

Working On Legacy Code

2 jours

Méthodes pédagogiques :
30% théorique, 70% pratique

Prochaines sessions : retrouvez les dates et le calendrier complet sur notre site www.arolla.fr/training

Offre inter-entreprise : Petit-déjeuner - Déjeuner - Café et collation

Objectifs :

Découvrir les meilleures techniques permettant de :

- ↳ Reprendre le contrôle d'une application devenue au fil du temps sauvage et incompréhensible
- ↳ La réfectionner dans les règles de l'art et à iso fonctionnalités
- ↳ Transformer un étalon engourdi et indiscipliné en un pur-sang tout aussi malléable que fougueux, qui vous obéira au doigt et à l'œil.

Jeune Padawan ! Nous passerons en revue tout un ensemble d'outils et de techniques dignes d'un Grand Maître Jedi du Code Legacy. Ensemble, nous nous entraînerons dur, à travers toute une panoplie d'exercices

pratiques qui vous permettront d'intégrer les bons réflexes, de reprendre à la fois confiance en vous et en votre code. L'objectif de cette formation est de vous armer d'un bagage méthodologique qui vous permettra de devenir autonome face à du code Legacy.

Néanmoins ne vous faites pas d'illusion : vous confronter à du code Legacy restera toujours un exercice très difficile ! Cependant, cette tâche ne vous semblera plus jamais insurmontable.

A travers cette formation, nous nous fixerons notamment comme objectifs :

- Rendre testable une application a priori réputée non testable en l'état
- Contrôler des dépendances sur lesquelles

on n'a pas forcément la main à l'aide de bouchons

- Savoir par où commencer et avec quelles stratégies
- De l'importance de connaître parfaitement son IDE
- De la nécessité d'automatiser au maximum

A l'issue de cette formation vous maîtriserez les points suivants:

- Définir ce qu'est du code Legacy et l'identifier
- Faire la différence entre un mock et un stub
- Établir la stratégie la plus adaptée pour rendre du code Legacy testable
- Maîtriser son IDE pour éviter des erreurs manuelles

Programme :

Jour 1

- Définir et identifier ce qu'est du code Legacy
- Reconnaître les « code smells »
- Entraînement aux techniques de refactoring en s'appuyant sur un IDE
- Casser les dépendances afin de pouvoir placer un harnais de tests à l'aide des bouchons

Jour 2

La journée entière sera consacrée à des mises en situation, à travers le traitement de plusieurs petits projets legacy qui permettront de mettre en œuvre ce qui a été abordé en première journée.

A savoir :

- ↳ La formation est donnée en français.
- ↳ Les supports écrits sont pour la plupart en anglais.
- ↳ Les exemples de code éventuels sont en Java.

Public :

Cette formation s'adresse à un public de développeurs.

Pré-requis :

- ↳ Avoir pratiqué le TDD (Test-Driven Development) et être familier de la notion de Clean Code est un plus.
- ↳ Savoir utiliser Git/SVN.
- ↳ Bonne pratique de la conception objet.
- ↳ Pour les développeurs Java, savoir utiliser Maven.
- ↳ Pour les développeurs C#, la connaissance de Resharper et NCrunch est recommandée.
- ↳ Les participants doivent avoir un poste de travail avec un environnement de développement (si besoin, nous consulter).
- ↳ Git ou SVN.
- ↳ Pour les développeurs Java, un IDE similaire à Eclipse avec le plugin infinitest ou JUnit Loop et EclEmma.
- ↳ Pour les développeurs C#, Visual Studio avec Resharper et NCrunch.

Matériel pédagogique :

Le formateur distribuera les supports de la formation au format électronique PDF à la fin de la formation.