THE LINUX FOUNDATION | Research

Red Hat

FUTUREWEI Technologies

OpenSSF OPEN SOURCE SECURITY FOUNDATION

LF AI & DATA

生成AI時代におけるオープンソースソフトウェア開発の進め方

プロジェクトコミュニティの課題と機会



October 2026

Sam Boysel, *The Linux Foundation*

Anni Lai, *Generative AI Commons*

Foreword by Brian Proffitt, *Red Hat*

生成AI時代におけるオープンソースソフトウェア開発の進め方

大規模なAIによる貢献に備える

メンテナーの59%（大規模プロジェクトでは69%）が、AIによる貢献量が増加していると認識している。



質の低いPRをフィルタリングする



新規コントリビューションの4件に1件は、AIスロップです。大規模なプロジェクトでは、この割合は31%にまで上昇します。

AIを活用した貢献の優先順位付け

メンテナーの半数が、バグチェックやコードレビューにAIを活用しています。しかし、4人に1人はまったく利用していません。





レビューサイクルを効率化する

質の低い貢献と貢献数の増加が相まって、メンテナーのレビューに負担がかかっています。

スケーラブルな検証システムの構築

ポリシーとツールを活用することで、レビューの負担を貢献者に還元し、プルリクエストの品質を向上させ、メンテナーのバーンアウトのリスクを軽減することができます。



明確なAIガバナンスを確立する

AIの貢献を認める正式なポリシーを定めているプロジェクトは、3件に1件にとどまっています。



新しい貢献者へのメンタリングに投資しましょう

メンテナーの大半は、AIの導入によって貢献者への指導が「減る」のではなく「増える」必要があると述べています。



AIによる貢献の帰属を標準化しましょう

現在、貢献におけるAIの支援を常に開示しているのは10人中3人にとどまり、10人中2人は決して開示していません。



プレサブミッションテストの義務化

OSS開発者の5人中3人は、AIを活用した貢献内容をプレサブミッションテストで徹底的にテストしてから提出しています。



まずはAIガバナンスのリスクに対処しましょう

現在の障壁は技術的なものですが、AIの導入に消極的な企業にとっては、モデルの来歴、トレーニングにおける倫理、およびデータプライバシーの改善の方がより重要です。



OSS開発におけるAIの効率性を活用する

コーディング、レビュー、課題レポートの各工程で最大50%の時間短縮を実現します。メンテナーは、効率的なプロジェクト内の議論の恩恵を受けることができます。



Reinvest AIがOSSに還元



生成AIを活用することで、貢献者は技術的負債の解消、新機能の開発、スキルアップにより多くの時間を割くことができます。

Copyright © 2026 The Linux Foundation | October 2026. 本レポートは、 Creative Commons Attribution-NoDerivatives 4.0 International Public License.の下で提供されています。インフォグラフィックをご利用の際は、本調査レポートを引用してください。引用方法については、レポートの最終ページをご参照ください。

目次

序文 4

エグゼクティブサマリー 6

はじめに 7

生成がプロジェクトおよびメンテナーに与える影響 8

- 大規模プロジェクトほどレビュー時間増とバーンアウトリスクが大きい 8
- AIスロップがPRの約4件に1件に迫る 9
- コントリビューション量は大規模ほど急増 11
- メンテナーもAI対応にAIを活用 11
- 多くのプロジェクトはAI貢献ポリシーが未整備 12
- 検証はツールよりも信頼頼み 13
- AIは参入障壁を下げるがコミュニティの信頼を損ないうる 14

OSS開発における生成AIの使われ方 16

- 生成AIは貢献の障壁を下げ、プロジェクトのガードレールが歯止めに 16
- AIツールは標準化したが、モデルの出自への関心は依然高 18
- 大半はAIコードを入念にレビューするが、帰属表示はまちまち 20
- 非採用者は信頼の問題は解決可能、技術的問題は本質的と見る 22

生成AIによる生産性への影響 23

- 生成AIはOSS開発を全般的に高速化 23
- タスクが複雑になるほど時間短縮効果は縮小 25
- 浮いた時間は技術的負債と新機能へ 26

結論 27

調査方法 29

回答者属性 30

- OSS開発 30
- 生成AIの利用経験 34
- メンテナーの属性 35

著者について 36

謝辞 36



序文

過去30年間、私はオープンソースプロジェクトやコミュニティが、ウェブの台頭、携帯電話の登場、そしてそれに伴うビッグデータやクラウドネイティブコンピューティングの台頭など、次々と押し寄せる変革を吸収していく様子を見てきました。こうした一連の技術的波において、コミュニティは、自らをより強く、より効率的なものにする形で適応してきました。生成AIは、こうした波の中で最も新しいものであり、おそらくこれまでで最も大きな影響をもたらすものとなるでしょう。その違いは、この技術がどこに作用するかという点にあります。オープンソースは、コードと同様に、信頼、協働、そして分担された労力に常に依存してきました。したがって、AIのような技術が、貢献の書き方、提出方法、審査の仕方を変え始めると、それはまさにオープンソース・コミュニティの機能の核心に直撃することになります。

レッドハットにとって、私たちの信念は単純明快です。AIの未来はオープンであるべきだということです。AIは、単一のベンダーによって一方的に決定されるのではなく、オープンな環境で構築され、透明性をもって管理され、コミュニティによって形作られるべきです。ベンダー中立でコミュニティ主導のAIエコシステムこそが、人々が検証し、信頼し、改善できるモデルやツールを実現するための、唯一の持続可能な道なのです。

これまでの議論の多くは、単なる体験談に留まっていました。「AIスロップ」や、対応に追われるメンテナ、そして信頼の低下といった話が延々と語られてきたのです。AIとオープンソースに関する議論で欠けていたのは、厳密なデータでした。234人のコントリビューターとメンテナを対象とした「生成AIとオープンソースソフトウェア開発に関する調査」は、そうした印象を具体的な数値として明らかにしており、オープンソースコミュニティが今後の展開を見据えるためにまさに必要とされる実証的な研究なのです。

今回の調査結果のうち、2つの点は、単純な解釈には当てはまらないため、特に注目に値します。1つ目は、生成AIの影響がすべてのプロジェクトに等しく及ぶわけではないということです。大規模なプロジェクトと小規模なプロジェクトでは、その影響の受け方が根本的に異なります。大規模なプロジェクトでは、50人以上の貢献者がいるプロジェクトにおいて、コードの品質に代わってメンテナーのバーンアウトが最大の懸念事項となり、最優先の課題となっています。最大規模のプロジェクトでは、レビュー時間が最も長く、質の低い「AIスロップ」と呼ばれる貢献の割合が最も高く、貢献量の増加ペースも最も速いことが報告されています。つまり、規模が大きくなるほど、負担がより多くの手に分散されるのではなく、かえって増幅されてしまうのです。私たちのエコシステムにおいて最も注目され、重要なプロジェクトこそが、この生成AIの波による最大の打撃を受けており、それらは私たちが決してバーンアウトさせてはならないプロジェクトの一つなのです。



2つ目の興味深い発見は、この過熱した期待を冷ますもう一つの材料となり得るものです。それは、生成AIによる生産性向上の効果は、タスクが複雑になるにつれて低下するということです。調査によると、コード作成のような、明確に定義されたタスク指向の作業では、劇的な時間の節約が見られます。しかし、その効果は、調整や判断、あるいは結論が定まらない議論といった作業になると、着実に減少していきます。最も難易度の高いタスクについては、開発者のかなりの割合が、AIを使用しても以前より時間がかかると回答しており、時間が短縮されたとは感じていないことが明らかになりました。

AIツールの仕組みを理解している方にとっては、これは決して驚くべきことではないはずです。AIは定型業務を大幅に効率化する素晴らしいツールですが、あらゆるプロジェクトの中核をなす、批判的思考を要する困難な作業は、依然として人間の専門知識に依存しており、そのことはデータからも明らかです。データによると、節約された時間は、技術的負債の解消、新機能の開発、そして新しい貢献者の受け入れといった形で、プロジェクトに再投資されていることが示されています。

これらと、本調査のその他の結果とを総合すると、非常に理にかなった結論が導き出されます。それは、今まさに押し寄せつつある生成AIの波に対処しているコミュニティが、AIを全面的に禁止しているわけでも、熱心に導入しているわけでもないということです。コミュニティのメンバーは、オープンソースコミュニティの最も重要な要素である「信頼」と「品質」を失うことなく、AIの恩恵を享受できるようにするための、ポリシー、検証、メンターシップといったインフラを構築するという、より困難でありながらも価値の高い取り組みを行っているのです。

Sam Boysel氏とAnni Lai氏の丁寧なご尽力、Hilary Carter氏およびLinux Foundation Researchチームの皆様による本調査の厳密な実施、そしてご自身の経験を共有してくださったすべての参加者の皆様に、心より感謝申し上げます。本調査から浮かび上がる状況は、ますます二極化するメディアで飛び交うような「ユートピア」でも「黙示録」でもなく、むしろ、オープンソースコミュニティと生成AIが実際にどのような状況にあるのかを冷静に捉えた、より有益な視点です。本レポートがきっかけとなり、活発な議論が生まれることを楽しみにしております。

Brian Proffitt
Senior Principal Community Architect - Strategic Insights,
Red Hat Open Source and AI Program Office

エグゼクティブサマリー

生成AIはすでにオープンソースソフトウェア（OSS）の開発に多大な影響を与えており、AIを活用した貢献が爆発的に増加しています。これに伴い、コードの量や品質、そして時間的制約のあるメンテナーへの負担増といった懸念が急浮上しています。こうした問題のある傾向がある一方で、生産性の向上や、エコシステムへの新たな貢献者の参入といった潜在的なメリットも存在します。生成AIとOSSをめぐる議論の進展に確固たる実証的根拠を提供するため、私たちは234名のOSS貢献者およびメンテナを対象としたアンケート調査を開発、実施、分析しました。本研究では、生成AIがOSSのワークフローにおいて具体的にどのように活用されているか、それが貢献のダイナミクスをどのように変化させているか、そしてプロジェクトリーダーたちが新たな機会と浮上する課題が複雑に交錯する状況をどのように乗り切っているかを探求しています。

私たちの調査結果によると、生成AIがプロジェクトやワークフローに与える影響は、多くの場合、プロジェクトの規模に左右されることが示されています。現在、調整の取れたメンテナチームを抱える大規模なプロジェクトほど、管理上の負担が最も重くなっています。こうしたチームは貢献量の急増に直面しており、大規模プロジェクトでは、寄せられる貢献のうち最大31%が質の低い「AIスロップ」であると報告されています。その結果、これらのプロジェクトのメンテナーの64%が、レビュー期間の長期化を報告しており、バーンアウトを最大の懸念事項として挙げています。プロジェクトリーダーたちは、AIが貢献への障壁を効果的に下げるという点では概ね同意していますが、このアクセスのしやすさは、コミュニティの信頼の低下、新規貢献者に対するメンタリング需要の高まり、そしてAIによる貢献に関するポリシーや検証方法の極めて断片化された状況という代償を伴っています。

こうした課題があるにもかかわらず、生成AIはオープンソースエコシステム全体において、依然として生産性を飛躍的に高める強力なツールであり続けています。貢献者やメンテナーによると、これらのツールによりOSS開発に必要な時間が短縮されており、タスクによっては平均で19%から52%の時間短縮効果が得られているとのことです。

メンテナーは特に、非常に効率的な技術的な議論や提案のレビューの恩恵を受けており、作業時間を25%から40%削減できています。この削減分について、回答者の66%が技術的負債の解消に直接再投資しています。さらに、生成AIは開発者が馴染みのない技術を積極的に探求し、新しいプロジェクトに貢献する意欲を高めています。特に重要な点として、慎重な導入が行われていることを示す強力な証拠が見られます。貢献者の61%は、AIが生成したコードを提出する前に、詳細な行単位のレビューと完全な機能テストを実施しており、80%は、基盤となるソースモデルやトレーニングデータの来歴を理解することの重要性を強調しています。

OSS開発における生成AIの利用をめぐる根本的な課題を解決するためには、プロジェクトは事後対応型の管理から、意図的な貢献方針とワークフローの設計へと移行する必要があります。成功しているコミュニティでは、生成AIを禁止したり、生成AIによる結果を無批判に受け入れたりするのではなく、責任ある生成AIの利用を義務付け、生成AIの出典表示を標準化し、提出されるコードに対する期待事項を明確に定義することで、レビューの負担を貢献者に委ねる、明確かつ透明性の高いガイドラインを策定しています。同時に、メンテナーは、増大する作業負荷に対処するため、自らAIの活用を検討すべきです。つまり、「火事には火で対抗する」という手法で、貴重なリソースを取り戻すのです。極めて重要なのは、プロジェクトリーダーが、AIは参入障壁を下げ、技術の導入を加速させる一方で、人間の知識や指導に取って代わることはできないという事実を認識することです。コミュニティの健全性を長期的に維持するためには、品質を犠牲にしたり信頼を損なったりすることなく、新規参加者を温かく迎え入れる実践的な指導や貢献の規範への継続的な投資が不可欠です。



はじめに

生成AIがソフトウェア開発にもたらす変革的な影響は、いくら強調してもしすぎることはありません。機能的なコードを生成するために必要な人的リソースや時間を削減することで、これらの新しいツールは、初心者からベテランの開発者まで、あらゆる開発者のワークフローに大きな影響を与える可能性があります。しかし、こうした動きは現在、オープンソースソフトウェア（OSS）開発における長年の規範や慣行と対立しています。OSS開発においては、技術的なスキルと同様に、あるいはそれ以上に、コラボレーションにおける社会的側面が重要視されることが多いためです。AIを活用した貢献がOSSプロジェクトに殺到しているという報告を受け、コードの品質、メンテナーの作業負荷、そして貢献者コミュニティ内の信頼の低下に対する懸念が高まっています。プロジェクトコミュニティは、高品質なOSSを作り続けていく方法を模索しており、その多くが、AIを活用した開発が今後も定着していくことを認めています。

本稿では、単なる事例に基づく証拠のみに頼るのではなく、幅広いOSSプロジェクトのサンプルとメンテナーの視点を用いて、生成AIの影響を明らかにすることから始めます。まず、生成AIが貢献の量や質をどの程度変化させるか、それがプロジェクトの規模によってどのように異なるか、そしてそれがメンテナーの作業負荷にどのような影響を与えるかを明らかにすることが重要です。また、本調査では、メンテナーがAIによる貢献に関するポリシーをどのように策定し、AIを活用した貢献をどのように検証し、AIを貢献のレビュー支援にどのように活用しているかについても探求しています。

本調査の次の段階では、OSS開発者が生成AIツールをどのように活用しているか、またこれらのツールが貢献行動にどのような影響を与えているかを調査します。アンケートでは、回答者に対し、生成AIをどのような具体的なOSS開発タスクに活用しているか、どのようなプロジェクトに貢献しているか、そしてAIを活用した貢献内容をどのように検証し、帰属させているかについて尋ねています。また、一部の開発者が生成AIを利用しない理由や、将来的に導入を促す要因についても探っています。

最後に、生成AIがOSS開発の生産性に与える影響について検証します。本調査では、回答者に対し、生成AIを使用することでさまざまなタスクにおいてどれだけの時間を節約できるか、その効果はタスクや貢献者の役割によってどのように異なるか、そして節約した時間をどのように再活用しているかについて推定してもらいました。このセクションでは、生成AIがOSS開発のあり方をどのように変革しつつあるか、またそれがメンテナーや貢献者、さらにはOSSコミュニティ全体にとってどのような意味を持つのかについて、洞察を提供します。

生成がプロジェクトおよびメンテナーに与える影響

メンテナーは、生成AIによる貢献の押し寄せる波の受け手となっており、その体験はプロジェクトの規模によってほぼ完全に左右されます。多くの大規模プロジェクトでは、AIによって生成された質の低い貢献が大量に流入したことにより、多大な負担に直面しています。貢献量の増加と検証負担の増大が相まって、メンテナーの貴重な時間を奪い、バーンアウトのリスクを高めています。このセクションでは、生成AIがメンテナーの作業負荷や貢献の質に与える悪影響を裏付ける調査結果を提示します。これらの影響を、投稿量の増加、プロジェクトのAI投稿に関する方針、およびメンテナーがAI支援による投稿に対して用いる検証・レビュー手法と関連付けて分析します。

大規模プロジェクトほどレビュー時間増とバーンアウトリスクが大きい

生成AIがOSSのメンテナに与える全体的な影響を把握するため、まずはAIを活用した貢献がメンテナに抱かせる主な懸念事項を特定することから始めます。プロジェクトの規模は、メンテナがAIによる貢献の波をどのように受け止めるかにおいて重要な要素となるため、結果をプロジェクトの規模ごとに分類するように配慮しています。

メンテナーたちは、AIを活用した貢献における根本的な問題として、貢献の質の低下を挙げています。しかし、プロジェクトの規模が拡大するにつれて[1]、その懸念はさらに差し迫った問題によって増幅されています。すなわち、大規模なプロジェクト（貢献者が50人を超えるもの）では、メンテナーのバーンアウトが主要な問題となっているのです。これは、貢献量の増加が、何の問題もなく吸収されるのではなく、そのままレビュー担当者の疲労に直結していることを示唆しています（表1）。

負担の最も明確な兆候は「時間」です。 **AIを活用した貢献により、それらを評価しなければならないメンテナのレビュープロセスが、測定可能なほど長くなっています。**リードメンテナーと共同メンテナーが最も重い負担を負っており、レビュー時間の増加率が最も高くなっています（図1）。単独メンテナは、レビュー時間の短縮が「増加」と報告した割合に迫るか、それを上回る唯一の役割として際立っています。これは、彼らが通常、規模が小さく、より馴染みのある貢献者グループを管理していることに起因していると考えられます。プロジェクトの規模もこの影響を増幅させています。大規模プロジェクトでは、レビュー時間の増加割合が64%と、小規模（46%）や中規模（55%）のプロジェクトよりも著しく高くなっており、最大規模で注目度の高いプロジェクトが、追加された作業負荷の大部分を吸収していることが確認されています。

表1：大規模プロジェクトにおいて、メンテナーのバーンアウトが最大の懸念事項となってきています

懸念度ランク	小規模プロジェクト	中規模プロジェクト	大規模プロジェクト
第1位	投稿の質［51%］	投稿の質［50%］	***メンテナーのバーンアウト［63%］***
第2位	セキュリティ上の懸念［40%］	技術的負債の増加［35%］	投稿の質［60%］
第3位	寄与度［35%］	メンテナーのバーンアウト［35%］	寄与度［43%］

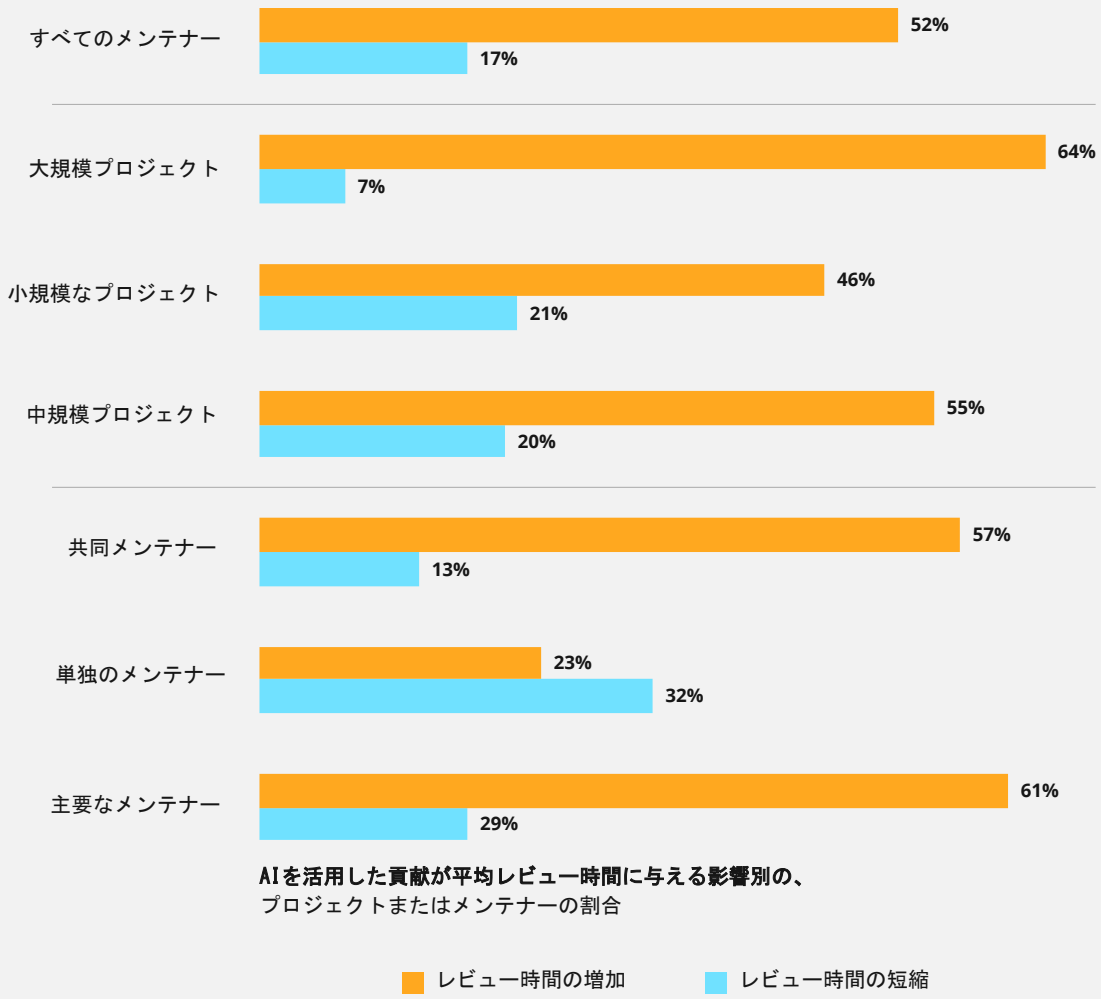
出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q36 × Q29。回答者数＝133名

1 本報告書では、プロジェクトの規模について以下の定義を採用しています。小規模プロジェクト＝貢献者が10人未満。中規模プロジェクト＝貢献者が11人から50人。大規模プロジェクト＝貢献者が50人を超える。

図1

AIを活用した貢献は、大規模なプロジェクトや主要なメンテナにおいて、レビューにかかる時間を延ばす可能性がはるかに高いです

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q39 × Q28 × Q29。回答者数＝174名。



AIスロップがPRの約4件に1件に迫る

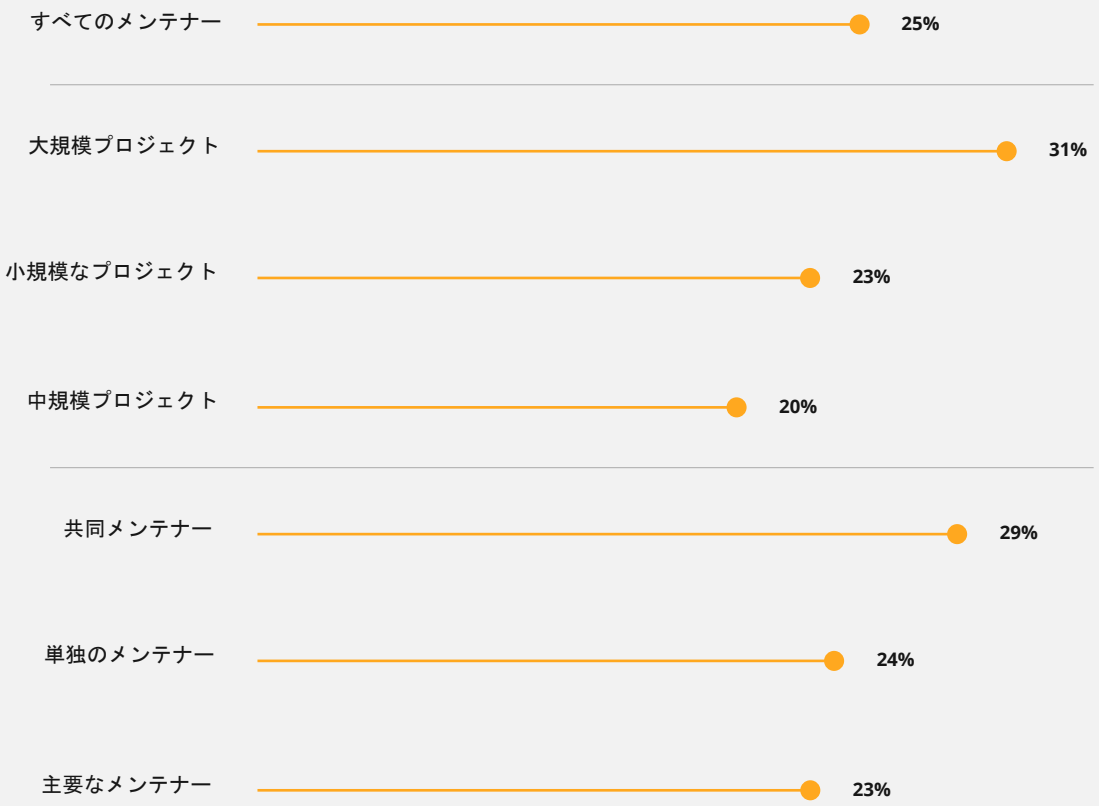
こうしたレビュー負担の増加の背景には、品質の問題があります。「AIスロップ」とは、AIによって生成された質の低い投稿のことで、レビューに費やす時間に見合う価値があるとは考えにくいものです。大規模なプロジェクトは、投稿活動が活発化する中心地であり、メンテナーの懸念が最も高まる場所でもあります。

平均して、**提出されるプルリクエスト（PR）の25%は、AIによって生成された質の低い貢献であると推定されています（図2）**。この負担もまた、プロジェクトの規模に比例して増大します。大規模プロジェクトでは、中小規模のプロジェクトに比べて、平均スロープシェアが著しく高くなっています（31%対23%および20%）。これは、レビュー担当者の数が増えることで負担が分散されるのではなく、むしろ規模が大きくなるほど、手抜きなAIによる提出物への曝露が増幅されることを裏付けています。また、プロジェクトの規模とは別に、役割も重要な要素となります。共同メンテナは、各役割の中で最も高い平均スロープシェア（29%）を報告しており、これはリードメンテナや単独メンテナよりも6パーセントポイント近く高い数値です。これは、単一のリードメンテナが通常提供できる直接的な監督よりも、より広範な貢献者ネットワークが存在し、直接的な監督が及ばないことを反映している可能性があります。

図 2

提出されるPRの4件に1件は、AIスロップです。共同メンテナチームを擁する大規模なプロジェクトでは、その割合が最も高くなっています。

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q38 × Q28 × Q29。回答者数＝111名。
上段のグラフは、全回答者のサンプル平均を示しています。中段のグラフは、プロジェクトの規模別にメンテナを分類したものです。下段のグラフは、メンテナの種類別に回答者を分類したものです。



受け付けたPRのうち、低品質および／またはAIによって生成されたものと推定されるものの平均割合

すべてのOSSプロジェクトが「スロップ」を受け取っているわけではなく、また、スロップを受け取り続けているわけでもありません。Curlプロジェクトのリーダーであるdaniel Stenberg氏は2025年、Curlプロジェクトが脆弱性レポートにおけるAIスロップに圧倒されていたと報告しました[2]y、しかし、2026年4月までに、同氏はCurlプロジェクトが「AIスロップによるセキュリティレポートを受け取らなくなった」と報告し、その代わりに「AIの助けを借りて作成された、非常に質の高いセキュリティレポートがますます増えている」と述べました[3]。 このことは、AIモデルの精度が向上し、各プロジェクトがAIスロープに対抗する取り組みを進めるにつれて、スロープは時間の経過とともに減少させることができることを示唆しています。ただし、各プロジェクトは、AIスロープは容認できないこと、またAIによって生成された成果物（脆弱性レポートなど）は、プロジェクトに提出する前に慎重に検証されなければならないことを明確にする必要があります。

2 Daniel Stenberg, “Death by a Thousand Slops,” Daniel Stenberg’s Blog, July 14, 2025, https://daniel.haxx.se/blog/2025/07/14/death-by-a-thousand-slops/.

3 Daniel Stenberg, LinkedIn post, April 2, 2026. https://www.linkedin.com/posts/danielstenberg_hackerone-share-7446667043380076545-RX9b/

コントリビューション量は大規模ほど急増

先ほど述べた作業負荷の重さと品質に関する懸念の根底には、規模における根本的な変化、すなわち寄せられる貢献の膨大な量があります。質の低い貢献の量が増加したことで、すでに過重な業務を抱えているメンテナーにとって、問題はさらに深刻化しています。

生成AIの台頭以来、ほとんどのプロジェクトで貢献量の増加が見られます。メンテナーの59%が増加をレポートしており、減少をレポートしたのはわずか4%にとどまっています（図3）。これは、貢献が完全に激減しているという説を否定する証拠である一方で、前述した作業負荷や品質面でのプレッシャーの根本的な要因でもあります。というのも、参加のハードルが下がることで、必ずしも質が向上するわけではないにもかかわらず、より多くの投稿が集まってしまうからです。ただし、その増加は均等に分布しているわけではありません。プロジェクトの規模に応じてその傾向に差が見られます。大規模プロジェクト（貢献者51名以上）の69%が増加をレポートしているのに対し、小規模プロジェクトでは57%、中規模プロジェクトでは52%となっています。大規模プロジェクトにおけるバーンアウトのリスクが高まっていることとも一致して、注目度の高いプロジェクトが、**AIを活用した新たな貢献のトラフィックの不釣り合いなほど大きな割合を吸収しています**。小規模プロジェクト（貢献者10人未満）の状況はより複雑で、単独のメンテナーと同様に、どのセグメントよりもばらつきが大きくなっています。40%が「全く変化なし」とレポートする一方で、21%は50%以上の増加をレポートしており、この二極化は、その規模において、たった1人の活発なAI支援型貢献者が数値をどれほど歪める可能性があるかを反映していると考えられます。

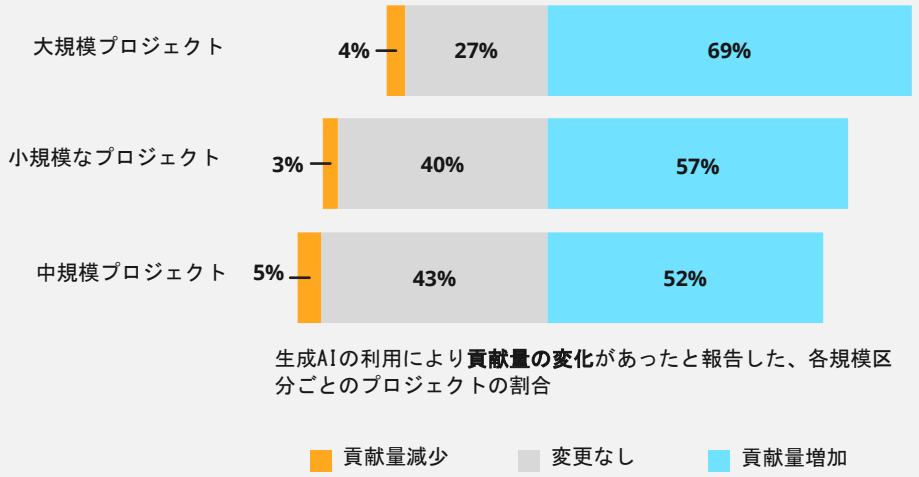
メンテナーもAI対応にAIを活用

メンテナーは、独自の手段を持たずにこうした増大する課題に立ち向かう必要はありません。中には、AIを自ら活用してメンテナンス作業の負担を軽減し、「火事には火で対抗する」ような対応を取っている人もいます。もちろん、AIがメンテナーの専門知識に取って代わるものではありませんが、熟練したオペレーターであれば、AIツールを活用することで、量も複雑さも増す一方の貢献提案の未処理案件を、より効率的に精査することができます。

図 3

AIを活用した貢献量は、大規模なプロジェクトほど最も大きく増加します

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q37、Q29（プロジェクト規模、小・中・大に集約）による分類。回答者数＝111名。



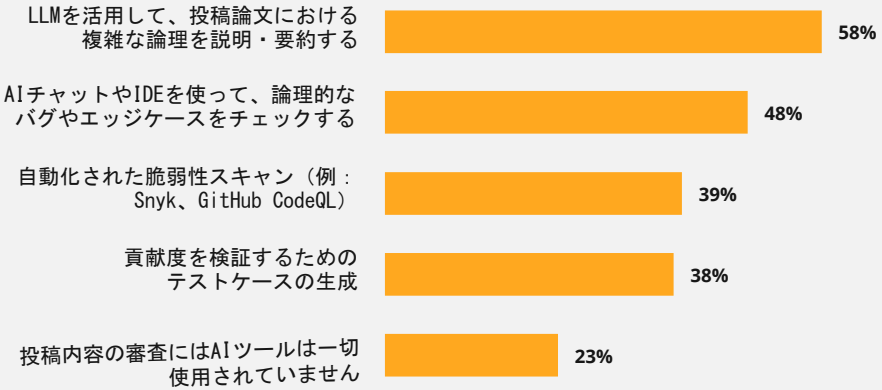
生成AIの利用により**貢献量の変化**があったと報告した、各規模区分ごとのプロジェクトの割合

ほとんどのメンテナーは、寄せられる貢献内容を理解するため［58%］や、論理的なバグをチェックするため［48%］にAIを活用しており、貢献量の増加を後押ししているのと同じ技術が、増加するレビュー負担に対する部分的な相殺要因となっています。とはいえ、23%は貢献内容の審査にAIツールを一切使用しておらず（図4）、かなりの割合のメンテナーが、完全に手作業による方法で増え続けるレビュー負担を自ら引き受けることを選択しています。

図 4

メンテナーの4人のうち3人が、AIを活用した貢献を管理するためにAIツールを積極的に活用しています

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q33. 回答者数＝92名。回答総数＝193件。

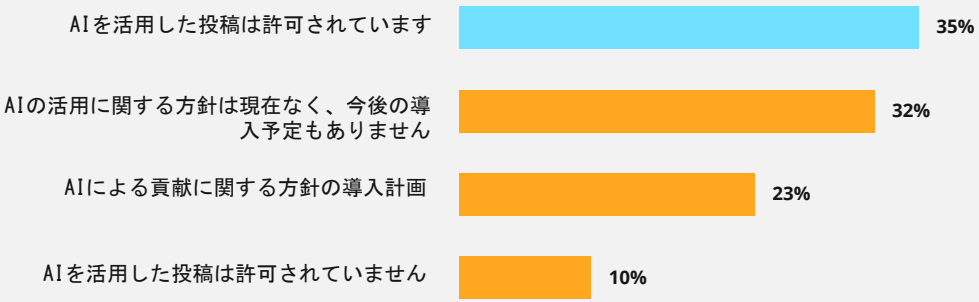


AIが生成した貢献を精査するためにこの方法を利用している回答者の割合

図 5

OSSプロジェクトのうち、AIによる貢献を認める正式なポリシーを定めているのは3分の1に過ぎません

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q30。回答者数＝119名。



AI貢献ポリシーの適用状況別の回答者の割合

多くのプロジェクトはAI貢献ポリシーが未整備

AIを活用した貢献の増加や品質の低下、その他の懸念を踏まえると、各プロジェクトが何らかの正式な回答を策定しているのかどうかという疑問が自然と浮かびます。貢献に関するポリシーは、全面的な禁止から責任ある利用のための詳細なガイドラインに至るまで、さまざまな形態や正式化のレベルをとることができます。さらに、AIを活用した貢献に対するポリシー上の回答はプロジェクトごとに異なり、特定の1つのアプローチが主流となっているわけではありません。

ほとんどのプロジェクトでは、生成AIによる貢献に関するポリシーがまだ標準化されていません。 **生成AIの使用を許可する正式なポリシーを定めているプロジェクトは35%にとどまり、32%はポリシーがなく、策定の予定もないほか、10%は明示的に禁止しています**（図5）。これは、大多数のプロジェクトが、AIを活用した貢献について、その都度、ケースバイケースで対応していることを示唆しています。しかし、正式なポリシーの策定に向けた動きが加速しつつあります。プロジェクトの23%が、将来的にAIによる貢献に関するポリシーを導入する計画を持っています。正式なAI利用ポリシーは、貢献者に対する明確な期待を示すとともに、メンテナーのレビューにかかる負担を軽減します。

ポリシーが存在する場合、その多くはAIの利用を全面的に制限するのではなく、その利用状況を可視化することに重点を置いている傾向があります。既存のポリシーでは、透明性や情報開示の要件［76%］および貢献者によるレビューの責任［76%］が最も多く取り上げられている一方で、ライセンスや著作権についてはわずか39%のケースでしか扱われていません（図6）。このギャップにより、非公式で信頼に基づく解決が最も難しい問題について、多くのプロジェクトが困惑を余儀なくされています。

生成AIに関するポリシーは、法的な不確実性の解消に役立ちます：多くの法域（EUや米国を含む）では、人間の創造的な関与を一切伴わずにAIによって生成された作品は、著作権の対象とはなりません。すでに著作権で保護されている作品の複製である場合は、なお著作権の対象となります。これを裏付ける判例としては、Bartz v. Anthropic PBC 事件や Kadrey v. Meta Platforms, Inc. 事件などが挙げられます。Linux Foundation は、これらの問題について論じた「オープンソースソフトウェア開発における生成AIツールの利用に関するガイダンス」[4] を公開しています。とはいえ、AI を使用する一般的なソフトウェア開発や脆弱性特定作業において、そのような複製が行われる可能性は比較的低いと考えられます。さらに、一部がライセンスの対象であり、他の部分が著作権の対象とならない作品であっても、作品全体としてはそのライセンスの対象となります。ほとんどの開発者は法律の専門家ではないため、この点を明確に示す方針は開発者にとって有益です。

ごく一部のプロジェクトでは、明示的な禁止措置が取られています。場合によっては、この禁止措置が極端な形をとって、セキュリティ脆弱性レポートを含むAI生成のレポートをすべて却下することさえあります。このような方針は、ユーザーを極めて大きなリスクにさらすことになります。 Michael Catanzaro氏が指摘するように、「レポートの一部がAIによって生成されたという理由だけで、良質な脆弱性レポートを排除することは容認できません。AIを活用した脆弱性レポートは、新たな業界標準となっています……問題報告を禁止することは、ソフトウェアの品質と安全性を低下させ、ユーザーに不利益をもたらすことになります」[5]

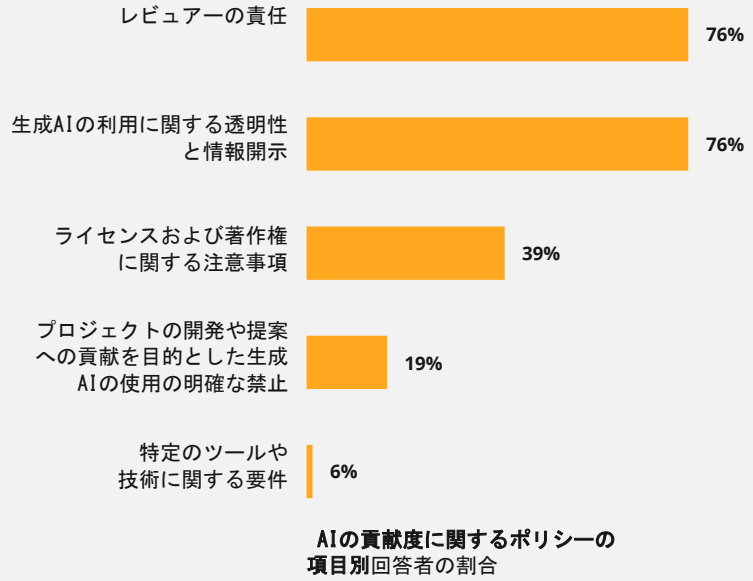
検証はツールよりも信頼頼み

詳細な貢献に関する方針が定まっていない状況下では、メンテナーはAIを活用した貢献の妥当性を審査・評価する上で、新たな課題に直面しています。メンテナーは当然のことながら、AIが生成したコードがコードベースに取り込まれる際、単なる技術的な品質だけでなく、ライセンス、著作権、データの出所といった問題についても懸念を抱いています。書面による方針が存在しない場合でも、AIを活用した貢献の検証と審査は、プロジェクトの品質と信頼を維持するための重要なステップとなります。

図 6

AIによる貢献に関するポリシーは、レビューの負担を貢献者に戻し、AIの使用開示を義務付けるものがほとんどです

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q31。回答者数＝54名。回答総数＝122件。



4 https://www.linuxfoundation.org/legal/generative-ai

5 Michael Catanzaro, “Please Do Not Ban AI-Assisted Issue Reports,” Michael Catanzaro’s Blog, June 8, 2026, https://blogs.gnome.org/mcatanzaro/2026/06/08/please-do-not-ban-ai-assisted-issue-reports/.

プロジェクトのメンテナーは、さまざまな検証戦略を組み合わせて採用しており、その結果を総合すると、こうした作業の大部分は**依然として形式的なプロセスではなく、非公式な判断に**依存していることが示唆されています。具体的には、検証は主に貢献者への信頼［47%］および目視による手動の審査［44%］に依存しており、26% は現在、検証メカニズムをまったく導入していない状況です（図7）。

ポリシーの採用と同様に、**検証の形式性もプロジェクトの規模に比例する傾向があります。**証明要件（貢献者に求められる明示的な声明や出所確認）は、プロジェクトの規模とともに急激に増加しており、小規模プロジェクトでは20%であるのに対し、大規模プロジェクトでは41%に達しています。これは、貢献者の数が多くなりすぎて信頼に基づくレビューが現実的ではなくなった時点で初めて、形式的な出所確認が標準となることを示唆しています。小規模プロジェクトでは「検証なし」の割合が最も高く（33%）、どの規模においても、信頼に基づく審査や手動による審査が依然として主流の仕組みとなっています。したがって、プロジェクトが拡大しても、問題のあるAI支援による貢献を検知するために、ほとんどのメンテナはツールではなく、自身の判断や人間関係を頼りにしているのが現状です。

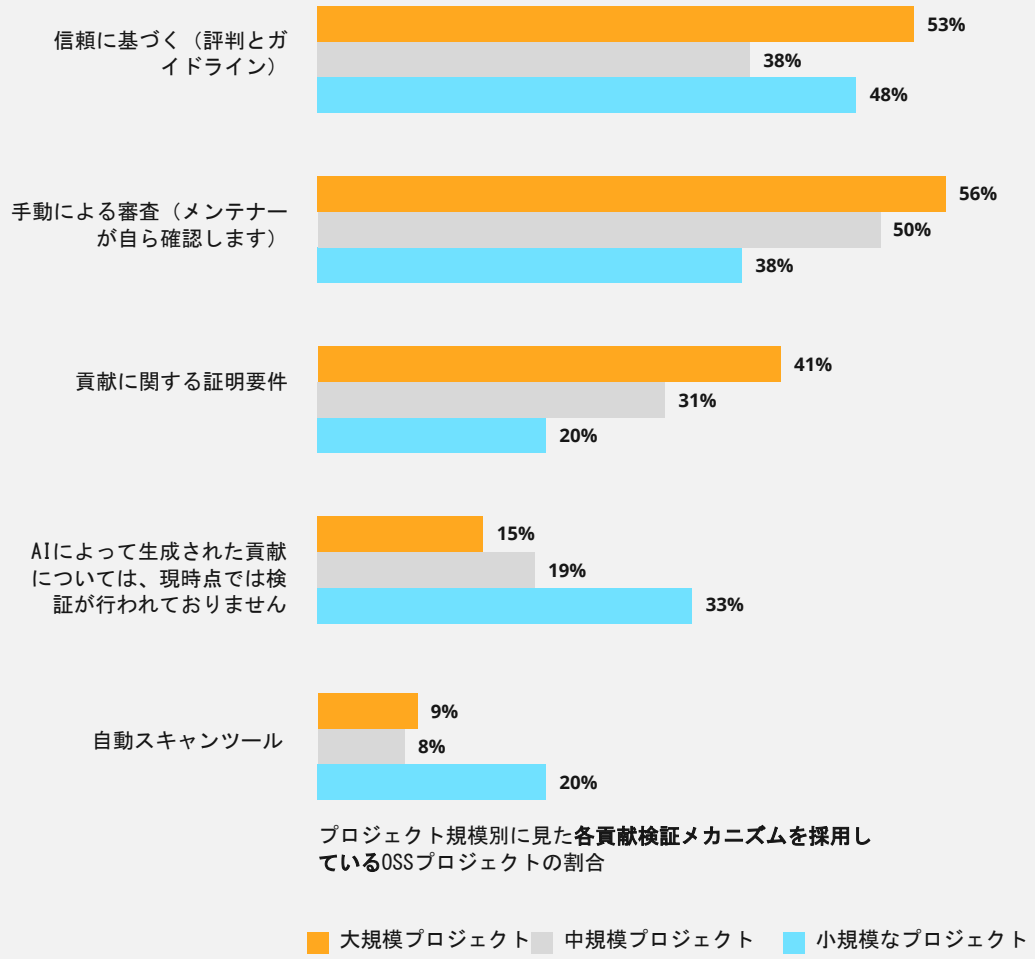
AIは参入障壁を下げるがコミュニティの信頼を損ないうる

また、本研究では、OSS開発における生成AIの利用が、プロジェクトやコミュニティの健全性に及ぼす広範な影響について、メンテナーの視点から検証しています。その結果、メンテナーは、自身のプロジェクト内における当面の作業負荷への影響に加え、コミュニティの信頼、新規貢献者への指導体制、そしてコミュニティの結束力に対する長期的な影響についても懸念を抱いていることがわかりました。

図 7

多くのプロジェクトでは、自動化されたプロセスではなく、AIを活用した貢献について、投稿者への信頼と手動によるレビューに依存しています

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q32 × Q29。回答者数＝131名。

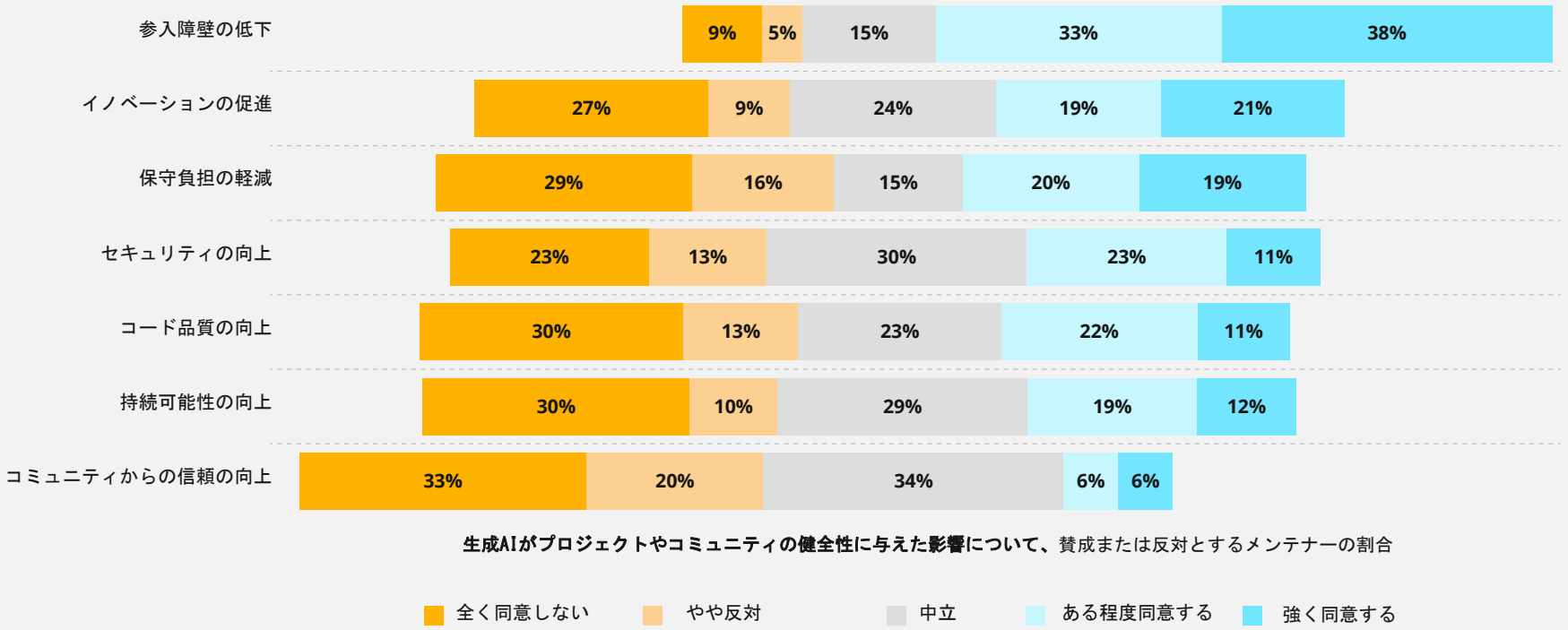


プロジェクト規模別に見た**各貢献検証メカニズムを採用しているOSS**プロジェクトの割合

図 8

メンテナンス担当者の間では、一般的に、生成AIは貢献のハードルを下げる一方で、その代償として持続可能性やコミュニティの信頼が損なわれるという点で意見が一致しています。

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q35。回答者数＝118名。



メンテナーの71%が、生成AIがOSSへの参入障壁を低下させるとの意見に同意しています（図8）。この強い合意は、OSSへの貢献における生成AIの重要な利点が、潜在的な貢献者の層を拡大し、新しいアイデアをもたらし、プロジェクト内の包摂性を促進することにあることを浮き彫りにしています。一方、**コミュニティの信頼は最も明らかな犠牲となっています。**：信頼が向上したとの意見に同意したのはわずか13%であるのに対し、53%が同意していません。参入障壁を下げるというアクセシビリティの向上は、貢献者コミュニティを結びつける信頼を直接的に犠牲にして成り立っているのです。

この緊張関係は、新規貢献者の受け入れ方法にも及んでいます。メンテナーの報告によると、**OSS開発における生成AIの活用は、新規貢献者に対するメンタリングの必要性を減少させる（32%）よりも、むしろ増加させる（46%）可能性が高いとのことです**（図9）。上記の参入障壁に関する調査結果と併せて考えると、AIはより多くの新規参加者を惹きつけている一方で、彼らが成功裏に貢献するためには、人間の指導への依存度を低下させるどころか、むしろ高めていることが示唆されます。これは、すでに増加する貢献量や長引くレビュー時間に対応している同じメンテナたちに、さらなるメンタリングの負担を課すことになります。

OSS開発における生成AIの使われ方

メンテナーの立場から、生成AIがプロジェクトをどのように変革しているかを検討してきましたので、ここでは貢献者の視点に目を向けます。具体的には、開発者が実際にこれらのツールをどのように活用しているか、どのツールを利用しているか、そして生成された出力をどの程度慎重に検証しているかについて見ていきます。また、生成AIを活用した開発を行っていない少数の回答者についても考察し、その選択の背景にある要因や、彼らが生成AIを利用するようになるためにどのような変化が必要かについても探っていきます。

生成AIは貢献の障壁を下げ、プロジェクトのガードレールが歯止めに

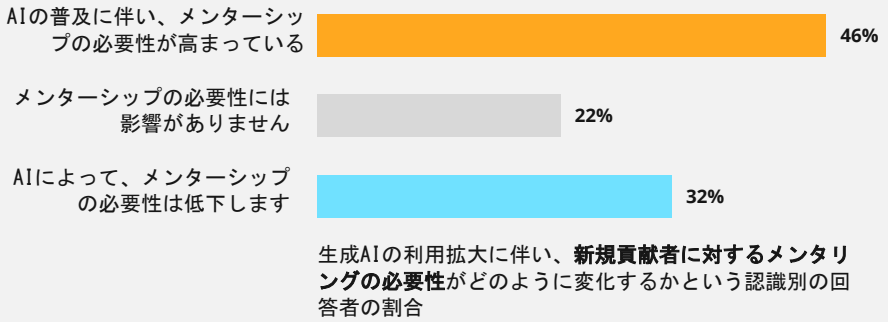
OSS開発において、生成AIの最も一般的な活用事例にはどのようなものがあるでしょうか。本調査では、貢献者が一般的なOSSタスクにおいてこれらのAI搭載ツールを具体的にどのように活用しているかを調査しており、その結果、開発ライフサイクル全体にわたって幅広い活用事例が明らかになりました。

OSS開発における生成AIの最も一般的な活用事例は、新機能の開発［83%］、デバッグ・テスト［78%］、リファクタリング［71%］、およびドキュメント作成［69%］です（図10）。**これは、OSSの貢献者が、生成AIツールをソフトウェア開発ライフサイクル（SDLC）のほぼすべての段階において汎用的な補完ツールとして捉えていることを示唆しています。**[6][7] さらに、徹底的なテストやドキュメント作成といった品質保証タスクにおいて、開発者に生成AIの利用を奨励することは、責任ある利用を促進し、OSSプロジェクトへの低品質な貢献が提出されるリスクを低減するのに役立つ可能性があります。

図 9

メンテナの皆さんにとっては、生成AIによって新規貢献者への指導の必要性が減るというよりは、むしろ増えると感じる可能性が高いでしょう

出典：生成AIとオープンソースソフトウェアの開発に関する調査。
Q34。回答者数＝114名。



生成AIの利用拡大に伴い、**新規貢献者に対するメンタリングの必要性**がどのように変化するかという認識別の回答者の割合

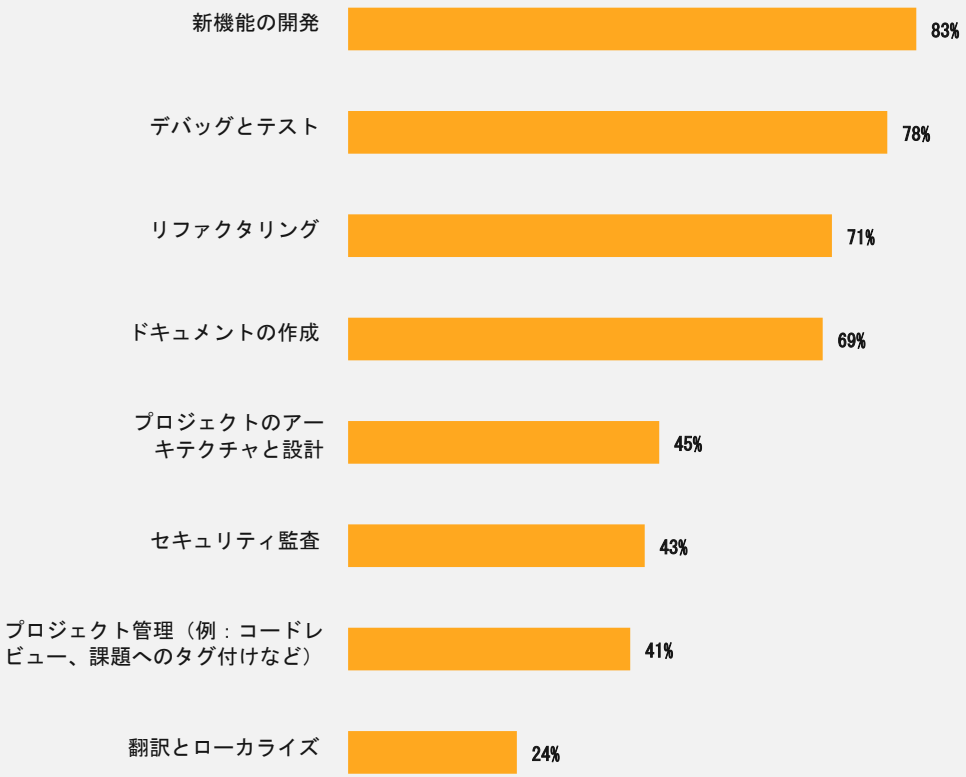
6 Addy Osmani, Shubham Saboo, and Sokratis Kartakis, “The New SDLC With Vibe Coding,” white paper, Kaggle, May 2026, https://www.kaggle.com/whitepaper-the-new-SDLC-with-vibe-coding.

7 Louis Claxton, “The AI-Native SDLC Playbook,” Claude Blog, August 21, 2026, https://claude.com/blog/the-ai-native-sdlc-playbook.

図 10

生成AIは、OSS開発のほぼすべての段階で活用されています

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q13。回答者数＝139名。回答総数＝644件。



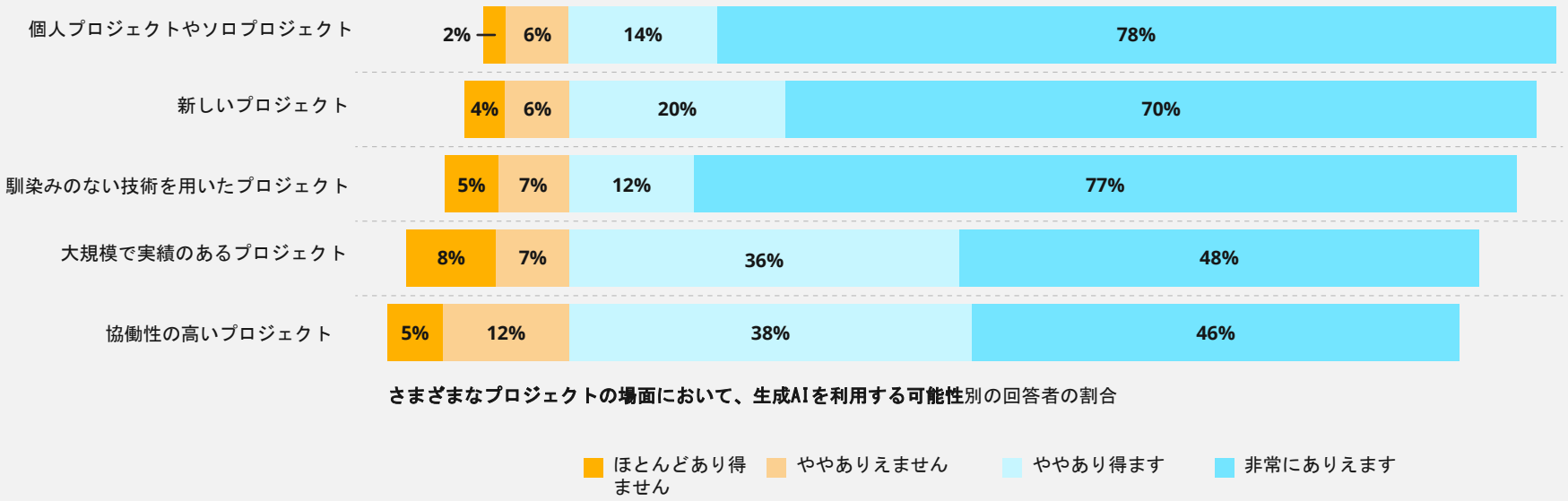
「OSSの貢献者たちは、生成AIツールを、ソフトウェア開発ライフサイクル（SDLC）のほぼすべての段階において汎用的な補完ツールとして捉えています。」

OSS開発における生成AIの利用意向は、プロジェクトの文脈に大きく左右されます。大規模で定評のあるプロジェクトでは、幅広い貢献者層、定着したコミュニティの規範、そして質の低い貢献を排除するための構造的な安全策が備わっています。こうしたコラボレーション上の摩擦は、新規の貢献者にとって苛立ちの原因となることもありますが、貢献のスピードと品質のバランスをとる上で重要な役割を果たしています。一方、新規プロジェクトや個人プロジェクトの貢献者は、制度的な障壁が少なく、厳格なガバナンスよりも迅速なプロトタイピングを優先します。その結果、生成AIの利用可能性は、個人プロジェクトやソロプロジェクト（78%が「非常に可能性が高い」）や、馴染みのない技術に関わるタスク（77%が「非常に可能性が高い」）で最も高く、大規模で確立されたプロジェクト（48%が「非常に可能性が高い」）に比べて、導入の可能性は著しく低くなっています（図11）。**OSS開発者は、共同作業における摩擦が少ない状況において、生成AIを利用する可能性が最も高くなります。** また、OSS開発において生成AIが特に有用な用途の一つが、馴染みのない技術であるという点も注目に値します。新規参入者は、新しい言語やフレームワークを使用することで、より容易に効果的な貢献を行うことができます。

図 11

コラボレーションの摩擦が少ない状況では、貢献者がジェネレーティブAIを利用する傾向が強くなります

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q14。回答者数＝128名。



こうした未知の領域に対する安心感は、具体的な行動の変化にもつながっています： **生成AIは、馴染みのないOSSプロジェクトへの貢献意欲を大幅に高め、**71%が「貢献する可能性が高まった」（43%が「大幅に」、28%が「やや」）と回答し、減少したと回答したのはわずか3%でした（図12）。genAIを利用するOSS開発者は、自分の「快適ゾーン」を超えて活動するよう後押しされています。事実上、genAIはオンボーディングを加速させる役割を果たしており、貢献者に対して、そうでなければ避けていたであろうコードベースや技術に踏み込むための十分な背景知識と自信を与えています。

AIツールは標準化したが、モデルの出自への関心は依然高い

生成AIツールには、オートコンプリート機能からチャットボット、エージェント型フレームワークなど、さまざまな形態があります。本調査では、回答者に対し、OSS開発にどのツールを使用しているか、使用頻度はどの程度か、そしてそれらのツールを支える基盤となるモデルに対してどの程度の精査を行っているかについて尋ねました。

図 12

生成AIのおかげで、開発者が馴染みのないプロジェクトに貢献する可能性がはるかに高くなります

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q17。回答者数＝136名。

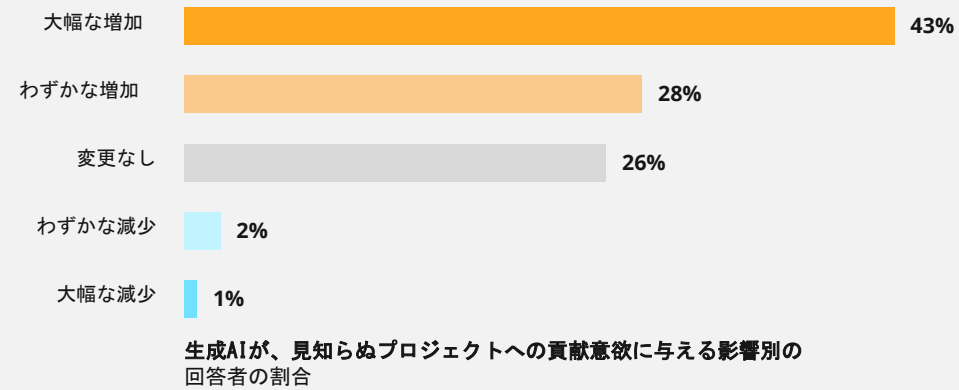
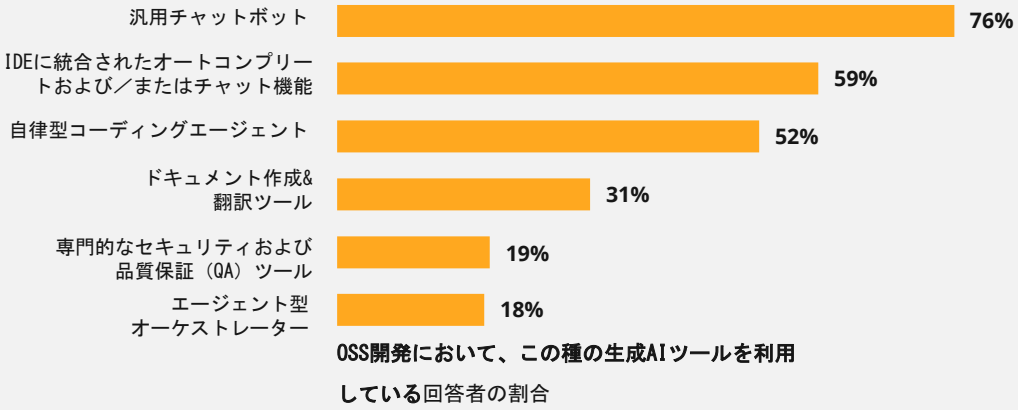


図 13

OSS開発において、チャットボットがオートコンプリートや自律型エージェントを牽引しています

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q15。回答者数＝139名。
回答総数＝367件。



OSS開発者が利用する生成AIツールとしては、汎用チャットボット［76%］が最も多く、次いでIDE統合型ツール［59%］、自律型コーディングエージェント［52%］が続いています（図13）。この利用状況から、より統合されたオプションがすでに広く普及しているにもかかわらず、**ほとんどの貢献者は依然として、生成AIをワークフローに組み込まれた自律的な要素としてではなく、主に会話の補助ツールとして利用していることがうかがえます。**

利用頻度もこの傾向を裏付けています。OSS開発者がジェネレーティブAIを利用する際、76%が対話型チャットを毎日、あるいはほぼ毎日利用しています（図14）。ジェネレーティブAIチャットボットは汎用的なツールであり、幅広いタスクにおいてソフトウェア開発者自身の専門知識を補完する役割を果たしています。開発者は依然として、コード補完（65%）やエージェント型フレームワーク（53%）といったより高度なツールを頻繁に利用していますが、平均的にはやや利用頻度が低くなっており、これは**生成AIツールの導入曲線が、開発者が手放す制御の度合いと反比例していることを示しています。**

こうした懸念は、エージェント型ツールにとどまらず、貢献者がモデルそのものをどのように捉えているかという点にも及んでいます。 **OSS貢献者の80%が、生成AIコーディングツールのソースモデルやトレーニングデータの来歴を把握することが重要であると考えています**（44%が「非常に重要」、37%が「ある程度重要」と回答）（図15）。生成AIツールを日常的に利用している貢献者であっても、来歴や信頼性に関する懸念が最優先事項となっています。

図 14

コントリビューターは、毎日ジェネレーティブAIを活用しています

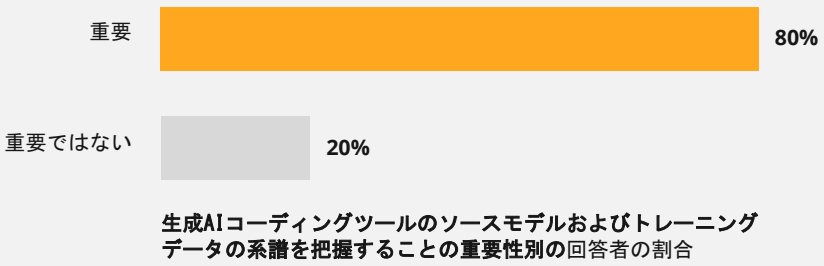
出典：生成AIとオープンソースソフトウェア開発に関する調査。Q12。回答者数＝139名。



図 15

OSSの貢献者の5人中4人が、AIモデルの来歴を把握することが重要だと考えています

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q18。回答者数＝126名。



大半はAIコードを入念にレビューするが、帰属表示はまちまち

「AIスロップ」に関する主な懸念点は、コントリビューターが、自身のパッチがアップストリームのコードベースと互換性があり、コードベースの標準に準拠しているかどうかを独自に検証する努力を十分に払っていないことです。これにより、すでに時間的制約に直面しているメンテナの負担がさらに増大してしまいます。多くの場合、AIの支援を受けて生成されたコントリビューションであることは明確に識別できますが、さらに根本的な問題として、コントリビューターが、コントリビューションのプルリクエストやコミットメッセージにおいて、AIツールを使用したことを明記していない可能性があるという点があります。貢献者による十分な注意と透明性は、増加する貢献量をメンテナが効率的に処理する上で大いに役立ちます。

私たちは、貢献者の方々に、OSSプロジェクトに提出する前に、AIを活用した貢献内容をどの程度検証し、出典を明記しているかについて尋ねました。心強いことに、安易な対応ではなく、慎重な姿勢が主流となっています： **61%の方が、提出前にAIが生成したコードについて、行単位での詳細なレビューと完全な機能テストを実施しており、最小限のレビューしか行わない、あるいは全く行わないのはわずか5%にとどまっています**（図16）。これは、ほとんどの貢献者が単にモデルの出力を転送しているのではなく、それを依然として自身の精査を必要とする草案として扱っていることを示す証拠です。

しかしながら、**情報開示の慣行は、審査の厳格さに比べて一貫性に欠けています。**出典明記の慣行にはばらつきが見られます。「ほとんどの場合」または「常に」AIを活用した作品であることを明記しているのは52%である一方、決して明記しないという回答も19%に上ります（図17）。前述したように、このテーマに関するポリシーの網羅性が不十分であることと相まって、この二極化は、出典明記が現在、**プロジェクトのポリシーではなく個人の規範に依存している**分野であることを示唆しており、当然ながら結果にばらつきが生じています。

生成AIによる貢献の帰属表示に関する提案

この問題の一因は、帰属表示に関する、広く採用されている単一の機械可読な規約が存在しないことにあります。特に、こうした提案が通常行われるコミットメッセージにおいては、その傾向が顕著です。Apache Software Foundation（ASF）は「Generated-by:」の使用を推奨しています。[8] しかし、コードの一部が人間によって作成または編集された場合、この表記は誤解を招く恐れがあります。また、「Co-authored-by:」を使用する例もありますが、ほとんどの法域においてAIが生成したコードは著作権の対象とならないため、この表記も誤解を招く恐れがあります。

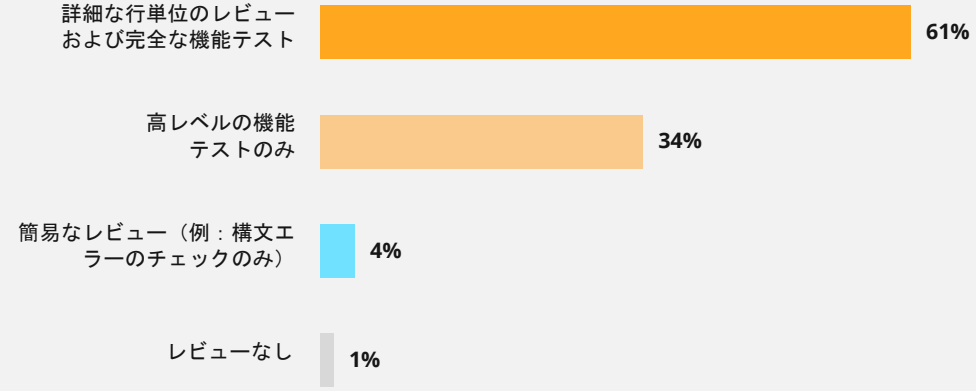
Linuxカーネルプロジェクトでは、コミット時に「Assisted-by: < agent >:< model_version > [tool1] [tool2]」という形式で出典を明記することが求められています。ここで、ツールとは専用ツールを指します。例えば、「Assisted-by: Claude:claude-4.5-sonnet coccinelle sparse」といった形です。この形式により、誤解を招く表現を用いることなく、専用ツールを含む詳細な出典情報を提供することができます。詳細については、 https://docs.kernel.org/process/coding-assistants.html をご覧ください。

8 https://www.apache.org/legal/generative-tooling.html

図 16

開発者の5人のうち3人は、AIによる貢献結果を投稿する前に徹底的にテストしています

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q20。回答者数＝137名。

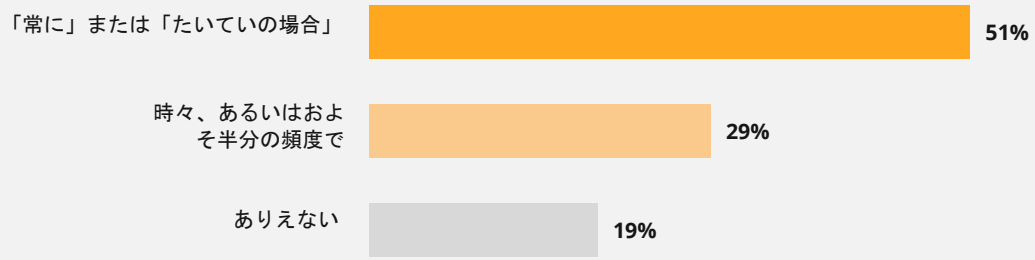


プロジェクトへの提出前にAIによる貢献内容の精査をどの程度徹底しているかによる回答者の割合

図 17

寄稿者の半数は、AIによる支援があった場合は必ずその旨を明記しています。5人に1人は、決して明記しません。

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q19。回答者数＝134名。



自身の貢献において生成AIによる支援を明示的に記載している回答者の割合（**記載頻度別**）

非採用者は信頼の問題は解決可能、技術的問題は本質的と見る

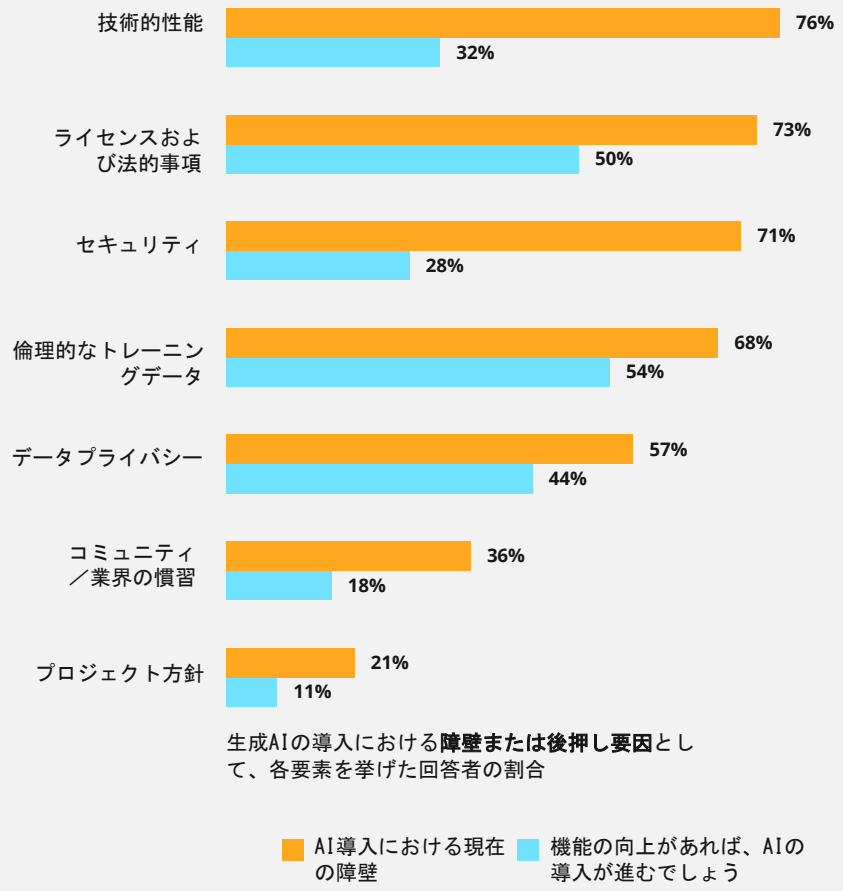
すべての貢献者が生成AIを受け入れているわけではありませんが、その理由を理解することよりも、導入状況そのものを追跡することの方が重要です。この調査では、生成AIを導入していない人々に、使用を避けている理由や、どのような要因があれば考えを変える可能性があるかを特定するよう求めています。

図18では、非導入者がOSS開発にジェネレーティブAIの利用を検討するようになるために、どのような要因が現在の障壁となっているか、またどのような要因の改善が必要となるかを比較しています。データのトレーニングに関する倫理やプライバシーを含む信頼性とガバナンスに関する懸念は、障壁として挙げられる頻度とほぼ同じ割合（約4分の3の確率）で説得材料へと転換されますが、技術的なパフォーマンスやセキュリティについては、その転換率が半分以下にとどまっています。言い換えれば、**非採用者は、信頼性やセキュリティを、信頼やガバナンスよりも重要だと必ずしも考えてはいません。** これらを挙げる頻度は同じ程度ですが、 **解決がより困難であると考えており、** これらの面での改善が実際に自分の考えを変えるだろうという点については、はるかに懐疑的です。「個人的な哲学」[68%]には、これと比肩し得る説得要因が全く存在せず、25%は「上記のいずれでもない」ことが自身の考えを変える要因になると回答しています。これらを合わせると、透明性やセキュリティ、パフォーマンスをいくら向上させても、おそらく説得できない堅固な非採用者層を形成しています。

図 18

生成AIの信頼性とガバナンスの向上は、導入していない企業にとって最優先課題と見なされています

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q26 × Q27。回答者数＝96名。



生成AIによる生産性への影響

投稿量や導入動向に加え、生成AIがOSS活動にもたらす価値を測る最も直接的な方法は「時間」です。つまり、これらのツールによってどれだけの時間が節約されるか、誰が最も恩恵を受けるか、そしてその節約された時間がどのように活用されるか、ということです。

生成AIはOSS開発を全般的に高速化

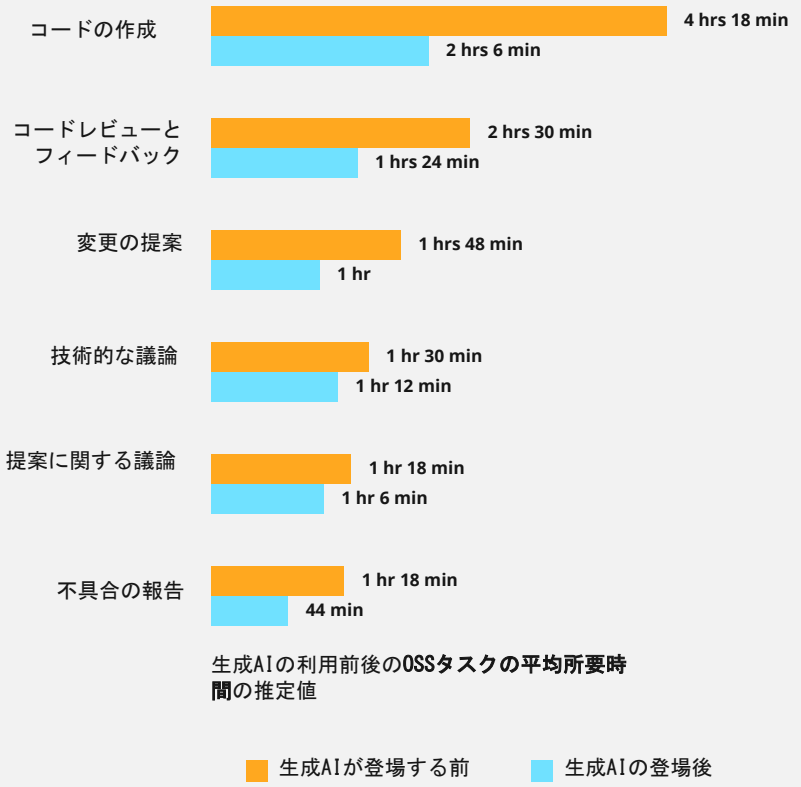
これまで見てきたように、生成AIツールはソフトウェア開発ワークフローのほぼすべての段階で活用されています。これらのツールは、コードの記述を加速させるだけでなく、複雑なコードベースの整理、問題のデバッグ、脆弱性の特定、アーキテクチャに関する指針の提供にも活用されています。しかし、生成AIによる生産性向上の効果は、さまざまなOSS開発タスクに具体的にどのように波及しているのでしょうか？OSSのコラボレーションのあり方は、純粋に技術的なものだけではありません。多くのやり取りには、貢献提案、バグや課題の報告、技術的な議論といった社会的交流が求められます。そこで、生成AIを利用している調査回答者に対し、生成AIの利用開始前と利用開始後において、典型的なOSS開発タスクに平均してどれくらいの時間がかかるかを尋ねました。

全体として、**生成AIは統合サンプルにおける測定対象の6つの活動すべてにおいて、**推定作業時間を短縮しており、その効果はコード作成（-52%）とコードレビュー・フィードバック（-44%）で最も大きく、技術的な議論（-20%）や提案に関する議論（-19%）では、その効果は小さいものの依然として有意な短縮が見られました（図19）。この傾向は、生成AIの強みが、明確に定義された機械的なタスクにおいて最も顕著であり、自由度が高く判断を要するコミュニケーションにおいては最も弱いという特徴と一致しています。

図 19

生成AIは、OSSタスク全般にわたる平均生産性を向上させます

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q23。回答者数＝96名。



オープンソース開発の役割別に削減効果を分析すると、2つの異なる傾向が明らかになります（図20）。 **メンテナーは、特に議論を要するタスクにおいて、はるかに大きな時間短縮効果を実感しています。**具体的には、提案に関する議論（−26% 対 −7%）や技術的な議論（−25% 対 −12%）です。これは、メンテナーのレビューやコミュニケーションの作業負荷がより重いという実情と一致しており、genAIがまさにプロジェクト規模に比例して増大する調整の負担を軽減していることを示唆しています。対照的に、コントリビューターは、時間の多くが費やされているコード作成（−58% 対 −49%）において最も大きな恩恵を受けています。つまり、合計で−52%という数値は、役割間で均等に分かれたものではなく、主にコントリビューターによるものであると言えます。

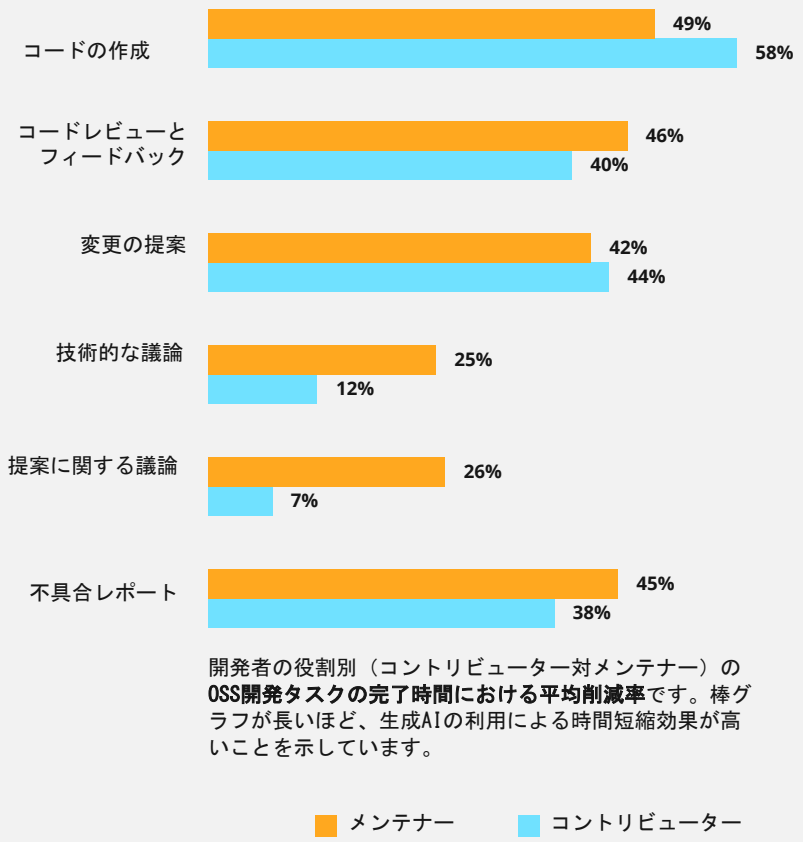
「本レポートで明らかになったデータは、エコシステムでこれまで目にしてきた現象を数値で裏付けています。生成AIは開発者の生産性を向上させますが、オープンソースプロジェクトを支える相互の信頼やプロセスに負担をかけています。このギャップを埋めることは、どのメンテナーも一人では成し遂げられません。そこにはプロジェクト横断的な連携が必要であり、まさにLF AI & DataやLinux Foundationが存在する意義は、そうした連携を提供することにあります。」

MARK COLLIER GM OF AI & INFRASTRUCTURE, LINUX FOUNDATION

図 20

生成AIは、プロジェクトの議論において、特にメンテナーにとって有益です

出典：生成AIとオープンソースソフトウェア開発に関する調。Q2、Q23。
回答者数＝96名。

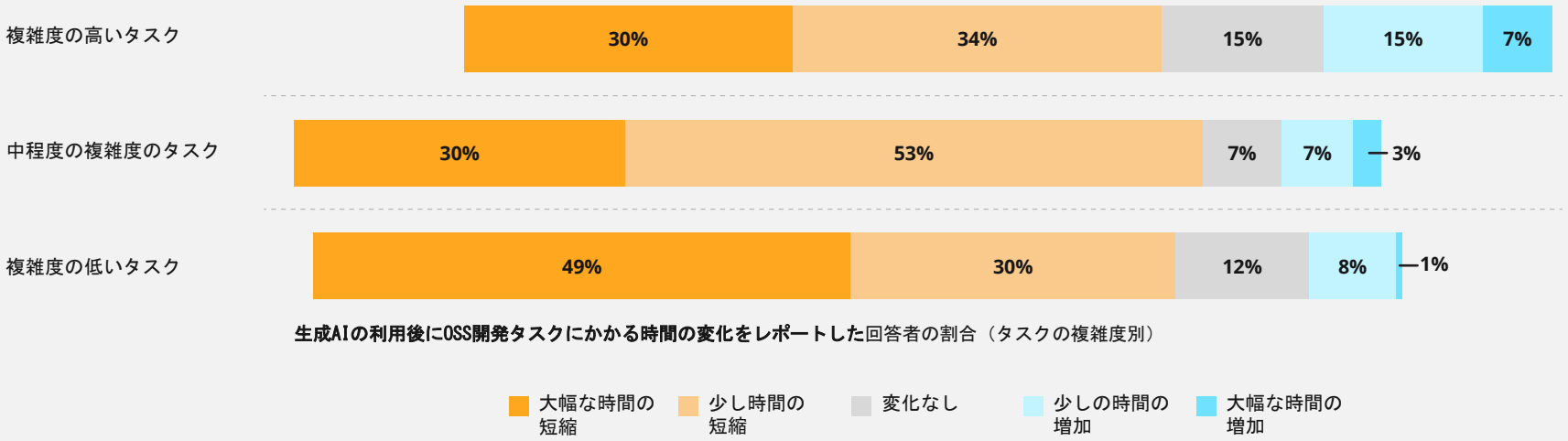


開発者の役割別（コントリビューター対メンテナー）の**OSS開発タスクの完了時間における平均削減率**です。棒グラフが長いほど、生成AIの利用による時間短縮効果が高いことを示しています。

図 21

生成AIは平均的には効率を向上させますが、より複雑なタスクにおいては必ずしもそうとは限りません。s

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q24。回答者数＝136名。



タスクが複雑になるほど時間短縮効果は縮小

時間短縮効果は、複雑さのレベルによって一様ではありません。生成AIは、複雑度の低いタスク［79%の時間短縮］および中程度のタスク［83%の時間短縮］において、最も確実に時間を節約します（図21）。一方、複雑度の高いタスクでは結果にばらつきが見られます。64%のケースでは依然として時間が節約されていますが、21%のケースでは以前よりも時間がかかると報告されています。タスクの難易度が上がるにつれて、**生成AIによる生産性向上の効果は縮小し、場合によっては逆転することさえあります。**

AIは複雑性の低いタスクにおいてより効果的であるという知見は、他の研究結果とも一致しています。Kwaら（2025）は、AIシステムを「50%タスク完了時間軸」（AIモデルが50%の成功率で完了できるタスクを、人間が完了するのに通常かかる時間）として測定することを提案しています。同研究によると、2025年時点で最先端のAIモデルの50%タスク完了時間軸は約50分であることが判明しました。しかし、AIシステムが今後改善されないと想定するのは賢明ではありません。同研究では、最先端AIの時間軸が2019年以降、およそ7ヶ月ごとに倍増していることも明らかになりました。[9]

9 Thomas Kwa et al., “Measuring AI Ability to Complete Long Software Tasks,” preprint, submitted March 18, 2025, https://arxiv.org/abs/2503.14499.

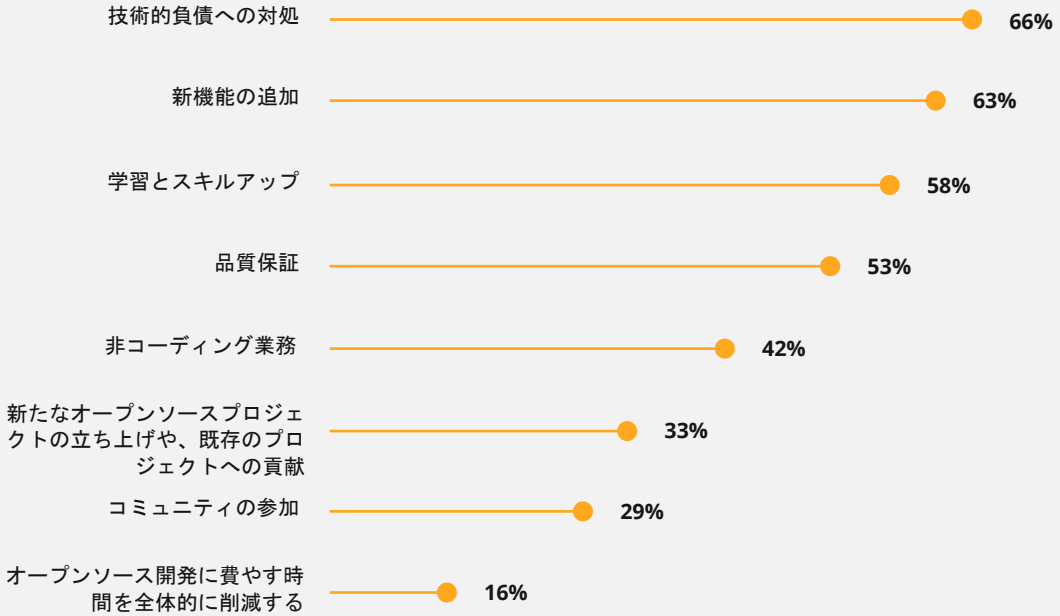
浮いた時間は技術的負債と新機能へ

単に時間が節約されただけでは、それほど大きな意味を持ちませんが、実際にはその時間が明らかに別の用途に振り向けられています。**生産性の向上による利益は、技術的負債の解消［66%］、新機能の追加［63%］、学習・スキルアップ［58%］に再投資されるケースが最も多いです**（図22）。OSS開発以外でこの時間を活用する可能性ははるかに低いため［16%］、**生成AIによる生産性の向上分は、そのままOSSプロジェクトに再投資されています。言い換えれば、生成AIはOSSの全体的な作業負荷を縮小させているというよりは、技術的負債の解消からスキル向上に至るまで、より付加価値が高く、これまで先送りされてきた作業へと、その負荷を再配分しているのです。**

図 22

OSS開発者は、生成AIによる生産性向上の成果を、プロジェクトや人的資本に再投資しています

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q25. 回答者数＝139名。回答総数＝513件。



OSS開発における**生成AIの活用によって得られた生産性向上の成果を、どのように再投資**したかによる回答者の割合



結論

生成AIが一般的なソフトウェア開発に与える影響は、いくら強調しても過言ではありません。OSSコミュニティは、その影響を直接かつ大規模に体験しています。OSSはコードと同様に、社会的信頼、コラボレーションの規範、そして共有されたレビュー作業にも大きく依存しているため、AIを活用した貢献が急増することは、他の開発環境に比べて、これらのプロジェクトの機能の核心に深く関わるものとなります。生成AIは、ベストプラクティスやコラボレーションのマナー、長年にわたるコミュニティの規範に代わるものではありません。これは生産性を向上させることができるツールですが、同時に、管理しなければならない新たな課題やリスクももたらします。貢献の量や質、そしてメンテナーのバーンアウトに対する懸念の高まりは、抽象的なものでも些細なものでもありません。これらは、OSSが依存している貢献者の労力の持続可能性に直結する問題なのです。オープンソースを健全な状態に保つためには、コミュニティやプロジェクトリーダーは、生成AIが貢献の力学をどのように変えているかを理解し、この技術がソフトウェア開発にさらに深く統合されていく中で、最適な適応戦略を見極める必要があります。

これまで、生成AIがOSSに与える影響に関する公の議論の多くは、単なる事例報告にとどまっていました。注目を集めた事例としては、「AIスロップ」の氾濫、レビュー待ちの蓄積、メンテナーのバーンアウトの増加などが挙げられていました。本調査では、OSSコミュニティの幅広いサンプルに基づく直接的な証拠を用いて、こうした懸念の妥当性を検証しています。個人プロジェクトから数百人の貢献者がいる組織に至るまで、 生成AIの導入組織と非導入組織、そしてあらゆる主要なガバナンスモデルを網羅した234名のOSS貢献者およびメンテナーを対象に調査を行った結果、どの事例に基づく懸念が大規模な状況でも当てはまるか（レビュー時間の増加、品質の低下、大規模プロジェクトへの不均衡な負担）、また、生産性の向上、導入パターン、そしてそうしたリスクを相殺する、概して良心的な検証慣行など、最も声高に語られている話よりも実際にはより微妙な側面を持つ懸念がどれであるかを確認することができました。これまでの議論は印象論にとどまっていましたが、本レポートでは実測データに基づいた分析を提供しています。

本調査では、OSSコミュニティやプロジェクトリーダーが生成AIへの移行を進める上で検討すべき、具体的な提言をいくつか提示しています：

- AIを活用した貢献や脆弱性の報告について、非公式なその場限りの慣行に任せるのではなく、明確な書面によるポリシーを策定してください。
- そのポリシーにおいて、透明性と情報開示の要件を優先させてください。貢献者は、どの生成AIツールを使用したかについて、メンテナに対して明確かつ一貫した情報を提供する必要があります。
- 提出前に、AIを活用した変更点について自己申告、テスト、および文書化を義務付けることで、レビューの負担を貢献者に移してください。
- AIによる貢献の帰属表示の慣行を標準化し、開示が個人の裁量に委ねられたままにならないようにします。一つの解決策として、コミットメッセージにおいてLinuxカーネルの「Assisted-by:」形式を採用し、その使用を義務付けることが挙げられます。
- AIをメンテナンス資産として活用します。AIツールを使用することで、寄せられた貢献に含まれる課題の選別、要約、フラグ付けを行い、レビューの負担を軽減し、メンテナーのバーンアウトを軽減することができます。
- 新規コントリビューターに対する人間によるメンタリングへの投資を継続しましょう。生成AIは参入障壁を下げますが、コントリビューターが参加した後の指導の必要性は、減るどころかむしろ増すことになります。
- 生成AIによるアクセシビリティの向上と、コミュニティの信頼に対するコストを天秤にかけ、品質や信頼を犠牲にすることなく新規参加者を歓迎する貢献規範を設計しましょう。
- AIに懐疑的な人々にとっては、技術的なパフォーマンスの向上よりも、質の高い生成AIのガバナンスと信頼の方が重要です。

本研究はあくまで初期段階の考察であり、決定的な結論ではありません。今後の研究[10]では、AIの貢献に関するポリシーや検証ツールが成熟するにつれて、こうした動向を縦断的に追跡するとともに、技術分野やガバナンスモデルによって結果がどのように異なるかをより深く掘り下げ、また、生成AIをまったく導入していないOSS開発者（全体の約5人に1人）にもっと注目すべきです。すでに導入している開発者を追跡することと同様に、彼らが考えを変えるには何が必要かを理解することも重要です。

生成AIがソフトウェア開発ライフサイクル（SDLC）全般に変革をもたらし続けている中[11]、OSS開発のパラダイムもまた適応していくことでしょう。生成AIを望まないコミュニティに無理に押し付けるべきではありませんが、この移行を最も成功裏に管理しているプロジェクトやコミュニティは、生成AIを禁止することもなければ、無批判に採用することもありません。彼らは、オープンソースが常に真の競争優位性としてきた信頼と品質を犠牲にすることなく、生成AIのメリット——迅速なオンボーディング、無駄な作業の削減、そして再投資された時間——を享受できるよう、ポリシー、検証、およびメンターシップのインフラを構築しているのです。

10 Related to this work, LF Research has studied how open source program offices (OSPOs) are preparing for the rise of genAI in OSS development. See **https://www.linuxfoundation.org/research/ospo-2025** for additional insights.

11 **https://claude.com/blog/the-ai-native-sdlc-playbook**



調査方法

データ収集方法	ウェブアンケート
調査期間	45問
調査対象者	オープンソースソフトウェアの開発において、生成AIを使うことができるOSSの貢献者やプロジェクトのメンテナ
アンケートの配布状況	Linux Foundationのニュースレター購読者、Linux Foundationのソーシャルメディアチャンネル、研究パートナーのメール／ソーシャルメディアチャンネル（Red Hat、OpenSSF、Kaiyuanshe、GOSIM、LF AI & Data）
調査実施期間	2026年2月～4月
調査の回収期間	2026年4月～6月
分析対象のサンプルサイズ	234名の有効回答者
誤差範囲（信頼水準95%）	±6.41%
データへのアクセス	https://zenodo.org/communities/linux-foundation-research
研究プロセスにおけるAIの活用に関する声明	LF Researchでは、二次資料の集約、データの統合、レポート作成の支援、校正、翻訳など、調査関連の業務において、生成AIやAIを活用した技術をさまざまな程度で活用しています。AIによって生成されたコンテンツはすべて、必要に応じて確認・修正が行われます。

回答者属性

OSS開発

回答者たちは、OSS活動において深く多様な経験を持っています。回答者の多くは、両方の役割を担っています。調査対象となった234人のコントリビューターおよび／またはメンテナーのうち、206人がコントリビューター、134人がメンテナーであると回答しています（図23）。この大きな重複は、メンテナーが自身のプロジェクトや他のプロジェクトにおいて、引き続き活発なコントリビューターであり続けることがいかに一般的であるかを反映しています。

参加の主な動機は個人的な関心です。貢献者の81%は個人的な関心から自主的に参加しており、そのうちのほぼ半数［47%］は、雇用主から報酬を受け取って貢献しています（図24）。これらの数値は互いに排他的ではなく、内発的動機と有償の労働は競合するのではなく、しばしば共存していることを示しています。

図 23

以下の役割のうち、OSS への関わり方を最もよく表しているのはどれですか？（該当するものすべてを選択してください）

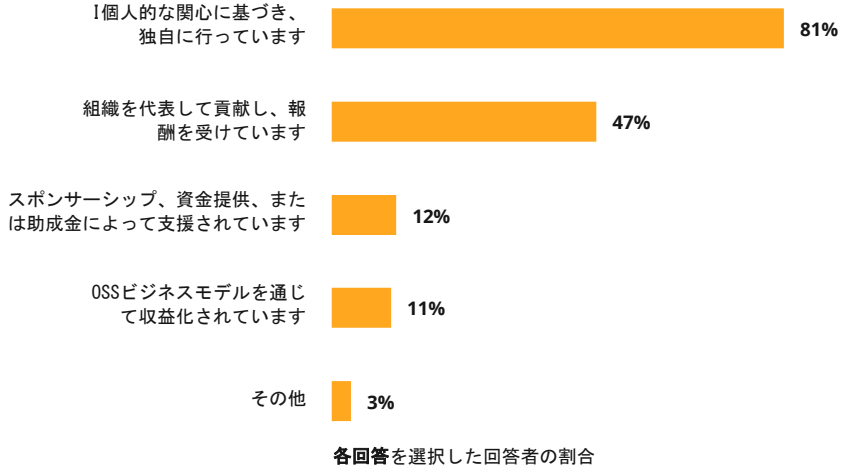
出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q2. 回答者数＝234名。回答総数＝512件。



図24

OSSプロジェクトには、どのような形で貢献されていますか？（該当するものをすべてお選びください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q6. 回答者数＝234名。回答総数＝361件。



その取り組みは在籍期間にも表れています。貢献者は非常に経験豊富で、約52%がプロジェクトへの貢献歴が11年以上であるとレポートしています（図25）。メンテナーとしての経験は在籍期間の区分ごとに比較的均等に分布しており、回答者の4分の1がメンテナーとしての経験が全くないとレポートしています。これは、このサンプルが、長年にわたりプロジェクトを率いてきたリーダーと、まだメンテナーの役割を担っていない貢献者の両方を捉えていることを示しています。

その活動自体は集中的ですが、範囲が狭いわけではありません。回答者が行う貢献の種類としては、コードや開発［89%］が最も多く、ドキュメント作成［60%］およびテスト／QA［59%］がトップ3を占めています（図26）。 これは、多くの回答者がプロジェクトを健全に維持するための周辺業務も担っている一方で、何よりもまず実践的な技術的貢献者であることを示しています。彼らが支援するプロジェクトは、インフラやシステムソフトウェアに偏っています。オペレーティングシステム／Linuxディストリビューション［37%］、ネットワーク／インフラ［28%］、プログラミング言語／ランタイム［27%］が最も一般的な分野です（図27）。これらは通常、個人／単独［63%］またはコミュニティ主導［58%］の取り組みとして組織されていますが、Foundation主導［49%］や企業支援［39%］のプロジェクトも相当数存在しています（図28）。これは、個人が運営するリポジトリから正式な組織構造を持つFoundationに至るまで、ガバナンスの形態が幅広い範囲に及んでいることを示しています。

図 25

以下のOSS関連の職務における実務経験年数を記入してください。
（1行につき1つ回答してください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。第4四半期。回答者数＝234名。

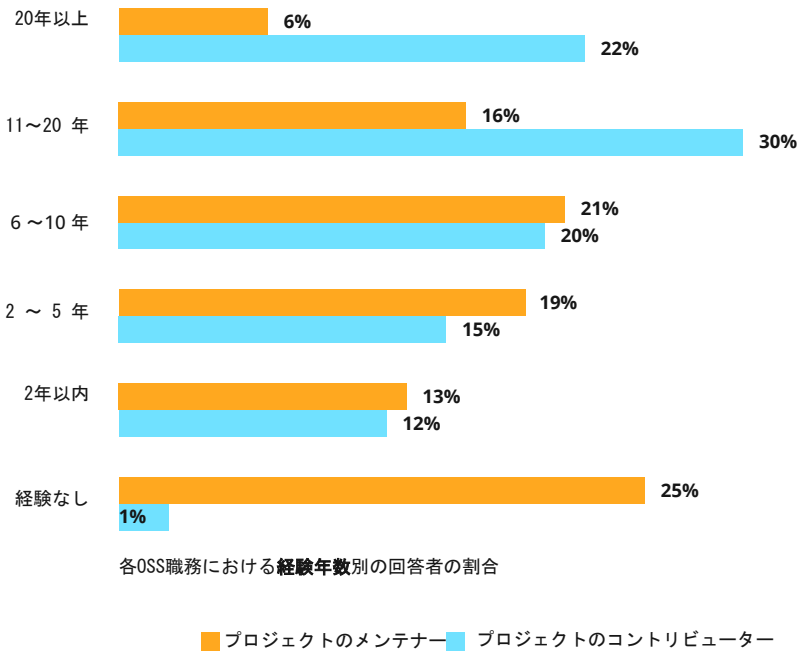


図 26

OSSエコシステムにどのような形で貢献されていますか？
（該当するものをすべてお選びください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q5. 回答者数＝234名。回答総数＝967件。

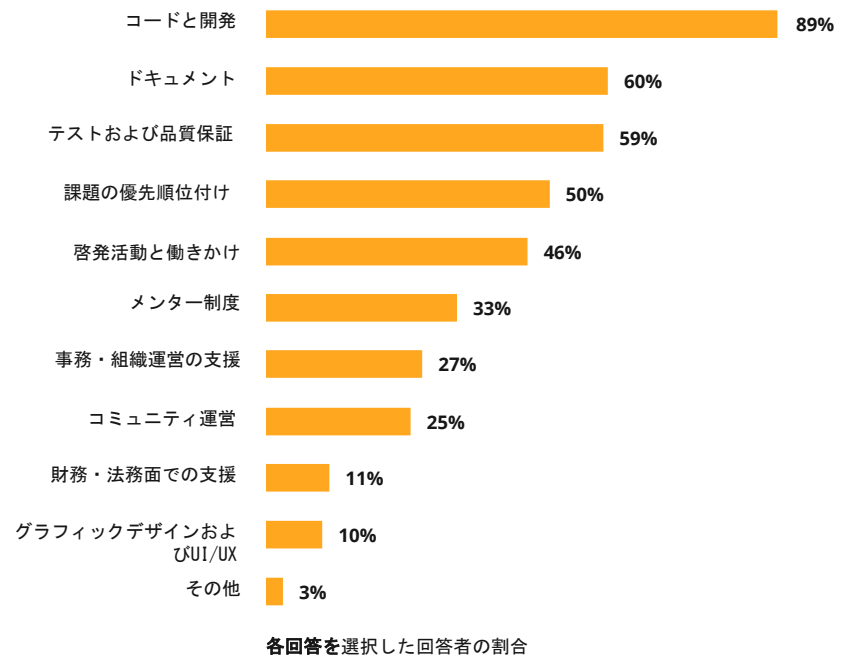
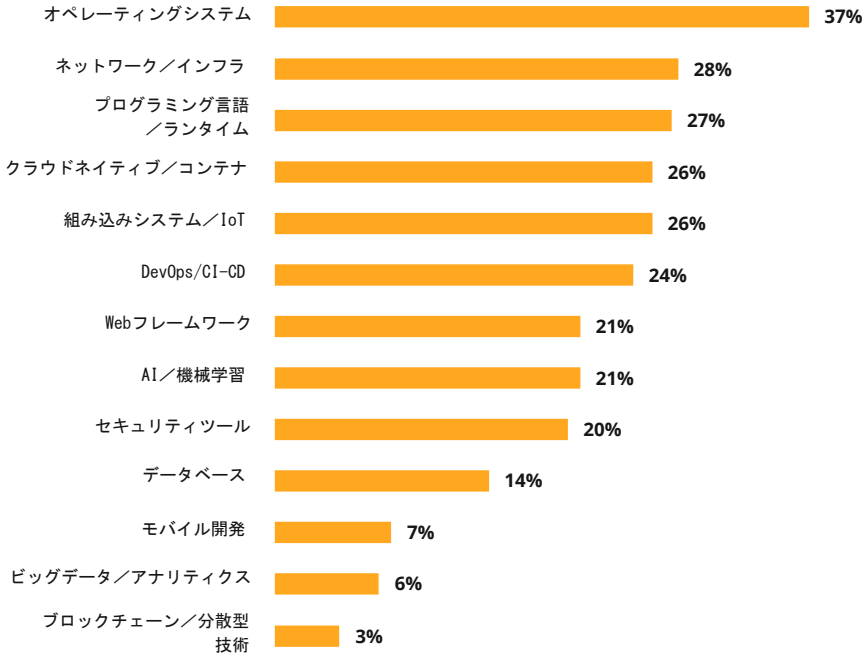


FIGURE 27

あなたが貢献しているOSSプロジェクトの主な分野として、以下のうちどれが最も当てはまりますか？（該当するものをすべて選択してください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q7. 回答者数＝213名。回答総数＝553件。

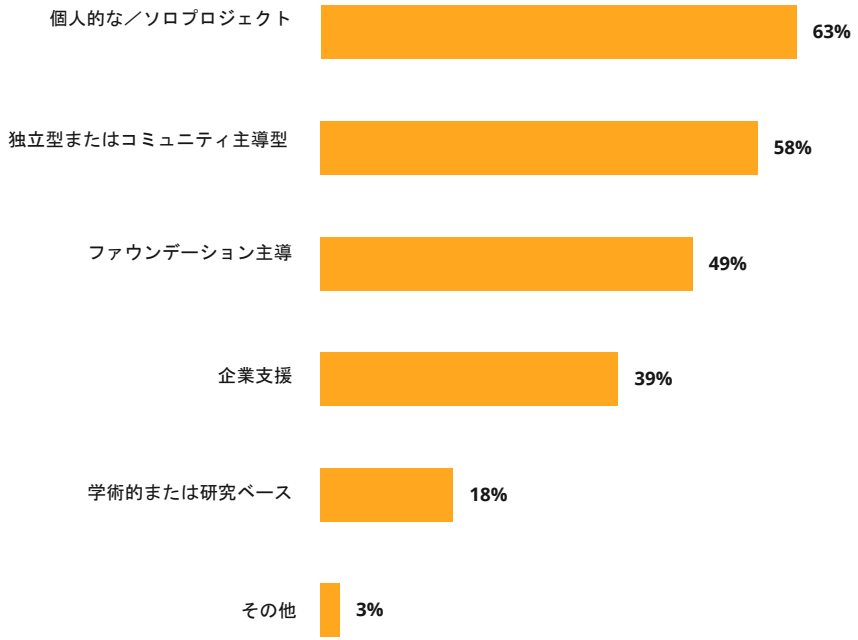


Percentage of respondents selecting **each response**

FIGURE 28

あなたが貢献しているOSSプロジェクトについて、以下のガバナンスモデルのうち、最も当てはまるものはどれですか？（該当するものすべてを選択してください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q8. 回答者数＝234名。回答総数＝536件。



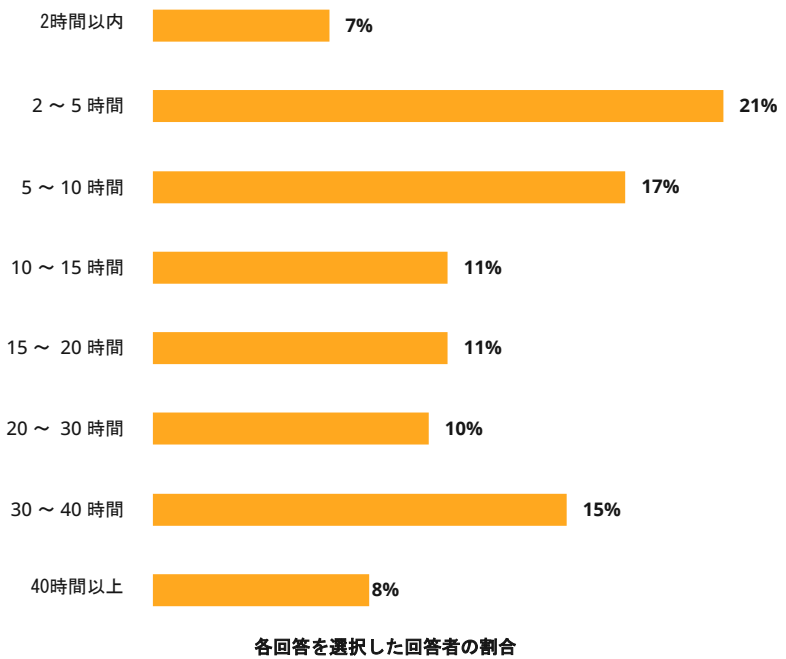
各回答を選択した回答者の割合

その作業のうち、何時間が費やされているかについては、バイモーダルな傾向が見られます。最も多い回答は「週2〜5時間」［21%］ですが、15%は「週30〜40時間」を費やしており、これは典型的なプロフィールというよりは、気軽に参加している人と深く関与している人が混在していることを示しています（図29）。

FIGURE 29

OSS開発には、週に何時間ほど割いていますか？（いずれか1つお選びください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q21。回答者数＝138名。

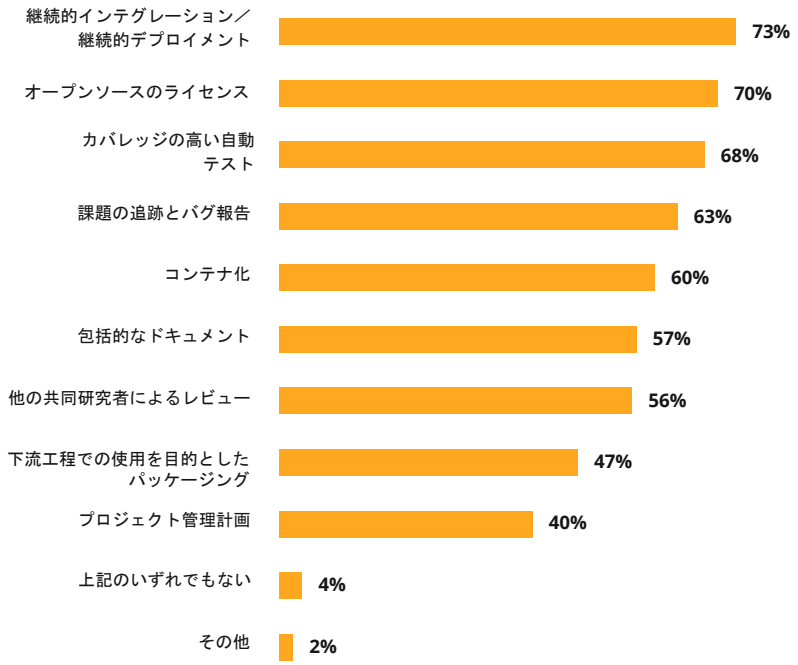


また、この対象グループは、成熟したエンジニアリングの取り組みを定着させていることも特徴です。CI/CD［73%］、オープンソースライセンス［70%］、自動テスト［68%］が最も広く採用されているベストプラクティスであり（図30）、本レポートの前半で説明した生成AIのワークフローは、その場しのぎのプロセスに後付けされるのではなく、すでに確立されたエンジニアリングの規律に従っているプロジェクトに組み込まれていることが示唆されます。

FIGURE 30

OSS開発のワークフローにおいて、以下のベストプラクティスのうち、どのようなものを実践されていますか？（該当するものすべてを選択してください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q16。回答者数＝139名。



生成AIの利用経験

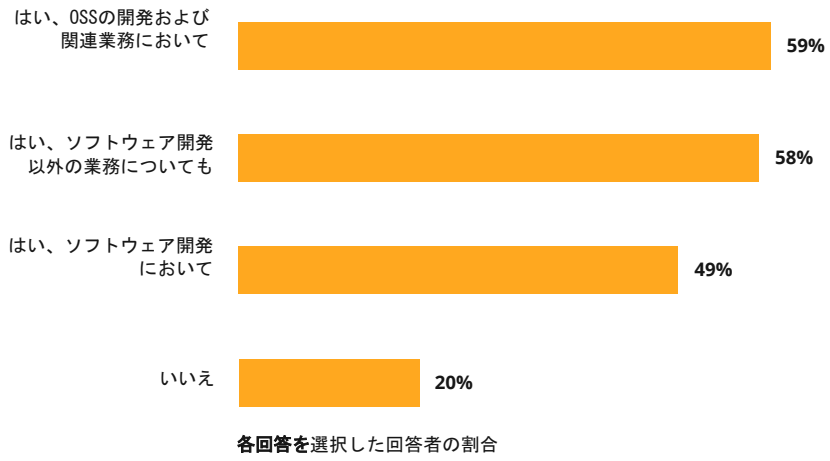
この対象層において、生成AIの利用はもはや例外ではなく、むしろ当たり前となっています。全く利用していないとレポートしたのはわずか20%にとどまり、その普及率はOSS開発［59%］、その他のタスク［58%］、非OSSソフトウェア開発［49%］の各分野でほぼ均等に広がっています （図31）– これは、レポートの前半で考察した動向が、生成AIを単なる目新しさではなく、すでにツールキットの日常的な一部として活用している対象層を反映していることを裏付けています。

利用状況は高頻度層に大きく集中しています。52%が毎日、23%がほぼ毎日ジェネレーティブAIを利用しており、つまり4分の3が少なくともほぼ毎日利用していることになります（図32）。その利用頻度は利用期間とも一致しており、過半数［57%］が1年以上ジェネレーティブAIを利用しています（図33）。これは、レポートで収集された見解の多くが、最近利用を始めた人やたまにしか利用しない人ではなく、経験豊富で習慣的に利用しているユーザーから得られたものであることを示しています。

FIGURE 31

生成AIをご利用ですか？（該当するものをすべてお選びください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。
Q1. 回答者数＝234名。回答総数＝437件。

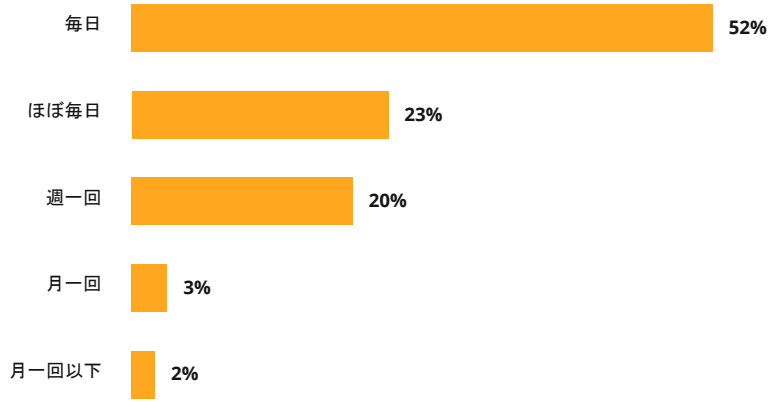


各回答を選択した回答者の割合

図 32

生成AIはどのくらいの頻度で利用されていますか？（1つお選びください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q10。回答者数＝187名。

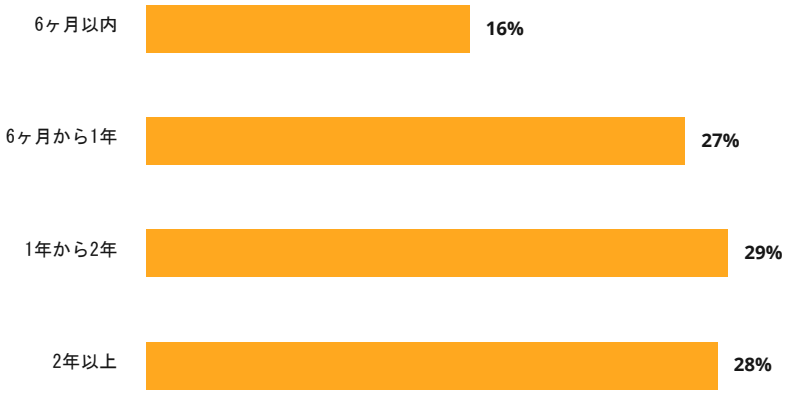


各回答を選択した回答者の割合

図 33

生成AIをどのくらいお使いですか？（いずれか1つお選びください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q11。回答者数＝187名。



各回答を選択した回答者の割合

メンテナーの属性

本レポート全体を通じて、メンテナーに関する調査結果を分類する上で、「プロジェクトの規模」と「役割」という2つの視点が用いられているため、メンテナーのサブサンプルについては、より詳しく検討する価値があります。

ほとんどのメンテナーは小規模なプロジェクト［2〜10人の貢献者：42%］に取り組んでいますが、マイクロ（単独）プロジェクトと超大規模（500人以上の貢献者）プロジェクトは、それぞれ12%ずつと、ごくわずかな割合で同程度に見られます（図34）。彼らの役割は、リード［32%］、共同メンテナ［25%］、単独［24%］の各役割にほぼ均等に分散しています（図35）。このバランスの取れた構成により、本レポートの他の箇所で示されているプロジェクト規模や役割に基づく調査結果は、小規模なリポジトリを単独で管理するメンテナから、大規模で分散したチームを統括するリードメンテナに至るまで、メンテナの経験全般を幅広く網羅しています。

図 34

あなたがメンテナンスを担当しているプロジェクトの、貢献者の規模はどの程度ですか？（いずれか1つを選択してください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。質問29。回答者数＝134名。

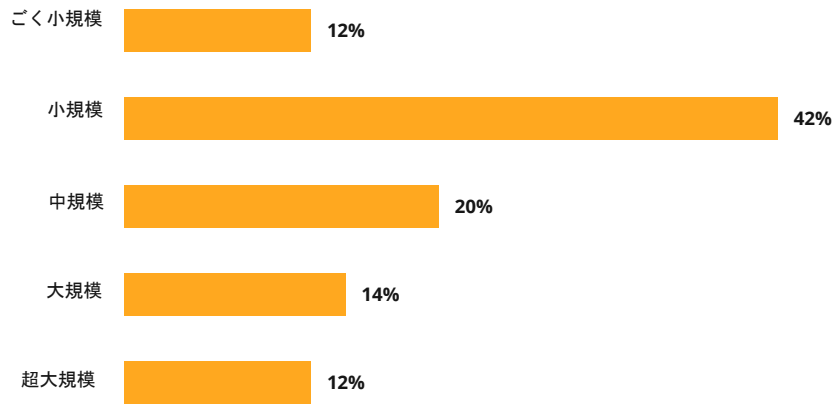
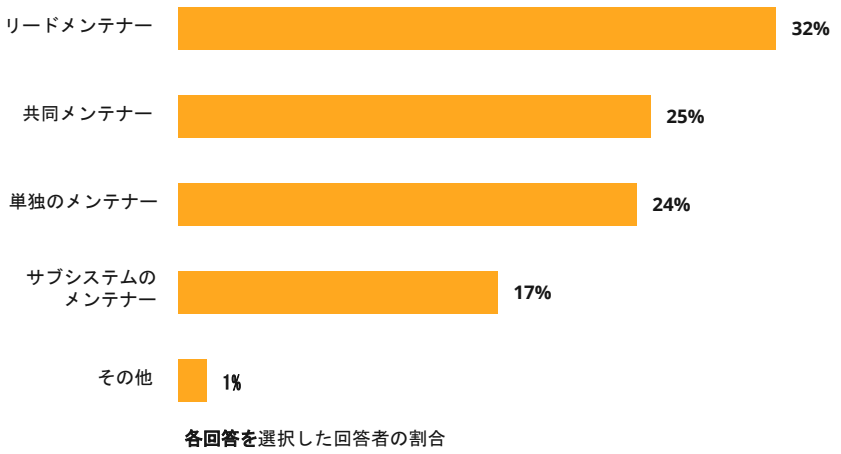


図 35

ご自身が保守に携わっているプロジェクトにおける、ご自身の担当業務についてご説明ください（1つ選択してください）

出典：生成AIとオープンソースソフトウェア開発に関する調査。Q28。回答者数＝135名。



著者について



SAM BOYSEL氏は、Linux Foundationのデータサイエンティストです。同氏はこの役職において、オープンソースコミュニティに関する新たな知見を導き出すための、さまざまな実証的な取り組みに貢献しています。同氏の研究では、ミクロ経済学の理論を活用し、オープンソースエコシステム全体における行動、インセンティブ、コミュニティの力学を分析することが多くあります。Linux Foundationに入職する前は、ハーバード大学のイノベーション科学研究所で博士研究員を務めていました。経済学の学士号、修士号、博士号を取得しています。



ANNI LAI 氏は、長年にわたりオープンソースコミュニティの構築、推進、啓蒙活動に携わってきた人物です。10年近くにわたりグローバルなオープンソースエコシステムに積極的に関与し、5大陸40カ国以上の開発者コミュニティを支援してきました。現在は、Generative AI Commonsの共同議長およびLF AI & Dataの選出理事を務めるほか、LF Europe、GOSIM、KAIYUANSHEのアドバイザーを務め、Futureweiではオープンソース運用部門を統括しています。アニーは、AIが倫理、信頼、そしてオープン性を基盤として、世界中の人々に奉仕すべきであると強く信じています。アニー氏は、ラトガース大学でコンピュータサイエンスの学士号を、サンノゼ州立大学で同分野の修士号をそれぞれ取得しています。

謝辞

本調査にご協力いただいた皆様に心より感謝申し上げます。特に、調査プロセスの各段階にご尽力いただいたLinux Foundationの同僚およびコラボレーションパートナーの皆様、Kimberly Craven氏、Anni Lai氏、Adrienn Lawson氏、Frank Nagle氏、Brian Proffitt氏、Christopher “CRob” Robinson氏、David A. Wheeler氏、Laura Guazzelli氏に深く感謝いたします。

Linux Foundationは、本調査を支援してくださった以下の組織に感謝申し上げます：

生成AI時代におけるオープンソースソフトウェア開発の進め方 | 36

Red Hat

レッドハットは、オープンハイブリッドクラウド技術のリーダーとして、企業における変革的なITおよび人工知能（AI）アプリケーションのための、一貫性のある包括的なFoundationを提供しています。フォーチュン500企業から信頼されるアドバイザーとして、レッドハットは、クラウド、開発者向け、Linux、自動化、アプリケーションプラットフォームの各技術に加え、数々の賞を受賞したサービスも提供しています。

THE LINUX FOUNDATION | Research

2021年に設立された**Linux Foundation Research** は、拡大を続けるオープンソースのコラボレーションの規模を調査し、新たな技術トレンド、ベストプラクティス、およびオープンソースプロジェクトが世界にもたらす影響に関する知見を提供しています。プロジェクトデータベースやネットワークを活用し、定量的・定性的手法におけるベストプラクティスを徹底することで、Linux Foundation Researchは、世界中の組織の利益となる、オープンソースに関する知見の頼れる情報源を構築しています。



x.com/linuxfoundation

facebook.com/TheLinuxFoundation

linkedin.com/company/the-linux-foundation

youtube.com/user/TheLinuxFoundation

github.com/linuxfoundation

Copyright © 2026 **The Linux Foundation**

本レポートは、**Creative Commons Attribution-NoDerivatives 4.0 International Public License**. の下で提供されています。

本論文を引用される際は、以下の形式に従ってください: Sam Boysel and Anni Lai, “Navigating Open Source Software Development in the Generative AI Era: Challenges and Opportunities for Project Communities,” foreword by Brian Proffitt, The Linux Foundation, October 2026.

この日本語文書は、英語版を機械翻訳し、参考訳として、The Linux Foundation Japanが便宜上提供するものです。
翻訳協力：吉田　行男