SRE の基本と組織への導入

川端 俊司

Google Cloud
ストラテジック クラウド エンジニア

製品ライフサイクルにおける摩擦

大規模障害のあとに何が起きるのか

ある担当者が本番環境に変更を加えたところ、想定外の相互作用を引き起こし、Google の主要なサービスが約 4 分間、
停止してしまいました。

そのあと、この担当者に何が起きたと思いますか？

大規模障害のあとに何が起きるのか

ある担当者が本番環境に変更を加えたところ、想定外の相互作用を引き起こし、Google の主要なサービスが約 4 分間、停止してしまいました。

そのあと、この担当者に何が起きたと思いますか？

ボーナスが与えられた

どちらがユーザーを幸せにするのか

SRE の基本

01

Google Cloud

“SRE とは、ソフトウェア エンジニアに運用の設計を依頼すると起きることです。”

ベンジャミン・トレイナー・スロス
Vice President of 24x7 Engineering, Google

原則 1

いかなるシステムにおいて
最も重要な**機能**は
信頼性である。

原則 2

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。

原則 3

巧妙に設計されたソフトウェアは 99.9 %

巧妙に設計された運用は 99.99 %

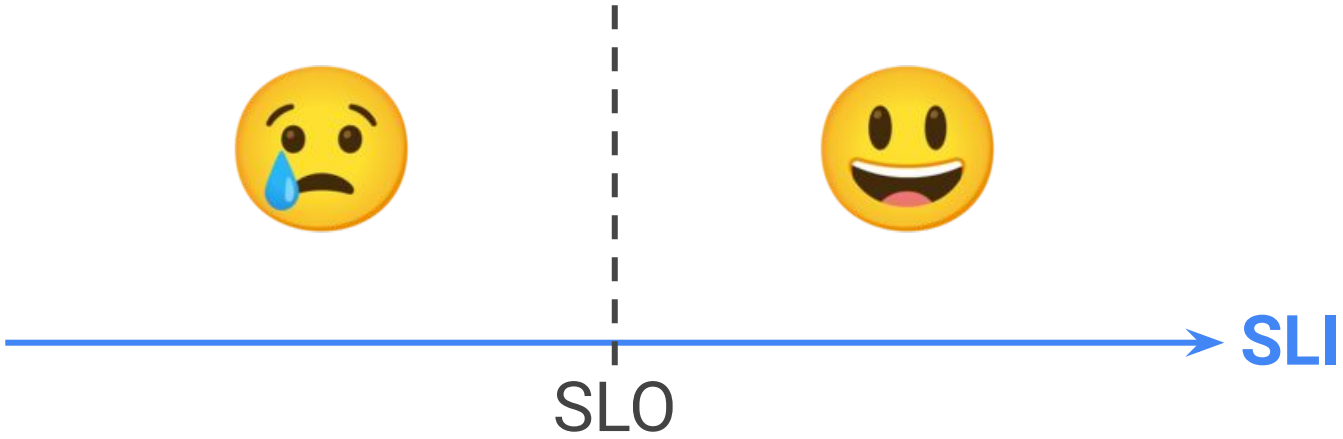
巧妙に設計されたビジネスは 99.999 %

“可用性 100% を信頼性の目標にすることは間違いである。”

ベンジャミン・トレイナー・スロス
Vice President of 24x7 Engineering, Google

信頼性とは何か？

ユーザーの期待に応えているから、ユーザーは満足するのです。



クリティカル ユーザー ジャーニー (CUJ)

ユーザーは
目的を達成するために
サービスとやりとりします

サービスレベル指標 (SLI)

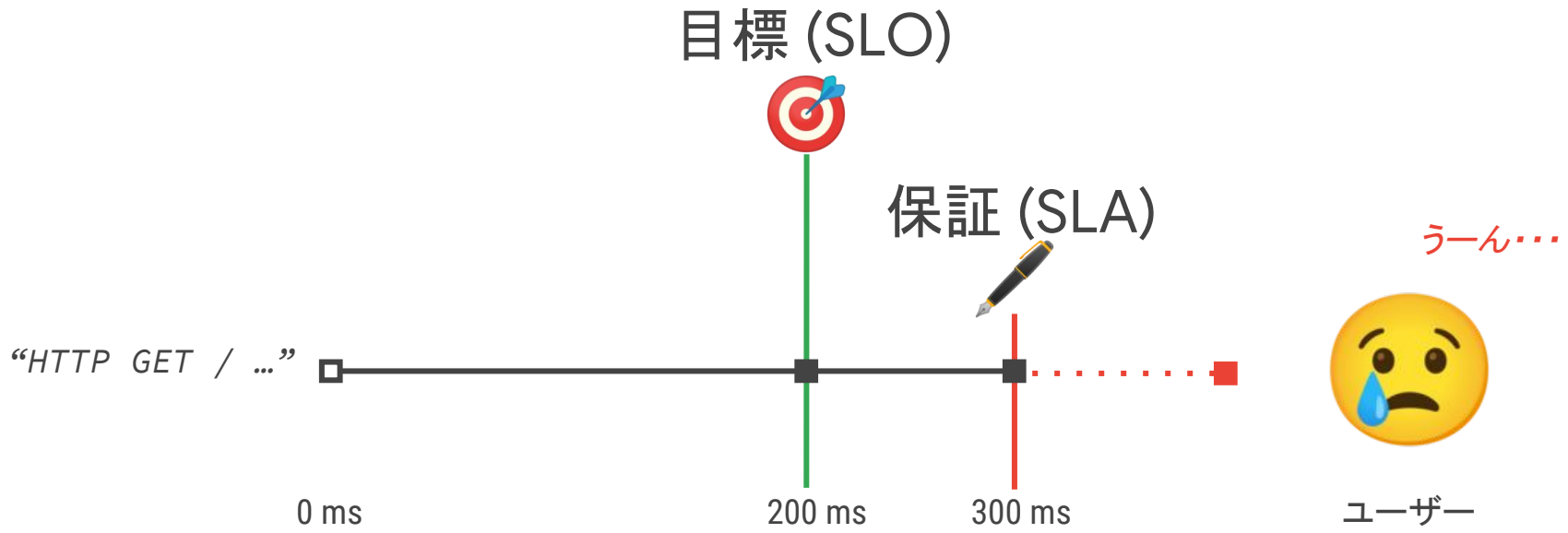
$$\left(\frac{\text{(CUJ を達成した) 良い イベント}}{\text{(CUJ に該当する) 有効な イベント}} \right) \times 100\ [\%]$$

サービスレベル指標 (SLI)

SLI Menu

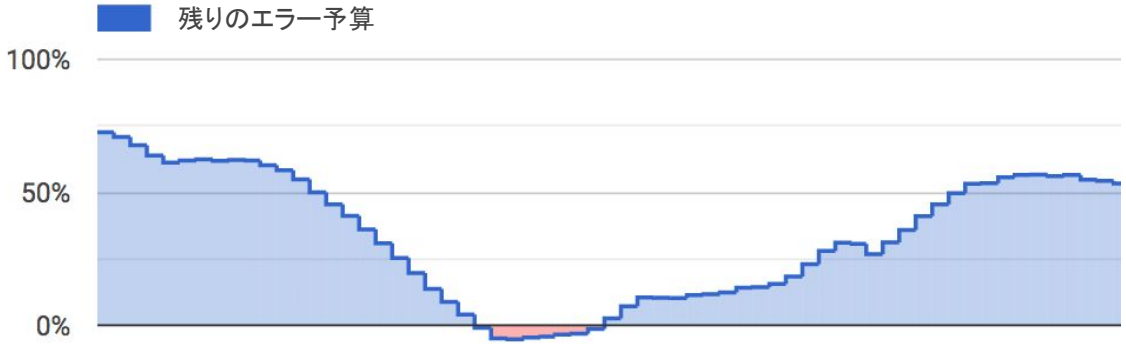
リクエスト / レスポンス	可用性 遅延 品質
データ処理	カバレッジ 正確性 鮮度 スループット
ストレージ	スループット 遅延

SLO と SLA



エラー予算 = 1 - SLO

私たちが達成しようとしている信頼性は、
実際には**許容できる不具合の量**のことなのです。



エラー予算 = 1 - SLO

信頼性のレベル	許容可能な不具合の期間		
	年間	四半期間	30 日間
90%	36.5 日	9 日	3 日
95%	18.25 日	4.5 日	1.5 日
99%	3.65 日	21.6 時間	7.2 時間
99.5%	1.83 日	10.8 時間	3.6 時間
99.9%	8.76 時間	2.16 時間	43.2 分
99.95%	4.38 時間	1.08 時間	21.6 分
99.99%	52.6 分	12.96 分	4.32 分
99.999%	5.26 分	1.30 分	25.9 秒

エラー予算 = 1 - SLO

信頼性のレベル	許容可能な不具合の期間		
	年間	四半期間	30 日間
90%	36.5 日	9 日	3 日
95%	18.25 日	4.5 日	1.5 日
99%	3.65 日	21.6 時間	7.2 時間
99.5%	1.83 日	10.8 時間	3.6 時間
99.9%	8.76 時間	2.16 時間	43.2 分
99.95%	4.38 時間	1.08 時間	21.6 分
99.99%	52.6 分	12.96 分	4.32 分
99.999%	5.26 分	1.30 分	25.9 秒

エラー率	許容可能期間
100%	21.6 分
10%	3.6 時間
1%	36 時間
0.1%	15 日
<0.05%	ずっと

エラー予算の使い方

- SLO を定義する
- データに基づいて判断する
- 予算が残っている限りは、通常の DevOps のリズムは崩さない
- 予算をスプリントと合わせる

リスク

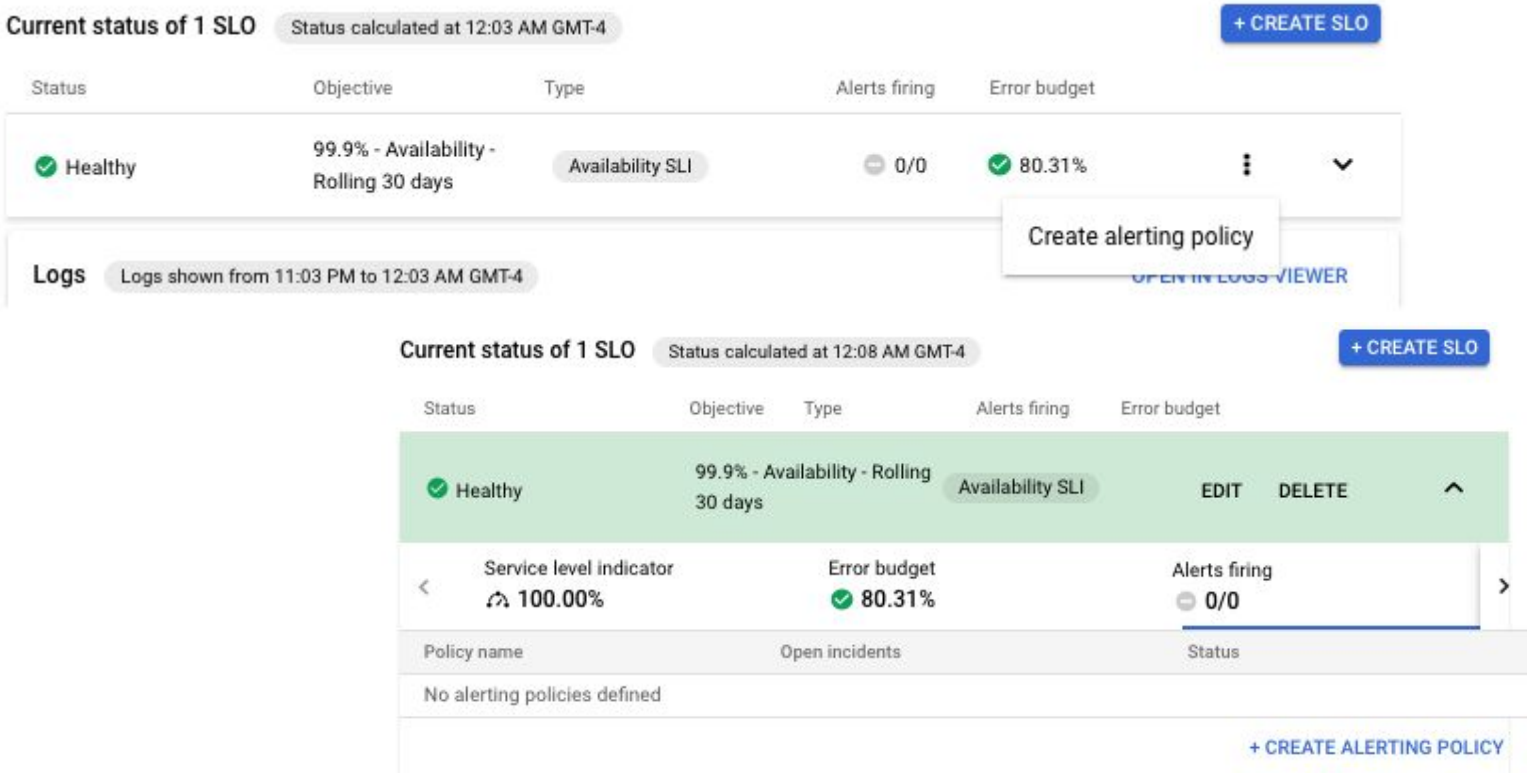
人間の脳は、リスクの比較と評価において、信頼できないということが
知られています。

我々は、記憶に残る特異な出来事には過大に反応し、偶発的な事故や
抽象的な出来事、自然現象には、実際よりも過小に反応します。

実際に、我々はインフルエンザよりも炭疽菌のほうを心配しています。
しかし、年間死亡者数を比較すると、インフルエンザは 25 ～ 50 万人、
炭疽菌はほぼゼロなのです。

ダニエル・ギルバート
Edgar Pierce Professor of Psychology, Harvard University

エラー予算の使い方



エラー予算を使うことによるメリット

- **開発と SRE の共有のインセンティブになる**
  イノベーションと信頼性の適切なバランスを見つけられる
- **開発チームが自分でリスクを管理できる**
  エラー予算をどう使うかは開発チームが決められる
- **非現実的な信頼性の目標は魅力的でなくなる**
  そういった目標はイノベーションの速度を遅くすることがわかる
- **システム稼働時間に対する責任を共有する**
  インフラの障害も開発のエラー予算を消費する

リスク分析

	影響			
可能性		壊滅的な被害	中程度の被害	ほぼ被害なし
	頻繁に発生			
	時々発生			
	ほぼ 発生しない			

リスク分析

	影響			
可能性		壊滅的な被害	中程度の被害	ほぼ被害なし
	頻繁に発生			
	時々発生			
	発生しない			

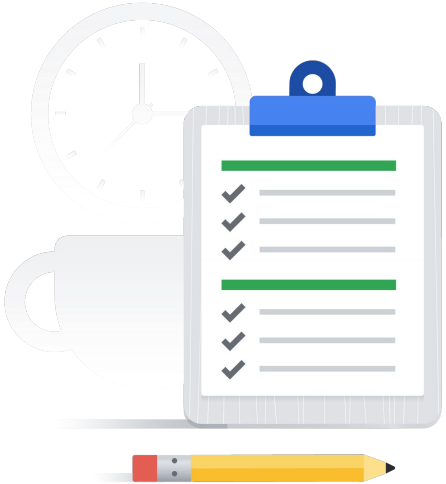
リスク分析

SLO に危険をもたらす可能性のあるリスクだけを特定します。
(ETTD + ETTR) * (% impact) / (ETTF / 365.25)

例: 設定ミスでリソース減少、過負荷でリクエストを取りこぼす
=> (30 分 + 120 分) * 20 % / (90 日 / 365.25) ≒ 122 分/年

SLO の作り方

1. クリティカル ユーザー ジャーニー を検討する
2. 候補となる SLI を選ぶ
3. SLI の目標、SLO を仮決めする
4. リスク分析を行って、リスクを定量化する
5. SLO を検証し、リスクに対処して達成できるようにする
6. エラー予算の使い方を検討して、実行する
7. SLO を定期的にレビューする



SRE の組織への導入

02

Google Cloud

SRE の実践分野

指標と監視

- 監視と警告
- SLO に深刻な脅威がある場合に人間を巻き込む
- トリガーとアクション

キャパシティ計画

- 有機的な成長
- 非有機的な成長
  - セール
  - 新機能追加
- 余剰

変更管理

- ゆっくり変更
- 効率の良いロールバック
- 約 70% の障害は変更を加えることで発生すると心得る

緊急対応

- 明確な「インシデント」の基準
- 事前定義済みの RACI (業務責任)
- 手順書とその他のドキュメント

文化

- 心理的安全性
- 非難しない
- データ駆動 (客観的なデータに基づいた合理的な判断と行動)

class SRE implements DevOps

```
interface DevOps {
  /* 開発、運用、システム構成や組織のサイロを
     解消するためのガイドライン  */

  void 組織の壁を減らす();

  void 失敗を普通のこととして受け入れる();

  void 段階的に変更を加える();

  void ツールと自動化を活用する();

  void すべて測る();
}
```

```
class SRE implements DevOps {
  /* 信頼性と開発の速度の均衡をとりながらユーザー
     を幸せにする工学的な手法、または職務  */

  void 組織の壁を減らす() {
    同じツール、同じ手法を使う  };

  void 失敗を普通のこととして受け入れる() {
    失敗を前提にしてエラー予算を導入する  };

  void 段階的に変更を加える() {
    失敗のコストを下げる  };

  void ツールと自動化を活用する() {
    手作業を自動化して半分以下にする  };

  void すべて測る() {
    信頼性を定量化する  };
}
```

SRE を導入するための段階的なアプローチ

ひとつのサービスに関連するチームから
始めて、必要に応じて繰り返す

他のチームへ横展開

- 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

Google による SRE 導入のご支援

SRE Core

SRE の根幹となる文化と実践について、開発者、運用者、製品責任者を実践的な演習とともに教育し、推進派チームを作ります。SLO を作成し、リスク分析を行って、お客様のアプリケーションのいくつかに導入します。

SRE Teams & Policies

インシデント対応、ポストモーテム、オンコール、トイル管理などの SRE の実践項目に対する Google のアプローチを学びます。お客様の SRE チームの活動基本方針を設計するとともに、SRE が開発者と連携するための関わり方を整理します。

Design & Operations Review

SRE Core 以降にお客様が実施された改善活動を確認し、SLO のリスク分析を行います。SRE 定着のための戦略的および戦術的なロードマップとともに、特定されたすべてのリスクに対する推奨事項を含むレポートを作成します。

SRE Scaling & Upskilling

スキルギャップ分析を行い、既存の担当者の技能向上計画を作成します。新しい SRE のための導入教育計画を確立します。SRE の候補者の人材像を構築し、面接をするための訓練を行います。

次のステップ

01

google.com/sre を訪問

組織が SRE の実践を開始するのに役立つ多くのリソースを利用できます。前述のSRE の本は、オンラインで *無料で* 読むことができます。

02

SLO を導入する

まずは、今のままでもほぼ達成できるようなサービスレベル目標を定義しましょう。そのためには、ビジネス、開発、運用が、お互いに協力しなければなりません。ひとつのアプリケーションを選んで、それに関連するチームから変革を始めることをお勧めします。

03

ポストモーテムを実践する

人を非難しないポストモーテム (ふり返り) の文化を確立することこそが、システムの信頼性を継続的に高めつつ成功する SRE 組織を維持するために重要です。

04

Google と一緒に始める

Google のプロフェッショナル サービスは、これまでに培った豊富な経験を生かし、Google Cloud のお客様が SRE の実践と文化の導入を成功させるためのガイダンスを提供します。

さあ、一緒に SRE の旅を始めましょう！

Thank you.