

Énoncé : A

Question #1

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

- a) Remettre à zéro l'historique d'un dépôt
- b) Créer un dépôt distant
- c) Créer un dépôt local
- d) Copier le dépôt distant sur le poste local
- e) Créer un fichier README dans le dépôt local
- f) d et e
- g) Aucune de ces réponses

Question #2

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

- a) Copier sur le dépôt distant les commits du dépôt local
- b) Copier sur le dépôt local les commits du dépôt distant
- c) Fusionner une branche distante avec une branche locale
- d) Fusionner une branche locale avec une branche distante
- e) Annuler l'effet d'un `git push`
- f) Aucune de ces réponses

Question #3

Quel est le rôle de `github.com` dans votre écosystème de développement?

- a) Effectuer le partage des tâches dans l'équipe
- b) Héberger le dépôt distant de l'équipe
- c) Installer git sur les postes de travail de l'équipe
- d) Fournir des outils de communication pour le travail en équipe
- e) Faire la mise en forme automatique du code source
- f) Effectuer des tests en ligne
- g) Toutes ces réponses

Question #4

Le programmeur A et le programmeur B sont dans la même équipe de développement (qui travaille avec git) et ils sont en train de modifier le même fichier source au même moment.

Le programmeur A fait un commit.

Le programmeur A fait un push.

Le programmeur B fait un commit.

Le programmeur B fait un push.

Que survient-il lors du push du programmeur B?

- a) Le programmeur A doit gérer un conflit
- b) Le programmeur B doit gérer un conflit
- c) Rien, tout se passe normalement car git fait la fusion des modifications
- d) Le push échoue parce que les sources ne sont pas à jour
- e) Le push échoue parce que le fichier est réservé par le programmeur A
- f) Aucune de ces réponses

Question #5

Quelle est la durée maximale recommandée pour une itération?

- a) 1 an
- b) 6 mois
- c) 3 mois
- d) 2 mois
- e) 1 mois
- f) Toutes ces réponses, ça dépend de la durée du projet

Question #6

Selon le manifeste Agile, quel type d'équipe produit les meilleures architectures, spécifications et conceptions?

- a) Une équipe auto-organisée
- b) Une équipe avec un gestionnaire de projet
- c) Une équipe sans leader
- d) Une équipe en télé-travail
- e) Une équipe qui travaille dans le même bureau à aire ouverte
- f) Une équipe qui discute en face-à-face

Question #7

Quel élément n'est pas une étape dans le modèle de développement en cascade?

- a) Analyse des besoins
- b) Analyse fonctionnelle
- c) Analyse organique
- d) Programmation
- e) Tests
- f) Stabilisation
- g) Maintenance

Question #8

Que devrait-on faire à la toute fin d'une itération?

- a) Des tests
- b) Planifier l'itération suivante
- c) Réfléchir sur les éléments qu'on peut améliorer sur notre façon de travailler
- d) Demander l'avis du client
- e) Demander l'avis des utilisateurs
- f) Célébrer
- g) Discuter des fonctionnalités à livrer dans la prochaine itération

Question #9

En Java, où est utilisé le type de casse PascalCase?

- a) Dans le nom des constantes
- b) Dans le nom des classes
- c) Dans le nom des méthodes publiques
- d) Dans le nom des variables d'instance publiques
- e) Dans le nom des variables locales
- f) Dans le nom des méthodes statiques

Question #10

Quel élément de style a le plus grand impact sur la lisibilité du code source?

- a) La position des accolades
- b) L'aération verticale
- c) L'aération horizontale
- d) L'ordre des éléments dans un fichier source
- e) La nomenclature
- f) L'indentation

Question #11

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 #12

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 #13

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 #14

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 #15

Laquelle des affirmations suivantes est fausse?

- a) Le modèle de développement en cascade est un modèle assez vieux.
- b) Le modèle de développement en cascade est encore largement utilisé.
- c) Le modèle de développement en cascade est optimal lorsque les spécifications du logiciel sont connues au départ et ne changeront pas.
- d) Le modèle de développement en cascade favorise la planification et les estimations précises.
- e) Aucune de ces réponses

Question #16

Dans quel contexte est-il préférable d'utiliser un gestionnaire de sources?

- a) Pour un petit projet (moins de 3 mois)
- b) Pour un gros projet (plus d'un an)
- c) Pour un développeur unique
- d) Pour une petite équipe de développement (moins de 5 personnes)
- e) Pour une grande équipe de développement (plus de 10 personnes)
- f) Pour un travail universitaire
- g) Toutes ces réponses

Question #17

Quelle affirmation suivante est vraie?

- a) git est dispendieux
- b) git est lent, surtout lorsqu'on doit faire des branches
- c) git est un gestionnaire de sources décentralisé
- d) git est un gestionnaire de sources obsolète
- e) Subversion est une version améliorée de git
- f) SVN est un gestionnaire de sources décentralisé

Question #18

Quel est l'avantage d'adhérer à une norme de codification?

- a) Un exécutable plus performant
- b) Un processus de développement plus prévisible
- c) Du code source uniforme
- d) Moins de bogues dans les logiciels
- e) Détecter les bogues plus rapidement
- f) Des coûts de développement moins élevés
- g) d et e
- h) b et f
- i) Aucune de ces réponses

Question #19

Quel est l'objectif principal du refactoring?

- a) Améliorer la qualité du code source
- b) Améliorer la productivité des développeurs
- c) Assurer un meilleur suivi des corrections de bogues
- d) Permettre un retour en arrière en cas d'erreur
- e) Améliorer la gestion des erreurs du logiciel
- f) Toutes ces réponses

Question #20

Comment le modèle de développement itératif permet-il d'obtenir une meilleure satisfaction du client?

- a) Après chaque itération, on déploie le logiciel chez le client pour le tester
- b) Après chaque itération, on présente le logiciel au client pour recueillir son feedback
- c) Après chaque itération, on demande de nouvelles spécifications
- d) Avant chaque itération, on demande au client de définir ses attentes
- e) Aucune de ces réponses

Question #21

Dans quel contexte le modèle de développement en cascade est-il un **bon** choix?

- a) Lorsque les spécifications sont inconnues.
- b) Lorsque les spécifications changent régulièrement.
- c) Lorsque que le projet est risqué.
- d) Lorsque les estimations doivent absolument être respectées.
- e) Lorsque les spécifications sont connues et ne changeront pas.
- f) Aucune de ces réponses.

Question #22

À quel moment devrions-nous faire du refactoring?

- a) En fin de projet, s'il reste du temps.
- b) Après le déploiement en production.
- c) Après chaque itération.
- d) Un peu chaque jour.
- e) Toutes ces réponses.

Question #23

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 #24

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 #25

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 #26

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 #27

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 #28

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 #29

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 #30

Qui devrait rédiger une norme de codification?

- a) La direction
- b) Le client
- c) L'équipe de développement
- d) Les ressources humaines
- e) Un consultant
- f) Le développeur avec le plus d'expérience