

すべての新入社員を AI 人材へ

後編では、研修プログラムの内容を詳細に紹介していく。短期集中の垂直離陸型研修は、4つの講義と自習で構成されている（図表 1）。前半の 2 つが主に座学であり、その後の Python 事前自習期間をはさんで、後半 2 つが開発環境を用意してのハンズオンセッションとなっている。研修講義に用いたマテリアルと講義ビデオは、イントラネットの SBI AI Digital Portal で SBI グループ内に公開されており、研修対象者以外にも参照可能となっている（図表 2）。

図表 1 生成 AI 研修の構成

第 1 講	生成 AI 基礎概論
事後課題	第 1 講の復習と深掘りのための課題
第 2 講	クラウドサービスの基礎
事前学習	次講準備：Python のローカル環境セットアップと基本コーディング
第 3 講	API で LLM を利用：仮想環境での業務ユースケースのハンズオン
事前学習	次講準備：UNIX 基本コマンド、関数、クラス、メソッドの理解
第 4 講	マルチエージェント AI 作成と Web アプリケーションへの実装

図表 2 ポータルサイトの案内ページ



第1講

第1講では、導入として生成 AI 概論の講義を行った。講師は、入社2年目の AI・デジタル戦略推進部のスタッフである。本講義では、座学だけに止まらず、その場で実践してみるハンズオンを組み合わせることで、1) AI 活用に対する心理的ハードルを引き下げ、2) 各参加者の実務の場での再現可能性を高めるとともに、3) 講義中の集中力維持や脱落防止に配慮した構成としている。

まず、広義 AI の歴史的背景から最新の技術動向までを概観し、学びの動機付けのため、AI が現代のビジネス環境にどのような革命的变化をもたらしているかを解説した。

次に、生成 AI アプリケーションの構成要素として、Web アプリケーションの仕組み、LLM（大規模言語モデル）、API、クラウド、認証認可などの基本概念を説明した。AI、生成 AI、機械学習、深層学習、LLM などのコンセプトも混乱しやすいので整理を行っている。

生成 AI にできること（何を生成するか）を網羅的に紹介し、その具体例として、一般に提供されているサービスの代表的事例を紹介した。AI 検索エンジン（Perplexity、Genspark）や、汎用型 RAG としても利用可能な資料読解支援アプリ（NotebookLM）、画像や動画、音楽、漫画の生成サービス、コーディング支援などである。金融ビジネスに関連する代表的な業務活用事例をあげ、新入社員が各々従事しているビジネスのなかにも多くの活用可能領域があることを発見してもらった。

次に、利用の注意点として、倫理面からの問題点（バイアス、公平性、不適切な情報拡散、責任境界など）と、信頼性（ハルシネーション、学習情報のリミットデイト、説明不可能性）、プライバシー、セキュリティ、知財・著作権などを解説している。こうした一般論に続いて、SBI グループ内で策定・運用されている生成 AI 利用ガイドラインの解説を行い、業務運営に活用する場合の注意すべき点を学んだ。

ハンズオンの導入として、AI との対話技術である「プロンプトエンジニアリング」を体験してもらった。具体的には、メール要約や返信草案作成、議事録作成など、新入社員がすぐに活用できそうな題材を取り上げている。業務で使える具体的なシナリオを通じて、AI から意図した通りの高品質なアウトプットを引き出すための「プロンプトの型」の習得を目標としている。

その際、単に指示を出すだけでなく、明確な役割設定、背景情報の付与、出力形式の指定、そして改善を繰り返す反復的アプローチを体験し、AI を駆使するための思考法を体験してもらった。セキュリティ上のリスクを認知してもらうためにプロンプトインジェクションによるパスワード奪取ゲームも行っている。

LLM が確率的言語生成モデルであることを解説した後、LLM の各種パラメータ

(Temperature や Top-P/K フィルター等) を様々に変えてのアウトプットの変化を観察することも体感してもらっている。

講義後には、知識の定着のために以下のような課題を課している。

- ・講義で十分には理解できなかった箇所を生成 AI で調べる。どのようなプロンプトを作成したか、その際、何を意識して設計したかを書き出して提出。
- ・SBI 生成 AI 利用ガイドラインを読み、「機密情報」に関する内容を 300 字程度でまとめる。
- ・生成 AI に以下の情報を出力させるプロンプトを作成し、提出する。答えそのものをプロンプトで与えるのは禁止にしている。出力情報：{'会社名': 'SBI ホールディングス', '本社最寄駅': '六本木一丁目', '創立年数': 26}

図表 3 研修の様様：第 1 講



第 2 講

生成 AI アプリケーション作成ではクラウドや仮想環境の利用が必須となるため、一般常識としてのクラウドサービス理解を含めて、実務家（SBI 生命の情報システム部）が講義を行った。受講者は生成 AI サービスの利用体験はあっても、パブリッククラウドを利用した経験があるものはごく一部であり、クラウドサービスというものが何であるか、世界がどれほどクラウド依存で動いているかを理解している者は少ない。

そこで、受講者がゼロ知識であることを前提に、1) クラウドサービスとは何か、2) クラウドと AI の関係性、3) SBI グループ企業でのクラウドの利用事例について講義を行った。1)

の内容は一般的なものであるため、ここでの紹介は省略する。2) では、多種多様な LLM がマネージドサービスとしてクラウドで提供されているだけでなく、顧客提供サービスを組み上げるに際してクラウドが高い優位性を持っていることが紹介された。仮想サーバ、データストレージ、データパイプライン、データ集計分析、各種の広義 AI 領域のサービス（機械学習モデル構築、画像認知、STT/TTS (Speech-to-Text/Text-to-Speech)、様々な自然言語処理）がパッケージとして用意されている点などである。

3) では、住信 SBI ネット銀行が AWS 上にネットバンキングインフラを構築している概要や、SBI 証券が国内株式取引システムのクラウド移行を行った事例が紹介された。SBI 新生銀行では生成 AI の活用事例として、Amazon Bedrock 上で Claude 3.5 Sonnet を用いて、法人営業の業務効率化や情報・データの可視化、情報格差の削減を実現していることなどが紹介された。SBI 証券の手数料ゼロ革命ほか様々なビジネス戦略をクラウド活用が支えていることを理解してもらった。また、SBI 生命では AWS を用いてコールセンターの AI 化を図った事例などが、クラウドのアーキテクチャ図や処理フロー図の解説とともに示された。

最後に、クラウド開発を進める場合のポイントや注意事項など実践的なアドバイスも提供された。コンペでの実装に際して大変有益となるアドバイスだが、この段階ではそれを知る由もなく、振り返りの学び直しで気が付くことになる。

第3～4講の事前学習

研修後半はコーディングのハンズオンに入るため、Python の基本を習得する必要がある。ノーコード開発ツールや AI によるコード生成が如何に発展しようとも、この基礎スキルが欠落しては開発や保守、アップデートはおぼつかない。

まず、自前の PC に Python が動く環境を作り、Jupyter notebook からローカル仮想環境として立ち上がった Python を Web ブラウザを通じた対話型実行環境として操作する練習を行ってもらった。具体的には、Python Japan にある[環境構築ガイド](#)を指示ノートにそって独習してもらった。次に、同サイトの「[ゼロからの Python 入門講座](#)」を最後まで演習してもらい、基本的な文法やコーディング方法を身に付けてもらった。

プログラミングの経験の有無で本自習のハードルは大きく相違する。経験がない者にとっては本研修の最大難関ポイントとなるため、座学の段階からチーム分けをしておき、チームメンバーや AI・デジタル戦略推進部のメンターに相談してもらいながら、壁を超えていくチャレンジをしてもらうこととなった。予想通り、参加者の学びの進捗状況は様々であったが、最終的な脱落者は一人も出なかった。若手メンターが自由参加型の勉強会や相談会を適宜開催し、丁寧フォローアップした点も効果的であった。

プログラミングに限らず、ビジネス英語や財務会計、IT やクラウド、データ分析などのスキ

ル獲得を OJT のみで進めるのは難しい。学びの対象物にもよるが、何らかの自習行為がないと能力の蓄積には限界がある。IT 企業であれば IT 研修体制やサポートがありうるが、一般企業において社員の IT スキルをどのような手法で積み上げていくかは、大きな課題となっている。バックグラウンドが多様な参加者を対象とした本研修では、最低限の IT スキルを揃えるため、前述の自習課題に加えて第 3～4 講のハンズオン実践で様々な工夫を施している。

なお、自習コンテンツと第 3 講のマテリアルは本レポートの筆者が担当した。開発環境の準備と第 4 講のマテリアルは AI・デジタル戦略推進部のスタッフが担当している。

第 3 講：開発環境の紹介

生成 AI アプリケーションの開発では、「マルチモーダル LLM を API 経由で動かし、アウトプットを得る」ことが基本動作となる。実際のサービスでは LLM へのリクエストとレスポンスの一往復ではなく、さらに複雑なフロー処理が必要となる。まずは基本形の習得に主眼を置いたハンズオンを実施し、後述する応用系にもチャレンジした。応用演習ではビジネス業務に直結する事例を取り上げている。

こうしたハンズオンには LLM と開発環境が必要となる。LLM は Azure にデプロイした GPT のエンドポイント url 情報と API-Key を提供した。これらの情報管理の重要性についてもレクチャーし、なぜ講義開始直前に提供され終了後に即クローズされるか（エンドポイントと API-Key を変更）を理解してもらった。

開発環境については、SBI グループが 2020 年からデータ利活用・分析推進のため [DataRobot](#) 社と [資本提携](#) し、機械学習やビッグデータ分析の環境を整えてきたことを活用した。具体的には、DataRobot が提供する CodeSpaces を参加者各自に開発環境として提供した（UI は図表 4 を参照）。CodeSpaces は、DataRobot プラットフォーム内でコード主体の開発・実行環境を統合的に提供する機能であり、従来から提供されていた Notebook 環境（前述の Jupyter notebook 類似）を拡張し、複数ファイルの管理、永続的なファイルシステム、Git 連携、統合ターミナルなどを備えた本格的な開発空間として設計されたものである。PC 単体のローカル環境で利用される Jupyter notebook の標準コードフォーマット（.ipynb ノートブック）だけでなく、一般的なコードを扱える汎用性を持つ。

本研修ではハンズオンに必要な機能に限定して CodeSpaces を提供した。多数のユーザに対して共通の機能セットを備えた実行環境を提供するためコンテナ技術を用いており、ユーザごとに分離された環境をコンテナ展開した。こうした特長は集合研修での利用に便利である。また、受講者が UNIX 環境に習熟していなくとも、GUI で開発環境を簡単に立ち上げ、操作することが可能となる点も大きなメリットであった。

本研修の技術的な中核は、CodeSpaces 上の Python Notebook 環境から、Azure 上の GPT に

アクセスし、LLM をプログラムから制御する一連のプロセスを体験的に理解する点にある。2つのクラウドサービスを使っていることの理解も第2講を実践体験するかたちで進めることができる。

図表 4 DataRobot の CodeSpaces イメージ



第3講：内容

ハンズオンの冒頭では、環境変数と設定管理の扱いを学習した。API キーやエンドポイント情報をコード中に直接記述せず、.env ファイルを通じて外部化している（図表 6）。これは単なるセキュリティ配慮にとどまらず、開発・検証・本番といった複数環境を切り替えて利用する際の基本的な設計思想を理解させる狙いがある。生成 AI を業務システムに組み込む際、こうした設定管理の巧拙が運用負荷やリスクに直結することを学べる構成となっている。

次に、LLM への API リクエストとレスポンス処理を実行した。Python コードからモデルを指定し、プロンプトを入力として送信し、生成結果をレスポンスとして受け取る。単に「生成された文章を見る」ことが目的ではなく、リクエスト構造、レスポンスのデータ形式、ステータス管理といった API の基本要素を明示的に理解することを重視した。生成 AI をブラックボックスとして扱うのではなく、「外部サービスの一種」として捉える視点を持つことを目的としている。

生成 AI 特有のパラメータ制御も実践した。temperature や最大トークン数といったパラメータを変更し、同一の入力に対して出力がどのように変化するかを検証することで、LLM が確率的モデルであること、すなわち出力の再現性が保証されないことが理解できる。出力の揺らぎや再現性に対するリテラシーは、業務適用において極めて重要である。生成結果をそのまま業務判断に用いるのか、後段で検証・補正を行うのかといった設計判断は、こうしたモ

デル特性の理解なしには行えない。

次に、生成 AI を単発で呼び出す段階から一歩進み、生成結果を変数として保持して後続処理に利用するフローを学んだ。これにより、LLM がアプリケーション全体の中でどのような位置づけにあるのか、すなわち入力データの加工、条件分岐、ログ保存といった通常のプログラム処理とどのように接続するのか、業務ロジックと LLM をどのように接続するのかを具体的にイメージできる。生成 AI を「文章生成器」から「システム部品」へと位置づけし直す重要なステップである。

このように、ハンズオン研修は生成 AI の最新技術を単に紹介するものではなく、API 連携、クラウド実行環境、設定管理、確率的出力の理解、パラメータ制御、アプリケーション統合といった実務に直結する要素を一貫した流れで体験する点に特徴がある。生成 AI 活用が PoC 段階から本格導入へと進む中で、これらの学びは、DX 推進や AI 活用基盤の内製化において重要な基盤となる。

ハンズオンの後半では、図表 5 に示したようなビジネス実践に直結する事例を扱い、受講者の関心と次のコンペステージへのヒントにもなるような題材を提供している。ちなみに、こうした例題の作成準備にも生成 AI を活用している。図表 6 には二番目の事例のコードを示した。

図表 5 実務を意識した実践的なハンズオン

外部ドキュメント自動取得と批判的分析	Web 上の PDF からテキストを抽出し、要約のみならず論理的な矛盾や主張の根拠の弱さを特定する「批判的検討」までを自動化する手法を学ぶ。対象は BIS のレポートで、ステーブルコインが金融システムにもたらす影響を検討したもの。
非構造データからの課題抽出と戦略策定 (図表 6 を参照)	時系列の顧客対応履歴(遅延・返信なし・交換遅延・窓口混雑・返品状況不明などの時系列ログ)から、①問い合わせ要点(問題の種類・頻度)、②注意すべきリスク(感情・不満蓄積・信用毀損など)③推奨対応方針(表現の工夫・謝罪の要否まで)、を抽出する。不満の蓄積や企業信用への影響といったリスクを分析し、ベテラン担当者の視点で最適な次回対応方針を立案するシステムをプロンプティングで設計する。
Function Calling による情報の構造化抽出	業務委託契約書ドラフトを題材に、LLM に不利条項(解除条件・損賠など)の抽出、そう判断した理由、修正提案を行わせる。JSON 形式で抽出する Function Calling を学び、システム連携を視野に入れた高精度な処理を体験する。
ドメイン知識を要する高度なリーガルチェック	法務のプロでないと見抜けないような仕組まれたトラップがある業務委託契約書を用い、再委託制限、瑕疵対応期間、一方的解除、成果物定義における曖昧さの残存、準拠法の国内外ギャップなど、専門知識が不可欠な「隠れたリスク」を指摘させ、リスクが存在する根拠を解説し、対案としての契約書を作成するシステムを体験する。これを保存し次の課題に利用することで、複雑な業務フローへの対応を体験する。
AI による成果物の自動評価と品質管理 (LLM-as-a-Judge)	上述の課題で AI が生成した回答を、人間が用意した正解リストと自動照合・採点させ、1) 最も近いチェックポイント番号とそう判断した理由、2) 契約法観点からの妥当性スコアリング(1~5点)とその判断理由、について生成 AI に作成させる。業務の効率化とアウトプットの品質担保を両立させる手法を学ぶ。

図表 6 顧客対応履歴から要点・リスク・改善点を抽出

```

❖ 例題 5：顧客対応の履歴情報から課題と改善方法を引き出す

# 顧客からの過去問い合わせ履歴（複数行の長文）
customer_history = """
1月12日：注文番号#A123の発送が遅れていると連絡がありました。メールでも電話でも返信がなかったとのことで、非常に不満を抱えていました。
1月15日：再度電話で問い合わせがあり、「今後は御社を使いたくない」と怒りを表明。担当者が平謝りし、割引クーポンで対応。
2月1日：別の注文#B456の商品に不備（部品の欠損）があり、交換を依頼。こちらも返信が遅れ、顧客は「誠意が感じられない」と発言。
2月5日：前回の件について対応が遅れている旨、再度の苦情。交換品は届いたが、同梱物の説明書が欠けていた。
3月3日：サポート窓口が繋がりにくいことに対する苦言。「何度かけても15分以上待たされた」とのこと。
3月10日：「返品が受理されたのか不明」との問い合わせ。システムに処理はされていたが、確認連絡がされていなかった。
"""

# プロンプトテンプレート（f文字列で顧客履歴を挿入）
prompt = f"""
あなたはカスタマーサポートのベテランです。以下は顧客Y12345の過去の問い合わせ履歴です。この顧客への次回対応のために、以下の3点を出力してください：
1. 問い合わせの要点（問題の種類や頻度）
2. 注意すべきリスクポイント（感情・不満の蓄積・企業信用など）
3. 次回対応方針の推奨（表現の工夫や謝罪の有無も含めて）
【問い合わせ履歴】
{customer_history}
"""

# Chat Completionリクエストの送信
response = client.chat.completions.create(
    model=deployment_name,
    messages=[
        {"role": "system", "content": "あなたはプロのカスタマーサポート担当です。"},
        {"role": "user", "content": prompt}
    ],
    temperature=0.5,
    max_tokens=1000
)

# 結果の出力
print(response.choices[0].message.content)

```

第4講の事前学習

第3講のハンズオンでは、解説とコードを notebook 形式で配布し、開発環境のターミナルでコマンドを打つことなく、対話型実行環境で Python コードを順次実行できる環境を用意した。しかし、アプリケーション開発になると、ターミナルで UNIX コマンドを操作したり、プログラミングにおける関数やクラス概念の理解と具体的なコーディング法を理解しておく必要に迫られる。

ここでの事前学習では、以下の課題を自習してもらった。前回の事前学習と同様に、チーム内での相談やメンターによるフォローアップも用意された。

1) 実行環境の遷移：対話型実行環境からバッチ方式へ

実際のアプリケーション開発では、プログラム全体を一括実行する「バッチ方式」の理解が不可欠となる。CodeSpaces の Linux 環境でバッチ方式の.py ファイルの作成と実行を行える

よう学習マテリアルを用意した。CLI 操作の基礎として基本コマンドを実践し、テキストエディタで記述した.py ファイルを CLI から実行する手順を確認した。

2) コードの構造化：関数の重要性

コードの再利用性と可読性を高めるための関数（Function）の基本構造を学び、実践した。

3) オブジェクト指向の基礎：クラスとインスタンス

AI エージェント開発フレームワークを理解する上で避けて通れないのがクラス概念である。クラス、インスタンス、属性、メソッドなどをカプセル化して管理する方法やコンストラクタなどの重要概念を実践付きで学習した。

これらは、単純なスクリプト記述から、モジュール化・オブジェクト指向を意識したエンジニアリングへのステップアップに必須のものである。特に、関数やクラスを自作する経験は、既存のライブラリ（Azure OpenAI クライアント等）がどのような構造で動作しているかを理解する一助となる。

第4講：内容

生成 AI を活用した自律型 AI エージェントの設計と実装訓練を行い、マルチエージェント・システムの技術的側面とその実装手法について解説を行った。最初に重要概念の講義を行っている。比較的新しい技術内容であり、本レポートの読者も関心があるかもしれないので、その中身を紹介する。

エージェントとは、単なる対話型のチャットボットではなく、ユーザに代わってワークフローを自律的に実行するシステムを指す。従来の RPA（Robotic Process Automation）が固定されたルールに基づく自動化であったのに対し、AI エージェントは生成 AI の推論能力を活用することで、複雑な判断、頻繁に更新されるルールへの適応、および自然言語ベースの情報処理を可能にする。

エージェントを構成する技術的 3 要素は以下の通りで、まずは、これらの概念を解説した。

- ・ Model：推論と意思決定を司るマルチモーダル LLM
- ・ Instructions：役割、制約、プロンプトを通じたワークフローの定義
- ・ Tools：外部 API や検索エンジンなど、現実世界へ働きかけるための「手足」

エージェントをイチから開発するのは不効率である。実装にはクラウドサービスやフレームワークライブラリを活用する。第3講では、Azure OpenAI と Agent SDK（OpenAI が提供するエージェント開発を支えるフレームワークコンポーネント群）を利用した。第4講では、

Google Cloud が提供する AI プラットフォーム Vertex AI と、エージェント開発用フレームワークである Google Agent Developer Kit (ADK) を採用している。同 SDK (Software Development Kit) を利用することで、ツールの定義、セッション管理、エージェント間のオーケストレーションを抽象化し、高度なワークフローを効率的に記述することが可能となる。

演習 1：エージェントを動かす

ハンズオンでは ADK を用いて最小構成のエージェントを実装するところから始めた。エージェントの機能は、「与えられたテーマ・議題に向けて、ポジティブな面から意見を述べる」ものとした。

コーディングの視点から解説すると、Agent クラスをインスタンス化し、InMemoryRunner を用いて Runner というクラスにエージェントを登録し、これを通じて実行を管理する。技術的に重要なポイントはセッション管理となる。LLM 自体はステートレス（状態を保持しない）であるため、会話のコンテキストを維持するには LLM の外部で履歴を保持し、リクエストごとにモデルへ送る必要がある。ADK の InMemoryRunner は会話履歴を保持する仕組みを提供し、ユーザごとの一貫した対話を実現している。

ハンズオンでは、1) 受講者がインストラクション（与えられたテーマについてポジティブな意見を述べるよう指示）をコード中に書き込み、2) エージェントを Runner としてインスタント化した後、3) これにメッセージ（具体的なテーマ設定）を与えてエージェントを動かし、4) どのような形で会話履歴が保持されているのか、セッションの中身を覗いて確認することを行った。

演習 2：エージェントにツールを与える

LLM は文字を出力するだけであり、web 検索やファイルを操作することはできない。エージェントの「手足」となるツール利用は、生成 AI が直接 API を叩くのではなく、関数を呼び出すためのコード（または JSON 形式の指示）を LLM が出力することで実現される。LLM が書いたコードを実行する仕組みを外で組んでおけば、あたかも生成 AI が関数を実行したように見せることができる。この仕組みがツールである。

こうした理解のもとで、前述のポジティブ議論エージェントを拡張し、議論を活性化させるファシリテーター機能も兼ね備えたエージェントを実装した。エージェントの役割は、ディスカッションテーマについて議論の認識共有を図り、議論の活発化のために必要な情報を収集して提供し、議論が出尽くしたのち各参加者の主張にスコアを付すというものである。検索を行うため、Google の ADK から検索ツールを組み込みで利用している（注意：この段階では一つのエージェントに上記の機能を全部詰め込んでおり、後に機能分解していく）。

完成したものに「今日公開されている SBIHD のプレスリリースを 1 つ検索して、議題に取り上げて討論してください」というメッセージを与え、作成したエージェントの機能を確認した。

演習 3：マルチエージェント・システムへの拡張

単一のエージェントに全てのタスクを詰め込むと、ステップが多数となり条件分岐が増えるなどワークフローが複雑化する。これに伴い精度も低下しがちになる。コードの見通しを善くするという観点からも、複数のエージェントに役割を分担させた方が望ましい。

そこで前述の課題を発展させてマルチエージェント・システムを実装する演習を行った。具体的には、ポジティブな議論を展開するエージェント、逆にネガティブな論陣を張るエージェント、議論のファシリテーターとスコア判定エージェントの 3 つに分解した。なお、ツールについても、検索エージェントや検索条件を生成するエージェントなどに機能を分化させることで、ツール選択の失敗が抑制可能となる。

実装では、Google ADK に用意されている SequentialAgent クラスを用い、複数の役割を持つエージェントをシンプルに直列に繋ぐワークフローを作り、実際に動かしてみる演習を行った。

演習 4：Flask を用いた Web アプリケーション化

いよいよ最終課題である。ここまでで構築したマルチエージェントを Web アプリケーションへ統合するプロセスを学ぶ。Web アプリケーションのフレームワークとして Flask を用い、サーバーサイドでエージェントを呼び出し、フロントエンド側でユーザと対話させることで、実運用可能な AI サービスとしての形態を整える。

まず、Web アプリケーションのフレームワークの用法に習熟するため、Hello, World! をブラウザ表示させるだけの Web アプリを作成した。ブラウザからのリクエストが Web サーバに送られ、アプリケーションサーバーで Hello, World! を返す関数が動き、レスポンスがブラウザに表示されるというシンプルな構造である。この演習により Web アプリケーションシステムの基本的な全体像を理解することができる。リクエストやレスポンスを扱うための GET メソッドや POST メソッドの学習も行っている。

次に、これを動的な Web サイトに変更するため、時刻など状況に依存した情報を含む html が返ってくるような拡張を行った。そのうえで、適宜の指示により生成 AI が html を書き換えて返すような拡張を施した。例えば、時刻や情報更新ボタンを含めるよう変更した動的 html を、もっとクールなデザインに変更するような指示を与え、これを生成 AI に実行させてコー

ドを書き換えることで Web アプリをアップデートする演習を行った。

最後に、テーマ討議を行うマルチエージェント・システムを Web アプリケーションとして実装した。ターミナル上でバッチ方式により動かしていた前述のテーマ討議システムが、Web ブラウザから実行可能なことを確認している。図表 7 では、与えられたテーマ（SBI の本日のプレスリリース）に関する材料をネットから収集している様子が示されている。図表 8 では、上記テーマと材料についてポジティブ・ネガティブ議論が陳述された後に（当該部は省略）、第三のエージェントであるファシリテーターAI が議論を展開・発展させてしている様子が示されている。

エージェントフレームワーク ADK を用いたマルチエージェントの構築経験は、複雑なビジネスプロセスの自動化において高い応用価値を持つものであり、コンペではその応用が期待される。もちろん、作成したいアプリケーションの機能や使いたい生成 AI サービスコンポーネントに応じて、LangChain/LangGraph のような汎用フレームワークや、OpenAI の Agents SDK を利用することも考えられる。例えば、Google Vertex AI を利用する構成であれば、第 4 講で利用した Google ADK が相性がよいであろう。

図表 7 AI 駆動のテーマ討議 Web アプリ（1）材料収集

Flask × Google ADK アプリ

エージェントに議論させたいトピックを書いてください:

SBIの今日のプレスリリースを取り上げて

実行

結果

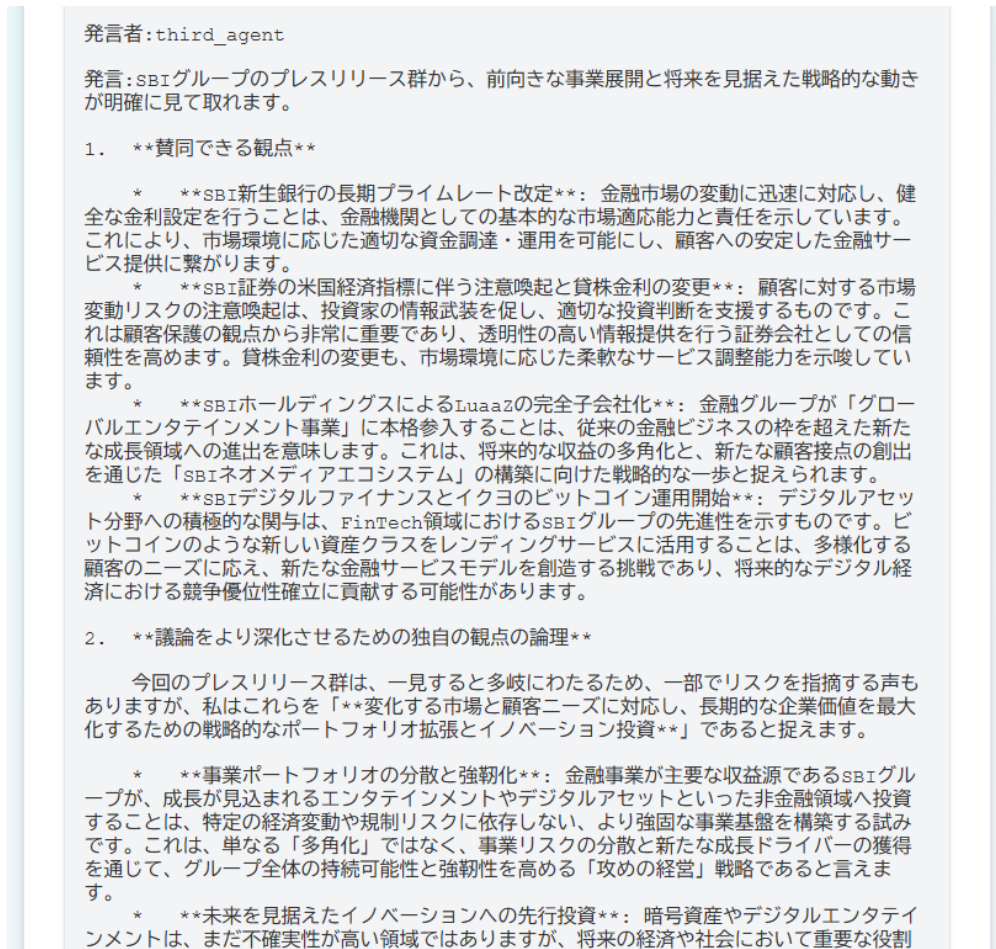
発言者: first_agent

発言: SBIの2026年1月9日付けのプレスリリースは以下の通りです。

- * **SBI新生銀行**は、2026年1月9日より長期プライムレートを改定すると発表しました。
- * **SBI証券**からは、2026年1月9日付で、米国消費者物価指数（CPI）および雇用統計発表に伴う相場変動に関する注意喚起や、貸株金利の変更に関するお知らせが複数出ています。
- * 株式会社Luuaazは、2026年1月9日に、**SBIホールディングス**の完全子会社となったことを発表しました。これにより、LuuaazはSBIネオメディアエコシステムの一員として、グローバルエンタテインメント事業を共同で推進していくとのことです。
- * イクヨは、2026年1月9日付で、**SBIデジタルファイナンス**とのパートナーシップに基づくレンディングサービスを活用したビットコイン運用を開始することを発表しました。

注) 研修実施日は 2025 年秋であったが、本レポート作成用に後日、同一課題を与えたため、2026 年入り後のプレスリリースを討議テーマとして収集している。

図表 8 AI 駆動のテーマ討議 Web アプリ（2）ファシリテーターAI の Output



おわりに

以上が新入社員垂直離陸型の生成 AI 研修のコンテンツである。注目すべきは、コーディングの経験がゼロの受講者であっても、短期間のうちに生成 AI アプリケーションの開発が「体験できる」ようになったという点である。これを可能としたのは、生成 AI 関連のツールやフレームワークが爆発的に進化した（現在もさらに発展を続けている）ことにある。

もちろん、即席養成コースであるため重要な知識やスキルの習得を省略している。「開発が体験できる」と「開発ができる」の間には大きな隔たりがある。しかし、高いゴール設定を行い、爆速でリフトオフする体験は、「自分でも作れるのである」という強烈な意識づけをもたらす。内製化への第一歩がここにある。この体験をもって、次のステージであるアプリ開発コンペが既にスタートしている。

開発の民主化・ビジネス現場への移行は、生成 AI をシステム化する技術の進歩によって更に加速するであろう。マルチモーダル LLM 本体の能力拡大が注目されがちであるが、これをユースケースとして実装するための技術革新も重要である。

こうした動きが続くことにより、IT インフラの身の回り開発は今どきのビジネスパーソンの基本素養になっていくと思われる。まずは SBI グループの未来を担う若者から学びを開始し、これを中堅ベテラン層に拡大していくことが、組織横断的な AI・デジタル戦略推進部の主要タスクのひとつとなっている。実際、グループ企業内で様々な取り組みが始まって久しい。その取り組みや効果の紹介は別の機会としたい。