

言語化する (＝スキル化する)

頭の中にある設計を、AIにスキル化を頼める形にまとめる



言語化とは、 感覚を「人に渡せる形」にすること

診断の「言語化」の軸は、自分のやり方を説明できるか、を見ていた。

そして説明する中身は、もう手元にある。

・第2回：委任の判断基準 ・第3回：検証スキル ・第4回：止めどきの条件

いまは頭の中とメモに散らばっている。

1か所に集めれば、AIにスキル化を頼める形になる。



なぜ、言語化が最後に来るのか

言語化は、それ以前の3つがある程度、形になっていないと、書くことがなくて空回りする。
逆に言えば、ここまで3回やってきたあなたには、すでに書くべき中身が揃っている。

その場かぎりの指示（プロンプト）は、ざっくりでいい。
繰り返し使う判断のルールは、理由まで書く。今回の話はこちら。

新しく何かを生み出す回ではない。すでにあるものを、人に渡せる形に「整える」回。



「自分のメモ」と 「渡した相手が再現できる文章」は別物

AIに「スキル化して」と頼むと、AIは作業を見て判断を推測して書く。
だから「なぜそうしたか」が抜けやすい。
抜けた前提は、自分の頭の中にだけ残っている。

「分かっている」と「渡した相手が再現できる」の間には、はっきりした段差がある。



実習：3つの設計を、1つのドキュメントにまとめる

並べる

Step1 3つの設計を並べる

内容・委任範囲・検証基準・止めどき条件を集める。整理はAIでよい。

なぜ

Step2 「なぜ」を1文ずつ添える

そう決めた理由を短く書く。ここだけは自分の仕事。今日の中心。

渡す

Step3 渡して再現できるか試す

人か別のAIに渡し、そのとおり動けるか確かめる。テストもAIに。



Step2 — 「なぜ」を添えると、どう変わるか

決定事項だけのルールは、他人には「なぜそうなのか」が分からず、調整できない。

✕ 3回まで直す（決定事項だけ）

○ 3回まで直す。トーンは基準を決めにくく、4回目以降は改善が頭打ちになるため

理由まで書いてあるからこそ、人はそれを自分の作業に当てはめ直せる。



Step3 どちらかの方法で、渡してみる

的外れな返しがきたら、書き方がまだ足りない証拠。そこを補っていく。

人に渡す

- ・友人・同僚・SNSで公開する
- ・その手順を、実際に真似してもらう

別セッションのAIに渡す

- ・新しいチャットに、そのドキュメントだけを渡す
- ・「私の代わりに検証をやってみて」と頼む



「人に渡せる」かどうかの、たった1つの基準

判断に迷ったら、この問いを使う。

このドキュメントだけを渡された人やAIは、一切質問せず、同じ判断ができるだろうか？

質問が浮かぶなら、書かれていない前提がまだ残っている。

その質問への答えを、ドキュメントに足していく。



質問が浮かぶうちは、 まだ「自分の中」にある

書かれていない前提を、1つずつ言葉にして、足していく。
地道だが、これが、感覚を人に渡せる形に変える作業。

質問が消えたとき、その型は、はじめてあなたの手を離れ、人のものになる。



今日のチェックリスト

- ・ 委任・検証・止めどきを1つのドキュメントにまとめた
- ・ それぞれの決定に「なぜ」を1文ずつ添えた
- ・ 人や別のAIに渡して、再現できるか試した
- ・ 「質問せずに同じ判断ができるか」で見直した

次回は、この型を実際の環境に置く。

Claude Codeなら hooks・skills・agents を
どこに置き、どう登録するか——ディレクトリ構造マップの回。

