

## Code d'honneur

En répondant à cet examen à domicile ("take-home") de LOG3000, je m'engage à :

- avoir complété cet examen par moi-même, sans l'aide d'autres personnes ;
- ne remettre que mes propres travaux, qui ne contiennent aucun matériel plagié ;
- ne divulguer pas les questions posées dans cet examen et les réponses aux questions à quiconque ne suit pas LOG3000 cette session ;
- ne pas participer à des activités malhonnêtes visant à améliorer mes résultats ou à nuire aux résultats d'autres personnes.



---

Signature :

Date et lieu : 25 février 2022

## **1 Question sur l'utilité des processus (1 point)**

La communication interne et externe est une utilité du processus, car à l'interne, il permet aux employés qui travaillent sur le projet de bien comprendre les étapes de développement d'un logiciel. Ils peuvent facilement connaître le but des activités et l'information disponible. De plus, à l'externe, le processus peut être évalué selon des standards et avoir des certifications comme la CMMI ou le ISO15504. Le client peut également demander le processus.

L'évaluation des pratiques pour un projet est également une utilité du processus. On peut faire des diagnostics pour découvrir les problèmes causés par les méthodes actuelles pour le développement logiciel. Par la suite, on peut retirer, remplacer ou créer des méthodes pour améliorer le processus et optimiser le développement logiciel.

## **2 Question sur le choix d'un cycle de vie (4 points)**

- a) Le modèle spirale, car pour qu'une voiture soit complètement autonome, il va avoir plusieurs risques importants. Le modèle permet de faire une analyse des risques très élaborée pour prioriser l'ordre des cycles ce qui permet de mieux gérer les risques.
- b) Le modèle transformationnel, car on doit tester la performance du logiciel, donc il est préférable d'avoir les résultats directs des utilisateurs et obtenir les commentaires. Par la suite on peut décider d'améliorer l'algorithme utilisé ou la rejeter.
- c) Le modèle incrémental, car comme l'application va permettre la communication entre différents appareils électroniques, on peut fractionner le projet en itérations qui se concentrent sur un appareil et une dernière itération pour faire communiquer les appareils ensemble.
- d) Le modèle cascade, car on a déjà toutes les exigences au début du projet et il y a très peu d'incertitudes.

## **3 Question sur la gestion du risque (4 points)**

- a) R1 : Risque indirect, l'équipe n'a aucun contrôle sur les défauts de fonctionnement des librairies.

R2 : Risque direct, c'est l'équipe qui a conçu l'algorithme, donc ils ont un certain contrôle sur les performances.

R3 : Risque direct, c'est en suivant un processus créé par l'équipe, donc ils ont un certain contrôle sur le code, la documentation qui affecte la maintenabilité.

- b) R1 : L'équipe peut éviter le risque. Si au début du projet l'équipe sait déjà que la librairie peut ne pas être fonctionnelle pour la reconnaissance de défauts de fabrication dans un cadre industriel, ils peuvent simplement utiliser une librairie moins connue qui permet d'obtenir des meilleurs résultats dans ce domaine.

R2 : L'équipe peut contourner le risque. Ils peuvent concevoir plusieurs algorithmes dans la phase de conception pour pouvoir changer d'algorithme rapidement si celui choisi pour l'implémentation n'a pas des bonnes performances.

R3 : L'équipe peut mitiger le risque. Au début, ils peuvent utiliser un processus avec des activités de vérification plus robustes à la fin des activités d'implémentations pour s'assurer que l'implémentation respecte la conception prévue. L'équipe peut également faire des suivis à la fin des activités pour mieux documenter le travail.

#### **4 Question sur la discipline de gestion de projet (3 points)**

- a) En tant que gestionnaire de projet, je me situerais très près du sommet de la gauche du « Iron triangle » adapté au génie logiciel. Très proche du sommet de « quality », car c'est un logiciel qui permet de produire des impulsions électroniques selon les données d'électrodes, il y a plusieurs risques importants. On veut obtenir un logiciel de très bonne qualité avec une très grande précision.
- b) Le modèle spirale est celui le plus compatible avec ma position dans le triangle.
- c) Comme le logiciel demandé va avoir plusieurs risques donc avec le modèle on va mieux gérer le risque et en réduisant les risques on va pouvoir développer un logiciel avec une meilleure qualité.

#### **5 Question sur la qualité des exigences (3 points)**

##### **Partie A**

- a) Exigence fonctionnelle, elle décrit que le système permet aux responsables de la sécurité de faire.
- b) Exigence fonctionnelle, elle décrit une caractéristique du format de message envoyé aux utilisateurs.

## **Partie B**

- a) Problèmes : L'exigence est non-testable  
Correction : Le système doit être fonctionnel 99% du temps.
- b) Problème : L'exigence est composée, utilisation du pluriel  
Correction :
  - Le système doit permettre la distribution des alertes sur plusieurs plates-formes.
  - Le système doit permettre à un utilisateur avec un rôle d'autorité de contrôler la distribution des alertes.

## **6 Question sur l'analyse qualitative de processus (4 points)**

- a)
  - Le code source peut avoir beaucoup d'incohérence avec les exigences demandées.
  - Le code peut être très mal structuré.
  - Les tests peuvent être incompatibles avec le code source.
- b) Dans l'activité de remues méninges, on peut faire participer l'implémenteur pour qu'il discute avec l'analyste sur ce qui est faisable dans les exigences et ce qui n'est pas réaliste et ajouter une activité de validation avant l'activité d'intégration fait par l'analyste et l'implémenteur.

Avant la phase d'implémentation, il est préférable d'inclure une phase de conception avec des activités de conception comme la conception des composants fait par le designer et des activités de vérifications comme la vérification de la solution de conception fait par l'analyste et le designer.

Faire une activité de suivi pour le code source dans la phase d'implémentation par l'implémenteur et le testeur pour permettre au testeur de se familiariser avec l'implémentation et d'utiliser le suivi pour la rédaction des cas de tests pour que les tests soient plus compatibles avec le code source.

## 7 Question sur l'analyse quantitative de processus (5 points)

- a) Le cycle de vie utilisé ressemble au modèle incrémental avec des petites modifications. On peut voir que pour chaque semaine une grande partie du temps est dédiée à une seule étape comme les exigences, l'analyse et la conception, l'implémentation ou les tests qui concerne une fraction d'exigences et une itération dure 4 semaines avant que le cycle recommence.
- b)
- Plus on avance dans le projet, moins on fait de tests et à la fin, on va avoir beaucoup de difficultés pour identifier les problèmes potentiels durant la maintenance.
  - On néglige de plus en plus l'analyse et la conception, il est préférable qu'on le fasse le long du projet pour vérifier si l'implémentation a bien été fait et d'adapter les documents pour faciliter la maintenance.
- c) Je vais m'assurer de demander aux équipes de développement de toujours fournir un suivi des tests pour l'équipe de maintenance à la fin de chaque étape de test pour connaître les problèmes rencontrés du logiciel. De plus, je vais les demander de toujours réserver du temps pendant pour l'analyse et la conception pour s'assurer que les documents de conception sont à jour.

## 8 Question sur la théorie de la mesure (5 points)

- a) Question 1 : Es ce que les membres collaborent et intègrent constamment leurs travaux ensemble ?

Métrique : le nombre de « merges » fait pendant le projet.

Question 2 : Es ce que les tâches complétées par les membres sont conformes aux exigences ?

Métrique : le nombre de cas de tests qui passent.

- b) i.) échelle intervalle, on peut dire que l'heure d'achat 10h30 et 11h00 sont différentes. On peut dire que l'achat à 11h00 a été fait plus tard que 10h30. On peut dire que la différence entre 9h30 à 10h00 est la même que la différence entre 10h30 et 11h00.
- ii.) échelle ratio, on peut comparer le nombre d'erreurs d'un utilisateur avec le nombre d'erreurs d'un autre utilisateur et on peut calculer une proportion.

iii.) échelle ordinale, la satisfaction de l'utilisateur peut être comparé. On peut voir si un utilisateur est plus satisfait du système qu'un autre. On ne peut pas appliquer des opérations mathématiques élémentaires.

iv.) échelle nominale, on peut classer selon chaque type d'activité. Les types sont mutuellement exclusives et on ne peut pas comparer entre 2 types.

## **9 Réflexion (1 point)**

J'ai trouvé que la modélisation d'un processus selon le contexte avant le développement d'un projet est utile, car ça montre comment planifier et structurer le développement d'un logiciel à un très haut niveau sans connaître les limitations. J'ai également trouvé que la présentation des différents modèles de cycle de vie utile, car ça m'a aidé à comprendre pourquoi les compagnies utilisent différents modèles pour s'adapter à leur situation.

J'ai trouvé qu'il y a manqué d'un exemple concret pour mieux expliquer le modèle spirale. Je pense qu'il serait mieux de trouver un exemple de contexte où on utilise le modèle spirale pour gérer les risques d'un projet dans le domaine.