Formation Agile : Automatisation des tests

Durée :	2 jours
Public :	Développeurs, chefs de projets
Pré-requis :	Avoir des connaissance de la méthodologie agile du niveau de la formation 918, Scrum : Gestion de projet agile, ou de la formation 1815, Développement agile piloté par les tests. Vous devez avoir également au moins un an d'expérience en développement de logiciels
Objectifs :	- Optimiser la qualité de vos projets agiles grâce à l'automatisation des tests - Développer des tests pour les histoires utilisateur et les fonctionnalités avec le développement piloté par les tests d'acceptation (ATDD) et par le comportement (BDD) afin de valider la valeur apportée par le produit - Utiliser des outils de tests fonctionnels et non fonctionnels - Appliquer des design patterns pour élargir la couverture des tests - Structurer le code et les données de test pour garantir la réussite de vos projets sur le long terme
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	TES100304-F
Note de satisfaction des participants:	5,00 / 5

Motivations

Transformer les besoins en matière d’automatisation des tests en processus agiles
Justifier les résultats performants de l’automatisation

Dimensions de l’automatisation

Exploiter tous les aspects de la Pyramide des tests pour améliorer la qualité
Classer les processus d’automatisation dans les Quadrants des tests agiles
Utiliser le développement 4D des stratégies TDD

Bonnes pratiques

Définir des règles d’engagement dans un manifeste
Choisir des tests de retour sur investissement élevé via un indice d’automatisation

Stratégies de test

Modéliser les processus de test en API ou en interfaces utilisateur
Classer les tests métier dans les quatre catégories
Utiliser les design patterns pour la réussite de vos projets sur le long terme
Appliquer des cas de test avec les principaux patterns d'exécution

ATDD / BDD

Formaliser les cas de test avec les cartes ATDD
Créer des tests avec des exemples représentatifs
Appliquer les règles métier avec le BDD

Automatisation des tests de l'interface utilisateur

Choisir les patterns CUIT
Organiser les tests de l'interface par couches pour améliorer la résilience

Analyse de la conception

Examiner les composants lors des revues de conception
Appliquer les règles de conception orientée objet pour optimiser les stratégies de tests unitaires
Mesurer la couverture des tests pour évaluer la réussite du projet

TDD

Appliquer les patterns TDD pour éviter les changements de code
Utiliser des maquettes pour améliorer la couverture
Identifier les principaux design patterns simplifiant la maintenance des tests
Éviter les anti-patterns pour limiter la dette technique

Planification pilotée par les besoins non fonctionnels (NFR)

Utiliser les dimensions NFR pour améliorer la planification
Répartition dans des sous-dimensions pour évaluer avec précision la qualité du projet

Outils de tests non fonctionnels

Apprendre à choisir les outils permettant de concevoir des tests à moindre coûts
Développer des tests multidimensionnels avec les outils adéquats

Principes de l'intégration continue

Intégrer les tests au processus du pipeline
Accélérer le transfert du code en recueillant un feedback rapide après les tests

Pipelines de déploiement

Configurer des tests adaptés à la structure du pipeline de déploiement
Automatiser l'analyse du code lors de la phase de validation
Automatiser les tests lors de la phase de déploiement

Environnements de test

Utiliser la virtualisation pour faciliter la gestion des tests
Créer des environnements configurés par des lignes de code