

SRE

Technology Map

SRE Technology Map 2023

「Site Reliability Engineering (SRE) とは、Google社が提唱、実践しているシステム管理とサービス運用の方法論^[1]」です。

サイバーエージェントの事業成長とともに、SRE領域へも注力してきました。

SRE Technology Mapは、当社のSREチームの取り組みを知ってもらうための冊子です。

[1] 「サイトリライアビリティエンジニアリング」 『ウィキペディア (Wikipedia): フリー百科事典』

サイバーエージェントが提供する幅広い事業サービスの信頼性向上に、私達SREsは日々取り組んでいます。事業領域や事業フェーズ、組織規模が異なれば、SREsのアプローチも違ってきます。

それぞれのSRE組織が、様々な課題解決に取り組んだことによって得られた知見や考え方などを多くの人に知ってもらいたいと考え、「SRE Technology Map」を作成しました。

「SRE Technology Map」を通して、少しでもサイバーエージェントに興味を持っていただければ幸いです。また、私達と同様にSRE推進に取り組んでいる人々の助けや学びになることを願っております。

メディア統括本部
サービスリライアビリティグループ
マネージャー 兼
Developer Expert of SRE

柘植 翔太

Shota Tsuge



専務執行役員
技術担当

長瀬 慶重

Norishige Nagase



2023年3月で当社は25周年を迎えました。

創業来、インターネット産業の拡大とともに事業成長を続けてきたサイバーエージェントでは、2015年頃からお客さまに安心・安全なサービスを提供するためにシステムの維持・向上に向けて、SREに取り組みはじめました。

当初、メディア事業から始まったSREですが、今ではゲーム事業、インターネット広告事業、AI・DX領域でもSREの専門チームが発足するなど、その活動の幅は年々広がっています。

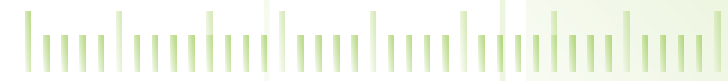
この「SRE Technology Map」では、サイバーエージェントのSREチームが日々向き合っている課題や取り組みについてご紹介しています。

本冊子を通して当グループの知見を展開できることを願っています。

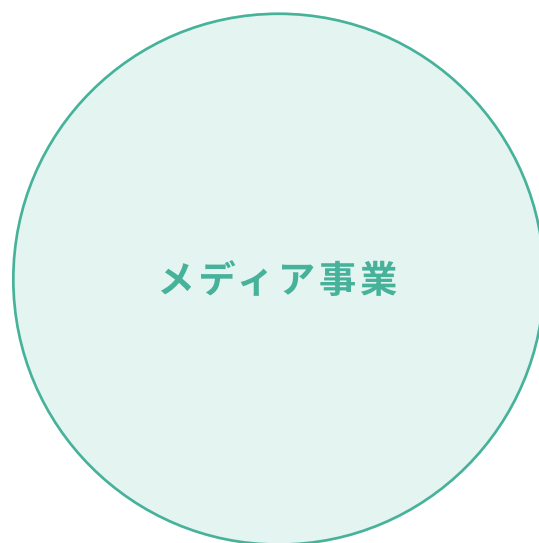
Index

01. Introduction	02	05. 事業部毎のSRE組織情報	29	07. コラム	65
02. サイバーエージェントグループの SRE組織と実装パターン	06	5.1. メディア事業 SRG、ABEMA、IU（CAM、タップル）、CyberZ		7.1. 新卒1年目のSREが考えるSREの魅力とは	
2.1. SRE組織パターン		5.2. ゲーム事業 CA GameFi、アプリボット、サムザップ		7.2. SREに求められるエンジニアとしての素養	
2.2. SRE実装パターン		5.3. AI・DX事業 アドテクディビジョン		08. 編集	72
03. データで見るSRE	11	06. SREsの取り組み	48		
3.1. キャリアと組織の傾向		6.1. Datadog Incident Management使ってAmeba のインシデント管理体制整備した話			
3.2. 採用技術の傾向		6.2. CyberZでのトイル可視化の取り組み			
04. SREs紹介	26	6.3. Embedded SRE としてサービスに SLO を導入 してみる			
		6.4. SRE成熟度評価について			
		6.5. Grafana OnCallの活用			

サイバーエージェントグループの SRE組織と実装パターン



メディア事業、インターネット広告事業、ゲーム事業を中心に事業を展開。
各事業に、SRE組織も立て付けられており、SRE推進に取り組んでいます。



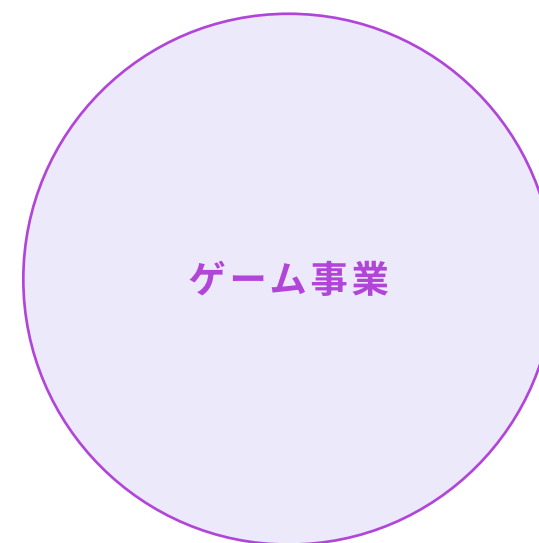
メディア事業

- 「ABEMA」
- 「Ameba」
- 「タップル」 等



インターネット
広告事業

- AI 事業
- DX 事業
- AI 技術の研究・開発 等



ゲーム事業

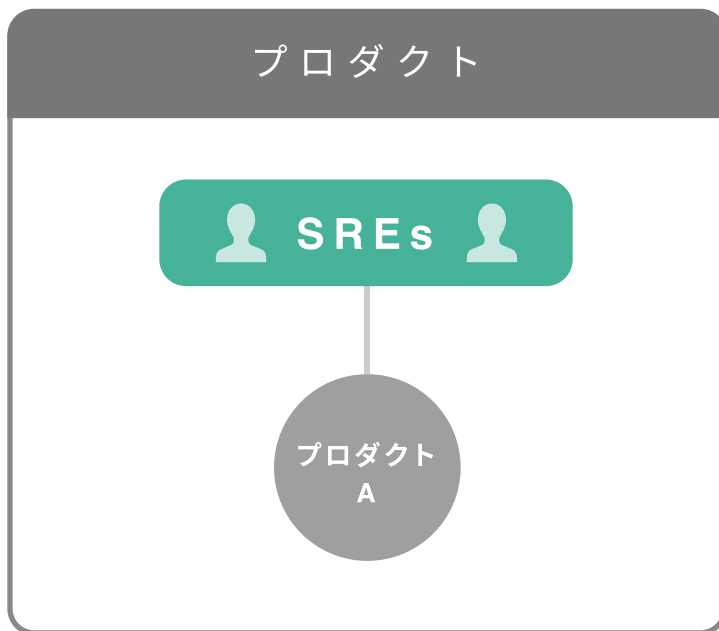
- 「グランブルーファンタジー」
- 「ウマ娘 プリティーダービー」
- 「戦国炎舞 -KIZNA-」 等



08
01
02
03
04
05
06
07
08

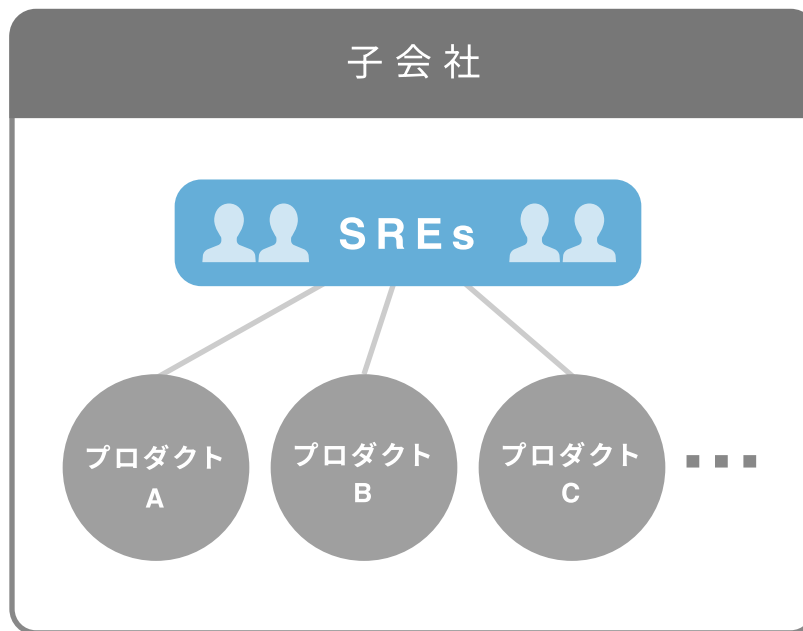
サイバーエージェントグループのSRE組織パターン

① 単一プロダクト専任SREs



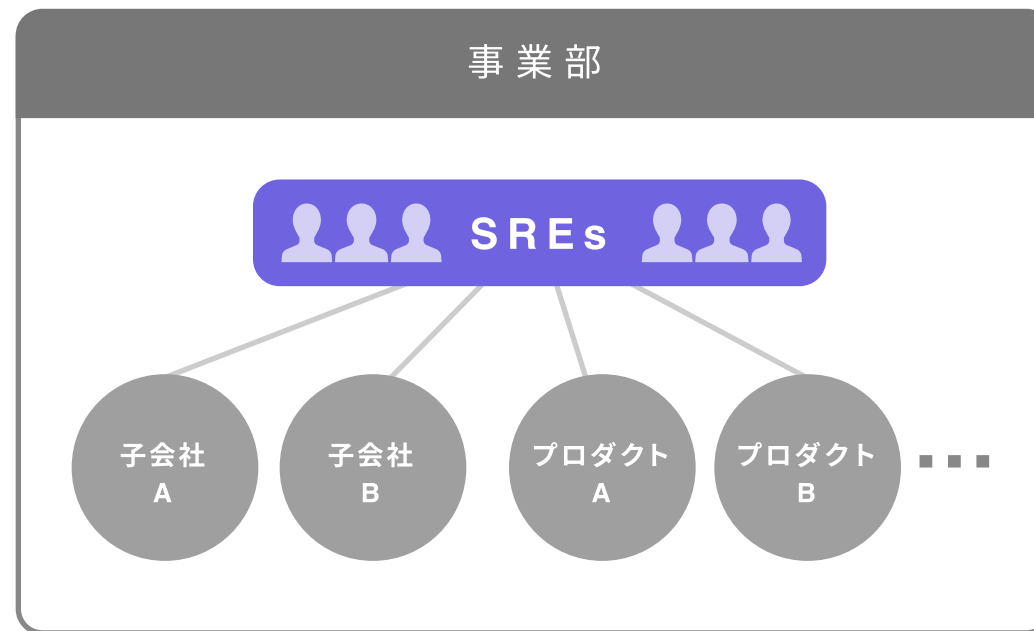
SRE組織とプロダクトは、1対1の関係

② 子会社専任SREs



SRE組織とプロダクトは、1対Nの関係

③ 事業部横断SREs



SRE組織とプロダクトは1対Nの関係だが、
複数の子会社を跨いでSRE推進を行う

サイバーエージェントグループのSRE実装パターン

Product SRE

各プロダクトの開発チームに所属し、信頼性向上へ取り組むSREs

Embedded SRE

Enablementフェーズ：SREを組織へ浸透し、自立支援に取り組むSREs
Evangelistフェーズ：SREの強化や改善を行うSREs

Platform SRE

共通基盤の信頼性や開発生産性向上へ取り組むSREs。

SRE Center of Practice

横断的に展開できるSREプラクティス開発や導入へ取り組むSREs

Movable Embedded SRE

緊急かつ高重要度の案件毎にチーム編成し、案件完遂に取り組むSREs

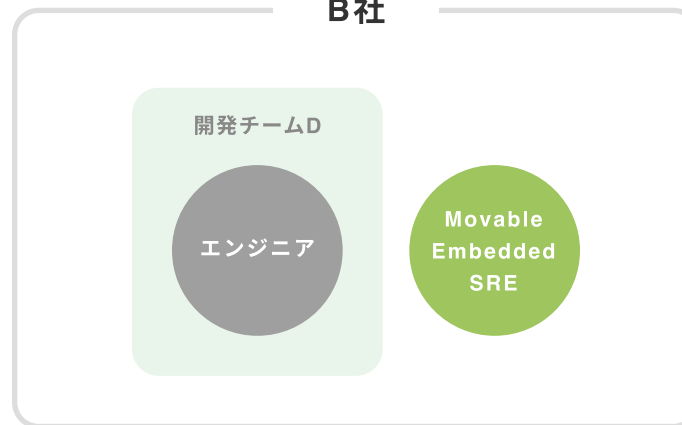
SRE組織俯瞰

A事業

A社



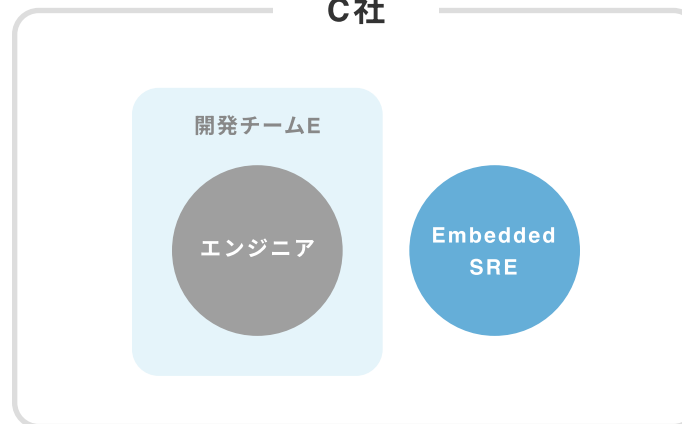
B社



Service C



C社



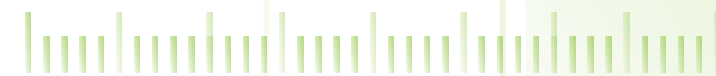
共通基盤



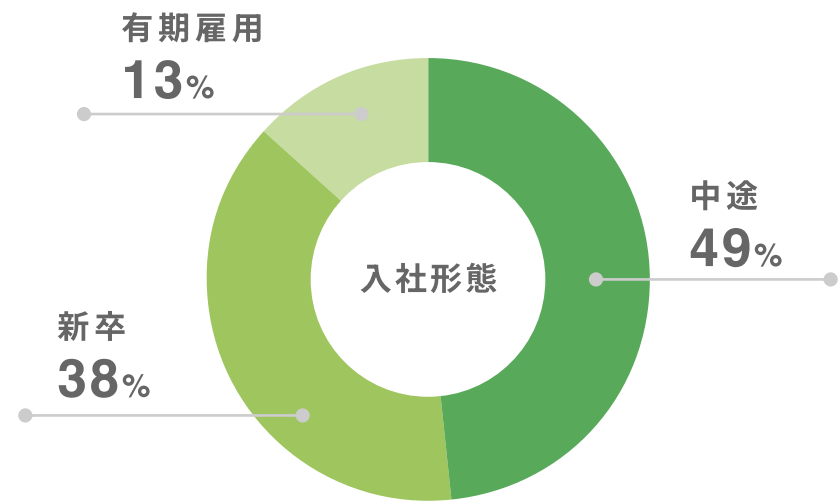
SREチーム



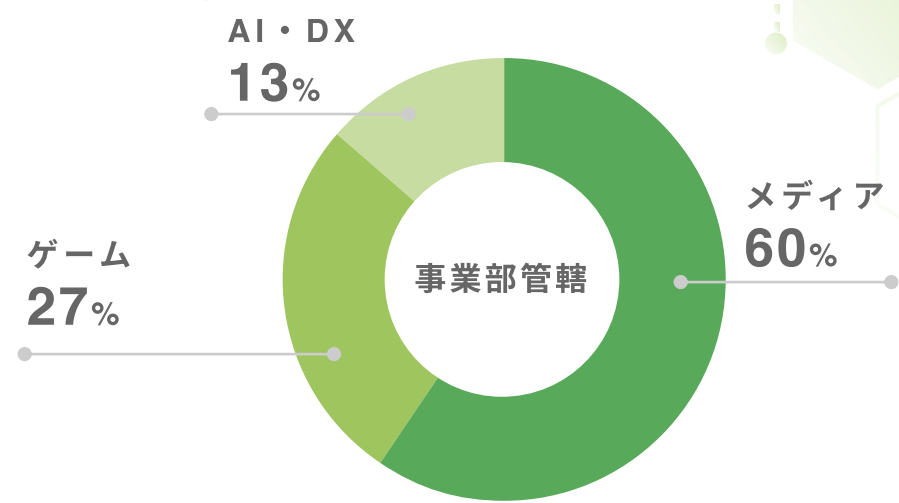
データで見るSRE キャリアと組織の傾向



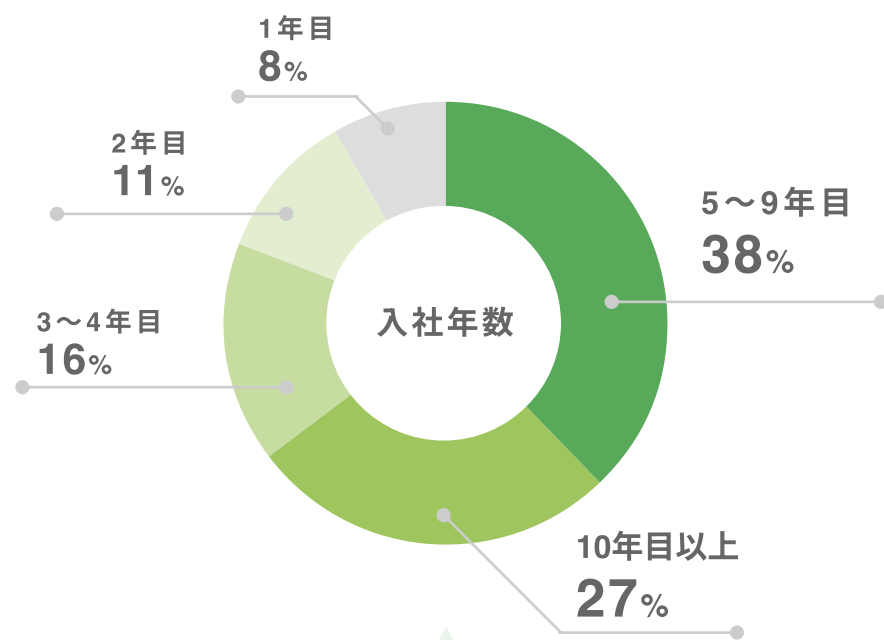
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08



新卒と中途の割合は、おおよそ半々
多くの新卒SREsが活躍している

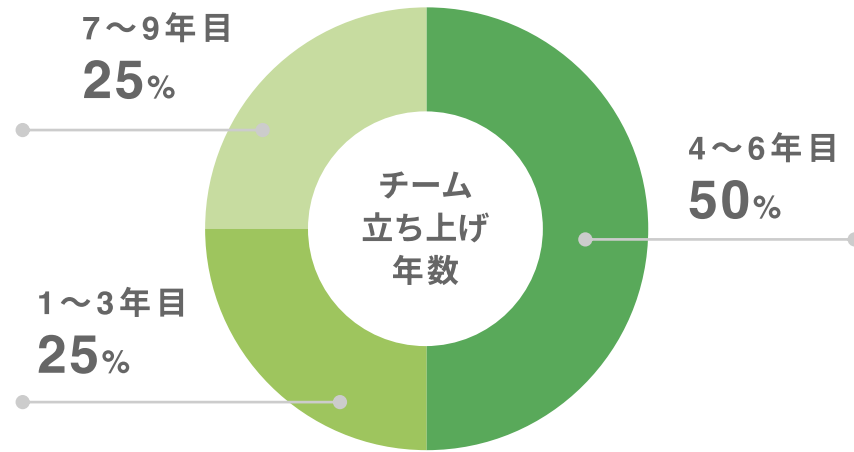
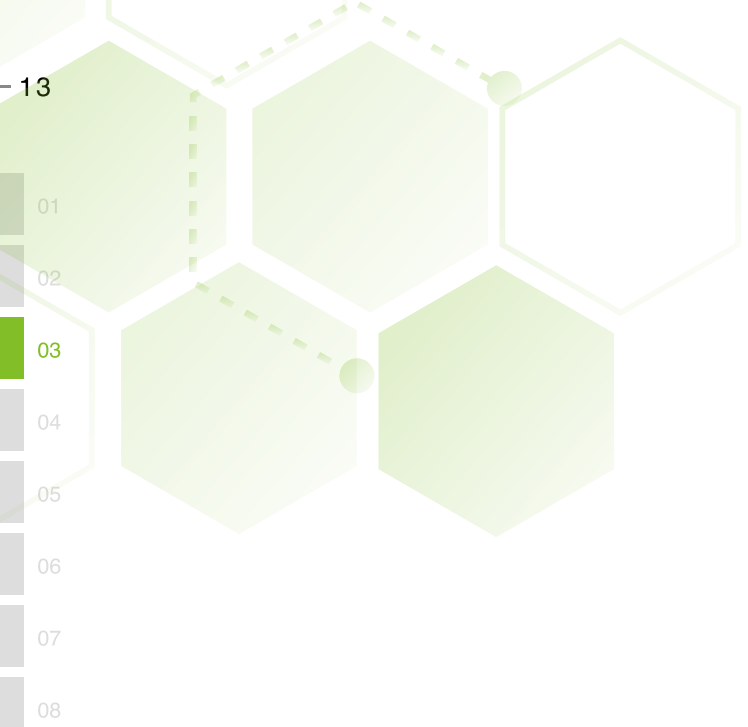


現在は、メディアの割合が多いが、
ゲームやAIでの需要も増えている

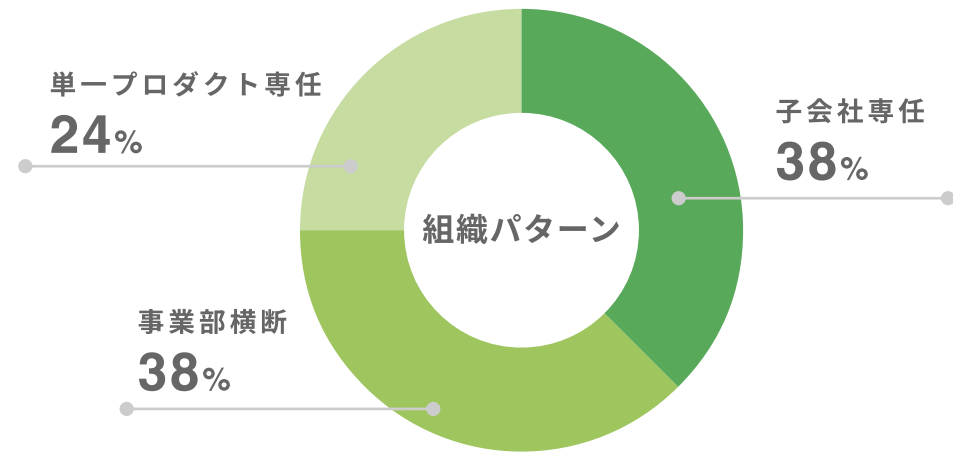


入社5年以上が、6割を超えている

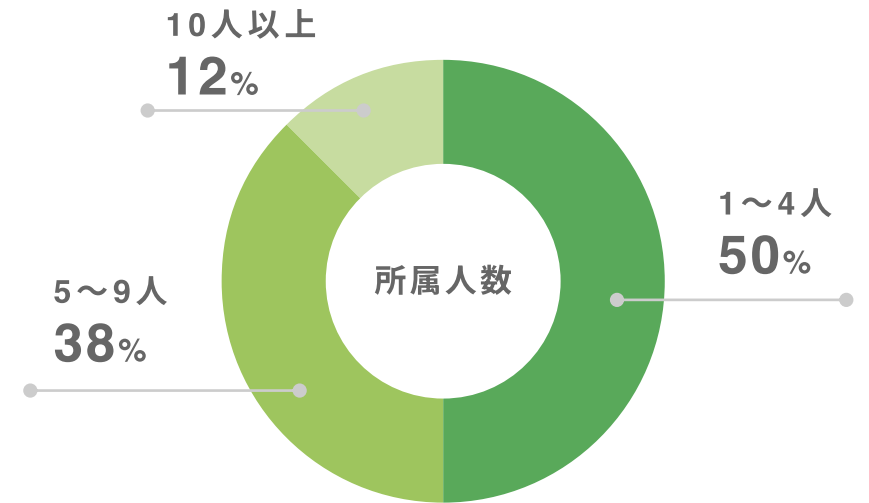




立ち上げて4年以上のSRE組織が、7割を超えている



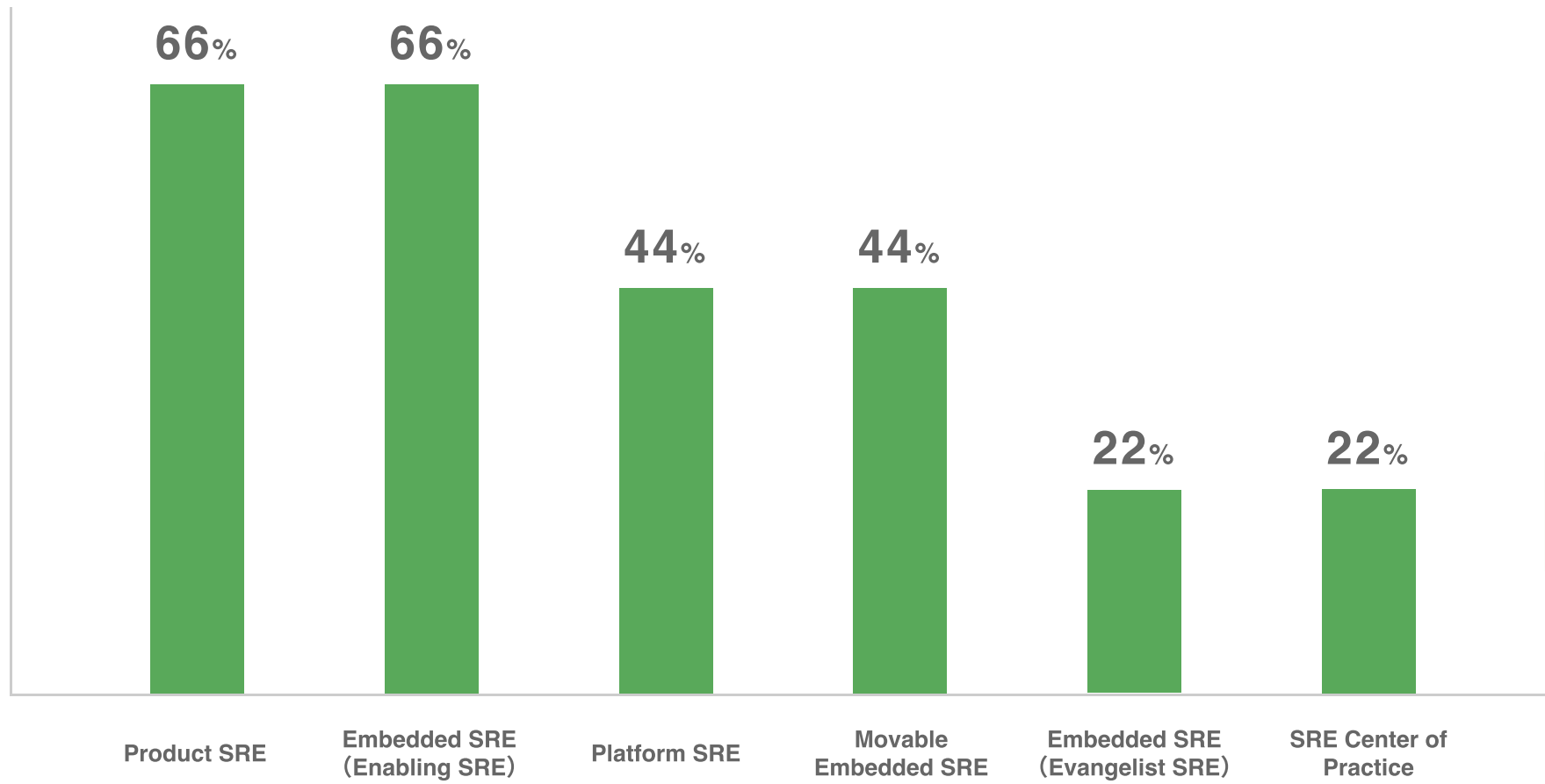
8割のSRE組織が、複数プロダクトのSRE推進に取り組んでいる



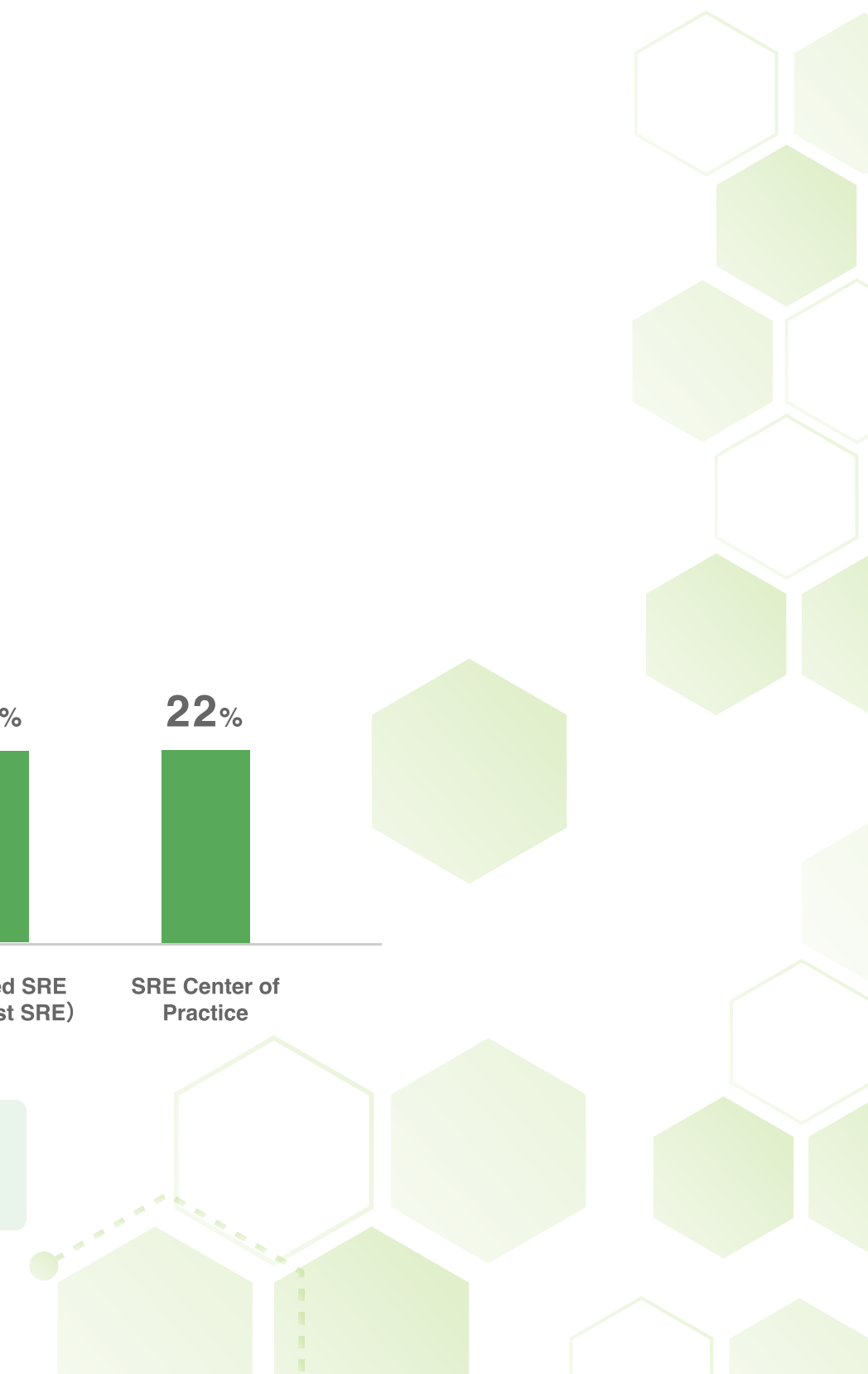
5割のSRE組織が、4人以下

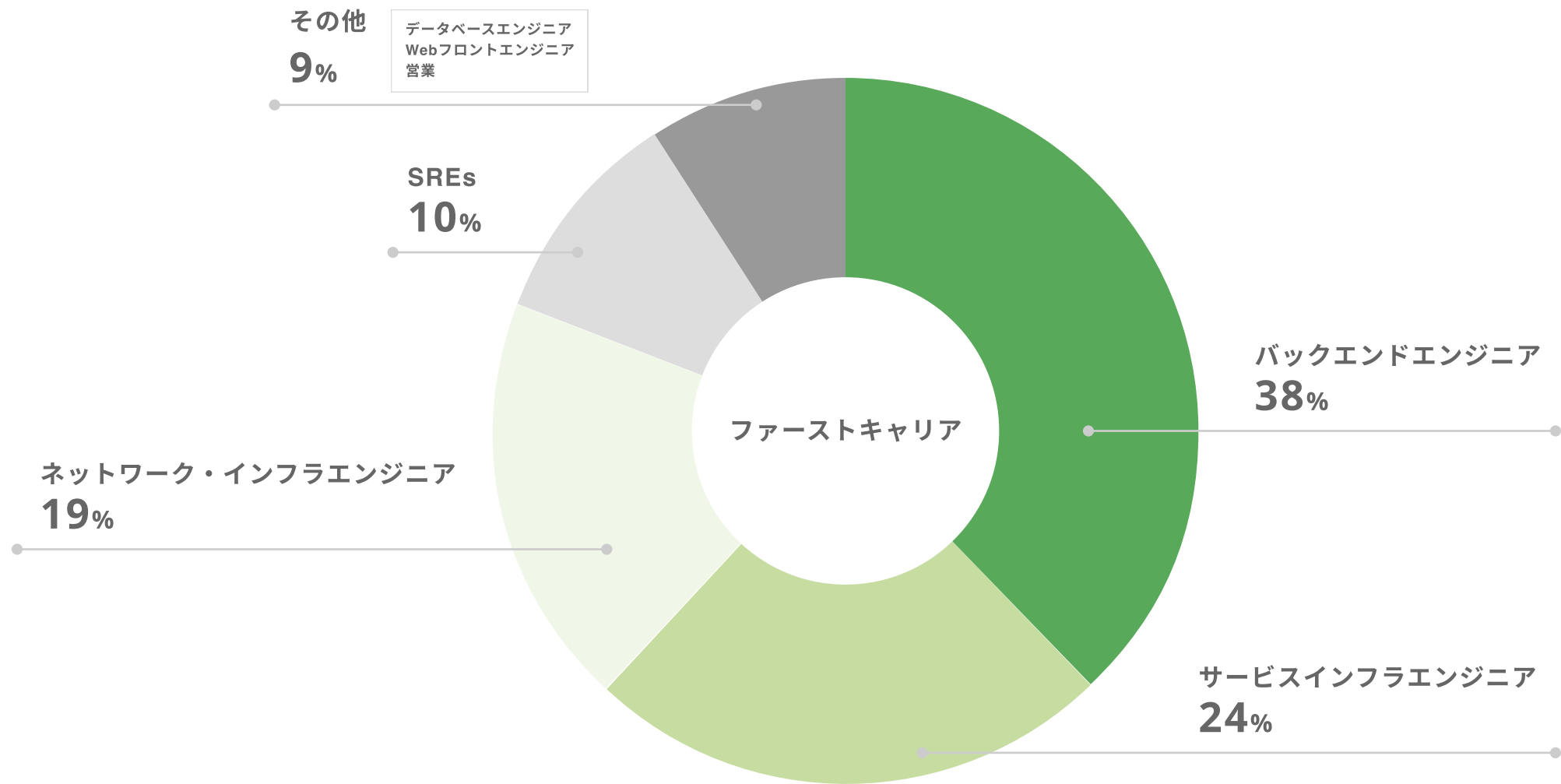
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08

SRE実装パターン



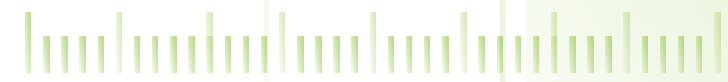
Product SREとEnabling SREが多い傾向
全ての事業部横断SREsが、SRE Center of Practiceを実装している





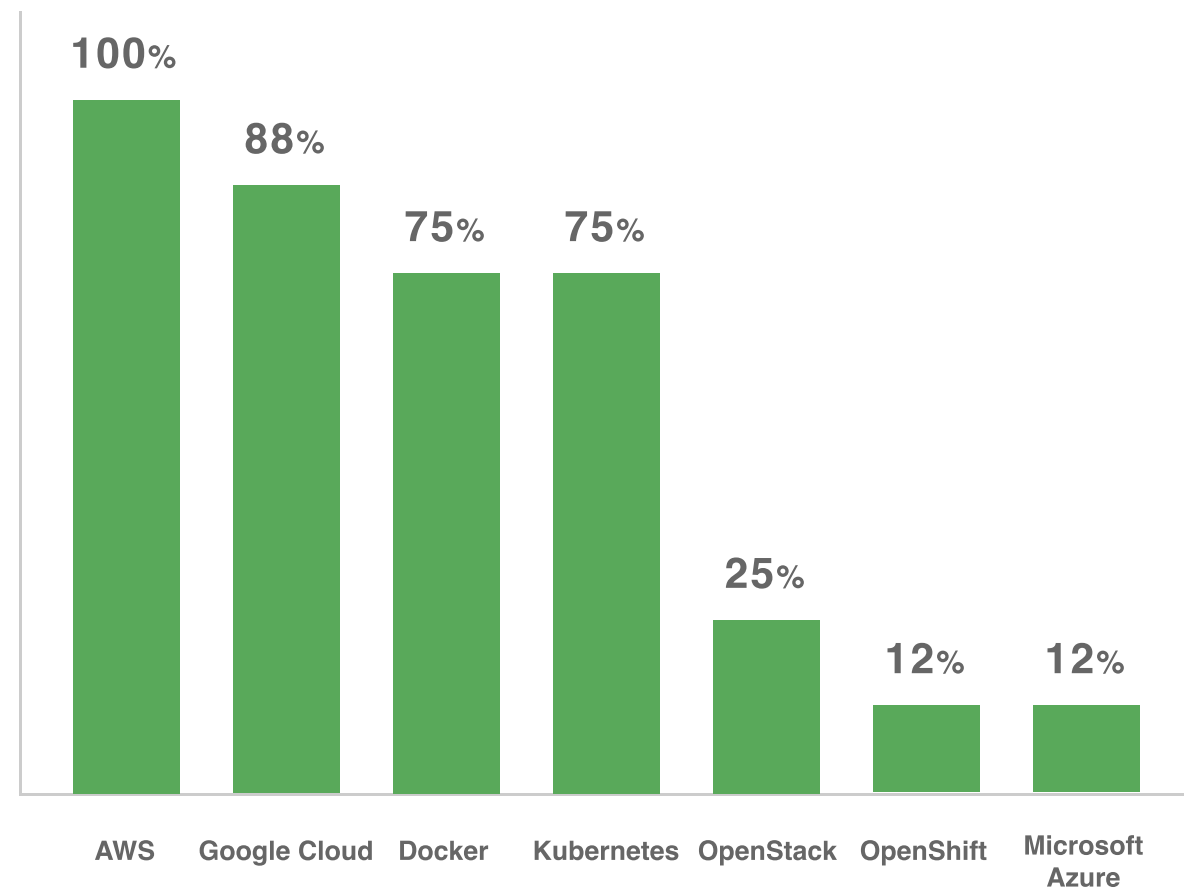
ファーストキャリアの8割が、バックエンドやインフラエンジニア
新卒SREsの割合も増えてきている

採用技術の傾向



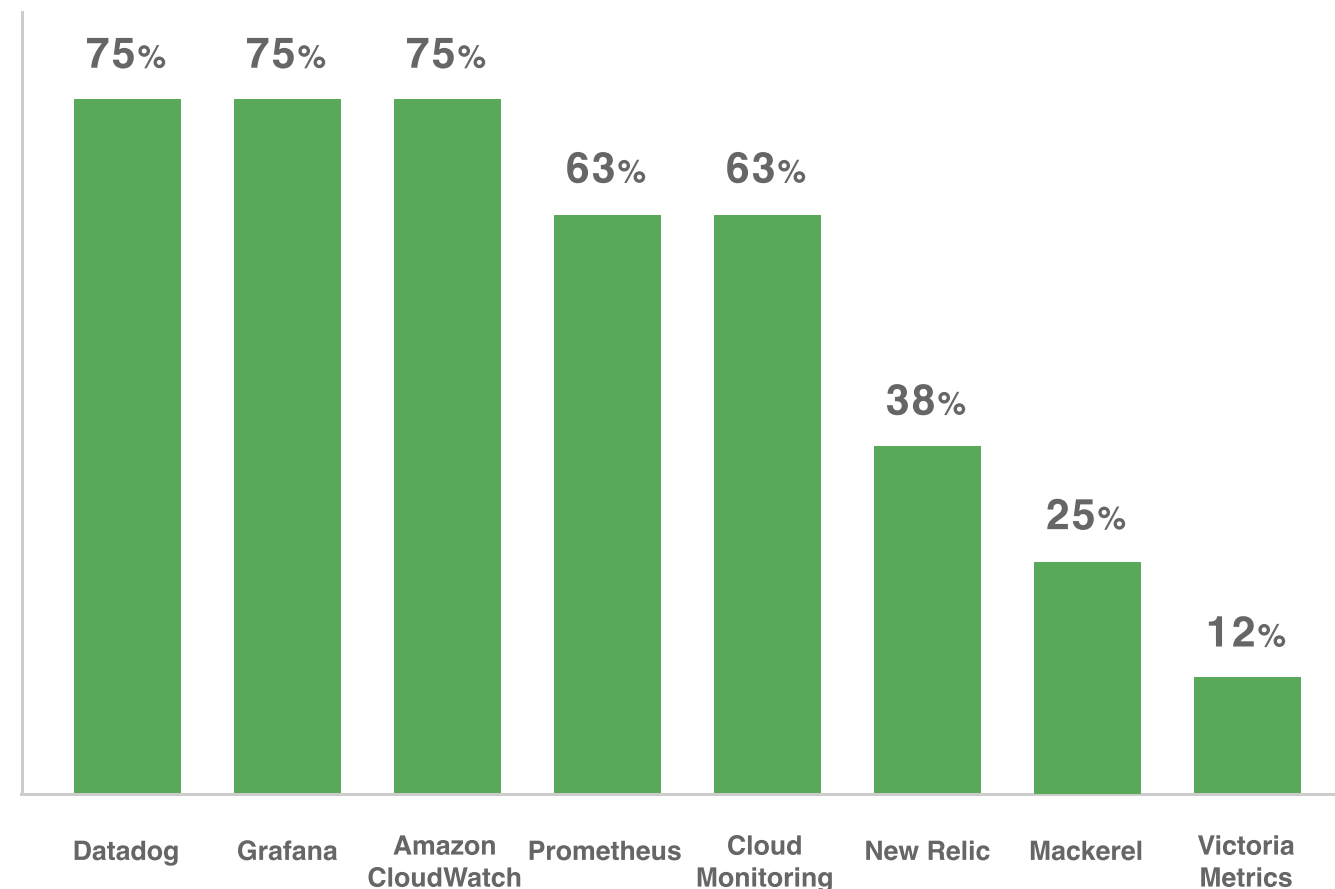
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08

1.プラットフォーム

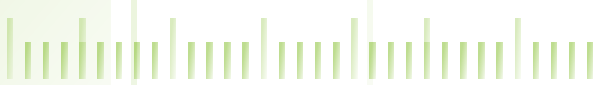


全SRE組織が、AWSを活用している
また、7割以上のSRE組織が、Kubernetes (k8s) を活用している

2.監視

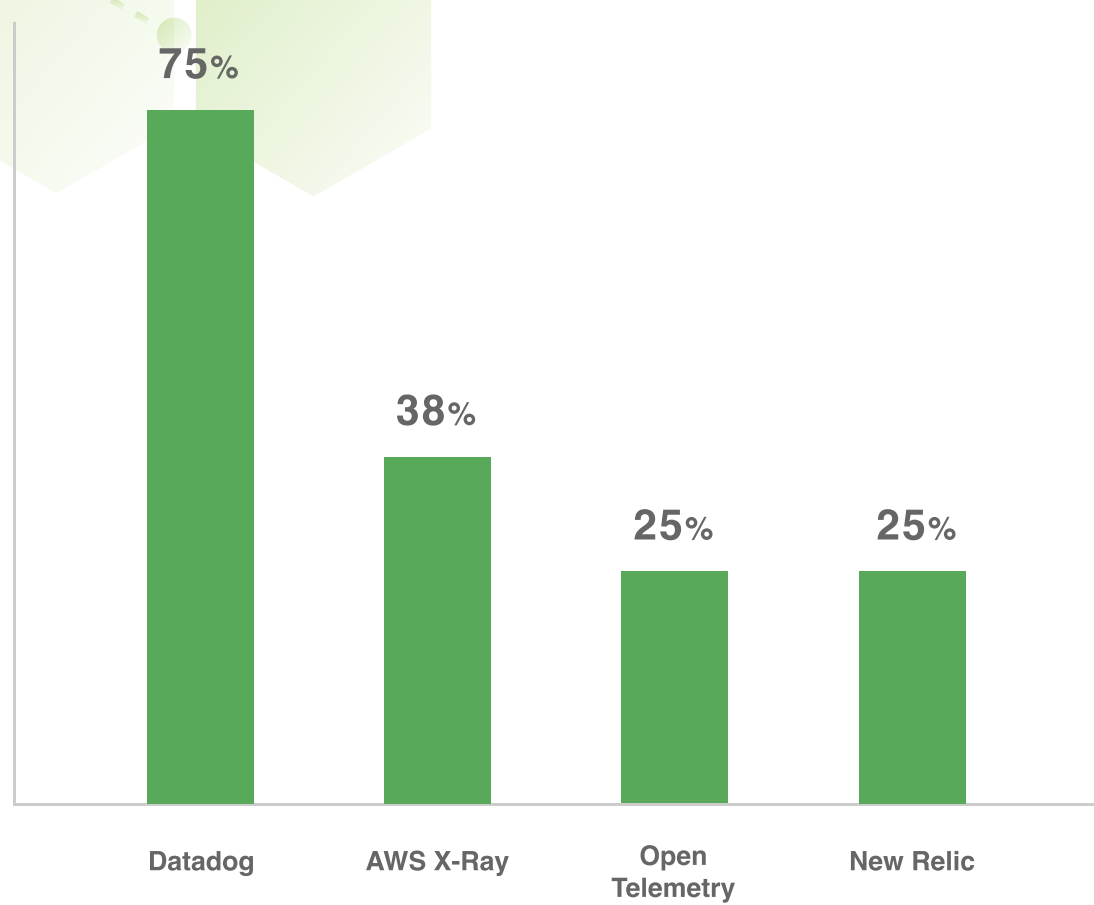


Datadogは、SLO監視や可観測性ツールとしての活用も多い傾向がみられた



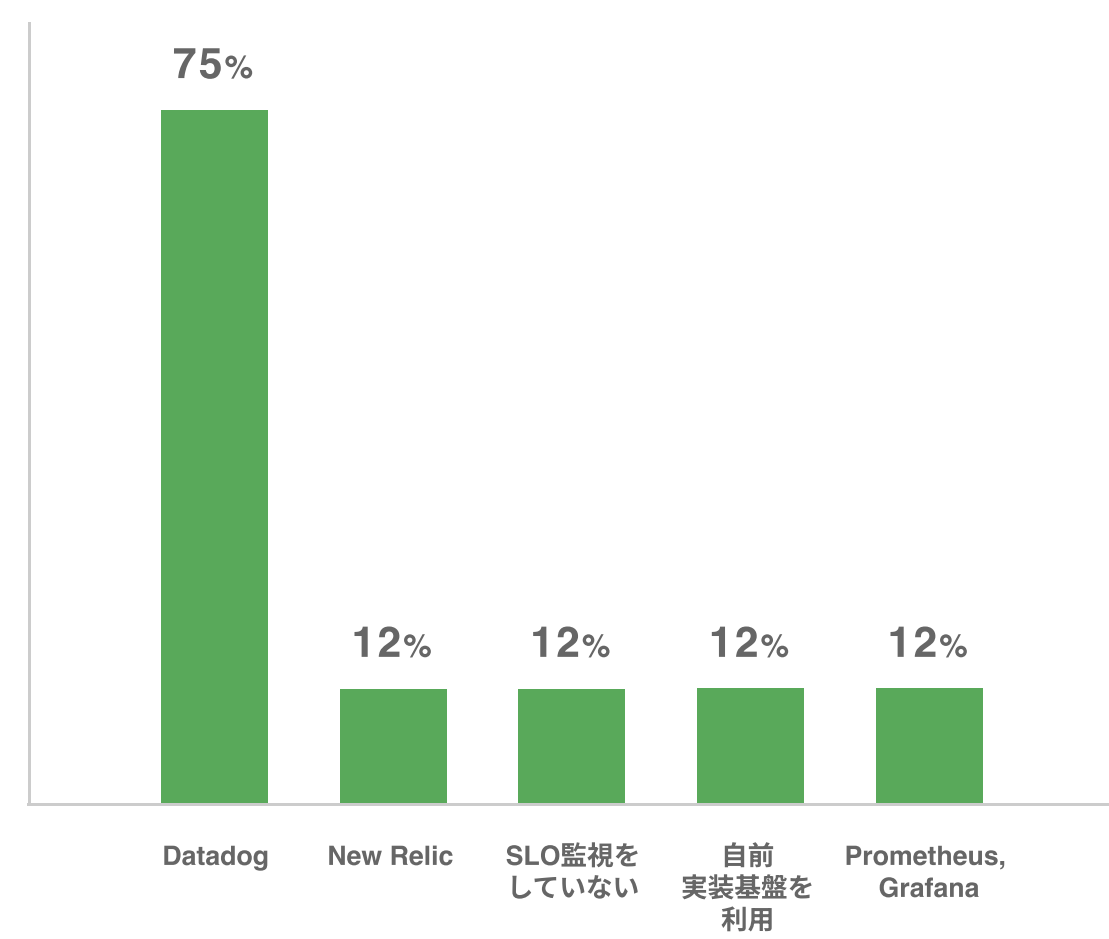
18
01
02
03
04
05
06
07
08

3.可観測性



7割以上のSRE組織が、Datadogを活用している
監視と同じソリューションを活用する傾向がみえた

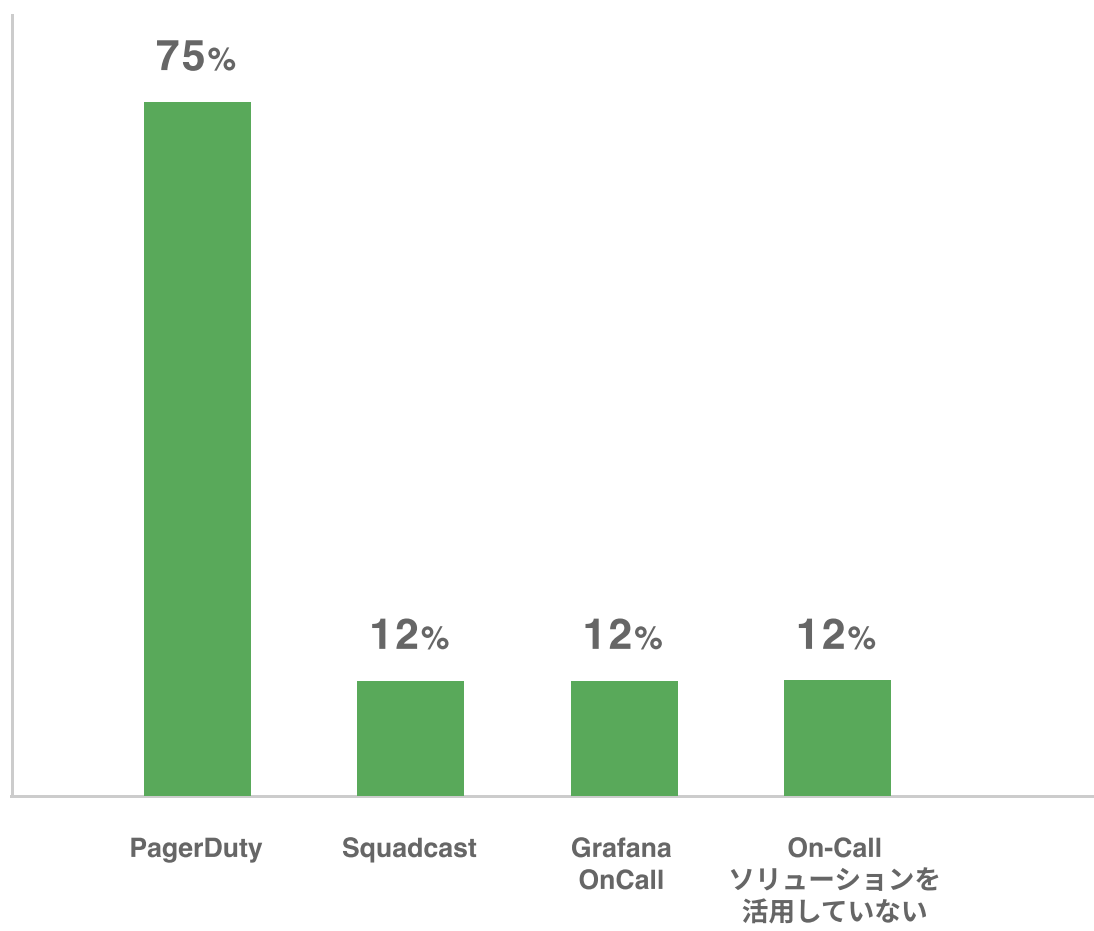
4.SLO監視



可観測性と同じソリューションを活用する傾向がみえた

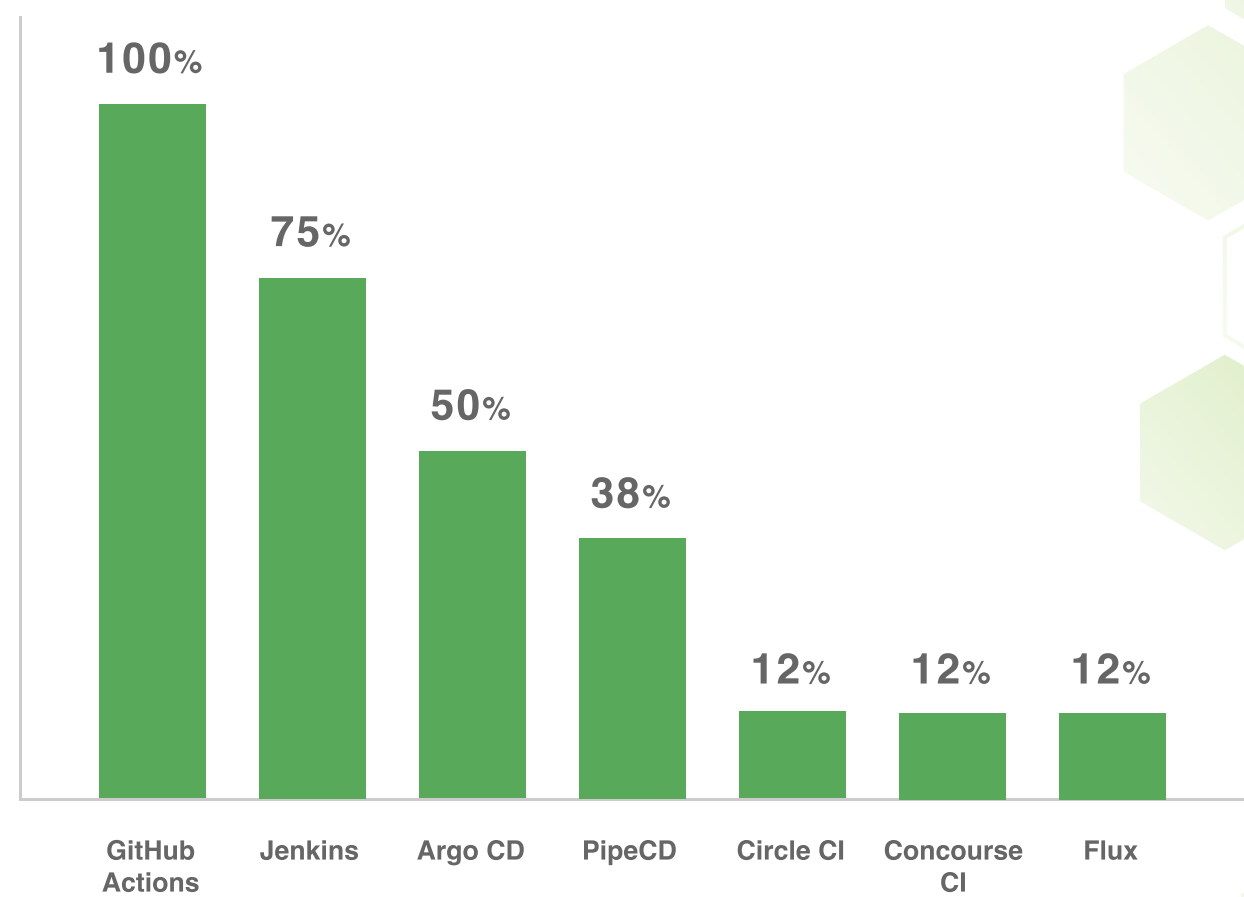
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08

5. On-Call

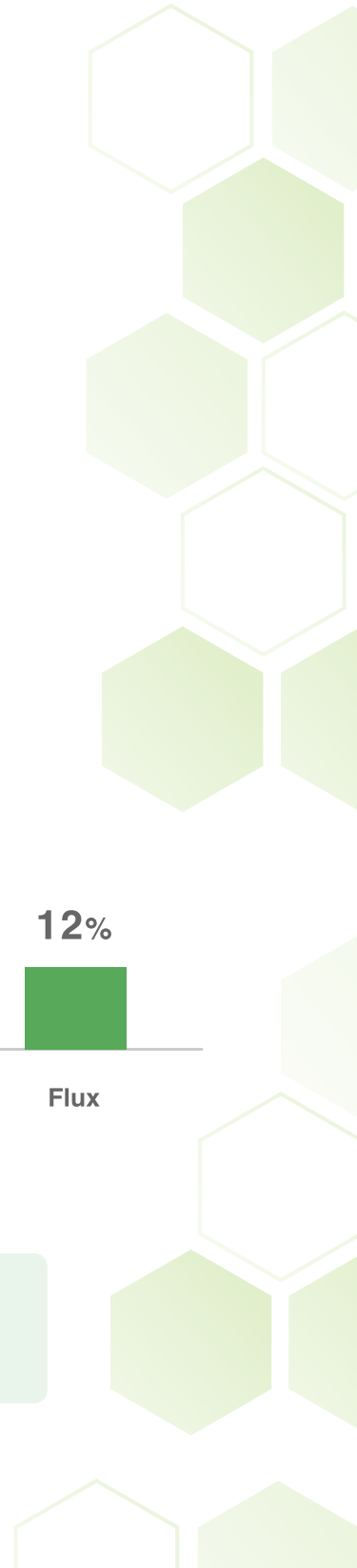


9割のSRE組織が、On-Callソリューションを活用している

6. CI/CD

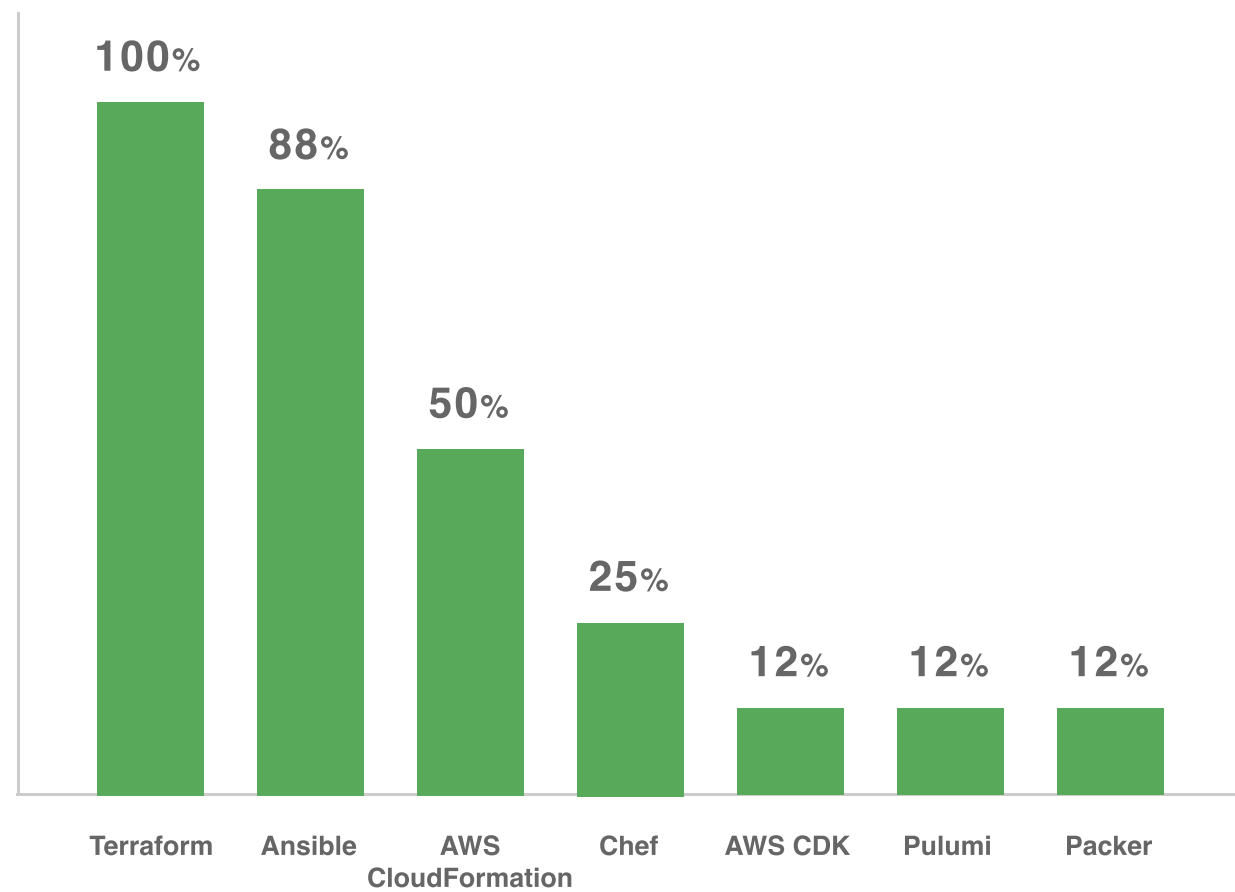


長期運用しているサービスでは、Jenkinsの活用が多い傾向にあったが、Argo CDとPipeCDの活用も増えている



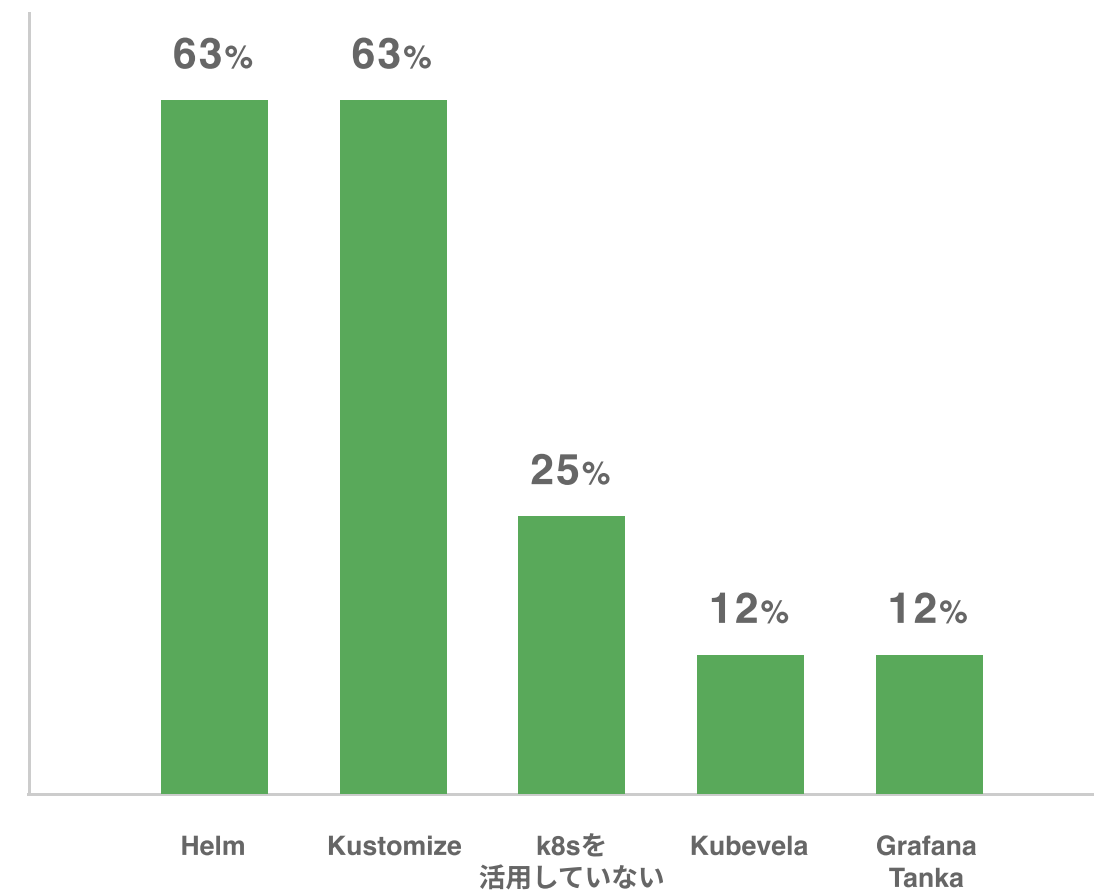
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08

7. IaC/CaC

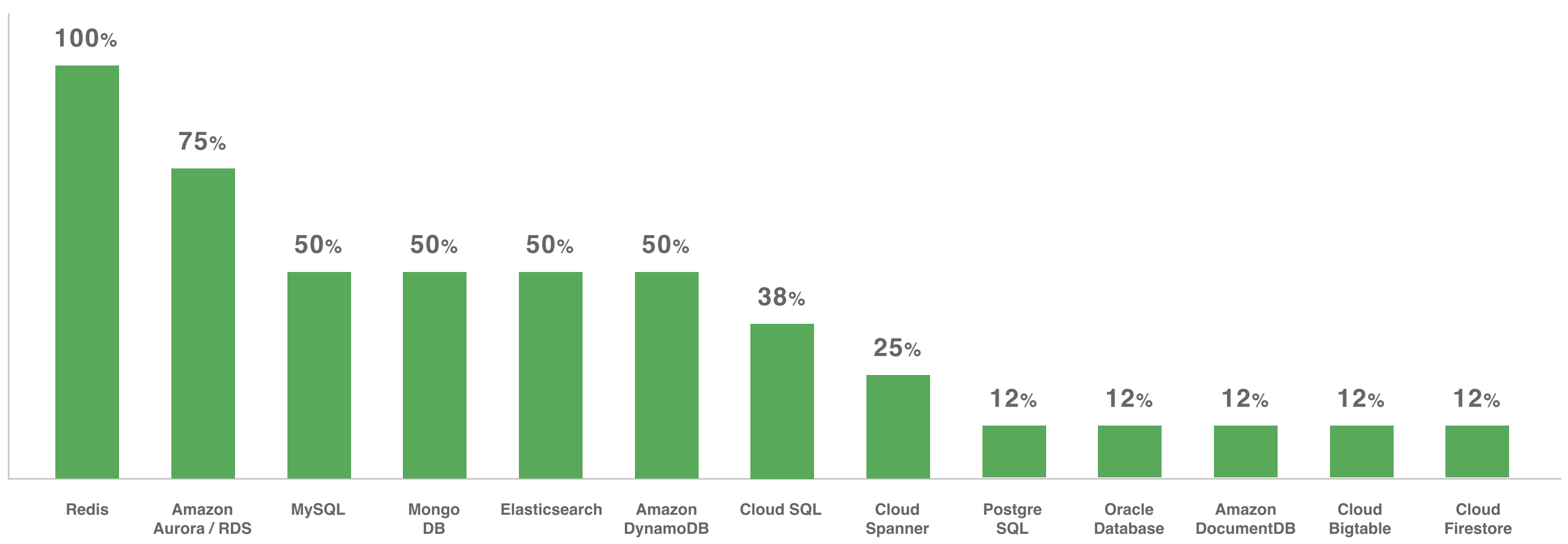
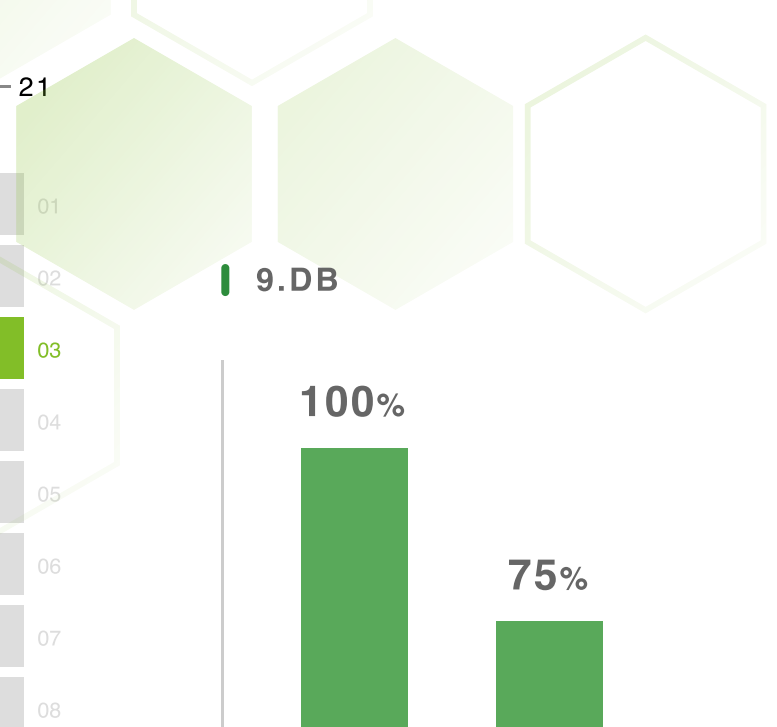


全SRE組織が、Terraformを活用している

8. k8s Manifest/deploy管理



Helm、Kustomizeの活用が多い傾向にある

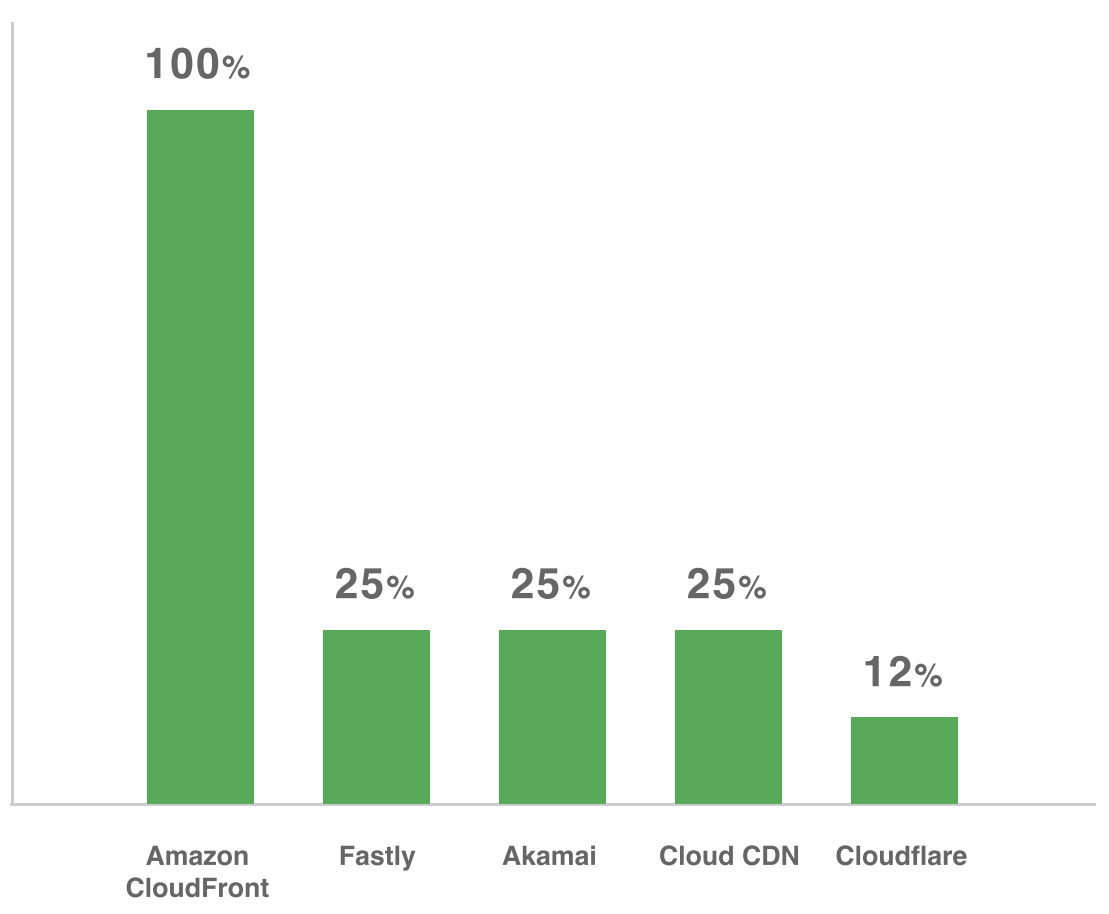


全SRE組織が、Redisを活用している
MySQLとMongoDBの活用が多い傾向にある



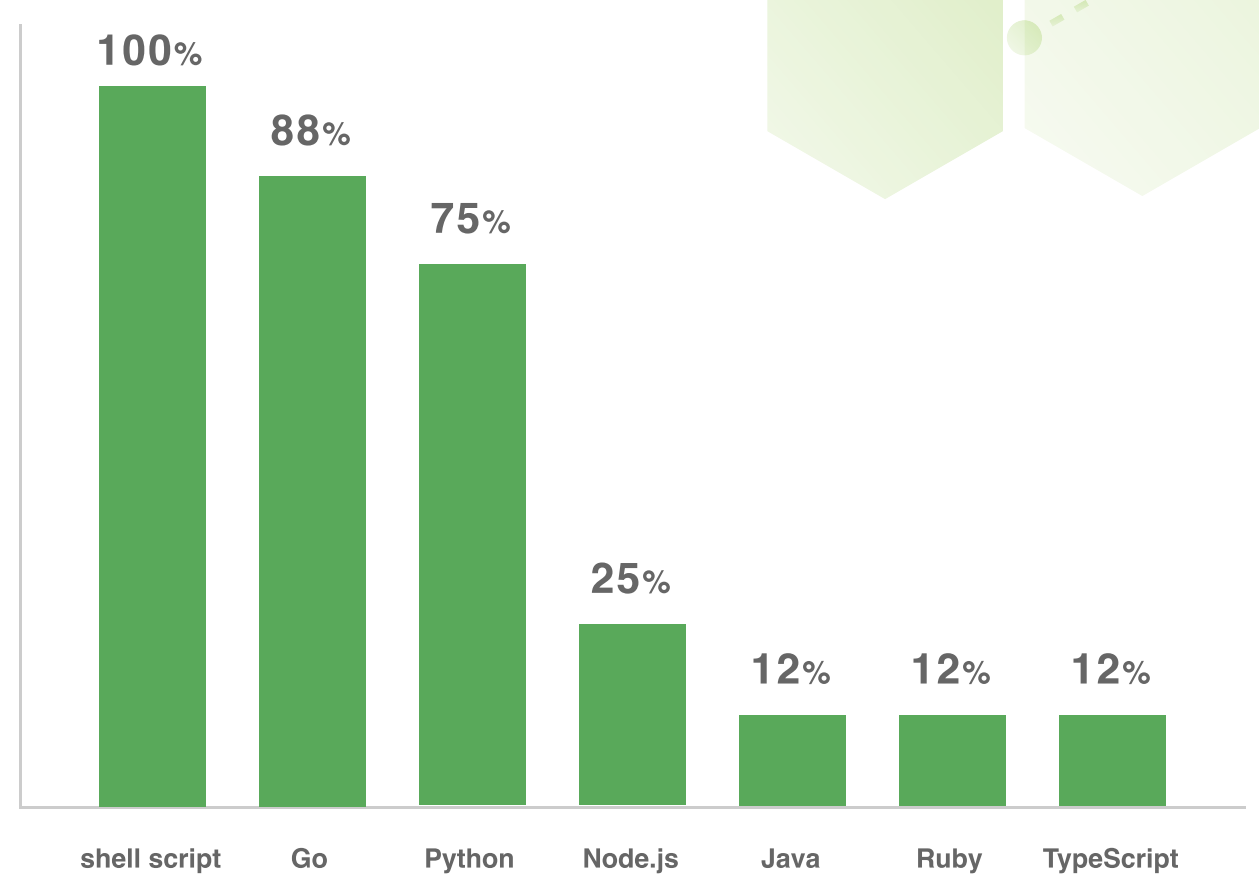
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08

10. CDN

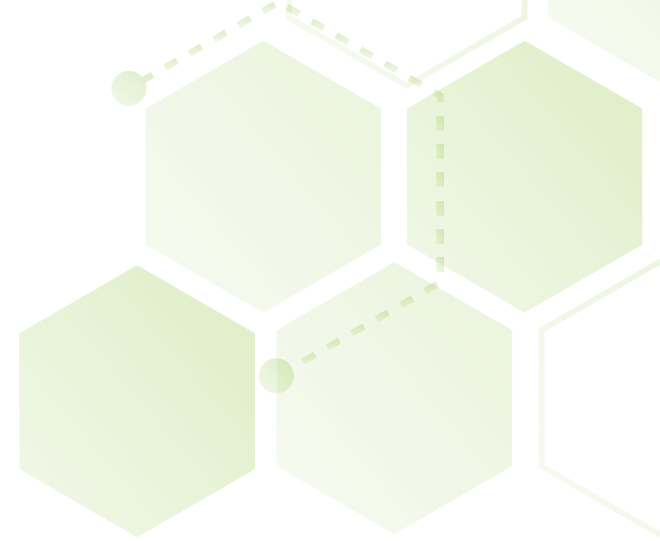


全SRE組織が、Amazon CloudFrontを活用している

11. 開発言語 (SREsが業務で使っている)



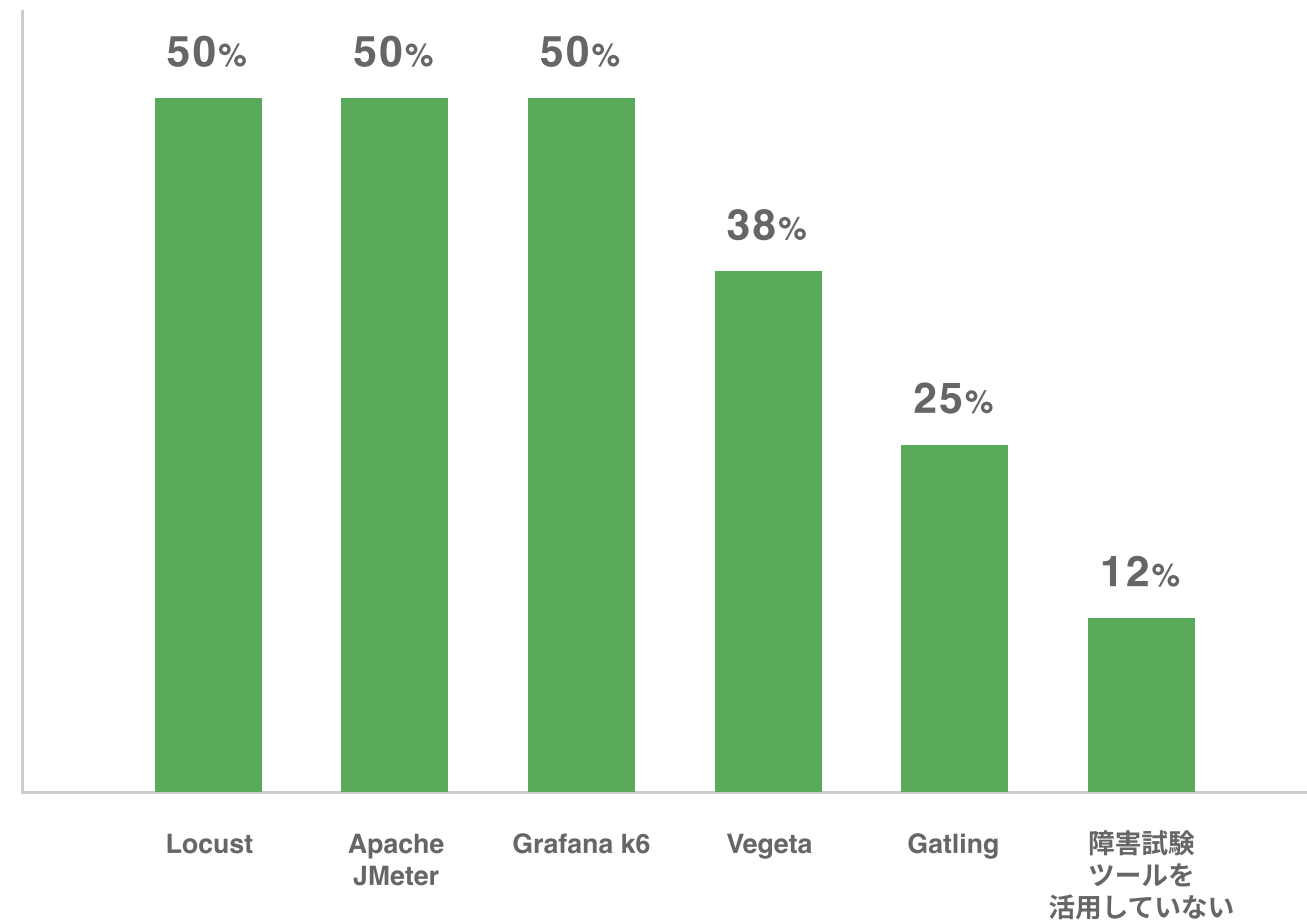
全SRE組織が、shell scriptを活用している
また、Go、Pythonの活用が多い傾向にある



- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08



12. 負荷・障害試験ツール

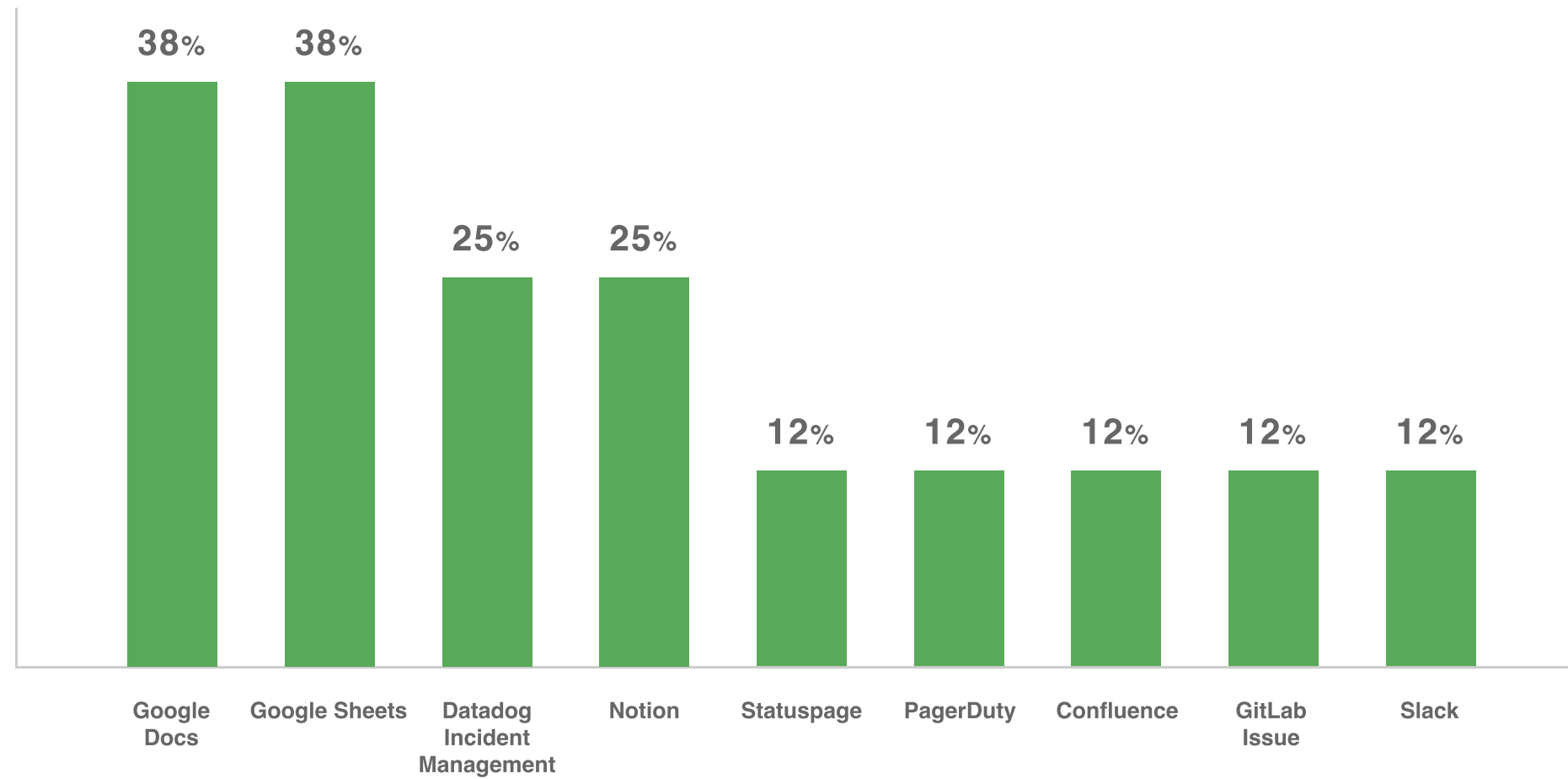


Locust、Apache JMeterの活用が多い傾向にある
近年は、Grafana k6の活用も増加傾向にある

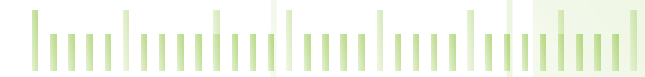


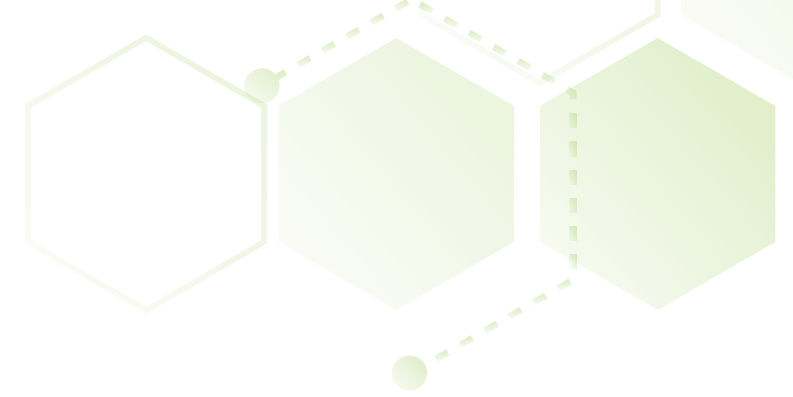


13. インシデント管理

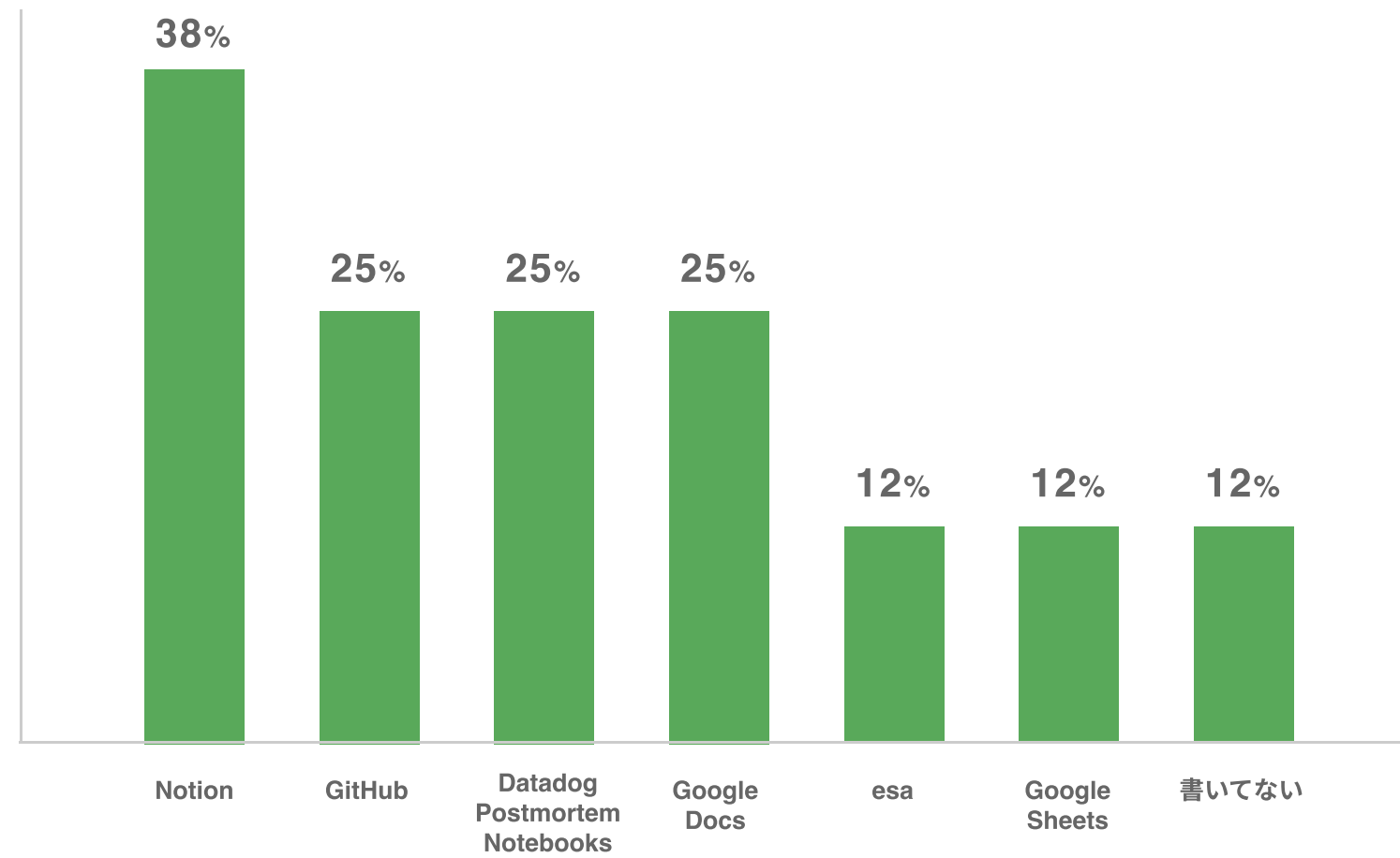


ドキュメントソリューション活用が多い傾向にあったが、
Datadog Incident Managemetの活用も増えている





14. Postmortem



ドキュメントソリューションの活用が多い傾向にある

SREs 紹介





松田 正也

Matsuda Masaya

メディア統括本部
サービスリライアビリティグループ

2019年度新卒入社。Embedded SREとして、Amebaのシステム刷新やAWS移設、運用改善などに従事。

近年は、Datadog Incident Managementを活用したインシデント管理の効率化や Postmortem文化の推進などを行っている。

好きなTerraformの機能は、movedブロック。



赤野 裕喜

Akano Yuki

株式会社タップル 技術本部
SREグループ

2013年度新卒入社。Amebaピグ事業部にてサーバーサイドエンジニアとして従事した後、タップルSREチームに異動。

現在はタップルのSREとして、サービスを安定運用する上での技術的な課題解決の推進を担当。



岩永 勇祐

Iwanaga Yusuke

株式会社AbemaTV 開発本部
開発局 SRE

2017年度中途入社。アドテク事業部(現AI事業本部)にて広告運用システムのバックエンドエンジニアとして従事した後、株式会社AbemaTVのSREチームに異動。

現在は信頼性の維持・向上のためのツール開発を行っている。



岡 麦

Oka Mugi

株式会社CAM
Creative Division

2022年度新卒入社。株式会社CAMのSREチームに所属し、マルチクラウド(AWS/GCP)で構成されている複数サービスの運用や保守、新規サービスの構築を行っている。

現在は、Kubernetes(GKE)環境の運用高度化を行い開発サイクルレベルでのDX向上を一つのミッションとして活動している。



長谷部 充洋

Hasebe Mitsuhiro

株式会社アプリボット
SRE

2013年度中途入社。多数のサービスのインフラ運用等を経験。一度退職し大手クラウド事業者にてサポートエンジニアを経験した後、2017年にサイバーエージェントへ再入社。現在はアプリボットに所属し、SREとして主にゲームのインフラを担当。安定リリース、安定運用に向き合う。



田口 雅教

Taguchi Masanori

AI事業本部 アドテクディビジョン
SREグループ SREチーム

2012年度中途入社。スマートフォン向けプラットフォーム開発のサーバサイドエンジニアとしてメディア事業に携わり、以後Unity エンジニアやネイティブエンジニア、PMなどフルスタックに経験を積み、2020年より SRE チームの開発責任者として従事する。社内リソースの有効活用や、サーバサイドなどを含んだジャンルに囚われない働き方を好む。



藤井 貴大

Fujii Takahiro

株式会社CyberZ
開発本部

2019年度新卒入社。CyberZのOPENREC事業部にてサーバサイドエンジニアとして従事した後、SREチームに異動。現在はCyberZ開発本部のSREとして、複数のサービスでKubernetes (Amazon EKS)の運用や負荷対策、DevOpsの推進などを担当。好きなAWSのサービスはAmazon DynamoDB。

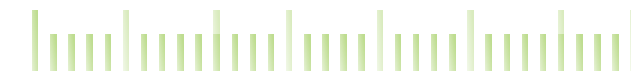


吉岡 賢

Yoshioka Suguru

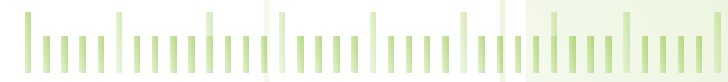
株式会社サムザップ プロダクトDiv.
SRE

2016年度新卒入社。サムザップに出向し、主力サービスのオンプレミスからAWSへの移設において設計、運用、開発に従事。エンジニア統括組織のメンバーやエンジニアの採用育成責任者などを歴任。現在は、Embedded SRE (Enablement フェーズ)として運用改善と文化作りに力を入れている。お気に入りのAWSサービスはAmazon Route 53。



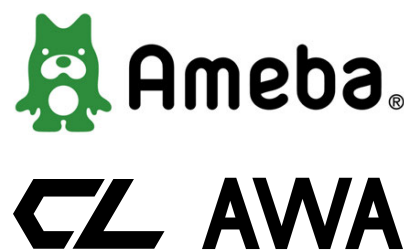
事業部毎のSRE組織情報

メディア事業



Service Reliability Group

メディア統括本部 / サービスリライアビリティグループ



Service Reliability Group (通称SRG) は、「サイバーエージェントグループが提供するプロダクトの信頼性を”横断的に”向上させる」というミッションの元、横断SREとしてメディア事業・そしてサイバーエージェントグループへのSRE導入を促進し、信頼性を向上するための取り組みを行っている組織です。

SRE組織パターン

事業部横断SRE

SRE実装パターン

SRE Center of Practice、Embedded SRE (Enabling SRE、Evangelist SRE) 、 Movable Embedded SRE

SRE活動

7～9年目

所属人数

10～15人

担当領域

AMEBA、AWA、CL、社内基盤など、100以上のサービス・システム

技術領域（スキルセット）

- ・ 監視：Datadog, Mackerel, Prometheus, Grafana, Amazon CloudWatch, Cloud Monitoring
- ・ 可観測性：Datadog, AWS X-Ray, New Relic
- ・ SLO：New Relic, Datadog
- ・ インシデント管理：Datadog Incident Management, Statuspage, Google Docs
- ・ Postmortem：GitHub, Datadog Postmortem Notebooks, Google Docs, Notion
- ・ On-call：PagerDuty, Squadcast
- ・ プラットフォーム：AWS, Google Cloud, OpenStack, OpenShift, Docker, Kubernetes
- ・ CI/CD：Jenkins, GitHub Actions, CircleCI, Concourse CI, Argo CD, PipeCD
- ・ 開発言語：GO, Java, Python, shell script
- ・ DB：MySQL, Oracle Database, MongoDB, Redis, Elasticsearch, Amazon Aurora / RDS, Cloud SQL, Amazon DynamoDB
- ・ CDN：Amazon CloudFront, Fastly, Akamai, Cloudflare, Cloud CDN
- ・ 負荷・障害試験ツール：Locust, Apache JMeter, Vegeta, Grafana k6
- ・ IaC/CaC：Terraform, Pulumi, Ansible, Chef
- ・ k8s Manifest/deploy管理：Helm, Kustomize, Kubevela, Grafana Tanka

チームの文化や体制、働く環境

- ・ 組織活性化施策として、[srg_chatting](#)、[srg_study](#)などを行っている
- ・ 横断組織なので、幅広いドメインの知見が得られる

業務内容

業務遂行する上で主軸となる3つのバリュー

- ・ 各事業の技術ノウハウを集約し、展開する
- ・ 信頼性を明確にし、組織で改善する
- ・ プロダクトに対し、技術力で直接貢献する

業務内容の一例

- ・ 事業における技術的もしくは組織的課題の発見および解決
- ・ アーキテクチャの初期設計や改善、IaCベースの提供
 - ・ CDN移設、AWS移設、Terraform moduleの提供
- ・ 様々な技術要件・規模のシステムのパフォーマンスチューニングや負荷対策
 - ・ 新規サービス、ABEMA「FIFA ワールドカップ カタール 2022」全64試合無料生中継、Ameba認証刷新
- ・ SRE文化の醸成
 - ・ プロダクトチーム、事業責任者へのインストール / Enabling
- ・ SREプラクティスの開発や様々な規模の組織への導入、改善
 - ・ SRE成熟度評価、DBワーキンググループ、コスト最適化

- ・ 各種オペレーションの自動化や効率化を支援するツールの開発運用や改善
- ・ SRE・運用プラクティスの改善
 - ・ 監視見直し、インシデント対応体制整備、ポストモテム見直し、SLO導入、トイルの撲滅
- ・ SREプラクティスの発信と組織間連携の強化

これまでの取り組みやアウトプット

- ・ CyberAgent Developers Blog
 - ・ [SRGメンバーが執筆した記事一覧](#)
 - ・ [データベースワーキンググループ活動の紹介](#)
- ・ 登壇
 - ・ [横断的なSRE推進と成熟度評価 @6社合同SRE勉強会](#)
 - ・ [シングルクラスターマルチテナンシーを目指しているEKS上でkubvelaの運用をしてみた@CNDDT2021](#)
 - ・ [Well-Architectedな組織を実現するためのチャレンジ - なぜ、CA W-Aを作ろうと思ったのか -@jawsdays2019](#)
 - ・ [事業責任者も必見! AWS Well-Architected Frameworkのビジネスへの有効活用@AWSSummitTokyo2019](#)
 - ・ [事業と歩む Ameba システム刷新の道 @AWSSummitTokyo2022](#)
- ・ ポータルサイト（[SRG Portal](#)）
 - ・ [Embedded SRE としてサービスに SRE をやってみた](#)
 - ・ [MySQL8.0のバックアップはどれがいいのか](#)

ABEMA

株式会社AbemaTV / 開発本部 / SRE Team



横断的なインフラを担当していたメンバーを中心に発足し、2021年頃にクラウド基盤を管理するチームとSREプラクティスの導入を推進しているチームに分離。

以降、ユーザー向け機能を提供するチームを中心にSREプラクティスの導入を行い、継続的に適切な信頼性への投資をもってデリバリー速度の向上を目指す。

SRE組織パターン

単一プロダクト専任SRE

SRE実装パターン

Platform SRE、Embedded SRE (Enabling SRE)

SRE活動

4～6年目

所属人数

1～4人

担当領域

ABEMA

技術領域（スキルセット）

- ・ 監視：Prometheus, Grafana, New Relic, Amazon CloudWatch, Cloud Monitoring
- ・ 可観測性：OpenTelemetry, Cloud Profiler, Cloud Trace
- ・ SLO：Prometheus, Grafana
- ・ インシデント管理：PagerDuty, GitHub
- ・ Postmortem：esa, Notion
- ・ On-call：PagerDuty
- ・ プラットフォーム：Google Cloud, Kubernetes, AWS
- ・ CI/CD：PipeCD, GitHub Actions, CircleCI
- ・ 開発言語：Go, Python, shell script
- ・ DB：MongoDB, Redis, Elasticsearch, Cloud Spanner, Cloud Bigtable, Cloud Firestore
- ・ CDN：Akamai, Cloud CDN, Fastly, Amazon CloudFront,
- ・ 負荷・障害試験ツール：Grafana k6, Locust
- ・ IaC/CaC：Terraform, AWS CloudFormation, Packer
- ・ k8s Manifest/deploy管理：Kustomize, Helm

チームの文化や体制、働く環境

- ・ 活動における技術領域の制限はなく信頼性の視点で課題の発掘から解決までを行う
- ・ 抽象的な問題を扱うことが多いため、ディスカッションを中心としたコミュニケーションは多い

業務内容

- ・ 信頼性維持におけるツール開発、および体制の構築
- ・ SRE、およびDevOps関連のプラクティスや実施サポート
- ・ 大型配信時の取りまとめ、およびサポート

業務内容の一例

- ・ SLI/SLO設定・可視化・アラート基盤の構築(Platform)
- ・ SLI/SLO導入・運用サポート(Enabling)
- ・ インシデント管理ツールの導入・構築
- ・ 障害レベルの導入・対応フローの策定
- ・ 負荷・障害試験の実行基盤構築
- ・ 大型配信時におけるキャパシティプランニング・試験実施・監視体制の構築

これまでの取り組みやアウトプット

- ・ 登壇
 - ・ [ABEMA Developer Conference 2021: ABEMA SREの挑戦～Frontend & Backend TeamへのEmbedded SRE導入～](#)
 - ・ [SRE NEXT: 事業の成長と共に歩む、ABEMA SRE探求の歴史](#)
 - ・ [大規模イベントを成功させるための負荷・障害・セキュリティ対策](#)
 - ・ [FIFA ワールドカップ 2022に向けたキャパシティ確保の軌跡](#)
 - ・ [AbemaTV：Google Cloud の活用による徹底した負荷対策によって世界的スポーツ イベントにおいて安定した視聴体験を提供](#)

IU-SRE



I CONNECT WITH YOU

Always connect with our customers through interaction.
An organization that includes people, business, and culture.
Inspiration is generated by young people to lead the market.



横断的なインフラを担当していたメンバーを中心に発足し、2021年頃にクラウド基盤を管理するチームとSREプラクティスの導入を推進しているチームに分離。

以降、ユーザー向け機能を提供するチームを中心にSREプラクティスの導入を行い、継続的に適切な信頼性への投資をもってデリバリー速度の向上を目指す。

SRE組織パターン

事業部横断SRE

SRE実装パターン

IU-SRE: Embedded SRE (Enabling SRE) ,Platform SRE,
SRE Center of Practice,Movable Embedded SRE
Tapple: Product SRE,Embedded SRE (Enabling SRE)

SRE活動

4～6年目

所属人数

5～9人

担当領域

エンタメコンテンツ、ビジネスバラエティメディア、ライフスタイルメディア、Tapple、社内基盤など、80以上のサービス

技術領域（スキルセット）

- 監視：Datadog, Prometheus, Grafana, Amazon CloudWatch, Cloud Monitoring
- 可観測性：Datadog
- SLO：Datadog
- インシデント管理：Notion, Datadog
- postmortem：Notion
- On-call：Grafana OnCall
- プラットフォーム：AWS, Google Cloud, Microsoft Azure, Docker, Kubernetes
- CI/CD：Jenkins, GitHub Actions, Argo CD
- 開発言語：Go, Node.js, shell script
- DB：MySQL, MongoDB, Redis, Elasticsearch, Amazon Aurora / RDS, Amazon DocumentDB
- CDN: Amazon CloudFront
- 負荷・障害試験ツール：Apache JMeter, Grafana k6
- IaC/CaC：Terraform, AWS CloudFormation, Ansible
- k8s Manifest/deploy管理：Helm

チームの文化や体制、働く環境

- 会社の方針に合わせ週2日のリモート
- 横断組織なので、幅広いドメインの知見が得られる
- 新規事業が多い環境なので設計から関われる

業務内容

- 新規案件の設計/構築
- 運用改善、負荷対策

業務内容の一例

- ArgoCDを用いた環境の刷新
- 個人情報などセキュアな情報を扱う基盤設計や構築
- MongoDB Atlas/Elasticsearchのパフォーマンス・チューニングとサイジング

これまでの取り組みやアウトプット

- [CyberAgent Developers Blog](#)
 - [75億ドキュメント以上のデータを保持するMongoDBを、Amazon EC2からMongoDB Atlasへ約3ヶ月で移設した方法](#)
 - [SRE 内定者アルバイト成果ブログ](#)
 - [タップルSREはタップルの成長をどこまで支えられるか | CyberAgent Developers Blog](#)
- CAM
 - [負荷試験環境に k6-operator を活用し、ChatOps ライクな負荷試験を実現できた話](#)
 - [GCPリソースをTerraform管理してみよう](#)
- 登壇
 - [タップルにおけるBFFとopenAPI導入事例の紹介 | CA BASE NEXT - CyberAgent Developer Conference by Next Generations](#)
 - [100を超えるクラウドサービス環境をSREで効率的にコスト最適化した話 | CA BASE NEXT - CyberAgent Developer Conference by Next Generations](#)
 - [計画的に負荷リスクを排除するためのキャパシティプランニング - Speaker Deck](#)

CyberZ

株式会社CyberZ / 開発本部



CyberZのSREは、ライブ配信プラットフォームであるOPENREC.tvなどのサービスを中心に、システムの安定性を向上させるために日々従事しています。最近ではそれ以外の新規サービスについてもSRE業務を兼任しており、SREとしてよりスケールメリットのあるチームを目指しています。

SRE組織パターン

子会社専任（複数プロダクト横断）SRE

SRE実装パターン

Product SRE、Embedded SRE（Enabling SRE）、Movable Embedded SRE

SRE活動

4～6年目

所属人数

1～4人

担当領域

OPENREC.tvおよび一部の新規事業



技術領域（スキルセット）

- プラットフォーム: AWS, Kubernetes
- 監視: Prometheus, Grafana, Amazon CloudWatch
- 可観測性: OpenTelemetry, AWS X-ray
- SLO: 独自実装基盤を利用
- オンコール: Grafana Oncall
- CI/CD: GitHub Actions, Argo CD, PipeCD
- IaC/CaC: Terraform, AWS CloudFormation, Ansible
- k8s Manifest/deploy管理: Kustomize, Helm
- DB: Redis, Elasticsearch, Amazon Aurora / RDS, Amazon DynamoDB
- CDN: Amazon CloudFront
- 開発言語: shell script, Python, Go
- 負荷試験ツール: Locust
- インシデント管理: Slack, Notion
- Postmortem: GitHub, Notion, Google Docs
- ドキュメントツール: Notion, Google Docs

チームの文化や体制、働く環境

- インフラよりも、バックエンドなどの経験があるエンジニアが多い
- 場合によってはバックエンドの実装を行うこともある

業務内容

- SREプラクティスの模索、実践
- 各開発チームへDevOps関連知識の啓蒙や実践サポート
- OPENREC.tvの一部開発業務

業務内容の一例

- Kubernetes環境での分散トレーシングの検証および導入
- SLI/SLOやオンコール体制など、より良い開発体験を補助する仕組みを検討
- バリューストリームマッピングの実施主導
- FourKeys計測基盤の構築、提供
- 古いアプリケーションの廃止およびKubernetesへの移行など

これまでの取り組みやアウトプット

- Blogs
 - [OPENREC.tvにおけるOpenTelemetryの活用](#)
 - [DevOpsの取り掛かりとしてのバリューストリームマッピングのススメ](#)
 - [CyberZがAmazon DynamoDBを使用してフォロアタイムラインの表示に必要なRead-Light方式を実現した方法術](#)
- 登壇
 - [VMとAWS ECSがメインのインフラにKubernetesを導入した効能](#)
 - [CyberZのSREにおけるプロダクト開発チームとの関わり方](#)
 - [OPENREC.tvにおけるDynamoDBを用いたデザインパターン](#)

ゲーム事業



CA GameFi

CA GameFi / SRE



CA GameFiのSREはCA GameFiの全プロダクトに注力し、ユーザ体験を損なわせないことを目的にプロダクトの信頼性を向上させる取り組みを行っています。

SRE組織パターン

子会社専任（複数プロダクト横断）SRE

SRE実装パターン

Product SRE、Platform SRE

SRE活動

1～3年目

所属人数

1～4人

担当領域

マルチテナントインフラ基盤および
CA GameFi内の全プロジェクト

技術領域（スキルセット）

- 監視：Datadog, Grafana, Cloud Monitoring, VictoriaMetrics
- 可観測性：Datadog
- SLO: Datadog
- インシデント管理：Notion
- postmortem：Notion
- On-call：PagerDuty
- プラットフォーム：AWS, Google Cloud, Docker, Kubernetes
- CI/CD：Jenkins, GitHub Actions, Flux
- 開発言語：Go, Python, shell script
- DB：Redis, Cloud SQL, Cloud Spanner
- CDN：Amazon CloudFront
- 負荷・障害試験ツール：Apache JMeter, Grafana k6
- IaC/CaC：Terraform, Ansible
- k8s Manifest/deploy管理：Kustomize

チームの文化や体制、働く環境

- SREで運用しているインフラ基盤を担当しつつ、各プロジェクトがインフラ基盤を利用するに当たって導入や運用も担当します
- 会社の方針に合わせ週2日リモートによる開発体制をとっています

業務内容

- Kubernetesを用いたマルチテナントインフラ基盤の運用
- 各プロダクトのインフラ構築、運用、セキュリティチェック、コスト管理など
- SREの実践

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08

アプリボット 事業部 / SRE



アプリボットのSREチームは、アプリボット内の全プロジェクトのインフラを一手に担う横断組織です。小規模な新規事業から大型のIPゲームタイトルまで、様々なプロジェクトに寄り添い、臨機応変に対応しています。最近では、サイバーエージェントのゲーム・エンターテインメント事業部（SGE）に所属する各社を横断したプロジェクトや、非ゲーム領域のWebサービス開発事業など、より活動領域が拡大しています。

SRE組織パターン

子会社専任（複数プロダクト横断）SRE

SRE実装パターン

Product SRE、Platform SRE

SRE活動

7～9年目

所属人数

5～9人

担当領域

アプリボット内の全プロジェクト
（小規模な新規事業から大型のIPゲームタイトルまで様々）



技術領域（スキルセット）

- 監視：Datadog
- 可観測性：Datadog
- SLO：Datadog
- インシデント管理：Google Docs, Google Sheets, Confluence, GitLab Issue
- On-call：PagerDuty
- プラットフォーム：AWS, Google Cloud, Docker
- CI/CD：Jenkins, GitHub Actions
- 開発言語：Go, Python, shell script
- DB：MySQL, PostgreSQL, Redis, Amazon Aurora / RDS, Amazon DynamoDB
- CDN：Amazon CloudFront
- 負荷・障害試験ツール：Gatling, Vegeta
- IaC/CaC：Terraform, Ansible

チームの文化や体制、働く環境

- 全社方針に合わせ、毎週火曜、木曜のみリモート推奨（2023年6月現在）
- 主担当のプロジェクトを持ちつつ他プロジェクトも横断して担当する人や、1つのプロジェクトを専属として担当する人、全プロジェクトを俯瞰して臨機応変に対応する人など、本人の適正や状況に合わせて担当を割り振っています。

業務内容

- インフラ環境の設計/構築/運用/改善等
- 安定リリース/安定運用の実現に向けた負荷対策やモニタリング、キャパシティプランニングなど
- ユーザーのゲームプレイ体験向上や開発者の開発体験向上のための各種改善
- セキュリティの担保

業務内容の一例

- Applibot Gems Vol.1
 - <https://techbookfest.org/product/5646370278998016>
- Applibot Gems Vol.2
 - <https://blog.applibot.co.jp/2020/09/11/applibot-gems-vol-2/>

サムザップ

株式会社サムザップ / SRE



サムザップでは各プロジェクトにSRE 担当がおります。プロジェクトに深く入り込み、開発メンバーと一緒に SRE導入や運用を見据えた各プロジェクトに最適なSRE文化作りなど、サービスの信頼性を向上するために日々取り組んでいます。

また、プロジェクトを超えた情報共有を行い、技術知見の蓄積、標準化、SRE文脈に基づいた文化づくり、サイバーエージェントのゲーム・エンターテインメント事業部（SGE）各社との技術交流など幅広く活動しております。

SRE組織パターン

単一プロダクト専任SRE

SRE実装パターン

Product SRE / Embedded SRE (Enabling SRE)

SRE活動

4～6年目

所属人数

1～4人

担当領域

運用サービス、新規サービスのSRE実践

技術領域（スキルセット）

- ・ 監視：Datadog, New Relic, Amazon CloudWatch
- ・ 可観測性：Datadog, New Relic
- ・ SLO：Datadog
- ・ インシデント管理：Google Sheets
- ・ Postmortem：Google Sheets
- ・ On-call：PagerDuty
- ・ プラットフォーム：AWS, Google Cloud, Docker
- ・ CI/CD：Jenkins, GitHub Actions, AWS CodeBuild
- ・ 開発言語：Go, shell script
- ・ DB：Redis, Amazon Aurora / RDS
- ・ CDN：Amazon CloudFront
- ・ 負荷試験ツール：Locust, Gatling, Vegeta
- ・ 障害試験ツール：障害試験ツールを使用せず、自分たちでインスタンスを強制停止したりして確認している
- ・ IaC/CaC：Terraform, Ansible
- ・ k8s Manifest/deploy管理：k8sを使用していない

チームの文化や体制、働く環境

- ・ 主担当制による責任（独立性）と情報共有
 - ・ 共有されている資産を元に構築することが可能
 - ・ 各プロジェクトにおける技術的な導入判断ができる
- ・ 情報共有
 - ・ 各サービスの主担当が集まり、情報共有する定例
 - ・ 会社の枠を超えて子会社間で情報共有する定例
- ・ 働く環境
 - ・ 会社の方針に合わせて週2日のリモート
 - ・ サービスやチームで相談し柔軟に対応

業務内容

- ・ インフラ環境の設計、構築、運用、改善などコストを含む管理
- ・ プロダクトに対する負荷試験、セキュリティチェック、障害試験などの実施
- ・ SREの実践（運用改善、コスト管理、負荷対策、監視、障害管理など）

これまでの取り組みやアウトプット

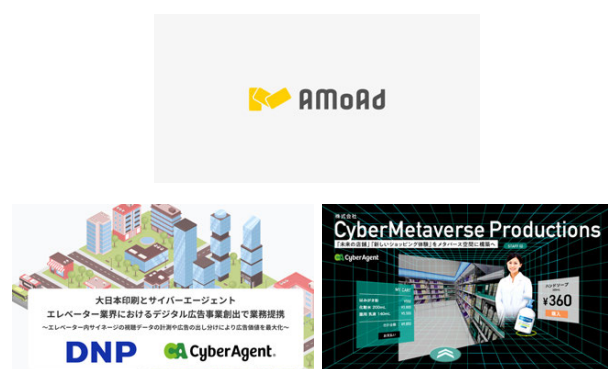
- ・ Sumzap Engineering Blog
 - ・ [Glueを用いてAuroraからBigQueryへのデータ転送処理を高速化した話](#)
 - ・ [AWS の Blue/Green Deployment でハマったことをつらつらと](#)
 - ・ [社内制度を利用して AWS 認定を10個取得した話](#)
 - ・ [時限オートスケーリングとスポットインスタンスによるリソース最適化](#)
 - ・ [CodeBuildを使用したFargate開発環境の自動構築](#)
- ・ 登壇
 - ・ [AWS Summit 2019 Tokyo, ある SRE チームの挑戦 運用 6 年目の大規模ゲームを AWS 移設後に安定運用するための技術と今後の展望](#)
 - ・ [AWS Summit Online Japan 2020, 『このファン』におけるマネージドサービスドリブンなアーキテクチャへの挑戦](#)
 - ・ [ゲームインフラ運用・開発メソッドLT大会！ - AWS, Kubernetes, Jenkins - 2020, 「戦国炎舞 - KIZNA-」で行ったAWSのコスト最適化の話](#)

AI・DX事業



アドテクディビジョン

AI事業本部 / アドテクディビジョン / SREグループ



SREグループは、AI事業本部が提供するAI・DXを中心とした複数の事業・サービスと一体となり、各サービスに所属しながらSREの視点を持ってインフラ構築やバックエンドなどの開発を行っています。
また、横軸組織としてもSRE観点の横展開や、技術の共有を行い効率化を目指しています。

SRE組織パターン

事業部横断SRE

SRE実装パターン

Product SRE、Embedded SRE (Evangelist SRE)、Movable Embedded SRE

SRE活動

1～3年目

所属人数

5～9人

担当領域

- ・ アドテクディビジョン
 - ・ CPI事業部：CPI/ノンインセンティブ型広告配信
 - ・ AF事業部：CPA/インセンティブ型広告配信
 - ・ AMoAd事業部：アドネットワークを中心とした配信
- ・ 協業DXディビジョン
 - ・ エレベーターサイネージカンパニー
- ・ AICG関連
 - ・ 株式会社CyberMetaverse Productions
 - ・ AIクリエイティブディビジョン Future Event/Live事業部

技術領域（スキルセット）

- ・ 監視：Datadog, Mackerel, Prometheus, Grafana, New Relic, Amazon CloudWatch, Cloud Monitoring
- ・ 可観測性：Datadog, AWS X-Ray
- ・ SLO：Datadog
- ・ インシデント管理：Google Docs, Google Sheets
- ・ Postmortem：Datadog Postmortem Notebooks
- ・ On-call：PagerDuty
- ・ プラットフォーム：AWS, Google Cloud, OpenStack, Docker, Kubernetes
- ・ CI/CD：Jenkins, GitHub Actions, Argo CD
- ・ 開発言語：Go, Python, Ruby, Node.js, shell script, TypeScript
- ・ DB：MySQL, MongoDB, Redis, Amazon Aurora / RDS, Cloud SQL, Amazon DynamoDB
- ・ CDN：Amazon CloudFront
- ・ 負荷・障害試験ツール：Locust, Apache JMeter, Grafana k6
- ・ IaC/CaC：Terraform, AWS CloudFormation, AWS CDK, Ansible, Chef
- ・ k8s Manifest/deploy管理：Kustomize, Helm

チームの文化や体制、働く環境

- ・ 基本会社の方針に合わせていますが、担当チームの働き方に合わせるケースも多いです。現状リモートは多めではありますが、出社しているメンバーもいます。
- ・ チーム内勉強会として、GoogleのSRE本の輪読であったり、ベストプラクティスの輪読会などを実施（週1）
- ・ 横軸組織として複数の事業部を見ているため、ナレッジ共有会（週1）
- ・ チーム内懇親会(月1)

業務内容

- ・ 基本、Embedded SREとして、事業並びに、サービスと一体になって開発を行っています。一体になることでコミュニケーションロスを減らし、パフォーマンスのあげやすい状態を作る。また、横軸として、技術の共有を行い、ノウハウなどの効率化を図る
- ・ 既存システムから、マイクロサービスなどへの移行
- ・ Kubernetesなどコンテナ環境の構築・支援
- ・ 監視/アラート設計、システムの安定に向けたサービスのモニタリング、キャパシティプランニング、SLI/SLOなど策定、またこれらのファシリテート
- ・ インフラ（サーバ、ネットワーク）設計・構築
- ・ システム障害時の対応

- ・ 要件に応じたプラットフォーム、アーキテクチャの選定・構築
- ・ CI/CDの構築、運用
- ・ MLOpsの構築並び、機械学習の導入
- ・ アプリケーションの開発、支援
- ・ 新規開発、サーバ構築全般支援

これまでの取り組みやアウトプット

- ・ 登壇
 - ・ 協業DXDiv Developer's way （AI事業本部内のみ）

SREsの取り組み



Datadog Incident Management使ってAmebaのインシデント管理体制整備した話

はじめに

横軸組織のSRGがEmbedded SRE (Site Reliability Engineer)としてAmebaに参加し、SRE文化の構築の一端としてIncidentMangementを行うことでサービス品質の向上に寄与したお話をします。

背景

インシデント対応を行うメンバーの偏りやインシデントコマンダー不在による連携力低下、トリアージ・インシデント対応フローの未定義に起因した問題解決遅延によりMTTRの肥大化が問題となった。

	現在	理想
属人性	気づいた人が対応というケースがやや多い →大体同じ人(良い奴)が対応してくれる	ローテーションでIC・OCが対応 チーム全体で一丸となって解決するような空気感
インシデント対応難易度	インシデントフローはあるものの、 SEV1相当のフローしか存在しないため汎用性が低い ICの負担大	SEVレベル毎に対応フローを整備 ICに加えてOCを追加し、 責務分離することでICの負担軽減
障害網羅性	トリアージを設定しているものの、 インシデント対象となる最低レベルのボーダーが高く、 全ての障害に適用できない。	適切に設定したトリアージを元に 各インシデントの対応優先度が明確になり、 判断・対応が迅速に行われる
インシデント振り返り	肌感インパクトが大きかった障害に対して Postmortemを実施	特定のSEVレベル以上インシデントに対して ICが自発的にPostmortemを実施する
評価	いくらインシデント対応をしたとしても、 明確な評価に結びつきにくい	高いレベルでインシデント対応してくれた人に対する表彰を機会を実施するなど、インシデント対応者を評価する文化を組み込む
運用改善難易度	MTTRなどが未計測であるため、 改善対応による効果測定が不可	MTTR計測による改善対応の効果測定

図1: 現在と理想とのギャップ

課題整理

課題整理を行ったところ、以下のような問題に直面した。(図1)

体制

既存の運用ではインシデント発生時にメンバー各人がどういった役割として問題解決に向けて稼働するかが決定していなかったため、特にIncidentCommandarとOperationCommandarを役割と責務を明確化し、当該職務が重要な役割をになっている事を説明。(図2)

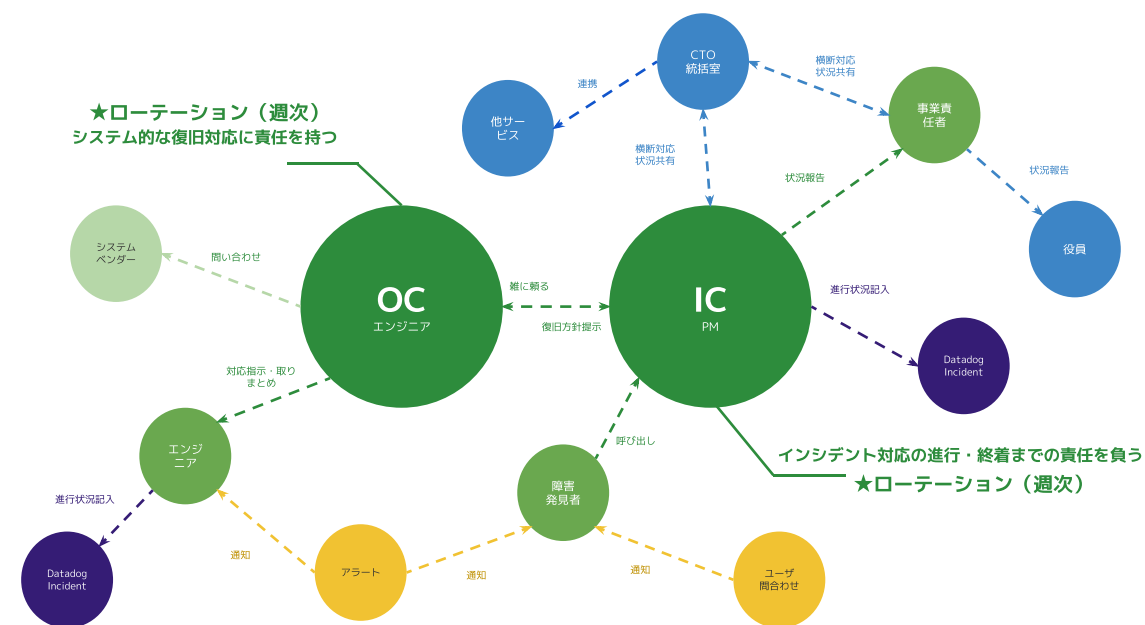


図2: 体制 (相関図)

トリアージ

重大障害に対してのみトリアージが設定されていたものを、復旧が必要となる全ての障害をインシデントのスコープとして再定義。（図3）



※SEV5はユーザ影響なしとする

図3: これからのトリアージ

当該トリアージに関してはDatadogIncidentを利用すれば、[Setting]にて表記可能となるため、記載した上でチーム全体に案内すると良い。（図4）

Severity Levels

Severity describes the impact an incident has on your customers. You can tailor the descriptions (displayed on incident declare) to your organization.

◆ SEV-1 Critical	メディア事業の継続可能性に関わるインシデント Ameba単体でなく、事業部ひいては全社レベルで対応・方針策定が必要なレベル
◆ SEV-2 High	Ameba事業の継続可能性に関わるインシデント 事業運営にクリティカルな影響があり事業責任者レベルでの対応方針の決定が必要
◆ SEV-3 Moderate	対ユーザーへのアナウンス・コミュニケーションが必要なインシデント 現場エンジニアだけでは完結せず、関連サービスや対外的なコミュニケーションが必要なインシデント PMやエンジニアリーダーレベルでの対応方針のハンドリングや意思決定が必要
◆ SEV-4 Low	復旧対応が必要な全てのインシデント（アラート） 現場エンジニアのみで復旧可能であり、インシデントコマンダーは介入しない
◆ SEV-5 not critical	ユーザ影響なし

Start at SEV-0

[+ Add a New Severity](#)

Status Levels

Status describes the current state of the incident response. You can tailor the descriptions (displayed on incident declare) to your organization.

ACTIVE	障害が進行中。復旧の目処が立たず。
STABLE	障害が進行中。復旧までの対応フローが決まっている。
RESOLVED	暫定対応及び自然復旧により障害が復旧済み。
COMPLETED	Postmortemの記載及び振り返りが完了しており、当該振り返りによって作成された恒久対応を含めて全ての対応が完了している状態。

図4: DatadogIncident

DatadogIncident

DatadogIncidentでのインシデント管理は特にサマりに相当する以下の3点をリアルタイムでアップデートすることが重要です。

1. What happened(何が発生したか)
2. Impact(ユーザ影響/いつからいつまで)
3. Why it happened(なぜ起こったか)

これらは事後に振り返りを行った際に必要な情報となるだけでなく、インシデント発生時において最新化することでサービスのメンバーのキャッチアップコストが大幅に削減することが可能です。

事業責任者目線においては状況を的確かつ迅速に把握することで、意思決定のスピードを向上させます。

エンジニア目線においては障害解決チームへのヘルプ難易度が低下するといった効果が見込めます。

DatadogPostmortem

インシデント収束後に振り返りを行うことは重要です。DatadogPostmortemを利用すれば、発生した障害に当該Postmortemを紐づけることが可能であり、障害概要や発生時刻、解決に至るまでのタイムラインなどの情報がPostmortemに引き継がれるため運用が非常に楽になります。

評価

Incident対応における評価は運用が困難です。どのチームもパッシブな対応となるインシデント解決を目標として挙げることはなかなかありません。こういったケースだとチームとして設定しているミッションを優先することが個人としての評価向上の近道となってしまうため、インシデント対応に対する評価を明確化しましょう。

注意点はMTTRの短縮などの長期的かつ個人ではコントロールできない数値を追うのはNGです。インシデントコマンダー担当数やMTTAといった間接指標で評価していくと良いでしょう。

よくつまづく問題

どれだけ情報整備を行なっても、人は自らDocsを確認して意欲的に対応を進めることはできません。

IncidentManagementにおいてはIncidentCommandarを中心にインシデント管理手法について理解をしてもらうためにフォローを積極的に行うことでインシデント発生時に解決まで責任をもつ人材を育成することができます。

また、「そもそも設定したトリアージに即したインシデントが起票されない」という問題も発生しがちです。これはIncidentManagementを行う人間がアラート、CSからの問い合わせやリリース後に確認された障害などに対して起票を行うことを促すことで文化として醸成が進みます。

終わりに

まずは体制・トリアージやインシデント対応フローの整備は最低限必要として、ビジネスサイドを巻き込むには「地道に声かけ・質問への返答速度向上」が一番大事です。

その上でもインシデント運用が漏れるケースはあるので、代わりに起票・更新したりといった手厚いサポートが必須となります。

CyberZでの Toil 可視化の取り組み

はじめに

はじめまして。CyberZでSREをしている藤井と申します。普段はOPENREC.tvなどを中心に、いくつかのサービスにSREとして関わっています。

CyberZではSREチームが出来てからというもの、各種古いアプリケーション基盤をKubernetesに移行したり、TerraformによるIaCを推し進めたり、各種DevOpsの実践を牽引・啓蒙したりしています。その結果として、普段の業務におけるToilの割合はかなり減ったと感じています。しかし、明確にToilについて計測はこれまでできていませんでした。

今回は、今一度Toilに向き合い、Toilの計測を始めた話についてお話しできればと思います。

Toilの定義

Toilの定義は、GoogleのSRE本 第五章「[Eliminating Toil](#)」に従っています。

- ・ 手動での作業を伴うもの
- ・ 繰り返されるもの

- ・ 自動化できるもの
- ・ 戦略的でなく、戦術的であるもの
- ・ 一時的な対処であり、永続的な価値をもたらさないもの
- ・ サービスの成長に伴い作業量が増えるもの

基本的には実際の作業者が、上記の定義に則りそれをToilとみなすかどうかなどを判断します。

Toilの計測・可視化

本題の計測および可視化についてですが、今回は計測に際して新たな基盤を構築したりはしていません。

CyberZでは昨年からFourKeysの計測のため、OSSである[dora-team/fourkeys](#) に少し手を加えた基盤を構築および運用しています。データストアであるBigQueryには、紐づけたGitHub Repositoryの各種イベントがWebhook経由で書き込まれており、これをGrafanaで可視化することでFourKeysメトリクスとして役立てています。

今回のToil計測は、このFourKeys基盤のデータを活用して構築しています。具体的には、GitHub Issueのイベントを用い、特定のラベルの付いたIssueの件数などをGrafana

で可視化しています。

参考程度ですがGrafanaダッシュボードで実行しているSQLの一部を以下に貼っておきます。

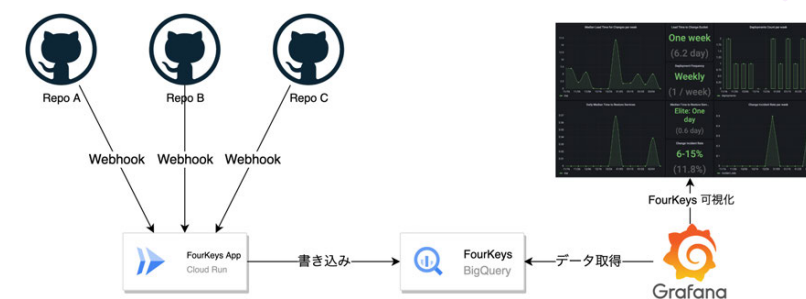


図1: FourKeys基盤

FourKeysはGitHubのWebhookイベントを1行ずつBigQueryのevents_rawテーブルに書き込んでいます。event_typeがissuesのイベントの最新一件から、Toilレベルの付与されたもののみ限定することで、Toil Issueを抽出しています。


```

SELECT
  *
FROM
  (
    SELECT
      AS VALUE ARRAY_AGG(
        events_raw
      )
    ORDER BY
      time_created DESC
    LIMIT
      1
  ) [OFFSET(0)]
FROM
  fourkeys.events_raw
WHERE
  event_type = "issues"
GROUP BY
  id
) tolis

```

```

WHERE
  REGEXP_CONTAINS(
    JSON_EXTRACT(metadata,
      '$.issue.labels'),
    '"name": "([^\"]*?)Toil([^\"]*?)"'
  )
;

```

運用フローとしては以下の通りです。

トイルが発生した場合は、作業者はその都度GitHub Issueを作成します。

例として、EKSノードのディスクサイズが減少した時のアラート対応を挙げます。Issueにはトイルであることを示すToilラベルと、そのトイルがどの程度の負荷であるかのVolumeラベルを付与します。

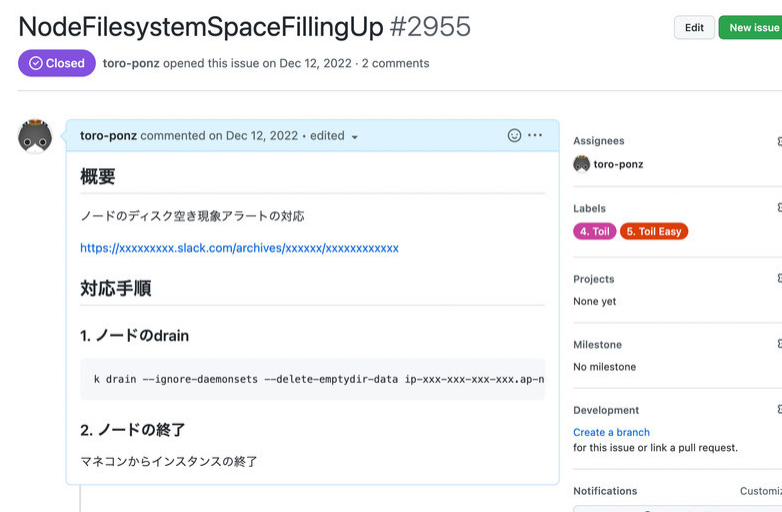


図2: トイル計測用Issue

Volumeは一時間未満で作業が完了する「Easy」、数時間を要する「Normal」そして数日以上必要な「Hard」の三種類があります。この区分は、Googleの公開している下記の記事を参考にしています。

参考: [Identifying and tracking toil using SRE principles](#)

Volumeは、可視化される過程でトイルの負荷を表現するトイルポイントに変換されます。それぞれ、Easyは1、Normalは3、そしてHardは9ポイントになります。Grafanaダッシュボードは、指定された期間の全体のポイント合計と、各人の負荷をグラフで可視化します。

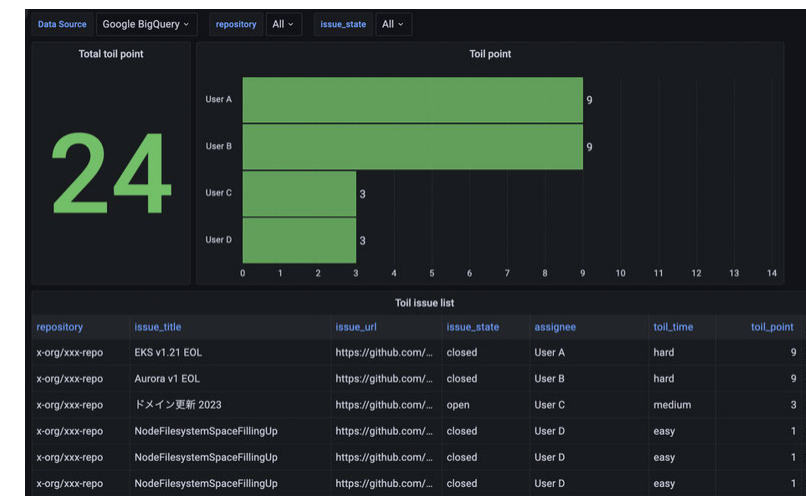


図3: トイル可視化ダッシュボード

SREチームでは、このダッシュボードを定期的を確認することで、トイルについての見直しを行っています。トイルの絶対的な量が多すぎないか、メンバー間での偏りが顕著でないか、トイルのVolumeが適切かなどの認識をすり合わせます。

01 明らかな作業の偏りがあれば他の人にオフロードできないか
02 検討しますし、頻繁に実施されているEasyなトイルがあれば
03 自動化に動きます。上記の図で言うと

04 NodeFilesystemSpaceFillingUpがそれにあたるでしょ
05 う。実際にはここまで単純なデータではないため、GitHub
06 のGUIで把握するのは難しいですが、Grafanaダッシュボー
07 ドにすることで日々のトイルが一目瞭然になりました。

08 所感

トイルの測定を始め、メインの目的であるトイルの把握以外
にも、いくつかの副次的な効果も得られました。特に、トイ
ルが発生した時にGitHub Issueを作成するという作業が追
加されることにより、トイル自体への負荷が若干高まりまし
た。今までは慣れている作業でほんの数分で終えていたもの
も、Issue作成の一手間が増えることで、思ったより頻繁に
発生していたり、面倒なタスクであると改めて気がつくきっ
かけになりました。実際に定期的に発生している
Prometheusのディスクアラートから、Thanos導入の検証
を始めたり、更なるトイル撲滅の動きにつながっています。

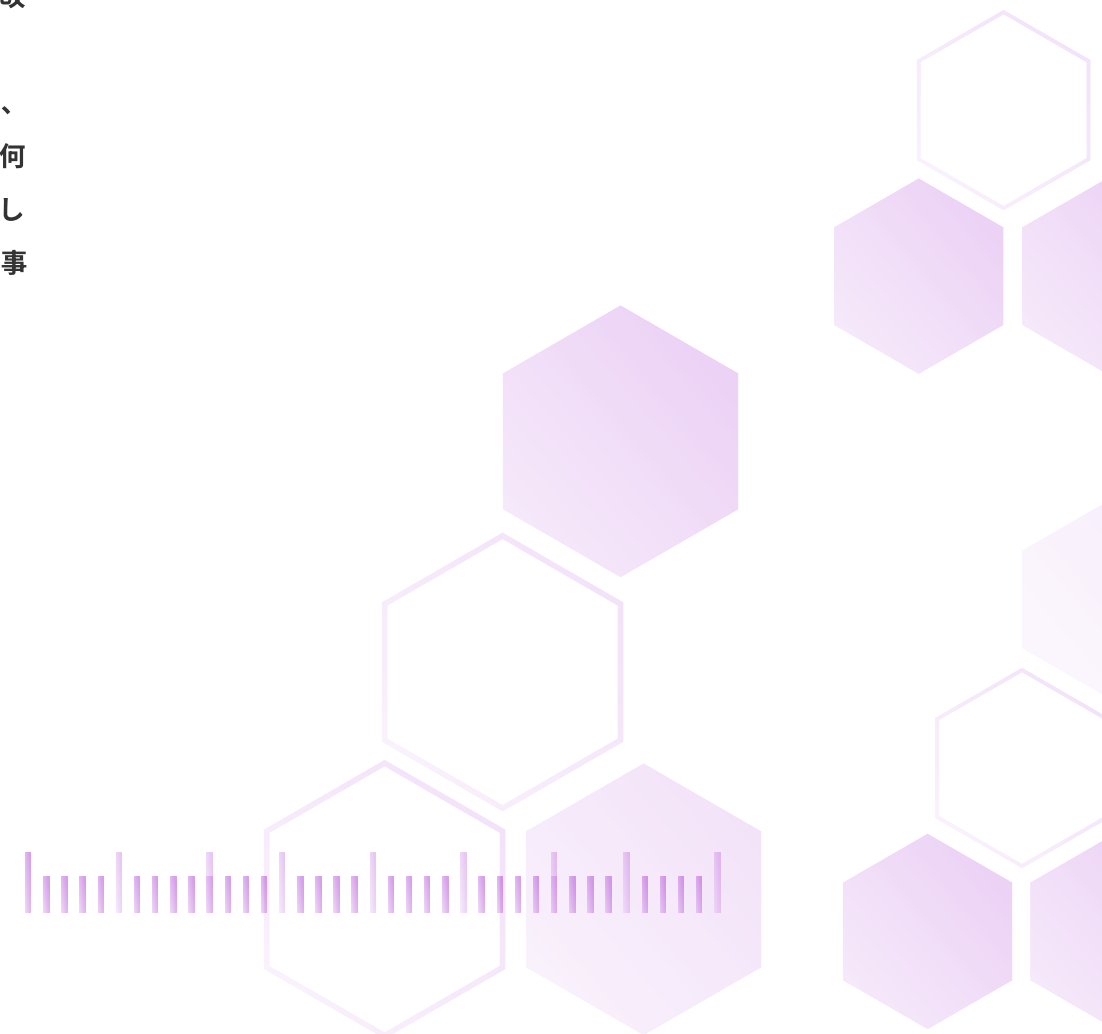
また、自分以外のSREメンバーが日々実施しているトイルに
も目が向くようになりました。組織が変化に強くなるため
には、システムや技術だけでなく人や知識も流動的でなければ
なりません。特定の作業者に知見が偏りがちなトイルを減ら

すことで、より新しい人が受け入れ易い、変化に強いチーム
になるはずです。Issueを介して知識の属人化を排除し、
チームとしてトイルに向き合う動きを今後より一層強めてい
きたいと思います。

おわりに

私たちのSREチームは、トイルについてはある程度減らせて
いると感じていましたが、いざ計測してみるとまだまだ改
善の余地がありました。

もちろん全てのトイルを撲滅することは理想論でしかなく、
費用対効果との天秤であるのはいうまでもありませんが、何
はともあれまず計測、現状把握です。もしトイルの削減はし
ているものの計測ができていない方がいましたら、この記事
がトイル計測の一助となれば幸いです。



Embedded SRE としてサービスに SLO を導入してみる

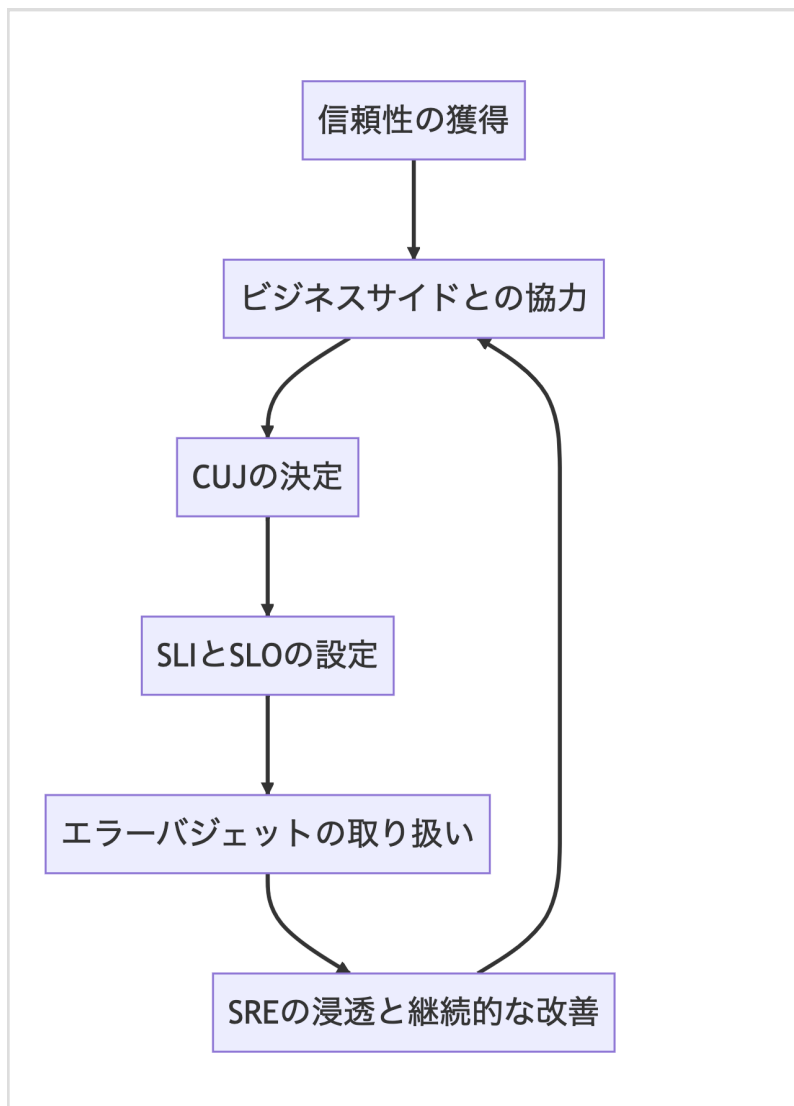
はじめに

横軸組織の SRG が Embedded SRE (Site Reliability Engineer) としてドットマネー by Ameba (以下ドットマネー) に参加し、SRE 文化の構築やサービス品質の向上を行うまでにやってきたことをご紹介します。

この記事ではあくまで個人的なやり方であり、多くのパターンで当てはまるやり方ではないかもしれません。そのため、手法の1つだと思っていただけると幸いです。

SRE を導入するきっかけ

SRE を継続して行うことでサービスの可用性と安定性が向上し、サービスの品質が向上します。SRE を導入することで可視化できないサービス品質を可視化でき、開発チームと運用チーム間の連携が向上し、問題の早期発見・解決が容易になります。



1. 信頼性の獲得と初期の取り組み

私は Embedded SRE として、サービス側の部署から信頼性を獲得するために Toil の削減やコミュニケーションを重視しました。これは私自身が SRE 初心者であり、サービス側が SRE のメリットを享受する前に疲れる可能性があるためです。

2. ビジネスサイドとの協力

サービス品質の維持にはビジネスサイドの協力が不可欠です。特にエラーバジェットが枯渇した後の対応は、プロダクトマネージャーの許可が必要です。

3. CUJの決定

ドットマネーでは、最もビジネスインパクトが大きい部分から CUJ を決定しました。今回は「ユーザーがサイトに訪れる → 商品ページ一覧を開ける → 交換ができる」という一連の流れを CUJ としました。

4.SLIとSLOの設定

今回のCUJに対して、SLIは「3秒以内に正常にレスポンスが返せる」と設定しました。また、怪しいリクエストは除外し、定期的に見直しを行います。

ドットマネーで現在利用している「ユーザーがトップページに訪れる」際の Datadog クエリを例にあげます。

Total Event

=====

```
service:cloudfront @http.ident:d-money.jp
@http.url_details.path:"/" -
@http.status_code:[400 TO 499]
host:"arn:aws:cloudfront::masked" -
@http.useragent_details.browser.family:curl
```

Good Event

=====

```
service:cloudfront @http.ident:d-money.jp
@http.url_details.path:"/" @http.status_code:
([200 TO 299] OR [300 TO 399])
host:"arn:aws:cloudfront::masked" -
@http.useragent_details.browser.family:curl
@duration:<3s
```

5.エラーバジエットの取り扱い

ドットマネーでは、エラーバジエットが枯渇した際の対応を「機能のリリースを禁止する」としましたが、特定の条件下では例外を認めます。

ドマネのエラーバジエット枯渇時のリリース規約

Aa リリース内容	リリース可否	外部会社関連
本番障害への緊急対応	可	<input type="checkbox"/>
信頼性回復のための改修	可	<input type="checkbox"/>
上記以外を目的とする機能リリース	可	<input checked="" type="checkbox"/>
上記以外を目的とする機能リリース*	不可	<input type="checkbox"/>

+ New

6.SREの浸透と継続的な改善

SREは終わりのない文化であり、定例での振り返りやサービス内での浸透が重要です。また、全員が参加するチャンネルで週に1回SLOサマリーを投稿するツールを作成しました。

<https://github.com/rluisr/datadog-slo-insufflate>



7.今後の展望

私はまだSREを完全に理解していませんが、導入から改善に努めることで知見が得られると考えます。今後はSREとビジネスインパクトの紐付けを可視化し、SREがビジネスに対して実際に影響があることを示したいと思います。

導入時の工夫

サービス側の部署から信頼性を獲得するためにToilの削減やコミュニケーションを重視しました。

Toil などを探し改善をする

サービスのインフラ構成を把握したり、デプロイフローなどを調査しているうちに Toil が見つかりました。

例えば、

- ・ 全くメンテナンスされていない EC2 と IaC
- ・ 過剰なリソース割当によるコストの無駄
- ・ 長時間かかるデプロイフロー
- ・ インシデント発生時の対応フローがない

など。

EC2 と Ansible、過剰なリソース割当によるコストの無駄、デプロイフローの改善は全て、EC2 から Fargate へ移行を行うことでサービス側へ数値として明確な成果を示すことができました。

コミュニケーション

ドットマネーでは毎日サーバーサイドの夕会や、週に1回の全体会議などがあるのでそこへ顔を出したり、IaC に対する PR への積極的なレビュー、（当たり前ですが）インシデントの対応などを行いました。

インシデント対応フローの確立

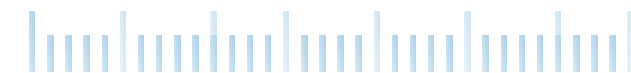
サービス品質を担保するためにもインシデント対応フローを確立し MTTR(Mean Time To Repair: 平均修復時間)を短縮することも SRE の1つです。そしてその文化を継続させなければなりません。インシデント対応フローのドキュメントは古くそれに沿ったフローでは動いていなかったためビジネスフローも含めて確立しました。

ここでは、オンコールの導入、インシデント対応フローの確立、Datadog Incident を活用した MTTR の計測とポストモーテムの運用に関する物事を進めてきました。

SRE 導入時の壁

自身の信頼性をサービス側の部署から得る際に Toil 削減の目的で EC2 から Fargate への移行を進めていました。その際に AWS Copilot という ECS ツールの検証、導入をしていましたが SRE の導入には新しいツールやプロセスが必要となるためそれらを導入・整備するのは手間がかかり、コストが増加する場合があります。

SRE はやることが多いため小さな成功体験を残しておき、組織内で SRE 疲れを感じた際に振り返りが出来る状態を作っておくと良いと思います。



SRE成熟度評価について

はじめに

メディア事業の横断SRE組織であるSRG（サービスリアビリティグループ）が、横断的にSRE推進するために開発したSRE成熟度評価の取り組みについてご紹介します。

取り組みの背景

物理的に、全プロダクトへEmbeddedするのは難しいので、横断的にSRE推進する方法を私たちは探していました。また、全体を俯瞰してみる為のデータや指標がない為、組織として効率

成熟度概要	Lv.1	プロセスが定義されておらず場当たりの状態
	Lv.2	チーム内でプロセスが定義、共有されている状態
	Lv.3	ベストプラクティスに基づいたプロセスで運用できている状態
サービスレベル目標	Lv.1	信頼性の定義がされておらず、SLOが設定されていない
	Lv.2	SLOが計測され、信頼性の数値としてチーム内で認知されている
	Lv.3	ベストプラクティスに基づいたSLO運用ができている
監視	Lv.1	監視についての要件やルールがなく、場当たりの監視になっている
	Lv.2	監視ルールが整備され、プロダクトの異常が適切に検知できる状態になっている
	Lv.3	ベストプラクティスに基づいた監視項目の設定、通知が行われている
インシデント対応	Lv.1	対応フローが定義されておらず、対応が属人化している
	Lv.2	対応フローが定義され、チーム全体で反復可能な状態になっている
	Lv.3	ベストプラクティスに基づいた対応フローが行われている
ポストモーテム	Lv.1	障害報告は書いているが、障害から学習できる状態になっていない
	Lv.2	障害を振り返り、対応プロセスやプロダクトの改善につながっている
	Lv.3	ベストプラクティスに基づいたポストモーテムの導入、振り返りが行われている
トイルの撲滅	Lv.1	運用作業と開発タスクが分断しており、運用作業の改善は場当たりに行われている
	Lv.2	運用作業の見直しや改善が計画性をもって行われている
	Lv.3	ベストプラクティスに基づいたトイルの計測や改善が行われている

図1. 成熟度概要

的にリソースを割くことができず、リスク管理においても後手に回ることが少なくありませんでした。そういった課題を解決するために、私たちはSRE成熟度評価を開発しました。

SRE成熟度評価とは

事業部全体を俯瞰しデータ化するために、能力成熟度モデル統合をベースに作成したものです。また、サービス信頼性の断層などを参考に必要項目をリスト化し、評価しやすくする為に極力シンプルにしています。

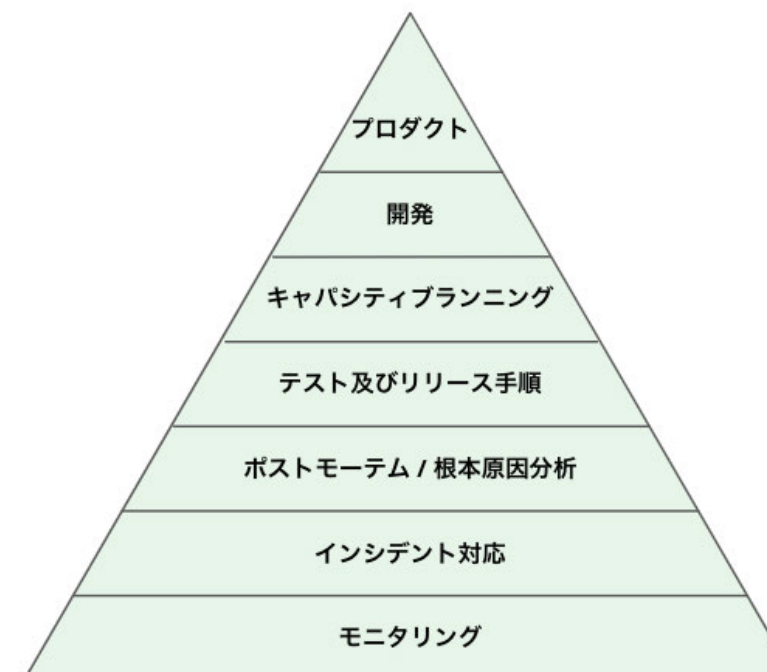


図2. サービス信頼性の断層

SRE成熟度評価によって何ができるのか？

SRE成熟度評価を活用することによって、横断的にSRE推進（Enablement含め）を行うことができます。また、自分たちの現在地を知ることができるので、改善計画を立てやすくなり、プロダクトにとっての理想状態へ近づくことが出来る様になります。

SRE成熟度評価の流れ

SRE成熟度評価は、大きく分けて4つのステップで実施します。

1. 準備
2. 評価と計画
3. 改善実施
4. 振り返り

1. 準備

SRE成熟度評価を実施するにあたり、SRE成熟度評価のコンセプトや活用フロー、Lv.3ガイドラインの説明をします。

- ・Lv.3ガイドライン
 - ・各項目のベストプラクティスを考える上での観点を質問にしたもの
 - ・各プロダクトの理想状態 = Lv.3

- ・各プロダクトにおける理想状態は違うので、全てを満たしている必要はない

2. 評価と計画

各項目のLv.3を参考にしながら、各項目の現在の成熟度レベルと理想状態の認識合わせを行います。各項目のLv.3は、下記になります（一部抜粋）。

1. サービスレベル目標

- ・ 定義
 - ・ SLO導入の目的と意義が、経営層にも認識されている
- ・ 計測
 - ・ ユーザーに近い位置から計測されており、SLOの悪化とサービス影響が紐づいている
- ・ 振り返り
 - ・ SLOを振り返る機会が定期的に設定されている

2. 監視

- ・ 要件定義
 - ・ 監視の目的が定義され、要件にあった監視ソリューションが選択されている

- ・ メトリクス
 - ・ メトリクスの取得頻度や保存期間が、適切に設定されている
- ・ アラート
 - ・ アラートメッセージに必要な情報が整理されている
 - ・ 送られたアラートログを元に、定期的アラート設定は見直されている
- ・ コスト
 - ・ 監視にかかっているコストが可視化され、定期的に見直しされている

3. インシデント対応

- ・ インシデントレベルと役割
 - ・ ユーザー影響に伴ったインシデントレベルが定義されており、インシデントレベル毎に業務時間外での対応方針やメンテナンスイン基準が定義されている
- ・ オンコール、検知とトリアージ
 - ・ 休暇や手当などのオンコールに対する補償が用意されている
 - ・ ユーザーの問い合わせよりも早く、インシデント発生を検知することができる
- ・ 情報共有、事後対応
 - ・ 必要に応じて、ユーザーに対してインシデント状況の報告が行えている

4. ポストモーテム

- ・ インシデント発生時
 - ・ ポストモーテムを書くかどうかの基準が定められている
- ・ 記載内容
 - ・ システム影響だけでなく、ユーザー影響まで記載されている
- ・ 振り返り
 - ・ ポストモーテム自体の過不足や書くことへの負荷について振り返りができている

5. トイルの撲滅

- ・ トイルの整理
 - ・ トイルに該当する運用作業が定められており、共通認識できている

プロダクト管轄		プロダクト	事業責任者	開発責任者	評価項目	2022/06	2022/09	2022/12	2023/03	2023/06	2023/9	
1	計画	子会社 A	Product A	田中	佐藤	サービスレベル目標	Lv.1	Lv.1	Lv.1	Lv.2	Lv.2	Lv.2
						監視	Lv.2	Lv.2	Lv.2	Lv.2	Lv.3	Lv.3
						インシデント対応	Lv.1	Lv.2	Lv.2	Lv.2	Lv.2	Lv.3
						ポストモーテム	Lv.1	Lv.1	Lv.2	Lv.3	Lv.3	
	実績					トイルの撲滅	Lv.1	Lv.1	Lv.1	Lv.2	Lv.2	Lv.2
						サービスレベル目標	Lv.1	Lv.1	Lv.1	Lv.1	Lv.2	Lv.2
						監視	Lv.2	Lv.2	Lv.2	Lv.2	Lv.3	Lv.3
						インシデント対応	Lv.1	Lv.2	Lv.2	Lv.2	Lv.2	Lv.2
2	計画	子会社 A	Product B	高橋	渡辺	サービスレベル目標	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1
						監視	Lv.2	Lv.3	Lv.3	Lv.3	Lv.3	Lv.3
						インシデント対応	Lv.2	Lv.3	Lv.3	Lv.3	Lv.3	Lv.3
						ポストモーテム	Lv.2	Lv.2	Lv.3	Lv.3	Lv.3	Lv.3
	実績					トイルの撲滅	Lv.1	Lv.1	Lv.1	Lv.2	Lv.2	Lv.3
						サービスレベル目標	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1
						監視	Lv.2	Lv.3	Lv.3	Lv.3	Lv.3	Lv.3
						インシデント対応	Lv.2	Lv.3	Lv.3	Lv.3	Lv.3	Lv.3
3	計画	子会社 B	Product C	伊藤	鈴木	サービスレベル目標	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1
						監視	Lv.1	Lv.2	Lv.2	Lv.2	Lv.2	Lv.3
						インシデント対応	Lv.1	Lv.1	Lv.2	Lv.2	Lv.2	Lv.2
						ポストモーテム	Lv.1	Lv.1	Lv.1	Lv.2	Lv.3	Lv.3
	実績					トイルの撲滅	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1
						サービスレベル目標	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1
						監視	Lv.1	Lv.2	Lv.2	Lv.2	Lv.2	Lv.2
						インシデント対応	Lv.1	Lv.1	Lv.2	Lv.2	Lv.2	Lv.2
ポストモーテム	Lv.1	Lv.1	Lv.1	Lv.2	Lv.3	Lv.3						
トイルの撲滅	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1	Lv.1						

図3. SRE成熟度評価シート

- ・ トイルに該当する運用作業が継続的に、可視化されている
- ・ トイルの改善
 - ・ 機能開発と同じ枠組みで、計画的にトイルの改善が行えている

認識合わせが行えたら、最後に改善計画を立てます。まずは、四半期単位での改善計画を作成し、そこからアクションアイテムとオーナーを整理していきます。また、監視、インシデント対応、ポストモーテムがLv.1の場合は、優先的に改善計画を立てることを推奨しています。

SRE成熟度改善計画

プロダクト名	事業責任者	開発責任者	SRG
Product A	田中	佐藤	柘植

現在のSRE成熟度

サービスレベル目標	監視	インシデント対応	ポストモーテム	トイルの撲滅
Lv.1	Lv.2	Lv.1	Lv.1	Lv.1

改善計画

年度	予定	改善担当者	ステータス
2023年度下期 2023/04-2023/09	ポストモーテムをLv.2へ改善予定	木村	進行中
	インシデント対応をLv.2へ改善予定	佐藤	開始前
2024年度上期 2023/10-2024/03	サービスレベル目標をLv.2へ改善予定	佐藤	開始前
	監視をLv.3へ改善予定	木村	開始前
2024年度下期 2024/04-2024/09			開始前
2025年度上期 2024/10-2025/03			開始前
2025年度下期 2025/04-2025/09			開始前

図4. SRE成熟度改善計画書

3.改善実施

他サービスでのナレッジを活用しつつ、各項目の成熟レベルを改善していきます。ポストモーテムやインシデント対応については、すぐに活用できるテンプレートも提供しています。



図5. ナレッジデータベース

4.振り返り

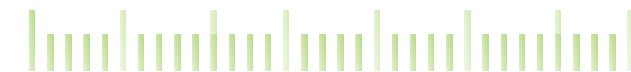
改善を実施したら、四半期または半期毎に振り返りを実施し、改善計画を見直します。まずは、四半期毎に振り返りを実施し、運用負荷が高ければ、半期毎にするのが良いです。

SRE成熟度評価によって得られたもの

データとして、事業部全体を俯瞰してみることができるようになったので、優先的にリソースを割くべきプロダクトや改善項目を決めやすくなりました。また、この取り組みを通して、把握できていなかった社内のプラクティスを知ることが出来たのも良かったです。

終わりに

SRE成熟評価によって得られたものは沢山ありましたが、改善の余地はまだまだあると思っています。この取り組みをより広く展開するためにも、スケール可能な体制作りの為の整備や各項目のLv.3の内容をブラッシュアップするなどして、品質向上していければと考えています。



Grafana OnCallの活用

はじめに

CAMやTappleが所属するIU管轄でオンコール対応で活用するツールについて紹介します。

背景

IU-SREでは、監視およびアラートツールにDatadogとPagerDutyを活用しておりました。

Datadog/Datadog APMでインフラストラクチャ、アプリケーションの監視を行い、PagerDutyを活用してサービス担当者へオンコール対応に必要な通知をしていました。

この記事では、PagerDutyからGrafana OnCallへ移行した理由、Grafana OnCallの活用について紹介します。

それぞれのサービスについては、下記からご確認ください

PagerDuty: <https://www.pagerduty.com/platform/devops/on-call-management/>

Grafana OnCall: <https://grafana.com/products/cloud/oncall/>

PagerDutyを利用していた理由

PagerDutyは、インシデント管理およびアラート通知に特化したプラットフォームです。

複数ある機能のうち下記を主に活用しておりました

- ・ SMS通知
- ・ スケジュールやエスカレーションポリシーの作成
- ・ SSO可能
- ・ Slack連携

これらを活用することで、社内のユーザーにアクセスを絞り、システムやアプリケーションからのアラートを受信しています。

また、週毎にシフトスケジュールを組み、適切なチームや個人に通知し、Slack上での情報連携によりシステム障害に対して早急な対応を実施することができていました。

Grafana OnCallを採用した理由

Grafana OnCallは、PagerDutyと同様にアラート管理およびオンコールスケジューリングを提供するプラットフォームです。オンコールのスケジュールとエスカレーションポリシーに基づいて通知を行うことができます。

PagerDutyに比べて圧倒的に安く活用することができます。

- ・ PagerDuty(<https://www.pagerduty.com/pricing/incident-response/>)
 - ・ \$49：1ユーザーあたりの月額
- ・ Grafana OnCall(<https://grafana.com/pricing/>)
 - ・ \$8：1アクティブユーザーあたりの月額

※その他、プランなどの詳細は、各リンクをご確認ください。

PagerDutyで利用していたSMS通知/スケジュールやエスカレーションポリシーの作成//Slack連携が利用でき、スケジュールされた担当者へ、電話での通知の機能を主に活用するため乗り換えることにしました。

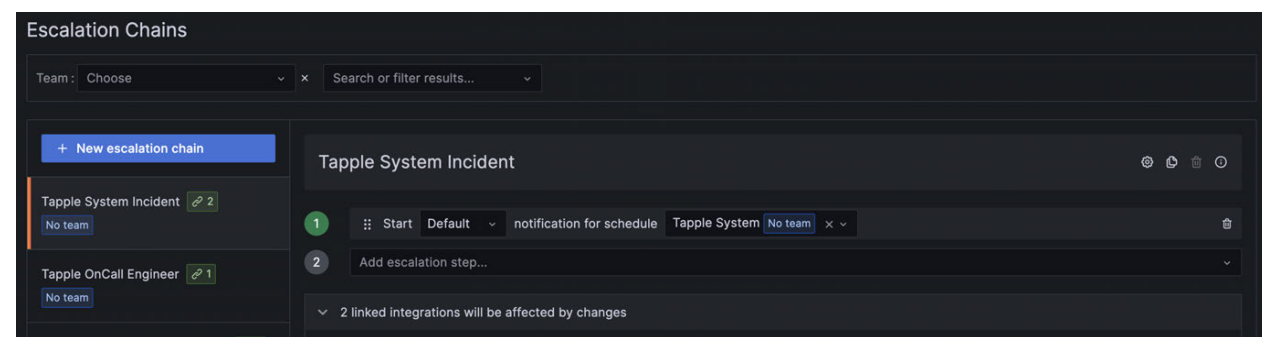
Grafana OnCallの活用

Schedulesで On Callの当番を管理してます。Schedulesの作成には Google.com のアカウントで作成した Googleカレンダー を利用しています。

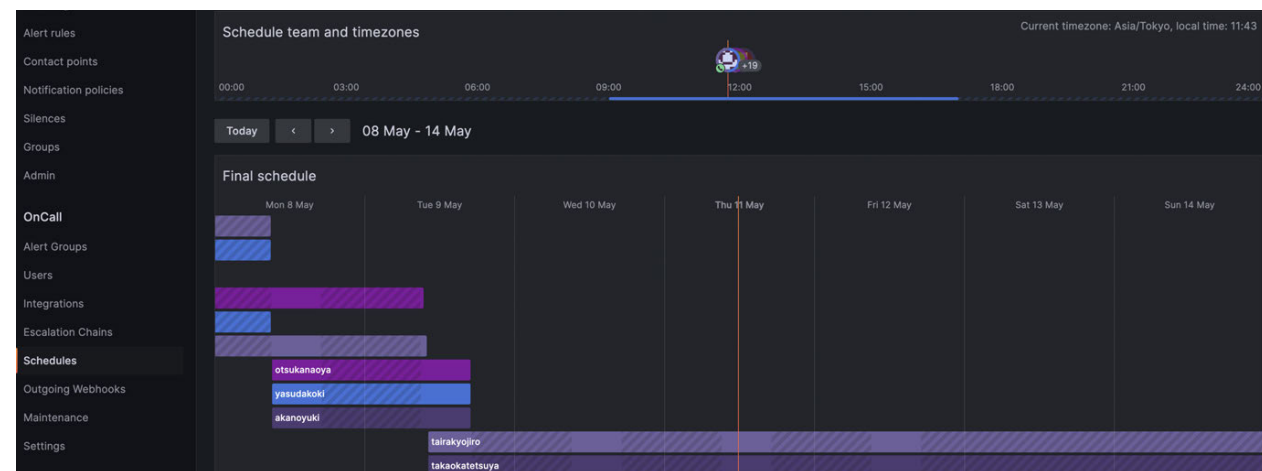
Escalation Chainを作成します。

On Callがトリガーされたときのエスカレーションフローを定義できます。Notify users from on-call schedule を設定すると先ほど登録した schedule に通知できます。

ここでは担当者へ電話とSlackの通知が届くシンプルな設定にしています。



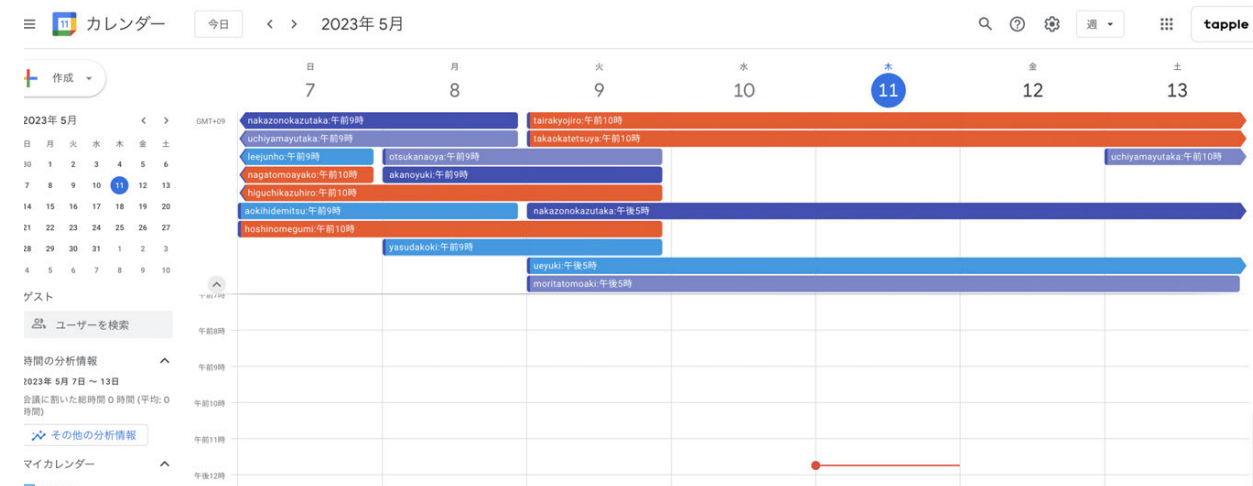
Escalation Chains



OnCall Schedule

Grafana OnCallでOn Call当番を管理する

Grafana OnCallの Schedule の更新は連携したGoogleカレンダーで行います。

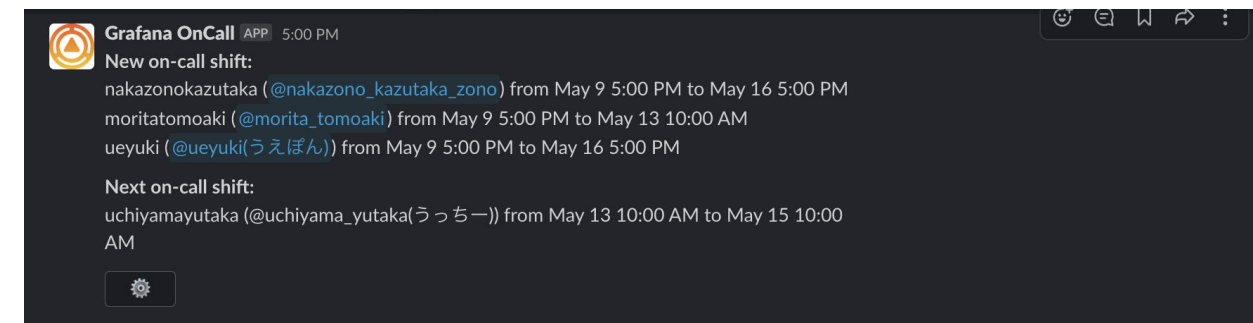


イベント名を **Grafanaユーザー名** にしてGoogleカレンダーに予定を登録すると、

Grafana Schedule側に当番として連携されます。

開発メンバーはカレンダーを確認して、担当を理解します。

また、担当が切り替わった時にSlack通知もしています。



移行中に遭遇した課題や解決策

また、当番用のスケジュールを管理するのにカレンダー連携が必須でした。

社内のGoogle Workspace内のカレンダーは外部共有が許可されておりませんでした。

Workspaceの共有設定を変更することはリスクがあると判断したため、google.comでカレンダーを作成し、作成したカレンダーに社内のGoogle Workspaceユーザーを編集権限で追加する方法で運用することにしました。

Googleカレンダーを活用してよかった点は、使い慣れたインターフェースのため、開発メンバー含めて担当の変更などの操作が容易でツールへの習熟が短期間で済みました。

おわりに

本記事では、私たちがPagerDutyからGrafana OnCallへの移行を行った経緯、プロセス、そしてその結果について解説しました。

オンコール対応は、サービス提供する上で顧客満足度の向上に不可欠なものとなっており、オンコール対応に必要なモニタリングとアラートツールの選定も大事になっています。

Grafana OnCallの導入により、PagerDutyを利用していたときと同様の機能を保ちつつ、費用を抑えることができました。

また、スケジュール管理が使い慣れたインターフェースであったことから担当の振り分け変更にもスムーズに対応できる環境を作ることができました。

最後に、本記事がオンコール対応のためのツール検討をしている方々にとって参考になることを願っています。



コラム



新卒1年目のSREが考えるSREの魅力とは

自分なりのSREを確立して組織に貢献したい

— 新卒入社して最初の配属先にCAMでのSRE業務を選んだ理由を教えてください。

岡：学生時代に参加したWeb企業の長期インターンでインフラ業務に携わり、面白いと思ったことがきっかけでした。そこでKubernetesやDevOpsについて学ぶうちに、その先にあるSRE（Site Reliability Engineering）という仕事がやりたいと思うようになりました。

SREは会社や組織によってそのあり方が全く違う職種で、「SREとは？」と10人に聞いたら10通りの答えが返ってくるぐらい、新しい職種と言えます。自分なりのSREを体現して実践し、それによって組織に貢献できるようになれば、すごく幸せで楽しいのではないかと思います。

サイバーエージェントに内定をもらった後は、株式会社CAMで内定者バイトとして働きました。とても充実した日々を過ごせたので、入社後の配属先もCAMを希望しました。

庭木：初めて会話をしたのは内定者バイトの受け入れ面談で

したよね。そこでインターン経験やKubernetesの知識があることを聞き、「GCPはある程度環境構築ができるのでAWSでもできるようにになりたい」という本人の希望を聞いて、それを実現できるようなタスクを用意しました。

CAMがOSSで公開している管理画面ツール「Viron」のステージング環境や本番環境をAWS CloudFormationで構築したり。IU管轄（※注）のタップルでAWS Fargate Spotを使ったコスト削減や、Amazon EBSへの移行に取り組んでもらったり。タップルが使っている構成管理ツール

Terraformにセキュリティチェックツールを導入してもらったりと、かなり幅広くやってもらいました。

※注：CAMやタップルなど6社からなる子会社連合の総称。スタートアップに特化した組織で、サイバーエージェントの専務執行役員 飯塚勇太が管轄する(2023年6月現在)

—確かに幅広いですね。その後入社して1年間働いた中で、自分の成長を感じた部分は？

岡：好きな技術にただ触れていた学生のころと比べると、コストの部分は特に考えるようになりました。例えば「自分が



やりたいこと」と「組織としてやるべきこと」の判別や、「既存サービスのコスト削減」と「新規サービスの迅速なリリース」のどちらに今は注力すべきか、といった優先順位も意識するようになったと思います。もちろん私一人で判断するのではなく、上長の庭木とすり合わせながら動いています。

庭木：彼とは「こういう風に作っていこう」という話であったり、コスト削減と新規の業務バランス、つまり先ほどの優先順位の部分であったり。今後どうなりたいかといった未来の話や、モチベーションが上がる仕事は何かというような話もしています。

岡：今後やりたいことや思い描く将来像について話せるのは本当にありがたいと思っています。何か間違ったことを言っ



たときも、頭ごなしに否定するのではなく、いったん受け止めた上でアドバイスをしていただけるので、心理的安全性が高いことをすごく感じています。

共通基盤としての負荷試験環境を構築しコスト削減に貢献

—CAMやIU、さらにはグループ全体のSREについてどういった展望を描いていますか。

庭木：IU発足以降、急に仲間が増えました。しかしIUの中でもCAMとタップルにはSREがいるけれど、SREがない会社もあります。そこで最初は設定周りやコスト削減のやり方など、CAMやタップルでうまくいっている事例を他の子会社に広めることで、IUとしての一体感を強めていくことを目指しました。

そうやってIUの仲間であるCAMとタップルで協力しつつ、お互いに幅を広げながら事業を加速させて、信頼性を向上していく。まずはこの取り組みを良い形に持っていければと思っています。ゆくゆくはSRG（メディア事業部横断のSREチーム）や他のSREチームとも相互に連携を強化していきたいですね。



もちろんCAMのSREとしてCAMの中でやるべきこともまだまだあります。多くのサービスを抱えるCAM特有の事情として、コスト削減が大きなインパクトになりやすい面があります。1つ1つのコストは小さくとも、掛け算すると大きなコストになります。例えば岡が先日実現したコスト削減の施策もかなり効果は大きかったですね。

岡：ありがとうございます。サービスを横断して使用できる独自の負荷試験環境を設計・構築したのですが、これは周囲の人からの反響も大きかったと思います。詳細はCAMのブログに記事を載せています。

これまではCAMが抱える多数のサービスにおいてその都度負荷試験環境を構築しており、大きなコストがかかっていました。そこで負荷試験ツールk6をKubernetes上で動かせる

01 k6 Operatorを活用し、全てのサービスに使える負荷試験環
02 境を構築したという取り組みです。コスト削減に加え、ナ
03 レッジを蓄積できるメリットも大きいと思っています。

04 –SREを目指す学生に向けてアドバイスがあればお願いしま
05 す。

06 岡：SREはフロントエンドからインフラまで幅広い知識を求
07 められます。そこが大変なところでもあり、魅力でもあると
08 思っていて、その気になれば10年単位で深掘りしていける職
種だと思います。

最初のきっかけとして例えば、HTMLのシンプルなページで
もいいので、自分でWebサイトを作り、自分で用意したサー
バーで公開し、それを運用してみるのが良いと思っています
。本当に基礎的な部分ですが、実際にやってみると全てが
完璧にうまくいくことはほぼないと思います。電源が落ちた
り、バグやエラーが発生したり、ファイルが破損したり。

問題を解決するために原因を調べたり、あれこれ試行錯誤す
ることが得がたい経験になるので、是非取り組んでみてほし
いですね。理想を言えば物理サーバーですね。それが難しけ
ればコンテナではなくVMで。その方が身に付くものは大き
いと思います。

庭木：今はRaspberry Piなどもありますし、昔よりは自前
で物理サーバーを構築するハードルは下がっているかもしれ
ませんね。



SREに求められるエンジニアとしての素養

■ ベストプラクティスだけでは解決できない課題 に向き合う

—今回なぜテクノロジーマップを作ろうと思ったのですか？

柘植：制作の目的は2つあります。「社内のSREチームの連携強化」と、「サイバーエージェントのSREを社内外に知ってもらうこと」です。

Service Reliability Group（以下、SRG）は、サイバーエージェントのメディア事業部において横断的にSREに取り組んでいる組織です。前身はメディア事業部のインフラ組織でした。AWSやGCPへのシフトが進むにつれ我々の役割も変わっていき、2015年に組織名をSRGに変更し本格的にSREに取り組んでいくことになりました。

SRG発足当初はGoogleのプラクティスに従いトイルの撲滅（※注）運動などを継続的に行っていたものの、抱える事業数や組織形態が異なる我々にとってはベストプラクティスではないことを実感しました。そこで改めて我々に適したSREを模索していった結果、横断組織の強みを活かせるようなSREが良いのではないかと考えるようになりました。

（※注）：トイル（Toil）とは「手作業、繰り返される、自動化が可能、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」のこと。GoogleのSREはトイルに費やされる時間を勤務時間の50%未満にすることを目指しているという（参考：[Google Cloud Japan ブログ](#)）

横断組織の強みはナレッジやプラクティスが蓄積しやすいところです。その強みを活かし、蓄積された知見を展開していくことで各プロダクトをサポートしたり、AWSのWell-Architected Framework（以下、ウェルアーキ）にアレンジを加えた独自のウェルアーキを作成し、ワークロード改善に役立ててもらおうといった活動を行いました。

これらの活動を通じ、メディア事業部内のプロダクトに関し



てはSREとしてある程度貢献できたと思っています。次なる一歩として、今後はSRG以外のSREチームとの連携を強化することで、より幅広く信頼性向上に繋がる貢献をしていきたいと考えました。これがテクノロジーマップを作った1つ目の目的です。

藤井：私は2019年にCyberZに入社し、バックエンドを経てSREチームに異動しました。着任時は「OPENREC.tv」のSREという位置付けでしたが、2021年の組織改編で他のプロダクトも見る横断的な立ち位置に変わりました。

GoogleのSREは1つの理想形ではあるものの、「我々の組織では現実的ではない」という部分も当然あります。その際に「相談する相手がない」「参考にできるようなナレッジを得られない」というのは大きな課題です。これは多かれ少なかれこのSRE組織でも抱えている悩みだと思います。

横の連携を作っていくためにも、まずは自分たちの仕事について発信していこうという取り組みを2022年から始めました。そうしてちょうど動き始めたところに柘植さんから声を掛けられたので、是非にということで今回のテクノロジーマップにも協力させていただきました。

01 柘植：実のところ、これまでも弊社でSREチームの横の連
02 携を強めるような取り組みがなかったわけではないと思いま
03 す。ただナレッジやプラクティスの共有を考えたとき、誰し
04 も「テイク」はしたいけれど「ギブ」のモチベーションを持
05 つことは難しいですね。

06 SRGの場合はもともと横断組織ゆえに「ギブ」にモチベ
07 ーションを持っているメンバーが多く、またそこを評価される
08 枠組みもありました。しかし通常の開発現場にそれを求める
のは無理があります。

とはいえ自分たちが見ている範囲だけにスコープを狭めてしま
うと、得られる情報の幅が狭まり、やがては今後チャレンジ
をしていくときの幅も狭めてしまうことになりかねませ
ん。



またSRE人材の採用を考えたとき、「サイバーエージェント
のSRE」をブランディングしていく必要もあります。しかし
弊社には多くのSREチームがあり「サイバーエージェントの
SRE」をひとまとめで語ることはほぼ不可能ですし、そもそ
もそれぞれの事業に適したSREがあるので全てが同じである
必要ありません。

しかし「サイバーエージェントのSREはこう」と言い切るこ
とはできなくとも、「各組織がどういったSREを実践してい
るのか」「どういうチームがあるのか」を発信することはで
きます。

まずは知ってもらい、知ってもらうことで「一緒に働きた
い」「話を聞いてみたい」といった繋がりができたら嬉し
い。これがテクノロジーマップを作った2つ目の目的です。

■ SREに求められる資質とは

— どのような人がSREに向いていると思いますか？

藤井：SREの業務は「人」が大きく関わってきます。SREを
組織に根付かせるためには文化の醸成が重要な要素になるた
め、チームや人を考慮しつつ動いていく必要があるからで
す。そのため技術だけでなく、組織の文化や開発プロセスに

も目を向けられる視野を持つ人が向いていると思います。言
葉にするのは簡単ですが、なかなか一筋縄では行かないのが
現実です。組織の数だけ正解があるところもSREが難しいと
いわれる要因の1つだと思います。

柘植：SRE人材が欲しいという企業は多いものの、「採れな
い」「育成方法がわからない」という悩みを多くの企業が抱
えています。実は採用する側も「求めているSREエンジニア
像」をうまくイメージできていないのが実情かもしれませ
ん。

私個人の考えとしては、「サービスに貢献したい」「サービ
スを良くしたい」という目線が大事だと思っています。単に
技術をやりたいという形だとギャップが生まれやすい。事業
責任者に対して提案をする場面もあるので、事業に目線を合
わせて貢献するマインドが求められます。

プロダクトへの貢献に主体性を持ってアクションをすること。
目線を事業に合わせた上でミッションを設定すること。
その技術を選んだ理由を自分の言葉で話せること。これまで
SRE業務に従事してきて、特に重要だと感じたのはこうい
った資質です。

SRE (Site Reliability Engineering) は日本語に訳すと
「サイト信頼性エンジニアリング」です。信頼性というワー

ドが一番大切で、その信頼性を得るためには人や組織から信頼されることも必要だと思っています。

藤井：個人的にはSREという職種は「SREかバックエンドか」という分かれ道の先にあるとは考えていません。バックエンドの先にSRE的な考えもあると思っています。これはバックエンドからSREになった自分自身の経験からくる考え方もかもしれません。

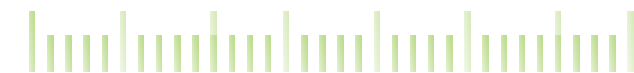
インフラからサービスに関わりたい人が集まってSREとして動くという流れがあり、バックエンドにもSRE的な考え方をを持った人がいる。SRE的な思考をSREエンジニアだけに留まらず、バックエンドやインフラにも広めていければ、結果としていい状態を作れるのではないかと思います。その意味でも実際に手を動かしてサービスに貢献するという部分は私も非常に重要なところだと思っています。

柘植：いずれはSREもDBRE (Database Reliability Engineer) のように、特定の技術領域に特化した形になっていく可能性はありますね。藤井くんが今言ったように、バックエンドやインフラ側の人たちがある程度SREのベース部分を担ってくれるようになれば、必然的にSREの役割も変化していくことになるので。

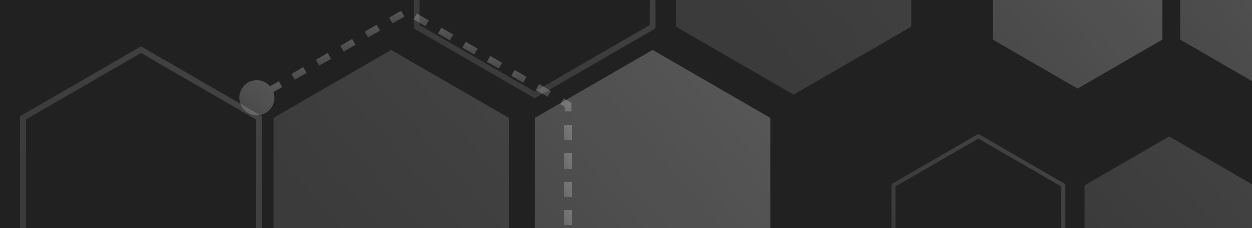
藤井：CyberZのSREチームに関して言うと、SREの人数は増えないにもかかわらずSREが見るプロダクト数は増えているというのが現在抱えている課題です。その解決策になり得る形として期待しているのが、Embedded SREという関わり方です。

Embedded SREとは、簡単に言えば「開発チームに組み込まれた (Embedded)」SREです。開発チームに入ってチームと一緒に手を動かしつつ、SREとしての役割を果たしていくという形です。

最近SREをやりたいという学生も増えてきましたが、学生である程度の規模のサービスの運用経験がある人はほとんどいません。SRE志望だけど運用は未経験、でも開発経験はあるというようなスキルセットの場合、Embedded SREは良い選択肢になり得ると思います。柘植さんもおっしゃっているように、「サービスを良くしたい」という目線をSREが他のエンジニアと同じように持つことで、開発と運用のコラボレーション、いわゆるDevOpsが実現でき、よりよい開発組織、ひいてはよりよいサービスの提供に繋がると思います。



編集



Member

編集

柘植 翔太 (メディア統括本部)
藤井 貴大 (株式会社CyberZ)
庭木 勝也 (株式会社CAM)
岩立 稜佑 (株式会社CA GameFi)
岡 麦 (株式会社CAM)
岩永 勇祐 (株式会社AbemaTV)
伊藤 崇洋 (株式会社アプリボット)
吉岡 賢 (株式会社サムザップ)
田口 雅教 (AI事業本部)

佐藤 恒平 (技術人事本部)
関本 育久 (技術広報)
北村 瑠美 (株式会社CyberZ)
田口 佳澄 (株式会社CAM)
武田 佳那子 (SGE統括本部)
高橋 佳那 (AI事業本部)
田爪 裕子 (AI事業本部)

デザイン

植田 祥子 (技術広報)
柴 尚子 (Design Factory)
山田 彩加 (Design Factory)
横山 恵 (Design Factory)

SRE Technology Map 2023

公開日 2023年 6 月 1 日
公開元 株式会社サイバーエージェント
〒150-0042 東京都渋谷区宇田川町40-1 Abema Towers
© 2023 CyberAgent, Inc

