

Official



# PeopleCert DevOps Fundamentals

January 2021

Syllabus v1.0J



電子メール：[info@peoplecert.org](mailto:info@peoplecert.org), [www.peoplecert.org](http://www.peoplecert.org)

**Copyright © 2021 PeopleCert International Ltd.**

すべての著作権は、PeopleCert International Ltd.に帰属します。本資料のいかなる部分も、PeopleCert International Ltd.が書面で許可した場合を除き、いかなる形式、いかなる手段(電子的、コピー、記録、その他)でも複製や送信をすることはできません。本資料の複製、送信、またはいかなる目的であっても使用の許可に関するお問い合わせは、発行元までご連絡ください。

**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基づいて、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、ビジネスの目的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、自動化テクノロジーの活用です。

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、多数のテクノロジーやツール（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（次のレベルの資格）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解説しています。

## 2. PeopleCert DevOps ファンダメンタル認定

### 2.1. ファンダメンタル資格（基礎レベル）の目的

基礎レベル資格の目的は、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と応用力を持っていること、そしてDevOps環境のメンバーと共に、またはDevOps環境のメンバーの一員として効果的に仕事ができることを確認することです。ファンダメンタル資格は、この上位の**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

**ファンダメンタル**レベルは、PeopleCertが提供する**PeopleCert DevOps**資格のスキームの初級のレベルとなっており、DevOps環境の有能なメンバ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**を持つこと、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、**基本的な**レベルの知識を持ち、さまざまツールを活用し、DevOpsをしっかり理解していることを証明されます。

なお、この上位の**上級／リーダーシップ**レベルのスキルと知識は、**PeopleCert DevOps**資格のスキームの次のレベルでカバーしています。

## 3. 学習目標

**ファンダメンタル**レベルのコースで受験者が学ぶことは、DevOpsの基本的なコンセプト、用語、原則、DevOpsに使用されるツールです。それと同時に、今日の企業におけるDevOpsの必要性、スクラムの方法論、人と文化の意味するものについても学びます。また、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、テクノロジーなどについても学習します。

**ファンダメンタル**認定の**資格保有者**は、次に示すようなDevOpsの知識と理解と応用力を実証することができます。

- DevOps の歴史と必要性
- DevOps の主要なコンセプト
- DevOps のビジネスバリュー
- 「C.A.L.M.S.」「3つの道」「継続的デリバリ」そして「フルスタック」のスキームの基盤となるコンセプト
- DevOps の文化、変革的リーダーシップ、DevOps の組織構造とチーム編成
- DevOps のプラクティスの進化と、DevOps の 15 の基本プラクティス
- アジャイル型プロジェクトマネジメントとスクラムの方法論
- クラウド技術と仮想化の主要なコンセプト、デプロイメントパイプラインの自動化、および継続的デリバリのアーキテクチャ

### 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 の紹介	アジャイル型プロジェクトマネジメント
DevOps の必要性	スクラム
DevOps の主要な原則と考え方	IT サービスマネジメント (ITSM)
フルスタック – 人と文化	仮想化
フルスタック – プロセスとプラクティス	
フルスタック – テクノロジーと自動化	

## 4. ファンダメンタル試験

**ファンダメンタル**認定試験は、受験者が上記のようなDevOpsの基礎知識を理解しているかどうか、また、現代の企業において、これらの知識をどのように応用してツールやプラクティスを使いこなしていくのかを検証できるように設計されています。

**ファンダメンタル**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**2つ**のカテゴリに焦点を当てています。

- 知る
- 理解する

### 4.1. 評価アプローチ

**ファンダメンタル**認定で使用する評価アプローチは、「知る」と「理解する」の**2つの基本的な**分類に焦点を当てています。

「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、概念、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の方法

「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、

---

<sup>1</sup> 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。

単に情報を思い出すだけでなく、以下のことも含まれます。

- 事実、概念、原則の理解
- 教材の解釈（コード、図、グラフ、文章、ダイアグラム）
- 使用するプロセス、手続き、方法の正当性の理解

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

この試験には、公式の受験基準やトレーニング要件は**ありません**。

**ファンダメンタル**試験の受験は、受験者が基本的なDevOpsの用語、原則、プロセス、プラクティス、ツールについて知識と理解を実証する必要があります。そのため、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**を受講することを推奨します。

## 4.3. 試験の実施形式

**ファンダメンタル**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 40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持つ受験者の場合は、30分の追加時間が認められています。
合格基準	70%（40問中28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	なし
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**14時間から18時間**の認定トレーニングを受けることを**推奨**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	
1.0 DevOps の紹介	1.1 DevOps とは?	1.1.1 DevOps の定義	1.1.1.1	DevOps を定義し、何が DevOps で、何がそうでないかを理解します。	
		1.1.2 DevOps の歴史	1.1.2.1	DevOps の歴史における重要なできごとを挙げます。	
			1.1.2.2	DevOps ムーブメントの最前線に立つオピニオンリーダーや重要人物を挙げます。	
			1.1.2.3	現在、DevOps で成功している業界や企業の事例を挙げます。	
	1.1.3 DevOps のフルスタック			1.1.3.1	DevOps のフルスタックの3つの階層を挙げます。
				1.1.3.2	DevOps を、文化とプラクティスと自動化のフルスタックと考えるアプローチを説明します。
2.0 今こそ DevOps	2.1 ビジネス バリュー の提供の 問題	2.1.1 ビジネスバリューの 次元	2.1.1.1	「ビジネスバリュー」を定義します。	
			2.1.1.2	価値を創造するうえで、ビジネスと顧客との関係を説明します。	
			2.1.1.3	コスト、品質、スピードというビジネスバリューの3つの次元をまとめます。	

カテゴリ	トピック	スキルセット	参照	知識／作業項目
			2.1.1.4	今日の IT とそのステークホルダ、パートナー、顧客との関係に影響を与えている重要な問題をまとめます。これらの問題が、DevOps ムーブメントを推進しています。
			2.1.1.5	狩野モデルと、ビジネスバリューと IT の関係についてのその考え方を理解します。
		2.1.2 変化の外部要因	2.1.2.1	DevOps ムーブメントを押し進める組織の変化の主要な外部要因を 4 つ挙げます。
			2.1.2.2	変化の 4 大要因のそれぞれが、ビジネスバリューと IT にどのような影響を与えるかをまとめます。
		2.1.3 DevOps の組織上のゴール	2.1.3.1	DevOps の組織上のゴールを挙げます。
			2.1.3.2	「デジタルトランスフォーメーション」を定義します。
	2.2 IT バリューの提供の問題	2.2.1 IT とサイロ	2.2.1.1	組織の文化や階層が時とともに進化し、IT 内にサイロメンタリティを生み出してきたことを説明します。
			2.2.1.2	サイロ、システム思考、部分最適化が、IT がビジネスバリューを提供する能力に及ぼす影響をまとめます。
			2.2.1.3	「サイロメンタリティ」「システム思考」「混乱の壁」「部分最適化」を定義します。
		2.2.2 官僚主義の負の影響	2.2.2.1	官僚主義の負の影響の結果を挙げます。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		2.2.3 複雑なプロセスがムダを生み出す	2.2.3.1	プロセスとプラクティスが時とともにどのように進化して、IT 内に冗長性、複雑性、非効率性を生み出すに至ったのかを説明します。
		2.2.4 大きな変化が必要	2.2.4.1	プロセスの冗長性、複雑性、非効率性が、IT がビジネスバリューを提供する能力に及ぼす影響をまとめます。
			2.2.4.2	現在のシステムやツールが、統合と自動化を最適にサポートできない環境はどのように作られたのかを説明します。
			2.2.4.3	レガシーなシステムやツールへの依存が、IT がビジネスバリューを提供する能力に及ぼす影響をまとめます。
		2.2.5 IT の下降スパイラル	2.2.5.1	IT 内で起こりうる下降スパイラルの各ステージをまとめます。
			2.2.5.2	レガシーなシステムやツールへの依存が、IT がビジネスバリューを提供する能力に及ぼす影響をまとめます。
		2.2.6 IT と DevOps	2.2.6.1	「技術的負債」を定義します。
			2.2.6.2	ビジネスバリューを提供できない場合、ソリューションとしての DevOps の導入に向けて進んでいる IT が、今後どのように変わっていくかをまとめます。
			2.2.6.3	DevOps の IT のゴールを挙げ、DevOps が IT のバランスをとるように進化するために重要な属性を 2 つ挙げます。

カテゴリ	トピック	スキルセット	参照	知識／作業項目	
3.0 DevOps の原則と 考え方	3.1 C.A.L.M. S. と 3 つ の道	3.1.1 C.A.L.M.S.	3.1.1.1	「C.A.L.M.S.」に関連する価値を挙げます。	
		3.1.2 3つの道	3.1.2.1	DevOps の 3 つの道を挙げます。	
			3.1.2.2	第 1 の道：フローと DevOps にとっての重要性。	
			3.1.2.3	制約理論についてまとめます。	
			3.1.2.4	第 2 の道：フィードバックと DevOps にとっての重要性。	
			3.1.2.5	「フィードバックループ」を定義します。	
			3.1.2.6	第 3 の道：継続的改善と DevOps にとっての重要性。	
	3.2 デプロイ メントパイ プライン	3.2.1 デプロイメントパイ プラインを横断す る継続的デリバリ		3.2.1.1	「デプロイメントパイプライン」「継続的インテグレーション」「継続的デリバリ」「継続的テスト」「継続的デプロイ」を定義します。
				3.2.1.2	継続的インテグレーションと継続的デリバリと継続的デプロイの関係を説明します。
				3.2.1.3	DevOps における継続的デリバリの重要性を説明します。
			3.2.1.4	機能テストと非機能テストを比較対照します。	
			3.2.1.5	デプロイメントパイプラインを横断する継続的デリバリの利点を挙げます。	

カテゴリ	トピック	スキルセット	参照	知識／作業項目
	3.3 DevOps のスコープ	3.3.1 DevOps は人と文化を変える	3.3.1.1	DevOps を実現するカギとなる文化の価値、一連のプロセス、そしてテクノロジーについてまとめます。
			3.3.1.2	人と文化の変革者としての DevOps の重要性をまとめます。
			3.3.1.3	DevOps が人と文化を変える主な方法を挙げます。
		3.3.2 DevOps はプロセスとプラクティスを変える	3.3.2.1	プロセスとプラクティスの変革者としての DevOps の重要性をまとめます。
			3.3.2.2	DevOps がプロセスとプラクティスを変革する主な方法を挙げます。
		3.3.3 DevOps はテクノロジーと自動化を変える	3.3.3.1	テクノロジーと自動化の変革者としての DevOps の重要性をまとめます。
			3.3.3.2	DevOps がテクノロジーと自動化を変革する主な方法を挙げます。
4.0 フルスタック：人と文化	4.1 DevOps 文化の定義	4.1.1 組織文化	4.1.1.1	「組織文化」を定義します。
		4.1.2 コラボレーションの重要性	4.1.2.1	DevOps 文化におけるコラボレーションと共有について理解し、効果的なコラボレーションの特性を挙げます。
			4.1.2.2	病理学的文化と生成的文化を比較対照します。
			4.1.2.3	DevOps 文化における、経験と継続的な学習の重要性を説明します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
	4.2 変革的リーダーシップ	4.2.1 変革的リーダーシップとは？	4.2.1.1	「変革的リーダーシップ」を定義します。
			4.2.1.2	変革的リーダーシップの特性を示します。
		4.2.2 変革的リーダーシップの原則	4.2.2.1	変革的リーダーシップの理解におけるビジョンの重要性を説明します。
			4.2.2.2	変革的リーダーシップの特性を理解します。
			4.2.2.3	リーダーシップのスタイルを理解します。
		4.2.3 チーム	4.2.3.1	自己組織的チームと伝統的チームの違いを理解し、DevOps にとって自己組織的チームの重要性を説明します。
			4.2.3.2	内発的動機づけと外発的強化に関する現在の主な研究成果を示します。
	4.3 DevOps の組織構造とチーム編成	4.3.1 DevOps の組織モデル	4.3.1.1	DevOps にとっての「コンウェイの法則」の重要性を説明します。
			4.3.1.2	組織モデルの候補を比較検討し、DevOps 文化を最もうまくサポートする組織モデルを選びます。
		4.3.2 DevOps チームの進化	4.3.2.1	DevOps にとっての自己組織的チームの重要性を説明します。また、タスク指向が DevOps チーム内のコラボレーションの能力に及ぼす影響を説明します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
			4.3.2.2	DevOps との関連において「チーム」を定義します。
			4.3.2.3	DevOps のチーム編成の進化の段階を挙げます。
		4.3.3 DevOps は実践的なスペシャリストを必要とする	4.3.3.1	DevOps チーム編成のパターンとアンチパターンを挙げます。
		4.3.4 プロダクトチームとプラットフォームチームへの変化	4.3.4.1	「DevOps エンジニア」を定義します。
		4.3.5 機能横断的チームの構築	4.3.5.1	機能横断型チームの属性を挙げます。
			4.3.5.2	DevOps チーム内での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を説明します。
			4.3.5.3	役割ごとのサイロ構造のチームと、プロダクトチームとプラットフォームチームとを比較対照します。
5.0 フルスタック：プロセスとプラクティス	5.1 DevOps プラクティスの進化	5.1.1 DevOps と他の方法論	5.1.1.1	IT サービスマネジメント (ITSM：IT Service Management)を定義します。
			5.1.1.2	ITIL を、採用して適応させるべき ITSM フレームワークとして定義します。
			5.1.1.3	ITIL のサービスバリューシステムをまとめます。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
			5.1.1.4	リーンを、フローを重視した品質システムとして定義し、その原則を示します。
			5.1.1.5	アジャイル宣言の価値についてまとめます。
			5.1.1.6	アジャイルソフトウェアの12の原則を挙げます。
			5.1.1.7	DevOps の進化にとってのITSM、ITIL、リーン、アジャイルの重要性を説明します。
	5.2 DevOps の 15 の基本プラクティス	5.2.1 顧客の声	5.2.1.1	「顧客の声」を定義し、DevOps にとってのその重要性を説明します。
			5.2.1.2	有用性と保証のコンセプトを比較対照します。
			5.2.1.3	「重要品質要因（CTQ：Critical To Quality）」を定義します。
		5.2.2 関係管理	5.2.2.1	関係管理プラクティスの目的をまとめ、DevOps にとってのその重要性を説明します。
			5.2.2.2	事業関係マネージャ（BRM：Business Relationship Manager）の役割についてまとめます。
		5.2.3 リーンプロセス最適化	5.2.3.1	DevOps にとってのリーンプロセス最適化のプラクティスの重要性を説明します。
			5.2.3.2	リーンの3種類の作業を比較対照します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
			5.2.3.3	リーンが特定するムダの 8 つのカテゴリを挙げます。
		5.2.4 バリューストリームマッピング	5.2.4.1	DevOps にとってのバリューストリームマッピングのプラクティスの重要性を説明します。
			5.2.4.2	バリューストリームマッピングのプラクティスの主要構成要素をまとめます。
		5.2.5 ナレッジ管理	5.2.5.1	DevOps における「ナレッジ管理」を定義し、DevOps にとってのその重要性を説明します。
			5.2.5.2	DIKW モデルに沿って、データと情報とナレッジと知恵の関係をまとめます。
			5.2.5.3	DevOps におけるナレッジ管理システムの例を挙げます。
		5.2.6 目に見える管理	5.2.6.1	DevOps にとっての目に見える管理のプラクティスの重要性を説明します。
			5.2.6.2	目に見える管理が、フローの管理と WIP 制限の実現にどのように役立つかを説明します。
			5.2.6.3	カンバンボードの主要属性を挙げます。
			5.2.6.4	リーンにおける「プル」を定義します。
			5.2.6.5	「WIP (Work in Progress : 進行中の作業) 制限」を定義します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		5.2.7 アジャイル型プログラムマネジメントとスクラム	5.2.7.1	DevOps にとってのアジャイル型プログラムマネジメントとスクラムのプラクティスの重要性を説明します。
			5.2.7.2	「スクラム」を定義します。
			5.2.7.3	スクラムの 3 本柱についてまとめます。
			5.2.7.4	アジャイルスクラムのプラクティスの主要構成要素をまとめます。
		5.2.8 シフトレフトテスト	5.2.8.1	DevOps にとってのシフトレフトテストのプラクティスの重要性を説明します。
			5.2.8.2	「シフトレフトテスト」を定義します。
			5.2.8.3	品質第一アプローチの利点を説明します。
			5.2.8.4	「テスト駆動開発」を定義します。
		5.2.9 変更コントロール	5.2.9.1	DevOps にとっての変更コントロールのプラクティスの重要性を説明します。
			5.2.9.2	ITIL にしたがって管理しなければならない 3 つの変更の種類を比較対照します。
			5.2.9.3	伝統的な ITIL の変更コントロールと承認プロセスについてまとめます。
			5.2.9.4	DevOps 環境において、変更コントロールをどのように適応させるべきかをまとめます。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		5.2.10 サービス構成管理	5.2.10.1	DevOps における「サービス構成管理」を定義し、DevOps にとってのその重要性を説明します。
			5.2.10.2	「IAC (Infrastructure As Code)」を定義します。
		5.2.11 リリース管理とデプロイメント管理	5.2.11.1	DevOps における「リリース管理とデプロイメント管理」を定義し、DevOps にとってのその重要性を説明します。
			5.2.11.2	DevOps におけるリリース管理とデプロイメント管理の主要技法を挙げます。
			5.2.11.3	「ブルーグリーンデプロイメント」を定義します。
		5.2.12 インシデント管理	5.2.12.1	DevOps にとってのインシデント管理のプラクティスの重要性を説明します。
			5.2.12.2	DevOps 環境において、インシデント管理をどのように適応させるべきかをまとめます。
			5.2.12.3	ITIL によるインシデントと問題の定義を比較対照します。
		5.2.13 問題管理とカイゼン	5.2.13.1	「カイゼン」を定義し、DevOps にとって問題管理とカイゼンの重要性を説明します。
			5.2.13.2	DevOps 環境において、問題管理をどのように適応させるべきかをまとめます。
			5.2.13.3	DMAIC サイクルのステップをまとめます。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		5.2.14 継続的改善	5.2.14.1	DevOps にとっての継続的改善のプラクティスの重要性を説明します。また、継続的改善段階のプロセスと DevOps との関係性を説明します。
			5.2.14.2	DevOps にとっての標準化の重要性を説明します。
			5.2.14.3	「継続的改善記録簿（CIR：Continual Improvement Register）」を定義し、継続的改善がどのようにフルスタック全体に方向性を与え、サポートするかを説明します。
		5.2.15 耐脆弱性	5.2.15.1	「耐脆弱性」を定義し、DevOps にとっての耐脆弱性のプラクティスの重要性を説明します。
			5.2.15.2	災害復旧と対障害弾力性と耐脆弱性の違いを理解します。
			5.2.15.3	耐脆弱性を実現する方法の例として、Netflix が Chaos Monkey と Chaos Kong で使用して成功したプラクティスをまとめます。
			5.2.15.4	「カオスエンジニアリング」を定義します。
6.0 フルスタック：テクノロジーと自動化	6.1 デプロイメントパイプラインの自動化	6.1.1 自動化とフルスタック	6.1.1.1	DevOps にとっての、ツールと自動化の重要性を説明します。
		6.1.2 デプロイメントパイプラインの自動化	6.1.2.1	ツールチェーンをどのように利用すればデプロイメントパイプラインの自動化に役立つかを説明します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
		6.1.3 DevOps ツールチェーンのツール	6.1.3.1	DevOps ツールチェーンにおいて、デプロイメントパイプラインの各段階で利用できるツールの例を挙げます。
			6.1.3.2	DevOps ツールチェーンの代表的な要素を挙げます。
		6.1.4 DevOps の自動化の段階	6.1.4.1	自動化と継続的デリバリの導入を行うとき、組織が経ることになる成熟度の各段階をまとめます。
	6.2 クラウド技術と仮想化	6.2.1 クラウド技術とクラウドコンピューティングの出現	6.2.1.1	クラウド技術と仮想化サービスと DevOps の関係を説明します。
			6.2.1.2	クラウドコンピューティングの基本特性を挙げます。
			6.2.1.3	クラウド技術と仮想化サービスと DevOps の関係を説明します。
		6.2.2 クラウドコンピューティングのデプロイモデルとサービスモデル	6.2.2.1	クラウドコンピューティングの3つのデプロイモデルを比較対照します。
			6.2.2.2	クラウド技術の歴史におけるキープレイヤを挙げます。
			6.2.2.3	伝統的なオンプレミスから SaaS (Software as a Service) に至るクラウドコンピューティングのサービスモデルの進化を説明します。
			6.2.2.4	「XaaS (Everything as a Service)」と「EaC (Everything as Code)」を定義します。

カテゴリ	トピック	スキルセット	参照	知識／作業項目
	6.3 継続的デリバリのためのアーキテクチャ	6.3.1 プロダクトチームとプラットフォームチーム	6.3.1.1	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説明します。
		6.3.2 コンテナ化とマイクロサービス	6.3.2.1	「コンテナ化」を定義します。
			6.3.2.2	DevOps にとってのコンテナ化の重要性を説明します。
			6.3.2.3	モノリシックアーキテクチャとマイクロサービスアーキテクチャを比較対照します。
			6.3.2.4	マイクロサービスの利点を挙げます。
		6.3.3 API (Application Programming Interface)	6.3.3.1	「ストラングラーアプリケーション」と「ストラングラーアプリケーションパターン」を定義します。

## 6. 出題比率

ファンダメンタル試験は、以下の6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 の紹介	5.0%
2.0	今こそ必要な DevOps	10.0%
3.0	DevOps の原則と考え方	15.0%

4.0	フルスタック — 人と文化	20.0%
5.0	フルスタック — プロセスとプラクティス	30.0%
6.0	フルスタック — テクノロジーと自動化	20.0%
	<b>計</b>	<b>100.0%</b>

## 7. 参考情報

1. 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>
2. Bass, Len; Weber, Ingo; Zhu, Liming (2015). *DevOps: A Software Architect's Perspective*. ISBN 978-0134049847.
3. Baukes, Mike. “*The Best DevOps Resources Online*,” *UpGuard* (blog), April 4, 2017. Accessed May 8, 2018. <https://www.upguard.com/blog/devops-resources-online>
4. 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/>
5. Bhargava, Rajat. “*Is DevOps a Title?*” *DevOps.com* (blog), March 20, 2014. Accessed May 8, 2018. <https://devops.com/is-devops-a-title/>
6. Birrell, N.D., Ould, M.A. “*A Practical Handbook for Software Development*”. Cambridge, UK: Cambridge University Press, 1985. I(SBN 0-521-25462-0).
7. 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>
8. Cannon, David. *ITIL Service Strategy*. London: TSO (The Stationery Office), 2011.
9. 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>
10. Caum, Carl. “*Continuous Delivery Vs. Continuous Deployment: What's the Diff?*” *Puppet* (blog), August 30, 2013. Accessed May 8, 2018.
11. 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/>
12. Cockcroft, Adrian, MesosCon. *Keynote: Cloud Trends, DevOps, and Microservices*. YouTube, 44:06. August 26, 2015.

- <https://www.youtube.com/watch?v=c0wSmr-u5vQ>
13. 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.mountangoatsoftware.com/blog/two-types-of-authority-leaders-must-give-to-self-organizing-teams>
  14. Conway, Melvin. “Conway’s Law.” Accessed June 25, 2019. [http://www.melconway.com/Home/Conways\\_Law.html](http://www.melconway.com/Home/Conways_Law.html)
  15. 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
  16. 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/](http://itrevolution.com/devops_enterprise_forum_guidance/)
  17. 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>
  18. 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/>
  19. 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/>
  20. 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>
  21. 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>
  22. Fowler, Martin. “InfrastructureAsCode.” MARTINFOWLER.COM. March 1, 2016. Accessed May 22, 2018. <https://martinfowler.com/bliki/InfrastructureAsCode.html>

23. 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>
24. 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>
25. 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
26. 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【世界で戦える会社が変わる"アジャイル・DevOps"導入の原則】』技術評論社 (2017/1/25) 978-4774186634
27. 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.
28. 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>
29. 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>
30. Humble, Jez, and David Farley. "Continuous Delivery: Anatomy of the Deployment Pipeline." *informIT*. September 7, 2010. Accessed May 8, 2018. <http://www.informit.com/articles/article.aspx?p=1621865&seqNum=2>
31. 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

32. Humble, Jez, n.d. "What Is Continuous Delivery?" *Continuous Delivery*. Accessed May 28, 2018. <https://continuousdelivery.com/>
33. 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/>
34. 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>
35. 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
36. 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
37. 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/>
38. 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/>
39. 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/>
40. Loukides, Mike (2012-06-07). "What is DevOps?" <http://radar.oreilly.com/2012/06/what-is-devops.html>
41. 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/>
42. 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>
43. Mello Jr., John P., n.d. *"The Best DevOps Conferences of 2018."* TechBeacon. Accessed May 8, 2018. <https://techbeacon.com/best-devops-conferences-2018>
  44. 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](http://se.inf.ethz.ch/~meyer/publications/computer/quality_first.pdf)
  45. Middelburg, Jan-Willem. *Service Automation Framework*. Netherlands: Van Haren Publishers, 2016.
  46. 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>
  47. Mueller, Ernest. *"What Is DevOps?" the agile admin* (blog), July 24, 2017. Accessed May 8, 2018. <https://theagileadmin.com/what-is-devops/>
  48. 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/>
  49. Null, Christopher, n.d. *"10 Companies Killing It at DevOps."* TechBeacon. Accessed May 8, 2018. <https://techbeacon.com/10-companies-killing-it-devops>
  50. 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>
  51. 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>
  52. O' Reilly FREE media/e-books available @ <https://www.oreilly.com/webops/free/>
  53. Panko, Riley. *"The Elusive Definition of DevOps," DevOps.com* (blog), February 14, 2017. Accessed May 8, 2018. <https://devops.com/elusive-definition-devops/>
  54. Rafferty, Alannah Eileen, and Mark Griffin. *"Dimensions of Transformational Leadership: Conceptual and Empirical Extensions."* *The Leadership Quarterly* 15, no. 3 (2004): 329–354.

55. Rivington, Mark. “*Secrets to Collaboration in DevOps.*” *DevOps.com*. April 7, 2014. Accessed May 8, 2018. <https://devops.com/secrets-collaboration-devops/>
56. Schien, Edgar H. Humble *Inquiry: The Gentle Art of Asking Instead of Telling*. California: Berrett-Koehler Publishers, Inc., 2013.
57. Schien, Edgar H. *The Corporate Culture Survival Guide*. California: Jossey-Bass, 2009.
58. 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/>
59. Schwaber, Ken & Jeff Sutherland, “*SCRUM Guide*”, <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>, 2016.
60. 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/>
61. 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/>
62. Steinberg, Scott. “*Twenty Surprising Mobile Stats for 2016: The Smartphone Takeover.*” *Mobile Business Insights*. June 29, 2016. Accessed May 28, 2018.
63. 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/>
64. 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/>.
65. Vehent, Julien. (2018). *Securing DevOps: Security in the Cloud*. Manning Publications. ISBN 978-1617294136.
66. Walls, Mandy. (2013). *Building a DevOps Culture*. O'Reilly Media. ISBN 978-1-449-36893-7
67. 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>.

68. Westrum, R. "A Typology of Organisational Cultures." *Qual Saf Health Care* 13, suppl. II (2004): ii22–ii27.
69. 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>
70. Womack, James P., and Daniel T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York: Free Press, 2003.
71. AXELOS Limited. *ITIL® Foundation ITIL 4 Edition*. TSO, 2019.
72. "2016 State of DevOps Report." *Puppet*. Accessed May 28, 2018. <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>
73. "Amazon." *Wikipedia*. Accessed May 9, 2018. [https://en.wikipedia.org/wiki/Amazon\\_\(company\)](https://en.wikipedia.org/wiki/Amazon_(company))
74. "Fordism." *Wikipedia*. Accessed May 9, 2018. <https://en.wikipedia.org/wiki/Fordism>
75. "Internet Usage Statistics." *Internet World Stats*. Accessed May 28, 2018. <http://www.internetworldstats.com/stats.htm>
76. "Kano Model." *Wikipedia*. Accessed May 9, 2018. [https://en.wikipedia.org/wiki/Kano\\_model](https://en.wikipedia.org/wiki/Kano_model)
77. "Microservices." *Wikipedia*. Accessed May 25, 2018. <https://en.wikipedia.org/wiki/Microservices>
78. "Principles of Chaos Engineering." *principlesofchaos*. Last updated May 2018. Accessed May 22, 2018. <http://principlesofchaos.org/>
79. "Scientific Management." *The Economist*. February 9, 2009. Accessed May 9, 2018. <https://www.economist.com/node/13092819>
80. "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>
81. "The World Is Flat." *Wikipedia*. Accessed May 9, 2018. [https://en.wikipedia.org/wiki/The\\_World\\_Is\\_Flat](https://en.wikipedia.org/wiki/The_World_Is_Flat)
82. "What Is the Kano Model?" *kanomodel.com*. Accessed May 9, 2018. <https://www.kanomodel.com/>.

## 8. 用語集 (50音順)

用語	定義
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	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。
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ーをサポートし、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。
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。
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、またはサービスに影響を与えていないサービスコンポーネントの障害。†
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。†
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ー作業を行う専門家で作成されるアジャイルのスクラムチーム。‡
カンバン	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。
基軸価値観	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。
技術的負債	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。

用語	定義
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	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。
継続的改善	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	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	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。
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。
サービス構成管理	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。†
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)	ITIL®によれば、主要なサービスの目標と責任、サービスに期待される保証と有用性を定義するための、IT サービスプロバイダと企業の間で結ばれる合意文書。†
サービスレベル管理	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	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。

用語	定義
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	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て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ついて合意された受け入れ基準にしたがい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。 <sup>11</sup>
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。 <sup>11</sup>
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。 <sup>11</sup>
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。 <sup>11</sup>
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。

用語	定義
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、スプリントゴールを実現するための計画から構成されるスクラムの成果物。 <sup>  </sup>
スプリントレトロスペクティブ（振り返り）	スクラムチームが自分たちを振り返り、次のスプリントで実行するための改善計画を作成する機会を提供するスクラムイベント。 <sup>  </sup>
スプリントレビュー	スプリントの最後に実施されるスクラムイベントで、インクリメントを検査し、必要に応じてプロダクトバックログを調整する。 <sup>  </sup>
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。
耐脆弱性	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。 <sup>  </sup>
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、プログラム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 <sup>®</sup> のプラクティス。 <sup>†</sup>
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 <sup>®</sup> のプラクティス。 <sup>†</sup>
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。

用語	定義
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。
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。
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。
部分最適化	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。
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ー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。 <sup>  </sup>
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。 <sup>  </sup>
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 <sup>®</sup> 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。 <sup>†</sup>
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。

用語	定義
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。 <sup>†</sup>
問題管理	インシデントの実際の原因と潜在的な原因を識別し、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。 <sup>†</sup>
ユーザストーリー	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。ユーザストーリーは、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。
有用性	サービスの機能要件。有用性は、目的に適したサービスの要件、「サービスは何を行うことになっているか」を表わす。
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	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。 <sup>†</sup>



- このページは空白です。

Official

PeopleCert  
DevOps.

