

EmzStyle LLC

# AI PoC が「使え そうで使えな い」理由

生成 LLM と RAG では越えられない判断支援の構造的壁

前田 稔

2026 年 1 月 14 日

## 摘要（要約）

本白書は、生成 LLM と RAG を用いた AI 活用 PoC において頻発する「説明はできるが、判断できない」という違和感を起点に、その原因を技術的性能ではなく構造的欠陥として分析したものである。

多くの PoC では、判断の対象・前提・制約・判断不能条件が定義されないまま実装が進み、

RAG 強化によって情報量や説明力のみが増幅される。

その結果、AI の出力は一見もっともらしくなる一方で、判断責任を負う人間が結果を引き取れない状態に陥る。

本稿は、

- AI は判断を行う存在ではないこと
- 「AI が構造を推論すればよい」という期待が誤解であること
- 判断支援には事前に定義された判断構造が不可欠であること

を明らかにしたうえで、

判断支援を成立させるための必要条件として、

- 判断目的
- 前提条件
- 制約
- 例外・判断不能条件
- 人と AI の役割分担

を整理する。

さらに、これらを安定的に機能させる設計原理として、

**OS（統制）／Adapter（判断支援）／RAG（材料調達）の分離**を提示し、  
その中で EBA（[EmzStyle Business Advisor](https://www.emz-style.com/emzstyle-business-advisor)<sup>1</sup>）が果たす役割を、  
実務者視点から位置づけた。

本白書は、AI 活用の否定や解決策の提示を目的とするものではなく、  
安易な PoC 実装に進む前に

**\*\*「まだ整理されていない判断構造は何か」\*\*を問い直すための**

---

<sup>1</sup> <https://www.emz-style.com/emzstyle-business-advisor>

思考フレームを提供するものである。

## 目次

はじめに .....	1
第 1 章 なぜ AI 活用は「それらしい説明だが、評価できない」状態に陥るのか .....	3
第 2 章 現場で実際に起きている PoC の構造的問題 .....	6
第 3 章 RAG 強化が根本的解決にならない理由 .....	9
第 4 章 よくある反論と、その構造的誤解 .....	12
第 5 章 判断支援を成立させるための必要条件 .....	15
1. 判断の対象と目的 .....	16
2. 前提条件の明示 .....	16
3. 制約条件の設定 .....	16
4. 例外と判断不能条件 .....	17
5. 人と AI の役割分担 .....	17
第 6 章 OS / Adapter / RAG 分離という設計原理 .....	19
OS（統制レイヤ） .....	19
Adapter（判断支援レイヤ） .....	20
RAG（材料調達レイヤ） .....	20
第 7 章 EBA が担う役割（実務者視点） .....	22
1. ヒアリング時 .....	22
2. 構造化分析時 .....	23
3. プロンプト作成時 .....	23
4. 運用時 .....	24
第 8 章 本稿の位置づけと今後の検証課題 .....	26

## はじめに

本書を読み進める前に、あらかじめ一つの**重要な前提**を明示しておきたい。

そもそも AI は、「判断を行う存在」ではない。

生成 AI が行っているのは、与えられた文脈とデータに基づき、  
統計的・確率的にもっともらしい説明や候補を生成することである。

それ自体は有用である一方、

「判断」とは本来、前提と制約を受け、  
その結果に対する責任を伴って行われる行為である。

本書は、AI の出力を前にして

「AI の出してくる答え」に違和感を気が付き始めた人を、  
主な読者として想定している。

もし本書の冒頭で語られる問題意識に

「なぜそれが問題なのか」と感じるならば、  
それ自体が、本書が扱おうとしている構造の入口である。

---

近年、生成 AI や LLM を活用した PoC（概念実証）が各所で進められている。

その多くは、「情報整理が容易になった」「説明資料の作成が高速化した」といった形で、一定の成果を示している。

一方で、実務の現場、とりわけ**最終的な判断責任を負う立場**にある人々からは、別の声が聞かれるようになった。

それは、「説明はできるようになったが、本当に正しく判断しているかどうかわからない」という違和感である。

「ここで指摘してくるのであれば、こちらでも指摘してしかるべきなのだが、そうではない。。。。」

AI が示す出力は一見もっともらしく、情報量も豊富である。

しかし、それを**根拠**として信じ、自らの判断を確定させることに、強い不安を覚える

---

本稿は、この感覚が偶然でも心理的な問題でもなく、**構造的に必然である可能性**に注目する。

---

本稿の対象読者は、AI 活用を企画・推進する立場にありながら、最終的な判断や説明責任を免れない実務者である。  
必ずしも AI 技術の専門家である必要はないが、  
「なぜ AI はこの判断に至ったのか」を自らの言葉で説明する責任を負っている人を想定している。

一方で、本稿は以下の内容を扱わない。

- 特定の AI 製品・ベンダーの評価
- 成功事例や導入ノウハウの提示
- AI による意思決定の自動化手法
- 経営判断の代替や最適解の提示

本稿の目的は、「どうすれば AI で判断できるか」を示すことではない。  
**なぜ現在の AI 活用が、判断を支援するどころか困難にしまうのか、**  
その構造的理由を明らかにすることにある。

---

本稿で扱う中心的な問いは、次の一点に集約される。

論理的な説明、RAG による根拠提示、多角的・網羅的な文章など、  
AI の説明能力が向上しても、  
なぜ実務における「判断」は自動化できないのか。

この問いに対し、本稿は結論を急がない。  
実装結果や完成された理論モデルを提示するのではなく、  
工学的・構造的な観点から、現場で起きている現象を整理し、  
どこに前提の欠落や構造的な誤解があるのかを検討する。

本稿は、AI 活用の可能性を否定するものではない。  
むしろ、安易な実装や誤った期待によって、  
AI 本来の役割が見失われている現状に対し、  
**立ち止まって構造を見直すための思考実験**として位置づけられる。

---

続く第 1 章では、  
多くの PoC で観察される「かなり説明できるようになったが、実運用に耐えるか評価できない。」という違和感を起点に、  
なぜそのような状態が生じるのかを整理していく。

## 第1章 なぜAI活用は「それらしい説明だが、評価できない」状態に陥るのか

多くのAI活用PoCでは、

「ある程度やらせてみて、その結果を知見者が確認する」  
という進め方が採られている。

これは一見、人材育成における

「新人に任せて、上位者がレビューする」構図と似ている。

しかし**両者には、決定的な違い**がある。

人間の新人の場合、知見者は

結果だけでなく、思考過程や前提の置き方、  
迷いのポイントを確認できる。

そのため、判断そのものを評価し、引き取ることができる。

一方、AIが提示するのは結果であり、

そこに至る判断の思想や迷いを確認することはできない。

その結果、知見者は確認時点において

「結果はそれなりに妥当に見えるが、  
本当にこれでよいのかどうか分からない。確認しようがない。」  
という状態に置かれる。

本書が扱う「判断できなくなる現象」とは、

まさにこの違和感が構造的に増幅された状態を指している。

---

AIが提示する説明は、概ね筋が通っている。

情報量も多く、参照元も明示されている。

一見すると、「これ以上何が足りないのか分からない」状態に見える。

それにもかかわらず、

その出力を根拠として「評価する＝判断を確定させる」ことには、  
強い躊躇が生じる。

- なぜ、この結論を提示しているのか
- どの前提が固定され、どこが揺れているのか
- 判断不能の場合、どこで止まるのか

これらが明確にならないまま、  
**処理だけが前に進んでしまう感覚が残る。**

結果として現れるのが、  
「それらしい説明だが、評価<sup>2</sup>できない」  
という状態である。

---

この違和感は、  
利用者の理解不足や心理的抵抗によって生じているのではない。  
また、モデルの性能不足や精度の問題でもない。

むしろ逆に、  
AIの説明能力が向上したからこそ、判断の不安定さが露呈している  
と見るべきである。

説明が洗練され、情報が豊富になるほど、  
判断に必要な「前提」「制約」「例外」「判断不能条件」が  
どこにも明示されていないことが、かえって目立つようになる。

---

ここで重要なのは、  
**判断とは単に「もっともらしい答えを選ぶ行為」ではない、**という点である。

判断には必ず、

- 何を判断しようとしているのか
- どこまでを AI に委ね、どこからを人が引き取るのか
- 判断できない場合、どう扱うのか

といった**構造的な前提**が必要になる。

これらが定義されないまま AI を導入すると、  
AI は説明を続けるが、  
人は判断を引き受けられなくなる。

この構図こそが、

---

<sup>2</sup> 本章で言う“評価できない”とは、意思決定を引き取れない状態を指す。



多くの PoC で繰り返し観察されている現象である。

---

つまり、AI が提示する答えは、判断そのものではないことを認識すべきということである。

それは、「どのような前提関係や制約のもとで物事を捉えるか」という**構造**が成り立つのであれば、

この結果になる可能性が高い、という条件付きの帰結にすぎない。

重要なのは、その構造や前提が**現在の状況と合致しているかどうか**を、人が判別できることである。

構造が明示されていない場合、答えが妥当に見えても、それを今の状況に適用してよいかどうかを判断することはできない。

一方、構造が明示されていれば、

「今はその構造が成り立っているのか」

「一部が変わっていないか」

を確認したうえで、答えを使うかどうかを判断できる。

---

本章では、

AI 活用の成否を分けているのが「技術の高度さ」ではなく、**判断構造の欠如**である可能性を提示した。

続く第 2 章では、

この問題が実際の PoC 現場でどのような形で現れているのかを、より具体的に整理していく。

## 第2章 現場で実際に起きている PoC の構造的問題

AI 活用 PoC の現場では、ある共通した進め方が見られる。  
それは、**設計よりも先に実装が始まる**という進め方である。

LLM や RAG といった技術は、導入の敷居が低い。  
API を接続し、社内文書やデータを投入すれば、  
短期間で「それらしい出力」が得られる。

このため、多くの PoC では次のような流れが生まれる。

- まず動かしてみる
- 出力を見ながら調整する
- 使えそうかどうかを評価する

一見すると合理的に見えるが、  
この進め方には**構造的な欠陥**が含まれている。

---

最大の問題は、  
「何を判断したいのか」が定義されないまま進んでしまう点にある。

多くの PoC では、

- どの業務判断を支援したいのか
- 判断の結果、何が変わるのか
- 判断不能な場合、誰が引き取るのか

といった点が明確にならないまま、  
情報の収集と結合だけが進められる。

結果として、AI は大量の情報を提示するが、  
それを**判断にどう使うのか**は人に委ねられたままになる。

---

この状態は、しばしば  
「精度が足りない」  
「もっとデータを増やせばよい」  
と解釈される。

しかし実際には、

問題は情報量ではなく、**判断構造の欠如**にある。

判断構造とは、

- 判断の目的
- 判断の手順
- 判断に必要な前提条件
- 判断できない場合の扱い

といった要素の集合である。

これらが定義されていない場合、  
AI は「説明」を生成することはできても、  
**判断を成立させる足場を提供できない。**

---

現場では、この状態が次のような形で現れる。

- 出力内容は正しいように見える
- しかし、どこを根拠に判断すべきか分からない
- 最終的には、人が感覚的に決める

結果として、PoC は「使えなくはないが、任せられない」という評価に落ち着く。

ここで重要なのは、  
これは PoC の失敗ではなく、  
**進め方そのものが判断支援に適していない**という点である。

---

さらに問題なのは、  
この状態でも PoC が「前に進んでいるように見える」ことである。

- 処理時間は短縮される
- 説明資料は充実する
- デモとしては成立する

しかし、判断責任を負う立場から見ると、  
**判断の不確実性だけが蓄積されていく。**

このギャップこそが、  
第 1 章で述べた違和感の正体である。

---

本章では、  
多くの PoC が「技術的には進んでいるように見えながら」、  
実際には判断支援から遠ざかっている理由を整理した。

続く第 3 章では、  
この問題が RAG 強化によってなぜ解決しないのか、  
むしろどのような副作用を生むのかを見ていく。

## 第3章 RAG 強化が根本的解決にならない理由

第2章で見たように、多くの PoC では、  
判断構造が定義されないまま、情報の収集と結合が進められている。  
この問題に対し、現場で最も多く採られている対処が、**RAG (Retrieval-Augmented Generation) の強化**である。

すなわち、

- 参照文書を増やす
- 検索精度を高める
- ヒット順を工夫する

といった方法によって、  
「より良い答えが得られるはずだ」という期待が寄せられる。

しかし、結論から言えば、  
**RAG の強化はこの問題の根本的解決にはならない。**

---

RAG が担う役割は明確である。  
それは、**参照可能な情報を適切に取得することであり、**  
判断そのものを設計することではない。

RAG は、

- 何を判断したいのか
- どの前提を固定すべきか
- 判断不能時にどう振る舞うか

といった要素を定義しない。  
それらが与えられていない場合、  
RAG は単に「より多くの材料」を供給する装置として振る舞う。

この点を見誤ると、  
情報が増えるほど判断が容易になるという、  
直感的だが誤った期待が生まれる。

---

実際には、逆の現象が起きる。

参照情報が増えるほど、

- 前提条件の揺らぎ
- 解釈の分岐
- 重要度の異なる情報の混在

が目立つようになる。

判断構造が定義されていない場合、

これらを整理・選別する基準が存在しないため、

**判断の足場そのものが不安定になる。**

結果として、

AI の出力は一見すると豊かで説明的になるが、

判断に使おうとした瞬間に、不確実性が露呈する。

---

ここで注意すべき点は、

この問題を「精度不足」や「モデル性能」の問題として

片付けてしまうことである。

RAG を強化しても、

- 判断の目的
- 進行手順
- 判断不能時の止まり方

が定義されていなければ、

出力は毎回異なる前提の上に構築される。

その結果、

- 昨日は納得できた説明が、今日は腑に落ちない
- 同じ問いを投げているはずなのに、結論の組み立てが違う
- 「なぜ違うのか」を説明できない

といった現象が起きる。

これは、RAG が不安定だからではない。

判断構造が存在しない状態で RAG を使っていることの必然的帰結である。

---

重要なのは、  
RAG が悪いのでも、不要なのでもないという点である。

RAG は、

- 判断に必要な材料を集める
- 情報探索の負荷を下げる

という点では、非常に有効な技術である。

しかし、  
判断構造を与えない限り、RAG は判断支援を完成させない。

それどころか、  
使い込むほどに前提が動き、  
判断の再現性が失われていく。

---

本章では、  
RAG 強化が「説明力の向上」には寄与する一方で、  
判断支援の根本課題を解決しない理由を整理した。

続く第4章では、  
この現象に対してしばしば投げかけられる  
「AI が構造を推論すればよいのではないか」という反論が、  
なぜ成立しないのかを検討する。

## 第4章 よくある反論と、その構造的誤解

——「AI が構造を推論すればよい」という誤解

第3章で述べたように、  
RAG を強化しても判断の不安定さは解消されず、  
むしろ使い込むほど再現性が失われていく。  
この点を指摘すると、必ずと言ってよいほど、  
次の反論が返ってくる。

「AI が構造を推論すればよいのではないか」

この反論は直感的にはもっともらしく聞こえる。

しかし、この一文には、

**AI が行っている処理の性質についての重大な誤解**が含まれている。

---

ここで、一つの挿話を考えてみたい。

日本には「風が吹けば桶屋が儲かる」という有名な落語がある。

風が吹くと砂ぼこりが立ち、目を患う人が増え、  
三味線が売れ、猫が減り、ネズミが増え、  
桶をかじられて、結果として桶屋が儲かる——  
という因果の連鎖である。

この話は、荒唐無稽ではあるが、  
少なくとも「途中に何が起きているか」を仮に置いている。  
つまり、人間が直感的に思い浮かべる「推論」とは、  
本来このように、  
事象と結果の間にある関係を仮定してつなぐ営みである。

---

しかし、ここで問題にしている AI の「推論」は、  
このような仮定を行っていない。

AI が扱っているのは、  
「風が吹いた」という事象と、  
「桶屋が儲かった」という結果の結びつきであり、  
その間にある因果の中身ではない。



言い換えれば、  
AI が提示しているのは、  
「A であれば B になりやすい」という対応関係である。  
それは統計的にもっともらしい形で示されるが、  
A と B の間にある仕組みや理由を  
仮説として立てて検証しているわけではない。

---

一般に「推論」という言葉から想起されるのは、  
因果関係を段階的に組み立てる思考である。

- ある前提があり
- 途中に関係性があり
- それらを踏まえて結論に至る

という構造を、多くの人は無意識に想定している。

しかし、現在の LLM が行っている処理は、  
このような因果構造の組み立てではない。  
LLM は、与えられた文脈の中で、  
条件と結果の結びつきを  
確率的に再構成しているに過ぎない。

---

この違いを見誤ると、  
AI が因果関係を理解し、  
構造を推論しているかのように見えてしまう。  
しかし実際には、  
AI は「結びつき」を示しているだけであり、  
その中身を考えているわけではない。

第 3 章で述べた

「使い込むほど出力が不安定になる」現象は、  
この誤解の帰結として理解できる。

RAG によって参照情報が変わり、  
文脈の重み付けが変わるたびに、  
A と B の対応関係が組み替えられる。  
その結果、

- 昨日と今日で、結論の形が変わる
- なぜ変わったのかを説明できない
- 同じ業務を同じやり方で再現できない

という事態が生じる。

これは「高度な推論が行われている」状態ではない。

同一の対応関係を安定して再現できなくなっている状態である。

---

ここで改めて、先の挿話に戻る。

「風が吹けば桶屋が儲かる」は、  
因果関係を仮に置いた説明の物語である。

一方、LLM+RAG による出力は、  
この途中の因果を仮定することなく、  
事象と結果の結びつきだけを扱っている。

したがって、

「AI が構造を推論すればよい」という反論は、  
前提の段階で成立していない。

AI は、構造を推論していない。

統計的に、

「これが正しいように見える」と示しているだけである。

---

本章では、

「AI が構造を推論すればよい」という反論が、  
なぜ誤解に基づいているのかを整理した。

次章では、

このような AI の性質を前提としたうえで、  
判断支援を成立させるために、  
人間側がどのような構造を明示的に与える必要があるのかを、  
整理していく。

## 第5章 判断支援を成立させるための必要条件

第4章までで見てきたように、  
AI活用PoCが「説明できるが、判断できない」状態に陥るのは、  
技術の未熟さやモデル性能の問題ではない。

問題は一貫して、  
判断を成立させるための条件が定義されていないことにある。

本章では、  
判断支援が成立するために最低限必要となる条件を、  
技術論ではなく**構造**として整理する。

---

まず確認すべき点は、  
判断とは「答えを出すこと」ではない、ということである。

判断とは、

- 不確実な状況下で
- 複数の選択肢の中から
- 責任を引き受ける形で
- 行動を確定させる行為

である。

したがって、  
判断支援とは「正解を提示すること」ではなく、  
**判断を引き受けられる状態を成立させることを意味する。**

なお、本稿で用いる「判断」という言葉は、日常的に使われている意味よりも、厳密に定義されている。

実務において混同されがちな行為は、次の三つに分けて考える必要がある。

### 1. AIによる出力生成

与えられた文脈とデータに基づき、条件付きの結論や説明を生成する行為。

### 2. AI出力の評価

その出力が妥当に見えるか、違和感がないかを確認する行為。

### 3. 人間による意思決定

不確実性と責任を引き受け、「これで進む」と行動を確定させる行為。

本稿で問題にしているのは、①や②が高度化することで、  
③が行われていないにもかかわらず、あたかも判断が完了したかのように錯覚される  
構造である。

---

判断支援を成立させるためには、  
少なくとも次の要素が明示されていなければならない。

## 1. 判断の対象と目的

- 何について判断しようとしているのか
- 判断の結果、何が変わるのか

これが定義されていない場合、  
AI の出力は説明の集合に留まり、  
判断に結びつく足場を持たない。

---

## 2. 前提条件の明示

- どの条件を固定して考えるのか
- どの条件が変動しうるのか

前提が暗黙のままでは、  
出力ごとに異なる判断軸が持ち込まれ、  
再現性が失われる。

---

## 3. 制約条件の設定

- 法令・規程・予算・時間などの制約
- 判断において越えてはならない境界

制約が定義されていない場合、  
AI は「理屈として正しいが、使えない」提案を生成し続ける。

---

#### 4. 例外と判断不能条件

- 想定外のケースをどう扱うか
- 判断できない場合、どこで止まるのか

この定義がないままでは、  
AI は常に何らかの説明を生成し続け、  
人は「止めどころ」を失う。

---

#### 5. 人と AI の役割分担

- どこまでを AI が支援し
- どこからを人が引き取るのか

役割分担が曖昧な場合、  
AI の出力は責任の所在をぼかし、  
判断者の不安を増幅させる。

---

重要なのは、  
これらの条件は**後から出力を見て調整できるものではない**という点である。

判断支援においては、

- 情報取得
- 推論（と呼ばれている処理）
- 出力生成

の前に、  
判断構造そのものが設計されていなければならない。

これが欠けた状態では、  
どれほど高度な AI を用いても、  
判断は成立しない。

---

本章で整理したのは、  
「こうすればうまくいく」という方法論ではない。

むしろ、

これらが定義されていない限り、  
判断支援は成立しない

という**必要条件**である。

続く第 6 章では、  
これらの条件をどのようにシステムとして分離・配置すべきか、  
OS／Adapter／RAG という設計原理として整理する。

## 第 6 章 OS / Adapter / RAG 分離という設計原理

第 5 章では、判断支援を成立させるための必要条件を整理した。

これらの条件は、AI の性能向上によって自然に満たされるものではない。

むしろ、どの機能をどのレイヤに配置するかという設計判断に強く依存する。

本章では、抽象度を一段上げ、判断支援を安定して成立させるために必要となる **OS/Adapter/RAG 分離**という設計原理を提示する。

---

多くの PoC が破綻してきた理由の一つは、  
判断・推論・情報取得といった異なる役割が、  
**同一レイヤに混在している**点にある。

具体的には、

- 判断の統制
- 判断を助ける処理
- 判断材料の取得

が明確に分離されないまま、  
一つの AI システムとしてまとめられてきた。

その結果、  
どこで前提が決まり、  
どこで判断が成立し、  
どこで処理が止まるのかが不透明になる。

---

ここで提案する分離は、  
機能の優劣ではなく、**役割の違い**に基づくものである。

### OS（統制レイヤ）

OS は、判断を直接行うものではない。  
その役割は、**判断構造を定義・固定すること**にある。

- 判断の目的
- 前提条件
- 制約

- 判断不能条件
- 人と AI の役割分担

これらを明示し、  
判断の進め方が毎回変わらないように統制する。

OS が存在することで、  
「何をしてよいか」「どこで止まるべきか」が初めて確定する。

---

## Adapter（判断支援レイヤ）

Adapter は、OS で定義された判断構造の中で、  
人の判断を支援する役割を担う。

- 情報の整理
- 観点の提示
- 選択肢の比較
- 判断材料の可視化

ここで重要なのは、  
Adapter が**判断を代行しない**という点である。

Adapter はあくまで、  
判断者が引き受けるべき判断を、  
引き受けられる形に整えるための装置である。

---

## RAG（材料調達レイヤ）

RAG は、判断に必要な情報を取得するための仕組みである。

- 文書検索
- データ参照
- 過去事例の取得

といった役割を担うが、  
判断構造を定義する権限は持たない。

RAG は、OS や Adapter から要求された条件に従って、  
材料を供給する位置に置かれるべきである。



---

この三つを分離することで、  
これまで PoC で繰り返されてきた問題を構造的に回避できる。

- RAG が前提を動かしてしまう
- Adapter が判断を代行してしまう
- 判断不能でも処理が進んでしまう

といった事態は、  
レイヤ分離がなされていないことに起因する。

---

重要なのは、  
この分離が「アーキテクチャの選択肢」ではなく、  
**判断支援を成立させるための必要条件**である点である。

OS が判断構造を固定し、  
Adapter が判断を支援し、  
RAG が材料を供給する。

この関係が保たれて初めて、  
AI は判断を脅かす存在ではなく、  
判断を支える存在として機能する。

---

本章では、  
判断支援が破綻してきた理由を、  
レイヤ混在という観点から整理し、  
OS／Adapter／RAG 分離という設計原理を提示した。

続く第 7 章では、  
この設計原理の中で、  
EBA ([EmzStyle Business Advisor](#)) がどの役割を担うのかを、  
実務者の視点から整理する。

## 第 7 章 EBA が担う役割（実務者視点）

第 6 章では、判断支援を成立させるための設計原理として、OS／Adapter／RAG 分離という考え方を整理した。

本章では、その中で

EBA（[EmzStyle Business Advisor](#)）が担う役割を、実務者の視点から具体的に位置づける。

最初に明確にしておくべき点は、

EBA は**判断を代行する AI ではない**ということである。

EBA の役割は一貫して、

判断構造の整合性を点検し、可視化することにある。

---

多くの PoC で問題となっているのは、

判断の内容そのものが誤っていることではない。

- なぜその判断に至ったのか分からない
- どの前提が固定され、どこが揺れているのか説明できない
- 判断したのか、判断できなかったのかが曖昧

といった、

判断の状態そのものが把握できないことにある。

EBA は、この「見えない状態」に対して介入する。

---

EBA は、業務プロセスのすべてに常駐する AI ではない。

判断が不安定になりやすい**特定の局面**においてのみ、その役割を発揮する。

実務上、EBA の使いどころは、次の四つのフェーズに整理できる。

---

### 1. ヒアリング時

EBA が最初に力を発揮するのは、要件整理や初期ヒアリングの段階である。

この段階では、多くの場合、

- 目的と手段が混在している

- 判断したいことと、説明したいことが混ざっている
- 制約や前提が暗黙のまま語られている

といった状態が見られる。

EBA は、ここで結論を出さない。

代わりに、\*\*「まだ定義されていない判断条件」\*\*を浮かび上がらせる。

これにより、

後工程で「判断できない理由が分からない」状態に陥ることを未然に防ぐ。

---

## 2. 構造化分析時

次に EBA が関与するのは、

議論や情報を構造として整理する段階である。

この局面では、

- 論点の粒度が揃っていない
- 前提と結論が循環している
- 判断不能条件が存在しない

といった構造的欠陥が顕在化しやすい。

EBA は、

正しい答えを導くのではなく、

**判断構造として成立しているかどうか**を点検する。

説明としては成立しているが、

判断に使えない構造を、ここで明確に切り分ける。

---

## 3. プロンプト作成時

三つ目の介入点が、LLM に対するプロンプト設計の段階である。

ここで EBA が担うのは、

「良いプロンプト」を生成することではない。

EBA は、

- このプロンプトは、どの判断を支援しようとしているのか

- 前提や制約は固定されているか
- 判断不能時の振る舞いが定義されているか

といった、**進め方の固定**を行う。

この整理がないままでは、  
プロンプトを工夫しても、  
出力の再現性は確保されない。

---

## 4. 運用時

EBA が最も重要な役割を果たすのが、実際の運用フェーズである。

多くの PoC では、導入直後は一定の評価を得るが、  
時間の経過とともに次のような違和感が現れる。

- 昨日は納得できた出力が、今日は信用できない
- 同じ問いを投げているはずなのに、進め方が違う
- 「なぜそうなったのか」を説明できない

この段階で問題となっているのは、  
精度やモデル性能ではない。

判断の前提や進め方が、運用の中で徐々にずれていくことである。

EBA は運用時において、

- 初期に定義した判断前提が維持されているか
- 判断不能条件が無視されていないか
- いつの間にか AI が判断を代行していないか

といった点を点検する。

これは改善提案ではなく、  
**判断構造のドリフトを検知する役割**である。

---

これら四つの介入点に共通するのは、  
EBA が一貫して判断を代行しないという点である。

EBA は、

- 判断を速くする装置ではなく
- 判断を賢くする装置でもなく

判断できない理由を、構造として明らかにする装置である。

---

このため、EBA は  
導入した瞬間に成果を出すタイプの AI ではない。

むしろ、

- PoC がなぜ前に進まないのか
- なぜ説明はできるのに判断できないのか

といった違和感が顕在化したときに、  
初めてその価値が理解される。

EBA は、  
判断を高速化するための AI ではなく、  
判断を安定して引き受けるための基盤として位置づけられる。

EBA は万能ではない。参考までに EBA が向かないケースも挙げておく。

- 定型業務
- 判断責任が発生しない処理
- 完全自動化前提の業務

---

本章では、  
EBA の役割を「できること」ではなく、  
\*\*「どこで、何をしないか」\*\*という観点から整理した。

最終章では、  
本稿全体の位置づけを再確認した上で、  
今後検証すべき論点と PoC の切り方について整理する。

## 第 8 章 本稿の位置づけと今後の検証課題

本稿では、AI 活用 PoC がなぜ

「説明できるが、判断できない」状態に陥るのかを起点に、その原因を技術論ではなく、**構造の問題**として整理してきた。

繰り返しになるが、本稿は

AI の性能評価や実装手法を示すものではない。

また、判断を自動化する方法論を提示するものでもない。

本稿の位置づけは、

判断支援が成立しない理由を、構造として言語化するための思考実験である。

---

第 1 章から第 4 章にかけて示したのは、

多くの PoC で観察される違和感と、その背後にある誤解である。

- 説明力の向上が、判断力の向上と誤認されていること
- RAG 強化が、判断構造の欠如を覆い隠してしまうこと
- 「AI が推論すればよい」という言葉が、  
人間の推論と異なる意味で使われていること

これらは個別の失敗ではなく、

同じ構造的前提のもとで必然的に生じている現象である。

---

第 5 章から第 7 章では、

判断支援を成立させるための必要条件と、

それを支える設計原理、実務上の介入点を整理した。

ここで提示した

OS／Adapter／RAG 分離という考え方や、

EBA の役割は、完成された解ではない。

しかし、少なくとも次の点は明確になった。

- 判断支援は、情報量や精度の問題ではない
- 判断構造は、後から生成物を見て調整できるものではない
- 運用の中で判断が不安定になるのは、偶然ではない

これらを見放したまま PoC を進める限り、

同じ問題は形を変えて繰り返される。

---

本稿が提示するのは、

「こうすれば成功する」というロードマップではない。

むしろ、

- どこを整理しない限り、判断支援は成立しないのか
- どの前提を曖昧にしたまま進むと、PoC は必ず歪むのか

といった、検証すべき論点の一覧である。

今後の PoC において検証されるべきポイントは、例えば次のようなものになる。

- 判断構造を明示的に定義した場合、出力の再現性はどう変わるか
- 判断不能条件を設計に組み込んだとき、運用はどう変化するか
- OS 的統制を導入した場合、RAG や LLM の使われ方はどう変わるか

これらは、実装前に整理できる問いであり、

**実装によって初めて気づくべきものではない。**

---

最後に強調しておきたいのは、

本稿が AI 活用を否定するものではない、という点である。

本稿が問題にしているのは、

AI に対して過剰な期待を抱くことでも、

技術を過小評価することでもない。

判断という行為の構造を整理しないまま、

AI にそれを委ねようとする姿勢である。

判断の構造が明確になったとき、

AI は判断を脅かす存在ではなく、

判断を支える道具として、初めて安定して機能する。

---

本稿が、

安易な実装に進む前に立ち止まり、

「何をまだ整理していないのか」を問い直すための  
一つの参照点となることを願っている。

## 本ホワイトペーパーについて

本ホワイトペーパーは、

前田 稔（エムズスタイル LLC）による独自の調査・分析および  
構造知性フレームワークに基づき作成されています。

本資料は、特定の解決策や結論を提示するものではなく、  
判断に必要な構造や視点を整理することを目的としています。

---

## 著作権・利用条件

本資料に含まれる文章・図表・分析内容・構造フレームワークは、  
著作権法および関連法令により保護されています。

本資料の利用条件は、以下に定める

「ホワイトペーパー利用規約」に従うものとします。

 <https://emz-style.com/whitepaper-terms>

---

## 利用区分の概要

- 無料版（要約・抜粋）  
社内共有・紹介目的での利用は可能です（改変・商用利用不可）
- 有料版（個人）  
個人学習目的に限り利用可能です（社内共有不可）
- 法人向けライセンス  
社内での配布・研修・教育用途での利用が可能です

※詳細は上記利用規約をご確認ください。

---

## 最後に

本資料をお読みになり、

- 判断に迷う点がある
- 自社の状況に当てはめると違和感がある
- このまま進めてよいのか確信が持てない

と感じられた場合は、

それ自体が重要なサインです。

ご相談・ご質問は、以下よりお気軽にお寄せください。

 <https://emz-style.com/contact>

（※法人向けのご相談・講演・研修のご依頼もこちらから承っています）