



Musterexamen

Ausgabe 201803

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 reproduced, stored, utilized or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written permission from EXIN



Inhalt

Einleitung	4
Musterexamen	5
Antwortschlüssel	27
Evaluierung	67

Einleitung

Dies ist das Musterexamen EXIN DevOps Master™ (DEVOPSM.DE). Es gelten die EXIN Examen Regeln und Vorschriften.

Dieses Musterexamen erfolgt im Multiple-Choice-Verfahren und umfasst 50 Fragen. Von den pro Frage gegebenen Antworten ist jeweils nur eine richtig.

Die maximal erreichbare Punktzahl beträgt 50 Punkte. Jede richtige Antwort zählt einen Punkt. Das Examen gilt als bestanden, wenn ein Kandidat 33 oder mehr Punkte erreicht hat. Die Dauer des Examens ist 120 Minuten.

Viel Erfolg!

Musterexamen

1 / 50

Was ist ein gutes Argument für die Einführung von DevOps in einer Organisation?

- A) DevOps bietet bei neuen Services häufigere Feedback-Zyklen, weil die Entwicklung schneller ist.
- B) DevOps sorgt für bessere Business Continuity und Agilität, weil Prozesse optimiert und nur wertschöpfende Aktivitäten durchgeführt werden.
- C) DevOps sorgt für häufigere Software-Releases und stellt so neue Services Just-in-Time (JiT) bereit.
- D) DevOps vereint Entwicklung und Betrieb und reduziert damit Zeitverschwendung in der Kollaboration

2 / 50

Der CTO denkt, dass es am wirksamsten wäre, bestimmte Lean-Konzepte bei der Umsetzung von DevOps anzuwenden

Welche Prinzipien oder Praktiken von Lean sind bei der Einführung von DevOps am **wirksamsten**?

- A) **Kaizen und 5S.** Da Agile und DevOps auf zentralen Konzepten von Lean basieren und Kaizen und 5S die Grundlage von Lean bilden, sind diese beiden Methoden bei der Einführung von DevOps am wirksamsten.
- B) **Kaizen in Advance.** DevOps funktioniert nur mit Feedback vom Betrieb an die Entwicklung. Kaizen in Advance sorgt für Feedback zu den vorgelagerten Einheiten und unterstützt damit die Anwendung dieses Prinzips in DevOps.
- C) **Obeya System.** DevOps integriert verschiedene Management-Stile und Prozesse. Das Obeya System hilft bei der Visualisierung des Gesamtprozesses und ermöglicht die erfolgreiche Einführung von DevOps.
- D) **One-piece-flow (in einem Rutsch) und JKK.** DevOps profitiert vom Aufbau von Upstream-Prozessen und einem Einzelwertstromfluss. Das Prinzip des One-piece-flow (in einem Rutsch) ermöglicht dies und JKK hilft dabei den Flow zu straffen und umzusetzen.

3 / 50

Was muss gegenüber einem konventionellen Projekt geändert werden, damit ein Projekt in DevOps erfolgreich ist?

- A) Es sollte eine IT-Service Lieferkette mit Pull-System und One-piece-flow (in einem Rutsch) erstellt werden.
- B) Die Entwickler sollten sich dem Betriebsteam anschließen, um die schnelle Instandhaltung der Services sicherzustellen.
- C) Der Betrieb muss für das Entwicklungsteam arbeiten. Deshalb heißt es DevOps.
- D) Die Mitglieder des Betriebsteams sollten sich dem Entwicklungsteam anschließen.

4 / 50

Alle Teams die DevOps einführen, könnten als Teamverbund bezeichnet werden.

Welches Prinzip sorgt am **besten** dafür, dass ein Teamverbund funktioniert?

- A) Dynamische Zusammenarbeit
- B) Laufende Kommunikation
- C) Reduzierung der Zyklen
- D) Gemeinsame Verantwortlichkeit

5 / 50

Für die Implementierung von DevOps stehen verschiedene Quellen an Wissen, Standards und Praktiken zur Verfügung.

Welche der folgenden Quellen gilt für die erfolgreiche DevOps-Implementierung als **entscheidend**?

- A) CMMI Level 3
- B) Disziplinierte Agilität
- C) ISO 20001
- D) PMI PMBok

6 / 50

Kollaboration ist einer der vier Säulen des wirksamen DevOps.

Warum ist dies so wichtig?

- A) Nur mit Kollaboration erreicht man in DevOps One-piece-flow (In einem Rutsch).
- B) Der durch die DevOps-Methode generierte Mehrwert für das Unternehmen wird durch die Zusammenarbeit verschiedener Teams erreicht.
- C) DevOps Organisationen sind klein, die Teams müssen daher gut zusammenarbeiten.
- D) Kollaboration fördert Veränderungen und ist einfach. Sie reduziert außerdem die Entwicklungskosten.

7 / 50

Sie sind der Eigentümer eines kleinen DevOps-Unternehmens mit 5 Mitarbeitern, das mobile Apps für behinderte Kinder baut. Eine der Apps, auf die Ihr Team äußerst stolz ist, ist eine App, mit der autistische Kinder ihren Tag selbst planen können.

Weil die App so erfolgreich war, wird Ihr Team gebeten, sie auszubauen und auch auf andere Menschen anzupassen, die ebenfalls von der selbständigen Planung ihres Tags profitieren würden. Diese Bitte wird den Code unweigerlich komplexer machen und geht mit einigen technischen Herausforderungen einher, die das Team meistern muss.

Sie gehen davon aus, dass Sie für den Auftrag sehr gut bezahlt werden und nehmen diesen an. Nach ein paar Wochen jedoch streitet sich Ihr Team. Sie werden ebenfalls wütend und achten stärker darauf, was die einzelnen Mitglieder Ihres Teams tun. Sie arbeiten regelmäßig gemeinsam mit Ihrem Team und können so alle eventuellen Fehler im Code, die Sie entdecken, schnell korrigieren.

Obwohl genügend Arbeit erledigt wird, ist Ihr Team auch weiterhin wütend und Sie erkennen, dass das Team teilweise nicht die richtige Denkweise (Mindset) für DevOps hat.

Welche Strategie eignet sich am **besten** zur Lösung dieses Problems?

- A) Erweitern Sie Ihr Team um neue Mitglieder und achten Sie dabei auf mehr Diversität, indem Sie nicht-technisches Personal einstellen
- B) Bitten Sie externe DevOps-Experten, ihre Teammitglieder als Sponsoren und Mentoren zu unterstützen
- C) Suchen Sie ein gemeinsames Ziel und arbeiten Sie gemeinsam an der Verbesserung der Kollaboration
- D) Nehmen Sie sich ein paar Monate Zeit und arbeiten Sie zuerst am Team Building, um mehr über die anderen Teammitglieder zu erfahren

8 / 50

Sie halten Ihr Entwicklungsteam für ein echtes Team.

Was ist ein sicheres Zeichen dafür, dass Ihr Entwicklungsteam nicht nur eine Gruppe, sondern ein echtes Team ist?

- A) Das Team hält sich an die Regeln, die in den Team-Meetings vereinbart wurden.
- B) Das Team hält wirksame Besprechungen ab, die von den Mitgliedern selbst geleitet werden.
- C) Das Team arbeitet stetig auf das Erreichen des gemeinsamen Ziels hin.
- D) Das Team löst Probleme durch Befragung der zuständigen Teammitglieder.

9 / 50

AppAtoZ verzeichnet ein phänomenales Wachstum bei der Entwicklung und Bereitstellung von mobilen Anwendungen für iPhone und Android-Plattformen.

Die Entwicklungsteams dieses Startups stehen unter unglaublichem Druck, weil sie für ihre aktuellen mobilen Applikationen schnell Verbesserungen in kürzester Zeit bereitstellen müssen. Die Teams haben in den letzten sechs Monaten im Schnitt 60 Stunden pro Woche gearbeitet. Die Geschäftsführung ist sehr zurückhaltend was Neueinstellungen anbetrifft und eher darauf bedacht, die Umsatzerlöse zu steigern und gleichzeitig die operativen Kosten und die Entwicklungskosten zu senken.

In den letzten Monaten war eine Erhöhung der Fehlzeiten festzustellen. Mitarbeiter meldeten sich krank und einige Mitarbeiter von AppAtoZ kündigten sogar, was die Arbeitsbelastung der aktuellen Mitarbeiter noch weiter erhöht. Aufgrund der für die Einstellung und Einarbeitung eines neuen Mitarbeiters benötigten Zeit wird sich die Arbeitsbelastungen des Entwicklungsteams nicht so schnell verringern.

Welche **langfristigen** Strategien sind hinsichtlich Mitarbeiter-Burnout und Stress zu berücksichtigen?

- A)
 - Verstärken Sie das Team mit einer Mischung aus angestellten und freien Mitarbeitern und Freiberuflern, um die Arbeitsbelastung zu bewältigen.
 - Sorgen Sie dafür, dass die Geschäftsführung und das Entwicklungsteam zu einem Burnout beitragenden Faktoren der Arbeitsumgebung identifizieren.
 - Entwickeln Sie einen Plan, der diese Faktoren anspricht.

- B)
 - Entlassen Sie das Führungsteam, da dieses unfähig ist.
 - Stellen Sie ein kompetenteres Führungsteam ein, das weiß, wie man für eine gute Work-Life-Balance und eine nachhaltige und realistische Kultur sorgt.
 - Geben Sie dem Entwicklungsteam die Möglichkeit, sich die Zeit zu nehmen, um ins Gleichgewicht zu kommen.

- C)
 - Veranlassen Sie die Mitglieder des Entwicklungsteams, sich frei zu nehmen.
 - Arbeiten Sie mit freien Mitarbeitern zusammen, um die höhere Arbeitsbelastung in Spitzenzeiten abzufangen.
 - Fordern Sie die Entwickler auf, sich bei Bedarf professionelle psychologische Hilfe zu holen.

- D)
 - Veranlassen Sie die Mitglieder des Entwicklungsteams, sich frei zu nehmen.
 - Sorgen Sie dafür, dass Führung und Entwicklung die zu einem Burnout beitragenden Faktoren der Arbeitsumgebung identifizieren.
 - Entwickeln Sie einen Plan, der diese Faktoren anspricht.

10 / 50

Sie arbeiten in einem DevOps-Team als Entwickler und versuchen die Zusammenarbeit zwischen allen Mitgliedern des Teams zu erleichtern. Ihr Team umfasst zwei männliche Senior-Entwickler und eine Junior-Systemadministratorin.

Ihre Team-Mitglieder hatten einen schwierigen Anfang und haben sich nicht vertragen. Sie sind damals eingeschritten und haben versucht, zwischen den Mitgliedern zu vermitteln. Das hat funktioniert, denn Ihr Team streitet sich nicht mehr und erledigt viel mehr Arbeit.

Jetzt bemerken Sie ein neues Muster. Die Junior-Administratorin stimmt den Senior-Entwicklern bei allem zu.

Wie sollten Sie auf dieses Muster reagieren?

- A) Solange sich die Teammitglieder nicht streiten, sollten Sie gar nichts unternehmen. Endlich verstehen sie sich und das sollten Sie nicht stören.
- B) Es ist wichtig, dass der IT-Betrieb alle Bedenken bezüglich eines Builds äußert. Sie sollten die Systemadministratorin daher anleiten weniger entgegenkommend zu sein und sich besser durchzusetzen.
- C) Die Senior-Entwickler haben eine größere Verantwortung. Sie sollten Sie daher bitten, netter zu sein und an die Bedenken des IT-Betriebs zu denken, falls die Junior-Systemadministratorin dies nicht tut.
- D) Obwohl Sie selbst ein Entwickler sind, sollten Sie erst einmal im Betriebsteam mitarbeiten, um die Junior-Administratorin anzuleiten.

11 / 50

Warum versprechen Agile und Scrum eine schnellere und besser planbare Software-Entwicklung?

- A) Weil sie vor der Entwicklung eine bessere und vollständige Erfassung und Bearbeitung der Anforderungen ermöglichen
- B) Weil sie kleine, autonome Teams ermöglichen, die sich selbst organisieren und planen
- C) Weil sie dem Product Owner die Teilnahme an täglichen Einsatzbesprechungen ermöglichen
- D) Weil sie es dem Projektmanager ermöglichen, seine Prioritäten bei Bedarf schnell zu ändern

12 / 50

Was versteht man unter „Lightweight ITSM“?

- A) Ein ITSM, das sich an der Business Continuity orientiert
- B) Eine neue ITIL-Version, die als Standard vorgeschlagen wurde
- C) Eine schlechte Umsetzung der ITIL-Prozesse
- D) Ein ITSM, das sich am Release Management orientiert

13 / 50

Sie arbeiten für ein Unternehmen, das Lean- und Agile-Praktiken umsetzt. Ihr CEO ist nicht überzeugt, dass sich die Implementierung einer weiteren neuen Methode, nämlich DevOps, lohnt.

Sie sind DevOps-Experte. Sie sind der Meinung, dass DevOps allen Unternehmen, die Software entwickeln, Vorteile bietet.

Welche Vorteile bietet DevOps für Ihr Unternehmen?

- A) Mit DevOps werden bessere User Stories und fachliche Anforderungen erstellt.
- B) Mit DevOps verringert sich die Zahl der Releases, so dass Kunden sich nicht regelmäßig mit Updates herumschlagen müssen.
- C) Mit DevOps kann man Experten schulen, die sowohl in der Entwicklung als auch im Betrieb arbeiten können.
- D) Mit DevOps erstellt man Code, der schneller bereitgestellt wird und dies wirkt sich positiv auf die Unternehmensergebnisse aus.

14 / 50

Die Geschäftsführung fordert die IT-Abteilung auf, den Business Support und die Anpassung an die Unternehmensziele zu verbessern. Sie sind der CIO und entschließen sich u.a. die Arbeitsbelastung im IT-Betrieb zu reduzieren.

Wie hilft Ihnen DevOps am **besten** dabei, Ihr bestehendes Service Level Management zu verbessern?

- A) Durch den Verzicht auf ITSM Best Practices, denn diese sind einfach ein schwerfälliger Ansatz
- B) Durch die Erstellung besserer Operating Level Agreements (OLA) zwischen Entwicklung und Betrieb
- C) Durch die Umsetzung neuer ITSM-Prozesse basierend auf ITIL Version 3
- D) Durch die Anpassung von ITSM zur Einführung des Konzepts der "erforderlichen Mindestanforderungen" (MRI)

15 / 50

Welche Umsetzung von DevOps eignet sich für ein Unternehmen, das mit dem System-of-Records-Ansatz (SoR) arbeitet am **besten** ?

- A) Kollaboration
- B) Continuous Delivery (Kontinuierliche Lieferung)
- C) Die Toyota-Methode

16 / 50

Was ist der **größte** Vorteil, den die Verwendung des Obeya-Systems bietet?

- A) Das System erleichtert Kundenbeschwerden und sorgt so dafür, dass das Team genügend Feedback für die kontinuierliche Verbesserung bekommt.
- B) Das System geht mit Stress innerhalb des Teams um, damit die Teammitglieder ein nachhaltiges Tempo beibehalten können.
- C) Das System verbessert die tägliche Auswertung von Fehlern, sorgt damit für weniger Nacharbeit und reduziert die Weitergabe von Bugs an andere Arbeitsplätze
- D) Das System ermöglicht ausgehend von dem aktuellen Status durch die schnelle Erhebung und Weitergabe von Informationen eine schnelle Entscheidungsfindung.

17 / 50

Sie arbeiten in einem Software-Unternehmen, das gerade erst beginnt, DevOps einzusetzen. Sie realisieren, dass sich nicht nur die Kultur der Organisation, sondern auch ihre Praktiken und Tools ändern sollten. Das Unternehmen überwacht bereits Softwareprojekte bezüglich eventueller Probleme.

Sie schlagen die Ergänzung folgender Maßnahmen vor:

- Automatisierung des Change Management-Prozesses
- Implementierung von Authentifizierung, damit nur autorisierte Änderungen durchgeführt werden

Warum ist dies notwendig?

- A) Durch Automatisierung sollen Änderungen schneller und ohne Vertrauensverlust umgesetzt werden. Durch Authentifizierung sollen Problemlösungsheuristiken und ungeplante Serviceausfälle vermieden werden.
- B) Durch Automatisierung soll die Zahl der Änderungen verlangsamt werden. Durch Authentifizierung sollen Kunden daran gehindert werden, die Software selbst, ohne Ihr Wissen und ohne Steuerung Ihrerseits zu ändern.
- C) Die Automatisierung trägt dazu bei, das IT-Betriebsteam überflüssig zu machen. Die Authentifizierung muss umgesetzt werden, weil DevOps-Projekte in einer Cloud-Umgebung durchgeführt werden, die mit höheren Risiken einhergeht.

18 / 50

Virtualisierung und Cloud Computing sind Techniken die DevOps-Praktiken unterstützen und erleichtern können.

Inwiefern unterstützen sie DevOps?

- A) Cloud Computing ermöglicht Remote-Zugriff und gibt dadurch dem Kunden mehr Kontrolle.
- B) Virtuelle Umgebungen lassen sich einfacher standardisieren und nutzen Hardware effizienter.
- C) Virtuelle Infrastruktur ist leichter verständlich und erfordert keine Instandhaltung.

19 / 50

Service Level Agreements (SLA) sind für jedes Projekt wichtig, weil das, was Sie mit dem Kunden vereinbaren, spezifiziert wird. Bei DevOps dienen die SLA darüber hinaus noch einem weiteren wichtigen Zweck.

Welcher Zweck ist dies?

- A) Die Kunden sind dafür verantwortlich, die SLA für das DevOps-Team zu erstellen. Es ersetzt daher eine förmliche Aufgabe des Betriebsteams.
- B) Im SLA kann der Kunde alle seine nicht fachlichen Anforderungen spezifizieren, damit sich das Entwicklungsteam auf diese konzentrieren kann.
- C) Das SLA spezifiziert das akzeptable Service Level. Die Entwicklung sollte das SLA verstehen und den Betrieb dabei unterstützen, das SLA aufrechtzuerhalten.

20 / 50

Ihr Team muss für ein neues Produkt eine Deployment Pipeline entwickeln. Im Rahmen der Continuous Integration (kontinuierlichen Integration) müssen Sie die Commit-Phase der Pipeline definieren. Sie erörtern diese Phase mit den Mitgliedern Ihres Teams.

Der Process Master sagt: „Die Definition of Done (Definition von 'fertiggestellt') sollte in oder vor der Commit-Phase festgelegt werden. Ist der Code bei der Durchführung des Commits nicht Done (fertiggestellt), sollte die Arbeit eingestellt werden.“

Ist diese Aussage richtig?

- A) Ja. Ist die Arbeit nicht Done (fertiggestellt), dann hat der Process Master seine Aufgabe nicht gut erfüllt. Dies sollte unmittelbar behoben werden.
- B) Ja. Mit Arbeit, die nicht Done (fertiggestellt) ist, sollte kein Commit durchgeführt werden, da dies dem Kunden keinen Mehrwert bietet.
- C) Nein. Die Definition of Done (Definition von 'fertiggestellt') wird nur in Besprechungen mit dem Kunden festgelegt. Darauf zu warten, würde die Arbeit zu sehr verzögern.
- D) Nein. Die Arbeit in einer Deployment Pipeline sollte kontinuierlich weitergeführt werden. Ist der Code nicht Done (fertiggestellt), dann muss er nur inaktiv sein.

21 / 50

Was ist der **größte** Vorteil, den mehr Teamdiversität mit einer größeren Bandbreite an persönlichen Hintergründen und Kulturen bietet?

- A) Es führt zu mehr Erfahrungen und unterschiedlichen Perspektiven.
- B) Es führt zu weniger Reibung im Team.
- C) Diversität begrenzt die Originalität und die Möglichkeit neuer Einblicke.
- D) Es dauert länger bis man zu einer spezifischen Entscheidung kommt.

22 / 50

Sie arbeiten für ein kleines Unternehmen mit nur einem DevOps-Team. Ihr DevOps-Team arbeitet an einer Anwendung, die mehrere Komponenten umfasst. Manche dieser Komponenten sind neu, andere müssen lediglich aktualisiert werden.

Derzeit hat jede Komponente ihre eigene Deployment Pipeline. Das Team ist stolz, dass es bei den meisten Deployment Pipelines Continuous Delivery (kontinuierliche Lieferung) durchführt. Das Team produziert viel und seine Ergebnisse sind qualitativ hochwertig.

Was sollten Sie tun?

- A)** Sie sollten die verschiedenen Pipelines beibehalten und das Team ermutigen, Continuous Delivery noch weiter auszubauen.
- B)** Sie sollten nur die Pipelines beibehalten, in denen Continuous Delivery implementiert ist und die anderen Pipelines ohne Continuous Delivery zusammenlegen.
- C)** Sie sollten nur zwei Pipelines haben: Eine für die Entwicklung und eine für die Instandhaltung
- D)** Sie sollten Ihr Team überzeugen, sich auf eine Pipeline zu beschränken und ihm die Risiken erklären, die mit mehreren Pipelines einhergehen.

23 / 50

Sie bewerten das mittelständisch bis große Unternehmen Company Builders, das vor einigen Jahren die DevOps-Praktiken eingeführt hat.

Company Builders hat Sie damit beauftragt, den aktuellen DevOps-Reifegrad im Unternehmen zu bestimmen. Danach sollen Sie Verbesserungsvorschläge unterbreiten. Das Unternehmen möchte wissen, auf welchen Bereich es sich konzentrieren soll, um die nächste Stufe zu erreichen: *Stufe 2 - Quantitativ gesteuert*.

Sie stellen fest, dass der Reifegrad in den meisten Bereichen der *Stufe 1 - konsistent - entspricht*. Es gibt jedoch zwei Ausnahmen:

1. **Environments and Deployment.** Dieser Bereich ist für abgestimmte Deployments zuständig und prüft Release- und Rollback-Prozesse.
2. **Build Management and Continuous Integration (kontinuierliche Integration).** In diesem Bereich werden regelmäßig automatisierte Builds und Tests durchgeführt. Jeder Build kann in einem automatisierten Prozess vom Source Code heraus wiederhergestellt werden.

Bestimmen Sie mit Hilfe der bereitgestellten Informationen den Reifegrad dieser beiden Bereiche. Geben Sie dann Ihre Empfehlung ab, was schwerpunktmäßig verbessert werden sollte.

Auf welchen dieser beiden Bereiche sollte das Unternehmen Company Builders sich konzentrieren, bevor es zur Stufe 2 übergeht?

- A) Der Reifegrad von Environments and Deployment sowie Build Management und Continuous Integration (kontinuierliche Integration) entspricht jeweils der Stufe 0. Es sollte an beiden Umgebungen gleichzeitig gearbeitet werden.
- B) Environments and Deployment sowie Build Management und Continuous Integration (kontinuierliche Integration) entspricht jeweils der Stufe 1 oder höher. Um Fortschritte zu erzielen, sollte der Reifegrad in anderen Bereichen gesteigert werden.
- C) Der Reifegrad von Environments und Deployment entspricht der Stufe 0. Der Reifegrad von Build Management and Continuous Integration (kontinuierliche Integration) entspricht der Stufe 1. Das Unternehmen sollte sich zuerst auf Environments and Deployment konzentrieren.
- D) Der Reifegrad von Environments and Deployment entspricht der Stufe 2. Der Reifegrad von Build Management and Continuous Integration (kontinuierliche Integration) entspricht der Stufe 0. Das Unternehmen sollte sich ausschließlich auf Build Management and Continuous Integration (kontinuierliche Integration) konzentrieren.

24 / 50

Ihr Unternehmen verkauft seinen Kunden einen Online-Backup-Dienst. Jetzt hat einer der Kunden Sie gebeten, neue Funktionen in Ihren Service zu implementieren. Der Kunde möchte die neuen Funktionen innerhalb einer Woche. Anderenfalls sucht er sich einen anderen Anbieter.

Sie glauben, dass die neuen Funktionen wichtig sind und wissen, dass das Entwicklungsteam diese schnell bauen kann. Sie stoßen jedoch auf folgende Probleme:

- Die Tester brauchen lange, um Bugs zu schließen.
- Die Tester finden Bugs, die von den Entwicklern schon lange behoben wurden.
- Demonstrationen der funktionierenden Anwendung sind selten.
- Showcases finden nur selten statt.

Was ist Ihr Problem und wie sollten Sie dieses lösen?

- A)** Sie stellen fehlerhafte Codes bereit. Um dieses Problem zu beheben, sollten Sie für ein besseres Verständnis des Deployment-Prozesses sorgen, die Kollaboration steigern und disziplinierter arbeiten.
- B)** Sie haben ein schlechtes Configuration Management. Um dies zu lösen, sollten Sie die Kollaboration zwischen Entwicklung und Betrieb fördern und stärker auf Überwachung, Protokollierung und Virtualisierung setzen.
- C)** Ihr Prozess der Continuous Integration (kontinuierlichen Integration) ist nicht ordentlich verwaltet. Um dies zu lösen, sollten Sie automatisierte Tests und die Commit-Phase beschleunigen und ein besseres Verständnis des Prozesses der Continuous Integration fördern.
- D)** Ihre Teststrategie ist nicht wirksam. Um dies zu lösen, sollten Sie die Tests automatisieren und die Kollaboration zwischen den Testern und dem restlichen Team verstärken.

25 / 50

Welchen Vorteil bietet die Continuous Integration (kontinuierliche Integration) im Rahmen des wirksamen DevOps?

- A)** Umfassende Testzyklen vor dem Release von Funktionen
- B)** Lange Zeiträume zwischen den einzelnen Releases von Funktionen
- C)** Häufigere und pünktlichere Releases von Funktionen

26 / 50

Eine multinationale Organisation will die Commits aus den dezentralen Standorten in einer zentralen Ablage in Dallas, im US-Bundesstaat Texas, zusammenführen und hat dabei einige Probleme. Die dezentralen Standorte sind u.a. Mexico City, Paris, San Diego und England. Die Regionen führen ihre Commits zu unterschiedlichen Zeiten durch. Manchmal ist es unklar, ob Tests fehlgeschlagen sind oder nicht.

Vier verschiedene Praktiken sind möglich:

1. Auf einen fehlgeschlagenen Build sollte man keinen Code einchecken.
2. Lassen Sie alle Commit Tests vor dem eigentlichen Commit immer lokal laufen oder sorgen Sie dafür, dass der für die Continuous Integration (kontinuierliche Integration) zuständige Server dies für Sie erledigt.
3. Warten Sie bis die Commit-Tests abgeschlossen sind, bevor Sie weitermachen.
4. Fehlgeschlagene Tests sollten nicht auskommentiert werden

Welche dieser Lösungen eignen sich **am besten**, um sicherzustellen, dass die dezentralen Standorte ihre aktuellen Probleme ansprechen?

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

27 / 50

Denken Sie an die Anatomie einer elementaren Deployment Pipeline.

Welche Stufe stellt sicher, dass das System auf funktionaler und nicht-funktionaler Ebene funktioniert?

- A) Automatisierte Abnahmetests
- B) Build und Unit Test
- C) Manuelle Abnahmetests
- D) Versionskontrolle

28 / 50

Eine der Best Practices von DevOps besteht darin, für das Software-Deployment in jeder Umgebung, in der Ihre Anwendung läuft, den gleichen Prozess zu verwenden. So wird sichergestellt, dass der Build wirksam getestet wird. Sie automatisieren Ihren Prozess für den Build und die Bereitstellung mit Hilfen von Skripten.

Welche Methode eignet sich hierfür **am besten**?

- A) Sie nutzen pro Umgebung ein Skript und pflegen diese Skripte im Rahmen des Versionskontrollsystems.
- B) Sie nutzen für jede Umgebung ein spezifisches Skript, um die Unterschiede zwischen den Umgebungen zu berücksichtigen.
- C) Sie nutzen für jede Umgebung das gleiche Skript und verwenden für spezifische Konfigurationen manuelle Parameter.
- D) Sie nutzen für das Deployment in den einzelnen Umgebungen das gleiche Skript und verwalten die Konfigurationsinformationen separat.

29 / 50

Nach dem Release eines neuen IT Service fällt ein Job im IT-Betrieb unerwartet weg.

Was ist **kein** stichhaltiges Argument hierfür?

- A) Der Gatekeeper hat die Zusammenhänge zwischen den Arbeitsaufgaben und den Kriterien für die Serviceabnahme (SAC) nicht berücksichtigt.
- B) Die nicht-funktionalen Anforderungen waren aufgrund einer unzureichenden User Story nicht eindeutig.
- C) Der Service Master und der Benutzer konnten sich nicht auf das End-of-Life (EOL) des bereitzustellenden Service einigen.
- D) Wie die Kriterien für die Serviceabnahme (Service Acceptance Criteria, SAC) belegen, gab es zu wenig Arbeit in den Entwicklungsprozessen.

30 / 50

Das Unternehmen AppBC arbeitet mit DevOps. Das Unternehmen hat Continuous Deployment (kontinuierliche Bereitstellung) umgesetzt, verfügt über eine solide Deployment Pipeline mit hoch automatisierten Abnahmetests und liefert täglich neue Software an die Produktivumgebung.

AppBC hat eine große Datenbank und viele Benutzer. Das Unternehmen verfügt über eine umfassende und solide Strategie für Kapazitätstests. Da die IT-Umgebung von AppBC relativ groß und komplex ist, tauchen mit jeder neuen Version einige Bugs in der Produktivumgebung auf.

Welche Strategie könnte AppBC am **besten** helfen, diese Bugs zu vermeiden?

- A) Die Einführung von Canary-Releases
- B) Die Automatisierung der Kapazitätstests
- C) Die Senkung der Lieferrate
- D) Die Einführung von Blue-Green Deployment

31 / 50

DevOps bedient sich wichtiger Konzepte von Agile, die von dem Toyota Produktions-System (TPS) abgeleitet wurden.

Warum ist One-piece-flow (in einem Rutsch) bei der Einführung von DevOps wichtig?

- A) Es ermöglicht Ihrem Team, in einem nachhaltigen Tempo mit planbarer Teamgeschwindigkeit zu arbeiten.
- B) Es hilft dem Team, sich auf die Ergänzung von Funktionen zu konzentrieren, die dem Unternehmen möglichst viel Mehrwert bieten.
- C) Es stärkt die gemeinsame Verantwortung der Teammitglieder für die Arbeit.
- D) Es begrenzt die Anzahl der gleichzeitig erledigten Aufgaben und reduziert so Engpässe.

32 / 50

Die S-Gruppe, ein mittelständischer KfZ-Zulieferer beliefert ein großes Unternehmen, die T Motors-Gruppe mit KfZ-Teilen. Das Geschäft mit der T Motors-Gruppe beträgt fast 60% des Gesamtumsatzes der S-Gruppe.

Der Vorstand trifft sich, um über die neue Zusammenarbeit zu sprechen. Die T Motors-Gruppe verlangt, dass die S-Gruppe auf Just-in-Time-Lieferung umstellt. Anderenfalls wird die T Motors-Gruppe ihr Geschäft von der S-Gruppe abziehen. Die S-Gruppe würde das nicht überleben, die Umstellung auf Just-in-Time (JiT) ist damit dringend. Die Änderung muss innerhalb von 6 Monaten abgeschlossen sein. Es bleiben also maximal 5 Monate zur Vorbereitung.

Eine der umzusetzenden Maßnahmen ist die Rückverfolgung von Teilen mittels Radio Frequency Identification (RFID). Dies soll zur Transparenz des Produktionsprozesses beitragen. Zur leichteren Umstellung auf RFID muss der aktuelle Prozess schnell überprüft werden.

Die CIO wird gebeten, das Management des Change Prozesses zu übernehmen. Sie glaubt, dass die Umstellung gelingen kann, wenn die S-Gruppe mit einem DevOps-Ansatz ein Minimalrelease erstellt. Im Idealfall wird für die Produktion mit RFID zuerst ein Konzept entwickelt und das Produktionsleitsystem mit Hilfe der RFID-Daten dann erst im letzten Schritt umgesetzt. Aufgrund der Kürze der Zeit können diese Schritte jedoch nicht der Reihe nach durchgeführt werden und sollen daher gleichzeitig erfolgen.

Die CIO stellt Em für das Projekt ab. Em ist Scrum Master. Das Entwicklungsteam bereitet sich auf den Bau einer Deployment Pipeline vor.

Em stellt fest, dass das Entwicklungsteam mit Begeisterung bei der Sache ist und fleißig arbeitet. Allerdings mangelt es dem Team etwas an Disziplin. Außerdem muss die Releasefrequenz erhöht werden.

Auf was sollte sich Em **zuerst** konzentrieren?

- A) Em sollte sich auf die Kommunikation konzentrieren, da diese bei DevOps am wichtigsten ist. Em sollte zuerst das Eis im Team brechen und dann einige Regeln für die Kommunikation festlegen.
- B) Em sollte zuerst eine Wertstromanalyse durchführen und mit dem Team eine schlanke Produktion nach dem Prinzip One-piece-flow (in einem Rutsch) erstellen, denn straffe Prozesse sind äußerst wichtig.
- C) Em sollte zuerst die Infrastruktur und das Arbeitsumfeld mit den Teammitgliedern besprechen, da DevOps am wirkungsvollsten ist, wenn alle Tools (Hilfsmittel) und Praktiken funktionieren.
- D) Em sollte zuerst, alle Stakeholder versammeln, diese über DevOps aufklären und sie bitten, den für DevOps erforderlichen Kulturwandel zu unterstützen.

33 / 50

Ihr DevOps-Team arbeitet gut und in einem nachhaltigen Tempo zusammen. Sie planen genügend Pufferzeiten in den Prozess ein, so dass das Team über genügend Zeit und Konzentration verfügt, um die Builds sorgfältig zu prüfen und zu testen. Derzeit erfolgen Deployment und Testen in Ihrem Team manuell. Das Tempo des Teams ist hoch genug, um dem Unternehmen regelmäßig großen Mehrwert zu liefern.

Ihr CEO hat Sie um Ihren Rat bezüglich der Automatisierung in diesem Team gebeten.

Welchen Rat sollten Sie ihm geben?

- A) Automatisieren Sie so viel Sie können, damit das Team mehr Funktionen ergänzen und noch schneller einen Mehrwert für das Unternehmen demonstrieren kann.
- B) Automatisieren Sie die Abnahmetests, aber nicht die Deployments, denn der manuelle Prozess ist sicherer.
- C) Automatisieren Sie die Deployments, um die Zykluszeit zu verkürzen, aber nicht das Testen, damit man von den Bugs lernen kann.
- D) Führen Sie keine Automatisierung bei diesem Team ein, weil die derzeit von dem Team ausgeführten Methoden großartige Ergebnisse erzielen.

34 / 50

Eine CIO teilt Michael, ihren zuverlässigsten Mitarbeiter und Scrum Master, in ein Projekt ein. Das Entwicklungsteam bereitet sich auf den Bau einer Deployment Pipeline vor.

Michael vertraut den guten Absichten und der Spontanität des Entwicklungsteams, wünscht sich aber etwas mehr Disziplin. Außerdem sollte die Releasefrequenz erhöht werden. Michael möchte, dass das Entwicklungsteam häufigere Releases umsetzt.

Ein Teammitglied sagt dazu: „Am wichtigsten ist die Automatisierung der Deployment Pipeline. Daher sollten wir die Deployment Pipeline zuerst automatisieren“.

Ist diese Aussage richtig?

- A) Ja, das ist richtig. Die Automatisierung der Deployment Pipeline ist für die Verbesserung der Effizienz am wichtigsten.
- B) Ja, das ist richtig. Wenn man sich auf die Erstellung einer automatisierten Deployment Pipeline konzentriert, vermeidet man potenzielle Probleme, die zu einem späteren Zeitpunkt auftreten können.
- C) Nein, das ist nicht richtig. Erste Priorität sollte sein, das Prinzip des One-piece-flow (in einem Rutsch) und einen soliden Deploymentprozess umzusetzen. Die Automatisierung des Prozesses kann später erfolgen.
- D) Nein, das ist nicht richtig. Nicht die Deployment Pipeline, sondern der Testprozess sollten zuerst automatisiert werden.

35 / 50

Ihr Unternehmen stellt seine Arbeitsweise auf DevOps um. Ihr Team ist in diese Änderung eingebunden. Die Teammitglieder diskutieren die Best Practices für die Commit-Phase des Codes.

Ihr Kollege Sun sagt: „Wenn ein Build fehlschlägt und keiner die Verantwortung übernimmt, sollten wir den Verantwortlichen ermitteln und ansprechen, damit er den Build reparieren kann.“

Ist dies eine gute Idee?

- A) Ja. Denn fehlgeschlagene Builds können nur von den Verursachern behoben werden. Daher sollte man die Verantwortlichen identifizieren, auch wenn das für sie unangenehm ist.
- B) Ja. Denn, wer den Fehlschlag eines Build verursacht hat, sollte auch stets die Verantwortung dafür übernehmen. Tun Sie dies nicht, dann werden Ihre Kollegen diese Regel möglicherweise durchsetzen.
- C) Nein. DevOps baut auf eine Kultur frei von Schuldzuweisungen. Will ein Kollege die Verantwortung nicht übernehmen, dann sollte man ihn nicht dazu zwingen.
- D) Nein. Zuerst sollte man den Build reparieren. Danach sollte man sich die Zeit nehmen, den Verantwortlichen zu identifizieren und zu bestrafen.

36 / 50

Das Entwicklungsteam von XAppGo ist mit seinen aktuellen Testpraktiken zahlreichen Herausforderungen begegnet. Aktuell setzt das Team für die Abnahme einen manuellen Testprozess ein. Die Entwickler glauben, dass die von ihnen erstellte Unit-Testsuite ausreichend Schutz vor Regressionen bietet.

Das Entwicklungsteam gibt bei jedem Release 1 Million für die manuellen Abnahmetests aus. Die obere Leitung hat vorgegeben, dass das Entwicklungsteam automatisierte Abnahmetests einführen muss, um die Gesamtkosten für das Testen zu reduzieren und die Zahl der Code-Fehler und Regressionen in der Produktivumgebung auf ein Mindestmaß zu beschränken.

Welche Grundsätze müssen Sie im Hinblick auf die Automatisierung bei der Festlegung der Abnahmekriterien für Ihre Anwendung befolgen?

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

37 / 50

Was ist der **effektivste** Mechanismus für die automatisierte Datenmigration?

- A) Die Erstellung eines Versionierungsschemas für die bestehenden Daten und deren Pflege mit Hilfe der Versionskontrolle
- B) Die Erstellung und Verwaltung kleinerer Datensätze, damit die Migration einfacher wird
- C) Die Sicherstellung, dass alle Skripte vor der Datenmigration ordnungsgemäß geprüft wurden
- D) Die Sicherstellung, dass man über ein Rollback-Verfahren verfügt, falls die Migration fehlschlägt

38 / 50

Das Unternehmen X-AppGo hat Schwierigkeiten mit seinen Rollback-Prozessen. Bei der Ausführung von Rollback-Skripten kommt es häufig zu kritischen Datenverlusten in den Datenbanken der Produktivumgebung.

Wann können Rollback-Skripte **nicht** ohne kritische Datenverluste ausgeführt werden?

- A) Das Rollback-Skript löscht Daten, die nur von der neuen Version genutzt werden.
- B) Im Rahmen des Rollback-Skripts muss man eine Spalte von einer in die andere Tabelle verschieben.
- C) Das Rollback-Skript fügt Daten aus temporären Tabellen ein.

39 / 50

ACMECONST hat bei der Bereitstellung von Software-Upgrades für seine Anwendung und bei den Hardware-Aktualisierungen seiner Router und Switches viele Ausfälle der Anwendung und der Hardware erlebt.

Nachdem diese Ausfälle während des für die Instandhaltung reservierten Zeitfensters aufgetreten sind, war es für das Unternehmen sehr schwierig, den ursprünglichen Zustand wieder herzustellen. Die Wiederherstellung hat das normalerweise für die Instandhaltung vorgesehen Zeitfenster überschritten und die Ausfallzeit ihrer kritischen Anwendungen verlängert.

Eine automatisierte Bereitstellung und autonome Infrastruktur kann in einer solchen Situation hilfreich sein. Dabei gibt es jedoch Einiges zu berücksichtigen.

Was muss sorgfältig gemanagt werden, um das Risiko einer Störung beim Deployment in die Produktivumgebung zu reduzieren?

- A) Die detaillierten Überwachungsprotokolle zur Fehlerbehebung bei Ausfällen im Rahmen von Anwendungsupdates
- B) Externe Integrationspunkte, wie z. B. externe Systeme und Services
- C) Server-Konfigurationen und die zugrundeliegenden Informationen zum User Account
- D) Automatisierungstools und autonome Architektur

40 / 50

Das Unternehmen X-AppGo hat Schwierigkeiten mit seiner zentralen Applikation. Die Applikation arbeitet nicht ordnungsgemäß mit anderen externen Applikationen zusammen. Diese externen Applikationen müssen effektiv bestimmte Datenvariablen erhalten, damit bestimmte Calls ausgeführt werden können. Die Kernapplikation wird von einem Team entwickelt, das das Unternehmen aus nachvollziehbaren Unternehmensgründen beibehalten möchte.

Einer der Entwickler schlägt vor, eine Komponente aus der X-AppGo Codebasis auszugliedern, um das Schnittstellenproblem zu bearbeiten.

Was sind in diesem Fall gute Argumente für diese Ausgliederung und die separate Betrachtung einer Komponente?

- A) Die Anpassung einiger Plugins in der Codebase in eine monolithische Codebase
- B) Die Begrenzung von Auswirkungen von Änderungen und das Erleichtern von Änderung der Codebase
- C) Die X-AppGo-Codebase muss aufgesplittet und von verschiedenen Teams verwaltet werden
- D) Es gibt keine guten Argumente für diese Maßnahme und die Kompilierung wird mehr Zeit in Anspruch nehmen

41 / 50

Selbst bei den kleinsten Anwendungen besteht eine Abhängigkeit zu anderen Komponenten oder Libraries (Bibliotheken). Das Verstehen und die Verwaltung der Abhängigkeiten ist daher eine der wichtigsten Aktivitäten innerhalb der Continuous Deployment (kontinuierlichen Bereitstellung), um den Flow in der Deployment Pipeline aufrechtzuerhalten.

Sie haben eine Anwendung gebaut, die zwei Libraries nutzt. Jede dieser Libraries basiert auf einer dritten Basis-Library. Allerdings beziehen sie sich auf verschiedene Versionen. Dies führt zu einer besonderen Abhängigkeit.

Welche Lösung eignet sich am **besten**, um diese Abhängigkeit zu beheben oder zu vermeiden?

- A) Fügen Sie alle Libraries zu einer einzigen Library zusammen, damit Sie sich direkt auf die Library beziehen und das Problem so vermeiden können
- B) Verwalten Sie die Libraries mit Hilfe der Versionskontrolle. So sehen Sie es unmittelbar, wenn Sie diese Art der Abhängigkeit schaffen
- C) Behalten Sie alle Ihre Abhängigkeit mit Hilfe von Post-Its auf einer großen Tafel im Blick, damit Sie den Flow überwachen können
- D) Prüfen Sie nur kleine Teile Ihrer Werkzeugkette (Toolchain), damit Sie eventuelle Probleme beim Check-in leicht beheben können

42 / 50

In einer Umgebung der Continuous Deployment (kontinuierlichen Bereitstellung) ist es wichtig, dass alles versionskontrolliert ist, damit Sie Fehler schnell finden oder bei Bedarf ein Rollback durchführen können.

Bei einem binären Output, wird jedoch von einer Versionskontrolle **abgeraten**.

Warum wird hier eine Ausnahme gemacht?

- A) Ein binärer Output befindet sich tendenziell in großen Dateien, die bei jedem Build verändert und automatisch aktualisiert werden.
- B) An binären Dateien arbeiten verschiedene Mitglieder des Teams, daher ist hier eine Versionskontrolle nicht praktisch.
- C) Binärer Output dient als Input für die Compiler, die bereits der Versionskontrolle unterliegen.
- D) Es besteht keine Notwendigkeit dazu, da im Rahmen des Build-Prozesses regelmäßig eine erneute Kompilierung erfolgt.

43 / 50

Sie suchen nach einem ganzheitlichen Ansatz für das Management Ihrer IT-Infrastruktur

Welche zwei Prinzipien sind Ihnen dabei am **besten** behilflich?

- A)**
 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Änderungskontrolle unterliegt.
 2. Dank Monitoring (Überwachung) und Event Management sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
- B)**
 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Änderungskontrolle unterliegt.
 2. Dank Instrumentierung und Incident Management sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
- C)**
 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Versionskontrolle unterliegt.
 2. Dank aktuellem Incident Management und Event Management sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
- D)**
 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Versionskontrolle unterliegt.
 2. Dank Instrumentierung und Monitoring (Überwachung) sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.

44 / 50

Teams, die gut zusammenarbeiten haben synchronisierte Arbeitstickets. Ein technischer Direktor hat sich persönlich davon überzeugt, wie das Betriebsteam funktioniert. Das Betriebsteam legt nach jedem Release die Betriebsinfrastruktur neu fest.

Was ist der **beste** Ratschlag, um diese Praxis zu verbessern?

- A)** Das Betriebsteam sollte nichts tun. Es gibt nichts zu verbessern, weil eine solche Neufestlegung stets erfolgen sollte.
- B)** Das Betriebsteam sollte versuchen, die Betriebsinfrastruktur und die Authentifizierung auf die Betriebsumgebung zu modellieren.
- C)** Das Betriebsteam sollte die Betriebsinfrastruktur prüfen, damit dies ein automatisierter Prozess wird.
- D)** Das Betriebsteam sollte sein Wissen über den Deployment-Prozess mit dem Entwicklungsteam teilen.

45 / 50

Wann sollte der Betrieb (Operations) die Entwicklung (Development) über eine Änderungen im Betrieb informieren?

- A)** Die Entwicklung (Development) muss gar nicht informiert werden. Änderungen im Betrieb betreffen nur das Betriebsteam.
- B)** Umgehend. Die Entwicklung muss baldmöglichst informiert werden.
- C)** Im Scrum-of-Scrum-Meeting am nächsten Morgen.
- D)** Nachdem das Betriebsteam den Abnahmetest durchgeführt hat.

46 / 50

Sie wollen den Reifegrad Ihrer DevOps-Organisation steigern. Dies ist auf viele verschiedene Arten möglich.

Welche Vorgehensweise eignet sich **nicht** dazu, den Reifegrad Ihrer DevOps-Organisation zu steigern?

- A) Sie legen eindeutige Ziele als Meilensteine fest, damit die Mitglieder Ihres Teams den Wert ihrer täglichen Aktivitäten besser einschätzen können.
- B) Sie definieren Prozesse eindeutig und ermöglichen es den Mitgliedern Ihres Teams dadurch, Prozesse täglich zu verbessern.
- C) Sie zeichnen alle Besprechungen auf, damit die Mitglieder Ihres Teams leicht auf sämtliche Kommunikation zugreifen können.
- D) Sie überwachen die täglichen Aktivitäten und zeichnen diese auf, damit selbst die kleinsten, täglich erzielten Fortschritte identifiziert und entsprechend gefeiert werden können.

47 / 50

Sie arbeiten für einen IT-Dienstleister. Im Rahmen Ihres Business Continuity-Plans möchten Sie garantieren, dass Sie die vereinbarten Service Level stets erfüllen können.

Sie wollen die IT Service Continuity sicherstellen.

Wie kann DevOps Sie beim IT Service Continuity Management unterstützen?

- A) Affinität und Kollaboration sind Werte der DevOps-Kultur die dafür sorgen, dass der Service von den Mitgliedern des DevOps-Teams hoch geschätzt wird.
- B) DevOps bringt absichtlich Chaos in das System und sorgt so für Notfallroutinen und Obeya-Praktiken im Team.
- C) Maßnahmen zur Risikoreduzierung und Optionen zur Wiederherstellung im Notfall werden wahrscheinlich direkt in den Code eingebaut, denn Betrieb (Operations) und Entwicklung (Development) arbeiten eng zusammen.
- D) Das Service Level Management spielt bei DevOps eine wichtigere Rolle, denn es muss vom Process Master überwacht werden.

48 / 50

Das Unternehmen ACMECONST hat seine globale Präsenz aggressiv ausgebaut und dazu die Zahl der Neueinstellungen und die Zahl der technischen Teams weltweit erhöht. Mit einem Zuwachs von 30% hat das Unternehmen auch seinen Kundenstamm drastisch erhöht.

Seitdem sich das Technik-Team nicht mehr in einem Raum befindet dauern einfache Entscheidungen viel länger. Dies verursacht in der gesamten Organisation wachsende Frustration. Freigaben müssen jetzt mehrere Managementebenen durchlaufen und der Prozess ist umfangreicher und komplexer. Viele Ingenieure sind daher von dem Entscheidungsprozess insgesamt desillusioniert.

Auch bezüglich der Verantwortung und Zuständigkeiten herrscht zunehmend Verwirrung. Die Ingenieure sind der Meinung, dass die zusätzlichen Prozesse und das Mehr an Bürokratie ihre Kreativität einschränken. Dies wirkt sich allmählich auf die Moral aus.

Wie lassen sich die Probleme in diesem Szenario am **besten** lösen?

- A) Behalten Sie die aktuellen Prozesse bei, aber sorgen Sie bei jedem Prozess für eindeutige Rollen, Verantwortlichkeiten und Zuständigkeiten. Führen Sie eine wirksame Methode ein, um Produktivität und Risiko gegeneinander abzuwägen, machen Sie schrittweise Änderungen und schaffen Sie sichere Räume für Experimente.
- B) Überprüfen Sie die aktuellen Prozesse noch einmal, um zu sehen, wo sie diese straffen können und sorgen Sie bei jedem Prozess für klare Rollen, Verantwortlichkeiten und Zuständigkeiten. Führen Sie eine wirksame Methode ein, um Produktivität und Risiko gegeneinander abzuwägen, machen Sie schrittweise Änderungen und schaffen Sie sichere Räume für Experimente.
- C) Überprüfen Sie die aktuellen Prozesse noch einmal, um zu sehen, wo sie diese straffen können und sorgen Sie bei jedem einzelnen Prozess für klare Rollen, Verantwortlichkeiten und Zuständigkeiten. Führen Sie eine wirksame Methode ein, um Produktivität und Risiko gegeneinander abzuwägen, machen Sie schrittweise Änderungen und reduzieren Sie Experimente auf das absolut erforderliche Mindestmaß, um unnötige Ausfälle der Anwendung zu vermeiden.

49 / 50

Im Unternehmen X-AppGo gibt es einen Konflikt zwischen dem Betriebsteam in Kolumbien und dem Entwicklungsteam in Irland. Grund für den Konflikt sind unterschiedliche Prioritäten und Ziele. Aufgrund des Konflikts nimmt die Lösung von Problemen, die sich auf den Geschäftsbetrieb auswirken, mehr Zeit in Anspruch.

Welche zentralen Praktiken sollte X-AppGo in Betracht ziehen, um das Konfliktpotenzial zu reduzieren und die Kollaboration zwischen dem Entwicklungs- und dem Betriebsteam zu verbessern?

- A)
 1. Um Konflikte zu vermeiden, sollte X-AppGo es den Teams ermöglichen, unabhängig voneinander zu arbeiten, falls die Teams dies vorziehen.
 2. Die Entwicklungs- und Betriebsteams sollten die volle Unterstützung des Vorstands haben.
- B)
 1. Ein Mitglied des Vorstands sollte die Rolle des Förderers übernehmen und dem DevOps-Team erläutern, wie wichtig eine gute Zusammenarbeit ist.
 2. Das Entwicklungs- und das Betriebsteam sollten in DevOps-Praktiken geschult werden, damit jedes Team die Arbeit des jeweils anderen Teams übernehmen kann.
- C)
 1. Das Entwicklungs- und das Betriebsteam sollten andere Unternehmen besuchen, in denen DevOps erfolgreich angewendet wird.
 2. Das Entwicklungs- und das Betriebsteam sollten mehr Mittel erhalten, um die Teams besser bei der Erfüllung der gestiegenen Anforderungen zu unterstützen.
- D)
 1. Die Teams sollten sich gegenseitig an ihren jeweiligen Standorten besuchen, um den Teamgeist zu stärken und ein Verhältnis des Vertrauens und gegenseitigen Verständnisses aufzubauen.
 2. Wissen und Erfahrungen sollte zwischen dem Entwicklungs- und dem Betriebsteam ausgetauscht und verbreitet werden, um für eine effizientere Zusammenarbeit zu sorgen.

50 / 50

Ein Entwicklungsteam interessiert sich für DevOps. Das Team interessiert sich dabei hauptsächlich für die Continuous Integration (kontinuierlichen Integration). Aktuell entwickelt und pflegt das Team mit Scrum-Praktiken 3 größere und 4 kleinere Lösungen. Jeder Sprint dauert 4 Wochen und führt im Schnitt alle 10 bis 15 Tage zum Commit von 1 Release an die Testumgebung sowie zum Commit von 1 Release pro Monat an die Produktivumgebung. Das Team möchte für das Management einen qualitativen Business Case erstellen, um die Unterstützung des Managements für die Investitionen und die Bemühungen zur Einführung der Continuous Integration (kontinuierliche Integration) zu gewinnen.

Welche greifbaren Vorteile der Continuous Integration (kontinuierlichen Integration) eignen sich **am besten** zur Unterstützung des Business Case?

- A) Ein tägliches Deployment an die Testumgebung könnte den Nutzen für das Unternehmen steigern und die Entwicklungskosten drastisch reduzieren.
- B) Sie unterstützt den Team-Geist. Da das Team bereits mit Scrum arbeitet, bringt Continuous Integration (Kontinuierliche Integration) dem Unternehmen **keinen** messbaren Nutzen.
- C) Es erhöht die Stabilität und Qualität von Releases durch automatisiertes Testen und erhöht die allgemeine Release-Geschwindigkeit.
- D) Ein tägliches Release an die Produktivumgebung könnte den Nutzen für das Unternehmen steigern und die Entwicklungskosten drastisch senken.

Antwortschlüssel

1 / 50

Was ist ein gutes Argument für die Einführung von DevOps in einer Organisation?

- A) DevOps bietet bei neuen Services häufigere Feedback-Zyklen, weil die Entwicklung schneller ist.
 - B) DevOps sorgt für bessere Business Continuity und Agilität, weil Prozesse optimiert und nur wertschöpfende Aktivitäten durchgeführt werden.
 - C) DevOps sorgt für häufigere Software-Releases und stellt so neue Services Just-in-Time (JiT) bereit.
 - D) DevOps vereint Entwicklung und Betrieb und reduziert damit Zeitverschwendung in der Kollaboration
-
- A) Falsch. Es ist zwar richtig, dass die Entwicklung schneller ist und es dadurch häufigere Feedback-Zyklen gibt, aber der Grund hierfür ist in Scrum oder einer anderen Agilen Methode zu suchen und nicht per se in DevOps.
 - B) Richtig. Mehrwert zu schaffen und Prozesse zu optimieren sind der Schlüssel für eine bessere Business Continuity und mehr Agilität des Unternehmens. Denken Sie daran, was es bedeutet, dass IT-Services stets das Unternehmen unterstützen sollen und was der Wert und der Zweck von DevOps sind. (*Literatur: C, Kapitel 2*)
 - C) Falsch. Just-in-Time (JiT)-Bereitstellung ist zwar großartig, aber an sich noch kein Argument für die Einführung von DevOps. Die Implementierung von Lean-Prozessen wäre zum Erreichen dieses Ziels besser.
 - D) Falsch. Durch die Zusammenlegung von zwei Teams sorgt man noch nicht für die Vermeidung von Muda (Verschwendung). Nur wenn die Teams auch ihre Praktiken ändern, kann Verschwendung vermieden werden.

2 / 50

Der CTO denkt, dass es am wirksamsten wäre, bestimmte Lean-Konzepte bei der Umsetzung von DevOps anzuwenden

Welche Prinzipien oder Praktiken von Lean sind bei der Einführung von DevOps am **wirksamsten**?

- A) **Kaizen und 5S.** Da Agile und DevOps auf zentralen Konzepten von Lean basieren und Kaizen und 5S die Grundlage von Lean bilden, sind diese beiden Methoden bei der Einführung von DevOps am wirksamsten.
 - B) **Kaizen in Advance.** DevOps funktioniert nur mit Feedback vom Betrieb an die Entwicklung. Kaizen in Advance sorgt für Feedback zu den vorgelagerten Einheiten und unterstützt damit die Anwendung dieses Prinzips in DevOps.
 - C) **Obeya System.** DevOps integriert verschiedene Management-Stile und Prozesse. Das Obeya System hilft bei der Visualisierung des Gesamtprozesses und ermöglicht die erfolgreiche Einführung von DevOps.
 - D) **One-piece-flow (in einem Rutsch) und JKK.** DevOps profitiert vom Aufbau von Upstream-Prozessen und einem Einzelwertstromfluss. Das Prinzip des One-piece-flow (in einem Rutsch) ermöglicht dies und JKK hilft dabei den Flow zu straffen und umzusetzen.
-
- A) Falsch. Lean, Agile und DevOps hängen zwar zusammen, aber Kaizen und 5S sind nicht die besten Methoden zur Unterstützung eines erfolgreichen DevOps-Launch. Kaizen kann nach Einführung von DevOps zur kontinuierlichen Verbesserung eingesetzt werden und 5S kann zur Aufrechterhaltung guter Praktiken beitragen. Beides jedoch gilt erst nach der erfolgreichen Einführung von DevOps.
 - B) Falsch. Feedback ist zwar immer willkommen, garantiert aber nicht notwendigerweise die wirksamste Anwendung von Lean bei der Umsetzung von DevOps.
 - C) Falsch. Visualisierung mag hilfreich sein, ist aber nicht die wirkungsvollste Lean Praktik bei der Umsetzung von DevOps.
 - D) Richtig. Der Bau einer funktionierenden Deployment Pipeline nach dem Prinzip One-piece-flow (in einem Rutsch) trägt zur erfolgreichen Umsetzung von DevOps bei. Das Wichtigste an DevOps ist der Aufbau nachgelagerter Prozessen von der Entwicklung zum Betrieb insbesondere im Hinblick auf eine Deployment Pipeline nach dem One-piece-flow (in einem Rutsch)-Prinzip. JKK ist hierfür die wirksamste Methode. (*Literatur: A, Kapitel 7*)

3 / 50

Was muss gegenüber einem konventionellen Projekt geändert werden, damit ein Projekt in DevOps erfolgreich ist?

- A) Es sollte eine IT-Service Lieferkette mit Pull-System und One-piece-flow (in einem Rutsch) erstellt werden.
 - B) Die Entwickler sollten sich dem Betriebsteam anschließen, um die schnelle Instandhaltung der Services sicherzustellen.
 - C) Der Betrieb muss für das Entwicklungsteam arbeiten. Deshalb heißt es DevOps.
 - D) Die Mitglieder des Betriebsteams sollten sich dem Entwicklungsteam anschließen.
-
- A) Richtig. Ein Projekt gilt dann als erfolgreich, wenn der Prozess ein Jidoka-basiertes Pull-System nutzt, um wertvolle IT Services (oder Produkte) zu erstellen. (*Literatur: C, Kapitel 4 und B, Kapitel 1*)
 - B) Falsch. DevOps bedeutet nicht, dass sich Entwickler dem Betriebsteam anschließen.
 - C) Falsch. Kollaboration alleine macht aus einem Projekt noch keinen DevOps-Erfolg.
 - D) Falsch. DevOps bedeutet nicht, dass sich das Betriebsteam dem Entwicklungsteam anschließt.

4 / 50

Alle Teams die DevOps einführen, könnten als Teamverbund bezeichnet werden.

Welches Prinzip sorgt am **besten** dafür, dass ein Teamverbund funktioniert?

- A) Dynamische Zusammenarbeit
 - B) Laufende Kommunikation
 - C) Reduzierung der Zyklen
 - D) Gemeinsame Verantwortlichkeit
-
- A) Falsch. Dynamische Zusammenarbeit ist kein Prinzip des Teamverbunds.
 - B) Richtig. Zu den Prinzipien eines DevOps-Teamverbunds zählen laufende Kommunikation, gemeinsame, klar definierte Ziele und die dynamische Anpassung und Wiederherstellung von Verständnis. (*Literatur: A, Kapitel 2*)
 - C) Falsch. Reduzierung der Zyklen ist kein Prinzip des Teamverbunds.
 - D) Falsch. Gemeinsame Verantwortlichkeit ist kein Prinzip des Teamverbunds.

5 / 50

Für die Implementierung von DevOps stehen verschiedene Quellen an Wissen, Standards und Praktiken zur Verfügung.

Welche der folgenden Quellen gilt für die erfolgreiche DevOps-Implementierung als **entscheidend**?

- A) CMMI Level 3
 - B) Disziplinierte Agilität
 - C) ISO 20001
 - D) PMI PMBok
-
- A) Falsch. Eine CMMI-Zertifizierung könnte hilfreich sein, ist aber für die DevOps-Implementierung nicht entscheidend. Ein spezifischer CMMI-Level ist das noch viel weniger.
 - B) Richtig. Disziplinierte Agilität ist eine entscheidendes Kriterium für eine erfolgreiche DevOps-Implementierung (*Literatur: C, Kapitel 4i*)
 - C) Falsch. Eine Zertifizierung nach ISO 20001 könnte hilfreich sein, ist aber für die DevOps-Implementierung nicht entscheidend.
 - D) Falsch. PMI PMBok ist ein Nachschlagewerk für das Projektmanagement und für die DevOps-Implementierung nicht entscheidend.

6 / 50

Kollaboration ist einer der vier Säulen des wirksamen DevOps.

Warum ist dies so wichtig?

- A) Nur mit Kollaboration erreicht man in DevOps One-piece-flow (In einem Rutsch).
 - B) Der durch die DevOps-Methode generierte Mehrwert für das Unternehmen wird durch die Zusammenarbeit verschiedener Teams erreicht.
 - C) DevOps Organisationen sind klein, die Teams müssen daher gut zusammenarbeiten.
 - D) Kollaboration fördert Veränderungen und ist einfach. Sie reduziert außerdem die Entwicklungskosten.
-
- A) Falsch. One-piece-flow (in einem Rutsch) lässt sich auch ohne Kollaboration erreichen. Dies ist ein anderer Mechanismus.
 - B) Richtig. Die Zusammenarbeit aller beteiligten Teams (u.a. auch der Entwicklung und des Betriebs) ist von größter Wichtigkeit, um durch mehr Kommunikation, Automatisierung und qualitativ hochwertigere Software Mehrwert für das Unternehmen zu generieren. (*Literatur: A, Kapitel 6 und 7*)
 - C) Falsch. DevOps Organisationen können sehr groß sein. Die Teams müssen zusammenarbeiten, um genügend Unternehmensmehrwert zu generieren.
 - D) Falsch. Tools fördern Veränderung und sorgen für schnelle Erfolge. Zusammenarbeit ist schwierig. Kollaboration kann zwar die Entwicklungskosten senken, dies ist aber nicht das wichtigste Ziel. Das wichtigste Ziel ist, die Qualität zu verbessern.

7 / 50

Sie sind der Eigentümer eines kleinen DevOps-Unternehmens mit 5 Mitarbeitern, das mobile Apps für behinderte Kinder baut. Eine der Apps, auf die Ihr Team äußerst stolz ist, ist eine App, mit der autistische Kinder ihren Tag selbst planen können.

Weil die App so erfolgreich war, wird Ihr Team gebeten, sie auszubauen und auch auf andere Menschen anzupassen, die ebenfalls von der selbständigen Planung ihres Tags profitieren würden. Diese Bitte wird den Code unweigerlich komplexer machen und geht mit einigen technischen Herausforderungen einher, die das Team meistern muss.

Sie gehen davon aus, dass Sie für den Auftrag sehr gut bezahlt werden und nehmen diesen an. Nach ein paar Wochen jedoch streitet sich Ihr Team. Sie werden ebenfalls wütend und achten stärker darauf, was die einzelnen Mitglieder Ihres Teams tun. Sie arbeiten regelmäßig gemeinsam mit Ihrem Team und können so alle eventuellen Fehler im Code, die Sie entdecken, schnell korrigieren.

Obwohl genügend Arbeit erledigt wird, ist Ihr Team auch weiterhin wütend und Sie erkennen, dass das Team teilweise nicht die richtige Denkweise (Mindset) für DevOps hat.

Welche Strategie eignet sich am **besten** zur Lösung dieses Problems?

- A) Erweitern Sie Ihr Team um neue Mitglieder und achten Sie dabei auf mehr Diversität, indem Sie nicht-technisches Personal einstellen
 - B) Bitten Sie externe DevOps-Experten, ihre Teammitglieder als Sponsoren und Mentoren zu unterstützen
 - C) Suchen Sie ein gemeinsames Ziel und arbeiten Sie gemeinsam an der Verbesserung der Kollaboration
 - D) Nehmen Sie sich ein paar Monate Zeit und arbeiten Sie zuerst am Team Building, um mehr über die anderen Teammitglieder zu erfahren
-
- A) Falsch. Eine Teamvergrößerung ist nur dann ratsam, wenn das Team nicht über die Kompetenz verfügt, die zur Ausführung der Aufgabe notwendig ist. Allerdings ist die Einstellung von Mitarbeitern ohne technische Aufgaben zur Aufrechterhaltung der Teameffizienz weder ratsam noch wünschenswert. In diesem Fall wird genügend Arbeit erledigt, so dass die Einstellung von nicht-technischem Personal die Probleme, die Sie derzeit erleben, noch weiter verstärken würde, weil es an Vertrauen und Kollaboration im Team fehlt.
 - B) Falsch. Dies ist zwar eine großartige Idee, wird aber nicht dazu beitragen, die fehlende Kollaboration und Affinität zu DevOps in Ihrem Unternehmen zu lösen. Kollaboration wird durch gemeinsame Ziele und Erfolge verbessert.
 - C) Richtig. Dies ist eine gute Möglichkeit, um die aktuellen Probleme zu lösen und sollte ohne großen Zeitaufwand für die Problemlösung und möglicherweise ohne Verlust des Auftrags zur Lösung des Problems führen. Gemeinsame Ziele tragen zur Stärkung von Vertrauen, Affinität und Kollaboration bei. (*Literatur: A, Kapitel 7*)
 - D) Falsch. Dies mag funktionieren und zu einem Team führen, das sich gegenseitig unterstützt und zusammenarbeitet, was Ihr Ziel sein sollte. Dies ist aber nicht die beste Lösung des Problems, weil Sie zu viel Zeit dafür aufwenden und dadurch den Verlust des Auftrags riskieren.

8 / 50

Sie halten Ihr Entwicklungsteam für ein echtes Team.

Was ist ein sicheres Zeichen dafür, dass Ihr Entwicklungsteam nicht nur eine Gruppe, sondern ein echtes Team ist?

- A) Das Team hält sich an die Regeln, die in den Team-Meetings vereinbart wurden.
 - B) Das Team hält wirksame Besprechungen ab, die von den Mitgliedern selbst geleitet werden.
 - C) Das Team arbeitet stetig auf das Erreichen des gemeinsamen Ziels hin.
 - D) Das Team löst Probleme durch Befragung der zuständigen Teammitglieder.
-
- A) Falsch. Auch Gruppen können sich sehr gut an Regeln halten. Dies bedeutet nicht unbedingt, dass es sich um ein Team handelt.
 - B) Falsch. Auch Gruppen können sehr wirksame Besprechungen abhalten. Dies ist nicht unbedingt ein Zeichen für ein Team.
 - C) Richtig. Ein echtes Team stellt ein stetiges Arbeitstempo sicher und arbeitet kontinuierlich am Erreichen der gemeinsamen Ziele. (*Literatur: A, Kapitel 9*)
 - D) Falsch. Ein Team löst die Probleme gemeinsam und fängt nicht damit an, einzelne Teammitglieder in Frage zu stellen. DevOps baut auf eine Kultur frei von Schuldzuweisungen.

9 / 50

AppAtoZ verzeichnet ein phänomenales Wachstum bei der Entwicklung und Bereitstellung von mobilen Anwendungen für iPhone und Android-Plattformen.

Die Entwicklungsteams dieses Startups stehen unter unglaublichem Druck, weil sie für ihre aktuellen mobilen Applikationen schnell Verbesserungen in kürzester Zeit bereitstellen müssen. Die Teams haben in den letzten sechs Monaten im Schnitt 60 Stunden pro Woche gearbeitet. Die Geschäftsführung ist sehr zurückhaltend was Neueinstellungen anbetrifft und eher darauf bedacht, die Umsatzerlöse zu steigern und gleichzeitig die operativen Kosten und die Entwicklungskosten zu senken.

In den letzten Monaten war eine Erhöhung der Fehlzeiten festzustellen. Mitarbeiter meldeten sich krank und einige Mitarbeiter von AppAtoZ kündigten sogar, was die Arbeitsbelastung der aktuellen Mitarbeiter noch weiter erhöht. Aufgrund der für die Einstellung und Einarbeitung eines neuen Mitarbeiters benötigten Zeit wird sich die Arbeitsbelastungen des Entwicklungsteams nicht so schnell verringern.

Welche **langfristigen** Strategien sind hinsichtlich Mitarbeiter-Burnout und Stress zu berücksichtigen?

- A)
 - Verstärken Sie das Team mit einer Mischung aus angestellten und freien Mitarbeitern und Freiberuflern, um die Arbeitsbelastung zu bewältigen.
 - Sorgen Sie dafür, dass die Geschäftsführung und das Entwicklungsteam zu einem Burnout beitragenden Faktoren der Arbeitsumgebung identifizieren.
 - Entwickeln Sie einen Plan, der diese Faktoren anspricht.
- B)
 - Entlassen Sie das Führungsteam, da dieses unfähig ist.
 - Stellen Sie ein kompetenteres Führungsteam ein, das weiß, wie man für eine gute Work-Life-Balance und eine nachhaltige und realistische Kultur sorgt.
 - Geben Sie dem Entwicklungsteam die Möglichkeit, sich die Zeit zu nehmen, um ins Gleichgewicht zu kommen.
- C)
 - Veranlassen Sie die Mitglieder des Entwicklungsteams, sich frei zu nehmen.
 - Arbeiten Sie mit freien Mitarbeitern zusammen, um die höhere Arbeitsbelastung in Spitzenzeiten abzufangen.
 - Fordern Sie die Entwickler auf, sich bei Bedarf professionelle psychologische Hilfe zu holen.
- D)
 - Veranlassen Sie die Mitglieder des Entwicklungsteams, sich frei zu nehmen.
 - Sorgen Sie dafür, dass Führung und Entwicklung die zu einem Burnout beitragenden Faktoren der Arbeitsumgebung identifizieren.
 - Entwickeln Sie einen Plan, der diese Faktoren anspricht.

- A) Richtig. Dies ist die einzige Antwort mit lauter langfristigen Lösungen. Alle anderen Antworten enthalten mindestens eine kurzfristige Lösung. (*Literatur: A, Kapitel 8*)
- B) Falsch. Die Entlassung der Geschäftsführung trägt nicht zur langfristigen Lösung bei, sondern ist, wenn überhaupt, nur ein kurzfristiger Fix. Sich Zeit zu nehmen, um in die Balance zu kommen, ist zwar eine gute Idee, aber es kann sein, dass diese Idee ohne Plan nicht funktioniert.
- C) Falsch. Eine Auszeit zu nehmen ist zwar möglicherweise eine großartige Idee, aber nur eine kurzfristige Lösung. Psychologische Hilfe ist ebenfalls großartig, aber auch sie ist ohne Änderung der zugrunde liegenden Ursachen nur eine kurzfristige Lösung. Die Idee mit den freien Mitarbeitern kann funktionieren und ist großartig.
- D) Falsch. Die Erarbeitung eines Plans und die Identifizierung der Faktoren in der Arbeitsumgebung sind großartige Ideen. Aber es gibt eine bessere Antwort, da Auszeiten nur kurzfristig funktionieren.

10 / 50

Sie arbeiten in einem DevOps-Team als Entwickler und versuchen die Zusammenarbeit zwischen allen Mitgliedern des Teams zu erleichtern. Ihr Team umfasst zwei männliche Senior-Entwickler und eine Junior-Systemadministratorin.

Ihre Team-Mitglieder hatten einen schwierigen Anfang und haben sich nicht vertragen. Sie sind damals eingeschritten und haben versucht, zwischen den Mitgliedern zu vermitteln. Das hat funktioniert, denn Ihr Team streitet sich nicht mehr und erledigt viel mehr Arbeit.

Jetzt bemerken Sie ein neues Muster. Die Junior-Administratorin stimmt den Senior-Entwicklern bei allem zu.

Wie sollten Sie auf dieses Muster reagieren?

- A)** Solange sich die Teammitglieder nicht streiten, sollten Sie gar nichts unternehmen. Endlich verstehen sie sich und das sollten Sie nicht stören.
 - B)** Es ist wichtig, dass der IT-Betrieb alle Bedenken bezüglich eines Builds äußert. Sie sollten die Systemadministratorin daher anleiten weniger entgegenkommend zu sein und sich besser durchzusetzen.
 - C)** Die Senior-Entwickler haben eine größere Verantwortung. Sie sollten Sie daher bitten, netter zu sein und an die Bedenken des IT-Betriebs zu denken, falls die Junior-Systemadministratorin dies nicht tut.
 - D)** Obwohl Sie selbst ein Entwickler sind, sollten Sie erst einmal im Betriebsteam mitarbeiten, um die Junior-Administratorin anzuleiten.
-
- A)** Falsch. In dieser Situation ist Konfliktlösung gefragt. Das Junior-Mitglied des Teams wendet Entgegenkommen als Methode der Konfliktlösung an. Dies ist keine produktive Art der Konfliktlösung. Jede Rolle muss ihrer Verantwortung gerecht werden, sonst läuft man Gefahr, dass Bedenken bezüglich eines Builds nicht rechtzeitig angesprochen werden.
 - B)** Richtig. Dies ist in dieser Situation die richtige Lösung, denn hier ist Konfliktlösung gefragt. Das Junior-Mitglied des Teams wendet Entgegenkommen als Methode der Konfliktlösung an. Dies ist jedoch keine produktive Art der Konfliktlösung und birgt außerdem die Gefahr, dass der Betrieb Bedenken bezüglich des Builds nicht äußert. (*Literatur: A, Kapitel 7 und 14*)
 - C)** Falsch. Jeder trägt die gleiche Verantwortung. Dies hängt weder vom Grad der Seniorität noch dem Geschlecht ab. Außerdem können Entwickler Probleme bezogen auf den Build gar nicht in der gleichen Art und Weise sehen wie dies die Mitglieder des Betriebs tun. Deshalb müssen bei DevOps die verschiedenen Bereiche interagieren.
 - D)** Falsch. Sie können nicht einfach das Team wechseln, selbst wenn "Vorleben" ein großartiges Coaching ist. Entwickler sehen Probleme bei Builds anders als die Mitglieder des Betriebsteams. Diese Maßnahmen würde also die Qualität der Builds beeinträchtigen.

11 / 50

Warum versprechen Agile und Scrum eine schnellere und besser planbare Software-Entwicklung?

- A) Weil sie vor der Entwicklung eine bessere und vollständige Erfassung und Bearbeitung der Anforderungen ermöglichen
 - B) Weil sie kleine, autonome Teams ermöglichen, die sich selbst organisieren und planen
 - C) Weil sie dem Product Owner die Teilnahme an täglichen Einsatzbesprechungen ermöglichen
 - D) Weil sie es dem Projektmanager ermöglichen, seine Prioritäten bei Bedarf schnell zu ändern
-
- A) Falsch. Dieser Ansatz entspricht eher dem Wasserfall-Modell.
 - B) Richtig. Dies wird von den Scrum und Agile-Prinzipien vorgeschlagen, um schnellere und bessere Ergebnisse zu liefern. (*Literatur: A, Kapitel 4*)
 - C) Falsch. Der Product Owner kann zwar teilnehmen, aber dies spielt für die Einhaltung des Versprechens eine schnellere Softwareentwicklung und mehr Planungssicherheit zu liefern, keine Rolle.
 - D) Falsch. Der Projektmanager sollte die Prioritäten nicht ändern, dies ist die Aufgabe des Product Owners.

12 / 50

Was versteht man unter „Lightweight ITSM“?

- A) Ein ITSM, das sich an der Business Continuity orientiert
 - B) Eine neue ITIL-Version, die als Standard vorgeschlagen wurde
 - C) Eine schlechte Umsetzung der ITIL-Prozesse
 - D) Ein ITSM, das sich am Release Management orientiert
-
- A) Richtig. ITIL erscheint schwerfällig und für die schnellen DevOps-Prozesse ungeeignet. „Lightweight“ ITSM ist ein ITSM, das für DevOps optimiert wurde, sich auf die Business Continuity konzentriert und die erforderlichen Mindestinformationen abdeckt. (*Literatur: C, Kapitel 4iii*)
 - B) Falsch. Eine ITIL-Version, die als Standard vorgeschlagen wurde, gibt es bislang nicht.
 - C) Falsch. „Lightweight“ ITSM ist keine schlechte Umsetzung von ITSM, sondern eine „abgespeckte“ Version, die sich auf die Business Continuity und die Reduzierung des für das IT-Management erforderlichen Arbeitsaufwands konzentriert.
 - D) Falsch. ITSM orientiert sich am Service Management, nicht am Release Management. Im ITSM unterstützt der Release-Prozess den Service.

13 / 50

Sie arbeiten für ein Unternehmen, das Lean- und Agile-Praktiken umsetzt. Ihr CEO ist nicht überzeugt, dass sich die Implementierung einer weiteren neuen Methode, nämlich DevOps, lohnt.

Sie sind DevOps-Experte. Sie sind der Meinung, dass DevOps allen Unternehmen, die Software entwickeln, Vorteile bietet.

Welche Vorteile bietet DevOps für Ihr Unternehmen?

- A) Mit DevOps werden bessere User Stories und fachliche Anforderungen erstellt.
 - B) Mit DevOps verringert sich die Zahl der Releases, so dass Kunden sich nicht regelmäßig mit Updates herumschlagen müssen.
 - C) Mit DevOps kann man Experten schulen, die sowohl in der Entwicklung als auch im Betrieb arbeiten können.
 - D) Mit DevOps erstellt man Code, der schneller bereitgestellt wird und dies wirkt sich positiv auf die Unternehmensergebnisse aus.
-
- A) Falsch. Mit Agile-Praktiken verfassen Sie bereits großartige User Stories und fachliche Anforderungen. DevOps beruht auf Agile-Prinzipien. Selbst bei frühzeitiger Beteiligung des IT-Betriebs werden sich die nach Agile erstellten User Stories und fachlichen Anforderungen bei DevOps nicht stark ändern.
 - B) Falsch. DevOps setzt auf schnellere kontinuierliche Releases, um schneller Mehrwert für das Unternehmen zu generieren. Dies ist ebenfalls ein Lean-Prinzip. Häufige Updates seitens der Kunden stellen kein Problem dar, wenn die neuen Funktionen Mehrwert bieten. Außerdem können Sie häufige Releases bereitstellen, aber die Endbenutzer trotzdem nur zu geplanten Zeiten Updates durchführen lassen.
 - C) Falsch. Ziel von DevOps ist, dass der IT-Betrieb und die IT-Entwicklung gemeinsam im Team arbeiten und nicht dass einzelne Personen beide Rolle erfüllen. Dies mag in sehr kleinen Organisationen möglich sein, aber es ist nicht per se typisch für DevOps.
 - D) Richtig. Lean und Agile-Praktiken sorgen bereits jetzt für die Konzentration auf den Unternehmenswert und ändern die fachlichen Anforderungen. Bei DevOps werden die Releases zur unmittelbaren Unterstützung der Unternehmensergebnisse häufiger an die Produktivumgebung bereitgestellt, indem man sich auf die Erstellung einer Pipeline für die Continuous Deployment (kontinuierliche Bereitstellung) konzentriert. Dabei wird der IT-Betrieb möglichst frühzeitig mit einbezogen und es werden möglichst viele Aktivitäten automatisiert. (*Literatur: C, Kapitel 1*)

14 / 50

Die Geschäftsführung fordert die IT-Abteilung auf, den Business Support und die Anpassung an die Unternehmensziele zu verbessern. Sie sind der CIO und entschließen sich u.a. die Arbeitsbelastung im IT-Betrieb zu reduzieren.

Wie hilft Ihnen DevOps am **besten** dabei, Ihr bestehendes Service Level Management zu verbessern?

- A) Durch den Verzicht auf ITSM Best Practices, denn diese sind einfach ein schwerfälliger Ansatz
 - B) Durch die Erstellung besserer Operating Level Agreements (OLA) zwischen Entwicklung und Betrieb
 - C) Durch die Umsetzung neuer ITSM-Prozesse basierend auf ITIL Version 3
 - D) Durch die Anpassung von ITSM zur Einführung des Konzepts der "erforderlichen Mindestanforderungen" (MRI)
-
- A) Falsch. Ein Verzicht auf ITSM Best Practices führt nicht zu einem besseren SLM.
 - B) Falsch. Die Ausarbeitung besserer OLA zwischen Entwicklung und Betrieb trägt nicht zur Verringerung der Arbeitsbelastung bei, sondern schafft eher mehr Bürokratie. Dies ist ganz sicherlich kein Schwerpunkt von DevOps.
 - C) Falsch. Dies führt zu schwerfälligeren Prozessen, die sich nicht für DevOps eignen.
 - D) Richtig. Die Entwicklung eines Lightweight ITSM zählt zu den wichtigsten Anforderungen bei der Implementierung von DevOps. (*Literatur: C, Kapitel 4iii*)

15 / 50

Welche Umsetzung von DevOps eignet sich für ein Unternehmen, das mit dem System-of-Records-Ansatz (SoR) arbeitet am **besten** ?

- A) Kollaboration
 - B) Continuous Delivery (Kontinuierliche Lieferung)
 - C) Die Toyota-Methode
-
- A) Richtig. Der Schwerpunkt liegt in diesem Fall auf der schnellen und häufigen Bereitstellung von IT-Services und einem zuverlässigen Betrieb, dies eignet sich für SoE und SoR am besten. (*Literatur: C, Kapitel 8*)
 - B) Falsch. Dies eignet sich am besten für Lieferanten von digitalen Produkten
 - C) Falsch. Dies eignet sich am besten für IT Service Provider

16 / 50

Was ist der **größte** Vorteil, den die Verwendung des Obeya-Systems bietet?

- A) Das System erleichtert Kundenbeschwerden und sorgt so dafür, dass das Team genügend Feedback für die kontinuierliche Verbesserung bekommt.
 - B) Das System geht mit Stress innerhalb des Teams um, damit die Teammitglieder ein nachhaltiges Tempo beibehalten können.
 - C) Das System verbessert die tägliche Auswertung von Fehlern, sorgt damit für weniger Nacharbeit und reduziert die Weitergabe von Bugs an andere Arbeitsplätze
 - D) Das System ermöglicht ausgehend von dem aktuellen Status durch die schnelle Erhebung und Weitergabe von Informationen eine schnelle Entscheidungsfindung.
-
- A) Falsch. Obeya macht Kundenbeschwerden nicht einfacher.
 - B) Falsch. Obeya kann zwar beim Umgang mit stressigen Situationen helfen, dies ist jedoch nicht der Hauptvorteil.
 - C) Falsch. Obeya trägt nicht zur Verbesserung der Fehlerauswertung bei.
 - D) Richtig. Obeya oder "war room" bezeichnet ein Produktionssystem / Lean Tool bei Toyota, das Projektteams hilft, alle relevanten Informationen im Blick zu halten. Es fördert schnelle Interaktionen und die schnelle Kommunikation in kleinen Teams und sorgt so dafür, dass die für eine Entscheidung erforderlichen Informationen schneller eingeholt werden. (*Literatur: C, Kapitel 7iii*)

17 / 50

Sie arbeiten in einem Software-Unternehmen, das gerade erst beginnt, DevOps einzusetzen. Sie realisieren, dass sich nicht nur die Kultur der Organisation, sondern auch ihre Praktiken und Tools ändern sollten. Das Unternehmen überwacht bereits Softwareprojekte bezüglich eventueller Probleme.

Sie schlagen die Ergänzung folgender Maßnahmen vor:

- Automatisierung des Change Management-Prozesses
- Implementierung von Authentifizierung, damit nur autorisierte Änderungen durchgeführt werden

Warum ist dies notwendig?

- A)** Durch Automatisierung sollen Änderungen schneller und ohne Vertrauensverlust umgesetzt werden. Durch Authentifizierung sollen Problemlösungsheuristiken und ungeplante Serviceausfälle vermieden werden.
- B)** Durch Automatisierung soll die Zahl der Änderungen verlangsamt werden. Durch Authentifizierung sollen Kunden daran gehindert werden, die Software selbst, ohne Ihr Wissen und ohne Steuerung Ihrerseits zu ändern.
- C)** Die Automatisierung trägt dazu bei, das IT-Betriebsteam überflüssig zu machen. Die Authentifizierung muss umgesetzt werden, weil DevOps-Projekte in einer Cloud-Umgebung durchgeführt werden, die mit höheren Risiken einhergeht.
- A)** Richtig. Zitat: "Wir sind zwar im Allgemeinen keine Fans von Nutzungsbeschränkungen und Freigabeprozessen, aber wenn es um Ihre produktive Infrastruktur geht, ist dies von wesentlicher Bedeutung. Da wir der Meinung sind, dass Sie Ihre Testumgebungen genauso behandeln sollten wie Ihre Produktivumgebungen sollte folglich für beide Umgebungen der gleiche Prozess gelten. Bei der Produktivumgebung sind Nutzungsbeschränkungen von größter Wichtigkeit um unautorisierte Zugriffe nicht nur von Externen zu verhindern, sondern auch von Mitarbeitern – selbst Mitarbeitern des IT-Betriebs. Anderenfalls ist die Versuchung zu groß, sich, wenn etwas schief geht, in die fragliche Umgebung einzuloggen und zu versuchen, das Problem zu lösen (ein Prozess, der manchmal höflich als heuristisches Lösungsverfahren bezeichnet wird). Dies ist aus zwei Gründen fast immer eine ganz schlechte Idee. Erstens führt es für gewöhnlich zu Serviceausfällen (die Mitarbeiter versuchen tendenziell das System neu zu booten oder irgendwelche Service Packs anzuwenden). Zweitens gibt es, wenn später etwas schief geht, keine Aufzeichnungen darüber, wer was wann gemacht hat. Das heißt, es ist unmöglich, die Ursache eines Problems herauszufinden. In dieser Situation kann man ebenso gut die Umgebung ganz neu aufsetzen, damit man wenigstens ihren Zustand kennt." (*Literatur: B, Kapitel 11*)
- B)** Falsch. Durch Automatisierung kann man Änderungen schneller und ohne Vertrauensverlust in den Prozess durchführen. Durch Authentifizierung kann man Kunden zwar an der Änderung der Software hindern, dies ist aber nicht das Hauptziel.
- C)** Falsch. Das Betriebsteam ist niemals überflüssig. Die Mitarbeiter sind aber vielleicht weniger verärgert und belastet. DevOps-Projekte können, müssen aber nicht in einer Cloud-Umgebung durchgeführt werden. Die Authentifizierung trägt zwar zur Sicherheit der Software in einer Cloudumgebung bei, dies ist aber nicht das wichtigste Ziel.

18 / 50

Virtualisierung und Cloud Computing sind Techniken die DevOps-Praktiken unterstützen und erleichtern können.

Inwiefern unterstützen sie DevOps?

- A) Cloud Computing ermöglicht Remote-Zugriff und gibt dadurch dem Kunden mehr Kontrolle.
 - B) Virtuelle Umgebungen lassen sich einfacher standardisieren und nutzen Hardware effizienter.
 - C) Virtuelle Infrastruktur ist leichter verständlich und erfordert keine Instandhaltung.
-
- A) Falsch. Cloud Computing erleichtert zwar den Remote-Zugriff, das alleine bedeutet jedoch noch nicht, dass der Kunde tatsächlich mehr Kontrolle hat. Beides unterstützt DevOps in keiner Weise.
 - B) Richtig. Die Virtualisierung erleichtert die Konsolidierung der Continuous Integration (kontinuierlichen Integration) und der Testinfrastruktur und kann daher Entwicklungsteams als Service angeboten werden. Außerdem bietet sie mehr Effizienz was die Nutzung der Hardware anbelangt. Die Virtualisierung ermöglicht es Ihnen, eine Hardwarekonfiguration für physische Umgebungen zu standardisieren und gleichzeitig virtuell die verschiedensten Umgebungen und Plattformen zu betreiben. (*Literatur: B, Kapitel 11*)
 - C) Falsch. Eine virtuelle Infrastruktur ist per se nicht einfacher zu verstehen. Auch eine virtuelle Infrastruktur muss instandgehalten und verwaltet werden, wenn auch nicht in gleichem Umfang wie eine Infrastruktur vor Ort. Dies ist also kein gültiges Argument.

19 / 50

Service Level Agreements (SLA) sind für jedes Projekt wichtig, weil das, was Sie mit dem Kunden vereinbaren, spezifiziert wird. Bei DevOps dienen die SLA darüber hinaus noch einem weiteren wichtigen Zweck.

Welcher Zweck ist dies?

- A) Die Kunden sind dafür verantwortlich, die SLA für das DevOps-Team zu erstellen. Es ersetzt daher eine förmliche Aufgabe des Betriebsteams.
 - B) Im SLA kann der Kunde alle seine nicht fachlichen Anforderungen spezifizieren, damit sich das Entwicklungsteam auf diese konzentrieren kann.
 - C) Das SLA spezifiziert das akzeptable Service Level. Die Entwicklung sollte das SLA verstehen und den Betrieb dabei unterstützen, das SLA aufrechtzuerhalten.
-
- A) Falsch. Das SLA ist immer ein Vertrag zwischen dem Kunden und dem Bereich, der den Service bereitstellt, deshalb tragen beide Seiten zum Inhalt des SLA bei.
 - B) Falsch. Einige der nicht fachlichen und der fachlichen Anforderungen mögen vom SLA kommen, die meisten Anforderungen aber werden direkt und nicht über das SLA an das DevOps-Team gestellt.. Außerdem ist die Bereitstellung der nicht-fachlichen Anforderungen nicht der Zweck eines SLA.
 - C) Richtig. In der Regel sind die im SLA festgelegten Bedingungen für den Betrieb am wichtigsten. Das Entwicklungsteam sollte das Betriebsteam unterstützen und ihm seine Aufgabe so leicht wie möglich machen. Genau das ist der Unterschied zwischen DevOps und der normalen Entwicklung. (*Literatur: B, Kapitel 12*)

20 / 50

Ihr Team muss für ein neues Produkt eine Deployment Pipeline entwickeln. Im Rahmen der Continuous Integration (kontinuierlichen Integration) müssen Sie die Commit-Phase der Pipeline definieren. Sie erörtern diese Phase mit den Mitgliedern Ihres Teams.

Der Process Master sagt: „Die Definition of Done (Definition von 'fertiggestellt') sollte in oder vor der Commit-Phase festgelegt werden. Ist der Code bei der Durchführung des Commits nicht Done (fertiggestellt), sollte die Arbeit eingestellt werden.“

Ist diese Aussage richtig?

- A) Ja. Ist die Arbeit nicht Done (fertiggestellt), dann hat der Process Master seine Aufgabe nicht gut erfüllt. Dies sollte unmittelbar behoben werden.
 - B) Ja. Mit Arbeit, die nicht Done (fertiggestellt) ist, sollte kein Commit durchgeführt werden, da dies dem Kunden keinen Mehrwert bietet.
 - C) Nein. Die Definition of Done (Definition von 'fertiggestellt') wird nur in Besprechungen mit dem Kunden festgelegt. Darauf zu warten, würde die Arbeit zu sehr verzögern.
 - D) Nein. Die Arbeit in einer Deployment Pipeline sollte kontinuierlich weitergeführt werden. Ist der Code nicht Done (fertiggestellt), dann muss er nur inaktiv sein.
-
- A) Falsch. Der Process Master hat die Aufgabe für eine Definition of Done (Definition von 'fertiggestellt') zu sorgen. Wird mit einem Code, der nicht Done (fertiggestellt) ist, ein Commit durchgeführt, so sollte die Arbeit eingestellt werden. Das heißt aber nicht unbedingt, dass der Process Master seine Aufgabe nicht gut macht.
 - B) Richtig. Ist die Arbeit nicht Done (fertiggestellt), dann liefert der Start der Deployment Pipeline nicht genügend Wert für den Kunden. In Anbetracht des One-piece-flow (in einem Rutsch) würde dies den Fluss der wertvolleren Arbeit verzögern. (*Literatur: B, Kapitel 3*)
 - C) Falsch. Die Definition of Done (Definition von 'fertiggestellt') ist einer der ersten Aspekte, die in einem Projekt vereinbart wird. Sie wird nicht bei Besprechungen mit dem Kunden festgelegt. Die Definition of Done (Definition von 'fertiggestellt') sollte bereits zu Beginn der Codeentwicklung bekannt sein. Wie würden Sie sonst wissen, wann der Code fertig ist?
 - D) Falsch. Stimmt etwas mit dem Code nicht oder bietet der Code keinen Mehrwert, so ist dies Grund genug, die Deployment Pipeline anzuhalten, um dies zu beheben oder für mehr Wertschöpfung in der One-piece-flow (in einem Rutsch) Pipeline zu sorgen.

21 / 50

Was ist der **größte** Vorteil, den mehr Teamdiversität mit einer größeren Bandbreite an persönlichen Hintergründen und Kulturen bietet?

- A) Es führt zu mehr Erfahrungen und unterschiedlichen Perspektiven.
 - B) Es führt zu weniger Reibung im Team.
 - C) Diversität begrenzt die Originalität und die Möglichkeit neuer Einblicke.
 - D) Es dauert länger bis man zu einer spezifischen Entscheidung kommt.
-
- A) Richtig. Diversität bedeutet die Teammitglieder haben ganz unterschiedliche Hintergründe. Dazu zählen verschiedene Aspekte, wie z.B. Rasse, Geschlecht, sexuelle Orientierung, soziale Schicht, Bildungsstand, Sprache und Berufserfahrung. Alle diese einzigartigen Aspekte sorgen dafür, dass die Organisation über einen größeren Erfahrungsschatz und mehrere verschiedene Standpunkte verfügt. *(Literatur: A, Kapitel 7)*
 - B) Falsch. Es besteht die Möglichkeit, dass mehr Diversität auch zu mehr Druck und Reibung führt, da Menschen mit verschiedenen kulturellen Werten zusammenarbeiten müssen.
 - C) Falsch. Mehr Diversität bedeutet mehr unterschiedliche Standpunkte. Dies führt in der Regel zu mehr Originalität.
 - D) Falsch. Dies gilt in der Regel als Nachteil. (Obwohl langsamere Entscheidungsprozesse durchaus auch von Vorteil sein können.) Mehr Diversität kann dazu führen, dass es länger dauert bis man einen Konsens erreicht.

22 / 50

Sie arbeiten für ein kleines Unternehmen mit nur einem DevOps-Team. Ihr DevOps-Team arbeitet an einer Anwendung, die mehrere Komponenten umfasst. Manche dieser Komponenten sind neu, andere müssen lediglich aktualisiert werden.

Derzeit hat jede Komponente ihre eigene Deployment Pipeline. Das Team ist stolz, dass es bei den meisten Deployment Pipelines Continuous Delivery (kontinuierliche Lieferung) durchführt. Das Team produziert viel und seine Ergebnisse sind qualitativ hochwertig.

Was sollten Sie tun?

- A) Sie sollten die verschiedenen Pipelines beibehalten und das Team ermutigen, Continuous Delivery noch weiter auszubauen.
 - B) Sie sollten nur die Pipelines beibehalten, in denen Continuous Delivery implementiert ist und die anderen Pipelines ohne Continuous Delivery zusammenlegen.
 - C) Sie sollten nur zwei Pipelines haben: Eine für die Entwicklung und eine für die Instandhaltung
 - D) Sie sollten Ihr Team überzeugen, sich auf eine Pipeline zu beschränken und ihm die Risiken erklären, die mit mehreren Pipelines einhergehen.
-
- A) Falsch. Wenn ein Team mehr als eine Deployment Pipeline hat, so birgt dies Risiken. Die Priorisierung der verschiedenen Pipelines gestaltet sich schwierig und manchmal muss die Arbeit an Einträgen unterbrochen werden, weil eine andere Pipeline Priorität hat. Dies führt wiederum zu einem gewissen Maß an Chaos und Multi-Tasking, das DevOps zu vermeiden versucht. Mit nur einer einzigen Deployment Pipeline steigen Effizienz und Unternehmenswert.
 - B) Falsch. Jede Deployment Pipeline profitiert von Continuous Delivery (kontinuierlicher Lieferung). Continuous Delivery-Praktiken senken jedoch nicht das Risiko, das mit mehreren Deployment Pipelines einhergeht.
 - C) Falsch. Selbst wenn man nur zwei Deployment Pipelines hat und selbst wenn diese fachlich aufgeteilt sind, gelten die gleichen Risiken.
 - D) Richtig. Das ist die richtige Lösung. Ihr Team verfügt hoffentlich aus seiner Übung mit Continuous Delivery über genügend Erfahrung, um die Continuous Delivery in der gesamten Pipeline umzusetzen. (Literatur: B, Kapitel 13)

23 / 50

Sie bewerten das mittelständisch bis große Unternehmen Company Builders, das vor einigen Jahren die DevOps-Praktiken eingeführt hat.

Company Builders hat Sie damit beauftragt, den aktuellen DevOps-Reifegrad im Unternehmen zu bestimmen. Danach sollen Sie Verbesserungsvorschläge unterbreiten. Das Unternehmen möchte wissen, auf welchen Bereich es sich konzentrieren soll, um die nächste Stufe zu erreichen: *Stufe 2 - Quantitativ gesteuert*.

Sie stellen fest, dass der Reifegrad in den meisten Bereichen der Stufe 1 - konsistent - entspricht. Es gibt jedoch zwei Ausnahmen:

- **Environments and Deployment.** Dieser Bereich ist für abgestimmte Deployments zuständig und prüft Release- und Rollback-Prozesse.
- **Build Management and Continuous Integration (kontinuierliche Integration).** In diesem Bereich werden regelmäßig automatisierte Builds und Tests durchgeführt. Jeder Build kann in einem automatisierten Prozess vom Source Code heraus wiederhergestellt werden.

Bestimmen Sie mit Hilfe der bereitgestellten Informationen den Reifegrad dieser beiden Bereiche. Geben Sie dann Ihre Empfehlung ab, was schwerpunktmäßig verbessert werden sollte.

Auf welchen dieser beiden Bereiche sollte das Unternehmen Company Builders sich konzentrieren, bevor es zur Stufe 2 übergeht?

- A) Der Reifegrad von Environments and Deployment sowie Build Management und Continuous Integration (kontinuierliche Integration) entspricht jeweils der Stufe 0. Es sollte an beiden Umgebungen gleichzeitig gearbeitet werden.
- B) Environments and Deployment sowie Build Management und Continuous Integration (kontinuierliche Integration) entspricht jeweils der Stufe 1 oder höher. Um Fortschritte zu erzielen, sollte der Reifegrad in anderen Bereichen gesteigert werden.
- C) Der Reifegrad von Environments und Deployment entspricht der Stufe 0. Der Reifegrad von Build Management and Continuous Integration (kontinuierliche Integration) entspricht der Stufe 1. Das Unternehmen sollte sich zuerst auf Environments and Deployment konzentrieren.
- D) Der Reifegrad von Environments and Deployment entspricht der Stufe 2. Der Reifegrad von Build Management and Continuous Integration (kontinuierliche Integration) entspricht der Stufe 0. Das Unternehmen sollte sich ausschließlich auf Build Management and Continuous Integration (kontinuierliche Integration) konzentrieren.
- A) Falsch. Der Reifegrad von Bereich 1 entspricht der Stufe 2 und der Reifegrad von Bereich 2 entspricht Stufe 0.
- B) Falsch. Der Reifegrad von Bereich 1 entspricht der Stufe 2 und der Reifegrad von Bereich 2 entspricht Stufe 0.
- C) Falsch. Der Reifegrad von Environments and Deployment entspricht bereits der Stufe 2, so dass daran nicht mehr gearbeitet werden muss.
- D) Richtig. Der Reifegrad von Bereich 2 entspricht der Stufe 0. Hier sollte zuerst der Reifegrad der Stufe 1 erreicht werden, damit die Organisation dann in einem zweiten Schritt den Reifegrad der Stufe 2 anstreben kann. (*Literatur: B, Kapitel 15*)

24 / 50

Ihr Unternehmen verkauft seinen Kunden einen Online-Backup-Dienst. Jetzt hat einer der Kunden Sie gebeten, neue Funktionen in Ihren Service zu implementieren. Der Kunde möchte die neuen Funktionen innerhalb einer Woche. Anderenfalls sucht er sich einen anderen Anbieter.

Sie glauben, dass die neuen Funktionen wichtig sind und wissen, dass das Entwicklungsteam diese schnell bauen kann. Sie stoßen jedoch auf folgende Probleme:

- Die Tester brauchen lange, um Bugs zu schließen.
- Die Tester finden Bugs, die von den Entwicklern schon lange behoben wurden.
- Demonstrationen der funktionierenden Anwendung sind selten.
- Showcases finden nur selten statt.

Was ist Ihr Problem und wie sollten Sie dieses lösen?

- A)** Sie stellen fehlerhafte Codes bereit. Um dieses Problem zu beheben, sollten Sie für ein besseres Verständnis des Deployment-Prozesses sorgen, die Kollaboration steigern und disziplinierter arbeiten.
- B)** Sie haben ein schlechtes Configuration Management. Um dies zu lösen, sollten Sie die Kollaboration zwischen Entwicklung und Betrieb fördern und stärker auf Überwachung, Protokollierung und Virtualisierung setzen.
- C)** Ihr Prozess der Continuous Integration (kontinuierlichen Integration) ist nicht ordentlich verwaltet. Um dies zu lösen, sollten Sie automatisierte Tests und die Commit-Phase beschleunigen und ein besseres Verständnis des Prozesses der Continuous Integration fördern.
- D)** Ihre Teststrategie ist nicht wirksam. Um dies zu lösen, sollten Sie die Tests automatisieren und die Kollaboration zwischen den Testern und dem restlichen Team verstärken.
-
- A)** Richtig. Stellt man fehlerhaften oder unzureichenden Code bereit, so führt dies zu langen Bereitstellungszeiten, einer geringen Teamgeschwindigkeit, Skepsis bezüglich der Release-Daten, Vertrauensverlust in die Umgebung der Continuous Integration, eine längere Zeit bis zur Behebung der Bugs, dem Auffinden von Bugs, die die Entwickler schon lange behoben haben sowie weniger Demonstrationen und Showcases. Die genannte Lösung eignet sich zur Behebung dieser Probleme. *(Literatur: B, Kapitel 15)*
- B)** Falsch. Ein schlechtes Konfigurationsmanagement führt insbesondere zu unerklärlichen Produktionsausfällen, nicht beherrschbaren Ereignissen bei der Bereitstellung, mehr Zeit für die Konfiguration der Umgebung und einer längeren Wiederherstellungsdauer im Falle eines Ausfalls. Die genannte Lösung eignet sich zur Behebung dieser Probleme.
- C)** Falsch. Wird der Prozess der Continuous Integration nicht ordentlich verwaltet, so hat dies zur Folge, dass Code seltener als einmal täglich bereit gestellt und die Commit(-Phase) unterbrochen wird. Außerdem führt es zu langen Integrationsphasen zwischen den Releases. Die genannte Lösung eignet sich zur Behebung dieser Probleme.
- D)** Falsch. Eine nicht wirksame Teststrategie hat zur Folge, dass Bugs wiederholt auftreten, viel Zeit auf das Beheben von Bugs aufgewendet wird, viele Beschwerden seitens der Kunden eingehen und führt zu Produkten von geringer Qualität und gestressten Entwicklern. Die genannte Lösung eignet sich zur Behebung dieser Probleme.

25 / 50

Welchen Vorteil bietet die Continuous Integration (kontinuierliche Integration) im Rahmen des wirksamen DevOps?

- A) Umfassende Testzyklen vor dem Release von Funktionen
 - B) Lange Zeiträume zwischen den einzelnen Releases von Funktionen
 - C) Häufigere und pünktlichere Releases von Funktionen
- A) Falsch. Man braucht zwar vernünftige Testmethoden, nicht aber umfassende Testzyklen. Außerdem wäre dies kein Vorteil.
- B) Falsch. Genau das Gegenteil ist der Fall.
- C) Richtig. Die Continuous Integration (kontinuierliche Integration) trägt zu schnelleren und häufigeren Releases bei, weil die Integration automatisiert abläuft. (*Literatur: B, Kapitel 3*)

26 / 50

Eine multinationale Organisation will die Commits aus den dezentralen Standorten in einer zentralen Ablage in Dallas, im US-Bundesstaat Texas, zusammenführen und hat dabei einige Probleme. Die dezentralen Standorte sind u.a. Mexico City, Paris, San Diego und England. Die Regionen führen ihre Commits zu unterschiedlichen Zeiten durch. Manchmal ist es unklar, ob Tests fehlgeschlagen sind oder nicht.

Vier verschiedene Praktiken sind möglich:

1. Auf einen fehlgeschlagenen Build sollte man keinen Code einchecken.
2. Lassen Sie alle Commit Tests vor dem eigentlichen Commit immer lokal laufen oder sorgen Sie dafür, dass der für die Continuous Integration (kontinuierliche Integration) zuständige Server dies für Sie erledigt.
3. Warten Sie bis die Commit-Tests abgeschlossen sind, bevor Sie weitermachen.
4. Fehlgeschlagene Tests sollten nicht auskommentiert werden

Welche dieser Lösungen eignen sich am besten, um sicherzustellen, dass die dezentralen Standorte ihre aktuellen Probleme ansprechen?

- A) 1 & 2
 - B) 1, 2 & 3
 - C) 2 & 3
 - D) 2, 3 & 4
- A) Falsch. 1 eignet sich hier nicht, 3 & 4 sind ebenfalls wichtig.
- B) Falsch. 1 ist nicht wichtig.
- C) Falsch. 4 ist ebenfalls wichtig.
- D) Richtig. Diese drei Praktiken eignen sich für dieses Szenario am besten. Da es keinen Beleg dafür gibt, dass die dezentralen Standorte Code auf einen gebrochenen Build einchecken, gilt diese Praktik hier nicht. (*Literatur: B, Kapitel 3*)

27 / 50

Denken Sie an die Anatomie einer elementaren Deployment Pipeline.

Welche Stufe stellt sicher, dass das System auf funktionaler und nicht-funktionaler Ebene funktioniert?

- A) Automatisierte Abnahmetests
 - B) Build und Unit Test
 - C) Manuelle Abnahmetests
 - D) Versionskontrolle
- A) Richtig. Automatisierte Abnahmetests sorgen dafür, dass das System auf der funktionalen und nicht-funktionalen Ebene funktioniert und das Systemverhalten die Anforderungen der Benutzer und die Kundenspezifikationen erfüllt. (*Literatur: B, Kapitel 8*)
- B) Falsch. Build Tests und Unit Tests sorgen dafür, dass der neue Code an sich in Ordnung ist. Die Integration in den bestehenden Build wird dabei nicht überprüft.
- C) Falsch. Dies könnte eine korrekte Antwort sein. Aber in einer funktionierenden Deployment Pipeline erwarten wir automatisierte Abnahmetests.
- D) Falsch. Die Versionskontrolle dient zur Reparatur gebrochener Builds oder zur Problembhebung. Sie dient nicht zum Nachweis, dass das System auf funktionaler oder nicht-funktionaler Ebene gut funktioniert.

28 / 50

Eine der Best Practices von DevOps besteht darin, für das Software-Deployment in jeder Umgebung, in der Ihre Anwendung läuft, den gleichen Prozess zu verwenden. So wird sichergestellt, dass der Build wirksam getestet wird. Sie automatisieren Ihren Prozess für den Build und die Bereitstellung mit Hilfen von Skripten.

Welche Methode eignet sich hierfür am **besten**?

- A) Sie nutzen pro Umgebung ein Skript und pflegen diese Skripte im Rahmen des Versionskontrollsystems.
 - B) Sie nutzen für jede Umgebung ein spezifisches Skript, um die Unterschiede zwischen den Umgebungen zu berücksichtigen.
 - C) Sie nutzen für jede Umgebung das gleiche Skript und verwenden für spezifische Konfigurationen manuelle Parameter.
 - D) Sie nutzen für das Deployment in den einzelnen Umgebungen das gleiche Skript und verwalten die Konfigurationsinformationen separat.
- A) Falsch. Diese Antwort ist aufgrund des Aufwands für die Pflege der Skripte und der möglichen Fehler, die sich durch diese komplexe Lösung ergeben, eine schlechte Wahl.
- B) Falsch. Unterschiedliche Skripte könnten bei Änderungen zu Problemen führen. Diese wiederum können Probleme im Prozess verursachen, die schwer rückzuverfolgen bzw. zu beheben sind.
- C) Falsch. Bei Build (Bau) und Deployment (Bereitstellung) sollte keine manuelle Interaktion erfolgen. Dieser Prozess sollte automatisiert werden, um ihn zu beschleunigen und sicherzustellen, dass er fehlerfrei ist.
- D) Richtig. Um sowohl den Build als auch den Delivery-Prozess wirksam zu testen, sollten die gleichen Skripte verwendet werden. Die Unterschiede zwischen den Umgebungen, wie URI, IP, etc. sind im Rahmen des Konfigurationsmanagement-Prozesses zu verwalten. (*Literatur: B, Kapitel 6*)

29 / 50

Nach dem Release eines neuen IT Service fällt ein Job im IT-Betrieb unerwartet weg.

Was ist **kein** stichhaltiges Argument hierfür?

- A) Der Gatekeeper hat die Zusammenhänge zwischen den Arbeitsaufgaben und den Kriterien für die Serviceabnahme (SAC) nicht berücksichtigt.
 - B) Die nicht-funktionalen Anforderungen waren aufgrund einer unzureichenden User Story nicht eindeutig.
 - C) Der Service Master und der Benutzer konnten sich nicht auf das End-of-Life (EOL) des bereitzustellenden Service einigen.
 - D) Wie die Kriterien für die Serviceabnahme (Service Acceptance Criteria, SAC) belegen, gab es zu wenig Arbeit in den Entwicklungsprozessen.
-
- A) Falsch. Dies ist durchaus eine mögliche Ursache.
 - B) Falsch. Dies ist durchaus eine mögliche Ursache.
 - C) Richtig. Dies ist keine mögliche Ursache, weil das End-of-Life nicht vorab bestimmt werden kann, sondern immer erst festgelegt werden muss, wenn der Service live ist. (*Literatur: C, Kapitel 7*)
 - D) Falsch. Dies ist durchaus eine mögliche Ursache.

30 / 50

Das Unternehmen AppBC arbeitet mit DevOps. Das Unternehmen hat Continuous Deployment (kontinuierliche Bereitstellung) umgesetzt, verfügt über eine solide Deployment Pipeline mit hoch automatisierten Abnahmetests und liefert täglich neue Software an die Produktivumgebung.

AppBC hat eine große Datenbank und viele Benutzer. Das Unternehmen verfügt über eine umfassende und solide Strategie für Kapazitätstests. Da die IT-Umgebung von AppBC relativ groß und komplex ist, tauchen mit jeder neuen Version einige Bugs in der Produktivumgebung auf.

Welche Strategie könnte AppBC am **besten** helfen, diese Bugs zu vermeiden?

- A) Die Einführung von Canary-Releases
 - B) Die Automatisierung der Kapazitätstests
 - C) Die Senkung der Lieferrate
 - D) Die Einführung von Blue-Green Deployment
-
- A) Richtig. Bei einem Canary-Release wird die neue Version einer Anwendung nur auf einen Teil der Server in der Produktivumgebung ausgerollt, um schnelles Feedback zu erhalten. So werden eventuelle Probleme in der neuen Version schnell entdeckt, ohne sich auf die meisten Benutzer auszuwirken. Die schrittweise Erhöhung der Last bei gleichzeitiger Messung der Reaktionszeiten und anderer Leistungsmetriken senkt das mit einem neuen Softwarerelease verbundene Risiko und beschleunigt die Suche nach Bugs und deren Lösung. (*Literatur: B, Kapitel 10*)
 - B) Falsch. Die Kapazitätstests sollten in diesem Kontext bereits automatisiert sein. Aber selbst wenn dies nicht der Fall wäre, würde eine Automatisierung der Tests in diesem Szenario nicht zur Entdeckung der Bugs beitragen.
 - C) Falsch. Dies entspricht nicht den DevOps-Praktiken.
 - D) Falsch. Für ein Blue-Green-Deployment sind zu viele Ressourcen erforderlich, die bei diesem Szenario sehr kostspielig wären. Darüber hinaus könnte diese Strategie bei einer großen Datenbank zu Ausfallzeiten oder Situationen führen, in der die Datenbank nur im Read-Only-Modus verfügbar ist, falls ein Rollback erforderlich wird. Ein Blue-Green-Deployment trägt darüber hinaus nicht zur Verbesserung der Kapazitätstests bei..

31 / 50

DevOps bedient sich wichtiger Konzepte von Agile, die von dem Toyota Produktions-System (TPS) abgeleitet wurden.

Warum ist One-piece-flow (in einem Rutsch) bei der Einführung von DevOps wichtig?

- A) Es ermöglicht Ihrem Team, in einem nachhaltigen Tempo mit planbarer Teamgeschwindigkeit zu arbeiten.
 - B) Es hilft dem Team, sich auf die Ergänzung von Funktionen zu konzentrieren, die dem Unternehmen möglichst viel Mehrwert bieten.
 - C) Es stärkt die gemeinsame Verantwortung der Teammitglieder für die Arbeit.
 - D) Es begrenzt die Anzahl der gleichzeitig erledigten Aufgaben und reduziert so Engpässe.
-
- A) Falsch. Für dies sorgt Rhythm (Rhythmus). Setzt man einen nachhaltigen Rhythm sorgt man für ein planbares Tempo, vermeidet Burnout und sorgt für eine gute Work-Life Balance der Mitarbeiter.
 - B) Richtig. Dies erreicht man durch One-piece-flow. Mit One-piece-flow können Sie die Funktion wählen oder aktualisieren, die Ihnen am meisten Wert bietet und diese als nächste in die Pipeline stellen. So bleiben Sie agil. Arbeitet man nur an einer Funktion, dann kann man in der Regel Work-in-Progress (WiP) beschränken, damit die Funktion auch tatsächlich fertig wird. (*Literatur: C, Kapitel 7*)
 - C) Falsch. Dies ist zwar wichtig für DevOps, aber nicht direkt mit Work-in-Progress (WiP), One-piece-flow, Rhythm oder Ji Koutei Kanketsu (JKK) verbunden.
 - D) Falsch. Genau darum geht es bei Work-in-Progress (WiP) Durch die Beschränkung der Zahl der Aufgaben, an denen Sie und Ihr Team gleichzeitig arbeiten, stellen Sie die Ihnen zugewiesenen Aufgaben erfolgreich fertig. Sie vermeiden Engpässe, weil Sie Work-in-Progress (WiP) so beschränken, dass immer jemand auf die fertiggestellte Arbeit wartet.

32 / 50

Die S-Gruppe, ein mittelständischer KfZ-Zulieferer beliefert ein großes Unternehmen, die T Motors-Gruppe mit KfZ-Teilen. Das Geschäft mit der T Motors-Gruppe beträgt fast 60% des Gesamtumsatzes der S-Gruppe.

Der Vorstand trifft sich, um über die neue Zusammenarbeit zu sprechen. Die T Motors-Gruppe verlangt, dass die S-Gruppe auf Just-in-Time-Lieferung umstellt. Anderenfalls wird die T Motors-Gruppe ihr Geschäft von der S-Gruppe abziehen. Die S-Gruppe würde das nicht überleben, die Umstellung auf Just-in-Time (JiT) ist damit dringend. Die Änderung muss innerhalb von 6 Monaten abgeschlossen sein. Es bleiben also maximal 5 Monate zur Vorbereitung.

Eine der umzusetzenden Maßnahmen ist die Rückverfolgung von Teilen mittels Radio Frequency Identification (RFID). Dies soll zur Transparenz des Produktionsprozesses beitragen. Zur leichteren Umstellung auf RFID muss der aktuelle Prozess schnell überprüft werden.

Die CIO wird gebeten, das Management des Change Prozesses zu übernehmen. Sie glaubt, dass die Umstellung gelingen kann, wenn die S-Gruppe mit einem DevOps-Ansatz ein Minimalrelease erstellt. Im Idealfall wird für die Produktion mit RFID zuerst ein Konzept entwickelt und das Produktionsleitsystem mit Hilfe der RFID-Daten dann erst im letzten Schritt umgesetzt. Aufgrund der Kürze der Zeit können diese Schritte jedoch nicht der Reihe nach durchgeführt werden und sollen daher gleichzeitig erfolgen.

Die CIO stellt Em für das Projekt ab. Em ist Scrum Master. Das Entwicklungsteam bereitet sich auf den Bau einer Deployment Pipeline vor.

Em stellt fest, dass das Entwicklungsteam mit Begeisterung bei der Sache ist und fleißig arbeitet. Allerdings mangelt es dem Team etwas an Disziplin. Außerdem muss die Releasefrequenz erhöht werden.

Auf was sollte sich Em **zuerst** konzentrieren?

- A) Em sollte sich auf die Kommunikation konzentrieren, da diese bei DevOps am wichtigsten ist. Em sollte zuerst das Eis im Team brechen und dann einige Regeln für die Kommunikation festlegen.
- B) Em sollte zuerst eine Wertstromanalyse durchführen und mit dem Team eine schlanke Produktion nach dem Prinzip One-piece-flow (in einem Rutsch) erstellen, denn straffe Prozesse sind äußerst wichtig.
- C) Em sollte zuerst die Infrastruktur und das Arbeitsumfeld mit den Teammitgliedern besprechen, da DevOps am wirkungsvollsten ist, wenn alle Tools (Hilfsmittel) und Praktiken funktionieren.
- D) Em sollte zuerst, alle Stakeholder versammeln, diese über DevOps aufklären und sie bitten, den für DevOps erforderlichen Kulturwandel zu unterstützen.

- A) Falsch. Kommunikation ist für ein funktionierendes DevOps Team zwar wichtig, aber dieses Team hat schon vor der Einführung von DevOps zusammengearbeitet. Für den ersten Schritt gibt es daher einfacher zu erreichende Ziele. Wichtiger ist es, dass das Team den Kundenmehrwert in den Fokus seiner Überlegungen stellt. Sobald dies der Fall ist, kann man sich daran machen, die Kommunikationsfähigkeiten zu verbessern.
- B) Richtig. Die Prozesse müssen verschlankt und die Wertstromanalyse erstellt werden, damit das Team mit möglichst geringem Aufwand eine möglichst hohe Wertschöpfung erzielen kann. Danach sollte man definieren, was sich ändern könnte und ändern sollte und was bereits großartig funktioniert. Dies betrifft auch Tools (Hilfsmittel), Kommunikation und Kultur. DevOps muss nicht in jedem Unternehmen gleich sein, aber es muss dazu führen, dass sich das Team darauf konzentriert, Mehrwert für den Kunden zu schaffen. (*Literatur: B, Kapitel 1, und A, Kapitel 1 und 2*)
- C) Falsch. Tools (Hilfsmittel) und Automatisierung sind zwar wichtige Elemente von DevOps, die es zu berücksichtigen gilt, aber sie haben nicht unbedingt die höchste Priorität. Stattdessen sollte man besser prüfen, ob man die Produktion oder den Serviceprozess insgesamt vereinfachen, verkürzen oder kostengünstiger gestalten kann, indem man eine Wertstromanalyse erstellt und diese mit dem Team bespricht. Dies dient als Auslöser für den notwendigen Kreislauf der kontinuierlichen Verbesserung. Alle anderen Schritte kommen dann danach.
- D) Falsch. Ein Kulturwandel ist in vielen Unternehmen unerlässlich. Dies hat jedoch nicht unbedingt die höchste Priorität. Genauso wenig sollte man die Stakeholder zusammenrufen, um diesen Kulturwandel zu erörtern. Der Kulturwandel sollte sich aus der Frage ergeben, wie man die höchste Wertschöpfung für den Kunden oder Stakeholder erzielen kann. Eine großartige Idee wäre es, die verschiedenen Stakeholder darum zu bitten, einen Blick auf die Wertschöpfungsanalyse zu werfen und ihre Ideen beizutragen.

33 / 50

Ihr DevOps-Team arbeitet gut und in einem nachhaltigen Tempo zusammen. Sie planen genügend Pufferzeiten in den Prozess ein, so dass das Team über genügend Zeit und Konzentration verfügt, um die Builds sorgfältig zu prüfen und zu testen. Derzeit erfolgen Deployment und Testen in Ihrem Team manuell. Das Tempo des Teams ist hoch genug, um dem Unternehmen regelmäßig großen Mehrwert zu liefern.

Ihr CEO hat Sie um Ihren Rat bezüglich der Automatisierung in diesem Team gebeten.

Welchen Rat sollten Sie ihm geben?

- A)** Automatisieren Sie so viel Sie können, damit das Team mehr Funktionen ergänzen und noch schneller einen Mehrwert für das Unternehmen demonstrieren kann.
 - B)** Automatisieren Sie die Abnahmetests, aber nicht die Deployments, denn der manuelle Prozess ist sicherer.
 - C)** Automatisieren Sie die Deployments, um die Zykluszeit zu verkürzen, aber nicht das Testen, damit man von den Bugs lernen kann.
 - D)** Führen Sie keine Automatisierung bei diesem Team ein, weil die derzeit von dem Team ausgeführten Methoden großartige Ergebnisse erzielen.
-
- A)** Richtig. Man sollte stets so viel wie möglich automatisieren. So erhält man mehr Kontrolle über den Prozess und kann schneller einen größeren Mehrwert für das Unternehmen demonstrieren. (*Literatur: B, Kapitel 1 und 8*)
 - B)** Falsch. Die Automatisierung von Abnahmetests ist zwar eine großartige Idee, aber manuelle Deployments sind nicht sicherer.
 - C)** Falsch. Die Automatisierung von Releases ist eine tolle Idee, aber durch manuelles Testen lernt man nicht mehr von Bugs als durch automatisiertes Testen.
 - D)** Falsch. Das Team leistet erstaunliche Arbeit, aber wenn Sie Aktivitäten, die automatisiert werden können, nicht automatisieren, verschwenden Sie das Potenzial des Teams.

34 / 50

Eine CIO teilt Michael, ihren zuverlässigsten Mitarbeiter und Scrum Master, in ein Projekt ein. Das Entwicklungsteam bereitet sich auf den Bau einer Deployment Pipeline vor.

Michael vertraut den guten Absichten und der Spontanität des Entwicklungsteams, wünscht sich aber etwas mehr Disziplin. Außerdem sollte die Releasefrequenz erhöht werden. Michael möchte, dass das Entwicklungsteam häufigere Releases umsetzt.

Ein Teammitglied sagt dazu: „Am wichtigsten ist die Automatisierung der Deployment Pipeline. Daher sollten wir die Deployment Pipeline zuerst automatisieren“.

Ist diese Aussage richtig?

- A) Ja, das ist richtig. Die Automatisierung der Deployment Pipeline ist für die Verbesserung der Effizienz am wichtigsten.
 - B) Ja, das ist richtig. Wenn man sich auf die Erstellung einer automatisierten Deployment Pipeline konzentriert, vermeidet man potenzielle Probleme, die zu einem späteren Zeitpunkt auftreten können.
 - C) Nein, das ist nicht richtig. Erste Priorität sollte sein, das Prinzip des One-piece-flow (in einem Rutsch) und einen soliden Deploymentprozess umzusetzen. Die Automatisierung des Prozesses kann später erfolgen.
 - D) Nein, das ist nicht richtig. Nicht die Deployment Pipeline, sondern der Testprozess sollten zuerst automatisiert werden.
-
- A) Falsch. Die Deployment Pipeline sollte anfangs immer nach dem Prinzip One-piece-flow (in einem Rutsch) gebaut werden. Dies funktioniert sehr gut ohne Automatisierung. Funktioniert die Deployment Pipeline stabil, dann kann man den Prozess bei Gelegenheit automatisieren. Dies sollte jedoch stets erst im zweiten Schritt, nach dem Bau einer soliden Deployment Pipeline erfolgen.
 - B) Falsch. Die Deployment Pipeline sollte anfangs immer nach dem Prinzip One-piece-flow (in einem Rutsch) gebaut werden. Dies funktioniert sehr gut ohne Automatisierung. Funktioniert die Deployment Pipeline stabil, dann kann man den Prozess bei Gelegenheit automatisieren. Dies sollte jedoch stets erst im zweiten Schritt, nach dem Bau einer soliden Deployment Pipeline erfolgen.
 - C) Richtig. Die Deployment Pipeline sollte anfangs immer nach dem Prinzip One-piece-flow (in einem Rutsch) gebaut werden. Dies funktioniert sehr gut ohne Automatisierung. Funktioniert die Deployment Pipeline stabil, dann kann man den Prozess bei Gelegenheit automatisieren. Dies sollte jedoch stets erst im zweiten Schritt, nach dem Bau einer soliden Deployment Pipeline erfolgen. (*Literatur: B, Kapitel 5*)
 - D) Falsch. Das automatisierte Testen zählt zu den zentralen Aktivitäten. Aber wenn Sie die Wahl haben, entweder eine solide Deployment Pipeline zu bauen oder funktionierende Tests zu automatisieren, so sollten Sie sich stets zuerst auf den Bau einer soliden Deployment Pipeline konzentrieren. Steht diese, kann man die Tests automatisieren und so die Effizienz verbessern.

35 / 50

Ihr Unternehmen stellt seine Arbeitsweise auf DevOps um. Ihr Team ist in diese Änderung eingebunden. Die Teammitglieder diskutieren die Best Practices für die Commit-Phase des Codes.

Ihr Kollege Sun sagt: „Wenn ein Build fehlschlägt und keiner die Verantwortung übernimmt, sollten wir den Verantwortlichen ermitteln und ansprechen, damit er den Build reparieren kann.“

Ist dies eine gute Idee?

- A) Ja. Denn fehlgeschlagene Builds können nur von den Verursachern behoben werden. Daher sollte man die Verantwortlichen identifizieren, auch wenn das für sie unangenehm ist.
 - B) Ja. Denn, wer den Fehlschlag eines Build verursacht hat, sollte auch stets die Verantwortung dafür übernehmen. Tun Sie dies nicht, dann werden Ihre Kollegen diese Regel möglicherweise durchsetzen.
 - C) Nein. DevOps baut auf eine Kultur frei von Schuldzuweisungen. Will ein Kollege die Verantwortung nicht übernehmen, dann sollte man ihn nicht dazu zwingen.
 - D) Nein. Zuerst sollte man den Build reparieren. Danach sollte man sich die Zeit nehmen, den Verantwortlichen zu identifizieren und zu bestrafen.
-
- A) Falsch. Es ist zwar wahrscheinlich am einfachsten, den Problemverursacher das Problem suchen zu lassen, aber es ist nicht notwendig. DevOps baut auf eine Kultur frei von Schuldzuweisungen. Will ein Kollege die Verantwortung nicht übernehmen, dann sollte man ihn auch nicht dazu zwingen. Jemanden zu etwas zu zwingen, zeugt nicht von Respekt.
 - B) Falsch. DevOps baut auf eine Kultur frei von Schuldzuweisungen. Will ein Kollege die Verantwortung nicht übernehmen, dann sollte man ihn auch nicht dazu zwingen. Jemanden zu etwas zu zwingen zeugt nicht von Respekt.
 - C) Richtig. Jemanden zu etwas zu zwingen, zeugt nicht von Respekt. Fehler zu machen ist OK. Die Mitglieder des Teams arbeiten kollaborativ zusammen, um Fehler und Herausforderungen gemeinsam zu meistern. (*Literatur: B, Kapitel 3 und A, Kapitel 4*)
 - D) Falsch. Der Build muss nicht repariert werden, Sie können einen Rollback auf eine frühere Version durchführen. Und selbst wenn es keine schlechte Idee ist, einen Build zu reparieren, jemanden für einen Fehler zu bestrafen ist definitiv eine schlechte Idee. DevOps baut auf eine Kultur frei von Schuldzuweisungen. Will ein Kollege die Verantwortung nicht übernehmen, so sollte man ihn nicht dazu zwingen. Jemanden zu etwas zu zwingen, zeugt nicht von Respekt.

36 / 50

Das Entwicklungsteam von XAppGo ist mit seinen aktuellen Testpraktiken zahlreichen Herausforderungen begegnet. Aktuell setzt das Team für die Abnahme einen manuellen Testprozess ein. Die Entwickler glauben, dass die von ihnen erstellte Unit-Testsuite ausreichend Schutz vor Regressionen bietet.

Das Entwicklungsteam gibt bei jedem Release 1 Million für die manuellen Abnahmetests aus. Die obere Leitung hat vorgegeben, dass das Entwicklungsteam automatisierte Abnahmetests einführen muss, um die Gesamtkosten für das Testen zu reduzieren und die Zahl der Code-Fehler und Regressionen in der Produktivumgebung auf ein Mindestmaß zu beschränken.

Welche Grundsätze müssen Sie im Hinblick auf die Automatisierung bei der Festlegung der Abnahmekriterien für Ihre Anwendung befolgen?

- A) Agile Grundsätze
 - B) ATAM-Grundsätze
 - C) INVEST-Grundsätze
-
- A) Falsch. Die zur Erstellung wartbarer Testsuites für die Abnahmetests empfohlenen Prinzipien werden mit dem Akronym INVEST bezeichnet. Genaue Informationen darüber, warum ATAM und Agile nicht empfohlen werden, gibt es nicht. Agile bietet keine spezifische Orientierung bzw. keine Prinzipien für die Testautomatisierung.
 - B) Falsch. Die zur Erstellung wartbarer Testsuites für die Abnahmetests empfohlenen Prinzipien werden mit dem Akronym INVEST bezeichnet. Genaue Informationen darüber, warum ATAM und Agile nicht empfohlen werden, gibt es nicht.
 - C) Richtig. Abnahmetests werden von den Abnahmekriterien abgeleitet. Folglich müssen Sie bei der Erstellung der Abnahmekriterien für Ihre Anwendung bereits an die Automatisierung denken und die INVEST-Prinzipien befolgen. Das Akronym INVEST steht hier für Independent (unabhängig), Negotiable (verhandelbar), Valuable (nützlich), Estimable (quantifizierbar), Small (klein) und Testable (prüfbar). (*Literature: B, Kapitel 8*)

37 / 50

Was ist der **effektivste** Mechanismus für die automatisierte Datenmigration?

- A) Die Erstellung eines Versionierungsschemas für die bestehenden Daten und deren Pflege mit Hilfe der Versionskontrolle
 - B) Die Erstellung und Verwaltung kleinerer Datensätze, damit die Migration einfacher wird
 - C) Die Sicherstellung, dass alle Skripte vor der Datenmigration ordnungsgemäß geprüft wurden
 - D) Die Sicherstellung, dass man über ein Rollback-Verfahren verfügt, falls die Migration fehlschlägt
-
- A) Richtig. Die Versionierung der Datenbank ist der beste Mechanismus für die automatisierte Datenmigration. (*Literatur: B, Kapitel 12*)
 - B) Falsch. Dies ist nicht der beste Mechanismus für die automatisierte Datenmigration, sondern konzentriert sich hauptsächlich darauf, wie man Datensätze effektiver verwalten kann.
 - C) Falsch. Dies ist nicht die beste Antwort. Hier geht es eher um Testaktivitäten als um Migrationsaktivitäten.
 - D) Falsch. Hier geht es eher um Maßnahmen zur Wiederherstellung, falls die Migration fehlschlägt.

38 / 50

Das Unternehmen X-AppGo hat Schwierigkeiten mit seinen Rollback-Prozessen. Bei der Ausführung von Rollback-Skripten kommt es häufig zu kritischen Datenverlusten in den Datenbanken der Produktivumgebung.

Wann können Rollback-Skripten**nicht** ohne kritische Datenverluste ausgeführt werden?

- A) Das Rollback-Skript löscht Daten, die nur von der neuen Version genutzt werden.
 - B) Im Rahmen des Rollback-Skripts muss man eine Spalte von einer in die andere Tabelle verschieben.
 - C) Das Rollback-Skript fügt Daten aus temporären Tabellen ein.
-
- A) Falsch. In diesem Fall löscht das Rollback-Skript nur einige Daten, die nur von der neuen Version verwendet werden und führt bei einem Rollback nicht zu einem kritischen Datenverlust.
 - B) Falsch. Das Rollback-Skript verändert das Datenbankschema auf eine Art und Weise, die nicht zu einem kritischen Datenverlust führt.
 - C) Richtig. Bei diesem Szenario ist ein Rollback-Skript nicht möglich. (*Literatur: B, Kapitel 12*)

39 / 50

ACMECONST hat bei der Bereitstellung von Software-Upgrades für seine Anwendung und bei den Hardware-Aktualisierungen seiner Router und Switches viele Ausfälle der Anwendung und der Hardware erlebt.

Nachdem diese Ausfälle während des für die Instandhaltung reservierten Zeitfensters aufgetreten sind, war es für das Unternehmen sehr schwierig, den ursprünglichen Zustand wieder herzustellen. Die Wiederherstellung hat das normalerweise für die Instandhaltung vorgesehen Zeitfenster überschritten und die Ausfallzeit ihrer kritischen Anwendungen verlängert.

Eine automatisierte Bereitstellung und autonome Infrastruktur kann in einer solchen Situation hilfreich sein. Dabei gibt es jedoch Einiges zu berücksichtigen.

Was muss sorgfältig gemanagt werden, um das Risiko einer Störung beim Deployment in die Produktivumgebung zu reduzieren?

- A) Die detaillierten Überwachungsprotokolle zur Fehlerbehebung bei Ausfällen im Rahmen von Anwendungsupdates
 - B) Externe Integrationspunkte, wie z. B. externe Systeme und Services
 - C) Server-Konfigurationen und die zugrundeliegenden Informationen zum User Account
 - D) Automatisierungstools und autonome Architektur
-
- A) Falsch. Die Fehlerbehebung erfolgt nach dem Upgrade. Daher ist dies kein valider Punkt, um das Risiko des Deployments in die Produktivumgebung zu reduzieren.
 - B) Richtig. Dieser Punkt muss sorgfältig gemanagt werden, um das Risiko bei einem Deployment in jede Art von Produktivumgebung zu reduzieren. (*Literatur: B, Kapitel 11*)
 - C) Falsch. Dies wird im Rahmen der Information über das Betriebssystem und dessen Konfiguration sowohl für die Test- als auch für die Produktivumgebungen abgedeckt und ist folglich kein Punkt, den Sie separat managen.
 - D) Falsch. Um das Risiko zu reduzieren, muss man nicht die Tools selbst verwalten. Sie helfen bei der korrekten Prozessimplementierung und dem richtigen Kontext für eine automatisierte Bereitstellung und eine autonome Infrastruktur.

40 / 50

Das Unternehmen X-AppGo hat Schwierigkeiten mit seiner zentralen Applikation. Die Applikation arbeitet nicht ordnungsgemäß mit anderen externen Applikationen zusammen. Diese externen Applikationen müssen effektiv bestimmte Datenvariablen erhalten, damit bestimmte Calls ausgeführt werden können. Die Kernapplikation wird von einem Team entwickelt, das das Unternehmen aus nachvollziehbaren Unternehmensgründen beibehalten möchte.

Einer der Entwickler schlägt vor, eine Komponente aus der X-AppGo Codebasis auszugliedern, um das Schnittstellenproblem zu bearbeiten.

Was sind in diesem Fall gute Argumente für diese Ausgliederung und die separate Betrachtung einer Komponente?

- A) Die Anpassung einiger Plugins in der Codebase in eine monolithische Codebase
 - B) Die Begrenzung von Auswirkungen von Änderungen und das Erleichtern von Änderung der Codebase
 - C) Die X-AppGo-Codebase muss aufgesplittet und von verschiedenen Teams verwaltet werden
 - D) Es gibt keine guten Argumente für diese Maßnahme und die Kompilierung wird mehr Zeit in Anspruch nehmen
-
- A) Falsch. Bei der Erstellung einer Komponente bewegt man sich von einer monolithischen Codebase hin zu einer modularen Codebase beruhend auf Komponenten. Dies ist also eigentlich das Gegenteil.
 - B) Richtig. Die Verantwortlichkeiten sollten bei der Softwareentwicklung und -pflege klar beschrieben sein. Dies wiederum beschränkt die Auswirkung von Veränderungen und erleichtert das Verständnis und die Änderung der Codebase. (*Literature: B, Kapitel 13*)
 - C) Falsch. Da diese Maßnahme, die X-AppGo-Anwendung nicht in unabhängige Komponenten trennt, besteht keine Notwendigkeit, die Unternehmensgründe in Frage zu stellen und das Team aufzusplitten.
 - D) Falsch. Es gibt viele gute Argumente für diese Maßnahmen, die Antwortmöglichkeit 2 ist eines davon. Außerdem nimmt die Kompilierung und die Verbindung des Codes bei einer modularen Codebase weniger Zeit in Anspruch, nicht mehr.

41 / 50

Selbst bei den kleinsten Anwendungen besteht eine Abhängigkeit zu anderen Komponenten oder Libraries (Bibliotheken). Das Verstehen und die Verwaltung der Abhängigkeiten ist daher eine der wichtigsten Aktivitäten innerhalb der Continuous Deployment (kontinuierlichen Bereitstellung), um den Flow in der Deployment Pipeline aufrechtzuerhalten.

Sie haben eine Anwendung gebaut, die zwei Libraries nutzt. Jede dieser Libraries basiert auf einer dritten Basis-Library. Allerdings beziehen sie sich auf verschiedene Versionen. Dies führt zu einer besonderen Abhängigkeit.

Welche Lösung eignet sich am **besten**, um diese Abhängigkeit zu beheben oder zu vermeiden?

- A) Fügen Sie alle Libraries zu einer einzigen Library zusammen, damit Sie sich direkt auf die Library beziehen und das Problem so vermeiden können
 - B) Verwalten Sie die Libraries mit Hilfe der Versionskontrolle. So sehen Sie es unmittelbar, wenn Sie diese Art der Abhängigkeit schaffen
 - C) Behalten Sie alle Ihre Abhängigkeit mit Hilfe von Post-Its auf einer großen Tafel im Blick, damit Sie den Flow überwachen können
 - D) Prüfen Sie nur kleine Teile Ihrer Werkzeugkette (Toolchain), damit Sie eventuelle Probleme beim Check-in leicht beheben können
-
- A) Falsch. Dies ist keine gute Idee. Nutzen Sie stattdessen die Versionskontrolle, um sich auf die neueste Library zu beziehen oder nutzen Sie ein Automatisierungs-Tool, das Sie dabei unterstützt, automatisch die neueste Version der Libraries zu nutzen.
 - B) Richtig. Dies ist eine gute Lösung. Außerdem hilft Ihnen die Versionskontrolle in Ihren Libraries Probleme zu beheben, die von Nutzern verursacht werden, die eine alte Version Ihrer Software betreiben. Die Nutzung eines Automatisierungs-Tools, das Sie dabei unterstützt, die neueste Version der Libraries zu nutzen ist ebenfalls eine gute Lösung. (*Literatur: B, Kapitel 13*)
 - C) Falsch. Damit können Sie zwar möglicherweise den Flow verfolgen, es ist aber keine Lösung für den Umgang mit Abhängigkeiten. Kanban-Boards helfen Ihnen nicht wirklich bei der Implementierung der Versionskontrolle.
 - D) Falsch. Sie sollten stattdessen die vollständige Werkzeugkette (Toolchain) einchecken. So finden Sie gegenseitige Abhängigkeiten oder Inkompatibilitäten viel schneller und zuverlässiger.

42 / 50

In einer Umgebung der Continuous Deployment (kontinuierlichen Bereitstellung) ist es wichtig, dass alles versionskontrolliert ist, damit Sie Fehler schnell finden oder bei Bedarf ein Rollback durchführen können.

Bei einem binären Output, wird jedoch von einer Versionskontrolle **abgeraten**.

Warum wird hier eine Ausnahme gemacht?

- A) Ein binärer Output befindet sich tendenziell in großen Dateien, die bei jedem Build verändert und automatisch aktualisiert werden.
 - B) An binären Dateien arbeiten verschiedene Mitglieder des Teams, daher ist hier eine Versionskontrolle nicht praktisch.
 - C) Binärer Output dient als Input für die Compiler, die bereits der Versionskontrolle unterliegen.
 - D) Es besteht keine Notwendigkeit dazu, da im Rahmen des Build-Prozesses regelmäßig eine erneute Kompilierung erfolgt.
-
- A) Richtig. Der Output ist erstens sehr groß und wird für jeden Check-in der kompiliert wird und das automatische Testen durchläuft erneut erstellt. Zweitens kann der Output vom Source Code wieder erstellt werden, indem man das Build Skript erneut laufen lässt. (*Literatur: B, Kapitel 2*)
 - B) Falsch. Dies ist nicht der Grund.
 - C) Falsch. Binärer Output ist der Output von Compilern nicht der Input. Ansonsten ist die Argumentation richtig.
 - D) Falsch. Die regelmäßige erneute Kompilierung im Rahmen eines neuen Build-Prozesses ist nicht ratsam. Richtig ist, dass Sie durch ein erneutes Kompilieren einen neuen binären Output erhalten.

43 / 50

Sie suchen nach einem ganzheitlichen Ansatz für das Management Ihrer IT-Infrastruktur

Welche zwei Prinzipien sind Ihnen dabei am **besten** behilflich?

- A)
 - 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Änderungskontrolle unterliegt.
 - 2. Dank Monitoring (Überwachung) und Event Management sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
 - B)
 - 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Änderungskontrolle unterliegt.
 - 2. Dank Instrumentierung und Incident Management sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
 - C)
 - 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Versionskontrolle unterliegt.
 - 2. Dank aktuellem Incident Management und Event Management sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
 - D)
 - 1. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Versionskontrolle unterliegt.
 - 2. Dank Instrumentierung und Monitoring (Überwachung) sollten Sie stets über den tatsächlichen Zustand Ihrer Infrastruktur informiert sein.
- A) Falsch. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Versionskontrolle und nicht der Änderungskontrolle unterliegt. Punkt 2 ist ebenfalls nicht korrekt: Event Management ist nicht korrekt und das wichtige Prinzip der Instrumentierung wird außer Acht gelassen.
- B) Falsch. Sie sollten den Wunschzustand Ihrer Infrastruktur mit Hilfe einer Konfiguration spezifizieren, die der Versionskontrolle und nicht der Änderungskontrolle unterliegt. Punkt 2 ist ebenfalls nicht korrekt: Incident Management ist nicht korrekt und das wichtige Prinzip des Monitoring (Überwachung) wird außer Acht gelassen.
- C) Falsch. Punkt 1 ist korrekt. Punkt 2 ist nicht korrekt: das ist nicht eines der Prinzipien die Ihnen bei einem ganzheitlichen Ansatz am **besten** behilflich sind.
- D) Richtig. Dies sind zwei der Prinzipien die Ihnen bei einem ganzheitlichen Ansatz im Infrastrukturmanagement am **besten** behilflich sind. (*Literatur: B, Kapitel 11*)

44 / 50

Teams, die gut zusammenarbeiten haben synchronisierte Arbeitstickets. Ein technischer Direktor hat sich persönlich davon überzeugt, wie das Betriebsteam funktioniert. Das Betriebsteam legt nach jedem Release die Betriebsinfrastruktur neu fest.

Was ist der **beste** Ratschlag, um diese Praxis zu verbessern?

- A) Das Betriebsteam sollte nichts tun. Es gibt nichts zu verbessern, weil eine solche Neufestlegung stets erfolgen sollte.
 - B) Das Betriebsteam sollte versuchen, die Betriebsinfrastruktur und die Authentifizierung auf die Betriebsumgebung zu modellieren.
 - C) Das Betriebsteam sollte die Betriebsinfrastruktur prüfen, damit dies ein automatisierter Prozess wird.
 - D) Das Betriebsteam sollte sein Wissen über den Deployment-Prozess mit dem Entwicklungsteam teilen.
-
- A) Falsch. Dies ist unnötige Arbeit und kann durchaus verbessert werden.
 - B) Falsch. Das ist zwar nett, aber das Team wird diese Tätigkeit dann ständig wiederholen und dies ist Zeitverschwendung
 - C) Falsch. Es gibt keine eindeutige Möglichkeit, um diesen Prozess ohne Einbindung der Entwicklungsabteilung zu automatisieren.
 - D) Richtig. Das ist der richtige Ansatz: Wissen vermitteln und dann weitere Schritte unternehmen.
(Literatur: A, Chapter 17)

45 / 50

Wann sollte der Betrieb (Operations) die Entwicklung (Development) über eine Änderungen im Betrieb informieren?

- A) Die Entwicklung (Development) muss gar nicht informiert werden. Änderungen im Betrieb betreffen nur das Betriebsteam.
 - B) Umgehend. Die Entwicklung muss baldmöglichst informiert werden.
 - C) Im Scrum-of-Scrum-Meeting am nächsten Morgen.
 - D) Nachdem das Betriebsteam den Abnahmetest durchgeführt hat.
-
- A) Falsch. Die Entwicklung (Development) ist umgehend zu informieren, damit das Team eventuelle Risiken und Probleme voraussehen kann.
 - B) Richtig. Die Entwicklung ist umgehend zu informieren, damit das Team eventuelle Risiken und Probleme voraussehen kann. (Literatur: C, Kapitel 5 und 7)
 - C) Falsch. Die Entwicklung ist umgehend zu informieren, damit das Team eventuelle Risiken und Probleme voraussehen kann.
 - D) Falsch. Die Entwicklung ist umgehend zu informieren, damit das Team eventuelle Risiken und Probleme voraussehen kann.

46 / 50

Sie wollen den Reifegrad Ihrer DevOps-Organisation steigern. Dies ist auf viele verschiedene Arten möglich.

Welche Vorgehensweise eignet sich **nicht** dazu, den Reifegrad Ihrer DevOps-Organisation zu steigern?

- A) Sie legen eindeutige Ziele als Meilensteine fest, damit die Mitglieder Ihres Teams den Wert ihrer täglichen Aktivitäten besser einschätzen können.
 - B) Sie definieren Prozesse eindeutig und ermöglichen es den Mitgliedern Ihres Teams dadurch, Prozesse täglich zu verbessern.
 - C) Sie zeichnen alle Besprechungen auf, damit die Mitglieder Ihres Teams leicht auf sämtliche Kommunikation zugreifen können.
 - D) Sie überwachen die täglichen Aktivitäten und zeichnen diese auf, damit selbst die kleinsten, täglich erzielten Fortschritte identifiziert und entsprechend gefeiert werden können.
-
- A) Falsch. Dies trägt durchaus dazu bei, den Reifegrad der DevOps-Organisation zu steigern.
 - B) Falsch. Dies trägt durchaus dazu bei, den Reifegrad der DevOps-Organisation zu steigern.
 - C) Richtig. Dies trägt nicht zur Steigerung des Reifegrads der DevOps-Organisation bei. Es besteht keine zwingende Notwendigkeit, alle Besprechungen zur späteren Durchsicht aufzuzeichnen. Vereinbarungen müssen protokolliert werden, nicht aber vollständige Besprechungen. (*Literatur: B, Kapitel 15*)
 - D) Falsch. Dies trägt durchaus dazu bei, den Reifegrad der DevOps-Organisation zu steigern.

47 / 50

Sie arbeiten für einen IT-Dienstleister. Im Rahmen Ihres Business Continuity-Plans möchten Sie garantieren, dass Sie die vereinbarten Service Level stets erfüllen können.

Sie wollen die IT Service Continuity sicherstellen.

Wie kann DevOps Sie beim IT Service Continuity Management unterstützen?

- A) Affinität und Kollaboration sind Werte der DevOps-Kultur die dafür sorgen, dass der Service von den Mitgliedern des DevOps-Teams hoch geschätzt wird.
 - B) DevOps bringt absichtlich Chaos in das System und sorgt so für Notfallroutinen und Obeya-Praktiken im Team.
 - C) Maßnahmen zur Risikoreduzierung und Optionen zur Wiederherstellung im Notfall werden wahrscheinlich direkt in den Code eingebaut, denn Betrieb (Operations) und Entwicklung (Development) arbeiten eng zusammen.
 - D) Das Service Level Management spielt bei DevOps eine wichtigere Rolle, denn es muss vom Process Master überwacht werden.
-
- A) Falsch. Die Werte der DevOps-Kultur tragen zwar dazu bei, dass sich die Mitarbeiter wohler fühlen, besser und in einem gleichmäßigen Tempo arbeiten, Fehler machen und aus diesen Fehlern lernen. Dies trägt aber an sich noch nicht zum Service Level Management bei.
 - B) Falsch. Absichtlich für Chaos zu sorgen, wie es das Beispiel des Chaos Monkey-Tools von Netflix zeigt, könnte Ihnen dabei helfen, Risikoreduktionsmaßnahmen und Optionen zur Wiederherstellung in den Code einzubauen. Dies trägt aber an sich noch nicht zum Service Level Management bei. Außerdem sollte Chaos im System durch den Code und nicht durch krisenstabartige Lösungen behoben werden.
 - C) Richtig. Ein systematischer Prozess kann Risiken für die Infrastruktur und Anwendungen vermeiden, vorhersagen oder reduzieren. Diese Risiken stellen eine Gefahr für den Service dar, denn sie können ihn unterbrechen oder Störungen hervorrufen. (*Literatur: B, Kapitel 11, und C, Kapitel 4*)
 - D) Falsch. Das Service Level Management zu überwachen ist nicht in jedem DevOps-Team die vorrangige Aufgabe des Process Master.

48 / 50

Das Unternehmen ACMECONST hat seine globale Präsenz aggressiv ausgebaut und dazu die Zahl der Neueinstellungen und die Zahl der technischen Teams weltweit erhöht. Mit einem Zuwachs von 30% hat das Unternehmen auch seinen Kundenstamm drastisch erhöht.

Seitdem sich das Technik-Team nicht mehr in einem Raum befindet dauern einfache Entscheidungen viel länger. Dies verursacht in der gesamten Organisation wachsende Frustration. Freigaben müssen jetzt mehrere Managementebenen durchlaufen und der Prozess ist umfangreicher und komplexer. Viele Ingenieure sind daher von dem Entscheidungsprozess insgesamt desillusioniert.

Auch bezüglich der Verantwortung und Zuständigkeiten herrscht zunehmend Verwirrung. Die Ingenieure sind der Meinung, dass die zusätzlichen Prozesse und das Mehr an Bürokratie ihre Kreativität einschränken. Dies wirkt sich allmählich auf die Moral aus.

Wie lassen sich die Probleme in diesem Szenario am **besten** lösen?

- A) Behalten Sie die aktuellen Prozesse bei, aber sorgen Sie bei jedem Prozess für eindeutige Rollen, Verantwortlichkeiten und Zuständigkeiten. Führen Sie eine wirksame Methode ein, um Produktivität und Risiko gegeneinander abzuwägen, machen Sie schrittweise Änderungen und schaffen Sie sichere Räume für Experimente.
 - B) Überprüfen Sie die aktuellen Prozesse noch einmal, um zu sehen, wo sie diese straffen können und sorgen Sie bei jedem Prozess für klare Rollen, Verantwortlichkeiten und Zuständigkeiten. Führen Sie eine wirksame Methode ein, um Produktivität und Risiko gegeneinander abzuwägen, machen Sie schrittweise Änderungen und schaffen Sie sichere Räume für Experimente.
 - C) Überprüfen Sie die aktuellen Prozesse noch einmal, um zu sehen, wo sie diese straffen können und sorgen Sie bei jedem einzelnen Prozess für klare Rollen, Verantwortlichkeiten und Zuständigkeiten. Führen Sie eine wirksame Methode ein, um Produktivität und Risiko gegeneinander abzuwägen, machen Sie schrittweise Änderungen und reduzieren Sie Experimente auf das absolut erforderliche Mindestmaß, um unnötige Ausfälle der Anwendung zu vermeiden.
-
- A) Falsch. Die aktuellen Prozesse beizubehalten ist nicht die beste Idee. Das Unternehmen ist gewachsen, und die bestehenden Prozesse funktionieren nicht mehr. Aber einen sicheren Raum für Experimente zu schaffen ist eine gute Idee.
 - B) Richtig. Bei einem wachsenden Unternehmen funktionieren diese Maßnahmen gut. Eine erneute Überprüfung der Prozesse ist notwendig, weil die alten Prozesse nicht mehr funktionieren. Ein sicherer Raum für Experimente ist ebenfalls nötig, um kontinuierliche Verbesserungen zu erleichtern.
(Literature: A, Chapter 15)
 - C) Falsch. Eine Überprüfung der bestehenden Prozesse ist zwar notwendig, aber Experimente auf das absolut erforderliche Mindestmaß zu reduzieren ist eine schlechte Idee. Fehlender Raum für Experimente hemmt die nötige kontinuierliche Verbesserung.

49 / 50

Im Unternehmen X-AppGo gibt es einen Konflikt zwischen dem Betriebsteam in Kolumbien und dem Entwicklungsteam in Irland. Grund für den Konflikt sind unterschiedliche Prioritäten und Ziele. Aufgrund des Konflikts nimmt die Lösung von Problemen, die sich auf den Geschäftsbetrieb auswirken, mehr Zeit in Anspruch.

Welche zentralen Praktiken sollte X-AppGo in Betracht ziehen, um das Konfliktpotenzial zu reduzieren und die Kollaboration zwischen dem Entwicklungs- und dem Betriebsteam zu verbessern?

- A)**
1. Um Konflikte zu vermeiden, sollte X-AppGo es den Teams ermöglichen, unabhängig voneinander zu arbeiten, falls die Teams dies vorziehen.
 2. Die Entwicklungs- und Betriebsteams sollten die volle Unterstützung des Vorstands haben.
- B)**
1. Ein Mitglied des Vorstands sollte die Rolle des Förderers übernehmen und dem DevOps-Team erläutern, wie wichtig eine gute Zusammenarbeit ist.
 2. Das Entwicklungs- und das Betriebsteam sollten in DevOps-Praktiken geschult werden, damit jedes Team die Arbeit des jeweils anderen Teams übernehmen kann.
- C)**
1. Das Entwicklungs- und das Betriebsteam sollten andere Unternehmen besuchen, in denen DevOps erfolgreich angewendet wird.
 2. Das Entwicklungs- und das Betriebsteam sollten mehr Mittel erhalten, um die Teams besser bei der Erfüllung der gestiegenen Anforderungen zu unterstützen.
- D)**
1. Die Teams sollten sich gegenseitig an ihren jeweiligen Standorten besuchen, um den Teamgeist zu stärken und ein Verhältnis des Vertrauens und gegenseitigen Verständnisses aufzubauen.
 2. Wissen und Erfahrungen sollte zwischen dem Entwicklungs- und dem Betriebsteam ausgetauscht und verbreitet werden, um für eine effizientere Zusammenarbeit zu sorgen.
- A)** Falsch. Die Teams sollten zur Zusammenarbeit ermuntert werden und nicht unabhängig voneinander arbeiten dürfen. Sie brauchen sich gegenseitig und müssen voneinander lernen. Kollaboration ist nicht möglich, wenn man sich nie sieht. Unterstützung des Vorstands hilft den Teams nicht bei der Selbstorganisation der Zusammenarbeit.
- B)** Falsch. Ein Förderer im Vorstand mag dazu beitragen, dass sich die Teams wertgeschätzt fühlen. Dies trägt jedoch nicht zur Kollaboration bei. Schulung alleine ist nicht alles. Eine bessere kollaborative Zusammenarbeit kann nur entstehen, wenn die Teams auch tatsächlich zusammenarbeiten.
- C)** Falsch. Besuche bei anderen Unternehmen können zwar inspirierend sein, aber DevOps ist sehr speziell auf das Umfeld des jeweiligen Unternehmens ausgerichtet. Diese Maßnahme trägt nicht direkt zu einer besseren Kollaboration und zur Konfliktreduzierung bei. Die Teams sollten stattdessen zusammenarbeiten und ihr Wissen und ihre Erfahrung austauschen. Eine Erhöhung der bereitgestellten Mittel kann zwar bei Arbeitsüberlastung (d.h. bei zu wenigen Mitarbeitern und zu viel Arbeit) die Situation etwas verbessern, trägt aber nur bis zu einem gewissen Punkt zur Konfliktreduzierung und Förderung der kollaborativen Zusammenarbeit bei.
- D)** Richtig. Dies sind für dieses Szenario die besten Maßnahmen, um Konflikte zu reduzieren und die Kollaboration zwischen dem Betriebs- und dem Entwicklungsteam zu verbessern (*Literatur: A, Kapitel 15*)

50 / 50

Ein Entwicklungsteam interessiert sich für DevOps. Das Team interessiert sich dabei hauptsächlich für die Continuous Integration (kontinuierlichen Integration). Aktuell entwickelt und pflegt das Team mit Scrum-Praktiken 3 größere und 4 kleinere Lösungen. Jeder Sprint dauert 4 Wochen und führt im Schnitt alle 10 bis 15 Tage zum Commit von 1 Release an die Testumgebung sowie zum Commit von 1 Release pro Monat an die Produktivumgebung. Das Team möchte für das Management einen qualitativen Business Case erstellen, um die Unterstützung des Managements für die Investitionen und die Bemühungen zur Einführung der Continuous Integration (kontinuierliche Integration) zu gewinnen.

Welche greifbaren Vorteile der Continuous Integration (kontinuierlichen Integration) eignen sich **am besten** zur Unterstützung des Business Case?

- A) Ein tägliches Deployment an die Testumgebung könnte den Nutzen für das Unternehmen steigern und die Entwicklungskosten drastisch reduzieren.
 - B) Sie unterstützt den Team-Geist. Da das Team bereits mit Scrum arbeitet, bringt Continuous Integration (Kontinuierliche Integration) dem Unternehmen **keinen** messbaren Nutzen.
 - C) Es erhöht die Stabilität und Qualität von Releases durch automatisiertes Testen und erhöht die allgemeine Release-Geschwindigkeit.
 - D) Ein tägliches Release an die Produktivumgebung könnte den Nutzen für das Unternehmen steigern und die Entwicklungskosten drastisch senken.
-
- A) Falsch. Ein schnelleres Deployment an die Testumgebung ist OK und eine der Folgen von Continuous Integration (kontinuierlicher Integration). Dies schafft jedoch keinen Nutzen für das Unternehmen.
 - B) Falsch. Continuous Integration (kontinuierliche Integration) kann dem Team dabei helfen, Releases schneller an die Produktivumgebung zu liefern sowie Bugs schneller und wirtschaftlicher zu finden. Ob das Team Scrum nutzt oder nicht ist dabei unerheblich.
 - C) Richtig. Continuous Integration (kontinuierliche Integration) wird zu einer höheren Release-Geschwindigkeit führen, da jede Veränderung in der Integrationsumgebung getestet wurde und funktionstüchtig ist. CI wird außerdem die Stabilität und Qualität der Software erhöhen, da Fehler und Probleme schneller erkannt und behoben werden können. (Literatur: B, Kapitel 3)
 - D) Falsch. Schnellere Releases in die Produktivumgebung ist ein großer Vorteil von Continuous Delivery (kontinuierliches Liefern) und nicht von Continuous Integration (kontinuierlicher Integration). Bei kontinuierlicher Integration wird der Code in eine produktionsähnliche Umgebung geschoben, so dass der gesamte Programmcode integriert und automatisiert getestet werden kann.

Evaluierung

Die nachfolgende Tabelle zeigt Ihnen die jeweils richtigen Antworten auf die in dieser Musterprüfung enthaltenen Fragen.

Nummer	Antwort	Nummer	Antwort
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

Kontakt EXIN

www.exin.com

