

# AI Native人材への第一歩

# Agent Skills

---

小山奏汰

2026.02.19

# 自己紹介

---

- ✓ 小山奏汰（こやまかなた） GitHub koka0630
- ✓ 29歳 Sier企業 エンジニア
- ✓ 言語: Python, Typescript
- ✓ 得意・興味: GCP/Azure, AI
- ✓ 経歴
  - ✓ 新卒から5年間スタートアップにてエンジニア
    - ✓ IOSアプリにて画像生成AIの学習
    - ✓ B2B SaaSにてインフラ構築, SRE, 採用など
  - ✓ 2025/12からSier企業にてAIエージェント構築



# ターゲット・目的

---

## ターゲット

- ✓ Claude Codeなどコーディングツールを業務で活用されている方
- ✓ 特に、Agent Skillsを使い始めた方・興味がある方

## 目的

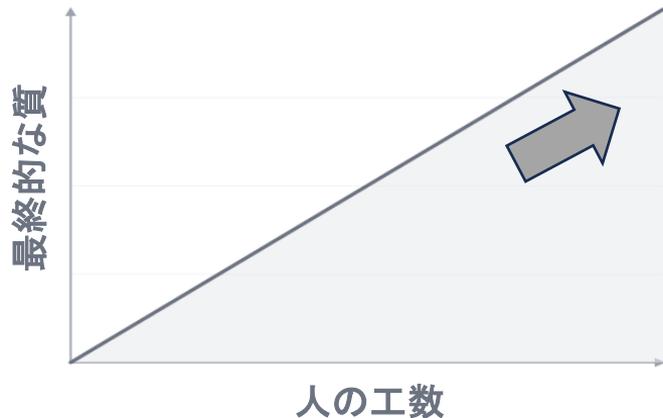
- ✓ AI Nativeへの第一歩として、Agent Skillsの重要性を伝えたい

生成AIがもたらした変化は？

# 生成AIがもたらした変化: 人駆動からAI駆動へ

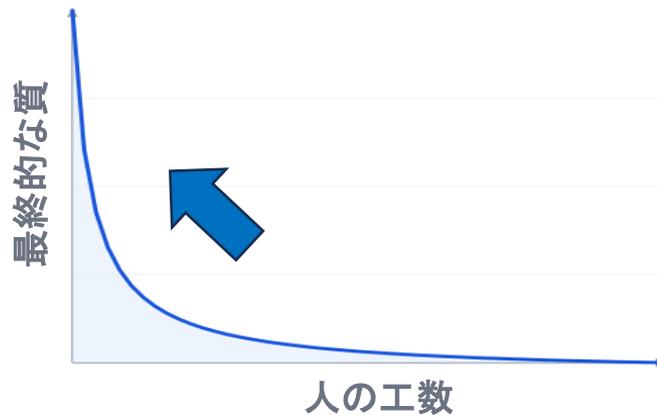
## 生成AI(GPT-4)の登場以前・以降

### 生成AI以前



AIは道具 時間をかけて磨く

### 生成AI以後



AIと壁打ち 迅速に試行錯誤

# AI駆動 - とはいえまだ万能ではない

## LLMの限界



ハルシネーション

対策: LLMの継続的学習、RAG



直したバグを掘り返すなど 能力のムラ

対策: CoT, プロンプトで知識を補強

LLMや  
エージェントサービス  
の継続的改善



コンテキストウィンドウ上限が厳しい

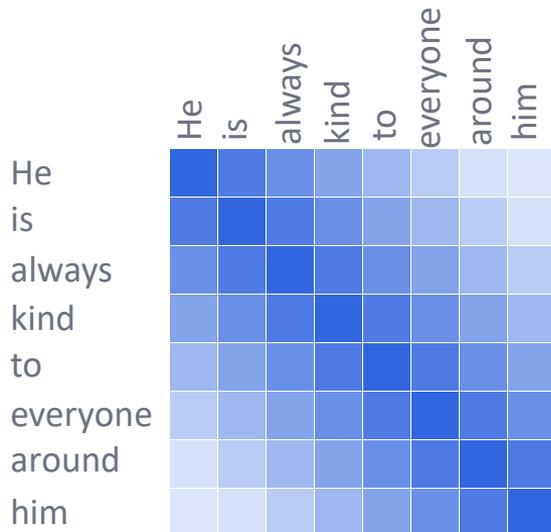
対策: 効率的なコンテキスト管理

エンドユーザーの  
力量次第

# コンテキストウィンドウ なぜ有限か？

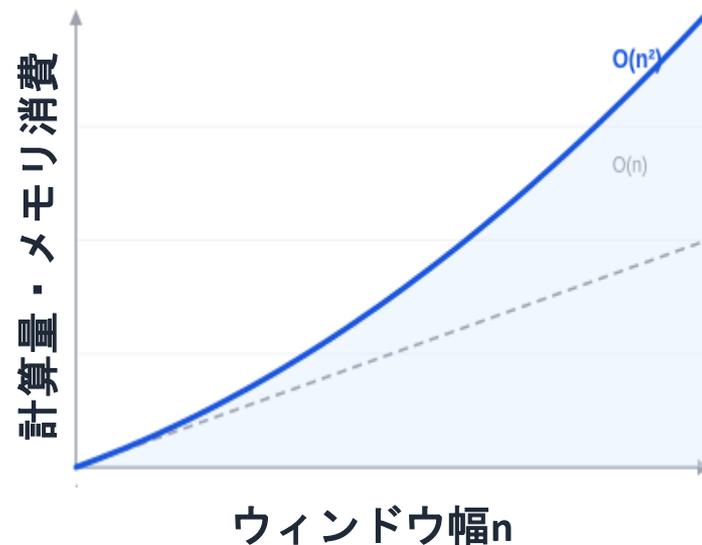
## LLMの核: Self-Attention

Self-Attention: 単語間の関連度の総当たり表



8単語x8単語  
=64組

計算量  
 $O(n^2)$



Attention計算が $O(n^2)$ で増大  
-> GPUメモリ・コストの圧迫を防ぐ安全弁が必要

# 実際のコンテキストウィンドウ幅

## 身近な開発業務で考える

Claude Sonnet 4.5 200K token

=> 1行50文字のマークダウンで約2700行

|                          |               |               |                   |             |
|--------------------------|---------------|---------------|-------------------|-------------|
| API, DBスキーマ定義<br>500行 x2 | コード規約<br>300行 | 他ポリシー<br>300行 | CLAUDE.md<br>300行 | 空き枠<br>800行 |
|--------------------------|---------------|---------------|-------------------|-------------|

問題:  
10ラリー程度の議論ですぐ頭打ち...



イシュー: 「トクパ」向上の仕組みをどう設計するか?

# 「トクパ」の良い仕組みとは？

## 先ほどの例: ドキュメント群を一括投入



トクパ改善の余地あり

# 「トクパ」の良い仕組みとは？

アイデア: 全体俯瞰してから深掘りしたい



タグ => コンテンツという段階的読み込み = Agent Skills

# Agent Skillsの設計思想: 段階的開示

pdf/SKILL.md

YAML Frontmatter

```
name: PDF Processing
description: Comprehensive PDF toolkit for extracting text and tables, merging/splitting documents, and filling-out forms.
---
```

Markdown

## Overview

This guide covers essential PDF processing operations using Python libraries and command-line tools. For advanced features, see `./reference.md`. If you need to fill out a PDF form, read `./forms.md` and follow its instructions.

## Quick Start

```
... python
from pypdf import PdfReader, PdfWriter

# Read a PDF
reader = PdfReader("document.pdf")
print(f"Pages: {len(reader.pages)}")

# Extract text
text = ""
for page in reader.pages:
    ... text += page.extract_text()
```

= スキル用途

常に読み込み

= スキル内容

スキル使用時のみ

pdf/reference.md

```
# PDF Processing Advanced Reference

This document contains advanced PDF processing features, detailed examples, and additional libraries not covered in the main skill instructions.

## pypdfium2 Library (Apache/BSD License)

### Overview
pypdfium2 is a Python binding for PDFium (Chromium's PDF library). It's excellent for fast PDF rendering, image generation, and serves as ...
```

pdf/forms.md

```
If you need to fill out a PDF form, first check to see if the PDF has fillable form fields. Run this script from this file's directory:
`python scripts/check_fillable_fields <file.pdf>`,
and depending on the result go to either the "Fillable fields" or "Non-fillable fields" and follow those instructions.

## Fillable fields
If the PDF has fillable form fields:
- Run this script from this file's directory:
`python scripts/extract_form_field_info.py <input.pdf> <fields.json>`.
...
```

= 参考資料

必要に応じて

# Skillsの価値: ① トクパ向上

検証: 本当にSkillsがトクパ向上に効くのか？

Skills 利用なし

Context Usage  
21k/200k tokens



「以下のプロンプトに従って、APIクライアントの基本構造を作成してください。  
...(プロンプト全文)」

+2万token

Context Usage  
41k/200k tokens



Skills 利用あり

Context Usage  
21k/200k tokens



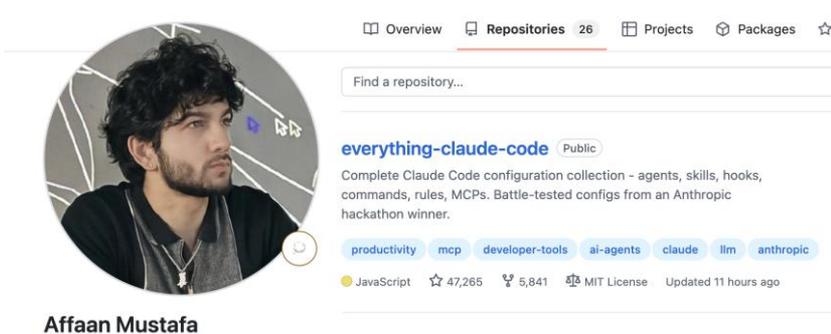
「**python-api-client** スキルを用いて、APIクライアントの基本構造を作成してください。」

+1万token

Context Usage  
31k/200k tokens

# Skillsの価値: ②技術が一瞬でスケールする

Anthropicのハッカソン優勝者がClaude Code設定を公開している！



Complete Claude Code configurati

☆ Star 47.3k

Jan 18, 2026, 10:49 AM GMT+9

最初のコミットから一ヶ月で★47k

トップエンジニアのナレッジに簡単にリーチできる

# Skillsの価値: ②技術が一瞬でスケールする

## Everything-claude-code: 開発関連Skillが充実



### 実装計画

/write-plan でタスク分解・依存関係・リスクを自動整理



### アーキテクチャ設計

Clean Architecture, SOLID原則に沿った設計レビュー



### テスト駆動開発

テストファースト開発を強制。Red → Green → Refactor



### セキュリティレビュー

Red-team / Blue-team / Auditor の3エージェント構成で脆弱性検出



### 言語別コードレビュー

TypeScript / Python / Go / Java Spring Boot 等、言語固有のベストプラクティス

トップエンジニアのスキル設定を雛形にできる

# ついでに得た気づき1

## Strategic Compaction: 前もってトークン上限を通知するHooks

### CC標準機能: Auto Compaction

### 必要な環境

- Python 3.11+
- pip
- venv

... +416 lines (ctrl+o to expand)

+ Compacting conversation... (3m 14s · ↑ 4.6k tokens)

⚠ トークン上限直前で自動圧縮

突然、重要な文脈が消失

### カスタムHooks: Strategic Compaction

```
"PreToolUse": [  
  {  
    "matcher": "Update|Write",  
    "hooks": [{  
      "type": "command",  
      "command": "node /home/btkoyamakanata/.claude/scripts/hooks/suggest-compact.js"  
    }]  
  },  
],
```

```
if (count === threshold) {  
  log(`[StrategicCompact] ${threshold} tool calls reached -  
  }  
  · consider /compact if transitioning phases`);  
}
```

✅ 前倒しで上限通知

計画的にコンテキスト整理できる

# ついでに得た気づき2

## Continuous Learning: 会話で得たナレッジを外部メモとして蓄積

### CC標準機能: Auto Compaction

```
Claude Code v2.1.1.34
Welcome back k2nt!
Sonnet 4.5 · Claude Max
~/Working/dvop-smaaz-dev-1tncr3/industry_research

*Conversation compacted (ctrl+o for history)
└─ Read report_schemas.py (66 lines)
└─ Read report_prompts.py (121 lines)
```

⚠️ 機械的な要約のみ引き継ぎ

しかも一時的キャッシュ



### Skill: Continuous Learning

#### How It Works

This skill runs as a **Stop hook** at the end of each session:

1. **Session Evaluation**: Checks if session has enough messages (default: 10+)
2. **Pattern Detection**: Identifies extractable patterns from the session
3. **Skill Extraction**: Saves useful patterns to `~/.claude/skills/learned/`

✅ セッション終了時にスキル抽出

重要なナレッジを永続化できる

# まとめ

---

- ✓ **Problem:** AI駆動の時代においてコンテキストウィンドウの狭さがネック
- ✓ **Solution:** 段階的開示の仕組みを持つ**Agent Skills**によって「トクパ」向上
- ✓ 副次的効果: トップレベルのエンジニアを簡単に憑依
  - ✓ 開発関連Skills
  - ✓ コンテキスト管理の知見

Agent Skillsを活用しAI nativeへの第一歩を 🔥

**Thank You**

---