



Exemple d'examen

Édition 201812

Copyright © EXIN Holding B.V. 2018. All rights reserved.

EXIN® is a registered trademark.

DevOps Master™ is a registered trademark.

No part of this publication may be published, reproduced, copied or stored in a data processing system or circulated in any form by print, photo print, microfilm or any other means without written permission by EXIN.



Table des matières

Introduction	4
Exemple d'examen	5
Solutions de l'examen	26
Évaluation	66

Introduction

Voici l'exemple d'examen DevOps Master™ (DEVOPSM.FR). Les règles et réglementations d'examens EXIN s'appliquent à cet examen.

Cet examen consiste en 50 questions à choix multiples. Chaque question à choix multiple comporte un certain nombre de réponses possibles dont seulement une est correcte.

Le maximum de points qui peut être obtenu lors de l'examen est de 50. Chaque réponse correcte rapporte un point. Si vous obtenez 33 points ou plus vous réussissez votre examen.

Le temps alloué lors de l'examen est de 120 minutes.

Bonne chance !

Exemple d'examen

1 / 50

Qu'est ce qui constitue une bonne raison de mettre en œuvre DevOps dans une organisation ?

- A) DevOps présente des cycles de feedback plus fréquents pour les nouveaux services, parce que la vitesse de développement est plus élevée.
- B) DevOps améliore la continuité et l'agilité de l'entreprise, parce que les processus sont optimisés et seules des activités à valeur ajoutée sont effectuées.
- C) DevOps fournit les nouveaux services juste-à-temps (Just-in-Time), parce que les versions logicielles se mettent en production plus fréquemment.
- D) DevOps réduit le gaspillage au niveau de la collaboration entre le Développement et l'Exploitation, parce que ces deux équipes sont unifiées.

2 / 50

Le Directeur des Systèmes d'Information (CTO) estime qu'il serait très efficace d'appliquer certains concepts Lean lors de la mise en œuvre de DevOps.

Quels principes ou pratiques Lean sont les **plus efficaces** lors de l'introduction de DevOps ?

- A) **Kaizen et 5S.** Agile et DevOps étant basés sur des concepts-clés de Lean, et Kaizen et 5S étant à la base du Lean, ils seront très efficaces lors de l'introduction de DevOps.
- B) **Kaizen in advance.** DevOps requiert feedback de l'Exploitation vers le Développement. Kaizen in advance crée une boucle de feedback vers l'amont, ce qui contribue à appliquer ce principe dans DevOps.
- C) **Système Obeya.** DevOps intègre des processus de divers styles de gestion. Le système Obeya permet de visualiser l'ensemble du processus, favorisant ainsi une introduction réussie de DevOps.
- D) **Flux pièce par pièce (one-piece-flow) et JKK.** DevOps tire parti de la mise en place de processus en amont et d'un flux à valeur unique. Le flux pièce par pièce rend cela possible et JKK contribue à la rationalisation et à la mise en œuvre du flux.

3 / 50

En comparaison avec un projet traité de manière conventionnelle, que doit-on changer à un projet pour qu'il réussisse dans DevOps ?

- A) Il faudrait construire une chaîne d'approvisionnement en services informatiques, utilisant un système de traction et un flux pièce par pièce (one-piece-flow).
- B) Les développeurs devraient rejoindre l'équipe d'Exploitation pour une maintenance rapide des services.
- C) L'Exploitation doit travailler pour l'équipe de Développement. C'est là d'où vient le nom DevOps.
- D) Des membres de l'équipe d'Exploitation devraient se joindre à l'équipe de Développement.

4 / 50

Toute équipe qui adopte DevOps peut être qualifiée de Compact.

Quel est le principe **le plus approprié** pour faire en sorte qu'une Compact fonctionne ?

- A) La collaboration dynamique
- B) La communication perpétuelle
- C) La réduction des cycles
- D) La responsabilité partagée

5 / 50

Plusieurs sources de connaissance, de normes et de pratiques sont disponibles pour mettre en œuvre DevOps.

Qu'est-ce qui est considéré comme étant **indispensable** pour une mise en œuvre réussie de DevOps ?

- A) Niveau 3 de certification CMMI
- B) Agile discipliné (Disciplined Agile)
- C) ISO 20001
- D) PMBOK Guide du PMI

6 / 50

La collaboration est l'un des quatre piliers d'un DevOps efficace.

Pourquoi est-ce aussi important ?

- A) La collaboration est la seule manière d'obtenir un flux pièce par pièce (one-piece-flow) dans DevOps.
- B) La plus-value DevOps pour le business est obtenue grâce à la coopération entre les équipes.
- C) Les organisations DevOps sont petites, les équipes doivent correctement collaborer.
- D) Cela entraîne des changements et c'est un gain facile. Cela réduit également les frais de développement.

7 / 50

Vous êtes le propriétaire d'une petite entreprise DevOps comptant cinq collaborateurs qui construisent des applis mobiles pour les enfants handicapés. L'une des applis dont votre équipe est la plus fière permet, à des enfants autistes, de planifier leur journée.

Cette appli a rencontré un tel succès que l'on demande à votre équipe d'étendre son utilisation à d'autres personnes qui pourraient tirer avantage du fait de pouvoir planifier leur journée. Cette demande entraînera certainement plus de complexité au niveau des codes ainsi que quelques défis techniques qui devront être surmontés par l'équipe.

Vous vous attendez à être bien payé pour cette commande et vous l'acceptez donc. Toutefois, après quelques semaines, les membres de votre équipe se disputent. Vous vous fâchez également et vous commencez à faire plus attention à ce que les membres de l'équipe font. Vous travaillez régulièrement à leurs côtés de manière à pouvoir corriger rapidement toute erreur de code que vous remarquez.

Même si la quantité de travail abattue est suffisante, l'atmosphère dans votre équipe reste mauvaise et vous admettez qu'il manque une partie de la mentalité DevOps.

Quelle est la **meilleure** stratégie pour résoudre votre problème ?

- A) Ajouter des membres à l'équipe aspirant à plus de diversité, en recrutant du personnel non technique complémentaire
- B) Demander à des experts en DevOps externes à votre entreprise de parrainer et former les membres de votre équipe
- C) Trouver un objectif commun et commencer à œuvrer ensemble à l'atteinte de cet objectif afin d'accroître la collaboration
- D) Réserver quelques mois pour commencer à travailler à la consolidation d'équipe et pour mieux apprendre à se connaître les uns les autres

8 / 50

Vous estimez que votre équipe de Développement est réellement une équipe.

Quel signe indique avec certitude que c'est une équipe et non un groupe ?

- A) L'équipe suit les règles dont ses membres ont convenu dans leurs réunions d'équipe.
- B) L'équipe organise des réunions efficaces qu'elle dirige elle-même.
- C) L'équipe maintient un rythme de travail constant pour atteindre ses objectifs communs.
- D) L'équipe résout les problèmes en interrogeant le membre de l'équipe responsable.

9 / 50

AppAtoZ a connu un taux de croissance phénoménal au niveau du développement et du déploiement d'applications mobiles pour les plateformes iPhone et Android.

Les équipes de développement de cette startup ont subi d'énormes pressions en déployant rapidement des améliorations à leurs applications mobiles actuelles selon un échéancier agressif. Elles ont travaillé en moyenne 60 heures par semaine au cours des six derniers mois. L'équipe de direction a été réticente à embaucher du personnel supplémentaire et se préoccupe plus de l'augmentation des revenus tout en réduisant les coûts d'exploitation et de développement.

Au cours des derniers mois, il y a eu une hausse du taux d'absentéisme et du nombre d'employés se mettant en arrêts-maladie. Plusieurs démissions chez AppAtoZ ont entraîné une augmentation de la charge pour ceux qui sont restés. Le recrutement et le temps d'adaptation des nouveaux employés ne parvient pas à alléger rapidement la charge de travail de l'équipe de développement.

Quelles stratégies à **long terme** doivent être envisagées pour résoudre l'épuisement professionnel et le stress des employés ?

- A)
 - Ajouter à l'équipe un mixte d'employés permanents et contractuels afin de mieux gérer la charge de travail
 - Faire en sorte que l'équipe de direction et le Développement identifient les facteurs contribuant à l'épuisement dans l'environnement professionnel
 - Élaborer un plan pour résoudre l'épuisement professionnel et le stress des employés
- B)
 - Licencier l'équipe de direction car elle est incompétente
 - Embaucher une équipe de direction plus compétente qui sait comment créer un équilibre travail-vie et une culture durable et réaliste
 - Donner à l'équipe de développement la possibilité de prendre le temps de trouver son équilibre
- C)
 - Mettre les membres de l'équipe de développement en congés
 - Engager des développeurs pour gérer la charge de travail accrue dans le service du développement pendant les périodes de pointe de l'année
 - Faire en sorte que les développeurs puissent faire appel à une assistance psychologique en fonction de leurs besoins
- D)
 - Mettre les membres de l'équipe de développement en congés
 - Faire en sorte que l'équipe de direction et le Développement identifient tous les facteurs de l'environnement professionnel qui peuvent contribuer à l'épuisement
 - Élaborer un plan pour résoudre l'épuisement professionnel et le stress des employés

10 / 50

Vous travaillez en tant que développeur dans une équipe DevOps. Vous tentez de faciliter la collaboration entre tous les membres de votre équipe qui compte deux hommes développeurs seniors et une femme, administratrice système junior.

Dès le début, les membres de votre équipe ont eu des relations difficiles. Vous êtes alors intervenu pour tenter d'améliorer leur collaboration. Et cela a fonctionné car ils ont arrêté de se disputer et se sont mis à travailler plus efficacement.

Maintenant, vous remarquez que l'administratrice junior est systématiquement d'accord avec les développeurs seniors.

Comment devriez-vous réagir à ce mode de fonctionnement ?

- A) Vous devriez vous abstenir d'intervenir tant que les membres de l'équipe ne se disputent pas. Ils semblent s'entendre et il vaut donc mieux ne pas intervenir.
- B) Il est important qu'Exploitation exprime toute réserve quant aux versions (builds). Vous devriez donc entraîner l'administratrice pour qu'elle cesse d'être accommodante et devienne plus assertive.
- C) Les développeurs seniors ont plus de responsabilités ; vous devriez donc leur demander d'être plus aimables et de penser aux inquiétudes de l'Exploitation si l'administratrice ne le fait pas.
- D) Vous devez commencer à travailler pour l'équipe d'Exploitation, pour le moment, même si vous êtes vous-même développeur, afin de pouvoir montrer le chemin à l'administratrice junior.

11 / 50

De quelle manière Agile et Scrum promettent-ils un développement de logiciels plus rapide et prévisible ?

- A) En permettant une collecte et un traitement des spécifications meilleurs et plus exhaustifs avant la conception
- B) En permettant de petites équipes autonomes qui s'autoorganisent et gèrent elles-mêmes leur planning
- C) En permettant au Propriétaire de produit de participer aux mêlées quotidiennes
- D) En permettant au Responsable de projet de modifier rapidement des priorités selon les nécessités

12 / 50

Qu'est-ce qu'un ITSM allégé ?

- A) Un ITSM centré sur la continuité des opérations
- B) Une nouvelle version d'ITIL proposée comme une norme
- C) Une mauvaise mise en œuvre des processus ITIL
- D) Une gestion des mises en production orientée sur ITSM

13 / 50

Vous travaillez pour une entreprise qui applique les pratiques Lean et Agile. Votre président directeur général (PDG) n'est pas convaincu de l'intérêt de mettre en œuvre encore une autre méthode : DevOps.

Vous êtes un expert DevOps. Vous avez la conviction que DevOps est un atout pour toute entreprise qui développe des logiciels.

Qu'est-ce que DevOps peut apporter à l'entreprise pour laquelle vous travaillez ?

- A) Créer des Récits Utilisateur et des exigences fonctionnelles de meilleure qualité
- B) Réduire la cadence des mises en production pour que les clients souffrent moins des impacts des mises à jour
- C) Former les professionnels à la fois au Développement et à l'Exploitation
- D) Écrire des codes et les mettre plus rapidement en production, tout en facilitant le résultat du business

14 / 50

La Direction demande au service informatique un meilleur soutien au business et un alignement sur les objectifs de ce dernier. En votre qualité de Directeur des Systèmes d'Information (CTO), vous avez décidé, entre autres mesures, de réduire la charge de travail supportée par Exploitation.

De quelle manière est-ce que DevOps pourra le **mieux** vous aider à améliorer la gestion des niveaux de service (SLM) existante ?

- A) En abandonnant les bonnes pratiques ITSM car cette approche est trop lourde
- B) En élaborant de meilleurs accords sur les niveaux opérationnels (OLA) entre Dev et Ops
- C) En mettant en œuvre de nouveaux processus ITSM basés sur la version 3 d'ITIL
- D) En réalignant la gestion des services informatique (ITSM) pour utiliser un ensemble d'informations minimales requises (MRI)

15 / 50

Quelle mise en œuvre de DevOps est la **plus** adaptée à une entreprise qui utilise l'approche système d'enregistrement (SoR) ?

- A) Collaboration
- B) Livraison continue
- C) La façon Toyota

16 / 50

Quel est le **principal** avantage de l'utilisation du système Obeya ?

- A) Facilite l'émission de plaintes client, afin de s'assurer que l'équipe reçoit suffisamment de feedback pour s'améliorer continuellement
- B) S'attaque au stress au sein des équipes, de manière à ce que les membres de l'équipe puissent tenir une cadence durable
- C) Améliore la signalisation quotidienne des bogues, ce qui réduit les remaniements ainsi que les transferts de bogues à d'autres terminaux
- D) Une prise de décision rapide basée sur le statut actuel, grâce à une collecte et un partage d'information rapides

17 / 50

Vous travaillez dans une entreprise informatique qui vient juste d'adopter DevOps. Vous vous rendez compte que non seulement la culture de l'entreprise doit changer mais également ses pratiques et ses outils. L'entreprise surveille déjà les projets logiciels pour détecter tout problème.

Vous proposez d'ajouter les mesures suivantes :

- automatiser le processus de gestion des changements
- mettre en œuvre un contrôle des accès pour prévenir de tout changement non autorisé

Pourquoi cela est-il nécessaire ?

- A) L'automatisation permet la mise en œuvre plus rapide des changements, tout en conservant la confiance. Le contrôle d'accès est nécessaire pour éviter une résolution de problèmes heuristique et des interruptions de service non planifiées.
- B) L'automatisation est nécessaire pour réduire le nombre de changements. Le contrôle d'accès devrait éviter que les clients modifient eux-mêmes le logiciel, sans que vous en soyez informé et sans que vous puissiez contrôler la situation.
- C) L'automatisation contribuera à rendre l'équipe d'Exploitation inutile. Le contrôle d'accès doit être mis en œuvre parce que les projets DevOps fonctionnent dans le Cloud, ce qui implique des risques supplémentaires.

18 / 50

La virtualisation et le Cloud Computing sont des techniques susceptibles d'aider et de faciliter les pratiques DevOps.

De quelle manière aident-elles DevOps ?

- A) Le Cloud Computing rend l'accès à distance possible, ce qui donne aux clients plus de contrôle.
- B) Des environnements virtuels sont plus faciles à standardiser et utilisent le matériel plus efficacement.
- C) Une infrastructure virtuelle est facile à comprendre et ne nécessite aucune maintenance.

19 / 50

Quel que soit le projet, les accords sur les niveaux de service (ANS) sont importants car ils indiquent ce dont vous convenez avec le client. Toutefois, dans DevOps, le ANS a un autre objectif important.

Quel est cet objectif ?

- A) Les clients sont responsables de l'élaboration d'un ANS pour l'équipe DevOps. En cela, il remplace une phase formelle parmi les responsabilités d'Exploitation.
- B) Dans le ANS, le client peut indiquer toutes ses exigences non fonctionnelles, de manière à ce que le Développement puisse se concentrer sur ces dernières.
- C) Le ANS indique le niveau de service acceptable. Le Développement doit être en mesure de comprendre le ANS et à aider l'Exploitation à les respecter.

20 / 50

Votre équipe doit développer un pipeline de déploiement pour un nouveau produit. Dans le cadre de l'intégration continue, vous devez définir l'étape de commissionnement de code dans le pipeline. Vous discutez de cette étape avec les membres de votre équipe.

Le Maître de processus dit : « La définition d'achevé (Definition of Done) doit être définie pendant ou après l'étape de commissionnement. Si le code n'est pas achevé lorsque les modifications sont enregistrées dans la version centrale du programme, il faut arrêter le travail. »

Est-ce vrai ?

- A) Oui. Si le travail n'est pas achevé, le Maître de processus ne fait pas du bon travail. Ce point devrait être résolu immédiatement.
- B) Oui. Un travail qui n'est pas achevé ne devrait pas être enregistré dans la base d'intégration continue, car il n'a aucune valeur pour le client.
- C) Non. La définition d'achevé a lieu lors des réunions avec le client. Cela ralentirait trop le travail d'attendre une telle réunion.
- D) Non. Le travail dans un pipeline de déploiement ne doit jamais s'arrêter. Si un code n'est pas achevé, il doit juste être inactif.

21 / 50

Quel est le **principal** avantage d'accroître la diversité de l'équipe pour inclure un éventail plus large de situations personnelles et de cultures ?

- A) Il enrichit l'équipe d'un plus grand nombre d'expériences et de points de vue.
- B) Il conduit à une diminution des frictions entre les membres de l'équipe.
- C) Il limite l'originalité et la capacité à trouver de nouvelles idées.
- D) Il prend plus de temps pour prendre une décision spécifique.

22 / 50

Vous travaillez pour une petite entreprise qui dispose d'une seule équipe DevOps. Cette dernière travaille à une application qui est constitué de plusieurs composants. Certains sont nouveaux et d'autres nécessitent juste une mise à jour.

À l'heure actuelle, chaque composant dispose de son propre pipeline de déploiement. L'équipe est fière de la livraison continue qu'elle réalise dans la grande majorité des pipelines de déploiement. Sa production est également élevée et de bonne qualité.

Que devriez-vous faire ?

- A) Conserver les différents pipelines et encourager l'équipe à accroître sa livraison continue
- B) Conserver uniquement les pipelines présentant une livraison continue et fusionner les autres pipelines sans livraison continue
- C) Conserver uniquement deux pipelines : l'un pour le développement et l'autre pour la maintenance
- D) Viser un seul et unique pipeline en collaboration avec votre équipe, en expliquant les risques de multiples pipelines

23 / 50

Vous êtes en train d'évaluer Company Builders, une organisation de moyenne à grande taille qui a adopté les pratiques DevOps il y a quelques années.

L'entreprise vous a engagé afin de déterminer son niveau actuel de maturité. Lorsque vous aurez terminé, on attend de vous que vous suggériez des améliorations. L'entreprise veut connaître le domaine sur lequel elle doit se concentrer pour atteindre le niveau de maturité suivant : *Niveau 2 - Gestion quantitative*.

D'après vos investigations, la majorité des domaines se situe au *Niveau 1 - Cohérent*, à deux exceptions près :

- A. **Environnements et déploiement.** Ce domaine gère les déploiements orchestrés et a testé les processus de mise en production et de restauration.
- B. **Gestion des versions et intégration continue.** Dans ce domaine, vous trouvez des développements et des tests réguliers automatisés, et que toute version peut être recrée à partir du contrôle de source via un processus automatisé.

Déterminez le niveau de maturité de ces deux domaines, d'après les informations données. Ensuite, donnez votre recommandation par rapport au point d'amélioration.

Sur lequel de ces deux domaines, Company Builders devrait travailler, avant de passer au niveau 2 ?

- A) Environnements et déploiement d'une part, et gestion des versions et intégration continue, d'autre part, se trouvent tous les deux au niveau 0. Les efforts doivent porter simultanément sur les deux environnements.
- B) Environnements et déploiement d'une part, et gestion des versions et intégration continue, d'autre part, sont soit au niveau 1, soit au-dessus de ce niveau. Les efforts doivent porter sur les autres domaines afin de progresser.
- C) Environnements et déploiement se trouve au niveau 0. Gestion des versions et intégration continue se trouve au niveau 1. L'attention doit tout d'abord porter sur environnement et déploiement.
- D) Environnements et déploiement se trouve au niveau 2. Gestion des versions et intégration continue se trouve au niveau 0. L'attention devrait uniquement porter sur gestion des versions et intégration continue.

24 / 50

Votre entreprise commercialise un service de sauvegarde en ligne. Un de vos clients demande d'ajouter de nouvelles fonctionnalités à votre service. Il veut ces nouvelles fonctionnalités dans un délai d'une semaine sinon il ira voir ailleurs.

Vous estimez que les nouvelles fonctionnalités sont importantes et vous savez que l'équipe de Développements peut les produire rapidement. Toutefois, vous rencontrez les problèmes suivants :

- Les testeurs ont besoin de beaucoup de temps pour régler les bogues.
- Les testeurs trouvent des bogues que les développeurs ont résolu il y a longtemps.
- Il est rarement possible de faire une démonstration de l'application car cette dernière fonctionne rarement.
- Les présentations se font rares.

Quel est votre problème et comment devriez-vous le résoudre ?

- A)** Vous déployez des codes défectueux. Vous devriez résoudre cette situation en augmentant la compréhension du processus de déploiement, en augmentant la collaboration et en travaillant avec plus de discipline.
- B)** Votre gestion des configurations laisse à désirer. Vous devriez résoudre ce problème en augmentant la collaboration entre Développement et Exploitation, ainsi qu'en augmentant la surveillance, la journalisation et la virtualisation.
- C)** Votre processus d'intégration continue n'est pas géré correctement. Vous devriez résoudre ce problème en augmentant la rapidité des tests automatisés et la rapidité de l'étape de commissionnement ainsi qu'en augmentant la compréhension du processus d'intégration continue.
- D)** Votre stratégie de test n'est pas efficace. Vous devriez résoudre ce problème en automatisant les tests et en augmentant la collaboration entre les testeurs et le reste de l'équipe.

25 / 50

Quel est l'avantage de l'intégration continue pour un DevOps efficace ?

- A)** Des cycles de tests prolongés avant la mise en production de fonctionnalités
- B)** De longues périodes de temps entre les mises en production de fonctionnalités
- C)** Des mises en production de fonctionnalités plus fréquentes et en temps opportun

26 / 50

Une entreprise multinationale est confrontée à de nombreux défis dans la consolidation des envois de commissionnement à partir de sites locaux vers le référentiel central basé à Dallas, Texas. Les sites concernés se trouvent à Mexico, Paris, San Diego et en Angleterre. Le mode d'envoi des commissionnements varie d'une région à l'autre et parfois, il n'est pas évident de déterminer clairement si certains tests ont échoué ou non.

Il y a quatre approches possibles :

1. Ne pas envoyer une version défectueuse
2. Toujours procéder à tous les tests locaux avant l'envoi, ou s'assurer que le serveur d'intégration continue à le faire
3. Attendre que les tests de commissionnement soient satisfaisants avant de poursuivre
4. Ne pas commenter les échecs aux tests

Parmi ces approches, lesquelles sont les **plus** pertinentes afin de s'assurer que les différents sites traitent leurs difficultés actuelles ?

- A) 1 & 2
- B) 1, 2 & 3
- C) 2 & 3
- D) 2, 3 & 4

27 / 50

Examinons la constitution d'un pipeline de déploiement de base.

Quelle étape détermine que le système fonctionne au niveau fonctionnel et non-fonctionnel ?

- A) Le test de validation automatisé
- B) Le test unitaire et test d'assemblage
- C) Le test de validation manuel
- D) Le contrôle de version

28 / 50

Le fait d'utiliser un même processus de déploiement pour chaque environnement dans lequel votre application fonctionne est une bonne pratique de DevOps. Cela assure que la version est testée efficacement. Vous utilisez des scripts pour automatiser votre processus de construction et de déploiement.

Quelle est la **meilleure** manière de le faire ?

- A) Utiliser un script pour chaque environnement et l'entretenir dans le cadre du système de contrôle des versions.
- B) Utiliser un script spécifique pour chaque environnement afin de tenir compte des différences entre les environnements.
- C) Utiliser le même script pour chaque environnement, en optant pour des paramètres manuels pour des configurations spécifiques.
- D) Déployer un même script dans chaque environnement et gérer séparément les informations de configuration.

29 / 50

Lors de la mise en production d'un nouveau service informatique, une tâche en cours s'arrête inopinément.

Qu'est-ce qui ne peut **pas** constituer une cause valable de cet événement ?

- A) Le Gardien n'a pas tenu compte de l'interrelation entre les éléments de travail et des critères d'acceptation de services (SAC).
- B) Les exigences non fonctionnelles n'étaient pas claires à cause d'un Récit Utilisateur imparfait.
- C) Le Maître de service n'est pas parvenu à se mettre d'accord avec l'utilisateur sur la fin de vie du service à mettre en production.
- D) Les processus de développement n'ont pas bénéficié d'autant de travail que nécessaire, comme l'indiquent les critères d'acceptation de services.

30 / 50

La société AppBC utilise DevOps. Elle a mis en œuvre le déploiement continu ainsi qu'un solide pipeline de déploiement comportant des tests de validation hautement automatisés et fournit chaque jour de nouveaux logiciels à la production.

AppBC possède une grande base de données et de nombreux utilisateurs. Elle applique une stratégie de test de performance solide et détaillée. Son environnement étant très vaste et complexe, chaque nouvelle version introduit quelques bugs en production.

Quelle stratégie est la **plus** susceptible d'aider AppBC à éviter ces bogues ?

- A) Tester à petite échelle (Canary Releasing)
- B) Automatiser les tests de performance
- C) Réduire le taux de livraison
- D) Utiliser le déploiement en mode bleu-vert

31 / 50

DevOps reprend de très importants concepts d'Agile issus du système de production Toyota.

Pourquoi est-ce qu'un flux pièce par pièce (one-piece-flow) est important pour l'adoption de DevOps ?

- A) Cela permet à votre équipe de travailler à une cadence durable avec une vitesse prévisible.
- B) Cela aide l'équipe à se concentrer sur l'ajout de fonctionnalités dont la valeur métier est la plus importante possible.
- C) Cela augmente le partage de responsabilité du travail entre les différents membres de l'équipe.
- D) Cela réduit les goulets d'étranglement en limitant le nombre de tâches effectuées simultanément par une personne.

32 / 50

S Corporation est une entreprise de taille moyenne. Elle fournit des pièces détachées automobiles à une grande entreprise T Motors Corporation. La part de T Motors Corporation représente près de 60 % du total des ventes de S Corporation.

Le Comité de Direction se réunit pour discuter du nouveau partenariat. T Motors Corporation exige que S Corporation passe en mode de livraison juste-à-temps (Just-in-Time), sous peine de rompre ses activités commerciales avec S Corporation. La perte de ce client serait fatale pour S Corporation, il y a donc une certaine urgence à adopter le mode juste-à-temps. Ce changement doit se produire dans les 6 mois, donc il y a 5 mois au plus pour se préparer.

La finalité consiste à mettre en œuvre le suivi des pièces par le biais de l'identification par radio fréquence (RFID). Cela devrait assurer la transparence du processus de production. Un examen rapide du processus actuel devrait faciliter le passage à un processus intégrant le système RFID.

La Directrice des Systèmes d'Information (CTO) est sollicitée pour piloter le changement. Elle estime cela possible, si une approche DevOps est utilisée pour créer une mise en production optimale. Idéalement, le concept pour la production avec la RFID devrait être développé en premier. Un système de contrôle de la production utilisant les données de la puce RFID devrait être mis en œuvre en fin de parcours. Cependant, il n'y a pas assez de temps pour procéder de manière séquentielle. En conséquence, ces trois étapes doivent se dérouler en même temps.

La CTO affecte Em, un Scrum Master, au projet. Le Développement se prépare à construire un pipeline de déploiement.

Em constate qu'ils sont enthousiastes et travaillent dur, mais qu'ils pourraient être plus disciplinés. En outre, la fréquence des mises en production doit être plus élevée.

Sur quel point Em devrait-il se concentrer **en premier** ?

- A) Em devrait se concentrer sur la communication, puisque c'est la chose la plus importante dans DevOps. Em devrait commencer par briser la glace avec l'équipe et fixer quelques règles de communication.
- B) Em devrait commencer par discuter d'une Cartographie des chaînes de valeur et construire un flux pièce par pièce avec l'équipe, sachant que le flux et la rationalisation des processus sont très importants.
- C) Em devrait commencer par discuter de l'infrastructure et de l'environnement de travail avec les membres de l'équipe, sachant que l'efficacité de DevOps est maximale lorsque tous les outils et les pratiques fonctionnent.
- D) Em devrait commencer par rassembler toutes les parties prenantes, les sensibiliser à DevOps et leur demander leur aide dans le cadre de la diffusion du changement de culture, étant donné que le changement de culture est un prérequis pour DevOps.

33 / 50

Les membres de votre équipe DevOps travaillent bien ensemble à une cadence soutenue. En intégrant suffisamment de temps mort dans le processus, l'équipe a le temps et la concentration nécessaire pour vérifier et tester minutieusement les versions (builds). À l'heure actuelle, votre équipe procède manuellement aux tests et au déploiement. Sa cadence est suffisamment élevée pour apporter régulièrement une importante valeur métier.

Votre président directeur général (PDG) a sollicité votre conseil sur l'automatisation au sein de cette équipe.

Quel conseil devez-vous donner ?

- A) Automatiser autant que possible, de manière à ce que l'équipe puisse ajouter plus de fonctionnalités et fournir encore plus rapidement de la valeur métier
- B) Automatiser les tests d'acceptation, mais pas les déploiements, car un processus manuel est plus fiable
- C) Automatiser les déploiements pour améliorer la durée d'un cycle, mais pas les tests, afin de permettre un apprentissage à partir des bogues
- D) Ne pas automatiser les méthodes de cette équipe, car les méthodes actuellement appliquées par l'équipe fournissent d'excellents résultats

34 / 50

Un Directeur des Systèmes d'Information (CTO) affecte Michael, son employé le plus fiable et qui est également Scrum Master, à un projet. L'équipe de Développement se prépare à construire un pipeline de déploiement.

Michael a confiance dans les bonnes intentions et la spontanéité de l'équipe de développement, mais souhaite que ses membres soient plus disciplinés. De plus, la fréquence des mises en production devrait être plus soutenue. Michael veut que l'équipe de développement mette plus souvent en œuvre des mises en production.

L'un des membres de l'équipe dit : « La chose la plus importante à propos de ce nouveau pipeline de déploiement est son automatisation. Nous devrions d'abord automatiser le pipeline de déploiement ».

Cette déclaration est-elle correcte ?

- A) Oui, c'est correct. L'automatisation du pipeline de déploiement est le plus important facteur d'accroissement de l'efficacité.
- B) Oui, c'est correct. En vous concentrant sur la création d'un pipeline de déploiement automatisé, vous évitez les éventuels problèmes que vous pouvez rencontrer plus tard.
- C) Non, ce n'est pas correct. La réalisation d'un flux pièce par pièce (single-piece-flow) et un processus de déploiement solide devraient constituer la première priorité. L'automatisation du processus peut venir plus tard.
- D) Non, ce n'est pas correct. Au lieu d'automatiser le pipeline de déploiement, le processus de test devrait être automatisé en premier.

35 / 50

Votre entreprise change ses pratiques et commence à travailler avec DevOps. Votre équipe vous suit dans la mise en œuvre de ce changement. Vous êtes en train de discuter des bonnes pratiques pour l'étape de commissionnement du code.

Votre collègue Sun déclare : « Lorsqu'une version (build) ne fonctionne plus correctement et que personne n'en assume la responsabilité, il faudrait trouver la personne qui a causé ce dysfonctionnement et l'interpeller afin qu'elle le répare. »

Est-ce une bonne idée ?

- A) Oui. Seule la personne qui a causé ce dysfonctionnement peut le résoudre, il faut donc l'identifier, même si cela la met mal à l'aise.
- B) Oui. Il faut toujours trouver la personne responsable du dysfonctionnement d'une version. Si vous ne le faites pas, vos collègues peuvent appliquer cette règle.
- C) Non. DevOps est un environnement exempt de blâme. Si un collègue n'assume pas la responsabilité, ne le forcez pas à le faire.
- D) Non. Il faut d'abord réparer la version. Ensuite, il faut prendre le temps d'identifier la personne responsable et la punir.

36 / 50

L'équipe de développement d'XAppGo est confrontée à de nombreux défis concernant leurs pratiques de test actuelles. Elle utilise, à ce jour, un processus de test de validation manuel. Les développeurs pensent que la suite de tests unitaires qu'ils ont créée est suffisamment méticuleuse pour protéger des régressions.

L'équipe de développement engage de grosses dépenses pour les tests manuels de validation pour chaque version. La haute direction a demandé à ce que l'équipe de développement mette en œuvre des tests de validation automatiques afin de réduire les coûts globaux du testing et de minimiser le nombre de codes défectueux et de régressions introduites dans l'environnement de production.

Quels principes doivent être suivis lors de la définition des critères de validation de votre application en vue d'une automatisation ?

- A) Les principes d'Agile
- B) Les principes d'ATAM
- C) Les principes d'INVEST

37 / 50

Quel est le mécanisme le **plus** efficace pour migrer des données d'une manière automatisée ?

- A) Créer un schéma des versions de base de données et le conserver dans le contrôle des versions
- B) Créer et gérer des jeux de données plus petits, de manière à faciliter la migration
- C) S'assurer que tous vos scripts ont été correctement testés avant de procéder à la migration des données
- D) S'assurer que vous avez une procédure de restauration au cas où la migration tourne mal

38 / 50

L'entreprise X-AppGo a rencontré des problèmes avec ses processus de restauration. Cela entraîne souvent des pertes de données sensibles dans les bases de données d'applications de production lorsque des scripts de restaurations sont exécutés.

Dans quelle situation est-il **impossible** d'exécuter des scripts de restauration sans perdre de données sensibles ?

- A) Le script de restauration efface les données que seule la nouvelle version utilise.
- B) Le script de restauration implique le déplacement d'une colonne d'une table à l'autre.
- C) Le script de restauration remet en place des données issues de tables temporaires.

39 / 50

ACMECONST a essayé de nombreux dysfonctionnements au niveau des applications et matériels après avoir déployé les logiciels et renouvelé le matériel de leurs routeurs et commutateurs.

Il leur a été très difficile de les rétablir dans leur état d'origine, après avoir observé ces pannes pendant leurs fenêtres de maintenance. Cette situation a entraîné de nombreuses heures de rétablissement, au-delà des fenêtres de maintenance normale et une panne de longue durée de leurs applications vitales.

Un approvisionnement automatisé et une infrastructure autonome peuvent aider dans cette situation, toutefois certaines choses doivent être prises en considération.

Quels éléments doivent être gérés avec précaution afin de réduire le risque d'interruption lors du déploiement de l'environnement de production ?

- A) Des journaux de surveillance détaillés pour résoudre les problèmes en cas d'échec de mise à jour d'une application
- B) Des points d'intégration externes tels que des systèmes et des services externes
- C) Configurations de serveurs et informations sur le compte utilisateur sous-jacentes
- D) L'ensemble d'outils d'automatisation de l'approvisionnement et l'architecture autonome

40 / 50

L'entreprise X-AppGo a rencontré des problèmes avec son application de base. L'application ne se connecte pas correctement aux autres applications externes. Celles-ci doivent obtenir de manière efficace des variables déterminées, de manière à pouvoir exécuter des appels déterminés. Une équipe est en train de développer l'application fondamentale, et l'entreprise veut conserver cette organisation pour de très bonnes raisons commerciales.

L'un des développeurs suggère de séparer un composant du codebase X-AppGo, pour résoudre les problèmes d'interfaçage.

Qu'est-ce qui constitue une bonne raison de séparer un composant dans ce cas ?

- A) Convertir un jeu de plug-in dans le codebase en un codebase monolithique
- B) Limiter l'impact des changements et faciliter la modification du codebase
- C) Le codebase X-AppGo devra être séparé et géré par différentes équipes
- D) Il n'y a aucune bonne raison et cela prendra plus de temps pour compiler

41 / 50

Même les plus petites applications développeront une dépendance à d'autres composants et bibliothèques. En conséquence, la compréhension et la gestion des dépendances est une activité capitale au sein du déploiement continu, afin de préserver le flux dans le pipeline de déploiement.

Vous avez construit une application qui utilise deux bibliothèques. Chacune de ces bibliothèques dépend d'une troisième bibliothèque sous-jacente bien qu'elles réfèrent à des versions différentes. Cela engendre une dépendance spécifique.

Quelle est la **meilleure** solution pour résoudre ou éviter cette dépendance ?

- A) Regrouper toutes les bibliothèques en une seule bibliothèque, de manière à pouvoir directement renvoyer à la bibliothèque et éviter le problème
- B) Gérer les bibliothèques à l'aide du contrôle des versions, de manière à pouvoir immédiatement voir si vous créez ce type de dépendance
- C) Conserver un aperçu visuel de toutes vos dépendances sur des Post-It sur un grand tableau, de manière à pouvoir suivre le flux
- D) Vérifier uniquement de petites portions de votre chaîne de compilation, de manière à pouvoir facilement déboguer les problèmes qui pourraient se produire lors de l'enregistrement

42 / 50

Dans un environnement de déploiement continu, il est important que tout soit soumis à un contrôle des versions, de manière à pouvoir trouver rapidement les erreurs ou à procéder à une restauration si nécessaire.

Toutefois, il n'est **pas** recommandé de conserver des fichiers binaires compilés dans le contrôle des versions

Pourquoi fait-on cette exception ?

- A) Les fichiers binaires compilés tendent à se trouver dans de gros fichiers qui changent avec chaque version (build) et sont mis à jour automatiquement.
- B) Plusieurs membres de l'équipe travaillent sur les fichiers binaires, il n'est donc pas pratique de les conserver dans le contrôle des versions.
- C) Les fichiers binaires alimentent vos compilateurs, qui sont déjà conservés dans le contrôle des versions.
- D) Cela n'est pas nécessaire, considérant que la recompilation est effectuée dans le cadre du processus auquel une version est normalement soumise.

43 / 50

Vous souhaitez adopter une approche holistique pour gérer toutes votre infrastructure informatique.

Quels sont les deux principes sur lesquels il est **préféré** de baser cette approche ?

- A)
 1. L'état que vous envisagez pour votre infrastructure devrait être spécifié par le biais d'une configuration contrôlée par les changements.
 2. Vous devriez toujours connaître l'état réel de votre infrastructure grâce à la surveillance et à la gestion des événements.
- B)
 1. L'état que vous envisagez pour votre infrastructure doit être spécifié par le biais d'une configuration contrôlée par les changements.
 2. Vous devriez toujours connaître l'état réel de votre infrastructure grâce aux instruments en place et à la gestion des incidents.
- C)
 1. L'état que vous envisagez pour votre infrastructure devrait être spécifié par le biais d'une configuration en version contrôlée.
 2. Vous devriez toujours connaître l'état réel grâce à la gestion des incidents en cours et des événements.
- D)
 1. L'état que vous envisagez pour votre infrastructure devrait être spécifié par le biais d'une configuration en version contrôlée.
 2. Vous devriez toujours connaître l'état réel de votre infrastructure grâce aux instruments et à la surveillance.

44 / 50

Les équipes ayant de bonnes pratiques de collaboration ont des tickets de travail synchronisés. Un Directeur des Systèmes d'Information (CTO) a utilisé « go and see » pour examiner le fonctionnement de l'équipe d'Exploitation. Après la mise en production, l'équipe d'Exploitation redéfinit toujours l'infrastructure d'exploitation.

Quel est le **meilleur** conseil pour améliorer cette pratique ?

- A) Le mieux est de ne rien faire. Aucune amélioration n'est possible car il faut toujours repasser par l'étape de redéfinition.
- B) Elle devrait envisager une manière de modéliser la structure d'exploitation et de contrôler l'accès à l'environnement d'exploitation.
- C) Elle devrait procéder à la revue de l'infrastructure d'exploitation, de manière à ce que cela devienne un processus automatisé.
- D) Elle devrait commencer à partager ses connaissances du processus de déploiement avec l'équipe de Développement.

45 / 50

Quel est le moment propice pour que l'Exploitation informe le Développement d'un changement opérationnel ?

- A) Il n'est pas nécessaire d'informer le Développement. Les changements opérationnels concernent uniquement l'équipe d'Exploitation.
- B) Immédiatement. Le Développement doit être informé dès que possible.
- C) Lors de la mêlée des mêlées Scrum le matin suivant.
- D) Lorsque l'équipe d'Exploitation a procédé au test de validation.

46 / 50

Vous souhaitez que votre organisation DevOps soit mature. Il y a plusieurs façons d'y arriver.

Qu'est-ce qui ne constitue **pas** une manière d'améliorer la maturité de votre organisation DevOps ?

- A) Définir clairement des objectifs ponctuels pour permettre aux membres de votre équipe de déterminer si leurs activités quotidiennes ont de la valeur.
- B) Définir clairement les processus, et permettre aux membres de l'équipe d'améliorer quotidiennement le processus.
- C) Conserver des enregistrements de toutes les réunions, de sorte que les membres de votre équipe aient facilement accès à toutes les communications.
- D) Surveiller et enregistrer les activités quotidiennes pour contribuer à l'identification des petits domaines de progrès quotidiens et à leur mise en valeur.

47 / 50

Vous travaillez pour un prestataire de services informatiques. Dans le cadre de votre plan de continuité des opérations, vous voulez garantir votre capacité à satisfaire continuellement les niveaux de service minimum convenus.

Vous voulez assurer la continuité des services informatiques.

Comment est-ce que DevOps peut vous être utile en matière de Gestion de la continuité des services informatiques ?

- A) Les valeurs culturelles de DevOps, Affinité et Collaboration, font en sorte que Service est très apprécié par les membres de l'équipe DevOps.
- B) DevOps prépare les routines à appliquer par l'équipe en cas de catastrophe et les pratiques Obeya en mettant délibérément le chaos dans le système.
- C) Les mesures de réduction des risques et les options de restauration sont probablement codifiées, car les équipes d'Exploitation travaillent en collaboration avec celles du Développement.
- D) La gestion des niveaux de service devient plus importante dans DevOps, car c'est la tâche principale du Maître de processus que de surveiller ce point.

48 / 50

La société ACMECONST a activement élargi sa présence internationale en augmentant le nombre d'embauches et des équipes d'ingénieurs réparties dans le monde entier. Elle a également augmenté sa clientèle au rythme annuel effréné de 30%.

Des décisions prises autrefois rapidement, lorsque l'équipe d'ingénieurs se trouvait dans un seul local, prennent maintenant beaucoup plus de temps, causant une frustration dans toute l'organisation. Le processus de prise de décision est soumis à de nombreuses couches hiérarchiques et est plus étendu, ce qui cause la désillusion des ingénieurs vis-à-vis du processus dans son ensemble.

Il existe également une confusion croissante quant à la paternité des divers problèmes qui se présentent, ce qui provoque une hésitation au niveau des prises de décision. Les ingénieurs ont également le sentiment que leur créativité est étouffée par les processus supplémentaires et la bureaucratie, et cela commence à miner leur moral.

Quelle est la **meilleure** approche pour résoudre cette situation ?

- A) Conserver les processus actuels, mais établir clairement les rôles, la responsabilité et la propriété de chaque processus, établir une méthode efficace pour évaluer la productivité par rapport aux risques, procéder à des changements progressifs et créer des lieux sûrs pour l'expérimentation.
- B) Réévaluer les processus afin de déterminer ce qui peut être rationalisé et établir clairement les rôles, la responsabilité et la propriété de chaque processus, établir une méthode efficace pour évaluer la productivité par rapport aux risques, procéder à des changements progressifs et créer des lieux sûrs pour l'expérimentation.
- C) Réévaluer les processus afin de déterminer ce qui peut être rationalisé et établir clairement les rôles, la responsabilité et la propriété de chaque processus, établir une méthode efficace pour évaluer la productivité par rapport aux risques, procéder à des changements progressifs et réduire le nombre d'expérimentations afin d'éviter les défaillances inutiles d'application.

49 / 50

Dans l'entreprise X-AppGo, il existe un conflit entre l'équipe d'Exploitation en Colombie et l'équipe de Développement en Irlande, causé par le fait qu'elles ont des priorités et des objectifs différents. En raison de ce conflit, le temps et les efforts nécessaires pour résoudre les problèmes, qui ont un impact sur l'entreprise, ne cessent d'augmenter.

Quelles pratiques-clés, l'entreprise X-AppGo devrait-elle envisager en vue de réduire les conflits et améliorer la collaboration entre les équipes de Développement et d'Exploitation ?

- A)
 - 1. Permettre aux équipes de Développement et d'Exploitation de travailler séparément l'une de l'autre, si elles le souhaitent, afin d'éviter les conflits.
 - 2. Obtenir l'appui de l'ensemble du conseil d'administration par rapport au soutien des équipes de Développement et d'Exploitation.
- B)
 - 1. Demander à un sponsor du conseil d'administration de l'entreprise de parler à l'équipe DevOps de l'importance de la collaboration.
 - 2. Former les équipes de Développement et d'Exploitation aux pratiques du DevOps, afin qu'elles apprennent à faire le travail l'une par l'autre.
- C)
 - 1. Faire en sorte que les équipes de Développement et d'Exploitation visitent d'autres entreprises dans lesquelles DevOps fonctionne bien.
 - 2. Mettre à disposition le budget pour mieux faire face aux exigences auxquelles les équipes d'Exploitation et de Développement sont confrontées.
- D)
 - 1. Recommander des visites mutuelles des équipes de Développement et d'Exploitation afin d'établir des liens, et développer la confiance et la compréhension mutuelle.
 - 2. Diffuser des connaissances auprès des équipes de Développement et d'Exploitation de sorte qu'elles collaborent plus efficacement.

50 / 50

Une équipe de Développement s'intéresse à DevOps. Elle est surtout intéressée par l'intégration continue. Ils développent et maintiennent actuellement 3 solutions majeures et 4 plus petites en utilisant les pratiques de Scrum. Chaque sprint dure 4 semaines, entraînant en moyenne 1 mise en environnement de test tous les 10 à 15 jours et 1 mise en production par mois. L'équipe veut créer un cas d'affaire qualitatif pour leur Direction afin de soutenir leurs investissements et efforts pour créer une pratique d'intégration continue.

Quels avantages tangibles de l'intégration continue aident le **plus** ce cas d'affaires ?

- A) Le déploiement de l'environnement de test une fois par jour pourrait augmenter les avantages pour l'entreprise et diminuer considérablement les coûts de développement.
- B) Cela contribue à l'esprit d'équipe. Comme l'équipe utilise déjà Scrum, l'intégration continue ne générera **pas** d'avantages mesurables pour l'entreprise.
- C) Cela augmente la stabilité et la qualité de la version grâce à un test automatisé, tout en facilitant et augmentant la vitesse générale de mise à jour.
- D) La mise en production une fois par jour pourrait augmenter les avantages pour l'entreprise et diminuer considérablement les coûts de développement.

Solutions de l'examen

1 / 50

Qu'est ce qui constitue une bonne raison de mettre en œuvre DevOps dans une organisation ?

- A) DevOps présente des cycles de feedback plus fréquents pour les nouveaux services, parce que la vitesse de développement est plus élevée.
 - B) DevOps améliore la continuité et l'agilité de l'entreprise, parce que les processus sont optimisés et seules des activités à valeur ajoutée sont effectuées.
 - C) DevOps fournit les nouveaux services juste-à-temps (Just-in-Time), parce que les versions logicielles se mettent en production plus fréquemment.
 - D) DevOps réduit le gaspillage au niveau de la collaboration entre le Développement et l'Exploitation, parce que ces deux équipes sont unifiées.
-
- A) Incorrect. Les cycles de feedback sont plus fréquents lorsque la vitesse de développement est plus élevée, mais cela est généralement dû à Scrum ou à toute autre méthode Agile, et non à DevOps en particulier.
 - B) Correct. L'ajout de valeur et l'optimisation des processus sont la clé de l'amélioration de la continuité des opérations et de l'agilité de l'entreprise. Il faut penser à ce qu'implique le fait que les services informatiques doivent toujours soutenir l'activité, il faut penser à la valeur et à l'objectif de DevOps. (*Ouvrage : C, Chapitre 2*)
 - C) Incorrect. Une livraison juste-à-temps est une très bonne chose, mais ce n'est pas, en soi, une bonne raison de mettre en œuvre DevOps. La mise en œuvre des processus Lean serait plus efficace pour atteindre cet objectif.
 - D) Incorrect. La fusion de deux équipes ne garantit pas en soi l'élimination du gaspillage (Muda). Pour éliminer le gaspillage, elles doivent commencer par modifier leurs pratiques.

2 / 50

Le Directeur des Systèmes d'Information (CTO) estime qu'il serait très efficace d'appliquer certains concepts Lean lors de la mise en œuvre de DevOps.

Quels principes ou pratiques Lean sont les **plus efficaces** lors de l'introduction de DevOps ?

- A) **Kaizen et 5S.** Agile et DevOps étant basés sur des concepts-clés de Lean, et Kaizen et 5S étant à la base du Lean, ils seront très efficaces lors de l'introduction de DevOps.
 - B) **Kaizen in advance.** DevOps requiert feedback de l'Exploitation vers le Développement. Kaizen in advance crée une boucle de feedback vers l'amont, ce qui contribue à appliquer ce principe dans DevOps.
 - C) **Système Obeya.** DevOps intègre des processus de divers styles de gestion. Le système Obeya permet de visualiser l'ensemble du processus, favorisant ainsi une introduction réussie de DevOps.
 - D) **Flux pièce par pièce (one-piece-flow) et JKK.** DevOps tire parti de la mise en place de processus en amont et d'un flux à valeur unique. Le flux pièce par pièce rend cela possible et JKK contribue à la rationalisation et à la mise en œuvre du flux.
- A) Incorrect. Bien que Lean, Agile et DevOps sont interconnectés, Kaizen et 5S ne sont pas les plus adaptés pour contribuer au succès du lancement de DevOps. Une fois DevOps en place, Kaizen peut être utilisé en vue de l'amélioration continue et 5S peut être utilisé pour entretenir les bonnes pratiques. Toutefois, ils entrent tous deux en scène après l'introduction réussie de DevOps.
- B) Incorrect. Le feedback est toujours le bienvenu, mais cela ne garantit pas nécessairement l'application la plus efficace de Lean lors de la mise en œuvre de DevOps.
- C) Incorrect. La visualisation peut être utile, mais ce n'est pas la pratique Lean la plus efficace lors de la mise en œuvre de DevOps.
- D) Correct. L'établissement d'un pipeline de déploiement fonctionnel et d'une seule pièce, contribue à la mise en œuvre réussie de DevOps. La chose la plus importante dans DevOps est l'élaboration de processus en amont, du Développement vers l'Exploitation, tout particulièrement pour un seul pipeline de déploiement. JKK est le comportement de travail le plus efficace pour y parvenir. (*Ouvrage : C, Chapitre 4*)

3 / 50

En comparaison avec un projet traité de manière conventionnelle, que doit-on changer à un projet pour qu'il réussisse dans DevOps ?

- A) Il faudrait construire une chaîne d'approvisionnement en services informatiques, utilisant un système de traction et un flux pièce par pièce (one-piece-flow).
 - B) Les développeurs devraient rejoindre l'équipe d'Exploitation pour une maintenance rapide des services.
 - C) L'Exploitation doit travailler pour l'équipe de Développement. C'est là d'où vient le nom DevOps.
 - D) Des membres de l'équipe d'Exploitation devraient se joindre à l'équipe de Développement.
- A) Correct. Un projet est un succès lorsque le processus utilise un système de traction sur base Jidoka pour créer des services (ou des produits) informatiques ayant de la valeur. (*Ouvrage : C, Chapitre 4 et B, Chapitre 1*)
- B) Incorrect. DevOps n'implique pas qu'un développeur rejoigne l'équipe d'Exploitation.
- C) Incorrect. Une simple collaboration ne garantit pas le succès d'un projet DevOps.
- D) Incorrect. DevOps ne veut pas simplement dire que l'Exploitation se joint au Développement.

4 / 50

Toute équipe qui adopte DevOps peut être qualifiée de Compact.

Quel est le principe **le plus approprié** pour faire en sorte qu'une Compact fonctionne ?

- A) La collaboration dynamique
- B) La communication perpétuelle
- C) La réduction des cycles
- D) La responsabilité partagée

- A) Incorrect. La collaboration dynamique n'est pas un principe relatif à une Compact.
- B) Correct. Les principes d'une Compact DevOps sont la communication perpétuelle, des objectifs partagés clairement définis, un ajustement dynamique et une rectification des accords. (*Ouvrage : A, Chapitre 2*)
- C) Incorrect. La réduction des cycles n'est pas un principe relatif à une Compact.
- D) Incorrect. La responsabilité partagée n'est pas un principe relatif à une Compact.

5 / 50

Plusieurs sources de connaissance, de normes et de pratiques sont disponibles pour mettre en œuvre DevOps.

Qu'est-ce qui est considéré comme étant **indispensable** pour une mise en œuvre réussie de DevOps ?

- A) Niveau 3 de certification CMMI
- B) Agile discipliné (Disciplined Agile)
- C) ISO 20001
- D) PMBOK Guide du PMI

- A) Incorrect. La certification CMMI est un facteur potentiel de succès mais n'est pas indispensable à la mise en œuvre de DevOps. Et cela est d'autant plus vrai pour un niveau de certification CMMI spécifique.
- B) Correct. Agile discipliné est un prérequis indispensable pour une mise en œuvre réussie de DevOps. (*Ouvrage : C, Chapitre 4i*)
- C) Incorrect. La certification ISO 20001 est un facteur potentiel de succès mais n'est pas indispensable à la mise en œuvre de DevOps.
- D) Incorrect. Le PMBOK Guide du PMI est un ouvrage de référence en matière de gestion de projet et ne constitue pas indispensable pour la mise en œuvre de DevOps.

6 / 50

La collaboration est l'un des quatre piliers d'un DevOps efficace.

Pourquoi est-ce aussi important ?

- A) La collaboration est la seule manière d'obtenir un flux pièce par pièce (one-piece-flow) dans DevOps.
 - B) La plus-value DevOps pour le business est obtenue grâce à la coopération entre les équipes.
 - C) Les organisations DevOps sont petites, les équipes doivent correctement collaborer.
 - D) Cela entraîne des changements et c'est un gain facile. Cela réduit également les frais de développement.
-
- A) Incorrect. Il est possible d'obtenir un flux pièce par pièce sans collaborer. C'est un mécanisme différent.
 - B) Correct. La collaboration entre toutes les équipes impliquées (Développement et Exploitation incluses) est d'une importance capitale pour générer une plus-value pour le business par le biais d'une communication et d'une automatisation plus poussées et d'un logiciel de meilleure qualité. (*Ouvrage : A, Chapitre 6 et 7*)
 - C) Incorrect. Les organisations DevOps peuvent être de très grande taille. Les équipes doivent collaborer pour apporter suffisamment de valeur au business.
 - D) Incorrect. Les outils engendrent des changements et c'est un gain facile. La collaboration est difficile. La collaboration peut réduire les frais de développement mais ce n'est pas l'objectif principal. L'objectif est d'augmenter la qualité.

7 / 50

Vous êtes le propriétaire d'une petite entreprise DevOps comptant cinq collaborateurs qui construisent des applis mobiles pour les enfants handicapés. L'une des applis dont votre équipe est la plus fière permet, à des enfants autistes, de planifier leur journée.

Cette appli a rencontré un tel succès que l'on demande à votre équipe d'étendre son utilisation à d'autres personnes qui pourraient tirer avantage du fait de pouvoir planifier leur journée. Cette demande entraînera certainement plus de complexité au niveau des codes ainsi que quelques défis techniques qui devront être surmontés par l'équipe.

Vous vous attendez à être bien payé pour cette commande et vous l'acceptez donc. Toutefois, après quelques semaines, les membres de votre équipe se disputent. Vous vous fâchez également et vous commencez à faire plus attention à ce que les membres de l'équipe font. Vous travaillez régulièrement à leurs côtés de manière à pouvoir corriger rapidement toute erreur de code que vous remarquez.

Même si la quantité de travail abattue est suffisante, l'atmosphère dans votre équipe reste mauvaise et vous admettez qu'il manque une partie de la mentalité DevOps.

Quelle est la **meilleure** stratégie pour résoudre votre problème ?

- A) Ajouter des membres à l'équipe aspirant à plus de diversité, en recrutant du personnel non technique complémentaire
 - B) Demander à des experts en DevOps externes à votre entreprise de parrainer et former les membres de votre équipe
 - C) Trouver un objectif commun et commencer à œuvrer ensemble à l'atteinte de cet objectif afin d'accroître la collaboration
 - D) Réserver quelques mois pour commencer à travailler à la consolidation d'équipe et pour mieux apprendre à se connaître les uns les autres
-
- A) Incorrect. Il peut être recommandé d'agrandir l'équipe lorsqu'il manque l'expertise nécessaire pour faire ce qui doit être fait. Mais l'embauche de personnel chargé de tâches non techniques n'est ni conseillée ni désirable pour préserver l'efficacité de l'équipe. Toutefois, dans ce cas, sachant que l'équipe est suffisamment productive, cela ne fera qu'augmenter les problèmes que vous rencontrez actuellement car il n'y a ni confiance ni collaboration.
 - B) Incorrect. C'est une très bonne idée mais cela ne résoudra pas le manque de collaboration et d'affinité dont votre entreprise souffre et qui fait partie de la mentalité DevOps. La collaboration augmente du fait de partager un objectif et de réussir à l'atteindre ensemble.
 - C) Correct. C'est une bonne manière de résoudre les problèmes actuels. Cela devrait réduire les bagarres, prendra peu de temps à résoudre le problème et évitera le risque de perdre la commande. Les objectifs partagés contribuent à la confiance, aux affinités et à la collaboration. (*Ouvrage : A, Chapitre 7*)
 - D) Incorrect. Cela pourrait fonctionner et permettre d'établir une équipe dont les membres se font confiance et collaborent, ce qui devrait être votre objectif. Ce n'est toutefois pas la meilleure manière de résoudre le problème, considérant que vous risquez de perdre la commande en prenant trop de temps à résoudre le problème.

8 / 50

Vous estimez que votre équipe de Développement est réellement une équipe.

Quel signe indique avec certitude que c'est une équipe et non un groupe ?

- A) L'équipe suit les règles dont ses membres ont convenu dans leurs réunions d'équipe.
 - B) L'équipe organise des réunions efficaces qu'elle dirige elle-même.
 - C) L'équipe maintient un rythme de travail constant pour atteindre ses objectifs communs.
 - D) L'équipe résout les problèmes en interrogeant le membre de l'équipe responsable.
-
- A) Incorrect. Il peut être très facile pour un groupe de personnes de suivre des règles. Cela n'en fait pas nécessairement une équipe.
 - B) Incorrect. Des groupes de personnes peuvent tenir des réunions très efficaces. Cela n'est pas nécessairement un signe qu'il s'agit d'une équipe.
 - C) Correct. Une véritable équipe conserve un rythme de travail régulier et s'assure de suivre l'objectif commun. (*Ouvrage : A, Chapitre 9*)
 - D) Incorrect. Les membres d'une équipe résolvent les problèmes ensemble et ne commencent pas à interroger un membre. La culture de DevOps rejette le blâme.

9 / 50

AppAtoZ a connu un taux de croissance phénoménal au niveau du développement et du déploiement d'applications mobiles pour les plateformes iPhone et Android.

Les équipes de développement de cette startup ont subi d'énormes pressions en déployant rapidement des améliorations à leurs applications mobiles actuelles selon un échéancier agressif. Elles ont travaillé en moyenne 60 heures par semaine au cours des six derniers mois. L'équipe de direction a été réticente à embaucher du personnel supplémentaire et se préoccupe plus de l'augmentation des revenus tout en réduisant les coûts d'exploitation et de développement.

Au cours des derniers mois, il y a eu une hausse du taux d'absentéisme et du nombre d'employés se mettant en arrêts-maladie. Plusieurs démissions chez AppAtoZ ont entraîné une augmentation de la charge pour ceux qui sont restés. Le recrutement et le temps d'adaptation des nouveaux employés ne parvient pas à alléger rapidement la charge de travail de l'équipe de développement.

Quelles stratégies à **long terme** doivent être envisagées pour résoudre l'épuisement professionnel et le stress des employés ?

- A)
 - Ajouter à l'équipe un mixte d'employés permanents et contractuels afin de mieux gérer la charge de travail
 - Faire en sorte que l'équipe de direction et le Développement identifient les facteurs contribuant à l'épuisement dans l'environnement professionnel
 - Élaborer un plan pour résoudre l'épuisement professionnel et le stress des employés
- B)
 - Licencier l'équipe de direction car elle est incompetente
 - Embaucher une équipe de direction plus compétente qui sait comment créer un équilibre travail-vie et une culture durable et réaliste
 - Donner à l'équipe de développement la possibilité de prendre le temps de trouver son équilibre
- C)
 - Mettre les membres de l'équipe de développement en congés
 - Engager des développeurs pour gérer la charge de travail accrue dans le service du développement pendant les périodes de pointe de l'année
 - Faire en sorte que les développeurs puissent faire appel à une assistance psychologique en fonction de leurs besoins
- D)
 - Mettre les membres de l'équipe de développement en congés
 - Faire en sorte que l'équipe de direction et le Développement identifient tous les facteurs de l'environnement professionnel qui peuvent contribuer à l'épuisement
 - Élaborer un plan pour résoudre l'épuisement professionnel et le stress des employés

- A) Correct. C'est la seule réponse où tous les choix sont à long terme. Le reste des réponses comporte au moins un choix à court terme. (*Ouvrage : A, Chapitre 8*)
- B) Incorrect. Le fait de virer l'équipe de direction ne permet pas de résoudre les problèmes sous-jacents à long terme. Cela peut être une solution à court terme. Il est une bonne idée de prendre le temps de trouver l'équilibre, mais sans planification, cela n'ira pas de soi.
- C) Incorrect. Prendre des congés peut être une bonne idée, mais cela ne fonctionne qu'à court terme. Une assistance psychologique est aussi une excellente idée, mais encore une fois, sans changements sous-jacents, cela reste une solution à court terme. L'embauche de développeurs peut être efficace et c'est une excellente idée.
- D) Incorrect. L'élaboration d'un plan et l'identification des facteurs liés à l'environnement professionnel sont de très bonnes idées. Toutefois, il existe une réponse encore meilleure car le fait de prendre des congés ne fonctionne qu'à court terme.

10 / 50

Vous travaillez en tant que développeur dans une équipe DevOps. Vous tentez de faciliter la collaboration entre tous les membres de votre équipe qui compte deux hommes développeurs seniors et une femme, administratrice système junior.

Dès le début, les membres de votre équipe ont eu des relations difficiles. Vous êtes alors intervenu pour tenter d'améliorer leur collaboration. Et cela a fonctionné car ils ont arrêté de se disputer et se sont mis à travailler plus efficacement.

Maintenant, vous remarquez que l'administratrice junior est systématiquement d'accord avec les développeurs seniors.

Comment devriez-vous réagir à ce mode de fonctionnement ?

- A) Vous devriez vous abstenir d'intervenir tant que les membres de l'équipe ne se disputent pas. Ils semblent s'entendre et il vaut donc mieux ne pas intervenir.
 - B) Il est important qu'Exploitation exprime toute réserve quant aux versions (builds). Vous devriez donc entraîner l'administratrice pour qu'elle cesse d'être accommodante et devienne plus assertive.
 - C) Les développeurs seniors ont plus de responsabilités ; vous devriez donc leur demander d'être plus aimables et de penser aux inquiétudes de l'Exploitation si l'administratrice ne le fait pas.
 - D) Vous devez commencer à travailler pour l'équipe d'Exploitation, pour le moment, même si vous êtes vous-même développeur, afin de pouvoir montrer le chemin à l'administratrice junior.
-
- A) Incorrect. Cette situation nécessite une résolution des conflits. La juniore de l'équipe se montre accommodante pour résoudre les conflits. Or, ce n'est pas un style productif. Chaque rôle doit assumer ses responsabilités afin d'éviter la résolution trop tardive de problèmes potentiels dans la version.
 - B) Correct. C'est la solution adéquate à cette situation. Cette situation nécessite une résolution des conflits. La juniore de l'équipe se montre accommodante pour résoudre les conflits. Or, ce n'est pas un style productif. Par ailleurs, cela entraîne le risque que l'Exploitation n'exprime pas les réserves qu'elle pourrait avoir vis-à-vis de la version. (*Ouvrage : A, Chapitre 7 et 14*)
 - C) Incorrect. Chaque membre assume une part de responsabilité égale. L'ancienneté ou le genre n'ont rien à voir là-dedans. Par ailleurs, les développeurs abordent les problèmes relatifs aux versions différemment des gens d'Exploitation. C'est pourquoi DevOps nécessite l'interaction des différents domaines.
 - D) Incorrect. Vous ne pouvez pas simplement changer d'équipe, même si le fait de montrer la manière de procéder est une excellente méthode de formation. Les développeurs n'abordent pas les problèmes relatifs aux versions de la même manière que l'Exploitation. Procéder ainsi sera donc dommageable à la qualité des versions.

11 / 50

De quelle manière Agile et Scrum promettent-ils un développement de logiciels plus rapide et prévisible ?

- A) En permettant une collecte et un traitement des spécifications meilleurs et plus exhaustifs avant la conception
 - B) En permettant de petites équipes autonomes qui s'autoorganisent et gèrent elles-mêmes leur planning
 - C) En permettant au Propriétaire de produit de participer aux mêlées quotidiennes
 - D) En permettant au Responsable de projet de modifier rapidement des priorités selon les nécessités
-
- A) Incorrect. Cette approche est plutôt basée sur un modèle en cascade.
 - B) Correct. C'est la procédure proposée par les principes Scrum et Agile pour être en mesure de livrer plus rapidement une meilleure qualité. (*Ouvrage : A, Chapitre 4*)
 - C) Incorrect. Même si le Propriétaire de produit peut participer, cela n'a aucune importance quant au respect de la promesse d'un développement logiciel plus rapide et plus prévisible.
 - D) Incorrect. Le Responsable de projet ne devrait pas changer les priorités ; c'est le Propriétaire de produit qui s'en charge.

12 / 50

Qu'est-ce qu'un ITSM allégé ?

- A) Un ITSM centré sur la continuité des opérations
 - B) Une nouvelle version d'ITIL proposée comme une norme
 - C) Une mauvaise mise en œuvre des processus ITIL
 - D) Une gestion des mises en production orientée sur ITSM
-
- A) Correct. ITIL semble trop lourd et non-adapté à la rapidité des processus de DevOps. Un ITSM allégé est un ITSM réaligné pour DevOps, centré sur la continuité des opérations avec un ensemble d'informations minimales requises. (*Ouvrage : C, Chapitre 4iii*)
 - B) Incorrect. Aucune version d'ITIL de ce genre n'a encore été proposée.
 - C) Incorrect. Un ITSM allégé n'est pas une mauvaise mise en œuvre, mais plutôt une version 'écrémée', axée sur la continuité des opérations et la réduction des tâches de gestion.
 - D) Incorrect. ITSM est orienté sur la gestion des services et non sur la gestion des mises en production. Dans le concept ITSM, la mise en production est un processus qui sous-tend le service.

13 / 50

Vous travaillez pour une entreprise qui applique les pratiques Lean et Agile. Votre président directeur général (PDG) n'est pas convaincu de l'intérêt de mettre en œuvre encore une autre méthode : DevOps.

Vous êtes un expert DevOps. Vous avez la conviction que DevOps est un atout pour toute entreprise qui développe des logiciels.

Qu'est-ce que DevOps peut apporter à l'entreprise pour laquelle vous travaillez ?

- A) Créer des Récits Utilisateur et des exigences fonctionnelles de meilleure qualité
 - B) Réduire la cadence des mises en production pour que les clients souffrent moins des impacts des mises à jour
 - C) Former les professionnels à la fois au Développement et à l'Exploitation
 - D) Écrire des codes et les mettre plus rapidement en production, tout en facilitant le résultat du business
-
- A) Incorrect. Les pratiques Agile vous permettent déjà d'écrire des Récits Utilisateur et des exigences fonctionnelles de grande qualité. DevOps adhère aux principes Agile, ces derniers en constituant la base. Même si vous impliquez l'Exploitation dès le début, les Récits Utilisateur et les exigences fonctionnelles créés en suivant Agile ne changeront pas grand-chose dans DevOps.
 - B) Incorrect. DevOps utilise des mises en production continues et plus rapides, pour apporter plus rapidement de la valeur à l'activité. C'est aussi un principe de Lean. Les mises à jour fréquentes ne sont pas un problème pour les clients dans la mesure où les nouvelles fonctionnalités apportent réellement de la valeur ajoutée. Par ailleurs, il est possible de procéder à des mises en production fréquentes et de faire en sorte que les utilisateurs procèdent aux mises à jour à des moments planifiés.
 - C) Incorrect. DevOps a pour objectif de faire collaborer l'Exploitation et le Développement et non de faire assumer ces deux rôles par un même individu. Même si cela est théoriquement possible dans les toutes petites entreprises, cela ne correspond pas rigoureusement à DevOps.
 - D) Correct. Les pratiques Lean et Agile concentrent déjà l'attention sur les valeurs métier et les demandes de fonctionnalités variables. DevOps y ajoute une fréquence plus élevée de mise en production, épaulant ainsi directement les résultats du business, en se concentrant sur la création d'un pipeline de déploiement continu avec l'implication précoce d'Exploitation et une automatisation aussi poussée que possible. (*Ouvrage : C, Chapitre 1*)

14 / 50

La Direction demande au service informatique un meilleur soutien au business et un alignement sur les objectifs de ce dernier. En votre qualité de Directeur des Systèmes d'Information (CTO), vous avez décidé, entre autres mesures, de réduire la charge de travail supportée par Exploitation.

De quelle manière est-ce que DevOps pourra le **mieux** vous aider à améliorer la gestion des niveaux de service (SLM) existante ?

- A) En abandonnant les bonnes pratiques ITSM car cette approche est trop lourde
 - B) En élaborant de meilleurs accords sur les niveaux opérationnels (OLA) entre Dev et Ops
 - C) En mettant en œuvre de nouveaux processus ITSM basés sur la version 3 d'ITIL
 - D) En réalignant la gestion des services informatique (ITSM) pour utiliser un ensemble d'informations minimales requises (MRI)
-
- A) Incorrect. L'abandon des bonnes pratiques ITSM n'aura pas pour effet d'améliorer SLM.
 - B) Incorrect. L'élaboration de meilleurs OLA entre Dev et Ops ne contribuera pas à réduire la charge de travail, elle aura plutôt tendance à augmenter la bureaucratie. Ce n'est certainement pas un point sur lequel DevOps se concentre.
 - C) Incorrect. Cela entraînera un plus grand nombre de processus lourds inadaptés à DevOps.
 - D) Correct. L'instauration d'une gestion des services informatiques légère est un prérequis incontournable à la mise en œuvre de DevOps. (*Ouvrage : C, Chapitre 4iii*)

15 / 50

Quelle mise en œuvre de DevOps est la **plus** adaptée à une entreprise qui utilise l'approche système d'enregistrement (SoR) ?

- A) Collaboration
 - B) Livraison continue
 - C) La façon Toyota
-
- A) Correct. Elle se concentre uniquement sur la fourniture de services informatiques fréquents et rapides et sur une Exploitation fiable, ce qui est le plus adapté au système d'engagement (SoE) et au système d'enregistrement (SoR). (*Ouvrage : C, Chapitre 8*)
 - B) Incorrect. Le plus adapté aux fournisseurs de produits numériques.
 - C) Incorrect. Le plus adapté aux prestataires de services informatiques.

16 / 50

Quel est le **principal** avantage de l'utilisation du système Obeya ?

- A) Facilite l'émission de plaintes client, afin de s'assurer que l'équipe reçoit suffisamment de feedback pour s'améliorer continuellement
 - B) S'attaque au stress au sein des équipes, de manière à ce que les membres de l'équipe puissent tenir une cadence durable
 - C) Améliore la signalisation quotidienne des bogues, ce qui réduit les remaniements ainsi que les transferts de bogues à d'autres terminaux
 - D) Une prise de décision rapide basée sur le statut actuel, grâce à une collecte et un partage d'information rapides
-
- A) Incorrect. Obeya n'a pas pour vocation de permettre aux clients de se plaindre.
 - B) Incorrect. Même s'il peut permettre de mieux gérer les situations stressantes, ce n'est pas son principal atout.
 - C) Incorrect. Obeya ne permet pas d'améliorer la signalisation des bogues.
 - D) Correct. Obeya room ou War room est un système de production Toyota / un outil Lean qui permet aux équipes de projet de disposer de toutes les informations pertinentes et promeut également une interaction et un partage d'informations rapides entre de petites équipes, accélérant ainsi la collecte d'informations en vue des prises de décision. (*Ouvrage: C, Chapitre 7iii*)

17 / 50

Vous travaillez dans une entreprise informatique qui vient juste d'adopter DevOps. Vous vous rendez compte que non seulement la culture de l'entreprise doit changer mais également ses pratiques et ses outils. L'entreprise surveille déjà les projets logiciels pour détecter tout problème.

Vous proposez d'ajouter les mesures suivantes :

- automatiser le processus de gestion des changements
- mettre en œuvre un contrôle des accès pour prévenir de tout changement non autorisé

Pourquoi cela est-il nécessaire ?

- A)** L'automatisation permet la mise en œuvre plus rapide des changements, tout en conservant la confiance. Le contrôle d'accès est nécessaire pour éviter une résolution de problèmes heuristique et des interruptions de service non planifiées.
- B)** L'automatisation est nécessaire pour réduire le nombre de changements. Le contrôle d'accès devrait éviter que les clients modifient eux-mêmes le logiciel, sans que vous en soyez informé et sans que vous puissiez contrôler la situation.
- C)** L'automatisation contribuera à rendre l'équipe d'Exploitation inutile. Le contrôle d'accès doit être mis en œuvre parce que les projets DevOps fonctionnent dans le Cloud, ce qui implique des risques supplémentaires.
- A)** Correct. Citation : « Si, en général, nous ne sommes pas favorables au verrouillage et à la mise en place de processus d'approbation, nous estimons que cela est essentiel lorsqu'il s'agit de votre infrastructure de production. En conséquence, comme nous estimons que l'on doit traiter nos environnements de test de la même manière que nos environnements de production, le même processus devrait être appliqué à ces deux environnements. Il est essentiel de verrouiller les environnements de production afin d'éviter tout accès non autorisé, non seulement des personnes externes à l'organisation, mais également du personnel faisant partie de votre organisation, et même des membres d'Exploitation. Dans le cas contraire, en cas de problème, il est trop tentant de se connecter à l'environnement en question et de bricoler à droite, à gauche, pour résoudre les problèmes (un processus parfois appelé pudiquement résolution de problème heuristique). C'est généralement une très mauvaise idée, et cela pour deux raisons. Premièrement, cela mène généralement à des interruptions de service (on a tendance à essayer, au hasard, un redémarrage ou des service packs). Deuxièmement, si un problème se produit ultérieurement, il n'y aura aucun enregistrement de ce qui a été fait, par qui, et à quel moment, ce qui veut dire qu'il est impossible d'identifier la cause du problème auquel vous êtes confronté. Dans une telle situation, il peut être préférable de recréer l'environnement à partir de zéro de manière à savoir à quoi vous en tenir. »
(Ouvrage : B, Chapitre 11)
- B)** Incorrect. L'automatisation contribue à accélérer le nombre de changements possibles sans perdre confiance dans le processus. Les contrôles d'accès peuvent tenir les clients à distance, mais ce n'est pas l'objectif principal.
- C)** Incorrect. Exploitation n'est jamais inutile. Toutefois, cette équipe peut se sentir moins chargée et moins ennuyée. Les projets DevOps peuvent se trouver dans le Cloud, mais pas nécessairement. Et quand bien même le contrôle d'accès contribue à sécuriser un logiciel dans le Cloud, cela n'en constitue pas l'objectif principal.

18 / 50

La virtualisation et le Cloud Computing sont des techniques susceptibles d'aider et de faciliter les pratiques DevOps.

De quelle manière aident-elles DevOps ?

- A)** Le Cloud Computing rend l'accès à distance possible, ce qui donne aux clients plus de contrôle.
 - B)** Des environnements virtuels sont plus faciles à standardiser et utilisent le matériel plus efficacement.
 - C)** Une infrastructure virtuelle est facile à comprendre et ne nécessite aucune maintenance.
-
- A)** Incorrect. Il est vrai que le Cloud Computing facilite un accès à distance, mais en soi, cela n'implique pas que les clients auront plus de contrôle. Ces deux choses n'aident aucunement DevOps.
 - B)** Correct. La virtualisation permet de fusionner facilement l'intégration continue et l'infrastructure de test, de manière à pouvoir les proposer comme un seul et unique service aux équipes de livraison. Cela est également plus efficace en termes d'utilisation du matériel. La virtualisation permet la standardisation sur une configuration matérielle unique des environnements physiques, tout en exécutant virtuellement des environnements et des plateformes d'une grande variété. (*Ouvrage : B, Chapitre 11*)
 - C)** Incorrect. Une infrastructure rendue virtuelle n'est pas forcément plus facile à comprendre. Par ailleurs, elle a également besoin de gestion et d'entretien. Toutefois, pas au même degré que sur une infrastructure sur site. Il ne s'agit donc pas d'une raison valable.

19 / 50

Quel que soit le projet, les accords sur les niveaux de service (ANS) sont importants car ils indiquent ce dont vous convenez avec le client. Toutefois, dans DevOps, le ANS a un autre objectif important.

Quel est cet objectif ?

- A)** Les clients sont responsables de l'élaboration d'un ANS pour l'équipe DevOps. En cela, il remplace une phase formelle parmi les responsabilités d'Exploitation.
 - B)** Dans le ANS, le client peut indiquer toutes ses exigences non fonctionnelles, de manière à ce que le Développement puisse se concentrer sur ces dernières.
 - C)** Le ANS indique le niveau de service acceptable. Le Développement doit être en mesure de comprendre le ANS et à aider l'Exploitation à les respecter.
-
- A)** Incorrect. Un ANS est toujours un contrat entre le client et le domaine qui fournit le service. En conséquence chacune des parties a voix au chapitre quant au contenu de ce document.
 - B)** Incorrect. Certaines des exigences fonctionnelles et non fonctionnelles peuvent être issues d'ANS mais la majorité d'entre elles sont communiquées directement à l'équipe DevOps sans passer par un ANS. De plus, la raison d'être d'un ANS n'est pas de fournir des exigences non fonctionnelles.
 - C)** Correct. Généralement, les conditions décrites dans un ANS sont de la plus grande importance pour l'Exploitation. Le Développement devrait épauler l'Exploitation en rendant son travail le plus facile possible. C'est ce que distingue DevOps d'un développement habituel. (*Ouvrage : B, Chapitre 12*)

20 / 50

Votre équipe doit développer un pipeline de déploiement pour un nouveau produit. Dans le cadre de l'intégration continue, vous devez définir l'étape de commissionnement de code dans le pipeline. Vous discutez de cette étape avec les membres de votre équipe.

Le Maître de processus dit : « La définition d'achevé (Definition of Done) doit être définie pendant ou après l'étape de commissionnement. Si le code n'est pas achevé lorsque les modifications sont enregistrées dans la version centrale du programme, il faut arrêter le travail. »

Est-ce vrai ?

- A) Oui. Si le travail n'est pas achevé, le Maître de processus ne fait pas du bon travail. Ce point devrait être résolu immédiatement.
 - B) Oui. Un travail qui n'est pas achevé ne devrait pas être enregistré dans la base d'intégration continue, car il n'a aucune valeur pour le client.
 - C) Non. La définition d'achevé a lieu lors des réunions avec le client. Cela ralentirait trop le travail d'attendre une telle réunion.
 - D) Non. Le travail dans un pipeline de déploiement ne doit jamais s'arrêter. Si un code n'est pas achevé, il doit juste être inactif.
-
- A) Incorrect. Il incombe au Maître de processus de s'assurer qu'il existe une définition d'achevé. Lorsqu'un code inachevé est validé, les travaux doivent cesser. Cependant, le Maître de processus ne fait pas nécessairement un mauvais travail lorsqu'un code inachevé est envoyé.
 - B) Correct. Lorsqu'une tâche n'est pas achevée, elle ne représente pas une valeur suffisante pour le client pour la mettre en pipeline de déploiement. Compte tenu du flux pièce par pièce (one-piece-flow), cela retarderait le flux de tâches plus utiles. (*Ouvrage : B, Chapitre 3*)
 - C) Incorrect. La définition d'achevé est l'une des premières choses qui est convenue dans le cadre d'un projet. Cela n'est pas défini pendant les réunions avec le client. Lors du début du codage, une définition d'achevé devrait déjà être connue. Sinon, comment savoir lorsqu'il faut s'arrêter de coder ?
 - D) Incorrect. S'il y a quelque chose qui cloche dans le code ou s'il n'ajoute pas de valeur, cela constitue une raison suffisante pour arrêter le pipeline de déploiement et réparer le code, ou pour lancer quelque chose d'une plus grande valeur dans le pipeline du flux pièce par pièce.

21 / 50

Quel est le **principal** avantage d'accroître la diversité de l'équipe pour inclure un éventail plus large de situations personnelles et de cultures ?

- A) Il enrichit l'équipe d'un plus grand nombre d'expériences et de points de vue.
 - B) Il conduit à une diminution des frictions entre les membres de l'équipe.
 - C) Il limite l'originalité et la capacité à trouver de nouvelles idées.
 - D) Il prend plus de temps pour prendre une décision spécifique.
-
- A) Correct. La diversité culturelle couvre une grande variété de bagages, dont des aspects relatifs à la race, le sexe, la sexualité, la classe sociale, le niveau d'éducation, la langue, et la quantité d'expérience professionnelle. Tous ces aspects uniques enrichissent une organisation d'un plus grand nombre d'expériences et de points de vue. (*Ouvrage : A, Chapitre 7*)
 - B) Incorrect. Il est possible que l'augmentation de la diversité puisse conduire à une augmentation de la pression et des frictions, parce que différentes valeurs culturelles doivent collaborer.
 - C) Incorrect. Une diversité accrue signifie un plus grand nombre de points de vue différents. Habituellement, cela conduit à plus d'originalité.
 - D) Incorrect. Cela est habituellement considéré comme un inconvénient. (Bien que la lenteur d'un processus de décision puisse être bénéfique.) Une plus grande diversité pourrait conduire à des discussions plus longues pour parvenir à un consensus.

22 / 50

Vous travaillez pour une petite entreprise qui dispose d'une seule équipe DevOps. Cette dernière travaille à une application qui est constitué de plusieurs composants. Certains sont nouveaux et d'autres nécessitent juste une mise à jour.

À l'heure actuelle, chaque composant dispose de son propre pipeline de déploiement. L'équipe est fière de la livraison continue qu'elle réalise dans la grande majorité des pipelines de déploiement. Sa production est également élevée et de bonne qualité.

Que devriez-vous faire ?

- A) Conserver les différents pipelines et encourager l'équipe à accroître sa livraison continue
 - B) Conserver uniquement les pipelines présentant une livraison continue et fusionner les autres pipelines sans livraison continue
 - C) Conserver uniquement deux pipelines : l'un pour le développement et l'autre pour la maintenance
 - D) Viser un seul et unique pipeline en collaboration avec votre équipe, en expliquant les risques de multiples pipelines
- A) Incorrect. Le fait de disposer de plus d'un pipeline de déploiement par équipe comporte des risques. Il est difficile d'établir la priorité entre les différents pipelines. Les travaux sur certains éléments devront être suspendus si quelque chose dans un autre pipeline reçoit la priorité. Cela réintroduit un certain niveau de chaos et de multitasking que DevOps tente justement d'éviter. L'utilisation d'un unique pipeline de déploiement augmente l'efficacité et la valeur métier.
- B) Incorrect. N'importe quel pipeline de déploiement bénéficie grandement de la livraison continue. Toutefois, la mise en pratique de la livraison continue ne réduit pas les risques afférents aux multiples pipelines de déploiement.
- C) Incorrect. Les mêmes risques demeurent, même si vous n'avez que deux pipelines de déploiement et même si leur fonctionnement est séparé.
- D) Correct. C'est ce qu'il faut faire. Normalement, votre équipe devrait avoir tiré suffisamment d'enseignements de sa pratique de la livraison continue pour l'intégrer à l'ensemble du pipeline.
(Ouvrage : B, Chapitre 13)

23 / 50

Vous êtes en train d'évaluer Company Builders, une organisation de moyenne à grande taille qui a adopté les pratiques DevOps il y a quelques années.

L'entreprise vous a engagé afin de déterminer son niveau actuel de maturité. Lorsque vous aurez terminé, on attend de vous que vous suggériez des améliorations. L'entreprise veut connaître le domaine sur lequel elle doit se concentrer pour atteindre le niveau de maturité suivant : *Niveau 2 - Gestion quantitative*.

D'après vos investigations, la majorité des domaines se situe au *Niveau 1 - Cohérent*, à deux exceptions près :

- A. **Environnements et déploiement.** Ce domaine gère les déploiements orchestrés et a testé les processus de mise en production et de restauration.
- B. **Gestion des versions et intégration continue.** Dans ce domaine, vous trouvez des développements et des tests réguliers automatisés, et que toute version peut être recrée à partir du contrôle de source via un processus automatisé.

Déterminez le niveau de maturité de ces deux domaines, d'après les informations données. Ensuite, donnez votre recommandation par rapport au point d'amélioration.

Sur lequel de ces deux domaines, Company Builders devrait travailler, avant de passer au niveau 2 ?

- A) Environnements et déploiement d'une part, et gestion des versions et intégration continue, d'autre part, se trouvent tous les deux au niveau 0. Les efforts doivent porter simultanément sur les deux environnements.
 - B) Environnements et déploiement d'une part, et gestion des versions et intégration continue, d'autre part, sont soit au niveau 1, soit au-dessus de ce niveau. Les efforts doivent porter sur les autres domaines afin de progresser.
 - C) Environnements et déploiement se trouve au niveau 0. Gestion des versions et intégration continue se trouve au niveau 1. L'attention doit tout d'abord porter sur environnement et déploiement.
 - D) Environnements et déploiement se trouve au niveau 2. Gestion des versions et intégration continue se trouve au niveau 0. L'attention devrait uniquement porter sur gestion des versions et intégration continue.
-
- A) Incorrect. Le domaine 1 est au niveau 2 et le domaine 2 est au niveau 0.
 - B) Incorrect. Le domaine 1 est au niveau 2 et le domaine 2 est au niveau 0.
 - C) Incorrect. Environnements et déploiement se trouve déjà au niveau 2 et n'a pas besoin de plus de travail.
 - D) Correct. Le domaine 2 est au niveau 0 et il devrait d'abord mûrir au niveau 1 avant que l'organisation n'envisage de le faire passer au niveau 2. (*Ouvrage : B, Chapitre 15*)

24 / 50

Votre entreprise commercialise un service de sauvegarde en ligne. Un de vos clients demande d'ajouter de nouvelles fonctionnalités à votre service. Il veut ces nouvelles fonctionnalités dans un délai d'une semaine sinon il ira voir ailleurs.

Vous estimez que les nouvelles fonctionnalités sont importantes et vous savez que l'équipe de Développement peut les produire rapidement. Toutefois, vous rencontrez les problèmes suivants :

- Les testeurs ont besoin de beaucoup de temps pour régler les bogues.
- Les testeurs trouvent des bogues que les développeurs ont résolu il y a longtemps.
- Il est rarement possible de faire une démonstration de l'application car cette dernière fonctionne rarement.
- Les présentations se font rares.

Quel est votre problème et comment devriez-vous le résoudre ?

- A)** Vous déployez des codes défectueux. Vous devriez résoudre cette situation en augmentant la compréhension du processus de déploiement, en augmentant la collaboration et en travaillant avec plus de discipline.
- B)** Votre gestion des configurations laisse à désirer. Vous devriez résoudre ce problème en augmentant la collaboration entre Développement et Exploitation, ainsi qu'en augmentant la surveillance, la journalisation et la virtualisation.
- C)** Votre processus d'intégration continue n'est pas géré correctement. Vous devriez résoudre ce problème en augmentant la rapidité des tests automatisés et la rapidité de l'étape de commissionnement ainsi qu'en augmentant la compréhension du processus d'intégration continue.
- D)** Votre stratégie de test n'est pas efficace. Vous devriez résoudre ce problème en automatisant les tests et en augmentant la collaboration entre les testeurs et le reste de l'équipe.
- A)** Correct. Le déploiement de code défectueux ou en quantité insuffisante mène à de longs délais de déploiement, une faible vélocité, du scepticisme quant aux dates de mise en production, une perte de confiance dans l'environnement d'intégration continue, une augmentation des délais de résolution des bogues, l'identification de bogues que les développeurs ont résolue il y a longtemps, et des démonstrations et présentations peu nombreuses. La solution fournie résout ces problèmes.
(Ouvrage : B, Chapitre 15)
- B)** Incorrect. Une mauvaise gestion de configuration mène particulièrement à des échecs de production inexplicables, des déploiements ingérables, des délais de configuration de l'environnement plus longs et un délai de rétablissement plus long en cas de panne. La solution fournie résout ces problèmes.
- C)** Incorrect. Une gestion incorrecte du processus d'intégration continue engendre particulièrement une cadence de déploiement inférieure à une par jour, une interruption de l'étape de validation, et de longues phases d'intégration entre les mises en production. La solution fournie résout ces problèmes.
- D)** Incorrect. Une stratégie de testing inefficace débouche sur des bogues récurrents, de nombreuses heures passées à résoudre les bogues, de nombreuses plaintes des clients, un produit de faible qualité et des développeurs stressés. La solution fournie résout ces problèmes.

25 / 50

Quel est l'avantage de l'intégration continue pour un DevOps efficace ?

- A) Des cycles de tests prolongés avant la mise en production de fonctionnalités
 - B) De longues périodes de temps entre les mises en production de fonctionnalités
 - C) Des mises en production de fonctionnalités plus fréquentes et en temps opportun
- A) Incorrect. Des pratiques de test saines sont nécessaires, mais elles n'ont pas besoin de suivre des cycles poussés. Par ailleurs, ce n'est pas un avantage.
- B) Incorrect. C'est le contraire de ce qui se produit.
- C) Correct. L'intégration continue contribue à une mise en production plus rapide et plus fréquente parce que l'intégration est automatisée. (*Ouvrage : B, Chapitre 3*)

26 / 50

Une entreprise multinationale est confrontée à de nombreux défis dans la consolidation des envois de commissionnement à partir de sites locaux vers le référentiel central basé à Dallas, Texas. Les sites concernés se trouvent à Mexico, Paris, San Diego et en Angleterre. Le mode d'envoi des modifications varie d'une région à l'autre et parfois, il n'est pas évident de déterminer clairement si certains tests ont échoué ou non.

Il y a quatre approches possibles :

1. Ne pas envoyer une version défectueuse
2. Toujours procéder à tous les tests locaux avant l'envoi, ou s'assurer que le serveur d'intégration continue à le faire
3. Attendre que les tests de commissionnement soient satisfaisants avant de poursuivre
4. Ne pas commenter les échecs aux tests

Parmi ces approches, lesquelles sont les **plus** pertinentes afin de s'assurer que les différents sites traitent leurs difficultés actuelles ?

- A) 1 & 2
 - B) 1, 2 & 3
 - C) 2 & 3
 - D) 2, 3 & 4
- A) Incorrect. 1 n'est pas applicable ici. 3 & 4 sont également importants.
- B) Incorrect. 1 n'est pas important.
- C) Incorrect. 4 est également important.
- D) Correct. Chacune de ces trois approches est particulièrement applicable au scénario actuel. Il n'y a aucune preuve que les versions défectueuses sont archivées, donc cela n'est pas applicable. (*Ouvrage : B, Chapitre 3*)

27 / 50

Examinons la constitution d'un pipeline de déploiement de base.

Quelle étape détermine que le système fonctionne au niveau fonctionnel et non-fonctionnel ?

- A) Le test de validation automatisé
 - B) Le test unitaire et test d'assemblage
 - C) Le test de validation manuel
 - D) Le contrôle de version
- A) Correct. La phase de test de validation automatisé détermine que le système fonctionne au niveau fonctionnel et non-fonctionnel, et que son comportement répond aux besoins des utilisateurs et aux spécifications du client. (*Ouvrage : B, Chapitre 8*)
- B) Incorrect. Le test de version (assemblage) et le test unitaire s'assurent que le nouveau morceau de code est sain. Ils ne vérifient pas l'intégration à la version actuelle.
- C) Incorrect. Cela pourrait être une bonne réponse. Cependant, le test de validation est censé être automatisé dans un pipeline de déploiement en fonctionnement.
- D) Incorrect. Le contrôle de version est utilisé pour réparer des versions défectueuses et résoudre des problèmes. Il ne sert pas à démontrer que le système fonctionne à un niveau fonctionnel ou non-fonctionnel.

28 / 50

Le fait d'utiliser un même processus de déploiement pour chaque environnement dans lequel votre application fonctionne est une bonne pratique de DevOps. Cela assure que la version est testée efficacement. Vous utilisez des scripts pour automatiser votre processus de construction et de déploiement.

Quelle est la **meilleure** manière de le faire ?

- A) Utiliser un script pour chaque environnement et l'entretenir dans le cadre du système de contrôle des versions.
 - B) Utiliser un script spécifique pour chaque environnement afin de tenir compte des différences entre les environnements.
 - C) Utiliser le même script pour chaque environnement, en optant pour des paramètres manuels pour des configurations spécifiques.
 - D) Déployer un même script dans chaque environnement et gérer séparément les informations de configuration.
- A) Incorrect. Cette réponse est un mauvais choix de par la quantité d'effort d'entretien et par les erreurs potentielles que vous risquez d'introduire en ajoutant à la complexité.
- B) Incorrect. La diversité des scripts peut entraîner des problèmes au niveau des modifications, et donc engendrer, au sein du processus, des problèmes qui sont difficiles à détecter et à résoudre.
- C) Incorrect. Aucune interaction manuelle ne doit avoir lieu lors de la construction et du déploiement, car ce processus devrait être automatisé, tant pour la rapidité et pour le rendre exempt d'erreurs.
- D) Correct. Les scripts doivent être identiques pour assurer un test efficace de la version et du processus de livraison. Les différences entre environnements, par ex. URI, IP, etc., doivent être gérées dans le cadre du processus de gestion des configurations. (*Ouvrage : B, Chapitre 6*)

29 / 50

Lors de la mise en production d'un nouveau service informatique, une tâche en cours s'arrête inopinément.

Qu'est-ce qui ne peut **pas** constituer une cause valable de cet événement ?

- A) Le Gardien n'a pas tenu compte de l'interrelation entre les éléments de travail et des critères d'acceptation de services (SAC).
- B) Les exigences non fonctionnelles n'étaient pas claires à cause d'un Récit Utilisateur imparfait.
- C) Le Maître de service n'est pas parvenu à se mettre d'accord avec l'utilisateur sur la fin de vie du service à mettre en production.
- D) Les processus de développement n'ont pas bénéficié d'autant de travail que nécessaire, comme l'indiquent les critères d'acceptation de services.

- A) Incorrect. C'est une cause possible.
- B) Incorrect. C'est une cause possible.
- C) Correct. Ce n'est pas une cause possible car il est impossible de déterminer cela par avance. Cela doit toujours être déterminé lorsque le service est en production. (*Ouvrage : C, Chapitre 7*)
- D) Incorrect. C'est une cause possible.

30 / 50

La société AppBC utilise DevOps. Elle a mis en œuvre le déploiement continu ainsi qu'un solide pipeline de déploiement comportant des tests de validation hautement automatisés et fournit chaque jour de nouveaux logiciels à la production.

AppBC possède une grande base de données et de nombreux utilisateurs. Elle applique une stratégie de test de performance solide et détaillée. Son environnement étant très vaste et complexe, chaque nouvelle version introduit quelques bugs en production.

Quelle stratégie est la **plus** susceptible d'aider AppBC à éviter ces bogues ?

- A) Tester à petite échelle (Canary Releasing)
- B) Automatiser les tests de performance
- C) Réduire le taux de livraison
- D) Utiliser le déploiement en mode bleu-vert

- A) Correct. Le test à petite échelle implique le déploiement d'une nouvelle version d'une application dans un sous-ensemble de serveurs de production afin d'obtenir rapidement un retour. Cela permet de découvrir rapidement tout problème avec la nouvelle version, sans impact sur la majorité des utilisateurs par la montée en puissance graduelle de la charge, tout en mesurant les temps de réponse et d'autres métriques de performance, réduisant le risque d'une nouvelle version et contribuant à trouver et corriger les bogues plus rapidement. (*Ouvrage : B, Chapitre 10*)
- B) Incorrect. Dans ce contexte, les tests de capacité devraient déjà être automatisés, mais l'automatisation de ces tests ne va pas aider à détecter les bogues dans ce scénario.
- C) Incorrect. Cela est contraire aux pratiques DevOps.
- D) Incorrect. Bleu-vert exigera trop de ressources et cela sera très coûteux dans ce scénario. De plus, l'utilisation de cette stratégie avec une grande base de données pourrait engendrer des interruptions ou des situations en lecture seule si un retour en arrière est nécessaire. En outre, elle ne contribue pas à l'élaboration d'un meilleur test des capacités.

31 / 50

DevOps reprend de très importants concepts d'Agile issus du système de production Toyota.

Pourquoi est-ce qu'un flux pièce par pièce (one-piece-flow) est important pour l'adoption de DevOps ?

- A) Cela permet à votre équipe de travailler à une cadence durable avec une vitesse prévisible.
 - B) Cela aide l'équipe à se concentrer sur l'ajout de fonctionnalités dont la valeur métier est la plus importante possible.
 - C) Cela augmente le partage de responsabilité du travail entre les différents membres de l'équipe.
 - D) Cela réduit les goulets d'étranglement en limitant le nombre de tâches effectuées simultanément par une personne.
-
- A) Incorrect. C'est ce à quoi sert la cadence. En établissant une cadence soutenable, vous vous assurez que votre vitesse reste prévisible, que votre équipe ne s'épuise pas et que vos employés conservent un bon équilibre entre le travail et la vie privée.
 - B) Correct. C'est ce à quoi sert le flux pièce par pièce. Un flux pièce par pièce vous permet de sélectionner, ou de mettre à jour, la fonctionnalité qui fournit le plus de valeur et la met ensuite dans le pipeline. Cela préserve votre agilité. En travaillant sur une seule fonctionnalité, vous pouvez mieux limiter le Travail en Cours (Work in Progress ; WiP) de manière à pouvoir effectivement mener les fonctionnalités à terme. (*Ouvrage : C, Chapitre 7*)
 - C) Incorrect. Cela est important pour DevOps, mais n'a pas de lien direct avec Travail en Cours, flux pièce par pièce, cadence ou Ji-Kotei-Kanketsu (JKK).
 - D) Incorrect. C'est exactement la raison d'être de Travail en Cours. En limitant le nombre de tâches sur lesquelles vous travaillez simultanément avec votre équipe, vous parvenez effectivement à terminer dans de bonnes conditions les tâches affectées. Vous évitez les goulets d'étranglement parce que vous régulez la quantité de Travail en Cours de manière à ce qu'il y ait toujours quelqu'un qui attende que le travail soit terminé.

32 / 50

S Corporation est une entreprise de taille moyenne. Elle fournit des pièces détachées automobiles à une grande entreprise T Motors Corporation. La part de T Motors Corporation représente près de 60 % du total des ventes de S Corporation.

Le Comité de Direction se réunit pour discuter du nouveau partenariat. T Motors Corporation exige que S Corporation passe en mode de livraison juste-à-temps (Just-in-Time), sous peine de rompre ses activités commerciales avec S Corporation. La perte de ce client serait fatale pour S Corporation, il y a donc une certaine urgence à adopter le mode juste-à-temps. Ce changement doit se produire dans les 6 mois, donc il y a 5 mois au plus pour se préparer.

La finalité consiste à mettre en œuvre le suivi des pièces par le biais de l'identification par radio fréquence (RFID). Cela devrait assurer la transparence du processus de production. Un examen rapide du processus actuel devrait faciliter le passage à un processus intégrant le système RFID.

La Directrice Système d'Information (CTO) est sollicitée pour piloter le changement. Elle estime cela possible, si une approche DevOps est utilisée pour créer une mise en production optimale. Idéalement, le concept pour la production avec la RFID devrait être développé en premier. Un système de contrôle de la production utilisant les données de la puce RFID devrait être mis en œuvre en fin de parcours. Cependant, il n'y a pas assez de temps pour procéder de manière séquentielle. En conséquence, ces trois étapes doivent se dérouler en même temps.

La CTO affecte Em, un Scrum Master, au projet. Le Développement se prépare à construire un pipeline de déploiement.

Em constate qu'ils sont enthousiastes et travaillent dur, mais qu'ils pourraient être plus disciplinés. En outre, la fréquence des mises en production doit être plus élevée.

Sur quel point Em devrait-il se concentrer **en premier** ?

- A) Em devrait se concentrer sur la communication, puisque c'est la chose la plus importante dans DevOps. Em devrait commencer par briser la glace avec l'équipe et fixer quelques règles de communication.
- B) Em devrait commencer par discuter d'une Cartographie des chaînes de valeur et construire un flux pièce par pièce avec l'équipe, sachant que le flux et la rationalisation des processus sont très importants.
- C) Em devrait commencer par discuter de l'infrastructure et de l'environnement de travail avec les membres de l'équipe, sachant que l'efficacité de DevOps est maximale lorsque tous les outils et les pratiques fonctionnent.
- D) Em devrait commencer par rassembler toutes les parties prenantes, les sensibiliser à DevOps et leur demander leur aide dans le cadre de la diffusion du changement de culture, étant donné que le changement de culture est un prérequis pour DevOps.

- A)** Incorrect. Bien que la communication soit importante pour le fonctionnement de l'équipe DevOps, ils ont déjà travaillé ensemble auparavant. Il y a d'autres solutions plus faciles avec lesquelles commencer. Plus important encore, il faut désormais penser en termes de valeur pour le client. Une fois que cette façon de penser est en place, on peut passer au perfectionnement des compétences de communication.
- B)** Correct. Les processus doivent être rationalisés et la carte des chaînes de valeur doit être mise en place, de sorte que l'équipe peut commencer à ajouter le plus de valeur possible avec une quantité d'efforts minimale. Après cela, il est temps de définir ce qui peut et doit être changé et ce qui fonctionne déjà très bien, sans oublier les outils, la communication et la culture d'entreprise. DevOps peut varier d'une entreprise à l'autre, mais il doit axer l'équipe sur l'ajout de valeur pour le client. *(Ouvrage : B, Chapitre 1 et A, Chapitre 1 et 2)*
- C)** Incorrect. Les outils et l'automatisation sont des éléments importants de DevOps et il ne faut pas les oublier. Cependant, ils ne sont pas nécessairement la première priorité. Au lieu de cela, il est sage de voir si vous pouvez rendre le processus de production ou de services plus facile, plus court et moins cher, en créant une carte des chaînes de valeur et en en discutant avec l'équipe. Cela lance l'inévitable et nécessaire cycle d'amélioration continue. Toutes les autres mesures passent en arrière-plan.
- D)** Incorrect. De nombreuses entreprises doivent opérer un changement de culture. Cependant, ce n'est pas nécessairement la première chose à faire. Il n'est pas non plus nécessaire de rassembler les parties prenantes pour parler de ce changement de culture. Le changement culturel doit découler de la façon par laquelle la plus grande valeur est créée pour les clients ou intervenants. Il peut être intéressant de demander à différentes parties prenantes de jeter un œil à la carte des chaînes de valeur pour voir ce qu'elles peuvent ajouter.

33 / 50

Les membres de votre équipe DevOps travaillent bien ensemble à une cadence soutenue. En intégrant suffisamment de temps mort dans le processus, l'équipe a le temps et la concentration nécessaire pour vérifier et tester minutieusement les versions (builds). À l'heure actuelle, votre équipe procède manuellement aux tests et au déploiement. Sa cadence est suffisamment élevée pour apporter régulièrement une importante valeur métier.

Votre président directeur général (PDG) a sollicité votre conseil sur l'automatisation au sein de cette équipe.

Quel conseil devez-vous donner ?

- A) Automatiser autant que possible, de manière à ce que l'équipe puisse ajouter plus de fonctionnalités et fournir encore plus rapidement de la valeur métier
 - B) Automatiser les tests d'acceptation, mais pas les déploiements, car un processus manuel est plus fiable
 - C) Automatiser les déploiements pour améliorer la durée d'un cycle, mais pas les tests, afin de permettre un apprentissage à partir des bogues
 - D) Ne pas automatiser les méthodes de cette équipe, car les méthodes actuellement appliquées par l'équipe fournissent d'excellents résultats
-
- A) Correct. Toujours automatiser dans la mesure du possible, afin de mieux maîtriser vos processus et accélérer l'obtention d'une valeur métier. (*Ouvrage : B, Chapitres 1 et 8*)
 - B) Incorrect. C'est une excellente idée d'automatiser les tests d'acceptation, mais les déploiements manuels ne sont pas plus sûrs.
 - C) Incorrect. C'est une excellente idée d'automatiser la mise en production, mais un test manuel ne permet pas mieux qu'un test automatisé de tirer des apprentissages des bogues.
 - D) Incorrect. L'équipe fait un travail exceptionnel mais d'une certaine manière vous gâchez son potentiel en n'automatisant pas les activités qui peuvent l'être.

34 / 50

Un Directeur des Systèmes d'Information (CTO) affecte Michael, son employé le plus fiable et qui est également Scrum Master, à un projet. L'équipe de Développement se prépare à construire un pipeline de déploiement.

Michael a confiance dans les bonnes intentions et la spontanéité de l'équipe de développement, mais souhaite que ses membres soient plus disciplinés. De plus, la fréquence des mises en production devrait être plus soutenue. Michael veut que l'équipe de développement mette plus souvent en œuvre des mises en production.

L'un des membres de l'équipe dit : « La chose la plus importante à propos de ce nouveau pipeline de déploiement est son automatisation. Nous devrions d'abord automatiser le pipeline de déploiement ».

Cette déclaration est-elle correcte ?

- A) Oui, c'est correct. L'automatisation du pipeline de déploiement est le plus important facteur d'accroissement de l'efficacité.
 - B) Oui, c'est correct. En vous concentrant sur la création d'un pipeline de déploiement automatisé, vous évitez les éventuels problèmes que vous pouvez rencontrer plus tard.
 - C) Non, ce n'est pas correct. La réalisation d'un flux pièce par pièce (single-piece-flow) et un processus de déploiement solide devraient constituer la première priorité. L'automatisation du processus peut venir plus tard.
 - D) Non, ce n'est pas correct. Au lieu d'automatiser le pipeline de déploiement, le processus de test devrait être automatisé en premier.
-
- A) Incorrect. Le pipeline de déploiement devrait toujours être en premier lieu un pipeline de déploiement d'un flux pièce par pièce. Cela peut bien fonctionner sans aucune automatisation. Une fois que le processus est bien installé, il est possible de l'automatiser, là où cela s'avère possible. Cependant, cela devrait toujours être secondaire à la constitution d'un solide pipeline de déploiement.
 - B) Incorrect. Le pipeline de déploiement devrait toujours être en premier lieu un pipeline de déploiement d'un flux pièce par pièce. Cela peut bien fonctionner sans aucune automatisation. Une fois que le processus est bien installé, il est possible de l'automatiser, là où cela s'avère possible. Cependant, cela devrait toujours être secondaire à la constitution d'un solide pipeline de déploiement.
 - C) Correct. Le pipeline de déploiement devrait toujours être en premier lieu un pipeline de déploiement d'un flux pièce par pièce. Cela peut bien fonctionner sans aucune automatisation. Une fois que le processus est bien installé, il est possible de l'automatiser, là où cela s'avère possible. Cependant, cela devrait toujours être secondaire à la constitution d'un solide pipeline de déploiement. (*Ouvrage : B, Chapitre 5*)
 - D) Incorrect. L'automatisation des tests est une activité clé. Toutefois, si vous devez choisir entre créer un solide pipeline de déploiement et automatiser les tests de travail, vous devez toujours d'abord vous concentrer sur la création d'un solide pipeline de déploiement. Une fois ce dernier en place, il est possible d'augmenter l'efficacité grâce à l'automatisation des tests.

35 / 50

Votre entreprise change ses pratiques et commence à travailler avec DevOps. Votre équipe vous suit dans la mise en œuvre de ce changement. Vous êtes en train de discuter des bonnes pratiques pour l'étape de commissionnement du code.

Votre collègue Sun déclare : « Lorsqu'une version (build) ne fonctionne plus correctement et que personne n'en assume la responsabilité, il faudrait trouver la personne qui a causé ce dysfonctionnement et l'interpeller afin qu'elle le répare. »

Est-ce une bonne idée ?

- A) Oui. Seule la personne qui a causé ce dysfonctionnement peut le résoudre, il faut donc l'identifier, même si cela la met mal à l'aise.
 - B) Oui. Il faut toujours trouver la personne responsable du dysfonctionnement d'une version. Si vous ne le faites pas, vos collègues peuvent appliquer cette règle.
 - C) Non. DevOps est un environnement exempt de blâme. Si un collègue n'assume pas la responsabilité, ne le forcez pas à le faire.
 - D) Non. Il faut d'abord réparer la version. Ensuite, il faut prendre le temps d'identifier la personne responsable et la punir.
-
- A) Incorrect. Il est probablement plus facile de laisser la personne qui crée un problème trouver le problème en question, mais cela n'est pas nécessaire. DevOps est un environnement exempt de blâme. Si un collègue n'assume pas la responsabilité, ne le forcez pas à le faire. Forcer qui que ce soit à faire quoi que ce soit n'est pas respectueux.
 - B) Incorrect. DevOps est un environnement exempt de blâme. Si un collègue n'assume pas la responsabilité, ne le forcez pas à le faire. Forcer qui que ce soit à faire quoi que ce soit n'est pas respectueux.
 - C) Correct. Forcer qui que ce soit à faire quoi que ce soit n'est pas respectueux. Ça peut arriver de faire des erreurs. Les membres de l'équipe travaillent en collaboration afin d'éliminer toutes les erreurs ou résoudre toutes les difficultés. (*Ouvrage : B, Chapitre 3 et A, Chapitre 4*)
 - D) Incorrect. Il n'est pas nécessaire de réparer la version, il suffit de retourner à une version précédente. En outre, la réparation de la version peut-être une bonne idée, mais punir quelqu'un qui fait une erreur est certainement une mauvaise idée. DevOps est un environnement exempt de blâme. Si un collègue n'assume pas la responsabilité, ne le forcez pas à le faire. Forcer qui que ce soit à faire quoi que ce soit n'est pas respectueux.

36 / 50

L'équipe de développement d'X-AppGo est confrontée à de nombreux défis concernant leurs pratiques de test actuelles. Elle utilise, à ce jour, un processus de test de validation manuel. Les développeurs pensent que la suite de tests unitaires qu'ils ont créée est suffisamment méticuleuse pour protéger des régressions.

L'équipe de développement engage de grosses dépenses pour les tests manuels de validation pour chaque version. La haute direction a demandé à ce que l'équipe de développement mette en œuvre des tests de validation automatiques afin de réduire les coûts globaux du testing et de minimiser le nombre de codes défectueux et de régressions introduites dans l'environnement de production.

Quels principes doivent être suivis lors de la définition des critères de validation de votre application en vue d'une automatisation ?

- A) Les principes d'Agile
 - B) Les principes d'ATAM
 - C) Les principes d'INVEST
- A) Incorrect. INVEST est l'ensemble de principes à adopter lors de la création de suites de tests de validation qu'il est possible d'entretenir. Il n'y a aucune information spécifique sur la raison pour laquelle ATAM et Agile ne sont pas recommandés. Agile ne fournit aucune orientation ni aucun principe précis sur l'automatisation des tests.
- B) Incorrect. INVEST est l'ensemble de principes à adopter lors de la création de suites de tests de validation qu'il est possible d'entretenir. Il n'y a aucune information spécifique sur la raison pour laquelle ATAM et Agile ne sont pas recommandés.
- C) Correct. Les tests de validation sont issus de critères de validation. En conséquence, les critères de validation de votre application doivent être rédigés en gardant l'automatisation à l'esprit et ils doivent suivre les principes d'INVEST, qui signifie : indépendant, négociable, précieux, estimable, petit et testable. (Ouvrage : B, Chapitre 8)

37 / 50

Quel est le mécanisme le **plus** efficace pour migrer des données d'une manière automatisée ?

- A) Créer un schéma des versions de base de données et le conserver dans le contrôle des versions
 - B) Créer et gérer des jeux de données plus petits, de manière à faciliter la migration
 - C) S'assurer que tous vos scripts ont été correctement testés avant de procéder à la migration des données
 - D) S'assurer que vous avez une procédure de restauration au cas où la migration tourne mal
- A) Correct. L'établissement de versions de votre base de données est le meilleur mécanisme pour migrer des données d'une manière automatisée. (Ouvrage : B, Chapitre 12)
- B) Incorrect. Cela ne fournit pas le meilleur mécanisme pour favoriser des migrations automatisées et est principalement axé sur une meilleure efficacité de la gestion des jeux de données.
- C) Incorrect. Ce n'est pas la meilleure réponse. Par ailleurs, l'accent porte plus sur les activités de testing que sur les activités de migration.
- D) Incorrect. Ce choix se concentre sur les actions de rétablissement à prendre si la migration échoue.

38 / 50

L'entreprise X-AppGo a rencontré des problèmes avec ses processus de restauration. Cela entraîne souvent des pertes de données sensibles dans les bases de données d'applications de production lorsque des scripts de restaurations sont exécutés.

Dans quelle situation est-il **impossible** d'exécuter des scripts de restauration sans perdre de données sensibles ?

- A) Le script de restauration efface les données que seule la nouvelle version utilise.
 - B) Le script de restauration implique le déplacement d'une colonne d'une table à l'autre.
 - C) Le script de restauration remet en place des données issues de tables temporaires.
- A) Incorrect. Le script de restauration efface uniquement certaines données que seule la nouvelle version utilise et n'entraîne aucune perte de données sensibles lorsque vous procédez à la restauration.
- B) Incorrect. Le script de restauration qui modifie le schéma de la base de données d'une manière qui n'entraîne aucune perte de données.
- C) Correct. C'est un scénario dans lequel un script de restauration ne sera pas possible. (*Ouvrage : B, Chapitre 12*)

39 / 50

ACMECONST a essayé de nombreux dysfonctionnements au niveau des applications et matériels après avoir déployé les logiciels et renouvelé le matériel de leurs routeurs et commutateurs.

Il leur a été très difficile de les rétablir dans leur état d'origine, après avoir observé ces pannes pendant leurs fenêtres de maintenance. Cette situation a entraîné de nombreuses heures de rétablissement, au-delà des fenêtres de maintenance normale et une panne de longue durée de leurs applications vitales.

Un approvisionnement automatisé et une infrastructure autonome peuvent aider dans cette situation, toutefois certaines choses doivent être prises en considération.

Quels éléments doivent être gérés avec précaution afin de réduire le risque d'interruption lors du déploiement de l'environnement de production ?

- A) Des journaux de surveillance détaillés pour résoudre les problèmes en cas d'échec de mise à jour d'une application
 - B) Des points d'intégration externes tels que des systèmes et des services externes
 - C) Configurations de serveurs et informations sur le compte utilisateur sous-jacentes
 - D) L'ensemble d'outils d'automatisation de l'approvisionnement et l'architecture autonome
- A) Incorrect. La résolution de problèmes à lieu après la mise à jour. Ce n'est donc pas un élément valable pour réduire les risques de déploiement vers l'environnement de production.
- B) Correct. Cet élément doit être géré avec précaution afin de réduire le risque des déploiements vers tout environnement semblable à un environnement de production. (*Ouvrage : B, Chapitre 11*)
- C) Incorrect. Cela est couvert par les informations du système d'exploitation et sa configuration tant pour des environnements de test que de production. Ce n'est donc pas un élément que vous gèrerez séparément.
- D) Incorrect. Il n'est pas important de gérer les outils en soi pour réduire le risque. Ils contribuent à fournir un processus de mise en œuvre et un contexte adéquates afin d'obtenir un approvisionnement automatisé et une infrastructure autonome.

40 / 50

L'entreprise X-AppGo a rencontré des problèmes avec son application de base. L'application ne se connecte pas correctement aux autres applications externes. Celles-ci doivent obtenir de manière efficace des variables déterminées, de manière à pouvoir exécuter des appels déterminés. Une équipe est en train de développer l'application fondamentale, et l'entreprise veut conserver cette organisation pour de très bonnes raisons commerciales.

L'un des développeurs suggère de séparer un composant du codebase X-AppGo, pour résoudre les problèmes d'interfaçage.

Qu'est-ce qui constitue une bonne raison de séparer un composant dans ce cas ?

- A)** Convertir un jeu de plug-in dans le codebase en un codebase monolithique
 - B)** Limiter l'impact des changements et faciliter la modification du codebase
 - C)** Le codebase X-AppGo devra être séparé et géré par différentes équipes
 - D)** Il n'y a aucune bonne raison et cela prendra plus de temps pour compiler
-
- A)** Incorrect. La création de composants consiste à passer d'un codebase monolithique à un codebase modulaire basé sur des composants, c'est donc exactement le contraire.
 - B)** Correct. Cela nous incite à concevoir et entretenir un logiciel en délimitant clairement les responsabilités, ce qui, à son tour, limite l'impact des changements et facilite la compréhension du codebase et sa modification. (*Ouvrage : B, Chapitre 13*)
 - C)** Incorrect. Considérant que cela ne sépare pas l'application X-AppGo en des composants indépendants, il n'y a aucune raison de remettre en cause les motifs et de diviser l'équipe.
 - D)** Incorrect. Il y a plusieurs bonnes raisons et l'option numéro 2 en est une. De plus, la création d'un code base modulaire nécessitera moins de temps pour la compilation et ne liera plus le code.

41 / 50

Même les plus petites applications développeront une dépendance à d'autres composants et bibliothèques. En conséquence, la compréhension et la gestion des dépendances est une activité capitale au sein du déploiement continu, afin de préserver le flux dans le pipeline de déploiement.

Vous avez construit une application qui utilise deux bibliothèques. Chacune de ces bibliothèques dépend d'une troisième bibliothèque sous-jacente bien qu'elles réfèrent à des versions différentes. Cela engendre une dépendance spécifique.

Quelle est la **meilleure** solution pour résoudre ou éviter cette dépendance ?

- A) Regrouper toutes les bibliothèques en une seule bibliothèque, de manière à pouvoir directement renvoyer à la bibliothèque et éviter le problème
 - B) Gérer les bibliothèques à l'aide du contrôle des versions, de manière à pouvoir immédiatement voir si vous créez ce type de dépendance
 - C) Conserver un aperçu visuel de toutes vos dépendances sur des Post-It sur un grand tableau, de manière à pouvoir suivre le flux
 - D) Vérifier uniquement de petites portions de votre chaîne de compilation, de manière à pouvoir facilement déboguer les problèmes qui pourraient se produire lors de l'enregistrement
-
- A) Incorrect. Ce n'est pas une bonne idée. Au lieu de cela, vous devriez utiliser le contrôle des versions pour renvoyer à la dernière bibliothèque, ou utiliser un outil d'automatisation qui peut vous aider à automatiser à l'aide de la dernière version des bibliothèques.
 - B) Correct. C'est une bonne solution. De plus, le contrôle des versions sur vos bibliothèques vous aide à résoudre les problèmes causés par l'utilisation d'anciennes versions de votre logiciel. Une autre bonne solution consiste à utiliser un outil d'automatisation pour vous aider à utiliser la dernière version des bibliothèques. (*Ouvrage : B, Chapitre 13*)
 - C) Incorrect. Cela peut aider à suivre le flux, mais ce n'est pas une solution pour traiter les dépendances. Les tableaux Kanban n'offrent pas d'aide particulière dans la mise en œuvre du contrôle des versions.
 - D) Incorrect. Au lieu de cela vous devriez enregistrer l'ensemble de la chaîne de compilation. Cela vous aide à trouver les interdépendances ou les incompatibilités bien plus rapidement et d'une manière beaucoup plus fiable.

42 / 50

Dans un environnement de déploiement continu, il est important que tout soit soumis à un contrôle des versions, de manière à pouvoir trouver rapidement les erreurs ou à procéder à une restauration si nécessaire.

Toutefois, il n'est **pas** recommandé de conserver des fichiers binaires compilés dans le contrôle des versions

Pourquoi fait-on cette exception ?

- A)** Les fichiers binaires compilés tendent à se trouver dans de gros fichiers qui changent avec chaque version (build) et sont mis à jour automatiquement.
 - B)** Plusieurs membres de l'équipe travaillent sur les fichiers binaires, il n'est donc pas pratique de les conserver dans le contrôle des versions.
 - C)** Les fichiers binaires alimentent vos compilateurs, qui sont déjà conservés dans le contrôle des versions.
 - D)** Cela n'est pas nécessaire, considérant que la recompilation est effectuée dans le cadre du processus auquel une version est normalement soumise.
-
- A)** Correct. Tout d'abord, un fichier binaire compilé est très volumineux et recréé pour chaque enregistrement qui est compilé et réussit les tests automatisés. Ensuite, il peut être recréé à partir du code source en exécutant à nouveau le script de la version. (*Ouvrage : B, Chapitre 2*)
 - B)** Incorrect. Ce n'est pas la raison.
 - C)** Incorrect. Les fichiers binaires sont issus des compilateurs, ce n'est pas un élément entrant de ces derniers. Le reste du raisonnement est correct.
 - D)** Incorrect. Il n'est pas conseillé de procéder à la recompilation dans le cadre du processus de version normal. Il est vrai que la recompilation vous donnerait un nouvel élément sortant binaire.

43 / 50

Vous souhaitez adopter une approche holistique pour gérer toutes votre infrastructure informatique.

Quels sont les deux principes sur lesquels il est **préférable** de baser cette approche ?

- A)
 - 1. L'état que vous envisagez pour votre infrastructure devrait être spécifié par le biais d'une configuration contrôlée par les changements.
 - 2. Vous devriez toujours connaître l'état réel de votre infrastructure grâce à la surveillance et à la gestion des événements.
- B)
 - 1. L'état que vous envisagez pour votre infrastructure doit être spécifié par le biais d'une configuration contrôlée par les changements.
 - 2. Vous devriez toujours connaître l'état réel de votre infrastructure grâce aux instruments en place et à la gestion des incidents.
- C)
 - 1. L'état que vous envisagez pour votre infrastructure devrait être spécifié par le biais d'une configuration en version contrôlée.
 - 2. Vous devriez toujours connaître l'état réel grâce à la gestion des incidents en cours et des événements.
- D)
 - 1. L'état que vous envisagez pour votre infrastructure devrait être spécifié par le biais d'une configuration en version contrôlée.
 - 2. Vous devriez toujours connaître l'état réel de votre infrastructure grâce aux instruments et à la surveillance.

- A) Incorrect. L'état que vous envisagez pour votre infrastructure devrait être précisé par le biais d'une configuration contrôlée par des versions et non contrôlée par des changements. Le point 2 est aussi incorrect : la gestion des évènements n'est pas correcte et il manque le principe d'instrumentation.
- B) Incorrect. L'état que vous envisagez pour votre infrastructure devrait être précisé par le biais d'une configuration contrôlée par des versions et non contrôlée par des changements. Le point 2 est aussi incorrect : la gestion des incidents n'est pas correcte et il manque le principe de surveillance.
- C) Incorrect. Le point 1 est correct. Le point 2 n'est pas correct : ce n'est pas l'un des principes sur lesquels il est préférable de baser une approche holistique.
- D) Correct. Ce sont deux des principes sur lesquels il est préférable de baser une approche holistique visant à gérer toutes les infrastructures. (*Ouvrage : B, Chapitre 11*)

44 / 50

Les équipes ayant de bonnes pratiques de collaboration ont des tickets de travail synchronisés. Un Directeur des Systèmes d'Information (CTO) a utilisé « go and see » pour examiner le fonctionnement de l'équipe d'Exploitation. Après la mise en production, l'équipe d'Exploitation redéfinit toujours l'infrastructure d'exploitation.

Quel est le **meilleur** conseil pour améliorer cette pratique ?

- A) Le mieux est de ne rien faire. Aucune amélioration n'est possible car il faut toujours repasser par l'étape de redéfinition.
 - B) Elle devrait envisager une manière de modéliser la structure d'exploitation et de contrôler l'accès à l'environnement d'exploitation.
 - C) Elle devrait procéder à la revue de l'infrastructure d'exploitation, de manière à ce que cela devienne un processus automatisé.
 - D) Elle devrait commencer à partager ses connaissances du processus de déploiement avec l'équipe de Développement.
-
- A) Incorrect. C'est un travail inutile et qui peut être amélioré.
 - B) Incorrect. Bien que cela puisse paraître attrayant, cette opération devra être répétée et il s'agit donc de gaspillage.
 - C) Incorrect. Il n'existe pas de manière évidente de lancer l'automatisation de ce processus sans impliquer l'équipe de Développement.
 - D) Correct. C'est la manière de procéder : partager des connaissances puis prendre des mesures qui s'en suivent. (*Ouvrage : A, Chapitre 17*)

45 / 50

Quel est le moment propice pour que l'Exploitation informe le Développement d'un changement opérationnel ?

- A) Il n'est pas nécessaire d'informer le Développement. Les changements opérationnels concernent uniquement l'équipe d'Exploitation.
 - B) Immédiatement. Le Développement doit être informé dès que possible.
 - C) Lors de la mêlée des mêlées Scrum le matin suivant.
 - D) Lorsque l'équipe d'Exploitation a procédé au test de validation.
-
- A) Incorrect. Le Développement doit être immédiatement informé, de manière à pouvoir anticiper les risques et les problèmes potentiels.
 - B) Correct. Le Développement doit être immédiatement informé, de manière à pouvoir anticiper les risques et les problèmes potentiels. (*Ouvrage : C, Chapitres 5 et 7*)
 - C) Incorrect. Le Développement doit être immédiatement informé, de manière à pouvoir anticiper les risques et les problèmes potentiels.
 - D) Incorrect. Le Développement doit être immédiatement informé, de manière à pouvoir anticiper les risques et les problèmes potentiels.

46 / 50

Vous souhaitez que votre organisation DevOps soit mature. Il y a plusieurs façons d'y arriver.

Qu'est-ce qui ne constitue **pas** une manière d'améliorer la maturité de votre organisation DevOps ?

- A) Définir clairement des objectifs ponctuels pour permettre aux membres de votre équipe de déterminer si leurs activités quotidiennes ont de la valeur.
 - B) Définir clairement les processus, et permettre aux membres de l'équipe d'améliorer quotidiennement le processus.
 - C) Conserver des enregistrements de toutes les réunions, de sorte que les membres de votre équipe aient facilement accès à toutes les communications.
 - D) Surveiller et enregistrer les activités quotidiennes pour contribuer à l'identification des petits domaines de progrès quotidiens et à leur mise en valeur.
-
- A) Incorrect. Cela contribue à la maturation de l'organisation DevOps.
 - B) Incorrect. Cela contribue à la maturation de l'organisation DevOps.
 - C) Correct. Cela ne contribue pas à la maturation de votre organisation DevOps. Il n'est pas strictement nécessaire de consigner par écrit tous les échanges d'une réunion et de les relire. Il est nécessaire d'écrire les accords, mais pas l'ensemble d'une réunion. (*Ouvrage : B, Chapitre 15*)
 - D) Incorrect. Cela contribue à la maturation de l'organisation DevOps.

47 / 50

Vous travaillez pour un prestataire de services informatiques. Dans le cadre de votre plan de continuité des opérations, vous voulez garantir votre capacité à satisfaire continuellement les niveaux de service minimum convenus.

Vous voulez assurer la continuité des services informatiques.

Comment est-ce que DevOps peut vous être utile en matière de Gestion de la continuité des services informatiques ?

- A) Les valeurs culturelles de DevOps, Affinité et Collaboration, font en sorte que Service est très apprécié par les membres de l'équipe DevOps.
 - B) DevOps prépare les routines à appliquer par l'équipe en cas de catastrophe et les pratiques Obeya en mettant délibérément le chaos dans le système.
 - C) Les mesures de réduction des risques et les options de restauration sont probablement codifiées, car les équipes d'Exploitation travaillent en collaboration avec celles du Développement.
 - D) La gestion des niveaux de service devient plus importante dans DevOps, car c'est la tâche principale du Maître de processus que de surveiller ce point.
-
- A) Incorrect. Les valeurs culturelles permettent au personnel de se sentir mieux, de mieux travailler, à un rythme régulier, de faire des erreurs et d'en tirer les leçons. Elles ne contribuent pas, en soi, à la gestion des niveaux de service.
 - B) Incorrect. Le fait de semer le chaos, à l'instar de Chaos Monkey de Netflix, pourrait vous aider à commencer le codage de mesures de réduction des risques et d'options de récupération, mais en soi, cela ne contribue pas à la gestion des niveaux de service. De plus, le chaos dans le système doit être résolu par le codage, pas par des solutions dignes d'un centre de crise.
 - C) Correct. Il s'agit d'un processus systématique pour éviter, prévoir et gérer les risques liés aux infrastructures et aux applications, qui, s'ils ne sont pas traités peuvent conduire à des perturbations et à des incidents, perturbant potentiellement les services. (*Ouvrage : B, Chapitre 11 et C, Chapitre 4*)
 - D) Incorrect. La surveillance de la gestion des niveaux de service ne constitue pas une tâche prioritaire du Maître de processus. Tout au moins, ce n'est pas une priorité standard des équipes DevOps.

48 / 50

La société ACMECONST a activement élargi sa présence internationale en augmentant le nombre d'embauches et des équipes d'ingénieurs réparties dans le monde entier. Elle a également augmenté sa clientèle au rythme annuel effréné de 30%.

Des décisions prises autrefois rapidement, lorsque l'équipe d'ingénieurs se trouvait dans un seul local, prennent maintenant beaucoup plus de temps, causant une frustration dans toute l'organisation. Le processus de prise de décision est soumis à de nombreuses couches hiérarchiques et est plus étendu, ce qui cause la désillusion des ingénieurs vis-à-vis du processus dans son ensemble.

Il existe également une confusion croissante quant à la paternité des divers problèmes qui se présentent, ce qui provoque une hésitation au niveau des prises de décision. Les ingénieurs ont également le sentiment que leur créativité est étouffée par les processus supplémentaires et la bureaucratie, et cela commence à miner leur moral.

Quelle est la **meilleure** approche pour résoudre cette situation ?

- A) Conserver les processus actuels, mais établir clairement les rôles, la responsabilité et la propriété de chaque processus, établir une méthode efficace pour évaluer la productivité par rapport aux risques, procéder à des changements progressifs et créer des lieux sûrs pour l'expérimentation.
 - B) Réévaluer les processus afin de déterminer ce qui peut être rationalisé et établir clairement les rôles, la responsabilité et la propriété de chaque processus, établir une méthode efficace pour évaluer la productivité par rapport aux risques, procéder à des changements progressifs et créer des lieux sûrs pour l'expérimentation.
 - C) Réévaluer les processus afin de déterminer ce qui peut être rationalisé et établir clairement les rôles, la responsabilité et la propriété de chaque processus, établir une méthode efficace pour évaluer la productivité par rapport aux risques, procéder à des changements progressifs et réduire le nombre d'expérimentations afin d'éviter les défaillances inutiles d'application.
-
- A) Incorrect. Le maintien des processus actuels n'est pas la meilleure idée : l'organisation a grandi et les processus actuels ne sont plus efficaces. Toutefois, c'est une bonne idée de créer un endroit sûr pour les expérimentations.
 - B) Correct. Ces actions ont un effet positif sur une entreprise en pleine croissance. Il est nécessaire de réexaminer les processus car les anciens processus ne fonctionnent plus. Un endroit sûr pour les expérimentations est également nécessaire pour permettre l'amélioration continue. (*Ouvrage : A, Chapitre 15*)
 - C) Incorrect. Il est nécessaire de réexaminer les processus. Cependant, la réduction des expérimentations est une mauvaise idée : en entravant les expérimentations, cela compromet l'amélioration continue qui doit avoir lieu.

49 / 50

Dans l'entreprise X-AppGo, il existe un conflit entre l'équipe d'Exploitation en Colombie et l'équipe de Développement en Irlande, causé par le fait qu'elles ont des priorités et des objectifs différents. En raison de ce conflit, le temps et les efforts nécessaires pour résoudre les problèmes, qui ont un impact sur l'entreprise, ne cessent d'augmenter.

Quelles pratiques-clés, l'entreprise X-AppGo devrait-elle envisager en vue de réduire les conflits et améliorer la collaboration entre les équipes de Développement et d'Exploitation ?

- A)**
 - 1. Permettre aux équipes de Développement et d'Exploitation de travailler séparément l'une de l'autre, si elles le souhaitent, afin d'éviter les conflits.
 - 2. Obtenir l'appui de l'ensemble du conseil d'administration par rapport au soutien des équipes de Développement et d'Exploitation.
 - B)**
 - 1. Demander à un sponsor du conseil d'administration de l'entreprise de parler à l'équipe DevOps de l'importance de la collaboration.
 - 2. Former les équipes de Développement et d'Exploitation aux pratiques du DevOps, afin qu'elles apprennent à faire le travail l'une par l'autre.
 - C)**
 - 1. Faire en sorte que les équipes de Développement et d'Exploitation visitent d'autres entreprises dans lesquelles DevOps fonctionne bien.
 - 2. Mettre à disposition le budget pour mieux faire face aux exigences auxquelles les équipes d'Exploitation et de Développement sont confrontées.
 - D)**
 - 1. Recommander des visites mutuelles des équipes de Développement et d'Exploitation afin d'établir des liens, et développer la confiance et la compréhension mutuelle.
 - 2. Diffuser des connaissances auprès des équipes de Développement et d'Exploitation de sorte qu'elles collaborent plus efficacement.
- A)** Incorrect. Il faut encourager les équipes à travailler ensemble, et leur interdire de travailler séparément. Elles ont besoin l'une de l'autre et doivent apprendre l'une de l'autre. Aucune collaboration ne se produit lorsque vous ne vous voyez pas l'un l'autre. L'appui du conseil d'administration n'aide pas vraiment les équipes à collaborer.
- B)** Incorrect. Bien qu'un sponsor du conseil d'administration puisse faire en sorte que les équipes se sentent plus appréciées, il ne favorise pas leur collaboration mutuelle. La formation a également ses limites. La seule manière d'améliorer la collaboration des deux équipes est, justement, de travailler ensemble.
- C)** Incorrect. La visite d'autres entreprises peut être source d'inspiration, mais DevOps doit être adapté à la spécificité de l'organisation de l'entreprise. Cela ne contribue pas directement à réduire les conflits et à améliorer la collaboration. Au lieu de cela, les équipes doivent travailler ensemble et partager leurs connaissances. Une augmentation du financement peut avoir un effet positif lorsqu'il y a trop de travail à faire pour trop peu de personnel, mais n'a pas d'autre effet sur la réduction des conflits et l'accroissement de la collaboration.
- D)** Correct. Ce sont les pratiques les plus appropriées afin de favoriser le scénario actuel de réduction des conflits et d'amélioration de la collaboration entre les équipes d'Exploitation et de Développement. (*Ouvrage : A, Chapitre 15*)

50 / 50

Une équipe de Développement s'intéresse à DevOps. Elle est surtout intéressée par l'intégration continue. Ils développent et maintiennent actuellement 3 solutions majeures et 4 plus petites en utilisant les pratiques de Scrum. Chaque sprint dure 4 semaines, entraînant en moyenne 1 mise en environnement de test tous les 10 à 15 jours et 1 mise en production par mois. L'équipe veut créer un cas d'affaire qualitatif pour leur Direction afin de soutenir leurs investissements et efforts pour créer une pratique d'intégration continue.

Quels avantages tangibles de l'intégration continue aident le **plus** ce cas d'affaires ?

- A) Le déploiement de l'environnement de test une fois par jour pourrait augmenter les avantages pour l'entreprise et diminuer considérablement les coûts de développement.
 - B) Cela contribue à l'esprit d'équipe. Comme l'équipe utilise déjà Scrum, l'intégration continue ne générera **pas** d'avantages mesurables pour l'entreprise.
 - C) Cela augmente la stabilité et la qualité de la version grâce à un test automatisé, tout en facilitant et augmentant la vitesse générale de mise à jour.
 - D) La mise en production une fois par jour pourrait augmenter les avantages pour l'entreprise et diminuer considérablement les coûts de développement.
-
- A) Incorrect. Un déploiement plus rapide en environnement de test est OK, et une conséquence de l'intégration continue, mais n'engendrera aucun avantage pour l'entreprise.
 - B) Incorrect. L'intégration continue peut aider l'équipe à livrer plus rapidement à la production, à trouver des bugs plus rapidement et à moindre coût, le fait qu'elle utilise ou non Scrum n'est pas pertinent.
 - C) Correct. L'intégration continue a pour conséquence d'accroître la vitesse de mise à jour, chaque modification étant fonctionnelle et intégrée au code principal, prête à être livrée à la production, le produit étant toujours en état de fonctionnement. De plus, cela augmentera la stabilité et la qualité de la mise à jour en détectant et en corrigeant les bogues plus rapidement grâce à des tests automatisés. (*Ouvrage : B, Chapitre 3*)
 - D) Incorrect. Une mise en production plus rapide est l'un des principaux avantages de la livraison continue, et non l'avantage direct de l'Intégration continue, qui vise la livraison dans un environnement similaire à la production afin de s'assurer que la version est en état de fonctionner et que les tests automatisés ont été menés.

Évaluation

Le tableau ci-dessous indique les bonnes réponses aux questions de cet exemple d'examen.

Question	Réponse	Question	Réponse
1	B	26	D
2	D	27	A
3	A	28	D
4	B	29	C
5	B	30	A
6	B	31	B
7	C	32	B
8	C	33	A
9	A	34	C
10	B	35	C
11	B	36	C
12	A	37	A
13	D	38	C
14	D	39	B
15	A	40	B
16	D	41	B
17	A	42	A
18	B	43	D
19	C	44	D
20	B	45	B
21	A	46	C
22	D	47	C
23	D	48	B
24	A	49	D
25	C	50	C

Contacter EXIN

www.exin.com

