

- *Connaître les meilleures pratiques pour écrire un code de qualité favorisant la « lecture » et la maintenance d'une application*
- *Comprendre comment organiser ses tests pour produire une application de qualité*
- *Connaître les outils nécessaires à la fabrication logicielle pour produire des livrables de qualité*
- *Savoir utiliser Git pour gérer les codes sources*
- *Disposer d'une première expérience de la gestion des bugs avec Jira*

3**Prix : 1?930 € € / HT****OUTILS PÉDAGOGIQUES****MODALITÉS D'ÉVALUATION****MODALITÉS DE FINANCEMENT****MODALITÉS ET DÉLAIS D'ACCÈS****OBJECTIFS PÉDAGOGIQUES****ACCESSIBILITÉ****LES POINTS FORTS DE LA FORMATION****PRÉ-REQUIS**

- Disposer d'une première expérience de développement
- Maîtriser un langage de programmation (C#, Java ou C++)

MODALITÉS ET DÉLAIS D'ACCÈS**ATTESTATION OBTENUE****EFFECTIF DE LA FORMATION****CERTIFICATION****MODALITÉ PÉDAGOGIQUE**

Cours dispensé en mode présentiel avec une alternance d'apports théoriques et méthodologiques, et de mises en situations pratiques

PROCHAINES SESSIONS

Le 17/01/2022

Le 04/04/2022

Le 17/01/2022

Le 04/04/2022

Le 17/01/2022

Le 04/04/2022

PROGRAMMES DE ANGULAR 2+ – DÉVELOPPEMENT D'APPLICATIONS WEB**GÉNIE LOGICIEL ET QUALITÉ**

- Structuration d'un bon code source
- Les libraires : gestion des dépendances/couplages faibles
- Technique de mutualisation de code

- Origine des défauts logiciels
- Intérêt, évaluation
- Sensibilisation au coût d'un programme non testé
- Indicateur

TESTER PENDANT LE CYCLE DE VIE LOGICIEL

- Les tests dans le cadre du mode Agile VS cycle en V
- Niveaux de tests : composants, intégration, système
- Cible des tests : fonctionnels / non fonctionnels, architecture logicielle, non régression

LES TECHNIQUES DE TEST

- La revue de code
- Techniques « boîtes noires »
- Techniques « boîtes blanches »
- Choisir sa technique de test

AUTOMATISATION DES TESTS ET DE LA PRODUCTION DE LIVRABLE

- Outillage : Comment bien choisir sa toolchain de test : Construire sa fabrique logicielle
- Étude des différences entre Maven, NPM et Composer
- Automatisation de tests GUI/IHM
- Exécution et génération de rapport de test
- Les tests en mode Agile
- Granularité de tests : composants, intégration, système
- Les différents types : fonctionnels / non fonctionnels, architectural, non régression

GIT : TRAVAILLER EN ÉQUIPE AU JOUR LE JOUR

- La décentralisation
- Ajout, modification, suppression de fichiers et répertoires
- Gestion des commits
- Synchronisation avec un référentiel distant
- Comparaison
- Utilisation des tags
- Créer et appliquer des patches

GIT : GESTION DES BRANCHES

- Création de branches
- Navigation entre branches
- Fusion de branches
- Résolution des conflits
- Branche temporaire

JIRA : GESTION DES BUGS

- Le bug tracking
- Jira dans la communication avec ses utilisateurs
- Suivi de l'activité