

DEVELOPPEMENT AGILE PILOTE PAR LES TESTS



Nombre de jours -> 3

Participants -> Les testeurs, développeurs, utilisateurs d'Agile, analystes de business, gestionnaires de projet des logiciels ou autres personnes désireuses de se perfectionner en test Agile.

Pré-requis -> Des connaissances des principes de la méthodologie agile sont utiles. Vous devez notamment connaître les ressorts du cycle des itérations et de l'esprit de collaboration qui sous-tendent les méthodologies agiles. Vous pouvez participer à cette formation même si vous n'avez jamais réalisé de tests logiciels.

Objectifs -> Les organisations ayant opté pour la méthodologie agile sont nombreuses à devoir gérer la difficulté de trouver le juste équilibre entre des délais de production accélérés et la livraison de produits de qualité. Cet état de fait a profondément modifié le rôle de l'assurance qualité et posé de nouvelles problématiques à un grand nombre d'organisations agiles en matière de mise en œuvre d'activités de test efficaces et à moindre coût. Au cours de cette formation, vous allez apprendre à utiliser un framework de test agile professionnel conçu pour vous aider à améliorer la qualité de vos produits tant en profitant pleinement des nombreux avantages de la méthode agile.

Moyens pédagogiques, techniques et d'encadrement

- > ● 1 poste de travail complet par personne
- De nombreux exercices d'application
- Mise en place d'ateliers pratiques
- Remise d'un support de cours
- Passage de certification(s) dans le cadre du [CPF](#)
- Mise en place de la Charte contrôle et qualité OPCA
- [Notre plateforme d'évaluation](#) :
 - Evaluation des besoins et objectifs en pré et post formation
 - Evaluation technique des connaissances en pré et post formation
 - Evaluation générale du stage

1 - Théorie générale des tests

- Évaluer les principes fondamentaux des tests
- Différencier les pratiques agiles des pratiques traditionnelles
- Présenter la théorie des tests Lean/Agile

2 - Effectuer la transition vers l'approche agile pour améliorer la livraison et la qualité du logiciel

- Appliquer les valeurs et les principes agiles aux tests
- Inspecter les quadrants agiles
- Tirer parti du développement piloté par les tests (TDD, Test Driven Development)
- Automatiser les tests pour une meilleure agilité

3 - Acquérir une base de tests

- Tester la charte et les principales fonctionnalités
- Se concentrer sur la valeur pour le client et les utilisateurs type (personas)
- Écrire des cas de test utiles à partir des user stories
- Développer des critères d'acceptation pour les stories

4 - Automatiser via le développement piloté par les tests d'acceptation (A-TDD, Acceptance Test Driven Development)

- Concevoir l'anatomie d'un test agile
- Créer un catalogue d'idées de tests
- Raffiner une définition d'un logiciel terminé
- Anticiper les critères de validation avec le développement piloté par le comportement (BDD, Behavior Driven Development)

5 - Préparer les tests d'acceptation utilisateur

- Spécifier à l'aide d'exemples et de scénarios
- Mettre en place des tests d'utilisabilité et exploratoires
- Utiliser le story-mapping pour améliorer la couverture
- Gérer les processus de test

6 - Créer les tests unitaires

- Définir les unités candidates
- Définir les données de test
- Tester d'abord et écrire les assertions d'abord
- Mettre en place et démonter un test
- Exploiter des outils pour mieux gérer et analyser la couverture

7 - Maîtriser avec le TDD

- Obtenir le feu vert
- Identifier les bons tests
- Travailler avec de grands systèmes

8 - Employer des techniques de simulation

- Programmer des objets factices et espions
- Simuler l'accès aux données
- Identifier des interfaces et des frameworks de simulation
- Tester des systèmes existants

9 - Valider pour FURPS+

- Trouver les schémas de test non fonctionnels
- Simuler les interfaces et conduire des tests de performance grâce à l'automatisation
- Évaluer les systèmes existants

10 - Appliquer des tests de non-régression

- Adopter un catalogue de patterns de refactorisation
- Identifier des raisons de refactoriser

11 - Valider les modifications en retestant

- Synchroniser les tests et les modifications du code
- Ajouter des fonctionnalités et retester
- Gérer et retester les bogues
- Tendre vers la tolérance zéro

12 - Livrer le logiciel

- Mapper le pipeline de déploiement d'Agile
- Incorporer les principes clé de la gestion de configuration pour les livraisons fréquentes
- Adapter le contrôle des versions et les stratégies de branchement aux exigences d'Agile

13 - Gérer les pipelines de déploiement

- Créer un processus de construction et un script de déploiement
- Contrôler les modifications du code et protéger l'intégrité du produit
- Intégrer l'automatisation des tests au processus d'intégration continue

14 - Créer un écosystème d'intégration continue

- Planifier l'infrastructure et les environnements
- Gérer les composants et les dépendances
- Mettre en place l'intégration continue