

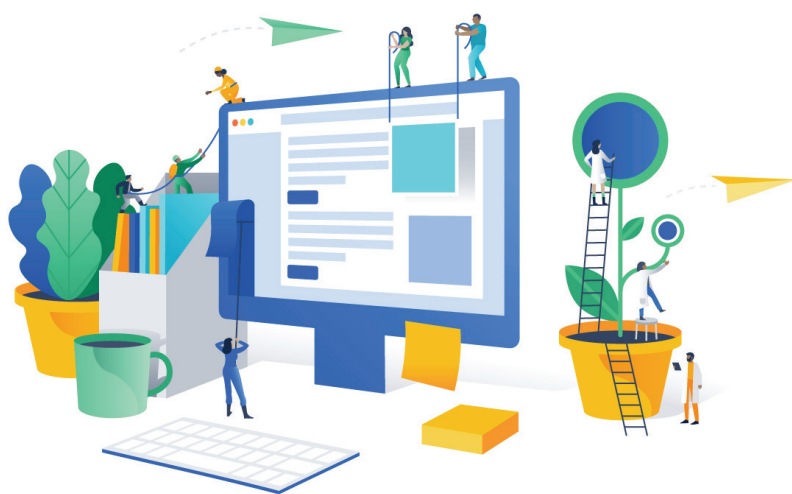
DevOps成功のための4ステップ

成功を収めるDevOps チームは、DevOps が1 人だけの仕事ではなく、全員の仕事であることを理解しています。ここでは、組織でのDevOps 成功に向けて、適切なチーム、文化、プラクティスを構築するためのヒントを紹介します。

ボーナス ヒント:このガイドを開発チームと運用チームで共有しましょう

1 コラボレーション文化の構築

責任の共有、透明性、およびコラボレーションは、高パフォーマンスチームの基盤です。文化というものが「ここでのやり方」として定義される場合、DevOpsの文化の構築を開始する方法は以下のとおりです。



・ チームで相互交流する

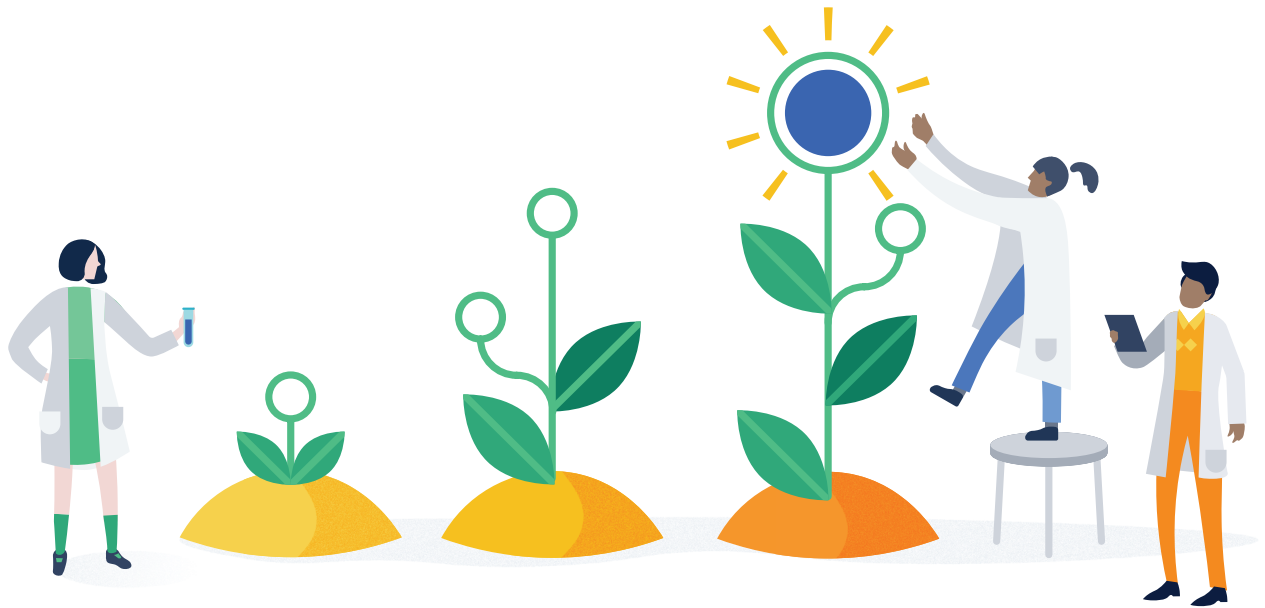
運用チームのメンバーを、開発チームのスプリント計画、日次のスタンドアップ ミーティング、およびスプリントデモに参加するように招待したり、逆に開発チームのメンバーを運用チームに招待したりしましょう。これは、互いのプロジェクト、アイデア、努力の動向を把握するための迅速かつ本質的な方法です。専門知識の聞き取りと共有に費やされる時間は、リリースおよび緊急時のトラブルシューティングの効率が高まることで相殺できます。

・ チームで問題を解決する

開発者、運用担当者、および顧客サポート担当者が集まり、チームとして問題を解決するようにしましょう。メンバーには、チームメイトがその時点での情報とリソースによって最善の意思決定を行ったという前提から始めるよう促します。個人の失敗に注目するのではなく、プロセスの修正とチームとしての解決に集中します。

・ 利害関係者に対する透明性を確保する

自分がしていることを組織全体が理解していない場合、そのやり方は間違っています。すべてのチームが互いの目標、優先順位、および苦勞を理解し、チームのプロセスが他のチームのプロセスに流れ込むようにしましょう。チームが課題と作業項目を完全に把握できるようにし、コミュニケーション チャンネルを開いて、誰もが頻繁に話せる状態を維持します。必要に応じて目標を調整します。



2 自動化とスマートな作業

繰り返しの手作業を排除し、反復可能なプロセスを生み出し、信頼性の高いシステムを構築するために、自動化に投資しましょう。まだ自動化を導入していないチームにとっては、ビルド、テスト、デプロイおよびプロビジョニングの自動化が典型的な出発点です。

- ・ **継続的デリバリを重視する**

デプロイを自動化すると、バグをより早く検出するためのテストを実行することができ、開発者がバグをより簡単に修正できるようになります。また、自動デプロイによって、リリース時の想定外の出来事を軽減および排除できるため、環境間のサーバーの「ドリフト」を IT/運用チームに警告することも可能になります。

- ・ **継続的にプロセスを改善する**

素早く行動し、アグレッシブかつ迅速であるようにしましょう。継続的な改善と失敗は密接に関係しています。失敗を吸収し、リカバリし、学ぶことができるようにチームをセットアップします。失敗がない場合、何かを成し遂げたとは言えません。失敗を利用して、常に改善し続けるのです。インシデントや失敗が起こった後の報告は、チーム間で「何を学んだか」ということを振り返って共有し、最高に価値のあるサービスをビジネスに提供するための最良の方法です。



3 監視と計測

重大な問題のフィードバックにかかる時間を短縮しましょう。助けになるツールおよび技術が多数ありますが、何から何まで計測したくなる衝動は抑えてください。以下のような基本的な計測から開始することをお勧めします。

- ・ スプリント内の完了したストーリーポイント
- ・ アクティブ ユーザーの数
- ・ 定期的なバグ / 障害
- ・ ユーザー定着
- ・ システムの障害後の復旧にかかった時間
- ・ ユーザー離れ

最終的には、機能の使用状況、顧客の行動、NPS、およびサービス レベル アグリーメント (SLA) に関する洗練された指標を得ることができます。チームはこのデータすべてを使用してより良い意思決定を行うことができますが、このデータを他のチームや部門とも共有するようにしてください。たとえそのデータに利点がありません、改善が困難な場合であっても、ロードマップを支持するユーザー データを提供することで、コンセンサスを構築しやすくなるとともに、利害関係者から賛同が得やすくなります。

4 責任と成功の共有

険しい道のりを共に進むことで、成功のお祝いがよりいっそう素晴らしいものになります。リリース日に開発チームが運用チームにベールを持っていくのを見ると、DevOps の文化が会社に定着していることがわかります。

そのためのいくつかのアイデア：



・ 責任を共有する

開発者は、運用チームの最大の負担の1つであるポケベルを持ち運ぶことで即座に信用を得ることができます。何らかの障害が発生したとき、午前2 時に呼び出されることを誰もが恐れています。開発者が本番環境の問題を修正する責任を負うということは、呼び出しを受ける必要があることを意味しています。DevOps は、アプリケーションを構築するのと同じ人がその提供と実行に関わる必要があるという考えを基盤にしています。

・ 開発者にサポートを依頼する

本番環境の問題をトラブルシューティングしながらエンド ユーザーが検出した問題に対処するオンコール エンジニアをローテーションで配置することを検討しましょう。この担当者は、顧客から報告された緊急の問題に対応し、必要に応じてパッチを作成し、顧客から報告された障害のバックログを処理することで、運用チームにとって非常に重宝される存在になり、チーム間に信頼と相互の尊重が構築されます。

・ 賞賛する

同僚からの肯定的なフィードバックは、給料やキャリアの向上心と同じくらいモチベーションを高めてくれます。リリース前に厄介なバグを検出したチームメイトを公に賞賛することは大いに意味のあることです。部署に従業員を労うための自由裁量の予算がある場合は、ぜひそれを使いましょう。

迅速に提供。確実に提供。

Atlassian と DevOps の詳細については、
アトラシアン株式会社までお問い合わせください。