



生成AIが切り拓くソフトウェア開発の未来像

生成AIの登場により、ソフトウェア開発は大きな転換点を迎えています。これまでAIは人の開発作業を支援する存在でしたが、今後は、AIが人の実現したい価値や意図を理解し、設計や実装、品質保証、運用保守までを主体的に担うようになると考えられます。その先にあるソフトウェア開発はどのような姿になるのでしょうか。本稿では、人とAIが協働してソフトウェアを継続的に創出、進化させる未来像を示すとともに、その実現に向けてNTTコンピュータ&データサイエンス研究所が進める要求具体化、品質保証、運用保守の研究開発について紹介します。

キーワード：#ソフトウェア開発、#生成AI、#ソフトウェア進化

わかばやし けい
若林 慧

NTTコンピュータ&データサイエンス研究所

生成AIによって変わるソフトウェア開発

生成AIの急速な発展により、ソフトウェア開発におけるAI活用が大きく進展しています。当初は、プログラムコードの生成や補完など、開発者の個別作業を支援する用途が中心でした。近年では、Princeton Universityなどの研究者が開発した、

ソフトウェアの修正やテストを自律的に行うAIエージェント^{*1}であるSWE-agent⁽¹⁾のように、開発者から与えられた目的や指示を基に、既存のソースコードやドキュメントを理解し、一連の開発タスクを自律的に遂行する技術が登場しています。さらに、複数のAIエージェントが役割を分担しながら開発を進めるなど、AIの活用範囲は個々の作業支援

から開発プロセス全体へと広がりつつあります（図1）。

一方で、AIが大量かつ高速にソフトウェアを生成するようになると、新たな課題も生じます。例えば、大規模かつ高い信頼性を求められるソ

^{*1} AIエージェント：与えられた目的に基づき、生成AIが状況を理解して必要な作業を判断し、外部ツールなどを利用しながら複数の処理を自律的に実行する仕組み。

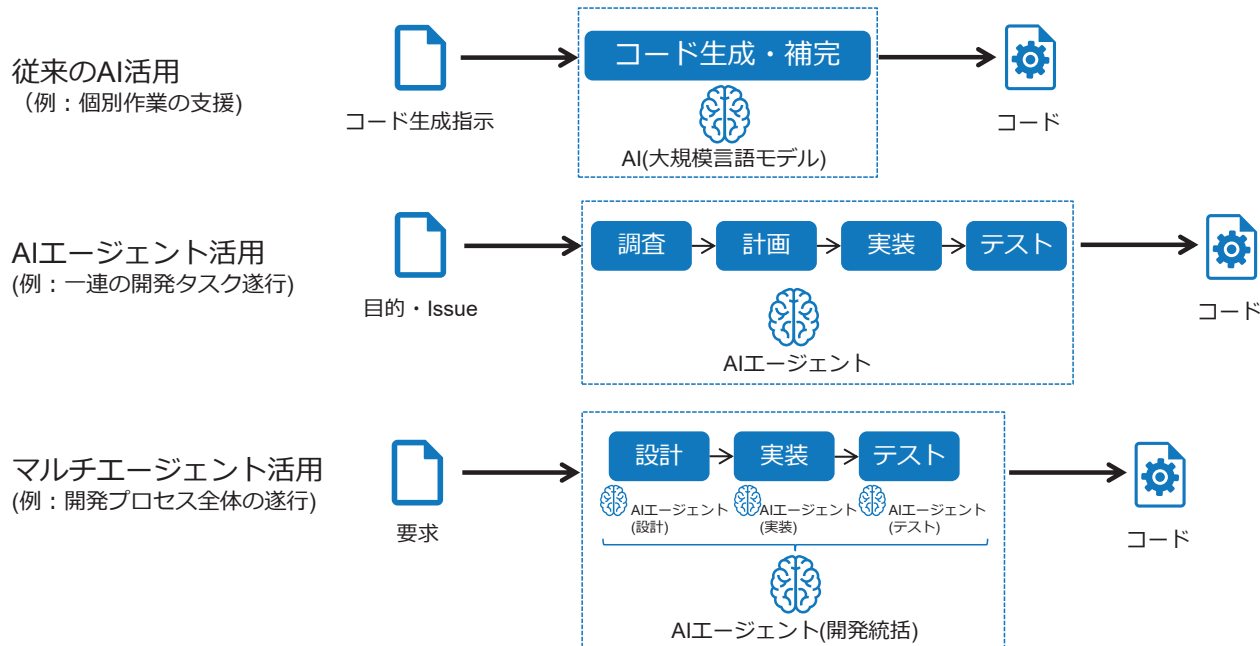


図1 ソフトウェア開発におけるAIの利用形態

ソフトウェア開発にAIを活用する場合、「何を、なぜつくるのか」という開発意図をAIと共有すること、その意図を満たしていることを迅速に確認すること、さらに運用中に得られる情報を次の開発や改善につなげることが重要になります。AIによる開発を高度化するためには、単にコード生成能力を高めるだけではなく、ソフトウェア開発を通じて生み出される情報や知識を、人とAIの双方が理解、活用できるかたちで扱うことが求められます。また、AI利用の拡大に伴い、AIの利用量に応じて発生するコストや、その投資対効果への関心も高まっています。こうしたAI利用コストを共通の指標で把握、評価するための枠組みづくりを目的として、The Tokenomics Foundation⁽²⁾が設立されています。

このような変化を踏まえ、NTTコンピュータ&データサイエンス研究所では、生成AIによる個々の開発作業の効率化にとどまらず、AIが

開発運用をより主体的に担う新たなソフトウェア開発のあり方を見据え、研究開発を進めています。

AIと人が協働するソフトウェア開発の未来像

将来のソフトウェア開発では、人が実現したい価値や方向性を示し、AIがその意図を理解して、要求の具体化から設計、実装、品質保証、運用までを一貫して支援し、さらにはAI自身が主体的に開発を進めるかたちへと変化していくと考えています(図2)。

このようなソフトウェア開発では、人とAIの役割も変化します。AIは、人から与えられた個別の指示に従って成果物を生成するだけではなく、開発の目的や背景、過去の判断、成果物間の関係などを理解し、その情報を基に次に必要となる作業を判断します。一方、人は、M. Aleneziが「コード作成中心から、AIの指揮、

検証、ガバナンス中心へ移行する」⁽³⁾と述べるように、すべての開発作業を直接実施するのではなく、実現したい価値や方向性を示すことや、AIによる判断や成果が目的に沿っているかを確認することなど、より上位の判断を担います。これにより、人とAIが互いに補完しながら、高速かつ継続的にソフトウェアを創出、改善できるようになると考えています。

その実現には、人とAIが開発に関する意図や判断基準を共有できることが重要です。従来のソフトウェア開発では、要件定義書や設計書、ソースコード、テスト仕様書などの成果物に開発情報が記録される一方で、設計時の判断理由や関係者間の議論など、一部の情報は暗黙知として残る場合があります。AIが開発を主体的に担うためには、こうした情報も含め、ソフトウェアが「何のためにつくられ、なぜその構造や振る舞いとなっているのか」を継続的に

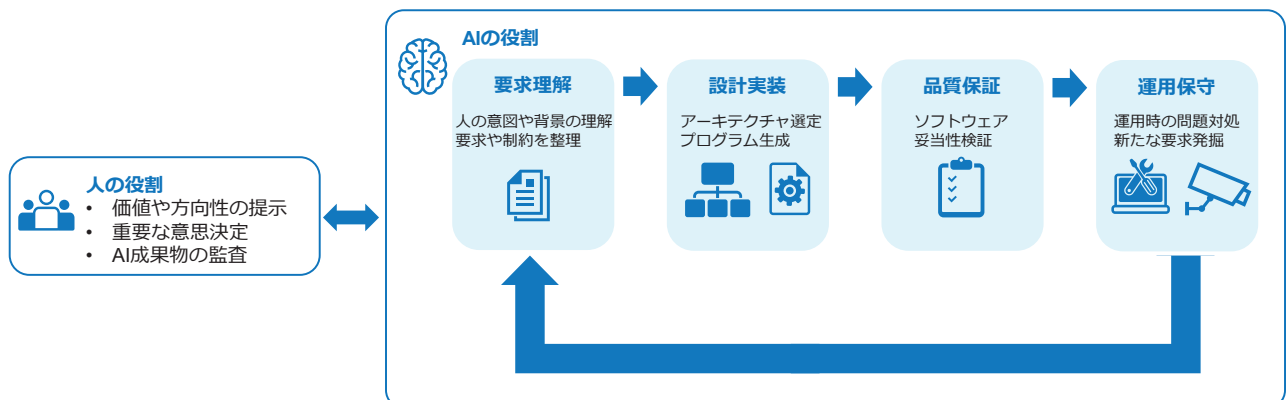


図2 AIと人が協働するソフトウェア開発の未来像

把握できることが求められます。

さらに、ソフトウェア開発を一方の工程としてとらえるのではなく、運用を通じて得られた情報を次の要求や開発へ反映し、継続的に改善していくことも重要です。例えば、利用状況やシステムの振る舞い、障害などの運用情報をAIが理解し、改善すべき事項を抽出することで、新たな要求の具体化やソフトウェアの変更につなげることができます。このように、要求具体化、開発、品質保証、運用で生み出される知識を蓄積し、相互に活用することで、AIがソフトウェアを継続的に理解し、進化させることが可能になります。

未来のソフトウェア開発を支える技術開発

前述した未来像の実現に向け、「要求具体化」「品質保証」「運用保守」の3つの領域で研究開発を進めています。要求具体化では、人の要求や

意図をAIが理解し、ソフトウェアとして具体化します。品質保証では、AIが要求、設計、ソースコードなどの開発情報を活用し、生成されたソフトウェアが要求や設計意図を満たしているかを確認します。運用保守では、AIがシステムの状態や運用中に得られる情報を収集、分析し、次の開発や改善につなげます。

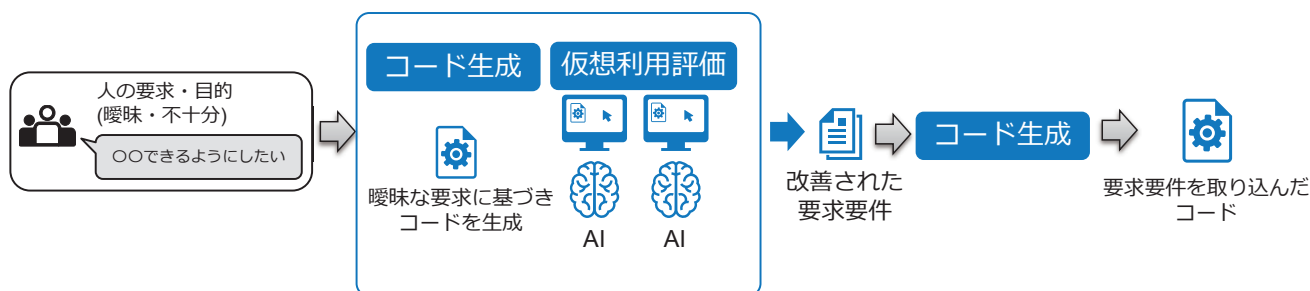
これらの領域を相互に連携させることで、要求から運用までを一貫して支援し、ソフトウェアを継続的に進化させることをめざしています。次に、各領域における主な技術開発の取り組みを紹介します。

要求具体化に関する主な取り組み

ソフトウェア開発の起点となる要求は、必ずしも開発に必要な情報が十分に整理された状態で与えられるとは限りません。利用者が実現したいことや業務上の課題として示され

る要求には、曖昧な表現や暗黙の前提が含まれる場合があります。開発者はソフトウェア利用と関係するさまざまなステークホルダとの対話を通じてその背景や意図を理解し、ソフトウェアとして実現すべき機能や制約へと要件を具体化してきました。AIがソフトウェア開発を主体的に担うためには、このような要求をAI自身が具体化していく過程が重要になります。そこで、人が示した要求や目的から、開発に必要な情報を段階的に具体化する技術の研究開発に取り組んでいます。人の要求に基づいて生成したソフトウェアに対して、AIにソフトウェア利用者を模倣させてユーザビリティを評価することで、ソフトウェア要求に不足している事項を抽出し、利用者の意図をより明確な要求要件へと定義することが可能となります（図3）。

さらに、将来的には、検証時にAIで模倣する対象を、ソフトウェアの利用者だけでなく、実際のソフトウェ



利用者をAIにより模倣し、ソフトウェアを評価
曖昧な要求・目的を明確化する技術

図3 要求具体化に関する取り組み

AIに携わる開発運用担当やシステム企画担当などステークホルダの役割へと拡充することで、ソフトウェアライフサイクル全体の品質を早期に確保する技術の確立をめざします。

品質保証に関する主な取り組み

生成AIによってソフトウェアを高速に生成、変更できるようになる一方で、その成果物が要求や設計意図を満たしていることをどのようにレビューするかが重要になります。AIが生成するソフトウェアの規模や変更頻度が増加すると、人がすべての成果物をレビューすることは難し

くなり、開発速度の向上に対して品質チェックがボトルネックとなる可能性があります。そのため、AIによる開発の高度化と併せて、品質保証についてもAIを活用した仕組みへと変えていく必要があります。

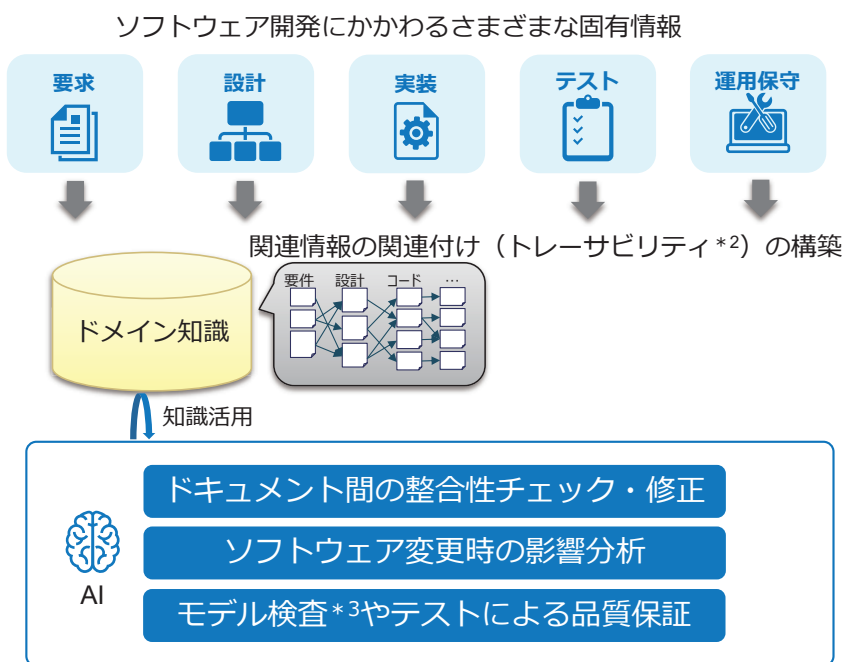
AIを活用した品質保証を実現するため、開発中に作成される要求、設計、ソースコード、テスト、運用保守などの情報を相互に関連付け、生成AIがソフトウェアの内容や変更の影響を理解するための技術に取り組んでいます。これにより、要求と設計、設計と実装などの間に不整合がないかを確認するとともに、確認すべき範囲を特定することで、開発成果物を横断した品質確認を可能に

します。また、生成AIを活用してテスト観点やテストケースを網羅的に生成する技術や、ソフトウェアの振る舞いをモデルとして表現し、想定した性質が満たされていることを検証する技術についても研究を進めています（図4）。

これらの技術を組み合わせることで、人が生成された成果物を1つひとつレビューするのではなく、AIが開発意図との整合性やソフトウェアの振る舞いを継続的に確認し、人は重要な判断や最終的な確認に注力できる開発をめざしています。

運用保守に関する主な取り組み

ソフトウェアの価値を継続的に高めるためには、開発時に想定した仕様や品質だけでなく、実際の運用環境でシステムがどのように利用され、どのように振る舞っているかを把握することが重要です。特に、AIがソフトウェアを継続的に変更するようになると、運用中に得られる情報を理解し、必要な対応や改善につなげることが、AIによる自律的な開発を実現するための重要な要素となり



AIが主体となりソフトウェアに関する品質を保証

図4 品質保証に関する取り組み

*2 トレーサビリティ：要求、設計、ソースコード、テストなど、ソフトウェア開発で作成される情報の対応関係を追跡できる状態、またはその仕組み。

*3 モデル検査：ソフトウェアの状態や振る舞いをモデル化し、要求された性質を満たすかを網羅的に探索して検証する手法。

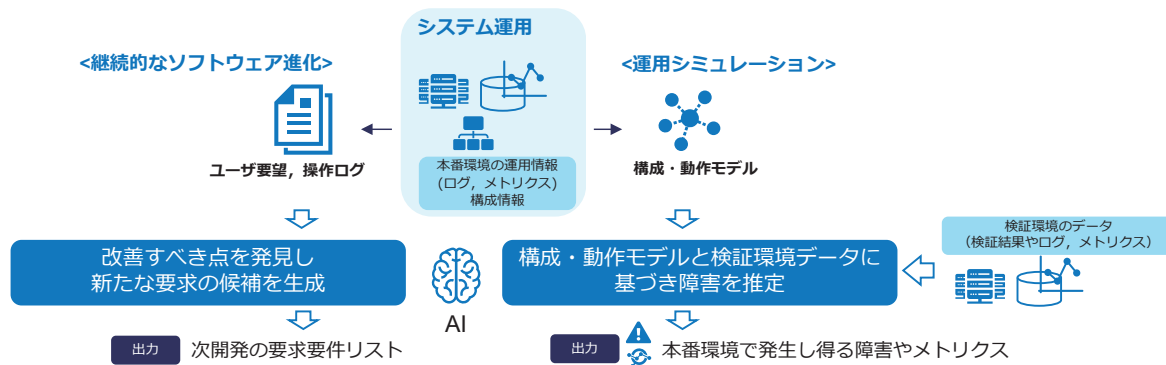


図5 運用保守に関する取り組み

ます。

そのため、システムの構成や振る舞い、運用時に発生する事象をAIが理解し、運用判断や開発へのフィードバックに活用する技術に取り組んでいます。例えば、検証環境と本番環境では、システム構成や負荷、利用状況などが異なる場合があります。検証環境では問題が確認されなくても、本番環境でのみ不具合が顕在化することがあります。そこで、システムの構成や動作をモデル化し、本番環境における振る舞いをシミュレーションすることで、実際の環境で発生し得る問題を事前に把握することをめざしています（図5）。

さらに、将来的には、運用中に得られる利用状況やシステムの状態などをAIが把握し、利用者のニーズや市場動向、技術動向などの外部情報を取り入れながら改善すべき点を自ら見出して、新たな要求の候補として提示することをめざしています。これにより、運用で得られた知識を次の要求具体化や開発へつなげ、ソ

フトウェアを継続的に改善することができるようになると考えています。

今後の展望

本稿では、生成AIが個々の開発作業を支援する段階から、開発運用のさまざまな活動を主体的に担う段階へと進展する中で、人とAIが協働してソフトウェアを継続的に進化させる未来像と、その実現に向けた技術開発について紹介しました。

今後は、各領域で生み出される情報や知識を相互に連携させ、要求の具体化から開発、品質保証、運用保守、さらに次の改善までを一貫して支援できる技術の確立を進めます。将来的には、人が実現したい価値や方向性を示すことで、AIが必要なソフトウェアを開発し、運用状況を踏まえて継続的に改善する世界をめざします。また、AIモデルの最適な切替などによるAI利用コストの削減も組み合わせ、実用的かつ持続可能なAIと人の協働によるソフトウェ

ア開発の実現に向けて、研究開発を進めていきます。

参考文献

- (1) J. Yang, C. E. Jimenez, A. Wettig, et al.: "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering," *Advances in Neural Information Processing Systems*, Vol. 37, 2024.
- (2) The Linux Foundation: "Linux Foundation Launches the Tokenomics Foundation to Define the Economics and ROI of AI Value," Aug. 2026.
- (3) M. Alenezi: "Human-AI Collaboration and the Transformation of Software Engineering Work," 2026.



若林 慧

人とAIが協働するソフトウェア開発を実現するための研究開発を推進していきます。

◆問い合わせ先

NTTコンピュータ&データサイエンス研究所
革新的コンピューティングアーキテクチャ
研究プロジェクト