

ORDRE DES INGÉNIEURS DU QUÉBEC

SESSION DE NOVEMBRE 2021

Toute documentation permise
Calculatrices : modèles autorisés seulement
Durée de l'examen : 3 heures

19-LO-A6 Assurance-qualité logicielle

Cet examen comporte cinq (5) questions. 40 points.

I. Choix d'activités d'AQL (4 x 2,5 = 10 points)

L'entreprise Baguette & Co. est responsable d'un grand projet dont le processus consiste en des étapes de l'analyse des requis, la conception (design), le codage et l'intégration des changements des développeurs. Chaque étape a une probabilité de 25% d'introduire un défaut. Depuis des années, Baguette & Co. utilise des revues techniques des requis (efficacité de 50%, coût pour enlever des défauts de 1), des tests unitaires (50%, 10) et des tests d'intégration (50%, 20). Finalement, le coût pour enlever des défauts en production, c'est-à-dire après que les utilisateurs auront installé le système, est de 40.

Il est à noter que l'efficacité d'une activité d'AQL est définie comme le pourcentage de défauts qui peuvent être détectés par les gens responsables de cette activité. Le coût pour les éliminer est spécifié en termes d'une unité imaginaire.

(1) Quel pourcentage des défauts sera observé par les clients et quel est le coût pour enlever ces défauts?

(2) Quel est le coût total pour réaliser l'approche d'AQL de Baguette & Co. (sans inclure le coût pour enlever les défauts du côté client de la question précédente)?

Après avoir suivi des cours d'AQL, Baguette & Co. décide d'ajouter une des deux approches suivantes aux activités existantes : inspection formelle de la conception (60%, 2.5) ou bêta-tests (60%, 25).

(3) À base d'une simulation avec les numéros ci-dessus, laquelle des deux approches est la meilleure en termes d'efficacité et de coût? Expliquez aussi pourquoi ces résultats numériques font du sens.

(4) Comparez aussi le coût des deux approches avec celui de l'approche originale? Pourquoi est-ce que l'on obtient un tel résultat?

II. Exigences de qualité (4 + 3 + 3 x 1 = 10 points)

L'entreprise Tesla est une des pionnières en voitures autonomes (self-driving cars). Leur environnement logiciel FSD (Full Self Driving) est développé de façon incrémentale, avec de nouvelles fonctionnalités et résolutions de défauts installées par connexion sans-fil. Des fonctionnalités prévues incluent : le changement automatique de voie, le stationnement automatique, ou même la convocation automatique de la voiture garée (c'est à dire la voiture trouvera son propre chemin du stationnement vers son propriétaire).

Élaborez de façon détaillée et concrète, dans le contexte spécifique de l'environnement logiciel FSD de Tesla, aux questions suivantes :

(1) En utilisant le modèle de qualité de Deutsch et Willis, identifiez et motivez les cinq (5) facteurs de qualité les plus essentiels dans le cas de FSD.

(2) En focalisant sur ces cinq (5) facteurs, et tenant compte du processus incrémental de développement de FSD, proposez une stratégie d'AQL pour FSD. Dans l'intérêt du temps, vous pouvez vous limiter aux aspects de choix d'activités d'AQL ainsi que les facteurs de qualité traités par les activités.

(3) Quelques questions plus détaillées, toujours dans le contexte de FSD:

- (3a) Illustrez les concepts de « faute », « défaillance » et « erreur ». L'illustration doit être un exemple spécifique, pas la définition générique.

- (3b) Donnez un exemple concret qui illustre la différence entre l'assurance qualité et le contrôle de la qualité.

- (3c) Donnez un exemple concret qui illustre la différence entre les concepts de validation et de vérification (IEEE 1012).

III. Gestion de risque et mesure (2 + 3 + 3 = 8 points)

Une organisation multinationale n'arrive plus à livrer ses produits logiciels aux clients dans les délais prévus. Une des causes suspectées est le processus de gestion des configurations, en particulier l'utilisation des branches. L'organisation n'avait pas mis en place de directives concrètes sur la profondeur des branches, et maintenant il y a des équipes qui travaillent dans une branche de la branche principale (profondeur 1), d'autres qui travaillent dans une branche d'une branche (profondeur 2), etc.

Formulez une réponse aux questions suivantes:

(1) Conceptuellement, est-ce que la profondeur des branches pourrait impacter l'AQL? Motivez votre réponse.

(2) Afin de formuler des directives pour améliorer la situation, l'organisation prévoit faire des analyses des données de développement afin de mesurer l'impact de la profondeur des branches sur l'AQL. Proposez un programme de mesures à ce point, en focalisant sur les étapes de « déterminer les buts et objectifs liés aux processus-clé », « bâtir le programme de mesures » et « décrire les systèmes d'information à mettre en place ».

(3) Sachant que les résultats du programme seront utilisés pour créer des directives pour les développeurs à travers toute l'organisation (multinationale), identifiez et évaluez les risques du programme.

IV. La déontologie de l'ingénieur logiciel (2 x 1 = 2 points)

(1) Vrai/faux (et pourquoi) : le code de déontologie est un algorithme qui génère des décisions éthiques?

(2) Choisissez une seule réponse: le code de déontologie de l'ingénieur logiciel vise à :

- a. Documenter les engagements moraux et professionnels des ingénieurs logiciels.
- b. Forcer les individus à respecter des standards de qualité.
- c. Documenter les bonnes pratiques des ingénieurs logiciels.

V. Les tests (5 x 2 = 10 points)

« NextDate » est une fonction avec trois variables : mois, jour, année. Elle retourne la date du jour après la date donnée :

- Exemple: « NextDate » (octobre, 1, 2021) retourne <octobre, 2, 2021>
- Années valides : 1818-2018.
- Doit gérer les mois à 30 et 31 jours, de même que le mois de février.
- Doit gérer les années bissextiles, tenant compte des exceptions :
 - 1900 est divisible par quatre, mais les débuts de siècle n'ont pas de 29 février.
 - 2000 est un début de siècle, mais les débuts de millénaire ont un 29 février.

Formulez une réponse aux questions suivantes :

- 1) Utilisez les approches de classes d'équivalence et de valeurs frontalières afin de générer des cas de test pour la fonction « NextDate ».
- 2) Est-ce que ces deux approches de test pourraient suffire afin de complètement tester la fonction « NextDate »? Donnez au moins trois arguments pour motiver votre réponse.
- 3) Vrai/faux (et pourquoi) : des tests unitaires par définition sont des tests en boîte noire.
- 4) Est-ce que la notion de traçabilité joue un rôle important pour des tests? Motivez votre réponse.
- 5) Est-ce que des tests peuvent viser des requis non-fonctionnels? Motivez votre réponse, incluant un exemple concret dans le contexte d'un logiciel utilisant la fonction « NextDate ».