

Effective DevOps

2018年11月28日 15:40

DevOpsとは(みんなに聞いてみた)

- ❖ 開発者が開発から運用の自動化までを行う
- ❖ 組織の壁を低くする作業
- ❖ DevとOpsが同じ仕組みでライフサイクルを回すこと
- ❖ ビジネスや課題に対して、組織論ではなく、チームとして取り組んでいく活動
- ❖ DevとOpsのセクショナリズムより、お互い協力して顧客に価値を届け続ける
- ❖ アジャイルや継続的改善が開発チームだけでなく外に広がったもの
- ❖ 分断されていた組織やチームを1つにして、提供できる価値の最大化を目指すこと
- ❖ プロダクトを迅速・タイムリーにリリースするために、関係者全員が協力し合う組織と仕組み
- ❖ 組織として付加価値を継続的に素早くリリースしていくための取り組み
- ❖ 個人ではなく多様性をもったチームで問題に対応していく方法
- ❖ ビジネスの達成のために開発と運用が互いに敬意・信頼をし合い協力すること
- ❖ みんなでつくる「作り方」

正しそうな問題にアプローチする

- ❖ 全ての組織はそれぞれ違う
- ❖ **自分の組織にとって正しそうなことや求める結果は何か？**
が重要になる
 - ❖ 異なる道筋を通して異なる問題や対立を解決する
- ❖ 「DevOpsをやりたい？」は正しそうな問題設定なのか
 - ❖ 「アジャイルをやりたい」もよく見た
- ❖ **「正しい」は無理なので「正しそう」**
- ❖ **優先順位は必要（プロダクトバックログと同じ）**



フォーカスするとよい4つのテーマ

- ❖ 効果的なDevOpsのための4本柱
 - ❖ コラボレーション
 - ❖ アフィニティ
 - ❖ ツール
 - ❖ スケーリング
 - ❖ 4本柱を組み合わせることで組織の文化的側面と技術的側面の両方に対応可能
 - ❖ **文化、価値観、個人間のコミュニケーションといったものが土台**となる
 - ❖ 一度にすべてをやろうとしない
-

1. コラボレーション

- ❖ チーム内のそれぞれのメンバーが協力しあって仕事を進める
 - ❖ 同じ目標に向かって働き、成功に対する責任を共有する
 - ❖ **これができていないチームが、他のチームとうまく協力しあうことはできない**
 - ❖ **それぞれの人たちが以前よりも効果的に共同作業を進められるようにすることがDevOpsの成功の大部分に寄与する**
-

2. アフィニティ

- ❖ それぞれ別の目標をもった顔の見えない他の組織の人たちと、どうやって最大の成果を出せるというのか？
 - ❖ チーム間や組織間の関係を作る
 - ❖ 組織の共通目標の達成のために、個々のチームの目標の違いを乗り越える
 - ❖ お互いが共感し、他のチームからも学習し続ける

3. ツール

✧ 加速装置

- ✧ 現在の文化と向かう先を踏まえて、変化を推進する
- ✧ ツールが既存の環境にどのような影響を与えているのか理解する必要
- ✧ ツール単独では壊れた文化や機能しないコミュニケーションを解決しない
- ✧ ツールが「重要ではない」という意味ではない

GRAY MATTER

Why Some Teams Are
Smarter Than Others

高いパフォーマンスのチームの特徴



Oleksa Zagoroff

- ✧ 高いパフォーマンスを示す賢いチームには以下のような特徴があった (2015/1 New York Times, アニータ・ウーリー)

- ✧ 平等な参加
- ✧ 心の理論
- ✧ コミュニケーション

うまくいっているチームに共通する要素の仮説

- ❖ 仕事以外のプライベートでも親しい
- ❖ 頻繁に飲食をともにしている
- ❖ 内向的な人同士や外向的な人同士など似た特性の人でチームを構成している
- ❖ 興味や関心が似ている
- ❖ 圧倒的なリーダーシップやカリスマ性がある
- ❖ ボーナスや報酬によるモチベーション
- ❖ 共通の趣味
- ❖ 共通の学歴

チームのルールやふるまいに注目

- ❖ 個人の能力の合計とチームの能力はイコールではなかった
- ❖ 個々に対するアプローチより**集団に対するアプローチ**が効果がある
- ❖ 成果を出したチームにほかのことをやらせても成功するが、失敗するチームは何をやっても失敗する現象
- ❖ よい**集団規範**の有無が及ぼす影響が大きい
 - ❖ 不文律・習慣・行動基準（明文化の有無は関係ない）
- ❖ **均等な発言機会と共感力・他者理解力があると成功しやすかった**
 - ❖ **=心理的安全性**

個人の違いを受け入れる

- ❖ それぞれの文化的背景、キャリアなどによって、お互いの仕事の仕方に影響を与える
 - ❖ 仕事上のキャリア（大企業、スタートアップ、異業種…、技術力、職種の階層、この仕事に就くきっかけ、経験年数）
 - ❖ 個人の特性、性別、性的指向、宗教、人種、階級、母国語、能力、教育レベル
- ❖ つまり多様であることを受け入れる
- ❖ 対立を緩和するために期待値やプロセス、行動規範等を調整する
- ❖ 人を尊重し、人を排除せず、**安心できる環境**を作る

認知スタイルも人ごとに多様

- ❖ 認知スタイルとは個人の情報処理の仕方
 - ❖ 内向・外向・両向
 - ❖ 質問と推測
 - ❖ スターターとフィニッシャー
 - ❖ 分析思考・批判的思考・水平思考
 - ❖ 純粋主義者と現実主義者
 - ❖ 特定の認知スタイルをよいものとしたくないような環境にしていく
 - ❖ スタイルの違いはコミュニケーションによって解決するしかない
-

2つのマインドセット

- ❖ **固定思考**：スキルは生まれつきのもので変わらない。他の人に自分のことを証明しないとイケない。失敗は愚かさ・無能さの証明だと考える。そのため失敗から距離を置こうとする。つまり不確実性を避けたがる
- ❖ **成長思考**：学習と学習環境に身を委ねる。スキルや知識は時間とともに変化する。今は知識がなくても獲得すれば構わない。人に教えて貰って練習すればよい。失敗は個人の本質的な欠陥ではなく、学習プロセスの一部と考える
- ❖ これらは個人のマインドセットであると同時にチームや組織にもあてはまる
- ❖ 変化の速度や競争の激しい今の時代に必要なのはどちらか？
- ❖ **失敗しないことが成功を意味しない**

成長思考を育むためにどうするか

- ❖ チームとして自分の役割として必要なスキルを学ぶ
- ❖ ニッチな領域を見つける
- ❖ 自分の得意なことを知り、それを伸ばす
- ❖ 学んだことを実践に投入する
- ❖ 自分の作業スタイルを確立して改善する
- ❖ チームみんなが足並みを揃えて協力できる仕事の仕方をみつける
- ❖ これらをするには**時間的な余裕が必須（焼畑農業をしない）**

成長思考はどう作られるか

- ❖ 質の高い**フィードバック**を継続的に受ける
 - ❖ その人が「どういう人なのか」ではなく、「その人がどんなことができるのか」に注目する
 - ❖ 持って生まれた資質や変更不能なことにフォーカスしない
 - ❖ 行動に着目し、過去にやったこと、将来すべきことにフォーカスする

フィードバックの方法

- ❖ **定期的な1:1** (毎週～隔週30分など)
 - ❖ 期待値のすり合わせ、キャリア開発、メンタリング、コーチング、業績改善
 - ❖ ネガティブなフィードバックも必要
 - ❖ 個人攻撃ととられないように
 - ❖ 個人攻撃とはとらない
 - ❖ あなた vs わたし ではなく、問題 vs わたしたち
 - ❖ I messageを使う
- ❖ 360度評価

非難文化

- ❖ ミスが発生したときに、個人や組織を非難し処罰する傾向のこと
 - ❖ ポストモートムやレトロスペクティブで犯人探しが行われる
 - ❖ 人を**固定思考**で見ている（ミスした人は賢くない、問題があるという思考）
 - ❖ 透明性を尊重せず分断された環境でよく見られる
 - ❖ 自分が犯人にされないように他人や他のチームに非難が向くようにする
 - ❖ 非難されないように情報を隠す
 - ❖ コラボレーションとは正反対の文化
 - ❖ **心理的安全性の欠如**
-

非難のない文化

- ❖ ミスを、システムのどこかにある問題の兆候と捉える
- ❖ つまりミスは個人のものではなく、構造的な問題と考える
- ❖ ミスを学習の機会だと考える
- ❖ 人を**成長思考**で見る
- ❖ 透明性が尊重され、他人からの**フィードバック**を受けやすくなる
- ❖ コラボレーションの実現の土台となる
- ❖ **継続的改善を支える**

評価システム

- ✧ 文化や思考方法は評価システムの影響を大きく受ける
 - ✧ 過度に結果にフォーカスすると個人は学習より成果を重視してしまう
 - ✧ チームや組織の成果より個人の成果を優先してしまう
 - ✧ 非難文化と減点法の評価では問題は隠す方向に向かう
 - ✧ 自身の成果を達成するのが最重要となりコラボレーションは阻害される
 - ✧ 年に1-2度の評価ではフィードバックサイクルが長い

コラボレーションとコミュニケーション

- ✧ チームの中では考え方の違いによって摩擦や対立が起こる
 - ✧ 何を優先するのか、どれくらいの時間を使うのか、この先どうするのか
 - ✧ 対立そのものは健全
- ✧ 意見の違いを解決したり、交渉して落とし所を探っていく必要がある
 - ✧ **共通の理解**を作る
 - ✧ **共通のビジョンや目的**を作る
- ✧ そのためには効果的なコミュニケーションが必要
- ✧ 効果的なコミュニケーションによって信頼と共感の度合いが高まる
- ✧ **Trust and Verify** (信頼するのと丸投げするのは別。まず信頼し、あとから妥当かを確認)

コミュニケーションツールを使い分ける

コミュニケーション手段	即時性	オーディエンスへの浸透度	受け手の負担	コンテキスト	構成の緻密さ
電子メール	低い	高い	平均的	多い	平均的
臨時の会議 (またはビデオ会議)	高い	低い	平均的	少ない	低い
チャット	平均的	平均的	軽い	多い	低い
正式な会議	とても低い	高い	重い	少ない	高い
Twitterなどの マイクロブログ	低い	平均的	軽い	多い	低い
GitHubのプルリクエスト	低い	平均的	平均的	平均的	平均的
付箋紙のメモ	とても低い	平均的	軽い	多い	低い
PagerDutyページ	高い	高い	重い	平均的	低い
Nagiosアラート	平均的	高い	重い	平均的	低い
書籍またはブログ記事	とても低い	低い	平均的	平均的	高い
画像、グラフ	低い	低い	軽い	多い	とても低い

理解・行動するまで伝える

- ❖ 自分の意見や考えが伝わらないのは発信元の責任
- ❖ 一度言ったくらいでは伝わらない、分からない
- ❖ 相手が理解するだろう、行動するだろう、ではなく、分かるまで伝える

コラボレーションの誤解

- ❖ 古くからのシステム管理者に新しい手法は教えられない
 - ❖ 成長志向、時間や必要なリソースを与える
- ❖ 急成長したいときにはスーパースターを採用しなければならない
 - ❖ 企業を成長させたかったらチームも育てないといけない。最初の人たちが企業文化を決める
- ❖ 多様性に満ちたチームは効果的にコラボレーションできない
 - ❖ 短期的には対立が増えるが、長期的には創造性や問題解決能力は高くなる

コラボレーションの問題解決 (考えてみよう)

1. チームの誰かが持ち分をこなせていない
2. 社員を辞めさせるかどうかを決めなければいけない
3. 私は働きすぎだ、ストレスが溜まっている、燃え尽きた
4. チームのなかに軽く見られていると感じている人がいる
5. コミュニケーションが不十分な人がいる
6. 社員(または候補者)に技術的には優れているけれども不愉快な人間がいる
7. 現在のチーム / 組織にいる限り自分のキャリアを先に進められる気がしない
8. 誰も私の言うことを聞いてくれない
9. 組織再編や人員整理を行ったばかりだ

DevOps

最終的には、社員とチームが一緒になって、自分の組織で機能するものとそうでないものを見分けるという大変な仕事を自分でしなければいけなくなる。長続きする文化の改革を買ってくることはできない。それは、内部から生み出さなければいけないものなのだ。