



模擬試験

2025 年 12 月版

Copyright © EXIN Holding B.V. 2025. 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.



目次

はじめに	4
解答集	5
模擬試験	15
評価	36

はじめに

これは EXIN DevOps Professional (DEVOPSP. JP) のサンプル試験です。この試験は EXIN 試験の規則および規定を適用します。

本試験は選択式の問題が 40 問で構成されます。各問題には、選択肢が複数ありますが、そのうち正解は 1 つのみです。

この試験で取得できる最大点数は 40 点です。各正解には 1 点の価値があります。試験に合格するには 26 点以上が必要です。

本試験の制限時間は 90 分です。

ご健闘をお祈りいたします。

解答集

1 / 40

アジャイル・マニフェストの原則はどれですか？

- A) 継続的でダイナミックな学習のカルチャーを創る
- B) フィードバックとフィードフォワードのループを作業システム内に創る
- C) 働くソフトウェアを2週間から2ヶ月の間隔で頻繁に提供する
- D) 作業を見える化し、バッチサイズと作業インターバルを削減し、品質を向上させることによってフロー（流れ）を増やす

2 / 40

三つの道のうち、「日々の業務の改善を制度化する」を含むのはどれですか？

- A) 継続的な学習と実験
- B) フィードバック
- C) フロー（流れ）

3 / 40

システム・オブ・エンゲージメント (SoE) とシステム・オブ・レコード (SoR) の変更に関するペース面での違いは何ですか？

- A) SoEとSoRの変更ペースは通常同じである
- B) SoEの変更ペースは通常SoRよりも大幅に早い
- C) SoEの変更ペースは通常SoRよりも大幅に遅い
- D) この関係性は情報システムによって異なる

4 / 40

開発とオペレーション（運用）に共有ツールを使わせるとどのような利点がありますか？

- A) 統合バックログによって全員がグローバルな視点から改善プロジェクトの優先順位を決められる
- B) 開発者は、開発したアプリケーションの本番環境でのパフォーマンスに対するフィードバックを得る。また、アプリケーションの不具合があれば修正できる
- C) チームは、通常の営業時間内にデプロイメントを実装することができる。また、単純なロールバックを実施できる
- D) オペレーション（運用）の知識を、自動化コードに書き込む事で、信頼性や広範な再利用性が格段に向上する

5 / 40

オペレーション（運用）の能力を開発チームにうまく統合すると、両方の効率性と生産性が高まり、マーケット志向がさらに高い結果を生み出すことができます。

これを達成するのに**最適な**アプローチはどれですか？

- A) 開発連絡係をオペレーション（運用）チームに割り当てる
- B) インフラストラクチャーのセルフサービス能力を作成する
- C) オペレーション（運用）チームを外注する
- D) 開発者がオペレーション（運用）業務を行えるようにトレーニングする

6 / 40

ある銀行では、オペレーションエンジニアが新しい環境を構築するのに遅れが生じたため、新製品または改良した製品を市場に出すまでの期間を延長する必要があります。

環境の自動構築について正しいのはどれですか？

- A) 自動構築された環境はすべての環境で使用可能である。
- B) 自動構築された環境は、セキュリティ制限のため、本番環境以外のすべての環境で使用可能である。
- C) DevOpsでは、四つの目（四つの視点）の原則に基づき、本番環境を自動で構築する場合はオペレーション（運用）によるレビューが必要である。
- D) DevOpsでは、四つの目（四つの視点）の原則に基づき、本番環境を自動で構築する場合はオペレーション（運用）によるマニュアル合意が必要である。

7 / 40

バリューマップのプロセスブロックに含めなければならない価値はどれですか？

- A) アプリケーションの遠隔計測
- B) 完全かつ正確な割合
- C) チームのベロシティ
- D) 仕掛（WIP）

8 / 40

DevOpsでビジネスに対するリスクを最小限にするためのバージョン管理の**主要な**目標はどれですか？

- A) 望ましい状態から構成が変更したときにアラートを出す能力を確保する
- B) テスト環境の以前の状態を再構築する能力を確保する
- C) 本番環境を再構築してプロセスを構築する能力を確保する
- D) 異なる開発者チーム間でソースコードを共有する能力を確保する

9 / 40

あるITチームは、DevOpsの導入を進めるために実施すべき変更をレビューするために集まりました。チームメンバーはDevOpsに適合した完成の定義 (DoD) について合意する必要があります。

DevOpsに適合するDoDはどれですか？

- A) コードは主要ブランチに統合されていて自動化ユニットテストに合格している。
- B) コードは開発者のラップトップ上で予定通りに動作していて自動化ユニットテストに合格している。
- C) コードは本番に近い環境で動作していてユーザ受け入れテストに合格している。

10 / 40

環境の構築と構成を自動化するのに**最適な**ツールはどれですか？

- A) 開発、テスト、または受け入れ環境のプロビジョニングをするためのチケットシステム
- B) 本番環境の設定を開発、テスト、受け入れ環境にコピーするツール
- C) 各環境を同期させるためにマニュアルで配布と保守がおこなわれている環境ごとの構成ファイル
- D) プログラマーが環境を自分たちで変更できるようにするインフラストラクチャ・アズ・コードの構成管理ツール

11 / 40

自動化テストセットの設計目標の一つは、テスト中にできるだけ早くエラーを検出することです。理想のテスト自動化ピラミッドにはテスト実施の正しい順番が示されています。

最初に実施しなければならないのはどのテストですか？

- A) 自動化APIテスト
- B) 自動化コンポーネントテスト
- C) 自動化結合テスト
- D) 自動化ユニットテスト

12 / 40

あるDevOpsチームはテスト駆動開発 (TDD) を使ってベロシティを高めたいと考えています。

作業の正しい順番はどれですか？

- A) 1. リファクタリング
2. テストケース記述
3. 機能コード記述
- B) 1. 機能コード記述
2. テストケース記述
3. リファクタリング
- C) 1. テストケース記述
2. 機能コード記述
3. リファクタリング

13 / 40

ある保険会社は、DevOpsプロフェッショナルを雇用して、分岐戦略の選択について社内のDevOpsチームに助言をするよう依頼しました。DevOpsプロフェッショナルは2つの戦略を評価しました。

1. 個人の生産性の最適化
2. チームの生産性の最適化

この2つの戦略についての正しい記述はどれですか？

- A) どちらの戦略も同じくらいの統合する労力を生じる
- B) 1の戦略は2よりもはるかに多くの統合する労力を生じる
- C) 2の戦略は1よりもはるかに多くの統合する労力を生じる

14 / 40

ある小売企業は開発プロセスをウォーターフォールからDevOpsへと抜本的に変更しました。短期間に多くの決定をしなければならなかったため、いくつかの技術的負債が起きています。例えば、市場に出るまでの時間は大幅に増加しました。しかし、大半の遅れを解決する方法があります。

バリューストリームマッピングによると、別々のコードブランチに置かれているコードのポーティングにスプリントタイムの20パーセントが費やされています。

このような技術的負債に対してフロー（流れ）を早くするためのソリューションはどれですか？

- A) トランクベースの開発を導入する
- B) 開発者のワークステーションでテストの失敗を再現する
- C) カナリア・リリースの実施を開始する
- D) 遠隔計測の使用を増やす

15 / 40

生産性、テスト容易性、安全性をもたらすアーキテクチャーの特性はどれですか？

- A) 緩く定義されたインターフェース
- B) 密接に結合（密結合）しているアーキテクチャー
- C) 明確に定義されたAPI

16 / 40

DevOpsで問題を解決するのに**最適な**テレメトリー（遠隔計測）手法はどれですか？

- A) 遠隔計測のツールへの投資が重要であり、本番環境、デプロイメント・パイプライン、プレ本番に重点を置く必要がある。
- B) 遠隔計測のツールへの投資が重要であり、本番環境だけに重点を置く必要がある。
- C) 遠隔計測のツールへの投資は重要ではありません。低価格の冗長なサービスを再起動することに重点を置く必要がある。
- D) 遠隔計測のツールへの投資は重要ではありません。より完全で低価格なモニタープロビジョニングを行えるアプリケーションのユーザーに重点を置く必要がある。

17 / 40

モニタリング（監視）フレームワークでは、3つのレイヤーからデータを収集する必要があります。

3つのレイヤーに**含まれない**のはどれですか？

- A) アプリケーション
- B) ビジネスロジック
- C) ビジネス測定
- D) オペレーティングシステム（OS）

18 / 40

テレメトリー（遠隔計測）へのセルフサービスアクセスの価値を享受できる対象者は誰ですか？

- A) 開発者のみ
- B) 開発者と運用者のみ
- C) 開発者と運用者と利害関係者のみ
- D) 開発者と運用者と利害関係者と顧客

19 / 40

テスト作業、迅速なデプロイメント・プロセス、十分なテレメトリー（遠隔計測）を自動化した環境において、ビジネスに**最も多くの**DevOpsによる利点を提供する技術はどれですか？

- A) フィックス・フォワード
- B) ロールバック
- C) 故障サーバーの切り離し
- D) 故障機能の遮断

20 / 40

あるソフトウェア会社は以下のローンチ要件のガイドを作成しました。

- **欠陥の数および重要度**：アプリケーションは設計通りに動作するか？
- **モニタリング範囲**：モニタリングの範囲は、問題が起きた場合にサービス復旧をするのに十分か？
- **システムアーキテクチャー**：密結合されたサービスは、本番環境での頻度の高いインシデントを解決するのに十分か？
- **呼び出しアラートの種類と頻度**：アプリケーションは本番環境で対応できない数のアラートを生成していないか？

これらのローンチ要件のガイドのうち、DevOpsの仕事の進め方に**適合しない**ものはどれですか？

- A) 欠陥の数および重要度
- B) モニタリング範囲
- C) システムアーキテクチャー
- D) 呼び出しアラートの種類と頻度

21 / 40

引き継ぎ準備調査（HRR）だけの特性であり、ローンチ準備調査（LRR）の特性ではないのはどれですか？

- A) HRRはもっと厳しく、受け入れ基準が高い。
- B) HRRはプロダクトチームが自分たちで報告する。
- C) HRRは新しいサービスが公式に利用可能になる前にサインオフされる。
- D) HRRは新しいサービスが実際の本番通信を受信するまえにサインオフされる。

22 / 40

コンテキスト調査（文脈的質問法）は、インタラクションとUX（ユーザー体験）設計で使われる最も効果的な手法の一つです。

コンテキスト調査に最も当てはまる説明はどれですか？

- A) プロダクトチームは、チームが実施するアプリケーションのデモに応じるようユーザーに依頼する
- B) プロダクトチームは、家庭でのアプリケーションの使用についてユーザーにインタビューする
- C) プロダクトチームは、アプリケーションをユーザー各自の環境で使用しているユーザーを観察する
- D) プロダクトチームは、テスト機器を備えた特別室でユーザー受け入れテストを行ってユーザーを調査する

23 / 40

ベンチャー企業のABC社は、ユーザーの期待に応えるモバイルナビゲーションアプリの機能を開発するにあたって深刻な課題に直面しています。

そのユーザー体験（UX）型モバイルナビゲーションアプリに関して機能がユーザーの期待にぴったり合っていることを確認する最適な方法はどれですか？

- A) より完全なUXのモバイルナビゲーションアプリを開発して、消費者にUXをカスタマイズするためのオプションを提供することで、消費者がUXでナビゲートするためのより高度な機能と選択肢を利用できるようにする。
- B) コントロールかトリートメントの2つのバージョンを持つUX型モバイルナビゲーションアプリを開発し、消費者を無作為に選択して2つのバージョンのいずれか一つを表示する。
- C) そのモバイルナビゲーションアプリの現行のUX機能に修正を加えてABC社が品質を犠牲にせずにUXの主要機能を保持できるようにする。
- D) そのモバイルナビゲーションアプリの機能を一つだけリリースして3週間の期限で現行の機能について消費者からのフィードバックを得る。

24 / 40

望ましいビジネス成果の達成を支援するために開発チームが使えるアプローチはどれですか？

- A) 多くの実験を慎重に実施すること
- B) 完全な回帰テスト
- C) 仮説駆動開発
- D) SDLC

25 / 40

ABC社は、自社の給与（支給）アプリケーションのコード変更の影響が予想できるある課題に直面しています。現在、実装されたコード変更は十分に文書化がされていません。例えば、最後のコード修正は「給与（支給）アプリケーションの問題1801番を修正」と記録されています。同社はこれが不適切なプルリクエストであることに気づきました。

適切なプルリクエストには以下が含まれています。

- 修正を実装する理由は何か？
- 修正を実装する方法は？

この他に適切なプルリクエストに含まれていなければならないのは次のどれですか？

- A) この修正によって影響を受けるビジネスユニット
- B) 修正実装に伴う潜在的リスクと対策
- C) 修正実装をサポートするリリースのスケジュール

26 / 40

開発者はコードのウォークスルーを行い、同僚がフィードバックを与える。

これはどのレビュー手法の説明ですか？

- A) 肩越し
- B) ペア・プログラミング
- C) ピアレビュー
- D) ツール支援のレビュー

27 / 40

コーディングエラーを減らすのに直接つながるレビュープラクティスはどれですか？

- A) 肩越し点のレビュー
- B) ペア・プログラミング
- C) ピアレビュー
- D) ツール支援のレビュー

28 / 40

あるクラウドサービスプロバイダは、サル軍団（The Simian Army）を使って自社のサービスのキャパシティを増やしたいと考えています。

この場合に必要なのはどのサルですか？

- A) ドクターモンキー
- B) 掃除モンキー
- C) 遅延モンキー

29 / 40

ABC社はDevOpsの仕事の進め方を導入しているところで、オープンで非難のない学習環境を促進したいと考えています。ABC社では最近、大規模なアプリケーション障害が発生しましたが、アプリケーションサービスを復旧することができました。

非難なきポストモータム（事後検証）ミーティングで完了しなければならない**最初**のタスクは通常何ですか？

- A) そのアプリケーション障害が発生した際の関連イベントを発生順に並べるタイムラインを作成する
- B) そのアプリケーション障害の今後の再発を防止する対策を特定する
- C) そのアプリケーション障害の根本原因を特定し、今後の再発防止のための是正処置を提案する
- D) 非難なきポストモータム（事後検証）を一か所に公開することで、全員がアクセスしてその障害から学習できるようにする

30 / 40

本番環境の失敗を投入してレジリエンスを生み出す際に必要なことは何ですか？

- A) 故障モードの定義
- B) ポストモータム（事後検証）ミーティングの編成
- C) 人員の教育
- D) テスト環境の使用

31 / 40

ゲーム・デーを実施する計画の**最初**の段階はどれですか？

- A) 訓練の定義と実施
- B) 問題の識別と対処を行いテストする
- C) 停止の計画
- D) SPOFを準備して排除する

32 / 40

非機能要件（NFR）の例はどれですか？

- A) バージョンの前後に互換性があること
- B) スケジュールの遅れを報告できること
- C) ホテル予約システムで入出金トランザクションを登録すること

33 / 40

再利用可能なオペレーション・ユーザー・ストーリーを適切に作成するために実行すべき行動はどれですか？

- A) オペレーション・ユーザー・ストーリーに関連する開発の強化と欠陥に結びつける。
- B) 手渡しプロセス中に行動を定義し、適切なツールと支援するワークフローを使ってそれらの行動を自動化する。
- C) 必要なオペレーションの業務活動とそれらの活動を完了するのに必要な人員すべてを識別する。

34 / 40

以下の要素を検討してください。

1. 変更要求
2. デプロイメント・パイプライン・ツール
3. コンパイルされた実行可能なプログラム
4. チュートリアルおよび規格

通常、共有された単一のソースコード・リポジトリに格納される2つの要素はどれとどれですか？

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

35 / 40

ローカル（局所的）な発見をグローバル（全体）な改善に変換することの目標は何ですか？

- A) 開発と運用だけではなく、組織全体のプラクティスの状態を向上させること
- B) 新規と既存のサービスすべてが集合知を活用しやすい状態にすること
- C) 仕事のカルチャーをより協同的にしてシステムの安全とレジリエンスを高めること
- D) 全員が快適で責任感を感じるカルチャーを強化すること

36 / 40

開発者は、アプリケーションと環境におけるロギングと暗号化に関する基準を、どのエンジニアでも正しく作成して使用できるように支援します。

この行動に役立つ共有ソースコード・リポジトリの項目ではないのはどれですか？

- A) コードライブラリーと推奨された構成
- B) デプロイメント・パッケージ
- C) オペレーティングシステム（OS）のパッケージおよびビルド
- D) 秘密情報の管理ツール

37 / 40

開発者がコードを導入する際に不正アクセスを可能にしてしまうリスクが常に存在します。

このリスクを軽減しないコントロールはどれですか？

- A) コードレビュー
- B) コードテスト
- C) 効果的なパッチ作業
- D) 侵入テスト

38 / 40

アプリケーションでのテレメトリ（遠隔計測）作成の例はどれですか？

- A) オペレーティングシステム（OS）の変更
- B) 日々のシステムログのレビュー
- C) セキュリティグループの変更
- D) ユーザーパスワードのリセット

39 / 40

オペレーション（運用）とセキュリティのリスクを軽減し、また、コンプライアンス要件をサポートする主要なコントロールとして作用するのはどのプロセスですか？

- A) 変更管理プロセス
- B) 構成管理プロセス
- C) リリースおよびデプロイメント管理プロセス
- D) サービスレベル管理プロセス

40 / 40

職務管理の分割を導入する際の欠点はどれですか？

- A) 職務管理の分割は、エンジニアが各自の作業に対して受け取るフィードバックの遅延と減少をもたらして開発の活動をしばしば妨げる可能性がある。
- B) 職務管理の分割は開発者に変更をコードライブラリアンに提出するよう要求し、コードライブラリアンはその変更が製品に送られる前にレビューと承認を実施する。
- C) 職務管理の分割はコードのチェックインとコードのレビューを要求して作業の品質に対する必要な保証を提供するため、不要な作業をもたらす。

模擬試験

1 / 40

アジャイル・マニフェストの原則はどれですか？

- A) 継続的でダイナミックな学習のカルチャーを創る
 - B) フィードバックとフィードフォワードのループを作業システム内に創る
 - C) 働くソフトウェアを2週間から2ヶ月の間隔で頻繁に提供する
 - D) 作業を見える化し、バッチサイズと作業インターバルを削減し、品質を向上させることによってフロー（流れ）を増やす
- A) 不正解。これは継続的な学習と実験の原則の一つです。
- B) 不正解。これはフィードバックの原則の一つです。
- C) 正解。これはアジャイル・マニフェストの主要原則の一つです。他の原則には、小人数の自発的なチームの必要性、信頼できるマネジメントモデルで働くことがあります。（参考文献：A、Introduction of Part 1）
- D) 不正解。これはフロー（流れ）の原則の一つです。

2 / 40

三つの道のうち、「日々の業務の改善を制度化する」を含むのはどれですか？

- A) 継続的な学習と実験
 - B) フィードバック
 - C) フロー（流れ）
- A) 正解。三番目の道の原則では、日々の業務の改善の制度化、ローカル（局所的）な学習を組織全体で使えるグローバル（全体）な学習に変換すること、また、日々の業務に継続的に緊張を導入することが必要です。（参考文献：A、第4章）
- B) 不正解。「日々の業務の改善を制度化する」は継続的な学習と実験の原則です。
- C) 不正解。「日々の業務の改善を制度化する」は継続的な学習と実験の原則です。

3 / 40

システム・オブ・エンゲージメント (SoE) とシステム・オブ・レコード (SoR) の変更に関するペース面での違いは何ですか？

- A) SoEとSoRの変更ペースは通常同じである
 - B) SoEの変更ペースは通常SoRよりも大幅に早い
 - C) SoEの変更ペースは通常SoRよりも大幅に遅い
 - D) この関係性は情報システムによって異なる
- A) 不正解。SoEは通常SoRよりも大幅に早い変更ペースです。
- B) 正解。SoEはユーザーインターフェースを示すため、変更ペースはSoRよりも早くなります。また、SoRの変更はSoEよりも複雑です。（参考文献：A、第5章）
- C) 不正解。SoEは通常SoRよりも大幅に早い変更ペースです。
- D) 不正解。一般的な説明が可能です。SoEは通常SoRよりも大幅に早い変更ペースです。

4 / 40

開発とオペレーション（運用）に共有ツールを使わせるとどのような利点がありますか？

- A) 統合バックログによって全員がグローバルな視点から改善プロジェクトの優先順位を決められる
 - B) 開発者は、開発したアプリケーションの本番環境でのパフォーマンスに対するフィードバックを得る。また、アプリケーションの不具合があれば修正できる
 - C) チームは、通常の営業時間内にデプロイメントを実装することができる。また、単純なロールバックを実施できる
 - D) オペレーション（運用）の知識を、自動化コードに書き込む事で、信頼性や広範な再利用性が格段に向上する
- A) 正解。統合バックログがあれば、組織にとって最も価値のある業務、または技術的負債の削減に最も効果のある作業を選択して、全員がグローバルな視点から改善プロジェクトの優先順位を決めることができます。（参考文献：A、第6章）
- B) 不正解。これは開発がオペレーション（運用）と呼び出し当番業務を行う時に共有する利点です。
- C) 不正解。これは低リスクのリリースを自動化して可能にすることの利点です。
- D) 不正解。これはサービスチームにオペレーションエンジニアを組み込むことの利点です。

5 / 40

オペレーション（運用）の能力を開発チームにうまく統合すると、両方の効率性と生産性が高まり、マーケット志向がさらに高い結果を生み出すことができます。

これを達成するのに**最適な**アプローチはどれですか？

- A) 開発連絡係をオペレーション（運用）チームに割り当てる
 - B) インフラストラクチャーのセルフサービス能力を作成する
 - C) オペレーション（運用）チームを外注する
 - D) 開発者がオペレーション（運用）業務を行えるようにトレーニングする
- A) 不正解。逆でなければなりません。「オペレーション（運用）連絡係を開発チームに割り当てる」のが正解です。
- B) 正解。これはオペレーション（運用）を開発に統合するための三大戦略の一つです。三大戦略は以下の3つです。
- 1) サービスチーム内の開発者が生産性を高められるようにセルフサービス能力を作成する。
 - 2) サービスチームにオペレーションエンジニアを組み込む。
 - 3) オペレーション（運用）の組み込みができない場合、オペレーション（運用）連絡係をサービスチームに割り当てる。（参考文献：A、第8章）
- C) 不正解。外注は戦略として推奨されていません。外注先の人員は開発チームからさらに遠くなります。
- D) 不正解。開発者はインフラストラクチャをコードとして（インフラストラクチャ・アズ・コード）使用できますが、それによってオペレーション（運用）機能が使用されなくなるわけではありません。

6 / 40

ある銀行では、オペレーションエンジニアが新しい環境を構築するのに遅れが生じたため、新製品または改良した製品を市場に出すまでの期間を延長する必要があります。

環境の自動構築について正しいのはどれですか？

- A) 自動構築された環境はすべての環境で使用可能である。
 - B) 自動構築された環境は、セキュリティ制限のため、本番環境以外のすべての環境で使用可能である。
 - C) DevOpsでは、四つの目（四つの視点）の原則に基づき、本番環境を自動で構築する場合はオペレーション（運用）によるレビューが必要である。
 - D) DevOpsでは、四つの目（四つの視点）の原則に基づき、本番環境を自動で構築する場合はオペレーション（運用）によるマニュアル合意が必要である。
- A) 正解。継続的デプロイメントでは、スクリプトを使って実行するマニュアル動作は不要です。スクリプトは自動化されなければなりません。これはすべての環境で有効です。（参考文献：A、第9章）
- B) 不正解。本番環境の構築と保守は完全に自動的に行わなければなりません。
- C) 不正解。自動化されたスクリプトはレビューの必要はありません。いずれにせよ、自動化プロセスは監視されます。
- D) 不正解。スクリプトおよびスクリプトのデプロイメントツールの受け入れは変更管理のもとにあるため、マニュアルの四つの目（四つの視点）の原則は必要ありません。

7 / 40

バリューマップのプロセスブロックに含めなければならない価値はどれですか？

- A) アプリケーションの遠隔計測
- B) 完全かつ正確な割合
- C) チームのベロシティ
- D) 仕掛（WIP）

- A) 不正解。アプリケーションの遠隔計測は、アプリケーションの動作中の挙動を監視してヘルスステータスを報告するのに使われ、迅速で正確なフィードバックとエラー検出を行います。
- B) 正解。完全かつ正確な割合は、そのビジネスに必要なバリューストリームによって提供される品質を理解するための主要な指標に含まれるため、バリューマップのプロセスブロックに含めなければなりません。（参考文献：A、第6章）
- C) 不正解。チームのベロシティは、スプリント中に測定され、スプリントミーティングで使われる価値で、そのスプリント中に完了可能な作業量を評価します。
- D) 不正解。WIP（仕掛）はカンバンボードから得られる、またはカンバンボードに含める必要がある概念で、バリューマップ（VSM）の一部ではありません。バリューストリーム分析に使う価値を提供するものでもありません。

8 / 40

DevOpsでビジネスに対するリスクを最小限にするためのバージョン管理の**主要な**目標はどれですか？

- A) 望ましい状態から構成が変更したときにアラートを出す能力を確保する
- B) テスト環境の以前の状態を再構築する能力を確保する
- C) 本番環境を再構築してプロセスを構築する能力を確保する
- D) 異なる開発者チーム間でソースコードを共有する能力を確保する

- A) 不正解。バージョン管理は、他のツールがバージョン管理情報を使って望ましい状態からの構成ドリフトを検出するのに重要ですが、バージョン管理自体の目標ではないため、この問題では適切な回答ではありません。バージョン管理には、環境を迅速に再構築してサービスやフロー（流れ）を維持する目的があります。
- B) 不正解。これはバージョン管理の利点となることもありますが、DevOpsでのバージョン管理の主要な目標ではありません。また、通常、テスト環境はテストごとに再構築されるため、以前の状態を再構築するのはそれほど有益ではありません。
- C) 正解。DevOpsでのバージョン管理は、本番環境の再構築とプロセスの構築を可能にしなければなりません。この手法を導入する組織は導入していない組織よりもパフォーマンスが向上します。（参考文献：A、第9章）
- D) 不正解。これは開発視点だけで使う場合のバージョン管理の目的ですが、DevOpsでの主要な目標ではありません。DevOpsでのバージョン管理では、すべての環境でパフォーマンスやフロー（流れ）、チーム間のコミュニケーション、予測性を達成してムダを削減するために、すべての領域の成果物、ツール、プロセスをその領域内に保存することが必要です。

9 / 40

あるITチームは、DevOpsの導入を進めるために実施すべき変更をレビューするために集まりました。チームメンバーはDevOpsに適合した完成の定義 (DoD) について合意する必要があります。

DevOpsに適合するDoDはどれですか？

- A) コードは主要ブランチに統合されていて自動化ユニットテストに合格している。
 - B) コードは開発者のラップトップ上で予定通りに動作していて自動化ユニットテストに合格している。
 - C) コードは本番に近い環境で動作していてユーザ受け入れテストに合格している。
- A) 不正解。これは開発者視点の標準的なDoDであり、ビジネスに価値を提供するDevOpsの目標は達成しません。DoDには、コードをメインブランチに統合することではなく、動くコードを本番に近い環境に出荷することを含める必要があります。
- B) 不正解。開発者のラップトップですべてが動作していても、本番に近い環境で動作する保証はありません。
- C) 正解。これはコードが予定通りに動作し、出荷できる可能性があることを保証します。（参考文献：A、第9章）

10 / 40

環境の構築と構成を自動化するのに**最適な**ツールはどれですか？

- A) 開発、テスト、または受け入れ環境のプロビジョニングをするためのチケットシステム
 - B) 本番環境の設定を開発、テスト、受け入れ環境にコピーするツール
 - C) 各環境を同期させるためにマニュアルで配布と保守がおこなわれている環境ごとの構成ファイル
 - D) プログラマが環境を自分たちで変更できるようにするインフラストラクチャ・アズ・コードの構成管理ツール
- A) 不正解。環境のプロビジョニングを行うチケットシステムはデプロイメント・パイプラインのフロー（流れ）を妨げ、手渡しを増やすため、ムダが生じます。環境のプロビジョニングを自動化するのにツールを使う必要があります。
- B) 不正解。これはDevOpsが提案する作業方法ではありません。環境はゼロから構築する必要があります。また、本番環境のデータはD-T-A 環境で使うことはできません。
- C) 不正解。環境の構成ファイルは自動で配信されなければなりません。
- D) 正解。これは環境の構築と構成を自動化する選択肢の一つです。開発者は環境の構築と構成を自分たちで行うためにインフラストラクチャ・アズ・コードの構成管理ツールを使うことができます。（参考文献：A、第9章）

11 / 40

自動化テストセットの設計目標の一つは、テスト中にできるだけ早くエラーを検出することです。理想のテスト自動化ピラミッドにはテスト実施の正しい順番が示されています。

最初に実施しなければならないのはどのテストですか？

- A) 自動化APIテスト
 - B) 自動化コンポーネントテスト
 - C) 自動化結合テスト
 - D) 自動化ユニットテスト
- A) 不正解。自動化ユニットテストを最初に実施しなければなりません。自動化APIテストは5つのうち4番目に実施します。
- B) 不正解。自動化ユニットテストを最初に実施しなければなりません。自動化コンポーネントテストは5つのうち2番目に実施します。
- C) 不正解。自動化ユニットテストを最初に実施しなければなりません。自動化結合テストは5つのうち3番目に実施します。
- D) 正解。自動化ユニットテストは最初に実施するテストです。（参考文献：A、第10章）

12 / 40

あるDevOpsチームはテスト駆動開発（TDD）を使ってベロシティを高めたいと考えています。

作業の正しい順番はどれですか？

- A) 1. リファクタリング
2. テストケース記述
3. 機能コード記述
 - B) 1. 機能コード記述
2. テストケース記述
3. リファクタリング
 - C) 1. テストケース記述
2. 機能コード記述
3. リファクタリング
- A) 不正解。最初にテストをしてからコーディングを行います。リファクタリングは3番目のステップです。
- B) 不正解。最初にテストをしてからコーディングを行います。リファクタリングは3番目のステップです。
- C) 正解。最初にテストをしてからコーディングを行います。リファクタリングは3番目のステップです。（参考文献：A、第10章）

13 / 40

ある保険会社は、DevOpsプロフェッショナルを雇用して、分岐戦略の選択について社内のDevOpsチームに助言をするよう依頼しました。DevOpsプロフェッショナルは2つの戦略を評価しました。

1. 個人の生産性の最適化
2. チームの生産性の最適化

この2つの戦略についての正しい記述はどれですか？

- A) どちらの戦略も同じくらいの統合する労力を生じる
 - B) 1の戦略は2よりもはるかに多くの統合する労力を生じる
 - C) 2の戦略は1よりもはるかに多くの統合する労力を生じる
- A) 不正解。1の戦略は人単位のブランチをもたらすため、統合する作業は悪夢のようなものです。どのバージョン管理システムを使うにしても、2の戦略がはるかに優れています。
- B) 正解。個人の生産性とはDevOps開発者ごとのブランチを意味します。このブランチごとに開発者は個別にソフトウェアを作成することができ、統合の必要はありません。2の戦略ではブランチをまったく持たなくてよいので、開発は途切れることなく行うことができます。（参考文献：A、第11章）
- C) 不正解。2の戦略では、全員が共通の領域で作業するため、統合する労力は少なくなります。

14 / 40

ある小売企業は開発プロセスをウォーターフォールからDevOpsへと抜本的に変更しました。短期間に多くの決定をしなければならなかったため、いくつかの技術的負債が起きています。例えば、市場に出るまでの時間は大幅に増加しました。しかし、大半の遅れを解決する方法があります。

バリューストリームマッピングによると、別々のコードブランチに置かれているコードのポーティングにスプリントタイムの20パーセントが費やされています。

このような技術的負債に対してフロー（流れ）を早くするためのソリューションはどれですか？

- A) トランクベースの開発を導入する
 - B) 開発者のワークステーションでテストの失敗を再現する
 - C) カナリア・リリースの実施を開始する
 - D) 遠隔計測の使用を増やす
- A) 正解。トランクベースの開発ではブランチ作成が行われなため、統合作業はなくなります。（参考文献：A、第11章）
- B) 不正解。問題なのは、統合作業の際に問題を起こすブランチ作成です。テスト作業は遅れを解決しません。
- C) 不正解。カナリア・リリースが継続的デプロイメントの視点に良い影響を与えたとしても、テスト作業は遅れを解決しません。
- D) 不正解。遠隔計測は統合作業の問題を解決しません。

15 / 40

生産性、テスト容易性、安全性をもたらすアーキテクチャの特性はどれですか？

- A) 緩く定義されたインターフェース
 - B) 密接に結合（密結合）しているアーキテクチャ
 - C) 明確に定義されたAPI
- A) 不正解。明確に定義されたインターフェースは、モジュールの相互接続方法を強制することによる生産性、テスト容易性、安全性を実現するアーキテクチャの特性です。緩く定義された（loosely-defined）インターフェースではなくて、疎結合（loosely-coupled）アーキテクチャという用語があります。
- B) 不正解。緊密に結合しているアーキテクチャは、モノリシックであったり、モジュールが相互に連携しすぎたりしています。トランクにコードをコミットするたびに、グローバルな問題が発生するリスクがあります。小さな変更を行うたびに、何日もかけて膨大な量のコミュニケーションと調整を行い、影響を受ける可能性のあるすべてのグループから承認を受ける必要があります。
- C) 正解。明確に定義されたAPI（アプリケーションプログラミングインターフェース）によってアーキテクチャは生産性、サービスのテスト容易性、安全性を提供します。（参考文献：A、第13章）

16 / 40

DevOpsで問題を解決するのに**最適な**テレメトリー（遠隔計測）手法はどれですか？

- A) 遠隔計測のツールへの投資が重要であり、本番環境、デプロイメント・パイプライン、プレ本番に重点を置く必要がある。
 - B) 遠隔計測のツールへの投資が重要であり、本番環境だけに重点を置く必要がある。
 - C) 遠隔計測のツールへの投資は重要ではありません。低価格の冗長なサービスを再起動することに重点を置く必要がある。
 - D) 遠隔計測のツールへの投資は重要ではありません。より完全で低価格なモニタープロビジョニングを行えるアプリケーションのユーザーに重点を置く必要がある。
- A) 正解。問題は本番環境だけではなく他の場所でも発生します。DevOpsは迅速なフィードバックを必要とします。フィードバックはそのパイプラインの開始時点で始めます。（参考文献：A、第14章）
- B) 不正解。本番環境だけに重点をおくのでは不十分です。エラーを早期に見つけるためにデプロイメント・パイプラインとプレ本番も重要です。
- C) 不正解。再起動は高くつきます。エラーを見つけるために多くの生産時間が無駄になります。
- D) 不正解。もしそのユーザーがそのサービスの唯一のモニターであれば、モニタープロビジョニングの方がはるかに早く失敗の箇所と対象を特定することができ（正しく実装されていれば）、さらにそのユーザーはすべてのエラーは発見できないため、多くの生産時間が失われます。そのユーザーが見たエラーのみが検出され、バックグラウンドで起きているエラーは検出されません。

17 / 40

モニタリング（監視）フレームワークでは、3つのレイヤーからデータを収集する必要があります。

3つのレイヤーに**含まれない**のはどれですか？

- A) アプリケーション
 - B) ビジネスロジック
 - C) ビジネス測定
 - D) オペレーティングシステム（OS）
- A) 不正解。アプリケーションは、モニタリング（監視）フレームワークにおいてデータを収集する3つのレイヤーの一つです。
- B) 不正解。ビジネスロジックは、モニタリング（監視）フレームワークにおいてデータを収集する3つのレイヤーの一つです。
- C) 正解。ビジネス測定は、モニタリング（監視）フレームワークにおいてデータを収集する3つのレイヤーの一つではありません。ビジネス測定はどちらかといえばモニタリングの結果です。（参考文献：A、第14章）
- D) 不正解。オペレーティングシステム（OS）は、モニタリング（監視）フレームワークにおいてデータを収集する3つのレイヤーの一つです。

18 / 40

テレメトリー（遠隔計測）へのセルフサービスアクセスの価値を享受できる対象者は誰ですか？

- A) 開発者のみ
 - B) 開発者と運用者のみ
 - C) 開発者と運用者と利害関係者のみ
 - D) 開発者と運用者と利害関係者と顧客
- A) 不正解。顧客と利害関係者と運用者にも価値を与えます。
- B) 不正解。顧客と利害関係者にも価値を与えます。
- C) 不正解。顧客にも価値を与えます。
- D) 正解。遠隔計測へのセルフサービスのアクセスはビジターすべてに価値を与えます。（参考文献：A、第14章）

19 / 40

テスト作業、迅速なデプロイメント・プロセス、十分なテレメトリー（遠隔計測）を自動化した環境において、ビジネスに**最も多くの**DevOpsによる利点を提供する技術はどれですか？

- A) フィックス・フォワード
- B) ロールバック
- C) 故障サーバーの切り離し
- D) 故障機能の遮断

- A) 正解。リスクのある選択になる場合もありますが、テスト作業、迅速なデプロイメント・プロセス、十分な遠隔計測を自動化したこのような環境においては完全に安全です。すべてが本番に向けて正しく機能しているか、ビジネスに新しい機能と価値を提供しているかどうかを迅速に確認することができます。付加された価値は、新しい機能を削除したり故障したサーバーを切り離したりするのではなく、エラーが修正されるということです。（参考文献：A、第16章）
- B) 不正解。「故障機能の遮断」と同じように、「ロールバック」はエラーを起こした機能を排除することで問題を解決し、デプロイメントを以前の状態にロールバック（巻き戻し）します。そのため、エラーの修正と生産からの新しい価値取得は行いません。
- C) 不正解。この選択肢はサービスの継続性を重視していますが、適切なプロセス、遠隔計測、テスト作業が利用できることを考えると、新しいビジネス価値を提供する方が優れた選択肢です。
- D) 不正解。故障機能を遮断するのは最もリスクの低い選択肢ですが、エラーの修正と本番からの新しい価値取得は行いません。

20 / 40

あるソフトウェア会社は以下のローンチ要件のガイドを作成しました。

- 欠陥の数および重要度：アプリケーションは設計通りに動作するか？
- モニタリング範囲：モニタリングの範囲は、問題が起きた場合にサービス復旧をするのに十分か？
- システムアーキテクチャー：密結合されたサービスは、本番環境での頻度の高いインシデントを解決するのに十分か？
- 呼び出しアラートの種類と頻度：アプリケーションは本番環境で対応できない数のアラートを生成していないか？

これらのローンチ要件のガイドのうち、DevOpsの仕事の進め方に**適合しない**ものはどれですか？

- A) 欠陥の数および重要度
- B) モニタリング範囲
- C) システムアーキテクチャー
- D) 呼び出しアラートの種類と頻度

- A) 不正解。この要件はDevOpsの仕事の進め方に適合しています。
- B) 不正解。この要件はDevOpsの仕事の進め方に適合しています。
- C) 正解。DevOpsの仕事の進め方では、アーキテクチャーの結合は密接ではなく緩やかである必要があります。本番環境での頻繁な変更とデプロイメントをサポートするため、サービスは緩く結合していなければなりません。（参考文献：A、第16章）
- D) 不正解。この要件はDevOpsの仕事の進め方に適合しています。

21 / 40

引き継ぎ準備調査（HRR）だけの特性であり、ローンチ準備調査（LRR）の特性ではないのはどれですか？

- A) HRRはもっと厳しく、受け入れ基準が高い。
 - B) HRRはプロダクトチームが自分たちで報告する。
 - C) HRRは新しいサービスが公式に利用可能になる前にサインオフされる。
 - D) HRRは新しいサービスが実際の本番通信を受信するまえにサインオフされる。
- A) 正解。これはHRRだけの特性です。HRRは、サービスが運用が管理する状態に移行した時点で受け渡しされます。（参考文献：A、第16章）
- B) 不正解。これはLRRだけの特性であり、HRRの特性ではありません。
- C) 不正解。これはLRRだけの特性であり、HRRの特性ではありません。
- D) 不正解。これはLRRだけの特性であり、HRRの特性ではありません。

22 / 40

コンテキスト調査（文脈的質問法）は、インタラクションとUX（ユーザー体験）設計で使われる最も効果的な手法の一つです。

コンテキスト調査に最も当てはまる説明はどれですか？

- A) プロダクトチームは、チームが実施するアプリケーションのデモに応じるようユーザーに依頼する
 - B) プロダクトチームは、家庭でのアプリケーションの使用についてユーザーにインタビューする
 - C) プロダクトチームは、アプリケーションをユーザー各自の環境で使用しているユーザーを観察する
 - D) プロダクトチームは、テスト機器を備えた特別室でユーザー受け入れテストを行ってユーザーを調査する
- A) 不正解。コンテキスト調査はユーザーがアプリケーションをどのように使うかを観察することであり、サプライヤが実施するアプリケーションのデモに関わるものではありません。
- B) 不正解。コンテキスト調査はユーザーがアプリケーションをどのように使うかを観察することであり、アプリケーションの使用についてのインタビューは含まれません。
- C) 正解。この手法は、顧客が各自の環境でどのようにアプリケーションを使用しているかを観察するものです。（参考文献：A、第16章）
- D) 不正解。コンテキスト調査はユーザーがアプリケーションをどのように使うかを観察することであり、テスト機器を備えた特別室では行いません。

23 / 40

ベンチャー企業のABC社は、ユーザーの期待に応えるモバイルナビゲーションアプリの機能を開発するにあたって深刻な課題に直面しています。

そのユーザー体験（UX）型モバイルナビゲーションアプリに関して機能がユーザーの期待にぴったり合っていることを確認する最適な方法はどれですか？

- A) より完全なUXのモバイルナビゲーションアプリを開発して、消費者にUXをカスタマイズするためのオプションを提供することで、消費者がUXでナビゲートするためのより高度な機能と選択肢を利用できるようにする。
 - B) コントロールかトリートメントの2つのバージョンを持つUX型モバイルナビゲーションアプリを開発し、消費者を無作為に選択して2つのバージョンのいずれか一つを表示する。
 - C) そのモバイルナビゲーションアプリの現行のUX機能に修正を加えてABC社が品質を犠牲にせずにUXの主要機能を保持できるようにする。
 - D) そのモバイルナビゲーションアプリの機能の一つだけリリースして3週間の期限で現行の機能について消費者からのフィードバックを得る。
- A) 不正解。提供された機能が望ましい結果を達成していない時にその機能に追加や強化を行うと、他の新しい機能に優先順位を低くされる可能性があり、そのパフォーマンスの低い機能は当初のビジネスゴールに決してたどりつくことはありません。
- B) 正解。最新のUXプラクティスで最もよく使われているA/Bテスト手法には、ウェブサイトのビジターがランダムに選択され、コントロール（A）かトリートメント（B）の2バージョンから一つを提示されるしゅみのウェブサイトがあります。ユーザー集団2つのその後の挙動を統計的に分析して、2つのバージョンの結果に有意な違いがあるかどうかを説明し、そのトリートメントと結果に対して因果関係を確認します。（参考文献：A、第17章）
- C) 不正解。提供された機能が望ましい結果を達成していない時にその機能に修正を加えると、他の新しい機能に優先順位を低くされる可能性があり、そのパフォーマンスの低い機能は当初のビジネスゴールに決してたどりつくことはありません。
- D) 不正解。通常、どの実験もトライアルのたびに非常に時間がかかるものであり、完了するまでに数週間、時には数か月を要します。

24 / 40

望ましいビジネス成果の達成を支援するために開発チームが使えるアプローチはどれですか？

- A) 多くの実験を慎重に実施すること
 - B) 完全な回帰テスト
 - C) 仮説駆動開発
 - D) SDLC
- A) 不正解。これは望ましいビジネス成果の達成を支援するアプローチではありません。
- B) 不正解。これは望ましいビジネス成果の達成を支援するアプローチではありません。
- C) 正解。これは望ましいビジネス成果の達成を支援する最適なアプローチです。（参考文献：A、第17章）
- D) 不正解。これは望ましいビジネス成果の達成を支援するアプローチではありません。

25 / 40

ABC社は、自社の給与（支給）アプリケーションのコード変更の影響が予想できるある課題に直面しています。現在、実装されたコード変更は十分に文書化がされていません。例えば、最後のコード修正は「給与（支給）アプリケーションの問題1801番を修正」と記録されています。同社はこれが不適切なプルリクエストであることに気づきました。

適切なプルリクエストには以下が含まれています。

- 修正を実装する理由は何か？
- 修正を実装する方法は？

この他に適切なプルリクエストに含まれていないなければならないのは次のどれですか？

- A) この修正によって影響を受けるビジネスユニット
 - B) 修正実装に伴う潜在的リスクと対策
 - C) 修正実装をサポートするリリースのスケジュール
- A) 不正解。修正実装に伴う潜在的リスクと対策が含まれていなければなりません。
- B) 正解。効果的なプルリクエストには、修正を実装する理由は何か、修正を実装するのは誰か、そして、その修正の実装に伴う潜在的リスクと対策が含まれていなければなりません。（参考文献：A、第18章）
- C) 不正解。修正実装に伴う潜在的リスクと対策が含まれていなければなりません。

26 / 40

開発者はコードのウォークスルーを行い、同僚がフィードバックを与える。

これはどのレビュー手法の説明ですか？

- A) 肩越し
 - B) ペア・プログラミング
 - C) ピアレビュー
 - D) ツール支援のレビュー
- A) 正解。肩越しの手法は、コードを記述したあとに作成者と同僚の間で行われるインタラクションです。（参考文献：A、第18章）
- B) 不正解。このレビュー手法はコーディング後ではなくコーディング中に実施されるものです。
- C) 不正解。これはコーディング後に実施されますが、作成者が同席する必要はありません。
- D) 不正解。このレビューは人ではなくツールによって実施されるものです。

27 / 40

コーディングエラーを減らすのに直接つながるレビュープラクティスはどれですか？

- A) 肩越し点のレビュー
- B) ペア・プログラミング
- C) ピアレビュー
- D) ツール支援のレビュー

- A) 不正解。肩越しのレビューは、すでに記述されたソフトウェアを読む際にフィードバックを与えません。
- B) 正解。ペア・プログラミングとは一緒にソフトウェアを記述することです。一人が入力し、もう一人がチェックします。フィードバックを直接与えてミスが減らすのにつながります。（参考文献：A、第18章）
- C) 不正解。ピアレビューはソフトウェアを記述した後に実施します。
- D) 不正解。ツール支援のレビューはフィードバックが限られており、コードのチェック後に多数のチェックが実施されます。

28 / 40

あるクラウドサービスプロバイダは、サル軍団（The Simian Army）を使って自社のサービスのキャパシティを増やしたいと考えています。

この場合に必要なのはどのサルですか？

- A) ドクターモンキー
- B) 掃除モンキー
- C) 遅延モンキー

- A) 不正解。ドクターモンキーは各インスタンス上で動作するヘルスチェックを利用して状態の悪いインスタンスを検出し、オーナーが根本原因を解消するのに間に合わないとそのインスタンスを積極的に閉じていきます。キャパシティを改善するものではありません。
- B) 正解。掃除モンキーは、使われていないリソースを探して捨てることにより、クラウド環境にごみやムダがない状態で動作していることを確実にします。（参考文献：A, Appendix 9）
- C) 不正解。遅延モンキーは、RESTful なクライアントとサーバー間のコミュニケーションレイヤーに人為的な遅延やダウンタイムをもたらしてサービス低下をシミュレートし、従属するサービスが適切に応答することを確実にします。キャパシティを改善するものではありません。

29 / 40

ABC社はDevOpsの仕事の進め方を導入しているところで、オープンで非難のない学習環境を促進したいと考えています。ABC社では最近、大規模なアプリケーション障害が発生しましたが、アプリケーションサービスを復旧することができました。

非難なきポストモーテム（事後検証）ミーティングで完了しなければならない**最初の**タスクは通常何ですか？

- A) そのアプリケーション障害が発生した際の関連イベントを発生順に並べるタイムラインを作成する
 - B) そのアプリケーション障害の今後の再発を防止する対策を特定する
 - C) そのアプリケーション障害の根本原因を特定し、今後の再発防止のための是正処置を提案する
 - D) 非難なきポストモーテム（事後検証）を一か所に公開することで、全員がアクセスしてその障害から学習できるようにする
- A) 正解。非難なき ポストモーテム（事後検証）ミーティングの最初のタスクは、関連イベントを発生順にタイムラインを可能な限り理解できるように記録することです。これは、すべての実施した活動、いつ（チャットログによる証拠があることが望ましい）、どのような事象（単に主観的な説明ではなく本番環境の遠隔計測による特定の計測値であることが望ましい）が観測されたか、たどった調査経路すべて、どのような解決策が検討されたかを含める必要があります。（参考文献：A、第19章）
- B) 不正解。対策を特定することは非難なき ポストモーテム（事後検証）ミーティングの最初のタスクではありません。これは事象のタイムライン作成後に行う活動です。
- C) 不正解。根本原因を特定することは、必ずしも非難なきポストモーテム（事後検証）ミーティングで行うべきタスクの一つではなく、決して最初に行うべきことでもありません。
- D) 不正解。ポストモーテム（事後検証）を公開することは、非難なきポストモーテムミーティングの最初のタスクではありません。これはポストモーテムが完全に文書化された後にのみ行う可能性のある活動です。

30 / 40

本番環境の失敗を投入してレジリエンスを生み出す際に必要なことは何ですか？

- A) 故障モードの定義
 - B) ポストモーテム（事後検証）ミーティングの編成
 - C) 人員の教育
 - D) テスト環境の使用
- A) 正解。故障モードが設計通りに動作することを確保するため、故障モードの定義は重要です。（参考文献：A、第19章）
- B) 不正解。ポストモーテム（事後検証）ミーティングは投入手法には含まれません。
- C) 不正解。人員は失敗の投入から学習します。
- D) 不正解。必要なのは本番環境のみです。

31 / 40

ゲーム・デーを実施する計画の**最初**の段階はどれですか？

- A) 訓練の定義と実施
 - B) 問題の識別と対処を行いテストする
 - C) 停止の計画
 - D) SP0Fを準備して排除する
- A) 不正解。計画の最初の局面は「大規模な故障の投入によるプラント停止を計画する」です。「計画を定義して訓練を実施する」がゲームメイカーのタスクです。
- B) 不正解。計画の最初の局面は「大規模な故障の投入によるプラント停止を計画する」です。
- C) 正解。正しい段階は以下の通りです。1. 停止を計画する 2. 計画に従った手段を使ってチームは停止を準備する 3. その手段にはテストの必要がある手順を含めてもよい4. 計画通りに停止を実施する5. 定義されたプロセスに従わなければならない。（参考文献：A、第19章）
- D) 不正解。計画の最初の局面は「大規模な故障の投入によるプラント停止を計画する」です。

32 / 40

非機能要件（NFR）の例はどれですか？

- A) バージョンの前後に互換性があること
 - B) スケジュールの遅れを報告できること
 - C) ホテル予約システムで入出金トランザクションを登録すること
- A) 正解。バージョンの前後に互換性があることはNFRの例です。（参考文献：A、第20章）
- B) 不正解。報告は機能であるためNFRではありません。
- C) 不正解。入出金トランザクションは機能です。NFRsはシステムの品質に関連します。

33 / 40

再利用可能なオペレーション・ユーザー・ストーリーを適切に作成するために実行すべき行動はどれですか？

- A) オペレーション・ユーザー・ストーリーに関連する開発の強化と欠陥に結びつける。
 - B) 手渡しプロセス中に行動を定義し、適切なツールと支援するワークフローを使ってそれらの行動を自動化する。
 - C) 必要なオペレーションの業務活動とそれらの活動を完了するのに必要な人員すべてを識別する。
- A) 不正解。これは再利用可能なオペレーション・ユーザー・ストーリーを適切に作成するために考慮すべき行動ではありません。
- B) 正解。これは再利用可能なオペレーション・ユーザー・ストーリーを適切に作成するために考慮すべき行動です。（参考文献：A、第20章）
- C) 不正解。これは再利用可能なオペレーション・ユーザー・ストーリーを適切に作成するために考慮すべき行動ではありません。

34 / 40

以下の要素を検討してください。

1. 変更要求
2. デプロイメント・パイプライン・ツール
3. コンパイルされた実行可能なプログラム
4. チュートリアルおよび規格

通常、共有された単一のソースコード・リポジトリに格納される2つの要素はどれとどれですか？

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

- A) 不正解。変更要求（1）は知識と学習を変換する成果物ではないため、共有された単一のソースコード・リポジトリには含まれません。変更要求は、共有された単一のソースコード・リポジトリには含まれないプロダクト・バックログに記録されます。デプロイメント・パイプライン・ツール（2）は共有された単一のソースコード・リポジトリに含まれます。
- B) 不正解。変更要求（1）は知識と学習を変換する成果物ではないため、共有された単一のソースコード・リポジトリには含まれません。変更要求は、共有された単一のソースコード・リポジトリには含まれないプロダクト・バックログに記録されます。チュートリアルおよび規格（4）は共有された単一のソースコード・リポジトリに含まれます。
- C) 不正解。コンパイルされた実行可能なプログラム（3）は知識と学習を変換する成果物ではないため、共有された単一のソースコード・リポジトリには含まれません。実行可能なプログラムは、共有された単一のソースコード・リポジトリ（デプロイメント・パイプラインのインプット）ではなく、アーチファクトデポジトリ（デプロイメント・パイプラインのアウトプット）にバイナリで格納されます。デプロイメント・パイプライン・ツール（2）は共有された単一のソースコード・リポジトリに含まれます。
- D) 正解。デプロイメント・パイプライン・ツール（2）とチュートリアルおよび規格（4）の2つは知識と学習を変換する成果物であるため、共有された単一のソースコード・リポジトリに含まれます。（参考文献：A、第20章）

35 / 40

ローカル（局所的）な発見をグローバル（全体）な改善に変換することの目標は何ですか？

- A) 開発と運用だけではなく、組織全体のプラクティスの状態を向上させること
 - B) 新規と既存のサービスすべてが集合知を活用しやすい状態にすること
 - C) 仕事のカルチャーをより協同的にしてシステムの安全とレジリエンスを高めること
 - D) 全員が快適で責任を感じるカルチャーを強化すること
- A) 正解。これはローカル（局所的）な発見をグローバル（全体）な改善に変換することの目標です。
（参考文献：A、第20章）
- B) 不正解。これは成文化された非機能要件（NFR）を通じた運用の基礎設計の目標です。
- C) 不正解。これは組織的な学習と改善を生み出す時間を保持することの目標です。
- D) 不正解。これは学習を日々の業務に取り入れて活性化することの目標です。

36 / 40

開発者は、アプリケーションと環境におけるロギングと暗号化に関する基準を、どのエンジニアでも正しく作成して使用できるように支援します。

この行動に役立つ共有ソースコード・リポジトリの項目ではないのはどれですか？

- A) コードライブラリーと推奨された構成
 - B) デプロイメント・パッケージ
 - C) オペレーティングシステム（OS）のパッケージおよびビルド
 - D) 秘密情報の管理ツール
- A) 不正解。コードライブラリーと推奨された構成は、アプリケーションで使うコンポーネントに対して、効果的なセキュリティ固有の構成を持つため、どのエンジニアにとってもアプリケーションのロギングと暗号化に関する基準を正しく作成して使用することが容易になります。
- B) 正解。デプロイメント・パッケージは開発者（開発）よりもエンジニア（運用）の成果物であるため、開発者の行動を支援する項目ではありません。この質問は、アプリケーションのロギングと暗号化に関する基準をどのエンジニアでも正しく作成して使用できるようにするために、開発者は何ができるかということです。（参考文献：A、第22章）
- C) 不正解。オペレーティングシステムのパッケージおよびビルドは、アプリケーションで使うコンポーネントに対して、効果的なセキュリティ固有の構成を提供するため、どのエンジニアにとってもアプリケーションのロギングと暗号化に関する基準を正しく作成して使用することが容易になります。
- D) 不正解。秘密情報の管理ツールは、アプリケーションで使うコンポーネントに対して、接続設定や暗号キーなどの効果的なセキュリティ固有の構成を提供するため、どのエンジニアにとってもアプリケーションのロギングと暗号化に関する基準を正しく作成して使用することが容易になります。

37 / 40

開発者がコードを導入する際に不正アクセスを可能にしてしまうリスクが常に存在します。

このリスクを軽減しないコントロールはどれですか？

- A) コードレビュー
 - B) コードテスト
 - C) 効果的なパッチ作業
 - D) 侵入テスト
- A) 不正解。コードレビューは、バックドア経由のアクセスを許可するマルウェアのコード挿入を明らかにします。
- B) 不正解。コードテストは不正アクセスの許可を明らかにします。
- C) 正解。効果的なパッチ作業は開発者のコード挿入を明らかにしません。バグを解決するだけです。
(参考文献：A、第22章)
- D) 不正解。侵入テストは、不正アクセスを許可する目的で作成された、もしくは残されているアプリケーションの弱点を明らかにします。

38 / 40

アプリケーションでのテレメトリ（遠隔計測）作成の例はどれですか？

- A) オペレーティングシステム（OS）の変更
 - B) 日々のシステムログのレビュー
 - C) セキュリティグループの変更
 - D) ユーザーパスワードのリセット
- A) 不正解。これはアプリケーションをホストするインフラストラクチャ環境への遠隔計測の使用であり、アプリケーションを特定するものではありません。
- B) 不正解。これはアプリケーションをホストするインフラストラクチャ環境への遠隔計測の使用であり、アプリケーションを特定するものではありません。
- C) 不正解。これはアプリケーションをホストするインフラストラクチャ環境への遠隔計測の使用であり、アプリケーションを特定するものではありません。
- D) 正解。これはアプリケーションを特定する遠隔計測の使用です。（参考文献：A、第22章）

39 / 40

オペレーション（運用）とセキュリティのリスクを軽減し、また、コンプライアンス要件をサポートする主要なコントロールとして作用するのはどのプロセスですか？

- A) 変更管理プロセス
- B) 構成管理プロセス
- C) リリースおよびデプロイメント管理プロセス
- D) サービスレベル管理プロセス

- A) 正解。かなりの規模のほとんどの IT 組織には、既存の変更管理プロセスがあります。これは、運用とセキュリティのリスクを軽減するための主要なコントロールです。コンプライアンスおよびセキュリティ マネージャーは、コンプライアンス要件を変更管理プロセスに依存しており、彼らは通常すべての変更が適切に承認されているという証拠を必要とします。（参考文献：A、第23章）
- B) 不正解。構成管理プロセスはサービスと各構成要素に関する情報を提供します。これは潜在的なリスクと改善について分析と検出をするために使用できますが、それらのリスクを軽減するためのプロセスではありません。構成管理は構成要素の計画、実装、コントロール、および報告を行い、また構成要素を検証します。
- C) 不正解。リリースおよびデプロイメント管理プロセスは、承認されたリリースを実際に実装するプロセスであり、承認された多数の変更を含めることができます。変更管理プロセスは変更をコントロールして承認し、その後の作業についてリリースおよびデプロイメント管理プロセスと調整します。
- D) 不正解。サービスレベル管理プロセスは、ビジネスで合意された必要なサービスレベルのデリバリーを確実にします。このプロセスでは必要なコンプライアンスとセキュリティ要件が記述されますが、それらのリスクを軽減するのに役立つプロセスではありません。

職務管理の分割を導入する際の欠点はどれですか？

- A) 職務管理の分割は、エンジニアが各自の作業に対して受け取るフィードバックの遅延と減少をもたらして開発の活動をしばしば妨げる可能性がある。
 - B) 職務管理の分割は開発者に変更をコードライブラリアンに提出するよう要求し、コードライブラリアンはその変更が製品に送られる前にレビューと承認を実施する。
 - C) 職務管理の分割はコードのチェックインとコードのレビューを要求して作業の品質に対する必要な保証を提供するため、不要な作業をもたらす。
- A) 正解。職務管理の分割はエンジニアが各自の作業に対して受け取るフィードバックの遅延と減少をもたらしてこれをしばしば妨げる可能性があります。分割によってエンジニアは各自の作業の品質について全面的な責任を負うことが困難になり、組織が組織的な学習を生み出す能力が低下します。したがって、可能である限り、職務管理の分割をコントロールとして使うことは避けるべきです。その代わりとして、ペアプログラミングやコードチェックインの継続的な調査やコードレビューなどのコントロールを選択するべきです。（参考文献：A、第23章）
- B) 不正解。この選択肢は、DevOpsではなく、ソフトウェア開発ライフサイクル（SDLC）を構成する古いアプローチを反映しています。DevOpsでの推奨コントロールにはペアプログラミングやコードチェックインの継続的な調査やコードレビューなどのコントロールがあります。これらのコントロールによって作業の品質に対する必要な再保証が得られます。さらに、これらのコントロールを準備することで、職務の分割が必要な場合には、作成したコントロールを使って同等の結果が得られると示すことができます。
- C) 不正解。職務管理の分割をコントロールとして使うことは避けるべきです。その代わりとして、ペアプログラミングやコードチェックインの継続的な調査やコードレビューなどのコントロールを選択するべきです。さらに、これらのコントロールを準備することで、職務の分割が必要な場合には、作成したコントロールを使って同等の結果が得られると示すことができます。

評価

次の表に、本模擬試験問題の正解を示します。

番号	正解	番号	正解
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



Driving Professional Growth

EXIN の連絡先

www.exin.com