

Grille de correction détaillée – EC09

Informations générales

- **Épreuve** : EC09 – Guide de bonnes pratiques
- **Durée** : 4h – Individuelle
- **Projet fil rouge** : SkillHub – Guide interne de qualité logicielle
- **Nature** : Épreuve écrite (guide au format Markdown ou PDF)

Critères d'évaluation et attentes

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
C8.6	Normes de code et documentation	Guide propose des conventions claires : nommage, organisation du projet, documentation minimale (README, commentaires, schémas).	Vérifier que les règles sont concrètes, pas trop génériques. Exiger des exemples précis (bonnes/mauvaises pratiques).
C9.6	Stratégie de tests	Présentation d'une stratégie cohérente : types de tests (unitaires, intégration, E2E), couverture minimale attendue, outils associés.	Vérifier si les recommandations sont réalistes et applicables (ex : Jest, PHPUnit, Cypress). Pas de stratégie trop vague (« faire des tests »).
C14.3	Qualité logicielle globale	Guide propose des critères mesurables (revues de code, respect de normes, couverture de tests). Approche	Attention aux guides trop théoriques ou irréalistes. Vérifier présence de checklists concrètes.

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
		pragmatique et applicable par une équipe.	

Livrables à corriger

- **Guide de bonnes pratiques** : document structuré et lisible (Markdown ou PDF)
- **Exemples concrets** : conventions de code, Git, checklist de review
- **Stratégie de tests** : plan clair avec couverture minimale et outils proposés
- **Annexes éventuelles** : sous-dossier `exemples/` , README explicatif

Barème indicatif (sur 20 points)

Axe évalué	Points
Conventions de code et documentation (C8.6)	/7
Stratégie de tests (C9.6)	/7
Qualité logicielle globale (C14.3)	/6

Tolérance : ± 2 points selon la lisibilité, la cohérence et la pertinence globale du guide.

Points positifs attendus

- Guide clair, structuré avec titres, sous-parties et checklists
- Conventions de code détaillées (noms de variables, branches Git, architecture projet)
- Exemples concrets de bonnes et mauvaises pratiques
- Stratégie de tests réaliste avec répartition claire (unitaires, intégration, E2E)
- Références à des outils précis (linters, formateurs, frameworks de test)
- Approche pragmatique : règles applicables en contexte d'équipe

Erreurs fréquentes à surveiller

- Guide trop théorique, copié-collé de normes génériques
- Absence d'exemples concrets (difficile à appliquer en équipe)
- Stratégie de tests floue ou irréaliste (ex : exigence de 100% de couverture sans justification)
- Oubli des aspects documentation (README, commentaires, schémas)
- Checklists trop superficielles ou inapplicables
- Document désorganisé, illisible ou trop long sans synthèse

Rappel pédagogique

Cette épreuve doit valider que l'apprenant sait :

- **Formaliser des standards de qualité** pour encadrer une équipe de développement
- Définir une **stratégie de tests pragmatique et mesurable**
- Produire un **document exploitable** dans un contexte professionnel (lisible, applicable, concret)

Elle permet d'évaluer la capacité de l'apprenant à adopter une posture de **développeur responsable**, garant de la qualité et de la maintenabilité d'un projet collectif.