

SFE

Revue de code et refactoring : Module de mesure de température

Centrale Nantes

P.-E. Hladik, pehladik@ec-nantes.fr

—

Version bêta (26 mai 2025)

Objectifs pédagogiques

- Identifier des problèmes de qualité dans un code source.
- Analyser et améliorer la lisibilité, la robustesse, la modularité et la réutilisabilité d'un programme.
- Comprendre l'intérêt de la séparation des responsabilités et des fonctions bien nommées.
- Travailler en équipe pour construire une critique argumentée et proposer un refactoring.

Contexte simulé

Vous êtes intégrés dans une équipe de développement embarqué travaillant sur un prototype de station météo. Votre collègue vous transmet un premier prototype fonctionnel de lecture de plusieurs capteurs de température et de détection de seuil d'alerte. Il fonctionne, mais ne respecte pas les normes de qualité de votre entreprise. Votre mission : effectuer une revue de code complète, en équipe, et proposer un plan de refactoring clair, accompagné de vos recommandations.

Matériel fourni

- Fichier `temp_monitor.c`
- Fiche de revue de code.
- Grille d'évaluation des qualités du code.
- (Facultatif) Cahier des charges fonctionnel (si tu veux en ajouter un).

Déroulement pédagogique

1. Lecture : Lecture individuelle du code fourni (15 min)
2. Revue technique en binômes : identifier les défauts et proposer des améliorations (20 min)
3. Synthèse de groupe (4 pers) : Mise en commun, priorisation des critiques, rédaction de la grille de revue de code (15 min)
4. Rédaction du rapport : Chaque groupe rédige un rapport clair (1 page, voir ci-dessous) (20 min)
5. Mise en commun : Présentation orale rapide par chaque groupe, discussion collective (20 min)
6. Refactoring de code en binôme : Modifier le code pour en augmenter les qualité (20 min)

7. Relecture du code refactoré (binôme) : relecture et évaluation du code en remplissant la grille (20 min)

8. Bilan de la séance

En fonction de la durée restante : utiliser les outils d'analyse de complexité et de respect des normes MISRA sur le code, puis écrire quelques contrats Frama-C.

Vous déposerez le petit rapport, votre code refactoré ainsi que les deux grilles de revue sur Hippocampus.

Consignes pour le rapport de groupe

Votre rapport (1 page max) devra inclure :

- Une synthèse claire des problèmes identifiés dans le code source initial.
- Des propositions de correction/refactoring pour chacun des problèmes.
- Une conclusion sur les apports de l'exercice (retour d'expérience, bonnes pratiques retenues).