AIエージェント構築編

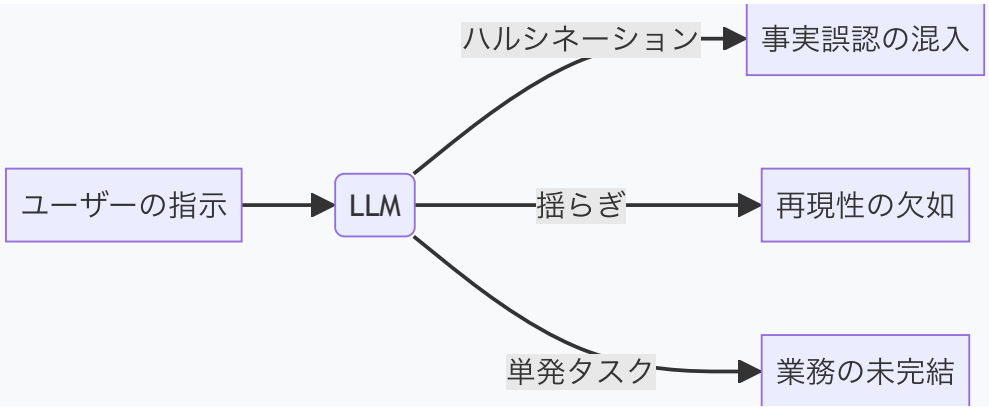
導入提案資料

～「言葉」で仕様を綴り、AIを確実な業務資産へ変える設計技法～

制作・監修・提供: 株式会社dbE, Director, Kazuki Arita

コンセプト: プログラミングコードは書かない。ロジックでAIを制御する。

現在の課題（なぜAI導入は現場で止まるのか）

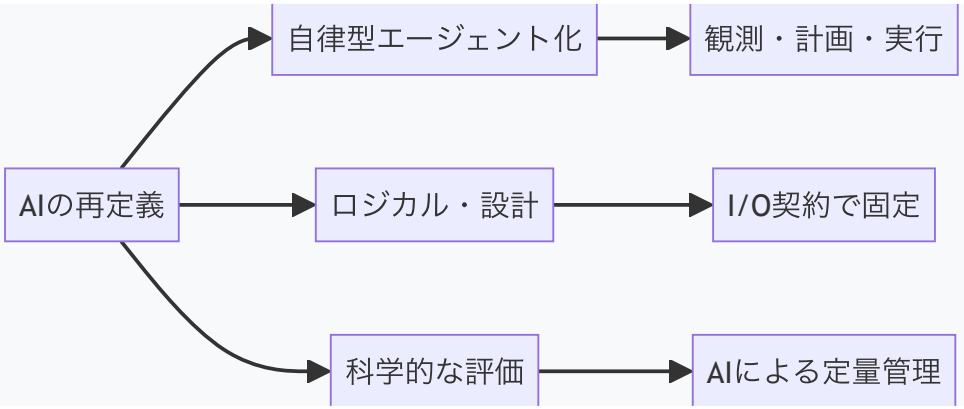


「なんとなく動く」の限界が、実務投入を阻んでいる

- 精度の不安定さ: 確率的に言葉を繋ぐ性質上、事実誤認（ハルシネーション）が「仕様」として混入する。
- 制御不能な揺れ: Temperature（温度）などの設定が適切でないと、回答が毎回変わり再現性が保てない。
- 業務の未完結: チャット形式の質疑応答に留まり、後続の「判断・実行・記録」という業務フローに繋がらない。

> ✅ **ポイント：チャットAIは便利ですが、そのままでは会社のシステムとしては不安定です。明確なルール作りが必要です！**

解決策（本講座のコア・コンセプト）

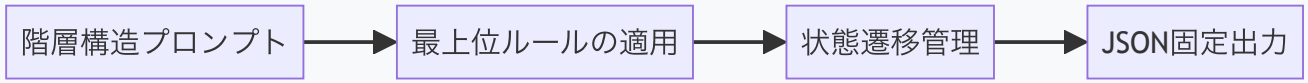


AIを「万能の神」ではなく「有能だが確認が必要な部下」として再定義する

- 自律型エージェント化: 周囲を「観測」し、「計画」を立てて「実行」する5段階の行動ループを実装する。
- ロジカル・エンジニアリング: プログラミングの代わりに、論理的な「I/O契約（入出力の約束）」を用いて挙動を固定する。
- 科学的な評価: 人間の主観を排除し、「AI as a Judge」によって品質を定量的に管理する。

> ✅ **ポイント：AIを魔法のように扱うのではなく、優秀な新人スタッフを指導するように論理的な仕組みを作りましょう。**

特長1：プログラミング不要の「仕様固定」技術

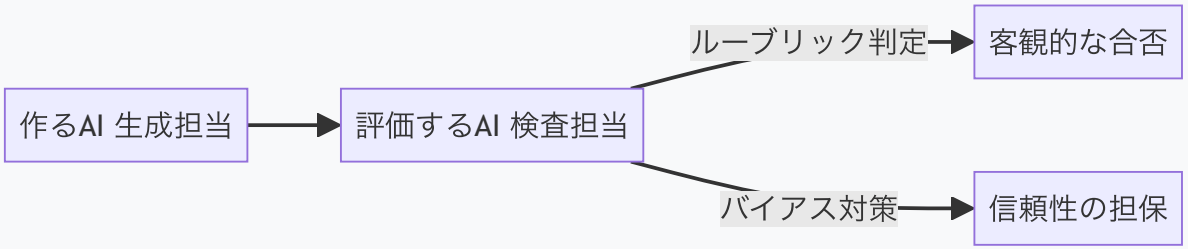


「契約」と「状態」でAIをシステムの一部として制御する

- 9つの必須キー（JSON形式）: 出力を厳格なデータ形式に固定し、システム連携を容易にする。
- 状態遷移（State）管理: 「情報不足」や「承認待ち」など、AIが次に進むべき出口を明確に定義する。
- 階層構造プロンプト: システムプロンプトを「最上位ルール」とし、計算の迷いを断ち切るレールを敷く。

> ✅ **ポイント：プログラミング言語が分からなくても、日本語でしっかり「約束事」を書けば堅牢なシステムは作れます。**

特長2：AI as a Judge による品質保証

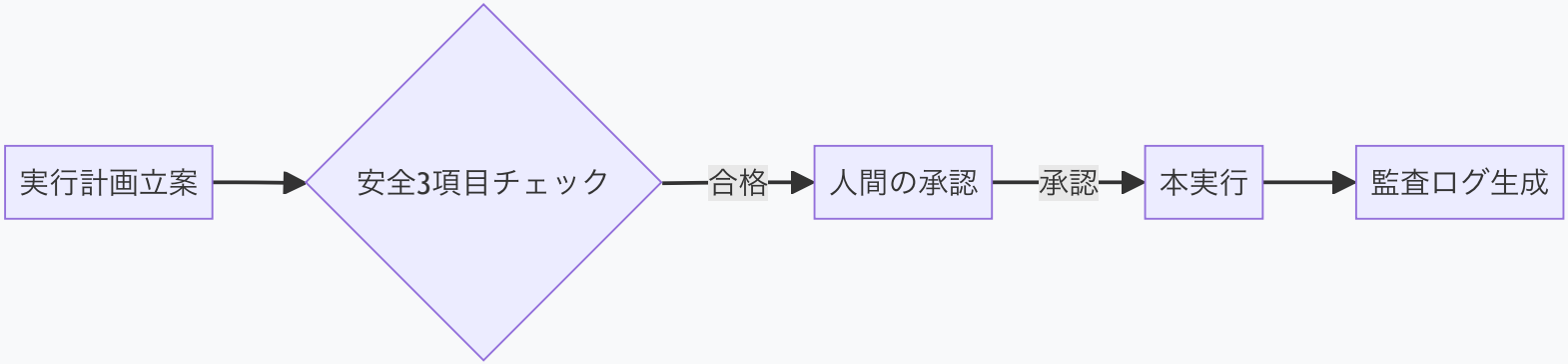


感覚に頼らない、客観的な精度向上サイクル（SOP）の確立

- 役割の物理的分離: 「作るAI」と「評価するAI」を分け、自己評価トラップを回避する。
- ブレないルーブリック設計: 0/1（合格/不合格）や具体的なYes/No条件で構成された判定基準を構築する。
- バイアス対策: 順序バイアスを排除するABテストや多数決評価により、判定の信頼性を担保する。

> ✅ **ポイント：人間が一つずつ目視チェックするのは大変です。別のAIを「検査官」に任命して自動で採点させましょう。**

特長3：実務を守る「安全設計」と「証跡」



事故を未然に防ぎ、透明性の高いAI運用を実現する

- 二段階コミット: 実行前に「計画」を提示させ、人間が承認して初めて「実行」される仕組みを徹底する。
- 安全3項目の自動チェック: 送信禁止、推測禁止、個人情報保護を常に検証する。
- 監査ログの自動生成: AIの思考根拠（reasons）や参照した証拠（evidence）を記録し、後からの監査を可能にする。

> ✅ **ポイント：AIが勝手に暴走しないよう、最終的な決定ボタンを押す権利は必ず人間が持つように設計します。**

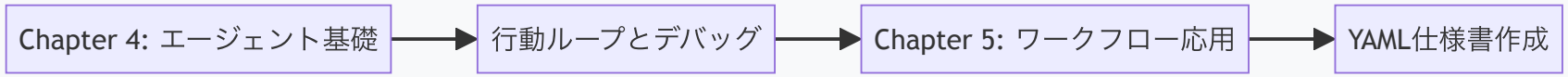
カリキュラム（基礎編：NLPとプロンプト）

技術の本質を理解し、言語を「精密な部品」として扱う

- Chapter 1（自然言語処理）：意味の座標（Embedding）やTransformerの革命を学び、言葉を処理する仕組みを把握する。
- Chapter 2（プロンプト実験室）：モデルの勝率比較や、Temperatureによる揺れの制御、構造化出力（JSON/YAML）を習得する。

> ✅ **ポイント：まずはAIがどうやって言葉を理解しているのか、その基礎と最新トレンドをしっかり押さえます。**

カリキュラム（実践編：構築と応用）



実務ワークフローを「設計図（YAML）」に落とし込む

- Chapter 4（エージェント基礎）：行動ループの設計と、症状・原因・対処のステップによる高度なデバッグ技術を習得する。
- Chapter 5（ワークフロー応用）：5つのワークフローパターンを活用し、機械が実行可能なYAML形式で「動く仕様」を完成させる。

> ✅ **ポイント：自社の実際の業務フローを、AIが読める設計図に翻訳していく実践的なステップです。**

不具合注入による実戦デバッグ訓練

「壊れたものを直す」経験が、運用現場での即戦力を生む

- 意図的な不備の修正: 分岐漏れ、安全チェックの欠落、契約違反などの不具合を自ら特定・修正するサイクルを身につける。
- 改善成果の定量化: AI as a Judgeを用いて、修正前後の改善率（Refinement Delta）を算出する手法を学ぶ。

> ✅ **ポイント：避難訓練と同じで、あえてエラーを起こすことで本番でのトラブル対応力が劇的に上がります。**

受講概要・コスト・助成金

プロフェッショナルなスキルを、最適なコストで導入する

- 受講時間: 約12時間（集中講義および演習含む）。
- 受講費用: 受講者1名あたり 20万円（税別）。
- 助成金活用: 「リスキリング助成金（人材開発支援助成金等）」の申請が可能であり、実質的なコスト負担を軽減できる可能性があります。
- 制作・監修・提供: 株式会社dbE, Director, Kazuki Arita

> ✅ **ポイント：国の支援制度をうまく活用して、チーム全体で最新スキルを獲得しましょう！**