

Cloud Operator Days Tokyo 2026「運用者に光を！AIとどう向き合うか」 AI時代のSRE・運用を変える —SECIモデルでまわすチーム変革の実践

KDDI株式会社

コア技術統括本部 オペレーション本部 クラウドエンジニアリング部

木村 明雅

2026/07

1

オープニング

2

AIを使いはじめた後に見えてくる、チームの課題

3

AI時代のSECIモデルでSRE・運用を変える

4

SECIを加速するためのAI活用事例

5

まとめ

留意事項

- 本資料に記載された内容は、発表者個人の見解に基づくものであり、所属する組織の公式見解を示すものではありません。
- 本資料の内容は発表時点の情報に基づくものであり、今後変更される可能性があります。
- 本資料に含まれる技術情報や事例は一例であり、特定の環境における動作や効果を保証するものではありません。
- 本資料では説明の都合上、内容を一部簡略化・一般化している場合があります。

オープニング

今日お話しすること



**SRE・運用を変えるカギは、AIそのものではなく、
AIを組み込んだチームの知識循環、成長のデザインにある**



木村 明雅

Kimura Akimasa

KDDI株式会社
クラウドエンジニアリング部
プライベートクラウドG
Embedded SRE
チームリーダー

ミッション

クラウド技術とAI&ソフトウェア技術を駆使して
お客様に安定したサービスをお届けする

主な業務

- ① プライベートクラウドおよびコンテナ基盤の運用、
仮想インフラの設計・構築・運用(Embedded SRE)
- ② 監視・自動化・標準化による運用改善とCI/CD整備
- ③ AIOps / AI駆動開発の推進・整備

関心領域

SRE / AIOps / オブザーバビリティ / チームビルディング

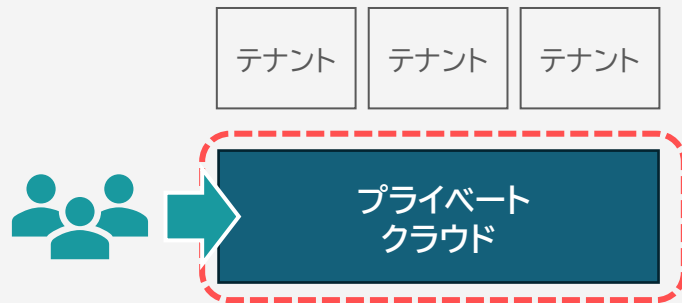
我々のチームのあゆみ

プライベートクラウドの運用から出発して、社内におけるSREの役割を段階的に拡張
仮想インフラの設計・構築からコンテナ基盤まで提供範囲を広げ、システムの安定運用を推進

2018年～

プライベートクラウド運用

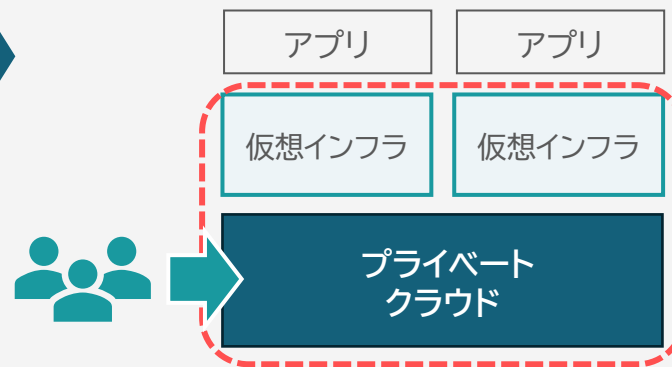
70を超えるテナントシステムを抱える
プライベートクラウドのインシデント・障害対応、
テナント申告対応、HWメンテ、運用改善



2020年～

仮想インフラ構築

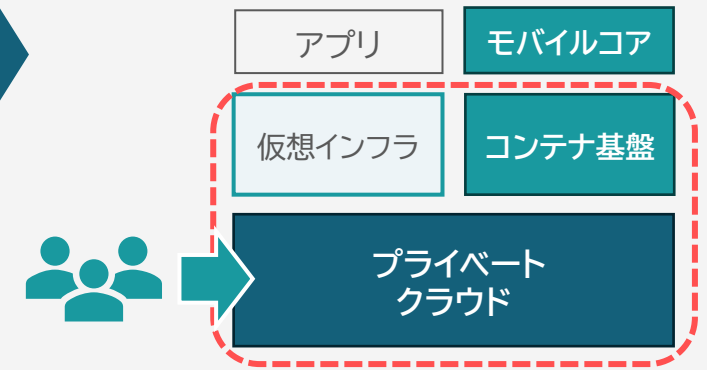
IaCを使ったプライベートクラウド上の
サーバ・ネットワークといった仮想インフラの
設計・構築・運用(Embedded SRE)



2024年～

モバイルコア (コンテナ基盤運用)

30を超える社内システムのSRE、
モバイルコア向けコンテナ基盤の運用



→ システムの大規模化・複雑化

AIを使いはじめた後に見えてくる、チームの課題

AIを入れても、運用があまり変わらないケース

個人利用やワンショットの活用に留まるケース

- プロンプトは各自のメモ帳に眠ったままで
便利な使い方が共有されない
 - AI活用の経験差により、活用度にばらつきが出る
 - レビュー用エージェントを作ったが結局使われず...
- 
- ベテランのノウハウやコツがAIにうまく適用できない
 - チーム全体でToilがなかなか減らない
 - 新人のオンボーディングにつまずく

SRE・運用に必要なこと

- 実践を通して新しい知見を獲得する
- 現場で得た知見をチームのナレッジにつなげる
- チーム全体のToilが削減される
- 運用プロセス全体がスムーズに回る
- チームの持続的な成長

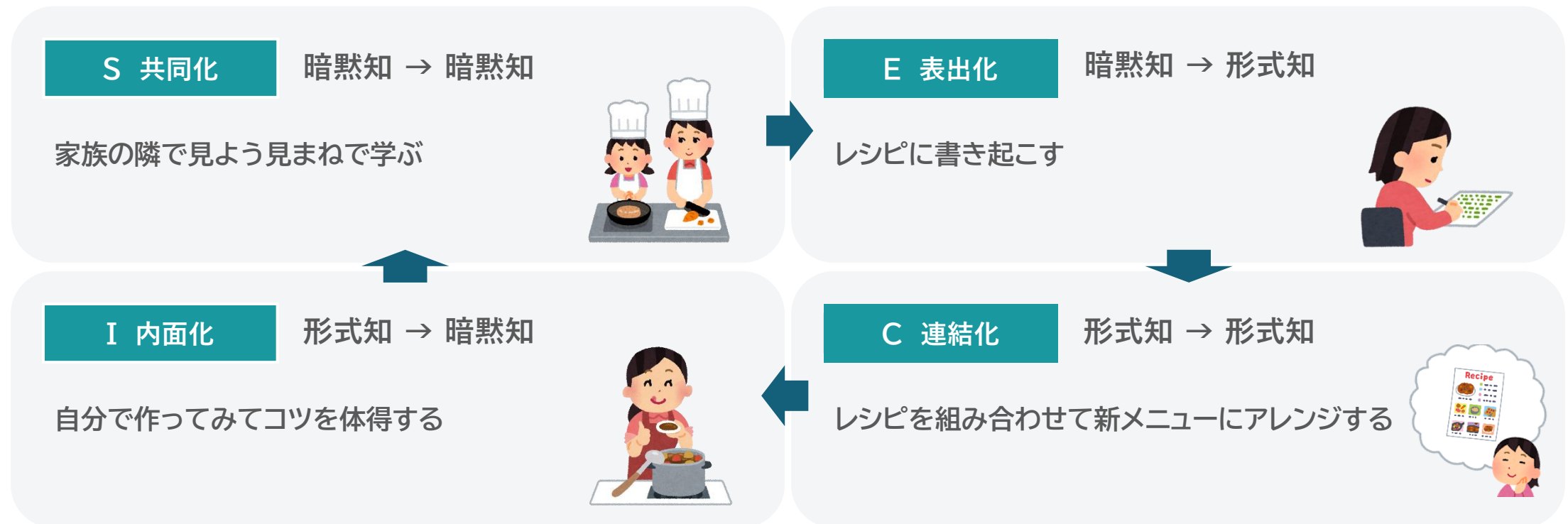


問題はAIやツールではなく、「経験やノウハウがチームに循環していないこと」だった
そこで着目したのが、日本発の知識創造理論

SECIモデルとは — 料理のレシピと家庭の味

SECIモデルは、暗黙知と形式知の相互変換により、チームで知識が活用され生み出されていくプロセス

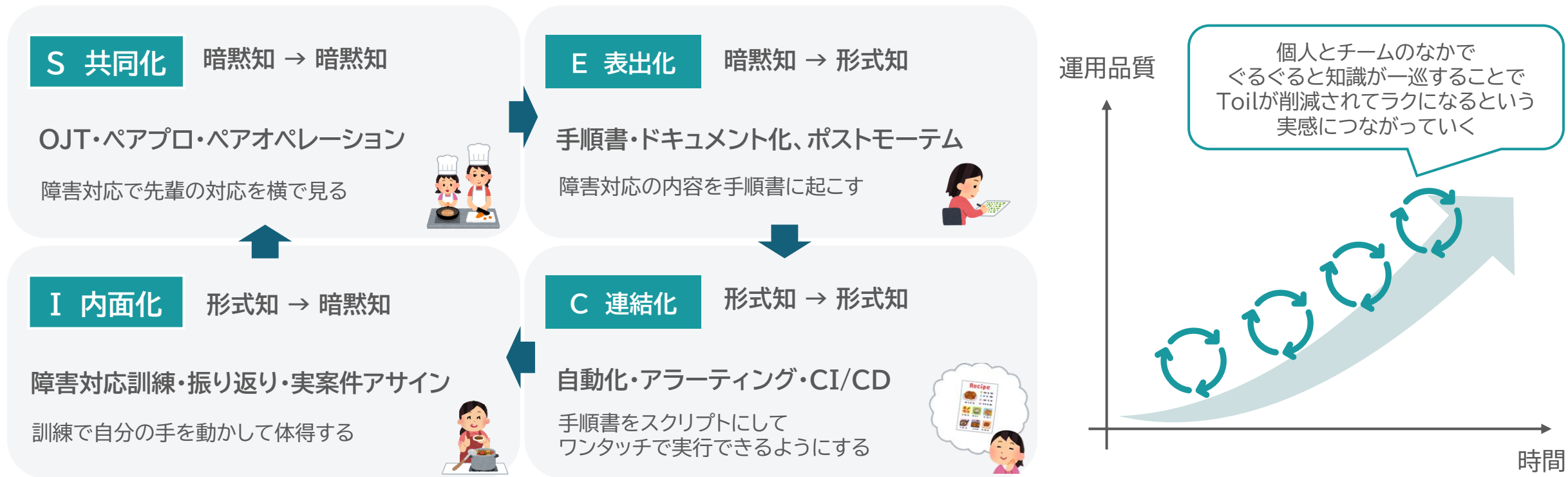
暗黙知: 個人の経験・勘・コツなど言語化が難しい知識、形式知: 文書・図・手順書など共有できる知識



SRE・運用の現場でも、我々のチームではゆっくりでも無意識にSECIを回し続けていた

SRE・運用におけるSECIモデル

すぐに運用を変えるのは無理だが着実にサイクルを回すことで業務が改善されていく



しかし、現実問題としてうまくいかないことが多い

例: 「忙しくて手順書の作成が後回しになってしまう」「そもそも自動化に必要な知識が足りない」

現場で起こりがちな4つの詰まり

SRE・運用におけるSECIのスパイラルを阻む、詰まりのパターン



学習・育成の不足

学ぶ・育てる時間がない

「新人が入ったけど、面倒を見る時間がなかなか取れない」



属人化

ベテラン依存とノウハウ・コツの散逸

「あの対応は〇〇さんしかできない」



Toil

稼働を消費する繰り返し作業

「サーバ故障の対応で毎回数時間取られてしまう」



アラート疲れ

調査・分析・対応・記録に稼働を取られる

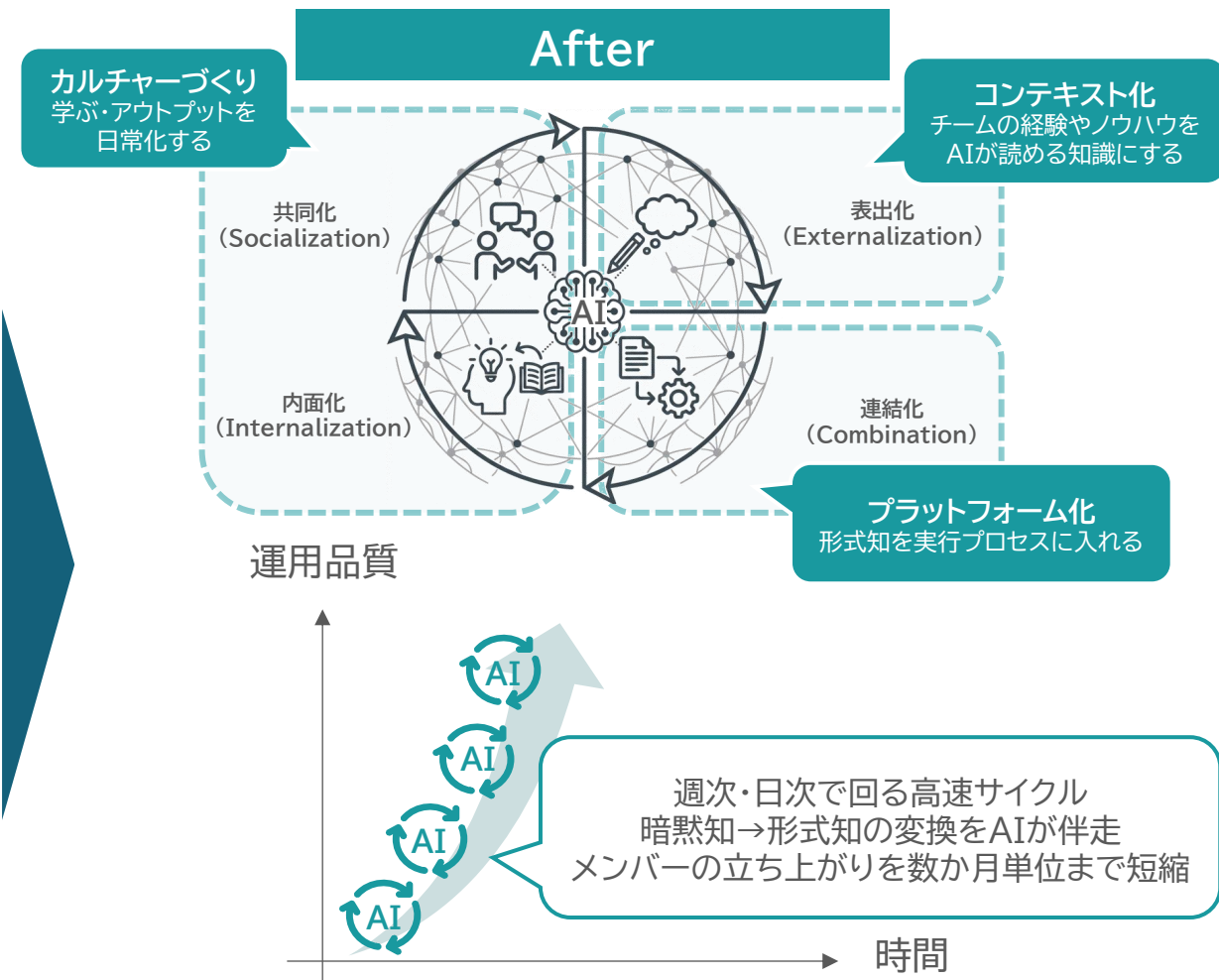
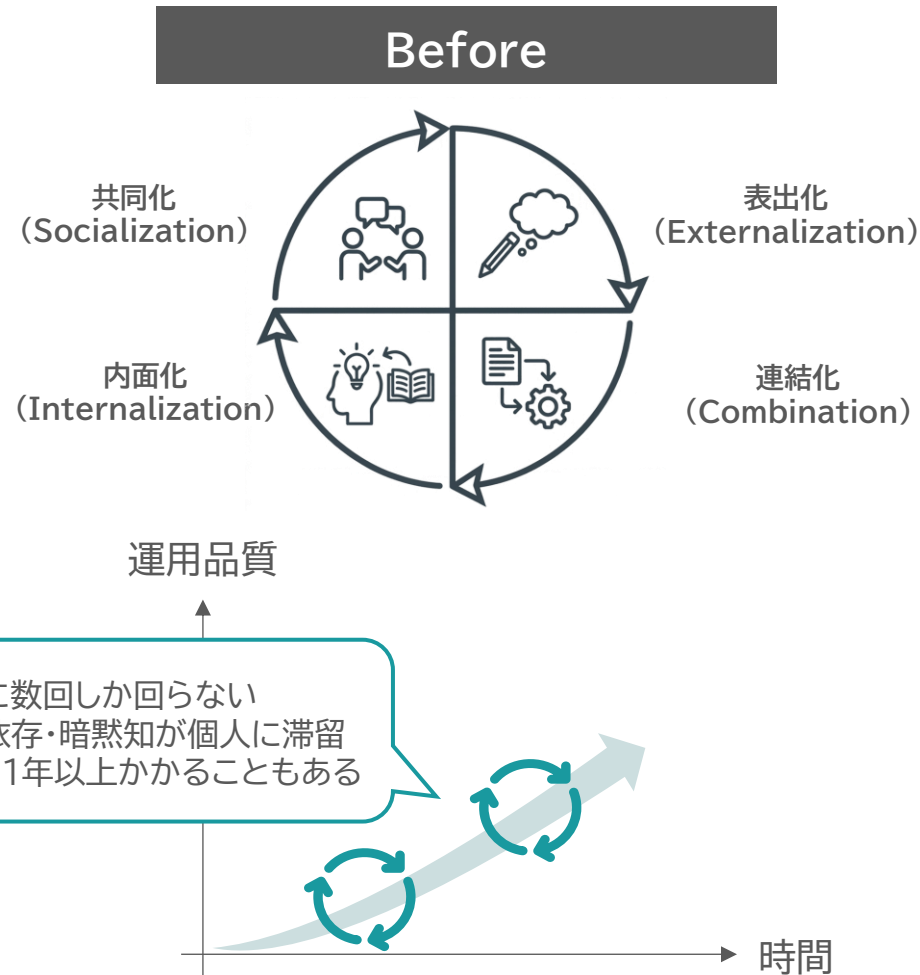
「調査してみたけど誤検知アラートだった」

システムの複雑化・大規模化に対して、サイクルの速度が現場の負荷と暗黙知の量に追いつかない
この「詰まり」を解消するためにAIを活用する

AI時代のSECIモデルでSRE・運用を変える

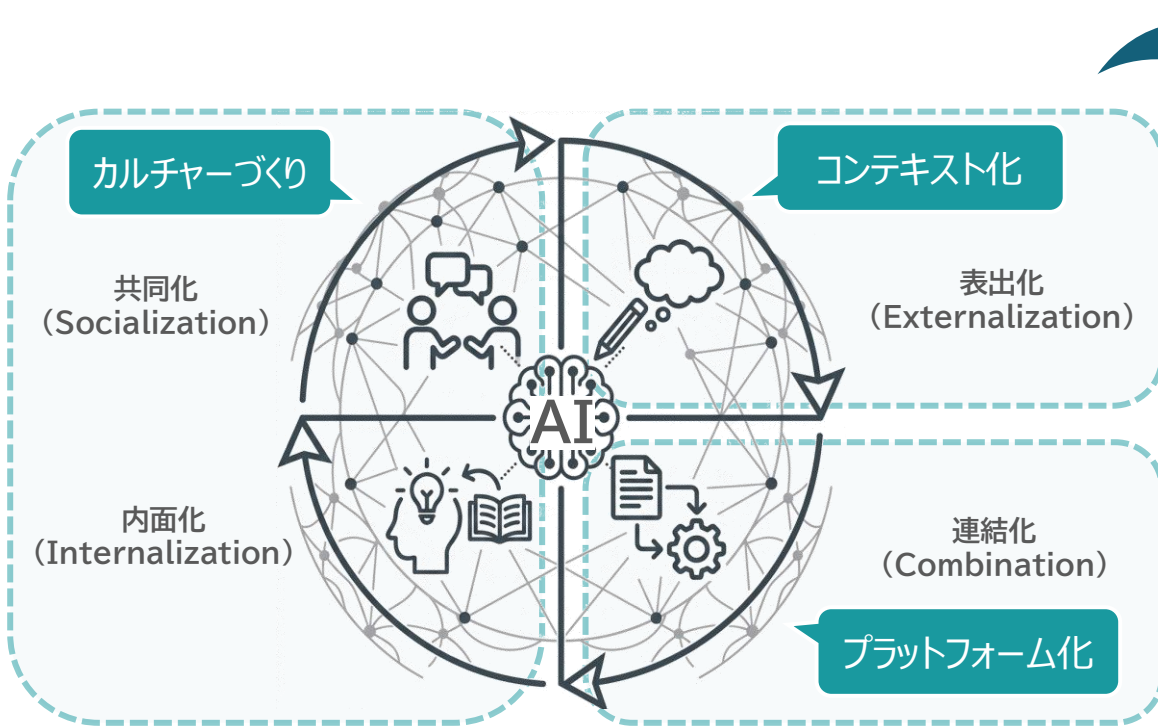
AIはSECIのサイクルを加速させる

AIのアシストを得てSECIのサイクルを高速に回し続け、運用品質の改善サイクルを速める



SECIの詰まりに対する3つの打ち手

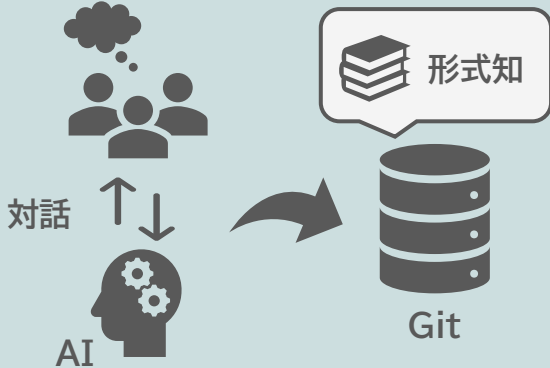
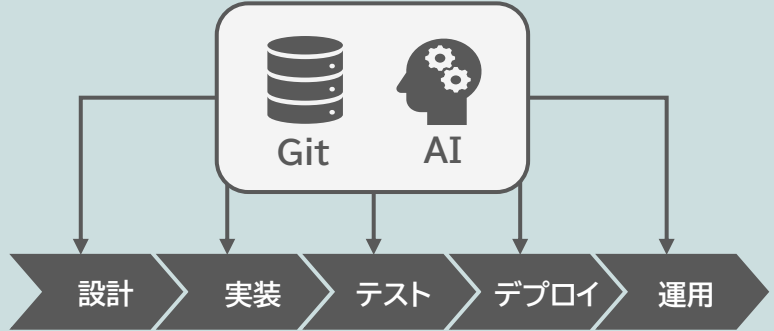
SRE・運用でAIをSECIのサイクルに組み込み、高速かつ持続的にまわすための打ち手



詰まり	SECIの課題	打ち手
学習・育成不足	共同化・内面化の機会が足りない	カルチャーづくり コンテキスト化
属人化	暗黙知が個人に滞留する	コンテキスト化
Toil	形式知が実行プロセスに入らない	プラットフォーム化
アラート疲れ	運用知見が再利用されない	コンテキスト化 プラットフォーム化

SECIを加速するためのAI活用事例

カルチャーづくりからしくみ化までSECIにAIを組み込むための事例を紹介

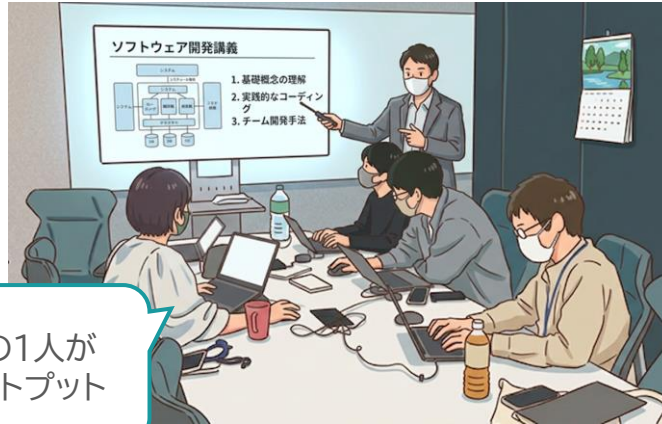
カルチャーづくり	コンテキスト化	プラットフォーム化
「共同化」(S)「内面化」(I)	「表出化」(E)	「連結化」(C)
学び・アウトプットを日常化する 1分アウトプット勉強会 - レクチャーする機会づくり	チームの経験やノウハウを AIが参照・再利用できる知識にする チームの経験やコツ、ノウハウをコンテキストとして一元管理する	形式知を実行プロセスに入れる チームの形式知が自然と組み込まれる、AI-Readyな開発運用プラットフォームづくり
		

SECIを加速するためのAI活用事例

取り組み事例 — カルチャーづくり

- ① デイリースタンドアップに学び・アウトプットの機会を組み込む
- ② メンバーが講師となってチーム内外にレクチャーする機会を提供する

1分アウトプット勉強会



毎朝1分でチームの1人が
学び・気づきをアウトプット

取り組み内容

- ・ 毎朝のミーティングで1人1分1スライドで説明
- ・ 内容は本業や技術に縛られず、趣味やプライベートでの学びもOK
- ・ クオリティよりも続けることを優先する

レクチャーする機会づくり



敢えて若手を
勉強会の講師に！

KubernetesとOpenShiftの違い

OpenShiftがKubernetesに加えて提供する独自機能や強化ポイントを紹介

■ 完成車 (OpenShift)
= エンジン (Kubernetes) +
RedHat拡張 (自動車の部品) + エンタープライズ機能 (安心サポート)



取り組み内容

- ・ チーム外や他部門、新入社員向けの勉強会を開催し、2～3年目の若手が講師として登壇
- ・ 説明スライド、デモ・ハンズオン環境の作成を通して基礎知識を足場固め

取り組み事例 — カルチャーづくり

アウトプットドリブンな学習を、ストレスフリーな形でチームの日常にする



12/18(木) 食中毒怖い…

美味しいものでもリスクあり、飲み会などでは要注意を…

■ ウイルス性胃腸炎になったため、食中毒についてまとめてみました…

- ノロウイルス (今回は恐らくこれ)
 - ・ 患者数が圧倒的に多い
 - ・ 牡蠣による原因が有名
 - ・ 集団感染のリスクが高い (今回、一緒に生牡蠣食べた人はピンピンしてます)
 - ウエルシュ菌
 - ・ カレーなどの大鍋料理の作り置きが原因に
 - ・ 通常の加熱調理では死滅しない
 - カンピロバクター (年始になりました)
 - ・ 鶏レバーなどの加熱不足の料理が原因に
 - ・ しっかり加熱することが大事
- 食べ物には気を付けてください…

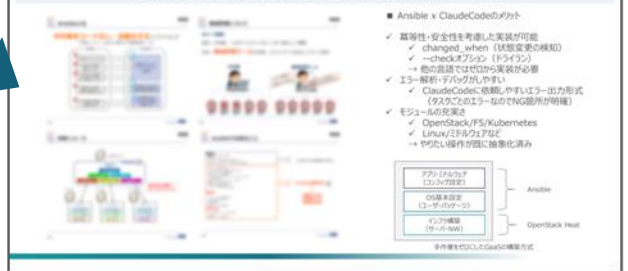


プライベート・
趣味のはなし

業務・技術の
はなし

4/6(月) ClaudeCode × Ansible

ClaudeCodeでAnsibleプレイブックを作成してみた



技術発信するメンバーが一部に偏りがちだった
→ 内容は問わず敷居を低く、チーム全員に
アウトプットする機会をつくる

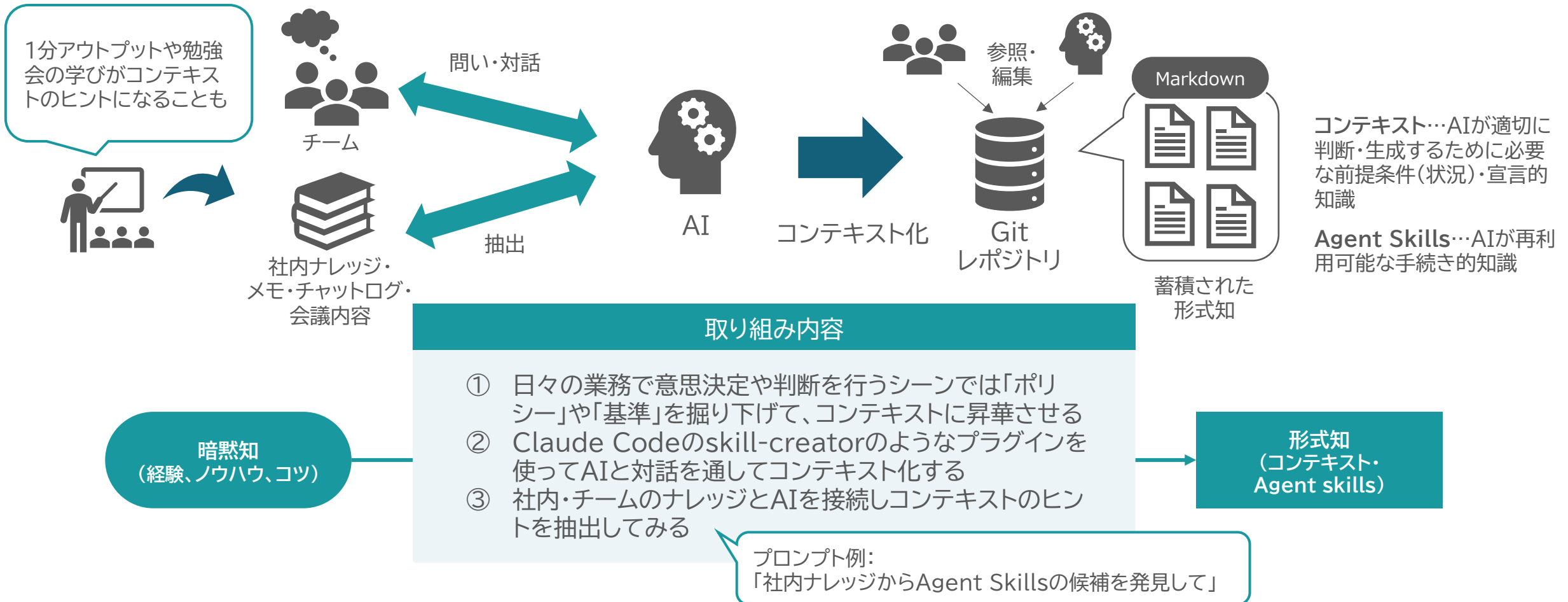
技術のアウトプットも徐々に増加

得られた効果・知見

- ・ 以前より各メンバーに閉じがちなノウハウ(暗黙知)がチームに共有されやすくなり、次のコンテキスト化にもつなげやすくなった
- ・ レクチャーすることでメンバーのファンダメンタルスキル強化、アウトプットによるチーム内外とのコラボレーション促進
- ・ 表出化(E)・連結化(C)で作ったアウトプットも、腹落ちのカルチャーがなければチームに根付かない
— 共同化(S)・内面化(I)もAIのサポートを得て定期的に回し続けることが重要

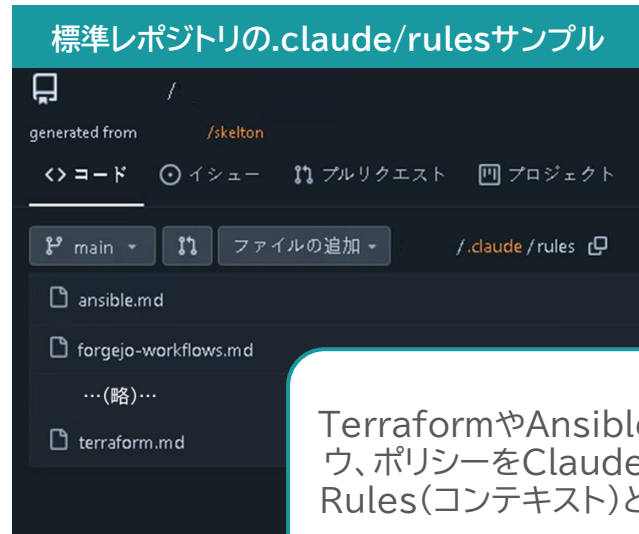
取り組み事例 — コンテキスト化

チームの経験やノウハウといった暗黙知をAIが読める「コンテキスト」として一元管理する



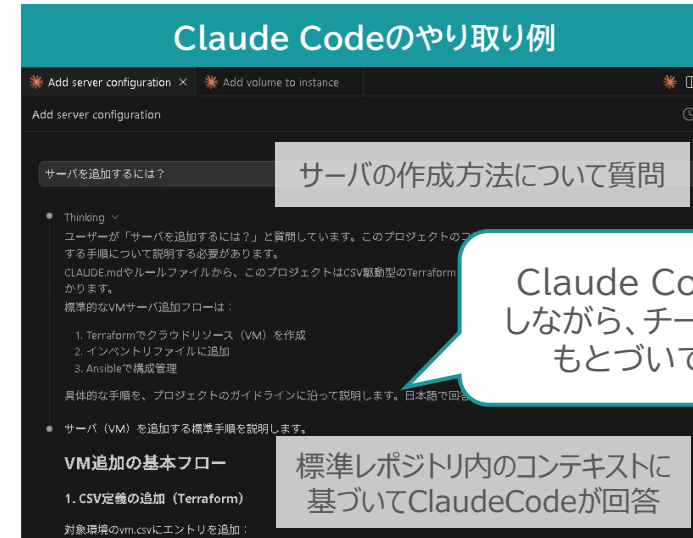
取り組み事例 — コンテキスト化

チームやAIがメンテできるような形式知としてコンテキストをGitレポジトリで一元管理



TerraformやAnsibleのノウハウ、ポリシーをClaudeCodeのRules(コンテキスト)として定義

```
1 # Terraform適用ルール
2
3 このプロジェクトのTerraform適用における必須ルールと推奨事項。
4
5 # アーキテクチャ原則
6
7 # CSV駆動型アーキテクチャの遵守
8
9 **必須**: このプロジェクトは**CSV駆動型**インフラ定義を採用している。
10
11 - すべてのリソース定義は 'code/terraform/environments/(env)/config/*.csv' で管理
12 - Terraformコード (.tf ファイル) には直接リソースパラメータをハードコードしない
13 - 新しいリソースタイプを追加する場合も、CSV + モジュール構造を維持する
14
```



Claude Codeと壁打ちをしながら、チームのノウハウにもとづいてIaCを構築

得られた効果・知見

- IaCを使った構築案件で、新人・若手3名のオンボーディングを1か月で立ち上げに成功
- インシデント初動の迅速化やチームメンバーの対応レベル底上げ(年間 約100時間の稼働削減)
- **AIとの対話や積み上がったチームのチャットログやメモ、Wikiを最大限に活用しコンテキストのプロトタイプを作成し育てる**

取り組み事例 — コンテキスト化

Before

インシデントに関するチャットログや会議が散在していて、探すのに一苦労

あの件どうしたっけ？

あのアラートはベテランの先輩に聞かないと分からない



After

skillsで過去のインシデントから類似のものを横断的に検索、初動をより迅速に！

過去の対処策を発見！

コンテキストを通してベテランと同等のノウハウを活用可能



インシデント調査skills

```
---
name: incident-search
description: システムアラーム発生時や障害調査時に、リポジトリ内の過去のインシデント記録から類似事例を検索して対処方法を提示します。「過去のインシデントを調べて」「似たような障害はないか」「このエラーの対処法を探して」などのフレーズで起動します。
---
```

```
# Incident Search Skill
## 検索対象
- `incidents/` ディレクトリ
```

```
## 類似インシデントの基準
類似インシデントとは、以下のいずれかに該当するものとみなしてください。
- 同種または類似のシステム／サービスに関するインシデント
- 原因や発生要因が似ているインシデント
- 影響範囲やユーザー影響が似ているインシデント
```

① 類似の基準

```
## ワークフロー
```

```
### SRE観点の分析
#### 再発パターン
- 同じ原因で複数回発生していないか
- 再発防止策は実施されたか
#### 根本原因 vs 表面的原因
- エラーメッセージだけでなく、なぜ起きたか
- プロセスの問題、設計の課題有無
#### 一時対応 vs 恒久対策
- その場しのぎの対応だったか
- 根本解決まで行ったか
```

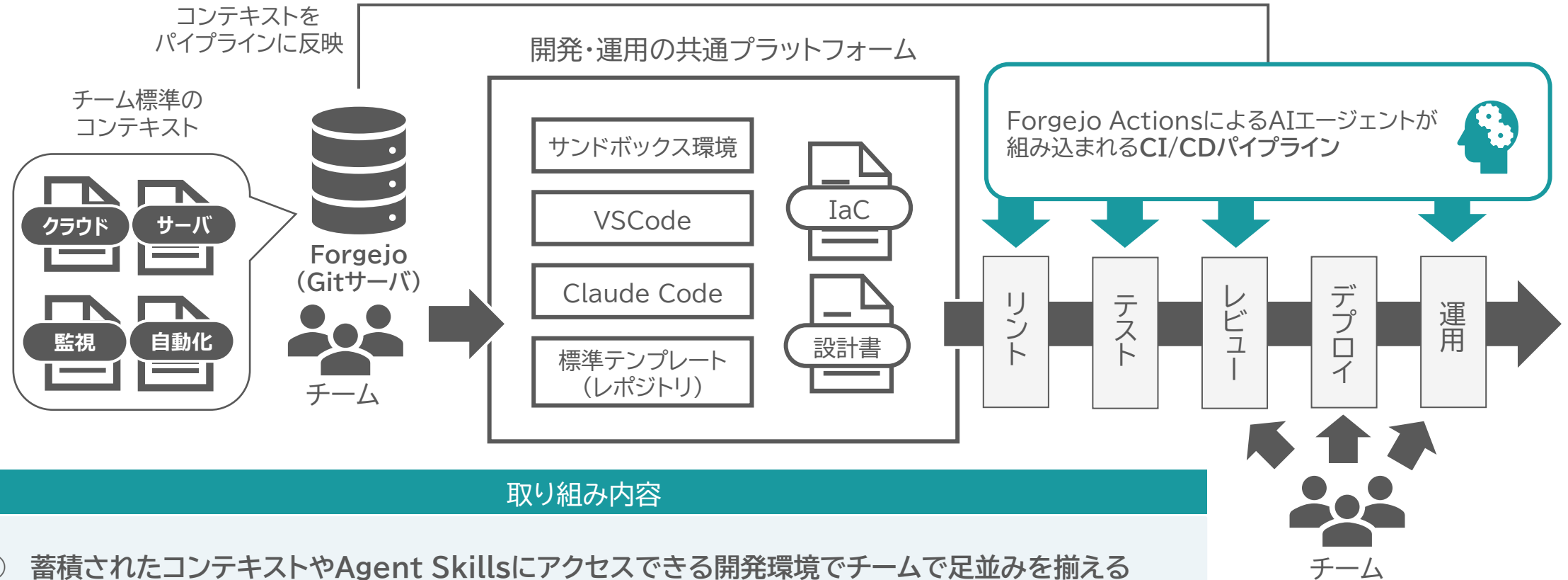
② 新人や若手が見落としがちなチェック観点

```
## 出力フォーマット
見つかった類似インシデントについて、**日本語で**次の項目を**表形式**でまとめてください。
- 類似インシデントのタイトルまたは簡単な名称
- 類似インシデントの概要(3行以内)
```

③ 活用しやすい出力形式

取り組み事例 — プラットフォーム化

蓄積された形式知を開発運用の実行プロセスに組み込んだプラットフォームをつくる

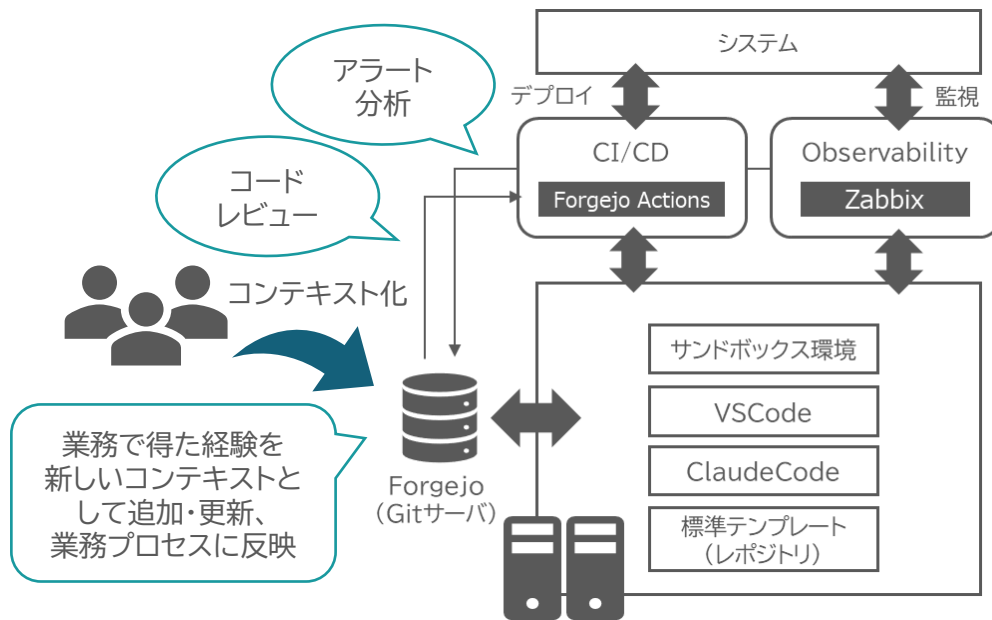


取り組み内容

- ① 蓄積されたコンテキストやAgent Skillsにアクセスできる開発環境でチームで足並みを揃える
- ② コンテキストが意識せずとも活用できるようにAIエージェントを開発運用のパイプラインに組み込む

取り組み事例 — プラットフォーム化

開発と運用に欠かせない、サンドボックス環境、コーディングエージェント、CI/CDパイプライン、統合監視、レポジトリ管理を1つのプラットフォームにまとめて提供



得られた効果・知見

- コンテキストを押さえたエージェントのパイプライン組込みにより、コード作成やレビューチェックなどの工程でエージェントを活用し、デリバリー品質や変更のリードタイムを改善(例:インフラ変更のリードを数日→半日～一日に短縮)
- 蓄積されたコンテキストを開発運用のプロセスに組み込むことで、Toilを削減して現場をラクにして、次の循環を生み出す

まとめ

ふりかえり — チームはどう変わったか

4つの詰まりに対して、SECI × AI の3つの打ち手でどこまで変えられたか

観点	Before	After
学習・育成の不足	新人育成に半年～1年 育成に手が回らない	約1か月程度でIaC構築案件のオンボーディング可能
属人化	「〇〇さんに聞かないと 分からない」 ノウハウが個人に滞留	コンテキスト化で ベテラン同等の知見にチーム全員がアクセス可能
Toil	繰り返し作業、変更対応に 数日単位で稼働消費	プラットフォーム化で インフラ変更のリードタイムを短縮
アラート疲れ	調査のたびに 数時間かかってしまう	インシデント調査Skillsで 初動迅速化

やってみた気づき

- ① 形式知だけでは、チームは動かない
共同化(S)/内面化(I) の場で腹落ちしてはじめて
形式知は「生きた知識」になる
- ② 完璧より、小さく回す
完璧なコンテキストに書き上げようとする挫折する
プロトタイプをつくって試しながらブラッシュアップする
- ③ 「役に立った」の聲が、次の循環を生む
形式知が現場で役に立ち感謝が返ってくことで、メンバーのモチベーションが上がり、SECIのサイクルが自走しはじめた



SRE・運用を変えるカギは、AIそのものではなく、 AIを組み込んだチームの知識循環、成長のデザインにある

カルチャーづくり

チームは今週何を学んだか？



コンテキスト化

その学びは、
AIが読める形で
残っているか？



プラットフォーム化

その形式知は、
明日の業務に自動で
組込まれるか？



3つの問いに対してYESが揃ったとき、SECIが回りはじめる
AI時代のSRE・運用チームの変革は、そこからはじまる

「つなぐチカラ」を進化させ、
誰もが思いを実現できる社会をつくる。

KDDI VISION 2030

