
AIに運用を任せきれない現実と、 それでも任せるための工夫

高橋 和真

クラウドエース株式会社



自己紹介



高橋 和真 (Takahashi Kazuma)

クラウドエース株式会社 DevSecOps 事業部 SoftwareEngineer

Work : AI x 運用保守システムの基盤開発、運用を担当

Tech : SRE O11y Serverless

Award : PTE 2025 / 2026 Rookie of the Year, Google Cloud All Cert 25

SNS : X, @kazzuvi Github, @kazuvi

今日の結論：運用者の本業は「コンテキストマネジメント」になる

AIの性能そのものより、「何を渡せるか」が運用品質を決める。

- AIは運用作業を代替できる場面が増えている
- ただし運用は、前提情報が欠けると判断を誤る
- 重要なのはAIの性能ではなく、何を渡せるか

TODAY

- 1 なぜ任せきれないのか
- 2 どう任せなのか
- 3 実例で何が変わるのか

暗黙知と、無制限に増える文脈

Skill / Hook と Work AI

インシデント対応の現場から

開発と運用では、AIに必要なコンテキストの性質が違う

	開発 (Development)	運用 (Operations)
時間軸	これから作るもの	すでに動いているシステム
主な入力	要件・ユースケース	制約・履歴・過去判断
AIの強みが効く点	探索・発想・コード生成	—（暗黙知に阻まれやすい）
効いてくる情報	ライブラリ知識・パターン	組織文脈・禁止事項・例外ルール
失敗のしかた	動かない／要件を満たさない	動くが触ってはいけない所を触る

→ 同じ「AIに任せる」でも、必要な情報の重心が違う。

任せきれない理由は「暗黙知」と「無制限に増える文脈」にある

VISIBLE / AIに渡せる情報

ドキュメント / コード / 手順書 / チケット

TACIT / AIに渡せていない情報

- 会議で決まったが文書化されていない取り決め
- 顧客ごとの触ってはいけない時間帯・例外ルール
- アラート時の連絡順、誰に先に当たるべきか
- 「あのリポジトリは触らない」など口頭ルール

文脈は本質的に有限化できない — だから「任せ方」の設計がいる。

CAUSAL CHAIN

判断に必須なのに、AIに渡せる形になっていない。

AIが知らない



AIが正しく判断できない



運用を任せきれない

「任せきれない前提」で任せる設計 — 2つの方針

01 PILLAR

コードエージェントに、調査可能な範囲を安全に与える

Agent	Skill	Hook
Claude Code	共有調査資産	危険操作の禁止

Skill で属人ノウハウを共有資産に。Hook で危険操作を構造的に止める。

02 PILLAR

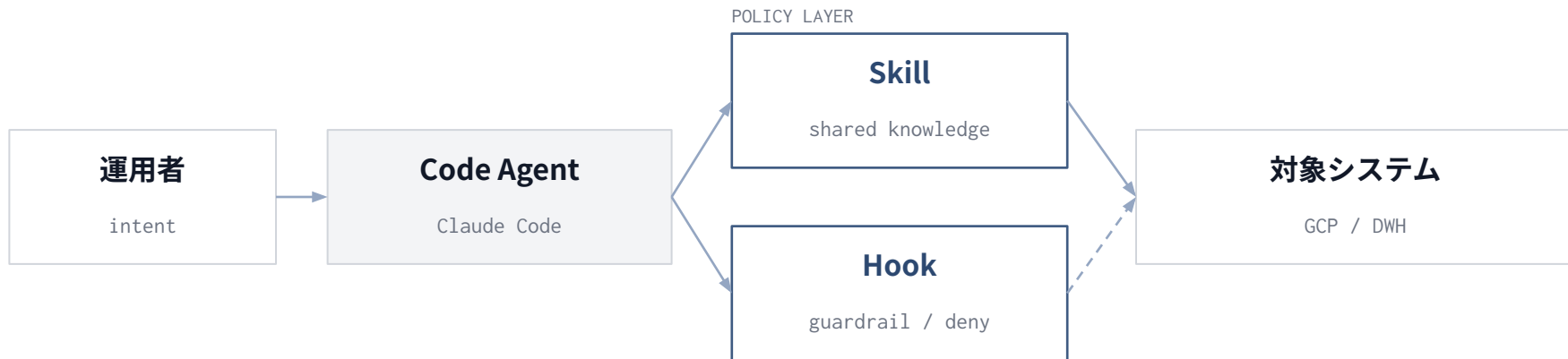
Work AI で、暗黙知を検索可能なコンテキストに変える

Gemini	Glean	MCP
Enterprise	横断検索	Agentから参照

議事録・録画・社内情報を横断参照し、暗黙知を検索可能なコンテキストへ寄せる。

→ AIの自由度を上げるのではなく、安全に任せられる条件を整える。

Skill と Hook で、調査をチーム資産として半自動化する



Skill 属人ノウハウをチーム全体へ展開する共有資産。誰がやっても同じ精度で調べられる状態をエージェント側に持たせる。

Hook AIに判断させて止めるのではなく、変更・削除を最初からできなくする。「判断ミス」をシステム側で排除する。

Work AI で、会議や議事録を「AIが読める文脈」に変える



→ 暗黙知の壁はゼロにはできない。だが「かなり低く」できる。

実例：ETL ジョブ障害 — AIが進め、人間が前提を集めた

事象 ETL ジョブでインシデント発生 / 原因特定～解消まで、コードエージェントを活用して実施

TIMELINE →



→ AIが作業を進め、人間が「なぜそのイベントが起きたか」という前提を回収した。

本当に価値があったのは「正しい解決策を選ぶための文脈」だった

SIGNAL

異常な形式のファイル
アップロードが起点と判明

PATH A / without context

× システム側を直す

利用側オペレーションと噛み合わず、再発リスクが残る。

PATH B / with context

◎ 利用側のオペレーションを修正

- データ連携を進めるメンバーの存在を知っていた
- DWH 関連リポジトリの所在を把握していた
- 「修正すべきはシステムではない」と判断できた

→ 決め手は周辺の運用文脈。

これからの運用者は「手を動かす人」から「文脈を整える人」へ

BEFORE

手を動かす人

- ログを確認する
- コマンドを実行する
- 手順書どおりに復旧する
- ノウハウは個人に蓄積する



AFTER

文脈を整える人

- 前提情報を収集する
- 決まり事を明文化する
- 情報源を更新・共有する
- AIが読める形に整える

→ AIのアウトプット品質は、渡したコンテキスト品質に強く依存する。

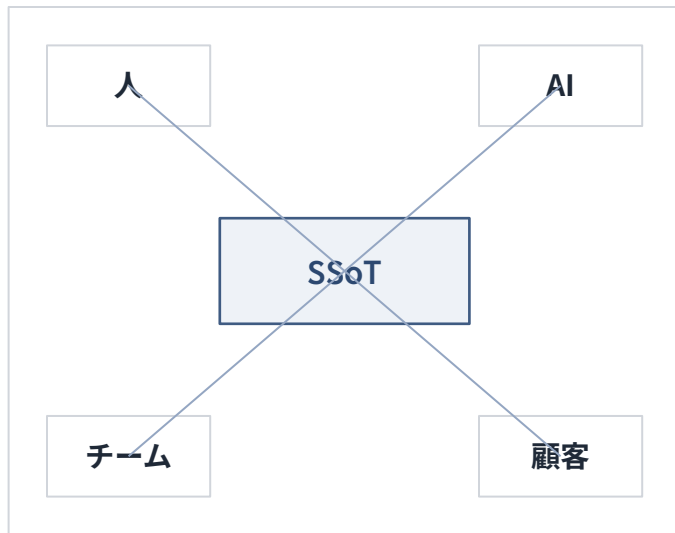
現場でまずやること：暗黙知の明文化と「単一ソース化」

CHECKLIST

- ☒ 会議で決まったことを残す
- ☒ 顧客ごとの例外ルールを検索可能にする
- ☒ 誰が・いつ・なぜ変更したかを追えるようにする
- ☒ AIとチームが同じ情報源を参照できる状態を保つ

→ 同じ情報源を参照することで、AIの判断と人の判断が揃う。

SINGLE SOURCE OF TRUTH



反発するより、任せきれないAIに任せる設計を進める

「全部任せられるか」ではなく、「どこまで安全に任せられるか」。

TAKEAWAYS

- 1 AIは今後も進化し、具体作業の多くを肩代わりしていく
- 2 問いは「どこまで安全に任せられるか」の設計
- 3 運用者の価値は、正しい前提を与え続けることへ移る

ご清聴ありがとうございました！