



Webinar technique

Initiation à la méthode TDD



Programme

Intervenants



Rémy JALLAN
Responsable développement Software
BE Software Toulouse



Quentin Martinez
Consultant Développeur.NET
Agence Rennes

Déroulé

| | |
|--------------|--|
| Introduction | Contexte et présentation |
| Partie 1 | La méthode TDD : présentation et objectifs |
| Partie 2 | Méthodologie |
| Partie 3 | Mise en application |
| Partie 4 | Live Coding |
| Partie 5 | Questions & Réponses |

Introduction

Contexte

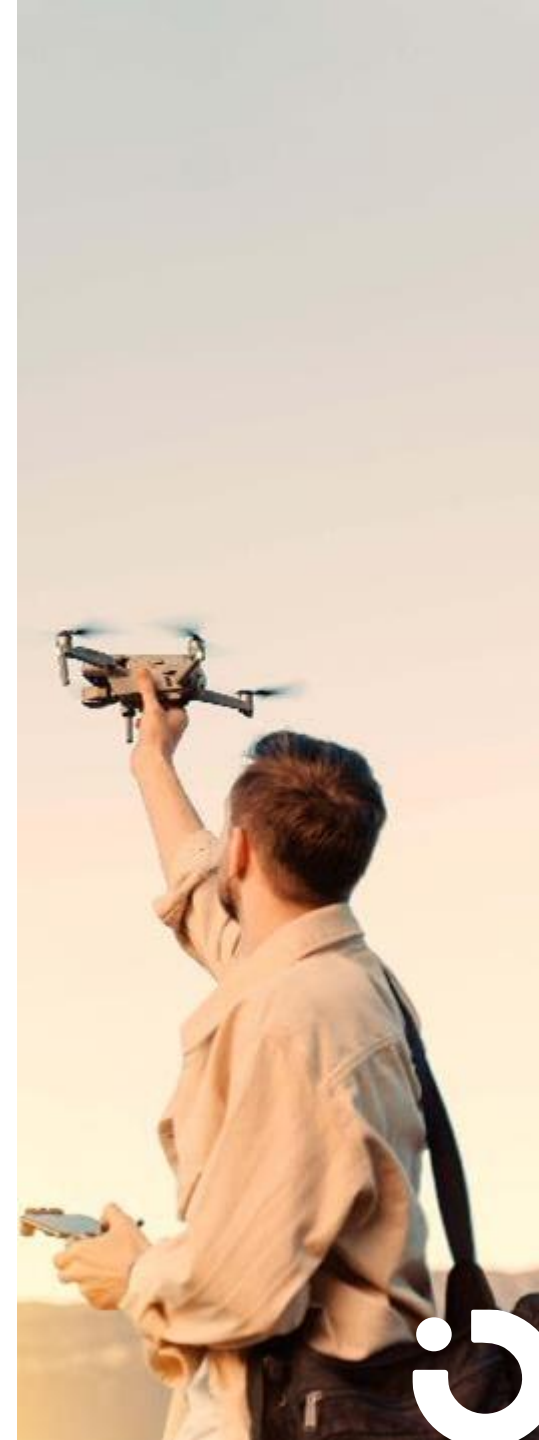
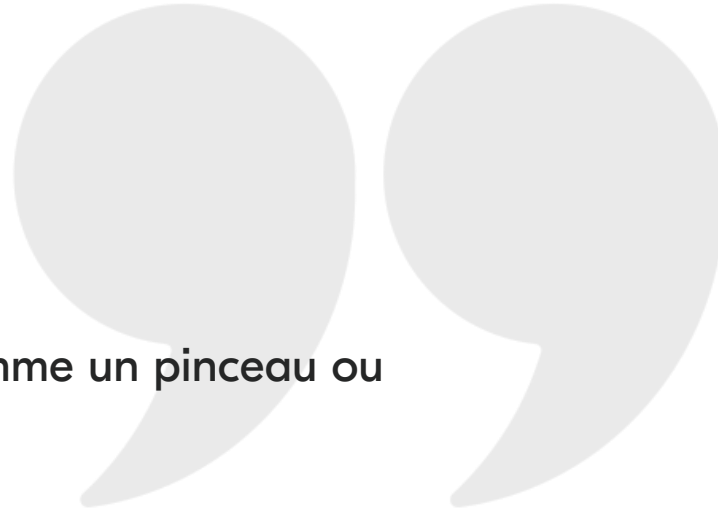


L'informatique n'est qu'un outil, comme un pinceau ou un crayon.

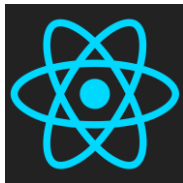
Griffo, Auteur de BD

Informatique : Alliance d'une science inexacte et d'une activité humaine faillible.

Luc Fayard, Auteur spécialisé



Présentation



Sources

Livres de Robert. C. Martin

Formations & Podcast de Michaël Azerhad

Articles spécialisés en ligne

Expériences professionnelles en lien avec le sujet

TDD, c'est un GPS jusqu'à l'algorithme parfait.

Michaël Azerhad, Développeur Expérimenté & Formateur

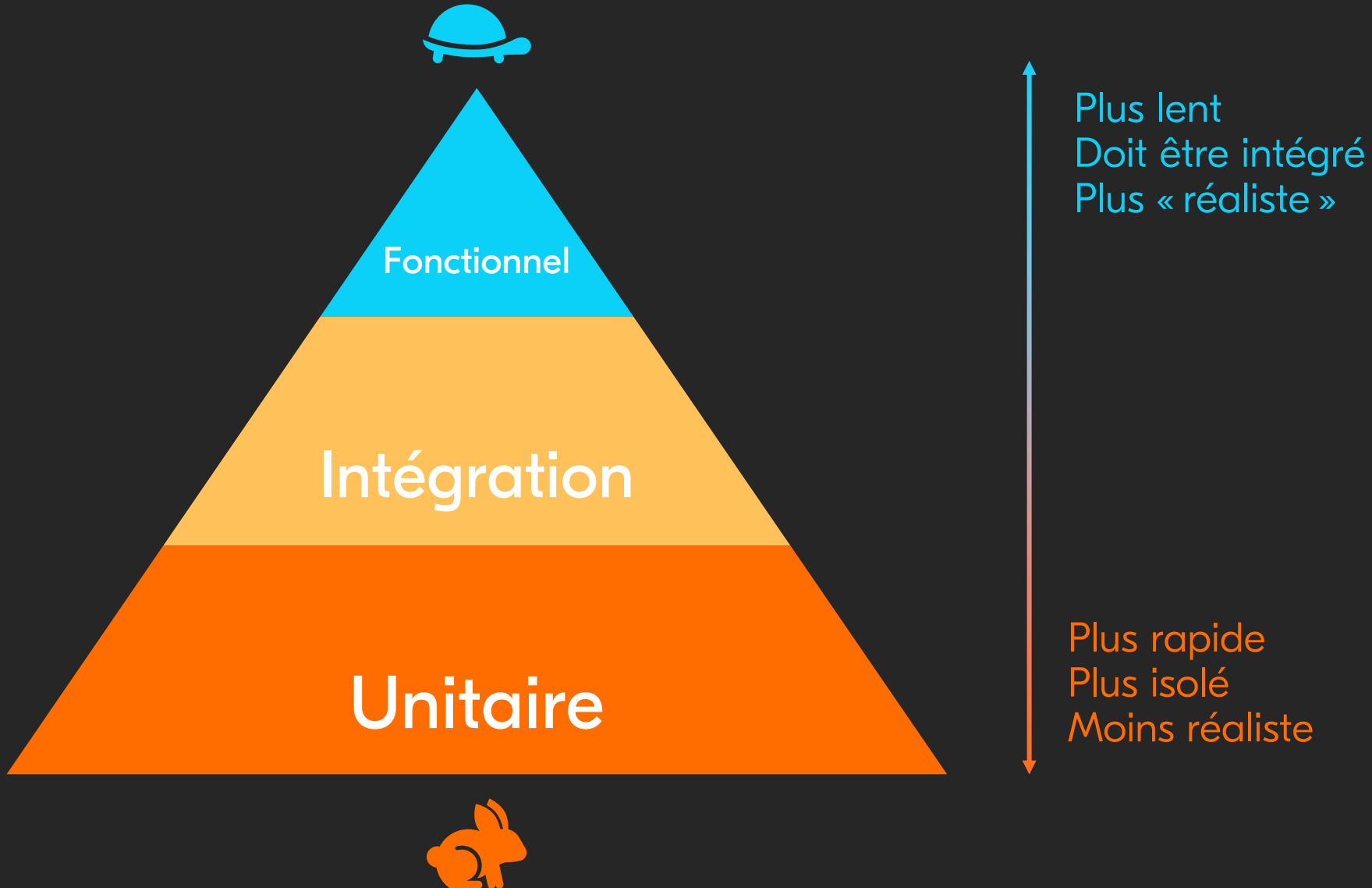
Historique

Co-inventé par Kent Beck et l'oncle Bob. (Robert C. Martin) et bien d'autres.

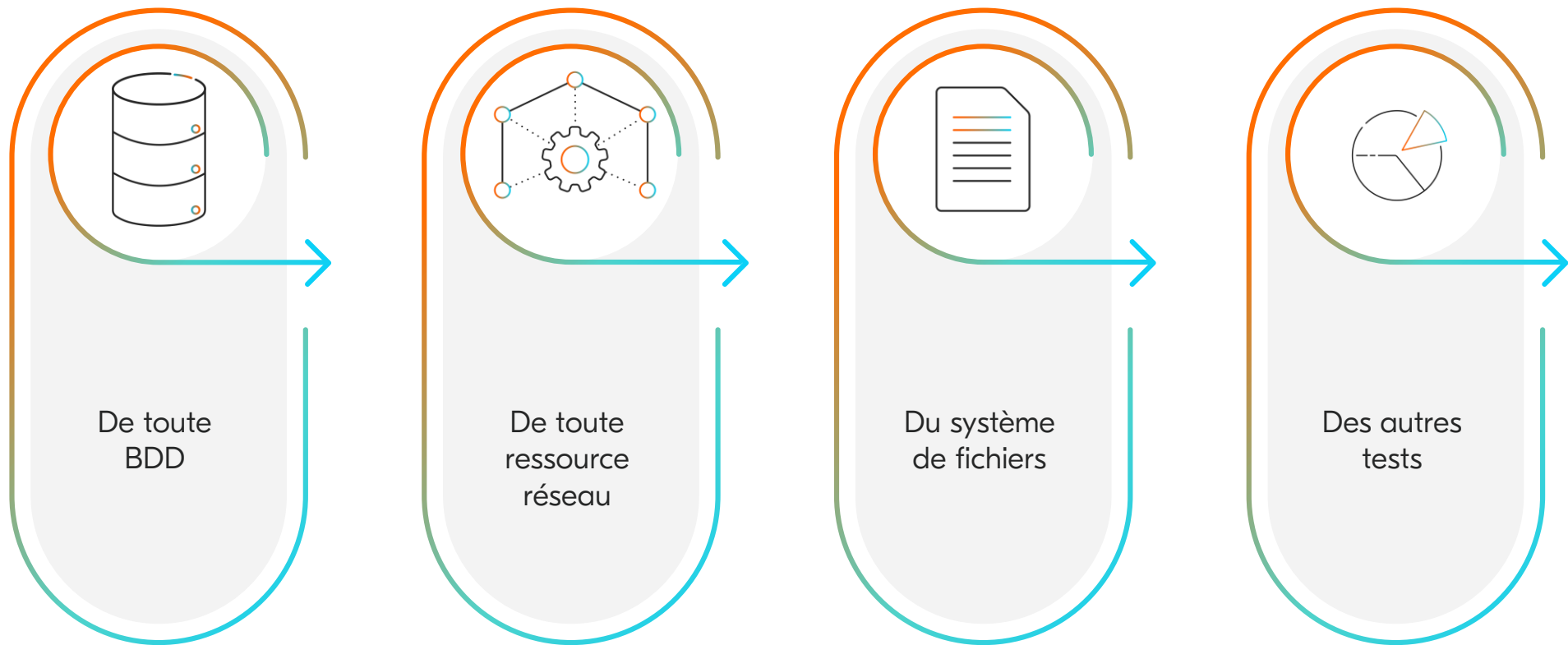
Kent Beck

- Inventeur de l'extreme programming
- L'un des premiers auteurs sur le TDD
- Co-inventeur de Junit
- Co signataire du manifeste Agile en 2001
- Conférencier international
- A travaillé pour Apple & Facebook

Les différentes catégories de tests :



Un Test Unitaire (TU) doit être indépendant :



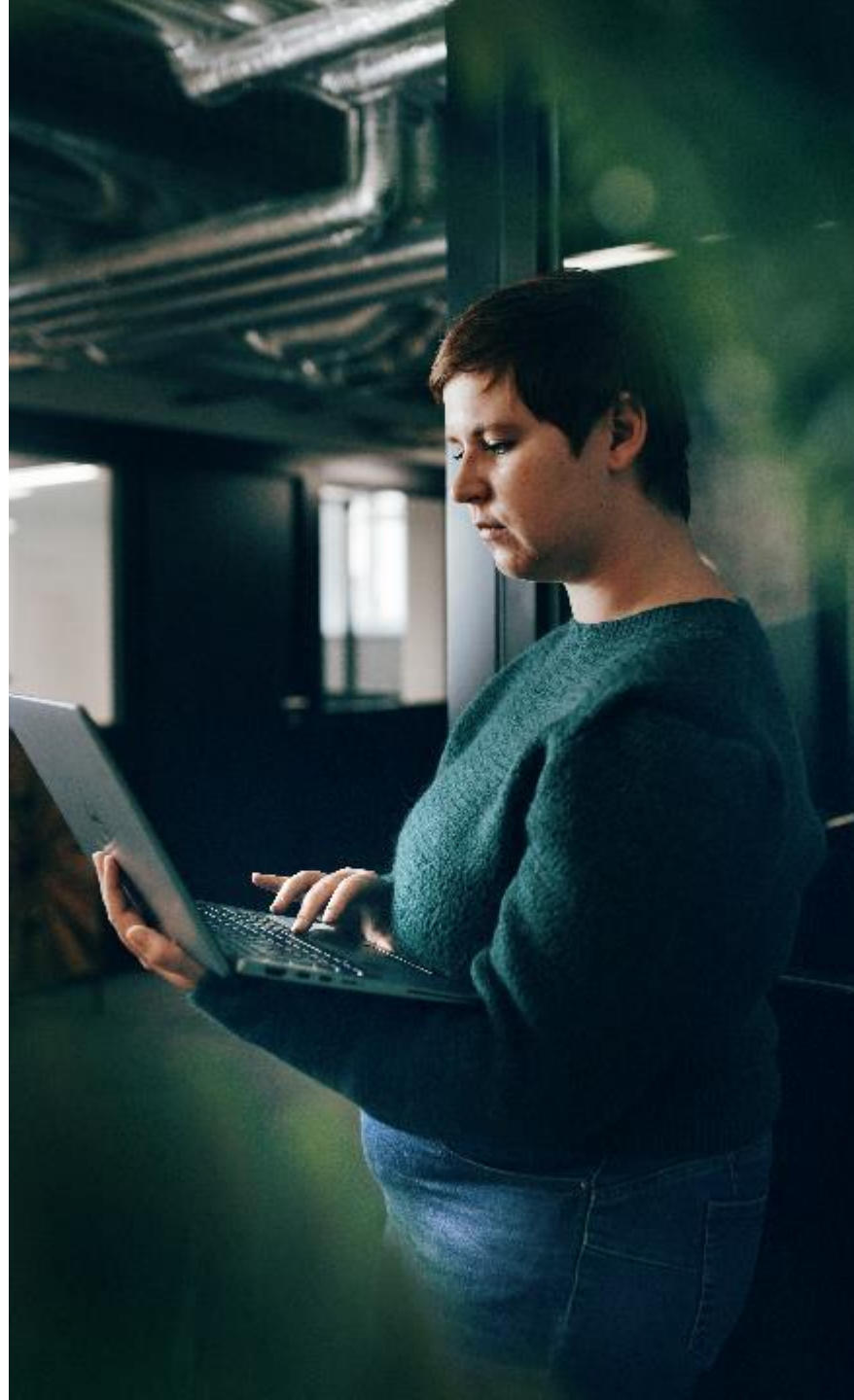
Partie 1 : Qu'est-ce que le TDD ?

Définition

Procédé, méthode de développement par les tests.

Conception de logiciel par des itérations successives très courtes.

L'approche TDD correspond à l'agilité appliquée à la programmation.



Objectif

Améliorer la qualité logicielle en réduisant les bugs.

Programmer efficacement des fonctionnalités complexes.

Favoriser la satisfaction et l'engagement client.

Comment maîtriser le TDD ?

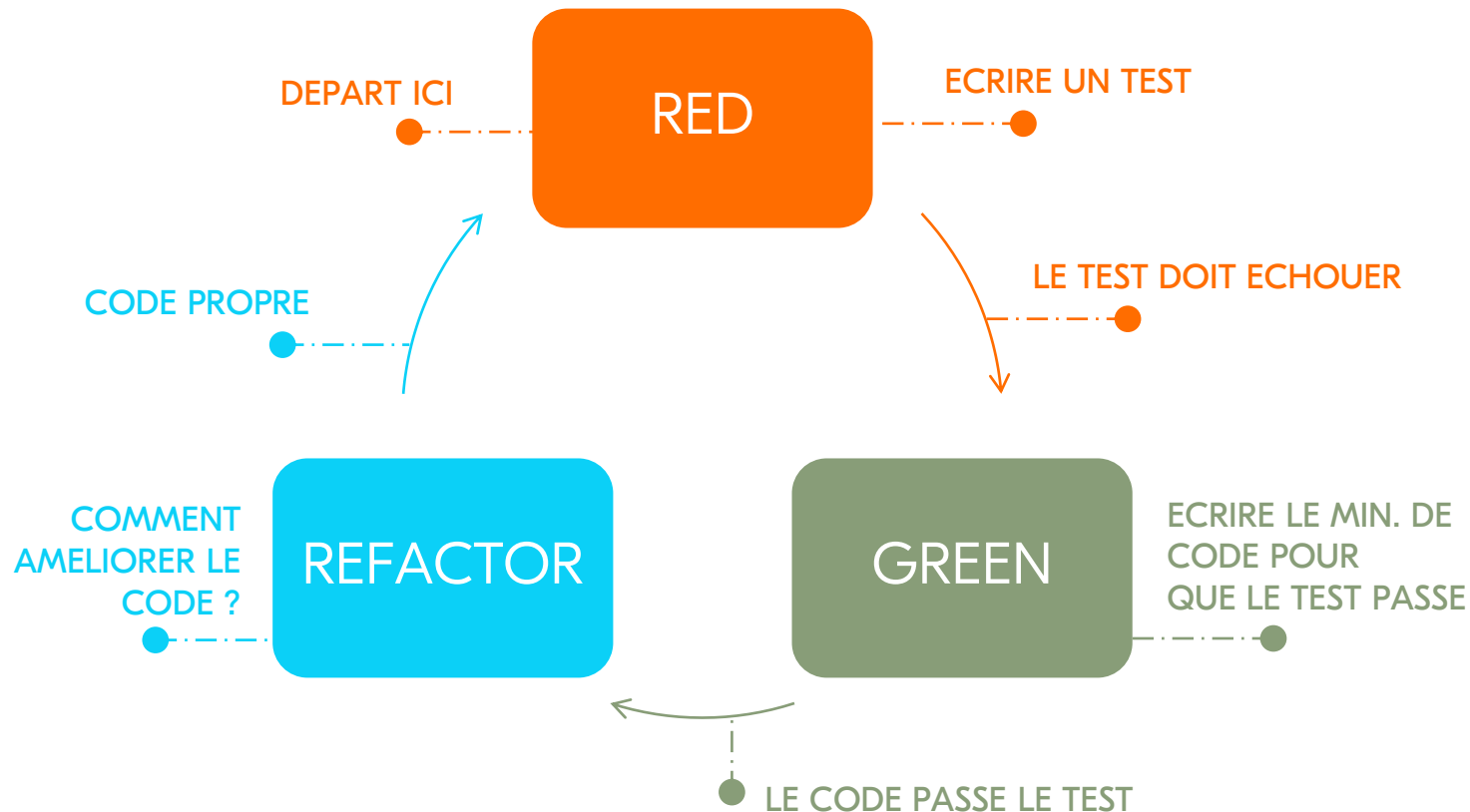
Réapprentissage.

La pratique d'exercices de plus en plus complexes.

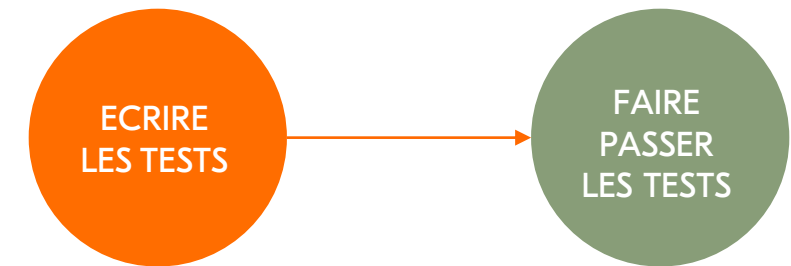


Différence Test Driven Development & Test-First Programming

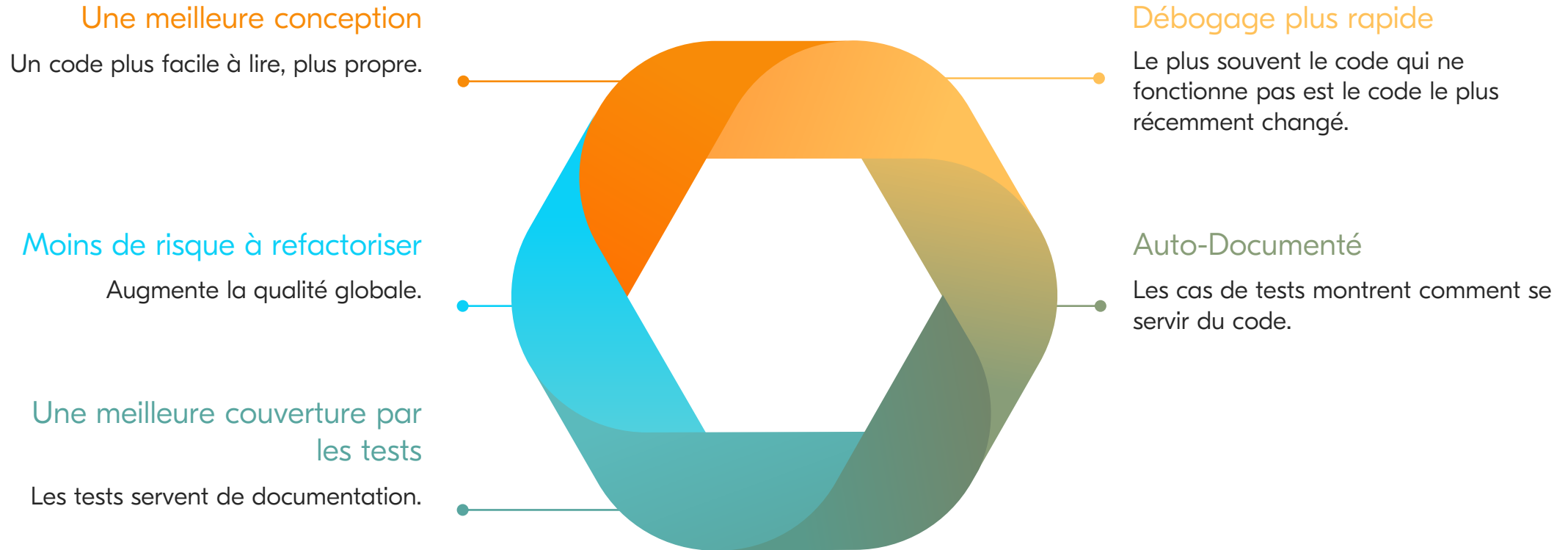
Test Driven Development



Test-First Programming



Les bénéfices



Partie 2 : Méthodologie

Les 5 commandements

Ces 5 commandements ne sont qu'une simplification de la méthode TDD. *Voir sources.*



Les freins usuels à l'utilisation de la méthode

Je ne peux pas faire cela sur mon projet parce que :



Je suis là pour écrire du code, pas des tests

Contrairement aux idées reçues, le développeur est beaucoup plus engagé dans la qualité et particulièrement en contexte agile.



Nous avons déjà des tests/testeurs

Les tests évoqués ici (test ihm, e2e) déjà présents sont bien souvent plus coûteux et plus long à exécuter.



Nous sommes sous l'eau, la deadline approche.

Se contenter de produire du code à toute vitesse, puis livrer et attendre les retours de la prod n'est pas professionnel. Tôt ou tard cela se payera.



C'est du code legacy, si je touche !

Bien qu'il soit moins aisé de tester dans ces conditions, il est de notre devoir d'entretenir l'application.

Je ne sais pas écrire un test unitaire

Cela tombe bien nous allons revoir cela!



Partie 3 : Mise en application

Expérience Professionnelle

Objectif 1^{er}: augmenter la capacité à réaliser des applications complexes

Sur logiciel de réservation en ligne (à l'instar de Planity/Doctolib)

Imaginons QUE

Une 1 prestation dure 1h. Horaires 9H — 19H. 1 seule ressource salariée

1^{er} Test

Quand_AucuneRessourceDisponible_ShouldReturn_0Creneau

2^{ème} Test

QuandDureePrestation1H_Et_DisponibleTouteLaJournee_ShouldReturn_10Creneaux



RESERVED

Le Kata – Le Bowling en TDD

Une partie = 10 tours /Joueur.

1 tour = 2 essais (Max)

Score de base = Nb quille tombés après les 2 lancés.

Si le joueur fait tomber:

- les 10 quilles en 2 lancers => **Spare**.

Score d'un **Spare** = 10, auquel on ajoute le nombre de quilles tombées au lancer suivant.

- les 10 quilles en un seul lancer => **Strike**.

Score **Strike** = 10, auquel on ajoute le total de quilles tombées aux deux lancers suivants.

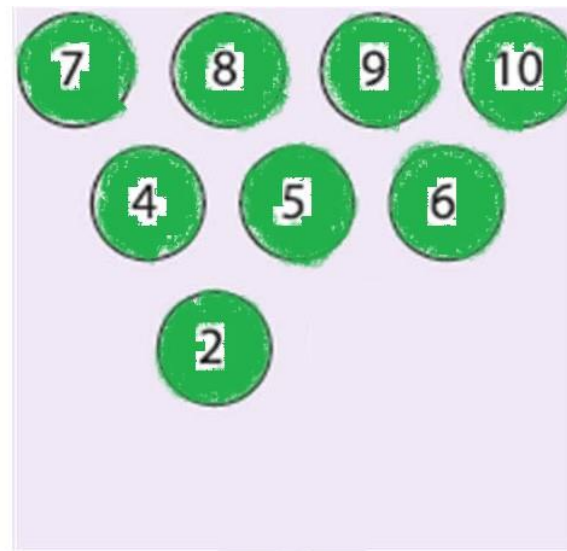
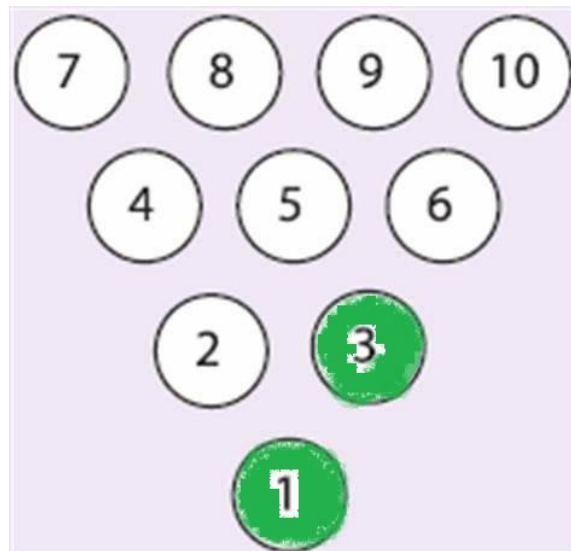
Lors du dernier tour :

- Si **Spare**, le joueur gagne **un** lancer supplémentaire
- Si **Strike**, le joueur gagne **deux** lancer supplémentaire



Exemple Spare

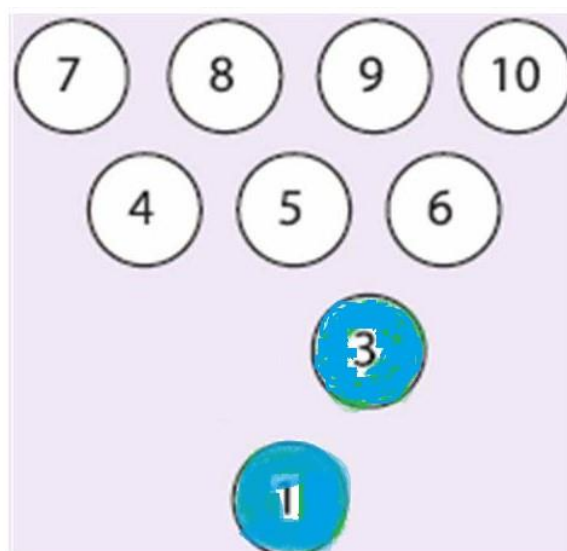
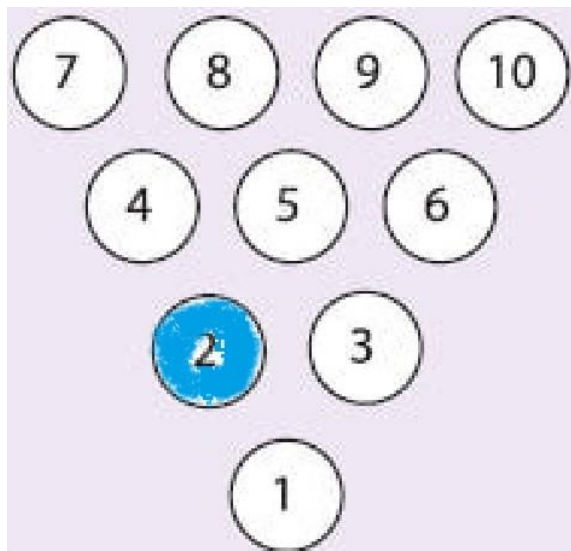
Tour 1



Résultat:

$$2 + 8 + 1 = 11$$

Tour 2



$$1 + 2 = 3$$

Score Total: 14



Partie 4: Live coding

Quel est le **premier** cas le plus simple ?

Premier test

Une partie faite uniquement de gouttière

C'est la partie la plus simple à implémenter,
10 tours x 2 Lancés x 0 Quilles = 0 Points



Partie 4: Live coding

Quel est le **second** cas le plus simple ?

Second test

Une partie faite d'une quille tombée par lancé

C'est la seconde partie la plus simple à implémenter,
10 tours x 2 Lancés x1 Quilles = 20 Points



Partie 4: Live coding

Quel est le **prochain** cas le plus simple ?

Troisième
test

Une partie avec un Spare puis 3 quilles tombées

SPARE $(6 + 4) + 3 = 10 + 3 + 3 = 16$



Merci pour votre attention !

Q&A