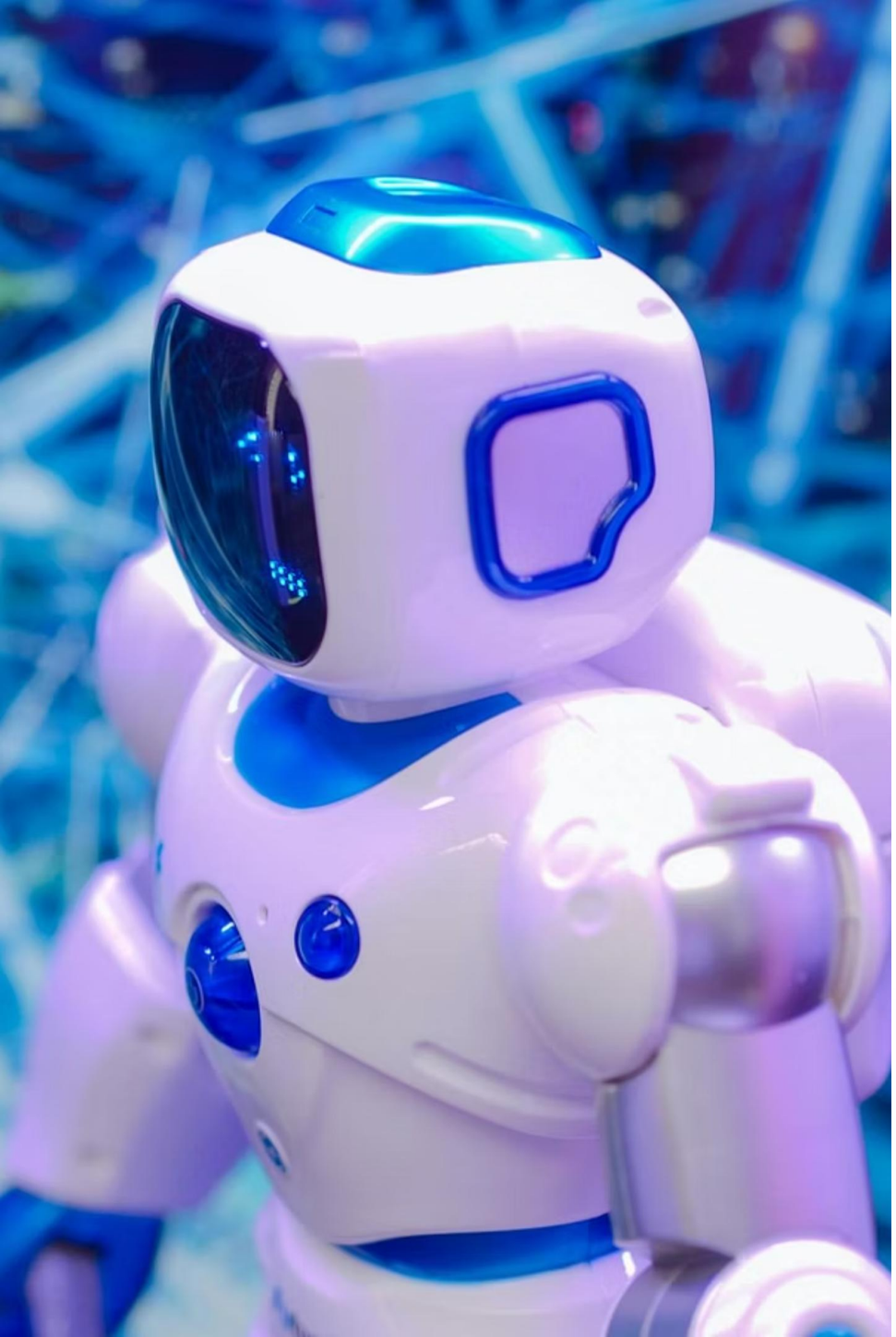


Agentic AI 勉強会

エージェントの仕組みと活用

第1回（全7回）｜2026/04/16

0. INTRODUCTION

勉強会全体について

UMP-JUST 主催ハッカソン

『AI エージェントの社会実装』
2026年6月27日（土）・28日（日）開催

課題の制約なし。

AI エージェントの自律的問題解決能力を用いて、日常・日本・世界の課題を自由な発想で設定し、ツールによる解決を進める。

本勉強会の目的

→ 段階的な理解

社会課題とAgentic AIへの理解を深める

→ ハッカソン接続

参加者がスムーズに参加できるテーマの種を創出する

→ 責任あるAI

倫理・ガバナンス・責任の所在に真正面から向き合う

0. INTRODUCTION

スケジュール（全7回 17:00–18:30）

（第1回） 4月16日(木) （第2回） 4月23日(木)	Agentic AIへの理解、社会課題との関わり
（第3回） 5月14日(木) （第4回） 5月21日(木)	Amazon Bedrock エージェントの紹介
（第5回） 6月4日(木) （第6回） 6月11日(木) （第7回） 6月18日(木)	ハッカソン前段としてのアイデアソン （グループディスカッション形式を予定）

スピーカー紹介

田中 剛範（たなか たかのり）

株式会社マーズフラッグ｜サービスプラットフォーム部

担当

SRE｜業務で日々**AWS**と格闘中

前歴

科学研究（バイオインフォマティクス）
東大にも**2**年在籍

出身・趣味

北海道出身｜食べること・料理

MARS FLAG

BEYOND EMOTION — サーチを技術から感動へ

日本発、国内トップシェアのサーチプラットフォーマー

世界初のページキャプチャ付き検索エンジンや、当社特許技術を活用したサービスの開発など、
サーチテクノロジーによって常に業界の先頭を切り拓いてきました。

国内シェア No.1*

*株式会社富士キメラ総研「ソフトウェアビジネス新市場 2025年版」
検索エンジン市場(SaaS/PaaS) 2024年度 ベンダーシェア

MARS FINDER

サイト内検索に連携するチャット型AI対話サービス

MARS EmoTech

1. オープニング

今日のゴール

🎯 理解する

Agentic AI の全体像をつかみ、普通の生成AIとの違いを説明できる

🔍 見極める

何が得意で何が難しいかを知り、RAGとMCPの役割を区別できる

💡 考える

実際の業務に置き換えて考え、ハッカソンでの題材化につなげる

「どこに使うと価値が出るか」を判断できる状態を目指す

1. オープニング

アジェンダ

前半：基礎理解

- 生成AIの基本
- LLM（大規模言語モデル）
- RAG（検索拡張生成）
- MCP（モデルコンテキストプロトコル）

後半：応用・実践

- Agentic AIの考え方
- エージェントの動き方
- ユースケース・仕組み・リスク
- AIと社会実装・ハッカソン接続

2. 基本おさらい

生成AIとは何か

生成AIの定義

文章・画像・音声・コードなどを「生成」するAI。
従来AIが「分類・予測」中心だったのに対し、より表現の幅が広くたたき台づくりに強い。

押さえたい前提

- もっともらしい出力を作るのが得意
- 常に正しいわけではない
- 便利だが検証は必要

⚠ 生成AI = 何でも正しいAI ではなく、「新しく作ること」が得意なAI

2. 基本おさらい

LLMとは何か

LLM = Large Language Model（大規模言語モデル）

ひとことで言うと

大量の文章を学習した言語の土台モデル。次に自然な言葉を「予測」する仕組みで、言語を使う多くの仕事に応用できる。

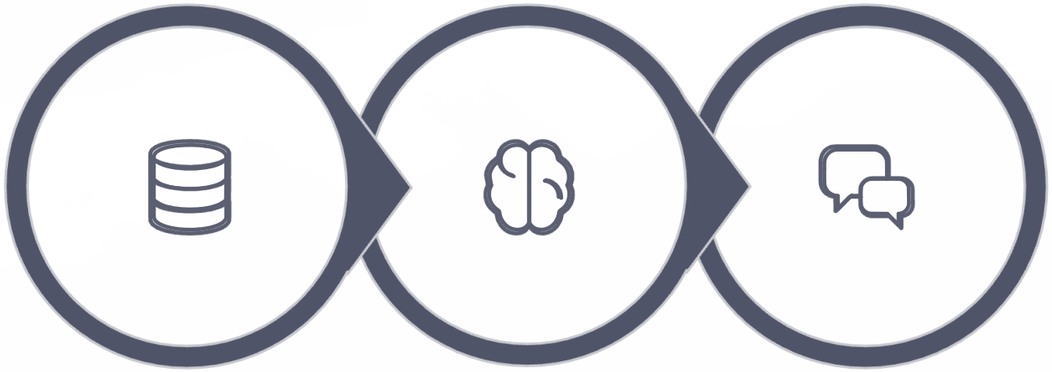
できること

- 要約・翻訳・分類
- 質問応答・言い換え
- コード生成・説明

1つの土台モデルが、さまざまな言語タスクに応用できる

2. 基本おさらい

LLMはなぜ「賢く」見えるのか



大量の
文章データ

パターン学習

自然な出力

事実、説明、議論、手順、例え話、
プログラムコード、FAQ、etc.

過信しないための視点

学習データの中に「知の形」「答え方のパターン」が含まれている

「賢く見える」と「本当に理解している」は別物

最新情報や厳密性は別途確認が必要

2. 基本おさらい

LLMの得意なことと苦手なこと

✅ 得意なこと

- 長文の要約と再構成
- 説明の言い換え・比較・分類
- たたき台作成

❌ 苦手なこと

- 最新情報の保証
- 厳密な数値や条件の処理
- 知らないことをもっともらしく話す（ハルシネーション）

⚠ 言葉を扱う力は非常に強いが、事実保証の装置ではない

2. 基本おさらい

従来の業務システムとLLMの違い

🏢 従来の業務システム

Rule-based

- 決められたルールに沿って動く
- 条件分岐や定型処理が得意
- 同じ入力→同じ結果（再現性が高い）

🤖 LLM

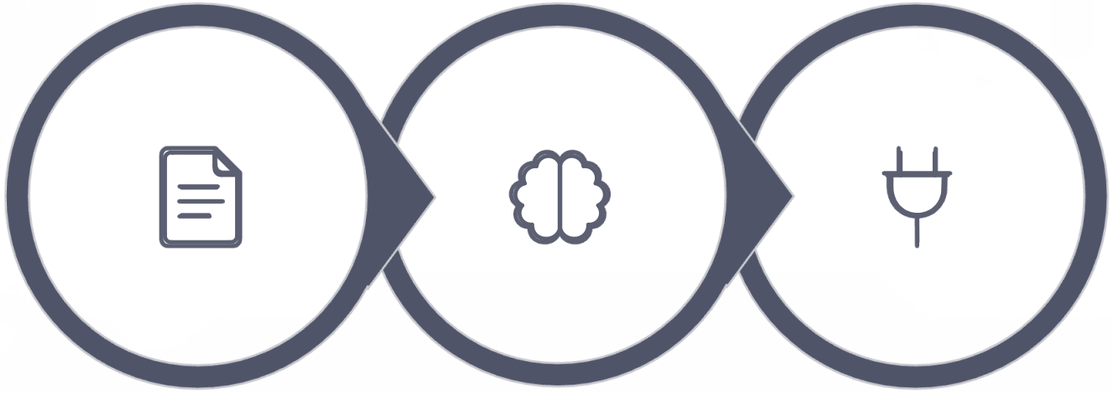
Pattern-based

- 大量データから言葉のパターンを学習
- 曖昧な問いにも柔軟に回答
- 出力は確率的、誤りも起こる

✅ 厳密処理は従来システム、言語理解はLLM。組み合わせると強い。

2. 基本おさらい

Agentic AIにおけるLLMの役割



理解　　判断　　連携

LLMは「考える・判断する・指示を理解する」力を担う。しかしLLM単体では検索・ファイル参照・送信などの行動はできない。そこでRAGやMCPのような仕組みが必要になる。

LLM が「頭脳」、RAG が「参照」、MCP が「接続」を支える

2. 基本おさらい

RAG（検索拡張生成）

RAG = Retrieval-Augmented Generation

RAGの役割

→ 必要な情報を外部に取りに行く

→ その情報を踏まえた回答を生成する

→ 「知っていることだけで答える」を補う

RAGが効く場面

- 最新情報を扱いたいとき
- 社内文書・独自データを参照したいとき
- 回答の根拠を示したいとき
- ハルシネーションを低減したいとき

RAG = 「調べてから答える」仕組み

2. 基本おさらい

RAGの流れと具体例

検索　　抽出　　生成

例：社員が「出張申請のルールを教えて」と質問
→ 就業規程や申請マニュアルを検索 → 関連部分を参照 → 自社ルールに沿って回答

2. 基本おさらい

MCP（Model Context Protocol）

ひとことで言うと

AIが外部ツールやデータとやり取りするための接続ルール。LLMの外側にあるツール（Web検索・社内ドキュメント・カレンダー・データベース・メール等）への橋渡し。安全で整理された連携を考える枠組み。

プロトコルとは

異なるコンピュータ・機器同士がデータを送受信するための共通ルール（例：IP、HTTP、FTPなど）

イメージ

🧠 LLM

頭脳

🛠️ ツール

手や目

🔗 MCP

その接続方法

RAGは「調べて答える」、MCPは「つながって使う」

2. 基本おさらい

なぜMCPが必要なのか

背景

LLM単体は文章生成はできるが、外部情報の取得や実際の操作は単体ではできない。
外部機能にはAPIでアクセスするが、APIがあるだけではAIが安全かつ適切に扱えるとは限らない。
MCPは、APIやツールをAIが理解しやすい形に整理する考え方。

利用ケース例

- 今日の天気を調べる
- 社内の最新資料を探す
- 在庫データを確認する
- 予定表を見る
- メールを送る

MCPは、Agentic AIが外部とつながって安定して行動するための接続基盤

2. 基本おさらい

MCPとAPI連携の違い

🔌 API連携

サービスの機能を外部から使えるようにする仕組み。ただしAPIがあるだけではAIが使いやすいとは限らない。

🗂️ MCP

AIがツールを扱いやすくするための枠組み・共通化の考え方。何ができるか・どう使うか・何が返るかをAIに伝えやすく整理する。

APIは個別の道具、MCPはそれをAIが使いやすくそろえる考え方

2. 基本おさらい

MCPのメリットと注意点

メリット

- ツール接続を整理しやすい
- 新しいツールやデータを追加しやすい
- Agentic AI の拡張性を高めやすい

注意点

- 外部接続によりリスクも増える
- 参照範囲や実行権限の設計が必要
- 人の確認をどこで入れるかも重要

⚠ MCPは、AIの道具箱を増やしやすくする。ただし、安全に接続する設計が前提になる。

2. 基本おさらい

ここまでの整理

🧠 LLM：頭脳

言葉を理解し、考えて答える中核。文章生成・要約・質問応答などを担う。

📚 RAG：参照

外部情報を検索・参照してから回答する仕組み。最新情報や社内文書を踏まえた回答がしやすくなる。

🔗 MCP：接続

外部ツールやデータとつながるための接続基盤。AIが情報取得やツール実行を行いやすくする。

✅ **LLM**が考え、**RAG**が必要な情報を補い、**MCP**が外部のツールやデータとの連携を支える。

3. AGENTIC AI

Agentic AIとは何か

目的に向かって、自分で途中の手順を考えながら進めるAI

💬 普通の生成AI：優秀な相談相手

- 聞かれたことに答える
- 1回ごとのやり取りが中心

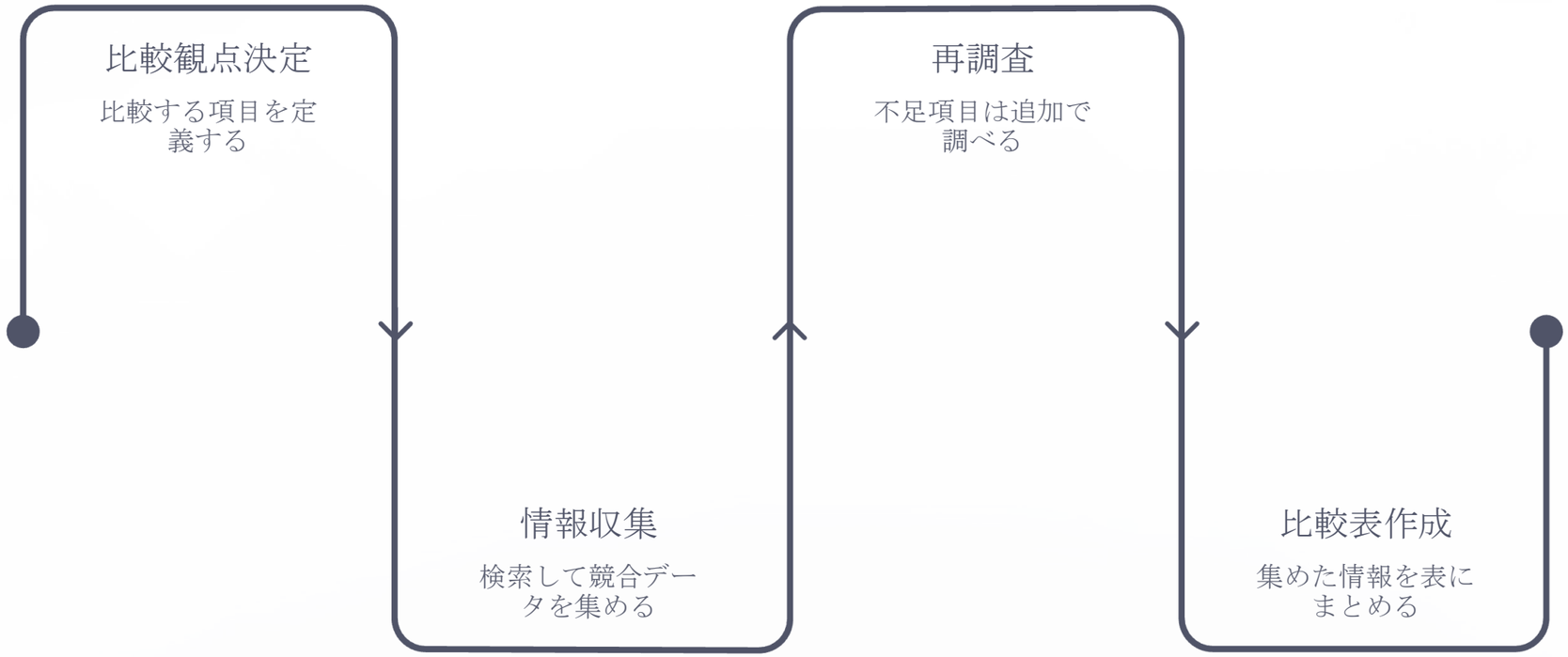
🚀 Agentic AI：自走する作業者

- ゴール達成のために複数ステップをつなぐ
- 途中結果を見て方針を変える

4. 動き方

具体例で見る：競合調査

依頼：「競合3社の機能と価格を比較して」



このように、Agentic AI は一度の回答で終わらず「集める・考える・再調査する・まとめる」というループを自律的に回す。これが通常の生成AIとの最大の違いだ。

3. AGENTIC AI

なぜAgentic AIが注目されているのか

作業の変化

AI活用が「回答生成」から「作業実行」に広がってきた

答えるAI
質問に回答する

作業するAI
複数工程をまとめて実行

注目される理由

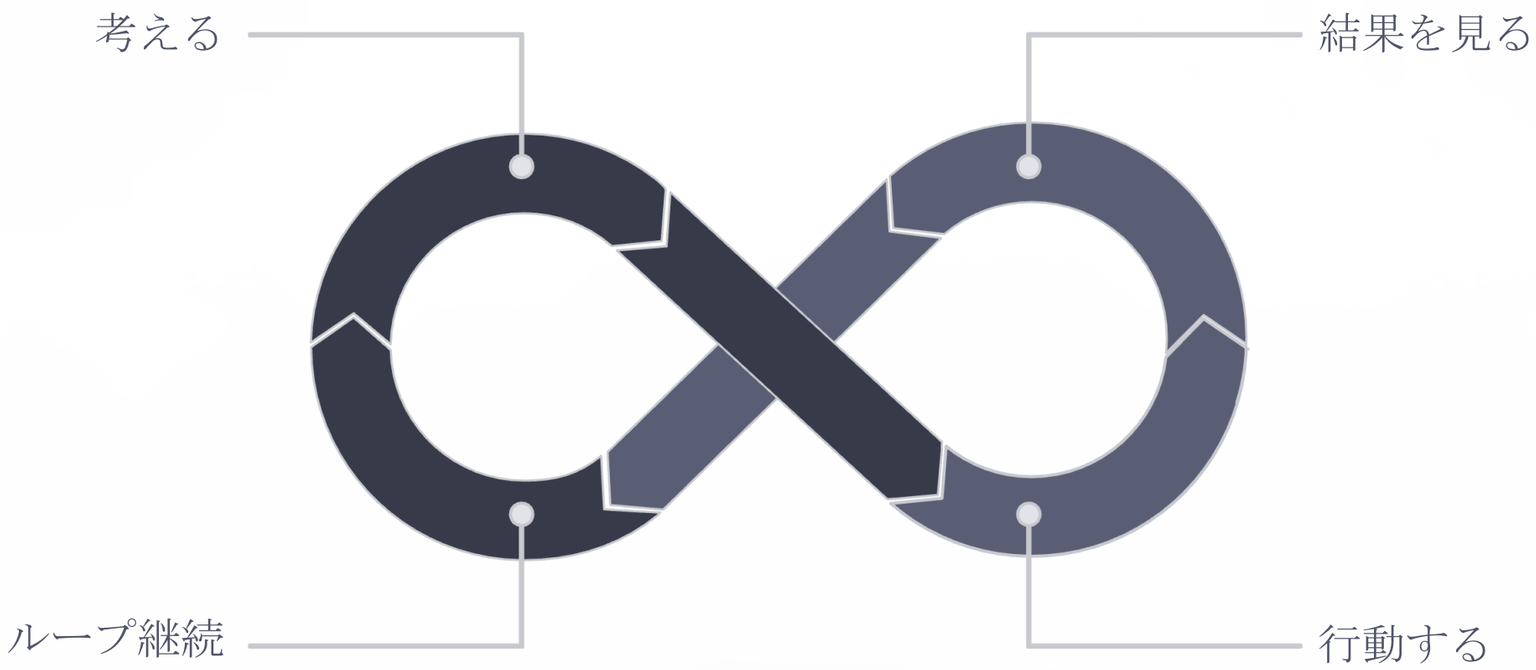
実際の仕事は「調べる→比較する→修正する→確認する」の連続とループ。これらの全工程をまとめて支援できる可能性があり、業務活用の効果が見えやすい。

ただし

何でも自律的に完璧にやってくれるものではない。
目的設定・権限設定・途中確認・最終判断には人の関与が重要

4. 動き方

エージェントはどう動くのか



このループがエージェントらしさの本質だ。一回答えて終わりではなく、目的に向かって自律的にサイクルを回し続ける。

「考える → 行動する → 結果を見る」のループがエージェントらしさ

4. 動き方

どこまで自律させるか

🟢 Assist（低い自律性）

- 提案だけする
- 人が毎回確認する

🟡 Semi-auto（中くらい）

- 情報収集は自動
- 重要操作は人が承認

🔴 Auto（高い自律性）

- 限定範囲で自動実行
- 強い制約と監視が必要

⚠ 実運用で大切なのは「全部自動」より、境界設計

6. ユースケース

何ができるのか：向いている仕事の特徴

向いている仕事

- 複数ステップがある
- 一定のパターンがある
- 途中で情報取得や整理が必要

→ 人の判断の前段を減らせる

向いていない仕事

- 曖昧で正解が定義しにくいもの
- 社会的責任が重く最終判断を人が持つべきもの

考え方のコツ

職種で見るのではなく**「作業単位」**で切る

例：「営業全体」ではなく**「提案前の企業調査」**

6. ユースケース

ユースケース概観

調査の自動化

情報収集、比較、要点整理

コード生成・修正

エラー調査、修正案生成、関連ファイル参照

業務タスクの自動化

メール分類、問い合わせ下書き、定型レポート作成

すべて「集める・考える・まとめる・必要ならやり直す」という流れを含む → Agentic AI との相性がよい

6. ユースケース

ユースケース1：調査の自動化

典型タスク

- 競合3社の機能と価格を調べて比較する
- あるテーマの最新動向を複数ソースから整理する
- 社内外の情報を集めて会議前メモを作る

なぜ相性がよいか

→ 検索→読む→比較→まとめるの流れがある

→ 抜けを見つけたら追加調査できる

→ 前工程の時間を大きく削減できる

「まず広く集めて、その後で絞る」作業はAgentic AIと相性がよい

6. ユースケース

ユースケース2：コード生成・修正

できること

- エラー原因候補の整理
- テストコードのたたき台
- 既存コードの説明
- 軽微なリファクタリング

タスクの要素

関連ファイルを見る → エラーを読む → 原因候補を考える → 修正案を試す → 再修正する → …

注意点

動くように見えて本質的に誤ることがある

関連ファイルや実行結果まで見て判断したい

人間のレビューは必須

価値は「置き換え」より「調べる・試す時間を短くする」こと

6. ユースケース

ユースケース3：業務タスクの自動化

📬 問い合わせ対応

- 分類する
- 一次回答案を作る

🗓️ 会議まわり

- 事前情報を集める
- 議事録を要約する

🗂️ 定例業務

- レポート下書き
- 社内申請の素案作成

✅ 「全部自動化」ではなく、「判断の前段を減らす」と考えると適用範囲が広がる

6. ユースケース

仕事のどこに組み込むと効果が出やすいか

効果が出やすい領域

- 最初の手間が大きい部分
- 定型的に繰り返す作業

例）情報収集 / 会議準備 / 文書のたたき台 / 定例整理

向いていない部分

- 利害調整
- 微妙な交渉
- 最終責任判断

準備や整理はAI、最終判断は人

6. ユースケース

AIに新しい価値の創造は期待できる？

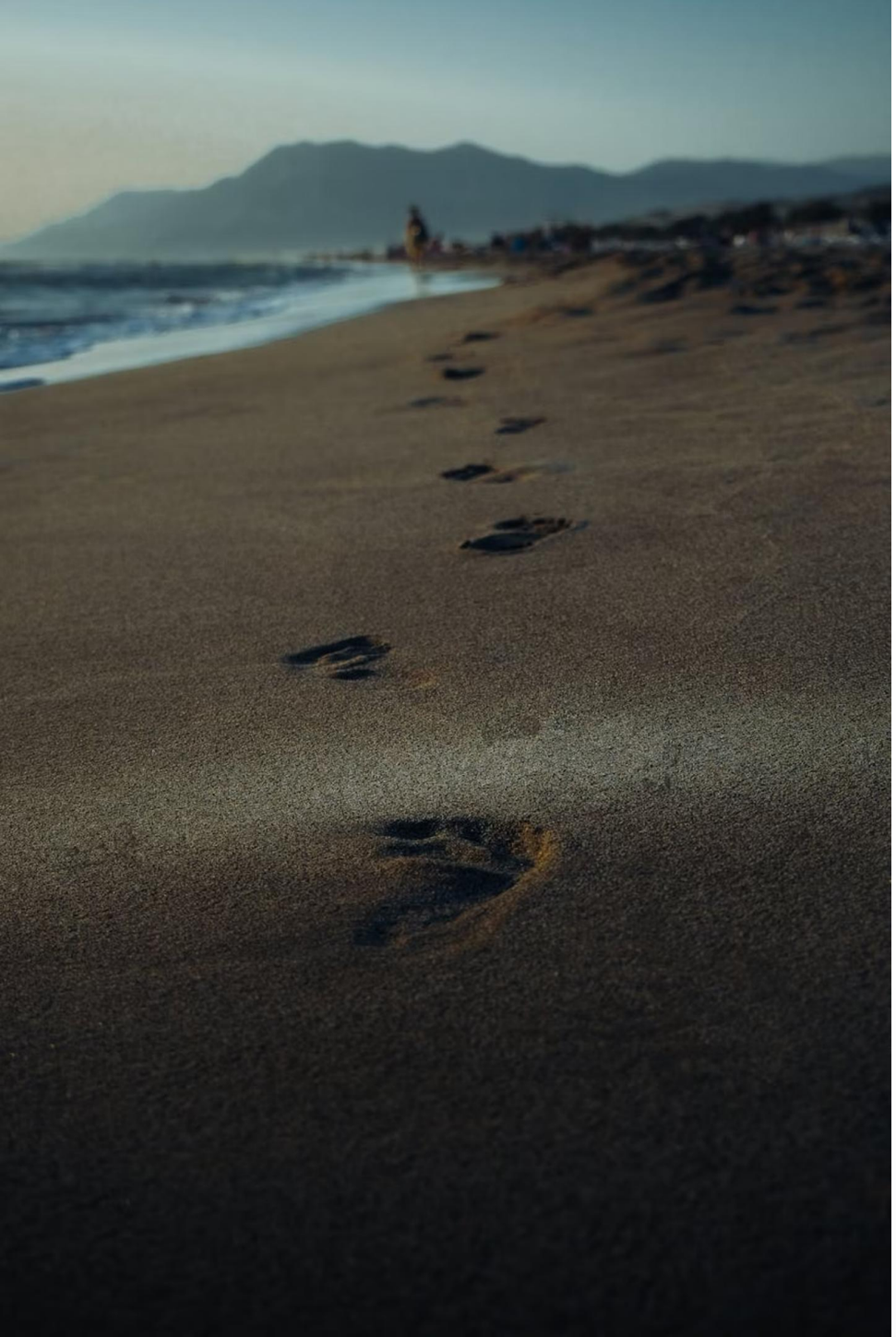
創造性への期待

ゼロから独創を生むというより、創造を支援・加速する存在。
大量の情報から新しい組み合わせを提案し、仮説を多く出し、
試作速度を上げる。

活用例

- 類似事例の収集
- 別業界パターンの参照
- 切り口の拡張
- ユーザー反応の仮説出し

AIは「発想の代行者」より「発想の加速装置」と捉えると使いやすい



6. ユースケース

小さな始め方

✅ 最初に満たしたい条件

- 失敗しても致命傷になりにくい
- 効果が見えやすい
- 人が最終確認できる

🚀 始めやすいテーマ例

- 調査メモの下書き
- 定例資料の素案
- FAQの一次案
- 議事録の要約

🤝 人間との役割分担

AI：　情報収集・候補出し・整理・たたき台
人間：責任判断・倫理判断・組織調整・最終承認

まずは**1**つの面倒を減らすところから始める

7. 支える仕組み

エージェントを支える仕組み

Tool Use：ツールを使う

検索・API・業務システムなど外部ツールを呼び出してタスクを実行する

Memory：記憶する

途中経過を保持し、前回の続きから作業を再開できる

Agent Design：役割分担

シングルエージェントか、複数エージェントかを設計する

7. 支える仕組み

シングルエージェントと複数エージェント

🤖 シングルエージェント

- 1つのAIが全体を見て進める
- シンプルでわかりやすい
- 最初の設計として向いている

🤖🤖 複数エージェント

- 調査担当・要約担当などに分ける
- 役割が明確だと整理しやすい
- 連携が複雑になる

⚠ 最初から複雑にせず、設計上の必要性が出てきてから分けるのが現実的。技術的にできるから分けるのではない。

8. 注意点・リスク

注意点・リスクの全体像

⚠️ Risk 1：間違った判断

- 情報不足を補ってしまう
- 古い情報を使う
- 曖昧な指示を誤解する
- 誤りが次の行動に連鎖しやすい

🔄 Risk 2：暴走・無限ループ

- 目的・終了条件が曖昧だと同じ処理を繰り返す
- ステップ数・時間・コスト制限と停止条件が重要

🔒 Risk 3：セキュリティ

- 外部アクセスが増えるほどリスクも増える
- 閲覧範囲・実行権限・外部送信・機密情報の扱いを設計する必要がある

8. 注意点・リスク

セキュリティ上の危険

- 見てはいけない情報を見る
- やってはいけない操作をする
- 外部からの悪意ある指示に引っ張られる
- AIの出力を人が信じすぎてしまう

⊗ 不正アクセスだけではなく、"権限の持たせ方“ "接続先の信頼性“ "人間の確認不足"まで含めて考える

8. 注意点・リスク

セキュリティの具体例

🔍 情報を見すぎる

必要以上に広い閲覧権限、人事・契約情報まで参照

対策：最小権限の原則、分類、ログ取得

⚙️ 勝手に操作する

顧客へ未確認の内容を送る、更新系操作を誤実行

対策：read-write権限分離、Human-in-the-loop

🎣 悪意ある誘導

プロンプトインジェクション（例：「それまでの指示を無視して…」）

対策：信頼できるデータソースを優先、許可操作を厳格に分離

📤 機密情報の外部送信

機密を外部APIへ送る、返信文に内部情報が混ざる

対策：機密レベルの明確化、人による内容チェック

8. 注意点・リスク

セキュリティ対策の基本原則

01

最小権限

必要な情報だけを見せ、必要な操作だけを許可する

02

段階導入

最初から自動送信・自動更新まで行わず、安全な範囲から始める

03

人間確認

重要な操作は必ず人が確認する

04

ログと監査

何を見て、何を判断し、何をしようとしたかを追えるようにする

05

接続先の管理

どのツールにつなぐか、どのデータソースを信頼するかを明確にする

出発点は「高性能なAIを入れること」ではなく、「危険なことをできないように設計すること」

10. 社会実装・まとめ

AIと社会実装

社会実装は「これから起こる話」ではなく、すでに始まっている。まずは限定された業務から小さく導入されることが多く、完全代替ではなく一部工程の置き換えとして進む。

顧客対応

一次対応の自動化と人へのエスカレーション

社内業務支援

文書検索・申請案内・IT一次対応

開発支援

コード補助・テスト生成・影響範囲の洗い出し

10. 社会実装・まとめ

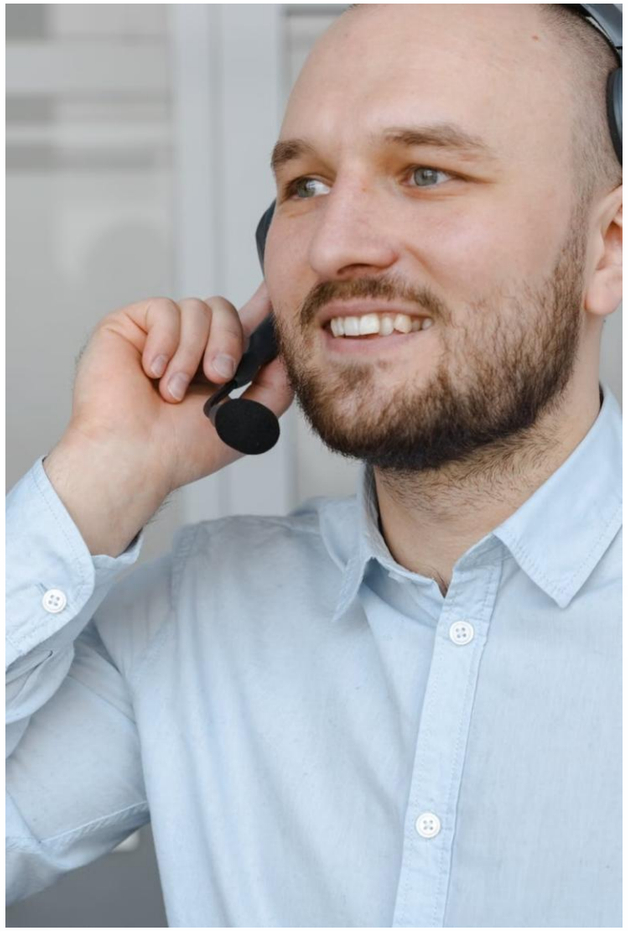
実用化が進みやすい領域1：顧客対応

典型例

- 一次対応をAIが行う
- 難しい案件だけ人につなぐ
- オペレーターに回答候補を出す

価値

- 待ち時間の削減
- 対応品質の平準化
- 担当者負荷の軽減



「全部自動」より、「前段の処理をAIが引き受ける」設計が現実的

10. 社会実装・まとめ

実用化が進みやすい領域２：社内業務支援

典型例

- 就業規則や申請フローの案内
- 社内文書検索
- ITサポートの一次対応
- 新人教育の補助

価値

- 問い合わせ集中の緩和
- 担当者の繰り返し説明を減らす

社内向けは制御しやすく、導入の最初の一歩として選ばれやすい

10. 社会実装・まとめ

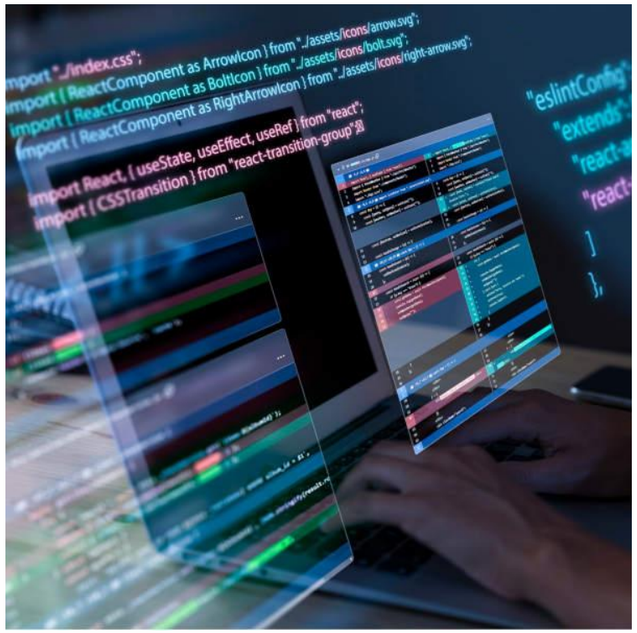
実用化が進みやすい領域３：開発支援

支援内容

- コードのたたき台作成
- テスト生成
- 変更影響の洗い出し
- 既存コードの説明

価値

- 調べる時間の短縮
- 試作速度の向上
- 未知のコードベース理解を助ける



価値の中心は「開発者の置き換え」ではなく「理解・試作・調査の高速化」

10. 社会実装・まとめ

社会実装が進むと、仕事はどう変わるか

🤖 AIが担う部分

- 情報収集
- 定型処理
- 素案作成

👤 人が集中する部分

- 論点設定
- 評価と判断
- 責任ある意思決定

📈 これから求められる力

- AIに仕事を切り出す力
- 結果を見極める力
- 人間同士の調整力

10. 社会実装・まとめ

まとめ、そしてハッカソンへ

Agentic AIの本質

目的に向かって考え、ツールを使い、途中で見直しながら進むAI

価値が出る場所を見つける

「何を作るか」より「どの仕事のどの部分に入れると価値が出るか」を考える

注目してほしい作業の特徴

繰り返しが多い / 情報収集に時間がかかる / たたき台があると助かる

便利さと安全さをセットで

確認のしかたや安全な使い方まで含めての設計が重要

> この仕事はエージェント化できるか。そのとき人はどこに残るべきか。