TUTO GIT (la base…)

Sousmission :
git add --all
git commit -a -m « *message* »
git push origin master

Recuperation :
git pull origin master

Branch :
git branch *nonBranche*
git checkout *nonBranche*

Merge :
git merge *nonBranche*
Puis aller sur branche master pour envoyer

git status	*-> Obtenir l'état des données*
git add --all	*-> Considération (ajout) de tout les nouveaux fichier*
git commit -a -m « *commentaire*»	*-> Prise en compte des modification par git*
git push origin master	*-> Envoie (depot local -> depot en ligne)*
git pull origin master	*-> Récupération (depot en ligne -> depot local)*
git branch *nonBranch*	*-> Creation d'une branche*
git checkout *nonBranch*	*-> Changement de branche de travail*
git pull origin master --allow-unrelated-histories	*-> si changement de repertoir (force le pull)*

/!\ ATTENTION /!\

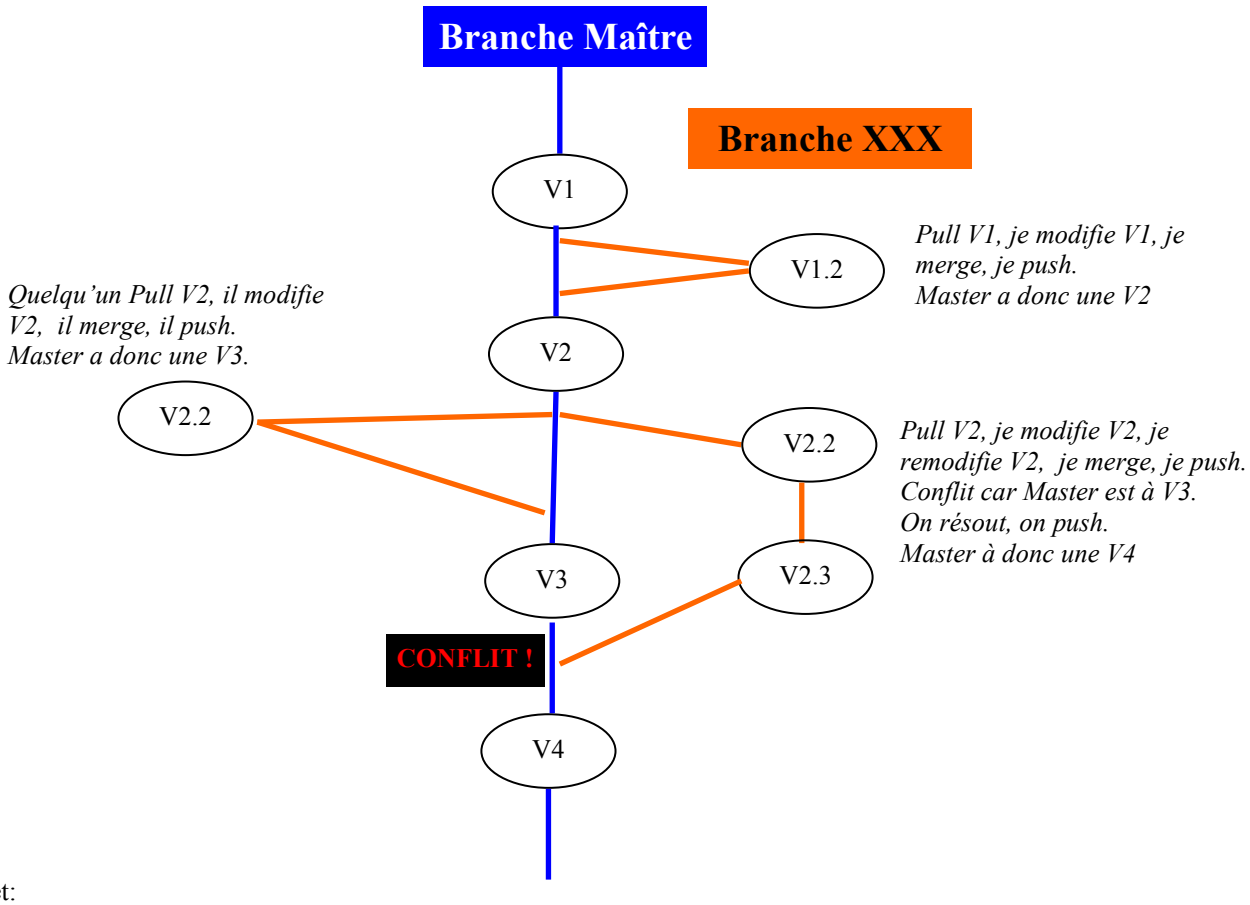
Mettre à jour dépôt local avant chaque modif !

GESTION DES CONFLITS :

Chacun travaillera sur un branche perso qu'il mergera avec la branche maître à la fin de son travail.

Sachant que le dépôt local se situe directement à l'emplacement des fichiers sources de yii, lors du changement de branche les fichiers de la branche Maître seront comme « remplacer » par ceux de la nouvelle branche. Il sera donc possible de tester immédiatement les modification sans affecter la branche maître.

En image :



Interet:

Le travail peut être fait sur la branche master directement. L'intérêt des branche est de pouvoir retrouver le travail perso sur notre branche avant d'avoir résolut le conflit (en cas de rater). On dispose donc d'une copie de notre travail.

PS : il est possible de créer autant de branche que nécessaire .. Vous pouvez créer une branche « testing » par exemple