Tache 1 : Intérêt de la gestion de version et présentation des outils existants de gestion de **versions**

**Gestion de Versions :**

La gestion de versions, également appelée \*\*contrôle de versions\*\* ou \*\*historique de versions\*\* (en anglais, \*Version Control System\* ou \*VCS\*), est une technique utilisée dans le domaine numérique pour suivre l'évolution d'un fichier ou d'un ensemble de fichiers au fil du temps. Elle permet de conserver plusieurs versions d'un projet, facilitant ainsi le suivi des modifications et la récupération d'anciennes versions si nécessaire.

**Principes de la Gestion de Versions :**

Suivi des Changements : Enregistrant chaque modification, elle permet de revenir facilement à une version précédente en cas de besoin.

Collaborative : Elle facilite le travail d'équipe en permettant à plusieurs utilisateurs d'apporter des modifications sur un même projet sans conflit.

Applications : Elle est applicable à un fichier unique (par exemple un document texte) ou à un ensemble de fichiers dans un projet de développement logiciel.

**Types de Systèmes de Gestion de Versions :**

1. Gestion de Versions Locale :

* Dans un système de gestion de versions locale, le dépôt (ou \*repository\*) qui conserve les modifications est stocké sur la même machine que les fichiers de travail.
* Cela signifie que l’historique des modifications reste localisé, ce qui peut être limitatif pour le travail collaboratif, mais pratique pour des projets personnels.

2. Gestion de Versions Centralisée :

* Dans la gestion de versions centralisée, le dépôt des modifications est conservé sur un serveur central, séparé des machines locales.
* Les utilisateurs travaillent sur la dernière version des fichiers de travail, mais doivent se connecter au serveur central pour récupérer les mises à jour ou enregistrer de nouvelles modifications.
* Ce système favorise la collaboration, car il permet à plusieurs utilisateurs de partager les mêmes ressources à partir d’un point central.

3. Gestion de Versions Distribuée (ou Décentralisée) :

* La gestion de versions distribuée combine les avantages des systèmes locaux et centralisés.
* Deux dépôts sont créés : l'un est local, sur la machine de travail, tandis que l'autre est centralisé sur un serveur distant (souvent une plateforme comme GitHub ou GitLab).
* Ce système offre une grande flexibilité pour le travail en équipe, permettant de travailler hors ligne et de synchroniser les modifications avec le serveur central ultérieurement.

**Gestion de Versions Distribuée**

La **gestion de versions distribuée** est un système qui permet aux utilisateurs de gérer les modifications d'un projet de manière indépendante et collaborative. Ce modèle est largement utilisé dans les projets open source et repose sur des dépôts locaux et centraux pour une flexibilité maximale.

**Fonctionnement de la Gestion Distribuée**

* **Exemple avec Git et GitHub** :
  + Dans ce système, chaque contributeur dispose d'un **dépôt local** sur son ordinateur, souvent dans un dossier nommé .git, qui contient l'historique complet du projet.
  + Ce dépôt local peut être synchronisé avec un **dépôt central** hébergé sur une plateforme comme **GitHub**. Ce dépôt central est accessible à plusieurs personnes.

**Avantages de la Gestion Distribuée**

1. **Contributions Simultanées** :
   * Plusieurs développeurs peuvent **contribuer** au même projet en envoyant leurs modifications au dépôt central. Ces changements peuvent ensuite être intégrés et partagés avec les autres utilisateurs.
2. **Base pour de Nouveaux Projets** :
   * Le dépôt central peut aussi être utilisé comme une **base** pour d'autres projets. Cela permet à la communauté de construire des dérivés ou des extensions à partir du projet initial.
3. **Travail Asynchrone** :
   * La gestion distribuée permet à chaque contributeur de travailler à son propre rythme, sans dépendre du timing des autres. Une fois que les modifications sont prêtes, elles peuvent être échangées et intégrées avec les autres dépôts.

**Importance pour les Projets Open Source**

La gestion de versions distribuée est cruciale pour les projets open source, car elle permet une collaboration flexible et indépendante. Grâce à des outils comme **Git** et **GNU Arch**, chaque contributeur peut travailler librement, tout en ayant accès à l’historique complet du projet. Les modifications sont partagées facilement, rendant le développement ouvert et transparent pour tous. Ce modèle favorise une participation active et permet au projet d’évoluer grâce aux contributions de chacun, sans dépendance au serveur central.

**Exemples d'Outils de Gestion de Versions**

Les outils de gestion de versions sont variés et adaptés à différents besoins, qu'il s'agisse de documenter des changements, de collaborer sur des projets logiciels, ou de suivre l'historique d'édition dans divers contextes. Voici quelques exemples courants :

1. **Wiki** : Suivi d'historique de versions dans les systèmes inspirés des wikis.
   * **Moodle** : Plateforme pédagogique permettant de retracer les versions de documents partagés.
   * **GitHub** : Utilisé pour organiser la documentation de projets et faciliter la collaboration.
   * **MediaWiki** : Logiciel open source qui gère l'historique des pages de Wikipédia et d'autres sites comme EduTech Wiki.
2. **Logiciels de Traitement de Texte** : Les versions en ligne de certains logiciels permettent de retracer l'historique des documents.
   * **Microsoft Office** (en ligne)
   * **Google Docs**
3. **CVS et SVN (Subversion)** : Outils de gestion de versions centralisés.
   * **CVS** (Concurrent Versioning System) et son successeur **SVN** permettent de gérer des projets avec un dépôt central.
4. **Logiciels SCM Décentralisés** :
   * **Mercurial** et **Git** : Ce sont des systèmes de gestion de versions décentralisés et open source, adaptés aux projets collaboratifs, avec une copie complète du dépôt pour chaque utilisateur.
5. **TFS (Team Foundation Server)** :
   * Outil de Microsoft, **TFS** est une solution payante de gestion centralisée des versions, souvent utilisée dans les grandes entreprises.