

Environnement de développement

Au minimum

- **MySQL 5**
- **Eclipse 3.7** (ou 4.2) ou bien NetBeans
- projet de type "dynamic web project" ou "entreprise app (web+ejb)" sans maven
- modélisation avec **StarUML (ou ...)** sans génération de code évoluée
- pas de gestion de version , échanges de fichier (entre développeurs de l'équipe) par copies (usb ou réseau)

Conseillé

- Environnement minimum +
- **Référentiel de code source (SVN ou GIT ou ...)** pour :
 - gérer les différentes versions
 - partager le code entre les membres d'une équipe
 - sécuriser l'accès au code (en cas de perte locale , le référentiel est d'un bon secours)
- Utilisation de **maven** pour structurer le(s) projet(s) pour :
 - gérer les dépendances (vis à vis de Spring / Hibernate / JSF , ...)
 - ne stocker dans le référentiel que le code source
 - reconstruire facilement le projet
 -

Eventuellement (phase 2)

- **Intégration continue** avec **Jenkins** (maven indispensable)
- Référentiel d'artifacts maven avec **Nexus 2** (pour gérer ses propres frameworks ,)
- Tests d'intégration avec **Selenium** ou ...
- Tests de performances avec **JMeter** ou ...
- Génération de code évoluée avec **MDA/accéléo_m2t** (à partir de modèle **UML2** construit avec *topcasedUML* ou ...)
-

Conseils :

- **La mise au point d'un bon environnement de développement demande du temps (et donc de s'y prendre à l'avance).**
- Pour le référentiel de code source GIT ou SVN , on pourra au choix :
 - s'appuyer sur des offres gratuites de type "googlecode" ou "gitHub"
 - s'appuyer sur un référentiel GIT ou SVN géré par une de ses propres machines (machine virtuelle linux avec virtualBox en mode pont/bridge ou ...)
- Il faudra éventuellement penser à installer et configurer les plugins eclipse nécessaires (pour GIT/SVN , Maven , ...)
- **Tout assemblage de technologies doit être testé (eclipse + maven + svn/git) !!!!**
- Ne pas sous estimer les éventuels réglages de sécurité (mots de passe,)
- ...