Rédigé le :
29/09/2023

Année universitaire
2023/2024

Université Paris Cité

Compte-rendu de TD semaine n°2

Semaine n°1
Date : 29/09/2023
Rédacteur de ce compte-rendu : Camelia MAZOUZ

Partie 1

Question 2 :

Réalisé avec **:** Lynda Neggaz **,**Lina Youssef

Durant son cycle de vie, un logiciel est soumis à des **exigences** et des contraintes qu'on appelle 'Software requirements' sur lesquelles il est évalué et validé. Ces exigences logicielles expriment les besoins et les contraintes imposées à un produit logiciel qui contribuent à la résolution d'un problème du monde réel. Chaque **exigence logicielle** fournie par le processus d'exigences, doit être définie de manière très précise afin de garantir une bonne conception et maintenance du logiciel.

Les exigences fonctionnelles décrivent les fonctions que le logiciel doit réaliser, tandis que les **exigences non fonctionnelles** aussi appelées des exigences de qualité, sont celles qui contraignent la solution.

Les propriétés émergentes dépendent de l'architecture du système, en effet elles dépendent de l'interaction de tous les composants du système.

Les exigences logicielles doivent être définies avec clarté et sans ambiguïté.

Les exigences système englobent l'ensemble des composants d'un système (hardware, software, firmware), tandis que les exigences logicielles sont dérivées des exigences système.

Question 3 :

Dès la première semaine de l'UE projet on a été charge de rédiger le cahier des charges, document regroupant toutes les exigences imposées à notre logiciel.

En ce qui concerne les exigences logicielles et les contraintes, on aurait dû mieux les étudier et les décrire pour qu'a la phase de développement il n'y ai pas d'ambiguïté entre les membres du groupe et qu'on ait tous la même vision des fonctionnalités a implémenter.

Partie 2 :

Réalisé avec **: Alain Martin-Savornin**

Langages choisis : C, Java

Liste de contrôle du langage C : (réalisé par Camelia MAZOUZ)

- Ne pas oublier les points virgules à la fin de chaque ligne.
- Vérifier que toute parenthèse ouvrante a été fermé.
- Vérifier que tout crocher ouvrant a été fermer.
- Vérifier que le type de retour d'une fonction correspond au type de la définition de la fonction.
- Vérifier que toute allocation de mémoire a bien été désallouer à la fin du programme.
- Vérifier que tout fichier ouvert a été fermé.
- Vérifier qu'il n'y ait pas de boucles infinies.
- Bien commenter son code.
- Ecrire des tests unitaires pour chaque fonction.
- Vérifier qu'il n'y ait pas de doublons de fonctions.

Alain m'a conseillé de rajouter :

- Vérification qu'on a importé les packages nécessaires.
- Bien indenter son code.

Liste de contrôle du langage Java : (réalisé par Alain Martin-Savornin)

- Vérifier que l'indentation est correcte.
- Mettre en place des blocs try et catch pour gérer les erreurs.
- Pour la POO, bien définir les classes et répartir les fonctions dans ces différentes classes.
- Bien écrire les contructeurs.
- Le Super pour les classes filles.
- StringBuffer pour économiser de la mémoire.
- vider le ByteBuffer quand on lit des fichiers.
- éviter les boucles infinies.
- vérifier que l'on a bien a fait les imports nécessaires.

Note : Les définitions des termes présents dans le SWEBOOK ont été traduite en français.