



Musterprüfung

Ausgabe 202606



Copyright © EXIN Holding B.V. 2026. All rights reserved.
EXIN® is a registered trademark of EXIN.
DevOps Master™ is a registered trademark of EXIN.

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	15
Beurteilung	39

Einführung

Dies ist die EXIN DevOps Foundation (DEVOPSF.DE) Musterprüfung. Es gilt die Prüfungsordnung von EXIN.

Die Musterprüfung besteht aus 40 Multiple-Choice-Fragen. Zu jeder Multiple-Choice-Frage werden mehrere Antwortmöglichkeiten angeboten. Ist nichts anderes angegeben, so gibt es 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 60 Minuten.

Viel Erfolg!

Musterprüfung

1 / 40

Welche Aussage passt **nicht** zu Agile?

- A) Unternehmen und Entwicklung arbeiten während des gesamten Projekts zusammen.
- B) Auf eine Änderung zu reagieren ist wichtiger als sich an einen Plan zu halten.
- C) Den Kunden durch die präzise Erfüllung der Anforderungen zufriedenzustellen ist eine der Prioritäten.
- D) Funktionierende Software ist die primäre Messgröße für den Erfolg.

2 / 40

Was ist laut der US-Bundesbehörde National Institute of Standards and Technology (NIST), **kein** wesentliches Merkmal von Cloud Computing?

- A) Weitreichender Netzzugang
- B) Nutzungsabhängiges Gebührensystem
- C) Schnelle Elastizität
- D) Ressourcenbündelung

3 / 40

Die IT-Arbeit innerhalb von DevOps ist anders organisiert als bei traditionelleren Vorgehensweisen. Um diese Umstellung zu ermöglichen, sind mehrere Infrastrukturmanagement-Technologien erforderlich.

Welche sind **zwei** dieser Infrastrukturmanagement-Technologien?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Digitale Kanban-Tafeln
- B) Infrastructure as a Service (IaaS)
- C) Low-Code-/No-Code-Plattformen
- D) Plattform für Testautomatisierung
- E) Virtualisierung (Hypervisoren)

4 / 40

DevOps baut stark auf den Prinzipien und Praktiken von Lean Production auf. Zu den Verschwendungsarten in der IT zählt auch das 'Task Switching'.

Welcher Verschwendungsart in Lean Production entspricht das 'Task Switching' in der IT?

- A) Bewegung
- B) Überproduktion
- C) Verkehr
- D) Wartezeiten

5 / 40

Die wertvollsten Informationen einer Wertstromanalyse stammen aus drei Schlüsselkennzahlen.

Welche **drei** Schlüsselkennzahlen sind gemeint?

Bitte denken Sie daran, 3 Antworten auszuwählen.

- A) Flow
- B) Durchlaufzeit
- C) C/A-Anteil, das heißt der Anteil der Leistungen die vollständig und richtig erbracht wurden
- D) Prozesszeit
- E) Prozesszeit geteilt durch Durchlaufzeit
- F) Fertiggestellter Wert minus Verschwendung

6 / 40

Von der Umstellung auf DevOps erwartet man sich eine höhere IT-Rentabilität.

Inwiefern unterscheidet sich DevOps bezüglich der Erfüllung dieser Erwartung von Agile, Scrum und Lean?

- A) DevOps beschleunigt die Lieferung neuer und geänderter Produkte zwischen Entwicklung und Betrieb.
- B) DevOps beschleunigt die Lieferung neuer und geänderter Produkte an Markt und Kunden.
- C) DevOps sorgt für schnellere Reaktionen auf Änderungen der IT-Infrastruktur innerhalb des vereinbarten Kostenrahmens.
- D) DevOps sorgt für schnellere Reaktionen auf Zwischenfälle (Incidents) die den Wertstrom der Organisation behindern.

7 / 40

Eine allgemein anerkannte Definition des Begriffs „Time-to-Market“ lautet:

Die Zeit zwischen der Geschäftsidee und dem Zeitpunkt, zu dem ein Kunde das Produkt erwerben kann.

Welche **zwei** DevOps-Praktiken tragen am **meisten** dazu bei, die Time-to-Market zu verkürzen?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Automatisiertes Testen
- B) Zentralisierung der Entscheidungsfindung
- C) Lose gekoppelte Architektur
- D) Verwendung von Kanban-Boards
- E) Wertstromanalyse (VSM)

8 / 40

Welche **zwei** gängigen Praktiken reduzieren technische Schulden **am direktesten**?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Die Praktiken des Change Managements
- B) Die Praktiken des Incident Managements
- C) Die Erhöhung von Ressourcen
- D) Sich dem Problem stellen
- E) Refactoring
- F) Die Praktiken des Release Managements

9 / 40

Welchen großen Vorteil bietet DevOps einer Organisation?

- A) DevOps eliminiert fragile Unternehmenssysteme.
- B) DevOps verringert die Kosten für den Kunden.
- C) DevOps reduziert kulturelle Schwierigkeiten.

10 / 40

DevOps wird manchmal für einen Teil von Agile gehalten.

Auf welche Frage, die mit Agile überhaupt nichts zu tun hat, gibt DevOps eine Antwort?

- A) Was kann man tun, um das Release der von uns entwickelten Software so zu beschleunigen, dass wir unseren Marktanteil vergrößern?
- B) Welche Art von Kundenbeziehung brauchen wir, um die Anforderungen unserer Kunden besser zu verstehen?
- C) Was sollte nach dem Release mit einem Produkt getan werden, damit es nützlich und einfach zu betreiben ist?

11 / 40

Why is the concept 'value stream' so important in DevOps?

- A) The value stream helps to analyze the as-is map and any attempts to improve metrics.
- B) The value stream helps to realize a smooth and uniform flow through all process steps.
- C) The value stream helps to realize local optimization in the current work practice.

12 / 40

Was ist bei der Wertstromanalyse der **erste** Schritt?

- A) Die Erstellung der Anforderungsspezifikation
- B) Die Dokumentation der fertiggestellten Arbeit
- C) Die Identifizierung der wichtigsten Prozessschritte

13 / 40

Eine Wertstromanalyse zeigt deutlich, wo sich im Wertstrom Ineffizienzen verbergen.

Warum sollte eine Wertstromanalyse durchgeführt werden?

- A) Um die Verbesserung der Unternehmensprozesse zu unterstützen
- B) Um zu sehen, welche Mitarbeiter nicht hart genug arbeiten
- C) Um Work-in-Progress (WIP) zu beschleunigen
- D) Um zu visualisieren, welche Produktlinie eingestellt werden sollte

14 / 40

Aufgaben müssen priorisiert werden. Die Priorisierung erfolgt in der Queue (Warteschlange) vor Eintritt in den Wertstrom.

Warum verursacht diese Priorisierung häufig Probleme?

- A) Weil an diesem Punkt die Entscheidung über den Bau der Deployment Pipeline für die automatisierbaren Aufgaben gefällt wird. Dies ist zeitaufwändig und führt zu Verzögerungen.
- B) Weil die Befähigung zur Messung der wichtigsten Kennzahlen im Wertstrom falsch oder ineffizient erfolgt und daher Probleme verursacht.
- C) Weil die Nutzung von Visualisierungstools in Kombination mit WIP-Begrenzungen (WIP-Limits), durch die sich Engpässe bei der Aufgabe identifizieren lassen, falsch umgesetzt wird.
- D) Weil sowohl die Ist-Version als auch die Soll-Version des Wertstroms und eine Liste der erforderlichen Änderungen erstellt werden müssen und dies nimmt viel Zeit in Anspruch.
- E) Weil man, falls nach wie vor der traditionelle Ansatz verwendet wird, noch vor Beginn der eigentlichen Arbeit viele Entscheidungen treffen muss, was massive Verzögerungen zur Folge hat.

15 / 40

Welches Bild liegt dem Begriff 'Deployment Pipeline' zugrunde?

- A) Eine Pipeline mit Flüssigkeit, die für Flow sorgt
- B) Ein Montageband wie zum Beispiel in einer Autofabrik
- C) Moderne Prozessoren, die paralleles Pipelining nutzen
- D) Die Idee mehr als ein Montageband einzusetzen

16 / 40

Bei der Implementierung der Deployment Pipeline kommt es zu Problemen. Die Zahl der vorab entwickelten Tests reicht anfänglich nicht aus, um einen konstanten Betrieb der Produktionsumgebung sicherzustellen.

Was ist die **beste** Lösung für dieses Problem?

- A) Die Pipeline und so viele automatisierte Prozesse wie möglich sollten erstellt, aber nicht genutzt werden, bis alle ordnungsgemäßen Tests vorhanden sind.
- B) Die Abdeckung des Codes sollte durch Tests gesteigert werden und muss ähnlich wie technische Schulden schnellstmöglich angegangen werden.
- C) Die Pipeline sollte mit den entwickelten Tests laufen und potenzielle Probleme in der Produktion sollten gewidmet werden, wenn diese auftreten.
- D) Die Pipeline sollte lediglich als Integrationssystem genutzt werden, um den erstellten Code an die für das Testen und die Qualitätssicherung (QA) zuständige Abteilung zu liefern.

17 / 40

Ein gutes System zur Versionskontrolle ist bei DevOps einer der besten Prädiktoren für hohe Leistungsfähigkeit.

Was ist für eine erfolgreiche Versionskontrolle nötig?

- A) Ein Kulturwandel der Arbeit mit Informationen und Konfigurationen
- B) Eine signifikant schnellere Umsetzung von Änderungen
- C) Die absichtliche Einbringung von Chaos und Instabilität in die Produktion
- D) Die Nutzung eines förmlichen und automatisierten Change-Management-Prozesses

18 / 40

Wichtig bei DevOps ist ein ausgewogenes Verhältnis zwischen einer schnellen Reaktion und der Aufrechterhaltung der Zuverlässigkeit der Anwendung.

Inwiefern wird dies durch die Versionskontrolle unterstützt?

- A) Indem sie es jedem Mitglied des Teams ermöglicht, unnötige Dateien oder Dokumente ohne Einschränkung zu löschen
- B) Indem sie die Bildung kleiner, unabhängiger und autarker Entwicklungsteams ermöglicht
- C) Indem sie Spezialwerkzeuge zur Verringerung oder Eliminierung von Verschwendung und zur Verbesserung von Prozessen einsetzt

19 / 40

Welche Vorteile bietet das Konfigurationsmanagement?

- A) Es ermöglicht allen Teammitgliedern das risikofreie Löschen unerwünschter Dateien.
- B) Es ermöglicht den wichtigsten Team-Mitgliedern die problemfreie Abwesenheit.
- C) Es ermöglicht den Teammitgliedern zu sehen, wer wann welchen Code geändert hat.

20 / 40

Das Konfigurationsmanagement ermöglicht die Skalierung der IT-Infrastruktur und Softwaresysteme ohne erhöhten Personalaufwand.

Wie sollten Änderungen in einer solchen skalierten Umgebung **am besten** durchgeführt werden?

- A) Mittels Continuous Integration (kontinuierlicher Integration)
- B) Mittels vollständig gelenkter Skripten
- C) Mittels Testautomatisierung
- D) Mittels der Deployment Pipeline

21 / 40

Eine eindeutige Definition of Done (DoD, Definition von 'Fertiggestellt') ist bei DevOps von entscheidender Bedeutung und berücksichtigt den Wert für den Kunden.

Wie lautet bei DevOps eine gute Beschreibung für Done?

- A) Eine Anforderung gilt als Done, nachdem sie gebaut wurde.
- B) Eine Anforderung gilt als Done, nachdem sie getestet wurde.
- C) Eine Anforderung gilt als Done, nachdem das Produkt abgenommen wurde.
- D) Eine Anforderung gilt als Done, nachdem das Produkt in der Produktion ist.

22 / 40

Nutzt man die traditionellen Praktiken, so kann bei einem Release Einiges schief gehen: Manche Änderungen werden nicht dokumentiert, es gibt kein vollständiges Backup des Systems und der Zustand des Systems vor dem Release wurde nicht gespeichert.

Wie sorgt DevOps für häufige Releases ohne diese Probleme?

- A) Indem Releases automatisiert werden
- B) Indem der Betrieb die Releases durchführt
- C) Indem Releases sehr klein gemacht werden
- D) Indem nicht alle Änderungen dokumentiert werden

23 / 40

Ein Unternehmen nutzt Continuous Deployment (kontinuierliche Bereitstellung).

Wer sollte entscheiden, wann das Release der neuen Funktionalität erfolgt?

- A) Das Unternehmen
- B) Der Kunde
- C) Die IT-Abteilung
- D) Der Benutzer

24 / 40

Was ist laut DevOps-Praktiken der **beste** Weg, um den Grad der Betriebslenkung zu steigern?

- A) Die Automatisierung aller manuellen Betriebsvorgänge
- B) Die Definition geeigneter Rollen und Zuständigkeiten
- C) Die Entwicklung von Lenkungsverfahren
- D) Die Verbesserung der Betriebsführung

25 / 40

Wie löst DevOps Zwischenfälle (Incidents)?

- A) DevOps eskaliert Zwischenfälle an das Problem Management-Team und stellt eine Lösung bereit, bis das Team den Zwischenfall behebt.
- B) DevOps untersucht den Zwischenfall, führt eine Diagnose durch, findet dann einen Workaround (Umgehungslösung) und setzt diesen um.
- C) DevOps prüft, ob ein ähnlicher Zwischenfall schon einmal aufgetreten ist und setzt dann eine ähnliche Lösung für das Problem um.
- D) DevOps verfolgt den Vorfall zurück auf ein vor kurzem stattgefundenes Deployment und führt dann ein Roll-Back des Systems auf den vormals stabilen Zustand durch.

26 / 40

Welche Maßnahmen sollten bei DevOps ergriffen werden, wenn man Prozessmängel feststellt?

- A) Alle Änderungen sollten in Form eines Backlogs eingereicht werden, damit sie im Rahmen eines Projekts oder einer Kaizen-Veranstaltung freigegeben werden können.
- B) Möglichkeiten der Fehlerbehebung müssen ermittelt und nach Entdecken der Mängel schnellstmöglich umgesetzt werden.
- C) Möglichkeiten der Fehlerbehebung müssen ermittelt, vom Change Manager genehmigt und entsprechend ihrer Priorität freigegeben werden.
- D) Möglichkeiten der Fehlerbehebung müssen ermittelt, vom Continuous Improvement Manager genehmigt und umgehend freigegeben werden.
- E) Möglichkeiten der Fehlerbehebung sollten aufgeschoben werden, bis sie in einen geeigneten Sprint integriert werden können.

27 / 40

Was unterstützt ein DevOps-Team **nicht** bei der erfolgreichen Entwicklung und Lieferung einer funktionierenden Software?

- A) Die Bildung eines DevOps-Teams für eine kurze Zeit während eines Projekts
- B) Die Fehler erkennen, beheben und daraus lernen, sobald sie gefunden werden
- C) Die Organisation der DevOps-Teams rund um eine organisatorische Aufgabe
- D) Die Erstellung von Code für eine Software, bei der das Hauptziel die Integration von Qualität ist

28 / 40

DevOps empfiehlt kontinuierliche Aktualisierungen der fertiggestellten Arbeit.

Wie hält man alle Teammitglieder **am besten** kontinuierlich über den Fortschritt eines Flows auf dem Laufenden?

- A) Indem man es im Laufe des Tages mit Teammitgliedern bespricht
- B) Indem man den Fortschritt beim täglichen Stand-up-Meeting bespricht
- C) Indem man den Product Owner persönlich informiert
- D) Indem man dem Scrum Master Aktualisierungsmails schickt
- E) Indem man Visualisierung, wie zum Beispiel ein Kanban-Board, verwendet

29 / 40

Ein Team fügt Elemente aus dem Backlog zu einem Batch hinzu.

Welche **zwei** Dinge müssen berücksichtigt werden?

Bitte denken Sie daran, dass Sie 2 Antworten wählen müssen.

- A) Die zur Erledigung des Backlog-Elements erforderliche Kapazität
- B) Die aktuelle Höhe der technischen Schulden
- C) Die gesamte Pipeline, von Anfang bis Ende
- D) Die bevorzugten Besprechungszeiten der Teammitglieder
- E) Die Urlaubszeiten der Teammitglieder

30 / 40

Auf welche **zwei** Arten berücksichtigt DevOps betriebliche Anforderungen?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Durch die Erstellung einer Definition of Done (DoD, Definition von „Fertiggestellt“)
- B) Durch die Erstellung einer Wertstromanalyse (VSM)
- C) Durch den Einsatz von Agile- und Scrum-Prozessen
- D) Durch die Nutzung von Geschäftsanforderungen als Grundlage
- E) Durch das Festlegen von Anforderungen im Voraus

31 / 40

Welchen Vorteil hat es, wenn ein DevOps-Team über einen längeren Zeitraum zusammenarbeitet?

- A) Das Team muss den Prozess nicht mehr verbessern.
- B) Das Team nutzt seine Erfahrung zur Innovation und Prozessverbesserung.
- C) Das Team beginnt, unabhängiger zu arbeiten.
- D) Das Team hat mehr Zeit für die häufigere Bearbeitung unerwarteter Anträge.

32 / 40

Ein Team arbeitet in einwöchigen Iterationen und stößt dabei häufig auf Engpässe.

Wie sollte das Team **am besten** reagieren, wenn ein Engpass identifiziert wird?

- A) Das Team sollte den Engpass baldmöglichst beseitigen, nachdem er identifiziert wurde.
- B) Das Team sollte nur die Iteration, in der ein Engpass festgestellt wurde, verlängern.
- C) Das Team sollte die reguläre Anzahl von Aufgaben im Batch begrenzen, um die Batchgröße zu reduzieren.
- D) Das Team sollte Visualisierungstools in Kombination mit einer WIP-Begrenzung (WIP-Limit) einsetzen.

33 / 40

Wann kann die Nutzung von DevOps bei organisatorischen und technischen Änderungen zu Chaos und Kontrollverlust führen?

- A) Wenn das Kerngeschäft des Unternehmens in hohem Maße von der Informationstechnologie abhängt
- B) Wenn komplexe Organisationen, chronische Probleme lösen wollen
- C) Wenn die Organisation schnelle Änderungen braucht, um neue Geschäftsideen zu testen
- D) Wenn sich die von der Organisation genutzte Informationstechnologie schnell ändert

34 / 40

Es gibt viele Gründe, warum sich eine Organisation für DevOps interessieren sollte.

Wann sollten Unternehmen damit beginnen, sich für DevOps zu interessieren?

- A) Wenn Agile-Praktiken nicht mehr zur Organisation zu passen scheinen
- B) Wenn die notwendigen Ergebnisse sich mit keiner anderen Methode erreichen lassen
- C) Wenn Scrum- und Lean-Praktiken umgesetzt sind

35 / 40

Was kann bei der Einführung von DevOps zu Schwierigkeiten führen?

- A) Funktionsübergreifende Teams
- B) Begrenzte Nutzung von Virtualisierung
- C) Microservice-Architektur

36 / 40

Ein IT-System wird aktuell noch von vielen Mitarbeitern als Ganzes entwickelt und gepflegt.

Wo sind Schwierigkeiten bei der Einführung von DevOps-Praktiken zu erwarten?

- A) Bei der Zuweisung der DevOps-Teams zu getrennten Zuständigkeitsbereichen
- B) Bei der Schaffung funktionsübergreifender Teams zur Arbeit an organisatorischen Strukturen
- C) Bei der Pflege und Versionierung mehrerer Application Programming Interfaces (APIs) zur Gewährleistung der Abwärtskompatibilität

37 / 40

Handelsübliche Standardsoftware (COTS) wird eingesetzt, um schnell die erforderlichen Ergebnisse zu erzielen, da die Entwicklung individueller Software Zeit in Anspruch nimmt.

Welche Aussage zu COTS ist **korrekt**?

- A) COTS beseitigt in der Regel die Integrationskomplexität.
- B) COTS senkt die Gesamtbetriebskosten für das Unternehmen.
- C) COTS erfordert Anpassungen, um das System zu konfigurieren.
- D) COTS sollte zur Unterstützung strategischer Geschäftsbereiche eingesetzt werden.

38 / 40

Was ist eine der Schwierigkeiten einer starren oder monolithischen IT-Architektur?

- A) Die Änderung und Entwicklung der IT-Architektur selbst ist schwierig zu bewerkstelligen.
- B) Die Änderung der Services innerhalb der Architektur erfolgt unabhängig.
- C) Die Aktualisierung auf eine neue Version ohne Deaktivierung der aktuellen Version.
- D) Das Warten bis alle Komponenten für eine große Migration bereit sind.

39 / 40

Laut Empfehlung sollen Organisationen DevOps-Praktiken auswählen, die bei ihnen funktionieren und diese dann an ihre Bedürfnisse anpassen. Zu diesem Zweck müssen organisationsspezifische Fragen gestellt und organisationsspezifische Antworten gefunden werden.

Warum ist dies eine gute Idee?

- A) Weil DevOps-Veröffentlichungen nicht immer der Realität entsprechen und zu wenig über Schwierigkeiten und Fehler berichten
- B) Weil dies die beste Möglichkeit ist, um ein DevOps-Ingenieur zu werden, der für die Umsetzung von DevOps eingestellt wird
- C) Weil die Managementteams am besten wissen, wie sie DevOps in der eigenen Organisation am besten implementieren können.
- D) Weil es zu wenige Veröffentlichungen und Veranstaltungen zu DevOps gibt, um sich unabhängige Meinungen zu bilden.

40 / 40

Eine Organisation hat eine alte IT-Infrastruktur. Die Organisation möchte DevOps einführen.

Welcher Ansatz wird in einer solchen Organisation häufig gewählt?

- A) Man beginnt die Implementierung von DevOps als Softwareprodukt, installiert es und startet es.
- B) Man beginnt die Implementierung mit einer grundlegenden Pipeline, die zumindest den Bau und erste Tests durchführt.
- C) Man beginnt die Implementierung mit der Auswahl des Produkts, das die größten Möglichkeiten zur Optimierung bietet.
- D) Man beginnt die Implementierung mit der Identifizierung der Systeme, die lose mit anderen gekoppelt sind.
- E) Man beginnt die Implementierung, indem man einen bestimmten Teil der Arbeitszeit für die Verbesserung zuteilt.

Antwortschlüssel

1 / 40

Welche Aussage passt **nicht** zu Agile?

- A) Unternehmen und Entwicklung arbeiten während des gesamten Projekts zusammen.
- B) Auf eine Änderung zu reagieren ist wichtiger als sich an einen Plan zu halten.
- C) Den Kunden durch die präzise Erfüllung der Anforderungen zufriedenzustellen ist eine der Prioritäten.
- D) Funktionierende Software ist die primäre Messgröße für den Erfolg.

- A) Falsch. Laut dem Agilen Manifest müssen die Mitarbeiter des Unternehmens und die Entwickler bei einem Projekt täglich zusammenarbeiten.
- B) Falsch. Die Aussage, dass die Reaktion auf eine Änderung wichtiger ist als die Einhaltung eines Plans findet sich im Agilen Manifest.
- C) Richtig. Die Kunden durch präzise Erfüllung ihrer Anforderungen zufriedenzustellen ist nicht Teil des Agilen Manifests und zählt nicht zu den Zielen von Agile. (Literatur: A, Kapitel 1.1.1)
- D) Falsch. Die Aussage, dass funktionierende Software eine primäre Messgröße für den Erfolg ist, findet sich im Agilen Manifest.

2 / 40

Was ist laut der US-Bundesbehörde National Institute of Standards and Technology (NIST), **kein** wesentliches Merkmal von Cloud Computing?

- A) Weitreichender Netzzugang
- B) Nutzungsabhängiges Gebührensystem
- C) Schnelle Elastizität
- D) Ressourcenbündelung

- A) Falsch. Ein weitreichender Netzzugang ist eines der von der US-Bundesbehörde NIST genannten wesentlichen Merkmale des Cloud Computing.
- B) Richtig. Das nutzungsabhängige Gebührensystem wird zwar häufig verwendet, ist aber keines der von der US-Bundesbehörde NIST genannten wesentlichen Merkmale. Andere Vertragsarten sind ebenfalls möglich. (Literatur: A, Kapitel 1.1.2)
- C) Falsch. Schnelle Elastizität ist eines der von der US-Bundesbehörde NIST genannten wesentlichen Merkmale des Cloud Computing.
- D) Falsch. Die Ressourcenbündelung ist eines der von der US-Bundesbehörde NIST genannten wesentlichen Merkmale des Cloud Computing.

3 / 40

Die IT-Arbeit innerhalb von DevOps ist anders organisiert als bei traditionelleren Vorgehensweisen. Um diese Umstellung zu ermöglichen, sind mehrere Infrastrukturmanagement-Technologien erforderlich.

Welche sind **zwei** dieser Infrastrukturmanagement-Technologien?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Digitale Kanban-Tafeln
 - B) Infrastructure as a Service (IaaS)
 - C) Low-Code-/No-Code-Plattformen
 - D) Plattform für Testautomatisierung
 - E) Virtualisierung (Hypervisoren)
-
- A) Falsch. Kanban-Tafeln sind eine Methode zum Workflow-Management, keine Infrastrukturtechnologie.
 - B) Richtig. IaaS und Virtualisierung sind die grundlegenden Infrastrukturtechnologien, auf die sich die Frage bezieht. (Literatur: A, Kapitel 1)
 - C) Falsch. Low-Code-/No-Code-Plattformen sind Anwendungsentwicklungsplattformen und keine grundlegenden Infrastrukturtechnologien für DevOps.
 - D) Falsch. Plattformen für Testautomatisierung sind zwar nützlich für Softwarequalität und Liefargeschwindigkeit, gehören aber nicht zu den Infrastrukturmanagement-Technologien, die die Neuorganisation der IT-Arbeit durch DevOps ermöglicht haben.
 - E) Richtig. IaaS und Virtualisierung sind die grundlegenden Infrastrukturtechnologien, auf die sich die Frage bezieht. (Literatur: A, Kapitel 1)

4 / 40

DevOps baut stark auf den Prinzipien und Praktiken von Lean Production auf. Zu den Verschwendungsarten in der IT zählt auch das 'Task Switching'.

Welcher Verschwendungsart in Lean Production entspricht das 'Task Switching' in der IT?

- A) Bewegung
 - B) Überproduktion
 - C) Verkehr
 - D) Wartezeiten
-
- A) Falsch. Die Verschwendungsart 'Bewegung' bei Lean entspricht der Verschwendungsart 'Hand-off-Prozess' in der IT.
 - B) Falsch. Die Verschwendungsart 'Überproduktion' bei Lean entspricht der Verschwendungsart 'Extra Feature' in der IT.
 - C) Richtig. Die Verschwendungsart 'Verkehr' bei Lean entspricht der Verschwendungsart 'Task Switching' in der IT. (Literatur: A, Kapitel 2.1.1)
 - D) Falsch. Die Verschwendungsart 'Wartezeiten' in der IT entspricht der Verschwendungsart 'Wartezeiten' bei Lean.

5 / 40

Die wertvollsten Informationen einer Wertstromanalyse stammen aus drei Schlüsselkennzahlen.

Welche **drei** Schlüsselkennzahlen sind gemeint?

Bitte denken Sie daran, 3 Antworten auszuwählen.

- A) Flow
 - B) Durchlaufzeit
 - C) C/A-Anteil, das heißt der Anteil der Leistungen die vollständig und richtig erbracht wurden
 - D) Prozesszeit
 - E) Prozesszeit geteilt durch Durchlaufzeit
 - F) Fertiggestellter Wert minus Verschwendung
-
- A) Falsch. Flow ist ein Lean-Konzept, aber keine Kennzahl an sich.
 - B) Richtig. Die Durchlaufzeit ist eine Schlüsselkennzahl. (Literatur: A, Kapitel 3.1)
 - C) Richtig. Der C/A-Anteil ist eine Schlüsselkennzahl der Wertstromanalyse, die dabei hilft, die wertvollsten Informationen zu gewinnen. (Literatur: A, Kapitel 3.1)
 - D) Richtig. Die Prozesszeit ist eine Schlüsselkennzahl der Wertstromanalyse, die dabei hilft, die wertvollsten Informationen zu gewinnen. (Literatur: A, Kapitel 3.1)
 - E) Falsch. Prozesszeit und Durchlaufzeit sind zwar beide wichtige Kennzahlen, ihr Verhältnis zueinander ist jedoch kein aussagekräftiger Maßstab.
 - F) Falsch. Erbrachter Wert minus Verschwendung ist ebenfalls keine Kennzahl, auch wenn beide Begriffe im DevOps-Kontext relevant sind.

6 / 40

Von der Umstellung auf DevOps erwartet man sich eine höhere IT-Rentabilität.

Inwiefern unterscheidet sich DevOps bezüglich der Erfüllung dieser Erwartung von Agile, Scrum und Lean?

- A) DevOps beschleunigt die Lieferung neuer und geänderter Produkte zwischen Entwicklung und Betrieb.
 - B) DevOps beschleunigt die Lieferung neuer und geänderter Produkte an Markt und Kunden.
 - C) DevOps sorgt für schnellere Reaktionen auf Änderungen der IT-Infrastruktur innerhalb des vereinbarten Kostenrahmens.
 - D) DevOps sorgt für schnellere Reaktionen auf Zwischenfälle (Incidents) die den Wertstrom der Organisation behindern.
-
- A) Falsch. Die schnellere Lieferung von Produkten zwischen den Abteilungen ist eine Folge von DevOps. Da diese jedoch keinen Mehrwert für den Kunden liefert, zählt sie nicht zu den Werten, die man sich von DevOps zur Steigerung der IT-Rentabilität erwartet.
 - B) Richtig. Indem Produkte schneller an die Kunden in der Produktion geliefert werden, hilft DevOps dem Unternehmen Wert schneller und besser zu realisieren und steigert damit die IT-Rentabilität. (Literatur: A, Kapitel 1.2)
 - C) Falsch. Änderungen an der Infrastruktur sind eine interne Notwendigkeit der IT; das Gleiche gilt für die Einhaltung des vorgegebenen Kostenrahmens. Die reine Änderung der IT-Infrastruktur allein, bringt dem Unternehmen noch keine Steigerung der IT-Rentabilität.
 - D) Falsch. Dies ist keine Möglichkeit zur Steigerung der IT-Rentabilität. Eine schnellere Reaktion mag die Benutzererfahrung verbessern, führt aber für sich genommen noch nicht zu mehr Rentabilität.

7 / 40

Eine allgemein anerkannte Definition des Begriffs „Time-to-Market“ lautet:

Die Zeit zwischen der Geschäftsidee und dem Zeitpunkt, zu dem ein Kunde das Produkt erwerben kann.

Welche **zwei** DevOps-Praktiken tragen am **meisten** dazu bei, die Time-to-Market zu verkürzen?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Automatisiertes Testen
 - B) Zentralisierung der Entscheidungsfindung
 - C) Lose gekoppelte Architektur
 - D) Verwendung von Kanban-Boards
 - E) Wertstromanalyse (VSM)
-
- A) Richtig. Schnelle, zuverlässige Tests verkürzen die Feedbackschleifen und ermöglichen schnellere, sicherere Releases. Unabhängige Deployments, die durch eine lose gekoppelte Architektur ermöglicht werden, erlauben kleinere und häufigere Releases und verkürzen so die Markteinführungszeit. (Literatur: A, Kapitel 1)
 - B) Falsch. Zentralisierte Entscheidungsfindung und Stage Gates verzögern Entscheidungen und Releases.
 - C) Richtig. Schnelle, zuverlässige Tests verkürzen die Feedbackschleifen und ermöglichen schnellere, sicherere Releases. Unabhängige Deployments, die durch eine lose gekoppelte Architektur ermöglicht werden, erlauben kleinere und häufigere Releases und verkürzen so die Markteinführungszeit. (Literatur: A, Kapitel 1)
 - D) Falsch. Die Visualisierung der Arbeit auf einem Kanban-Board kann zwar dazu beitragen, Arbeit besser sichtbar zu machen, gewährleistet aber nicht, dass sie schneller erledigt wird oder sich die Markteinführungszeit verkürzt.
 - E) Falsch. Sie visualisiert Flow, Warteschlangen und Engpässe, verkürzt aber nicht unmittelbar die Markteinführungszeit.

8 / 40

Welche **zwei** gängigen Praktiken reduzieren technische Schulden **am direktesten**?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Die Praktiken des Change Managements
- B) Die Praktiken des Incident Managements
- C) Die Erhöhung von Ressourcen
- D) Sich dem Problem stellen
- E) Refactoring
- F) Die Praktiken des Release Managements

- A) Falsch. Change-Management-Praktiken können technische Schuld erhöhen, statt sie abzubauen.
- B) Falsch. Incident-Management-Praktiken können technische Schuld erhöhen, statt sie abzubauen.
- C) Falsch. Eine Aufstockung von Ressourcen trägt nicht direkt zur Reduzierung technischer Schuld bei, auch wenn technische Schuld mit mehr Entwicklern möglicherweise schneller abgebaut werden kann.
- D) Richtig. Zwei der häufig verwendeten Praktiken sind Refactoring und das aktive Angehen des Problems. (Literatur: A, Kapitel 1.3)
- E) Richtig. Zwei der häufig verwendeten Praktiken sind Refactoring und das aktive Angehen des Problems. (Literatur: A, Kapitel 1.3)
- F) Falsch. Release-Management-Praktiken können technische Schuld erhöhen, statt sie abzubauen.

9 / 40

Welchen großen Vorteil bietet DevOps einer Organisation?

- A) DevOps eliminiert fragile Unternehmenssysteme.
- B) DevOps verringert die Kosten für den Kunden.
- C) DevOps reduziert kulturelle Schwierigkeiten.

- A) Richtig. Von DevOps erwartet man dass: es die Produkteinführungszeit verkürzt, die technische Schulden abbaut und fragile Systeme eliminiert. (Literatur: A, Kapitel 1.3.3)
- B) Falsch. DevOps-Praktiken können kostspieliger sein. Die Reduzierung der dem Kunden entstehenden Kosten ist keine Priorität.
- C) Falsch. DevOps fördert zwar die Vielfalt in den Teams, aber das heißt noch lange nicht, dass kulturelle Schwierigkeiten verringert werden.

10 / 40

DevOps wird manchmal für einen Teil von Agile gehalten.

Auf welche Frage, die mit Agile überhaupt nichts zu tun hat, gibt DevOps eine Antwort?

- A) Was kann man tun, um das Release der von uns entwickelten Software so zu beschleunigen, dass wir unseren Marktanteil vergrößern?
 - B) Welche Art von Kundenbeziehung brauchen wir, um die Anforderungen unserer Kunden besser zu verstehen?
 - C) Was sollte nach dem Release mit einem Produkt getan werden, damit es nützlich und einfach zu betreiben ist?
-
- A) Falsch. Das Release eines Produkts ist sowohl Teil von Agile als auch von DevOps. Agile unterstützt ein schnelles Release der funktionierenden Teile des Endprodukts. DevOps erkundet darüber hinaus, wie man Kunden besser echten Mehrwert bieten kann.
 - B) Falsch. Die Beziehung zum Kunden und die Erhebung der Anforderungen sind sowohl Teil von Agile als auch von DevOps. Product Owner halten engen Kontakt zum Kunden, um Mehrwert für das Produkt sicher zu stellen.
 - C) Richtig. Die Berücksichtigung der Phasen Release und Betrieb bei der Entwicklung ist nicht unbedingt Teil von Agile, aber ein wesentlicher Teil von DevOps. (Literatur: A, Kapitel 1.5.1)

11 / 40

Why is the concept 'value stream' so important in DevOps?

- A) The value stream helps to analyze the as-is map and any attempts to improve metrics.
 - B) The value stream helps to realize a smooth and uniform flow through all process steps.
 - C) The value stream helps to realize local optimization in the current work practice.
-
- A) Incorrect. The optimization work should not be limited to analyzing the as-is map. It is necessary to develop a to-be map, which may be quite different from the current work practice.
 - B) Correct. The concept of the value stream ensures that a smooth and uniform flow from one step to the next in the process can be created. (Literature: A, Chapter 3.1)
 - C) Incorrect. The value stream helps to identify and eliminate bottlenecks, while avoiding the local optimization trap.

12 / 40

Was ist bei der Wertstromanalyse der **erste** Schritt?

- A) Die Erstellung der Anforderungsspezifikation
 - B) Die Dokumentation der fertiggestellten Arbeit
 - C) Die Identifizierung der wichtigsten Prozessschritte
-
- A) Falsch. Der erste Schritt sollte darin bestehen, die wichtigsten Schritte für den jeweiligen Prozess und die Organisation des Teams zu identifizieren.
 - B) Falsch. Die Dokumentation der fertiggestellten Arbeit erfolgt erst nachdem die wichtigsten Schritte identifiziert sind. Arbeit an unwesentlichen Schritten ist nicht Teil der Wertstromanalyse.
 - C) Richtig. Dies sollte der erste Schritt einer Wertstromanalyse sein. Die Organisation sollte prüfen, für welchen Prozess sie eine Wertstromanalyse erstellen will und dann die wichtigsten Schritte identifizieren, an denen Arbeit durchgeführt und Mehrwert geschaffen wird. (Literatur: A, Kapitel 3.1)

13 / 40

Eine Wertstromanalyse zeigt deutlich, wo sich im Wertstrom Ineffizienzen verbergen.

Warum sollte eine Wertstromanalyse durchgeführt werden?

- A) Um die Verbesserung der Unternehmensprozesse zu unterstützen
 - B) Um zu sehen, welche Mitarbeiter nicht hart genug arbeiten
 - C) Um Work-in-Progress (WIP) zu beschleunigen
 - D) Um zu visualisieren, welche Produktlinie eingestellt werden sollte
-
- A) Richtig. Das ist der Grund, an den man denken sollte, wenn man eine Wertstromanalyse erstellt und hierfür bringt die Wertstromanalyse den größten Nutzen. (Literatur: A, Kapitel 3.1)
 - B) Falsch. Eine Wertstromanalyse wird nicht durchgeführt, um zu sehen, wer überflüssig ist, wem zuerst gekündigt werden sollte oder wer sich nicht genug einbringt. Sie sollte durchgeführt werden, um die Prozesse zu verbessern und damit die Mitarbeiter im Unternehmen beginnen können, größere Mengen in besserer Qualität zu produzieren.
 - C) Falsch. Eine Wertstromanalyse beschleunigt zwar möglicherweise den Prozess, nicht unbedingt aber die Arbeit selbst. Um Fehler zu vermeiden, mehr nach dem Prinzip 'First-Time-Right' zu fertigen und damit einen Schritt in der Qualitätskontrolle einzusparen, kann die Arbeit an sich sogar langsamer werden. Ziel ist nicht die Beschleunigung der Arbeit, sondern die Eliminierung von Verschwendung und die Wertschöpfung für das Unternehmen.
 - D) Falsch. Zwar kann sich das Unternehmen auf der Basis der Wertstromanalyse zur Einstellung eines Geschäfts- oder Produktzweigs entscheiden, aber dies ist nicht das eigentliche Ziel der Wertstromanalyse. Stattdessen sollte die Wertstromanalyse nur für funktionale Produkte durchgeführt werden.

14 / 40

Aufgaben müssen priorisiert werden. Die Priorisierung erfolgt in der Queue (Warteschlange) vor Eintritt in den Wertstrom.

Warum verursacht diese Priorisierung häufig Probleme?

- A) Weil an diesem Punkt die Entscheidung über den Bau der Deployment Pipeline für die automatisierbaren Aufgaben gefällt wird. Dies ist zeitaufwändig und führt zu Verzögerungen.
 - B) Weil die Befähigung zur Messung der wichtigsten Kennzahlen im Wertstrom falsch oder ineffizient erfolgt und daher Probleme verursacht.
 - C) Weil die Nutzung von Visualisierungstools in Kombination mit WIP-Begrenzungen (WIP-Limits), durch die sich Engpässe bei der Aufgabe identifizieren lassen, falsch umgesetzt wird.
 - D) Weil sowohl die Ist-Version als auch die Soll-Version des Wertstroms und eine Liste der erforderlichen Änderungen erstellt werden müssen und dies nimmt viel Zeit in Anspruch.
 - E) Weil man, falls nach wie vor der traditionelle Ansatz verwendet wird, noch vor Beginn der eigentlichen Arbeit viele Entscheidungen treffen muss, was massive Verzögerungen zur Folge hat.
-
- A) Falsch. Dieser Schritt erfolgt zu einem späteren Zeitpunkt im Prozess und verursacht keine Probleme in der Queue beim Eintritt in den Wertstrom.
 - B) Falsch. Dies ist kein Ziel des ersten Schritts im Wertstrom (Maximierung der Zahl der Metriken). Er sollte an diesem Punkt nicht durchgeführt werden und daher auch nicht die erwähnten Probleme verursachen.
 - C) Falsch. So erzielt man einen gleichmäßigen Flow ohne Verzögerungen und zeigt Engpässe auf. Dies ist jedoch nicht das Problem, das an dem genannten Punkt auftritt.
 - D) Falsch. Dies erfolgt erst zu einem späteren Zeitpunkt im Wertstromprozess und sollte folglich an diesem Punkt keine Probleme verursachen.
 - E) Richtig. Setzen Organisationen auch weiterhin stark auf herkömmliche Praktiken und Methoden, bei denen alle Entscheidungen bezüglich der Arbeit noch vor Beginn der eigentlichen Arbeit getroffen werden, verursacht dies die genannten Probleme. (Literatur: A, Kapitel 4.10)

15 / 40

Welches Bild liegt dem Begriff 'Deployment Pipeline' zugrunde?

- A) Eine Pipeline mit Flüssigkeit, die für Flow sorgt
 - B) Ein Montageband wie zum Beispiel in einer Autofabrik
 - C) Moderne Prozessoren, die paralleles Pipelining nutzen
 - D) Die Idee mehr als ein Montageband einzusetzen
-
- A) Falsch. Diese Meinung ist zwar weit verbreitet, aber leider falsch und entspricht nicht der eigentlichen Bedeutung des Begriffs.
 - B) Falsch. Diese Meinung ist zwar weit verbreitet, aber leider falsch und entspricht nicht der eigentlichen Bedeutung des Begriffs.
 - C) Richtig. Humble und Farley stellten klar, dass sie, als sie den Begriff prägten, an das Pipelining moderner Prozessorarchitekturen dachten. Durch Pipelining können diese Architekturen die gewünschten Ergebnisse viel schneller produzieren. (Literatur: A, Kapitel 3.2)
 - D) Falsch. Diese Meinung ist zwar weit verbreitet, aber leider falsch und entspricht nicht der eigentlichen Bedeutung des Begriffs.

16 / 40

Bei der Implementierung der Deployment Pipeline kommt es zu Problemen. Die Zahl der vorab entwickelten Tests reicht anfänglich nicht aus, um einen konstanten Betrieb der Produktionsumgebung sicherzustellen.

Was ist die **beste** Lösung für dieses Problem?

- A) Die Pipeline und so viele automatisierte Prozesse wie möglich sollten erstellt, aber nicht genutzt werden, bis alle ordnungsgemäßen Tests vorhanden sind.
 - B) Die Abdeckung des Codes sollte durch Tests gesteigert werden und muss ähnlich wie technische Schulden schnellstmöglich angegangen werden.
 - C) Die Pipeline sollte mit den entwickelten Tests laufen und potenzielle Probleme in der Produktion sollten gewidmet werden, wenn diese auftreten.
 - D) Die Pipeline sollte lediglich als Integrationssystem genutzt werden, um den erstellten Code an die für das Testen und die Qualitätssicherung (QA) zuständige Abteilung zu liefern.
-
- A) Falsch. Die Pipeline zu bauen und sie erst zu nutzen, wenn alle Tests vorliegen, dauert zu lange, dadurch verliert das Unternehmen Geld. Angestrebt werden sollte ein iterativer Ansatz, bei dem zuerst die wichtigsten Tests und dann nach und nach neue Tests generiert werden und die Abdeckung so gesteigert wird.
 - B) Richtig. Eine höhere Testabdeckung des Codes ist die einzig richtige Lösung für dieses Problem. (Literatur: A, Kapitel 3.2)
 - C) Falsch. Betreibt man die Pipeline mit nur wenigen Tests, so führt dies zu einer schlechten Testabdeckung und verursacht möglicherweise viele Probleme in der Produktion. Dies wiederum wirkt sich hinderlich auf die Annahme der Pipeline.
 - D) Falsch. Ziel der Deployment-Pipeline ist die Lieferung von funktionsfähigem Code an die Produktion und nicht nur das Testen und die Qualitätssicherung. Das heißt die Deployment-Pipeline wird ihren Zweck nicht erfüllen.

17 / 40

Ein gutes System zur Versionskontrolle ist bei DevOps einer der besten Prädiktoren für hohe Leistungsfähigkeit.

Was ist für eine erfolgreiche Versionskontrolle nötig?

- A) Ein Kulturwandel der Arbeit mit Informationen und Konfigurationen
 - B) Eine signifikant schnellere Umsetzung von Änderungen
 - C) Die absichtliche Einbringung von Chaos und Instabilität in die Produktion
 - D) Die Nutzung eines förmlichen und automatisierten Change-Management-Prozesses
-
- A) Richtig. Durch die Versionierung lassen sich alle relevanten Teile des Systems im Betrieb lenken; dies lässt sich mit anderen Werkzeugen nicht erreichen. Eine gute Versionskontrolle erfordert einen anderen Umgang mit Informationen und Konfigurationen. (Literatur: A, Kapitel 3.3)
 - B) Falsch. Durch die Nutzung virtueller Cloud-Technologien ist der Grad der Automatisierung ebenso wie die Geschwindigkeit der Umsetzung von Änderungen in den letzten Jahren beträchtlich gestiegen; dies ist jedoch für die Versionskontrolle nicht erforderlich.
 - C) Falsch. Eine der großartigen DevOps-Praktiken zur Eliminierung von fragilen Systemen ist die absichtliche Einbringung von Chaos und Instabilität in die Produktionsumgebung. Für diese Methode gibt es mehrere Bezeichnungen: Game Day, Chaos Gorilla, Simian Army. Für die Versionskontrolle ist dies jedoch nicht erforderlich.
 - D) Falsch. Um mit fragilen Systemen in der IT-Infrastruktur umzugehen, nutzen manche Organisationen formalisierte und automatisierte Prozesse für das Change Management, die darauf ausgelegt sind, den Änderungsfluss zu strukturieren und die mit der Umsetzung der Änderung verbundenen Risiken zu minimieren. Für die Versionskontrolle ist dies jedoch nicht erforderlich.

18 / 40

Wichtig bei DevOps ist ein ausgewogenes Verhältnis zwischen einer schnellen Reaktion und der Aufrechterhaltung der Zuverlässigkeit der Anwendung.

Inwiefern wird dies durch die Versionskontrolle unterstützt?

- A) Indem sie es jedem Mitglied des Teams ermöglicht, unnötige Dateien oder Dokumente ohne Einschränkung zu löschen
 - B) Indem sie die Bildung kleiner, unabhängiger und autarker Entwicklungsteams ermöglicht
 - C) Indem sie Spezialwerkzeuge zur Verringerung oder Eliminierung von Verschwendung und zur Verbesserung von Prozessen einsetzt
-
- A) Richtig. Die Versionskontrolle ermöglicht es allen Teammitgliedern, unnötige Dateien und Dokumente ohne Einschränkung zu löschen und ohne Risiko, dass dabei wichtige Informationen oder Produkte versehentlich verloren gehen. (Literatur: A, Kapitel 3.3)
 - B) Falsch. Die Bildung kleiner, unabhängiger und autarker Teams ist zwar einer der Kerngedanken von DevOps, hat aber nichts damit zu tun, wie die Versionskontrolle die Agilität und Zuverlässigkeit des Systems unterstützt.
 - C) Falsch. Dies beschreibt eine praktische Anwendung der Kerngedanken der Lean-Produktion auf die IT. Bei dieser Anwendung wird Verschwendung mit Hilfe von Spezialwerkzeugen identifiziert und dann mit Hilfe von anderen Spezialwerkzeugen verringert oder ganz eliminiert. Dies hat jedoch nichts damit zu tun, wie die Versionskontrolle zur Agilität und Zuverlässigkeit des Systems beiträgt.

19 / 40

Welche Vorteile bietet das Konfigurationsmanagement?

- A) Es ermöglicht allen Teammitgliedern das risikofreie Löschen unerwünschter Dateien.
 - B) Es ermöglicht den wichtigsten Team-Mitgliedern die problemfreie Abwesenheit.
 - C) Es ermöglicht den Teammitgliedern zu sehen, wer wann welchen Code geändert hat.
-
- A) Falsch. Das Versionsmanagement ermöglicht allen Teammitgliedern das uneingeschränkte Löschen von Informationen. Falls etwas schief geht, lässt sich die vorherige Version wiederherstellen.
 - B) Richtig. Werden alle Änderungen vom Konfigurationsmanagement gelenkt, dann kann das System bei Bedarf automatisch auf einen früheren stabilen Zustand zurückgesetzt werden. Ein weiterer Vorteil ist, dass kein Wissen verloren geht, falls Teammitglieder das Unternehmen verlassen, da das Wissen in der Konfiguration verankert ist. (Literatur: A, Kapitel 3.4)
 - C) Falsch. Die Aufzeichnung was wann, von wem geändert wurde ist Teil der Versionskontrolle nicht des Konfigurationsmanagements.

20 / 40

Das Konfigurationsmanagement ermöglicht die Skalierung der IT-Infrastruktur und Softwaresysteme ohne erhöhten Personalaufwand.

Wie sollten Änderungen in einer solchen skalierten Umgebung **am besten** durchgeführt werden?

- A) Mittels Continuous Integration (kontinuierlicher Integration)
 - B) Mittels vollständig gelenkter Skripten
 - C) Mittels Testautomatisierung
 - D) Mittels der Deployment Pipeline
-
- A) Falsch. Continuous Integration ist die zweite Phase in der Implementierung einer Deployment Pipeline, nicht aber die einzige Art und Weise, in der es Administratoren erlaubt sein sollte, etwas in der Produktion zu ändern. Änderungen erfolgen idealerweise mittels automatisierter Skripten.
 - B) Richtig. Eigentlich könnte man anführen, dass auch Administratoren in der Produktion nicht das Recht haben sollten, etwas zu ändern. Auch sie sollten Änderungen nur mit Hilfe von vollständig gelenkten (und automatisierten) Skripten durchführen dürfen. (Literatur: A, Kapitel 3.3 und 3.4)
 - C) Falsch. Die Testautomatisierung ist die dritte Phase in der Umsetzung einer Deployment Pipeline, nicht aber die einzige Art und Weise, in der es Administratoren erlaubt sein sollte, etwas in der Produktion zu ändern. Änderungen erfolgen idealerweise mit Hilfe von automatisierten Skripten.
 - D) Falsch. Auf einer abstrakten Ebene ist eine gut funktionierende und vollständig automatisierte Deployment Pipeline eine automatisierte Manifestation des Prozesses, mit dem die Software von der Versionskontrolle an den Benutzer bereitgestellt wird. Eine manuelle Deployment Pipeline ist jedoch möglich. Die Deployment Pipeline beschreibt aber nicht die einzige Art und Weise, in der es Administratoren erlaubt sein sollte, etwas in der Produktion zu ändern. Änderungen erfolgen idealerweise mit Hilfe von automatisierten Skripten.

21 / 40

Eine eindeutige Definition of Done (DoD, Definition von 'Fertiggestellt') ist bei DevOps von entscheidender Bedeutung und berücksichtigt den Wert für den Kunden.

Wie lautet bei DevOps eine gute Beschreibung für Done?

- A) Eine Anforderung gilt als Done, nachdem sie gebaut wurde.
 - B) Eine Anforderung gilt als Done, nachdem sie getestet wurde.
 - C) Eine Anforderung gilt als Done, nachdem das Produkt abgenommen wurde.
 - D) Eine Anforderung gilt als Done, nachdem das Produkt in der Produktion ist.
-
- A) Falsch. Eine gute Beschreibung für Done bei DevOps lautet wie folgt: Etwas ist Done, wenn dem Kunden Mehrwert entstanden ist. Build ist lediglich eine Phase der DevOps-Pipeline, Mehrwert ist zu diesem Zeitpunkt noch nicht entstanden.
 - B) Falsch. Eine gute Beschreibung für Done bei DevOps lautet wie folgt: Etwas ist Done, wenn dem Kunden Mehrwert entstanden ist. Testen ist lediglich eine Phase der DevOps-Pipeline, Mehrwert ist zu diesem Zeitpunkt noch nicht entstanden.
 - C) Falsch. Eine gute Beschreibung für Done bei DevOps lautet wie folgt: Etwas ist Done, wenn dem Kunden Mehrwert entstanden ist. Die Abnahme ist lediglich eine Phase der DevOps-Pipeline, Mehrwert ist zu diesem Zeitpunkt noch nicht entstanden.
 - D) Richtig. Eine gute Beschreibung für Done bei DevOps lautet wie folgt: Etwas ist Done, wenn dem Kunden Mehrwert entstanden ist. Dies ist der Fall, sobald das Produkt in der Produktionsumgebung eingeführt ist. (Literatur: A, Kapitel 3.5)

22 / 40

Nutzt man die traditionellen Praktiken, so kann bei einem Release Einiges schief gehen: Manche Änderungen werden nicht dokumentiert, es gibt kein vollständiges Backup des Systems und der Zustand des Systems vor dem Release wurde nicht gespeichert.

Wie sorgt DevOps für häufige Releases ohne diese Probleme?

- A) Indem Releases automatisiert werden
 - B) Indem der Betrieb die Releases durchführt
 - C) Indem Releases sehr klein gemacht werden
 - D) Indem nicht alle Änderungen dokumentiert werden
-
- A) Richtig. Die Automatisierung ist ein wichtiger Faktor, um häufige Releases sicherzustellen und dafür zu sorgen, dass der Release-Prozess zur Routine wird. Werden alle menschlichen Faktoren bei Backups, Dokumentation und Roll-Back automatisiert, so verringert sich die Wahrscheinlichkeit von Problemen bei Releases dramatisch. Kann ein Release nicht erfolgreich umgesetzt werden, so alarmiert das System das Team, das daraufhin Maßnahmen ergreifen muss. (Literatur: A, Kapitel 4.1)
 - B) Falsch. Die erwähnten Probleme lassen sich nicht dadurch vermeiden, dass man dem Betrieb mehr Kontrolle gibt. Sieht man sich den Betrieb an und integriert man die Entwicklung in die Praktiken des Betriebs, so wirft dies möglicherweise Licht darauf, welche Teile des Release-Prozesses automatisiert und standardisiert werden müssen. Die reine Übergabe von Releases an den Betrieb löst das genannte Problem nicht.
 - C) Falsch. Wenn es um das Funktionieren routinemäßiger Releases geht, dann ist die Größe des Release weniger wichtig als die Automatisierung. Die genannten Probleme lassen sich durch kleinere Releases nicht unbedingt vermeiden. Werden Releases häufiger durchgeführt, so führt dies automatisch zu kleineren Releases.
 - D) Falsch. Dies macht die Probleme noch schlimmer. Die Einführung eines Versionierungssystems, vorzugsweise eines automatisierten Versionierungssystems, trägt dazu bei, die durch unsachgemäße Dokumentation verursachten Probleme zu beheben.

23 / 40

Ein Unternehmen nutzt Continuous Deployment (kontinuierliche Bereitstellung).

Wer sollte entscheiden, wann das Release der neuen Funktionalität erfolgt?

- A) Das Unternehmen
 - B) Der Kunde
 - C) Die IT-Abteilung
 - D) Der Benutzer
- A) Richtig. Nutzt man Continuous Deployment, dann entscheidet das Unternehmen darüber, wann eine neue Funktionalität zur Benutzung freigegeben wird. Es kann dabei vorkommen, dass die Funktionalität schon seit geraumer Zeit bereitgestellt wird, bevor sich das Unternehmen entscheidet, diese zu aktivieren. Die IT-Abteilung führt das Deployment der Funktionalitäten in ihrem eigenen Tempo durch und das Unternehmen entscheidet dann über das Release der neuen Funktionalität. (Literatur: A, Kapitel 4.1)
- B) Falsch. Dieser Stakeholder spielt zwar bei der Entscheidung eine wichtige Rolle, aber die Entscheidung zur durch Durchführung des Release liegt in erster Linie beim Unternehmen.
- C) Falsch. Dieser Stakeholder spielt zwar bei der Entscheidung eine wichtige Rolle, aber die Entscheidung zur durch Durchführung des Release liegt in erster Linie beim Unternehmen.
- D) Falsch. Dieser Stakeholder spielt zwar bei der Entscheidung eine wichtige Rolle, aber die Entscheidung zur durch Durchführung des Release liegt in erster Linie beim Unternehmen.

24 / 40

Was ist laut DevOps-Praktiken der **beste** Weg, um den Grad der Betriebslenkung zu steigern?

- A) Die Automatisierung aller manuellen Betriebsvorgänge
 - B) Die Definition geeigneter Rollen und Zuständigkeiten
 - C) Die Entwicklung von Lenkungsverfahren
 - D) Die Verbesserung der Betriebsführung
- A) Richtig. Die Automatisierung möglichst vieler Betriebsvorgänge verschafft vollständige Kontrolle, da alle Betriebsvorgänge sofort und einheitlich werden. Ist ein Vorgang suboptimal, so kann man durch eine Änderung der Automatisierung gleichzeitig alle künftigen Vorgänge ändern. (Literatur: A, Kapitel 4.1.3)
- B) Falsch. Die Definition von Rollen und Zuständigkeiten ist zwar wichtig, wirkt sich aber nicht unmittelbar auf die Steuerung aller Betriebsvorgänge aus.
- C) Falsch. Allein die Entwicklung neuer Verfahren, ist nicht hilfreich. Die Verfahren sollten nach der Entwicklung automatisiert werden, damit keine Fehler entstehen und alle Vorgänge einheitlich sind.
- D) Falsch. Dies ist zwar möglicherweise wünschenswert, aber die Betriebsführung allein, führt noch nicht zu mehr Lenkung und Kontrolle. Automatisierung dagegen sorgt für vollständige Lenkung und Kontrolle.

25 / 40

Wie löst DevOps Zwischenfälle (Incidents)?

- A) DevOps eskaliert Zwischenfälle an das Problem Management-Team und stellt eine Lösung bereit, bis das Team den Zwischenfall behebt.
 - B) DevOps untersucht den Zwischenfall, führt eine Diagnose durch, findet dann einen Workaround (Umgehungslösung) und setzt diesen um.
 - C) DevOps prüft, ob ein ähnlicher Zwischenfall schon einmal aufgetreten ist und setzt dann eine ähnliche Lösung für das Problem um.
 - D) DevOps verfolgt den Vorfall zurück auf ein vor kurzem stattgefundenes Deployment und führt dann ein Roll-Back des Systems auf den vormals stabilen Zustand durch.
-
- A) Falsch. Diese Lösung kann zwar zu einem funktionierenden Workaround führen, das tatsächliche Problem würde dann aber niemals gelöst.
 - B) Falsch. Durch die Umsetzung eines Workarounds wird der Zwischenfall selbst nicht gelöst. So sollte man nicht vorgehen.
 - C) Falsch. Die Lösung ist mit höchster Wahrscheinlichkeit nur ein Workaround. Das ist nicht wünschenswert. Man sollte nicht warten, bis ein anderer das Problem behebt.
 - D) Richtig. Die Literatur sagt hierzu Folgendes: Wird ein Zwischenfall auf ein vor kurzem stattgefundenes Deployment zurückgeführt, so führt das Steuersystem der Pipeline automatisch ein 'Roll-Back' auf den letztbekanntesten stabilen Zustand durch. Genau dies sollte passieren, und zwar ganz unabhängig davon, ob der Prozess automatisiert ist oder nach wie vor manuell durchgeführt wird. (Literatur: A, Kapitel 4.1)

26 / 40

Welche Maßnahmen sollten bei DevOps ergriffen werden, wenn man Prozessmängel feststellt?

- A) Alle Änderungen sollten in Form eines Backlogs eingereicht werden, damit sie im Rahmen eines Projekts oder einer Kaizen-Veranstaltung freigegeben werden können.
 - B) Möglichkeiten der Fehlerbehebung müssen ermittelt und nach Entdecken der Mängel schnellstmöglich umgesetzt werden.
 - C) Möglichkeiten der Fehlerbehebung müssen ermittelt, vom Change Manager genehmigt und entsprechend ihrer Priorität freigegeben werden.
 - D) Möglichkeiten der Fehlerbehebung müssen ermittelt, vom Continuous Improvement Manager genehmigt und umgehend freigegeben werden.
 - E) Möglichkeiten der Fehlerbehebung sollten aufgeschoben werden, bis sie in einen geeigneten Sprint integriert werden können.
-
- A) Falsch. Eine große Change- oder Kaizen-Veranstaltung mag nützlich sein, aber im Rahmen der kontinuierlichen Verbesserung sollten regelmäßige Änderungen am Prozess schnellstmöglich durchgeführt werden.
 - B) Richtig. Prozessmängel, die nicht umgehend nach ihrer Feststellung behoben werden, können zu unnötigen Problemen führen. "Daher nutzt DevOps einen anderen Ansatz: Alle festgestellten Prozessmängel sollten umgehend behoben werden." (Literatur: A, Kapitel 4.1)
 - C) Falsch. Bei DevOps gibt es keinen Change Manager und die Fehlerbehebung sollte umgehend umgesetzt werden.
 - D) Falsch. Einen Continuous Improvement Manager, der die Fehlerbehebung genehmigt, gibt es bei DevOps nicht.
 - E) Falsch. Die Fehlerbehebung aufschieben ist keine gute Idee. Das System bleibt dadurch ineffizient und das wirkt sich negativ auf die Arbeit aus.

27 / 40

Was unterstützt ein DevOps-Team **nicht** bei der erfolgreichen Entwicklung und Lieferung einer funktionierenden Software?

- A) Die Bildung eines DevOps-Teams für eine kurze Zeit während eines Projekts
 - B) Die Fehler erkennen, beheben und daraus lernen, sobald sie gefunden werden
 - C) Die Organisation der DevOps-Teams rund um eine organisatorische Aufgabe
 - D) Die Erstellung von Code für eine Software, bei der das Hauptziel die Integration von Qualität ist
- A) Richtig. DevOps-Teams werden für längere Zeit gebildet. Nur so können sie ihre Erfahrung künftig nutzen, um neue Software schneller bereitzustellen und für kontinuierliche Innovationen zu sorgen. (Literatur: A, Kapitel 4.2)
- B) Falsch. Bei kleineren Batches kann man Fehler leichter finden und umgehend korrigieren. DevOps folgt dem Lean-Prozess, der darauf abzielt, Fehler schnell zu finden, umgehend im Prozess zu beheben und den Prozess zu optimieren.
- C) Falsch. Einer der Hauptvorteile von DevOps besteht darin, dass das Team dem Unternehmen Mehrwert bietet, indem es seine Arbeit an ein bestimmtes organisatorisches Ziel anpasst.
- D) Falsch. Die Konzentration auf die Integration von Qualität ist eines der Merkmale von Lean, das in DevOps eingebunden wurde. Achtet man schon bei der Code-Erstellung auf Qualität, dann kann die Erstellung zwar länger dauern, bietet aber letztendlich einen Mehrwert, da weniger Fehler gefunden werden und das System robuster ausgelegt ist.

28 / 40

DevOps empfiehlt kontinuierliche Aktualisierungen der fertiggestellten Arbeit.

Wie hält man alle Teammitglieder **am besten** kontinuierlich über den Fortschritt eines Flows auf dem Laufenden?

- A) Indem man es im Laufe des Tages mit Teammitgliedern bespricht
 - B) Indem man den Fortschritt beim täglichen Stand-up-Meeting bespricht
 - C) Indem man den Product Owner persönlich informiert
 - D) Indem man dem Scrum Master Aktualisierungsmails schickt
 - E) Indem man Visualisierung, wie zum Beispiel ein Kanban-Board, verwendet
- A) Falsch. Gespräche zwischen Teammitgliedern sind eine eher ineffiziente Methode, um alle kontinuierlich auf dem Laufenden zu halten, selbst wenn sich alle im selben Raum befinden.
- B) Falsch. DevOps erfordert kein tägliches Stand-up, obwohl dies gängige Praxis ist, da DevOps und Scrum gut zusammenpassen. In einem täglichen Stand-up werden nur die Themen des jeweiligen Tages besprochen. Ein Stand-up ist nicht als kontinuierliche Aktualisierung aller erledigten Arbeiten gedacht.
- C) Falsch. DevOps verlangt, dass alle Teammitglieder kontinuierlich auf dem Laufenden gehalten werden. Der Product Owner ist nur ein einzelnes Teammitglied. Außerdem ist ein persönliches Gespräch kein geeigneter Weg, um alle kontinuierlich zu informieren.
- D) Falsch. DevOps verlangt, dass alle Teammitglieder kontinuierlich auf dem Laufenden gehalten werden. Der Scrum Master ist nur ein einzelnes Teammitglied. Außerdem ist E-Mail kein geeignetes Mittel, um alle kontinuierlich zu informieren.
- E) Richtig. Ein Kanban-Board wird verwendet, um den Fortschritt des Teams darzustellen. Alle Teammitglieder können das Board jederzeit einsehen und so den gesamten Fortschritt verfolgen. (Literatur: A, Kapitel 4.3)

29 / 40

Ein Team fügt Elemente aus dem Backlog zu einem Batch hinzu.

Welche **zwei** Dinge müssen berücksichtigt werden?

Bitte denken Sie daran, dass Sie 2 Antworten wählen müssen.

- A) Die zur Erledigung des Backlog-Elements erforderliche Kapazität
 - B) Die aktuelle Höhe der technischen Schulden
 - C) Die gesamte Pipeline, von Anfang bis Ende
 - D) Die bevorzugten Besprechungszeiten der Teammitglieder
 - E) Die Urlaubszeiten der Teammitglieder
-
- A) Richtig. Die zur Erledigung des Backlog-Elements erforderliche Kapazität bestimmt, wie viele Elemente in den Batch bzw. die Timebox aufgenommen werden können, und hilft, eine Überlastung zu vermeiden. (Literatur: A, Kapitel 4)
 - B) Falsch. Technische Schulden sollten über spezielle Backlog-Elemente oder reservierte Kapazitäten verwaltet werden; ihr Gesamtumfang bestimmt nicht, welche neuen Elemente dem Batch hinzugefügt werden.
 - C) Richtig. Die Betrachtung der gesamten Pipeline von Anfang bis Ende stellt sicher, dass jedes Element die Definition of Done erfüllen und alle Phasen ohne Engpässe durchlaufen kann. (Literatur: A, Kapitel 4)
 - D) Falsch. Bevorzugte Besprechungszeiten haben keinen Einfluss darauf, welche Elemente ausgewählt werden; Terminpräferenzen werden getrennt von der Batch-Planung behandelt.
 - E) Falsch. Urlaubspläne werden in der Gesamtkapazität berücksichtigt; die Auswahl sollte sich nach Kapazität und Priorität richten und nicht nach individuellen Urlaubsdetails.

30 / 40

Auf welche **zwei** Arten berücksichtigt DevOps betriebliche Anforderungen?

Bitte denken Sie daran, dass Sie 2 Antworten auswählen müssen.

- A) Durch die Erstellung einer Definition of Done (DoD, Definition von „Fertiggestellt“)
 - B) Durch die Erstellung einer Wertstromanalyse (VSM)
 - C) Durch den Einsatz von Agile- und Scrum-Prozessen
 - D) Durch die Nutzung von Geschäftsanforderungen als Grundlage
 - E) Durch das Festlegen von Anforderungen im Voraus
-
- A) Richtig. Durch die Einbeziehung von Betriebskriterien in das DoD werden diese überprüfbar und für jedes Inkrement verbindlich. (Literatur: A, Kapitel 4.6)
 - B) Richtig. VSM betrachtet den gesamten Prozess von Anfang bis Ende und deckt betriebliche Anforderungen in den Phasen Build, Deploy und Run auf, um diese in die Bereitstellung zu integrieren. (Literatur: A, Kapitel 4.6)
 - C) Falsch. Frameworks allein gewährleisten keine Betriebsfähigkeit; operative Anforderungen müssen explizit definiert und priorisiert werden.
 - D) Falsch. Geschäftliche Vorgaben leiten die Prioritäten, aber operative Anforderungen erfordern explizite Service-Level-Vorgaben und die Einbindung der Operations-Abteilung in das Backlog/DoD.
 - E) Falsch. DevOps erfasst und verfeinert operative Anforderungen iterativ in Backlog-Elementen und Akzeptanzkriterien, nicht über eine feste Vorab-Spezifikation.

31 / 40

Welchen Vorteil hat es, wenn ein DevOps-Team über einen längeren Zeitraum zusammenarbeitet?

- A) Das Team muss den Prozess nicht mehr verbessern.
 - B) Das Team nutzt seine Erfahrung zur Innovation und Prozessverbesserung.
 - C) Das Team beginnt, unabhängiger zu arbeiten.
 - D) Das Team hat mehr Zeit für die häufigere Bearbeitung unerwarteter Anträge.
-
- A) Falsch. DevOps-Teams suchen immer nach Verbesserung, denn genau darum geht es bei Continuous Improvement (kontinuierlicher Verbesserung). Ein Team, das schon länger zusammenarbeitet, traut sich vielleicht sogar eher, Routinen zu verändern und Prozesse zu verbessern.
 - B) Richtig. DevOps-Teams, die länger zusammenarbeiten, können ihre Erfahrungen in künftigen Entwicklungen nutzen und so schneller Leistung erbringen und Prozesse erneuern. (Literatur: A, Kapitel 4.9)
 - C) Falsch. Zusammenarbeiten heißt nicht, dass das Team unabhängiger arbeitet. Das Team muss sich bei der Ausrichtung seiner Arbeit an der Mission des Unternehmens orientieren. Davon abgesehen sollte das Team von Anfang an unabhängig sein.
 - D) Falsch. Ein Batch wird gemäß der Priorität der Backlog-Einträge (Backlog Items) geplant. Die Verarbeitung vieler unerwarteter Anträge sollte nie Ziel von DevOps sein. Die Anträge sollten in das Backlog aufgenommen und priorisiert werden. Erst wenn dies geschehen ist, sollten sie in der nächsten Iteration aufgegriffen und verarbeitet werden.

32 / 40

Ein Team arbeitet in einwöchigen Iterationen und stößt dabei häufig auf Engpässe.

Wie sollte das Team **am besten** reagieren, wenn ein Engpass identifiziert wird?

- A) Das Team sollte den Engpass baldmöglichst beseitigen, nachdem er identifiziert wurde.
 - B) Das Team sollte nur die Iteration, in der ein Engpass festgestellt wurde, verlängern.
 - C) Das Team sollte die reguläre Anzahl von Aufgaben im Batch begrenzen, um die Batchgröße zu reduzieren.
 - D) Das Team sollte Visualisierungstools in Kombination mit einer WIP-Begrenzung (WIP-Limit) einsetzen.
-
- A) Richtig. Das Team sollte schnellstens nach Möglichkeiten suchen, um die Ursache des Engpasses zu beseitigen. Ist der Engpass beseitigt, dann könnte die Arbeit eventuell sogar noch in der Iteration fertiggestellt werden, in der sie versprochen wurde. Häufig ist dies jedoch nicht der Fall. (Literatur: A, Kapitel 4.11)
 - B) Falsch. Dies ist in diesem Fall nicht die beste Lösung für das Problem. Zwar können Iterationen bei Scrum gelegentlich verlängert werden, aber DevOps legt größeren Wert auf die Einführung eines regelmäßigen Rhythmus als Scrum. Die Verlängerung der Iteration sollte daher der letzte Ausweg sein.
 - C) Falsch. Eine Begrenzung der Batch-Größe hilft, die Probleme zu identifizieren, die den Engpass verursachen. Sie sollte jedoch im Rahmen der DevOps-Praktiken und nicht erst dann umgesetzt werden, wenn das Team einen Engpass identifiziert hat.
 - D) Falsch. Dies trägt zwar zur Identifizierung von Engpässen bei, sollte aber nicht erst umgesetzt werden, nachdem das Team einen Engpass identifiziert hat.

33 / 40

Wann kann die Nutzung von DevOps bei organisatorischen und technischen Änderungen zu Chaos und Kontrollverlust führen?

- A) Wenn das Kerngeschäft des Unternehmens in hohem Maße von der Informationstechnologie abhängt
 - B) Wenn komplexe Organisationen, chronische Probleme lösen wollen
 - C) Wenn die Organisation schnelle Änderungen braucht, um neue Geschäftsideen zu testen
 - D) Wenn sich die von der Organisation genutzte Informationstechnologie schnell ändert
-
- A) Falsch. Unternehmen, deren Kerngeschäft stark von der Informationstechnologie abhängt, sollten sich für DevOps interessieren.
 - B) Richtig. In komplexen Situationen bietet DevOps wahrscheinlich wenig Nutzen und definitiv keine schnellen Erfolge. Chronische Probleme sollten vorsichtig, überlegt und vernünftig gelöst werden. Man sollte sich nicht einfach darauf verlassen, dass DevOps ein Zaubermittel ist, das alle Probleme löst. (Literatur: A, Kapitel 5.1)
 - C) Falsch. Organisationen, deren Kerngeschäft schnelle Änderungen verlangt, um neue Geschäftsideen oder -hypothesen zu testen, sollten sich für DevOps zu interessieren.
 - D) Falsch. Organisationen, deren Informationstechnologie sich sehr schnell ändert, sollten sich für DevOps interessieren.

34 / 40

Es gibt viele Gründe, warum sich eine Organisation für DevOps interessieren sollte.

Wann sollten Unternehmen damit beginnen, sich für DevOps zu interessieren?

- A) Wenn Agile-Praktiken nicht mehr zur Organisation zu passen scheinen
 - B) Wenn die notwendigen Ergebnisse sich mit keiner anderen Methode erreichen lassen
 - C) Wenn Scrum- und Lean-Praktiken umgesetzt sind
-
- A) Falsch. Dies sollte eher Auslöser dafür sein, die Einführung von DevOps in einer Organisation nochmals ernsthaft zu überdenken. Es schadet zwar nichts, sich DevOps-Praktiken nochmals genauer anzusehen, wenn Agile-Praktiken für ein Unternehmen nicht relevant erscheinen, aber da DevOps-Praktiken viele Konzepte von Agile umfassen, ist DevOps für diese Organisation möglicherweise nicht geeignet.
 - B) Richtig. Unternehmen sollten damit beginnen, sich für DevOps zu interessieren, wenn alle anderen bewährten Methoden der Produktivitätssteigerung keine nennenswerten Ergebnisse mehr erzielen. Dies ist zwar nicht der beste Grund, um mit der Nutzung von DevOps-Praktiken zu beginnen, aber wenn nichts anderes mehr funktioniert, sollte man sich DevOps definitiv einmal näher ansehen. (Literatur: A, Kapitel 5.1)
 - C) Falsch. Um mit DevOps-Praktiken zu beginnen, muss man nicht warten bis Scrum- und Lean-Praktiken umgesetzt sind. Viele DevOps-Praktiken bauen auf Scrum- und Lean-Konzepten auf. Die Praktiken sollten nahtlos miteinander funktionieren. Bei DevOps kann die Organisation wählen, was für sie am besten funktioniert.

35 / 40

Was kann bei der Einführung von DevOps zu Schwierigkeiten führen?

- A) Funktionsübergreifende Teams
 - B) Begrenzte Nutzung von Virtualisierung
 - C) Microservice-Architektur
-
- A) Falsch. DevOps-Teams sind funktionsübergreifend; funktionsübergreifende Teams verschaffen Organisationen einen Vorsprung bei DevOps.
 - B) Richtig. Organisationen, die Virtualisierung nur in geringem Umfang nutzen, werden bei der Umsetzung von DevOps-Praktiken auf Schwierigkeiten stoßen. (Literatur: A, Kapitel 5.1)
 - C) Falsch. Die Microservice-Architektur hat sich aus einer Reihe von DevOps-Ideologien entwickelt, die Organisationen zu einem Vorsprung bei DevOps verhelfen.

36 / 40

Ein IT-System wird aktuell noch von vielen Mitarbeitern als Ganzes entwickelt und gepflegt.

Wo sind Schwierigkeiten bei der Einführung von DevOps-Praktiken zu erwarten?

- A) Bei der Zuweisung der DevOps-Teams zu getrennten Zuständigkeitsbereichen
 - B) Bei der Schaffung funktionsübergreifender Teams zur Arbeit an organisatorischen Strukturen
 - C) Bei der Pflege und Versionierung mehrerer Application Programming Interfaces (APIs) zur Gewährleistung der Abwärtskompatibilität
-
- A) Richtig. Eine monolithische, starre IT-Architektur stellt bei der Umsetzung von DevOps-Praktiken ein signifikantes Hindernis dar. Führt man kleine Teams ein, so muss man jedem dieser Teams einen separaten Zuständigkeitsbereich zuweisen können. In einer Situation, in der das betreffende IT-System noch von Dutzenden oder Hunderten von Mitarbeitern als Ganzes entwickelt wird, ist es schwierig, Teile für individuelle und unabhängige Teams abzutrennen, die asynchron voneinander arbeiten. (Literatur: A, Kapitel 5.1)
 - B) Falsch. Bei der Bildung funktionsübergreifender Teams gibt es keine Schwierigkeiten.
 - C) Falsch. Bei einer monolithischen Anwendung müssen die Entwickler lediglich den Klassennamen und die API ändern. Bei Microservices müssen die Entwickler die Versionsnummer der API ändern und mehrere API pflegen, um die Abwärtskompatibilität sicherzustellen. In diesem Fall wurden bislang noch keine Microservices umgesetzt, so dass hier noch keine Schwierigkeiten zu erwarten sind. Außerdem kann ein System zur Versionskontrolle erforderlich sein. Dies stellt jedoch kein Problem dar, sondern ist ein Fortschritt und eines der Ergebnisse, die von der Umsetzung der DevOps-Praktiken erwartet wird. Letztendlich ist es ein Vorteil.

37 / 40

Handelsübliche Standardsoftware (COTS) wird eingesetzt, um schnell die erforderlichen Ergebnisse zu erzielen, da die Entwicklung individueller Software Zeit in Anspruch nimmt.

Welche Aussage zu COTS ist **korrekt**?

- A) COTS beseitigt in der Regel die Integrationskomplexität.
 - B) COTS senkt die Gesamtbetriebskosten für das Unternehmen.
 - C) COTS erfordert Anpassungen, um das System zu konfigurieren.
 - D) COTS sollte zur Unterstützung strategischer Geschäftsbereiche eingesetzt werden.
-
- A) Falsch. Handelsübliche Standardsoftware (COTS) schränkt die Flexibilität immer ein. Es ist nicht immer möglich, das System so zu konfigurieren, wie es das Team wünscht oder benötigt.
 - B) Falsch. Die Konfiguration solcher Software nimmt Zeit in Anspruch und verursacht daher zusätzlichen Aufwand und zusätzliche Kosten. Zudem wird der Nutzen, den das Unternehmen tatsächlich benötigt, möglicherweise nicht erzielt.
 - C) Richtig. Handelsübliche Standardsoftware (COTS) lässt sich mithilfe von Skripten konfigurieren. Allerdings kann es Einschränkungen geben, und die Konfiguration des Systems nimmt dennoch Zeit in Anspruch. (Literatur: A, Kapitel 5.2)
 - D) Falsch. Handelsübliche Standardsoftware (COTS) unterstützt strategische Geschäftsbereiche nicht automatisch. Dafür sind Skripte erforderlich, sofern dies überhaupt möglich ist. Die Nutzung handelsüblicher Standardsoftware für strategische Geschäftsbereiche wird nicht empfohlen.

38 / 40

Was ist eine der Schwierigkeiten einer starren oder monolithischen IT-Architektur?

- A) Die Änderung und Entwicklung der IT-Architektur selbst ist schwierig zu bewerkstelligen.
 - B) Die Änderung der Services innerhalb der Architektur erfolgt unabhängig.
 - C) Die Aktualisierung auf eine neue Version ohne Deaktivierung der aktuellen Version.
 - D) Das Warten bis alle Komponenten für eine große Migration bereit sind.
-
- A) Richtig. Dies ist eines der Probleme einer starren IT-Architektur. Je größer und starrer die Architektur, umso schwieriger wird es, irgendetwas zu ändern und dabei im Blick zu behalten, was diese Änderung an der restlichen Architektur bewirkt. (Literatur: A, Kapitel 5.3)
 - B) Falsch. Dies ist eine der Lösungen für das Problem. Können alle Services unabhängig voneinander geändert werden, dann ist die Architektur nicht mehr starr.
 - C) Falsch. Dies kann auch bei nicht starren Architekturen eine Herausforderung darstellen.
 - D) Falsch. Die Durchführung einer großen Migration und zu warten bis alle Komponenten bereit dafür sind, bedeutet nicht automatisch, dass es sich um eine starre Architektur handelt. Dies könnte bei jeder Art von Architektur passieren.

39 / 40

Laut Empfehlung sollen Organisationen DevOps-Praktiken auswählen, die bei ihnen funktionieren und diese dann an ihre Bedürfnisse anpassen. Zu diesem Zweck müssen organisationsspezifische Fragen gestellt und organisationsspezifische Antworten gefunden werden.

Warum ist dies eine gute Idee?

- A) Weil DevOps-Veröffentlichungen nicht immer der Realität entsprechen und zu wenig über Schwierigkeiten und Fehler berichten
 - B) Weil dies die beste Möglichkeit ist, um ein DevOps-Ingenieur zu werden, der für die Umsetzung von DevOps eingestellt wird
 - C) Weil die Managementteams am besten wissen, wie sie DevOps in der eigenen Organisation am besten implementieren können.
 - D) Weil es zu wenige Veröffentlichungen und Veranstaltungen zu DevOps gibt, um sich unabhängige Meinungen zu bilden.
-
- A) Richtig. Der bei weitem größte Teil an Fachliteratur bereitet das Team nicht unbedingt auf alle Schwierigkeiten und Fehler vor, die in der Realität auftreten können. Wichtig ist, dass man die Informationen entsprechend filtert und dann entscheidet, was zu der Situation, in der sich die Organisation befindet. (Literatur: A, Kapitel 5.6)
 - B) Falsch. DevOps wird nicht 'implementiert' und einen Ingenieur, der eingestellt wird, um diese neue Ordnung in der IT einzuführen, gibt es nicht.
 - C) Falsch. Eine Implementierung von DevOps gibt es nicht.
 - D) Falsch. Ganz im Gegenteil. Es gibt viele Veröffentlichungen und Veranstaltungen, die die Organisation in ihrem Entscheidungsprozess unterstützen.

40 / 40

Eine Organisation hat eine alte IT-Infrastruktur. Die Organisation möchte DevOps einführen.

Welcher Ansatz wird in einer solchen Organisation häufig gewählt?

- A) Man beginnt die Implementierung von DevOps als Softwareprodukt, installiert es und startet es.
 - B) Man beginnt die Implementierung mit einer grundlegenden Pipeline, die zumindest den Bau und erste Tests durchführt.
 - C) Man beginnt die Implementierung mit der Auswahl des Produkts, das die größten Möglichkeiten zur Optimierung bietet.
 - D) Man beginnt die Implementierung mit der Identifizierung der Systeme, die lose mit anderen gekoppelt sind.
 - E) Man beginnt die Implementierung, indem man einen bestimmten Teil der Arbeitszeit für die Verbesserung zuteilt.
-
- A) Falsch. DevOps ist kein Softwareprodukt, das installiert und gestartet werden kann.
 - B) Falsch. Dies ist kein Ansatz für die Einführung von DevOps. Dies ist der Ansatz zum Bau einer Deployment Pipeline für den automatisierbaren Teil des Wertstroms.
 - C) Falsch. Dies beschreibt den Beginn der Entwicklung der Wertstromanalyse.
 - D) Richtig. DevOps kann an jedem beliebigen Punkt ansetzen, an dem sich das Unternehmen aktuell befindet. Die Identifizierung der lose miteinander verbundenen Systeme ist der erste Schritt. (Literatur: A, Kapitel 5.6)
 - E) Falsch. Dies bezieht sich auf die technischen Schulden.

Beurteilung

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

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



EXIN



Certified for what's next

Kontakt EXIN

www.exin.com

