

# QCM \_ Test\_2023\_ section 1

Lire les questions attentivement et répondre soigneusement.

Adresse e-mail \*

amine.ayadi@enis.tn

- ✓ Soit le test suivant, pour que ce test génère un pass il suffit que la liste des tâches contient une seule tâche \*

```
it('lists tasks', async () => {  
  const taskCount = await this.todoList.taskCount()  
  const task = await this.todoList.tasks(taskCount)  
  assert.equal(task.id.toNumber(), taskCount.toNumber())  
  assert.equal(task.content, 'Model Dapp')  
  assert.equal(task.completed, false)  
  assert.equal(taskCount.toNumber(), 1)  
})
```

- ☐ Vrai
- ☒ Faux



- ✓ Laquelle des qualités suivantes est **PLUS susceptible** d'être présente chez un testeur que chez un développeur ? \*

- ☐ Une expérience sur laquelle baser leur travail
- ☒ La capacité à voir ce qui pourrait mal fonctionner
- ☐ Une bonne communication avec les membres de l'équipe
- ☐ L'attention aux détails



✓ Quelle réponse **compare LE MIEUX** les objectifs des tests de confirmation et \* des tests de régression ?

- ☐ L'objectif des tests de régression est de s'assurer que tous les tests précédemment exécutés fonctionnent toujours correctement, alors que l'objectif des tests de confirmation est de s'assurer que toute correction faite sur une partie du système n'a pas affecté négativement d'autres parties
- ☒ L'objectif des tests de confirmation est de vérifier qu'un défaut trouvé précédemment a été corrigé, alors que l'objectif des tests de régression est de s'assurer que d'autres parties du système n'ont pas été affectées négativement par la correction. ✓
- ☐ L'objectif des tests de confirmation est de confirmer que les changements sur le système ont été faits avec succès, alors que l'objectif des tests de régression est d'exécuter les tests ayant précédemment échoué pour s'assurer qu'ils fonctionnent maintenant correctement

✗ Soit le code suivant, le nombre cyclomatique associé à ce code est égale \*

```
if C1 then while (C2) do X1;  
else X2;  
X3;
```

- ☐ 0
- ☐ 1
- ☒ 2
- ☐ 3

✗



✓ Lequel des énoncés suivants fournit la MEILLEURE description d'un cas de test ? \*

- ☐ Document spécifiant une séquence d'actions pour l'exécution d'un test. Aussi connu sous le nom de script de test ou script de test manuel.
- ☒ Un ensemble de valeurs d'entrée et de résultats attendus, avec des conditions préalables à l'exécution et des conditions postérieures à l'exécution, développé pour une condition de test particulière. ✓
- ☐ Élément ou événement d'un système qui pourrait être vérifié par une ou plusieurs conditions de test

✓ Lequel des problèmes suivants ne peut pas être détecté lors d'un test statique ? \*

- ☐ Une fonction contient une boucle infinie
- ☐ Une variable est initialisée mais jamais utilisée, ce qui peut provoquer une certaine confusion
- ☒ Au-delà de 50 utilisateurs simultanés, les temps de réponse sont trop élevés ✓
- ☐ Un mot de passe est stocké en clair

✓ Quel énoncé ci-dessous décrit mieux le test non fonctionnel ? \*

- ☒ Le Test non fonctionnel est le processus de tester les attributs du système, tels que l'utilisabilité, fiabilité ou de maintenabilité. ✓
- ☐ Le Test non fonctionnel est le processus de tester un système intégré pour vérifier qu'il répond aux exigences spécifiées.
- ☐ Le Test non fonctionnel vérifie la conformité d'un système aux normes de codage.
- ☐ Le Test non fonctionnel est le processus de tester sans référence à la structure interne d'un système.



✓ L'agilité s'oppose-t-elle au test ? \*

- ☐ Non. L'agilité s'oppose au test manuel mais met l'accent sur les tests automatisés.
- ☒ Non. L'agilité favorise une exécution fréquente des tests afin d'éviter les régressions d'une itération à l'autre. ✓
- ☐ Oui. Les méthodes agiles ne favorisent pas l'utilisation des tests

✓ Est-il possible de tester avant même que le développement n'ait commencé ? \*

- ☐ Oui, en effectuant des tests dynamiques sur les exigences
- ☐ Oui, en établissant des classes d'équivalences sur le cahier des charges
- ☒ Oui, en réalisant une revue statique des spécifications ✓

✓ Lequel des exemples suivants est un **exemple de défaillance** d'un système de régulation de vitesse d'une voiture ? \*

- ☐ A. Le développeur du système a oublié de renommer les variables après une opération de copier-coller.
- ☐ B. Du code inutile déclenchant une alarme en cas de marche arrière a été inclus dans le système.
- ☒ C. Le système cesse de maintenir une vitesse définie lorsque le volume de la radio est augmenté ou diminué. ✓
- ☐ D. La spécification de conception du système indique à tort des vitesses en miles/h.



✓ Qu'est ce qui décrit le mieux un objectif pour une équipe de tests ? \*

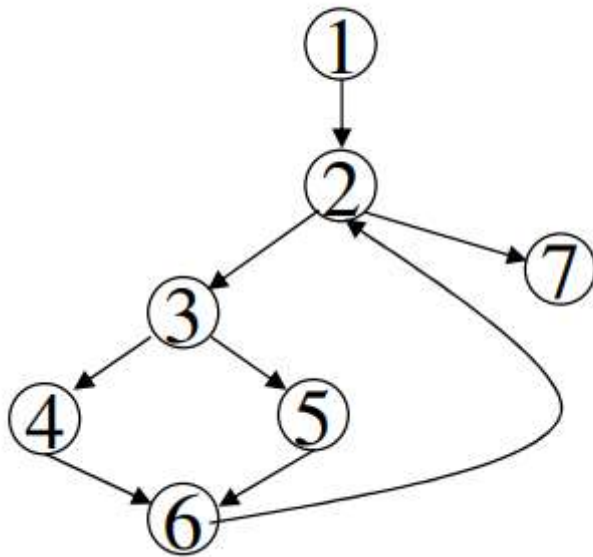
- ☐ Déterminer si suffisamment de composants ont été testés
- ☒ Causer le plus de défaillances possibles de façon à ce que les défauts soient identifiés et corrigés ✓
- ☐ Prouver que tous les défauts sont identifiés
- ☐ Prouver que tous les défauts restants ne causeront pas de défaillance

✗ En projet « en V », les tests de recette sont joués en regard de la phase de \*

- ☐ Spécification
- ☒ conception détaillée ✗
- ☐ Conception préliminaire
- ☐ codage
- ☐ expression de besoins



✓ Soit le graphe de flow de contrôle suivant, choisissez la bonne expression de \* chemins



- ☐ 123(4+5)627
- ☐ 123[(4+5)62]\*7
- ☒ 12[3(4+5)62]\*7
- ☐ 1[23(4+5)62]\*7
- ☐ aucun chemin n'est correct



✗ Lesquels des éléments suivants sont vrais \*

- ☒ Tester rigoureusement et corriger les défauts trouvés peut aider à réduire le risque de problèmes qui arrivent dans l'environnement opérationnel ✓
- ☒ Tester rigoureusement est parfois utilisé pour prouver que toutes les défaillances ont été identifiées ✗
- ☒ Les tests logiciels sont principalement nécessaires pour améliorer la qualité du système ✓



✗ Quels tests ne peuvent pas être joués en « boîte noire » ? \*

- ☐ Les tests de validation
- ☐ Les tests d'acceptation
- ☒ Les tests d'intégration
- ☐ Les tests de composants



✓ Quel type de test vont typiquement réaliser les développeurs ? \*

- ☒ Les tests de composants
- ☒ Les tests d'intégration
- ☐ Les tests système
- ☐ Les tests de recette



✓ Quand faut-il commencer à tester ? \*

- ☐ Dès que le code est suffisamment stable. Le risque, en commençant les tests trop tôt, est de confondre ce qui est en cours et ce qui contient réellement des défauts.
- ☒ Dès qu'il existe un brouillon de spécification. Des bugs peuvent déjà exister dans la documentation. ✓
- ☐ Dès le début du développement. C'est ce qu'on appelle le TDD (test driven development)
- ☐ Dès qu'une version alpha du projet est disponible. Il ne faut pas attendre la version bêta, car elle est directement déployée chez le client.



✓ Une fois la couverture de tests est égale à 100%, cela veut dire que tous les défauts sont détectés et donc le test est terminé \*

☐ Vrai

☒ Faux



✓ Le critère tous les chemins indépendants couvrent toutes les instructions mais pas toutes les décisions \*

☐ Vrai

☒ Faux



✗ Les quels des énoncés suivant concernant TDD sont corrects ? \*

☒ TDD est une approche "Test en premier" pour développer des tests automatisés réalisables ✓

☐ Le cycle de TDD est utilisé en continue jusqu'à la livraison du produit

☒ TDD permet de documenter le code pour la future maintenance ✓

☐ Le Refactoring permet le nettoyage du code pour améliorer la qualité

Ce formulaire a été créé en dehors de votre domaine. - [Conditions d'utilisation](#) - [Règles de confidentialité](#)

Google Forms











