



生成 AI AWS を使用して でソフトウェア開発ライフサイクルを加速する

AWS 規範ガイド



AWS 規範ガイド: 生成 AI AWS を使用して でソフトウェア開発ライフサイクルを加速する

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

序章	1
目的	1
対象者	2
開発経験	3
生成 AI の使用	4
5-1 フレームワーク	5
フレームワークの概要	5
SDLC との統合	7
基本的な機能	8
プロジェクト管理	14
要件管理	17
アーキテクチャと設計	18
コラボレーション	18
DevSecOps	19
オペレーションとメンテナンス	27
AI アシスタント	29
分析とインサイト	31
ナレッジ管理	33
拡張性	34
ベストプラクティス	36
統合されたツールチェーン	36
DevSecOps パイプライン	37
コラボレーションツールとプラクティス	37
タスクの自動化	37
レビューと反復	38
プロジェクト管理プラクティス	38
ナレッジ管理	39
拡張性とカスタマイズ	39
最適化	39
データ駆動型インサイト	40
プラットフォームベースのアプローチ	40
成功の測定	41
デプロイ速度	41
コード品質	42

運用効率	42
チームの生産性と満足度	43
ビジネスへの影響	43
結論	45
リソース	45
ドキュメント履歴	47
用語集	48
#	48
A	49
B	51
C	53
D	56
E	60
F	63
G	64
H	65
I	67
L	69
M	70
O	74
P	77
Q	80
R	80
S	83
T	87
U	88
V	89
W	89
Z	90
.....	xcv

生成 AI AWS を使用して でソフトウェア開発ライフサイクルを加速する

チェタン・マクバナ、アマゾン ウェブ サービス

2025 年 4 月 ([ドキュメント履歴](#))

高品質のソフトウェアに対する需要の高まりにより、組織はソフトウェア開発ライフサイクル (SDLC) を加速する方法を常に模索しています。組織が競争力を維持するには、製品品質を維持または改善しながら、市場投入までの時間を短縮することが重要です。これらの課題に対処するには、ソフトウェア開発エクスペリエンスを進化させ、プロセスを合理化し、ソフトウェア開発チームが生産性と創造性を高める最先端のテクノロジー、方法論、プラクティスを使用する必要があります。次世代の開発エクスペリエンスの出現は、ソフトウェアの構想、構築、テスト、デプロイ方法の大きな変化を示しています。クラウドネイティブ開発、AI 駆動型オートメーション、高度なプロジェクト管理、コラボレーションツール、DevSecOps など、さまざまな機能が統合されており、SDLC の効率と有効性をまとめて強化します。

この変換の最前線にあるのは、ソフトウェアエンジニアリングにおける生成 AI の台頭です。[ガートナー](#)によると、2023 年の 5% と比較して、プラットフォームエンジニアリングチームの 40% が AI を使用して 2027 年までに SDLC のすべてのフェーズを強化します。このレポートでは、ソフトウェアエンジニアリングのリーダーは、開発プロセスにとって重要な幅広い分野で生成 AI を採用する準備をする必要があることも述べています。別のレポートでは、[McKinsey](#) の調査によると、デベロッパー速度インデックスが高い企業は収益が 4~5 倍速く、株主収益が 60% 高く、イノベーションが 55% 高いことがわかります。コード生成だけでなく生成 AI を採用することで、組織はソフトウェア開発ワークフローで新しいレベルの効率、生産性、イノベーションを引き出すことができます。これにより、手動作業を減らし、インサイトを明らかにし、人間の専門知識を高めることができます。

目的

この戦略ドキュメントでは、生成 AI を使用して SDLC を高速化するのに役立つフレームワーク、基本的な機能、ユースケース、ベストプラクティス、成功メトリクスの概要を説明します。製品の品質と効率を向上させるために、すべての開発段階で生成 AI を効果的に統合する方法について説明します。

この戦略ドキュメントは、お客様とお客様の組織が次の目標を達成するのに役立ちます。

- フレームワーク、基本的な機能、ユースケース、ベストプラクティス、成功メトリクスを実装して、生成 AI で SDLC を高速化します。

- すべての開発段階で生成 AI を効果的に統合し、製品の品質、リリース速度、開発効率を向上させます。
- プロセスを合理化し、開発チームを強化する最先端の AI テクノロジー、方法論、プラクティスを組み込むことで、次世代のソフトウェア開発に適応します。

対象者

この戦略ドキュメントは、IT リーダー、エンジニアリングマネージャー、最高技術責任者、ソフトウェア開発チームが、開発プラクティスに生成 AI を適用してソフトウェア開発のライフサイクルを加速したいと考えている方を対象としています。

ソフトウェア開発エクスペリエンスについて

ソフトウェア開発エクスペリエンスには、ソフトウェア開発ライフサイクル (SDLC) 全体で開発チームが使用する環境、ツール、プロセスが含まれます。これには、統合開発環境 (IDE)、コラボレーションプラットフォーム、テストフレームワーク、ナレッジ管理システム、デプロイパイプラインなどが含まれます。

適切に設計された開発エクスペリエンスにより、ワークフローが合理化され、手動作業が軽減され、チームが価値の高いタスクに集中できるようになり、最終的に SDLC が高速化されます。例えば、IDE、バージョン管理システム、デプロイツールをシームレスに統合することで、デベロッパーは、手動のハンドオフとコンテキスト切り替えを必要とするフラグメント化されたツールチェーンと比較して、コードの書き込み、テスト、デプロイの速度と効率が向上します。同様に、堅牢なナレッジ管理フレームワークを統合することで、チームは組織の知識、ベストプラクティス、ドキュメントに簡単にアクセスして共有できます。これにより、全体的な生産性と問題解決能力が向上します。

ソフトウェア開発の経験は、ソフトウェア開発チームの全体的なパフォーマンスと成功に直接影響します。最適ではないエクスペリエンスは、以下につながる可能性があります。

- 生産性の低下 – 非効率的なツール、複雑なワークフロー、自動化の欠如によりチームの生産性が低下し、機能や更新の提供が遅くなります。
- 技術的負債の増加 – ツールとアドホックプロセスの統合が不十分な場合、技術的負債が発生する可能性があります。時間の経過とともにソフトウェアシステムの保守とスケーリングがより困難になります。
- イノベーションの低下 – 手動の反復的なタスクに悩まされると、チームが新しいテクノロジーを探索し、イノベーションを推進する能力が制限されます。
- 侵害された品質 – 断片化されたテストとデプロイプロセスにより、ソフトウェアの欠陥や脆弱性のリスクが高まります。これは、配信されるソフトウェアの全体的な品質に悪影響を及ぼす可能性があります。

適切に設計されたソフトウェア開発エクスペリエンスに投資することで、市場投入までの時間の短縮、ソフトウェア品質の向上、ソフトウェア開発チームの満足度の向上、ビジネスの俊敏性の向上など、大きなメリットを引き出すことができます。

生成 AI によるソフトウェア開発エクスペリエンスの強化

生成 AI をソフトウェア開発ライフサイクル (SDLC) に統合することは、ソフトウェア開発チーム全体のソフトウェアソリューションの構想、設計、実装、保守方法のパラダイムシフトを表します。生成 AI は、プロジェクト管理、要件収集、設計、コーディング、テスト、デプロイ、メンテナンスなど、SDLC のすべてのフェーズに変革をもたらす可能性があります。

生成 AI を活用した開発エクスペリエンスは、製品マネージャー、デザイナー、ソリューションアーキテクト、開発者、テスター、運用担当者など、ソフトウェア開発チーム全体のインテリジェントな共同作業員として機能します。コンテキストに応じた支援を提供し、アーティファクト (ユーザーストーリー、設計モックアップ、コードスニペット、テストケースなど) を生成し、ほぼリアルタイムの提案を提供し、潜在的な問題が発生する前に予測します。この AI 拡張アプローチは、チームメンバーの認知負荷を大幅に軽減します。これにより、生成 AI がより日常的で反復的なタスクを処理しながら、高レベルの戦略的意思決定と複雑な問題解決に集中できます。

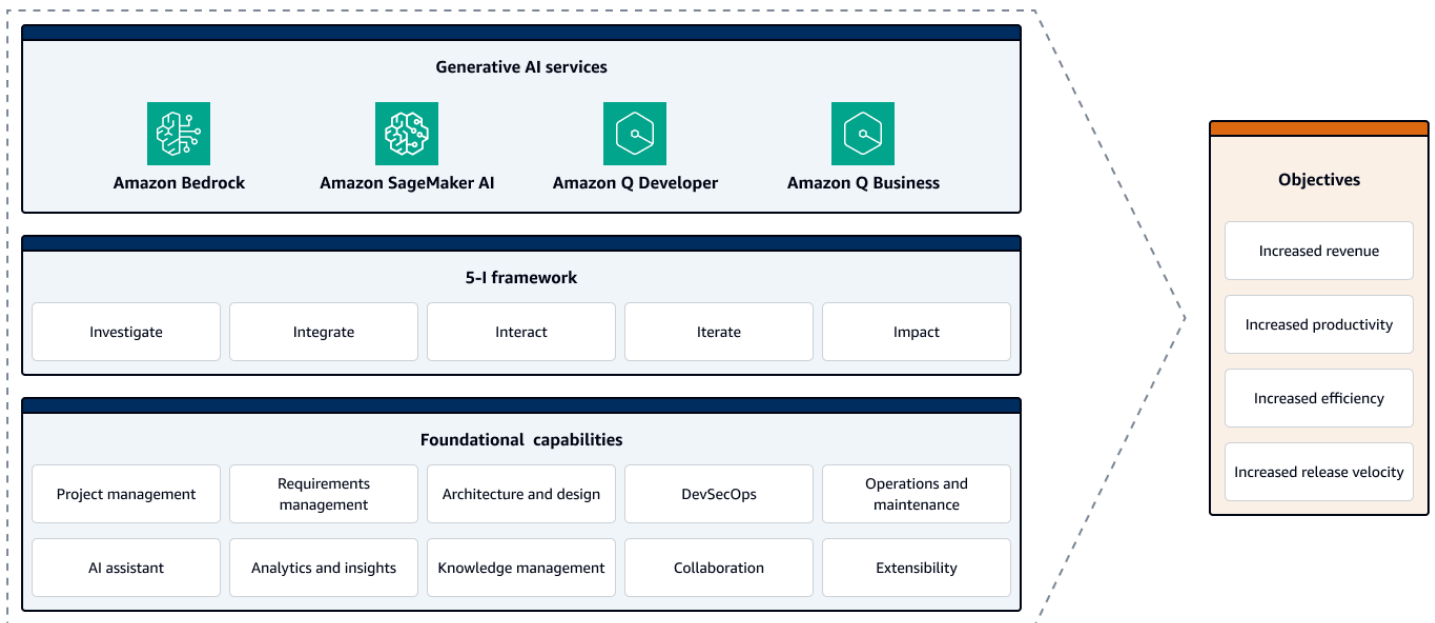
生成 AI は知識の蓄積にも役立ちます。これにより、チームメンバーは膨大なデータリポジトリから関連情報、ベストプラクティス、パターンにすばやくアクセスできます。これにより、組織全体で専門知識を効果的に民主化できます。開発ツールチェーン全体で生成 AI 機能をシームレスに統合することで、ソフトウェア開発チーム全体でより直感的で効率的で生産的な環境を作成できます。この拡張された開発エクスペリエンスにより、SDLC が高速化され、全体的な品質が向上します。また、チームメンバーが新しいアイデアやアプローチをより迅速に検討できるため、エラーを減らし、イノベーションを促進します。

生成 AI を活用した開発経験を組織で採用するには、次の主要な要素を考慮してください。

- [5-1 フレームワーク](#) – 5 つのディメンションで構成される 5-1 フレームワーク は、最新のソフトウェア開発のプロセスをナビゲートするための包括的なアプローチを提供します。SDLC のすべてのステージに生成 AI を体系的に適用するのに役立つ構造化された方法を提供します。
- [基本的な機能](#) – 最新のソフトウェア開発のディメンション全体で生成 AI の能力を最大限に活用するには、堅牢な基盤機能確立する必要があります。これらの機能は、AI を活用した開発エクスペリエンスのバックボーンを形成します。これらの機能は、SDLC 全体で生成 AI を統合して使用するのに役立ちます。

5-1 フレームワークと基本的な機能は、ソフトウェア開発エクスペリエンスを再考するための戦略を形成します。5 つのディメンションは、生成 AI を適用するための戦略的フレームワークを提供し、基盤となる機能は、この AI 主導のアプローチをサポートするために組織を準備します。 [Amazon](#)

[Bedrock](#) AWS のサービス、[Amazon SageMaker AI](#)、[Amazon Q Developer](#)、[Amazon Q Business](#) などの、ソフトウェア開発エクスペリエンスに統合できる生成 AI 機能と機能を提供します。



AI を活用したソフトウェア開発エクスペリエンスのための 5-I フレームワーク

5-I フレームワークは、ソフトウェア開発チームが生成 AI を開発プラクティスに効果的に統合するための構造化されたアプローチを提供します。SDLC 全体で生成 AI を使用するための堅牢な基盤を確立するのに役立ちます。また、生成 AI の可能性を最大限に活用するための適切な開発プラクティス、ワークフロー、考え方を設定するのにも役立ちます。

このセクションは、以下のトピックで構成されます。

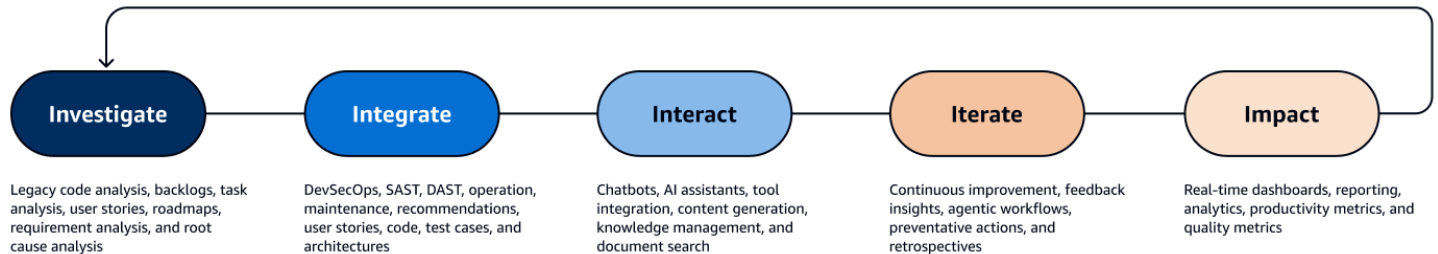
- [フレームワークの概要](#)
- [ソフトウェア開発ライフサイクルとの統合](#)

フレームワークの概要

5-I フレームワークは、調査、統合、操作、反復、影響の 5 つの主要なディメンションを中心に構築されています。各ディメンションは、生成 AI がソフトウェア開発プロセスを大幅に強化する重要な領域を表します。これらの側面にわたって生成 AI を戦略的に統合することで、フレームワークは最新のソフトウェア開発の進化するニーズに対応します。これにより、認知負荷を軽減し、創造的な可

能性を高めることができます。理想的な開発エクスペリエンスはツールだけでなく、AI があらゆる段階で人間の能力をシームレスに強化する環境をつくることであると認識しています。

次の図は、AI を活用したソフトウェア開発の 5 つの側面を示しています。各ディメンションについて、効率とイノベーションを促進するために生成 AI をどこに統合できるかを示します。



フレームワークの 5 つのディメンションは次のとおりです。

- 調査** – 生成 AI を使用して、ソフトウェア開発プロセス内のすべての分析タスクを強化します。生成 AI を使用して、要件の理解、大量のデータの処理、パターンの認識、人間の能力を超えたり、生成にかなり時間がかかるインサイトの生成を行います。これらのインサイトは、より多くの情報に基づいた意思決定を行い、改善の機会をすばやく特定し、高品質のソフトウェアをより効率的に提供するのに役立ちます。生成 AI は、SDLC 全体の分析プロセスのインテリジェントなパートナーになることができます。生成 AI を活用することで、要件の収集、レガシーコードベースの検査、製品バックログの最適化など、重要な領域に詳細な分析を適用できます。たとえば、製品所有者は生成 AI を使用して、ユーザーストーリーを作成する前にユーザージャーニーや要件を分析できます。開発チームは非効率性を発見し、既存のコードベースの最適化の機会を特定できます。DevOps エンジニアは根本原因分析を適用して、パフォーマンスの問題やセキュリティの脆弱性を迅速に診断できるため、信頼性を向上させることができます。
- 統合** – 生成 AI を統合して、SDLC 全体の幅広いタスクとプロセスを自動化します。これには、コードスニペット、テストケース、アーキテクチャ設計、ユーザーストーリー、デプロイパイプラインの自動生成が含まれます。これらの通常手動タスクを自動化することで、チームはより戦略的で革新的な作業に集中できるため、市場投入までの時間を短縮し、高品質のアプリケーションを実現できます。統合ディメンションは、AI が開発プロセスの不可欠な部分となるソフトウェア開発のパラダイムシフトを表します。ソフトウェア開発チームと協力して、生産性の向上、品質の向上、イノベーションの促進を行います。これにより、市場投入までの時間が短縮されます。ソフトウェア開発チームは、各ステップで「これは自動化できますか？」と尋ねることで、プロセスとワークフローを定期的に評価する必要があります。
- インタラクション** – 生成 AI を活用したアシスタントを使用して、さまざまなタスクやクエリで瞬時にコンテキストに応じたサポートをチームに提供します。これらのインテリジェントアシスタントは、膨大な情報リポジトリから派生する知識豊富な共同作業員として機能します。コーディング

に関する質問への回答、設計の提案、標準運用手順の説明、複雑な問題のトラブルシューティングに役立ちます。これらの AI アシスタントを開発ワークフローに統合すると、生産性が向上し、より協力的で問題解決の環境が促進されます。

- 反復 — 生成 AI を使用して、SDLC 全体でデータ駆動型の迅速な調整を可能にします。顧客のフィードバック、使用パターン、市場傾向、チームパフォーマンスメトリクスなどのソースからのデータを継続的に分析して、情報に基づいた意思決定を迅速に行うことができます。この適応性により、ソフトウェア開発は、事前に定義された静的なプロセスから、柔軟で応答性の高いアプローチに絞り込まれます。バックログの動的な優先順位付け、柔軟なリソース割り当て、適応テスト戦略、進化するドキュメント、応答性の高いデプロイプロセスなど、さまざまな方法で現れます。たとえば、製品マネージャーは AI が生成したインサイトを使用してバックログの順序を変更し、新しい顧客要件と市場トレンドをほぼリアルタイムで統合できます。DevOps エンジニアは、パフォーマンス分析に基づいてデプロイプランとインフラストラクチャ設定を適応させ、アプリケーションの耐障害性と最適化を維持できます。開発チームは、スプリントの遡及からフィードバックを次のイテレーションのための実用的な改善に変換し、継続的なプロセス強化の文化を推進できます。
- 影響 — 生成 AI を適用して、ソフトウェア開発プロセスの有効性とパフォーマンスを評価します。AI を活用した分析とメトリクスを使用することで、開発効率、コード品質、ユーザーエンゲージメント、アプリケーション全体のパフォーマンスに関するより深いインサイトを得ることができます。このデータ駆動型アプローチは、情報に基づいた意思決定を行い、開発ワークフローを最適化し、アプリケーションの品質とユーザーエクスペリエンスを継続的に改善するのに役立ちます。ソフトウェアチームの生産性を評価する際、生成 AI はコードのコミット頻度、問題解決時間、リリース速度、機能配信速度など、さまざまなデータポイントを分析します。また、コードレビューの品質、コラボレーションツールの有効性、さまざまな開発プラクティスがチーム全体の出力に与える影響を評価することもできます。これらのメトリクスをプロジェクトの成果に関連付けることで、AI は人間のアナリストが見逃す可能性のあるパターンと傾向を特定し、チームの生産性を向上させる実用的なインサイトを提供できます。さらに、生成 AI は、業界標準や履歴データに対してチームのパフォーマンスをベンチマークし、改善のためのパーソナライズされた推奨事項を提供するのに役立ちます。また、開発プロセスにおける潜在的なボトルネックやリスクを予測して、予防的な対策を講じることもできます。

ソフトウェア開発ライフサイクルとの統合

SDLC は複数のフェーズで構成され、組織ごとに異なる場合があります。一般的に、これらのフェーズには、要件と計画、設計とアーキテクチャ、実装、テスト、デプロイ、運用とメンテナンスが含まれます。

次の表は、5-I フレームワークのディメンションを SDLC フェーズにマッピングし、各ディメンションの統合レベルを示しています。

フレームワークディメンション	要件と計画	設計とアーキテクチャ	実装	テスト	デプロイメント	オペレーションとメンテナンス
調査	高	低	低	低	低	中
統合	中	中	高	中	高い	高
インタラクション	高い	高い	高	中	中	高
反復	中	低	低	低	低	中
Impact	高	中	高	低	高	高

統合のレベルは高から低までさまざまです。マッピングでは、各ディメンションの主要な重点領域が明らかになります。例えば、調査は要件と計画フェーズで高い強度を示します。統合は、実装、デプロイ、運用、メンテナンスの各フェーズで高い強度を示します。

このマッピングを使用すると、作業を効果的に優先順位付けできます。高、中、低に集中することをお勧めします。生成 AI によるソフトウェア開発エクスペリエンスを強化する、バランスの取れた影響力のあるアプローチを採用してください。

AI を活用したソフトウェア開発エクスペリエンスの基本機能

生成 AI を活用したソフトウェア開発エクスペリエンスを正常に実装するには、組織内の複数のペルソナにまたがる一連の基本的な機能を確認する必要があります。これらの機能は、AI を活用したソフトウェア開発のコンテキストでリソースを効果的にデプロイし、プロセスを実装し、望ましい成果を達成する能力を表します。これらの機能を育成することで、SDLC のすべてのステージで生成 AI をシームレスに統合できる堅牢な基盤を構築できます。

AWS は、これらの機能の実装に役立つ主要なサービスを提供します。例えば、[Amazon Q Developer](#) は AI を活用したアシスタントとして機能することで、ソフトウェア開発の加速を支援します。[Amazon Q Business](#) は、差し迫った質問に対する迅速かつ適切な回答の取得、問題の解決、コンテンツの生成に役立ちます。また、ソフトウェア開発に関連するツールを統合することで、ユー

ザーに代わって動作することもできます。[Amazon Bedrock](#) は、基盤モデルとさまざまな機能にアクセスして、特定の開発ワークフローと要件をカスタマイズできます。

を通じてこれらの機能を育成することで AWS のサービス、SDLC のすべてのステージで生成 AI をシームレスに統合できる堅牢な基盤を構築できます。

以下は、重点を置くべき基本的な機能です。

- [プロジェクト管理](#)
- [要件管理](#)
- [アーキテクチャと設計](#)
- [コラボレーション](#)
- [DevSecOps](#)
- [オペレーションとメンテナンス](#)
- [AI アシスタント](#)
- [分析とインサイト](#)
- [ナレッジ管理](#)
- [拡張性](#)

各基本的な機能は、フレームワークのディメンションと SDLC のさまざまなステージと統合されます。この統合により、ソフトウェア開発プロセス全体で AI 機能を効果的に使用できます。これにより、すべてのステップで効率、品質、イノベーションが向上します。これらの基本的な機能、フレームワーク、SDLC ステージ間のシナジーにより、AI を活用したソフトウェア開発のための包括的なエコシステムが作成されます。これにより、生成 AI の可能性を最大限に活用し、継続的な改善を推進し、開発サイクルを加速し、高品質のソフトウェア製品を提供できます。

次の表は、基本的な機能とサブ機能がフレームワークディメンションと SDLC フェーズにどのようにマッピングされるかを示しています。

機能: サブ機能	調査	統合	操作	反復	Impact
プロジェクト管理: 問題管理	要件と計画	なし	なし	なし	なし

機能: サブ機能	調査	統合	操作	反復	Impact
プロジェクト管理: スプリントとタスクの管理	要件と計画	要件と計画	なし	なし	なし
プロジェクト管理: 製品バックログ管理	要件と計画	なし	なし	要件と計画	なし
プロジェクト管理: ユーザストーリーマッピング	要件と計画	なし	なし	なし	なし
プロジェクト管理: レポートと分析	要件と計画	なし	なし	なし	要件と計画
プロジェクト管理: 製品ロードマップ管理	要件と計画	なし	要件と計画	なし	なし
プロジェクト管理: フィードバックループ	なし	なし	なし	要件と計画	なし
プロジェクト管理: 遡及的	なし	なし	なし	要件と計画	なし
要件管理	要件と計画	要件と計画	なし	なし	なし

機能: サブ機能	調査	統合	操作	反復	Impact
アーキテクチャと設計: ソリューション設計	設計とアーキテクチャ	設計とアーキテクチャ	なし	なし	なし
コラボレーション: ドキュメント管理	すべての SDLC フェーズ	なし	すべての SDLC フェーズ	なし	なし
コラボレーション: ナレッジ共有	すべての SDLC フェーズ	なし	すべての SDLC フェーズ	なし	なし
コラボレーション: プロジェクトアセット管理	なし	すべての SDLC フェーズ	すべての SDLC フェーズ	なし	なし
DevSecOps: CI/CD	テスト、デプロイ	実装、テスト、デプロイ	デプロイメント	なし	なし
DevSecOps : DevOps セキュリティ	実装	実装、テスト、運用、メンテナンス	なし	実装、テスト、運用、メンテナンス	なし
DevSecOps : アプリケーションパフォーマンスのモニタリング	なし	オペレーションとメンテナンス	なし	なし	なし

機能: サブ機能	調査	統合	操作	反復	Impact
DevSecOps: ログの集約と分析	オペレーションとメンテナンス	オペレーションとメンテナンス	なし	なし	なし
DevSecOps: AIOps	オペレーションとメンテナンス	なし	なし	オペレーションとメンテナンス	なし
DevSecOps: 継続的な改善	なし	なし	なし	オペレーションとメンテナンス	なし
DevSecOps: ダッシュボードのモニタリング	なし	オペレーションとメンテナンス	なし	なし	なし
DevSecOps: パフォーマンスインサイト	オペレーションとメンテナンス	なし	なし	オペレーションとメンテナンス	なし
運用とメンテナンス: インシデント管理	なし	なし	なし	オペレーションとメンテナンス	なし
オペレーションとメンテナンス: コードのアップグレード	なし	オペレーションとメンテナンス	なし	なし	なし
オペレーションとメンテナンス: コードの最適化	オペレーションとメンテナンス	オペレーションとメンテナンス	なし	なし	なし

機能: サブ機能	調査	統合	操作	反復	Impact
運用とメンテナンス: 技術的負債管理	なし	オペレーションとメンテナンス	オペレーションとメンテナンス	なし	なし
運用とメンテナンス: 変更管理	なし	実装、デプロイ	なし	なし	なし
運用とメンテナンス: リバースエンジニアリング	オペレーションとメンテナンス	なし	なし	なし	なし
運用とメンテナンス: コードのモダナイゼーション	なし	実装	なし	なし	なし
運用とメンテナンス: パフォーマンスの最適化	なし	オペレーションとメンテナンス	なし	オペレーションとメンテナンス	なし
分析とインサイト	なし	要件と計画	なし	なし	すべての SDLC フェーズ
AI アシスタント	なし	なし	すべての SDLC フェーズ	なし	なし
ナレッジ管理	なし	なし	すべての SDLC フェーズ	なし	なし

機能: サブ機能	調査	統合	操作	反復	Impact
拡張性	なし	デプロイメント	なし	なし	なし

プロジェクト管理の生成 AI ユースケース

効果的なプロジェクト管理は、ソフトウェア開発の成功の中心です。生成 AI のコンテキストでは、プロジェクト管理は新しいディメンションを取ります。より予測的、適応的、データ駆動型になる可能性があります。AI を活用したプロジェクト管理ツールは、過去のプロジェクトデータを分析して、より正確な時間とリソースの見積もりを生成します。ビジネス目標とチームのキャパシティに基づいてタスクに自動的に優先順位を付けることができ、潜在的な障害が発生する前に予測することもできます。例えば、プロジェクトマネージャーは生成 AI を使用して、プロジェクトの要件と類似プロジェクトの履歴データに基づいて予備プロジェクト計画を作成できます。AI は、スキル、ワークロード、プロジェクトのニーズを考慮した最適なチーム構成を提案できます。プロジェクト全体で AI 駆動型ダッシュボードは、レポートを自動的に生成し、注意が必要な領域を強調することで、プロジェクトのステータスに関するほぼリアルタイムのインサイトを提供します。

プロジェクト管理に対するこの AI 拡張アプローチは、効率を高めることができます。これは、プロジェクトマネージャーが日常的な管理タスクに悩まされるのではなく、戦略的意思決定とチームリーダーシップに集中するのに役立ちます。

次の表は、生成 AI で強化できるプロジェクト管理のユースケースと、それらのユースケースを担当するペルソナを示しています。

サブ機能: ユースケース	ペルソナ
問題管理: 問題を作成して割り当てる	プロジェクトマネージャー
問題管理: テスト中に問題を検出し、ログに記録する	テストエンジニア
問題管理: 重要度に基づいて問題に優先順位を付け、デベロッパーに割り当てる	プロジェクトマネージャー
問題管理: 重複する問題を特定してマージする	プロジェクトマネージャー

サブ機能: ユースケース	ペルソナ
問題管理: プロジェクトの主要な問題、メトリクス、全体的な状態に関するレポートを追跡して生成します。	プロジェクトマネージャー
スプリントとタスクの管理: タスクの労力を見積もり、チームのキャパシティに基づいてストーリーポイントを割り当てる	スクラムマスター
スプリントとタスクの管理: チームメンバー間でタスクを分散し、スプリント全体にワークロードを均等に分散します。	スクラムマスター
スプリントとタスクの管理: チームの取り組みをスプリントの目標と整合させるスプリント計画セッションを促進する	スクラムマスター
製品バックログ管理: ビジネス価値、緊急性、ユーザーフィードバックに基づいてバックログ項目を並べ替える	プロダクトオーナー
製品バックログ管理: 新しい顧客のフィードバックと市場インサイトを製品バックログに統合して、ほぼリアルタイムの優先順位付けを行います。	プロダクトオーナー
製品バックログ管理: バックログ項目間の依存関係を特定および管理して開発を効率化する	製品マネージャー
ユーザーストーリーマッピング: ユーザージャーニーのマップを作成して、必要なすべての機能とそれに対応するユーザーストーリーを特定します。	プロダクトオーナー
ユーザーストーリーマッピング: ユーザーフローのギャップや欠落しているステップを特定する	ビジネスアナリスト

サブ機能: ユースケース	ペルソナ
ユーザーストーリーマッピング: ビジネス価値への影響に基づいてユーザーストーリーを優先する	製品マネージャー
レポートと分析: スプリント速度や問題解決率などの主要なプロジェクトメトリクスを視覚化するほぼリアルタイムのダッシュボードを生成する	プロジェクトマネージャー
レポートと分析: 履歴データを分析し、潜在的な遅延やボトルネックなど、将来のプロジェクト成果を予測する	プロジェクトマネージャー
レポートと分析: チームのパフォーマンスやプロジェクトのステータスレポートなど、さまざまなステークホルダーに合わせたカスタムレポートを作成します。	プロジェクトマネージャー
製品ロードマップ管理: 主要なマイルストーンとリリース日の概要を示す製品ロードマップを作成および維持する	プロジェクトマネージャー
製品ロードマップ管理: プロジェクトの優先順位やタイムラインの変化に基づいてロードマップを更新する	製品マネージャー
製品ロードマップ管理: ステークホルダーとロードマップを共有し、製品の方向性を可視化する	製品マネージャー
フィードバックループ: 各スプリント後にチームからフィードバックを収集し、改善すべき分野を特定する	スクラムマスター
遡及的: フィードバックを実行可能な項目に変換して次のスプリントを行い、継続的な改善を促進します。	スクラムマスター

サブ機能: ユースケース	ペルソナ
遡及: 以前の遡及から実装された変更の影響を追跡して、その有効性を測定します。	スクラムマスター

要件管理のための生成 AI ユースケース

要件管理は、プロジェクト管理に密接に関連する重要なプロセスです。AI ツールを使用して顧客のフィードバック、市場傾向、ステークホルダーのインプットを分析する製品所有者を想像してみてください。AI ツールは、包括的なユーザーストーリーと要件のセットを生成し、自動的に分類し、潜在的な競合やギャップを検出し、ビジネス価値と実装の複雑さに基づいて優先順位付けを提案することもできます。プロジェクトが進行し、要件が進化するにつれて、AI は要件を継続的に更新および絞り込み、変化するビジネスニーズと技術的な制約に確実に一致させることができます。要件管理に対するこの動的な AI 主導のアプローチは、プロジェクトライフサイクルを通じて、開発作業がユーザーのニーズとビジネス目標に厳密に整合していることを確認するのに役立ちます。

次の表は、生成 AI で強化できる要件管理のユースケースと、それらのユースケースを担当するペルソナを示しています。

ユースケース	ペルソナ
ビジネス要件を作成する	ビジネスアナリスト
機能からエピックを作成する	プロダクトオーナー
関連するユーザーストーリーの完了をモニタリングしてエピックの進行状況を追跡する	製品マネージャー
ユーザーストーリーを作成する	プロダクトオーナー
各使用ストーリーに必要な労力を見積もり、ストーリーポイントを割り当てます。	スクラムマスター
各ユーザーストーリーの受け入れ基準を定義する	プロダクトオーナー

アーキテクチャと設計の生成 AI ユースケース

プロジェクト管理の強固な基盤と明確に定義された要件により、次の重要な機能はアーキテクチャと設計です。ここでは、生成 AI によって、堅牢でスケーラブル、効率的なソフトウェアアーキテクチャを作成するための新しい可能性が開かれています。AI を活用した設計ツールは、要件と制約を分析して、最適なアーキテクチャパターンと設計アプローチを提案できます。複数の設計代替案を生成し、それぞれがパフォーマンス、スケーラビリティ、保守性など、さまざまな優先順位に合わせて最適化されます。たとえば、ソリューションアーキテクトは AI アシスタントを使用して、プロジェクト要件に基づいていくつかの高レベルのアーキテクチャ設計をすばやく生成できます。この AI 拡張アプローチは、設計プロセスを加速し、アーキテクトがより多くの情報に基づいた意思決定を行うのに役立ちます。これにより、より堅牢で将来を見据えたソフトウェア設計が可能になります。

次の表は、生成 AI で強化できるアーキテクチャと設計のユースケースと、それらのユースケースを担当するペルソナを示しています。

ユースケース	ペルソナ
アーキテクチャドキュメントを作成する	ソリューションアーキテクト
詳細な設計ドキュメントを作成する	テクニカルリード
既存のアーキテクチャと設計標準を理解する	ソリューションアーキテクト
ユーザーインターフェイスの詳細なモックアップとプロトタイプを開発する	UX/UI デザイナー

コラボレーションの生成 AI ユースケース

ソフトウェア開発は本質的に共同作業です。生成 AI を使用して、ソフトウェア開発チームのコラボレーションを強化できます。AI を活用したコラボレーションツールは、シンプルなメッセージングやファイル共有にとどまりません。長いディスカッションスレッドを要約し、主要な決定事項を強調し、さらにはチームメンバーのスケジュールと生産性パターンに基づいて会議に最適な時間を提案することで、より効果的なコミュニケーションを促進します。AI は、潜在的な問題を自動的に特定し、改善を提案し、レビュワーに複雑な変更を説明することで、コードレビューを支援できます。ブレインストーミングセッション中、AI はファシリテーターとして行動し、アイデアを生成し、考えを整理し、さらには議論を仲介して、すべての音声が高らかに聞こえるようにすることができます。分散型チームにとって、AI は文化や言語の障壁を埋めるのに役立ちます。チャットやビデオ通話でほぼリアルタイムの言語翻訳を提供し、誤解を防ぐために文化的コンテキストを提供できます。AI

との人間によるコラボレーションを強化することで、チームはより効率的かつ効果的に作業できるため、イノベーションが促進され、プロジェクト全体の成果が向上します。

次の表は、生成 AI を使用してコラボレーションのユースケースを強化する方法を示しています。

サブ機能: ユースケース	ペルソナ
ドキュメント管理: 一元化されたドキュメントリポジトリを作成して維持する	テクニカルライター
ドキュメント管理: 複数のチームメンバーがリアルタイムでドキュメントを共同作業できるようにする	開発チーム
ナレッジ共有: デベロッパーが質問し、知識を共有し、共同で問題をトラブルシューティングするためのプラットフォームとしてディスカッションフォーラムを使用する	開発チーム
ナレッジ共有: ディスカッションフォーラムを使用して、プロジェクトのディスカッション中に行われた決定を文書化して追跡し、重要な決定の根拠がキャプチャされ、将来の参照のためにアクセス可能であることを確認します。	製品マネージャー
プロジェクトアセット管理: プロジェクト関連リソースの共有を容易にする	開発チーム
プロジェクトアセット管理: 共有コンテンツのバージョン管理を実装して、チームメンバーが変更を追跡し、以前のバージョンに戻して、コンテンツ更新でコラボレーションできるようにします。	開発チーム

DevSecOps の生成 AI ユースケース

AI を活用した DevSecOps ツールは、ソフトウェア配信パイプラインの多くの側面を自動化します。たとえば、インテリジェントなコードレビューの実行、潜在的なバグの検出、セキュリティの脆弱

弱性の検出、デベロッパーがコードを記述する際のパフォーマンスの問題のほぼリアルタイムの特定を行うことができます。AI は包括的なテストスイートを生成して実行し、コードベースの進化に合わせて自動的に更新します。DevSecOps に対するこの AI 拡張アプローチは、配信パイプラインを加速し、配信されるソフトウェアのセキュリティと信頼性を大幅に強化します。

次の表は、生成 AI で強化できる DevSecOps ユースケースと、それらのユースケースを担当するペルソナを示しています。

サブ機能: ユースケース	ペルソナ
DevOps と継続的デリバリー: デプロイパイプライン全体を自動化	DevOps エンジニア
DevOps と継続的デリバリー: コードの品質と潜在的な問題に関するほぼリアルタイムのフィードバックを受け取る	ソフトウェア開発者
DevOps と継続的デリバリー: ほぼリアルタイムのセキュリティ問題と修復に関する推奨事項を受け取る	ソフトウェア開発者
DevOps と継続的デリバリー: ほぼリアルタイムのコードとベストプラクティスの提案を受け取る	ソフトウェア開発者
DevOps と継続的デリバリー: 反復タスクを自動化し、コマンドをスクリプトに統合する	DevOps エンジニア
DevOps と継続的デリバリー: コードを作成し、各コードコミット後にアーティファクトを自動的に生成する	ソフトウェア開発者
DevOps と継続的デリバリー: 組織の標準とフレームワークに従ってコードを構築する	ソフトウェア開発者
DevOps と継続的デリバリー: すべてのコミットでユニットテストを自動的に実行して、開発プロセスの早い段階でエラーをキャッチする	ソフトウェア開発者

サブ機能: ユースケース	ペルソナ
DevOps と継続的デリバリー: ユニットテストのカバレッジを分析し、すべての重要なコードパスがテストされていることを確認します。	ソフトウェア開発者
DevOps と継続的デリバリー: ブランチを管理し、変更をマージする	ソフトウェア開発者
DevOps と継続的デリバリー: コードとアーティファクトのバージョンングを管理する	ソフトウェア開発者
DevOps と継続的デリバリー: ビルドアーティファクトと依存関係の保存と管理	DevOps エンジニア
DevOps と継続的デリバリー: ビルドプロセス中に依存関係を解決して取得する	ソフトウェア開発者
DevOps と継続的デリバリー: 統合テストを生成して実行し、コンポーネントが期待どおりに連携することを確認します。	テストエンジニア
DevOps と継続的デリバリー: 統合テスト中にモックサービスを使用して外部システムとのインタラクションをシミュレートする	テストエンジニア
DevOps と継続的デリバリー: さまざまな負荷でアプリケーションのパフォーマンスをベンチマークする	パフォーマンスエンジニア
DevOps と継続的デリバリー: トラフィックの多いシナリオをシミュレートして、アプリケーションのスケーラビリティと応答時間をテストする	パフォーマンスエンジニア
DevOps と継続的デリバリー: サーバーのクラッシュやネットワークの停止などの障害から回復するシステムの機能をテストする	サイト信頼性エンジニア

サブ機能: ユースケース	ペルソナ
DevOps と継続的デリバリー: カオスエンジニアリングを実行する	サイト信頼性エンジニア
DevOps と継続的デリバリー: テストを実行して、アプリケーションがビジネス要件を満たしていることを確認します。	QA エンジニア
DevOps と継続的デリバリー: ユーザー受け入れテストを実施する	プロダクトオーナー
DevOps と継続的デリバリー: 脆弱性とライセンスコンプライアンスの問題の依存関係をスキャンします	セキュリティエンジニア
DevOps と継続的デリバリー: オープンソースの依存関係を監視および管理して、それらが最新で安全であることを確認します。	セキュリティエンジニア
DevOps と継続的デリバリー: ソフトウェア部品表 (SBOM) を生成して維持し、すべてのコンポーネントと依存関係を追跡する	セキュリティエンジニア
DevOps と継続的デリバリー: SBOM を使用して規制コンプライアンスの監査を実施する	コンプライアンス責任者
DevOps と継続的デリバリー: リリースノートを作成する	リリースマネージャー
DevOps と継続的デリバリー: リリースの計画と調整	リリースマネージャー
DevOps と継続的デリバリー: ロールバックとリリース管理の標準運用手順を実装する	リリースマネージャー
DevOps と継続的デリバリー: 機能フラグを使用して、新しいコードをデプロイせずに本番環境の機能を有効化または無効化する	製品マネージャー

サブ機能: ユースケース	ペルソナ
DevOps と継続的デリバリー: 機能フラグを使用して A/B テストを実行し、さまざまな機能がユーザー動作に与える影響を測定します。	製品マネージャー
DevOps と継続的デリバリー: パイプラインの障害の分析とモニタリング	DevOps エンジニア
DevOps と継続的デリバリー: インフラストラクチャリソースの作成と管理	DevOps エンジニア
DevOps とセキュリティ: ハードコードされたシークレットのコードリポジトリをスキャンします。	DevOps エンジニア
DevOps とセキュリティ: ほぼリアルタイムの検出を実装して、シークレットがリポジトリにコミットされた場合に開発者にすぐに警告する	DevOps エンジニア
DevOps とセキュリティ: 継続的なコード品質モニタリングを実施する	ソフトウェア開発者
DevOps とセキュリティ: コード内の潜在的なセキュリティ脆弱性のインジケータを検出してフラグ付けする	ソフトウェア開発者
DevOps とセキュリティ: Open Worldwide Application Security Project (OWASP) の上位 10 のセキュリティリスクの自動テストを実装し、アプリケーションが業界標準のセキュリティプラクティスに準拠していることを確認します。	セキュリティエンジニア
DevOps とセキュリティ: 開発プロセスにチェックを統合することで、OWASP リスクについてデベロッパーを定期的に更新して教育する	セキュリティエンジニア

サブ機能: ユースケース	ペルソナ
DevOps とセキュリティ: サードパーティーのライブラリと依存関係をスキャンして、既知のセキュリティの脆弱性を検出する	DevOps エンジニア
DevOps とセキュリティ: アプリケーションコードとインフラストラクチャをスキャンして脆弱性を検出する	DevOps エンジニア
DevOps とセキュリティ: デプロイ前にコードの脆弱性を分析する	セキュリティエンジニア
DevOps とセキュリティ: 重大な脆弱性を持つコードがマージされないようにしてセキュリティポリシーを適用する	セキュリティエンジニア
DevOps とセキュリティ: ロールベースのアクセスコントロール (RBAC) を実装して、機密性の高いシステムやデータへのアクセスを制限し、承認された担当者のみが重要なリソースにアクセスできるようにします。	セキュリティエンジニア
DevOps とセキュリティ: チーム構造の変化に適応することで、役割と責任に基づいてアクセスコントロールを調整する	DevOps エンジニア
DevOps とセキュリティ: 本番環境への攻撃をシミュレートして、実行中のアプリケーションをほぼリアルタイムでテストします。	セキュリティエンジニア
DevOps とセキュリティ: デプロイされたアプリケーションにセキュリティの脆弱性を継続的にモニタリングする	DevOps エンジニア
DevOps とセキュリティ: すべての環境で定期的な脆弱性スキャンをスケジュールして、セキュリティの弱点を特定して対処する	セキュリティエンジニア

サブ機能: ユースケース	ペルソナ
DevOps とセキュリティ: 脆弱性スキャンの結果に基づいてパッチと更新を適用し、安全なシステムを維持するのに役立ちます。	DevOps エンジニア
アプリケーションパフォーマンスのモニタリング: アプリケーションのパフォーマンスをほぼリアルタイムで継続的にモニタリングし、ユーザーに影響を与える前にパフォーマンスの問題を検出して診断します。	サイト信頼性エンジニア
アプリケーションパフォーマンスのモニタリング: 応答時間の急増やエラー率の増加などのパフォーマンスの異常を検出し、アラートを開始します。	DevOps エンジニア
アプリケーションパフォーマンスのモニタリング: 分散システムを介して伝達されるリクエストをトレースして、パフォーマンスのボトルネックとレイテンシーの問題を特定します。	DevOps エンジニア
アプリケーションパフォーマンスのモニタリング: 分散トレースを使用して、障害やパフォーマンスの低下の原因となっているサービスやコンポーネントを正確に特定する	DevOps エンジニア
ログの集約と分析: 複数のソースからのログを一元化されたシステムに集約し、傾向と問題を特定するための検索と分析を容易にします。	サイト信頼性エンジニア
ログの集約と分析: 自動ログ解析を実装して関連情報を抽出し、問題を示す可能性のあるパターンや異常を検出します。	DevOps エンジニア
ログの集約と分析: 主要なパフォーマンスメトリクスを収集して視覚化する	サイト信頼性エンジニア

サブ機能: ユースケース	ペルソナ
ログの集約と分析: 事前定義されたサービスレベルアグリーメント (SLAs)	製品マネージャー
AI オペレーション: インシデントを検出し、根本原因を分析し、人間の介入なしに是正措置を開始する	DevOps エンジニア
AI オペレーション: 停止を回避するために、将来のリソース需要を予測し、キャパシティプランニングを最適化する	サイト信頼性エンジニア
継続的な改善: アプリケーションとの実際のユーザーインタラクションをモニタリングして、パフォーマンスに関するインサイトを収集し、改善すべき分野を特定する	UX デザイナー
継続的な改善: さまざまな地理的リージョンのアプリケーションパフォーマンスを追跡し、一貫したユーザーエクスペリエンスをグローバルに確保する	製品マネージャー
ダッシュボードモニタリング: カスタマイズ可能なダッシュボードを作成して重要なメトリクス、ログ、トレースをほぼリアルタイムで視覚化し、システムの状態を包括的に把握できます。	サイト信頼性エンジニア
ダッシュボードモニタリング: さまざまなチーム (開発チーム、運用チーム、製品チームなど) のダッシュボードを作成し、重点分野に基づいて関連するインサイトを提供します。	DevOps エンジニア
パフォーマンスインサイト: アプリケーションパフォーマンスの詳細な分析を実施して非効率性を特定し、コードまたはインフラストラクチャを最適化する	ソフトウェア開発者

サブ機能: ユースケース	ペルソナ
パフォーマンスインサイト: パフォーマンスインサイトを使用してアプリケーションパフォーマンスを繰り返し改善し、時間の経過とともにユーザーエクスペリエンスを最適化する	製品マネージャー

運用とメンテナンスのための生成 AI ユースケース

ソフトウェアがデプロイされると、フォーカスはオペレーションとメンテナンスに移行します。生成 AI は、よりプロアクティブで効率的なシステム管理を提供することで、従来のアプローチを強化できます。AI を活用した運用ツールは、システムパフォーマンスを継続的にモニタリングし、潜在的な問題をユーザーに影響を与える前に予測します。問題が発生したときに自動根本原因分析を実行するため、解決までの平均時間が大幅に短縮されます。AI は、システムパフォーマンスをほぼリアルタイムで最適化します。負荷パターンとユーザー動作の変化に基づいて設定を自動的に調整します。例えば、運用チームは AI アシスタントを使用して予測メンテナンススケジュールを生成し、失敗する可能性が高いコンポーネントを自動的に特定し、先制アクションを提案する場合があります。AI は、使用状況の傾向を分析し、将来のリソースニーズを高精度に予測することで、キャパシティプランニングにも役立ちます。

次の表は、生成 AI で強化できるオペレーションとメンテナンスのユースケースと、それらのユースケースを担当するペルソナを示しています。

サブ機能: ユースケース	ペルソナ
インシデント管理: モニタリングツールをチャットプラットフォームと統合してインシデントをほぼリアルタイムで管理し、チームがチャット環境内で直接問題を検出、議論、解決できるようにします。	サイト信頼性エンジニア
インシデント管理: チームがチャットインターフェイスから直接デプロイを開始し、スクリプトを実行し、コマンドを実行できるようにします。これにより、オペレーションが効率化されます。	DevOps エンジニア

サブ機能: ユースケース	ペルソナ
コードのアップグレード: コードの依存関係とライブラリをアップグレードして手動作業を減らし、コードベースが最新バージョンで最新の状態を維持できるようにします。	ソフトウェア開発者
コードの最適化: 最適化の機会のコードを確認する	ソフトウェア開発者
コードの最適化: コードのボトルネックを特定し、リファクタリングまたは最適化してパフォーマンスを向上させる	ソフトウェア開発者
技術的負債管理: 開発プロセスの一環として技術的負債を記録する	製品マネージャー
技術的負債管理: 影響、リスク、コストに基づいて技術的負債を優先順位付けして対処し、通常のスプリント計画プロセスに統合します。	ソフトウェア開発者
技術的負債管理: 既存のアプリケーションコードの技術的負債の削減	ソフトウェア開発者
変更管理: すべてのコード変更がデプロイ前に必要な利害関係者によってレビュー、テスト、承認されていることを確認する変更承認プロセスを実装します。	変更マネージャー
変更管理: 提案された変更の影響分析を実行する	DevOps エンジニア
リバースエンジニアリング: レガシーコードの構造と動作を分析して理解する	ソリューションアーキテクト
リバースエンジニアリング: 既存のコードの説明とドキュメントの生成	ソフトウェア開発者

サブ機能: ユースケース	ペルソナ
コードモダナイゼーション: あるプログラミング言語から別のプログラミング言語にコードを翻訳する	ソフトウェア開発者
コードモダナイゼーション: レガシーコードを最新のプログラミング言語にモダナイズする	ソフトウェア開発者
パフォーマンスの最適化: リソースの割り当て、負荷分散、アプリケーションの再設定を最適化して、システムパフォーマンスを継続的にモニタリングおよび調整する	サイト信頼性エンジニア
パフォーマンスの最適化: 速度とシステムの応答性を向上させるために、パフォーマンスの低下を引き起こしているコードを特定してリファクタリングする	ソフトウェア開発者

ソフトウェア開発における生成 AI アシスタントのユースケース

AI アシスタント機能は、生成 AI を活用した開発エクスペリエンスの中心です。このインテリジェントなコンテキスト対応システムは、SDLC 全体のすべてのチームメンバーの仮想共同作業員として機能します。複雑なコードを扱う開発者を想像してみてください。AI アシスタントに支援を求めるだけで、関連するコードスニペットを提供したり、複雑なアルゴリズムを説明したり、現在のコンテキストやベストプラクティスに基づいて最適化を提案したりできます。AI アシスタントは、ITOps マネージャーが内部ドキュメントに基づく標準運用手順を理解するのに役立ちます。AI アシスタントは、瞬時にコンテキストに応じたサポートを提供することで、チームメンバーの認知負荷を大幅に軽減します。これにより、より高いレベルの問題解決と創造的なタスクに集中できます。この機能は、ソフトウェア開発のすべての段階で生産性と品質を向上させる力の乗数として機能します。

次の表は、AI アシスタントと利点のあるペルソナで強化できるユースケースを示しています。

ユースケース	ペルソナ
要件、アーキテクチャ、標準運用手順などの質問に答えて、開発チームにすぐに支援を提供する	ソフトウェア開発チーム
自然言語クエリを使用して、広範なドキュメントから抜粋を検索または取得するか、概要を生成する	ソフトウェア開発チーム
要件ドキュメント、アーキテクチャ設計ドキュメント、内部プロセスなどの長い技術ドキュメントを要約する	ソフトウェア開発チーム
チームが一般的なタスクに使用できるプロンプトのライブラリを維持する	ソフトウェア開発チーム
生成 AI を既存のツールやシステムにシームレスに統合する	ソフトウェア開発チーム
さまざまなプラットフォーム、ツール、内部システムにわたるタスクを自動化する	ソフトウェア開発チーム
ベストプラクティス、プロジェクト固有の情報、チームの知識など、すべてのチームメンバーがアクセスできる一元的な知識リポジトリを作成します。	ソフトウェア開発チーム
タスクのコンテキストに基づいてリポジトリから関連する知識を取得する	ソフトウェア開発チーム
自動コードレビュー、根本原因分析の実行、改善の提案、潜在的なバグの検出、トラブルシューティングの実行	ソフトウェア開発者、DevOps エンジニア、サイト信頼性エンジニア
パフォーマンスデータを分析して、パフォーマンスの最適化に関する意思決定に役立つ傾向とパターンを特定する	サイト信頼性エンジニア

ユースケース	ペルソナ
効率の向上、複雑さの軽減、セキュリティの強化に関する推奨事項を提供する	ソフトウェア開発者
スケーリングの推奨事項やコスト削減戦略など、クラウドリソースの使用の最適化を提案する	ソフトウェア開発者、DevOps エンジニア、サイト信頼性エンジニア、ソリューションアーキテクト
コード、ユーザーガイド、製品機能リリースに基づくドキュメントなどの新しいコンテンツを生成する	ソフトウェア開発チーム

分析とインサイトの生成 AI ユースケース

分析とインサイトの機能は、膨大な量のデータを実用的なインサイトに変換し、意思決定と継続的な改善を促進するのに役立ちます。生成 AI を使用することで、この機能はコードリポジトリ、プロジェクト管理ツール、チームコラボレーションプラットフォームなど、さまざまなソースからのデータを処理し、開発プロセスとチームの生産性を包括的に把握できます。生成 AI は、予測分析と規範分析を提供するために、従来のメトリクスを超えています。潜在的な問題を予測し、ターゲットを絞った改善を提案できます。たとえば、コードコミット、バグ解決率、機能配信速度のパターンを分析して、パフォーマンスの高いチームを特定し、ボトルネックを特定し、プロセスの最適化を提案できます。さらに、チームのダイナミクスと個々のパフォーマンスに関するインサイトを提供できます。これらのインサイトは、リーダーがワークロードの分散、トレーニングニーズ、チーム構成に関するデータ主導型の意思決定を行うのに役立ちます。インタラクティブダッシュボードを通じてこれらのインサイトを提示することで、あらゆるレベルのステークホルダーが情報に基づいた意思決定を行い、プロセスを最適化し、チームの生産性を継続的に向上させ、高品質のソフトウェアを迅速に提供できるようになります。

次の表は、生成 AI で強化できる分析ユースケースと、それらのユースケースを担当するペルソナを示しています。

ユースケース	ペルソナ
個人とチームの生産性をモニタリングする	開発マネージャー

ユースケース	ペルソナ
生産性の傾向を分析して潜在的なバーンアウトを検出し、チームのウェルビーイングと生産性を維持するために積極的な対策を講じることができます。	開発マネージャー
コード変更が本番環境にデプロイされる頻度を追跡し、開発プロセスの速度と俊敏性を測定します。	製品マネージャー
デプロイ頻度データを分析して、プロセスの非効率性やリソースの制約を示す可能性のあるデプロイアクティビティが少ない期間を特定する	製品マネージャー
開発およびデプロイプロセスを合理化する機会を特定するために、コードがデプロイにコミットされるまでの時間を測定します。	開発マネージャー
リリースプロセスの信頼性を評価するために、即時修復が必要な障害につながるデプロイの割合を追跡する	サイト信頼性エンジニア
変更失敗率メトリクスを使用して、頻繁に問題を引き起こすコード領域を特定し、ターゲットを絞ったリファクタリングとテスト作業をガイドします。	ソフトウェア開発者
停止またはインシデント後のサービスの復元にかかる時間をモニタリングして、ダウンタイムを短縮し、システム全体の耐障害性を向上させることができます。	サイト信頼性エンジニア
復旧時間の傾向を分析してインシデント対応プロセスを強化し、システム障害からの復旧を迅速化	DevOps エンジニア

ユースケース	ペルソナ
開発と運用の健全性を包括的に把握するために、デプロイ頻度、リードタイム、変更失敗率などの主要なメトリクスを集約するカスタマイズされたダッシュボードを作成する	製品マネージャー
開発、運用、ビジネスなど、特定の責任分野に焦点を当てたインサイトを提供するために、さまざまなチームのニーズに合わせたダッシュボードを作成する	製品マネージャー
収益への影響、顧客満足度、市場シェアなどのビジネス主要業績評価指標 (KPIs) を追跡し、開発の取り組みをより広範なビジネス目標に合わせる	製品マネージャー
ビジネス KPIs で、その成功を評価し、将来の製品開発をガイドする	ビジネスアナリスト
コードベースが維持可能で安全であることを確認するために、コードの複雑さ、テストカバレッジ、バグ密度などのコード品質メトリクスをモニタリングする	ソフトウェア開発者
長期的な持続可能性を推進し、技術的負債を減らすためにリファクタリングが必要なコードベースの領域を特定する	ソリューションアーキテクト

ナレッジ管理のための生成 AI ユースケース

ソフトウェア開発組織では、知識は重要なアセットです。生成 AI を活用したナレッジ管理機能により、このアセットのキャプチャ、整理、使用方法が強化されます。従来のナレッジ管理システムでは、情報が多すぎたり、古いコンテンツが含まれていたり、関連情報をすばやく見つけるために検索するのが難しいことがよくあります。

生成 AI は、これらの課題に正面から対処します。コードの変更、会話、プロジェクトアーティファクトに基づいて、ドキュメントを自動的に生成および更新します。これにより、チームメンバーの手

動作業を必要とせずに、ナレッジベースを最新の状態に保つことができます。さらに重要なのは、AI はこの知識を直感的な方法でアクセスできるようにすることです。チームメンバーは自然言語で質問でき、AI は関連する回答を提供できます。AI は、公式ドキュメント、コードコメント、ディスカッションスレッド、外部リソースなど、さまざまなソースから取得できます。たとえば、特定のコンポーネントを理解しようとする新しいチームメンバーが AI に「認証モジュールの仕組み」と尋ねることがあります。AI は、簡潔な説明と、関連するコードセクション、アーキテクチャ図、最近の変更へのリンクを提供します。チームメンバーの役割と専門知識のレベルに基づいてこの情報を調整することもできます。

この機能により、オンボーディングが高速化され、反復的な質問が軽減され、組織全体の知識共有が促進されます。組織の知識を保持するため、チームは時間の経過とともに複雑なシステムを簡単に維持および進化できます。

次の表は、生成 AI で強化できるナレッジ管理のユースケースと、それらのユースケースを担当するペルソナを示しています。

ユースケース	ペルソナ
プロジェクト関連のすべての知識に簡単にアクセスできる統合プラットフォームを作成する	ソフトウェア開発チーム
さまざまな開発アクティビティから知識をキャプチャする	ソフトウェア開発チーム
リポジトリ内の関連する知識をすばやく見つけるための高度な検索機能を提供する	ソフトウェア開発チーム
チームの学習モジュールとパスをパーソナライズする	ソフトウェア開発チーム

拡張のための生成 AI ユースケース

拡張性により、組織は AI システムを特定のニーズに合わせて調整しながら、既存のツールやワークフローとシームレスに統合できます。この機能は、一般的な開発およびプロジェクト管理ツールへの AI 機能の統合を容易にする堅牢な APIs、SDKs、カスタマイズ可能なインターフェイスを提供します。たとえば、組織は AI を活用した機能を使用して Jira を強化し、チケットの優先順位付け、労力の見積もり、スプリント計画を自動化できます。Jenkins パイプラインを AI で拡張して、インテリジェントなビルドの最適化と予測テストの選択を行うことができます。

さらに、拡張性により、統合開発環境 (IDEs)、バージョン管理システム、コードレビュープラットフォームとの深い統合が可能になります。AI は、コードの作成、コードレビューの自動化、コンテキストドキュメントの生成に役立ちます。

この機能は、組織固有のデータに関する AI モデルのトレーニングと微調整もサポートしています。これにより、AI は会社固有のコーディングパターン、アーキテクチャの好み、ドメインの知識を理解できます。結果は、すべての統合ツールでより関連性が高く、コンテキストに応じた支援を提供します。このレベルの柔軟性と統合を提供することで、拡張性により、AI を活用した開発エクスペリエンスが組織とともに進化します。既存のツールチェーンやワークフローをシームレスに強化しながら、変化するテクノロジーやビジネスニーズに適応できます。

次の表は、生成 AI で強化できる拡張性のユースケースと、それらのユースケースを担当するペルソナを示しています。

ユースケース	ペルソナ
サードパーティーのツールを開発環境に統合する	DevOps エンジニア
チーム固有の開発プロセスに合わせてカスタマイズされたカスタムオートメーションワークフローを作成する	DevOps エンジニア
さまざまな APIs とサービスに接続する	DevOps エンジニア
クロスプラットフォームツール用のコネクタを作成する	DevOps エンジニア

ソフトウェア開発で生成 AI を使用するためのベストプラクティス

このセクションでは、生成 AI をソフトウェア開発ライフサイクル (SDLC) に統合するためのベストプラクティスについて説明します。これらのガイドラインは、シームレスなツールチェーンと DevSecOps パイプラインの実装からコラボレーションの促進や反復タスクの自動化まで、AI の能力を活用して開発プロセスとエクスペリエンスを向上させるのに役立ちます。これらのベストプラクティスに従うことで、ソフトウェア開発チームは作業における新しいレベルの効率、イノベーション、品質を引き出すことができます。

このセクションでは、以下のベストプラクティスについて説明します。

- [シームレスでend-to-endの統合ツールチェーンの実装](#)
- [DevSecOps 用のend-to-end CI/CD パイプラインの実装](#)
- [コラボレーションツールとプラクティスの採用](#)
- [反復タスクの自動化](#)
- [開発エクスペリエンスを定期的に見直し、反復する](#)
- [効果的なプロジェクト管理プラクティスの採用](#)
- [ナレッジ管理の実装](#)
- [拡張性とカスタマイズの提供](#)
- [オペレーションの最適化](#)
- [データ駆動型インサイトの使用](#)
- [プラットフォームベースのアプローチを採用する](#)

シームレスでend-to-endの統合ツールチェーンの実装

シームレスでend-to-endの統合ツールチェーンを実装することは、生成 AI を活用した開発エクスペリエンスを作成するための基本的なベストプラクティスです。コアアイデアは、ソフトウェアチームが SDLC 全体で使用できるツールとプラットフォームの一貫したエコシステムを確立することです。チームはツールチェーンを使用して、継続的な運用を計画、考案、コード化、構築、テスト、デプロイ、管理できます。生成 AI 機能をこのツールチェーンに統合することで、あらゆる段階で AI 支援を利用できるようにします。この統合により、手動のハンドオフが軽減または排除され、コンテキストの切り替えが軽減され、データとアーティファクトが異なる開発フェーズ間でスムーズに流れるようになります。例えば、統合開発環境 (IDE) からの AI 生成コードスニペットはバージョン管理シ

システムにシームレスに流れ、デプロイプラットフォームからの AI を活用した分析はプロジェクト管理ツールに情報を提供できます。これにより、開発プロセスを改善する継続的なフィードバックループが作成されます。

DevSecOps 用の end-to-end CI/CD パイプラインの実装

この統合されたツールチェーンに基づいて構築するには、DevSecOps end-to-end の継続的インテグレーションと継続的デプロイ (CI/CD) パイプラインを実装します。この AI を活用したパイプラインは、ソフトウェア配信プロセスを合理化する重要なコンポーネントです。これにより、新しいアプリケーションや更新をより迅速かつ確実にリリースできます。SDLC 全体にセキュリティプラクティスを埋め込むことで、脆弱性をより早く特定して対処できるため、全体的なコストとリスクを軽減できます。パイプラインには、継続的な統合とテストからセキュリティチェックとデプロイまで、あらゆる段階で AI を組み込む必要があります。たとえば、AI を使用してコードコミットをほぼリアルタイムで分析できるため、潜在的な統合の問題が発生する前に予測できます。CI/CD パイプラインでは、生成 AI を使用して、最新の脅威インテリジェンスに基づいてセキュリティポリシーを自動的に更新することもできます。

コラボレーションツールとプラクティスの採用

開発インフラストラクチャを強化するときは、ヒューマン要素を忘れないでください。ソフトウェア開発は本質的に共同作業です。これには、開発者、デザイナー、製品マネージャー、スクラムマスター、ビジネスアナリスト、その他の利害関係者で構成される部門横断的なチームが含まれます。これらの個人は丸手となってアイデアを生み出します。最新のコラボレーションツールを使用し、オープンなコミュニケーションと知識共有の文化を育むことで、ソフトウェア開発チームの生産性と有効性を大幅に向上させることができます。AI を活用したソフトウェア開発エクスペリエンスでは、これらのツールは新しい側面を取ります。AI をコラボレーションプラットフォームに統合して、チームメンバー間のコミュニケーションと知識共有をより効果的に行うことができます。AI アシスタントは、一般的な質問に答えたり、議論をまとめたり、競合を仲介したりできます。生成 AI は、改善を自動的に提案したり、潜在的な問題を特定したりすることで、コードレビュープロセスを強化できます。さらに、AI を使用して、プロジェクトが進化するにつれてほぼリアルタイムで更新する動的でコンテキスト対応のドキュメントを作成し、すべてのチームメンバーが最新の関連情報にアクセスできるようにすることができます。

反復タスクの自動化

生成 AI を使用して時間のかかる日常的なアクティビティを処理することで、ソフトウェアチームはイノベーションを促進し、ビジネスインパクトをもたらす価値の高いクリエイティブな作業に集中

できるようになります。反復タスクの例としては、定型コードの生成、テストデータの作成、ドキュメントの記述、初期プロジェクト計画のドラフト作成などがあります。これらのタスクを AI にオフロードすることで、チームメンバーはよりクリエイティブで戦略的な作業に集中できます。たとえば、AI を活用したコード補完ツールは、コンテキストとコーディングパターンに基づいて関連するコードスニペットを提案することで、コーディングプロセスを大幅に高速化できます。同様に、生成 AI はコードの変更に応じて技術ドキュメントを自動的に作成および更新できます。これにより、ドキュメントが最新に保たれ、このタスクに通常必要な手動作業が軽減されます。テストでは、AI は要件とコード分析に基づいて包括的なテストケースを生成できるため、テストカバレッジが向上し、エッジケースが見落とされる可能性が低くなります。これらの反復タスクをインテリジェントに自動化することで、生成 AI は開発タイムラインを短縮し、一貫性を向上させ、人為的ミスを減らします。その結果、ソフトウェア出力の品質が向上します。

開発エクスペリエンスを定期的に見直し、反復する

ソフトウェア開発エクスペリエンス自体は、継続的な改良が必要な製品として扱う必要があります。これには、開発ライフサイクル、ツール、プラクティスのあらゆる側面を定期的に見直し、反復するための体系的なプロセスを確立することが含まれます。ツールチェーン、ワークフロー、プロセス全体を定期的に変更します。製品マネージャー、デザイナー、アーキテクト、開発者、テスター、運用担当者など、さまざまな役割のすべてのチームメンバーからフィードバックを収集します。問題点、ボトルネック、強化の機会を特定するように依頼します。たとえば、チームは CI/CD パイプラインのパフォーマンスを四半期ごとにレビューし、ビルド時間、デプロイ頻度、エラー率などのメトリクスを分析して、最適化が必要な領域を特定できます。生成 AI 機能は急速に進化し続けるため、ワークフローをさらに合理化したり、SDLC のすべてのロールで機能を強化したりする可能性のある新しい AI を活用したツールや機能を一貫して評価することが重要です。

効果的なプロジェクト管理プラクティスの採用

複雑なソフトウェア開発作業を効果的に調整するには、AI で強化されたプロジェクト管理プラクティスを採用します。このコンテキストでは、効果的なプロジェクト管理は従来の方法論を超えています。SDLC 全体の計画、実行、モニタリングを強化する AI 拡張アプローチを採用しています。アジャイルフレームワークは柔軟性、コラボレーション、迅速な反復を促進し、生成 AI を使用してこれらのプロセスを最適化できます。例えば、生成 AI は過去のプロジェクトデータを分析してより正確な見積りを行い、ビジネス目標と顧客のフィードバックに基づいてユーザーストーリーを自動的に生成して優先順位付けし、チームのパフォーマンスに関するインテリジェントなインサイトを提供できます。AI を活用したプロジェクト管理ツールは、潜在的な障害を予測し、チームメンバーのスキルとワークロードに基づいて最適なタスク割り当てを提案できます。AI を活用した機能をプロジェ

クト管理プラクティスに統合することで、可視性を高め、データ駆動型の意味決定を迅速に行い、チームメンバーが共通の目標に向けて連携し、効率的に作業できるようになります。

ナレッジ管理の実装

AI を活用したソフトウェア開発エクスペリエンスが成熟したら、堅牢なナレッジ管理システムを実装します。堅牢なナレッジ管理システムは、貴重なインサイト、ベストプラクティス、ソリューションの取得、整理、アクセス許可の付与に役立ちます。SDLC のすべてのチームメンバーは、システムに簡単にアクセスできます。生成 AI を使用して、組織とともに進化する動的でインテリジェントなナレッジベースを作成します。例えば、AI はコードの変更、会話、プロジェクトアーティファクトに基づいてドキュメントを自動的に生成および更新できるため、手動による介入なしに情報が最新のままになります。生成 AI はインテリジェントな検索機能を強化し、チームメンバーが正確な用語を知らなくても、自然言語クエリを使用して関連情報をすばやく見つけるのに役立ちます。さらに、生成 AI は、現在のタスクや課題に基づいて、関連情報をチームメンバーにプロアクティブに提示できます。これは、すべてのロールの意味決定と問題解決を強化する仮想の指導者として機能します。AI を活用したナレッジ管理システムを実装することで、サイロを壊し、オンボーディングを加速し、冗長な作業を減らし、ソフトウェア開発チーム全体で継続的な学習とイノベーションの文化を育むことができます。

拡張性とカスタマイズの提供

ソフトウェア開発における生成 AI の利点を最大化するには、AI を活用したツールとプラットフォームが拡張可能でカスタマイズ可能であることを確認してください。これにより、AI 機能を特定のニーズ、ワークフロー、テクノロジースタックに合わせて調整できます。たとえば、独自のコードベースとドキュメントで AI モデルを微調整したり、特定のタスク用のカスタム AI 搭載ツールを作成したり、AI 機能を既存のツールやプロセスに統合したりできます。この拡張性は、AI を活用した開発エクスペリエンスを進化させ、組織の変化するニーズを満たすのに役立ちます。また、特定のドメインまたはプロジェクトタイプのエクスペリエンスを最適化するのに役立ちます。

オペレーションの最適化

生成 AI は、ソフトウェアオペレーションとメンテナンスの最適化に重要な役割を果たします。AI 機能を運用ツールとプロセスに統合することで、運用に最適化します。例えば、生成 AI を使用して、ログデータをほぼリアルタイムで分析し、潜在的なシステム障害を予測し、定期的なメンテナンスタスクを自動化します。生成 AI は、複雑な分散システム間でイベントを相関させることで、根本原因の分析にも役立ちます。これにより、システムの信頼性が向上し、ダウンタイムが短縮され、運用チームがより戦略的なイニシアチブに集中できるようになります。

データ駆動型インサイトの使用

AI を活用した開発ジャーニー全体で、データ駆動型のインサイトを使用します。SDLC のすべてのステージからデータを収集、分析、処理するシステムを実装します。これには、コードメトリクス、テスト結果、デプロイデータ、ユーザーフィードバック、運用パフォーマンスが含まれます。生成 AI を使用して、人間のオブザーバーには明らかではないパターンやインサイトを発見します。次に、これらのインサイトを開発プロセスにフィードバックし、アーキテクチャ上の決定から機能の優先順位付けまで、あらゆる情報を提供します。

プラットフォームベースのアプローチを採用する

ソフトウェア開発における生成 AI の利点を最大限に活用するには、プラットフォームベースのアプローチを採用します。SDLC のすべての側面に AI 機能を組み込む包括的な統合プラットフォームを作成します。プラットフォームは、一貫したユーザーエクスペリエンス、一元的な管理とデータ、さまざまなツールとプロセス間のシームレスな統合を提供する必要があります。これにより、AI の利点が組織全体で一様に利用可能になり、複数の異なる AI ツールを管理するオーバーヘッドが削減され、AI 機能の継続的な改善と拡張のための基盤が提供されます。

ソフトウェア開発における生成 AI の成功を測定する

生成 AI を活用したソフトウェア開発エクスペリエンスを実装した場合の効果を効果的に測定するには、ソフトウェア開発ライフサイクル (SDLC) のさまざまな側面にまたがる包括的なメトリクスセットを確立する必要があります。これらのメトリクスは、効率と生産性の即時の改善を把握し、ソフトウェア品質、チームの満足度、ビジネス価値の長期的な向上を反映する必要があります。

このセクションで推奨されるメトリクスを効果的に使用するには、以下を実行します。

1. ベースラインを確立する – AI を活用した開発エクスペリエンスの実装を開始する前に、これらのメトリクス全体で現在のパフォーマンスに関する包括的なデータを収集してください。これにより、明確な出発点が提供され、後で意味のある比較を行うのに役立ちます。
2. 現実的なターゲットを設定する – ベースラインを手元に用意し、メトリクスごとに達成可能な改善目標を設定します。野心的でありながら現実的であること。持続可能な進行状況は、多くの場合増分であることに注意してください。
3. 継続的モニタリングの実装 – 自動ツールを使用して、環境内のこれらのメトリクスのデータを絶えず収集および分析します。ほぼリアルタイムのモニタリングは、進捗状況をモニタリングし、問題や機会をすばやく特定するのに役立ちます。
4. 定期的なレビューの実施 – 自分とチームが目標に対する進捗状況を徹底的に評価する四半期または半期のレビューセッションをスケジュールします。これらのセッションを使用して、さらなる改善が必要な分野を特定し、成功を祝います。
5. 反復と調整 — 得られたインサイトに基づいて、生成 AI の実装を継続的に改善し、必要に応じてターゲットを調整します。

このセクションでは、以下のカテゴリのメトリクスについて説明します。

- [デプロイ速度](#)
- [コード品質](#)
- [運用効率](#)
- [チームの生産性と満足度](#)
- [ビジネスへの影響](#)

デプロイ速度

以下のデプロイ速度メトリクスを測定することを検討してください。

メトリクス	説明
市場投入までの時間	アイデアの構想から本番デプロイまでの時間を短縮する
スプリント速度	チームによってスプリントごとに完了したストーリーポイントの増加を追跡する
コードコミット頻度	開発サイクルの加速を示すコードコミットの増加をモニタリングする
プルリクエストの解決時間	リポジトリ内のコード変更を確認およびマージするのにかかる時間の減少を評価する
リリース速度	四半期または年あたりのリリース数の増加を測定する

コード品質

次のコード品質メトリクスを測定することを検討してください。

メトリクス	説明
欠陥密度	ソフトウェアバグの削減を測定する
コードカバレッジ	コードベース全体のテストカバレッジ率の増加を追跡する
技術的負債	特定された技術的負債の経時的な減少をモニタリングする
静的コード分析スコア	自動分析ツールに基づいてコード品質の改善を評価する

運用効率

以下の運用効率メトリクスを測定することを検討してください。

メトリクス	説明
デプロイ頻度	成功したデプロイの数の増加を測定する
平均復旧時間 (MTTR)	システム障害からの復旧にかかる時間の削減を追跡する
変更失敗率	デプロイの失敗につながる変更の割合の減少をモニタリングする

チームの生産性と満足度

以下のチームの生産性と満足度のメトリクスを測定することを検討してください。

メトリクス	説明
生産性の向上	各タスクの生産性の向上率をモニタリングする
満足度スコア	定期的なアンケートを実施して、チームの士気と仕事への満足度の向上を評価します。
ナレッジ共有の効率	チームが情報の検索や反復的な質問に費やす時間の削減を測定する
オンボーディング時間	新しいチームメンバーが生産性を上げるために必要な時間の減少を追跡する

ビジネスへの影響

以下のビジネスへの影響メトリクスを測定することを検討してください。

メトリクス	説明
機能の採用率	リリースした新機能によるユーザーエンゲージメントの増加を測定する

メトリクス	説明
顧客満足度スコア	ユーザーのフィードバックと評価の改善を追跡する
収益への影響 (直接的および間接的)	リリース速度の向上または生産性の向上に起因する収益の増加を評価する

結論

この戦略ドキュメントでは、生成 AI を活用したソフトウェア開発エクスペリエンスの概要を説明します。[55-フレームワークの 5 つのディメンション](#)である、調査、統合、対話、反復、影響について説明します。これらのディメンションは、ソフトウェア開発ライフサイクル (SDLC) 全体で生成 AI を統合するための戦略的ロードマップを提供します。また、このフレームワークを正常に実装するために必要な[基本的な機能](#)についても説明します。機能は、プロジェクト管理、DevSecOps、AI アシスタント、ナレッジ管理などの分野にまたがります。生成 AI を統合する際に考慮すべき[ベストプラクティス](#)を提供し、[メトリクス](#)を使用して生成 AI がソフトウェア開発エクスペリエンスに与える影響を測定するのに役立ちます。

生成 AI をソフトウェア開発プロセスに統合することは、イノベーションを加速し、品質を向上させ、生産性を高める可能性のあるパラダイムシフトを表しています。ただし、これは 1 回限りの実装ではないことを認識することが重要です。これは、持続的な労力と継続的な改良を必要とする継続的な進化です。

このジャーニーを開始するときは、まず組織の現在の機能と準備状況を徹底的に評価することをお勧めします。[AWS 評価ツール](#)は AI を活用したソフトウェア開発評価ツールで、優先度の高い分野を特定し、カスタマイズされた実装ロードマップを作成するのに役立ちます。

リソース

主要な優先順位領域を特定したら、以下のリソースがロードマップの実装に役立ちます。

AWS ドキュメント

- [Amazon Bedrock を使用して AWS インフラストラクチャオペレーションを自動化する](#) (AWS 規範ガイダンス)
- [インラインコード生成とアシスタントコード生成のための Amazon Q Developer のベストプラクティス](#) (AWS 規範ガイダンス)
- [Amazon Bedrock エージェントとナレッジベースを使用して、完全に自動化されたチャットベースのアシスタントを開発する](#) (AWS 規範ガイダンス)
- [生成 AI AWS を使用したでのアプリケーション開発およびメンテナンス運用モデルの変換](#) (AWS 規範ガイダンス)
- [Amazon Q Developer をコーディングアシスタントとして使用して生産性を高める](#) (AWS 規範ガイダンス)

AWS ブログ投稿とチュートリアル

- [Amazon Q ブログ記事](#)
- [Amazon Q でソフトウェア開発ライフサイクルを加速する](#) (AWS ブログ記事)
- [ソリューションアーキテクト AI エージェントの構築 AWS : Amazon Bedrock を活用してアーキテクチャとデプロイを自動化する](#) (AWS ビデオ)
- [生成 AI を活用したテクノロジーオペレーション](#) (AWS ブログ記事)
- [Amazon Q Developer を使用して Java アプリケーションをモダナイズする](#) (AWS ブログ記事)
- [Amazon Bedrock を使用してソフトウェア開発パイプラインのコードを生成、評価、理解する](#) (AWS ブログ記事)

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#) をサブスクライブできます。

変更	説明	日付
初版発行	該当しない	2025 年 4 月 18 日

AWS 規範ガイドの用語集

以下は、AWS 規範ガイドによって提供される戦略、ガイド、パターンで一般的に使用される用語です。エントリを提案するには、用語集の最後のフィードバックの提供リンクを使用します。

数字

7 Rs

アプリケーションをクラウドに移行するための 7 つの一般的な移行戦略。これらの戦略は、ガートナーが 2011 年に特定した 5 Rs に基づいて構築され、以下で構成されています。

- リファクタリング/アーキテクチャの再設計 — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更します。これには、通常、オペレーティングシステムとデータベースの移植が含まれます。例: オンプレミスの Oracle データベースを Amazon Aurora PostgreSQL 互換エディションに移行する。
- リプラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの Oracle 用の Amazon Relational Database Service (Amazon RDS) に移行する。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: 顧客関係管理 (CRM) システムを Salesforce.com に移行する。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの EC2 インスタンス上の Oracle に移行する。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) — 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例: Microsoft Hyper-Vアプリケーションをに移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを行き移るためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。
- 廃止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

「[属性ベースのアクセス制御](#)」をご覧ください。

抽象化されたサービス

「[マネージドユーザー](#)」をご覧ください。

ACID

「[原子性、一貫性、分離性、耐久性 \(ACID\)](#)」をご覧ください。

アクティブ/アクティブ移行

(双方向レプリケーションツールまたは二重書き込み操作を使用して) ソースデータベースとターゲットデータベースを同期させ、移行中に両方のデータベースが接続アプリケーションからのトランザクションを処理するデータベース移行方法。この方法では、1 回限りのカットオーバーの必要がなく、管理された小規模なバッチで移行できます。[アクティブ/パッシブ移行](#)よりも柔軟な方法ですが、さらに多くの作業が必要となります。

アクティブ/パッシブ移行

ソースデータベースとターゲットデータベースを同期させながら、データがターゲットデータベースにレプリケートされている間、接続しているアプリケーションからのトランザクションをソースデータベースのみで処理するデータベース移行方法。移行中、ターゲットデータベースはトランザクションを受け付けません。

集計関数

複数行に処理を行い、グループ全体を対象に単一の戻り値を計算する SQL 関数。集計関数の例としては、SUM や MAX などがあります。

AI

「[人工知能](#)」をご覧ください。

AIOps

「[AI オペレーション](#)」をご覧ください。

匿名化

データセット内の個人情報を完全に削除するプロセス。匿名化は個人のプライバシー保護に役立ちます。匿名化されたデータは、もはや個人データとは見なされません。

アンチパターン

繰り返し起こる問題に対して頻繁に用いられる解決策で、その解決策が逆効果であったり、効果がなかったり、代替案よりも効果が低かったりするもの。

アプリケーション制御

マルチテナントからシステムを保護するために、承認されたアプリケーションのみを使用できるようにするセキュリティアプローチ。

アプリケーションポートフォリオ

アプリケーションの構築と維持にかかるコスト、およびそのビジネス価値を含む、組織が使用する各アプリケーションに関する詳細情報の集まり。この情報は、[ポートフォリオの検出と分析プロセス](#)の重要な要素であり、移行、モダナイズ、最適化するアプリケーションを特定し、優先順位を付けるのに役立ちます。

人工知能 (AI)

コンピューティングテクノロジーを使用し、学習、問題の解決、パターンの認識など、通常は人間に関連づけられる認知機能の実行に特化したコンピュータサイエンスの分野。詳細については、「[人工知能 \(AI\) とは何ですか?](#)」をご覧ください。

AI オペレーション (AIOps)

機械学習技術を使用して運用上の問題を解決し、運用上のインシデントと人の介入を減らし、サービス品質を向上させるプロセス。AWS 移行戦略での AIOps の使用方法については、[オペレーション統合ガイド](#)を参照してください。

非対称暗号化

暗号化用のパブリックキーと復号用のプライベートキーから成る 1 組のキーを使用した、暗号化のアルゴリズム。パブリックキーは復号には使用されないため共有しても問題ありませんが、プライベートキーの利用は厳しく制限する必要があります。

原子性、一貫性、分離性、耐久性 (ACID)

エラー、停電、その他の問題が発生した場合でも、データベースのデータ有効性と運用上の信頼性を保証する一連のソフトウェアプロパティ。

属性ベースのアクセス制御 (ABAC)

部署、役職、チーム名など、ユーザーの属性に基づいてアクセス許可をきめ細かく設定する方法。詳細については、AWS Identity and Access Management (IAM) ドキュメントの「[ABAC AWS](#)」を参照してください。

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリーバージョンを保存する場所。匿名化、編集、仮名化など、データを処理または変更する目的で、信頼できるデータソースから他の場所にデータをコピーすることができます。

アベイラビリティゾーン (AZ)

他のアベイラビリティゾーンの障害から AWS リージョン 隔離され、同じリージョン内の他のアベイラビリティゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドへの移行を成功させるための効率的で効果的な計画を立て AWS するための、のガイドラインとベストプラクティスのフレームワークです。AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを整理しています。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、AWS CAF は、クラウド導入を成功させるための組織の準備に役立つ人材開発、トレーニング、コミュニケーションに関するガイダンスを提供します。詳細については、[AWS CAF ウェブサイト](#)と [AWS CAF のホワイトペーパー](#) を参照してください。

AWS ワークロード認定フレームワーク (AWS WQF)

データベース移行ワークロードを評価し、移行戦略を推奨し、作業見積もりを提供するツール。AWS WQF は AWS Schema Conversion Tool (AWS SCT) に含まれています。データベーススキーマとコードオブジェクト、アプリケーションコード、依存関係、およびパフォーマンス特性を分析し、評価レポートを提供します。

B

不正なボット

個人や組織に混乱や損害を与えることを目的とした [ボット](#)。

BCP

「[ビジネス継続性計画 \(BCP\)](#)」をご覧ください。

動作グラフ

リソースの動作とインタラクションを経時的に示した、一元的なインタラクティブビュー。Amazon Detective の動作グラフを使用すると、失敗したログオンの試行、不審な API 呼び出し、その他同様のアクションを調べることができます。詳細については、Detective ドキュメントの「[動作グラフのデータ](#)」を参照してください。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「[エンディアン性](#)」もご覧ください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの一つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

要素がセットのメンバーであるかどうかをテストするために使用される、確率的でメモリ効率の高いデータ構造。

ブルー/グリーンデプロイ

それぞれが独立しているが、同一の環境を 2 つ作成するデプロイ戦略。現在のアプリケーションバージョンを 1 つの環境 (ブルー) で実行し、新しいアプリケーションバージョンを別の環境 (グリーン) で実行します。この戦略は、最小限の影響で迅速にロールバックするのに役立ちます。

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティややり取りをシミュレートするソフトウェアアプリケーション。インターネット上の情報のインデックスを作成するウェブクロウラーなど、一部のボットは有用または有益です。悪質なボットと呼ばれる他のボットの中には、個人や組織を混乱させたり、損害を与えたりすることを意図したものもあります。

ボットネット

[マルウェア](#)に感染しており、ボットハーダーまたはボットオペレーターと呼ばれる単一の当事者によって制御されている[ボット](#)のネットワーク。ボットネットは、ボットとその影響力を拡大する仕組みとして、非常によく知られています。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといます。既存のブランチから新しいブランチを作成し、その新しいブランチで機能を開発した

り、バグを修正したりできます。機能を構築するために作成するブランチは、通常、機能ブランチと呼ばれます。機能をリリースする準備ができたなら、機能ブランチをメインブランチに統合します。詳細については、「[ブランチの概要](#)」(GitHub ドキュメント)を参照してください。

ブレイクグラスアクセス

例外的な状況では、承認されたプロセスを通じて、ユーザーが AWS アカウント 通常アクセス許可を持たない にすばやくアクセスできるようにします。詳細については、AWS Well-Architected ガイドの「[ブレイクグラス手順の実装](#)」インジケータを参照してください。

ブラウフィールド戦略

環境の既存インフラストラクチャ。システムアーキテクチャにブラウフィールド戦略を導入する場合、現在のシステムとインフラストラクチャの制約に基づいてアーキテクチャを設計します。既存のインフラストラクチャを拡張している場合は、ブラウフィールド戦略と[グリーンフィールド](#)戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

価値を生み出すためにビジネスが行うこと (営業、カスタマーサービス、マーケティングなど)。マイクロサービスのアーキテクチャと開発の決定は、ビジネス能力によって推進できます。詳細については、[AWSでのコンテナ化されたマイクロサービスの実行](#)ホワイトペーパーの「[ビジネス機能を中心に組織化](#)」セクションを参照してください。

ビジネス継続性計画 (BCP)

大規模移行など、中断を伴うイベントが運用に与える潜在的な影響に対処し、ビジネスを迅速に再開できるようにする計画。

C

CAF

「[AWS クラウド導入フレームワーク](#)」を参照してください

カナリアデプロイ

エンドユーザーへのバージョンリリースを、時間をかけて段階的に行うこと。確信が持てたら新規バージョンをデプロイして、現在のバージョン全体を置き換えます。

CCoE

「[Cloud Center of Excellence](#)」を参照してください。

CDC

「[変更データキャプチャ](#)」を参照してください。

変更データキャプチャ (CDC)

データソース (データベーステーブルなど) の変更を追跡し、その変更に関するメタデータを記録するプロセス。CDC は、ターゲットシステムでの変更を監査またはレプリケートして同期を維持するなど、さまざまな目的に使用できます。

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストすること。[AWS Fault Injection Service \(AWS FIS\)](#) を使用して、AWS ワークロードにストレスを与え、その応答を評価する実験を実行できます。

CI/CD

「[継続的インテグレーションと継続的デリバリー](#)」を参照してください。

分類

予測を生成するのに役立つ分類プロセス。分類問題の機械学習モデルは、離散値を予測します。離散値は、常に互いに区別されます。例えば、モデルがイメージ内に車があるかどうかを評価する必要がある場合があります。

クライアント側の暗号化

ターゲットがデータ AWS のサービスを受信する前のローカルでのデータの暗号化。

Cloud Center of Excellence (CCoE)

クラウドのベストプラクティスの作成、リソースの移動、移行のタイムラインの確立、大規模変革を通じて組織をリードするなど、組織全体のクラウド導入の取り組みを推進する学際的なチーム。詳細については、AWS クラウド エンタープライズ戦略ブログの [CCoE 投稿](#) を参照してください。

クラウドコンピューティング

リモートデータストレージと IoT デバイス管理に通常使用されるクラウドテクノロジー。クラウドコンピューティングは、一般的に、[エッジコンピューティング](#)に接続されています。

クラウド運用モデル

IT 組織において、1 つ以上のクラウド環境を構築、成熟、最適化するために使用される運用モデル。詳細については、「[クラウド運用モデルの構築](#)」を参照してください。

導入のクラウドステージ

組織が、AWS クラウドへの移行時に通常実行する 4 つの段階。

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーン作成、CCoE の定義、運用モデルの確立など)
- 移行 — 個々のアプリケーションの移行
- 再発明 — 製品とサービスの最適化、クラウドでのイノベーション

これらのステージは、AWS クラウド エンタープライズ戦略ブログのブログ記事「[クラウドファーストへのジャーニー](#)」と「[導入のステージ](#)」で Stephen Orban によって定義されました。移行戦略との関連性については、AWS「[移行準備ガイド](#)」を参照してください。

CMDB

「[構成管理データベース \(CMDB\)](#)」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、GitHub や Bitbucket Cloud があります。コードの各バージョンはブランチと呼ばれます。マイクロサービスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

空である、または、かなり空きがある、もしくは、古いデータや無関係なデータが含まれているバッファキャッシュ。データベースインスタンスはメインメモリまたはディスクから読み取る必要があり、バッファキャッシュから読み取るよりも時間がかかるため、パフォーマンスに影響します。

コールドデータ

めったにアクセスされず、通常は過去のデータです。この種類のデータをクエリする場合、通常は低速なクエリでも問題ありません。このデータを低パフォーマンスで安価なストレージ階層またはクラスに移動すると、コストを削減することができます。

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオといった、ビジュアル形式の情報を分析および抽出する [AI](#) の分野。例えば、Amazon SageMaker AI では、CV 用の画像処理アルゴリズムを利用できます。

設定ドリフト

ワークロードにおいて、設定が想定した状態から変化すること。これによって、ワークロードが非準拠になる可能性があります。この状態は、徐々に生じ、意図的なものではありません。

構成管理データベース (CMDB)

データベースとその IT 環境 (ハードウェアとソフトウェアの両方のコンポーネントとその設定を含む) に関する情報を保存、管理するリポジトリ。通常、CMDB のデータは、移行のポートフォリオの検出と分析の段階で使用します。

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。YAML テンプレートを使用して、コンフォーマンスパックを AWS アカウント および リージョンの単一のエンティティとしてデプロイすることも、組織全体にデプロイすることもできます。詳細については、AWS Config ドキュメントの「[コンフォーマンスパック](#)」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「[継続的デリバリーの利点](#)」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「[継続的デリバリーと継続的なデプロイ](#)」を参照してください。

CV

[「コンピュータビジョン」](#) を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

ネットワーク内のデータを重要度と機密性に基づいて識別、分類するプロセス。データに適した保護および保持のコントロールを判断する際に役立つため、あらゆるサイバーセキュリティのリスク管理戦略において重要な要素です。データ分類は、AWS Well-Architected フレームワークのセキュリティの柱のコンポーネントです。詳細については、「[データ分類](#)」を参照してください。

データドリフト

実稼働データと ML モデルのトレーニングに使用されたデータとの間に有意な差異が生じたり、入力データが時間の経過と共に有意に変化したりすることです。データドリフトは、ML モデル予測の全体的な品質、精度、公平性を低下させる可能性があります。

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

非一元的で分散型のデータ所有権を持つとともに、一元的な管理およびガバナンスを行えるアーキテクチャフレームワーク。

データ最小化

厳密に必要なデータのみを収集し、処理するという原則。でデータ最小化を実践 AWS クラウドすることで、プライバシーリスク、コスト、分析のカーボンフットプリントを削減できます。

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、「[AWS でのデータ境界の構築](#)」を参照してください。

データの前処理

raw データをお客様の機械学習モデルで簡単に解析できる形式に変換すること。データの前処理とは、特定の列または行を削除して、欠落している、矛盾している、または重複する値に対処することを意味します。

データ出所

データの生成、送信、保存の方法など、データのライフサイクル全体を通じてデータの出所と履歴を追跡するプロセス。

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには、一般的に、大量の履歴データが含まれており、多くの場合、それらはクエリや分析に使用されます。

データベース定義言語 (DDL)

データベース内のテーブルやオブジェクトの構造を作成または変更するためのステートメントまたはコマンド。

データベース操作言語 (DML)

データベース内の情報を変更 (挿入、更新、削除) するためのステートメントまたはコマンド。

DDL

「[データベース定義言語](#)」を参照してください。

ディープアンサンブル

予測のために複数の深層学習モデルを組み合わせます。ディープアンサンブルを使用して、より正確な予測を取得したり、予測の不確実性を推定したりできます。

深層学習

人工ニューラルネットワークの複数層を使用して、入力データと対象のターゲット変数の間のマッピングを識別する機械学習サブフィールド。

多層防御

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリティの手法。この戦略を に採用するときは AWS、リソースの保護に役立つように、AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加します。たとえば、多層防御アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

では AWS Organizations、互換性のあるサービスが AWS メンバーアカウントを登録して組織のアカウントを管理し、そのサービスのアクセス許可を管理できます。このアカウントを、そのサービスの委任管理者と呼びます。詳細、および互換性のあるサービスの一覧は、AWS

Organizations ドキュメントの「[AWS Organizationsで利用できるサービス](#)」を参照してください。

トラブルシューティング

アプリケーション、新機能、コードの修正をターゲットの環境で利用できるようにするプロセス。デプロイでは、コードベースに変更を施した後、アプリケーションの環境でそのコードベースを構築して実行します。

開発環境

「[環境](#)」を参照してください。

検出管理

イベントが発生したときに、検出、ログ記録、警告を行うように設計されたセキュリティコントロール。これらのコントロールは副次的な防衛手段であり、実行中の予防的コントロールをすり抜けたセキュリティイベントをユーザーに警告します。詳細については、「AWSでのセキュリティコントロールの実装」の「[検出的コントロール](#)」を参照してください。

開発バリューストリームマッピング (DVSM)

ソフトウェア開発ライフサイクルのスピードと品質に悪影響を及ぼす制約を特定し、優先順位を付けるために使用されるプロセス。DVSM は、もともとリーンマニユファクチャリング・プラクティスのために設計されたバリューストリームマッピング・プロセスを拡張したものです。ソフトウェア開発プロセスを通じて価値を創造し、動かすために必要なステップとチームに焦点を当てています。

デジタルツイン

建物、工場、産業機器、生産ラインなど、現実世界のシステムを仮想的に表現したものです。デジタルツインは、予知保全、リモートモニタリング、生産最適化をサポートします。

ディメンションテーブル

[スタースキーマ](#)において、ファクトテーブルの定量データに関するデータ属性が含まれる小さいテーブル。ディメンションテーブルの属性は、通常、テキストフィールド、またはテキストのように扱える個別の数値で示されます。これらの属性は、一般的に、クエリの制約、フィルタリング、結果セットのラベル付けに使用されます。

デザスタ

ワークロードまたはシステムが、導入されている主要な場所でのビジネス目標の達成を妨げるイベント。これらのイベントは、自然災害、技術的障害、または意図しない設定ミスやマルウェア攻撃などの人間の行動の結果である場合があります。

ディザスタリカバリ (DR)

[ディザスタ](#)によるダウンタイムとデータ損失を最小限に抑えるための戦略とプロセス。詳細については、AWS Well-Architected フレームワークの「[でのワークロードのディザスタリカバリ](#)」[AWS: クラウドでのリカバリ](#)」を参照してください。

DML

「[データベース操作言語](#)」を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計:ソフトウェアの中心における複雑さへの取り組み) で紹介されています (ポストン: Addison-Wesley Professional、2003)。strangler fig パターンでドメイン駆動型設計を使用する方法の詳細については、「[コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET \(ASMX\) ウェブサービスを段階的にモダナイズ](#)」を参照してください。

DR

「[ディザスタリカバリ](#)」を参照してください。

ドリフト検出

ベースライン設定からの偏差を追跡します。たとえば、AWS CloudFormation を使用して[システムリソースのドリフトを検出](#)したり、を使用して AWS Control Tower、ガバナンス要件への準拠に影響する[ランディングゾーンの変更を検出](#)したりできます。

DVSM

「[開発バリューストリームマッピング](#)」を参照してください。

E

EDA

「[探索的データ分析](#)」を参照してください。

EDI

「[電子データ交換](#)」を参照してください。

エッジコンピューティング

IoT ネットワークのエッジにあるスマートデバイスの計算能力を高めるテクノロジー。[クラウドコンピューティング](#)と比較すると、エッジコンピューティングは通信レイテンシーを短縮し、応答時間を改善できます。

電子データ交換 (EDI)

組織間で行う、ビジネスドキュメントの自動交換。詳細については、[「電子データ交換とは」](#)を参照してください。

暗号化

人間が読み取り可能なプレーンテキストデータを暗号文に変換するコンピューティング処理。

暗号化キー

暗号化アルゴリズムが生成した、ランダム化されたビットからなる暗号文字列。キーの長さは決まっておらず、各キーは予測できないように、一意になるように設計されています。

エンディアン

コンピュータメモリにバイトが格納される順序。ビッグエンディアンシステムでは、最上位バイトが最初に格納されます。リトルエンディアンシステムでは、最下位バイトが最初に格納されます。

エンドポイント

[「サービスエンドポイント」](#)を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) 内でホストして、他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これらのアカウントまたはプリンシパルは、インターフェイス VPC エンドポイントを作成することで、エンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon VPC) ドキュメントの [「エンドポイントサービスを作成する」](#)を参照してください。

エンタープライズリソースプランニング (ERP)

エンタープライズの主要なビジネスプロセス (会計、[MES](#)、プロジェクト管理など) を自動化および管理するシステム。

エンベロープ暗号化

暗号化キーを、別の暗号化キーを使用して暗号化するプロセス。詳細については、AWS Key Management Service (AWS KMS) ドキュメントの「[エンベロープ暗号化](#)」を参照してください。

環境

実行中のアプリケーションのインスタンス。クラウドコンピューティングにおける一般的な環境の種類は以下のとおりです。

- **開発環境** — アプリケーションのメンテナンスを担当するコアチームのみが利用できる、実行中のアプリケーションのインスタンス。開発環境は、上位の環境に昇格させる変更をテストするときに使用します。このタイプの環境は、テスト環境と呼ばれることもあります。
- **下位環境** — 初期ビルドやテストに使用される環境など、アプリケーションのすべての開発環境。
- **本番環境** — エンドユーザーがアクセスできる、実行中のアプリケーションのインスタンス。CI/CD パイプラインでは、本番環境が最後のデプロイ環境になります。
- **上位環境** — コア開発チーム以外のユーザーがアクセスできるすべての環境。これには、本番環境、本番前環境、ユーザー承認テスト環境などが含まれます。

エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、AWS CAF セキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。AWS 移行戦略のエピックの詳細については、[プログラム実装ガイド](#)を参照してください。

ERP

「[エンタープライズリソース計画](#)」を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、統計の概要を計算し、データの可視化を作成することによって実行されます。

F

ファクトテーブル

[スタースキーマ](#)の中央にあるテーブル。ビジネスオペレーションに関する定量的データが保存されます。一般的に、ファクトテーブルは、2種類の列で構成されます。1つは測定値が含まれる列、もう1つはディメンションテーブルへの外部キーが含まれる列です。

フェイルファスト

開発ライフサイクルを短縮するために、頻繁かつ段階的にテストを行う哲学であり、アジャイルアプローチでは、この考え方がきわめて重要です。

障害分離境界

では AWS クラウド、障害の影響を制限し、ワークロードの耐障害性を高めるのに役立つアベイラビリティゾーン AWS リージョン、コントロールプレーン、データプレーンなどの境界。詳細については、「[AWS 障害分離境界](#)」を参照してください。

機能ブランチ

「[ブランチ](#)」を参照してください。

特徴量

お客様が予測に使用する入力データ。例えば、製造コンテキストでは、特徴量は製造ラインから定期的にキャプチャされるイメージの可能性もあります。

特徴量重要度

モデルの予測に対する特徴量の重要性。これは通常、Shapley Additive Deskonations (SHAP) や積分勾配など、さまざまな手法で計算できる数値スコアで表されます。詳細については、「[を使用した機械学習モデルの解釈可能性 AWS](#)」を参照してください。

機能変換

追加のソースによるデータのエンリッチ化、値のスケーリング、単一のデータフィールドからの複数の情報セットの抽出など、機械学習プロセスのデータを最適化すること。これにより、機械学習モデルはデータの恩恵を受けることができます。例えば、「2021-05-27 00:15:37」の日付を「2021年」、「5月」、「木」、「15」に分解すると、学習アルゴリズムがさまざまなデータコンポーネントに関連する微妙に異なるパターンを学習するのに役立ちます。

数ショットプロンプト

[LLM](#) に、タスクと望ましい出力を示す例を少数提示した後に、類似のタスクを実行させること。この手法は、プロンプトに記述された例(ショット)からモデルが学習する「インコンテキスト学

習」の一種です。数ショットプロンプトは、特定のフォーマット、推論、専門知識が必要なタスクに効果的です。「[ゼロショットプロンプト](#)」も参照してください。

FGAC

「[きめ細かなアクセス制御](#)」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

[変更データのキャプチャ](#)による継続的なデータ複製を利用して、段階的なアプローチではなく、可能な限り短時間でデータを移行するデータベース移行方法。目的はダウンタイムを最小限に抑えることです。

FM

「[基盤モデル](#)」を参照してください。

基盤モデル (FM)

大規模な深層学習ニューラルネットワークであり、一般化およびラベル付けされていないデータからなる大規模データセットでトレーニングされています。FMにより、言語理解、テキストおよび画像生成、自然言語での会話といった、一般的な各種タスクを実行できます。詳細については、「[基盤モデルとは何ですか?](#)」を参照してください。

G

生成 AI

[AI](#) モデルのサブセット。大量のデータでトレーニングされており、シンプルなテキストプロンプトを使用して、画像、動画、テキスト、オーディオなどの新しいコンテンツやアーティファクトを作成できます。詳細については、「[生成 AI とは何ですか?](#)」を参照してください。

ジオブロッキング

「[地理的制限](#)」を参照してください。

地理的制限 (ジオブロッキング)

特定の国のユーザーがコンテンツ配信にアクセスできないようにするための、Amazon CloudFront のオプション。アクセスを許可する国と禁止する国は、許可リストまたは禁止リスト

を使って指定します。詳細については、CloudFront ドキュメントの「[コンテンツの地理的ディストリビューションの制限](#)」を参照してください。

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローは古いと見なされている方法であり、[トランクベースのワークフロー](#)は推奨されている新しい方法です。

ゴールデンイメージ

システムまたはソフトウェアのスナップショットであり、システムまたはソフトウェアの新規インスタンスをデプロイするテンプレートとして使用されます。製造の例で言えば、ゴールデンイメージを使用すると、複数のデバイスにソフトウェアをプロビジョニングして、デバイス製造オペレーションの速度、スケーラビリティ、生産性を向上させることができます。

グリーンフィールド戦略

新しい環境に既存のインフラストラクチャが存在しないこと。システムアーキテクチャにグリーンフィールド戦略を導入する場合、既存のインフラストラクチャ (別名 [ブラウンフィールド](#)) との互換性の制約を受けることなく、あらゆる新しいテクノロジーを選択できます。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

ガードレール

組織単位 (OU) 全般のリソース、ポリシー、コンプライアンスを管理するのに役立つ概略的なルール。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。これらは、サービスコントロールポリシーと IAM アクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは AWS Config、AWS Security Hub CSPM、Amazon GuardDuty、AWS Trusted Advisor Amazon Inspector、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「[高可用性](#)」を参照してください。

異種混在データベースの移行

別のデータベースエンジンを使用するターゲットデータベースへお客様の出典データベースの移行 (例えば、Oracle から Amazon Aurora)。異種間移行は通常、アーキテクチャの再設計作業の一部であり、スキーマの変換は複雑なタスクになる可能性があります。[AWS は、スキーマの変換に役立つ AWS SCT を提供します。](#)

高可用性 (HA)

課題や災害が発生した場合に、介入なしにワークロードを継続的に運用できること。HA システムは、自動的にフェイルオーバーし、一貫して高品質のパフォーマンスを提供し、パフォーマンスへの影響を最小限に抑えながらさまざまな負荷や障害を処理するように設計されています。

ヒストリアンのモダナイゼーション

製造業のニーズによりよく応えるために、オペレーションテクノロジー (OT) システムをモダナイズし、アップグレードするためのアプローチ。ヒストリアンは、工場内のさまざまなソースからデータを収集して保存するために使用されるデータベースの一種です。

ホールドアウトデータ

[機械学習](#) モデルのトレーニング用データセットから保留される、ラベル付き履歴データの一部。ホールドアウトデータを使用すると、モデル予測をホールドアウトデータと比較して、モデルのパフォーマンスを評価できます。

同種データベースの移行

お客様の出典データベースを、同じデータベースエンジンを共有するターゲットデータベース (Microsoft SQL Server から Amazon RDS for SQL Server など) に移行する。同種間移行は、通常、リホストまたはリプラットフォーム化の作業の一部です。ネイティブデータベースユーティリティを使用して、スキーマを移行できます。

ホットデータ

リアルタイムデータや最近の翻訳データなど、頻繁にアクセスされるデータ。通常、このデータには高速なクエリ応答を提供する高性能なストレージ階層またはクラスが必要です。

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性が高いため、通常の DevOps のリリースワークフローからは外れた形で実施されます。

ハイパーケア期間

カットオーバー直後、移行したアプリケーションを移行チームがクラウドで管理、監視して問題に対処する期間。通常、この期間は 1~4 日です。ハイパーケア期間が終了すると、アプリケーションに対する責任は一般的に移行チームからクラウドオペレーションチームに移ります。

I

laC

「[Infrastructure as Code](#)」を参照してください。

ID ベースのポリシー

AWS クラウド 環境内のアクセス許可を定義する 1 つ以上の IAM プリンシパルにアタッチされたポリシー。

アイドル状態のアプリケーション

90 日間の平均的な CPU およびメモリ使用率が 5~20% のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するか、オンプレミスに保持するのが一般的です。

IIoT

「[インダストリアル IoT](#)」を参照してください。

イミュータブルインフラストラクチャ

既存インフラストラクチャの更新、パッチ適用、変更などを行わずに、本番環境ワークロードに使用する新規インフラストラクチャをデプロイするモデル。本質的に、イミュータブルインフラストラクチャは、[ミュータブルインフラストラクチャ](#)よりも一貫性、信頼性、予測性に優れています。詳細については、AWS Well-Architected フレームワークにある「[イミュータブルインフラストラクチャを使用してデプロイする](#)」のベストプラクティスを参照してください。

インバウンド (受信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーションの外部からネットワーク接続を受け入れ、検査し、ルーティングする VPC。[AWS Security Reference Architecture](#) では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

I

増分移行

アプリケーションを 1 回ですべてカットオーバーするのではなく、小さい要素に分けて移行するカットオーバー戦略。例えば、最初は少数のマイクロサービスまたはユーザーのみを新しいシステムに移行する場合があります。すべてが正常に機能することを確認できたら、残りのマイクロサービスやユーザーを段階的に移行し、レガシーシステムを廃止できるようにします。この戦略により、大規模な移行に伴うリスクが軽減されます。

インダストリー 4.0

2016 年に [Klaus Schwab](#) 氏が提唱した用語で、接続、リアルタイムデータ、オートメーション、分析、AI/ML の進歩による、ビジネスプロセスのモダナイズを意味します。

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

アプリケーションのインフラストラクチャを一連の設定ファイルを使用してプロビジョニングし、管理するプロセス。IaC は、新しい環境を再現可能で信頼性が高く、一貫性のあるものにするため、インフラストラクチャを一元的に管理し、リソースを標準化し、スケールを迅速に行えるように設計されています。

インダストリアル IoT (IIoT)

製造、エネルギー、自動車、ヘルスケア、ライフサイエンス、農業などの産業部門におけるインターネットに接続されたセンサーやデバイスの使用。詳細については、「[インダストリアル IoT \(IIoT\) デジタルトランスフォーメーション戦略の構築](#)」を参照してください。

インスペクション VPC

AWS マルチアカウントアーキテクチャでは、VPC (同一または異なる 内 AWS リージョン)、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査を管理する一元化された VPCs。 [AWS Security Reference Architecture](#) では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

IoT

インターネットまたはローカル通信ネットワークを介して他のデバイスやシステムと通信する、センサーまたはプロセッサが組み込まれた接続済み物理オブジェクトのネットワーク。詳細については、「[IoT とは](#)」を参照してください。

解釈可能性

機械学習モデルの特性で、モデルの予測がその入力にどのように依存するかを人間が理解できる度合いを表します。詳細については、[「を使用した機械学習モデルの解釈可能性 AWS」](#)を参照してください。

IoT

[「IoT」](#)を参照してください。

IT 情報ライブラリ (ITIL)

IT サービスを提供し、これらのサービスをビジネス要件に合わせるための一連のベストプラクティス。ITIL は ITSM の基盤を提供します。

IT サービス管理 (ITSM)

組織の IT サービスの設計、実装、管理、およびサポートに関連する活動。クラウドオペレーションと ITSM ツールの統合については、[オペレーション統合ガイド](#)を参照してください。

ITIL

[「IT 情報ライブラリ」](#)を参照してください。

ITSM

[「IT サービス管理」](#)を参照してください。

L

ラベルベースアクセス制御 (LBAC)

強制アクセス制御 (MAC) の実装で、ユーザーとデータ自体にそれぞれセキュリティラベル値が明示的に割り当てられます。ユーザーセキュリティラベルとデータセキュリティラベルが交差する部分によって、ユーザーに表示される行と列が決まります。

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロードとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、[「安全でスケーラブルなマルチアカウント AWS 環境のセットアップ」](#)を参照してください。

大規模言語モデル (LLM)

大量のデータで事前トレーニングされた深層学習 [AI](#) モデル。LLM では、質問への回答、ドキュメントの要約、他言語へのテキスト翻訳、文を完成させるなど、さまざまなタスクを実行できます。詳細については、「[大規模言語モデル \(LLM\) とは何ですか?](#)」を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

「[ラベルベースアクセス制御](#)」を参照してください。

最小特権

タスクの実行には必要最低限の権限を付与するという、セキュリティのベストプラクティス。詳細については、IAM ドキュメントの「[最小特権アクセス許可を適用する](#)」を参照してください。

リフトアンドシフト

「[7 Rs](#)」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「[エンディアン性](#)」もご覧ください。

LLM

「[大規模言語モデル](#)」を参照してください。

下位環境

「[環境](#)」を参照してください。

M

機械学習 (ML)

パターン認識と学習にアルゴリズムと手法を使用する人工知能の一種。ML は、モノのインターネット (IoT) データなどの記録されたデータを分析して学習し、パターンに基づく統計モデルを生成します。詳細については、「[機械学習](#)」を参照してください。

メインブランチ

「[ブランチ](#)」を参照してください。

マルウェア

コンピュータのセキュリティやプライバシーを侵害するように設計されたソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスを招く可能性があります。マルウェアの例には、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS のサービスはインフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を運用し、エンドポイントにアクセスしてデータを保存および取得します。マネージドサービスの例として、Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB が挙げられます。このサービスは、抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するソフトウェアシステムであり、工場では、これによって、原材料から製品を完成させます。

MAP

[「Migration Acceleration Program」](#) を参照してください。

メカニズム

ツールを作成してその導入を推進し、導入結果を調べて調整を行うための包括的なプロセス。メカニズムとは、運用中にそれ自体を強化し改善するサイクルを意味します。詳細については、AWS 「Well-Architected フレームワーク」の [「メカニズムの構築」](#) を参照してください。

メンバーアカウント

組織の一部である管理アカウント AWS アカウント 以外のすべて AWS Organizations。アカウントが組織のメンバーになることができるのは、一度に 1 つのみです。

MES

[「製造実行システム」](#) を参照してください。

Message Queuing Telemetry Transport (MQTT)

[発行/サブスクリプション](#) のパターンに基づく、軽量のマシンツーマシン (M2M) 通信プロトコルであり、リソースに限りのある [IoT](#) デバイスに使用されます。

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス

機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、[AWS「サーバーレスサービスを使用したマイクロサービスの統合」](#)を参照してください。

マイクロサービスアーキテクチャ

各アプリケーションプロセスをマイクロサービスとして実行する独立したコンポーネントを使用してアプリケーションを構築するアプローチ。これらのマイクロサービスは、軽量 API を使用して、明確に定義されたインターフェイスを介して通信します。このアーキテクチャの各マイクロサービスは、アプリケーションの特定の機能に対する需要を満たすように更新、デプロイ、およびスケーリングできます。詳細については、「[でのマイクロサービスの実装 AWS](#)」を参照してください。

Migration Acceleration Program (MAP)

組織がクラウドに移行するための強力な運用基盤を構築し、移行の初期コストを相殺するのに役立つコンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。MAP には、組織的な方法でレガシー移行を実行するための移行方法論と、一般的な移行シナリオを自動化および高速化する一連のツールが含まれています。

大規模な移行

アプリケーションポートフォリオの大部分を次々にクラウドに移行し、各ウェーブでより多くのアプリケーションを高速に移動させるプロセス。この段階では、以前の段階から学んだベストプラクティスと教訓を使用して、移行ファクトリー チーム、ツール、プロセスのうち、オートメーションとアジャイルデリバリーによってワークロードの移行を合理化します。これは、[AWS 移行戦略](#) の第 3 段階です。

移行ファクトリー

自動化された俊敏性のあるアプローチにより、ワークロードの移行を合理化する部門横断的なチーム。移行ファクトリーチームには、通常、運用、ビジネスアナリストおよび所有者、移行エンジニア、デベロッパー、およびスプリントで作業する DevOps プロフェッショナルが含まれます。エンタープライズアプリケーションポートフォリオの 20~50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、このコンテンツセットの[移行ファクトリーに関する解説](#)と [Cloud Migration Factory ガイド](#)を参照してください。

移行メタデータ

移行を完了するために必要なアプリケーションおよびサーバーに関する情報。移行パターンごとに、異なる一連の移行メタデータが必要です。移行メタデータの例としては、ターゲットサブネット、セキュリティグループ、AWS アカウントなどがあります。

移行パターン

移行戦略、移行先、および使用する移行アプリケーションまたはサービスを詳述する、反復可能な移行タスク。例: AWS Application Migration Service を使用して Amazon EC2 への移行をリホストします。

Migration Portfolio Assessment (MPA)

オンラインツール。これによって、AWS クラウドに移行するビジネスケースの検証に必要な情報を得られます。MPA は、詳細なポートフォリオ評価 (サーバーの適切なサイジング、価格設定、TCO 比較、移行コスト分析) および移行プラン (アプリケーションデータの分析とデータ収集、アプリケーションのグループ化、移行の優先順位付け、およびウェーブプランニング) を提供します。[MPA ツール](#) (ログインが必要) は、すべての AWS コンサルタントと APN パートナー コンサルタントが無料で利用できます。

移行準備状況評価 (MRA)

AWS CAF を使用して、組織のクラウド準備状況に関するインサイトを取得し、長所と短所を特定し、特定されたギャップを埋めるためのアクションプランを構築するプロセス。詳細については、[移行準備状況ガイド](#)を参照してください。MRA は、[AWS 移行戦略](#)の第一段階です。

移行戦略

ワークロードを AWS クラウドに移行するために使用するアプローチ。詳細については、この用語集の [7 Rs](#) エントリと、「[組織を動員して大規模な移行を加速する](#)」を参照してください。

ML

「[機械学習](#)」を参照してください。

モダナイゼーション

古い (レガシーまたはモノリシック) アプリケーションとそのインフラストラクチャをクラウド内の俊敏で弾力性のある高可用性システムに変換して、コストを削減し、効率を高め、イノベーションを活用します。詳細については、「[AWS クラウドでのアプリケーションのモダナイズ戦略](#)」を参照してください。

モダナイゼーション準備状況評価

組織のアプリケーションのモダナイゼーションの準備状況を判断し、利点、リスク、依存関係を特定し、組織がこれらのアプリケーションの将来の状態をどの程度適切にサポートできるかを決定するのに役立つ評価。評価の結果として、ターゲットアーキテクチャのブループリント、モダナイゼーションプロセスの開発段階とマイルストーンを詳述したロードマップ、特定されたギャップに対処するためのアクションプランが得られます。詳細については、「[AWS クラウドでのアプリケーションのモダナイゼーションの準備状況を評価する](#)」を参照してください。

モノリシックアプリケーション (モノリス)

緊密に結合されたプロセスを持つ単一のサービスとして実行されるアプリケーション。モノリシックアプリケーションにはいくつかの欠点があります。1つのアプリケーション機能エクスペリエンスの需要が急増する場合は、アーキテクチャ全体をスケーリングする必要があります。モノリシックアプリケーションの特徴を追加または改善することは、コードベースが大きくなると複雑になります。これらの問題に対処するには、マイクロサービスアーキテクチャを使用できます。詳細については、「[モノリスをマイクロサービスに分解する](#)」を参照してください。

MPA

「[Migration Portfolio Assessment](#)」を参照してください。

MQTT

「[Message Queuing Telemetry Transport](#)」を参照してください。

多クラス分類

複数のクラスの予測を生成するプロセス (2 つ以上の結果の 1 つを予測します)。例えば、機械学習モデルが、「この製品は書籍、自動車、電話のいずれですか?」または、「このお客様にとって最も関心のある商品のカテゴリはどれですか?」と聞くかもしれません。

ミュータブルなインフラストラクチャ

本番ワークロードに使用する既存のインフラストラクチャを更新および変更するためのモデル。Well-Architected AWS フレームワークでは、一貫性、信頼性、予測可能性を向上させるために、[イミュータブルインフラストラクチャ](#)の使用をベストプラクティスとして推奨しています。

O

OAC

「[オリジンアクセス制御](#)」を参照してください。

OAI

「[オリジンアクセスアイデンティティ](#)」を参照してください。

OCM

「[組織変更管理](#)」を参照してください。

オフライン移行

移行プロセス中にソースワークロードを停止させる移行方法。この方法はダウンタイムが長くなるため、通常は重要ではない小規模なワークロードに使用されます。

OI

「[オペレーション統合](#)」を参照してください。

Ola

「[オペレーショナルレベルアグリーメント](#)」を参照してください。

オンライン移行

ソースワークロードをオフラインにせずにターゲットシステムにコピーする移行方法。ワークロードに接続されているアプリケーションは、移行中も動作し続けることができます。この方法はダウンタイムがゼロから最小限で済むため、通常は重要な本番稼働環境のワークロードに使用されます。

OPC-UA

「[Open Process Communications - Unified Architecture](#)」を参照してください。

Open Process Communications - Unified Architecture (OPC-UA)

産業オートメーション用のマシンツーマシン (M2M) 通信プロトコル。OPC-UA により、相互運用の際に、データ暗号化、認証、認可の各スキームを標準化できます。

オペレーショナルレベルアグリーメント (OLA)

サービスレベルアグリーメント (SLA) をサポートするために、どの機能的 IT グループが互いに提供することを約束するかを明確にする契約。

運用準備状況レビュー (ORR)

質問と関連するベストプラクティスのチェックリスト。インシデントや起こり得る障害を理解、評価、防止したり、その範囲を縮小したりする際に役立ちます。詳細については、AWS Well-Architected フレームワークの「[Operational Readiness Reviews \(ORR\)](#)」を参照してください。

運用テクノロジー (OT)

産業オペレーション、機器、インフラストラクチャを制御するために物理環境と連携させるハードウェアおよびソフトウェアシステム。製造分野では、[Industry 4.0](#) への変革を進める上で、OT と情報技術 (IT) システムの統合に焦点が当てられています。

オペレーション統合 (OI)

クラウドでオペレーションをモダナイズするプロセスには、準備計画、オートメーション、統合が含まれます。詳細については、[オペレーション統合ガイド](#)を参照してください。

組織の証跡

組織 AWS アカウント 内のすべてのイベント AWS CloudTrail をログに記録することによって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウントに作成され、各アカウントのアクティビティを追跡します。詳細については、CloudTrail ドキュメントの「[組織の証跡の作成](#)」を参照してください。

組織変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCM は、変化の導入を加速し、移行問題に対処し、文化や組織の変化を推進することで、組織が新しいシステムと戦略の準備と移行するのを支援します。AWS 移行戦略では、クラウド導入プロジェクトに必要な変化のスピードにより、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、[OCM ガイド](#)を参照してください。

オリジンアクセス制御 (OAC)

Amazon Simple Storage Service (Amazon S3) コンテンツを保護するための、CloudFront のアクセス制限の強化オプション。OAC は AWS リージョン、すべての S3 バケット、AWS KMS (SSE-KMS) によるサーバー側の暗号化、S3 バケットへの動的 PUT および DELETE リクエストをサポートします。

オリジンアクセスアイデンティティ (OAI)

CloudFront の、Amazon S3 コンテンツを保護するためのアクセス制限オプション。OAI を使用すると、CloudFront が、Amazon S3 に認証可能なプリンシパルを作成します。認証されたプリンシパルは、S3 バケット内のコンテンツに、特定の CloudFront ディストリビューションを介してのみアクセスできます。[OAC](#) も併せて参照してください。OAC では、より詳細な、強化されたアクセス制御が可能です。

ORR

「[運用準備状況レビュー](#)」を参照してください。

OT

「[運用テクノロジー](#)」を参照してください。

アウトバウンド (送信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション内から開始されたネットワーク接続を処理する VPC。[AWS Security Reference Architecture](#) では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

P

アクセス許可の境界

ユーザーまたはロールが使用できるアクセス許可の上限を設定する、IAM プリンシパルにアタッチされる IAM 管理ポリシー。詳細については、IAM ドキュメントの[アクセス許可の境界](#)を参照してください。

個人を特定できる情報 (PII)

直接閲覧した場合、または他の関連データと組み合わせた場合に、個人の身元を合理的に推測するために使用できる情報。PII の例には、氏名、住所、連絡先情報などがあります。

PII

「[個人を特定できる情報](#)」を参照してください。

プレイブック

クラウドでのコアオペレーション機能の提供など、移行に関連する作業を取り込む、事前定義された一連のステップ。プレイブックは、スクリプト、自動ランブック、またはお客様のモダナイズされた環境を運用するために必要なプロセスや手順の要約などの形式をとることができます。

PLC

「[プログラマブルロジックコントローラー](#)」を参照してください。

PLM

「[製品ライフサイクル管理](#)」を参照してください。

ポリシー

次の操作を可能にするオブジェクト: アクセス許可を定義する ([ID ベースのポリシー](#)を参照)。アクセス条件を指定する ([リソースベースのポリシー](#)を参照)。AWS Organizations の組織における全アカウントにアクセス許可の上限を定義する ([サービスコントロールポリシー](#)を参照)。

多言語の永続性

データアクセスパターンやその他の要件に基づいて、マイクロサービスのデータストレージテクノロジーを個別に選択します。マイクロサービスが同じデータストレージテクノロジーを使用している場合、実装上の問題が発生したり、パフォーマンスが低下する可能性があります。マイクロサービスは、要件に最も適合したデータストアを使用すると、より簡単に実装でき、パフォーマンスとスケーラビリティが向上します。

ポートフォリオ評価

移行を計画するために、アプリケーションポートフォリオの検出、分析、優先順位付けを行うプロセス。詳細については、「[移行の準備状況の評価](#)」を参照してください。

述語

true または false を返すためのクエリ条件。一般的に、WHERE 句に記述されます。

述語プッシュダウン

データベースクエリを最適化する手法。これによって、転送前にクエリ内のデータをフィルタリングします。この手法を取ると、リレーショナルデータベースから取得し処理する必要のあるデータの量が減少するため、クエリのパフォーマンスが向上します。

予防的コントロール

イベントの発生を防ぐように設計されたセキュリティコントロール。このコントロールは、ネットワークへの不正アクセスや好ましくない変更を防ぐ最前線の防御です。詳細については、「AWSでのセキュリティコントロールの実装」の「[予防的コントロール](#)」を参照してください。

プリンシパル

アクションを実行し AWS、リソースにアクセスできるのエンティティ。このエンティティは通常、IAM AWS アカウントロール、またはユーザーのルートユーザーです。詳細については、IAM ドキュメントの「[ロールに関する用語と概念](#)」にあるプリンシパルを参照してください。

プライバシーバイデザイン

開発プロセス全体を通してプライバシーが考慮されているシステムエンジニアリングのアプローチ。

プライベートホストゾーン

1 つ以上の VPC 内のドメインとそのサブドメインへの DNS クエリに対し、Amazon Route 53 がどのように応答するかに関する情報を保持するコンテナ。詳細については、Route 53 ドキュメントの「[プライベートホストゾーンの使用](#)」を参照してください。

プロアクティブコントロール

非準拠リソースのデプロイ防止を目的とした[セキュリティコントロール](#)。このコントロールにより、プロビジョニング前にリソースをスキャンします。コントロールに準拠していないリソースは、プロビジョニングされません。詳細については、AWS Control Tower ドキュメントの「[コントロールリファレンスガイド](#)」および「[セキュリティコントロールの実装](#)」の「[プロアクティブコントロール](#)」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

製品の設計、開発、発売から、成長、成熟、衰退、廃棄に至る、製品のライフサイクル全体を通してデータとプロセスを管理すること。

本番環境

「[環境](#)」を参照してください。

プログラマブルロジックコントローラー (PLC)

製造分野で使用される、信頼性と適応性に優れたコンピュータであり、これによって、マシンをモニタリングするとともに、製造プロセスを自動化します。

プロンプトチェイニング

1 つの [LLM](#) プロンプトによる出力を次のプロンプトの入力に使用して、より良いレスポンスを生成します。この手法を使用すると、複雑なタスクをサブタスクに分割したり、事前レスポンスを繰り返し改良または拡張したりできます。これによって、モデルのレスポンスの精度と関連性が向上し、粒度の高いパーソナライズされた結果を得られます。

仮名化

データセット内の個人識別子をプレースホルダー値に置き換えるプロセス。仮名化は個人のプライバシー保護に役立ちます。仮名化されたデータは、依然として個人データとみなされます。

発行/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。これにより、スケーラビリティと応答性を向上させます。例えば、マイクロサービスベースの [MES](#) の場合、マイクロサービスは、他のマイクロサービスがサブスクライブ可能なチャンネルにイベントメッセージを発行できます。このシステムでは、発行サービスの変更なしに、新規マイクロサービスを追加できます。

Q

クエリプラン

手順などの一連のステップであり、SQL リレーショナルデータベースシステムのデータにアクセスするために使用されます。

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

「[実行責任者、説明責任者、協業先、報告先 \(RACI\)](#)」を参照してください。

RAG

「[検索拡張生成](#)」を参照してください。

ランサムウェア

決済が完了するまでコンピュータシステムまたはデータへのアクセスをブロックするように設計された、悪意のあるソフトウェア。

RASCI マトリックス

「[実行責任者、説明責任者、協業先、報告先 \(RACI\)](#)」を参照してください。

RCAC

「[行と列のアクセス制御](#)」を参照してください。

リードレプリカ

読み取り専用で使用されるデータベースのコピー。クエリをリードレプリカにルーティングして、プライマリデータベースへの負荷を軽減できます。

リアーキテクト

「[7 Rs](#)」を参照してください。

目標復旧時点 (RPO)

最後のデータリカバリポイントからの最大許容時間です。これにより、最後の回復時点からサービスが中断されるまでの間に許容できるデータ損失の程度が決まります。

目標復旧時間 (RTO)

サービスが中断から復旧までの最大許容遅延時間。

リファクタリング

「[7 Rs](#)」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョンは、耐障害性、安定性、耐障害性を提供するために、他のリージョンから分離され、独立しています。詳細については、「[アカウントが使用できる AWS リージョンを指定する](#)」を参照してください。

リグレッション

数値を予測する機械学習手法。例えば、「この家はどれくらいの値段で売れるでしょうか?」という問題を解決するために、機械学習モデルは、線形回帰モデルを使用して、この家に関する既知の事実 (平方フィートなど) に基づいて家の販売価格を予測できます。

リホスト

「[7 Rs](#)」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「[7 Rs](#)」を参照してください。

リプラットフォーム

「[7 Rs](#)」を参照してください。

再購入

「[7 Rs](#)」を参照してください。

回復性

中断に抵抗または中断から回復するアプリケーションの機能。AWS クラウドでの回復力を計画する際には、一般的に、[高可用性](#)と[ディザスタリカバリ](#)が考慮されます。詳細については、「[AWS クラウドの耐障害性](#)」を参照してください。

リソースベースのポリシー

Amazon S3 バケット、エンドポイント、暗号化キーなどのリソースにアタッチされたポリシー。このタイプのポリシーは、アクセスが許可されているプリンシパル、サポートされているアクション、その他の満たすべき条件を指定します。

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R)、説明責任 (A)、協議 (C)、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートが含まれる場合は RASCI マトリックスと呼ばれ、含まれない場合は RACI マトリックスと呼ばれます。

レスポンスコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、「AWSでのセキュリティコントロールの実装」の「[レスポンスコントロール](#)」を参照してください。

保持

「[7 Rs](#)」を参照してください。

廃止

「[7 Rs](#)」を参照してください。

検索拡張生成 (RAG)

[生成 AI](#) の技術。これにより、[LLM](#) では、レスポンスの生成前に、トレーニングデータソースの外部にある信頼できるデータソースが参照されます。例えば、RAG モデルによって、組織のナレッジベースまたはカスタムデータのセマンティック検索を実行できる場合があります。細については、「[RAG \(検索拡張生成\) とは何ですか?](#)」を参照してください。

ローテーション

定期的に[シークレット情報](#)を更新して、攻撃者が認証情報にアクセスするのをより困難にするプロセス。

行と列のアクセス制御 (RCAC)

アクセスルールが定義された、基本的で柔軟な SQL 表現の使用。RCAC は行権限と列マスクで構成されています。

RPO

「[目標復旧時点](#)」を参照してください。

RTO

「[目標復旧時間](#)」を参照してください。

ランブック

特定のタスクを実行するために必要な手動または自動化された一連の手順。これらは通常、エラー率の高い反復操作や手順を合理化するために構築されています。

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーテッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーを IAM で作成しなくても、AWS マネジメントコンソールにログインしたり AWS、API オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントの「[SAML 2.0 ベースのフェデレーションについて](#)」を参照してください。

SCADA

「[監視制御とデータ取得](#)」を参照してください。

SCP

「[サービスコントロールポリシー](#)」を参照してください。

シークレット

暗号化された形式で保存する AWS Secrets Manager パスワードやユーザー認証情報などの機密情報または制限付き情報。シークレット値とそのメタデータで構成されます。シークレット値には、バイナリ、1 つの文字列、複数の文字列を指定できます。詳細については、Secrets Manager ドキュメントの「[Secrets Manager シークレットの概要](#)」を参照してください。

セキュリティバイデザイン

開発プロセス全体を通してセキュリティが考慮されているシステムエンジニアリングのアプローチ。

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、主に 4 つの種類があります。4 つとは、[予防](#)、[検出](#)、[レスポンス](#)、[プロアクティブ](#)です。

セキュリティ強化

アタックサーフェスを狭めて攻撃への耐性を高めるプロセス。このプロセスには、不要になったリソースの削除、最小特権を付与するセキュリティのベストプラクティスの実装、設定ファイル内の不要な機能の無効化、といったアクションが含まれています。

Security Information and Event Management (SIEM) システム

セキュリティ情報管理 (SIM) とセキュリティイベント管理 (SEM) のシステムを組み合わせたツールとサービス。SIEM システムは、サーバー、ネットワーク、デバイス、その他ソースからデータを収集、モニタリング、分析して、脅威やセキュリティ違反を検出し、アラートを発信します。

セキュリティレスポンスの自動化

セキュリティイベントへの自動レスポンスまたは自動修復を目的として、事前定義およびプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ[検出的](#)または[応答的](#)な AWS セキュリティコントロールとして機能します。自動レスポンスアクションの例には、VPC セキュリティグループの変更、Amazon EC2 インスタンスへのパッチ適用、認証情報の更新などがあります。

サーバー側の暗号化

送信先で、それ AWS のサービスを受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCP は、管理者がユーザーまたはロールに委任するアクションに、ガードレールを定義したり、アクションの制限を設定したりします。SCP は、許可リストまたは拒否リストとして、許可または禁止するサービスやアクションを指定する際に使用できます。詳細については、AWS Organizations ドキュメントの「[サービスコントロールポリシー](#)」を参照してください。

サービスエンドポイント

のエンドポイントの URL AWS のサービス。ターゲットサービスにプログラムで接続するには、エンドポイントを使用します。詳細については、「AWS 全般のリファレンス」の「[AWS のサービス エンドポイント](#)」を参照してください。

サービスレベルアグリーメント (SLA)

サービスのアップタイムやパフォーマンスなど、IT チームがお客様に提供すると約束したものを明示した合意書。

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

[サービスレベルインジケータ](#)によって測定され、サービスの状態を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、 はクラウドのセキュリティを担当します。詳細については、「[責任共有モデル](#)」を参照してください。

SIEM

「[Security Information and Event Management システム](#)」を参照してください。

単一障害点 (SPOF)

特定のアプリケーションを構成する単一の重要なコンポーネントで発生し、システム稼働に支障をきたす可能性のある障害。

SLA

「[サービスレベルアグリーメント](#)」を参照してください。

SLI

「[サービスレベルインジケータ](#)」を参照してください。

SLO

「[サービスレベルの目標](#)」を参照してください。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケーリングと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、「[AWS クラウドでのアプリケーションをモダナイズするための段階的アプローチ](#)」を参照してください。

SPOF

「[単一障害点](#)」を参照してください。

スタースキーマ

データベースの編成構造を意味し、1つの大きいファクトテーブルにトランザクションデータまたは測定データが保存され、1つ以上の小さいディメンションテーブルにデータ属性が保存されます。この構造は、[データウェアハウス](#)やビジネスインテリジェンスを用途とするように設計されています。

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主に取って代わるイチジクのつるを例えています。そのパターンは、モノリシックシステムを書き換えるときのリスクを管理する方法として [Martin Fowler により提唱されました](#)。このパターンの適用方法の例については、「[コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET \(ASMX\) ウェブサービスを段階的にモダナイズ](#)」を参照してください。

サブネット

VPC 内の IP アドレスの範囲。サブネットは、1つのアベイラビリティゾーンに存在する必要があります。

監視制御とデータ取得 (SCADA)

製造分野において、ハードウェアとソフトウェアを使用して物理アセットと本番運用をモニタリングするシステム。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

ユーザーとのやり取りをシミュレートして、起こり得る問題を検出したり、パフォーマンスをモニタリングしたりすることで、システムをテストします。[Amazon CloudWatch Synthetics](#) を使用すると、こうしたテストを作成できます。

システムプロンプト

コンテキスト、指示、ガイドラインなどを提示して、[LLM](#) に動作を指示する手法。システムプロンプトは、コンテキストを設定して、ユーザーとやり取りするルールを確立するのに有用です。

T

タグ

AWS リソースを整理するためのメタデータとして機能するキーと値のペア。タグは、リソースの管理、識別、整理、検索、フィルタリングに役立ちます。詳細については、「[AWS リソースのタグ付け](#)」を参照してください。

ターゲット変数

監督された機械学習でお客様が予測しようとしている値。これは、結果変数のことも指します。例えば、製造設定では、ターゲット変数が製品の欠陥である可能性があります。

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要がある一般的なタスクのリストが含まれています。各一般的なタスクには、推定所要時間、所有者、進捗状況が含まれています。

テスト環境

「[環境](#)」を参照してください。

トレーニング

お客様の機械学習モデルに学習するデータを提供すること。トレーニングデータには正しい答えが含まれている必要があります。学習アルゴリズムは入力データ属性をターゲット (お客様が予測したい答え) にマッピングするトレーニングデータのパターンを検出します。これらのパターンをキャプチャする機械学習モデルを出力します。そして、お客様が機械学習モデルを使用して、ターゲットがわからない新しいデータでターゲットを予測できます。

トランジットゲートウェイ

VPC とオンプレミスネットワークを相互接続するために使用できる、ネットワークの中継ハブ。詳細については、AWS Transit Gateway ドキュメントの「[トランジットゲートウェイとは](#)」を参照してください。

トランクベースのワークフロー

デベロッパーが機能ブランチで機能をローカルにビルドしてテストし、その変更をメインブランチにマージするアプローチ。メインブランチはその後、開発環境、本番前環境、本番環境に合わせて順次構築されます。

信頼されたアクセス

ユーザーに代わって AWS Organizations およびそのアカウントで組織内でタスクを実行するために指定したサービスにアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要とときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、ドキュメントの「[Using AWS Organizations with other AWS services](#) AWS Organizations」を参照してください。

チューニング

機械学習モデルの精度を向上させるために、お客様のトレーニングプロセスの側面を変更する。例えば、お客様が機械学習モデルをトレーニングするには、ラベル付けセットを生成し、ラベルを追加します。これらのステップを、異なる設定で複数回繰り返して、モデルを最適化します。

ツーピザチーム

2 枚のピザを分け合えることができるくらい小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

予測機械学習モデルの信頼性を損なう可能性がある、不正確、不完全、または未知の情報を指す概念。不確実性には、次の 2 つのタイプがあります。認識論的不確実性は、限られた、不完全なデータによって引き起こされ、弁論的不確実性は、データに固有のノイズとランダム性によって引き起こされます。

未分化なタスク

ヘビーリフティングとも呼ばれ、アプリケーションの作成と運用には必要だが、エンドユーザーに直接的な価値をもたらさなかったり、競争上の優位性をもたらしたりしない作業です。未分化なタスクの例としては、調達、メンテナンス、キャパシティプランニングなどがあります。

上位環境

「[環境](#)」を参照してください。

V

バキューミング

ストレージを再利用してパフォーマンスを向上させるために、増分更新後にクリーンアップを行うデータベースのメンテナンス操作。

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

プライベート IP アドレスを使用してトラフィックをルーティングできる、2 つの VPC 間の接続。詳細については、Amazon VPC ドキュメントの「[VPC ピア機能とは](#)」を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

頻繁にアクセスされる最新の関連データを含むバッファキャッシュ。データベースインスタンスはバッファキャッシュから、メインメモリまたはディスクからよりも短い時間で読み取りを行うことができます。

ウォームデータ

アクセス頻度の低いデータ。この種類のデータをクエリする場合、通常は適度に遅いクエリでも問題ありません。

ウィンドウ関数

現在のレコードに何らかの形で関連している行のグループに計算を実行する SQL 関数。ウィンドウ関数は、移動平均を計算したり、現在の行の相対位置に基づいて他の行の値にアクセスするといったタスクの処理に役立ちます。

ワークロード

ビジネス価値をもたらすリソースとコード (顧客向けアプリケーションやバックエンドプロセスなど) の総称。

ワークストリーム

特定のタスクセットを担当する移行プロジェクト内の機能グループ。各ワークストリームは独立していますが、プロジェクト内の他のワークストリームをサポートしています。たとえば、ポートフォリオワークストリームは、アプリケーションの優先順位付け、ウェーブ計画、および移行メタデータの収集を担当します。ポートフォリオワークストリームは、これらの設備を移行ワークストリームで実現し、サーバーとアプリケーションを移行します。

WORM

「[Write-Once-Read-Many](#)」を参照してください。

WQF

「[AWS ワークロード資格フレームワーク](#)」を参照してください。

Write-Once-Read-Many (WORM)

データを 1 回のみ書き込むことで、データの削除や変更を防ぐストレージモデル。承認済みユーザーは、必要な回数だけデータを読み取ることができますが、変更することはできません。このデータストレージインフラストラクチャは、[イミュータブル](#)と見なされます。

Z

ゼロデイエクスプロイト

[ゼロデイ脆弱性](#)を悪用した攻撃（一般的にマルウェアによる）。

ゼロデイ脆弱性

実稼働システムにおける未解決の欠陥または脆弱性。脅威アクターは、このような脆弱性を利用してシステムを攻撃する可能性があります。開発者は、よく攻撃の結果で脆弱性に気付きます。

ゼロショットプロンプト

[LLM](#) にタスク実行の手順は提示するが、実行のガイドとして役立つ例（ショット）は提示しない方法。LLM は、事前トレーニング済みの知識を使用してタスクを処理する必要があります。ゼロショットプロンプトの有効性は、タスクの複雑さとプロンプトの品質によって異なります。「[数ショットプロンプト](#)」も参照してください。

ゾンビアプリケーション

平均 CPU およびメモリ使用率が 5% 未満のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するのが一般的です。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。