

**Commencé le** mardi 30 mars 2021, 14:13

**État** Terminé

**Terminé le** mardi 30 mars 2021, 14:43

**Temps mis** 30 min

**Note** 15,00 sur 20,00 (75%)

Question **1**

Terminer

Note de 3,00 sur 5,00

Nommez un défi de la modularité basée sur les bibliothèques. Comment les architectures orientées services résolvent-elles ce défi?

Un des problèmes de la modularité par des librairies est qu'il y a le partage de code. Ainsi, il y a des dépendances technologiques et temporelles.

Avec les bibliothèques on permet d'avoir des frontières de modularité plus douces. Aussi, les appels en mémoire sont plus rapides et moins chers.

Commentaire :

La question est "Comment les architectures orientées services résolvent-elles ce défi?". La deuxième partie de la réponse n'est pas correcte.

Question **2**

Terminer

Note de 5,00 sur 5,00

Décrivez dans vos propres mots l'antipatron de la loi sur le drapeau rouge (Red Flag Law). (La description relative au logiciel)

Pour représenter cet antipatron, on peut penser à des personnes qui devaient marcher devant des automobiles avec des drapeaux rouges. En effet, les automobiles devaient circuler à la même vitesse que les piétons. Au niveau des projets informatiques, cela signifie que les développeurs n'utilisent pas les bonnes pratiques des microservices. En effet, tout ce qui a attiré au processus en cascade, aux tests manuels et aux commits pendant des moments de maintenance fait en sorte que la vitesse d'efficacité est très lente. Effectivement, le but des microservices et de travailler sur différentes tâches afin d'être plus efficace et de mieux séparer les tâches. Pour qu'on puisse utiliser une architecture en microservice, il faut utiliser des pratiques DevOps pour apporter un côté d'automatisation (plutôt que de faire des tests manuellement), organiser les équipes adéquatement et utiliser le modèle Strangler Application.

Commentaire :

Question **3**

Terminer

Note de 7,00 sur 10,00

Vous développez un portail de statistiques sportives. Les utilisateurs peuvent accéder aux statistiques du basketball, du hockey, du baseball et du football. Ils peuvent voir les statistiques des saisons actuelles ou passées, pour des équipes spécifiques ou comparer des joueurs. Comment implémenteriez-vous ce système? Quels services développeriez-vous? Comment organiseriez-vous vos données? Comment organisez-vous votre équipe de développement?

Pour ce système, on pourrait utiliser une architecture en microservices. En effet, nous aurions un service pour la gestion des statistiques des différents sports. Pour chaque sport on pourrait avoir une architecture en microservice. En effet, un microservice pour les statistiques de basketball, un autre pour le hockey, etc. On aurait un autre microservice pour la comparaison des joueurs. Finalement, le dernier microservice serait en lien avec les statistiques des différentes équipes pour chaque sport.

Au niveau de la gestion de l'équipe, on aurait une équipe front-end, une équipe back-end, des testeurs pour le back-end, des testeurs pour le front-end, une équipe d'assurance qualité et des intégrateurs systèmes externes. On limiterait le nombre de personnes dans chaque équipe à 12 maximum.

Au niveau des données, on utiliserait une base de données. On pourrait opter pour une base de données NoSQL. À cet effet, les documents seraient une bonne option car on aurait beaucoup de données à traiter.

Commentaire :

- Plusieurs services par sport? C'est un risque de nanoservices.
- Une seule BD pour tous les sports? Est-il efficace?

Aller à...