|  |  |  |  |
| --- | --- | --- | --- |
| Q1: Est-ce qu’il y a assez de tests? |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Pour évaluer la première question on commence par compter le nombre de méthodes de tests qu’il y a par rapport au nombre de méthodes fonctionnels il y a dans le projet. Ainsi on établie la première métrique qui est le rapport de tests par rapport au nombre de méthodes fonctionnelles. De cette manière nous arrivons à avoir un rapport 1 pour 1 qui démontrerait grossièrement que chaque méthode fonctionnel du projet a un test. Parfois il est nécessaire de faire attention avec cette métrique car elle n’élicite pas une couverture de code mais en fait une estimation.

Nous pouvons répondre à la deuxième question à l’aide de la métrique *méthodes de test / méthodes fonctionnelles*. Présumant que chaque méthode fonctionnel aie besoin que d’une méthode de test on peut estimer que les tests sont à jour avec le restant du code s’il y a au moins un test par méthode de l’application.

En utilisant simplement

Le nombre d’assertions par méthode de test serait une bonne métrique de complexité.

Quote du projet que l’on se sert.

LOC (Lines of code): It counts the lines of count, ignoring empty lines and comments (i.e., it's Source Lines of Code, or SLOC). The number of lines here might be a bit different from the original file, as we use JDT's internal representation of the source code to calculate it.