



ダイキン工業のDXから学ぶ これまでとこれからの IoTサービス開発と運用

Takuya Azuma

Takuya Azuma



New Relic – Senior Customer Success Manager 製造業担当

略歴:

- ERPパッケージベンダーで機能開発、基盤開発、クラウドサービス運用・SRE
キーワード : **AWS、K8S、Cassandra、APM・Zabbix・Prometheus・Grafana、Ansible、Lambda、Spark** 自動化
- IT コンサルとしてBtoCサービス構築支援
キーワード : **AWS、Heroku、CICD、モバイル**開発



現代のIoTサービス提供の流れ

- ✓ クラウドや手軽に利用できるサービスが発展し、構想～PoCが短期化
- ✓ 競争優位性のあるうちに拡大・マネタイズ

構想 ～ PoC

PoC ～ ローンチ

ローンチ ～ マネタイズ

現代のIoTサービス提供の流れ

開発の悩み

運用の悩み

構想 ~ PoC

PoC ~ ローンチ

ローンチ ~ マネタイズ

新規開発
スピード重視

機能/非機能開発
品質向上

継続的な改善

デジタルサービス運用準備
スケール前準備

無駄の削減
運用効率化

現代のIoTサービス提供の流れ

これまで

これから

開発の悩み

運用の悩み

構想 ~ PoC

PoC ~ ローンチ

ローンチ ~ マネタイズ

新規開発
スピード重視

機能/非機能開発
品質向上

継続的な改善

デジタルサービス運用準備
スケール前準備

無駄の削減
運用効率化





試行錯誤の力でビジネスに貢献する エンジニアチームへ。

“私たちの開発したアプリケーションがどれだけ多く
ユーザーに体験してもらえるか、現場のフィードバック
を受けながらいかに改善していくか、そこが私たちの
勝負です。”

片岡 太郎 氏
テクノロジー・イノベーションセンター
情報通信技術グループ

Challenge

ウェアラブルデバイスやサーバレスアーキテクチャなど先端的な技術で構築され、今後世界展開による大規模化が予測されるシステムを End-to-End で把握し、コードレベルで問題を特定すること。

Solution

地理的な要因を勘案せず、かつ IoT デバイスや Serverless 環境でも開発チームが性能管理可能な New Relic One を採用。

Impact

本番環境での運用を見据えた、改善や試行錯誤可能なソリューション全体の性能可視化能力を獲得



TDP



FSO



ダイキン工業のDXから学ぶ これまでとこれからのIoTサービス開発と運用

ダイキン工業株式会社 テクノロジー・イノベーションセンター
幸浦 祐作

- 幸浦 祐作（こうら ゆうさく）
- テクノロジー・イノベーションセンター所属
- 2016年 新卒でダイキン工業入社
- 入社～2018年 中国向けの空調機IoTシステムの設計開発
- 2019年～現在 コネクテッドワーカーソリューションの設計開発
主にTHINKLETのモバイルアプリ開発

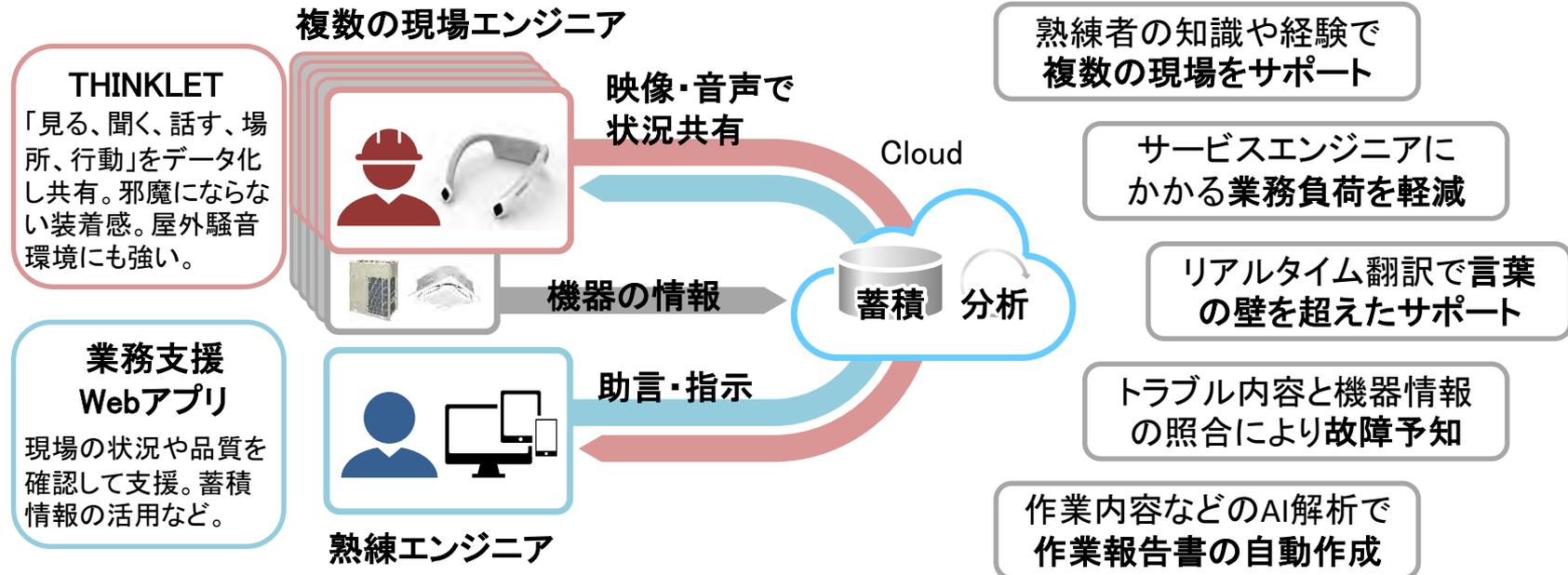


- ダイキンの取り組み紹介
- 開発・運用での課題とアプローチ
- これから取り組んでいくこと
- まとめ



ダイキンの取り組み紹介

現場作業者と遠隔地の熟練者が、THINKLETと業務支援Webアプリを用いて通信を行い、指導や助言を得ることにより、経験の浅いサービスエンジニアでも作業フローや使用部材の管理方法、環境負荷低減のための冷媒量管理方法、故障診断のポイントなどを迅速／的確に把握できる遠隔作業支援ソリューション



作業内容や現場の状況を遠隔地と共有し、作業に関する全ての情報をデータ化することで、作業者の記憶や知識の再活用を可能とするウェアラブルデバイス



特徴

- 5個のマイクを搭載し、装着者の音声だけでなく**対面者の音声も同時に収録**
- 800万画素の超広角カメラを搭載し、**クリアで精細な作業時の映像を収録**
- IP54の防水防塵性能で、**雨の中で使用可能**
- 人間工学に基づいた軽量首掛け型デザインで、**装着者の手を塞がず作業の邪魔をしない**

作業内容や現場の状況を遠隔地と共有し、作業に関する全ての情報をデータ化することで、
作業者の記憶や知識の再活用を可能とするウェアラブルデバイス

現在このTHINKLETを国内の各拠点に配布し、
さまざまな環境でフィールドテストを実施している

国外含め1000台単位で配布済みで、
さらなる改善・改良を進めていく

特徴

■ 両目カメラを搭載し、装着者の音声だけでなく、
対面者の音声も同時に収録

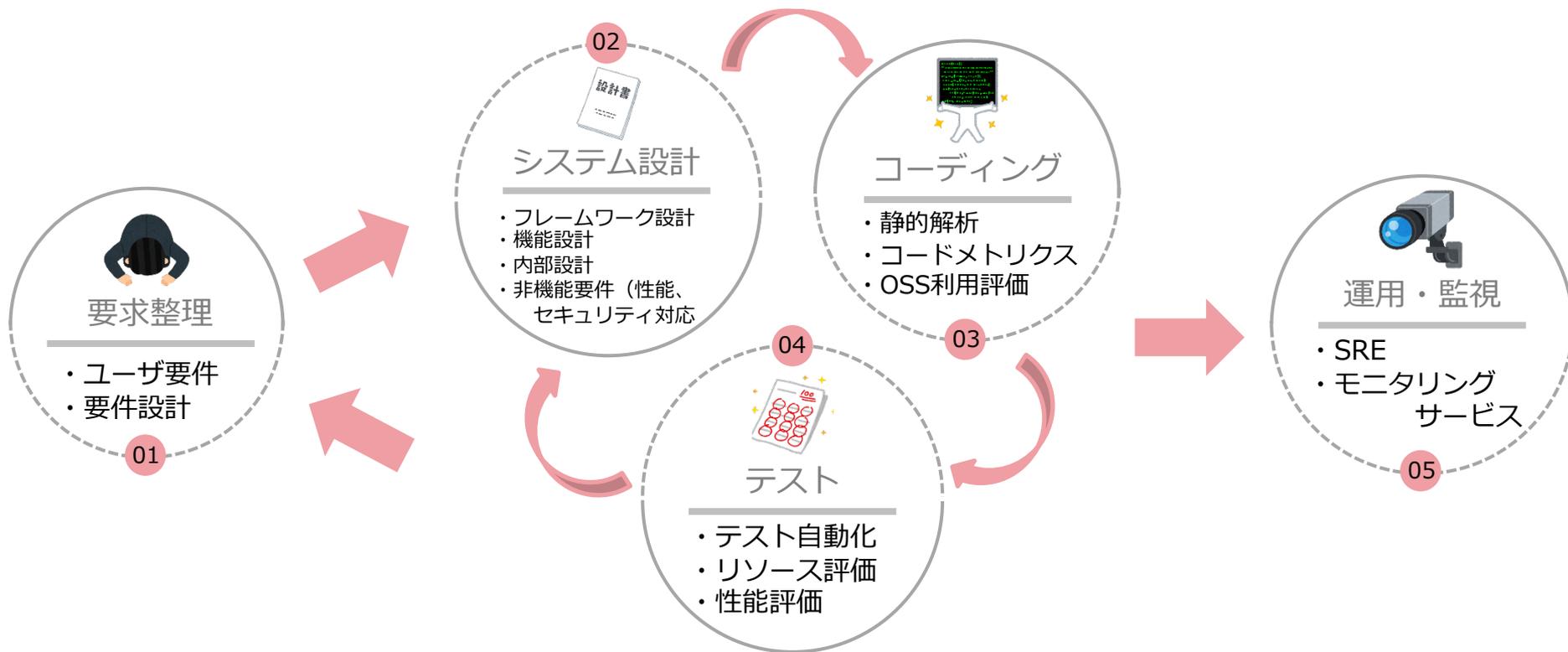
■ 800万画素の超広角カメラを搭載し、リア
で精細な作業時の映像を収録

■ IP54の防水防塵性能で、雨の中で使用可能
■ 人間工学に基づいた軽量首掛け型デザインで、
装着時の負担を軽減

■ 作業中の邪魔をしない



開発・運用での課題とアプローチ



プロセス管理、構成管理、CI/CD



課題①

さまざまなドキュメントがあらゆる場所に点在し、どのバージョンが最新なのかわからない

原因

- ドキュメントの体系・ルール・バージョンがきちんと管理されていない

解決策

- AsciiDocを使って、ドキュメントもソースコード化して管理する



AsciiDoc

- 軽量マークアップ言語
- 出力もHTML, PDF, ePubなどが行え、個人出版から開発ナレッジ共有まで様々な用途で利用可能

```
== 改訂履歴

[cols="1,4,1,1", options="header"]
|===
| 版
| 内容
| 更新者
| 日付

| 1.0
| 初版登録
| 宇浦
| 2021.03.01

|
|
|
|===

== はじめに

image::./image/asciidoc.png[]

本書は、GitレポジトリでAsciiDocを用いて設計書を記述する際のサンプルをまとめたレポジトリである。

== URLの記載

[diatum], diagram-classes, png]
....
include::./diagram/sample.puml[]
```

- ソースコードと同様にGitで管理が可能
 - バージョン、編集者の管理が容易に
- レビューもPull Requestで実施
 - 更新するためには管理者のレビューが必要
- GitHub Pagesを利用して、Webブラウザから閲覧可能

課題②

ビルド・デプロイやインフラ構築、コード管理など、実装以外に大きく工数がかかる

原因

- 上記作業を手作業で実施している
- 開発プロセスに関するルールが明文化されていない

解決策

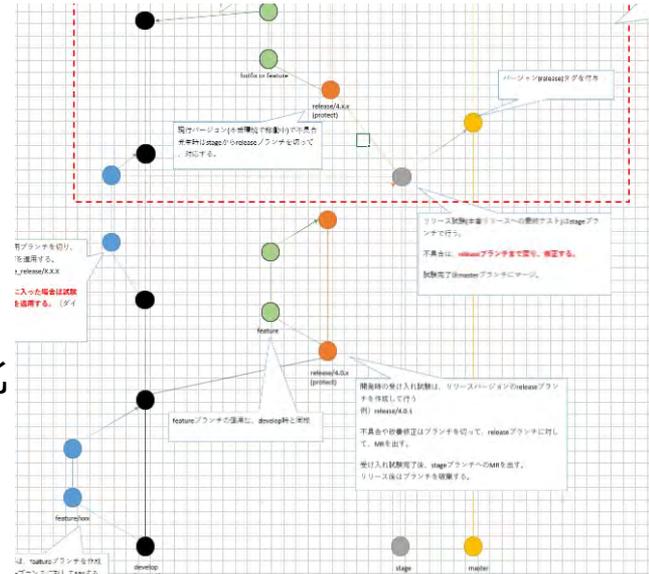
- CI/CDを取り入れる
- 開発のルールをきちんと定義する

-開発プロセスの明文化

- Gitフローを図示
 - ブランチの種類、承認フロー・担当者等を明示
 - 正式リリース時は、手動で確認後にデプロイ

- CI/CDの導入

- GitLab CIを利用して、コミットからデプロイまでを自動化
- アプリだけでなく、インフラもIaC化して同様に管理



Ver5.0.0リリース

2 jobs for release/5.0.0 in 5 minutes and 30 seconds (queued for 2 seconds)

87748673

No related merge requests found.

Pipeline Needs Jobs 2 Tests 0

Build Deploy

release_build release_deploy

課題③

各担当者がどんな仕事をやっていて、今何で困っているのかなどの状況が共有されていない

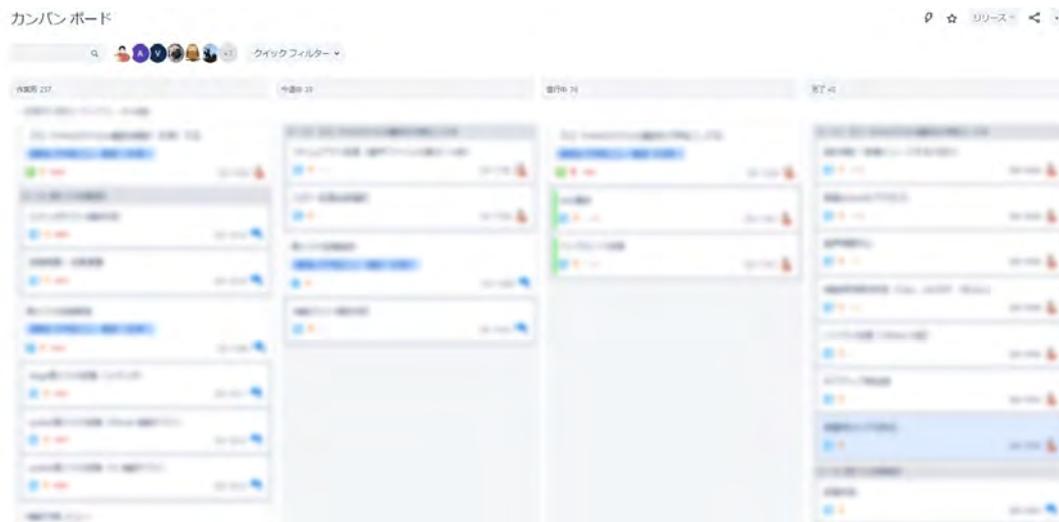
原因

- それぞれの状況をチーム内で共有できていない
- 自分の業務に手一杯で、他人の業務を見る暇がない

解決策

- チーム内で現状共有する場を作る
- 会議の議事録を全員が見えるように共有する

- 週次定例の場を作り、現状を報告する
 - タスク管理ツールのJiraを使って、業務の進捗が見える化



- 顔文字で自分の進捗状況や自信を可視化

先週のふりかえり



←順調 普通 ヤバい→

今週の予定



(←自信满满 不安一杯→)

- 開発会議の書記担当をローテーションする
 - 参加せざるを得ない環境を作る
- 全ての議事録をConfluenceのチームスペースで共有
 - 他業務都合で参加できなかった会議の議事録も検索可能
 - 議事録だけでなく、開発中に調べた内容やTipsなども共有
- インセプションデッキ作成
 - チームとしての方向性を見失わないように
 - 月1回メンテナンスを実施
- スプリントごとに開発の振り返りを実施
 - ベンダーさんも交えて
 - 隔週でミニ振り返りも実施

①共有時間の膨大化

→個人の作業時間と共有時間のバランス
を取れる仕組みづくり

②定例会のフォーマットが定まっていない

→Try&Errorを繰り返し、チームに合った
定例会の型作り

06_ふりかえり目安箱

Try

- 定例会のルールでIssueがプロットされている。
- 議題・議題、ではメンバーが中心の発言、サービスマンが中心
- 既にすでにあるフレームワークがあるようなワークを作成する（まずはコンフレット）
- 定例会・コンフレット・メンバーが参加しやすくなる仕組みづくり（定例）

Keep / Good

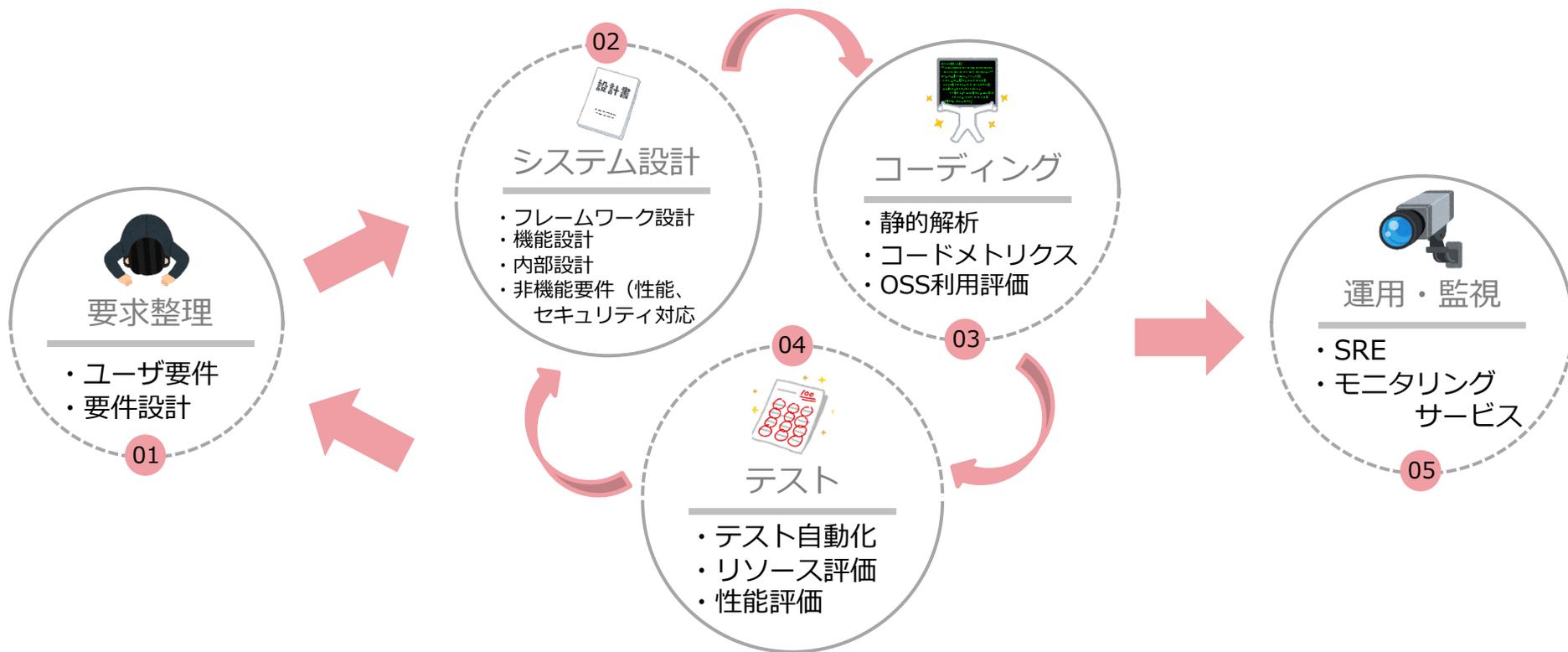
- 定例会のことをスクリプト化する過程（Issueから、ユーザー体験をIssueで定義し）
- 定例会のAgendaを定めておくことで、これはチームのAgendaをチームメンバーに共有して書いてもらう、みたいな定めておく
- 定例会のルールは、定例会でIssueとProblemを定義してあり
- インベントリ・チェックの共通化も定めてやる。
 - 定例会にやるルーティンにする
- 「定例会」フレームワークを作る
- 定例会の定例会について定例会で確認する

Problem

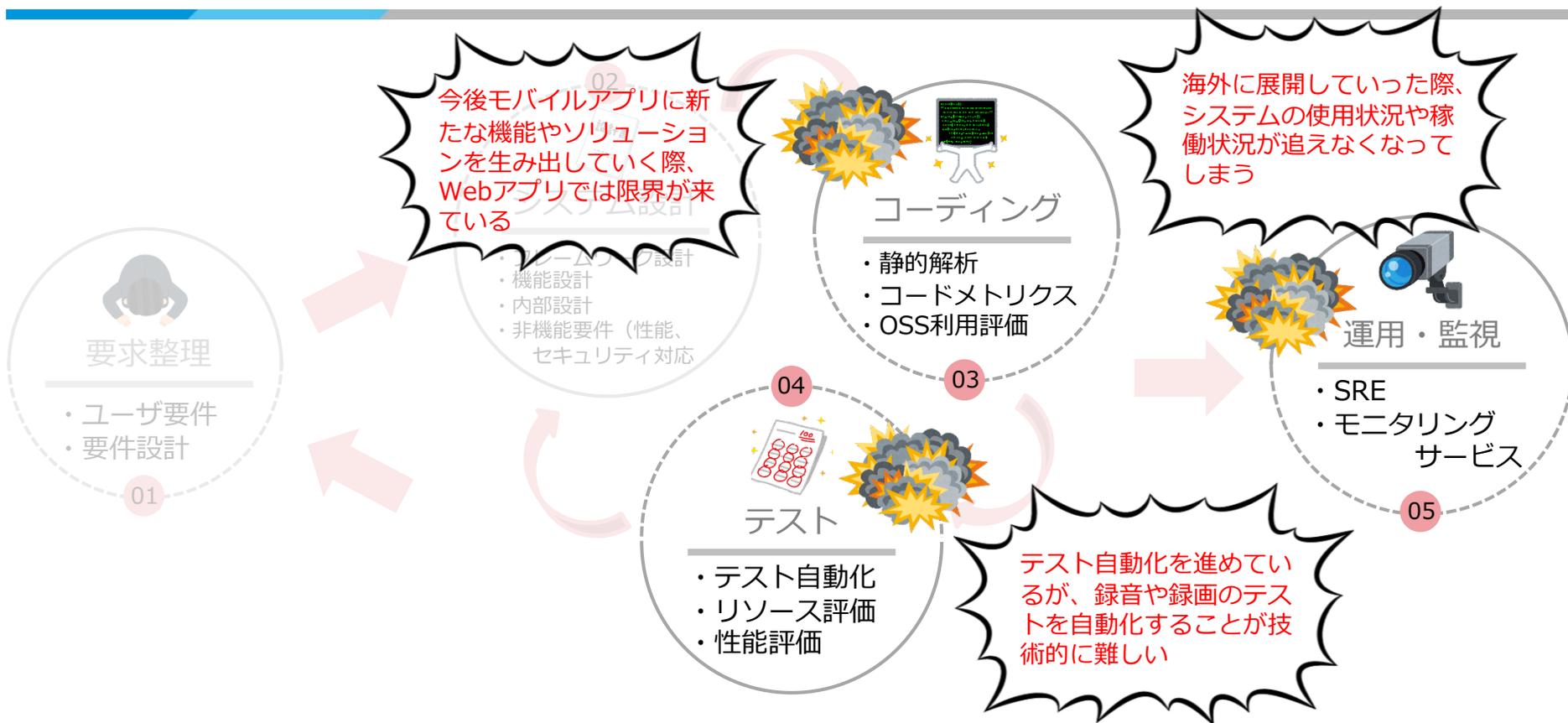
- 定例会のルールを定例会が定めてしまう、すでにあるフレームワークがない
- スター・メンバーが定例会の定例会を定例会の定例会定めない
 - 定例会の定例会（定例会・定例、定例会の定例会）になると、メンバーは定例会定めない
 - 定例会の定例会に定例会定めない
 - 定例会の定例会を定例会定めない
- 定例会の定例会定めない定例会の定例会定めない
 - Issue定例、定例会



これから取り組んでいくこと



プロセス管理、構成管理、CI/CD



とある現場とのやり取り



THINKLETの動きがおかしいから見てほしい



わかりました。ログを確認するので
THINKLETをごちに送付いたします

~~~~開発部門にTHINKLET到着~~~~



アプリにバグがありましたので、  
次のバージョンで修正します



修正版がリリースされるまで、  
〇〇はしないように運用いたします  
すぐにTHINKLETをお返しします

~~~~現場にTHINKLETが戻る~~~~

とある現場とのやり取り



THINKLETの動きがおかしいから見てほしい



わかりました。ログを確認するので
THINKLETをごちに送付いたします

〜〜〜開発部門にTHINKLET到着〜〜〜



アプリにバグがありましたので、
次のバージョンで修正します



修正版がリリースされるまで、
〇〇はしないように運用いたします
すぐにTHINKLETをお返しします

〜〜〜現場にTHINKLETに戻る〜〜〜

バグ発見から確認まで
およそ2週間かかる

郵送する手間や費用も増え、
バグが多発するPoCでは致命的

- 今後海外拠点に展開していくにあたって、確実にクリアしないとイケない大きな課題
 - 輸送コスト・リスクが高い
 - リードタイム、輸送費、貿易管理、etc
 - 管理が難しい
 - 1000台単位のTHINKLETが、各海外拠点の拠点長→営業店→リーダー→作業員とリレーのように渡る
 - どの拠点で誰がどのTHINKLETを使っているかを現場で判断するのは不可能

- 今後海外拠点に展開していくにあたって、確実にクリアしないといけない大きな課題
 - 輸送コスト・リスクが高い
 - リードタイム、輸送費、貿易管理、etc
 - 管理が難しい
 - 1000台単位のTHINKLETが、各海外拠点の拠点長→営業店→リーダー→作業員とリレーのように渡る
 - どの拠点で誰がどのTHINKLETを使っているかを現場で判断するのは不可能



遠隔からTHINKLETの状態を監視し、
見える化するシステムが必要

- 今後海外拠点に展開していくにあたって、確実にクリアしないといけない大きな課題
 - 輸送コスト・リスクが高い
 - リードタイム、輸送費、貿易管理、etc
 - 管理が難しい
 - 1000台単位のTHINKLETが、各海外拠点の拠点長→営業店→リーダー→作業員とリレーのように渡る
 - どこの拠点で誰がどのTHINKLETを使っているかを現場で判断するのは不可能



遠隔からTHINKLETの状態を監視し、
見える化するシステムが必要

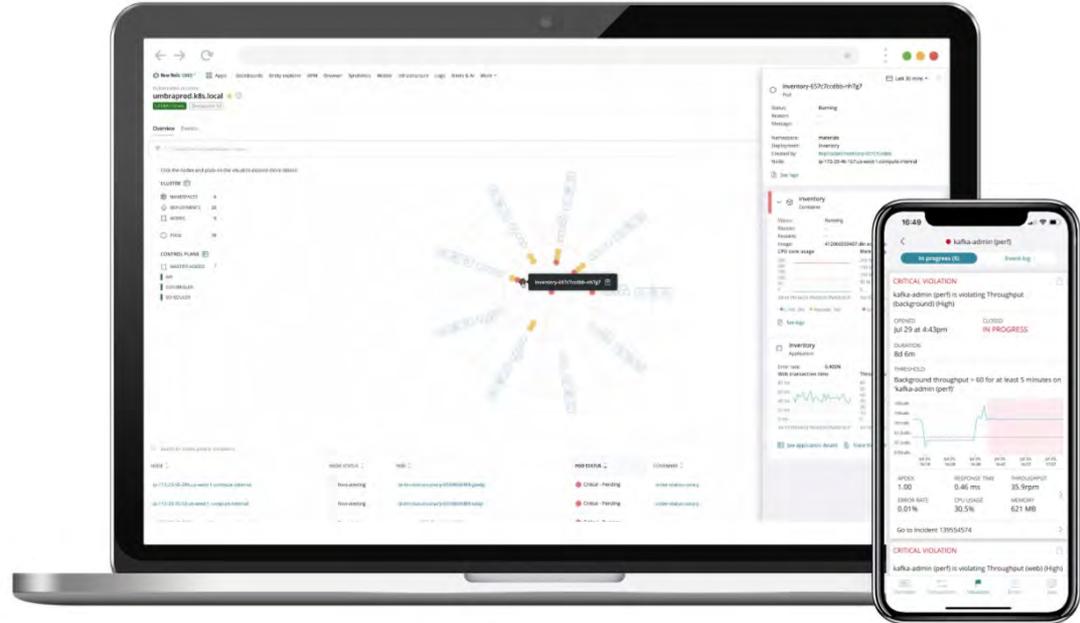


不具合を即時検知・修正できるような
オブザーバビリティを実現する

フロントエンド（モバイル・Web）からバックエンドまで、一つのシステムとして一括で監視可能

それぞれのデータをアプリケーションごとにカスタマイズして、各事業所向けの様々なダッシュボードを作ることが可能

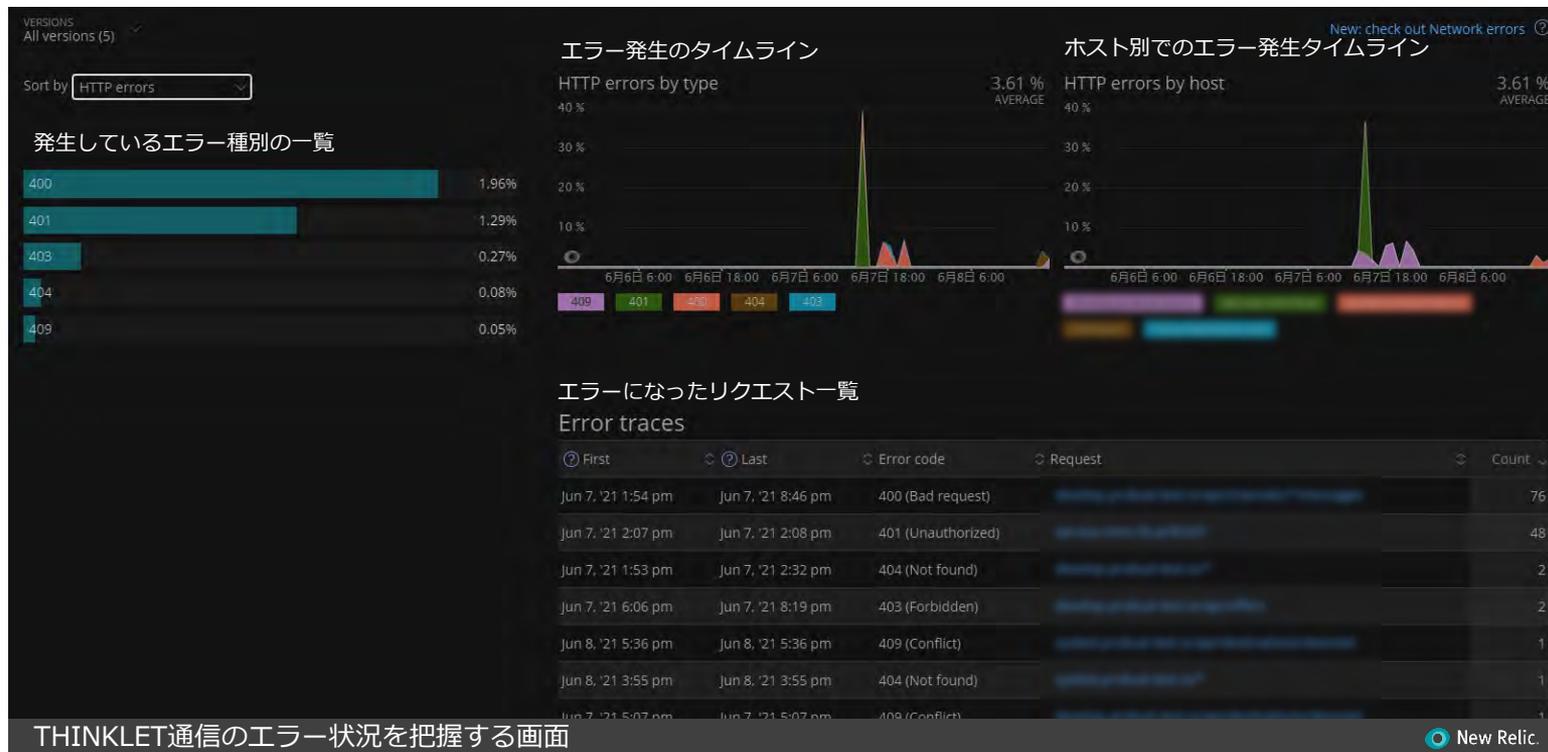
現在は…
システムのフロントエンド、バックエンド、モバイルの各アプリケーションへの組み込みと各拠点本部向けのダッシュボードの作成を実施し、国内でのPoCで運用中



Observability Platform である New Relic One の画面。表示は Kubernetes Cluster Explorer の UI。システムのリアルタイム性能分析を可能にする。SaaS なのでモバイルでも同様のデータ分析が可能。

- 不具合の迅速な発見と修正

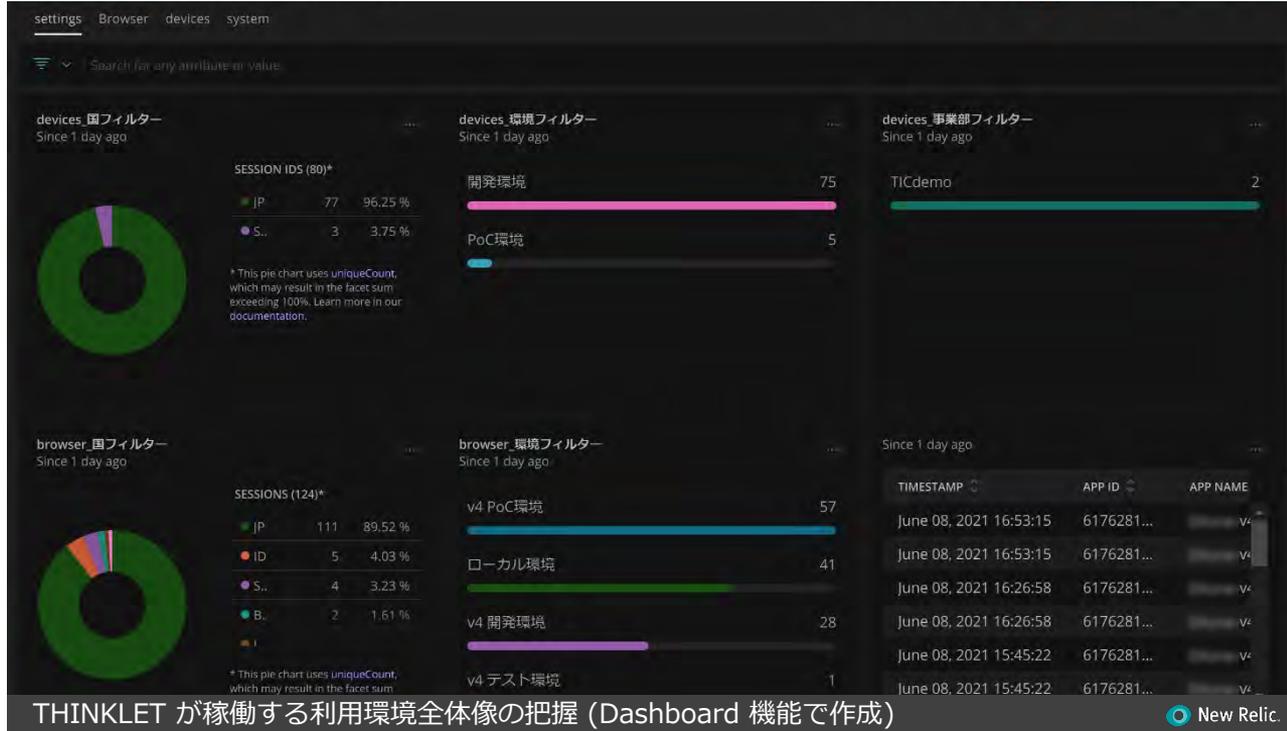
- 現場から報告があった大きな不具合はもちろん、使用者では見えないような軽微なバグまで検知し、修正を行う



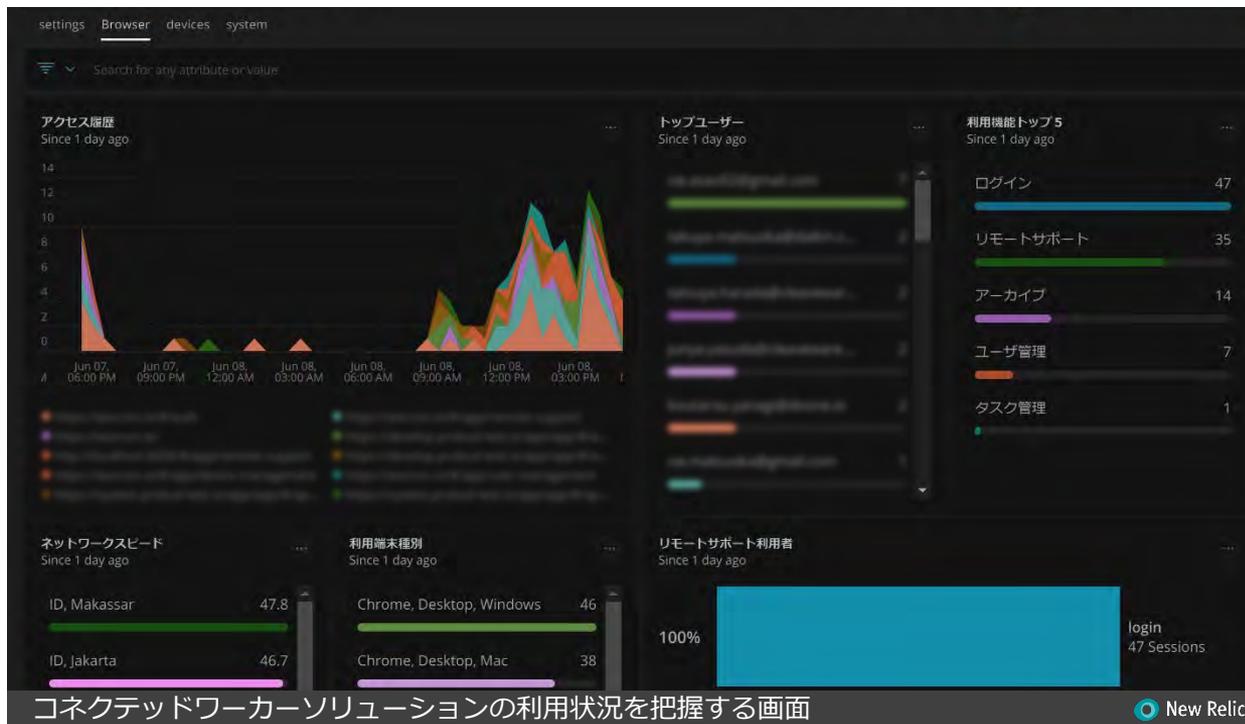
THINKLET通信のエラー状況を把握する画面

- 稼働率・通信状況などのメトリクスの収集

- どれぐらいの人に使われていて、それぐらいのパフォーマンスが出ているのかを収集し、サービスの改善や見直しに役立てる



- 現場のマネージャーがダッシュボードから稼働状況を確認
 - 予定通り現場で動いているか、不具合が起きていないかなどを確認し、開発チームに即座に報告



- 不具合の迅速な発見と修正
 - 現場から報告があった大きな不具合はもちろん、使用者では見えないような軽微なバグまで検知し、修正を行う
- 稼働率・通信状況などのメトリクスの収集
 - どれぐらいの人に使われていて、それぐらいのパフォーマンスが出ているのかを収集し、サービスの改善や見直しに役立てる
- 現場のマネージャーがダッシュボードから稼働状況を確認
 - 予定通り現場で動いているか、不具合が起きていないかなどを確認し、開発チームに即座に報告

**大規模な環境で高いサービス品質を維持するための
より高度なオブザーバビリティの実現**

○現在のベーシックな機能

遠隔支援機能

現場から離れた熟練者とハンズフリーでビデオ通話

アーカイブ機能

遠隔支援で録画した動画を閲覧

現場エンジニア



映像・音声で
状況共有



助言・指示

Cloud

熟練エンジニア

○現在のベーシックな機能

遠隔支援機能

現場から離れた熟練者とハンズフリーでビデオ通話

アーカイブ機能

遠隔支援で録画した動画を閲覧

→ Webアプリとして実現

現場エンジニア



映像・音声で
状況共有



熟練エンジニア

助言・指示

Cloud

○現在のベーシックな機能

遠隔支援機能

現場から離れた熟練者とハンズフリーでビデオ通話

アーカイブ機能

遠隔支援で録画した動画を閲覧

→ Webアプリとして実現

現場エンジニア



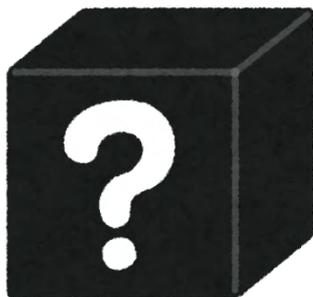
映像・音声で
状況共有



熟練エンジニア

助言・指示

Cloud



もっと高度な機能（ビデオ・音声を加工するなど）になると、Webアプリでは実現が難しい（特にiOS）

○現在のベーシックな機能

遠隔支援機能

現場から離れた熟練者とハンズフリーでビデオ通話

アーカイブ機能

遠隔支援で録画した動画を閲覧

→ Webアプリとして実現

現場エンジニア



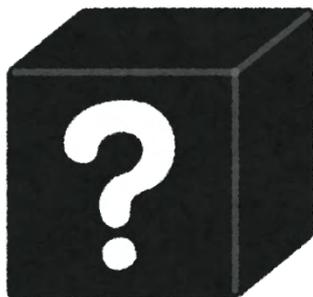
映像・音声で
状況共有



熟練エンジニア

助言・指示

Cloud



もっと高度な機能（ビデオ・音声を加工するなど）になると、
Webアプリでは実現が難しい（特にiOS）

より高度な機能開発のために
ネイティブアプリ化は必要不可欠

- 映像の品質をどうチェックするか
 - 判定方法や判定基準など
- 異常系のテストをどう行うか
 - ネットワーク切断など
- 物理ボタン起動であるTHINKLETをどのようにして自動テストするか
 - 内部的に起動することはできるが、それでテストできたと言えるのか

- 映像の品質をどうチェックするか
 - 判定方法や判定基準など
- 異常系のテストをどう行うか
 - ネットワーク切断など
- 物理ボタン起動であるTHINKLETをどのようにして自動テストするか
 - 内部的に起動することはできるが、それでテストできたと言えるのか

テストに関する技術課題は多いが
機能開発と同時並行で調査を進めていく



まとめ

- ダイキンではTHINKLETというウェアラブルデバイスを用いて、現場作業者と熟練者をつなぐコネクテッドワーカーソリューションを拡大している
- これまでに解決してきたこと
 - JiraやConfluenceを用いて業務・進捗の共有化
 - GitLab CIを用いたCI/CDとそのルール作り
 - AsciiDocを用いたドキュメント管理
- これから取り組んでいくこと
 - NewRelicを用いた更に高度なオブザーバビリティの実現
 - ネイティブアプリとしてさらに高度な機能開発
 - 遠隔支援機能に対する自動テストの技術確立



現代のIoTサービス提供の流れ

これまで

これから

開発の悩み

運用の悩み

構想 ~ PoC

PoC ~ ローンチ

ローンチ ~ マネタイズ

新規開発
スピード重視

機能/非機能開発
品質向上

継続的な改善

デジタルサービス運用準備
スケール前準備

無駄の削減
運用効率化





ユーザー体験からインフラまで
すべてを追跡。システムを横断
的に分析することを可能にする

