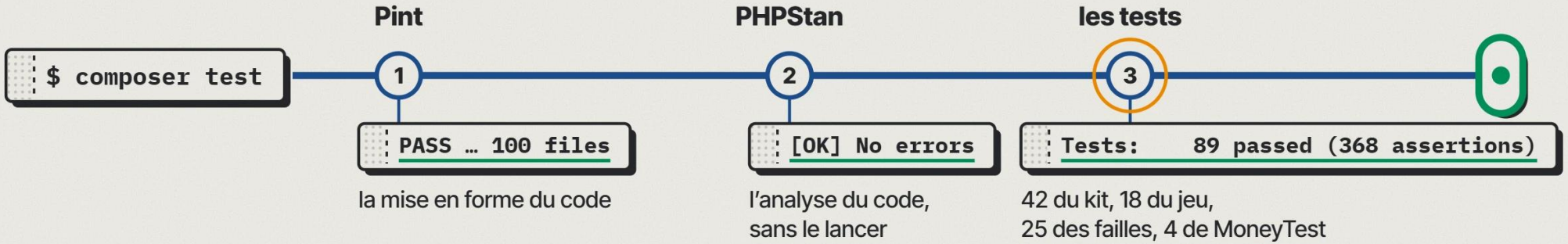
que vous reconnaissez ces failles dans votre propre code

Rien ne se remet : vous vérifiez vous-même que `composer test` est vert.

À gauche, la preuve par les tests ; à droite, la preuve par votre regard sur votre propre code.

? Au chapitre 2.01, composer test s'arrêtait à PHPStan. Que doit-il dire maintenant ?

`composer test` enchaîne Pint, PHPStan et les 89 tests, et doit finir vert de bout en bout.



AU CHAPITRE 2.01



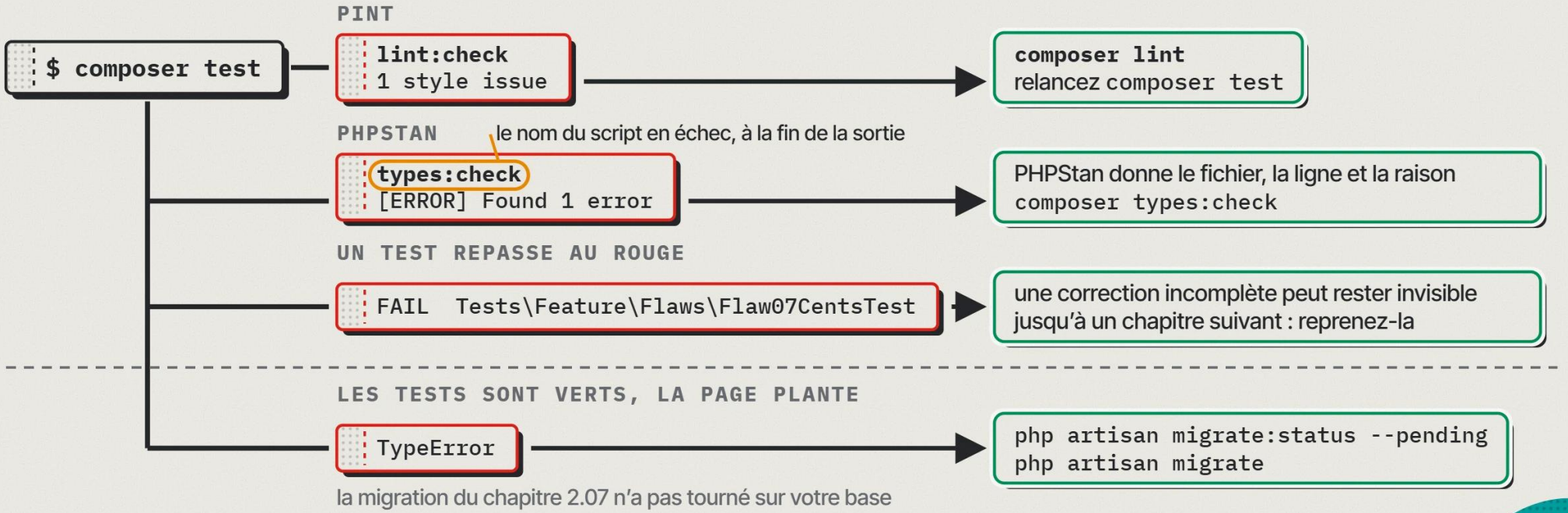

```
$ php artisan test
24 failed, 61 passed
```

les 25 tests des failles : 24 rouges, et un vert qui devait le rester

Lisez la sortie dans l'ordre des stations : mise en forme, analyse, tests. Le vert de la dernière est le critère de réussite.

? composer test finit en rouge. Comment savoir où regarder, et quoi faire ?

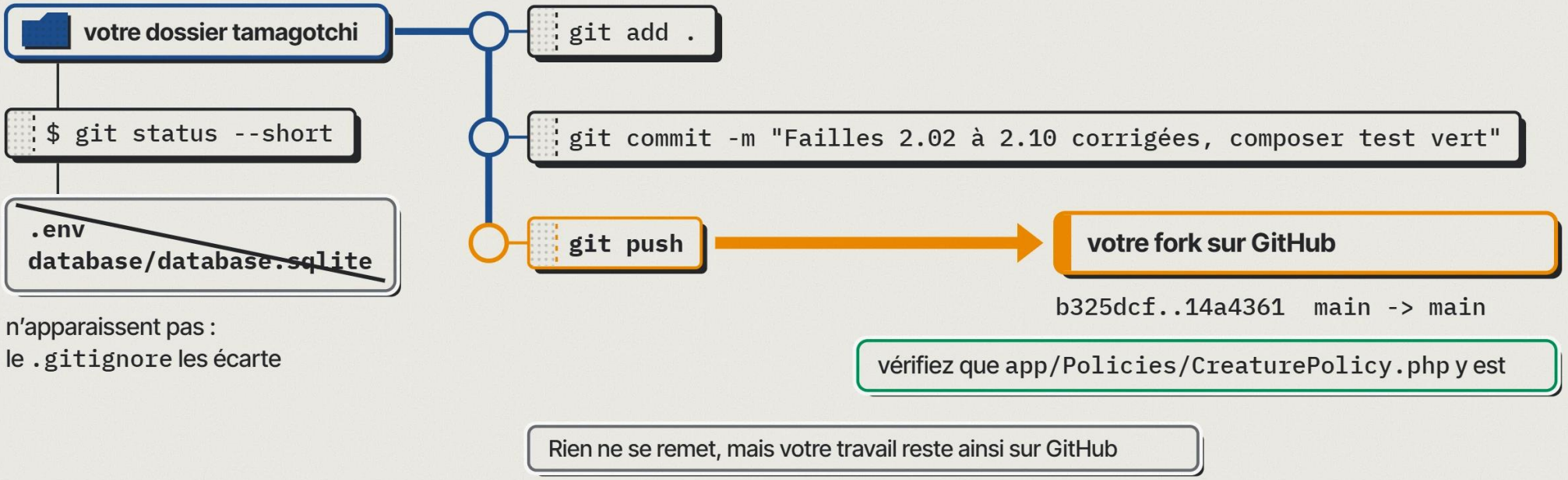
`composer test` s'arrête à la première étape en échec, et nomme à la fin le script qui a échoué.



Lisez la fin de la sortie : le nom du script dit quelle rangée suivre. La dernière panne ne se voit pas dans les tests.

? Votre dossier est le clone de votre fork. Comment y envoyer vos corrections ?

`git push` envoie vos corrections sur votre fork, sans `.env`.



Suivez la ligne : enregistrer, puis envoyer sur votre fork. Le fichier `.env`, qui porte la clé, ne quitte jamais votre machine.

? Une ligne du carnet dit « Dans les contrôleurs. ». Suffit-elle, et que lui manque-t-il ?

Une bonne ligne nomme une méthode, montre la vérification faite et donne une raison.

Faille	La règle	Dans mon portfolio
2.02 Changer le numéro dans l'adresse : le contrôle d'accès et la policy		

le gabarit

Nora Benali, le portfolio des exemples du temps 1	Chaque action sur une donnée vérifie côté serveur, avec une policy, que la personne connectée a le droit de la toucher. Un lien masqué ne protège rien.	1 `ProjectController@show` : `abort_if($project->published_at === null, 404)` refuse un brouillon. 2 Vérifié déconnectée sur `/projets/carnet-de-recettes` : 404. 3 Back-office : pas de policy, parce que l'inscription est fermée depuis le chapitre 1.10 et que le seul compte est le mien.

TROIS LIGNES CREUSES

« Il faut toujours vérifier les droits. »
la règle répétée, aucun endroit

« Dans les contrôleurs. »
plus tard, vous ne saurez plus lesquels, ni ce que vous y aviez vérifié

« Nulle part. »
sans raison, vous ne saurez plus si vous aviez cherché

En vert, la ligne du portfolio de Nora Benali : une méthode, une vérification, une raison. En rouge, ce qui manque aux trois autres.

? Votre carnet a trouvé un trou dans votre portfolio en ligne. Que faites-vous ?

Là où le carnet a trouvé un trou, corrigez le portfolio, relancez vos tests, puis redéployez.

À FAIRE, POUR LA LIGNE 2.02

1. Déconnectez-vous et ouvrez l'adresse d'un brouillon de votre portfolio.
2. Si son contenu s'affiche, écrivez le trou tel quel dans le carnet, puis corrigez-le.
3. Relancez vos tests, puis redéployez comme au chapitre 1.22.

DEUX QUESTIONS FONT LE BILAN

- [ ] `composer test` du Tamagotchi passe en entier : `89 passed`.
- [ ] Chaque faille du carnet a sa règle et un endroit précis, ou une raison honnête de n'en avoir aucun.

le projet en binôme reprend la policy du 2.02 et la liste de contrôle du 2.11

Trois étapes à gauche, sur votre portfolio ; les deux questions du bilan à droite.