



**Musterprüfung**

Ausgabe 202301

Copyright © EXIN Holding B.V. 2023. 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

Einführung	4
Musterprüfung	5
Antwortschlüssel	24
Beurteilung	57

# Einführung

Dies ist die EXIN DevOps Master™ (DEVOPSM.DE) Prüfung. Es gilt die Prüfungsordnung von EXIN.

Die Prüfung besteht aus 40 Multiple-Choice-Fragen. Zu jeder Multiple-Choice-Frage werden mehrere Antwortmöglichkeiten angeboten. Es gibt jeweils eine richtige Antwort.

Sie können maximal 40 Punkte erreichen. Jede richtige Antwort zählt 1 Punkt. Um die Prüfung zu bestehen, müssen Sie mindestens 26 Punkte erzielen.

Die Bearbeitungszeit beträgt 90 Minuten.

Viel Erfolg!

# Musterprüfung

1 / 40

Welche Lösung ist **am effektivsten**, um Daten automatisiert zu migrieren?

- A) Erstellen von kleinere Datensätze, damit die Migration überschaubar bleibt
- B) Entwickeln eines Rollback-Verfahrens, falls die Migration fehlschlägt
- C) Sicherstellen, dass die Skripte vor der Datenmigration ordnungsgemäß getestet werden
- D) Einrichten einer Datenbankversionierung und Befolgen einer strengen Versionskontrolle

2 / 40

Ein Team hat Continuous Deployment (kontinuierliche Bereitstellung) implementiert und alles wird über Versionskontrolle verwaltet. Das Team hält dies für notwendig, um Fehler schnell identifizieren und bei Bedarf jederzeit ein Rollback durchführen zu können.

Der Teamleiter führt jedoch an, dass es nicht empfehlenswert sei, auch die Binärausgaben über die Versionskontrolle zu verwalten.

Warum sagt der Teamleiter das?

- A) Weil eine erneute Kompilierung fester Bestandteil eines jeden normalen Build-Prozesses ist
- B) Weil Binärausgaben große Dateien sind, die jeden Build verändern und aus dem Quellcode wiederhergestellt werden können
- C) Weil an Binärausgaben mehrere Teammitglieder arbeiten, was die Versionskontrolle erschwert
- D) Weil Binärausgaben Input für die Compiler sind und nicht richtig in der Versionskontrolle verwaltet werden können

### 3 / 40

Ein Team arbeitet an der Erstellung einer Deployment Pipeline. Das Team hat für seinen Prozess erfolgreich Single-Piece-Flow erzeugt und verfügt über einige grundlegende, automatisierte Build- und Deployment Prozesse. Die Unit- und Akzeptanztests werden jedoch nach wie vor manuell ausgeführt. Das Team führt aktuell jeden Monat ein Release durch.

Die Mitglieder diskutieren, was das Team als Nächstes tun sollte, um seine Deployment Pipeline zu verbessern.

- **Aki** sagt: „Wir sollten zuerst unsere Releasefrequenz erhöhen. Die fehlenden Elemente ergeben sich dann aus den Problemen, auf die wir dabei stoßen.“
- **Em** sagt: „Wir sollten die restlichen Tests automatisieren. Sobald wir die Unit- und Akzeptanztests automatisiert haben, können wir mit der Releaseautomatisierung beginnen.“
- **Ken** sagt: „Die Deployment Pipeline ist Bestandteil der Wertschöpfungskette der Organisation. Wir sollten daher zuerst eine Work-in-Progress-Begrenzung (WIP-Limit) implementieren, bevor wir irgendetwas anderes unternehmen.“
- **Mart** sagt: „Es läuft aktuell gut und wir haben Single-Piece-Flow erzeugt. Aktuell gibt es keinerlei Grund, irgendwelche Maßnahmen zu ergreifen. Wir sollten warten, bis die Organisation einen höheren Reifegrad erzielt.“

Wer macht in diesem Szenario den **besten** Vorschlag?

- A) Aki
- B) Em
- C) Ken
- D) Mart

### 4 / 40

Ein Team muss für ein neues Produkt eine Deployment Pipeline entwickeln. Das Team diskutiert über die Commit-Phase der Deployment Pipeline und arbeitet in Richtung Continuous Integration (kontinuierliche Integration).

Ein Teammitglied sagt: „Die Definition of Done (DoD, Definition von 'Fertiggestellt') sollte vor oder spätestens in der Commit-Phase festgelegt werden. Entspricht der Code beim Einchecken nicht der DoD, so sollte die Arbeit daran eingestellt werden.“

Ist diese Aussage richtig?

- A) Ja, weil das Teammitglied seine Arbeit nicht korrekt ausführt und umgehend ermahnt werden muss.
- B) Ja, weil Arbeit, die nicht der DoD entspricht, keinen Mehrwert bietet und daher nicht eingecheckt werden sollte.
- C) Nein, weil die DoD in den Besprechungen mit dem Kunden festgelegt wird und vor der Commit-Phase noch gar nicht vorliegt.
- D) Nein, weil für die Arbeit in der Deployment Pipeline ein steter Flow sichergestellt sein muss und man die Arbeit nie plötzlich einstellen werden darf.

**5 / 40**

Nachdem es Upgrades der Anwendungssoftware bereitgestellt hat, erlebt ein Team viele Fehler und Ausfälle der Anwendung, aber auch der Hardware. Der Versuch, einen früheren, bekanntermaßen funktionierenden Systemzustand wiederherzustellen, dauerte sehr lange. Die Ausfälle führten zu längeren Betriebsunterbrechungen bei kritischen Anwendungen. Das Management hat das Team gebeten, das Risiko von Betriebsunterbrechungen für die nächsten Deployments (Bereitstellungen) zu senken.

Das Team diskutiert, wie dieses Problem zu lösen sei.

Was sollte das Team als **Erstes** unternehmen?

- A) Vollständig automatisieren alle Tests ebenso wie die Build-, Deployment- und Release-Prozesse
- B) Sicherstellen, dass alles in der Versionskontrolle verwaltet wird, um eine schnelle Wiederherstellung zu ermöglichen
- C) Sorgen für Kommunikation zwischen dem Entwicklungs- und dem Betriebsteam
- D) Schulen dem Entwicklungsteam, damit es den Deployment-Prozess versteht

**6 / 40**

Eine Organisation plant die Implementierung von Continuous Integration (kontinuierlicher Integration). Das Entwicklungsteam der Organisation ist mit dieser neuen Arbeitsweise einverstanden und hat bereits Tools (Hilfsmittel) für die Automatisierung ausgewählt, die die Software von der Befehlszeile aus erstellen.

Welche andere Aktivität ist ebenfalls eine Voraussetzung für die kontinuierliche Integration?

- A) Die Beseitigung von absoluten Pfaden. Dies ist für die Integration der Libraries von Drittanbietern erforderlich.
- B) Die Kontrolle des Zugriffs auf die Infrastruktur. Dies sorgt dafür, dass keiner unautorisierte Änderungen vornehmen kann.
- C) Die Dokumentation von Wertströmen. Dies hilft bei der Identifizierung, in welchen Bereichen Verbesserungsbedarf besteht.
- D) Die Einführung von Versionskontrolle. Denn diese zeigt, was wann, von wem und warum getan wurde.

### 7 / 40

Ein Team diskutiert über seine Benutzerabnahmetests. Aktuell wendet das Team viel Zeit, Arbeit und Geld für manuelle Akzeptanztests auf. Das Management hat das Team gefragt, ob es hierfür eine bessere Alternative gibt. Das Management bevorzugt in der Regel die Lösung, die mit dem geringsten Risiko für Produktionsunterbrechungen verbunden ist.

Das Team kommt zu dem Schluss, dass es intern über die erforderlichen Fähigkeiten verfügt und alle Benutzerabnahmetests automatisieren kann. Dies würde jedoch etwas Zeit in Anspruch nehmen und mehr Kosten verursachen. Ein Teammitglied merkt an, dass ein Wettbewerber alle Akzeptanztests durch automatisierte Unit- und Komponententests ersetzt hat.

Sollte das Team den Benutzerabnahmetest in diesem Szenario automatisieren?

- A) Ja, weil Automatisierung auf lange Sicht günstiger ist als manuelles Testen und das Risiko für Produktionsunterbrechungen dadurch niedrig gehalten wird.
- B) Ja, weil dadurch nur noch ein manueller Akzeptanztest pro Monat erforderlich ist, um das Risiko für Produktionsunterbrechungen niedrig zu halten.
- C) Nein, weil manuelle Akzeptanztests sicherstellen, dass alle Fehler vor dem Release entdeckt werden, um das Risiko für Produktionsunterbrechungen niedrig zu halten.
- D) Nein, weil Unit- und Komponententests die Akzeptanztests ersetzen und der Wettbewerber beweist, dass dies das Risiko für Produktionsunterbrechungen niedrig hält.

### 8 / 40

Ein Start-up hat eine App für Haustierbesitzer erstellt. Die App ermöglicht Kunden für einen bestimmten Tag einen Haustiersitter zu finden. Die Unternehmensinhaberin spricht mit einem wichtigen Kunden. Sie geht davon aus, dass das Unternehmen schnell wachsen wird.

Das Entwicklungsteam verfügt aktuell über einen manuellen Prozess für das Testen. Dies hat bislang sehr gut funktioniert.

Was sollte das Entwicklungsteam als Nächstes unternehmen?

- A) Neue Tests für das Betriebsteam erstellen, um das Deployment (Bereitstellung) zu automatisieren
- B) Das manuelle Testen beibehalten und bei Bedarf neue manuelle Tests ergänzen
- C) Warten bis klar ist, wie es für die neuen Kunden Mehrwert schaffen kann
- D) Gemeinsam mit dem Betriebsteam an der Erstellung von automatisierten Tests arbeiten

### 9 / 40

Ein DevOps-Team nutzt die Daten in seinen Datenbanken für die Tests der Softwareentwicklung.

Was ist für dieses Team potenziell das **größte** Problem?

- A) Der Datenzugriff von Standorten außerhalb der Testdatenbank
- B) Die Zugänglichkeit von Produktionsdaten aus dem Homeoffice
- C) Die Verfügbarkeit repräsentativer Testdaten für das automatisierte Testen
- D) Die Sicherheit der verfügbaren Sicherheitsvorkehrungen für die Testdaten



**10 / 40**

Kamala ist als Analystin in der Informationssicherheit dafür zuständig, den aktuellen Software-Entwicklungsprozess eines Open-Finance-Unternehmens zu verbessern. Eines ihrer Hauptziele besteht darin, in der Phase der Code-Entwicklung das Thema Informationssicherheit stärker ins Bewusstsein zu rücken.

Kamalas Ratschlag lautet, dass die Teams einen „Shift Left“ zur Informationssicherheit durchführen sollten. Mit anderen Worten, Informationssicherheit sollte in den Softwareentwicklungszyklus integriert werden, von der Codeerstellung bis zum Betrieb.

Ihr Vorgesetzter Rob ist mit diesem Vorschlag nicht einverstanden. Er schlägt vor, einen Spezialisten einzustellen, der für Informationssicherheit in den Produkten sorgt, wenn diese für das Deployment (Bereitstellung) bereit sind. Rob argumentiert, dass die Entwickler keine Sicherheitsexperten sind, und sich auf die Entwicklung konzentrieren sollten.

Wer hat Recht?

- A) Kamala, weil eine frühere Integration der Informationssicherheit die Einhaltung der maßgeblichen Gesetze und Vorschriften sicherstellt. Dies macht die Erstellung sicherer Software langfristig günstiger.
- B) Kamala, weil die Teammitglieder sich zu Sicherheitsexperten schulen lassen können. Dies spart die Lohnkosten für einen Sicherheitsexperten.
- C) Rob, weil es am besten funktioniert, wenn man ausgehend von den Best Practices im Bereich der Informationssicherheit, eine Reihe von Richtlinien erstellt und dann einen Experten einstellt, der diese Richtlinien in den Produkten implementiert.
- D) Rob, weil die Entwickler sich tatsächlich auf ihre Kernaufgabe, die Entwicklung von Software, konzentrieren können und nicht damit belastet werden sollten, auch Experten im Bereich der Sicherheit zu werden.

**11 / 40**

Was ist der **größte** Vorteil von Continuous Delivery (kontinuierlicher Lieferung)?

- A) Sie ermöglicht transparente Konformität und Leistung der Softwareentwicklung.
- B) Sie sorgt dafür, dass die Softwareentwicklungsteams die Corporate-Governance-Prozesse umsetzen.
- C) Sie konzentriert sich auf die Time-to-Market (Markteinführungszeit) des Deployments (Bereitstellung) und eine verbesserte Geschäftsleistung.
- D) Sie führt Automatisierung ein, um die Anforderung an Einhaltung und Governance von Software zu reduzieren.

### 12 / 40

Ein Team arbeitet an einer Anwendung und möchte Continuous Delivery (kontinuierliche Lieferung) einführen. Bei einer Besprechung zur Prozessverbesserung bespricht das Team, welche Praktik es nutzen sollte, um Continuous Delivery einzuführen.

Was sollte das Team tun?

- A) Das Team sollte Scrum als Managementpraktik kombiniert mit einer vierteljährlichen Release-Planung nutzen, um agiler zu werden.
- B) Das Team sollte daran arbeiten, die Continuous Integration (kontinuierliche Integration) der Branches und das automatisierte Testen seiner Anwendung zu verbessern.
- C) Das Team sollte an einer Kulturveränderung arbeiten und Agile Praktiken für das Change Management sowie Feedbackschleifen für das Berichtswesen nutzen.
- D) Das Team sollte an der Implementierung von Kanban auf Teamebene arbeiten und gleichzeitig eine lose gekoppelte Architektur errichten.

### 13 / 40

Ein Unternehmen hat seinen Entwicklungsprozess radikal von der Wasserfallmethode auf DevOps umgestellt. Das Team möchte die Zeit zwischen den Releases reduzieren, trifft dabei aber auf einige Probleme.

Die vom Team durchgeführte Prozessanalyse hat folgende Probleme aufgezeigt:

- 20 Prozent der Sprint-Dauer wird für das Portieren von Code aufgewendet, weil alle Entwickler an separaten Code-Branches arbeiten.
- Werden Probleme entdeckt, die den Build brechen, so muss ein Rollback der Produktionsumgebung durchgeführt werden.
- Das Team verfügt zwar über ein System zur Versionskontrolle, aber ein Monitoring der Key Performance Indicators (KPI, wichtigsten Leistungskennzahlen) findet nicht statt.
- Je höher die Zahl der Releases, desto mehr Kundenfeedback, einschließlich Beschwerden, erhält das Team.

Was hilft diesem Team **am besten** die Zeit zwischen den Releases zu reduzieren?

- A) Die Einführung von Trunk-basierter Entwicklung, um das Portieren von Code überflüssig zu machen
- B) Das Reproduzieren von Testabbrüchen, um Rollbacks in der Produktionsumgebung zu vermeiden
- C) Die Nutzung von Canary-Releases, um die Zahl der Kundenbeschwerden zu reduzieren
- D) Der verstärkte Einsatz von Telemetrie für das Monitoring von KPI, die den Geschäftswert zeigen

### 14 / 40

Was zählt **nicht** zu den Vorteilen einer lose gekoppelten Architektur?

- A) Die Entwickler können unabhängig an Änderungen der Software arbeiten.
- B) Die Systeme können unabhängig voneinander verändert und validiert werden.
- C) Das Projektmanagement wird nur minimal überwacht.
- D) Unit-Tests und Integrationstests laufen automatisiert.

**15 / 40**

Tina ist für die Verbesserung der Architektur und Anwendungsumgebungen eines Produkts zuständig. Bei ihrer ersten Bewertung der Anwendungen stellt sie Folgendes fest:

- Das Team kann jederzeit Deployments (Bereitstellungen) durchführen, vorausgesetzt es benachrichtigt die Service Owner der betroffenen Services.
- Die Anwendungen könnten mit Hilfe unabhängiger Komponenten bereitgestellt werden.
- Obwohl alle Teammitglieder die Anwendungen bereitstellen könnten, hat nur ein Entwickler den entsprechenden Zugriff und die Erlaubnis, dies zu tun.
- Um eine Verbindung zu den angrenzenden Diensten herzustellen, von denen die Anwendungen abhängen, muss das Team eine Reihe von Skripten ausführen.

Wie kann Tina ihrem Team **am besten** dabei helfen, eine bessere Infrastrukturmgebung zu schaffen?

- A) Indem sie dem Team hilft, eine lose gekoppelte Architektur zu erstellen, um unabhängig von anderen Services, zu denen eventuell eine Abhängigkeit besteht, Code oder ein Release der Anwendung bereitstellen zu können.
- B) Indem sie einen Hackathon mit allen technischen Führungskräften und erfahrenen Entwicklern organisiert, um die aktuelle Infrastruktur zu erneuern und durch eine neue, eng gekoppelte Architektur zu ersetzen.
- C) Indem sie die angrenzenden Anwendungen neu erstellt, um Abhängigkeiten zu beheben und dem Team so zu ermöglichen, Skripte schneller auszuführen und seine Anwendungen unabhängig von anderen Anwendungen bereitzustellen.
- D) Indem sie die Teammitglieder darin schult, die Anwendungen sicher bereitzustellen, ihnen die entsprechenden Privilegien einräumt und sie im Change Advisory Board (CAB) aufnimmt.

**16 / 40**

Eine Deployment Pipeline umfasst in der Regel mehrere Feedbackschleifen.

Was ist der **größte** Vorteil, den die Nutzung von Feedbackschleifen bietet?

- A) Bessere Kommunikation zwischen dem Entwicklungs- und dem Betriebsteam
- B) Häufigere und zuverlässigere Lieferungen qualitativ hochwertiger Software an die Benutzer
- C) Einbindung der Kunden in mehreren Phasen des Entwicklungsprozesses
- D) Regelmäßige Durchführung von Benutzerabnahmetests, um die Meinung der Stakeholder einzuholen

**17 / 40**

Ein Software-Entwicklungsteam wird kritisiert, weil es Features bereitstellt, die nicht den Kundenerwartungen entsprechen.

Wie kann das Team diese Situation **am besten** verbessern?

- A) Einen separaten Zyklus von Benutzerabnahmetests erstellen vor dem Deployment (Bereitstellung)
- B) Die Anforderungen anderer Stakeholder in der Priorität zurückstufen oder vollständig entfernen
- C) Die Anforderungen zu Beginn der Entwicklungsprojekte statisch gestalten oder festlegen
- D) Die Erkenntnisse der Stakeholder während des gesamten Lebenszyklus der Entwicklung aktiv einholen

**18 / 40**

DevOps, Agile, Scrum, Lean und Kanban sind alle auf ihre eigene Art und Weise in der Lage, einer Organisation Mehrwert zu bieten.

Welchen **eindeutigen** Wert bietet DevOps einer Organisation?

- A) DevOps legt einen nachhaltigen Entwicklungsrhythmus fest und ermöglicht so regelmäßige Feedbackschleifen seitens der Kunden.
- B) DevOps sorgt dafür, dass die Entwicklungs- und Betriebsteams das gleiche Taskboard zur Koordinierung der Aufgaben nutzen.
- C) DevOps legt Wert auf kontinuierliche Releases, um dem Kunden in einem effizienten Prozess schnell Wert zu liefern.
- D) DevOps stellt neue Services Just-in-Time bereit und legt gleichzeitig Wert auf Effizienz, indem es verschwenderische Arbeit eliminiert.

**19 / 40**

DevOps zielt darauf ab, die Kundenzufriedenheit und Kundenerfahrung zu verbessern, indem es den Fokus auf die Wertschöpfung legt.

Wie wird dies **in erster Linie** bewerkstelligt?

- A) Durch Automatisieren von Testen und Deployment (Bereitstellung)
- B) Durch häufigere Lieferung von Produkten
- C) Durch Entwicklung gut funktionierender Produkte
- D) Durch Einsatz hochkompetenter Entwickler

**20 / 40**

Viele DevOps-Teams setzen bei der Steuerung ihres Entwicklungsprozesses auf Visualisierung, beispielsweise in Form einer Kartenwand. Kartenwände erhöhen die visuelle Kontrolle des Teams.

Inwiefern unterstützt Visualisierung die Praktiken von DevOps?

- A) Sie heitern die Büroumgebung auf und steigern so Produktivität und Mitarbeiterzufriedenheit.
- B) Sie bieten den Führungskräften die Möglichkeit, das Team für seine Aktivitäten zur Verantwortung zu ziehen und die Arbeit aufzuteilen.
- C) Sie helfen dem Management, das Team und die zu erledigende Arbeit in den Griff zu bekommen.
- D) Sie verdeutlichen den Arbeitsfortschritt und wirken sich dadurch positiv auf die Kultur und die Leistung des Teams aus.

## 21 / 40

Das Entwicklungsteam kann innerhalb der geforderten Zeit neue Features erstellen. Häufig werden diese aber nicht pünktlich geliefert. Eine Untersuchung fördert folgende Probleme zutage:

- Es treten wiederholt Regressionsfehler auf.
- Obwohl Zeit für die Fehlerbehebung aufgewendet wird, beschwerten sich die Kunden immer wieder über Fehler in der Produktionsumgebung.
- Wird ein neues Feature gefordert, gerät das Entwicklungsteam stark unter Stress.

Das zugrundeliegende Problem ist, dass das Team nicht über eine effektive Teststrategie verfügt.

Was sollte das Team unternehmen, um dies zu beheben?

- A) Das Team sollte alle Tests und die Commit-Phase automatisieren und durch Schulungen dafür sorgen, dass die Teammitglieder den Prozess Continuous Integration (kontinuierlichen Integration) besser verstehen.
- B) Das Team sollte alle Unit-Tests, Komponententests und Integrationstests automatisieren und die Zusammenarbeit zwischen den Testern und dem Entwicklungsteam verbessern.
- C) Das Team sollte die Prozesse für Testen und Deployment (Bereitstellung) automatisieren und bei Bedarf ein Hardware-Upgrade durchführen oder das Konfigurationsmanagement des Systems aktualisieren.
- D) Das Team sollte die Zusammenarbeit zwischen dem Entwicklungs- und dem Betriebsteam durch mehr Monitoring und Protokollierung sowie die Nutzung von Virtualisierung steigern.
- E) Das Team sollte sicherstellen, dass das Entwicklungsteam den Deployment-Prozess versteht und die Zusammenarbeit mit dem Betriebsteam verstärken.

## 22 / 40

Zu DevOps gehört es auch, sicherzustellen, dass relevante Informationen für alle Entwickler visualisiert werden.

Was sind die **wichtigsten** Arten von Informationen, die für die Entwickler visualisiert werden sollten?

- A) Das tägliche Kundenfeedback einschließlich Beschwerden und Anfragen von Features, damit die Entwickler schnell scheitern können
- B) Die Produktivität der einzelnen Entwickler einschließlich der von ihnen geleisteten Arbeitsstunden, damit die Entwickler miteinander konkurrieren können
- C) Den Fortschritt des Gesamtprojekts einschließlich der Budget-Informationen, damit die Entwickler strategische Entscheidungen treffen können
- D) Daten zur Produktqualität einschließlich der Ausfall- und Fehlerraten, damit die Entwickler fundierte Entscheidungen treffen können

**23 / 40**

Ein Unternehmen hat sich damit einverstanden erklärt, nach DevOps zu arbeiten. Das Unternehmen sucht nach einer Möglichkeit, mit der es den Umfang aller seiner Projekte managen kann, die zu dieser Arbeitsweise passen.

Welche Möglichkeit eignet sich **am besten**, um den Umfang eines DevOps-Projekts zu managen?

- A) Eine Iterationsplanung durchführen, um sich am Anfang jeder Iteration darauf zu einigen, an welchen Items gearbeitet wird
- B) Eine vollständige Release-Planung für Features erstellen, die den Kunden vorab darüber informiert, was im Release enthalten sein wird
- C) Das Minimum Viable Product (MVP) festlegen und so oft wie nötig überprüfen
- D) Die Prioritäten der Arbeit vor jeder Iteration festlegen und den Teammitgliedern beim Aussuchen der Aufgaben helfen

**24 / 40**

Single-Piece-Flow bedeutet in der Software-Entwicklung, dass man sich auf ein einziges Feature konzentriert und kontinuierlich an diesem arbeitet, bis es fertig ist.

Warum ist Single-Piece-Flow bei DevOps wichtig?

- A) Es ermöglicht dem Team, in einem nachhaltigen und planbaren Tempo mit konstanter Geschwindigkeit zu arbeiten.
- B) Es hilft dem Team bei der Wertschöpfung, weil es Features mit hoher Priorität dann liefert, wenn sie gebraucht werden.
- C) Es stärkt die gemeinsame Verantwortung aller Teammitglieder bei der Auswahl der Aufgaben.
- D) Es reduziert die Engpässe, weil es die Zahl der Teammitglieder begrenzt, die an einem Feature arbeiten.

25 / 40

Tool Company arbeitet an der Entwicklung einer E-Commerce-Website. Das Projekt ist von größter Wichtigkeit. Die Kunden sind unzufrieden und beschweren sich, dass die E-Commerce-Website nicht wie erwartet funktioniert.

Führende Manager des Unternehmens argumentieren, dass ein Agiles Vorgehen, das Risiko noch steigern würde, während erfahren IT-Manager sagen, dass dies die einzige Möglichkeit sei, um schnell Wert zu liefern. Die IT-Manager möchten Agile mit DevOps kombinieren. Sie wünschen sich, häufige Releases und Input.

Das Unternehmen Tool Company beauftragt einen Berater, um sicher zu gehen, dass es das Richtige tut.

Welchen Rat sollte der Berater dem Unternehmen geben?

- A) Arbeiten Sie nicht Agil oder stellen Sie nicht auf DevOps um, da sich die Kunden sowieso schon beschweren, dass die E-Commerce-Website nicht wie erwartet funktioniert.
- B) Arbeiten Sie Agil, ergänzen Sie DevOps-Praktiken, geben Sie Ihren Entwicklern den Freiraum, Deployments (Bereitstellungen) ohne externe Genehmigung durchzuführen sobald sie können und ständig Kundenfeedback einzuholen.
- C) Nutzen Sie die Methoden Agile und DevOps zur Erstellung der E-Commerce Website, aber ergänzen Sie manuelle Benutzerabnahmetests und einen Prozess mit Change Approval Board (CAB) vor dem Release.
- D) Nutzen Sie die Methoden Agile und DevOps im Rahmen der Softwareentwicklung, aber geben Sie nur fertige Produkte an die Kunden frei, um Beschwerden und Unzufriedenheit seitens der Kunden zu vermeiden.

## 26 / 40

Ein Team hat gut zusammengearbeitet, aber die Kunden beschwerten sich trotzdem noch. Das Team würde gerne mehr experimentieren und DevOps einführen.

Das Management erlaubt dem Team zu experimentieren und seine Arbeitsweise zu ändern, vorausgesetzt die Risiken sind annehmbar. Das Unternehmen verfolgt die Politik, Risiken möglichst auf ein Mindestmaß zu reduzieren. Das Team diskutiert das Problem.

- Laut **Hichem** sollte das Team aktuell weder experimentieren noch seine Arbeitsweise ändern, da dies bei den Kunden zu noch größerer Unzufriedenheit führen könnte. Das Team sollte stattdessen ein Formular für Kundenfeedback erstellen.

- **Kira** sagt, dass es immer gut sei, Veränderungen in einem großen Release zu bündeln, um Störungen und Ausfälle in der Betriebsumgebung auf ein Mindestmaß zu beschränken. Ein umfangreiches Release sollte eine höhere Qualität des Produkts in der Produktionsumgebung sicherstellen.

- **Nazim** führt an, dass das Team erst auf die DevOps-Methode umstellen könne, sobald es gute Leistung erbringt und sich die Zahl der Kundenbeschwerden verringert hat. Zusätzliche manuelle Tests sollten dazu beitragen können, die Zahl der Beschwerden zu reduzieren.

- **Rose** sagt, dass es zwar unlogisch klingt, dass aber die Arbeitsweise nach der DevOps-Methode, bei der man viel experimentiert, das Risiko am stärksten senkt. Dies sei die beste Wahl, vorausgesetzt das Team habe Tests und Versionskontrolle gut automatisiert.

Wer unterbreitet den Vorschlag, der in diesem Szenario **am besten** funktioniert?

- A) Hichem
- B) Kira
- C) Nazim
- D) Rose



**27 / 40**

Ein Web-Shop spezialisiert sich auf Bekleidung aus den 1970er und 1980er Jahren. Der Web-Shop ist sehr erfolgreich. Sein aktueller Kundenstamm setzt sich zusammen aus Privatpersonen, die historische Kostüme suchen, und Kostümverleihern. Das Unternehmen möchte expandieren.

Ein Team diskutiert verschiedene Wachstumsoptionen und entscheidet sich dann für die hypothesengetriebene Entwicklung. Eine mögliche Hypothese für diesen Webshop ist, dass der Shop mehr Kunden anziehen wird, wenn sein Inhalt auch in einer mobilen App und nicht nur auf der Website präsentiert wird.

Was sollte unternommen werden?

- A)** - Der Kunde definiert eine Hypothese und das Team legt fest, welcher Test beziehungsweise welches Experiment durchgeführt werden sollte.
  - Im nächsten Schritt baut das Team ein Minimum Viable Product (MVP), das die formulierte Hypothese unterstützt und testet die Reaktionen der Kunden.
  - Nach einer kurzen Feedbackrunde mit den Kunden wird das MVP überarbeitet, um maximalen Kundenwert sicherzustellen.
- B)** - Der Kunde entwickelt auf der Grundlage seiner Bedürfnisse eine Hypothese und das Team spricht mit den Kunden.
  - Im nächsten Schritt baut und testet das Team einen Prototyp und ermittelt das finale MVP.
  - Der Prototyp wird zum MVP umgebaut und dem Kunden präsentiert, damit dieser die Hypothese testet.
- C)** - Das Team erhebt Informationen vom Kunden und formuliert auf der Grundlage dieser Informationen eine Hypothese.
  - Danach bestimmt das Team, welches Experiment oder welcher Test durchgeführt werden kann, um die Hypothese zu belegen oder zu widerlegen.
  - Das Experiment wird durchgeführt und analysiert und der Kunde dann informiert und um mehr Feedback gebeten.
- D)** - Das Team erhebt Informationen von Stakeholdern und formuliert auf dieser Grundlage alternative Hypothesen.
  - Danach baut und testet das Team ein MVP und präsentiert dieses dem oberen Management der Organisation.
  - Die finale Hypothese wird dann auf der Grundlage der Ergebnisse aus dem Akzeptanztest und der Reaktion des Managements gebildet.

**28 / 40**

Was zählt **nicht** zu den wahrscheinlichen Folgen der Praktiken von Lean Management?

- A)** Die Organisation erhält eine generativere Kultur.
- B)** Die Leistung der Software-Lieferung wird verbessert.
- C)** Die Teammitglieder sind weniger Burnout gefährdet.
- D)** Die Manager achten stärker auf Gewinne.

**29 / 40**

Ein Team hat Schwierigkeiten bei der Identifizierung von Problemen und beim Troubleshooting für Anwendungen, die in seiner Infrastruktur bereitgestellt werden. Dies verursacht viele Störungen und Ausfälle und ist daher ein Problem. Die Kunden rufen beim Service Desk häufig an und beschweren sich. Der Service Desk muss die vielen Fragen dann an das Team weiterleiten, sodass dieses unter Zeitdruck gerät.

Das Team hat zwar seinen Workflow dokumentiert und hält die Prozesse ein, hat aber keine Idee, was die Ursache für das Problem sein könnte. Das Team arbeitet mit einem Kanban-Board und scheint Flow erzielt zu haben.

Welche Lösung hilft langfristig **am besten** die Zahl der Probleme zu reduzieren?

- A) Einen Berater anstellen, der den Workflow neu abbildet und neue Prozesse auf dem Kanban-Board des Teams einrichtet
- B) Wertvolle Zeit gewinnen, indem man den Service Desk anweist, keine Fragen mehr an das Team weiterzuleiten
- C) Messen, wie viele Probleme jedes einzelne Teammitglied hat und einen Wettbewerb ausloben, um Verbesserung zu erzielen
- D) Eine Monitoringsstrategie einrichten, indem man Daten speichert, ein Dashboard erstellt und Benachrichtigungen einrichtet

**30 / 40**

Ein Team hat Schwierigkeiten, die Arbeit rechtzeitig fertigzustellen. Das Team hat Probleme, die Produkte pünktlich an die Kunden zu liefern.

Inwiefern helfen Work-in-Progress-Begrenzungen (WIP-Limits) diesem Team, Flow zu etablieren?

- A) Sie verändern die Rollen aller involvierten Stakeholder und Teammitglieder.
- B) Sie unterstützen die Identifizierung von Engpässen, die dem Team helfen, die Arbeit zu verbessern und zu optimieren.
- C) Sie verbessern das Lebenszyklusmanagement von Anwendungen und Kommunikationstools (Hilfsmitteln).
- D) Sie begrenzen die Anzahl der Tools und Anwendungen, die das Team zum Arbeiten benötigt.

**31 / 40**

Ein Team arbeitet an mehreren Projekten, die laut Management alle die gleiche Priorität haben. Das Team steht daher unter Druck, täglich an allen Projekten arbeiten und wöchentliche Fortschrittsberichte für alle Projekte abgeben zu müssen.

Das Team braucht einen Weg, um maximale Teameffizienz und qualitativ hochwertige Ergebnisse sicherzustellen. Das Team möchte den Kunden und dem Unternehmen maximalen Wert liefern.

Was sollte das Team unternehmen?

- A) Ein Teammitglied zur Verbindungsperson ernennen, damit sich die Manager nicht direkt an das Team wenden. Der Nachteil ist, dass das Team in diesem Fall kein direktes Kundenfeedback mehr erhält.
- B) So lange das Team alle zufriedenstellen kann, sollte es so weiter arbeiten wie bisher. Der Nachteil ist, dass manche Teammitglieder häufig die Aufgaben wechseln müssen.
- C) Work-in-Progress (WIP) begrenzen und eine Visualisierungstechnik für das Team einführen. Der Nachteil ist, dass das Team sich an die neue Arbeitsweise anpassen muss.
- D) Das Team in kleinere Unterteams mit zwei bis drei Teammitgliedern aufteilen, die separat an den einzelnen Projekten arbeiten. Der Nachteil ist, dass nicht alle Teams fähig sind, die Arbeit in ihrem Team fertigzustellen.

**32 / 40**

BigBlue Software ist ein kleines Unternehmen, das für andere Unternehmen Software erstellt. Das Unternehmen verfügt über ein Entwicklungsteam und ein Betriebsteam. Die Teams haben jeweils eigene Manager, arbeiten aber gut zusammen. Da das Unternehmen klein und der Geschäftsführer ehrgeizig ist, stehen die Teams manchmal unter Zeitdruck.

Unter diesem Zeitdruck machen beide Teams Fehler, die bereits zu Software-Bugs und Ausfällen geführt haben. Die Teams haben in diesen Fällen mit vereinten Kräften daran gearbeitet, die Ausfälle zu beheben oder ein Problem schnellstmöglich zu lösen. Die Manager sind stolz auf ihre Teammitglieder, weil diese ihre Fehler offen kommunizieren, gemeinsam daran arbeiten, diese zu lösen und Kundenwert liefern.

Würde diese Organisation von einer Veränderung ihrer Unternehmenskultur nach Westrum profitieren?

- A) Ja, die Teamstruktur mit einem Manager zeigt, dass die Unternehmenskultur bürokratisch ist und geändert werden sollte.
- B) Ja, die Fehler, die unter Druck gemacht werden, zeigen, dass die Unternehmenskultur pathologisch ist und geändert werden sollte.
- C) Nein, die Zusammenarbeit und der offene Umgang mit Fehlern zeigen, dass die Unternehmenskultur generativ ist und nicht geändert werden sollte.
- D) Nein, die Forderungen nach Kundenwert zeigen, dass die Unternehmenskultur marktorientiert ist und nicht geändert werden sollte.

**33 / 40**

Ein Teammitglied sagt, dass das Unternehmen organisationales Lernen benötigt, das manchmal auch als generative Kultur bezeichnet wird. Organisationales Lernen ermöglicht es, den Kunden zu helfen, sorgt für Qualität und schafft einen Wettbewerbsvorteil sowie eine engagierte Belegschaft.

Warum ist dies richtig?

- A) Weil organisationales Lernen bedeutet, dass man Fehler identifiziert und einzelne Teammitglieder dafür verantwortlich macht, um aus diesen Fehlern zu lernen
- B) Weil organisationales Lernen bedeutet, dass man das Team ständig in den neuesten Methoden von Agile und Lean schult, um so für mehr Flow zu sorgen
- C) Weil organisationales Lernen ein Denken voraussetzt, das den Mitarbeitern das Gefühl vermittelt, dass sie unterstützt werden und ihr Urteilsvermögen geschätzt wird
- D) Weil organisationales Lernen eine Kultur des Forschens fordert, die automatisch Hypothesen erstellt, um so fundierte Entscheidungen zu treffen

**34 / 40**

Ein Personalteam wird gebeten, eine neue Richtlinie für die Anstellung neuer DevOps-Teammitglieder zu verfassen.

Was ist die **beste** Richtlinie für die Anstellung von DevOps-Teammitgliedern?

- A) Anonymisierte Bewerbungen zu ermöglichen und Bewerber zu bevorzugen, die für Diversität im Team sorgen, weil dies die Ergebnisse des Teams verbessert
- B) Den Managern die Möglichkeit zu geben, nachzuforschen, welche Mitarbeiter in einem anderen Unternehmen die beste Leistung erbringen und dann zu versuchen, diese Mitarbeiter abzuwerben, um die besten Mitarbeiter für das Team zu gewinnen
- C) Nur Bewerber zu berücksichtigen, die das Gefühl haben, dass sie nicht voll und ganz zu der Organisation oder den anderen Teammitgliedern passen, weil dies Diversität am stärksten fördert
- D) Den Teammitgliedern zu erlauben, das Teammitglied auszuwählen, mit dem sie sich am stärksten identifizieren, weil dies für Harmonie und Gleichheit im Team sorgt

### 35 / 40

Eine Organisation hat seine Managementpraktiken auf Lean-Leadership umgestellt. Das ist eine große Veränderung gegenüber der früheren Arbeitsweise der Organisation. Früher hat das Management die Kollegen regelmäßig öffentlich für ihre Fehler bloßgestellt.

Eines der Teams ist ein verteiltes Team. Das Team ist mit dem neuen Führungsstil zwar viel glücklicher, aber seine Kommunikation ist nicht optimal. Die Teammitglieder diskutieren nicht über ihre Software-Bugs und arbeiten auch nicht zusammen, um Probleme schnell zu beheben.

Der Vorgesetzte des Teams hat deshalb die Unternehmensversion eines Chat-Tools (Chat-Hilfsmittels) eingeführt, das die meisten Teammitglieder bereits privat nutzen. Aber auch diese Maßnahme scheint nicht den gewünschten Erfolg zu haben.

Was passiert in diesem Team und wie sollte der Manager das lösen?

- A) - Die Teammitglieder verfügen noch nicht über die erforderliche Kompetenz im Umgang mit dem Chat-Tool.  
- Der Manager sollte interne oder externe Schulungen ermöglichen, um dieses Problem zu lösen.
- B) - Die Teammitglieder sind noch in der Umstellung vom früheren auf den neuen Führungsstil.  
- Hierbei handelt es sich um einen vorübergehenden Prozess, in den der Manager nicht eingreifen muss.
- C) - Die Teammitglieder vertrauen dem neuen Führungsstil noch nicht.  
- Der Manager muss hier überlegt vorgehen und seinen Stolz über das Team zum Ausdruck bringen.

### 36 / 40

Was ist der **wichtigste** unmittelbare Vorteil, den man erhält, wenn man dafür sorgt, dass die Teammitglieder eine sinnstiftende Arbeit haben?

- A) Es trägt dazu bei, die Anzahl von Burnouts zu verringern und Mitarbeiterloyalität und -engagement zu steigern.
- B) Es trägt dazu bei, alle DevOps-Projekte des Teams pünktlich und im Rahmen des Budgets zu liefern.
- C) Es trägt dazu bei, die Überzeugung des Teams zu stärken, dass es die DevOps-Arbeit bewältigen kann.
- D) Es trägt dazu bei, die Teammitglieder an die Organisation zu binden, weil diese ihren Job nicht aufgeben wollen.

**37 / 40**

Ein Unternehmen ist in den letzten beiden Jahren exponentiell gewachsen. Das Entwicklungsteam steht infolgedessen unter Zeitdruck. Alle Teammitglieder fühlen sich urlaubsreif und manche denken sogar an Kündigung. Die Krankheitstage im Team sind gegenüber den Vorjahren gestiegen.

Der Teammanager hat mehrere neue Teammitglieder eingestellt, aber die Arbeitsbelastung wird immer noch als hoch empfunden, obwohl anscheinend alle Fristen pünktlich eingehalten werden. Um das Problem zu lösen, möchte sich der Teammanager direkter mit den Beschwerden zu Burnout und Stress befassen.

Welche Maßnahmen eignen sich dazu **am besten**?

- A) - Unterauftragnehmer beauftragen, um die höhere Arbeitsbelastung in Spitzenzeiten zu bewältigen  
- Den Teammitgliedern Zugriff auf professionelle psychologische Hilfsprogramme anzubieten, die vom Unternehmen finanziert werden
- B) - Allen Teammitgliedern einen Monat Urlaub geben, um einen Neuanfang zu gewährleisten, wenn sie wieder an ihren Arbeitsplatz zurückkehren  
- Die Manager bitten, bevor das Team wieder an seinen Arbeitsplatz zurückkehrt, alle Faktoren zu identifizieren, die möglicherweise zur Burnout-Problematik beigetragen haben
- C) - Dafür sorgen, dass die Teammitglieder alle Aufgaben im Prozess besprechen, um herauszufinden, welche Aufgaben sinnstiftend sind  
- Nicht sinnstiftende Aufgaben sollten gestrichen und sichergestellt werden, dass das Team die meisten Entscheidungen selbstständig fällen kann
- D) - Einen kompetenteren Manager einstellen, der weiß, wie man die Work-Life-Balance der Teammitglieder sicherstellt  
- Den Teammitgliedern mehr Zeit geben, damit sie sich in ihre neue Umgebung eingewöhnen und ein neues Gleichgewicht finden können

**38 / 40**

Transformationale Führung zählt angeblich zu den Aspekten von DevOps, die am häufigsten übersehen werden.

Warum ist transformationale Führung für den unternehmensweiten Erfolg von DevOps wichtig?

- A) Weil man mit Commitment des Top-Managements Prozesse verändern und so potenziellem Widerstand entgegenwirken kann
- B) Weil DevOps scheitern wird, wenn nicht alle Mitarbeiter, auch das Top-Management, DevOps gut kennen und verstehen
- C) Weil die für Continuous Delivery (kontinuierliche Lieferung) notwendigen Änderungen vom Top-Management angeordnet werden müssen
- D) Weil das Top-Management Vertrauen in seine Mitarbeiter haben und diese ermutigen muss, zu experimentieren, ohne sich vor Strafen fürchten zu müssen

**39 / 40**

Das kontinuierliche Monitoring des DevOps-Reifegrads in einer Organisation ist eine entscheidende Fähigkeit. Sie ermöglicht die fachübergreifende Zusammenarbeit in der IT und verringert oder beseitigt unproduktive Verhaltensweisen, die man in Organisationen mit funktionalen Silos noch viel zu häufig vorfindet. Das kontinuierliche Monitoring unterscheidet sich vom herkömmlichen Monitoring.

Was ist die **wichtigste** Anforderung an das kontinuierliche Monitoring?

- A) Eine Komponente vollständig automatisiert auf den gewünschten Zustand zu bringen
- B) Die Zusammenarbeit von Entwicklung und Betrieb in jedem Wertstrom zu erreichen
- C) Die Organisation auf eine Linie zu bringen und die Produktionsumgebung in den Vordergrund zu rücken
- D) Eine gesamtheitliche Sicht auf die komplexe Anwendungsumgebung der Organisation zu haben

**40 / 40**

Betrachtet man den DevOps-Reifegrad einer Organisation, so ist eine mehrdimensionale Perspektive der Organisation, ihrer Fähigkeiten und ihres Reifegrads wichtig. Die kontinuierliche Bewertungsarchitektur von DevOps lässt sich in Form eines Würfels darstellen.

Was repräsentieren die Seiten des „DevOps-Würfels“?

- A) Flow, Feedback, Lernen und Experimentieren, Governance, Deployment (Bereitstellung) und Qualitätssicherung
- B) Monitoring, Information, Kommunikation, Bewertung, Kontrolle und Aktivitäten
- C) Menschen, Prozesse, Partner, Technologie, Qualitätssicherung und Kultur
- D) Anforderungen, Prozesse, Ressourcen, Organisation, Reifegrad und Fähigkeit

# Antwortschlüssel

1 / 40

Welche Lösung ist **am effektivsten**, um Daten automatisiert zu migrieren?

- A) Erstellen von kleinere Datensätze, damit die Migration überschaubar bleibt
  - B) Entwickeln eines Rollback-Verfahrens, falls die Migration fehlschlägt
  - C) Sicherstellen, dass die Skripte vor der Datenmigration ordnungsgemäß getestet werden
  - D) Einrichten einer Datenbankversionierung und Befolgen einer strengen Versionskontrolle
- A) Falsch. Die Implementierung von Versionskontrolle ist für die Automatisierung wichtiger. Die Größe des Datensatzes sollte keine Rolle spielen.
- B) Falsch. Hier geht es um die Wiederherstellung, falls die Migration fehlschlägt. Für die Automatisierung ist es jedoch wichtiger, Versionskontrolle zu implementieren, denn diese verhindert, dass die Migration fehlschlägt.
- C) Falsch. Obwohl Testen wichtig ist, ist die Implementierung von Versionskontrolle ist für die Automatisierung wichtiger.
- D) Richtig. Die beste Lösung für die Automatisierung der Datenmigration ist die Versionierung der Datenbank. (Literatur: B, Kapitel 12)

2 / 40

Ein Team hat Continuous Deployment (kontinuierliche Bereitstellung) implementiert und alles wird über Versionskontrolle verwaltet. Das Team hält dies für notwendig, um Fehler schnell identifizieren und bei Bedarf jederzeit ein Rollback durchführen zu können.

Der Teamleiter führt jedoch an, dass es nicht empfehlenswert sei, auch die Binärausgaben über die Versionskontrolle zu verwalten.

Warum sagt der Teamleiter das?

- A) Weil eine erneute Kompilierung fester Bestandteil eines jeden normalen Build-Prozesses ist
  - B) Weil Binärausgaben große Dateien sind, die jeden Build verändern und aus dem Quellcode wiederhergestellt werden können
  - C) Weil an Binärausgaben mehrere Teammitglieder arbeiten, was die Versionskontrolle erschwert
  - D) Weil Binärausgaben Input für die Compiler sind und nicht richtig in der Versionskontrolle verwaltet werden können
- A) Falsch. Die erneute Kompilierung würde zwar Binärausgaben schaffen, aber eine Rekompilierung als festen Bestandteil des normalen Build-Prozesses durchzuführen, ist nicht ratsam.
- B) Richtig. Binärausgaben sind große Dateien. Sie werden für jeden Check-in, der kompiliert wird und die automatisierten Tests besteht, erneut erstellt und können durch erneutes Ausführen des in der Versionskontrolle verwalteten Build-Skripts wiederhergestellt werden. Daher ist es nicht notwendig, diese großen Dateien ebenfalls zu speichern. (Literatur: B, Kapitel 2)
- C) Falsch. Die Versionskontrolle für Binärausgaben ist nicht schwierig, nur unpraktisch.
- D) Falsch. Binärausgaben dienen nicht als Input für die Compiler, sondern sind der Output der Compiler. Sie könnten zwar in der Versionskontrolle verwaltet werden, dies ist aber unnötig.



### 3 / 40

Ein Team arbeitet an der Erstellung einer Deployment Pipeline. Das Team hat für seinen Prozess erfolgreich Single-Piece-Flow erzeugt und verfügt über einige grundlegende, automatisierte Build- und Deployment Prozesse. Die Unit- und Akzeptanztests werden jedoch nach wie vor manuell ausgeführt. Das Team führt aktuell jeden Monat ein Release durch.

Die Mitglieder diskutieren, was das Team als Nächstes tun sollte, um seine Deployment Pipeline zu verbessern.

- **Aki** sagt: „Wir sollten zuerst unsere Releasefrequenz erhöhen. Die fehlenden Elemente ergeben sich dann aus den Problemen, auf die wir dabei stoßen.“
- **Em** sagt: „Wir sollten die restlichen Tests automatisieren. Sobald wir die Unit- und Akzeptanztests automatisiert haben, können wir mit der Releaseautomatisierung beginnen.“
- **Ken** sagt: „Die Deployment Pipeline ist Bestandteil der Wertschöpfungskette der Organisation. Wir sollten daher zuerst eine Work-in-Progress-Begrenzung (WIP-Limit) implementieren, bevor wir irgendetwas anderes unternehmen.“
- **Mart** sagt: „Es läuft aktuell gut und wir haben Single-Piece-Flow erzeugt. Aktuell gibt es keinerlei Grund, irgendwelche Maßnahmen zu ergreifen. Wir sollten warten, bis die Organisation einen höheren Reifegrad erzielt.“

Wer macht in diesem Szenario den **besten** Vorschlag?

- A) Aki
  - B) Em
  - C) Ken
  - D) Mart
- 
- A) Falsch. Die Zahl der Releases zu erhöhen, bevor der Release-Prozess automatisiert oder Single-Piece-Flow erzeugt ist, ergibt keinen Sinn.
  - B) Richtig. Die erste Priorität gilt der Erzeugung des Single-Piece-Flow der Pipeline. Sobald dies, wie in diesem Szenario, erfolgt ist, sollte die Automatisierung der Releases vollständig automatisiert werden. (Literatur: B, Chapter 5)
  - C) Falsch. Da das Team bereits mit Single-Piece-Flow arbeitet, verfügt es wahrscheinlich bereits über eine Work-in-Progress-Begrenzung (WIP-Limit) beziehungsweise besteht keine Notwendigkeit, eine solche zu implementieren. Die Automatisierung dagegen bringt die Deployment Pipeline weiter.
  - D) Falsch. Nichts zu tun und einfach zu warten, bis die restliche Organisation einen höheren Reifegrad erreicht, ist nicht wirklich hilfreich. Die Automatisierung dagegen bringt die Deployment Pipeline weiter.

4 / 40

Ein Team muss für ein neues Produkt eine Deployment Pipeline entwickeln. Das Team diskutiert über die Commit-Phase der Deployment Pipeline und arbeitet in Richtung Continuous Integration (kontinuierliche Integration).

Ein Teammitglied sagt: „Die Definition of Done (DoD, Definition von 'Fertiggestellt') sollte vor oder spätestens in der Commit-Phase festgelegt werden. Entspricht der Code beim Einchecken nicht der DoD, so sollte die Arbeit daran eingestellt werden.“

Ist diese Aussage richtig?

- A) Ja, weil das Teammitglied seine Arbeit nicht korrekt ausführt und umgehend ermahnt werden muss.
  - B) Ja, weil Arbeit, die nicht der DoD entspricht, keinen Mehrwert bietet und daher nicht eingecheckt werden sollte.
  - C) Nein, weil die DoD in den Besprechungen mit dem Kunden festgelegt wird und vor der Commit-Phase noch gar nicht vorliegt.
  - D) Nein, weil für die Arbeit in der Deployment Pipeline ein steter Flow sichergestellt sein muss und man die Arbeit nie plötzlich einstellen werden darf.
- 
- A) Falsch. Die Arbeit sollte gestoppt werden, weil sie keinen Mehrwert bietet und deshalb nicht eingecheckt werden sollte. Das bedeutet jedoch nicht zwangsläufig, dass das Teammitglied seine Arbeit nicht ordnungsgemäß ausführt oder gar ermahnt werden sollte.
  - B) Richtig. Entspricht Arbeit nicht der DoD, so bietet sie dem Kunden nicht genügend Wert, um in die Deployment Pipeline eingecheckt zu werden. Sie würde im Single-Piece-Flow den Flow von wertvollere Arbeit verzögern. (Literatur: B, Kapitel 3)
  - C) Falsch. Die DoD ist eines der ersten Dinge, die in einem Projekt vereinbart werden. Sie wird nicht im Rahmen von Kundenbesprechungen festgelegt. Die DoD sollte vorliegen, wenn man mit der Arbeit an der Entwicklung des Codes beginnt.
  - D) Falsch. Die Deployment Pipeline zu stoppen, ist dann sinnvoll, wenn etwas an dem eingecheckten Code nicht stimmt. Arbeit, die nicht der DoD entspricht, bietet keinen Mehrwert und sollte deshalb gar nicht erst in die Pipeline gelangen.

5 / 40

Nachdem es Upgrades der Anwendungssoftware bereitgestellt hat, erlebt ein Team viele Fehler und Ausfälle der Anwendung, aber auch der Hardware. Der Versuch, einen früheren, bekanntermaßen funktionierenden Systemzustand wiederherzustellen, dauerte sehr lange. Die Ausfälle führten zu längeren Betriebsunterbrechungen bei kritischen Anwendungen. Das Management hat das Team gebeten, das Risiko von Betriebsunterbrechungen für die nächsten Deployments (Bereitstellungen) zu senken.

Das Team diskutiert, wie dieses Problem zu lösen sei.

Was sollte das Team als **Erstes** unternehmen?

- A) Vollständig automatisieren alle Tests ebenso wie die Build-, Deployment- und Release-Prozesse
  - B) Sicherstellen, dass alles in der Versionskontrolle verwaltet wird, um eine schnelle Wiederherstellung zu ermöglichen
  - C) Sorgen für Kommunikation zwischen dem Entwicklungs- und dem Betriebsteam
  - D) Schulen dem Entwicklungsteam, damit es den Deployment-Prozess versteht
- 
- A) Falsch. Das ist zwar im Allgemeinen eine gute Idee, aber die Aufgabenstellung lautet hier, das Risiko von Betriebsunterbrechungen zu reduzieren. Sorgt man dafür, dass alles in der Versionskontrolle verwaltet wird, löst man das Problem, indem man ein Rollback auf einen früheren, bekanntermaßen funktionierenden Systemzustand durchführt, selbst wenn die restliche Arbeit dann manuell vorgenommen werden muss.
  - B) Richtig. Sorgt man dafür, dass alles in der Versionskontrolle verwaltet wird, löst man das Problem durch ein Rollback auf einen früheren, bekanntermaßen funktionierenden Systemzustand. (Literatur: B, Kapitel 1)
  - C) Falsch. Das ist zwar im Allgemeinen eine gute Idee, aber die Aufgabenstellung lautet in diesem Fall, das Risiko von Betriebsunterbrechungen zu reduzieren. Sorgt man dafür, dass alles in der Versionskontrolle verwaltet wird, löst man das Problem, indem man ein Rollback auf einen früheren, bekanntermaßen funktionierenden Systemzustand durchführt, selbst wenn die Teams nicht miteinander reden.
  - D) Falsch. Eine Schulung kann dem Team zwar helfen, die Zahl der Ausfälle zu reduzieren, verringert aber nicht das Risiko einer Betriebsunterbrechung im Falle eines Ausfalls. Entscheidend ist in diesem Fall die Versionskontrolle.

6 / 40

Eine Organisation plant die Implementierung von Continuous Integration (kontinuierlicher Integration). Das Entwicklungsteam der Organisation ist mit dieser neuen Arbeitsweise einverstanden und hat bereits Tools (Hilfsmittel) für die Automatisierung ausgewählt, die die Software von der Befehlszeile aus erstellen.

Welche andere Aktivität ist ebenfalls eine Voraussetzung für die kontinuierliche Integration?

- A) Die Beseitigung von absoluten Pfaden. Dies ist für die Integration der Libraries von Drittanbietern erforderlich.
  - B) Die Kontrolle des Zugriffs auf die Infrastruktur. Dies sorgt dafür, dass keiner unautorisierte Änderungen vornehmen kann.
  - C) Die Dokumentation von Wertströmen. Dies hilft bei der Identifizierung, in welchen Bereichen Verbesserungsbedarf besteht.
  - D) Die Einführung von Versionskontrolle. Denn diese zeigt, was wann, von wem und warum getan wurde.
- 
- A) Falsch. Dies ist keine spezifische Voraussetzung für die Einführung von kontinuierlicher Integration. Absolute Pfade sind zwar nicht wünschenswert, manchmal jedoch erforderlich, so zum Beispiel, wenn man Libraries von Drittanbietern integriert, die auf fest kodierte Pfade angewiesen sind.
  - B) Falsch. Dies ist zwar eine wichtige DevOps-Aktivität, aber keine spezifische Voraussetzung für die Einführung von kontinuierlicher Integration.
  - C) Falsch. Wertströmen sollten formuliert werden, aber dies ist keine spezifische Voraussetzung für die Einführung von kontinuierlicher Integration.
  - D) Richtig. Will man kontinuierliche Integration einführen, braucht man drei Dinge: Versionskontrolle, einen automatisierten Build und die Zustimmung des Teams. In diesem Szenario hat die Versionskontrolle gefehlt. (Literatur: B, Kapitel 3)

7 / 40

Ein Team diskutiert über seine Benutzerabnahmetests. Aktuell wendet das Team viel Zeit, Arbeit und Geld für manuelle Akzeptanztests auf. Das Management hat das Team gefragt, ob es hierfür eine bessere Alternative gibt. Das Management bevorzugt in der Regel die Lösung, die mit dem geringsten Risiko für Produktionsunterbrechungen verbunden ist.

Das Team kommt zu dem Schluss, dass es intern über die erforderlichen Fähigkeiten verfügt und alle Benutzerabnahmetests automatisieren kann. Dies würde jedoch etwas Zeit in Anspruch nehmen und mehr Kosten verursachen. Ein Teammitglied merkt an, dass ein Wettbewerber alle Akzeptanztests durch automatisierte Unit- und Komponententests ersetzt hat.

Sollte das Team den Benutzerabnahmetest in diesem Szenario automatisieren?

- A) Ja, weil Automatisierung auf lange Sicht günstiger ist als manuelles Testen und das Risiko für Produktionsunterbrechungen dadurch niedrig gehalten wird.
  - B) Ja, weil dadurch nur noch ein manueller Akzeptanztest pro Monat erforderlich ist, um das Risiko für Produktionsunterbrechungen niedrig zu halten.
  - C) Nein, weil manuelle Akzeptanztests sicherstellen, dass alle Fehler vor dem Release entdeckt werden, um das Risiko für Produktionsunterbrechungen niedrig zu halten.
  - D) Nein, weil Unit- und Komponententests die Akzeptanztests ersetzen und der Wettbewerber beweist, dass dies das Risiko für Produktionsunterbrechungen niedrig hält.
- 
- A) Richtig. Kann das Team die Akzeptanztests automatisieren, so nimmt dies Druck aus den Projekten, während die Fehler, die in den Anwendungsszenarien (User Scenarios) auftreten können, trotzdem gefunden werden. So bleibt das Risiko für Produktionsunterbrechungen niedrig. Außerdem ist Automatisierung auf lange Sicht immer günstiger als manuelle Arbeit. (Literatur: B, Kapitel 8)
  - B) Falsch. Das Team sollte die Akzeptanztests zwar automatisieren, aber das Ziel ist nicht, die Zahl der manuellen Tests zu verringern, sondern diese komplett zu ersetzen.
  - C) Falsch. Automatisierung ist auf lange Sicht immer günstiger als manuelle Arbeit und dabei nicht weniger exakt, vorausgesetzt die Tests werden gut implementiert.
  - D) Falsch. Unit- und Komponententests erfassen die gesamte Erfahrung der Anwendungsszenarien (User Scenarios) nicht präzise genug. Dadurch wächst das Risiko für Produktionsunterbrechungen unabhängig davon, was der Wettbewerber macht.

8 / 40

Ein Start-up hat eine App für Haustierbesitzer erstellt. Die App ermöglicht Kunden für einen bestimmten Tag einen Haustiersitter zu finden. Die Unternehmensinhaberin spricht mit einem wichtigen Kunden. Sie geht davon aus, dass das Unternehmen schnell wachsen wird.

Das Entwicklungsteam verfügt aktuell über einen manuellen Prozess für das Testen. Dies hat bislang sehr gut funktioniert.

Was sollte das Entwicklungsteam als Nächstes unternehmen?

- A) Neue Tests für das Betriebsteam erstellen, um das Deployment (Bereitstellung) zu automatisieren
  - B) Das manuelle Testen beibehalten und bei Bedarf neue manuelle Tests ergänzen
  - C) Warten bis klar ist, wie es für die neuen Kunden Mehrwert schaffen kann
  - D) Gemeinsam mit dem Betriebsteam an der Erstellung von automatisierten Tests arbeiten
- 
- A) Falsch. Die Teams sollten gemeinsam an der Erstellung der Tests arbeiten. Das Entwicklungsteam sollte keine Tests für das Betriebsteam erstellen.
  - B) Falsch. Manuelle Tests beizubehalten, wenn das Unternehmen wächst, ist äußerst riskant. Alle manuellen Aktivitäten sind fehleranfällig und lassen sich nur schwer skalieren.
  - C) Falsch. Es besteht keinerlei Notwendigkeit, auf neue Kunden zu warten. Viele der standardisierten, automatisierten Tests sollten auch ohne Kunden durchgeführt werden können.
  - D) Richtig. Die Teams sollten gemeinsam an der Erstellung automatisierter Tests arbeiten. Da automatisierte Tests skalierbar sind, trägt dies selbst bei schnellem Wachstum zu einer beständig hohen Testleistung bei. (Literatur: B, Kapitel 8)

9 / 40

Ein DevOps-Team nutzt die Daten in seinen Datenbanken für die Tests der Softwareentwicklung.

Was ist für dieses Team potenziell das **größte** Problem?

- A) Der Datenzugriff von Standorten außerhalb der Testdatenbank
  - B) Die Zugänglichkeit von Produktionsdaten aus dem Homeoffice
  - C) Die Verfügbarkeit repräsentativer Testdaten für das automatisierte Testen
  - D) Die Sicherheit der verfügbaren Sicherheitsvorkehrungen für die Testdaten
- 
- A) Falsch. Der Zugriff auf die Daten in den Datenbanken kann stets gleichzeitig erfolgen.
  - B) Falsch. Für die Tests der Softwareentwicklung werden niemals Produktionsdaten genutzt.
  - C) Richtig. Die Testdaten müssen für die Produktionsdaten repräsentativ sein. (Literatur: A, Kapitel 4)
  - D) Falsch. Beim Testen sind in der Regel keine Sicherheitsvorkehrungen für die Testdaten erforderlich.

10 / 40

Kamala ist als Analystin in der Informationssicherheit dafür zuständig, den aktuellen Software-Entwicklungsprozess eines Open-Finance-Unternehmens zu verbessern. Eines ihrer Hauptziele besteht darin, in der Phase der Code-Entwicklung das Thema Informationssicherheit stärker ins Bewusstsein zu rücken.

Kamalas Ratschlag lautet, dass die Teams einen „Shift Left“ zur Informationssicherheit durchführen sollten. Mit anderen Worten, Informationssicherheit sollte in den Softwareentwicklungszyklus integriert werden, von der Codeerstellung bis zum Betrieb.

Ihr Vorgesetzter Rob ist mit diesem Vorschlag nicht einverstanden. Er schlägt vor, einen Spezialisten einzustellen, der für Informationssicherheit in den Produkten sorgt, wenn diese für das Deployment (Bereitstellung) bereit sind. Rob argumentiert, dass die Entwickler keine Sicherheitsexperten sind, und sich auf die Entwicklung konzentrieren sollten.

Wer hat Recht?

- A) Kamala, weil eine frühere Integration der Informationssicherheit die Einhaltung der maßgeblichen Gesetze und Vorschriften sicherstellt. Dies macht die Erstellung sicherer Software langfristig günstiger.
  - B) Kamala, weil die Teammitglieder sich zu Sicherheitsexperten schulen lassen können. Dies spart die Lohnkosten für einen Sicherheitsexperten.
  - C) Rob, weil es am besten funktioniert, wenn man ausgehend von den Best Practices im Bereich der Informationssicherheit, eine Reihe von Richtlinien erstellt und dann einen Experten einstellt, der diese Richtlinien in den Produkten implementiert.
  - D) Rob, weil die Entwickler sich tatsächlich auf ihre Kernaufgabe, die Entwicklung von Software, konzentrieren können und nicht damit belastet werden sollten, auch Experten im Bereich der Sicherheit zu werden.
- 
- A) Richtig. Informationssicherheit ist ein Pflichtelement von DevOps und sollte Teil der Unternehmenskultur sein. Außerdem ist es wirtschaftlicher, bereits während der Entwicklung an die Sicherheit zu denken, als diese nachträglich einzubauen. (Literatur: A, Kapitel 4 und 6)
  - B) Falsch. Eine Schulung kann den Teams zwar helfen, das Thema zu durchdringen, reicht aber nicht aus, um eine Veränderung der Gesamtumgebung herbeizuführen. Einsparungen ergeben sich nicht aus den Lohnkosten für einen Sicherheitsexperten, sondern aus einer robusteren Software, die den Gesetzen und Vorschriften entspricht und nicht überarbeitet werden muss.
  - C) Falsch. Ein Top-Down-Ansatz wirkt sich möglicherweise nur wenig auf die Teams aus, die das Thema Informationssicherheit auf kultureller Ebene ansprechen müssen. Außerdem ist es wirtschaftlicher und besser, eine robuste Software zu erstellen, die den Gesetzen und Vorschriften entspricht und nicht überarbeitet werden muss.
  - D) Falsch. Informationssicherheit ist ein Pflichtelement von DevOps und sollte Teil der Unternehmenskultur sein. Die Erstellung robuster Software, die den Gesetzen und Vorschriften entspricht und nicht überarbeitet werden muss, ist daher wirtschaftlicher und besser.

11 / 40

Was ist der **größte** Vorteil von Continuous Delivery (kontinuierlicher Lieferung)?

- A) Sie ermöglicht transparente Konformität und Leistung der Softwareentwicklung.
  - B) Sie sorgt dafür, dass die Softwareentwicklungsteams die Corporate-Governance-Prozesse umsetzen.
  - C) Sie konzentriert sich auf die Time-to-Market (Markteinführungszeit) des Deployments (Bereitstellung) und eine verbesserte Geschäftsleistung.
  - D) Sie führt Automatisierung ein, um die Anforderung an Einhaltung und Governance von Software zu reduzieren.
- 
- A) Richtig. Der zentrale Grundsatz von Continuous Delivery lautet, sowohl Konformität als auch Leistung zu erzielen. (Literatur: B, Kapitel 15)
  - B) Falsch. Continuous Delivery konzentriert sich nicht nur auf eine Art von Governance, sondern sorgt für ein Gleichgewicht zwischen Corporate Governance und Business Governance.
  - C) Falsch. Die Time-to-Market ist zwar wichtig und lässt sich mit Hilfe von Continuous Delivery erzielen, der größte Vorteil besteht jedoch im Erreichen von Konformität und Leistung.
  - D) Falsch. Corporate Governance beziehungsweise die Einhaltung von Vorschriften werden durch Automatisierung nicht überflüssig.

12 / 40

Ein Team arbeitet an einer Anwendung und möchte Continuous Delivery (kontinuierliche Lieferung) einführen. Bei einer Besprechung zur Prozessverbesserung bespricht das Team, welche Praktik es nutzen sollte, um Continuous Delivery einzuführen.

Was sollte das Team tun?

- A) Das Team sollte Scrum als Managementpraktik kombiniert mit einer vierteljährlichen Release-Planung nutzen, um agiler zu werden.
  - B) Das Team sollte daran arbeiten, die Continuous Integration (kontinuierliche Integration) der Branches und das automatisierte Testen seiner Anwendung zu verbessern.
  - C) Das Team sollte an einer Kulturveränderung arbeiten und Agile Praktiken für das Change Management sowie Feedbackschleifen für das Berichtswesen nutzen.
  - D) Das Team sollte an der Implementierung von Kanban auf Teamebene arbeiten und gleichzeitig eine lose gekoppelte Architektur errichten.
- 
- A) Falsch. Scrum kann Teams zwar dabei unterstützen, agiler zu werden, ist aber für die Arbeit nach der DevOps-Methode nicht unbedingt erforderlich. Das Team sollte sich stattdessen auf Continuous Integration und Automatisierung konzentrieren.
  - B) Richtig. Continuous Integration, Versionskontrolle und Testroutinen sind Praktiken, die einem Team helfen, Möglichkeiten zur Verbesserung der Liefergeschwindigkeit und -qualität zu entdecken. (Literatur: B, Kapitel 3 und 5)
  - C) Falsch. Dies sind Praktiken für das organisationales Change Management und nicht für die Continuous Delivery. Das Team sollte sich stattdessen auf die Continuous Integration und Automatisierung konzentrieren.
  - D) Falsch. Lose gekoppelte Architekturen sind zwar nützlich, haben aber nicht unbedingt etwas mit DevOps zu tun. Das Team sollte sich stattdessen auf Continuous Integration und Automatisierung konzentrieren.



### 13 / 40

Ein Unternehmen hat seinen Entwicklungsprozess radikal von der Wasserfallmethode auf DevOps umgestellt. Das Team möchte die Zeit zwischen den Releases reduzieren, trifft dabei aber auf einige Probleme.

Die vom Team durchgeführte Prozessanalyse hat folgende Probleme aufgezeigt:

- 20 Prozent der Sprint-Dauer wird für das Portieren von Code aufgewendet, weil alle Entwickler an separaten Code-Branches arbeiten.
- Werden Probleme entdeckt, die den Build brechen, so muss ein Rollback der Produktionsumgebung durchgeführt werden.
- Das Team verfügt zwar über ein System zur Versionskontrolle, aber ein Monitoring der Key Performance Indicators (KPI, wichtigsten Leistungskennzahlen) findet nicht statt.
- Je höher die Zahl der Releases, desto mehr Kundenfeedback, einschließlich Beschwerden, erhält das Team.

Was hilft diesem Team **am besten** die Zeit zwischen den Releases zu reduzieren?

- A) Die Einführung von Trunk-basierter Entwicklung, um das Portieren von Code überflüssig zu machen
  - B) Das Reproduzieren von Testabbrüchen, um Rollbacks in der Produktionsumgebung zu vermeiden
  - C) Die Nutzung von Canary-Releases, um die Zahl der Kundenbeschwerden zu reduzieren
  - D) Der verstärkte Einsatz von Telemetrie für das Monitoring von KPI, die den Geschäftswert zeigen
- 
- A) Richtig. Bei der trunk-basierten Entwicklung ist Branching verboten. Eine Zusammenführung der separaten Trunks findet somit nicht statt. Dies spart Zeit und trägt dazu bei, die Zeit zwischen den Releases zu reduzieren. (Literatur: A, Kapitel 4)
  - B) Falsch. Ursache ist das Branching, das bei der Zusammenführung Probleme verursacht. Testen bietet hier keine Lösung.
  - C) Falsch. Canary-Releases können sich zwar positiv auf Continuous Deployment (kontinuierlichen Bereitstellung) auswirken, aber das größte Problem ist die Zeit, die für das Portieren von Code aufgewendet wird.
  - D) Falsch. Die Einführung von Telemetrie ist zwar eine gute Idee, aber das Hauptproblem ist die Zeit, die für das Portieren von Code aufgewendet wird.

### 14 / 40

Was zählt **nicht** zu den Vorteilen einer lose gekoppelten Architektur?

- A) Die Entwickler können unabhängig an Änderungen der Software arbeiten.
  - B) Die Systeme können unabhängig voneinander verändert und validiert werden.
  - C) Das Projektmanagement wird nur minimal überwacht.
  - D) Unit-Tests und Integrationstests laufen automatisiert.
- 
- A) Falsch. Dies ist sowohl eine Folge als auch ein Vorteil einer lose gekoppelten Architektur, weil es die Arbeit beschleunigt.
  - B) Falsch. Das ist die Definition einer lose gekoppelten Architektur und bietet viele Vorteile.
  - C) Richtig. Dies ist ein Aspekt der Agilen Arbeit und hat nichts mit lose gekoppelten Architekturen zu tun. (Literatur: A, Kapitel 5)
  - D) Falsch. Dies ist eine der Folgen einer lose gekoppelten Architektur und einer ihrer Vorteile, weil es Arbeit spart und hilft, Continuous Integration (kontinuierliche Integration) und Continuous Development (kontinuierliche Entwicklung) einzuführen.

## 15 / 40

Tina ist für die Verbesserung der Architektur und Anwendungsumgebungen eines Produkts zuständig. Bei ihrer ersten Bewertung der Anwendungen stellt sie Folgendes fest:

- Das Team kann jederzeit Deployments (Bereitstellungen) durchführen, vorausgesetzt es benachrichtigt die Service Owner der betroffenen Services.
- Die Anwendungen könnten mit Hilfe unabhängiger Komponenten bereitgestellt werden.
- Obwohl alle Teammitglieder die Anwendungen bereitstellen könnten, hat nur ein Entwickler den entsprechenden Zugriff und die Erlaubnis, dies zu tun.
- Um eine Verbindung zu den angrenzenden Diensten herzustellen, von denen die Anwendungen abhängen, muss das Team eine Reihe von Skripten ausführen.

Wie kann Tina ihrem Team **am besten** dabei helfen, eine bessere Infrastrukturmgebung zu schaffen?

- A) Indem sie dem Team hilft, eine lose gekoppelte Architektur zu erstellen, um unabhängig von anderen Services, zu denen eventuell eine Abhängigkeit besteht, Code oder ein Release der Anwendung bereitstellen zu können.
  - B) Indem sie einen Hackathon mit allen technischen Führungskräften und erfahrenen Entwicklern organisiert, um die aktuelle Infrastruktur zu erneuern und durch eine neue, eng gekoppelte Architektur zu ersetzen.
  - C) Indem sie die angrenzenden Anwendungen neu erstellt, um Abhängigkeiten zu beheben und dem Team so zu ermöglichen, Skripte schneller auszuführen und seine Anwendungen unabhängig von anderen Anwendungen bereitzustellen.
  - D) Indem sie die Teammitglieder darin schult, die Anwendungen sicher bereitzustellen, ihnen die entsprechenden Privilegien einräumt und sie im Change Advisory Board (CAB) aufnimmt.
- 
- A) Richtig. Eine lose gekoppelte Architektur ermöglicht hohe Leistung und Testbarkeit. (Literatur: A, Kapitel 5)
  - B) Falsch. Hackathons sind nützlich, um Menschen zusammenzubringen oder ein bestimmtes Problem zu lösen, nicht aber um eine bessere Infrastrukturmgebung zu schaffen. Außerdem sollte die Architektur nicht eng gekoppelt sein.
  - C) Falsch. Die Neuerstellung der angrenzenden Anwendungen würde zu viel Zeit in Anspruch nehmen und die Gesamtkosten der Anwendung steigern.
  - D) Falsch. CAB-Besprechungen gehen möglicherweise genau in die entgegengesetzte Richtung von Continuous Delivery (kontinuierlicher Lieferung). Das CAB sollte abgeschafft und nach Möglichkeit durch robustes Testen und Automatisierung ersetzt werden.

16 / 40

Eine Deployment Pipeline umfasst in der Regel mehrere Feedbackschleifen.

Was ist der **größte** Vorteil, den die Nutzung von Feedbackschleifen bietet?

- A) Bessere Kommunikation zwischen dem Entwicklungs- und dem Betriebsteam
  - B) Häufigere und zuverlässigere Lieferungen qualitativ hochwertiger Software an die Benutzer
  - C) Einbindung der Kunden in mehreren Phasen des Entwicklungsprozesses
  - D) Regelmäßige Durchführung von Benutzerabnahmetests, um die Meinung der Stakeholder einzuholen
- 
- A) Falsch. Die Kommunikation zwischen den Teams hängt nicht von der Nutzung von Feedbackschleifen ab.
  - B) Richtig. Feedbackschleifen dienen der häufigeren und zuverlässigeren Lieferung von Software. (Literatur: A, Kapitel 4)
  - C) Falsch. Kundeneinbindung ist zwar gut, aber nicht der größte Vorteil, den Feedbackschleifen bieten.
  - D) Falsch. Benutzerabnahmetests sind eine Form des Feedbacks und nicht einer der Hauptvorteile von Feedbackschleifen.

17 / 40

Ein Software-Entwicklungsteam wird kritisiert, weil es Features bereitstellt, die nicht den Kundenerwartungen entsprechen.

Wie kann das Team diese Situation **am besten** verbessern?

- A) Einen separaten Zyklus von Benutzerabnahmetests erstellen vor dem Deployment (Bereitstellung)
  - B) Die Anforderungen anderer Stakeholder in der Priorität zurückstufen oder vollständig entfernen
  - C) Die Anforderungen zu Beginn der Entwicklungsprojekte statisch gestalten oder festlegen
  - D) Die Erkenntnisse der Stakeholder während des gesamten Lebenszyklus der Entwicklung aktiv einholen
- 
- A) Falsch. Diese Aktivität erfolgt zu spät und ist zu isoliert, um die Abstimmung zwischen dem Entwicklungsteam und den Kundenerwartungen zu verbessern.
  - B) Falsch. Die Anforderungen sollten nach verschiedenen Kriterien priorisiert werden und nicht danach, welche Stakeholder-Gruppe, sich vor kurzem beschwert hat.
  - C) Falsch. Bei festgelegten Anforderungen kann man nicht auf Änderungen der Kundenanforderungen oder des geschäftlichen Umfelds reagieren.
  - D) Richtig. Erhält das Team nicht automatisch Feedback, sollte das Team diesen Input proaktiv einholen, um die Produktqualität zu verbessern. (Literatur: A, Kapitel 8)

**18 / 40**

DevOps, Agile, Scrum, Lean und Kanban sind alle auf ihre eigene Art und Weise in der Lage, einer Organisation Mehrwert zu bieten.

Welchen **eindeutigen** Wert bietet DevOps einer Organisation?

- A) DevOps legt einen nachhaltigen Entwicklungsrhythmus fest und ermöglicht so regelmäßige Feedbackschleifen seitens der Kunden.
  - B) DevOps sorgt dafür, dass die Entwicklungs- und Betriebsteams das gleiche Taskboard zur Koordinierung der Aufgaben nutzen.
  - C) DevOps legt Wert auf kontinuierliche Releases, um dem Kunden in einem effizienten Prozess schnell Wert zu liefern.
  - D) DevOps stellt neue Services Just-in-Time bereit und legt gleichzeitig Wert auf Effizienz, indem es verschwenderische Arbeit eliminiert.
- 
- A) Falsch. Häufigere Feedbackzyklen sind das Ergebnis einer höheren Entwicklungsgeschwindigkeit. Diese ist jedoch in der Regel eine Folge von Scrum oder einer anderen Agilen Methode und kein eindeutiges Merkmal von DevOps.
  - B) Falsch. Die Arbeit an nur einem Taskboard, ist typisch für Kanban. DevOps dagegen fordert neben der Nutzung des gleichen Taskboards auch, dass die Teams zusammenarbeiten.
  - C) Richtig. Mehrwert und Prozessoptimierung sind die Schlüssel zur Verbesserung der Business Continuity und Agilität des Unternehmens. Durch die Optimierung von Prozessen, das Ermöglichen von Experimentieren und die Einführung von Continuous Delivery (kontinuierlicher Lieferung) kann man dem Kunden schneller mehr Wert liefern. (Literatur: A, Kapitel 1)
  - D) Falsch. Just-in-time zu liefern und Verschwendung zu eliminieren ist zugegebenermaßen großartig, beruht aber in erster Linie auf der Einführung von Lean und ist kein eindeutiges Merkmal von DevOps.

**19 / 40**

DevOps zielt darauf ab, die Kundenzufriedenheit und Kundenerfahrung zu verbessern, indem es den Fokus auf die Wertschöpfung legt.

Wie wird dies **in erster Linie** bewerkstelligt?

- A) Durch Automatisieren von Testen und Deployment (Bereitstellung)
  - B) Durch häufigere Lieferung von Produkten
  - C) Durch Entwicklung gut funktionierender Produkte
  - D) Durch Einsatz hochkompetenter Entwickler
- 
- A) Falsch. Automatisierungspraktiken wirken sich nicht auf den Kunden aus.
  - B) Richtig. Ziel von DevOps ist, häufiger hochqualitative Software bereitzustellen und so für eine bessere Kundenerfahrung zu sorgen. (Literatur: A, Kapitel 2)
  - C) Falsch. Die Kunden gehen davon aus, dass das Produkt gut funktioniert. Daher würde dies die Kundenerfahrung nicht grundlegend verbessern.
  - D) Falsch. Die Kunden nehmen die Kompetenz des Entwicklers nicht wahr.

20 / 40

Viele DevOps-Teams setzen bei der Steuerung ihres Entwicklungsprozesses auf Visualisierung, beispielsweise in Form einer Kartenwand. Kartenwände erhöhen die visuelle Kontrolle des Teams.

Inwiefern unterstützt Visualisierung die Praktiken von DevOps?

- A) Sie heitern die Büroumgebung auf und steigern so Produktivität und Mitarbeiterzufriedenheit.
  - B) Sie bieten den Führungskräften die Möglichkeit, das Team für seine Aktivitäten zur Verantwortung zu ziehen und die Arbeit aufzuteilen.
  - C) Sie helfen dem Management, das Team und die zu erledigende Arbeit in den Griff zu bekommen.
  - D) Sie verdeutlichen den Arbeitsfortschritt und wirken sich dadurch positiv auf die Kultur und die Leistung des Teams aus.
- 
- A) Falsch. Visualisierungen mögen in einem Büro gut aussehen, aber gutes Aussehen alleine fördert weder Produktivität noch Mitarbeiterzufriedenheit.
  - B) Falsch. Visualisierung wirkt sich nicht auf die Zuteilung der Arbeit und die Verantwortung aus.
  - C) Falsch. Die Steuerung seitens des Managements ist nicht Ziel der Visualisierung und unterstützt auch nicht die DevOps-Praktiken.
  - D) Richtig. Visualisierung verdeutlicht den Arbeitsfortschritt und trägt so zur Produktivität bei. (Literatur: A, Kapitel 7)

## 21 / 40

Das Entwicklungsteam kann innerhalb der geforderten Zeit neue Features erstellen. Häufig werden diese aber nicht pünktlich geliefert. Eine Untersuchung fördert folgende Probleme zutage:

- Es treten wiederholt Regressionsfehler auf.
- Obwohl Zeit für die Fehlerbehebung aufgewendet wird, beschwerten sich die Kunden immer wieder über Fehler in der Produktionsumgebung.
- Wird ein neues Feature gefordert, gerät das Entwicklungsteam stark unter Stress.

Das zugrundeliegende Problem ist, dass das Team nicht über eine effektive Teststrategie verfügt.

Was sollte das Team unternehmen, um dies zu beheben?

- A)** Das Team sollte alle Tests und die Commit-Phase automatisieren und durch Schulungen dafür sorgen, dass die Teammitglieder den Prozess Continuous Integration (kontinuierlichen Integration) besser verstehen.
  - B)** Das Team sollte alle Unit-Tests, Komponententests und Integrationstests automatisieren und die Zusammenarbeit zwischen den Testern und dem Entwicklungsteam verbessern.
  - C)** Das Team sollte die Prozesse für Testen und Deployment (Bereitstellung) automatisieren und bei Bedarf ein Hardware-Upgrade durchführen oder das Konfigurationsmanagement des Systems aktualisieren.
  - D)** Das Team sollte die Zusammenarbeit zwischen dem Entwicklungs- und dem Betriebsteam durch mehr Monitoring und Protokollierung sowie die Nutzung von Virtualisierung steigern.
  - E)** Das Team sollte sicherstellen, dass das Entwicklungsteam den Deployment-Prozess versteht und die Zusammenarbeit mit dem Betriebsteam verstärken.
- 
- A)** Falsch. Ein unzureichendes Management des Prozesses Continuous Integration führt insbesondere dazu, dass Code seltener als einmal täglich bereitgestellt und die Commit-Phase unterbrochen wird. Außerdem führt es zu langen Integrationsphasen zwischen den Releases.
  - B)** Richtig. Eine ineffektive Teststrategie hat zur Folge, dass Bugs wiederholt auftreten, viel Zeit für die Fehlerbehebung aufgewendet wird und viele Kundenbeschwerden eingehen. Außerdem führt es zu Produkten von geringer Qualität und gestressten Entwicklern. Dies lässt sich durch Automatisierung und Zusammenarbeit mit den Testern lösen. (Literatur: B, Kapitel 15)
  - C)** Falsch. Führt man nicht genügend Deployments durch, so führt dies zu langen Bereitstellungszeiten, einer geringen Geschwindigkeit, Skepsis gegenüber den genannten Release-Daten, Vertrauensverlust in die Umgebung der Continuous Integration und lange Bearbeitungszeiten bei der Fehlerbehebung. Automatisierung und Upgrades können dazu beitragen, dieses Problem zu beheben.
  - D)** Falsch. Dies ist die Lösung für ein unzureichendes Konfigurationsmanagement. Ein schlechtes Konfigurationsmanagement führt insbesondere zu unerklärlichen Ausfällen in der Produktionsumgebung, nicht beherrschbaren Ereignissen bei der Bereitstellung, einem höheren Zeitbedarf für die Umgebungskonfiguration und einer langen Wiederherstellungsdauer nach Ausfällen.
  - E)** Falsch. Stellt man fehlerhaften Code bereit oder führt man nicht genügend Deployments durch, so führt dies zu langen Bereitstellungszeiten, einer geringen Geschwindigkeit, Skepsis bezüglich der Release-Daten, Vertrauensverlust in die Umgebung der Continuous Integration, einer längeren Dauer der Fehlerbehebung, dem Auffinden von Bugs, die von den Entwicklern schon lange behoben wurden, sowie weniger Demos und Präsentationen.

**22 / 40**

Zu DevOps gehört es auch, sicherzustellen, dass relevante Informationen für alle Entwickler visualisiert werden.

Was sind die **wichtigsten** Arten von Informationen, die für die Entwickler visualisiert werden sollten?

- A) Das tägliche Kundenfeedback einschließlich Beschwerden und Anfragen von Features, damit die Entwickler schnell scheitern können
  - B) Die Produktivität der einzelnen Entwickler einschließlich der von ihnen geleisteten Arbeitsstunden, damit die Entwickler miteinander konkurrieren können
  - C) Den Fortschritt des Gesamtprojekts einschließlich der Budget-Informationen, damit die Entwickler strategische Entscheidungen treffen können
  - D) Daten zur Produktqualität einschließlich der Ausfall- und Fehlerraten, damit die Entwickler fundierte Entscheidungen treffen können
- 
- A) Falsch. Kundenfeedback wird während der Entwicklung kontinuierlich entgegengenommen und muss nicht offen dargestellt werden.
  - B) Falsch. Daten zur Produktivität der einzelnen Mitarbeiter sollten nicht offen visualisiert werden. Miteinander zu konkurrieren, entspricht nicht der DevOps-Philosophie.
  - C) Falsch. Diese Art von Informationen sind für die Entwickler nicht bedeutend. Hierbei handelt es sich um Managementdaten, die nicht offen dargestellt werden müssen.
  - D) Richtig. Zu den relevanten Informationen für Entwickler, die visualisiert werden sollten, zählen Informationen über die Qualität einschließlich der Ausfall- und Fehlerraten. (Literatur: A, Kapitel 7)

**23 / 40**

Ein Unternehmen hat sich damit einverstanden erklärt, nach DevOps zu arbeiten. Das Unternehmen sucht nach einer Möglichkeit, mit der es den Umfang aller seiner Projekte managen kann, die zu dieser Arbeitsweise passen.

Welche Möglichkeit eignet sich **am besten**, um den Umfang eines DevOps-Projekts zu managen?

- A) Eine Iterationsplanung durchführen, um sich am Anfang jeder Iteration darauf zu einigen, an welchen Items gearbeitet wird
  - B) Eine vollständige Release-Planung für Features erstellen, die den Kunden vorab darüber informiert, was im Release enthalten sein wird
  - C) Das Minimum Viable Product (MVP) festlegen und so oft wie nötig überprüfen
  - D) Die Prioritäten der Arbeit vor jeder Iteration festlegen und den Teammitgliedern beim Aussuchen der Aufgaben helfen
- 
- A) Falsch. Die Iterationsplanung legt die Arbeit für eine künftige Iteration fest, nicht für den Gesamtumfang eines Projekts.
  - B) Falsch. Release-Planungen dienen in erster Linie dazu, vorherzusagen, was in einem Release enthalten ist, nicht um ein Projekt gesamtumfänglich zu managen.
  - C) Richtig. Das MVP-Management wird genutzt, um den Projektumfang zu managen. Dies ist sehr kundenorientiert und passt daher zu DevOps. (Literatur: A, Kapitel 8)
  - D) Falsch. Prioritäten können zwar hilfreich bei der Planung sein, aber das MVP ist wichtiger. Außerdem wird die Priorisierung der Arbeit in erster Linie dazu genutzt, die Arbeit für die nächste Iteration festzulegen.

24 / 40

Single-Piece-Flow bedeutet in der Software-Entwicklung, dass man sich auf ein einziges Feature konzentriert und kontinuierlich an diesem arbeitet, bis es fertig ist.

Warum ist Single-Piece-Flow bei DevOps wichtig?

- A) Es ermöglicht dem Team, in einem nachhaltigen und planbaren Tempo mit konstanter Geschwindigkeit zu arbeiten.
  - B) Es hilft dem Team bei der Wertschöpfung, weil es Features mit hoher Priorität dann liefert, wenn sie gebraucht werden.
  - C) Es stärkt die gemeinsame Verantwortung aller Teammitglieder bei der Auswahl der Aufgaben.
  - D) Es reduziert die Engpässe, weil es die Zahl der Teammitglieder begrenzt, die an einem Feature arbeiten.
- 
- A) Falsch. Dafür ist der Rhythmus zuständig. Single-Piece-Flow ermöglicht, dass Features mit der höchsten Priorität schnellstmöglich fertiggestellt werden.
  - B) Richtig. Single-Piece-Flow ermöglicht, dass Features mit der höchsten Priorität schnellstmöglich fertiggestellt werden. (Literatur: A, Kapitel 2)
  - C) Falsch. DevOps-Teams tragen zwar gemeinsam Verantwortung, aber Single-Piece-Flow ist wichtig, weil es ermöglicht, dass Features mit der höchsten Priorität schnellstmöglich fertiggestellt werden.
  - D) Falsch. Single-Piece-Flow ermöglicht, dass Features mit der höchsten Priorität schnellstmöglich fertiggestellt werden und begrenzt nicht die Zahl der Mitarbeiter, die an einem Feature arbeiten dürfen.



25 / 40

Tool Company arbeitet an der Entwicklung einer E-Commerce-Website. Das Projekt ist von größter Wichtigkeit. Die Kunden sind unzufrieden und beschweren sich, dass die E-Commerce-Website nicht wie erwartet funktioniert.

Führende Manager des Unternehmens argumentieren, dass ein Agiles Vorgehen, das Risiko noch steigern würde, während erfahren IT-Manager sagen, dass dies die einzige Möglichkeit sei, um schnell Wert zu liefern. Die IT-Manager möchten Agile mit DevOps kombinieren. Sie wünschen sich, häufige Releases und Input.

Das Unternehmen Tool Company beauftragt einen Berater, um sicher zu gehen, dass es das Richtige tut.

Welchen Rat sollte der Berater dem Unternehmen geben?

- A) Arbeiten Sie nicht Agil oder stellen Sie nicht auf DevOps um, da sich die Kunden sowieso schon beschweren, dass die E-Commerce-Website nicht wie erwartet funktioniert.
  - B) Arbeiten Sie Agil, ergänzen Sie DevOps-Praktiken, geben Sie Ihren Entwicklern den Freiraum, Deployments (Bereitstellungen) ohne externe Genehmigung durchzuführen sobald sie können und ständig Kundenfeedback einzuholen.
  - C) Nutzen Sie die Methoden Agile und DevOps zur Erstellung der E-Commerce Website, aber ergänzen Sie manuelle Benutzerabnahmetests und einen Prozess mit Change Approval Board (CAB) vor dem Release.
  - D) Nutzen Sie die Methoden Agile und DevOps im Rahmen der Softwareentwicklung, aber geben Sie nur fertige Produkte an die Kunden frei, um Beschwerden und Unzufriedenheit seitens der Kunden zu vermeiden.
- 
- A) Falsch. Dies mag sicher erscheinen, weil sich die Kunden bereits beschweren, verschlimmert aber das Problem. Nur wenn man schnell und oft scheitert, kann man schnellen Fortschritt erzielen und den Kunden den Wert bieten, den sie möchten.
  - B) Richtig. Dies scheint zwar zu mehr Unzufriedenheit und Beschwerden zu führen, aber Feedback zu sammeln und Kundenprobleme schnell zu lösen, ist der direkte Weg zu zufriedenen Kunden. (Literatur: A, Kapitel 8)
  - C) Falsch. Den Arbeitsmethoden von Agile und DevOps sollte man nicht entgegenwirken, indem man die Releasegeschwindigkeit durch manuelle Tests und CAB-Prozesse verlangsamt.
  - D) Falsch. Den Arbeitsmethoden von Agile und DevOps sollte man nicht entgegenwirken, indem man nur fertiggestellte Produkte freigibt und damit Kundenwert und Feedback verlangsamt.

## 26 / 40

Ein Team hat gut zusammengearbeitet, aber die Kunden beschwerten sich trotzdem noch. Das Team würde gerne mehr experimentieren und DevOps einführen.

Das Management erlaubt dem Team zu experimentieren und seine Arbeitsweise zu ändern, vorausgesetzt die Risiken sind annehmbar. Das Unternehmen verfolgt die Politik, Risiken möglichst auf ein Mindestmaß zu reduzieren. Das Team diskutiert das Problem.

- Laut **Hichem** sollte das Team aktuell weder experimentieren noch seine Arbeitsweise ändern, da dies bei den Kunden zu noch größerer Unzufriedenheit führen könnte. Das Team sollte stattdessen ein Formular für Kundenfeedback erstellen.

- **Kira** sagt, dass es immer gut sei, Veränderungen in einem großen Release zu bündeln, um Störungen und Ausfälle in der Betriebsumgebung auf ein Mindestmaß zu beschränken. Ein umfangreiches Release sollte eine höhere Qualität des Produkts in der Produktionsumgebung sicherstellen.

- **Nazim** führt an, dass das Team erst auf die DevOps-Methode umstellen könne, sobald es gute Leistung erbringt und sich die Zahl der Kundenbeschwerden verringert hat. Zusätzliche manuelle Tests sollten dazu beitragen können, die Zahl der Beschwerden zu reduzieren.

- **Rose** sagt, dass es zwar unlogisch klingt, dass aber die Arbeitsweise nach der DevOps-Methode, bei der man viel experimentiert, das Risiko am stärksten senkt. Dies sei die beste Wahl, vorausgesetzt das Team habe Tests und Versionskontrolle gut automatisiert.

Wer unterbreitet den Vorschlag, der in diesem Szenario **am besten** funktioniert?

- A) Hichem
- B) Kira
- C) Nazim
- D) Rose

- A) Falsch. Nichts zu verändern und nicht zu experimentieren wäre in diesem Szenario ein Fehler. Experimentieren ist notwendig, um sicherzustellen, dass die Kunden auch etwas anderes sehen können, was ihnen möglicherweise gefällt.
- B) Falsch. Dieses Vorgehen mag logisch klingen, aber ein umfangreiches Release ist riskanter als kleine Experimente freizugeben und schnell Feedback einzuholen. Bei guter Release- und Testautomatisierung sind Rollbacks einfach durchzuführen, sodass ein solches Release weniger riskant sein sollte als ein umfangreiches Release.
- C) Falsch. Experimentieren klingt zwar riskant, ermöglicht es aber den Kunden, Feedback zu kleinen, weniger riskanten Veränderungen zu geben und sollte so zu mehr Kundenzufriedenheit beitragen. Zu warten bis die Kundenzufriedenheit steigt, wäre in diesem Fall die falsche Lösung.
- D) Richtig. Nicht jede Veränderung führt zu mehr Risiken. Ein Release ist dann riskant, wenn es groß und nicht gut dokumentiert ist, die Tests nicht automatisiert sind und sich menschliche Fehler einschleichen. Häufige, kleine Veränderungen sind weniger riskant. Dies gilt vor allem in Fällen, in denen ein Rollback einfach möglich ist. (Literatur: A, Kapitel 4)

27 / 40

Ein Web-Shop spezialisiert sich auf Bekleidung aus den 1970er und 1980er Jahren. Der Web-Shop ist sehr erfolgreich. Sein aktueller Kundenstamm setzt sich zusammen aus Privatpersonen, die historische Kostüme suchen, und Kostümverleihern. Das Unternehmen möchte expandieren.

Ein Team diskutiert verschiedene Wachstumsoptionen und entscheidet sich dann für die hypothesengetriebene Entwicklung. Eine mögliche Hypothese für diesen Webshop ist, dass der Shop mehr Kunden anziehen wird, wenn sein Inhalt auch in einer mobilen App und nicht nur auf der Website präsentiert wird.

Was sollte unternommen werden?

- A)** - Der Kunde definiert eine Hypothese und das Team legt fest, welcher Test beziehungsweise welches Experiment durchgeführt werden sollte.
    - Im nächsten Schritt baut das Team ein Minimum Viable Product (MVP), das die formulierte Hypothese unterstützt und testet die Reaktionen der Kunden.
    - Nach einer kurzen Feedbackrunde mit den Kunden wird das MVP überarbeitet, um maximalen Kundenwert sicherzustellen.
  - B)** - Der Kunde entwickelt auf der Grundlage seiner Bedürfnisse eine Hypothese und das Team spricht mit den Kunden.
    - Im nächsten Schritt baut und testet das Team einen Prototyp und ermittelt das finale MVP.
    - Der Prototyp wird zum MVP umgebaut und dem Kunden präsentiert, damit dieser die Hypothese testet.
  - C)** - Das Team erhebt Informationen vom Kunden und formuliert auf der Grundlage dieser Informationen eine Hypothese.
    - Danach bestimmt das Team, welches Experiment oder welcher Test durchgeführt werden kann, um die Hypothese zu belegen oder zu widerlegen.
    - Das Experiment wird durchgeführt und analysiert und der Kunde dann informiert und um mehr Feedback gebeten.
  - D)** - Das Team erhebt Informationen von Stakeholdern und formuliert auf dieser Grundlage alternative Hypothesen.
    - Danach baut und testet das Team ein MVP und präsentiert dieses dem oberen Management der Organisation.
    - Die finale Hypothese wird dann auf der Grundlage der Ergebnisse aus dem Akzeptanztest und der Reaktion des Managements gebildet.
- 
- A)** Falsch. Die Hypothese und der Test sollten gemeinsam von der gleichen Einheit entwickelt werden. Die Hypothese wird in der Regel nicht von den Kunden erstellt. Liegt eine Grundlage vor, könnte das Team darauf aufbauen. Das MVP sollten erst gebaut werden, nachdem bewiesen wurde, dass die Hypothese wahr ist.
  - B)** Falsch. Die Hypothese wird in der Regel nicht von den Kunden erstellt. Liegt eine Grundlage vor, könnte das Team darauf aufbauen. Das MVP sollten erst gebaut werden, nachdem bewiesen wurde, dass die Hypothese wahr ist.
  - C)** Richtig. Das Team sollte die Bedürfnisse der Kunden kennen und deren Hypothesen testen. Ein MVP sollte nicht erstellt werden. (Literatur: A, Kapitel 4 und 8)
  - D)** Falsch. Diese Option schließt die Kunden fast völlig aus, obwohl Kundenfeedback das Wertvollste ist.

28 / 40

Was zählt **nicht** zu den wahrscheinlichen Folgen der Praktiken von Lean Management?

- A) Die Organisation erhält eine generativere Kultur.
  - B) Die Leistung der Software-Lieferung wird verbessert.
  - C) Die Teammitglieder sind weniger Burnout gefährdet.
  - D) Die Manager achten stärker auf Gewinne.
- 
- A) Falsch. Lean-Management-Praktiken legen Wert auf die Schaffung von Flow und Verbesserungen. Dies entspricht einer generativeren Kultur.
  - B) Falsch. Lean-Management-Praktiken legen Wert auf die Schaffung von Flow und die Reduzierung von Fehlern. Dies führt für gewöhnlich zu einer besseren Leistung im Bereich der Software-Lieferung (Delivery).
  - C) Falsch. Lean-Management-Praktiken legen Wert auf die Schaffung eines nachhaltigen Orts und darauf, Hindernisse aus dem Weg zu räumen. Dies bedeutet in der Regel, dass die Teammitglieder weniger Burnout gefährdet sind.
  - D) Richtig. Lean Management sollte keinen Wert auf Gewinne legen, sondern sich darauf konzentrieren Flow zu etablieren, was wahrscheinlich zu mehr Kundenzufriedenheit führt. Letzteres führt dann möglicherweise zu höheren Gewinnen, aber das ist nicht das, worauf sich das Management konzentriert. (Literatur: A, Kapitel 7)

## 29 / 40

Ein Team hat Schwierigkeiten bei der Identifizierung von Problemen und beim Troubleshooting für Anwendungen, die in seiner Infrastruktur bereitgestellt werden. Dies verursacht viele Störungen und Ausfälle und ist daher ein Problem. Die Kunden rufen beim Service Desk häufig an und beschweren sich. Der Service Desk muss die vielen Fragen dann an das Team weiterleiten, sodass dieses unter Zeitdruck gerät.

Das Team hat zwar seinen Workflow dokumentiert und hält die Prozesse ein, hat aber keine Idee, was die Ursache für das Problem sein könnte. Das Team arbeitet mit einem Kanban-Board und scheint Flow erzielt zu haben.

Welche Lösung hilft langfristig **am besten** die Zahl der Probleme zu reduzieren?

- A) Einen Berater anstellen, der den Workflow neu abbildet und neue Prozesse auf dem Kanban-Board des Teams einrichtet
  - B) Wertvolle Zeit gewinnen, indem man den Service Desk anweist, keine Fragen mehr an das Team weiterzuleiten
  - C) Messen, wie viele Probleme jedes einzelne Teammitglied hat und einen Wettbewerb ausloben, um Verbesserung zu erzielen
  - D) Eine Monitoringsstrategie einrichten, indem man Daten speichert, ein Dashboard erstellt und Benachrichtigungen einrichtet
- 
- A) Falsch. Das Team arbeitet bereits mit einem Kanban-Board und hat Flow erzielt. Es gibt daher keinen offensichtlichen Grund, die Prozesse zu ändern oder den Workflow neu zu gestalten. Das Problem liegt wahrscheinlich außerhalb der direkten Prozesse des Teams und sollte durch eine gute Monitoringsstrategie aufgezeigt werden.
  - B) Falsch. Dies mag die Probleme des Teams kurzfristig lösen, aber das eigentliche Problem liegt wahrscheinlich außerhalb der direkten Prozesse des Teams und sollte durch eine gute Monitoringsstrategie aufgezeigt werden.
  - C) Falsch. Wettbewerb ist niemals im Sinne von DevOps. Das Team sollte zusammenarbeiten und sich trauen, Fehler zu machen. Das Problem liegt wahrscheinlich außerhalb der direkten Prozesse des Teams und sollte durch eine gute Monitoringsstrategie aufgezeigt werden.
  - D) Richtig. Eine solide Monitoringsstrategie einzurichten bietet Einblicke, was schief gehen kann und wie oft es tatsächlich schief geht. Damit unterstützt die Strategie das Team, die Ursachen der Probleme zu identifizieren und hilft langfristig am besten. (Literatur: B, Kapitel 11)

**30 / 40**

Ein Team hat Schwierigkeiten, die Arbeit rechtzeitig fertigzustellen. Das Team hat Probleme, die Produkte pünktlich an die Kunden zu liefern.

Inwiefern helfen Work-in-Progress-Begrenzungen (WIP-Limits) diesem Team, Flow zu etablieren?

- A)** Sie verändern die Rollen aller involvierten Stakeholder und Teammitglieder.
  - B)** Sie unterstützen die Identifizierung von Engpässen, die dem Team helfen, die Arbeit zu verbessern und zu optimieren.
  - C)** Sie verbessern das Lebenszyklusmanagement von Anwendungen und Kommunikationstools (Hilfsmitteln).
  - D)** Sie begrenzen die Anzahl der Tools und Anwendungen, die das Team zum Arbeiten benötigt.
- 
- A)** Falsch. Das Team mag der Ansicht sein, eine Rolle, die einen Engpass darstellt, verändern zu müssen, aber WIP-Limits verändern nicht unmittelbar oder immer die Rolle der Stakeholder oder Teammitglieder.
  - B)** Richtig. Auferlegte WIP-Limits tragen immer dazu bei, Engpässe aufzuzeigen und helfen dem Team folglich, den Flow zu verbessern. (Literatur: A, Kapitel 7)
  - C)** Falsch. Das Team mag der Meinung sein, es sollte das Lebenszyklusmanagement seiner Anwendungen und Kommunikationstools ändern, weil diese einen Engpass darstellen, aber WIP-Limits helfen in diesem Fall nicht immer und nicht unmittelbar.
  - D)** Falsch. Das Team mag der Meinung sein, es sollte die Anzahl der Tools und Anwendungen, mit denen es arbeitet, begrenzen, weil diese einen Engpass darstellen, aber WIP-Limits helfen in diesem Fall nicht immer und nicht unmittelbar.

### 31 / 40

Ein Team arbeitet an mehreren Projekten, die laut Management alle die gleiche Priorität haben. Das Team steht daher unter Druck, täglich an allen Projekten arbeiten und wöchentliche Fortschrittsberichte für alle Projekte abgeben zu müssen.

Das Team braucht einen Weg, um maximale Teameffizienz und qualitativ hochwertige Ergebnisse sicherzustellen. Das Team möchte den Kunden und dem Unternehmen maximalen Wert liefern.

Was sollte das Team unternehmen?

- A) Ein Teammitglied zur Verbindungsperson ernennen, damit sich die Manager nicht direkt an das Team wenden. Der Nachteil ist, dass das Team in diesem Fall kein direktes Kundenfeedback mehr erhält.
  - B) So lange das Team alle zufriedenstellen kann, sollte es so weiter arbeiten wie bisher. Der Nachteil ist, dass manche Teammitglieder häufig die Aufgaben wechseln müssen.
  - C) Work-in-Progress (WIP) begrenzen und eine Visualisierungstechnik für das Team einführen. Der Nachteil ist, dass das Team sich an die neue Arbeitsweise anpassen muss.
  - D) Das Team in kleinere Unterteams mit zwei bis drei Teammitgliedern aufteilen, die separat an den einzelnen Projekten arbeiten. Der Nachteil ist, dass nicht alle Teams fähig sind, die Arbeit in ihrem Team fertigzustellen.
- 
- A) Falsch. Dies mag als logische Lösung erscheinen, aber die Forschung hat gezeigt, dass eine größere Entfernung zwischen den DevOps-Teams und den Kunden die Qualität senkt und die Entwicklungsdauer von Projekten verlängert.
  - B) Falsch. Die Kunden sind bereits unzufrieden. Mit dieser Methode weiterzuarbeiten ist keine Lösung.
  - C) Richtig. Am besten hilft dem Team die Arbeit über die Einführung von WIP-Limits und einer Visualisierungstechnik, wie beispielsweise einem Kanban-Board, zu steuern und Flow zu schaffen. Dies sorgt für die Lieferung von maximalem Wert. (Literatur: A, Kapitel 7)
  - D) Falsch. Ein Problem zu lösen und dabei ein anderes Problem zu schaffen, ist nicht die beste Lösung. Die Kunden mögen zwar das Gefühl haben, dass Ressourcen für ihre Arbeit aufgewendet werden, aber Abhängigkeiten machen es schwer, Qualität in einer vernünftigen Geschwindigkeit bereitzustellen.

**32 / 40**

BigBlue Software ist ein kleines Unternehmen, das für andere Unternehmen Software erstellt. Das Unternehmen verfügt über ein Entwicklungsteam und ein Betriebsteam. Die Teams haben jeweils eigene Manager, arbeiten aber gut zusammen. Da das Unternehmen klein und der Geschäftsführer ehrgeizig ist, stehen die Teams manchmal unter Zeitdruck.

Unter diesem Zeitdruck machen beide Teams Fehler, die bereits zu Software-Bugs und Ausfällen geführt haben. Die Teams haben in diesen Fällen mit vereinten Kräften daran gearbeitet, die Ausfälle zu beheben oder ein Problem schnellstmöglich zu lösen. Die Manager sind stolz auf ihre Teammitglieder, weil diese ihre Fehler offen kommunizieren, gemeinsam daran arbeiten, diese zu lösen und Kundenwert liefern.

Würde diese Organisation von einer Veränderung ihrer Unternehmenskultur nach Westrum profitieren?

- A) Ja, die Teamstruktur mit einem Manager zeigt, dass die Unternehmenskultur bürokratisch ist und geändert werden sollte.
  - B) Ja, die Fehler, die unter Druck gemacht werden, zeigen, dass die Unternehmenskultur pathologisch ist und geändert werden sollte.
  - C) Nein, die Zusammenarbeit und der offene Umgang mit Fehlern zeigen, dass die Unternehmenskultur generativ ist und nicht geändert werden sollte.
  - D) Nein, die Forderungen nach Kundenwert zeigen, dass die Unternehmenskultur marktorientiert ist und nicht geändert werden sollte.
- 
- A) Falsch. Die Teams haben zwar Manager, aber das alleine macht noch keine bürokratische Kultur. Die Teammitglieder scheuen sich nicht, ihre Fehler offen zu kommunizieren und arbeiten mit anderen zusammen. Dies sind Zeichen für eine generative Kultur, in der alle Teams und Teammitglieder zusammenarbeiten, um Kundenwert zu liefern.
  - B) Falsch. Die Teams machen zwar viele Fehler und stehen unter Zeitdruck, fürchten sich aber nicht, Fehler offen zu kommunizieren und arbeiten gemeinsam daran, diese Fehler zu beheben. Dies sind Zeichen für eine generative Kultur, in der alle Teams und Teammitglieder zusammenarbeiten, um Kundenwert zu liefern.
  - C) Richtig. Die Teams zeigen, dass sie Fehler machen, fürchten sich aber nicht, diese zu kommunizieren und arbeiten auch teamübergreifend zusammen an der Lösung dieser Fehler. Die Manager ermutigen dieses Verhalten. Dies sind Zeichen für eine generative Kultur, in der alle Teams und Teammitglieder zusammenarbeiten, um Kundenwert zu liefern. (Literatur: A, Kapitel 3)
  - D) Falsch. Ob die Organisation marktorientiert ist oder nicht, ist nicht klar, da dies nicht Teil der Unternehmenstypologie nach Westrum ist. Die Teammitglieder scheuen sich nicht, ihre Fehler offen zu kommunizieren und arbeiten mit anderen zusammen. Dies sind Zeichen für eine generative Kultur, in der alle Teams und Teammitglieder zusammenarbeiten, um Kundenwert zu liefern.



33 / 40

Ein Teammitglied sagt, dass das Unternehmen organisationales Lernen benötigt, das manchmal auch als generative Kultur bezeichnet wird. Organisationales Lernen ermöglicht es, den Kunden zu helfen, sorgt für Qualität und schafft einen Wettbewerbsvorteil sowie eine engagierte Belegschaft.

Warum ist dies richtig?

- A) Weil organisationales Lernen bedeutet, dass man Fehler identifiziert und einzelne Teammitglieder dafür verantwortlich macht, um aus diesen Fehlern zu lernen
  - B) Weil organisationales Lernen bedeutet, dass man das Team ständig in den neuesten Methoden von Agile und Lean schult, um so für mehr Flow zu sorgen
  - C) Weil organisationales Lernen ein Denken voraussetzt, das den Mitarbeitern das Gefühl vermittelt, dass sie unterstützt werden und ihr Urteilsvermögen geschätzt wird
  - D) Weil organisationales Lernen eine Kultur des Forschens fordert, die automatisch Hypothesen erstellt, um so fundierte Entscheidungen zu treffen
- 
- A) Falsch. Lernende Organisationen können zwar Fehler identifizieren, um aus diesen zu lernen, aber es ist nicht Ziel einer generativen Kultur, die Teammitglieder für Fehler verantwortlich zu machen.
  - B) Falsch. Eine lernende Organisation bedeutet, dass die Organisation über eine Kultur verfügt, in der Experimentieren und Lernen möglich sind. Dies verringert Burnout und führt zu kontinuierlichem Fortschritt. Alleine durch Schulungen in Agile und Lean kann dies nicht erreicht werden.
  - C) Richtig. Mitarbeiter, die sich von ihren Arbeitgebern unterstützt fühlen, über die entsprechenden Tools (Hilfsmittel) und Ressourcen verfügen, um ihre Arbeit korrekt ausführen zu können und das Gefühl haben, dass ihr Urteilsvermögen geschätzt wird, liefern qualitativ hochwertigere Arbeit. Bessere Arbeitsergebnisse führen zu einer besseren Leistung im Bereich der Software-Lieferung (Delivery) und dies wiederum hebt die Leistung der Organisation auf ein höheres Niveau. (Literatur: A, Kapitel 10)
  - D) Falsch. Eine lernende Organisation kann, muss aber nicht immer bedeuten, dass wissenschaftliche Forschung betrieben wird. Eine lernende Organisation braucht keine Forschungskultur und erstellt nicht automatisch Hypothesen.

34 / 40

Ein Personalteam wird gebeten, eine neue Richtlinie für die Anstellung neuer DevOps-Teammitglieder zu verfassen.

Was ist die **beste** Richtlinie für die Anstellung von DevOps-Teammitgliedern?

- A) Anonymisierte Bewerbungen zu ermöglichen und Bewerber zu bevorzugen, die für Diversität im Team sorgen, weil dies die Ergebnisse des Teams verbessert
  - B) Den Managern die Möglichkeit zu geben, nachzuforschen, welche Mitarbeiter in einem anderen Unternehmen die beste Leistung erbringen und dann zu versuchen, diese Mitarbeiter abzuwerben, um die besten Mitarbeiter für das Team zu gewinnen
  - C) Nur Bewerber zu berücksichtigen, die das Gefühl haben, dass sie nicht voll und ganz zu der Organisation oder den anderen Teammitgliedern passen, weil dies Diversität am stärksten fördert
  - D) Den Teammitgliedern zu erlauben, das Teammitglied auszuwählen, mit dem sie sich am stärksten identifizieren, weil dies für Harmonie und Gleichheit im Team sorgt
- 
- A) Richtig. Dafür zu sorgen, dass Einstellungsverfahren möglichst frei von Vorurteilen sind und Diversität aktiv den Vorzug zu geben, macht das DevOps-Team smarter und hilft ihm, bessere Leistungen zu erbringen. (Literatur: A, Kapitel 10)
  - B) Falsch. Diversität ist gut für DevOps-Teams. Es macht sie smarter und hilft ihnen, bessere Leistung zu erbringen. Wirbt man Mitarbeiter von anderen Unternehmen ab, erhält man zwar Kompetenz, nicht aber Diversität oder Zugehörigkeitsgefühl.
  - C) Falsch. Diversität ist gut für DevOps-Teams. Es macht sie smarter und hilft ihnen, bessere Leistung zu erbringen. Diversität alleine ist jedoch nicht genug; damit DevOps erfolgreich ist, müssen sich die Mitarbeiter auch zugehörig fühlen. Stellt man Mitarbeiter ein, die das Gefühl haben nicht dazuzugehören, erreicht man genau das Gegenteil.
  - D) Falsch. Diversität ist gut für das DevOps Team. Es macht die Teams smarter und hilft ihnen, bessere Leistung zu erbringen. Ähneln sich die eingestellten Teammitglieder stark, erreicht man das Gegenteil von Diversität.

**35 / 40**

Eine Organisation hat seine Managementpraktiken auf Lean-Leadership umgestellt. Das ist eine große Veränderung gegenüber der früheren Arbeitsweise der Organisation. Früher hat das Management die Kollegen regelmäßig öffentlich für ihre Fehler bloßgestellt.

Eines der Teams ist ein verteiltes Team. Das Team ist mit dem neuen Führungsstil zwar viel glücklicher, aber seine Kommunikation ist nicht optimal. Die Teammitglieder diskutieren nicht über ihre Software-Bugs und arbeiten auch nicht zusammen, um Probleme schnell zu beheben.

Der Vorgesetzte des Teams hat deshalb die Unternehmensversion eines Chat-Tools (Chat-Hilfsmittels) eingeführt, das die meisten Teammitglieder bereits privat nutzen. Aber auch diese Maßnahme scheint nicht den gewünschten Erfolg zu haben.

Was passiert in diesem Team und wie sollte der Manager das lösen?

- A)** - Die Teammitglieder verfügen noch nicht über die erforderliche Kompetenz im Umgang mit dem Chat-Tool.  
- Der Manager sollte interne oder externe Schulungen ermöglichen, um dieses Problem zu lösen.
  - B)** - Die Teammitglieder sind noch in der Umstellung vom früheren auf den neuen Führungsstil.  
- Hierbei handelt es sich um einen vorübergehenden Prozess, in den der Manager nicht eingreifen muss.
  - C)** - Die Teammitglieder vertrauen dem neuen Führungsstil noch nicht.  
- Der Manager muss hier überlegt vorgehen und seinen Stolz über das Team zum Ausdruck bringen.
- 
- A)** Falsch. Das Team nutzt das gleiche Tool auch privat. Es ist daher unwahrscheinlich, dass dies der Grund für die mangelnde Kommunikation ist. Da das Team nicht über Bugs und Probleme spricht, ist es wahrscheinlicher, dass es dem neuen Führungsstil noch nicht traut.
  - B)** Falsch. Das Team ist mit dem neuen Führungsstil glücklicher. Es ist daher unwahrscheinlich, dass dies der Grund für die mangelnde Kommunikation ist. Da das Team nicht über Bugs und Probleme spricht, ist es wahrscheinlicher, dass es dem neuen Führungsstil noch nicht traut.
  - C)** Richtig. Da das Team nicht über Bugs und Probleme spricht, ist es am wahrscheinlichsten, dass es dem neuen Führungsstil noch nicht traut. Der Teamleiter sollten auf die Führungsphilosophie des Supportive Leadership (unterstützende Führung) sowie motivierende Kommunikation setzen, um das Problem zu lösen. (Literatur: A, Kapitel 11)

36 / 40

Was ist der **wichtigste** unmittelbare Vorteil, den man erhält, wenn man dafür sorgt, dass die Teammitglieder eine sinnstiftende Arbeit haben?

- A) Es trägt dazu bei, die Anzahl von Burnouts zu verringern und Mitarbeiterloyalität und -engagement zu steigern.
  - B) Es trägt dazu bei, alle DevOps-Projekte des Teams pünktlich und im Rahmen des Budgets zu liefern.
  - C) Es trägt dazu bei, die Überzeugung des Teams zu stärken, dass es die DevOps-Arbeit bewältigen kann.
  - D) Es trägt dazu bei, die Teammitglieder an die Organisation zu binden, weil diese ihren Job nicht aufgeben wollen.
- 
- A) Richtig. Der größte Vorteil besteht darin, dass die Anzahl von Burnouts gesenkt und das Engagement der Mitarbeiter gesteigert wird. (Literatur: A, Kapitel 10)
  - B) Falsch. Dies kann zwar eine indirekte Folge sein, ist aber kein direkter Vorteil von sinnstiftender Arbeit. Der größte Vorteil besteht darin, dass die Anzahl von Burnouts gesenkt und das Engagement der Mitarbeiter gesteigert wird.
  - C) Falsch. Das Team mag sich zwar kompetent fühlen, das bedeutet aber nicht, dass es auch tatsächlich die Kompetenz hat, etwas fertigzustellen. Außerdem führt sinnstiftender Arbeit nicht notwendigerweise zu mehr Selbstvertrauen. Der größte Vorteil besteht darin, dass die Anzahl von Burnouts gesenkt und das Engagement der Mitarbeiter gesteigert wird.
  - D) Falsch. Die Loyalität der Mitarbeiter mag zwar steigen, aber anzunehmen, dass keiner seinen Arbeitsplatz wechseln möchte, ist außerordentlich optimistisch. Der größte Vorteil besteht darin, dass die Anzahl von Burnouts gesenkt und das Engagement der Mitarbeiter gesteigert wird.

**37 / 40**

Ein Unternehmen ist in den letzten beiden Jahren exponentiell gewachsen. Das Entwicklungsteam steht infolgedessen unter Zeitdruck. Alle Teammitglieder fühlen sich urlaubsreif und manche denken sogar an Kündigung. Die Krankheitstage im Team sind gegenüber den Vorjahren gestiegen.

Der Teammanager hat mehrere neue Teammitglieder eingestellt, aber die Arbeitsbelastung wird immer noch als hoch empfunden, obwohl anscheinend alle Fristen pünktlich eingehalten werden. Um das Problem zu lösen, möchte sich der Teammanager direkter mit den Beschwerden zu Burnout und Stress befassen.

Welche Maßnahmen eignen sich dazu **am besten**?

- A)** - Unterauftragnehmer beauftragen, um die höhere Arbeitsbelastung in Spitzenzeiten zu bewältigen  
- Den Teammitgliedern Zugriff auf professionelle psychologische Hilfsprogramme anzubieten, die vom Unternehmen finanziert werden
  - B)** - Allen Teammitgliedern einen Monat Urlaub geben, um einen Neuanfang zu gewährleisten, wenn sie wieder an ihren Arbeitsplatz zurückkehren  
- Die Manager bitten, bevor das Team wieder an seinen Arbeitsplatz zurückkehrt, alle Faktoren zu identifizieren, die möglicherweise zur Burnout-Problematik beigetragen haben
  - C)** - Dafür sorgen, dass die Teammitglieder alle Aufgaben im Prozess besprechen, um herauszufinden, welche Aufgaben sinnstiftend sind  
- Nicht sinnstiftende Aufgaben sollten gestrichen und sichergestellt werden, dass das Team die meisten Entscheidungen selbstständig fällen kann
  - D)** - Einen kompetenteren Manager einstellen, der weiß, wie man die Work-Life-Balance der Teammitglieder sicherstellt  
- Den Teammitgliedern mehr Zeit geben, damit sie sich in ihre neue Umgebung eingewöhnen und ein neues Gleichgewicht finden können
- 
- A)** Falsch. Dies scheint zwar die Probleme des Teams zu lösen, aber diese Lösung befasst sich nicht direkt mit den Beschwerden aufgrund von Stress und Burnout. Sie schafft lediglich vorübergehende Abhilfe und geht nicht auf die Ursachen ein.
  - B)** Falsch. Zwar wären alle Mitglieder des Teams zweifelsohne glücklich, einen Monat Urlaub zu bekommen, aber für das Unternehmen wäre diese Maßnahme äußerst schädlich. Außerdem sind die Manager nicht unbedingt diejenigen, die dieses Problem am besten lösen können.
  - C)** Richtig. Sicherzustellen, dass die Teammitglieder genügend Freiraum haben, um bei ihrer Arbeit selbst Entscheidungen treffen zu können, und hauptsächlich sinnstiftende oder zumindest notwendige Aufgaben ausführen, ist langfristig die beste Lösung gegen Burnout-Beschwerden und Stress. Beide Maßnahmen steigern Mitarbeiterengagement und -zufriedenheit. (Literatur: A, Kapitel 11 und Anhang A)
  - D)** Falsch. Der Manager des Teams hat das Problem korrekt erfasst und scheint durchaus kompetent zu sein. Einen neuen Manager einzustellen und dem Team mehr Zeit zu geben ist keine Lösung, die auf die Probleme des Teams eingeht.

38 / 40

Transformationale Führung zählt angeblich zu den Aspekten von DevOps, die am häufigsten übersehen werden.

Warum ist transformationale Führung für den unternehmensweiten Erfolg von DevOps wichtig?

- A) Weil man mit Commitment des Top-Managements Prozesse verändern und so potenziellem Widerstand entgegenwirken kann
  - B) Weil DevOps scheitern wird, wenn nicht alle Mitarbeiter, auch das Top-Management, DevOps gut kennen und verstehen
  - C) Weil die für Continuous Delivery (kontinuierliche Lieferung) notwendigen Änderungen vom Top-Management angeordnet werden müssen
  - D) Weil das Top-Management Vertrauen in seine Mitarbeiter haben und diese ermutigen muss, zu experimentieren, ohne sich vor Strafen fürchten zu müssen
- 
- A) Falsch. Commitment seitens der Unternehmensführung sind zwar zu begrüßen, sollten aber keine Prozesse verändern. Die Unternehmensführung sollte stattdessen für eine generative Kultur sorgen, in der die Mitarbeiter die notwendigen Veränderungen vornehmen können.
  - B) Falsch. Es ist zwar wichtig, DevOps zu verstehen, aber die Unternehmensführung muss lediglich eine generative Kultur schaffen, in der Mitarbeiter die entsprechenden Veränderungen vornehmen dürfen, damit DevOps funktioniert.
  - C) Falsch. Die Unternehmensführung kann Veränderungen zwar anordnen, aber das ist kein erfolgsversprechender Weg für die Transformation der Organisation oder um die Unterstützung der Mitarbeiter für die Veränderungen sicherzustellen. Die Unternehmensführung sollte stattdessen besser für eine generative Kultur sorgen, in der die Mitarbeiter die notwendigen Veränderungen vornehmen können.
  - D) Richtig. Eine Unternehmensführung, die die Transformation unterstützt, sorgt für eine generative, von großem Vertrauen geprägte Unternehmenskultur, in der DevOps gedeihen kann. (Literatur: A, Kapitel 11)

**39 / 40**

Das kontinuierliche Monitoring des DevOps-Reifegrads in einer Organisation ist eine entscheidende Fähigkeit. Sie ermöglicht die fachübergreifende Zusammenarbeit in der IT und verringert oder beseitigt unproduktive Verhaltensweisen, die man in Organisationen mit funktionalen Silos noch viel zu häufig vorfindet. Das kontinuierliche Monitoring unterscheidet sich vom herkömmlichen Monitoring.

Was ist die **wichtigste** Anforderung an das kontinuierliche Monitoring?

- A) Eine Komponente vollständig automatisiert auf den gewünschten Zustand zu bringen
  - B) Die Zusammenarbeit von Entwicklung und Betrieb in jedem Wertstrom zu erreichen
  - C) Die Organisation auf eine Linie zu bringen und die Produktionsumgebung in den Vordergrund zu rücken
  - D) Eine gesamtheitliche Sicht auf die komplexe Anwendungsumgebung der Organisation zu haben
- 
- A) Falsch. Continuous Deployment (kontinuierliche Bereitstellung) erfordert, dass eine Komponente immer voll automatisiert in den gewünschten Zustand gebracht werden kann, unabhängig von ihrem Anfangszustand oder davon, wie oft die Komponente konfiguriert wird.
  - B) Falsch. Damit man dem Kunden Wert liefern kann, müssen Entwicklung und Betrieb in Wertströmen zusammenarbeiten und gemeinsame Ziele und Praktiken verfolgen.
  - C) Falsch. Herkömmliches Monitoring erfordert es, die Organisation auf eine Linie zu bringen und konzentriert sich nur auf die Produktionsumgebung.
  - D) Richtig. Kontinuierliches Monitoring erfordert einen ganzheitlichen Blick auf die komplexe Anwendungsumgebung der gesamten Organisation. (Literatur: C, Kapitel 3)

40 / 40

Betrachtet man den DevOps-Reifegrad einer Organisation, so ist eine mehrdimensionale Perspektive der Organisation, ihrer Fähigkeiten und ihres Reifegrads wichtig. Die kontinuierliche Bewertungsarchitektur von DevOps lässt sich in Form eines Würfels darstellen.

Was repräsentieren die Seiten des „DevOps-Würfels“?

- A) Flow, Feedback, Lernen und Experimentieren, Governance, Deployment (Bereitstellung) und Qualitätssicherung
  - B) Monitoring, Information, Kommunikation, Bewertung, Kontrolle und Aktivitäten
  - C) Menschen, Prozesse, Partner, Technologie, Qualitätssicherung und Kultur
  - D) Anforderungen, Prozesse, Ressourcen, Organisation, Reifegrad und Fähigkeit
- 
- A) Richtig. Der DevOps-Würfel umfasst sechs Dimensionen, die kontinuierlich bewertet werden müssen, will man den Kurs der DevOps-Bemühungen des Unternehmens korrigieren. Die ersten drei repräsentieren die Drei Wege von DevOps, die anderen drei stehen für durchgehende Continuous Deployment (kontinuierliche Bereitstellung), Qualitätssicherung und ordnungsgemäße Governance. (Literatur: C, Kapitel 5)
  - B) Falsch. Die hier beschriebenen Dimensionen stammen aus einer der ersten Anwendungen von Würfelmodellen zur Darstellung der multidimensionalen Aspekte von Frameworks für die Bewertung und Steuerung. Ausgangspunkt für viele Frameworks, die Würfelmodelle nutzen, war das Committee of Sponsoring Organizations (COSO).
  - C) Falsch. Mitarbeiter, Prozesse, Partner und Technologiedimensionen werden als wichtige Teile des Microsoft Operations Framework (MOF) und Microsoft Solutions Framework (MSF) häufig genutzt, um einen ausgeglichenen Ansatz für das Management von Technologieumgebungen sicherzustellen.
  - D) Falsch. (Business-)Anforderungen, Prozesse und Ressourcen sind die drei Domänen eines anderen bekannten Würfelmodells. Es handelt sich hierbei um die drei Hauptdomänen, die in älteren COBIT (Control Objectives for Information and Related Technologies)-Versionen eingesetzt werden, um für einen ausgeglichenen Governance-Ansatz zu sorgen. COBIT ging zwar auch auf die Organisationsgestaltung/Rollen sowie den Reifegrad der Fähigkeiten ein, aber diese Dimensionen waren auf dem COBIT-Würfel nicht präsentiert.



# Beurteilung

Die richtigen Antworten auf die Fragen in dieser Musterprüfung finden Sie in nachstehender Tabelle.

Frage	Antwort	Frage	Antwort
1	D	21	B
2	B	22	D
3	B	23	C
4	B	24	B
5	B	25	B
6	D	26	D
7	A	27	C
8	D	28	D
9	C	29	D
10	A	30	B
11	A	31	C
12	B	32	C
13	A	33	C
14	C	34	A
15	A	35	C
16	B	36	A
17	D	37	C
18	C	38	D
19	B	39	D
20	D	40	A



Driving Professional Growth

**Kontakt EXIN**

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