



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



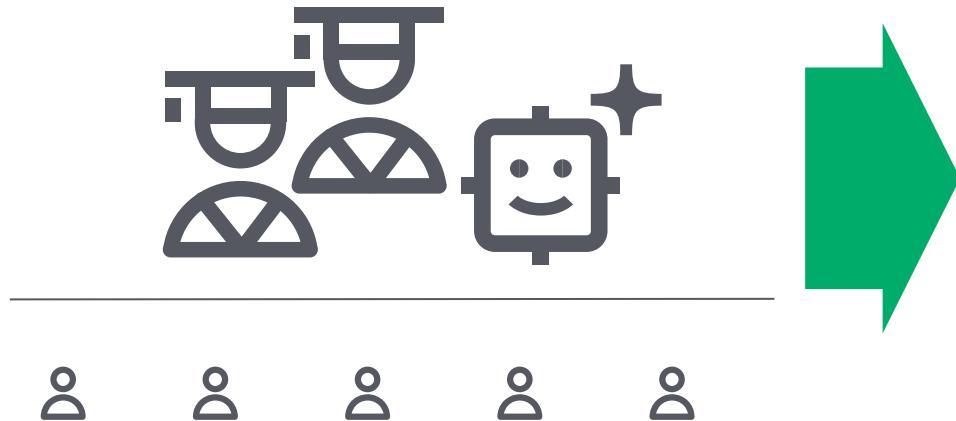
技術統括 カスタマーサクセス部
担当部長

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

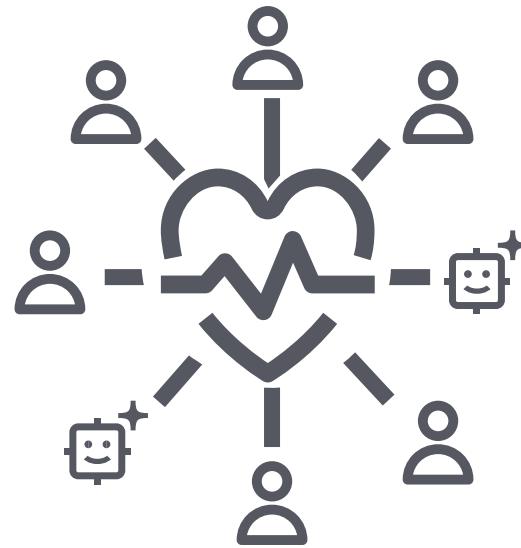


持ち帰っていただきたいヒント

“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

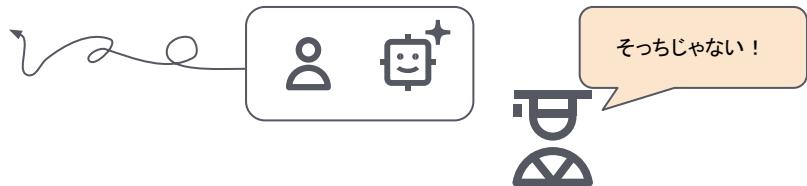


お客様から聞く、Jr.エンジニアのAI活用の悩み

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。

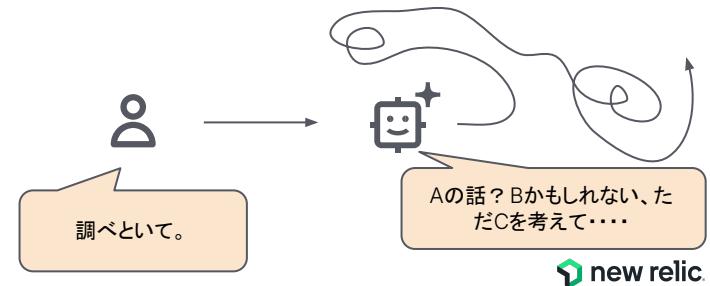
まず何を確認したら良いかわからないので、知っている知識を元に手探りで動いてしまう。

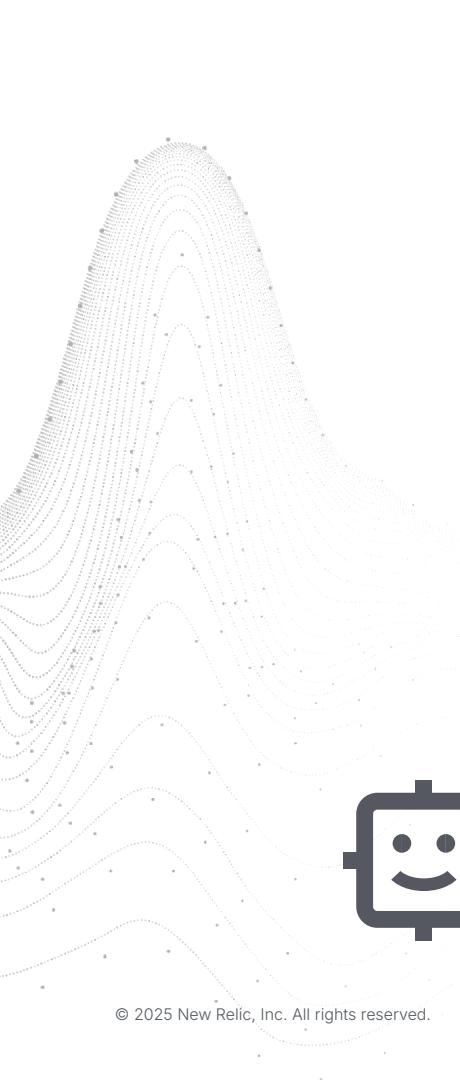


知識がないとAIも使えない

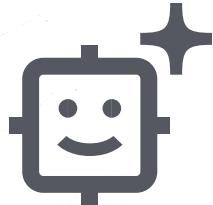
AIは、"知識が豊富だけど自分から動いてくれない優等生"

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ふ。





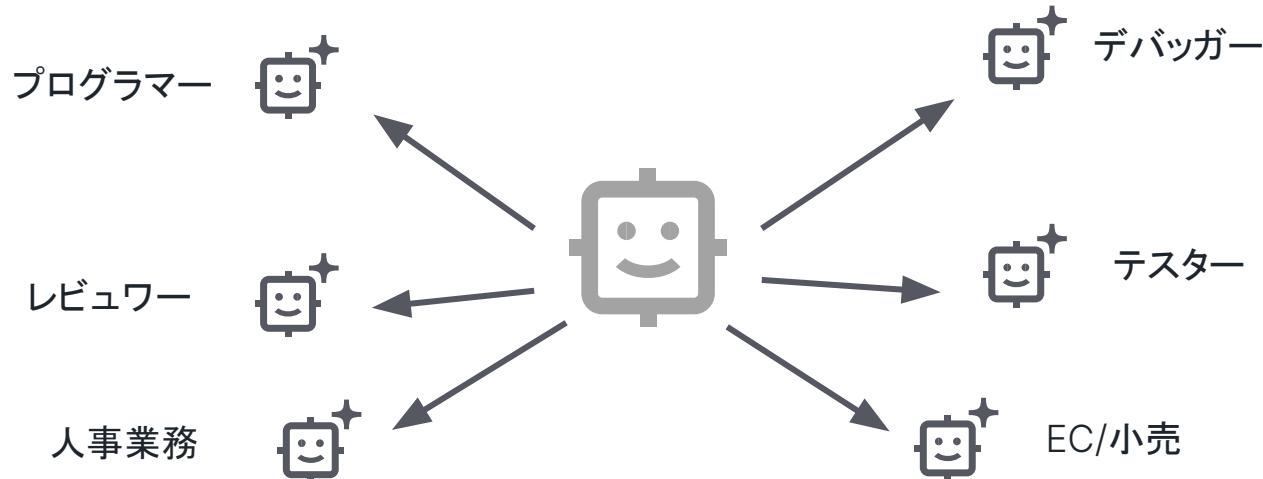
AI vs スーパーエンジニア



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。

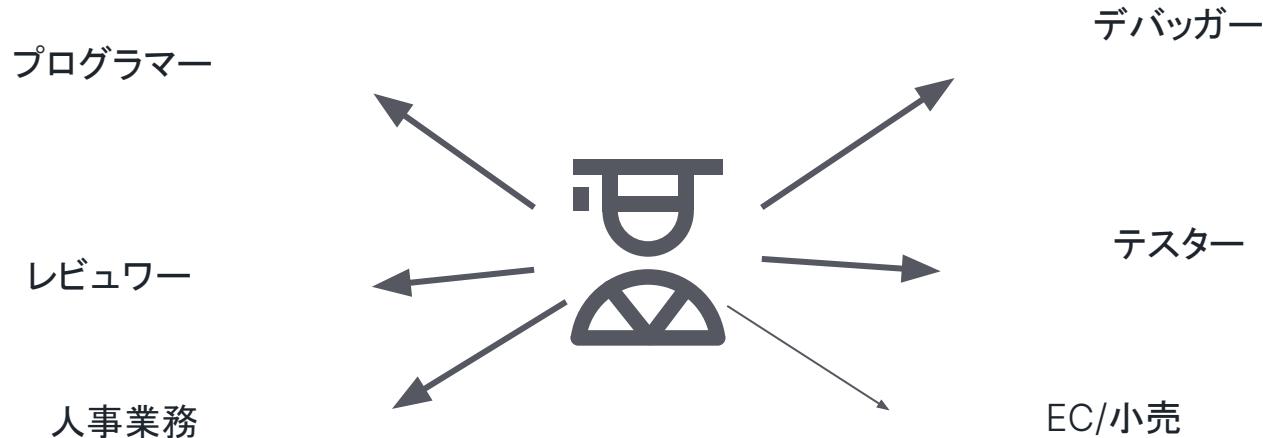
ただし、完璧に1つのAIが業務をこなすことは難しく、専門的にタスクを行う "サブエージェント" を定義したり、プロンプトエンジニアリング / コンテキストエンジニアリングによってAIを賢く使う方法が考えられてきています。



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。

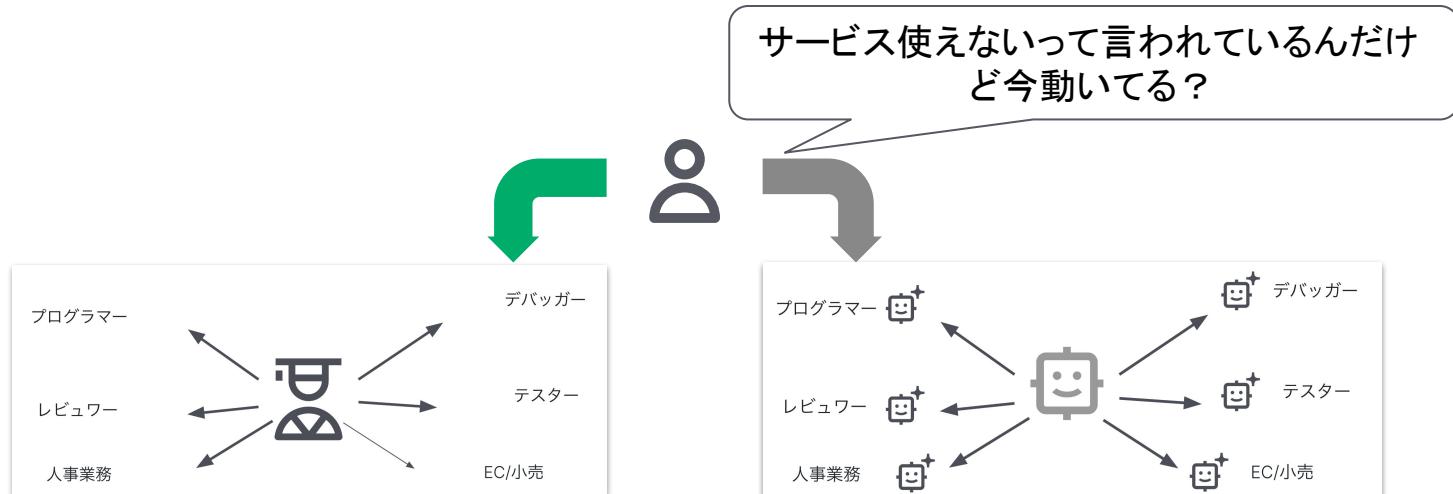
得意分野/不得意分野があったとしても、経験を生かすことで開発スピードや障害対応スピードが 他のメンバーより早い、神技が使えます。



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。



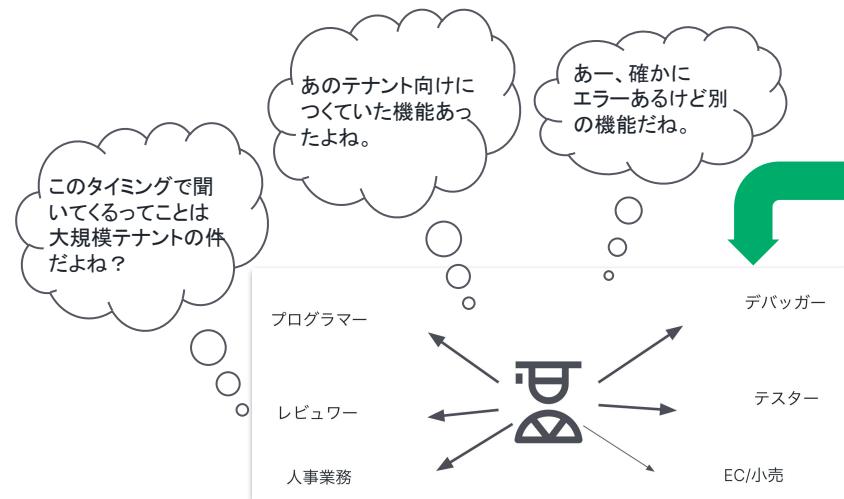
エラー出てますが、局所的ですね。新規の
お客様には問題なさそうですよ。

エラーが大量発生しています！
至急調査をすべきです！

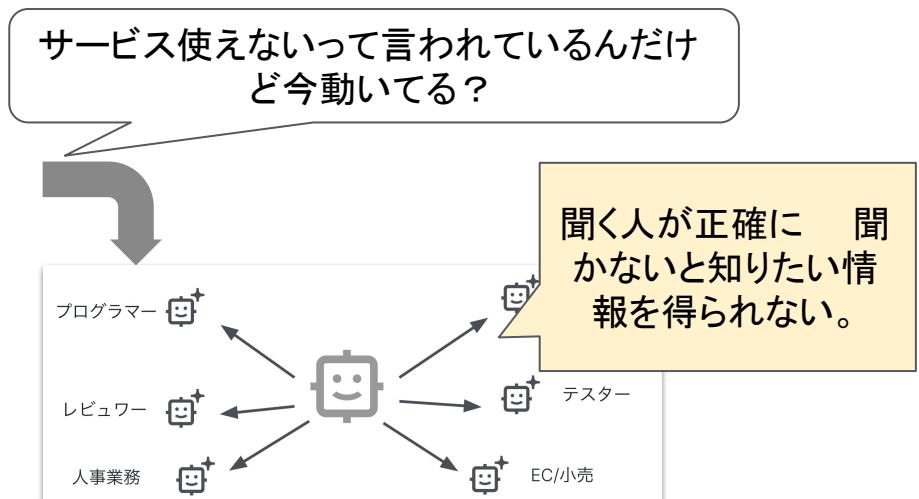
本番環境でスーパーエンジニアになれないAI

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。



エラー出でますか、局所的ですね。新規のお客様には問題なさそうですよ。

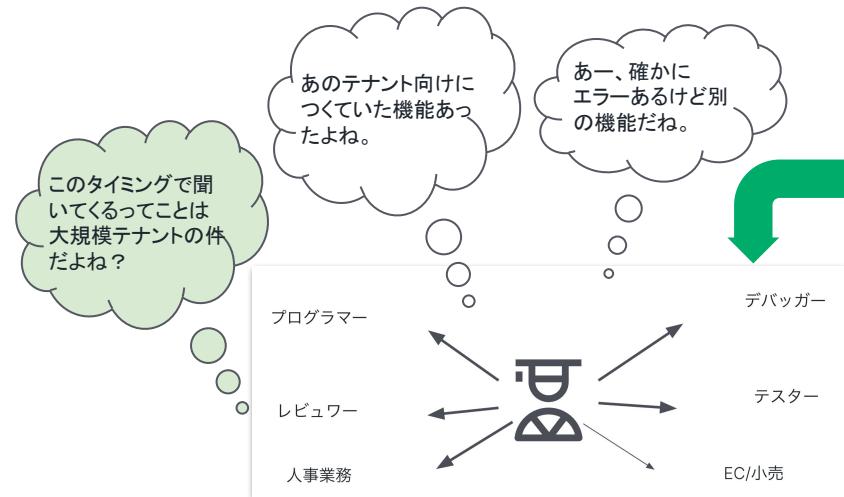


エラーが大量発生しています！
至急調査をすべきです！

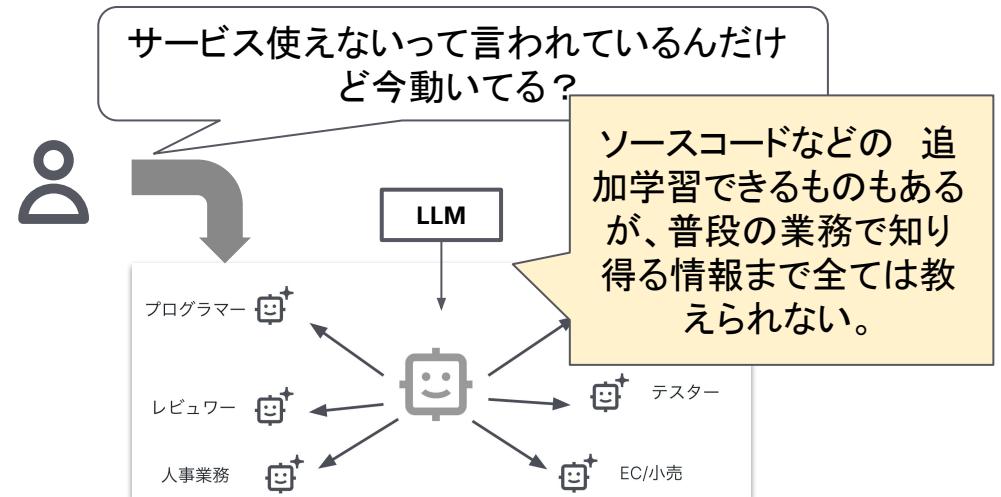
本番環境でスーパーエンジニアになれないAI

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。



エラー出てますが、局所的ですね。新規のお客様には問題なさそうですよ。

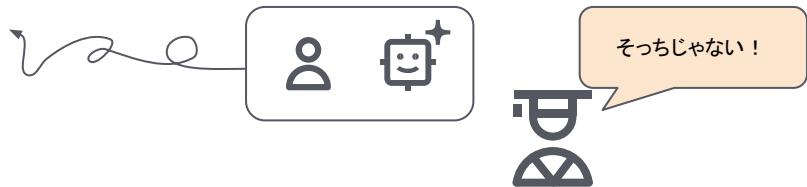


お客様から聞く、Jr.エンジニアのAI活用の悩み

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。

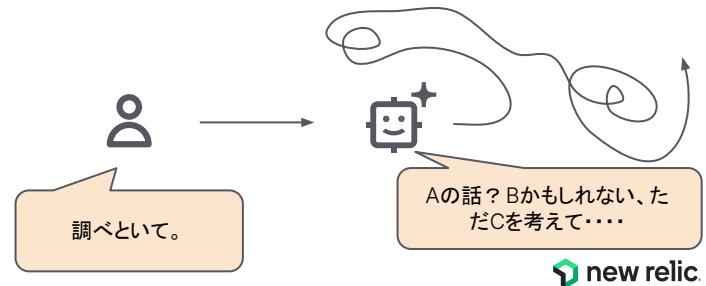
まず何を確認したら良いかわからないので、知っている知識を元に手探りで動いてしまう。

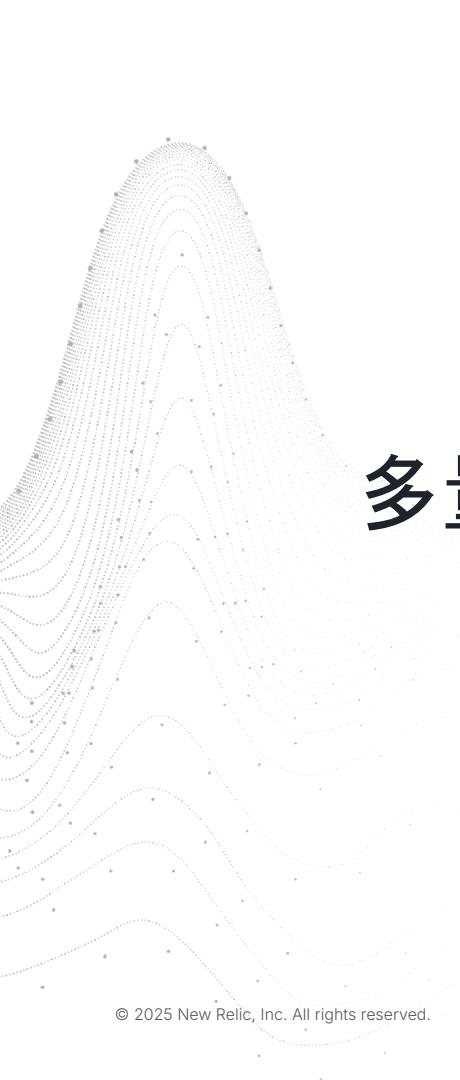


知識がないとAIも使えない

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ふ。





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



Front-End

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



Back-End

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

世界最速のデータ収集と検索

New Relic New Relic AI

生成AIによるオブザーバビ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



Perfect
Software



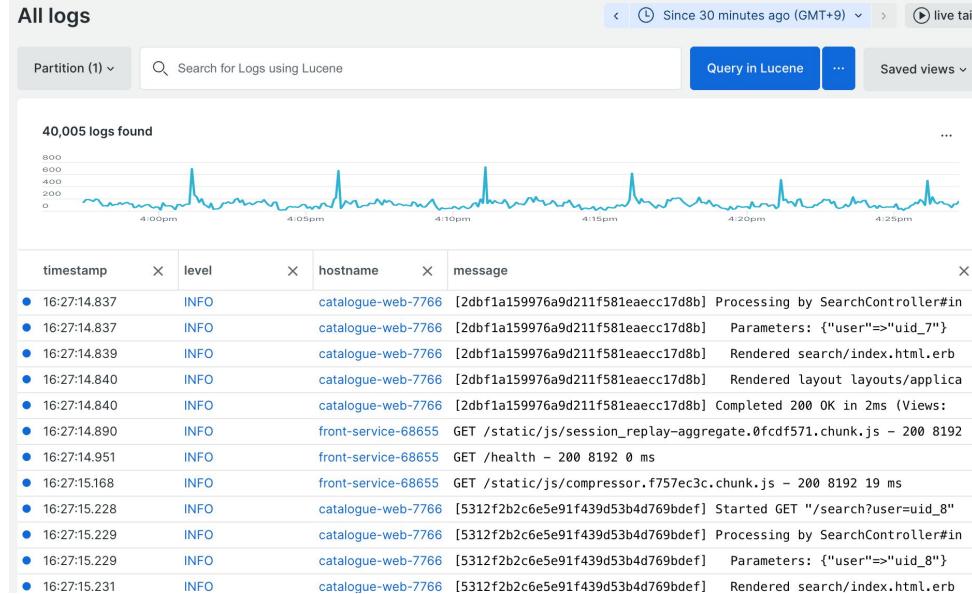
複雑かつ大規模システムの管理

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

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に入れてています。

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

New Relic
技術支援社員



どんな事象が起こりましたか？
"全て"ってなんだろう。。。ISO27002順守？
ログの仕様を教えていただけますか？

技術支援が難しい例

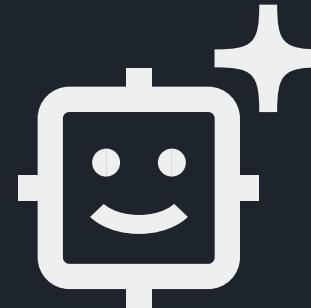
New Relic初心者の
お客様



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

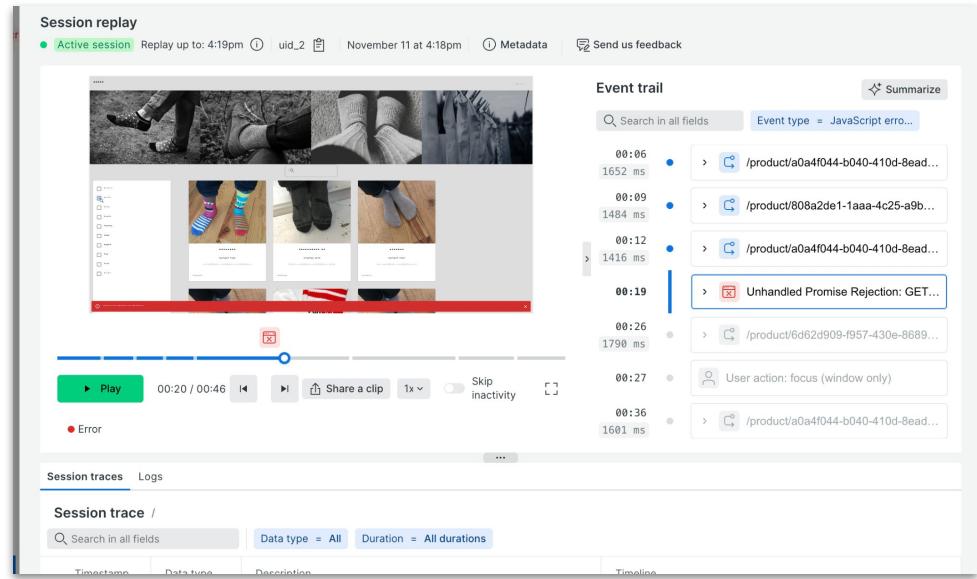
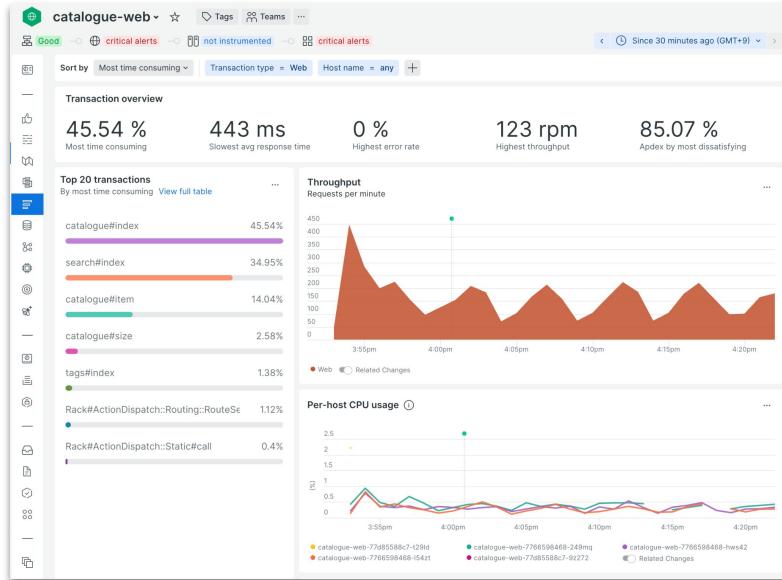
"ログ"と言われても情報や形式は無限大。何が起こっているかはサービスを知らないと分析できない。

AIでも同じ



どんな事象が起き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える環境



技術支援の一例

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。
APMは導入済みです。

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。

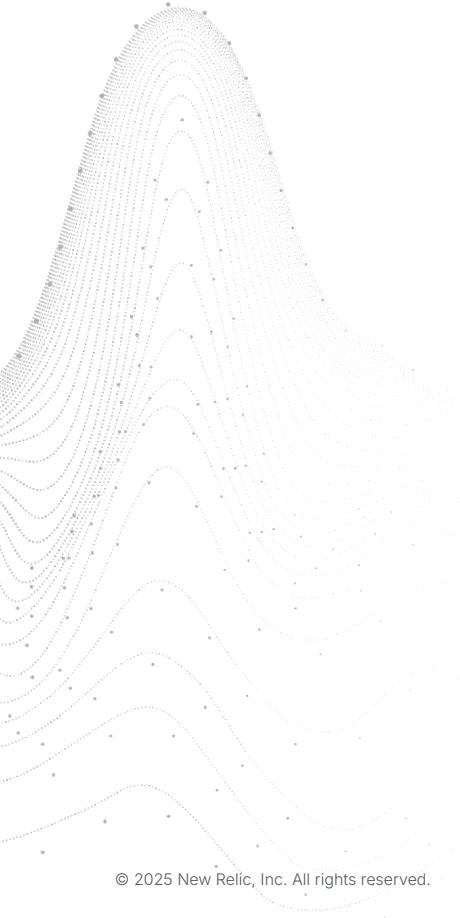
New Relic
技術支援社員



どんな事象が起こりましたか？
…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



ヨシ、ログに残しておこう

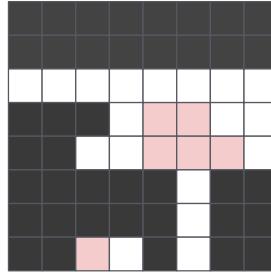
大量データ(主にログ)依存の課題

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 = 間違データ

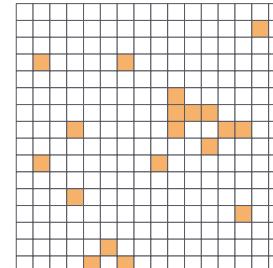


データが複雑になると知識が必要

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。



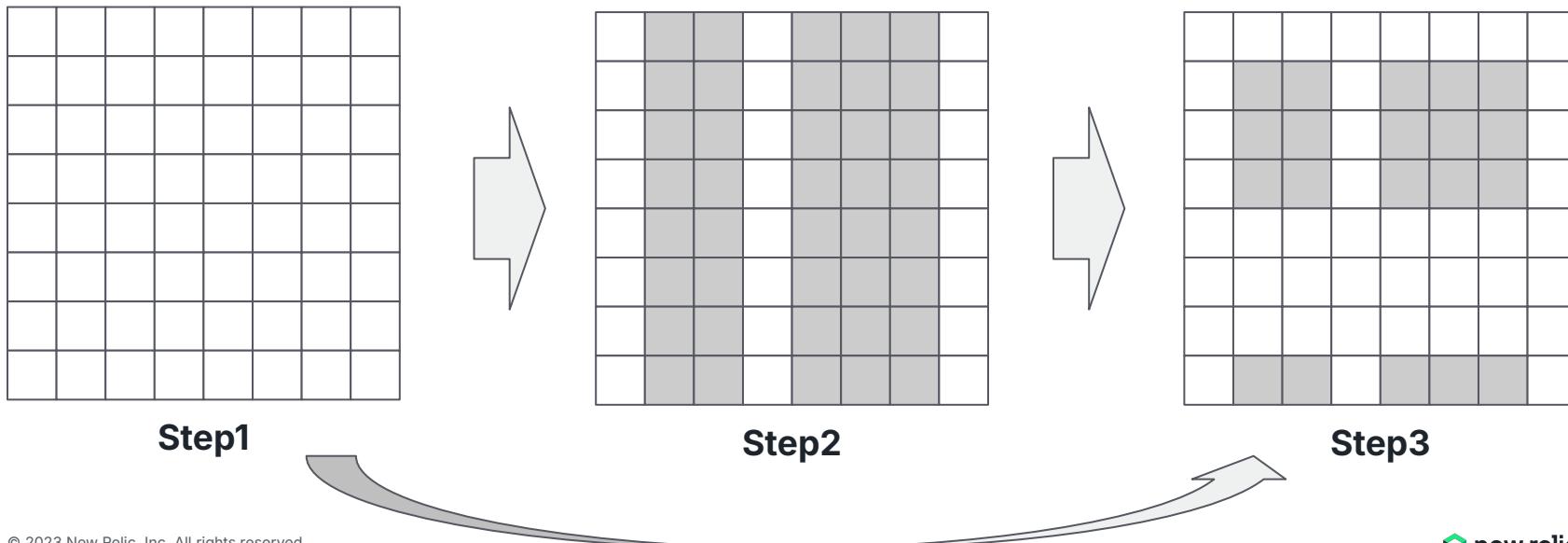
ちゃんと障害対応できるエンジニアから学ぶ

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



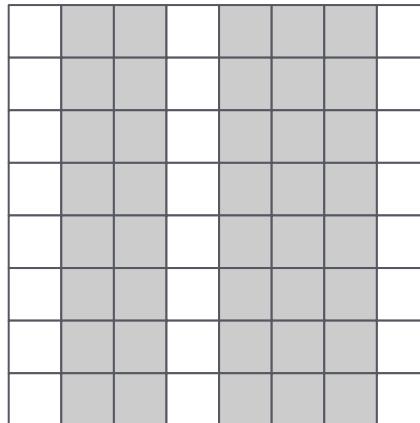
ちゃんと障害対応できるエンジニアから学ぶ

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。

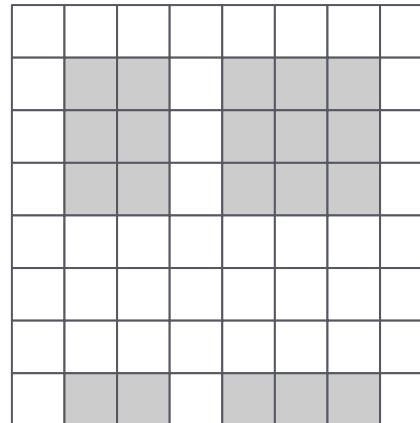
"問題ではない情報"を省いていく、"問題である情報"を収集する、両軸で整理していきます。

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。

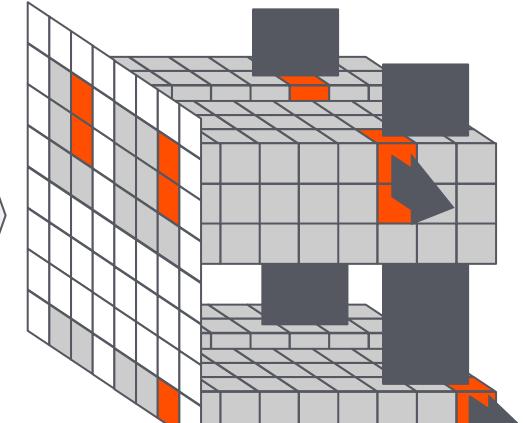
全体の状況を確認でき、確認したいポイントを抑えて徐々に問題を特定していく。軸が多いほど解像度が上がる。



Step2



Step3



Step4

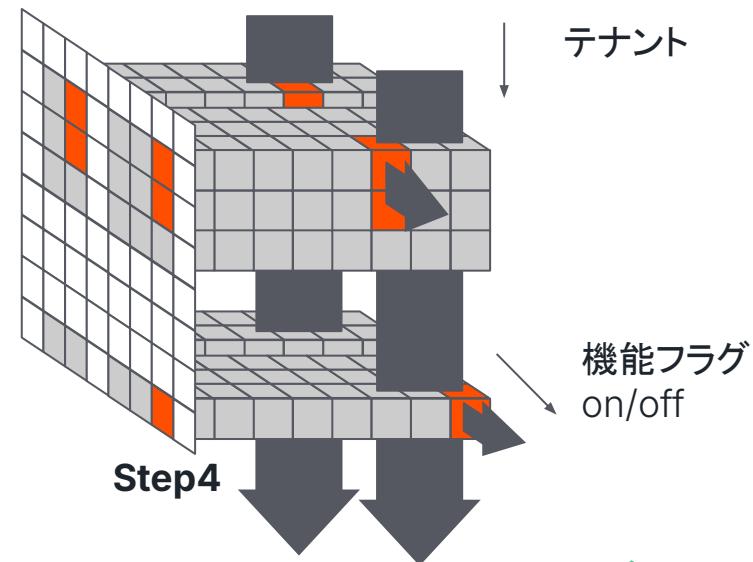
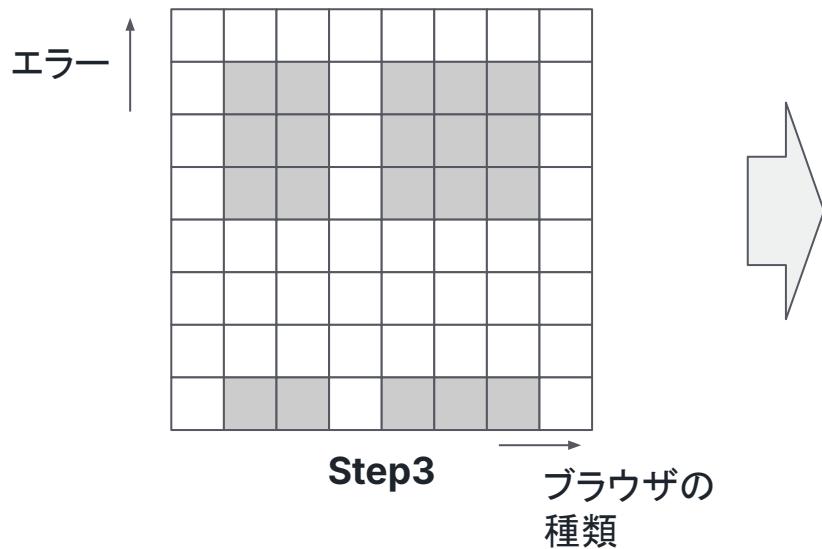
ちゃんと障害対応できるエンジニアから学ぶ

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。

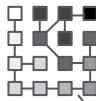
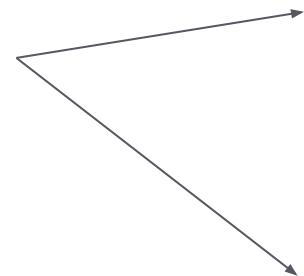
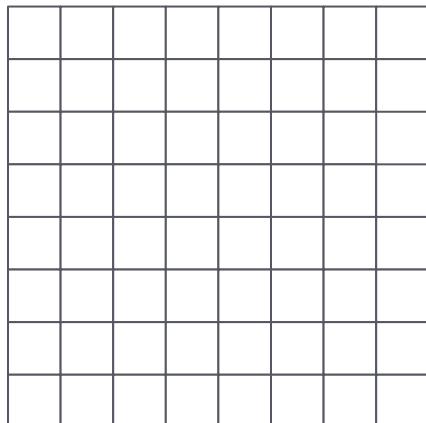


データを"増やす"ではなく"解像度を上げる"

データを増やすと、"データの関係性"がど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



より詳細なデータを増やしていく

繋がり方の仕様知識が必要
調査のステップ数が多くなる。



データの解像度を上げる

利用したいデータは繋がる
前提知識がなくても扱える
データを増やすより安価



Eventのデータ構造 = シンプル & スキーマレス

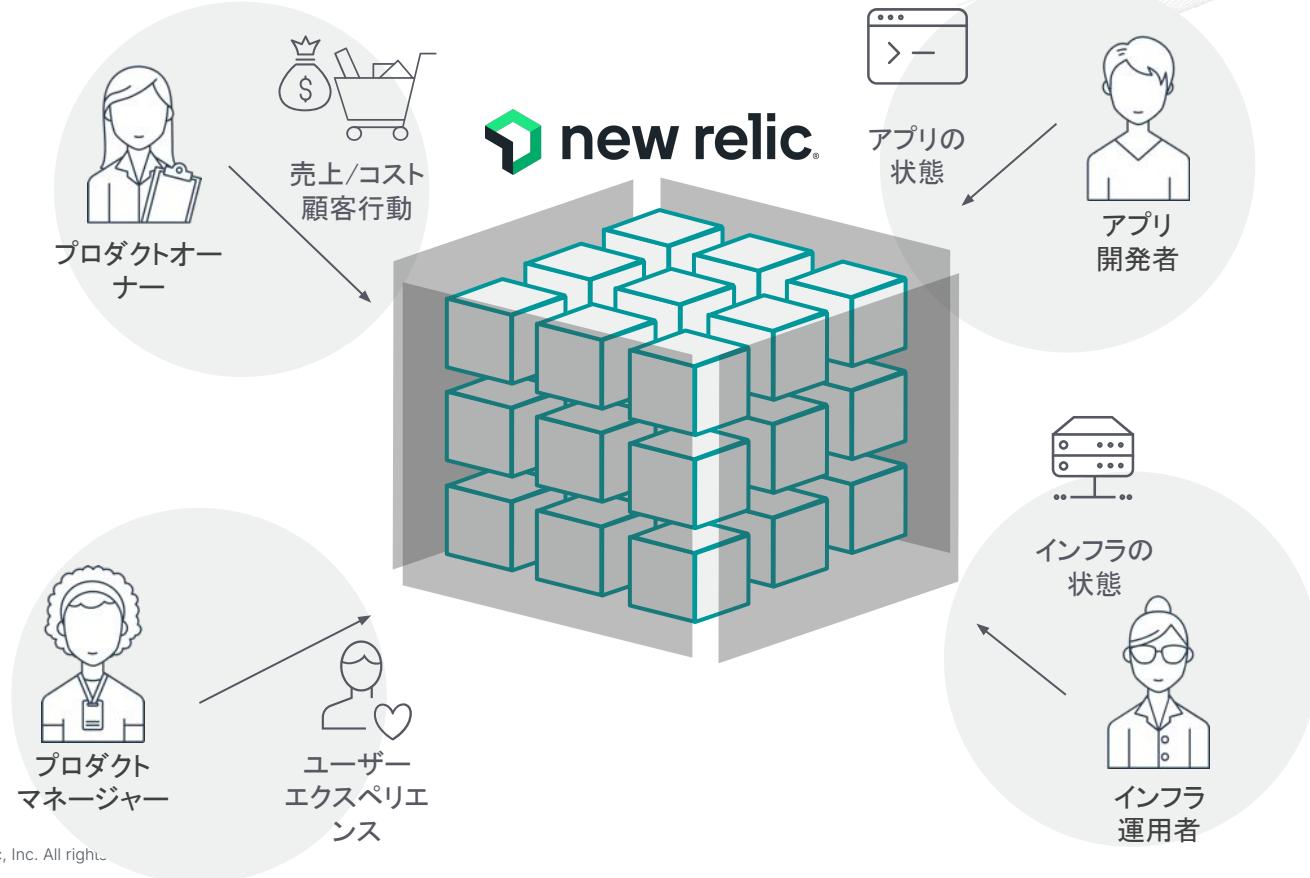
```
{  
    "appName": "front-service",  
    "clusterName": "newrelic",  
    "containerName": "front-service",  
    "databaseCallCount": 3,  
    "databaseDuration": 0.014582519,  
    "duration": 0.06666215,  
    "request.method": "GET",  
    "request.uri": "/",  
    "userId": "1234567890",  
    "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

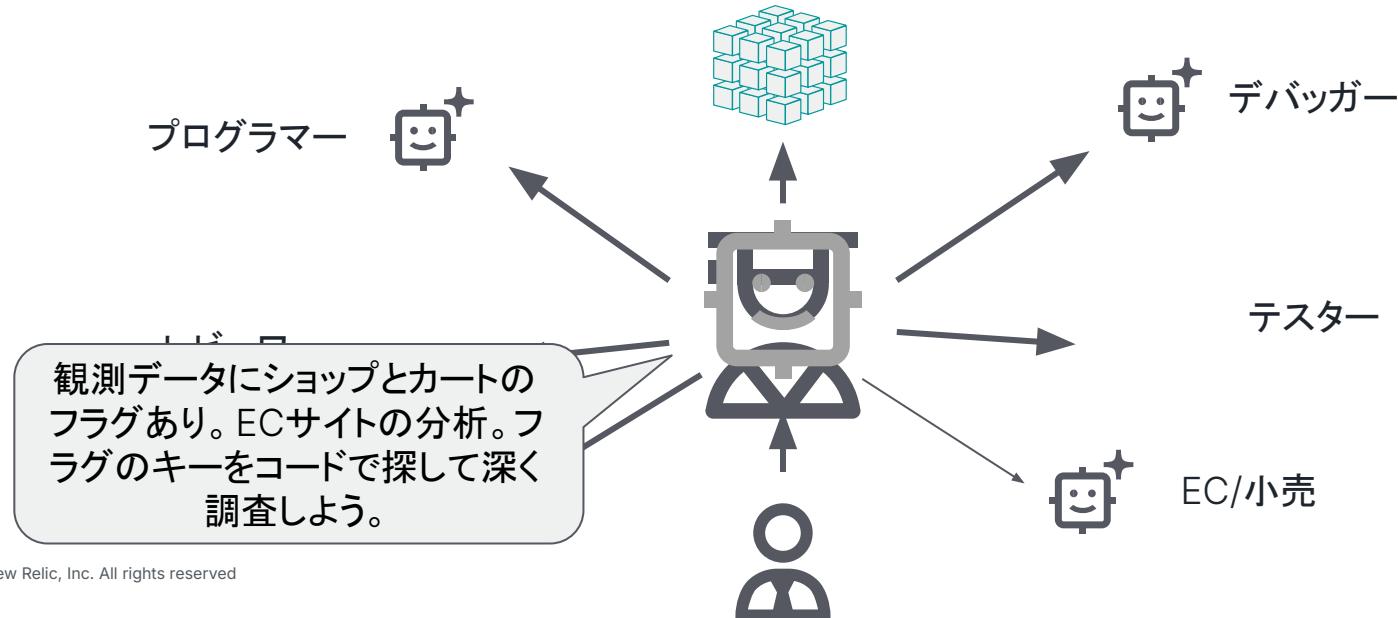
データ行を増やすのではなく、データの次元を増やすイメージ

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。
そうなれば、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。



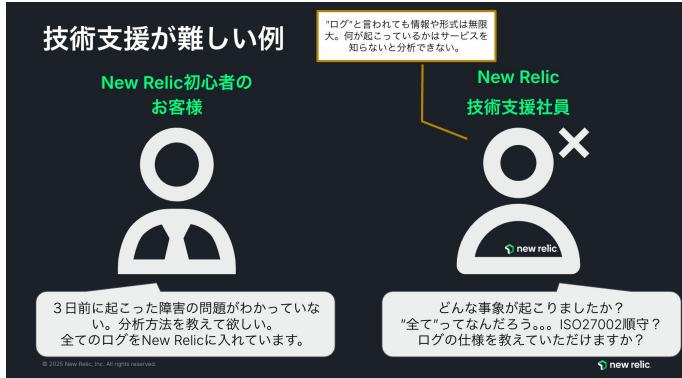
分析しやすいデータがあれば知識も少しで良い

"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。

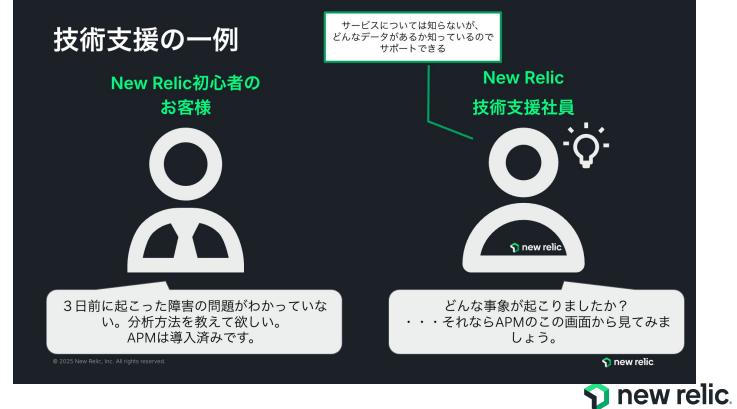
また、人間が見て分析が難しい情報は、AIが見ても難しいようです。



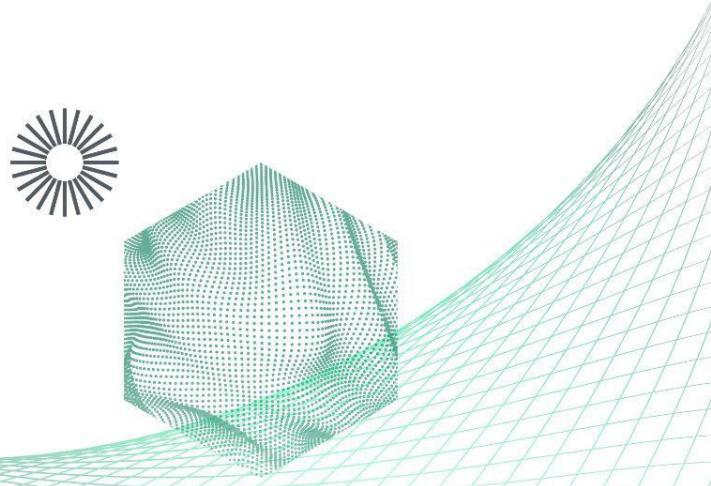
"意味がわかるデータ"を増やす

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。

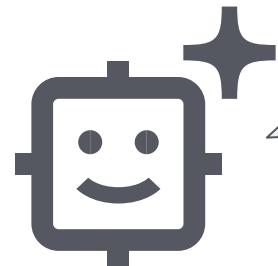


イベントデータを 充実
させるには





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。

The screenshot shows a developer's workspace with three main windows:

- ApprovalService.php**: An IDE editor showing the code for the `ApprovalService` class. The code uses Laravel's Eloquent ORM to query the `Approval` model based on various filters like `ids`, `user_id`, and `approver_id`. It includes comments in Japanese explaining the purpose of each method and parameter.
- Claude Code**: A terminal window titled "Claude Code" showing a welcome message and help information. The command `/help` is displayed, along with the current working directory (`cwd: /Users/tazuma/develop/github/Azumaxoid/nrkk-workflow`). A prompt at the bottom suggests trying to edit `DashboardController.php`.
- New Relic**: A monitoring interface showing real-time data for the application. The top bar displays metrics such as CPU, Memory, and Network usage. Below the bar, a timeline shows performance trends over time, with specific data points highlighted in green and red.

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parts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。

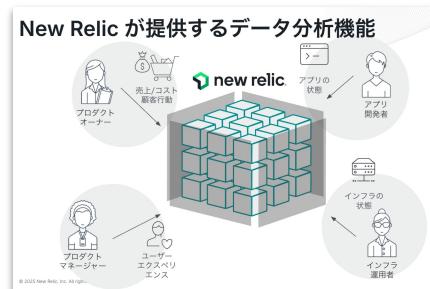
経験が浅くても活躍できるにはどんなことが必要かを考えてみましょう。



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

- 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

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

The screenshot shows the New Relic documentation landing page. At the top, there's a search bar and navigation links for 'Docs', 'Community', and 'Learn'. The main heading is 'New Relicのドキュメントへようこそ。' (Welcome to the New Relic Documentation). Below it, a large green box says '簡単な4ステップで開始' (Get started in 4 simple steps). The steps are:

- ① アカウントを作成
まだアカウントをお持ちではありませんか？
[こちらからサインアップ](#)
- ② New Relicの概要
New Relicの監視およびオブザーバビリティツールの詳細をご覧ください
[スタートガイド](#)
- ③ New Relicのインストール
UIに移動し、New Relicを設定して技術スタックを計装する
[インストール](#)
- ④ チュートリアル
New Relicを使用して現実的問題を解決する一般的な方法
[チュートリアルを読む](#)

Below these steps, there's a section titled '人気のドキュメント' (Popular Documents) with three items:

- すべてのアプリとサービスを監視
[詳細情報](#)
- ブラウザモニタリングをデプロイする
[詳細情報](#)
- Syntheticモニタリングを開始
[詳細情報](#)

ログイン後画面 (Integrations & Agents)

The screenshot shows the 'Integrations & Agents' dashboard after logging in. The left sidebar has a 'Quick Find' search bar and navigation links for 'All Capabilities', 'Catalogs', 'Dashboards', 'APM & Services', 'Logs', 'Traces', 'Synthetic Monitoring', 'Alerts', 'Infrastructure', 'Kubernetes', 'Browser', 'Mobile', 'Errors Inbox', 'Apps', 'On-Host Integrations', 'Teams', 'Help', and 'Add User'. The main area is titled 'Integrations & Agents'.

The 'Featured' section includes:

- New Relic Control** (NEW)
Simplify the deployment and management of telemetry data collection across your entire organization, regardless of the technology stack.
[Learn More](#) [See our docs](#)
- Pipeline Control**
Filter your data in your environment or in our cloud or before we ingest it.
[Set up Pipeline Control](#)

The 'Recommended' section includes four items:

- Guided install
- 30 coverage gaps
- Browser monitoring
- Java

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

