## Environnements de développement

L’**Usine logicielle** est un ensemble structuré de logiciels qui fonctionnent ensemble pour permettre le développement. Il s’agit non seulement des outils de programmation, compilation, tests, déploiement et de sauvegarde mais également de gestion de documents, de gestion de cycle de vie d’un projet, de suivi des activités, de gestion des licences. Tous ces outils étant bien entendu connecté entre eux.

En soit il s’agit d’un tableau de bord.

## Gestion de la configuration

La gestion de la configuration a trois buts principaux :

1. Gérer la description technique d'un système et de ses divers composants càd gérer l’inventaire de tous les éléments de la configuration (configuration items).

2. Gérer l'ensemble des modifications apportées au cours de l'évolution du système.

3. Gérer les versions (au travers de la nomenclature (convention de nommage) que l’on donne au fichier notamment).

Pourquoi ?

1. Gestion de la complexité (Nombreux développeurs, sous-systèmes indépendants, différents langages, évolution des systèmes).
2. Pour avoir un logiciel fiable, robuste mais surtout maintenable.

Les objectifs poursuivis sont :

1. Que les demandes de changement et les rapports de problèmes soient gérés dans le système. Pour cela on peut :
   * Utiliser un programme de gestion de bugs
   * Vérifier que chaque demande de client soit introduite dans le logiciel.
2. Que des bugs résolus précédemment ne réapparaissent pas mystérieusement. Afin de nous aider on peut :
   * Si cela ouvrir, rouvrir le bug et le marquer dans le programme de gestion des bugs.
   * Compter le nombre de bugs rouverts dans une version (Il devrait être égal à 0).
3. Que les activités de développement soient sous contrôle. Pour cela on peut :
   * Essayer de produire un état des développements en moins d’une demi-journée.
4. Que la documentation soit à jour et correcte (Ceci est difficile à atteindre). Pour cela on peut :
   * Maintenir un numéro de version et la date de dernière mise à jour du document dans chaque document produit.
   * Chaque document devrait avoir un numéro de version et la date lui correspondant.
5. Que toutes les « configurations items » soient identifiés.
6. Que chaque release puisse être construite précisément.
7. Que les différences entre deux releases soient facilement identifiées.
8. Qu’une release envoyée à un client soit complète (pas de fichier manquant).

## Inventaire des éléments et leur gestion

L’inventaire peut être composée de plusieurs choses :

* Documentation
* Méthodes de travail
* Spécifications techniques
* Définition des bases de données
* Scénarios de tests
* Enregistrement des tests
* Dossiers de validation
* Composants externes
* Dossiers sécurité
* Code source
* Environnement
* Exécutables
* Ressources nécessaires au développement
* Fichiers INI
* Scripts pour créer la base de données

Cela permet d’organiser les informations, de créer des conventions de nommage, de permettre une nomenclature de versions des fichiers, d’avoir des documents relatifs à l’assurance et au contrôle qualité.

Pour cela on va utiliser un standard. Il s’agit d’une spécification technique établie par l’équipe et qui repose sur les acquis de la science, des technologies et de l’expérience.

## Organisation des informations

On va définir une arborescence pour stocker les différents items de configuration et le lien avec le numéro de version du logiciel :

* Les fichiers de documentation
* Les exécutables
* Les jeux de tests
* Les enregistrements de l’exécution des tests

On va également définir la structure des numéros de version du logiciel et définir la structure des documents.

Afin de mieux organiser l’information, on va utiliser une nomenclature : n.m.ooooo.p. Pour lequel n.m représente la release majeure, ooooo le numéro de build et p le numéro séquentiel de branchement (par défaut : 0).

On peut également utiliser des **templates** : tous les documents devront dès lors avoir le même modèle.