

SOMMAIRE

1. Le test Logiciel
2. Le Test automatique
3. Principes fondamentaux du Développement dirigé par les tests : *TDD Test Driven Development*
4. Framework Junit
5. Framework de tests automatisés

Abstract geometric lines in the top left corner, consisting of several overlapping, irregular polygons and lines in a light brown color.

PITCH DECK

Sacha Belisle

DÉMARCHE DE TDD

1. Identifier qu'est ce qu'on va tester (coder😊).
2. Coder son test !
 - Il faut disposer d'un Framework de test permettant de créer et d'exécuter des tests automatisés.
 - bibliothèque Java : JUnit
 - Un IDE : Eclipse



DÉMARCHE DE TDD

Identifiez le système à tester

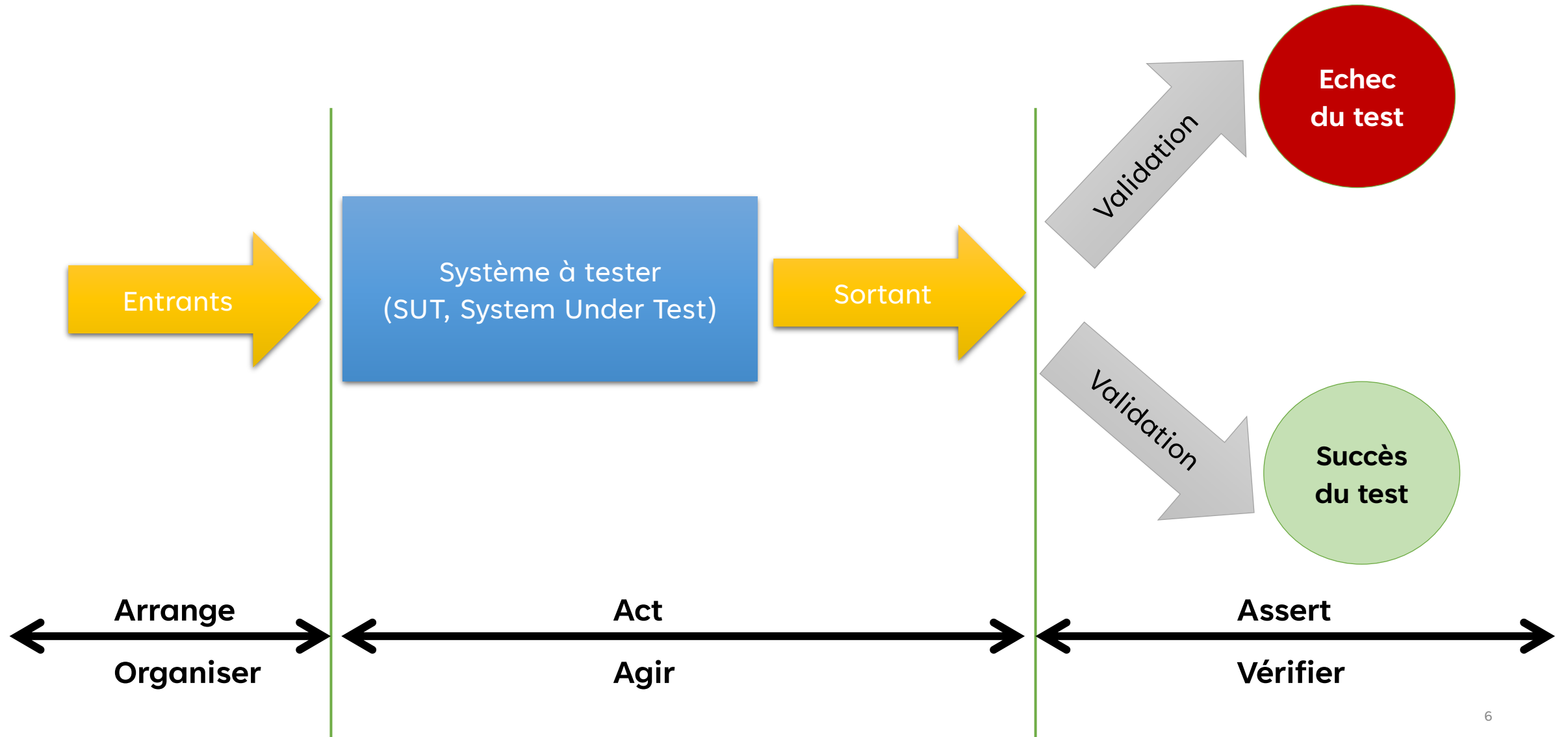
- Le système à tester ou **SUT** (system under test). Le système à tester donne des résultats (ou plus généralement des **sortants**) à partir de données ou de paramètres de test (ou plus généralement des **entrants**). Puis, en vérifiant les résultats, le test est déclaré en succès ou en échec.

DÉMARCHE DE TDD

Structure d'un test : Méthode AAA : Arrange - Act - Assert :

- **Arrange** (organiser) : initialisez tous les entrants nécessaires et le système à tester si besoin.
- **Act** (agir) : exécutez le système à tester avec les entrants précédemment initialisés dans des sortants que vous conservez.
- **Assert** (vérifier) : validez les sortants en fonction de ce qui est attendu par rapport aux entrants. Vous concluez alors si c'est en **succès** ou en **échec**.

DÉMARCHE DE TDD



DÉMARCHE DE TDD

Avantages de l'automatisation

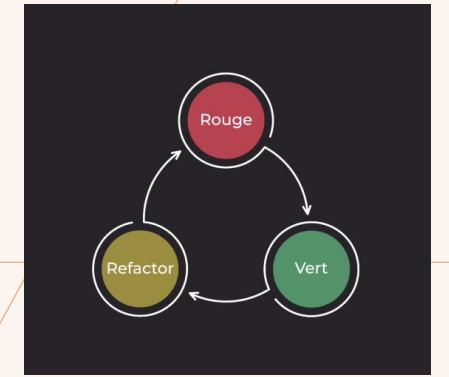
- On se focalise sur le système à tester
- On ne perd pas de temps sur la manière dont les tests vont être exécutés par les outils.
- C'est le Framework de tests, et qui s'occupe de tout l'aspect automatiquement ensuite.

CYCLE DE DÉVELOPPEMENT TDD

Cycle rouge-vert-refactor

- Inventé par Kent Beck
- Le TDD, consiste à ce que le codage de l'application suive un plan fixé par les tests.
- On a plutôt tendance à faire l'inverse, coder l'application, puis la tester de manière manuelle ou automatique.
- Toutefois, le modèle en V a préconisé partiellement cette démarche.
- Mais en codant d'abord le test, on se demande directement quel objectif doit accomplir le code.
- Le TDD suit le modèle *red-green-refactor*

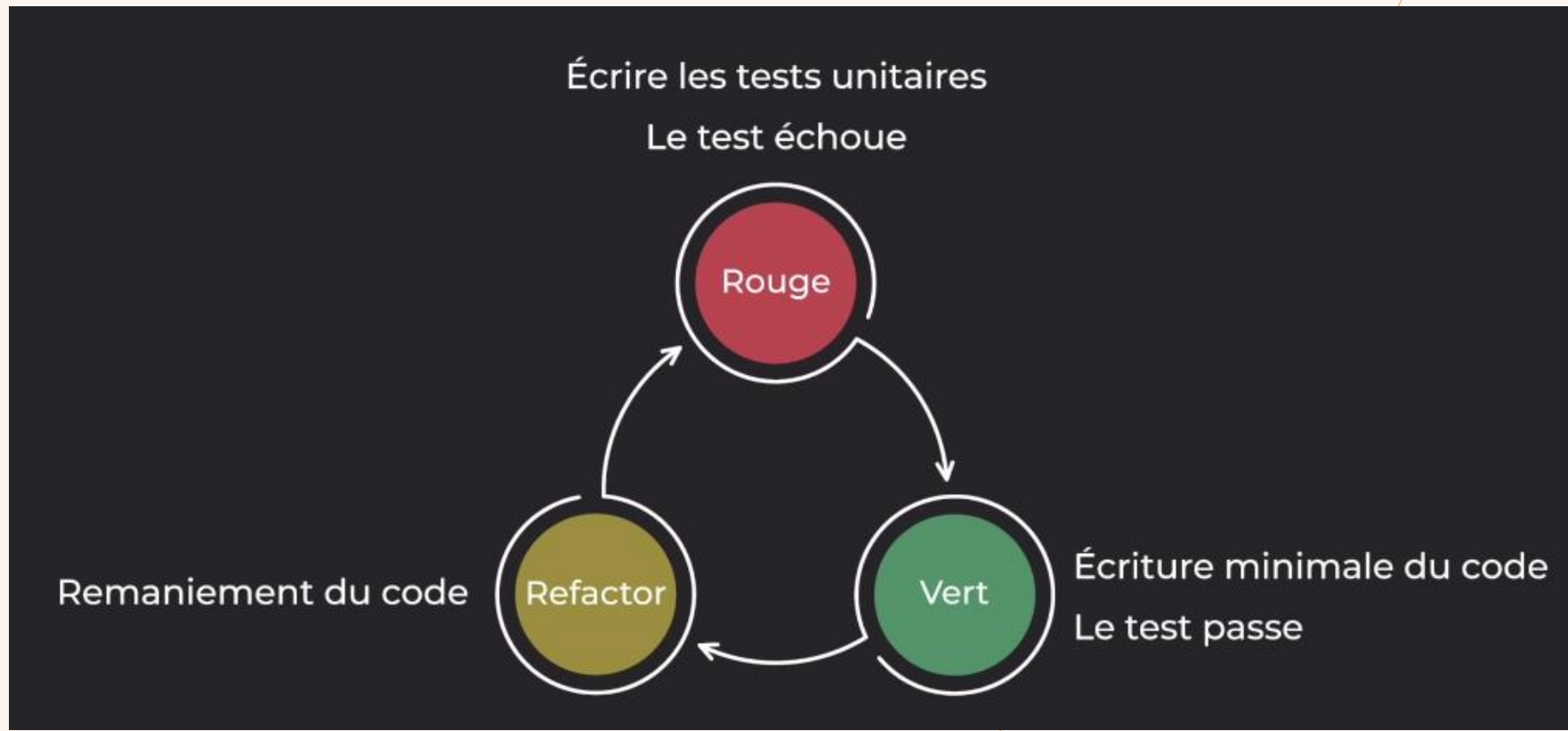
CYCLE DE DÉVELOPPEMENT TDD



Le cycle Rouge-vert-refactor

- **Rouge** : on écrit un test simple qui échouera avant même d'avoir du code qui l'accompagne. Le test échoue, sans le code. Du coup, le test est rouge.
- **Vert** : on écrit le code le plus simple possible pour faire passer le test, même si le code est simplifié. Le test réussit, il est donc vert.
- **Refactor** : améliorer/compléter le code, le rendre plus lisible et/ou plus sécurisé, plus évolutif sans changer son comportement.
- Cette démarche donne un code opérationnel bien structuré et évolutif

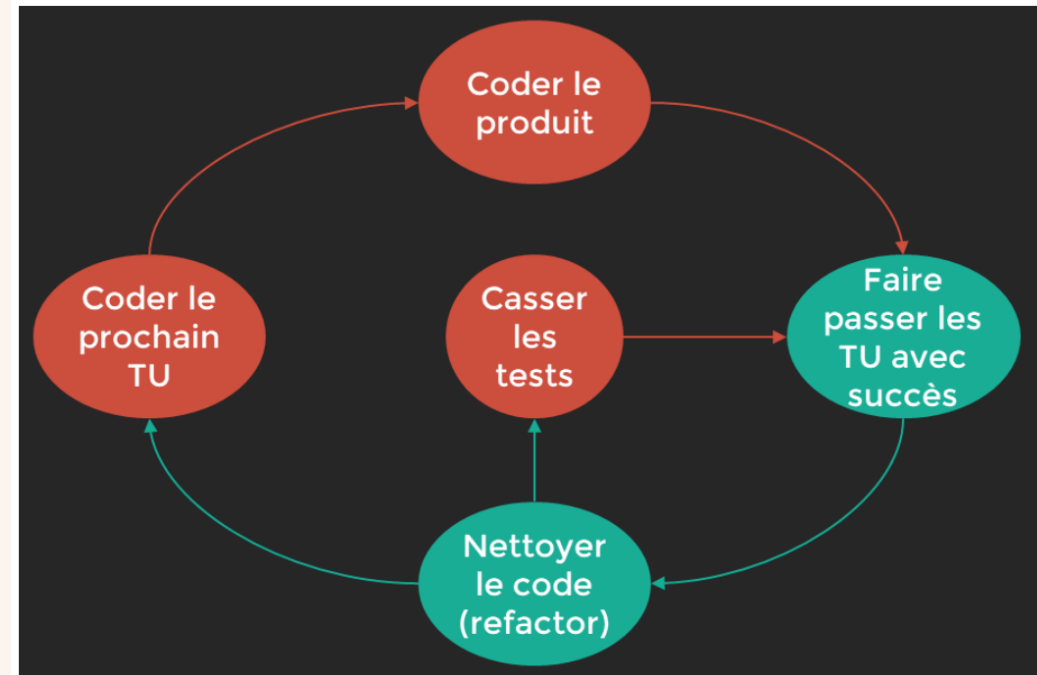
CYCLE DE DÉVELOPPEMENT TDD



STRATÉGIE DE FAIRE PASSER LES TESTS

Dans ce modèle, vous répétez cycliquement les étapes suivantes :

1. Écrire un test unitaire qui échoue.
2. Écrire le code qui permet de réussir le test.

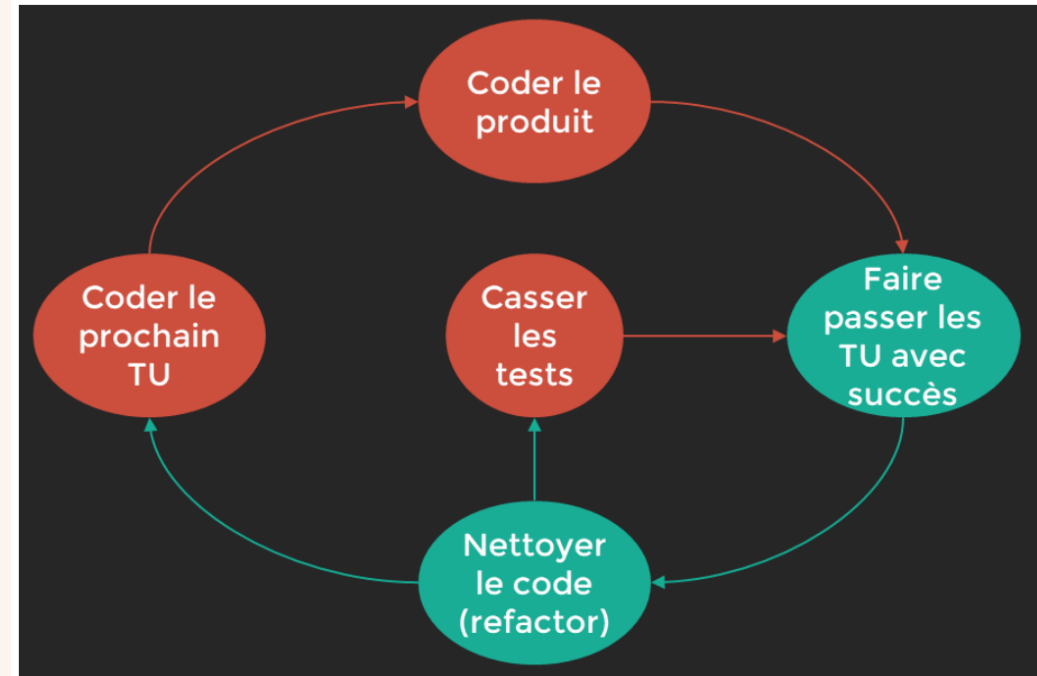


STRATÉGIE DE FAIRE PASSER LES TESTS

Dans ce modèle, vous répétez cycliquement les étapes suivantes :

3. Nettoyer le code tout en gardant les tests en succès. Étant donné que le test est déjà en place, on verra immédiatement si on casse le moindre comportement, garantissant que le développeur soit toujours concentré sur la fonctionnalité à tester en priorité.

4. Écrire le prochain test et recommencer.



CYCLE DE DÉVELOPPEMENT TDD

Avantages :

- Si on écrit le code après le test, ce code est plus facile à tester.
- Cela ajoute de la lisibilité au code.
- Coder les tests, ajoute des connaissances sur le code qu'on va développer pour réussir ces tests.
- Cela nous oriente vers une conception plus modulaire. Le code modulaire est flexible et plus facile à modifier.
- Le TDD facilite la tâche de développement.

A series of thin, light brown lines forming an abstract geometric pattern on the left side of the slide. The lines intersect to create various polygons and shapes, some of which are filled with a very light brown color. The pattern is dense and occupies the left third of the slide.

MERCI