Nombre d'Interfaces Implémentées :

Définition : c'est le nombre d'interfaces implémentées par classe. L'implémentation d'une interface nécessite la redéfinition de ses méthodes abstraites (on exclut les méthodes par défaut), ce qui augmente le nombre de méthodes par classe (la métrique Nombre de Méthodes par Classe), conduisant ainsi à une complexité accrue du code et à un nombre de lignes plus important (la métrique Nombre Total de Lignes Brutes), et donc éventuellement à un risque accru d'erreurs.

Nombre d'Imports par Classe :

Définition : C'est le nombre de classes importées par une autre classe. Importer un plus grand nombre de classes augmente la complexité du code, et donc le risque d'erreurs dû à une utilisation incorrecte éventuelle des méthodes importées. Même si certaines classes sont importées sans être utilisées, elles alourdissent le programme et conduisent à un temps de compilation plus long.

Nombre d'Imports Effectifs par Classe :

Définition : Dans ce cas, nous nous intéressons uniquement aux classes effectivement importées, c'est-à-dire les classes utilisées dans le code (pour être plus précis).

Nombre de Méthodes Utilisées des Classes Importées :

Définition : C'est le nombre total de méthodes obtenues à partir des classes importées et utilisées dans la classe.

Nombre d'Attributs Instanciables :

Définition : Un nombre important d'attributs objets (de type non primitif) augmente la probabilité de bugs dans ce code orienté objet en raison du polymorphisme, de l'héritage et du fait que l'attribut peut être utilisé sans être instancié.

Nombre Total d'Exceptions Possibles :

Définition : Dans cette mesure, nous comptons le nombre d'expressions de code pouvant lever une exception.

Nombre d'Exceptions Gérées :

Définition : Dans cette mesure, nous nous intéressons au nombre d'expressions gérées dans le code.

Facteur d'Exceptions : Nombre d'Exceptions Gérées / Nombre Total d'Exceptions Possibles.

Cette mesure représente le pourcentage d'exceptions possibles gérées par le développeur dans son code.

Nombre de méthodes utilisées à partir des imports

Gestion des dépendances : Une utilisation élevée des méthodes provenant des imports peut indiquer un couplage fort entre les modules ou les classes, ce qui pourrait rendre le code plus difficile à maintenir et à étendre.

Cohésion du code : Si une classe utilise un grand nombre de méthodes provenant de divers imports, cela pourrait suggérer que la classe a plusieurs responsabilités ou manque de cohésion. Il peut être nécessaire de refactoriser pour améliorer la clarté et la maintenabilité.

Complexité des tests : Les classes avec une utilisation élevée de méthodes provenant des imports pourraient nécessiter des configurations de tests plus complexes en raison des dépendances à des fonctionnalités externes. Cela peut augmenter l'effort requis pour les tests et potentiellement introduire plus de points de défaillance.

Impact sur les performances : Importer de grands modules juste pour utiliser quelques méthodes peut avoir un impact sur les performances de l'application, surtout si ces imports sont lourds ou ont un coût d'initialisation significatif.

Problèmes de code : La dépendance excessive aux méthodes provenant des imports pourrait être un signe de mauvais choix de conception ou d'implémentation, tels que la violation du Principe de Responsabilité Unique ou l'incapacité à encapsuler correctement les fonctionnalités liées.