Site Reliability Engineeringとは。

高信頼性運用を実現するSREという新潮流 @ InternetWeek 2017

藤崎 正範

自己紹介

@fujisaki_hb

fujisaki.masanori

藤崎正範

株式会社ハートビーツ　代表取締役　https://heartbeats.jp/
株式会社ウォルティ　代表取締役https://walti.io/
日本MSP協会　　理事　　https://mspj.jp

HEARTBEATS　Walti　MSPJ MANAGED SERVICE PROVIDER'S Association Japan

インフラエンジニア勉強会hbstudy　➜ SRE大全という特集を企画
July Tech Festa 実行委員　➜ SREセッションを担当
InternetWeek 2017 プログラム委員　➜ 本プログラム担当

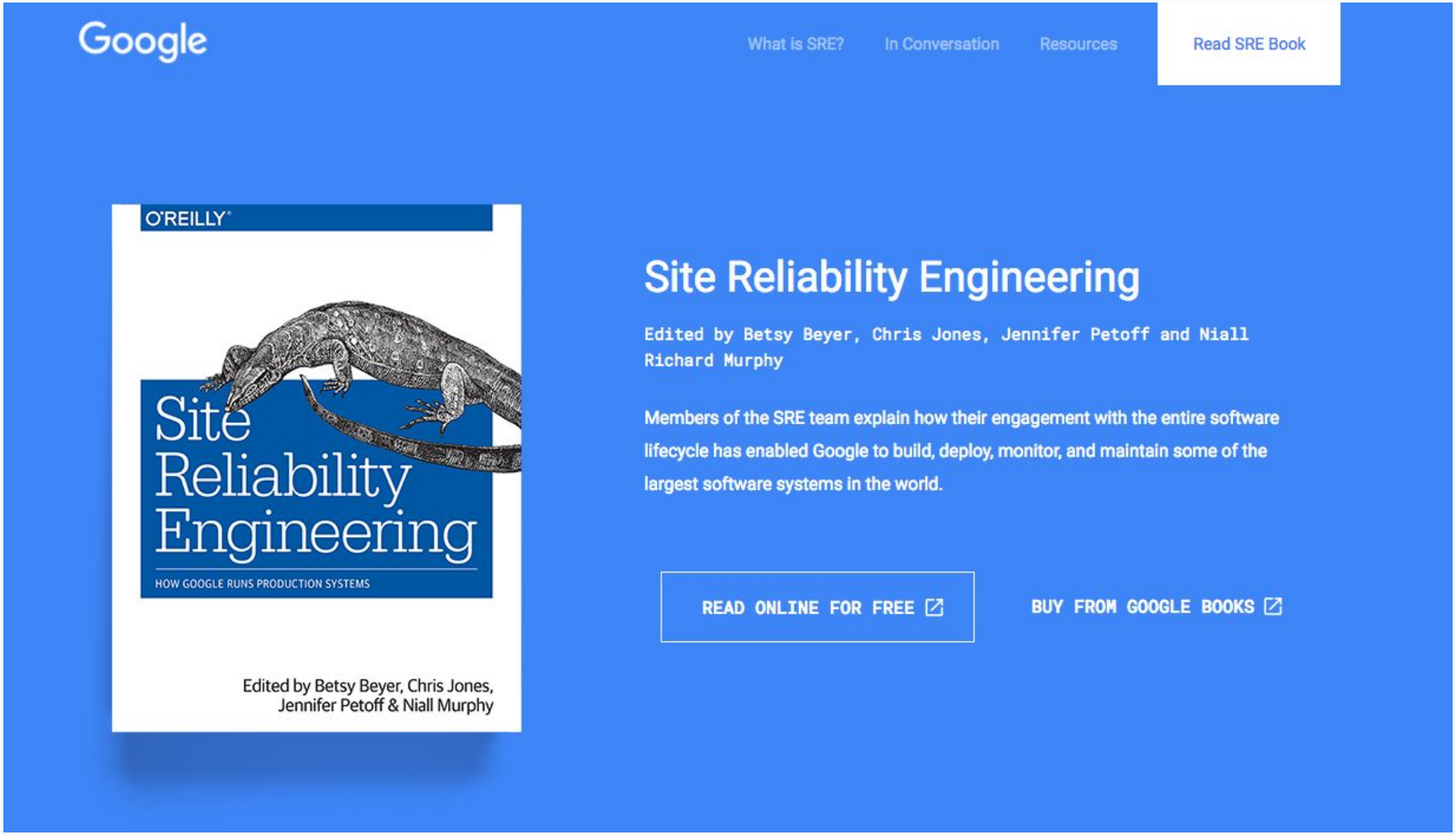
今日のAgenda

- Site Reliability Engineering
- 国内のSRE動向
- SREという役割
  - SREの信条
  - ポストモーテムの文化
  - トイル
  - SREにおけるソフトウェアエンジニアリング
  - SREの成長を加速させる方法
- まとめ

Site Reliability Engineering

The SRE Book

https://landing.google.com/sre/book.html



もちろん日本語版もあります。

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

Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy　編、澤田 武男、関根 達夫、細川 一茂、矢吹 大輔　監訳、Sky株式会社 玉川 竜司　訳
2017年08月 発行
590ページ
ISBN978-4-87311-791-1
フォーマット Print PDF ePub mobi
原書: Site Reliability Engineering

オライリー・ジャパンで書籍を購入:
定価5,184円
カートに入れる

Ebook Storeで電子版を購入:
価格4,147円
カートに入れる

ツイート　G+　いいね！ 342

https://www.oreilly.co.jp/books/9784873117911/

SRE

- **Site Reliability Engineering**（SRE）
  - サイトの信頼性向上を目的としたエンジニアリング手法

- **Site Reliability Engineer**（SRE)
  - Ops側の職種・人を指す
  - サイトの信頼性に関すること全てに関わる
  - Site Reliability Engineeringを実践・組織に浸透させる役割を担う

  - SREチーム
    - ソフトウェアエンジニア約50%～60%
      特定分野のスペシャリスト約40%

Site Reliability Engineering

by Wikipedia

SREはいつからあるか？

- Ben Terynor @Google
- 2003年頃、7人のエンジニアで、複数のサイトを支障なく効率的に、より信頼性を持って運用していくタスクをこなしていたところ、**新しい機能を継続的に導入すると同時に、巨大なシステム群を運用する枠組み**が必要だった(意訳)

https://en.wikipedia.org/wiki/Site_reliability_engineering

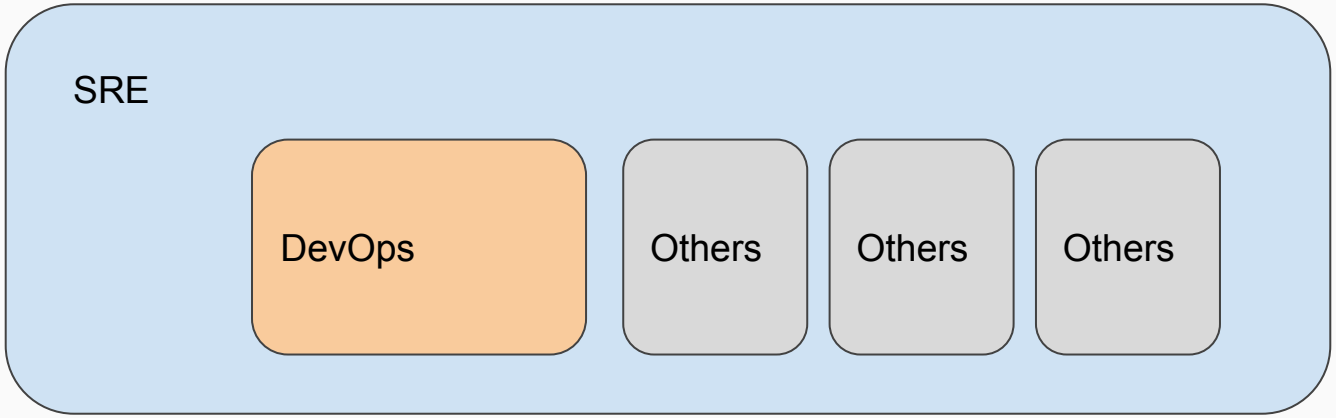
Googleの他の海外のSREチーム

- Microsoft
- Apple
- Twitter
- Facebook
- Dropbox
- Amazon
- IBM
- Xero
- Oracle
- GitHub　….他多数

https://en.wikipedia.org/wiki/Site_reliability_engineering

DevOps vs SRE

"Site Reliability Engineering is a superset of processes that would, inherently, include DevOps as a subset."



https://en.wikipedia.org/wiki/Site_reliability_engineering

国内のSRE動向

国内のSREチーム

- メルカリ
- ミクシィ
- クックパッド
- Retty
- freee
- サイボウズ
- クラスメソッド
- SmartHR
- オールアバウト
- nulab
- DMM.com ラボ
- 弁護士ドットコム
- エウレカ
- UZABASE
- ココナラ
- Gunosy

....等多数

Googleに(検索で)聞いてみた。

2015年11月！

WantedlyでSRE関連の採用数

2017年8月　60件
2017年11月　87件

3ヶ月で
45%増！

国内のSRE求人の掲載企業(Wantedly)

- freee
- サイボウズ
- エニグモ
- GVA Tech
- WAMazing
- ネクストビート
- MAMORIO
- CAMPFIRE
- キュア・アップ
- TABI LABO
- FOLIO
- Kaizen Platform
- Misoca
- オズビジョン
- イタンジ
- カケハシ
- オルトプラス
- BASE
- Progate
- メディカルノート
- Gunosy
- エウレカ
- dely
- LOB
- Repro
- カオナビ
- EMTG
- mixi

....等多数

SREという役割

SREチーム の役割

- 可用性
- レイテンシ
- パフォーマンス
- 効率性
- 変更管理
- モニタリング
- 緊急対応
- キャパシティプランニング

SREの活動は大まかに次のように分類される
- ソフトウェアエンジニアリング
- システムエンジニアリング
- トイル
- オーバーヘッド

➜**単なる運用組織**ではない

SRE の信条　(Google)

1章より

SRE の信条　(Google)

- エンジニアリングへの継続的な注力の保証
- SLOを下回ることなく、変更速度の最大化を追求
- モニタリング
- 緊急対応
- 変更管理
- 需要の予測とキャパシティプランニング
- プロビジョニング
- 効率とパフォーマンス

SRE の信条　(Google)

- エンジニアリングへの継続的な注力の保証
  - **運用作業50%**まで
  - 残りは**コーディングスキル**を使うプロジェクトの作業にあてる
  - 運用作業が50%超えたら運用業務を**プロダクト開発チームへ差し戻し**
    - バグとチケットを開発チームに割り当てなおし
    - 開発者をオンコールのページャーローテーションに組み込んだり

  - すべての大きなインシデントには「**ポストモーテム**」を書くべき
    - ポストモーテムで**非難をしない文化**
      - 問題を回避・縮小したりするのではなく、問題を明らかにし、**エンジニアリングで問題を解決することを目標**

SRE の信条　(Google)

- SLOを下回ることなく、変更速度の最大化を追求
  - 協力関係のために、DevとOpsの構造的な対立を取り除く必要がある
    - **イノベーションの速度** vs **プロダクトの安定性**

  - 「**エラーバジェット**」の導入
    - 100%を目標にするのは、基本的に間違っている
      - 100%と99.999%の0.001%差を埋める労力はメリットがない
    - SLO 99.999%の場合、0.001%がエラーバジェット
      - エラーバジェットを**リスクを取るために活用**し**素早いローンチ**
    - サービス障害は「悪い」ことではない
      - イノベーションのプロセスにおいて**予想済み**
      - SREの目標は「サービス障害をゼロにすること」ではない

SRE の信条　(Google)

- モニタリング
  - システムの健全性と可用性を追跡するための主要な手段
    - メールを読み、アクションの有無を判断しないといけない
      - 根本的に欠点がある
    - アラート内容は**ソフトウェアが解釈**を行う
    - 人が**アクションしないといけない時だけ通知**するべき

  - 効果的な出力
    - **アラート**: 人がアクションを起こし、緊急に対応が必要な場合
    - **チケット**: 緊急にではないが、人が対応必要な場合
    - **ロギング**: 人が見る必要がないもの
      (通常、誰かが読むことは期待されていない)

SRE の信条　(Google)

- 緊急対応
  - 信頼性(稼働率)は、**平均故障時間**（mean time to failure = **MTTF**)と**平均修復時間**(mean time to repair = **MTTR**)で算出
  - この信頼性を向上させるには、人が介在するMTTRを無くす(自動復旧)、もしくは、事前に手順書を記録しておく(MTTRが3倍改善する)

稼働率　$$A(MTBF, MTTR) = \frac{MTBF}{(MTBF + MTTR)}$$

※MTBF(平均故障間隔
=Mean time between failure.

https://www.vividcortex.com/blog/the-factors-that-impact-availability-visualized

  - SREは、手順書を使ってロールプレイングの訓練も行う

SRE の信条　(Google)

- 変更管理
  - 70%のサービス障害は、動作中のシステム変更に起因していた
  - 自動化によって以下を実現
    - **段階的な**ロールアウトの実装
    - 高速かつ正確な問題の検出
    - 問題が発生した際の**安全なロールバック**
  - 変更に起因する**トラブルの影響を、効率的に最小化**する
  - 自動化のメリット
    - 繰り返しタスクが減り、疲労しない
    - 慣れ・軽視・不注意といった**人だから発生する問題を回避**
  - 結果的に、リリースの速度と安全性が共に高まる
    - **リリースエンジニアリング**

SRE の信条　(Google)

- 需要の予測とキャパシティプランニング
  - 需要に対して、可用性を維持するために、キャパシティと冗長性を確保
  - 需要の予測
    - **自然な成長**：利用者増からなる自然に生じる
    - **突発的な成長**：新機能のローンチや広告キャンペーン等に起因する

  - キャパシティプランニングのステップ
    - 自然な需要の正確な予測を行う
      （リソース確保のリードタイムも計算に入れる）
    - 需要予測に、突発的な需要を正確に盛り込む
    - サービスのキャパシティと、リソースのキャパシティの関連を把握
      - システムへ**定期的な負荷テスト**を実施

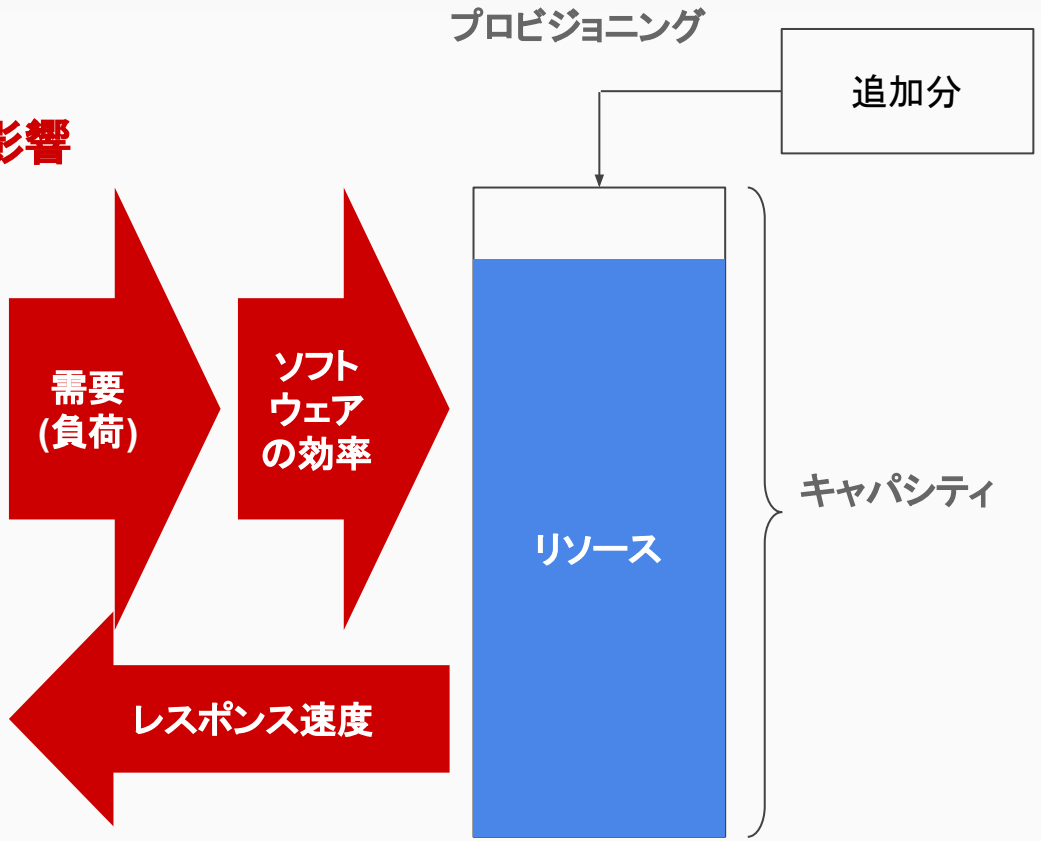
SRE の信条　(Google)

- プロビジョニング
  - 変更管理とキャパシティプランニングの組み合わせ

  - 操作リスクが大きく、注意が必要
    - 具体的には
      - キャパシティは**高価**なので、素早くかつ**必要な場合のみ**実施
      - リソースが活用されるように、サービスに**ちゃんと組み込む**
      - 追加されたリソースが**正常動作するか検証**する

SRE の信条　(Google)

- 効率とパフォーマンス
  - サービスの**トータルコストに影響**
  - リソースの利用
    - 需要（負荷）
    - ソフトウェアの効率
    - キャパシティ
  - **レスポンス速度の目標**を満たすようにプロビジョニング



ポストモーテムの文化

15章より

ポストモーテムの文化

- **ポストモーテム**とは**検死**のこと
  - 障害が発生した際に残すべき**ドキュメント**
    - 何が起きたのか
    - 対応の有効性
    - 次回、変えるべき行動は何か
    - インシデントの再発防止のためにすることは何か

- 書く目的
  - インシデントが**ドキュメント化**されること
  - 影響を及ぼした**すべての原因(群)が十分に理解**されること
  - 再発の可能性や影響を削減するため、
    効果的な**予防策が確実に導入**されるようにすること

ポストモーテムの文化

- ポストモーテムを**書く条件を事前に明示**しておくことが大切

  ポストモーテムを書く条件の例
  - **ユーザー影響**が一定の閾値を超えた場合
  - 種類の如何を問わず、**データの損失**が生じた場合
  - **オンコールエンジニアの介入**が必要だった場合
    （リリースのロールバック、トラフィックのルート調整など）
  - **解決までの時間**が一定の閾値を超えた場合
  - モニタリングの障害（**人によって発見**された場合）

ポストモーテムの文化

- ポストモーテムで**批判を行わない**
  - 個人やチームの「間違った」行いに対する**指弾と不名誉の文化**が広がってしまえば、人は**処罰を恐れて問題を公表しなくなる**
  - ポストモーテムは、非難をするためのものではなく、サービスの**どの部分をどうすれば改善できるのか提起**するものでなければならない

- 非難を避け、**建設的であり続ける**こと
  - ポストモーテムから非難を取り除けば、人は**自信をもって怖がらずに問題をエスカレーション**するようになる
  - ある人物やチームが頻繁にポストモーテムを作成しているというな**烙印を押したりしない**ことも重要

ポストモーテムの文化

- ポストモーテムの書き方
  - **リアルタイム**コラボレーション
  - **オープン**なコメント / アノテーションシステム
  - メールによる**通知**

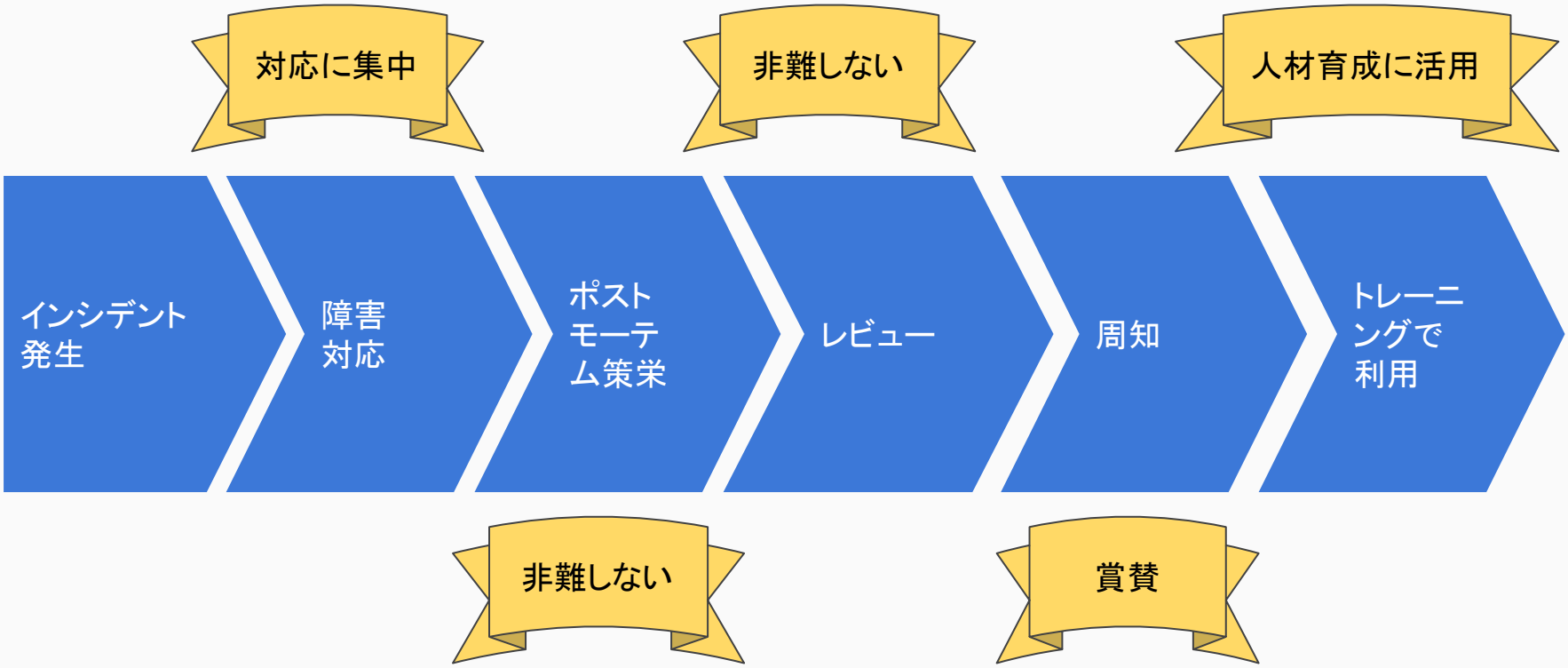
- ポストモーテムのレビュー
  - 後々のためにインシデントの**主要なデータ**は収集されたか？
  - **インパクトの分析**は完全か？
  - **根本原因**は十分に分析されているか？
  - **アクションプラン**は適切で、バクの修正の**優先順位は適切**か？
  - 結果は関係する**ステークホルダーたちと共有**されたか？

ポストモーテムの文化

- ポストモーテムを**文化にするための活動**例
  - 今月のポストモーテム
  - Google+ポストモーテムグループ
  - ポストモーテム読書会
  - 不運の輪
    （ロールプレイングによるトレーニングで過去のポストモーテムの再現）

- 組織に導入するために
  - ポストモーテムを**ワークフローに落とし込む**
  - 効果的なポストモーテムを書くことは**報われること**であり、
    賞賛されることであるようにする
  - **上位のリーダーたちの承認と参加**を奨励する

ポストモーテムの文化


対応に集中
非難しない
人材育成に活用
インシデント発生
障害対応
ポストモーテム策栄
レビュー
周知
トレーニングで利用
非難しない
賞賛

トイル

5章より

トイルの定義

- トイルとは
  - プロダクションサービスを動作させることに関係する作業
  - **手作業**で繰り返し行われ、**自動化が可能**
  - 戦術的で**長期的な価値を持たない**
  - 作業量が**サービスの成長に比例する**といった傾向を持つ

- オーバーヘッド (not トイル)
  - チームミーティング、ゴールの設定と評価、
    作業内容の記録(スニペット)、人事の事務作業等

トイルの定義

- トイルとは
  - 手作業であること
    - スクリプトを手作業で実行するようなタスク
    - **人間が使う時間**は**トイルの時間**(not スクリプトの実行時間)
  - 繰り返されること
    - 2回目程度であればトイルではない
    - 繰り返し何度も何度も行われること
      （新たな問題の解決はトイルではない）
  - 自動化できること
    - マシンが人間同様に行えるタスク
    - 人間の判断が欠かせないのであればトイルではない

トイルの定義

- 戦術的であること
  - 戦略的や予測から始まるものではなく、**割り込み**で始まる
    - 問題などが生じたことへの対応として行われる作業
  - トイルを完全に無くすことはできないが、
    最小限になるよう継続的に努力が必要

- 長期的な価値を持たないこと
  - タスクを終えた後、サービスが同じ状態のままなら、きっとトイル
  - サービスに**恒久的な改善が加えられたら**、それはトイルではない
- サービスの成長に対して比例して増えること
  - サービスのトラフィック量・ユーザー数等に
    比例してスケールするようなタスクはトイル

SREにおけるエンジニアリング

- トイルを減らし、サービスをスケールさせる作業
  - エンジニアリング作業によって
    SREの組織の規模はサービス規模に比例しない
  - 純粋な開発・運用チームに比べて、
    より効率的にサービス管理できるようになる
  - 採用でも**SREが典型的な運用の組織ではないことを約束**する
    - SREの組織を、単なる運用チームに退化させないように

トイルは常に悪なのか？

- **少量**であれば気にすることはない
  - **達成感**と、手っ取り早い**勝利の感覚**を生んでくれる
  - **多少のトイルは避けられない**ものだと誰もが認識しておく

- 大量になってしまうと、、、
  - **キャリアの停滞**
    - プロジェクトに時間が使えなくなると、キャリアップが遅延・停滞
    - つまらない仕事からキャリアを生み出すことはできない
  - **モラルの低下**
    - 耐えられるトイルの量には誰にも限界がある
    - あまりに多すぎるトイルは、燃え尽きや倦怠、不満につながる

トイルは常に悪なのか？

エンジニアリングに使うべき時間までトイルに使いすぎると、次のような問題

- 混乱の発生
  - エンジニアリングを行う組織のはずなのに、**SREの職務について混乱**
- 進歩の速度低下
  - プロダクトの**機能開発の速度が低下**
- 習慣づけ
  - 他のチームから**トイルを受け取ってくれる先と見られる**
- 摩擦の発生
  - 自分はトイルに不満がなくても、**チームメイトはトイルが好きではない**
- 信義違反
  - SREで採用された人は**騙されたと思う**。モラルが低下する。

SREにおける
ソフトウェアエンジニアリング

18章より

SREにおけるソフトウェアエンジニアリング

- **SREとしての経験を持つエンジニア**が**ソフトウェアを開発**するメリット
  - 自社固有のプロダクトに幅広く深い知識を持っている
    - **十分に考慮**を払いながらソフトウェア設計・作成を行える
  - 「プロダクション環境を動作させ続けるためのツール開発」という領域に普段から取り組む
    - **要求や必要事項を容易に理解**できる
  - 想定されるユーザーは直接関わりがある同僚
    - 率直で発信力の強いユーザーフィードバックが得られる
    - 開発とイテレーションを**より高速に**行うことができる

SREにおけるソフトウェアエンジニアリング

- ソフトウェアを開発する際のガイドライン
  - **明確なメッセージ**の作成と発信
    - SREの助けとなる注目せざるをえないようなケースから始めよう
  - 組織の機能の評価
    - SREは**PM経験**などは採用では重視していない
    - **自社内で**必要なスキルを持っている人の力を借りよう
  - ローンチ＆イテレート
    - 第1ラウンドは、比較的単純明快かつ**達成可能**で、
      議論の余地や既存のソリューションがないような目標とすべき
    - 高速なデリバリーとフィードバックが得られるように
      プッシュオングリーンモデルに移行も
  - 基準は下げない
    - プロダクト**開発チームと同じ基準**を自分に課して抵抗しよう

SREの採用

SREの採用

- 基準が非常に高い
  - ソフトウェアエンジニアを採用して運用をさせる

  - ジェネラリストであることが多い
    - 技術の深さより技術の幅を優先で学びたい
  - ソフトウェアエンジニアリングの経験
  - 顧客からの機能リクエストへの対応経験
  - 懐疑主義であることを重視
    - 慎重である

SREの成長を加速する方法

28章より

SREの教育方法

表28-1　SREの教育の方法

推奨されるパターン	アンチパターン
具体的で順序立てられた学習体験を設計する	訓練として下働きのような作業（アラートやチケットのトリアージ）で手一杯にさせる。厳しい試練を与えて試す
リバースエンジニアリング、統計的な思考、基本原則に基づく作業	運用手順、チェックリスト、作業手順書に厳密に従う訓練
ポストモーテムを読むことをすすめ、障害の分析を推奨する	非難を避けるためにサービス障害を秘密として隠す
限定的ではあるがリアルな障害を人工的に発生させ、学習者に本物のモニタリングやツールを使って修復させる	オンコールにすでに入った後に何かを修復する初めての機会が生じる
仮説的な障害のロールプレイングをグループで行い、複数の問題解決のアプローチをチームで試してみる	チーム内で技術や知識が細分化されたエキスパートを作り出す
オンコールローテーションにシャドウとして早い段階で入れるようにして、自分のノートをメインのオンコール担当のノートと比較する	サービスを全体として理解する前にメインのオンコール担当に入れてしまう
SREのエキスパートとペアを組み、オンコールのトレーニング計画の目的のセクションを見直す	オンコールのトレーニング計画を分野ごとのエキスパート以外は変更できない決まり切ったものとして扱う
単純ではないプロジェクトの作業を切り出して渡し、サービススタック中の一部を受け持つ機会を与える	プロジェクトの新しい作業はほぼすべてシニアSREに渡し、ジュニアSREには断片的な作業だけを渡す

https://www.oreilly.co.jp/books/9784873117911/

SREの教育方法

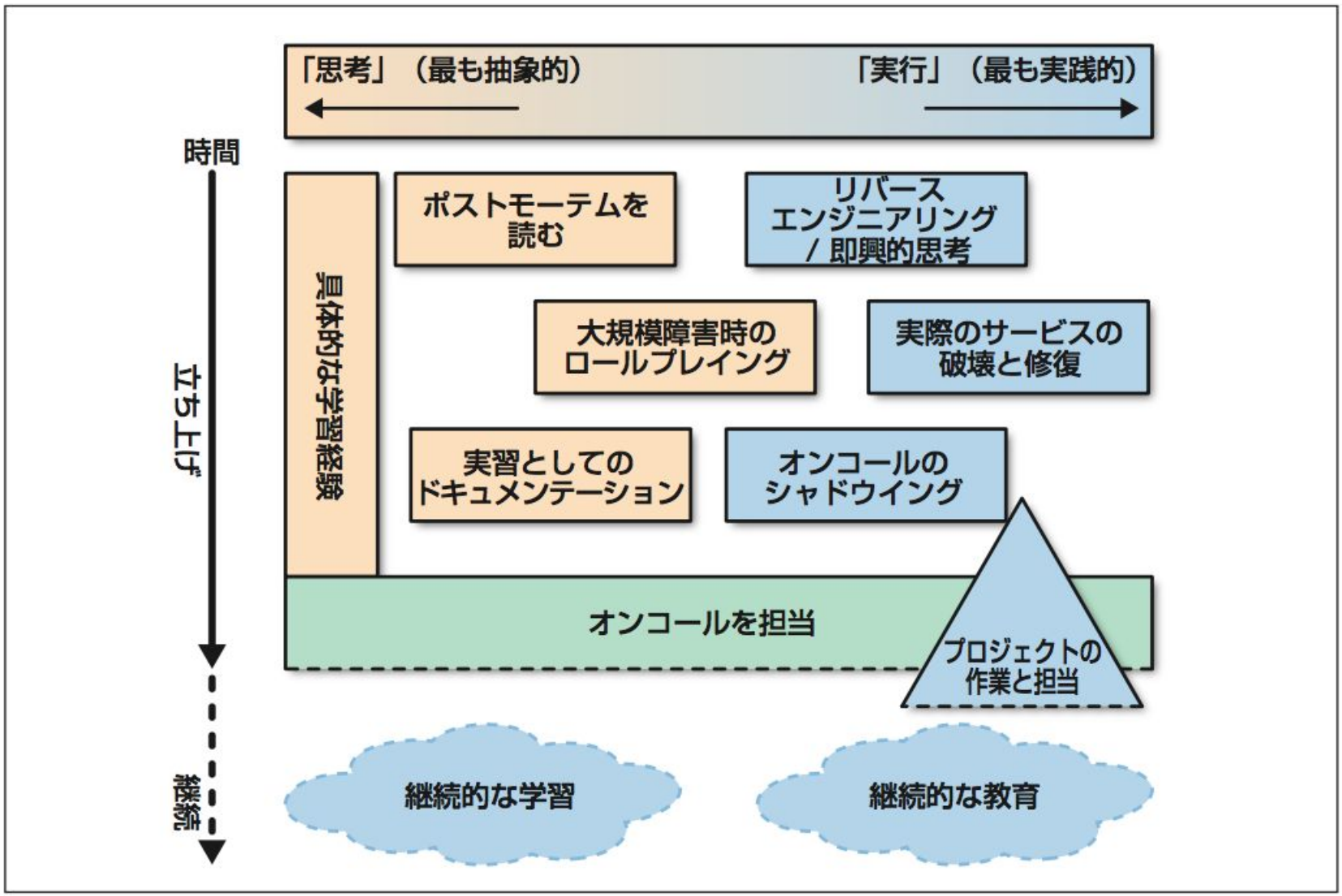


図28-1　オンコールとその先へのSREの立ち上げの設計図

https://www.oreilly.co.jp/books/9784873117911/

まとめ

SREとは。

- 可用性
- レイテンシ
- パフォーマンス
- 効率性
- 変更管理
- モニタリング
- 緊急対応
- キャパシティプランニング

SREの活動は大まかに次のように分類される
- ソフトウェアエンジニアリング
- システムエンジニアリング
- トイル
- オーバーヘッド

ポストモーテムのような文化
マネジメント層の理解・参加

資料一覧

資料一覧

- SRE Book
  https://landing.google.com/sre/book.html
- 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
  https://www.oreilly.co.jp/books/9784873117911/
- Wikipedia “Site reliability engineering“
  https://en.wikipedia.org/wiki/Site_reliability_engineering
- The Factors That Impact Availability, Visualized
  https://www.vividcortex.com/blog/the-factors-that-impact-availability-visualized

参考資料

- hbstudy #74「SRE大全: 序章」
  資料: https://www.slideshare.net/dragan10/hb-study-74-site-reliability-engineering
  動画: https://www.youtube.com/watch?v=P1CXOb6VtO
- hbstudy #75「SRE大全: メルカリ編」
  資料: http://tech.mercari.com/entry/2017/08/21/120000
  動画: https://www.youtube.com/watch?v=SPFFjCQwbek
- hbstudy #76「SRE大全: XFLAG スタジオ編」
  資料: https://xflag.com/blog/hbstudy76_sre_xflag.html
  動画: https://www.youtube.com/watch?v=sAspG6s6NCU
- hbstudy #79「SRE大全: クックパッド編」
  資料: https://speakerdeck.com/rrreeeyyy/sre-at-cookpad
  資料: http://sssslide.com/speakerdeck.com/rrreeeyyy/introduction-to-prometheus-practice
  動画: https://www.youtube.com/watch?v=G7_gkpwzNiY