



ぐるなび

『月間4100万UUの ぐるなびがレガシーから脱却し、 ハイブリットコンテナアーキテクチャで モダン組織へ変革した話』

株式会社ぐるなび CTO 岩本 俊明
New Relic株式会社 清水 毅



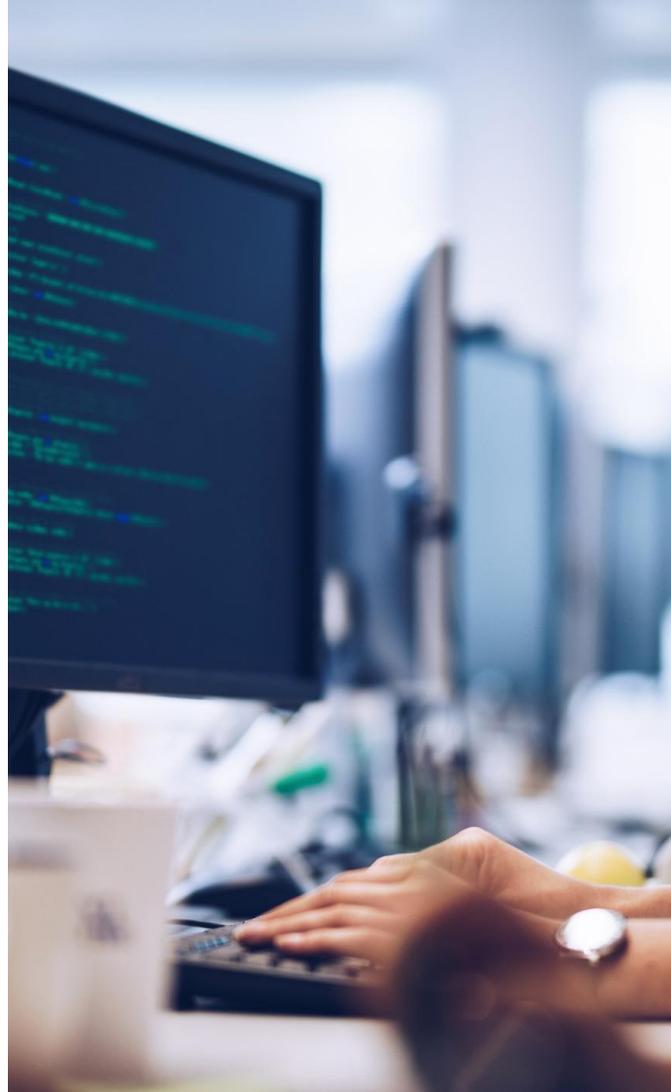
対象者とゴール

対象者

- **モダン開発チームへの変革**について知りたい方

ゴール

- **モダン開発チーム**とはどういうことなのかを知る
- **ぐるなび**社の事例を知り、
DevOpsを進める一歩を踏み出せる



Agenda

1. New Relic

『DevOpsを実践するモダン開発チームとは』

2. ぐるなび（インタビュー）

『月間4100万UUのぐるなびがレガシーから脱却し、
ハイブリットコンテナアーキテクチャで
モダン組織へ変革した話』

DevOpsを実践するモダン開発チームとは

New Relic株式会社 清水 毅



Tsuyoshi Shimizu

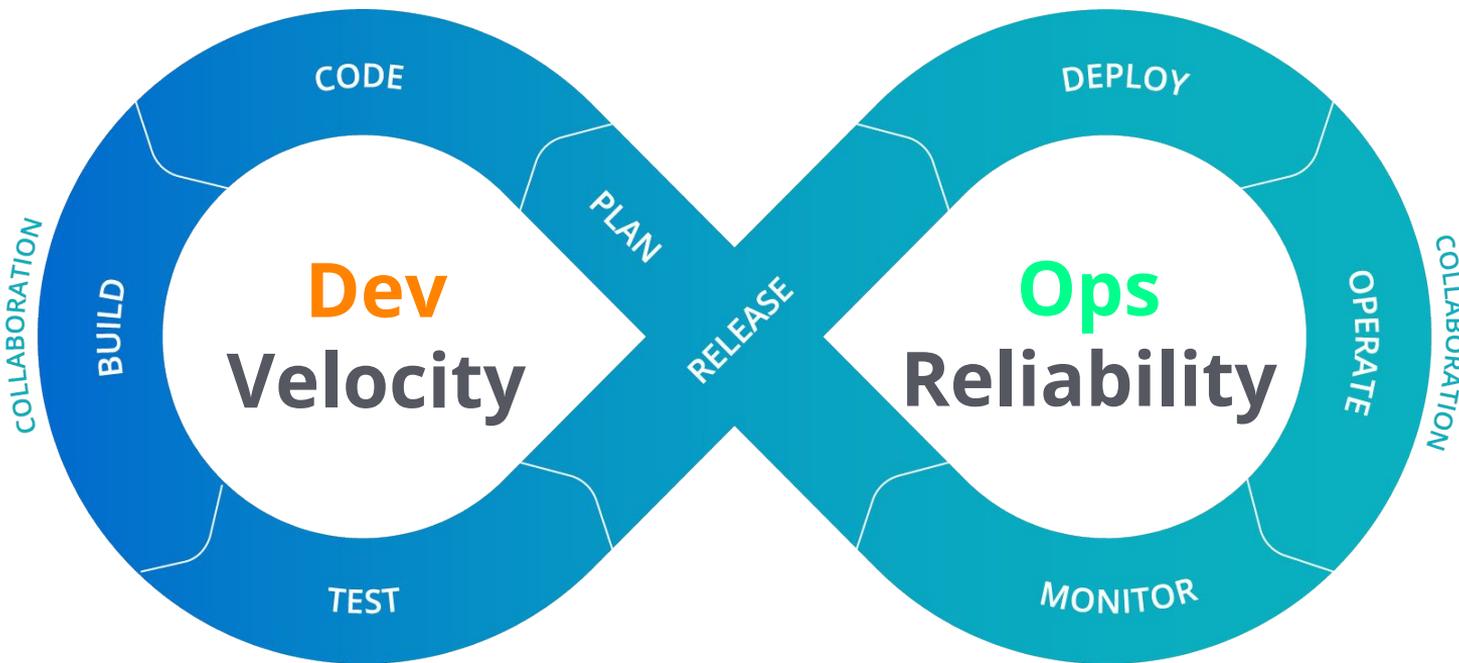
Package Vender & Cloud Vendor

- CRM
 - Java *Developer*
 - Performance and Security Specialist
- E-Commerce
 - SaaS *SRE*
- SaaS Solutions Architect
 - Architecting
 - *DevOps & SRE Support*

New Relic

- *DevOps Consulting*

ソフトウェアライフサイクルとDevOps

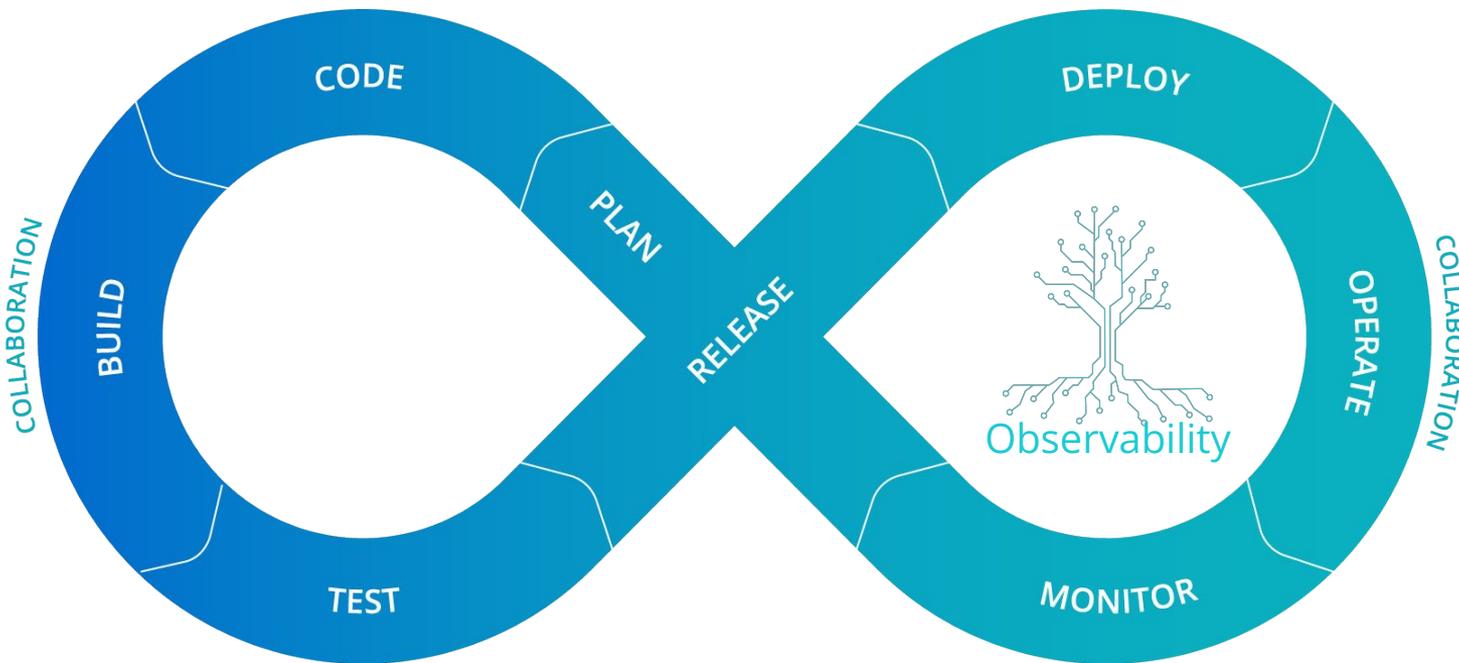


開発



運用/SRE

運用/SREのオブザーバビリティ

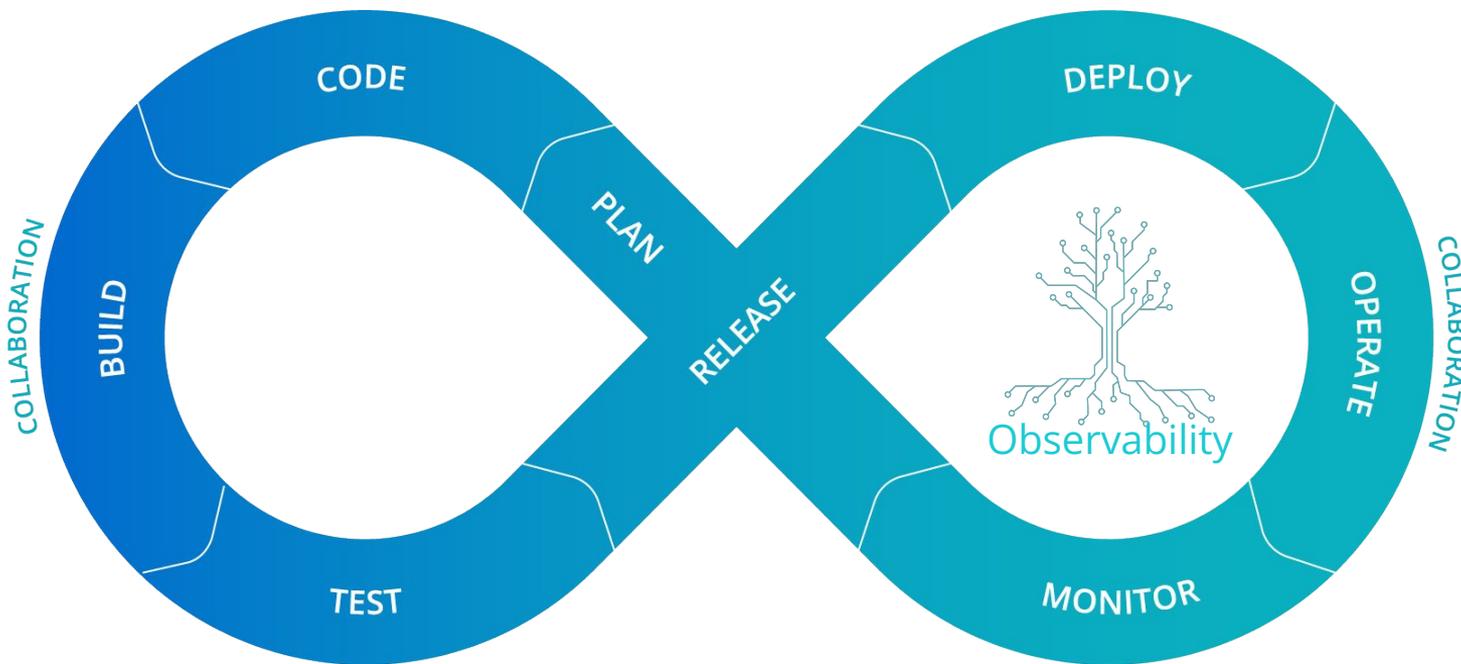


開発



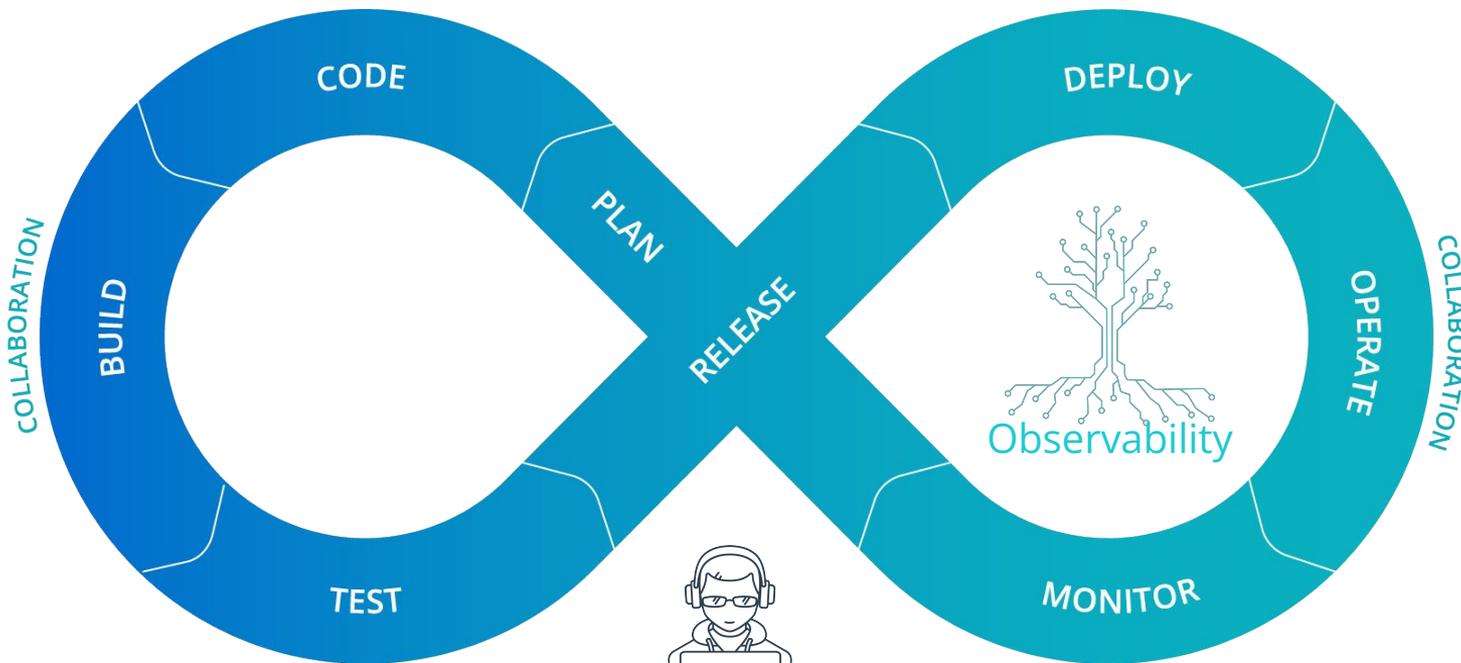
運用/SRE

開発者とSREのコラボレーション



開発 運用/SRE

開発者のためのオブザーバビリティ①運用段階



開発

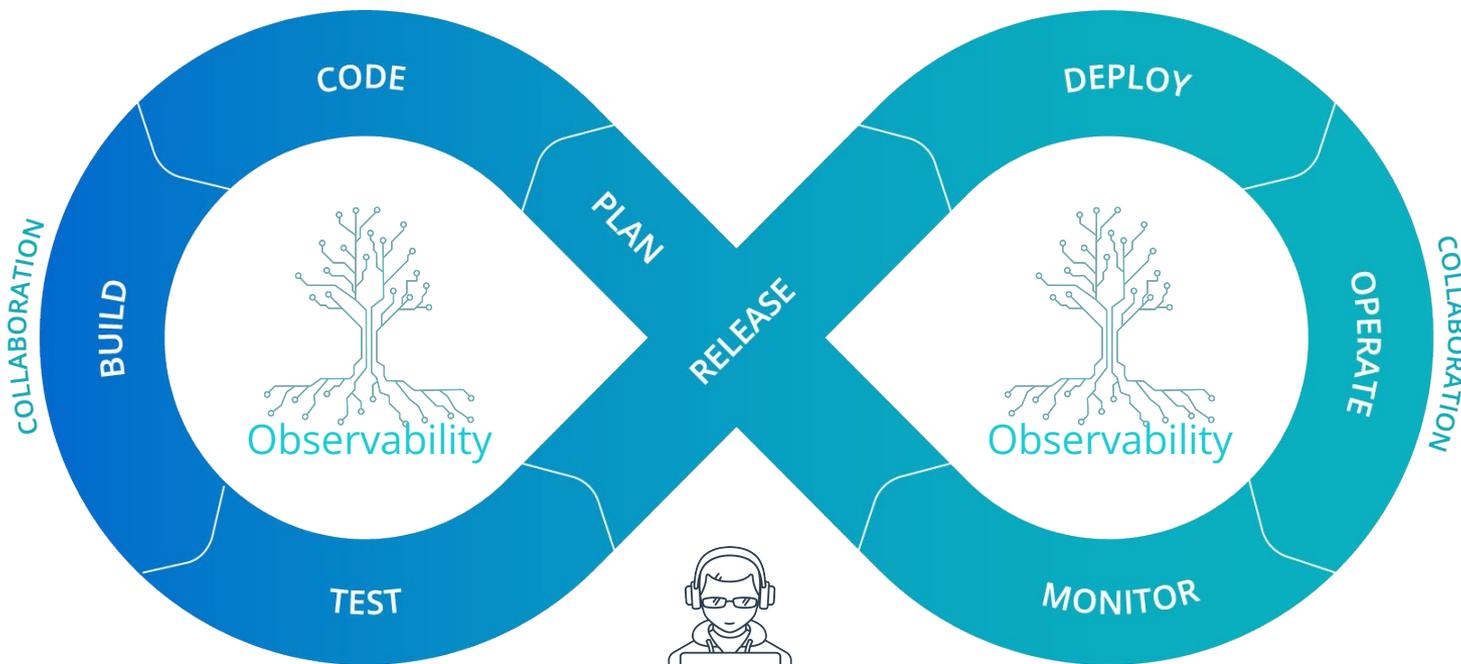


運用/SRE



Security

開発者のためのオブザーバビリティ②開発段階



開発



Biz



運用/SRE



Security

Building Microservices 2nd Edition

- 10章: 監視から**オブザーバビリティ**へ
- 11章: **セキュリティ**
- 14章: **マイクロフロントエンド**
- 15章: **疎結合**と**コンウェイ**～組織とアーキテクチャの関係

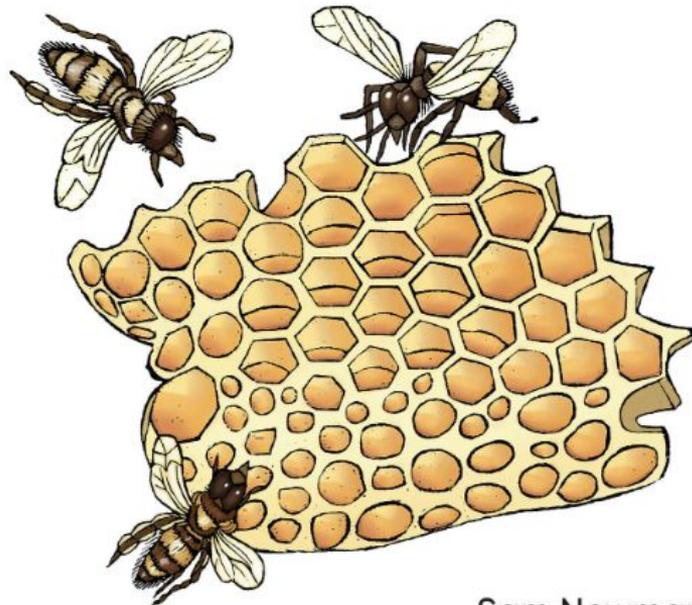
- **Amazon**
 - Two-Pizza Teams
- **Netflix**
 - Full Cycle Developers
- **Spotify**
 - Spotify Model

O'REILLY®

Building Microservices

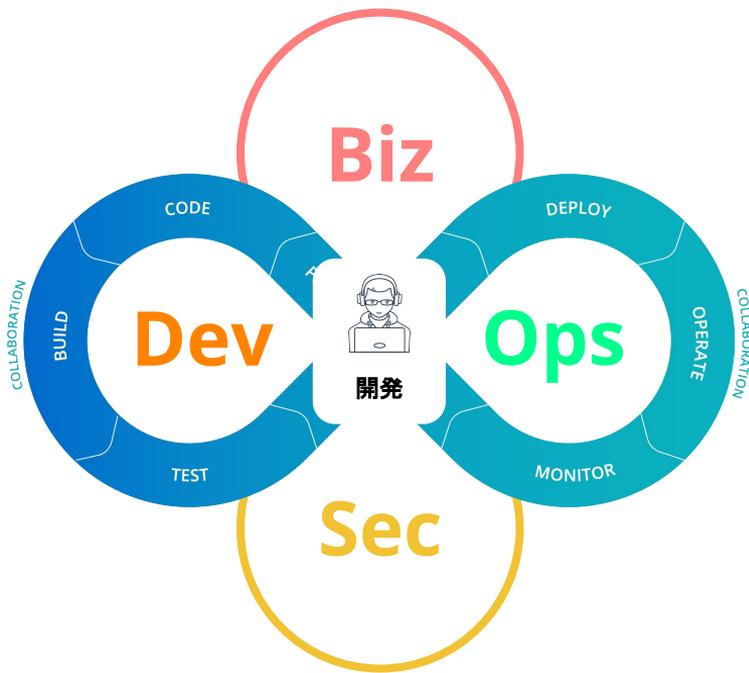
Designing Fine-Grained Systems

Second
Edition

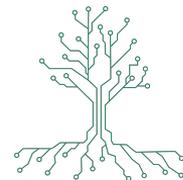


Sam Newman

DevOpsを実践するモダン開発チームとは



フルサイクルに
開発チームが中心となって活動する



Observability



モダン
アーキテクチャ



モダン組織

オブザーバリティを備えた
モダンアーキテクチャに
フルサイクルなモダン組織が実践



ぐるなび

インタビュー

株式会社ぐるなび CTO 岩本 俊明

New Relic株式会社 清水 毅

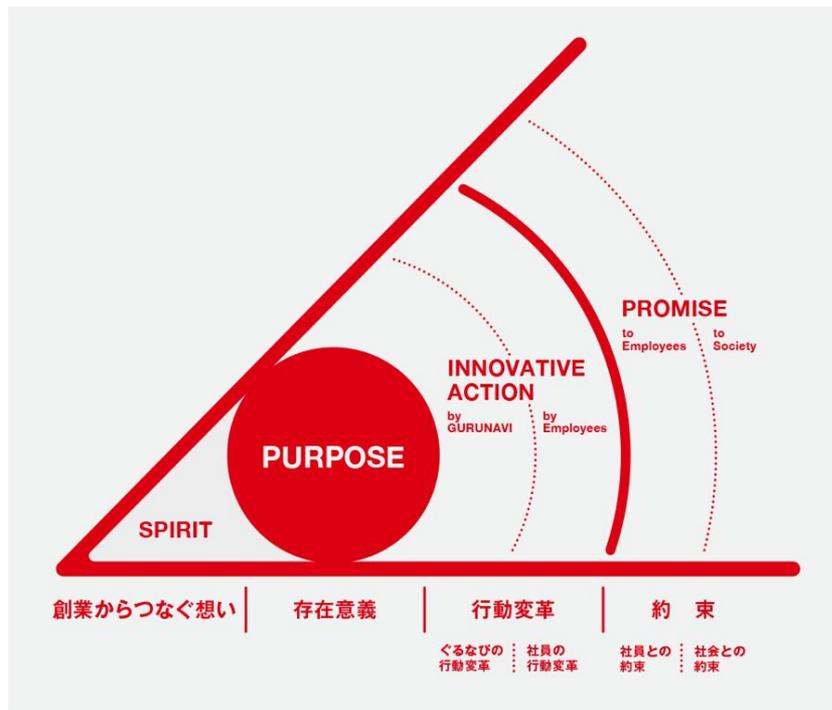
創業からつなぐ想い SPIRIT

日本の食文化を守り育てる

存在意義 PURPOSE

食でつなぐ。人を満たす。

私たちぐるなびは食の可能性を信じ、
世界中のヒト・モノ・コトをつなげ、
人々が満たされる場を創出します。



ぐるなびの持つ「強み」をさらに強固なものに。



人的サポート体制と
飲食店ネットワーク

ぐるなび Rakuten

楽天との
資本業務提携



2,271万人の
会員ユーザー
(2022年7月1日時点)

RED
RYORININ'S
EMERGING
DREAM II-35

今年の一皿

“食”を軸とした
高いプレゼンス



ぐるなび 台帳



ぐるなび FineOrder

店舗の
業務支援ツール



情報の信頼性に基づく
グローバル企業との連携

Toshiaki Iwamoto / 岩本 俊明 株式会社ぐるなび CTO

自社の全プロダクト・サービスにおける技術戦略の責任者としてサービス構築・品質向上に最適な技術の選択、意思決定を行う。また、在籍する Engineer 全体に対して技術的なビジョンを示し、エンジニア組織を牽引。

昨年の大規模コンテナ移行・クラウドマイグレーションプロジェクトを成功。2022年7月 CTO 拝命





Q. コンテナ移行プロジェクト発足のきっかけや目的は？

A. コンテナ移行プロジェクト発足のきっかけや目的は？

➤ Dev

- プログラムが**複雑**で開発効率が悪く、改修コストがかかりすぎる
- インフラ保守費用が最適化されていない為、費用対効果が悪い
- レガシーなアーキテクチャが**エンジニアの成長**の妨げになる
- **ビジネスロジック**が複雑になりすぎている

➤ Biz

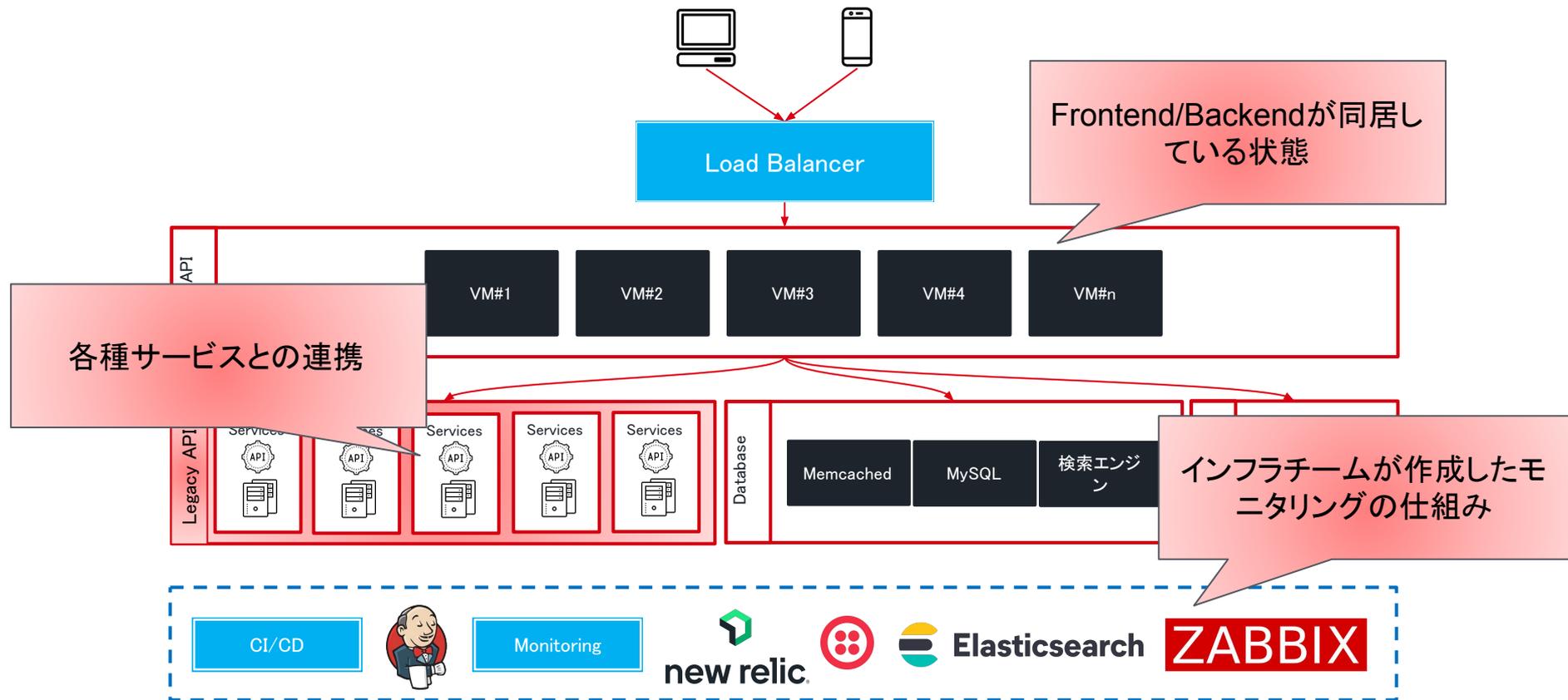
- 表示スピードを**改善**したい
- 追加したい機能が**実装**できない

サービスの成長の為にBiz/Devが協力して進めることが重要

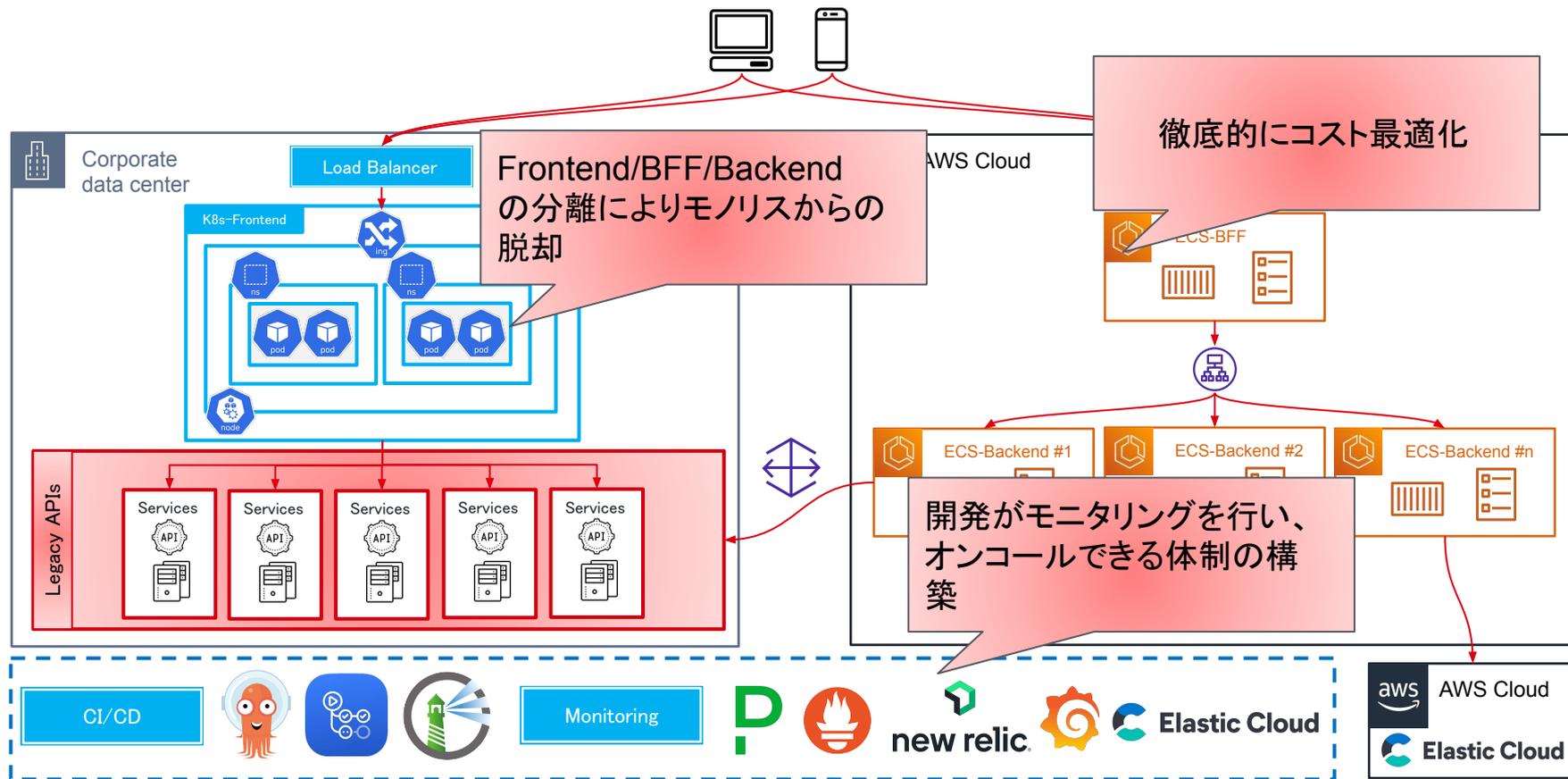


**Q. 移行手順を簡単に
教えていただけますか？**

A.移行手順 GURUNAVI Architecture (移行前)



A.移行手順 GURUNAVI Architecture (移行後)





Q. この大規模のプロジェクト 成功の鍵は？

A. この大規模プロジェクト、成功の鍵は？

➤ 過去を捨て去る

- 同じ仕様を綺麗にしたところで限界が来る。実現するためには過去技術負債を持っていくことができない。
- 現在のビジネスロジックから作るのではなく、Bizと一緒に考え1から作る。

➤ モダン開発組織の立ち上げ

- モダン開発を実現できるメンバーをアサインする。
- 旧システムはリリースまで並行稼働する。旧システムの運用と機能リリースをするチームと、新機能のみに集中するチームを完全に分けて新組織を作る。既存案件に干渉されず進めることができる。
- 旧システム担当者は新システムへの影響をシェアすることで、問題を未然に回避できる。

➤ 技術習得期間もスケジュールに入れる

- 当初はコンテナもAWSも知識ゼロ、Elasticsearchも初めてというエンジニアチームだった。
- Bizは要求整理・要件定義を実施する間に、エンジニアは技術を習得する。



**Q. エンジニアの働き方は
どのように変化しましたか？**

A. エンジニアの働き方はどのように変化しましたか？

➤ フルサイクル開発者の実現

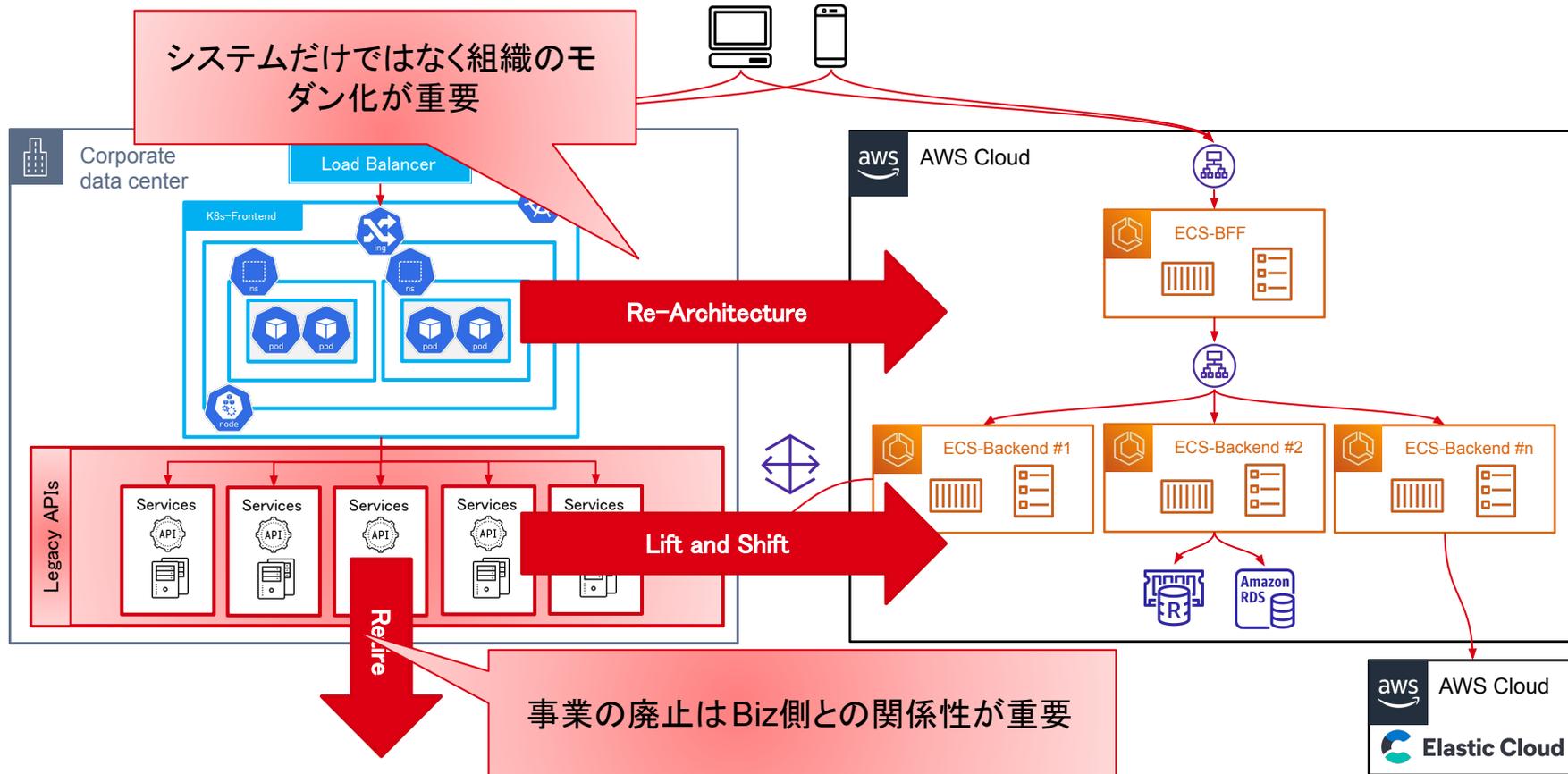
- インフラチームは運用(障害対応・モニタリング)に参加しない。
- 開発チームが開発・運用・改善(キャパシティプランニングやコスト最適化)を担当する。
- **Observabilityは超重要**。開発者は自分たちで運用できなければいけない。開発者フレンドリーな「**New Relic**」を活用することでサービスの定常状態を維持できるようになる。
- オンコール体制を組むことで開発チーム内の **運用ノウハウが蓄積される**
- フルサイクルに担当することで、**サイト品質が向上し・開発効率も向上する**

➤ Bizとの関係性

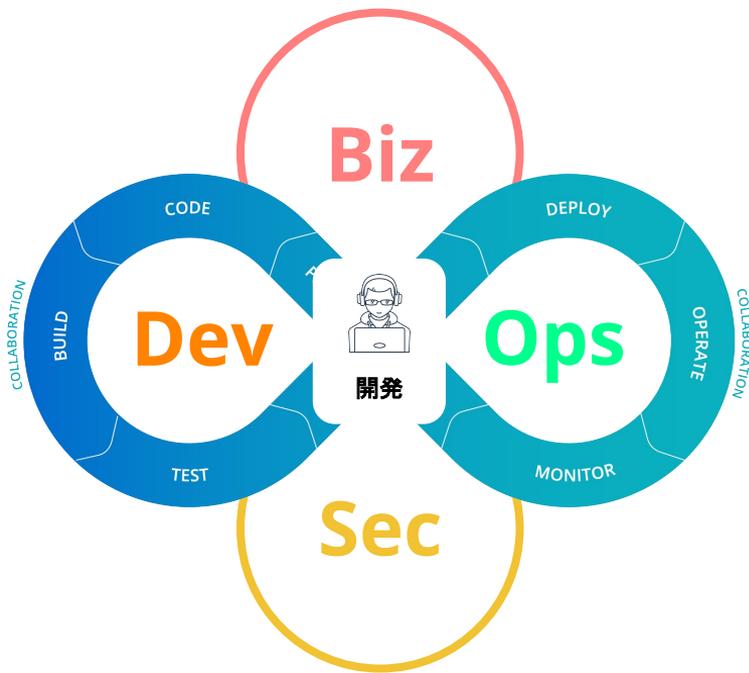
- 事業を理解した開発ができるようになる。無駄なものを作らず、**ユーザ満足度の高いサービスをいち早くリリースが可能になる**
- Biz側との関係性が良くなることで、コミュニケーションが活発になる。**エンジニアが事業理解も深まりプロダクト駆動開発を実現できればMVPリリースも可能になっていく**

サービスの成長の為にはBiz/Devが協力して進めることが重要

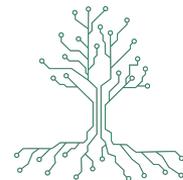
GURUNAVI Architecture (今後)



DevOpsを実践するモダン開発チームとは(再掲)



フルサイクルに
開発チームが中心となって活動する



Observability



モダン
アーキテクチャ



モダン組織

オブザーバリティを備えた
モダンアーキテクチャに
フルサイクルなモダン組織が実践

