

TDD, par la pratique.



Plan

- Ce que n'est pas le TDD
- Ce qu'est le TDD
- Quand faut-il le faire?

Exemple de cas pratique

Ce que n'est pas le TDD?

- TDD, n'est pas une architecture
- TDD, n'est pas du DDD (Domain-Driven Design)
- TDD, n'est pas du Test-First
- TDD, **n'a pas pour objectif de tester son application**, c'est plutôt une **conséquence** de cette pratique.
- TDD, n'est pas une perte de temps

C'est quoi le TDD?

Test-Driven Development

- Développement piloté par les tests
- **Pratique** qui te **guide** dans l'implémentation d'une solution à un certain problème
- Processus itératif, jusqu'à la solution finale
- Plus de code inutile
- Finir la peur de refactoriser

Comment ça marche?

01 02 03

ROUGE

J'écris le premier cas de test de la fonctionnalité à développer. Ce test échouera.

VERT

J'écris ensuite le minimum de code qu'il faut pour faire passer ce cas.

Ce test doit passer.

Refactorisation

J'améliore mon code.

L'objectif est de maintenir la qualité du code tout en ajoutant de nouvelles fonctionnalités

Quand fautil faire du TDD?

- Il n'y a pas de réponse magique à cette question
- Tu peux faire du TDD, du moment où tu as un usecase un peu sensible à gérer (avec plusieurs règles métiers, par exemple).
- Si c'est juste pour finalement, avoir une méthode à 1 ou 2 lignes gérant deux règles métiers, ça ne vaut pas le coup.

Cas pratique

Gestion du panier utilisateur, sur un site e-commerce.

https://github.com/ulrich-geraud/kata-tdd-ecommerce-php/tree/main