

Clean Code

Concevoir et écrire un code propre ou améliorer un code existant

DESCRIPTION

Les artisans du logiciel savent que la qualité du code va de pair avec la rapidité, la simplicité et l'efficacité du développement. Les pratiques, principes et standards qui définissent cette qualité sont rassemblées sous le vocable "Clean Code". Ils permettent à toute équipe de créer, de maintenir (ou de rétablir) un code de grande qualité, et concernent de multiples aspects de la programmation : conventions de style, règles de nommage, critères de qualité des fonctions, des classes, des relations entre les classes. A l'issue de cette formation, vous pourrez mettre en œuvre les pratiques de base permettant d'obtenir un code de grande qualité, testé et flexible.

OBJECTIFS PÉDAGOGIQUES

Identifier, expliquer et illustrer les principaux éléments qui forment le "Clean Code"

Utiliser ces standards sur un projet exemple, ainsi que sur du code existant

Définir le lien entre la qualité du code et la facilité d'évolution des applications

PUBLIC CIBLE

Chef de projet en développement

Développeur

Testeur ayant une fibre développement

Architecte

Technical Leader

PRÉ-REQUIS

- Connaissances de la programmation objet.
- Expérience de base du développement de logiciel.
- Avoir suivi la formation "Qualité des développements avec Test Driven Development" (TDD01) serait un plus.

MÉTHODE PÉDAGOGIQUE

Formation pratique, visant à l'acquisition d'un savoir-faire, basée sur des exercices ainsi que des échanges et retours d'expérience du formateur.

Les participants doivent apporter plusieurs extraits de code sur lesquels ils travaillent ou ont travaillé.

PROFILS DES INTERVENANTS

Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.

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. Une évaluation à chaud sur la satisfaction des stagiaires est réalisée systématiquement en fin de session et une attestation de formation est délivrée aux participants mentionnant les objectifs

Stage pratique en présentiel QUALITÉ DU LOGICIEL

Code :
TDD03

Durée :
3 jours (21 heures)

Exposés :
20%

Cas pratiques :
70%

Échanges d'expérience :
10%

Sessions à venir :

13 - 15 nov. 2019

Lille / 1 990 eur

9 - 11 déc. 2019

Paris / 1 990 eur

Tarif & dates intra :
Sur demande

de la formation, la nature, le programme et la durée de l'action de formation ainsi que la formalisation des acquis.

POUR ALLER PLUS LOIN :

- Parcours professionnalisant "S'appropriier les pratiques du développement logiciel de qualité"
- Formation "Revue de code : améliorer son code grâce au Code Review" (REVCO)
- Formation "Qualité des développements avec Test Driven Development" (TDD01)
- Formation "Test Driven Development sur du code legacy" (TDD02)
- Formation "Donner et recevoir des feedbacks efficaces" (FFEED)
- Ouvrage "Clean Code" - Robert Martin
- Ouvrage "Working Effectively with Legacy Code" – Feathers

Programme pédagogique détaillé par journée

Jour 1

CONNEXION

- Tour de table
- Présentation des participants
- Questionnaire/brainstorm : que valorisez-vous dans votre code ? Quels principes essayez-vous de suivre ? Quelles pratiques utilisez-vous ?

PRESENTATION SUR LA QUALITE DU CODE

- Le code pourri. Développement, tests et revue. Systèmes complexes et obsolescence. Réécriture sans tests. Les tests de développeurs. Importance du feedback. Conserver la maintenabilité
- Exercices pratiques
 - "Réagissez à la présentation en indiquant si vous êtes d'accord, pas d'accord ou si vous avez expérimenté les idées exprimées."
 - "Identifiez un code que vous jugez intéressant (cela peut-être un code de bonne qualité ou de mauvaise qualité) et expliquez pourquoi."

ENJEUX ET CRITERES DE QUALITE DU CODE PROPRE

- Pourquoi est-ce important ?
- Qualité structurelle – Le code comme environnement de travail – Le code comme design
- Propriétés du code TDD : passe tous les tests – exprime l'intention – ne contient pas de répétition – minimal
- Pratiques et outils pour faire du code propre
- La règle "boy scout rule"
- Exercice pratique : "Identifiez un code que vous trouvez particulièrement intéressant. Lisez-le à voix haute. Après lecture, les autres participants doivent expliquer ce que fait le code."

REGLES DE QUALITE

- Règles de nommage
- Qualité des fonctions – Step Down Rule
- Qualité des commentaires
- Exercice pratique : "Identifiez un code que vous trouvez particulièrement intéressant. Imprimez ce code. Avec un stylo, surlignez tous les points du code qui posent problème. Essayez de surligner le maximum de points en 2 minutes. Passez le code revu à vos voisins et revoyez le code qu'ils vous passent."

ABSTRACTION ET DESIGN ORIENTE OBJET

- Abstraction ou détails, il faut choisir
- Principes SOLID : Single Responsibility - Open/Closed - Liskov Substitution - Interface Segregation - Dependency Inversion
- Loi de Demeter
- Exercice pratique : "Identifiez l'ensemble du code implémentant un scénario utilisateur de votre application. Imaginez un changement fonctionnel dans ce scénario. Déterminez combien d'endroits distincts du code seraient à modifier en vue d'implémenter le changement."

CLOTURE DU JOUR 1

Jour 2

CLEAN CODE EN PRATIQUE SUR UN PROJET SIMPLE

- Exercices pratiques
 - "Collecte d'information : quels sont vos raccourcis clavier les plus utilisés ?"
 - "Projet exemple : clonez le code du projet. Lisez le code. Appliquer une revue de 2 minutes."
- Présentation fonctionnelle du projet exemple
- Exercice pratique : "Ecrivez un cas de tests sur le projet exemple afin de pouvoir refactorer le code."

REGLES DE NOMMAGE

- Rappel des règles de nommage
- Exercice pratique : "Appliquez sur le code du projet les règles de nommages vues précédemment. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."

REGLES DE QUALITE DES FONCTIONS

- Rappel des règles de qualité des fonctions
- Exercices pratiques
 - "Appliquez sur le code du projet la pratique Don't Repeat Yourself. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."
 - "Appliquez sur le code du projet la pratique Step Down Rule. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."

REGLES DE DESIGN ORIENTE OBJET

- Rappel des règles de qualité des fonctions
- Exercice pratique : "Appliquez sur le code du projet le principe Single Responsibility Principle. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."

Jour 3

CLEAN CODE EN PRATIQUE SUR VOTRE PROJET

- Exercice pratique : "Sélectionnez une partie du code de votre projet qui pose des problèmes de qualité, pour lequel vous disposez de quelques tests."

REGLES DE NOMMAGE

- Rappel des règles de nommage
- Exercice pratique : "Appliquez sur le code de votre projet les règles de nommages vues précédemment. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."

REGLES DE QUALITE DES FONCTIONS

- Rappel des règles de qualité des fonctions
- Exercices pratiques
 - "Appliquez sur le code de votre projet la pratique Don't Repeat Yourself. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."
 - "Appliquez sur le code de votre projet la pratique Step Down Rule. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."

REGLES DE DESIGN ORIENTE OBJET

- Rappel des règles de qualité des fonctions
- Exercice pratique : "Appliquez sur le code de votre projet le principe Single Responsibility Principle. Utilisez uniquement les commandes du menu refactor et les raccourcis clavier."

BILAN ET CLÔTURE DE LA SESSION

- Tour de table : ce que j'ai appris – ce que ça change – ce que je vais faire différemment.