I. GENERALITES ET PROCESSUS

1. Qu'est-ce qui n'est pas un facteur d'échec pour un projet ? a. le manque de support de la hiérarchie b. l'absence de résultats mesurables réguliers c. l'intégration des utilisateurs à la fin du développement d. l'utilisation d'un langage de développement orienté objet e. la réduction du temps dédié à la conception	D

2. Qu'est-ce qui n'est pas un principe fondamental du GL ? a. la modularisation b. l'abstraction c. l'encapsulation d. UML e. la séparation des préoccupations	D

3. Pourquoi faire une étude de faisabilité ? a. pour préparer les morceaux de code les plus importants b. pour identifier les exigences conflictuelles c. pour résoudre les conflits d. pour s'assurer de l'intérêt technique et « business » d'un logiciel à venir e. pour tester le logiciel à l'issue de son développement	D

4. Quelle est l'une des limites majeures des processus en cascade : a. l'absence de méthode formelle associée b. la difficulté de faire des tests c. l'absence de conception architecturale d. le manque de flexibilité e. leur coût en termes humain et monétaire	D

5. Quand n'est-il pas recommandé d'utiliser un processus agile ? a. lorsque les exigences sont mal connues b. lorsque les exigences sont évolutives c. lorsque les délais sont courts d. lorsque les clients sont peu disponibles e. lorsque les performances attendues du logiciel sont fortes	D

II. GESTION DES EXIGENCES

6. Qu'est-ce qu'une exigence non fonctionnelle? a. une exigence liée à une fonction ayant peu ou pas d'importance b. une exigence liée au à plusieurs fonctions c. une exigence définie par plusieurs *stakeholders* d. une exigence liée à une qualité logicielle e. une exigence non liée à une fonction	D

7. Quel est la meilleure façon d'identifier les exigences d'un logiciel très interactif? a. faire des interviews structurées b. faire des interviews non structurées c. faire des cas d'utilisation d. faire un prototype e. faire un « workshop »	D

8. Quel est le but premier d'un modèle d'analyse ? a. préparer le code, notamment objet b. préparer la conception c. exprimer des exigences non fonctionnelles d. exprimer des exigences de façon semi formelle e. exprimer des exigences de façon formelle	D

9. Qu'est-ce qu'un acteur ? a. un humain interagissant avec le logiciel b. un humain fournissant des entrées au logiciel c. un porteur d'intérêt ou une partie prenante d. un humain ou un système interagissant avec le logiciel e. un humain ou un système fournissant des entrées au logiciel	D

10. Quel est le but premier des cas d'utilisation ? a. définir les fonctions essentielles du logiciel b. définir les exigences non fonctionnelles essentielles du logiciel c. définir les acteurs d. définir la portée des actions des acteurs e. définir les tests de la future recette	A

III. CONCEPTION ET ARCHITECTURE

11. Qu'est-ce qu'un modèle ? a. une représentation abstraite, simplifiée et biaisée d'une réalité b. une spécification d'architecture c. une représentation UML d. une représentation complète d'un aspect donné e. un schéma d'interaction simplifié	A

12. Qu'est ce que le couplage a. une mesure qualitative des liens liant les modules d'un logiciel b. une mesure quantitative des liens liant les modules d'un logiciel c. une mesure de la capacité d'évolution d'un logiciel d. une mesure de la cohérence sémantique des modules d'un logiciel e. une mesure de la réutilisation dans un logiciel	A

13. Trouver l'affirmation qui est fausse a. une architecture correspond à la première étape de conception b. une architecture sert à identifier les acteurs c. une architecture sert à affiner les délais et les coûts d. une architecture sert à répartir les efforts de développement sur différentes équipes e. une architecture sert à effectuer les premières validations	B

14. La conception en logiciel repose essentiellement sur a. l'étude des exigences b. l'étude des exigences et les règles internes de l'entreprise c. Les règles internes de l'entreprise d. les « design patterns » et les règles internes de l'entreprise e. les « design patterns »	B

15. Le style d'architecture de type « publish/subscribe »est a. flexible à l'exécution b. clair c. facile à tester d. sécurisé e. conçu pour faciliter la maintenance	A

-

16. UML est a. une notation minimaliste b. une notation maximaliste c. un outil de génération d'architecture d. une méthodologie orientée objet e. un ensemble de notation formelle	B

17. Indiquer l'affirmation vraie a. Les « design patterns » permettent la réutilisation de conception b. Les « design patterns » permettent la réutilisation en Java essentiellement c. Les « design patterns » se concentrent sur la réutilisation de code d. Les « design patterns » visent la définition de tests réutilisables e. Les « design patterns » se limitent à l'approche orientée objet	A

18. Comment valider un diagramme UML a. en le montrant aux clients b. en construisant des diagrammes d'instance c. en y appliquant des cas d'utilisation d. en essayant d'ajouter de nouvelles exigences e. en générant du code et en le testant	B

19. La validation a. est une activité permanente b. est une activité survenant uniquement en fin de développement c. se concentre sur le test unitaire de code d. se concentre sur la sécurité et la disponibilité e. se concentre sur la sécurité, la disponibilité et la montée en charge	A

20. Les optimisations de code a. ne nuisent pas à la lisibilité du code b. ne nuisent pas à la maintenance c. ne nuisent pas à la définition des tests fonctionnels d. ne nuisent pas au deboguage e. ne nuisent pas au « refactoring » du code	C