GitLab

成熟するDevSecOpsの動向

2021年のグローバル調査結果

- 回答者数4300人
- リリース/デプロイの頻度、自動化、セキュリティの劇的な進歩
- 開発者、運用担当者、セキュリティ担当者による率直な意見

目次

はじめに

4年連続で、私たちはDevOpsチームに、慣行やプロセス、課題、キャリアについての真実を語ってもらいました。世界的なパンデミックの渦中にもかかわらず、今年の2月、驚くべきことに4,300人近い人々がそのために時間を割いてくれました。

その結果はさらに驚くべきものでした。今年、これまでで初めてDevOpsが重要であると捉えられたのです。その傾向は地味ですが確実に成長しています。*そしてそれは現在進行形です。*新型コロナウイルス感染症に関する質問は全く行いませんでしたが、長引く状況に対応した経験則が得られつつあるようです。世界中のチームが災厄に直面して、自動化、テスト、最先端のテクノロジーの採用など、最も重要なことに焦点を当てたものと考えられます。

2021年には、チームはDevOpsの「文化」の戦いから抜け出して、テクノロジーの実装と（驚くほど）明るい結果を出すための実作業に入る準備ができています。

60%の開発者が、DevOpsのおかげでコードのリリースが以前より2倍速くなったと回答しており、（パンデミック前の）2021年に比べて25%増加しています。

セキュリティ担当者の72%が、組織のセキュリティ対策を「良好」または「強固」と評価しており、この数は2021年に比べて13%増加しています。

運用チームのメンバーの56%が、業務が「完全に」または「ほとんど」自動化されていると回答し、2021年から10%増加しました。

回答者の約25%が完全なテスト自動化を行っていると答えており、2021年から13%増加しています。

75%のチームが、テスト/コードレビューにAI/MLやボットを利用しているか、利用を計画しており、2021年から41%増加しています。

昨年、開発者、セキュリティ担当者、運用担当者はそれぞれ、今後のキャリアのために、より良いコミュニケーションとコラボレーションのスキルが必要だと答えました。ソフトスキルを集中的に強化する時期を経て、今年は、AI/ML（開発者）、対象に関する専門知識（セキュリティ担当者）、高度なプログラミング（運用担当者）へと優先順位が大きく変化しています。

意識改革、厳しい議論、詳細な分析など、結果を出すために大変な努力をしたことが伝わってきました。DevOpsは（当社にとっても）簡単なことではありませんが、実際のデータを使って成果に焦点を当てることで、あらゆる国のあらゆる規模のチームが前進できるようになります。

これは当社が行った調査ですので、回答者が当社について言及したり、当社製品を使用したりしているのは当然のことです（調査対象者の約50%はGitLabのお客様です）。また、調査対象者の43%が3年から5年以上にわたってDevOpsを「実践」しており、熟練した実践者であるため、結果が理想的なものとなる傾向も見られます。あなたの結果は異なるかもしれませんが、問題ありません。

では始めましょう。

概要

2021年のDevSecOps調査の主な調査結果

DevOps＝より広い技術基盤

CI/CDは依然として重要ですが、DevOpsプラットフォームは、AI/MLと同様に増加しています。

テストに関するトラブル

リリース延期の主な理由として3年連続で「テスト」が挙げられました。

より良いコード品質を求めて

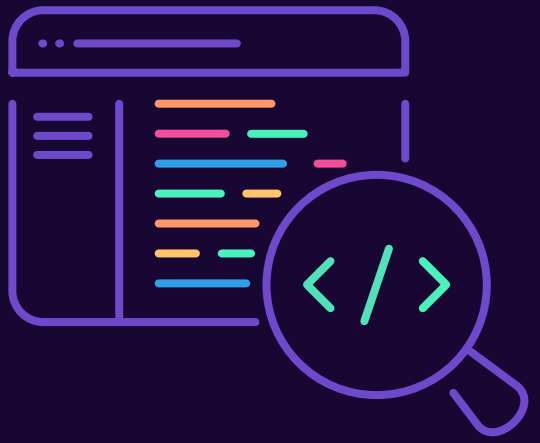
そうなると、やはりDevOpsの出番です。他にも、市場投入までの時間の短縮や計画の精度向上などのメリットがあります。

真剣に取り組む

昨年は、Kubernetesやマイクロサービスについて話していたチームが、今年はそれらをすでに使用しているか、まもなくの導入を計画しています。

開発と運用部門以外に広がるユーザー

DevOpsプラットフォームは、文字通り誰もが活用できるものです。また、回答者の23%が社内の全員がDevOpsプラットフォームを使用しているとしています。



出発点

ここでは、2021年2月から3月上旬にかけて調査に協力してくれた約4300人（4294人）について詳しく紹介します。

役職

41.67% ソフトウェア開発者/ソフトウェアエンジニア

8.59% DevOpsエンジニア

6.87% 開発/エンジニアリングリーダーシップ

7.88% ソフトウェアアーキテクト

4.56% その他

2.52% DevOpsリーダーシップ

2.81% テクノロジーエグゼクティブ - CIO/CTO/VP

4.11% プロジェクトマネージャー

1.66% サイト信頼性エンジニア

2.57% システム管理者

1.91% システムエンジニア/ネットワークエンジニア

2.23% プロダクトマネージャー

1.56% 品質保証

1.14% 運用リーダーシップ

性別

14.95% 女性

81.86% 男性

0.93% ノンバイナリー/第三の性

0.51% 自由回答を希望

1.75% 回答拒否

業種

40.04% コンピューターハードウェア/サービス/ソフトウェア/SaaS

6.3% 銀行/金融サービス

4.78% その他

18.05% 教育

5.29% ビジネスサービス/コンサルティング

3.4% 電気通信

2.19%メディア・エンターテイメント

1.91% ヘルスケア

2.24% 官公庁

地域

20.28% 欧州・ロシア

10.78% 北米

50% アジア

2.28% 南米

0.77% オーストラリア・ニュージーランド

1.82% アフリカ

0.98% 中東

従業員数

25.37% 1-10人

21.34% 11-100人

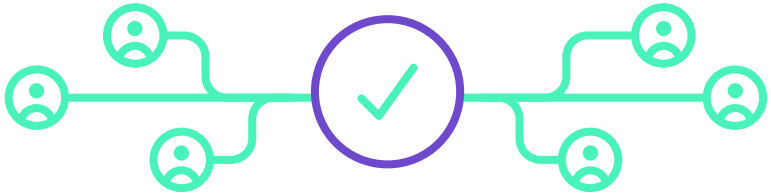
5.11% 101-500人

8.14% 501-1,000人

12.31% 1,001-10,000人

11.08% 10,000人超

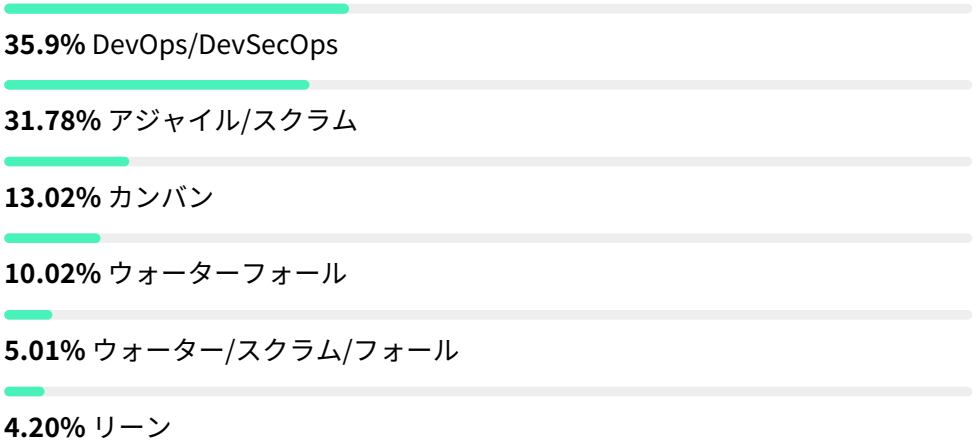
5.92% 分からない



今日のソフトウェア開発

2021年には、調査回答者の過半数（35.9%）が、チームでDevOpsまたはDevSecOpsを使ってソフトウェアを開発していると答え、次にアジャイル/スクラムが31.78%と僅差で続きました。これは、「自称」DevOpsの使用率が1年で大きく上昇したことを意味します。2020年には、自分たちのプロセスをDevOpsまたはDevSecOpsと表現したチームはわずか27%でした。10%強のチームがウォーターフォールモデルを使用していると回答しました（2020年の8%弱から上昇）。また、昨年と同じように、5%のチームが「クリエイティブ」で、プロセスを「ウォーター/スクラム/フォール」と表現しています。

最も実践されている開発手法は次のようなものです。



回答者の30%強が、自社でのDevOps実践開始から1年〜3年が経過したと回答しています。DevOpsを導入してから1年以内の人は約27%であり、5年以上の人は約23%です。約20%が3年から5年というDevOpsの「スイートスポット」に位置しており、成功体験があり、プロセスやルーチンに慣れていることを意味します。

今日のDevOpsの実装はどのようなものでしょうか？CI/CDが最も多く、次いでDevSecOps、テスト自動化、DevOpsプラットフォームの順となっています。2020年には、DevOpsでAI/MLを使用していると回答したのはわずか4%でしたが、今年は11.5%が使用していると回答しています。

その他の技術としては、以下のものが挙げられました。

> Infrastructure as Code（IaC）
>
> GitOps
>
> Kubernetes
>
> NetDevOps（大規模なネットワーキング）
>
> CI/CDを使用したプラットフォーム「ブートストラップ」
>
> SREの広範な使用
>
> Unix

2年連続で、回答者は、DevOpsの実践から恩恵を受ける可能性が最も高いのは開発者であると回答しました（36%）。続いて、運用（22%）、セキュリティ（16%）、そしてQAと事業部門がそれぞれ13%となりました。

DevOpsを選択する理由の上位3つとしては、コードの品質、市場投入までの時間短縮、そしてセキュリティが挙げられています。DevOpsを実践することで得られるその他の明確なメリットとしては、コミュニケーションやコラボレーションの改善、開発者の幸福度の向上が挙げられ、いずれも2020年よりも2021年の方がはるかに高い評価を得ています。

調査回答者の約59%が、自分たちのチームは1日に複数回、1日1回、または数日に1回デプロイしていると回答しました。この割合は昨年とほぼ同じであり、従業員数500人以下の企業に勤務する調査回答者の61%以上と一致していると思われます。全体では、28%が継続的に（1日に何度も）デプロイしており、15%が週に1回、10%が月に1回、7%以下が数ヶ月に1回の頻度でデプロイしています。

当然のことながら、調査回答者の大半がオープンソースプロジェクトに参加しており、昨年の63%から今年は69%以上に増加しています。29%以上が「GitLab」に貢献していると答え、14%が「Kubernetes」、13%が「VS Code」に関わっていると答えています。ただし、19%近くは「その他」のプロジェクトに参加していると回答しました。その多くは、小規模であまり知られていないプロジェクトです（昨年もこの傾向が見られました）。

「テストですべてが遅れる。」

「多くの人がコードをレビューすることを面倒に感じている。」

「計画を急ぎすぎたり、無視したりしたことによる運用の失敗。」

依然として厳しいテスト

3年連続で、調査対象者の過半数が、最も遅延が発生しやすい分野としてテストを挙げています。その他のボトルネックとしては、企画、コード開発、コードレビューなどが挙げられ、これも2019年と2020年の調査で見られた内容を反映しています。

この分野については、以下のようなコメントも見られました。

「新型コロナウイルス感染症**（何度も言及）。」

「テストでは書き込みと実行の両方が遅くなる可能性がある。」

「事業部門と開発部門のコミュニケーションが難しい。」

「セキュリティが開発プロセスにまったく統合されていない :(。」

「厳格なコードレビュープロセスのせいで、レビュー担当者がレビューのリクエストに対応するまでに数日かかることがよくあります。」

「コードレビュー担当者を見つけるのが難しい場合があります（平均で1日が必要）。その後、ビジネステストが完了するまで時間がかかります（平均2〜4日）。」

「テストはまだデプロイサイクルで完全に自動化されていません。BitBucket + Jenkins/DroneからGitLabへの移行で改善したいと考えています。」

「遅延の原因の多くは、複数のチームが異なる管理体制の下で関与し、経営層にまとまったビジョンがないことにあります。」

「5年前に選んだ技術スタックを未だに使用しているので、TDD、バージョンコントロール、ヘキサゴナルアーキテクチャーなどの最新のプラクティスに対応できていない。」

「開発者がコードレビューの必要を認識していないことがある。レビューの実施方法と、その効果の有無が分かっていない。時にはプロセスを進めるためにスキップされることもある。」

テストをやり通すのは大変なことですが、少しずつ前進の兆しが見えてきました。約25%のチームがテストを完全に自動化しており（昨年の2倍以上）、28%の回答者が少なくとも半分は自動化していると答えています。調査対象者の約34%が、開発者が自分のコードの一部をテストすると回答しており（昨年の31%から増加）、32%はコードが書かれた時点で自動テストが行われると回答しており、2020年の25%から大きく増加しています。

しかし、25%のチームは、テストの自動化を行っていないか、検討を始めたばかりで、9%のチームは、テストを十分にシフトレフト（プロセス初期段階に移動）していないと認めています。

当然のことながら、自動化されていないテストに対しては次のような不満があります。

> 「自動テストが『時間の制約のため』に無視される。」
>
> 「テスト？それは面白い。」
>
> 「TDDをやるつもりでしたが、結局は多くの場合後付けになります。」
>
> 「可能な限りTDDでコードを書こうとしています。Reactコンポーネントを書くときは複雑ですし、多くの副作用と多くの入力を伴うテストされていない関数を変更するときは、技術責任者にリファクタリングが禁じられます.... ='(。」

テストという迷宮の先にある一筋の光は、人工知能/機械学習の利用にあるかもしれません。2020年の調査では、コードをテストする「ボット」やテスト用のAI/MLツールを導入していると回答した人はわずか16%でしたが、今年は41%強となりました。全体では、回答者の25%がボットを使ってコードをテストし、16%がAI/MLを使って人間が確認する前にコードをレビューし、34%がAI/MLの採用を検討しているがまだ何もしていないと回答しています。テストにAI/MLを使用していないという回答者は、実に4分の1に上ります。

ツールのランク付け

調査参加者のほぼ85%が、ソース管理にGitを使い（昨年の92%から減少）、ほぼ4%がTeam Foundation Serverを使用し、2%がCVSを使っています。わずか5%が、「ソース管理に何もツールを使わない」と回答しました。

CI/ビルド用のツールとしては、GitLabが34%で、次いでJenkins（21%）、GitHub Actions（14%）、BitBucket（8%）が挙げられています。

調査対象者の37%強がマイクロサービスを「部分的に」使用していると回答し、34%が完全に使用しており（昨年の26%から増加）、28%は全く使用していません。回答者の中には、「マイクロサービスの使用を計画している、または調査している」という人もいれば、「今後1、2年でマイクロサービスに移行する予定だ」という人もいました。

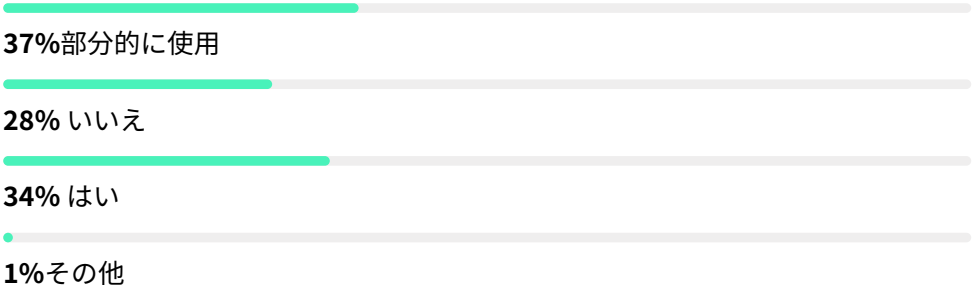
しかし、Kubernetesに関しては、間違いなく「たった1年で大きく変わる」という状況です。2020年には、私たちの調査対象者のうち、知られているようにK8sを使用しているのはわずか38%でした。今年は、46%がKubernetesを使用しており、37%は使用していません（昨年の50%から減少）。

まだ導入していないDevOpsチームは、昨年よりも今年の方がKubernetesの実際の導入に近づいています。

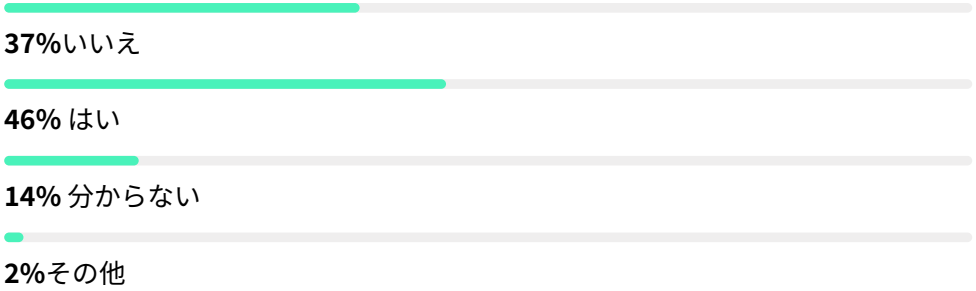
> 「まだ導入していませんが、最優先のアプローチとして検討中です。」
>
> 「2021年に導入予定です。」
>
> 「導入予定ですが、すべての作業に対応する予定はありません。ECS、サーバーレスも使用しています。」
>
> 「試行錯誤しましたが、今では小規模な本番環境には必要ありません。」
>
> 「ついに今年、K8sに移行します。」

ローコード/ノーコードの開発ツールも、今年はより真剣に検討されています。昨年は75%が「使っていない」と回答しましたが、今年は41%が「ローコード/ノーツール」を使い、59%が「使っていない」と回答しました。

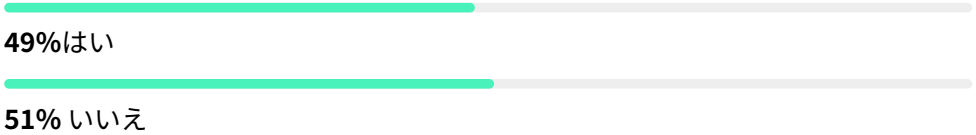
あなたの組織はマイクロサービスを採用していますか？



あなたの組織はKUBERNETESを使用していますか？



組織でローコードツールまたはノーコードツールを使用していますか？



DevOpsプラットフォームの役割

今年の調査では、回答者に初めてDevOpsプラットフォームの利用状況を質問しました。7割強のチームが「DevOpsプラットフォームを使用している」と回答しました（ただし、DevOpsプラットフォームが意味するものの定義は調査対象者に委ねました）。DevOpsプラットフォームの最大のメリットとしては、「DevOpsの向上」、「コラボレーションの改善」、「自動化の促進」、「可視性/トレーサビリティ」などが挙げられています。DevOpsプラットフォームがチームの役に立つ例として以下のような例も挙げられています。

「製品に関わるすべてのことに対するオーナーシップが強まる。」

「平均復旧時間（MTTR）の短縮、市場投入までの時間の短縮、修正のリードタイムの短縮、変更の失敗の減少。」

「信頼性、再現性、一貫性、生産性。」

当然とも言えますが、DevOpsプラットフォームを使用する可能性が最も高いグループはDevOpsチーム（43%）であり、23%は社内の「全員」がプラットフォームを使用していると回答しています。

開発者

開発における主な調査結果

DevOps = より高速なリリース

約60%の開発者が、DevOpsのおかげでコードのリリースが2倍速くなったと回答しています。

プロセスへの投資

2021年のDevOpsチームは、技術的に小さな変化を加えたのではなく、大規模なものを追加しました。SCM、CI/CD、DevOpsプラットフォーム、自動化されたテストなどです。

現状足りないものは？

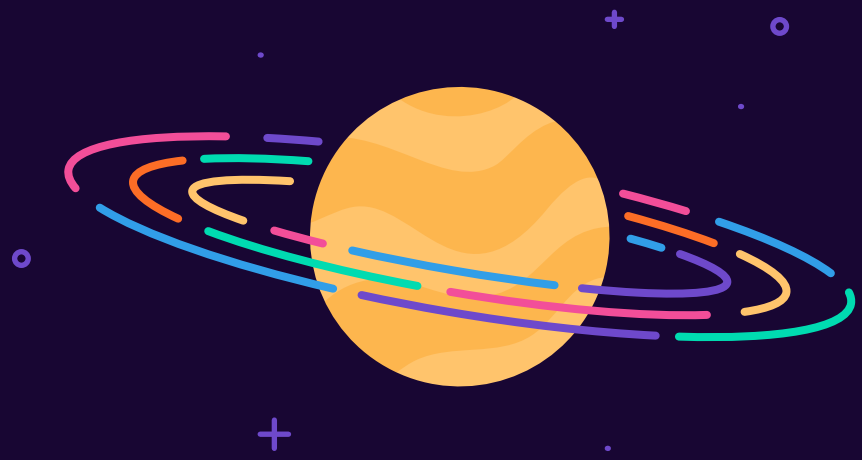
あらゆる種類のテストとより多くの（種類の異なる）コードレビューの実施。

役割の変化が継続

これまで運用部門が単独で責任を負っていた仕事を開発者が引き受ける傾向が続いています。

将来への展望

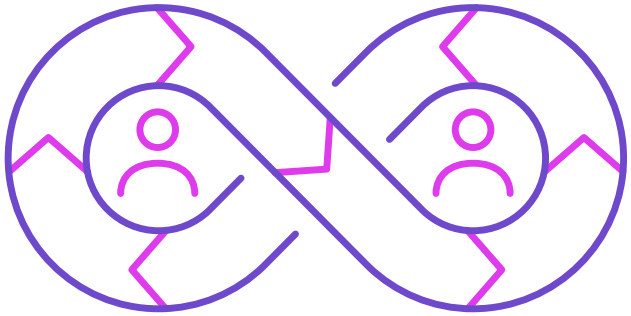
開発者の実に30%が、AI/MLの理解が自分のキャリアの次のステップに不可欠であると考えています。



開発とDevOps

（コードのリリースに）スピードが必要だと感じているなら、DevOpsは正しい選択であり続けます。84%以上の開発者が、以前よりも早くコードをリリースできるようになったと回答しています。約57%がコードのリリース速度が2倍になったと回答し（昨年の35%から大幅に増加）、19%がコードのリリース速度が10倍になったと回答しました。

なぜコードのリリースが早くなったのでしょうか？開発者の方々に、何が変わったのか聞いてみました。調査回答者の21%強が、DevOpsの実践にソースコード管理を加えたと答えており（昨年の15%から増加）、約18%がCIを、13%がCDを加えたと回答しています。12%近くが、DevOpsプラットフォームを追加することでプロセスがスピードアップしたと回答し、10%強が自動テストを追加したと回答しています。



ソフトウェア開発プロセスにどのような変更を加えましたか？

21.02% ソースコード管理

17.74% 継続的インテグレーション

13.59% 継続的デリバリー

11.65% DevOpsプラットフォーム

10.38% 自動テスト

5.29% ツールチェーンの統合

4.12% 計画ツール/方法論

3.95% クラウドネイティブ

3.52% サーバーレス

2.89% 要件管理

2.7% 自動セキュリティテスト/シフトレフトセキュリティ

2.38% リリースオーケストレーション

0.6% その他

DevOpsチームは新しいプロセスを追加していますが、同時に考え方の転換も行われています。開発者の皆さんに、コードをより早くリリースできるようになった本当の理由を深く追求してもらいました。

多くの人が、このプロセスは全体的な影響を持つものだと回答していました。

「開発段階でより多くのエラーや好ましくない慣行を捕捉することで、より頻繁に、より安全に本番環境にデプロイできます。」

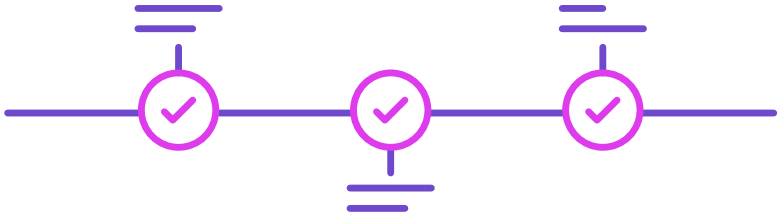
「私たちのチームでは、新しいプロジェクトでマイクロサービスを採用し、その後、継続的デリバリーを全面的に採用しました。継続的デリバリーを実現するためには品質保証が必要なため、自動テストを組み込んでいます。これらの分野に投資することで、以前は年に6回程度だった本番環境へのデプロイを、2000回まで増やすことができました。」

「特定の場所への自動デプロイではなく、グローバルにコードをリリースしています。主に、（バッチ処理をなくし）コミットからライブまでの時間を短縮することで、（スコープ拡大戦略による対応をなくし）間接費の少ない小規模な変更を促すことができました。」

「問題を分割して処理するようにしています。コードをより多くのモジュールに分割することで、デバッグ時間の短縮、安定性の向上、ミックス＆マッチのアプローチが可能になりました。」

「並行して開発されたバージョンの相互依存度が低くなるように、リリース計画を変更しました」。

「チームを評価し、バリューストリームマッピングで理想的な状態を確定しました。ほとんどのケースでは、チームが早い段階で対応できるよう、迅速なデリバリーと即時のフィードバックのために自動化されたパイプラインを必要としていることがわかりました。また、開発者がセキュリティ問題を迅速に解決できるように、早期の段階にセキュリティ対策を移動させました。また、開発者がプルリクエストを使って共同でコードレビューを確実に行うようにもしています。」



一方で、自動化も重要な課題でした。

「開発から本番環境まで完全に自動化されています。」

「自動セマンティックバージョニングを使って、リリースを高速化しています。」

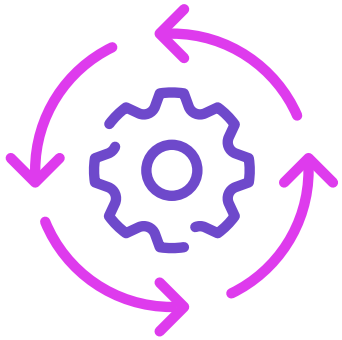
「可能な限りすべてを自動化して、製品を実環境に近い環境でテストできるようにしています。これにより信頼性が高まり、すべてのテストが簡素化されます。」

「手動でのデプロイを廃止しました。」

「CI機能を大幅に拡張したことで、デプロイ時間が大幅に短縮されました。」

「統合テストは、自信を持って自動リリースし、バージョン提供する上で大きなプラスとなりました。」いつでもデリバリーできるようになりました。」

「開発者はテストの実行状況を把握する必要がなく、プッシュするだけで、マスターに統合する前にパイプラインがコードをチェックしてくれるので助かります。」



「正直なところ、顧客への納期を早めるために継続的デリバリーが最も役立った。」

「プロジェクトの主要部分の構築プロセス、厳密なコードチェック、テストを自動化し、これら全てをGitLab CIに投入し、生産性の向上を実現した。」

「開発者がデプロイ前にコードのテストを忘れずに作成・実行することに依存しない。」

「あらゆる方法での自動化。」

これらの自動化により、開発者は以下のようなことをする必要がなくな

「手動でバックアップを作成し、手動でテストとデプロイを行う必要。」

「チートシートを書く必要がない。」

「手作業でのバックエンドコードのテスト。」

りました。

「セキュリティと依存関係の管理。」

「Rawコーディング。」

「厳密なチェックの実行。（git commitを使うと自動実行されます。）」

「コンテナの構築。」

「コードレビューに頼ってすべてのテストシナリオを捕捉すること。今では、カバレッジスキャンツールを使って、すべてが揃っているかどうかを判断しています。」

「待機。」

「長期にわたる意思決定不在の期間。」

「ローカル環境でのテストはもはや不要です。」

「面倒なセキュリティレビュー、何週間もかけて行うサーバーの設定、変更管理のための遅延。」

「開発チームはもはや本番環境デプロイを行わない。」

「開発プッシュのたびにブロックされることがなくなりました。」

「個々の変更についてのコミュニケーションではなく、GitLabをプラットフォームとして利用することで、より効果的なコラボレーションやコミュニケーションが可能になります。」

開発者が取り組んでいないことで、取り組みたいことは？多数のテストとコードレビューに始まり、非常に多数に上ります。

シフトレフトセキュリティ

ダイナミックテスト（例：RAM、メモリリーク、CPU）

より多くのコードレビュー（100倍）

より多くのテスト、より多くのコミットの自動化

マイクロサービス

より多くのオープンソース

パフォーマンスの最適化とアクセシビリティ

より早い段階での静的解析により、プルリクエストのフィードバックループを短縮

より自動化されたテスト。コードカバレッジ分析。インテリジェントなテストサブセットの選択による、ブランチの失敗の迅速化。

既にデプロイされているコード/ロジックを新しいプロジェクトで再利用

GitOps運用モデルへの切り替え

計画の改善、より詳細な要求の作成、より多くのステークホルダーの参加

ステークホルダーとの承認プロセスの改善

AIテスト

DevSecOpsプラットフォーム

コードを書くためのAI/MLの統合

TDD、BDD、モックに対するテスト、CI/CD

バグが修正されるたびに、システム的に回帰テストの追加が必要

機械学習とペアプログラミングの強化

非常に古いレガシーコードへの対応

「二度考えて、一度だけコード化する」

開発者の日常

2020年に始まったトレンドでは、開発者の役割はシフトし続けており、従来は運用部門の役割であった業務につきより多くの責任を負うようになっています。約26%が、書いたコードを本番環境モニタリングのために計測していると回答しました（昨年はわずか18%でした）。また、38%が、アプリケーションが動作するインフラを定義または作成しています。そのインフラを監視・対応するのは約13%です。

調査回答者の約45%がコードレビューを毎週行っていると答え、22%が隔週で行っていると回答しました（昨年の14%から増加）。しかし、チーム内でのコードレビューについては、コードレビューをまったく行わない開発者から、マージリクエストやチケット、プルのたびにコードレビューを行う開発者まで、さまざまな状況が見られます。毎日、または一日に何度もコードを見直していると多数が回答してくれました。当然ながら、約60%の開発者が、コードレビューはセキュリティとコード品質に関して「非常に価値がある」と答えています。コードレビューは、オンラインの「チャット」サービスを介して行われることが多く、開発者は、ブラウザよりもIDEでコードレビューを行う方がはるかに好ましいと回答しています。

開発者はコードレビューに時間を割いていますが、ツールチェーンの統合やメンテナンスにはあまり時間を割いていません。41%が毎月の作業に費やす時間は業務時間の10%未満と答え、20%がこれらの作業に11%〜20%の時間を費やしていると答えました。

開発の優先順位を決めるのは誰でしょう？今年は、開発者の43%が「自分で優先順位を決める」と答え（わずか24%がこう回答した2020年からは大きな変化）、38%が「プロダクトマネージャー」、19%が「事業部門」と答えました。業務や機能に優先順位をつける際、開発者にとって最も重要な優先事項は「開発コスト」（43%）で、次いで「開発者の作業量」（34%）、「製品ロードマップ」（31%）となっています。

セキュリティ

開発者の39%が組織内のセキュリティに全面的な責任を感じている（昨年の28%から増加）一方で、32%が他のチームと分担していると回答しています。75%が、自分の所属する組織では侵入を避けられると感じていると回答し、セキュリティに関する楽観的な傾向が見て取れます。

開発者は、チームの安全を保つためにどんな努力をしていると回答しているのでしょう？

> 「AI」
>
> 「セキュリティ担当のレッドチームとブルーチームが実行中のアプリケーションを監査し、品質ツールを使用し、依存関係をチェックしています。」
>
> 「コードレビューと自動テストを通して行っています。」
>
> 「設計段階でリスク分析をすることで、セキュリティでの問題を回避できます。」
>
> 「主にライブラリ、コンテナ画像、ホストの脆弱性スキャンを行います。」

「すべては開発者次第！」

「コードレビュー、セキュリティのベストプラクティスの監視と導入、頻繁なバックアップ、強固なパスワード。」

未来を見据えて

2020年からの顕著な変化として、開発者の30%が、AI/MLの理解が将来のキャリアにとって最も重要なスキルであると回答しました。昨年は22%で、ソフトスキルに次いで2番目でした。コミュニケーションやコラボレーションなどのソフトスキルは依然として重要で、最先端のプログラミング言語とともに、いずれも18%の回答者が挙げており、次いでGitOpsが14%、IoT/ブロックチェーンが11%となっています。

また、クラウド/クラウドネイティブ、クロスプラットフォーム開発、ローコード、データサイエンス、Python、暗号などについても知りたいという声が聞かれました。以下のコメントがこの傾向を特によく表しています。

> 「プラットフォームに責任を持つ開発者が大幅に増えると考えています。これは、DevOpsの概念が大きく、速くなったものです。例えば、マイクロサービスの世界を結びつけるためには、かなりの作業が必要となります。」

「現在、CI/CDパイプラインで静的コード解析を実行しているが
それ以上のことをする必要がある。」

「いくつかの高価なツールがあれば十分。」

セキュリティ

セキュリティに関する主要調査結果

DevSecOpsは本物

セキュリティ担当者の72%が、自社のセキュリティ対策を「強固」または「良好」と評価しています。希望的観測かもしれませんが、通常必ずしも前向きとは言えないグループの中でも、楽観的な意見が大幅に増えています。

シフトレフトの傾向は続く

DevSecOpsチームは、これまで以上に多くのDAST、SAST、コンテナ、および依存関係スキャンを実行しています。

セキュリティと開発には親和性がある

一方で、誰がセキュリティを「所有」しているかについてはまだ混乱しており、責任を避けあう傾向があります。

未来に向けて

セキュリティ担当者は、将来のキャリアで活躍するためには、専門知識が必要だと感じていますが、ほぼ同じ割合で、ソフトスキルが最も重要であると回答しています。関心が持たれている分野として高度なプログラミングとAI/MLがあります。

セキュリティとDevSecOps

2021年は、DevSecOpsが実現する年となるでしょうか？可能性はあります。セキュリティ担当者の72%が、組織のセキュリティ対策は「良好」または「強固」であると回答しており、前例のない結果となりました。これは、こう回答したのが59%にとどまった昨年と比べて大きな変化です。前年比で最も増加したのは「強固」カテゴリーで、昨年は回答者の19.95%しか自社のセキュリティ態勢をそのように考えていなかったのに対し、2021年には33%近くになっています。

役割は変化している

セキュリティ担当者が自分たちのチームの取り組みに自信を持っている理由の一つは、セキュリティの役割が進化し続けていることかもしれません。28%近くが「部門横断的なチームの一員になった」と答えています（昨年の結果と同様）。一方、26%が「コンプライアンスに力を入れるようになった」（2020年から4%近く増加）、24%が「日常的に携わるようになった/手をかけるようになった」（昨年からわずかに減少）と回答しています。自分の役割に変化がないと答えたのは約20%で、これは昨年の調査とほぼ同じ割合です。

組織のセキュリティに対する取り組みをどのように評価しますか？

39.52% 良好

32.51% 強固

20.25% 妥当

6.37% 貧弱

1.35% その他

あなたの経験では、セキュリティの役割はどのように変化していますか？

27.61% セキュリティに特化した部門横断的チームの一員になることが増えた

24.42% 日常的に携わるようになった/手をかけるようになった

25.89% コンプライアンスにより重点を置くようになった

19.99% 役割は変わらない

「セキュリティというのは、ほとんどの場合、一人の人間が専門とするものではないと思います。セキュリティは、フロントエンドの開発者からシステム管理者まで、チームのすべてのメンバーが（技術者以外の役割も）実践する必要があります。」

シフトレフト

もうひとつ明るい兆しがあります。セキュリティもシフトレフトし続けており、そのペースはこれまでにないほど速いものです。セキュリティ担当者の70%以上が、シフトレフト（開発プロセスの初期段階にセキュリティを移動させること）にチームが移行したと回答しており、昨年の65%から増加しています。

しかし、もっと掘り下げてみると、不思議な二分法が明らかとなります。スキャンは確かに増加しました。現在、53%の開発者がSASTスキャンを実行しており（昨年の40%未満から大幅に増加）、44%がDASTスキャンを実行しています（昨年の27%から大幅に増加）。また、50%以上のセキュリティ担当者が、開発者がコンテナのスキャン、依存関係のスキャン、ライセンスコンプライアンスのチェックを行っていると報告しています。

しかし、より多くのスキャンが実行されているにもかかわらず、開発者はほとんどの結果を簡単には入手できません。実際、SASTの軽量スキャナをWeb　IDEに入れているチームはわずか23%で、スキャン結果を開発者向けのWebパイプラインレポートにまとめているのは20%に過ぎません。DAST、依存関係、コンテナのスキャンについては、さらに悪い結果となりました。DASTと依存関係スキャンが簡単に利用できるのは16%で、コンテナスキャンについては14%です。この結果は、2020年に比べてほとんど改善されていないことを示しています。昨年は、SASTの結果を開発者向けのレポートに記載している企業は19%未満、DASTでは14%未満でした。

セキュリティにおいて、早期のバージョンでのテストが機能するためには、開発者がIDEを使っている間に結果にアクセスできるようにする必要があります。

また、セキュリティチームと開発チームはときに対立することがありますが、これも進行中の課題です。今回の調査でも相変わらずお互いを非難する声が上がっていますが、驚くべきことに、その割合はこれまでよりもずっと低いものでした。昨年の調査では、93%のセキュリティ担当者が、開発者が既存のコードで発見可能なバグの25%以下しか発見できていないと回答しています（つまり、バグの4分の3はそのままで、後でセキュリティ担当者が発見）。今年、同じコメントをしたセキュリティチームのメンバーはわずか45%で、前代未聞の37%が、開発者が実際にバグの4分の1から2分の1を発見していると答えています。

また、セキュリティ担当者の83%は、バグの発見が開発者のパフォーマンス指標であると、ある程度同意していますが、ほぼ同じ割合（81%）の人は、開発者にバグ修正を優先させるのは難しいと訴えています。最終的には、77%のセキュリティ担当者が、コードがテスト環境にマージされた後にバグを発見するのは大抵開発者ではなく自分であるということにほぼ同意しています。

誰が責任者なのか？

セキュリティの「オーナーシップ」はほぼすべての組織で依然として厄介な問題であり、特にセキュリティチームに関してはその傾向が強いようです。約31%が「自部門（セキュリティ）に全責任がある」と答えましたが、約28%は「全員に責任がある」と答えました。この回答は昨年のものと酷似しており、この問題を明確にする必要があることを示しています。

あなたの組織では、どのグループが主にセキュリティに責任を負っていますか？

バグについて

セキュリティテストは、チームメンバーの間でも問題視されています。42%以上がテストがプロセスの後半で行われていると回答し、ほぼ同じ割合が、脆弱性の解明、処理と修正に苦労していると答えています。約37%がバグ修正の状況を把握するのが大変だと答え、33%が修正の優先順位をつけるのが大変だと答えました。最後に、32%が「問題を解決してくれる人を見つけるのが難しい」と答えています。

バグが発見されたとき、最も重要なのは「深刻度」（59%）で、次いで「カテゴリー」（51%）、「解決した脆弱性の数」（41%）、「発見からの経過時間」（37%）となっています。

マイクロサービスやコンテナを含む最新のアプリケーション開発戦略はますます普及していますが、それらを監視・保護するためのプロセスを導入していると回答したセキュリティ担当者は約半数にとどまりました。モニタリングツールを使用しているチームは、Prometheus、Grafana、AWS Watchを最も多く挙げています。

> 「開発者は独自の観測スタックを持っています。」
>
> 「本当に [プロセスが] 整備されていません。」
>
> 「私たちのフラッグシップは、ユーザーの入力をほとんど受け付けないクローズドなシステムのように機能し、App Engineの上で動作するため、コンテナは実質的に自己保護されています。」

しかし、クラウドネイティブやサーバーレスに関しては、セキュリティに関する見通しが少し明るくなっています。昨年は64%の回答者が組織にクラウドネイティブやサーバーレスのセキュリティを確保するための対策が何もないと答えましたが、今年は53%のチームがそれを組み込んでいます。

未来を見据えて

将来のキャリアに何が最も役立つかについては、セキュリティ担当者の意見はほぼ均等に分かれました。22.95%が「専門知識」、22.91%が「コミュニケーションやコラボレーションなどのソフトスキル」と回答しました。（2020年にはソフトスキルが1位でしたが、今年の変化は、パンデミックの影響でコラボレーションの改善が必要となり、他のものに注力する余地ができたことを反映しているのかもしれません。）約21%が「高度なプログラミング」と答え、ほぼ同じ割合の回答者が「AI/ML」と答えています。

「セキュリティ知識、ペンテスト、バグ報奨金、Linuxに関する十分な知識、使用されるテクノロジー、およびランタイム（コンテナ、サーバーなど）。」

> 「クラウドとサーバーレスアーキテクチャ。」
>
> 「クラウドスキル。」
>
> 「マルウェア分析と脅威インテリジェンス、侵入テスト。」
>
> 「自動化とDevOps」

運用

運用における主な調査結果

DevOps = 変更

62%以上の人が、DevOpsのおかげで新しい、または異なる責任を負うことになったと報告しています。

クラウドとクラウド以外

運用担当者の半数以上は、主にクラウドサービスを管理していますが、ほぼ同じ割合で、ハードウェアやインフラの管理にも力を入れています。

自動化が現実に

運用チームの約19%が完全な自動化を達成しているとしており、37%が「大半」の自動化を達成しています。

今後...

運用担当者は、プログラミングが最も重要なスキルになると考えており、ソフトスキルを重視する2020年から大きく変化しています。

運用

過去、現在、そして未来の技術や方法論の門番役を務める運用部門では、このような苦労がつきものです。驚くことではありませんが、運用部門では役割が急速に進化し続けており、運用担当者はDevOpsがその理由だと語っています。

これらの変更はどのようなものになるのでしょうか？

「サーバーのプロビジョニングから人の管理まで、あらゆることに対応しています。それらの業務に共通するのは、日々の仕事をするための自動化プラットフォームの構築です。」

「ソフトウェア開発のための会社のロードマップを計画し、開発者チーム全体を管理し、研究開発の取り組みを検討しています。」

「DevOpsツールチェーンのツールを運用可能な状態に維持し、プラットフォームとプラクティスの改善を続けています。」

> 「DevOpsのコーチ役を務めています。」
>
> 「私はプラットフォームエンジニアです。」
>
> 「なんでも屋的にあらゆるものに少しずつ関わっています。」
>
> 「DevOps、SREの監視に加え、プラットフォームの動作を担保しています。」

現在、運用担当者の49%が、自分の役割は主にハードウェアとインフラの管理であると考えており（昨年の42%から大幅に増加）、56%が最優先事項はクラウドサービスの管理であると答えています（昨年から4ポイント増加）。

また、2020年に比べてコンプライアンスにも時間をかけています。昨年は55%以上が監査やコンプライアンスの問題に対処するためにほとんど時間をかけていない（業務時間の10%以下）と回答していましたが、今年は時間をかけていないと回答したのはわずか36%でした。実際、29%が監査やコンプライアンスの問題に対処するために約4分の1の時間を費やしていると答えています。

依然としてツールが過多

そして、それは多くのツールがなければ操作できません。運用担当者の約50%は、チームで2つから5つのモニタリングツールを使用していると回答しました（昨年の65%から大幅に減少）が、28%はモニタリングツールをまったく使用していません。全体では運用チームの72%が開発者にリアルタイムのデータを簡単に提供できるツールを使用しています（そのうちの40%強は、組織がDevOpsプラットフォームを使用しているからだと回答しています）。

最も重要な監視カテゴリーはロギング（アプリケーションログの収集と表示）で、次に指標が続きます。

ロギングツールのトップはElasticsearch（28%）、次いでSplunk（21%）、Datadog（19%）となっています。（昨年は、運用チームの38%がElasticsearchを使用し、14%がSplunkを、12%がDatadogを選択していました。）アプリケーション指標のトラッキングに関しては、21%がDatadogを、14%近くがNew Relicを使用しています。しかし、20%近くは指標管理ツールを使っていません。Prometheusは、2年連続で時系列の指標を取得するためのツールとしてトップに選ばれています。運用チームの30%がこのツールを使用しており、次いでDatadogが約23%となっています。

ログの表示/収集にどのようなツールを使っていますか？

	2020年	2021年
Elasticsearch (Elk Stack)	**37.51%**	**28%**
その他	**7.52%**	**該当なし**
Splunk	**13.64%**	**20.58%**
Datadog	**11.71%**	**19.3%**
なし	**12.44%**	**11.92%**
Sumologic	**2.95%**	**6.83%**
Logz.io	**3.5%**	**5.99%**

運用チームの大多数（約34%）がAWSクラウドを利用しており、24%がMicrosoft Azure、23%がGoogle Cloud Platformを利用しています。パブリッククラウドを利用していないか、利用しているパブリッククラウドがわからないと答えたのは約13%でした。2020年から最も大きな変化があったのはAzureです。昨年、マイクロソフトのクラウドサービスを利用していたのは18%未満にすぎませんでした。

アプリの指標（トレース）にはどのようなツールを使っていますか？

	2020年	2021年
DataDog	**14.51%**	**21.02%**
なし	**29.12%**	**19.46%**
New Relic	**13.68%**	**13.79%**
Jaeger	**11.73%**	**11.6%**
Zipkin	**4.94%**	**10.67%**
AppDynamics	**6.38%**	**10.15%**
Dynatrace	**5.56%**	**7.86%**
その他	**14.09%**	**5.46%**

時系列の指標を取得するためにどのようなツールを使用していますか？

	2020年	2021年
Prometheus	**35.66%**	**30.06%**
Datadog	**12.18%**	**22.98%**
Nagios	**12.57%**	**12.38%**
なし	**16.21%**	**12.17%**
Solarwinds	**5.4%**	**8.97%**
Sensu	**3.44%**	**8.76%**
その他	**14.54%**	**4.67%**

開発部門との協業

運用チームの55%以上が、ソフトウェア開発ライフサイクルを完全に、またはほとんど自動化していると回答しています。27%強が「部分的に自動化されている」、11%が「自動化が始まったばかり」と答えています。約6%が「全く自動化されていない」と答えています。2020年には、完全に自動化されているとするチームはわずか8%でしたが、今年は19%近くがそう回答しています。

例えば、開発状況を把握できることが非常に重要である、または極めて重要であると回答した運用担当者は71%、開発プロジェクトをサポートするために十分な通知を受けることがある程度できると回答した人は79%となっており、依然として運用と開発の関係が安定している面もあります。しかし、運用担当者の約77%が、開発者がテスト環境を用意できるようになったと回答しており、これは昨年より8%増加しています。また、実際にDevSecOpsが実践されてもいます。運用チームの76%以上が、開発者が開発プロセス中にセキュリティ問題に関する通知を受け、これに対処できるという点にある程度同意しています（昨年より10%増加）。

運用担当者は、組織内のセキュリティに対する責任をますます感じています。チームが単独で責任を負っていると答えたのは28%（昨年の21%から増加）ですが、大多数（34%）は、より大きなチームの一員として責任を負っていると考えています。

「多様なプラットフォームやツールの統合」

「スクリプト、基本的なハードウェアとネットワーキング、クラウドの基本」

未来を見据えて

世界的なパンデミックの影響とリモートワークの課題は、DevOpsのすべての役割で明確に感じられ、運用も例外ではありません。昨年の調査では、開発、セキュリティ、テストの担当者と同様に、運用担当者の過半数近く（31%）が、コミュニケーションやコラボレーションなどのソフトスキルが将来のキャリアにとって最も重要であると考えていると回答しました。今年は、25%が「プログラミング」が最も重要であると答え、次いで「ソフトスキル」（24%）、「専門知識」（21%）、「AI/ML」（20%）が続きました。

将来については、以下のような意見も聞かれました。

「システム管理のスキルは未だに大きな武器になる」

「分散化、倫理的なアーキテクチャとセキュリティ」

「世界が本当に変化しているとは言えないが、クラウドに関してはそこそこ良好です。しかし、コンピューターネットワークやOSの基本を知っていれば、大きな強みになります！ハードスキルはもちろん必須ですが、コミュニケーションは時に大きな問題になります。」

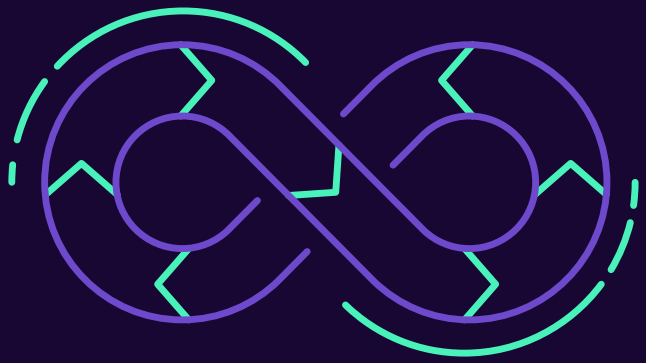
DevOpsの勢いを維持する

2021年の投資計画を尋ねた調査回答者からは、DevOpsの「勢いが増している」状態が、はっきりと伝わってきました。昨年、DevOpsチームは自動化、CI/CD、DevOpsといった基本的なことに注力する計画でした。それが功を奏したようで、今年の優先事項には顕著な違いが見られます。

2021年には、調査対象者の大半がクラウドに投資を集中させ、次いでAIに投資するとしています。それに比べて、昨年はクラウドが4位に選ばれ、AIは8位にすぎませんでした。今年の3位と4位には「自動化」と「DevOps」が選ばれました。さらに、調査対象者からは機械学習やデータサイエンスに対する興味も見られました。

DevOpsチームが、前に進むために欠かせない真の（そして大変な）努力を重ねていることは明らかです。そして、大多数の人が今後の展開に備えているようです。調査対象者の48%が将来に向けて「やや準備ができている」と回答し、27%が「十分に準備ができている」と回答しました。「手に負えないと感じる」と答えたのはわずか6%で、昨年の出来事を考えると比較的少ない数字です。

これらの観察結果をチームで共有し、自分たちの取り組みと比較してみてください。DevOpsは目的地ではなく旅のようなもので、その勢いを維持することが大切です。




GitLab