



DEVELOPPEMENT DE LOGICIELS AGILE ET MODELISATION

- Public** Aux assistants à maîtrise d'ouvrage, testeurs Assurance Qualité, programmeurs, chefs de projets logiciels, ainsi qu'à toute personne souhaitant découvrir les principes fondamentaux du développement de logiciels moderne. Les analystes, concepteurs, programmeurs et testeurs pourront acquérir une expérience pratique de réalisation des tâches incombant à chaque rôle au sein d'une équipe de développement de logiciels. Les responsables seront eux en mesure de mieux comprendre les différentes activités qui composent le cycle de vie du développement d'un logiciel, ainsi que les compétences dont doivent faire preuve les équipes agiles pour assurer la réussite de leurs projets.
- Durée** 4 jours - 28 heures
- Pré-requis** Cette formation offre un aperçu global de toutes les activités nécessaires pour mener à bien un projet de développement de logiciels. Elle s'adresse aux personnes qui découvrent les technologies de l'information ou qui exercent un rôle au sein d'une équipe de développement de logiciels. Une expérience préalable de développement de logiciel est utile sans toutefois être obligatoire.
- Objectifs** Au cours de cette formation, vous apprendrez à analyser, développer, mettre en œuvre et tester des applications qui répondent aux exigences des utilisateurs au sein d'une simulation d'étude de cas. Vous allez acquérir une expérience pratique des différents rôles qui composent une équipe de développement de logiciel.

1. DÉFINIR LES BONNES PRATIQUES DE PROGRAMMATION

- Identifier les rôles et les activités du développement logiciel
- Choisir entre méthodologies pilotées par un plan-et méthodologies agiles
- Modèle en cascade
- Modèle en V
- Modèle en spirale

2. EFFECTUER LA TRANSITION VERS LES PROCESSUS AGILES

- Comparer XP et Scrum
- Identifier les valeurs, les processus et les bonnes pratiques agiles et déterminer la longueur idéale d'une itération

3. RECUEILLIR LES EXIGENCES

- Recueillir les exigences auprès des utilisateurs
- Rédiger les user stories
- Prioriser et estimer le travail

4. OPTIMISER LES ÉQUIPES AGILES

- Constituer des équipes agiles interfonctionnelles
- Identifier les bonnes pratiques pour la productivité de l'équipe
- Augmenter la communication avec des réunions quotidiennes (« stand-up » meetings), des tableaux de tâches et une réflexion régulière
- Maximiser la productivité de l'équipe

5. GÉRER DES PROJETS AGILES

- Organiser les projets par fonctionnalités
- Mesurer le travail achevé avec des backlogs
- Suivre l'avancement avec des burn-down charts
- Calculer la vélocité et s'adapter au changement
- Prévoir les coûts et les dates d'achèvement des projets

6. MODÉLISER LES EXIGENCES AVEC UML

- Identifier les acteurs et les cas d'utilisation, décrire les exigences avec des diagrammes de cas d'utilisation
- Représenter les interactions entre les utilisateurs et le système avec des diagrammes d'activités
- Employer des diagrammes de séquence et des diagrammes de communication, décrire les transitions du système avec des diagrammes d'états-transitions

7. DÉTAILLER LES CAS D'UTILISATION

- Identifier les préconditions et les post-conditions
- Spécifier un chemin nominal et des chemins alternatifs
- Documenter les exigences non fonctionnelles et système

8. MODÉLISER LES APPLICATIONS AVEC DES DIAGRAMMES DE CLASSE UML

- Représenter les données et le comportement des classes
- Reconnaître la nécessité de l'héritage
- Le principe de responsabilité unique / « ouvert-fermé »

9. ARCHITECTURER LES SYSTÈMES ORIENTÉS OBJET

- Concevoir un logiciel pour en faciliter la maintenance
- Simplifier le code de l'interface utilisateur avec le pattern MVC, définir les exigences des données
- Spécifier les relations et les multiplicités

10. DÉFINIR LES NIVEAUX DE TEST

- Test d'acceptation, test système, test d'intégration et test unitaire, employer des tests automatisés et manuels
- Spécifier les conditions limites

11. VÉRIFIER QUE LE LOGICIEL RÉPOND AUX EXIGENCES AVEC DES TESTS D'ACCEPTATION UTILISATEUR

- Écrire des scripts pour les tests d'acceptation
- Exécuter des tests exploratoires, spécifier les critères d'acceptation, déterminer les conditions de satisfaction

12. RÉDUIRE LES BOGUES AVEC DES TESTS UNITAIRES AUTOMATISÉS

- Utiliser des outils d'automatisation des tests unitaires
- Écrire le code des tests unitaires, isoler les classes pour tester
- Programmer des objets factices

13. PRODUIRE DES MAQUETTES ET DES PROTOTYPES

- Mettre au point des interfaces utilisateur avec des maquettes
- Expérimenter avec des prototypes fonctionnels

14. CONDUIRE DES REVUES SPRINT ET DES RÉTROSPECTIVES

- Appliquer des techniques pour recueillir du feedback
- Analyser les possibilités d'amélioration
- S'adapter au changement
- Réfléchir aux bonnes pratiques agiles

15. CONDITIONS PÉDAGOGIQUES

- 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 OPCO
- Modalité d'évaluation des acquis :
 - Evaluation des besoins et objectifs en pré et post formation
 - Evaluation technique des connaissances en pré et post formation
 - Evaluation générale du stage
- Remise d'une attestation de stage

NOUS CONTACTER

Centre de formation

87, RUE GÉNÉRAL MANGIN
38000 GRENOBLE

Téléphone

04 76 23 20 50 - 06 81 73 19 35

Siège social

16, ALLÉE FRANÇOIS VILLON
38130 ÉCHIROLLES

E-mail

contact@audit-conseil-formation.com

Suivez-nous sur les réseaux sociaux, rejoignez la communauté !



ACF Audit Conseil Formation



@ACF_Formation



ACFauditconseilformation

Accessibilité handicapés

Au centre d'affaires ELITE partenaire d'ACF à 20 m.
Guide d'accessibilité à l'accueil.