Gestion de projet

PHASES DE VALIDATION

Romain SESSA

- Tests d’acceptation (Agile)
- Tests unitaires
- Tests d’intégration
- Tests de non régression

Tests d'acceptation

Contexte Agile (stories)

Précondition (état du système)
Quand (acteur et comportement)
Alors (résultat)

Tests d'acceptation

User story : En tant qu'utilisateur, je veux pouvoir payer mon panier en carte bancaire afin de finaliser ma commande.

Tests d'acceptation :

1) L'utilisateur a constitué son panier. Quand il choisit le règlement bancaire, il est redirigé sur le site de la banque.

2) …

Tests d'acceptation

Bonne pratique :

Pour chaque US, au moins 2 tests d'acceptation :

1) Cas nominal
2) Cas d'erreurs

Méthode : Acceptance Test Driven Development.

Application

Ecriture des tests d'acceptations des US rédigées pour l'exercice Fun Italy.

Tests unitaires

Validation d'un code restreint de l'application.

On définit :

- Les données en entrée
- Les données en sortie

On valide la donnée à travers une assertion.

Tests unitaires

Les tests unitaires doivent remplir 2 critères :

- Qualitatif : les assertions sont bien définies.
- Quantitatif : une quasi-totalité du code est testé (coverage)

Tests unitaires

```java
package romain.project.testing;

public class ClassToTest {

    public int multiple(int x, int y) {
        return x * y;
    }

}
```

Tests unitaires

```java
import junit.framework.Assert;
import junit.framework.TestCase;

public class MyTest extends TestCase {

    public void testMultiple() {

        int x = 2;
        int y = 2;
        int expectedResult = 4;

        ClassToTest ctt = new ClassToTest();
        int result = ctt.multiple(x, y);

        Assert.assertEquals(expectedResult, result);

    }
```

Tests d'intégration

Validation du bon fonctionnement de plusieurs composants mis bout à bout.

Nécessite d'un environnement de tests (incluant par exemple une base de données de tests).

Intégration continue

Mise en place d'outils qui permettent la vérification des différents tests lors de chaque mis à jour du code.

Exemple : Gitlab-CI / Travis / Jenkins

Tests de non régressions

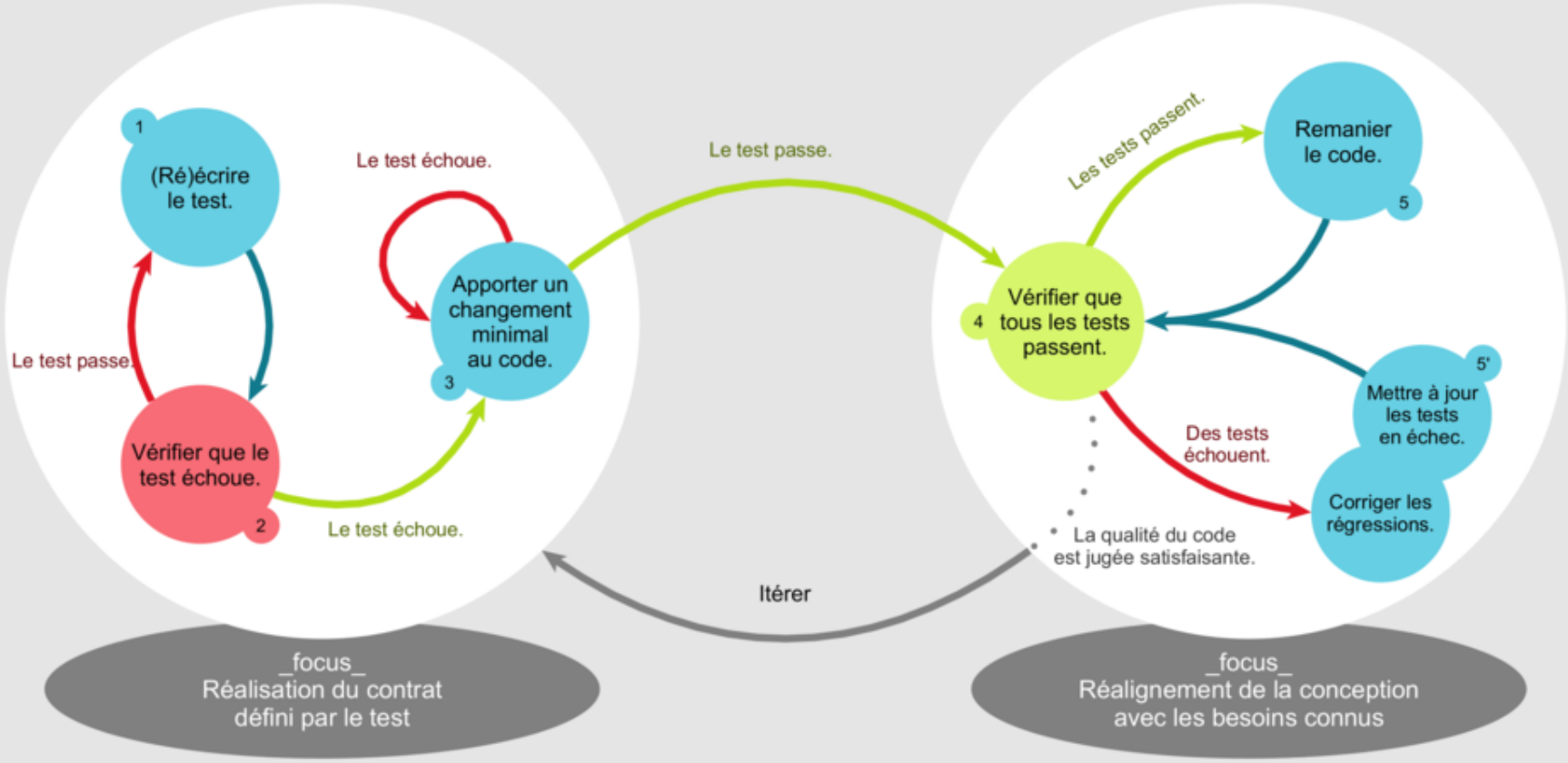
Validation qu'un nouveau code ne casse pas le fonctionnement d'une fonctionnalité implémentée au préalable.

Test Driven Développement

Piloter le développement par les tests.

Etapes du TDD

- **Ajoutez** un premier test
- **Vérifiez** que ce test échoue (car le code n'existe pas encore)
- **Écrivez** le code suffisant pour passer le test
- **Validez** immédiatement le test de ce code
- **Améliorez** le code sans modifier ni le test ni la fonctionnalité



Questions

?