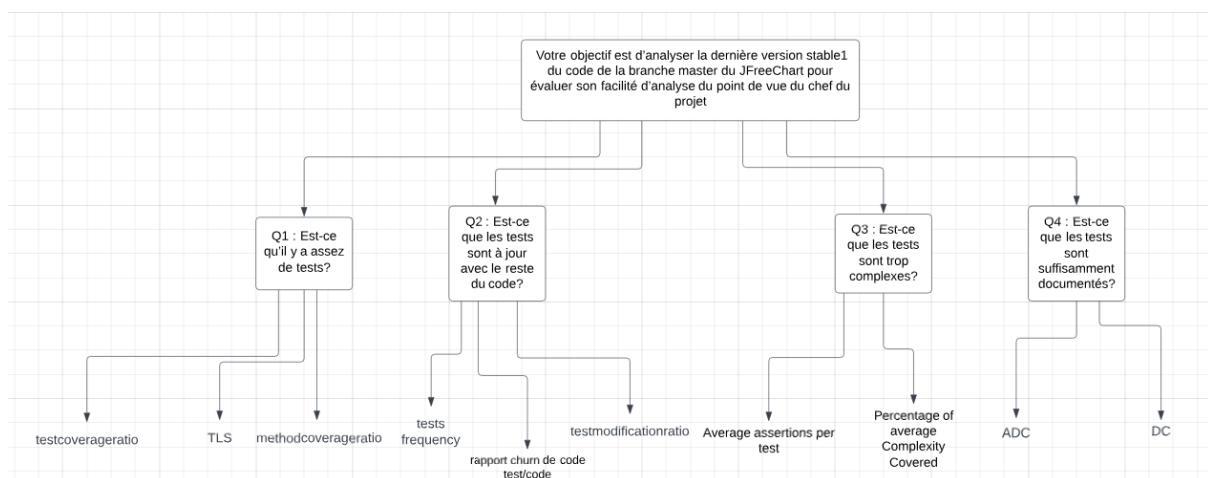


RAPPORT TP2 -IFT3913

Nom: Ould Saada
Prénom: Ibrahim
Matricule: 20151421

Nom: Mahdi
Prénom: Moghadasi
Matricule: 20184188

Tâche 1:



Q1: y a-t-il assez de tests ?

- TLS:

Nous utilisons la métrique TLS qui paraît très pratique pour calculer le nombre de lignes de code par rapport aux tests. Plus cette valeur est petite, plus ça veut dire qu'on a des asserts.

nous rapprochant d'une réponse positive à la question.

mesure: le code TLS du tp1 avec un peu de modification

- TestcoverageRatio :

Cette métrique calcule le pourcentage d'instructions couvert par le code, elle nous permet de savoir combien d'instructions sont couvertes par les tests, plus il y a d'Instruction couverte plus ça veut dire qu'on se rapproche d'un nombre satisfaisant de tests.

mesure: Librairie Jacoco et notre code java pour manipuler ses données

- methodcoverageRatio:

Cette métrique calcule pourcentage de méthodes couvertes, couverte par les tests. Elle nous permet de savoir combien de méthodes de codes sont couvertes par les tests. Plus il y a d'Instructions couverte, plus ça veut dire qu'on se rapproche d'un bon nombre satisfaisant de tests.

mesure: Librairie Jacoco et notre code java pour manipuler ses données

l'utilisation de ces métriques combinées nous permettra de déterminer qu'il y a déjà un nombre suffisant de tests et de vérifier que ces tests testent le maximum d'instructions et de méthodes.

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

- TestCommitFrequency : C'est le pourcentage de commit de tests. Une faible fréquence de commits de test pourrait indiquer que les tests ne sont pas mis à jour aussi fréquemment que le reste du projet. Cela pourrait être un signe que les tests ne sont pas à jour avec le code.
mesure: nouvelle implémentation pour ce tp
- testModificationRatio: C'est le rapport entre les commits de test et les commits de code qui ne sont pas de tests. Un faible rapport de modification des tests pourrait également indiquer que les tests ne sont pas mis à jour aussi fréquemment que le code. Cela pourrait être un autre signe que les tests ne sont pas à jour avec le code.
mesure: nouvelle implémentation pour ce tp
- rapport churn de code test/code: Le churn de code est une mesure de combien de lignes de code ont été ajoutées ou supprimées sur une **période spécifique**. Un churn de code élevé peut indiquer que la base de code subit des changements significatifs ou une refonte. Cela pourrait également signifier que de nouvelles fonctionnalités sont ajoutées ou supprimées. Le rapport consiste à diviser le churn de code de test sur churn de code de code. plus cette valeur est élevée, plus ça veut dire qu'il y a de changement dans notre test par rapport au code dans cette période spécifique.
mesure: nouvelle implémentation pour ce tp

Les métriques sélectionnées offrent une vision approfondie de la fréquence à laquelle les tests sont mis à jour. En examinant le rapport entre les commits de test et les commits totaux et le rapport churn de code, nous pouvons évaluer si les tests sont régulièrement maintenus et synchronisés avec le code principal. Cette approche croisée garantit une évaluation plus complète de la mise à jour des tests, offrant ainsi une assurance supplémentaire quant à la cohérence et la fiabilité de notre suite de tests

Q3: Est ce que les tests sont trop complexes?

- Average assertions per test: mesure le nombre d'assertions par tests, plus le nombre d'assertion par tests est grand plus il y a de chance que nos tests soient trop complexes.
mesure: nouvelle implémentation pour ce tp
- Percentage of average Complexity Covered: le pourcentage de complexité moyenne couverte est une métrique qui représente la proportion de la complexité cyclomatique moyenne couverts par des tests, par rapport à la complexité totale moyenne des tests (couverts et manqués). Plus cette mesure est proche de 100, plus les tests couvrent de scénarios et de chemins dans notre code source et donc les tests sont plus complexes.
mesure: Librairie Jacoco et notre code java pour manipuler ses données

cette question est un peu vague parce qu'on était pas sur comment définir complexité. avec la mesure Average assertions per test on vérifie la complexité des classes tests et des méthodes tests et avec l'autre, étant Percentage of average Complexity Covered, on mesure la complexité dans le sens où à quel points nos tests couvrent tous les différents chemins du code source cyclomatiquement.

Q4: Est ce que les tests sont suffisamment documentés ?

- DC (densité de commentaires) :
En examinant le pourcentage de commentaire dans les fichiers de tests, on peut se donner un avis sur la documentation globale des tests. Pour ceci, on a divisé le CLOC sur TLOC.
mesure: réutilisation de mesures du tp1 avec modification
- Average DC : mesure le pourcentage de commentaire moyen par classe de tests, ce qui donne une répartition de la documentation sous forme de commentaire dans les classes de tests.
mesure: réutilisation de mesure du tp1 avec modification

La combinaison de ces deux métriques nous permet d'avoir une bonne idée sur le pourcentage de documentation de tests en forme de commentaire dans les tests et sa répartition à travers des classes de tests.

TÂCHE 2:

Pour la tâche deux nous avons utilisé jacoco ainsi que les metrique utilisé au TP1 pour mesurer un bon nombre de métriques,
Nous avons décidé d'implémenter nous même les métriques Testcommitfrequency, TestModificationRatio, rapport churn de code test/code,averageassertion,

TÂCHE 3:

Pour Q1 :

Résultat:

TLS : 13.620962764863197

seuil = 8

Test Coverage Ratio: 0.552815705552242

seuil = 80%

Method Coverage Ratio: 0.5996583996632228

- pour TLS on a mis un seuil supérieur de 8 ligne de code pour 1 seul assert et comme TLS= 13, alors elle n'atteint pas le seuil. Pour les 2 autres métriques, nous avons établi un minimum de 80% et comme seulement 55% des instructions et 59% des méthodes sont testées, on n'atteint pas nos objectifs . Comme la majorité des seuils ne sont pas atteints, la reponses pour le nombre de tests semble être NON. Il n'y a pas assez de tests

Pour Q2:

Test Commit Frequency: 0.08647239990523573	seuil = 25%
Test Modification Ratio: 0.09465767634854771	seuil = 25%
rapport churn de code test/code (derniere année): 0/104	seuil = 25%

Ici notre seuil pour les 2 premières mesures est posé à 0.25, indiquant que pour 4 commits de code on est censé avoir au moins 1 commit de test. Comme on voit seulement 8.6% des commits totaux sont des tests et la ratio test modification est égale à 0.09, on n'atteint pas le seuil. En plus, selon rapport churn de code test/code, on a 0 modification des tests dans la dernière année par rapport à 104 modification de code ce qui confirme notre résultat indiquant que les tests ne sont pas à jour. pour rapport churn de code test/code , on a ainsi pris en compte un seuil de 0.25.

Donc on arrive à la conclusion que NON les tests ne sont pas a jours

Pour Q3:

Average assertions per test: 4.17636761487965	seuil = 6
Percentage of average Complexity Covered: 0.4362609251758687	seuil = 80%

pour average assertion per test nous avons mis un seuil supérieur de 6. Comme notre résultat est 4, nous sommes dans le seuil, ce qui indique que les tests écrits ne sont pas trop complexes puisqu'on a seulement 4 assertions dans chaque test en moyenne.

Cependant pour *percentage of average complexity covered*, on a 43% et puisque le seuil n'est pas atteint, on peut dire que les tests ne sont pas trop complexes et que les tests ne couvrent pas assez de chemins cyclotomiques dans notre code source.

Dans les 2 cas et les 2 sens de complexité, on conclut que les tests ne sont pas trop complexes.

Pour Q4 :

Average Comment Density (ACD) : 44,43%	seuil = 30 %
Comment Density (CD) : 37,67%	seuil = 30 %

Ici, nous avons décidé de mettre un seuil minimum de 30% que CD et ACD atteignent pour les commentaires et nous pouvons conclure que OUI les tests sont assez documentés.

Conclusion:

En conclusion, nous pensons que donner une réponse sur la facilité d'analyse est une tâche complexe. Cependant, les métriques que nous avons trouvé peuvent donner une bonne idée sur les questions même si les questions sont assez vagues et qu'il est difficile de trouver des métriques pour avoir des résultats très précis. De plus, d'autres critères peuvent entrer en jeu car selon le contexte, une même mesure peut être interprétée différemment, tout cela fait qu'il est difficile de donner une réponse définitive sur la question de la facilité d'analyse du projet.