

ホワイトペーパー

トークン税(トークンコスト) という代償: 自律型AIエージェントが、 本来守るはずだった利益率を 圧迫する理由

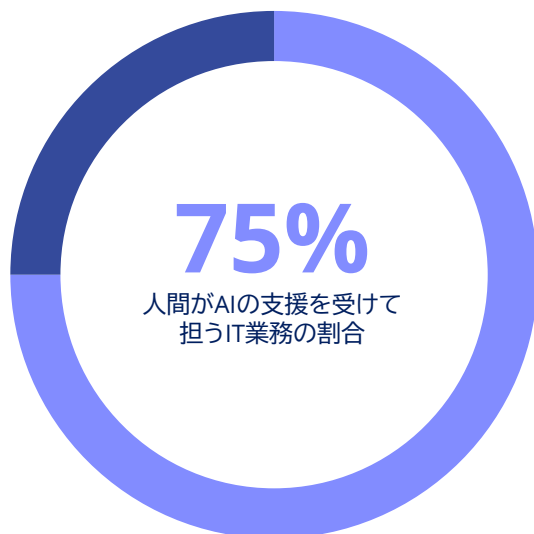
COO、CXO、オペレーション担当VP向けガイド



大きな可能性を秘めるAI

AIによって、ソフトウェア開発のあり方は一変しました。かつてないほど迅速かつ容易に、そして高度な開発が可能になっています。

開発者は、わずか1年前には考えられなかったスピードで開発し、リリースできるようになりました。現在では、開発者の84%がAIを活用して作業を効率化しています([Stack Overflow Developer Survey, 2025年](#))。以前なら数週間かかっていたアプリケーション全体のプロトタイプも、今では数日で骨組みを構築できます。ソフトウェア開発の民主化は現実のものとなり、その流れはさらに加速しています。CIOも、この先に何が待っているのかを認識しています。Gartnerは、2030年までにIT業務の75%をAIの支援を受けた人間が担い、25%をAIだけで遂行するようになる**AIがまったく関与しないIT業務はなくなると予測しています**([Gartner, 2025年11月](#))。これは遠い未来の予測ではありません。多くの企業のIT部門が今まさに、この未来を見据えて計画を立てています。



AIへの投資には十分な根拠があり、その方向性も間違っていないかもしれません。しかし、より多くの製品をリリースできるようになったからといって、取締役会が期待するような抜本的なビジネス成果には、まだつながっていません。そして、その理由の説明を求められているのがITリーダーです。

このギャップは、テクノロジーの問題ではなく、アーキテクチャの問題です。その核心には次の3つの問いがありますが、そのいずれについても、明確な答えを出している企業はまだ多くありません。

1. AIを活用した開発は、エンジニアを増員するよりも、実はコストがかかっていないか？
2. AIが生成したコードを、企業の業務を大規模に運用するために信頼して使用できるか？
3. AIが何を構築し、何をデプロイしているのかを、どのように可視化し、管理し続けるのか？

コストの問題:技術的負債を増やすために、余計なコストを払っていないか？

AI支援型開発がもたらすメリットは明確でした。ソフトウェア開発にかかるコストと時間を削減することです。しかし多くの企業では、当初のビジネスケースでは想定されていなかった形へと、コスト構造が変化しているのが現実です。

ソフトウェア開発で最もコストがかかるのは、コードを作成する作業ではありません。コーディングがアプリケーションのライフサイクル全体に占める割合は、およそ10%にすぎません。残りの90%を占めるのは、レビュー、テスト、セキュリティパッチの適用、デバッグ、リファクタリング、そしてビジネス要件の変化に応じた継続的な保守です。AIコーディングアシスタントによって、この10%の作業は高速化しました。しかし、増え続けるデータが示しているのは、同時に残りの90%のコストが増大しているという現実です。

Google、Microsoft、Metaおよびその他の企業が所有するリポジトリの2億1,100万行のコードについて、[GitClearが2025年に実施した分析](#)によると、AI支援型プロジェクトでは、2年前と比べてコードブロックの重複が8倍に増加していました。また、作成後2週間以内に修正または破棄されたコード行を示す「コードチャーン」は、2020年から2024年にかけて2倍になっています。GitClear創業者のBill Harding氏は、AI生成コードを、技術的負債を積み上げる「新しいクレジットカード」にたとえています。その場では有意義な投資に見えても、請求書が届くのは後になってからです。[Harnessの『The State of Software Delivery 2025』](#)でも、現在では大半の開発者が、新機能の開発よりもAI生成コードのデバッグに多くの時間を費やしていることが明らかになっています。

そして、トークン税(トークンコスト)という代償がこの問題をさらに深刻にします。AIがコードを1行生成するたびに、トークンのコストが発生します。顧客との対話に使われるトークンとは異なり、アプリケーションコードの生成に使われたトークンが生み出すのは、その後も継続的な保守投資を必要とする資産です。その資産が価値を生み出すより速いペースで負債を積み上げれば、経済性は初日から悪化の一途をたどります。

また、AI開発ツール自体のコスト構造も予測しにくくなっています。GitHub Copilotは2026年6月、従来の定額制のプレミアムリクエスト単位モデルから、トークンベースのAIクレジットを使用する従量課金制へと移行しました。GitHub自身も、旧モデルでは「簡単なチャットでの質問と、数時間に及ぶ自律型コーディングセッションに同じ料金がかかる」可能性があったこと、さらに増大する推論コストを負担し続けることは「もはや持続可能ではない」と説明しています([GitHub Blog, 2026年4月](#))。AI開発コストを定額料金として予算化してきた企業は今、ワークフローごとの支出額を把握しにくい変動費型の環境で運用することを余儀なくされています。

本来問うべきことは、AI開発ツールを使う価値があるかどうかではありません。価値はあります。問うべきことは、AIを使ってより多くのコードをより速く生成するという現在のアプローチが、本当に開発総コストを削減しているのか、それとも単に提供スピードを前倒しする一方で、翌年の予算に跳ね返ってくる保守負担を積み上げているだけなのか、ということです。

パフォーマンスとセキュリティの問題: AIが構築したものを信頼できるか？

迅速に提供できても、提供したものが企業規模の運用に耐えられると信頼できなければ、価値にはつながりません。この点について、現在明らかになりつつある事実は、ITリーダーが慎重になる十分な理由を示しています。

CodeRabbitが実際の470件のプルリクエストを分析したところ、AI生成コードでは、人間が作成したコードと比較して、問題の総数が1.7倍、特にセキュリティ脆弱性は2.74倍多く発生していました([CodeRabbit『State of AI vs. Human Code Generation』、2025年12月](#))。[ISSRE 2025で発表されたIEEEの調査](#)でも、AI生成コードは構造的には単純であるにもかかわらず、高リスクのセキュリティ脆弱性がより高い割合で発生することが示されています。つまり、リスクの原因は単にコードの複雑性にあるのではなく、ロジックの構築方法そのものにあることが示唆されています。

AIツールが生成するのは、目の前にある局所的な問題に対する解決策です。システム全体のアーキテクチャ、社内の抽象化レイヤー、共有ユーティリティ、さらには環境内でのデータの流れを規定するセキュリティ上の制約まで、すべてを完全に把握しているわけではありません。その結果、単体テストを通過し、最初のレビューでは問題がなくても、エッジケースや本番環境の負荷下で初めて表面化する脆弱性がコードに入り込む可能性があります。

AIコーディングツールを使用している開発者の76%が、少なくとも時折、自分では完全に理解していないコードを生成したことがあると回答しています([Stack Overflow、2026年](#))。この数字は、運用面で重大な意味を持ちます。規制当局、法務部門、取締役会から「なぜシステムがこのような動作をしたのか」とCIOが問われたとき、「AIが生成したもので、開発者自身も仕組みをよく理解していなかった」という説明は通用しません。しかし、構造的なガバナンスレイヤーを設けずにAI生成コードを本番環境にデプロイしている組織は、実質的にそのような説明責任体制を抱えていることとなります。

Gartnerが発表した2026年の主要予測には、AI生成アプリケーションの出力に対するガバナンス不足を背景として、「AIによる死亡」をめぐる法的請求が世界で2,000件を超えるという予測も含まれています([Gartner、2025年10月](#))。これは将来起こるかもしれない仮説上のリスクではありません。規制対象または顧客向けのワークフローを動かす実質的な権限を、AI生成コードにすでに与えている企業にとって、現在進行形のリスクなのです。

信頼性の問題: AIが何を構築しているのかを把握できているか？

スピードとセキュリティは運用上の課題です。一方、監査可能性は構造上の課題です。CIOやエンタープライズアーキテクトにとって、後から対応することが最も難しい問題とも言えます。

ケースの振り分け、適格性の判定、アクセス権の付与、顧客向け出力の生成など、アプリケーションが何らかの判断を下す場合、その判断に対して責任を負うのは、コードを作成したAIではなくIT部門です。そのアプリケーションをデプロイしたのはIT部門であり、それを承認したのはそのリーダーだからです。

従来の開発環境では、ロジックを確認できるため、こうした説明責任を果たすことができます。監査担当者から判断の理由を問われれば、開発者が該当するコードパスをたどって答えを見つけることができます。しかし、アプリケーションロジックがAIによって生成された環境では、このトレーサビリティが保証されるとは限りません。AI生成コードは動作する結果を生み出しますが、個々の実装方法が選択された理由は、人間が意図を持って設計した場合のように文書化されません。監査で「このエッジケースにおいて、なぜシステムはこの判断を下したのか」と問われたとき、その答えが「AIがそのようにコードを作成したから」では、正当な説明にはなりません。

企業レベルで最も重要な信頼性の問題はここにあります。コードが平均的に正しく動作するかどうかが問題なのではありません。将来のどの時点で、誰から特定の結果について説明を求められても、相手が理解し、行動につなげられる言葉で説明できるかが重要なのです。

視覚的な宣言型アプリケーションロジックなら、この問題を構造的に解決できます。ビジネスルール、デシジョンテーブル、ワークフローのルーティングを、生成コードではなく確認可能な設定として表現すれば、フォレンジック分析を行わなくても、あらゆるデシジョンパスを追跡できます。ロジックは、エンジニアだけでなく、コンプライアンス部門、法務部門、ビジネス部門の関係者も理解できるレイヤーに存在します。その結果、監査は「コードを発掘して調査する作業」ではなく、「設定をレビューする作業」になります。

3つの問いすべてに答えるアーキテクチャ

コスト、セキュリティ、信頼性という3つの問題を同時に解決するために必要なアーキテクチャの転換は、AIの利用を制限することではありません。開発スタックのどの段階でAIを活用するかを変えることです。

AIが真価を発揮するのは、データモデル、ワークフロー定義、デシジョンテーブル、宣言形式で表現されたルーティングロジックなど、アプリケーションの上位構造の生成を高速化する場合です。これらは、命令型コードとして実装方法を記述するのではなく、アプリケーションが「何をするのか」を記述する成果物です。生成コードとは異なり、読みやすく、監査可能で、保守もしやすくなります。ビジネス要件が変わった場合も、モデルを変更すれば、それに応じてアプリケーションの動作も変わります。コードを次々に修正する必要も、それに伴うリグレッションリスクもありません。

これが、モデルファースト開発の実践です。AIには、本当に得意な仕事をさせます。つまり、意図を構造に変換し、自然言語から設定を生成し、アーキテクチャの定義を高速化します。その成果物はコードではなくモデルです。必要に応じて行われるコード生成は、その下層で実装上の詳細として処理されるものであり、主要な成果物ではありません。

運用面でのメリットも明確です。保守対象はコードベースではなくモデルであるため、開発チームが保守に費やす時間を削減できます。ビジネスロジックが宣言形式で表現されるため、セキュリティレビューも容易になります。デシジョンレイヤーをエンジニアの支援なしで確認できるため、コンプライアンス監査も迅速になります。そして、トークンへの支出を、開発において最も価値の高い部分、つまりアプリケーションを「どのように動かすか」ではなく「何をさせるか」を定義する部分に集中できます。

[Gartnerが2025年後半に実施したI&O調査](#)によると、ROIへの期待を完全に満たしているAIインフラストラクチャのユースケースは、わずか28%です。期待と現実のギャップは、AIの能力不足によるものではありません。アーキテクチャの問題です。AIを開発に活用してROIを実現している企業が、必ずしもより高度なモデルを使用しているわけではありません。そうした企業は、生成コードほど急速に価値が低下しない成果物を構築するためにAIを活用しているのです。



アーキテクチャという選択

ITリーダーが迫られているのは、スピードか安全性かという二者択一ではありません。そのように捉えてしまうからこそ、現在の状況が解決困難な制約のように見えるのです。実際に選ぶべきは、長期的なコストとリスクの特性が大きく異なる、2つのアーキテクチャです。

AI生成コードを中心とするアーキテクチャでは、初期段階では迅速に成果を得られる一方、時間とともに保守負担が積み上がり、監査可能性が制限され、コードベースが大きくなるほどセキュリティリスクも増大します。これに対して、AIを組み込んだモデルファースト開発を中心とするアーキテクチャなら、最初から監査可能な設計で成果物を構築できます。モデルレイヤーで保守できるうえ、開発において持続的な価値を実際に生み出す部分にトークンを使うことができます。

IT部門のリーダーは、AIによる開発高速化を求めるビジネス部門に「ノー」と言う必要はありません。答えは、開発を減速することではありません。ビジネスが持続的に活用でき、IT部門がその妥当性を説明できる、別の開発アプローチを選ぶことです。

適切なAI投資とは、コードをより速く作成することではありません。作成し直す必要のないアプリケーションモデルを構築することです。





ありがとうございました

Pegaについて

Pegaは、エンタープライズDXを実現する最先端のAI搭載プラットフォームを提供します。世界有数の企業・組織が、ワークフローの自動化、顧客体験のパーソナライズ、レガシーシステムのモダナイゼーションを実現するためにPegaのテクノロジーを信頼して採用し、業務のあり方を再構築しています。1983年以来、拡張性と柔軟性の高いアーキテクチャにより継続的なイノベーションを促進し、お客様の自律型企业へのジャーニーを加速しています。**[PEGA.COM/JA](https://www.pega.com/ja)**