

Répondez dans le cahier de réponses.

Question #1 – 10%

Lors de la rédaction de tests unitaires, quelle est l'utilité d'un mock object?

Question #2 – 10%

Que faut-il éviter de faire dans un bon test unitaire?

Question #3 – 10%

Selon l'auteur de «Coder proprement», quelle approche est la meilleure pour gérer les erreurs dans le code, les exceptions ou les codes de retour? Expliquez votre réponse. (Gestion des erreurs – chapitre 7)

Question #4 – 10%

Si vous gérez un projet de logiciel libre utilisant les bibliothèques «commons» d'Apache et regroupant une vingtaine de collaborateurs à travers le monde, quel outil de construction automatisée allez-vous privilégier et pourquoi?

Question #5 – 10%

Les langages de programmation dynamique comme Ruby et Groovy sont souvent décrits comme des langages Agiles. Pourquoi?

Question #6 – 10%

Le déploiement quotidien est une des pratiques d'eXtreme Programming les plus difficiles à mettre en place. Quels sont les prérequis nécessaires afin de pouvoir appliquer cette pratique?

Question #7 – 10%

La méthode Scrum ne précise pas quelles pratiques de la réalisation de logiciels doit-on utiliser lors d'un projet (ex. tests unitaires, refactoring, révision de code). Si la méthode ne le précise pas, comment savoir quelles pratiques utiliser lors de l'écriture du code?

Question #8 – 10%

La méthode Scrum propose de tenir une courte réunion quotidienne appelée le «daily scrum». Quels sont les avantages d'adhérer à une telle pratique?

Question #9 – 20% (5% par affirmation)

Vrai ou faux? Pour chacune des affirmations suivantes, indiquez si elle est vraie ou fausse. **Justifiez** chacune de vos réponses.

1. L'outil de build Ant est un outil de type «Convention over configuration»
2. L'utilisation d'un makefile favorise la portabilité d'un logiciel
3. L'expressivité de Java est supérieure à celle de Groovy
4. eXtreme Programming est une méthode stricte et contraignante