**EVALUATION GENIE LOGICIELLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **UEA** : | **Analyse et Conception** | **Semestres** : | 3 |
| **ECUEA** : | G-L | **Durée** : | 3 H |
| **Prénoms :** |  | | |
| **NOM :** |  | **Filière :** | DBE |

**NB :** Documents sont et **d’internet** autorisés, **Pas de ChatGPT**.

**Travail individuel**.

1. **Questions de Cours :**

**Exercice 1 :**

1. Qu’est-ce qui caractérise les méthodes agiles ? Citer une méthode agile.

Ils sont caracterisees par l’adaptabilite aux changements des besoins clients,l’auto organisation des equipes et la colaboration entre l’equipe et le client.

Exemple :la methode scrum

1. Dans un environnement Agile, quelle est la principale responsabilité d’un testeur ?

Garantir la qualite logicielle

Automatiser les tests

Colaborer avec les developpeurs

1. Qu'est-ce qu'un « repository » dans GIT ?

C’est un espace d’stockage qui contient le travail complet des modifications d’un projet comme code ou fichiers …

4-Quelles sont les étapes pour effectuer un ajout d'un fichier « fichier1 » à la branche master ?

Les etapes sont :

-ajouter le fichier a l’index : git add fichier1

-commiter les changements : git commit –m « ajout de fichier1 »

-pousser vers la branche master si le depot distant existe : git push origine master

1. Quels sont les avantages d'utiliser GIT ?

Sauvegarder nos projets, collaboration c’est-à-dire plusieurs developpeurs peuvent travailler en parallele.

**Exercice 2 :**

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 iii i) Processus et outils

2) Répondre au changement ii ii) Suivre un plan

3) Individus et interactions au cours i iii) Négociation du contrat

4) Logiciel fonctionnel terminé iv iv) Documentation complète

**Exercice 3 :**

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

1. Les testeurs doivent travailler en étroite collaboration avec les développeurs tout en conservant une perspective objective vrai
2. Les gestionnaires de tests n'existent pas dans les organisations effectuant du développement agile faux
3. Il n'y a aucune différence entre ce que font les testeurs et les développeurs sur les projets agiles faux
4. Les développeurs doivent s'appuyer sur des testeurs pour créer les tests de régression automatisés faux
5. Une sélection d'utilisateurs peut effectuer des tests bêta sur le produit après l'achèvement d'un série d'itérations vrai

**Exercice 4 :**

Quelles activités ci-dessous représentent le mieux les responsabilités qui sont cohérentes avec l’approche d’équipe entière du développement agile ?

1. Les testeurs sont responsables du développement des tests unitaires qu'ils transmettent aux développeurs pour essai.
2. Les représentants des entreprises doivent sélectionner les outils que l'équipe utilisera pendant la projet
3. Les testeurs doivent travailler avec les représentants des clients pour créer des tests d'acceptation
4. Toute l'équipe, et pas seulement les testeurs, est responsable de la qualité du produit.
5. Les développeurs doivent tester les exigences non fonctionnelles (performances, convivialité, sécurité, etc.)

**Exercice 5 :**

1. Comparer la relation d’agrégation et la relation de composition.

Difference cruciale en UML pour la modelisation des relations entre objets.

1. Comparer le modèle de cycle de vie par prototypage non jetable au modèle par incréments ?

Le premier est exploratoire tandis que le second est structure pour une livraison progressive.

1. En considérant le cycle de vie d’un logiciel
   1. Indiquer la ou les phases où est produit chacun des documents suivants : Manuel d’utilisation, conception architecturale, plan d’assurance qualité, spécification des modules, code source, cahier de charges, plan de test, manuel utilisateur préliminaire, conception détaillée, estimation des couts, calendrier du projet, rapport des tests, documentation

- Cahier de charges → Analyse des besoins.

- Conception architecturale → Conception globale.

- Spécification des modules → Conception détaillée.

- Plan d’assurance qualité→ Planification (début de projet).

- Code source → Implémentation.

- Plan de test → Conception détaillée/Implémentation.

- Manuel utilisateur préliminaire → Implémentation (tests alpha).

- Manuel d’utilisation final → Validation/Maintenance.

- Rapport des tests → Validation.

- Documentation technique→Conception/Implémentation.

- Estimation des coûts/Calendrier → Planification.

* 1. Comment peut‐on combiner le modèle en cascade ou en V avec le modèle en spirale ? par un petit schéma montrer de quelle manière on peut l’utiliser, en illustrant l’enchaînement des différentes étapes.

En v les documents sont produits a des etapes claires et Spirale ajoute une gestion proactive des risques.

**Exercice 6 :**

L'utilisateur d'un logiciel signale les problèmes suivants :

* Le logiciel ne fournit pas toutes les fonctionnalités attendues.
* L'utilisateur trouve que l'apprentissage pour utiliser le logiciel est difficile.
* Les résultats donnés par le logiciel sont parfois erronés.
* Le logiciel consomme beaucoup de CPU pour des requêtes qui semblent simples.

Question : Quelles sont les qualités manquantes dans ce logiciel ?

Le logiciel manque de :

-fonctionnalite

-utilisabilite

-performance

-exactitude

**Exercice 7 :**

Une entreprise de génie logiciel souhaite réaliser une petit application de jeux sur Internet, cette demande est inhabituelle pour cette société. Le développement de ce projet ne comporte pas de risques techniques. Un cahier des charges précis est donné par le client.

* 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.

Mon proposition est :le modele en casade (cycle en v)

Les avantages sont : la documentation rigoureuse , l’adaptation aux besoins clairs et stable…

les inconvénients sont : livraison tartive du jeu , les risques technique…

Exercice 8 :

Une université veut s’équiper d’un système de gestion des étudiants qui prendrait en compte tous les détails concernant les étudiants, y compris les informations personnelles, les cours suivis et les notes obtenues aux examens. Les trois approches possibles sont :

1. Acheter un système de gestion de bases de données et développer son propre système basé sur cet outil.

Erreurs de conception et maintenance complexe

Approche : developper avec un (SGBD)

1. Acheter un système comparable à celui d’une autre université et le modifier pour ses propres besoins.

Erreurs de compatibilite , la dependance au fournisseur et les fonctionnalites inutiles .

Approche : addaptation d’un logiciel existant

1. Se joindre à un groupe d’autres universités, établir un cahier des charges commun, contacter une société de logiciels qui développera un seul système pour tous.

Problemes de negociations longues et coordination complexe.

Approche :consortium avec d’autres universites.

Identifiez les risques possibles pour chacune de ces stratégies et décider quelle approche adopter.

**Exercice 8.**

A partir des données suivantes, calculez la productivité du programmeur en LOC/j.

On suppose en outre que :

- Le projet 1 contenait 7200 LOC/j et le projet 2 en contenait 180 ;

- Une journée normale de travail compte 7 heures

- Les réunions ne sont pas comptabilisées pour le calcul de la productivité

Productivit( LOC/jour)=total des LOC/nombre de jour effectifs

- pour le projet 1 :

Temps=42h=6jours

Productiv=7200LOC/6jours=1200LOC/jours **productivite=1200LOC/jours**

**-**pour le projer 2 :

**Productivite=180 LOC/joiurs**

**-**productivite moyenne globale :

Moyenne = total LOC/total jours

=7200+180/7

=1054,29



**Exercice 2.**

Le tableau suivant récapitule l'avancement du projet géré à la date du 15/04.

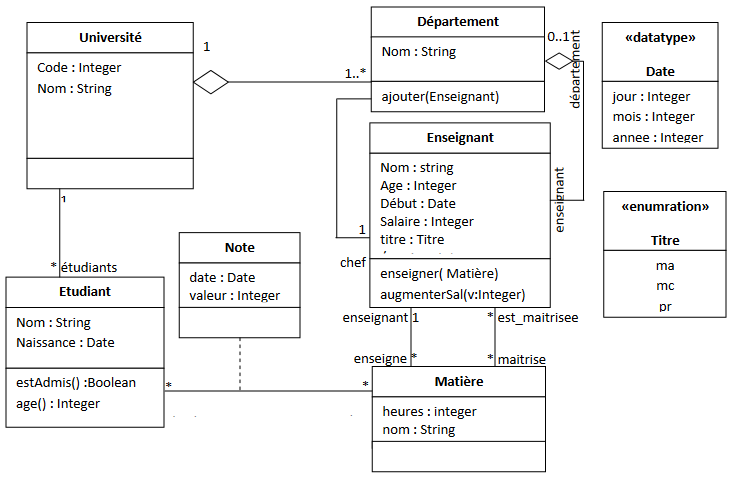


1. Quelle est la date de début du projet (nous supposons que la semaine commence dimanche et fini le jeudi soir) et le calcul se fait sur l’année 2015.

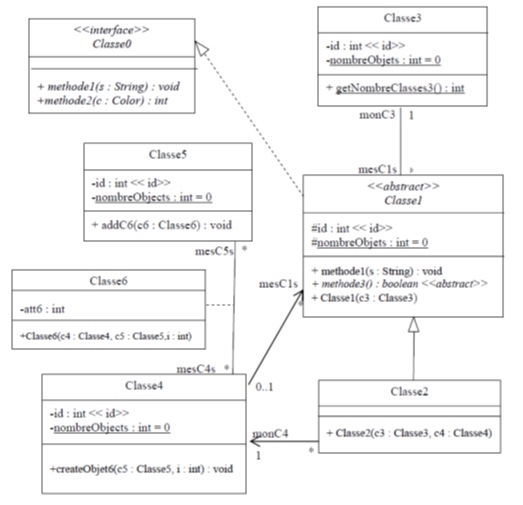
Date de debut = 15/04/2015- 12

Semaines =22/01/2015

1. Calculer les indicateurs d'avancement au 15/04 (CBA, CBTP, CBTE, CRTE, VA, IPT, VE, IC et VC).
2. Que deviennent ces indicateurs, à supposer que la tâche 3 a également été achevée avec 1400 jh de travail et que nous sommes le 01/07 ?
3. **TRONSFORMER LES MODELES UML SUIVANTS VERS JAVA, JS, PHP ET SQL.**

****

**2.**

****