

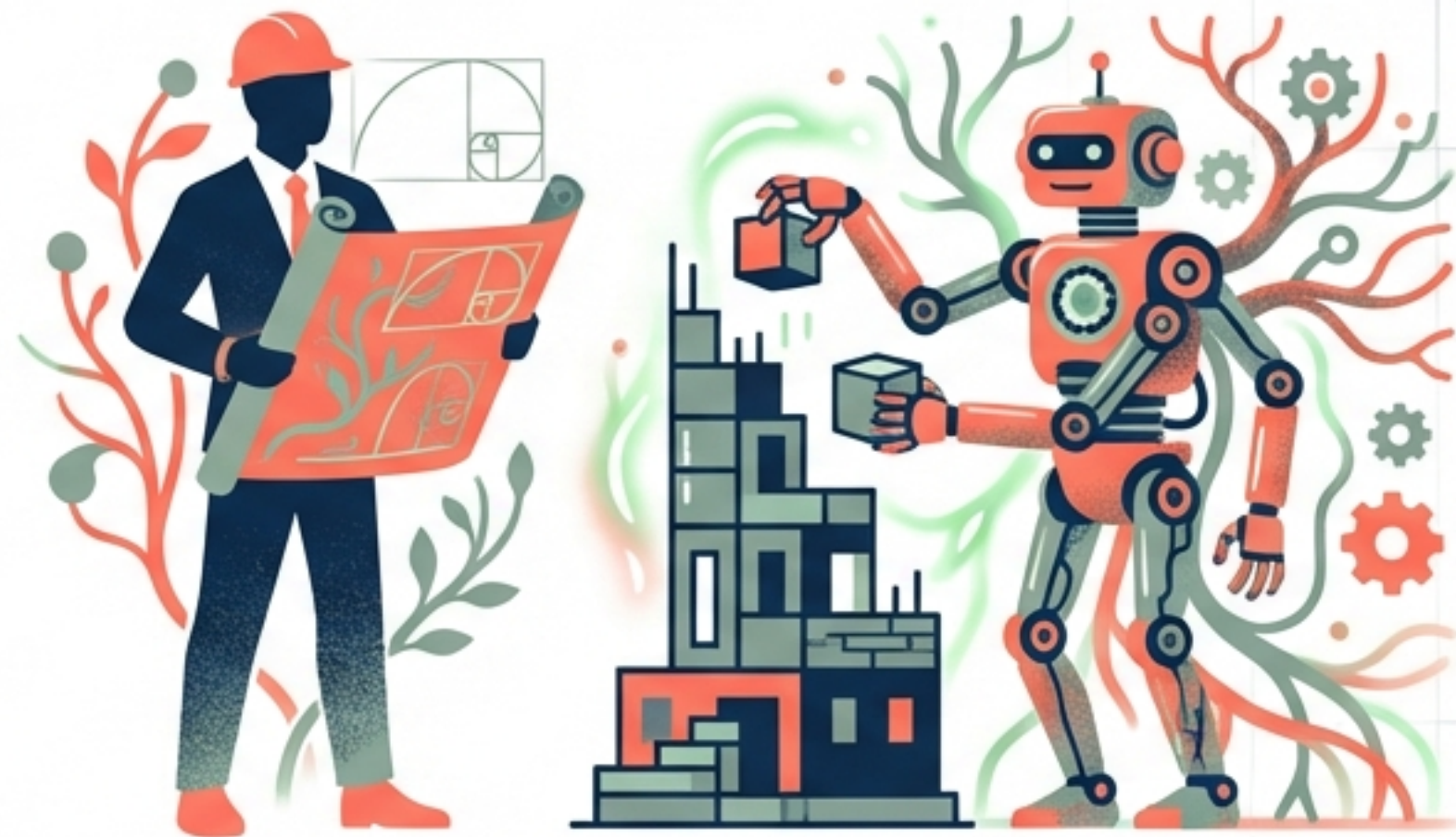
相転移：AI支援からAI駆動へ

2023-2025: AI-Assisted



- エフェメラルな文脈 (Ephemeral Context)
- 人間がドライバー

2026: AI-Driven



- 永田浩矢氏提唱「育てる開発」
- コンテキストの永続性 (Context Persistence)
- 自律的リファクタリング

「AI All-In」：南場智子会長のマンドート

経営目標

業務執行・活用

ツール・技術基盤

技術インフラ・組織

- AIシステム部 & AI戦略推進室 (Infrastructure)
- SWET (Quality Governance)
- 100 Use Cases (現場の活用事例)

人材の50%再配置
新規事業の創出

100のユースケース
ログ分析・運用監視

Claude Code / GitHub Copilot
内製IdP / MCP

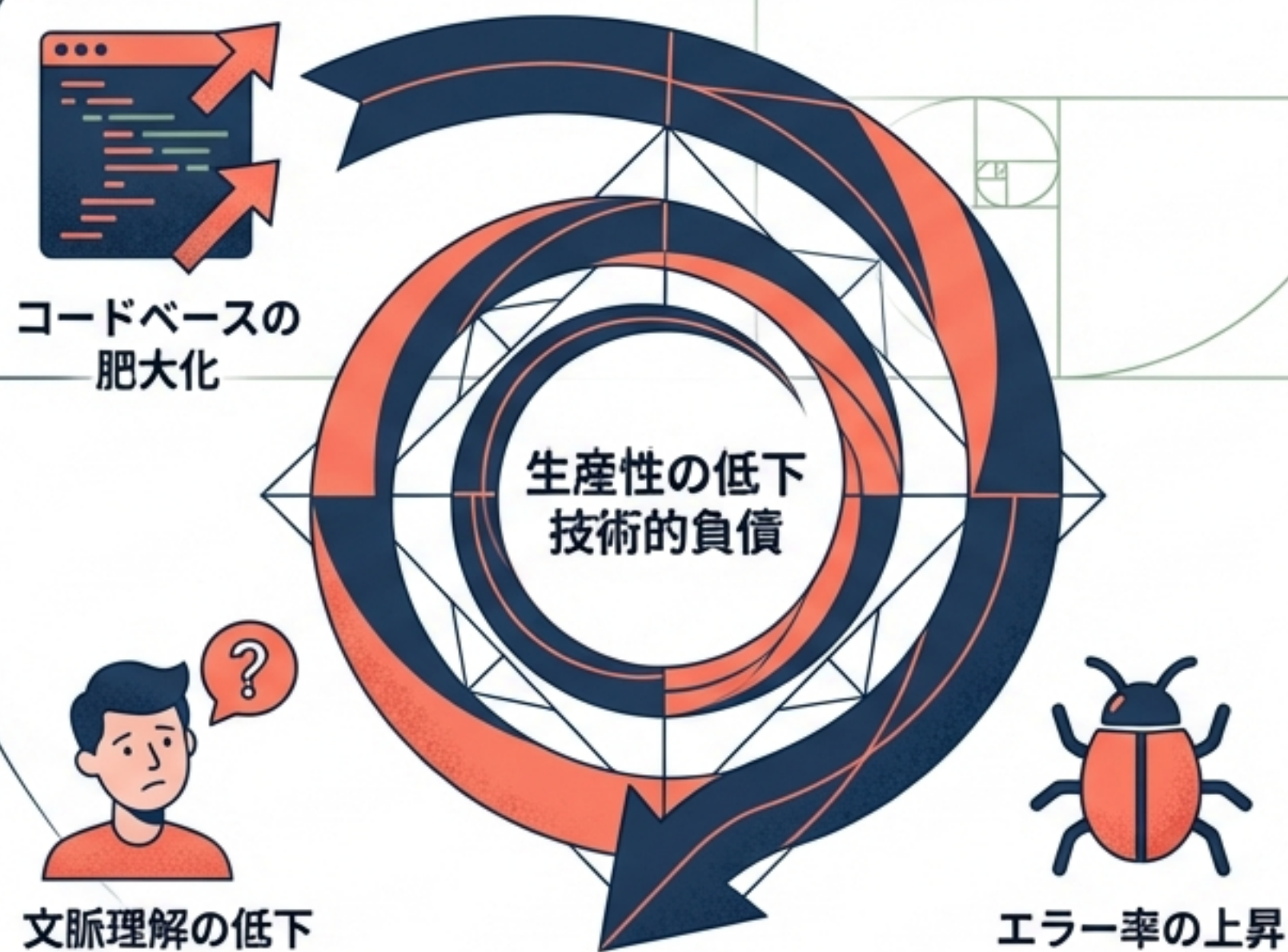
AIシステム部 / SWET
AIOps / Observability

50%

既存事業の人員をAIで削減し、
新規事業へ再配置

哲学：ジュニアエンジニアとしてのAIを「育てる」

Traditional (従来の開発)

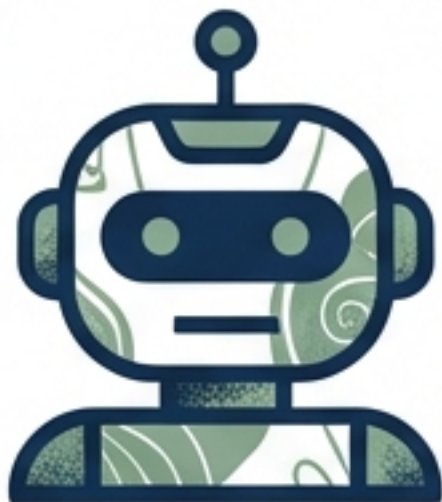


DeNA AI-Driven (逆エントロピー)



AI is not a tool, but a junior engineer learning the dialect.

汎用AIの限界と「コンテキスト・ギャップ」



Generic AI
汎用AI



Project Codebase
プロジェクトコードベース

The Problem 問題点

2024-2025年の「幻滅の谷」
Noto Serif JP

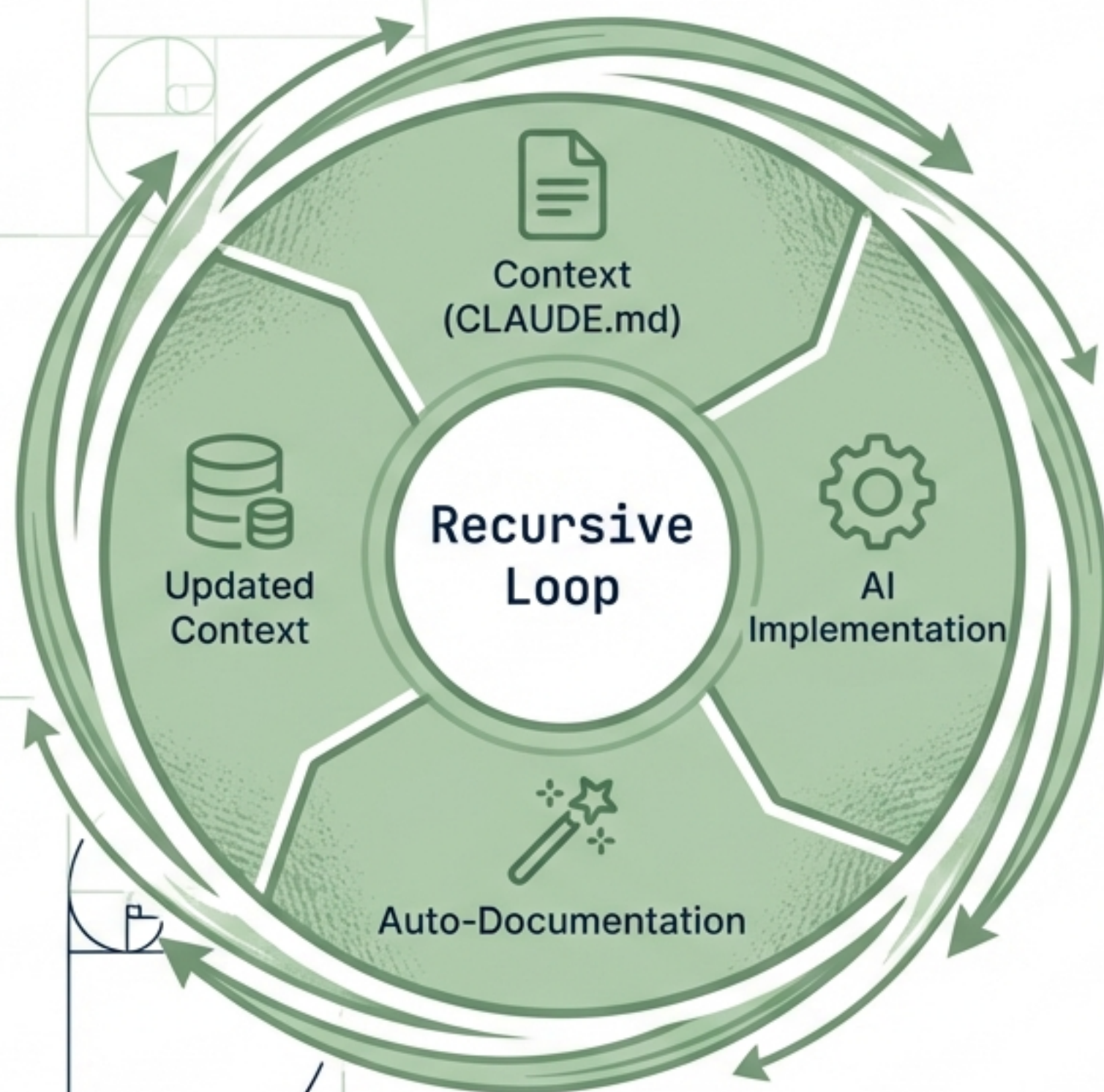
Reason 原因

プロジェクト固有の「暗黙知」の欠如
Noto Serif JP

Result 結果

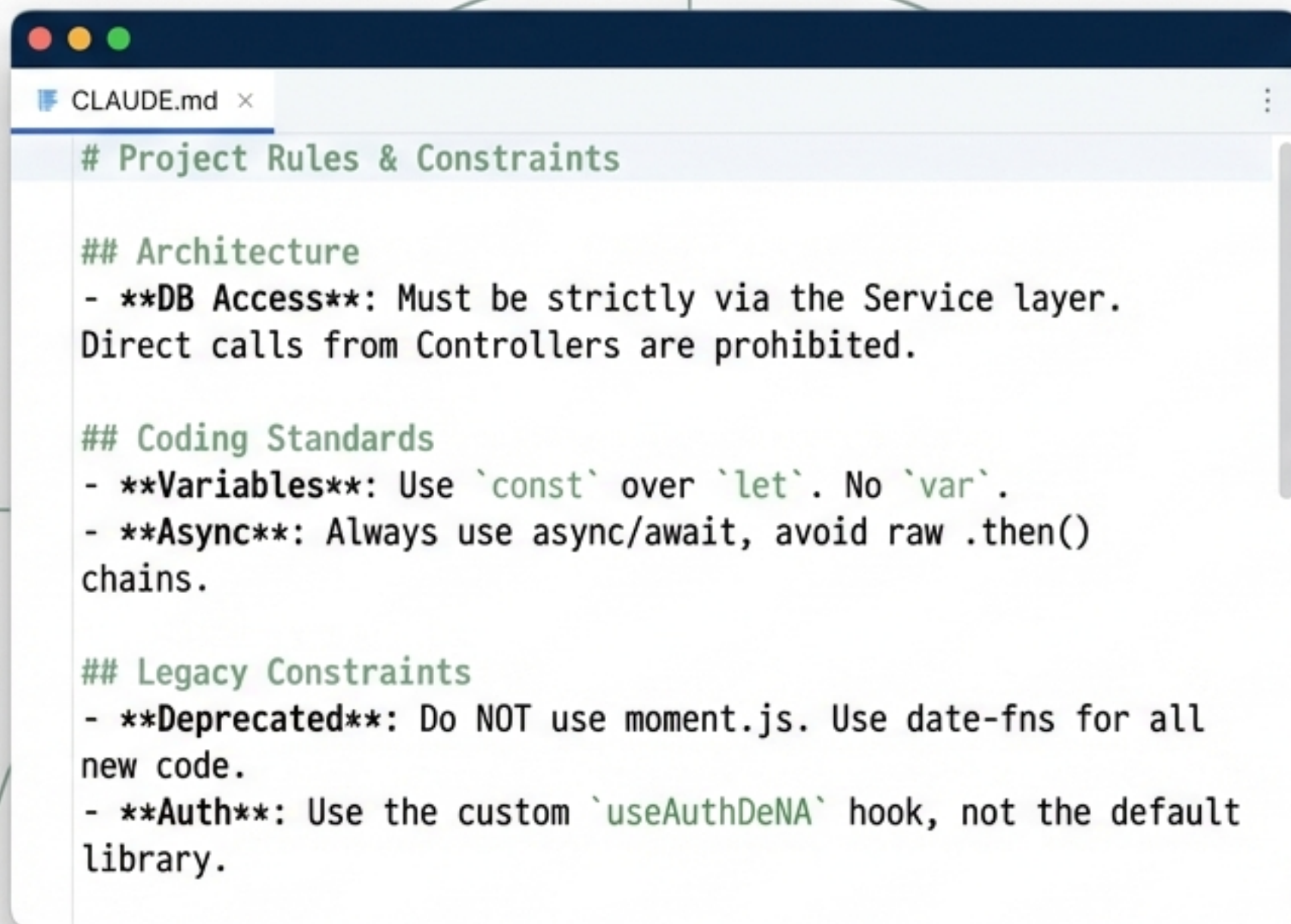
レビュアー疲労 (Reviewer Fatigue)
Noto Serif JP
 AIの修正に人間が時間を奪われる
Noto Serif JP

解法：再帰的コンテキスト注入



- **Cycle:** コンテキストの蓄積が、AIの推論精度を向上させる
- **Outcome:** 育てるほど楽になる
- **Mechanism:** プロジェクトの「掟」をシステムプロンプトとして永続化

システムの脳：CLAUDE.md / context.md



```
# Project Rules & Constraints

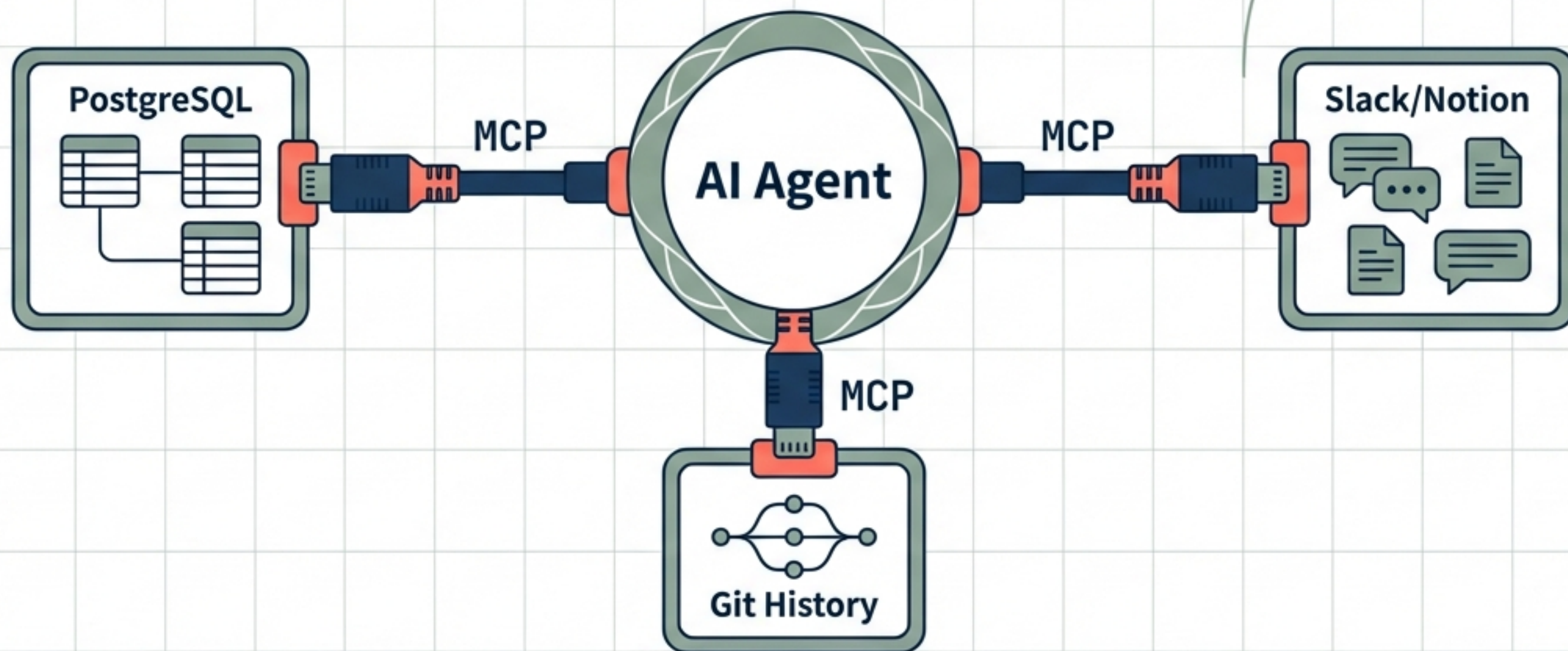
## Architecture
- DB Access: Must be strictly via the Service layer. Direct calls from Controllers are prohibited.

## Coding Standards
- Variables: Use const over let. No var.
- Async: Always use async/await, avoid raw .then() chains.

## Legacy Constraints
- Deprecated: Do NOT use moment.js. Use date-fns for all new code.
- Auth: Use the custom useAuthDeNA hook, not the default library.
```

- Role: 人間向けのドキュメントではなく、AI向けの「システムプロンプト」
- Effect: 繰り返しの指摘（Re-do）をゼロにする

神経系：Model Context Protocol (MCP)



Definition

AIのための USB-C
(The USB-C for AI)

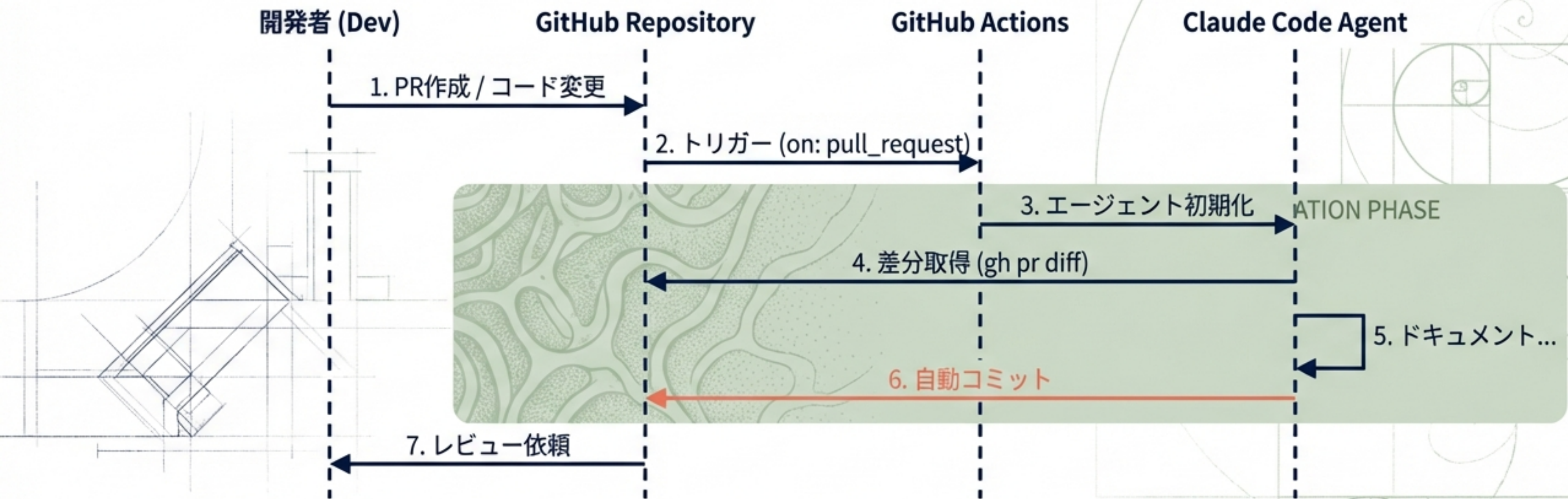
Connections

- PostgreSQL: スキーマ理解
- Git History: 変更の経緯 (Why) の理解
- Slack/Notion: 社内ナレッジの検索

Impact

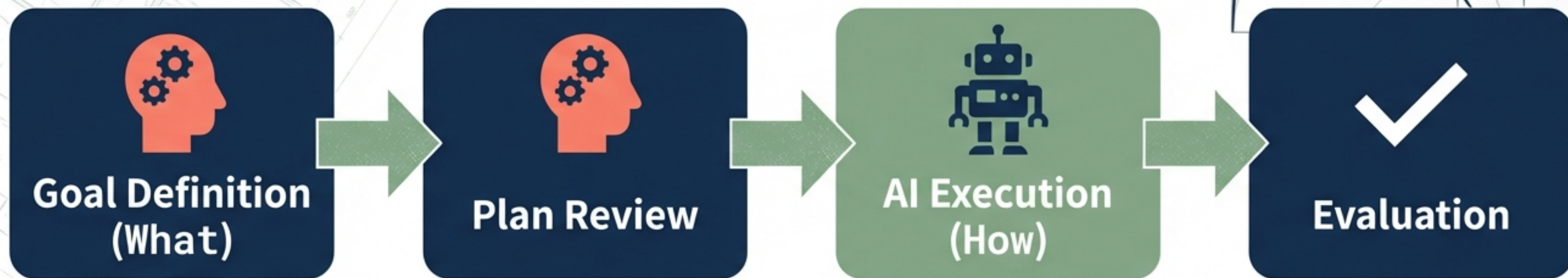
テキスト予測器から、状態を把握
するエンジニアへの進化

実践：自己修正型ドキュメンテーション



1. Trigger: `git diff` analysis via GitHub Actions
2. Detect: 影響範囲の特定 (e.g., auth.ts changed)
3. Action: AIがドキュメント修正案をPRに直接コミット
4. Result: コードとドキュメントの完全同期

新手法：評価駆動デリバリー（EDD）



Test Driven (TDD) → Evaluation Driven (EDD)

- Human Role: ドライバーから、プランレビューアへ
- Process: Human defines 'What', AI solves 'How'

ガバナンス：ループの中のSWET



SWET / Golden Datasets

Guardrails

セキュリティ脆弱性や
幻覚パッケージの自動阻止

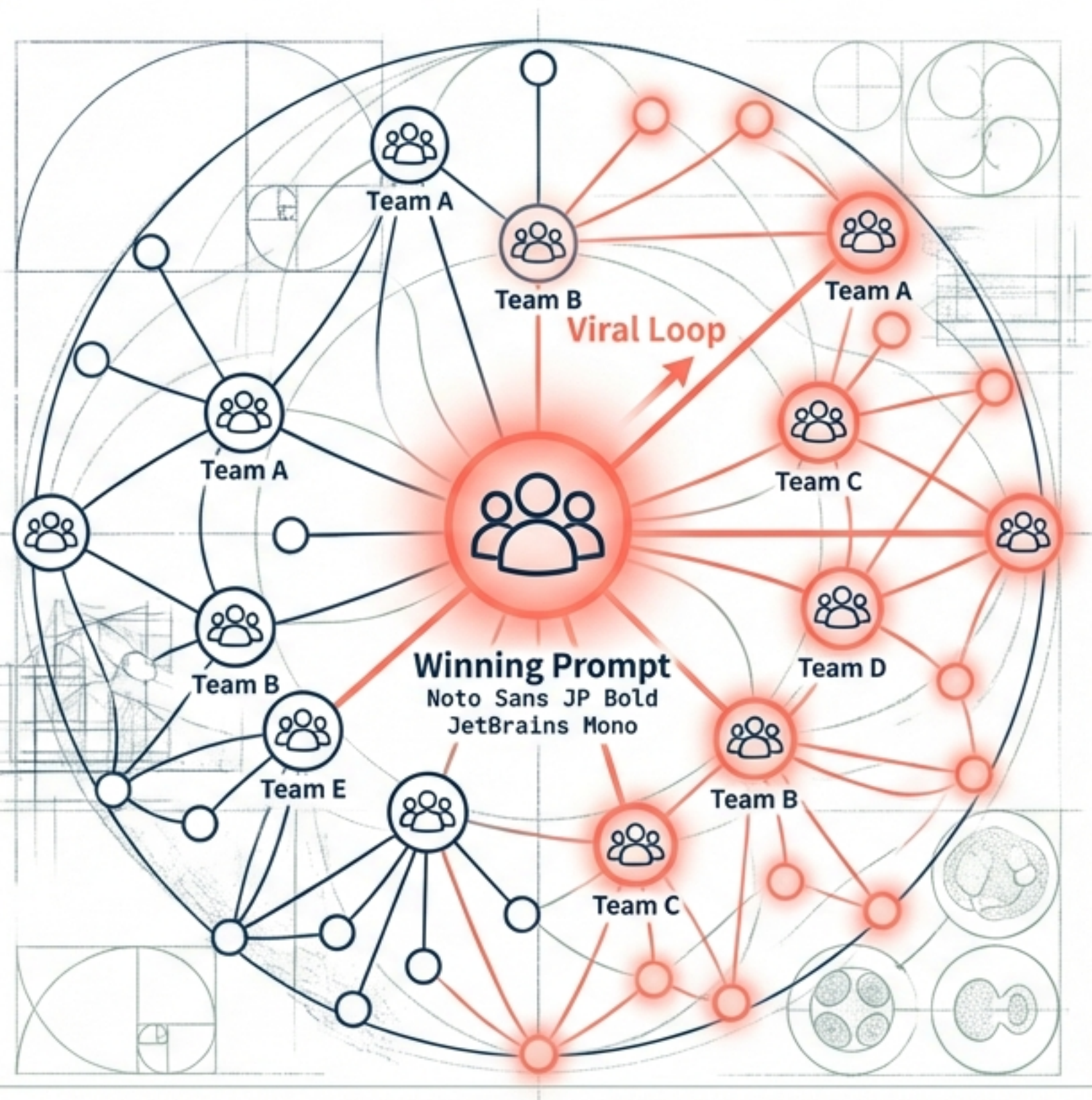
Golden Datasets

AIの正答率を測る
「正解データ」の管理

Metrics

AI生成コードの修正率、
採用率のモニタリング

集合知：「どやる時間」とソーシャルエンジニアリング



- Concept:
成功事例を自慢（どやる）し合う共有会
- Mechanism:
分散型遺伝的アルゴリズム
(Distributed Genetic Algorithm)
- Effect:
優れたプロンプトやMCP設定が全社に急速伝播し、AIリテラシーを底上げする

新しいエンジニア像：コンテキスト・アーキテクト

Noto Sans JP Bold

Old Competency



Syntax / Boilerplate

New Competency



Context Engineering /
Constraints

Junior Engineer : AIコードを批判的に読み解くレビュー能力

Senior Engineer : コンテキストの設計者 (Context Architect)

Mission : Proactive - AIが正しく動ける「庭」を整える

2030年へのロードマップ：エージェント分隊



Now (2026)

AI Driven / Recursive
Nurturing
JetBrains Mono



Future (2027-2030)

Agentic
Orchestration

Agent A: Architect



Agent B:
Coder



Agent C:
Tester

Requirement: 全社内ツールの「MCP準拠」 (USB-C化)



レンガ積み職人から、知性の設計者へ

- 
1. Shift: Use AI (使う) → Nurture AI (育てる)
 2. Value: コードそのものではなく、コンテキスト (文脈) に宿る
 3. Vision: 開発の摩擦が蒸発し、純粋な「意図」だけが残る世界へ