PeopleCert DevOps.

PeopleCert DevOps リーダーシップ

February 2021

シラバス v1.0J

電子メール： info@peoplecert.org, www.peoplecert.org

免責事項
本資料は、読者の皆様に役立つ情報を提供することを目的としています。発行元である PeopleCert International Ltd.は、本資料の作成にあたり、あらゆる注意を払っておりますが、本資料に含まれる情報の完全性、正確性、信頼性、適合性、可用性に関して、明示的であれ黙示的であれ、いかなる表明も保証も行うものではありません。また、PeopleCert International Ltd は、本資料に含まれている情報、指示、またはアドバイスにもとづいて、またはその結果として、いかなる損失または損害（特別、間接的、結果的なものであることを示していますが、これに限定されません）が生じたとしても、何らの責任を負うものではありません。

1. はじめに

DevOpsは、開発（Development）と運用（Operations）を組み合わせた語です。その名が示すとおり、DevOpsはソフトウェア開発（Dev）とITの運用（Ops）を共に進めるソフトウェア開発手法です。DevOpsがめざすのは、システム開発のライフサイクルを短縮しながら、ビジネス目標に沿ったより良いソフトウェアリリースを、より速く、より安く、高い頻度で提供することです。そのアプローチは、3つの重要成功要因（CSF）をカバーしています。その3つとは、文化の変化、プラクティスとプロセスの改善、自動化テクノロジの活用です。

業務に対するDevOpsアプローチは、いろいろな組織で成功を収めてきました。そのため、**DevOps**に精通した人材を求める組織の数は増加の一途をたどり、業界を超えて、その人材の獲得競争が起きています。組織が求めているのは、文化を変え、DevOpsの環境で快適に働き、多数のテクノロジやツール（いわゆる「ツールチェーン」）を使いこなして、コーディング、ビルド、テスト、パッケージング、リリース、構成、モニタリングなどのプロセスをマネジメントできる人材です。

日々進化しているこの状況の中で、PeopleCertはDevOpsの資格を設計しました。この資格は、多くの組織が直面しているスキルギャップを解消し、ビジネス目標の実現をサポートすること、そしてコミュニケーション、標準化、コラボレーション、自動化を向上させて、高品質なソフトウェアプロダクトをより良く、より速く、より低コストで提供することを求める市場のニーズを反映したものです。

このニーズを反映するため、PeopleCertのDevOpsの資格は以下のような構成をしています。

- **PeopleCert DevOpsファンダメンタル**（Fundamentals）：14～18時間のトレーニング
  受験者は、15の基本プラクティスを通して、効果的なDevOpsを提供する方法について実践的なガイダンスを得ることができます。これは、確立されたITSM（ITサービスマネジメント）に特に関連しています。
- **PeopleCert DevOpsリーダーシップ**（Leadership）：14～18時間のトレーニング
  受験者は、DevOpsのプラクティスの導入を主導する方法や、より良いコラボレーションとコミュニケーションをめざすための文化の変化を達成する方法について、ガイダンスを得ることができます。
- **PeopleCert DevOps エンジニアリング**（Engineering）（近日発表）
  受験者は、自動化ツールの使用方法や、これらのツールがどのようにDevOpsの文化とプラクティスをサポートするかについての実践的なガイダンスを得ることができます。

PeopleCert DevOpsファンダメンタル（以下、ファンダメンタル）認定は、受験者がDevOpsの原則とプラクティスに関する知識とスキルを構築するために必要となる基本的な**知識**をカバーしています。さらに、**PeopleCert DevOpsリーダーシップ**（以下、リーダーシップ）認定は、DevOpsに関するより高度なスキルとプラクティスと知識をカバーしています。

これらのスキルの基礎となる知識体系は、PeopleCertがATO（Accredited Training Organization：認定教育機関）に提供した公式コースウェアで紹介しています。コースのシラバスの主な目的は、DevOpsに携わる人々の認定に必要な基本事項を紹介することです。シラバスは、この資格に関する学習によって得られる成果を文書化し、これらの成果が特定の資格レベルで達成されていることを証明するために、受験者が満たすべき要件を示しています。

2. PeopleCert DevOps リーダーシップ認定

2.1. リーダーシップ資格の目的

リーダーシップ資格の目的は、受験者がDevOpsの基本的なスキルについて十分な知識と理解をもっていること、そしてDevOps環境のメンバと共に、またはそのメンバのリーダーとして、これらのスキルや知識を分析し、適用して、効果的に仕事ができることを確認することです。**PeopleCert DevOps ファンダメンタル**（以下、**ファンダメンタル**）資格が、この**リーダーシップ**資格の前提条件になっています。

2.2. 対象となる受験者

リーダーシップ資格は、**PeopleCert DevOps**資格のスキームの**2番目のレベル**となっており、DevOps環境の有能なリーダーになりたい人を対象としています。受験者には、DevOpsの用語、原則、ツール、プラクティスについてしっかりとした**知識**と**理解**をもつこと、および効率的かつ効果的にツールを使いこなす**応用力**があることを示すことが求められます。また、この認定は個人として認定を求める受験者にも対応しています。

この認定の資格保有者は、**リーダー**レベルの知識をもち、さまざまなツールを活用するDevOpsリーダーシップのプラクティスをしっかりと理解して、それをDevOpsのプラクティスに関わる日常業務に適用できることを証明されます。 なお、この前提となる基礎レベルのスキルと知識は、**PeopleCert DevOps**資格のスキームの**ファンダメンタル**レベルでカバーしています。

3. 学習目標

リーダーシップレベルのコースで受験者が学ぶことは、DevOpsのリーダーが、DevOpsの変革を明確化し、計画し、アプローチし、さらにその結果を検証し、維持するために使用するコンセプト、用語、原則、ツールです。さらに、DevOpsのフルスタックアプローチを組織にどのように導入し、実行していくかを学びます。それと同時に、現代の企業がDevOpsのリーダーシップを必要とするのはなぜか、そして価値の提供とどのように整合させていくのかを理解します。また、スクラムの方法論、人と文化の意味合い、組織にDevOpsを適応させるために使用するプラクティス、プロセス、自動化、テクノロジについても学習します。

リーダーシップ認定の**資格保有者**は、次に示すようなDevOpsの知識と理解と実際的な応用力を実証することができます。

- リーダーシップとは、そして DevOps におけるリーダーシップとは
- DevOps の変革によりどのように組織を導くか
- DevOps の緊急性を浸透させる
- ビジネス目標を明確化し調整する
- 変革のビジョンと戦略を構築する
- 不可欠なステークホルダを特定し動かす
- 自己組織化された文化と機能横断型チームを導く
- フィードバックを集め、伝えて、組み込む

- バリューストリームを横断するフローを可能にする
- 作業をイテレーションに分解して学習と経験を加速させる
- 継続的デリバリへ向かって導く
- 継続的改善の文化を導く
- イノベーションへ向かって導く
- 組織の対障害弾力性と持続可能性を向上させる

3.1. 資格スキームレベル

上記の学習目標を通して、受験者は以下に示した分野に関連する知識とスキルを実証します。

主要トピック
DevOps のリーダーシップの紹介
価値の提供へ向けた DevOps の変革の明確化と調整
DevOps の変革の計画とアプローチ
DevOps のフルスタックのエンゲージと実行
DevOps の変革の結果検証と維持

4. リーダーシップ試験

リーダーシップ認定試験は、受験者が上記のようなDevOpsリーダーシップの原則の知識をもち理解しているかどうか、また、現代の企業の実際の現場で、これらの知識をどのように応用し、分析して、DevOpsの変革を支援することができるかを検証できるように設計されています。

リーダーシップ認定試験では、学習レベルの違いに関する**ブルームの目標分類学**[1]（Bloom's taxonomy）の認知領域における以下の**4つ**のカテゴリに焦点を当てています。

- **知る**
- **理解する**
- **応用する**
- **分析する**

4.1. 評価アプローチ

リーダーシップ認定で使用する評価アプローチは、「知る」「理解する」「応用する」「分析する」の**4つの基本的な**分類に焦点を当てています。

[1] ブルームの目標分類学では、認知領域における学習を「知る、理解する、応用する、分析する、評価する、創造する」の 6 つに定義している。これらのレベルを、単純なものから複雑なものへと連続的に積み上げていく。6 つ目の学習レベルを達成するには、そこまでの 5 つのレベルを確実に習得していることが必要とされる。

「**知る**」は、事実から理論にわたって、以前に学習した内容を思い出すことと定義されます。これは、認知領域の中で最も低いレベルの学習成果を表すものです。 このような学習成果について、以下のことを知り、それを思い出すことを評価目標としています。

- 共通の および/または 基本的な用語、定義、コンセプト、原則
- 具体的なプロセス
- プロセス、手順、プロジェクトマネジメント手法

「**理解する**」は、理解度の中で最も低いレベルであり、その過程での解釈、変換、推定などの要素を含め、教えられた教材の意味を把握する能力を必要とします。 このような学習成果や評価目標は、単に情報を思い出すだけではなく、以下のことも含まれます。

- 事実、コンセプト、原則の理解
- 教材の解釈（コード、図、グラフ、文章、ダイヤグラムなど）
- 使用するプロセス、手続き、手法の正当性の理解

「**応用する**」は、受験者がもっている知識と理解を組み合わせて、抽象化できるようになることが求められるレベルです。より具体的には、受験者がもっている知識と理解を応用して、抽象化したもの、一般的な原則、あるいは手法を具体的な状況に適用することが期待されています。このような学習成果や評価目標は、単に情報を思い出すだけではなく、以下のことも含まれます。

- 新しい、具体的な状況下で、アイデア、原則、理論を使用する
- 与えられた状況下で、適切な方法やツールを選択する、原則を適用する、特定のアプローチを使用する、あるいは選択肢を識別することができる
- 学んだことを新しい状況に適用する
- ルール、手法、コンセプト、原則、理論を適用する
- この分野の学習成果は、「理解する」以下のレベルよりも高度の理解を必要とする

「**分析する**」は、「応用する」を超えるレベルであり、受験者が情報を構成要素に分解し、その組織構造を理解し、推論することができるようになる必要があります。より具体的には、受験者が情報を分解し、識別し、図示し、検出し、区別し、説明できることが必要です。これらはすべてこの学習レベルで重要な作業であり、これまでの「知る」「理解する」「応用する」レベルを含みます。このような学習成果や評価目標は、単に知り、理解し、分析するだけではなく、以下のことも含まれます。

- 問題を分析するために使用できるパターンを見分ける
- 因果関係を見極める
- 推論を行う
- 一般化の根拠を見出す
- 部品を特定し、部品間の関係を分析し、その組織構造の原理を認識する

評価は、上記の認知領域のカテゴリに対応した評価目標を使用しているため、これらの学習成果が組み込まれています。

4.2. 合格基準とトレーニングの要件

この試験では、受験者が前提資格として**ファンダメンタル**認定の**資格**を保有していることが必須です。トレーニング要件はありません。

リーダーシップレベルの資格を得るために、受験者は、DevOpsリーダーシップの基本的な用語、原則、プロセス、プラクティスの知識と理解があり、現代の企業の実際の現場で、これらの知識を応用し、分析して、DevOpsの変革を支援することができることを実証できなければなりません。また、受験者は、PeopleCert認定トレーニングパートナーによる**認定トレーニング**を受講することが推薦されています。

4.3. 試験の実施形式

リーダーシップ試験の実施形式の詳細は、以下の表のとおりです。

試験配信	コンピュータ （オンライン監督付きまたは教室）
出題形式	**多岐選択問題 20問** *各問とも正解を 1 つ選択します。*
試験時間	**1時間（60分）** *母国語以外で受験する場合や障害をもつ受験者の場合は、30分の追加時間が認められています。*
合格基準	**70% （20問中14問正解）**
試験監督	**あり** *在室の監督員またはオンライン監督です。*
参考資料の持ち込み	**禁止** *受験する室内に資料はいっさい持ち込めません。*
前提条件	**PeopleCert DevOps Fundamentals認定の資格を保有していること**
成績優秀者対応	**N/A**
認定の有効期間	**無期限**

試験問題は、以下の仕様に基づいて定期的に更新される問題テストバンク(QTB：Question Test Bank)から抽出した問題セットから出題されます。難易度はどの問題セットも同じレベルで、１つの問題セットの試験の難易度は、他の問題セットの試験の難易度と同じです。受験者が複数回受験した場合でも、同じ問題セットが割り当てられることはありません。

5. シラバスの詳細

シラバスは、**大項目**の分類と、それを細分化したセクションから構成されており、各セクションには1桁のセクション番号が付いています。合計**18時間**の認定トレーニングの受講が推奨されています。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
1.0 DevOps のリーダーシップとは	1.1 リーダーシップとは？	1.1.1 導くとはどういうことか？	1.1.1.1	DevOps の変革において、管理することと導くことの違いを理解します。
			1.1.1.2	DevOps における変革的リーダーシップのコンセプトの重要性を復習します。
			1.1.1.3	Jim Collins の著作から、5 つのレベルのリーダーとその特性を示します。
			1.1.1.4	3 種類のリーダーを示します。
	1.2 DevOps の基本原則	1.2.1 復習：DevOps とは？	1.2.1.1	DevOps ファンダメンタルで学んだ DevOps の定義を復習し、自身の組織にとってそれが何を意味するかを考えます。
		1.2.2 DevOps のフルスタック	1.2.2.1	DevOps を「フルスタック」として考えるアプローチを復習します。
		1.2.3 DevOps の基本原則	1.2.3.1	CALMS に関連する価値について復習します。
		1.2.4 DevOps の 15 の基本プラクティス	1.2.4.1	DevOps リーダーシップの 15 の基本プラクティスの重要性について復習します。
		1.2.5 テクノロジと自動化の活用	1.2.5.1	DevOps リーダーシップに対する、継続的インテグレーションとデリバリとデプロイメントとの関係について復習します。
	1.3 変革により組織を導く	1.3.1 Lewin の変革モデル	1.3.1.1	20-20 モデルがどのように形成されたかを示します。
		1.3.2 継続的改善	1.3.2.1	ITIL® の継続的改善モデルを示します。
		1.3.3 変革のための明確な 20/20 ビジョン	1.3.3.1	Lewin の変革の 3 段階と、DevOps の変革によるその実行について示します。
		1.3.4 20/20 変革モデル	1.3.4.1	20/20 変革モデルの詳細を示します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
2.0 価値の提供へ向けた DevOps の変革の明確化と調整	2.1 DevOps の緊急性を浸透させる	2.1.1 IT バリューの提供の問題	2.1.1.1	DevOps ムーブメントを推進する組織において、IT がビジネスバリューを提供するのを妨げている障壁と、DevOps の緊急性を理解するうえでのその重要性について復習します。
		2.1.2 変化の推進要因	2.1.2.1	DevOps ムーブメントに向かう組織における変化の 4 大外部要因と、DevOps の緊急性を理解するうえでのその重要性について復習します。
		2.1.3 テクノロジの採用と変化	2.1.3.1	Nicholas Carr の「テクノロジの採用曲線」を示します。
		2.1.4 複雑さが脆弱性と負債を生み出す	2.1.4.1	DevOps ムーブメントを推進する組織において、IT がビジネスバリューを提供するのを妨げている障壁と、DevOps の緊急性を理解するうえでのその重要性について復習します。
		2.1.5 標準化の必要性	2.1.5.1	David Sward の組織改革モデルにより、標準化の必要性を示します。
		2.1.6 標準化 対 複雑さ	2.1.6.1	標準化と複雑さを対比させて、標準化がどのように時間とコストの節約になるかを示します。
		2.1.7 Gleicher の変化の公式	2.1.7.1	Gleicher の変化の公式を示します。
	2.2 ビジネス目標を明確化し調整する	2.2.1 復習：ビジネスバリューとは？	2.2.1.1	ビジネスバリューの定義と 3 つの次元について復習します。
		2.2.2 価値が整合していなければどうなるか？	2.2.2.1	価値を整合させることの重要性を強調します。
		2.2.3 基軸（True North）の重要性	2.2.3.1	基軸（True North）を定義します。
			2.2.3.2	基軸の重要性を説明します。
		2.2.4 基軸価値観と原則の確立	2.2.4.1	基軸価値観を確立します。
		2.2.5 ミッションとビジョンの定義	2.2.5.1	変革の目的の理解という点からミッションとビジョンのコンセプトを比較対照します。
		2.2.6 基軸に沿ったシステムの構築	2.2.6.1	例を挙げて基軸に沿ったシステムの構築ステップを示します。
		2.2.7 ビジネス目標の例	2.2.7.1	ビジネス目標の例を示します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		2.2.8 計画された企業バックログ	2.2.8.1	計画された企業バックログを定義し、それをビジネス目標とどう整合させる必要があるかを示します。
		2.2.9 計画外の作業とチームのバックログ	2.2.9.1	計画外の作業がフローにどのような影響を与えるかを示します。
		2.2.10 要望の発生源	2.2.10.1	すべての要望の発生源を特定します。
		2.2.11 すべての種類の作業に可視性を与える	2.2.11.1	計画された作業と計画外の作業の両方に可視性を与えることを説明します。
		2.2.12 基軸のケーススタディ	2.2.12.1	ケーススタディにより基軸を説明します。
3.0 DevOps の変革の計画とアプローチ	3.1 変革のビジョンと戦略を構築する	3.1.1 ビジョンの重要性	3.1.1.1	Andy Stanley の引用によりビジョンの重要性について説明します。
			3.1.1.2	明確な変革のビジョンの重要性を示します。
		3.1.2 DevOps の変革の進化	3.1.2.1	DevOps の変革の進化の 3 段階について説明します。
		3.1.3 バイモーダル IT あるいは変速 IT	3.1.3.1	「バイモーダル IT」という語を定義し、DevOps の変革におけるその重要性について説明します。
		3.1.4 DevOps チームの規模拡大パターン	3.1.4.1	DevOps チームの規模拡大パターンを示します。
		3.1.5 サイロに橋を架けるプラクティスのコミュニティ	3.1.5.1	チームメンバをプラクティスのコミュニティに参加させることにより、サイロに橋を架ける方法を説明します。
		3.1.6 現在の状態の明確化	3.1.6.1	自身の現在の状態を明確化します。
		3.1.7 システム思考	3.1.7.1	システム思考を定義します。
			3.1.7.2	システム思考について説明します。
		3.1.8 氷山モデル	3.1.8.1	氷山モデルを定義します。
			3.1.8.2	システム思考を適用して、組織における DevOps の明確で説得力のあるビジョンをまとめ、ビジネス価値に沿った測定可能な目標、ミッション、目的、価値を提案します。
		3.1.9 現在の状態の例	3.1.9.1	例を挙げて現在の状態を示します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		3.1.10 現在の状態の評価	3.1.10.1	氷山モデルを作成して、自身の組織の現在の状態を明確にします。
		3.1.11 DevOps の変革の将来の状態の例	3.1.11.1	例を挙げて DevOps の変革の将来の状態を明確にします。
		3.1.12 DevOps のメンタルモデルと構造	3.1.12.1	DevOps のメンタルモデルと構造を示します。
	3.2 不可欠なステークホルダを特定し動かす	3.2.1 DevOps の変革にとって重要なステークホルダのマップを作成する	3.2.1.1	DevOps の変革にとって不可欠なステークホルダのマップを作成して示します。
		3.2.2 ステークホルダマネジメントプロセス	3.2.2.1	ステークホルダマネジメントプロセスの 4 つのステップを示します。
		3.2.3 DevOps の変革におけるさまざまなステークホルダグループを特定する際の主な考慮点	3.2.3.1	DevOps の変革におけるさまざまなステークホルダグループを特定する際の主な考慮点を説明します。
		3.2.4 抵抗を克服し、重要なステークホルダに影響を与えて、組織の DevOps のビジョンと戦略の策定に全面的に参加してもらう方法	3.2.4.1	DevOps の変革にとって重要なステークホルダのマップを作成して、その抵抗を克服し、影響を与えて、組織の DevOps のビジョンと戦略の策定に全面的に参加してもらう方法を見つけます。
		3.2.5 ステークホルダの支援を見積もる	3.2.5.1	支援レベルと影響力レベルの図を使用して、さまざまなステークホルダからの支援を見積もります。
4.0 DevOps のフルスタックのエンゲージと実行	4.1 自己組織化された文化と機能横断型チームを導く	4.1.1 混乱の壁の打破	4.1.1.1	混乱の壁について復習します。
		4.1.2 病理学的文化と官僚的文化と生成的文化	4.1.2.1	Westrum による病理学的文化と官僚的文化と生成的文化について復習し、その違いを比較します。
		4.1.3 タスク専門型 対 機能横断型	4.1.3.1	専門分野指向のバリューチェーンとプロダクトライン指向のバリューチェーンを比較します。
		4.1.4 機能横断型チームの重要性	4.1.4.1	機能横断型チームをもつことの重要性を示します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		4.1.5 自己組織化を可能にする	4.1.5.1	DevOps にとっての自己組織的チームの重要性を説明します。
			4.1.5.2	どうすれば自己組織的チームがうまく機能するかを説明します。
		4.1.6 アジャイルスクラムチーム	4.1.6.1	アジャイルスクラムチームを示します。
			4.1.6.2	開発チームのメンバが特定の役職にしがみつかない理由を示します。
		4.1.7 アジャイルチーム対 DevOps チーム	4.1.7.1	アジャイルチームと DevOps チームを比較します。
		4.1.8 リーダーシップとチームの権限	4.1.8.1	さまざまなレベルのチームと各チームに対応する権限を示します。
		4.1.9 DevOps チーム内でゼネラリストと純粋なスペシャリストとのバランスをとることの重要性	4.1.97.1	DevOps チーム内でゼネラリストと純粋なスペシャリストとのバランスをとることの重要性について復習します。
		4.1.10 DevOps チーム編成の進化のフェーズ	4.1.10.1	DevOps チーム編成の進化の 3 つのフェーズを挙げます。
		4.1.11 プロダクトチームやプラットフォームチームにおける役割ごとのサイロ構造	4.1.11.1	プロダクトチームやプラットフォームチームにおける役割ごとのサイロ構造について復習します。
			4.1.11.2	DevOps チームをどのように構成すれば、クラウドの原則と役割を採用できるかを復習します。
		4.1.12 チーム編成の変化	4.1.12.1	DevOps の変革に伴うチームの変化を紹介します。
		4.1.13 文化の変化と構造の変化	4.1.13.1	DevOps の変革に伴う文化の変化と構造の変化を紹介します。
		4.1.14 信頼と管理責任のモデル	4.1.14.1	信頼と管理責任のモデルを用いて、リーダーシップは、チーム編成戦略および構築した文化と合ったものでなければならないことを説明します。
		4.1.15 ナレッジとスキルの計画	4.1.15.1	ナレッジとスキルマネジメントにおける重要用語を定義します。
		4.1.16 ナレッジとスキル	4.1.16.1	ナレッジとスキルを定義し比較します。
		4.1.17 要員と能力マネジメント	4.1.17.1	要員と能力マネジメントのさまざまな方法を示します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		4.1.18 スキルナレッジマトリクスの作成	4.1.18.1	スキルナレッジマトリクスを作成する 3 つのステップを説明します。
		4.1.19 公表された IT 能力フレームワーク：ECF と SFIA	4.1.19.1	公表されている 2 つの IT 能力フレームワークである ECF と SFIA を紹介します。
	4.2 フィードバックを集め、伝えて、組み込む	4.2.1 将来の状態を特定するためのインプット	4.2.1.1	フィードバックを集めるためのリーンの 4 つのツールを示します。
		4.2.2 VOC と CTQ としての価値	4.2.2.1	リーンの VOC と重要品質要因（CTQ：Critical to Quality）の技法を用いて組織のビジネスバリューを分析し、DevOps のビジョンと価値の提供の方向性が一致していることを確認します。
		4.2.3 重要品質要因（CTQ：Critical to Quality）を測定する	4.2.3.1	測定ツールとして CTQ、メトリクス、KPI、CSF を定義します。
		4.2.4 差異はコントロールを示す	4.2.4.1	共通要因差異と特殊要因差異を定義し比較します。
		4.2.5 顧客のエンゲージメントに関する役割：デリバリ	4.2.5.1	顧客のエンゲージメントに関する役割、関係マネージャとプロダクト／サービスオーナーについて定義し説明します。
		4.2.6 プロダクト／サービスオーナーの考慮点	4.2.6.1	プロダクトオーナーとサービスオーナーについて説明し比較します。
			4.2.6.2	プロダクト／サービスオーナーにとって重要なコンピテンシについて説明します。
		4.2.7 関係マネージャの役割	4.2.7.1	DevOps の変革において、関係マネージャの役割が、他の役割やプロセスとどのように相互作用するかを説明します。
		4.2.8 エンゲージメントの役割と構築	4.2.8.1	ビジネスアナリストの役割を、アジャイルおよび DevOps 環境におけるエンゲージメントの役割として定義し説明します。
	4.3 バリューストリームを横断するフローを可能にする	4.3.1 複雑さはフローと時間に影響を与える	4.3.1.1	リーンのムリ、ムラ、ムダについて復習します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		4.3.2 リーンの 3 種類の作業	4.3.2.1	リーンの 3 種類の作業について復習し比較します。
		4.3.3 標準化の戦略的観点	4.3.3.1	標準化の戦略的観点を示し、説明します。
		4.3.4 複雑さと計画外の作業の影響	4.3.4.1	理想的なパターンと現在のパターンにおける計画外の作業の割合を示します。
		4.3.5 バリューストリームの改善フェーズ	4.3.5.1	バリューストリームの改善の 2 つのフェーズを説明します。
		4.3.6 バリューストリームマッピング	4.3.6.1	バリューストリームマッピングのプラクティスの重要な構成要素についてまとめます。
			4.3.6.2	バリューストリームマッピングのプラクティスの重要な構成要素について、例を挙げてまとめます。
		4.3.7 プロセスのムダ	4.3.7.1	さまざまな種類のムダを特定します。
		4.3.8 メトリクス：何を測定するべきか？	4.3.8.1	バリューストリームマッピングにおけるさまざまな種類のメトリクスを特定します。
		4.3.9 リーダーが使用する目に見える管理	4.3.9.1	目に見える管理の重要性とリーダーがそれを使用することについて説明します。
		4.3.10 目に見える管理の例	4.3.10.1	目に見える管理の例をいくつか挙げます。
		4.3.11 スクラムにおけるカンバン（「スクラムバン」）	4.3.11.1	スクラムにおけるカンバンについて定義し説明します。
			4.3.11.2	「WIP（Work in Progress：進行中の作業）制限」を定義します。
		4.3.12 計画外の作業を可視化する	4.3.12.1	カンバンボードを使用して計画外の作業を可視化する方法について説明します。
		4.3.13 共通のコミュニケーションチャネルを構築する重要性	4.3.13.1	共通のコミュニケーションチャネルを構築する重要性を示します。
		4.3.14 コミュニケーションと透明性のソリューション	4.3.14.1	ナレッジを共有し、コミュニケーションの透明性を高めるさまざまなソリューションを挙げます。
		4.3.15 コミュニケーションの考慮点	4.3.15.1	コミュニケーションツールを選択する際の考慮点を説明示します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
	4.4 作業をイテレーションに分解して学習と経験を加速する	4.4.1 アジャイル型 対 ウォーターフォール型プロジェクトマネジメント	4.4.1.1	ウォーターフォール型開発モデルとアジャイル型開発モデルを比較対照します。
		4.4.2 反復型プロダクトマネジメント	4.4.2.1	通常のデザイン思考とリーンのデザイン思考を比較します。
		4.4.3 アジャイルとスクラムの柱	4.4.3.1	アジャイル原則について復習します。
		4.4.4 リーダーシップの観点から見たスクラム	4.4.4.1	アジャイルスクラムのプラクティスの主要構成要素について復習します。
		4.4.5 アジャイル XP によりシフトレフトを可能にする	4.4.5.1	テスト駆動開発を定義します。
		4.4.6 アジャイル XP のプラクティス	4.4.6.1	XP プラクティスのペアプログラミングと持続可能なペースについて説明します。
			4.4.6.2	XP プラクティスのコードの共同所有と継続的リファクタリングについて説明します。
		4.4.7 ベロシティ改善を可視化する	4.4.71.	図によるベロシティ改善の可視化について説明します。
	4.5 継続的デリバリのためのリーダーシップ	4.5.1 自動化の要求	4.5.1.1	組織文化が共有された継続的デリバリのパイプラインを受け入れ、使用する前に、リーダーが対処する必要のある、組織の変化に関する 5 つの質問を示します。
		4.5.2 継続的デリバリ	4.5.2.1	「継続的デリバリ」について復習します。
			4.5.2.2	DevOps にとっての継続的デリバリの重要性について復習します。
			4.5.2.3	継続的インテグレーションとデリバリとデプロイメントの関係について復習します。
		4.5.3 継続的デリバリの大規模適用	4.5.3.1	DevOps を展開する際に留意するべき継続的デリバリの 5 つの主な目的を示します。
		4.5.4 継続的テスト	4.5.4.1	継続的テストについて復習します。
		4.5.5 コンポーネントテスト	4.5.5.1	統合におけるコンポーネントテストについて説明します。
		4.5.6 サブシステムテストまたはアプリケーションテスト	4.5.6.1	サブシステムテストやアプリケーションテストに役立つサービス仮想化の使用方法について説明します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		4.5.7 エンドツーエンドの企業システムテスト	4.5.7.1	エンドツーエンドの企業システムテストについて説明します。
		4.5.8 孤立した機能ブランチ	4.5.8.1	伝統的な孤立した機能ブランチの例で、ブランチの課題を示します。
		4.5.9 継続的インテグレーション	4.5.9.1	継続的インテグレーションの例で、ブランチの課題を示します。
		4.5.10 CI の重要性	4.5.10.1	CI の重要性を強調します。
		4.5.11 トランク管理とゲーテッドコミット	4.5.11.1	統合のための受け入れ基準の重要性を説明します。
		4.5.12 リリースとデプロイの歩調を合わせる	4.5.12.1	アジャイルおよび DevOps 環境におけるリリースとデプロイの計画を示します。
		4.5.13 トランク管理とリリース戦略	4.5.13.1	アジャイルおよび DevOps 環境における 2 つのリリースおよびデプロイ戦略を示します。
		4.5.14 ブルーグリーンデプロイメント	4.5.14.1	「ブルーグリーンデプロイメント」を定義します。
		4.5.15 DevOps ツールチェーンにおける各ツールの機能と役割を理解することの重要性	4.5.15.1	DevOps ツールチェーンにおける各ツールの機能と役割を理解することの重要性を強調します。
		4.5.16 DevOps ツールチェーンの機能	4.5.16.1	DevOps ツールチェーンの機能を説明します。
		4.5.17 オーケストレーションと統合	4.5.17.1	統合ツールの例を挙げます。
		4.5.18 オープンソースソフトウェアのメリットとデメリット	4.5.18.1	オープンソースソフトウェアのメリットとデメリットを示します。
		4.5.19 より大きなツールエコシステム	4.5.19.1	ツールエコシステムを説明します。
		4.5.20 自動化とツール戦略	4.5.20.1	統合のツールを選択する戦略を説明します。
5.0 DevOps の変革の結果検証と維持	5.1 継続的改善の文化を導く	5.1.1 継続的改善の考え方	5.1.1.1	継続的改善の考え方の 3 つの特徴を説明します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		5.1.2 経験の文化の構築	5.1.2.1	個人的に説明責任をもつ従業員と被害者意識をもつ従業員という 2 種類の従業員を説明します。
		5.1.3 リーダーと従業員のエンゲージメント	5.1.3.1	さまざまな従業員のエンゲージメントレベルを説明します。
		5.1.4 リーンのリーダーシップとゲンバ	5.1.4.1	リーンのリーダーシップとゲンバを定義し、チームの効果的な支援方法を示します。
		5.1.5 全レベルでの導き	5.1.5.1	Bill Hybels の' Courageous Leadership'より、「全レベルでの導き」モデルの自己リーダーシップを説明します。
			5.1.5.2	Bill Hybels の' Courageous Leadership'より、「全レベルでの導き」モデルの上方導きを説明します。
			5.1.5.3	Bill Hybels の' Courageous Leadership'より、「全レベルでの導き」モデルの水平導きを説明します。
			5.1.5.4	Bill Hybels の' Courageous Leadership'より、「全レベルでの導き」モデルの下方導きを説明します。
		5.1.6 変化の性質	5.1.6.1	カイゼン、カイカク、カクシンの 3 種類の変化を説明します。
		5.1.7 プロアクティブな問題解決と継続的改善	5.1.7.1	プロアクティブな問題解決と継続的改善を比較します。
		5.1.8 DMAIC モデル	5.1.8.1	DMAIC サイクルのステップを復習します。
		5.1.9 カイゼンのイベント	5.1.9.1	5 日間のカイゼンイベントについてまとめます。
	5.2 イノベーションへ向かって導く	5.2.1 イノベーションとは？	5.2.1.1	イノベーションを推進するとはどういうことかを示します。
			5.2.1.2	3 種類のイノベーションを説明します。
		5.2.2 コラボレーションのパラドックス	5.2.2.1	コラボレーションのパラドックスを示します。
		5.2.3 イノベーションのパラドックス	5.2.3.1	イノベーションのパラドックスを示します。
		5.2.4 イノベーションを導く	5.2.4.1	イノベーションを導く 2 つのステップについてまとめます。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		5.2.5 イノベーションのスペクトラム	5.2.5.1	イノベーションのスペクトラムについて説明します。
		5.2.6 コラボレーションのリーダーシップのパラドックス	5.2.6.1	イノベーションのスペクトラムのパラドックス 1 と 2 を説明します。
			5.2.6.2	イノベーションのスペクトラムのパラドックス 3 と 4 を説明します。
			5.2.6.3	イノベーションのスペクトラムのパラドックス 5 と 6 を説明します。
	5.3 組織の耐障害弾力性と持続可能性を向上させる	5.3.1 耐脆弱性	5.3.1.1	耐脆弱性の例を示します。
		5.3.2 対応から学習へ	5.3.2.1	災害復旧、対障害弾力性、耐脆弱性を定義し比較します。
		5.3.3 DevOps のプラクティスと災害復旧	5.3.3.1	DevOps のプラクティスが災害復旧をどう支援するか説明します。
		5.3.4 DevOps のプラクティスと対障害弾力性	5.3.4.1	DevOps のプラクティスが対障害弾力性をどう支援するか説明します。
		5.3.5 DevOps のプラクティスと耐脆弱性	5.3.5.1	DevOps のプラクティスが耐脆弱性をどう支援するか説明します。
		5.3.6 学習のツールとして失敗を受け入れる	5.3.6.1	Netflix が Chaos Monkey と Chaos Kong で成功したプラクティスを、耐脆弱性の達成方法の例として、ファンダメンタルの復習をします。

6. 出題比率

リーダーシップ試験は、以下の**5つ**のセクションで構成されています。

カテゴリ	内　容	出題比率（%）
1.0	DevOps のリーダーシップの紹介	5.0%
2.0	価値の提供へ向けた DevOps の変革の明確化と調整	15.0%
3.0	DevOps の変革の計画とアプローチ	15.0%
4.0	DevOps のフルスタックのエンゲージと実行	50.0%
5.0	DevOps の変革の結果検証と維持	15.0%
	計	**100.0%**

7. 参考情報

1. APMG and Smith, Richard et al. *The Effective Change Manager's Handbook.* London, UK: Kogan Page. 2014, page 196.

2. AXELOS. ITIL Continual Service Improvement Model, 2011. https://www.axelos.com

3. Basiri, Ali, Lorin Hochstein, Abhijit Thosar, and Casey Rosenthal. "*Chaos Engineering Upgraded," The Netflix Tech Blog*, September 24, 2015. Accessed May 22, 2018. https://medium.com/netflix-techblog/chaos-engineering-upgraded-878d341f15fa

4. Bass, Len; Weber, Ingo; Zhu, Liming (2015). *DevOps: A Software Architect's Perspective*. ISBN 978-0134049847.

5. Baukes, Mike. *"The Best DevOps Resources Online," UpGuard* (blog), April 4, 2017. Accessed May 8, 2018. https://www.upguard.com/blog/devops-resources-online

6. Beckhard, Richard, and Harris, Reuben T. *Organizational Transitions*. Massachusetts: Addison-Wesley Publishing Company, 1987.

7. Beedle, Mike, Alistair Cockburn, Ward Cunningham, Martin Fowler, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al., n.d. "Manifesto for Agile Software Development." *agilemanifesto*. Accessed May 9, 2018. http://agilemanifesto.org/

8. Bhargava, Rajat. *"Is DevOps a Title?" DevOps.com* (blog), March 20, 2014. Accessed May 8, 2018. https://devops.com/is-devops-a-title/

9. Birrell, N.D., Ould, M.A. "*A Practical Handbook for Software Development"*. Cambridge, UK: Cambridge University Press, 1985. I(SBN 0-521-25462-0).

10. Blazeclan, Gurmeet. *Blazeclan Technologies* (Blog). A Guide to TCO for Open Source Software – 7 Reasons Why Open Source Is Not Free. Accessed January 13, 2019. https://24by7masti.wordpress.com/2013/01/20/a-guide-to-tco-of-opensource-software-7-reasons-why-open-source-is-not-free

11. Blessing-White. *"X Model of Employee Engagement."* GP Strategies. (Article). March 2016. Accessed January 13, 2019. https://blessingwhite.com/the-x-model-of-employee-engagement/

12. Boutelle, Jonathan. "Understanding Organizational Stakeholders for Design Success." *Boxes and Arrows*, 2004.

13. Brown, Kyle G. *"Apply the Strangler Application Pattern to Microservices Applications."* IBM. February 13, 2017. Accessed May 28, 2018. https://www.ibm.com/developerworks/cloud/library/cl-strangler-application-pattern-microservices-apps-trs/index.html

14. Cannon, David. *ITIL Service Strategy.* London: TSO (The Stationery Office), 2011.

15. Carr, Nicholas, W.W Norton and Company, n.d. "*Book Review: The Big Switch: Rewiring the World, From Edison to Google.*" Review of *The Big Switch: Rewiring the World, From Edison to Google*, by Mike Evans. *e-Zine*. Accessed May 9, 2018.http://www.cambashi.com/contentmgr/showdetails.php/id/1865/page/3

16. Carr, Nicholas. www.nicholascarr.com (Blog). Accessed January 13, 2019. http://www.nicholascarr.com/

17. Caum, Carl. "*Continuous Delivery Vs. Continuous Deployment: What's the Diff?" Puppet* (blog), August 30, 2013. Accessed May 8, 2018. https://puppet.com/blog/continuous-delivery-vs-continuous-deployment-what-s-diff

18. Ceresani, Necco. *"9 Companies You Wouldn't Expect To Be Using DevOps," XebiaLabs* (blog), September 11, 2015. Accessed May 8, 2018. https://blog.xebialabs.com/2015/09/11/9-companies-you-wouldnt-expect-to-be-using-devops/

19. Chui, Michael et al. *The Social Economy: Unlocking Value and Productivity Through Social Technologies.* McKinsey Global Institute, 2012. Accessed January 13, 2019. https://www.mckinsey.com/~/media/McKinsey/Industries/High%20Tech/Our%20Insights/The%20social%20economy/MGI_The_social_economy_Full_report

20. Cockcroft, Adrian, MesosCon. *Keynote: Cloud Trends, DevOps, and Microservices*.YouTube, 44:06. August 26, 2015. https://www.youtube.com/watch?v=c0wSmr-u5vQ

21. Cohn, Mike. *"Two Types of Authority Leaders Must Give to Self-Organizing Teams," Mountain Goat Software (blog)*, August 15, 2017. Accessed May 8, 2018. https://www.mountaingoatsoftware.com/blog/two-types-of-authority-leaders-must-give-to-self-organizing-teams

22. Collins, Jim. *Level 5 Leadership: Good to Great: Why Some Companies Make the Leap* and *Others Don't.* New York, NY: Harper Collins, 2001.

23. Complexity Labs, *Systems Thinking*. YouTube, 2015. https://www.youtube.com/watch?v=Miy9uQcwo3U&feature=youtu.be

24. Connors, Roger, et al. *The Oz Principle: Getting Results Through Individual and Organizational Accountability*. New York, N.Y: Portfolio, 2004.

25. Davis, Jennifer and Ryn Daniels. (2016). *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale.* Sebastopol, CA: O'Reilly Media. ISBN 978-149192630). 邦訳『Effective DevOps ― 4本柱による持続可能な組織文化の育て方』オライリージャパン（2018/3/24） 978-4873118352

26. DevOps Enterprise Forum 2016, n.d. *"Evaluating Organizational Models With a DevOps Lens." IT Revolution*. Accessed May 2, 2018. http://itrevolution.com/devops_enterprise_forum_guidance/

27. Earnshaw, Aliza. “Why I Hired a DevOps Engineer: Bess Sadler of Stanford University,” *Puppet* (blog), July 30, 2013. Accessed May 8, 2018. https://puppet.com/blog/why-i-hired-a-devops-engineer-bess-sadler-of-stanford-university

28. Edwards, Damon. (2013-07-23). *"Integrating DevOps tools into a Service Delivery Platform".* http://dev2ops.org/2012/07/integrating-devops-tools-into-a-service-delivery-platform-video/

29. Eoyang, Glenda. Containers, Significant Differences & Transforming Exchanges. In *Conditions for Self-Organization in Human Systems*. Ph.D. Dissertation, Union Institute, Cincinnati, Ohio, 2001.

30. Floyd, David. “Who Killed Sears? 50 Years on the Road to Ruin.” *Investopedia*. Last modified April 23, 2018. Accessed May 9, 2018. https://www.investopedia.com/news/downfall-of-sears/

31. Forsgren, Nicole, Gene Kim, Jez Humble, Alanna Brown, and Nigel Kersten. *“2017 State of DevOps Report. See the Difference DevOps Makes—and How to Get There.” Puppet*. 2017. Accessed May 8, 2018. https://puppet.com/resources/whitepaper/state-of-devops-report

32. Fowler, Martin. “*How to Break a Monolith into Microservices.”* MARTINFOWLER.COM. April 24, 2018. Accessed May 28, 2018. https://martinfowler.com/articles/break-monolith-into-microservices.html

33. Fowler, Martin. *“InfrastructureAsCode.”* MARTINFOWLER.COM. March 1, 2016. Accessed May 22, 2018. https://martinfowler.com/bliki/InfrastructureAsCode.html

34. Gilbert, Clark, and Joseph L. Bower. “Disruptive Change When Trying Harder Is Part of the Problem.” *Harvard Business Review*. May 2002. Accessed May 9, 2018. https://hbr.org/2002/05/disruptive-change-when-trying-harder-is-part-of-the-problem

35. Gleeson, Brent, and Megan Rozo. “The Silo Mentality: How to Break Down the Barriers.” Forbes. October 2, 2013. Accessed May 9, 2018. https://www.forbes.com/sites/brentgleeson/2013/10/02/the-silo-mentality-how-to-break-down-the-barriers/#5783e43a8c7e

36. Goldratt, Eliyahu, and Jeff Cox. *The Goal: A Process of Ongoing Improvement*. Massachusetts: The North River Press, 2004.
邦訳『ザ・ゴール ― 企業の究極の目的とは何か』ダイヤモンド社 (2001/5/18) 978-4478420409

37. Goodman, M. *The Iceberg Model*. Hopkinton, MA: Innovation Associates Organizational Learning. Accessed January 13, 2019. http://www.ascd.org/ASCD/pdf/journals/ed_lead/el200910_kohm_iceberg.pdf

38. Gruver, Gary, Tommy Mouser and Gene Kim (foreword). (2015). *Leading the Transformation: Applying Agile and DevOps Principles at Scale*. IT Revolution Press. ISBN 978-1942788010.
邦訳『変革の軌跡【世界で戦える会社に変わる"アジャイル・DevOps"導入の原則】』技術評論社

(2017/1/25) 978-4774186634

39. Hackman, J. Richard. *"The Psychology of Self-Management in Organizations." In Psychology and Work: Productivity, Change, and Employment*, by Pallack M.S. and Perloff R.O. Washington, DC: American Psychological Association, 1986. http://dx.doi.org/10.1037/10055-003

40. Harvey, Cynthia (2017-09-01). "10 Ways DevOps is Changing the Enterprise". https://www.datamation.com/data-center/slideshows/10-ways-devops-is-changing-enterprise-it.html

41. Hassan, Qusay F. "Demystifying Cloud Computing." *CrossTalk*, Jan/Feb 2011. Accessed May 9, 2018. https://static1.1.sqspcdn.com/static/f/702523/10181434/1294788395300/201101-Hassan.pdf

42. Hill, Linda A., et al. *Collective Genius: The Art and Practice of Leading Innovation*. Boston: Harvard Business Review Press, 2014.

43. Holman, Peggy, et al. *The Change Handbook: The Definitive Resource on Today's Best Methods for Engaging Whole Systems.* San Francisco: Berrett-Koehler Publishers, 2007, 1C2.1, Page 36-38.

44. Humble, Jez, and David Farley. *"Continuous Delivery: Anatomy of the Deployment Pipeline." informIT.* September 7, 2010. Accessed January 13, 2019. http://www.informit.com/articles/article.aspx?p=1621865&seqNum=2 https://puppet.com/blog/continuous-delivery-vs-continuous-deployment-what-s-diff

45. Humble, Jez, Joanne Molesky, and Barry O'Reilly. *Lean Enterprise—How High Performance Organizations Innovate at Scale*. California: O'Reilly Media, Inc., 2015. 邦訳『リーンエンタープライズ ― イノベーションを実現する創発的な組織づくり』オライリージャパン (2016/10/15) 978-4873117744

46. Humble, Jez, n.d. *"What Is Continuous Delivery?" Continuous Delivery.* Accessed May 28, 2018. https://continuousdelivery.com/

47. Hunter, Tim. "A Personal Reinterpretation of the Three Ways." IT Revolution. July 10, 2013. Accessed May 8, 2018. https://itrevolution.com/a-personal-reinterpretation-of-the-three-ways/

48. Hybels, Bill. *Courageous Leadership*. Grand Rapids, Mich.: Zondervan, 2002.

49. Jesuthasan, Ravin, and John Boudreau. "Thinking Through How Automation Will Affect Your Workforce." *Harvard Business Review*. April 20, 2017. Accessed May 9, 2018. https://hbr.org/2017/04/thinking-through-how-automation-will-affect-your-workforce

50. Kim, Gene and Kevin Behr (2018). *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win 5th Anniversary Edition*. IT Revolution Press. ISBN 978-194278829).

邦訳『The DevOps 逆転だ!』日経BP(2014/8/12) 978-4822285357

51. Kim, Gene, Patrick Debois, Hohn Willis, Jez Humble (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. Portland, OR: IT Revolution Press. ISBN 978-1942788003.
邦訳『The DevOps ハンドブック 理論・原則・実践のすべて』日経BP (2017/6/22) 978-4822285487

52. Kim, Gene. *"The Three Ways: The Principles Underpinning DevOps." IT Revolution*. August 22, 2012. Accessed May 8, 2018. https://itrevolution.com/the-three-ways-principles-underpinning-devops/

53. Kotter, John. *Leading Change*. Boston: Harvard Business Review Press, 2012. p. 59.
邦訳『企業変革力』日経BP (2002/4/13) 978-4822242749

54. Krstic, Natasa. "Stakeholder Management from the Business Perspective." *Megatrend Review* 11 (2014): 165-182.

55. Kumar, Ajoy. *"How 'Everything as Code' Changes Everything?" WordPress* (blog). March 9, 2016. Accessed May 28, 2018. https://cloudmatters.wordpress.com/2016/03/09/how-everything-as-code-changes-everything/

56. Kumar, Ajoy. *"Infrastructure and How 'Everything as Code' Changes Everything." BMC* (blog). August 17, 2016. Accessed May 28, 2018. https://www.bmc.com/blogs/infrastructure-everything-code-changes-everything/

57. Loukides, Mike (2012-06-07). *"What is DevOps?"* http://radar.oreilly.com/2012/06/what-is-devops.html

58. Mann, Andi. *"Four Key Vectors for Closing the DevOps Feedback Loop," DevOps.com* (blog), September 29, 2014. Accessed May 8, 2018. https://devops.com/four-key-vectors-closing-devops-feedback-loop/

59. Mayer, Chris. *"S Is for Sharing Tells the Father of DevOps, Patrick Debois,"* JAXenter. October 17, 2012. Accessed May 8, 2018. https://jaxenter.com/s-is-for-sharing-tells-the-father-of-devops-patrick-debois-105077.html

60. Mello Jr., John P., n.d. *"The Best DevOps Conferences of 2018." TechBeacon*. Accessed May 8, 2018. https://techbeacon.com/best-devops-conferences-2018

61. Meyer, Bertrand. *"Practice to Perfect: The Quality First Model." EiffelSoft*, (1997): 102–106. Accessed May 22, 2018. http://se.inf.ethz.ch/~meyer/publications/computer/quality_first.pdf

62. Middelburg, Jan-Willem. *Service Automation Framework*. Netherlands: Van Haren Publishers, 2016.

63. Mirchandani, Sanjay. *The State of DevOps in 2017: Transformational Leadership Is Key.* By Chris Cancialosi. Forbes, June 19, 2017. Accessed May 8, 2018. https://www.forbes.com/sites/chriscancialosi/2017/06/19/the-state-of-devops-in-2017-transformational-leadership-is-key/#7cb8b17c63dc

64. Mueller, Ernest. *"What Is DevOps?" the agile admin* (blog), July 24, 2017. Accessed May 8, 2018. https://theagileadmin.com/what-is-devops/

65. Newman, Lily Hay. *"How Netflix DDoS'd Itself to Help Protect the Entire Internet." WIRED*. Last modified July 28, 2017. Accessed May 22, 2018. https://www.wired.com/story/netflix-ddos-attack/

66. Nickolaisen, Neil, and Pixton, Pollyanna. *The Agile Culture: Leading Through Trust and Ownership.* Massachusetts: Addison-Wesley Publishing Company, 2014.

67. Null, Christopher, n.d. "*10 Companies Killing It at Devops." TechBeacon*. Accessed May 8, 2018. https://techbeacon.com/10-companies-killing-it-devops

68. Null, Christopher, n.d. "*Infrastructure as Code: The Engine at the Heart of DevOps."* TechBeacon. Accessed May 22, 2018. https://techbeacon.com/infrastructure-code-engine-heart-devops

69. Nygard, Michael. *"The New Normal: Embracing Failure With Netflix, The Chaos Monkey, and Chaos Kong." cognitect* (blog), March 24, 2016. Accessed May 22, 2018. http://blog.cognitect.com/blog/2016/3/24/the-new-normal-embracing-failure-with-netflix-the-chaos-monkey-and-chaos-kong

70. O' Reilly FREE media/e-books available @ https://www.oreilly.com/webops/free/

71. Panko, Riley. "*The Elusive Definition of DevOps," DevOps.com* (blog), February 14, 2017. Accessed May 8, 2018. https://devops.com/elusive-definition-devops/

72. Pink Think Tank*, Leadership, Structural & Cultural Enablers for DevOps.*

73. Rafferty, Alannah Eileen, and Mark Griffin. *"Dimensions of Transformational Leadership: Conceptual and Empirical Extensions." The Leadership Quarterly* 15, no. 3 (2004): 329–354.

74. Rivington, Mark. "*Secrets to Collaboration in DevOps." DevOps.com.* April 7, 2014. Accessed May 8, 2018. https://devops.com/secrets-collaboration-devops/

75. Schien, Edgar H. Humble *Inquiry: The Gentle Art of Asking Instead of Telling*. California: Berrett-Koehler Publishers, Inc., 2013.

76. Schien, Edgar H. *The Corporate Culture Survival Guide.* California: Jossey-Bass, 2009.

77. Schouten, Edwin. "*Cloud Computing Defined: Characteristics and Service Levels." IBM* (blog).

January 31, 2014. Accessed May 28, 2018. https://www.ibm.com/blogs/cloud-computing/2014/01/31/cloud-computing-defined-characteristics-service-levels/

78. Schwaber, Ken & *Jeff* Sutherland, "*SCRUM Guide*", http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100, 2016.

79. Senge, Peter M. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY: Doubleday/Currency, 1990.

80. Shafer, Andrew Clay, n.d. *"Eating Sushi With Andrew Clay Shafer."* Produced by Arrested DevOps. Podcast, MP3 audio, 41:10. Accessed May 9, 2018. https://www.arresteddevops.com/eating-sushi-with-andrew-clay-shafer/

81. Skelton, Matthew. "*What Team Structure Is Right for DevOps to Flourish?" Matthew Skelton* (blog), October 22, 2013. Accessed May 8, 2018. https://blog.matthewskelton.net/2013/10/22/what-team-structure-is-right-for-devops-to-flourish/

82. Stanley, Andy. *Visioneering: Your Guide for Discovering and Maintaining Personal Vision*. New York, NY: Penguin Random House LLC, 2016.

83. Steinberg, Scott. *"Twenty Surprising Mobile Stats for 2016: The Smartphone Takeover." Mobile Business Insights*. June 29, 2016. Accessed May 28, 2018. https://mobilebusinessinsights.com/2016/06/twenty-surprising-mobile-stats-for-2016-the-smartphone-takeover/

84. Sward, David S. *Measuring the Business Value of Information Technology: Practical Strategies for IT and Business Managers*. Hillsboro, Oregon: Intel Press, 2006.

85. Taylor, Twain. "*The Top 10 Thought Leaders in DevOps," Sweetcode* (blog), May 8, 2017. Accessed May 8, 2018. https://sweetcode.io/top-10-thought-leaders-devops/

86. Thompson, Lee. "*Q&A: Lee Thompson, Former Chief Technologist of E*TRADE Financial."* By Damon Edwards. *dev2ops*, September 28, 2009. Accessed May 9, 2018. http://dev2ops.org/2009/09/qa-lee-thompson-former-chief-technologist-of-etrade-financial/.

87. Vehent, Julien. (2018). *Securing DevOps: Security in the Cloud.* Manning Publications. ISBN 978-1617294136.

88. Walls, Mandy. (2013). *Building a DevOps Culture*. O'Reilly Media. ISBN 978-1-449-36893-7

89. Watkins, Michael D. *"What Is Organizational Culture? And Why Should We Care?" Harvard Business Review*. May 15, 2013. Accessed May 8, 2018. https://hbr.org/2013/05/what-is-organizational-culture.

90. Westrum, R. *"A Typology of Organisational Cultures." Qual Saf Health Care* 13, suppl. II (2004): ii22–ii27.

91. Wilsenach, Rouan, n.d. *"3 DevOps Techniques for Stress-Free Release Management." TechBeacon.* Accessed May 22, 2018. https://techbeacon.com/3-devops-techniques-stress-free-release-management

92. Womack, James P., and Daniel T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation.* New York: Free Press, 2003.

93. "*2016 State of DevOps Report." Puppet*. Accessed May 28, 2018. https://puppet.com/resources/whitepaper/2016-state-of-devops-report

94. *"Amazon." Wikipedia.* Accessed May 9, 2018.https://en.wikipedia.org/wiki/Amazon_(company)

95. *"Fordism." Wikipedia.* Accessed May 9, 2018. https://en.wikipedia.org/wiki/Fordism

96. "*Internet Usage Statistics." Internet World Stats.* Accessed May 28, 2018. http://www.internetworldstats.com/stats.htm

97. *"Kano Model." Wikipedia.* Accessed May 9, 2018. https://en.wikipedia.org/wiki/Kano_model

98. *"Microservices." Wikipedia*. Accessed May 25, 2018. https://en.wikipedia.org/wiki/Microservices

99. *"Principles of Chaos Engineering." principlesofchaos*. Last updated May 2018. Accessed May 22, 2018. http://principlesofchaos.org/

100. *"Scientific Management." The Economist*. February 9, 2009. Accessed May 9, 2018.https://www.economist.com/node/13092819

101. *"The Netflix Simian Army." The Netflix Tech Blog*, July 19, 2011. Accessed May 22, 2018. https://medium.com/netflix-techblog/the-netflix-simian-army-16e57fbab116

102. *"The Netflix Simian Army"*, The Netflix Tech Blog. https://medium.com/netflix-techblog/the-netflix-simian-army-16e57fbab116

103. *"The World Is Flat." Wikipedia*. Accessed May 9, 2018. https://en.wikipedia.org/wiki/The_World_Is_Flat

104. "*What Is the Kano Model?" kanomodel.com*. Accessed May 9, 2018. https://www.kanomodel.com/.

8. 追加資料

1. *2017* State of DevOps Report – Puppet Labs. https://puppet.com/resources/whitepaper/state-of-devops-report

2. "*Branching* (Version Control)." *Wikipedia.com*. Accessed January 13, 2019. https://en.wikipedia.org/wiki/Branching_(version_control)

3. *Cunliff*, Ed. "Connecting Systems Thinking and Action." *The Systems Thinker* (Article). *n.d.* Accessed January 13, 20 19. https://thesystemsthinker.com/connecting-systems-thinking-and-action/

4. *Cohn*, Mike. "Two Types of Authority Leaders Must Give to Self-organizing Teams." *Mountain Goat Software* (Article). August 15, 2017. Accessed January 13, 20 19. https://www.mountaingoatsoftware.com/blog/two-types-of-authority-leaders-must-give-to-self-organizing-teams

5. *Fowler*, Martin. "Branch by Abstraction." *MartinFowler.com* (Article). January 7, 2014. Accessed January 13, 2019. https://martinfowler.com/bliki/BranchByAbstraction.html

6. *Fowler*, Martin. "Feature Branch." *MartinFowler.com* (Article). September 3, 2009. Accessed January 13, 2019. https://martinfowler.com/bliki/FeatureBranch.html

7. *Fowler*, Martin. "Feature Toggle." *MartinFowler.com* (Article). October 29, 2010. Accessed January 13, 2019. https://martinfowler.com/bliki/FeatureToggle.html

8. Glesson, Brent. "The Silo Mentiality: How to Break Down the Barriers." *Forbes.com* (Blog). October 2, 2013. Accessed January 13, 2019. https://www.forbes.com/sites/brentgleeson/2013/10/02/the-silo-mentality-how-to-break-down-the-barriers/#7505326e8c7e

9. *Nayar*, Vineet. Three Differences Between Managers and Leaders. *Harvard Business Review*. https://hbr.org/2013/08/tests-of-a-leadership-transiti

10. *Pink* Think Tank, 2017 White Paper – *Critical Success Factors for DevOps.*

11. *Pink* Think Tank, 2017 White Paper #2 – *Leadership, Structural & Cultural Enablers for DevOps*.

12. *Newman*, Lily H. "How Netflix DDOS'd Itself To Help Protect The Entire Internet." *Wired.com* (Blog). July 28, 2017. Accessed January 13, 2019. https://www.wired.com/story/netflix-ddos-attack/

13. *Nygard*, Michael. "The New Normal: Embracing Failure With Netflix, The Chaos Monkey, And Chaos Kong." *Cognitect.com* (Blog). March 24, 2016. January 13, 2019. http://blog.cognitect.com/blog/2016/3/24/the-new-normal-embracing-failure-with-netflix-the-chaos-monkey-and-chaos-kong

14. “The Elusive Definition of DevOps.” *DevOps.com* (Blog). February 14, 2017. Accessed January 13, 2019. https://devops.com/elusive-definition-devops/

15. *Westrum*, R. “A Typology of Organisational Cultures.” *Qual Saf Health Care* 2004;13(Suppl II):ii22–ii27. doi: 10.1136/qshc.2003.009522 Accessed January 13, 2019. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/pdf/v013p0ii22.pdf

16. “*You’re* Doing it Wrong.” *Truckbaseddevelopment.com* (Article). *n.d.* Accessed January 13, 2019. https://trunkbaseddevelopment.com/youre-doing-it-wrong/

9. 用語集 (50音順)

用語	定義
20/20 モデル	Pink Elephant の「20/20 変革モデル」は、広く受け入れられている組織変革マネジメントの原則を用いて、企業のIT環境で真の組織変革を実現するために必要なステップを定義している。
3つの道	DevOpsの三大原則であるフロー、フィードバック、継続的な経験と学習をさす。
C.A.L.M.S（カームス）	DevOpsの価値観を示す、文化（Culture）、自動化（Automation）、リーン（Lean）、測定（Measurement）、共有（Sharing）の頭文字をとったもの。
クリテイカル・サクセス・ファクター(CSF)	重要成功要因（Critical Success Factor）。チーム、プロセス、成果など、成功するために必要なもの。
クリティカル・ツウ・クオリテイ(CTQ)	重要品質要因（Critical To Quality）。リーンの「顧客の声（VOC）」では、顧客の「欲しいものリスト」に並ぶ多くの項目が、顧客に価値を提供するうえで絶対に重要なものになる。
DevOps	ITIL、リーン、アジャイルなどの既存のITベストプラクティスを、自動化と継続的デリバリをサポートし、コラボレーションと学習の文化を奨励する開発および運用アプローチへと進化させることを表現するために作られた語。ITがこれまで以上により良く、より速く、より安くビジネスバリューを提供できるようにすることをめざす。
DevOps チーム	ある一面から見ると、DevOpsは、ビジネスに一貫して価値を提供するために、既存の垂直型のITの技術領域や部門を、プロダクトの作成・デプロイ・管理を行う「プロダクトチーム」と、必要になるさまざまなプラットフォームを構築・管理する「プラットフォームチーム」に再構築する。
DMAIC / ディーマイック	シックスシグマのコンセプトで、DMAIC（定義（Define）、測定（Measure）、分析（Analyze）、改善（Improve）、管理（Control））とは、ビジネスプロセスや設計を改善し、最適化し、安定化するためのデータ駆動型の改善サイクルのことをさす。
DMAICサイクル	定義（Define）、測定（Measure）、分析（Analyze）、改善（Improve）、管理（Control）の5段階による継続的改善の指針を提供するモデル。
ITIL®	IT サービスマネジメントに関するベストプラクティスガイダンス。
ITIL サービスオーナー	特定のサービスの提供に関して説明責任を負う ITIL®の役割。†
KPI / 重要業績評価指標	重要業績評価指標（Key Performance Indicators）。傾向を示す。経時的に測定し比較することにより、全体的な傾向を示すメトリクス。
VOC / 顧客の声	顧客の声（Voice Of the Customer）。リーンから生まれた重要なツールで、顧客とは誰で、いったい何を価値と定義しているのかを理解するのに役立つ。
WIP / ワークインプロセス	進行中の作業（Work In Progress）。リーンは、特定のチームやプロセスのステップの WIP 容量には限界があるという明白な事実を明らかにしている。この WIP 制限は、「プルシステム」を定義する際に使用される。

用語	定義
アジャイル型プロジェクトマネジメント	プロジェクトマネジメントにあたって適応性のあるアプローチをとり、プロジェクトの初期に作成される要求が、プロジェクトが進むにつれ変化し、進化していくことを想定する。こうした変化に対応し、変化を組み込んでいくために、継続的なイテレーションを使用する。
アジャイル原則	アジャイル宣言を構成する12の原則。
アジャイル宣言	プロセスやツールよりも個人と対話を、 包括的なドキュメントよりも動くソフトウェアを、 契約交渉よりも顧客との協調を、 計画にしたがうことよりも変化への対応を、 価値とする。*
インシデント	ITIL®によれば、サービスの計画外の中断、またはサービスの品質の低下。†
インシデント管理	通常のサービス運用を可能な限り迅速に復旧することで、インシデントの悪影響を最小限に抑えるITIL®のプラクティス。†
ウォーターフォール型プロジェクトマネジメント	プロジェクトマネジメントに対する伝統的なアプローチ。1960 年代に初めてソフトウェアプロジェクトに適用された。このアプローチは、要求の収集から始まり、設計、構築、テスト、デプロイ、リリースと続く。作業は、各段階で左から右へ、滝（ウォーターフォール）のように、まっすぐに下へと向かって流れていく。
オペレーショナルレベルアグリーメント（OLA）	社内の IT サービスプロバイダとその他の部門との間の、サービスの提供を支援する書面による合意。サービスレベルアグリーメント（SLA）の内容を支えることを狙いとするものでなければならない。
開発チーム	各スプリントの終わりに、「完了」し、リリース判断可能なプロダクトのインクリメントのデリバリ作業を行う専門家で構成されるアジャイルのスクラムチーム。‖
価値の提供	価値は、品質、スピード、コストの 3 つの大きな要素から構成されている。DevOps は、この 3 つすべてを実現することを究極の目標としている。プロジェクトも作業するアクティビティもプロセスもすべてが、顧客に価値を提供するという大きな目標にどの程度貢献しているかに基づいて評価されなければならない。最終的に、何が顧客にとって価値があるかを決めるのは顧客である。
関係管理	関係管理は、プロバイダのプロダクトやサービスに対するビジネスの需要を刺激し、表面化し、形にすることにより、そのプロダクトやサービスのもつ潜在的な価値を確実に捕捉し、最適化できるようにする。
カンバン /Kanban(看板)	カンバンは、1940 年代にリーン生産方式の初期段階の 1 つの進化形態として登場し、組立ラインの作業員が部品やその他の作業の需要を下流のパートナーに通知するための方法を提供した。これにより、透明性が高まりコミュニケーションが促進され、プロセスの標準化につながった。
「完了」の定義	アジャイルスクラムでは、プロダクトインクリメントを「完了」とみなすために満たさなければならない基準のリスト。DevOpsでは、プロダクトにエラーがなく、本番環境にデプロイされ、顧客にリリースされて使用されており、価値を提供していることを意味する。
基軸（トゥルーノース）	前進し意思決定を行う際に、羅針盤が常に指し示すべき場所。シンプルでわかりやすい方法で確立され、明確でまとめやすいものでなければならない。

用語	定義
基軸価値観 / トゥルーノース・バリュー	基軸価値観は、前進し意思決定を行う際に、羅針盤が常に指し示すべき場所である。シンプルでわかりやすい方法で確立され、明確でまとめやすいものでなければならない。
技術的負債 / テクニカルデット	最善のソリューションではなく、安易なソリューションを絶えず実装していった末にできてくる、複雑なワークアラウンドや手戻りが積み重なったもの。
機能テスト	ユニットテスト、APIテスト、統合テスト、システムテストなど、プロダクトが動作するために必要な機能のテスト。
継続的インテグレーション / コンテニュアス・インテグレーション	1日を通して、すべての開発者の作業結果を共有のメインライン（コードリポジトリやメインコードトランク）にマージするプラクティス。継続的インテグレーションは、自動化された継続的デリバリプロセスの中で、主にビルド段階をカバーする。通常の場合、開発環境内でのコードの統合、ビルド、テストに利用される。
継継続的改善 / コンテニュアス・インプルーブメント	製品およびサービスの効果的な管理に関与するすべての要素の継続的な識別と、改善を通して、変化していくビジネスニーズに組織のプラクティスとサービスを整合させるITIL®のプラクティス。†
継続的テスト/ ファンクショナルテスト	デプロイメントパイプラインの各段階で自動テストを実行し、各段階で即座にフィードバックを返すことにより、リスクを軽減する。自動化された継続的テストは、継続的インテグレーションと継続的デリバリの重要なコンポーネントである。これにより、コードと環境が適切に動作し、デプロイ可能な状態を維持できるようになる。
継続的デプロイメント（展開）	継続的デリバリのコンセプトを拡張したもので、自動テストに合格したすべての変更が自動的に本番に投入される。継続的デリバリで以前は手作業で実施されていたステップを自動化し、1日に複数回のデプロイメントを可能にする。
継続的デリバリ / コンテニュアス・デリバリ	実行可能な成果物を本番に合わせた環境に投入し、問題を検出するための自動テストを実施することで、本番までのライフサイクルを通して、コードを常に迅速かつ安全にデプロイできるように設計された一連のプラクティス。
ゲンバ	リーンの用語で、仕事をする場所をさす。リーンでは、リーダーやマネージャは、毎日最低1回は「ゲンバ歩き」をするべきだとしている。実際に作業をする場所に行って、人が実際に何をしているのかを知らなければならない。
コンウェイの法則	「システムを設計する組織は、その組織によく似た構造の設計をする」。‡
コンテナ化	ランタイム環境全体を1つのパッケージつまり「コンテナ」にバンドルすることで、アプリケーションプラットフォームとその依存性、OSディストリビューションや基礎となるインフラの違いを抽象化できるようにすること。
混乱の壁	開発と運用の間に存在する仮想的な壁。IT の世界では、開発と運用がまったく異なる課題や目標をもって動いていることが多い。 この語は、開発が運用に向かって「壁越しに投げ渡す」という感覚からきている。
サービスオーナー	特定サービスの成功に全面的な説明責任を負う人。
サービス構成管理 / サービスコンフィグレーションマネジメント	サービスの構成とそれをサポートする構成アイテムに関する正確で信頼できる情報を、必要なときに必要なところで利用できるようにする ITIL®のプラクティス。†

用語	定義
サービス指向アーキテクチャ（SOA）	機能をサービスという別個の単位に分けて考えるアーキテクチャ。開発者がネットワークを介してサービスにアクセスできるようにして、ユーザがそれを組み合わせて再利用することによりアプリケーションを作り出すことができるようにする。これらのサービスとその利用者は、相互通信により、明確に定義された共有フォーマットでデータを渡したり、複数のサービス間のアクティビティを連携したりする。
サービスマネジメント	顧客にとっての価値をサービスの形で実現する、組織の専門能力の集まり。†
サービスレベルアグリーメント（SLA）	必要とされるサービスとサービスで期待されるレベルを規定するための、サービスプロバイダと顧客との間で結ばれる合意文書。†
サービスレベル管理 / サービスレベルマネジメント	サービスパフォーマンスの明確な目標値を事業に基づいて設定し、その目標値に対して、サービスの提供を適切に評価、モニタリング、管理できるようにする ITIL®のプラクティス。†
災害復旧 / デザスタリカバリー	最悪のシナリオに対応し、インシデントやサービス停止から重要なシステムを守るための手段。
サイロ	ほとんどの農場に設置されている背の高い縦長の貯蔵庫。企業に関して使用される場合には、各サイロ間の境界を越えることがほとんどない、垂直型の組織構造をさす。多くの場合、改善やコミュニケーションのすべてがサイロ内だけで行われる。
サイロメンタリティ	チームや部門が、共通のタスクを共有してはいても他のグループとは別個に活動することにより、機能とのつながりを独占したり、技術的なナレッジをチームや部門内でのみ共有したりするような場合に生み出される。
システム思考	機能を、境界によって定義され、部分の合計を超える、より大きなシステムの相互に関連し相互に依存する一部分として理解すること。
自動化 / オートメーション	電子機器のような高度に自動化された手段によってプロセスを操作または制御し、人間の介入を最小限に抑える技術、手法、またはシステム。
シフトレフト	開発プロセスの早期に前倒しで品質を作りこむことにより、課題を早期に検出して解決し、欠陥やエラーが本番稼働に影響を与えないようにする。
重要品質要因（CTQ）	リーンの原則によれば、優先順位をつけて重点的に取り組むべき価値をもつ項目。
出荷判断可能なプロダクトインクリメント	機能要件と非機能要件の両方に関する受け入れ基準と、合意された「完了の定義」に従って完了したスプリントバックログ項目で構成されるスクラム成果物。
処理時間	実際にプロダクトやサービスを作るのにかかった時間の合計。
スクラム	さまざまなプロセス、ツール、技法を利用できる適応性をもつアジャイルプロセスのフレームワーク。反復的な方法を採用することによりプロダクトの開発効率を高め、その結果として、可能な限り最高の品質をもつ成果物のリリース頻度を上げることが可能になる。
スクラム開発チーム	スクラム開発チームは、各スプリントの終わりに、「完了」し、リリース判断可能なプロダクトのインクリメントのデリバリ作業を行う専門家で構成される。
スクラムスプリント	スクラムの中心的なコンセプト。1 ヶ月以内のタイムボックスで、「完了」し、使用可能で、リリース判断可能なプロダクトのインクリメントを作成する。‖

用語	定義
スクラムマスター	スクラムの役割の 1 つで、スクラムの理解度を上げて適切に実行されるようにする。スクラムチームに、スクラムの理論やプラクティスやルールを確実に順守させることにより、この役割を果たす。‖
ステークホルダ	取り組みやプロジェクトや変更によって利益を得る、または影響を受ける個人またはグループ。
ストラングラー・アプリケーション	モノリシックアプリケーションを「絞め殺す」つまり置き換えるために使用されるマイクロサービスアプリケーション。「ストラングラーアプリケーションパターン」も参照。
ストラングラー・アプリケーションパターン	ストラングラーアプリケーションのマイクロサービスを段階的に導入していくことにより、機能の特定の部分をシームレスに置き換えていき、徐々にモノリシックアプリケーションを「絞め殺す」つまり置き換えていくこと。
スプリント	スクラムの中心的なコンセプト。1 ヶ月以内のタイムボックスで、「完了」し、使用可能で、リリース判断可能なプロダクトのインクリメントを作成する。‖
スプリント計画	スプリントで行う作業を計画するスクラムイベント。この計画は、スクラムチーム全体の共同作業で作成される。‖
スプリント実行	スプリント計画の後に始まり、スプリントレビューとレトロスペクティブの時間をスプリント全体から差し引いた時点で終了するスクラムイベント。実行に入ると、スプリントバックログ中の項目が「作業中」となり、「完了の定義」に従って完了するまで作業が行われる。
スプリントバーンダウンチャート	スクラムの目に見える管理のツールで、開発チームのペースを把握し、スプリントに入れる作業を適正な量にするために、スプリント計画セッションで使用されるもの。
スプリントバックログ	スプリント用に選択されたプロダクトバックログ項目の集まりに加えて、プロダクトインクリメントをデリバリし、スプリントの目標を達成するための計画から構成されるスクラムの成果物。‖
スプリントバックログボード	スクラムの目に見える管理のツールで、スプリントの目標を可視化し、進行中の作業項目や完了した作業項目を追跡するもの。
スプリントレトロスペクティブ（振り返り）	開発チームが終了したスプリントを振り返り、スプリントの実施方法について調整と精緻化（検査と適応）を行うための機会を提供するスクラムイベント。‖
スプリントレビュー	スプリント実行の終了時に実施されるスクラムイベントで、「完了の定義」に従って完了したすべての項目を、プロダクトオーナー、顧客、その他の主要ステークホルダに提示する。これにより、全員が最終プロダクトの受け入れに合意することができる。‖
制約理論	複雑なシステムやプロセスにおいては、その中で最大の阻害要因となっているボトルネックや制約が全体の効率や働きを定めるとする。制約理論は、組織が、システム全体を制約し、組織全体のスピードを制限している、最も遅く、最も非効率な領域を識別し、その領域に重点的に取り組むのに役立つ。
組織文化	組織内で学習し共有する前提や価値のパターン。組織内で働く人が、組織内の行動パターンを観察することにより、時間をかけて身につけていく。
対障害弾力性	あらゆる種類のインシデントやサービス停止に対応するだけでなく、それが発生しても機能を継続するための手段。

用語	定義
耐脆弱性 / レジリエンス	あらゆる種類のインシデントやサービス停止に対応し、それが発生しても機能を継続するだけでなく、それを学習と適応の機会として活用するための手段。
デイリースクラム	開発チームがアクティビティを同期させ、次の24時間の計画を作成するための15分間のスクラムのイベント。‖
デイリースクラムボード	スクラムの目に見える管理のツールで、各チームメンバが何に取り組んでいるかを可視化し、メンバの毎日の気分を示すもの。
デジタルトランスフォーメーション	すべての組織活動、プロセス、スキル、文化的な態度を含めた大幅な変革。
テスト駆動開発 (TDD)	プログラムを書く前にテストシナリオを準備しておくことにより、プログラマの目標をそのテストに合格できるプログラムを書くことにするプラクティス。
デプロイメント（展開）管理	新規または変更されたハードウェア、ソフトウェア、文書、プロセス、その他のサービスコンポーネントを稼働環境に移動させる ITIL®のプラクティス。†
デプロイメントパイプライン	現在のデリバリのプロセスを段階的にモデル化することにより、エンドツーエンドのデリバリで、ボトルネック、自動化の機会、コラボレーションの機会がないか検討することができる。
トータルサイクルタイム	プロセスの開始から終了までの合計時間。
ナレッジ管理	組織全体の情報とナレッジを効果的、効率的、便利に使用できる状態を維持し、改善していくための ITIL®のプラクティス。†
バイモーダル IT	ガートナーが提唱したコンセプト。バイモーダルとは、2つの別個ではあるが一貫した仕事のスタイル、ひとつは予測可能性に、もうひとつは探索に焦点を当てたスタイルを管理するプラクティスである。モード1（通常のビジネス）は、より予測可能でよく理解されている分野で最適化する。その一方でモード2は、新しい問題を解決するための探索的で実験的なものであり、不確実性の高い分野で最適化する。どちらもデジタルトランスフォーメーションにおいて重要な役割を果たす。
バリューストリーム	エンドツーエンドでビジネスバリューを提供する IT の全体像を示す。リーダーシップとガバナンスには、バリューストリームにおける自身の位置づけだけでなく、ストリーム全体を理解することが、不可欠である。
非機能テスト	パフォーマンステスト、セキュリティテスト、コンプライアンステスト、キャパシティテストなど、特定の結果を生み出すテストではない、システムの動作を確認するテスト。
ビジネスバリュー	サービスが顧客の期待に応えるレベル、またはそれを超えるレベル。
ビジョン	プロジェクトや取り組みや組織全体の将来の状態。組織が成功するために、そこに向かって動く必要がある長期的な（年単位の）方向性。
氷山モデル / アイスバーグモデル	氷山の大部分（80%）は水面下にあり、目に見えない。このモデルでは、DevOpsの変革を検討する際には「目に映るよりはるかに多くのもの」があることを理解してもらうために、氷山の例を用いる。
フィードバック	アウトプットを同じ部分へのインプットとして使用することにより、因果関係のつながりをもつループが形成される。

用語	定義
標準化 / スタンダライゼーション	これまではテクノロジを論じるために使用されることが多かったが、DevOps とリーンが、そのスコープを広げて、IT の価値の創造のあらゆる側面を含むようになってきている。ポートフォリオ、プロジェクト、プラクティス、プロセス、手続きといったものは、技術的に優れているばかりでなく、標準化されている必要がある。これにより、標準作業の定義が存在する環境を提供することができる。
付加価値を生まない作業	リーンによれば、排除するべきプロセス中の作業。
付加価値を生まない必要作業	リーンによれば、最小化するべきプロセス中の作業。付加価値は生み出さないが、実施しなければならない作業。
付加価値を生む作業	リーンによれば、最適化するべきプロセス中の作業。顧客が実際に経験して、その価値を認めている作業でなければならない。顧客は、その作業に対して喜んで支払いをする。
複雑さ/ コンプレクシティ	IT や DevOps において、複雑さはテクノロジに留まらない、ずっと大きいものをさす。IT は長い間、緊急性と（多くの場合）顧客からの圧力のもとで、非常に狭い範囲の局所的な部分に狙いを絞って問題を解決してきた。この「迅速な修正」モードは、システム全体を統合することはおろか、効果的なコミュニケーションすらできない、局所的な部分やサードパーティのサプライヤの作業結果のつぎはぎをもたらしてきた。
部分最適化 / ローカル・オプティマイゼーション	個人やチームにとって最良の結果を生み出すために構成され、構築された環境。部分的な効率性を達成することも重要だが、サイロ内で部分的に最適化されたプロセス設計がどのような問題を引き起こす可能性があるかを理解しておく必要がある。
ブルーグリーン・デプロイメント	DevOps においては、2 つの同一のクラウドベースのインフラを運用して、そのひとつを「ブルー」、もうひとつを「グリーン」とするコンセプト。ゼロダウンタイムのデプロイを推進するために、ブルーを現状の本番環境とし、グリーンをテスト環境として、新規の変更や更新を組み込んで本番レベルのテストを実行する。テスト環境が安定したら、本番環境を切り替え、このプロセスを繰り返す。
フルスタック	重要成功要因である、適切な人による適切な文化の創造、適切なプロセスとプラクティスの導入、その文化とプラクティスにテクノロジと自動化を加えて合理化と加速化を図る、という DevOps の実践における 3 つの重要な側面を表わす。
フロー	人や情報やプロダクトがプロセスの中をどのように動くか。
プルシステム	顧客の需要に基づいてプロセスを発生させてプロダクトやサービスを引き出すシステム。対照的に、需要予測に基づいてプロセスを発生させてプロダクトやサービスを押し出すのがプッシュシステムである。プルシステムにより、プロセスのムダが削減される。
プロセス	特定の目的を達成するために設計され構造化された一連の活動。1 つまたは複数の定義されたインプットを、定義されたアウトプットに変換する。
プロダクトオーナー	スクラムの役割の 1 つで、プロダクトバックログの作成と維持を担当する。顧客と常にコミュニケーションをとり、チームとコラボレーションを行う。
プロダクトオーナー（スクラムチーム）	スクラムのプロダクトオーナーは、開発チームの作業の成果であるプロダクトの価値を最大化する責任があり、プロダクトバックログの唯一の管理責任者である。

用語	定義
プロダクトバックログ	プロダクトに必要とされる可能性のあるすべてのものを順番に並べたリストからなるスクラムの成果物。プロダクトに変更を加えるための要件は、ここからしか発生しない。‖
プロダクトバックログ（スクラムチーム）	プロダクトに必要とされることがわかっているすべてのものを順番に並べたリスト。プロダクトに変更を加えるための要件は、ここからしか発生しない。
プロダクトバックログボード	スクラムの目に見える管理のツールで、プロジェクトの目標を可視化し、バックログの先頭に移った作業項目や完了した作業項目を追跡するもの。
変革（DevOps）	ゲームを変えること。DevOps の目的は、既存の IT のゲームのルールを変えることではなく、IT がやっているゲームの内容を完全に変えることである。DevOps の狙いのもとで、文化と人、プロセスとプラクティス、テクノロジと自動化のすべてが、完全に変革されていく。
変革的リーダーシップ	組織文化を生成的なものに変化させ、優先順位と目標の共有を推し進めて、DevOps をサポートするリーダーシップのスタイル。
変更コントロール	製品およびサービスの変更の成功数を最大にするために、変更を進めることを許可し、変更スケジュールを管理する ITIL®のプラクティス。これにより、リスクが確実に適切に評価されるようにする。†
保証	製品またはサービスが合意済みの要件を満たすことに対する確約。保証は、「サービスがどのように提供されるか」であると言いかえられ、サービスが「使用に適しているか」を判断するために使用できる。多くの場合、保証はサービス消費者のニーズに沿ったサービスレベルに関連する。これは、正式な合意に基づく場合や、広告メッセージやブランドイメージである場合がある。保証は一般的に、サービスの可用性、キャパシティ、セキュリティレベル、および継続性などの領域を対象とする。定義され合意された条件がすべて満たされている場合、サービスは受け入れ可能であることの確約、つまり「保証」を提供していると言える。†
マイクロサービスアーキテクチャ（MSA）	1 つの機能を 1 つのサービスに関連付け、ノード間でサービスを分散させながら規模を拡大していくアーキテクチャ。
ムダ	リーンが特定するプロセス内のムダ。不良・手直し、過剰処理、造りすぎなど。
ムラ	リーンが特定するプロセス内の差異や変動。量のばらつきやプロセスの結果の広がりなど。
ムリ	リーンが特定するプロセス内の過負荷や硬直化。需要の増減不能など。
メトリクス	データポイント。メトリクスとは、特定の時期に測定された一連のデータポイントをさす。「スナップショット」とも呼ばれる。
問題	ITIL®によれば、1つまたは複数のインシデントの原因または潜在的な原因。†
問題管理	インシデントの実際の原因と潜在的原因を識別し、ワークアラウンドと既知のエラーを管理することで、インシデントの発生可能性と発生した場合の影響を低減するITIL®のプラクティス。†
ユーザストーリ	ソフトウェアシステムの1つまたは複数の機能を非形式的な自然言語で記述したもの。ユーザストーリは、システムのエンドユーザやユーザの視点から書かれることが多い。

用語	定義
有用性	特定のニーズを満たすために製品またはサービスによって提供される機能。有用性は「サービスが何を行うか」であると言いかえられ、サービスが「目的に適しているか」を判断するために使用できる。サービスが有用性をもつためには、（サービス）消費者のパフォーマンスをサポートするか、または（サービス）消費者の制約を取り除くか、いずれかをしなければならない。多くのサービスは、その両方を行っている。†
リーダーシップ	リーダーシップは、人のグループを動機づけて、共通の目標を達成できるように行動させる能力である。ビジネスの場では、ビジョンと戦略で作業者や同僚を鼓舞して、組織のニーズを満たしていくことを意味する。
リードタイム	インプットからアウトプットまでの時間。インプットを受け取った時点から開始される。
リーン	フローを重視した品質システム。顧客価値の向上、ムダの排除、継続的改善をめざす。
リーンカイゼン	問題を解決するための構造化されたアプローチ。フローとプロセスを段階的に改善することを目的としたもので、組織内のあらゆる階層の人に、できるだけ、簡単かつ迅速に実行できる小さなアイデアを探すように奨励する態度や考え方を示す。カイゼンは、組織の日常の文化の一部であるべきだとする。
リリース管理 / リリースマネジメント	新規および変更されたサービスや機能を使用できるようにする ITIL®のプラクティス。†
リリースバーンダウンチャート	スクラムの目に見える管理のツールで、プロジェクト開始時の項目と、スプリントごとに完了した項目を示して、進捗状況を伝えるもの。

参 考

* Beedle, Mike, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, Jim Highsmith, Andrew Hunt, et al. “Manifesto for Agile Software Development.”, 2001. https://agilemanifesto.org/.

† AXELOS Limited. *ITIL® Foundation ITIL 4 Edition*. TSO, 2019.

‡ Conway, Melvin. “Conway’s Law.” Accessed June 25, 2019. http://www.melconway.com/Home/Conways_Law.html.

‖ Schwaber, Ken, and Jeff Sutherland. “The Scrum Guide™.” Scrum.Org and ScrumInc., 2014.

- このページは空白です。

PeopleCert

DevOps.