# Évaluation de la qualité du logiciel

Un logiciel de qualité doit satisfaire toutes les exigences fonctionnelles et non-fonctionnelles qui ont été identifiées lors de l'analyse des exigences. Dans un cadre académique, les exigences sont l'énoncé du travail.

En théorie, une exigence devrait être clair, complète, précise et sans ambiguïté. Si vous vous trouvez devant une exigence ayant plusieurs interprétations possibles, posez des questions pour clarifier la situation.

En plus des exigences de l'énoncé, ces critères de qualité suivants sont également évalués :

- le respect des standards du web;
- la qualité du code source.

## Le respect des standards du web

Les standards sont séparés par technologie ou concept.

### HTML

- Le HTML doit être valide. Voir https://validator.w3.org/.
- Les noms d'éléments et d'attributs doivent être en minuscules.
- Les classes et id doivent être en minuscules.
- La sémantique des éléments doit être respectée.

### CSS

- Le CSS doit être valide. Voir https://jigsaw.w3.org/css-validator/.

### Front-end

- Le front-end doit être responsive sur ordinateur, tablette et téléphone.
- Le front-end doit être utilisable sur la dernière version des fureteurs Chrome, Firefox, Edge.

### REST

- Les URLs doivent suivre les patrons vus en classe.
- Les services doivent être documentés.
- Les services POST, PUT, PATCH doivent retourner un objet utile au client.
- Les services doivent échanger des données en JSON, sauf si spécifié autrement.

### HTTP

- Il faut respecter la sémantique des méthodes HTTP.
- Une route doit retourner un code de statut HTTP le plus précis possible, tout en respectant la sémantique des codes de statut.

### Conception

- Utilisez le patron POST-REDIRECT-GET, sauf si spécifié autrement.
- Tous les éléments de configuration (noms d'utilisateur, mots de passe, jetons pour l'utilisation d'un API) doivent être dans un fichier de configuration. La structure de ce fichier doit être documentée.

## La qualité du code source

Un effort constant doit être fait afin de rendre le code simple, lisible et facile à maintenir. Ceci implique les éléments suivants :

- La nomenclature doit être de qualité. Les variables doivent être nommées selon leur contenu. Les classes doivent être nommées selon leur concept du domaine ou leur responsabilité. Les packages ou modules doivent être nommés selon leur thème. Le logiciel doit être nommé. Les fonctions ou méthodes doivent être nommées selon l'action qu'elles accomplissent, en débutant par un verbe.
- Le style doit être uniforme et suivre les standards du langage utilisé (ex. Java Coding Conventions pour Java; PEP8 pour Python, etc.).
- Une attention particulière doit être donnée à l'indentation, qui doit être uniforme et constante. L'indentation doit être faite avec des espaces, et non des tabulations.
- Les fonctions et méthodes ne font d'une seule opération. Les fonctions et méthodes doivent demeurer courtes. Considérez un maximum de 25 lignes de code par fonctions/méthodes, sauf si spécifié autrement. Ceci s'applique également au main d'un programme.
- Les classes n'ont qu'une seule responsabilité.
- La responsabilité d'une classe devrait être documentée à l'aide d'un commentaire au début de la classe. Chaque méthode et fonction doit également être documentée à l'aide d'un commentaire.
- Toutes les erreurs systèmes (les erreurs normales) doivent être gérées par le logiciel. En aucun cas un utilisateur ne devrait voir un « stack trace ».
- Évitez de mélanger les langages de programmation dans un même fichier.