

Devoir 2 – IFT3913

Tâche 1:

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

Métrique 1: **Taux de couverture de code par les tests.**

- Raisonement: Un taux élevé de couverture indique que la majeure partie du code est testée.
- Comment mesurer: Utiliser des outils comme JaCoCo ou Cobertura.

Métrique 2: **Nombre de tests par classe/module.**

- Raisonement: Indique la densité des tests pour chaque unité de code.
- Comment mesurer: Examiner le code des tests.

Q2: Est-ce que les tests sont à jour avec le reste du code?

Métrique 1: **Taux de couverture de code par les tests** (Réutilisation).

- Raisonement: Une baisse du taux de couverture peut indiquer que le code a évolué, mais pas les tests.
- Comment mesurer: Utiliser des outils comme JaCoCo ou Cobertura.

Métrique 3: **Date du dernier commit pour les tests vs. Date du dernier commit pour le code.**

- Raisonement: Compare la fraîcheur des tests par rapport au code.
- Comment mesurer: Utiliser l'historique des commits sur GitHub.

Q3: Est-ce que les tests sont trop complexes?

Métrique 2: **Nombre de tests par classe/module** (Réutilisation).

- Raisonement: Un grand nombre de tests pour une petite unité de code pourrait indiquer une complexité excessive.
- Comment mesurer: Examiner le code des tests.

Métrique 4: **Complexité cyclomatique moyenne des tests.**

- Raisonement: Mesure directement la complexité des tests.
- Comment mesurer: Utiliser des outils d'analyse de code.

Q4: Est-ce que les tests sont suffisamment documentés?

Métrique 5: **Taux de commentaires dans les tests.**

- Raisonnement: Mesure directement la documentation des tests.
- Comment mesurer: Utiliser des outils d'analyse de code.

Métrique 3: **Date du dernier commit pour les tests vs. Date du dernier commit pour le code** (Réutilisation).

- Raisonnement: Une date récente peut indiquer des ajouts ou des modifications, donc une possibilité de mise à jour de la documentation.
- Comment mesurer: Utiliser l'historique des commits sur GitHub.