

Problématique Versionning

➤ Gestion de versions centralisée ou décentralisée ?

L'un n'offre que qu'un seul dépôt qui fera référence tandis que l'autre pourra créer autant de dépôt que de collaborateurs au projet. Donc plusieurs dépôts pour un même logiciel.

Concrètement, l'intérêt d'utiliser un **VCS (Version Control System)** décentralisé plutôt qu'un **VCS Centralisé**, c'est que chaque membres du projet pourra travailler sur son dépôt local sans forcément avoir besoin d'être connecté. La plupart des opérations seront également traités en local.

Pour information, je précise qu'un dépôt est un espace de stockage géré par un **VCS**. Donc, un espace de stockage où se trouvera notre '**repository**' contenant l'ensemble des fichiers du projet.

➤ Git, et les autres ?

Mercurial, Bazaar sont apparus après la sortie de **Git**.

Mais nous choisirons d'utiliser **Git** car celui-ci nous est familier (Expériences & Connaissances de l'équipe & très documenté).

Git, est un gestionnaire de version décentralisée qui offre en tout point, un travail collaboratif. C'est à dire qu'il se distingue surtout par sa rapidité et sa gestion des branches qui permettent de travailler en parallèle sur le développement de l'application.

➤ Où héberger son projet ?

Nous utiliserons **GitHub** pour héberger notre projet **Git**. Le '**repository**' qui sera créée, sera accessible par tout les membres de l'équipe. Il sera ensuite '*forké*' par chacun des membres. Tous devront cloner le dépôt stocké sur le serveur vers sa machine.

➤ Et les branches ?

Alors, pour les branches, nous allons fonctionner de la manière suivante:

master : sera la branche qui contiendra la '*release*'.

dev : sera la branche de développement.

Puis, pour chaque ticket créé, nous allons créer des branches en référençant le numéro d'une issue qui elle même correspondra au travail à réaliser.

➤ Plus de détails

Le wiki du projet est accessible via ce lien :

<https://github.com/christophechevalier/MaderaMMB/wiki>

On y retrouve les méthodes de travail avec **Git**, l'Organisation de l'équipe projet, ainsi que les conventions de nommage qui seront appliqués lors du développement.