Test Driven Development sur du code legacy

Appliquer les techniques de refactoring pour faire évoluer du code legacy vers du code de qualité en minimisant les risques

DESCRIPTION

Le code legacy est une métaphore pour parler de ce code existant dans nos applications, difficile à maintenir, souvent de mauvaise qualité et non testé automatiquement.

Ce module forme les développeurs aux pratiques permettant de maintenir et de faire évoluer du code legacy sans risque, et ainsi trouver les trajectoires de retour à une haute qualité. Ce sera l'occasion d'apprendre à mettre en place des tests automatisés quelles que soient les possibilités initiales du code.

OBJECTIFS PEDAGOGIQUES

- Diagnostiquer les problèmes de qualité du code et mettre en œuvre des techniques de refactoring adaptées.
- Identifier les points d'entrée pour faire évoluer du code legacy en minimisant les risques
- Appliquer les techniques pour mettre en place des tests automatisés autour de code legacy
- Evaluer les risques pour choisir une stratégie adaptée de réduction de la dette technique

PUBLIC CIBLE

- Chef de projet en développement
- Développeur
- Testeur ayant une fibre développement
- Architecte
- Technical Leader

PRE-REQUIS

- Compétences en programmation et génie logiciel.
- Une expérience dans le développement piloté par les tests (TDD) et la programmation orientée objet est un plus.
- Avoir suivi la formation « Qualité des développements avec Test Driven Development : optimiser son développement logiciel par les tests » (TDD01).

METHODE PEDAGOGIQUE

Stage pratique
Qualité du logiciel - Software Craftsmanship

Code :
TDD02

Durée :
3 jour(s) (21,00 heures)

Exposés : **20 %**
Cas pratiques : **70 %**
Echanges d'expérience : **10 %**

Inter-entreprises :
Prochaines sessions disponibles sur notre site web.
Tarif : 2 110,00 € HT / participant

Intra-entreprise :
Tarifs et dates sur demande.

Formation principalement composée d'exercices pratiques qui fourniront aux participants des outils qu'ils pourront mettre en pratique immédiatement dans leurs projets actuels.

Echanges sur les contextes des participants et retours d'expérience du formateur, complétés d'apports théoriques.

PROFIL DES INTERVENANTS

Cette formation est dispensée par un·e ou plusieurs consultant·es d'OCTO Technology ou de son réseau de partenaires, expert·es reconnus des sujets traités.

Le processus de sélection de nos formateurs et formatrices est exigeant et repose sur une évaluation rigoureuse leurs capacités techniques, de leur expérience professionnelle et de leurs compétences pédagogiques.

MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION

L'évaluation des acquis se fait tout au long de la session au travers des ateliers et des mises en pratique.

Afin de valider les compétences acquises lors de la formation, un formulaire d'auto-positionnement est envoyé en amont et en aval de celle-ci.

En l'absence de réponse d'un ou plusieurs participants, un temps sera consacré en ouverture de session pour prendre connaissance du positionnement de chaque stagiaire sur les objectifs pédagogiques évalués.

Une évaluation à chaud est également effectuée en fin de session pour mesurer la satisfaction des stagiaires et un certificat de réalisation leur est adressé individuellement.

PROGRAMME PEDAGOGIQUE DETAILLE

Jour 1

ANTI PATTERN : LEGACY CODE

- Du code que nous avons reçu en héritage, qui a une valeur pour l'entreprise, et qu'il faut modifier
- Les quatre raisons de modifier un code legacy

- Couvrir les tests avec un harnais de tests unitaires
- Améliorer le délai de feedback du code sur le développeur
- Difficultés du TDD sur du code legacy
- Le dilemme du code legacy

STRATÉGIE DE MODIFICATION D'UN CODE LEGACY

- Identifier un point de changement
- Trouver les points de test
- Casser les dépendances
- Ecrire des tests
- Effectuer le changement et refactorer
- Types de raccords : préprocesseur, faux collaborateurs

PATTERN : TEST DE CARACTÉRISATION

Problème : le code est non testé, la documentation est absente ou obsolète

Solution: écrire des tests qui caractérisent le système tel qu'il est

Démarche :

- Appeler un morceau de code depuis un harnais de test
- Ecrire une assertion dont vous savez qu'elle échouera
- En échouant le test indique quel est le comportement du code
- Modifier le test de façon à ce qu'il attende le comportement que produit le code
- Répéter

Heuristique générale

Test de caractérisation sur un bug

- Exercice pratique : « Installer la base de code TriviaGame – examiner les classes – écrire des tests de caractérisation. »
- Débrief toutes les 25 minutes

MISE EN PRATIQUE

CLÔTURE DU JOUR 1

Jour 2

PATTERN : DIAGRAMME D'EFFET

- Problème : on souhaite étudier l'impact que pourrait avoir un changement sur le code
- Solution : tracer les effets de chaque variable sur les méthodes impactées

MISE EN PRATIQUE

CLÔTURE DU JOUR 2

Jour 3

PATTERN : EXTRACT INTERFACE

- Problème : une classe collabore avec une classe posant une dépendance extérieure
- Solution : extraire une interface de la classe posant la dépendance
- Autres patterns de rupture de dépendances extérieures

MISE EN PRATIQUE

BILAN ET CLÔTURE DE LA FORMATION

Accessibilité

L'inclusion est sujet important pour OCTO Academy.
Nos référent·es sont à votre disposition pour faciliter l'adaptation de votre formation à vos besoins spécifiques.
Pour les contacter : academy.accessibilite@octo.com