

MGL7250 – Processus de développement Agile

TDD

Jacques Berger

Objectifs

Introduire la pratique du TDD

Prérequis

Programmation

Tests unitaires

Du code pour tester le code

La forme de test la plus légère, avec la portée la plus limitée

Ne s'applique qu'à une seule fonctionnalité d'une classe

Plusieurs tests unitaires peuvent être nécessaires sur une méthode

Tests unitaires

Sert de test fonctionnel à très petite échelle

Sert de test de régression

Tests unitaires

Sur une classe, on peut tester :

- La fonctionnalité

- La non-fonctionnalité

- La gestion des erreurs

- Les exceptions

- Les cas limites

- Les cas hors bornes

Tests unitaires

Les propriétés d'un test unitaire :

Exécution très rapide

Code simple

Indépendant des autres tests

Doit pouvoir s'exécuter en tout temps

Tests unitaires

Un test unitaire ne doit pas :

Manipuler un fichier

Traiter avec une base de données

Effectuer une communication sur un réseau

Dépendre d'un environnement de test

Communiquer avec le système d'exploitation

Avantages

Permet de détecter les erreurs plus tôt

Transfert de connaissances

Facilite la maintenance et le refactoring

Très payant à long terme

Inconvénients

Couplage fort entre la classe du domaine et la classe de tests

Plus de code à maintenir (code de test)

Les tests peuvent être bogués

Activité souvent difficile

TDD

Test-Driven Development

Créé par Kent Beck

Méthode Agile qui prône la création de tests unitaires et le refactoring

TDD

Idée de base : On veut une couverture de test la plus haute possible et l'on rédige toujours nos tests avant le code

TDD

Red – green – refactor

1: Écrire un test pour une nouvelle fonctionnalité

2: Écrire une implémentation qui fait échouer le test

3: Écrire une implémentation qui fait marcher le test

4: Faire du refactoring jusqu'à l'obtention d'un design satisfaisant

TDD

Avantages

- Première réflexion sur l'utilisation de la classe
- Grande banque de tests
- Meilleure maintenabilité
- Favorise le refactoring
- Design for Testability

TDD

Inconvénients

Nouvelles fonctionnalités seulement
Discipline personnelle

Plus loin...

Test-Driven Development by Example
Kent Beck
Addison-Wesley Professional, 2002