

Série d'Exercices

N° 01

1. Quiz :

1. Le modèle de développement des logiciels en cascade est
 - a. Une approche raisonnable lorsque les exigences sont bien définies.
 - b. Une bonne approche lorsqu'un programme de travail est requis rapidement.
 - c. La meilleure approche à utiliser pour les projets avec de grandes équipes de développement.

2. Le modèle incrémentiel de développement de logiciel est
 - a. Une approche raisonnable lorsque les exigences sont bien définies.
 - b. Une bonne approche lorsqu'un produit de base est exigé rapidement.
 - c. La meilleure approche à utiliser pour les projets avec de grandes équipes de développement.
 - d. Un modèle révolutionnaire qui n'est pas utilisé pour les produits commerciaux.

3. Modèles de processus évolutifs
 - a. Ils sont de nature itérative.
 - b. Peut facilement répondre aux changements des exigences du produit.
 - c. Ne produisent généralement pas de systèmes jetables.
 - d. Tout ce qui précède.

4. Le modèle de prototypage de développement des logiciels est
 - a. Une approche raisonnable lorsque les exigences sont bien définies.
 - b. Une approche utile lorsqu'un client ne peut pas définir clairement les exigences.
 - c. La meilleure approche à utiliser pour les projets avec de grandes équipes de développement.
 - d. Un modèle risqué qui produit rarement un produit significatif.

5. Le modèle en spirale de développement de logiciels
 - a. Se termine par la livraison du produit logiciel.
 - b. Est plus chaotique que le modèle incrémental.
 - c. Comprend l'évaluation des risques du projet au cours de chaque itération.

6. Le modèle de développement concurrent est
 - a. Un autre nom pour l'ingénierie concurrente.
 - b. Définit les événements qui déclenchent les transitions d'état de l'activité d'ingénierie.
 - c. Utilisé uniquement pour le développement de systèmes parallèles ou distribués.
 - d. Utilisé chaque fois qu'un grand nombre de demandes de modification sont anticipées.
 - e. a et b

7. Le modèle de développement à base de composants est
 - a. Seulement approprié pour la conception du matériel informatique.
 - b. N'est pas capable de supporter le développement de composants réutilisables.
 - c. Dépendant de l'approche orientée objet.

- d. N'est pas rentable selon les mesures logicielles quantifiables connues.
8. Le modèle des méthodes formelles de développement de logiciels utilise des méthodes mathématiques pour
- Définir la spécification des systèmes informatiques.
 - Développer des systèmes informatiques sans défaut.
 - Vérifier l'exactitude des systèmes informatiques.
 - Tout ce qui précède.
9. Laquelle parmi les suivantes n'est pas une phase du modèle RUP (Rational Unified Process)?
- Phase de création
 - Phase d'élaboration
 - Phase de construction
 - Phase de validation
10. Lequel de ces éléments n'est pas une caractéristique du Processus Logiciel Personnel? (PSP : Personal Software Process)?
- Met l'accent sur la mesure personnelle du produit de travail.
 - Le praticien exige une supervision minutieuse par le chef de projet.
 - Le praticien individuel est responsable de l'estimation et de la planification.
 - Le praticien a l'habileté de contrôler la qualité des produits logiciels.
11. Quel est l'objectif du Processus Logiciel d'Equipe (TSP : Team Software Process)?
- Accélérer l'amélioration des processus logiciels
 - Permettre une meilleure gestion de temps par des professionnels hautement qualifiés
 - Créer des équipes de logiciels auto-dirigés
 - Montrer aux gestionnaires comment réduire les coûts et maintenir la qualité
 - b et c
12. Les outils technologiques des processus permettent aux entreprises de logiciels de compresser les plannings en ignorant les activités sans importance.
- Vrai
 - Faux
13. Il est généralement admis que l'on ne peut pas avoir de processus logiciels faibles et créer des produits finis de haute qualité.
- Vrai
 - Faux

Questions de recherche :

- Donner les raisons de votre réponse en fonction du type de système en cours de développement, proposer le modèle de processus logiciel générique le plus appropriée qui pourrait être utilisé comme une base pour la gestion de développement des systèmes suivants :

- ♣ Un système pour contrôler le freinage anti-blocage (Anti-lock Braking System) dans une voiture.
 - ♣ Un système de réalité virtuelle pour soutenir la maintenance des logiciels.
 - ♣ Un système de comptabilité de l'université qui remplace un système existant.
 - ♣ Un système interactif de planification de voyage qui aide les utilisateurs à planifier leurs voyages avec le plus bas impact sur l'environnement.
2. Expliquer pourquoi le développement incrémental est l'approche la plus efficace pour le développement des systèmes logiciels de commerce. Pourquoi ce modèle est moins approprié à l'ingénierie des systèmes de temps réel ?
 3. Considérons le modèle de processus basé sur l'intégration et la configuration montré dans Chapitre 2. Expliquer pourquoi il est essentiel d'avoir deux activités séparées des exigences dans le processus.
 4. Proposer pourquoi il est important de faire une distinction entre le développement des exigences de l'utilisateur et le développement des exigences du système dans le processus de l'ingénierie des exigences.
 5. Décrire les principales activités du processus de conception de logiciels et les sorties (résultats) de ces activités. En utilisant un diagramme, montrer les relations possibles entre les sorties de ces activités.
 6. Expliquer pourquoi le changement est inévitable dans les systèmes complexes et donner des exemples (en dehors de prototypage et la livraison incrémentielle) des activités de processus logiciel qui aident à prédire les changements et rendre le logiciel étant développé plus résistant aux changements.
 7. Expliquer pourquoi les systèmes développés comme prototypes ne devraient, normalement, pas être utilisés comme une production de systèmes.
 8. Expliquer pourquoi le modèle en spirale de Boehm est un modèle adaptable qui peut soutenir à la fois les activités de l'évitement du changement et de la tolérance au changement. Dans la pratique, ce modèle n'a pas été largement utilisé. Proposer pourquoi cela pourrait être le cas.