

# TDD de la vraie vie

AlpesCraft 2022

# Qui sommes-nous ?



Johan MARTINSSON

Coach tech  
Dev



Xavier NOPRE

Artisan Développeur  
CTO @ vrtice

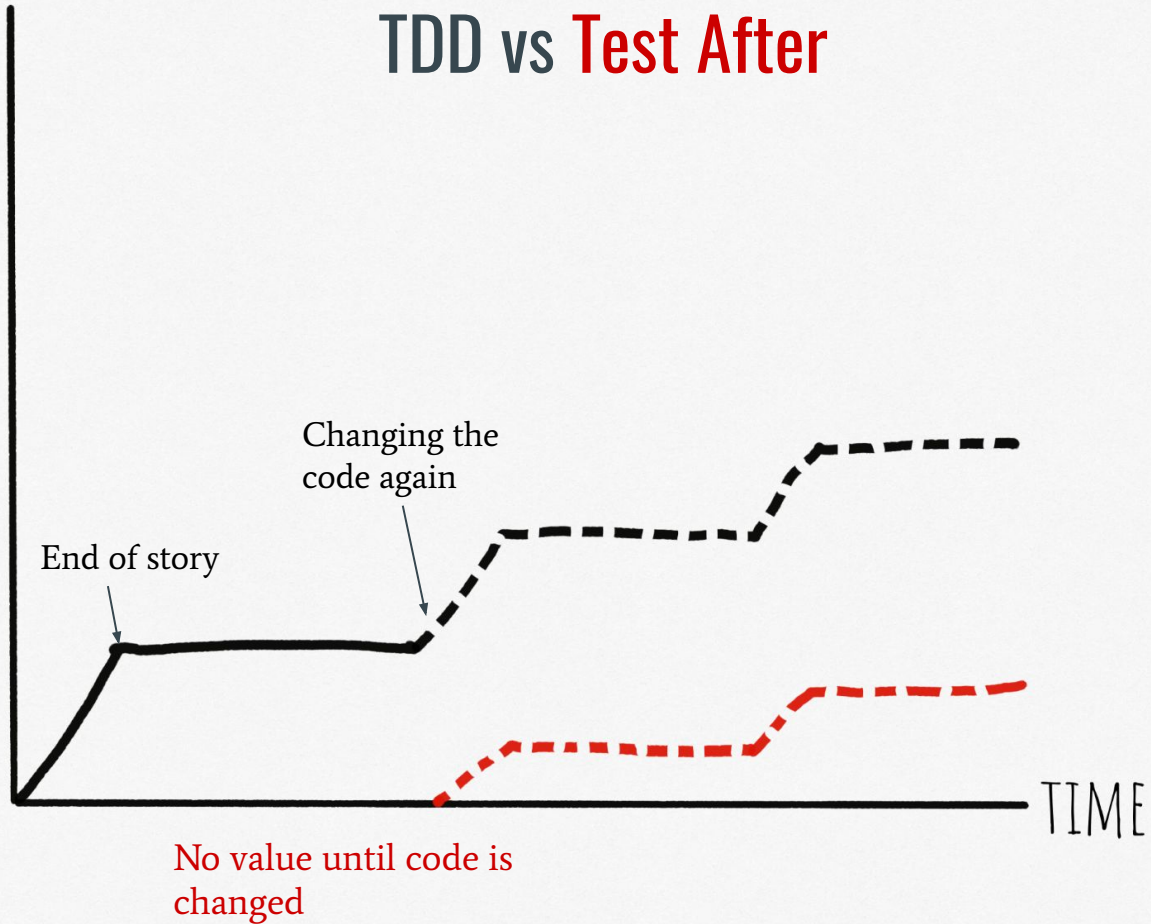
# Constat / Notre but

- Intérêt des tests unitaires ?  $\Rightarrow$  OK pour beaucoup de monde !
- Qui pratique le TDD régulièrement ?  $\Rightarrow$  ... 😞
- Quel est le problème ?  $\Rightarrow$  Passage à l'acte au quotidien

**$\Rightarrow$  Montrer notre approche du TDD pour la vraie vie !**

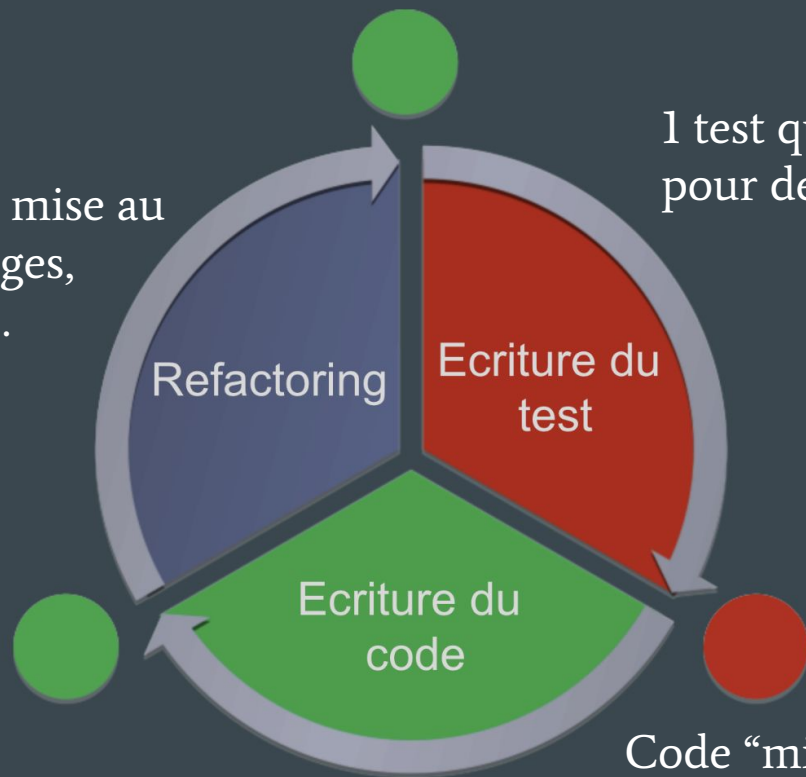
VALUE

## TDD vs Test After



# TDD : comment ?

Remaniements, mise au propre, nommages, factorisations, ...




1 test qui ne passe pas pour de bonnes raisons

Code “minimum” pour faire passer ce test

# De la valeur au plus tôt !

- TDD sur du code algorithmique  $\Rightarrow$  OK mais éloigné de notre quotidien
- "Ecrire une spécification exécutable" n'est pas spécifique au "bas niveau"
  - Cela se fait aussi bien à tout niveau comme par exemple sur des "test d'intégration"

# De la valeur au plus tôt !

- TDD sur algo  $\Rightarrow$  OK mais éloigné de notre quotidien
- "Écrire une spécification exécutable"  $\nRightarrow$  "bas niveau"
  - TDD  tous les tests

# Démo

## **Vous verrez**

- Du TDD en action
- Une approche de TDD basée sur l'usage
  - “Test d'intégration”

## **Vous ne verrez pas**

- De la logique métier complexe
- Du DDD
- De l'archi hexagonale
- Du code très propre

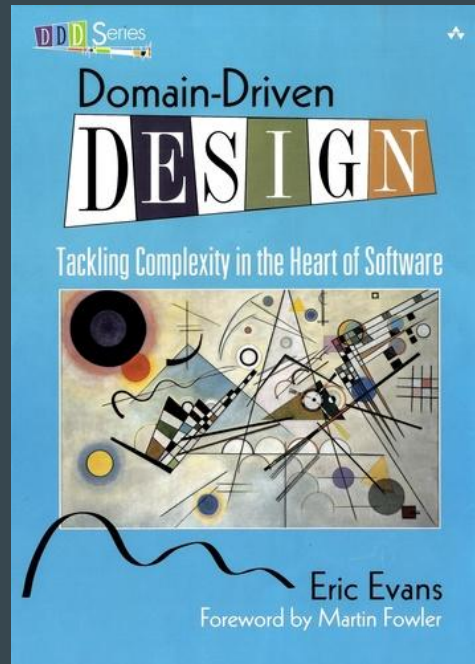


# Vous verrez

- TDD en action
- Du TDD sans TU



# Vous ne verrez pas



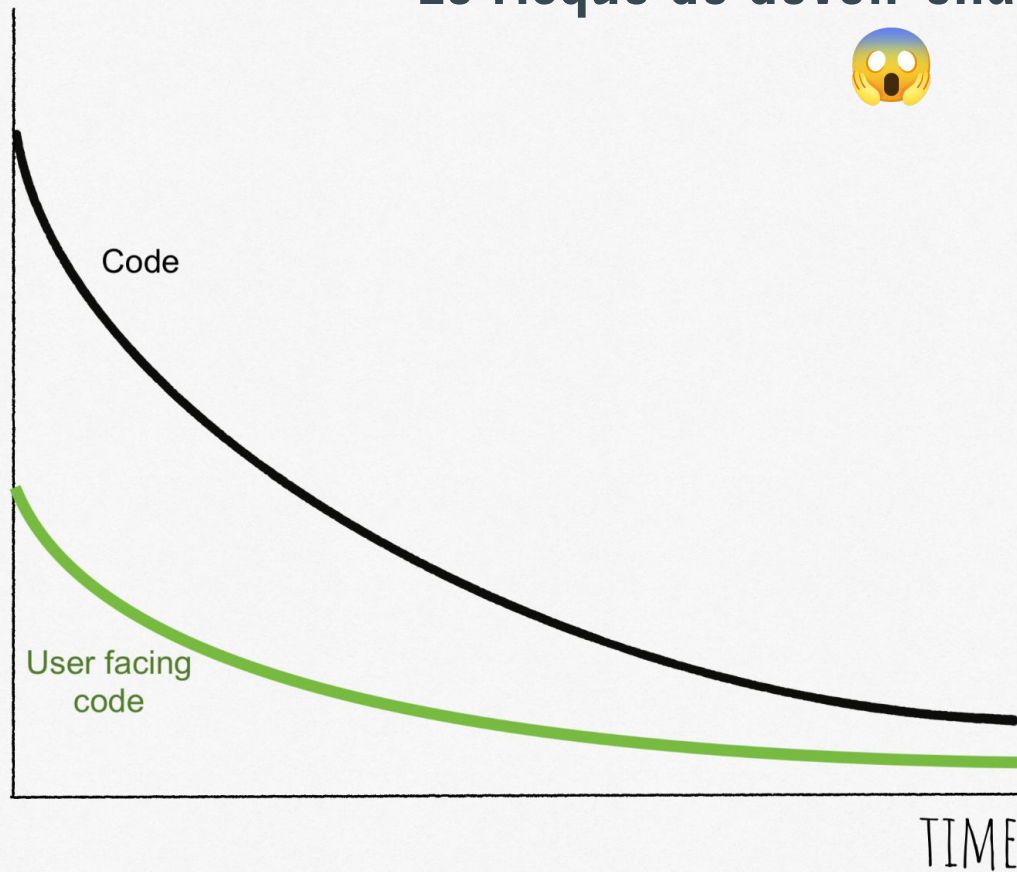
# Démo : contexte

- Application de vote pour votre langage préféré !
- Prototype front en approche “front first”

Démo !

RISK OF CHANGE

# Le risque de devoir changer les tests



# Problème

Modifier une interface sous test

⇒

- Pas de protection
- Modification du test



**Testons des interfaces stables**

**Comment reconnait-on une interface stable 🤔 ?**

1. Proche du besoin utilisateur/client
2. Vielle
3. Le ratio surface / volume faible.

# A retenir

- + Le TDD ça vaut le coup, et c'est faisable dans la vraie vie
- Respectez le cycle du TDD et les règles de base (pour apprendre)
- + Le TDD n'a rien à voir avec le type du test
- Ecrivez les tests qui vous aide à avancer et vous apporte vite de la valeur
- + Au début, le TDD est plus intéressant avec les tests proches de l'usage
  - Plus tard dans une story ou dans un projet les tests bas niveau sont incontournables
- Le besoin côté utilisateur est connu au début et stable dans le temps
- Le feedback immédiat a de la valeur

# A retenir

- Le TDD ça vaut le coup, et c'est faisable dans la vraie vie
- Le TDD n'a rien à voir avec le type du test
  - On ne doit pas forcément faire du TDD sur des tests unitaires bas niveau
- Pour du TDD sur du code nouveau, des tests proches de l'usage peuvent être plus intéressants
  - Plus tard dans une story ou dans un projet les tests bas niveau sont incontournables

**Merci !**  
**Des questions ?**