

2024年4月

STATE OF CONTINUOUS INTEGRATION & CONTINUOUS DELIVERY REPORT

The Evolution of Software Delivery Performance

継続的インテグレーション&継続的デリバリーの近況

ソフトウェア デリバリー パフォーマンスの進化

Created by



Sponsored by



CONTINUOUS DELIVERY FOUNDATION について

Continuous Delivery Foundation (CDF) は、最も急成長している継続的インテグレーション/継続的デリバリー (CI/CD) プロジェクトのベンダー中立的な拠点です。業界トップの開発者、エンドユーザー、ベンダー間におけるベンダー中立的なコラボレーションを促進し、CI/CD のベスト プラクティスと業界仕様をさらに推進します。その使命は、広範な成長を続ける継続的デリバリーエコシステムの一部であるプロジェクトを成長させ、維持することです。CDF とそのプロジェクトについては、<https://cd.foundation/> をご覧ください。



CDFメンバー - プレミア



CDFメンバー - ゼネラル



CDFメンバー - エンドユーザー



CDFメンバー - アソシエイト



このレポートのデータを共有できますか？

1. 使用許諾

このレポートは、[Creative Commons Attribution-NoDerivatives Licence 4.0 \(International\)](#) の下でライセンスされています。簡単に言えば、このライセンスの条件に従って、次のことが自由に行えます。

共有 — 商業目的および非商業目的で、レポートを複製したり、レポートの一部を 1 つ以上の文書または出版物に組み込んだりすることができます。

以下の条件が適用されます。

帰属 — このレポートのスポンサーであるSlashData™および Continuous Delivery Foundation のクレジットを適切に表示し、変更が加えられた場合はその旨を表示する必要があります。その場合、合理的な方法であればどのような方法でも構いませんが、>SlashData™ があなたやあなたの使用を支持していることを示唆するような方法は使用しないでください。

改変禁止 — レポートの内容をリミックスまたは改変することはできません。改変したコンテンツを配布することはできません。

2. 責任の制限

SlashData™は、本出版物に含まれる記述が信頼できると考えられる情報に基づいていると考えていますが、それが正確または完全であることを表明するものではなく、そのようなものとして依拠すべきではありません。表明された意見は、本出版物に掲載された日付現在の意見であり、ここに含まれる意見を含む情報は、予告なしに変更されることがあります。どのような目的であれ、第三者による本書の使用は、当該第三者が本書の内容を確認する上で十分な注意を払うことを免責するものではなく、また免責されるものでもありません。SlashData™は、商品性または特定目的への適合性を含むがこれに限定されない、すべての黙示的保証を否認します。

SlashData™、その関連会社、および代表者は、本書に基づいて行われた、または行われなかった決定、あるいは講じられた、または講じられなかった措置の結果として第三者が被った直接的、偶発的、特別、または結果的な損害または利益損失（もしある場合）について、一切の責任を負いません。

The analyst of the developer economy | formerly known as VisionMobile
SlashData © Copyright 2024 | Some rights reserved

この日本語文書は、以下のレポートの参考訳としてThe Linux Foundation Japanが便宜上提供しているものです。

[State of Continuous Integration & Continuous Delivery Report](#)



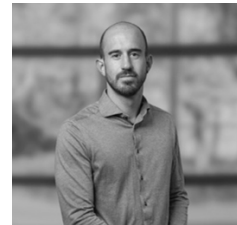
Liam Dodd

Senior Market Research Analyst

元反物質実験物理学者で、CERN在籍中に物理学博士号を取得。クラウド開発の変化、サイバーセキュリティ、技術開発と社会への影響の関心を持つ。

✉ liam.dodd@slashdata.co

著者について



Brayton Noll

Senior Market Research Analyst

Brayton Noll は気候変動と環境研究のバックグラウンドを持つ行動科学者。デルフト工科大学で計算社会科学の博士号を取得し、人間の行動ダイナミクスと気候適応に焦点を当てた論文を発表。データ分析の実務経験は5年。

✉ brayton.noll@slashdata.co

おもな考察	<u>7</u>	<u>8</u>	はじめに
誰がDevOpsに関心を持っているのか？	<u>10</u>	<u>16</u>	ソフトウェア デリバリーの パフォーマンスは どのように進化してきたか？
ソフトウェア デリバリー パフォーマンスの原動力	<u>23</u>	<u>39</u>	結論
付録	<u>42</u>	<u>44</u>	方法論



おもな考察

- DevOps の採用率は依然として高く、2024 年第 1 四半期時点で開発者の 83% が DevOps 関連の活動に携わっています。→
- ソフトウェア開発の経験が少ない開発者は、DevOps のプラクティスやテクノロジーの採用率が低くなります。→
- 継続的インテグレーションと継続的デリバリーは、DevOps 関連のアクティビティの中で 4 番目と 5 番目に人気があります。→
- ソース コントロール管理とイシュー トラッキングは、最も広く使用されている DevOps テクノロジーのトップの座を占めています。→
- コード変更のリードタイム、デプロイメント頻度、サービス復旧時間のトップパーフォーマーの割合は、2020年第3四半期に追跡を開始したときより減少していますが、現在は安定しているように見えます。→
- 各デプロイメント パフォーマンス指標におけるパフォーマンスの低いものの割合が増加しており、これは憂慮すべき傾向です。→
- 従業員数が 1,000 人を超える組織の開発者の間では、デプロイメント パフォーマンス指標は安定しています。→
- 開発者が使用しているDevOpsテクノロジーの数と、追跡した3つのパフォーマンス指標すべてにおいてトップ パフォーマーになる可能性との間には、強い相関関係があります。→
- CI/CD ツールを使用すると、すべての DORA 指標でデプロイメントパフォーマンスが向上しますが、マネージド CI/CD ツールとセルフホスト CI/CD ツールの両方を同時に使用している場合、その効果は最も大きくなります。→
- 同じ形式の複数の CI/CD ツールを使用すると、相互運用性に関連する課題が原因で、デプロイメント パフォーマンスが低下します。→

はじめに

01

1. はじめに

継続的インテグレーション (CI) と継続的デリバリー (CD) は、多くの場合、合わせて CI/CD と呼ばれ、開発者がコード変更を頻繁に統合し、ソフトウェア更新を確実かつ安全に頻繁にリリースできるようにするソフトウェア開発手法です。より広範な DevOps 文化的運動と密接に関連している CI/CD は、ソフトウェア配信プロセスを自動化および合理化するための一連のプラクティスで構成されています。これらのプラクティスにより、開発チームは定期的にユーザーからのフィードバックを収集し、最も重要な製品機能と修正を優先して、リスクを軽減することで、より迅速にイノベーションを起こすことができます。

ここで疑問が湧きます。開発者はどの程度まで - 本当に - CI/CD のプラクティスと DevOps 文化を受け入れ、ソフトウェア開発とリリースプロセスの有効性を高めているのでしょうか？

Continuous Delivery Foundation (CDF) の委託を受け、SlashDataが執筆した本レポートでは、CI/CD開発者エコシステムの現状と、過去3年半の間にこれがどのように進化してきたかを探ります。より広範な DevOps 分野に関与している開発者の割合と、特に CI/CD テクノロジーに取り組んでいる開発者の数を調べます。

さらに、開発者のソフトウェア デリバリー パフォーマンスが¹時間の経過とともにどのように変化したかを確認します。最後に、開発者が複数の CI/CD ツールを使用することでソフトウェア デリバリー パフォーマンスにどのような影響もたらされるか、またさまざまな種類の DevOps テクノロジーを活用することによる影響について探ります。

本レポートの調査結果は、SlashDataが過去8回実施したDeveloper Nation調査のデータに基づくもので、2020年第3四半期から2024年第1四半期までの3年半にわたり、全世界で15万人以上の回答者を対象としています。

¹コード変更のリードタイム、デプロイメント頻度、サービスの復旧時間

誰がDevOpsに関心を持っているのか？

02

2. 誰がDevOpsに関心を持っているのか？

過去8回連続で行われたSlashDataによる半年に一度の Developer Nation 調査では、開発者に対し、CI、CD、アプリケーションのセキュリティ脆弱性のテストなど、一般的に DevOps の範囲に含まれるアクティビティに関与しているかどうかを尋ねました。

2024年第1四半期現在、開発者の83%がDevOps関連の活動に関与しています。これは、開発者がDevOps「スペシャリスト」であると自認していなくても、DevOpsプラクティスを広範囲に採用していることを表しています。現在、関係者の割合は2023年第1四半期のピーク(85%)からわずかに減少していますが、この減少はおもに新しい開発者によるものです。ソフトウェア開発経験が2年未満の開発者の4分の1は、DevOps関連の活動に一切関与していません。これは、新規開発者の大多数がDevOps活動に取り組んでいるものの、新規開発者には、キャリアを通じてスキル開発を最大限に高める上でのDevOpsプラクティスの利点と有用性について、さらに教育を施す必要があることを示しています。あるいは、より熟練した/経験豊富な開発者がソフトウェア配信プロセスに取り組むことを望んでいる組織がある可能性も示唆しています。要約すると、DevOpsの関与は昨年よりわずかに減少しましたが、それでも2022年第1四半期の77%よりも高い数値となっています。

すべてのDevOps関連アクティビティに関与する開発者の割合がわずかに減少しました。この要因のほんの一部は、関与しない開発者の増加によるものですが、主要な要因は、過去18か月間、開発者がより少ない活動に集中する傾向が続いていることです。これは、DevOpsに対する開発者と組織のアプローチの成熟度を表していると考えられます。

DevOpsの人気と認知度が高まるにつれ、開発者や組織はDevOpsから得られるメリットを模索しながら、幅広い活動に従事するようになりました。より成熟したDevOps環境に移行すると、アクティビティの有用性と有効性がよりよく理解されるようになります。これは、開発者が自分の役割に最も関連のあることに集中し、組織は開発者とチーム間で責任を分担している可能性が高いことを意味しています。さらに、もともと開発者の責任であった部分が自動化されている可能性があり、その場合、これらのプロセスを管理するために必要な開発者の数が少なくなります。

2. 誰がDevOpsに関心を持っているのか？

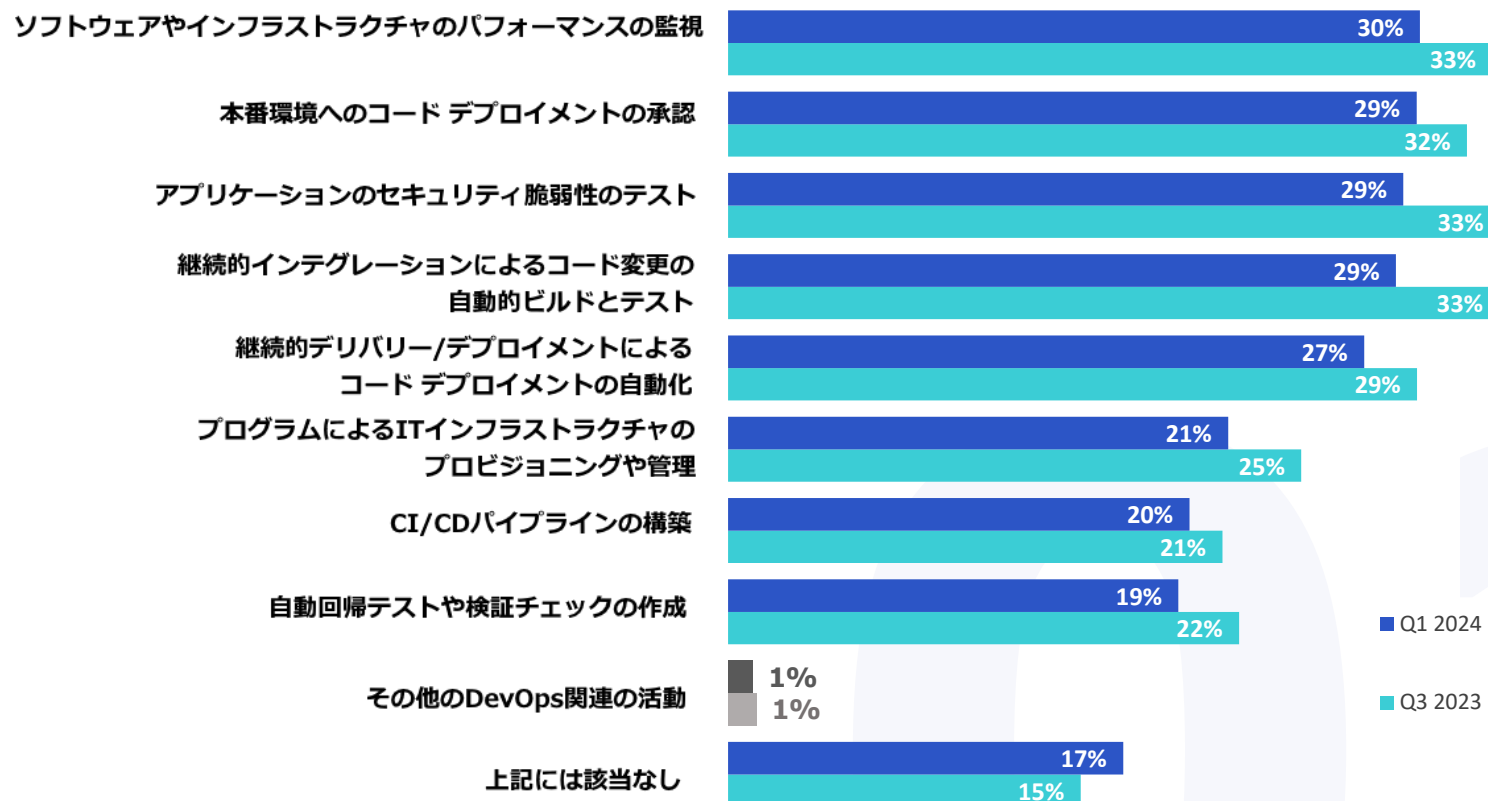
DevOps 関連のプラクティスはすべての開発分野で広く採用されていますが、セクターによって関与率が異なります。産業用 IoT 開発者の 90% が DevOps 関連の活動に従事していますが、ゲーム開発者の場合は 80% です。以前の結果と比較すると、機械学習/人工知能 (ML/AI) 開発者のうち DevOps に関与する割合が、データサイエンスの開発者を上回っています。ML/AI の新しい開発者の多くはソフトウェア開発全般の経験があるため、DevOps の関与は 85% で安定しています。一方、データサイエンスは、DevOps の使用率が低い経験の浅い開発者がこの分野に参入したため、2023 年第 1 四半期の 88% から 2024 年第 3 四半期には 83% に減少しました。

組織の規模に関して、当社のデータによると、中規模企業が DevOps プラクティスに最も関与していることがわかります (90%)。ただし、他の規模の組織では、DevOps 活動に取り組む開発者の割合が大幅に増加しています。現在、中小企業と大企業の両方で、開発者の 88% が DevOps 関連の活動に携わっており、2022 年第 1 四半期からそれぞれ 4 パーセントポイントと 3 パーセントポイント増加しています。しかし、DevOps に携わるフリーランサーの割合 (79%) は過去 2 年間安定しています。

2. 誰がDevOpsに関心を持っているのか？

開発者の大多数（83%）がDevOpsの実践に携わっている

DevOpsプラクティスの人気



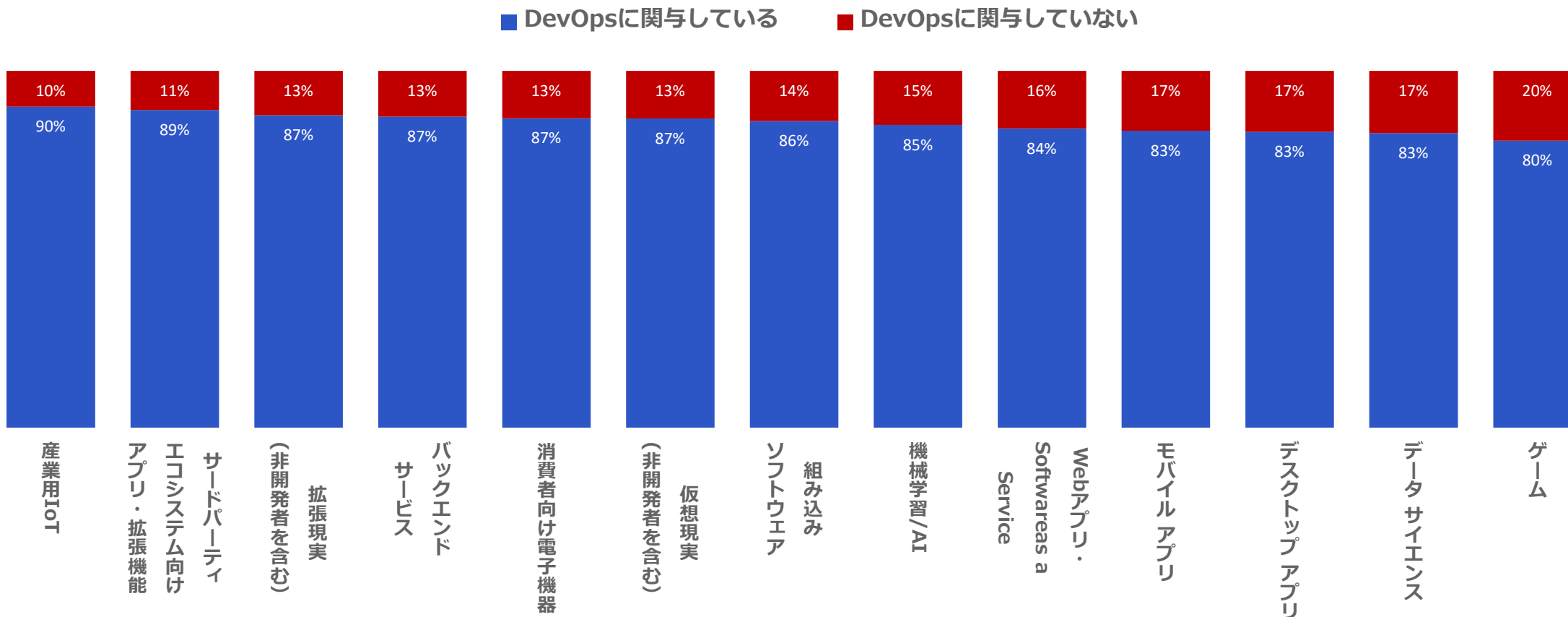
質問文：以下の開発活動のどれに関わっていますか？

開発者の割合2023年第3四半期 (n=16,111) | 2024年第1四半期 (n=9,807)

2. 誰がDevOpsに関心を持っているのか？

データサイエンスとゲーム開発者はDevOpsに關与する可能性が最も低い

ソフトウェア開発分野別のDevOpsへの関与

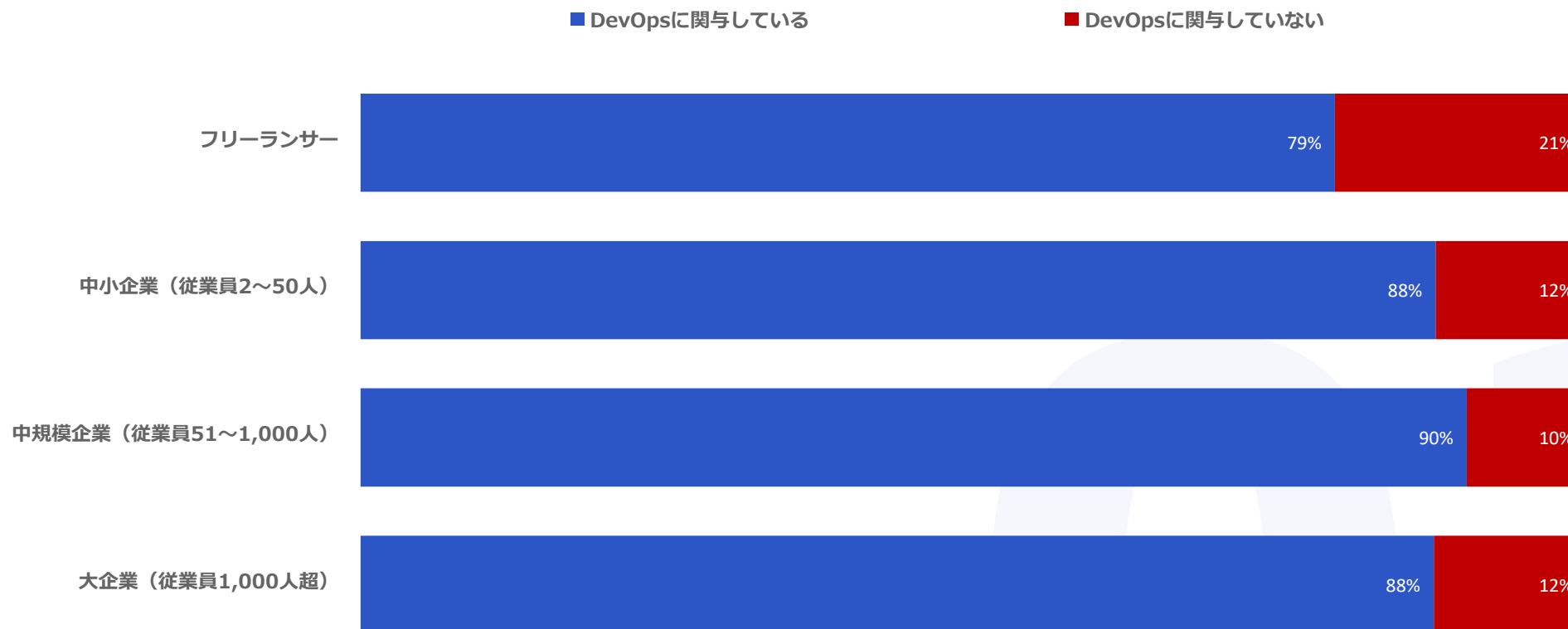


質問文：以下の開発活動のどれに関わっていますか？
ソフトウェア分野別のDevOps活動に携わる開発者の割合（9,807人）

2. 誰がDevOpsに関心を持っているのか？

組織規模を問わず、開発者はDevOps活動に同様のレベルで関与している

ソフトウェア開発分野別のDevOpsへの関与



質問文：以下の開発活動のどれに関わっていますか？
企業規模別に見たDevOps活動に携わるプロの開発者の割合（7,062人）

ソフトウェア デリバリーのパフォーマンスは
どのように進化してきたか？

03

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

個人や組織が CD の取り組みの有効性を測定するには、一連の堅牢なパフォーマンス指標が必要です。私たちの調査では、開発者に4つの DevOps Research and Assessment (DORA)指標のうち3つについてパフォーマンスを尋ねています。¹: 変更のリードタイム、デプロイメント頻度、サービス復旧時間。これらの指標はソフトウェア配信のパフォーマンスを測定し、組織のパフォーマンスを予測します。

私たちのデータでは、一般開発者層において、過去 3 年半にわたってコード変更の速度が向上したという明確な兆候は見られません。この期間中、リードタイムが 1 日未満のトップ パフォーマーの割合は 13% から 17% の間で変動し、現在は 2024 年第 1 四半期で 14% となっています。DevOps プラクティスの増加がまだパフォーマンスにプラスの影響を与えるまでには至っていない可能性があります。しかし、DevOps はここ数年間、専門的な開発の主力となっています。むしろ、DevOps プラクティスの普及により、開発者や組織が関与するプロジェクトの複雑さが増し、開発速度のメリットが相殺されている可能性があります。言い換えれば、DevOps プラクティスにより、複雑なプロジェクトの開発速度が、DevOps プラクティスのない単純なプロジェクトと同等になったと考えられます。

¹ Forsgren, N., Humble, J., Kim, G. 2018. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

デプロイメント頻度に関しては、トップ パフォーマー（1 日に複数回のデプロイメントを行う開発者）の割合が継続的に減少していることがわかりました。しかし、この減少は時間の経過とともに小さくなっており、懸念されるものの、それだけでは大きな危機を示すものではありません。しかし、月に 1 回未満の頻度でデプロイする開発者の割合が増加していることは懸念されます。

予期しないダウンタイム後のサービス復旧時間についても同様の傾向が見られます。1 時間以内にサービスを復旧できる開発者の割合は、2020 年第 3 四半期の 17% からは減少したものの、2022 年第 3 四半期以降は約 11% で推移しています。しかし、サービスの復旧に 1 週間以上かかる最もパフォーマンスの悪い開発者の割合は着実に増加しており、現在では開発者の 41% がその状況にあります。

DevOps では、パフォーマンスに悪影響を与える可能性のある、ますます複雑化するプロジェクトのスコープ設定が可能になる一方で、DevOps テクノロジーの使用の統合も要因の 1 つとなる可能性があります。次のセクションで検討する通り開発者は、使用しているさまざまな DevOps テクノロジーの数を着実に減らしています。DevOps が成熟するにつれて、開発者は空間を探索するのではなく、最も有用と思われるテクノロジーのみに焦点を当てるようになります。ただし、有用性は必ずしもデプロイメントパフォーマンスと直接相関するわけではありません。

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

DevOps ツールを追加することで、デプロイメントのパフォーマンスは向上したかもしれませんが、作業にさらなる精神的負担が加わりました。具体的な例としては、よく報道される警戒疲労の問題が挙げられます²。幅広いツールを使用することで、開発者はデプロイメントのパフォーマンスを向上できたかもしれませんが、メリットのない追加の側面も伴い、安全で迅速なデプロイメントには不可欠であっても、開発者のワークフローにはあまり役に立たないと考えられています。

開発者がワークフローで使用しているテクノロジーの数は、開発者の速度と強く相関しています。特に、最も速度の低いグループでは、より少ないテクノロジーを使用しているグループの割合が非常に高くなっています。

² *Understanding and fighting alert fatigue, Atlassian*

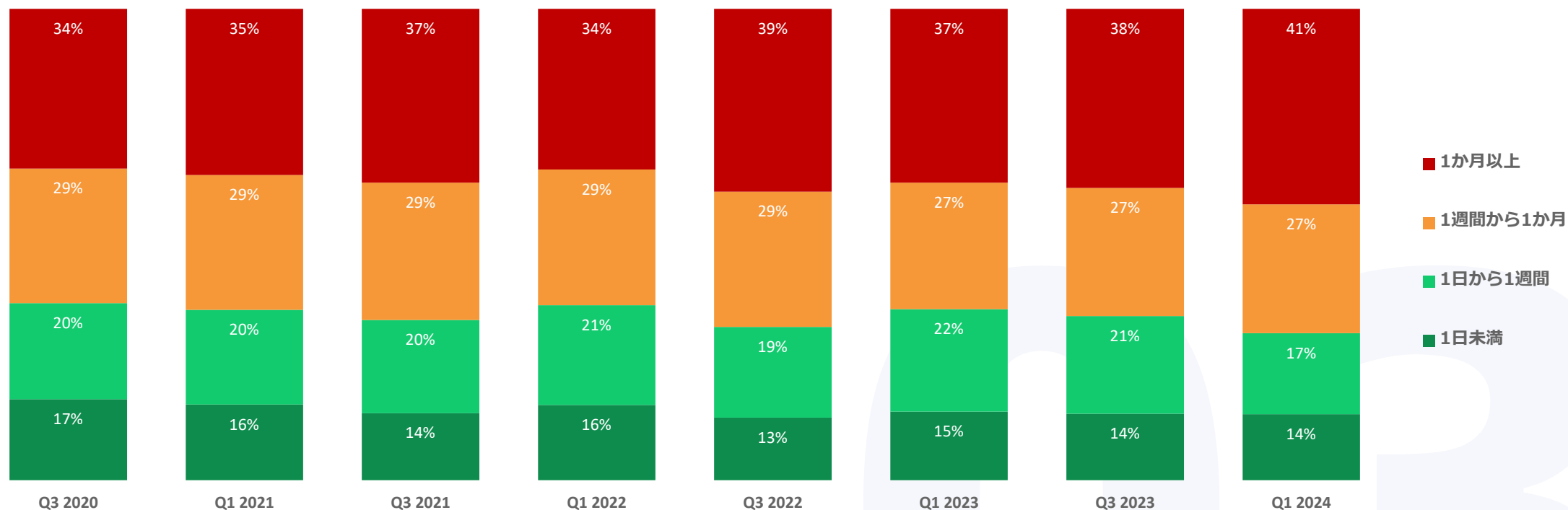
速度と安定性の指標は、一方が他方を損なうのではなく、強い相関関係を維持します。コード変更のリードタイムのパフォーマンスが低い開発者の大多数 (61%) は、サービス復旧時間のパフォーマンスも低いです。一方、リードタイムでトップの成績を収めている企業の 37% は、サービス復旧でもトップの成績を収めています。

さらに、サービス復旧時間のパフォーマンスが 2 番目に優れたグループの開発者の 44% は、リードタイムが 1 週間未満であり、安定性ではトップクラスのパフォーマンスではないものの、速度が良好であることを示しています。

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

コード変更リードタイムの低パフォーマンスの割合は増加し続けている

ソフトウェア デリバリー パフォーマンス – コード変更のリードタイム



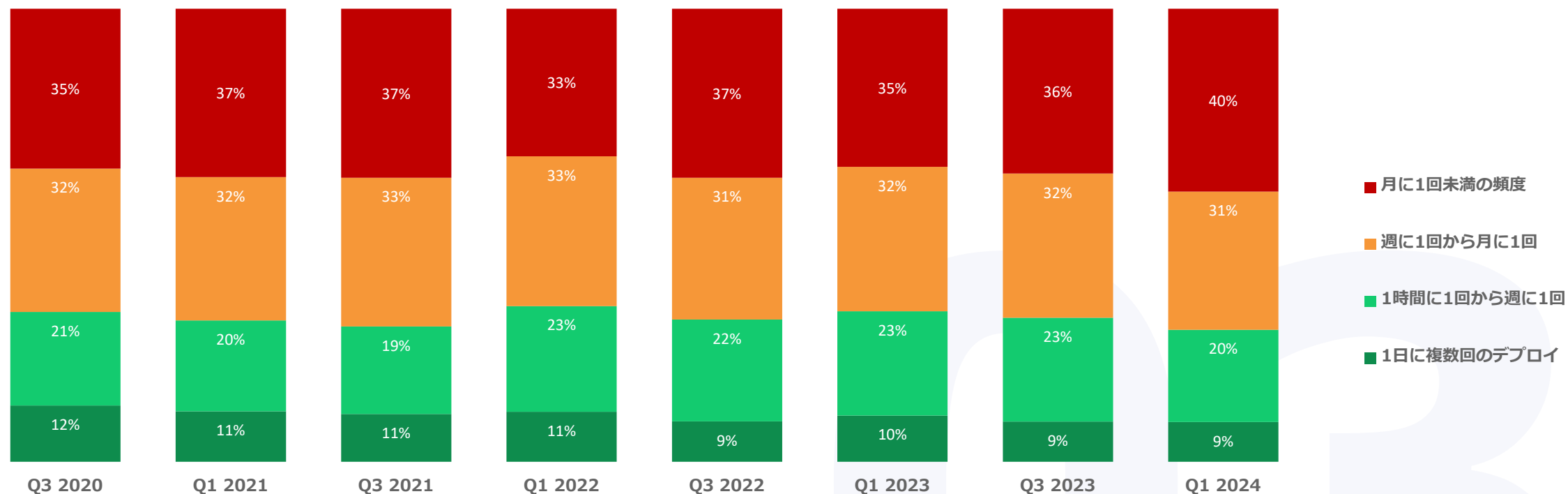
質問文：コミットされたコードから本番環境で正常に実行されるコードになるまでに、平均してどのくらいの時間がかかりますか？

DevOps実践者の割合（2020年第3四半期 n=10,252 | 2021年第1四半期 n=7,814 | 2021年第3四半期 n=8,784 | 2022年第1四半期 n=9,640 | 2022年第3四半期 n=13,108 | 2023年第1四半期 n=13,048 | 2023年第3四半期 n=12,055 | 2024年第1四半期 n=7,276）

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

トップ パフォーマーの割合（9%）は、2022年第3四半期以降ほぼ変わらない

ソフトウェア デリバリーパフォーマンス – デプロイメントの頻度



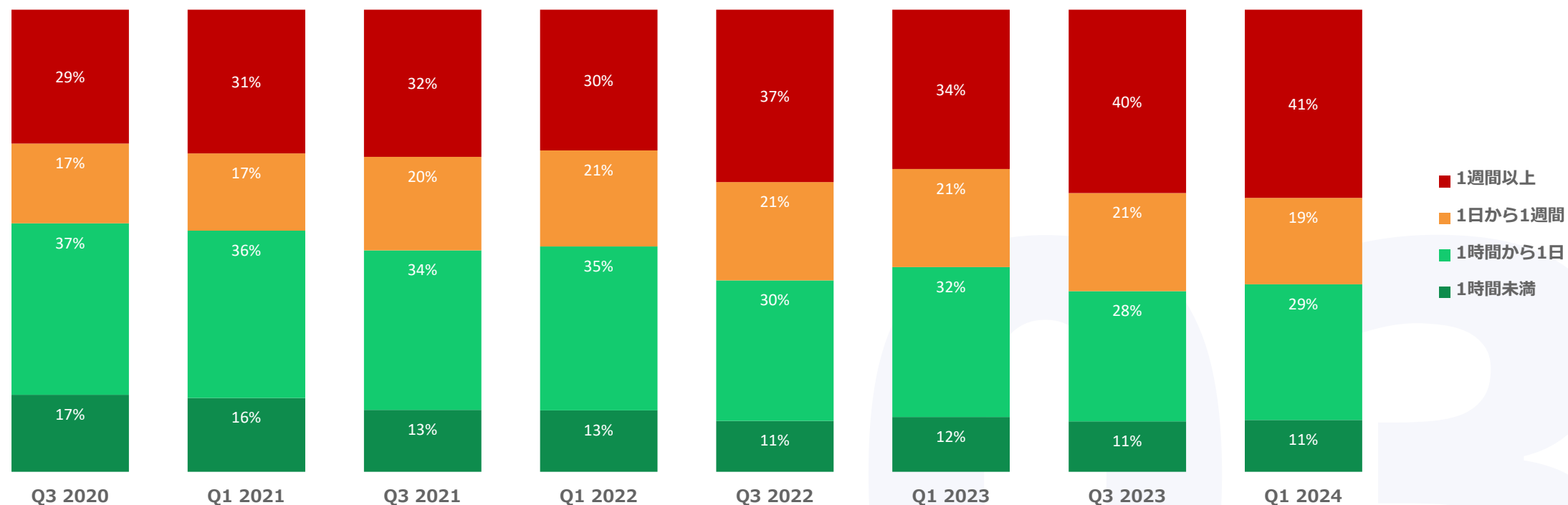
質問文： あなたまたはあなたのチームは、平均してどのくらいの頻度でコードを本番環境にデプロイしますか？

DevOps実践者の割合（2020年第3四半期 n=10,119 | 2021年第1四半期 n=7,613 | 2021年第3四半期 n=8,619 | 2022年第1四半期 n=9,473 | 2022年第3四半期 n=12,912 | 2023年第1四半期 n=12,747 | 2023年第3四半期 n=11,822 | 2024年第1四半期 n=7,149）

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

サービス復旧におけるトップ パフォーマーの割合は、2022年第3四半期以降安定しているが、低パフォーマーの割合は、2022年第3四半期以降増加している

ソフトウェア デリバリー パフォーマンス – サービスの復旧時間

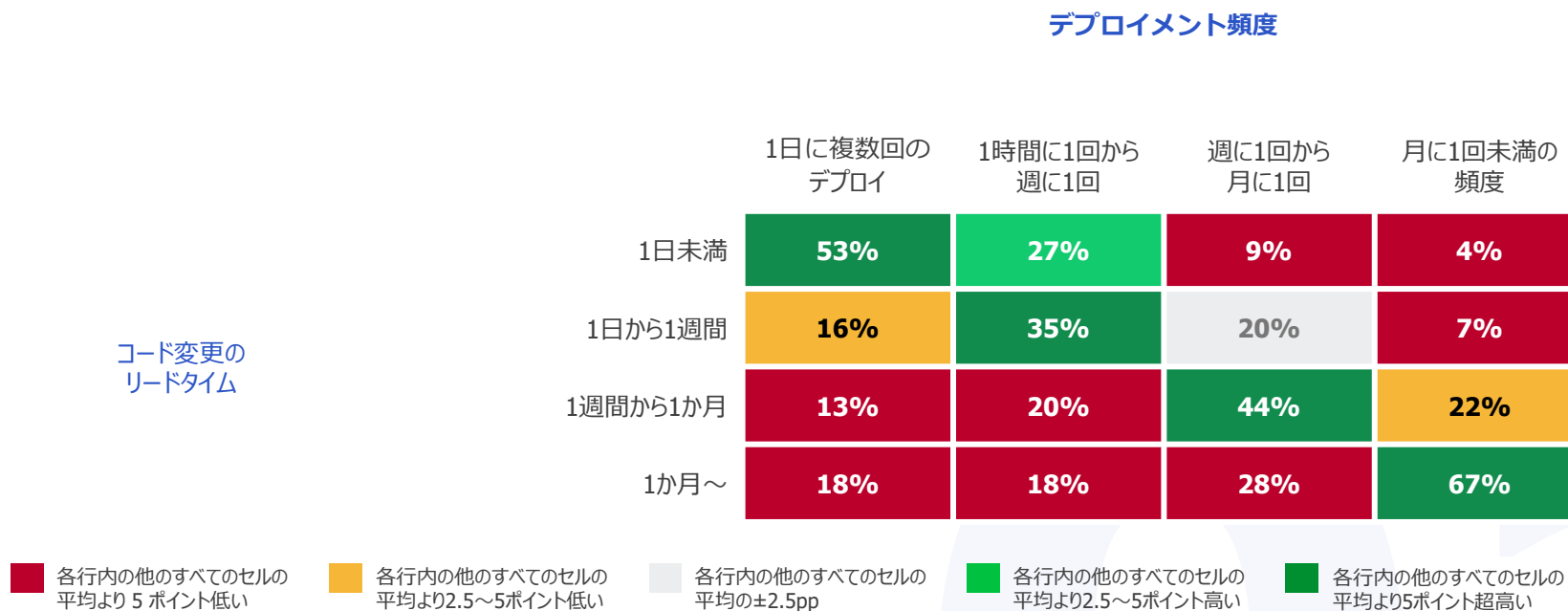


質問文： あなたやあなたのチームが計画外の停止やサービス障害からサービスを復旧するのに、平均してどのくらいの時間がかかりますか？

DevOps実践者の割合（2020年第3四半期 n=9,349 | 2021年第1四半期 n=7,221 | 2021年第3四半期 n=8,126 | 2022年第1四半期 n=8,927 | 2022年第3四半期 n=12,385 | 2023年第1四半期 n=12,250 | 2023年第3四半期 n=11,562 | 2024年第1四半期 n=6,909）

3. ソフトウェア デリバリー パフォーマンスはどのように進化してきたか？

コード変更のリードタイムとデプロイメント頻度のパフォーマンスは密接に関連している



質問文： コミットされたコードから本番環境で正常に実行されるコードになるまでに、平均してどのくらいの時間がかかりますか？ | あなたやあなたのチームが計画外の停止やサービス障害からサービスを復旧するのに、平均してどのくらいの時間がかかりますか？
DevOps実践者の割合（6,662）

ソフトウェア デリバリー パフォーマンスの原動力

04

4. ソフトウェア デリバリー パフォーマンスの原動力

4.1 DevOpsテクノロジーの使用

私たちの調査では、ソースコードの管理ツールからアプリケーションのパフォーマンスを監視するツールまで、開発者が使用する幅広い DevOps 関連テクノロジーに関する情報を収集しています。リストされているテクノロジーのうち、DevOps 実践者が使用するテクノロジーの平均数は、最近減少しており、2023 年第 1 四半期以前は平均 4 つ以上でしたが、2024 年第 1 四半期には 3.4 に減少しています。

DevOps プラクティスの傾向と同様に、DevOps テクノロジーは時間の経過とともに使用量が着実に減少していますが、相対的な人気は維持しています。アジャイル プロジェクト管理ツールと AI 支援コーディング ツールのみが、それらを使用する開発者の割合を維持しました。ただし、開発者の人口は継続的に増加しており、利用可能なツールの数も増加していることに注意することが重要です。そのため、DevOps 技術を使用する開発者の割合がわずかに減少したとしても、必ずしも使用率の低下を示すものではありません。

さらなる影響としては、開発者やチームが DevOps 中の「シフトレフト」¹ 開発哲学に取り組む方法が変わるかもしれません。セキュリティへの懸念²、開発者が十分な権限を与られていないと感じていること、DevOps チームや DevSecOps スチームがプロセスの集中化や管理を重要視していることなどが混在しているため、開発者自身が DevOps 活動を行うことは少なくなっています。

DevOps を単なるテクノロジーの集合体としてではなく哲学として取り入れるには、プロセスが開発者のエクスペリエンスと速度を向上させるように機能していることを確認するための時間と資本の投資が必要です。シフトレフトの問題に関するさらなる証拠は、従業員 1,000 人を超える企業の開発者に見られます。測定された 3 つの指標すべてにわたる開発速度は、過去 3 年半にわたって安定していますが、小規模組織ではパフォーマンスが低下しています。

¹ *"Why 'shift left' is now a dirty term in some security circles", Ericka Chickowski, Reversing Labs*

² *"The State of Software Supply Chain Security Maturity", Liam Dodd, SlashData & Red Hat*

4. ソフトウェア デリバリー パフォーマンスの原動力

さまざまな DevOps テクノロジーの相対的な採用率とは別に、より多くのテクノロジーの使用が開発者のパフォーマンスに影響を与えるかどうかを理解することが重要です。

私たちの目的は、DevOps のさまざまな側面とそれをサポートするテクノロジーに幅広く精通した開発者を擁することで、総合的にパフォーマンスを向上させることができるかどうかを調査することです。逆に、開発者にとっては、焦点や責任が狭く特定され、管理するツールや技術が少なくて済むことは有益かもしれません。

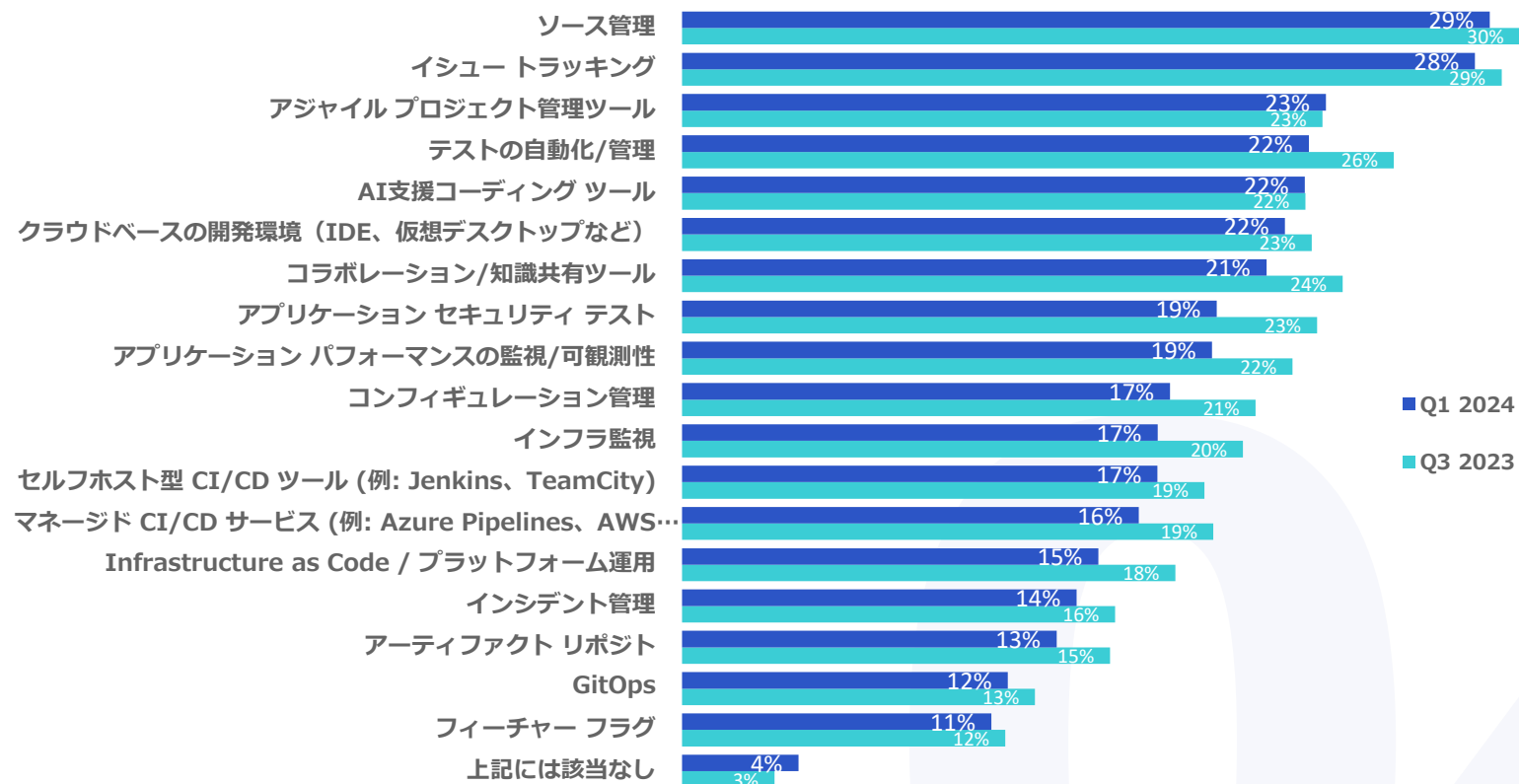
複数の DevOps テクノロジーの使用は、開発者のパフォーマンスの向上と強い相関関係にあることがわかりました。しかし、技術的な規模が大きいところではトップ パフォーマーが増加していますが、それ以上に際立っているのは、低パフォーマーの割合が大幅に減少していることです。

2024 年第 1 四半期時点で、1 つのテクノロジーを使用している開発者のうち、コード変更のリードタイムについては 50%、デプロイメント頻度については 47%、サービス復旧時間については 57% が低パフォーマンス グループに属しています。10 個以上のテクノロジーを使用している開発者で、前述のパフォーマンス指標の低パフォーマンス グループに属するのは、それぞれ 13%、18%、6% のみです。DevOps テクノロジーを幅広く使用すると、トップ パフォーマーになる可能性が高くなりますが、さらに重要なことに、低パフォーマーになる可能性が大幅に低くなります。

4. ソフトウェア デリバリー パフォーマンスの原動力

AI支援コーディングおよびアジャイル プロジェクト管理ツールの使用率は、過去1年間安定している

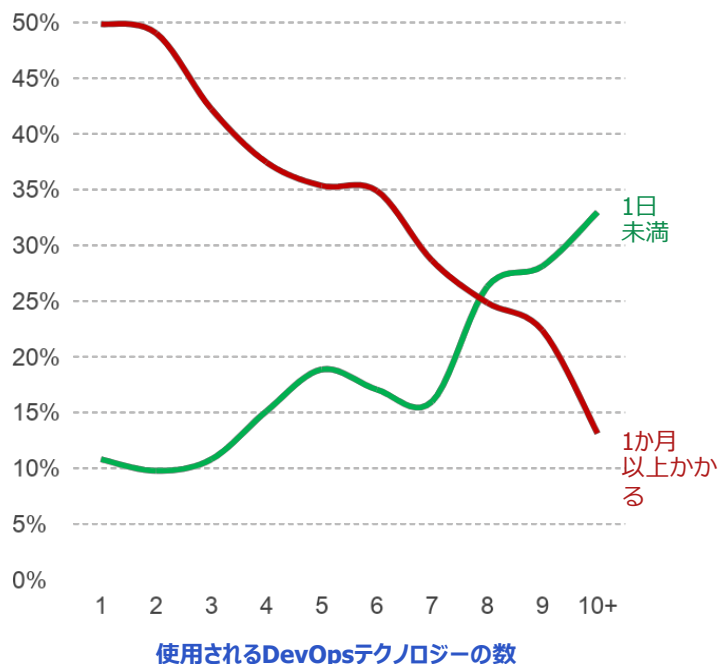
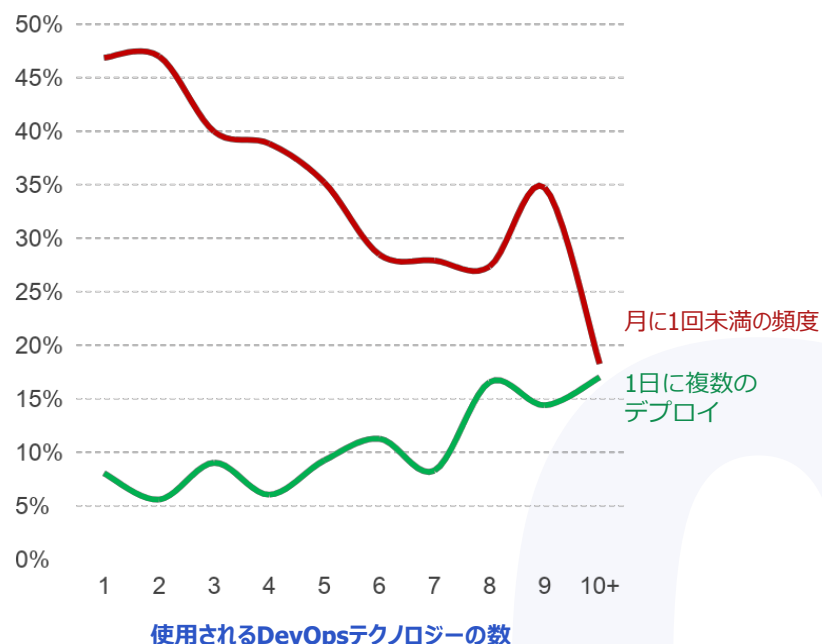
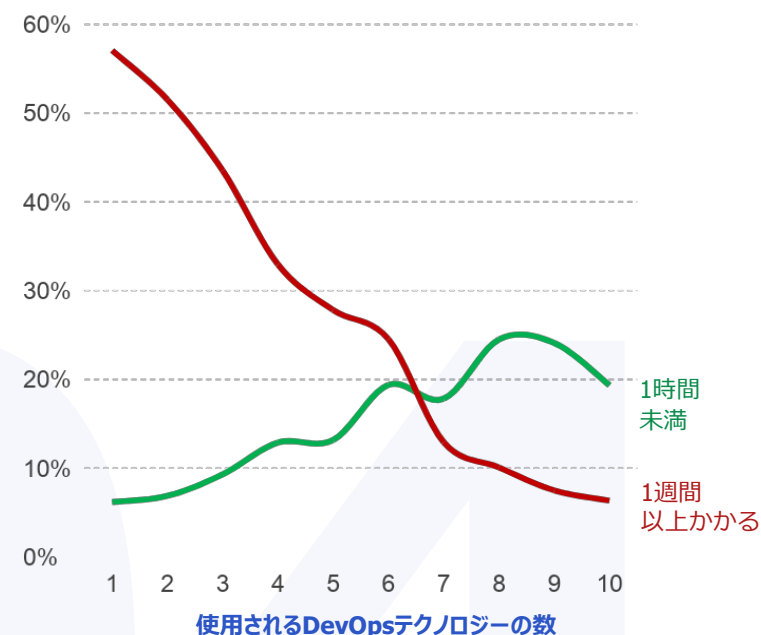
DevOpsテクノロジーの人気



質問文：過去 12 か月間の開発活動の一環として、次のテクノロジーのどれを使用しましたか？
DevOps実践者の割合2023年第3四半期 (n=13,599) | 2024年第1四半期 (n=8,102)

4. ソフトウェア デリバリー パフォーマンスの原動力

広範なDevOpsテクノロジーの使用は、デリバリー パフォーマンスの向上につながる

使用されるDevOpsテクノロジーの数に対する
コード変更のリードタイム使用されるDevOpsテクノロジーの数に対する
デプロイメント頻度使用されるDevOpsテクノロジーの数に対する
サービス復旧時間

質問文：過去 12 か月間に開発活動の一環として、次のテクノロジーのどれを使用しましたか？ | コミットされたコードから本番環境で正常に実行されるコードになるまでに、平均してどのくらいの時間がかかりますか？ | あなたまたはあなたのチームは、平均してどのくらいの頻度でコードを本番環境にデプロイしますか？ | あなたやあなたのチームが計画外の停止やサービス障害からサービスを復旧するのに、平均してどのくらいの時間がかかりますか？
使用されているDevOpsテクノロジーの数によるDevOps実践者の割合 (6,930)

4. ソフトウェア デリバリー パフォーマンスの原動力

4.2 セルフホスト型およびマネージド型プラットフォームの使用法と相互運用性

幅広いテクノロジーの使用は開発速度の向上に効果があることが示されていますが、特定のツールは他のツールよりも影響力が大きいことがわかっています。マネージドまたはセルフホストの CI/CD ツールを使用する DevOps 実践者と使用しない実践者を比較すると、3 つの開発指標すべてにわたってパフォーマンスに明確な違いが見られます。

CI/CD ツールを使用する開発者は、使用しない開発者に比べて、最高のパフォーマンスを発揮する可能性が高くなります。これは、サービス復旧までの時間でトップ パフォーマーとなる場合に特に顕著で、CI/CD ツールを使用する企業は、使用しない企業 (11%) よりもトップ パフォーマーとなる可能性が大幅に高くなります (最低でも 15%)。さらに顕著なのは、CI/CD ツールをまったく使用していない人は、CI/CD ツールを使用している人よりも低パフォーマンスである可能性が大きいということです。これはコード変更のリードタイムで最も顕著で、CI/CD ツールをまったく使用していない開発者の 40% が低パフォーマーで、これに続き、セルフホスト型の CI/CD プラットフォームのみを使用している開発者の 33% が低パフォーマーです。

さらに、開発者を、セルフホスト型 CI/CD プラットフォームを使用しているか、マネージド CI/CD プラットフォームを使用しているか、あるいはその両方を使用しているかに基づいて分類します。3 つの DORA 指標すべてにおいて、マネージド CI/CD プラットフォームを使用している企業は、セルフホスト型ソリューションのみを使用している企業よりも低パフォーマンスの可能性が少なくなります (リードタイムでは 30% 対 33%、デプロイメント頻度では 34% 対 32%、サービス復旧では 29% 対 25%)。マネージド CI/CD プラットフォームには、セルフホスト型プラットフォームよりも手動介入が少なく済むスケラビリティの利点と、開発者がインフラストラクチャの管理ではなくコア コンピテンシーに集中できるという利点の両方があります。

ただし、セルフホスト型とマネージド型の両方の CI/CD プラットフォームを使用している企業は、すべての DORA メトリックで最高のパフォーマンスを発揮しています。これらの開発者は、両方のアプローチの長所を活用でき、管理対象プラットフォームの利便性や組み込み機能などの側面と、必要に応じてセルフホスト型 CI/CD プラットフォームのカスタマイズを組み合わせ、効率性を高めることができます。これらの開発者は、CI/CD プラクティスをさらに最適化できる立場にある可能性もあります。マネージド プラットフォームやセルフホスト プラットフォームをいつ使用するかを選択することで、より自然な学習の機会が生まれ、反復的な学習プロセスの開発に役立ち、プロセスに大きな改善をもたらされます。

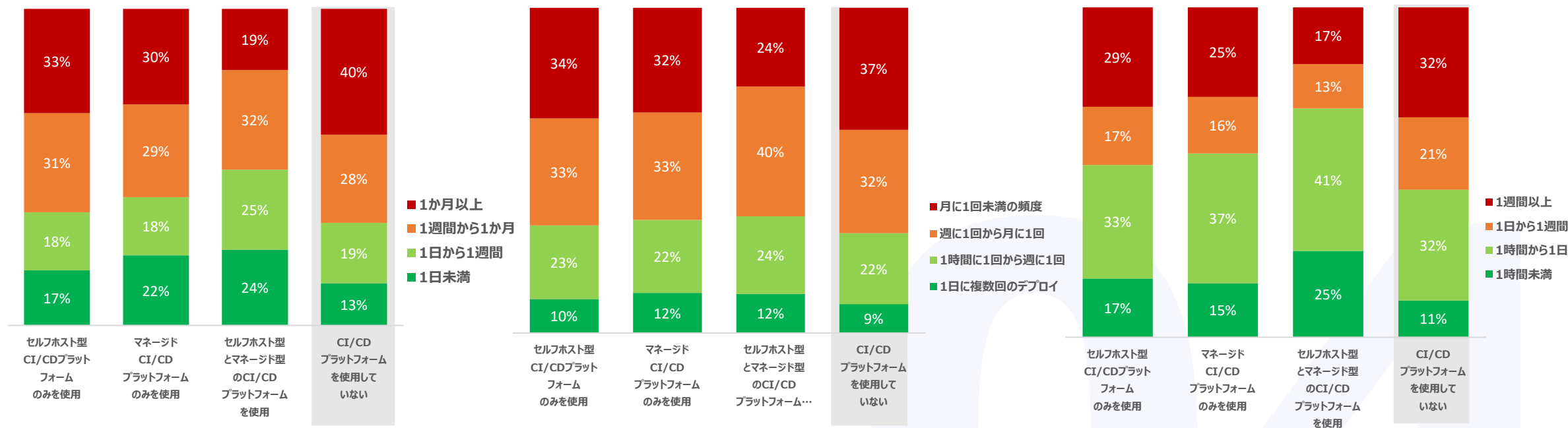
4. ソフトウェア デリバリー パフォーマンスの原動力

セルフホスト型とマネージド型の両方のCI/CDプラットフォームを使用すると、デリバリー パフォーマンスが向上する

コード変更リードタイムの パフォーマンス

デプロイメント頻度の パフォーマンス

サービス復旧時間の パフォーマンス



質問文: 過去 12 か月間に開発活動の一環として、次のテクノロジーのどれを使用しましたか？ | コミットされたコードから本番環境で正常に実行されるコードになるまでに、平均してどのくらいの時間がかかりますか？ | あなたまたはあなたのチームは、平均してどのくらいの頻度でコードを本番環境にデプロイしますか？ | あなたやあなたのチームが計画外の停止やサービス障害からサービスを復旧するのに、平均してどのくらいの時間がかかりますか？
各CI/CD構成のDevOps実践者の割合 (6,930)

4. ソフトウェア デリバリー パフォーマンスの原動力

4.3 複数のCI/CDプラットフォームの相互運用性

CI/CD ツールを使用すると、コード変更のリードタイム、デプロイメントの頻度、サービスの復旧時間全体で DevOps 実践者がトップ パフォーマーになる可能性が高くなることがわかっています。マネージドプラットフォームとセルフホストプラットフォームを組み合わせることは有益であることがわかりましたが、同じ形式の複数のCI/CDプラットフォームを使用すると相互運用性の問題が発生する可能性があるという懸念があります。

3つのデリバリー パフォーマンス指標すべてにおいて、使用されるセルフホスト型 CI/CD ツールの数が増えてもパフォーマンスが向上するわけではないことがわかります。この傾向は、複数のマネージド CI/CD プラットフォームを使用する場合にも同様ですが、デリバリーパフォーマンス指標によってはそれほど極端ではありません。

コード変更のリードタイムとデプロイメントについては、使用するセルフホスト型 CI/CD プラットフォームの数を増やしてもパフォーマンスは向上しないことがわかります。特に、ワークフローで3つのセルフホスト型 CI/CD プラットフォームを使用している開発者については、トップ パフォーマーが10% まで低下していることがわかります。デプロイメント頻度やサービス復旧時間でも、3つの CI/CD プラットフォームを使用する開発者には同様の低下が見られますが、それほど深刻ではありません。一方、使用されるセルフホスト型 CI/CD ツールの数が増えるにつれて、低パフォーマーの割合は劇的に増加します。これは、開発者が使用する CI/CD ツールの数を増やすと、収益が減少することを示唆しています。

4. ソフトウェア デリバリー パフォーマンスの原動力

マネージド CI/CD ツールの場合、追加ツールを導入すると、コード変更リードタイムパフォーマンスのトップ パフォーマーの割合が減少します。マネージド CI/CD プラットフォームが 1 つ以上になると低フォーマーの割合が増加しますが、マネージド CI/CD プラットフォームが 2 つから 4 つでは、その割合は一定のままです。これは、相互運用性の問題の複雑さは単一の追加プラットフォームの導入から生じるが、リードタイムパフォーマンスが低い場合、それ以上の追加では問題が同じ割合で増加しないことを示しています。

使用するツールの数が増えるのは、複雑さが増したことへの対応である可能性もありますが、これは開発者のパフォーマンスに悪影響を及ぼしています。同様に、特にセルフホスト型ツールを使用する場合、複数のツールの統合が最適に実装されない可能性があり、機能の重複が発生してパフォーマンスに影響を及ぼします。

開発者のデプロイメント頻度に関しては、より多くのマネージド CI/CD プラットフォームがワークフローに同時に導入されるにつれ、低フォーマーの割合が増加しています。ただし、全体的にトップ パフォーマーの割合は、1 つのプラットフォームのみを使用している開発者 (11%) と比較して、5 つ以上のプラットフォームを使用している開発者 (13%) の方が高くなっています。ただし、増加は非常に小さく、複数のマネージド CI/CD プラットフォームを使用すると、デプロイメント頻度の改善への影響は最小限に抑えられ、パフォーマンスの低下が増加する可能性の方がはるかに高いことが示唆されています。

サービス復元時間メトリックでは、使用されるセルフホスト型 CI/CD ツールの数の増加により、低フォーマーが最も劇的に増加します。1 つのセルフホスト型 CI/CD ツールを使用している開発者で、トップ パフォーマー (25%) になる可能性は、低フォーマー (18%) になる可能性よりも高くなります。ただし、2~3 のセルフホスト型 CI/CD ツールを超えると、状況は劇的に逆転します。5 つ以上のツールを使用する開発者のうち、62% が低フォーマーで、トップ パフォーマーはわずか 14% です。

4. ソフトウェア デリバリー パフォーマンスの原動力

サービスの復旧時間に非常に大きな悪影響を及ぼすセルフホスト型ツールの使用が増えている理由としては、複数の説明が考えられます。

However, interoperability issues may be at the center of many of them. 複数のツールを使用すると、それらすべてをうまく統合することが困難になり、サービスに影響を与える問題を切り分けることがさらに困難になる可能性があります。

サービスの復旧時間にも、ワークフロー内の管理対象 CI/CD プラットフォームの数の増加による悪影響が見られますが、他の DORA 指標ほど（特にコード変更のリードタイムに比べれば）深刻ではありません。さらに、複数のセルフホスト型ツールの影響と比較すると、悪影響ははるかに小さくなります。考えられる理由の 1 つは、コード変更のリードタイムがプラットフォームの使用だけでなく、CIプロセスと開発プロセスの影響も受けることです。複数のマネージド CI/CD プラットフォームにより、CI プロセスが断片化され、より大きな悪影響が生じる可能性があります。同様に、コードレビュー、コラボレーション、テストなどの開発プラクティスは、ワークフロー全体で複数のプラットフォームに適應する必要があることで影響を受ける可能性があり、この難しさがリードタイムのパフォーマンスに大きな影響を与えます。

一方、複数のマネージド CI/CD プラットフォームによってサービス復旧プロセスが複雑化しても、マネージド CI/CD プラットフォームに一般的に関連付けられている機能によって、悪影響が軽減される可能性があります。たとえば、マネージド CI/CD プラットフォームには、自動化されたロールバックや転送メカニズム、または分離された運用環境が備わっていることが多く、サービス復旧時間を短縮できます。複数の CI/CD プラットフォームがあると、問題の原因を特定してサービスを正しく復旧するための相互運用性の問題が発生しますが、プラットフォーム自体はサービス復旧にプラスの影響を与えます。

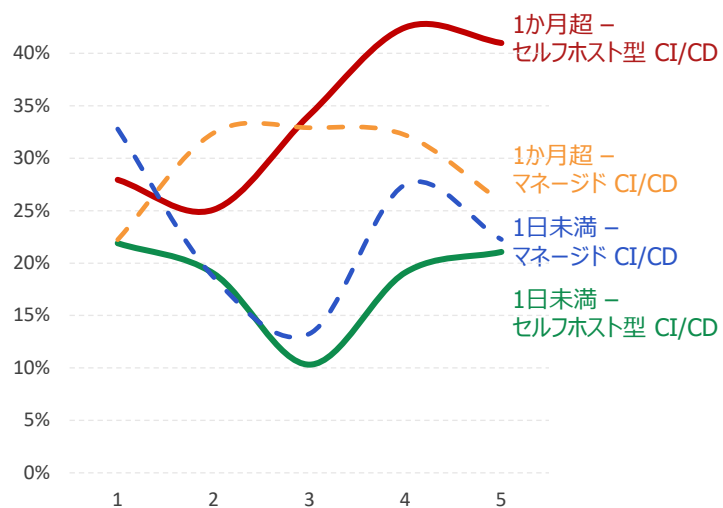
セルフホスト型ツールと比較すると、ツール間の標準化が不十分なため、すべてのツールがうまく連携することが難しくなり、サービス障害への対処の課題も増大する可能性があります。マネージド ツールよりも相互運用性に関する問題がはるかに大きくなります。

4. ソフトウェア デリバリー パフォーマンスの原動力

複数のセルフホスト型CI/CDツールを使用すると、サービス停止に対する応答時間が長くなる可能性がある

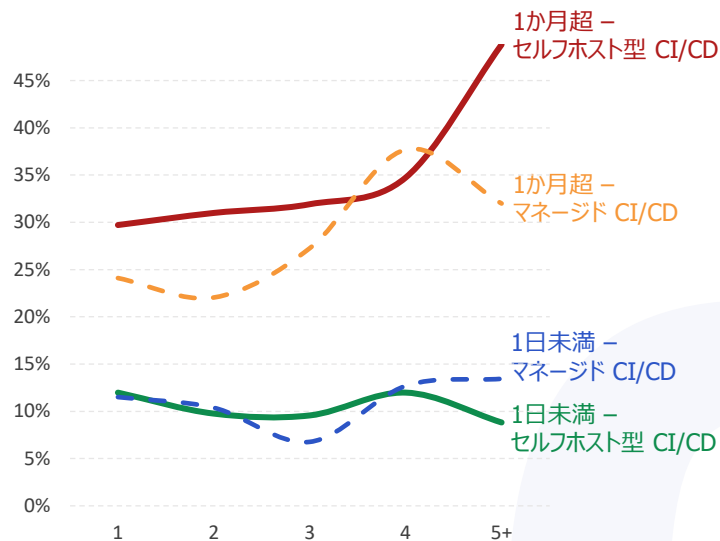
使用されるセルフホスト型 CI/CD プラットフォームの数がデリバリー パフォーマンスに与える影響

コード変更リードタイムのパフォーマンス



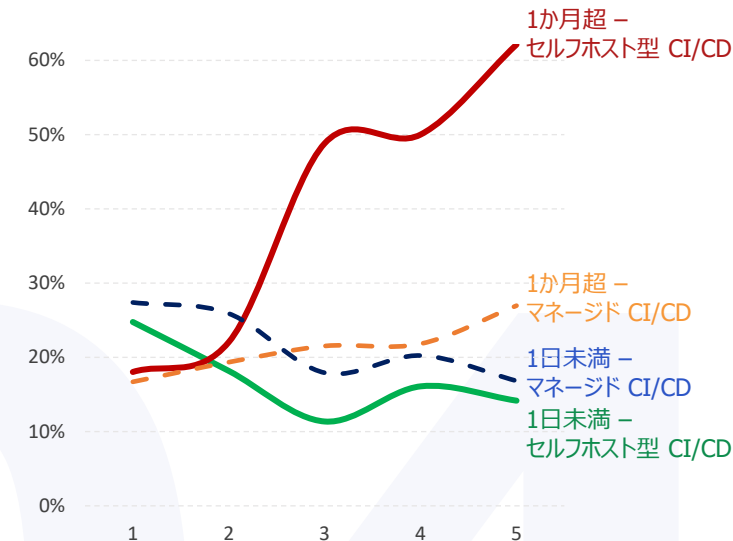
使用されているCI/CDツールの数

デプロイメント頻度のパフォーマンス



使用されているCI/CDツールの数

サービス復旧時間のパフォーマンス



使用されているCI/CDツールの数

質問文： 過去 12 か月間に、次のセルフホスト型 CI/CD ツールのうちどれを使用しましたか？ | 次のマネージド CI/CD サービスのうち、ご存じのもの、または現在使用しているものはどれですか？ | コミットされたコードから本番環境で正常に実行されるコードになるまでに、平均してどのくらいの時間がかかりますか？ | あなたまたはあなたのチームは、平均してどのくらいの頻度でコードを本番環境にデプロイしますか？ | あなたやあなたのチームが計画外の停止やサービス障害からサービスを復旧するのに、平均してどのくらいの時間がかかりますか？

% of DevOps practitioners for each performance metric using multiple CI/CD tools simultaneously (Self-hosted: n=1,295, Managed: n=1,192)

4. ソフトウェア デリバリー パフォーマンスの原動力

4.4 開発者の経験

前述のように、ソフトウェア開発の経験が少ない開発者は、経験豊富な同僚と同程度の速度で DevOps のプラクティスやテクノロジーを導入していません。経験豊富な開発者は、より複雑なプロジェクトに取り組んだり、所有権や責任のレベルが高まったりするにつれて、ワークフローに DevOps アクティビティをさらに追加することが予想される一方で、新しい開発者の間での採用が大幅に低いことは懸念されます。

現代のソフトウェア開発は、開発者や組織が DevOps の哲学とそれに関連するテクノロジーに取り組むことで大きなメリットを得ています。開発者はキャリアの早い段階でこれらを採用することで、成長を目指すために大きな学習の飛躍を必要とせずに、キャリアが進むにつれて開発ワークフローをカスタマイズするだけでなく、すぐにその恩恵を受けることができます。

ただし、経験に基づいた開発者の DORA 指標を見ると、経験の浅い開発者が経験豊富なチームメンバーから恩恵を受けられるような開発チームの構築に組織やチームが失敗していることを示しているのかもしれない。

経験豊富な開発者は、測定されたすべてのデリバリー パフォーマンス指標においてトップ パフォーマーである可能性が高くなります。コード変更のリードタイムに関しては、経験年数が 11 ~ 15 年 (22%) および 16 年以上 (21%) の開発者は、経験の浅い開発者と比較してトップ パフォーマーになる可能性が約 2 倍高く、経験年数が 5 年以下の開発者では約 10% です。

4. ソフトウェア デリバリー パフォーマンスの原動力

サービスの復旧時間については、その影響はさらに大きく、経験年数が2年以下の人のうちトップ パフォーマーはわずか5%であるのに対し、経験年数が11～15年の場合は16%、16年以上の場合は22%となっています。一方、デプロイメント頻度のトップ パフォーマーは、経験豊富な開発者の割合が依然として高いものの、経験の影響は他の2つのパフォーマンス指標ほど大きくありません。

しかし、開発者の経験を積むにつれてトップ パフォーマーが大幅に増加する一方で、低パフォーマーの減少はさらに大きくなります。コード変更のリードタイムに関しては、低パフォーマーは、経験の最も浅い開発者は56%ですが、最も経験豊富な開発者は29%と減少しています。デプロイメント頻度についても同様の傾向が見られ、51%から34%に減少し、最も顕著なのはサービス復旧で、67%からわずか9%に減少しています。

経験豊富な開発者の方が一般的にはより優れたパフォーマンスを発揮することが期待されていますが、特に組織内でのリリース用ソフトウェアの構築の性質を考えると、DORA指標はチーム規模の能力に焦点を当てていることから、これは深刻な懸念事項です。これは開発空間において、「プロセスにおけるチームや組織の一体性の欠如」や「組織のプロセスについて経験の浅い開発者が無自覚である」という2つの懸念すべき状況が発生する可能性があることを示唆しています。

経験の浅い開発者は、全体的に見て、チームが使用するプロセスに対する認識が薄く、そのため、チームが成果を出す速度も認識していない可能性があります。これにより、利用可能な最良の方法を使用しなかったり、組織が使用するDevOpsプラクティスから切り離して作業したりするため、パフォーマンスが低下する可能性があります。これは、DevOpsが特定のチームまたはチームメンバーに集中していることによっても影響を受ける可能性があります。つまり、経験の浅い開発者は、自分自身が頻繁にデプロイしないため、たとえば、チームがコード変更をデプロイする頻度を把握していないこととなります。

4. ソフトウェア デリバリー パフォーマンスの原動力

組織レベルでの結束力の欠如という2番目の要因によって、状況はさらに悪化するでしょう。他の研究でも強調されているように¹開発者はベストプラクティスに取り組もうと努力することが多いものの、組織全体でベストプラクティスを実施または標準化していないために失望することになります。経験の浅い開発者は、より経験豊富な開発者の監督や協力なしに、単独で作業することはほとんどありません。経験豊富な開発者の方がデリバリー パフォーマンス指標の結果が優れている場合、効果的な方法が新しい開発者と共有されていないことを示唆しています。

経験豊富な開発者は、経験の浅い開発者よりも平均して多くの DevOps テクノロジーを使用しています。経験が 2 年未満の開発者は平均 2.3 個のツールを使用していますが、経験が 16 年以上ある開発者は平均 5.2 個のツールを使用しています。これに加えて、彼らははるかに高い割合でテクノロジーを導入しています。たとえば、16 年以上の経験を持つ開発者の 65% がソース コントロール管理ツールを使用していますが、3 ~ 5 年の経験を持つ開発者ではわずか 22% で、経験年数が少ない開発者ではその割合はさらに低くなります。

すべてのチームが認識し関与する標準化されたアプローチを組織が決定しないと、DevOps と CI/CD アプローチの両方の利点を最大限に活用する能力が損なわれるリスクがあります。経験豊富な開発者はテクノロジーを活用して利益を得ることができますが、経験の浅い開発者は取り残されます。

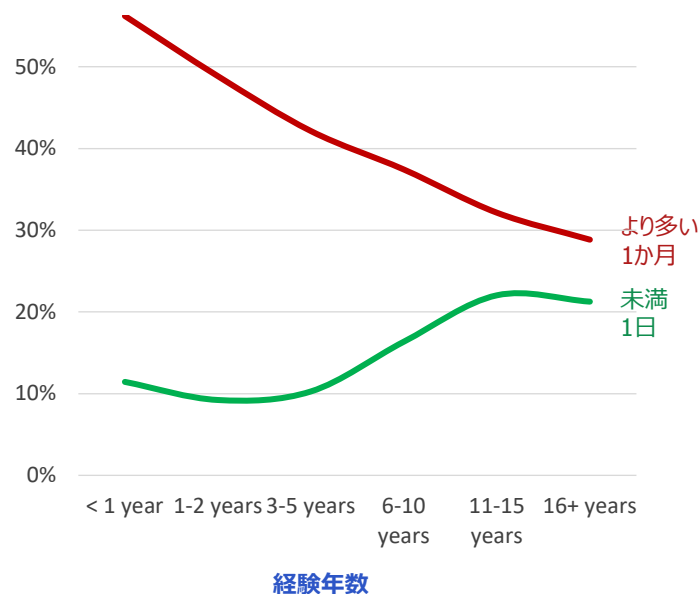
² "The State of Software Supply Chain Security Maturity", Liam Dodd, SlashData & Red Hat

4. ソフトウェア デリバリー パフォーマンスの原動力

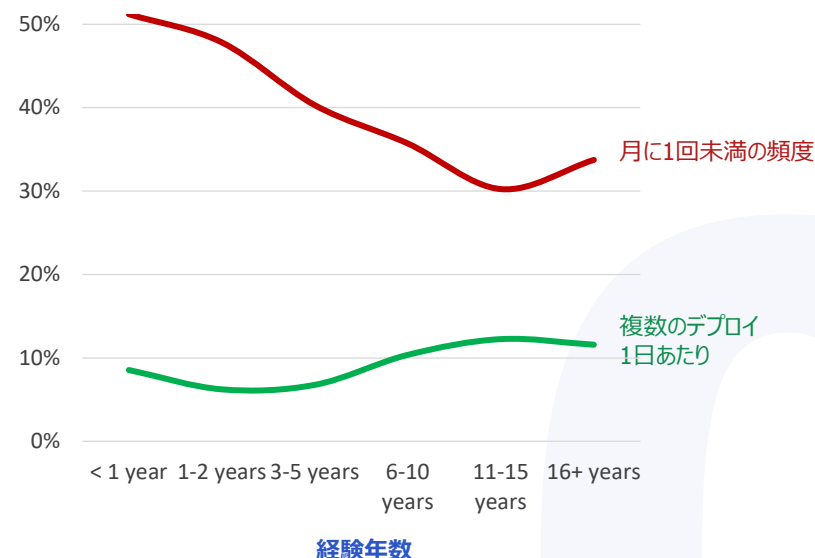
開発者のソフトウェア開発経験が増えれば、低パフォーマンスになる可能性は大幅に減少

使用されるセルフホスト型 CI/CD プラットフォームの数がデリバリー パフォーマンスに与える影響

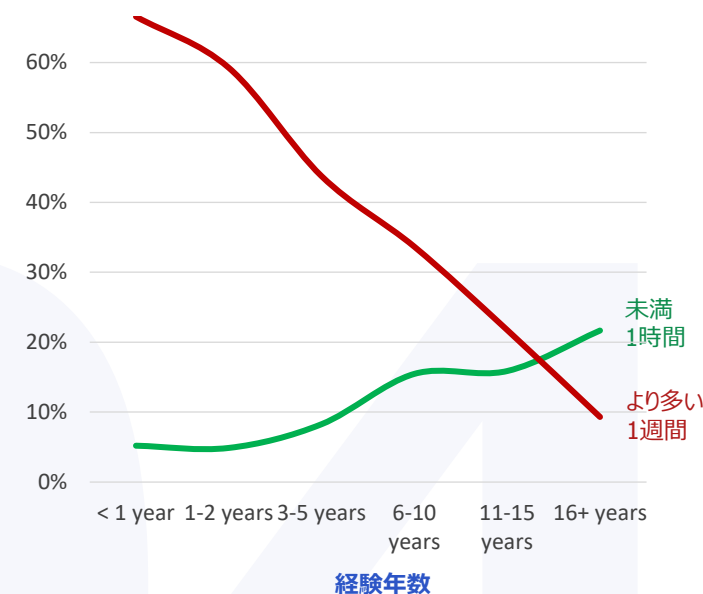
コード変更リードタイムのパフォーマンス



デプロイメント頻度のパフォーマンス



サービス復旧時間のパフォーマンス



質問文：ソフトウェア開発全般に携わって何年になりますか？|コミットされたコードから本番環境で正常に実行されるコードになるまでに、平均してどのくらいの時間がかかりますか？|あなたまたはあなたのチームは、平均してどのくらいの頻度でコードを本番環境にデプロイしますか？|あなたやあなたのチームが計画外の停止やサービス障害からサービスを復旧するのに、平均してどのくらいの時間がかかりますか？

複数の CI/CD ツールを同時に使用している各パフォーマンス指標の DevOps 実践者の割合 (セルフホスト: n=1,295、マネージド: n=1,272)

結論

05

5. 結論

このレポートは、DevOps を採用することで開発者や組織がソフトウェア配信のパフォーマンスを向上できることを示す既存の強力な証拠の基盤に追加されます。

本レポートは、DevOpsの導入が開発者と組織のソフトウェア デリバリー パフォーマンスの向上に役立つことを示す、従来の強力な証拠を補強するものです。幅広い DevOps テクノロジーの使用は、CI/CD ツールの具体的な使用と同様に、パフォーマンスの向上につながります。特に、マネージド CI/CD ツールとセルフホスト CI/CD ツールを併用すると、最高のデリバリー パフォーマンス結果が得られます。

しかし、開発者による DevOps への関与が減少し、DevOps テクノロジーの使用も減少する傾向にあります。これは新人の開発者の間で最も多く見られ、対処しないと、DevOps への関心や慣れが少ない開発者の集団が広がり、個人の成長と組織のパフォーマンスの両方に影響を及ぼす可能性があります。

このレポートの観察の根底にある重要な要素は、組織とチームリーダーがチームをより高いパフォーマンスに導く上で果たす重要な役割です。開発者がワークフローを単一のセルフホスト型および/またはマネージド型の CI/CD ツールに統合して相互運用性の問題を防ぐのを支援することや、新しい開発者が DevOps 全般や組織で使用されているプラクティスに精通していることを保証することなどです。これはまた、デリバリー パフォーマンスが開発者個人の経験に左右されることなく、最も経験豊富な開発者の才能を活用して経験の浅い開発者を向上させることができるよう、組織全体でベスト プラクティスを導入・構築するためのポジションを開くことにもなります。

5. 結論

最後に、開発者が使用しているテクノロジーの数が減っているのは、デリバリー パフォーマンスにメリットがあるにもかかわらず、開発者自身がワークフローに対する個人的な快適さよりもデリバリー パフォーマンスを重視する可能性が低いためかもしれません。DevOpsツールを提供するベンダーは、開発者の技術離れを防ぐために、技術の利用方法を検討する際に開発者の疲労を念頭に置く必要があります。同様に、幅広いDevOpsテクノロジーの使用は開発者のパフォーマンスに大きな影響を与えるため、ベンダーは自社のツールが他のテクノロジーとシームレスに統合できることを保証し、その機能も宣伝する必要があります。

組織とチーム リーダーは、デリバリー パフォーマンスを向上させるための計画の中心に開発者経験を置く必要があります。そのメリットは実証されていますが、開発者が開発経験に悪影響を与えるとして、これらのプロセスを無視したり回避したりすれば、メリットはそれほど大きくならないでしょう。どのプロセスをシフトレフトし、どのプロセスを特定のチームと開発者の責任として維持するかのバランスを取ることが、開発者とデリバリー パフォーマンスを向上させるためのより効果的な戦略であると考えられます。

付録

付録

経験豊富な開発者は、より多くのDevOpsテクノロジーを使用し、導入率も高くなる傾向がある

		ソフトウェア開発の経験						
		1年未満	1~2年	3~5年	6~10年	11~15年	16年以上	
Dev Ops テ ク ノ ロ ジ ー	ソースコントロール管理	14%	16%	22%	29%	44%	65%	各行内の他のすべてのセルの平均より5ポイント低い
	イシュー トラッキング	16%	16%	23%	32%	42%	55%	
	アジャイル プロジェクト管理ツール	10%	16%	23%	27%	34%	35%	各行内の他のすべてのセルの平均より2.5~5ポイント低い
	テストの自動化/管理	14%	17%	20%	27%	27%	34%	
	AI支援コーディング ツール	19%	21%	22%	24%	23%	24%	各行内の他のすべてのセルの平均の±2.5pp
	クラウドベースの開発環境 (IDE、仮想デスクトップなど)	16%	20%	22%	26%	22%	22%	
	コラボレーション/知識共有ツール	14%	16%	19%	25%	29%	29%	各行内の他のすべてのセルの平均より2.5~5ポイント高い
	アプリケーション セキュリティ テスト	16%	16%	20%	21%	21%	22%	
	アプリケーション パフォーマンスの監視/可観測性	12%	13%	17%	21%	28%	30%	各行内の他のすべてのセルの平均より5ポイント超高い
	コンフィギュレーション管理	13%	14%	16%	19%	23%	24%	
	インフラ監視	9%	11%	16%	21%	28%	26%	各行内の他のすべてのセルの平均より5ポイント超高い
	セルフホスト型 CI/CD ツール (例: Jenkins、TeamCity)	9%	12%	16%	19%	25%	28%	
	マネージドCI/CDサービス (例: Azure Pipelines、AWSCodePipeline)	8%	11%	14%	19%	27%	27%	各行内の他のすべてのセルの平均より5ポイント超高い
	Infrastructure as Code / プラットフォーム運用	9%	12%	14%	17%	21%	20%	
	インシデント管理	9%	10%	13%	15%	21%	22%	各行内の他のすべてのセルの平均より5ポイント超高い
	アーティファクト リポジトリ	7%	9%	11%	13%	19%	28%	
	GitOps	10%	12%	11%	15%	12%	10%	各行内の他のすべてのセルの平均より5ポイント超高い
	フィーチャー フラグ	8%	10%	10%	12%	15%	15%	
	上記以外	8%	4%	4%	2%	4%	4%	

質問文: ソフトウェア開発全般に携わって何年になりますか? | 過去 12 か月間に開発活動の一環として、次のテクノロジーのどれを使用しましたか?
DevOps テクノロジーを使用している各経験グループの DevOps 実践者の割合 (n=8,102)

第 26 回 Developer Nation Survey では、世界 135 か国から 10,000 人を超える回答者が集まりました。このように、Developer Nation の一連の調査は、モバイル、デスクトップ、産業用 IoT、民生用電子機器、組み込み、サードパーティ アプリ エコシステム、クラウド、Web、ゲーム、拡張現実と仮想現実、機械学習の開発者とデータサイエンティストを合わせた、これまでに実施された中で最もグローバルな独立調査です。このレポートは、2023 年 11 月から 2024 年 2 月までの 10 週間にわたって SlashData が設計、作成、実施した大規模なオンライン開発者調査に基づいています。

オンライン調査の回答者は、米国、中国、インド、イスラエル、英国などの主要なアプリ開発・機械学習開発のホットスポットのほか、ケニア、ブラジル、ヨルダンまでを含む 136 か国に及びました。この調査の地理的範囲は、まさに開発者経済の世界規模を反映しています。オンライン調査は、英語に加えて、簡体字中国語、繁体字中国語、フランス語、スペイン語、ポルトガル語、ベトナム語、ロシア語、日本語、韓国語の 9 つの言語に翻訳され、ソフトウェア開発業界内の 75 を超える主要なコミュニティとメディアパートナーによって宣伝されました。

回答者の年齢層は幅広く、18 歳未満の若いプログラマーやクリエイターから 55 歳以上のベテランまで多岐にわたります。

回答者には、調査対象の 13 件のうちどの種類のプロジェクトに関わっているか尋ねました。具体的には、Web アプリ / SaaS、モバイル アプリ、デスクトップ アプリ、バックエンド サービス、拡張現実、仮想現実、ゲーム、データサイエンス、機械学習 / 人工知能、産業用 IoT、家電機器、組み込みソフトウェア、およびサードパーティ アプリ エコシステム向けのアプリ / 拡張機能です。

また、彼らは、プロとして、趣味人として、学生として、あるいはこれらの組み合わせとして、それぞれの分野に携わっているのか、また、それぞれに何年の経験があるかを語りました。

地域的なサンプリング バイアスの影響を排除するために、まず地域内で過剰に代表されている個々の国を補正するために重み付けを行いました。次に、Developer Nation 調査で特定された地域分布と成長傾向によって決定された係数を使用して、9 つの地域にわたる地域分布に重み付けを行いました。モバイル、デスクトップ、産業用 IoT、家電機器、組み込みソフトウェア、サードパーティ アプリ エコシステム、クラウド、Web、ゲーム、拡張現実と仮想現実、データサイエンスと機械学習の各分野は、それぞれ個別に重み付けされ、その後統合されました。

アウトリーチ チャンネル全体にわたるその他の重要なサンプリング バイアスを最小限に抑えるために、回答に重み付けを行い、使用されたテクノロジーと開発者セグメントの代表的な分布を導き出しました。アンサンブル モデリング手法を使用し研究パートナーのチャンネルを除いた独立した代表的なチャンネルからのデータに基づいて加重分布を導き出して、これらのチャンネルを通じて募集された回答者による、サンプリング バイアスを排除しました。

これも、モバイル、産業用 IoT、家電機器、組み込みソフトウェア、サードパーティ アプリ エコシステム、デスクトップ、クラウド、Web、ゲーム、拡張現実/仮想現実、データサイエンスと機械学習のそれぞれについて個別に実行されました。

弊社の方法論の詳細について、[私たちの方法論のページ](#) を参照してください。



私たちは世界が開発者を理解するよう支援しています

弊社は、Web、デスクトップ、クラウド、モバイル、産業用 IoT、AR/VR、機械学習とデータサイエンス、ゲーム、コンシューマーエレクトロニクス、およびサードパーティエコシステム向けアプリ/拡張機能などの 30,000 人以上の開発者を毎年調査し、開発者がどのような人々で、何を購入し、次にどこに向かうのかを企業が理解できるよう支援しています。



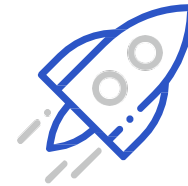
開発者とはだれなのか

開発者人口の規模
開発者のセグメンテーション



彼らは何を購入するのか

開発者が競合製品を採用する理由と、それを修正する方法



彼らはどこへ行くのか

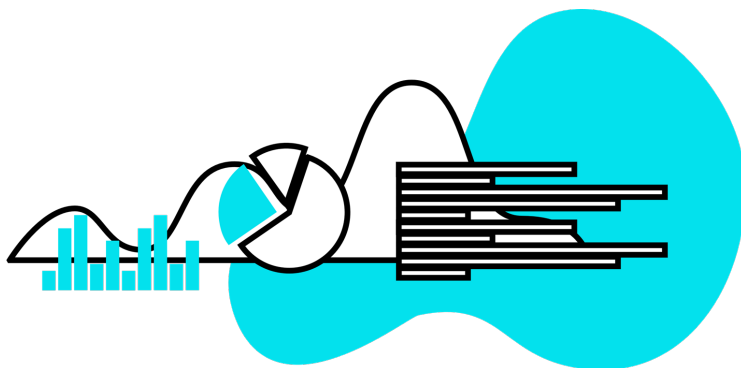
新興プラットフォーム –
拡張現実と仮想現実、機械学習

信頼ある実績

主要なテクノロジー プラットフォーム



私たちのクライアント



/DATA

THE ANALYST OF THE DEVELOPER ECONOMY

www.slashdata.co