

Question #1

Selon le manifeste Agile, lequel des éléments suivants n'est pas une priorité lors d'un projet?

- a) Des logiciels opérationnels
- b) L'adaptation au changement
- c) Le suivi d'un plan
- d) La collaboration avec les clients
- e) Aucune de ces réponses

Question #2

Quelle pratique permet d'accueillir positivement un changement de besoin lors d'un projet assez avancé?

- a) Le développement itératif
- b) Le refactoring
- c) Les tests
- d) Le développement en cascade
- e) Écrire du code propre
- f) Aucune de ces réponses

Question #3

Selon le manifeste Agile, «*les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.*» Pourquoi?

- a) Pour assurer un suivi plus serré des développements
- b) Pour obtenir régulièrement un feedback de qualité
- c) Pour assurer une productivité constante des développeurs
- d) Pour vérifier le respect des échéanciers
- e) Toutes ces réponses

Question #4

Quel est le moyen de communication recommandé par le manifeste Agile?

- a) Le courriel
- b) Le texto
- c) Le téléphone
- d) L'échange de documents normalisés
- e) Aucune de ces réponses

Question #5

Selon le manifeste Agile, quelle est la principale mesure d'avancement d'un projet?

- a) Les fonctionnalités opérationnelles
- b) Les documents d'analyse et le code source
- c) La satisfaction du client
- d) La satisfaction des utilisateurs
- e) Le budget consommé
- f) Aucune de ces réponses

Question #6

Parmi les éléments suivants, lequel a été un facteur motivant la création du manifeste Agile?

- a) Les mauvaises estimations des développeurs
- b) La piètre qualité des développeurs sur le marché
- c) L'attitude négative des développeurs face au changement
- d) L'insatisfaction constante des clients
- e) Toutes ces réponses

Question #7

Parmi les éléments suivants, lequel n'est pas une partie prenante lors d'un projet de développement d'un logiciel?

- a) Les développeurs
- b) Les commanditaires
- c) L'équipe de maintenance
- d) L'équipe d'assurance-qualité
- e) La direction
- f) Aucune de ces réponses

Question #8

Dans un gestionnaire de sources, qu'est-ce qu'un dépôt centralisé?

- a) Un dépôt sur la machine du développeur, que les autres développeurs peuvent utiliser
- b) Un dépôt privé pour un développeur centralisé
- c) Un dépôt unique contenant tous les projets de l'entreprise
- d) Un dépôt qu'on accède au travers d'un réseau et que tous les membres de l'équipe utilisent
- e) Un logiciel de publication de code source
- f) Aucune de ces réponses

Question #9

Qu'est-ce que le typage dynamique?

- a) Le type doit être connu à la compilation
- b) Le type doit être respecté à l'exécution
- c) Le type peut changer à la compilation mais pas à l'exécution
- d) Le type est connu seulement à l'exécution et peut changer durant l'exécution
- e) Le type n'est jamais connu
- f) Le langage n'utilise qu'un seul type

Question #10

Parmi ces langages de programmation, lequel est le moins expressif?

- a) Groovy
- b) Java
- c) Python
- d) C++
- e) Ruby
- f) C
- g) Langage d'assemblage
- h) L'expressivité est la même pour tous ces langages

Question #11

Quel serait l'avantage d'utiliser le langage de programmation Groovy?

- a) Beaucoup de fonctionnalités en peu de lignes de code
- b) Excellente compatibilité avec Java et la JVM
- c) Meilleure productivité des programmeurs
- d) Meilleure lisibilité du code source
- e) Toutes ces réponses

Question #12

Quel serait l'inconvénient d'utiliser le langage de programmation Groovy?

- a) Peu de bibliothèques offertes
- b) Difficile à mettre en place dans un projet Java
- c) Le mélange du Groovy et du Java pourrait nuire à la lisibilité
- d) Langage verbeux
- e) Toutes ces réponses

Question #13

Selon le modèle de développement en cascade, quand devons-nous effectuer des tests sur le logiciel produit?

- a) À chaque itération
- b) Après la réalisation
- c) Pendant la maintenance
- d) Après l'analyse organique
- e) Jamais
- f) Après l'analyse des besoins
- g) Après le déploiement

Question #14

Qu'est-ce qu'un écosystème de développement?

- a) L'ensemble des développeurs de l'équipe
- b) L'ensemble des développeurs de l'entreprise
- c) L'ensemble des règles de codification
- d) L'ensemble des outils des développeurs
- e) L'environnement physique des développeurs

Question #15

Pourquoi est-il important d'avoir un style uniforme dans le code source?

- a) Améliorer la performance du logiciel
- b) Améliorer la vitesse d'écriture du code
- c) Améliorer la lisibilité du code source
- d) Améliorer le sentiment d'appartenance des développeurs dans l'équipe
- e) Toutes ces réponses

Question #16

Pourquoi voudrions-nous faire du refactoring?

- a) Pour rendre le code plus facile à modifier
- b) Pour mieux gérer le risque
- c) Pour que les développeurs aient une bonne confiance en eux
- d) Parce que l'équipe d'assurance-qualité l'exige
- e) Toutes ces réponses.

Question #17

Vous faites parti d'une équipe de trois développeurs et vous avez toujours suivi le même style instinctivement. Un nouveau développeur arrive dans l'équipe et vous réalisez que son style est différent. Comment allez-vous adresser ce problème?

- a) Ce n'est pas un problème; le nouveau développeur va sûrement finir par faire comme les autres
- b) L'équipe demande au nouveau développeur de suivre le style déjà en place dans le code
- c) L'équipe en profite pour se rencontrer et élaborer ensemble une norme de codification
- d) L'équipe rejette le nouveau développeur
- e) Toutes ces réponses, dans cet ordre

Question #18

Laquelle des affirmations suivantes est fausse?

- a) Le développement itératif vise à produire un logiciel exécutable à la fin de chaque itération
- b) Le développement incrémental vise à ajouter des nouvelles fonctionnalités au fur et à mesure que le projet avance
- c) Le développement évolutif vise à constamment faire évoluer le logiciel
- d) Le développement itératif vise à produire la documentation du projet au départ
- e) Toutes ces réponses

Question #19

Quel élément n'est pas une méthode efficace pour gérer le risque d'un projet?

- a) Faire l'analyse et la documentation complète du système avant de commencer la programmation
- b) Utiliser le prototypage
- c) Commencer le projet par les fonctionnalités les plus difficiles et complexes
- d) Effectuer des preuves de concept
- e) Aucune de ces réponses

Question #20

Quelle est l'utilité de la commande `git pull`?

- a) Créer un dépôt local avec les sources d'un dépôt distant
- b) Créer un dépôt local à partir d'un dépôt local d'un collègue
- c) Mise à jour du dépôt local avec les changements du dépôt distant
- d) Mise à jour du dépôt distant avec les changements du dépôt local
- e) Tirer les sources du projet
- f) Connexion avec le dépôt distant

Question #21

Qu'est-ce qu'un beautififier?

- a) Un outil de vérification automatique du style
- b) Un outil de mise en forme automatique du code source
- c) Un outil pour produire de la documentation en format texte
- d) Un outil pour renommer automatiquement les variables et classes
- e) Une norme pour structurer la documentation dans le code source
- f) Aucune de ces réponses

Question #22

Quelle affirmation suivante est fausse?

- a) La consommation du budget est la principale mesure d'avancement d'un projet
- b) L'excellence technique favorise l'agilité
- c) Une bonne phase d'analyse renforce l'agilité
- d) Le développement itératif complexifie la négociation contractuelle
- e) Aucune de ces réponses

Question #23

Quelle affirmation suivante est vraie?

- a) Groovy utilise du typage statique
- b) Java utilise du typage statique
- c) Le compilateur Java supporte aussi le Groovy
- d) Groovy est plus verbeux que Java
- e) La métaprogrammation n'est pas possible avec Groovy
- f) Toutes ces réponses

Question #24

Quelle affirmation suivante est fausse?

- a) En Java, l'indentation doit être de 4 espaces
- b) En Java, l'accolade ouvrante apparaît en fin de ligne
- c) Il peut être pertinent d'avoir un comité qui détermine les règles de style
- d) Les règles de style peuvent être vérifiées par des outils
- e) En Java, les noms de méthodes sont en camelCase
- f) Aucune de ces réponses

Question #25

Quel est le principal avantage d'avoir un dépôt local avec Git?

- a) Permettre de renommer un fichier et de conserver son historique
- b) Pouvoir faire des modifications dans les fichiers sources
- c) Pouvoir utiliser une interface web comme github.com
- d) Pouvoir bénéficier des avantages du gestionnaire de sources, même sans connectivité au réseau
- e) Aucune de ces réponses

Question #26

Quel type de document ne devrait-on pas retrouver dans un gestionnaire de sources?

- a) Un document d'analyse
- b) Un prototype jetable
- c) Un script de tests automatisés
- d) La documentation utilisateur
- e) Aucune de ces réponses

Question #27

Qu'est-ce que le refactoring?

- a) Modifier du code existant, sans modifier la fonctionnalité
- b) Réécrire du code pour le rendre plus lisible
- c) Retravailler le code pour en améliorer la maintenabilité
- d) Refaire la conception dans le but de la simplifier
- e) Toutes ces réponses

Question #28

Que devons-nous faire si nous découvrons un bogue dans le code actuel pendant que nous effectuons une activité de refactoring?

- a) Corriger le bogue en même temps que le refactoring
- b) Terminer le refactoring et corriger le bogue ensuite
- c) Laisser le bogue en place et le documenter pour une correction ultérieure
- d) Annuler l'activité de refactoring pour corriger le bogue car la correction aura peut-être un impact significatif sur le code, ce qui rendrait notre refactoring inutile
- e) Aucune de ces réponses

Question #29

Comment peut-on faire du refactoring de façon sécuritaire si nous n'avons pas de tests unitaires dans le projet?

- a) Écrire des tests unitaires sur les parties touchées par le refactoring
- b) Faire des petites modifications peu risquées à l'aide des outils de refactoring automatique de NetBeans
- c) Faire plusieurs petits changements et tester manuellement l'application après chaque changement
- d) Travailler minutieusement et faire vérifier son travail par un collègue expérimenté
- e) Toutes ces réponses

Question #30

L'utilisation d'un gestionnaire de sources est un prérequis à l'agilité. Pourquoi?

- a) Parce que ça permet aux développeurs d'annuler rapidement une modification erronée et de remettre le dépôt dans un état stable
- b) Parce que toutes les équipes de développeurs utilisent ce genre d'outil
- c) Parce que le refactoring nécessite l'utilisation d'un gestionnaire de sources
- d) Parce que le développement itératif nécessite l'utilisation d'un gestionnaire de sources
- e) Toutes ces réponses