**EVALUATION GENIE LOGICIEL**

**Exercice1 :**

1.**LES METHODES AGILES** sont caractérisées par la flexibilé,

\_Une méthode agile: Scrum.

2.La principale responsabilité d'un testeur dans un environment agile est d'assurer la qualité du produit.

3.Un repository dans Git c'est un espace de travail ou est stocké tout les modifications de notre projet.

Pour ajouter un fichier qui se nomme fichier1 à notre branche master on place le fichier dans le dossier du projet et on exécute les commandes suivantes :

git add fichier1

git commit -m ajout de fichier1

git checkout master

git merge

git push.

4.Les avantages d'utiliser Git une forte collaboration,l'historique des modifications, la securite ainsi que l'efficacité

**Exercice2 :**

Le Manifeste Agile comporte 4 valeurs. Faites correspondre la valeur agile à gauche (1 – 4) avec son homologue traditionnel à droite (i – iv).

1) Collaboration client terminée -- i) Processus et outils

2) Répondre au changement --iii) Négociation du contrat

3) Individus et interactions au cours --ii) Suivre un plan

4) Logiciel fonctionnel terminé -- iv) Documentation complète

**Exercice3 :**

Parmi les affirmations suivantes, lesquelles sont vraies sur les projets agiles ?

* Les testeurs doivent travailler en étroite collaboration avec les développeurs tout en conservant une perspective objective ***: VRAI***
* Les gestionnaires de tests n'existent pas dans les organisations effectuant du développement agile ***: FAUX***
* Il n'y a aucune différence entre ce que font les testeurs et les développeurs sur les projets agiles *:* ***FAUX***
* Les développeurs doivent s'appuyer sur des testeurs pour créer les tests de régression automatisés :***FAUX***

. Une sélection d'utilisateurs peut effectuer des tests bêta sur le produit après l'achèvement d'un série d'itération ***: VRAI***

**Exercice4 :**

Les activités qui representent le mieux les responsabilités qui sont coherentes avec l'approche d'équipe entiére du développement :

* Les testeurs doivent travailler avec les représentants des clients pour créer des tests d'acceptation
* Toute l'équipe, et pas seulement les testeurs, est responsable de la qualité du produit.
* Les développeurs doivent tester les exigences non fonctionnelles (performances, convivialité, sécurité, etc.)

**Exercice5:**

**1.**Comparerons la relation d’agrégation et la relation de composition:

La relation d'agregation est une relation faible ou un objet peut exister independamment de l'objet auquel il est associé.

la relation de composition c'est une relation forte ou un objet depend completement de l'objet contenant pour exister.

**2.Comparaison entre les modèles de cycle de vie :**

Prototypage non jetable c'est un prototype évolue progressivement en produit final, sans être jeté.

modèle de cycle de vie c'est un prototype évolue progressivement en produit final, sans être jeté. Le développement se fait en plusieurs modules ou incréments successifs.

**3. Cycle de vie d'un logiciel et documents associés :**

Document **=** Phase de production

Cahier des charges

Estimation des coûts

Calendrier du projet

Plan d'assurance qualité

Conception architecturale

Conception détaillée

Spécification des modules

Code source

Plan de test

Rapport des tests

Documentation

Manuel utilisateur préliminaireManuel d’utilisation

**5 Schéma :**

Phase de définition

Phase de conception

Phase de développement

Phase de tests

Phase de maintenance.

**EXERCICE6:**

Fonctionnalités non conformes aux attentes :

Qualité manquante : Conformité aux exigences fonctionnelles.

Le logiciel n'est pas aligné avec les besoins ou les attentes des utilisateurs, indiquant des lacunes dans la phase de collecte ou d'analyse des exigences.

Difficulté d'apprentissage pour l'utilisateur :

Une interface utilisateur mal conçue ou un manque de documentation claire peut rendre le logiciel complexe à utiliser.

**Exercice7:**

**Exercice 7 :**

** Que proposez-vous comme cycle de vie de développement. Argumentez votre proposition, montrez les avantages et inconvénients de votre proposition par rapport à d’autres possibles.**

**Pour le développement d'une application de jeux en ligne, je propose d'adopter une méthodologie agile avec Scrum. Cette approche est idéale car elle permet une flexibilité et une adaptabilité aux changements, essentielles dans l'industrie du jeu. Les livraisons itératives et les retours fréquents des utilisateurs facilitent les ajustements pendant le développement. De plus, la collaboration étroite entre l'équipe de développement et le client assure que les attentes soient respectées.**

**Cependant, elle nécessite la disponibilité des parties prenantes et peut être complexe pour une équipe novice. En comparaison, le cycle en V ou la méthode Waterfall sont plus rigides et moins adaptés à des projets évolutifs comme celui-ci, car ils limitent les retours d'expérience jusqu'à la fin du développement.**

**En conclusion, Scrum est le choix le plus approprié pour garantir un produit final qui répond aux besoins du client.**

**Exercice8 :**

La productivité du programmeur en LOC

1. LOC créées :

Projet 1 : 7200 LOC.

Projet 2 : 180 LOC.

Total : 7380 LOC.

2. Temps de travail :

Matin (08 :30 à 12 :00) : 3,5 heures.

Après-midi (12 :00 à 16 :30) : 4,5 heures.

Total par jour : 8 heures.

Durée du projet : 6 jours.

Temps total : 48 heures.

3. Productivité :

Productiviteˊ=738048≈153,75 LOC/heure

2 : Date de début du projet : Dimanche 18 janvier 2015

Calculons des indicateurs au 15/04 :

CBA : Coût budgété de l'activité = 80 unités.

CBTP : Coût budgété du travail prévu = 120 unités.

CBTE : Coût budgété du travail effectué = 80 unités.

CRTE : Coût réel du travail effectué = 25 unités.

VA : Valeur acquise = 80 unités.

IPT : Indice de performance des coûts = VA / CRTE = 80 / 25 = 3,2.

VE : Variation des coûts = VA - CRTE = 80 - 25 = 55 unités.

IC : Indice de performance temporelle = VA / CBTP = 80 / 120 ≈ 0,67.

Les nouveaux calculs des indicateurs après achèvement de la tâche 3 avec 1400 jours/hommes de travail au 01/07 :

CBA : Coût budgété de l'activité = 120 unités (le total reste inchangé).

CBTP : Coût budgété du travail prévu = 120 unités (le total reste inchangé).

CBTE : Coût budgété du travail effectué = 120 unités (toutes les tâches sont désormais achevées).

CRTE : Coût réel du travail effectué = 25 + 1400 = 1425 unités.

VA : Valeur acquise = 120 unités (le projet est terminé).

IPT : Indice de performance des coûts = VA / CRTE = 120 / 1425 ≈ 0,084.

VE : Variation des coûts = VA - CRTE = 120 - 1425 = -1305 unités.

IC : Indice de performance temporelle = VA / CBTP = 120 / 120 = 1