

# 1 EXIGENCES

Elles sont stockées dans un fichier XML dont la structure est décrite ci-dessous.

## 1.1 Composition

Chaque exigence est composée de 10 champs :

**numero** : Numéro formé comme suit : REQ\_XXXXX, où x est un chiffre de 0 à 9

**titre** : Titre ou description courte

**corps** : Expression de l'exigence

**type** : Type de l'exigence : Utilisateur - Métier - Système - Contrainte

**nature** : Nature de l'exigence : Fonctionnelle - Ergonomie - Robustesse - Performance - Sécurité

**origine** : D'où vient une exigence ?

**version** : Initiale - Intermédiaire - Finale

**priorite** : Priorité selon la méthode MoSCoW - Must - Should - Could - Won't

**validee** : L'exigence a-t-elle été validée ? (Oui / Non)

**lien** : il peut y en avoir plusieurs ; ils sont regroupés dans l'élément "Liens".

**test** : Définition du test qui validera l'exigence.

## 1.2 Edition

Elle peut se faire avec « Eclipse », qui permet la validation (par rapport au schéma) au fur et à mesure de la saisie. Une commodité dans « Eclipse » est la liste des énumérations disponibles sur un élément, en tapant Ctrl + Espace dans l'élément.

## 1.3 Extraction

L'export des informations contenues dans le fichier XML se fait à l'aide de scripts XSLT :

**Exigences.xml** : génère le fichier « Exigences.tex » qui sera inclus automatiquement dans le rapport du projet.

**Exigences\_csv.xml** : génère le fichier « Exigences.csv » qui peut être lu par n'importe quel tableur.

**Exigences\_html.xml** : génère le fichier « Exigences.html » qui peut être lu par n'importe quel navigateur internet.

J'utilise le processeur XSLT de Saxonica. Ce processeur fonctionne avec Java et ne s'installe pas, il suffit juste de le désarchiver dans un dossier.

Le fichier « gen.cmd » permet de lancer la génération ; avant il faut mettre à jour la variable d'environnement « XsltProcPath » avec le chemin d'accès au processeur « Saxon ».