



# VPoEが今すぐ着手すべき AI時代の観測データ戦略

~ New Relicで始める開発組織の生産性改革~

技術統括 カスタマーサクセス部 担当部長  
東 卓弥



東 卓弥  
あずま たくや



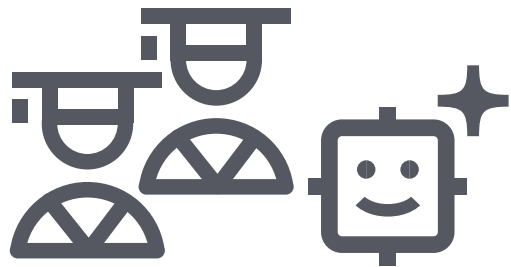
技術統括 カスタマーサクセス部  
担当部長

- 開発経験 20年
- チームリーダー / マネージャー経験 10年
- Olly製品の社内啓蒙活動

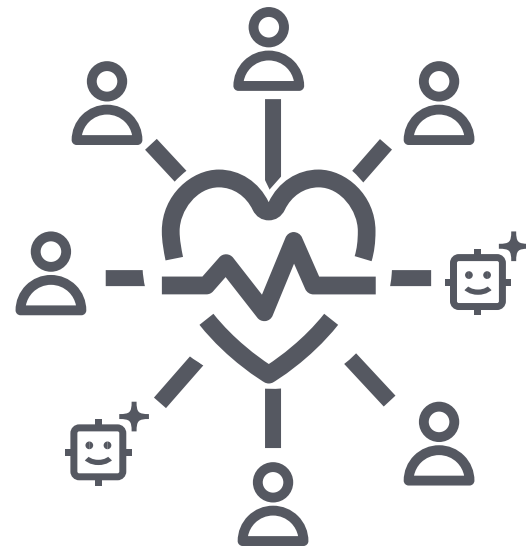


# 持ち帰っていただきたいヒント

“知識/経験が有利な組織”



“誰でも戦力になる組織”



# 誰でも戦力になる組織のポイント

- 1、Jr. エンジニアでも迷わない観測データ基盤を作る
- 2、事前知識がなくても筋の良い分析ができるデータを用意する

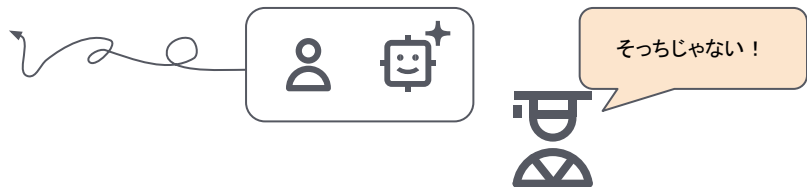
# サービス分析 を行う時の悩み

# お客様から聞く、Jr.エンジニアのAI活用の悩み

## 分析の初手で間違ってしまう

サービスのことをよく知らない新参のエンジニアからすると、データの在処さえわからない。

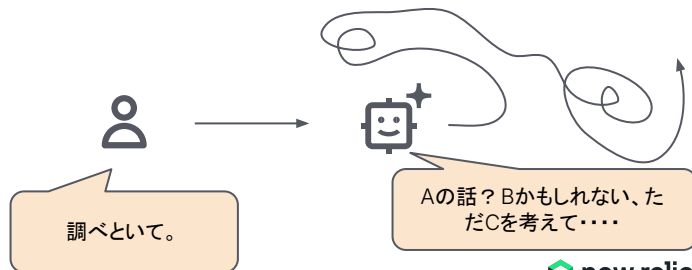
まず何を確認したら良いかわからないので、知っている知識を元に手探りで動いてしまう。



## 知識がないとAIも使えない

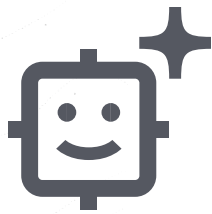
AIは、"知識が豊富だけど自分から動いてくれない優等生"

知識があるからこそ、やってほしいことを正しく伝えないと様々な知識を引っ張り出して進んでいる風になってしまう。





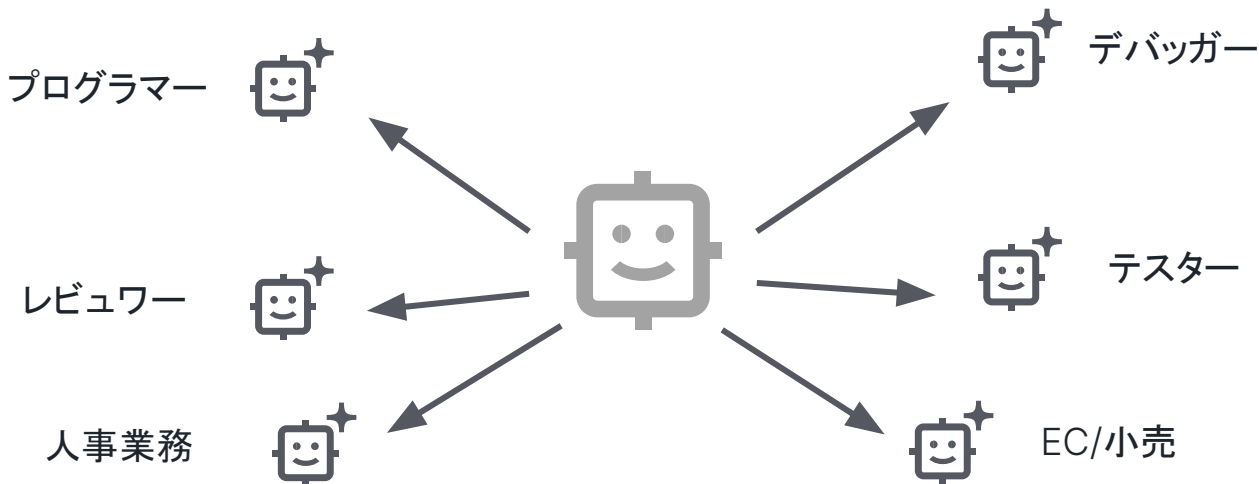
# AI vs スーパーエンジニア



# 多様化するAIの活用方法

Generative AIによるコーディングや分析はIT業界には急速に広まり、ほとんどのお客様が利用している状況。

ただし、完璧に1つのAIが業務をこなすことは難しく、専門的にタスクを行う "サブエージェント" を定義したり、プロンプトエンジニアリング / コンテキストエンジニアリングによって AIを賢く使う方法が考えられてきています。

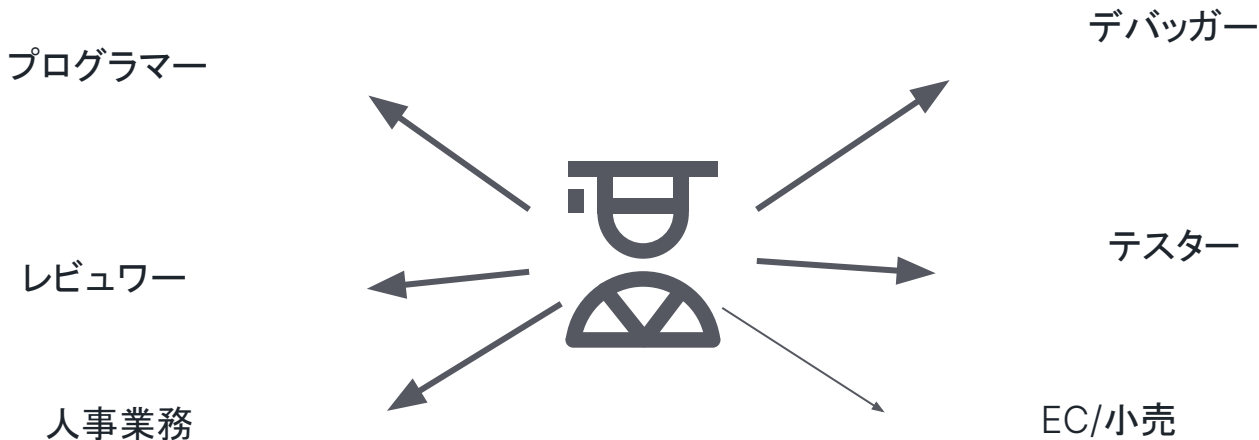




# 活躍するスーパーエンジニア

創業から在籍しているCTOやリードエンジニア、業務上の難題をクリアしてきたシニアエンジニアは AIのような並列作業はできないものの、多くの顔 / 技術を持っています。

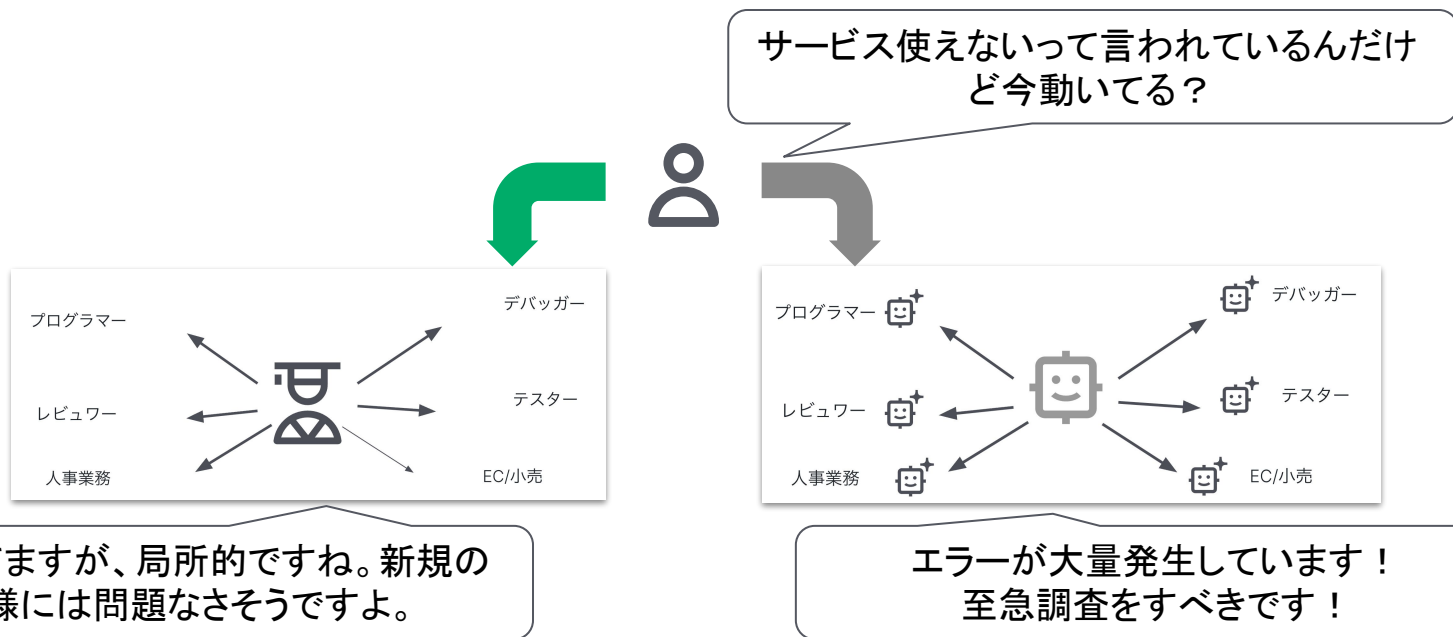
得意分野/不得意分野があったとしても、経験を生かすことで開発スピードや障害対応スピードが 他のメンバーより早い、神技が使えます。



# 本番環境でスーパーエンジニアにならないAI

AIは知識が豊富で、大抵のことは答えられます。

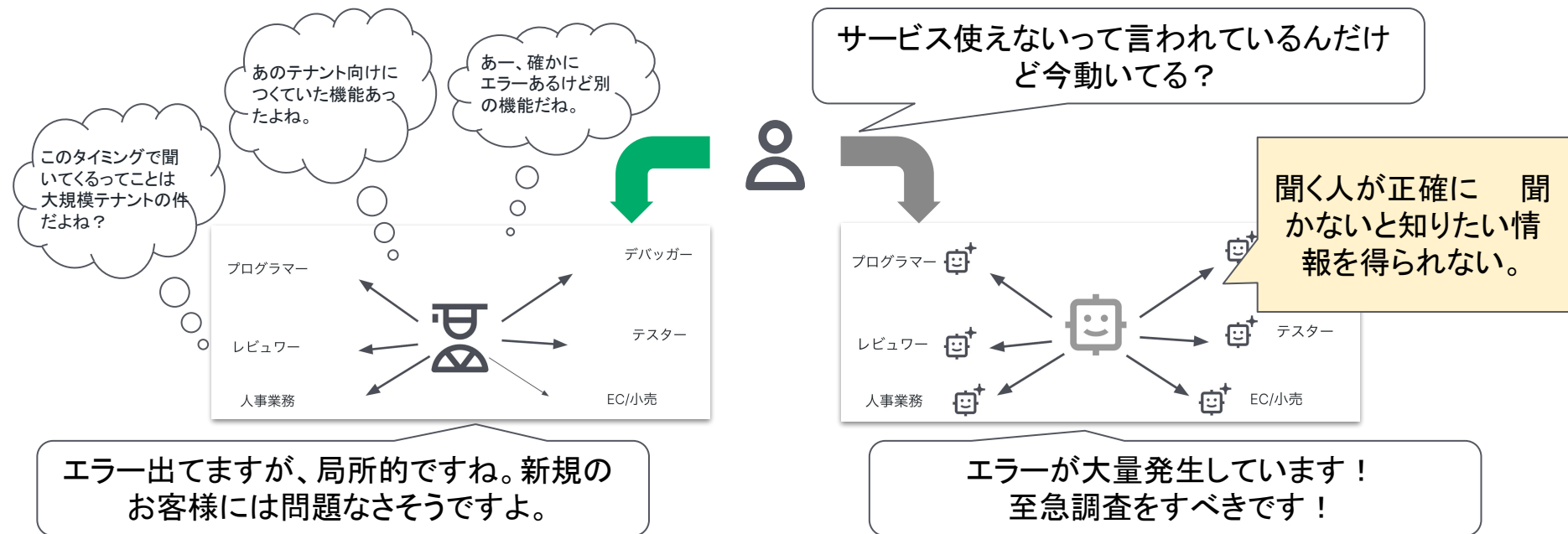
特定の一般的な知識ならスーパーエンジニアを超えますが、担当しているサービスに対して同じ質問をした時、スーパーエンジニアしか正確に答えられないケースがあります。



# 本番環境でスーパーエンジニアにならないAI

AIは知識が豊富で、大抵のことは答えられます。

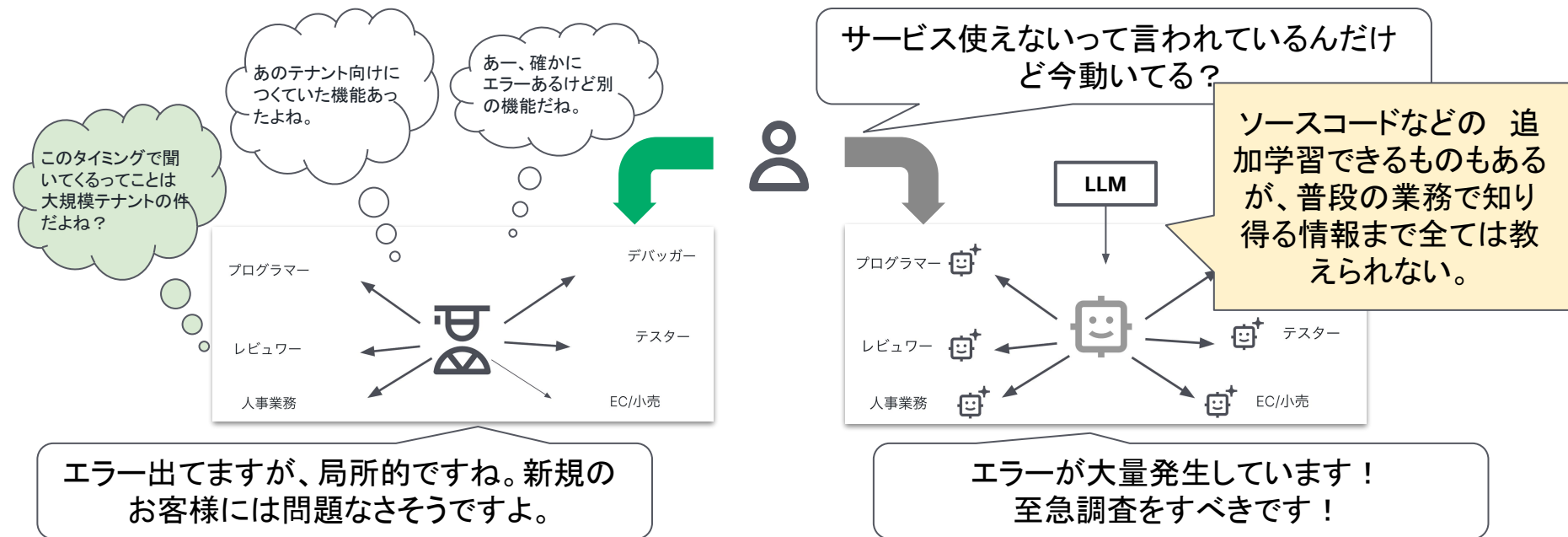
特定の一般的な知識ならスーパーエンジニアを超えますが、担当しているサービスに対して同じ質問をした時、スーパーエンジニアしか正確に答えられないケースがあります。



# 本番環境でスーパーエンジニアになれないAI

AIは知識が豊富で、大抵のことは答えられます。

特定の一般的な知識ならスーパーエンジニアを超えますが、担当しているサービスに対して同じ質問をした時、スーパーエンジニアしか正確に答えられないケースがあります。

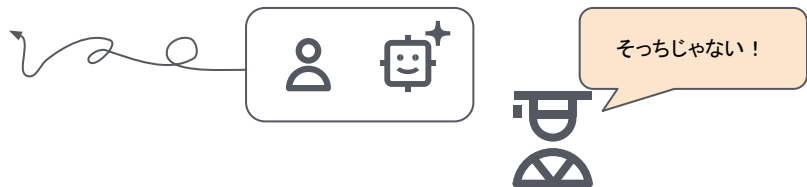


# お客様から聞く、Jr.エンジニアのAI活用の悩み

## 分析の初手で間違ってしまう

サービスのことをよく知らない新参のエンジニアからすると、データの在処さえわからない。

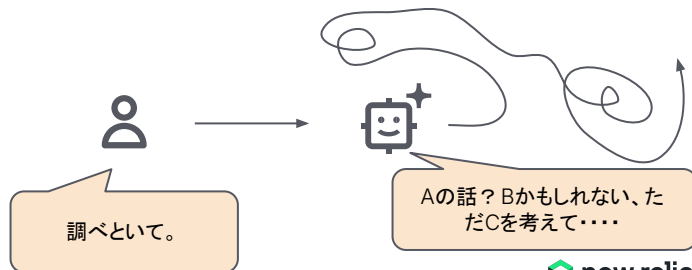
まず何を確認したら良いかわからないので、知っている知識を元に手探りで動いてしまう。



## 知識がないとAIも使えない

AIは、"知識が豊富だけど自分から動いてくれない優等生"

知識があるからこそ、やってほしいことを正しく伝えないと様々な知識を引っ張り出して進んでいる風になってしまう。





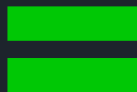
多量の知識/経験に頼らない分析



オブザーバビリティ

# オブザーバビリティ概要

# オブザーバビリティ



"システムから出力されるデータを用いて、その  
内部状態をどれだけ正確に推測できるかを示  
す指標"

オブザーバビリティとは？

<https://newrelic.com/jp/blog/best-practices/what-is-observability>



# new relic 様々なデータをリアルタイム に収集する オブザーバビリティプラットフォーム

より良いソフトウェアの開発と実行

## New Relic Service Level Management

Service Levelの迅速な設定と可視化分析

## New Relic BROWSER

ブラウザ体験モニタリング  
ユーザー目線でページロードやエラーを把握

## New Relic MOBILE

モバイル環境をモニタリング  
iOSとAndroidアプリに対応

## New Relic SYNTHETICS

外形モニタリング  
世界複数拠点からの外形監視

顧客体験の改善

## New Relic APM

アプリケーション性能モニタリング  
8言語と70を超えるフレームワークに対応

## New Relic INFRASTRUCTURE

あらゆるインフラ環境をモニタリング  
パブリッククラウドとオンプレミス

## New Relic LOGS

ログ収集と高速検索  
MELT を高速収集し検索可能に

## New Relic NPM

SNMP、NetFlow、VPC などのネットワークを  
容易に収集可視化

## New Relic CodeStream

IDE上でNew Relicで特定したErrorから該当  
ソースコードに直接ジャンプ可能

## New Relic New Relic AI

生成AIによるオブザーバリティアシスタント

## New Relic Alert & AI

AIを利用した検知とインシデント対応の効率化

## New Relic Dashboards

ダッシュボード開発チャートビルダーで分析を  
超高速化し、あらゆるテレメトリデータの可視化  
を実現

## New Relic Security RX

アプリ/OS脆弱ライブラリの自動特定や、  
システム全体の脆弱性を統合的に可視化

複雑かつ大規模システムの管理

Perfect  
Software

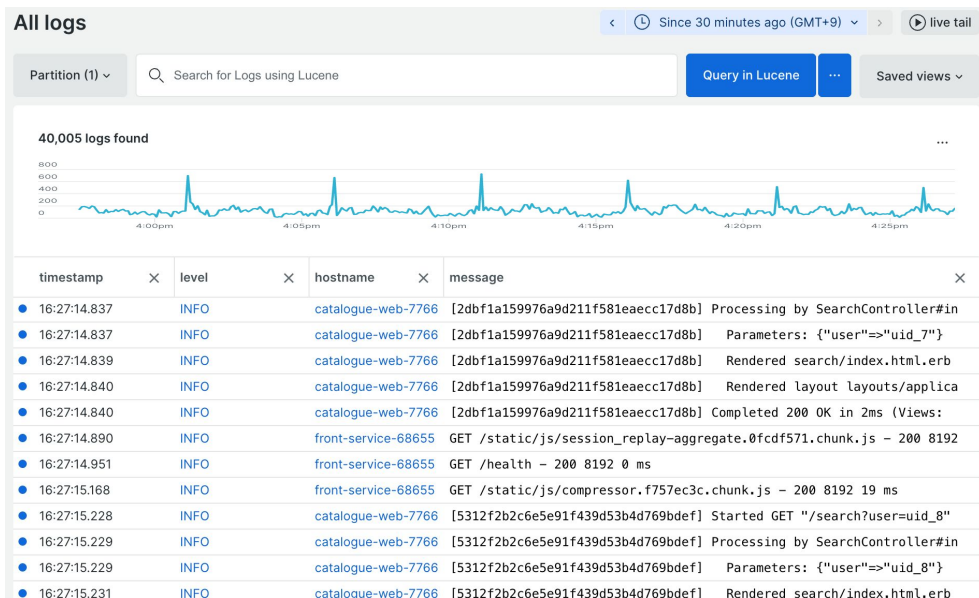
# Telemetry Data Platform

# 誰でも戦力になる組織のポイント

1、Jr. エンジニアでも迷わない観測データ基盤を作る

2、事前知識がなくても筋の良い分析ができるデータを用意する

# 整理されていないログからの分析は知識が必要



# 技術支援が難しい例

"ログ"と言われても情報や形式は無限大。何が起きているかはサービスを知らないと分析できない。

New Relic初心者の  
お客様



3日前に起こった障害の問題がわかっていない。  
分析方法を教えて欲しい。  
全てのログをNew Relicに入れています。

New Relic  
技術支援社員



どんな事象が起きましたか？  
"全て"ってなんだろう。。。ISO27002順守？  
ログの仕様を教えてくださいませんか？

```
org.springframework.web.util.NestedServletException: Request processing failed; nested exception is java.lang.IllegalArgumentException: Item ID
cannot be null or empty
    at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:982)
    at org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:872)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:689)
    at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:846)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:770)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
    at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
    at org.springframework.web.filter.RequestContextFilter.doFilterInternal(RequestContextFilter.java:99)
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:107)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
    at org.springframework.web.filter.HiddenHttpMethodFilter.doFilterInternal(HiddenHttpMethodFilter.java:93)
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:107)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
    at org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:197)
    at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:107)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
    at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:202)
    at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:97)
```

# 技術支援が難しい例

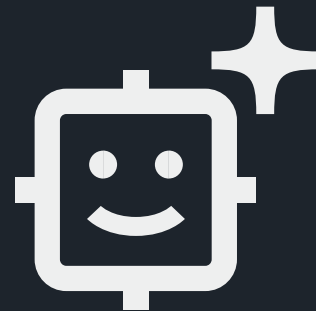
New Relic初心者の  
お客様



3日前に起こった障害の問題がわかっていない。  
分析方法を教えて欲しい。  
全てのログをNew Relicに入れています。

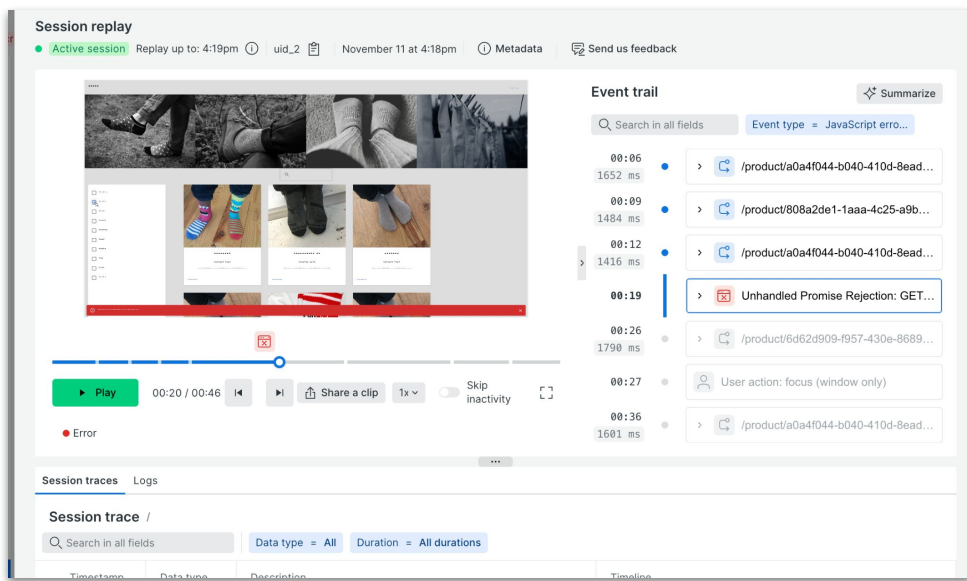
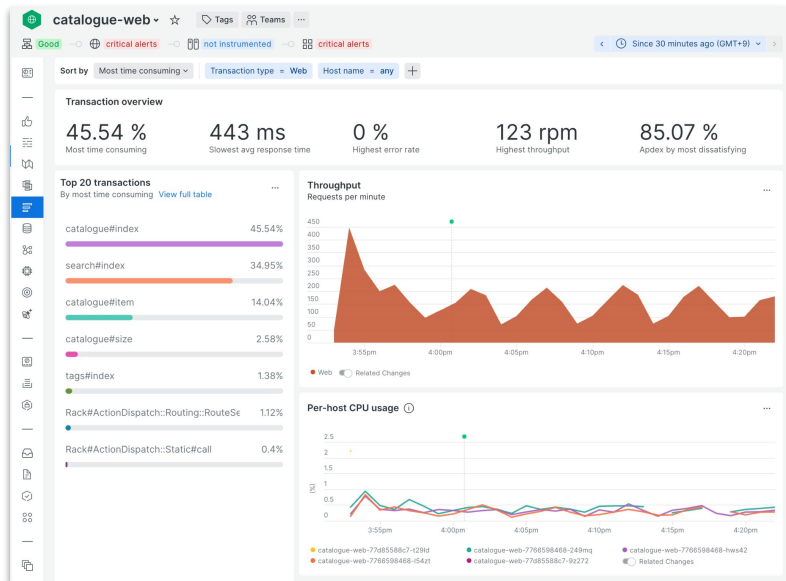
"ログ"と言われても情報や形式は無限大。何が起きているかはサービスを知らないと分析できない。

**AIでも同じ**



どんな事象が起こったか確認してみましょう。  
ログ量が多いので10分間に絞ります。  
10分間では問題が起こっていないようです。  
遡ってログを分析してみましょう。

# 用途によって整理されたデータを使える環境



# 技術支援の一例

サービスについては知らないが、  
どんなデータがあるか知っているので分  
析手順が読める。

New Relic初心者の  
お客様



3日前に起こった障害の問題がわかっていない。  
分析方法を教えて欲しい。  
APMは導入済みです。

New Relic  
技術支援社員



どんな事象が起きましたか？  
…それならAPMのこの画面から見てみましょ  
う。

# 誰でも戦力になる組織のポイント

1、Jr. エンジニアでも迷わない観測データ基盤を作る

2、事前知識がなくても筋の良い分析ができるデータを用意する





よし、ログに残しておこう

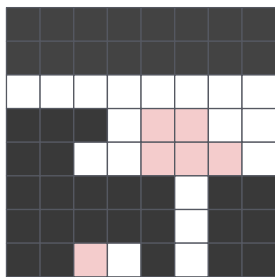
# 大量データ(主にログ)依存の課題

## データ量最適化でサンプリングする

確実に必要とわかっているデータ以外は削除対象となりやすい。

ただし、“観測”していつもと違う異常を発見したり、分析したいのであれば、正常データも重要。

□ = データ    ■ 削減したデータ    = 正常データ



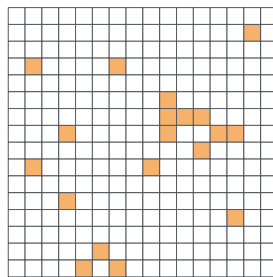
比較できるデータが少ない

## データが複雑になると知識が必要

ログに情報を点々と残してしまうと、“データの組み合わせ”が重要となる。

“ログにどんなデータが書かれるか”の知識が必要になる。

データ分析ロジックも作り込まないといけないので、気軽に分析できない。



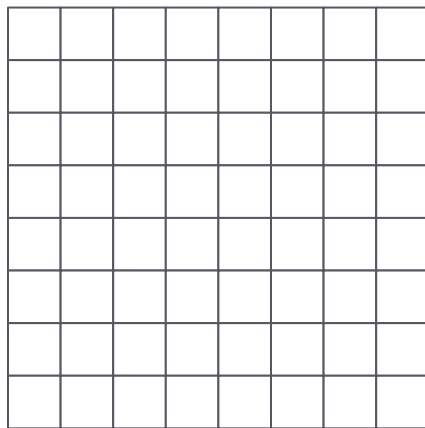
# ちゃんと障害対応できるエンジニアから学ぶ

症状を聴きつつも、本質は何かを考察しながら障害を特定していきます。

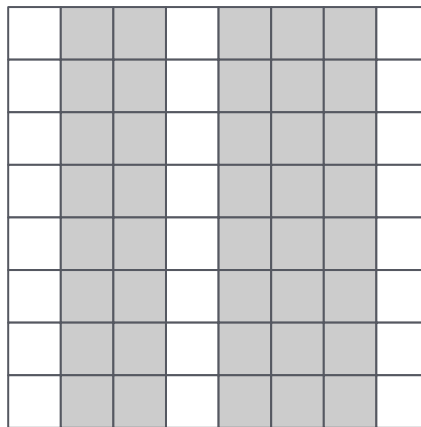
“問題ではない情報”を省いていく、“問題である情報”を収集する、両軸で整理していきます。

事象によっては経験則でステップ数少なく分析できます。

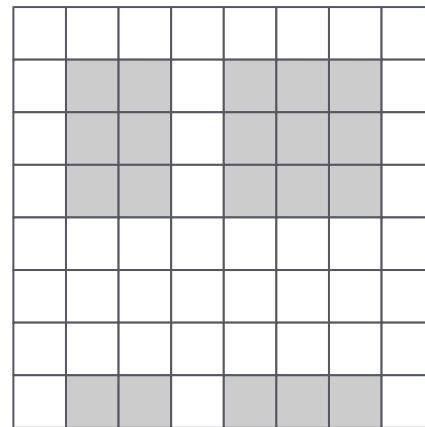
全体の状況を確認でき、確認したいポイントを抑えて徐々に問題を特定していく



Step1



Step2



Step3



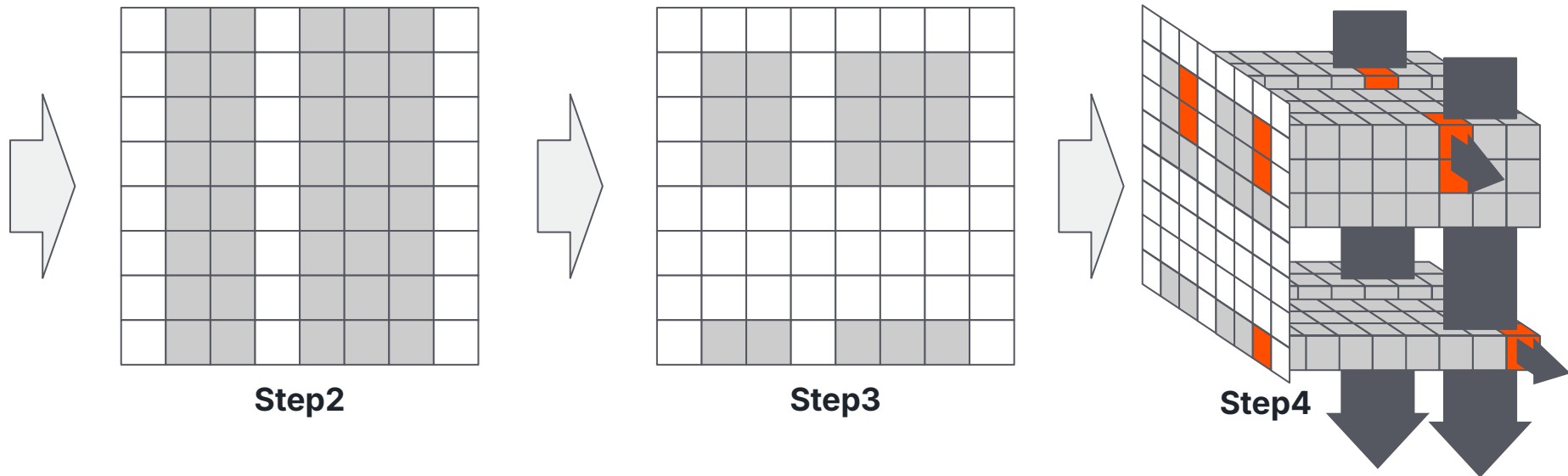
# ちゃんと障害対応できるエンジニアから学ぶ

症状を聴きつつも、本質は何かを考察しながら障害を特定していきます。

“問題ではない情報”を省いていく、“問題である情報”を収集する、両軸で整理していきます。

事象によっては経験則でステップ数少なく分析できます。

全体の状況を確認でき、確認したいポイントを抑えて徐々に問題を特定していく。軸が多いほど解像度が上がる。



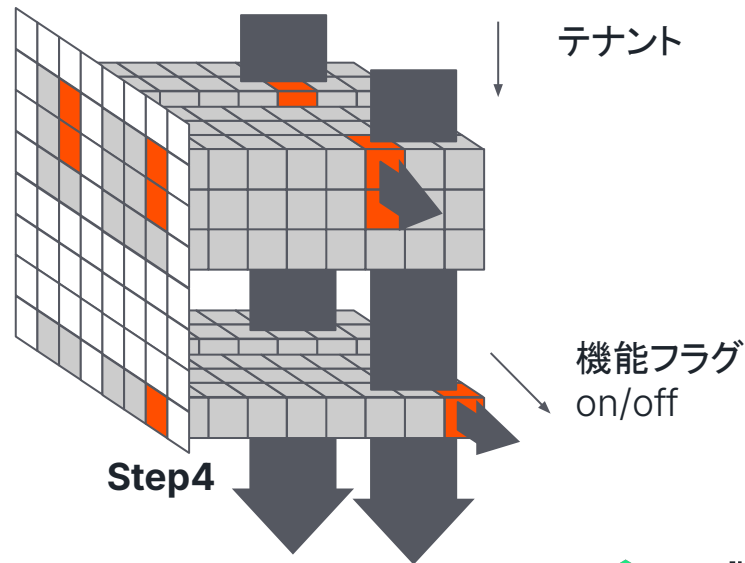
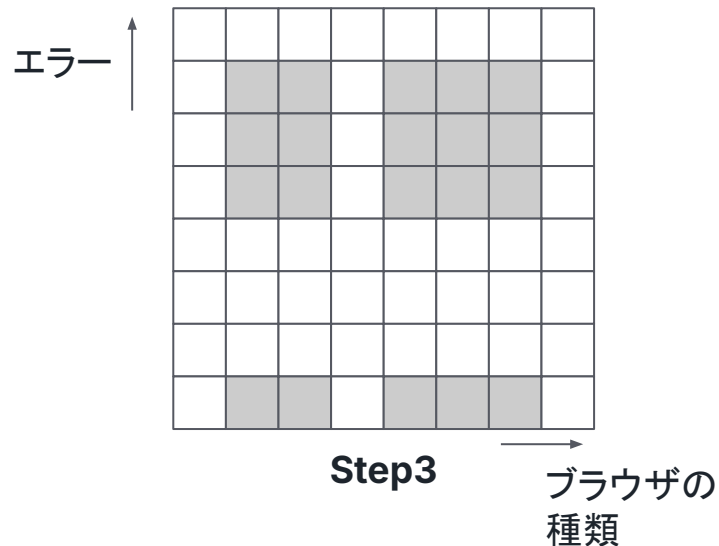
# ちゃんと障害対応できるエンジニアから学ぶ

症状を聴きつつも、本質は何かを考察しながら障害を特定していきます。

“問題ではない情報”を省いていく、“問題である情報”を収集する、両軸で整理していきます。

事象によっては経験則でステップ数少なく分析できます。

全体の状況を確認でき、確認したいポイントを抑えて徐々に問題を特定していく。軸が多いほど解像度が上がる。

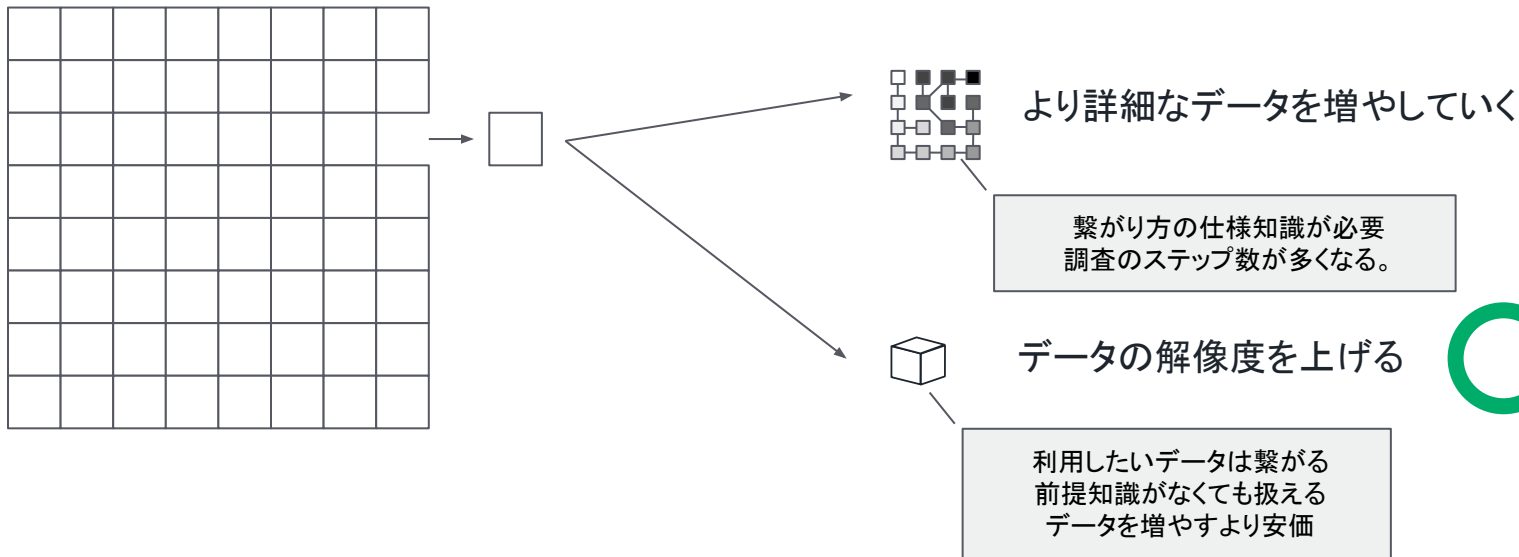


# データを"増やす"ではなく"解像度を上げる"

データを増やすと、"データの関係性"がんな設計なのか知識が必要になってしまいます。

データの分析軸を多くしておけば、知識がなくても "この軸で見てみよう" というように、試行錯誤が容易になります。

全体の状況を確認でき、確認したいポイントを抑えて徐々に問題を特定していく



# Eventのデータ構造 = シンプル & スキーマレス

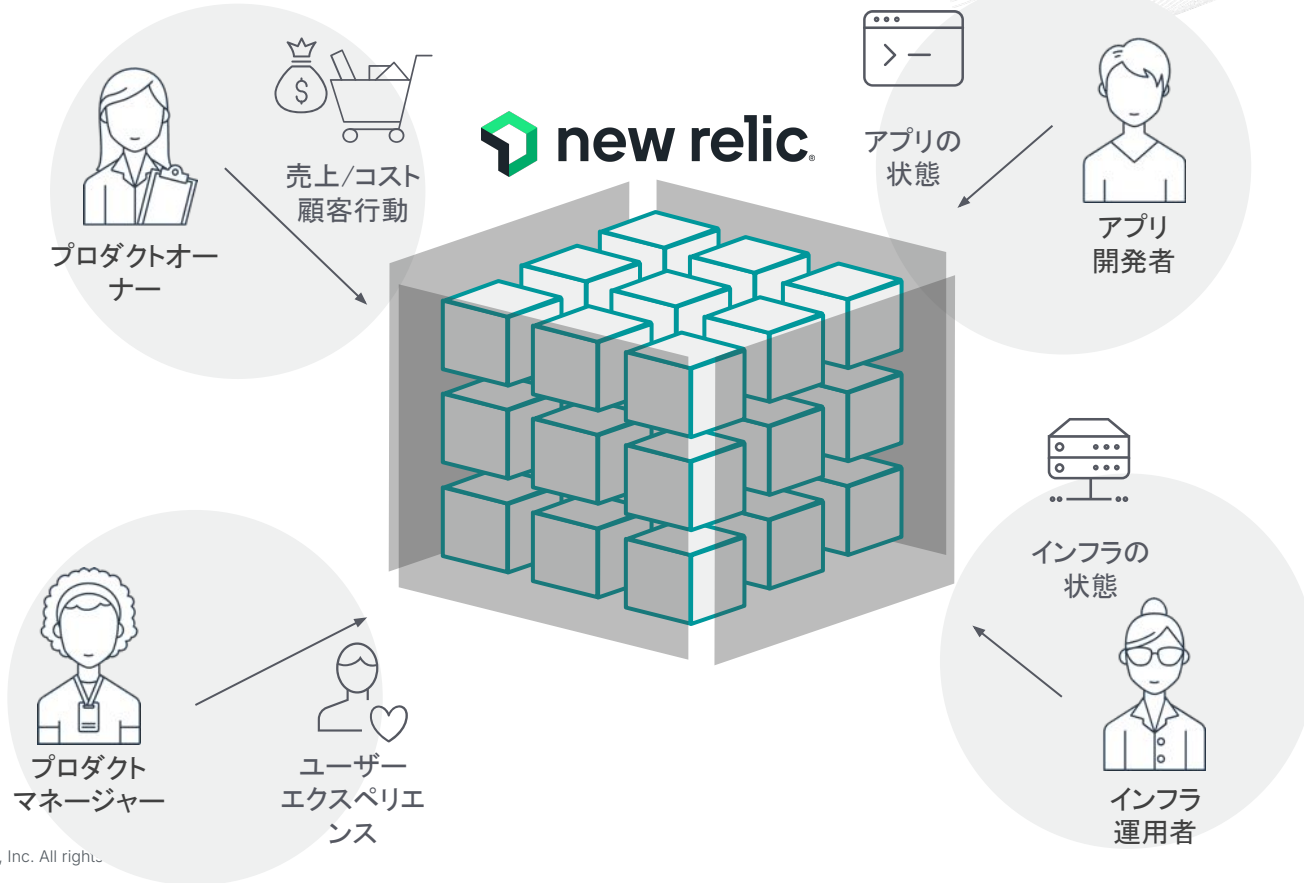
```
{  
  "appName": "front-service",  
  "clusterName": "newrelic",  
  "containerName": "front-service",  
  "databaseCallCount": 3,  
  "databaseDuration": 0.014582519,  
  "duration": 0.06666215,  
  "request.method": "GET",  
  "request.uri": "/",  
  "userId": "1234567890",  
  "tenantId": "awesometenant",  
  "featureFlag1": "true",  
  "notificationType": "mail",  
}
```

デフォルトの観測データ

ユーザーが追加できるコンテキスト

データ行を増やすのではなく、  
データの次元を増やすイメージ

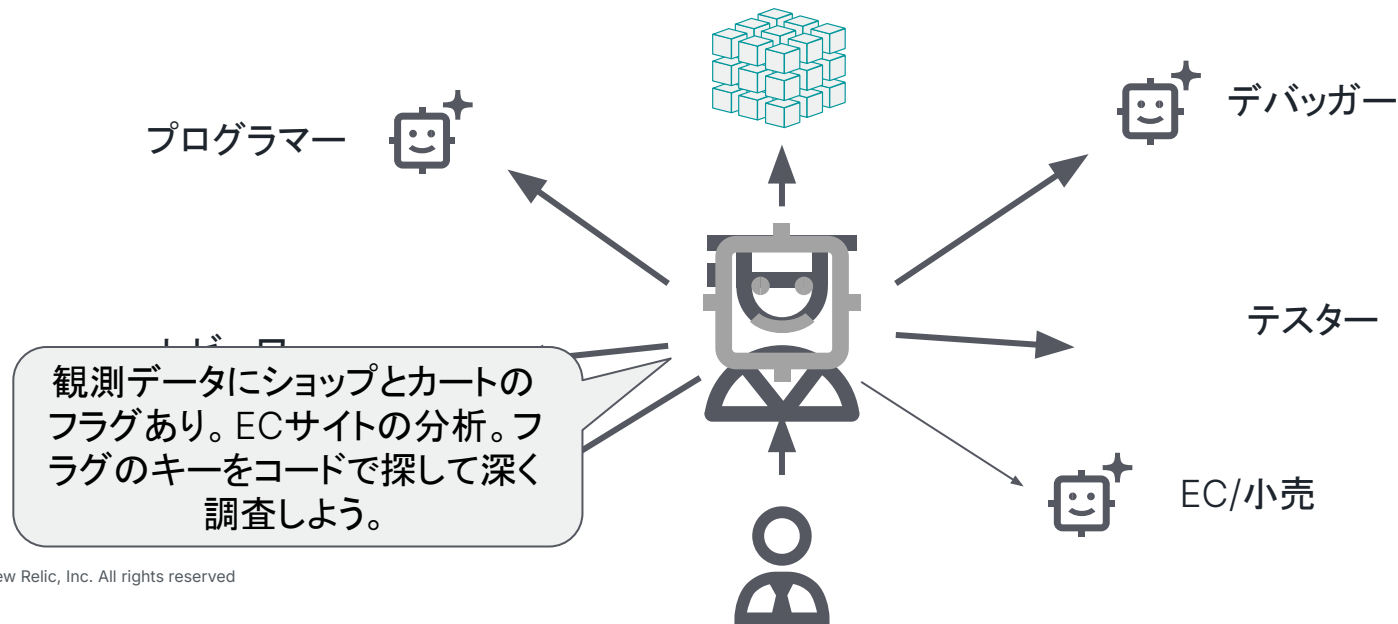
# New Relic が提供するデータ分析機能





# エンジニア / AIがアクションできるデータとなる

"データが低次元"だと憶測/推測で闇雲に進めるしかありません。  
分析のためのデータがついていれば、そこから分析方法を決めることができます。  
そうすれば、人間にも理解できさらに AIで高速化できるようになるのです。



# 分析しやすいデータがあれば知識も少しで良い

## "大量なデータ"だけでは判断が鈍る

"オブザーバビリティ"達成のためになんでも情報を集める！というのは間違い。

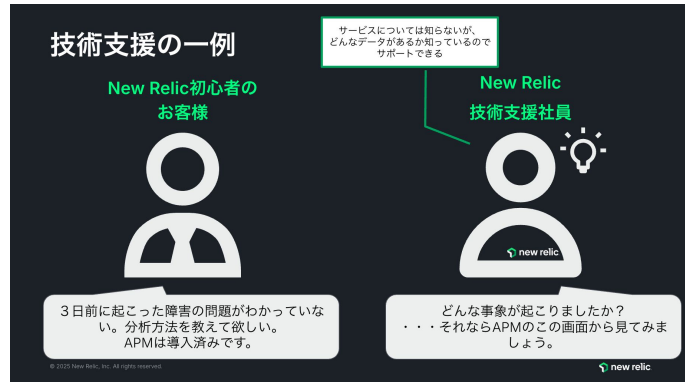
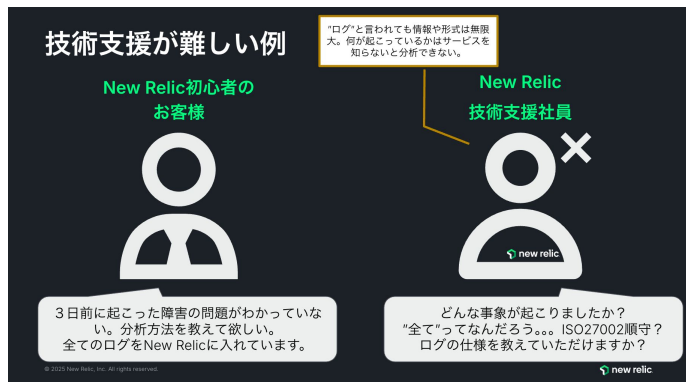
量が多いだけでは正確な判断はできません。

また、人間が見て分析が難しい情報は、AIが  
見ても難しいようです。

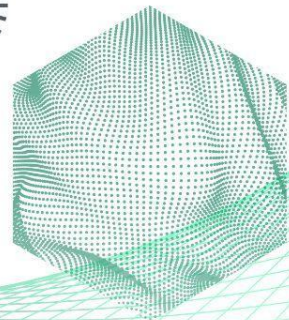
## "意味がわかるデータ"を増やす


New Relicのようなツールを使うと、構造化されたデータを集め分析できます。

"一般的な問題"を分析する知識と、"整理された構造化データ"があれば、どんなシステムでも思考停止せず分析可能。

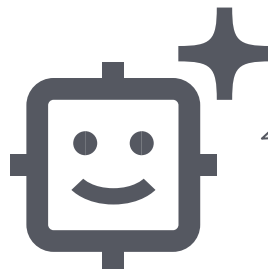


イベントデータを 充実  
させるには

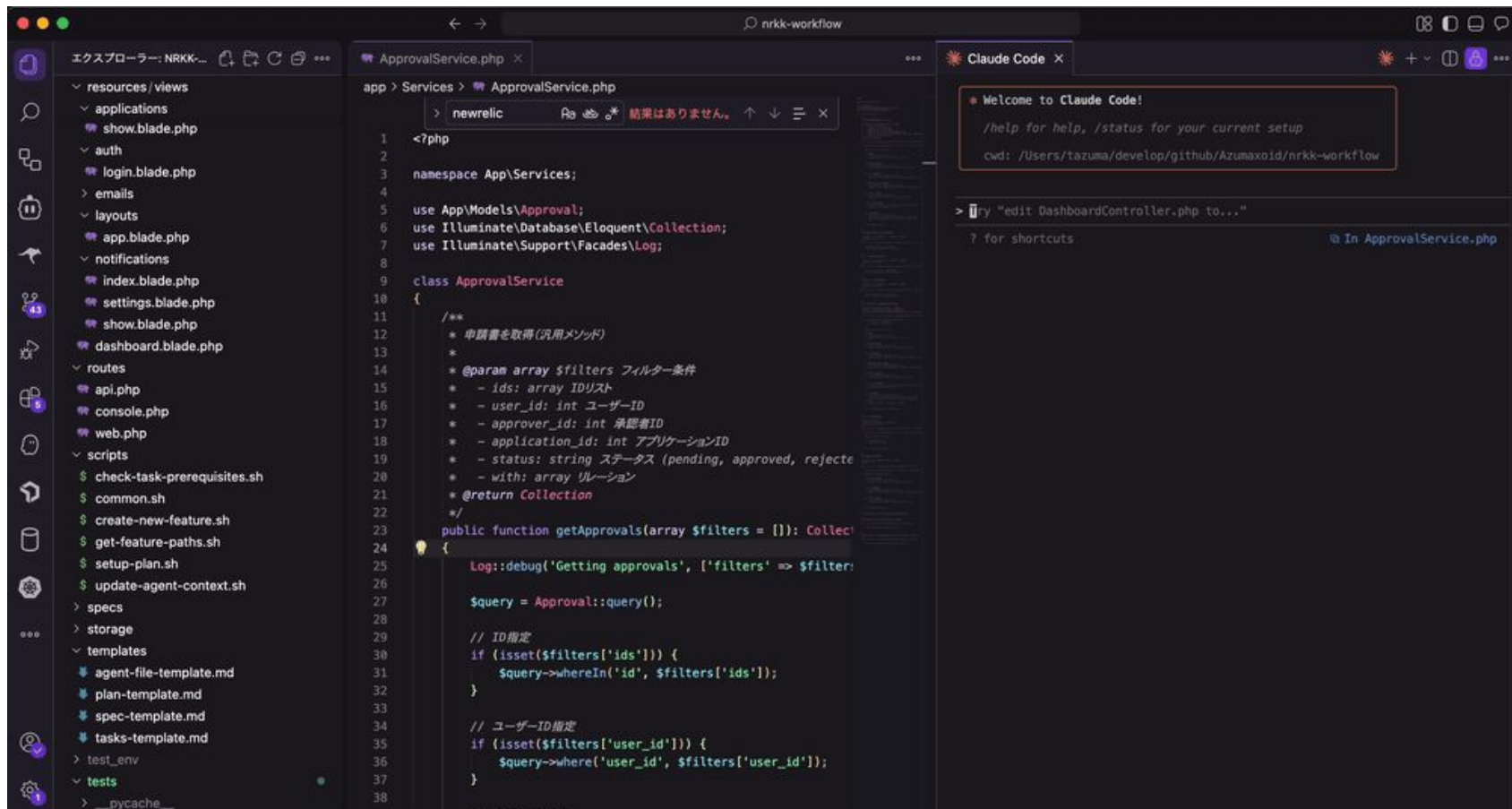




## このサービスを理解できる New Relicのカスタム属性を追加して



素晴らしいアイデアですね！  
それではNew Relicのカスタム属性を追加  
していきましょう。  
まずは現在の実装を確認します。



# ドメイン有識者がデータを作成するのがベスト

特に本番環境で発生する問題は、想定されるエラーを潰した後に発生する難解なものが多い。ソフトウェアエンジニアにとっての難解はソースコードのパーツには不具合が見当たらないもの。その場合、データの流れやお客様の設定内容など流動的な業務(ドメイン)を把握することが重要

例:

- カレンダー連携機能を使っているユーザーだけ通知が届かないことが多い
  - 通知機能のtypeや連携機能の on/offフラグがあると調査がスムーズ
- ユーザー作成に何度も失敗しているらしい
  - Emailはログに出せないなので、不可逆暗号して一意性だけでも調べる
- これとこれの設定の組み合わせは安定しないらしい
  - 機能フラグを確認できるようにする



# まとめ



# 誰でも戦力になる組織を作るためのポイント

## 経験に頼らない仕組み

スーパーエンジニアがもっと作業が楽になる、だけでは置いてけぼりになるエンジニアが出てしまいます。

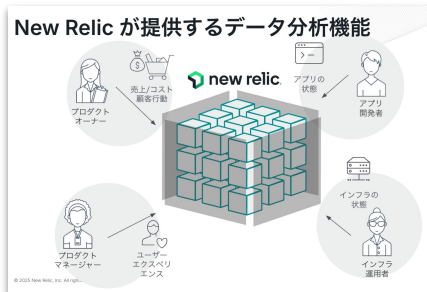
経験が浅くても活躍できるにはどんなことが必要かを考えてみましょう。



## データで知識をカバー

データの役割が決まっている  
データの解像度が高い

これをクリアすると知識が浅くても試行錯誤できるようになります。



## 有識者がデータ作成

ソースコードやドメイン知識、サービスの状況を理解しているエンジニアの方が要点を絞った整理が可能。

コードに依存するので、AIコーディングエージェントも活用可能。





# New Relicの始め方

# New Relic サインアップのご案内

デジタル  
オプティマ  
イゼーション

- 全機能が無料で使い放題（1名）！
- 転送データが 100GB/月まで無料！
- 必要事項記入ですぐに利用開始！
- クレジットカード記入不要！
- 利用期限なし！



 [newrelic.com/jp/sign-up-japan](https://newrelic.com/jp/sign-up-japan)

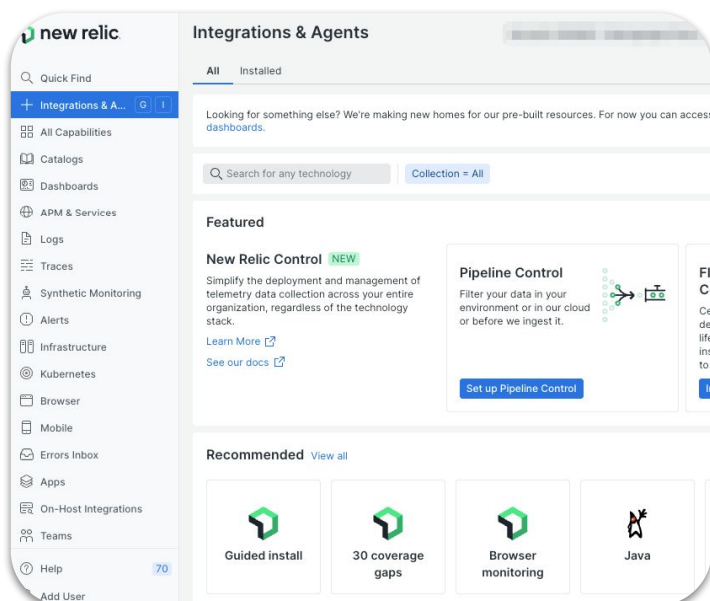
# 導入手順の確認方法

導入手順は公式ドキュメントや New RelicのUIでご紹介しています。

## 公式ドキュメント



## ログイン後画面 (Integrations & Agents)



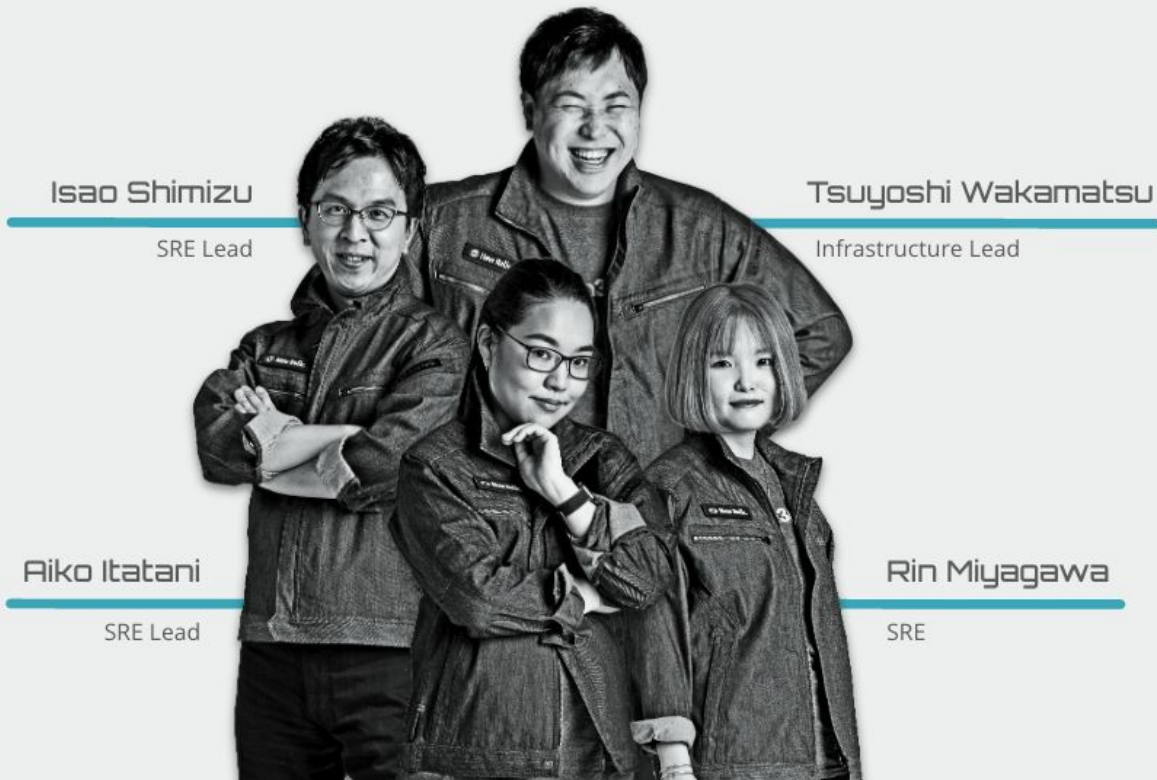
# NRUG

## ぬるぐで学ぶ

New Relic ユーザーが集い、実践事例や最新機能紹介などを実施。初心者支部や SRE 支部などが形成されており、エンジニア同士でのネットワーキングや信頼性の高い情報交換が可能。



Slackチャンネルに参加



New Relic User Group

