

# Claude Code /insights: AIがあなたの開発スタイルを「メタ認知」する

エージェントック・エンジニアリングにおける自己開発分析とワークフローの進化

> claude /insights



# 「ただ速く書く」から「賢く指揮する」へ

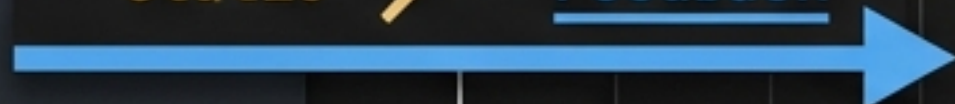
## Blind Speed (ただ速く書く)

```
return true;
def Muted Cyan = #56B6C2; name($te) {
Soft Purple = #C67800; name($te) {
this.starttime = settings;
this.to = $Uchete.ppt;
script.collectors;
sonse.path; 'sos23at';
); hructaethn; si eER;
stafloresn = seet;
this.sutterapor =
adnat seaveUser
new foadesd; fver;
if (inputStarttsiess(Decidont.oopenfsico) + stuzer'Dis & seer);
response.focdfinents('*@!') + $C9C8Y61);
back($aim 'backer', 'backer');
```



CLAUDE.md  
Static

Dynamic  
Feedback



## Meta-Engineering (賢く指揮する)



Problem: AIによるコーディング速度は劇的に向上したが、フィードバックループが欠如している。  
「Skill Atrophy (スキルの萎縮)」のリスクと、方向性のない「過剰な実装」が課題。

Solution: リリースノートに記載のない隠しコマンドとして登場した `/insights (v2.1.30)` は、AIに「鏡」を持たせ、開発者自身を分析させるメタ認知ツールである。

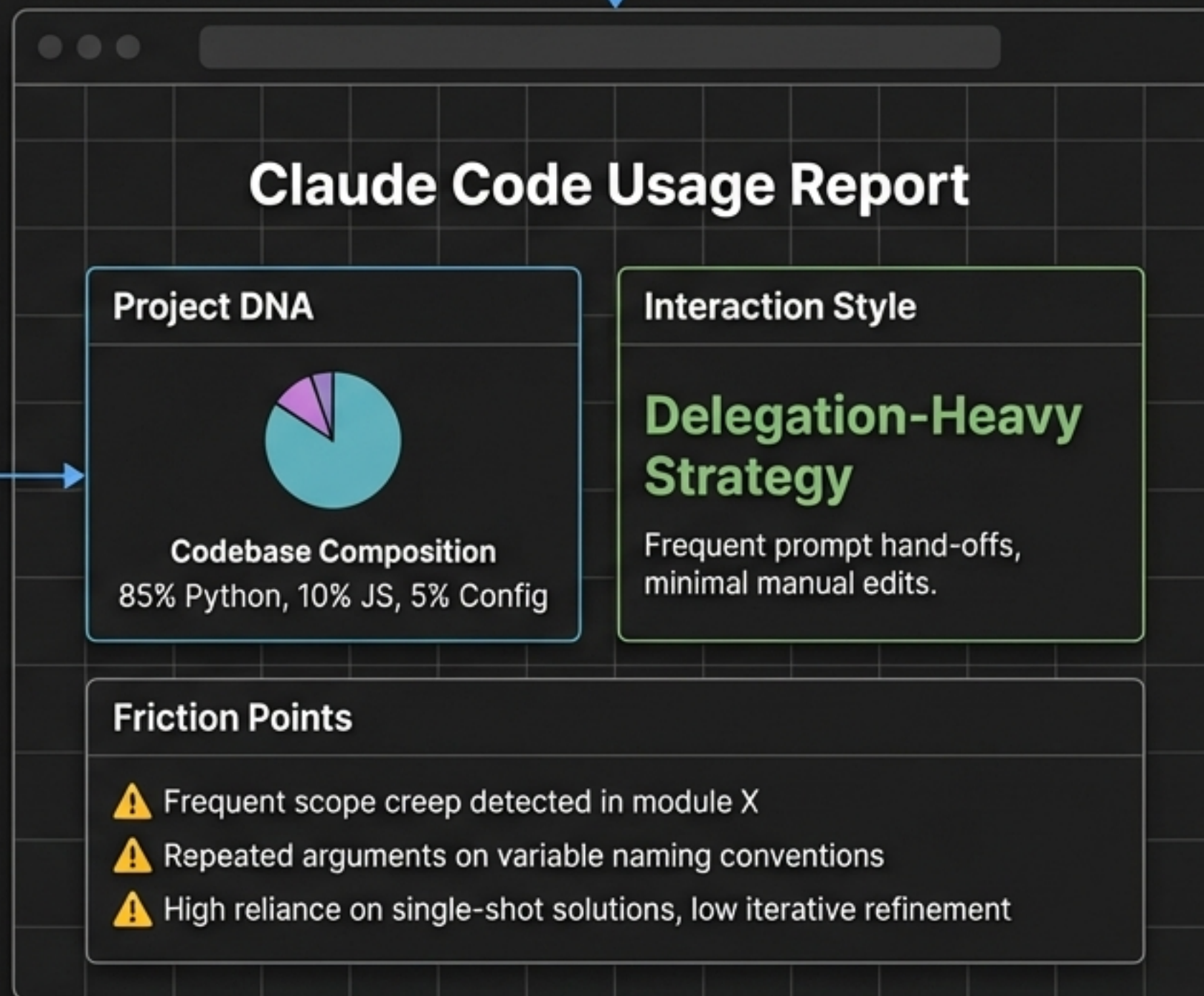
# /insights とは何か？：AIによる「開発者ロースト」

- 過去30日間のセッション履歴を読み込み、プロジェクトの要約、癖、摩擦点、改善案をまとめたHTMLレポートを自動生成する機能。
- 単なる統計ではない。「選択的な介入を伴う委任重視スタイル」といった客観的評価や、「どこでAIと喧嘩したか」という痛烈なフィードバック (**Roast**) が含まれる。

**30 Days History**  
HISTORY

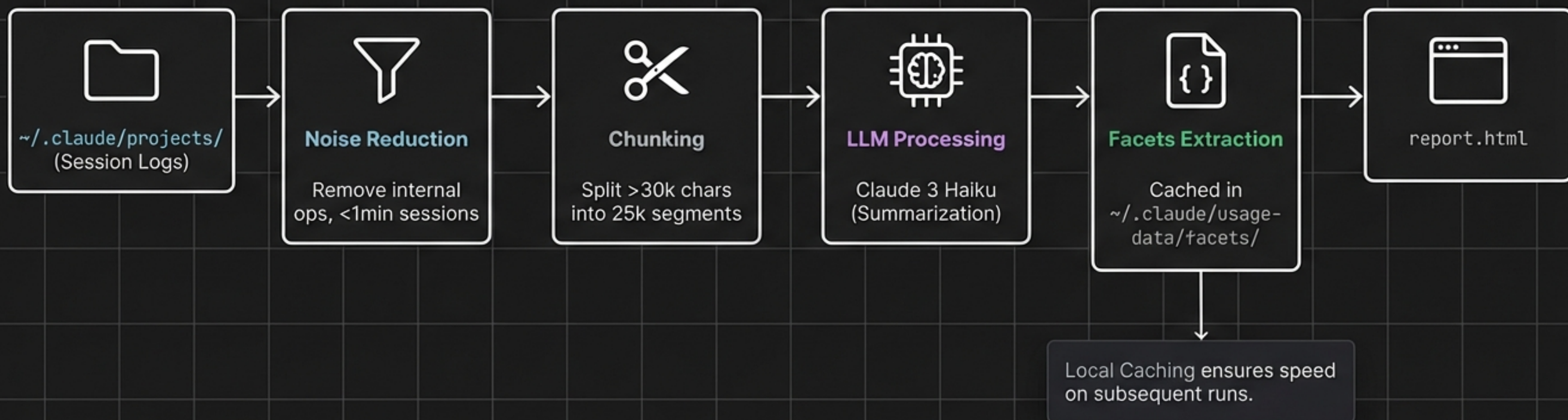
**~20,000 Characters**  
CHARACTERS

**100% Automated**  
AUTOMATED

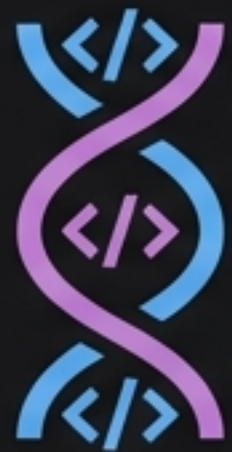


“自分のコーディング癖を暴かれる” — Community Reaction

# Architecture: データパイプラインと処理フロー



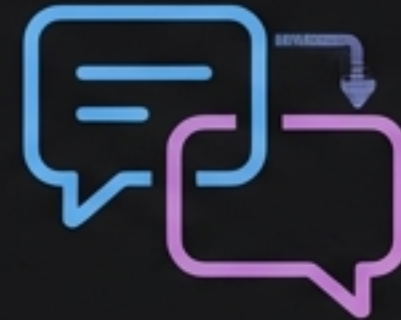
# Qualitative Analysis: AIが見た「あなたの姿」



## Project DNA

あなたは何を作っているのか？  
(What are you building?)

- 4-5 Key Areas Identified



## Interaction Style

Micromanager vs Delegator?

Analysis of iteration speed and instruction detail.

The Soul of the Report



## Friction Points

Where do you fight?

```
`misunderstood_request`  
`buggy_code`  
`wrong_approach`
```



## Success Patterns

High-efficiency workflows.

# Quantitative Metrics: 数字で見る生産性とリスク

Session Count

248

Lines of Code (LOC)

135,000+

Token Usage

8,500,000

Est. Cost (USD)

\$212.50

Helvetica Now Display

## CASE STUDY: The BlunderGOAT Incident

102,000 Lines of Code

in 68 Days

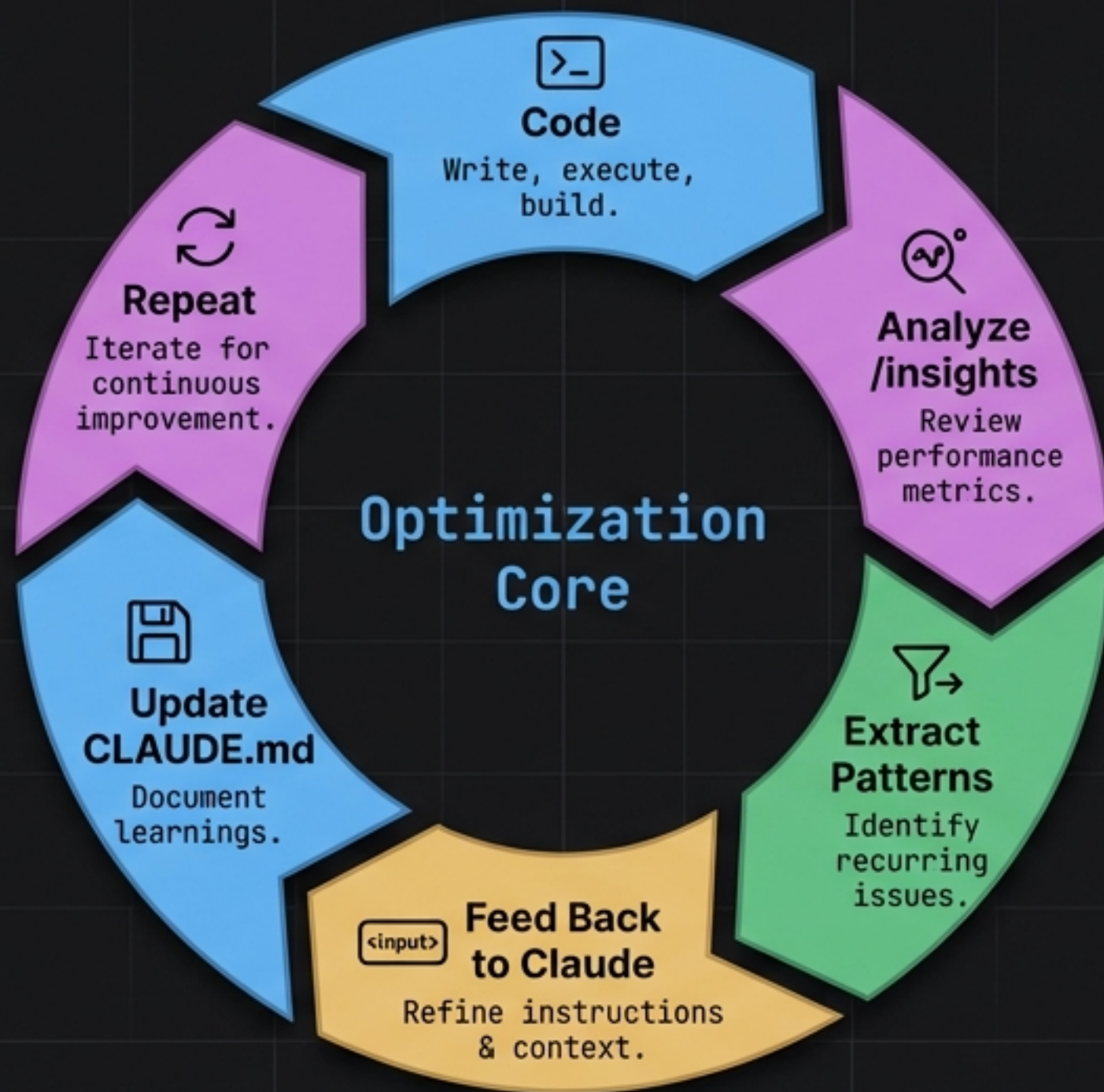
106 Hours / 28 Days

Analysis Text (Noto Sans JP Regular)

Insight: 圧倒的な生産量だが、レポートは「**バグ修正と機能追加への執着**」「**誤ったアプローチによる摩擦**」を指摘。

**! Lesson: 'Over-shibaking' (Overworking) without direction leads to massive output but high friction.**

# Insight to Action: 最適化のループを回す



"The report is not the destination; it is the seed for optimization."

# Action 1: CLAUDE.md の要塞化

## Problem/Method

PROBLEM:


繰り返される「誤解」と「手戻り」。

METHOD:

レポートの 'Friction Details' セクションをClaudeに渡す。

> このレポートの改善提案に基づき、  
CLAUDE.mdのルールを更新して

## Outcome

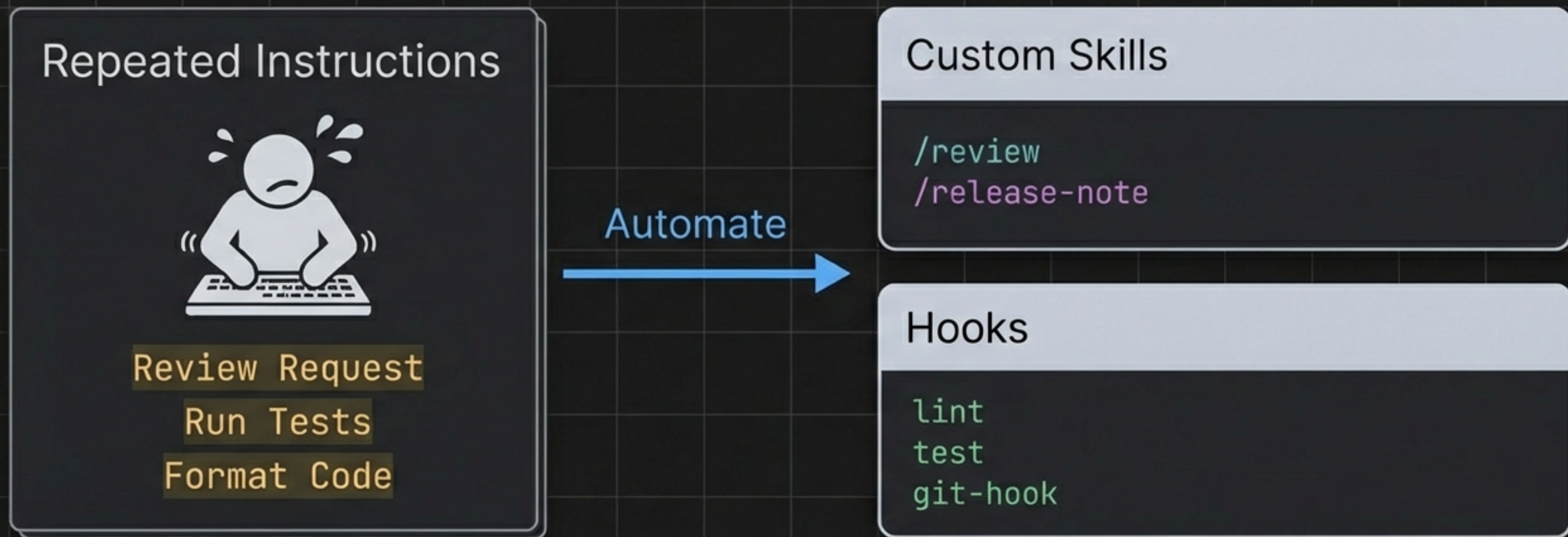
 CLAUDE.md



```
# Guidelines
- Read before fixing
- Check the whole stack
- Run preflight on every
  change
```

Guardrails against recurring errors.

# Action 2: ルーチンの「Skills」化と自動化

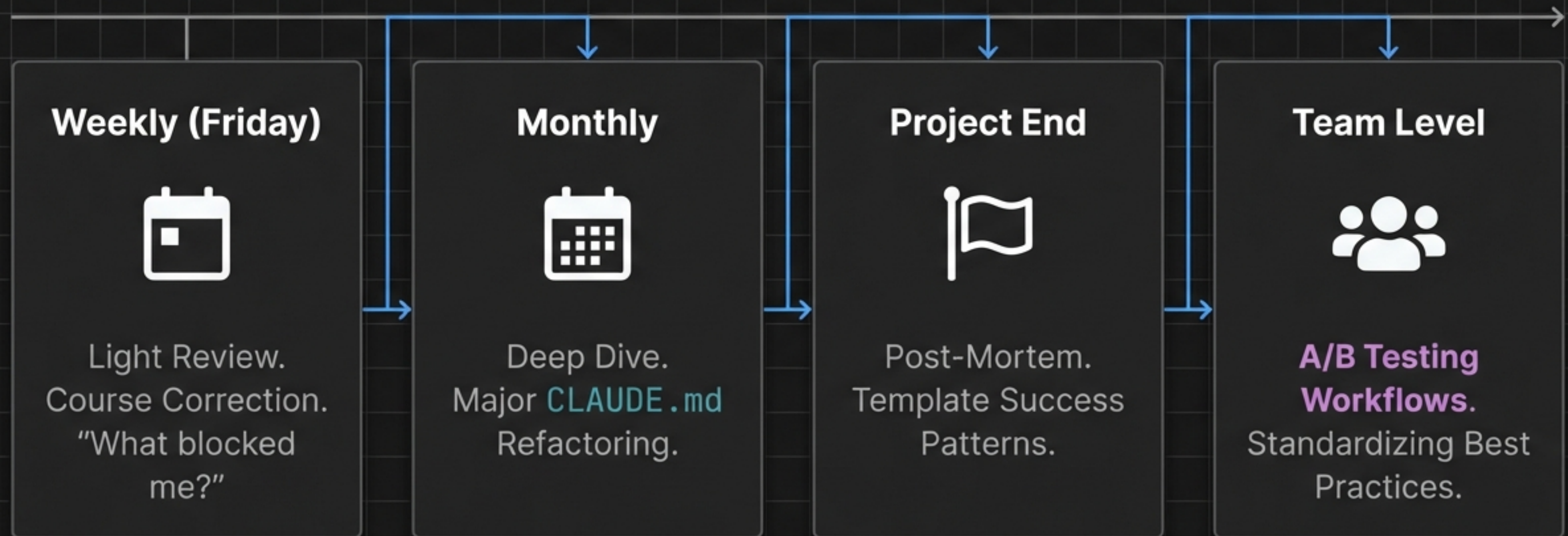


「2回指示したら、それは自動化の合図だ。」

# Action 3: コンテキスト腐敗を防ぐ 「記憶」 管理



# Strategic Use Cases: いつ実行すべきか？



# Cost & Requirements: コストと要件

Command Cost	Free (Token usage applies)
Pro Plan (\$20/mo)	Minimum Requirement ⚠️, Rate Limits Risk
Max Plan (\$100/mo)	Recommended for Heavy Users ✅

Expect **heavy token load** on first run (reads up to 1 month history).

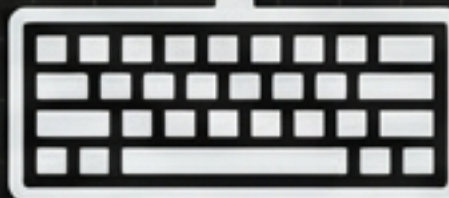
Data stays local (~/.claude/).

# The New Developer Identity: コーダーから「指揮者」へ



Source: Anthropic Internal Data

```
1 <code  
2   function() {  
3     <senat_code />  
4     return  
5   }
```



A white keyboard icon with a blue arrow pointing from the code block above it.

Old: Typing  
Syntax



New: Planning,  
Reviewing, Steering

You are the manager of an **infinite** intern.  
**/insights** is your performance review tool.

# The 'Pro' Checklist: 今すぐやるべき4ステップ

- ✓ **Run:** `> claude /insights` を実行する (Pro+ Recommended).
- ✓ **Summarize:** Ask Claude to summarize 'Improvement Proposals'.
- ✓ **Update:** Apply changes to `CLAUDE.md` immediately.
- ✓ **Create:** Create 1 new Custom Skill based on data.

**Data without action is vanity. (行動なきデータは虚栄である)**

AI analyzes your code.

The `/insights` command analyzes **you**.



```
> claude /insights
```