



**Examen de muestra**

Edición 201804

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.



# Índice

Introducción	4
Examen de muestra	5
Soluciones	26
Evaluación	65



# Introducción

Éste es el examen de muestra de EXIN DevOps Master™ (DEVOPSM.SP). Las normas y reglamentos de exámenes EXIN se aplican a este examen.

Este examen consiste en 50 preguntas. Cada pregunta tiene un número de respuestas posibles, de las cuales sólo una es correcta.

El número máximo de puntos que se pueden obtener en este examen es de 50. Cada respuesta correcta tiene un valor de un punto. Si usted consigue 33 puntos o más, habrá aprobado el examen.

El tiempo permitido para este examen es de 120 minutos.

¡Buena suerte!

# Examen de muestra

1 / 50

¿Cuál es un buen motivo para implementar DevOps en una empresa?

- A) DevOps tiene ciclos de feedback más frecuentes para nuevos servicios, ya que la velocidad de desarrollo es mayor.
- B) DevOps mejora la continuidad y la agilidad de negocio, ya que los procesos están optimizados y solo se llevan a cabo actividades de valor añadido.
- C) DevOps presta los nuevos servicios Just-in-Time, ya que realiza entregas de software con mayor frecuencia.
- D) DevOps reduce los desperdicios (waste) en la colaboración entre Desarrollo y Operaciones, ya que están unificados.

2 / 50

El director de tecnología (CTO) piensa que podría ser más efectivo el aplicar algunos conceptos Lean mientras se implementa DevOps.

¿Qué conceptos o métodos Lean serán los **más** eficaces cuando se está implantando DevOps?

- A) **Kaizen y 5S.** Porque Agile y DevOps están basados en principios fundamentales de Lean, y Kaizen y 5S son la base de Lean, luego serán los más eficaces para introducir DevOps.
- B) **Aplicar la metodología Kaizen in Advance.** DevOps necesita información del equipo de Operaciones y del de Desarrollo. Aplicar la metodología Kaizen in Advance crea un bucle ascendente de retroalimentación (up-stream feedback loop), lo que ayuda a introducir este concepto en DevOps.
- C) **El sistema Obeya.** DevOps integra procedimientos de diferentes tipos de gestión. El sistema Obeya facilita la visualización del proceso al completo lo que permite una introducción de DevOps eficaz.
- D) **Flujo de una pieza y JKK.** DevOps aprovecha la creación de procesos ascendentes (up-stream) y un único flujo de valor. El flujo de una pieza lo permite, y JKK ayuda en la implementación y simplificación del flujo.

3 / 50

En comparación con un proyecto que se ejecuta de forma convencional, ¿qué hay que cambiar para que un proyecto tenga éxito en DevOps?

- A) Debería crearse una cadena de suministro de servicios de TI, usando un sistema pull y el flujo de una pieza.
- B) Los desarrolladores deberían unirse al equipo de Operaciones para el mantenimiento rápido de los servicios.
- C) El equipo de Operaciones debe trabajar para el equipo de Desarrollo. Por eso se denomina DevOps.
- D) Los integrantes del equipo de Operaciones deberían unirse al equipo de Desarrollo.

4 / 50

Cualquier equipo que adopta DevOps podría describirse como un "pacto" (Compact).

¿Qué principio es **más** aplicable para que un pacto funcione?

- A) Una colaboración dinámica
- B) Una comunicación continua
- C) La reducción de los ciclos
- D) Un reparto de las responsabilidades

5 / 50

Hay numerosas fuentes de conocimiento, estándares y prácticas disponibles para implementar DevOps.

¿Cuál de los siguientes se considera un elemento **clave** para una implantación con éxito de DevOps?

- A) Nivel 3 del CMMI
- B) Disciplined Agile
- C) ISO 20001
- D) PMBok del PMI

6 / 50

La colaboración es uno de los cuatro pilares de un DevOps eficaz.

¿Por qué es tan importante?

- A) La colaboración es la única manera de lograr un flujo de una pieza en DevOps.
- B) El valor de negocio de DevOps se alcanza mediante la cooperación entre equipos.
- C) Las empresas que implementan DevOps son pequeñas, por lo que los equipos deben tener una buena colaboración.
- D) Impulsa el cambio y es un beneficio fácil de lograr. Además, reduce los costes de desarrollo.

**7 / 50**

Usted es el propietario de una pequeña empresa en la que se implementa DevOps, que cuenta con 5 empleados y que crea aplicaciones móviles para niños con discapacidad. Una de las aplicaciones de las que más se enorgullece su equipo es una aplicación para niños con autismo, que les permite planificar el día por sí mismos.

Dado el gran éxito de esta aplicación, se le pide a su equipo que amplíe las funciones de la misma para otras personas que también pueden beneficiarse de planificar su día. Sin lugar a dudas, esta solicitud generará una mayor complejidad en el código y algunas dificultades técnicas que el equipo tendrá que abordar.

Usted espera que la remuneración por este encargo sea elevada, por lo que decide aceptarlo. Sin embargo, tras unas semanas, su equipo está discutiendo. Usted también se enfada y empieza a prestar más atención a lo que hace cada miembro del equipo. Normalmente, trabaja con ellos para poder corregir rápidamente cualquier error que detecte en el código.

Aunque se saca adelante el trabajo suficiente, su equipo permanece enfadado y usted se percata de que se ha perdido una parte de la mentalidad de DevOps.

¿Cuál es la **mejor** estrategia para resolver su problema?

- A) Incorporar más miembros al equipo para aumentar la diversidad, contratando a personal complementario no técnico
- B) Pedir a expertos en DevOps externos a su empresa que patrocinen y proporcionen mentoría a los miembros de su equipo
- C) Encontrar un objetivo común y empiecen a trabajar conjuntamente para conseguirlo a fin de aumentar la colaboración
- D) Reservar unos meses para trabajar en primer lugar en la consolidación del equipo y que todos los integrantes se conozcan

**8 / 50**

Usted cree que su equipo de Desarrollo es realmente un equipo.

¿Cuál es una señal clara e indiscutible de que son un equipo y no un grupo?

- A) El equipo sigue las normas que se han acordado en las reuniones de equipo.
- B) El equipo mantiene reuniones útiles que lideran ellos mismos.
- C) El equipo mantiene un ritmo de trabajo sostenible hacia un objetivo común.
- D) El equipo resuelve los problemas cuestionando al integrante del equipo responsable del asunto.

9 / 50

AppAtoZ ha experimentado un ritmo de crecimiento espectacular en cuanto al desarrollo y despliegue de aplicaciones móviles en las plataformas iPhone y Android.

Los equipos de desarrollo de esta startup han sufrido una presión inmensa para desplegar mejoras rápidas sobre sus aplicaciones móviles en unos plazos muy agresivos. Han estado trabajando una media de 60 horas a la semana durante los últimos seis meses. La dirección ha sido reacia a contratar a más personas para ayudar, y les preocupa más incrementar los beneficios y reducir los costes operativos y de desarrollo.

En los últimos meses ha aumentado la tasa de absentismo de los trabajadores, de bajas por enfermedad e incluso algunos empleados de AppAtoZ dimitieron, lo que provocó una mayor carga de trabajo en los trabajadores que quedaron. El elevado tiempo de contratación y aprendizaje hace que la presión de trabajo que sufre el equipo de desarrollo no se alivie lo suficientemente rápido.

¿Qué estrategias **a largo plazo** deben tenerse en cuenta para resolver el problema del estrés y el cansancio de los empleados?

- A)
  - incorporar al equipo una mezcla de trabajadores permanentes y externos para que puedan gestionar mejor el trabajo
  - que la dirección y el departamento de Desarrollo identifiquen las causas en el ambiente laboral que puedan estar provocando el agotamiento
  - desarrollar un plan para mitigarlas
- B)
  - despedir al equipo directivo por su incapacidad
  - contratar un equipo directivo que sepa crear conciliación entre la vida personal y profesional y una cultura que sea sostenible y realista
  - concederle la oportunidad al equipo de Desarrollo de tomarse su tiempo y encontrar su equilibrio
- C)
  - darles unos días de descanso a los miembros del equipo de Desarrollo
  - contratar desarrolladores que gestionen el aumento de la carga de trabajo de desarrollo en periodos puntuales del año
  - ayudar a los desarrolladores a encontrar psicólogos o psiquiatras si los necesitan
- D)
  - darles unos días de descanso a los miembros del equipo de Desarrollo
  - hacer que la dirección y el equipo de desarrollo identifiquen las causas que puedan estar provocando el agotamiento
  - desarrollar un plan para mitigarlas



**10 / 50**

Usted trabaja en un equipo de DevOps como desarrollador. Intenta facilitar la colaboración entre todos los miembros de su equipo. En su equipo hay dos hombres en el puesto de desarrolladores senior y una mujer en el puesto de administradora de sistemas junior.

Los miembros de su equipo han tenido unos comienzos complicados y no han congeniado. Usted intervino en su momento para intentar ayudarles a que se llevaran mejor. Eso funcionó, ya que dejaron de discutir y empezaron a terminar una mayor cantidad de trabajo.

Ahora, se ha percatado del siguiente patrón: la administradora junior siempre se muestra de acuerdo con los desarrolladores senior.

¿Cómo debería reaccionar ante este patrón?

- A) Mientras que los integrantes del equipo no discutan, no debería intervenir en la situación. Tienen una relación cordial, usted no debería alterar esta situación.
- B) Es importante que Operaciones exprese cualquier inquietud respecto a las compilaciones, por lo que debe orientar a la administradora para que deje de ser complaciente y sea más firme.
- C) Los desarrolladores senior tienen una mayor responsabilidad, por lo que debería pedirles que sean más amables y que piensen en las inquietudes relacionadas con Operaciones si la administradora junior no lo hace.
- D) Debería empezar a trabajar en el equipo de Operaciones por el momento, incluso si usted en realidad es un desarrollador, para poder orientar a la administradora junior.

**11 / 50**

¿Por qué Agile y Scrum garantizan un desarrollo de software más rápido y predecible?

- A) Posibilitan una recopilación y un tratamiento de los requisitos mejores y más completos antes del diseño
- B) Posibilitan el trabajo en equipos autónomos reducidos que se organizan y planifican por sí mismos
- C) Permiten que el Propietario del Producto participe en las reuniones diarias de planificación
- D) Permiten que el gestor de proyectos cambie rápidamente las prioridades según sea necesario

**12 / 50**

¿Qué es gestión simplificada de servicios TI (light-weight ITSM)?

- A) Una gestión de servicios TI (ITSM) orientada estrictamente a la continuidad del negocio
- B) Una nueva versión de ITIL sugerida como la versión estándar
- C) Una implementación deficiente de los procesos ITIL
- D) Una gestión de servicios TI orientada a la gestión de entregas

**13 / 50**

Usted trabaja para una empresa que ha implantado prácticas Lean y Agile. Su consejero delegado no está convencido del valor que aporta implantar, una vez más, algo nuevo: DevOps.

Usted es un experto en DevOps. Considera que DevOps es beneficioso para cualquier empresa que desarrolla software.

¿Qué aporta DevOps a la empresa para la que trabaja?

- A) Se crean mejores Historias de Usuario y requisitos funcionales
- B) Las entregas se realizan con menos frecuencia, por lo que los clientes no se ven afectados por actualizaciones constantes
- C) Se forma a los profesionales para que puedan trabajar tanto en Desarrollo como en Operaciones
- D) Se escribe un código que se entrega con mayor velocidad, apoyando así los resultados de negocio

**14 / 50**

La alta dirección de la empresa solicita al departamento de TI una mejora en el apoyo empresarial y en la alineación con los objetivos del negocio. Como Director de Informática, entre otras medidas, usted decide reducir la carga de trabajo de gestión en el área de Operaciones.

¿En qué manera le ayudará DevOps **principalmente** a mejorar la gestión de niveles de servicio existente?

- A) Abandonando las buenas prácticas de gestión de servicios de TI (ITSM), puesto que no son más que una carga
- B) Diseñando mejores acuerdos sobre el nivel de operaciones entre Desarrollo y Operaciones
- C) Implantando un nuevo conjunto de procesos ITSM basados en la versión 3 de la ITIL
- D) Reajustando la ITSM para utilizar un conjunto de información mínima necesaria

**15 / 50**

¿Qué implementación de DevOps es **más** adecuada para una empresa que utiliza el enfoque de System of Record (SoR)?

- A) La colaboración
- B) Entrega continua
- C) El método Toyota

**16 / 50**

¿Cuál es el **principal** beneficio de usar el sistema Obeya?

- A) Ayuda con las quejas de los clientes, para garantizar que el equipo reciba comentarios suficientes para mejorar constantemente
- B) Gestiona el estrés en los equipos, de modo que los miembros del mismo puedan mantener un ritmo sostenible
- C) Mejora el proceso diario de informe de errores, lo cual garantiza que se reduzca el retrabajo y que se pasen los errores a otras estaciones de trabajo
- D) Una rápida toma de decisiones, basada en el estado actual, acelerando la recopilación y distribución de información

**17 / 50**

Usted trabaja en una empresa de software que está empezando a implementar DevOps. Se percata de que no sólo debe cambiar la cultura de la empresa, sino también sus prácticas y herramientas. La empresa ya supervisa los proyectos de software con el fin de detectar cualquier problema.

Usted propone que se adopten las medidas siguientes:

- automatizar el proceso de gestión de cambios;
- implantar controles de acceso para evitar que alguien realice cambios sin consentimiento.

¿Por qué es esto necesario?

- A)** La automatización permite que se implementen los cambios con mayor rapidez a la vez que se mantiene la confianza. El control de acceso es necesario para evitar heurísticas de la resolución de problemas y alteraciones imprevistas del servicio.
- B)** La automatización es necesaria para ralentizar el número de cambios. El control de acceso debería evitar que los clientes modifiquen el software por sí mismos, sin que ustedes lo sepan y fuera de su control.
- C)** La automatización ayudará a que el equipo de Operaciones sea innecesario. El control de acceso debe implementarse, ya que los proyectos de DevOps funcionan en un entorno de nube, donde los riesgos asociados son más elevados.

**18 / 50**

La virtualización y la computación en la nube son técnicas que podrían ayudar a facilitar las prácticas de DevOps.

¿Cómo ayudan a DevOps?

- A)** La computación en la nube posibilita el acceso remoto, lo cual otorga a los clientes un mayor control.
- B)** Los entornos virtuales son más fáciles de estandarizar y hacen un uso más eficaz del hardware.
- C)** La infraestructura virtualizada es más fácil de entender y no requiere ningún mantenimiento.

**19 / 50**

Los Acuerdos de Nivel de Servicio (SLAs) son importantes para cada proyecto, ya que en ellos se especifica lo acordado con el cliente. No obstante, en DevOps, los SLAs tienen además otra finalidad importante.

¿De qué finalidad se trata?

- A)** Los clientes tienen la responsabilidad de crear el SLA para el equipo de DevOps. Por lo tanto, sustituye una tarea formal y la convierte en responsabilidad de Operaciones.
- B)** En el SLA, el cliente puede especificar todos sus requisitos no funcionales, de modo que Desarrollo pueda centrarse en los mismos.
- C)** En el SLA se especifica el nivel de servicio aceptable. Desarrollo debería entender el SLA y apoyar a Operaciones en su cumplimiento.

**20 / 50**

Cuando se trata de un producto nuevo, su equipo necesitará desarrollar un flujo de despliegue. Como parte de la integración continua hay que definir una fase de commit (commit stage) en el flujo. Discute esta fase con los integrantes de su equipo.

El Process Master afirma: "Deberíamos establecer la Definición de Hecho antes o durante la etapa de fase de commit. Si el código no está Hecho cuando se hace el commit, el trabajo debería detenerse".

¿Es verdad?

- A) Sí. Si el trabajo no está Hecho, el Process Master no está realizando bien su función. Esto es algo que debería resolverse de inmediato.
- B) Sí. No debería hacerse un commit de trabajo que no cumple con la Definición de Hecho ya que no está aportando valor al cliente.
- C) No, la Definición de Hecho se establece solo en las reuniones con el cliente. Esperar a que se haga retrasaría demasiado el trabajo.
- D) No, el trabajo en el flujo de despliegue debería continuar siempre. Si el código no está Hecho es necesario que esté inactivo.

**21 / 50**

¿Cuál es la mayor ventaja que tiene aumentar la diversidad del equipo de forma que incluya un espectro más amplio de culturas y personalidades?

- A) Aporta mayor cantidad de experiencias y puntos de vista.
- B) Conlleva menos roces entre los integrantes del equipo.
- C) Pone límites a la originalidad y la capacidad de proponer nuevas ideas.
- D) Se tardará más en llegar a un punto concreto de toma de decisiones.

**22 / 50**

Usted trabaja para una empresa pequeña que cuenta con un único equipo de DevOps. Su equipo de DevOps está trabajando en una aplicación formada por múltiples componentes. Algunos son nuevos y otros solo requieren actualizaciones.

Por ahora, cada componente tiene su propio flujo de despliegue. El equipo está orgulloso de la entrega continua que realizan en la mayoría de los flujos de despliegue y su producción es elevada y de buena calidad.

¿Qué debería hacer usted?

- A) Mantener los distintos flujos y alentar al equipo a que amplíen la entrega continua
- B) Conservar únicamente los flujos que tengan una entrega continua y fusionar el resto de flujos sin entrega continua
- C) Conservar exclusivamente dos flujos diferentes: uno para el desarrollo y otro para el mantenimiento
- D) Trabajar con su equipo para conseguir un único flujo, explicando los riesgos que entraña tener múltiples flujos

23 / 50

Usted está asesorando a Company Builders, una empresa mediana-grande que introdujo los métodos DevOps hace un par de años.

Le han contratado para determinar su madurez actual. Cuando termine, debería ofrecerles algunas sugerencias para mejorar. Quieren saber en qué área deben centrarse para pasar al siguiente nivel de madurez: *nivel 2 - Gestionado cuantitativamente*.

Se da cuenta de que en la mayoría de las áreas se encuentran en el *nivel 1- Uniforme* con dos excepciones:

- **entornos y despliegue.** Esta área gestiona despliegues orquestados y ha realizado pruebas en los procesos de Despliegue y Roll-back.
- **gestión de compilaciones (Build management) e integración continua.** En esta área comprueba que existen compilaciones y pruebas automatizadas que se ejecutan periódicamente, y cada compilación puede volverse a crear desde el control de versiones de código fuente mediante un proceso automatizado.

Primero, establezca el nivel de madurez en estas dos áreas basándose en la información obtenida. Después, presente sus recomendaciones sobre lo que se deben centrar en mejorar.

¿En qué dos áreas debería trabajar Company Builders antes de pasar al nivel 2?

- A) “Entornos y despliegue” y “gestión de compilaciones e integración continua” están en el nivel 0. Deberían trabajar en ambas a la vez.
- B) “Entornos y despliegue” y “gestión de compilaciones e integración continua” están en el nivel 1 o por encima. Deberían trabajar en otras áreas de mejora para progresar.
- C) “Entornos y despliegue” está en el nivel 0. “gestión de compilaciones e integración continua” está en el nivel 1. Deberían centrarse primero en “entornos y despliegue”.
- D) “Entornos y despliegue” está en el nivel 2. “gestión de compilaciones e integración continua” está en el nivel 0. Deberían centrarse solo en “gestión de compilaciones e integración continua”.

**24 / 50**

Su empresa vende un servicio de copia de seguridad en línea a sus clientes. Ahora, uno de los clientes ha pedido que se implanten nuevas funcionalidades en su servicio. Quieren que las nuevas funcionalidades estén listas en el plazo de una semana, de lo contrario buscarán otra empresa que lo haga.

Usted considera que las nuevas funcionalidades son importantes y que el equipo de Desarrollo puede crearlas en poco tiempo. No obstante, se encuentra con los problemas siguientes:

- Los evaluadores (testers) tardan mucho en solucionar los errores.
- Los evaluadores están encontrando errores que los desarrolladores corrigieron hace mucho tiempo.
- Rara vez se puede demostrar que la aplicación está funcionando.
- Rara vez se realizan presentaciones o demos.

¿Cuál es su problema y cómo debería resolverlo?

- A)** Está desplegando códigos con errores. Debería resolver esto entendiendo mejor el proceso de Despliegue, aumentando la colaboración y trabajando con una mayor disciplina.
- B)** Tiene una gestión de la configuración deficiente. Debería resolver esto mejorando la colaboración entre Desarrollo y Operaciones y aumentando la monitorización, el registro de eventos y la virtualización.
- C)** Su proceso de integración continua no se está gestionando adecuadamente. Debería resolver esta situación acelerando las pruebas automatizadas y la fase de commit, así como mejorando la comprensión del proceso de integración continua.
- D)** Su estrategia de pruebas no es eficaz. Debería resolver esto usando pruebas automatizadas y aumentando la colaboración entre los evaluadores y el resto del equipo.

**25 / 50**

¿Cuál es uno de los beneficios de la integración continua en el contexto de un DevOps eficaz?

- A)** Los ciclos de pruebas exhaustivos antes de la entrega de funcionalidades
- B)** Los largos períodos entre las entregas de funcionalidades
- C)** El aumento de la frecuencia y la puntualidad en las entregas de funcionalidades

**26 / 50**

Una empresa multinacional está encontrando muchos problemas a la hora de hacer el merge de los commits de código que vienen desde sus ubicaciones distribuidas en un repositorio central ubicado en Dallas (Texas). Las ubicaciones distribuidas están en Ciudad de México, París, San Diego e Inglaterra. Se han observado comportamientos incoherentes en cuanto al momento en el que las regiones están haciendo el commit del código y en ocasiones no está claro si las pruebas han dado error o no.

Existen cuatro métodos posibles:

- No registrar el código cuando la compilación está dañada
- Realizar las pruebas de confirmación locales siempre antes de hacer el commit o dejar que sea el servidor de integración continua quien lo haga por tí
- Dejar terminar las pruebas de la fase de commit antes de continuar
- No desactivar las pruebas que dan error

¿Cuáles de estas opciones es la **más** adecuada para garantizar que las ubicaciones distribuidas corrigen sus puntos débiles?

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

**27 / 50**

Analice la anatomía de un flujo de despliegue básico.

¿Qué fase confirma que el sistema es válido a nivel funcional y no funcional?

- A) Prueba de aceptación automatizada
- B) Prueba unitaria y de compilación
- C) Prueba de aceptación manual
- D) Control de Versiones

**28 / 50**

Una de las prácticas DevOps más eficaces es utilizar el mismo procedimiento para realizar el despliegue en todos los entornos en los que esté presente la aplicación. Esto garantiza que se ponga a prueba la compilación de forma eficaz. Está utilizando scripts para automatizar los procesos de compilación y despliegue.

¿Cuál es la **mejor** manera de hacerlo?

- A) Utilizar un script para cada entorno y guardarlos como parte de un sistema de control de versiones.
- B) Utilizar un script concreto para cada entorno para corregir las diferencias entre los entornos.
- C) Utilizar los mismos scripts para todos los entornos aplicando parámetros manuales para configuraciones específicas.
- D) Utilizar los mismos scripts para el despliegue en cada entorno y gestionar la información de configuración por separado.

**29 / 50**

Tras la entrega de un nuevo servicio de TI, un proceso (job) en Operaciones termina de manera inesperada.

¿Cuál **no** puede ser una causa válida para ello?

- A) El gatekeeper no tuvo en cuenta la interrelación entre las tareas del trabajo y los criterios para la aceptación de servicios.
- B) Los requisitos no funcionales no estaban claros, debido a una historia de usuario incompleta.
- C) El Service Master no llegó a un acuerdo con el usuario acerca del final de la vida útil del servicio que se va a entregar.
- D) No se trabajó lo suficiente en los procesos de desarrollo para cumplir con los criterios de aceptación del servicio.

**30 / 50**

La empresa AppBC está utilizando DevOps. Han implementado el despliegue continuo y un flujo de despliegue estable, con pruebas de aceptación muy automatizadas y pasan software a producción diariamente.

AppBC tiene una base de datos enorme con muchos usuarios. Disponen de una muy buena estrategia de pruebas de capacidad Debido a lo voluminoso y complejo del entorno, con cada nueva versión aparecen errores en Producción.

¿Qué estrategia ayudará **más** a AppBC a evitar estos errores?

- A) Introducir el Canary Releases
- B) Automatizar las pruebas de capacidad
- C) Reducir la tasa de entregas
- D) Utilizar despliegues "azul-verde" (Blue-Green deployment)

**31 / 50**

DevOps toma conceptos muy importantes de Agile, derivado del sistema de producción de Toyota.

¿Por qué el flujo de una pieza es importante para la adopción de DevOps?

- A) Permite que su equipo trabaje a un ritmo sostenible y a una velocidad predecible.
- B) Permite que el equipo se centre en añadir funcionalidades con el mayor valor de negocio posible.
- C) Aumenta el reparto de responsabilidades del trabajo entre los diferentes integrantes del equipo.
- D) Reduce los cuellos de botella limitando el número de tareas que realiza de manera simultánea.



### 32 / 50

La empresa S Corporation, una compañía de tamaño medio de piezas de automóviles, es proveedor de una gran compañía, T Motors Corporation. Proporcionan a T Motors Corporation piezas de automóvil que representan casi el 60% del total de sus ventas.

La junta se reúne para tratar una nueva colaboración. T Motors Corporation exige que S Corporation cambie su modo de entrega al método Just-in-Time o dejarán de trabajar con ellos. S Corporation no podría sobrevivir si perdiera este negocio, de forma que es bastante urgente hacer el cambio a Just-in-Time. El cambio se deberá llevar a cabo dentro de los 6 meses siguientes, por lo que quedan como mucho 5 meses para prepararse.

Una de las implementaciones tiene que ver con el seguimiento de piezas mediante la identificación por radiofrecuencia (RFID). Esto facilitaría la transparencia en el proceso de producción. Se debería llevar a cabo una rápida revisión del proceso actual para facilitar el cambio a un proceso que sea adecuado para su uso con RFID.

Se pide a la directora de informática (CIO) gestionar el proceso de cambio. Cree que será posible si se utiliza el enfoque DevOps para crear una versión mínima. Lo más idóneo sería que se desarrollara primero el concepto para la producción con RFID. En última instancia se implementaría un sistema de control de producción que utilice los datos RFID. Sin embargo, no hay tiempo suficiente para dar estos dos pasos de forma consecutiva. Las tres cosas deberán hacerse al mismo tiempo.

La CIO asigna al proyecto a Em, que es Scrum Master. Desarrollo se prepara para crear un flujo de despliegue.

Em se percata de que el equipo de Desarrollo es muy entusiasta y trabaja muy duro, pero podrían ser más disciplinados. Además, la frecuencia de entregas debería ser mayor.

¿En qué debería centrarse **primero** Em?

- A) Debería centrarse en la comunicación, ya que es lo más importante en DevOps. Debería empezar rompiendo el hielo en el equipo y establecer algunas normas de comunicación.
- B) Debería empezar por hablar con el equipo del mapa de flujo de valor y crear un flujo de una pieza, ya que el flujo y los procesos simplificados son muy importantes.
- C) Debería comenzar por debatir la infraestructura y el entorno de trabajo con los miembros del equipo, ya que DevOps es más eficaz cuando las herramientas y los métodos funcionan.
- D) Debería comenzar por juntar a todos los interesados, formarles en DevOps y pedirles su apoyo para transmitir el cambio cultural, ya que esto último es lo que DevOps necesita.

**33 / 50**

Su equipo de DevOps trabaja bien conjuntamente a un ritmo sostenible. Al integrar un margen de demora suficiente en el proceso, el equipo tiene el tiempo y la concentración suficientes para comprobar y probar cuidadosamente las compilaciones (builds). Actualmente, su equipo realiza pruebas y despliegues manualmente. Su ritmo es lo suficientemente elevado como para aportar regularmente un gran valor a la empresa.

Su consejero delegado le ha pedido su consejo sobre la automatización en dicho equipo.

¿Qué debería aconsejarle?

- A) Automatice todo lo que se pueda, para que el equipo pueda añadir más funcionalidades y demostrar antes el valor de negocio
- B) Automatice las pruebas de aceptación, pero no los despliegues, ya que el proceso manual es más seguro
- C) Automatice los despliegues para mejorar el tiempo de ciclo, pero no las pruebas, para poder aprender de los errores
- D) No añada la automatización a los métodos de este equipo, ya que los métodos que ejecuta el mismo en la actualidad están dando unos resultados magníficos

**34 / 50**

La directora de los servicios informáticos (CIO) asigna al proyecto a su empleado más fiable, Michael, que es Scrum Master. El equipo de desarrollo se prepara para elaborar un flujo de despliegue.

Michael confía en las buenas intenciones y la espontaneidad del equipo de Desarrollo, pero le gustaría que tuvieran una mayor disciplina. Además, la frecuencia de entregas debería ser mayor. Michael quiere que el equipo de Desarrollo implemente versiones con más frecuencia.

Uno de los integrantes del equipo sugiere: "Lo más importante del nuevo flujo de despliegue es automatizarlo. Deberíamos automatizar primero el flujo de despliegue".

¿Es correcta esta sugerencia?

- A) Sí, es correcta. El aspecto más importante, si queremos mejorar la eficacia, es la automatización del flujo de despliegue.
- B) Sí, es correcta. Si nos centramos en crear un flujo de despliegue automatizado, subsanamos los problemas potenciales que podamos encontrarnos después.
- C) No, no es correcta. Debería ser prioritario conseguir un flujo de una pieza y un proceso de despliegue estable. La automatización del proceso vendrá después.
- D) No, no es correcta. Debería automatizarse el proceso de pruebas primero en lugar del flujo de despliegue.

### 35 / 50

Su empresa está cambiando y empieza a trabajar con DevOps. Su equipo va a participar en el cambio. Están debatiendo las prácticas idóneas en la commit stage del código.

Su compañero Sun argumenta: "Cuando se daña una compilación (build) y nadie se hace responsable, deberíamos averiguar quién ha sido y reclamarles que la arreglen".

¿Sería una buena idea?

- A) Sí. La única persona que puede arreglar una compilación es quien la ha roto, por lo que hay que identificar quién ha sido incluso si les hace sentir incómodos.
- B) Sí. Siempre debes hacerte responsable de los daños que provoques al build. Si no es así, los compañeros tienen derecho a imponer esta regla.
- C) No, DevOps es un entorno libre de culpa. Si un compañero no quiere hacerse responsable, no debemos obligarle.
- D) No, hay que arreglar la compilación primero. Después ya habrá tiempo para identificar a los responsables y castigarles.

### 36 / 50

El equipo de Desarrollo de X-AppGo ha encontrado numerosos problemas con sus métodos de prueba actuales. Ahora mismo están utilizando un proceso de pruebas de aceptación manual. Los desarrolladores creen que el conjunto de pruebas unitarias que han creado es lo suficientemente riguroso como para proteger contra las regresiones.

El equipo de Desarrollo tiene que invertir 1 millón de dólares en las pruebas de aceptación manuales para cada entrega. La alta dirección ha exigido que el equipo de Desarrollo implemente la automatización de las pruebas de aceptación para reducir los costes totales de las pruebas y para minimizar la cantidad de errores y regresiones de código que se introducen en el entorno de producción.

¿Qué principios se deben seguir para definir los criterios de aceptación de la aplicación teniendo en cuenta la automatización?

- A) Criterios Agile
- B) Criterios ATAM
- C) Criterios INVEST

### 37 / 50

¿Cuál es el mecanismo **más** eficaz para migrar datos de una manera automatizada?

- A) Crear un esquema de Control de Versiones de la base de datos y conservarlo bajo el sistema de Control de Versiones
- B) Crear y gestionar conjuntos de datos más pequeños, para que la migración sea más sencilla
- C) Asegurarse de que todos sus scripts se han probado adecuadamente antes de la migración de los datos
- D) Asegurarse de que cuenta con un procedimiento de marcha atrás para revertir los cambios en caso de que falle la migración

**38 / 50**

La empresa X-AppGo ha estado teniendo dificultades con sus procedimientos de marcha atrás (roll-back). A menudo, esto conlleva pérdidas de datos críticos en sus bases de datos de aplicación de producción cuando ejecutan los scripts de roll-back.

¿Cuándo **no** se pueden ejecutar scripts de roll-back sin perder datos críticos?

- A) El script de roll-back elimina los datos utilizados únicamente por la nueva versión.
- B) El script de roll-back implica trasladar una columna entre diferentes tablas.
- C) El script de roll-back restaurará datos de tablas temporales.

**39 / 50**

En ACMECONST se están viendo afectados por numerosos errores en la aplicación y el hardware tras desplegar actualizaciones de software en la aplicación y actualizaciones de los enrutadores y conmutadores.

Han tenido muchas dificultades para volver al estado original tras encontrar estos fallos durante sus ventanas de mantenimiento. Como consecuencia, se ha tenido que trabajar durante largas horas en la recuperación, más allá de las ventanas de mantenimiento, y han experimentado una indisponibilidad más prolongada de sus aplicaciones críticas.

El aprovisionamiento automatizado y una infraestructura autónoma pueden ayudar en esta situación, pero se deben tener en cuenta distintos aspectos.

¿Qué elementos deben gestionarse cuidadosamente para reducir el riesgo de sufrir alteraciones a la hora de desplegar al entorno de producción?

- A) Los registros de monitorización detallados para resolver los fallos en las actualizaciones de la aplicación
- B) Los puntos de integración externos, como los sistemas y servicios externos
- C) Las configuraciones del servidor y la información sobre las cuentas de usuario relacionadas
- D) El conjunto de herramientas para automatización del aprovisionamiento y arquitecturas autónomas

**40 / 50**

La empresa X-AppGo ha estado teniendo dificultades con su aplicación principal. La aplicación no está comunicándose adecuadamente con otras aplicaciones externas. Estas aplicaciones externas necesitan obtener variables de datos específicas de manera eficaz, de modo que puedan ejecutarse llamadas específicas. La aplicación principal está siendo desarrollada por un equipo, y la empresa quiere que siga siendo así por motivos de negocio.

Uno de los desarrolladores sugiere que se separe un componente del código de base de X-AppGo para resolver los problemas de intercomunicación.

¿Qué se considera como buenas razones para separar un componente en este caso?

- A) Convertir un conjunto de complementos (plugins) en una base de código monolítico
- B) Limitar el impacto de los cambios y facilitar las modificaciones al código base
- C) Habrá que dividir el código base de X-AppGo y que lo gestionen equipos diferentes
- D) No hay buenas razones que lo justifiquen y su compilación requeriría más tiempo

**41 / 50**

Hasta las aplicaciones más pequeñas tendrán cierta dependencia de otros componentes o bibliotecas. Por lo tanto, entender y gestionar las dependencias es una actividad clave en el despliegue continuo a fin de mantener el flujo en el *pipeline* de despliegue.

Ha creado una aplicación que utiliza dos bibliotecas. Cada una de estas bibliotecas dependen de una tercera biblioteca subyacente, aunque hacen referencia a versiones diferentes. Esto genera una dependencia específica.

¿Cuál es la **mejor** solución para resolver o evitar esta dependencia?

- A) Unificar todas las bibliotecas, de modo que pueda consultar la biblioteca directamente y evitar el problema
- B) Gestionar las bibliotecas usando el Control de Versiones, de modo que pueda ver directamente si genera este tipo de dependencia
- C) Mantener un resumen visual de todas sus dependencias usando notas adhesivas en un tablón grande, de modo que pueda hacer un seguimiento del flujo
- D) Realizar el check-in únicamente en partes pequeñas de su cadena de herramientas, de modo que pueda depurar con facilidad los problemas que puedan ocurrir en el check-in

**42 / 50**

En un entorno de despliegue continuo, es importante que todo esté bajo el Control de Versiones, de modo que pueda detectar los errores rápidamente o realizar una reversión cuando sea necesario.

Sin embargo, **no** es aconsejable mantener algunos archivos binarios en el Control de Versiones.

¿Por qué se hace esta excepción?

- A) Los resultados binarios suelen ser archivos grandes que cambian con cada compilación y se actualizan automáticamente.
- B) En los archivos binarios trabajan varios miembros del equipo, por lo que no es práctico mantenerlos en el Control de Versiones.
- C) Los resultados binarios son la entrada para sus compiladores, que ya se conservan en el Control de Versiones.
- D) No es necesario hacer esto, ya que la recompilación se efectúa como una parte habitual del proceso normal de compilación.

**43 / 50**

Quiere adoptar un enfoque global para gestionar toda la infraestructura TI.

¿En qué dos criterios se podrá basar **mejor** este enfoque?

- A)
  1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de cambios.
  2. Siempre deberá conocer el estado real de su infraestructura por medio de la monitorización y la gestión de eventos.
- B)
  1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de cambios.
  2. Siempre deberá conocer el estado real de su infraestructura por medio de la instrumentación y la gestión de incidencias
- C)
  1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de versiones.
  2. Siempre deberá conocer el estado real de su infraestructura por medio de las incidencias actuales y la gestión de eventos.
- D)
  1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de versiones.
  2. Siempre deberá conocer el estado real de su infraestructura por medio de la instrumentación y la monitorización.

**44 / 50**

Los equipos en los que se emplean buenas prácticas de colaboración tienen tickets de tareas sincronizados. Un director de tecnología realizó visitas al Gemba (Go and See) para investigar cómo trabajaba el equipo de Operaciones. El equipo de Operaciones siempre redefine la infraestructura operativa tras una entrega.

¿Cuál es el **mejor** consejo que puede dar para mejorar esta práctica?

- A) No deberían hacer nada. No hay nada que mejorar, ya que la redefinición es algo que siempre debe hacerse.
- B) Deberían examinar cómo conformar la infraestructura operativa y el control de accesos del entorno operativo.
- C) Deberían revisar la infraestructura operativa, de modo que se convierta en un proceso automatizado.
- D) Deberían empezar a compartir sus conocimientos sobre el proceso de despliegue con el equipo de Desarrollo.

**45 / 50**

¿Qué momento es el adecuado para que Operaciones informe a Desarrollo de un cambio operativo?

- A) No hay que informar a Desarrollo. Los cambios operativos sólo incumben a Operaciones.
- B) Inmediatamente. Se debe informar a Desarrollo lo antes posible.
- C) En el Scrum de Scrums de la mañana siguiente.
- D) En el momento en que el equipo de Operaciones haya llevado a cabo las pruebas de aceptación.

**46 / 50**

Usted espera que su organización DevOps madure. Hay varias formas de llevarlo a cabo.

¿Cuál de los siguientes métodos **no** facilitará la maduración de la organización DevOps?

- A) Definir claramente los objetivos como hitos para facilitar que los integrantes del equipo puedan juzgar si sus actividades diarias aportan valor.
- B) Definir claramente los procesos y apoyar y permitir a los miembros del equipo que mejoren los procesos diariamente.
- C) Guardar actas de todas las reuniones para que los integrantes del equipo tengan un fácil acceso a la información.
- D) Supervisar y registrar las actividades diarias para poder identificar el progreso diario y poder celebrarlo.

**47 / 50**

Usted trabaja para un proveedor de servicios TI. Como parte del plan de continuidad del negocio quiere garantizar que siempre se puedan cumplir los niveles mínimos de servicio acordados.

Quiere garantizar la continuidad de los servicios TI.

¿Cómo podría DevOps ayudarle con la gestión de la continuidad de servicios TI?

- A) Los principios culturales de DevOps, afinidad y colaboración, aseguran que los miembros del equipo DevOps valoren mucho el servicio.
- B) DevOps prepara los procedimientos de actuación en caso de desastre y los métodos Obeya mediante la incorporación deliberada de caos en el sistema.
- C) Las medidas de reducción de riesgos y las opciones de recuperación probablemente están programadas debido a que Operaciones está trabajando de forma conjunta con Desarrollo.
- D) La gestión de nivel de servicio pasa a tener mayor importancia en DevOps, ya que la labor del Process Master es supervisarla.

#### 48 / 50

ACEMCONST ha expandido su presencia global de forma agresiva aumentando el número de contrataciones y de equipos de ingeniería distribuidos por todo el mundo. También ha aumentado su cartera de clientes al ritmo de un 30% anual.

Las decisiones, que antes se tomaban cuando el equipo de ingenieros estaba en la misma sala, ahora se demoran muchísimo, lo que provoca frustración de toda la empresa. Hay que pasar por más capas de autorizaciones y el proceso es más largo, lo que provoca que algunos ingenieros estén desencantados con todo el proceso de toma de decisiones.

También ha aumentado la confusión en cuanto a la propiedad y responsabilidades sobre los problemas que van apareciendo, lo que crea dudas a la hora de tomar decisiones. Los ingenieros piensan también que su creatividad se frena con todos estos procesos y burocracia adicionales, lo que ha empezado a minar su moral.

¿Cuál es la **mejor** manera de afrontar este escenario?

- A) Mantener los procedimientos actuales, pero creando roles, responsabilidades y titularidades claros para cada procedimiento, instaurar un método efectivo para equilibrar la productividad y el riesgo, hacer los cambios de forma gradual y crear entornos seguros para los experimentos.
- B) Revisar los procedimientos para ver cuáles se pueden simplificar y crear roles, responsabilidades y titularidades claros para cada procedimiento, instaurar un método efectivo para equilibrar la productividad y el riesgo, hacer los cambios de forma gradual y crear entornos seguros para los experimentos.
- C) Revisar los procedimientos para ver cuáles se pueden simplificar y crear roles, responsabilidades y titularidades claros para cada procedimiento, instaurar un método efectivo para equilibrar la productividad y el riesgo, hacer los cambios de forma gradual y minimizar la cantidad de experimentos para evitar fallos innecesarios de la aplicación.



**49 / 50**

Dentro de la empresa X-AppGo hay un desacuerdo entre el equipo de Operaciones que está en Colombia y el equipo de Desarrollo que está en Irlanda debido a que tienen diferentes prioridades y distintos objetivos. Debido a este problema está aumentando la cantidad de tiempo y de esfuerzos que conlleva resolver problemas que afectan a la empresa.

¿Qué métodos clave debería contemplar X-AppGo para reducir los desacuerdos y mejorar la colaboración entre los equipos de Desarrollo y Operaciones?

- A)
  - 1. Permitir que los dos equipos, Operaciones y Desarrollo, trabajen por separado si lo prefieren, con el objetivo de evitar problemas.
  - 2. Conseguir la implicación absoluta del consejo de administración en el apoyo a ambos equipos, Desarrollo y Operaciones.
- B)
  - 1. Conseguir un mentor dentro del consejo de administración de la empresa que hable con el equipo DevOps sobre la importancia de trabajar conjuntamente.
  - 2. Implicar a los equipos de Desarrollo y Operaciones en formaciones sobre los métodos DevOps de forma que aprendan a hacer el trabajo del otro.
- C)
  - 1. Asegurarse de que los equipos de Desarrollo y Operaciones visiten otras empresas donde DevOps funciona correctamente.
  - 2. Aumentar el presupuesto para poder apoyar mejor las crecientes necesidades a las que se enfrentan los dos equipos, Desarrollo y Operaciones.
- D)
  - 1. Recomendar las visitas entre los dos equipos, Desarrollo y Operaciones, para favorecer un acercamiento, aumentar la confianza y el entendimiento.
  - 2. Difundir los conocimientos entre los equipos de Desarrollo y Operaciones para que puedan trabajar de forma más eficaz.

**50 / 50**

Un equipo de desarrollo se ha interesado por DevOps. Fundamentalmente les interesa la integración continua (CI). Actualmente desarrollan y mantienen 3 soluciones importantes y 4 más pequeñas. Utilizan los métodos Scrum. Cada Sprint tarda 4 semanas, lo que conlleva una media de 1 versión confirmada al entorno de pruebas cada 10-15 días y 1 versión al mes a producción. Quieren presentar un caso de negocio cualitativo para que la dirección respalde la inversión y el esfuerzo que supone crear un método de CI.

¿Cuál de los beneficios tangibles de la CI beneficia **más** al caso de negocio?

- A) El despliegue una vez al día al entorno de pruebas aumentaría los beneficios de la empresa y reduciría considerablemente los costes de desarrollo.
- B) Contribuye al espíritu de equipo. Como ya se utiliza Scrum, la CI **no** generaría beneficios cuantificables para la empresa.
- C) Se incrementa la estabilidad de las entregas y la calidad al mejorar y automatizar las pruebas al tiempo que se facilita e incrementa la velocidad de las entregas.
- D) El despliegue a producción una vez al día aumentaría los beneficios de la empresa y reduciría considerablemente los costes de desarrollo.

# Soluciones

1 / 50

¿Cuál es un buen motivo para implementar DevOps en una empresa?

- A) DevOps tiene ciclos de feedback más frecuentes para nuevos servicios, ya que la velocidad de desarrollo es mayor.
  - B) DevOps mejora la continuidad y la agilidad de negocio, ya que los procesos están optimizados y solo se llevan a cabo actividades de valor añadido.
  - C) DevOps presta los nuevos servicios Just-in-Time, ya que realiza entregas de software con mayor frecuencia.
  - D) DevOps reduce los desperdicios (waste) en la colaboración entre Desarrollo y Operaciones, ya que están unificados.
- 
- A) Incorrecto. Los ciclos de feedback son más frecuentes cuando la velocidad de desarrollo es mayor, aunque a menudo esto se debe a Scrum o a otra metodología Agile, no a DevOps en sí mismo.
  - B) Correcto. Aportar valor añadido y optimizar los procesos son las claves para mejorar la continuidad y la agilidad de negocio. Debe pensar en qué implica el hecho de que los servicios de TI deben apoyar el negocio en todo momento y cuál es el valor y el propósito de DevOps. (*Bibliografía: C, capítulo 2*)
  - C) Incorrecto. La entrega Just-in-Time es estupenda, pero no es una buena razón por sí misma para implantar DevOps. La implementación de procesos Lean funcionaría mejor para la consecución de este objetivo.
  - D) Incorrecto. Limitarse a unificar dos equipos no va a eliminar los desperdicios. Para empezar a eliminar los desperdicios, deben cambiar sus prácticas.

2 / 50

El director de tecnología (CTO) piensa que podría ser más efectivo el aplicar algunos conceptos Lean mientras se implementa DevOps.

¿Qué conceptos o métodos Lean serán los **más** eficaces cuando se está implantando DevOps?

- A) **Kaizen y 5S.** Porque Agile y DevOps están basados en principios fundamentales de Lean, y Kaizen y 5S son la base de Lean, luego serán los más eficaces para introducir DevOps.
  - B) **Aplicar la metodología Kaizen in Advance.** DevOps necesita información del equipo de Operaciones y del de Desarrollo. Aplicar la metodología Kaizen in Advance crea un bucle ascendente de retroalimentación (up-stream feedback loop), lo que ayuda a introducir este concepto en DevOps.
  - C) **El sistema Obeya.** DevOps integra procedimientos de diferentes tipos de gestión. El sistema Obeya facilita la visualización del proceso al completo lo que permite una introducción de DevOps eficaz.
  - D) **Flujo de una pieza y JKK.** DevOps aprovecha la creación de procesos ascendentes (up-stream) y un único flujo de valor. El flujo de una pieza lo permite, y JKK ayuda en la implementación y simplificación del flujo.
- A) Incorrecto. A pesar de que Lean, Agile y DevOps están interconectados, Kaizen y 5S no son los más adecuados para respaldar un lanzamiento exitoso de DevOps. Una vez DevOps se haya introducido, Kaizen se podrá utilizar en la mejora continua y 5S podrá ayudar a mantener las buenas prácticas. Sin embargo, ambas ayudan una vez se ha introducido DevOps adecuadamente.
- B) Incorrecto. La información siempre es bien recibida, pero eso no garantiza necesariamente que la aplicación de Lean sea la más eficaz cuando se implementa DevOps.
- C) Incorrecto. La visualización puede ser útil, pero no es el método Lean de mayor repercusión cuando se implementa DevOps.
- D) Correcto. Crear un flujo de despliegue de una pieza y viable ayudará a implementar DevOps de forma adecuada. Lo más importante en DevOps es crear procesos ascendentes (up-stream) desde Desarrollo hasta Operaciones, específicos para un solo flujo de despliegue. JKK es el sistema de trabajo más eficaz para conseguir esto. *(Bibliografía: C, capítulo 4)*

3 / 50

En comparación con un proyecto que se ejecuta de forma convencional, ¿qué hay que cambiar para que un proyecto tenga éxito en DevOps?

- A) Debería crearse una cadena de suministro de servicios de TI, usando un sistema pull y el flujo de una pieza.
  - B) Los desarrolladores deberían unirse al equipo de Operaciones para el mantenimiento rápido de los servicios.
  - C) El equipo de Operaciones debe trabajar para el equipo de Desarrollo. Por eso se denomina DevOps.
  - D) Los integrantes del equipo de Operaciones deberían unirse al equipo de Desarrollo.
- A) Correcto. Un proyecto tiene éxito cuando el proceso emplea un sistema pull basado en Jidoka para crear servicios (o productos) de TI valiosos. *(Bibliografía: C, capítulo 4; B, capítulo 1)*
- B) Incorrecto. DevOps no implica que un desarrollador tenga que unirse al equipo de Operaciones.
- C) Incorrecto. La colaboración por sí sola no conlleva que el proyecto sea un éxito en DevOps.
- D) Incorrecto. DevOps no quiere decir simplemente que Operaciones se una a Desarrollo.

4 / 50

Cualquier equipo que adopta DevOps podría describirse como un "pacto" (Compact).

¿Qué principio es **más** aplicable para que un pacto funcione?

- A) Una colaboración dinámica
- B) Una comunicación continua
- C) La reducción de los ciclos
- D) Un reparto de las responsabilidades

- A) Incorrecto. La colaboración dinámica no forma parte de los principios del pacto.
- B) Correcto. Los principios de un pacto de DevOps son: la comunicación continua, unos objetivos comunes y claramente definidos, la adaptación dinámica y las reparaciones del entendimiento. *(Bibliografía: A, capítulo 2)*
- C) Incorrecto. La reducción de los ciclos no forma parte de los principios del pacto.
- D) Incorrecto. El reparto de responsabilidades no forma parte de los principios del pacto.

5 / 50

Hay numerosas fuentes de conocimiento, estándares y prácticas disponibles para implementar DevOps.

¿Cuál de los siguientes se considera un elemento **clave** para una implantación con éxito de DevOps?

- A) Nivel 3 del CMMI
- B) Disciplined Agile
- C) ISO 20001
- D) PMBok del PMI

- A) Incorrecto. La certificación CMMI podría ayudar, pero no es clave para la implantación de DevOps, y menos a un nivel específico de CMMI
- B) Correcto. Disciplined Agile se considera un requisito clave para una implantación con éxito de DevOps. *(Bibliografía: C, capítulo 4i)*
- C) Incorrecto. La certificación ISO 20001 podría ayudar, pero no es clave para la implantación de DevOps.
- D) Incorrecto. PMBok del PMI es una publicación de consulta para el gestor de proyectos; no es clave para una implantación de DevOps.

6 / 50

La colaboración es uno de los cuatro pilares de un DevOps eficaz.

¿Por qué es tan importante?

- A) La colaboración es la única manera de lograr un flujo de una pieza en DevOps.
  - B) El valor de negocio de DevOps se alcanza mediante la cooperación entre equipos.
  - C) Las empresas que implementan DevOps son pequeñas, por lo que los equipos deben tener una buena colaboración.
  - D) Impulsa el cambio y es un beneficio fácil de lograr. Además, reduce los costes de desarrollo.
- 
- A) Incorrecto. Se puede lograr un flujo de una pieza incluso si no hay colaboración. Se trata de un mecanismo diferente.
  - B) Correcto. La colaboración entre todos los equipos implicados (incluyendo Desarrollo y Operaciones) es de vital importancia para alcanzar un mayor valor de negocio, mediante una mejora de la comunicación y la automatización y un software de mejor calidad. (*Bibliografía: A, capítulos 6 y 7*)
  - C) Incorrecto. Las empresas que implementan DevOps pueden tener un gran tamaño. Los equipos deben colaborar para aportar el valor de negocio suficiente.
  - D) Incorrecto. Las herramientas impulsan el cambio y son beneficios fáciles de lograr. La colaboración no es algo sencillo. La colaboración puede reducir los costes de desarrollo, pero este no es el principal objetivo. El objetivo es mejorar la calidad.

7 / 50

Usted es el propietario de una pequeña empresa en la que se implementa DevOps, que cuenta con 5 empleados y que crea aplicaciones móviles para niños con discapacidad. Una de las aplicaciones de las que más se enorgullece su equipo es una aplicación para niños con autismo, que les permite planificar el día por sí mismos.

Dado el gran éxito de esta aplicación, se le pide a su equipo que amplíe las funciones de la misma para otras personas que también pueden beneficiarse de planificar su día. Sin lugar a dudas, esta solicitud generará una mayor complejidad en el código y algunas dificultades técnicas que el equipo tendrá que abordar.

Usted espera que la remuneración por este encargo sea elevada, por lo que decide aceptarlo. Sin embargo, tras unas semanas, su equipo está discutiendo. Usted también se enfada y empieza a prestar más atención a lo que hace cada miembro del equipo. Normalmente, trabaja con ellos para poder corregir rápidamente cualquier error que detecte en el código.

Aunque se saca adelante el trabajo suficiente, su equipo permanece enfadado y usted se percata de que se ha perdido una parte de la mentalidad de DevOps.

¿Cuál es la **mejor** estrategia para resolver su problema?

- A) Incorporar más miembros al equipo para aumentar la diversidad, contratando a personal complementario no técnico
  - B) Pedir a expertos en DevOps externos a su empresa que patrocinen y proporcionen mentoría a los miembros de su equipo
  - C) Encontrar un objetivo común y empiecen a trabajar conjuntamente para conseguirlo a fin de aumentar la colaboración
  - D) Reservar unos meses para trabajar en primer lugar en la consolidación del equipo y que todos los integrantes se conozcan
- 
- A) Incorrecto. Ampliar el equipo puede ser aconsejable cuando no se cuenta con los conocimientos necesarios para lograr lo que se debe hacer. Sin embargo, la contratación de personal sin funciones técnicas nunca es aconsejable ni conveniente para conservar la eficacia del equipo. No obstante, en este caso, se está sacando adelante el trabajo suficiente, por lo que esto solo va a agravar los problemas que está experimentando actualmente, ya que hay una falta de confianza y colaboración.
  - B) Incorrecto. Es una idea excelente, pero no resolverá esa parte de la mentalidad de DevOps relativa a la falta de colaboración y afinidad en su empresa. La colaboración aumenta al tener un objetivo en común y alcanzando juntos el éxito.
  - C) Correcto. Esta es una buena manera de resolver los problemas actuales. Debería reducir la cantidad de discusiones, sin dedicar demasiado tiempo a resolver el problema y, potencialmente, perder el encargo. Compartir objetivos hace que aumente la confianza, la afinidad y la colaboración. *(Bibliografía: A, capítulo 7)*
  - D) Incorrecto. Es posible que esto funcione y se forme un equipo cuyos miembros tienen confianza mutua y colaboran entre sí, lo cual debería ser su objetivo. Sin embargo, no es la mejor manera de resolver el problema, ya que corre el riesgo de perder el encargo al dedicar demasiado tiempo a resolver el problema.

8 / 50

Usted cree que su equipo de Desarrollo es realmente un equipo.

¿Cuál es una señal clara e indiscutible de que son un equipo y no un grupo?

- A) El equipo sigue las normas que se han acordado en las reuniones de equipo.
  - B) El equipo mantiene reuniones útiles que lideran ellos mismos.
  - C) El equipo mantiene un ritmo de trabajo sostenible hacia un objetivo común.
  - D) El equipo resuelve los problemas cuestionando al integrante del equipo responsable del asunto.
- 
- A) Incorrecto. Los grupos de personas pueden ser muy buenos en seguir normas. Esto no hace que sean un equipo necesariamente.
  - B) Incorrecto. Algunos grupos de personas pueden llevar a cabo reuniones efectivas. Esta no es una señal inequívoca de que sean un equipo.
  - C) Correcto. Un equipo de verdad garantiza un ritmo de trabajo sostenible y sigue trabajando hacia un objetivo común. (*Bibliografía: A, capítulo 9*)
  - D) Incorrecto. Los verdaderos equipos resuelven los problemas juntos sin cuestionar a sus miembros. DevOps es una cultura libre de culpas.

9 / 50

AppAtoZ ha experimentado un ritmo de crecimiento espectacular en cuanto al desarrollo y despliegue de aplicaciones móviles en las plataformas iPhone y Android.

Los equipos de desarrollo de esta startup han sufrido una presión inmensa para desplegar mejoras rápidas sobre sus aplicaciones móviles en unos plazos muy agresivos. Han estado trabajando una media de 60 horas a la semana durante los últimos seis meses. La dirección ha sido reacia a contratar a más personas para ayudar, y les preocupa más incrementar los beneficios y reducir los costes operativos y de desarrollo.

En los últimos meses ha aumentado la tasa de absentismo de los trabajadores, de bajas por enfermedad e incluso algunos empleados de AppAtoZ dimitieron, lo que provocó una mayor carga de trabajo en los trabajadores que quedaron. El elevado tiempo de contratación y aprendizaje hace que la presión de trabajo que sufre el equipo de desarrollo no se alivie lo suficientemente rápido.

¿Qué estrategias **a largo plazo** deben tenerse en cuenta para resolver el problema del estrés y el cansancio de los empleados?

- A)
  - incorporar al equipo una mezcla de trabajadores permanentes y externos para que puedan gestionar mejor el trabajo
  - que la dirección y el departamento de Desarrollo identifiquen las causas en el ambiente laboral que puedan estar provocando el agotamiento
  - desarrollar un plan para mitigarlas
  
- B)
  - despedir al equipo directivo por su incapacidad
  - contratar un equipo directivo que sepa crear conciliación entre la vida personal y profesional y una cultura que sea sostenible y realista
  - concederle la oportunidad al equipo de Desarrollo de tomarse su tiempo y encontrar su equilibrio
  
- C)
  - darles unos días de descanso a los miembros del equipo de Desarrollo
  - contratar desarrolladores que gestionen el aumento de la carga de trabajo de desarrollo en periodos puntuales del año
  - ayudar a los desarrolladores a encontrar psicólogos o psiquiatras si los necesitan
  
- D)
  - darles unos días de descanso a los miembros del equipo de Desarrollo
  - hacer que la dirección y el equipo de desarrollo identifiquen las causas que puedan estar provocando el agotamiento
  - desarrollar un plan para mitigarlas

**A)** Correcto. Esta es la única respuesta en la que estrategias son a largo plazo. El resto de las respuestas tiene, al menos, una estrategia a corto plazo. (*Bibliografía: A, capítulo 8*)

**B)** Incorrecto. Despedir al equipo directivo no resuelve ningún problema subyacente a largo plazo. Puede ser una corrección a corto plazo. Darse un tiempo para encontrar el equilibrio es una buena idea, pero sin un plan no funcionará de manera espontánea.

**C)** Incorrecto. Unos días libres puede ser una gran idea, pero solo es una solución a corto plazo. La salud mental también es importante, pero, de nuevo, sin cambios subyacentes, termina siendo una solución a corto plazo. Contratar desarrolladores puede funcionar y sí, es una gran idea.

**D)** Incorrecto. Desarrollar un plan e identificar las causas en el entorno de trabajo son muy buenas ideas. Hay una respuesta todavía más adecuada, ya que tomarse unos días libres sólo funciona a corto plazo.



10 / 50

Usted trabaja en un equipo de DevOps como desarrollador. Intenta facilitar la colaboración entre todos los miembros de su equipo. En su equipo hay dos hombres en el puesto de desarrolladores senior y una mujer en el puesto de administradora de sistemas junior.

Los miembros de su equipo han tenido unos comienzos complicados y no han congeniado. Usted intervino en su momento para intentar ayudarles a que se llevaran mejor. Eso funcionó, ya que dejaron de discutir y empezaron a terminar una mayor cantidad de trabajo.

Ahora, se ha percatado del siguiente patrón: la administradora junior siempre se muestra de acuerdo con los desarrolladores senior.

¿Cómo debería reaccionar ante este patrón?

- A) Mientras que los integrantes del equipo no discutan, no debería intervenir en la situación. Tienen una relación cordial, usted no debería alterar esta situación.
  - B) Es importante que Operaciones exprese cualquier inquietud respecto a las compilaciones, por lo que debe orientar a la administradora para que deje de ser complaciente y sea más firme.
  - C) Los desarrolladores senior tienen una mayor responsabilidad, por lo que debería pedirles que sean más amables y que piensen en las inquietudes relacionadas con Operaciones si la administradora junior no lo hace.
  - D) Debería empezar a trabajar en el equipo de Operaciones por el momento, incluso si usted en realidad es un desarrollador, para poder orientar a la administradora junior.
- 
- A) Incorrecto. Esta situación requiere una resolución de conflictos. La administradora junior opta por ser complaciente como estilo de resolución de conflictos, lo cual no es un estilo productivo. En cada puesto se deben desempeñar las funciones asignadas al mismo con el fin de evitar que se añada el riesgo de no abordar las inquietudes a tiempo durante la compilación.
  - B) Correcto. Esta es la solución correcta para esta situación. Esta situación requiere una resolución de conflictos. La administradora junior opta por ser complaciente como estilo de resolución de conflictos, lo cual no es un estilo productivo. Además, acarrea el riesgo de que Operaciones no exprese cualquier inquietud sobre la compilación. (*Bibliografía: A, capítulos 7 y 14*)
  - C) Incorrecto. La responsabilidad es la misma para todos. No está relacionada con el nivel de experiencia o el género. Además, los desarrolladores no detectan los problemas con la compilación del mismo modo que los operadores. Este es el motivo por el que en DevOps se requiere la interacción de los diferentes campos.
  - D) Incorrecto. No puede simplemente cambiar de equipo, incluso si la orientación es una buena forma de proporcionar coaching. Los desarrolladores no detectan los problemas con la compilación del mismo modo que los operadores, por lo que, al hacer esto, perjudicará la calidad de las compilaciones.

11 / 50

¿Por qué Agile y Scrum garantizan un desarrollo de software más rápido y predecible?

- A) Posibilitan una recopilación y un tratamiento de los requisitos mejores y más completos antes del diseño
  - B) Posibilitan el trabajo en equipos autónomos reducidos que se organizan y planifican por sí mismos
  - C) Permiten que el Propietario del Producto participe en las reuniones diarias de planificación
  - D) Permiten que el gestor de proyectos cambie rápidamente las prioridades según sea necesario
- 
- A) Incorrecto. Este enfoque es más bien un modelo en Cascada.
  - B) Correcto. Esto es lo que proponen los principios Scrum y Agile para poder entregar mejor y con mayor rapidez. *(Bibliografía: A, capítulo 4)*
  - C) Incorrecto. Si bien el Propietario del Producto puede participar, no es relevante para garantizar un desarrollo de software más rápido y predecible.
  - D) Incorrecto. No es el gestor de proyectos quien debería cambiar sus prioridades, sino el Propietario del Producto.

12 / 50

¿Qué es gestión simplificada de servicios TI (light-weight ITSM)?

- A) Una gestión de servicios TI (ITSM) orientada estrictamente a la continuidad del negocio
  - B) Una nueva versión de ITIL sugerida como la versión estándar
  - C) Una implementación deficiente de los procesos ITIL
  - D) Una gestión de servicios TI (ITSM) orientada a la gestión de entregas
- 
- A) Correcto. ITIL parece ser muy pesada y poco adecuada para procesos rápidos como los de DevOps. Una gestión simplificada de servicios TI es una versión de ITSM reorientada a DevOps y centrada en la continuidad del negocio con un conjunto mínimo de información necesaria (Minimum Required Information). *(Bibliografía: C, capítulo 4iii)*
  - B) Incorrecto. No se ha propuesto todavía esa versión ITIL.
  - C) Incorrecto. Una gestión simplificada de servicios (light-weight ITSM) no es una implementación deficiente, sino más bien una versión reducida que se centra en la continuidad del negocio y la reducción de la carga de trabajo de gestión.
  - D) Incorrecto. El foco de ITSM es la gestión de servicios TI en su totalidad y no únicamente la gestión de entregas. Dentro del concepto de ITSM, el proceso de gestión de entregas es un proceso que apoya y fundamenta el servicio.

13 / 50

Usted trabaja para una empresa que ha implantado prácticas Lean y Agile. Su consejero delegado no está convencido del valor que aporta implantar, una vez más, algo nuevo: DevOps.

Usted es un experto en DevOps. Considera que DevOps es beneficioso para cualquier empresa que desarrolla software.

¿Qué aporta DevOps a la empresa para la que trabaja?

- A) Se crean mejores Historias de Usuario y requisitos funcionales
  - B) Las entregas se realizan con menos frecuencia, por lo que los clientes no se ven afectados por actualizaciones constantes
  - C) Se forma a los profesionales para que puedan trabajar tanto en Desarrollo como en Operaciones
  - D) Se escribe un código que se entrega con mayor velocidad, apoyando así los resultados de negocio
- 
- A) Incorrecto. Las prácticas Agile ya le ayudan a escribir Historias de Usuario y requisitos funcionales adecuados. DevOps se adhiere a los principios Agile como su base. Incluso añadiendo una implicación temprana en las operaciones, las Historias de Usuario y los requisitos funcionales creados siguiendo Agile no cambiarán en gran medida en DevOps.
  - B) Incorrecto. DevOps usa una entrega más rápida y continua para aportar valor a la empresa a una mayor velocidad. Este también es uno de los principios Lean. Entregar actualizaciones a los clientes con frecuencia no es un problema si las nuevas funcionalidades realmente aportan valor. Además, puede entregar con frecuencia y realizar las actualizaciones para los usuarios finales en horarios programados.
  - C) Incorrecto. El objetivo de DevOps es que Operaciones y Desarrollo trabajen juntos y no como personas que desempeñan ambas funciones a nivel individual. Aunque, en teoría, esto sea posible en empresas muy pequeñas, no es DevOps en sí.
  - D) Correcto. Las prácticas Lean y Agile ya garantizan un énfasis en el valor de negocio y en las peticiones de cambio de las funcionalidades. DevOps añade el aumento de la frecuencia de entregas a la producción en beneficio directo de los resultados de negocio, centrándose en la creación de un flujo de despliegue continuo con una implicación temprana de Operaciones y la mayor automatización posible. *(Bibliografía: C, capítulo 1)*

**14 / 50**

La alta dirección de la empresa solicita al departamento de TI una mejora en el apoyo empresarial y en la alineación con los objetivos del negocio. Como Director de Informática, entre otras medidas, usted decide reducir la carga de trabajo de gestión en el área de Operaciones.

¿En qué manera le ayudará DevOps **principalmente** a mejorar la gestión de niveles de servicio existente?

- A) Abandonando las buenas prácticas de gestión de servicios de TI (ITSM), puesto que no son más que una carga
  - B) Diseñando mejores acuerdos sobre el nivel de operaciones entre Desarrollo y Operaciones
  - C) Implantando un nuevo conjunto de procesos ITSM basados en la versión 3 de la ITIL
  - D) Reajustando la ITSM para utilizar un conjunto de información mínima necesaria
- 
- A) Incorrecto. Abandonar las mejores prácticas de ITSM no mejorará la gestión del nivel de servicio.
  - B) Incorrecto. La formulación de mejores OLA entre Desarrollo y Operaciones no contribuirá a la reducción de la carga de trabajo, sino que más bien aumentará la burocracia. Sin lugar a dudas, no es un área prioritaria para DevOps.
  - C) Incorrecto. Esto generará más procesos que supondrán una carga y que son inadecuados para DevOps.
  - D) Correcto. La producción de una gestión simplificada de servicios TI es un requisito clave para la implantación de DevOps. *(Bibliografía: C, capítulo 4iii)*

**15 / 50**

¿Qué implementación de DevOps es **más** adecuada para una empresa que utiliza el enfoque de System of Record (SoR)?

- A) La colaboración
  - B) Entrega continua
  - C) El método Toyota
- 
- A) Correcto. Esto se centra únicamente en proporcionar servicios de TI de forma rápida y frecuente y un funcionamiento fiable, por lo que es más adecuado para System of Engagement (SoE) y SoR. *(Bibliografía: C, capítulo 8)*
  - B) Incorrecto. Es más adecuado para proveedores de productos digitales.
  - C) Incorrecto. Es más adecuado para proveedores de servicios de TI.

16 / 50

¿Cuál es el **principal** beneficio de usar el sistema Obeya?

- A) Ayuda con las quejas de los clientes, para garantizar que el equipo reciba comentarios suficientes para mejorar constantemente
  - B) Gestiona el estrés en los equipos, de modo que los miembros del mismo puedan mantener un ritmo sostenible
  - C) Mejora el proceso diario de informe de errores, lo cual garantiza que se reduzca el retrabajo y que se pasen los errores a otras estaciones de trabajo
  - D) Una rápida toma de decisiones, basada en el estado actual, acelerando la recopilación y distribución de información
- 
- A) Incorrecto. Obeya no ayuda con las quejas de los clientes.
  - B) Incorrecto. Si bien podría ayudar a gestionar situaciones estresantes, este no es el principal beneficio.
  - C) Incorrecto. Obeya no ayuda a mejorar el proceso de informe de errores.
  - D) Correcto. La sala Obeya, conocida también como "War Room", es una herramienta del sistema de producción de Toyota / Lean que permite a los equipos de proyectos tener toda la información pertinente disponible y a la vista, además de promover interacciones e intercambios de información rápidos entre pequeños equipos, lo cual acelera la recopilación de información para la toma de decisiones. (*Bibliografía: C, capítulo 7iii*)

**17 / 50**

Usted trabaja en una empresa de software que está empezando a implementar DevOps. Se percata de que no sólo debe cambiar la cultura de la empresa, sino también sus prácticas y herramientas. La empresa ya supervisa los proyectos de software con el fin de detectar cualquier problema.

Usted propone que se adopten las medidas siguientes:

- automatizar el proceso de gestión de cambios;
- implantar controles de acceso para evitar que alguien realice cambios sin consentimiento.

¿Por qué es esto necesario?

- A)** La automatización permite que se implementen los cambios con mayor rapidez a la vez que se mantiene la confianza. El control de acceso es necesario para evitar heurísticas de la resolución de problemas y alteraciones imprevistas del servicio.
- B)** La automatización es necesaria para ralentizar el número de cambios. El control de acceso debería evitar que los clientes modifiquen el software por sí mismos, sin que ustedes lo sepan y fuera de su control.
- C)** La automatización ayudará a que el equipo de Operaciones sea innecesario. El control de acceso debe implementarse, ya que los proyectos de DevOps funcionan en un entorno de nube, donde los riesgos asociados son más elevados.
- A)** Correcto. Cita: "Aunque, por lo general, no nos apasione establecer controles y procesos de aprobación, es algo esencial en lo que respecta a su infraestructura de producción. Por ende, puesto que consideramos que debería tratar a su entorno de pruebas del mismo modo que trata a sus entornos de producción, debería aplicarse el mismo proceso a ambos. Es fundamental proteger los entornos de producción para evitar un acceso no autorizado, no solo por parte de personas ajenas a su empresa, sino también de sus propios empleados, incluyendo al personal de Operaciones. De lo contrario, es demasiado tentador, cuando algo sale mal, acceder al entorno en cuestión y hurgar en el mismo para resolver los problemas (un proceso que a menudo recibe cortésmente el nombre de heurística de resolución de problemas). Esto, casi siempre, suele ser muy mala idea por dos razones: en primer lugar, suele conllevar alteraciones en el servicio (la gente suele intentar hacer un reinicio o aplicar paquetes de servicio al azar); en segundo lugar, si algo sale mal más adelante, no queda registrado quién hizo qué y cuándo, por lo que es imposible esclarecer la causa del problema que experimenta. En esta situación, puede ser preferible recrear todo el entorno desde cero para asegurar que está en un estado conocido." (*Literatura: B, capítulo 11*)
- B)** Incorrecto. La automatización ayuda a acelerar la cantidad de cambios que puede realizar sin perder confianza en el proceso. Es posible que el control de acceso impida el acceso de los clientes, pero este no es el principal objetivo.
- C)** Incorrecto. El equipo de Operaciones nunca es innecesario. Aunque sí puede que se sientan menos irritados y agobiados. Los proyectos de DevOps no siempre funcionan en un entorno de nube. Si bien el control de acceso contribuye a la seguridad del software en un entorno de nube, este no es el principal objetivo.

**18 / 50**

La virtualización y la computación en la nube son técnicas que podrían ayudar a facilitar las prácticas de DevOps.

¿Cómo ayudan a DevOps?

- A)** La computación en la nube posibilita el acceso remoto, lo cual otorga a los clientes un mayor control.
- B)** Los entornos virtuales son más fáciles de estandarizar y hacen un uso más eficaz del hardware.
- C)** La infraestructura virtualizada es más fácil de entender y no requiere ningún mantenimiento.

- A) Incorrecto. La computación en la nube sin duda facilita el acceso remoto, pero esto por sí mismo no implica que los clientes tengan un mayor control, en absoluto. Estos dos elementos no ayudan a DevOps en ningún modo.
- B) Correcto. La virtualización facilita la consolidación de la integración continua y de las infraestructuras para pruebas, de modo que pueda ofrecerse como un servicio a los equipos de entrega. Además, hace un uso más eficaz del hardware. La virtualización permite estandarizar una única configuración de hardware para los entornos físicos, pero ejecutar diversidad heterogénea de entornos y plataformas virtuales. *(Bibliografía: B, capítulo 11)*
- C) Incorrecto. La infraestructura virtualizada no es más fácil de entender de por sí. Además, también requiere mantenimiento y gestión. No es similar a la infraestructura local. Por lo tanto, esta no es una razón válida.

### 19 / 50

Los Acuerdos de Nivel de Servicio (SLAs) son importantes para cada proyecto, ya que en ellos se especifica lo acordado con el cliente. No obstante, en DevOps, los SLAs tienen además otra finalidad importante.

¿De qué finalidad se trata?

- A) Los clientes tienen la responsabilidad de crear el SLA para el equipo de DevOps. Por lo tanto, sustituye una tarea formal y la convierte en responsabilidad de Operaciones.
  - B) En el SLA, el cliente puede especificar todos sus requisitos no funcionales, de modo que Desarrollo pueda centrarse en los mismos.
  - C) En el SLA se especifica el nivel de servicio aceptable. Desarrollo debería entender el SLA y apoyar a Operaciones en su cumplimiento.
- 
- A) Incorrecto. El SLA es, en todo momento, un contrato entre el cliente y el proveedor del servicio, por lo que ambas partes tienen algo que decir sobre el contenido.
  - B) Incorrecto. Aunque es posible que algunos de los requisitos no funcionales y funcionales se deriven del SLA, la mayoría de ellos se entregan directamente al equipo DevOps y no a través del SLA. Además, la provisión de requisitos no funcionales no es la finalidad última del SLA.
  - C) Correcto. Por lo general, las condiciones descritas en el SLA suelen ser de mayor relevancia para Operaciones. Desarrollo debería apoyar a Operaciones, facilitando su trabajo en la medida de lo posible. Esto es lo que diferencia a DevOps del desarrollo convencional. *(Bibliografía: B, capítulo 12)*

20 / 50

Cuando se trata de un producto nuevo, su equipo necesitará desarrollar un flujo de despliegue. Como parte de la integración continua hay que definir una la fase de commit (commit stage) en el flujo. Discute esta fase con los integrantes de su equipo.

El Process Master afirma: "Deberíamos establecer la Definición de Hecho antes o durante la etapa de fase de commit. Si el código no está Hecho cuando se hace el commit, el trabajo debería detenerse".

¿Es verdad?

- A) Sí. Si el trabajo no está Hecho, el Process Master no está realizando bien su función. Esto es algo que debería resolverse de inmediato.
  - B) Sí. No debería hacerse un commit de trabajo que no cumple con la Definición de Hecho ya que no está aportando valor al cliente.
  - C) No, la Definición de Hecho se establece solo en las reuniones con el cliente. Esperar a que se haga retrasaría demasiado el trabajo.
  - D) No, el trabajo en el flujo de despliegue debería continuar siempre. Si el código no está Hecho es necesario que esté inactivo.
- 
- A) Incorrecto. El trabajo del Process Master es asegurarse de que existe una Definición de Hecho y cuando se confirma que el código no está Hecho, debería detenerse el trabajo. Sin embargo, el Process Master no estaría necesariamente haciendo mal su trabajo si se hace un commit de trabajo que no cumple la Definición de Hecho.
  - B) Correcto. Cuando el trabajo no está Hecho, no se crea suficiente valor para el cliente como para disparar el flujo de despliegue. Si se tiene en cuenta el flujo de una pieza, esto retrasaría el flujo de otro trabajo más valioso. (*Bibliografía: B, capítulo 3*)
  - C) Incorrecto. La Definición de Hecho es una de las primeras cosas que se acuerdan en un proyecto. No se establece en las reuniones con el cliente. Cuando se comienza a programar ya se debería saber cuál es la Definición de Hecho. Si no fuese así, ¿cómo se sabría cuándo hay que dejar de programar?
  - D) Incorrecto. Que haya problemas con el código o no aporte ningún valor son razones con el suficiente peso como para detener el flujo de despliegue y solucionarlo o para incorporar algo de mayor valor al flujo.



21 / 50

¿Cuál es la mayor ventaja que tiene aumentar la diversidad del equipo de forma que incluya un espectro más amplio de culturas y personalidades?

- A) Aporta mayor cantidad de experiencias y puntos de vista.
  - B) Conlleva menos roces entre los integrantes del equipo.
  - C) Pone límites a la originalidad y la capacidad de proponer nuevas ideas.
  - D) Se tardará más en llegar a un punto concreto de toma de decisiones.
- 
- A) Correcto. La diversidad engloba una amplia variedad de que incluyen factores de raza, género, orientación sexual, clase, nivel educativo, idioma y experiencia laboral. Todos estos factores exclusivos dan a una empresa un mayor número de experiencias y puntos de vista. (*Bibliografía: A, capítulo 7*)
  - B) Incorrecto. Existe la posibilidad de que el aumento de la diversidad suponga un aumento de la presión y los roces ya que los diferentes valores culturales tienen que trabajar juntos.
  - C) Incorrecto. Una mayor diversidad supone una cantidad mayor de puntos de vista. Normalmente esto deriva en una mayor originalidad.
  - D) Incorrecto. De hecho, esto suele considerarse una desventaja. (Aunque en ocasiones los procesos de toma de decisiones más lentos pueden ser positivos). Una mayor diversidad puede conducir a un mayor lapso de tiempo para lograr un consenso.

22 / 50

Usted trabaja para una empresa pequeña que cuenta con un único equipo de DevOps. Su equipo de DevOps está trabajando en una aplicación formada por múltiples componentes. Algunos son nuevos y otros solo requieren actualizaciones.

Por ahora, cada componente tiene su propio flujo de despliegue. El equipo está orgulloso de la entrega continua que realizan en la mayoría de los flujos de despliegue y su producción es elevada y de buena calidad.

¿Qué debería hacer usted?

- A) Mantener los distintos flujos y alentar al equipo a que amplíen la entrega continua
  - B) Conservar únicamente los flujos que tengan una entrega continua y fusionar el resto de flujos sin entrega continua
  - C) Conservar exclusivamente dos flujos diferentes: uno para el desarrollo y otro para el mantenimiento
  - D) Trabajar con su equipo para conseguir un único flujo, explicando los riesgos que entraña tener múltiples flujos
- 
- A) Incorrecto. La existencia de más de un flujo de despliegue por equipo entraña riesgos. Es difícil priorizar los diferentes flujos y tendrá que interrumpirse el trabajo en algunos elementos si se otorga prioridad a algún elemento de otro flujo. Esto conlleva cierto nivel de caos y la realización de varias tareas de manera simultánea, algo que DevOps intenta evitar. La eficacia y el valor empresarial aumentan con el uso de un único flujo de despliegue.
  - B) Incorrecto. Todo flujo de despliegue se beneficia enormemente de la entrega continua. Sin embargo, la existencia de prácticas de entrega continua no reduce los riesgos asociados con la existencia de múltiples flujos de despliegue.
  - C) Incorrecto. Los riesgos son los mismos incluso si solo tiene dos flujos de despliegue, e incluso si estos están divididos a nivel funcional.
  - D) Correcto. Esta es la mejor forma de proceder. Con suerte, su equipo habrá aprendido lo suficiente de su práctica con la entrega continua como para incorporar esas lecciones a todo el flujo. (*Bibliografía: B, capítulo 13*)

23 / 50

Usted está asesorando a Company Builders, una empresa mediana-grande que introdujo los métodos DevOps hace un par de años.

Le han contratado para determinar su madurez actual. Cuando termine, debería ofrecerles algunas sugerencias para mejorar. Quieren saber en qué área deben centrarse para pasar al siguiente nivel de madurez: *nivel 2 - Gestionado cuantitativamente*.

Se da cuenta de que en la mayoría de las áreas se encuentran en el *nivel 1- Uniforme* con dos excepciones:

1. **entornos y despliegue**. Esta área gestiona despliegues orquestados y ha realizado pruebas en los procesos de Despliegue y Roll-back.
2. **gestión de compilaciones (Build management) e integración continua**. En esta área comprueba que existen compilaciones y pruebas automatizadas que se ejecutan periódicamente, y cada compilación puede volverse a crear desde el control de versiones de código fuente mediante un proceso automatizado.

Primero, establezca el nivel de madurez en estas dos áreas basándose en la información obtenida. Después, presente sus recomendaciones sobre lo que se deben centrar en mejorar.

¿En qué dos áreas debería trabajar Company Builders antes de pasar al nivel 2?

- A) "Entornos y despliegue" y "gestión de compilaciones e integración continua" están en el nivel 0. Deberían trabajar en ambas a la vez.
- B) "Entornos y despliegue" y "gestión de compilaciones e integración continua" están en el nivel 1 o por encima. Deberían trabajar en otras áreas de mejora para progresar.
- C) "Entornos y despliegue" está en el nivel 0. "gestión de compilaciones e integración continua" está en el nivel 1. Deberían centrarse primero en "Entornos y despliegue".
- D) "Entornos y despliegue" está en el nivel 2. "gestión de compilaciones e integración continua" está en el nivel 0. Deberían centrarse solo en "gestión de compilaciones e integración continua".
- A) Incorrecto. El área 1 está en el nivel 2 y el área 2 está en el nivel 0.
- B) Incorrecto. El área 1 está en el nivel 2 y el área 2 está en el nivel 0.
- C) Incorrecto. Gestión de compilaciones e integración continua ya está en el nivel 2 así que no necesitan trabajar más en esta área.
- D) Correcto. El área 2 está a nivel 0 y debería madurar en primer lugar al nivel 1 para que la empresa pueda alcanzar el nivel 2. (*Bibliografía: B, capítulo 15*)

## 24 / 50

Su empresa vende un servicio de copia de seguridad en línea a sus clientes. Ahora, uno de los clientes ha pedido que se implanten nuevas funcionalidades en su servicio. Quieren que las nuevas funcionalidades estén listas en el plazo de una semana, de lo contrario buscarán otra empresa que lo haga.

Usted considera que las nuevas funcionalidades son importantes y que el equipo de Desarrollo puede crearlas en poco tiempo. No obstante, se encuentra con los problemas siguientes:

- Los evaluadores (testers) tardan mucho en solucionar los errores.
- Los evaluadores están encontrando errores que los desarrolladores corrigieron hace mucho tiempo.
- Rara vez se puede demostrar que la aplicación está funcionando.
- Rara vez se realizan presentaciones o demos.

¿Cuál es su problema y cómo debería resolverlo?

- A)** Está desplegando códigos con errores. Debería resolver esto entendiendo mejor el proceso de Despliegue, aumentando la colaboración y trabajando con una mayor disciplina.
- B)** Tiene una gestión de la configuración deficiente. Debería resolver esto mejorando la colaboración entre Desarrollo y Operaciones y aumentando la monitorización, el registro de eventos y la virtualización.
- C)** Su proceso de integración continua no se está gestionando adecuadamente. Debería resolver esta situación acelerando las pruebas automatizadas y la fase de commit, así como mejorando la comprensión del proceso de integración continua.
- D)** Su estrategia de pruebas no es eficaz. Debería resolver esto usando pruebas automatizadas y aumentando la colaboración entre los evaluadores y el resto del equipo.
- A)** Correcto. El despliegue de códigos con errores o un despliegue con una frecuencia insuficiente conlleva el retraso en el tiempo de Despliegue, una baja velocidad, escepticismo sobre las fechas de entrega, una pérdida de la confianza en el entorno de integración continua, una mayor tardanza en corregir los errores, que se encuentren errores que los desarrolladores corrigieron hace mucho tiempo y que se realicen pocas demostraciones y presentaciones. La solución proporcionada resuelve estos problemas. (*Bibliografía: B, capítulo 15*)
- B)** Incorrecto. Una gestión de la configuración deficiente conduce específicamente a fallos inexplicables en la producción, despliegues ingestionables, un mayor tiempo en configurar el entorno y un largo tiempo de recuperación en caso de fallo. La solución proporcionada resuelve estos problemas.
- C)** Incorrecto. Una gestión inadecuada del proceso de integración continua conduce específicamente a desplegar menos de una vez al día, a una ruptura en la fase de commit y a una prolongada fase de integración entre las entregas. La solución proporcionada resuelve estos problemas.
- D)** Incorrecto. Una estrategia de pruebas ineficaz conlleva la presencia de errores recurrentes, que se dedique mucho tiempo a corregir los errores, recibir numerosas quejas por parte del cliente, un producto de baja calidad y que los desarrolladores sufran estrés. La solución proporcionada resuelve estos problemas.

25 / 50

¿Cuál es uno de los beneficios de la integración continua en el contexto de un DevOps eficaz?

- A) Los ciclos de pruebas exhaustivos antes de la entrega de funcionalidades
  - B) Los largos períodos entre las entregas de funcionalidades
  - C) El aumento de la frecuencia y la puntualidad en las entregas de funcionalidades
- A) Incorrecto. Realizar pruebas adecuadas es algo necesario, pero no tiene por qué tratarse de ciclos exhaustivos. Además, esto no se considera un beneficio.
- B) Incorrecto. Esto es lo contrario a lo que sucede realmente.
- C) Correcto. La integración continua ayuda a aumentar la velocidad y frecuencia de las entregas, ya que la integración está automatizada. (*Bibliografía: B, capítulo 3*)

26 / 50

Una empresa multinacional está encontrando muchos problemas a la hora de hacer el merge de los commits de código que vienen desde sus ubicaciones distribuidas en un repositorio central ubicado en Dallas (Texas). Las ubicaciones distribuidas están en Ciudad de México, París, San Diego e Inglaterra. Se han observado comportamientos incoherentes en cuanto al momento en el que las regiones están haciendo el commit del código y en ocasiones no está claro si las pruebas han dado error o no.

Existen cuatro métodos posibles:

1. No registrar el código cuando la compilación está dañada
2. Realizar las pruebas de confirmación locales siempre antes de hacer el commit o dejar que sea el servidor de integración continua quien lo haga por tí
3. Dejar terminar las pruebas de la fase de commit antes de continuar
4. No desactivar las pruebas que dan error

¿Cuáles de estas opciones es la **más** adecuada para garantizar que las ubicaciones distribuidas corrijan sus puntos débiles?

- A) 1 y 2
  - B) 1, 2 y 3
  - C) 2 y 3
  - D) 2, 3 y 4
- A) Incorrecto. La 1 no es adecuada en este caso. La 3 y la 4 también son importantes.
- B) Incorrecto. La 1 no es importante.
- C) Incorrecto. La 4 también es importante.
- D) Correcto. Tres de los métodos son bastante adecuados para este escenario. No hay pruebas de que estén añadiendo código a una compilación dañada, así que esa no sería adecuada. (*Bibliografía: B, capítulo 3*)

27 / 50

Analice la anatomía de un flujo de despliegue básico.

¿Qué fase confirma que el sistema es válido a nivel funcional y no funcional?

- A) Prueba de aceptación automatizada
  - B) Prueba unitaria y de compilación
  - C) Prueba de aceptación manual
  - D) Control de Versiones
- A) Correcto. La fase de prueba de aceptación automatizada confirma que el sistema está operativo a nivel funcional y no funcional, que su comportamiento cumple con las necesidades de los usuarios y las especificaciones del cliente. *(Bibliografía: B, capítulo 8)*
- B) Incorrecto. Las pruebas de compilación y unitarias garantizan que código sea válido en sí mismo. No comprueba su integración con la compilación existente.
- C) Incorrecto. Podría ser una respuesta correcta. Sin embargo, en un flujo de despliegue en funcionamiento, las pruebas de aceptación deberían estar automatizadas.
- D) Incorrecto. El Control de Versiones se utiliza para arreglar compilaciones dañadas y solucionar problemas y errores. No se usa para ver si el sistema funciona correctamente a nivel funcional y no funcional.

28 / 50

Una de las prácticas DevOps más eficaces es utilizar el mismo procedimiento para realizar el despliegue en todos los entornos en los que esté presente la aplicación. Esto garantiza que se ponga a prueba la compilación de forma eficaz. Está utilizando scripts para automatizar los procesos de compilación y despliegue.

¿Cuál es la **mejor** manera de hacerlo?

- A) Utilizar un script para cada entorno y guardarlos como parte de un sistema de control de versiones.
  - B) Utilizar un script concreto para cada entorno para corregir las diferencias entre los entornos.
  - C) Utilizar los mismos scripts para todos los entornos aplicando parámetros manuales para configuraciones específicas.
  - D) Utilizar los mismos scripts para el despliegue en cada entorno y gestionar la información de configuración por separado.
- A) Incorrecto. Esta respuesta no es una buena opción debido al esfuerzo de mantenimiento y a los errores potenciales que se pueden introducir si tenemos esta dificultad añadida.
- B) Incorrecto. Los scripts distintos pueden crear un problema con las modificaciones creando, por tanto, problemas durante el proceso que serán difíciles de encontrar y resolver.
- C) Incorrecto. En una compilación y en un despliegue no debería haber interacciones manuales, este proceso debería estar automatizado para hacerlo más rápido y minimizar los errores.
- D) Correcto. Los scripts deberían ser los mismos para asegurar que ambos procesos, el de compilación y el de entrega se pueden poner a prueba con eficacia. Las diferencias entre los entornos, como URI, IP, etc., se deben gestionar dentro del proceso de gestión de la configuración. *(Bibliografía: B, capítulo 6)*

29 / 50

Tras la entrega de un nuevo servicio de TI, un proceso (job) en Operaciones termina de manera inesperada.

¿Cuál **no** puede ser una causa válida para ello?

- A) El gatekeeper no tuvo en cuenta la interrelación entre las tareas del trabajo y los criterios para la aceptación de servicios.
- B) Los requisitos no funcionales no estaban claros, debido a una historia de usuario incompleta.
- C) El Service Master no llegó a un acuerdo con el usuario acerca del final de la vida útil del servicio que se va a entregar.
- D) No se trabajó lo suficiente en los procesos de desarrollo para cumplir con los criterios de aceptación del servicio.

- A) Incorrecto. Esta es una causa posible.
- B) Incorrecto. Esta es una causa posible.
- C) Correcto. Esta no es una causa posible, ya que esto no puede determinarse de antemano y tendrá que determinarse siempre una vez que el servicio esté activo. (*Bibliografía: C, capítulo 7*)
- D) Incorrecto. Esta es una causa posible.

30 / 50

La empresa AppBC está utilizando DevOps. Han implementado el despliegue continuo y un flujo de despliegue estable, con pruebas de aceptación muy automatizadas y pasan software a producción diariamente.

AppBC tiene una base de datos enorme con muchos usuarios. Disponen de una muy buena estrategia de pruebas de capacidad. Debido a lo voluminoso y complejo del entorno, con cada nueva versión aparecen errores en Producción.

¿Qué estrategia ayudará **más** a AppBC a evitar estos errores?

- A) Introducir el Canary Releases
- B) Automatizar las pruebas de capacidad
- C) Reducir la tasa de entregas
- D) Utilizar despliegues "azul-verde" (Blue-Green deployment)

- A) Correcto. Los despliegues de tipo Canary Release implementan una nueva versión de una aplicación en un subconjunto de servidores de producción para conseguir un feedback rápido. Esto saca a la luz cualquier problema que haya en la nueva versión sin repercutir en la mayoría de los usuarios, ya que el incremento de la carga es gradual, al mismo tiempo que mide los tiempos de respuesta y otras métricas de rendimiento reduciendo así el riesgo que supone el lanzamiento de una nueva versión y facilitando la localización y la solución rápida de errores. (*Bibliografía: B, capítulo 10*)
- B) Incorrecto. En este contexto, las pruebas de capacidad ya deberían estar automatizadas, pero su automatización no ayudará a encontrar los errores en este escenario.
- C) Incorrecto. Esto es totalmente contrario a los métodos DevOps.
- D) Incorrecto. El despliegue "azul-verde" (Blue-Green deployment) exige demasiados recursos que tendrían un coste muy elevado en este escenario. Además, utilizar esta estrategia con una gran base de datos puede causar lapsos de inactividad o situaciones de solo lectura si se necesita revertir alguna acción. Tampoco ayudará a mejorar las pruebas de capacidad.

DevOps toma conceptos muy importantes de Agile, derivado del sistema de producción de Toyota.

¿Por qué el flujo de una pieza es importante para la adopción de DevOps?

- A) Permite que su equipo trabaje a un ritmo sostenible y a una velocidad predecible.
  - B) Permite que el equipo se centre en añadir funcionalidades con el mayor valor de negocio posible.
  - C) Aumenta el reparto de responsabilidades del trabajo entre los diferentes integrantes del equipo.
  - D) Reduce los cuellos de botella limitando el número de tareas que realiza de manera simultánea.
- 
- A) Incorrecto. Esto es lo que se logra mediante el ritmo. Fijando un ritmo sostenible, puede garantizar que este sea predecible, que su equipo no sufra desgaste y que sus empleados disfruten de un buen equilibrio entre la vida profesional y la personal.
  - B) Correcto. Esto es lo que se logra mediante el flujo de una pieza. El flujo de una pieza le permite escoger la funcionalidad o actualizar la funcionalidad que aporta un mayor valor y la sitúa en la posición de siguiente en el flujo. Esto hace que se mantenga ágil. Al trabajar en una única funcionalidad, suele ser capaz de limitar el Work-in-Progress, de modo que también se finalizan las funcionalidades. (*Bibliografía: C, capítulo 7*)
  - C) Incorrecto. Esto es algo importante para DevOps, pero no está directamente relacionado con Work-in-Progress, el flujo de una pieza, el ritmo o Ji Koutei Kanketsu.
  - D) Incorrecto. En esto consiste Work-in-Progress. Al limitar el número de tareas en el que puede trabajar de manera simultánea con su equipo, puede finalizar satisfactoriamente las tareas asignadas. Se evitan los cuellos de botella, ya que usted regula el límite de Work-in-Progress para que siempre haya alguien esperando el trabajo finalizado.



### 32 / 50

La empresa S Corporation, una compañía de tamaño medio de piezas de automóviles, es proveedor de una gran compañía, T Motors Corporation. Proporcionan a T Motors Corporation piezas de automóvil que representan casi el 60% del total de sus ventas.

La junta se reúne para tratar una nueva colaboración. T Motors Corporation exige que S Corporation cambie su modo de entrega al método Just-in-Time o dejarán de trabajar con ellos. S Corporation no podría sobrevivir si perdiera este negocio, de forma que es bastante urgente hacer el cambio a Just-in-Time. El cambio se deberá llevar a cabo dentro de los 6 meses siguientes, por lo que quedan como mucho 5 meses para prepararse.

Una de las implementaciones tiene que ver con el seguimiento de piezas mediante la identificación por radiofrecuencia (RFID). Esto facilitaría la transparencia en el proceso de producción. Se debería llevar a cabo una rápida revisión del proceso actual para facilitar el cambio a un proceso que sea adecuado para su uso con RFID.

Se pide a la directora de informática (CIO) gestionar el proceso de cambio. Cree que será posible si se utiliza el enfoque DevOps para crear una versión mínima. Lo más idóneo sería que se desarrollara primero el concepto para la producción con RFID. En última instancia se implementaría un sistema de control de producción que utilice los datos RFID. Sin embargo, no hay tiempo suficiente para dar estos dos pasos de forma consecutiva. Las tres cosas deberán hacerse al mismo tiempo.

La CIO asigna al proyecto a Em, que es Scrum Master. Desarrollo se prepara para crear un flujo de Despliegue.

Em se percata de que el equipo de Desarrollo es muy entusiasta y trabaja muy duro, pero podrían ser más disciplinados. Además, la frecuencia de entregas debería ser mayor.

¿En qué debería centrarse **primero** Em?

- A) Debería centrarse en la comunicación, ya que es lo más importante en DevOps. Debería empezar rompiendo el hielo en el equipo y establecer algunas normas de comunicación.
- B) Debería empezar por hablar con el equipo del mapa de flujo de valor y crear un flujo de una pieza, ya que el flujo y los procesos simplificados son muy importantes.
- C) Debería comenzar por debatir la infraestructura y el entorno de trabajo con los miembros del equipo, ya que DevOps es más eficaz cuando las herramientas y los métodos funcionan.
- D) Debería comenzar por juntar a todos los interesados, formarles en DevOps y pedirles su apoyo para transmitir el cambio cultural, ya que esto último es lo que DevOps necesita.

**El feedback se encuentra en la página siguiente.**

- A) Incorrecto. Aunque la comunicación es importante para que funcione un equipo DevOps, ya han trabajado juntos antes. Hay cosas más importantes de las que ocuparse. Lo que es más importante, debe cambiarse la perspectiva en cuanto al valor para el cliente. Una vez se haya establecido esta forma de pensar, Em puede pasar a trabajar en pulir las habilidades comunicativas.
- B) Correcto. Se deben simplificar los procesos y debe crearse el mapa de flujo de valor para que el equipo pueda comenzar a proporcionar un valor añadido con el menor esfuerzo posible. Después de esto llegará el momento de definir lo que podría y lo que debería cambiar y aquello que ya funciona perfectamente; incluyendo herramientas, comunicación y cultura. DevOps no debe parecerse ni sentirse de igual forma en todas las empresas, pero sí debe centrarse en el valor añadido para el cliente. *(Bibliografía: B, capítulo 1 y A, capítulo 1 y 2)*
- C) Incorrecto. Las herramientas y la automatización son partes importantes en DevOps y no deberían olvidarse. Sin embargo, no tienen que ser necesariamente lo primero en lo que centrarse. En cambio, es razonable ver si se puede facilitar el proceso completo de producción, acortarlo o reducir sus costes creando un mapa de flujo de valor y hablándolo con el equipo. Lo que dará comienzo al inevitable y necesario ciclo de mejora continua. Todos los demás pasos deberían ser secundarios.
- D) Incorrecto. En muchas compañías, debe darse el cambio cultural. Sin embargo, no es necesariamente lo primero que hay que hacer. De la misma forma que no es necesario que las partes interesadas se reúnan para debatir el cambio de cultura. El cambio cultural debe provenir de cómo se crea el mayor valor para los clientes o las partes interesadas. Podría ser una buena idea pedirles a las partes interesadas que revisen el mapa de flujo de valor y ver lo que puedan aportar.

### 33 / 50

Su equipo de DevOps trabaja bien conjuntamente a un ritmo sostenible. Al integrar un margen de demora suficiente en el proceso, el equipo tiene el tiempo y la concentración suficientes para comprobar y probar cuidadosamente las compilaciones (builds). Actualmente, su equipo realiza pruebas y despliegues manualmente. Su ritmo es lo suficientemente elevado como para aportar regularmente un gran valor a la empresa.

Su consejero delegado le ha pedido su consejo sobre la automatización en dicho equipo.

¿Qué debería aconsejarle?

- A) Automatice todo lo que se pueda, para que el equipo pueda añadir más funcionalidades y demostrar antes el valor de negocio
  - B) Automatice las pruebas de aceptación, pero no los despliegues, ya que el proceso manual es más seguro
  - C) Automatice los despliegues para mejorar el tiempo de ciclo, pero no las pruebas, para poder aprender de los errores
  - D) No añada la automatización a los métodos de este equipo, ya que los métodos que ejecuta el mismo en la actualidad están dando unos resultados magníficos
- 
- A) Correcto. Automatice siempre todo lo que pueda, así obtendrá un mayor control sobre su proceso y se demostrará antes el valor empresarial. *(Bibliografía: B, capítulos 1 y 8)*
  - B) Incorrecto. Automatizar las pruebas de aceptación es una idea estupenda, pero los despliegues manuales no son más seguros.
  - C) Incorrecto. Automatizar las entregas es una idea estupenda, pero no se aprende más de los errores con las pruebas manuales que con las pruebas automatizadas.
  - D) Incorrecto. El equipo hace un trabajo magnífico pero, en cierto modo, está desaprovechando su potencial al no automatizar las actividades que pueden automatizarse.

**34 / 50**

La directora de los servicios informáticos (CIO) asigna al proyecto a su empleado más fiable, Michael, que es Scrum Master. El equipo de desarrollo se prepara para elaborar un flujo de despliegue.

Michael confía en las buenas intenciones y la espontaneidad del equipo de Desarrollo, pero le gustaría que tuvieran una mayor disciplina. Además, la frecuencia de entregas debería ser mayor. Michael quiere que el equipo de Desarrollo implemente versiones con más frecuencia.

Uno de los integrantes del equipo sugiere: "Lo más importante del nuevo flujo de despliegue es automatizarlo. Deberíamos automatizar primero el flujo de despliegue".

¿Es correcta esta sugerencia?

- A) Sí, es correcta. El aspecto más importante, si queremos mejorar la eficacia, es la automatización del flujo de despliegue.
  - B) Sí, es correcta. Si nos centramos en crear un flujo de despliegue automatizado, subsanamos los problemas potenciales que podamos encontrarnos después.
  - C) No, no es correcta. Debería ser prioritario conseguir un flujo de una pieza y un proceso de despliegue estable. La automatización del proceso vendrá después.
  - D) No, no es correcta. Debería automatizarse el proceso de pruebas primero en lugar del flujo de despliegue.
- 
- A) Incorrecto. El flujo de despliegue debería ser siempre de una pieza (one-piece-flow). Puede funcionar correctamente sin la automatización. Una vez que sea estable habrá posibilidad de automatizar el proceso cuando sea viable. Sin embargo, lo prioritario es crear un flujo de despliegue estable.
  - B) Incorrecto. El flujo de despliegue debería ser siempre de una pieza. Puede funcionar correctamente sin la automatización. Una vez sea estable habrá posibilidad de automatizarlo cuando sea viable. Sin embargo, lo prioritario es crear un flujo de despliegue estable.
  - C) Correcto. El flujo de despliegue debería ser siempre de una pieza. Puede funcionar correctamente sin la automatización. Una vez sea estable habrá posibilidad de automatizarlo cuando sea viable. Sin embargo, lo prioritario es crear un flujo de despliegue estable. *(Bibliografía: B, capítulo 5)*
  - D) Incorrecto. La automatización de pruebas es una actividad clave. Sin embargo, cuando hay que decidir entre crear un flujo de despliegue estable y automatizar las pruebas de funcionamiento, debería centrarse siempre en la creación de un flujo de despliegue estable. Una vez conseguido este objetivo, llegará el momento de crear mejoras en la eficiencia a través de la automatización de pruebas.

35 / 50

Su empresa está cambiando y empieza a trabajar con DevOps. Su equipo va a participar en el cambio. Están debatiendo las prácticas idóneas en la commit stage del código.

Su compañero Sun argumenta: "Cuando se daña una compilación (build) y nadie se hace responsable, deberíamos averiguar quién ha sido y reclamarles que la arreglen".

¿Sería una buena idea?

- A) Sí. La única persona que puede arreglar una compilación es quien la ha roto, por lo que hay que identificar quién ha sido incluso si les hace sentir incómodos.
  - B) Sí. Siempre debes hacerte responsable de los daños que provoques al build. Si no es así, los compañeros tienen derecho a imponer esta regla.
  - C) No, DevOps es un entorno libre de culpa. Si un compañero no quiere hacerse responsable, no debemos obligarle.
  - D) No, hay que arreglar la compilación primero. Después ya habrá tiempo para identificar a los responsables y castigarles.
- 
- A) Incorrecto. Probablemente lo más fácil es dejar que la persona que ha creado el problema lo solucione, pero no es necesario. DevOps es un entorno libre de culpa. Si un compañero no quiere hacerse responsable, no debemos obligarle. Obligar a alguien a hacer algo no es nada respetuoso.
  - B) Incorrecto. DevOps es un entorno libre de culpa. Si un compañero no quiere hacerse responsable, no debemos obligarle. Obligar a alguien a hacer algo no es nada respetuoso.
  - C) Correcto. Obligar a alguien a hacer algo no es nada respetuoso. Es aceptable cometer errores. Los integrantes del equipo trabajan conjuntamente para superar los errores y los desafíos. (*Bibliografía: B, capítulo 3, y A, capítulo 4*)
  - D) Incorrecto. La compilación no tiene por qué arreglarse, se puede revertir a una versión anterior. Además, arreglar una compilación puede no ser mala idea, pero castigar a alguien que ha cometido un error sí lo es. DevOps es un entorno libre de culpa. Si un compañero no quiere hacerse responsable, no debemos obligarle. Obligar a alguien a hacer algo no es nada respetuoso.

**36 / 50**

El equipo de Desarrollo de X-AppGo ha encontrado numerosos problemas con sus métodos de prueba actuales. Ahora mismo están utilizando un proceso de pruebas de aceptación manual. Los desarrolladores creen que el conjunto de pruebas unitarias que han creado es lo suficientemente riguroso como para proteger contra las regresiones.

El equipo de Desarrollo tiene que invertir 1 millón de dólares en las pruebas de aceptación manuales para cada entrega. La alta dirección ha exigido que el equipo de Desarrollo implemente la automatización de las pruebas de aceptación para reducir los costes totales de las pruebas y para minimizar la cantidad de errores y regresiones de código que se introducen en el entorno de producción.

¿Qué principios se deben seguir para definir los criterios de aceptación de la aplicación teniendo en cuenta la automatización?

- A) Criterios Agile
  - B) Criterios ATAM
  - C) Criterios INVEST
- A) Incorrecto. Los criterios INVEST se recomiendan cuando se está creando un conjunto de pruebas de aceptación fácil de mantener. No hay información específica sobre por qué ATAM y Agile no son recomendables. Agile no ofrece ninguna orientación o criterio específicos para la automatización de pruebas.
- B) Incorrecto. Los criterios INVEST se recomiendan cuando se está creando un conjunto de pruebas de aceptación fácil de mantener. No hay información específica sobre por qué ATAM y Agile no son recomendables.
- C) Correcto. Las pruebas de aceptación derivan de los criterios de aceptación, por lo que estos criterios deben escribirse teniendo en cuenta la automatización, y deben seguir los criterios INVEST, cuyas siglas significan independiente, negociable, valioso, medible, pequeño y comprobable. (*Bibliografía: B, capítulo 8*)

**37 / 50**

¿Cuál es el mecanismo **más** eficaz para migrar datos de una manera automatizada?

- A) Crear un esquema de Control de Versiones de la base de datos y conservarlo bajo el sistema de Control de Versiones
  - B) Crear y gestionar conjuntos de datos más pequeños, para que la migración sea más sencilla
  - C) Asegurarse de que todos sus scripts se han probado adecuadamente antes de la migración de los datos
  - D) Asegurarse de que cuenta con un procedimiento de marcha atrás para revertir los cambios en caso de que falle la migración
- A) Correcto. El Control de Versiones de su base de datos es el mejor mecanismo para migrar los datos de forma automatizada. (*Bibliografía: B, capítulo 12*)
- B) Incorrecto. Estas acciones no ofrecen el mejor mecanismo para apoyar las migraciones automatizadas y, principalmente, se centran en cómo gestionar los conjuntos de datos de manera más eficaz.
- C) Incorrecto. Esta no es la mejor respuesta, se centra más bien en actividades de prueba y no en actividades de migración.
- D) Incorrecto. Esto se centra en las acciones de recuperación que se pueden adoptar si falla la migración.

**38 / 50**

La empresa X-AppGo ha estado teniendo dificultades con sus procedimientos de marcha atrás (roll-back). A menudo, esto conlleva pérdidas de datos críticos en sus bases de datos de aplicación de producción cuando ejecutan los scripts de roll-back.

¿Cuándo **no** se pueden ejecutar scripts de roll-back sin perder datos críticos?

- A) El script de roll-back elimina los datos utilizados únicamente por la nueva versión.
  - B) El script de roll-back implica trasladar una columna entre diferentes tablas.
  - C) El script de roll-back restaurará datos de tablas temporales.
- 
- A) Incorrecto. El script de reversión sólo elimina datos utilizados únicamente por la nueva versión y no conlleva la pérdida de datos críticos cuando ejecuta la reversión.
  - B) Incorrecto. El script de reversión está modificando el esquema de la base de datos de manera que no conducirá a una pérdida de datos.
  - C) Correcto. En esta situación no será posible ejecutar un script de reversión. (*Bibliografía: B, capítulo 12*)

**39 / 50**

En ACMECONST se están viendo afectados por numerosos errores en la aplicación y el hardware tras desplegar actualizaciones de software en la aplicación y actualizaciones de los enrutadores y conmutadores.

Han tenido muchas dificultades para volver al estado original tras encontrar estos fallos durante sus ventanas de mantenimiento. Como consecuencia, se ha tenido que trabajar durante largas horas en la recuperación, más allá de las ventanas de mantenimiento, y han experimentado una indisponibilidad más prolongada de sus aplicaciones críticas.

El aprovisionamiento automatizado y una infraestructura autónoma pueden ayudar en esta situación, pero se deben tener en cuenta distintos aspectos.

¿Qué elementos deben gestionarse cuidadosamente para reducir el riesgo de sufrir alteraciones a la hora de desplegar al entorno de producción?

- A) Los registros de monitorización detallados para resolver los fallos en las actualizaciones de la aplicación
- B) Los puntos de integración externos, como los sistemas y servicios externos
- C) Las configuraciones del servidor y la información sobre las cuentas de usuario relacionadas
- D) El conjunto de herramientas para automatización del aprovisionamiento y arquitecturas autónomas

- A) Incorrecto. La resolución de problemas tendrá lugar tras la actualización, por lo que no se considera una forma adecuada para reducir el riesgo del despliegue al entorno de producción.
- B) Correcto. Este elemento debe gestionarse cuidadosamente para reducir el riesgo del despliegue a cualquier entorno de producción o similar. (*Bibliografía: B, capítulo 11*)
- C) Incorrecto. Esto ya está cubierto en la información del sistema operativo y su configuración tanto para el entorno de pruebas como para el de producción, por lo que no es un elemento que se gestione por separado.
- D) Incorrecto. La gestión de las herramientas en sí no es importante para reducir el riesgo. Contribuyen a proporcionar la implementación del proceso adecuada y el contexto para producir un suministro automatizado y una infraestructura autónoma.

#### 40 / 50

La empresa X-AppGo ha estado teniendo dificultades con su aplicación principal. La aplicación no está comunicándose adecuadamente con otras aplicaciones externas. Estas aplicaciones externas necesitan obtener variables de datos específicas de manera eficaz, de modo que puedan ejecutarse llamadas específicas. La aplicación principal está siendo desarrollada por un equipo, y la empresa quiere que siga siendo así por motivos de negocio.

Uno de los desarrolladores sugiere que se separe un componente del código de base de X-AppGo para resolver los problemas de intercomunicación.

¿Qué se considera como buenas razones para separar un componente en este caso?

- A) Convertir un conjunto de complementos (plugins) en una base de código monolítico
  - B) Limitar el impacto de los cambios y facilitar las modificaciones al código base
  - C) Habrá que dividir el código base de X-AppGo y que lo gestionen equipos diferentes
  - D) No hay buenas razones que lo justifiquen y su compilación requeriría más tiempo
- 
- A) Incorrecto. La creación de un componente consiste en pasar de un código base monolítico a uno modular basado en componentes, por lo que en realidad es lo contrario.
  - B) Correcto. Nos animan a diseñar y mantener software con una clara delimitación de responsabilidades, lo cual, a su vez, limita los efectos del cambio y hace que el código base sea más fácil de entender y cambiar. (*Bibliografía: B, capítulo 13*)
  - C) Incorrecto. Puesto que no se está separando la aplicación X-AppGo en componentes independientes, no es necesario ir en contra de los motivos de negocio y dividir al equipo.
  - D) Incorrecto. Hay varias razones buenas y la opción 2 es una de ellas. Además, crear un código base modularizado requerirá menos tiempo para compilar y vincular el código, no más.

41 / 50

Hasta las aplicaciones más pequeñas tendrán cierta dependencia de otros componentes o bibliotecas. Por lo tanto, entender y gestionar las dependencias es una actividad clave en el despliegue continuo a fin de mantener el flujo en el *pipeline* de despliegue.

Ha creado una aplicación que utiliza dos bibliotecas. Cada una de estas bibliotecas dependen de una tercera biblioteca subyacente, aunque hacen referencia a versiones diferentes. Esto genera una dependencia específica.

¿Cuál es la **mejor** solución para resolver o evitar esta dependencia?

- A) Unificar todas las bibliotecas, de modo que pueda consultar la biblioteca directamente y evitar el problema
  - B) Gestionar las bibliotecas usando el Control de Versiones, de modo que pueda ver directamente si genera este tipo de dependencia
  - C) Mantener un resumen visual de todas sus dependencias usando notas adhesivas en un tablón grande, de modo que pueda hacer un seguimiento del flujo
  - D) Realizar el check-in únicamente en partes pequeñas de su cadena de herramientas, de modo que pueda depurar con facilidad los problemas que puedan ocurrir en el check-in
- 
- A) Incorrecto. Esto no es buena idea. En su lugar, utilice el Control de Versiones para consultar la última biblioteca o utilice una herramienta de automatización que le pueda ayudar a automatizar usando la versión de bibliotecas más reciente.
  - B) Correcto. Esta es una buena solución. Además, el Control de Versiones en sus bibliotecas le ayuda a depurar problemas derivados del uso de versiones antiguas de su software por parte de los usuarios. Otra buena solución es utilizar una herramienta de automatización que le ayude a utilizar la versión de bibliotecas más reciente. (*Bibliografía: B, capítulo 13*)
  - C) Incorrecto. Es posible que esto ayude a hacer un seguimiento del flujo, pero no es una solución para abordar las dependencias. Los paneles Kanban no le ayudan realmente a implementar el Control de Versiones.
  - D) Incorrecto. En su lugar, debería hacer un check-in de toda la cadena de herramientas. Esto le ayuda a encontrar interdependencias o incompatibilidades de manera más rápida y fiable.



42 / 50

En un entorno de despliegue continuo, es importante que todo esté bajo el Control de Versiones, de modo que pueda detectar los errores rápidamente o realizar una reversión cuando sea necesario.

Sin embargo, **no** es aconsejable mantener algunos archivos binarios en el Control de Versiones.

¿Por qué se hace esta excepción?

- A) Los resultados binarios suelen ser archivos grandes que cambian con cada compilación y se actualizan automáticamente.
  - B) En los archivos binarios trabajan varios miembros del equipo, por lo que no es práctico mantenerlos en el Control de Versiones.
  - C) Los resultados binarios son la entrada para sus compiladores, que ya se conservan en el Control de Versiones.
  - D) No es necesario hacer esto, ya que la recompilación se efectúa como una parte habitual del proceso normal de compilación.
- 
- A) Correcto. En primer lugar, la salida es muy grande y se reproduce para cada check-in que se compila y que supera las pruebas automatizadas. En segundo lugar, se pueden reproducir desde el código fuente volviendo a ejecutar el script de la compilación. (*Bibliografía: B, capítulo 2*)
  - B) Incorrecto. Esta no es la razón.
  - C) Incorrecto. Los resultados binarios son la salida de los compiladores, no la entrada. El resto del razonamiento es correcto.
  - D) Incorrecto. No es necesario recompilar como parte habitual del proceso normal de compilación. Es cierto que la recompilación le daría un nuevo resultado binario.

#### 43 / 50

Quiere adoptar un enfoque global para gestionar toda la infraestructura TI.

¿En qué dos criterios se podrá basar **mejor** este enfoque?

- A)
  - 1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de cambios.
  - 2. Siempre deberá conocer el estado real de su infraestructura por medio de la monitorización y la gestión de eventos.
- B)
  - 1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de cambios.
  - 2. Siempre deberá conocer el estado real de su infraestructura por medio de la instrumentación y la gestión de incidencias.
- C)
  - 1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de versiones.
  - 2. Siempre deberá conocer el estado real de su infraestructura por medio de las incidencias actuales y la gestión de eventos.
- D)
  - 1. El estado deseado de su infraestructura debería especificarse mediante configuraciones sometidas al control de versiones.
  - 2. Siempre deberá conocer el estado real de su infraestructura por medio de la instrumentación y la monitorización.

- A) Incorrecto. El estado deseado de la infraestructura debería especificarse a través de configuraciones sometidas al control de versiones y no a control de cambios. Además, el punto número 2 es incorrecto: la gestión de eventos no aplica en este caso y falta el criterio de la instrumentación.
- B) Incorrecto. El estado deseado de la infraestructura debería especificarse a través de configuraciones sometidas al control de versiones y no a control de cambios. Además, el punto número 2 es incorrecto: la gestión de incidencias no aplica en este caso y falta el criterio la monitorización.
- C) Incorrecto. El punto número 1 es correcto. El punto número 2 no es correcto: no es uno de los criterios en los que se puede basar **mejor** este enfoque.
- D) Correcto. Estos son dos de los criterios en los que se puede basar **mejor** la adaptación de un enfoque global para gestionar toda la infraestructura. (*Bibliografía: B, capítulo 11*)

#### 44 / 50

Los equipos en los que se emplean buenas prácticas de colaboración tienen tickets de tareas sincronizados. Un director de tecnología realizó visitas al Gemba (Go and See) para investigar cómo trabajaba el equipo de Operaciones. El equipo de Operaciones siempre redefine la infraestructura operativa tras una entrega.

¿Cuál es el **mejor** consejo que puede dar para mejorar esta práctica?

- A) No deberían hacer nada. No hay nada que mejorar, ya que la redefinición es algo que siempre debe hacerse.
- B) Deberían examinar cómo conformar la infraestructura operativa y el control de accesos del entorno operativo.
- C) Deberían revisar la infraestructura operativa, de modo que se convierta en un proceso automatizado.
- D) Deberían empezar a compartir sus conocimientos sobre el proceso de despliegue con el equipo de Desarrollo.

- A) Incorrecto. Se trata de una tarea innecesaria y puede mejorarse.
- B) Incorrecto. Aunque esto puede estar bien, seguirán repitiendo esta acción, lo cual es desperdicio (waste).
- C) Incorrecto. No hay una manera clara de empezar a automatizar este proceso sin la participación de Desarrollo.
- D) Correcto. Esto es justo lo que se debe hacer: compartir el conocimiento y seguir avanzando. *(Bibliografía: A, capítulo 17)*

#### 45 / 50

¿Qué momento es el adecuado para que Operaciones informe a Desarrollo de un cambio operativo?

- A) No hay que informar a Desarrollo. Los cambios operativos sólo incumben a Operaciones.
  - B) Inmediatamente. Se debe informar a Desarrollo lo antes posible.
  - C) En el Scrum de Scrums de la mañana siguiente.
  - D) En el momento en que el equipo de Operaciones haya llevado a cabo las pruebas de aceptación.
- 
- A) Incorrecto. Debe informarse a Desarrollo de forma inmediata para que puedan prever posibles riesgos y problemas.
  - B) Correcto. Debe informarse a Desarrollo de forma inmediata para que puedan prever posibles riesgos y problemas. *(Bibliografía: C, Capítulo 5 y 7)*
  - C) Incorrecto. Debe informarse a Desarrollo de forma inmediata para que puedan prever posibles riesgos y problemas.
  - D) Incorrecto. Debe informarse a Desarrollo de forma inmediata para que puedan prever posibles riesgos y problemas.

#### 46 / 50

Usted espera que su organización DevOps madure. Hay varias formas de llevarlo a cabo.

¿Cuál de los siguientes métodos **no** facilitará la maduración de la organización DevOps?

- A) Definir claramente los objetivos como hitos para facilitar que los integrantes del equipo puedan juzgar si sus actividades diarias aportan valor.
  - B) Definir claramente los procesos y apoyar y permitir a los miembros del equipo que mejoren los procesos diariamente.
  - C) Guardar actas de todas las reuniones para que los integrantes del equipo tengan un fácil acceso a la información.
  - D) Supervisar y registrar las actividades diarias para poder identificar el progreso diario y poder celebrarlo.
- 
- A) Incorrecto. Esto es útil para ayudar a madurar a la organización DevOps.
  - B) Incorrecto. Esto es útil para ayudar a madurar a la organización DevOps.
  - C) Correcto. Esto no ayudará a su organización DevOps a madurar. No hay una necesidad estricta de escribir todas las notas de una reunión y volverlas a leer. Es necesario tomar nota de los acuerdos alcanzados en una reunión, pero no es preciso registrar todo lo hablado durante las mismas. *(Bibliografía: B, capítulo 15)*
  - D) Incorrecto. Esto es útil para ayudar a madurar a la organización DevOps.



47 / 50

Usted trabaja para un proveedor de servicios TI. Como parte del plan de continuidad del negocio quiere garantizar que siempre se puedan cumplir los niveles mínimos de servicio acordados.

Quiere garantizar la continuidad de los servicios TI.

¿Cómo podría DevOps ayudarle con la gestión de la continuidad de servicios TI?

- A) Los principios culturales de DevOps, afinidad y colaboración, aseguran que los miembros del equipo DevOps valoren mucho el servicio.
  - B) DevOps prepara los procedimientos de actuación en caso de desastre y los métodos Obeya mediante la incorporación deliberada de caos en el sistema.
  - C) Las medidas de reducción de riesgos y las opciones de recuperación probablemente están programadas debido a que Operaciones está trabajando de forma conjunta con Desarrollo.
  - D) La gestión de nivel de servicio pasa a tener mayor importancia en DevOps, ya que la labor del Process Master es supervisarla.
- 
- A) Incorrecto. Los valores culturales ayudan a que las personas se sientan mejor, trabajen mejor, a un ritmo continuado, a que cometan errores y aprendan de ellos. De por sí, no es útil para la gestión de nivel de servicio.
  - B) Incorrecto. Introducir caos, como hace el Chaos Monkey de Netflix, puede ser útil para que se empiecen a programar las medidas de reducción de riesgos y las opciones de recuperación, pero por sí mismo no es útil para la gestión de nivel de servicio. Es más, el caos en el sistema debería resolverse a través del desarrollo (código), no de soluciones tipo war-room.
  - C) Correcto. Es un proceso sistemático para prevenir, prever y gestionar los riesgos que afectan a la infraestructura y la aplicación que, de no resolverse, pueden conllevar alteraciones e incidencias que pueden interrumpir los servicios. (*Bibliografía: B, capítulo 11 y C, capítulo 4*)
  - D) Incorrecto. La labor prioritaria del Process Master no es mantener la gestión de nivel de servicio bajo control. Al menos no es una prioridad en todos los equipos DevOps.

#### 48 / 50

ACEMCONST ha expandido su presencia global de forma agresiva aumentando el número de contrataciones y de equipos de ingeniería distribuidos por todo el mundo. También ha aumentado su cartera de clientes al ritmo de un 30% anual.

Las decisiones, que antes se tomaban cuando el equipo de ingenieros estaba en la misma sala, ahora se demoran muchísimo, lo que provoca frustración de toda la empresa. Hay que pasar por más capas de autorizaciones y el proceso es más largo, lo que provoca que algunos ingenieros estén desencantados con todo el proceso de toma de decisiones.

También ha aumentado la confusión en cuanto a la propiedad y responsabilidades sobre los problemas que van apareciendo, lo que crea dudas a la hora de tomar decisiones. Los ingenieros piensan también que su creatividad se frena con todos estos procesos y burocracia adicionales, lo que ha empezado a minar su moral.

¿Cuál es la **mejor** manera de afrontar este escenario?

- A) Mantener los procedimientos actuales, pero creando roles, responsabilidades y titularidades claros para cada procedimiento, instaurar un método efectivo para equilibrar la productividad y el riesgo, hacer los cambios de forma gradual y crear entornos seguros para los experimentos.
  - B) Revisar los procedimientos para ver cuáles se pueden simplificar y crear roles, responsabilidades y titularidades claros para cada procedimiento, instaurar un método efectivo para equilibrar la productividad y el riesgo, hacer los cambios de forma gradual y crear entornos seguros para los experimentos.
  - C) Revisar los procedimientos para ver cuáles se pueden simplificar y crear roles, responsabilidades y titularidades claros para cada procedimiento, instaurar un método efectivo para equilibrar la productividad y el riesgo, hacer los cambios de forma gradual y minimizar la cantidad de experimentos para evitar fallos innecesarios de la aplicación.
- 
- A) Incorrecto. Mantener los procedimientos actuales no es una buena idea, ya que ha habido un crecimiento y los procedimientos actuales ya no funcionan. Pero crear un entorno seguro para los experimentos sí lo es.
  - B) Correcto. Estas medidas funcionan bien en una compañía en proceso de crecimiento. Revisar los procedimientos es necesario porque los antiguos ya no funcionan. El entorno seguro para los experimentos también es necesario para facilitar la mejora continua. (*Bibliografía: A, capítulo 15*)
  - C) Incorrecto. Revisar los procedimientos es necesario. Sin embargo, minimizar la experimentación no es una buena idea porque no promovemos la mejora continua, que es necesaria.

49 / 50

Dentro de la empresa X-AppGo hay un desacuerdo entre el equipo de Operaciones que está en Colombia y el equipo de Desarrollo que está en Irlanda debido a que tienen diferentes prioridades y distintos objetivos. Debido a este problema está aumentando la cantidad de tiempo y de esfuerzos que conlleva resolver problemas que afectan a la empresa.

¿Qué métodos clave debería contemplar X-AppGo para reducir los desacuerdos y mejorar la colaboración entre los equipos de Desarrollo y Operaciones?

- A)
  - 1. Permitir que los dos equipos, Operaciones y Desarrollo, trabajen por separado si lo prefieren, con el objetivo de evitar problemas.
  - 2. Conseguir la implicación absoluta del consejo de administración en el apoyo a ambos equipos, Desarrollo y Operaciones.
- B)
  - 1. Conseguir un mentor dentro del consejo de administración de la empresa que hable con el equipo DevOps sobre la importancia de trabajar conjuntamente.
  - 2. Implicar a los equipos de Desarrollo y Operaciones en formaciones sobre los métodos DevOps de forma que aprendan a hacer el trabajo del otro.
- C)
  - 1. Asegurarse de que los equipos de Desarrollo y Operaciones visiten otras empresas donde DevOps funciona correctamente.
  - 2. Aumentar el presupuesto para poder apoyar mejor las crecientes necesidades a las que se enfrentan los dos equipos, Desarrollo y Operaciones.
- D)
  - 1. Recomendar las visitas entre los dos equipos, Desarrollo y Operaciones, para favorecer un acercamiento, aumentar la confianza y el entendimiento.
  - 2. Difundir los conocimientos entre los equipos de Desarrollo y Operaciones para que puedan trabajar de forma más eficaz.

- A) Incorrecto. Se debe animar a los equipos para que trabajen juntos, y no permitir que trabajen por separado. Los equipos se necesitan el uno al otro y necesitan aprender los unos de los otros. Es imposible la colaboración si no se ven. La implicación del consejo de administración en realidad no facilita que los equipos trabajen de forma conjunta.
- B) Incorrecto. Aunque los equipos se pueden sentir más valorados por parte de un mentor del consejo de administración, no les ayudará en la colaboración entre ellos. La formación tiene un impacto limitado Para facilitar que los equipos trabajen mejor juntos solo es necesario que trabajen juntos.
- C) Incorrecto. La visita a otras empresas puede resultar inspiradora, pero DevOps es bastante exclusivo del entorno de cada empresa. Esto no facilita directamente ni la colaboración ni la disminución de los enfrentamientos. En su lugar los equipos deberán trabajar juntos y compartir sus conocimientos. Aumentar el presupuesto puede ser útil si hay demasiado trabajo y muy pocas personas para hacerlo, pero tiene un impacto limitado en lo que se refiere a disminuir los enfrentamientos y mejorar la colaboración.
- D) Correcto. Estos son los métodos más apropiados para apoyar este escenario de reducir los enfrentamientos y mejorar la colaboración entre los equipos de Desarrollo y Operaciones.  
(Bibliografía: A, capítulo 15)

## 50 / 50

Un equipo de desarrollo se ha interesado por DevOps. Fundamentalmente les interesa la integración continua (CI). Actualmente desarrollan y mantienen 3 soluciones importantes y 4 más pequeñas. Utilizan los métodos Scrum. Cada Sprint tarda 4 semanas, lo que conlleva una media de 1 versión confirmada al entorno de pruebas cada 10-15 días y 1 versión al mes a producción. Quieren presentar un caso de negocio cualitativo para que la dirección respalde la inversión y el esfuerzo que supone crear un método de integración continua.

¿Cuál de los beneficios tangibles de la CI beneficia **más** al caso de negocio?

- A) El despliegue una vez al día al entorno de pruebas aumentaría los beneficios de la empresa y reduciría considerablemente los costes de desarrollo.
  - B) Contribuye al espíritu de equipo. Como ya se utiliza Scrum, la CI **no** generaría beneficios cuantificables para la empresa.
  - C) Se incrementa la estabilidad de las entregas y la calidad al mejorar y automatizar las pruebas al tiempo que se facilita e incrementa la velocidad de las entregas.
  - D) El despliegue a producción una vez al día aumentaría los beneficios de la empresa y reduciría considerablemente los costes de desarrollo.
- 
- A) Incorrecto. Un despliegue más rápido al entorno de pruebas está bien y es consecuencia de la CI, pero no aporta ningún beneficio a la empresa.
  - B) Incorrecto. La CI contribuiría a que las entregas a producción fuesen más rápidas y a localizar antes los errores reduciendo también los costes, irrelevante si se usa Scrum o no.
  - C) Correcto. Incrementar la velocidad de las entregas será una consecuencia de la CI, ya que cada cambio estará operativo e integrado en el código principal, listo para ser entregado a producción, gracias a que el producto siempre estará en un estado operativo. Además incrementará la estabilidad y la calidad gracias a encontrar y corregir los errores rápidamente debido a la automatización de las pruebas. (*Bibliografía: B, capítulo 3*)
  - D) Incorrecto. Una entrega a producción más rápida es una de las principales ventajas de la entrega continua y no una ventaja directa de la CI, cuyo objetivo es la entrega en un entorno similar al de producción para asegurar que la versión es operativa y se hayan pasado las pruebas automatizadas.



# Evaluación

En la siguiente tabla se indican las respuestas correctas a las preguntas.

Número	Resposta	Número	Resposta
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



# Contacto EXIN

[www.exin.com](http://www.exin.com)

