

Fonctionnalités attendues sur projet Atod (2014)

1.1. Quatre grands axes complémentaires

- **Modélisation** (avant tout fonctionnelle et un peu technique / conception)
- **Ergonomie**
- **Technicité et modularité du code**
- **Intégration et gestion de projet** (travail d'équipe efficace)

1.2. Axe "Modélisation" (et présentation) – 5 pts

- **Bonne utilisation d'UML** ("uses cases" , "diag classes" , "stéréotypes" , ...)
- **Bonne modélisation** des principaux **processus métiers** (via **UML** et/ou **BPMN**)
- **Bonne rédaction des SFG/SFD** et "**spécifications d'architecture technique**"
- Prise en compte des aspects distribués (*SOA* , *urbanisation* , *MDM* , *référentiels*).
- **Bonne présentation** (claire , efficace ,) des points **pertinents** de la modélisation .

1.3. Axe "Ergonomie" – 5 pts

- **Application opérationnelle** (qui **fonctionne** et qui **couvre le périmètre minimum attendu**)
- **Interface graphique WEB claire et intuitive**
- **bonne utilisation** (directe ou indirecte) d'**ajax** (auto-complétion, rafraîchissements partiels, ...)
- **Guide utilisateur** et/ou **aide intégrée à l'application** (menu "help/aide")
- **Bonne gestion des messages d'erreurs/anomalies** et **contrôles de saisies** .

1.4. Axe "Technicité / modularité du code" - 5pts

- **Persistance** opérationnelle (via **Hibernate/JPA** ou autre) et **base de données**
- **Services métiers** (**EJB3** ou bien **Spring**) opérationnels (avec pour chacun d'eux une classe de **test unitaire JUnit**)
- **Services WEB** opérationnels (avec tests "soap-ui" ou)
- **Code modulaire** (applications de certains "**design patterns**" (ex: **DAO** , **DTO** , **Factory**)) + éventuelles couches logicielles + éventuels sous projets (ex: modules/projets "maven")
- Un minimum de "**secrétité**" (ex: **authentification**) et de **technologie(s) asynchrone(s)** (envoi d'e-mail et/ou **JMS** , ...)

1.5. Axe "Intégration / gestion de projet" - 5pts

- **Bonne utilisation d'un référentiel de code source** (ex: **SVN** , **GIT** , **Mercurial** , ...)
- Un minimum attendu de **maven** (et un maximum conseillé , éventuellement avec intégration continue (**Jenkins**))
- **Modélisation/développement par itérations** (en suivant les recommandations des méthodes agiles (**XP** , **SCRUM** ,))
- **Bonne dérivation "Modèles UML/BPMN → code java/..."** (via éventuelle utilisation d'un générateur de code paramétrable (ex: **Acceleo M2T / MDA**) ou via)
- **Bonne intégration** des technologies (via **classes utilitaires** ou **framework** "maison")