

Le “Vibe Coding” désigne principalement :

- A. Une approche structurée basée sur des tests unitaires
- B. Une programmation imprécise guidée par des prompts vagues
- C. Une architecture d’agents IA
- D. Une méthode Agile officielle

Quel est le risque principal du “Vibe Coding” ?

- A. Une perte de performance CPU
- B. Une dette technique rapide et massive
- C. Une impossibilité de faire des démos
- D. Un code trop documenté

Un code généré rapidement par une IA, même "plausible", peut être dangereux pour la maintenabilité.

- A. Vrai
- B. Faux

Le rôle du développeur évolue vers :

- A. Tester manuellement la totalité du code
- B. Remplacer l’IA par du code artisanal
- C. Devenir un pilote/architecte d’agents IA
- D. Automatiser uniquement le déploiement

Selon KALI, ajouter des tests et vérifier la qualité augmente inutilement le temps de traitement.

- A. Vrai
- B. Faux

(Et c’est une objection à laquelle les membres répondent.)

Quelle est la réponse donnée à l'objection de KALI concernant le "temps perdu" avec les tests ?

- A. Le plus important est la vitesse brute
- B. Les tests n'ont pas d'intérêt réel
- C. La qualité coûte moins cher sur le long terme
- D. On peut les ignorer en production

CrewAI est présenté comme :

- A. Un compilateur optimisé pour l'IA
- B. Un framework permettant d'organiser plusieurs agents IA
- C. Un outil de monitoring de serveurs
- D. Un moteur de génération d'images IA

SonarQube est utilisé dans le document pour :

- A. Créer des modèles IA
- B. Automatiser le déploiement
- C. Appliquer des Quality Gates et détecter les erreurs du code généré
- D. Contenir les logs applicatifs

Dans la conclusion, KALI affirme que :

- A. La qualité du code dépend uniquement de la puissance du modèle IA
- B. La responsabilité finale repose sur le pilotage humain
- C. Les agents n'ont plus besoin de supervision
- D. Le TDD est inutile