

SI6

Gestion de Versions

GitHub

TFS

Le versioning

- * C'est la possibilité de gérer des modifications
- * Avec un historique
- * Sur des fichiers
- * De partager l'accès
- * De rendre disponible le source
- * De le faire progresser

Le versioning

- * Est nécessaire (ou pas : travail seul ?)
- * Parce que les OS évolues
- * Les langages évolues
- * Gestion d'un produit sur plusieurs plateformes (OS)

Le versioning

- * Beaucoup de produits :
 - * Git : décentralisé
 - * TFS : centralisé
 - * Mercurial : décentralisé
 - * SVN : centralisé
 - * Tortoise
 - * CVS : centralisé

Dépôt

- * Système centralisé :
 - * Un seul dépôt (connexion) : remote
 - * Une seule référence
- * Système décentralisé :
 - * Plusieurs dépôts (offline) : local, remote
 - * Plusieurs références
 - * Plusieurs historiques

Git et TFS

- * Git (linus Tovarld) : développement de linux (freix)
 - * Depuis 2005
 - * Réseau social GitHub (développeurs)
 - * Checksum (SHA1) : indexation
- * TFS à permis le développement des outils Windows
 - * Forge logicielle depuis 2005
 - * Lié à Visual Studio Team Edition (menu Team)

Git & TFS

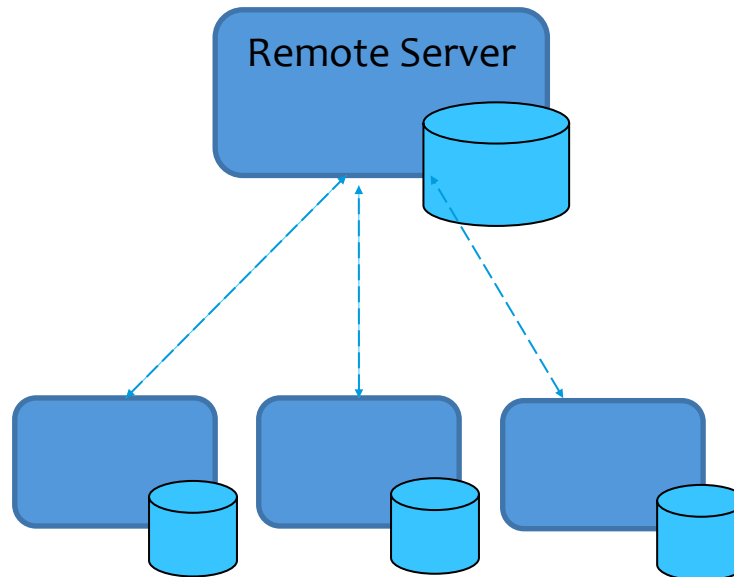
- * Git, gitHub = open source = gratuit
- * TFS = \$\$\$ (gratuit pour une équipe de 5)
- * Utilité du versioning :
 - * Gérer le travail en équipe (des développeurs)
 - * Avancement de différentes versions
 - * Diff, merge,
 - * Fork, branch,

Git & TFS

- * Git est un système distribué
- * Git peut fonctionner offline
- * Repository local avec Xcode
- * VS avec TFS nécessite une connexion
- * VS et TFS supportent Git

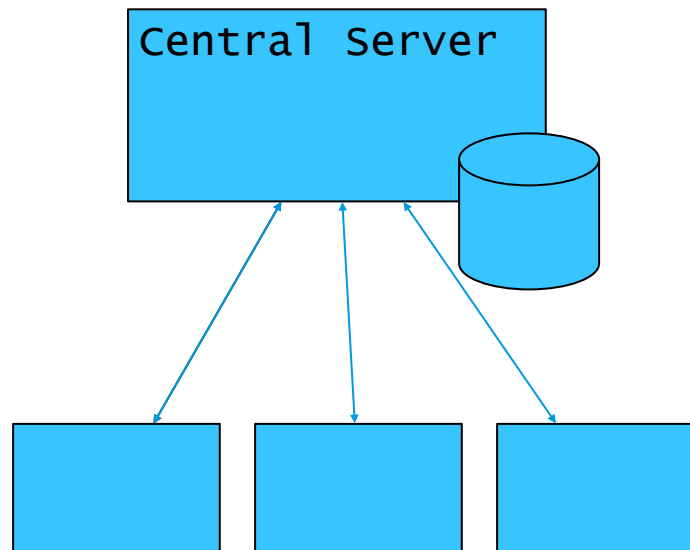
Git

- * Système distribué : (client/serveur) = asynchrone



TFS

- * Système centralisé : (client/serveur) = panne ???



Git & TFS

- * Eclipse, Netbeans, Xcode = Git voir TFS
- * Visual Studio = TFS voir Git
- * Projet GitTFS
- * L'utilisation d'une application Git favorise le versioning avec Xcode
- * TFS fait du source control comme Git
- * TFS + VS + ALM = solution complète d'industrialisation de la conception

ALM

- * Définir (utilisateur)
 - * Spécifications fonctionnelles
- * Développer (développeurs)
 - * Spécification techniques
 - * UML/SCRUM (agilité)
- * Exploiter (métiers)
 - * Gestion des incidents, débogage
 - * Déploiement
- * Git ne sait pas faire tout cela