

## Git

### Gérer le versionning avec Git

Un à un les grands projets open-source adoptent Git comme gestionnaire de versions en remplacement de CVS ou de Subversion. Cette migration n'est pas liée à un effet de mode. Git, de part sa conception radicalement différente, ouvre de nouvelles perspectives aux équipes de développement. Les deux principaux apports sont les performances et la souplesse d'utilisation.

Git est un gestionnaire de versions distribué : chaque développeur possède en local une copie de l'historique des sources d'où le gain de performances et la possibilité d'offrir des fonctionnalités impossibles à implémenter avec des gestionnaires centralisés comme CVS ou Subversion.

Le premier frein à l'adoption de Git en entreprise est le manque d'intégration avec les environnements de développement, ce problème est en passe d'être résolu avec l'apparition de nombreux outils graphiques conviviaux.

Le second frein est la nécessité pour les développeurs de comprendre la philosophie de Git et notamment d'oublier certains réflexes liés à l'utilisation de CVS ou de Subversion. C'est dans cette optique que nous avons conçu cette formation Git.

#### Détails

- Code : UL-GIT
- Durée : 2 jours ( 14 heures )

#### Public

- Chefs de projets
- Architectes
- Développeurs
- Ingénieurs
- Consultants

#### Pré-requis

- Conseillé : connaissance préalable d'un logiciel de gestion de versions

#### Objectifs

- Comprendre les principes d'un gestionnaire de versions distribués
- Découvrir par la pratique la philosophie de Git et ses apports

#### Programme

##### Présentation de Git

- La notion de gestionnaire de versions distribué
- Les principes techniques de Git
- Aperçu des workflows possibles

##### Prise en main

- Installation et configuration de git
- Création d'un premier référentiel
- Utilisation de la ligne de commande pour les opérations de base

##### Comprendre les principes de Git

- Référentiels
- Clonage de référentiels
- Index
- Répertoire de travail

##### Travailler en équipe au jour le jour

- Connexion à un référentiel

- Ajout, modification, suppression de fichiers et répertoires
- Gestion des commits
- Synchronisation avec un référentiel distant
- Comparaison
- Utilisation des tags
- Créer et appliquer des patches

##### Gestion des branches

- Création de branches
- Navigation entre branches
- Fusion de branches
- Résolution des conflits
- Branche temporaire

##### Compléments

- Interagir avec des référentiels partagés via GitHub
- Recherche par dichotomie

##### Méthodologie et organisation

- Présentation des différents patterns

#### Modalité

- Stage pratique en présentiel
- Stage pratique en distanciel

- Nombre de stagiaires minimum : 4
- Nombre de stagiaires maximum : 10

### Méthodes pédagogiques

- Exposés
- Cas pratiques

### Profils des intervenants

- Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.

### Modalités d'évaluation

- Evaluation des acquis de la formation par le biais de cas pratiques et/ou mises en situation.
- Attestation de formation remise à chaque participant.

### Démarche qualité

- Questionnaire d'évaluation de satisfaction à chaud complété par chaque participant à l'issue de la formation.

### Moyens pédagogiques

- Salle équipée de PC (1 poste par stagiaire), vidéo-projecteur.
- Espace de pause.

Dernière mise à jour le 12/11/2020