

# 情報システム部門における生成 AI 活用事例とソフトウェア開発への影響

Gemini Deep Research

## はじめに

近年、情報システム部門において生成 AI（Generative AI）の活用が急速に進展しています。生成 AI は、大量のデータから学習し、新しいコンテンツやアイデアを生成する AI 技術です。情報システム部門の様々な業務において、効率化や新たな価値創造の可能性を秘めています。<sup>1</sup>

この記事では、情報システム部門における生成 AI 活用の事例を、国内外を網羅的に調査し、特にソフトウェア開発におけるコーディング作業の効率化に焦点を当てて解説します。さらに、生成 AI 活用によるソフトウェア開発の将来展望についても考察し、今後の情報システム部門における AI 活用の在り方について示唆を提供します。

## 生成 AI を活用したソフトウェア開発事例

生成 AI は、ソフトウェア開発の様々な工程で活用されています。このセクションでは、生成 AI の種類（LLM、コード生成 AI など）ごとに分類し、具体的な事例を紹介します。

### LLM（大規模言語モデル）の活用事例

LLM は、自然言語処理を得意とする AI であり、ソフトウェア開発においては、主に以下の用途で活用されています。<sup>2</sup>

- **要件定義・設計支援:** 自然言語で記述された要件を分析し、矛盾や曖昧さを検出したり、設計ドキュメントの自動生成を支援したりします。
- **テストケース作成:** テストケースの自動生成や、テストデータの作成を支援します。
- **ドキュメント作成:** ソースコードから API ドキュメントやユーザマニュアルなどを自動生成します。
- **コミュニケーション支援:** 開発者間のコミュニケーションを円滑化するためのチャットボットや、質問応答システムなどに利用されます。

### 具体的な事例

- **事例①:** ある企業では、LLM を活用して、顧客とのコミュニケーションを自動化するチャットボットを開発しました。これにより、顧客対応の効率化と顧客満足度向上を実現しました。<sup>3</sup>
- **事例②:** ある開発チームでは、LLM を活用して、ソースコードから API ドキュメントを自動生成するツールを開発しました。これにより、ドキュメント作成の手間を削減し、常に最新の状態に保つことができるようになりました。<sup>1</sup>

## コード生成 AI の活用事例

コード生成 AI は、プログラムコードの生成を支援する AI であり、ソフトウェア開発においては、主に以下の用途で活用されています。

- **コードの最適化**
  - コード補完: 開発者が記述中のコードを予測し、候補を表示することで、コーディングの速度と正確性を向上させます。
  - コード生成: 自然言語による指示や、既存のコードを参考に、新しいコードを自動生成します。
  - リファクタリング: 既存のコードをより効率的かつ可読性の高いコードに書き換えることを支援します。
- **コードの品質向上**
  - コードレビュー: コードの品質をチェックし、バグやセキュリティ上の脆弱性を検出します。
  - テストの自動化: 単体テストなどのテストコードを自動生成します。

### 具体的な事例

- **事例③**: ある企業では、コード生成 AI を導入することで、開発者が記述するコード量を削減し、開発工数を **30%削減**することに成功しました。<sup>4</sup>
- **事例④**: ある開発チームでは、コード生成 AI を活用して、単体テストのコードを自動生成するツールを開発しました。これにより、テストコードの作成にかかる時間を大幅に短縮し、テストの網羅性を向上させることができました。<sup>5</sup>
- **事例⑤**: LINE ヤフー株式会社では、約 **7,000 人**のエンジニアを対象に、AI ペアプログラマー「**GitHub Copilot for Business**」を導入しました。GitHub Copilot は、AI がコーディングを支援してくれるツールであり、コーディング作業の効率化、品質向上に効果を発揮しています。<sup>6</sup>
- **事例⑥**: NEC では、システム開発プロセスの中に **GitHub Copilot** を導入し業務の効率化を図っています。<sup>8</sup>

### その他の生成 AI 活用事例

- **音楽生成 AI**: OpenAI Jukebox や Magenta by Google など、音楽生成 AI は、様々なジャンルやスタイルの音楽を生成することができます。
- **画像生成 AI**: DALL-E や StyleGAN など、画像生成 AI は、テキストによる指示から画像を生成することができます。
- **パーソナライズされた UI**: 生成 AI を用いて、ユーザーの行動や好みに合わせた UI を動的に生成することができます。<sup>9</sup>
- **予測メンテナンス**: AI アルゴリズムを用いて、ソフトウェアの障害や欠陥を事前に予測することができます。<sup>1</sup>
- **インテリジェントデバッグ**: AI を用いて、バグの追跡と問題解決を効率化することができます。<sup>1</sup>
- **デプロイの自動化**: AI を用いて、デプロイを自動化し、最適なデプロイ時期を予測することができます。<sup>1</sup>
- **セキュリティ、コンプライアンス、脅威検出**: AI を用いて、セキュリティとコンプライア

ンスを強化し、脅威を検出することができます。<sup>1</sup>

- **ドキュメントの強化:** 生成 AI は、コードの機能を要約したり、アルゴリズムを説明したり、コンテキストを提供したりすることで、エンジニアリングチームがドキュメントを生成するのを支援することができます。<sup>10</sup>

## コーディング作業の効率化に関する分析

生成 AI は、コーディング作業の効率化に大きく貢献することが期待されています。このセクションでは、具体的な分析結果を交えながら、その効果について考察します。

### 自動化率と時間短縮効果

生成 AI によるコーディング作業の自動化率は、現状では 10% から 15% 程度と言われています。<sup>11</sup> これは、生成 AI がまだすべてのコーディング作業を自動化できるわけではないためです。しかし、包括的なアプローチをとることで、30% 以上の効率化も可能であるとされています。<sup>4</sup> 包括的なアプローチには、生成 AI によるコード生成だけでなく、他のタスクへの生成 AI の活用、適切なベースラインとメトリクスの設定などが含まれます。

また、中国の Ant Group が行った調査では、生成 AI を利用したプログラマーの生産性は、利用していないプログラマーと比較して 55% 向上したという結果が出ています。<sup>12</sup> このことから、生成 AI はコーディング作業の時間短縮に大きく貢献すると言えるでしょう。

### 品質への影響

生成 AI が生成するコードの品質については、まだ議論の余地があります。GitHub によると、開発者は Copilot のコード補完を 30% の確率で受け入れているとのこと。<sup>13</sup> これは、生成 AI が生成するコードが必ずしも完璧ではないことを示唆しています。また、生成 AI によってコード生成が加速され、全体的なコードの生産量が増加するため、技術的負債が増加する可能性も懸念されています。<sup>13</sup>

しかし、生成 AI はコードレビューの自動化や、バグの検出にも活用することができるため<sup>1</sup>、適切に利用することで、コーディング作業の品質向上に貢献することが期待されます。重要なのは、開発者が生成 AI の出力を盲目的に信頼するのではなく、注意深く **oversight** を行うことです。<sup>13</sup>

### 生成 AI によるコーディング作業効率化の分析における洞察

- 生成 AI は、特にジュニア開発者の生産性に大きな影響を与える可能性があります。<sup>12</sup> これは、ジュニア開発者が生成 AI の提案を受け入れる割合が、シニア開発者よりも高いためと考えられます。
- 生成 AI による効率化を最大限に引き出すためには、コード生成だけでなく、包括的なアプローチをとる必要があります。<sup>4</sup>
- 生成 AI によってコード生成量が増加すると、適切に管理されないとコードの品質が低下する可能性があります。<sup>13</sup>

# 生成 AI 活用によるソフトウェア開発の将来展望

生成 AI は、今後さらに進化し、ソフトウェア開発に大きな影響を与えることが予想されます。

## 生成 AI の進化の方向性

- **文脈理解の向上:** より高度な自然言語処理技術により、生成 AI は開発者の意図をより正確に理解し、より適切なコードを生成できるようになるでしょう。<sup>1</sup>
- **バイアスの軽減:** 生成 AI の学習データに含まれるバイアスを軽減するための研究が進み、より公平で信頼性の高いコード生成が可能になるでしょう。<sup>1</sup>
- **専門分野への特化:** 特定のプログラミング言語やフレームワークに特化した生成 AI が登場し、より高度なコーディング作業を支援できるようになるでしょう。<sup>1</sup>

## ソフトウェア開発への影響

- **開発者の役割の変化:** 生成 AI がコーディング作業の一部を自動化することで、開発者はより創造的な作業に集中できるようになり、より高度なスキルが求められるようになるでしょう。<sup>10</sup>
- **開発プロセスの変化:** 生成 AI を活用することで、ソフトウェア開発プロセス全体がより効率化され、開発期間の短縮やコスト削減につながるでしょう。<sup>4</sup>
- **新たな開発手法の登場:** 生成 AI を基盤とした、これまでにない新しい開発手法やツールが登場する可能性があります。<sup>10</sup>
- **自然言語インターフェース:** 生成 AI は、ソフトウェア開発ツールのための自然言語インターフェースを可能にする可能性を秘めています。<sup>14</sup> 開発者は、自然言語コマンドを使用して開発環境、デバッグ、バージョン管理システムと対話できるようになり、プログラミングの知識が限られている人にとってよりアクセスしやすくなる可能性があります。
- **ソフトウェア開発の民主化:** 生成 AI は、プログラミングの知識が限られている人でもソフトウェア開発にアクセスできるようにすることで、ソフトウェア開発を民主化する可能性を秘めています。<sup>14</sup>

## 課題と解決策

生成 AI 活用によるソフトウェア開発には、いくつかの課題も存在します。

- **セキュリティ:** 生成 AI が生成するコードにセキュリティ上の脆弱性が含まれる可能性があります。
- **倫理:** 生成 AI が生成するコードの著作権や、AI による雇用への影響など、倫理的な問題も検討する必要があります。
- **人材育成:** 生成 AI を活用できる人材の育成が重要になります。

これらの課題に対しては、以下の解決策が考えられます。

- セキュリティ対策を強化した生成 AI の開発
- 倫理的なガイドラインの策定
- 生成 AI に関する教育プログラムの提供

## 結論

生成 AI は、情報システム部門におけるソフトウェア開発の効率化に大きく貢献する可能性を秘めています。コーディング作業の自動化や品質向上、開発プロセスの効率化など、様々な効果が期待されます。

今後、生成 AI はさらに進化し、ソフトウェア開発に大きな影響を与えることが予想されます。情報システム部門は、生成 AI の進化を注視し、そのメリットを最大限に活かせるよう、積極的に活用していく必要があるでしょう。

特に、生成 AI は開発者の役割を大きく変え、より創造的な仕事に集中できる環境を作る可能性があります。同時に、情報システム部門は、セキュリティや倫理、人材育成といった課題にも適切に対処していく必要があります。生成 AI と人間が協調することで、より質の高いソフトウェアを、より効率的に開発できる未来を目指していくことが重要です。

## 生成 AI 活用事例一覧

Company	Generative AI Type	Use Case	Development Process	Software	Effect	Challenges
NTT データ	LLM	顧客要望の分析	要件定義	-	要件定義の工数削減	-
-	LLM	API ドキュメントの自動生成	ドキュメント作成	-	ドキュメント作成の効率化	-
-	コード生成 AI	コーディング作業の自動化	実装	-	開発工数の 30%削減	-
-	コード	単体テ	テスト	-	テスト	-

	生成 AI	ストコードの自動生成			コード作成時間の短縮	
LINE ヤフー	コード生成 AI	AI ペアプログラマー「GitHub Copilot for Business」の導入	実装	-	コーディング作業の効率化、品質向上	-
NEC	コード生成 AI	システム開発プロセスへの GitHub Copilot の導入	-	-	業務効率化	-

## 引用文献

1. Top Generative AI Use Cases for Businesses in 2024 - Labellerr, 3月8, 2025 にアクセス、<https://www.labellerr.com/blog/the-most-elaborative-guide-for-generative-ai-with-top-7-use-cases/>
2. Top 11 Generative AI Use Cases in Software Development - Index.dev, 3月8, 2025 にアクセス、<https://www.index.dev/blog/11-generative-ai-use-cases-software-development>
3. 生成 AI の国内開発事例 9 選！開発に必要なことや活用事例も紹介 - WEEL, 3月8, 2025 にアクセス、<https://weel.co.jp/media/gen-ai-national-case>
4. Beyond Code Generation: More Efficient Software Development ..., 3月8, 2025 にアクセス、<https://www.bain.com/insights/beyond-code-generation-more-efficient-software-development-tech-report-2024/>
5. AI でコーディングを自動化しよう！コード生成 AI の使い方やおすすめを紹介 - aslead, 3月8, 2025 にアクセス、<https://aslead.nri.co.jp/ownedmedia/development/post-1009/>
6. 【生成 AI 導入】システム開発を加速！メリットと企業事例を紹介 - ナンバーワンソリューションズ, 3月8, 2025 にアクセス、<https://no1s.biz/blog/6510/>
7. 株式会社スパイクスタジオ - 生成 AI を活用したシステム開発事例 8 選 ..., 3月8, 2025 にアクセス、<https://spikestudio.jp/blog/EEX9xs5r>
8. 生成 AI でシステム開発ができる時代が到来！活用事例や効率化できる ..., 3月8, 2025 にアクセス、<https://weel.co.jp/media/gen-ai-system-dev/>

9. The Role of Generative AI in Application Development - OutSystems, 3月8, 2025 にアクセス、<https://www.outsystems.com/ai/generative/software/>
10. Generative AI for Developers | IBM, 3月8, 2025 にアクセス、<https://www.ibm.com/think/topics/generative-ai-for-developers>
11. www.bain.com, 3月8, 2025 にアクセス、<https://www.bain.com/insights/beyond-code-generation-more-efficient-software-development-tech-report-2024/#:~:text=In%20practice%2C%20generative%20AI%20appears,AI%20and%20a%20broader%20agenda.>
12. Generative AI and labour productivity: a field experiment on coding, 3月8, 2025 にアクセス、<https://www.bis.org/publ/work1208.htm>
13. Does Using AI Assistants Lead to Lower Code Quality? - DevOps.com, 3月8, 2025 にアクセス、<https://devops.com/does-using-ai-assistants-lead-to-lower-code-quality/>
14. Enhancing Developer Productivity with Generative AI - XenonStack, 3月8, 2025 にアクセス、<https://www.xenonstack.com/blog/developer-productivity-with-generative-ai>
15. 生成 AI で変わる業務効率化！事例と使いこなし方を詳しく紹介, 3月8, 2025 にアクセス、<https://ai.yoshidumi.co.jp/navi/generative-ai-business-efficiency-improvement>