

TP2: Qualité de la conception, anomalies et refactoring

Ce travail pratique se concentre sur les sujets de la qualité de conception, l'identification des anomalies à l'aide de métriques et la correction de ces anomalies par l'application de refactorings. Toutes les questions doivent être répondues au système de AntennaPod (<https://github.com/AntennaPod/AntennaPod>). Pour répondre aux questions, vous pouvez utiliser n'importe lequel des outils présentés ou mentionnés dans le laboratoire ou en classe. Dans tous les cas, vous devez mentionner explicitement tous les outils et sources que vous avez utilisés. Soumettre un seul document par équipe avec les réponses aux questions. Inclure le nom de votre équipe et les noms de tous les membres de l'équipe dans le rapport. Inclure toutes les références externes, y compris les articles, les liens, la documentation et les outils.

Date de remise : 21 mars, 23h59

Question 1: Mesurer la qualité de conception d'AntennaPod. (30pts)

- a. Présentez un rapport avec toutes les métriques pour AntennaPod (cohésion, complexité, couplage, taille) y compris les moyennes, les maximums, les écarts-types, etc. Présentez les métriques dont nous avons déjà discuté (NOM, LOC, LCOM, WMC, CC, CBO etc.). Présenter les métriques au niveau approprié (projet, package, classe).

(5 pts)

Q: Quelles métriques devons-nous présenter ? Comment savoir si nous en avons inclus suffisamment ?

R: Règles simples : "Présentez tous les indicateurs auxquels vous faites référence dans le rapport. Présentez toutes les métriques que vous utilisez pour calculer d'autres métriques."

- b. Commentez la qualité globale du système et son impact sur le développement et la maintenance du système (par exemple, "le système est cohérent, ce qui garantit le principe de responsabilité unique (SRP) et facilite la maintenance"). (5 pts)

Q : Comment commentons-nous ? Quel est le niveau attendu des commentaires ?

R: Commentaire basé sur ce que nous avons vu jusqu'à présent, concernant les principes, les propriétés de qualité de conception. Le niveau doit être assez élevé et abstrait. Ne vous concentrez pas (encore) sur des cas spécifiques.

- c. Dans les diapositives 32 à 34 du troisième cours, nous avons vu comment nous pouvons également mesurer les métriques pour les critères ISO 9126 (Fonctionnalité, Réutilisabilité, Compréhensibilité, Flexibilité, Efficacité, Extensibilité). Utilisez ce guide et l'article référencé pour quantifier les critères de qualité. (5 points)
- d. Décrivez la qualité du système en fonction des critères métriques. Il n'y a pas de valeurs de référence, mais supposez qu'une valeur élevée implique une qualité plus élevée. (Remarque : la compréhensibilité peut être négative, mais ce n'est pas un problème) (5 pts)
- e. Corréler ces valeurs avec les valeurs des métriques individuelles (NOM, LOC, LCOM, WMC, CC, CBO etc.) par rapport à la qualité générale du système. Commenter l'impact des métriques individuelles sur les critères de qualité. Un guide pour cette explication se trouve dans l'article ci-joint. Donnez trois (3) exemples spécifiques qui démontrent cet impact. (10 pts)
Note : consultez cet article

<http://www.ptidej.net/team/admission/Bansiya02-QualityModel.pdf>

Question 2: Trouvez cinq (5) anomalies différentes dans AntennaPod. (30pts)

- a. Définissez les seuils des métriques selon le livre "Object-Oriented Metrics in Practice" (disponible sur Moodle) ou selon les diapositives de la Semaine 4.

Si nécessaire, combinez plusieurs métriques. Une anomalie peut être présente même si elle ne viole pas tous les seuils. Utilisez votre jugement pour identifier les anomalies. (15pts)

- b. Présentez les anomalies et justifiez leur présence à l'aide des métriques. Il est possible que les outils ne considèrent pas certains cas comme des anomalies. Dans ce cas, ignorez les outils. (10pts)
- c. Nommez les anomalies en termes de code smells (par exemple, low cohesion + high coupling = Feature Envy) (5pts)

Question 3: Corriger les anomalies en appliquant des refactorings (40pts)

- a. Expliquez pourquoi chaque anomalie est corrigée par le refactoring que vous avez choisi. (10 points)
- b. Appliquez les refactorings et présentez le code avant et après le refactoring. Si le code est trop long, présentez des diagrammes UML ou une version réduite du code (par exemple, avec seulement les signatures des méthodes). (10pts)
- c. Évaluez à nouveau la qualité par rapport aux métriques individuelles (LCOM, CC, WMC, CBO, etc.) et vérifiez que l'anomalie est corrigée du point de vue des métriques. La correction peut améliorer la moyenne du système ou simplement les métriques locales pour les classes. Rapportez tout. (10pts)
- d. Mesurer les métriques correspondant aux critères de qualité (Fonctionnalité, Réutilisabilité, Compréhensibilité, Flexibilité, Efficacité, Extensibilité). Y a-t-il une amélioration en termes de ces mesures ? Justifiez les changements et expliquez-les en fonction des métriques individuelles. (10 points)

Outils recommandés

Pour les métriques :

1. MetricsReloaded - <https://plugins.jetbrains.com/plugin/93-metricsreloaded>
2. MetricsTree - <https://plugins.jetbrains.com/plugin/13959-metricstree>
3. Better Code Hub <https://bettercodehub.com/>

4. Understand <https://scitools.com/features/>

Pour les refactorings/code smells:

1. Android Studio - <https://developer.android.com/studio>
2. SonarCloud - <https://sonarcloud.io/>

Remarques de soumission et d'évaluation

Nommez votre rapport comme « TP2_[nom_équipe].pdf ».

Le document sera évalué pour l'exactitude et l'exhaustivité des réponses et la qualité de l'écriture. Traitez-le comme un rapport officiel et professionnel.