



準備ガイド

2018年11月版

Copyright © EXIN Holding B.V. 2018. All rights reserved.
EXIN® is a registered trademark.
DevOps Master™ is a registered trademark.

No part of this publication may be published, reproduced, copied or stored in a data processing system or circulated in any form by print, photo print, microfilm or any other means without written permission by EXIN.



目次

1. 概要	4
2. 試験要件と試験仕様	8
3. 基本概念の一覧	12
4. 試験の参考文献	15

1. 概要

EXIN DevOps Master™ (DEVOPSM. JP)

適用範囲

EXIN DevOps Master™は、以下のトピックについての専門知識を認定します。

- DevOps の採用
- 計画、要件と設計
- 開発と展開
- 運用とスケールリング
- エンド・オブ・ライフ (EOL)

要約

DevOps とは「デベロップメント（開発）」と「オペレーション（運用）」を短縮した言葉です。DevOps とは、アプリケーションやサービスのライフサイクルを通して開発者、運用担当、支援スタッフといった IT プロフェッショナルの協業やコミュニケーションを重視した一連のベスト・プラクティスです。そして、

- 継続的インテグレーション：一日に数回、すべての開発された作業コピーを共有メインラインにマージ。
- 継続的デプロイメント：継続的なリリースや可能な限り頻繁にリリース。
- 継続的なフィードバック：全てのライフサイクルのステージでの関係者からのフィードバックを要求。

DevOps は、仕事に対する各個人の考え方を変えます。DevOps は、完了した仕事の多様性を評価し、ビジネスが価値を実現する速度を速めるための意思決定プロセスを支援し、また、社会的、技術的な変更による効果を測定します。DevOps は、持続可能な業務習慣を各個人や組織が開発して維持することを可能にする考え方であり、働き方です。

DevOps の成功は、

- ストーリーを共有し、共感を育てて、「他人を非難しない」文化を持つことです。人々やチームは、効率的かつ永続的な方法で技能を実践できるようになります。
- ビジネスのためにアプリケーションやサービスをジャストインタイム (JIT) で提供します。
- ビジネス要求のリスク視点でのアプローチで、IT サービスの継続性を確実にします。
- アプリケーションやサービスのエンド・オブ・ライフ (EOL) 条件を加味した完全なライフサイクルを管理します。

この認定試験は、知識のみならず実践的な能力の習得を目指しています。

チームの中で、DevOps を成功裏にすすめ、組織の中で基本概念を啓蒙できる DevOps マスターを育成します。

この認定試験は、DevOps 分野の専門家との協業により開発されたものです。

他の認定プログラムとの関連性

EXIN DevOps プログラム:



対象グループ

DevOps は、ソフトウェア開発の分野では、良く知られていますが、その基本概念は IT サービスや他のプロジェクトでも同様に適用可能です。EXIN DevOps Master™のトレーニングと認定は、組織で DevOps の実践を推進することを検討されているプロフェッショナルを対象としています。

EXIN DevOps Master™認定は、DevOps チームで働く方々や DevOps 的な働き方への変革を望む方に向けた資格です。対象とするグループには、プロダクトオーナー、アジャイル・スクラムマスター、プロジェクト管理者、テスト管理者、IT サービス管理者、プロセス管理者やリーン IT 実践者などが含まれます。この認定試験は、DevOps が適用される領域で、上位レベルの知識や経験を持つ、次の資格保持者の方に強くお勧めします。

認定のための要件

- 実践課題を含む EXIN 認定 EXIN DevOps Master™トレーニングの修了
- EXIN DevOps Master™試験の合格



試験の詳細内容

試験の形式:	スの多肢選択形式
問題数:	50
合格点:	65%
参考書やノートの持ち込み:	不可
電子機器の持ち込み:	不可
試験時間:	120 分

EXIN の試験規則はこの試験に適用されます。

ブルームレベル

EXIN DevOps Master™ 試験では、ブルームの改訂版タキノミーに基づき、ブルームレベル 3 およびレベル 4 で受験者をテストします。

- ブルームレベル 3：応用 - 認定候補者は、学習した内容を異なるコンテキストで活用できる能力があることを示します。このタイプの質問の目的は、認定候補者が取得した知識、事実、手法、およびルールを異なる方法で応用でき、新しい状況で問題を解決できること示すことです。この質問には、通常、ショートシナリオが含まれます。
- ブルームレベル 4：分析 - 認定候補者は、学習内容を細分化して理解する能力があることを示します。このブルームレベルは、主に実務でテストされます。このような実践課題の目的は、認定候補者が動機や原因を特定することで情報を検証および細分化して、推論し、一般的な結論につながる証拠を突き止める能力を示すことです。

教育・訓練

認定を取得するためにはトレーニングを受講する必要があります。認定候補者は、トレーニングに参加する前に、DevOps の原則とリーンおよびアジャイルの概念に関する基本的な知識を習得しておくことが期待されます。この知識は以下の認定や書籍で習得できます。

- EXIN DevOps プロフェッショナル認定
または
- 書籍『The Phoenix Project』を読む（文献リスト参照）。

授業時間

最小限の教育・訓練コースの最小受講時間は、24 時間です。これには、実践課題、試験の準備、小休憩が含まれます。宿題、試験準備、昼休みはこの時間に含まれません。

実践課題のためのガイドラインは、Partnernet で見つけることができます。実践課題に推奨される時間は最大 8 時間です。実践課題は、トレーニング環境の外部で完了できます。実践課題については、トレーナーは次の中から選択できます。

- GamingWorks シミュレーション（トレーナーに GamingWorks 認定が必要です）
- 書籍『Introduction to DevOps with Chocolate, LEGO and Scrum Game』¹のゲーム
- 独自の実践課題を開発する

¹ Introduction to DevOps with Chocolate, LEGO and Scrum Game

Dana Pylayeva
ISBN-13: 978-1-4842-2565-3
ISBN-10: 1484225643
Apress: 2017

トレーニングの長さは、トレーニングの適用範囲となる実践課題の作業量によって変わることがあります。以下のようなスケジュールになる場合がありますが、それ以外のスケジュールになることもあります。

- 理論についての2日間の授業と1日の実践課題
- 理論についての2日間のeラーニングトレーニングと作業環境における8時間の実践課題。認定候補者はeラーニングを通じて学習し、実践課題に取り組みます。トレーナーは実践課題に関する認定候補者の作業を評価します。
- 理論についての2日間の授業と作業環境における8時間の実践課題。トレーナーは実践課題に関する認定候補者の作業を評価します。

学習時間の目安

120時間、個人が習得している知識によります。この準備ガイドの第4章「試験の参考文献」の表は、認定試験でテストされる知識を習得できる参考文献です。

認定教育機関

認定教育事業者のリストを www.exin.com で参照できます。

2. 試験要件と試験仕様

試験要件	試験仕様	配分 %
1. DevOps の採用		28%
	1.1 DevOps の考え方とメリット	10%
	1.2 組織文化	12%
	1.3 原則と概念	6%
2. 計画、要求と設計		18%
	2.1 アプリケーション/サービス・ライフサイクル管理	4%
	2.2 プロジェクト憲章（スコープの定義）と目で見える管理	4%
	2.3 インフラストラクチャとアーキテクチャ設計	4%
	2.4 サービス・レベル・リクワイヤメント（SLR）とサービスレベルアグリーメント（SLA）	2%
	2.5 テスト戦略の導入	4%
3. 開発と展開		30%
	3.1 継続的デリバリと継続的インテグレーション	12%
	3.2 デプロイメント・パイプライン	4%
	3.3 継続的デプロイメント	4%
	3.4 自工程完結、リズム、仕掛（WIP）、一個流し	4%
	3.5 自動化、ツールとテスト	6%
4. 運用とスケーリング		22%
	4.1 データ、インフラストラクチャ、環境の管理：構成要素と要件	10%
	4.2 構成管理とバージョン管理	4%
	4.3 クラウドとイミュータブル・インフラストラクチャ	2%
	4.4 事業継続性	2%
	4.5 スケーリング	4%
5. エンド・オブ・ライフ（EOL）		2%
	5.1 プロダクト/サービスの EOL の条件	2%
合計		100%

試験仕様

1. DevOps の採用

- 1.1 DevOps の考え方とメリット
 - 1.1.1 DevOps 導入のシナリオの中で DevOps 失敗要素の分析
 - 1.1.2 DevOps の効果の説明
 - 1.1.3 DevOps が現在のソフトウェア開発プロセスに適している理由の説明
 - 1.1.4 DevOps では仕事のために特定の考え方が必要な理由の説明
 - 1.1.5 DevOps はどのような点でリーンやアジャイルスクラム手法と一致するかの説明
- 1.2 組織文化
 - 1.2.1 効果的な DevOps の4つの柱の重要性の理由を説明（協業、共感、ツール、スケールリング）
 - 1.2.2 DevOps の考え方の不足部分の為のシナリオの分析
 - 1.2.3 協業や DevOps の考え方、共感、信頼を育てることを通してグループからチームを作り上げる方法の説明
 - 1.2.4 協業に関する誤解がある場合の状況分析と正しい問題解決方法の識別
 - 1.2.5 コンフリクト・マネジメントの必要がある場合の状況分析と最善の解決策の識別
 - 1.2.6 人事管理によって多様性を促進する方法の説明と多様性が組織に与えるメリットの説明
- 1.3 DevOps の原則と概念
 - 1.3.1 他のソフトウェア開発手法（ウォーターフォール、アジャイル、スクラム）とその基本原理の有用性と使用の説明
 - 1.3.2 他の運用手法（IT サービスマネジメント）の有用性と使用の説明
 - 1.3.3 リーン・システム手法の有用性と使用の説明

2. 計画、要求と設計

- 2.1 アプリケーション／サービス・ライフサイクル管理
 - 2.1.1 DevOps が、現代のアプリケーション・ライフサイクル・マネジメント（ALM）に価値を追加する方法の説明
 - 2.1.2 DevOps が、サービス・ライフサイクル管理を使用した時に顧客の経験を改善する理由の説明
- 2.2 プロジェクト憲章と目で見える管理
 - 2.2.1 DevOps のプロジェクトスコープ（範囲）がどのように決定されるべきかの説明
 - 2.2.2 何故、DevOps プロジェクトを促進させる DevOps 手法よりも目で見える管理なのかの説明
- 2.3 インフラストラクチャとアーキテクチャ設計
 - 2.3.1 どのように DevOps は IT インフラとアーキテクチャを変えるか、または影響を与えるかの説明
 - 2.3.2 クラウドと仮想化技術が Dev（開発）と Ops（運用）をもっと容易に統合させる理由の説明

- 2.4 サービス・レベル・リクワイヤメント (SLR) とサービスレベルアグリーメント (SLA)
 - 2.4.1 DevOps では SLR と SLA を変える方法の説明
- 2.5 テスト戦略の導入
 - 2.5.1 テスト戦略が DevOps への移行期間中に変更される必要がある理由と方法の説明
 - 2.5.2 完全なユーザー・ストーリーのための分析

3. 開発と展開

- 3.1 継続的デリバリと継続的インテグレーション
 - 3.1.1 継続的デリバリは有効な DevOps にとって最も重要である理由の説明
 - 3.1.2 シナリオの中で継続的デリバリを統合する方法の分析
 - 3.1.3 シナリオの中で継続的デリバリの問題を解決する方法の分析
 - 3.1.4 継続的インテグレーションは有効な DevOps にとって最も重要である理由の説明
 - 3.1.5 分散チームまたは分散バージョン管理システム的环境中で、シナリオの中で継続的インテグレーションを完成させる方法の説明
 - 3.1.6 シナリオの中で継続的インテグレーションの問題を解決する方法の分析
- 3.2 デプロイメント・パイプライン
 - 3.2.1 DevOps デプロイメント・パイプラインの構造の論理の説明
 - 3.2.2 ビルドとデプロイメントのスクリプトを使用する方法の説明
- 3.3 継続的デプロイメント
 - 3.3.1 イテレーション計画とリリース計画が有効な DevOps にとって変更されなければならない理由の説明
 - 3.3.2 シナリオの中で継続的デプロイメントを導入する方法の分析
- 3.4 自工程完結、リズム、仕掛 (WIP) 、一個流し
 - 3.4.1 自工程完結、リズム、仕掛 (WIP) 、一個流しの概念の説明
 - 3.4.2 自工程完結、リズム、仕掛 (WIP) 、一個流しの課題のためのシナリオの分析と適した解決方法を見出す
- 3.5 自動化、ツールとテスト
 - 3.5.1 自動化が効果的な DevOps にとって重要な理由の説明
 - 3.5.2 一般的に DevOps を促進させるツールの使用方法の説明
 - 3.5.3 DevOps の考え方、文化を支えるツールの使用方法の説明
 - 3.5.4 DevOps のテストが自動化されることが重要である理由の説明
 - 3.5.5 自動化された受け入れテストの正しい方法の選択とシナリオの分析

4. 運用とスケーリング

- 4.1 管理しているデータ；インフラストラクチャと環境；そして、コンポーネント（構成部品）と従属条件
 - 4.1.1 DevOps でデータベースのデータを管理している時にどんな問題に直面するかの説明
 - 4.1.2 データベースが DevOps の中で使用され、問題の最善な解決策を提示するシナリオの分析
 - 4.1.3 デプロイメントのためのインフラ環境を準備する最善策またはデプロイメント以降のインフラ環境の管理とシナリオの分析
 - 4.1.4 コンポーネント（構成部品）を管理する一般に使用されている戦略の提案／示唆とシナリオの分析
 - 4.1.5 従属条件を管理する方法の説明
- 4.2 構成管理とバージョン管理
 - 4.2.1 バージョン管理が有効な DevOps にとって重要な要点である理由の説明
 - 4.2.2 データ、インフラストラクチャ、コンポーネントよりもバージョン管理を維持する方法の説明
 - 4.2.3 構成管理の問題を管理する最善の戦略の提案／示唆とシナリオの分析
- 4.3 クラウドとイミュータブル・インフラストラクチャ
 - 4.3.1 効果的な DevOps にとって、クラウドでのインフラ環境に移行する必然性のある時、もしくは無い時の説明
 - 4.3.2 クラウドでのインフラ環境が DevOps の中で管理する方法の説明
- 4.4 事業継続性
 - 4.4.1 DevOps はどの様に事業継続性のための手法を促進させるかの説明
- 4.5 スケーリング (4%)
 - 4.5.1 シナリオの分析、その状況の中でスケールアップもしくはスケールダウンに重要な理由と仮定の説明、それを実行する最善な方法の指摘
 - 4.5.2 スケーリングで何が上手くいかないかシナリオの分析と、問題解決の良い方法の指摘
 - 4.5.3 社会政策と採用習慣がスケーリング DevOps をどのように支援するかの説明

5. エンド・オブ・ライフ (EOL)

- 5.1 プロダクト／サービスの EOL の条件
 - 5.1.1 どの条件がサービスやプロダクトの終焉 (EOL) の前に確認が必要か説明

3. 基本概念の一覧

この章では、認定候補者が習熟しておく必要がある用語と略語を示します。

これらの用語を習得するだけでは、試験の対策として十分ではありません。認定候補者は用語の概念を理解し、その使用例について説明できる必要があります。

英語	日本語
A/B testing	A/Bテスト
Acceptance Tests	受け入れテスト
Affinity (<i>in DevOps</i>)	親密な関係
Agile	アジャイル
Anti-pattern	アンチパターン
Application Deployment	アプリケーション・デプロイメント
Artifact Management	成果物の管理
(Artifact) repository	(成果物の) リポジトリ
ATAM	ATAM
Automated testing	自動テスト
Automation	自動化
Binary files	バイナリーファイル
Blamelessness	無欠点、欠点の無い
Blue-Green deployment	ブルー・グリーン・デプロイメント
Build (Management)	ビルド (管理)
Build-time	ビルド時間
Canary releasing	カナリアリリース
Capacity testing	キャパシティテスト
Change Management	変更管理
Check-in	チェックイン
Cloud computing	クラウドコンピューティング
Collaboration (<i>in DevOps</i>)	協業
Commit (stage)	コミット (ステージ)
Communication styles	コミュニケーションスタイル
Compact	詰まった
Component (tests)	コンポーネント (テスト)
Configuration Management	構成管理
Containers	コンテナ
Continuous Delivery	継続的デリバリー
Continuous Deployment	継続的デプロイメント
Continuous Integration	継続的インテグレーション
Cycle time	サイクルタイム
Definition of Done (<i>in Agile Scrum</i>)	完了の定義
Dependency	依存
(Deployment) Pipeline	(デプロイメント・) パイプライン
Development Team	開発チーム
DevOps Engineer	DevOps エンジニア
Disciplined Agile	規律あるアジャイル
Distributed Team	分散チーム
Effective DevOps	効果的な DevOps

Event Management	イベント管理
Exploratory testing	予備テスト
Flow	フロー
Functional acceptance tests	機能受け入れテスト
Forensic tools	フォレンジックツール
Gatekeeper	ゲートキーパー
Happy path	ハッピーパス (happy path)
Human error	ヒューマンエラー
Incident Management	インシデント管理
Information radiators	情報ラジエータ
Infrastructure management	環境設定管理
Infrastructure Automation	環境設定の自動化
INVEST	INVEST (優れたユーザー・ストーリーの6原則)
Integration tests	統合テスト
Iteration	イテレーション (反復)
ITSM (IT Service Management)	ITSM (IT サービスマネジメント)
Ji-Kotei-Kanketsu (JKK)	自工程完結
Just-in-Time (JiT)	ジャストインタイム
Kaizen (<i>in Lean</i>)	カイゼン
Lean	リーン
Libraries	ライブラリ
Light-weight ITSM	軽量化した ITSM
Manual testing	手動テスト
Minimum Viable Product	MVP (価値ある最小限の製品)
Monitoring strategy	監視戦略
Negotiation styles	ネゴシエーションスタイル
Non-functional testing	非機能テスト
Obeya	大部屋
Orchestration	オーケストレーション
One-piece-flow	一個流し
Operations Team	運用チーム
Organizational Learning	組織学習
Plan-Do-Check-Act cycle (PDCA cycle)	PDCA
Process Master	プロセスマスター
(Product) Backlog	(製品) バックログ
Product Owner (<i>in Agile Scrum</i>)	プロダクトオーナー
Project charter	プロジェクト憲章
Pull system	プル型システム
Quality Assurance (QA)	品質保証 (QA)
Regression testing	回帰テスト
Release Coordinator	リリースコーディネータ
Reliability Engineer	信頼性エンジニア
Retrospective	振り返り
Rhythm (<i>in Lean</i>)	リズム
Root Cause Analysis	真因分析
Run time	ランタイム (稼働)
Sad path	サッドパス (sad path)
Scaling (<i>in DevOps or Agile Scrum</i>)	スケーリング
Scrum	スクラム

Scrum Master (<i>in Agile Scrum</i>)	スクラムマスター
Service Continuity	サービス継続性
Service Level Agreement (SLA)	サービスレベルアグリーメント (SLA)
Service Master	サービスマスター
Silos	サイロ
Sprint	スプリント
System of Engagement (SoE)	Web系システム (SoE)
System of Record (SoR)	業務系システム (SoR)
System tests	システムテスト
Test-Driven Development	テスト駆動開発
Tools	ツール
Toyota Production System (TPS)	トヨタ生産方式
Unit Test	単体テスト
Usability tests	ユーザビリティテスト
User Acceptance Testing (UAT)	ユーザー受け入れテスト (UAT)
User Story	ユーザー・ストーリー
Value Stream Mapping (VSM)	バリューストリーム・マップ (価値の流れ)
Velocity (<i>in Agile Scrum</i>)	ベロシティ (巡行速度)
(Vendor) lock-in	(ベンダー) ロックイン
Version Control	バージョン管理
Virtualization	仮想化
Waste (<i>in Lean</i>)	ムダ (リーン)
Waterfall	ウォーターフォール

4. 試験の参考文献

試験の参考文献

EXIN DevOps Master™試験に必要な知識は、次の文献に記載されています。

- A** **Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale**
Jennifer Davis, Katherine Daniels
ISBN-13: 978-1491926307
ISBN-10: 1491926309
O'Reilly Media; 初版 (2016年6月25日)

- B** **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**
Jez Humble, David Farley
ISBN-13: 978-0321601919
ISBN-10: 0321601912
Addison-Wesley Professional; 初版 (2010年8月6日)

- C** **エンタープライズ DevOps、DevOps によるビジネスを支援する IT サービス**
Koichiro (Luke) Toda, President Strategic Staff Services Corporation and
Director of TPS Certificate Institution
Nobuyuki Mitsui, CTO of Strategic Staff Services Corporation
White Paper; 2016年6月 (EXIN のサイトからダウンロードできます。)

追加文献

- D** 「The DevOps 逆転だ！究極の継続的デリバリー」
Gene Kim, Kevin Behr, George Spafford
ISBN-13: 978-4822285357
日経 BP 社

- E** **The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations**
Gene Kim, Jez Humble, Patrick Debois, John Willis
ISBN-10: 1942788002
ISBN-13: 978-1942788003
IT Revolution Press; 初版 (2016)

- F** その他の文献 (下記 URL 参照) :
<http://newrelic.com/devops>
<http://devops.com/>

コメント

追加の参考文献は、参考として知識を深めるためのものです。特に The DevOps 逆転だ！究極の継続的デリバリーはトレーニング前に読まれることを強く推奨します。



参考文献の表

1. DevOps の採用		
1.1	DevOps の考え方とメリット	
1.1.1	DevOps 導入のシナリオの中で DevOps 失敗要素の分析	A: 5 章 B: 1 章 C: 2 章
1.1.2	DevOps の効果の説明	B: 1 章 C: 4 章
1.1.3	DevOps が現在のソフトウェア開発プロセスに適している理由の説明	A: 3 章 B: 1 章 C: 4 章
1.1.4	DevOps では仕事のために特定の考え方が必要な理由の説明	A: 1、2、4 章 C: 4 章
1.1.5	DevOps はどのような点でリーンやアジャイルスクラム手法と一致するかの説明	B: 15 章 C: 1、4 章
1.2	組織文化	
1.2.1	効果的な DevOps の 4 つの柱の重要性の理由を説明（協業、共感、ツール、スケーリング）	A: 6、7 章
1.2.2	DevOps の考え方の不足部分の為のシナリオの分析	A: 7 章
1.2.3	協業や DevOps の考え方、共感、信頼を育てることを通してグループからチームを作り上げる方法の説明	A: 7、9 章
1.2.4	協業に関する誤解がある場合の状況分析と正しい問題解決方法の識別	A: 7、8、9、16 章
1.2.5	コンフリクト・マネジメントの必要がある場合の状況分析と最善の解決策の識別	A: 7、14 章
1.2.6	人事管理によって多様性を促進する方法の説明と多様性が組織に与えるメリットの説明	A: 7 章
1.3	DevOps の原則と概念	
1.3.1	他のソフトウェア開発手法（ウォーターフォール、アジャイル、スクラム）とその基本原理の有用性と使用の説明	A: 4 章 B: 1 章 C: 5 章
1.3.2	他の運用手法（IT サービスマネジメント）の有用性と使用の説明	C: 3、4 章
1.3.3	リーン・システム手法の有用性と使用の説明	A: 4 章
2. 計画、要求と設計		
2.1	アプリケーション/サービス・ライフサイクル管理	
2.1.1	DevOps が、現代のアプリケーション・ライフサイクル・マネジメント（ALM）に価値を追加する方法の説明	C: 1 章
2.1.2	DevOps が、サービス・ライフサイクル管理を使用した時に顧客の経験を改善する理由の説明	C: 4 章
2.2	プロジェクト憲章と目で見える管理	
2.2.1	DevOps のプロジェクトスコープ（範囲）がどのように決定されるべきかの説明	B: 10 章 C: 8 章
2.2.2	何故、DevOps プロジェクトを促進させる DevOps 手法よりも目で見える管理なのかの説明	B: 5 章 C: 7 章

2.3	インフラストラクチャとアーキテクチャ設計	
2.3.1	どの様に DevOps は IT インフラとアーキテクチャを変えるか、または影響を与えるかの説明	B: 11 章
2.3.2	クラウドと仮想化技術が Dev（開発）と Ops（運用）をもっと容易に統合させる理由の説明	B: 11 章
2.4	サービス・レベル・リクワイヤメント（SLR）とサービスレベルアグリーメント（SLA）	
2.4.1	DevOps では SLR と SLA を変える方法の説明	B: 12 章 C: 4 章
2.5	テスト戦略の導入	
2.5.1	テスト戦略が DevOps への移行期間中に変更される必要がある理由と方法の説明	B: 4 章
2.5.2	完全なユーザー・ストーリーのための分析	B: 4 章
3. 開発と展開		
3.1	継続的デリバリと継続的インテグレーション	
3.1.1	継続的デリバリは有効な DevOps にとって最も重要である理由の説明	B: 13、15 章
3.1.2	シナリオの中で継続的デリバリを統合する方法の分析	B: 3、5、15 章
3.1.3	シナリオの中で継続的デリバリの問題を解決する方法の分析	B: 15 章
3.1.4	継続的インテグレーションは有効な DevOps にとって最も重要である理由の説明	B: 3 章
3.1.5	分散チームまたは分散バージョン管理システムの環境で、シナリオの中で継続的インテグレーションを完成させる方法の説明	B: 3 章
3.1.6	シナリオの中で継続的インテグレーションの問題を解決する方法の分析	B: 3 章
3.2	デプロイメント・パイプライン	
3.2.1	DevOps デプロイメント・パイプラインの構造の論理の説明	B: 5、8 章 C: 7 章
3.2.2	ビルドとデプロイメントのスクリプトを使用する方法の説明	B: 1、6 章
3.3	継続的デプロイメント	
3.3.1	イテレーション計画とリリース計画が有効な DevOps にとって変更されなければならない理由の説明	C: 5、7 章
3.3.2	シナリオの中で継続的デプロイメントを導入する方法の分析	B: 10 章
3.4	自工程完結、リズム、仕掛（WIP）、一個流し	
3.4.1	自工程完結、リズム、仕掛（WIP）、一個流しの概念の説明	C: 4、7 章
3.4.2	自工程完結、リズム、仕掛（WIP）、一個流しの課題のためのシナリオの分析と適した解決方法を見出す	A: 1、2、9 章 B: 1 章 C: 4 章

	3.5	自動化、ツールとテスト	
	3.5.1	自動化が効果的な DevOps にとって重要な理由の説明	B: 1、8 章
	3.5.2	一般的に DevOps を促進させるツールの使用方法の説明	B: 8 章
	3.5.3	DevOps の考え方、文化を支えるツールの使用方法の説明	A: 12 章
	3.5.4	DevOps のテストが自動化されることが重要である理由の説明	A: 1 章 B: 3 章
	3.5.5	自動化された受け入れテストの正しい方法の選択とシナリオの分析	B: 8 章
	4. 運用とスケーリング		
	4.1	管理しているデータ；インフラストラクチャと環境；そして、コンポーネント（構成部品）と従属条件	
	4.1.1	DevOps でデータベースのデータを管理している時にどんな問題に直面するか説明	B: 12 章
	4.1.2	データベースが DevOps の中で使用され、問題の最善な解決策を提示するシナリオの分析	B: 12 章
	4.1.3	デプロイメントのためのインフラ環境を準備する最善策またはデプロイメント以降のインフラ環境の管理とシナリオの分析	B: 11 章
	4.1.4	コンポーネント（構成部品）を管理する一般に使用されている戦略の提案／示唆とシナリオの分析	B: 13 章
	4.1.5	従属条件を管理する方法の説明	B: 13 章
	4.2	構成管理とバージョン管理	
	4.2.1	バージョン管理が有効な DevOps にとって重要な要点である理由の説明	B: 2 章
	4.2.2	データ、インフラストラクチャ、コンポーネントよりもバージョン管理を維持する方法の説明	B: 10、11 章
	4.2.3	構成管理の問題を管理する最善の戦略の提案／示唆とシナリオの分析	B: 2 章
	4.3	クラウドとイミュータブル・インフラストラクチャ	
	4.3.1	効果的な DevOps にとって、クラウドでのインフラ環境に移行する必然性のある時、もしくは無い時の説明	A: 17 章 B: 11 章 C: 5、7 章
	4.3.2	クラウドでのインフラ環境が DevOps の中で管理する方法の説明	B: 11 章
	4.4	事業継続性	
	4.4.1	DevOps はどの様に事業継続性のための手法を促進させるかの説明	C: 4 章
	4.5	スケーリング	
	4.5.1	シナリオの分析、その状況の中でスケールアップもしくはスケールダウンに重要な理由と仮定の説明、それを実行する最善な方法の指摘	A: 14 章
	4.5.2	スケーリングで何が上手くいかないかシナリオの分析と、問題解決の良い方法の指摘	A: 15 章 B: 11 章
	4.5.3	社会政策と採用習慣がスケーリング DevOps をどのように支援するか説明	A: 6、7、14 章

5. エンド・オブ・ライフ (EOL)			
	5.1	プロダクト/サービスの EOL の条件	
	5.1.1	どの条件がサービスやプロダクトの終焉 (EOL) の前に確認が必要か説明	C: 7 章

注記：文献D「The DevOps 逆転だ！究極の継続的デリバリー」は、特に次の試験仕様項目の理解に役立ちます。

- 1.1
- 1.2
- 3.1
- 3.3
- 3.4
- 4.4

EXIN の連絡先

www.exin.com

