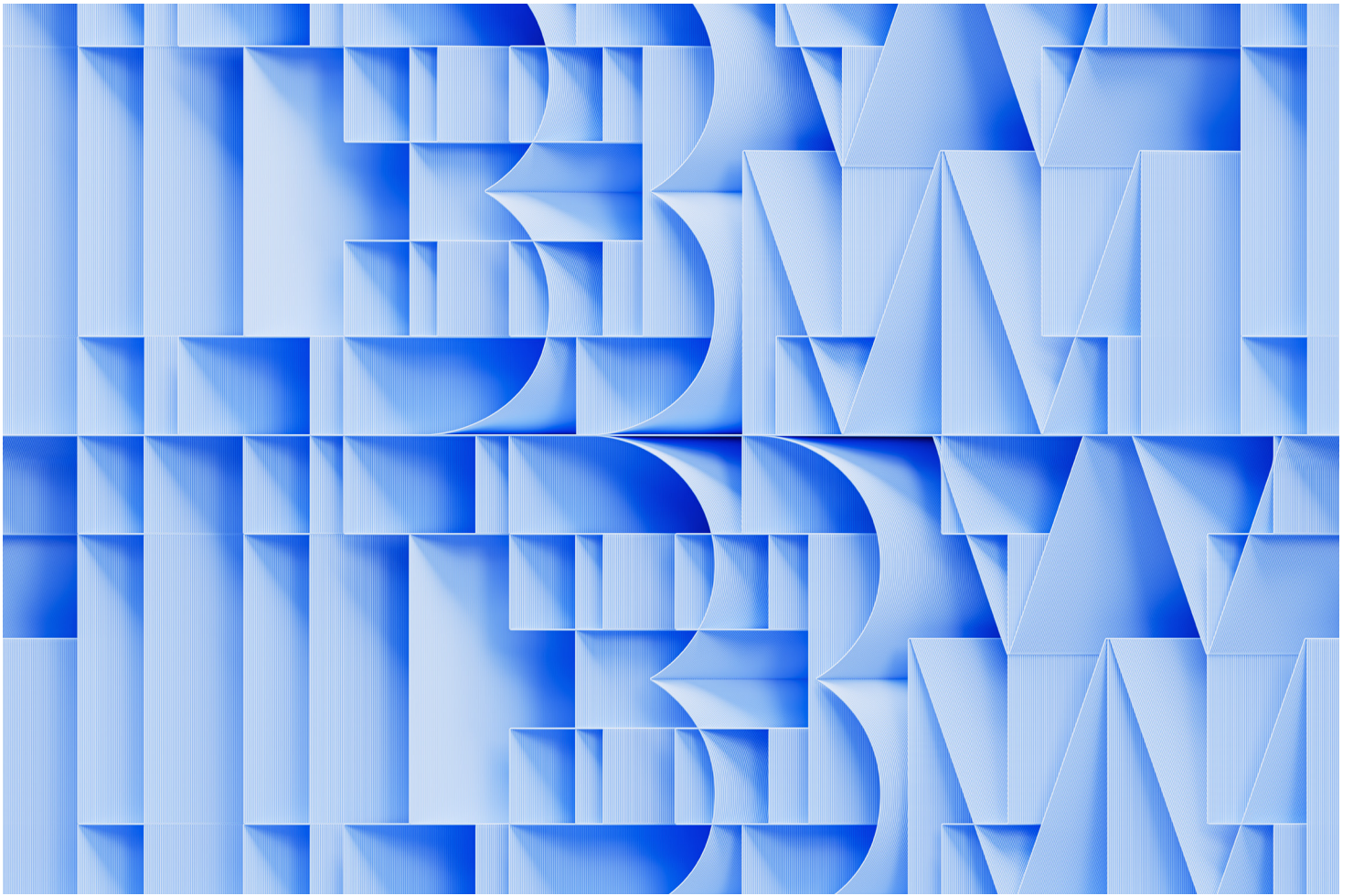




ソフトウェア開発 を変革したAI。 次は、デリバリー を変革します。

開発のあり方がAIによって大きく変わりつつあるなか、
企業はガバナンスとスケーラビリティを両立した
ソフトウェア・デリバリーを実現すべく、
統合システムを導入する必要があります。
その変革を実現する方法をご紹介します。

目次

01

ツールからデリバリー・
システムへ移行

03

エンタープライズ・デリバリー
方法の決定：構築か、導入か、
オーケストレーションか

02

デリバリー方法

04

エンタープライズ対応の
SDLCパートナーを採用

01 ツールから デリバリー・ システムへ移行



今日、エンタープライズ・ソフトウェアを提供する組織では、開発のアウトプットが大きく変化しています。開発者だけでなく、活用が進むAIエージェントも高いパフォーマンスを持続的に発揮できるようになっています。コード生成は高速化し、リファクタリングにかかる時間も数分（場合によっては数秒）に短縮しています。また、テスト・スイートも自動的に生成されるようになりました。かつてはアウトプットの妨げとなっていたローカルな制約も減りつつあります。

IBMでは、チームやエージェントがこれまでより長時間、集中してオペレーションを行えるようにしました。ここで重要なのは、こうしたキャパシティの強化が、システム・レベルでより質の高い成果につながっているのかどうかです。たとえば、次のような問いは重要です。

- － 意義のあるデリバリーを増やすことができているか、単にパフォーマンスを高めているだけなのか。
- － エンタープライズ・システム全体を強化しつつあるのか、それとも局所的な業務を迅速化しているだけなのか。
- － パフォーマンスは誰のために最適化されているのか。

今日の企業は、個人の生産性がこれまで以上に高まる一方で、組織内の一貫性がますます重要になりつつある、というパラドックスに直面しています。

一貫性のない迅速化

今日の企業が感じている重荷は、イノベーションへの抵抗のせいではなく、組織内部の整合性が取れていないことに起因します。AIコーディング・ツールにより、局所レベルでは先発優位のメリットが得られました。ただし、ツールはシステムではなく、あくまでも特定の領域を強化するものです。一方、システムは方向性と目的を統制します。

導入規模の拡大とともに、次のようなポイントを考慮する必要があります：

- － 生産性を持つ意味は、個人、チーム、企業レベルでそれぞれ異なる
- － 開発者アイデンティティーは、組織内でさまざま
- － SDLCモデルの統一のレベルは均一ではない
- － ガバナンス、セキュリティ、コンプライアンス、統合に関する要件が後回しにされ、下流での対応になっている
- － 企業特有の事情に基づいたオーケストレーションを行わずにエージェントをデプロイしている
- － 統合メカニズムが成熟するよりも速く、オートメーションが拡大している

このため、局所的な迅速化は進んでいるのに、企業全体としての成果につながっていないという問題が生まれます。

開発者の生産性向上に取り組む前に、開発者のアイデンティティーを考える

ソフトウェア・デリバリーの最適化に取り組む前に、社内のさまざまな開発者の役割を把握する必要があります。チームごとに取り組んでいる問題は異なります。問題が同じであると仮定して評価を行ったり、支援を行ったりすべきではありません。

大きな組織では、構造的に次のような違いが生まれます：

- － プラットフォーム・チームとプロダクト・チームの最適化が異なる
- － 厳しい規制を受ける環境と、社内のツール・グループの最適化が異なる
- － オートメーションの許容範囲はさまざま
- － ガバナンス要件はさまざま
- － 速度（Velocity）を測る指標はさまざま
- － リスクに対する体制はさまざま

こうした違いは、業務の進め方や評価方法に大きく影響します。開発者アイデンティティー、メトリクス、SDLC体制が一貫性を欠いていれば、エージェント機能の増加によって、一貫性が高まるどころか、さらに分断化が進む可能性があります。

長期的なメリット：ツールからシステムへ

企業における議論は変化しつつあります。もはやその焦点は、開発者に高性能なAIツールを提供する方法ではなく、持続可能なスケーリングのためには、計画的にチーム、ワークフロー、内部統制の調整を図ることが欠かせません。SDLC全体にAIをスケーリングするにはモデルの機能だけでなく、次を備えた、デリバリーのオーケストレーションが不可欠です。

- － 企業のポリシーに沿った開発システム
- － ガバナンスを踏まえたオートメーションを上流に組み込むこと
- － システム全体で人間とエージェント双方のアクションを把握するオブザーバビリティ
- － 認知負荷の分散および意思決定支援レイヤー
- － 設計段階からの部門横断的な統合
- － デプロイメント経路に組み込まれたレジリエンス
- － コストおよびトレードオフの可視化

エンドツーエンドのデリバリー・システムを運用化すると、ツール管理にかかる時間を短縮して、成果創出に多くの時間を使えるようになります。

企業向けAIが直面する真の課題は、能力ではなく連携です。

企画から本番、ガバナンスから監査、コードから結果まで、デリバリー・システム全体にAIをどう組み込むべきでしょうか。

チームですでに知
られている ↓

チーム間ではほとんど
知られていない ↓

開発中に浮上
→

開発中に浮上することはまれ
→



企業がソフトウェアを提供するには、さまざまなレベルの知識と可視性が必要です。開発者を中心とするエージェント・ハーネス（グレーの四角形）は、タスクが明確に定義され、コードのコンテキストが明示されている開発ワークフロー内で主に機能します。企業のデリバリー・システムとSDLCパートナー（青およびグレーの四角形）は、運用シグナル、ガバナンス上の制約、複雑なアーキテクチャまでを含むより広範な環境で運用しなければなりません。

SDLCパートナーの登場

心強いことに、優れたプロバイダー各社は、エンタープライズ・レベルのソリューションから、より企業に最適化されたシステムに移行することの重要性を認識しています。ツールからプラットフォームへの進化は、健全かつ必要な流れです。

IBMでは、企業のデリバリー・システムにこうしたシステムをどう包括的に統合するかについて、今後こうした議論をさらに深める必要があると考えています。

また、デリバリーの未来は当社がSDLCパートナーと呼んでいる、次のようなシステムの出現によって決まると考えています：

- 開発者のアイデンティティを把握する
- さまざまなレベルの生産性メトリクスの整合を図る
- ガバナンスや下流への影響を設計に組み込む
- エージェントを野放しにせず、管理できるようオーケストレーションする
- より広範な企業基盤に接続する
- 組織の成長に合わせて進化しながら、一貫性も確保する

ツールは業務を効率化できますが、パートナーはこうした業務とより大きな企業目標の整合性を高めます。今後、単に変化スピードの速さだけでなく、その一貫性も企業にとって重要になってきます。

02 デリバリー 方法



現在、IBMでは生成、リファクタリング、テスト、デプロイを持続的に高速で行えるようになっていきます。多くの企業にとって、技術的な能力はもはや足かせになりません。人間と機械のコラボにより、さまざまな可能性が広がりました。戦略的な焦点は、もはや前進が続けられるかどうかではなく、そうした前進をどのように組み立て、統制し、企業の目的にあったものにするかという点にあります。

組織が個別ツールから統合デリバリー・システムに移行するなかで、複数のテーマが繰り返し浮上してきます。特に注目すべき点は次の4つです。



チームのイネーブルメント：個人の生産性向上だけでは企業価値につながるとは限らない。チームが協力して業務を遂行することが企業価値を左右する



セキュリティのシフトレフト：監視型のセキュリティから、デリバリーにリアルタイム制御を組み込むセキュリティへの進化が必要に



大規模なモダナイゼーション：レガシー・システムの複雑さが依然として解消されておらず、複数の環境に対応できるシステムが必要に



コスト管理：AIにより、コスト変動モデルが登場したことで、積極的な管理とシステム単位の最適化が必要に

チームの強化

AIを活用した開発は、従来の各開発者のコーディング・スピードを高速化するだけだったものから、複数のアプリケーション・チーム、プラットフォーム・エンジニアリング、サイト信頼性エンジニアリング、セキュリティ、モダナイゼーションまでを包括するワークフローの一貫性をエンドツーエンドで高める方向に移行しつつあります。コンテキストアウェアなコラボレーション・レイヤーを通じて、人材とエージェントをSDLC全体にわたってオーケストレーションするシステムを組織に導入する必要があります。



アイデンティティ認識型ワークフロー

ワークフローは各開発者の属性や役割に応じて変える必要があります。そうすることで、機能、ガードレール、オートメーションも、各開発者の担当業務に合ったものに調整できます。標準とドキュメントを組み込むことで、若手エンジニア、新入社員、部門横断的な関係者の生産性を高めることができます。結果として、既存のデリバリー方法を維持しつつ、役割に合った実行体制、オンボーディング時間の短縮、開発者満足度の向上、実行可能な知識移転、一貫した生産性の向上につながります。

エージェントのワークフローを人間が管理

エージェントが担うタスクが複雑化するなか、開発者は、体系化されたワークフローを通じて戦略的監視を継続的に行う必要があります。このワークフローでは、エージェントが変更を提案した際に、開発者がその意図と影響を確認し、オーケストレーション機能でガバナンスを担保します。計画のプレビュー、差分、体系化された承認フローにより、不必要な負担を増やすことなく、オートメーションと監視をバランスよく活用できます。

システム横断型のコンテキスト・アウェアネス

同時に、開発者が複数のシステムを横断して依存関係と影響を把握するためには、システムがリポジトリ、サービス、API、インフラストラクチャーを横断的に推論して、コードのコンテキストと運用テレメトリーを統合する必要があります。AIシステムでコードベース、環境、運用テレメトリーを横断してインサイトを統合することで、広範なシステムの挙動に変更がどう影響するかを理解しやすくなります。

開発者の知識と開発フロー

エンタープライズ環境の多くは、組織のレガシー・アプリケーションやドキュメンテーション、開発者の専門知識として、数十年にわたって蓄積された知識に依存しています。チーム全体がこうした知識を活用できるようにするには、AIシステムでこうした知識を抽出、文書化、運用化する必要があります。AIでこうしたコンテキスト情報を直接開発環境に取り込んで、認知負荷を軽減し、複数ウィンドウを切り替える手間を省くことで、スムーズに作業を行い、迅速に意思決定を下せます。

シフトレフト・セキュリティ

従来、セキュリティ検証はデリバリー・ライフサイクルの後半に行われていたため、ポリシー違反、インフラストラクチャーのリスク、コンプライアンス要件への対応に時間がかかることで、リリースの遅れの原因にもなっていました。

AIを活用した開発では、セキュリティをアーキテクチャーのレベルで事前対応する形で組み込む必要があります。最新のDevSecOpsは、静的分析、動的検証、Policy as Code (PaC) により、より早い段階でセキュリティに対応するよう変わっています。AI開発システムは、開発ワークフローにセキュリティ・アウェアネスを直接埋め込むことで、こうした変化をさらに拡張します。



マージ前のセキュリティとPolicy as Codeによるガバナンス

AI開発プラットフォームは、オーサリング中およびプル・リクエスト時に生成された変更を評価し、コードが共有リポジトリにマージされる前に、アーキテクチャーのガードレールとセキュリティ・ルールを適用し、コンプライアンス要件に対応する必要があります。こうしたチェックを開発ライフサイクルの早い段階に組み込むことで、安全性の低いパターンが開発パイプラインに広がるリスクを、開発を遅らせることなく軽減できます。

同時に、Policy as Codeのフレームワークは、さまざまなチームやパイプラインが行うコード、インフラストラクチャー、アーキテクチャーの意思決定に対して、バージョン管理された機械可読なガバナンス・ルールを適用する必要があります。セキュリティ・ポリシーをさまざまなチームやパイプラインに適用できるようにバージョン管理を行うことで、運用付加を増やさずに、一貫したガバナンスを実現できます。

コードとインフラストラクチャーを対象とした継続的なセキュリティ検証IDEおよびCI/CD内で静的分析と動的分析を併用する継続的な検証により、脆弱性と設定ミスが発生時点で検知して、セキュリティを定期的な監査から、ソフトウェア・デリバリーの本質的属性にする必要があります。また、安全対策として、AIシステムが生成したアーティファクトと、取り込むコンテキスト情報の両方に含まれるシークレット、APIキー、機密データを検出する必要があります。

こうした方法で、脆弱性や設定ミスをデプロイメント後ではなく開発中に検出することで、デリバリーのスピードを維持しつつ、セキュリティ体制を強化できます。

境界の設定とエンドツーエンドのトレーサビリティ

AIアウェアな脅威保護により、プロンプト・インジェクション攻撃、悪意のあるコンテキスト操作、危険な生成パターンのリスクを軽減できます。また、ロールベースアクセス制御により、エージェントの活動範囲を企業のアイデンティティと権限の範囲内に制限できます。さらに、適切なインシデント調査、規制報告、説明責任を実現するためには、包括的な監査証跡でプロンプト、アーティファクト、承認、デプロイメントを記録し、人間のアクションと自動アクションの両方を対象に高いトレーサビリティを維持する必要があります。こうした機能を組み合わせることで、マージ前の違反の特定および対処、一貫したガバナンスの適用、エクスポージャー・リスクの軽減が可能です。加えて、コンプライアンスに準拠した透明性の高いデリバリーを大規模に維持できます。



大規模なモダナイゼーション

AIによってモダナイゼーションを劇的に加速できる可能性があります。既存の挙動を維持し、結果を検証できる体系化されたデリバリー・システムでITを運用する場合に限られます。



レガシー環境全体を縦断する依存関係マッピング

モダナイゼーションを効果的に行うには、本番運用の妨げとなることなく、システムの現在の挙動、コンポーネント間の依存関係、変更の発生状況を可視化する必要があります。企業のAI開発システムは、リポジトリ、サービス、API、データフロー間の依存関係をマッピングして、複数のファイルやリポジトリを横断して推論を行い、作業を始める前に相互作用のパターンと潜在的なリスクを明らかにする必要があります。組織としてのロジックを守るためにはテスト、挙動の比較、検証アーティファクトを生成することで、AIが提案した変更を人が確認し、リファクタリングや移行の過程を通じて、同じ機能が保たれていることを確認します。

リスクスコアに基づくモダナイゼーション・シーケンス

モダナイゼーションを安全に行うためには、複数の手順が必要です。企業向けのAIシステムは、構造的な改善と検証を段階的に行うために、複数フェーズで構成されたリファクタリングに対応している必要があります。

リスクスコアに基づくモダナイゼーション・シーケンスで、先に低リスクのコンポーネントに着手することで、より複雑な領域に取り組む前に確実性を高めておく必要があります。同時に、AIツールでレガシー・モジュールの要約、引き継がれてきたロジックの説明、トランスフォーメーションの意図の文書化、意思決定の記録を通じて、アーキテクチャーに関する知識を取り込む必要があります。

システム全体への影響の把握 サービス契約と下流の依存関係を分析することで、システム全体への影響を事前に評価するとともに、コンプライアンス・ワークフローと監査証跡で、変更内容、変更理由、検証方法の記録を検証可能な状態で保持する必要があります。モダナイゼーションを持続的に行うには、厳格に管理されたフレームワーク内でAIを使用して、システムの依存関係を明らかにし、組織のロジックを守り、変更を安全なシーケンスで進めて、あらゆるトランスフォーメーションを検証および理解する必要があります。

コスト管理

ソフトウェア開発ライフサイクル全体にAIを組み込む企業が増えるなか、開発の迅速化と同様に、デリバリーの経済性の管理が重要になっています。組織の課題は、モデル自体のコストだけではなく、コード生成、テスト、インフラストラクチャーの移行、継続的な運用ワークフローなど、ソフトウェア開発に関わる幅広いコスト要因を管理することとなっています。

企業向けAI開発システムは、リソース消費の可視化、適切なモデルやインフラストラクチャーに合わせたワークロードの調整、実行前にコスト・パフォーマンスのトレードオフを明らかにすることで、経済面を制御するレイヤーとしての役割を果たさねばなりません。

開発ワークフローにコスト・アウェアネスを組み込むことで、AIを予測不可能なコスト要因から、経済効率を高めるツールに変えることができます。



インテリジェントなモデル・ルーティング コストを管理するには、チームやパイプラインを越えてインテリジェントなモデル・ルーティング、予算管理、ワークフローごとの効率性追跡、リソース管理の詳細なオブザーバビリティが欠かせません。日常業務が自動的に軽量モデルにルーティングされる一方で、アーキテクチャーに関する推論やマルチファイル分析などの複雑な推論には、より高機能のモデルが活用されることがあります。これにより、開発ワークフローを妨げることなく、性能とコストを継続的に最適化できます。

予算しきい値の適用

予算にしきい値を設定し、ポリシー制御を行うことで予測可能な経済的境界を実現できる一方で、ライフサイクルを重視した効率性メトリクスで、トークン使用量を超えたAIの真のインパクトを評価できます。詳細なコスト配分とリアルタイムでの経済的フィードバック・ループにより、大規模なリファクタリング、マルチエージェント・タスク、インフラストラクチャー分析を実行する前にチームは、コストへの影響をプロアクティブに把握できます。これにより、コストが制御不能になるリスクを避け、安全にスケーリングできます。

効率性とコストの最適化

AIツールの成熟とともに、コード生成にとどまらず、開発者が広範な運用ワークフローに活用するソリューションの評価が必要になってきます。開発環境とインフラストラクチャー環境全体の運用上の無駄を削減するには、システム・テレメトリーと開発シグナルを分析できるシステムを優先する必要があります。

チームがデリバリーの信頼性と効率性を守りつつ、複雑化するシステムを管理するなか、開発ワークフローと運用に関するインサイトを連携できる機能が、今後ますます重要になります。また、こうしたインサイトにより、手戻りや不具合の削減、テスト生成の高速化、マージにかかる時間の短縮、デプロイメントの安定性向上などの、デリバリー・ライフサイクル全体に及ぶ改善を追跡することも求められます。

2025年の導入でわかったこと

エンタープライズ・システムが次に提供すべき機能

AIツールにより、個々の開発者の生産性が劇的に向上

組織全体の生産性向上が新たな最適化目標に

人間とAIのコラボにより、局所的な開発ワークフローが改善

チームとAIのコラボにより、管理されたデリバリー・システムの実現へ

コード生成により機能実装が高速化

エンドツーエンドのデリバリー・オーケストレーションにより、本番環境への展開から保守までのプロセスを一元管理

セキュリティ・レビュー・プロセスがAIの出力スピードに対応する必要

SDLC全体にコンプライアンスを組み込むことで、事前対応型のAI駆動型セキュリティを実現

オートメーション機能に遅れを取るガバナンス・モデル

ガバナンスはオートメーションの妨げではなく、開発チームによって積極的に活用される機能

AI活用により、コードのモダナイゼーションが新たにはI活用により、コードのモダナイゼーションが再加速

デリバリー・システムに継続的なモダナイゼーションを組み込むことで、エンドツーエンドで支援を強化

AIを活用したコード解析により、組織に蓄積された知識を可視化

アーキテクチャーに関するコンテキストを企業全体で共有

AIの有料利用が、開発チームにとっての新たな経済的考慮事項に

デリバリーの経済的ガバナンスで高速化コストを管理

開発ツールとインフラストラクチャーや運用システムとの連携が開始

運用とインフラストラクチャーに関するコンテキストを取り込む開発プラットフォーム

03

エンタープライズ・ デリバリー方法の 決定：構築か、 導入か、オーケスト レーションか

企業の関心がAIを活用した開発から運用段階のデリバリーへと移行するにつれ、AIを活用したデリバリー・システムを組織で独自開発すべきか、エンタープライズ向けに設計されたシステムを導入すべきか、または両方を併用すべきかという、実務上の問題が浮かび上がってきました。

実際には、ほとんどの企業は両方を併用することになるでしょう。市場では、個々の開発者レベルでの生産性を劇的に高める優れたAIツールやエージェント・ハーネスが生み出されてきました。しかし、導入規模が拡大するにつれ、より幅広いソフトウェア・デリバリー・ライフサイクル全体における、そうしたエージェントによるオペレーションを統制するシステムがますます不可欠になってきています。

通常、組織がAIを活用した開発システムを評価する際に、次の3つのアーキテクチャーに関する選択肢に直面します。

- 内部開発：モデル、エージェント・ハーネス、ガバナンス・レイヤー、デリバリー・ワークフローをカスタム・システムに組み込み
- AIツールチェーンの作成：複数のAIツールを導入し、そうしたツールのオーケストレーション、ガバナンス、統合を行う仕組みを内部構築
- SDLCパートナーを統合：エンタープライズ・デリバリー全体でAIをオーケストレーションするよう設計されたシステムをデプロイ

大半の企業では、単にツールを購入して連携するだけでは足りません。同時に、最新のデリバリー基盤をあらゆるレイヤーにわたって開発し保守できる組織は、現実にはほとんど存在しません。



最初はAIツールの導入を決定したつもりでも、あっという間にエンタープライズ・システムのアーキテクチャー・プロジェクトへと発展してしまいます。こうしたAIツールは、単独で運用するだけでは不十分です。システム全体が一貫して機能する必要があります。そうした統合なしには、組織は局所的な開発業務を高速化させるだけで、企業としてのデリバリー成果向上につながらないリスクを負うことになります。

次世代のソフトウェア開発で成功を収めるには、単なる高速化だけでなく、一貫性が必要です。

成功する企業は、次のようなシステムを運用するでしょう。

- 人間とエージェントのワークフローをオーケストレーションするシステム
- 運用およびアーキテクチャーに関するシグナルを可視化するシステム
- ガバナンスを開発に直接組み込むシステム
- 企業としての成果創出につながるよう開発者の生産性を高めるシステム

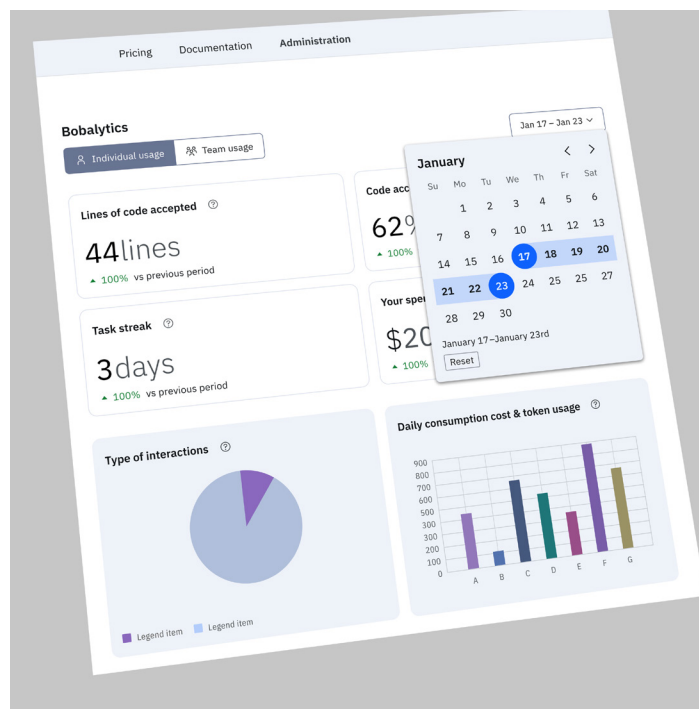
企業の経営者が現在直面している問題は、もはやAIがソフトウェア開発や生産性を高めるかどうかではありません。そのトランスフォーメーションはすでに完了しました。多くの企業にとって、単なる業務の高速化ではなく、組織の一体感を高めるシステムが問題解決の鍵となるでしょう。また、そのシステムは、ますますSDLCパートナーに近づいていくでしょう。

04 SDLCパートナーを 導入する

IBMは、企業から一貫して寄せられていたニーズに応えるかたちで、IBM Bob（SDLCパートナー）を開発しました。業界を問わず、組織が求めているのは、単なるコード生成の高速化ではありません。求められているのは、企業におけるデリバリーの実態を理解しているシステムです。

企業のニーズ

- 開発者とAIエージェントを包括的にオーケストレーションするシステム
- ガバナンスとセキュリティーを開発ワークフローに直接組み込み
- 複雑なアーキテクチャーとレガシー・システムを横断して動作するAI
- コスト、運用シグナルとデリバリー成果を可視化する機能



主要な問題は既存ツールによって解決されているものの、そうしたツールを組み合わせるシステムを構築する責任は依然として企業側にありました。BobはエンドツーエンドのSDLCパートナーとして設計され、こうした機能を開発システムに直接統合できるため、さまざまなモデル、ハーネス、ガバナンス・レイヤー、オーケストレーション・ツールから社内ですべてシステムを組み上げる必要はありません。

AIを活用したデリバリー・システムから、AIによってオーケストレーションされたデリバリー・システムへと移行する組織が、次世代のソフトウェア・デリバリーで他社をしのぐ成果を上げることになるでしょう。Bobは、企業のそうした変化、つまり連携されていない複数のツールから、一貫性があり統制されたSDLCパートナーへの移行を支援するために開発されました。

[IBM Bobの詳細はこちら](#) →

お客様の声

IBMとの連携を通じてIBM Bobのベータ版をテストしたお客様の声。

全コードのリファクタリングを行うことができました。問題になると考えていたことはすべて、Bobによって解決できました。古い.NET SOAPサービスは、クリーンなアーキテクチャーを備えた最新のREST APIに完全にリファクタリングされました。

Veran Pokornić氏

APIS IT社

ソリューション・アーキテクト

Bobを通じて、IBMと非常にスムーズにコラボできました。このフレームワークにより、取り組みを体系化、高速化して、確信を高めることができました。また、当社のビジネスとクライアントに数値化できる価値を提供できるようになりました。

Sareshan Govender氏

Blue Pearl社

グループCEO

Bobを使用して、ドキュメンテーションの欠如、大規模なコード・ベース、古いコードなど、当社が抱えていた複数の課題を解決できました。得られた成果物を新人の開発者に共有することで、すぐに状況を理解して、貢献してもらえるようになりました。

Todd Stewart氏

Heartland Co-op社

アプリケーション・アーキテクト

Bobは、AIを活用した開発に関する私たちの考えを大きく前進させる存在です。人がアカウントビリティを維持していることを実感しつつ、より迅速に前進することができます。

Uwe Graf氏

EasiRun Europa GmbH社

モダナイゼーション・アーキテクト

© Copyright IBM Corporation 2026

IBM、IBMのロゴ、IBM Bobは、米国およびその他の国々におけるInternational Business Machines Corporationの商標または登録商標です。その他の製品名およびサービス名は、IBMまたは他社の商標である場合があります。IBM商標の最新リストは、ibm.com/jp-ja/legal/copytradeでご確認いただけます。

本書は最初の発行日時点における最新情報を記載しており、IBMにより予告なしに変更される場合があります。

IBMが事業を展開しているすべての国または地域において、すべての製品を利用できるわけではありません。

示されている例は、説明のみを目的として提供されています。実際の結果はお客様の設定や条件により異なるため、一般的に期待される結果を提供するものではありません。

IBM製品およびプログラムを使って他社製品またはプログラムの動作を検証する場合は、お客様の責任で行ってください。IBMは、他社の製品やプログラムに責任を負いません。

本書の情報は「現状のまま」で提供されるものとし、明示または暗示を問わず、商品性、特定目的への適合性、および非侵害の保証または条件を含むいかなる保証もしないものとします。IBM製品は、IBM所定の契約書の条項に基づき保証されます。

ITシステムや製品は完全に安全であると捉えるべきではなく、不適切な使用やアクセスを防止する上で完璧な効果のある、製品、サービス、セキュリティ対策は1つ也没有せん。IBMでは、いずれの当事者による不正行為または違法行為によっても、いかなるシステム、製品もしくはサービスまたはお客様の企業に対して影響が及ばないことを保証することはありません。

お客様は、自己の責任ですべての関連法規および規則を順守しなければならないものとします。IBMは法律上の助言を提供せず、IBMのサービスまたは製品を使用することお客様による法律または規則の遵守が確約されると表明することも保証することありません。

