



Voorbeeldexamen

Editie 201901

Copyright © EXIN Holding B.V. 2019. 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.



Inhoud

Introductie	4
Voorbeeldexamen	5
Antwoordsleutel	26
Evaluatie	62

Introductie

Dit is het voorbeeldexamen voor het examen EXIN DevOps Master™ (DEVOPSM.NL). Op dit examen is het Reglement voor de Examens van EXIN van toepassing.

Dit voorbeeldexamen bestaat uit 50 meerkeuzevragen. Elke vraag heeft een aantal antwoorden, waarvan er één correct is.

Het maximaal aantal te behalen punten is 50. Elke goed beantwoorde vraag levert u 1 punt op. Bij 33 punten of meer bent u geslaagd.

De beschikbare tijd is 120 minuten.

Veel succes!

Voorbeeldexamen

1 / 50

Wat is een goede reden om DevOps te implementeren in een organisatie?

- A) DevOps heeft regelmatige feedbackcyclussen voor nieuwe services, omdat de ontwikkelsnelheid hoger ligt.
- B) DevOps verbetert de continuïteit en flexibiliteit van het bedrijf, omdat processen worden geoptimaliseerd en er alleen activiteiten met toegevoegde waarde plaatsvinden.
- C) DevOps levert de nieuwe services Just-in-Time, omdat software-releases regelmatig plaatsvinden.
- D) Dankzij DevOps is er minder verspilling (waste) bij de samenwerking tussen Development en Operations, omdat deze twee worden geïntegreerd.

2 / 50

De CTO denkt dat het toepassen van bepaalde Lean-concepten het meest effectief is bij het introduceren van DevOps.

Welke Lean-principes of -werkwijzen zijn het **meest** effectief bij het implementeren van DevOps?

- A) **Kaizen en 5S.** Omdat Agile en DevOps gebaseerd zijn op kernconcepten van Lean, en Kaizen en 5S de basis vormen van Lean, zijn deze het meest effectief bij het implementeren van DevOps.
- B) **Kaizen in advance.** DevOps heeft feedback nodig van Operations naar Development. Kaizen in advance creëert een up-stream "feedbackloop", waarmee dit principe gemakkelijker kan worden toegepast in DevOps.
- C) **Obeya-systeem.** DevOps integreert verschillende managementstijlen. Het Obeya-systeem helpt het gehele proces te visualiseren, waardoor DevOps gemakkelijker kan worden geïntroduceerd.
- D) **One-piece-flow en JKK.** DevOps heeft profijt van up-stream processen en een single-value-stream flow. One-piece-flow maakt dat mogelijk en JKK helpt bij het stroomlijnen en implementeren van de flow.

3 / 50

Wat moet er, vergeleken met een conventioneel project, veranderen om een project te laten slagen in DevOps?

- A) Er moet een leveringsketen voor IT-services worden gebouwd, die werkt met een pull-systeem en one-piece-flow.
- B) Ontwikkelaars moeten zich voegen bij het Operations-team om snel onderhoud aan services mogelijk te maken.
- C) Operations moet werken voor het Development-team. Daarom heet het immers DevOps.
- D) Teamleden uit het Operations-team moeten zich voegen bij het Development-team.

4 / 50

Elk team dat overstapt op DevOps, kan worden omschreven als een compact.

Welk principe kan het **beste** worden toegepast om een compact te laten werken?

- A) Dynamische samenwerking
- B) Constante communicatie
- C) Vermindering van cyclussen
- D) Gedeelde verantwoordelijkheid

5 / 50

Er zijn veel kennisbronnen, standaarden en werkwijzen beschikbaar voor de introductie van DevOps.

Welke bron wordt als cruciaal beschouwd voor een geslaagde DevOps-introductie?

- A) CMMI-niveau 3
- B) Gedisciplineerde Agile
- C) ISO/IEC 20000-1
- D) PMI PMBok

6 / 50

Samenwerking is een van de vier pijlers van effectieve DevOps.

Waarom is samenwerking zo belangrijk?

- A) Samenwerking is de enige manier om een one-piece-flow te bereiken in DevOps.
- B) De waarde van DevOps voor het bedrijf wordt gerealiseerd door samenwerking tussen teams.
- C) DevOps-organisaties zijn klein, dus de teams moeten goed samenwerken.
- D) Samenwerking stimuleert verandering en is een 'easy win'. Bovendien nemen de ontwikkelkosten hierdoor af.

7 / 50

U bent eigenaar van een klein DevOps-bedrijf met 5 medewerkers dat mobiele apps maakt voor kinderen met een beperking. Een van de apps waar uw team het meest trots op is, is een app voor kinderen met autisme, waarmee kinderen zelf hun dag kunnen plannen.

Omdat de app een groot succes was, wordt uw team gevraagd de functies ervan uit te breiden voor andere mensen die mogelijk baat hebben bij een goede dagplanning. Dit verzoek leidt gegarandeerd tot meer complexiteit in de code en diverse technische uitdagingen voor het team.

U verwacht erg goed betaald te krijgen voor deze opdracht en daarom neemt u deze aan. Na een paar weken heeft uw team ruzie. U wordt ook kwaad en begint beter te letten op wat alle teamleden doen. U werkt regelmatig met hen samen en kunt zo fouten in de code makkelijk corrigeren wanneer u deze opmerkt.

Hoewel er genoeg werk wordt verricht, blijft het team ruziën. U beseft dat een deel van de DevOps-mentaliteit ontbreekt.

Wat is de **beste** strategie om uw probleem op te lossen?

- A) Leden aan het team toevoegen om meer diversiteit te creëren, door complementair, niet-technisch personeel aan te nemen
- B) DevOps-experts van buiten uw bedrijf vragen om op te treden als sponsor en mentor van uw teamleden
- C) Een gemeenschappelijk doel bepalen en samen naar dat doel toe werken om de samenwerking te vergroten
- D) Enkele maanden vrijmaken om eerst aan team-building te doen en elkaar beter te leren kennen

8 / 50

U vindt dat uw Development-team een echt team is.

Wat is een duidelijk teken dat het echt een team en niet zomaar een groep is?

- A) Het team volgt de regels die tijdens de teamvergaderingen overeen zijn gekomen.
- B) Het team heeft effectieve vergaderingen die het team zelf leidt.
- C) Het team werkt in een vast en gestaag tempo naar het gemeenschappelijke doel.
- D) Het team lost problemen op door het verantwoordelijke teamlid te bevragen.

9 / 50

AppAtoZ groeit razendsnel met het ontwikkelen en implementeren van mobiele applicaties voor het iPhone- en Android-platform.

De Development-teams van deze startup staan onder enorme druk om de verbeteringen in hun huidige mobiele applicaties zo snel mogelijk uit te rollen. De teamleden hebben het afgelopen half jaar gemiddeld 60 uur per week gewerkt. De leiding neemt liever geen extra mensen aan, maar houdt zich vooral bezig met het vergroten van de omzet en het terugdringen van de operationele en ontwikkelkosten.

De laatste maanden is het aantal afwezigheidsmeldingen van het personeel toegenomen, melden werknemers zich ziek en hebben enkelen zelfs ontslag genomen, waardoor de werkbelasting voor de overige werknemers alleen maar is toegenomen. Het terughalen van personeel en aannemen van nieuwe mensen neemt de druk niet snel weg voor het Development-team.

Welke **langetermijnstrategieën** moeten worden overwogen om burn-out en stress van het personeel te bestrijden?

- A)
 - Voeg een combinatie van vaste werknemers en mensen met een tijdelijk contract toe aan het team om de werkbelasting beter aan te kunnen
 - Laat de leiding en het Development-team factoren in de werkomgeving identificeren die bijdragen aan burn-out
 - Stel een plan op om hieraan te werken
- B)
 - Ontsla het managementteam vanwege onbekwaamheid
 - Neem een competentere managementteam aan dat een juiste cultuur en balans tussen werk en privé kan creëren die duurzaam en realistisch zijn
 - Geef het Development-team meer tijd om de juiste balans te vinden
- C)
 - Las een time-out in voor de leden van het Development-team
 - Neem ontwikkelaars aan voor het managen van de toegenomen werkbelasting tijdens piekperiodes in de loop van het jaar
 - Bied ontwikkelaars de juiste professionele mentale hulp
- D)
 - Las een time-out in voor de leden van het Development-team
 - Laat de leiding en het Development-team factoren in de werkomgeving identificeren die bijdragen aan burn-out
 - Stel een plan op om hieraan te werken

10 / 50

U werkt als ontwikkelaar in een DevOps-team en probeert de samenwerking tussen al uw teamleden te bevorderen. In uw team zitten twee mannelijke senior ontwikkelaars en een vrouwelijke junior systeembeheerder.

Het begin was lastig, omdat de teamleden niet met elkaar konden opschieten. U hebt toen ingegrepen en geprobeerd hen beter met elkaar door één deur te laten gaan. Dat werkte, want het geruzie stopte en er werd veel meer werk verricht.

Echter, nu begint u een patroon op te vallen: de junior beheerder is het voortdurend eens met de senior ontwikkelaars.

Hoe moet u reageren op dit patroon?

- A) Zolang de teamleden geen ruzie maken, moet u de situatie laten voor wat het is. Ze kunnen het eindelijk met elkaar vinden en dit moet u niet verstoren.
- B) Het is belangrijk dat Operations alle mogelijke twijfels over builds uitspreekt. U moet de beheerder daarom coachen, zodat zij minder inschikkelijk is en assertiever wordt.
- C) De senior ontwikkelaars hebben meer verantwoordelijkheid. U moet hen daarom vragen aardiger te zijn en zelf eventuele Operations-twijfels te bedenken als de junior medewerker dit niet doet.
- D) U moet voorlopig gaan werken in het Operations-team, ook al bent u zelf ontwikkelaar. Zo kunt u de junior beheerder laten zien hoe het moet.

11 / 50

Waarom beloven Agile en Scrum snellere en beter voorspelbare softwareontwikkeling?

- A) Omdat ze het mogelijk maken voorafgaand aan het ontwerpen beter en completer vereisten te verzamelen en te verwerken
- B) Omdat ze kleine, autonome, zelfsturende en zelfplannende teams mogelijk maken
- C) Omdat ze de Product Owner de mogelijkheid bieden om deel te nemen aan dagelijkse standup meetings
- D) Omdat ze de projectmanager de mogelijkheid bieden om snel prioriteiten te wijzigen wanneer dit nodig is

12 / 50

Wat is lichtgewicht ITSM?

- A) Een op de bedrijfscontinuïteit gerichte ITSM
- B) Een nieuwe ITIL-versie die als standaard wordt voorgesteld
- C) Een slechte introductie van ITIL-processen
- D) Een op release management gerichte ITSM

13 / 50

U werkt voor een bedrijf dat Lean- en Agile-werkwijzen heeft geïmplementeerd. De CEO is niet overtuigd van de waarde van alweer een nieuwe introductie: DevOps.

U bent een DevOps-expert. U bent van mening dat DevOps gunstig is voor elk bedrijf dat software ontwikkelt.

Wat voegt DevOps toe aan het bedrijf waarvoor u werkt?

- A) Het ontwikkelen van betere user stories en functionele vereisten
- B) Het doen van minder releases, zodat klanten niet worden opgezadeld met regelmatige updates
- C) Het trainen van professionals in zowel Development- als Operations-werk
- D) Het schrijven van code die sneller wordt gereleased en zo de resultaten voor het bedrijf ondersteunt

14 / 50

Senior management vraagt de IT-afdeling om betere ondersteuning van het bedrijf en betere afstemming op de zakelijke doelstellingen. Als CIO neemt u diverse maatregelen, waaronder het verminderen van de management werklast voor Operations.

Hoe kan DevOps u het **beste** helpen om het bestaande service level management (SLM) te verbeteren?

- A) Door van de best practices voor ITSM af te stappen, omdat deze als aanpak behoorlijk zwaar zijn
- B) Door betere operating level agreements (OLA's) tussen Dev en Ops op te stellen
- C) Door een nieuwe set ITSM-processen op basis van ITIL te implementeren
- D) Door ITSM voortaan te laten werken met een set minimaal vereiste informatie (MRI, Minimum Required Information)

15 / 50

Welke DevOps-introductie is het **meest** geschikt voor een onderneming die de System of Record-aanpak (SoR, registratiesysteem) gebruikt?

- A) Samenwerking
- B) Continuous delivery (continue levering)
- C) De Toyota-manier

16 / 50

Wat is het **grootste** voordeel van het gebruik van het Obeya-systeem?

- A) Het faciliteert klachten van klanten, zodat het team gegarandeerd voldoende feedback krijgt voor continuous improvement (continu verbeteren).
- B) Het pakt stress binnen teams aan, zodat teamleden een houdbaar tempo kunnen aanhouden.
- C) Het verbetert de dagelijkse bug-rapportage, waardoor er minder werk opnieuw hoeft te worden gedaan en er minder bugs worden doorgegeven in de productielijn.
- D) Het bevordert snelle besluitvorming op basis van de huidige status, doordat informatie snel wordt verzameld en gedeeld.

17 / 50

U werkt bij een softwarebedrijf dat onlangs is overgestapt op DevOps. U beseft dat niet alleen de cultuur van de organisatie moet veranderen, maar ook de werkwijzen en tools. Het bedrijf monitort de softwareprojecten al op mogelijke problemen.

U stelt voor de volgende maatregelen toe te voegen:

- automatisering van het change management-proces;
- introductie van toegangscontrole om te voorkomen dat iemand wijzigingen aanbrengt zonder dat hiervoor goedkeuring is gegeven.

Waarom zijn deze maatregelen noodzakelijk?

- A) Automatisering maakt het mogelijk om wijzigingen sneller te implementeren terwijl de betrouwbaarheid intact blijft. Toegangscontrole is noodzakelijk om probleemoplossende heuristiek en ongeplande serviceonderbrekingen te voorkomen.
- B) Automatisering is noodzakelijk om het tempo van wijzigingen te vertragen. Met toegangscontrole wordt voorkomen dat klanten de software zelf wijzigen, zonder dat u het weet en zonder controle.
- C) Automatisering maakt Operations overbodig. Toegangscontrole moet worden geïmplementeerd, omdat DevOps-projecten in een cloudomgeving plaatsvinden, wat meer risico's met zich meebrengt.

18 / 50

Virtualisatie en cloud computing zijn technieken die DevOps-werkwijzen kunnen ondersteunen en faciliteren.

Op welke manier komen deze technieken DevOps ten goede?

- A) Cloud computing maakt toegang op afstand mogelijk, waardoor klanten meer controle krijgen.
- B) Virtuele omgevingen kunnen makkelijker worden gestandaardiseerd en gebruiken hardware efficiënter.
- C) Gevirtualiseerde infrastructuur is makkelijker te begrijpen en vereist geen onderhoud.

19 / 50

SLA's (service level agreements) zijn voor elk project van belang, omdat u hierin de afspraken met de klant vastlegt. In DevOps hebben SLA's echter nog een belangrijk doel.

Wat is dit doel?

- A) Klanten zijn verantwoordelijk voor het opstellen van de SLA voor het DevOps-team. De SLA vervangt daarmee de formele taak als verantwoordelijkheid van Operations.
- B) In de SLA kan de klant al diens niet-functionele vereisten opgeven, zodat Development zich daarop kan richten.
- C) In de SLA wordt het acceptabele service level opgegeven. Development moet de SLA begrijpen en Operations ondersteunen bij de naleving ervan.

20 / 50

Voor een nieuw product moet uw team een deployment pipeline (uitrolpijplijn) ontwikkelen. Als onderdeel van continuous integration (continue integratie) moet u de commit-fase van de pipeline definiëren. U bespreekt deze fase met uw teamleden.

De Process Master zegt daarover: "De Definition-of-Done (definitie van klaar) moet worden opgesteld tijdens of vóór de commit-fase. Wanneer code vóór die tijd niet Done is, moet het werk worden stopgezet".

Is dat waar?

- A) Ja. Als het werk niet Done is, heeft de Process Master zijn of haar werk niet goed uitgevoerd. Dit moet onmiddellijk worden opgelost.
- B) Ja. Werk dat niet Done is, moet niet worden doorgestuurd naar de commit-fase. Doorsturen draagt niet bij aan de waarde voor de klant.
- C) Nee. De Definition-of-Done wordt alleen opgesteld tijdens bijeenkomsten met de klant. Daarop wachten vertraagt het werk te veel.
- D) Nee. Werk in een deployment pipeline moet altijd doorgaan. Als code niet Done is, hoeft deze alleen maar inactief te zijn.

21 / 50

Wat is het **grootste** voordeel van het vergroten van de diversiteit van het team, wat leidt tot een breder scala van persoonlijke achtergronden en culturen?

- A) Het brengt meer ervaring en meer standpunten in.
- B) Het leidt tot minder spanningen binnen het team.
- C) Het beperkt originaliteit en de mogelijkheid om nieuwe inzichten aan te dragen.
- D) Het duurt langer om tot een specifieke beslissing te komen.

22 / 50

U werkt voor een klein bedrijf dat één DevOps-team heeft. Uw DevOps-team werkt aan een applicatie die uit meerdere componenten bestaat. Sommige componenten zijn nieuw en andere hebben alleen updates nodig.

Momenteel heeft elke component een eigen deployment pipeline (uitrolpijplijn). Het team is trots op de continuous delivery (continue levering) binnen de meeste pipelines. Bovendien is de productie hoog en van goede kwaliteit.

Wat moet u doen?

- A) De verschillende pipelines behouden en het team stimuleren om continuous delivery uit te breiden
- B) Alleen pipelines met continuous delivery behouden en de overige pipelines zonder continuous delivery samenvoegen
- C) Slechts twee verschillende pipelines behouden: één voor ontwikkeling en één voor onderhoud
- D) Met het team toewerken naar één pipeline door uit te leggen wat de risico's zijn van meerdere pipelines

23 / 50

U beoordeelt Company Builders, een middelgrote tot grote organisatie die enkele jaren geleden is overgestapt op de DevOps-aanpak.

Het bedrijf heeft u ingehuurd om de huidige volwassenheid van DevOps vast te stellen. Op basis van uw bevindingen geeft u suggesties voor verbetering. Het bedrijf wil weten op welk gebied het zich moet richten om *Niveau 2 - Kwantitatief beheerd* te bereiken.

U hebt vastgesteld dat de meeste gebieden binnen *Niveau 1 - Consistent* vallen, met twee uitzonderingen:

1. **Omgevingen en deployment (uitrol)**. Dit gebied beheert speciaal ingerichte introducties en heeft release- en roll-backprocessen getest.
2. **Build management en continuous integration (continue integratie)**. In dit gebied treft u regulier geautomatiseerde builds en tests aan en elke build kan opnieuw worden gegenereerd op basis van bronbeheer met behulp van een geautomatiseerd proces.

Bepaal de mate van volwassenheid in deze twee gebieden op basis van de hier verstrekte informatie. Geef vervolgens uw aanbevelingen voor de aandachtspunten ter verbetering.

Aan welke van deze twee gebieden moet Company Builders werken om door te kunnen groeien naar Niveau 2?

- A) Zowel omgevingen en deployment als build management en continuous integration bevinden zich op niveau 0. Het werk moet op beide gebieden tegelijkertijd plaatsvinden.
- B) Zowel omgevingen en deployment als build management en continuous integration hebben niveau 1 of hoger. Het werk moet in de overige gebieden worden uitgevoerd om door te groeien.
- C) Omgevingen en deployment heeft niveau 0. Build management en continuous integration heeft niveau 1. Omgevingen en deployment moet als eerste worden onderzocht.
- D) Omgevingen en deployment heeft niveau 2. Build management en continuous integration heeft niveau 0. Build management en continuous integration moet als enige worden onderzocht.

24 / 50

Uw bedrijf verkoopt een online-back-upservice aan haar klanten. Een van die klanten heeft u nu gevraagd nieuwe functies te implementeren in uw service. De klant wil die nieuwe functies binnen een week, anders gaat hij op zoek naar een andere leverancier.

U vindt de nieuwe functies belangrijk en weet dat het Development-team ze snel kan bouwen. Echter, u loopt tegen de volgende problemen aan:

- Het duurt lang voordat testers bugs sluiten.
- Testers vinden bugs die ontwikkelaars al lang geleden hebben opgelost.
- De applicatie kan zelden in goed werkende staat worden gedemonstreerd.
- Er worden zelden showcases gegeven.

Wat is uw probleem en hoe lost u dit op?

- A)** U deployt code die bugs bevat. Dit moet u oplossen door de kennis van het deploymentproces (uitrolproces) te vergroten, de samenwerking te verbeteren en gedisciplineerder te werken.
- B)** U hebt gebrekkig configuration management. Dit moet u oplossen door de samenwerking tussen Development en Operations te vergroten, monitoring en logboekregistratie te verbeteren en meer gebruik te maken van virtualisatie.
- C)** Uw proces voor continuous integration (continue integratie) wordt niet goed beheerd. Dit moet u oplossen door het tempo van geautomatiseerde tests en de commit-fase te verkorten, en de kennis van het proces voor continuous integration te vergroten.
- D)** Uw teststrategie is niet effectief. Dit moet u oplossen door tests te automatiseren en de samenwerking tussen de testers en de rest van het team te verbeteren.

25 / 50

Wat is een voordeel van continuous integration (continue integratie) binnen effectief DevOps?

- A)** Uitgebreide testcyclussen voorafgaand aan de releases van de functies
- B)** Lange perioden tussen de releases van de functies
- C)** Regelmatige en tijdige releases van de functies

26 / 50

Een multinational ervaart de nodige moeilijkheden om commits uit hun gedistribueerde locaties samen te voegen in een centrale repository die zich bevindt in Dallas, VS. De gedistribueerde locaties zijn Mexico-Stad, Parijs, San Diego en Engeland. Er is geen consistentie in de manier waarop de regio's hun commits uitvoeren en soms is het niet duidelijk of bepaalde tests wel of niet zijn gelukt.

Er bestaan vier mogelijke werkwijzen:

1. Check niet in bij een vastgelopen build.
2. Voer alle commit-tests altijd eerst lokaal uit of laat uw continuous integration-server dat voor u doen.
3. Wacht totdat de commit-tests zijn geslaagd voordat u verdergaat.
4. Doe geen "comment-out" bij mislukte tests.

Welke van deze werkwijzen zorgen er het **beste** voor dat de gedistribueerde locaties de huidige pijnpunten zullen aanpakken?

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

27 / 50

Neem de anatomie van een elementaire deployment pipeline (uitrolpijplijn) in aanmerking.

Welke fase bevestigt dat het systeem op zowel het functionele als niet-functionele niveau werkt?

- A) Geautomatiseerde acceptatietest
- B) Build- en unittest
- C) Handmatige acceptatietest
- D) Versiebeheer

28 / 50

Het is een best practice in DevOps om hetzelfde deploymentproces (uitrolproces) te gebruiken voor elke omgeving waarin uw toepassing wordt uitgevoerd. Dit zorgt ervoor dat de build effectief wordt getest. U gebruikt scripts om uw build- en deploymentproces te automatiseren.

Wat is de **beste** manier om dat te doen?

- A) Gebruik één script voor elke omgeving en onderhoud de scripts als onderdeel van het versiebeheersysteem.
- B) Gebruik voor elke omgeving één specifiek script om recht te doen aan de verschillen tussen de omgevingen.
- C) Gebruik dezelfde scripts voor elke omgeving en breng handmatig parameters aan voor specifieke configuraties.
- D) Gebruik dezelfde scripts voor elke omgeving en beheer de configuratiegegevens afzonderlijk.

29 / 50

Na de release van een nieuwe IT-service stopt een actieve taak onverwachts.

Wat is **geen** mogelijke oorzaak hiervan?

- A) De Gatekeeper heeft geen rekening gehouden met de samenhang tussen werkitems en de service acceptance criteria (SAC).
- B) De niet-functionele vereisten waren niet duidelijk wegens een onvolledige user story.
- C) De Service Master heeft met de gebruiker geen uitfasering afgesproken voor de service in de release.
- D) Er is te weinig werk verricht tijdens de ontwikkelprocessen, wat ook blijkt uit de SAC.

30 / 50

Bedrijf AppBC maakt gebruik van DevOps. Het bedrijf heeft continuous deployment (continue uitrol) en een goede deployment pipeline geïmplementeerd, met sterk geautomatiseerde acceptatietests. Dagelijks wordt nieuwe software aan productie geleverd.

AppBC heeft een grote database en veel gebruikers. Er bestaat een uitgebreide en solide strategie voor capaciteitstesten. Omdat de omgeving groot en complex is, komen er bij elke nieuwe versie wel enkele bugs in productie terecht.

Wat is voor AppBC de **beste** strategie om deze bugs te voorkomen?

- A) Kanarie releases instellen
- B) Capaciteitstesten automatiseren
- C) De leveringsfrequentie verlagen
- D) Blue-green-deployment gebruiken

31 / 50

DevOps maakt gebruik van zeer belangrijke concepten uit Agile, die weer zijn afgeleid van het Toyota-productiesysteem.

Waarom is one-piece-flow belangrijk voor het gebruiken van DevOps?

- A) Omdat een team hiermee in een beheersbaar tempo met een voorspelbare velocity kan werken.
- B) Omdat een team zich hiermee kan richten op het toevoegen van functies met een zo groot mogelijke waarde voor het bedrijf.
- C) Omdat verschillende teamleden hiermee in hogere mate de verantwoordelijkheid voor werk delen.
- D) Omdat hiermee minder knelpunten ontstaan, doordat het aantal gelijktijdige taken wordt beperkt.

32 / 50

De S Corporation, een middelgrote leverancier van auto-onderdelen, is toeleverancier van een grote onderneming, T Motors Corporation. Het bedrijf levert auto-onderdelen aan T Motors Corporation, wat optelt tot bijna 60% van de totale verkoop.

Het bestuur vergadert over het nieuwe partnerschap. T Motors Corporation eist van S Corporation dat de wijze van levering wordt omgezet in Just-in-Time levering, anders stoppen ze hun samenwerking met S Corporation. Omdat S Corporation het verlies aan omzet niet zou overleven, maakt dit de omschakeling naar Just-in-Time tot een urgente kwestie. Omdat deze wijziging binnen 6 maanden moet plaatsvinden, zijn er hooguit 5 maanden voor de voorbereiding.

Een van de zaken die moeten worden geïmplementeerd is het volgen van onderdelen via radio-frequency identification (RFID). Dit moet bijdragen aan een transparant productieproces. Er moet een snelle controle worden uitgevoerd van het huidige proces om de overstap naar een RFID-proces te vereenvoudigen.

De CIO wordt gevraagd het veranderingsproces te leiden. Zij gelooft dat dit mogelijk is als een DevOps-benadering wordt gebruikt om een minimale release te realiseren. Idealiter moet het concept van productie met RFID als eerste worden ontwikkeld. Als laatste moet een productiecontrolesysteem met gebruikmaking van de RFID-gegevens worden geïmplementeerd. Er is echter te weinig tijd om die stappen opeenvolgend uit te voeren. Daarom moeten deze drie dingen gelijktijdig worden uitgevoerd.

De CIO wijst het project toe aan Em, een Scrum Master. Development werkt aan de voorbereiding voor een deployment pipeline (uitrolpijlijn).

Em ziet dat Development enthousiast is en hard werkt, maar het team zou meer discipline kunnen gebruiken. Bovendien moet de releasefrequentie omhoog.

Waar moet Em zich als **eerste** op richten?

- A) Em moet zich vooral richten op communicatie, omdat dit het belangrijkste aspect in DevOps is. Em moet eerst het ijs breken en enkele communicatieregels voor het team opstellen.
- B) Em moet met team eerst het gebruik van value stream mapping (een waardestromeschema) bespreken en one-piece-flow invoeren, omdat flow en een gestroomlijnd proces bijzonder belangrijk zijn.
- C) Em moet met de teamleden eerst de infrastructuur en de werkomgeving bespreken, omdat DevOps met name effectief is wanneer alle tools en werkwijzen werken.
- D) Em moet eerst alle stakeholders bij elkaar roepen, ze opleiden in DevOps en hen vragen of ze mee willen helpen de cultuurverandering te verspreiden omdat een dergelijke verandering nodig is voor DevOps.

33 / 50

Uw DevOps-team werkt goed samen in een beheersbaar tempo. Door voldoende speling in te bouwen in het proces, heeft het team de benodigde tijd en concentratie om builds zorgvuldig te controleren en testen. Het testen en deployen (uitrollen) vindt momenteel handmatig plaats. Het tempo ligt hoog genoeg om regelmatig hoge waarde te leveren aan het bedrijf.

De CEO vraagt u nu om advies over automatisering binnen dit team.

Welk advies moet u geven?

- A) Automatiseer zo veel mogelijk, zodat het team meer functies kan toevoegen en sneller waarde voor het bedrijf kan laten zien
- B) Automatiseer de acceptatietests maar niet het deployen, aangezien het handmatige proces daarvoor veiliger is
- C) Automatiseer het deployen om de cyclustijd te verbeteren, maar niet de tests, zodat het team kan leren van bugs
- D) Automatiseer niets aan deze methoden van dit team, aangezien de huidige methoden fantastische resultaten opleveren

34 / 50

Een CIO wijst een project toe aan haar meest betrouwbare medewerker, Michael, een Scrum Master. Het Development-team bereidt het bouwen van een deployment pipeline (uitrolpijplijn) voor.

Michael heeft alle vertrouwen in de goede voornemens en spontaniteit van het Development-team, maar zou het team wel graag gedisciplineerder zien. Bovendien moet de releasefrequentie omhoog. Michael wil dat het Development-team vaker kan releasen.

Een van de teamleden zegt daarop: "Het belangrijkste voor de nieuwe deployment pipeline is dat deze moet worden geautomatiseerd. Dat moet dus als eerste gebeuren".

Is deze bewering juist?

- A) Ja, dat is juist. Het automatiseren van de deployment pipeline is de belangrijkste factor bij het verhogen van de efficiëntie.
- B) Ja, dat is juist. Door een deployment pipeline te implementeren die geautomatiseerd is, worden potentiële problemen voorkomen.
- C) Nee, dat is niet juist. Het realiseren van een one-piece-flow met een solide deploymentproces (uitrolproces) dient de voornaamste prioriteit te zijn. Automatisering van het proces kan in een later stadium plaatsvinden.
- D) Nee, dat is niet juist. Niet de deployment pipeline, maar het testproces moet als eerste worden geautomatiseerd.

35 / 50

Uw bedrijf verandert van werkwijze en gaat DevOps inzetten. Uw team doet mee met deze verandering. U bespreekt best practices voor de fase waarin code wordt doorgevoerd.

Uw collega Sun zegt daarover: "Wanneer een build vastloopt en niemand daarvoor de verantwoordelijkheid neemt, moeten we uitzoeken om welke persoon het gaat en hem of haar daarop aanspreken, zodat de build kan worden hersteld."

Is dat een goed idee?

- A) Ja. Alleen degene die het vastlopen van een build veroorzaakt, kan dat herstellen. Zoek daarom uit om wie het gaat, ook al zal die persoon zich daarbij ongemakkelijk voelen.
- B) Ja. Je moet altijd de verantwoordelijkheid nemen voor het laten vastlopen van een build. Doe je dat niet, dan hebben je collega's het recht je daarop aan te spreken.
- C) Nee. DevOps is gebaseerd op een verwijtloze cultuur. Als een collega geen verantwoordelijkheid neemt, moet hij of zij daar niet toe gedwongen worden.
- D) Nee. U moet de build eerst herstellen. Zoek dan pas uit wie de verantwoordelijke is en leg de juiste sanctie op.

36 / 50

Het Development-team van X-AppGo heeft te maken met talloze uitdagingen bij hun huidige testaanpak. Momenteel hanteert het bedrijf een handmatig proces bij acceptatietests. De ontwikkelaars zijn van mening dat het door hen samengestelde pakket unittests degelijk genoeg is om bescherming te bieden tegen regressies.

Het Development-team moet bij elke release 1 miljoen euro besteden aan handmatige acceptatietests. De bedrijfsleiding heeft besloten dat het Development-team geautomatiseerde acceptatietests moet implementeren om de algehele kosten voor testen te verlagen en tegelijkertijd het aantal defecten en regressies dat in de productieomgeving terechtkomt tot een minimum te beperken.

Welke principes moeten worden gehanteerd bij het definiëren van acceptatiecriteria voor uw applicatie met automatisering als uitgangspunt?

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

37 / 50

Wat is het **meest** effectieve mechanisme om gegevens op geautomatiseerde wijze te migreren?

- A) Het creëren van een schema voor databaseversies en dit bewaren onder versiebeheer
- B) Het creëren en beheren van kleinere datasets, zodat de migratie gemakkelijker wordt
- C) Het zekerstellen dat alle scripts goed zijn getest alvorens de gegevens te migreren
- D) Het zekerstellen dat u beschikt over een roll-backprocedure voor het geval dat de migratie mislukt

38 / 50

Bedrijf X-AppGo ondervindt problemen met de roll-backprocessen. Dit resulteert vaak in verlies van kritieke gegevens in de databases van de productieapplicatie.

Wanneer is het **niet** mogelijk om roll-backscripts uit te voeren zonder dat kritieke gegevens verloren gaan?

- A) Wanneer met het roll-backscript gegevens worden verwijderd die alleen door de nieuwe versie worden gebruikt.
- B) Wanneer met het roll-backscript een kolom wordt verplaatst tussen tabellen.
- C) Wanneer met het roll-backscript gegevens uit tijdelijke tabellen weer worden toegevoegd.

39 / 50

Na het deployen (uitrollen) van software-upgrades voor een applicatie en het vervangen van routers en switches, heeft bedrijf ACMECONST veel last van applicatie- en hardwarestoringen.

Toen deze storingen zich voordeden tijdens de onderhoudsperiode, bleek het erg lastig om herstel naar de oorspronkelijke staat uit te voeren. Dit leidde tot langere hersteltijden, die ver buiten de normale onderhoudsperiode vielen en tot meer downtime voor kritieke applicaties.

Geautomatiseerde voorzieningen en een autonome infrastructuur kunnen in deze situatie van pas komen, maar hierbij gelden enkele kanttekeningen.

Welke aspecten moeten zorgvuldig worden beheerd om het risico op onderbrekingen te beperken bij het deployen naar de productieomgeving?

- A) Gedetailleerde monitoring-logs om fouten tijdens applicatie-upgrades op te lossen
- B) Externe integratiepunten, zoals externe systemen en services
- C) Serverconfiguraties en onderliggende informatie over gebruikersaccounts
- D) De toolset voor automatisering en een autonome architectuur

40 / 50

Bedrijf X-AppGo ondervindt problemen met zijn kernapplicatie. De applicatie communiceert niet goed met andere externe applicaties. Deze externe applicaties moeten op effectieve wijze specifieke gegevensvariabelen verkrijgen, zodat specifieke aanroepen kunnen worden uitgevoerd. De kernapplicatie wordt ontwikkeld door één team en het bedrijf wil dat om zeer goede zakelijke redenen zo houden.

Een van de ontwikkelaars stelt voor een component uit de X-AppGo-codebasis te isoleren om zo de communicatieproblemen aan te pakken.

Wat is in deze situatie een goede reden om een component te isoleren?

- A) Om een set plug-ins in de codebase te converteren naar een monolithische codebasis
- B) Om de impact van wijzigingen te beperken en het makkelijker te maken om de codebasis te wijzigen
- C) Om de codebasis van X-AppGo op te splitsen en door verschillende teams te laten beheren
- D) Er is geen goede reden hiervoor en hierdoor gaat het compileren langer duren

41 / 50

Zelfs de kleinste applicaties zijn afhankelijk van andere componenten of bibliotheken. Daarom is het begrijpen en beheren van afhankelijkheden een belangrijke activiteit binnen continuous deployment (continue uitrol) om de flow in de deployment pipeline (uitrolpijplijn) te houden.

U hebt een applicatie gemaakt die gebruikmaakt van twee bibliotheken. Elk van deze bibliotheken is afhankelijk van een derde, onderliggende bibliotheek. Ze verwijzen echter allebei naar een andere versie van die derde bibliotheek. Hierdoor ontstaat een specifieke afhankelijkheid.

Wat is de **beste** oplossing om deze afhankelijkheid te verhelpen of voorkomen?

- A) Alle bibliotheken samenvoegen tot één enkele bibliotheek, zodat u rechtstreeks naar die bibliotheek kunt verwijzen en zo het probleem kunt voorkomen
- B) De bibliotheken beheren met behulp van versiebeheer, zodat u het direct ziet als u dit type afhankelijkheid maakt
- C) Met post-its op een groot bord een visueel overzicht van alle afhankelijkheden bijhouden, zodat u de flow kunt volgen
- D) Slechts kleine delen van de toolketen controleren, zodat u problemen die zich mogelijk voordien bij het inchecken, makkelijk kunt debuggen

42 / 50

In een omgeving met continuous deployment (continue uitrol) is het belangrijk dat alle versies worden beheerd, zodat u snel fouten kunt vinden of indien nodig een roll-back kunt uitvoeren.

Het wordt echter **niet** aanbevolen om binaire output te bewaren in versiebeheer.

Waarom wordt deze uitzondering gemaakt?

- A) Binaire output staat doorgaans in grote bestanden die voor elke build veranderen en automatisch worden bijgewerkt.
- B) Er werken meerdere teamleden aan de binaire bestanden, dus het is niet praktisch om deze te bewaren binnen versiebeheer.
- C) De binaire output is de input voor compilers, die al worden bewaard in versiebeheer.
- D) Dit is niet nodig, aangezien hercompilatie deel uitmaakt van het normale buildproces.

43 / 50

U wilt een holistische aanpak gebruiken voor het beheren van uw volledige IT-infrastructuur.

Op welke twee principes kan deze aanpak het **beste** worden gebaseerd?

- A)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd op basis van configuratie onder change management.
 - 2. U moet altijd op de hoogte zijn van de actuele status van uw infrastructuur door middel van monitoring en event management.
- B)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd op basis van configuratie onder change management.
 - 2. U moet altijd op de hoogte zijn van de actuele status van uw infrastructuur door middel van instrumentatie en van incident management.
- C)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd door middel van configuratie onder versiebeheer.
 - 2. U moet altijd op de hoogte zijn van de actuele status door middel van incident- en event management.
- D)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd door middel van configuratie onder versiebeheer.
 - 2. U moet altijd op de hoogte zijn van de actuele status van uw infrastructuur door middel van instrumentatie en van monitoring.

44 / 50

Teams die goed samenwerken, beschikken over gesynchroniseerde werktickets. Een CTO (Chief Technology Officer) heeft aan de hand van 'Go and See' onderzocht hoe het Operations-team functioneert. Na de release voert het Operations-team altijd een herdefinitie van de operationele infrastructuur uit.

Wat is het **beste** advies om deze werkwijze te verbeteren?

- A) Het team moet niets doen. Er is geen verbetering mogelijk, omdat herdefinitie een stap is die altijd moet worden uitgevoerd.
- B) Het team moet een manier zien te vinden om de operationele infrastructuur en de toegangscontrole van de operationele omgeving te modelleren.
- C) Het team moet de operationele infrastructuur reviewen, zodat dit een geautomatiseerd proces wordt.
- D) Het team moet eerst de eigen kennis van het deploymentproces (uitrolproces) gaan delen met het Development-team.

45 / 50

Op welk moment kan Operations het **beste** Development op de hoogte te stellen van een operationele verandering?

- A) Development hoeft daarvan niet op de hoogte te worden gesteld. Operationele veranderingen zijn alleen bedoeld voor het Operations-team.
- B) Onmiddellijk. Development moet zo snel mogelijk op de hoogte worden gesteld.
- C) Tijdens de Scrum of Scrums meeting de volgende ochtend.
- D) Nadat het Operations-team de acceptatietests heeft uitgevoerd.

46 / 50

U wilt dat uw DevOps-organisatie doorgroeit. Dit kan op veel manieren.

Welke manier is **niet** geschikt om uw DevOps-organisatie te helpen doorgroeien?

- A) Definieer heldere doelstellingen als mijlpalen aan de hand waarvan uw teamleden kunnen bepalen of hun dagelijkse activiteiten waardevol zijn.
- B) Definieer processen duidelijk en ondersteun de teamleden om het proces elke dag weer te verbeteren.
- C) Houd verslagen bij van alle vergaderingen, zodat uw teamleden eenvoudig toegang tot alle communicatie hebben.
- D) Bewaak dagelijkse activiteiten en leg deze vast om kleine dagelijkse successen te kunnen vaststellen en deze te vieren.

47 / 50

U werkt bij een IT-serviceprovider. Als onderdeel van het bedrijfscontinuïteitsplan wilt u de garantie inbouwen dat u altijd voldoet aan de overeengekomen minimale service levels.

U wilt de continuïteit van de IT-service waarborgen.

Hoe kan DevOps u daarbij helpen?

- A) De culturele DevOps-waarden affiniteit en samenwerking zorgen ervoor dat service hoog in het vaandel staat bij de DevOps-teamleden.
- B) DevOps bereidt het team voor op Obeya-werkwijzen en aanpak bij calamiteiten door opzettelijk chaos in het systeem in te voeren.
- C) Risicobeperkende maatregelen en herstelopties zijn hoogstwaarschijnlijk al in de code opgenomen, omdat Operations samenwerkt met Development.
- D) Service level management wordt belangrijker in DevOps, omdat het de taak van de Process Master is om dit te bewaken.

48 / 50

ACMECONST heeft zijn mondiale aanwezigheid sterk uitgebreid door meer mensen aan te nemen en het aantal engineeringteams wereldwijd te vergroten. Bovendien heeft het bedrijf het klantenbestand met liefst 30% per jaar vergroot.

Beslissingen die ooit gemakkelijk werden genomen toen het engineeringteam zich nog in dezelfde ruimte bevond, nemen nu veel meer tijd in beslag, wat in de organisatie voor de nodige frustratie zorgt. Voor goedkeuring moeten er meer managementlagen worden doorlopen en het proces is veelomvattender geworden. Daardoor raken veel engineers gedesillustioneerd over het gehele besluitvormingsproces.

Bovendien is er meer verwarring ontstaan rondom de verantwoordelijkheid voor de verschillende problemen die naar voren worden gebracht, waardoor men terughoudender is geworden in het nemen van beslissingen. De engineers vinden ook dat hun creativiteit wordt tegengewerkt door de extra processen en bureaucratische lijnen, en dat begint hun moreel aan te tasten.

Wat is de **beste** manier om hiermee om te gaan?

- A) Laat de huidige processen intact, maar wees duidelijk over de rollen, verantwoordelijkheid en eigenschap voor elk proces, stel een effectieve methode in voor de afweging van productiviteit versus risico's, pas veranderingen stapsgewijs toe en creëer veilige plekken om te experimenteren.
- B) Neem processen onder de loep om vast te stellen waar zaken kunnen worden gestroomlijnd, wees duidelijk over de rollen, verantwoordelijkheid en eigenschap voor elk proces, stel een effectieve methode in voor de afweging van productiviteit versus risico's, pas veranderingen stapsgewijs toe en creëer veilige plekken om te experimenteren.
- C) Neem processen onder de loep om vast te stellen waar zaken kunnen worden gestroomlijnd, wees duidelijk over de rollen, verantwoordelijkheid en eigenschap voor elk proces, stel een effectieve methode in voor de afweging van productiviteit versus risico's, pas veranderingen stapsgewijs toe en breng de hoeveelheid experimenten terug tot een minimum om onnodige fouten met applicaties te voorkomen.

49 / 50

Er bestaat binnen bedrijf X-AppGo een conflict tussen het Operations-team in Colombia en het Development-team in Ierland, omdat ze verschillende prioriteiten en doelstellingen hebben. Daardoor kost het steeds meer tijd en moeite om problemen op te lossen die van invloed zijn op het bedrijf.

Welke werkwijzen zou X-AppGo kunnen overwegen om het conflict te verminderen en de samenwerking tussen de Development- en Operations-teams te verbeteren?

- A)
 - 1. Laat de Development- en Operations-teams onafhankelijk van elkaar werken als ze dat liever hebben, om conflicten te vermijden.
 - 2. Zorg ervoor dat de bedrijfsleiding volledig instemt met de ondersteuning van de Development- en Operations-teams.
- B)
 - 1. Laat iemand van de bedrijfsleiding met het DevOps-team gaan praten over het belang van samenwerking.
 - 2. Train de Development- en Operations-teams in DevOps-werkwijzen, zodat ze leren hoe ze elkaars werk kunnen uitvoeren.
- C)
 - 1. Zorg ervoor dat de Development- en Operations-teams andere bedrijven bezoeken waar DevOps goed werkt.
 - 2. Vergroot de financiële middelen om de toegenomen aanspraak waar zowel het Operations- als het Development-team mee te maken heeft, te ondersteunen.
- D)
 - 1. Beveel onderlinge bedrijfsbezoeken aan tussen de Development- en Operations-teams om de verstandhouding te verbeteren en het vertrouwen en begrip te vergroten.
 - 2. Wissel kennis uit tussen de Development- en Operations-teams zodat ze effectiever kunnen samenwerken.

50 / 50

Een Development-team is geïnteresseerd in DevOps. Het team is voornamelijk geïnteresseerd in continuous integration (continue integratie, CI). Het team houdt zich momenteel bezig met het ontwikkelen en onderhouden van 3 grote en 4 kleinere oplossingen. Daarbij wordt gebruikgemaakt van de Scrum-werkwijze. Elke sprint duurt 4 weken, waarbij gemiddeld om de 10 of 15 dagen 1 gecommiteerde release voor de testomgeving wordt gemaakt en per maand 1 release voor productie. Het team wil een kwalitatieve business case voor het management opstellen ter ondersteuning van de investeringen en inspanningen om een CI-werkwijze te creëren.

Welke tastbare voordelen van CI helpen die business case daarbij het **beste**?

- A) Eenmaal per dag een deployment (uitrol) in de testomgeving uitvoeren kan voordelig zijn voor het bedrijf en kan de ontwikkelkosten aanzienlijk verlagen.
- B) Dit draagt bij aan de teamgeest. Omdat er al gebruik wordt gemaakt van Scrum, genereert CI **geen** meetbare voordelen voor het bedrijf.
- C) Het verhoogt de stabiliteit en kwaliteit van releases met betere en geautomatiseerde tests, waardoor de algehele releasesnelheid toeneemt.
- D) Eenmaal per dag een release voor productie uitbrengen kan voordelig zijn voor het bedrijf en kan de ontwikkelkosten aanzienlijk verlagen.

Antwoordsleutel

1 / 50

Wat is een goede reden om DevOps te implementeren in een organisatie?

- A) DevOps heeft regelmatige feedbackcyclussen voor nieuwe services, omdat de ontwikkelsnelheid hoger ligt.
 - B) DevOps verbetert de continuïteit en flexibiliteit van het bedrijf, omdat processen worden geoptimaliseerd en er alleen activiteiten met toegevoegde waarde plaatsvinden.
 - C) DevOps levert de nieuwe services Just-in-Time, omdat software-releases regelmatig plaatsvinden.
 - D) Dankzij DevOps is er minder waste (verspilling) bij de samenwerking tussen Development en Operations, omdat deze twee worden geïntegreerd.
-
- A) Onjuist. Feedbackcyclussen zijn regelmatig wanneer de ontwikkelsnelheid hoger ligt, maar dat komt doorgaans door Scrum of een andere Agile-methode, en niet door DevOps op zich.
 - B) Juist. Het toevoegen van waarde en het optimaliseren van processen zijn de sleutel tot het verbeteren van de continuïteit en de flexibiliteit van een bedrijf. Onthoud dat IT-services altijd het bedrijf moeten ondersteunen – dat zijn de waarde en het doel van DevOps. (*Literatuur: C, hoofdstuk 2*)
 - C) Onjuist. Just-in-Time leveringen zijn fantastisch, maar zijn op zich geen goede reden om DevOps te implementeren. Als dit het doel is, is introductie van Lean-processen een beter idee.
 - D) Onjuist. Wanneer twee teams eenvoudigweg worden samengevoegd, betekent dat nog niet dat de Muda (verspilling, waste) gegarandeerd ophoudt. Om een einde te maken aan verspilling moeten beide teams hun werkwijzen veranderen.

2 / 50

De CTO denkt dat het toepassen van bepaalde Lean-concepten het meest effectief is bij het implementeren van DevOps.

Welke Lean-principes of -werkwijzen zijn het **meest** effectief bij het introduceren van DevOps?

- A) **Kaizen en 5S.** Omdat Agile en DevOps gebaseerd zijn op kernconcepten van Lean, en Kaizen en 5S de basis vormen van Lean, zijn deze het meest effectief bij het implementeren van DevOps.
 - B) **Kaizen in advance.** DevOps heeft feedback nodig van Operations naar Development. Kaizen in advance creëert een up-stream “feedbackloop”, waarmee dit principe gemakkelijker kan worden toegepast in DevOps.
 - C) **Obeya-systeem.** DevOps integreert verschillende managementstijl. Het Obeya-systeem helpt het gehele proces te visualiseren, waardoor DevOps gemakkelijker kan worden geïntroduceerd.
 - D) **One-piece-flow en JKK.** DevOps heeft profijt van up-stream processen en een single-value-stream flow. One-piece-flow maakt dat mogelijk en JKK helpt bij het stroomlijnen en implementeren van de flow.
- A) Onjuist. Hoewel Lean, Agile en DevOps met elkaar samenhangen, zijn Kaizen en 5S niet het meest geschikt voor een succesvolle start van DevOps. Nadat DevOps is geïmplementeerd, kan Kaizen worden gebruikt voor continue verbetering en 5S voor het hanteren van de juiste werkwijzen. Maar beide komen pas na een succesvolle introductie van DevOps.
- B) Onjuist. Feedback is altijd welkom, maar dat biedt geen garantie op een effectieve toepassing van Lean bij de introductie van DevOps.
- C) Onjuist. Visualisatie kan handig zijn, maar dit is niet de meest effectieve Lean-werkwijze bij het implementeren van DevOps.
- D) Juist. Het bouwen van een werkbare, one-piece Deployment Pipeline draagt bij aan een succesvolle introductie van DevOps. Het belangrijkste bij DevOps is het bouwen van up-stream processen van Development tot Operations, en dan met name voor één specifieke Deployment Pipeline. JKK is daarvoor het meest effectieve gedrag op de werkvloer. (*Literatuur: C, hoofdstuk 4*)

3 / 50

Wat moet er, vergeleken met een conventioneel project, veranderen om een project te laten slagen in DevOps?

- A) Er moet een leveringsketen voor IT-services worden gebouwd, die werkt met een pull-systeem en one-piece-flow.
 - B) Ontwikkelaars moeten zich voegen bij het Operations-team om snel onderhoud aan services mogelijk te maken.
 - C) Operations moet werken voor het Development-team. Daarom heet het immers DevOps.
 - D) Teamleden uit het Operations-team moeten zich voegen bij het Development-team.
- A) Juist. Een project is geslaagd wanneer voor het proces een op Jidoka gebaseerd pull-systeem wordt gebruikt om waardevolle IT-services (of -producten) te maken. (*Literatuur: C, hoofdstuk 4 en Literatuur B, hoofdstuk 1*)
- B) Onjuist. DevOps betekent niet dat er een ontwikkelaar bij Operations komt.
- C) Onjuist. Alleen door samen te werken wordt het project nog geen DevOps-succes.
- D) Onjuist. DevOps betekent niet simpelweg dat Operations bij Development komt.

4 / 50

Elk team dat overstapt op DevOps, kan worden omschreven als een compact.

Welk principe kan het **beste** worden toegepast om een compact te laten werken?

- A) Dynamische samenwerking
- B) Constante communicatie
- C) Vermindering van cyclussen
- D) Gedeelde verantwoordelijkheid

- A) Onjuist. Dynamische samenwerking is geen compact-principe.
- B) Juist. De principes van een DevOps-compact zijn: contante communicatie, gedeelde en duidelijk gedefinieerde doelen, dynamische aanpassing en een plek van gedeeld begrip. (*Literatuur: A, hoofdstuk 2*)
- C) Onjuist. Vermindering van cyclussen is geen compact-principe.
- D) Onjuist. Gedeelde verantwoordelijkheid is geen compact-principe.

5 / 50

Er zijn veel kennisbronnen, standaarden en werkwijzen beschikbaar voor de introductie van DevOps.

Welke bron wordt als cruciaal beschouwd voor een geslaagde DevOps-introductie?

- A) CMMI-niveau 3
- B) Gedisciplineerde Agile
- C) ISO/IEC 20000-1
- D) PMI PMBok

- A) Onjuist. CMMI-certificering kan van pas komen, maar is niet cruciaal voor een DevOps-introductie, laat staan een specifiek CMMI-niveau.
- B) Juist. Gedisciplineerde Agile is cruciaal voor een geslaagde DevOps-introductie. (*Literatuur: C, hoofdstuk 4i*)
- C) Onjuist. ISO/IEC 20000-1-certificering kan van pas komen, maar is niet cruciaal voor een DevOps-introductie.
- D) Onjuist. PMI PMBok is een naslagwerk voor projectmanagement dat niet cruciaal is voor een DevOps-introductie.

6 / 50

Samenwerking is een van de vier pijlers van effectieve DevOps.

Waarom is samenwerking zo belangrijk?

- A) Samenwerking is de enige manier om een one-piece-flow te bereiken in DevOps.
 - B) De waarde van DevOps voor het bedrijf wordt gerealiseerd door samenwerking tussen teams.
 - C) DevOps-organisaties zijn klein, dus de teams moeten goed samenwerken.
 - D) Samenwerking stimuleert verandering en is een 'easy win'. Bovendien nemen de ontwikkelkosten hierdoor af.
-
- A) Onjuist. Een one-piece-flow is ook mogelijk zonder samenwerking. Het is een ander mechanisme.
 - B) Juist. Samenwerking tussen alle betrokken teams (inclusief Development en Operations) is van cruciaal belang om een hogere bedrijfswaarde te bereiken door meer communicatie, automatisering en software van betere kwaliteit. (*Literatuur: A, hoofdstukken 6 en 7*)
 - C) Onjuist. DevOps-organisaties kunnen heel groot zijn. Teams moeten samenwerken om voldoende toegevoegde waarde voor het bedrijf te creëren.
 - D) Onjuist. Tools stimuleren verandering en zijn 'easy wins'. Samenwerking is juist lastig. Door samenwerking kunnen de ontwikkelkosten dalen, maar dit is niet het primaire doel. Het doel is de kwaliteit te verbeteren.

7 / 50

U bent eigenaar van een klein DevOps-bedrijf met 5 medewerkers dat mobiele apps maakt voor kinderen met een beperking. Een van de apps waar uw team het meest trots op is, is een app voor kinderen met autisme, waarmee kinderen zelf hun dag kunnen plannen.

Omdat de app een groot succes was, wordt uw team gevraagd de functies ervan uit te breiden voor andere mensen die mogelijk baat hebben bij een goede dagplanning. Dit verzoek leidt gegarandeerd tot meer complexiteit in de code en diverse technische uitdagingen voor het team.

U verwacht erg goed betaald te krijgen voor deze opdracht en daarom neemt u deze aan. Na een paar weken heeft uw team ruzie. U wordt ook kwaad en begint beter te letten op wat alle teamleden doen. U werkt regelmatig met hen samen en kunt zo fouten in de code makkelijk corrigeren wanneer u deze opmerkt.

Hoewel er genoeg werk wordt verricht, blijft het team ruziën. U beseft dat een deel van de DevOps-mentaliteit ontbreekt.

Wat is de **beste** strategie om uw probleem op te lossen?

- A) Leden aan het team toevoegen om meer diversiteit te creëren, door complementair, niet-technisch personeel aan te nemen
 - B) DevOps-experts van buiten uw bedrijf vragen om op te treden als sponsor en mentor van uw teamleden
 - C) Een gemeenschappelijk doel bepalen en samen naar dat doel toe werken om de samenwerking te vergroten
 - D) Enkele maanden vrijmaken om eerst aan team-building te doen en elkaar beter te leren kennen
- A) Onjuist. Uitbreiding van het team kan raadzaam zijn wanneer er geen expertise beschikbaar is voor het werk dat moet worden gedaan. Mensen zonder technische kennis inhuren is echter nooit raadzaam of wenselijk om het team efficiënt te houden. In dit geval wordt er bovendien genoeg werk verricht. Daarom zou dit de problemen die u nu al ervaart alleen maar vergroten. Er is namelijk geen vertrouwen en samenwerking.
- B) Onjuist. Dit is een uitstekend idee. Echter, hiermee wordt het ontbreken van samenwerking en affiniteit als onderdeel van de DevOps-mentaliteit niet opgelost. Samenwerking neemt toe wanneer mensen een doel delen en samen ergens in slagen.
- C) Juist. Dit is een goede manier om de huidige problemen op te lossen. De onderlinge strijd neemt hierdoor af, zonder dat het veel tijd kost en u mogelijk de opdracht verliest. Door doelen te delen ontstaan vertrouwen, affiniteit en samenwerking. (*Literatuur: A, hoofdstuk 7*)
- D) Onjuist. Hiermee ontstaat mogelijk een team dat elkaar vertrouwt en goed samenwerkt, wat in principe uw doel is. Maar het is niet de beste manier om het probleem op te lossen: u loopt zo namelijk het risico de opdracht te verliezen omdat het te lang duurt om de kwestie uit de wereld te helpen.

8 / 50

U vindt dat uw Development-team een echt team is.

Wat is een duidelijk teken dat het echt een team en niet zomaar een groep is?

- A) Het team volgt de regels die tijdens de teamvergaderingen overeen zijn gekomen.
 - B) Het team heeft effectieve vergaderingen die het team zelf leidt.
 - C) Het team werkt in een vast en gestaag tempo naar het gemeenschappelijke doel.
 - D) Het team lost problemen op door het verantwoordelijke teamlid te bevragen.
-
- A) Onjuist. Groepen kunnen heel goed zijn in het opvolgen van regels. Maar dat maakt een groep niet per se tot een team.
 - B) Onjuist. Groepen kunnen heel effectief vergaderen. Maar dat is niet per se een bewijs dat het een team is.
 - C) Juist. Een echt team zorgt voor een vast en gestaag werktempo en blijft aan het gemeenschappelijke doel werken. (*Literatuur: A, hoofdstuk 9*)
 - D) Onjuist. Teams lossen problemen in onderling overleg op en bevragen geen individuele leden. DevOps is gebaseerd op een niet-beschuldigende cultuur.

9 / 50

AppAtoZ groeit razendsnel met het ontwikkelen en implementeren van mobiele applicaties voor het iPhone- en Android-platform.

De Development-teams van deze startup staan onder enorme druk om de verbeteringen in hun huidige mobiele applicaties zo snel mogelijk uit te rollen. De teamleden hebben het afgelopen half jaar gemiddeld 60 uur per week gewerkt. De leiding neemt liever geen extra mensen aan, maar houdt zich vooral bezig met het vergroten van de omzet en het terugdringen van de operationele en ontwikkelkosten.

De laatste maanden is het aantal afwezigheidsmeldingen van het personeel toegenomen, melden werknemers zich ziek en hebben enkelen zelfs ontslag genomen, waardoor de werkbelasting voor de overige werknemers alleen maar is toegenomen. Het terughalen van personeel en aannemen van nieuwe mensen neemt de druk niet snel weg voor het Development-team.

Welke **langetermijnstrategieën** moeten worden overwogen om burn-out en stress van het personeel te bestrijden?

- A)
 - Voeg een combinatie van vaste werknemers en mensen met een tijdelijk contract toe aan het team om de werkbelasting beter aan te kunnen
 - Laat de leiding en het Development-team factoren in de werkomgeving identificeren die bijdragen aan burn-out
 - Stel een plan op om hieraan te werken
- B)
 - Ontsla het managementteam vanwege onbekwaamheid
 - Neem een competentere managementteam aan dat een juiste cultuur en balans tussen werk en privé kan creëren die duurzaam en realistisch zijn
 - Geef het Development-team meer tijd om de juiste balans te vinden
- C)
 - Las een time-out in voor de leden van het Development-team
 - Neem ontwikkelaars aan voor het managen van de toegenomen werkbelasting tijdens piekperiodes in de loop van het jaar
 - Bied ontwikkelaars de juiste professionele mentale hulp
- D)
 - Las een time-out in voor de leden van het Development-team
 - Laat de leiding en het Development-team factoren in de werkomgeving identificeren die bijdragen aan burn-out
 - Stel een plan op om hieraan te werken

- A) Juist. Dit is het enige antwoord waarvan alle keuzes op de lange termijn zijn gericht. De overige antwoorden bevatten minstens één keuze voor de korte termijn. (*Literatuur: A, hoofdstuk 8*)
- B) Onjuist. Ontslag van het managementteam draagt niet bij aan een oplossing van onderliggende problemen op de lange termijn. Het is hooguit een oplossing voor de korte termijn. De tijd gunnen om de juiste balans te vinden is een goed idee, maar zonder uitgewerkt plan zal dat niet zomaar gaan werken.
- C) Onjuist. Het inlassen van een time-out is een prima idee, maar dat werkt alleen op de korte termijn. Ook mentale hulp is prima, maar ook hier geldt dat het een oplossing voor de korte termijn is als er niets wordt gedaan aan de onderliggende oorzaken. Het aantrekken van meer ontwikkelaars kan werken en is een goed idee.
- D) Onjuist. Het opstellen van een plan en het identificeren van factoren in de werkomgeving zijn prima ideeën. Er is echter een beter antwoord, omdat een time-out slechts op de korte termijn werkt.

10 / 50

U werkt als ontwikkelaar in een DevOps-team en probeert de samenwerking tussen al uw teamleden te bevorderen. In uw team zitten twee mannelijke senior ontwikkelaars en een vrouwelijke junior systeembeheerder.

Het begin was lastig, omdat de teamleden niet met elkaar konden opschieten. U hebt toen ingegrepen en geprobeerd hen beter met elkaar door één deur te laten gaan. Dat werkte, want het geruzie stopte en er werd veel meer werk verricht.

Echter, nu begint u een patroon op te vallen: de junior beheerder is het voortdurend eens met de senior ontwikkelaars.

Hoe moet u reageren op dit patroon?

- A) Zolang de teamleden geen ruzie maken, moet u de situatie laten voor wat het is. Ze kunnen het eindelijk met elkaar vinden en dit moet u niet verstoren.
 - B) Het is belangrijk dat Operations alle mogelijke twijfels over builds uitspreekt. U moet de beheerder daarom coachen, zodat zij minder inschikkelijk is en assertiever wordt.
 - C) De senior ontwikkelaars hebben meer verantwoordelijkheid. U moet hen daarom vragen aardiger te zijn en zelf eventuele Operations-twijfels te bedenken als de junior medewerker dit niet doet.
 - D) U moet voorlopig gaan werken in het Operations-team, ook al bent u zelf ontwikkelaar. Zo kunt u de junior beheerder laten zien hoe het moet.
-
- A) Onjuist. Deze situatie vraagt om conflictoplossing. Het junior teamlid gebruikt inschikkelijkheid als stijl om het conflict op te lossen, en dat is geen productieve stijl. Elke functie moet haar verantwoordelijkheid nemen om te voorkomen dat bezorgdheden niet op tijd worden aangepakt in de build.
 - B) Juist. Dit is de juiste oplossing voor deze situatie. Deze situatie vraagt om conflictoplossing. Het junior teamlid gebruikt inschikkelijkheid als stijl om het conflict op te lossen, en dat is geen productieve stijl. Bovendien ontstaat hiermee het risico dat Operations eventuele twijfels over de build niet uitspreekt. (*Literatuur: A, hoofdstukken 7 en 14*)
 - C) Onjuist. Iedereen is in dezelfde mate verantwoordelijk. Dat is niet afhankelijk van iemands senioriteit of geslacht. Bovendien kunnen ontwikkelaars problemen met builds niet op dezelfde manier bekijken als operators. Daarom is interactie tussen de verschillende gebieden nodig in DevOps.
 - D) Onjuist. U kunt niet zomaar van team wisselen, ook al is 'laten zien hoe het moet' een uitstekende manier van coachen. Ontwikkelaars kunnen problemen met builds niet op dezelfde manier bekijken als operators, dus dit zal nadelig zijn voor de kwaliteit van de builds.

11 / 50

Waarom beloven Agile en Scrum snellere en beter voorspelbare softwareontwikkeling?

- A) Omdat ze het mogelijk maken voorafgaand aan het ontwerpen beter en completer vereisten te verzamelen en te verwerken
- B) Omdat ze kleine, autonome, zelfsturende en zelfplannende teams mogelijk maken
- C) Omdat ze de Product Owner de mogelijkheid bieden om deel te nemen aan dagelijkse standup meetings
- D) Omdat ze de projectmanager de mogelijkheid bieden om snel prioriteiten te wijzigen wanneer dit nodig is

- A) Onjuist. Dit is meer een aanpak voor een Waterfallmodel.
- B) Juist. Dit is de manier die in de Scrum- en Agile-principes wordt voorgesteld om sneller en beter te leveren. (*Literatuur: A, hoofdstuk 4*)
- C) Onjuist. Dat de Product Owner kan deelnemen, is niet relevant om de belofte van snellere en beter voorspelbare softwareontwikkeling waar te maken.
- D) Onjuist. De projectmanager mag de prioriteiten niet wijzigen, dat mag alleen de Product Owner.

12 / 50

Wat is lichtgewicht ITSM?

- A) Een op de bedrijfscontinuïteit gerichte ITSM
 - B) Een nieuwe ITIL-versie die als standaard wordt voorgesteld
 - C) Een slechte introductie van ITIL-processen
 - D) Een op release management gerichte ITSM
-
- A) Juist. ITIL lijkt zwaar en niet geschikt voor de snelle processen van DevOps. Lichtgewicht ITSM is ITSM aangepast voor DevOps gericht op bedrijfscontinuïteit met een minimale set aan vereiste informatie. (*Literatuur: C, hoofdstuk 4iii*)
 - B) Onjuist. Een dergelijke ITIL-versie is nog niet voorgesteld.
 - C) Onjuist. Lichtgewicht ITSM is niet zozeer een slechte implementatie, maar wel een uitgekledde versie, gericht op bedrijfscontinuïteit en bedoeld om de werkbelasting voor het management te verkleinen.
 - D) Onjuist. ITSM heeft te maken met service management, niet met release management. Binnen het ITSM-concept is release een proces dat de service ondersteunt.

13 / 50

U werkt voor een bedrijf dat Lean- en Agile-werkwijzen heeft geïmplementeerd. De CEO is niet overtuigd van de waarde van alweer een nieuwe introductie: DevOps.

U bent een DevOps-expert. U bent van mening dat DevOps gunstig is voor elk bedrijf dat software ontwikkelt.

Wat voegt DevOps toe aan het bedrijf waarvoor u werkt?

- A) Het ontwikkelen van betere user stories en functionele vereisten
 - B) Het doen van minder releases, zodat klanten niet worden opgezadeld met regelmatige updates
 - C) Het trainen van professionals in zowel Development- als Operations-werk
 - D) Het schrijven van code die sneller wordt gereleased en zo de resultaten voor het bedrijf ondersteunt
- A) Onjuist. Dankzij de Agile-werkwijzen schrijft u al uitstekende user stories en functionele vereisten. De Agile-principes liggen ten grondslag aan DevOps. Zelfs door al in een vroeg stadium Operations te betrekken, veranderen de user stories en functionele vereisten die zijn ontstaan door Agile te volgen, niet veel binnen DevOps.
- B) Onjuist. DevOps maakt gebruik van snellere, continue releases om sneller waarde te leveren voor het bedrijf. Dit is ook een Lean-principe. Regelmatige updates voor de klant zijn geen probleem, zolang de nieuwe functies toegevoegde waarde hebben. Bovendien is het met veel releases nog steeds mogelijk dat eindgebruikers alleen op geplande tijden updates uitvoeren.
- C) Onjuist. Het doel van DevOps is dat Operations en Development samenwerken, niet dat bepaalde personen beide rollen vervullen. Hoewel dit theoretisch mogelijk is in zeer kleine bedrijven, is het op zich geen DevOps.
- D) Juist. De Lean- en Agile-werkwijzen verzekeren dat de focus al ligt op de waarde voor het bedrijf en veranderende functieverzoeken. Als directe ondersteuning van de resultaten voor het bedrijf, voegt DevOps daar nog een verhoogde releasefrequentie naar productie aan toe. Dit is te danken aan de focus op een pipeline met continuous deployment (continue uitrol), waarbij Operations al in een vroeg stadium wordt betrokken en zo veel mogelijk werk wordt geautomatiseerd. (*Literatuur: C, hoofdstuk 1*)

14 / 50

Senior management vraagt de IT-afdeling om betere ondersteuning van het bedrijf en betere afstemming op de zakelijke doelstellingen. Als CIO neemt u diverse maatregelen, waaronder het verminderen van de management werklast voor Operations.

Hoe kan DevOps u het **beste** helpen om het bestaande service level management (SLM) te verbeteren?

- A) Door van de best practices voor ITSM af te stappen, omdat deze als aanpak behoorlijk zwaar zijn
 - B) Door betere operating level agreements (OLA's) tussen Dev en Ops op te stellen
 - C) Door een nieuwe set ITSM-processen op basis van ITIL te implementeren
 - D) Door ITSM voortaan te laten werken met een set minimaal vereiste informatie (MRI, Minimum Required Information)
- A) Onjuist. Het afstappen van best practices voor ITSM zal het SLM niet verbeteren.
- B) Onjuist. Door het opstellen van een betere OLA tussen Dev en Ops zal de werklast niet afnemen. Dit leidt juist tot meer bureaucratie, wat zeker niet de focus van DevOps is.
- C) Onjuist. Hierdoor ontstaan meer zware processen die niet geschikt zijn voor DevOps.
- D) Juist. Dit is een belangrijk vereiste voor een DevOps-introductie: het creëren van lichtgewicht ITSM. (*Literatuur: C, hoofdstuk 4iii*)

15 / 50

Welke DevOps-introductie is het **meest** geschikt voor een onderneming die de System of Record-aanpak (SoR, registratiesysteem) gebruikt?

- A) Samenwerking
 - B) Continuous delivery (continue levering)
 - C) De Toyota-manier
-
- A) Juist. Deze introductie richt zich enkel op het leveren van snelle en regelmatige IT-services en een betrouwbare werking daarvan. Daarom is deze het meest geschikt voor System of Engagement (SoE, verbindingssysteem) en SoR. (*Literatuur: C, hoofdstuk 8*)
 - B) Onjuist. Deze introductie is het meest geschikt voor verkopers van digitale producten.
 - C) Onjuist. Deze introductie is het meest geschikt voor IT-serviceproviders.

16 / 50

Wat is het **grootste** voordeel van het gebruik van het Obeya-systeem?

- A) Het faciliteert klachten van klanten, zodat het team gegarandeerd voldoende feedback krijgt voor continuous improvement (continu verbeteren).
 - B) Het pakt stress binnen teams aan, zodat teamleden een houdbaar tempo kunnen aanhouden.
 - C) Het verbetert de dagelijkse bug-rapportage, waardoor er minder werk opnieuw hoeft te worden gedaan en er minder bugs worden doorgegeven in de productielijn
 - D) Het bevordert snelle besluitvorming op basis van de huidige status, doordat informatie snel wordt verzameld en gedeeld.
-
- A) Onjuist. Obeya faciliteert geen klachten van klanten.
 - B) Onjuist. Hoewel Obeya van pas kan komen bij de omgang met stressvolle situaties, is dit niet het voornaamste voordeel ervan.
 - C) Onjuist. Obeya draagt niet bij aan een betere bug-rapportage.
 - D) Juist. De Obeya-ruimte of 'war room' is een tool uit het Toyota-productiesysteem/Lean dat zorgt dat projectteams een visueel overzicht hebben van alle relevante informatie. Daarnaast bevordert het snelle interactie en informatie-uitwisseling tussen kleine teams, waardoor er sneller informatie kan worden verzameld om beslissingen te nemen. (*Literatuur: C, hoofdstuk 7iii*)

17 / 50

U werkt bij een softwarebedrijf dat onlangs is overgestapt op DevOps. U beseft dat niet alleen de cultuur van de organisatie moet veranderen, maar ook de werkwijzen en tools. Het bedrijf monitort de softwareprojecten al op mogelijke problemen.

U stelt voor de volgende maatregelen toe te voegen:

- automatisering van het change management-proces;
- introductie van toegangscontrole om te voorkomen dat iemand wijzigingen aanbrengt zonder dat hiervoor goedkeuring is gegeven.

Waarom zijn deze maatregelen noodzakelijk?

- A)** Automatisering maakt het mogelijk om wijzigingen sneller te implementeren terwijl de betrouwbaarheid intact blijft. Toegangscontrole is noodzakelijk om probleemoplossende heuristiek en ongeplande serviceonderbrekingen te voorkomen.
- B)** Automatisering is noodzakelijk om het tempo van wijzigingen te vertragen. Met toegangscontrole wordt voorkomen dat klanten de software zelf wijzigen, zonder dat u het weet en zonder controle.
- C)** Automatisering maakt Operations overbodig. Toegangscontrole moet worden geïmplementeerd, omdat DevOps-projecten in een cloudomgeving plaatsvinden, wat meer risico's met zich meebrengt.
- A)** Juist. Citaat: "Hoewel we in het algemeen geen voorstander zijn van afscherming van onderdelen en het gebruik van goedkeuringsprocessen, is dit voor de productie-infrastructuur noodzakelijk. Omdat we vinden dat u met testomgevingen hetzelfde moet omgaan als met productieomgevingen, vloeit daaruit voort dat dit proces op beide omgevingstypen moet worden toegepast. Afscherming van productieomgevingen is van cruciaal belang om onbevoegde toegang te voorkomen door niet alleen mensen buiten uw organisatie, maar ook door interne mensen – zelfs Operations-medewerkers. Als er iets misgaat, is het anders té verleidelijk om in te loggen in de betreffende omgeving en daar rond te neuzen om problemen op te lossen (een proces dat soms eufemistisch 'probleemoplossende heuristiek' wordt genoemd). Dit is om twee redenen bijna altijd een vreselijk idee. Ten eerste leidt het meestal tot serviceonderbrekingen (mensen zijn geneigd om het systeem opnieuw op te starten of willekeurig servicepacks toe te passen). Ten tweede, als er later iets misgaat, is er niet te achterhalen wie wat wanneer heeft gedaan. Dat betekent dat het onmogelijk is om de oorzaak van een willekeurig probleem te bepalen. In dat geval kunt u de omgeving net zo goed helemaal opnieuw maken, zodat de status weer bekend is." (*Literatuur: B, hoofdstuk 11*)
- B)** Onjuist. Dankzij automatisering kunt u sneller wijzigingen aanbrengen zonder vertrouwen in het proces te verliezen. Met toegangscontrole kunt u inderdaad klanten buiten houden, maar dat is niet het voornaamste doel ervan.
- C)** Onjuist. Operations is nooit overbodig. Het team kan zich hierdoor echter wel minder geërgerd en belast voelen. DevOps-projecten kunnen zowel in een cloudomgeving als daarbuiten bestaan. Hoewel toegangscontrole helpt de software in een cloudomgeving te beveiligen, is dat niet het voornaamste doel ervan.

18 / 50

Virtualisatie en cloud computing zijn technieken die DevOps-werkwijzen kunnen ondersteunen en faciliteren.

Op welke manier komen deze technieken DevOps ten goede?

- A) Cloud computing maakt toegang op afstand mogelijk, waardoor klanten meer controle krijgen.
 - B) Virtuele omgevingen kunnen makkelijker worden gestandaardiseerd en gebruiken hardware efficiënter.
 - C) Gevirtualiseerde infrastructuur is makkelijker te begrijpen en vereist geen onderhoud.
- A) Onjuist. Cloud computing maakt toegang op afstand zeker makkelijker, maar dat betekent op zich helemaal niet dat klanten meer controle krijgen. Beide aspecten komen DevOps op geen enkele wijze ten goede.
- B) Juist. Virtualisatie maakt het makkelijker om continuous integration (continue integratie) en de testinfrastructuur te consolideren, zodat het als service kan worden aangeboden aan leveringsteams. Ook wat betreft het hardwaregebruik is virtualisatie efficiënter. Met virtualisatie kunt u één enkele hardwareconfiguratie voor fysieke omgevingen standaardiseren, maar virtueel juist verscheidene heterogene omgevingen en platformen laten draaien. (*Literatuur: B, hoofdstuk 11*)
- C) Onjuist. Een gevirtualiseerde infrastructuur is op zich niet makkelijker te begrijpen. Bovendien moet deze net zo goed worden onderhouden en beheerd, zij het op een andere manier dan een on-premise infrastructuur. Dit is dus geen geldige reden.

19 / 50

SLA's (service level agreements) zijn voor elk project van belang, omdat u hierin de afspraken met de klant vastlegt. In DevOps hebben SLA's echter nog een belangrijk doel.

Wat is dit doel?

- A) Klanten zijn verantwoordelijk voor het opstellen van de SLA voor het DevOps-team. De SLA vervangt daarmee de formele taak als verantwoordelijkheid van Operations.
 - B) In de SLA kan de klant al diens niet-functionele vereisten opgeven, zodat Development zich daarop kan richten.
 - C) In de SLA wordt het acceptabele service level opgegeven. Development moet de SLA begrijpen en Operations ondersteunen bij de naleving ervan.
- A) Onjuist. De SLA is altijd een contract tussen de klant en het bedrijfsonderdeel dat de service levert. Daarom hebben beide partijen iets te zeggen over de inhoud ervan.
- B) Onjuist. Sommige niet-functionele en functionele vereisten kunnen weliswaar afkomstig zijn uit de SLA, maar de meeste worden rechtstreeks aan het DevOps-team gegeven, niet via de SLA. Bovendien is het bepalen van niet-functionele vereisten niet het doel van de SLA op zich.
- C) Juist. De voorwaarden die in de SLA worden beschreven, zijn het meest relevant voor Operations. Development moet Operations ondersteunen door hen het werk zo makkelijk mogelijk te maken. Dit is precies waarin DevOps afwijkt van reguliere ontwikkelactiviteiten. (*Literatuur: B, hoofdstuk 12*)

20 / 50

Voor een nieuw product moet uw team een deployment pipeline (uitrolpijplijn) ontwikkelen. Als onderdeel van continuous integration (continue integratie) moet u de commit-fase van de pipeline definiëren. U bespreekt deze fase met uw teamleden.

De Process Master zegt daarover: "De Definition-of-Done (definitie van klaar) moet worden opgesteld tijdens of vóór de commit-fase. Wanneer code vóór die tijd niet Done is, moet het werk worden stopgezet".

Is dat waar?

- A)** Ja. Als het werk niet Done is, heeft de Process Master zijn of haar werk niet goed uitgevoerd. Dit moet onmiddellijk worden opgelost.
 - B)** Ja. Werk dat niet Done is, moet niet worden doorgestuurd naar de commit-fase. Doorsturen draagt niet bij aan de waarde voor de klant.
 - C)** Nee. De Definition-of-Done wordt alleen opgesteld tijdens bijeenkomsten met de klant. Daarop wachten vertraagt het werk te veel.
 - D)** Nee. Werk in een deployment pipeline moet altijd doorgaan. Als code niet Done is, hoeft deze alleen maar inactief te zijn.
-
- A)** Onjuist. De Process Master heeft als taak ervoor te zorgen dat er een Definition-of-Done is en wanneer code die niet Done is, wordt doorgestuurd naar de commit-fase, moet het werk worden stopgezet. Dat betekent echter niet per se dat de Process Master slecht werk aflevert wanneer code naar de commit-fase gaat die niet Done is.
 - B)** Juist. Wanneer werk niet Done is, is er te weinig waarde voor de klant om het werk te starten in de deployment pipeline. Uitgaande van een one-piece-flow, zou dat de flow van waardevoller werk vertragen. (*Literatuur: B, hoofdstuk 3*)
 - C)** Onjuist. Definition-of-Done is een van de eerste dingen waarover overeenstemming wordt bereikt in een project. Het wordt niet gedefinieerd tijdens bijeenkomsten met de klant. Wanneer met codering wordt gestart, moet er al een Definition-of-Done bekend zijn. Hoe kun je anders weten wanneer je moet stoppen met coderen?
 - D)** Onjuist. Wanneer er iets mis is met de code, of wanneer de code geen waarde toevoegt, is dat voldoende reden om de deployment pipeline stop te zetten en de code te corrigeren, of om iets dat meer waarde heeft, op te nemen in de one-piece-flow pipeline.

21 / 50

Wat is het **grootste** voordeel van het vergroten van de diversiteit van het team, wat leidt tot een breder scala van persoonlijke achtergronden en culturen?

- A) Het brengt meer ervaring en meer standpunten in.
 - B) Het leidt tot minder spanningen binnen het team.
 - C) Het beperkt originaliteit en de mogelijkheid om nieuwe inzichten aan te dragen.
 - D) Het duurt langer om tot een specifieke beslissing te komen.
- A) Juist. Diversiteit betekent een breed scala van achtergronden wat betreft ras, geslacht, seksualiteit, maatschappelijke achtergrond, opleidingsniveau, taal en werkervaring. Al deze unieke aspecten zorgen ervoor dat er meer ervaring wordt ingebracht en er meer standpunten ter tafel komen. *(Literatuur: A, hoofdstuk 7)*
- B) Onjuist. De mogelijkheid bestaat dat vergroting van de diversiteit zelfs kan leiden tot meer druk en spanningen omdat mensen met verschillende culturele waarden samen moeten werken.
- C) Onjuist. Meer diversiteit betekent meer verschillende standpunten. Dat leidt doorgaans tot meer originaliteit.
- D) Onjuist. Dit wordt meestal als een nadeel ervaren. (Hoewel tragere beslissingsprocessen ook voordelen kunnen hebben.) Meer diversiteit kan ertoe leiden dat het langer duurt om overeenstemming te bereiken.

22 / 50

U werkt voor een klein bedrijf dat één DevOps-team heeft. Uw DevOps-team werkt aan een applicatie die uit meerdere componenten bestaat. Sommige componenten zijn nieuw en andere hebben alleen updates nodig.

Momenteel heeft elke component een eigen deployment pipeline (uitrolpijplijn). Het team is trots op de continuous delivery (continue levering) binnen de meeste pipelines. Bovendien is de productie hoog en van goede kwaliteit.

Wat moet u doen?

- A) De verschillende pipelines behouden en het team stimuleren om continuous delivery uit te breiden
 - B) Alleen pipelines met continuous delivery behouden en de overige pipelines zonder continuous delivery samenvoegen
 - C) Slechts twee verschillende pipelines behouden: één voor ontwikkeling en één voor onderhoud
 - D) Met het team toewerken naar één pipeline door uit te leggen wat de risico's zijn van meerdere pipelines
- A) Onjuist. Meer dan één deployment pipeline per team brengt risico's met zich mee. Het is lastig om prioriteiten toe te kennen aan de verschillende pipelines, en werk aan bepaalde zaken moet worden stilgelegd wanneer iets in een andere pipeline prioriteit krijgt. Dit brengt een zekere mate van chaos en multitasking met zich mee, die DevOps juist probeert te vermijden. De efficiëntie en waarde voor het bedrijf nemen toe wanneer het team met één enkele deployment pipeline werkt.
- B) Onjuist. Elke deployment pipeline heeft baat bij continuous delivery. Maar enkel en alleen het werken volgens werkwijzen voor continuous delivery, verkleint nog niet de risico's van meerdere pipelines.
- C) Onjuist. Zelfs wanneer u slechts twee deployment pipelines hebt én zelfs als deze functioneel gezien van elkaar gescheiden zijn, blijven dezelfde risico's van toepassing.
- D) Juist. Dit is de juiste aanpak. Hopelijk heeft het team genoeg geleerd van het werken met continuous delivery om deze werkwijze nu in de volledige pipeline te integreren. *(Literatuur: B, hoofdstuk 13)*

23 / 50

U beoordeelt Company Builders, een middelgrote tot grote organisatie die enkele jaren geleden is overgestapt op de DevOps-aanpak.

Het bedrijf heeft u ingehuurd om de huidige volwassenheid van DevOps vast te stellen. Op basis van uw bevindingen geeft u suggesties voor verbetering. Het bedrijf wil weten op welk gebied het zich moet richten om *Niveau 2 - Kwantitatief beheerd* te bereiken.

U hebt vastgesteld dat de meeste gebieden binnen *Niveau 1 - Consistent* vallen, met twee uitzonderingen:

1. **Omgevingen en deployment (uitrol)**. Dit gebied beheert speciaal ingerichte introducties en heeft release- en roll-backprocessen getest.
2. **Build management en continuous integration (continue integratie)**. In dit gebied treft u regulier geautomatiseerde builds en tests aan en elke build kan opnieuw worden gegenereerd op basis van bronbeheer met behulp van een geautomatiseerd proces.

Bepaal de mate van volwassenheid in deze twee gebieden op basis van de hier verstrekte informatie. Geef vervolgens uw aanbevelingen voor de aandachtspunten ter verbetering.

Aan welke van deze twee gebieden moet Company Builders werken om door te kunnen groeien naar Niveau 2?

- A) Zowel omgevingen en deployment als build management en continuous integration bevinden zich op niveau 0. Het werk moet op beide gebieden tegelijkertijd plaatsvinden.
 - B) Zowel omgevingen en deployment als build management en continuous integration hebben niveau 1 of hoger. Het werk moet in de overige gebieden worden uitgevoerd om door te groeien.
 - C) Omgevingen en deployment heeft niveau 0. Build management en continuous integration heeft niveau 1. Omgevingen en deployment moet als eerste worden onderzocht.
 - D) Omgevingen en deployment heeft niveau 2. Build management en continuous integration heeft niveau 0. Build management en continuous integration moet als enige worden onderzocht.
-
- A) Onjuist. Gebied 1 heeft niveau 2 en gebied 2 heeft niveau 0.
 - B) Onjuist. Gebied 1 heeft niveau 2 en gebied 2 heeft niveau 0.
 - C) Onjuist. Omdat omgevingen en deployment al niveau 2 heeft, hoeft hier niet aan gewerkt te worden.
 - D) Juist. Gebied 2 heeft niveau 0 en moet eerst doorgroeien naar niveau 1 zodat de organisatie zich kan richten op niveau 2. (*Literatuur: B, hoofdstuk 15*)

24 / 50

Uw bedrijf verkoopt een online-back-upservice aan haar klanten. Een van die klanten heeft u nu gevraagd nieuwe functies te implementeren in uw service. De klant wil die nieuwe functies binnen een week, anders gaat hij op zoek naar een andere leverancier.

U vindt de nieuwe functies belangrijk en weet dat het Development-team ze snel kan bouwen. Echter, u loopt tegen de volgende problemen aan:

- Het duurt lang voordat testers bugs sluiten.
- Testers vinden bugs die ontwikkelaars al lang geleden hebben opgelost.
- De applicatie kan zelden in goed werkende staat worden gedemonstreerd.
- Er worden zelden showcases gegeven.

Wat is uw probleem en hoe lost u dit op?

- A)** U deployt code die bugs bevat. Dit moet u oplossen door de kennis van het deploymentproces (uitrolproces) te vergroten, de samenwerking te verbeteren en gedisciplineerder te werken.
- B)** U hebt gebrekkig configuration management. Dit moet u oplossen door de samenwerking tussen Development en Operations te vergroten, monitoring en logboekregistratie te verbeteren en meer gebruik te maken van virtualisatie.
- C)** Uw proces voor continuous integration (continue integratie) wordt niet goed beheerd. Dit moet u oplossen door het tempo van geautomatiseerde tests en de commit-fase te verkorten, en de kennis van het proces voor continuous integration te vergroten.
- D)** Uw teststrategie is niet effectief. Dit moet u oplossen door tests te automatiseren en de samenwerking tussen de testers en de rest van het team te verbeteren.
-
- A)** Juist. Het deployen van een code met bugs of het niet vaak genoeg deployen, leidt tot lange deploymenttijden, een lage velocity, scepsis over releasedatums, verlies van vertrouwen in de omgeving voor continuous integration (continue integratie), extra tijd om bugs op te lossen, het vinden van bugs die ontwikkelaars al lang geleden hebben opgelost, en een gering aantal demonstraties en showcases. Deze problemen worden verholpen met de gekozen oplossing. (*Literatuur: B, hoofdstuk 15*)
- B)** Onjuist. Slecht configuration management leidt met name tot onverklaarbare productiestoringen, onbeheersbare opleveringen, meer tijd voor de omgevingsconfiguratie en een lange hersteltijd na storingen. Deze problemen worden verholpen met de gekozen oplossing.
- C)** Onjuist. Wanneer het proces voor continuous integration niet goed wordt beheerd, leidt dit met name tot minder dan dagelijkse opleveringen, een onderbroken commit-fase en een lange integratiefase tussen releases. Deze problemen worden verholpen met de gekozen oplossing.
- D)** Onjuist. Een ineffektieve teststrategie leidt tot terugkerende bugs, veel verloren tijd besteed aan het oplossen van bugs, veel klachten van klanten, een product van lage kwaliteit en ontwikkelaars die onder spanning staan. Deze problemen worden verholpen met de gekozen oplossing.

25 / 50

Wat is een voordeel van continuous integration (continue integratie) binnen effectief DevOps?

- A) Uitgebreide testcyclussen voorafgaand aan de releases de functies
 - B) Lange perioden tussen de releases van de functies
 - C) Regelmatige en tijdige releases van de functies
- A) Onjuist. Er moet zeer goed worden getest, maar dit hoeven geen uitgebreide cyclussen te zijn. Bovendien is dit geen voordeel.
- B) Onjuist. Dit is het tegenovergestelde van wat er gebeurt.
- C) Juist. Continuous integration maakt snellere en regelmatige releases mogelijk, omdat de integratie wordt geautomatiseerd. (Literatuur: B, hoofdstuk 3)

26 / 50

Een multinational ervaart de nodige moeilijkheden om commits uit hun gedistribueerde locaties samen te voegen in een centrale repository die zich bevindt in Dallas, VS. De gedistribueerde locaties zijn Mexico-Stad, Parijs, San Diego en Engeland. Er is geen consistentie in de manier waarop de regio's hun commits uitvoeren en soms is het niet duidelijk of bepaalde tests wel of niet zijn gelukt.

Er bestaan vier mogelijke werkwijzen:

1. Check niet in bij een vastgelopen build.
2. Voer alle commit-tests altijd eerst lokaal uit of laat uw continuous integration-server dat voor u doen.
3. Wacht totdat de commit-tests zijn geslaagd voordat u verdergaat.
4. Doe geen "comment-out" bij mislukte tests.

Welke van deze werkwijzen zorgen er het **beste** voor dat de gedistribueerde locaties de huidige pijnpunten zullen aanpakken?

- A) 1 en 2
 - B) 1, 2 en 3
 - C) 2 en 3
 - D) 2, 3 en 4
- A) Onjuist. 1 is hier niet van toepassing. 3 en 4 zijn ook belangrijk.
- B) Onjuist. 1 is niet belangrijk.
- C) Onjuist. 4 is ook belangrijk.
- D) Juist. Alle drie deze werkwijzen zijn volledig van toepassing op het huidige scenario. Omdat er geen bewijs is dat er wordt ingecheckt bij vastgelopen builds, is deze praktijk minder van toepassing. (Literatuur: B, hoofdstuk 3)

27 / 50

Neem de anatomie van een elementaire deployment pipeline (uitrolpijlijn) in aanmerking.

Welke fase bevestigt dat het systeem op zowel het functionele als niet-functionele niveau werkt?

- A) Geautomatiseerde acceptatietest
- B) Build- en unittest
- C) Handmatige acceptatietest
- D) Versiebeheer

- A) Juist. De fase van de geautomatiseerde acceptatietest bevestigt dat het systeem op het functionele en niet-functionele niveau werkt, dat het qua gedrag tegemoetkomt aan de behoeften van de gebruikers en voldoet aan de specificaties van de klant. (*Literatuur: B, hoofdstuk 8*)
- B) Onjuist. Met buildtests en unittests wordt gecontroleerd of het nieuwe stuk code intern in orde is. Hiermee wordt niet de integratie met de bestaande build gecontroleerd.
- C) Onjuist. Dit zou een juist antwoord kunnen zijn. Maar in een functionerende deployment pipeline moeten de acceptatietests geautomatiseerd zijn.
- D) Onjuist. Versiebeheer wordt gebruikt om vastgelopen builds of problemen en fouten te verhelpen. Dit wordt niet gebruikt om aan te tonen dat het systeem op een functioneel of niet-functioneel niveau werkt.

28 / 50

Het is een best practice in DevOps om hetzelfde deploymentproces (uitrolproces) te gebruiken voor elke omgeving waarin uw toepassing wordt uitgevoerd. Dit zorgt ervoor dat de build effectief wordt getest. U gebruikt scripts om uw build- en deploymentproces te automatiseren.

Wat is de **beste** manier om dat te doen?

- A) Gebruik één script voor elke omgeving en onderhoud de scripts als onderdeel van het versiebeheersysteem.
 - B) Gebruik voor elke omgeving één specifiek script om recht te doen aan de verschillen tussen de omgevingen.
 - C) Gebruik dezelfde scripts voor elke omgeving en breng handmatig parameters aan voor specifieke configuraties.
 - D) Gebruik dezelfde scripts voor elke omgeving en beheer de configuratiegegevens afzonderlijk.
- A) Onjuist. Dit is een slechte keuze vanwege de hoeveelheid werk om dit te onderhouden en de potentiële fouten die erin kunnen sluipen door het complexe karakter.
 - B) Onjuist. Verschillende scripts kunnen leiden tot problemen met wijzigingen, waardoor weer problemen binnen het proces ontstaan die niet eenvoudig op te sporen en te verhelpen zijn.
 - C) Onjuist. Er moet niet handmatig worden ingegrepen tussen het bouwen en de uitrol: dit proces moet geautomatiseerd worden, om redenen van snelheid en om het proces foutloos te maken.
 - D) Juist. Het script moet hetzelfde zijn om ervoor te zorgen dat zowel het build- als leveringsproces effectief kan worden getest. De verschillen tussen omgevingen, zoals URI, IP, etc., moeten worden beheerd als onderdeel van configuration management. (*Literatuur: B, hoofdstuk 6*)

29 / 50

Na de release van een nieuwe IT-service stopt een actieve taak onverwachts.

Wat is **geen** mogelijke oorzaak hiervan?

- A) De Gatekeeper heeft geen rekening gehouden met de samenhang tussen werkitens en de service acceptance criteria (SAC).
- B) De niet-functionele vereisten waren niet duidelijk wegens een onvolledige user story.
- C) De Service Master heeft met de gebruiker geen uitfasering afgesproken voor de service in de release.
- D) Er is te weinig werk verricht tijdens de ontwikkelprocessen, wat ook blijkt uit de SAC.

- A) Onjuist. Dit is wel een mogelijke oorzaak.
- B) Onjuist. Dit is wel een mogelijke oorzaak.
- C) Juist. Dit is geen mogelijke oorzaak, aangezien dit niet vooraf kan worden bepaald, maar juist altijd moet worden bepaald wanneer de service eenmaal live is. (*Literatuur: C, hoofdstuk 7*)
- D) Onjuist. Dit is wel een mogelijke oorzaak.

30 / 50

Bedrijf AppBC maakt gebruik van DevOps. Het bedrijf heeft continuous deployment (continue uitrol) en een goede deployment pipeline geïmplementeerd, met sterk geautomatiseerde acceptatietests. Dagelijks wordt nieuwe software aan productie geleverd.

AppBC heeft een grote database en veel gebruikers. Er bestaat een uitgebreide en solide strategie voor capaciteitstesten. Omdat de omgeving groot en complex is, komen er bij elke nieuwe versie wel enkele bugs in productie terecht.

Wat is voor AppBC de **beste** strategie om deze bugs te voorkomen?

- A) Kanarie releases instellen
 - B) Capaciteitstesten automatiseren
 - C) De leveringsfrequentie verlagen
 - D) Blue-green-deployment gebruiken
-
- A) Juist. Kanarie releases omvatten de roll-out van een nieuwe versie van een toepassing naar een subset van de productieservers om snel feedback te krijgen. Daarmee worden snel eventuele problemen met de nieuwe versie ontdekt zonder dat dit invloed heeft op het merendeel van de gebruikers door de werkbelasting geleidelijk te vergroten. Tegelijkertijd worden reactietijden en andere prestatiecijfers gemeten waardoor de risico's bij het vrijgeven van een nieuwe versie worden verminderd en de bugs sneller kunnen worden ontdekt en verholpen. (*Literatuur: B, hoofdstuk 10*)
 - B) Onjuist. Binnen deze context moeten capaciteitstests al zijn geautomatiseerd, maar automatisering van deze tests draagt niet bij aan het detecteren van bugs in dit scenario.
 - C) Onjuist. Dit is in strijd met de DevOps-werkwijze.
 - D) Onjuist. Voor blue-green zijn te veel resources nodig, wat te duur is voor dit scenario. Bovendien kan toepassing van deze strategie met een grote database leiden tot downtime of read-only situaties als rollback nodig is. Evenmin leidt dit tot betere capaciteitstests.

31 / 50

DevOps maakt gebruik van zeer belangrijke concepten uit Agile, die weer zijn afgeleid van het Toyota-productiesysteem.

Waarom is one-piece-flow belangrijk voor het gebruiken van DevOps?

- A) Omdat een team hiermee in een beheersbaar tempo met een voorspelbare velocity kan werken.
 - B) Omdat een team zich hiermee kan richten op het toevoegen van functies met een zo groot mogelijke waarde voor het bedrijf.
 - C) Omdat verschillende teamleden hiermee in hogere mate de verantwoordelijkheid voor werk delen.
 - D) Omdat hiermee minder knelpunten ontstaan, doordat het aantal gelijktijdige taken wordt beperkt.
-
- A) Onjuist. Hier zorgt ritme voor. Door een beheersbaar ritme te bepalen zorgt u dat het tempo voorspelbaar blijft, dat het team niet overwerkt raakt en dat medewerkers een goede balans tussen werk en privé hebben.
 - B) Juist. Dit is wat met one-piece-flow wordt bereikt. One-piece-flow stelt u in staat de functie die de meeste waarde oplevert te kiezen of te updaten, waarna deze als eerstvolgende in de pipeline wordt gezet. Zo blijft u Agile. Door aan slechts één functie te werken, kunt u doorgaans Werk in Uitvoering (WiP) beperken, waardoor de functie ook echt af komt. (*Literatuur: C, hoofdstuk 7*)
 - C) Onjuist. Dit is wel belangrijk voor DevOps, maar houdt niet direct verband met Werk in Uitvoering, one-piece-flow, ritme of Ji-Kotei-Kanketsu (JKK).
 - D) Onjuist. Dit is waar Werk in Uitvoering om draait. Door het aantal taken te beperken waaraan u gelijktijdig kunt werken met uw team, slaagt u erin toegewezen taken tot een goed einde te brengen. U vermijdt knelpunten, aangezien u de WiP-limiet zodanig reguleert dat er altijd iemand wacht op het voltooide werk.

32 / 50

De S Corporation, een middelgrote leverancier van auto-onderdelen, is toeleverancier van een grote onderneming, T Motors Corporation. Het bedrijf levert auto-onderdelen aan T Motors Corporation, wat optelt tot bijna 60% van de totale verkoop.

Het bestuur vergadert over het nieuwe partnerschap. T Motors Corporation eist van S Corporation dat de wijze van levering wordt omgezet in Just-in-Time levering, anders stoppen ze hun samenwerking met S Corporation. Omdat S Corporation het verlies aan omzet niet zou overleven, maakt dit de omschakeling naar Just-in-Time tot een urgente kwestie. Omdat deze wijziging binnen 6 maanden moet plaatsvinden, zijn er hooguit 5 maanden voor de voorbereiding.

Een van de zaken die moeten worden geïmplementeerd is het volgen van onderdelen via radio-frequency identification (RFID). Dit moet bijdragen aan een transparant productieproces. Er moet een snelle controle worden uitgevoerd van het huidige proces om de overstap naar een RFID-proces te vereenvoudigen.

De CIO wordt gevraagd het veranderingsproces te leiden. Zij gelooft dat dit mogelijk is als een DevOps-benadering wordt gebruikt om een minimale release te realiseren. Idealiter moet het concept van productie met RFID als eerste worden ontwikkeld. Als laatste moet een productiecontrolesysteem met gebruikmaking van de RFID-gegevens worden geïmplementeerd. Er is echter te weinig tijd om die stappen opeenvolgend uit te voeren. Daarom moeten deze drie dingen gelijktijdig worden uitgevoerd.

De CIO wijst het project toe aan Em, een Scrum Master. Development werkt aan de voorbereiding voor een deployment pipeline (uitrolpijlijn).

Em ziet dat Development enthousiast is en hard werkt, maar het team zou meer discipline kunnen gebruiken. Bovendien moet de releasefrequentie omhoog.

Waar moet Em zich als **eerste** op richten?

- A) Em moet zich vooral richten op communicatie, omdat dit het belangrijkste aspect in DevOps is. Em moet eerst het ijs breken en enkele communicatieregels voor het team opstellen.
- B) Em moet met team eerst het gebruik van value stream mapping (een waardestromingschema) bespreken en one-piece-flow invoeren, omdat flow en een gestroomlijnd proces bijzonder belangrijk zijn.
- C) Em moet met de teamleden eerst de infrastructuur en de werkomgeving bespreken, omdat DevOps met name effectief is wanneer alle tools en werkwijzen werken.
- D) Em moet eerst alle stakeholders bij elkaar roepen, ze opleiden in DevOps en hen vragen of ze mee willen helpen de cultuurverandering te verspreiden omdat een dergelijke verandering nodig is voor DevOps.

De vraag gaat verder op de volgende pagina.

- A) Onjuist. Hoewel communicatie belangrijk is voor een goed functionerend DevOps-team, hebben de leden al eerder met elkaar samengewerkt. Er zijn andere zaken die meer resultaat zullen opleveren op korte termijn. Het is belangrijker dat een switch wordt gemaakt naar denken in termen van waarde voor de klant. Zodra die switch is gemaakt, kan Em verder werken aan het afstemmen van de juiste communicatievaardigheden.
- B) Juist. De processen moeten worden gestroomlijnd en value stream mapping moet worden gemaakt, zodat het team zoveel mogelijk waarde met zo weinig mogelijk inspanning kan toevoegen. Vervolgens is het tijd om te definiëren wat er kan en moet veranderen en wat al prima werkt: tools, communicatie en cultuur inbegrepen. DevOps hoeft niet in elk bedrijf hetzelfde te zijn, maar moet er wel voor zorgen dat het team zich focust op het toevoegen van waarde voor de klant. (*Literatuur: B, hoofdstuk 1 en A, hoofdstukken 1 en 2*)
- C) Onjuist. Tools en automatisering zijn belangrijke onderdelen van DevOps en mogen niet worden vergeten. Maar dit zijn niet per se de aspecten waarnaar het eerst moet worden gekeken. Het is beter om eerst na te gaan of het gehele productie- of serviceproces gemakkelijker, korter en goedkoper kan worden door een value stream map te maken en deze met het team te bespreken. Zo wordt een begin gemaakt met de onvermijdelijke en noodzakelijke cyclus van continue verbetering. Alle andere stappen volgen daarna.
- D) Onjuist. In veel bedrijven moet een cultuurverandering plaatsvinden. Maar dat is niet per definitie het eerste wat moet gebeuren. Het is ook niet nodig om de stakeholders bij elkaar te roepen om die cultuurverandering te bespreken. De cultuurverandering moet voortkomen uit de wijze waarop de meeste waarde wordt gecreëerd voor klanten of stakeholders. Het is een goed idee om verschillende stakeholders te vragen naar de value stream map te kijken om te zien wat ze daaraan zouden kunnen toevoegen.

33 / 50

Uw DevOps-team werkt goed samen in een beheersbaar tempo. Door voldoende speling in te bouwen in het proces, heeft het team de benodigde tijd en concentratie om builds zorgvuldig te controleren en testen. Het testen en deployen (uitrollen) vindt momenteel handmatig plaats. Het tempo ligt hoog genoeg om regelmatig hoge waarde te leveren aan het bedrijf.

De CEO vraagt u nu om advies over automatisering binnen dit team.

Welk advies moet u geven?

- A) Automatiseer zo veel mogelijk, zodat het team meer functies kan toevoegen en sneller waarde voor het bedrijf kan laten zien
 - B) Automatiseer de acceptatietests maar niet het deployen, aangezien het handmatige proces daarvoor veiliger is
 - C) Automatiseer het deployen om de cyclustijd te verbeteren, maar niet de tests, zodat het team kan leren van bugs
 - D) Automatiseer niets aan deze methoden van dit team, aangezien de huidige methoden fantastische resultaten opleveren
-
- A) Juist. Automatiseer altijd alles wat kan worden geautomatiseerd, zodat u meer controle over het proces krijgt en u eerder meer waarde voor het bedrijf kunt laten zien. (*Literatuur: B, hoofdstukken 1 en 8*)
 - B) Onjuist. Het is een goed idee om acceptatietests te automatiseren, maar handmatig deployen is zeker niet veiliger.
 - C) Onjuist. Het is een goed idee om releases te automatiseren, maar het team leert bij handmatige tests niet meer van bugs dan bij geautomatiseerde tests.
 - D) Onjuist. Het team verricht uitstekend werk, maar u verspilt wel een deel van hun potentieel door het niet automatiseren van activiteiten die kunnen worden geautomatiseerd.

34 / 50

Een CIO wijst een project toe aan haar meest betrouwbare medewerker, Michael, een Scrum Master. Het Development-team bereidt het bouwen van een deployment pipeline (uitrolpijplijn) voor.

Michael heeft alle vertrouwen in de goede voornemens en spontaniteit van het Development-team, maar zou het team wel graag gedisciplineerder zien. Bovendien moet de releasefrequentie omhoog. Michael wil dat het Development-team vaker kan releasen.

Een van de teamleden zegt daarop: "Het belangrijkste voor de nieuwe deployment pipeline is dat deze moet worden geautomatiseerd. Dat moet dus als eerste gebeuren".

Is deze bewering juist?

- A) Ja, dat is juist. Het automatiseren van de deployment pipeline is de belangrijkste factor bij het verhogen van de efficiëntie.
 - B) Ja, dat is juist. Door een deployment pipeline te implementeren die geautomatiseerd is, worden potentiële problemen voorkomen.
 - C) Nee, dat is niet juist. Het realiseren van een one-piece-flow met een solide deploymentproces (uitrolproces) dient de voornaamste prioriteit te zijn. Automatisering van het proces kan in een later stadium plaatsvinden.
 - D) Nee, dat is niet juist. Niet de deployment pipeline, maar het testproces moet als eerste worden geautomatiseerd.
-
- A) Onjuist. De deployment pipeline moet altijd in eerste instantie een one-piece-flow deployment pipeline zijn. Dat kan prima werken zonder automatisering. Zodra het proces stevig is geïmplementeerd, kan het proces waar mogelijk alsnog worden geautomatiseerd. Dat moet echter altijd ondergeschikt zijn aan het bouwen van een solide deployment pipeline.
 - B) Onjuist. De deployment pipeline moet altijd in eerste instantie een one-piece-flow deployment pipeline zijn. Dat kan prima werken zonder automatisering. Zodra het proces stevig is geïmplementeerd, kan het proces waar mogelijk alsnog worden geautomatiseerd. Dat moet echter altijd ondergeschikt zijn aan het bouwen van een solide deployment pipeline.
 - C) Juist. De deployment pipeline moet altijd in eerste instantie een one-piece-flow deployment pipeline zijn. Dat kan prima werken zonder automatisering. Zodra het proces stevig is geïmplementeerd, kan het proces waar mogelijk alsnog worden geautomatiseerd. Dat moet echter altijd ondergeschikt zijn aan het bouwen van een solide deployment pipeline. (*Literatuur: B, hoofdstuk 5*)
 - D) Onjuist. Het automatiseren van tests is een kernactiviteit. Maar als u zich gesteld ziet voor de keuze om een solide deployment pipeline te maken of tests te automatiseren, moet u altijd eerst kiezen voor het maken van een solide deployment pipeline. Daarna is er gelegenheid om de efficiency te verhogen door middel van testautomatisering.

35 / 50

Uw bedrijf verandert van werkwijze en gaat DevOps inzetten. Uw team doet mee met deze verandering. U bespreekt best practices voor de fase waarin code wordt doorgevoerd.

Uw collega Sun zegt daarover: "Wanneer een build vastloopt en niemand daarvoor de verantwoordelijkheid neemt, moeten we uitzoeken om welke persoon het gaat en hem of haar daarop aanspreken, zodat de build kan worden hersteld."

Is dat een goed idee?

- A)** Ja. Alleen degene die het vastlopen van een build veroorzaakt, kan dat herstellen. Zoek daarom uit om wie het gaat, ook al zal die persoon zich daarbij ongemakkelijk voelen.
 - B)** Ja. Je moet altijd de verantwoordelijkheid nemen voor het laten vastlopen van een build. Doe je dat niet, dan hebben je collega's het recht je daarop aan te spreken.
 - C)** Nee. DevOps is gebaseerd op een verwijtloze cultuur. Als een collega geen verantwoordelijkheid neemt, moet hij of zij daar ook niet toe gedwongen worden.
 - D)** Nee. U moet de build eerst herstellen. Zoek dan pas uit wie de verantwoordelijke is en leg de juiste sanctie op.
-
- A)** Onjuist. Het is waarschijnlijk het gemakkelijkst om de persoon die het probleem heeft veroorzaakt ook het probleem op te laten lossen, maar noodzakelijk is dat niet. DevOps is gebaseerd op een niet-beschuldigende cultuur. Als een collega geen verantwoordelijkheid neemt, moet hij of zij daar niet toe gedwongen worden. Iemand dwingen om iets te doen is niet respectvol.
 - B)** Onjuist. DevOps is gebaseerd op een niet-beschuldigende cultuur. Als een collega geen verantwoordelijkheid neemt, moet hij of zij daar niet toe gedwongen worden. Iemand dwingen om iets te doen is niet respectvol.
 - C)** Juist. Iemand dwingen om iets te doen is niet respectvol. Het is OK om fouten te maken. De teamleden werken samen aan het oplossen van problemen of uitdagingen. (*Literatuur: B, hoofdstuk 3 en A, hoofdstuk 4*)
 - D)** Onjuist. De build hoeft niet hersteld te worden. Er kan worden teruggevallen op een vorige versie. Bovendien is het herstellen van de build misschien geen slecht idee, maar iemand een sanctie opleggen voor een fout is dat wel. DevOps is gebaseerd op een niet-beschuldigende cultuur. Als een collega geen verantwoordelijkheid neemt, moet hij of zij daar niet toe gedwongen worden. Iemand dwingen om iets te doen is niet respectvol.

36 / 50

Het Development-team van X-AppGo heeft te maken met talloze uitdagingen bij hun huidige testaanpak. Momenteel hanteert het bedrijf een handmatig proces bij acceptatietests. De ontwikkelaars zijn van mening dat het door hen samengestelde pakket unittests degelijk genoeg is om bescherming te bieden tegen regressies.

Het Development-team moet bij elke release 1 miljoen euro besteden aan handmatige acceptatietests. De bedrijfsleiding heeft besloten dat het Development-team geautomatiseerde acceptatietests moet implementeren om de algehele kosten voor testen te verlagen en tegelijkertijd het aantal defecten en regressies dat in de productieomgeving terechtkomt tot een minimum te beperken.

Welke principes moeten worden gehanteerd bij het definiëren van acceptatiecriteria voor uw applicatie met automatisering als uitgangspunt?

- A) Agile-principes
 - B) ATAM-principes
 - C) INVEST-principes
- A) Onjuist. INVEST is de aanbevolen groep principes bij het bouwen van onderhoudsvriendelijke acceptatietestpakketten. Er is geen specifieke informatie over de redenen waarom ATAM en Agile niet worden aanbevolen. Agile biedt geen specifieke informatie of principes voor het automatiseren van tests.
- B) Onjuist. INVEST is de aanbevolen groep principes bij het bouwen van onderhoudsvriendelijke acceptatietestpakketten. Er is geen specifieke informatie over de redenen waarom ATAM en Agile niet worden aanbevolen.
- C) Juist. Omdat acceptatietests zijn afgeleid van acceptatiecriteria, moeten de acceptatiecriteria voor uw applicatie geschreven worden met automatisering als uitgangspunt en moeten ze de INVEST-principes volgen. INVEST staat voor Independent, Negotiable, Valuable, Estimable, Small en Testable. (*Literatuur: B, hoofdstuk 8*)

37 / 50

Wat is het **meest** effectieve mechanisme om gegevens op geautomatiseerde wijze te migreren?

- A) Het creëren van een schema voor databaseversies en dit bewaren onder versiebeheer
 - B) Het creëren en beheren van kleinere datasets, zodat de migratie gemakkelijker wordt
 - C) Het zekerstellen dat alle scripts goed zijn getest alvorens de gegevens te migreren
 - D) Het zekerstellen dat u beschikt over een roll-backprocedure voor het geval dat de migratie mislukt
- A) Juist. Versiebeheer van uw database is het beste mechanisme om gegevens op geautomatiseerde wijze te migreren. (*Literatuur: B, hoofdstuk 12*)
- B) Onjuist. Dit is niet het beste mechanisme om geautomatiseerde migraties te ondersteunen. Dit mechanisme is hoofdzakelijk gericht op effectief beheer van datasets.
- C) Onjuist. Dit is niet het beste antwoord en draait meer om testactiviteiten in plaats van migratieactiviteiten.
- D) Onjuist. Dit draait om de benodigde herstelacties als de migratie mislukt.

38 / 50

Bedrijf X-AppGo ondervindt problemen met de roll-backprocessen. Dit resulteert vaak in verlies van kritieke gegevens in de databases van de productieapplicatie.

Wanneer is het **niet** mogelijk om roll-backscripts uit te voeren zonder dat kritieke gegevens verloren gaan?

- A) Wanneer met het roll-backscript gegevens worden verwijderd die alleen door de nieuwe versie worden gebruikt.
 - B) Wanneer met het roll-backscript een kolom wordt verplaatst tussen tabellen.
 - C) Wanneer met het roll-backscript gegevens uit tijdelijke tabellen weer worden toegevoegd.
-
- A) Onjuist. Met het roll-backscript worden slechts bepaalde gegevens verwijderd die alleen door de nieuwe versie worden gebruikt. Dit leidt niet tot verlies van kritieke gegevens bij een roll-back.
 - B) Onjuist. Het roll-backscript wijzigt het databaseschema op een manier die niet leidt tot gegevensverlies.
 - C) Juist. Dit is een scenario waarin een roll-backscript niet mogelijk is. (*Literatuur: B, hoofdstuk 12*)

39 / 50

Na het deployen (uitrollen) van software-upgrades voor een applicatie en het vervangen van routers en switches, heeft bedrijf ACMECONST veel last van applicatie- en hardwarestoringen.

Toen deze storingen zich voordeden tijdens de onderhoudsperiode, bleek het erg lastig om herstel naar de oorspronkelijke staat uit te voeren. Dit leidde tot langere hersteltijden, die ver buiten de normale onderhoudsperiode vielen en tot meer downtime voor kritieke applicaties.

Geautomatiseerde voorzieningen en een autonome infrastructuur kunnen in deze situatie van pas komen, maar hierbij gelden enkele kanttekeningen.

Welke aspecten moeten zorgvuldig worden beheerd om het risico op onderbrekingen te beperken bij het deployen naar de productieomgeving?

- A) Gedetailleerde monitoring-logs om fouten tijdens applicatie-upgrades op te lossen
 - B) Externe integratiepunten, zoals externe systemen en services
 - C) Serverconfiguraties en onderliggende informatie over gebruikersaccounts
 - D) De toolset voor automatisering en een autonome architectuur
-
- A) Onjuist. Problemen worden opgelost na de upgrade, waardoor dit geen aspect is om bij deployen naar de productieomgeving de risico's te beperken.
 - B) Juist. Dit aspect moet zorgvuldig worden beheerd om de deployment naar een productie-achtige omgeving de risico's te beperken. (*Literatuur: B, hoofdstuk 11*)
 - C) Onjuist. Dit valt voor zowel test- als productieomgevingen onder de informatie over het besturingssysteem en de bijbehorende configuratie. Het is dus geen aspect dat u apart beheert.
 - D) Onjuist. Om risico's te beperken is het niet belangrijk dat u de tools beheert. De tools bieden het juiste proces en context voor geautomatiseerde voorzieningen en een autonome infrastructuur.

40 / 50

Bedrijf X-AppGo ondervindt problemen met zijn kernapplicatie. De applicatie communiceert niet goed met andere externe applicaties. Deze externe applicaties moeten op effectieve wijze specifieke gegevensvariabelen verkrijgen, zodat specifieke aanroepen kunnen worden uitgevoerd. De kernapplicatie wordt ontwikkeld door één team en het bedrijf wil dat om zeer goede zakelijke redenen zo houden.

Een van de ontwikkelaars stelt voor een component uit de X-AppGo-codebasis te isoleren om zo de communicatieproblemen aan te pakken.

Wat is in deze situatie een goede reden om een component te isoleren?

- A) Om een set plug-ins in de codebase te converteren naar een monolithische codebasis
 - B) Om de impact van wijzigingen te beperken en het makkelijker te maken om de codebasis te wijzigen
 - C) Om de codebasis van X-AppGo op te splitsen en door verschillende teams te laten beheren
 - D) Er is geen goede reden hiervoor en hierdoor gaat het compileren langer duren
-
- A) Onjuist. Wanneer u een component maakt, gaat u van een monolithische codebasis naar een modulaire gebaseerd op componenten. Dit is dus juist het tegenovergestelde.
 - B) Juist. Zo worden we gestimuleerd om software met duidelijk gescheiden verantwoordelijkheden te ontwerpen en onderhouden. Hierdoor hebben wijzigingen minder impact en is de codebasis beter te begrijpen en makkelijker te wijzigen. (*Literatuur: B, hoofdstuk 13*)
 - C) Onjuist. Aangezien de applicatie van X-AppGo hiermee niet wordt opgedeeld in onafhankelijke componenten, is het niet nodig om de zakelijke redenen in twijfel te trekken en het team op te splitsen.
 - D) Onjuist. Er zijn meerdere goede redenen, en optie 2 is één van die redenen. Door een modulaire codebasis te maken is er bovendien juist minder tijd nodig om de code te compileren en koppelen, niet meer.

41 / 50

Zelfs de kleinste applicaties zijn afhankelijk van andere componenten of bibliotheken. Daarom is het begrijpen en beheren van afhankelijkheden een belangrijke activiteit binnen continuous deployment (continue uitrol) om de flow in de deployment pipeline (uitrolpijplijn) te houden.

U hebt een applicatie gemaakt die gebruikmaakt van twee bibliotheken. Elk van deze bibliotheken is afhankelijk van een derde, onderliggende bibliotheek. Ze verwijzen echter allebei naar een andere versie van die derde bibliotheek. Hierdoor ontstaat een specifieke afhankelijkheid.

Wat is de **beste** oplossing om deze afhankelijkheid te verhelpen of voorkomen?

- A) Alle bibliotheken samenvoegen tot één enkele bibliotheek, zodat u rechtstreeks naar die bibliotheek kunt verwijzen en zo het probleem kunt voorkomen
 - B) De bibliotheken beheren met behulp van versiebeheer, zodat u het direct ziet als u dit type afhankelijkheid maakt
 - C) Met post-its op een groot bord een visueel overzicht van alle afhankelijkheden bijhouden, zodat u de flow kunt volgen
 - D) Slechts kleine delen van de toolketen controleren, zodat u problemen die zich mogelijk voordoen bij het inchecken, makkelijk kunt debuggen
-
- A) Onjuist. Dit is geen goed idee. Gebruik in plaats hiervan, versiebeheer om te verwijzen naar de meest recente bibliotheek of gebruik een automatiseringstool die u helpt automatisch de nieuwste versie van bibliotheken te gebruiken.
 - B) Juist. Dit is een goede oplossing. Bovendien kunt u met versiebeheer voor uw bibliotheken problemen debuggen die ontstaan wanneer gebruikers oude versies van uw software gebruiken. Een andere goede oplossing is het gebruik van een automatiseringstool die zorgt dat de nieuwste versie van bibliotheken wordt gebruikt. (*Literatuur: B, hoofdstuk 13*)
 - C) Onjuist. Hiermee kan wel de flow worden bijgehouden, maar dit is geen oplossing voor afhankelijkheden. Kanban-borden helpen niet echt bij de introductie van versiebeheer.
 - D) Onjuist. In plaats hiervan moet u de volledige toolketen inchecken. Zo spoort u veel sneller en betrouwbaarder onderlinge afhankelijkheden en compatibiliteitsproblemen op.

42 / 50

In een omgeving met continuous deployment (continue uitrol) is het belangrijk dat alle versies worden beheerd, zodat u snel fouten kunt vinden of indien nodig een roll-back kunt uitvoeren.

Het wordt echter **niet** aanbevolen om binaire output te bewaren in versiebeheer.

Waarom wordt deze uitzondering gemaakt?

- A) Binaire output staat doorgaans in grote bestanden die voor elke build veranderen en automatisch worden bijgewerkt.
 - B) Er werken meerdere teamleden aan de binaire bestanden, dus het is niet praktisch om deze te bewaren binnen versiebeheer.
 - C) De binaire output is de input voor compilers, die al worden bewaard in versiebeheer.
 - D) Dit is niet nodig, aangezien hercompilatie deel uitmaakt van het normale buildproces.
-
- A) Juist. Ten eerste is de output zeer groot en wordt deze opnieuw gemaakt voor elke gecompileerde check-in die de geautomatiseerde tests doorstaat. Ten tweede kan de binaire output opnieuw worden gemaakt op basis van de broncode, door het buildscript opnieuw uit te voeren. (*Literatuur: B, hoofdstuk 2*)
 - B) Onjuist. Dit is niet de reden.
 - C) Onjuist. Binaire output is de output van compilers, niet de input. De rest van de redenering is wel juist.
 - D) Onjuist. Hercompilatie wordt niet aangeraden als regulier onderdeel het normale buildproces. Het is wel juist dat hercompilatie nieuwe binaire output oplevert.

43 / 50

U wilt een holistische aanpak gebruiken voor het beheren van uw volledige IT-infrastructuur.

Op welke twee principes kan deze aanpak het **beste** worden gebaseerd?

- A)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd op basis van configuratie onder change management.
 - 2. U moet altijd op de hoogte zijn van de actuele status van uw infrastructuur door middel van monitoring en event management.
 - B)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd op basis van configuratie onder change management.
 - 2. U moet altijd op de hoogte zijn van de actuele status van uw infrastructuur door middel van instrumentatie en van incident management.
 - C)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd door middel van configuratie onder versiebeheer.
 - 2. U moet altijd op de hoogte zijn van de actuele status door middel van incident- en event management.
 - D)
 - 1. De gewenste status van uw infrastructuur moet worden gespecificeerd door middel van configuratie onder versiebeheer.
 - 2. U moet altijd op de hoogte zijn van de actuele status van uw infrastructuur door middel van instrumentatie en van monitoring.
- A) Onjuist. De gewenste status van uw infrastructuur moet worden gespecificeerd door middel van een configuratie onder versiebeheer, en niet onder wijzigingen change management. Bovendien is punt 2 onjuist: event management klopt niet en het principe van instrumentatie ontbreekt.
- B) Onjuist. De gewenste status van uw infrastructuur moet worden gespecificeerd door middel van configuratie onder versiebeheer, en niet onder change management. Bovendien is punt 2 onjuist: incident management klopt niet en het principe van monitoring ontbreekt.
- C) Onjuist. Punt 1 is juist. Punt 2 is niet juist: dit is niet een van de principes waarop een holistische benadering het beste kan worden gebaseerd.
- D) Juist. Dit zijn twee van de principes waarop een holistische benadering voor het beheren van een volledige infrastructuur het beste kan worden gebaseerd. (*Literatuur: B, hoofdstuk 11*)

44 / 50

Teams die goed samenwerken, beschikken over gesynchroniseerde werktickets. Een CTO (Chief Technology Officer) heeft aan de hand van 'Go and See' onderzocht hoe het Operations-team functioneert. Na de release voert het Operations-team altijd een herdefinitie van de operationele infrastructuur uit.

Wat is het **beste** advies om deze werkwijze te verbeteren?

- A) Het team moet niets doen. Er is geen verbetering mogelijk, omdat herdefinitie een stap is die altijd moet worden uitgevoerd.
 - B) Het team moet een manier zien te vinden om de operationele infrastructuur en de toegangscontrole van de operationele omgeving te modelleren.
 - C) Het team moet de operationele infrastructuur reviewen, zodat dit een geautomatiseerd proces wordt.
 - D) Het team moet eerst de eigen kennis van het deploymentproces (uitrolproces) gaan delen met het Development-team.
-
- A) Onjuist. Dit is onnodig werk dat beter kan.
 - B) Onjuist. Hoewel dit best handig kan zijn, blijft het team dit steeds herhalen. Dat is verspilling (waste).
 - C) Onjuist. Er bestaat geen duidelijke manier om dit proces te automatiseren zonder Development erbij te betrekken.
 - D) Juist. Dit is de juiste strategie: kennis delen en daarna vervolgstappen nemen. (*Literatuur: A, hoofdstuk 17*)

45 / 50

Op welk moment kan Operations het **beste** Development op de hoogte te stellen van een operationele verandering?

- A) Development hoeft daarvan niet op de hoogte te worden gesteld. Operationele veranderingen zijn alleen bedoeld voor het Operations-team.
 - B) Onmiddellijk. Development moet zo snel mogelijk op de hoogte worden gesteld.
 - C) Tijdens de Scrum of Scrums meeting de volgende ochtend.
 - D) Nadat het Operations-team de acceptatietests heeft uitgevoerd.
-
- A) Onjuist. Development moet onmiddellijk op de hoogte worden gesteld, zodat het team rekening kan houden met eventuele risico's en problemen.
 - B) Juist. Development moet onmiddellijk op de hoogte worden gesteld, zodat het team rekening kan houden met eventuele risico's en problemen. (*Literatuur: C, hoofdstuk 5 en 7*)
 - C) Onjuist. Development moet onmiddellijk op de hoogte worden gesteld, zodat het team rekening kan houden met eventuele risico's en problemen.
 - D) Onjuist. Development moet onmiddellijk op de hoogte worden gesteld, zodat het team rekening kan houden met eventuele risico's en problemen.

46 / 50

U wilt dat uw DevOps-organisatie doorgroeit. Dit kan op veel manieren.

Welke manier is **niet** geschikt om uw DevOps-organisatie te helpen doorgroeien?

- A) Definieer heldere doelstellingen als mijlpalen aan de hand waarvan uw teamleden kunnen bepalen of hun dagelijkse activiteiten waardevol zijn.
 - B) Definieer processen duidelijk en ondersteun de teamleden om het proces elke dag weer te verbeteren.
 - C) Houd verslagen bij van alle vergaderingen, zodat uw teamleden eenvoudig toegang tot alle communicatie hebben.
 - D) Bewaak dagelijkse activiteiten en leg deze vast om kleine dagelijkse successen te kunnen vaststellen en deze te vieren.
-
- A) Onjuist. Dit helpt de DevOps-organisatie een hoger niveau van volwassenheid te bereiken.
 - B) Onjuist. Dit helpt de DevOps-organisatie een hoger niveau van volwassenheid te bereiken.
 - C) Juist. Dit helpt niet de DevOps-organisatie een hoger niveau van volwassenheid te bereiken. Het is niet strikt noodzakelijk om alles van een vergadering vast te leggen. Leg wel vast wat er overeen is gekomen, maar het is niet nodig de volledige vergadering te notuleren. (*Literatuur: B, hoofdstuk 15*)
 - D) Onjuist. Dit helpt de DevOps-organisatie een hoger niveau van volwassenheid te bereiken.

47 / 50

U werkt bij een IT-serviceprovider. Als onderdeel van het bedrijfscontinuïteitsplan wilt u de garantie inbouwen dat u altijd voldoet aan de overeengekomen minimale service levels.

U wilt de continuïteit van de IT-service waarborgen.

Hoe kan DevOps u daarbij helpen?

- A) De culturele DevOps-waarden affiniteit en samenwerking zorgen ervoor dat service hoog in het vaandel staat bij de DevOps-teamleden.
 - B) DevOps bereidt het team voor op Obeya-werkwijzen en aanpak bij calamiteiten door opzettelijk chaos in het systeem in te voeren.
 - C) Risicobeperkende maatregelen en herstelopties zijn hoogstwaarschijnlijk al in de code opgenomen, omdat Operations samenwerkt met Development.
 - D) Service level management wordt belangrijker in DevOps, omdat het de taak van de Process Master is om dit te bewaken.
-
- A) Onjuist. Door het toepassen van culturele waarden voelen mensen zich beter, werken ze beter en in een continu tempo, en leren ze van de fouten die ze maken. Culturele waarden op zich dragen niet bij aan service level management.
 - B) Onjuist. Het invoeren van chaos in het systeem, zoals bij Chaos Monkey van Netflix, kan helpen bij het coderen van risicobeperkende maatregelen en herstelopties, maar op zich draagt dit niet bij aan service level management. Bovendien moet chaos in het systeem worden opgelost met codering, en niet met oplossingen die thuishoren in een 'warroom'.
 - C) Juist. Het is een systematisch proces om risico's met infrastructuur en applicaties te voorkomen, te voorspellen en te beheren, die als er niets aan wordt gedaan, kunnen leiden tot verstoringen en incidenten, met als mogelijk gevolg de onderbreking van services. (*Literatuur: B, hoofdstuk 11 en C, hoofdstuk 4*)
 - D) Onjuist. De Process Master heeft geen primaire taak om service level management op orde te houden. Dit is op z'n minst geen prioriteit in elk DevOps-team.

48 / 50

ACMECONST heeft zijn mondiale aanwezigheid sterk uitgebreid door meer mensen aan te nemen en het aantal engineeringteams wereldwijd te vergroten. Bovendien heeft het bedrijf het klantenbestand met liefst 30% per jaar vergroot.

Beslissingen die ooit gemakkelijk werden genomen toen het engineeringteam zich nog in dezelfde ruimte bevond, nemen nu veel meer tijd in beslag, wat in de organisatie voor de nodige frustratie zorgt. Voor goedkeuring moeten er meer managementlagen worden doorlopen en het proces is veelomvattender geworden. Daardoor raken veel engineers gedesilluseerd over het gehele besluitvormingsproces.

Bovendien is er meer verwarring ontstaan rondom de verantwoordelijkheid voor de verschillende problemen die naar voren worden gebracht, waardoor men terughoudender is geworden in het nemen van beslissingen. De engineers vinden ook dat hun creativiteit wordt tegengewerkt door de extra processen en bureaucratische lijnen, en dat begint hun moreel aan te tasten.

Wat is de **beste** manier om hiermee om te gaan?

- A) Laat de huidige processen intact, maar wees duidelijk over de rollen, verantwoordelijkheid en eigenaarschap voor elk proces, stel een effectieve methode in voor de afweging van productiviteit versus risico's, pas veranderingen stapsgewijs toe en creëer veilige plekken om te experimenteren.
 - B) Neem processen onder de loep om vast te stellen waar zaken kunnen worden gestroomlijnd, wees duidelijk over de rollen, verantwoordelijkheid en eigenschap voor elk proces, stel een effectieve methode in voor de afweging van productiviteit versus risico's, pas veranderingen stapsgewijs toe en creëer veilige plekken om te experimenteren.
 - C) Neem processen onder de loep om vast te stellen waar zaken kunnen worden gestroomlijnd, wees duidelijk over de rollen, verantwoordelijkheid en eigenaarschap voor elk proces, stel een effectieve methode in voor de afweging van productiviteit versus risico's, pas veranderingen stapsgewijs toe en breng de hoeveelheid experimenten terug tot een minimum om onnodige fouten met applicaties te voorkomen.
-
- A) Onjuist. Vasthouden aan bestaande processen is niet het beste idee: er heeft groei plaatsgevonden en de huidige processen werken niet meer. Het creëren van een veilige plek om te experimenteren is daarentegen wel een goed idee.
 - B) Juist. Deze acties werken prima voor een groeiend bedrijf. Het onder de loep nemen van de processen is noodzakelijk omdat de oude processen niet meer functioneren. Een veilige plek om te experimenteren is ook nodig om continue verbetering mogelijk te maken. (*Literatuur: A, hoofdstuk 15*)
 - C) Onjuist. Het onder de loep nemen van processen is noodzakelijk. Maar het minimaliseren van de hoeveelheid experimenten is een slecht idee omdat daardoor de noodzakelijke continue verbetering wordt belemmerd.

49 / 50

Er bestaat binnen bedrijf X-AppGo een conflict tussen het Operations-team in Colombia en het Development-team in Ierland, omdat ze verschillende prioriteiten en doelstellingen hebben. Daardoor kost het steeds meer tijd en moeite om problemen op te lossen die van invloed zijn op het bedrijf.

Welke werkwijzen zou X-AppGo kunnen overwegen om het conflict te verminderen en de samenwerking tussen de Development- en Operations-teams te verbeteren?

- A)**
 - 1. Laat de Development- en Operations-teams onafhankelijk van elkaar werken als ze dat liever hebben, om conflicten te vermijden.
 - 2. Zorg ervoor dat de bedrijfsleiding volledig instemt met de ondersteuning van de Development- en Operations-teams.
 - B)**
 - 1. Laat iemand van de bedrijfsleiding met het DevOps-team gaan praten over het belang van samenwerking.
 - 2. Train de Development- en Operations-teams in DevOps-werkwijzen, zodat ze leren hoe ze elkaars werk kunnen uitvoeren.
 - C)**
 - 1. Zorg ervoor dat de Development- en Operations-teams andere bedrijven bezoeken waar DevOps goed werkt.
 - 2. Vergroot de financiële middelen om de toegenomen aanspraak waar zowel het Operations- als het Development-team mee te maken heeft, te ondersteunen.
 - D)**
 - 1. Beveel onderlinge bedrijfsbezoeken aan tussen de Development- en Operations-teams om de verstandhouding te verbeteren en het vertrouwen en begrip te vergroten.
 - 2. Wissel kennis uit tussen de Development- en Operations-teams zodat ze effectiever kunnen samenwerken.
- A)** Onjuist. De teams moeten worden aangemoedigd om samen te werken; laat niet toe dat ze los van elkaar werken. Ze hebben elkaar nodig en ze moeten van elkaar leren. Samenwerking gaat niet lukken als je elkaar niet ziet. Instemming van de bedrijfsleiding draagt niet werkelijk bij aan een betere samenwerking tussen de teams zelf.
- B)** Onjuist. Hoewel de teams zich wellicht meer gewaardeerd voelen door de inbreng van een lid van de bedrijfsleiding, draagt dat niet bij aan een betere samenwerking tussen de teams. Training heeft maar een beperkt effect. Om ervoor te zorgen dat de teams beter samenwerken, moeten ze gewoon samen aan het werk.
- C)** Onjuist. Een bezoek aan andere bedrijven kan inspirerend zijn, maar DevOps is in hoge mate uniek voor elk afzonderlijk bedrijf. Deze benadering draagt niet rechtstreeks bij aan een betere samenwerking en minder conflicten. In plaats daarvan moeten de teams gaan samenwerken en onderling kennis uitwisselen. Het inzetten van meer financiële middelen kan wel wat helpen wanneer er te veel werk voor te weinig mensen is, maar het heeft slechts een beperkte invloed op het verminderen van conflicten en het verbeteren van de samenwerking.
- D)** Juist. Dit is de meest geschikte werkwijze ter ondersteuning van het huidige scenario voor het verminderen van conflicten en het verbeteren van de samenwerking tussen de Operations- en Development-teams. (*Literatuur: A, hoofdstuk 15*)

50 / 50

Een Development-team is geïnteresseerd in DevOps. Het team is voornamelijk geïnteresseerd in continuous integration (continue integratie, CI). Het team houdt zich momenteel bezig met het ontwikkelen en onderhouden van 3 grote en 4 kleinere oplossingen. Daarbij wordt gebruikgemaakt van de Scrum-werkwijze. Elke sprint duurt 4 weken, waarbij gemiddeld om de 10 of 15 dagen 1 gecommiteerde release voor de testomgeving wordt gemaakt en per maand 1 release voor productie. Het team wil een kwalitatieve business case voor het management opstellen ter ondersteuning van de investeringen en inspanningen om een CI-werkwijze te creëren.

Welke tastbare voordelen van CI helpen die business case daarbij het **beste**?

- A) Eenmaal per dag een deployment (uitrol) in de testomgeving uitvoeren kan voordelig zijn voor het bedrijf en kan de ontwikkelkosten aanzienlijk verlagen.
 - B) Dit draagt bij aan de teamgeest. Omdat er al gebruik wordt gemaakt van Scrum, genereert CI **geen** meetbare voordelen voor het bedrijf.
 - C) Het verhoogt de stabiliteit en kwaliteit van releases met betere en geautomatiseerde tests, waardoor de algehele releasesnelheid toeneemt.
 - D) Eenmaal per dag een release voor productie uitbrengen kan voordelig zijn voor het bedrijf en kan de ontwikkelkosten aanzienlijk verlagen.
-
- A) Onjuist. Sneller deployen in de testomgeving is prima en een gevolg van CI, maar het levert geen enkel voordeel voor het bedrijf op.
 - B) Onjuist. CI kan het team helpen om sneller aan productie te leveren, waarbij bugs eerder en tegen lagere kosten worden gedetecteerd. Het is niet relevant of Scrum daarbij wel of niet wordt gebruikt.
 - C) Juist. Een hogere releasesnelheid zal een gevolg zijn van continuous integration, aangezien elke wijziging functioneel is en wordt geïntegreerd in de hoofdcode, en hierdoor klaar is voor levering aan productie. Dit komt doordat het product altijd in werkende staat verkeert. Bovendien verhoogt de stabiliteit en kwaliteit van releases hiermee, aangezien bugs sneller worden gevonden en verholpen dankzij het geautomatiseerde testen. (*Literatuur: B, hoofdstuk 3*)
 - D) Onjuist. Een snellere release naar productie is een van de belangrijkste voordelen van continuous delivery (continue levering), en niet het directe voordeel van continuous integration. Daarmee wordt namelijk levering naar een productie-achtige omgeving nagestreefd om te waarborgen dat de release in werkende staat verkeert en geautomatiseerde tests zijn geslaagd.

Evaluatie

De juiste antwoorden op de vragen in dit voorbeeldexamen staan in onderstaande tabel.

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

Contact EXIN

www.exin.com

