THE LINUX FOUNDATION | Research

オープンソースソフトウェアのセキュリティに関するメンテナーの視点

協力

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Stephen Hendrick, *The Linux Foundation*
Ashwin Ramaswami, *The Linux Foundation*

序文 Stephen Augustus, *Cisco*

2023 年 12 月

メンテナーを対象とした
調査によって得られた知見

セキュアなソフトウェア開発に
おけるベストプラクティスへの
取り組みについて

オープンソース ソフトウェアのセキュリティに関するメンテナーの視点

2023年末に、メンテナーとコア コントリビューターの72%が、**OSSはセキュアになる**と感じています。

使用されている OSSパッケージの セキュリティを評価する 第一位のアプローチは **SCAおよびSAST セキュリティツール**です。

メンテナーと コア コントリビューターの 39%はソースコードを **手動でレビュー** しています。

プロジェクトの ドキュメンテーションは 広く普及していますが、 どこにでもあるわけでは ありません。
87%のプロジェクトが、基本的なドキュメントを提供していると回答しました。

プロジェクトの56%は **再現可能な ビルドを サポートします。**

セキュリティツールの より賢い利用は、OSSサプライチェーン全体の セキュリティを向上させる 第一位のアプローチです。

自動化を通じて 開発者の負荷を 軽減することが OSSサプライチェーン 全体のセキュリティを 向上させる 第二位のアプローチです。

OSSコントリビューターの 69%はセキュアな ソフトウェア開発のため **確立された ベストプラクティス**を 求めています。

OSSプロジェクトを 維持する 第一位の理由は、**学ぶことの楽しさ** です。

OSSのコントリビューターの 49%はOSSへの貢献に対する **雇用主からの追加報酬**を 望んでいます。

メンテナーの4分の1(27%)は OSSの**セキュリティ ポリシーの定義**に関する 責任を負っています。

メンテナーの30%は **OSSセキュリティ ポリシーの実施**に 関する責任を負っています。

目次

序文

ここ数年、ソフトウェアのサプライチェーン セキュリティをめぐる会話はますます一般的になってきました。

LF Research が Open Source Security Foundation(OpenSSF) と共同で発表したこれらの知見に深く踏み込もうとしている方は、間違いなく、今日構築されているソフトウェアの大部分にオープンソース ソフトウェアが浸透しているという原則に精通していることでしょう。このことを念頭に置くと、セキュアなソフトウェア開発は、セキュアなオープンソース ソフトウェア プロジェクトとコミュニティを育成する方法に関する考慮を含んだ実践方法でなければならないのは理にかなっています。

私はセキュアなソフトウェア開発の専門家ではありません。しかし、私は日々の仕事の中で、オープンソース プロジェクトの消費者、コントリビューター、メンテナー、スポンサーという視点をいくつか持っています。

消費者として。私たちはプロジェクトの妥当性とこれからの発展に関心があります。このプロジェクトは、私が必要とし、期待することを行うか？ 導入は簡単か？ 私のチームが容易に理解できる言語で書かれているか？

コントリビューターとして。私たちの仕事上のコミットメントや、課題となる領域やその対応に対する個人的な興味から、プロジェクトに惹きつけられるのかもしれません。このような観点により、私たちはプロジェクトのニーズやコミュニティの振る舞いを理解し、そのエコシステムにとって価値がある効果的なメンバーになる方法を理解したいと考えます。

スポンサーとして。私たちはビジネスに重大な影響を与える分野に投資する傾向にあります。私たちにとって重要な開発分野に注意を払うというのみの義務感からプロジェクトを支援するのではなく、プロジェクト、その人材、そしてより広範なエコシステムの長期的な持続可能性に影響を与えるバランスの取れた取り組みを通じてプロジェクトを支援するよう努めるべきです。

これらの観点を念頭に置いて、私たちメンテナーは多方面にわたる役割を担っています。私たちは、自分達のプロジェクトを快適に利用いただけるよう全力を注いでいます。私たちは、学んだことをプロジェクトに還元する強い関心を持ってもらいたいと思っています。私たちは、コントリビューターがメンテナーになることをサポートできるような環境を作りたいと願っています。最後に、私たちが築き上げ、維持してきたものすべてが、スポンサーにとって十分魅力的な価値提案となり、支援していただけることを願っています。

ここで述べたことをまとめると、私たちは皆が、セキュアなソフトウェア開発の専門家ではないということです。LF Research のレポートが強調しているように、このメンテナーの視点から見ると、私たちはいくつかの重要な分野に取り組む必要があります。セキュアなソフトウェア開発の実践方法に関する教育、セキュアなコードをより容易に作成できるツールの実装と活用、セキュリティに関するベストプラクティスについて議論や改善、普及させるために必要となる他のオープンソースの専門家とのフィードバックループを作り上げることです。

このようなメンテナーの視点について考えるにあたり、この調査がオープンソースソフトウェアを安全にするためにあなた自身の立場からどのように関わると良いかという点について検討ができることを願っております。

結局のところ、オープンソースはグループ活動なのです。

STEPHEN AUGUSTUS
HEAD OF OPEN SOURCE, *CISCO*

はじめに

世界中の 90% 以上の組織が、オープンソース ソフトウェア (OSS) を利用しています。[1] OSS は、オペレーティング システムからインフラ、ミドルウェア、データ管理、サービス、フレームワーク、コンポーネント、アプリケーションまで、ソフトウェア スタック全体に存在します。

OSS のメンテナーは、オープンソース コンポーネントの開発と継続的な維持管理を担当します。メンテナーは、OSS エコシステムにおいて、OSS プロジェクトの方向性を舵取りし、その健全性と持続可能性を確保するという重要な役割を担っています。オープンソースは、機能の設計からドキュメントの作成、バグの修正、コードのレビューに至るまで、さまざまな作業を行うメンテナーやその他のコントリビューターからなるコミュニティという「人」に依存しています。これらの作業は、オープンソースのセキュリティ エコシステムをサポートし、安全性を確保するために不可欠です。しかし、長期的に効果的で持続可能であるためには、このような取り組みは、最終的にメンテナーを支援し、力を与えるものでなければならず、負担を増やすものであってはなりません。セキュリティに関するツール、実践方法、取り組みは、簡単に導入でき、オープンソース コミュニティのメンテナーを支援するものでなければなりません。

この目的のため、Linux Foundation Research は、OSS サプライチェーンのセキュリティに関する調査を実施しました。この調査は、OSS のセキュリティに対する考え方と、メンテナー、コア コントリビューター、エンドユーザー、およびその他の OSS エコシステムのメンバーによるセキュリティのベストプラクティスの導入と普及を理解することに重点を置いています。本調査では、OSS のメンテナーとコア コントリビューターに限定して、セキュアなソフトウェア開発に関する質問を行いました。調査は 2022 年 3 月に実施され、一部の調査結果は 2022 年 6 月に発表されました。本稿では、この調査から得られた、セキュアなソフトウェア開発のためのベストプラクティスの採用に関する過去に公開されていない情報を紹介します。この調査方法と調査対象の統計情報についての詳細は、本論文の評価方法のセクションを参照してください。

1. Adrienn Lawson and Stephen Hendrick, World of Open Source: Global Spotlight 2023 (San Francisco, The Linux Foundation, 2023), 9.

セキュアな開発のベストプラクティスは、Linux Foundation、特に Open Source Security Foundation（OpenSSF）が、セキュアなソフトウェア開発のためのベストプラクティスとして、**ベストプラクティス バッジ（bestpractices.dev）**や**スコアカード（securityscorecards.dev）**を定め、セキュアなソフトウェア開発のための無料のトレーニングと認定（**Developing Secure Software (LFD121)— Linux Foundation— Training**）を提供している分野です。

セキュリティの課題

OSS コンポーネントのセキュリティに対処するには、ベンダーがサポートするプロプライエタリなソフトウェアのセキュリティを確保する従来のアプローチとは異なるアプローチが必要です。典型的な OSS 開発による緩やかな構造とコミュニティ重視の性質は、ソフトウェア セキュリティの対処において異なる環境を示します。そこでは、少数の目に見える大規模プロジェクト（Linux カーネルや Kubernetes など）や多くの小規模プロジェクトが、OSS プロジェクトの提供方法を定義するのです。小規模なプロジェクトは、一般的にコントリビューターやリソースが少なく、そのため、開発とセキュリティに必要最小限のアプローチを採用する可能性が高くなります。

組織のソフトウェアにおける OSS の驚異的な普及とその恩恵は、OSS ソフトウェアのサプライチェーンにおけるセキュリティの危険性と相まって、私たちを岐路に立たせています。OSS を使用する組織や企業は、どのような依存関係を使用しているかをより認識し、すべてのコンポーネントの有用性、信頼性、脆弱性を積極的かつ定期的に監視する必要があります。最終的に、OSS は双方向です。OSS の消費者は、OSS コミュニティに貢献し、依存関係の健全性と存続性を保たなければなりません。OSS のユーザーが、そのソフトウェアが長期にわたって自分たちのニーズを満たすことを確保したいのであれば、貢献することなく OSS を単に使用するだけでは不十分であり、(a)OSS の依存関係という性質をサイバーセキュリティや開発の標準的な実施方法に組み込み、(b) 組織が依存する OSS コミュニティに貢献する必要があるのです。

オープンソース メンテナーと その他のオープンソース ソフトウェア コントリビューターの視点の比較

メンテナーとコア コントリビューターが、オープンソース コントリビューターの 36% を占めている

我々のオープンソース サプライチェーン調査には、OSS コントリビューターであると自身で判断している 441 名の回答者が含まれています。図 1 に示す OSS コントリビューターの分布は、メンテナー（20%）、コア コントリビューター（16%）、オケージョナル コントリビューター（49%）、ワンタイム コントリビューター（12%）、非開発コントリビューター（3%）です。それぞれの役割の定義は以下の通りです。

- **メンテナー：**ソフトウェアのメンテナーやパッケージのメンテナーは、ビルドやリリースされるソースコードのすべて、または一部に対する最終的な決定者です。メンテナーは、コア コントリビューターの下位グループとして分類される場合があります。
- **コア コントリビューター：**コア コントリビューターは、プロジェクトの立ち上げ当初から参加している場合もあれば、後から参加した場合もあります。また、コア コントリビューターは、コード ベースへのパッチを受け入れるなど、作業において重要な役割を継続的に担っています。プロジェクトのコミュニティは、コア コントリビューターを”コミッター”と呼ぶことがあります。
- **オケージョナル コントリビューター：**オケージョナル コントリビューターは、通常、進行中または毎週のプロジェクトのディスカッションには参加しませんが、長期にわたってプロジェクトに貢献することがあります。
- **ワンタイム コントリビューター：**ワンタイムコントリビューターとは、特定の提案や貢献を提供し、その作業が終わるとプロジェクトから抜ける人のことです。これらは”ドライブ バイ コミット”と呼ばれることもあります。
- **非開発コントリビューター：**ソフトウェアセキュリティに強い関心を持つ他の IT スタッフです。

図 1 を用いて、OSS コントリビューターを 2 つのカテゴリーに分類します。メンテナー、コア コントリビューター（36%）とその他の OSS コントリビューター（オケージョナル コントリビューター、ワンタイム コントリビューター、非開発コントリビューター（64%））です。この 2 つの OSS の区分は、OSS への貢献度や関与の度合いが高いコントリビューター（メンテナーやコア コントリビューター）と、貢献度や関与の度合いが低いコントリビューター（その他の OSS コントリビューター）を表しています。本レポートを通して、この 2 つの区分院おける、行動、新年、認識を比較し、何が同じであるか、何が異なるのかを確認します。

図 1

オープンソースソフトウェア開発における役割のうち、あなたに最もふさわしいものはどれですか？
（1 つを選択）

2022 年 OpenSSF サプライチェーン セキュリティ調査
Q14, サンプル数 = 441

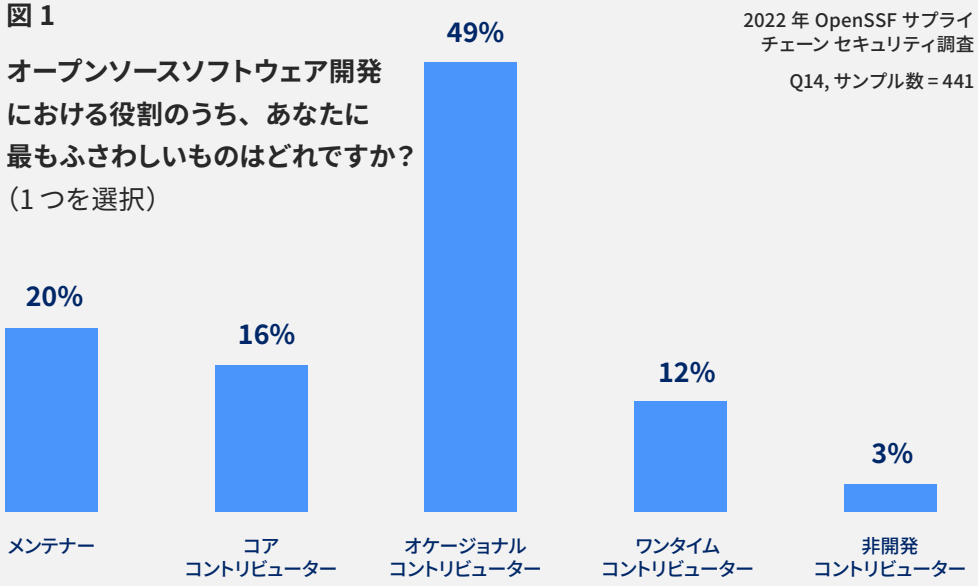
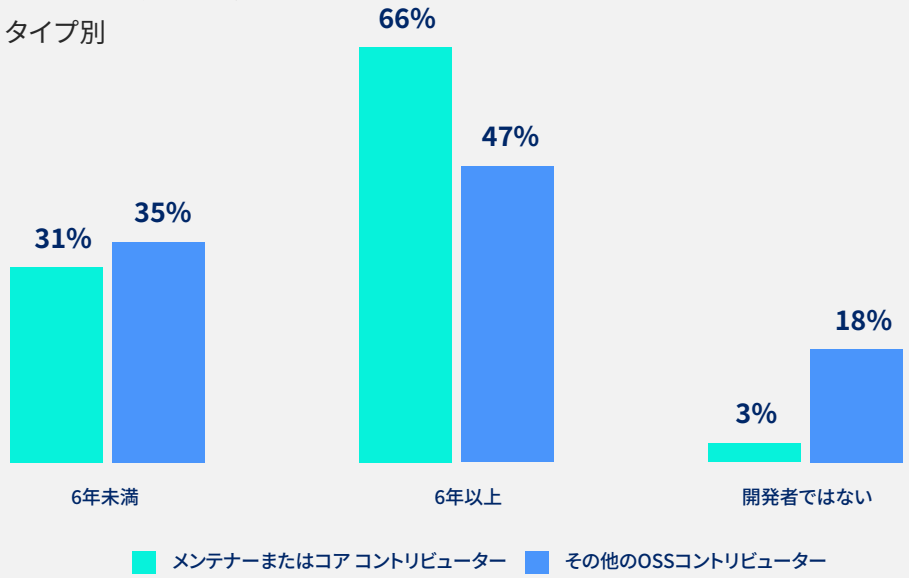


図 2

オープンソースソフトウェアの開発年数は？ (1 つを選択)
OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライチェーンセキュリティ調査

Q13 x Q14a、サンプル数 = 441



メンテナーとコア コントリビューターには、経験上のアドバンテージがある

メンテナーとコア コントリビューターの区分とその他の OSS コントリビューターの区分を比較した統計情報は、ほとんどの指標（年齢、雇用、地域、企業規模、業界、役割、担当分野）で類似していますが、1 つだけ差がある指標が経験年数です。図 2 を見ると、メンテナーとコア コントリビューターの 66% が 6 年以上の経験を持っているのに対し、その他の OSS コントリビューターは 47% です。

開発者ではないメンテナーやコア コントリビューターの割合も 3% であり、その他の OSS コントリビューターの 18% に比べて非常に低くなっています。これは、メンテナーやコア コントリビューターの役割がソースコードへのコントリビュートであることが多く、直接コードを書かないメンテナーであっても、どのようなコントリビューションが（コードを含む）承認を受けるべきかを決定し、新しい変更について文書化したり、コミュニティに知らせたりしなければならないからです。メンテナーがコードを理解することなく、コードのどのような変更が承認されるべきかを理解することは難しいでしょう。

オープンソースのコントリビューターはオープンソース ソフトウェアのセキュリティを意識しているが、あなたが期待するほどではない

図 3 は、OSS を開発、利用するプロセスがどれくらい安全であるかついて尋ねたもので、メンテナーやコア コントリビューター（62%）、その他の OSS コントリビューター（63%）は、OSS コードを開発、利用するプロセスを安全であると考えています。これは、OSS コントリビューターの 19% がこのプロセスを安全でないと考えていることと、OSS コントリビューターの 17 ～ 19% がどちらでもないと考えていることと対照的です。本レポートのデータは、ソフトウェア開発のセキュリティに関するより細かな状況を示すものであり、OSS コントリビューターがソフトウェア セキュリティに対処するために使用している方法と、これら 2 つの回答者区分に見られる行動と期待の違い（メンテナーやコア コントリビューターとその他の OSS コントリビューター）を示しています。

オープンソースのコントリビューターは、オープンソースのセキュリティがどのように改善されるかについて、疑念と期待を持っている

図 3 で、2022 年に開発中または使用中の OSS ソフトウェアが安全であると回答した OSS コントリビューターが 62% から 63% であったことを踏まえると、2022 年末までに OSS のその状態がどのように変化するかという点について、OSS コントリビューターが楽観的に見えることは興味深いことです。図 4 は、メンテナーとコア コントリビューターの 68% が、2022 年末までに開発中または使用中の OSS ソフトウェアが安全になると回答し（図 3 の 62% から増加）、安全でないと回答したのはわずか 8% でした（図 3 の 19% から減少）。残る 24% はどちらでもないと回答しました（安全でも安全でないとも言えない）。メンテナーとコア コントリビューターは、2022 年末までにソフトウェアのセキュリティが小幅に向上することを期待していますが、メンテナーとコア コントリビューターの内、OSS ソフトウェアが依然として安全でないと感じている人は、2022 年初頭の 19% から 2022 年末までに 8% へと大幅に減少しています。

図 3

現在、オープンソースソフトウェアを開発または利用する際のセキュリティはどの程度確保されていますか？ (1 つを選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライチェーン セキュリティ調査、Q17 x Q14a、サンプル数 = 348

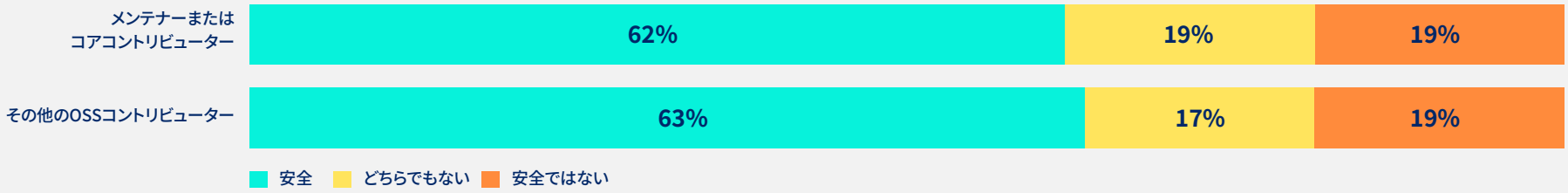
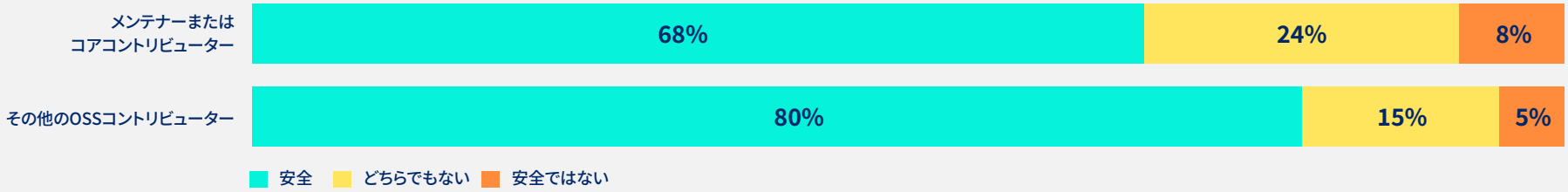


図 4

2022 年、あなたが開発または使用しているオープンソースソフトウェアのセキュリティはどのように変化すると思いますか？
(1 つを選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライチェーン セキュリティ調査、Q18 x Q14a、サンプル数 = 348



その他の OSS コントリビューターの動向はさらに極端です。図 4 は、その他の OSS コントリビューターの 80% が、2022 年末までに開発中または利用中の OSS ソフトウェアが安全であると考えており（図 3 の 63% から増加）、安全でないと考えているのはわずか 5%（図 3 の 19% から減少）であることを示しています。

続いて、2023 年に OSS のセキュリティがどのように進化するかについても質問しました。ここで、回答はさらに分かれました。図 5 を見ると、メンテナーとコア コントリビューターの 72% が、OSS は安全であると考えており、これは 2022 年後半（図 4）の値から 4 ポイント上昇し、2022 年前半（図 3）からは 10 ポイント上昇しました。2022 年初頭の 62% から 2023 年末までに 72% に上昇したことは、メンテナーがこの 2 年間で緩やかな改善を期待していることを示唆しています。このような 2 年間の改善を説明できるのは、以下のようなデプロイメントによるものです。

- セキュリティの問題を検出するための機械学習（ML）を用いた、静的アプリケーション セキュリティ テスト（SAST）や動的アプリケーション セキュリティ テスト（DAST）などのセキュリティ テスト ツールの改善
- より高いレベルのコミュニティ参加
- より良い依存関係の管理
- DevSecOps への取り組みの強化
- 政府の関与と規制
- 開発者トレーニングにおけるセキュアなソフトウェア開発の重要性の高まり

しかし、その他の OSS コントリビューターは、こうしたソフトウェア セキュリティの向上がさらに大きな影響を及ぼすと確信しているようです。図 5 を見ると、その他の OSS コントリビューターの 87%が、2023 年末までに OSS の利用や開発が安全になると感じており、2022 年末（図 4）から 7 ポイント、2022 年初頭（図 3）から 24 ポイント上昇しています。このデータは、その他の OSS コントリビューターと比較して、メンテナーやコア コントリビューターの視点のギャップが拡大していることを示しています。メンテナーと議論した結果、このような見解の大きな違いは、経験によるものであることが示唆されました。その他の OSS コントリビューターは、OSS のメンテナンスの周辺のみに参加しているため、OSS のセキュリティを向上させるための必要なステップを単純化しすぎている可能性があります。ある活動への関与が深くない場合、その活動は実際よりもずっと簡単に見えることがよくあります。これは、ダニング クルーガー効果の一例です。

図 5

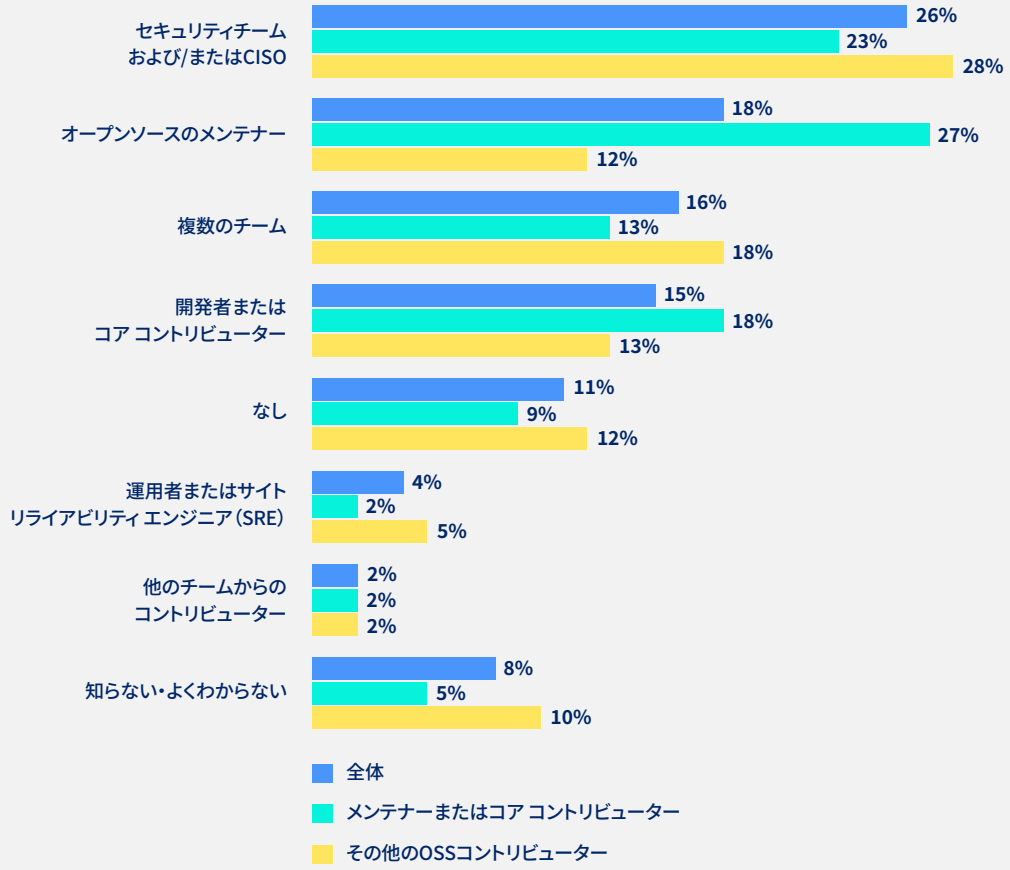
あなたが開発または使用しているオープンソースソフトウェアのセキュリティは、2023 年にどのように変化すると思いますか？
(1 つを選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q19 x Q14a、サンプル数 = 348

図 6

OSS セキュリティポリシーの策定責任者は誰ですか？ (1 つ選択)
OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q21 x Q14a、サンプル数 = 348



オープンソースのセキュリティ ポリシーの設定に、オープンソースのコントリビューターが関与している

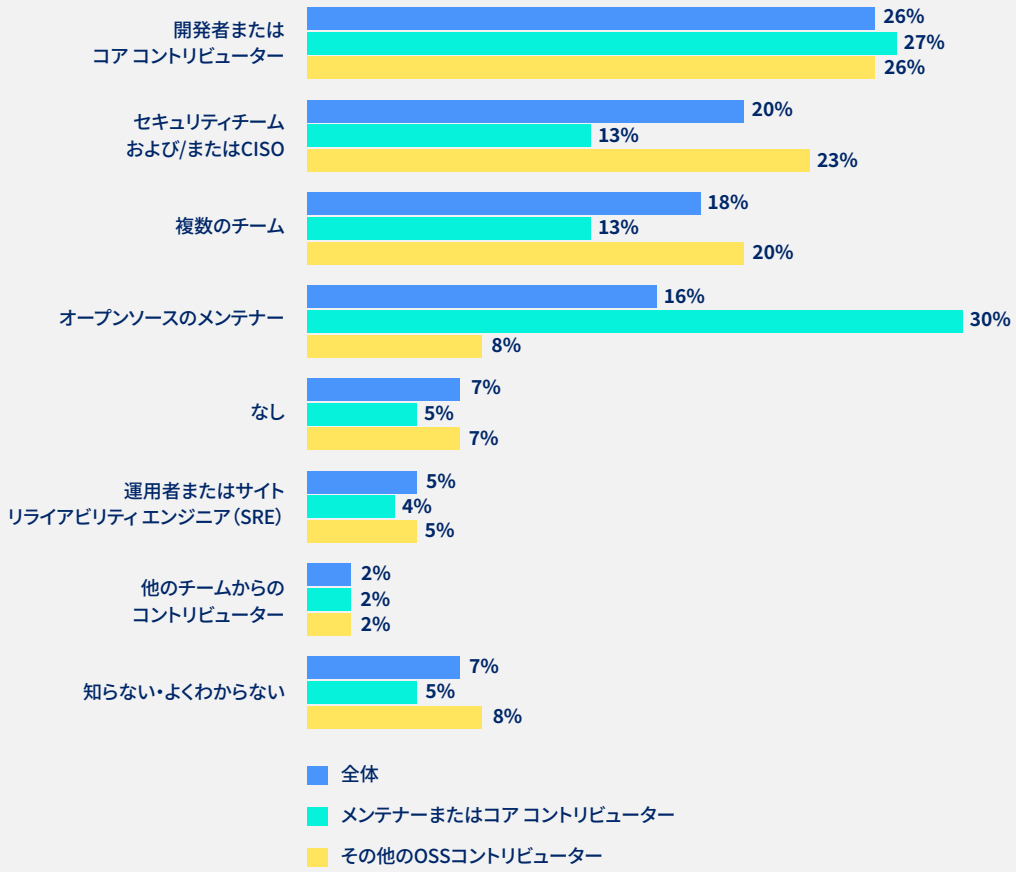
図 6 は、OSS コントリビューターの視点を反映したものですが、OSS のセキュリティ ポリシーの設定に OSS コントリビューターが高いレベルで関与していることを示しています。全体として、図 6 は、組織内で OSS のセキュリティ ポリシーを定義する主なアプローチは、CISO および / またはセキュリティ チームの責任であることを示しています。しかし、OSS メンテナーの 27% (その他の OSS コントリビューターの 12%) が、OSS のセキュリティ ポリシーの策定責任はメンテナーにあると回答しています。これは、セキュリティ チーム、CISO、またはオープンソース プログラム オフィス (OSPO) がその役割を担っていない場合に、多くの組織が取る体制であるようです。

図 6 は、OSS のセキュリティ ポリシーの定義に複数の IT チームを関与させることが、場合によっては解決策になることも示しています。その理由は、サイバーセキュリティのニーズが、開発者、ネットワーク セキュリティ、リスク管理、コンプライアンスなど、広範囲に及ぶからです。複数のチームを活用することで、多様な経験を生かすことができ、定義するポリシーにサイバー セキュリティのより多くの側面を含めることができます。

図 7

開発・利用の活動全体にわたるセキュリティ実装の主な責任者は誰ですか？ (1 つ選択)
OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q22 x Q14a、サンプル数 = 348



オープンソースのコントリビューターや開発者は、セキュリティポリシーの実装をすべて担っている

図 7 によると、セキュリティ ポリシーの実施に主に責任を持つのは誰かという質問に対して、開発者とコア コントリビューターという回答（すべての OSS コントリビューター）が 26% で、全体のトップでした。セキュリティチームおよび / または CISO（20%）、複数のチーム（18%）、オープンソースのメンテナー（16%）は、開発者の役割を強化すると同時に、セキュリティ ポリシーの実施におけるセキュリティ チームの重要性を明らかにしています。

やや意外なのは、図 7 のうち 3 つの回答で、メンテナーやコア コントリビューターとその他の OSS コントリビューターとの間に意見の違いが現れていることです。セキュリティチームと CISO、複数のチーム、オープンソースのメンテナー。それぞれのセグメントの経験と参照する範囲が、セグメント間の差異として影響を与えているようです。しかし、サイバー セキュリティ ポリシーの実施には、広範囲ゆえに協力する体制が必要であり、メンテナーや中心的なコントリビューターの関与が期待されるのは当然ですが、これには他の開発者やセキュリティチームも含まれると我々は評価しています。

メンテナーがサイバー セキュリティ対策において手作業による検査の重要性を示している

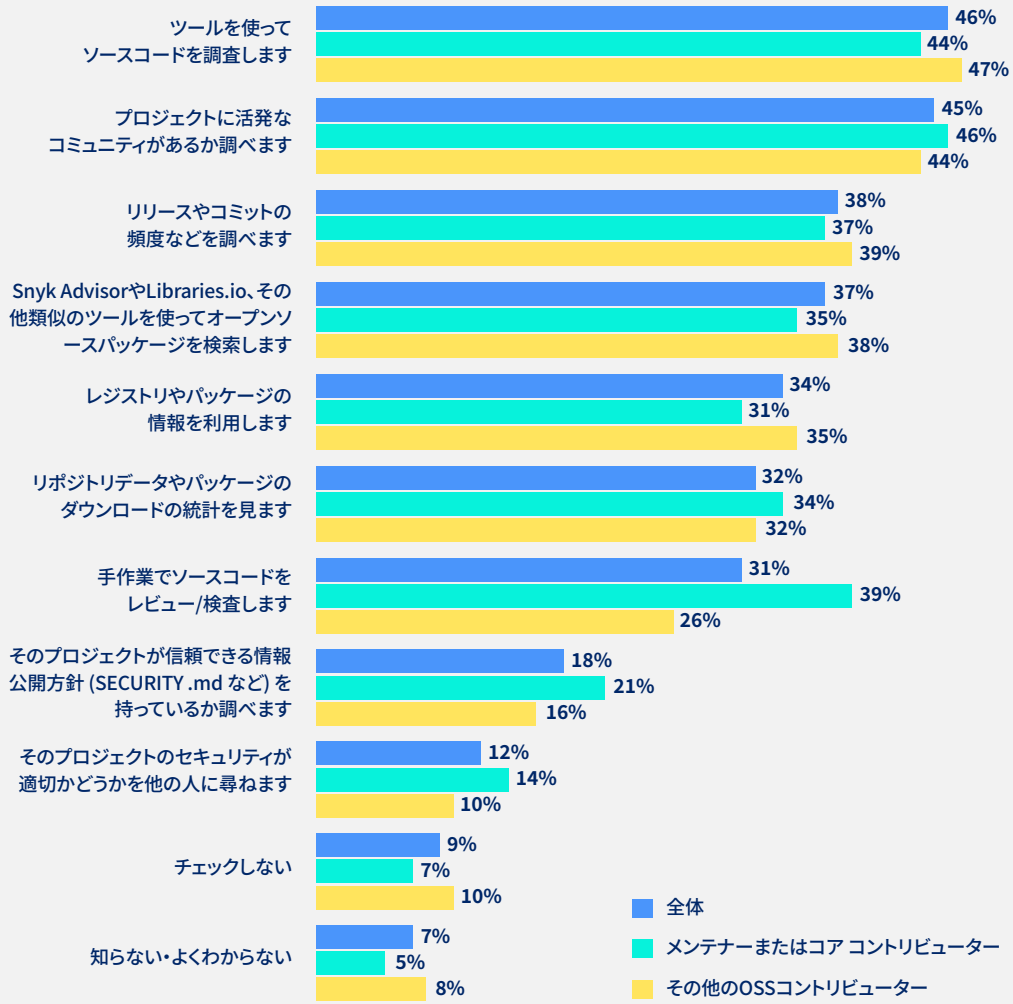
OSS パッケージの安全性を確認するための最も一般的なアプローチは、ソースコードを精査し、その活動を調査するためのツールを利用することです。図 8 を見ると、OSS コントリビューターの 46%がツールを使ってソースコードを調査しており、45%はプロジェクトに活発な開発者コミュニティがあるかどうかもチェックし、そして、38% はリリースとコミットの頻度を見ています。ソースコードの調査に対するこれらの主要なアプローチにおいて、OSS コントリビューターの両セグメントで一致しているという点も重要です。

図 8

利用しているオープンソースパッケージのセキュリティをどのようにチェックしていますか?

(該当するものをすべて選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q28 x Q14a、サンプル数 348、有効数 = 348、総数 = 1,070



ソースコードを調査するためにツールを使用することは、正当な理由のある代表的なアプローチです。開発者がツールを利用するのは、時間が限られているからです。ツールは、コードを分析するための迅速な方法を提供します。機械学習は、最終的にはセキュリティ ツールの能力を向上させるでしょうが、手作業による検査に代わるものはまだないというのがコンセンサスのようです。

図 8 で、一つ大きな違いが見られるものがあります。それは、39%のメンテナーとコア コントリビューターが手作業でソースコードのレビューと検査を行っているというものです。メンテナーとコア コントリビューターの主な役割は、コンポーネントの機能を向上させることです。しかし、それがどのように機能するかについて良いアイデアがない限り、コードを修正することはできません。そのため、メンテナーは手作業でコードをレビューし、その機能の提供内容を理解する必要があります。OSS コンポーネントのセキュリティを確保することは、コードの維持における煩わしい側面かもしれませんが、手作業によるコードレビューはそれを円滑にします。そのため、メンテナーはコードの機能性と品質を向上させるために変更や追加するコードに対して手作業によるコードレビューと検査を優先して行いますが、同時にこの機会を利用してセキュリティ上の問題を特定し対処も行っています。

セキュリティツールの使用は、提供される価値と相関関係がある

図 9 は、OSS コントリビューター全体の 50% がソフトウェア構成分析(SCA)ツールを使用していることを示しています。SCA ツールは、組織で使用されている OSS コンポーネント全体のライセンスのコンプライアンスに関する問題や CVE（Common Vulnerabilities and Exposures）を特定するのに非常に効果的です。SCA ツールは、CI/CD パイプラインに統合するのが難しくないため、採用が進んでいます。

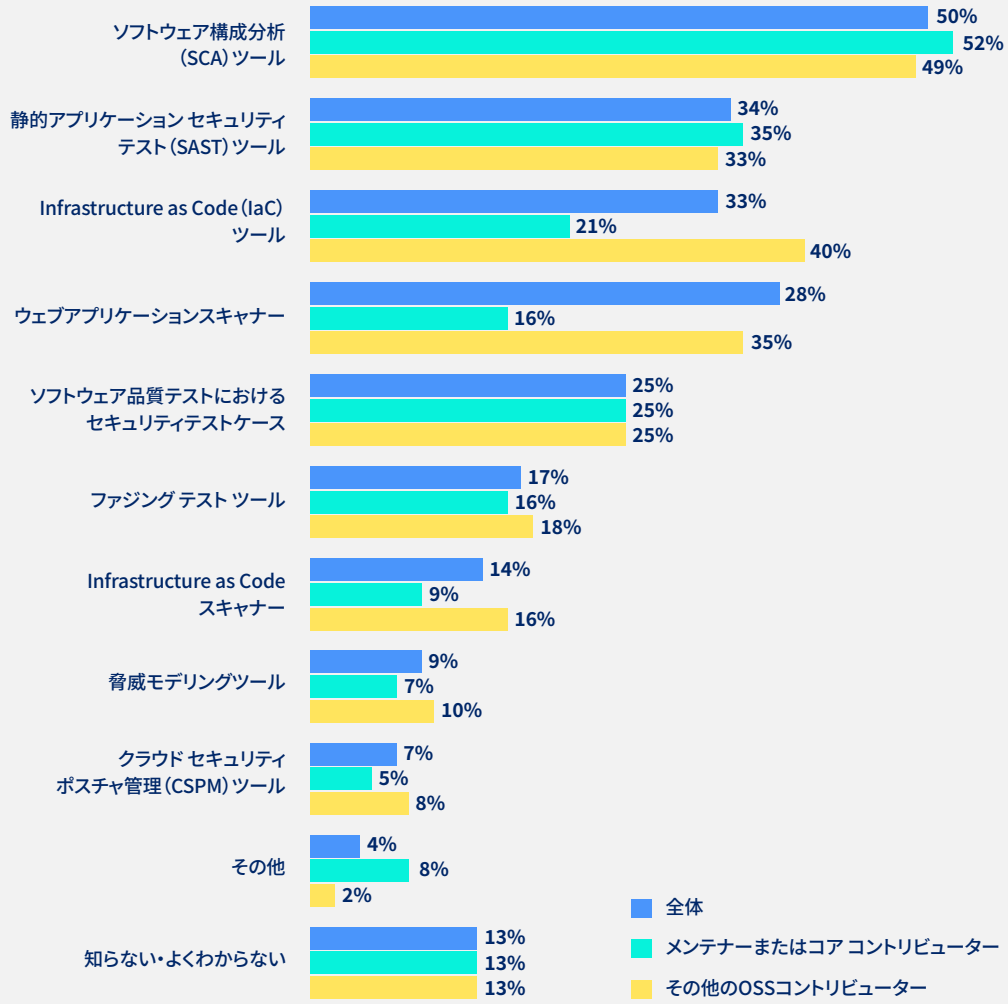
図 9 では、SAST ツールの使用率が 34%、Infrastructure as Code（IaC）ツールの使用率が 33% となっており、OSS コントリビューターによるツール使用の 2 番目の傾向を示しています。ソフトウェア開発における SAST ツールの一般的な使用方法は、SCA ツールと同様に、開発中、コードがコミットされる前、コードレビュー中、コードがコミットされた後、継続的インテグレーション中、デプロイ前にコードをテストすることです。セキュリティの脆弱性を発見することは、

図 9

オープンソース ソフトウェアを開発する際、どのようなセキュリティ ツールを日常的に使用していますか？

(該当するものをすべて選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q29 x Q14a、サンプル数 348、有効数 = 348、総数 = 813



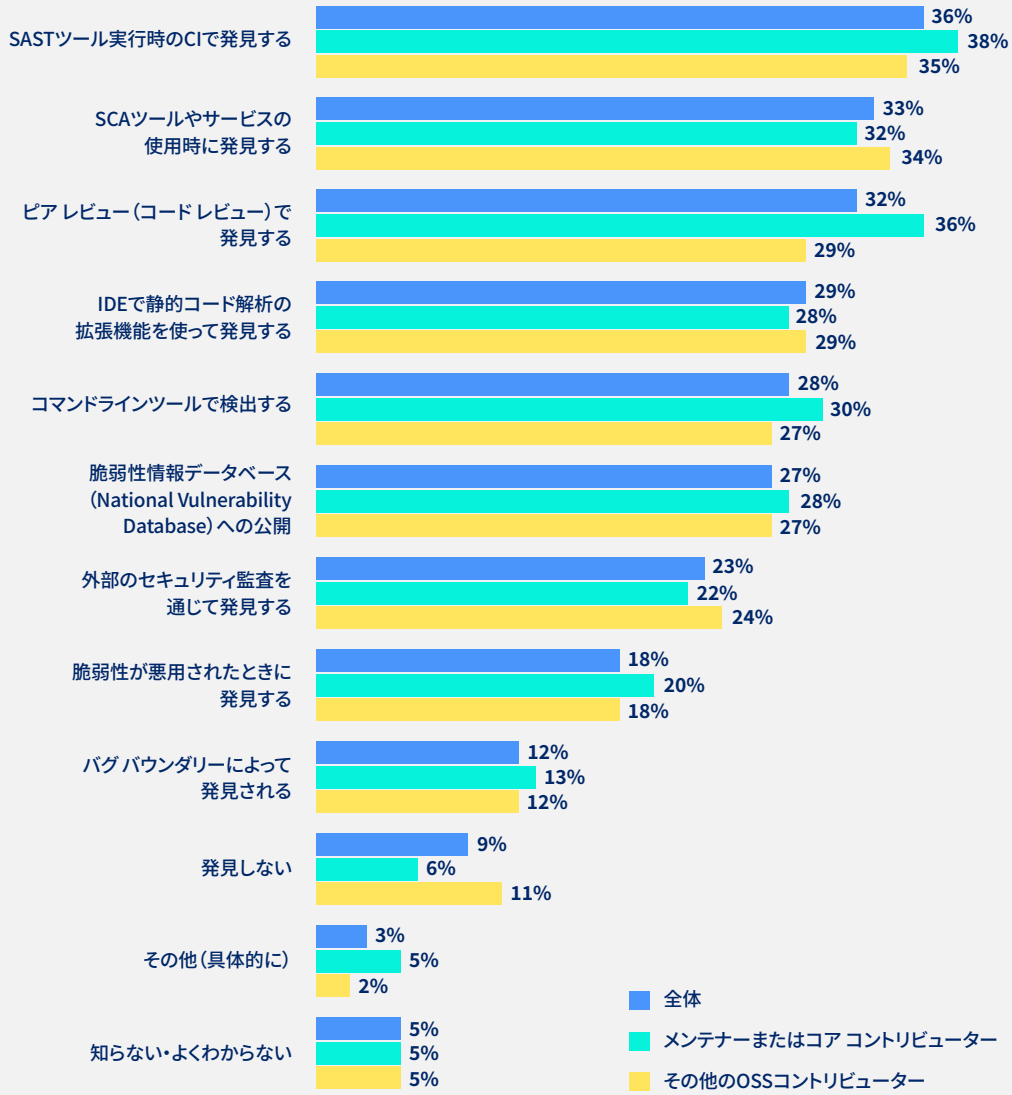
バグを発見することと同じで、ソフトウェア開発のできるだけ早い時期に行うべきです。脆弱性とバグは、開発中のタイミング以降で発見された場合、修正にかかる費用が指数関数的に高くなります。SAST ツールがよりインテリジェントになるにつれて、セキュリティ脅威に対する自動化された主要な防御としての役割を果たすため、その採用が大幅に増加することは心強いことです。

最後に、OSS コントリビューターの 33%が使用している IaC ツールは、主要なソフトウェア開発プロセスを自動化する上で非常に効果的です。CI/CD 活動全体の手動で行うタッチ ポイントを減らすことは、自動化によるプロビジョニング、管理、デプロイメントのようなミッション クリティカルな活動における危険を低減する重要な方法です。IaC ツールの使用について、その他の OSS コントリビューターの 40% と比較して、メンテナーとコア コントリビューターのわずか 21% であることに留意する必要があります。メンテナーとコア コントリビューターは、最終的に本番環境で使用されるコードをリリースする責任があることは明白ですが、その他の OSS コントリビューターの役割はやや曖昧です。データからわかっていることは、その他の OSS コントリビューターはツールの利用が多く、IaC ツール（40% 対 21%）、ウェブ アプリケーション スキャナー（35% 対 16%）、IaC スキャナー（16% 対 9%）となっています。言えることは、その他の OSS コントリビューターは、平均してより熱心なツールのユーザーであるということです。これは、彼らの経験が少ないため、あるいは、彼らの役割がソフトウェア開発ライフサイクル（SDLC）全体にわたってより多くの活動に関与しているためかもしれません。

図 10

コードのセキュリティの脆弱性をどのように発見しますか？（該当するものをすべて選択）
OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q30 x Q14a、サンプル数 = 348、有効数 = 348、総数 = 892



オープンソースのコントリビューターがセキュリティの脆弱性を発見する方法

すべての脆弱性を発見できる保証はありません。したがって、脆弱性を発見するためには、図 10 が示すように、OSS のコントリビューターがさまざまなアプローチを駆使する必要があります。一般的には、SAST（36%）や SCA（33%）などのセキュリティツールや、場合によっては、SAST 拡張機能を備えた統合開発環境（IDE）（29%）、コマンドライン ツール（28%）などの使用です。ただし、コード レビュー中に脆弱性を特定することは、OSS コントリビューターでは一般的（32%）であり、メンテナーとコア コントリビューターにおける脆弱性を発見する主要な方法（36%）です。この方法は、より多くの労力を要するにもかかわらず、単なる脆弱性の範囲を超えて、問題を特定することができるため、その重要性を過小評価すべきではありません。

SCA ツールは、OSS コントリビューターやユーザーが、繰り返し利用されるコンポーネントの既知の脆弱性を発見するための主要な手段です。SAST ツールは、未知の脆弱性を特定することができます。これが、SCA ツールと SAST ツールの両方を使用することが非常に重要である理由です。

図 10 によると、OSS コントリビューターの 36% が SAST ツールを、33% が SCA ツールを使用しています。全体では、32% がピア レビュー中に脆弱性を発見すると回答しており、やはりメンテナーとコア コントリビューター（36%）がその他の OSS コントリビューター（29%）よりもこのアプローチに依存しています。

セキュアなソフトウェア開発に関するメンテナーの視点

OSS セキュリティ調査では、OSS メンテナーとコア コントリビューターのみを対象に、メンテナーと開発責任をどのように果たしているかを明確にするための質問項目を設けました。前述の通り、図 1 に示すように、OSS コントリビューターの 36%がメンテナーまたはコア コントリビューターです（回答者数 159 名）。これらのメンテナーおよびコア コントリビューターのうち、72 人は、参加している主要なオープンソースプロジェクトに関する質問と、開発およびメンテナーの責任をどのように果たしているかについての質問に回答してくれました。

この調査では、メンテナーまたはコア コントリビューターによるセキュアなソフトウェア開発のベストプラクティスの採用について、52 の設問にわたる一連の 7 つの質問を行いました。この 7 つの質問は、プロジェクト管理、ソースコード管理、ビルドプロセス、ソフトウェア品質保証（SQA）、ソフトウェアセキュリティ、セキュリティテスト、セキュアコーディングなど、プロジェクトのフェーズから選択された OSS の開発に関わる活動で構成されています。

52 のベストプラクティス（セキュアなソフトウェア開発に関わる活動）それぞれについて、回答者は採用の段階を次の 5 つから選ぶことができました。

1. 現在使用中
2. 2022 年または 2023 年に使用予定
3. 使用予定なし
4. 該当なし
5. 知らない・よくわからない

なお、ロジスティクス曲線を特定の製品、技術、ベストプラクティスの市場導入の代理と見なす場合は、85 ～ 90%を「最大目標導入率」と見なすことに注意してください。その理由は、ベストプラクティスが必ずしもすべてのメンテナーやコア コントリビューターに適用できるとは限らないからです。その開発プロセスがベストプラクティスを必要とするか、サポートするかどうかを知らない、あるいは確信が持てない人が、常に何パーセントか存在します。

その結果、現在の使用と計画中の使用の組み合わせが、この最大目標採用範囲に近づくか、それを超えるものであれば、それは素晴らしい発見です。

ほとんどのメンテナーやコントリビューターは、OSS の利用や貢献に関する基本的な要件を支持している

まず、基本的なベストプラクティスの採用に関する調査を始めました。図 11 を見ると、現在広く使われている活動として、プロジェクトが基本的なドキュメントを確実に備えていること（87%）、ライセンスの条項を掲示すること（84%）、積極的にプロジェクトをメンテナンスすること（83%）、プロジェクトのウェブサイトでプロジェクトが何をするのかを説明すること（80%）、プロジェクトが人々が変更や問題について議論できるようにすること（79%）が挙げられます。

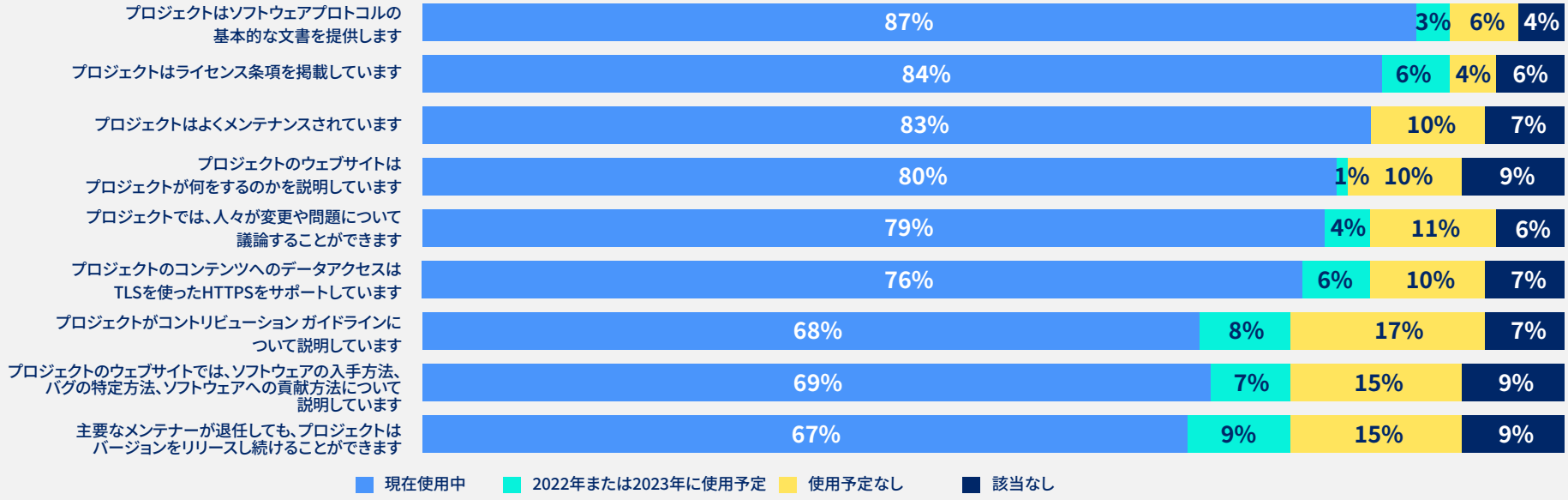
しかし、図 11 を見ると、メンテナーとコア コントリビューターの 87% が、プロジェクトがソフトウェアの基本的なドキュメントを提供していると回答している一方で、プロジェクトがコントリビューションガイドラインを説明していると回答したのは、これらの回答者の 68% のみでした。また、主要なメンテナーが退任した場合でも、プロジェクトがバージョンのリリースを続けられるという確信を示したメンテナーは、より少数でした。質的なインタビューでは、メンテナーは、持続可能性がソフトウェアのサプライチェーン セキュリティの要素でもあるという事実を指摘しています。メンテナーが退任したらどうなるのでしょうか？

図 11 の良い特徴は、採用の割合（現在使用中）が高いため、採用の段階が異なるメンテナーやコア コントリビューターの割合が減少することです。一般的に、ベストプラクティスを採用しているメンテナーやコア コントリビューターの数が増えるにつれて、計画中やその他の選択肢を回答するメンテナーやコア コントリビューターが減っていきます。最初の 6 つのベストプラクティスでは、このようなパターンが見られます。同様に残念なのは、最後の 3 つの活動において、プロジェクトがコントリビューション ガイドラインをどのように説明するか（17%）、バグをどのように特定してソフトウェアに貢献するか（15%）、リード メンテナーが退任したときにどうなるか（15%）、について取り組む予定がないメンテナーとコア コントリビューターの水準が高いことです。

図 11

プロジェクトは、以下の基本的なベストプラクティスをサポートしていますか？

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q38、サンプル数 = 72、分析から「知らない・よくわからない」を除外



メンテナーのソースコード管理と変更管理への取り組み方

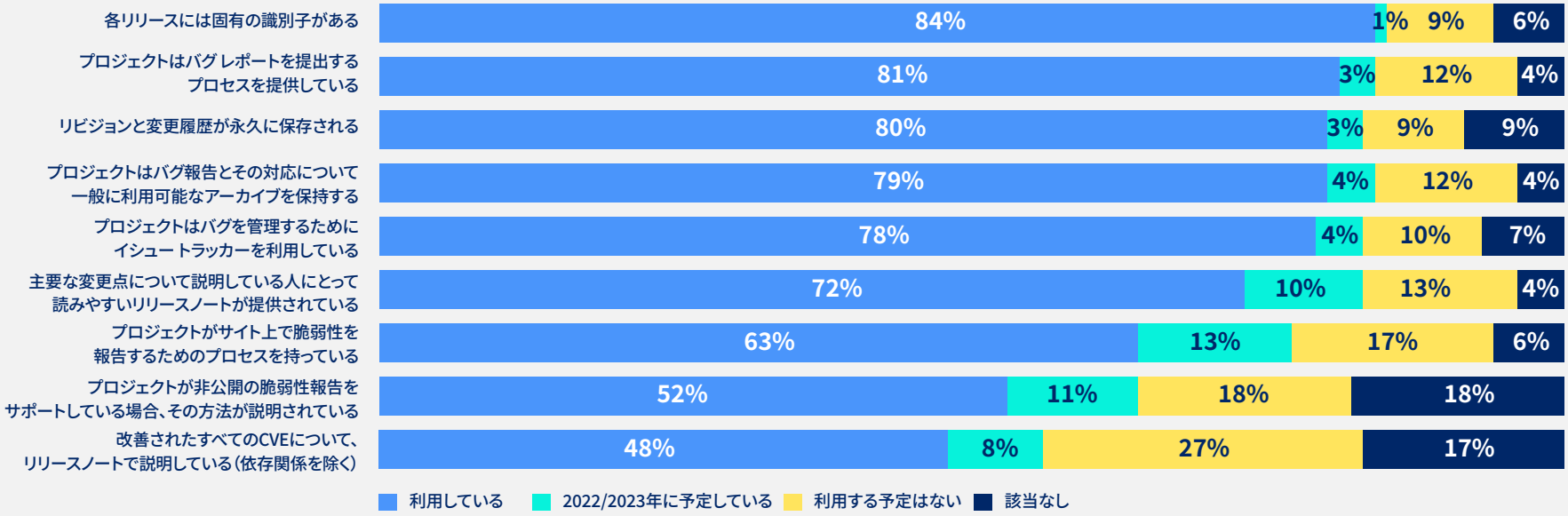
ソースコード管理と変更管理は、オープンソース プロジェクトの効果的な管理、コラボレーション、安定性、および成長のための基礎となります。これらは、オープンソース コミュニティが協力し、開発プロセスを確立するために必要なイン フラとプロセスを提供します。

図 12 によると、メンテナーとコア コントリビューターの 72 ～ 84% が、主要なソースコード管理と変更管理のベストプラクティスに従っています。しかし、脆弱性の特定と修復の報告についてはかなり遅れており、メンテナーとコア コントリビューターの採用率は 48 ～ 63% です。メンテナーへのインタビューでは、Twitter、電子メール、Slack などの非公式なチャネルを通じて、一般に知られている脆弱性について情報を発信することもあると述べている人もいましたが、プロジェクトによる脆弱性の管理については、常に記録システムが必要であり、CVE のレジストリやその他の脆弱性を特定するための非公式なチャネルを含めなければいけません。

図 12

プロジェクトは、以下のソースコード管理と変更管理のベストプラクティスをサポートしていますか？

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q39、サンプル数 = 72、分析から「知らない・よくわからない」を除外



ビルド プロセスに関するベストプラクティスの提供は、長所と短所を併せ持っている

このセクションでは、ビルド プロセスに関するベストプラクティスについて説明します。図 13 に示すように、メンテナーとコア コントリビューターの大多数が、一般的なビルド作業に取り組むためのベストプラクティスを採用しています。たとえば、ビルド定義をバージョン管理システム（VCS）に保存する（78%）、すべてのビルド手順をビルドスクリプトの一部として使用する（75%）、ビルド サービスを分離された環境で実行する（69%）、ビルド サービスを一時的な環境として実行する（63%）、コンピュータで読み取り可能な方法で依存関係を記載する（63%）、セキュア設計の原則を使用する（60%）、再現可能なビルドをサポートする（56%）などです。

ほとんどの開発者は、基本的な実装のベストプラクティスの多くに従いましたが、ビルドにおける改ざんの検証（証明）（36%）やリリースのデジタル署名（認証）（20%）など、より複雑な機能については、大幅な採用には至りませんでした。問題の一部は、この用語に対する一般的な理解が不足していることかもしれません。インタビューの中で、あるメンテナーは、多くのメンテナーはビルドにおける証明や認証という言葉を聞いたことがなくても、この問題をどのように説明するかは知っていると述べています。

図 13 はまた、24% のメンテナーがメタデータを改ざん（証明）できないビルドサービスを使用する予定がない、またはメタデータの改ざん（証明）に関するプラクティスは適用できないと見なしていることを示しています。同様に、メンテナーとコア コントリビューターの 42% は、リリースにデジタル署名をする予定がないか、この種の認証は適用されないと考えています。これらの割合はたしかに大きいですが、メンテナーが主に自分自身の使用のためにプロジェクトを保守しており、信頼が問題にならない場合もあります。メンテナーを信頼しているようなプロジェクトの他のユーザーは、改ざんの証明や認証の必要性を強く感じないかもしれません。最後に、多くの OSS プロジェクトはソフトウェアを直接ビルドしません。ソースコードを公開するだけです。もし OSS プロジェクトがビルド サービスを持っていないのであれば、ビルド サービスとその結果の安全性は関係ありません。

図 13

プロジェクトは、以下の構築のベストプラクティスをサポートしていますか？

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q41、サンプル数 = 72、分析から「知らない・よくわからない」を除外

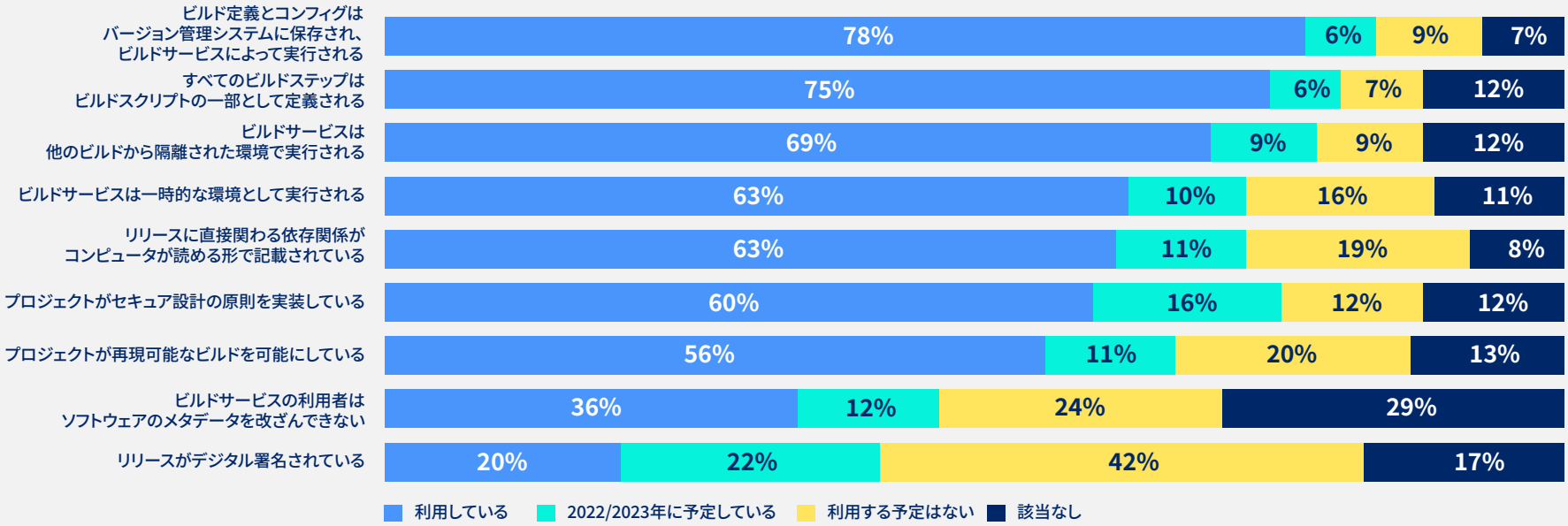
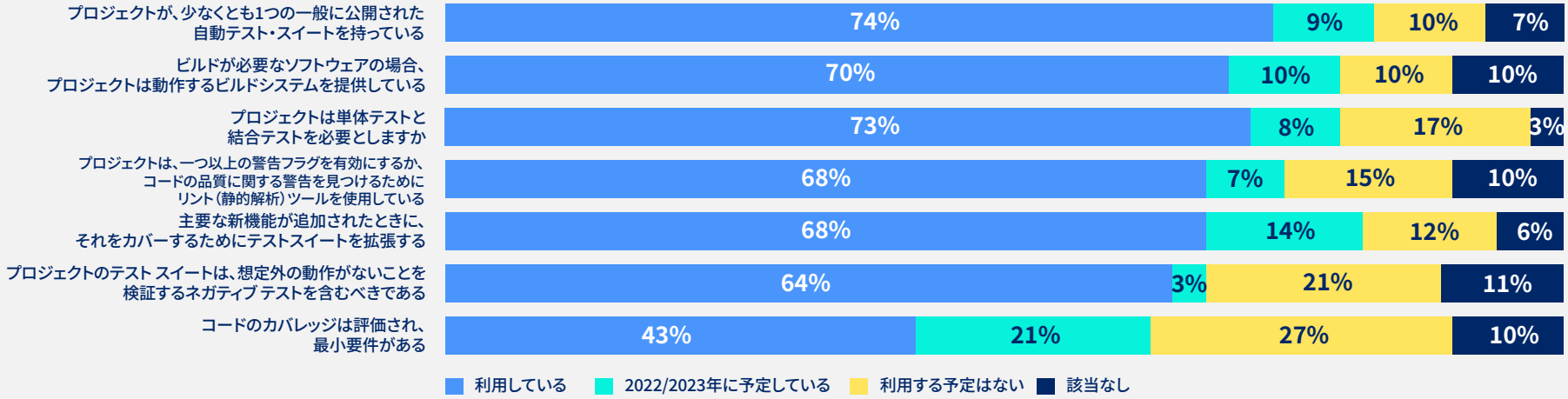


図 14

プロジェクトは、以下の品質に関するベストプラクティスをサポートしていますか？

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q42、サンプル数 = 72、分析から「知らない・よくわからない」を除外



ソフトウェア品質のベストプラクティスは、メンテナーとコア コントリビューターによって十分にサポートされている

ソフトウェアの品質は、いくつかのソフトウェア開発プロセスにおいて常に重要な目標です。先に述べたように、SDLC においてバグを早く発見できれば、解決するのに安い費用で済みます。図 14 は、メンテナーとコア コントリビューターが SQA（ソフトウェア品質保証）のベストプラクティスの選択にどの程度取り組んでいるかを示しています。結果は肯定的で、特に、図 14 に示されたものを含む、SQA に対するより従来のアプローチに対して肯定的です。採用率を「現在使用中」から「現在使用中＋計画中」までの範囲と見なした場合、より従来型の SQA のベストプラクティスの採用率は、公開テストスイート（74 ～ 83%）、動作するビルド システムの提供（70 ～ 80%）、単体テストと統合テストの要件（73 ～ 81%）、コード品質警告のためのコンパイラ フラグ(68～75%)、機能の拡張に伴うテスト スイートの拡張（68 ～ 82%）について、非常に良好です。これは、**OpenSSF Compiler Options Hardening Guide for C and C++** のような、これらの言語でソフトウェアを開発する際の広範なガイダンスを提供する取り組みにとって良い兆候です。

図 14 の想定外動作の評価には、ネガティブ テスト（ソフトウェアがすべきでないことをしないことを検証する）が含まれています。多くのメンテナーとコア コントリビューターがこれを採用しています（64%）が、まだ採用していないプロジェクトに追加する計画はほとんどありません。コード カバレッジの最小要件の採用率は控えめですが、採用率が 50% 増加する予定であることから、このベストプラクティスは支持されつつあります。コード カバレッジは、テストされたコードの量（ステートメントの割合や、ブランチの割合など）を測定するもので、テストが不十分な場合に定量的な証拠を提供することができます（多くのコード ステートメントやブランチが完全にテストされていない場合、テスト プロセスは必然的に多くの問題を見逃すことになります）。

より注意が必要なエリアのセキュリティ

最近のソフトウェア開発において、セキュリティは重要な要素です。SolarWinds 社の Orion に対する攻撃は、攻撃者がソフトウェアの構築プロセスの破壊を含め、ミッション クリティカルなシステムを盗聴、制御、または混乱させようとする努力において、非常に巧妙かつ大胆になる可能性があることを私たちに示しました。このため、私たちは、図 15 に示すソフトウェア セキュリティの基本要素に対処するためのベストプラクティスを検討しました。

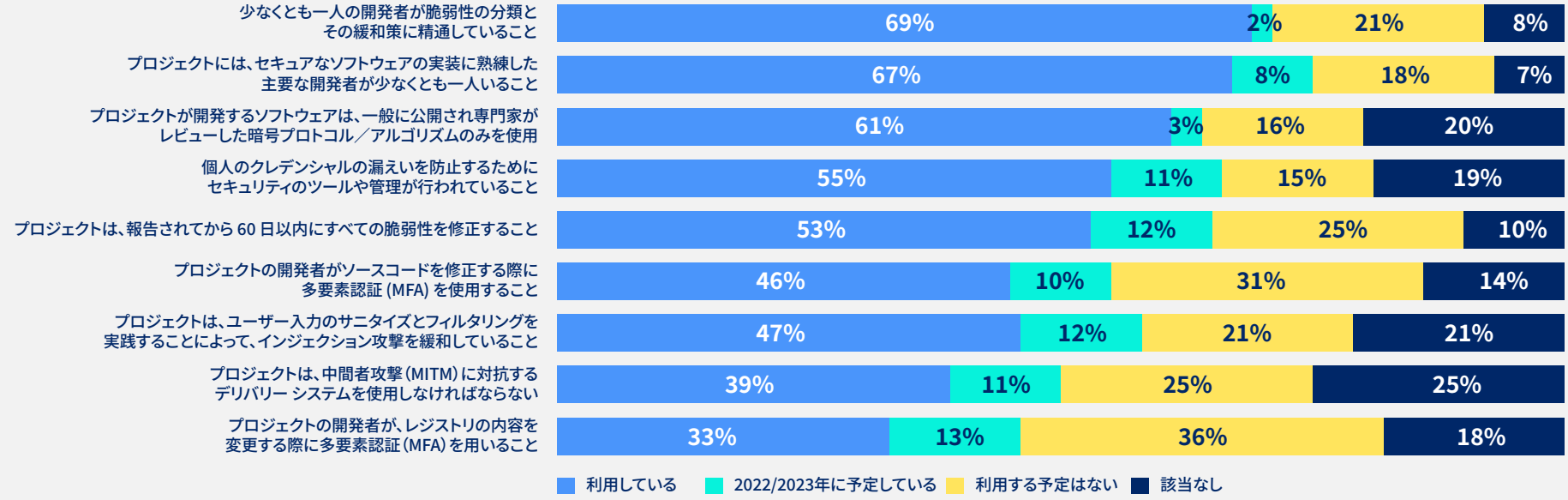
これらの各セキュリティのベストプラクティスをすでに採用している、または採用する予定のメンテナーとコア コントリビューターの組み合わせを考慮しても、この結果には改善の余地が残されています。図 15 で最も高い基準値に達したのは、セキュアなソフトウェアの実装に熟練した開発者が少なくとも 1 人いるプロジェクトで、75%でした。今日、ほとんどのメンテナーとコントリビューターは、個人認証情報の漏えいを防ぐためにセキュリティ ツールの使用をほとんど採用しておらず (55 ～ 66%)、報告後 60 日以内に脆弱性を修正していません (53 ～ 65%)。しかし、メンテナーを擁護するために言えば、すべての CVE がタイムリーに修正する必要があるわけではありませんが、重要度が中程度または高い CVE は、おそらくより早急に注意を払う必要があります。

ソース コードを変更する際に多要素認証 (MFA) を使用する (46% ～ 56%)、レジストリのコンテンツを変更する際に MFA を使用する (33% ～ 46%) といった

図 15

プロジェクトは、以下のセキュリティのベストプラクティスをサポートしていますか?

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q43、サンプル数 = 72、分析から「知らない・よくわからない」を除外



ベストプラクティスは、改善の余地が大きい分野のようです。2022 年の前半、さまざまな機能を提供する Forge やパッケージ リポジトリが、特定のケースで MFA を要求する方向に動き始めました。当時、多くの人がこれを見ることに興奮を示しましたが、不必要な負担と見て抵抗する人もいました。**OpenSSF は、MFA に向けた動きを明示的に支持し**、今日の攻撃者に対抗するために MFA が必要になりつつある理由を説明している、と投稿しています。この調査の時点から、MFA の利用は増加しています。**GitHub がコードへの貢献に 2FA を要求し始める** 2024 年以降、MFA の利用はさらに大幅に増加すると予想されます。

セキュリティ ツールが示すメンテナーの手動レビューへの傾向

OSS コントリビューターが、SCA と SAST ツールをよく使っていることは、すでに見ました（図 9）。同時に、その他の OSS コントリビューターは、他のセキュリティ テスト ツールに強い関心を持っていることがわかります。メンテナーとコア コントリビューターは、手作業によるコード レビューにより依存しています。このように手作業によるコード レビューが重視されるのは、メンテナーとコア コントリビューターが、自分たちの作業におけるサイバー セキュリティを修正、改善、対処する方法を包括的な形で理解する必要があるためです。

図 16 は、メンテナーとコア コントリビューターが、中程度または高程度の CVE をタイムリーに解決すること（55 ～ 67%）、SCA ツールの使用（42 ～ 61%）、SAST ツールの使用（41 ～ 51%）に重点を置いていることを確認するものです。しかし、メンテナーとコア コントリビューターが CSA と SAST ツールから得るサポート以上に、他のツールの使用は限られています。手作業によるコード レビューに時間を投資し、主要なツールを使用して補うことは、サイバー セキュリティのこの側面に対処するための受け入れ可能なアプローチであると思われるため、これは必ずしも問題であるとは考えていません。

図 16

プロジェクトは、以下のセキュリティテストのベストプラクティスをサポートしていますか？

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q44、サンプル数 = 72、分析から「知らない・よくわからない」を除外

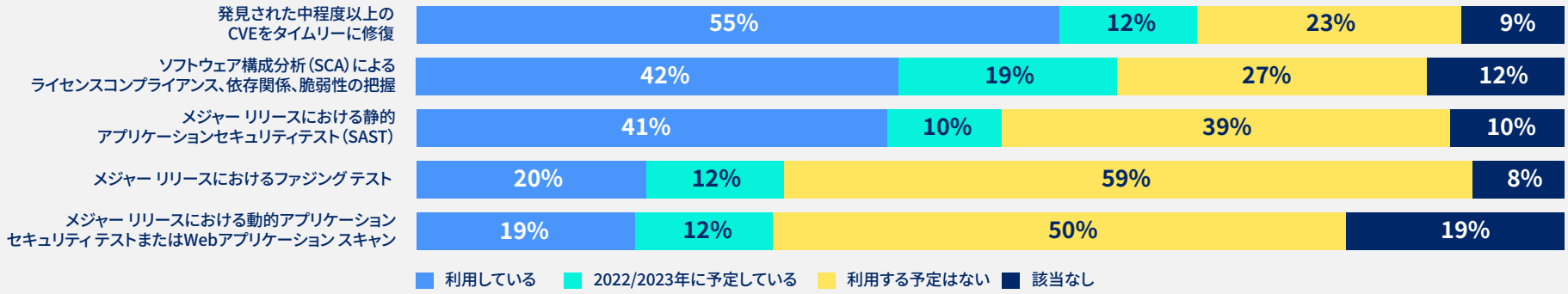
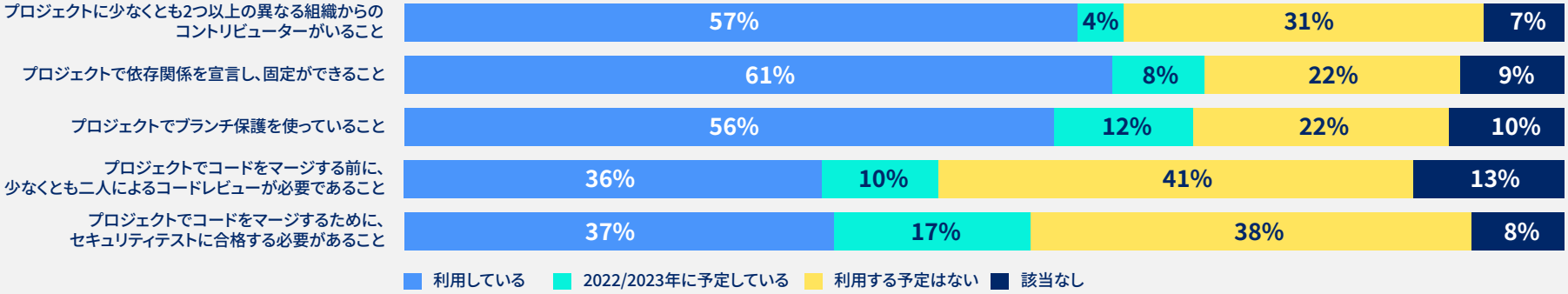


図 17

プロジェクトは、以下のセキュアなソース コーディングのベストプラクティスをサポートしていますか？

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q45、サンプル数 = 72、分析から「知らない・よくわからない」を除外



セキュア コーディング原則のサポートは、管理の容易さによって異なる

Linux Foundation と OpenSSF は、セキュアなソフトウェア開発のためのベストプラクティスに重点を置いています。セキュアなソフトウェア開発プロセスには、セキュアなソフトウェアの実装、いわゆるセキュア コーディングが含まれます。セキュアなソフトウェアを開発する際には、他のプロセス（セキュアな設計や検証など）も重要ですが、セキュアにソフトウェアを実装することが極めて重要です。図 17 は、メンテナーとコア コントリビューターが、特定のセキュア コーディングの実践方法を広く採用していることを示していますが、まだ改善の余地があります。

多くのメンテナーとコア コントリビューターが支持しているベストプラクティスには、2つの異なる組織からのコントリビューターを持つプロジェクト（57〜61%）、依存関係を宣言して固定する機能（61〜69%）、ブランチ保護の使用（56〜68%）などがあります。「異なる2つの組織からのコントリビューター」を持つことは、1つの組織が完全にコントロールできるわけではないことに注意してください。それを奨励し措置を講じることはできますが、最終的には、別の組織が参加するかどうかを決定しなければなりません。

あまり支持されていないセキュア コーディングの実践方法としては、少なくとも 2 人によるコード レビューが必要（36 〜 46%）、コードをマージする前にプロジェクトがセキュリティ テストに合格する必要がある（37 〜 54%）などがあります。2 人によるコード レビューは評価に値する目標です、しかし、その必要性は、他の人が対応できるかどうかや、マージされるコードの範囲や複雑さによって多少左右されます。**Josh Bressers による 2023 年の分析**によると、NPM のリリースの半分以上は、たった 1 人でメンテナーをしています。プロジェクトに 1 人しかいない場合、2 人でのコードレビューはできません。コードをマージする前にセキュリティ テストを避けることは、悪いアイデアに見えます。しかし、プロジェクトの特性や制約が邪魔をすることがよくあります。すべてのメンテナーやコア コントリビューターが、セキュリティ テストを実施する技術や専門知識を持っているとは限りませんし、プロジェクトの規模や範囲が、厳密なセキュリティ テストを必要としないかもしれません。プロジェクトによっては、マージ前のレビューではなく、マージ後のレビューに依存することもあり、継続的デリバリーのニーズが優先されることもあります。何が（いつ）テストが必要なのかに関するメンテナーの知恵が重要なるのです。

ソフトウェアのセキュリティと持続可能性を向上させる方法に関するオープンソース コントリビューターの視点

ソフトウェア セキュリティの向上は、IT 業界や政府にとって重要かつ継続的な活動であり、最重要課題です。このセクションでは、ソフトウェアのサプライ チェーン、セキュアなソフトウェア開発、および OSS の持続可能性を改善する方法に関する OSS コントリビューターの視点について見ていきます。

オープンソース ソフトウェアのサプライ チェーンのセキュリティを改善するオープンソース コントリビューターのガイダンス

図 18 によると、OSS のサプライチェーン セキュリティを向上させるための主要なアプローチは、ソフトウェア セキュリティ ツールのインテリジェント化（58%）、自動化の促進（54%）、セキュアなソフトウェア開発のベストプラクティスに従うこと（52%）、OSS 雇用者のインセンティブ向上（49%）、セキュリティ監査（47%）、ソースコードのピア レビュー（41%）などです。

OSS コントリビューターは、セキュリティ ツール、特に SCA と SAST ツールを広く使用しており、全体で 58%でした。その他の OSS コントリビューターは、IaC と DAST ツールを好んで使用しています。セキュリティ ツールは、脆弱性の自動検出、ライセンス コンプライアンス、脆弱性の侵入を防ぐ継続的インテグレーション機能、コード品質の向上、バグの早期検出と修正、コントリビューターによるセキュリティ問題の特定、優先順位付け、解決の容易化など、多くの利点を提供します。

セキュリティを侵害する可能性のある経路を排除するための自動化の強化（54%）により、市場投入までの時間と開発者の疲労が軽減され、手作業によるタッチポイントがなくなるため、悪意のある行為者が利用できる攻撃対象となる領域が減少します。自動化は、メンテナーやコントリビューターの負担を増やすことなく、その作業能力を向上させる手段でもあります。ある回答者は、よりインテリジェントなツールと自動化、および標準化によって、透明性の問題を解決できると指摘しています。人々にワーク フローを変更させるよりも、自動的に、あるいは「デフォルトで」セキュリティを提供することが望ましいでしょう。しかし、他の回答者は、自動化ツールにできることは限られており、調査が必要となる深い問題がある場合もあると指摘しています。さらに、SBOM 情報の統合や、ユーザーが常に使用するようなツールの使いやすさを実現するなど、ツール同士がうまく連携できるようにするために必要な作業がたくさんあります。

セキュアなソフトウェア開発のための包括的なベストプラクティスに従うこと（52%）から、SDLC 全体にわたってソフトウェア セキュリティに取り組むための、信頼され、公開されたベストプラクティスのコレクションを持つことに価値がるということを確認できます。Linux Foundation と OpenSSF は、すでに、セキュアなソフトウェア開発のためのベストプラクティスの包括的なリストを作成しており、このレポートの冒頭で、セキュアなソフトウェア開発に関する無料のトレーニング コースとベストプラクティスをどこから利用するかを示しています。

雇用主によるインセンティブの増加（49%）は、従業員が就業時間中に組織にとって重要な OSS プロジェクトに取り組めるようにすることや、メンテナシップ（メンテナーとしての役割、活動）やコミュニティへの参加に関連する追加的なインセンティブを提供することを意味します。これは、持続可能性への重要な手段です。また、組織が OSS（多くの場合、極めて依存している OSS）に「恩返し」することを支援するものであり、OSS のバリュー プロポジションを持続、拡大させる重要な手段でもあります。最後に、回答者は、雇用主や企業が OSS の改善のためのリソースの提供に大きな役割を果たすことができると指摘しました。ある回答者は、オープンソースの恩恵を受けられるエンド ユーザー企業がメンテナーに資金を提供することを提案しました。多くの場合、メンテナーにサポート料を支払うビジネス ケースはないようなので、企業や組織の OSPO は、戦略的なレベルで考え、組織が依存する OSS をサポートする上で大きな役割を果たすことができます。

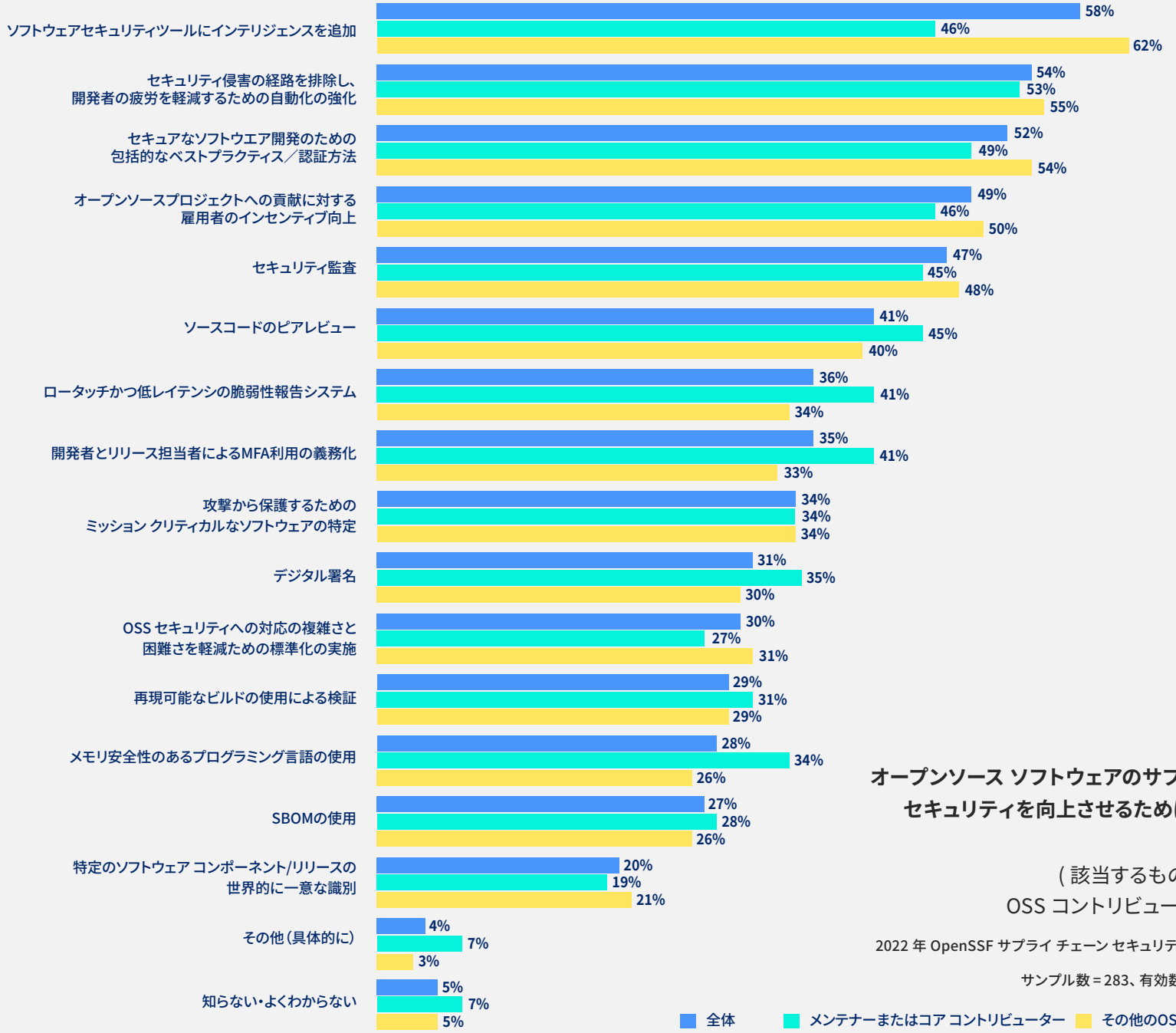


図 18

オープンソース ソフトウェアのサプライチェーンのセキュリティを向上させるために重要な活動はどれですか？

（該当するものをすべて選択）

OSS コントリビューターのタイプ別

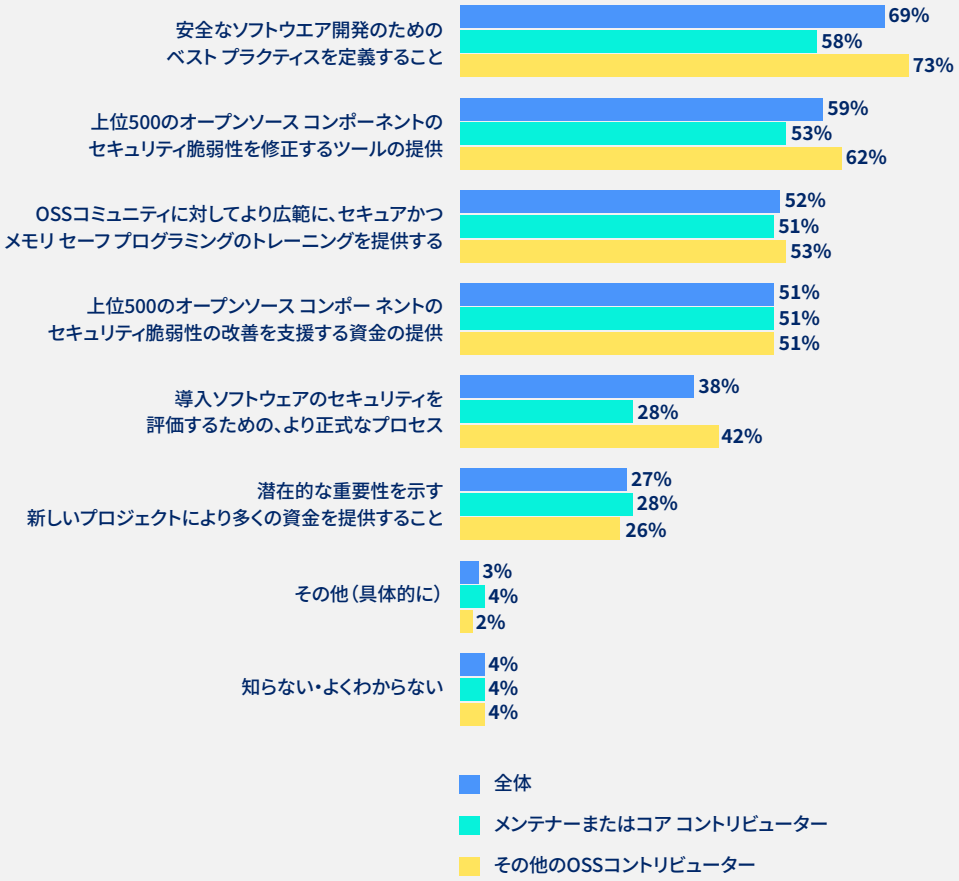
2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q48 x Q14a、

サンプル数 = 283、有効数 = 283、総数 = 1,645

図 19

IT 業界の組織がオープンソースソフトウェアの開発におけるセキュリティを改善する方法には、どのようなものがありますか？(該当するものをすべて選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q50 x Q14a、サンプル数 = 271、有効数 = 283、総数 = 821



Linux Foundationの過去の調査によると、セキュリティ監査は、非公式かつ散発的に行われることが多いようです。そのため、図18に示すように、セキュリティ監査 (47%) が認知されつつあることは喜ばしいことです。セキュリティ監査は、主要なコンポーネントのセキュリティを評価する有用な方法であると同時に、これらのコンポーネントをセキュアにするための重要なプロセスでもあります。

最後に、ソースコードのピア レビュー (41%) は、OSS コントリビューターにとって重要です。なぜなら、コンポーネントの機能とセキュリティを理解するために時間をかけることは、コードの修正、変更、追加が慎重かつ安全に行われ、結果として高品質なコンポーネントが得られるようにするための最も効果的な方法だからです。

オープンソース開発におけるセキュリティ向上のためのコントリビューター ガイダンス

質問の焦点を、サプライ チェーン全体で OSS の開発を改善する方法に絞ると、いくつかの重複する傾向と、いくつかの新たな発見が見られます。図 19 によると、OSS コントリビューターからの主な指導は、セキュアなソフトウェア開発のベストプラクティスを定義することです (69%)。セキュアな開発のためのベストプラクティスの定義とトレーニング コースは、多くの OSS コントリビューターが知らないリソースであるようです。これらのリソースへのリンクについては、本レポートのはじめにをご覧ください。また、多くの回答者は、IT 組織もベストプラクティスの定義に取り組むことができると指摘しています。しかし、回答者の中には、企業が策定したガイドラインが長くなりすぎたり、煩雑になったりする可能性があり、メンテナーがそれを読んだり、実践したりしないかもしれないという懸念を表明する人もいました。また別の回答者は、大規模な組織が集まって、プロジェクトのセキュリティ確保に関して何がうまくいき、何がうまくいかなかったかについて話し合うことは有益だと述べています。

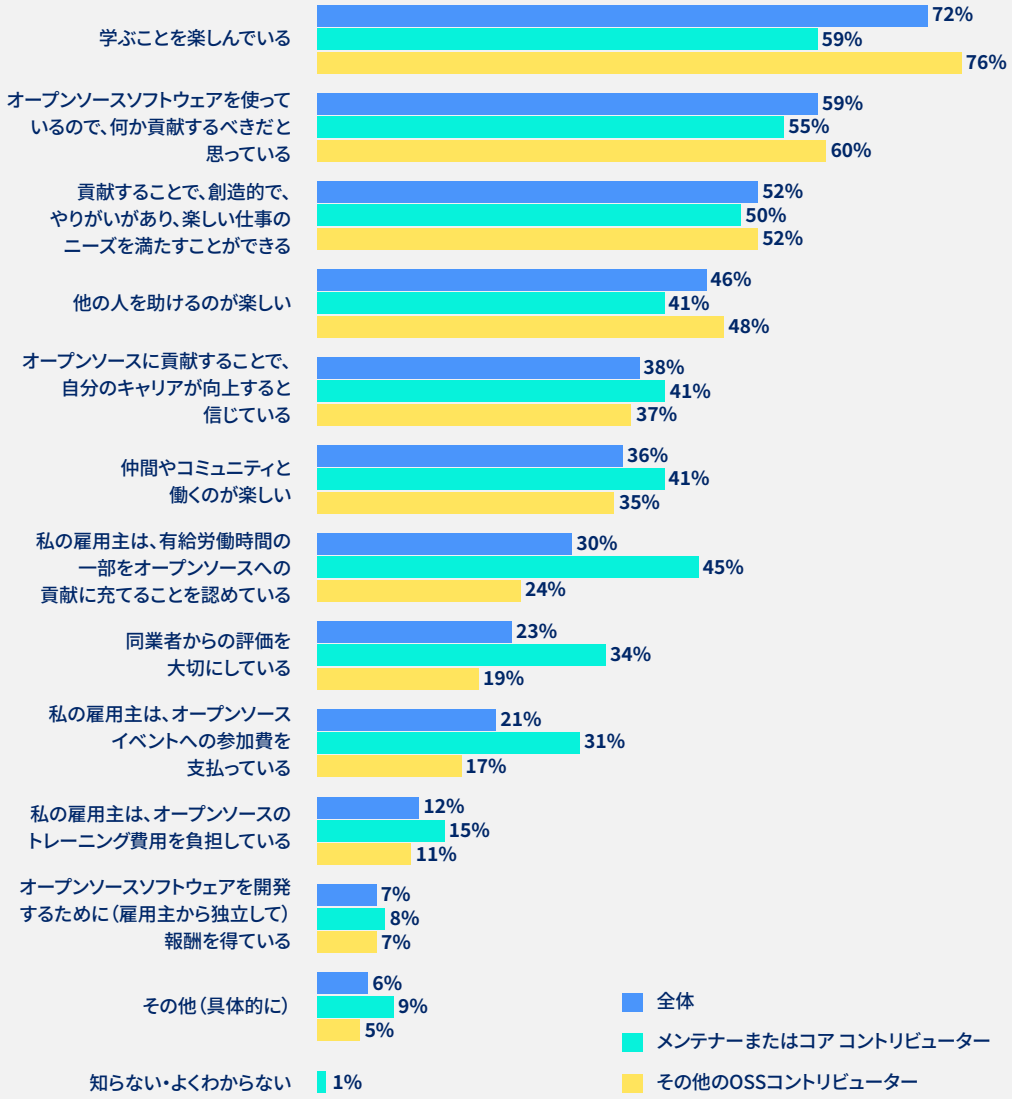
図 19 の回答の 2 番目の傾向は、上位 500 の OSS コンポーネントの脆弱性を修正するツールの提供 (59%)、セキュアおよびメモリ セーフ プログラミングのトレーニングの強化 (52%)、上位 500 の OSS コンポーネントの脆弱性を修正するための資金提供 (51%) などです。

図 20

OSS プロジェクトやコンポーネントの維持や貢献の原動力は何ですか？

(該当するものをすべて選択) OSS コントリビューターのタイプ別

2022 年 OpenSSF サプライ チェーン セキュリティ調査、Q51 x Q14a、サンプル数 = 271、有効数 = 283、総数 = 1,088



Alpha-Omega プロジェクトは、最も重要なオープンソース プロジェクトを特定し、安全性を確保することを目的とした OpenSSF 内の取り組みです。このプロジェクトの Alpha の部分は、最も広く使われている重要なオープンソース プロジェクトを積極的にセキュアにすることです。このプロジェクトの詳細については、**Alpha-Omega—Open Source Security Foundation (openssf.org)** を参照してください。

メモリ セーフなプログラミング言語のトレーニングを増やすことも、OSS のセキュリティを向上させる素晴らしい方法です。Rust（所有モデル）と Swift（自動参照カウント）は、メモリ管理に厳密なアプローチを取っており、C や C++ と比較してメモリ管理技術に基づく「より安全な」言語であり、なおかつ優れたパフォーマンスを提供します。Go、Java、JavaScript、C#、Python を含むほとんどの一般的な言語は、自動ガベージコレクションを含むことによって、部分的にメモリ安全性を提供しています。これは、メモリ管理の複雑さの多くを抽象化し、開発者が簡単に管理できるようにします。言語の選択は、パフォーマンスのニーズ、エコシステムの考慮事項、開発者の専門知識など、プロジェクトの特定の要件に依存します。メモリ セーフのプログラミング言語を選択することで、C や C++ のような言語で開発する場合に悩まされる多くの脆弱性を自動的に防ぐことができます。

オープンソースの持続可能性を向上させるためのオープンソース コントリビューターのガイダンス

図 20 は、個人的な信念と利他的な信念が OSS コントリビューターを動かしていることを示しています。個人的な面では、OSS コントリビューターは学ぶことを楽しんでいます（72%）。貢献することで、創造性、挑戦、楽しみの欲求を満たすことができます（52%）。利他的なレベルでは、OSS コントリビューターは、OSS を使用し、何かお返しをするべきだと感じており（59%）、OSS への参加はその他の人々を助ける方法であると考えています（46%）。

また、メンテナーやコア コントリビューターは、その他の OSS コントリビューターと比較して、OSS への貢献がキャリア アップにつながることを重視しています（41%）。彼らは仲間や OSS コミュニティと仕事をすることに楽しみを見出しており（41%）、雇用主は勤務時間中に OSS に取り組むことを許可することが多いです（45%）。

オープンソースの持続可能性の問題を解決するのは複雑で、多角的なアプローチが必要です。利他的な動機がオープンソース プロジェクトを推進することが多いため、継続的なメンテナンスと限られたリソースの中での開発における必要性のバランスを取ることは困難です。企業の管理責任やパートナーシップ、雇用主による資金提供や財政支援は、新しく歓迎すべき現象ですが、OSS コントリビューターが OSS やコミュニティとの関わりから得られる満足感を損なうようなものであってはなりません。オケージョナル OSS コントリビューターからメンテナーへの道が存在するように、コミュニティへの参加と成長をサポートすることは重要な目的です。認知、報酬、そして確立された行動規範を備えた健全なコミュニティ文化を構築することで、コミュニティへの参加を確実に支援することができます。最後に、このような個人的、組織的、文化的なニーズに対応しながら、プロセス モデルを進化・発展させ続け、OSS のセキュリティと品質を向上させる一方で、参加意欲を削ぐのではなく、むしろ奨励するような形でバランスを取ることが、絶対に最も重要です。

結論

OSS エコシステムのセキュリティを向上させる取り組みを進めると同時に、OSS のメンテナーが製品やビルド プロセスにセキュリティを組み込めるようにし、定型的な作業を自動化してメンテナーの労力を減らすべきです。この調査が、どのような実践方法が有効で、今後どのような取り組みが必要かを明らかにする一助となれば幸いです。本レポートで分析したデータに基づく主な結論は以下のとおりです。

オープンソースの持続可能性について考える際には、オープンソースのメンテナーの視点を念頭に置く

ソフトウェア開発は人間が行うものです。たとえ AI や ML がソフトウェアの生成に役立ったとしても、人間が何をすべきかを定義し、結果をレビューして修正しなければなりません。ソフトウェア開発は問題解決でもあり、開発者やメンテナーは、セキュリティなどの制約に縛られることが多いなかで機能的な問題を解決しなければなりません。問題を解決するためには、OSS のコントリビューター（メンテナーであれ、コア コントリビューターであれ、オケージョナル コントリビューターであれ、ワンタイム コントリビューターであれ）は、意思決定や行動を起こす前に、既存のコードをある程度理解する必要があります。プロジェクトの規模が大きくなると、コードのサイズや複雑さが増すため、コードを理解することが難しくなります。したがって、手作業によるコード レビューは、メンテナーやコントリビューターが行う重要な要素です。この調査から得られたポジティブな発見は、OSS のメンテナーとコア コントリビューターがコード レビューとピア レビューに高い優先順位を置いていることでした。メンテナーとコア コントリビューターにとってのこれらの活動の重要性は、一部の人が予想していたよりも大きいものでした。別の言い方をすれば、セキュリティを改善するための取り組みには、このような声も含める必要があるということです。セキュリティ対策は集団的な取り組みであり、共同作業を優先しなければなりません。

OSSのサイバー セキュリティは、OSSコミュニティがうまく対処しなければならない重要な問題です。セキュリティにギャップや不備があり、それが名誉を傷つけたり、金銭的な損害を与えたり、不正にデータを流出させたりすることは、OSSのイメージダウンにつながるだけではありません。過去の調査によると、OSSのイメージは、コミュニティ主導の開発アプローチから利益を得ており、OSSコンポーネントのセキュリティがプロプライエタリなコードよりも優れている機会を提供していると示しています。OSSコミュニティは、この期待に応え、コミュニティ文化に取り返しのつかない損害を与えることなく、セキュリティのニーズに対処する方法を見つけなければなりません。

オープンソースにおけるサイバー セキュリティに対処するには、セキュリティ ツールの使用と自動化が重要

本レポートを通して、OSS コントリビューターによるセキュリティ ツールの使用は、サイバー セキュリティのニーズに対応するための重要な要素となっています。SAST と SCA ツールは、多くの OSS コントリビューターに支持されていますが、一部の OSS コントリビューターは、DAST や IaC などのその他のセキュリティ ツールの使用にも前向きです。これら 4 つの機能的なツール カテゴリーをすべて使用することは、セキュリティ テストと自動化に取り組むための優れた基盤を提供することになるでしょう。

ツールは、個々の開発者の負担を軽減し、エコシステム全体の成功を支援します。ツールは、透明性とセキュリティの両方を向上させることができますが、メンテナーの負担になるべきではありません。むしろ、ツールは開発者の既存のワークフローと連携し、セキュリティ機能をデフォルトで提供すべきです。また、異なるツールを連携させるための相互運用性への投資も必要です。そして、現実的に考えて、ツールや自動化がすべての問題を解決できるわけではないことを理解しなければなりません。

教育とベストプラクティスは解決における重要な部分

本レポートの大部分では、メンテナーとコア コントリビューターが実践しているベストプラクティスを紹介しています。図 11 から図 17 は、セキュアなソフトウェア開発のためのベストプラクティスのメンテナーの使用状況と実施計画を示しており、いくつかのベストプラクティスが広く採用されていることを示しています。他の多くのベストプラクティスは、2023 年末までに広く普及する可能性があります。メンテナーとコア コントリビューターは、ベストプラクティスを知らないわけではありませんが、OSS サプライ チェーンのセキュリティ（図 18）とソフトウェア開発のセキュリティ（図 19）を改善する方法について尋ねたところ、ベストプラクティスへの支持が常に主要な回答でした。このことは、Linux Foundation が、セキュアなソフトウェア開発のトレーニングコースと同様に、SDLC 全体にわたるセキュアなソフトウエアを開発するためのベストプラクティスの広範囲なリストをすでに定義していることを、多くの OSS コントリビューターが知らないという知識のギャップがあるかもしれないことを示唆しています。これらのリソースへのリンクについては、このレポートのはじめにを参照してください。このことは、これらのリソースを知ってもらうための、より多くのマーケティングの必要性を示唆しています。

Robert Scholte が述べているように、「自分の知らないことはわからない」です。より広く言えば、ユーザーやメンテナーが使用できるツールや改善策はありますが、まず、それらとその使用法を認識する必要があります。私たちの調査に対する「知らない・よくわからない」（DKNS）という回答が非常に多いことも、利用可能なセキュリティ ツールについて教育することが良い第一歩であることを示しています。最後に、Brian Demers によると、コンピューター サイエンス（CS）、ソフトウェア エンジニアリング（SwE）など、現在の学部レベルのソフトウェア開発教育では、セキュリティに触れておらず、これは大きな問題です。セキュアなソフトウェアを開発する方法の基本は、CS、SwE、および同様の分野のすべての学部カリキュラムの一部であるべきです。

調査方法

この調査について

2022 年 3 月に Linux Foundation Research とそのパートナーによって実施された Web 調査がこの調査の基礎となっています。この調査の目的は、オープンソースのサプライ チェーンに関するセキュリティの状況についてグローバルな視点を提供することでした。2022 年 6 月に発行された最初のレポート、「Addressing Cybersecurity Challenges in Open Source Software（オープンソース ソフトウェアにおけるサイバー セキュリティの課題への取り組み）」は、この調査から得られた重要な結果をいくつか紹介しています。しかし、この最初のレポートには、リソース、長さ、時間の制約のため、セキュアなソフトウェア開発のためのベストプラクティスの採用に関するメンテナーとコア コントリビューターからのデータを収集した調査のセクション全体は含まれていませんでした。2023 年 12 月に発行されたこの追加レポート「Maintainer Perspectives on Open Source Software Security（オープンソース ソフトウェアセキュリティに関するメンテナーの視点）」は、このベストプラクティスのデータを提供し、OSS コントリビューターの区分に基づいて追加のコンテキストを含んでいます。この 2 つのレポートは、統計情報以外は重複していません。

このセクションでは、調査方法と回答者の統計情報を紹介します。調査の観点からは、サンプルのバイアスに対処し、高いデータ品質を確保することが重要でした。私たちは、Linux Foundation メンバーシップ、パートナー コミュニティ、ソーシャル メディア、サード パーティーのパネル プロバイダーから使用可能なサンプルを調達することで、サンプルのバイアスを回避しました。また、回答者が所属する組織の代表として質問に正確に回答できるよう、十分なオープンソースに関する知識と専門的な経験を有していることを確認するため、広範な事前スクリーニング、スクリーニング基準、およびデータ品質チェックを通じて、回答データの品質を確保しました。

オープンソース サプライチェーン セキュリティ調査は、55 の質問から構成され、以下の項目に焦点を当てました。

- 一般的なソフトウェア セキュリティ
- OSS ソフトウェア セキュリティ（セキュアなソフトウェア開発のためのベストプラクティスの採用に関する質問を含む）
- OSS ソフトウェアのセキュリティ改善方法

エンドユーザー企業、IT ベンダーやサービス プロバイダー、非営利団体、学術機関、政府機関から調査データを収集しました。回答者の業種や企業規模は多岐に渡り、調査対象地域は南北アメリカ、ヨーロッパ、アジア太平洋地域です。

2022 年 3 月に 2022 年オープンソース サプライ チェーンセキュリティ調査を実施しました。このデータは古くなったとはいえ、メンテナーが直面する課題や、セキュアなソフトウェア開発への取り組み方に関する意思決定について有益な知見を提供しています。本レポートで取り上げたサンプル数の内訳は以下の通りです。

1. 本調査では、1,175 名の回答者が調査を開始しました
2. スクリーニング基準（Q13,5,6）により 437 名の回答者が除外され、738 名の回答者が残りました
3. この 738 名のうち、分類に関する質問を完了し、サプライチェーンのセキュリティに関する質問に回答したのは 539 名のみでした
4. この 539 人のうち、441 人がオープンソースのメンテナー、コア コントリビューター、オケージョナルコントリビューター、ワンタイム コントリビューター、その他のコントリビューターであると自認しています
5. この 441 人のうち、オープンソースのメンテナーまたはコア コントリビューターを自認する回答者はわずか159人で、その他のOSSコントリビューター（オケージョナル コントリビューター、ワンタイム コントリビューター、その他のコントリビューター）は 282 人でした
6. この 159 名のうち、セキュアなソフトウェア開発にどのように取り組んでいるかという具体的な質問に回答したのは 72 名のみでした

本調査では、回答者にほぼすべての質問に回答していただくことを義務付けていますが、回答者の役割や経験の範囲外であるために回答できない場合もありました。このため、ほぼすべての質問について、回答リストに「知らない・よくわからない（DKNS：Don't know, Not Sure）」の回答を追加しました。しかし、この場合、DKNS の回答をどのように解釈すべきかという問題が生じます。

1 つのアプローチは、DKNS を他の回答と同様に扱い、質問に対して DKNS と回答した回答者の割合を知ることです。このアプローチの利点は、収集したデータの正確な分布を報告できることです。このアプローチの問題点は、有効な回答、つまり、回答者が質問に答えることができた場合に収集された回答の分布を歪めてしまう可能性があることです。

本レポートの分析の一部は、DKNS の回答を除外しています。これは、(a) DKNS を除いた回答の分布を理解することが重要でこと、(b) DKNS のデータが無作為に欠落（MAR：missing at random）または完全に欠落（MCAR：missing completely at random）のどちらかに分類できることより決定されます。

質問から DKNS のデータを除外しても、他の回答のカウント分布は変わりませんが、残りの回答全体の回答の割合を計算するために使用する分母のサイズは変わります。これにより、有効回答の割合が比例して増加します。DKNS のデータを除外することを選択した場合は、図の脚注に「DKNS の回答は除外」と記載しています。

四捨五入のため、本レポートのパーセンテージの合計が 100% にならない場合があります。

Data.World へのアクセス

Linux Foundation Research では、各実証プロジェクトのデータセットを Data.World で公開しています。このプロジェクトのデータセットには、調査の実施方法、生の調査データ、スクリーニングとフィルタリングの基準、調査の各質問の度数表が含まれています。このプロジェクト（2022 年 OpenSSF サプライ チェーン セキュリティ調査）を含む Linux Foundation Research のデータセットは、**data.world/thelinuxfoundation** で見つけることができます。

謝辞

セキュアなソフトウェア開発の状況について、自らの洞察と経験を共有してくれた調査参加者全員に感謝します。査読担当者、Linux Foundation の同僚、そして、多数のメンテナーの方には、この調査のさまざまな段階での視点を持ち、この調査に関わっていただいたことに特に感謝します。Omkhar Arasaratnam、Stephen Augustus、Brian Behlendorf、Hilary Carter、Brian Demers、Adrienn Lawson、Kim Lewandowski、Oleg Nenashev、Nick O'Leary、Jed Salazar、Robert Scholte、Daniel Stenberg、Kate Stewart、Liran Tal、Harry Toor、Dana Wang、David A. Wheeler、そして、Adolfo Garcia Vettia へ。

著者について

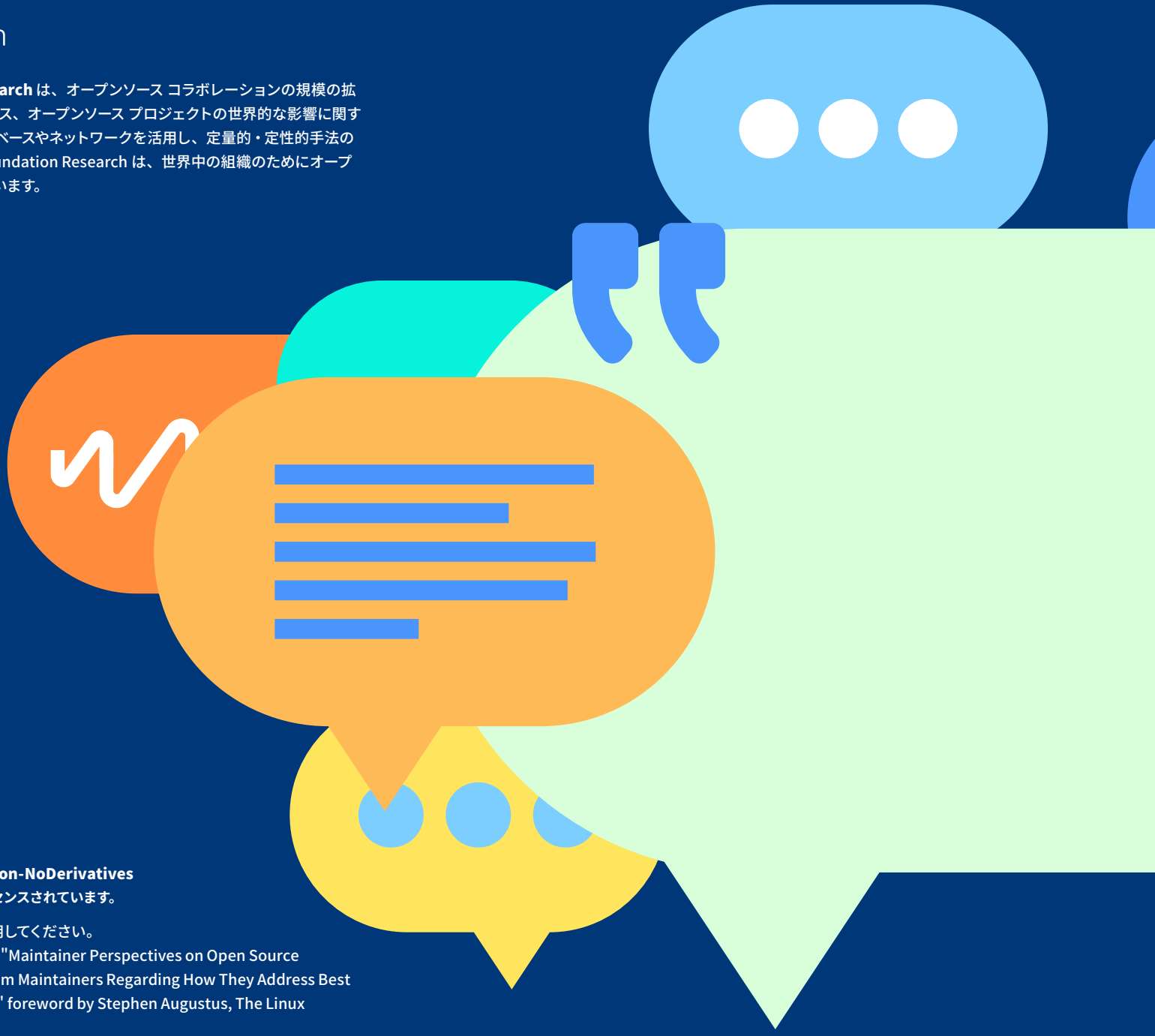
STEPHEN HENDRICK は、Linux Foundation の研究担当バイス プレジデントで、OSS が IT の生産者と消費者にとってイノベーションの原動力となることを理解する、Linux Foundation の中核となるさまざまな研究プロジェクトの主任研究員を務めています。スティーブは、ソフトウェア業界のアナリストとして 30 年以上にわたって培ってきた一次調査の技術を専門としています。スティーブは、DevOps、アプリケーション管理、意思決定分析など、アプリケーション開発とデプロイに関するトピックの専門家です。市場ダイナミクスを深く洞察するさまざまな定量・定性調査手法の経験を生かし、多くのアプリケーション開発・導入領域で先駆けた調査を行ってきました。1,000 以上の出版物を執筆し、シンジケート調査やカスタム コンサルティングを通じて、世界有数のソフトウェア ベンダーや注目のスタートアップに市場ガイダンスを提供しています。

ASHWIN RAMASWAMI はオープンソースのメンテナ、開発者、政策の研究者です。また、ウェブ アプリケーション アーキテクチャとサイバー セキュリティ、Linux Foundation でのリサーチと執筆にも携わっています。スタンフォード大学でコンピュータサイエンスの学士号を取得し、ジョージタウン大学で法学博士号を取得中です。

この日本語レポートは、以下の文書の参考訳です。
Maintainer Perspectives on Open Source Software Security
翻訳協力：松本央

THE LINUX FOUNDATION | Research

2021 年に設立された **Linux Foundation Research** は、オープンソース コラボレーションの規模の拡大を調査し、新たな技術動向、ベストプラクティス、オープンソース プロジェクトの世界的な影響に関する洞察を提供しています。プロジェクトのデータベースやネットワークを活用し、定量的・定性的手法のベストプラクティスに取り組むことで、Linux Foundation Research は、世界中の組織のためにオープンソースの知見を提供するライブラリを構築しています。



この著作物を参照する場合は、以下のように引用してください。
Stephen Hendrick and Ashwin Ramaswami, "Maintainer Perspectives on Open Source Software Security: Survey-based Insights from Maintainers Regarding How They Address Best Practices for Secure Software Development," foreword by Stephen Augustus, The Linux Foundation, December 2023.