

ADOPTER LE VERSIONING GIT

2 jours en présentiel (14 heures)

Objectifs pédagogiques

Cette formation vous permet d'identifier les fonctionnalités de GIT, de gérer efficacement les dépôts locaux et distants de GIT et de déboguer et gérer les conflits dans GIT.

Population visée

Développeurs, architectes, chefs de projet: toute personne désirant améliorer sa gestion de version, et accroître sa productivité avec GIT.

Pré-requis

Notions de réseaux, utilisation de la ligne de commande et Shell script.

Méthodes pédagogiques

8 participants maximum, un poste par stagiaire et un support de cours est remis à la fin du stage (livre). La formation est constituée d'explications théoriques, de démonstrations suivies d'exercices de mise en pratique.

Formateur

Formateur consultant expert programmeur et versioning.

Modalités de validation des acquis

Évaluation en ligne des acquis via un questionnaire. Attestation de fin de stage.

Contenu

Le versioning GIT

- Concepts élémentaires du versioning,
- La gestion centralisée ou distribuée,
- Les différentes solutions de gestion de versions,
- Principe de fonctionnement de la décentralisation.

Configuration initiale

- Installation sur Windows et UNIX et présentation de l'émulation Windows MSysgit,
- Configuration du fichier .gitconfig,
- La console.

Les fondamentaux de l'utilisation de GIT

- Le modèle objet GIT : tree, blob, tag et commit,
- Le répertoire GIT et le répertoire de travail,
- La zone de staging,
- Le dépôt: création et initialisation,
- Les branches tag & dépôt,
- Outil de visualisation Gitk.

Gestion de votre projet en local

- Recherche, ajout, modification et suppression de fichiers,
- Annulation et visualisation des versions,
- Parcours du versioning des différentes révisions,
- Les logs de formatage et de statistique.

Les branches

- La branche principale "master",
- Création de branches et de sous-branches,
- Gestion des conflits après une fusion de branche,
- Comparaison de deux branches.

Travail collaboratif

- Mise en place d'un dépôt distant public ou privé,
- Push: Publier ses modifications,
- Modifications de l'équipe,
- Les branches de suivi,

ADOPTER LE VERSIONING GIT

- Gestion des échecs.

Mise en œuvre des outils GIT

- La navigation graphique,
- La navigation graphique au sein des dépôts avec GITWeb
- Le service d'hébergement de dépôts avec GitHub,
- Le système de revue de code avec Gerrit.