Versioning - Outillage

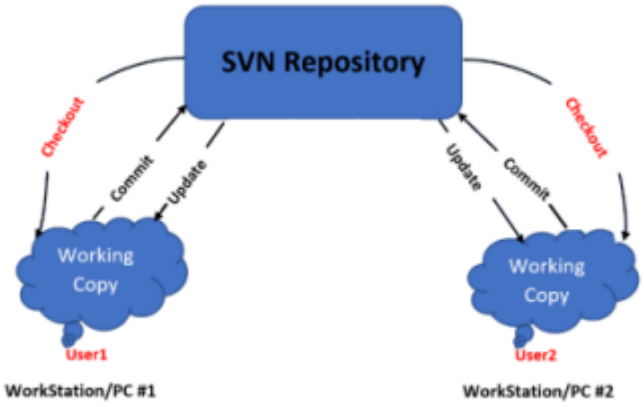
Rappel

La versioning est un **méthode de gestion de version** qui consiste à remonter des modifications dans le code source d'un projet tout en gardant les anciennes versions de celui-ci.

En cas d'erreur, il est alors possible de revenir en arrière et comparer les versions antérieures du code pour corriger les éventuelles erreurs. Un des nombreux logiciel permettant notamment de faire du versionning est ***Apache subversion (SVN)***.

Les principales opérations SVN

- **CHECKOUT**** :** Importation sur un client d'un projet mis en partage sur un serveur SVN
- **UPDATE**** :** Synchronisation du projet, du serveur vers le client (download des mises à jour)
- **COMMIT**** :** Synchronisation du projet, du client vers le serveur (upload des mises à jour)
- **DISCONNECT**** :** Déconnexion du partage (avec ou sans suppression des fichiers SVN locaux)



Organisation

Sur un serveur SVN, le projet, les Branches et les Tags sont stockés dans des dossiers spécifiques nommés res-pectivement « trunk », « branches », et « tags ». Ce n'est qu'une convention de nommage que vous êtes libres de respecter ou d'adapter, mais les bonnes pratiques vous conduiront naturellement à adopter ce nommage. L'intérêt des étiquettes sous Subversion est d'utiliser des noms symboliques plutôt que des numéros de révisions pour se référer à un état précis, comme par exemple 'release-1.1', plutôt que '488'. Un nom symbolique permet de revenir facilement à une version identifiée. Pour créer un tag, il suffit de copier l'état actuel d'une version de développement dans un sous-répertoire du dépôt. Les règles suivantes sont communément admises :

- Liste à puceLes différents tags correspondent chacun à un sous-répertoire, lui-même contenu dans un sous-répertoire nommé tags du projet ;
- On ne « commit » pas dans un tag ;

En fonction de l'ampleur et du nombre de projets contenus dans votre dépôt, l'emplacement de ces trois dos-siers peut varier. Il existe en fait deux formes recommandées en fonction de vos besoins :



Quelle que soit l'architecture choisie pour les dossiers de base, l'utilisateur crée tous les dossiers intermédiaires librement. Par exemple, la branche « germanVersion » pourra être stockée dans le dossier branches/germanVersion.
À consulter : http://www.lacl.fr/gava/cours/M2/lngLog/annexe3.pdf

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc2:prog:gen:svn&rev=1730816905

Last update: **2024/11/05 15:28**