

SOCIAL TRAINING / SEMINAR | セミナー

AI時代の 業務アプリ構築術

Claude Code を「最高のバディ」にする為の方法

まずは簡単な自己紹介から



会社紹介

ENロジカルは京大発 Tech×Educationカンパニーです

会社名	株式会社ENロジカル
設立年月日	2020年2月12日
代表者	廣瀬哲人
従業員数	30名(インターン、アルバイト含む)
所在地	京都府京都市左京区吉田牛ノ宮町6番1号
事業内容	法人向けAI研修事業 関西圏学生の人材派遣事業 デジタルマーケティング支援



INTRODUCTION

代表プロフィール

長崎県出身。

京都大学医学部医学研究科在学中に株式会社 ENロジカルを創業。受験コーチングサービスを立ち上げ、1年で事業売却。

その後、WEBマーケティング支援業を展開し、生成AIを活用したマーケティング支援に注力。

現在は、法人向けの生成AI研修講座をローンチし、これまでに10,000人以上の受講者に対して、延べ5,000回・10,000時間を超える受講数を記録する研修動画を制作。中小企業から大手企業、医療法人、自治体まで、幅広い分野におけるAI活用の現場を支援している。



廣瀬 哲人

株式会社ENロジカル
代表取締役社長

本日はお話できることを楽しみにしておりました。
どうぞよろしくお願いいたします。

**本日は最新のAI活用の方法を
リアルタイムで様々なクリエイティブを
作ることによって体感していただくセミナーと
なっております。**

**そもそも今AIはどんな種類があって
何を身につければいいのか？**

AIツールは「用途」で分かれている。

ChatGPT・Claude・Claude Code・Cursor・Gamma... 名前は無数。でも、用途で5カテゴリに整理すると、自分が今どこにいて、次に何を触ればいいのかが見える。

本日のテーマ

CATEGORY 01

文章生成AI

質問・要約・文章作成・調査の相棒

ChatGPT
Claude (Web)
Gemini
Microsoft Copilot

入口・誰でも今日から

CATEGORY 02

コーディングAI

業務アプリ・ツール・自動化を作る

Claude Code
Cursor
GitHub Copilot
Devin

今日の主役・伸びしろ最大

CATEGORY 03

スライド・資料

プレゼン・提案書・LPを生成

Gamma
Genspark
Canva AI
Tome

成果物が早く出る・効果実感◎

CATEGORY 04

画像・動画

ビジュアル素材を生成・編集

Midjourney
DALL-E
Sora
Runway

広報・販促で活躍

CATEGORY 05

音声・議事録

文字起こし・要約・読上げ

Notta
tl;dv
Whisper
ElevenLabs

会議・記録の負担減

ROADMAP | 非エンジニアが今日から触る順序

STEP 1 会話AIで慣れる
日本語でAIに頼む感覚を掴む



STEP 2 ★本日 コーディングAIで作る
業務アプリ・自動化を自分で組む



STEP 3 スライド・資料AIで発信
作ったものを社内へ広める



STEP 4 画像・音声AIで広げる
必要な領域から逐次拡張

コーディングAIとはなにか？

そもそも【Claude Code】って何ですか？

日本語で頼むだけで、業務アプリが出来上がるAI — それがClaude Code

プログラミングを学んだことがない方でも、話しかけるだけでツールが作れる時代になりました。

A NEW ERA | 新しい時代の始まり

「人事評価ダッシュボードが欲しい」
とパソコンに話しかけるだけで、
本当に動くアプリが出来上がる。

日本語OK

3つの安心ポイント | 非エンジニアでも大丈夫な理由

✓ コードを書かない

画面の前で日本語で話すだけ。「こうしたい」を伝えれば、AIが自動でコードを書く。

🕒 スピードが速い

これまで数ヶ月かかったアプリ作りが、数日～数時間で形になる。試行錯誤の壁が下がる。

🛡️ 業務知識が活きる

むしろ業務をよく知る人こそ強い。「現場で本当に使えるかどうか」を判断できる人が指揮官になる。

補足：今までは「IT部門にお願いする」しかなかった業務改善が、現場の手で形にできる時代になりました。

自然言語で指示 → 叩き台を即生成 → 対話で改善

非エンジニアでも、言葉で指示を出すだけでUIを組み立て・改善するサイクルを素早く回せる



これが「対話開発」—— 設計仕様書を書かずに、動きながら仕様を固めていく新しい現場の流儀

補足：第4部のライブデモでは、この3ステップで「人事評価DB」を実装する様子を公開（45分）

じゃあ【何ができる】の？

日常業務の【こんなこと】が出来ます

「アプリ」と聞くとハードルが高そうですが、要するに【自分の業務を楽にしてくれる道具】が作れます。



CAN DO 01

業務アプリを作る

「**人事評価ダッシュボード**を作って」と頼むだけ。一覧画面・グラフ・承認フローまで、ちゃんと動くものが出来る。

これまで：IT部門に依頼 → 数ヶ月待ち
これから：自分で作る → 数日で完成



CAN DO 02

資料・文書を作る

会議の録音を渡せば**議事録**に。データを渡せば**レポート**に。Excelの集計表もグラフ入りで自動作成。

これまで：会議のあと2時間かけて議事録作成
これから：会議終了の5分後にはメール配信



CAN DO 03

面倒な作業を自動化

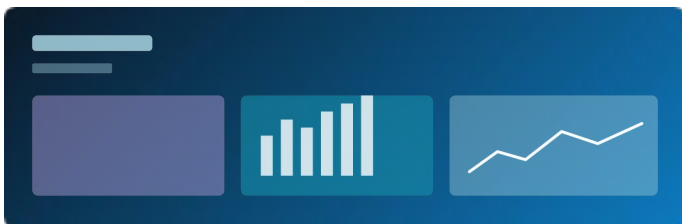
「毎月1日に**給与CSVを取り込んで通知メール**を送って」など、繰り返しの定型業務をまるごとお任せ。

これまで：月初に半日つぶれる手作業
これから：勝手に終わっている、確認するだけ

例えば、こんなものが作れます — 具体例

実際の作例で見てみましょう

いずれもプログラミング経験ゼロ、日本語で依頼するだけで、数時間～数日で動くものが出来上がります。



EXAMPLE 01

人事評価ダッシュボード

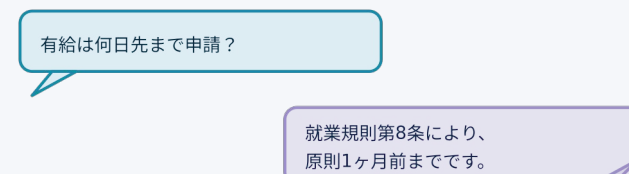
評価ステータス・推移グラフ・部門別の比較を1画面で確認できる業務アプリ。



EXAMPLE 02

議事録の自動作成ツール

録音データを渡せば、要約と決定事項リストを整えてメール配信。



EXAMPLE 03

社内FAQチャットボット

就業規則・経費規程を読み込ませ、Slackで自動回答する相談窓口。



EXAMPLE 04

週次レポート自動配信ツール

案件データから自動でグラフ入りレポートを作成し、毎週月曜に上司へ送信。

日付	出勤	退勤	時間
6/3 月	09:02	18:14	8.2h
6/4 火	08:55	19:30	9.5h
6/5 水	09:00	17:48	7.8h

EXAMPLE 05

勤怠管理アプリ

打刻・有給申請・残業集計を1画面に。月末の集計作業がワンクリックに。



EXAMPLE 06

経費精算アシスタント

領収書を撮るだけで日付・金額を自動読み取り、規程チェックまで実施。

補足：いずれも「日本語で頼む → AIが作る → 触って修正を頼む」を繰り返すだけで完成。本研修ではEXAMPLE 01を題材に進めます。

「SaaS is Dead」 — SaaSの死

「業務アプリは、買うものではなく、自分たちで作るもの」—— そんな空気が広がり始め、世界中のSaaS企業の株価が大きく揺れています。

GLOBAL IMPACT

▼ 2兆ドル

USD 2 TRILLION WIPED OUT

2025年末～2026年初にかけて、世界のSaaS関連企業から失われた時価総額。

出典: Tech Insider 「SaaS Stock Crash 2026」 / Forbes
「SaaS Apocalypse」

INDIVIDUAL STOCKS | 大手SaaS企業の2026年年初来下落率

Workday

人事・財務SaaS最大手

−36%

2026年初来・Reuters

Salesforce

顧客管理(CRM)SaaS最大手

−26%

2026年初来・Business Insider

Asana

プロジェクト管理SaaS

−36%

2026年初来・Business Insider

WHY | 投資家が織り込み始めた“内製化シナリオ”

01 コーディングAIが急成長し、自然言語だけでアプリが作れるように。

02 「SaaSを契約するより、自社で作る方が早く・安く・自由」へ。

03 年間サブスク契約の見直しが相次ぎ、SaaS各社の業績見通しが悪化。

HANDS-ON TIME

それでは早速、
作ってみましょう。
↳コーディングAIの最新の使い方

百聞は一見に如かず。

よりコーディングAIを使いこなすには？

A QUESTION FOR YOU | はじめに、一つだけ問わせてください

CursorもClaude Codeも、これほど強力なツールが揃った。
それなのに、なぜ——

なぜ、Claudeで作った業務アプリは 現場で**使われない**のか？

皆さんも一度は、こんな声を聞いたはずです。

VOICE 01

「動くけど、誰も触らない」

デモは成功。本番ではログインすらされない。

VOICE 02

「直しても、また壊れる」

仕様変更とバグ修正の無限ループ。

VOICE 03

「担当が抜けたら誰も触れない」

仕様書もテストも残らず、メンテ不能。

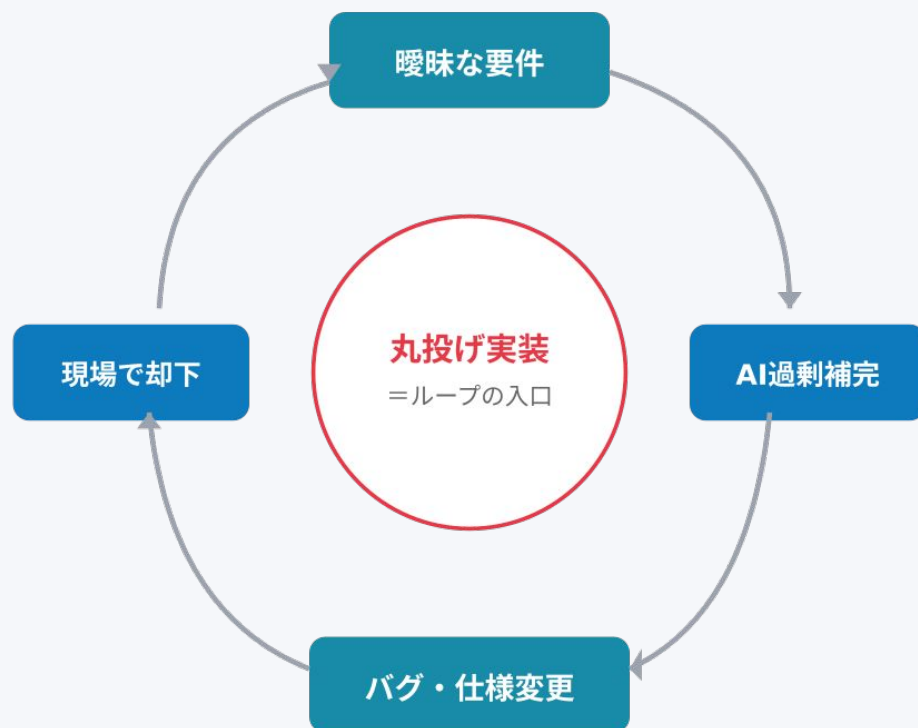
原因はツールの性能ではない —— 【プロセスの欠如】である。次のスライドで、その構造を解剖する。

現代の課題 — 手戻りループという罠

AIに頼んでも【使い物にならないアプリ】が生まれる本当の理由

原因はツールの性能ではなく、要件を固めずに実装に入る「プロセスの欠如」にある

REWORK LOOP | 手戻りループ図



CURRENT SYMPTOMS | 現場で起きていること

現象1 デモは動くが本番では使われない

画面はそれらしく出来るが、承認フローや権限境界が抜け、現場の業務に組み込めない。

現象2 バグ修正と仕様変更が無限ループする

「直しては壊れる」を繰り返し、工数の8割が手戻りに消費される事態が日常化する。

現象3 担当者が離脱するとメンテナンス不能

仕様書もテストもなく、生成コードだけが残る。属人化により改修・運用が止まる。

【丸投げ実装】こそが手戻りループの入口である

アジェンダ — 覚えるのは、たった3つの単語

本日お持ち帰りいただくのは、3つのキーワードだけです

この3単語が腹落ちすれば、Cursor / Claude Code を「最高のバディ」に変えられる。

KEYWORD 01

グリル

Grill

AIに自分を質問攻めにさせる。要件はAIに書かせず、暴かせる。

→ P5～P11（要点1）

KEYWORD 02

コンテキスト

Context

AIに業務知識を食べさせる。賢さは情報の量と質で決まる。

→ P12～P17（要点2）

KEYWORD 03

マルチエージェント

Multi-Agent

AIに相互レビューさせる。1人を信じず、4ペルソナで死角を潰す。

→ P18～P22（要点3）

01

要件定義は 「書かせる」 のではない

グリル&プラン / Cursor / Claude Code を前提に

正しい順序は【AIに暴かせる（グリル）】

人間が書ける仕様は穴だらけ。AIに逆質問させて隠れた前提を全て表面化する。

CONVENTIONAL | 従来のやり方

- 1 人間が頭の中で**曖昧な仕様**を書き起こす
- 2 AIが行間を**勝手に補完**して実装する
- 3 現場でズレが発覚し**手戻り**発生

補足：忖度しないAIほど、曖昧さがそのまま破綻として表出する。

RECOMMENDED | 推奨アプローチ

- 1 AIに**逆質問**させて穴を表面化
- 2 Planモードで章立て・**受け入れ条件**を合意
- 3 合意済みPlanを唯一の入力として**実装**へ

補足：グリル＝仕様の耐震診断。穴を可視化してから建てる。

補足：Cursor / Claude Code いずれもPlan先行が有効。コード生成は最終工程に位置づける。



忖度しないAIは、 曖昧さをバグで返す。

人間の同僚なら「察して」くれた前提を、AIは一切埋めない。
曖昧な指示はそのまま**ハルシネーション**として実装され、本番のバグになる。

そもそも【 Grill 】とは何か

Grill (grill) = AIに自分を質問攻めにさせる技法

語源は「直火で焦げ目がつくまで炙る」—— 仕様を逃げ場なく問い詰め、曖昧さを焼き切る、という比喻。

DEFINITION | 定義

Grill

= AIに先に質問させる

人間が要件を書くのではなく、AIに「分からない点」を10個質問させる。

回答するのは事実だけ。曖昧さが残れば、再度 Grill。穴がなくなるまで繰り返す。

METAPHOR | 3つの比喻で理解する



記者会見の記者

「なぜ・誰が・いつ・どれくらい？」—— 同じ事実を別角度から何度も問い直す姿勢。



熟練医の問診

「他に痛むところは？薬は？」—— 患者が言わない症状を、質問の連鎖で引き出す。



検察官の反対尋問

「本当にそう言い切れますか？」—— 主張の矛盾と前提を全て炙り出し、逃げ場を消す。

合言葉は3つ — 「分からない点を10個」 / 「事実だけ答える」 / 「穴がなくなるまで繰り返す」

グリルスキル — 誰でも今日から使える【公開スキル】

グリルは【スキルとして公開】されている — 自前で組まなくてよい

プロンプトを試行錯誤で作る時代は終わり、選んで呼び出す時代へ。AIに「グリルスキル」を装着するだけで使える。

PUBLISHED SKILL | 社内・社外で再利用可

v1.0

FREE TO USE

Grill Skill

要件を暴くための逆質問エンジン

- 1 曖昧要件から10個の質問を自動生成
- 2 回答が出るまで自動で深掘りを継続
- 3 最後に仕様書1ページを整形出力

Genspark Skill Gallery / Marketplace から呼び出し可能

HOW TO USE | 3ステップで装着

01 スキルを選択する

Genspark/Cursor/Claude Code の Skill Gallery から「Grill Skill」を選択。インストール不要、その場で有効化される。

02 要件を投げるだけ

「人事評価ダッシュボードを作りたい」など1行でOK。プロンプト設計や追加指示は不要——スキルが質問を引き出す。

03 仕様書1ページを受け取る

グリル完了後、対象範囲・ロール権限・承認状態・受け入れ条件が揃った仕様書が出力。そのままPlanモードに入れる。

補足：プロンプトのコピペ管理から解放される。スキルは社内で共有・改訂すれば、品質のばらつきもなくなる。

Why — AIは忖度しない

AIは【忖度（そんたく）しない】 — 曖昧さは必ずバグとして可視化される

「察してくれる人間の同僚」を期待した瞬間、ハルシネーション（もっともらしい誤実装）が本番に流れ込む

曖昧な仕様 | 典型的な指示

CASE 01

「評価を見やすくしてほしい」

CASE 02

「いい感じのダッシュボードを」

CASE 03

「権限・期間・承認状態」が未定義

そのまま実装される

起きること | 現場で発生する事故



ハルシネーション

もっともらしい誤実装。レビューが甘いと本番に流出する。



バグとして本番に露出

権限漏れ・誤集計など、人事評価では信頼に直結する障害。



現場の信頼を損なう

一度「使えない」と評価されると、内製化全体が頓挫しやすい。

Example — 人事評価ダッシュボードのグリル実演

「評価推移をグラフ化したい」だけでは、1行たりとも書けない

AIに逆質問させ、5つの観点で穴を埋める。回答が揃って初めて【仕様書1ページ】が完成する。

BEFORE | 曖昧な1行要件

「評価推移を グラフ化したい」

- AIが勝手に解釈して実装
- 現場ロールとズレた集計軸
- 承認前データまで見えてしまう
- 結果、再実装3往復

PITFALL

「自分も即答できない質問」が、今のあなたの仕様の穴である

補足：グリル後のPlanで画面ワイヤーを合意し、実装に進む。

GRILLING | AIからの逆質問

- Q1 対象は全社員か**特定部門**か
- Q2 **下書き評価**を含めて表示するか
- Q3 **ロール別**に閲覧範囲はどう変わるか
- Q4 縦軸は**総合点**か**目標達成率**か
- Q5 **承認ステータス**でフィルタするか



AFTER | 確定仕様1ページ

SPEC 01
ロール別閲覧範囲を定義

SPEC 02
承認済みのみ表示

SPEC 03
総合点・達成率の2軸グラフ

SPEC 04
CSV項目定義・入出力



仕様が固まるまで、
1行もコードを書かせるな。



02

賢さは
【情報の
量と質】で
決まる

コンテキスト飽和 / Context Saturation

コンテキストウィンドウを業務知識で【飽和】させる

同じモデル・同じプロンプトでも、供給する文脈の量と質が出力品質を別物に変える。

LAYER 01

入力 — 業務知識の塊を渡す

評価規定PDF・テーブル定義・CSVサンプル・過去Excel・ヒアリング音声 —— これらをすべて一つのプロジェクトに集約する。

補足：人事評価DBでは「規定」「ロール」「サンプルデータ」の三点が中核。

LAYER 02

処理 — AIが文脈を参照しながら生成

RAG的に必要な資料を都度引き、音声ダンプで語られた前提を判断に組み込む。汎用解ではなく、自社解を出す。

補足：Cursor / Claude Code の @ファイル参照やプロジェクト機能を活用する。

LAYER 03

出力 — 業務に整合したコード・UI

承認フロー・ロール権限・列名と型まで現場仕様に揃った成果物が、最初のドラフトから出てくる。

補足：以後の修正回数が劇的に減り、レビュー工数も短縮される。

補足：モデル名を上げる前に、あなたが供給する文脈の総量を上げる。これが成果を分ける。



文脈なきAIは、 最大公約数の生成器にすぎない。

同じモデルでも、渡す文脈の量と質で出力は別物になる。
賢さはモデル名ではなく、あなたが供給する業務知識で決まる。

どうすればコンテキストを増やせるか

コンテキストには【5つの源泉】がある — それぞれ集め方が違う

「文脈を渡せ」と言われても、何を渡すか分からない—— 5源泉に分解すれば、抜け漏れなく集められる。

SOURCE 01

規定・ルール

Documents

評価規定PDF・就業規則・業務マニユアル・社内Wiki

▶ 集め方

PDFをリポジトリに配置
／Confluence・Notionから@参照

SOURCE 02

データ構造

Schema & Samples

DDL・ER図・既存Excelスキーマ
・CSVサンプル行

▶ 集め方

DBから直接DDL出力／サンプル数
件のCSVをリポジトリへ

SOURCE 03

既存コード

Existing Codebase

類似機能の実装・命名規則・テスト
コード・型定義

▶ 集め方

Cursor/Claude Code に @file で参
照／.cursorrules に規約

SOURCE 04

暗黙知

Voice / Tacit Knowledge

現場の呼び方・過去の失敗・なぜ
作るかの背景・例外運用

▶ 集め方

音声ダンプ→文字起こし／現場ヒ
アリング録音をMarkdown化

SOURCE 05

受け入れ条件

Acceptance Criteria

グリルで合意した「誰が・何を
・いつ見えるか」の3条件

▶ 集め方

Planモードで凍結／後続のQA
レビュー基準になる

5源泉のうち、ほとんどの現場で04（暗黙知）と05（受け入れ条件）が抜け落ちている。ここを足すだけで品質が一段変わる。

補足：次スライド（P15）以降で、なぜ文脈が要るか（Why）、5源泉をいつどう使うか（How）、人事評価例（Example）を扱う。

Example — 人事評価DBに「食べさせる」3つの資産

規定PDF・既存Excel・社員テーブル定義 — 3点セットが品質を変える

この3資産が揃うと、修正回数は体感で半分以下になる。コンテキストが「ある」と「ない」の決定的差。

ASSET 01



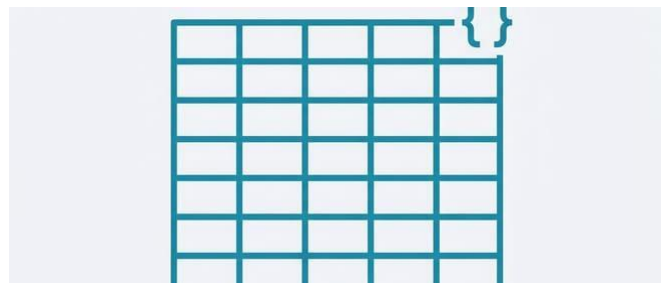
評価規定PDF

「ステップ2以上は**上長承認後のみ表示**」など、運用ルールをコードに反映させる。

運用例外も併記し、AIの推測余地を消す。

補足：規定をテキスト化しておくでハルシネーション抑制に直結する。

ASSET 02



既存Excel（スキーマ）

列名・型・集計ロジックをスキーマとして渡す。グラフの軸を誤らせない。

サンプル行を数件付けて、データ形のリアリティも渡す。

補足：CSV/DDDLで渡せば、AIは型エラーを自己検知できる。

ASSET 03



社員テーブル定義

ロール・部門・評価IDの関係をER図として渡す。

権限境界の根拠が明確になり、認可コードが正しく出る。

補足：この情報が次章のSecurityレビューで効いてくる。

補足：コンテキストの総量が増えるほど、修正回数は減る。これは現場で再現性のある法則となる。

03

1人のAIを 信じるな 【AIチーム】 を結成せよ

マルチエージェントレビュー / 4 Personas

4ペルソナで相互に批判・レビューさせる

実装AIとレビューAIの役割を分離し、人間のコードレビューと同型をAI内で再現する。



PERSONA 01

Architect

構成・拡張性・責務分離を攻撃。モジュール境界の崩れを検出する。



PERSONA 02

Security

認可・データ漏洩・IDOR・監査ログを検査。人事評価では最優先。



PERSONA 03

Senior Developer

命名・保守性・テスト容易性を点検。N+1や重複ロジックも検出する。



PERSONA 04

QA

受け入れ条件との差分と境界値を攻撃。Plan段階の条件と必ず照合する。

補足：実装したAIにレビューさせない。プロンプトで明示的に役割を分離する。



1人のAIを信じるな。 AIチームを結成せよ。

同じAIに作らせて直させると、同じ**バイアス**がそのまま残る。
Architect / Security / Senior Dev / QA —— 4ペルソナで死角を構造的に潰す。

Why — 単一プロンプトには構造的な限界がある

同じAIに作らせて直させると、同じバイアスがそのまま残る

死角は別ペルソナでないと見えない。これは人間のコードレビューと完全に同型の構造である。

1 知ったかぶりバイアス

実装したAIは自分のコードを**正当化**しがち。誤りを認めず、表面的な微修正で済ませる傾向が強い。

補足：「問題ありません」と返ってきた瞬間が一番危険。

2 セキュリティ死角

権限・**IDOR**・認可漏れは「機能が動くか」という観点では発見できない。視点が違う担当者が必要。

補足：人事評価DBでは最も致命的な事故源となる。

3 パフォーマンス死角

小規模データでは動くが、本番で**N+1クエリ**や非効率ループが顕在化。机上では発見が難しい。

補足：Senior Devペルソナがクエリ計画とキャッシュ戦略を点検する。

4 UX死角

承認者の操作導線・差戻し・コメント機能の抜けは、QAペルソナが受け入れ条件と照合して炙り出す。

補足：実装者AIは「機能要件は満たした」で止まりがち。

How — 4ペルソナレビューの実装サイクル

プロンプトで4役を【明示的にシミュレート】し、1サイクル回す

「あなたはSecurity担当として、認可とIDORの観点でこのコードを攻撃せよ」と役割を明確に与える。

ATTACK POINTS | 4ペルソナの攻撃観点

PERSONA 01 Architect

構成・拡張性・**責務分離**を攻撃。モジュール境界の崩れと、将来の規定改定への耐性を点検する。

PERSONA 02 Security

認可・漏洩・**IDOR**・監査ログを攻撃。URLパラメータ書き換えで他人の評価が見えないかを必ず確認する。

PERSONA 03 Senior Dev

命名・保守性・**テスト容易性**を攻撃。N+1や重複ロジックなど、技術的負債の芽を早期に摘む。

PERSONA 04 QA

受け入れ条件との**差分**・**境界値**を攻撃。Plan段階の条件と1項目ずつ突合してすき間を炙り出す。

REVIEW CYCLE | 1サイクル

1 実装AIがコードを生成



2 4ペルソナが個別にレビュー



3 指摘を集約 → 実装AIが修正



4 再レビュー（最大2周）

人事評価ダッシュボードでは、Securityペルソナを最優先で実行する。

補足：3周以上回るのは設計に問題がある兆候。Planに戻って見直す。



コードを書くな、 AIを調律せよ。

グリル・コンテキスト供給・レビュー統合・最終承認——人間の役割は**指揮官（オーケストレーター）**へ。
コードを書けない人でも、業務を語れる人なら誰でも指揮官になれる。

ここからAI人材育成について



AI人材は2種類。混ぜて育てると失敗する

「**作る人**（業務アプリ／エージェント開発）」と「**使う人**（LLMで日々の業務補助）」—— 育成プロセスも考え方もまるで違う

本セミナーは2種類とも扱う。ただし「主役は作る人」—— これが日本企業のAXIに最も足りない層

TYPE A / 本セミナーの主役

TALENT TYPE A

作る人 = 業務アプリ／ エージェントを開発する人

使う道具・成果物

- ・ 道具：Claude Code／Devin／Cursor＋devcontainer
- ・ 成果物：動くアプリ・エージェント・自動化スクリプト
- ・ 任せる単位：タスク～プロジェクト（聞くではなく「頼む」）

育成プロセス

- ・ 要件定義 → 受入条件 → 反復実装で鍛える
- ・ L2～L4に対応（部門ビルダー以上）
- ・ 想定6～18ヶ月、実プロジェクトでOJTが必須

TYPE B / 全社員ベース

TALENT TYPE B

使う人 = LLMベースで 日頃の業務補助に使う人

使う道具・成果物

- ・ 道具：ChatGPT／Claude／Gemini（チャット型）
- ・ 成果物：文章・要約・下書き・調査メモ
- ・ 任せる単位：1問1答（質問・要約・校正）

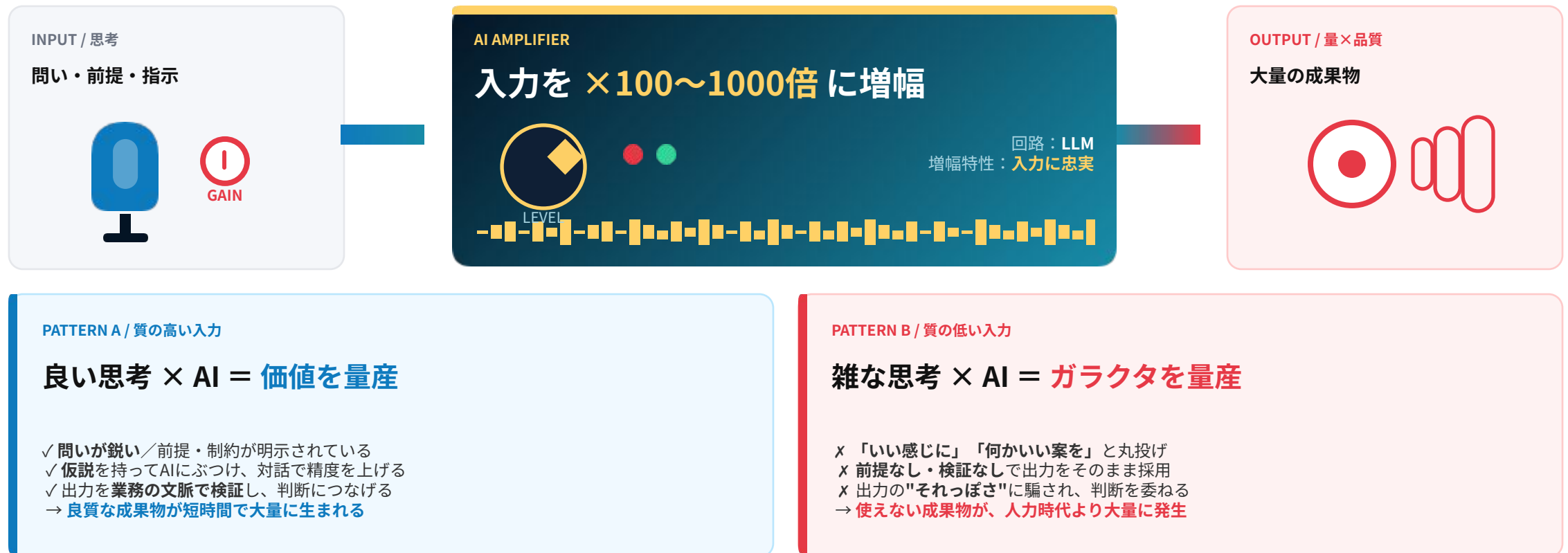
育成プロセス

- ・ プロンプト型・社内利用ルールで底上げ
- ・ L1全社員に対応（管理職含む）
- ・ 想定2～3ヶ月、eラーニング＋業務適用課題で完結

この2つは「ピラミッドの上下」ではなく「別カリキュラム」。混ぜて教えるとどちらも中途半端になる

AIはアンプ。質が低ければガラクタを量産する

AIは入力を増幅するだけ —— 入った思考の質が、そのまま出力の質×量になる



CONCLUSION / 育成投資の根拠

AIは思考を補えない。だからこそ人の思考力とAI活用スキルを両軸で鍛える育成投資が必要 —— 次スライドで両軸モデルを定義する

ヒューマンスキル × AI活用スキル — 両軸で高めるしかない

どちらか片方だけでは届かない — 掛け算で成果が決まる、だから両輪育成が要る

MATRIX / 二軸で見る人材ポジション

▲ ヒューマンスキル

高H × 低AI

考える力が高い
でも手が遅い

良い問いを立てられるが、
手作業のスピードで疲弊

★ 高H × 高AI

考えてAIに任せる
レバレッジ人材

良い問いをAIで増幅。
少人数で大きな成果

低H × 低AI

作業者として
淘汰圏

指示待ち+ツール未習得。
置き換えのリスクが大

△ 低H × 高AI

ガラクタ量産
"AIっぽい人"

出力を検証できない。
"それっぽさ"で意思決定

問い・論理・対話・判断

プロンプト・ツール・連携設計

AI活用スキル ▶

AXIS 1 / ヒューマンスキル（思考の質）

AIに"何を考えさせるか"を決める力

- 問題発見・課題分解：何を解くべきかを定義する
- 論理思考・仮説構築：前提と筋道を組み立てる
- 対話・要件定義：相手・顧客の真の意図を引き出す
- 判断・検証：出力の妥当性を業務文脈で評価する

AXIS 2 / AI活用スキル（増幅の技術）

AIに"どう仕事を任せるか"の技術

- プロンプト設計：前提・制約・出力形式を明示する
- ツール選定：Chat／IDE／Codeエージェントの使い分け
- 業務連携設計：SaaS・データ・ワークフローへ組み込む
- セキュリティ・倫理：守るべき情報・判断の境界を引く

CONCLUSION / なぜ両軸なのか

成果 = ヒューマンスキル × AI活用スキル — 片方がゼロなら成果もゼロ。だから研修は両軸を同時に鍛える設計が必要

AIで課題解決ができる人になるための5つの能力

プロンプトの上手さではなく **思考のフロー**——「問いの設計→分解→対話→検証→意思決定」を回せること

出典：ENロジカル「生成AI実践講座」設計／PwC・JUASのスキル定義／本セミナー登壇者の現場知見を統合

SKILL 01

問いの設計力

何を解くべきかを
言語化する

習得すること

- ・課題と原因を分けて記述
- ・前提・制約・成功条件の明示
- ・AIへの「指示」ではなく「依頼」として書く

陥りがちな失敗

「いい感じに作って」
「何かいい案ない？」

SKILL 02

課題分解力

塊で渡さず
サブ問いに割る

習得すること

- ・MECEで論点を切る
- ・5段以内のサブ問いに分解
- ・独立して検証可能な単位にする

陥りがちな失敗

「全部まとめてやって」
大雑把な丸投げ

SKILL 03

対話設計力

AIを壁打ち相手・
前提確認役にする

習得すること

- ・反復型／メタ型／
ソクラテス型を使い分け
- ・AIから質問させる側になる
- ・1問1答で終わらせない

陥りがちな失敗

一発で結論を求める／
会話を捨てて再質問

SKILL 04

検証・批判力

出力を鵜呑みにせず
裏取りする

習得すること

- ・ハルシネーションを見抜く
- ・出典・根拠を確認するクセ
- ・受け入れ条件で合否判定

陥りがちな失敗

出力をそのままコピペ／
"それっぽい"で採用

SKILL 05 / 中核

意思決定力

最終判断を
人がやる

習得すること

- ・決定／未決／次の
アクションに整理
- ・責任を持って選択する
- ・HITLを業務に組み込み

陥りがちな失敗

「AIがそう言った」を
判断の根拠にする

CONCLUSION / 能力定義

「AIを使えるか」ではなく「AIと一緒に課題解決を回せるか」—— 5能力のセットで測る

ENロジカルが提供する AI人材育成コンテンツ



4プログラムのラインナップ

全社の標準化から始まり、最先端AIの活用、高度な業務アプリ内製化までLv.0～10をカバー

① 基礎講座（Eラーニング）

 全社の土台づくり・標準化

形式: 動画オンデマンド

時間: 30時間・全86本

推奨Lv.: 0～5

② Google Workspace AI活用講座

 最先端AIの実感と機運醸成

形式: Zoomリアルタイム / 10時間・全5回

推奨Lv.: 3～7

※応用から始め、目に見える成果で導入を加速

③ AIワークフロー構築講座（Dify）

 ワークフロー化・自動化・RAG

形式: Zoomリアルタイム

時間: 10時間・全5回

推奨Lv.: 5～8

④ AI業務アプリ制作講座（Claude Code）

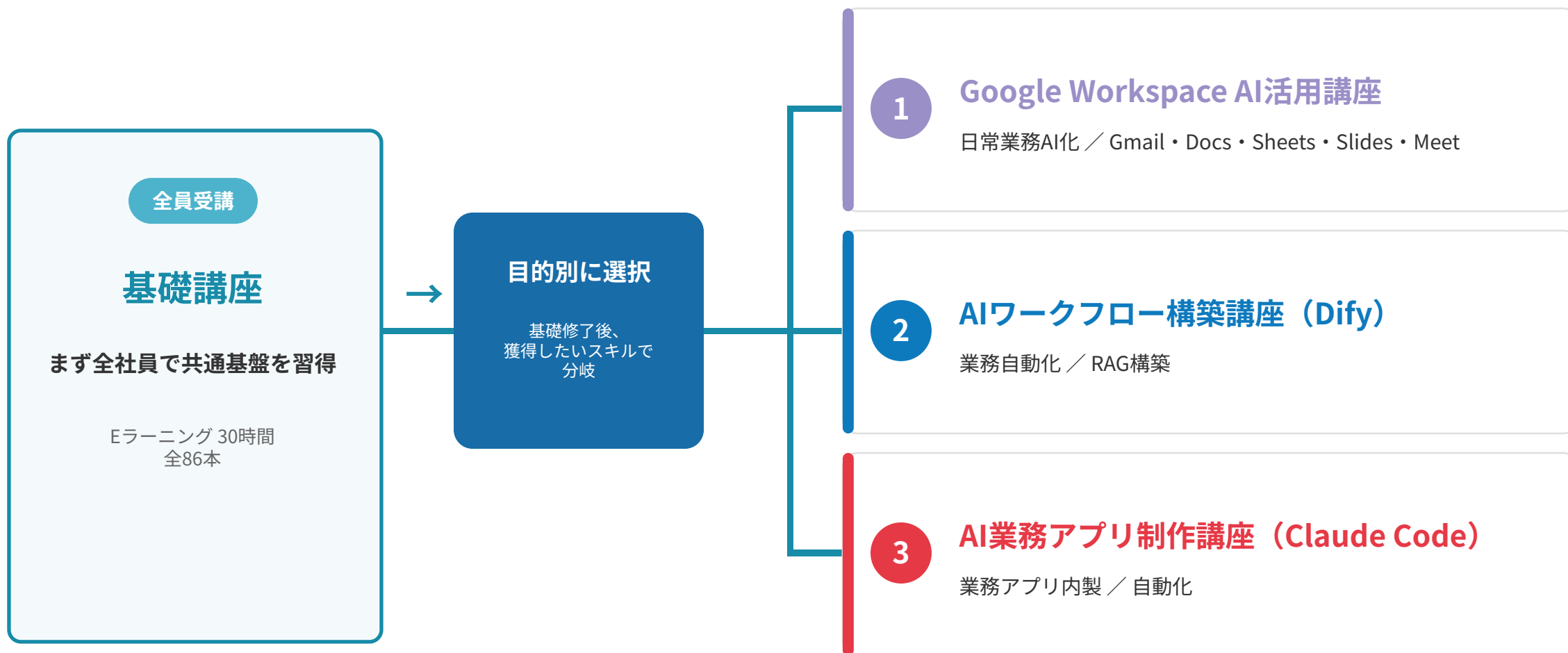
 専用ツール・業務アプリ化

形式: Zoomリアルタイム

時間: 10時間・全5回

推奨Lv.: 4～8

おすすめ受講ロードマップは基礎→目的別応用の順が最短



※ 基礎講座で全員の共通基盤を作り、その後に目的別の実践講座へ分岐する構成です。

トップダウン型 / 推奨受講期間：約1～2ヶ月

幹部・マネジメント層

学習目的

最新AIの体感と、経営視点でのDX推進知見の獲得

到達目標

自社のビジネスモデルにAIを組み込みAX化を牽引する

STEP 1 Google Workspace (最新AIの体感と戦略視点の獲得)

STEP 2 Dify講座 (最先端の自動化・RAG構築を体感)

STEP 3 Claude Code (アプリ内製化の全体像把握)

STEP 4 基礎講座 (補完) (全社共通知識の補完)

ボトムアップ型 / 推奨受講期間：約3～6ヶ月

一般社員・実務担当者

学習目的

自身の業務の棚卸しと実務における実践的なAI活用

到達目標

自律的に業務を改善・効率化できる「AI自走社員」への成長

STEP 1 基礎講座 (AIの基本と業務棚卸し)

STEP 2 Google Workspace (日常業務の改善と自動化)

STEP 3 Dify講座 (自動化・RAG構築の実践)

STEP 4 Claude Code (専用ツールの内製化)

AI化レベル10段階と4講座網羅スコープ

基礎講座を土台にDify・Claude Codeで内製体制へ到達し、
Google Workspace AI活用講座がLv.5～10の活用を横断的に底上げします。

- 基礎講座：Lv.0～5の土台づくり
- Dify+Claude Code：Lv.6～10の内製体制化
- Google Workspace：Lv.5～10を横断し、身近な業務からAI Studio/Vertex AIまで活用イメージを具体化

Lv	活用レベル	企業現場でのイメージ	主な到達講座	
0	未経験	AI未経験。まずはChatGPT/Claude/Geminiの違いと安全な使い方を理解する段階	基礎講座	
1	基本操作	検索・壁打ち・たたき台作成をAIに任せ、調査や企画メモの初速を上げる	基礎講座	
2	日常活用	メール返信、議事録、要約、報告書ドラフトなど日常業務をAIで短縮する	基礎講座	
3	プロンプト活用	プロンプト改善で出力品質を安定化し、部門内で使える定型プロンプトを整備する	基礎講座	
4	業務適用	自社業務に合わせてFAQ、文書作成、レビュー補助などの活用シーンを業務へ組み込む	基礎講座	
5	定型化	基礎講座の内容をルール化し、Workspaceでも日常業務のAI化を体感する	基礎講座	+ Google Workspace
6	部署実装（Dify/Google）	DifyでRAG構築、Google Apps ScriptやAppSheetで簡単な自社自動化を始める	基礎+Dify	+ Google Workspace
7	連携自動化（SaaS/WF）	SaaSや社内ワークフロー、Googleの複数アプリを横断接続して定型作業を自動化	基礎+Dify	+ Google Workspace
8	専用ツール・業務アプリ化	Claude Codeで現場専用ツールを内製、高度なVertex AI機能も視野に入れる	+ Claude Code	+ Google Workspace
9	全社横展開	基礎・Dify・Claude Code・Googleの4講座を部門横断で展開し、全体最適化	+ Claude Code	+ Google Workspace
10	変革推進・内製体制化	全社のAI活用基盤として定着し、Googleの最新AI機能も含む改善が回る体制	全講座を統合	+ Google Workspace

どういふ教材を選んていくべきか？



ケーススタディの質＝育成投資の成否の8割

結局は現場課題に合わせた使い方ができるか—— 汎用教材だけで研修を組むと、必ず行動が変わらない

出典：Kirkpatrick Level 3（行動変容）／パナソニックコネクト・NRI公開事例／ENロジカル研修実績

比較軸	汎用CS（質が低い）	現場適合CS（質が高い）
AXIS 01 題材	「議事録を要約する」「メールを書く」など どの会社にも当てはまる例 。学んでも翌日の業務に紐づかない	受講者の 実業務データ・実フォーマット を題材化（営業日報／顧客FAQ／品質クレーム等）
AXIS 02 ゴール設計	「 AIを触ってみた 」で完了。受講後に何を持ち帰るかが曖昧	「 自分の業務でX時間／週を削減する 」等の具体目標。受講後30日のKPIを定義
AXIS 03 演習形態	用意された サンプル課題 を解いて模範解答を見て終わり。横展開のヒントがない	自分の業務課題をワークで持ち込み 、講師＋AIで分解→対話→検証。アウトプットは社内で再現可能
AXIS 04 フォロー	受講して終わり。 行動変容は本人任せ で2週間後には元通り	30日後のアウトカム共有会 ＋現場で詰まった箇所をAI／講師で再支援

CONCLUSION / 育成投資の分岐点

結局は「現場課題に合わせた使い方ができるか」—— ケーススタディの質が全てを決める

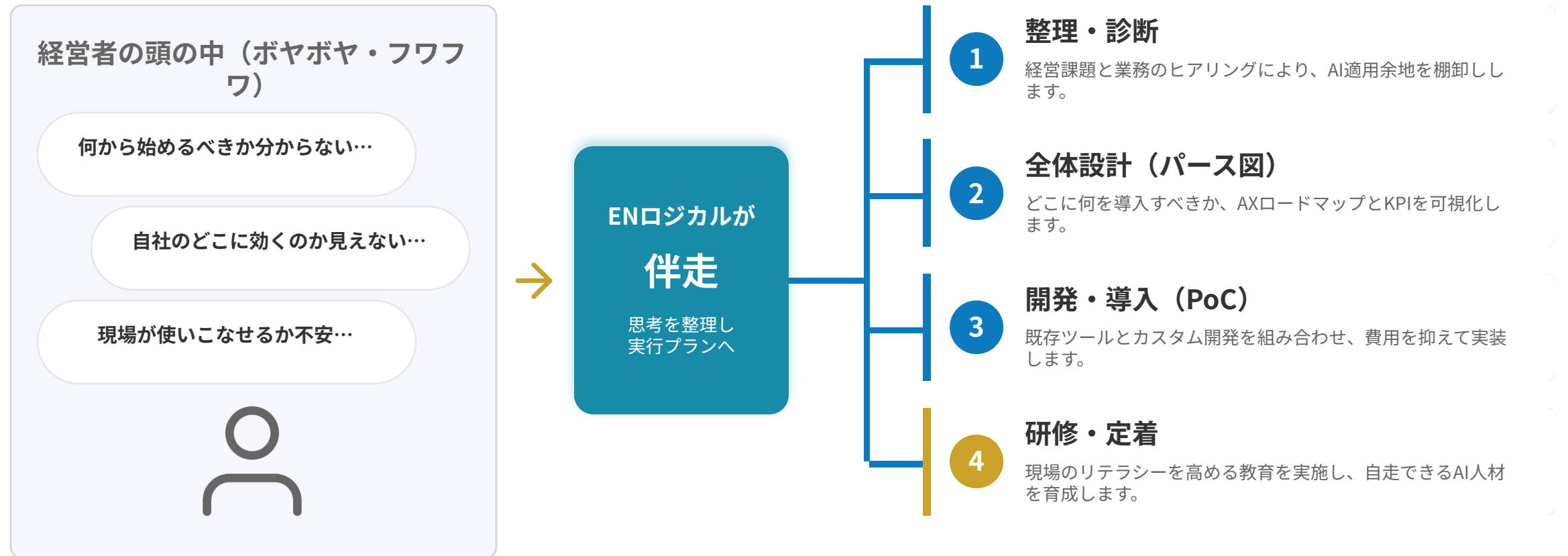
✓ 受講者の**実データ・実フォーマット**を題材に使う
✓ 受講後30日の**行動変容KPI**を講座設計時に定義

✓ 演習は**自分の業務課題**を持ち込む形式
✓ **フォローアップ会**と現場再支援をセット

補足：ENロジカル制作の講座は事前ヒアリング → 業務題材化 → 演習 → 30日後レビューを標準フローとする

経営者の「AIを活用したい」に、ENロジカルが丸ごと伴走します

課題の可視化から仕組み化・内製化まで、一気通貫で任せられる導入プロセス。



曖昧な構想を、実装可能な打ち手へ。構想整理から現場定着まで丸ごと支援します。

ご清聴いただきまして、
ありがとうございました。

