

Automatisation des tests logiciels

Cours Pratique de 4 jours - 28h

Réf : OTT - Prix 2024 : 2 750€ HT

L'automatisation des tests d'un développement logiciel, est un investissement lourd mais rentable. Découvrez les aspects organisationnels/financiers, les bonnes pratiques et les outils de l'automatisation des différents types de test : tests unitaires, tests d'intégration, tests fonctionnels, tests de performance.

OBJECTIFS PÉDAGOGIQUES

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

Utiliser un référentiel de tests et paramétrer des liens vers des outils connexes, pour automatiser une campagne de test

Faire une analyse statique de code et de couverture de tests et l'intégrer dans une chaîne de fabrication logicielle

Utiliser les techniques d'objets factices, souches ou simulacres et mettre en place un serveur d'intégration continue

Mettre en œuvre une automatisation de tests fonctionnels sur client lourd et sur Web

Automatiser les tests de performance, avec mise en place d'un test de détermination du seuil de déni de service

TRAVAUX PRATIQUES

Alternance de séquences théoriques et de travaux pratiques.

LE PROGRAMME

dernière mise à jour : 04/2022

1) Introduction, rappels sur le processus du test logiciel

- Rôle du test dans le processus de développement.
- Les tests : unitaires, fonctionnels, etc.
- Les différentes méthodes de test.
- Processus de test et stratégie de test.
- Outils et méthodes intervenant à différentes étapes.

Travaux pratiques : Installation de quelques outils.

2) Automatisation de la gestion des tests

- Gestion de la couverture des exigences par les tests. Notion de couverture et de granularité.
- Démarche de mise au point : organisation des suites de tests et création des cas.
- Faut-il automatiser un test ? Critères à prendre en compte ?
- Préparation à l'automatisation.
- Construction de la population de test.
- Mise au point et vérification des tests (Revue)
- Exécution, enregistrement des anomalies. Notion de rapport d'incident d'après l'IEEE.
- Gestionnaires d'anomalies. Automatisation de la création des anomalies.
- Analyse de résultats d'exécution de tests. Consolidation des tests.

Travaux pratiques : Utilisation d'un référentiel de tests. Paramétrages de ponts vers des outils connexes (TestLink/Squash/Quality Center ou autres...).

PARTICIPANTS

Développeurs, responsables qualité et tests logiciels, ingénieurs automatisation des tests, chefs de projet MOE.

PRÉREQUIS

Bonnes connaissances en programmation Java.

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.

3) Automatisation des tests unitaires

- Organisation et bonnes pratiques pour les tests unitaires.
- Critères d'automatisation.
- Tests unitaires : Tests Driven Development.
- Mesure de la couverture de code : couverture des tests structurels, couverture d'instructions et branches.
- Analyse statique de code : analyse outillée du code source hors exécution (règles de codage) : Checkstyle, Cobertura.
- Automatisation avec un fichier de configuration.
- Analyse dynamique de code : couverture des instructions, des branches, des prédicats...
- Automatisation avec un outil d'analyse de couverture.
- Organisation des tests unitaires, pair programming, pair testing.
- Utilisation des frameworks : gestion des scripts de tests, gestion des données de tests, récupération des résultats.

Travaux pratiques : Mise en œuvre de tests unitaires avec Junit/TestNG. Intégration dans une chaîne de fabrication de logiciel (Ant/Maven...).

4) Automatisation des tests d'intégration

- Stratégies propres à l'intégration : big-bang, "au fil de l'eau", par incréments etc.
- Intégration ascendante versus descendante. Intégrations mixtes.
- Objets simulacres : bouchons pour simuler les fonctions appelées, mocking pour remplacer un objet. Les frameworks.
- Intégration continue : mise en place de gestionnaire de configuration, constructeur de build, outils de pilotage.
- Focus sur un gestionnaire de configuration logiciel.
- Signalement automatique des anomalies.
- Exécution automatique et cyclique des tests logiciels.
- Focus sur un constructeur de build.
- Focus sur un serveur d'intégration continue.

Travaux pratiques : Utilisation d'objets simulacres.

Mise en place d'un serveur d'intégration continue (Jenkins, Hudson ou Continuum).

5) Automatisation du test fonctionnel

- Définition du test fonctionnel, non-régression.
- Le test simulant l'action des utilisateurs à partir des interfaces utilisateurs (IHM).
- Constats sur l'automatisation du test fonctionnel.
- Automatisation des tests via l'IHM, via des interfaces de programmation (API).
- Chaîne d'outils, robots de test, script (API publiques).
- Gestion de l'obsolescence des tests.

Travaux pratiques : Automatisation de tests fonctionnels sur client lourd et sur Web.

Utilisation de robots, d'une couche logicielle publique (API). Mise en œuvre d'un outil d'automatisation (Katalon Studio avec Selenium/HP QuickTest Pro).

6) Automatisation des tests système

- Les différents type de tests système : performance, charge, stress.
- Constats sur les tests de performance.
- Tests de performance, définition des indicateurs.
- Scripts de tests de performance. Mise en œuvre d'outils et analyse.
- Tests de montée en charge, tests de stress : présentation des indicateurs clés et de l'outillage.
- Gérer la base de données de tests. Gestion du contexte.

Travaux pratiques : Mise en place d'un test de charge.

7) Synthèse

- Automatisation des tests : coût directs, indirects.
- Gains quantitatifs de l'automatisation : volumétrie, modularité...

- Gains qualitatifs de l'automatisation : mode opératoire sauvegardé, pas d'erreur, confiance...
- Mesure du retour sur investissement.

LES DATES

CLASSE À DISTANCE
2024 : 11 juin, 17 sept., 19 nov.

PARIS
2024 : 10 sept., 12 nov.

LYON
2024 : 17 sept., 19 nov.

AIX-EN-PROVENCE
2024 : 17 sept., 19 nov.

BORDEAUX
2024 : 17 sept., 19 nov.

LILLE
2024 : 17 sept., 19 nov.

NANTES
2024 : 17 sept., 19 nov.

SOPHIA-ANTIPOLIS
2024 : 17 sept., 19 nov.

STRASBOURG
2024 : 17 sept., 19 nov.

TOULOUSE
2024 : 17 sept., 19 nov.