Sreake

何から始める？
組織へのSRE導入に向けて

アジェンダ

01 About 3-shake

02 SREとは？

03 SRE導入の難しさと進め方について

04 SRE総合支援サービスのご紹介

01

About 3-shake

会社概要

社名	株式会社スリーシェイク
設立日	2015年1月15日
代表取締役	吉田 拓真
所在地	本社：東京都新宿区大京町22-1 グランファースト新宿御苑3-4F
人員（2022/9）	130名（正社員：84名、業務委託：36名、アルバイト：10名）
資本金	1億円
事業内容	SREコンサルティング支援事業「Sreake」の運営 セキュリティ診断サービス「Securify」の運営 データ連携プラットフォーム事業「Reckoner」の運営 フリーランスエンジニア紹介プラットフォーム「Relance」の運営

Corporate MISSION / VISION

MISSION

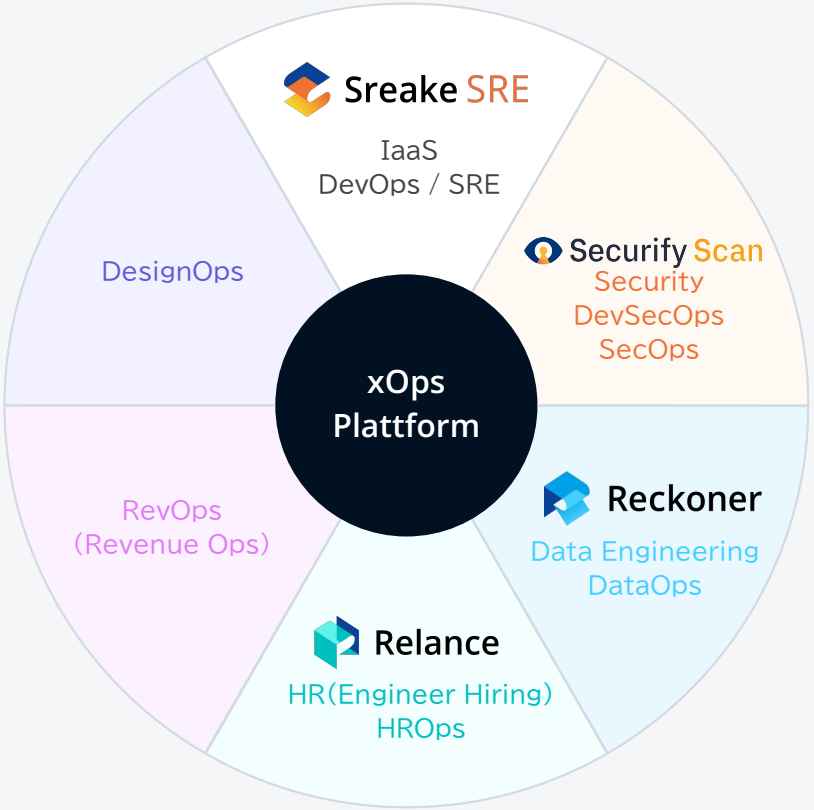
インフラをシンプルにして
イノベーションが起こりやすい世界を作ること

VISION

社会に蔓延る労苦(Toil)をなくす
プラットフォーマーになること

サービス一覧

スリーシェイク = xOps領域のプラットフォーマーへ



SRE

Sreake SRE

スリーク

日本のSREをリードする

SRE総合支援からセキュリティ
対策を全方位支援

Security

Securify Scan

セキュリファイ

事業者が抱える
セキュリティリスクを無くす

本格的な脆弱性診断を
無料で手軽に

Data Engineering

Reckoner

レコナー

あらゆるサービスを連携する
ハブになる

クラウド型ETL/データパイプ
ラインサービスの決定版

HR(Engineer Hiring)

Relance

リランス

良いエンジニアに良い案件を

フリーランスエンジニアに
「今よりいい条件」を

02

SREとは？

SRE とは ～SRE 本の序章より ～

3-SHAKE

‘‘What exactly is Site Reliability Engineering, as it has come to be defined at Google? My explanation is simple: SRE is what happens when you ask a software engineer to design an operations team. ’’

『Googleで定義されるようになったSite Reliability Engineeringとは一体何なのか？
私の説明は簡単です。SREとはソフトウェアエンジニアに運用チームの設計を依頼したときにできあがるものです。』

By ベンジャミン・トレイナー・スロス
(Vice President, Engineering of Google)
※GoogleでSREチームを立上げ

・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング　—Googleの信頼性を支えるエンジニアリングチーム （2017）オライリージャパン）
1.2 サービス管理への Googleのアプローチ：サイトリライアビリティエンジニアリング（ P5）
・Google - Site Reliability Engineering https://sre.google/sre-book/introduction/（参照 2022/11/14）

SRE とは

3-SHAKE

SRE(Site Reliablity Engineering)とは、

効率的なサービス運用をベースに、
サービスの信頼性と**イノベーションスピードとのバランス**を取りながら、
ユーザに提供する価値を最大化すること。
または、それらを行うチーム。

サービスの信頼性とイノベーション スピードとのバランス

効率的なサービス運用

SREの原則

1. リスクの受容
2. SLOの定義
3. トイルの撲滅
4. 自動化の推進
5. 適切なリリースエンジニアリング
6. 分散システムのモニタリング

・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング —Googleの信頼性を支えるエンジニアリングチーム （2017）オライリージャパン）第II部 原則
・Google - Site Reliability Engineering https://sre.google/sre-book/part-II-principles/（参照 2022/11/14）

02-01

SREにおける信頼性と
イノベーションスピードとのバランスについて

Googleの提唱するSREにおける信頼性

01 いかなるシステムにおいても最も重要な機能は信頼性である。

- 新しい機能が追加されるけど頻繁に使えなくなるサービス vs 機能古いけど安定したサービス。

02 監視が信頼性を決めるのではない。ユーザーが決めるのだ。

- 信頼性 = CPU使用率？
- 信頼性はユーザ利用への影響、ユーザ経験に基づき、定義されるべき。

03 可用性100%を信頼性の目標とすることは間違いである。

- 過度の信頼性はコストに跳ね返る
- ユーザは高い信頼性と極端に高い信頼性に気づかない

・Google Cloud で実践する SRE https://www.slideshare.net/GoogleCloudPlatformJP/google-cloud-sre-249557085（参照 2022/11/14）

サービスの信頼性とイノベーションスピードとのバランスについて

システム運用において**信頼性と機能開発はトレード・オフの関係**にある

運用「信頼性を担保するためにあまり変更したくない！」

※システム障害の7割は変更により発生

開発「新しい機能を提供したいので機能開発をたくさんしたい！」

運用と開発それぞれ目標が異なり、協調が難しく
サービスの価値を効率的に上げることが難しいこともしばしば

・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング　—Googleの信頼性を支えるエンジニアリングチーム　（2017）オライリージャパン）
1.1 サービス管理へのシステム管理者のアプローチ
・Google - Site Reliability Engineering https://sre.google/sre-book/introduction/（参照 2022/11/14）

Principle of SRE

1. リスクの受容
2. SLOの定義
3. トイルの撲滅
4. 自動化の推進
5. 分散システムのモニタリング
6. 適切なリリースエンジニアリング

信頼性を定量化し、定量化された信頼性に基づき、開発⇄運用間が同じ基準（共通言語）で、リスクを管理することで信頼性とイノベーションスピードのバランスをとる。

SLI（Service Level Indicator）：サービスレベル指標

・システムの良し悪しを判断するための指標。
例）ユーザのリクエストに対して正常に応答を返した割合をSLIとする。

SLO（Service Level Objectives）：サービスレベル目標

・サービスレベルに対する内的な目標値。SLIを元に定義。
例）過去30日間の正常な応答割合（SLI）が99.9%以上であることをSLOとする。

Error Budget：エラー予算

・許容できる不具合の割合（1－SLO）
エラー予算ポリシーを定め、ポリシーに従って対応する。
例）1からSLOを引いた0.1%をエラー予算（失敗を許容する割合）とする。
→失敗した割合が0.1%を超えるまでは、新規開発を優先
→失敗した割合が0.1%を超えた場合は、新規開発を停止し、SLO達成に向けた改修を行う等

・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング　—Googleの信頼性を支えるエンジニアリングチーム　（2017）オライリージャパン）
・Google - Site Reliability Engineering https://sre.google/sre-book/part-II-principles/（参照 2022/11/14）

02-02

効率的なサービス運用について

サービス運用上の問題

サービス運用の現場では様々な問題が存在する。

煩雑で繰り返しの多い
手動の運用作業

日々追われる障害対応と
メンバーへの過負荷・属人化

リリースの都度、
反復性のない個別対応

効率的なサービス運用とは

効率的なサービス運用とは、
ソフトウェアエンジニアリングでシステム運用の問題を解決すること
それが行えている状態

煩雑で繰り返しの多い
手作業(トイル)は
自動化によって撲滅

メンバへの過負荷・属人化は
適切なモニタリングにより
データに基づくハンドリング
過負荷を削減

リリース作業の負担は
信頼性の担保された
リリーススキームで軽減

3-SHAKE

Principle of SRE

1. リスクの受容
2. SLOの定義
3. **トイルの撲滅**
4. **自動化の推進**
5. 分散システムのモニタリング
6. 適切なリリースエンジニアリング

運用業務に含まれるトイル(苦労)を自動化により削減する。
トイルとは以下の特徴に当てはまるもの。

- 手作業であること
- 繰り返されること
- 非戦術的であること
- 自動化できること
- 長期的な価値を持たないこと
- サービスの成長に対してO(n)であること

トイルをそのままにすると、生産性低下・キャリアの
停滞・士気の低下等に繋がる。
GoogleのSREはいわゆる「トイル」に費やされる時間を
業務時間の50%未満にすることを目指しています。

・Site Reliability Engineering: How Google Runs Production Systems. (2016) O'Reilly Media.
(Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング —Googleの信頼性を支えるエンジニアリングチーム (2017)オライリージャパン)
・Google - Site Reliability Engineering https://sre.google/sre-book/part-II-principles/ (参照 2022/11/14)

Principle of SRE

1. リスクの受容
2. SLOの定義
3. トイルの撲滅
4. 自動化の推進
5. **分散システムのモニタリング**
6. 適切なリリースエンジニアリング

長期的なトレンドの把握や適切なアラートによる問題解決
を行うため、各種のモニタリング設定を行っていく。
また、システムのモニタリングに閉じず、業務上のトイルの発生頻度や、
トイルに割かれている時間についても可視化を行い、データドリブンな世界に。
（データが無ければSLI/SLOの妥当性や、自動化対象とするトイルの選定を行えない。）

闇雲に戦う世界から

データを元に意思決定が出来る世界へ

DATADOG Prometheus
Cloud watch Cloud Operations

・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング　—Googleの信頼性を支えるエンジニアリングチーム　（2017）オライリージャパン）
・Google - Site Reliability Engineering https://sre.google/sre-book/part-II-principles/（参照 2022/11/14）

3-SHAKE

Principle of SRE

1. リスクの受容
2. SLOの定義
3. トイルの撲滅
4. 自動化の推進
5. 分散システムのモニタリング
6. **適切なリリースエンジニアリング**

多くの障害は人の手が加わることによって発生する。
構成管理やリリースエンジニアリングの仕組みを利用し、
再現可能で自動化されたリリースプロセスを整備する。

- IaC（Infrastructure as Code）
- CI/CD 導入
  - デプロイの自動化
  - コード品質・脆弱性の検査
  - DevSecOps

・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング　—Googleの信頼性を支えるエンジニアリングチーム　（2017）オライリージャパン）
・Google - Site Reliability Engineering https://sre.google/sre-book/part-II-principles/（参照 2022/11/14）

SREの効果

- システムの状態ではなく、信頼性（ユーザ影響）に基づいて、監視を行うことで、
  不要なアラート対応の削減や、稼働時間の確保が見込める。

- SLO定義・エラー予算の導入により、
  障害対応と機能開発間の優先順位を明文化し、開発と運用間で意識統一が測れる。
  （感覚値ではなく、ユーザ影響に基づいて、明文化された基準を元に動ける）

- 手作業（トイル）を削減することで、本来実施したい作業にリソースを投下できる。
  また、ヒューマンエラーの削減による信頼性向上にも寄与する。

- データに基づいて、トイル解消等、取り組みの優先度を決めることができるため、
  納得感や、効率の向上が見込める。

03

SRE導入の難しさと進め方について

SREを導入する上での障壁とアプローチ

組織にSREを導入することは一筋縄では行かない
当社がこれまでの支援の中で直面した障壁とアプローチについて、
説明していきたいと思います。

- 新しいチーム立ち上げの難しさと進め方について
- SREの原則を実践する難しさと進め方について
- SRE人材を集める難しさと進め方について

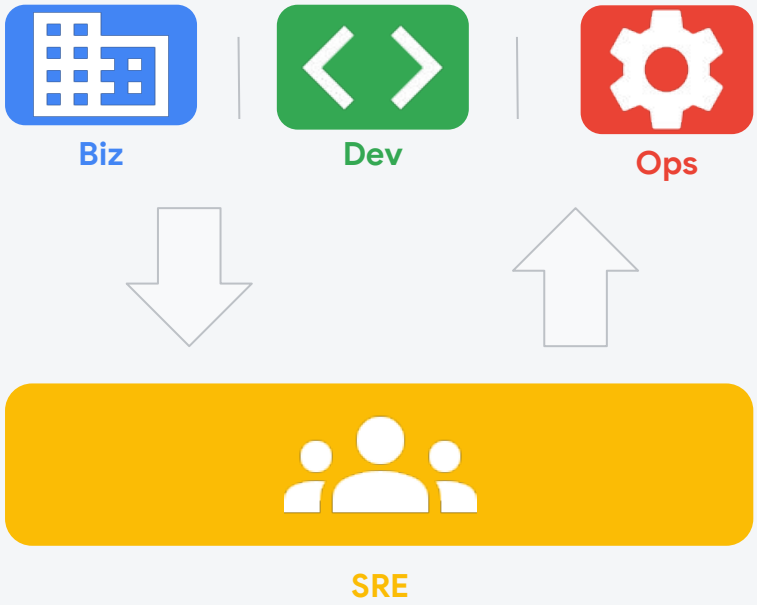
03-01

新しいチーム立ち上げの難しさと進め方について

SREは様々な組織や役割と密接に連携しながら実践していく必要がある。

SRE は、開発やこれまでの運用といったものと密接に関わります。
また、SLO の定義や SLA の定義を行うためには当然 SRE のみで終止するものでは無いため、ビジネスサイドとも密接に関わります。

SRE は単一組織として自分たちのタスクに終始すれば良いものではなく、様々な組織や役割と密接に連携し合いながら、実践していく必要があります。



SRE組織に対する認識のズレは期待通りの結果を得られない原因になる。

- 組織の役割が曖昧でメンバーに目的が浸透しておらずタスクが進まない
- 活動が他のチームや上層部から理解が得られず新しい考え方や仕組みを導入できない
- 現実と折り合いを付けずに到達困難な目標を立ててしまいメンバーが疲弊する
- 新しい活動に割けるリソースがなく、コミット量が少なくおもったような結果が得られない

Googleの提唱するSRE を導入するための段階的なアプローチ

- SRE 基礎教育
- 文化の醸成
- 最初のSLO構築
- 成熟度評価

- 関わり方の定義
- チーム憲章作成
- チーム方針草案
- 書面に残す仕組み
- 可観測性の仕組み
- プリモーテム演習

- ポストモーテム演習
- SLO 仕上げ
- リスク分析
- 設計と運用の見直し
- 成熟度評価

- スキルギャップ分析
- 人材像と採用計画の作成
- 組織外のイネーブルメント
- 導入教育準備
- 人材像と採用計画の適用
- 面接と採用基準の準備
- ポストモーテム輪講
- プリモーテム演習

- ポストモーテム演習
- 新チームの基礎を構築
- SLOのMVP 構築
- チーム方針の修正
- 継続的な改善

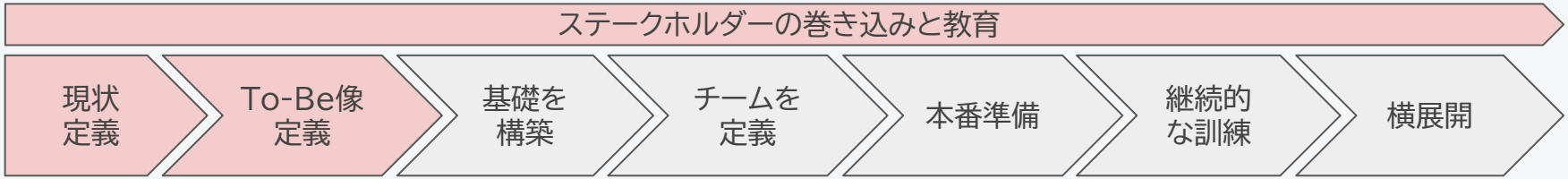
・SRE の基本と組織への導入（ https://services.google.com/fh/files/events/d3-infra-01.pdf）

SREを企業に導入するためのステップ（当社の実践例）

3-SHAKE

SREを導入する組織に併せて導入ステップをカスタマイズする。
（特にチーム組成・関係者巻き込みは組織に併せて慎重に）

例）私たちがあるお客様のSREチームを立ち上げた際のプロセス



- ステークホルダー特定とヒアリング
- 組織体制確認
- 現状課題の整理
- 優先度決め

- 期待効果の定義
- Dev側への期待整理
- SRE導入後の働き方

- 文化の醸成
- 最初のSLO 構築
- 成熟度評価

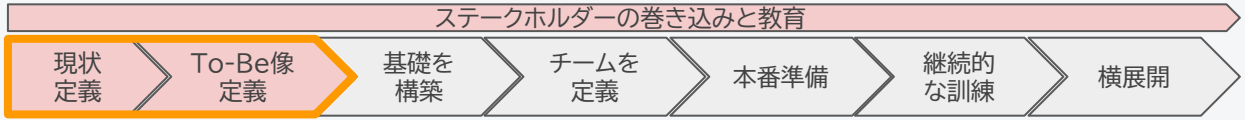
- 関わり方の定義
- チーム憲章作成
- チーム方針草案
- 書面に残す仕組み
- 可観測性の仕組み
- プリモーテム演習

- ポストモーテム演習
- SLO 仕上げ
- リスク分析
- 設計と運用の見直し
- 成熟度評価

- スキルギャップ分析
- 人材像と採用計画諜作成
- 組織外のイネーブルメント
- 導入教育準備
- 人材像と採用計画諜適用
- 面接と採用基準の準備
- ポストモーテム輪講
- プリモーテム演習

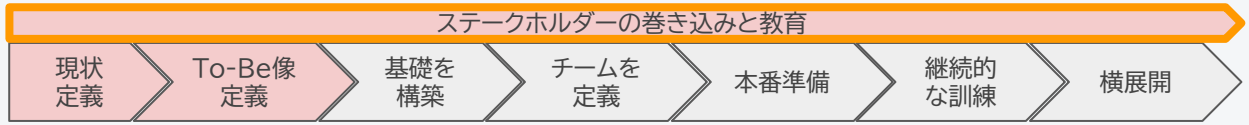
- ポストモーテム演習
- 新チームの基礎を構築
- SLOのMVP 構築
- チーム方針の修正
- 継続的な改善

現状の定義・To-Be像の定義



- 現状定義
  - ステークホルダーへのヒアリング
    - 現在の働き方や組織体制
    - 現状課題、優先度
- To-Be像の定義
  - SRE導入による期待効果
  - SRE導入後の働き方のイメージ
    - ロードマップ、体制図、会議体　etc
  - 関係組織・チームに期待すること
    - 関係組織と協働するために伝えたいこと、伝え方を検討
      - 組織内で通じる言葉になるように準備段階で入念に認識合わせ
      - 導入の経緯、方法、役割の期待値を明らかにして関係組織を困らせないように

ステークホルダの巻き込みと教育



- ステークホルダの巻き込みと教育
  - 不確定な情報を伝えて混乱や誤解を招かないように
    - **各ステップやステップを実現するためのヒアリング**ごとに必要になるメンバーに限定し、巻き込む
    - 前述の通り組織の組成・改革は慎重に行っていく
  - 教育については必要なときにSREに関する知識や事例を展開。
    - 関わるメンバーが一気に増える場面では勉強会やハンズオンを適宜、実施して知識を埋め合わせていく

03-02

SREの原則を実践する難しさと進め方について

SREの実践内容(例)

- 監視基盤導入
  - Logging
  - Monitoring
  - APM
- SLI / SLO の定義

- 運用体制整備
  - インシデント対応・管理
  - 効果的なアラート

- IaC (Infrastructure as Code)
  - 構成管理の品質チェック
  - GitOps

- CI/CD 導入
  - デプロイの自動化
  - コード品質・脆弱性の検査
  - DevSecOps

- パフォーマンス分析
  - 分散トレーシング
  - 負荷試験
  - カオスシナリオ試験

SREの実践内容(例)

- 監視基盤導入
  - Logging
  - Monitoring
  - APM
- SLI / SLO の定義
- CI/CD 導入
  - デプロイの自動化
  - コード品質・脆弱性の検査
  - DevSecOps

これを複数のサービスにまたがって一気に全部実行するのは相当困難。
本音を言うと不可能に近い。

- IaC (Infrastructure as Code)
  - 構成管理の品質チェック
  - GitOps

小さく始めて SRE チームとして成功体験を得よう

多数のシステムを一気に対象とせず、
まずはひとつのサービスを対象に SRE を始めて
徐々に広げていきましょう。

（イメージ）

1プロダクトSRE導入 → 1プロダクトSRE導入 → 1プロダクトSRE導入

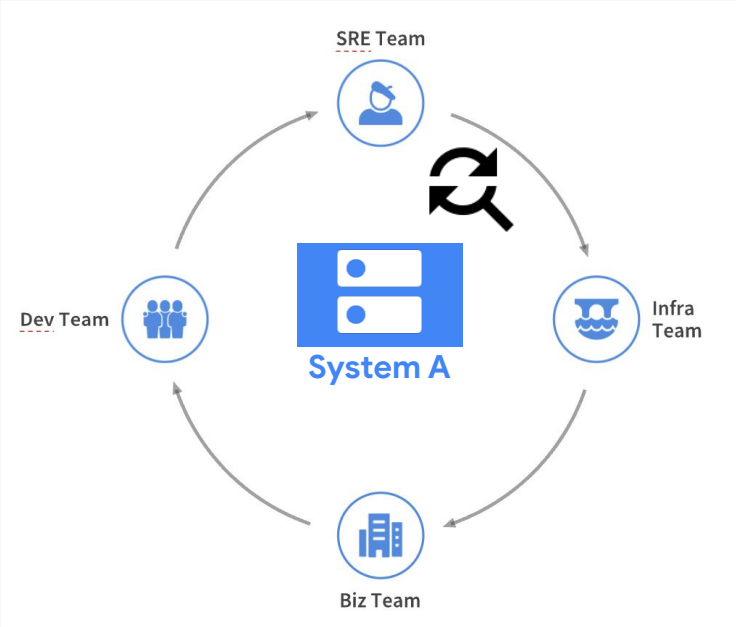
一定周期で各サービスへSRE導入を行う
（SLO・エラーバジェット策定 / 監視基盤の改善等）

横断的なSRE推進

横断的なSREに関する課題・インフラ整備に取り組む

出来れば新規サービスが望ましい。

- 新しく SLI / SLO 設定する元となる情報を取得する為の基盤導入が難しい。
- 既存の運用がある為、新しい活動への導入が難しいケースがある
- そもそも手を加える前提で作られていない...



ひとつのサービスと
それに関わるチームから SRE の活動を始める

3-SHAKE

上質な SLO を設定するためにもまずは あらゆる指標となりうるあらゆるメトリクスを収集し、可視化ができるようにしよう

SLI（Service Level Indicator）：サービスレベル指標
・システムの良し悪しを判断するための指標。

SLO（Service Level Objectives）：サービスレベル目標
・サービスレベルに対する内的な目標値。SLIを元に定義。

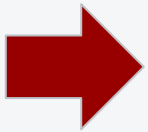
Error Budget：エラー予算
・許容できる不具合の割合（1ーSLO）
　エラー予算ポリシーを定め、ポリシーに従って対応する。

SRE実践例)トイルの削減

3-SHAKE

Toil とは、手作業、繰り返される、自動化が可能、非戦術的、長期的な価値がない、サービスの成長に比例して増加する、といった特徴を持つ作業です。

- Dev Team からの依頼対応
- 障害対応
- 割り込みタスク
- SRE としての定型業務
- 定期的に発生する
  メンテナンス作業



まずはこれらを
すべて測定することから

どういう作業がどの程度の頻度発生し、どれだけの
工数をかけてどういう効果が得られたのかを取得する

SRE実践例）ポストモーテム

ポストモーテムとは、インシデント、その影響、インシデントを軽減または解決するために取られたアクション、根本原因、およびインシデントの再発を防止するためのフォローアップアクションを書面で記録することです。

- 振り返りから学ぶ文化を作る為にポストモーテムを行います。
- その為には障害解析に対して追跡が出来るようにしておかなければなりません。
- フォーマットについてはまずは SRE 本にて提示されている Example を流用するのがおすすめ。
- それと何よりも大切なのが特定個人を非難しない文化であることです。
  ※ あくまで、システムや仕組みの問題であると考えます。

例えば決済基盤の信頼性を考えてみる

- 金銭の取引であるため1件もトランザクションを取りこぼしたくないケースもある
  - 例外的な、手作業による再実行処理はできるだけ実施しないが理想
  - 取りこぼしは決済店舗からの信頼を失うし、全店舗にSREを浸透させるのは令和では夢物語
  - ユーザが「システム側で再実行してくれるはずなので待つか」という世界は現実的ではない
- サービスの特性によってはエラーを許容するSLOやエラー予算の考え方と相性が悪い場合がある

単純にSLOでは計測できない信頼性が大事な要素である場合

サービスによっては数値で測定することが難しいものが、信頼性の一部となる大事な要素が存在する

理想

SLOを守ってるので
トイル削減！

SLOを守ってるので
新機能開発！

現実

SLO守っているが、
SNSが炎上してる...
トイル削減できない...

重要な決済機能で
エラー出てるから
新機能開発できない...

SREの原則と相性が悪い場合の進め方

例えばあなたのサービスで開発効率を上げることが一番大きい課題
→一時的にSLOによる、ハンドリングを一旦ストップ

別のデータに基づいて、開発生産性を向上させたり、別のSRE原則から始めるのも一手
＝SREを諦めるのではなくSREを導入するために改善を進めていく
　また効果が大きいところから小さくSREを始めることで実績を作って行くことが大事

ex. ソフトウェア開発のパフォーマンスを数値化したFour Keysやそれに準じた値により業務をコントロール

03-03

SRE人材を集める難しさと対応について

3-SHAKE

SRE人材は多くの能力を求められる

スキルセット

オペレーティングシステムの内部、ネットワーキング、モニタリングと アラート、トラブルシューティング、デバッグ、インシデント管理、ソフトウェアエンジニアリン グ、ソフトウェアのパフォーマンス、ハードウェア、分散システム、システム管理、キャパシティ プランニング、セキュリティ、その他にも多くの領域…

技術以外にもサービスと対峙する中で以下のような能力が求められる

- 関係組織にオンボードしてそのチームを知っている
- サービスにオンボードしてそのサービスを知っている
- 少なくとも 1 つの既存サービスに対して重要な変更を実施することができる
- 自分たちのシステムが関連する、他システムの変更に対処が行える
- 爆発的な成長や他の重要なシステム課題への対処ができる

これらすべてを満たす人材を見つけることは困難

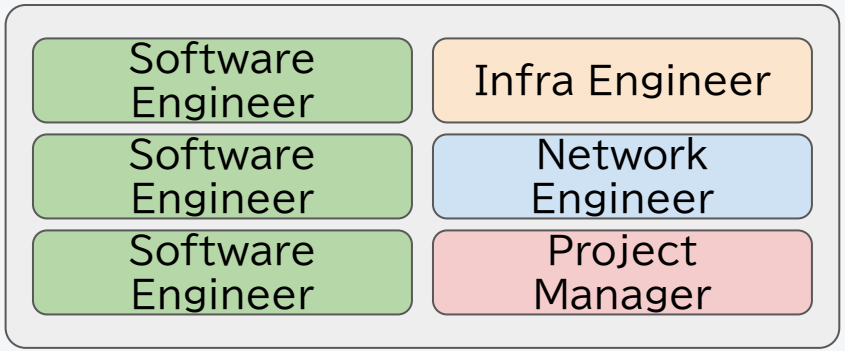
・Seeking SRE: Conversations About Running Production Systems at Scale（2018）O'Reilly Media.
（David N. Blank-Edelman 渡邉 了介 訳 SREの探求 ―様々な企業におけるサイトリライアビリティエンジニアリングの導入と実践　（2021）オライリージャパン）
20章 アクティブなティーチングとラーニング

各々得意なスキルを持ったメンバーで一枚岩でSREを実現する

3-SHAKE

自分たちのシステムの規模、やりたい事に応じて、
メンバーを構成して一つのチームとしてSREを提供する

※ Googleではソフトウェアエンジニアが50%～60%
　残りはそれ以外のSREに必要なスキルを持ったメンバー
　（とはいえGoogleの場合は、ほぼソフトウェアエンジニアに近いスキルの人材とのこと）



・Site Reliability Engineering: How Google Runs Production Systems.（2016）O'Reilly Media.
（Niall Richard Murphy 他 玉川 竜司 訳 SRE サイトリライアビリティエンジニアリング　—Googleの信頼性を支えるエンジニアリングチーム　（2017）オライリージャパン）
1.2 サービス管理への Googleのアプローチ：サイトリライアビリティエンジニアリング

SREを推進したい運用と導入対象のプロダクトでバーチャルチームを作成

- ToBe像を元に必要なメンバーを揃えてSRE導入を推進する部隊を結成
- 開発からSREに留学してもらうことで内部での運用について知ってもらう

内製化やさらなる増員を考えるとSREを教育していく必要もある

- SREの探求で紹介されているアクティブラーニングがおもしろい
  - アクティブラーニング
    - 座学ではなく生徒が能動的に考え、学習する教育方法
  - アクティブラーニングの例としてゲームが紹介されている
    - ロールプレイングゲームによる障害対応訓練
      a. リスクやプレッシャーの少ない
      b. 障害対応、原因分析、インシデント管理について学べる
    - カードゲームによるインシデント対応
      - インシデント管理で重要なコミュニケーション、調整、責任の分離について学ぶ

・Seeking SRE: Conversations About Running Production Systems at Scale（2018）O'Reilly Media.
（David N. Blank-Edelman 渡邉 了介 訳 SREの探求 ―様々な企業におけるサイトリライアビリティエンジニアリングの導入と実践　（2021）オライリージャパン）
20.1 アクティブラーニング、20.1.1 アクティブラーニングの実例：Wheel of Misfortune、20.1.2 アクティブラーニングの実例：Incident Manager（カードゲーム）
・インフラエンジニアの卵になってもらう研修　https://www.slideshare.net/kamaekawa/infraeduhb（参照 2022/11/14）

その他に学習習慣として紹介されている例は泥臭い。

- 障害やサービスについて議論するMTGに参加してもらい疑問点をリストにしてもらう
  - 後ほどメンターが回答して育てていく

- 初めて担当するシステムの要件を把握してから設計のスケッチを自分なりにまとめる
  - 何もわからない状態から自分で調査
  - 既存資料で答え合わせ
  - 時間はかかるけど楽しいし効率が良いと感じられるとのこと

・Seeking SRE: Conversations About Running Production Systems at Scale（2018）O'Reilly Media.
（David N. Blank-Edelman 渡邉 了介 訳 SREの探求 ―様々な企業におけるサイトリライアビリティエンジニアリングの導入と実践　（2021）オライリージャパン）
20.1.3 アクティブラーニングの実例：SRE Classroom

最後に

- Googleのとあるソフトウェアエンジニアがシステム運用を任されて、
  長い時間をかけて**理想を形にした結果**がSRE。

- Google の SRE は素晴らしい見本になるが、
  現在サービスを運用する上で最適だと言われる**方法論の一つ**であり、
  SRE本の通りに実践することが真のゴールではない。

- SREのプラクティスを自分たちなりに解釈し、**組織に合ったSRE**を手のつけ
  やすいところから小さく初めて行こう。

04

SRE総合支援サービスのご紹介

スリーシェイクのSRE総合支援（Sreake）とは何か

Sreake

- SREの考え方に従って、AWSやGoogleクラウドを利用しているサービスの
  技術戦略、設計、構築、運用までワンストップで対応。
  作って終わりのSIとは異なり、最終的にお客様が内製化・自走することをゴールとし、
  運用を常に考えたご支援が特徴です。

- マイクロサービス、k8sなどクラウドネイティブな技術領域に強み

スリーシェイクのSRE総合支援(Sreake)とは何か　～SREパッケージ～

Sreake

	GOAL			
	Entry	**Starter**	**Professional**	**Forward support**
Infra-Modernazation	・クラウドアセスメント ・Googleクラウドワークショップ ・AWSワークショップ	・ベストプラクティス環境パイロット	・ネットワーク設計・構築・運用支援 ・アーキテクチャー設計構築・運用支援 ・マイグレーション設計・構築・運用支援	・SRE導入支援 ・SRE組織構築支援 ・MSPサービス
App-Modernazation	・マイクロサービスの考え方 ・k8s入門 ・DevOpsの考え方	・EKS、GKEパイロット ・Cloud runパイロット ・CI/CDパイロット ・Anthosパイロット	・k8s設計・構築・運用支援 ・CI/CD設計・構築・運用支援	・業界別ワークショップ ・業界別ソリューション構築・運用支援
Data/Analytics	・BigQueryワークショップ ・Lookerワークショップ	・BigQueryパイロット ・Lookerパイロット	・データ基盤設計・構築・運用支援	―
AI/ML	・MLアセスメント ・MLシリーズ（BQ ML, Dataflow ML, VertexAI）ワークショップ	・MLシリーズパイロット	・ML基盤設計・構築・運用支援	―
Security	・クラウドセキュリティアセスメント	・クラウドセキュリティパイロット	・セキュリティ設計・構築・運用支援	・脆弱性診断サービス ・バグバウンティサービス
Monitoring	・APM・外形監視ワークショップ ・PagerDutyワークショップ	・DatadogやCloudMonitoringなどのパイロット ・PagerDutyのパイロット導入	・DatadogやCloudMonitoringなどを使用した監視設計・構築・運用支援 ・PagerDutyを利用した運用効率化支援	―
共通	・技術Q&A			

スリーシェイクのSRE総合支援(Sreake)とは　～SREパッケージ～

Sreake

	GOAL			
	Entry	Starter	Professional	Forward support
Infra-Modernazati	・クラウドアセスメント ・Googleクラウドワークショップ	・ベストプラクティス環境パイロット	・ネットワーク設計・構築・運用支援 ・アーキテクチャー設計構築・運用支援	・SRE導入支援 ・SRE組織構築支援 ービス ハンズオン
App-Modernazati				ークショップ リューション構築・ 援
Data/Analyti				
AI/ML				
Security				診断サービス ンティサービス
Monitoring	・APM・外形監視ワークショップ ・PagerDutyを利用したアラート対応改善	・DatadogやCloudMonitoringなどのパイロット ・PagerDutyのパイロット導入	・DatadogやCloudMonitoringなどを使用した監視設計・構築・運用支援 ・PagerDutyを利用した運用効率化支援	ー
共通	・技術Q&A			

お客様の状況を踏まえ、
SREのメソッドを利用したご支援をパッケージとしてご提供しております。

(例)
- エンジニア向けのマイクロサービスのワークショップ・ハンズオン
- Datadog・PagerDutyを利用したアラート対応効率化のご支援
- CI/CD環境の構築ご支援

SRE総合支援(Sreake)の事例

Sreakeでは、主に次の3パターンのご支援をしております。

- **新規システムの企画・設計からのご支援**
  新規システムを企画・設計する所からご支援をします。
  CI/CD、運用監視基盤などを含めたインフラ領域は全て対応します。

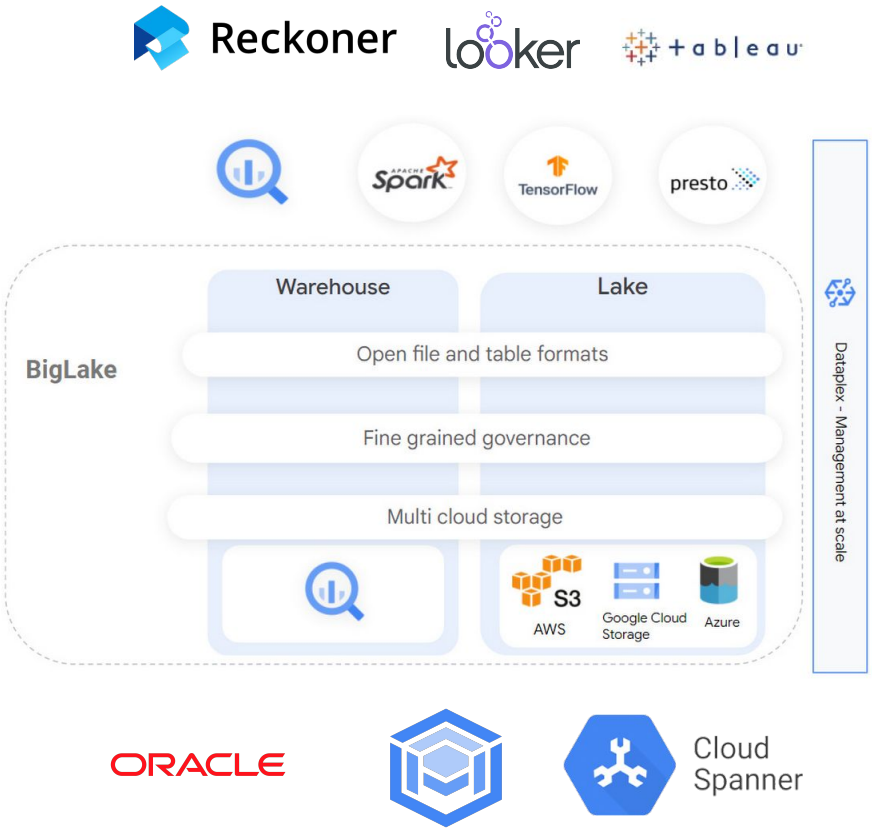
- **運用中システムの運用改善からのご支援**
  運用中の様々な課題を改善するところからご支援をします。
  徐々にSREの考え方に従い、効率的な運用を実現できるようにご支援をします。

- **SRE組織作りのご支援**
  自社の中にSREの組織を立ち上げる際のご支援をします。

お客様の割合

- 規模　スタートアップ:50%、Enterprise:50%
- 業種　金融:50%、 ITサービス業:30%、 製造業:10%、 その他:10%

（ケイパビリティ強化）データモダナイゼーション/DBRE支援



BigQuery/Dataplexを中心としたデータ基盤構築及び、BI（Looker, Tableau等）構築をフルスタックで支援

データレイヤーのプロフェッショナルチームが
Google Cloudをフル活用したデータモダナイゼーションを実現します。

DBREとしてCloud Spanner/AlloyDB/Oracle（RDBMS）のデータセキュリティ、パフォーマンス、運用最適化支援

RDBMS ～ NewSQLに対してのアーキテクト（スキーマ設計、クエリチューニング、トラブルシュート）、データ移行、アプリケーションエンジニアへの教育（SQL）支援、データガバナンス支援を網羅的に支援します。

オンプレミスからのAlloyDB/Spannerへの移管などを加速させ
他パートナーにない手厚いエンプラサポートを提供

（ケイパビリティ強化）SREエンジニア採用支援・育成支援

エンジニア不足を解消し、内製化を加速するご支援（インフラエンジニア、SRE・DevOpsエンジニア）

クラウドネイティブな領域を中心にSRE総合支援を行ってきたスリーシェイクでは、最終的にお客様が内製化することを目指しております。
しかし、エンジニア不足、エンジニアのスキルに課題を持つお客様が非常に多いです。

SREエンジニアの採用・育成に強みを持つスリーシェイクにて、お客様のエンジニア採用支援・育成支援を行います。

エンジニア育成支援例：

- キャリアパスやキャリア育成の策定支援
- エンジニア評価制度の作成支援
- エンジニアブランディング支援
- 各種ワークショップ・ハンズオン開催

SRE総合支援(Sreake)のパートナー企業

Sreake

プラットフォーム

aws

Google Cloud

CI/CD

GitLab

circleci

統合監視／アラート

DATADOG

PagerDuty

データ分析基盤

Looker

BigQuery

スリーシェイクではコンサル領域・技術面での強みを生かした、導入・利用サポートに強みがあります。
（パートナー企業の各種サービスについてスリーシェイク経由でのリセールも可能です）