

サイトリライアビリティ エンジニアリング

SREを成功させるための哲学、習慣、ツール

目次

| | |
|------------------------------|----|
| はじめに | 01 |
| 第1章：SREの哲学と原則 | 04 |
| 第2章：SREを成功させる要因 | 10 |
| 第3章：SREのツールとプロセス | 18 |
| 第4章：New RelicではSREの役割が進化している | 23 |
| 実行 | 29 |

はじめに

高度成長企業が安定性、信頼性、およびオートメーション・ファーストを中心とする慣行を改善する新たな方法を模索する中、開発者と運用エンジニアの日々の責任はますます進化しています。システムの規模が拡大するにつれてダウンタイムを短縮する必要があるため（手動による介入をより少なくするため）、多くの組織がサイトリライアビリティエンジニア（SRE）担当者を採用しています。

「サイトリライアビリティエンジニアリング」という言葉は、グーグルのエンジニアリング担当バイスプレジデントである Benjamin Treynor Sloss氏によって編み出されました。Sloss氏は2003年にグーグルに入社し、グーグルの生産システムの健全性を広範囲で高めるチームを組織する任務を任されました。同氏によると、サイトリライアビリティエンジニアリングとは「ソフトウェアエンジニアが運用を設計したらどうなるのか」です。SREは職能上の枠を超えた業務であり、従来は開発、運用、その他のITグループの内部でのみ活動していた役割を担います。



Sloss氏のチームはサイトリライアビリティエンジニアリングに関する原典とも言える本を書きました。この本のタイトルは[Googleサイトリライアビリティエンジニアリングブック](#)で、DevOpsの世界で素晴らしい最新のSRE慣行がどのようなものか疑問に思っている人にとって素晴らしい参考書です。

この本の中でSloss氏は、「システムが自らの意思で動作することはありません。これは普遍的に認められた真実です。では、システム（特に大規模に動作する複雑なコンピューティングシステム）を動作させるにはどうすべきでしょうか。」

グーグルの答えは、従来の組織ではIT運用チームが通常取り扱っている業務を担うソフトウェアエンジニアを雇うというものです。「当社のサイトリライアビリティエンジニアリングチームは、当社のプロダクトを運用するソフトウェアエンジニアの採用と、ソフトウェアエンジニアがいなければシステム管理者がしばしば手動で実行する作業を遂行するシステムの開発に注力しています」とSloss氏は説明しています。

グーグルから外の世界へ

この本が最初に出版された後、こうした役割のソフトウェアエンジニアは幅広い企業で急速に採用されるようになり、これを受けてテクノロジーニュースと分析サイトTechCrunchが2016年に「**サイトリライアビリティエンジニアは次のデータサイエンティストか?**」と題した記事を掲載しました。翌年、LinkedIn は **SREをハイテク分野における「最も有望な仕事の1つ」と呼びました**。2018年、JeliのソフトウェアエンジニアであるBeth Long氏は、「私の印象では、中小企業への緩やかなトリクルダウン効果が生まれています。グーグル、Netflix、アマゾン、Herokuなどの企業は、リソースとSREを必要とするだけの規模を有しているため、長期にわたりSREを行っていました。SREの役割はより規模の小さな企業でも見られ始めています。そうした企業は『ああ、当社にはこの役割を担う人材が必要だ』と気付いたのです」と述べています。

3年経った今でも、この言葉はそのまま当てはまります。大規模に実行される分散型マイクロサービススタイルのシステムを構築する組織が増えるにつれ、SREへの需要はかつてなく高まっています。

SREの旅を始める

SREの役割は組織によって大きく異なることに注意することが重要です。SREの職務明細と日常業務は多種多様ですが、SREを採用したソフトウェア組織ではその有用性がすぐに明らかになるようとしています。では、次にすべきことは何でしょうか。

自分の会社でサイト信頼性に関する慣行をどのように構築するか考えようとしているにせよ、既存のSREチームのプロセスと慣習を改善しようとしているにせよ、特にグーグルのような大規模な会社で機能することは中小規模の企業では機能しない可能性があるため、主

題への理解を深めれば深めるほど結果は良くなります。そのため、このeBookでは、SREを成功させるための哲学、習慣、ツールを、New Relicの定義、ガイドライン、および役割への期待とともに説明しています。

**SREの需要は依然
高まっています。**

第1章：

SREの哲学と原則

グーグルはSREを「運用志向のソフトウェアエンジニア」と定義していますが、それはどういう意味でしょうか。グーグルではSREチームが人員計画とプロビジョニングの両方を担当しています。チームは、問題に対するソフトウェアエンジニアリングソリューションを探究するという点で、純粋な運用チームとは異なります。ソリューションを探究するため、グーグルはSREが純粋な運用作業に費やす時間を50%に制限しています。つまり、グーグルのSREは自分の時間の少なくとも50%を、自動化やサービスの改善などのエンジニアリングタスクに割り当てる必要があります。

サイトリライアビリティエンジニアリングの目標、リスク、トレードオフ

最初にSREチームの役割について考えるとき、一般的にはシステムの稼働時間をモニタリングすることによって測定される信頼性の向上が主要な目標であると考えられるかもしれませんが、特定のポイントを超えるとそうした考え方は当てはまらなくなります。これは、ネットワークの信頼性など、SREチームの管轄外の要因が関係してくるためです。信頼性と開発チームのベロシティもトレードオフの関係にあります。



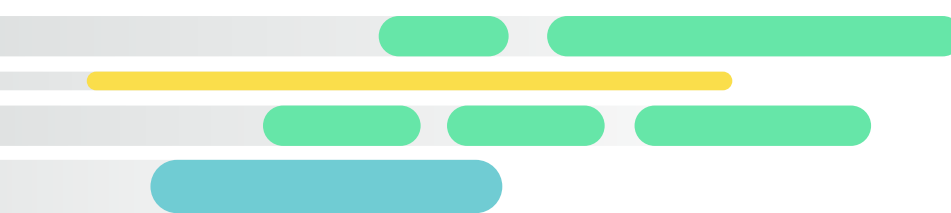
このため、サイトリライアビリティエンジニアリングは一般に、不可用性のリスクと、イノベーションを迅速に進めてサービス運用を効率化するという目標とのバランスを取ろうとします[グーグルのSREに関する本](#)の中でMarc Alvidrez氏は、「我々は十分な信頼性のあるサービスを提供するために努力しますが、必要とされる以上の信頼性は求めません。つまり、99.99%の可用性目標を設定した場合、我々はこの目標を超えたいと考えますが、目標を大きく上回ることはできません。システムに機能を追加したり、技術的負債を整理したり、運用コストを削減したりする機会を失うからです」と書いています。

こうしたトレードオフについて考え、管理するための1つの方法は、所与の製品またはサービスがライフサイクルの中でどの段階にあるかについて考えることです。比較的新しい製品の場合、サービスの稼働時間に厳しい目標を設定すると、イノベーションと実験のペースが望ましくない形で低下するため、逆効果になる可能性があります。逆に、製品が成熟し、それに依存する顧客基盤ができると、ダウンタイム（サービスの停止時間）はより大きな問題となり、サービスプロバイダーの利益に直接影響を与える可能性があります。この点で、稼働時間の目標を引き上げることは理にかなっていません。

ゴールデンシグナル

サービスの可用性を測定することは、特にユーザー向けのサービスにとって良い出発点となりますが、SREチームは通常、自らも追跡している他のいくつかのビジネス指向の主要メトリクスを持つこととなります。しばしば4つの**ゴールデンシグナル**を含むこれらのメトリクスは、特定のシステムが「正常」であるとはどういう意味なのかを定義するものとして非常によく考えられたものとなっています。

様々な種類のアプリケーションには異なるメトリクスが適用されます。例えば、ユーザー向けサービスでは可用性、レイテンシ、スループットが重視される場合がありますが、ビッグデータシステムではスループットとエンドツーエンドのレイテンシが重視される傾向があります。測定すること自体が目的ではないことに注意してください。重要なのは、測定結果がユーザー体験の質とシステムの有効性をどの程度示しているのかです。





SLOとSLIを使った信頼性測定

サービスレベル目標 (SLO) はサービスプロバイダーのパフォーマンスを測定するための一般的な方法であり、サイトリライアビリティエンジニアリングを成功させるうえでも同程度に重要です。プロダクトおよびサービスレベルで明確に定義され測定されたSLOメトリクスは、組織が次のことを実行するのに役立ちます：

- 投資と全体的な優先順位を調整して信頼性に関する目標を達成し、信頼性に関する高いレベルの目標を調整して企業戦略に合致させる
- 顧客の信頼を構築し、維持する
- チームが信頼性を重視すべき時期と信頼性を高める方法を判断できるようにする
- エンジニアがリスク許容度と自らの対応ペースについてより適切な仮説を立て、依存関係についてより適切に判断し、不必要な労力を減らせるようにする

例として、New RelicのシニアSREであるStephen Weberは[New Relicコアデータプラットフォーム](#)には3つの主要なメトリクスがあると語っています。第1のメトリクスは正確性 (正確な結果が得られたか?) で、第2のメトリクスはレイテンシ (許容可能な時間内に応答が得られたか?) です。「そして第3のメトリクスは、レイテンシを許容範囲に収めるために処理を停止して部分的な結果を提供する方法 (グレースフル・デグラデーションとも呼ばれています) です。そのためSREは、レイテンシを最小限にとどめるための3番目のSLIを持っています」。コアデータプラットフォームのSLOはこうした3つのメトリクスで評価されます。

チームが常にSLOを超過達成している (例えばすべてのサービスで99.9%の可用性を達成している) 場合、チームはより迅速に行動し、より多くのリスクを取り、より多くの機能を提供できる

優れたSLI / SLOの特徴は、メトリクスがビジネスの成果やユーザー体験と関連付けられることです。

可能性があります。チームが危険にさらされているか、SLOを満たしていない場合、SLOの超過達成は信頼性に焦点を合わせるために一歩後退して一時停止するよう促す信号であり、チームはそうすることで再び迅速に行動し始められるようになります。

SREチームは、必ずしもSLOの一部ではない信頼性を測定するために使用する他のサービスレベル指標 (SLI) を持っている場合もあります。これらのパフォーマンスメトリクスは、事業における複数の側面を追跡します。例えばデータベースサービスのためのSLIは、「エラーなしで200ミリ秒以内に正常に完了したユーザークエリの割合」のような数値を追跡する可能性があります。

チームは信頼性を測定するため、平均故障間隔 (MTBF)、平均検出時間 (MTTD)、平均修復時間 (MTTR) などのメトリクスに目を向けます。これらのメトリクスはいずれも、組織が「リスクマトリックス」を定義する

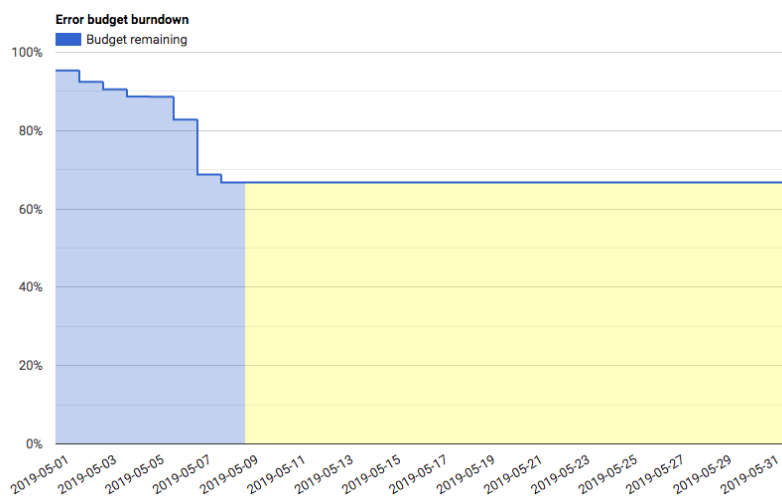
のに役立ちます。これらは、SLOに定量化可能な影響を与える問題とリスクに優先順位を付けるための強力なツールとなり、組織が特に緊急ではない可能性のある問題に取り組めるようになります。

優れたSLI/SLOの特徴は、メトリクスがビジネスの成果（多くの場合はユーザー体験）と関連付けられることです。例えば、エラー率が高い、または応答時間が遅いと、ユーザー体験に悪影響が生じます。CPU使用率が高いと、ユーザー体験に悪影響が及ぶ可能性があります。CPU使用率の高さとユーザー体験の低下との関係を把握するのは困難です。

エラーバジェット

最後に、必須ではないかもしれませんが、サービスのSLOは四半期ごとのエラーバジェットの決定に役立つ場合があります。エラーバジェットは、サービスの信頼性低下が1四半期内にどの程度許容されるかを決定する明確で客観的なメトリクスです。

チームは、エラーバジェットの減少ペースを示したバーンダウンチャートを作成して、それに応じて作業内容を調整することができます。興味深いことに、グーグルでは、サービスの稼働率が100%である場合、サービスを停止させます。そうすることで、依存するサービスがどのように反応するのか把握せざるを得なくなるからです。



もちろん、エラーバジェットが厳しすぎると、開発のペースが遅くなる可能性があります。エラーバジェットを設定することで、そうした問題を把握することが可能となり、おそらく開発チームがベロシティを高められるようにエラーバジェットを緩和するといった判断が下せるようになります。そうした状況では、プロダクトおよびSLAエンジニアはより迅速な開発を可能にするため、許容可能なエラーの数を増やすと判断する可能性があります。一部の組織では、アプリを「高信頼性」と「高ベロシティ」に分類し、それに応じてより厳しい/より緩いエラーバジェットを設定しています。

エラーバジェットは、その設定がどうであれ、インセンティブを調整し、ソフトウェアエンジニアリングと製品開発の間で共有される当事者意識を強めるために重要です。

第2章:

SREを成功させる 要因

SREを採用する際、候補者の技術面での貢献度合いは、特定の組織がその役割をどのように規定するか、あるいは役割にどのようにアプローチするのかによって異なります。すなわち、ある会社はより多くのソフトウェアエンジニアリングとコーディングの経験が必要とし、別の組織は運用または品質保証 (QA) スキルをより重視するかもしれません。バランスがどうであれ、「素晴らしい」と「十分に良い」を分ける要因は、多くの場合、エンジニアの習慣と特質の組み合わせが技術的な専門知識をどれだけ補完しているかです。

素晴らしいSREであることを確認する方法は次のとおりです。

SREが (はるかに) 広い視野に立っている

成功するソフトウェア開発者は、自分が作成したコードがビジネス全体の推進にどのように役立つかを理解しており、優れたSREはこれに関する独自の特性を持っています。New RelicのシニアソフトウェアエンジニアであるJason Qualmanは次のように述べています。「もし日常業務から離れた広い視野で、より高いレベルで物事を理解し、解釈できる人を探しているなら、成功するSREがまさにその人です。」変化は、その瞬間だけでなく将来にわたるリスクや影響を生み出す可能性があります。優れたSREは、変更を加える前に必ず徹底的な分析を行います。

自分の仕事が特定のシステム、チーム、またはより大きなインフラストラクチャの残りの部分にどのように影響を及ぼすかを検討する能力は、SREに求められる極めて現実主義的な側面です。壁の向こう側で座っている人にどのような影響が及ぶのかを気にすることなく変化を与えるサイロ化されたアプローチでは、長期的なメリットはほとんど生まれません。

QualmanはSREについて、「非常に低い階層で意思決定を行っています」と述べています。そうした意思決定は、はるか上位の階層に属する人々に影響を与えます。適切な意思決定を行うことにより、シームレスな移行が可能になります。

広い視野 に立つ

SREは好奇心旺盛で共感的

New RelicのシニアSREであるKat DoberとStephen Weberは、SREに求める重要な特性として好奇心を挙げています。

Doberは次のように述べています。「多くの企業はエンジニアリングに対しての好奇心、エンジニアリングがどのように機能するのか、エンジニアリングがどのような道筋をたどると失敗する可能性があるのか、こうしたことを考えることができるマインドセットを持つ人を探しているでしょう。」

Weberも次のように述べています。「多くの場合、最も有益な改善は『面白い』というところから始まります。そして、面白いことを掘り下げ続けるのです。」

Weberによると、好奇心に関連する特徴は顧客への共感です。「ページの平均読み込み時間はかなり良好かもしれませんが、一部の顧客の読み込み時間が非常に長い場合は、その顧客が体験している不都合を確認する必要があります」とWeberは述べています。

好奇心旺盛
で共感的

SREはあらゆる機会です自動化に挑戦する

SREの役割には常に手作業による調査が含まれることとなりますが、SREは作業における「労力」部分の削減を目指しています。Googleでは労力が特定の意味を持っています。Vivek Rau氏はSREに関する本の中で、「労力とは、手作業、反復的、自動化可能、戦術的、永続的な価値の欠如につながる傾向があるプロダクションサービスの維持に結び付けられ、サービスが成長するにつれて直線的に増加する種類の作業である」と指摘しています。

Google・リサーチ経由で公表されたRau氏の章の[フォローアップ](#)記事で、Rau氏らは、GoogleのBigtable (ビッグテーブル) サービスをサポートするSREチームの労力を削減するためのケーススタディを紹介しています。「Bigtableを担当したSREは作業量を大幅に削減できた。労力を段階的に削減するたびに、将来の労力削減に取り組むためのエンジニアリング時間が増えた…。2014年までにチームの運用は大幅に改善

された。チームはユーザーからの要求を、2013年初めに達した四半期あたり2,200件超のピークから同400件未満に減少させた。」

これを達成するうえでカギを握ったのが、一般的な種類の様々なサポート要求の自動化を徐々に進めたことです。より一般的な教訓として、SREは通常、面倒な手作業と労力を軽減するための重要な手法として自動化に注目することが分かりました。

Doberは、「自動化は、問題のある領域を理解した後、またはインフラストラクチャを理解した時点で、実際に機能し始めます。継続的に対処しなければならないことがわかっているタスクもあります」と述べています。「例えば、すべてのホストシステムをどのように構成するのか、あるいはそのホストが入っているレポジトリからコードを取得してアーティファクトまたはコンテナにパッケージ化し、さらにインフラストラクチャ全体にデプロイする方法を考えてください。これらのタスクを自動化することで、労力が軽減されるだけでなく、タスクが毎回安定的に正しく実行されるようになります。」

自動化に 挑戦する

Qualmanもこの考えに同意し、「SREの役割を持つ人の多くは、人々が行っている非効率的で時間のかかる作業のことを考えており、そうした作業をできるだけ早く終わらせようとしています。手作業の自動化を先送りするのではなく、『今すぐこれを自動化するための時間を取り、この苦痛を伴う義務的作業を他の誰かが行うのをやめさせる』ということです」と述べています。

こうした自動化へのこだわりは、SREとDevOpsの哲学における重要な原則です。実際、「DevOpsハンドブック」には、手作業を受け入れたプロセスがもたらす直観に反する影響について説明した章があります。「自動化」やそれに該当する言葉はSREの職務説明書の他のどの単語よりも頻繁に現れるようです。SRE求人リストにおける重要な役割として「自動化、自動化、自動化、そして…自動化!」と書かれていても不思議ではありません」。

**自動化への執着
は、SRE (および
DevOps) の
重要な信条
です。**

SREはチェンジメーカー

自信をもってSREイニシアチブを提唱する能力は、最高のSREだけが持つもう一つのスキルです。SREの仕事の一部は、簡単に言えば、最初はやりたくないと思うかもしれないことを実行するよう他の人に説得することです。例えば、プロダクト機能を短期間で提供することに注力しているソフトウェアエンジニアに対し、その機能を今後数年間で拡張するにはどうすればよいかを考えるよう説得します。

こうした説得は、SREがプロダクトチームに直接組み込まれていればより簡単に行えます。セールスフォースのHerokuのサイトリライアビリティエンジニアであるJohnny Boursiquot氏は、QCon Plusでの「外交官としてのSRE」というプレゼンテーションの中でSREの採用について言及し、変化を促進する方法として既存のプロダクトチームによるSREを組み込むことを慣行とするよう推奨しました。同氏は「2つの組織がサイトリライアビリティエンジニアリングを同じ方法で実行することはありません」と述べています。同氏は「事実、最初からSREの機能を拡大する時期に気がつくことはめったにありません。」と考えています。

同氏はこのテーマについてさらに次のように述べています。

「SRE採用に関する一連のベスト・プラクティスがありますが、組織の中でSREを擁護する立場の人間は、そうした規範的アプローチだけではSRE採用をスムーズで直接的影響のあるものにするために必要なすべての要件を満たせないことを知っています。

「サービスの開発から継続的な運用ニーズに至るまでの過程で全員が当事者意識を持つ組織ほど、こうした採用の課題が組織の隅々まで共有されているところはありません。こうした組織では、チーム固有の慣行を開発するのが一般的であり、必要でさえあります。こうした「全員が当事者意識を

チェンジメーカー

持つモデル」は、システムのライフサイクルの初期段階でビジネス目標を前進させるのに適していますが、成熟したチームが共通の信頼性慣行と道具立てを採用する必要がある場合、最終的にはいつの間にか手付かずの技術的負債と化してしまいます。

「こうしたリーダーシップの意図とチーム内の実践的意義との間のギャップを埋めるには、チェンジエージェントがSREという形でチームに組み込まれる必要があります。自分たちが充分満たされていると感じているチームは、業務の進め方を変えるよう求める伝統的な外部のSRE機能と連携するよう常に動機付けられているわけではありません。そうした変化が状況を著しく改善するとしても、です。理由は何であれ、こうしたチームの間

で友好関係を築くには、まずは信頼を確立する必要があります。もちろん、こうした信頼構築を容易にする1つの方法は、SREをそれらのチームの内部に直接埋め込むことです。

**優れたSREは、
有能な営業担当者
でもある。**

言い換えれば、優秀なSREは効果的な営業担当者でなければなりません。そうしたSREは、短期的な痛みを伴ったり、従来の規範に反しているように見えるプロセスやプロジェクトを同僚に受け入れさせたりする能力を持たなければなりません。Beth Long氏は、「覚悟

を決めて、『やめろ』と『いいえ』を言える必要があります。それは、一部のエンジニアリング組織では難しいかもしれません」と述べています。

採用マネージャーは、SREの役割を特定の経歴に限定しないことで最も成果を上げることができます。

SREは（必要であれば）新しいツールとアプローチを採用する

サイトリライアビリティエンジニアリングはまだかなり新しいアプローチであるため、現時点でこの肩書を持つ多くのエンジニアはこの役割を引き受ける前に別の業務に就いていました。一部の SREは開発者の経歴を持っていたり、従来の運用やシステム管理者の経歴を持っていたりするため、採用マネージャーはSREの役割を特定の経歴に限定しないことで最も成果を上げることができます。例えば、伝統的な品質保証 (QA) エンジニアがSREのポジションに適したスキルセットを持つ場合があります。

経歴に関係なく、SREの役割は、合理的に行動し、積極的に適応することを要求します。SREの役割は、快適なゾーンから踏み出し、新しいスキルを開発する意欲を持つよう鼓舞します。Weberは、「私は、開発者時代と比べて、多くの異なるシステム、異なるプログラミング言語、YAMLの異なるスタイルとやり取りしています。開発者だったころは自分がそうするようになるとは思ってもみませんでした」と語っています。「1日に5つの異なるプログラミング言語を使うことは必ずしも珍しくないので、

新しいツールと アプローチ

第3章:

SREのツール とプロセス

SREに求められるのは進んで柔軟になることと、思い切って飛び込むことです。SREにとっての合理的行動の1つは、意図は正しかったがもはや生産的ではないプロセス、手順、およびツールを破棄することです。

SREに関する普遍的な職務明細がないのと同様、この役割で使用する標準的なツールセットも存在しません。しかし、優れたSREは常に信頼性ツールとプロセスを最適化し、それらの効能を組織全体に伝えようとします。

SREのプラクティスを成功させ、DevOps の原則を適切に実装するうえでは、最適化がカギを握っています。ところで、SREはどのようなツールを標準化すべきでしょうか。各チームは、自分たちにとって何が最善かを判断する必要があります。良いニュースは、選択肢がたくさんあることです。

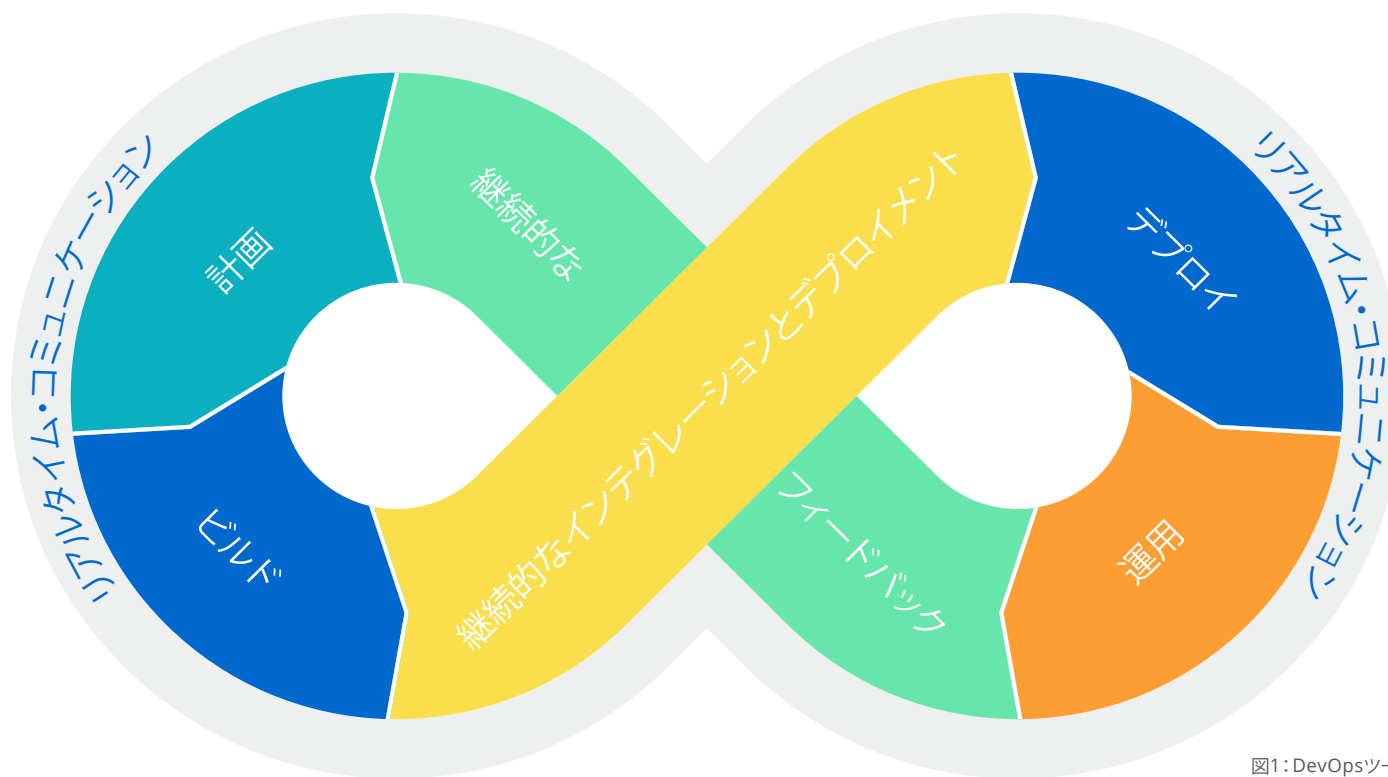


図1: DevOpsツールチェーン

DevOps (およびSRE) の各段階のためのツールチェーン

あなたが「SREの各段階」のためにツールチェーンを構築したとします。出来上がったツールチェーンがDevOpsツールチェーン (図1) と似ていたとしても驚かないでしょう。

パブリッククラウドの使用頻度が増加し、それに対応するコードツールとしてのインフラストラクチャの使用頻度も増加しています。これは、パブリッククラウドが急激に変化し、解約が急増している領域であることを意味します。しかし私たちは、現在広く使用されているツールとプラクティスの一部を概説することができます。

計画：

計画は、アジャイルプロジェクト管理ツールと追跡ツールの両方で構成されています。こうしたツールには、[Avaza](#)、[Jira](#)、[YouTrack](#)、[Trello](#)、[Pivotal Tracker](#)、またはその他のタスク管理ツールがあります。

ビルド：

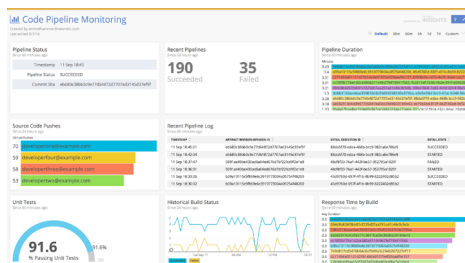
ここでは、コードツールとしてのインフラストラクチャを見つけることができます。こうしたインフラストラクチャには、再プロビジョニング環境を高速化し、より一貫性を持たせ、信頼性を高める[Ansible](#)、[Chef](#)、[Docker](#)、[Puppet](#)、[Terraform](#)などがあります。[Kubernetes](#)と[Docker](#)などのコンテナとオーケストレーターも役割を果たし、開発者とSREはプロダクションシステムの使い捨て可能な仮想レプリカを対象に作業することができます。

[Bitbucket](#)、[GitHub](#)、[GitLab](#)などのソース管理ツールおよびコラボレーティブコーディングツールのほかに、[IntelliJ IDEA](#)や[Visual Studio Code](#)などの統合開発環境 (IDE) も広く使用されています。

継続的な統合と情報配信：

開発者が共有リポジトリにあるコードを1日に数回確認し、一連の自動テストを実行し、一連のテストが成功したら更新されたコードを自動的に運用環境にリリースすることがますます一般的になっています。このアプローチは、[AWS Code Pipelines](#)、[Bitbucket pipelines](#)、[CircleCI](#)、[Jenkins](#)などのCI / CDツールを、[JUnit](#)、[Mabl](#)、[Sauce Labs](#)、[Selenium](#)などのテストツールと組み合わせたものです。継続的な情報配信に関する重要なポイントは、チームがデプロイ準備の整ったソフトウェアを持っていても、それをすぐにデプロイできるとは限らないということです（下記の「デプロイメント」を参照してください）。

New Relicの多くのお客様は、プロセスのこの段階を追跡するのに役立つパイプラインダッシュボードも構築しています。



運用:

これには通常、New Relicなどのモニタリングツールと、Jira Service DeskやStatuspage、PagerDuty、またはZendeskなどのインシデント、変更、および問題追跡ツールが含まれます。New Relicでは、SREとエンジニアがログ管理機能とカスタムインストールメンテーションを使用しています。

New RelicのApplied IntelligenceなどのAIOps ツールは、サービスの異常を先見的にモニタリングし、リアルタイムの障害警告と行動に結び付けられる詳細情報を通知するため、より迅速な調査を可能にします。インシデントは、PagerDutyなどのツールに直接配信することができます。

デプロイ:

継続的インテグレーションと情報配信を行っていても、まだ継続的なデプロイを実施していない場合、デプロイメントは別のステップとなります。上記の継続的インテグレーションの手順と同じツールを使用しますが、主な違いは、コードが準備できたらすぐにデプロイすることが初期設定となっているかどうかです。ビジネス上の理由で継続的なデプロイメントを行わないことがあります。SREのベストプラクティスは、自動的に発行される頻繁で小さな増分更新を行うことです。

継続的なフィードバック:

定期的な顧客フィードバックを収集するための文化とプロセスの両方をカバーするには、GetFeedback、Slack、Pendoなどのツールを利用します。ループのフィードバック部分には、パフォーマンスとプロセスに関するメトリクスも含まれるため、例えばDRI (Don't Repeat Incidents) に対応するJiraチケットが有効です。リリースダッシュボードも継続的なフィードバックの例です。

変えられないことは何もない

ある時点でSREが使用するツールは、組織がSREの旅のどこにいるかによって異なります。また、パブリッククラウドへの移行を背景に、役割も大きく変化しています。AIOpsツールを介して自動化する機能を含む新しいトレンドは、役割を再定義し続けるでしょう。

成熟度の低い組織はより専門的な運用ツールを使用する傾向がありますが、成熟度の高い組織ではSREとソフトウェアエンジニアリングツールチェーンを収斂する動きが見られます。つまり、万能のツールセットがないことは確かですが、SREは、すべての作業に高い信頼性をもたらすための新しくてより効率的な方法を模索する中で、適切なツールを試して採用する必要があります。

**AIOpsツールを
通じて自動化す
る能力が役割
を再定義し続け
るでしょう。**

第4章：

NEW RELICで はSREの役割 が進化している

グーグルの[サイトリライアビリティエンジニアリング](#)の本は、DevOpsの世界で優れた最新のSREのプラクティスがどのようなものを概説するという素晴らしい仕事をしています。しかし、グーグルほどの規模を持たない企業でSREのプラクティスを実践する場合はどうでしょうか。信頼性に関するプラクティスについて書かれているすべての内容に関して言うと、グーグル以外のエンジニアリング組織でSREが果たす日々の役割について具体的かつ詳細な説明をこの本の中で見いだすのは非常に困難です。インターネット上に関する説明の大半には、「SREはソフトウェアエンジニアリングと運用スキルセットの両方を備えている」や「SREはあらゆることを自動化する」などの漠然とした言い回しが含まれています。

役割の定義

New RelicのSREの職務明細の作成は時間を要する作業となり、完成した職務明細には個々のSREと経営幹部からの意見が反映されました。

New RelicのSREは、システムの信頼性の向上に注力し、主にその点で能力が認められているエンジニアです。ビジネスの観点からは、SREが行う作業の目標はお客様の信頼を構築して維持し、New Relicのプラットフォームのサービスごとに、さらにはホストごとに運用オーバーヘッドを着実に削減することで、ビジネスを拡大できるようにすることです。

大まかに言えば、SREはこれを次の方法で実現します：

- 信頼性に関するベストプラクティスの提示
- 回復力と労力削減を念頭に置いた設計とプロセスの指導
- 技術的な複雑さと無秩序な増加の削減
- ツールと一般的なコンポーネントの使用の促進
- 回復力を向上させ、運用を自動化するためのソフトウェアとツールの導入

役割の進化

New Relicが最初にSRE機能を作成したとき、そのSRE機能はグーグルの説明どおり、一元化されたチームに基づいていました。しかし現在では、New Relicの様々なプロダクトチームにSREが永続的に組み込まれています。この後者のアプローチは前述のBoursiquotのアプローチと似通っています。

組み込まれたSREが集中的に配置されているテレメトリーデータプラットフォームグループのエンジニアリング担当シニアディレクターであるGus Shafferは、信頼性を一元化する機能を持つことは、コーディングとリリースを1つのチームに任せるというDevOpsの目標に反すると指摘しています。これについてのShafferの説明は、「私たちは、信頼性に対する責任の放棄があったことを見いだしました。人々が『ま、いいさ。信頼性組織が存在し、その組織が信頼性に関する責任を負っているのだから』などと言っているのです。「それどころか、信頼性組織は実際には、測定と報告、人々が信頼性動向を理解するためのサポート業務、人々が正しい行動を取るのを支援するプロセスとポリシーのとりまとめの責任を負っていました」というものです。

「私たちは、人々が使用しているツールに信頼性に関するプラクティスを組み込んでいます。」

Stephen Weber, Senior SRE, New Relic

Weberはこの見解に対し、「SREをセントラルチームからプラットフォームチームに移管することでもたらされる最大の利点は、人々が使用しているツールに信頼性を組み込むというアイデアを取り入れることだと考えています」と反応しています。

新しい組織構造を採用することで、New RelicのSREがプロダクトアーキテクチャー全体を最新の状態に維持することが以前よりも容易になります。組織構造を変更することにより、監査作業の量が減り、SREがしばしば行う必要のある「bad-cop」（悪い警官）の役割を実行できるようになります。さらに、SREが開発により多くの時間を費やすことが容易になりました。これは、グーグルの50%の上限が目指すのと同じ目標を達成するためのもう一つの方法です。言い換えれば、組織構造を変更することでいくつかの問題が解消され、**逆コンウェイ戦略**が効果的に実行されるようになります。

ただし、この変更はいくつかの独自の課題を伴います。1つは、一元化されたSRE機能がないため、横断的な懸念事項への対処が難しくなることです。New Relicの場合、その例に該当するのが、すべてのNew Relicのデータパイプラインに使用されている**Apache Kafka**です。広い範囲で使用するという事は、プラットフォームの様々なクライアントがこの機能を可能な限り効率的に使用することが非常に重要であることを意味します。これを確実なものとするため、New Relicは割り当て制の導入を検討しており、生産エンジニアリングチームとエンジニアリングスタッフの交代勤務表を立ち上げました。「データプラットフォーム内のすべての異なるシステムに関する専門知識を備えるため、これらの異なるすべてのチームから人々を招き入れました。これは、影響を受けるチームに所属している人々はこの『一元化された』SREチームの一員であるため、組織変更は即座に賛同されることを意味します」とShafferは説明しています。



New RelicにおけるSREの役割も、他の要因に応じて進化しました。New Relicは、独自のデータセンターではなくパブリッククラウド・インフラストラクチャを使用する方向にますます変化しています。こうした変化に対応して、New Relicのチームはソフトウェア・ディファインド・インフラストラクチャで業務の行うという変化がもたらされました。

パブリッククラウドへの移行は、クラウドリソースを効率的に使用することがSREの役割の重要な部分になっていることも意味します。Shafferは、「SREは必ずしもAWSの請求書を監視しているわけではありませんが、彼らはリーダーシップが発信するシグナルに応じています。そうしたシグナルには例えば、『このシステムはかなり高価に思えます。そうあるべきと思われる水準よりも効果ですが、それについて調べてもらえますか?』といったものがあります。また、SREは過剰なプロビジョニングは行わないキャパシティ管理機能の一部でもあります」と述べています。

New RelicにおけるSREの業務内容

下の表はNew RelicのSREの現在の役割の概要を示しています。

| 仕事の種類 | 例 | メモ |
|---|--|--|
| New Relicの運用と信頼性のベストプラクティス (キャパシティプランニング、SLO、インシデント対応など) を学び、強化し、チームと協力してそれらのプラクティスを導入する。 | <ul style="list-style-type: none"> • チームのリスクマトリクスを更新する。 • 顧客の需要に備えてキャパシティを管理する。 • コストとクラウドリソースを効果的に使用する方法を検討する。 • チームを動かし、最も重要な信頼性に関する作業に優先順位を付ける。 | <ul style="list-style-type: none"> • これは、新しいSREおよび新しいチームと連携するSREが特に注力している業務です。 • すべてのSREは、プラットフォームツールとSREコミュニティのベストプラクティスを常に把握しています。 |
| コア共有内部コンポーネントを構築する、またはチームによるコンポーネントの採用を支援する。 | <ul style="list-style-type: none"> • チームと協力して、システムを共有デプロイメントパイプラインの新バージョンに移行させる。 • コンテナ・ランタイム・プラットフォームにコードまたはツールを提供する。 • 新しいツールを構築するのではなく、適切な既存のツールを選択するようにチームを導くことにより、技術が無秩序な状態にならないようにする。 | <ul style="list-style-type: none"> • SREは、新しいツールやシステムを導入するのではなく、既存のツールを使用することが期待されています。 |
| New Relicプラットフォームのモニタリングと可観測性を向上させる。 | <ul style="list-style-type: none"> • チームと協力して、ノイズとなる未使用のアラートを整理し、重要な問題がアラートにより警告されるようにする。 • 統合を進めてプラットフォームに新たな可視性をもたらす。 | <ul style="list-style-type: none"> • SREは、可能かつ効果的な場合はいつでも既存のNew Relicプロダクトを積極的に使用および拡張し、そうでない場合はプロダクト管理を動かして必要な機能を実装します。 |



成功に向けて自社のSREを配置する

このSREの役割の説明とアプローチは、New Relicではうまく機能していますが、他の組織には適していないかもしれません。それでも、この説明とアプローチは有用な例となり、SREの優れたプラクティスがもたらすことのできる非常に大きな価値を明確化するのに役立ちます。独自のガイドラインを作成することで、成功に向けたSREの配置が可能となります。また、SREのプラクティスは、成熟するにつれて、かつてなく複雑化しているコンピューティングプラットフォームをサポートするうえで不可欠な役割を果たすようになりますが、独自のガイドラインにより、こうした役割への集成的理解を深めることができます。

最後に、信頼性に深い関心を持ちベストプラクティスを共有するSREとその他の人々のために、実践的なコミュニティと関係を構築することが重要です。これにより信頼性の文化が生み出されるからです。

実行

SREの役割を定義し、適切な組織構造とインセンティブを設定すると、すべてが実行に移されます。SREチームを成功させるには、様々なスキルと特性が必要です。技術的なスキルはいつでも教えることができますが、共感や好奇心など、同じように不可欠な資質を与えることは必ずしもできません。

New Relicのようないくつかのエンジニアリングカルチャーは自律性を高く評価していますが、それはチームが独立して信頼性に取り組む必要があるという意味ではありません。チーム（および個々のSRE）が成功を収めるには、組織的なサポート、コミュニケーション、そして何よりも、信頼を必要としています。

SREを成功させるための指針となる哲学は、次のように表現できるかもしれません。「達成不可能なゴールを追い求めてはいけません。物が壊れるのを防ぐことはできません。代わりに、全体像を確認し、自動化機能を組み込み、健全なパターンを奨励し、新しいスキルとツールを学び、すべての作業の信頼性を向上させるために、努力を続けてください。完璧を達成することは不可能かもしれませんが、物事をより良くするために絶えず努力することが、可能な限り近づく方法です。」

成功するDevOpsはここから始まります。重要なことを測定し、イノベーションのスピードを加速させます。無料アカウントにサインアップ。