

DevOps成功への 道のり

データ収集からインサイト獲得まで
段階的に進みDevOpsを成功に導く

デジタルエコノミーはインターネットのようなスピードで推移しています。したがって組織も、ソフトウェアの開発と提供を俊敏でリユース可能なものにするによって、新しいソフトウェアや分散サービスをこれまで以上に速く提供し、改良し続けなければなりません。小規模なアップデートを頻繁に行うようにすれば、開発者は新しい機能をすばやく試すことができます。

頻繁なアップデートには次のメリットもあります。

- ・アプリケーションが顧客にとっていつも斬新で、最初に思い浮かぶものであり続けるようにする。
- ・脅威や業務ニーズの変化に迅速に対応する。
- ・重要なものを壊してしまうリスクを抑える。

アプリケーションを短いサイクルでリリースすることは、デジタルビジネスでは一般的であり、クラウドサービスとサービス指向のワークロード、モバイルアプリ、ソーシャルコミュニケーションチャネルなどのテクノロジーによって、組織の形も変わりつつあります。サイクルの短縮化によって、新しいアーキテクチャや設計パターン、アプリケーションスタック、さらには継続的デリバリーと呼ばれるリリーストレインも登場しています。

DevOpsは、コラボレーションの文化をベースとするフルライフサイクルアプローチであり、先進的な企業の多くが、アプリケーションを開発、提供、サポートする優れた方法として採用しています。DevOpsを採用した企業は、物事をスピーディーに進め、変化し続けるデジタルの世界で顧客のニーズに応え続けています。

しかし、DevOpsに力を注ぐ企業の夢と希望は、組織の現実に阻まれてしまうことも少なくありません。DevOpsが必ずしも期待に応えることができていない大きな要因は、技術面や組織面での複雑さにあります。データに基づく応答性の高い俊敏な文化と、それを支えるプロセスを創り出すことは、案外難しいものです。その道のりにある多くの課題の中でも、インフラとアプリケーションの複雑化、アプリケーションのテレメトリ、そしてデータ分析は、卓越したDevOpsを実現する上でしばしば障害となります。

データは単なるログ以上の価値を持っています。データは、アプリケーション、クラウド、インフラ、コンテナ、モバイルアプリ、データベース、IoT、ストリーミングネットワークデータなどから発生し、あらゆるアプリケーション、システム、顧客、API、産業システムのアクティビティとパフォーマンスが、ログ、メトリクス、イベントとして記録されます。

卓越した DevOps を実現するには、従来のパフォーマンス監視ではもはや不十分です。すべてのデータを分析して、開発者と運用担当者が問題への対応よりもビジネスの成功に集中できるようにすることが必要です。データを活用すれば、テクノロジスタック全体を踏まえた解決策をリアルタイムで見出し、ユーザーへの影響をすばやく把握して、ビジネスに影響が及ぶのを防ぐことができます。

DevOpsの現実と8フェーズのサイクル

DevOpsは、サービス指向の流動的なワークロードと運用に、継続的な反復を重視するアジャイル開発手法をもたらします。この手法では、コードを定義、テスト、デプロイ、観察、測定し、問題が発生した場合はコードの修正を行って、プロセスを再び開始し、同じことを繰り返していきます。

「コード」には、アプリケーションコードだけでなく、宣言型のアーキテクチャ、ネットワーク設定、監視なども含まれます。このプロセス全体で必要となるのが、スピードと俊敏性です。また、ソフトウェアを大々的にリリースするのではなく、漸進的な改善と「早い段階での失敗」を受け入れる企業文化も必要になります。

DevOpsのフィードバックループを成功させる上で鍵となるのが、使用状況、パフォーマンス、エラー、重要なビジネス指標を測定、分析したデータです。オブザーバビリティプラットフォームは、あらゆるタイプのデータを取り込んで、AIを活用した高度な分析機能によって答えを導き出し、レポートを作成する柔軟性を備えているため、DevOpsプロセスに最適です。

俊敏性はDevOpsの基本ですが、複雑さは俊敏性を損ないます。DevOpsプロセスには8つのステップがあり、多くの場合、各ステージで異なるツールが使用されます(図1)。こうしたツールはオープンソース製品や特定用途向けの製品であることが多いため、製品間の統合が難しく、重複や効率の悪さのために可視性や俊敏性が低下しがちです。そして、俊敏性が低下すると、問題の発見や解決に必要な以上に時間がかかり、生産性の低下とコストの増加につながります。



図1：一般的なDevOpsプロセスの主なステップ

DevOpsのステージはよく、従来の2種類のプロセスオーナーの相互関係をうまく表した無限大記号で描かれます。左側はソフトウェアの開発とテストのサイクル(Dev)、右側は導入と運用のサイクル(Ops)を表しています。ただし、サイクルのステップごとに特有のツール製品市場が存在し、各ステージでは一般に数十もの製品が使用されます。

このように選択肢が豊富であることは、開発チームと運用チームがそれぞれニーズと好みに合った製品を選ぶため望ましいことですが、その結果もたらされる混乱は、プロジェクトマネージャーやDevOpsのリーダー、企業の経営層にとってはまさに悪夢です。DevOpsシステムが混在し、それぞれが分離されたままデータを排出する状態では、プロセス全体の可視性が失われ、ツールが管理不能に陥りかねません。さらに、開発チームと運用チームの間で使用する用語に食い違いが生じたり、トラブルシューティングの流れが分断されたりして、問題発生時のMTTR（平均解決時間）が延びるというデメリットもあります。

このようにツールが分散して秩序が失われることは、「修正」によって根本的な問題を解決できたのか、それとも一時しのぎのパッチを提供できただけなのかを見極める方法がないことを意味します。コードの開発サイクル全体、そしてリリースや使用状況に関するデータを詳細に可視化できなければ、アプリケーションの品質やパフォーマンス、セキュリティを検証することはできません。たとえば、セキュリティ監視ソフトウェアは脆弱性を特定することができますが、その問題をソースのソフトウェアモジュールにまで遡る手段がなければ、脆弱性の修正は困難です。

さらに、DevOpsのメトリクスを可視化できていない企業は、ビジネスの状況の変化に応じて新しいアイデアを試したり、反復したりするために必要な俊敏性を持つことができません。その結果、品質が低下して、顧客の不満や顧客離れにつながる恐れもあります。

DevOpsでは、ビルドパイプラインがばらばらな状態だと、アプリケーションのデリバリープロセス中に時間のかかるハンドオフが何度も行われることになります。これは、余計な機能が導入されるリスクを生じさせ、タスクの切り替えによって効率が低下して、リソース全体を無駄にすることにもなります。コードの編集作業はいったん中断されると、再開には平均で10～15分かかるとい調査結果もあります*。

通常のトラブルシューティングプロセスでコンテキストの切り替えが何度も必要になると、開発者の生産性が大幅に低下します。さらに、デリバリーワークフローが分断されていると、十分なコラボレーションも難しくなります(各チームが異なる用語を使い、プロジェクト全体について異なる見解を持っているために、部署間でギャップが生じるからです)。こうして、コミュニケーションや情報の共有に摩擦が生じ、信頼や協力関係が損なわれて、DevOpsの根本的な前提が覆されることになります。

データに基づくフィードバックを提供するプラットフォームがなければ、DevOpsの成功はおよそありえません。

DevOpsの価値を最大限に引き出す

経営者の間ではよくDevOpsが流行語のように口にされますが、理解不足のために現実離れした期待を抱くケースも多く見られます。DevOpsがもたらす成果について大きな期待を持っている一方で、原則やプロセス、課題について正しく理解していなければ、失望という結末が待っています。

データに基づくフィードバックを提供するオブザーバビリティプラットフォームがなければ、DevOpsの成功はおよそありえません。この落とし穴を避ける唯一の方法は、メトリクスを活用してDevOpsの成果の測定と調査、実証を行い、さらにデータから得られたインサイトを活用して、DevOpsの最適化や改善に長期的に取り組むことです。

DevOpsの真価を得るには、流行だからと漠然と導入するのではなく、ソフトウェアの品質とデリバリーについて共通の言語とアプローチを確立する必要があります。

DevOpsの活用

成功する企業はたいてい、顧客のニーズを理解しています。顧客のニーズを把握するには、絶えずフィードバックを提供することによってIT、アプリケーション開発者、経営層、セキュリティチーム、監査役といった企業のあらゆる領域を支援するオブザーバビリティプラットフォームが必要です。オブザーバビリティプラットフォームは、事実に基づくインサイトを提供して意思決定をサポートすることで、企業が物事をスピーディーに進め、コストを削減し、新しいアイデアを取り入れるとともに、足かせとなっているアイデアを捨てることができるようにします。

オブザーバビリティプラットフォームを活用するには、すべてのワークロードに関するインサイトをすべての関係者に継続的に提供するためのデータ、ツール、プロセスが不可欠です。こうしてDevOpsを確立すれば、事業部門、アプリケーション、IT運用のすべての領域において、インサイトに基づいてすばやく行動できます。

実際に、DevOpsモデルを、AIを活用した包括的な分析と組み合わせることができれば、アプリケーションデリバリーの迅速化、アプリケーションの品質の向上、ビジネスの質の向上という3つの重要な利点を手に入れることができます。

* <http://blog.ninlabs.com/2013/01/programmer-interrupted/>

結果をすばやく出すために重要な4つの能力

DevOpsチームが俊敏性と顧客の満足を獲得するためには、4つの能力が鍵となります。

- **オブザーバビリティ**：DevOpsのビルドとデリバリーのワークフローに含まれるすべてのコンポーネントを、マスターリポジトリに収集されたデータを使用して計測し、さらに詳しい分析を行う必要があります。データなくして問題を理解し修復することは不可能です。
- **反復**：ツールチェーン全体にわたって相関付けられたデータから深いインサイトを獲得し、それに基づいて、コードの修正点と改善箇所を迅速に特定してトリアージし、開発する必要があります。
- **コラボレーション**：迅速なデリバリーのためには、DevOpsチームが同じ考えを持っていること、同じデータを使用していること、同じ測定値に基づいて行動を起こすことが必要です。
- **最適化**：リーダーには、事実に基づいて意思決定することによって、常にプロセスを改善しようとする姿勢が求められます。プロセスを最適化するには、データに基づいて次のような質問に答えられることが必要です。
 - どのくらいの割合でアイデアを利益に結び付けることができているか。収益を上げるまでのエンドツーエンドのスループットは、どのくらいか。
 - デリバリーパイプラインの各フェーズには、どのくらい時間がかかっているか。
 - コードの記述、テスト、レビューにチームはどのくらいの時間をかけているか。
 - どの開発チームの生産性が最も高いか。

データに基づくDevOpsの利点は直接ビジネスの成果となって現れます。効率や開発者の生産性の向上、アプリケーションデリバリーにかかる時間の短縮、コストの削減、顧客満足度の向上、収益の増加によって、ビジネス全体が改善されるのです。

リアルタイムのオブザーバビリティインサイトによってより質の高いアプリケーションを構築

DevOpsのツールチェーン全体から得られたデータが提供する情報によって、開発とテストのサイクルの早い段階で問題にプロアクティブに対応できるようになります。オブザーバビリティを実現すれば、開発チームと運用チームは、機能の問題やアプリケーションのクラッシュによって顧客やユーザーが不満を感じる前に、問題の解決策を見出すことができます。こうしたプロアクティブな対応を行うには、DevOpsチームを含む組織の全員が「唯一の真実」として使用できる、共通のオブザーバビリティプラットフォーム

が必要です。それは、ソースコード制御システムによって、開発者が成果を統合し、共有できることに似ています。

DevOpsのプロセス全体を正確に計測して分析することができれば、最終製品とデリバリープロセス全体を改善するために重要な実際のパフォーマンス、使用状況、エラーデータが明らかになります。さらに、データプラットフォームが一元化されていれば、ツールとインフラ全体で情報を相関付け、問題をプロアクティブに警告できます。たとえば、コードチェックのデータをパフォーマンス監視システムと関連付けることができれば、ユーザーがバグをレポートする前に問題を明らかにすることができます。

DevOpsのデリバリーサイクル全体のデータを統合するためには、前述した8つのフェーズサイクルで使用されるすべてのツールから、データをリアルタイムで取り込むことのできるプラットフォームが必要です。開発ツールは常に変化しており、データも常に変化しています。

データは、DevOpsで測定を行うための原材料です。ただし、コードが機能面と運用面の仕様、さらには品質のSLAを満たしているかどうかを数値化する、客観的なメトリクスから得られたデータである必要があります。コードのデプロイ後、各リリースがデリバリーパイプラインを流れていきますが、チームはその過程でアプリケーションの内部で何が起きているかを知る必要があります。それが分からないと、パイプラインを理解したり、アプリケーションのパフォーマンスやエンドユーザーエクスペリエンスとパイプラインのイベントを相関付けることができません。リアルタイムのオブザーバビリティが得られるインサイトを活用してアプリケーションデリバリーサイクル全体を分析すれば、次のような成果を期待できます。

- 開発チームとコードリリース全体のバグの割合を、測定と追跡が可能な一貫した方法でベンチマークする。
- 品質保証のメトリクスを可視化して、本番リリースの前に問題を特定する。
- 分散が進むマイクロサービスの状況をすばやく把握して、問題がいつどこで発生したかを特定する。
- クラウドのコストを最適化する。
- ストリーミングデータにAIを適用することで、アラートのスピードと精度を高め、インスタントトラブルシューティングによって問題解決を効率化し、インサイトをすばやく獲得する。
- DevOpsのデリバリープロセスにセキュリティを密接に統合して、近年注目されつつある「DevSecOps」を実現する。

開発チームは、ビルドサイクルの早い段階で脆弱性を特定して排除することにより、コードのセキュリティとコンプライアンスを向上できます。また、セキュリティチームは、アプリケーションのビルドパイプラインからデータに基づくインサイトをリアルタイムで得ることにより、セキュリティの問題があるコードがコードベースにチェックインされる前に、

リスクとなる挙動を特定し、適切なチームにアラートを送信できます。コードのセキュリティを確保するためのプロセスを自動化すれば、開発プロセスでセキュリティを最優先する体制を築くことができます。セキュリティを最優先することは、最新のアプリケーションに不可欠な考え方です。

ビジネス成果の向上

データに基づくDevOpsとオブザーバビリティ実現の利点は直接ビジネスの成果となって現れます。効率や開発者の生産性の向上、アプリケーションデリバリーにかかる時間の短縮、コストの削減、顧客満足度の向上、収益の増加によって、ビジネス全体が改善されるのです。

包括的オブザーバビリティプラットフォームの活用範囲を従来のIT運用の枠から広げると、次のような成果を見込めます。

- 新しくリリースしたアプリケーションの使用状況、パフォーマンス、信頼性、エラー、セキュリティインシデントをリアルタイムで可視化できます。アドホックレポートを使用するときのように、把握できるのが数時間後や数日後になることはありません。
- MTTRを大幅に短縮できます。問題の解決にかかる時間は、本番環境で最大70%、本番前の場合で40%短くなります。
- データの取り込みと分析が自動化されるため、効率が向上します。開発チームと運用チームは、ビジネスニーズに応えることに集中でき、監視ツールの導入や保守に時間を無駄にせずに済むようになります。

DevOpsチームの足並みをビジネス目標と確実に揃えるためには、絶えずデリバリープロセスを進めながら頻繁にリリースを行い、そのリリースを測定するとともに、実際のビジネスの結果と関連付ける必要があります。

リリースプロセスを短期化すれば、コードのパフォーマンスと有用性を高め、カスタマーエクスペリエンスの質を高めることができます。コードがよりよいものになれば顧客の満足度が高まり、それによって顧客のロイヤルティが高まります。オブザーバビリティプラットフォームが提供するデータを利用すれば、アプリケーションの使用状況や売上などのメトリクスを分析して、顧客満足度をコードリリースと直接結び付けることができます。オブザーバビリティで可視化できるビジネスメトリクスには、ほかに次のようなものがあります。

- 顧客のサインアップ率とダウンロード率
- 収益の変化(製品のリリースに伴う山と谷など)
- 顧客のエンゲージメントとセルスルー、カート放棄率における変化

分析機能がDevOpsの利点を数値化するとともに強化

デジタルの世界で成功するためには、俊敏性が不可欠です。俊敏性は、DevOpsのリリースプロセスを効率化することによって達成することができます。また、DevOpsプログラムの利点を最大限に引き出すにはオブザーバビリティプラットフォームが必要です。

DevOpsを実践する企業は必然的に、対応力やコラボレーション、セキュリティ、規制へのコンプライアンスの継続的な改善に取り組むことになります。これらのメトリクスは、革新的な製品を短期間で開発して企業の評価と顧客満足度を高めることに直接つながるからです。しかし、DevOpsを導入する際には多くの課題に直面しがちです。その要因が複雑なプロセスやツールの分散にあることが少なくありません。セキュリティデータ、システムのパフォーマンスデータ、アプリケーションのデータを別々のツールで管理していると、データの関連付けは困難になります。また、経営層、ITリーダー、開発チームがDevOpsの利点とその実現方法を理解できるよう、分かりやすい方法で説得することも重要です。

DevOpsを成功させる秘訣は、数値によって検証を行うことです。「数値化することができなければ改善することはできない」という格言は今も通用します。そして、統合的オブザーバビリティプラットフォームこそ、測定と評価のための理想的なシステムであるといえます。

DevOpsは多大なメリットをもたらしますが、そのメリットを享受するには適切に運用する必要があります。DevOps導入は短距離走ではなく長距離走なのです。DevOpsを真の意味で実現するには、ソフトウェアの開発方法だけでなく、リリース、運用、監視、トラブルシューティング方法も見直す必要があります。多くの変更が求められますが、DevOpsの実現によって生産性、開発者と顧客の満足度、リリース速度、信頼性が向上することは明らかです。

今日、DevOpsを成功に導くには、ソフトウェア開発プラクティスの基本要素としてオブザーバビリティを導入することが不可欠です。オブザーバビリティは、各ワークフローの状態とそのコンテキストを可視化し、理解するために役立ちます。ワークロードのクラウドネイティブ化が進み、複雑さが増す中、DevOpsチームにとってオブザーバビリティを実現することは、アプリケーションの状況をリアルタイムで把握し、ビジネス関係者に重要なインサイトを提供して、「未知の未知」を克服するための強力な武器となります。詳しくは、[Splunkの『オブザーバビリティ\(可観測性\)ピグナーガイド』](#)をお読みになるか、[Splunk Observability Cloud](#)のトライアル版をお試しください。

[Splunk for DevOps](#)の詳細をご覧ください。Splunkの[無料体験版](#)をお試しください。



営業へのお問い合わせはこちら：https://www.splunk.com/ja_jp/talk-to-sales.html
〒100-0004 千代田区大手町1-1-1 大手町パークビルディング 8階

www.splunk.com/ja_jp
splunkjp@splunk.com