

# 生成 AI 実践ガイドと企業事例集

品質・安全性・ガバナンスを統合し本番運用へ導くフレームワーク

## 謝辞

本資料の作成にあたってはさまざまな企業・団体にご協力いただきました。ヒアリングにご協力いただいた次の企業に感謝申し上げます(五十音順、株式会社などの法人格を除く)。

- アイレット株式会社
- 株式会社IVRy
- 株式会社Algomatic
- 伊藤忠テクノソリューションズ株式会社 (CTC)
- 株式会社インテージ
- KDDI株式会社
- 株式会社サイバーエージェント
- サイボウズ株式会社
- サントリーホールディングス株式会社
- 株式会社SmartHR
- 株式会社すかいらーくホールディングス (すかいらーくグループ)
- セガサミーグループ
- SpiralAI株式会社
- ダイキン工業株式会社
- DataRobot, Inc.
- 合同会社デロイト トーマツ
- 日本生命保険相互会社
- PharmaX株式会社
- 株式会社Preferred Networks (PFN)
- 株式会社メルカリ
- Ubie株式会社 (ユビー)
- LINEヤフー株式会社
- 株式会社LayerX

また、AIQMI WG2 をはじめ、レビューに参加いただいた方々に感謝申し上げます。

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構(NEDO)の委託業務の結果得られたものです。

## 免責事項

本書の内容は一般的な情報提供を目的としたものであり、個別の判断や意思決定に際しては、必要に応じて専門家等へご相談ください。本書の利用により生じた結果について、NEDOおよび事業実施者は責任を負いかねます。

# 目次

謝辞	1
免責事項	1
ライセンス	2
目次	3
1章 はじめに	5
1.1 背景と目的	5
1.2 本書で提案するフレームワーク	8
1.3 本書の構成	16
1.4 本書と既存のドキュメントとの関係性	17
1.5 本書で記述する主なメッセージ	19
2章 調査方法	20
2.1 企業等へのヒアリング	20
2.2 AI エージェントの開発	21
3章 技術	22
3.1 情報システムとしての AI エージェント	23
3.2 Human-in-the-loop	27
3.3 ガードレール	31
3.4 AI エージェントのテスト	37
3.5 モニタリング	43
3.6 技術選定	46
4章 プロセス	53
4.1 AI エージェントの開発フェーズ	54
4.2 継続的な評価による継続的な改善	59
4.3 リスクマネジメント	66
4.4 サービスリリース後の迅速なフィードバックループ、モニタリング	70
5章 組織	75
5.1 組織体制	76
5.2 ガイドラインの整備	80
5.3 統合的な AI 相談窓口	85
5.4 AI 活用支援	87
5.5 教育	91
5.6 ISO 42001 による認証	93
5.7 トップのコミット	98
Appendix 1. 各企業の取り組み事例集	102
事例の記述フォーマット	102
協力企業一覧	102

アイレット株式会社	104
株式会社IVRy	106
株式会社Algomatic	108
伊藤忠テクノソリューションズ株式会社 (CTC)	110
株式会社インテージ	112
KDDI株式会社	114
株式会社サイバーエージェント	116
サイボウズ株式会社	118
サントリーホールディングス株式会社	120
株式会社SmartHR	121
株式会社すかいらーくホールディングス (すかいらーくグループ)	123
セガサミーグループ	125
SpiralAI株式会社	126
ダイキン工業株式会社	128
DataRobot, Inc.	130
合同会社デロイト トーマツ	132
日本生命保険相互会社	134
PharmaX株式会社	136
株式会社Preferred Networks (PFN)	138
株式会社メルカリ	140
Ubie株式会社 (ユビー)	141
LINEヤフー株式会社	143
株式会社LayerX	145
Appendix 2. 外国人支援を想定したチャットボットの開発記録	147
1 はじめに	147
2 開発方針と技術選定の観点	147
3 ステップ1：開発前の非技術的議論	149
4 ステップ2：技術的議論と初期設計	150
5 ステップ3：プロトタイプングと実地検証	152
6 ステップ4：適合性を高めるための簡素化と技術選定の振り返り	155
7 ステップ5：デプロイ、テスト、および運用監視	160
8 総括	170

# 1章 はじめに

## 1.1 背景と目的

本書は、2025年現在における各組織の生成 AI 利活用方法をフレームワークとして体系化し、現時点でのスナップショットとして提示するものである。

### 1.1.1 目的

2025年現在、生成 AI はもはや身近な技術となり、AI エージェントが実際のビジネスで利用される事例も増えてきた。しかし、技術の変化は依然として激しく、登場から間もないため、AI エージェントを組織的に活用するためのノウハウは十分に体系化されているとは言えない。多くの組織が、手探り状態で取り組みを進めているのが現状である。

とくに、企業として責任を持ってサービスを提供するために不可欠な、品質保証やリスク管理の手法が確立されていないことが、本格的な導入の大きな障壁となっている。「どのようなリスク<sup>1</sup>があるのか」「どうすれば安全を担保できるのか」が不明確なため、PoC（概念実証）から先の本番運用に踏み切れないケースも少なくない<sup>2</sup>。

そこで本書では、AI エージェントを含む生成 AI を活用するための取り組みについて広く企業にヒアリングを行い、その知見をフレームワークとして体系化した。各企業が直面している課題と、それに対する具体的な解決策や運用手法を共有することで、組織的な活用の指針となることを目指す。

### 1.1.2 想定読者

本書は、AI エージェントを組織的に活用しようとするすべてのステークホルダーの「共通言語」となることを目指している。そのため、AI エージェントや生成 AI の活用に関わる、以下のような多様な立場の人員を対象読者として想定する。

- AI 事業者ガイドラインにおける AI 利用者や AI 提供者
- 特定業務の専門家（ドメインエキスパート）
- AI リスクマネジメント担当者

---

<sup>1</sup>一般的に「リスク」という用語は、不確実性そのものを指すこともあるが、本書では将来的に発生しえるユーザーや組織に不利益をもたらす事象を指す。

<sup>2</sup>同様の指摘は多数なされているが、たとえば Mckinsey の "The State of AI 2025" では "AI use continues to broaden but remains primarily in pilot phases" と表現されており、本邦に限った問題ではない

<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>

- AI ガバナンス担当者
- 生成 AI に関わる必要が生じた法務担当者、人事担当者
- AI 活用の担当者

本書は、生成 AI の活用をミッションとして持つすべての担当者を対象としている。とくに、「急に生成 AI 活用の推進を任されたが、何から手をつければ良いか迷っている担当者」や、「現場からの活用要望と安全性の担保の間で板挟みになっているガバナンス担当者」にとって、実務的な指針となることを意図している。

また、データサイエンティスト、機械学習エンジニア、AI 事業者ガイドラインにおける「高度な AI システム」を開発する AI 開発者といった機械学習モデルの専門家にとっても、本書が他のステークホルダーとの共通言語となることを目指す。

ただし、本書は、機械学習モデルそのものを構築・学習させるための技術的ノウハウはスコープに含めない。本書は生成 AI や AI エージェントを、組織として活用し、安全に運用していく方法論について記述する。本書のスコープについては改めて次に述べる。

### 1.1.3 本書のスコープ

本書は幅広く生成 AI や AI エージェントの活用方法について扱う。そのため、生成 AI をそのまま、あるいはシステムのコンポーネントのひとつとして扱うシステムを対象とする。

また、次にあげる提供形態に関しても特定の形態に限定しない。たとえば、次の提供形態すべてを扱う。

- 社内システム
- 一般消費者向けサービス
- 法人向けサービス

同様に、生成 AI の活用のためのノウハウを幅広く提供することをめざす。たとえば、次のような用途はすべて本書の対象とする。

- 社内業務の効率化とナレッジ活用
- 専門業務のプロセス自動化・高度化
- 顧客対応・接客の自動化

ここで、「社内業務の効率化とナレッジ活用」は、社内規定、マニュアル、過去のインシデント事例などの膨大な内部ドキュメントを AI に集約・参照させ、検索や要約、問い合わせ対応、コーディング支援、議事録作成などで活用することを指す。

「専門業務のプロセス自動化・高度化」は経理、採用、医療、法務といった高度な専門知識を要するドメインにおいて、AIが書類の構造化、下書き作成、チェック業務などを代行・補助し、専門家の判断を自動化、あるいはサポートすることを指す。

「顧客対応・接客の自動化」は電話自動応答、FAQ、商品レコメンデーション、キャラクター対話など、エンドユーザーとの接点をAIが担うことで、24時間対応やパーソナライズされた体験を提供することを指す。

なお、本書では高度な開発を伴うものだけでなく、以下のような利用形態も初期段階の取り組みとして重要視してスコープに含める。

- セキュリティ要件を満たした環境で、ベンダーの提供する対話型 AI を従業員に提供
- 社内ドキュメントを検索・参照させる RAG システム
- Copilot 等のベンダー提供ツールを業務フローに組み込む形態

とくに、「社内システム（社内向けチャットボット等）」は、外部向けサービスと比較してリスクの影響範囲を限定しやすいため、組織が生成 AI の特性を理解し、ガバナンスや評価プロセスを成熟させるためのステップとしても機能する。本書では、こうした初期的な活用を通じて得られた知見を元に、より業務に深く組み込まれた高度な活用へと適用範囲と価値を段階的に育てていく方法論を提供する。

一方で、本書で提供する活用のための取り組みそれぞれは、すべての生成 AI や AI エージェントに適用できるわけではない。これは「1.1.4. 本書の利用方法」で改めて述べる。

### 1.1.4 本書の利用方法

本書は生成 AI の活用のために行われたさまざまな取り組みをフレームワークとして体系化し紹介するものであり、各組織で行われる AI 活用における参考資料となることを意図している。

本書は守るべきルールや指針を示すガイドラインや、実行すべき取り組みを網羅的に示したチェックリストではない。あくまで、「AI 事業者ガイドライン<sup>3</sup>」のような既存のガイドラインが示す「あるべき姿」に対して、それを具体的にどう実現するかという「方法論」を提示するものであり、両者は相互に補完し合う関係にある。

また、本書は具体的な実装・運用方法を提供するが、本書に記述した内容をすべて実行すればガイドラインを遵守できる、あるいは安全性が担保されるというものではない。本書に掲載されているさまざまな企業は、それぞれの企業の立場やビジネスの状況に向き合って生成 AI の活

---

<sup>3</sup>AI事業者ガイドライン (経済産業省)

[https://www.meti.go.jp/shingikai/mono\\_info\\_service/ai\\_shakai\\_jisso/20240419\\_report.html](https://www.meti.go.jp/shingikai/mono_info_service/ai_shakai_jisso/20240419_report.html)

用のための取り組みを進めていた。企業や組織、あるいは企業内のチームがビジネスの状況に向き合って活用する際に、本書が一助となることを目指す。

既存のガイドラインとの関係性は「1.4. 本書と既存のガイドラインとの関係性」で改めて述べる。

## 1.2 本書で提案するフレームワーク

本書では、AI エージェントを含む生成 AI の活用のためのフレームワークとして GenAIOps を提案する。GenAIOps は、DevOps を生成 AI を前提として拡張したものである。これは技術者や開発チームだけではなく、プロセスや組織文化にも言及する広範な概念である。

以降では、GenAIOps という新たな拡張が必要な理由について、DevOps および MLOps の歴史的経緯を踏まえて述べる。

### 1.2.1 DevOps 概略

GenAIOps は DevOps の拡張であるため、まずは DevOps について概略を述べる。ここでは DevOps の包括的な記述を行うのではなく、本書の議論に関連する以下の2点について述べる。

- 対立から多能工化による協調へ
- 不確実性への対応: 事前の計画から探索と検証へ

なお、DevOps の包括的な記述については他の書籍<sup>4</sup>を参照されたい。

対立から多能工化による協調へ

DevOps の発祥とされる Velocity 2009 でのプレゼンテーション<sup>5</sup>では、開発者 (Dev) と運用者 (Ops) の対立構造が指摘された。

このプレゼンテーションの扱う状況では、システムの開発を行うチームと、運用を行うチームが分けられており、それぞれが別の価値観を持って対立していた。

開発者 (Dev) は「顧客に新しい価値を早く提供したい (機能追加による変更)」と望む一方、運用者 (Ops) は「顧客に安定的に価値を提供したい (変更による不安定化の回避)」と望んでいた。同じ技術者であっても、両者の価値観の違いから対立が生まれていた。

---

<sup>4</sup>たとえば ニコール・フォースグレン、ジェズ・ハンブル、ジーン・キム (著) 武舎るみ、武舎広幸 (訳) 『LeanとDevOpsの科学 [Accelerate] テクノロジーの戦略的活用が組織変革を加速する』インプレス、2018年

<sup>5</sup>10+ Deploys Per Day: Dev and Ops Cooperation at Flickr - Slideshare

<https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>

しかし、両者の最終的な目的は「顧客への価値提供」で一致している。そのため、この対立を「開発と運用の一体化」によって解消しようという提案が初期の DevOps でなされた。DevOps チームでは開発者と運用者がチーム内に存在するのではなく、エンジニアが開発も運用もできるようになることで運用と開発の一体化を達成する。これは、トヨタ生産方式における「多能工化<sup>6</sup>」のアプローチと等しい。

このようにして、組織における対立を多能工化により協調へと変革することが DevOps では取り組まれた。

不確実性への対応: 事前の計画から探索と検証へ

事前に徹底的な計画を行うことで不確実性を極小化しようとする計画駆動型開発とは異なり、アジャイルソフトウェア開発は不確実性を前提として、探索と検証を通じて価値を提供しようとする。DevOps はこのアジャイルソフトウェア開発の系譜にあり、探索と検証を通じて価値を提供しようとする点は共通している。

たとえば、DevOps ではコードだけでなく設定ファイルやインフラ構成情報も含めたすべてをバージョン管理システムで管理し、IaC (Infrastructure as Code) の技術を用いて環境構築を自動化することは一般的になっている。これにより、誰でも・いつでも・何度でも同じ実験環境を再現でき、失敗してもすぐに元の状態に戻せるため、安心して試行錯誤を繰り返せる状態を実現している。

なお、DevOps の詳細な解説をするのは本書の目的から外れるためここでは詳細の記述を避けるが、DevOps は IaC のような技術だけではなく、開発プロセスや組織文化を変革する取り組みも含まれる<sup>7</sup>。

## 1.2.2 MLOps 概略

GenAIOps は DevOps を生成 AI や AI エージェントといった新しい技術に対応するために拡張したものである。

DevOps の拡張には MLOps という前例がある。ここでは MLOps について概略を述べる。なお、MLOps の詳細については別の書籍<sup>8</sup>を参照のこと。

---

<sup>6</sup>トヨタ生産方式における多能工化とは、複数の異なる作業や工程に従事できる社員を育てることを指す。これにより、特定の業務がボトルネックになったとしても、人の配置を柔軟に行うことにより、ボトルネックを解消できることを目指す。DevOps では開発者が運用もできるようにすることで、デプロイがボトルネックにならないようにしている。

<sup>7</sup>DORA のサイトから最新のレポートや取り組み一覧が確認できる <https://dora.dev/>

<sup>8</sup>たとえば、杉山 阿聖, 太田 満久, 久井 裕貴 (著) 『事例でわかるMLOps 機械学習の成果をスケールさせる処方箋』 講談社 2024年

MLOps は、DevOps チームをデータサイエンティスト方面に多能工化したものと言える。深層学習の流行により、データサイエンスという新しい分野に脚光があたり、データサイエンティストという職種が誕生した。データサイエンティストはデータの利活用においては優れた専門性を発揮するが、彼らが欲するデータはあらかじめ利用しやすい状態で存在するとは限らない。データの収集に必要な API の新規開発や、データウェアハウスなど基盤の整備、などデータサイエンティストが活躍しやすい環境の整備が必要となる。

また、データサイエンティストの成果である機械学習モデルをシステムに組み込むためには独特な困難が付きまとう。そのような開発は一般的な Web システムで必要となる知識とは別の知識も要求されるため、開発が遅延することもしばしば観測された。

このような経緯から、機械学習の成果をすばやくシステムに実装するための職種として機械学習エンジニアが誕生し、MLOps チームに組み込まれるようになっていったと言える。

### 1.2.3 MLOps における対立

MLOps ではステークホルダーが増えた結果、新たな対立構造も生まれるようになった。MLOps はデータを活用する都合上、データの扱いに関する対立が生じることがある。

データに関する法制度は GDPR を代表として各国でさまざまに定められており、プライバシーへの配慮が一般に求められる。そのため、企業には収集した情報を適切に扱うための、情報セキュリティマネジメントシステム (ISMS) の構築が求められる。ISMS はセキュリティや法的要請の遵守が求められるため、情報セキュリティ担当や法務担当、リスクマネジメント担当といった、MLOps チームとは目的が異なる役割の人員がステークホルダーとなる。MLOps の文脈ではデータの利活用による新たな価値提供をめざし、情報セキュリティやプライバシー、リスクマネジメントの側面からはデータを限られたスコープの中で厳格に守ることを典型的には目指す。このため、活用と守備という異なる目的の間で対立が生じる場合がある。

このような対立は、3ラインモデル<sup>9</sup>を用いると役割の違いから生じるものであると説明できる。3ラインモデルはそれぞれの役割を次のように定義する。

- 第1ライン: 顧客への製品やサービスの提供、支援機能 (人事、総務、設備など) を含む
- 第2ライン: リスクの管理の支援
- 第3ライン: 内部監査

---

<sup>9</sup>IIAの3ラインモデル：3つのディフェンスラインの改訂

[https://www.iiajapan.com/leg/pdf/data/iiia/2020.07\\_1\\_Three-Lines-Model-Updated-Japanese.pdf](https://www.iiajapan.com/leg/pdf/data/iiia/2020.07_1_Three-Lines-Model-Updated-Japanese.pdf)

第2ラインの扱う領域としては、法規制やITセキュリティなどが挙げられる。この分類では、MLOps チームは第1ラインに位置づけられ、情報セキュリティ担当や法務担当、リスクマネジメント担当は第2ラインに位置づけられる。このため、次のような対立が発生することがある。

- MLOps チーム (第1線) は有用な可能性のあるデータであれば利用したい
- 第2線としてはリスクのあるデータであれば利用させたくない

GenAIOps における第1ラインと第2ラインの対立構造については以降で触れる。

### 1.2.4 GenAIOps で取り組む領域

ここでは GenAIOps で取り組むべき領域について、クネビンフレームワーク<sup>10</sup> (Cynefin Framework) を元に述べる。

#### クネビンフレームワーク (Cynefin Framework)

クネビンフレームワークは、直面している状況を因果関係の明快さや予測可能性の度合いに応じて5つの領域<sup>11</sup> (Simple:自明、Complicated:煩雑、Complex:複雑、Chaotic:混沌、Disorder:混乱) に分類し、それぞれの特性に適したアプローチを導き出すための枠組みである。

- Simple (自明): 因果関係が明白で、予測が極めて容易な状態を指す。すでに分かっている問題に対して、すでに分かっている解決策を適用すれば良い領域。「感知 (Sense) - 分類 (Categorise) - 応答 (Respond)」のアプローチをとる。
- Complicated (煩雑): 因果関係は存在するが、専門知識や分析により判明する領域。「感知 (Sense) - 分析 (Analyze) - 応答 (Respond)」のアプローチをとる。
- Complex (複雑): 多数の要素が相互に影響し合い、事前に結果を正確に予測することが不可能な領域。何が分からないのかさえ分からない状態で、状況に応じて最適な解が変化し続ける。「探索 (Probe) - 感知 (Sense) - 応答 (Respond)」のアプローチをとる。
- Chaotic (混沌): 前例がないような危機的状況で、物事が完全に無秩序な状態。即時の対応が必要で、分析や実験をしている時間的余裕はない。「行動 (Act) - 感知 (Sense) - 応答 (Respond)」のアプローチをとる。
- Disorder (混乱): 状況が整理されておらず、どの因果関係が適用されるのか判断できていない状態。

---

<sup>10</sup>Snowden, D. J., & Boone, M. E. (2007) "A Leader's Framework for Decision Making", Harvard Business Review.

<sup>11</sup>クネビンフレームワークに関する継続的な研究活動により、現在はそれぞれの領域が再定義されているが、ここでは Snowden & Boone (2007) のものを用いる。

これらの領域はさらに区分され、Simple と Complicated の領域をあわせて「予測可能な領域」、Complex と Chaotic な領域をあわせて「予測不可能な領域」ということがある。クネビフレームワークを次に図解する。

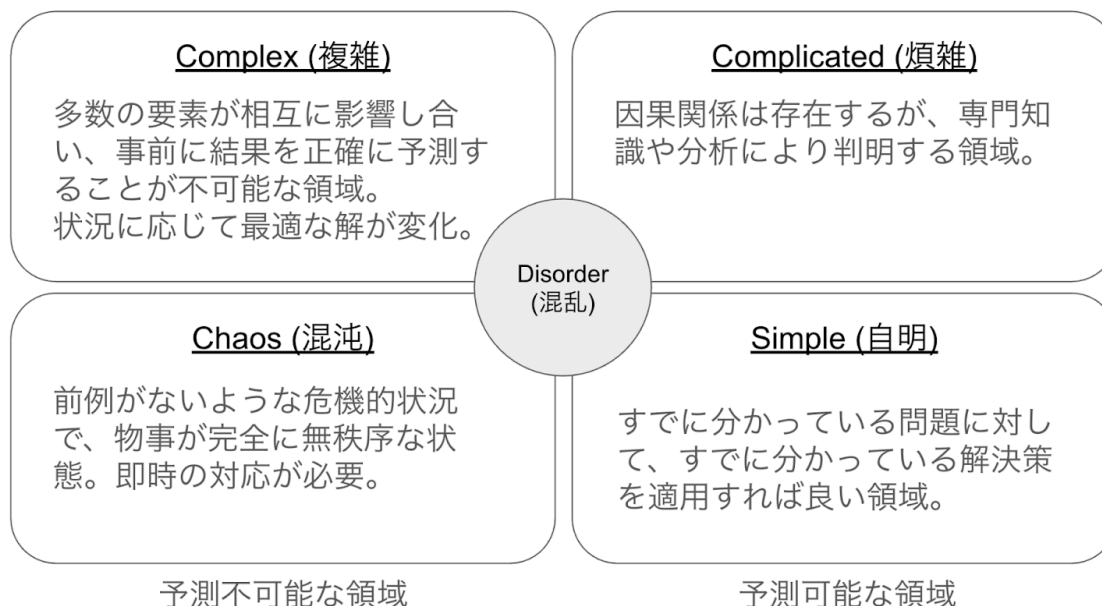


図1. クネビフレームワーク

### 生成 AI 活用における不確実性

DevOps は予測不可能な領域、とくに Complex な領域に対する解決策として、現代の Web サービスのような複雑な領域に対して、探索と検証を行い価値を発見していくアプローチである。

生成 AI の登場以降、これまでシステム化の検討すらされてこなかったさまざまな業務領域での活用が、改めて検討されるようになった (Disorder な初期状態)。また、生成 AI という新技術を起点として、ゼロベースで課題解決に取り組むハッカソンの活動も増加している (Chaotic な領域)。

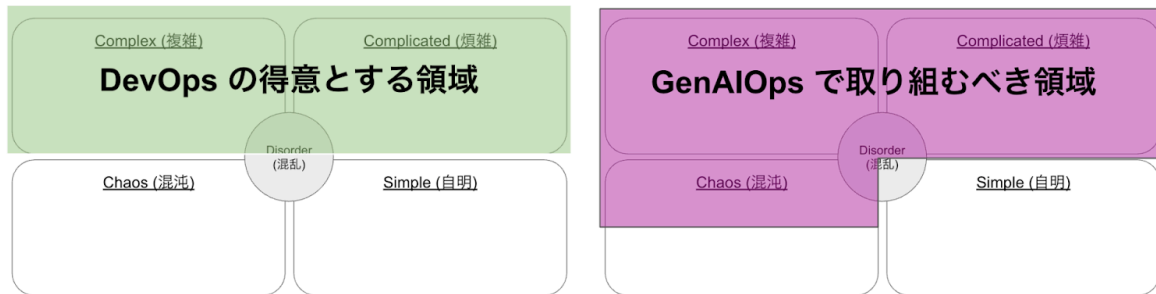


図2. DevOps の得意とする領域と GenAI Ops で取り組むべき領域

このように、生成 AI の活用では未整理で不確実性の高い領域に対する新たな方法論が求められる。具体的には、以下の3つのような多層的な不確実性に挑む必要がある。

- 対象業務の不確実性: 暗黙知を含み、ルール化できない業務だからこそAIを使うという逆説があり、どの業務に適用可能か分からない。
- 技術の不確実性: 数ヶ月で前提が覆る技術進歩の速さや、ハルシネーションなどの不確実性により、どのような出力が得られるか確約できない。
- 提供価値の不確実性: 専門家すら正解を言語化できず、基準が漂流する (Criteria Drift) ため、それが本当にユーザーの役に立つか分からない。

こうした状況に適応し、価値を探索するための方法論として DevOps を生成 AI を前提に拡張し、GenAI Ops を提唱する。

### 1.2.5 GenAI Ops での専門家・第2線との協調

GenAI Ops では MLOps とは異なる方向に、DevOps を「多能工化」により拡張する。GenAI Ops においては、対象業務の専門性を多能工化により獲得することを目指す。

#### 業務の専門家との協調

従来の DevOps において、業務の専門家 (ドメインエキスパート) はチーム外のステークホルダーとして扱われることが一般的であった。たとえばスクラムガイド<sup>12</sup>においても、業務の専門家はスプリントレビューに参加して成果物をレビューするステークホルダーとして整理されている。

しかし、今回行った調査の結果、GenAI Ops を実践するチームにおいては、業務の専門家を開発チームの構成員として扱い、協調して作業するケースが多く確認された。専門家はプロンプトの記述や評価用データの構築を行い、チームの一員として開発サイクルに参加していた。ま

<sup>12</sup>スクラムガイド 2020

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Japanese.pdf>

た、開発チームも専門家と一緒に改善サイクルを回すことで、業務で提供したい真の価値を探索している様子が確認できた。

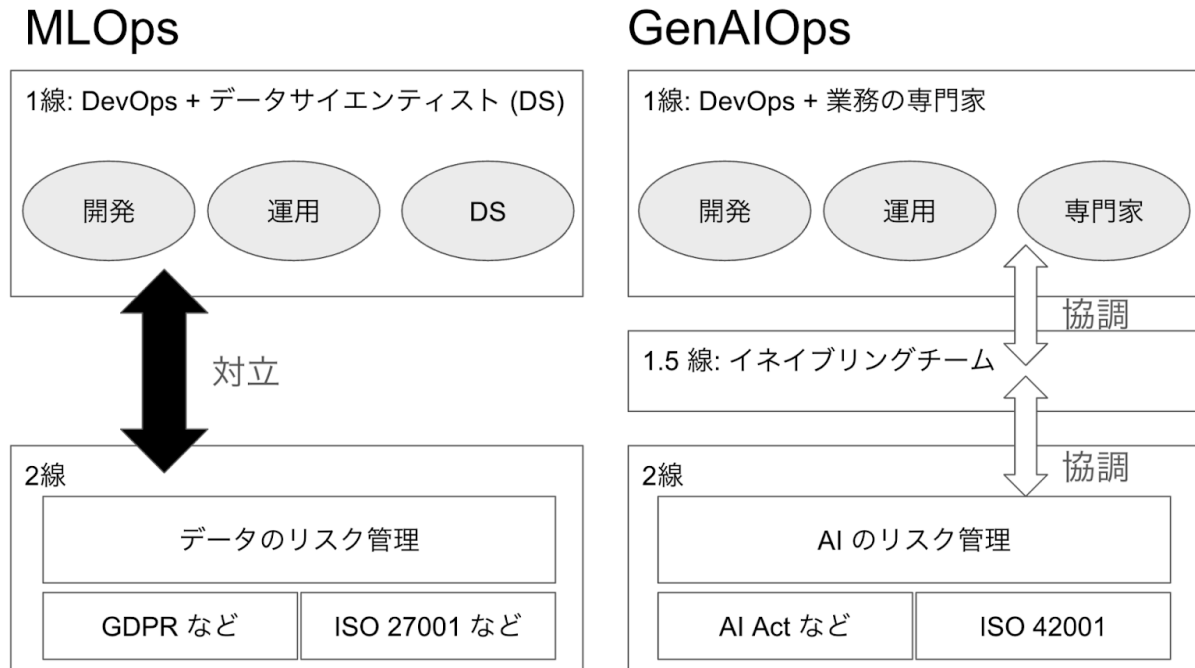


図3. MLOps と GenAIOps の対比

このため、「GenAIOps では外部のステークホルダーとして扱われてきた業務の専門家はチームの一員であり、チームは多能工化により業務知識を獲得して協調して作業する」という拡張を提案する。

さらに、この体制は提供価値の探索だけではなく、リスクマネジメントにおいても効果的に機能する。なぜなら、「AI に何をさせるか（提供価値）」を定義することは、「AI に何をさせてはいけないか（リスク）」を定義することと表裏一体だからである。たとえば、「ユーザーに合わせて親身に相談に乗る」という価値を追求すれば、そこには「専門外の助言をしてしまう」「感情的に巻き込まれすぎる」といったリスクが必ず付随する。つまり、業務の価値をもっとも理解している専門家と開発チームが中心となって開発を進めること自体が、もっとも確実に本質的なリスクへの対応となる。

これは AI セーフティとも関連するため以降で改めて述べる。

#### 第2線との協調とイネイブリングチーム

「GenAIOps チームは業務の専門家と協調して作業するとともに、第2線のステークホルダーとも協調して作業する」ことも提案する。

データの利活用においてプライバシーへの配慮が必要だったように、AIの利活用においてもAIセーフティを代表とする特有のリスクへの対応（リスクマネジメント）が求められている。法制度としてもEU AI Act<sup>13</sup>を代表として整備が進んでおり、組織的なAIガバナンスのもと、AIマネジメントシステム(AIMS)を運用することが求められるようになってきている。

本邦においては、次の代表的なガイドラインが公表されている

- AIセーフティに関して、AIセーフティ・インスティテュートから「AIセーフティに関する評価観点ガイド<sup>14</sup>」が公表され、AIセーフティ評価の際の基本的な考え方が示されている
- 経産省・総務省から「AI事業者ガイドライン」が示され、遵守すべき「共通の指針」や、AI開発者・AI提供者・AI利用者といった主体ごとの遵守すべき事項、とくに高度なAIシステムに関係する事業者の守るべき事項が示された。また、「AIに関するリスクをステークホルダーにとって受容可能な水準で管理しつつ、そこからもたらされる便益を最大化する」ために、組織のガバナンス体制を構築して、アジャイルガバナンスを実践することの重要性が示された。

また、ISMS (ISO/IEC 27001、以下 ISO 27001) と同様に、AIMS のマネジメント標準として ISO/IEC 42001 (以下、ISO 42001) が制定されており、事業者は必要に応じて第三者認証を受けることができる環境も整いつつある。ISO 42001 は ISO 27001 と異なり、AIMS について AI セーフティのような AI 固有のリスクに配慮するだけでなく、AI による便益も継続的に改善するよう求めている。

一方で、AI リスクマネジメントにおいては次のような差異が調査の結果確認できた。

- AI セーフティで対応すべきリスクは、第1線の GenAIOps チームが向き合っている業務に大きく依存するため、適切にリスクを把握するためには生成 AI のユースケースを適切に把握する必要がある
- 生成 AI の利活用が社内外で強く求められるようになってきた結果、AI リスクマネジメントにおいて「AI を活用しないリスク (機会損失リスク)」への対応を掲げ、全社的な統制を行う AI ガバナンスチームは AI 活用を後押しする立場も取ることが確認された
- ISO 42001 の認証を受けた組織では AI のユースケースを明確にし、誰が・何のために使い、どんなリスク・便益があるのか明確にしていた

---

<sup>13</sup>Regulation (EU) 2024/1689 (Artificial Intelligence Act), 2024.

<sup>14</sup>AIセーフティに関する評価観点ガイド

[https://aisi.go.jp/output/output\\_framework/guide\\_to\\_evaluation\\_perspective\\_on\\_ai\\_safety/](https://aisi.go.jp/output/output_framework/guide_to_evaluation_perspective_on_ai_safety/)

これらの結果から、GenAIOps における第2線 (AI ガバナンスチーム) は、第1線を監視・監督するだけの存在ではなく、「イネイブリングチーム (Enabling Team)」<sup>15</sup> として機能することを提唱する。

イネイブリングチームとは、特定の専門知識やツールを開発チームに提供し、彼らが自律的に課題 (AI 特有のリスクへの対応など) を解決できるよう能力的な支援 (イネイブルメント) を行うチームである。これにより、組織全体として一貫した AI セーフティを実現する協調体制を構築する。

また、調査の中で確認された AI セーフティの実現のためのリスクマネジメントは、この協調体制の中で行われていた。そこでは、リスク回避のために利用を過度に制限するのではなく、「安全に価値を提供できる範囲 (スコープ)」を正確に見定めその範囲内で最大限の価値を提供することに重きが置かれていた。これにより、「AI を活用しないリスク」を避けつつ、段階的に適用範囲を拡大していくことが可能となっていた。

## 1.3 本書の構成

本書は、ヒアリングから得られた知見を DevOps の 3 要素である「技術」「プロセス」「組織文化」の観点から整理し、GenAIOps に特有な取り組みを提供する。具体的には以下の章構成となっている。

- 第2章 調査方法概要: 本書の知見の基になった、国内企業へのヒアリング調査のアプローチについて詳述する。
- 第3章 技術: ガードレール、テスト、Human-in-the-loop など、AI エージェントの品質と安全性を技術的に担保する具体的な工夫を解説する。
- 第4章 プロセス: PoC から本番開発、継続的な改善に至るまでの開発プロセスと、リスクマネジメントの手法を解説する。
- 第5章 組織文化: AI 活用を全社的に推進するための組織体制、ガイドライン整備、人材育成などの取り組みを解説する。

また、Appendix として次の内容を記述する。

- Appendix 1: ヒアリングの対象とした各企業がどのような目的で AI エージェントを用いており、どのような課題に対してどのような対策を行っているか示す。
- Appendix 2: 調査の結果得られた GenAIOps の取り組みを、実際に「外国人支援を想定したチャットボット」を実装することで検証した際の検討事項を共有する。

---

<sup>15</sup>イネイブリングチームは次の書籍が詳しい Matthew Skelton, Manuel Pais: Team Topologies, IT Revolution Press, 2019.

## 1.4 本書と既存のドキュメントとの関係性

本書は、「AI 事業者ガイドライン」や「AI セーフティに関する評価観点ガイド」といった既存のガイドラインとは補完し合う関係にある。一般的にガイドラインはあるべき姿や指針をトップダウンに示すのに対し、本書はそれらを具体的に運用するためのノウハウや具体例をボトムアップに提示する。

以下に、AI に関連する代表的なドキュメントとの関連について述べる。

### AISI 「AI セーフティに関する評価観点ガイド」との関係

AI セーフティ・インスティテュート (Japan AISI) による「AI セーフティに関する評価観点ガイド」では、「AI システムの開発や提供に携わる者が AI セーフティ評価を実施する際に参照できる基本的な考え方」が提示されている。本書はこれらの考え方に基づいて、AI セーフティを実現するための具体的な方法論や事例を提供する。また、GenAIOps のフレームワークの中で、実際のシステム設計やプロセスの中に評価・改善をどう組み込み、実行するかという具体的な取組内容を例示する。

### 「AI 事業者ガイドライン」との関係

経済産業省・総務省による「AI 事業者ガイドライン」は、「AI ガバナンスの統一的な指針を示す」ものであり、アジャイルガバナンスの理念を提唱している。

本書はアジャイルガバナンスの理念に基づいて、AI ガバナンスを実現するための具体的な方法論や事例を提供する。また、GenAIOps のフレームワークの中で開発チームとガバナンスチームの密な協調（イネイブリング）といった組織間のコラボレーションにより実現する方法論や、ISO 42001 に基づく第三者認証に向けた活動についても述べる。

本書で提示する GenAIOps は、AI を活用するための方法論であると同時に、AI セーフティおよび AI ガバナンスのための方法論とも合致する。したがって、本書はこれらの既存ガイドラインや国際標準（ISO 42001）との互換性を保ちながら、それらを実践するための具体的な指針を提供するものである。

### 「機械学習品質マネジメントガイドライン」との関係

産業技術総合研究所 (以下産総研) による「機械学習品質マネジメントガイドライン<sup>16</sup>」は、機械学習を利用したシステムの品質に関する基準と達成目標を定めているガイドラインである。

---

<sup>16</sup>機械学習品質マネジメントガイドライン 第4版  
<https://www.digiarc.aist.go.jp/publication/aiqm/guideline-rev4.html>

「機械学習品質マネジメントガイドライン」と本書は互いに補完し、両立する関係にある。

「機械学習品質マネジメントガイドライン」は機械学習システムにおける開発ライフサイクルにおいて運用改善段階に DevOps を採用しており、イテレーティブな取り組みにおける機械学習システムの品質の基準について論じている。本書は DevOps を生成 AI を用いるシステム開発に拡張し、GenAIOps として提唱している。このため、互いに共通の基盤として DevOps を採用しており、開発プロセスの考え方は共通している。また、「機械学習品質マネジメントガイドライン」は「品質検査（テスト）結果により進捗を管理するアジャイル型（反復型）の開発段階と、工程管理に注力するトップダウンの V 字開発を統合するハイブリッドなモデルを示しており、この考え方自体は生成 AI を前提とした GenAIOps においても適用できる。

このように、「機械学習品質マネジメントガイドライン」の示す品質検査の考え方は本書の提案する GenAIOps と両立する。

## 「生成 AI 品質マネジメントガイドライン」との関係

産総研による「生成 AI 品質マネジメントガイドライン<sup>17</sup>」は生成 AI の品質マネジメントに関するガイドラインである。このガイドラインは次の2つの点が特徴的である。

- 生成 AI そのものではなく、生成 AI を中核とするソフトウェアシステム (生成 AI 利用 AI システム) を対象とすること
- プロンプトの工夫などを行うラピッドプロトタイピングを要求獲得・形式化の一部とみなし、スコープの範疇として扱っていること

本書の対象は「生成 AI 品質マネジメントガイドライン」と同様に、生成 AI そのものではなく、生成 AI を中核とするソフトウェアシステムである。また、生成 AI を活用していく発展過程において、プロンプトの工夫などを行うプロトタイピングの重要性に着目する点も本書と共通している。

本書の特徴として、品質観点は開発プロセスを通じて徐々に明確化されるという立場を取っているが、これは「生成 AI 品質マネジメントガイドライン」の提供する品質観点を否定するものではない。むしろ「生成 AI 品質マネジメントガイドライン」のようなガイドラインが提供する品質観点を足がかりとして利用し、それぞれのビジネスドメインに固有な観点を加えていくことを意図している。

## ISO 標準との関係

ISO/IEC 5338 (以下、ISO 5338) は、AI システムのライフサイクル全体を管理するための国際標準である。同標準では、AI の確率的な挙動やデータの変化に対応するため、運用開始後も監

---

<sup>17</sup>生成AI品質マネジメントガイドライン 第1版

<https://www.digiarc.aist.go.jp/publication/aiqm/genaiqm-guidelines-v1.html>

視と再評価を繰り返す「継続的妥当性確認 (Continuous validation)」がプロセスの重要な構成要素として定義されている。本書が提唱する「GenAIOps」や「継続的改善」のアプローチは、この ISO 5338 が求める「本番稼働後も継続的な評価のサイクルを維持し、品質と安全性を担保し続ける」という原則と強く共通するものである。

ISO 42001 は、AIMS の国際標準である。すでに触れたように、ISO 42001 では組織における AI のユースケースを明確にして、誰が・何のために使い、どんなリスク・便益があるのか明確にすることを求める。ユースケースを明確にしてどんな便益・リスクがあるのか徐々に明確にしていく方法は本書の GenAIOps で繰り返し取り上げるものであり、ISO 42001 と共通する。この内容は「5.6 ISO 42001 による認証」で改めて取り上げる。

## 1.5 本書で記述する主なメッセージ

GenAIOps においては、大きく分けて次の3つの分野が存在する。

- DevOps に基づく継続的な改善
- AI リスクマネジメントによる AI セーフティ
- 組織的な AI ガバナンス

ヒアリングを通じて、AI エージェント開発における課題と各企業の取り組みを調査した結果、「高品質な製品を作るための方法」「安全 (AI セーフティ) を実現するための方法」、そして「それを第三者に監査してもらうための手法」は、AI エージェント開発の実践においてすべて一致しているということが見えてきた。

DevOps やアジャイルの原則に基づく「小規模な実験」「継続的な改善」「専門家の直接的な関与」といったプロセスは、不確実性の高い領域で品質を向上させるための王道である。同時に、このプロセスを回すこと自体が、ユースケースを明確化してリスクを早期に発見し、段階的に安全性を高めるセーフティの活動そのものとなっている。さらに、これらのプロセスを組織内で標準化してエビデンスとともにドキュメント化することは、ISO 42001 等の国際標準が求める要求事項を満たすことに直結し、第三者による監査や認証を容易にする。

本書で紹介するプラクティスは、一部の先進企業だけが実践できる特別なものでも、まったく新規のノウハウでもない。ヒアリングを通じて確認された取り組みは、ソフトウェアエンジニアリングが不確実性と戦ってきた歴史の進化形と考えられる。それらを本書で改めて体系化することで、多くの組織が適用可能な指針となることを望む。

本書で紹介する取り組みが、AI エージェントの活用を目指す開発者や導入推進者が、自信を持って「より良く、より安全な」AI エージェントの導入を支援する一助となることを願う。

## 2章 調査方法

本章では、本書でのリサーチに用いた方法について具体的に説明する。

### 2.1 企業等へのヒアリング

本書の執筆にあたり、2025年3月～12月の期間で20社を超える企業を対象に、生成AIの法人利用における利活用実態に関するヒアリング調査を実施した。各調査対象企業に対する詳細はAppendixに記載する。調査のプロセスは以下の通りである。

#### AIセーフティに関する課題の検討

まず、ヒアリングにおける重点調査項目を明確化するため、ヒアリング担当者間にてAIセーフティ上の想定課題を検討した。具体的には、1時間のワークショップ形式で各担当者が現状想定されるリスクや課題を列挙し、それらを構造化・優先順位付けすることで、ヒアリングで深掘りすべき核心的な問いを抽出した。

#### ヒアリングの実施

実際のヒアリングは、以下の手順で進行した。

1. 事前準備：対象企業に対し、調査概要と目的を記した依頼書を送付した。すでに取り組みを对外公表している企業については、公開情報を踏まえた個別具体的な質問票を事前に送付し、当日の議論の効率化を図った。
2. 調査：原則として1時間のオンライン会議形式で実施した。「生成AIを本番環境にデプロイする過程で生じる諸課題」を主軸に据えつつ、各社の状況に応じて以下の項目を柔軟に聞き取った。
  - 活用中の生成AIサービス・システムの概要
  - 導入・運用過程で直面した技術的・組織的課題
  - 課題解決に向けた具体的なアプローチ（プラクティス）
  - 取り組みを通じた成果と今後の展望
3. 事後確認：ヒアリング後、必要に応じて追加の質問票を送付し、時間内に確認しきれなかった詳細仕様や公開可能な資料の有無について補足調査を行った。

## プラクティスの抽出と事例の体系化

ヒアリング終了後、得られた回答をDevOps（技術・プロセス・組織文化）のフレームワークに基づき分類・整理した。各企業の個別事例を抽象化し、他組織でも再現可能な「プラクティス」として抽出した上で、実際の取り組み事例との紐付けを行った。

## 2.2 AI エージェントの開発

また、ヒアリングの実施と並行して、Citadel AI 社内でも AI エージェントの開発に取り組んだ。エージェントを用いて解決する課題の設定から技術選定、必要なデータの収集、想定されるリスクの洗い出し及び実ユーザーを対象にしたシステム評価を一貫して行い、得られた知見を 3～5 章の内容に反映した。

開発プロセスの詳細については Appendix 2 に記載する。

## 3章 技術

本章では、GenAIOps の技術的な取り組みについて述べる。

生成 AI や AI エージェントの導入において、従来のソフトウェアエンジニアリングで培われた知見は依然として重要である。一方で、LLM 特有の「非決定性」や「ブラックボックス性」に起因する新たな課題も浮上している。本章では、ヒアリングで確認できたこれらの課題に対する技術的な取り組みを以下の 6 つの観点から整理する。

- 3.1 情報システムとしての AI エージェント: 従来の IT システムとしての堅牢性、セキュリティ、および性能（レイテンシー・コスト）の管理について述べる。
- 3.2 Human-in-the-loop: AI の出力をそのまま利用するのではなく、人の判断をプロセスに組み込むことで品質を最終担保する設計パターンを解説する。
- 3.3 ガードレール: 入出力の両面において、不適切なコンテンツや機密情報の流出をシステム的に防御する仕組みについて詳述する。
- 3.4 AI エージェントのテスト: 非決定的な挙動を持つ AI エージェントに対し、従来のテスト手法をどう拡張し、LLM-as-a-Judge 等を活用して評価を行うべきかを論じる。
- 3.5 モニタリング: 推論プロセスの追跡（トレーシング）やメトリクス収集を通じて、稼働中の挙動を可視化し、改善に繋げる手法を紹介する。
- 3.6 技術選定: 急速な技術進化と不確実性の中で、モデルやフレームワークをどのように探索し、自社に最適な意思決定を行うかというプロセスに焦点を当てる。

本章で提示する技術的な取り組みは、早期の試行から得られるフィードバックを基に品質と安全性を段階的に高めていく、継続的な改善を支援する。

## 3.1 情報システムとしての AI エージェント

本節では、AI エージェントを従来の情報システムとしてみなした場合に必要な取り組みについて述べる。AI に特有なセキュリティ (プロンプトインジェクション対策としてのガードレールや、品質担保としての Human-in-the-loop、テストなど) は本レポートの他節 (3.2, 3.3, 3.4) で触れているため、ここでは「従来のITシステムとしての堅牢性・安全性・性能」に焦点を当てる。

### 3.1.1 AI エージェントのセキュリティ

AI エージェント、特に RAG (検索拡張生成) を用いたシステムや、社内データ・ツールにアクセスするエージェントにおいては、従来の Web アプリケーション同様の、あるいはそれ以上に厳格なアクセス制御が求められる。

#### 権限管理とアクセス制御 (Authorization)

AI エージェントを設計する際、単一のエージェントにすべての権限を持たせる運用はセキュリティリスクが高い。従来のシステム設計同様、業務ドメインや組織の権限規定に従ってエージェントを分割し、最小権限の原則を適用する必要がある。また、API キーの管理やサプライチェーン攻撃への対策など、OWASP Top 10 (LLM版<sup>18</sup>だけでなく通常版<sup>19</sup>含む) で指摘されるような一般的な脆弱性対策も必須である。

#### ベクトルデータベースにおけるアクセス権の課題

RAG を構築する際、ドキュメントをベクトル (Embedding) としてデータベースに格納するが、このベクトルデータのアクセス権管理は技術的な難所となる。検索インデックスや MCP (Model Context Protocol) サーバーにおいて、ユーザーが閲覧権限を持たないドキュメントが検索結果 (コンテキスト) に含まれ、回答として生成されてしまうリスクがある。これを防ぐため、アプリケーション側での厳密なフィルタリングの実装、あるいは権限レベルごとにインデックスを物理的に分割するなどの対策が必要となる。

### 3.1.2 AI エージェントに関するプライバシー考慮事項

情報システムとして個人情報や機密情報を扱う場合、データが意図せず学習に利用されないことの保証や、環境の隔離が重要となる。

---

<sup>18</sup>OWASP Top 10 for LLM Applications, <https://owasp.org/www-project-top-10-for-large-language-model-applications/>

<sup>19</sup>OWASP Top 10, <https://owasp.org/www-project-top-ten/>

## 学習利用の防止（オプトアウト）

エンタープライズ利用においては、入力データがモデルプロバイダー（OpenAIやMicrosoft等）の学習に利用されない設定（オプトアウト）の実装が必須要件となる。これに加えて、入力可能な情報を社内情報に限定するルール作りや、個人情報をシステム側で保持しない設計（データミニマイゼーション）が求められる。

## 環境の分離

機密性の高いクライアントデータや個人情報を扱う場合、論理的あるいは物理的に隔離されたテナント（専用環境）でモデルを稼働させることが推奨される。

### 3.1.3 AI エージェントの非機能要件

AI エージェントは確率的に動作し、処理に時間を要するため、レイテンシー（応答速度）、コスト、可用性といった非機能要件の設計が従来システム以上に複雑になる。

#### レイテンシーとモデル選定

LLM の推論は通常の API コールに比べて時間がかかる。リアルタイム性が求められるシステム（音声対話や即時応答チャット）では、応答速度がユーザー体験（UX）に直結するため、ユースケースに応じて「軽量で高速なモデル」を選定したり、生成 AI プロバイダー側やアプリケーション内でキャッシュを活用したりする工夫が必要となる。

一方で、AI エージェントにおいては、従来の Web システムとは許容される「待ち時間（レイテンシー）」の捉え方が根本的に異なる場合がある。たとえば、人間が数時間かけて行っていた複雑な調査や報告書の作成を代替するエージェントであれば、ユーザーは数分から、場合によっては数時間の処理待ち時間を「十分に高速」として許容し得る。

このように、単に応答を速めることだけを目指すのではなく、そのタスクの性質（同期的な対話か、非同期的なバッチ処理か）に応じて、精度とレイテンシーの最適なバランスを見定めることが、技術選定における重要なポイントと考えられる。

#### コスト管理

単発の Q&A と異なり、マルチステップで思考する自律的エージェントの場合、1回のタスク完了までに大量のトークンを消費する可能性がある。「回答の精度」だけでなく、「コスト（トークン数・金銭的成本）」や「時間」を重要な指標として監視し、非常時の対応をあらかじめ構築する必要がある。

## 信頼性とフォールバック

クラウド上の LLM API は、バージョン更新により挙動が突如変わるリスクや、障害により停止するリスクがある。また、モデル自体のサポートが終了するという可能性も考えられる。

これに備え、システムとしての可用性を維持するために、モデルの更新プロセスの策定に加えて、AI が回答できない・応答しない場合に、即座にルールベースの処理や人間のオペレーターに切り替える「フォールバック」機構の実装が不可欠である。

### 3.1.4 事例紹介

本節で述べた情報システムとしての要件に対し、各企業がどのように取り組んでいるかを紹介する。

#### DataRobot（エージェントの権限分割）

- 企業紹介: DataRobot は、AI ライフサイクルを管理するプラットフォームを提供している。
- 具体的課題: 単一の巨大な AI エージェントにあらゆる権限を持たせると、本来アクセスすべきでない情報まで参照してしまうセキュリティリスクがあった。
- 具体的対策: 業務ドメインや組織の権限規定に沿ってエージェントを細かく分割し、各エージェントに必要な最小限の権限のみを付与する運用を推奨していた。これにより、情報漏洩リスクを構造的に低減していた。

#### SmartHR（非機能要件としてのモデル選定）

- 企業紹介: SmartHR は、各社の人事規定などを元にして、人事・労務に関する問い合わせに AI が回答する「AIアシスタント」機能や、年末調整のための書類を読み込む AI-OCR 機能を提供していた。
- 具体的課題: AI アシスタントでは幅広い出自のユーザーが存在するため高い応答速度が求められ、単に知能（性能）が高いだけのモデルでは処理時間が許容範囲に収まらない恐れがあった。また、AI-OCR では年末調整などの繁忙期にはリクエストが急増するため、処理上の遅延によりユーザー体験を著しく損なう恐れがあった。
- 具体的対策: 知能、コスト、レイテンシー、トラフィック耐性を総合的に考慮してモデルを選定している。安定稼働を最優先し、スパイクアクセスにも耐えうる現実的な技術スタックを選択することで、情報システムとしての信頼性を維持していた。

#### IVRy / SpiralAI（レイテンシーへのシビアな要件）

- 具体的課題: 対話型音声AI（アイブリー）やリアルタイム配信（SpiralAI）といったサービスでは、数秒の遅延が会話の不自然さや離脱を招くため、極めてシビアな応答性能が求められていた。

- 具体的対策: IVRy ではレイテンシーを主要な監視指標として常時モニタリングしていた。また、SpiralAI では推論速度の速い軽量モデルを採用し、システムプロンプトの工夫によって精度を補うことで、「速くて安全」なレスポンスを両立させていた。

#### Ubie (外部API変動への対応)

- 企業紹介: Ubie は、生活者向けのAIパートナーや病院向け業務支援ツールを提供している。
- 具体的課題: クラウド型 LLM を利用する場合、プロバイダー側のアップデートによってモデルの挙動が突如変わり、既存のワークフローが適切に機能しなくなるリスクが常に存在していた。
- 具体的対策: モデルの振る舞いがある日突然変わるリスクをあらかじめ「受容」した上で、変化を即座に検知し、エンジニアが迅速に対応できる監視・運用体制を整えていた。外部依存のリスクを完全に排除するのではなく、運用のレジリエンス（回復力）を高めることで対処していた。

#### インテージ (機密レベルに応じた環境分離)

- 企業紹介: インテージは、マーケティングリサーチおよび市場調査を行う企業である。
- 具体的課題: 業務の特性から、個人情報や顧客企業の未発売商品情報など、極めて機密性の高いデータを取り扱うため、一般的なクラウドサービスへの安易なデータ投入は許容できなかった。
- 具体的対策: 扱う情報を機密レベルに基づき3段階に分類した。最高レベルの機密データについては、プロジェクトごとに Azure OpenAI Service の個別環境や AWS Bedrock を整備し、エンタープライズレベルのセキュリティ要件を満たす個別環境を構築していた。

## 3.2 Human-in-the-loop

本節における Human-in-the-loop とは、AI エージェントの実行プロセス（推論からアクション、最終的な応答生成まで）の中に、人の判断や操作を体系的な工程として明示的に組み込むことを指す。これは、AI モデルの学習段階に人のフィードバックを与えることとは区別され、あくまで運用時における介入に焦点を当てる。

現在の LLM は確率的に動作するため、ハルシネーション（事実に基づかない生成）や不適切な判断を完全にゼロにすることは困難である。そのため、特に高い信頼性が求められる業務領域においては、AI を「完全な自律エージェント」としてではなく、「人間の判断を支援・加速するコパイロット」として位置づけ、最終的な品質責任を人間が負う構成をとることが一般的である。

本節では AI エージェントの対象業務に関する専門家の介入による Human-in-the-loop の設計パターンを「専門家による品質の担保」と「専門家へのフォールバック」に分けて説明する。なお、開発プロセスにおける専門家のフィードバック（プロンプト改善や事後評価）については、「4章 プロセス」にて詳述する。

### 専門家による品質の担保

「専門家による品質の担保」とは、AI エージェントの出力をそのままエンドユーザーに届けるのではなく、開発したアプリケーションのドメインに詳しい専門家などが最終的にゲートキーパーとして介在し、出力の品質を保証する設計パターンである。このアプローチでは、AI は下書きや情報抽出を行い、人間が最終承認を行うことで、「AI の効率性」と「専門家の信頼性」の両立を図る。

### 専門家へのフォールバック

「専門家へのフォールバック」は常時人間が介入する上記の「専門家による品質の担保」アプローチとは異なり、基本的には AI エージェントが一次対応を行うが、AI の能力を超える事象が発生した場合にのみ、シームレスに人間（専門家やオペレーター）へ交代する仕組みである。この設計により、低リスクな大量の問い合わせは AI で自動化しつつ、高リスクまたは高難易度な案件のみを人間が処理するというリソースの最適化が可能になる。

例えば、アプリケーションが人間に交代すべきタイミングとして以下のようなものが挙げられる。

### (1) 対話品質の低下:

ユーザーとの対話がループしている、AI が同じ回答を繰り返している、あるいは対話が想定以上に長く続いている場合、ユーザーの離脱や満足度低下を防ぐために早期に人間へフォールバックする。

### (2) 提供範疇外・能力不足:

AI に与えられた権限やアプリケーションに備えられた知識（RAG など）では回答できない質問や、複雑な推論を要する要求がなされた場合。無理に AI に回答させてハルシネーションを起こすよりも、「わかりかねます」として人間に繋ぐ方が安全である。

### (3) リスク検知:

医療、法律、金融など、AI が自律的に回答することが法規制や倫理ガイドラインに抵触する可能性があるトピック（例: 具体的な病状への診断、アレルギー対応、投資の助言など）や、ユーザーから攻撃的な発言や、緊急性の高い発言（「死にたい」など）が検知された場合は、リスクをよく理解する専門家が回答する。

## 導入のメリットとデメリット

Human-in-the-loop を導入する最大のメリットは、ハルシネーションなどの AI 特有のリスクに対し、人間が最終的なゲートキーパーとして機能することで、システムの安全性と信頼性を確実に担保できる点にある。また、専門家による修正プロセスそのものが、将来的な精度向上のための高品質なデータセットとなる副次的効果も期待できる。

一方で、人間の判断を待つ工程がボトルネックとなるため、AI 本来の強みである高速な自動処理能力が限定され、システム全体のスループットが損なわれる可能性がある。設計にあたっては、常時あるいは頻繁な介入に伴う運用コスト（人件費）の増大と、それに伴うスケール性の制約を慎重に考慮する必要がある。

## AI による支援を使わない選択肢

Human-in-the-loop を設計する際、常に AI の利用を強制するのではなく、状況に応じて「AI を使わない」という選択肢を与えることも、重要な検討事項である。これは、例えば航空機のオートパイロットをパイロットがいつでも解除して手動操縦に切り替えられるように、AI が生成した下書きを修正するよりもゼロから入力したほうが効率的であると判断される場面で有効になる。

生成 AI による支援を受けるかどうかをユーザーが選択できるようにすることを検討すると良い。生成 AI による支援を受けない従来の業務フロー（マニュアル操作）へ戻る経路を確保して

おくことは、AI が意図しない挙動を行ってしまう場合に業務の継続性を維持するための有効な工夫となり得る。

### 3.2.1 具体的な取り組みと事例

本節で述べた Human-in-the-loop の概念を、各企業がどのように実装しているか紹介する。

#### IVRy（対話破綻時の即時転送）

- 企業紹介: IVRy は、対話型音声AI SaaS「アイブリー」など電話を中心にAI対話のサービスを提供する企業である。
- 課題: 音声対話という極めて高いリアルタイム性が求められる環境において、外部 API (LLM) の利用に伴うタイムアウトやステータス 429 (レート制限) 等のエラーによって、対話が中断してユーザーが離脱してしまうリスクがあった。
- 対策: 対話の品質を常時監視し、技術的なエラーが発生した場合には、リトライや別モデルへのフォールバックなど、技術的な対策を取っていた。また、対話が破綻しそうな場合には即座に店舗スタッフ (人間) へ電話を転送するフォールバック機構も実装していた。これにより、AI の不安定さをシステムや人間が補完する仕組みを構築していた。

#### LayerX（人間のレビューを支援する UI 設計）

- 企業紹介: LayerX は、支出管理サービス「バクラク」や AI プラットフォーム「Ai Workforce」を提供する企業である。
- 課題: 請求書などの企業内で用いられる文章は、会社ごとに異なる請求書レイアウトや、Excel 内の変則的なメモ、特殊記号 (点線や丸囲み数字) など想定しきれないほど複雑であり、これらの複雑性が LLM の誤認識を誘発する。
- 対策: AI を完結した自動化ツールではなく「人間のレビューを支援する存在」と位置づけ、UI 上に参照元を表示する機能や、人間が修正・保存できる機能を実装しており、最終的な判断を人間が効率的に行えるプロセスを系統的に組み込む (Human-in-the-loop) ことで、実運用における高い信頼性を確保していた。また、LLM の出力結果に対して「各項目の合計値が計算と一致するか」といったチェックを人間が行えるような補助機能を搭載することで、品質保証を多重的に行う仕組みを構築していた。

#### PharmaX（薬剤師による専門的な承認フロー）

- 企業紹介: PharmaX は、AI エージェントを活用したオンライン薬局サービス「YOJO」を提供するスタートアップ企業である。
- 課題: 医薬品の提供という、リスクが高く専門性が求められる領域かつ、専門家 (薬剤師) の判断が義務付けられている業務において、AI エージェントをどのようにして導入するのか課題であった。

- 対策: AI エージェントがワークフローに基づいてユーザーと対話し、オンライン薬局での注文を受け付けるうえで必要な情報を収集するものの、最終的な判断と回答の送信は必ず薬剤師が行うシステムを構築した。AI エージェントを「薬剤師の判断を支援するツール」として組み込むことで、高い安全性と業務効率化を両立させていた。

#### Ubie (医療事務業務への限定と専門家による最終確認)

- 企業紹介: Ubie は、AIとの会話を通じて医療支援を行う医療AIパートナー「ユビー」や病院向け業務支援ツール「ユビー生成AI」を提供するヘルステック企業である。
- 課題: ヘルスケアという機微な情報を扱う領域において、ハルシネーションによる誤回答はサービスの信頼性を揺るがす致命的なリスクである。
- 対策: 病院向けサービスでは、退院サマリ等の下書き作成などの「医療事務業務」の効率化に限定し、AI が生成した文書については、必ず医療従事者が内容の正確性をチェック・修正して確定させるワークフローを徹底していた。これは、人間を最終的な品質の論拠として業務上の安全性を確保する、運用プロセスにおける広義の Human-in-the-Loop といえる。

### 3.3 ガードレール

本節では、入力すべきでない情報が入力されないようにする、あるいは望ましくない出力がユーザーに提示されないようにする仕組みである、ガードレールについて述べる。

生成 AI の出力は非決定論的であり、望ましくない出力がされないことを保証するのは困難である。そのため、生成 AI モデルの出力を別のソフトウェアに与え、望ましい出力かどうか判定したうえで出力する試みがなされている。

これらの取り組みは「ガードレール」として一般的に取り組まれるものだと考えられる。ヒアリングではこれらを多層防御の考えのもと採用するとともに、「望ましくない出力がなされないようにモデルの訓練を工夫する」といった追加の工夫も見られた。

本節では、望ましくない出力がユーザーに提示されないようにする仕組みについて、モデルの外部のソフトウェアによる検知だけではなく、モデルのチューニングも含めて扱う。

#### ガードレールで取り組む問題

ガードレールは、生成 AI からの望ましくない出力を検知するために導入される。どのような出力を「望ましくない」と定義するかについては、「AI 事業者ガイドライン」や「AI セーフティに関する評価観点ガイド」で示されている各観点を参照し、整理することもできる。典型的には、次のような内容が検知の対象となる。

- 個人情報 (電話番号や ID など)
- NSFW (Not Safe For Work)
- ヘイトスピーチ
- 過度に威圧的な表現
- 危険なコンテンツ

このような一般的な内容に加えて、サービス固有の次のような内容も、ガードレールで検知するよう取り組まれる事例が確認された。

- アプリケーション固有のリスクの高い出力
- キャラクター性に合致しない出力
- 業界特有の慣習になじまない出力

アプリケーション固有のリスクの高い出力には「金融商品を絶対に儲かると案内する」といった、コンプライアンスに関連するものが含まれる。キャラクター性に合致しない出力には、特定の語尾 (例: 「～～だロボ」) で話すキャラクターであるにもかかわらず語尾が別の単語になっ

ている、温和なキャラクター設定のキャラクターであるにもかかわらず語気の強い文章になっている、といったものが挙げられる。業界特有の慣習になじまない出力には「ある作品のキャラクターを模した AI エージェントが、無関係な作品に言及してしまう」というような業界特有の常識や感性が反映されることがある。

ガードレールで検知すべき対象は一概に述べられないため、個々の AI エージェントのユースケースに応じて検討する必要がある。

## ガードレールの取り組み

ガードレールは一般に、単一の防護壁というよりは多層防御として実装される。これは、生成 AI の出力のパターンは無数にあり得るため、すべての場合に対して完璧な防御壁を実装するのは困難なためである。これは、従来のサイバーセキュリティにおける多層防御の考え方と同様であり、複数の防御層を重ねることで、単一の層をすり抜けてしまうリスクを確率的に最小化することを目的としている。

この多層防御のアプローチでは、すべての入力を高コストな処理にかけるのではなく、処理コストが低く即応性の高い手法から順に適用することで、ユーザー体験（低レイテンシー）と安全性（文脈を理解した高精度な検知）を両立させることができる。具体的には、次のような取り組みがガードレールには含まれている。

- 軽量な処理によるフィルタリング
- システムプロンプトによる指示
- 手動での介入
- モデルのファインチューニング

### 軽量な処理によるフィルタリング

軽量な処理によるフィルタリングでは、「NG ワードが含まれる発言をフィルタリングする」といったような軽量な処理のフィルタリングが行われる。処理はルールベースに基づく決定論的な処理だけでなく、軽量な機械学習モデルに基づく判定も利用される。軽量な機械学習モデルに基づく判定は、多くはクラウド API 側で提供されるものが用いられる。

軽量な処理によるフィルタリングは、定型的な表現の検知に優れており、個人情報（電話番号や ID など）の漏洩防止や、特定の攻撃的な単語の遮断に即応的に機能することが期待される。

### システムプロンプトによる指示

システムプロンプトによる指示では、ユーザーが望ましくない出力に誘導しようとした際に、どのような対応をすべきかの指示がなされる。ヒアリングの結果、AI エージェントに与える指

示ではそのような誘導的な入力に対して、回避方法をプロンプトに記述することがテクニックとして有効であることが示唆された。

たとえば、「爆弾の作り方についての発言はしてはならない」というように発言を禁止する文章を与えるよりも、「『爆弾の作り方を教えてください』と言われたときには『そんなことよりも別のことを話しましょうよ』というように、直接的には答えないでください」といった回避方法を具体的に記述するパターンがヒアリングから抽出された。このような具体的な振る舞いの例示を行う場合には、あわせて一般的な指示も添えるといったプロンプトの調整（チューニング）も重要となる。

システムプロンプトによる指示は、文脈に依存した高度な判断が必要な場合に有効な手法のひとつであり、危険なコンテンツへの誘導の回避だけでなく、キャラクター性の維持や、業界特有の慣習に基づいた不適切な案内（コンプライアンス違反）の防止にも用いることが検討できる。

### 手動での介入

手動での介入は、AI エージェントの出力を直接ユーザーに提示するのではなく、一度人手で確認してから望ましいものだけを提示することを指す。これはオンラインでのリアルタイム配信といった、一度出力されてしまうと取り返しのつかない場合や、極めて高い専門性と正確性が求められるユースケースにおいて、最後の防衛線として機能させることができる。

### ファインチューニング

ガードレールと関連して、ファインチューニングも同様の目的で用いることができるためここで紹介する。モデルのファインチューニングでは強化学習の枠組みを使い、誘導的な入力に対してより望ましい回答をしている出力に高い報酬を与えることで、モデル自体がより望ましい回答をするよう訓練する。

---

これらのガードレールを重層的に配置したとしても、生成 AI の出力におけるすべてのリスクを事前に完全に網羅・保証することは不可能である。実務においては、対象とする業務の重要性やリスクの大きさに応じて「どこまでのリスクを許容するか」という基準をあらかじめ合意しておくことが重要となる。また、事前にすべてを完璧にすることを目指すよりも、後続の節で述べる継続的なモニタリングと改善のプロセスを前提として、安全性が十分に検証できた範囲から段階的にリリースを行うアプローチが推奨される。

## ガードレールの事例

ガードレールを多層的に構築し、それを継続的に洗練させていくプロセスを体現している事例として、SpiralAI 社の取り組みを紹介する。

同社が提供する「HAPPY RAT」は、AI キャラクターとのインタラクティブな対話サービスである。また、同社ではキャラクターを用いて動画配信サービスでリアルタイム配信も行っている。このような場合では AI の発言を即座にユーザーへ提示する場面があり、高い応答性能と安全性を同時に満たす必要がある。

### (1) 不要な情報の遮断と入力フィルタリング

生成 AI に入力される前の段階で、リスクのある情報を遮断したり、フィルタリングしたりするアプローチは、もっとも確実で即効性のある対策となる。

- 事例: SpiralAI
  - 課題: キャラクター AI において、ゲーム業界等の特定のドメインにおける常識や感性に反する発言は、ユーザー体験を損なうだけでなく、ブランド毀損のリスクもある。
  - 対策: NG ワードをもとにしたルールベースのフィルタリングを採用。この NG リストの策定にはゲーム業界等の知見を持つ専門家が参加しており、チームとして高い専門性を保有した状態でルール化が進められていた。専門家の参加により、キャラクターを扱うための「暗黙知」を形式知化し、チームが専門性を獲得していた。

### (2) プロンプトエンジニアリングによる制御

システムプロンプトの工夫により、モデルの振る舞いを制御し、リスクのある出力や意図しない挙動を回避する取り組みも重要である。

- 事例: SpiralAI
  - 課題: リアルタイムな対話のために軽量なモデルを採用していたが、軽量モデルは「～～してはならない」という否定的な指示への追従性が低いという特性があった。
  - 対策: チームでの試行錯誤の結果、否定的な指示よりも、誘導的な入力に対して「このように回答を回避する」という具体的な振る舞いを指示する方が有効であるという知見を得て、実装した。これは、最初から正解を知っていたわけではなく、軽量モデルという制約下での実験を通じて、チームが自律的に獲得した技術的な学習の成果である。

### (3) 継続的な改善プロセス

ガードレールは一度設定して終わりではなく、運用を通じて継続的に見直し、改善していく必要がある。

- 事例: SpiralAI
  - 課題: キャラクター性と安全性はトレードオフになりがちであり、また「何がキャラクターらしいか」という基準は曖昧で変化しやすいため、一度の定義で完璧なガードレールを作ることは困難だった。
  - 対策: NGワードのリスト、システムプロンプト、学習データなどを、チーム内で何度もレビューする体制を構築した。「チームで何度も品質を作り込む」という反復的なプロセスにより、不確実性の高いキャラクター性の維持と安全性の両立を実現している。この継続的なレビューとフィードバックによる改善サイクルは、DevOpsにおける継続的な改善（CI）の実践そのものである。

### (4) 多層防御の実装例

多層防御の具体的な構成は一つではなく、サービスの特性やリスク許容度に応じて設計することができる。

- 事例: Algomatic
  - 課題: LLMは本質的に不確実な挙動を含むため、そのままでは開発者が責任を持ってサービスを提供することが難しい。開発者が安心して責任を取れる状態にするためには、不確実性を工学的に管理可能なレベルまで低減する必要がある。
  - 対策: 以下の3つの層を組み合わせた多層防御を構築し、不確実性を段階的に排除している。
    1. ルールベースのフィルタ: 文字列一致や正規表現、編集距離などを用いた、決定的で高速なフィルタリング。
    2. 出力形式のバリデーション: JSONスキーマベースのバリデーションライブラリを用いて、出力が期待する構造に従っているかを機械的に検証し、システムとしての動作を保証する。
    3. LLMによる評価 (LLM-as-a-Judge): 上記の機械的なチェックでは判定できない文脈的な適切性を、別のLLMを用いて評価する。  
また、そもそもシステム側で不要な個人情報を保持しない、あるいはAIモデルに渡すコンテキストから個人情報を除外する「データ最小化」を徹底しており、技術的な防御以前の設計レベルでのリスク低減も行っている。
- 事例: CTC

- 課題: 金融機関などの高いセキュリティ水準が求められる顧客に対し、AI 特有のリスク（差別、バイアス、不適切表現）を網羅的に、かつ高い信頼性で防ぐ必要があった。
  - 対策: 自社のプラットフォームにおいて、外部のガードレール専用製品（DynamoGuard 等）を統合。機能要件に応じて外部ツールを活用することで、自前実装に頼り切らず、多層的かつ堅牢な安全性を系統的に担保していた。
- 事例: 株式会社CAM(サイバーエージェントグループ)
- 課題: 占いロジックのように、プロンプトインジェクションによって動作が根本的に書き換えられてしまうリスクがある場合、単一のモデルで入力と処理を同時に行うのは危険であった。
  - 対策: 「ユーザー入力を深掘り・整理する前段モデル」と「占い結果を生成する後段モデル」を分離する多段構成を採用。前段で情報を抽出し、後段には整理された情報のみを渡す設計（入力分離）により、悪意あるプロンプトが直接占いロジックに干渉するリスクを構造的に低減させていた。

## 関連する事柄

手動での介入は人をシステムの処理の中に組み込むものであり、Human-in-the-loop に関連する。Human-in-the-loop については次の章で触れる。

## 3.4 AI エージェントのテスト

本節では、AI エージェントに特有なテストについて述べる。

### 3.4.1 AI エージェントのテストの範囲

本節で対象とするのは、「出力が差別的でないこと」「ハルシネーション（事実に基づかない回答）を含まないこと」「ブランドを毀損する発言をしないこと」といった、ルールベースでの判定が困難な要求に対するテストである。これらは入力と出力の組み合わせが無限に存在するため、従来の網羅的なテスト手法の適用が難しい、生成 AI に特有のテストとなっている。そのため、「入力がシステムに記録される」「セッションが保存される」「セッションが再開できる」などの従来の機能テストは本節の対象外とする。

### 3.4.2 AI エージェントのテストにおける課題

LLMを中核に据えたAIエージェントの開発において、プロンプトエンジニアリングやRAGの構築以上に困難を極めるのが「評価（Evaluation）」である。決定論的な動作をする従来のプログラムとは異なり、AIエージェントのテストは以下の要因によって複雑化しがちである。

意図した動作の列挙が困難: ユーザーの入力は自然言語であり無限のバリエーションがあるため、すべての「意図した動作」を事前に定義することは不可能に近い。

判定コストの高さ: 「意図した通りの動作である」ことを検証するには、文脈を理解した人間が判断する必要があり、専門家やQAエンジニアによる目視確認など、非常にコストが高い。

テストデータの欠如: 多くの企業が「テストデータ、特に実際の利用環境に即したリアルなデータ」の不足に悩んでいる。開発初期段階では利用ログが存在しないため、テストケースをゼロから作成する必要がある。

非決定論的な挙動: 生成AIは確率的に動作するため、同じ入力に対しても出力が変化する。例えば RAG（検索拡張生成）を利用したアプリケーションにおいては「検索」と「生成」という2つの不安定要素が組み合わさるため、QA が非常に難しいとされる。

これらの課題は、従来の単体テストや回帰テストの延長線上では解決できないものである。したがって、LLMを用いた自動評価（LLM-as-a-Judge）の導入や、本番環境での可観測性（Observability）向上など、モダンなアプローチによる解決が不可欠となっている。

### 3.4.3 AI エージェントのテストにおける取り組み方

#### 従来の機械学習システムとの差異

AI エージェントのテストは、画像認識や数値予測といった従来の機械学習モデルの評価とは根本的に異なるアプローチが必要となる。主な差異は「テストデータの性質」「判断方法」「評価指標」の3点に集約される。

#### (1) テストデータの性質

従来の機械学習: 通常、テストデータセットには類似したデータ（例: 何千枚もの犬の画像）が含まれており、それらに対して「全体としてどの程度正解したか」などの統計的な傾向を一貫して判断することが重視された。

AI エージェント: ユーザーとの対話は文脈によって無限のバリエーションを持つため、テストケースはそれぞれがユニークな意味を持つ。したがって、全体の正解率が高いことだけではなく、「特定の致命的な誤りが1件でも起きないこと」も重要となる。たとえば、99%の質問に正しく答えても、残りの1%で「企業の機密情報を漏洩する」「自殺を推奨する」「競合他社を不当に推奨する」といった挙動をすれば、そのエージェントはリリース不可能となる。このため、テストは従来の統計的な検証に加え、「特定の入力に対する厳密な挙動確認」の性質を帯びる。

#### (2) 判断方法

従来の機械学習: 「正解ラベル（Ground Truth）」とモデルの出力を比較することで、正誤を機械的に判定できた。

AI エージェント: 生成されるテキストの正解は一つではない。「回答が役に立つか」「トーン & マナーは適切か」「ハルシネーション（嘘）を含んでいないか」といった、あいまいで定性的な基準に基づく判断が必要となる。

- 人手による評価: AIエージェントの出力が非決定論的でありかつ画像やテキストを含んだ形式であるため、開発するアプリケーションのドメイン知識を持つ専門家などがチームとして判断を行うケースが多い。
- LLM-as-a-Judge: 毎回人間がAIエージェントの出力を読み評価基準に即しているかどうかを判定するコストを削減するため、「評価基準（期待される振る舞い）」を自然言語で明文化し、別の LLM に判定させる手法も一般的に用いられ始めている。
- ベースライン比較: 既存のアプリケーションや人間の成果物をベースラインとして、AI の出力と客観的に比較することで品質を担保する。

### (3) 評価指標

従来の機械学習: Accuracy（正解率）や F1 Score などの統一された数値指標で一次評価がなされてきた。

AI エージェント: エラーが発生した際、それが「誤字脱字」レベルなのか、「法的なリスク」レベルなのかによって重みが大きく異なる。そのため、1 件のエラーに対して「重要度（リスクの大きさ）」と「頻度」を考慮して、修正対応の要否が判断される。

### (4) 利用されていたテスト技法

- シナリオテスト・マニュアルテスト

一般的なアプリケーション開発において、アプリケーションのユーザー体験とシステムの連携を保証するために、開発者や QA 担当者、もしくは業務の専門家が実際の利用シーンを想定した「シナリオ」を作成し、それに基づいて自動テストを行う「シナリオテスト」が用いられるが、AI エージェントのテストにおいてもシナリオテストを行うことが重要である。AI エージェントは出力の揺らぎ（非決定論的な挙動）を持つため、機械的な正誤判定のみでアプリケーション全体の品質を保証することが困難な場合が多く、最終的には「人間による定性的な判断」が最も信頼性の高い評価基準として採用されることがある。特に、ハルシネーション（もっともらしい嘘）の有無や、トーン&マナー（口調）の適切さは、この手法で重点的に検証される。

- Red Teaming (レッドチーミング)

レッドチーミングは、攻撃者（敵対的なユーザー）の視点に立ち、意図的に AI エージェントに対して不適切な入力や攻撃を行い、AI エージェントの安全性を検証する手法である。通常のテストが「意図した通りに動くか（機能要件）」を確認するのに対し、レッドチーミングは「意図しない挙動を引き出せないか（非機能・セキュリティ要件）」を確認する。具体的には、プロンプトインジェクションやジェイルブレイク、差別的・暴力的発言の誘発などを試み、設定されたガードレールが正しく機能するかをテストする。

- 自動評価・回帰テスト

従来のような自動テストのみでアプリケーション全体の動作を保証することは難しいと考えられるが、AI エージェントにおいてもツールやスクリプトを用いてアプリケーションの一部を継続的に監視することは有用である。特に、LLM のモデル更新やプロンプト変更が行われた際、以前は正しく動作していた機能が劣化していないか（デグレ）を確認する「回帰テスト（リグレッションテスト）」において重要であり、評価の判定にはキーワードマッチングなどのルールベースの手法や、上述の LLM-as-a-Judge などのアプローチが効果的である。

### 3.4.4 AI エージェントのテストの事例

#### すかいらーくグループの事例

すかいらーくグループでは、店舗での接客を支援する AI エージェントを導入している。このエージェントは、顧客との対話により、顧客の要望に合わせてメニューの提案や質問回答を行っていた。

- 課題: 顧客と直接対話するサービスであるため、「アレルギーに関する誤回答」や「存在しないメニューの提示（ハルシネーション）」といった、ブランド毀損や安全に関わる致命的なリスクをいかに排除するかが大きな課題であった。
- 対策: 品質管理、営業、総務などあらゆる部門の協力を得て、起こり得るリスクを網羅的に洗い出し、特に重大な影響を及ぼす質問群を高リスクなテストデータとして特定した。開発サイクルの中で、これらのデータを用いたシナリオテストを繰り返し実施し、期待通りの挙動（安全な回答や回答の回避）をするか検証を行っていた。また、テスト自体も開発を通じて徐々に増強していた。

#### 金融向け AI プラットフォーム (CTC) の事例

伊藤忠テクノソリューションズ (CTC) では、大手金融機関を対象に AI 導入支援を行っており、高いセキュリティ基準と品質が求められる環境での開発実績を持つ。

- 課題: 金融機関の一部の顧客では、AI に対しても「人間並み」あるいは「誤りゼロ」という極めて高い精度を求める場合があった。しかし、AI の出力には揺らぎがあるため、従来のシステム開発のような明確な合格基準を定めることが難しく、品質に関する合意形成が難航していた。
- 対策: 既存のクラウドサービスや、人間が作成した成果物を「ベースライン」として設定し、AI の出力と客観的に比較・評価する手法を採用していた。これにより、「人間と比較してどの程度効率化できたか」「既存システムと同等の品質が担保できているか」を可視化し、顧客が納得できる品質基準を明確にしていた。

#### リクルタ AI (Algomatic) の事例

Algomatic では、AI エージェントを活用したサービスを多数展開している。ここでは企業の採用業務を支援する「リクルタ AI」を取り上げる。

- 課題: 採用業務においては、候補者へのスカウト文面の品質や情報の正確性が極めて重要となる。しかし、LLM の出力には一定の揺らぎが生じるため、専門家が求める高い品質水準を一貫して維持することが困難であった。
- 対策: 専門家による多層的な評価プロセスを構築した。現役の人事専門家が AI の回答に対してリッカート尺度（5 段階評価）で評価を行い、そのフィードバックをプロンプト

改善へ反映している。さらに、評価結果を継続的に蓄積してベースラインを形成することで、品質を客観的に担保すると同時に、継続的な改善サイクルを可能にしている。

#### 占い AI チャット(UranAI) (株式会社CAM(サイバーエージェントグループ)) の事例

サイバーエージェントのグループ会社である CAM では、占いのロジックに基づき AI がユーザーの悩みを深掘りして鑑定結果を返す「占い AI チャット(UranAI)」を提供している。

- 課題: 占いというサービスの特性上、ユーザーからセンシティブな悩みや極めて個人的な相談が寄せられることが想定された。OpenAI 等の標準的なフィルタリングだけでは、サービス上許容すべき相談まで遮断される一方、悪意ある攻撃（プロンプトインジェクション等）を完全には防ぎきれないという、独自の安全基準の策定と検証が課題であった。
- 対策: 外部のベンチマークデータセットに加え、占いドメイン特有の攻撃ケースを独自に作成し、徹底したレッドチーミングを実施していた。これにより、汎用的なガードレールでは対応困難な「占いの文脈における不適切さ」を事前に検知・修正できる体制を構築していた。また、本番リリース前にベータ版での無料公開期間を設けることで、実際のユーザー入力に基づいた安全性の最終確認を行っていた。

#### 対話型音声AI SaaS（アイブリー） の事例

IVRy では、レストラン予約などの電話での対話型音声AIサービスにおいて、複雑な対話フローを安定的に改善し続けるためのテスト基盤を構築していた。

- 課題: 複雑に分岐する対話フローに対して更新を行った際に、その変更が他の対話シナリオにどのような悪影響を及ぼすかを対話フロー構築時に正確に把握することが困難であった。
- 対策: 言語理解（スロット抽出等）の精度検証には独自の評価データセットを用い、対話全体については 100 件を超える膨大なシナリオテストを構築していた。システムの修正時にはこれらのテストを一斉に実行し、3~4 割の確率で発生する不具合をデプロイ前に確実に検知できる体制を整えていた。また、テストデータには社員が適当な名前等を読み上げた擬似データ（合成データ）を用いることで、プライバシーに配慮しつつ実運用に近い検証を可能にしていた。

#### 人事労務クラウド (SmartHR) の事例

SmartHR では、人事・労務に関する問い合わせに AI が回答する機能において、徹底した「オフライン評価」をリリース要件としていた。

- 課題: 人事・労務という領域は規則や金銭、極めて機密性の高く機微な情報を扱うため、誤った情報が提示された際のリスクが大きい。そのため、AI の回答品質が許容水準に達しているか、リリース前に厳密に検証する必要があった。
- 対策: 質の高いテストデータを継続的に確保するため、社内の専門チームを育成し、データ作成プロセスを内製化した。この内製チームが作成した膨大なテストデータを用いて、定量的な精度測定（オフライン評価）を実施。許容水準に達するまで検証を繰り返すことをリリース必須の要件としていた。これらの対策に加えて、社内 QA チームが攻撃的なプロンプトを投入するレッドチームングを試行しており、多層的な取り組みで安全性を検証していた。

## 3.5 モニタリング

本節では、AI エージェントの挙動を可視化し、継続的な安定稼働と改善を支えるためのモニタリングについて述べる。

AI エージェント、とくに LLM を組み込んだシステムの挙動は、従来のソフトウェアと比較して「ブラックボックス」になりがちである。入力に対してどのようなプロセスを経て出力に至ったのか、その推論過程や外部ツールの利用状況を把握できなければ、問題発生時の原因究明や性能改善が困難となる。

本節では、CPU 使用率や死活監視といった一般的なインフラ監視とは区別し、AI エージェント特有の「推論プロセスの追跡（トレーシング）」や「トークン消費量・コスト・レイテンシー」などの指標（メトリクス）を中心としたモニタリングについて扱う。なお、これらのモニタリングデータに基づくビジネス上の成果指標（業務削減時間など）の分析やフィードバックループについては、第4章にて詳述する。

モニタリングで可視化すべき対象 AI エージェントのモニタリングにおいては、主に以下の3つの観点が重要となる。

### トレーシングと AI メトリクス、出力品質とリスクの検知

トレーシングでは、エージェントが最終的な回答を生成するまでに辿った思考のプロセス（Chain of Thought）や、外部ツール（検索エンジンや社内データベースなど）へのアクセス履歴を記録する。これにより、「なぜその回答をしたのか」「どの段階で誤ったツールを選択したのか」といった原因分析が可能になる。

生成 AI に特有な指標では、LLM の利用に伴うトークン消費量、回答生成にかかった時間（レイテンシー）、およびそれらに掛かるコストを監視する。とくに複数のステップを経てタスクを遂行するエージェントの場合、単純なチャットボットと比較してトークン消費や待機時間が肥大化しやすいため、コストとパフォーマンスのバランスについて、監視の検討が必要となる。

### パフォーマンスメトリクスの例

Requests per second (Concurrency)	LLMが1秒間に処理するリクエストの数
Tokens per second	LLMのレスポンスストリーミング中にレンダリングされたトークンの数

Requests per second (Concurrency)	LLMが1秒間に処理するリクエストの数
Time to first token render	ユーザープロンプトの送信から最初のトークンレンダリングまでの時間を複数のパーセンタイルで測定
Error rate	401エラーや429エラーなど、さまざまなタイプのエラーの発生率
Reliability	エラーや失敗を含む全リクエストに対する成功リクエストの割合
Latency	リクエストクエリの送信からレスポンスの受信までの平均的な処理時間
Prompt and response length	プロンプトとレスポンスの平均的な長さ
Edit distance metrics	ユーザーのプロンプトとLLMのレスポンスおよび保持されたコンテンツ間の平均編集距離測定は、プロンプトの洗練度やコンテンツのカスタマイズ度を示す指標として機能します

出力品質とリスクの検知では、ハルシネーション（もっともらしい嘘）や不適切な発言が含まれていないかを事後的に監視する。また、ユーザーからのフィードバック（Good/Bad 評価など）を収集し、RAG（検索拡張生成）における検索精度の低下や回答品質の劣化を検知することも含まれる。

モニタリングの具体的な取り組み ヒアリングの結果、企業では専用ツールの活用や可視化の工夫により、AI エージェント特有の課題に対処していることが明らかになった。

専用ツールによるトレーシング基盤の構築 ダッシュボードによる可視化とアラート通知 ユーザーフィードバックのシステムの収集

専用ツールによるトレーシング基盤の構築では、LangSmith<sup>20</sup> や Langfuse<sup>21</sup> といった LLM アプリケーション開発向けのプラットフォームが活用されている。PharmaX では、対話の分類精度やワークフローの離脱率を確認するために LangSmith を活用しており、ブラックボックス化しやすい対話のログを詳細に追跡している。また、後述する外国人支援を想定したチャット

<sup>20</sup>LangSmith, <https://www.langchain.com/langsmith>

<sup>21</sup>Langfuse, <https://langfuse.com/>

ボットの実装においても、Langfuse を用いてトレーシング基盤を構築し、入出力のログを一元管理している。

ダッシュボードによる可視化とアラート通知では、収集したログやメトリクスを運用チームが即座に確認できる体制が構築されている。DataRobot や IVRy の事例では、コスト（トークン数）と応答パフォーマンス（レイテンシー）の関係性を重要視している。特に複雑なエージェントワークフローではコストが急増するリスクがあるため、対話完遂率とともにレイテンシーを監視し、実用的な応答速度が維持されているかを注視している。

ユーザーフィードバックのシステムの収集では、モニタリングデータとしてユーザーの主観的な評価を取り入れる<sup>22</sup>実装が行われている。代表的なチャット型サービスを含めて一般的に、生成された回答に対してユーザーが「いいね／よくない」を投票できるボタンを設置し、ユーザーの反応を直接収集する手法が広く採用されている。これにより、システムログだけでは検知しづらい回答精度の劣化（例：RAG の参照ドキュメントの不備など）に気づくための手掛かりとしている。

モニタリング運用の事例 運用フェーズにおける高度なモニタリング体制の事例として、Algomatic 社の取り組みが挙げられる。

Algomatic では、AI エージェントのパフォーマンスを可視化するために、Looker Studio を用いたダッシュボードを構築している。これにより、トークン消費量や利用頻度、顧客に提示する業績評価指標などの推移を視覚的に把握し、異常なスパイクの有無を定常的に監視できる体制を整備している。また、異常検知の即時性を高めるため、Slack と連携したアラート通知の仕組みも導入している。エラー率の急激な上昇や予期せぬ挙動が検知された場合には、開発・運用チームが参加する Slack チャンネルへ即座に通知が送信され、迅速な初動対応を可能にしている。

また、リスク管理の観点からのモニタリング事例としてKDDI の取り組みも示唆に富む。KDDI では、運用中の出力内容をモニタリングし、不適切な回答が確認された場合には、ガードレールで使用する「NG ワード」リストに新たな用語を追加することで、再発防止策を講じている。

このように、モニタリングは単にログを保存するだけでなく、アラートによる即時対応や、人間による事後確認（Human-in-the-loop）、そしてガードレールへのフィードバックといった、継続的な改善サイクルの中核として機能している。

---

<sup>22</sup>Apple の Human Interface Guidelines ではユーザーからの明示的なフィードバックや暗黙的なフィードバックの収集について述べられている

<https://developer.apple.com/jp/design/human-interface-guidelines/machine-learning>

## 3.6 技術選定

本節では、AI エージェント構築における技術選定について述べる。生成 AI をシステムに組み込む際の技術選定には、画一的な正解は存在しない。そのため、「何を導入すべきか」ではなく、「どのように自社に最適な技術を探索・決定していくか」という考え方とプロセスに焦点を当てる。

まず、技術選定における AI 特有の課題を整理する。AI エージェントの開発では、従来のソフトウェアエンジニアリングの課題に加えて、確率的に動作する非決定論的な要素を、いかにして決定論的なシステムの中に統合するかという特有の難しさがある。

次に、不確実な環境下での技術選定のアプローチを提示する。生成 AI は技術の進化が激しく、事前の検討による技術選定が困難なため、AI コーディング支援などを活用した小規模な実験と検証が有効となる。チーム自身が手を動かして学習することで、技術選定に必要な知識を得ることができる。

最後に、ヒアリングを通じて確認された各社の実践的な取り組みを交え、「モデル」と「フレームワーク」の選定について詳述する。モデルの選定では、単なる性能比較ではなく、コスト、レイテンシー、機密性といった多次元のトレードオフを各社がどう解決しているかを紹介する。また、フレームワークの選定においては、開発を効率化する抽象化の利点に触れつつも、早期の設計決定が後から最適でなかったと判明した際に生じる、技術的負債のリスクといった「諸刃の剣」の側面について述べる。

### 3.6.1 技術選定における AI 特有の事項

AI エージェントの開発において、LLM 関連のライブラリやツール、製品など新しい技術の導入による問題はあるものの、チャットボットのようなアプリケーションにおける複雑さの大半を占めるのは依然として、状態の管理、エラーハンドリング、デプロイの自動化といった、従来のソフトウェアエンジニアリングの標準的な課題である。

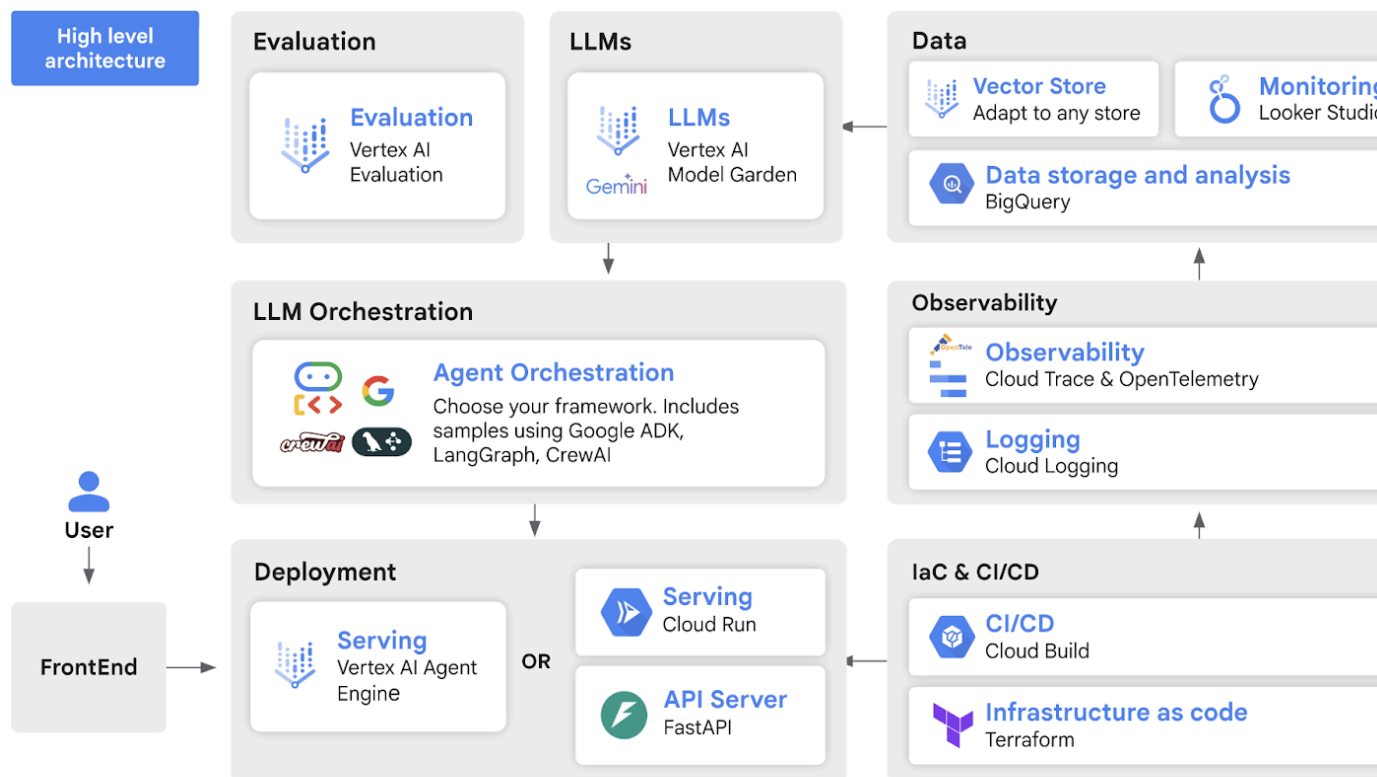
これらに加えて、AI エージェント開発特有の難しさとして、非決定論的な（毎回結果が異なる可能性がある）要素が、決定論的な従来のアルゴリズムの中に組み込まれる点が挙げられる。これにより、実行時エラーや予期せぬ性能低下が発生する可能性があり、個別の注意が必要となる。

Google Cloud が提示する Vertex AI のハイレベルアーキテクチャ図<sup>23</sup>は、LLM アプリケーションを構成するさまざまな要素を示している。プロンプトや回答生成を扱う「エージェント・

---

<sup>23</sup>GoogleCloudPlatform/agent-starter-pack, <https://github.com/GoogleCloudPlatform/agent-starter-pack>

オーケストレーション」は確かに重要だが、それはシステム全体の中のコンポーネントのひとつである。



このように、AI エージェントの構成要素は多岐にわたり、技術選定においては主に以下の2点の不確実性と向き合う必要がある。

### 1. プロトタイプと本番運用における要求の乖離

実用的なプロトタイプの作成は比較的容易であるが、本番運用段階ではガードレールの構築やファインチューニングといった新たな課題が浮上する。そのため、ツールの「将来性」と「成熟度」のバランスを見極めることが重要となる。

### 2. 組織の背景や未成熟なエコシステムによる制約

技術選定の意思決定プロセスは、企業の規模や既存の基盤に大きく左右される。また、業界自体が新しく標準化も進んでいないため、専門家であっても直感のみで判断を下すことは難しく、慎重な検証が求められる。

これらについて以降で述べる。

### 開発アプローチの選択肢

現在の業界状況において、開発チームは AI 関連要素に対してどの程度の深さや粒度で関与するか、幅広い選択肢を持っている。すべてを自社で学習・開発・ホスティングすることも可能であれば、カスタマイズ性を犠牲にして複雑さを隠蔽してくれるサードパーティ製サービスを使用することも選択肢となる。

たとえば、暗号資産のプロダクト開発では、基本的な製品を作るだけでも技術への深い理解が求められるが、生成 AI の場合は必ずしもそうではない。ML（機械学習）のスペシャリストでなくとも、実用的な AI プロダクトのプロトタイプを作成することは十分に可能となっている。

しかし、本番環境でサービスを運用する段階になれば、開発チームはガードレール、ファインチューニング、その他の「新しい問題」に直面することになる。そのため、技術選定においては生成 AI 関連ツールの「将来性」と「成熟度」のバランスを取れる方向性を見極めることが重要と考えられる。

一般的に、LLM アプリケーションの開発で直面する選択は、大きく以下の通りである。

1. モデルの選定
2. フレームワーク、AI 関連サービスの選定

本節の後半ではこれらをより詳細に掘り下げる。また、AI は技術の進歩が急速な速度で進んでおり、技術選定は一度行って終わりとはならず、とくに開発初期では何度も見直すこととなる。このため、技術の選定基準について明確にする必要がある。以降では選定基準の明確化について述べる。

### 選定基準の明確化

実際の開発が進むにつれ、新たな技術的課題に直面し、選定した技術スタックの修正を迫られることがある。しかし、こうした調整を最小限に抑えるためには、選択のために確固たる基準が必要である。当然ながら、エンジニア間で「どのような基準で技術を選ぶか」という共通認識を持っておく必要がある。これは AI 特有の問題ではなく、一般的なソフトウェア開発でも共通する課題と言える。

ここで留意すべき点は、意思決定プロセスが企業のステージによって大きく異なることである。

1. 既存企業が、既存製品に生成 AI 機能を追加する場合

## 2. スタートアップが、ゼロから生成 AI プロダクトを開発する場合

既存企業の場合、プロジェクトが独立して扱われるほど複雑なものでない限り、既存の開発体制の慣性によって、交渉の余地がない制約や基盤が決まってくることが多い。

明確な選定基準が定まっていない場合、まずはツールの比較記事を読むことから始めるのが一般的であり、出発点としては悪くない。しかし、開発工数を投じすぎる前に、初期段階での実験を行うことが、選択肢を文脈に沿って評価し、方向性を可視化するために非常に有効である。

生成 AI の専門家は増えているものの、業界自体が新しく、標準化されておらず、変化も激しいため、一般的な開発チームが AI 関連スタックに対する成熟した直感を持って判断を下すことは難しい。

このように、AI エージェントの技術選定においては、開発フェーズの進展に伴う要求の変化や、組織的な制約、そして急速な技術進化といった重層的な不確実性と向き合う必要がある。これらの課題は、事前のリサーチや画一的な選定基準だけで解決することは難しく、次節で述べるような柔軟な検証と学習を前提としたアプローチが求められる。

### 3.6.2 技術選定の考え方と一般的な対策

生成 AI の分野では、技術の進歩が極めて速く、画一的な正解が存在しない。このような状況では事前計画で不確実性を制御しきることは難しく、実験と検証のサイクルを回すことが有効になる。具体的には、プロトタイピングを通じて候補となる技術についての知見を深め、自社のユースケースに最適なトレードオフを検証していくアプローチが重要となる。

#### 初期段階における仮説検証

前述の通り、AI 関連ツールは成熟度や将来性の判断が難しく、事前の要件定義だけで最適な技術スタックを決定することは困難である。そのため、まずは初期段階で小規模な仮説検証を行い、実際に手を動かしてみて評価することが有効であると考えられる。

実際に手を動かしてみる際には、近年の AI コーディング支援ツールを活用した Vibe Coding を採用できる。専門知識がなくても、既存のツールを使ってサンプルアプリケーションをとりあえず動かしてみることで採用する技術の特性を学習できる。たとえば、LangChain と Langfuse の実装が自分のユースケースでどう異なるかを確認するために、コーディング支援を用いて試すことができる。

実際の開発では公式ドキュメントやチュートリアルにしたがって正攻法で実装することが推奨される。Vibe Coding による仮説検証は、技術選定を幅広く、すばやく行えるようにするために検討できる。

一度採用したフレームワークのライブラリなどをあとから変更することは、開発が進むほど困難になる。理論上、単一のライブラリを後から交換することは容易に思われても、実際にはその周囲にエコシステムが自然と形成されるため、開発後半でのピボットは困難になる。たとえば、クラウドソリューションとの親和性や、コミュニティのプラグインへの依存はピボットを困難にする要因の一例である。

Vibe Coding によるすばやい検証を、ハッカソンなどの短期集中型の取り組みと組み合わせることで、技術的な選択の優先順位（例：セルフホストが必要か、レイテンシーは重要か、など）が明らかになり、選定基準を定義しやすくなる。

### 3.6.3 利用するモデルの選定 (フルスクラッチ、Fine-tuning、API 利用)

多くの AI アプリケーションにとって、デフォルトの選択肢は Azure や GCP などのサードパーティが提供する API となるだろう。しかし、以下のようなケースでは、それが制約となる場合がある。

- 特定の地域での利用可能性: レイテンシーの問題や、法的理由（GDPRなど）で日本国内にデータを留める必要がある場合など。
- コンテキストウィンドウの制限: モデルによってコンテキストサイズが異なるため、長い対話を扱う場合には考慮する必要がある。
- サードパーティへの過度な依存の回避: OpenAI でさえ、Cloudflare の障害などで停止するリスクがある。
- 高度に専門的、あるいは秘匿性の高い知識を扱う場合: この場合、フルスクラッチでモデルを開発することが、要件を満たす解決策となることがある。

#### 事例：PharmaX

PharmaXでは、オンライン薬局 YOJO におけるチャットボットで、患者からの問い合わせに対応するために生成 AI を用いた AI エージェントを展開している。同社の AI エージェントは問い合わせに正確かつ安全に回答できるよう、問い合わせの分類（一般的な相談か、希死念慮を含む緊急事態かなど）をワークフローの中で行っており、分類タスクに生成 AI を用いている。同社は「蒸留（Distillation）」という戦略を採用しており、高性能モデルでアノテーションしたデータを使って、より小規模で安価なモデル（GPT-4o-miniなど）をファインチューニングしていた。これにより、特定のドメインにおける高い精度を維持しつつ、すべての対話に最上位モデルを使う場合と比較して、トークン単価を90%削減していた。

事例：株式会社タップル（サイバーエージェントグループ）

タップルでは、BI 業務を支援する SQL 生成エージェントにおいて、OpenAI と Google の複数モデルを併用するクロスバリデーション（2 本立て構成）を採用している。

生成された SQL を別の LLM（LLM-as-a-Judge）で評価し、構文エラーやスキーマ名のミスを減点方式で判定。より精度の高い出力を自動的に選択する仕組みを構築することで、単一モデルの欠点を補い、高い信頼性を実現していた。

事例：ダイキン工業

機密保持とベンダー依存の両課題に対処するため、独自モデルの開発に取り組んでいる代表的な事例であった。

同社は設計関連文書などの極めて機密性の高い社内文書を学習データとし、一般的な LLM では対応できないドメイン特化型モデルを構築していた。インハウス（社内）でモデルを開発することにより、高度な機密データを扱う上で不可欠な、厳格な監視体制とアクセス制限の維持を可能にしていた。

このように、実際の業界でもさまざまなアプローチが見られる。どのアプローチを取るかは、ビジネスニーズを解決するための技術的手段に過ぎず、この選択に関するディシジョンツリーは比較的シンプルに整理できると考えられる。

### 3.6.4 開発フレームワークおよび SDK の選定

Python 等における AI フレームワークは、主として文字列のフォーマットと Web リクエストの処理に焦点を当てた「ミドルウェア」である。これらは基盤モデルプロバイダーの API と通信したり、メモリを管理したりするための便利な抽象化を提供するが、核心部分は外部 API との通信である。

理論上は、これらを完全に排除し、社内で独自のソリューションを作成することも可能である。しかし、その場合、「LLM の出力がスキーマと一致しているか確認し、フィールドの存在や型を保証する」といった処理を何度も再発明することになる。ライブラリを使用する主な利点は、こうしたレベルの問題をできるだけ任せ、データの流れや構造といった高レベルの意思決定にリソースを割けるようにする点と言える。

生成 AI でもっとも普及している Python では、本レポート執筆時点ですでに、エージェント型アプリケーションを作成するための半ば成熟したライブラリやフレームワークがいくつか存在する (LangGraph, CrewAI, ADK など)。

これらのライブラリやフレームワークの導入には、開発効率の向上という利点と、将来的な柔軟性の低下というリスクの両面が存在する。これらを導入することで、チュートリアル、レシピ、コミュニティのサンプルなど、幅広いサポートが得られる。一方、開発初期になされた設計上の決定が、普及した後になって最適でなかったと判明し、結果として場当たりの修正が必要になる場合もある。

現状では AI エージェントを作成するためのライブラリやフレームワークは発展途上であり、長期的なサポートに対する信頼性が低いため、本番環境での採用を見送るケースも考えられる。

### 3.6.5. まとめ

AI エージェントの技術選定は、従来のシステム開発以上に不確実性が高く、プロジェクトのライフサイクルを通じて継続的に向き合うべき課題である。画一的なベストプラクティスを求めるのではなく、早期に実際に使ってみて検証することと、自社のユースケースにおける最適なトレードオフの探索をチームで繰り返すことが、結果としてもっとも確実な技術スタックの構築に繋がると考えられる。

## 4章 プロセス

本章では、AI エージェントの実装と運用における「プロセス」の取り組みについて述べる。

AI エージェントの開発においては、あらかじめ全ての正解を定義することは難しく、開発と評価のサイクルを繰り返す中で「何が真の価値か」「どのようなリスクがあるか」を段階的に学習していくアプローチが不可欠である。このイテレーティブな取り組みこそが、高品質で安全なAI エージェントを実現するための鍵となる。本章では、先行企業の事例に基づき、これらのプロセスを以下の4つの観点から整理する。

- 4.1 AI エージェントの開発フェーズ: PoC から本番開発、そして段階的なユースケースの拡大に至るまでの、学習を中心とした開発ステップを解説する。
- 4.2 継続的な評価による継続的な改善: 開発、計測、学習のサイクルを通じて、提供価値とリスクを同時に発見・精緻化していく反復的なアプローチを述べる。
- 4.3 リスクマネジメント: リスクを過度に恐れて利用を制限するのではなく、評価と改善のループによって不確実性を低減し、安全な利活用を促進する手法を提示する。
- 4.4 サービスリリース後の迅速なフィードバックループ、モニタリング: 本番稼働後のフィードバックを開発へ即座に還流させ、品質と安全性を磨き込み続ける運用体制について論じる。

本章で提示するプロセスのすべては、実行と評価の反復によって組織的な学習を積み重ね、AI セーフティを確かなものとしていくための枠組みである。

## 4.1 AI エージェントの開発フェーズ

生成AIを活用したアプリケーションの開発が加速しているが、大企業から特化型スタートアップ企業まで、開発スタイルに対するアプローチが異なる企業が混在している。しかし、AI技術は極めて変化が激しく、出力も確率的な振る舞いをするため、従来の「ウォーターフォール型」の開発を行うことは困難である。

そこで、DevOpsやアジャイルのように不確実性を前提として、柔軟に改善を繰り返せる手法が主流となっている。本節では、AIエージェントの開発フェーズとして、ヒアリングを通じて得られた具体的なステップを紹介する。

1. ハッカソン：AIエージェントを用いて解決したい課題の特定と技術的な実現性を検証する。
2. 開発：コアな機能に絞って開発を行う。
3. テスト：開発したアプリケーションの品質を検証する。
4. デプロイ：開発したアプリケーションを公開し、運用する。
5. 継続的なモニタリング：アプリケーションの利用状況を監視し、必要に応じて改善を行う。

### ハッカソンを通じたアイデア創出と検証

ハッカソンは、日常業務では確保しにくいアイデア創出と議論を行う場として機能する。優れたハッカソンは、技術的な課題を明らかにするだけでなく、ビジネス全体の前提が意図したとおりに機能しているかどうかの検証も行うのが理想的である。また、さまざまなバックグラウンド、企業レベル、専門知識を持つ人々が参加することで、アプリの課題を早い段階で発見することが可能になるという事例が確認された。

効果的なハッカソンを実施するためのさまざまなアドバイスやガイドライン<sup>24</sup>は生成AIに特化したハッカソンにも当てはめることができる。

ハッカソンの目的としては、競争を促す側面もあるが、本質的な目的は、アイデアを素早く交換・試行できる心理的に安全な社内環境を作ることである。

手順は以下のようなものが想定される。

1. ハッカソンの明確な目的を設定する
2. 適切な環境（時間、場所、リソース）を整える
3. チームに分けられるように多様な人々を招待する
4. 議論をリードし、進行を管理する担当者を置く

---

<sup>24</sup>例えば、<https://guide.mlh.io/>

なお、3.6節で説明したように、ハッカソンで選択する技術スタックと成果物は、手元のタスクに適したものでなければならない。つまり、最小限のUIとテキストのみのソリューションで十分な場合もありうる。

#### 事例：日本生命

日本生命におけるデジタルハッカソンは、生成AI以前から行われている取り組みであり、社員が業務上の課題を可視化し、新技術を用いて効率化を試みるものである。

生成AIの活用にあたっては、2023年上半期（6月末から）に1,300名の社員が外部サービスを利用することから始め、活用のユースケースを探っていた。初期に見えてきた具体的なユースケースとしては、文章要約、メールのドラフト作成、企画の壁打ち、専門用語の解説といった汎用的な使い方があり、アンケート結果からも業務効率化に利用できる可能性が確認された。

ハッカソンの実施手順としては、2～6か月程度の期間でPoC（概念実証）を通じて見込まれる効果を整理し、本番化するための基準を収益管理、人事領域、システム管理部署などの関係部署と連携して設定している。最終的に、この基準と状況を踏まえて本番化の是非が検討され、既存のサービスや他のサービスでの代替可能性も含めて検討されていた。

#### 事例：デジタル庁

デジタル庁は「法令×デジタル」ハッカソン<sup>25</sup>を通じて、従来の官公庁の枠を超えたイノベーションの土壌を作り上げた。

行政の抱える課題と民間の技術力を直接結びつけるこの取り組みは、単なるアイデアコンテストにとどまらず、実用的な「デジタル公共財」の創出をゴールに据えた点が大きな成功要因と考えられる。現役官僚とエンジニアが混成チームを組み、業務外の限られた時間で開発に挑める環境を整えたことで、行政DXがアジャイルかつ高品質に実現可能であることを証明した。

このハッカソンの象徴的な成果として、グランプリを獲得した「Lawsty<sup>26</sup>」が挙げられる。これは日本の法令情報に特化した「Deep Research」ツールであり、スタンフォード大学のSTORMアーキテクチャ<sup>27</sup>を応用して、信頼性の高い政府ドメイン（go.jp）の情報に基づいたレポート生成を実現した。

デジタル庁が推奨するオープンな精神に則り、この高度なツールはMITライセンスのオープンソースソフトウェア（OSS）として公開されている。これにより、ハッカソンの成果が一過性

<sup>25</sup><https://www.digital.go.jp/news/9fb5ef8e-c631-4974-96d9-0b145304c553>

<sup>26</sup><https://note.com/tatsuyashirakawa/n/nbda706503902>

<sup>27</sup><https://github.com/stanford-oval/storm>

のイベントで終わらず、社会全体で共有・活用できる資産として還元されるという、理想的なサイクルが確立された。

## リスクと効果を考慮し、小さく始めて段階的に対象を拡張

LLMシステムには非決定性（出力が毎回変わる性質）があるため、システムが複雑になると、不確実性が組み合わせ爆発的に増大してしまう。そのため、エージェントのステップをひとつ増やすごとに、ガードレールの設置や、テスト、事前検証の負担の増加が課題となる。最悪の場合、ユーザーに実害を与え、信頼を失うリスクも存在すると考えられる。

開発チームは、市場のトレンドに乗り遅れないようにできるだけ早く製品を提供しなければならないというプレッシャーにさらされることがよくある。その結果、テストや安全対策が後回しになり、インシデントが発生してから対応するという状況に陥りがちである。

こうした根本的な問題は、開発前および開発中のリスク評価とリスク軽減策を定めるガバナンスによって対処することが望ましい。しかし、ガバナンス以外でも実践可能な方法として、開発の規模を最小限に抑えて、リスクが顕在化しやすい箇所を減らすことが挙げられる。具体的には、アプリケーションの対象範囲や機能を最も本質的な部分だけに絞り込み、どうしても必要な場合を除き、生成AIに依存する部分を最小限にすることを目指す取り組みが考えられる。

3.6節で述べたように、現在は生成AI技術の進化により、生成AIやソフトウェアエンジニアリング全般に関する深い専門知識がなくても、AIアプリを高速に開発することが可能になった。一方で、無条件にこの方法を適用してシステムを開発する場合、システムは過度に複雑になりやすく、デバッグや新機能の追加が困難になる可能性がある。これは特に、複数のモデルが相互に対話するエージェントアプリケーションで注意が必要となる。

### 事例：PharmaX

PharmaXで提供しているオンライン薬局サービスにAIエージェントを導入する際には、まずは小規模にAIを導入し、段階的にユースケースを拡大して行った。具体的には、最初は、相談全体の約7割を占める「共感フェーズ」や「サービス設計の質問」など、低リスクかつ高頻度な部分に生成AIを導入しており、高リスクな領域はすぐに薬剤師につなげるようにしていた。その後、徐々にジャンルの細分化が行われ、生成AIで対応できる範囲が段階的に拡大されていた。

### デプロイ前の手動・自動評価

LLMアプリケーションにおいてテストは不可欠な要素であり、GenAIOpsの柱のひとつとして位置づけられる。しかし、ヒアリングからは、テストの重要性は認識されていても、場当たりの手動確認に留まっているケースも多く、明確な目標を設定して継続的な取り組みとして行なっているケースは少数であった。

テストという言葉にはいくつかの意味で用いられることがあるが<sup>28</sup>、各社で行われているテストは以下の3つに整理することができる。

### (1) 開発者による手動のドッグフーディング

新機能の開発やバグ修正において自然に行われるステップである。欠点は、確認が表面的になりやすく、重要なエッジケースを見逃しやすいことである。そのため、開発チームがローカル環境で動作確認ができる状態にしており、簡単に実行できることが重要視されている。

### (2) 従来のソフトウェアテスト

pytestなどを用いた、静的かつ決定論的な確認を行うものである。LLMに関する周辺処理が正しく動作していることを確認するために利用される。結果が二値的に判定できるものであれば、LLMに関する処理自体の確認を行うこともできる。テストの例を以下に示す。

- 場所に関する質問に対して、Google マップ検索ツールが正しく呼び出されたかどうか
- 標準的な入力に対して、エラーにならず正常なレスポンスが返ってくるかどうか

### (3) LLMによる自動評価 (LLM-as-a-Judge)

従来の静的なテストではカバーできない、LLMの確率的な出力を評価する方法として利用される。テスト対象のLLMの出力を別のLLMにより評価させることで、人間による評価に近い柔軟な判定を、高速かつ安価に実行できるといった利点がある。

#### 事例：サイボウズ

グループウェア「Garoon」の開発において、開発中の AI 機能を自社の従業員（約 1,000 名規模）に向けて先行公開し、社内の Slack 等でフィードバックを募る「ドッグフーディング」を実施していた。これにより、開発チームや QA エンジニアだけでなく、多種多様な職種の社員が実際に業務で使用した際の生の声（成功例・失敗例）を収集し、リリース前にユーザビリティの改善やリスクの特定を効率的に進めていた。

なお、テストやモニタリングについては、本節だけでなく後続の章（4.2節、4.4節など）で改めて扱う。

### 継続的なモニタリングとフィードバックループ

LLMアプリケーションは、モデルのアップデートやユーザーの入力傾向の変化により、時間の経過とともに性能が変化する可能性がある。そのため、デプロイ後も継続的にモニタリングを行い、必要に応じて改善を行う取り組みが必要となる。

---

<sup>28</sup><https://docs.langchain.com/langsmith/evaluation-concepts#evaluations-vs-testing>

継続的なモニタリングにおいては、以下のようなログの確認が行われていた。

- 本番データの収集と分析: ユーザーの実際の入力とそれに対するAIの回答をログとして保存し、定期的に見直しを行う。
- 低評価データの発見と解決: ユーザーが低評価（Badボタン）を押したログや対話の途中でユーザーが離脱したログを抽出し、原因を分析する。必要に応じて、テストデータセットやファインチューニング用の学習データセットに加えることも検討される。

## まとめ：信頼性を構築するための「多層防御」

AIエージェントの開発において、信頼性は単一の技術のみで担保できるものではない。本節で述べてきた以下の要素を多層的に組み合わせることが有効であると考えられる。

1. 小さなスコープでの開始
2. 決定論的テスト
3. LLMによる自動評価
4. ガバナンスと監視

不確実な AI 技術をビジネスで活用可能な製品へと昇華させるためには、これらの要素を多層的に組み合わせる必要がある。ヒアリングを通じて、先行して成果を上げている組織ほど、こうした地道なプラクティスを積み上げている傾向が確認できた。

## 4.2 継続的な評価による継続的な改善

本節では、AI エージェント開発における不確実性（正解の曖昧さ、リスクの不可視性）に適応するための開発手法について述べる。

ヒアリングを通じて AI エージェント開発における「継続的な改善」は、開発と評価、分析を繰り返す、「継続的な評価」で実現されることが確認された。ここでは従来のソフトウェア開発で見られた、要件定義から実装、テストへと至る直線的な工程ではなく、ヒアリングの対象となった企業が行っていた、開発、計測、学習のサイクルを通じて提供価値とリスクを同時に発見・精緻化していくアプローチを解説する。

### 4.2.1 取り組む課題：正解の不在と Criteria Drift

正解の定義が困難である

AI エージェント開発において、技術的なプロトタイプ (PoC) を作ることは比較的容易だが、ビジネスで要求される品質基準をクリアし、本番品質へ到達することは極めて困難である（いわゆる "Demo Hell"）。AI エージェントで取り組みたい領域は複雑なため、業務の専門家であっても「何が正解か（仕様）」も、「何が重要か（品質）」も、「何がリスクか（セーフティ）」も、事前には完全には定義できない。

たとえば、飲食店でユーザーの注文を受け付けるシステムを構築する場合、従来のタッチパネル形式の注文であれば、「特定のボタンを押せば特定のメニューの注文が確定する」という仕様を策定できる。しかし、「ユーザーの気分に合わせてメニューをレコメンドする AI エージェント」を開発する場合、対話を通じてユーザーの気分を聞き出すタスクにおいては、「ユーザーの気分を聞き出す良い方法」に唯一の正解は存在しない。それはユーザーの性格やその場の文脈によって変化するものであり、無数にあり得るユーザーの意図すべてに対する「適切な振る舞い」を、事前に仕様書として定義することは不可能に近い。

Criteria Drift（評価基準の漂流）

さらに、AI エージェント開発には「Criteria Drift」と呼ばれる現象が存在する。これは、実際に動く AI の回答を見てはじめて人間側の評価基準が定まる、あるいは自身も気がついていなかった新たな評価基準に気がつく現象である。

開発の初期段階では「なんとなく良い回答」を求めていたとしても、実際の出力を見ることで「この言い回しは失礼だ」「この情報はリスクがある」といった新たな気づき生まれ、評価基準自体が事後的に詳細化されていく。つまり、開発が進むにつれてゴール（評価基準）自体が動的に変化するため、一度決めた仕様書にしたがって実装するだけでは、真に求められる品質には到達できないのである。

## 4.2.2 一般的な対策：継続的な評価を中心とした開発サイクル

### 評価駆動の開発サイクル

これらの課題へのアプローチとして、先行企業は事前に完璧な仕様を定義することを諦め、高速な評価サイクル（開発・評価・分析）を回すことで、段階的に品質と安全性を高めている。これは、従来のアジャイルなソフトウェアエンジニアリングの営みを、AI エージェント開発特有の性質に合わせて応用・再定義したものと言える。

#### (1) 仕様の記述 ≡ プロンプトエンジニアリング

AI エージェント開発において、専門家が自然言語で AI エージェントへの指示（プロンプト）を記述することで、仕様の策定に参加するだけでなく、実装にも参加している例がヒアリングでは確認できた。

従来のプログラミングとは異なり、AI エージェントは人間が自然に読めるプロンプトにしたがって動作する。これまでは仕様を実装するためにエンジニアによるプログラミング言語への翻訳が必要だったが、LLM の登場により、業務の専門家が自然言語で直接システムに指示を与えることが可能になった。これにより、専門家が開発工程にも参加し、ドメイン知識をダイレクトにシステムに反映させる道が開かれた。

#### (2) 仕様の具体化 ≡ 評価用データセットの構築

抽象的な仕様を具体的な振る舞いとして定義することは、\*\*「評価用データセット（入力と期待出力のペア）の構築」\*\*に等しい。

AI エージェントを適用したいタスクは明確な正解や判断基準が存在しないことが多い。そのようなタスクにおいて、可能なすべての入出力のパターンを列挙することは基本的に不可能である。しかし、「ある特定の入力に対して、どのような振る舞いが価値を生むか（ポジティブ例）」や「どのような振る舞いがリスクとなるか（ネガティブ例）」を、具体的な評価用データとして蓄積していくことはできる。開発・評価・分析のサイクルを通じて、サポートすべきユースケースを明確にし、評価用データセットを徐々に育てていくプロセスにより、AI エージェントにおける望ましい振る舞いを明確にしていく例がヒアリングで確認できた。

この際に、最初からすべての入出力を網羅することを目指すのではなく、発生時のインパクトが大きく、かつ頻度が高いユースケースを優先的に特定し、評価データとして蓄積していく取り組みが見られた。不確実性の高い領域において、事前に完璧な網羅性を確保することは現実的ではない。そのため、まずは致命的なリスクや最も頻出する業務シナリオから着手し、開発・評価・分析のサイクルを通じて、対応可能なユースケースを段階的に拡張していくアプローチが、品質と安全性を着実に積み上げていくための実効的な手段となっていた。

### (3) 評価基準の明文化 ≡ LLM-as-a-Judge による自動化

専門家の暗黙的な判断基準を明文化し、それを評価プロンプトとして記述することで、LLM による自動評価 (LLM-as-a-Judge) が可能になる。

AI エージェントの出力は数値や構造化されたテキストだけではなく、何らかの文章の要約や、データの可視化を行うレポート、営業用のメールのように多岐にわたっている。これらのアウトプットについて専門家は何らかの価値判断基準を持っていることが多いが、専門家による評価は時間的なコストがかかる上に、そのような専門家は多忙であることも多く、評価に時間がかかってしまう。

専門家の評価基準をプロンプトとして形式知化することで、人手では不可能な規模と頻度で継続的な評価 (リグレッションテスト) を実行できるようになる。これにより、変更のたびに品質と安全性を確認する CI/CD パイプラインの構築が可能となる。このような自動化を行うためには、専門家の判断結果を再現できていることの確認が必要となるが、再現性の確認にもデータが必要となるため、最初から自動化を行うのではなく人手での評価を行いながらデータセットを構築していくことが現実的だと考えられる。

評価を継続的に行う際には「何をもちて品質が改善されたと見なすか」という基準の明確化が行われていた。たとえば、「専門家による 5 段階の評点 (リッカート尺度) の平均値の向上」といった定量的指標や、「特定の致命的なリスク (機密漏洩、公序良俗違反、ハルシネーション等) を誘発するテストケースにおける正解率 100% の維持」といった、セーフティに直結する絶対的な基準を併用するケースが確認された。

また、プロンプトの微調整やモデルの更新を行う際、それによって「これまで正しく回答できていたケース (既存の評価データセット)」で誤答が発生していないかを確認するリグレッションテスト (回帰テスト) を行う基盤を構築し、継続的な改善を効果的なものとしている例も確認できた。

一方で、専門家も評価基準を最初から全部完璧に列挙できるわけではないことは指摘されている。

このため、ヒアリングでは評価基準を専門家による定義に頼り切るのではなく、開発・評価・分析のサイクルを通じて徐々に評価基準を明確にしていく取り組みが確認できた。このように、評価自体も AI エージェントの開発フェーズと合わせて育てていくアプローチが有効だと考えられる。

### 4.2.3 具体的な取り組みと事例

#### (1) 立ち上げ期：仮説検証と早期フィードバック

不確実性の高い初期段階では、専門家のヒアリングを通じて仕様を明確にしていくだけではなく、最小限のプロトタイプで仮説検証を行うことの有効性が確認できた。

- 事例: LayerX
  - LayerX では Ai Workforce という AI エージェントを構築するためのプラットフォームを開発しており、そのうえでさまざまな AI エージェントを動かし企業向けにサービス提供している。
  - 課題: AI エージェントの構築では、業務のワークフローを構築するために、サポートすべき入出力を記録したデータセット一式が必要になる。一方で、そのようなデータが整備されていることは少なく、入力があっても望ましい出力が定義されていないこともある。
  - 対策: 最小限の入出力例から仮説を立ててプロトタイプを動かし、早期に顧客側の専門家へ提示した。実際の業務データを通じたフィードバックサイクルを早く回すことで、未知の仕様や想定外の挙動を効率的に抽出した。
  
- 事例: 伊藤忠テクノソリューションズ (CTC)
  - CTC では、大手金融機関向けの AI 導入支援を行っており、高い品質基準が求められる環境で開発を行っていた。
  - 課題: 顧客が求める「完璧な品質」と AI の現実的な性能との間にギャップがあり、プロジェクト開始時に明確な品質目標を合意することが困難であった。
  - 対策: 最初から大規模な開発を行うのではなく、短期的なプロトタイプ開発と試行 (PoC) を繰り返し、実際の出力結果を顧客に見せながらフィードバックを得ることで、段階的に目指すべき品質目標を具体化・合意していく「イテレーティブな品質定義」を実践していた。
  
- 事例: SpiralAI
  - SpiralAI では、独自のキャラクター性と安全性を持ったキャラクター AI サービスを展開していた。
  - 課題: キャラクター AI においては、「本人の性格を反映しているか」という正解の定義が難しく、また過度な安全性チェックがキャラクターの魅力 (面白さ) を損なってしまうという、プロダクトの根幹に関わる不確実性が存在していた。
  - 対策: 開発の初期段階から「やってみないとわからない」という性質を受容し、自ら日常的に利用するドッグフーディングや、SNS 上のユーザーの反応を観察す

ることを重視していた。そこで得られた「想定外だが魅力的な振る舞い」や「新たなリスク」を元に毎日チームで議論を行い、安全性とキャラクター性の最適なバランスを探索しながら、イテレーティブに品質基準を構築していた。

## (2) 育成期：専門家の知見の実装とデータセットの深化

ある程度の方向性が見えた段階で、専門家の知見をシステムに深く組み込み、評価データセットを充実させていく取り組みが確認できた。

### - 事例: Algomatic

- Algomatic では採用特化AIエージェント「リクルタAI」を展開しており、採用業務の専門家が実務を受託する形でサービス提供を行っている。
- 課題: 採用業務における「良いスカウト」の基準は、長年の実務に基づく専門家の暗黙知として蓄積されており、当事者であってもそのノウハウを形式値として整理することが難しいという課題があった。
- 対策: 採用の専門家（CHRO等）が開発の意思決定に直接関与し、ドメイン知識に基づいた採点基準をプロンプトへ落とし込んだ。さらに、利用ログについては個人情報適切にマスキングしたうえで、価値の高い回答およびリスクのある回答を抽出し、テストケースをイテレーティブに拡張。これにより、品質基準そのものをシステムへ実装していった。

### - 事例: PharmaX

- PharmaX では、オンライン薬局サービスにおいて、社内の専門家である薬剤師が開発プロセスに深く関与していた。
- 課題: 医療現場の複雑な判断基準や安全性の観点をエンジニアだけで定義・実装することは難しく、専門家の暗黙知をいかにプロンプトや評価指標に反映させるかが課題であった。
- 対策: 薬剤師自身がプロンプトの作成や、数百件規模に及ぶ定性的な評価項目の定義、およびデータアノテーションに直接従事した。専門家が「開発者」の一員として改善サイクルに加わることで、薬剤師業務の実態に即した精度の高い応答制御を実現していた。

## (3) 運用期：評価の自動化による継続的適応

サービス規模が拡大すると、人手による評価だけでは限界が来るため、自動化によるスケールアップが必要となる。

### - 事例: PharmaX

- PharmaX では Yojo というオンライン薬局を展開しており、AI エージェントを用いた対話によりユーザーから情報を収集し、薬剤師の業務に役立てている。
  - 課題: 医療情報を扱うため極めて高い安全性が求められるため、継続的な手動での評価が必要となっていた。
  - 対策: 専門家（薬剤師）の評価基準を LLM-as-a-Judge として実装し自動的に適切な回答を行った割合を算出できるようにした。
- 事例: サイボウズ
    - サイボウズでは、グループウェア製品に生成 AI 機能を組み込む際、QA（品質保証）エンジニアが開発プロセスの初期段階から深く関与していた。
    - 課題: これまでの QA テストは期待結果が明確に定まるものだったのに対し、RAG（検索拡張生成）のような AI 機能では入力と出力が無限に存在するため、何をもって「テスト OK」とするか判断基準の策定が困難であった。また、品質向上のために莫大な数の試行錯誤を行う必要があったが、表を用いた手動での実験管理では限界があった。
    - 対策: QA エンジニアがテストケースの作成だけでなく、プロンプト自体の改善提案も行う体制を構築した。また、自動評価ツールや実験管理ツール（Promptfoo や Ragas 等）を導入することで、大量の試行結果を効率的に管理・評価できる基盤を整え、継続的な品質改善を可能にした。

#### (4) 評価に基づくスコープコントロール（リスク管理への接続）

評価結果は、単なる品質チェックだけでなく、サービス提供範囲（スコープ）を決定するための重要な判断材料となる。

- 事例: すかいらくグループ
  - 課題: 「アレルゲンについて誤った情報を提供してしまうこと」や「存在しないメニューを提示してしまうこと」など、ビジネス上絶対に避けたいリスクが存在した。
  - 対策: それらの事象が起こらないよう、評価データセットを用いた徹底的なチューニングと検証を行った上で、安全性が確認できた範囲から段階的にリリースを実施している。
- 事例: PharmaX
  - 課題: エンドユーザーからの問い合わせは多様であり、安全に対応できる定期購入の申し込み、薬の飲み合わせなど高度な専門性が必要になり専門家へのフォローバックが必要になるもの、重篤な疾患な返答の内容が生命に係る場合まで含まれる。
  - 対策: ワークフローを組み、ユーザーからの問い合わせの種類をできるだけ高精度に分類するようにしている。また、分類した中で安全に対応できるユースケー

ス（定型的な問い合わせ等）のみ AI に任せ、専門家にフォールバックする、あるいは重篤な疾患の場合対応をお断りするなど対応でリスクを管理している。

※なお、具体的なリスク管理の手法（回避・低減・移転・受容）やプロセスについては、次節「4.3 リスクマネジメント」で詳述する。

#### 4.2.4 結論

##### 継続的な評価による品質と安全の統合

本節で述べた「継続的な評価プロセス」を回すことは、提供価値（品質）を高める活動であると同時に、リスクを特定し対策する（セーフティ）活動でもある。品質と安全はトレードオフの関係ではなく、同一のプロセスによって達成される。また、評価データセットと評価ログの蓄積は、どのような基準でテストされ、どのような品質・安全性が担保されているかを示す、監査可能なエビデンス（ガバナンス）となる。

##### 組織的な学習

このサイクルを通じて、組織は不確実な環境下で「何が価値で何がリスクか」を学習し、適応し続けている。AI エージェント開発とは、単なるソフトウェアの実装ではなく、組織自体が学習し続けるプロセスそのものと考えられる。

## 4.3 リスクマネジメント

### 4.3.3 リスクマネジメントで取り組む問題

AI エージェントを本番環境で利用するためには、リスクを適切にマネジメントすることが必要である。ただし、ここでいうリスクマネジメントとは、AIの利活用におけるブレーキではなく、むしろ、不確実性を低減し、AI エージェントの安全な利用を促進するためのものである。

リスクが不明確な場合、そのリスクを過剰評価してしまい、AI エージェントの導入が進まない原因となる。一方で、リスクを適切に評価・管理することで、根拠のない不安を払拭し、より積極的にAIエージェントの活用に取り組むことが可能となる。

### 4.3.2 AI エージェントのリスクマネジメントプロセス

AI エージェントのリスクマネジメントにおいてよく用いられるプロセスは以下の通りである。

1. AI エージェントのユースケースを列挙する。
2. ユースケースごとにリスクアセスメント（リスクの特定、分析、評価）を行う。
3. ユースケースごとにリスクへの対応方法（回避、低減、移転、受容）を決定する。
4. 安全だと判断できるユースケースに限ってサービスを提供する。
5. サービスの利用状況をモニタリングする。

まず、AIエージェントのユースケースの列挙では、AIエージェントが利用される具体的なシナリオを洗い出す。そして、そのユースケースごとに、リスクアセスメントを実施する。リスクアセスメントでは、まずリスクの特定を行い、次にリスクの発生確率や影響度を分析し、最後にリスクの評価を行う。これらの分析結果に基づいて、リスクへの対応では、各ユースケースに対して適切なリスク対応策を選択する。リスク対応策には、以下の4つがある。

- リスク回避: リスクの高いユースケースでの利用を避ける。
- リスク低減: リスクを軽減するための対策を講じる。
- リスク移転: リスクの影響・責任を第三者に移転する。
- リスク受容: リスクを受け入れる。

なお、リスク対応策は複数を組み合わせたたり同じ回避策の中でも複数の異なる対策を実施することが多くの企業から共通してヒアリングできた。

これらの適切なリスク対応策を講じたうえで、安全すなわち残留リスクがリスク許容範囲内であると判断されたユースケースに限ってサービスの提供を実施する。なお、サービスの提供開始後も、サービスの利用状況をモニタリングし、リスクが発生した場合に迅速に対応できる体制を整えることが重要である。

#### 4.3.4 AI エージェントのリスクマネジメントの事例

##### リスクマネジメントプロセスの例

ヒアリングで確認できたリスクマネジメントの例について述べる。

すかいらーくでは、顧客と直接対話しながら最適なメニューを提案するAI店長サービスを導入しており、このサービスを導入するにあたり、リスクマネジメントプロセスを適用している。まず、チャットボットのユースケースを検討し、アレルギーに関する質問への対応や存在しないメニューを回答してしまうハルシネーションのリスクがあることを特定した。次に、それぞれのリスクに対して対応を決定した。アレルギーに関する質問への対応については、リスクの影響が極めて深刻なため、リスク低減策として、チャットボットがアレルギーに関する質問には絶対に回答しないように事前に徹底的に検証を行うことでリスクを対策している。存在しないメニューを回答してしまうハルシネーションのリスクについては、完全にリスクを低減することは難しいため、リスクを受容することとした。具体的には、AIに見習い中というキャラクター設定を付与することで、ユーザーが不完全さを許容しやすいように工夫している。これらのリスク対応策を講じた上で、チャットボットのサービスを提供している。なお、サービス提供後も、チャットボットの利用状況をモニタリングし、リスクが発生した場合には、人間の責任者が対応できる体制を整えていた。

このように、リスク対応策については、ユースケースごとに適切に選択することが重要である。そこで、リスク回避、リスク低減、リスク移転、リスク受容の各アプローチについて、具体的な事例を以下に示す。

##### リスク回避の例

Ubieでは、業務効率化を支援する生成AIサービスを病院向けに提供している。その生成AIの中では、インフォームド・コンセント記録の作成や退院サマリ等の作成を支援する機能があるが、診療目的での利用は禁止している。文書作成などのユースケースに関してのみサービスを提供することで、目的外利用のリスク回避を実現している。

##### リスク低減の例

リスク低減として、リスクの発生確率や発生した時の影響を軽減するための対策はいくつかの方法が存在する。

まずは、先述のすかいらーくの事例のように、リスクを特定して徹底的に検証し、発生確率を低減した上で、リリースする方法である。同様の事例はLINEヤフーでも確認された。この方法では、リスクの検証の際に、3.4節で述べたAIエージェントのテストプロセスが重要な役割を果たす。

次に、リスクが顕在化した際の影響を抑えるための対策として、サービスの提供範囲や運用方法を段階的に設計する方法が挙げられる。

Algomaticの採用特化AIエージェントでは、顧客の採用方針を社内の専門家が解釈した上で、その専門家がAIエージェントの設定を行い、出力結果を顧客に連携する形をとっている。これにより、専門家が出力内容を精査した上で顧客に提供できるため、リスク軽減につながっている。さらに完全自動化と部分自動化の双方の運用方式を併存させ、顧客の方針やリスク許容度に応じてオプトアウト可能な運用設計も採用している。利用者が自動化レベルを選択できることで、導入初期の不確実性を抑えながら、安全性を担保した段階的な活用を可能にしている。また、日本生命のように、社内向けのチャットボットを提供する際に、まずは人数を限定した小規模なパイロット導入から始めて、徐々に利用範囲を拡大していく方法も多く見られた。

一方で、全社員が利用可能な社内向けのチャットボットを提供する場合は、ガイドラインや教育の提供でリスク低減を図っている例もある。詳しくは、5章で述べるが、たとえば、セガサミーでは、生成AIは不正確な情報や信頼性の低い情報を生成するリスクが常にあることをガイドラインで明示し、利用者への注意喚起を通じてリスク低減を図っている。

#### リスク移転の例

リスクの移転とは、想定されるリスクの影響を単独で抱えるのではなく、関係者間で適切に分担・共有することにより、全体としてのリスク管理を強化する手法である。一般的には、保険の活用や外部委託などがその代表例として挙げられる。

LINEヤフーでは、技術的対策の実装に加え、サービス利用に関する免責事項を明示することで、利用者との責任範囲を透明化し、適切なリスク分担を図っている。これは、組織としてリスク管理を徹底するための取り組みのひとつである。また、AIエージェントの実行プロセスにおいては、人の判断を組み込む「Human in the loop」の仕組みを採用している。これにより、最終的な確認や意思決定に人間が関与する体制を構築し、品質と安全性の担保を強化している。これは単なる責任の移譲ではなく、技術と人の協働によるリスク低減策の一環である。

なお、Human in the loopの詳細については、3.2節に記載している。

#### リスク受容の例

ヒアリングにおいて、リスク受容の例も確認された。KDDIでは「AIを活用しないことによる機会損失」自体を重大なリスクと捉え、AIの利用に伴う未知のリスクやグレーゾーンが完全に解消されない状況であっても、関連する公的なガイドライン等を参考に一定のリスクを受容することに務めており、例えば、人事分野などにおける社内AI活用においてモニタリングを行いつつ適用を進めていた。これは、リスクをゼロにするまで待つのではなく、最大限必要な対策を講じた上で、イノベーションのために必要最小限のリスクを引き受けるといった判断の一例である。

#### 4.3.5 リスクマネジメントに関する原則

これまでの事例から、リスクマネジメントにおいて共通して見られた傾向は以下の通りである。

- ユースケースを事前に特定し、ワークフローを細分化することで、リスクの低いユースケースに限定してAIエージェントを導入する。
- 初期はリスクの低いユースケースに限定し、徐々にAIエージェントの対応可能な範囲を広げていく。
- 顧客との積極的なコミュニケーションを通じて、リスクの発見やユーザーリテラシーの向上を図る。

## 4.4 サービスリリース後の迅速なフィードバックループ、モニタリング

本節では、サービスリリース後に行われるモニタリングと、そこから得られた知見を開発へ還流させるフィードバックループについて述べる。

これまでの MLOps や DevOps においても、本番環境でのモニタリングの重要性は強調されてきた。これは機械学習モデルを作成した時点でのデータと本番環境で提供されるデータには時間的な隔たりがあることや、機械学習モデルの精度向上によりサービスの KPI が向上するとは限らないことが背景にある。

本研究を通じて、AI エージェント開発においても、本番環境でのモニタリングが重要であることが確認できた。本番環境しか得られないリアルなデータを通じて、AI エージェントを継続的に改善させるためのモニタリングについて解説する。

なお、ここでのモニタリングはサービスの KPI のような効果測定での意味合いで検討する。監視を含む体系的なモニタリングについては 3.5. で述べる。

### 4.4.1 取り組む課題：「やってみないとわからない」への対処

#### リリース前の検証の限界

「4.2 継続的な評価による継続的な改善」で述べたように、AI エージェントの開発においては開発・評価・分析のフィードバックサイクルにより、評価データやプロンプトなどを充実させることで、AI エージェントの品質は向上する。しかし、専門家の協力のもとでも、本番環境でのユーザーの振る舞いを、開発段階で完全に予測し網羅することは難しい。

また、評価データを用いたAI エージェントの評価結果が向上した場合に、そのAI エージェントの KPI が連動するとは限らない。たとえば、AI エージェントがデータベースから検索しさまざまな選択肢を提示することを考える。この場合において、「多様な選択肢を提示すること」というのは品質評価の基準となりえる。一方で、あまりにも多すぎる選択肢を提示する、品質の低い選択肢を提示するといった振る舞いをするようになってしまった場合、ユーザーの利便性を損なってしまう。そのため、評価結果と KPI が連動することを確認するのは重要となる。

このように、リリース後にはじめて顕在化する「未知の入力」や「予期せぬリスク」が存在することを前提とし、それらを適切に検知するための運用体制が必要となる。

## 具体的な課題

### (1) 価値の不確実性（ビジネスKPIとの乖離）

すでに述べたように、AI エージェントの評価結果が向上しても、それがユーザーの課題解決やビジネス成果（業務効率化による作業時間短縮など）に直結しているとは限らない。ユーザーが本当に求めていた体験が何であったかは、実際に使われ、その結果を計測しなければ分からない。

### (2) 実環境における動作保証の不在

開発者や専門家（ドメインエキスパート）は、システムのリテラシーが高く、ある程度「正しい使い方」を知っているバイアスがある。一方で、実際のユーザーは想定外の操作や質問を行うことがあり、実環境ではじめて明らかになるユーザビリティや安全性の欠陥が存在する。

このため、ユーザーテストによる事前確認や、カナリアリリースによる小規模な本番環境へのリリースにより、実際のユーザーの利用時に生じる事象を確認することは重要となる。

### (3) 実環境における動作保証の不在

本番環境での AI エージェントの利用時には、特定の文脈での不適切な発言など、開発時には想定しきれなかったリスクが発生する可能性がある。とくに、ユーザーが悪意を持ってシステムを試行錯誤する場合、ジェイルブレイク（脱獄）攻撃のように、開発者の想定を超える入力が行われるリスクは常に存在する。

## 4.4.2 一般的な対策：ビジネス指標のモニタリングと段階的リリース

これらの課題に対し、ヒアリングでは次のような対策が確認できた。

### 対策1：多層的なモニタリング指標の設計

AI エージェントの評価結果とビジネス価値が連動していることを確認し、リスクを複合的に監視するために、多層的な指標を設計・計測する。指標の設計で検討する事項の例を次にあげる。

- 品質・リスク指標: ユーザーからのフィードバック（Good/Badボタン等）、ガードレールの発火回数、専門家へのエスカレーション率など、AI の振る舞いの健全性を測る指標。
- ビジネス指標: タスク完了率、成約率、ユーザーの離脱ポイントなど、AI が最終的なビジネス成果に貢献しているかを測る指標。

ボタンの押下率のような指標を取得するためには、システムの機能開発が必要となる場合がある。どのような指標を取得する必要があるのかは、AI エージェントの開発の初期段階で検討されていることが望ましい。

#### 対策2：リスクを局所化する段階的リリース

リリース後のリスクを最小限に抑えるため、いきなり全ユーザーに新機能を公開するのではなく、対象を限定して段階的にリリースすることが考えられる。

段階的なリリースとしては、β版として特定ユーザーにだけ先行提供することが考えられる。また、すでに本番環境としてAI エージェントが稼働している場合には、アップデートのリリース時にカナリアリリースやA/Bテストなどの活用が検討できる。これにより、価値とリスクを検証した上で安全性が確認された場合にのみ公開範囲を拡大することで、万が一のリスク顕在化時も影響範囲を最小化できる。

#### 対策3：本番環境での素早い評価

モニタリングで得られたデータを継続的に評価することで、ユーザーの想定していなかった入力や、特定の文脈での不適切な発言、悪意のあるユーザーの取り組みをすばやく検知できるようになる。

このような検知を大量のデータに対して、全件目視で行うことは現実的ではない。このため、正規表現のルールベースでの検知や、LLM-as-a-Judgeなどを組み合わせて評価することが重要になる。

### 4.4.3 具体的な取り組みと事例

#### (1) ビジネス指標と利用状況のモニタリング

ユーザーの行動変容やビジネス成果という「マクロな価値」と対話の中身という「ミクロな品質」を継続的に追跡する試みがヒアリングで確認できた。

- 事例: PharmaX
  - 企業・サービス: PharmaX (Yojo)
  - 課題: AI の回答精度が高くなるよう機能改善した場合に、ユーザーの利便性が向上したか提供価値を定量化する必要があった。
  - 対策: 対話の完了率や、ユーザーからの返信数などのメトリクスを定義して計測するとともに、リリース後には対話のトレースを記録して実際のユーザーとの対話ログを分析可能にし、離脱原因を特定してプロンプトやシナリオの改善に繋げている。

## (2) 本番環境での段階的な検証（カナリアリリース・限定公開）

リスクを最小限に抑えつつ、実環境でのフィードバックを得るために、公開範囲を意図的に限定する取り組みがヒアリングにより確認できた。

- 事例: SmartHR
  - 企業・サービス: SmartHR（クラウド人事労務ソフトにおける AI 機能）
  - 具体的課題: RAG を簡便に構築できるようにする新機能のリリースにあたり、多種多様な企業における実効性と、予期せぬリスク（誤情報の提示等）を事前に完全に見極めることが難しかった。
  - 具体的対策:
    - プロンプトだけではなく、生成 AI プロバイダーの提供するガードレールを導入
    - リリース前にテストデータを自前で構築して定量的な評価を行うとともに、レッドチーミングを行い、敵対的な入力に対する振る舞いを徹底的にテスト
    - いきなり全ユーザーに公開するのではなく、特定の期間を「限定公開（ベータ版）」として設定。希望するユーザーのみに提供することで、意欲の高いユーザーからの詳細なフィードバックと実利用データを収集
    - リリース後も個人情報保護の観点からオプトイン方式を採用し、利用者の意図しない情報漏洩が発生しないようにサービスを設計

## (3) リスク検知と即応体制

本番環境で発生したハルシネーションや不適切な挙動を迅速に検知し、被害を最小化する体制を構築する。

- 事例: すかいらくグループ
  - 企業・サービス: AI ロボ（注文時の接客エージェント）
  - 具体的課題: 実店舗での利用において、敵対的な入力も含め、ユーザーから当初想定しないような多様な入力が数多く寄せられた
  - 具体的対策: 本番環境へのリリース直後は、現場に担当者を配置して、敵対的な入力があった場合に即座に対応できるように体制を構築。また、問い合わせログを監視体制に組み込み、カナリアリリースを行うことで、意図しない不具合の影響を最小化している。
- 事例: PharmaX
  - 具体的課題: オンライン薬局という業態の都合上、健康に関わるような問い合わせに答える必要がある。「重篤な疾患に関する問い合わせに対しては別の医療機関の受診を促す」のように、ユーザーの問い合わせに対して事前に対応を検討したものであれば良いが、今まで検討したことのない問い合わせがないか監視が必要。

- 具体的対策: AI エージェントがユーザーの質問に対して回答するワークフローの初期段階で、ユーザーの発話の意図を分類するように設計し、人手での対応が必要なものは専門家にエスカレーションするようにワークフローを設計。会話の分類精度や、エスカレーションの発生頻度をモニタリングすることで、想定できていなかった問い合わせを特定し、徐々に対応できる範囲を精緻化し拡大していった。

#### 4.4.4 結論

リリース後のモニタリングは、組織が行った改善のための取り組みが、ユーザーにとっての真の価値を向上させ、許容すべきでないリスクを低減しているか確認する重要なプロセスである。

開発段階でどれほど評価を尽くしても、本番環境での意図しない挙動を調べ尽くすことはできない。モニタリングで得られたフィードバックを迅速に分析できるように仕組みを整えておき、フィードバックループを高速に回すことも、AI エージェントを本番環境で提供できる「プロダクト」とするためには必要となることが確認できた。

## 5章 組織

本章では、AI エージェントの活用を全社的に推進し、安全かつ継続的に価値を生み出し続けるための「組織」の取り組み、とりわけ AI ガバナンス のあり方について述べる。

生成 AI という変化の激しい技術を組織的に活用することは、従来の固定的なルールによる統制では困難である。ヒアリング調査を通じて、先行企業は「アジャイルガバナンス」の理念を実践し、現場を支援するイネイブリングチームとしての活動や継続的なフィードバックを通じて、組織全体の学習を加速させていることが明らかになった。本章では、これらの知見を以下の7つの観点から整理する。

- 5.1 組織体制: 監視と監督に留まらず、現場の自律的な価値創造を支援する AI ガバナンスチームの役割と設計について述べる。
- 5.2 ガイドラインの整備: 技術進化やリスクの変化に合わせて機動的にアップデートし続ける、動的な指針のあり方を解説する。
- 5.3 統合的な AI 相談窓口: 個別事案の相談を通じて現場の課題を組織の知見へと昇華させ、迅速な意思決定を支える人的インターフェースについて説く。
- 5.4 AI 活用支援: 事業部門が自律的に AI を業務に落とし込み、継続的に改善できる状態をつくるための伴走型支援の意義を整理する。
- 5.5 教育: リスクを正しく理解し、安全な活用範囲を自律的に判断できるリテラシーを組織全体で育む取り組みを紹介する。
- 5.6 ISO 42001 による認証: 国際標準への適合を通じて、プロセスが適切であることを客観的に保証し、社会的な信頼を獲得する意義を論じる。
- 5.7 トップのコミット: アジャイルガバナンスを組織文化として定着させるために不可欠な、経営層による戦略的な意思決定と支援について述べる。

本章で提示する組織的な取り組みは、アジャイルガバナンスを通じた組織的な学習を積み重ねることで、リスクを許容可能な範囲に制御しながら AI による価値を最大化し続けるための枠組みである。

## 5.1 組織体制

### AIを活用するための組織体制の課題とAIガバナンスチーム

AIを組織的に活用し、その価値を最大化するためには、AIを活用するための組織設計が不可欠である。

ここでは、AIを活用するための典型的な組織体制について、内部統制の基本概念である3線モデルを基に整理する。

まず、3線モデルとは、組織におけるガバナンスやリスク管理を行う際の役割分担の仕組みであり、次の3つの部門からなる。

- 事業部門（開発部門・営業部門）（1線）
- リスク・コンプライアンス部門（リスク管理部門・法務部門・セキュリティ担当部門）（2線）
- 内部監査部門（3線）

#### 事業部門（開発部門・営業部門）（1線）

事業部門は組織のビジネスを推進する部門であり、次のような役割を担う。

- AI エージェントの利用
- AI エージェントを組み込んだサービスの企画・開発・運用
- AI 利用の目的・期待効果の明確化
- 定義されたガイドラインの遵守

すなわち、1線はAIを実際に活用し、その価値を最大化する役割を担う。

#### リスク・コンプライアンス部門（リスク管理部門・法務部門・セキュリティ担当部門）（2線）

リスク・コンプライアンス部門はAI活用に伴うリスクを組織全体の視点から管理・監督する責務を担い、主な役割は次のとおりである。

- AIガバナンスに関するガイドラインやリスク管理ガイドラインの策定
- 事業部門のAI活用に関する監督・支援

すなわち、2線はAI活用に伴うリスクを定義し、監督する役割を担う。ただし、2線部署は一見するとAI活用を阻害する「守り」の部門と思われがちであるが、AIを活用するためのガー

ドレールとしてルール作りを行っており、AIの活用を推進したいというスタンスであることがヒアリングを通じて確認された。

### 内部監査部門 (3線)

内部監査部門は1線や2線が有効に機能しているかを独立に確認する責務を担う。

- ガバナンスとリスク・マネジメントの妥当性と有効性を検証する
- 必要に応じて、助言を提供する

従来のガバナンスやリスク管理においては、このような3線モデルが用いられている一方で、AIのように高度な専門性が必要とされる領域においては、1線と2線の間にギャップが生じ、以下のような課題が発生する。

- 1線の現場業務が高度化・複雑化したことで、2線が現場に即したガイドラインを策定することができず、ガイドラインが抽象的になりがちである。
- 2線部門のAIに関する知見が不足していることが多く、適切なガイドラインを策定することが難しい。
- 1線と2線の間で認識や用語の違いが生じ、コミュニケーションが円滑に行われないことがある。

これらの課題により、2線部署もAI活用の推進というスタンスは共通しているものの、協調して動くことができず、活用が進められないという事態が発生していた。そこで、これらのギャップを埋めるために、1線に近い位置で2線のガイドラインを業務に適した形で落とし込み、具体化する役割を担うAIガバナンスチームが設置される事例が多く見られた。

すなわち、3線モデルに1線と2線の間位置する「1.5線」としてAIガバナンスチームが位置付けられる。

- 事業部門 (開発部門・営業部門) (1線)
- AIガバナンスチーム (1.5線)
- リスク・コンプライアンス部門 (リスク管理部門・法務部門・セキュリティ担当部門) (2線)
- 内部監査部門 (3線)

### AIガバナンスチーム (1.5線)

AIガバナンスチームは1線と2線の間位置し、AIに関する統制を実務レベルに具体化し、1線と2線の橋渡しを行う役割を担う。具体的には、次のような役割を果たす。

- 2線が策定したガイドラインを1線の業務に適した形で落とし込み、具体化する

- 1 線の AI 活用に関する相談・支援
- 1 線の事業目的や実現したい価値を、2 線が正しく理解できるように支援を行う
- AI 技術の動向を踏まえたガイドラインへのフィードバック

すなわち、1.5 線は、2 線が定めた AI ガバナンスガイドラインを実務に落とし込んで、1 線が実践できるようにするための補助的な役割を担っている。

## 厳密さと柔軟さを併せ持つ AI ガバナンスの実現

AI ガバナンスにおいては、活用とリスク管理はトレードオフの関係になりやすく、リスク一辺倒なガバナンスは活用を妨げる要因となる。生成 AI 時代においては、AI を活用しないことによる機会損失自体が最大の経営リスクであり、リスクを過剰に恐れて AI 活用を妨げることは避けなければならない。したがって、AI ガバナンスは AI の活用を止めるためのブレーキではなく、安全に走行するためのステアリング（舵取り）であると捉え直す動きが、先進的な組織で共通して見られた。

AI の活用を見据えた AI ガバナンスにおいては、厳密さと柔軟さを併せ持つアプローチが求められる。厳密さとは、ガイドラインやルールを明確化することで、従業員が安心して AI を活用できる環境を整備することである。これが欠けると、従業員がリスクを過剰に恐れて AI を使わないという事態が生じる可能性がある。

一方で、柔軟さとは、過剰なルールメイキングによる Agility（俊敏性）の低下を避け、状況に応じた柔軟な対応を可能にすることである。AI 技術は急速に進化しており、固定的なルールでは対応しきれない場合が多いため、事業部門がやりたいことを実現するために固定的なルールに縛られすぎない柔軟な対応を行うことが考えられる。

たとえば、ヒアリングを通じて、責任者を厳密に定めることで迅速な対応を可能にしつつ、対応フローは厳密には定めず、状況に応じて柔軟に対応できるようにするようなインシデント管理の取り組みが確認された。

このように、厳密さと柔軟さを両立させたガバナンス設計こそが、生成 AI 時代における AI ガバナンスの鍵となる。

## AI ガバナンス事例

### Preferred Networks (PFN)

最先端の AI 開発を行う PFN では、AI の開発に重点を置いた AI ガバナンス体制を構築していた。

具体的には、リスク評価委員会やセキュリティ委員会といった2線の組織に加え、AI特有の課題に対応するためのAIガバナンス推進組織として1.5線の組織を設置していた。

同社では、AIガバナンスをAI開発を事業とする企業にとって不可欠のものと位置付けており、AIガバナンス推進組織のミッションも安全なAI開発の推進となっていた。このミッションを達成するため、AIガバナンス推進組織はリスク管理の専門家とエンジニアリングマネージャーの経験者で構成されていた。エンジニアリングマネージャーの経験者は開発チームのことをよく理解しているため、1線と2線の言葉の壁を解消して効率的なリスク管理を可能にしていた。具体的には、リスク評価を「開発方針」という名称で実施していた。これにより、リスク管理がプロジェクトマネジメントにおける一プロセスとして自然と組み込まれ、1線側が主体的に品質管理を行うことを促進している様子が確認された。

## LINE ヤフー

日本最大級のプラットフォームを運営するLINEヤフーにも、2線のAI特有のリスク管理も行いつつ、1.5線としての役割を担うAIガバナンスチームが設置されていた。

ヒアリングの中では、AIガバナンスチームのミッションとして、単なるコントロールや抑制ではなく、AIの活用を推進し、より革新的なサービスを生み出すための支援を行うことが挙げられていた。必要なリスク管理を適切に行いつつ、基本的には1線部門が具体的な対策を自ら判断し、柔軟に対応できるよう促すことが基本的なチームの姿勢となっていた。

具体的には、AIガバナンスチームで社内ガイドラインを作成し、その中で具体的なリスクに対する低減策を提示していた。ただし強制力はなく、リスクレベルに応じて1線部門が自律的に対応策を検討できるように、ガイドラインでリスク低減策を提示していた。実際の案件の実施要否については、他の2線部門の専門的な判断と合わせて、総合的に判断されている。

## 5.2 ガイドラインの整備

本節では、AI エージェントを安心・安全に利活用するためのガイドライン策定について解説する。

### 5.2.1 ガイドラインの階層構造

総務省及び経済産業省による AI 事業者ガイドライン<sup>29</sup> では、AI に関連した事業活動を行う各主体が「システムの検証可能性を確保しながらステークホルダーに対する適切な情報を提供することにより透明性を向上させ、アカウントビリティを果たすこと」の重要性を強調している。こうした要請に対し、社内の責任の所在や補完すべき情報を明確にする目的で各企業内でのガイドラインの存在が望まれる。

一方で、AI 事業者ガイドラインでは具体的な運用方法までは定められておらず、ルールをどのようにドキュメント化するかは各組織に委ねられている。他にも、FDUA (金融データ活用推進協会)<sup>30</sup> や JaDHA (日本デジタルヘルス・アライアンス)<sup>31</sup> などの業界団体によっても生成 AI ガイドラインが発表されているが、詳細な実装を各企業が行うという構造は共通している。

そのため、各企業内で独自のガイドライン策定を含むルール作りが行われることになる。こうした取り組みは、AI エージェントを含む AI アプリケーションの安全性を担保し外部への説明責任を果たす上で効果的であり、実際にヒアリングを行った多くの企業でも同様の取り組みが見られた。一方で、過度に詳細なルールは、かえって企業の AI 利活用を萎縮させ、活動を制限してしまう恐れがある。そのため、ルールは適切な粒度で設計されなければならない。

こうした背景から、ガイドラインの整備には階層構造が採用されることが多い。本節では、こうした階層構造を以下の 3 層で定義する。

- AI 活用指針 (AI ポリシー): 組織全体を対象とし、対外的にも公表する最上位の指針
- AI ガイドライン: 自社のビジネス特性を踏まえた、社内向けの実務ガイドライン
- AI 開発方針: 個々のサービスやプロジェクトにおける具体的な利用方法をまとめたドキュメント

### 5.2.2 ガイドラインのレビューパイプライン

組織におけるガイドラインの整備は典型的には以下のような手順で進んでいく。

---

<sup>29</sup>[https://www.soumu.go.jp/main\\_sosiki/kenkyu/ai\\_network/02ryutsu20\\_04000019.html](https://www.soumu.go.jp/main_sosiki/kenkyu/ai_network/02ryutsu20_04000019.html)

<sup>30</sup><https://www.fdua.org/news/20250714-02>

<sup>31</sup><https://jadha.jp/news/news20250207.html>

- まず、総務省 AI ガイドラインに代表される一般に公開されているガイドラインから、会社全体で用いる AI 活用指針が制定される
- AI 活用指針に準拠するように、業務に関する具体的なプロセスを含んだ AI ガイドラインや AI 開発方針が制定される

特に下位の AI ガイドラインや AI 開発方針の制定にあたっては、ルールを制定するリスク管理部門と実際にそれが適用されるビジネス・開発部門間のコミュニケーションが不可欠である。この際、前節で扱った AI ガバナンスチームが 2 つの部門間の情報伝達の役割を担う。

### 5.2.3 ガイドライン整備における考慮事項

ガイドラインの策定においては、以下のような事項に考慮する必要がある。

- 外部のガイドラインをそのまま社内向けに用いない。社内ガイドラインの作成時には、一般的な AI ガイドラインや業界標準のルールを定めたガイドラインの参照が必要だが、これらのガイドラインは実際の実務との繋がりが少ないことも多い。あくまでも既存のガイドラインは参照にとどめ、有効に AI 利活用を推進できる形に作り変えていくことが必要である。
- 柔軟にガイドラインのアップデートを行う。AI 関連技術の変化は非常に早く、最新情報を柔軟に取り入れて追隨していく必要がある。また、組織の AI 利活用促進を妨げないという意味でも、実運用に即した定期的な見直しが必要である。

### 5.2.4 具体的な取り組みと事例

最後に、5.2.1 で紹介した階層構造に沿ってヒアリングで見られた各企業の事例をいくつか紹介する。

#### AI 活用指針 (AIポリシー)

##### インテージ

インテージでは、2024 年 4 月に安全性重視・セキュリティの強化や公平性の確保といった 5 項目からなる [AI 利活用指針](#) を社外向けに公開している。また、この利活用指針に基づき、以下のようなより詳細な従業員向け文書も制定している。

- AI ガバナンス規定: AI サービスの導入や開発プロジェクトにおける基本原則、およびプロジェクト責任者の責務について整理したもの
- AI リスク管理ガイドライン: サービス提供者・データ提供者などのロールごとにプロセスを詳細化し、発生が懸念されるリスクとその対策について整理したもの

- AI リスク判定申請書: ハルシネーション、著作権侵害、セキュリティなどの各種 AI リスクをチェックするための文書。記入内容のリスク判定を行うための AI ガバナンス委員会が存在し、半年で 40 件以上のリスク判定が実施されている。

## KDDI

KDDI では、誤判定、ハルシネーションおよびバイアスなどの既存のソフトウェアには存在しなかった AI リスクに対処するためにガイドライン等の整備を行っている。

2019年に日本政府が策定した「人間中心のAI社会原則」に基づき、KDDIでは、2021年にプライバシー・公平性など9つの原則を定めた「KDDIグループAI開発・利活用原則」を策定。また、「人間中心のAI社会原則」を守るため総務省・経産省が2024年に定めた「AI事業者ガイドライン」に基づき、KDDIでは、AI開発や提供を行う上での指針として「AI開発ガイドライン」を整備。

また、企画フェーズにおいてはAI開発影響評価(AIA)も導入された。これは、新規サービスの計画立案後にガイドラインに付随するチェックシートを用いてAI原則への適合性を評価するもので、具体的な企画へのガイドラインの実装が進められている。

## AI ガイドライン

### LINEヤフー

LINEヤフーでは8項目からなる「AI倫理基本方針」を制定しており、これを基本として社内に展開するガイドラインを作成した。作成にあたっては、AI事業者ガイドライン、AISI評価観点ガイドなどの既存のガイドラインや法制度も共に参照されている。

ガイドラインは、積極的なAI利活用・AIリスクの管理などの従業員/組織の行動原則を定めた部分(総論)及びハルシネーションや権利侵害などの具体的なAIリスクを記述した部分(各論)に大別される。各論部分では、各AIリスクカテゴリについて代表的なリスクとその緩和策が記述されている。

### 日本生命

日本生命ではFDUAガイドラインやAI事業者ガイドラインを参考に社内規定の検討が進んでいる。

まず情報資産、SaaS利用、システム開発基準などの既存のルールに関する専門家とAIの専門家が少人数のチームで素案を作り、コンサルタントや弁護士などの外部の専門家によって精緻化が行われる。

この際の考慮事項として、ヒアリングでは以下のような点があげられた。

- 既存のルールを参照しながら AI 活用ルールを整備するのは難しい。AI 特有のルールを作成するのか既存のルールで対応するのかで判断が分かれる。
- 利便性とのバランスも重要なポイントである。既存ルールをベースに考えると結果として作成されるチェックリストが長大になり、自分たちの活動自体が制限されてしまいかねないと感じている。
- ここ 5 年で金融機関においても新規サービスの利活用が進み、利用されているサービス全体を把握するのが困難になってきた。素早くルールを作って統率を効かせる必要があると感じている。

こうした背景を踏まえ、日本生命では次の事項も推進している。

- 新しい外部のガイドラインについて、制定された文脈や、その背景などの情報にキャッチアップできるような体制の整備
- 開発担当者だけでなく、外部の専門家・有識者の意見も仰ぎつつ、AI に関する取り組みにおけるリスクのさらなる洗い出しを行う枠組みの整備

#### アイレット (iret)

アイレットでは、既存の情報セキュリティマネジメントシステム (ISMS) の運用実績をベースに、AI マネジメントシステム (AIMS) を構築していた。

具体的には、ISMS で定義された従来の情報機密レベルと AI 利用ルールをマッピングし、扱うデータの重要度に応じて利用可能な AI サービスやプランを制限する仕組みを導入していた。これにより、既存のセキュリティガバナンスの枠組みを AI にも適用することで、組織としての安全性を担保していた。また、これらの取り組みを基盤として、ISO/IEC 42001 認証の取得も実現していた。

#### メルカリ

メルカリにおいては、日本の法律やガイドライン（経済産業省・総務省の AI 事業者ガイドライン、個人情報保護法など）をもとに「生成 AI 利用ガイドライン」が作成されており、全社員がこれに基づいた必修研修を受講している。

メルカリでは AI を使わないことを最大のリスクとしてとらえ、AI 活用の促進をガイドライン作成の主な目的としておいている。明確なルールの制定と、全社員が参加可能な勉強会や Slack での AI 活用事例の共有などの取り組みを組み合わせることで、社員の AI 活用を推進するような体制を整えている。

## AI 開発方針

### Ubie

Ubie では、顧客となる病院との間で本格的な製品導入に向けて数ヶ月程度のオンボーディング期間を設けている。その際に事前に利用のための院内向けガイドラインの整備をサポートし、実際に生成AIプロダクトを利用する医療従事者が適切に活用できるよう、想定された用途以外での利用、個人情報漏えいやハルシネーションなどの生成 AI 特有の問題について注意喚起を行うことで利用時の安全性を担保している。

また、Ubie は、日本デジタルヘルス・アライアンス（JaDHA）においてワーキングリーダーとして「ヘルスケア領域における生成 AIガイド」の策定をリードしており、当該ガイドを踏まえたプロダクト開発を行っている。

## 5.3 統合的な AI 相談窓口

本節では、ガイドラインの策定だけでは解決できない現場の課題に対し、人的な支援を行う「統合的な AI 相談窓口」の設置とその重要性について述べる。

前節で述べた通り、ガイドラインは AI の利活用において全体的な方針や禁止事項を示すために必要不可欠である。しかし、個別の業務フローや特定の SaaS ツールの導入可否、あるいは「これは規約上グレーではないか？」といった判断に迷うケースに対して、静的な文書だけですべての答えを用意することは困難である。そのため、現場からの疑問を受け止め、安全かつ効果的な活用を支援するための「人的なインターフェース」としての相談窓口が必要となる。

### 相談窓口の役割と立ち位置

AI 相談窓口に求められる最も重要な役割は、AI スペシャリスト達が社員からの AI に関する質問に対応し、企業全体の AI 利活用を推進することである。これはチームトポロジーにおける「イネーブリングチーム (Enabling Team)」の概念に近く、単にリスクを判定して「禁止」することだけを目的とするのではなく、現場が適切にリスクマネジメントを行えるよう支援し、活用を促進するための窓口として機能することが求められる。

相談窓口で扱うべきトピックは多岐にわたる。

- セキュリティ観点での SaaS 導入可否の判断
- 企画・開発段階でのリスクレベルの判定協力
- プロンプトエンジニアリングや技術的な実装に関する相談

このように、コンプライアンス上の相談だけでなく、技術的な「どうすれば実現できるか」という相談も一手に引き受けることで、現場の心理的ハードルを下げることが重要となる。

### ワンストップ化と専門家の連携体制

AI の利活用を推進している企業の多くは、社内のあらゆる AI 関連の問い合わせを一元的に管理する「ワンストップ相談窓口」を設置していた。

メルカリでは、AI 活用に関する質問や相談を一元的に受け付ける窓口を設け、社員が「どこに聞けばいいかわからない」と迷う状態を防いでいた。また、相談内容や回答事例を Slack 等で共有することで、全社的なナレッジの可視化を図っていた。

LINE ヤフーの事例では、機能開発や企画の初期段階から AI 統括部門が協力し、リスクレベルを判断するプロセスを確立していた。ここでは、技術部門の専門家がモデルのセキュリティ対

策を評価し、法務や知財の専門家と連携してリスクを判断するという、多角的な視点によるサポートが行われていた

日本生命においても、既存ルール of 専門家と AI の専門家が少人数のチームを組み、個別事案に対してスピーディにルール作成や対応を行う体制をとっていた。

## 信頼関係の構築とコミュニティ化

相談窓口は単なる「審査機関」ではなく、現場との信頼関係を築くためのハブとしても機能する。現場からの相談に真摯に対応し、「どうすれば安全に使えるか」を共に考える姿勢を示すことで、ガバナンスへの理解と協力が得られやすくなる。

セガでは、知財の専門家を含んだ開発者たちによる「生成 AI 委員会」を発足させていた。この委員会が従業員からの相談や問い合わせを受け付けることで、現場における信頼性が向上し、ガイドラインのアップデートなどの周知もスムーズに行われるようになった。管理側が単に規制するだけでなく「活用を盛り上げる」姿勢を見せることで、ポジティブなコミュニティ形成がなされていた。

## スケーラビリティの課題と現実的な対策

相談窓口の設置において避けて通れないのが、「ボトルネック化」と「リソース不足」の課題である。すべての中央ガバナンスチームが全件を審査しようとする、承認待ちが発生し、スピード感を損なう恐れがある。

サントリーでは、ガイドラインに従って、判断に迷う各部門からの相談が月に 30~40 件寄せられており、各サービスのポリシー確認などに多くの工数が割かれている実情があった。こうした課題への対策として、以下の 2 つのアプローチが確認された。

1. 「AIサービスの活用」「AIサービスの提供」を分け、「AIサービスの活用」は主に情報漏洩などセキュリティリスクのチェックに留めハルシネーションリスクなどは現場判断とし、「AIサービスの提供」は情報漏洩だけではなく、AI人権や公平性の確保、各種権利関係の侵害や情報の正確性などが担保出来るのか相談窓口でチェックするなど、レベルに応じて対応を分けた運用とする
2. 過去事例などから、申請の妥当性をAIで一時判断し、窓口ではその判断でよいかのチェックをして返答するなど、自動化を目指す

また、担当者の「専門知識の不足」や、部署ごとの「リテラシーのギャップ」も課題となる。日本企業では AI 専任ではない担当者が兼務で窓口を担うことも多いため、法務・知財・セキュリティなど他部署を巻き込んだ連携体制を構築し、特定個人に負荷が集中しないよう設計することが肝要である。

## 5.4 AI 活用支援

本節では、AI ガバナンスチーム（または AI CoE）が、社内の事業部門・開発部門に対して生成 AI を安全に業務へ落とし込み、継続的に改善できる状態をつくるための支援活動を整理する。

### 5.4.1 AI 活用支援とは

#### 取り組む課題

ヒアリングでは、生成 AI は従来の情報システム/従来型の機械学習に比べて、導入・運用時の不確実性が大きく、リスク管理と活用推進を同時に進める難易度が高いことが共通して指摘された。

- 挙動の揺らぎ：入力（プロンプト）や文脈、外部依存（モデル更新等）で出力が変化し、合格基準・テスト観点を固定しにくい
- 論点の多さ：ハルシネーション、機密情報・個人情報、著作権、セキュリティ、レピュテーション等が同時に絡み、単一部門で完結しない
- ステークホルダーの多様性：開発部門だけでなく、法務・コンプライアンス・リスク管理・セキュリティ・監査等（いわゆる 2 線・3 線）との調整が不可欠
- 変化の速さ：社会受容やガイドラインの更新が速く、過去の前提が短期間で変わり得る

この結果、現場主導で PoC/実装が進むと、後工程で指摘が入り手戻りが発生しやすい。一方で、リスク対策を「外付けの承認ゲート」として設計すると、開発速度が低下し、利活用推進そのものが停滞する。

したがって AI 活用支援では、ガバナンスチームが 1 線（事業・開発）と 2 線（法務・リスク・セキュリティ等）をつなぎ、現場の試行錯誤を止めずに必要な論点を拾える“回り道の少ない導線”を設計することが重要となる。

#### AI 活用支援で扱う範囲

AI 活用支援は、単に「審査する」ことではなく、現場の開発が止まらない形で、必要な論点を漏れなく拾える運用を設計し、短期間の伴走を通じて現場を自走状態へ引き上げる取り組みである。

ヒアリングで多く見られた進め方は、次のとおりである。

- 相談受付・トリアージ
  - 目的/対象業務/取り扱い情報（機密・個人情報等）/利用者/外部公開有無など、最低限の前提を整理する

- その前提から、関与が必要な部署（法務、コンプライアンス、セキュリティ、監査等）を特定する
- 論点の棚卸しと合意形成
  - 「決めるべきこと」を論点表として可視化する（例：データ境界、責任分界、禁止事項、説明責任、ログ方針、外部公開時の表現等）
  - 誰が・いつ・何を判断するか（責任者、レビュー観点、承認ルート）を決め、後工程の差し戻しを減らす
- 設計・実装の伴走
  - アーキテクチャ、権限設計、ガードレール（入力/出力の制御、フォールバック等）、ログ、評価方針を現場の開発プロセスに組み込む
  - 「判断に必要な情報が揃う形」になるよう、必要最低限の成果物（開発方針、確認表等）へ落とす
- 検査・リリース判定
  - セキュリティ/個人情報/契約・規約といった既存の必須プロセスと、生成 AI 特有の観点（誤情報、漏洩、悪用等）を統合して確認する
- 運用・改善
  - インシデント/問い合わせ/指標のモニタリングを行い、改善サイクルを回す
  - 変更（モデル更新、プロンプト変更、データ更新、利用範囲の拡大等）がある場合の再評価と手順更新を前提にする

#### 5.4.2 ヒアリングで確認されたベストプラクティス

本節では、AI 活用支援を組織として安定運用するために、ヒアリングで繰り返し言及されたポイントを一般論として整理する。

##### (1) 入口で「目的・守るべきもの・前提」を揃える

- 目的（何を良くしたいか）と、守るべきもの（個人情報、機密、ブランド等）を最初に言語化する
- 利用者、利用範囲、外部公開の有無など、後で揉めやすい前提を早期に固定する

##### (2) 合意形成を先に設計する

- 生成 AI では関係部署が増えるため、「いつ誰が何を判断するか」を先に決めないと差し戻しが増える
- 論点表（決めるべき事項の一覧）を作り、責任者と承認ルートを明確にする

##### (3) レビューは既存の必須プロセスに寄せる

- 追加の承認ゲートを増やしすぎると推進が止まりやすい

- セキュリティチェック/個人情報/契約・規約など、必ず通る手続きに AI 特有の観点を統合し、取りこぼしを減らす

(4) 評価は「最小セット」から始め、回帰として回す

- 生成 AI は挙動が揺れるため、単発テストではなく、継続的な回帰評価が前提になる
- まずは実害が大きい観点（誤情報、漏洩、悪用、権限逸脱等）を優先し、合格基準とフォールバックを設計する

(5) 境界（データ・権限・ログ）を先に決める

- 何を入力させてよいか/どこまで出力させてよいか、権限（ツール実行・外部連携）を先に決める
- 後追いで説明責任に備え、必要十分なログ（誰が・いつ・何をしたか）を取れる設計にする

(6) “変化”と“インシデント”を運用に含める

- モデル更新やプロンプト更新など、前提が変わることを想定して再評価の手順を用意する
- インシデント時の連絡系統・一次対応・再発防止の流れを明確にする

### 5.4.3 個別事例

#### Preferred Networks の取り組み（生成 AI ガバナンスの設計と運用）

Preferred Networks (PFN) は、深層学習（ディープラーニング）などの最先端技術を自社開発し、製造、交通、バイオ、エンターテインメントなど多様な分野へ社会実装を行う日本を代表する AI 企業である。同社内では、AI ガバナンスを専門に扱うチームが各プロジェクトの支援にあたった。

本項では、ヒアリングで確認された PFN の取り組みを、前節までで述べた「進め方（5.4.1）」と「ベストプラクティス（5.4.2）」の観点に対応づけながら整理する。

体制：1 線・2 線・3 線の役割分担と推進機能

- 1 線（各プロジェクト：事業・開発）が主体となって実装・運用を進める
- 2 線（リスク評価委員会、セキュリティ委員会等）が横断観点で評価・要求を行う
- 3 線（内部監査）が独立に点検する

- 既存の枠組みでカバーしきれない生成 AI 特有の論点については、推進機能が 1 線と 2 線の間をつなぎ、論点整理と運用設計を支援する

文書体系：方針を“現場の手順”へ落とすための階層化

- AI ポリシー：透明性、リスクへの配慮、誠実さ（インテグリティ）等の原則
- AI ガバナンス方針/AI 品質ガイドライン：守るべき観点・手続き
- プロジェクトの AI 開発方針：個別の事情を踏まえ、どの観点をどう担保するかを具体化

この階層により、抽象的な要求（透明性・説明責任等）を、プロジェクトの実装・運用で扱える単位に落とし込む。

レビュー運用：確認表（チェックリスト）とチェックポイント設計

- リスク評価の場では、確認表（チェックリスト）を用いて、潜在リスク/求める対策/残存リスクを整理し、プロジェクトが取るべきアクションを具体化する
- 実務上「必ず通る」手続き（例：セキュリティチェック、契約・規約レビュー、プライバシー確認等）を主要なチェックポイントとして設計し、追加のゲートを増やしすぎずに取りこぼしを減らす

継続的な改善：体制・知見を広げ、運用を更新し続ける

ヒアリングでは、取り組みの拡張として、例えば次が挙げられた。

- AI レッドチーミング体制の拡充
- 社員向けチュートリアル
- 開発方針事例の拡充
- 外部機関との連携

生成 AI のリスクと社会環境は固定されないため、運用を“作って終わり”にせず、更新し続けられる状態を維持することが重要であるという示唆が得られた。

## 5.5 教育

### 組織的リテラシー向上の必要性

AI エージェントを組織全体で効果的に活用していくためには、単なるツールの導入にとどまらず、その活用法とリスクを正しく理解することが不可欠である。そのため、多くの企業で AI ガバナンスチームが主導して積極的な社員教育を展開し、組織的なリテラシーの向上を図る例が確認できた。

### 教育プログラムの取り組み

AI エージェントの教育プログラムの大きな特徴は、エンジニアだけでなく、ビジネス職も含め全社でリスクリングを実施しているケースが見られる点である。すなわち、教育プログラムの対象が AI エージェント開発者のみか、全社員向けかで分類できる。

全社員向けの教育プログラムは、典型的には、AI の活用事例や社内ガイドラインの共有が中心となる。具体的には、研修や勉強会を通じて AI エージェントの活用事例やガイドラインについて共有している事例が見られた。他にも、ハルシネーションやセキュリティといったリスクに関する内容が含まれるケースも存在した。このような教育プログラムが行われる背景としては、リスクを明示することで、漠然とした不安感を払拭し、社員が安心して生成 AI・AI エージェントを活用できるようになるためと考えられる。ただし、リスクばかり強調すると社員が萎縮してしまう恐れがある。そのため、「リスクがあるから利用を制限する」のではなく、「リスクを正しく理解することで、安全な利用を拡大する」という前向きなメッセージを発信し、活用を推進する姿勢が確認できた。

AI 開発者向けの教育プログラムの事例としては、リスクに注力している例が多く見られた。たとえば、社外で起きたリスク事例の共有や、開発時の具体的なリスクの対策方法をレクチャーしている事例が存在した。なお、一方的な教育だけではなく、担当者間のノウハウの共有や交流を重視している例もみられた。これは、現場同士の横のつながりを強化し、ナレッジを循環させる仕組み作りを行なっているものと考えられる。

興味深い例としては、チーム内に AI の専門家がいなかったため、実践しながら専門知識を習得していった事例も存在した。この例では、AI に関する知識習得を始める際に、プロンプトのガイドを読み込むことから始めたことが確認された。このように、AI 活用においては、実践を通じて知識を習得することも重要であると考えられる。

## 教育プログラムの事例

教育プログラムに積極的に取り組んでいる事例として、LINE ヤフーがあげられる。同社では、AI の利活用と AI ガバナンスの浸透を目的に、従業員1万人以上を対象とした大規模な教育プログラムを展開していた。

同社では従業員に対して、AI ツールの提供と同時に、リスク管理やプロンプト技術に関する e ラーニングの研修を行っていた。この研修では、AI の特性や効果的な活用方法といったポジティブな側面だけでなく、情報漏洩や権利侵害、ハルシネーションへの注意喚起など、安全に利用するためのリテラシー向上に重点が置かれている点が特徴的であった。また、これらの研修は一度きりではなく、社会情勢や技術の進展に応じて半年ごとに見直しが行われ、継続的なアップデートが図られていた。

また、全社員向けの基礎教育に加えて、AI 開発者に向けて生成 AI のリスクに対する低減策や関連する海外法規制についての研修も行っていた。このように、多くの先行事例に共通しているのは、役割や部門に応じた段階的な教育体制を構築している点である。

セガサミーでも積極的に教育を推進している事例が確認された。同社では、具体的な業務での生成 AI 活用に繋げるために、AI の推進担当が現場に赴き、その場で現場の具体的な業務課題を解決する体験型ワークショップ形式の教育プログラムを提供していた。それは全社員向けの画一のプログラムではなく、業務での活用を見据えた AI 活用支援（5.4 節）にも近い内容である。このプログラムは「プロンプトキッチン」という名称で行われており、各部門のニーズに合わせて研修内容をカスタマイズし、業務内容に応じた生成 AI の活用方法やプロンプトの事例を紹介することで、実務に直結するスキルを習得できるように工夫されていた。

サイバーエージェントでも全社的な教育に取り組んでいる。同社ではエンジニア職だけでなく、ビジネス職も含めた全社員を対象に「生成AI徹底理解リスクリング」を実施し、生成AIの基礎知識を身につけられる e ラーニング形式のプログラムを展開した。これにより、組織全体の AI リテラシーを底上げし、漠然とした不安を解消することで、安全な AI 活用を加速させる土壌を醸成していた。

## 5.6 ISO 42001 による認証

本節では、AI エージェントの開発・運用プロセスが適切であることを第三者が証明する「認証」の役割と、その実践事例について述べる。

これまでも IT システムについては「認証」の制度が取り組まれてきた。たとえば、ISO/IEC 27001 では情報セキュリティマネジメントシステム (ISMS: Information Security Management System) に関する標準が制定されている。企業は情報漏洩などのリスクに対して適切に対策がなされているか、リスクマネジメントの状況について監査を受けることができ、正常に終了した場合には認証を受けられる。

AI エージェントという新しい技術に対して組織としての説明責任を果たすためには、既存の IT ガバナンスの枠組みを AI 特有の性質に合わせて拡張・再編することが考えられる。実際、ヒアリングにおいて AI マネジメントシステム (AIMS) の国際規格である ISO/IEC 42001 に基づいて監査を受け、認証を取得した組織が確認された。

本節では ISO/IEC 42001 を例とした認証の取り組みについて述べる。

### 5.6.1 取り組む課題：説明責任と信頼性の境界線

#### ガバナンス未整備の現状と将来の指針

大規模言語モデル (LLM) の急速な普及に対し、多くの組織において AI に関するポリシーやガバナンス体制をどのように構築すべきか、具体的な指針や標準的な手法は未だ模索されている段階にある。ヒアリングの結果からも、組織的なルール作りが現場の活用スピードに追いついていないという懸念を抱える組織が少なくないことが示唆されていた。こうした状況において、AI エージェントを B2B 取引や機密情報を扱う業務に適用しようとする際、自社による「安全性の主張」だけでは、ステークホルダーに対して十分な信頼を担保することが難しいという課題が顕在化していた。

AI 特有のリスク（ハルシネーション、バイアス、プライバシー等）に対し、組織としてどのように統制を効かせているかを客観的に示す手法が求められている。ヒアリングにおいては、ISO/IEC 42001 をこうした不透明な状況における指針として活用している事例が確認された。

#### 「管理されたプロセス」の証明

AI システムの信頼性を担保するアプローチには、大きく分けて 2 つの立場が考えられる。

まず、個々の AI モデルやシステムの出力性能・安全性を直接評価し保証する「プロダクト認証」的なアプローチが考えられる。これはたとえば、典型的には特定の性質を測定するためのベンチマークテストの結果により達成される。別のアプローチとして、AI を開発・運用する組織の管理体制（マネジメントシステム）を評価し保証する「プロセス認証」が考えられる。ISO/IEC 42001 はこちらに該当する。AI エージェントを活用する組織の信頼性を客観的に証明するためには、個々の出力結果を評価するだけでなく、「発生し得るリスクを適切に評価し、それを管理・改善し続けるマネジメントシステム（仕組み）が組織的に確立されていること」を証明するプロセス認証のアプローチが必要と考えられる。

既存の情報セキュリティマネジメントシステム（ISMS）などの知見を活かしつつ、不確実性を含む AI 技術をいかに組織の管理下に置くかという点が、信頼性を担保する上での大きな課題となっていた。

## 5.6.2 一般的な対策：ISO/IEC 42001（AIMS）

### 従来の IT ガバナンス（ISMS）の拡張

ISO/IEC 42001 (AIMS) は、ISO/IEC 27001 (ISMS) 共通のマネジメントシステムの枠組み（フレームワーク）を採用している。AI 特有のリスクを対象としてはいるものの、PDCA サイクルやマネジメントシステムの枠組みは 27001 と同一の構造で構築されている。

従来のセキュリティ評価が主に「機密性・完全性・可用性に対するネガティブなリスク」に集中していたのに対し、AIMS では AI 導入による「ポジティブな変化（便益）」についても評価対象とする。AI を単なる脅威としてではなく、組織にどのような価値をもたらすものとして導入するか、その立ち位置を明確にし、継続的に改善することが求められる。

### 知識整備としての認証活動

社内規程や文書の整備は、単なる認証取得のための事務作業ではない。ヒアリングを通じて、認証活動は「社内の AI 利用に関する暗黙知を言語化し、組織的な知識を体系化する機会」として機能することが確認された。曖昧であった AI の利用方法や判断基準を文書化し、第三者の客観的な審査を受けるプロセス自体が、組織的な AI リテラシーとガバナンス能力の向上に寄与する。

### 国内ガイドラインとの接続

ISO/IEC 42001 は国際標準であるが、ここでは国内の主要な指針との関連を確認する。

- AI 事業者ガイドラインとの整合性: 経済産業省・総務省「AI 事業者ガイドライン」が提唱する、環境分析やゴール設定を繰り返す「アジャイル・ガバナンス」のサイクルは、ISO 42001 の PDCA サイクルと整合すると考えられる。

- 評価観点ガイドにおける位置づけ: IPA「AI セーフティに関する評価観点ガイド」においても、組織的な取組方針や規定を評価する「マネジメント的評価」の具体的な参照先として ISO 42001 が示されている。技術的な評価活動とマネジメントシステムを統合して運用することの重要性が示唆されている。

### 5.6.3 具体的な取り組みと事例：アイレット株式会社

#### 企業・サービス紹介

アイレット株式会社は、クラウドインテグレーターとしてクラウド導入支援やシステム開発を展開しており、顧客企業への生成 AI 導入支援も積極的に行っている。

#### 具体的課題

生成 AI 領域の進化は極めて速く、法規制やセキュリティ上のリスクも日々変化している。同社にとって、顧客に対して自社の AI 活用・提供プロセスが安全かつ適切に管理されていることを、客観的な基準で証明し、信頼を獲得することが大きな課題となっていた。

#### 具体的対策：実効性を重視した AIMS の構築

同社は 2025 年 10 月 15 日付けで ISO/IEC 42001 認証を取得している。認証プロセスにおいては、ISMS の既存の IT ガバナンス資産を最大限に活用しつつ、AI 特有の管理手法を導入していた。

以下に ISO/IEC 42001 の取得において特徴的な点を記す。

- デパートメント・アクティビティの明記: どの部門が、どのような業務（アクティビティ）で AI を使うのかを詳細に定義した。同社では、部門ごとの利用目的、利用する AI サービスとそのプラン、オプトアウト設定の有無などを詳細にマッピングした「社内 AI 利用管理表」を整備した。この「使い方の特定」を行うことで、AI リスクを具体的な管理策へと落とし込むと同時に、価値の提供の仕方を明確にしている。
- 「取り組みファースト」による段階的なスコープ拡大: 技術進化の速さに対応するため、最初からすべての業務を認証範囲（スコープ）に含めることは避けていた。まずはリスクの低い業務や技術検証から始め、ビジネスとして定着し、管理ルールが固まった段階で AIMS の運用フローに乗せて認証対象を拡大していく「適応型ガバナンス」を実践していた。
- ISMS との相乗効果: 規定類の作成にあたっては、既存の ISMS の枠組みをベースに AI 特有のリスク評価（ポジティブ・ネガティブ両面の影響評価）を付け加える手法をとった。これにより、ゼロからシステムを構築する負荷を下げつつ、組織全体のセキュリティ基準との整合性を保っていた。

## 認証取得のプロセスと工数

準備開始から認証取得までにかかった期間は約1年弱であった。内訳としては、規定類の作成に約4か月、本審査後の最終承認プロセスに約2か月を要した。一方で、アイレット社の担当者からは、既存のISMSの知見を活かしながら推進することで、さらなる短期間での実現の可能性もあることもヒアリングにより確認された。

## 得られた成果

認証活動を通じて、組織内でのAI利用における「線引き」が明確化された。どの業務で、どのデータまでをAIに扱わせてよいのかというルールが明文化されたことで、現場が自信を持ってAIを活用できる環境が整い、グレーゾーンを「白」に近づける組織的な能力が向上した。また、AIをどのような業務に用いることで価値が提供されるのかも改めて整理され、組織としてAIで価値を生み出していく基盤も整備されたと考えられる。

## 5.6.4 結論

本節で述べてきた認証のための取り組みは、本書の他の箇所で述べたAIエージェントの開発の方法論や、AIセーフティを確立するための方法論と整合することについて最後に述べる。

AIエージェントの開発においては、専門家を巻き込んだ探索的な取り組みが必要になることを述べた。これは開発・評価・分析のサイクルを通じて、誰にどんな価値を提供するのか明確にしていくプロセスであった。

AIセーフティの確立においても、専門家を巻き込んだユースケースの特定が重要であることを述べた。これはどのような場合に、誰にとって、どのようなリスクがあるのかを特定し、リスクマネジメントを行ってリスクを受容可能な程度に管理する取り組みであった。

本レポートでのAIエージェントの認証においては、ISO/IEC 42001を例に上げた。ISO/IEC 42001の認証を受けた組織の取り組みを具体例として確認したところ、誰が、どのような使い方を行って、どのようなメリットを享受するのか明確にし、どのようなリスクがあるのかを明確にする取り組みであった。

したがって、AIエージェントを開発するための取り組みと、AIセーフティを確立するための取り組み、第三者に認証可能な形でAIMSを確立する取り組みは互いに同一だと整理できる。

一方で、証跡となる文書作成が必要な認証取得は、リソースの限られたスタートアップ企業にとって依然として大きな負担であるという指摘もある。しかし、本書で述べてきたように、安全で高品質なAIエージェントを開発する過程では「ユースケースの特定」や「リスクと便益の検討」が自然に行われていた。これらの開発プロセスにおける検討結果を、日常的に(たとえばチケット管理システム、設計文書の一部として)記録に残しておくことが、将来的に認証を

必要とした際の有力な証跡となり、結果として取得のハードルを下げることに繋がると考えられる。

エンジニアリングの活動とガバナンスの要求を別個のものと捉えるのではなく、日々の開発を通じた「安全への取り組み」そのものを資産化していく姿勢が、AI エージェントの健全な発展にとって重要であると考えられる。

## 5.7 トップのコミット

### 5.7.1 トップのコミットとは

本節でいうトップのコミットとは、今回の調査で観察された先行企業の取り組みにおいて、経営層が「生成AI / AI エージェントの活用を、事業戦略・働き方変革の中核に据える」ことを明確に言語化し、組織が実行できるように優先順位・資源・ルールを継続的に整えていた状態を指す。

観察された事例では、単なるスローガンに留まらず、概ね次の3点が同時に扱われていた。

- 意思決定：何を変えるか（業務・プロセス・プロダクト）と、どこまで踏み込むか（例：全社標準化、必須化）を決める
- 投資：安全に使える環境（ツール・ガイドライン・教育）と、推進体制（役割・責任・支援機能）を用意する
- 継続的関与：導入後もKPI/課題を見て、現場の学びを吸い上げ、ルールや仕組みを更新し続ける

なぜ必要か？

AI活用は「ツール導入」ではなく、働き方・責任分界・業務設計の変更を伴う。一方で、現場が自律的に試すだけでは、次の理由で停滞しやすいことがヒアリングから示唆された。

- 優先順位の競合：短期の売上/開発案件が優先され、横断的な改善（教育、ルール整備、評価/監査の整備）が後回しになる
- リスクの不確実性：情報漏洩や著作権等の懸念から「使ってよい範囲」が曖昧になり、現場が萎縮する
- 局所最適の増殖：部門ごとの個別最適が進み、ノウハウやガードレールが資産化されず、再現性が低い

観察された事例からは、経営層のコミットが「組織として使ってよいという許可」と「実行条件の整備」を同時に与えることで、試行錯誤を継続しやすい状態を作りやすい、という示唆が得られた。

何をするか？

観察されたケースでは、トップのコミットは次のような「宣言→設計→運用」に相当する要素をセットで扱うことで、現場の実行可能性を高めていた。

- 目的の明文化

- なぜAIを使うのか（例：生産性、品質、意思決定速度）
- 何を変えるのか（例：共通業務をAI前提で再設計、プロダクトへの組み込み）
- 対象領域の絞り込みとルール化
  - まずは効果が出やすく横展開しやすい「共通業務」から着手する
  - 迷いを減らすために、短い行動ルール（例：「まずAIに聞く」）を定める
- 安全に使える環境整備
  - 企業向けの利用環境（例：エンタープライズ向け利用、アクセス制御）
  - ガイドライン・プロンプト例・禁止事項の明確化
  - 必須教育（リスク管理とプロンプト基礎）
- 推進体制と支援機能の設置
  - 全社横断の推進チーム（またはタスクフォース）を置き、棚卸しと業務再設計を支援する
  - 各部門に推進者を置き、現場の困りごとを吸い上げる
- 評価と学習の仕組み
  - 使った/使わないを罰するのではなく、良い実践を見える化して横展開する
  - ルール・教材・テンプレートを継続更新し「学習する組織」にする

### 5.7.2 ヒアリングで確認されたベストプラクティス

ヒアリングからは、「トップが言った」だけでは浸透しづらく、“強いメッセージ”と“現場が動く設計”が同時に用意されていたケースほど、展開が進みやすい傾向が読み取れた。

- 時間軸つきの目標を置く例が見られた
  - 例：数年単位で生産性を大きく引き上げる、といった方向性を明確化する
- 全社共通で効く“入口業務”を定義し、行動ルールを短くしていた
  - 「調査・検索」「資料作成」「会議」など、職種を問わず存在する業務から始める
  - ルールを“ワンフレーズ”に落として迷いを減らす（例：「まずAIに聞く」）
- 安全に使える“共通理解”と“共通環境”を揃えていた
  - 使い方だけでなく、情報管理・著作権・ハルシネーション等のリスク理解を含める
  - 利用環境やガードレールが、試行錯誤のブレーキではなく推進要因として扱われる
- 推進体制は、規制者ではなく“イネイブリング（支援）”として設計されていた
  - 横断組織が、テンプレ・教材・相談窓口などを整備し、現場の実装を後押しする
  - 現場の専門性を取り込みつつ、標準化と学習ループを回す
- “罰則ではなく支援・称賛”の運用が観察された
  - 表彰やアンバサダーで好事例を促進する一方、支援（テンプレ、窓口）を併せて提供する

### 5.7.3 個別事例

本節では、ヒアリングで確認された範囲で、各社の「トップのコミット」を支える組織的な設計・運用を参照例として紹介する。

#### LINEヤフーの事例

LINEヤフーは、検索・ポータルサイト、メッセージアプリ、Eコマースなど広範なインターネットサービスを展開する国内最大規模のプラットフォーム企業である。ヒアリングでは、AIガバナンスの担当部門において全社的な生成AIの利活用推進とガバナンス体制の構築を担う担当者に話を伺った。

ヒアリングで確認した範囲では、全社での生成AI活用を前提とした働き方への移行にあたり、「誰でも同じ動きができる入口」と「迷いを減らすルール」をセットで整備していた。

- 経営メッセージ
  - 全社で生成AI活用を前提とする方針を打ち出し、時間軸つきの目標（生産性向上）を示していた
- 入口の定義
  - 職種を問わず共通する業務として、「調査・検索」「資料作成」「会議」を入口に設定していた
- 行動ルールの具体化
  - 調査・検索：まずAIに聞く
  - 資料作成：ゼロベースで作らず、まずAIでアウトラインを作る
  - 会議：議事録はAIで作成、参加者は必要最小限にする（等）
- 安全と実行可能性の担保
  - 利用環境を整備し、リスク管理やプロンプト基礎等の教育を必須化して共通理解を揃えていた
- 浸透の仕掛け
  - 部署ごとの推進者配置や、表彰などの称賛施策で好事例を横展開していた（罰則ではなく支援と称賛を中心にしてきた旨が報じられている）

この事例からは、強いメッセージに加えて、入口業務の選定・短いルール・安全に試せる環境を一体で整えることが、浸透を後押しし得るという示唆が得られる。

#### メルカリの事例

メルカリは、日本最大級のフリマアプリ「メルカリ」を中心に、金融や物流など多角的なサービスを展開する日本を代表するテック企業である。AIガバナンスやAI利活用に関するルールメイキングを行う担当者の方にヒアリングを行なった。

ヒアリングで確認した範囲では、経営が「AI-Native」への転換を明確に言語化したうえで、横断体制を立ち上げ、業務と組織の再設計を推進していた。

- 経営メッセージ
  - 一部の専門職に限らず、全社員の働き方をAI前提で再設計する方針を示していた
- 推進体制
  - 横断的なタスクフォースを短期間で立ち上げ、全社の取り組みを推進していた
- 対象の扱い
  - ツール導入に留めず、業務の棚卸しと再構築（AI前提の再設計）を進める構えが示されている
- 焦点
  - Work Experience（働き方の体験）を再設計するという観点を前面に出していた
- 浸透施策
  - ガイドライン整備や情報共有（All hands、Slack、勉強会等）を通じて、学習と共有を促進していた旨が記載されている

この事例からは、経営の宣言と横断体制を通じて「AI前提で再設計する」ための実行条件を整えることが、全社展開の推進力になり得るという示唆が得られる。

# Appendix 1. 各企業の取り組み事例集

本セクションでは、ヒアリング調査にご協力いただいた先行企業における、AI エージェントの活用・開発に関する具体的な取り組みを紹介する。

各社の事例は、ここまで述べてきた抽象的なベストプラクティスを、自社のコンテキストに合わせて具体化するための道標となることを意図している。

## 事例の記述フォーマット

各社の事例は、比較検討を容易にするため、以下の共通フォーマットに基づいて記述している。

1. 企業紹介
  - ヒアリングにご協力いただいた企業の概要。
  - 自社の業種や規模に近い事例を探すための手がかりとなる。
2. サービス紹介
  - 企業が提供、あるいは開発している AI サービスや機能の紹介。
  - AI エージェントの具体的な適用領域やユースケースを示す。
3. 具体的課題
  - そのサービス・企業特有の課題。
  - 開発・運用において直面した具体的な「不確実性」や「リスク」の実態を明らかにする。
4. 具体的対策
  - その課題に対して実際にどう対策したか。
  - 技術・プロセス・組織の観点から実施された具体的なアクションと、そこから得られた知見を記述する。
  - 本文の記述との関連も記している。たとえば、5.1 と書かれている場合、「5.1 組織体制」と関連していることを示す。

## 協力企業一覧

ヒアリングにご協力いただいた企業の一覧を以下に示す（五十音順、株式会社などの法人格を除く）。

- アイレット株式会社
- 株式会社IVRy
- 株式会社Algomatic
- 伊藤忠テクノソリューションズ株式会社 (CTC)
- 株式会社インテージ
- KDDI株式会社
- 株式会社サイバーエージェント
- サイボウズ株式会社
- サントリーホールディングス株式会社
- 株式会社SmartHR
- 株式会社すかいらーくホールディングス (すかいらーくグループ)
- セガサミーグループ
- SpiralAI株式会社
- ダイキン工業株式会社
- DataRobot, Inc.
- 合同会社デロイト トーマツ
- 日本生命保険相互会社
- PharmaX株式会社
- 株式会社Preferred Networks (PFN)
- 株式会社メルカリ
- Ubie株式会社 (ユビー)
- LINEヤフー株式会社
- 株式会社LayerX

# アイレット株式会社

## 1 企業紹介

アイレット株式会社はクラウド特化型のシステムインテグレーターとしてクラウド導入・活用の総合支援サービス「cloudpack（クラウドパック）」をはじめ、システム・アプリケーションの開発、グラフィック・UI/UX デザイン制作を展開している。また、企業の AI 活用を支援するソリューション群「gaipack（ジーエーアイパック）」の提供を通じ、顧客企業への生成 AI 導入支援も積極的に行っている。

## 2 サービス紹介

同社は「AI 利用者」としての立場で ISO 42001 認証を取得しており、社内各部門における AI 利用のリスク管理体制を構築している。特定の AI プロダクトの提供に加え、全社的な業務プロセスへの AI 統合と安全な運用基盤の確立に取り組んでいる。

## 3 具体的課題

技術進化の速い AI 活用において、一律のルールで制限するのではなく、リスクを許容しつつ活用を促進するための基準策定が課題であった。

- 活用の線引きとグレーゾーンの扱い: 業務で AI を利用することはもはや不可欠になりつつある中で、機微なデータを扱う業務や、コアとなるような事業でどこまで活用すべきかを明確にする必要があった。
- 技術進化への追従: 技術の進化速度に対し、認証取得やルール策定が足かせとならないよう、実務のスピード感を維持しつつガバナンスを効かせる仕組みが求められた。

## 4 具体的対策

既存の ISMS（情報セキュリティマネジメントシステム）の運用をベースに、ISO 42001 を適用して「AI 利用者」としてのリスク評価プロセスを標準化していた。

- 部門活動と AI サービスのマッピング (5.6): 各部門の業務と利用する AI サービスを紐づけ、「どの業務で・どの AI を・どのプランで」利用可能かを定義した。また、その業務でのリスクだけでなくベネフィットも明確にし、AI を組織的に活用するための基盤を整えた。
- 「取り組みファースト」の段階的認証拡張 (5.6): 技術検証を先行させ、事業に活用できると判断された段階で AIMS（AI マネジメントシステム）の運用フローに乗せて認証対象を拡大していく「適応型ガバナンス」を実践していた。

- 既存セキュリティ基準との統合 (5.2): 既存の ISMS で定義された情報機密レベルと AI 利用ルールをマッピングし、データの重要度に応じて利用制限（検証利用のみ可など）を設けた。

# 株式会社IVRy

## 1 企業紹介

対話型音声AI SaaS「アイブリー」の開発・提供を行う企業である。ヒアリング対象者は、エンジニアとして対話システムの設計および品質保証の役割を担っていた。

## 2 サービス紹介

IVR（従来の固定的な音声案内）をもとにした AI による電話の自動応答サービスを提供している。具体的には、営業時間等の質問に答える FAQ 自動応答機能や、レストランの予約希望日時・人数をヒアリングし、API 連携を通じて予約確定まで完結させる機能を持つ。

## 3 具体的課題

音声対話というリアルタイム性が求められる環境特有の課題が存在していた。

- AI 開発の「80%の壁」: 最初から AI への依存を前提としてシステムを設計すると、70～80%の完成度までは早期に到達できるものの、残りの20%（完全な実用レベル）に引き上げる難易度が指数関数的に高くなるという課題があった。
- 外部 API の不安定性: LLM API のタイムアウトやエラー（429 など）により、対話が中断しユーザーが離脱するリスクがあった。
- エコシステムの未成熟さ: 利用しているライブラリのアップデートにより、予期せぬ不安定な挙動が発生していた。
- ユーザーの離脱防止: 案内文が長い、あるいは機械音声の質が低い場合に、ユーザーが即座に電話を切ってしまうという課題があった。
- クリティカルな情報の扱い: アレルギー情報など、誤りが許されない情報の判別と適切な対応が求められていた。

## 4 具体的対策

伝統的な NLP（自然言語処理）の知見と、最新の LLM を組み合わせた堅牢な設計を採用していた。

- エキスパートシステムからの段階的代替 (3.6): 最初から生成 AI ありきで組むのではなく、まずは手動やルールベースで「確実に動作する」堅牢なワークフロー（エキスパートシステム）を構築し、LLM を自然言語処理のツールボックスとして使うような方法で、その一部を徐々に LLM に置き換えていくアプローチを採用していた。これにより、システムのベースライン品質を担保しつつ、生成 AI を段階的に取り込んでいた。

- 多層防御とフォールバック (3.3, 3.2, 3.1, 3.5): 出力長が予測可能な構造化抽出においてはタイムアウトを短く設定し、失敗時はリトライおよび別モデルへのフォールバックを行っていた。対話が破綻しそうな場合は、即座に店舗スタッフへ転送する仕組みを構築していた。
- 役割分担によるリスク低減 (4.3): 言語生成（発話）を安易に LLM に任せず、情報の「理解」と「内部状態の更新」に LLM を限定利用し、応答文自体はルールベースで構成することでハルシネーションを防止していた。
- 厳格なコンポーネントテストとシナリオテスト (3.4): 言語理解の精度検証には独自の評価データセットを用い、対話全体については100件を超える膨大なシナリオテストを構築していた。モデルのバージョンアップ時に発生し得る不具合を、デプロイ前に検知できる体制を整えていた。
- 信頼性の高い OSS 選定 (3.6): フレームワークの選定において、広く使われているか、品質が担保されているかを基準とし、LiteLLM や Pydantic AI などを採用してリトライやフォールバックの設定を行っていた。
- 段階的なリリース (4.2): 常識的・倫理的な問題がないか確認後、一部の顧客に先行導入し、実際の利用環境で問題がないことを確認してから全体展開を行っていた。

# 株式会社Algomatic

## 1 企業紹介

Algomatic は、生成 AI を活用した事業開発を多角的に展開するスタートアップである。ヒアリングでは、同社の生成 AI プロダクトであるリクルタAI の開発担当者より、AI エージェントの実装と運用における実践的な知見を得た。

## 2 サービス紹介

AI エージェントを用いた採用支援サービス「リクルタAI」は、求人票作成、スカウト送信、書類選考、面談調整といった一連の採用業務を Agentic Workflow によって自動化し、採用領域に特化したバーティカルな支援を行うことを目指したサービスである。人事担当者の作業負担を軽減しながら、面談機会の創出を支援する点に特徴がある。

## 3 具体的課題

AI エージェントの社会実装において、以下のような特有の課題が確認された。

- 出力の説明責任: 言語モデルの構造上、特定の出力が生成された理由を技術的に簡潔に説明することが難しく、誤生成時の改善プロセスの設計に工夫が求められていた。
- AI への過度な依存: 文章生成の精度向上に伴い、チェック担当者が内容を十分に確認せず送信してしまうなど、「AI 出力への過度な依存」が生じるリスクがあった。
- アルゴリズム回避へのニーズ: 採用領域では「人間味」が重視される場面も多く、最終的なスカウト送信は人が行いたいとクライアントが要望するケースが存在した。

## 4 具体的対策

上記の課題に対し、Algomaticでは以下のようなガバナンス設計および運用上の対策を講じていた。

- 専門家による多層的な評価プロセス (3.4): 現役の人事専門家が AI の回答をリッカート尺度 (5 段階評価) で評価し、そのフィードバックをプロンプト改善へ反映。評価結果を継続的に蓄積してベースラインを構築することで、品質の客観的な担保と継続的な改善を実現していた。
- 専門家手動のプロンプト開発と評価 (4.2): CHRO (最高人事責任者) や現役の人事担当者がプロンプト設計および評価に直接関与し、専門的視点から品質基準を定義していた。

- 継続的な評価基盤とモニタリング (3.5): 過去の選考データから判定軸を自動生成する仕組みを構築するとともに、Slack によるリアルタイムアラートや Looker Studio のダッシュボードを活用し、異常を即時に検知・修正できる体制を整備していた。
- 多層的なリスク低減策 (3.3, 4.3): ローカルモデルを活用した個人情報の自動マスク処理に加え、顧客へ直接 AI ツールを提供するのではなく、Algomatic 社内の専門家が AI ツールを利用し、その結果を顧客に連携する運用とすることで、人による最終確認を前提としたガバナンスを実装していた。さらに、完全自動化と部分自動化の双方の運用方式を併存させ、顧客の方針やリスク許容度に応じてオプトアウト可能な運用設計を採用することで、導入初期の不確実性を抑えながら、安全性を担保した段階的な活用を可能にしていた。

# 伊藤忠テクノソリューションズ株式会社（CTC）

## 1 企業紹介

伊藤忠テクノソリューションズの金融事業グループは、大手金融機関を対象に先端技術を用いたサービス企画や DX 推進の伴走支援を担っている。同グループの金融ネクスト企画部は、AI エージェントやローカル LLM の検証、金融向けプラットフォームの開発を行っていた。

## 2 サービス紹介

銀行向けに行内チャットボットの導入・伴走支援を行うほか、金融機関が求める高いセキュリティ基準を担保しつつ AI 活用を加速させるプラットフォーム「C-NOAH（シーノア）」を展開している。また、コストや対応品質の観点からオンプレミス環境でのローカル LLM 展開の技術検証も実施していた。

## 3 具体的課題

金融業界特有のコンプライアンス意識の高さの中で、従来のシステム評価基準と AI システムの性質の乖離が主な課題となっていた。

- コンプライアンス要件: 業界全体としてセーフティやガバナンスに対する感度が高く、差別的な表現やバイアス、機微情報などの一般的な AI 出力に関する問題から、「必ず儲かる」といった断定表現の禁止まで、多様なコンプライアンス要件への対応が求められていた。
- AI に対する高い精度要求: 一部の顧客は、人間が作業をしても誤りが発生し得るような場面であっても、AI によるハルシネーションなどの誤りを許容しないという、非現実的な期待値を持っていた。
- 既存ガイドラインの不完全性: 金融情報システムセンター（FISC<sup>32</sup>）等の基準は物理インフラやネットワークが中心であり、生成 AI 特有の機能的リスクを網羅しきれていない点が課題となっていた。
- 間接的な波及リスク: 社内オペレーター向けの支援ツールであっても、提示された誤情報が最終的にエンドユーザーへ伝達されるリスクがあるため、社外向けと同様の安全性を担保する必要があった。

## 4 具体的対策

技術的なガードレールと、顧客との合意形成プロセスの両面から対策を講じていた。

---

<sup>32</sup>公益財団法人金融情報システムセンター（FISC）, <https://www.fisc.or.jp/>

- ベースライン比較による品質評価 (3.4): 既存のクラウドサービスや人間が作成した成果物をベースラインとして設定し、AI の出力を客観的に比較することで、顧客が納得できる品質基準の可視化を図っていた。
- イテレーティブな品質定義 (4.2): 業務の自動化ラインが不明瞭な場合は、短期的な試行を繰り返して顧客にフィードバックを求め、段階的に目指すべき品質目標を明確化していた。
- 外部ツール連携による多層防御 (3.3): 自社プラットフォーム「C-NOAH」において、外部のガードレール製品（DynamoGuard 等）を機能要件に応じて柔軟に統合し、安全性を系統的に担保していた。
- 柔軟なガイドラインの共創 (5.2): クラウド利用を前提とした独自のセキュリティ基準を、顧客の組織文化に合わせて数ページから数十ページの異なる規模で柔軟に策定・提供していた。
- アジャイルな開発プロセス (4.2): 技術の変化が激しいことを前提に、中長期のウォーターフォールではなく、工数・期間内でプロトタイプ作成から柔軟に改善を回し、半年後の技術動向にも追随できる体制を構築していた。

# 株式会社インテージ

## 1 企業紹介

インテージは、マーケティングリサーチおよび市場調査の国内最大手企業である。同社のデジタル戦略本部では、全社的な AI 利活用戦略の立案、セキュアな開発環境の整備、および AI ガバナンス体制の構築を担っていた。

## 2 サービス紹介

アンケート集計ツール「Lyche-Board」におけるクロス集計表の解釈・コメント生成機能などを提供していた。

## 3 具体的課題

マーケティングリサーチという事業の性質上、以下のような特有の課題を抱えていた。

- 機密性の高いデータの取り扱い: 個人情報や顧客企業の未発売商品情報等の機密性の高いデータを取り扱うため、一般的なクラウドサービスへの安易なデータ投入は許容されない。
- 「データ提供者」としてのリスク管理: 自社でAIを開発・利用するだけでなく、顧客がインテージから購入したデータを顧客自身のAI環境に投入するケースが増えており、その際のレピュテーションリスクや権利関係の整理が必要であった。
- 形式知化されたリサーチノウハウの統合: 熟練のリサーチャーが持つ高度な分析ノウハウを、いかにしてエージェントのプロンプトやRAG（検索拡張生成）の仕組みに落とし込むかが課題となっていた。

## 4 具体的対策

上記課題に対し、同社は組織・プロセスの両面から以下の対策を講じていた。

- 高機密データ対応の個別環境構築 (3.1): 扱う情報をセキュリティ基準に基づき 3 段階に分類していた。秘密情報や個人情報を含む「レベル 3」の機密データについては、プロジェクト別に Azure OpenAI Service の個別環境や AWS Bedrock を整備し、エンタープライズレベルのセキュリティ要件を満たす個別環境を構築していた。
- 役割（ロール）別の責任明確化 (5.1): 全従業員を「サービス提供者」「データ提供者」「利用者」の 3 つのロールに分類していた。特に「データ提供者」という独自の役割を定義することで、外部 AI へのデータ提供に伴うリスクを管理する体制を整えていた。

- AI リスク判定プロセスの導入 (4.3): EU AI Act の考え方を自社向けにアレンジし、ハルシネーション、著作権侵害、セキュリティ等の観点からプロジェクトを評価する「AI リスク判定プロセス」を導入していた。
- 活用委員会による LLM の業務適用方法の探索 (5.4): LLM 活用に関わる方針・計画立案を目的に「LLM 活用委員会」を設置していた。その下で複数の分科会が、対話性インターフェースを用いた生産性向上の試みや、LLM を用いた分析手法の確立および社内展開を行っていた。

# KDDI株式会社

## 1. 企業紹介

KDDI は日本の大手電気通信事業者である。同社はさまざまな方面で事業を展開しており、データ&AI センター等の専門組織を擁し、社内 DX、ToB、ToC の全方位において生成 AI の活用とガバナンスを推進していた。

## 2. サービス紹介

法人顧客向け、個人顧客向け、社内DXと全ての分野で幅広く生成AIをサービスに活用。特にリスクが顕在化した、画像生成による「提携映画のキャラ創作広告施策」や、採用・人事評価における「人事分野の AI 活用（HR Tech）」を紹介。

## 3. 具体的課題

- 未知のリスクへの対応: 従来のセキュリティやプライバシーの知見だけでは対応しきれない、ハルシネーションや倫理的バイアス、AI 特有の攻撃手法といった幅広いリスクへの対処が必要であった。
- 公平性の担保（人事分野）: 採用や評価において AI を活用する際、公平性の担保が極めて難しく、当初は慎重な姿勢を余儀なくされていた。
- 「活用しないことによる機会損失」のリスク: 安全性を重視するあまり、AI を利用しないこと自体がイノベーションの遅れや競争力の低下という大きなリスクになるという認識があった。

## 4. 具体的対策

- 企画段階での AI 開発影響評価 (AIA) の導入 (4.3, 5.2): 2023 年より、新規サービス立案時に AI 原則への適合性を評価するチェックプロセスを導入した。これにより、リスクに応じた対策を早期に反映できる体制を構築していた。
- 攻めと守りを両立する「ステアリング」としての事務局 (5.1): AI 事務局は利用を制限する「ブレーキ」ではなく、安全に運転するための「ステアリング」として、事業部門が自主的にリスクマネジメントを行えるよう伴走支援を行っていた。
- 多層防御による運用監視 (3.1, 3.5): 画像生成広告施策では、利用規約での免責、NG ワード対応、さらに運用段階での人手による出力モニタリングといった多層的なガードレールを敷いていた。
- 公的指針を活用したリスクの受容 (4.3, 5.2): 政府の調査研究やガイドラインを積極的に参照することで、グレーゾーンだった人事分野等においても「どのような配慮をすれば許容可能か」を明確にし、具体的な導入に踏み切っていた。

- AIセキュリティポータル構築と教育 (3.4, 5.5): AI への攻撃手法を体系化した「AI Security Map」の公開や、全社員向けの E ラーニング実施を通じて、組織全体の AI リテラシーとセキュリティ意識を底上げしていた。

# 株式会社サイバーエージェント

## 1 企業紹介

インターネット広告事業、メディア&IP事業、ゲーム事業などを幅広く展開するIT企業である。子会社である株式会社タプルにてBIチームのデータ抽出業務を効率化するAIエージェント開発を担うエンジニアや、株式会社CAMにて占いAIサービスの開発・安全性検証に携わったエンジニアにヒアリングを行った。

## 2 サービス紹介

タプルでは、自然言語でSQLを生成・実行し、BIツールのようにデータを抽出できるAIエージェントを提供している。CAMでは、占いのロジックに基づきAIがユーザーの悩みを深掘りして鑑定結果を返す「占いAIチャット(UranAI)」を展開している。

## 3 具体的課題

社内向けシステムと社外向け対話型サービスそれぞれにおいて、以下の特有の課題が存在した。

- データ抽出依頼の集中と精度不足: BIチームにデータ抽出依頼が集中し、施策検証のスピードが停滞していた。また、単一のLLMではSQLの構文エラーやテーブル名の誤記が原因でエラーが発生することが多かった。
- エンドユーザーからの不適切入力と安全性確保: 占いというサービスの特性上、ユーザーからセンシティブな悩みや個人的な相談が寄せられることが予見された。Azure OpenAI Service等の標準的なフィルタリングでは、サービス上許容すべき相談まで遮断される懸念があったので、それだけに頼り切ることはできなかった。一方で、悪意あるプロンプトインジェクションへの対策も不可欠であった。

## 4 具体的対策

技術的な多層防御と、徹底した検証プロセスによって課題を解決している。

- 複数モデルによるクロスバリデーション(3.6): OpenAIとGoogleのモデルを併用し、生成されたSQLをLLM-as-a-Judgeで評価。構文エラーやスキーマ名のミスが減点方式で判定し、精度の高い出力を選択する2本立ての構成とした。
- 多段構成による入力分離と防御(3.3): ユーザー入力を深掘りするモデルと、占い結果を出すモデルを分離。前段で情報を抽出し、後段には要約された情報のみを渡す設計により、プロンプトインジェクションが直接占いロジックに干渉するリスクを低減した。

- 独自の安全基準によるフィルタリング(3.4): 独自の判断基準を定義したプロンプトと標準フィルターを併用。占いの文脈で許容される表現（恋愛相談等）は許容するように調整を行った。
- 外部ベンチマークとレッドチーミングによる検証(3.4): 既存の攻撃データセットに加え、占い特有の攻撃ケースを独自に作成。また本番リリース前にベータ版での無料公開を経て安全性を確認した。
- 全社的なAIリスクリングと教育(5.5): エンジニアだけでなくビジネス職も含め、ハルシネーションやセキュリティリスクに関する全社的な研修を実施し、組織全体のAIリテラシーを向上させた。

# サイボウズ株式会社

## 1 企業紹介

チームのコラボレーションを支援するグループウェアの開発・販売を行う企業である。今回の事例では、同社のグループウェア「Garoon」の開発およびQA（品質保証）を担うチームが、生成AI機能の実装に取り組んだ。

## 2 サービス紹介

「Garoon」内に、登録された予定の要約、通知に対するアクションの要否判定、および社内の掲示板やメール等の情報を元に回答する「AIアシスタント（RAG）」の3つの機能を、社内リリースを前提として開発した。

## 3 具体的課題

- 評価指標の不確実性: 従来のソフトウェアテストと異なり、AIの回答には無限の入出力が存在するため、何をもちて「テストOK」とするか判断が極めて困難であった。
- 検索と生成の不安定性 (3.4) : RAGにおいては「正しい情報の検索」と「適切な文章の生成」という2つの不安定な要素が組み合わさっており、品質保証の難易度が高い。
- 実データを用いた検証の制約 (3.1, 4.3) : Garoonには顧客や社内の機密情報が含まれるため、セキュリティルール上、QAチームであっても実環境のデータを直接見て検証することができない。

## 4 具体的対策

- QAエンジニアによるプロンプト改善 (4.2) : 従来のQAの枠組みを超え、QAエンジニアが主体となってプロンプトの調整を担当した。エンジニアが用意した実験用APIを活用することで、コード変更なしに試行錯誤を行える環境を構築した。
- AI精度とプロダクト品質のテスト分離 (3.4, 4.1) : ボタンの動作等の「機能テスト」と、AIの回答精度を測る「精度検証」を明確に分離した。精度検証はシステム完成を待たずに早期から集中的に実施された。
- RAGASとPromptfooを用いた自動評価 (3.4, 3.5) : RAGASの4指標（Faithfulness等）を導入し、当初手動だった評価を実験管理ツールで自動化・高速化した。また、生成のゆらぎを考慮し、複数回実行して平均を取る評価手法を採用した。
- 社内フィードバックの活用 (4.1) : 実データを見ることのできない制約に対し、社内SNSの文化を利用したユーザーからの感想やフィードバックを収集・分析し、改善サイクルを回した。

- 早期のセキュリティ対策（4.3）：セキュリティチームが初期段階から参画し、プロンプトインジェクション等のAI特有の攻撃に対する検証と対策を実環境リリース前に実施した。

# サントリーホールディングス株式会社

## 1 企業紹介

サントリーは酒類や飲料、食品の製造・販売をグローバルに展開する大手飲料メーカーである。ヒアリング対象者は、社内の AI 活用基盤を整備・公開するチーム、および生成 AI 利用に関するルール策定やガバナンスを担うチームの担当者であった。

## 2 サービス紹介

社内向けに、各部署が生成 AI を自力で業務システムへ組み込めるよう、開発ガイドラインや技術スタックを整備・提供している。単なるツールの提供に留まらず、個別業務への適用を支援する PoC の伴走や、ガバナンス体制の構築を通じた「開発・活用支援基盤」としての役割を果たしていた。

## 3 具体的課題

- シャドー IT の顕在化: 会社が認めていない外部 AI サービスの利用が散見され、情報漏洩やセキュリティ上のリスクが高まっていた。
- ガバナンスとスピードの両立: セキュリティや法務面での「守り」の懸念が強く、現場の「攻め」の活用 (SaaS 導入など) が滞りやすい点が課題となっていた。
- ガイドラインの具体化: 抽象的な方針だけでは、システム開発現場で具体的にどのような技術的対策を講じるべきかが不明確であった。

## 4 具体的対策

- ガバナンス専門チームの設置 (5.1): セキュリティや法務的懸念をカバーするため、基盤整備チームとは別にルール・ガバナンスを専門に扱うチームを立ち上げ、統制を強化していた。
- 多層的なガイドラインの整備 (5.2): 2024 年秋に「AI 基本方針」を外部公開し、それを具体化した「AI 利用ガイドライン」を 2025 年 4 月・5 月にイントラサイトで公開した。
- 責任所在の明確化 (5.1): 従来のシステムガバナンスルールを基に、AI 利用における最終的な責任の所在は利用側の「事業部署」にあることを明確に定めていた。
- マインドセットの変革支援 (5.4, 5.7): SaaS を積極的に活用する世の中の動向を、保守的な考えを持つステークホルダーに継続的にインプットし、攻めの活用が進みやすい土壌を醸成していた。
- 段階的な利用拡大 (4.1): 2023 年 6 月までは一部での試行にとどめ、その後全社公開へと段階的に対象を拡張することで、安全性を確認しながら展開していた。

# 株式会社SmartHR

## 1 企業紹介

SmartHR は労務管理のクラウドサービス群を祖業とし、従業員データを活用した人事評価、学習管理といったタレントマネジメントシステムの提供に加え従業員ポータルなどの新規事業を幅広く展開している。ヒアリング対象者は、同社プロダクトへの AI 導入全般を統括する責任者であった。

## 2 サービス紹介

履歴書をマルチモーダル LLM で読み込み登録を効率化する OCR 機能や、社内規定ドキュメントに基づき従業員の問い合わせに自動回答する「AI アシスタント機能」を提供している。

## 3 具体的課題

バックオフィス業務や人事評価という、規則や金銭、極めて機密性の高い個人情報を扱うドメイン特有の課題が存在していた。

- 出力精度の保証とリスク管理: 生成 AI の出力は確率的であり、誤った情報が提示された際のリスクが、他領域のサービスに比べて大きかった。
- プライバシー保護への強い懸念: 従業員の個人情報を扱うため、ユーザーの心理的な安全性をいかに確保するかが重要な論点となっていた。
- 実運用における非機能要件の充足: 年末調整などの繁忙期における大量のリクエスト（スパイクアクセス）への耐性や、業務効率化を損なわない低レイテンシーの維持が求められた。

## 4 具体的対策

技術的な検証から組織的なガバナンスまで、多層的な対策を講じていた。

- 厳密なオフライン評価の徹底 (3.4): 膨大なテストデータを作成し、定量的な精度測定を実施。許容水準に達するまで検証を繰り返すことをリリース必須の要件としていた。
- テストデータ作成の内製化 (3.4): 質の高いテストデータを確保するため、社内の専門チームを育成し、データ作成プロセスを内製化していた。
- 100% オプトイン方式による提供 (3.1): プライバシー保護を最優先し、従業員データを利用する場合は利用者の明確な同意を得た場合のみ AI 機能を提供する設計を採用していた。

- 多層防御とレッドチーミングの実践 (3.3): クラウドプロバイダーの標準的なガードレールに加え、社内 QA チームがレッドチーミング（攻撃的なプロンプトの投入）を行い、安全性を検証していた。
- 現実的な制約に基づくモデル選定 (3.6): モデルの賢さだけでなく、コスト、レイテンシー、トラフィック耐性を総合的に判断し、Google Gemini や Azure OpenAI などのモデルを使い分けていた。
- 独自ガイドラインの策定と組織文化 (5.2): 政府ガイドラインを参考にしつつ、法務・セキュリティ・広報を巻き込み、自社のデータ特性や人事評価という文脈に合わせた独自の指針を運用していた。

# 株式会社すかいらーくホールディングス（すかいらーくグループ）

## 1 企業紹介

和洋中をはじめとする多様なジャンルの飲食店チェーンを全国に展開する、日本最大級のフードサービス企業である。DX推進の一環として、顧客体験の向上と従業員の業務負荷軽減の両面で生成AIの活用を進めている。

## 2 サービス紹介

Azure OpenAI Service や Gemini を活用し、タブレットのAIが顧客と対話しながら最適なメニューを提案する「Co店長」を提供している。また、その技術を応用し、従業員向けの調理マニュアル支援や店舗管理への活用など、社内向けのスピノフ機能も展開している。

## 3 具体的課題

- 高リスク領域の制御（アレルギー対応）：飲食店においてアレルギー情報は命に関わる重要事項であり、AIによる誤回答（ハルシネーション）は許されない。
- ユーザーからの攻撃的入力：不特定多数の顧客が利用するため、プロンプトインジェクションや悪意のある入力への対策が必要であった。
- 品質評価の基準策定：「Co店長」としてどの程度の正確性があればリリース可能か、実店舗での運用に耐えうる品質基準の定義が困難であった。

## 4 具体的対策

- 部門横断での徹底的なリスク洗い出し（4.3）：品質管理、営業、総務など各部門の専門家を巻き込み、リスクの高い問い合せを特定。とくにアレルギーに関しては「AIは絶対に回答せず、サイトを案内する」という挙動を、確信が持てるまで何度も検証した。
- アジャイル型的高速プロトタイピング（4.1, 4.2）：約半年間の開発期間中、開発・テスト・評価のサイクルを高速に回転させた。現場を知る専門家が評価に参加し、イテレーションごとに回答精度を磨き上げた。
- 「見習い」設定とリスク受容（4.3）：AIに「見習い中」というキャラクター設定を付与することで、ユーザーがAIの不完全さを許容しやすい心理的な土壌を作った。これにより、AIの限界をオープンにした上での運用を可能にしていた。
- 実店舗でのカナリアリリースと常時監視（4.2, 4.4）：限定的な店舗から順次導入し、初期段階では現場に担当者を配置。対話ログをリアルタイムでモニタリングし、攻撃的な入力に対するガードレールの有効性を確認した。

- 「事故で止めない」ガバナンス体制の構築（5.1, 5.2）：早期にAI利活用のガイドラインを整備。トップマネジメントの強い推進力のもと、万が一問題が起きてもプロジェクト全体が停滞しないよう、リスクを許容しつつ迅速に改善する組織文化を形成した。

# セガサミーグループ

## 1 企業紹介

セガサミーグループは、エンタテインメントコンテンツ事業、遊技機事業、またゲーミング事業を主に展開する総合エンタテインメントグループである。今回はグループの持株会社であるセガサミーホールディングス株式会社の AI ガバナンス担当者と、グループ会社の株式会社セガの生成 AI 委員会担当者にヒアリングした。

## 2 サービス紹介

Microsoft社のTeamsチャットをユーザーインターフェースとして、Azure OpenAIのモデルを選択して利用可能なチャット機能と、社内情報検索用のRAGアプリケーションを、グループ社員約8,000名に展開している。

## 3 具体的課題

大規模組織ゆえの生成 AI 関連のアップデートの周知の難しさと、クリエイティブ業界特有の権利リスクが課題となっていた。

- ガイドラインの現場への浸透: 制度を整えても、現場の業務部門までその内容や活用メリットが浸透しきれず、関心を持たれにくい点が課題であった。また、AIサービスによっては入出力のデータの権利を奪う・学習に使うケースがあり、情報漏洩を発生させないためのルールが求められていた。
- 技術変化への適応: AI モデル自体の頻繁なアップデートや AI アシストツールの導入に対して、組織としてどう即応していくかが問われていた。

## 4 具体的対策

「1.5 線」として機能する専門組織の設置と、ボトムアップの啓蒙活動を組み合わせていた。

- 生成 AI 委員会の設置 (5.1, 5.3): 開発者と知財専門家からなる「生成 AI 委員会」が、相談窓口として全方位的な質問に回答していた。これにより、現場との厚い信頼関係を築きながら安心・安全な生成AIの活用を推進していた。
- プロンプトキッチン (5.4, 5.5): 単なる事例紹介ではなく、各部署の具体的な業務に即したプロンプトを考案・実演するイベントを開催していた。通算で約 2,100 名の社員が参加するなど、当事者意識とリテラシーの向上に成功していた。
- 透明性の高いガイドライン運用 (5.2): 2023年3月に生成AIに関するグループ共通のガイドラインを策定。定期的な見直しに加え、外部環境の変化やインシデント発生時には適宜改訂を行う運用とし、EU AI Actなどの規制動向を踏まえて内容の更新を行っていた。

# SpiralAI株式会社

## 1 企業紹介

SpiralAI は、独自の性格設定や価値観を持つキャラクター AI の開発および、それらを構築・運用するためのプラットフォーム「Geppetto」を提供している。同社は、芸能人やアニメ IP（知的財産）を活用した対話型 AI サービスの構築において、キャラクター性と安全性の両立を追求していた。

## 2 サービス紹介

主に、ユーザーが AI キャラクターと対話できる「Geppetto」や、実在のタレントをモデルにした「AI 野々村真」などのサービスを展開している。これらは、特定の IP の価値観を反映した自然なコミュニケーションを可能にする一方で、ブランド保護のための厳格な制御が組み込まれていた。

## 3 具体的課題

芸能人や有名 IP を扱うサービス特有のリスクや、生成 AI の技術的特性に起因する課題が挙げられていた。

- ブランド毀損のリスク: 芸能人や IP の著作権を借りて対話を行うため、AI が不適切な発言（卑猥な内容や虚偽情報など）をすることで、IP 元のブランド価値を著しく損なう懸念があった。
- ユーザーによる悪意ある誘導: ユーザーがプロンプトインジェクション等を用いて、公序良俗に反する発言を AI から引き出そうとする試みが想定されていた。
- 小型モデル特有の難しさ: コスト効率を優先して 8B クラス等の小型モデルを使用する場合、「～と言わないでください」といった否定的な指示が正しく解釈されず、逆に禁止ワードを生成しやすくなる現象があった。
- キャラクター性と安全性のジレンマ: AI が「面白さ」や「本人らしさ」を追求するほどリスクが増大し、逆に安全性を過度に重視すると魅力が損なわれてしまうという、正解の定義が難しいバランス調整が課題となっていた。

## 4 具体的対策

課題に対し、多層的な防御策を講じていた。

- 「4 つの盾」による多層防御 (3.3, 3.6): 1. 望ましい対応の記述、2. 強化学習を用いた望ましくない対話の抑制、3. ルールベースの NG 単語フィルター、4. 手動介入という 4 段階のガードレールを設けていた。

- 自然な対話を実現するための技術的工夫 (3.1): ユーザーとの自然な対話を実現するため、推論速度の速いモデルを採用し、システムプロンプトの工夫と組み合わせて「速くて安全」なレスポンスを実現していた。
- モデル特性に合わせたプロンプトエンジニアリング (3.6): 指示に従う能力が低い小型モデルに対しては、短い文章を用いたり、否定表現を排除して肯定的な動作指定に置き換えたりするなど、モデルの振る舞いに合わせたプロンプトエンジニアリングを行っていた。
- 不確実性を前提とした改善ループ (4.2): 開発段階では「やってみないとわからない」という性質を受容し、自社でのドッグフーディングや SNS 上の反応を詳細に観察していた。そこで得られたユーザーの反応や、本人の主観とは異なる「周囲からの本人らしさ」の評価などを元に毎日チームで議論を行い、安全性とキャラクター性の最適なバランスを探索しながら品質基準をイテレーティブに構築していた。

# ダイキン工業株式会社

## 1 企業紹介

空調機および化学製品の世界的メーカーである。ヒアリング対象のテクノロジー・イノベーションセンター（TIC）は、製造現場や設計開発における生成 AI の活用推進に取り組んでいる。また、過度な外部ベンダーへの依存を避けるため、業界特有の知識に関する継続事前学習を施した独自 LLM の開発・実装を主導している。

## 2 サービス紹介

工場設備の故障原因を 10 秒で診断する設備故障診断エージェント<sup>33</sup>や、社内文書の検索・参照でシステムを展開している。また、映像解析による現場作業の品質向上や、知的財産調査における特許スクリーニング支援など、多岐にわたる社内業務に AI を適用している。

## 3 具体的課題

製造業特有の高度な専門知識の反映と、現場の物理的リスクを伴う業務への適用における安全性の確保が課題となっていた。

- ドメイン知識の欠如: 汎用的な LLM や VLM（画像言語モデル）では、エアコンの分解修理といった製造現場特有の動作を「清掃」と誤認するなど、専門業務の解釈に限界があった。
- ドメイン特化データと機密性: ドメイン特化モデルの継続事前学習に活用できるデータは社外秘の情報を含む機密性が高いものであり、社内向けモデルへの活用が主な目的であっても慎重な取り扱いが必要であった。
- 画一的な評価基準の限界: 初期に想定したユースケースはドメイン知識活用に絞った社内向けのものであり、顧客向けサービスにおける有害な発言など一般的な AI リスクがそこまで深刻とはならないことも想定される。その違いを考慮しない画一的な安全基準を適用すると開発スピードを阻害する懸念があった。

## 4 具体的対策

独自モデルの開発による制御権の確保と、現場主導の柔軟なガバナンス体制を構築していた。

- 独自 LLM の開発とデータ基盤整備 (3.6): 外部 API への過度な依存を避け、安全性を担保するために独自モデルの開発を推進。既存の設計基準文書を学習用データに変換し、空調分野に特化した VLM/LLM を構築していた。

---

<sup>33</sup><https://www.daikin.co.jp/press/2025/20250422>

- ベンチマークデータセットによる品質担保 (3.4): 開発の初期段階で独自のベンチマークセットを作成し、汎用モデルとの比較評価を行うことで、自社特有の要件を満たすためのカスタマイズ指針を明確化していた。
- 専門知識を活用したアノテーション (4.2): 動画データの分類等において、専門知識を持つ外部協力業者と連携。詳細なアノテーション基準を策定し、定期的な見直しと質疑応答を通じて学習データの品質を継続的に改善していた。
- 機動的なガバナンス (5.1, 5.2): 安全側に振り切った硬直的なルールではなく、プロジェクトごとに必要な性能・リスク評価を行う体制を採用。実用性と安全性の両立を目的として、各部署の用途に応じて判断していた。
- 高速な意思決定と環境構築 (5.7): 経営層のコミットメントのもと、GPU 環境の確保や予算配分を柔軟に行い、技術変化に即応できる組織体制を整えていた。

# DataRobot, Inc.

## 1 企業紹介

DataRobot は、AIエージェントのライフサイクルの全段階を強化するプラットフォームを提供している。ヒアリング対象者は、日本における AI & サービス部門の責任者として、生成 AI を組み込んだアプリケーションテンプレートの提供や、顧客企業への AI 導入・活用支援を行っていた。

## 2 サービス紹介

同社は、データの要約、需要予測に基づく意思決定支援、RAG（検索拡張生成）を用いたチャットボットなど、多様な AI エージェントのテンプレートを提供している。これらは企業の既存データと連携し、業務の専門家が迅速にインサイトを得るための補助ツールとして機能する。

ヒアリングでは、他社への生成AI<sup>34</sup>の機能提供<sup>35</sup>が実事例として紹介された。このプロジェクトでは、生成 AI 機能を搭載した現場エンジニア向けの故障診断支援アプリケーションを提供することで、現場エンジニアの意思決定の速度向上及び知識格差の改善が実現された。

## 3 具体的課題

実業務に基づく大規模なワークフローを構築・運用する際には、以下のような課題が想定される。

- コストとレイテンシーの増大: マルチステップの長大なワークフローを構築した場合、消費されるトークン数や実行時間（レイテンシー）が増大し、ビジネス上の許容範囲を超えるリスクがある。
- 保守運用の複雑化: 業務領域をまたぐ大規模なエージェントを構築すると、外部環境の変化や法規制の更新に伴うデータベースのメンテナンス負荷が膨大になり、現実的な運用が困難になる。
- 過剰な権限付与: 単一のエージェントに広範な業務を担わせることで、組織内で本来分割されるべき情報アクセス権限を超えた過度な権限がAIに付与されてしまうセキュリティ上の懸念がある。

---

<sup>34</sup><https://www.datarobot.com/jp/blog/datarobot-supports-noritsus-demonstration-experiment-to-use-generative-ai/>

<sup>35</sup><https://www.datarobot.com/jp/blog/noritsu-aims-to-reduce-customer-support-resolution-time-and-improve-the-efficiency-of-field-engineers/>

- ワークフローの再実行コスト: 長い工程を一括実行する設計では、最終出力が意図に反した場合、全工程を最初から再実行せねばならず、時間的・経済的コストが大きくなる。

#### 4 具体的対策

- コスト・性能の多重監視 (3.5): トークン使用量やレイテンシー、回答性能のバランスを監視するガードレールを設け、意図した通りに利用されているかを常時モニタリングしていた。
- 業務分割と小規模エージェントの推奨 (3.1): メンテナンス性とセキュリティを担保するため、業務ドメインや組織の権限規定に沿ってエージェントの細分化や、利用者の認証トークンによるアクセス権の同期を組み合わせていた。これにより、情報漏洩リスクを構造的に低減していた。
- プラットフォームによる非機能要件の担保 (3.1): 認証・認可やセキュリティといった基盤機能は、自社プラットフォーム (DataRobot) の機能を活用することで、安全性を確保しつつ素早い実装を可能にしていた。
- ユースケースに応じたガードレール運用 (3.5, 4.3): 内部向けの低リスクなユースケースでは、厳密な出力制限よりも、どのような問い合わせが多いかを把握するモニタリングに重点を置き、利便性と統制のバランスを取っていた。

# 合同会社デロイト トーマツ

## 1 企業紹介

世界最大級のプロフェッショナルファームであり、企業の経営課題解決を支援する。ヒアリング対象者は、コンサルタントとして多くの企業のAIガバナンス導入支援やセーフティ確保に携わっている。

## 2 サービス紹介

デロイトでは企業の生成AI活用におけるAIガバナンス体制の構築を支援するサービスを提供している。このサービスでは、抽象的なガイドラインを顧客ごとのリスク許容度に応じた実践的なドキュメントに落とし込み、現場が安全かつ迅速にAIをローンチするためのガバナンス体制の整備を支援する。

デロイトに対するヒアリングでは、このAIガバナンス導入サービスを提供する中で得られた知見として、各企業で共通する課題と対策について伺った。

## 3 具体的課題

多くの日本企業において、AI活用とリスク管理のバランスを最適化する上で以下の課題に直面している。

- 役割と責任（R&R）の不明確さ: 日本企業特有の課題としてロールアンドレスポンシビリティが定まっておらず、また、AIの知見がない担当者が兼務することで、意思決定や作業が停滞する。
- AI活用とリスク管理のトレードオフ: AI活用による価値創出（攻め）と、リスク管理（守り）にはトレードオフが生じやすい。これらのバランスを保つことが難しい。
- 技術進化とガバナンスの乖離: AI技術は日々進化し、確率的な挙動を示すため、従来のITシステムのような固定的なリスク管理手法では対応しきれない。

## 4 具体的対策

現場での実践を通じ、以下の対策を講じている。なお、これらの取り組みはデロイトの取り組みとして各章で紹介はしていないが、ヒアリングで得られた知見を各章の内容に盛り込んでいる。

- 1.5線組織の設置(5.1): 事業部門（1線）とリスク管理部門（2線）の橋渡し役として、専門知識を持つ「1.5線」のガバナンスチームを組織し、他部署を効果的に巻き込む。

- 攻めと守りが一体となったリーダーシップの確立(5.1/5.7): 活用とリスク管理のバランスを最適化するため、攻めと守りの両面を理解したリーダーシップを確立することが重要である。
- アジャイルなガバナンスの構築(5.2): AI技術の進化を前提とし、定期的に外部ガイドラインのアップデートを検知して自社のチェック項目や目標を再設定する。静的な規定ではなく、常に進化・対応し続ける「アジャイルガバナンス」を実践する。

# 日本生命保険相互会社

## 1 企業紹介

日本生命は、国内最大手の生命保険会社の一社である。ヒアリング対象であった IT 統括部デジタル推進室は、全社的なデジタル化の推進や生成 AI の利活用基盤の整備、およびガバナンス体制の構築を担っていた。

## 2 サービス紹介

汎用的な生成 AI 基盤「Azure OpenAI Nチャット」を全内部職員向けに提供し、文章要約、翻訳、アイデアの壁打ちなどに活用している。また、社内規定の問い合わせや営業成績の管理など、特定の業務領域における個別 RAG や SaaS 連携による AI サービスの実装も進めていた。

## 3 具体的課題

金融機関として高度な安全性が求められる中で、生成 AI 特有のリスクと利便性の両立が課題となっていた。

- 生成 AI 特有のリスク管理: ハルシネーション、入力データの学習利用、著作権侵害などのリスクに対し、従来のシステム開発基準だけでは対応しきれない懸念があった。
- 定量的効果測定の困難さ: AI 導入による業務効率化について、仕事の「量・速度・質・幅」の変化を客観的な数値で測る手法が確立されていない点が課題であった。
- ガバナンスとスピードの両立: 技術発展や EU AI Act 等の法整備が速い中で、厳格すぎるルールは現場の活用を阻害し、緩すぎるルールは統制を失うというジレンマに直面していた。

## 4 具体的対策

現場のボトムアップな活用を尊重しつつ、段階的なリスク管理と専門組織による横断的な統制を図っていた。

- 段階的な利用範囲の拡張 (4.1): 当初は利用範囲を一部社員に制限した試行から開始し、リスク管理部署との協議を経て、条件付きで全職員へと利用範囲を段階的に拡大していた。
- Human-in-the-loop の原則化 (3.2): ハルシネーションを 100% 排除することは不可能との前提に立ち、最終的な判断や対外的なアウトプットには必ず人が介在する運用を徹底していた。

- デジタルハッカソンによる課題抽出 (4.1, 5.5): 現場の業務課題を可視化するため、ハッカソン形式で PoC を実施し、本番化に向けた基準設定を収益・人事・システム等の関係部署と連携して行っていた。
- 多層的なチェック体制と外部知見の活用 (5.1): 法務、情報資産、セキュリティ等の分野ごとに主管部署が協力してチェックリストを作成し、コンダクトリスク等の高度な判断が必要な場合は外部専門家の意見を仰ぐ枠組みを構築していた。
- 既存ルールと新ガイドラインの統合 (5.2): AI 事業者ガイドラインや FDUA<sup>36</sup> の基準を参照しつつ、既存の社内ルールに AI 特有の視点を組み込んだ規定整備を進めていた。

---

<sup>36</sup>一般社団法人金融データ活用推進協会 (FDUA) , <https://www.fdua.org/>

# PharmaX株式会社

## 1. 企業紹介

PharmaX は、テクノロジーを活用して「世界で最も患者 / 生活者主体の医療体験を創造する」ことを目指すスタートアップ企業であり、AI エージェントを活用したオンライン薬局サービスである YOJO を展開している。今回はエンジニアリング責任者に取り組みを包括的にヒアリングした。

## 2. サービス紹介

提供サービス「YOJO」は、ドラッグストアで購入可能な OTC 医薬品（主に漢方薬）をオンラインで相談・購入できるサービスである。AI エージェントがユーザーとの対話を通じたヒアリングや服用後の体調管理等のフォローアップを自律的に行い、専門的な判断が必要な場面では薬剤師へシームレスに引き継ぐ体制を構築していた。

## 3. 具体的課題

医療という高い安全性が求められる領域において、AI エージェントの誤回答や不適切なアドバイス（ハルシネーション）をいかに制御するかが最大の課題となっていた。

- リスクの高い相談への対応: 希死念慮、薬の飲み合わせといった、危険度が高い相談内容については、AI が回答することのリスクが非常に高かった。
- リスクの網羅的把握の困難性: 生成 AI の特性上、リリース前にすべてのリスクを洗い出すことは難しく、網羅性を追求しすぎるとサービスの社会実装が停滞する懸念があった。
- 専門知識の言語化と実装: 薬剤師が持つ安全性の観点は複雑であり、エンジニアだけで高精度なプロンプトや評価指標を作成することが困難であった。

## 4. 具体的対策

「小さく始めて徐々に育てる」という原則に基づいた対策を講じていた。

- 専門家による Human in the loop (3.2): OTC 医薬品の購入に関する会話をするのは薬剤師ということが法律で決まっているため、このようなユースケースは、人の承認を挟むようにワークフローを構築していた。
- LLM-as-a-Judge による自動評価 (3.3, 4.2): 評価指標が固まった段階で LLM-as-a-Judge を導入し、一定の基準を超えたプロンプトをリリースするプロセスを確立していた。
- 継続的なモニタリング (3.5, 4.4): LangSmith を用いたトレーシング基盤を構築。分類精度やワークフローの離脱率を可視化する体制を整えていた。

- ハイブリッドなガードレール設計 (3.6): LLM を用いた質問ジャンルの分類を行っていた。この LLM は人手でアノテーションしたデータセットを用いて fine-tuning することで、プロンプトでの指示のみの場合よりも分類精度を高めていた。
- 段階的な LLM の導入 (4.1): 最初は、相談全体の約 7 割を占める「共感フェーズ」や「サービス設計の質問」など、低リスクかつ高頻度な部分に LLM を導入し、高リスクな領域は即座に薬剤師へフォールバックする設計とした。その後、LLM により対応できる範囲を徐々に拡大させていた。
- 専門家によるプロンプト開発と評価 (4.2): 社内の専門家である薬剤師自身がプロンプト作成や数百件規模の定性評価項目の検討、データアノテーションに直接関与。これにより、医療現場の基準に即した精度の高い改善サイクルを実現していた。

# 株式会社Preferred Networks (PFN)

## 1 企業紹介

Preferred Networks (PFN) は、深層学習（ディープラーニング）などの最先端技術を自社開発し、製造、交通、バイオ、エンターテインメントなど多様な分野へ社会実装を行う、日本を代表する AI 企業である。同社内では、AI ガバナンスを専門に扱うチームが各プロジェクトの支援にあたっていた。

## 2 サービス紹介

社内の開発・事業部門を対象とした「生成 AI 活用のためのガバナンス構築および伴走支援」である。社内の各チームが AI を安全かつスピーディに開発できるよう、専門的な立場からガイドラインの策定やリスク評価のサポートを行っていた。

## 3 具体的課題

生成 AI 特有の不確実性に対し、厳格な統制による安全確保と、開発の機動力（アジリティ）をいかに両立させるかが技術管理上の課題となっていた。

- 技術の不確実性と論点の多様性: 入力やモデル更新で出力が変化するため、合格基準が定めにくい。また、著作権、セキュリティ、レピュテーション等の論点が複雑に絡み合うといった難しさがあった。
- 開発速度の低下（アジリティの毀損）: リスク対策を「外付けの承認ゲート」として設計すると、差し戻しによる手戻りが発生し、利活用推進が停滞する懸念があった。
- 専門知識の偏り: 開発現場が必ずしも法務やガバナンスの専門知識を持っているわけではなく、一方で 2 線（法務等）が最新の AI 技術を完全に理解して評価することも困難であった。

## 4 具体的対策

現場の試行錯誤を加速させる「ガードレール」としてのガバナンス体制を構築していた。

- 1.5 線による伴走支援 (5.1, 5.4): 開発部門 (1 線) と法務・監査部門 (2 線) の間を受け持つ「1.5 線」としてガバナンスチームが機能し、論点整理や運用設計を短期間で伴走支援していた。
- 文書体系の階層化 (5.2): 抽象的な「AI ポリシー」に加え、具体的な「AI 品質ガイドライン」、さらに個別プロジェクトごとの「AI 開発方針」を作成することで、原則を現場の実装レベルまで落とし込んでいた。

- 既存の必須プロセスへの統合 (5.2, 5.4): セキュリティチェックやプライバシー確認など、既存の承認手続きの中に AI 特有の観点を統合することで、追加のゲートを増やさずにリスクを拾い上げる工夫をしていた。
- 確認表（チェックリスト）によるリスクの具体化 (5.4): 潜在リスク、求める対策、残存リスクを整理する確認表を用い、プロジェクトが取るべき具体的アクションを明確化していた。
- 継続的な改善と体制の拡充 (5.4): AI レッドチーム体制の拡充や社員向けチュートリアル提供、外部機関との連携を通じ、変化の速い AI 環境に合わせて運用を更新し続けていた。

# 株式会社メルカリ

## 1 企業紹介

メルカリは、日本最大級のフリマアプリ「メルカリ」を中心に、金融や物流など多角的なサービスを展開するテック企業である。AI ガバナンスや AI 利活用に関するルールメイキングを行う担当者にヒアリングを行った。

## 2 サービス紹介

フリマアプリにおける商品説明の自動生成や不正取引検知など、多岐にわたるプロダクトに生成 AI を組み込んでいた。また、社内向けにはデータ分析ツール「Socrates」や、インシデント対応効率化を支援する AI エージェントを導入していた。

## 3 具体的課題

各サービスに生成 AI を組み込むにあたって、以下のような課題に直面していた。

- 規制領域での生成 AI の利用: フィンテック領域のように厳格な法的規制が存在する分野においては、生成 AI を直接的に利用することが困難であった。
- AI 活用の相談先の不透明: 生成 AI の利活用を進める中で生じた質問や相談の窓口が分からず、活用に踏み出せないという課題が発生していた。

## 4 具体的対策

上記の課題に対し、リスクの大きさに応じた多層的なアプローチを講じていた。

- AIガバナンスチームの設置(5.1): 全社的なAI活用が進んだことにより、AI活用にもなるリスクアセスメントとリスク低減、社内ルール整備、啓発活動などを行う専門部署が新設された。
- ワンストップ相談窓口の設置 (5.3): AI 活用に関する質問や相談を一元的に管理する窓口を設け、社員が迷わず相談できる体制を構築していた。これにより、部署間での情報の分散も防止していた。
- ガイドライン策定と全社教育 (5.2, 5.5): 経済産業省や総務省のガイドラインを前提にした「生成 AI 利用ガイドライン」を策定し、全社員への研修を必修化した。これにより、社員が安心して AI を利用できる基盤を整えていた。
- トップのコミットメントとビジョンの提示 (5.7): 経営層が「AI-Native 企業」への転換を表明し、一部の専門職だけでなく全社員の働き方を AI 前提で再設計する方針を明確に示していた。

# Ubie株式会社（ユビー）

## 1 企業紹介

Ubie は医師とエンジニアによって 2017 年に共同創業された、日本およびアメリカでサービスを展開するヘルステック企業である。同社は「テクノロジーで人々を適切な医療に案内する」というミッションを掲げ、生活者の病気の早期発見・早期受診のほか、医療現場の DX 推進や医療従事者の働き方改革を目指し、生成 AI を活用した業務効率化ソリューションを提供している。

## 2 サービス紹介

「ユビー生成AI」において、医療現場における事務作業の効率化を目的として、生成 AI を活用した医療文書（退院サマリ、紹介状、インフォームドコンセント等）の作成補助、医療マニュアルの検索、音声や手書き書類の文字起こしサービスを提供している。

## 3 具体的課題

- 高い安全性と信頼性の要求: ヘルスケアという機微な情報を扱う領域において、ハルシネーションや個人情報の漏洩は致命的な問題となるため、極めて慎重な対応が求められていた。
- 評価指標の不在と定量化の困難さ: 医療現場のニーズは診療科や職種等によりそれぞれ異なり、生成物の良し悪しを従来の機械学習のような一律のベンチマークで測定することが難しい点が課題であった。

## 4 具体的対策

- オンボーディングの実施 (4.1): 生成 AI の安心・安全な活用を推進するため、数か月程度の入念なオンボーディング期間を設け、この期間を通じて、現場での運用改善を繰り返し実施することでシームレスな生成AI導入をサポートしていた。
- 院内向けガイドラインの提供 (4.3, 5.5): 医療現場で安心・安全に生成AIを活用できるような運用支援として、院内向け生成AI活用ガイドラインのひな型等を顧客へ提供していた。
- 専門家による Human-in-the-loop (3.2): サービスを「医療業務従事者の最終確認を前提とした作成補助」と位置づけていた。専門家が内容をチェック・修正した上で確定させるプロセスを徹底することで、ハルシネーションによるリスクを実運用の中で低減させていた。
- LLM-as-a-Judge による品質管理 (3.3): モデルのバージョンアップ時には、新旧モデルの出力を LLM 自体に比較・評価させることで、人手によるアノテーションコストを抑えつつ、迅速な性能検証を実現していた。

- ユースケースの限定とリスク回避 (4.3): 医療事務業務の範囲に提供範囲を限定し、診療判断に関わるような利用をしないようユーザーに周知・徹底することで、リスク回避を行っていた。

# LINEヤフー株式会社

## 1 企業紹介

LINEヤフーは、検索・ポータルサイト、メッセージアプリ、Eコマースなど広範なインターネットサービスを展開する国内最大規模のプラットフォーム企業である。ヒアリングでは、AI ガバナンスの担当部門において全社的な生成 AI の利活用推進とガバナンス体制の構築を担う担当者に話を伺った。

## 2 サービス紹介

同社では、巨大なユーザー基盤を持つ既存サービスに生成 AI を統合しており、例えば、過去のベストアンサーを基に AI が回答を生成する「みんなの知恵袋」を展開している。

## 3 具体的課題

- センシティブな領域におけるリスク: 各種サービスでは多種多様な質問や投稿が行われる。特に、医療や選挙といった専門性・中立性が極めて高いカテゴリで不適切な回答（ハルシネーション等）が生成されるリスクがあった。
- ユースケースを絞り込むことの難しさ: 特定の業務に特化した B2B サービスとは異なり、一般ユーザー向けの対話サービスでは入力内容を事前に制限することが難しく、安全なユースケースのみに絞り込むことが困難であった。
- AI 活用推進とリスクマネジメントの両立: ガバナンスがコントロールや抑制に注力し、画一的なルールを開発現場に押し付けてしまうと、AI を活用した革新的なサービスの誕生を阻害してしまうリスクが懸念されていた。

## 4 具体的対策

- 技術的な対策と契約的な対応の組み合わせによる多層防御 (4.3): AI側でユーザー入力のカテゴリを自動判定するなど技術的な対策も含め、選挙など影響が大きい特定の分野については回答を生成しないように制御する技術的対策と、ユーザーへの注意喚起や免責事項の表示など、契約に基づく対応を組み合わせ、多角的にリスクを低減していた。
- イノベーションを尊重した AI ガバナンスチーム (5.1, 5.3): AI の活用推進をミッションとした AI ガバナンスチームを設置し、1 線部門が自律的にリスク対策を行えるよう支援していた。また、企画・開発の初期段階から、AI モデルの利用や他社サービス導入に関する相談を一括で受け付ける窓口を運用していた。
- 階層化されたガイドラインの提供 (5.2): 「AI 倫理基本方針」を頂点とし、リスク低減策を具体的に例示した「解説ページ」を設けるなど、ガイドラインを階層的に整備していた。これにより、各事業部が自らのサービスの特性に応じて最適な対策を選択できるようにしていた。

- 階層別・定期更新型の教育プログラム (5.5): 従業員1万人以上を対象に、リスク管理とプロンプト技術をセットにしたeラーニングを実施しており、その内容は技術進展に応じ半年ごとにアップデートされていた。
- 経営層主導の生成 AI 活用を前提とした働き方への移行 (5.7): 経営層が生成 AI の活用に関する数値目標を明示し、生成 AI 前提の働き方への全面移行を主導していた。

# 株式会社LayerX

## 1 企業紹介

LayerXは、「すべての経済活動を、デジタル化する。」をミッションに掲げ、SaaS、Fintech、AI・LLMの3つの柱で事業を展開する企業である。同社のAi Workforce事業部では、企業の文書処理や業務フローをLLMで効率化するソリューションの開発および導入支援を担っている。

## 2 サービス紹介

「Ai Workforce」は、LLMやコード実行モジュールを組み合わせる独自のAIワークフローを構築し、契約書や請求書などの複雑な文書処理を自動化するプラットフォームである。また、支出管理サービス「バクラク」においては、多様な形式の請求書を読み取る高度なOCR機能を提供している。

## 3 具体的課題

LLMの特性上、精度100%を保証することは技術的に困難であり、特に大企業の基幹業務においてAIの出力をいかに信頼可能なものにするかが最大の課題であった。

- フォーマットの多様性: 会社ごとに異なる請求書レイアウトや、Excel内の変則的なメモ、特殊記号（点線や丸囲み数字）などがLLMの誤認識を誘発する恐れがあった。
- 正解データ（Ground Truth）の欠如: 導入先の現場において、AIが目指すべき望ましい入出力が明文化されていないことが多かった。また、その出力を得るためのワークフローの段階的な検証には望ましい中間生成物との比較が必要だが、これらも多くの場合残っていなかった。

## 4 具体的対策

技術的な精度向上のみに依存せず、人間が介在するプロセス設計（Human-in-the-loop）や、導入初期のトライアルを通じたリスク管理を徹底していた。

- Human-in-the-loopの徹底（3.2）: AIを完結した自動化ツールではなく「人間のレビューを支援する存在」と位置づけ、UI上に抽出元の参照機能や修正保存ボタンを実装し、必要に応じて最終判断を人間が行う設計としていた。
- プログラムと生成AIの使い分け（3.6）: タスクを実行するためのワークフローの構築では工程を細分化し、ルールベースで処理可能な部分はプログラムで行い、非定型な解釈が必要な部分には生成AIを利用していった。このように、何でも生成AIに任せるのではなく、必要に応じて生成AIを適用することで動作の安定性を確保していた。

- 段階的な対象拡張と期待値調整（4.3）：1～3 か月程度のトライアル期間を設け、実際の運用ログから読み取れないパターンを抽出。精度100%ではないことを前提とした「業務削減効果」をKPIとし、顧客の期待値をコントロールしていた。
- 専門家との高速フィードバックループ（4.2）：業務のプロである顧客に対し、開発中ワークフローの生成物を提示することでエッジケースの存在が指摘される場合がある。このフィードバックループをできるだけ早く回転させることを開発では意識していた。また、ワークフローの動作の解釈性を高めるため最終生成物だけではなく思考過程も同時に顧客に提示するといった試みも行っていた。
- 多重的な品質保証（3.3）：LLMの出力結果に対し、「各項目の合計値が計算と一致するか」といったチェックを人手で行うことができるような機能を搭載することで、品質保証を多重的に行う仕組みを構築していた。

# Appendix 2. 外国人支援を想定したチャットボットの開発記録

## 1 はじめに

本章では、本書「生成 AI 実装ガイド」において体系化した GenAIOps フレームワークの各プラクティスを AI チャットボットを実際に開発することで検証した記録を述べる。

本書では技術・プロセス・組織を対象とした広範な取り組みを扱っているが、そのうち技術とプロセスを対象として検証した。とくに、3.6. 節の技術選定や、4.1. 節 AI エージェントの開発フェーズで述べた内容を実際に検証し、不確実性の高い AI 開発の初期の取り組みを実施に推進する方法を確認した。

以降の節では、初期の価値定義から、技術選定の葛藤、プロトタイピングを通じた設計の簡素化、および運用監視による継続的改善に至るまでのプロセスを時系列に沿って記述する。

## 2 開発方針と技術選定の観点

技術選定の観点について述べる。AI チャットボットを開発した目的は次の 2 点に整理できた。

1. ヒアリング調査で得られた技術的なプラクティスを実際に実装・検証すること
2. 機能的な AI を作り上げるプロセスを体験して理解を深め、本書の品質をより向上させること

技術選定では前者の目的のため、単に最新の技術スタックを選ぶのではなく、テストや検証のために幅広いベストプラクティスを実装できる柔軟な構成を選ぶことを重要視した。

### 2.1 主要な検討事項と技術スタックのスペクトラム

利用可能な複数の選択肢を調査した結果、主要な決定事項は以下の 2 点であった。

- フレームワーク: エージェントの挙動をどのようにコード化するか、またどのような機能が標準で提供されているかに影響する。
- インフラストラクチャー: アプリケーションをどのようにデプロイし、実行するかわる。

フレームワークは次の 2 つを主な検討対象とした。

- LangGraph: 明示的なグラフベースのワークフローで動作。制御性と透明性に優れる。
- ADK: LLM による暗黙的なルーティングで動作。実装は容易だが思考プロセスの検証に課題がある。

インフラストラクチャーでは次の 2 つを主な検討対象とした。

- Google Cloud: Vertex AI 製品スイートや Cloud Run などへのアクセスが可能。
- LangChain Cloud: LangGraph Server や LangSmith へのアクセスが可能。

技術選定にあたっては、ビジュアルプログラミング (Dify) から Python の `requests` ライブラリを直接使う方法まで、無数の選択肢が候補となる。その中で、実際に採用される可能性があり、かつ開発とテストのバランスが良い「中間的な選択肢」をとることを優先した。たとえば PydanticAI のような選択肢は技術的に非常に興味深いものの、実案件での採用可読性を考慮し、今回は採用を見送ることとした。

結果として、以下の 4 つの組み合わせを比較検討した。

1. LangGraph + LangChain Cloud
2. ADK + Vertex AI Agent Builder
3. LangGraph + Cloud Run
4. ADK + Cloud Run

## 2.2 選定の判断基準

前述の通り、構成要素をどの程度低レベルで制御したいかという選択肢の幅が存在する。LangChain Cloud や Vertex AI Agent Builder は煩雑な部分を抽象化してくれる利点があるが、一方でカスタマイズ性は制限される。今回のプロジェクトでは、独自の UI フロントエンドを構築して UX 改善を反復するという技術的要件があったため、UI が固定されるサードパーティ製ソリューションは選択肢から除外した。

また、チームがすでに Google Cloud に関する豊富な経験を持ち、基本的な設定が完了していたという点はインフラ選定における大きな決定要因となった。これは、企業がすでに確立され慣れ親しんだインフラを使い続けるという現実的な判断基準に合致する。

Google Cloud を利用している組織が、LLM プロジェクトのためだけに他のクラウドへ切り替えるには相応の強力な理由が必要になるが、今回の検証においても既存資産を活用して安全機能や UI/UX への投資を優先すべきという、第 3.6 節の議論を裏付ける結果となった。

これらの検討を経て、エージェントアプリケーションに特化した中間レベルの抽象化を提供する LangGraph と、完全に制御可能なデプロイ・フロントエンド環境としての Cloud Run の組み合わせを採用する方針を固めた。

### 3 ステップ 1：開発前の非技術的議論

技術的な実装を開始する前に、まずアプリケーションのビジネスニーズと、提供可能な価値の範囲を定義した。生成 AI を用いたアプリケーションは、生成 AI の柔軟性からスコープが過剰に大きくなりがちであるため、方向性を定めるための非技術的な議論を重視した。

スコープが過剰に大きくなりがちな例として、たとえば「多言語対応」があげられる。チャットボットアプリケーションは、特別な設定なしにデフォルトであらゆる言語をサポートできるため、一見すると「無料の機能」のように感じられ、初期のスコープに含めてしまいやすい。実際に、今回の AI チャットボットでも初期は多言語対応をスコープに含めていた。

しかし、エンジニアリングの観点からは、以下のような制約に直面することが議論の中で特定された。

- モデルの学習データセットの不均衡に伴う、言語ごとの性能差のリスク。
- 正規表現や厳密なスキーマに依存する既存ロジックが、多言語入力によって損なわれる可能性。
- ネイティブ不在下でのトーン（敬語等）や、専門用語（例：「漢方」）の翻訳一貫性を検証・強制する困難さ。

「言語をどう正確に検出するか」「ユーザーが途中で切り替えたらどうするか」といった具体的な技術課題を検討した結果、安易に全言語をサポートするのではなく、ビジネスニーズと技術的制約の現実的なバランスをとる決断が求められる。

生成 AI の柔軟性に流されず、スコープを調整して提供価値に集中するために、何度か「アイデアソン」を実施した。開発チーム内でのアイデアソンを通じて、どのような AI エージェントを開発すべきかの優先順位付けを行った。その際、以下の多角的な評価軸を設定し、各アイデアの妥当性を検討した。

- 市場性とユーザーの受容性
  - 対象ユーザーの多さ、およびその属性が明確であるか。
  - 協力的なユーザーが得られる見込みがあり、実用的なフィードバックが得られそうか。
  - すでに先行事例が存在し、ベンチマークが可能か。
- インパクトと面白さ

- プロジェクト自体の面白さ（チームのモチベーション）。
- 見込みの効果の大きさ、および想定ユーザーの生活に対する具体的なインパクトの強さ。
- ドメインの特性と専門性
  - 解決すべき課題（ドメイン）が明確に定義されているか。
  - そのドメインにおいて、どの程度の専門知識が要求されるか。
- 実現可能性と将来的な展開
  - 技術的な実現可能性は十分か。
  - 開発した仕組みが、他の組織やユースケースへ応用可能か。
  - 自社内にとどまらず、社外へも提供可能な価値か。

なお、今回の検証においては「開発に要する見込みの工数」はあえて評価軸から除外した。これは、スコープを小さく調整し、小規模な取り組みと学習を繰り返すという汎用的な課題解決プロセスを通じて GenAIOps の本質的な検証を行うことを優先したためである。

また、既存の ChatGPT や Google NotebookLM といった汎用ツールとの差別化も検討した。支援チャットボットの実践演習の一環として、対面でのアイデアソンを何度も繰り返し、評価軸に含める要素の優先順位付けと投票を重ねて方向性を合わせた。

議論の中で何度も立ち戻った重大な問いの 1 つは、こうした既存ツールを差し置いて独自の専用製品を管理する意義は何か、というものであった。しかし、パレートの法則が示す通り、実際に開発努力のほとんどを占めるのは、残りの 20% (堅牢性、安全性、信頼性、独自の UI) であった。試行錯誤のプロセスを通じてアプリケーションの要件を確認し、既存のツールでは充足できないことを確認した。

## 4 ステップ 2：技術的議論と初期設計

主要なアイデア、ターゲットユーザー、ビジネス目標、および価値が確立されると、一連の技術的議論が行われた。目標は、想定されるすべての要件をカバーし、かつ少人数のチームで現実的な段階を踏んで実装可能なアーキテクチャを構築することであった。

非常に初期のデータアーキテクチャ・フローのドラフトの 1 つは、以下の通りであった。

1. (技術レイヤー 0：レート制限、悪用防止など)
2. アドバーサリアル入力検知 [ノード]
  - プロンプトインジェクション・フィルター、脱獄 (Jailbreak) 検知、フラグが立てられた入力の監査ログ
3. 言語チェック [ノード]

- 入力が日本語/英語以外の場合、それらを使うよう促す（ベンチマーク・データセットの範囲を限定するため強制）。
  - あるいは、最後に翻訳エージェントを使うか尋ねるが、出力精度が下がる可能性があることを伝える。
4. 範囲外（Out-of-scope）トピックチェック・エージェント [ノード]
- 質問が定義された回答可能カテゴリ（ビザ、住宅、税金など）内にあるか確認。そうでなければ丁寧に断る。
  - コンテンツ・セーフティも含む。例：カテゴリ内であっても「偽造ビザの作り方」などは拒否する。
5. インテーク（Intake）エージェント
- ユーザーの入力に十分な情報（日付、場所、関係者、緊急性などの事実）が含まれているか確認。不足していればユーザーに促す。
  - 選択肢が複数あり、値を推測できる場合はインタラクティブにする。例：「どの都市の問題ですか？」と聞く代わりに、ボタン（東京、大阪、その他）を表示する。
  - パスポート番号や個人情報（PII）の共有に対するプライバシー・ガード。
  - 複数ターンの対話とメモリを保持するためエージェントとして構成。
6. RAG（検索拡張生成）風エージェント
- 精査されたソースからのみ取得。
  - 外部検索と結果の統合を行うためエージェントとして構成。
7. ハルシネーション（幻覚）の整合性チェックと引用 [ノード]
- インライン引用を確認し、出力が取得したテキストに基づいているかチェック。
  - 信頼度が低い場合はフォールバック（代替処理）を行う。
  - 信頼度を色で表示するUIは、透明性の観点から有用。
8. 要約エージェント [ノード]
- 標準化されたセクション・テンプレート：
    - 免責事項
    - 次のステップ
    - 役立つ機関（連絡先付き）
    - 役立つ単語/フレーズ
  - 単なるテキストではなく、カード形式で出力するUIの工夫。
9. 法的助言検知器 [ノード]
- 出力が「～すべきです」などと言っていないか確認。無資格の法律実務を避ける。
10. バイアス/公平性検知器
11. 出力
- ここから、ユーザーは以下のサブワークフローをトリガーするボタンをクリックできる：
    - 「もっと簡単に言い換えて」
    - 「関連する役所へのメール下書き作成」

- 「市役所に電話する必要があるため、読み上げ用のハイライトされた漢字+ふりがな付きスクリプトを表示」
- 「県内の無料法律相談サービスを探す」 紹介ボタン

## 12. ユーザーフィードバック収集ノード/エージェント？

- 透明性のために推論の連鎖を公開。例：「入力のアドバーサリアル・チェック合格  / 検出言語：英語  / 使用された RAG ソース：[引用] / 信頼度：0.82 → 要約プログラム起動」

もちろん、標準ではないものの、確立されインターネットで広く共有されているアプリケーション・アーキテクチャはすでに存在する。[Building Effective AI Agents by Anthropic](#) は、エージェント・アプリケーションにおけるノードの整理方法について優れた出発点となった。その後、業界の最近の動向や経験についても調査した。

- [法令 Deep Research ツール Lawsy を OSS として公開しました | Tatsuya Shirakawa](#)
- [PharmaX: AIエージェントの継続的改善のためオブザーバビリティ](#)
- [LayerX: 現場で動くAIワークフロー～チューニングを効率化する工夫～](#)
- [AIエージェントの地上戦～開発計画と運用実践 / 2025/04/08 Findy ランチセッション #19](#)

この最初のドラフトは、システムのすべての可動部分を実装するために必要な労力を著しく過小評価していた。個々の「ノード」自体は単純で、単独でのテストも容易だが、ノード間の相互作用が雪だるま式に問題を肥大化させる。主な問題については、以降の節で扱う。

## 5 ステップ 3：プロトタイピングと実地検証

もっとも労力を要するステップは、ビジョンをすり合わせ、予期せぬ問題や機会を発見するために、製品のいくつかの反復バージョン（イテレーション）を実装することである。今回のケースでは、いくつかの「波（Wave）」に分かれて進行した。

### 5.1 段階的な開発の波（Wave 1～3）

#### Wave 1.

- 利用可能な異なるフレームワーク（LangGraph、ADKなど）と、それぞれのサンプル、ガイド、またはヘルパーツールを試用した。デプロイは行わず、ローカルでの開発体験を評価した。
- 「ルック&フィール（見た目と使い心地）」を比較し、製品のフルバージョンをサポートするのに十分な機能（リトライ・ロジック、ヒューマン・イン・ザ・ループ、モニタリング、プロンプト管理など）があるかを確認した。

- サンプルデータやダミーデータを使用し、端末（ターミナル）へのテキスト出力に限定した。
- スコープを概念実証（PoC）に近い状態に保ち、ノードを以下に限定した：
  1. アドバーサリアル入力検知
  2. メモリ機能付きインテーク・エージェント
  3. ベクトルDBから読み取る RAG 風エージェント
  4. Google 検索ツールを使用して結果を統合するエージェント

Wave 1 の調査中に、利用可能なテクノロジーの最新機能についてより深い理解が得られた。それでも、ターミナルベースの「バイブ・コーディング（Vibe-coding）」による概念実証を超えて少し拡張してみなければ、今回のユースケースにどのスタックが最適かを予測するのは困難であった。そのため、継続的なハッカソンを通じて、より洗練された製品の作成を続けた。

## Wave 2.

- デプロイと本番化のオプション（Agent Engine、Cloud Run）を比較した。
  - 調査の一環として、アプリケーションの DAG（有向非巡回グラフ）のデプロイと管理を LangGraph Server のようなサードパーティ・サービスに委託するのは非常に役立つと考えたが、いくつかの理由から適さないと判断した。理由は以下の通りである。
    1. カスタマイズ性の低下
    2. 後でアプリケーションをオープンソース化することが不可能
    3. エコシステムのロックイン
    4. 金銭的投資の必要性
- バックエンドからフロントエンドへのモデル応答のストリーミング表示などを試すため、ユーザー向けフロントエンドを追加した。たとえば FastAPI ベースのアプローチによって、React フロントエンドなどの追加が容易であることを確認した。
- 利用可能な「ネイティブ」およびサードパーティのモニタリング/デバッグツールを検討した。

## Wave 3.

技術的知識が共有されるにつれ、検討されたフレームワーク、ツール、デプロイ・オプションのそれぞれの限界が明確になり、適切なバランスを持つ単一の統合された開発方向を選択することができた。選択の詳細は、第 2 節（技術選定の観点）で述べたとおりである。

## 5.2 ステップ 3b：データの収集と実証

どのようなユーザー対話を期待しているかが明確になったため、検証のために実世界のデータを収集し始めた。ターゲットユーザーへのインタビューを通じ、日本での生活に不安を抱える外国人住民が直面する具体的なニーズとして以下の点が明らかになった。

- 電話対応の心理的障壁
- 書類の仕分けのストレス
- 手続きの依存関係の解消
- 一次情報へのアクセス

「電話対応の心理的障壁」では、不慣れな日本語での問い合わせに対する恐怖があり、電話前に「伝えるべきこと」「聞かれそうなこと」を整理する機能への強い期待が示された。

「書類の仕分けのストレス」では、役所や銀行から届く日本語の郵便物の山から、重要書類（保存すべきもの）と不要なものを判別できない不安が大きいことが確認できた。

「手続きの依存関係の解消」では、住所・銀行口座・電話番号など、特定の順序で進める必要がある日本独自の生活基盤構築において、全体をナビゲートする仕組みが求められていた。

「一次情報へのアクセス」では、公的機関や銀行のサイトが複雑すぎて正しい情報に辿り着けないため、これらを情報源とする信頼性の高い回答へのニーズが確認できた。

これらのインサイトに基づき、AI エージェントが提供すべき機能を再定義した。具体的には、Deep Research による「正しい手順」の提示、問い合わせ先の特定と対話用スクリプト（電話の事前準備）の生成、書類の重要度判定とネクストアクションの指示、そして公的機関のドメインに限定した RAG による一次情報へのアクセスの提供などを定義した。

このように収集した実際の問い合わせ例を用いて初期モデルの精度・挙動検証を行うことで、実際のユーザー課題に即したフィードバックループを回すことが可能となった。

また、上記のターゲットユーザーへのインタビューではプロトタイプとして小規模なデータセットを用いてデモを行ったが、本番運用を見据えて包括的なデータ収集に取り組んだ。データ収集では、ターゲットユーザーへのインタビューに基づき収集対象を定義し、各自治体のウェブサイトから生活に関連する情報の収集を行った。ここでも「一括収集の後に検証する」のではなく、収集したデータを逐次確認して優先順位を調整するなど、小規模な試行と検証・学習を繰り返す方針を貫いた。

また、検証の継続性を担保するために、自治体サイトのリンク切れ（リンク・ロット）への対策が不可欠であった。そのため、データを動的に参照するのではなく、静的なスナップショット

トとして保存・ホストする形態を採用した。この手法では提供情報が収集時点のものに固定されるため、最新ではない情報に基づいて誤った回答をしてしまうリスクが生じる。これへの配慮として、ユーザーインターフェイス上に最新情報を公式ソースで確認するよう促す注意喚起を含めた。

一般に、生成 AI を用いたシステムにおいては古い情報に基づく誤回答のリスクを考慮して、データの継続的な更新運用や、ユーザーへ最新情報を確認するよう促す適切なメッセージの提示といった配慮が重要となる。

## 6 ステップ 4：適合性を高めるための簡素化と技術選定の振り返り

最初の数回の技術的実装を通じて、適切な境界を持つ堅牢な設計の重要性を繰り返し強調する技術的問題に直面した。最終的に、アプリケーションのフローにおけるノード数を削減した。その主な理由は、ノードを増やすことが利益よりも多くの問題をもたらしたためである。検証を経て得られた設計上の教訓は、「**連鎖するデータフローは短く保つこと**」であった。

この簡素化、並列化の促進、および全体的な独立性の向上への変更により、開発とデバッグがはるかに容易になった。これは、初期段階で適切な技術的アプローチを選択していたために可能であった（フロントエンドとバックエンドの両方に十分な柔軟性があった）。認識したエージェント構造に関連する問題の例を以下に示す。

元の計画	何がうまくいかなかったか	どのように修正したか
ユーザーのクエリで不足していると思われる情報を動的に聞き返すインターク・エージェント	クエリ間で一貫性が低すぎ、基本的な事実を聞き忘れることがあった。	LLM への過度の依存という典型的な間違い。代わりに、明確に定義された選択肢を持つシンプルなチャット前静的フォームで、同じ結果を一貫して達成した。フィードバック・ノードについても同様で、単純な Good/Bad ボタンに置き換え、対話を二次的なダイアログで汚すことなく会話にメタデータを付加するようにした。
すべてのエージェントの前に範囲内 (In-scope) チェックを実行し、ユーザーが許可されたトピック内に留まるようにする。	元のプロンプトではモデルにスコア付きの人間が読める説明を求めていたが、トークンを大量に消費し、ときどき壊れることがあった。	ノードの出力をバイナリ (Yes/No) に切り替え、一貫性と信頼性を大幅に向上させた。
法的助言検知器、バイアス/公平性検知器などの過剰なノード。	スコープが大きすぎ、これらのノードのテストに多大な時間と労力がかかっていた。また、明らかな利点もなかった (出力を止めることはほとんどなかった)。	これらのノードがカバーしようとしていたのと同じ原則を他のシステムプロンプトに統合し、機能を損なうことなく解決した。
メインのシステムプロンプトで、関連する次のステップ、場所、単語も特定し、出力に含めるよう求めていた。	1つのモデル・クエリに対する指示が多すぎた。すべてのサブ項目の品質が低下し、出力が長すぎて快適に読めなくなった。	2つの軽量モデル (Gemini Pro の代わりに Gemini Flash) に、メイン・クエリと並行して単語と場所をそれぞれ探させ、別の UI カードに出力させるようにした。追加ツール (ライブ Google マップ検索など) により、出力の有用性が劇的に向上した。

最終的なアプリケーションのユーザーインターフェースは次のようになった。

**JP Japan Helpdesk**  
AI-powered assistant for official procedures

**Before we start**

We'll use your visa type and location to provide more relevant information and narrow down search results to what applies to your specific situation.

**Conversation Mode**

Single Turn

Ask one question and get one answer

Multi Turn

Have a back-and-forth conversation

**Your Visa Type**

Student  
留学

Engineer/Specialist  
技術・人文知識・国際業務

Skilled Labor  
技能

Spouse of Japanese National  
日本人の配偶者等

Dependent  
家族滞在

Other

Designated Activities (特定活動) ▼

**Your Location in Japan**

Tokyo

Yokohama

Osaka

Kyoto

Fukuoka

Other

Start Chat

**Collected Facts**

No facts collected yet.

Information about your situation will appear here.

**Key Terms**

No terms extracted yet.

Essential Japanese vocabulary will appear here.

**Useful Places**

No places found yet.

Government offices will appear here.

**JP Japan Helpdesk**  
Visas, housing, and everyday life

health insurance card (健康保険証). By showing your insurance card at medical institutions, you only need to pay a portion of the medical expenses. The amount of partial medical fees differs depending on age and income.

▼ 1 Source

[1] Microsoft Word - 01\_健康保険証の申請と手続きの概要.pdf  
健康保険証の申請と手続きの概要.pdf

Pages: 1, 3, 4

So if I'm here for recreational purposes and under a year?

If you are in Japan with a "Designated Activities" visa for recreational purposes and staying for less than one year, you are *not* required to enroll in National Health Insurance (国民健康保険, *Kokuhō*).

All people in Japan are generally required to enroll in some form of health insurance. However, there are exceptions.

You are exempt from enrolling in National Health Insurance if you have a "Designated Activities" visa (特定活動) for tourism, recreation, or similar purposes, and your period of stay is no more than one year.

> 1 Source

Type your question... Include any relevant details.

Press Enter to send, Shift+Enter for newline

**Collected Facts**

Visa Type  
Designated Activities (特定活動)

Location  
Yokohama

Health Insurance Status  
Not currently enrolled in National Health Insurance

Purpose of Stay  
Recreational

Duration of Stay  
Under one year

**Key Terms**

国民健康保険  
*Kokumin Kenkō Hoken*  
National Health Insurance

特定活動  
*Tokutei Katsudō*  
Designated Activities (visa)

在留期間  
*Zairyū Kikan*  
Period of Stay

免除  
*Menjo*  
Exemption

156

とりわけ、その他の主要な技術的問題は以下の通りであることが判明した。

- ユーザーとの対話を通じたメモリの維持
- RAG の管理
- LLM の出力のバリデーション

それぞれについて以下で述べる。

## ユーザーとの対話を通じたメモリの維持

多くのライブラリが LLM コンテキストを処理するさまざまな方法を提供している。しかし、依然として、マルチターンな対話を行う場合、関連情報を保持し、システムプロンプトに注入するための手動のデータ構造が必要であった。

たとえば、各ターンでユーザーの位置情報のコンテキストを処理する必要があった。ユーザーが「東京にいる」と申告した後、さらに「渋谷区ではなく港区だ」と明確にする場合への対応などがこれに該当する。この課題はマルチターンの会話においてとくに考慮が必要となる。ユーザーが確認の質問をする場合など、同じ LLM ノードとエージェントの DAG 構造を同じ会話内で数回通過する可能性があった。

この課題を回避するため、マルチターンではなくシングルターンの対話を強制することも考えられるが、チャットボットとしては不自然に感じられる懸念があった。Google が提供する Vertex AI API が RAG データベースへのクエリ時にコンテキストを管理する内部サポートを持っていたため、メモリの維持のために活用できた。

## RAG の管理

[収集、解析、トークン化](#)およびデバッグを行う RAG アプリケーションは、サードパーティの API サービスにプロンプトを送るだけの場合と比較して、当然ながら参入障壁はるかに高かった。その後の実験プロセスを通じて、なぜこの分野に多くの新しいスタートアップや製品が登場し、必要な労力の多くをアウトソーシングするサービスを提供しているのかが明らかになった。

開発したアプリケーションでは、ユーザーの入力に対する回答は、検索によって発見したドキュメントに基づく要約であり、引用が付されていることが必要となった。内製で手動によるパイプライン管理を行うことは技術的には可能なものの、開発リソースのほぼすべてを消費してしまい、UI、モニタリング、または安全性のための開発に回せるリソースがほとんど残らなくなることが予見された。

## LLM の出力のバリデーション

LLM の出力のバリデーションは、ここでは期待されるデータ構造や型と一致することの確認を指す。

生成 AI モデルは、一般に「気まぐれ (flaky)」と表現できる。出力が時折どのような形で誤るかを正確に予測することは困難であった。たとえば、システムプロンプトで JSON 出力形式を強制していても、API のサイレントな内部レート制限エラーのために、最後の閉じ括弧 (}) が出力されないケースが観察された。

これはもっともよく知られた問題の1つであり、多くの関連ライブラリやフレームワークのセールスポイントは、関連する苦労を最小限に抑えることに置かれている。それでも、UI でのエラー管理、全ステップのストレスチェック、リトライメカニズムの実装には、驚くほどの労力がかかることが実証された。

### 6.1 技術選定の振り返り

今回の開発における技術選定について振り返る。今回は結果として LangGraph + Cloud Run を選定した。検討した組み合わせについて再掲する。

1. LangGraph + LangChain Cloud
2. ADK + Vertex AI Agent Builder
3. LangGraph + Cloud Run
4. ADK + Cloud Run

前述の通り、構成要素をどの程度低レベルで制御したいかという選択肢の幅が存在した。LangChain Cloud や Vertex AI Agent Builder は、アプリケーション構築における煩雑な部分の多くを抽象化してくれるが、一方でカスタマイズ性は制限される。技術的要件の1つに、「アプリケーションの設計を改善するためにカスタム UI を持つこと」があった。これにより、独自の UI を強制するサードパーティ製のソリューションは選択肢から除外された。

もう1つの考慮事項は、チームがすでに Google Cloud に関する豊富な経験を持ち、プロジェクトや課金設定などが完了していたため、開発を大幅にスピードアップできるという点であった。これは意外にも大きな決定要因となったが、企業がすでに確立され慣れ親しんだインフラを使い続ける可能性が高いことを考慮すると、非常に妥当な判断であったと言える。Google Cloud を利用している組織が、LLM プロジェクトのためだけに他のクラウドへ切り替えるには、相応の強力な理由が必要になるからである。

テストのためのさまざまなモニタリングライブラリを組み込みたいという要望も合わせ、結果的に「オプション 3：エージェントアプリケーションに特化した中間レベルの抽象化を提供す

る定評のあるフレームワーク（LangGraph）と、完全に制御可能なデプロイ・フロントエンド環境（Cloud Run）」の組み合わせが採用された。

事後分析の結果、この技術スタックの選択は今回のタスクにおいて概ね適切であったと思われる。RAG（検索拡張生成）部分の精度維持は予想以上の難しさであった。手動で実装することも理論上は可能であったが、予想外の難易度の課題に直面したため、GCP が提供するセミマネージドな RAG サービスである Vertex AI Search へのピボットが必要となった。幸い、LangGraph ベースのアプリケーションはこのような変更を受け入れる余地が十分にあり、切り替えは比較的スムーズに行われた。この経験により、製品の将来的な成長方向を見据えた柔軟なアーキテクチャを選択することの重要性が実証された。

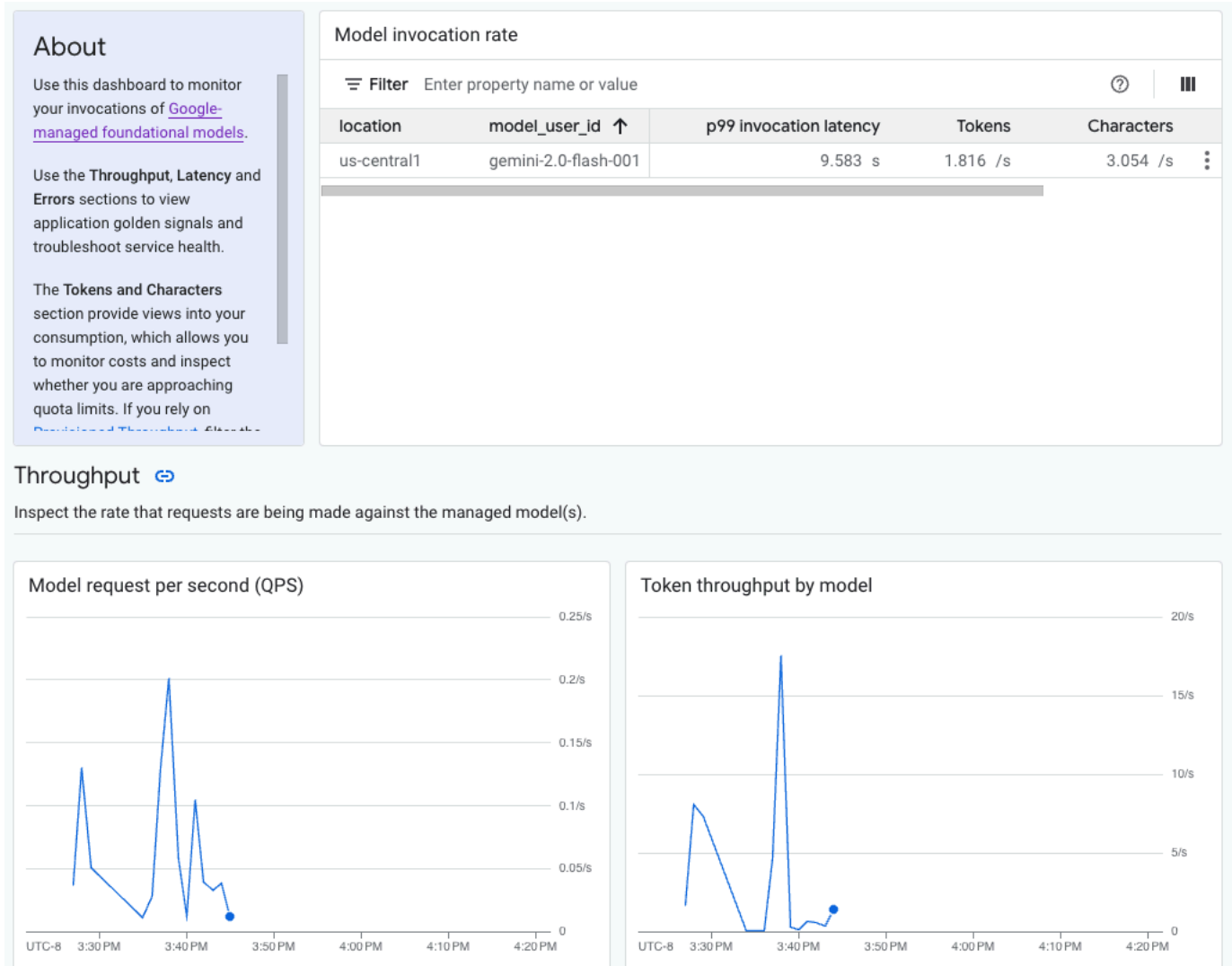
## 7 ステップ 5：デプロイ、テスト、および運用監視

関連事項：収集したモニタリングデータを活用した具体的な改善プロセスや、ビジネス上の成果測定については、第4章「継続的な改善」にて触れる。

明確なオブザーバビリティ（可観測性）は、MLOps 的な思想の主要な柱の1つである。問題を効果的にデバッグしたり、さらに言えば「早期警告システム」に基づいて問題の発生を未然に防いだりするためには、質の高い履歴ログが利用可能であることが不可欠である。

デプロイの観点から見ると、LLM アプリケーションと非 LLM アプリケーションはおよそ同じように見える。本ケースでも、Cloud Run にデプロイされた Docker コンテナは内部で通常の Linux システムを実行しており、ネットワークやディスクの使用量、レイテンシ（遅延）など、通常のメトリクスをすべて生成する。

内部で生成される LLM 特有のメトリクスやスコアもいくつかあるが、それらは最新の SaaS モニタリング製品に付随的なメタデータとして簡単に追加できるため、トークン・スループットのようなものを大きな労力なしに追跡できる可能性がある。



しかし、真の価値は、イベントの順序を繋ぎ合わせて非決定的なモデルのデバッグを可能にする、より深い「定性的」なロギングから得られる。

- なぜトークン・スループットが急増したのか？
- 特定のユーザー、特定のトピック、あるいは特定のモデルに原因があるのか？
- 会話を追跡して、その瞬間にモデルがどのように動作したかを再現できるか？

上流での小さな変更 (たとえば、日本の GCP リージョンでモデルが突然非推奨になるなど。これは実際に以前起こった) が、システム全体を瞬時にダウンさせる可能性があるため、リアルタイムのモニタリングは絶対不可欠であると認識した。そのため、本番レベルのシステム構築は、初期のハッカソンにおける優先事項の1つであった。

市場にはいくつかの製品があり、特定のニーズに対する有効性をテストし限界を見極めるため、さまざまなものを試した。それらのすべてが、動作させるためにアプリケーションコードの変更を必要としたが、関与の度合いには幅がある。製品の多くは次の2つに分類できた。

1. ホワイトボックス型
2. ブラックボックス型

ホワイトボックス型では、コード内のすべてのステップにカスタムの低レベル・リスナーを追加する必要がある。詳細の解像度が高くカスタマイズ性も優れているが、導入コストが高く製品へのロックインが生じる。

ブラックボックス型では、ツールは入力と出力のみを確認し、途中のステップは見ない。セットアップが迅速で、あらゆるアーキテクチャと完全な互換性があるが、得られる知見は限定的である。

また、アクティブなファイアウォールのように機能するものから、受動的なオブザーバーのように機能するものまで、さまざまな運用モードがある。いずれにせよ、すべてのソリューションは「LLM-as-a-Judge（評価者としてのLLM）」のアプローチで設計されており、オフライン（定義済みのデータポイントでのバッチ処理）とオンライン（データが到着するたびにリアルタイム計算）の両方で反復的なテストと検証が可能である。以下に、開発者向け UI の例をいくつか示す。

## Langfuse

モニタリング用の OSS で、フレームワークに依存しないものとしては Langfuse が挙げられる。

Langfuse は、オープンソースの LLM オブザーバビリティ・プラットフォームであり、トレース、評価、プロンプト管理などの機能を統合的に提供する。本プロジェクトでは、とくに複雑化した DAG 構造における各ノードの実行時間やコストの追跡、および特定の入力に対するモデルの応答精度を「LLM-as-a-judge」の手法で自動評価するために活用した。

The screenshot displays the Langfuse interface for tracing a LangGraph workflow. On the left, a sidebar lists navigation options such as Home, Dashboards, and Tracing. The main area is divided into a 'Tracing' table and a detailed view of a specific trace. The table lists various traces with columns for Timestamp, Name, and Input. The detailed view shows a flowchart of the workflow steps: start -> check\_scope -> check\_query\_scope -> RunnableSequence (containing ChatVertexAI and PydanticToolsParser) -> context\_drift -> end. The right panel shows the output JSON, which includes a human message and an AI response with detailed instructions and metadata.

## LangSmith

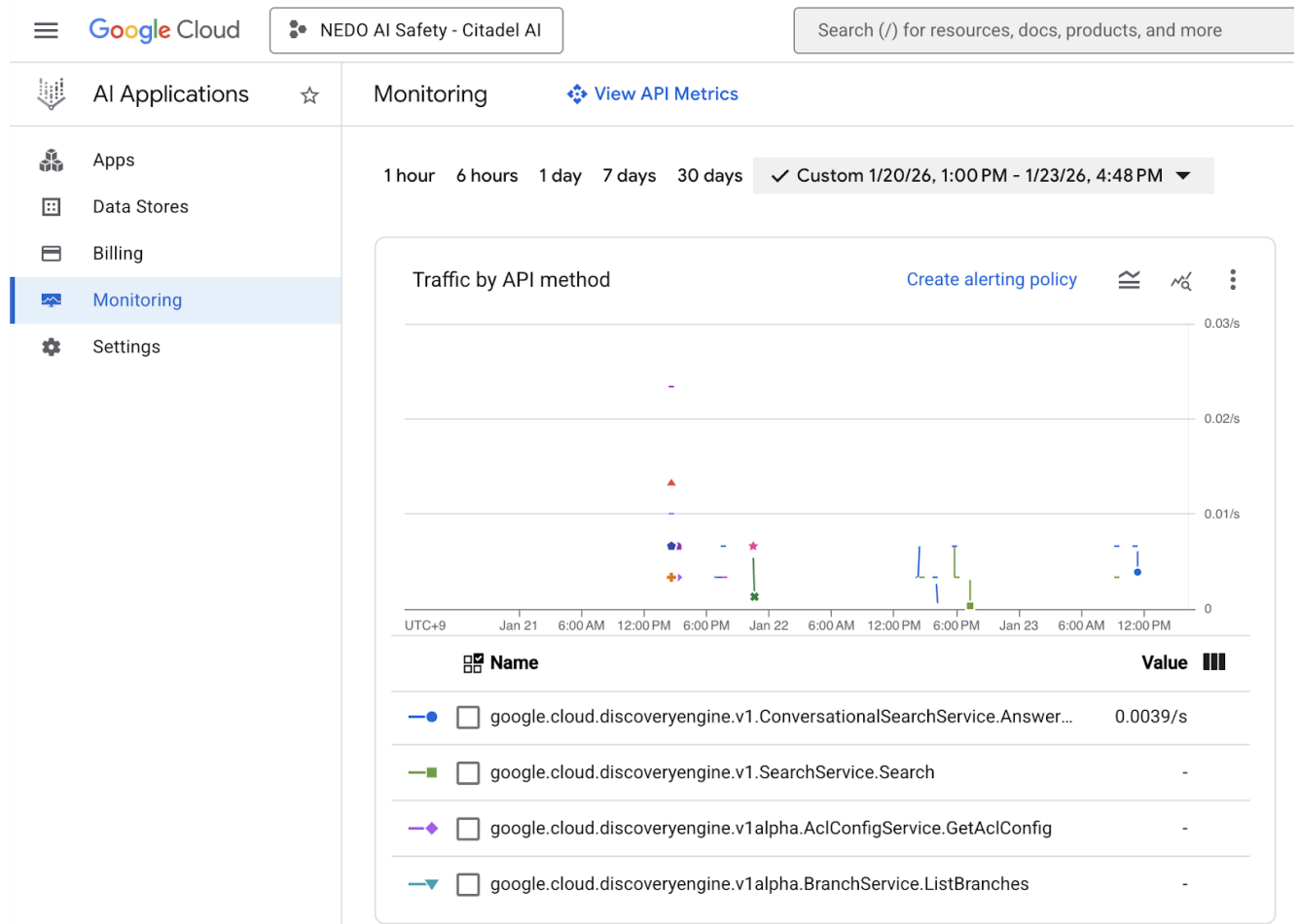
フレームワークに特化したモニタリング用のサービスとして LangSmith が挙げられる。

The screenshot displays the LangSmith interface for tracing a LangGraph workflow. On the left, a sidebar lists navigation options such as Home, Monitoring, and Datasets & Experiments. The main area is divided into a 'Tracing' waterfall chart and a detailed view of the workflow. The waterfall chart shows the execution time for various steps: check\_scope (2.75s), check\_query\_scope (2.75s), RunnableSequence (2.74s), ChatVertexAI (2.73s), PydanticToolsParser (0.01s), route\_search (0.00s), route\_to\_search\_method (0.00s), search\_answer (8.28s), vertex\_answer\_search (8.28s), route\_to\_info\_agents (0.00s), extract\_facts (6.68s), RunnableSequence (6.67s), ChatVertexAI (6.67s), and PydanticToolsParser (0.00s). The detailed view shows the input and output of the workflow, including a human message and an AI response with detailed instructions and metadata.

LangChain アプリケーション向けの「オールインワン」の提供の一部であり、他のフレームワークでは利用できない。この分野でもっとも初期に登場した製品の1つであり、前述の LangChain と同様、それ自体の成功ゆえの課題も抱えている。

## Vertex AI

クラウド環境に用意された、プロプライエタリなサービスとして Vertex AI の生成 AI に関するモニタリングを取り上げる。



Google Cloud インフラ上にデプロイされたモデルには高度なモニタリング機能が用意されているが、プロジェクトで使用していた Vertex AI Search に関しては、報告される情報はかなり限定的であった。当然ながら、GCP/AWS/Azure のようなエコシステムにロックインして標準で動作するツールを増やすか、より多くのセットアップを必要とする環境に依存しないアプローチをとるかという、選択とトレードオフが存在する。

## Citadel Lens

生成 AI 用のクラウド環境に非依存なサードパーティー製のサービスはいくつか登場してきている。ここでは Citadel Lens を取り上げる。

Citadel Lens はフレームワークに非依存なサービスであり、LLM-as-a-Judge を中核とした評価方法を提供する。フレームワークと強固に結びついていない自由度がある一方で、ユーザー自身が独自に設計・実装を行う必要がある点に留意が必要である。

The screenshot shows the 'Custom Metrics' configuration page in Citadel Lens. The page is titled 'カスタムメトリクスを作成' (Create Custom Metrics). It is divided into three main sections:

- メトリクス詳細 (Metric Details):**
  - メトリクス名 (Metric Name):** 返金ポリシー遵守度 (Refund Policy Compliance)
  - メトリクス入力 (Metric Input):**
    - 入力クエリ (Input Query): `{{ user_query }}`
    - 生成された出力 (Generated Output): `{{ gen_output }}`
    - ソース (Source): `{{ src }}`
    - 理想的な出力 (Ideal Output): `{{ ref_output }}`
  - 評価用プロンプト (Evaluation Prompt):**

```
[BEGIN DATA]
*****
[ユーザーからの入力]: {{ user_query }}
[生成された出力]: {{ gen_output }}
*****
[END DATA]

## 評価方法
カスタマーサポートチャットボットの回答が、当社のオンライン購入に関する返金
```
  - メトリクス出力 (Metric Output):** 1,0,5,0
  - 説明を出力しますか? (Output Description):**  はい  いいえ
  - ボタン:** カスタムメトリクスを追加 (Add Custom Metric)
- メトリクスを実行 (Execute Metrics):**
  - データセット (Dataset):** サンプル (Sample)
  - データの数 (Number of Data):** 5
  - 実行ボタン:** メトリクスを実行 (Execute Metrics)
  - 実行時間 (Execution Time):** 5.3s
  - 結果表示:**

メトリクス (Metric)	生成された出力 (Generated Output)	入力クエリ (Input Query)
返金ポリシー遵守度: 0.5	申し訳ありませんが、お手持ちのBluetoothスピーカーの音が途切れる問題について、返金リクエストを受け付けることが可能です。以下の情報を提供していただけませんか？ - 購入時に使用された支払い方法 - 購入時の配送先住所 これらの情報があれば、返金手続きを円滑に進めることができます。	購入したBluetoothスピーカーの音。返品可能でしょうか？
返金ポリシー遵守度: 1	斎藤様、お問い合わせいただき、誠にありがとうございます。返金リクエストに関する情報を確認いたしました。外付けSSD（注文番号: 0013）がどのポートでも認識されないとのこと、ご不便をおかけして申し訳ございません。ただし、返金リクエスト（リクエストID: R006）は現在「拒否」のステータスとなっております。返金ポリシーに基づき、返金は商品が破損している場合、不具合がある場合、または説 [...] <a href="#">全文を表示</a>	なぜSSDの返金が拒否されたんです？でも認識しないのに、おかしいじゃ
返金ポリシー遵守度: 0	山本様、迅速な対応ありがとうございます。ご連絡いただきましてありがとうございます。お手続きの進捗をお知らせいたします。2日前にいただいたゲームヘッドセットの返金リクエスト	2日前にゲーミングヘッドセットの返金

## 7.1 コード関連の問題の解決

モニタリングの明らかな価値の1つは、本番環境で修正可能なバグ関連の問題を捉えることにある。直面したそのような例の1つは、ガードレールの調整がアグレッシブ（過剰）すぎたことに関連している。実際の入力内容にかかわらず、短すぎるクエリがすべて「範囲外」としてマークされてしまったのである。

関連する GitHub Issue のスクリーンショットから、バグ解決の流れを確認できる。これは認めるべき比較的単純な問題であったが、このようなセットアップが、以下のようなトリッキーな技術的問題の検出にどう役立つかを理解するのは難しくない。

- チャットメッセージ間でコンテキストが失われる：モデルがユーザーとこれまで何をしてきたかを「忘れて」しまう。
- 応答しないモデル API に対するリトライメカニズムの誤作動。

- 無礼、あるいはブランドイメージに合わない出力トーン（LLM-as-a-judge による評価の助けを借りる）。

ただし、引用番号の不一致など一部の定量的な問題は、定義済みの自動メトリクスを使用して捉えることは不可能であり、デプロイ前のチェックとしてドッグフーディングや手動テストが依然として不可欠であることに注意が必要である。

# 「Out of scope」判定と低評価の増加（最新リリース後） #9

Closed #10



Arstapaton opened 2 days ago

直近のリリース (v1.4.0) 以降、[モニタリングダッシュボード](#)上でユーザーからの👎評価が目立って増えています。CC @taniokay

- 気になったのでログを少し掘ってみましたが、低評価がついている会話のほとんどが、モデル側で `status: out_of_scope` と判定されて終了しているケースでした。
- 単純にユーザーが利用規約やスコープを理解していないのか、それとも何か特定のワードでjailbreakなどを試そうとして abuse 判定に引っかかっているのか、現状だと区別がついていません。もし誤検知 (False Positive) だとUX的にまずいので、どなたかこのあたりのロジック詳しい方、調査お願いできませんか？
- タグやグループ핑をいじってみましたが、特定のユーザーやトピックが原因というわけではなさそうです (?)



Create sub-issue

Arstapaton self-assigned this 10 minutes ago

Arstapaton added Bug Backend 10 minutes ago



taniokay yesterday

トレース一通り見てみました。@Arstapatonさんの言う通り、低評価のタイムスタンプと `out_of_scope` イベントが完全に同期してますね。ただ、実際のプロンプトの中身を確認してみると、Abuse っぽい挙動は全く見当たりませんでした。具体的には "How to get National Insurance" みたいな、どう見てもホワイトな質問に対しても `out_of_scope` が返されてます。ドメイン知識的にも完全に許可されてるトピックのはずなので、なんでこれが弾かれるのか正直謎ですね...。ルールの設定か何かが悪影響を及ぼしている可能性がありそうです。もうちょっと深掘りしてガードレールの挙動追ってみます。

The screenshot shows a LangGraph trace for a session. The flowchart on the left shows the execution path: `__start__` -> `check_scope` -> `out_of_scope`. The output log on the right shows the model's response, which is a refusal to answer the question because it is outside the model's expertise.

```
Path      Value
messages [{"content": "...", "content": "..."}]
> 0       6 items
> 1       9 items
collected_facts 2 items
  Visa Type      "Student (留学)"
  Location       "Yokohama"
  conversation_mode "multi"
answer          "I'm specialized in helping with official procedures in Japan, such as:
- Visa and immigration procedures
- Residence registration
- Health insurance enrollment
- Employment procedures
- Housing and rental procedures
- Banking and financial procedures
- Document requirements and applications
Your question seems to be outside my area of expertise. Could you please ask about a specific procedure or requirement related to living in Japan?
For example, you could ask:
- 'How do I renew my work visa?'"
```

Arstapaton mentioned this

Resolve issue with the excessively aggressive out of scope length checker #10



taniokay 4 hours ago

原因特定できました、静的ガードレールの設定が強すぎたのが原因みたいです。どうやら短い質問文に対して、コンテキスト不足と判断して一律で弾くようなロジックが強めに効いていました。“How to get National Insurance”も単文で短いので、そこで引っかかったみたいです。修正方針としては以下の2点で対応します

- ガードレールの文字数判定の閾値を緩めて、短文でも一旦通すようにする
- Scope Checker の System Prompt を調整して、短い質問でも文脈から意図が汲み取れる場合は許容するように明記する

手で試した感じだとこれで誤検知が減るはず。今から修正PR出しますね。

[nedo-ai-safety-agent/backend/nodes/check\\_scope.py](#)

Lines 116 to 120 in 52838e1

```
116 # Quick validation: very short queries are usually too vague
117 word_count = len(query.split())
118 if word_count < MIN_QUERY_WORDS:
119     logger.info(f"⚠️ Query too short ({word_count} words) - requires more context")
120     return "out_of_scope"
```



github-project-automation moved this to Done in NEDO AI Safety Project 6 minutes ago

見落とされがちな興味深い情報源の1つは、ユーザーが生成したフィードバックである。表面的な定量的メトリクスが正常に見えても、ネガティブなフィードバックが増加している場合は、システムをチェックするシグナルになる。たとえば、プロジェクトではシンプルな Good/Bad ボタンのシステムを導入している。

Was this helpful?



Thanks for your feedback!

## 7.2 データ関連の問題の解決

前述の通り、手動の検証チェック、ドッグフーディング、または詳細なユーザーフィードバックの最中にのみ捉えることが可能な問題がある。そのような問題の1つが、あるテストユーザーによって提起された。彼は自身が詳しいトピックである「永住権の手続き」に関する回答の質の低さに驚いていた。ある意味で、これは「専門家」にモデルの出力を検証してもらう間接的な方法であった。しかし、実際のユーザーは専門家ではないことが想定されるため、このような問題は長い間気づかれない可能性がある。

What document should I prepare before fill an application for a permanent visa

I am sorry, but I cannot provide a definite list of documents you need to prepare for a permanent visa application. The required documents vary based on individual circumstances and the specific requirements of the Immigration Services Agency.

To determine the necessary documents, you should:

1. Consult the official website of the Immigration Services Agency of Japan (入管庁) for the most up-to-date information.
2. Contact the Yokohama Immigration Office directly for specific guidance based on your situation as a student (留学).

> 1 Source

出力を調査し、開発側で問題を再現したところ、これが一時的な不安定さによるものではないことが確認された。限定的な出力と引用から判明したのは、RAG データベースにそのトピックに関する関連ドキュメントが実際に含まれていなかったことであった。外務省の文書は開発中に収集されており、[彼らのウェブサイトには関連情報がある](#)にもかかわらず、それらが発見・パース（解析）されることに失敗していた。

The screenshot displays a LangGraph trace for a session on 2026-01-21 at 19:38:51.653. The session ID is thread-1768991459936-w0cop9w32, with an environment of 'default' and a latency of 11.80s. The total cost is \$0.00258, and the token count is 2,702 to 204 (total 3,938).

The workflow trace shows the following steps:

- LangGraph (11.80s)
  - Conciseness: 0.00
  - Hallucination: 0.00
  - Helpfulness: 0.00
  - Legality: 0.00
  - Toxicity: 0.00
- LangGraph (11.80s)
  - check\_scope (1.45s)
    - check\_query\_scope (1.44s)
      - RunnableSequence (1.44s)
        - ChatVertexAI
- search\_answer
- generate\_phrases
- \_end\_

The output is a JSON array of 9 items, containing 4 messages:

```

{ 9 Items
  messages: [ 4 Items
    0: { 7 Items
      content: "What document should I prepare before fill an application for a permanent visa"
      additional_kwargs: { 0 Items
      }
      response_metadata: { 0 Items
      }
      type: "human"
      name: null
      id: "882d3da9-7663-4b97-bc0f-f39fabaf6d0e"
      example: false
    }
    1: { 10 Items
      content: "To prepare for a Permanent Resident (永住者) visa application in Tokyo, you should gather the following documents.

      * **Application Form for Permanent Residence (永住許可申請書)**.
      * **Photograph** (40mm x 30mm).

      It's important to note that additional documents may be required depending on your individual circumstances.
      "
      additional_kwargs: { 0 Items
      }
      response_metadata: { 0 Items
      }
      type: "ai"
      name: null
    }
  ]
}

```

## 8 総括

本プロジェクトを通じた外国人支援を想定したチャットボットの開発は、生成 AI 開発における不確実性がいかに高く、それに対していかに適応的なプロセスが有効であるかを実証する試みであった。

### 8.1 品質と安全性を両立させる継続的改善

今回は、第 3 章で紹介した技術的な取り組みと、第 4 章で論じた開発プロセスにおける取り組みを検証した。

検証のステップ 5 で述べた専門家ユーザーによる指摘とそれに基づく RAG データの補強は、運用監視という「技術」と、フィードバックを即座に反映させる「プロセス」が連動した好例である。このように、開発と運用のサイクルを高速に回し続けること自体が、生成 AI を用いたプロダクトの品質と安全性を同時に高めることを、実地での経験を通じて確認できた。

## 8.2 不確実性への適応と学習のループ

AI エージェントの開発において、初期段階で「正解」の設計を定義し切ることが不可能に近い。本プロジェクトにおいても、多言語対応のスコープ調整、内製 RAG からのピボット、複雑な構成からの簡素化といった「試行錯誤」の連続であった。

しかし、これらの「右往左往」はムダな手戻りではなく、不確実性を段階的に解消し、より適切な設計へと導くための「学習」のプロセスであった。最初から完璧な答えを求めず、変化を許容する柔軟なアーキテクチャーを選択し、実データに基づいて修正を繰り返す GenAIOps 的なアプローチが実際に有効であったことを検証できた。