



**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

Inleiding	4
Voorbeeldexamen	5
Antwoordsleutel	16
Evaluatie	37

# Inleiding

Dit is het voorbeeldexamen EXIN DevOps Professional (DEVOPSP.NL). Op dit examen is het Reglement voor de Examens van EXIN van toepassing.

Dit examen bestaat uit 40 meerkeuzevragen. Elke vraag heeft een aantal antwoorden, waarvan er één correct is.

Het maximaal aantal te behalen punten is 40. Elke goed beantwoorde vraag levert u 1 punt op. Bij 26 punten of meer bent u geslaagd.

De beschikbare tijd is 90 minuten.

Veel succes!

# Voorbeeldexamen

**1 / 40**

Wat is een principe van het Agile Manifesto?

- A) Het creëren van een cultuur van continu en dynamisch leren.
- B) Het integreren van mechanismen voor feedback (terugkoppeling) en feedforward (voorwaartse koppeling) in het werksysteem.
- C) Het regelmatig opleveren van werkende software regelmatig, variërend van een paar weken tot een paar maanden.
- D) Het verbeteren van de flow door werk zichtbaar te maken, de omvang van batches (verwerkingseenheden) en intervallen te verkleinen, en te werken aan de kwaliteit.

**2 / 40**

Bij welke van de drie manieren (Three Ways) hoort 'verbeteringen in dagelijkse werkzaamheden formaliseren'?

- A) Continu leren en experimenteren
- B) Feedback
- C) Flow

**3 / 40**

Wat is het verschil tussen een System of Engagement (verbindingssysteem, SoE) en een System of Record (registratiesysteem, SoR), gelet op het tempo van veranderingen?

- A) Bij een SoE hebben veranderingen doorgaans hetzelfde tempo als bij een SoR.
- B) Bij een SoE ligt het tempo van veranderingen doorgaans veel hoger dan bij een SoR.
- C) Bij een SoE ligt het tempo van veranderingen doorgaans veel lager dan bij een SoR.
- D) Dit verschilt per informatiesysteem.

**4 / 40**

Wat is het voordeel als Development en Operations werken met een gedeelde tool?

- A) Een gemeenschappelijke backlog, waarbij iedereen vanuit een globaal perspectief prioriteit geeft aan verbeteringsprojecten.
- B) Ontwikkelaars krijgen feedback over de prestaties van hun applicaties in productie, inclusief het doen van herstelacties wanneer iets niet werkt.
- C) Het team kan producten deployen (uitrollen) tijdens normale werktijden en eenvoudige omschakelingen doorvoeren.
- D) Kennis in Operations kan worden omgezet in geautomatiseerde code die veel betrouwbaarder is en op allerlei manieren kan worden hergebruikt.

5 / 40

Resultaten kunnen marktgerichter worden gemaakt door de vaardigheden in Operations beter te integreren in Development-teams, zodat deze teams efficiënter en productiever worden.

Met welke aanpak kan dit het **beste** worden bereikt?

- A) Een Development-vertegenwoordiger toewijzen aan het Operations team.
- B) Self-service creëren voor de infrastructuur.
- C) Het Operations team uitbesteden.
- D) Ontwikkelaars trainen in het werk van Operations.

6 / 40

Wegens vertragingen in de ontwikkeling van nieuwe omgevingen door Operations-engineers, duurt het voor een bank langer om een nieuw of aangepast productaanbod op de markt te brengen.

Wat is **waar** met betrekking tot het automatisch aanmaken van omgevingen?

- A) Automatisch aangemaakte omgevingen kunnen worden gebruikt voor alle omgevingen.
- B) Automatisch aangemaakte omgevingen kunnen worden gebruikt voor alle omgevingen, behalve de productieomgeving vanwege veiligheidsredenen.
- C) Wegens het vier-ogen-principe vereist DevOps een controle door Operations wanneer omgevingen in productie automatisch worden aangemaakt.
- D) Wegens het vier-ogen-principe vereist DevOps handmatige goedkeuring door Operations wanneer omgevingen in productie automatisch worden aangemaakt.

7 / 40

Welke waarde moet worden toegevoegd aan een procesblok in een value stream map (waardestroomschema)?

- A) Applicatietelemetrie
- B) Percentage voltooid en nauwkeurig
- C) Velocity van het team
- D) Werk-in-uitvoering (WiP)

8 / 40

Wat is binnen DevOps een **primair** doel van versiebeheer om de risico's voor een bedrijf zo veel mogelijk te beperken?

- A) Zorgen dat er waarschuwingen kunnen worden gegeven wanneer de configuratie verandert in iets anders dan de gewenste staat
- B) Zorgen dat de vorige staat van de testomgeving kan worden hersteld
- C) Zorgen dat de productieomgeving kan worden hersteld en dat processen kunnen worden gebouwd
- D) Zorgen dat de broncode kan worden gedeeld tussen verschillende ontwikkelteams

9 / 40

Een IT-team komt samen om enkele wijzigingen te bespreken die nodig zijn voor de volgende stap in de DevOps-transitie. Het team moet overeenstemming bereiken over een Definition-of-Done (definitie van klaar, DoD) die is afgestemd op de DevOps-principes.

Welke DoD past het **beste** bij DevOps?

- A) De code is geïntegreerd in de primaire aftakking en de geautomatiseerde unittests zijn geslaagd.
- B) De code werkt zoals verwacht op de laptop van de ontwikkelaar en de unittests zijn geslaagd.
- C) De code werkt in een productie-achtige omgeving en de gebruikersacceptatietests zijn geslaagd.

10 / 40

Welke tooling kan het **beste** worden gebruikt om de bouw en configuratie van omgevingen te automatiseren?

- A) Een ticketsysteem voor het inrichten van een ontwikkel-, test- of acceptatieomgeving
- B) Een tool waarmee de instellingen van de productieomgeving worden gekopieerd naar de ontwikkel-, test- en acceptatieomgeving
- C) Configuratiebestanden per omgeving die handmatig worden gedistribueerd en onderhouden om de omgevingen synchroon te houden
- D) Configuration managementbestanden (wijzigingenbeheerbestanden) voor infrastructuur-als-code (infrastructure as code, IaC), waarmee de programmeurs de omgevingen zelf kunnen wijzigen

11 / 40

Een specifiek ontwerpdoel van een suite voor geautomatiseerd testen, is om in een zo vroeg mogelijke testfase fouten te vinden. De ideale testpiramide toont de juiste volgorde waarin tests moeten worden uitgevoerd.

Welke test moet als **eerste** worden uitgevoerd?

- A) Geautomatiseerde API-test
- B) Geautomatiseerde componententest
- C) Geautomatiseerde integratietest
- D) Geautomatiseerde unittest

**12 / 40**

Een DevOps-team wil hun velocity vergroten door gebruik te maken van test-driven development.

Wat is de juiste volgorde van de handelingen?

- A)
  1. Refactoren (code herstructureren)
  2. Een testcase schrijven
  3. De functionele code schrijven
- B)
  1. De functionele code schrijven
  2. Een testcase schrijven
  3. Refactoren
- C)
  1. Een testcase schrijven
  2. De functionele code schrijven
  3. Refactoren

**13 / 40**

Een verzekeringsmaatschappij huurt een DevOps-professional in om het DevOps-team te adviseren bij de keuze voor een aftakingsstrategie. De DevOps-professional evalueert twee strategieën:

1. Optimalisatie voor individuele productiviteit
2. Optimalisatie voor teamproductiviteit

Welke bewering over deze twee strategieën is **waar**?

- A) Beide strategieën resulteren in evenveel samenvoegwerk.
- B) Strategie 1 resulteert in veel meer samenvoegwerk dan strategie 2.
- C) Strategie 2 resulteert in veel meer samenvoegwerk dan strategie 1.

**14 / 40**

Een retailbedrijf heeft het ontwikkelproces radicaal veranderd van het gebruik van waterval naar DevOps-werkwijzen. In korte tijd moeten er veel keuzes worden gemaakt. Hierdoor ontstaat enige technische schuld. Zo is de marktintroductietijd dramatisch toegenomen. Er bestaan echter manieren om veel vertragingen te elimineren.

Uit een value stream map (waardestroomschema) blijkt dat 20% van de Sprint-tijd wordt besteed aan het overzetten van codes, die nu allemaal worden onderhouden op afzonderlijke code-aftakkingen.

Welke oplossing voor deze technische schuld resulteert in een snellere flow?

- A) Overstappen op ontwikkeling op basis van een stam (trunk)
- B) Testfouten reproduceren op de werkstations van ontwikkelaars
- C) Beginnen met kanarie-releases
- D) Meer telemetrie gebruiken



15 / 40

Wat is een kenmerk van een architectuur die productiviteit, testbaarheid en veiligheid mogelijk maakt?

- A) Losjes gedefinieerde interfaces
- B) Strak gekoppelde architectuur
- C) Goed gedefinieerde API's

16 / 40

Wat is de **beste** telemetrie-aanpak om problemen binnen DevOps op te lossen?

- A) Investeren in telemetrie-tooling is belangrijk. De focus moet daarbij liggen op de productieomgeving, de deployment pipeline (uitrolpijplijn) en pre-productie.
- B) Investeren in telemetrie-tooling is belangrijk. De focus moet daarbij uitsluitend op de productieomgeving liggen.
- C) Investeren in telemetrie-tooling is niet belangrijk. De focus moet liggen op het herstarten van redundante services, wat veel goedkoper is.
- D) Investeren in telemetrie-tooling is niet belangrijk. De focus moet liggen op de gebruiker van de applicatie, die een veel completere en goedkopere monitoring-voorziening vormt.

17 / 40

Binnen het monitoring-raamwerk moeten gegevens uit drie lagen worden verzameld.

Wat is **niet** een van die drie lagen?

- A) Applicatie
- B) Bedrijfslogica
- C) Bedrijfsmetrieken
- D) Besturingssysteem

18 / 40

Voor welke doelgroep heeft self-service-toegang tot telemetrie meerwaarde?

- A) Alleen voor ontwikkelaars
- B) Alleen voor ontwikkelaars en operators
- C) Alleen voor ontwikkelaars, operators en belanghebbenden
- D) Voor ontwikkelaars, operators, belanghebbenden en klanten

19 / 40

Welke techniek biedt in een omgeving met geautomatiseerde tests, snelle deployment (uitrol) processen en voldoende telemetrie de **meeste** DevOps-voordelen voor een bedrijf?

- A) Fix-forward
- B) Roll-back (terugdraaien)
- C) Verwijderen van defecte servers
- D) Uitschakelen van defecte functies

**20 / 40**

Een softwarebedrijf heeft de volgende vereisten voor lanceringsrichtlijnen opgesteld:

- Aantal en ernst van defecten: Werkt de applicatie conform het ontwerp?
- Dekking van de monitoring: Is de dekking van de monitoring voldoende om een service te herstellen wanneer er dingen fout gaan?
- Systeemarchitectuur: Is de service strak genoeg gekoppeld om een groot aantal incidenten in productie op te lossen?
- Type/frequentie van pager-meldingen: Genereert de applicatie een ondoenlijk aantal meldingen in productie?

Welke van deze vereisten voor lanceringsrichtlijnen voldoet **niet** aan de DevOps-werkwijze?

- A) Aantal en ernst van defecten: Werkt de applicatie conform het ontwerp?
- B) Dekking van de monitoring: Is de dekking van de monitoring voldoende om een service te herstellen wanneer het fout gaat?
- C) Systeemarchitectuur: Is de service strak genoeg gekoppeld om een groot aantal incidenten in productie op te lossen?
- D) Type/frequentie van pager-meldingen: Genereert de applicatie een ondoenlijk aantal meldingen in productie?

**21 / 40**

Wat is een uniek kenmerk van de hand-off readiness review (gereedheidsbeoordeling bij overdracht, HRR) en **niet** van de launch readiness review (gereedheidsbeoordeling bij lancering, LRR)?

- A) HRR is veel strenger en heeft hogere acceptatiestandaarden.
- B) HRR wordt door de productteams zelf gerapporteerd.
- C) HRR wordt officieel goedgekeurd voordat een nieuwe service algemeen beschikbaar wordt gemaakt.
- D) HRR wordt officieel goedgekeurd voordat een nieuwe service live-productieverkeer ontvangt.

**22 / 40**

Een van de krachtigste technieken voor het ontwerpen van de interactie- en gebruikerservaring (UX) is contextgebonden onderzoek.

Wat is de **beste** omschrijving van contextgebonden onderzoek?

- A) Het productteam vraagt gebruikers te reageren op een demonstratie van de applicatie, die door hen wordt gegeven.
- B) Het productteam interviewt gebruikers over het thuisgebruik van de applicatie.
- C) Het productteam observeert gebruikers die de applicatie in hun natuurlijke omgeving gebruiken.
- D) Het productteam bestudeert gebruikers tijdens de gebruikersacceptatietest in een speciale ruimte met testapparatuur.

**23 / 40**

Start-up ABC heeft behoorlijke moeite met het ontwikkelen van functies voor een mobiele navigatie-applicatie die voldoet aan de verwachtingen van de gebruiker.

Wat is de **beste** aanpak om te zorgen dat de functies direct aansluiten bij de verwachtingen van de gebruiker over de gebruikerservaring (user-experience, UX) van de mobiele navigatie-applicatie?

- A) De UX van de mobiele navigatie-applicatie volledig ontwikkelen, zodat consumenten beschikken over meer geavanceerde functies en keuzes om door de UX te navigeren, waardoor ze meer opties hebben om de UX aan te passen aan hun voorkeuren.
- B) Een UX van de mobiele navigatie-applicatie ontwikkelen waarin consumenten willekeurig worden geselecteerd voor de weergave van een van twee UX-versies: een controleversie (control) of een behandelde versie (treatment).
- C) Correcties aanbrengen aan de huidige UX-functies van de mobiele navigatie-applicatie, zodat ABC kan vasthouden aan de kernfunctionaliteit van de UX zonder dat dit ten koste gaat van de kwaliteit.
- D) Eén functie voor de mobiele navigatie-applicatie vrijgeven om gedurende een periode van drie weken feedback van de consument te krijgen over die functies.

**24 / 40**

Welke aanpak kan een Development-team gebruiken om levering van de gewenste bedrijfsresultaten te ondersteunen?

- A) Zorgvuldige uitvoering van veel experimenten
- B) Volledige regressietests
- C) Hypothese-gedreven ontwikkeling
- D) SDLC (Software Development Life Cycle)

**25 / 40**

Bedrijf ABC vond het in het verleden lastig om inzicht te krijgen in de uitwerking van codewijzigingen in de payroll-applicatie. Momenteel zijn geïmplementeerde codewijzigingen slecht gedocumenteerd. De laatste codecorrectie werd bijvoorbeeld als volgt gedocumenteerd: "Probleem nr. 1801 voor payroll-applicatie herstellen". Het bedrijf merkt dat dit een slechte pull-aanvraag (vraaggestuurde aanvraag) is.

Een goede pull-aanvraag omvat de volgende aspecten:

- Waarom implementeren we de oplossing?
- Wie implementeert de oplossing?

Wat **moet** nog meer worden vermeld in een goede pull-aanvraag?

- A) De business-unit waar de oplossing van toepassing is
- B) De mogelijke risico's en tegenmaatregelen voor de implementatie van de oplossing
- C) De releaseplanning voor de implementatie van de oplossing

**26 / 40**

Dit is het scenario:

"De ontwikkelaar doorloopt de code terwijl een collega feedback geeft."

Welke beoordelingstechniek wordt hier beschreven?

- A) Over-de-schouder
- B) In tweetallen programmeren (pair programming)
- C) Collegiale beoordeling (peer review)
- D) Door tools ondersteunde beoordeling

**27 / 40**

Welke beoordelingstechniek resulteert direct in een afname van het aantal codefouten?

- A) Over-de-schouder
- B) In tweetallen programmeren (pair programming)
- C) Collegiale beoordeling (peer review)
- D) Door tools ondersteunde beoordeling

**28 / 40**

Een cloud-serviceprovider wil de capaciteit van de service vergroten met behulp van een Simian Army Monkey.

Welke Monkey heeft de provider in dit geval nodig?

- A) Doctor Monkey
- B) Janitor Monkey
- C) Latency Monkey

**29 / 40**

Bedrijf ABC wil overstappen op de DevOps-werkwijze en een open, niet-verwijtende leeromgeving promoten. ABC werd onlangs getroffen door een grote applicatiestoring en wist de applicatieservice toen te herstellen.

Wat is de **eerste** taak die moet worden uitgevoerd in de verwijtloze post-mortem (bespreking achteraf)?

- A) Een tijdlijn van relevante gebeurtenissen opstellen in de volgorde dat deze zich voordeden tijdens de grote applicatiestoring
- B) Tegenmaatregelen identificeren om te voorkomen dat de grote applicatiestoring zich in de toekomst nogmaals voordoet
- C) De onderliggende oorzaak (root cause) van de grote applicatiestoring identificeren om een corrigerende actie te kunnen voorstellen waarmee wordt voorkomen dat de grote applicatiestoring zich nogmaals voordoet
- D) De post-mortem publiceren op een centrale locatie waar de hele organisatie toegang tot heeft, zodat iedereen kan leren van de grote applicatiestoring

**30 / 40**

Wat is vereist bij het creëren van weerbaarheid door het injecteren van productiefouten?

- A) Definiëren van de foutmodus
- B) Organisatie van post-mortems (bespreking achteraf)
- C) Training van mensen
- D) Gebruik van een testomgeving

**31 / 40**

Wat is de **eerste** stap in een plan voor een game-dag?

- A) Oefeningen vaststellen en uitvoeren
- B) Problemen identificeren, oplossen en testen
- C) De uitval plannen
- D) SPOF's (Single Points of Failure) voorbereiden en wegnemen

**32 / 40**

Wat is een voorbeeld van een niet-functionele vereiste (nfr)?

- A) Voor- en achterwaartse compatibiliteit tussen versies
- B) De mogelijkheid om melding te maken van vertragingen in planningen
- C) Registratie van financiële transacties in een boekingsstelsel van een hotel

**33 / 40**

Welke activiteit moet worden uitgevoerd om succesvol herbruikbare user story's voor Operations te maken?

- A) User story's voor Operations koppelen aan relevante verbeteringen en fouten in de ontwikkeling
- B) Activiteiten binnen het overdrachtsproces (handoff process) definiëren en deze activiteiten vervolgens automatiseren met toepasselijke tools en ondersteunende workflows
- C) Aanwijzen van de nodige operationele werkzaamheden en degenen die ze uitvoeren

**34 / 40**

Bekijk de volgende elementen:

1. Wijzigingsaanvragen
2. Tools voor de deployment pipeline (uitrolopijplijn)
3. Gecompileerde uitvoerbare programmabestanden
4. Handleidingen en standaarden

Welke **twee** elementen worden doorgaans opgeslagen in een enkelvoudige opslagplaats voor gedeelde broncode (single shared source code repository)?

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

**35 / 40**

Met welk doel worden lokale ontdekkingen omgezet in globale verbeteringen?

- A) Zodat de stand van zaken niet alleen voor Dev en Ops wordt verbeterd, maar voor de hele organisatie.
- B) Zodat alle nieuwe en bestaande services gemakkelijker gebruik kunnen maken van de collectieve kennis.
- C) Zodat de werkcultuur meer een cultuur van samenwerking wordt en de systemen veiliger en weerbaarder worden.
- D) Zodat er een cultuur ontstaat waarin iedereen zich op zijn en haar gemak en verantwoordelijk voelt.

**36 / 40**

Ontwikkelaars kunnen het voor engineers gemakkelijk maken om registratie- en versleutelingsstandaarden correct toe te voegen aan en te gebruiken in hun applicaties en omgevingen.

Wat is **geen** item in een opslagplaats voor gedeelde broncode (shared source code repository) waarmee dit wordt ondersteund?

- A) Codebibliotheken en de bijbehorende aanbevolen configuraties
- B) Deployment packages (uitrolpakketten)
- C) Pakketten en builds van een besturingssysteem (OS packages en builds)
- D) Verborgene management-tools

**37 / 40**

Wanneer ontwikkelaars code introduceren, bestaat er altijd een risico dat hiermee onbevoegde toegang mogelijk wordt gemaakt.

Met welke maatregel wordt dat risico **niet** beperkt?

- A) Codebeoordelingen
- B) Codetests
- C) Effectieve patching
- D) Penetratietests

**38 / 40**

Wat is een voorbeeld van het creëren van telemetrie in een applicatie?

- A) Wijzigingen in een besturingssysteem
- B) Dagelijkse controle van systeemlogboeken
- C) Wijzigingen in een beveiligingsgroep
- D) Reset van gebruikerswachtwoorden

**39 / 40**

Welk proces is een primaire maatregel om operationele en beveiligingsgerelateerde risico's te beperken en ondersteunt ook naleving?

- A) Proces voor change management (wijzigingenbeheer)
- B) Proces voor configuration management
- C) Proces voor release- en deploymentmanagement (uitrolmanagement)
- D) Proces voor service-levelmanagement

**40 / 40**

Wat is een nadeel als een bedrijf kiest voor de maatregel 'scheiding van taken'?

- A) De 'scheiding van taken' maatregel staat vaak ontwikkelingspanningen in de weg, omdat de feedback die engineers op hun werk krijgen, wordt vertraagd en afneemt.
- B) Bij een 'scheiding van taken' maatregel moet de ontwikkelaar wijzigingen indienen bij een codebeheerder, die de wijziging beoordeelt en goedkeurt voordat deze naar productie wordt doorgezet.
- C) Een 'scheiding van taken' maatregel leidt tot onnodig werk, omdat ingecheckte code moet worden geïnspecteerd en beoordeeld, wat enkel leidt tot onnodige zekerheid over de kwaliteit van het werk.

# Antwoordsleutel

1 / 40

Wat is een principe van het Agile Manifesto?

- A) Het creëren van een cultuur van continu en dynamisch leren.
  - B) Het integreren van mechanismen voor feedback (terugkoppeling) en feedforward (voorwaartse koppeling) in het werksysteem.
  - C) Het regelmatig opleveren van werkende software regelmatig, variërend van een paar weken tot een paar maanden.
  - D) Het verbeteren van de flow door werk zichtbaar te maken, de omvang van batches (verwerkingseenheden) en intervallen te verkleinen, en te werken aan de kwaliteit.
- 
- A) Onjuist. Dit hoort bij het principe van continu leren en experimenteren.
  - B) Onjuist. Dit hoort bij het principe van feedback.
  - C) Juist. Dit is een van de belangrijkste principes van het Agile Manifesto. Andere principes zijn: de noodzaak van kleine, zelfsturende teams en werken met een managementmodel dat is gebaseerd op vertrouwen. (Literatuur: A, inleiding van deel 1)
  - D) Onjuist. Dit hoort bij het principe van flow.

2 / 40

Bij welke van de drie manieren (Three Ways) hoort 'verbeteringen in dagelijkse werkzaamheden formaliseren'?

- A) Continu leren en experimenteren
  - B) Feedback
  - C) Flow
- 
- A) Juist. Het principe van de derde manier (Third Way) vereist het formaliseren van verbeteringen in dagelijkse werkzaamheden, het omzetten van lokaal leren naar globaal leren dat in de hele organisatie kan worden gebruikt, plus het boeiend houden van dagelijkse werkzaamheden. (Literatuur: A, Hoofdstuk 4)
  - B) Onjuist. 'Verbeteringen in dagelijkse werkzaamheden formaliseren' is een principe van continu leren en experimenteren.
  - C) Onjuist. 'Verbeteringen in dagelijkse werkzaamheden formaliseren' is een principe van continu leren en experimenteren.



**3 / 40**

Wat is het verschil tussen een System of Engagement (verbindingssysteem, SoE,) en een System of Record (registratiesysteem, SoR,) gelet op het tempo van veranderingen?

- A) Bij een SoE hebben veranderingen doorgaans hetzelfde tempo als bij een SoR.
  - B) Bij een SoE ligt het tempo van veranderingen doorgaans veel hoger dan bij een SoR.
  - C) Bij een SoE ligt het tempo van veranderingen doorgaans veel lager dan bij een SoR.
  - D) Dit verschilt per informatiesysteem.
- 
- A) Onjuist. Bij een SoE ligt het tempo van veranderingen doorgaans veel hoger dan bij een SoR.
  - B) Juist. Het tempo van veranderingen ligt bij een SoE hoger, aangezien dit systeem de gebruikersinterface vertegenwoordigt. Bovendien is het complexer om een SoR te veranderen. (Literatuur: A, Hoofdstuk 5)
  - C) Onjuist. Bij een SoE ligt het tempo van veranderingen doorgaans veel hoger dan bij een SoR.
  - D) Onjuist. Hier kan wel een algemene bewering over worden gedaan. Bij een SoE ligt het tempo van veranderingen doorgaans veel hoger dan bij een SoR.

**4 / 40**

Wat is het voordeel als Development en Operations werken met een gedeelde tool?

- A) Een gemeenschappelijke backlog, waarbij iedereen vanuit een globaal perspectief prioriteit geeft aan verbeteringsprojecten.
  - B) Ontwikkelaars krijgen feedback over de prestaties van hun applicaties in productie, inclusief het doen van herstelacties wanneer iets niet werkt.
  - C) Het team kan producten deployen (uitrollen) tijdens normale werktijden en eenvoudige omschakelingen doorvoeren.
  - D) Kennis in Operations kan worden omgezet in geautomatiseerde code die veel betrouwbaarder is en op allerlei manieren kan worden hergebruikt.
- 
- A) Juist. Met een gemeenschappelijke backlog geeft iedereen vanuit een globaal perspectief prioriteit aan verbeteringsprojecten, zodat werk wordt geselecteerd dat de meeste waarde heeft voor de organisatie of de meeste technische schuld vermindert. (Literatuur: A, Hoofdstuk 6)
  - B) Onjuist. Dit is een voordeel dat Development deelt wanneer oproeptaken worden gerouleerd met Operations.
  - C) Onjuist. Dit is een voordeel van het automatiseren en faciliteren van releases met een laag risico.
  - D) Onjuist. Dit is een voordeel van het toevoegen van Operations engineers aan serviceteams.

5 / 40

Resultaten kunnen marktgerichter worden gemaakt door de vaardigheden in Operations beter te integreren in Development-teams, zodat deze teams efficiënter en productiever worden.

Met welke aanpak kan dit het **beste** worden bereikt?

- A) Een Development-vertegenwoordiger toewijzen aan het Operations team.
  - B) Self-service creëren voor de infrastructuur.
  - C) Het Operations team uitbesteden.
  - D) Ontwikkelaars trainen in het werk van Operations.
- 
- A) Onjuist. Het zou juist andersom moeten zijn: 'Een vertegenwoordiger van Operations toewijzen aan het Development-team'.
  - B) Juist. Dit is een van de drie brede strategieën voor de integratie van Operations in Development:
    - self-service creëren zodat ontwikkelaars in serviceteams productiever kunnen zijn;
    - engineers uit Operations integreren in serviceteams;
    - een Operations contactpersoon toewijzen als het niet mogelijk is om een Operations engineer te integreren. (Literatuur: A, Hoofdstuk 8)
  - C) Onjuist. Uitbesteden wordt niet aangeraden als strategie. De medewerkers van de partij naar wie uitbesteed wordt, staan dan nog verder af van het Development-team.
  - D) Onjuist. Ontwikkelaars kunnen infrastructuur-als-code (infrastructure as code, IaC) gebruiken, maar daarmee komt de functie van Operations nog niet te vervallen.

6 / 40

Wegens vertragingen in de ontwikkeling van nieuwe omgevingen door Operations-engineers, duurt het voor een bank langer om een nieuw of aangepast productaanbod op de markt te brengen.

Wat is **waar** met betrekking tot het automatisch aanmaken van omgevingen?

- A) Automatisch aangemaakte omgevingen kunnen worden gebruikt voor alle omgevingen.
  - B) Automatisch aangemaakte omgevingen kunnen worden gebruikt voor alle omgevingen, behalve de productieomgeving vanwege veiligheidsredenen.
  - C) Wegens het vier-ogen-principe vereist DevOps een controle door Operations wanneer omgevingen in productie automatisch worden aangemaakt.
  - D) Wegens het vier-ogen-principe vereist DevOps handmatige goedkeuring door Operations wanneer omgevingen in productie automatisch worden aangemaakt.
- 
- A) Juist. Door het gebruik van scripts zijn er bij continuous deployment (continue uitrol) geen handmatige acties vereist. Deze scripts moeten worden geautomatiseerd. Dit geldt voor alle omgevingen. (Literatuur: A, Hoofdstuk 9)
  - B) Onjuist. De productieomgeving moet volledig automatisch worden gemaakt en onderhouden.
  - C) Onjuist. Het is niet nodig om geautomatiseerde scripts te reviewen. Geautomatiseerde processen worden toch al gemonitord.
  - D) Onjuist. Het handmatige vier-ogen-principe is niet vereist, aangezien de acceptatie van de scripts en de deployment-tool (uitrol-tool) van de scripts onder change management (wijzigingenbeheer) vallen.

7 / 40

Welke waarde moet worden toegevoegd aan een procesblok in een value stream map (waardestroomschema)?

- A) Applicatietelemetrie
- B) Percentage voltooid en nauwkeurig
- C) Velocity van het team
- D) Werk-in-uitvoering (WiP)

- A) Onjuist. Applicatietelemetrie wordt gebruikt om het gedrag van een applicatie te monitoren terwijl deze draait, om zodoende verslag uit te brengen over de status. Hierdoor is het mogelijk om snel en nauwkeurig feedback (terugkoppeling) te geven en fouten te detecteren.
- B) Juist. Dit moet worden toegevoegd aan een procesblok in een value stream map, aangezien het een belangrijke indicator is die inzicht geeft in de kwaliteit van de waarde stroom zoals het bedrijf die nodig heeft. (Literatuur: A, Hoofdstuk 6)
- C) Onjuist. De velocity van het team is een waarde die wordt gemeten tijdens een Sprint en die tijdens een Sprint-bijeenkomst wordt gebruikt om te schatten hoeveel werk er tijdens de Sprint kan worden gedaan.
- D) Onjuist. WiP is een concept dat moet worden toegevoegd aan of afgeleid van een Kanban-bord. Het maakt geen deel uit van een value stream map en heeft ook geen waarde voor de value stream-analyse (waardestroomanalyse).

8 / 40

Wat is binnen DevOps een **primair** doel van versiebeheer om de risico's voor een bedrijf zo veel mogelijk te beperken?

- A) Zorgen dat er waarschuwingen kunnen worden gegeven wanneer de configuratie verandert in iets anders dan de gewenste staat
- B) Zorgen dat de vorige staat van de testomgeving kan worden hersteld
- C) Zorgen dat de productieomgeving kan worden hersteld en dat processen kunnen worden gebouwd
- D) Zorgen dat de broncode kan worden gedeeld tussen verschillende ontwikkelteams

- A) Onjuist. Hoewel versiebeheer belangrijk is omdat andere tools deze informatie gebruiken om te detecteren wanneer configuraties afwijken van de gewenste staat, is dit niet het doel van versiebeheer op zich. Dit is hier dus ook niet het beste antwoord, aangezien versiebeheer het mogelijk moet maken om omgevingen sneller opnieuw te maken en zodoende de kwaliteit van services en de flow te behouden.
- B) Onjuist. Dit kan inderdaad een voordeel van versiebeheer zijn, maar is niet het primaire doel ervan binnen DevOps. Bovendien wordt de testomgeving doorgaans voor elke test opnieuw gemaakt, dus is het niet erg nuttig om een vorige staat ervan te herstellen.
- C) Juist. Versiebeheer moet binnen DevOps de mogelijkheid bieden om productieomgevingen te herstellen en processen te bouwen. Organisaties die kiezen voor deze aanpak, presteren beter dan organisaties die dat niet doen. (Literatuur: A, Hoofdstuk 9)
- D) Onjuist. Dit is het doel van versiebeheer wanneer het wordt gebruikt met de focus puur op Development. Dit is echter niet het doel binnen DevOps, aangezien voor versiebeheer in DevOps alle gebieden hun artefacten, tools en processen moeten bewaren om te kunnen presteren, flow te realiseren en communicatie tussen teams en voorspelbaarheid, met minder verspilling (waste) in alle omgevingen.

9 / 40

Een IT-team komt samen om enkele wijzigingen te bespreken die nodig zijn voor de volgende stap in de DevOps-transitie. Het team moet overeenstemming bereiken over een Definition-of-Done (definitie van klaar, DoD) die is afgestemd op de DevOps-principes.

Welke DoD past het **beste** bij DevOps?

- A) De code is geïntegreerd in de primaire aftakking en de geautomatiseerde unittests zijn geslaagd.
- B) De code werkt zoals verwacht op de laptop van de ontwikkelaar en de unittests zijn geslaagd.
- C) De code werkt in een productie-achtige omgeving en de gebruikersacceptatietests zijn geslaagd.

- A) Onjuist. Dit is de standaard DoD vanuit het perspectief van een ontwikkelaar. Hiermee wordt het doel van DevOps niet gerealiseerd, namelijk meerwaarde bieden aan het bedrijf. Daarom moet de DoD ook verzending van werkende code naar een productie-achtige omgeving omvatten en geen integratie in de primaire aftakking.
- B) Onjuist. Het feit dat alles werkt op de laptop van de ontwikkelaar, biedt geen enkele garantie dat het ook zal werken in een productie-achtige omgeving.
- C) Juist. Op deze manier wordt gewaarborgd dat de code werkt zoals verwacht en klaar is voor verzending (potentially shippable). (Literatuur: A, Hoofdstuk 9)

10 / 40

Welke tooling kan het **beste** worden gebruikt om de bouw en configuratie van omgevingen te automatiseren?

- A) Een ticketsysteem voor het inrichten van een ontwikkel-, test- of acceptatieomgeving
  - B) Een tool waarmee de instellingen van de productieomgeving worden gekopieerd naar de ontwikkel-, test- en acceptatieomgeving
  - C) Configuratiebestanden per omgeving die handmatig worden gedistribueerd en onderhouden om de omgevingen synchroon te houden
  - D) Configuration managementbestanden (wijzigingenbeheerbestanden) voor infrastructuur-als-code (infrastructure as code, IaC), waarmee de programmeurs de omgevingen zelf kunnen wijzigen
- 
- A) Onjuist. Een ticketsysteem voor het inrichten van omgevingen belemmert de flow in de deployment pipeline (uitrolpijplijn), veroorzaakt meer handmatige overdrachten en is daarom verspilling (waste). Er moeten tools gebruikt worden om het inrichten van omgevingen te automatiseren.
  - B) Onjuist. Dit is geen aanbevolen werkwijze in DevOps. Een omgeving moet vanaf nul worden gebouwd. Bovendien mogen productiegegevens niet worden gebruikt in de ontwikkel-, test- en acceptatieomgeving.
  - C) Onjuist. Configuratiebestanden voor omgevingen moeten automatisch worden gedistribueerd.
  - D) Juist. Dit is een van de opties om de bouw en configuratie van de omgeving te automatiseren. Infrastructuur als code kan door de ontwikkelaars worden gebruikt om de omgevingen zelf te bouwen en configureren. (Literatuur: A, Hoofdstuk 9)

**11 / 40**

Een specifiek ontwerpdoel van een suite voor geautomatiseerd testen, is om in een zo vroeg mogelijke testfase fouten te vinden. De ideale testpiramide toont de juiste volgorde waarin tests moeten worden uitgevoerd.

Welke test moet als **eerste** worden uitgevoerd?

- A) Geautomatiseerde API-test
  - B) Geautomatiseerde componententest
  - C) Geautomatiseerde integratietest
  - D) Geautomatiseerde unittest
- 
- A) Onjuist. De geautomatiseerde unittest moet als eerste worden uitgevoerd. Deze test is de 4e in een reeks van 5.
  - B) Onjuist. De geautomatiseerde unittest moet als eerste worden uitgevoerd. Deze test is de 2e in een reeks van 5.
  - C) Onjuist. De geautomatiseerde unittest moet als eerste worden uitgevoerd. Deze test is de 3e in een reeks van 5.
  - D) Juist. Dit is de 1e test die moet worden uitgevoerd. (Literatuur: A, Hoofdstuk 10)

**12 / 40**

Een DevOps-team wil hun velocity vergroten door gebruik te maken van test-driven development.

Wat is de juiste volgorde van de handelingen?

- A)
    1. Refactoren (code herstructureren)
    2. Een testcase schrijven
    3. De functionele code schrijven
  - B)
    1. De functionele code schrijven
    2. Een testcase schrijven
    3. Refactoren
  - C)
    1. Een testcase schrijven
    2. De functionele code schrijven
    3. Refactoren
- 
- A) Onjuist. Schrijf eerst een test case, daarna pas de code. Refactoren volgt als derde stap.
  - B) Onjuist. Schrijf eerst een test case, daarna pas de code. Refactoren volgt als derde stap.
  - C) Juist. Schrijf eerst de test case, daarna pas de code. Refactoren volgt als derde stap. (Literatuur: A, Hoofdstuk 10)

**13 / 40**

Een verzekeringsmaatschappij huurt een DevOps-professional in om het DevOps-team te adviseren bij de keuze voor een aftakingsstrategie. De DevOps-professional evalueert twee strategieën:

1. Optimalisatie voor individuele productiviteit
2. Optimalisatie voor teamproductiviteit

Welke bewering over deze twee strategieën is **waar**?

- A) Beide strategieën resulteren in evenveel samenvoegwerk.
- B) Strategie 1 resulteert in veel meer samenvoegwerk dan strategie 2.
- C) Strategie 2 resulteert in veel meer samenvoegwerk dan strategie 1.

- A) Onjuist. Omdat strategie 1 resulteert in een aftakking per persoon, wordt het samenvoegen zeer problematisch. Strategie 2 is veel beter, ongeacht het systeem dat voor versiebeheer wordt gebruikt.
- B) Juist. Individuele productiviteit komt neer op een aftakking per DevOps-ontwikkelaar, waarmee deze individueel software kan ontwikkelen zonder integratie. Bij strategie 2 zijn er helemaal geen aftakkingen, waardoor de ontwikkeling plaatsvindt in een 'ononderbroken lijn'. (Literatuur: A, Hoofdstuk 11)
- C) Onjuist. Strategie 2 resulteert in veel minder samenvoegwerk omdat iedereen in een gemeenschappelijke omgeving werkt.

**14 / 40**

Een retailbedrijf heeft het ontwikkelproces radicaal veranderd van het gebruik van waterval naar DevOps-werkwijzen. In korte tijd moeten er veel keuzes worden gemaakt. Hierdoor ontstaat enige technische schuld. Zo is de marktintroductietijd dramatisch toegenomen. Er bestaan echter manieren om veel vertragingen te elimineren.

Uit een value stream map (waardestroomschema) blijkt dat 20% van de Sprint-tijd wordt besteed aan het overzetten van codes, die nu allemaal worden onderhouden op afzonderlijke code-aftakkingen.

Welke oplossing voor deze technische schuld resulteert in een snellere flow?

- A) Overstappen op ontwikkeling op basis van een stam (trunk)
- B) Testfouten reproduceren op de werkstations van ontwikkelaars
- C) Beginnen met kanarie-releases
- D) Meer telemetrie gebruiken

- A) Juist. Bij ontwikkeling op basis van een stam zijn aftakkingen niet toegestaan, waardoor er niets samengevoegd hoeft te worden. (Literatuur: A, Hoofdstuk 11)
- B) Onjuist. Het probleem schuilt in de aftakkingen, waardoor het samenvoegen problemen veroorzaakt. De vertraging wordt niet verholpen met het uitvoeren van tests.
- C) Onjuist. De vertraging wordt niet verholpen door het uitvoeren van tests, zelfs als kanarie-releases een positief effect zouden hebben vanuit een continuous deployment (continue uitrol) perspectief.
- D) Onjuist. Het probleem door het samenvoegen wordt niet verholpen met telemetrie.

15 / 40

Wat is een kenmerk van een architectuur die productiviteit, testbaarheid en veiligheid mogelijk maakt?

- A) Losjes gedefinieerde interfaces
- B) Strak gekoppelde architectuur
- C) Goed gedefinieerde API's

- A) Onjuist. Goed gedefinieerde interfaces zijn een kenmerk van een architectuur die productiviteit, testbaarheid en veiligheid mogelijk maakt, aangezien deze afdwingen hoe modules verbinding maken met elkaar. Losjes gedefinieerde interfaces doen dat niet.
- B) Onjuist. Een strak gekoppelde architectuur is monolithisch of bevat modules die te nauw met elkaar verbonden zijn. Elke keer dat we code aan de stam (trunk) proberen toe te voegen (te committen), lopen we het risico algemene fouten te veroorzaken. Voor elke kleine verandering is veel communicatie en coördinatie nodig die dagen kan duren, plus goedkeuring van elke groep waarop de verandering mogelijk van invloed is.
- C) Juist. Goed gedefinieerde API's (Application Programming Interfaces) maken een architectuur mogelijk die productiviteit, gemakkelijker testen van services en veiligheid ondersteunt. (Literatuur: A, Hoofdstuk 13)

16 / 40

Wat is de **beste** telemetrie-aanpak om problemen binnen DevOps op te lossen?

- A) Investeren in telemetrie-tooling is belangrijk. De focus moet daarbij liggen op de productieomgeving, de deployment pipeline (uitrolopijplijn) en pre-productie.
  - B) Investeren in telemetrie-tooling is belangrijk. De focus moet daarbij uitsluitend op de productieomgeving liggen.
  - C) Investeren in telemetrie-tooling is niet belangrijk. De focus moet liggen op het herstarten van redundante services, wat veel goedkoper is.
  - D) Investeren in telemetrie-tooling is niet belangrijk. De focus moet liggen op de gebruiker van de applicatie, die een veel completere en goedkopere monitoring-voorziening vormt.
- 
- A) Juist. Problemen doen zich niet alleen voor in de productieomgeving, maar ook op andere plaatsen. Bij DevOps is snelle feedback (terugkoppeling) vereist. Die feedback begint vooraan in de pijplijn. (Literatuur: A, Hoofdstuk 14)
  - B) Onjuist. Het is niet voldoende om de focus alleen op de productieomgeving te hebben. De deployment pipeline en pre-productie zijn ook belangrijk om fouten vroegtijdig te herkennen.
  - C) Onjuist. Herstarten is niet goedkoper. Er gaat veel productietijd verloren met het opsporen van de fout.
  - D) Onjuist. Als de gebruiker als enige de service monitort, gaat er veel productietijd verloren. Het monitoren van de hele pijplijn (mits goed geïmplementeerd) kan veel sneller detecteren waar iets fout gaat en wat. Bovendien vindt de gebruiker alleen fouten die hij of zij kan zien, en niet de fouten die zich op de achtergrond voordoen.

17 / 40

Binnen het monitoring-raamwerk moeten gegevens uit drie lagen worden verzameld.

Wat is **niet** een van die drie lagen?

- A) Applicatie
- B) Bedrijfslogica
- C) Bedrijfsmetriecken
- D) Besturingssysteem

- A) Onjuist. 'Applicatie' is een van de drie lagen waaruit gegevens moeten worden opgehaald in het monitoring-raamwerk.
- B) Onjuist. 'Bedrijfslogica' is een van de drie lagen waaruit gegevens moeten worden opgehaald in het monitoring-raamwerk.
- C) Juist. 'Bedrijfsmetriecken' is niet een van de drie lagen waaruit gegevens moeten worden opgehaald in het monitoring-raamwerk. Bedrijfsmetriecken zijn juist het resultaat van die monitoring. (Literatuur: A, Hoofdstuk 14)
- D) Onjuist. 'Besturingssysteem' is een van de drie lagen waaruit gegevens moeten worden opgehaald in het monitoring-raamwerk.

18 / 40

Voor welke doelgroep heeft self-service-toegang tot telemetrie meerwaarde?

- A) Alleen voor ontwikkelaars
- B) Alleen voor ontwikkelaars en operators
- C) Alleen voor ontwikkelaars, operators en belanghebbenden
- D) Voor ontwikkelaars, operators, belanghebbenden en klanten

- A) Onjuist. Dit heeft ook meerwaarde voor klanten, belanghebbenden en operators.
- B) Onjuist. Dit heeft ook meerwaarde voor klanten, en belanghebbenden.
- C) Onjuist. Dit heeft ook meerwaarde voor klanten.
- D) Juist. Self-service-toegang tot telemetrie heeft meerwaarde voor alle bezoekers. (Literatuur: A, Hoofdstuk 14)



19 / 40

Welke techniek biedt in een omgeving met geautomatiseerde tests, snelle deployment (uitrol) processen en voldoende telemetrie de **meeste** DevOps-voordelen voor een bedrijf?

- A) Fix-forward
  - B) Roll-back (terugdraaien)
  - C) Verwijderen van defecte servers
  - D) Uitschakelen van defecte functies
- A) Juist. Hoewel dit een riskante keuze kan zijn, is dit volkomen veilig in deze omgeving met geautomatiseerde tests, snelle deploymentprocessen en voldoende telemetrie. Zo kan snel worden bevestigd of alles in productie correct werkt en kunnen snel nieuwe functies en waarde voor het bedrijf worden geleverd. De toegevoegde waarde is dat fouten worden gecorrigeerd, zonder dat nieuwe functies of defecte servers worden verwijderd. (Literatuur: A, Hoofdstuk 16)
- B) Onjuist. Net zoals bij 'Uitschakeling van defecte functies' het geval is, wordt bij het terugdraaien de deployment teruggedraaid naar de vorige staat. Problemen worden dus opgelost door de functie weg te nemen die de fout veroorzaakte. De fout wordt daarmee niet hersteld en nieuwe waarde wordt uit productie gehaald.
- C) Onjuist. Deze optie richt zich op de continuïteit van de service, maar omdat we beschikken over goede processen, telemetrie en tests, is het leveren van nieuwe bedrijfswaarde beter.
- D) Onjuist. Uitschakeling van defecte functies is de minst riskante optie, maar biedt geen oplossing voor de fout. Bovendien wordt nieuwe waarde uit productie verwijderd.

20 / 40

Een softwarebedrijf heeft de volgende vereisten voor lanceringsrichtlijnen opgesteld:

- Aantal en ernst van defecten: Werkt de applicatie conform het ontwerp?
- Dekking van de monitoring: Is de dekking van de monitoring voldoende om een service te herstellen wanneer er dingen fout gaan?
- Systeemarchitectuur: Is de service strak genoeg gekoppeld om een groot aantal incidenten in productie op te lossen?
- Type/frequentie van pager-meldingen: Genereert de applicatie een ondoenlijk aantal meldingen in productie?

Welke van deze vereisten voor lanceringsrichtlijnen voldoet **niet** aan de DevOps-werkwijze?

- A) Aantal en ernst van defecten: Werkt de applicatie conform het ontwerp?
  - B) Dekking van de monitoring: Is de dekking van de monitoring voldoende om een service te herstellen wanneer het fout gaat?
  - C) Systeemarchitectuur: Is de service strak genoeg gekoppeld om een groot aantal incidenten in productie op te lossen?
  - D) Type/frequentie van pager-meldingen: Genereert de applicatie een ondoenlijk aantal meldingen in productie?
- A) Onjuist. Deze vereiste voldoet wel aan de DevOps-werkwijze.
- B) Onjuist. Deze vereiste voldoet wel aan de DevOps-werkwijze.
- C) Juist. Binnen de DevOps-werkwijze moet de architectuur losjes gekoppeld zijn, niet strak. De service moet losjes gekoppeld zijn om een groot aantal wijzigingen en deployments (uitrol) in productie te ondersteunen. (Literatuur: A, Hoofdstuk 16)
- D) Onjuist. Deze vereiste voldoet wel aan de DevOps-werkwijze.

**21 / 40**

Wat is een uniek kenmerk van de hand-off readiness review (gereedheidsbeoordeling bij overdracht, HRR) en **niet** van de launch readiness review (gereedheidsbeoordeling bij lancering, LRR)?

- A) HRR is veel strenger en heeft hogere acceptatiestandaarden.
  - B) HRR wordt door de productteams zelf gerapporteerd.
  - C) HRR wordt officieel goedgekeurd voordat een nieuwe service algemeen beschikbaar wordt gemaakt.
  - D) HRR wordt officieel goedgekeurd voordat een nieuwe service live-productieverkeer ontvangt.
- 
- A) Juist. Dit is een uniek kenmerk van HRR, aangezien officiële goedkeuring vereist is wanneer de service overgaat naar een door Operations beheerde staat. (Literatuur: A, Hoofdstuk 16)
  - B) Onjuist. Dit is een uniek kenmerk van LRR, niet van HRR.
  - C) Onjuist. Dit is een uniek kenmerk van LRR, niet van HRR.
  - D) Onjuist. Dit is een uniek kenmerk van LRR, niet van HRR.

**22 / 40**

Een van de krachtigste technieken voor het ontwerpen van de interactie- en gebruikerservaring (user-experience, UX) is contextgebonden onderzoek.

Wat is de **beste** omschrijving van contextgebonden onderzoek?

- A) Het productteam vraagt gebruikers te reageren op een demonstratie van de applicatie, die door hen wordt gegeven.
  - B) Het productteam interviewt gebruikers over het thuisgebruik van de applicatie.
  - C) Het productteam observeert gebruikers die de applicatie in hun natuurlijke omgeving gebruiken.
  - D) Het productteam bestudeert gebruikers tijdens de gebruikersacceptatietest in een speciale ruimte met testapparatuur.
- 
- A) Onjuist. Contextgebonden onderzoek gaat over het observeren van de manier waarop gebruikers een applicatie gebruiken, niet om een demonstratie van de applicatie door de leverancier.
  - B) Onjuist. Contextgebonden onderzoek gaat over directe observatie van de manier waarop gebruikers een applicatie gebruiken. Hierbij worden die gebruikers niet geïnterviewd over hun gebruik van de applicatie.
  - C) Juist. Deze techniek gaat over het observeren van de manier waarop klanten een applicatie gebruiken in een natuurlijke omgeving. (Literatuur: A, Hoofdstuk 16)
  - D) Onjuist. Contextgebonden onderzoek gaat over het observeren van de manier waarop gebruikers een applicatie gebruiken in een natuurlijke omgeving, niet in een speciale ruimte met testapparatuur.

**23 / 40**

Start-up ABC heeft behoorlijke moeite met het ontwikkelen van functies voor een mobiele navigatie-applicatie die voldoet aan de verwachtingen van de gebruiker.

Wat is de **beste** aanpak om te zorgen dat de functies direct aansluiten bij de verwachtingen van de gebruiker over de gebruikerservaring (user-experience, UX) van de mobiele navigatie-applicatie?

- A) De UX van de mobiele navigatie-applicatie volledig ontwikkelen, zodat consumenten beschikken over meer geavanceerde functies en keuzes om door de UX te navigeren, waardoor ze meer opties hebben om de UX aan te passen aan hun voorkeuren.
  - B) Een UX van de mobiele navigatie-applicatie ontwikkelen waarin consumenten willekeurig worden geselecteerd voor de weergave van een van twee UX-versies: een controleversie (control) of een behandelde versie (treatment).
  - C) Correcties aanbrengen aan de huidige UX-functies van de mobiele navigatie-applicatie, zodat ABC kan vasthouden aan de kernfunctionaliteit van de UX zonder dat dit ten koste gaat van de kwaliteit.
  - D) Eén functie voor de mobiele navigatie-applicatie vrijgeven om gedurende een periode van drie weken feedback van de consument te krijgen over die functies.
- 
- A) Onjuist. Wanneer we ontdekken dat een bepaalde functie niet de gewenste resultaten geeft, krijgt het toevoegen van meer functies of verbeteringen aan die functies niet dezelfde prioriteit als andere nieuwe functies. Zo bereikt de ondermaats presterende functie nooit het beoogde bedrijfsdoel.
  - B) Juist. Tegenwoordig bestaat de meest gebruikte A/B-techniek voor UX uit het gebruik van een website waarop bezoekers willekeurig worden geselecteerd voor de weergave van één van twee mogelijke versies van een pagina: een controlepagina (de 'A') of een behandelde pagina (de 'B'). Door het daaropvolgende gedrag van deze twee gebruikersgroepen statistisch te analyseren, kunnen we aantonen of er een significant verschil bestaat tussen de resultaten van beide. Zo vinden we een oorzakelijk verband tussen de behandeling en het resultaat. (Literatuur: A, Hoofdstuk 17)
  - C) Onjuist. Wanneer we ontdekken dat een bepaalde functie niet de gewenste resultaten geeft, krijgt het corrigeren van die functie mogelijk niet dezelfde prioriteit als andere nieuwe functies. Zo bereikt de ondermaats presterende functie nooit het beoogde bedrijfsdoel.
  - D) Onjuist. Elk experiment is per testversie doorgaans zeer tijdrovend; de voltooiing duurt weken of zelfs maanden.

**24 / 40**

Welke aanpak kan een Development-team gebruiken om levering van de gewenste bedrijfsresultaten te ondersteunen?

- A) Zorgvuldige uitvoering van veel experimenten
  - B) Volledige regressietests
  - C) Hypothese-gedreven ontwikkeling
  - D) SDLC (Software Development Life Cycle)
- 
- A) Onjuist. Dit is geen aanpak om levering van gewenste bedrijfsresultaten te ondersteunen.
  - B) Onjuist. Dit is geen aanpak om levering van gewenste bedrijfsresultaten te ondersteunen.
  - C) Juist. Dit is de beste aanpak om levering van gewenste bedrijfsresultaten te ondersteunen. (Literatuur: A, Hoofdstuk 17)
  - D) Onjuist. Dit is geen aanpak om levering van gewenste bedrijfsresultaten te ondersteunen.

**25 / 40**

Bedrijf ABC vond het in het verleden lastig om inzicht te krijgen in de uitwerking van codewijzigingen in de payroll-applicatie. Momenteel zijn geïmplementeerde codewijzigingen slecht gedocumenteerd. De laatste codecorrectie werd bijvoorbeeld als volgt gedocumenteerd: "Probleem nr. 1801 voor payroll-applicatie herstellen". Het bedrijf merkt dat dit een slechte pull-aanvraag (vraaggestuurde aanvraag) is.

Een goede pull-aanvraag omvat de volgende aspecten:

- Waarom implementeren we de oplossing?
- Wie implementeert de oplossing?

Wat **moet** nog meer worden vermeld in een goede pull-aanvraag?

- A) De business-unit waar de oplossing van toepassing is
- B) De mogelijke risico's en tegenmaatregelen voor de implementatie van de oplossing
- C) De releaseplanning voor de implementatie van de oplossing

- A) Onjuist. De mogelijke risico's en tegenmaatregelen voor de implementatie van de oplossing moeten worden vermeld.
- B) Juist. Een effectieve pull-aanvraag moet details bevatten over de reden waarom de oplossing wordt geïmplementeerd, wie de oplossing implementeert, en de mogelijke risico's en tegenmaatregelen bij de implementatie van de oplossing. (Literatuur: A, Hoofdstuk 18)
- C) Onjuist. De mogelijke risico's en tegenmaatregelen bij de implementatie van de oplossing moeten worden vermeld.

**26 / 40**

Dit is het scenario:

"De ontwikkelaar doorloopt de code terwijl een collega feedback geeft."

Welke beoordelingstechniek wordt hier beschreven?

- A) Over-de-schouder
- B) In tweetallen programmeren (pair programming)
- C) Collegiale beoordeling (peer review)
- D) Door tools ondersteunde beoordeling

- A) Juist. De over-de-schouder-techniek draait om de interactie tussen de auteur en een collega na het schrijven van de code. (Literatuur: A, Hoofdstuk 18)
- B) Onjuist. Deze beoordelingstechniek wordt uitgevoerd tijdens het schrijven van de code, niet erna.
- C) Onjuist. Deze beoordeling wordt na afloop uitgevoerd, maar de auteur hoeft daarbij niet aanwezig te zijn.
- D) Onjuist. Deze beoordeling wordt uitgevoerd door een tool, niet door een persoon.

27 / 40

Welke beoordelingstechniek resulteert direct in een afname van het aantal codefouten?

- A) Over-de-schouder
  - B) In tweetallen programmeren (pair programming)
  - C) Collegiale beoordeling (peer review)
  - D) Door tools ondersteunde beoordeling
- 
- A) Onjuist. Hiermee kan feedback worden gegeven tijdens het doorlopen van software die al is geschreven.
  - B) Juist. In tweetallen programmeren betekent dat software samen wordt geschreven. De ene persoon typt, de andere controleert de invoer. Dit resulteert in directe feedback en minder vergissingen. (Literatuur: A, Hoofdstuk 18)
  - C) Onjuist. Collegiale beoordeling wordt uitgevoerd na het schrijven van de software.
  - D) Onjuist. Bij door tools ondersteunde beoordeling is feedback beperkt mogelijk en worden veel controles uitgevoerd nadat de code is ingecheckt.

28 / 40

Een cloud-serviceprovider wil de capaciteit van de service vergroten met behulp van een Simian Army Monkey.

Welke Monkey heeft de provider in dit geval nodig?

- A) Doctor Monkey
  - B) Janitor Monkey
  - C) Latency Monkey
- 
- A) Onjuist. De Doctor Monkey raadpleegt statuscontroles die op elke instantie worden uitgevoerd, vindt instanties met een slechte status en schakelt deze op proactieve wijze uit als eigenaren de onderliggende oorzaak niet op tijd herstellen. Hiermee wordt de capaciteit niet verbeterd.
  - B) Juist. De Janitor Monkey waarborgt dat de cloudomgeving kan draaien zonder overbodige items en verspilling (waste). Deze Monkey zoekt naar ongebruikte resources en verwijdert deze. (Literatuur: A, Bijlage 9)
  - C) Onjuist. De Latency Monkey omvat kunstmatige vertragingen of downtime op de RESTful-client/server-communicatielaag om zodoende serviceverslechtering te simuleren en te waarborgen dat afhankelijke services op gepaste wijze reageren. Hiermee wordt de capaciteit niet verbeterd.

**29 / 40**

Bedrijf ABC wil overstappen op de DevOps-werkwijze en een open, niet-verwijtende leeromgeving promoten. ABC werd onlangs getroffen door een grote applicatiestoring en wist de applicatieservice toen te herstellen.

Wat is de **eerste** taak die moet worden uitgevoerd in de verwijtloze post-mortem (bespreking achteraf)?

- A) Een tijdlijn van relevante gebeurtenissen opstellen in de volgorde dat deze zich voordeden tijdens de grote applicatiestoring
  - B) Tegenmaatregelen identificeren om te voorkomen dat de grote applicatiestoring zich in de toekomst nogmaals voordoet
  - C) De onderliggende oorzaak (root cause) van de grote applicatiestoring identificeren om een corrigerende actie te kunnen voorstellen waarmee wordt voorkomen dat de grote applicatiestoring zich nogmaals voordoet
  - D) De post-mortem publiceren op een centrale locatie waar de hele organisatie toegang tot heeft, zodat iedereen kan leren van de grote applicatiestoring
- 
- A) Juist. De eerste taak in een verwijtloze post-mortem is het vastleggen van een zo duidelijk mogelijke tijdlijn van relevante gebeurtenissen in de volgorde dat deze zich voordeden. Dit omvat alle ondernomen acties met bijbehorende tijden (bij voorkeur ondersteund door chatlogboeken, zoals IRC of Slack), de waargenomen effecten (bij voorkeur in de vorm van de specifieke metriecken uit de productie-telemetrie, niet alleen subjectieve verhalen), alle gevolgde onderzoekspaden, en de oplossingen die werden overwogen. (Literatuur: A, Hoofdstuk 19)
  - B) Onjuist. Het bepalen van tegenmaatregelen is niet de eerste taak in een verwijtloze post-mortem. Deze actie komt na het opstellen van de tijdlijn.
  - C) Onjuist. Het bepalen van de onderliggende oorzaak is niet de eerste taak in een verwijtloze post-mortem. Deze actie komt na het opstellen van de tijdlijn.
  - D) Onjuist. Het publiceren van de post-mortem is niet de eerste taak in een verwijtloze post-mortem. Deze actie komt pas nadat de post-mortem volledig is gedocumenteerd.

**30 / 40**

Wat is vereist bij het creëren van weerbaarheid door het injecteren van productiefouten?

- A) Definiëren van de foutmodus
  - B) Organisatie van post-mortems (bespreking achteraf)
  - C) Training van mensen
  - D) Gebruik van een testomgeving
- 
- A) Juist. Het definiëren van de foutmodus is belangrijk om te waarborgen dat deze foutmodussen werken volgens het ontwerp. (Literatuur: A, Hoofdstuk 19)
  - B) Onjuist. Post-mortems maken geen deel uit van de injectietechnieken.
  - C) Onjuist. Mensen leren van de injectie.
  - D) Onjuist. Er is alleen een productieomgeving nodig.

31 / 40

Wat is de **eerste** stap in een plan voor een game-dag?

- A) Oefeningen vaststellen en uitvoeren
  - B) Problemen identificeren, oplossen en testen
  - C) De uitval plannen
  - D) SPOF's (Single Points of Failure) voorbereiden en wegnemen
- 
- A) Onjuist. Het eerste aspect is plannen: 'Plan de uitval van de services door een grootschalige foutinjectie'. 'Oefeningen vaststellen en uitvoeren' is een taak voor de spelmakers.
  - B) Onjuist. Het eerste aspect is plannen: 'Plan de uitval van de services door een grootschalige foutinjectie'.
  - C) Juist. De stappen zijn als volgt:
    1. De eerste stap is het plannen van de uitval;
    2. Op basis van het plan kan het team de uitval voorbereiden door maatregelen te nemen;
    3. De maatregelen omvatten mogelijk procedures die moeten worden getest;
    4. De uitval wordt volgens planning uitgevoerd;
    5. Het gedefinieerde proces moet worden gevolgd. (Literatuur: A, Hoofdstuk 19)
  - D) Onjuist. Het eerste aspect is plannen: 'Plan de uitval van de services door een grootschalige foutinjectie'.

32 / 40

Wat is een voorbeeld van een niet-functionele vereiste (nfr)?

- A) Voor- en achterwaartse compatibiliteit tussen versies
  - B) De mogelijkheid om melding te maken van vertragingen in planningen
  - C) Registratie van financiële transacties in een boekingsstelsel van een hotel
- 
- A) Juist. Voor- en achterwaartse compatibiliteit tussen versies is een voorbeeld van een niet-functionele vereiste. (Literatuur: A, Hoofdstuk 20)
  - B) Onjuist. De melding is een functie en dus een functionele vereiste.
  - C) Onjuist. Financiële transacties zijn een functionaliteit. Niet-functionele vereisten hebben betrekking op de kwaliteit van het systeem.

**33 / 40**

Welke activiteit moet worden uitgevoerd om succesvol herbruikbare user story's voor Operations te maken?

- A) User story's voor Operations koppelen aan relevante verbeteringen en fouten in de ontwikkeling
- B) Activiteiten binnen het overdrachtsproces (handoff process) definiëren en deze activiteiten vervolgens automatiseren met toepasselijke tools en ondersteunende workflows
- C) Aanwijzen van de nodige operationele werkzaamheden en degenen die ze uitvoeren

- A) Onjuist. Dit is geen activiteit waarmee rekening moet worden gehouden om succesvol herbruikbare user story's voor Operations te maken.
- B) Juist. Dit is een activiteit waarmee rekening moet worden gehouden om succesvol herbruikbare user story's voor Operations te maken. (Literatuur: A, Hoofdstuk 20)
- C) Onjuist. Dit is geen activiteit waarmee rekening moet worden gehouden om succesvol herbruikbare user story's voor Operations te maken.

**34 / 40**

Bekijk de volgende elementen:

1. Wijzigingsaanvragen
2. Tools voor de deployment pipeline (uitrolopijplijn)
3. Gecompileerde uitvoerbare programmabestanden
4. Handleidingen en standaarden

Welke **twee** elementen worden doorgaans opgeslagen in een enkelvoudige opslagplaats voor gedeelde broncode (single shared source code repository)?

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

- A) Onjuist. Wijzigingsaanvragen (1) maken geen deel uit van een enkelvoudige opslagplaats voor gedeelde broncode, aangezien dit geen artefacten zijn waarmee kennis en leerhulpmiddelen worden gecodeerd. Wijzigingsaanvragen worden geregistreerd in de product backlog, die geen deel uitmaakt van de opslagplaats voor gedeelde broncode. Tools voor de deployment pipeline (2) maken deel uit van een enkelvoudige opslagplaats voor gedeelde broncode.
- B) Onjuist. Wijzigingsaanvragen (1) maken geen deel uit van een enkelvoudige opslagplaats voor gedeelde broncode, aangezien dit geen artefacten zijn waarmee kennis en leerhulpmiddelen worden gecodeerd. Wijzigingsaanvragen worden geregistreerd in de product backlog, die geen deel uitmaakt van de opslagplaats voor gedeelde broncode. Handleidingen en standaarden (4) maken deel uit van een enkelvoudige opslagplaats voor gedeelde broncode.
- C) Onjuist. Gecompileerde uitvoerbare programmabestanden (3) maken geen deel uit van een opslagplaats voor gedeelde broncode, aangezien dit geen artefacten zijn waarmee kennis en leerhulpmiddelen worden gecodeerd. Uitvoerbare bestanden worden binair opgeslagen in de opslagplaats voor artefacten (output van de deployment pipeline) en niet in de enkelvoudige opslagplaats voor gedeelde broncode (input van de deployment pipeline). Tools voor de deployment pipeline (2) maken deel uit van een enkelvoudige opslagplaats voor gedeelde broncode.
- D) Juist. Zowel tools voor de deployment pipeline (2) als handleidingen en standaarden (4) maken deel uit van een enkelvoudige opslagplaats voor gedeelde broncode, aangezien dit artefacten zijn waarin kennis en leermiddelen worden gecodeerd. (Literatuur: A, Hoofdstuk 20)



35 / 40

Met welk doel worden lokale ontdekkingen omgezet in globale verbeteringen?

- A) Zodat de stand van zaken niet alleen voor Dev en Ops wordt verbeterd, maar voor de hele organisatie.
- B) Zodat alle nieuwe en bestaande services gemakkelijker gebruik kunnen maken van de collectieve kennis.
- C) Zodat de werkcultuur meer een cultuur van samenwerking wordt en de systemen veiliger en weerbaarder worden.
- D) Zodat er een cultuur ontstaat waarin iedereen zich op zijn en haar gemak en verantwoordelijk voelt.

- A) Juist. Dit is het doel waarvoor lokale ontdekkingen worden omgezet in globale verbeteringen. (Literatuur: A, Hoofdstuk 20)
- B) Onjuist. Dit is een doel van het onderliggende ontwerp voor Operations door middel van gecodificeerde niet-functionele vereisten (nfr's).
- C) Onjuist. Dit is een doel van het reserveren van tijd om organisatieleren en verbeteren te stimuleren.
- D) Onjuist. Dit is een doel van het faciliteren en creëren van leermogelijkheden in dagelijkse werkzaamheden.

36 / 40

Ontwikkelaars kunnen het voor engineers gemakkelijk maken om registratie- en versleutelingsstandaarden correct toe te voegen aan en te gebruiken in hun applicaties en omgevingen.

Wat is **geen** item in een opslagplaats voor gedeelde broncode (shared source code repository) waarmee dit wordt ondersteund?

- A) Codebibliotheken en de bijbehorende aanbevolen configuraties
- B) Deployment packages (uitrolpakketten)
- C) Pakketten en builds van een besturingssysteem (OS packages en builds)
- D) Verborgen management-tools

- A) Onjuist. Aangezien codebibliotheken en de bijbehorende aanbevolen configuraties effectieve, beveiliging-specifieke configuratie-instellingen hebben voor de componenten die in de betreffende applicatie worden gebruikt, kunnen engineers gemakkelijk registratie- en versleutelingsstandaarden toevoegen en gebruiken.
- B) Juist. Deployment packages worden opgeleverd door een engineer (Operations) en niet door een ontwikkelaar (Development). Daarom is dit geen item waarmee dit wordt ondersteund. Deze vraag gaat over wat ontwikkelaars kunnen doen zodat alle engineers gemakkelijk registratie- en versleutelingsstandaarden correct toe kunnen voegen aan en gebruiken in hun applicaties. (Literatuur: A, Hoofdstuk 22)
- C) Onjuist. Pakketten en builds van een besturingssysteem maken het voor engineers gemakkelijk om registratie- en versleutelingsstandaarden correct toe te voegen aan en te gebruiken in hun applicaties, aangezien ze beide effectieve, beveiliging-specifieke configuratie-instellingen bieden voor de componenten die in de betreffende applicatie worden gebruikt.
- D) Onjuist. Verborgen management-tools maken het voor engineers gemakkelijk om registratie- en versleutelingsstandaarden correct toe te voegen aan en te gebruiken in hun applicaties, aangezien deze effectieve, beveiliging-specifieke configuratie-instellingen bieden, zoals verbindinginstellingen, versleutelingsleutels, enz.

37 / 40

Wanneer ontwikkelaars code introduceren, bestaat er altijd een risico dat hiermee onbevoegde toegang mogelijk wordt gemaakt.

Met welke maatregel wordt dat risico **niet** beperkt?

- A) Codebeoordelingen
- B) Codetests
- C) Effectieve patching
- D) Penetratietests

- A) Onjuist. Tijdens een codebeoordeling kan geïntroduceerde, schadelijke softwarecode aan het licht komen waarmee toegang via de achterdeur mogelijk is.
- B) Onjuist. Tijdens codetests kunnen mogelijkheden tot onbevoegde toegang aan het licht komen.
- C) Juist. Met effectieve patching komt de door de ontwikkelaar toegevoegde schadelijke code niet aan het licht. Er worden alleen bugs opgelost. (Literatuur: A, Hoofdstuk 22)
- D) Onjuist. De penetratietest kan zwakke plekken in een applicatie onthullen die opzettelijk zijn ontwikkeld of achtergelaten om onbevoegde toegang mogelijk te maken.

38 / 40

Wat is een voorbeeld van het creëren van telemetrie in een applicatie?

- A) Wijzigingen in een besturingssysteem
- B) Dagelijkse controle van systeemlogboeken
- C) Wijzigingen in een beveiligingsgroep
- D) Reset van gebruikerswachtwoorden

- A) Onjuist. Dit is gebruik van telemetrie voor de infrastructuuromgeving waarin de applicatie wordt gehost, niet specifiek voor de applicatie.
- B) Onjuist. Dit is gebruik van telemetrie voor de infrastructuuromgeving waarin de applicatie wordt gehost, niet specifiek voor de applicatie.
- C) Onjuist. Dit is gebruik van telemetrie voor de infrastructuuromgeving waarin de applicatie wordt gehost, niet specifiek voor de applicatie.
- D) Juist. Dit is gebruik van telemetrie specifiek voor een applicatie. (Literatuur: A, Hoofdstuk 22)

39 / 40

Welk proces is een primaire maatregel om operationele en beveiligingsgerelateerde risico's te beperken en ondersteunt ook naleving?

- A) Proces voor change management (wijzigingenbeheer)
  - B) Proces voor configuration management
  - C) Proces voor release- en deploymentmanagement (uitrolmanagement)
  - D) Proces voor service-levelmanagement
- 
- A) Juist. Bijna elke grotere IT-organisatie beschikt over processen voor change management. Dit zijn voor hen de primaire maatregelen om operationele en beveiligingsgerelateerde risico's te beperken. Nalevings- en beveiligingsmanagers vertrouwen voor hun nalevingsvereisten op de processen voor wijzigingsbeheer. Doorgaans hebben zij bewijs nodig dat alle wijzigingen op gepaste wijze zijn goedgekeurd. (Literatuur: A, Hoofdstuk 23)
  - B) Onjuist. Dit proces zorgt voor meer informatie over de services en elk configuratie-item. Deze informatie kan worden gebruikt om potentiële risico's en verbeterpunten te analyseren en detecteren, maar daarmee zorgt het proces nog niet voor een beperking van die risico's. Met configuration management worden configuratie-items gepland, geïmplementeerd, beheerd, gerapporteerd en geverifieerd.
  - C) Onjuist. Release- en deploymentmanagement is het proces waarmee een goedgekeurde release, die mogelijk diverse goedgekeurde wijzigingen bevat, daadwerkelijk wordt geïmplementeerd. Wijzigingen worden beheerd en goedgekeurd in het proces voor change management; de daaropvolgende werkzaamheden worden afgestemd op het proces voor release- en deploymentmanagement.
  - D) Onjuist. Het SLM-proces zorgt ervoor dat het met het bedrijf afgesproken serviceniveau wordt behaald. Het vermeldt de nalevings- en beveiligingsvereisten, maar is niet het proces dat de risico's beperkt.

40 / 40

Wat is een nadeel als een bedrijf kiest voor de maatregel 'scheiding van taken'?

- A) De 'scheiding van taken' maatregel staat vaak ontwikkelinspanningen in de weg, omdat de feedback die engineers op hun werk krijgen, wordt vertraagd en afneemt.
  - B) Bij een 'scheiding van taken' maatregel moet de ontwikkelaar wijzigingen indienen bij een codebeheerder, die de wijziging beoordeelt en goedkeurt voordat deze naar productie wordt doorgezet.
  - C) Een 'scheiding van taken' maatregel leidt tot onnodig werk, omdat ingecheckte code moet worden geïnspecteerd en beoordeeld, wat enkel leidt tot onnodige zekerheid over de kwaliteit van het werk.
- 
- A) Juist. Een 'scheiding van taken' maatregel staat dit inderdaad vaak in de weg, aangezien de feedback die engineers op hun werk krijgen, wordt vertraagd en afneemt. De engineers kunnen hierdoor geen volledige verantwoordelijkheid nemen voor de kwaliteit van hun werk en dit hindert het bedrijf in organisatieleren. Daarom moeten we waar mogelijk scheiding van taken als maatregel vermijden. In plaats daarvan kunnen we beter kiezen voor maatregelen als programmeren in tweetallen (pair programming), continue inspectie van ingecheckte code en codebeoordelingen. (Literatuur: A, Hoofdstuk 23)
  - B) Onjuist. Dit is de oude aanpak ter ondersteuning van de levenscyclus voor softwareontwikkeling (SDLC, Software Development Life Cycle), niet van DevOps. De aanbevolen maatregelen voor DevOps zijn onder andere in tweetallen programmeren, continue inspectie van ingecheckte code en codebeoordelingen. Deze maatregelen bieden de nodige zekerheid over de kwaliteit van het werk. Door het invoeren van deze maatregelen wanneer een scheiding van taken gevaagd wordt, kunnen we waarschijnlijk aantonen dat de eerder genoemde maatregelen voldoen.
  - C) Onjuist. We moeten scheiding van taken als maatregel zien te vermijden. In plaats daarvan kunnen we beter kiezen voor maatregelen als in tweetallen programmeren, continue inspecties van ingecheckte code en codebeoordelingen. Door het invoeren van deze maatregelen wanneer een scheiding van taken gevaagd wordt, kunnen we waarschijnlijk aantonen dat de eerdergenoemde maatregelen voldoen.

# Evaluatie

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

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

# Contact EXIN

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

