

Tests automatiques et intégration continue en C++

Cours Pratique de 4 jours - 28h

Réf : AOI - Prix 2024 : 2 750€ HT

Ce cours présente et met en œuvre un processus de test d'applications en environnement C++. Vous verrez comment chaque étape de la démarche de tests peut être automatisée. Vous apprendrez à vérifier l'efficacité de vos tests unitaires et à écrire des tests systèmes. Vous mettrez en place l'intégration continue.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Mettre en place le contenu et le suivi du projet d'automatisation

Définir la couverture du plan de tests et organiser le référentiel de tests

Mettre en œuvre des outils spécifiques pour automatiser les tests unitaires en C++

Automatiser des tests d'IHM, de Web Service et de montée en charge

Mettre en place l'intégration continue et analyser le reporting des résultats

MÉTHODES PÉDAGOGIQUES

Outils propres à C++ : Make, Gcov, GoogleTest (GTest) ou CppUnit, AutoIT (windows), QTest.

TRAVAUX PRATIQUES

Outils généraux : Testlink, Selenium, Mantis, Hudson ou Jenkins, SVN, SOAP UI.

LE PROGRAMME

dernière mise à jour : 01/2018

1) Rappels sur la démarche de tests

- Définition. Les niveaux de test : unitaires, intégration, systèmes et recette, confirmation et non-régression.
- Les différentes techniques de tests. Le processus de test.
- Les cycles de développement et de test.
- Stratégie de test. Intérêt des stratégies de bout en bout.
- Décision d'automatisation. Quel suivi ?

Echanges : Définition des tâches pour mener à bien un projet d'automatisation d'une application C++.

2) Analyse et conception

- Présentation de techniques de test.
- Exigences, test, condition de test, cas de test, procédure de test, prérequis.
- Qu'est-ce qu'une couverture ?
- Présentation d'un outil Open source testlink.
- Bonnes pratiques : couverture, valorisation, écriture des tests.

Travaux pratiques : Mise en œuvre du référentiel de tests. Panorama d'outils Open source et payants pour la gestion des tests, présentation d'autres outils (TFS).

3) Automatisation des tests unitaires

- Les tests unitaires : qu'est-ce qu'un pilote ?
- Utilisation des bouchons pour automatiser ses tests.

PARTICIPANTS

Responsables qualité et tests logiciels, développeurs, ingénieurs automatisation des tests.

PRÉREQUIS

Des connaissances de base en programmation C++ sont nécessaires.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Les types de couverture : instruction, branche, condition, décision.

Travaux pratiques : Mis en œuvre du framework de test unitaire de Google en C++ (GTest) et d'un bouchon de test. Vérification de couverture de code avec Gcov.

4) Automatisation des tests systèmes

- Organiser l'exécution des tests : environnements de test, gestionnaire d'anomalie.

- Automatiser les tests systèmes.

- Les tests d'API, Webservice. Les tests d'IHM. Les tests de charges.

- Bonnes pratiques en matière d'automatisation.

- Présentation de Sélénium pour le Web.

- Solutions pour les autres types d'IHM.

Travaux pratiques : Automatisation de test d'IHM/Web avec AutoIT/Selenium/QTTest.

Automatisation des tests de Web Service avec SOAP UI. Mise en place de tests de charge avec SOAP UI.

5) Intégration continue

- Intégration continue : principe et avantages.

- Présentation de Hudson ou Jenkins, présentation de plug-ins.

- Reporting des résultats.

Travaux pratiques : Mise en place avec Hudson SVN et Make. Reporting des résultats de couverture et test, intégration avec Testlink.

6) Conclusion

- Comment mettre en place une automatisation.

- "Keyword testing" : démonstration avec robot framework.

- Choix des outils. Outils Open source ou payants ?

LES DATES

CLASSE À DISTANCE

2024 : 02 juil., 24 sept., 10 déc.

PARIS

2024 : 25 juin, 17 sept., 03 déc.