# 生成AI時代の実務スキル成長法

**AIを使えば使うほど、上司のすごさが分かる**

— そんなあなたへ —

# 本日のアジェンダ

| セクション | 時間 | 内容 |
|---|---|---|
| **導入** | 5分 | 問題提起：「AIを使うほど上司のすごさが分かる」 |
| **パート1** | 10分 | 上司との差を知る：2つの力（①②） |
| **パート2** | 15分 | Spec駆動開発：AIを味方につける開発サイクル |
| **パート3** | 10分 | 知識の資産化：執筆で訓練（③） |
| **まとめ** | 5分 | 明日からのアクション |

# 導入

## 問題提起

# こんな経験、ありませんか？

### やったこと ✨

- 「AIでコードを爆速生成！」
- 「このライブラリ使えば簡単！」
- 「機能、実装できました！」

### 上司の反応 🤔

- 「なんでこれ作るの？」
- 「それ、本当に動くの？」
- 「1年後も大丈夫？」

> **「AIを使えば使うほど、上司のすごさが分かる...」**

# 本日のゴール

上司の「それ本当に大丈夫？」に

根拠を持って答えられる自分になる

— AIを味方につけて —

# AIを味方にすると、どう変わる？

| 上司のツッコミ | 今の自分 | AIを味方にした自分 |
|---|---|---|
| 「なんでこれ作るの？」 | 「…言われたので」 | 「〇〇のためです。△△は削りました」 |
| 「それ本当に動くの？」 | 「たぶん…」 | 「Issueで確認済みです」 |
| 「1年後も大丈夫？」 | 「……」 | 「メンテナンス状況も調べました」 |

> 今日は、この変化を実現する **具体的な方法** をお伝えします

# 根拠を持って答えるための「3つのアプローチ」

| アプローチ | 内容 | 本日のパート |
|---|---|---|
| **① 深堀り調査力を鍛える** | 表面で止まらず原文まで潜る | パート1-2 |
| **② 決める力を鍛える** | 何を作る/作らないを自分で判断 | パート1-2 |
| **③ 執筆で日々訓練する** | 書くたびに「答える力」が身につく | パート3 |

> この3つを回すことで、**上司の質問に自然と答えられる**ようになる

# 上司との差を知る

2つの力とは？

# 上司との差は「2つの力」

| 上司が持っている力 | AIの限界 | 呼び方 |
| --- | --- | --- |
| Issue/Discussionまで潜って調べる | 表面的な情報で止まる | **深堀り調査力** |
| 何を作るか/作らないか決められる | 全部盛りで出してくる | **決める力** |

> この2つの力を **意識的に鍛える** ことで、上司との差を埋められる

# 「深堀り調査力」とは？

| 観点 | AI | 若手 | 上司 |
|---|---|---|---|
| スピード | ◎ 爆速 | △ 時間かかる | △ 時間かかる |
| 網羅性 | ◎ 広い | △ 見落としがち | ○ 経験でカバー |
| 方向性 | ✗ 与えないと動けない | △ 迷いがち | ◎ 的確 |
| 深さ | △ プロンプト次第 | △ 浅い | ◎ Issue/Discussionまで |
| 最終判断 | ✗ できない | △ 自信がない | ◎ 根拠を持って決める |

# 「決める力」とは？

**必要最小限の原則**

- 本当に必要な機能だけを残す
- 「あったら便利」は削る
- シンプル = 保守しやすい

**AIは足し算が得意**

- 「〇〇も追加しましょう」
- 「△△も考慮すべきです」
- → **全部盛りになりがち**

> 上司の強みは **引き算** → 「それ、本当にいる？」と削れる

**AIに聞いた場合**

「〇〇の機能を作りたい」

↓

「こんな機能もあった方がいいですね！」

「ついでにこれも追加しましょう！」

「このパターンも考慮すると…」

↓

**全部盛り**

VS

**上司に聞いた場合**

「〇〇の機能を作りたい」

↓

「それ、誰が使うの？」

「今それ必要？」

「これだけでよくね？」

↓

**削ぎ落とし**

1

# ボトルネックは人間

AIは速い。網羅性もある。
深さも…プロンプト次第。

結局、**全部「人間次第」** じゃん。

👉 だからAIを味方につけて一緒に成長するんだ

# 【実体験】上司の経験値がすごかった話

若手

「このライブラリ良さそうです！」

上司

「ん〜、メンテ止まりそうだけど…」

↓ 半年後 ↓

**本当に開発終了…**

トレンドと使いやすさでは決められない

**上司は何を見ていた？**

過去の失敗・成功パターン

業界の動向

コミュニティの活性度

# Spec駆動開発

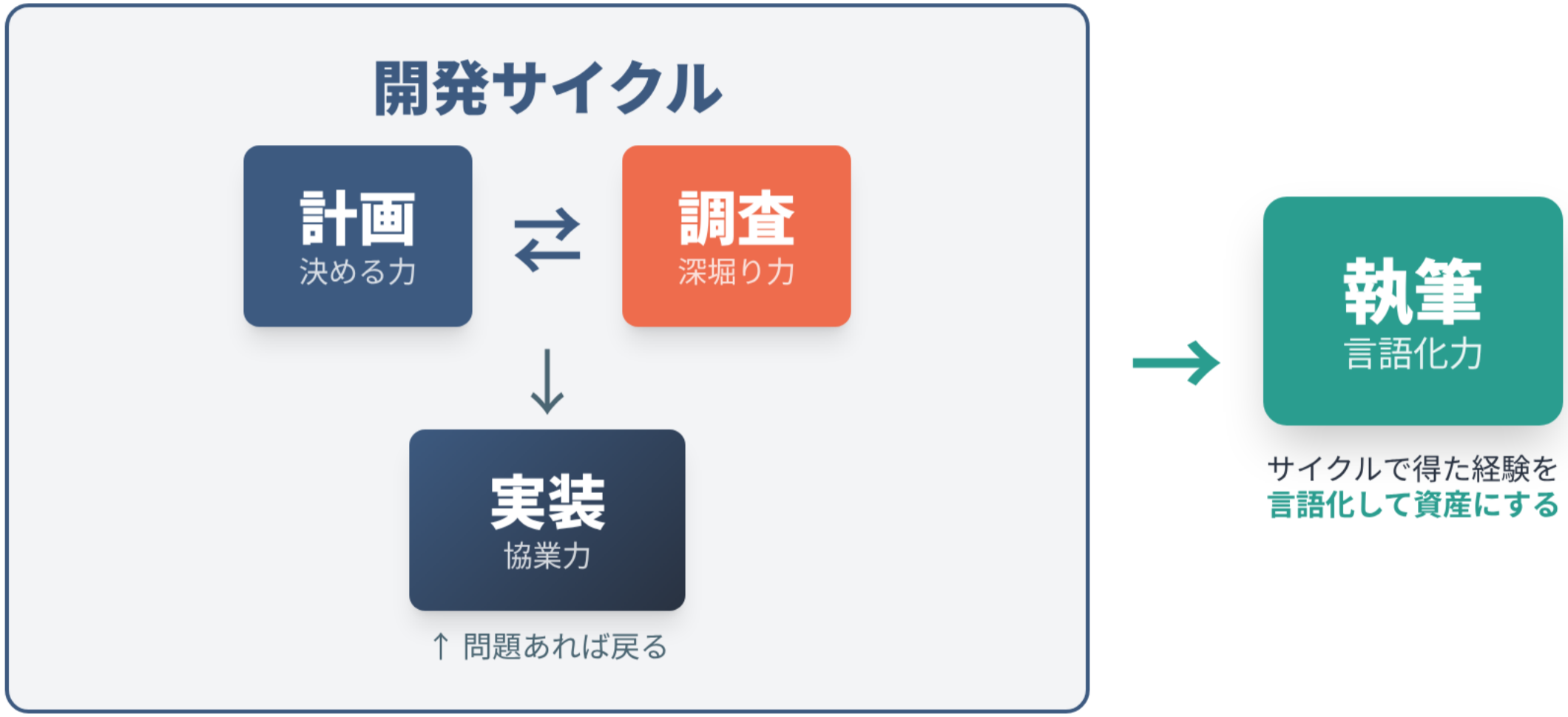
AIを味方につける開発サイクル

# Spec駆動開発とは？

**Spec（仕様書）をベースにAIと協業する開発スタイル**

| 従来の開発 | Spec駆動開発 |
|---|---|
| 頭の中のイメージで実装開始 | まず仕様書（Spec）を書く |
| AIに「〇〇作って」と丸投げ | AIにSpecを渡して指示 |
| 手戻り多発 | Specがあるから方向がブレない |

# Spec駆動開発


開発サイクル
計画
決める力
調査
深堀り力
実装
協業力
↑ 問題あれば戻る
執筆
言語化力
サイクルで得た経験を
言語化して資産にする

# 各フェーズの役割分担

| フェーズ | 位置づけ | 鍛える力 | AIに任せる | 自分でやる |
|---|---|---|---|---|
| **調査** | サイクル | ①深堀り調査力 | 速く広く集める | 原文を読んで判断 |
| **計画** | サイクル | ②決める力 | 選択肢を出す | 削ぎ落として決める |
| **実装** | サイクル | - | コードを書く | レビューする |
| **執筆** | 資産化 | ③言語化力 | 構成・校正を手伝う | 自分の言葉で書く |

# STEP 1: 調査

「深堀り調査力」を鍛える

# AIの調査力は半端ない

## AIの調査力

- ✓ **爆速**で情報を集める
- ✓ **複数ソース**を横断できる
- ✓ **要約・整理**もお手のもの

→ これを使わない手はない

→

## 使い方が大事

**助手**として使う
→ お任せする（受動的）

**ツール**として使う
→ 自分が操作する（能動的）

→ 主導権は自分

**AIの力を最大限引き出すのは自分次第**

# 調査フェーズで人間がやること

| 人間の役割 | 具体的にやること | AIに任せること |
|---|---|---|
| **① 方向を決める** | 「何を知りたいか」を明確にする | - |
| **② 指示を出す** | スコープ・観点・出力形式を指定 | 情報を集める |
| **③ 判断する** | 原文を読んで「本当か？」を確認 | 要約・整理 |
| **④ 足りなければ②へ** | 追加で調べるべき観点を洗い出す | 再収集 |

> **注意:** AIは立派なレポートを出してくるが、**原文を確認して記述が正しいかの検証は必須**

# 成果物の活用

AIが出力した
**調査ドキュメント**

→

1. **自分で読み込んで理解する**
   ここは**省略しない**
2. **計画に反映する**
   判断の**根拠**として
3. **プレゼン資料として流用**
   上司への**説明材料**に

**調査ドキュメントは「捨てる」のではなく「資産」にする**

# より良い計画は

# より良い調査から生まれる

知識がない？ → 調べればいい

それがAIを味方につけるということ

# STEP 2: 計画

「決める力」を鍛える

# 計画で問いかける3つの質問

**1 実現したい問題とは何か？**

「何を作るか」ではなく
**「何を解決するか」**

**2 不足している観点はあるか？**

自分が見落としている
**視点を探す**

**3 調査するべき内容とは？**

調査フェーズへの
**橋渡し**

**設計力に自信がない？ → AIを活用して突っ込んでいこう**

# 具体例：フォーム→API送信の場合

| 質問 | 若手の回答 | 上司レベルの回答 |
|---|---|---|
| ① 実現したい問題 | 「フォーム作る」 | 「ユーザーが迷わず入力完了できること」 |
| ② 不足している観点 | 「特にないです」 | 「エラー時のUXは？オフライン対応は？」 |
| ③ 調査すべき内容 | 「ライブラリの使い方」 | 「バリデーション設定時の処理は？」 |

> 問題を「機能」ではなく「解決したいこと」で捉える

# AIの活用ポイント

| 質問 | AIに聞くと良いこと |
| --- | --- |
| ① 問題の明確化 | 「この機能で解決したい課題を整理して」 |
| ② 不足観点の発見 | 「見落としてそうな観点ある？」 |
| ③ 調査内容の洗い出し | 「何を調べておくべき？」 |

> AIの「足し算」で観点を広げ、自分の「引き算」で絞る

# 良いSpec（計画書）とは？

実装時・実装後に

「なんか違う」「あれも必要だった」が発生しない

## ＝ 手戻りゼロ

# STEP 3: 実装

AIと協業する

# 実装は誰がやってもOK

**実装は人間がやってもAIがやってもOK**

大事なのは「**計画書ベース**」で動くこと

> 計画と調査が固まっていれば、実装は誰がやっても同じ方向に向かう

# AIに実装させる場合のポイント

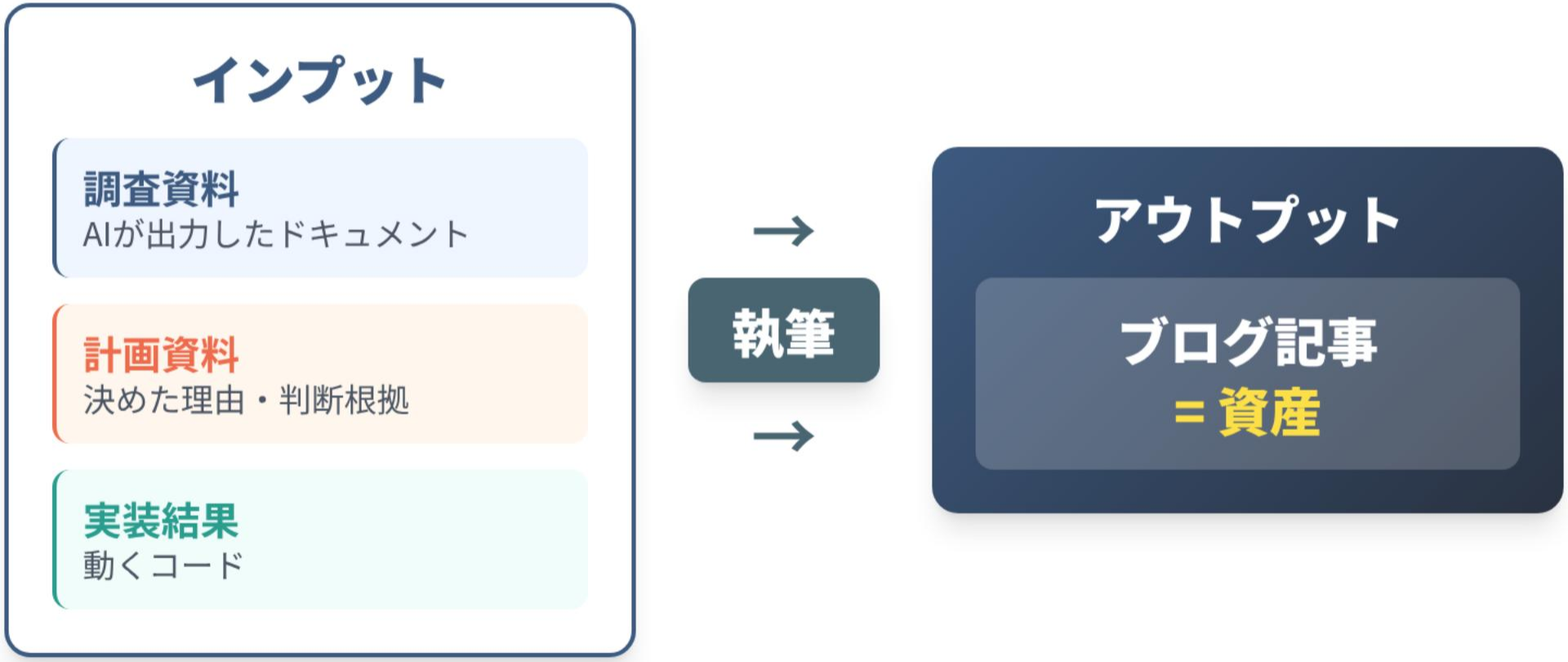
| ステップ | やること |
| --- | --- |
| ① ルールを明確にする | コーディング規約、使用ライブラリ等 |
| ② 行動計画を作成させる | 計画書から「何をどの順でやるか」を文章化 |
| ③ 実装に入る | 行動計画に沿って進める |
| ④ AIにレビューさせる | コードの品質・設計をAI or 機械的にチェックさせる |

> **カスタマイズは必須**（ディレクトリ構成・コーディングルール）
>
> **AIレビュー**で技術力・精度向上 → 自分では気づかない観点を指摘してもらえる

# 知識の資産化

執筆で定着・ブログを書く効果

# インプットとアウトプット



# 執筆でのAI活用

| 工程 | AIに任せる |
|---|---|
| 構成 | アウトライン案を出す |
| 下書き | 文章のたたき台を作る |
| 校正 | 誤字脱字・文法チェック |
| タイトル | キャッチーな案を複数出す |
| 図解 | 説明図・フロー図の作成 |
| レビュー | 読みやすさのフィードバック |

**やったことベースで記事を書く**
→ 悩まない

**執筆フローにAI活用を統合する**
→ AIは「編集者」として活用

# なぜ唐突に執筆なのか？

> 「計画・調査・実装の話をしていたのに、なぜ急にブログ？」
>
> 「執筆って、スキル成長と関係あるの？」

**本日のゴール:** 上司の「それ本当に大丈夫？」に **根拠を持って答えられる** 自分になる

# 執筆がスキルに変わる流れ

✍️
**執筆 = 教える**
読者に分かるように説明

→

🧠
**理解が深まる**
学習心理学で実証済み

→

💬
**答えられる**
上司の「なぜ？」に対応

| 効果 | 学習心理学の裏付け |
| --- | --- |
| 効果① 教えることで学ぶ | Protégé Effect |
| 効果② 書くと忘れない | Testing Effect |
| 効果③ 外部ストレージ | 認知負荷の軽減 |

# 効果① 教えることで学ぶ（Protégé Effect）

**読者に「説明する」と起こること**

1. **知識の再構築**
   - **分かったつもり**は通用しない
2. **メタ認知の強化**
   - 自分の理解度を客観視
3. **深い記憶への定着**
   - 教えた内容は忘れにくい

研究

**Stanford AAA Lab (2009)**

教える準備をした学習者は
そうでない学習者より
**テストで高得点**を記録

> **ブログ執筆 = 読者に教える = あなた自身の理解が深まる**

# 効果② 書くと忘れない（Testing Effect）

| やること | 記憶への効果 | 研究結果 |
|---|---|---|
| 資料を読み返す | △ 表面的 | 1週間後に40%忘れる |
| **書いてアウトプット** | **◎ 深く定着** | **再読の1.5倍の記憶保持** |

 研究

**Roediger & Karpicke (2006)**

「思い出す」作業が記憶を強化する→ 執筆 = 思い出しながら書く = 記憶が定着

> **読むだけでは忘れる。書けば残る**

# 効果③ 外部ストレージ

| 状況 | ブログがない | ブログがある |
|---|---|---|
| 同じ問題に再遭遇 | また1から調べ直し | 自分の記事を読めばOK |
| 後輩に質問された | 記憶を頼りに説明 | 「これ読んで」で済む |
| 1年後の自分 | 「何やったっけ…」 | 成長の軌跡が可視化 |
| **上司に聞かれた** | 「えーと…」 | **「記事にまとめてあります」** |

> **脳は忘れる。でもブログは忘れない**

# 執筆は「根拠を持って答える」訓練

| 執筆中の自問 | | 上司の質問 |
|---|---|---|
| **「なぜこう書く？」**<br>根拠を説明できる？ | = | **「なんでこれ作る？」**<br>根拠を説明できる？ |
| **「これで合ってる？」**<br>裏取りした？ | | **「本当に大丈夫？」**<br>裏取りした？ |
| **「読者に伝わる？」**<br>説明として成立してる？ | | **「1年後も大丈夫？」**<br>説明として成立してる？ |

**ブログを書くたびに「根拠を持って答える」訓練をしている**

# 余談：フィードバックは超絶チャンス

こんな経験ありませんか？

**強つよエンジニアのレビューでぶん殴られた**

こんな経験ありませんか？

**ブログ公開したらまさかり投げられた**

→

**超絶チャンス！**

そこには**ナレッジ**が詰まっている

将来の夢

**上司のナレッジを吸い出して**
**上司を模倣したAIエージェントを作る！**

それも
**知識の資産化！**

# まとめ

明日からのアクション

# 本日のメッセージ（再掲）

上司の「それ本当に大丈夫？」に

根拠を持って答えられる自分になる

— AIを味方につけて —

# 本日の整理

| パート | 内容 | ポイント |
|---|---|---|
| **Spec駆動開発** | 調査 → 計画 → 実装 | 良いSpec = 手戻りゼロ |
| **知識の資産化** | 執筆でスキル定着 | 書くことで「答える力」がつく |

| 鍛える力 | 方法 | AIとの役割分担 |
|---|---|---|
| 深堀り調査力 | 調査フェーズで原文確認 | AIで集める、自分で判断 |
| 決める力 | 計画フェーズで引き算 | AIは足し算、自分が引き算 |

# 明日からのアクション

## Spec駆動開発

**① 計画時に3つの質問を問う**

- 実現したい問題とは？
- 不足している観点は？
- 調査すべき内容は？

**② AIを「ツール」として使う**

- スコープ・観点・出力を縛る

## 知識の資産化

**③ 1つ記事を書いてみる**

- 小さくてOK、まず1本
- やったことベースで書く

**まず、やってみる**

# リソース紹介

**じゃあ具体的にどうするの？**

こちらのブログで「僕のもがき」が投稿されています

# ご清聴ありがとうございました

質問・フィードバックお待ちしています