AI プロダクト品質保証ガイドラインの自社適用と RAG システムにおける定量評価の実践

Applying QA4AI Guidelines In-House and Practicing Quantitative Evaluation in a RAG System

エムオーテックス株式会社
MOTEX Inc.
○河内 晴菜　玉田 雄大
○ Haruna Kochi　Yudai Tamada

Abstract　This paper reports our first in-house AI feature release: a RAG-based help-chat function delivered by adapting the Guidelines for Quality Assurance of AI-based Products and Services (QA4AI) to our product and designing a multi-faceted scoring evaluation based on expectation decomposition. With no in-house expertise on AI quality and no established criteria for judging AI output, we positioned a quality assurance engineer as a bridge between development and business decision-making, and built both a consensus-building framework and the scoring evaluation around that role. The effort established a shared quality standard among stakeholders and, by presenting concrete negative examples, eased consensus-building with business owners. It also showed that release decisions cannot rest on quantitative metrics alone: quantitative measures, qualitative judgment, and stakeholder consensus must all be combined. Because the approach does not depend on any specific technology, it remains applicable as AI systems evolve and is transferable to similar features in other systems.

1. はじめに

生成 AI を利用したテストの効率化（AI4QA）と，生成 AI を組み込んだ製品自体の品質保証（QA4AI）は，似て非なる取り組みである。本稿が扱うのは後者，生成 AI を用いた製品の品質保証に関してである。

研究対象は，自社の IT 資産管理ソフトウェアに搭載した，RAG 形式のヘルプチャット機能である。ユーザーの質問に対して回答本文と参照先となるマニュアルサイトの URL を返す，シンプルな構成である。自社では製品の企画・開発から販売，導入後のサポートまでを一貫して提供しており，本機能の回答品質はすなわち自社が提供するサービス全体の顧客体験に直結する。

この機能開発に際して，まず最初に，AI プロダクト品質保証ガイドライン [1] を調査した。これは開発者との認識合わせや AI システム全体の要点を把握する上で大いに役立った。ただし，その内容は特定の技術（機械学習・生成 AI 等）に限定されない包括的なものであるため，RAG や本製品品質特有の論点（検索結果や参照 URL の評価など）については独自の解釈を作成し，プロジェクトに落とし込む必要があった。

前述したように，自社では製品の企画・開発から販売，導入後のサポートまでを一貫して提供する組織体制を採っており，AI 機能のリリース可否についても社内のビジネスドメインの責任者が最終的に判断する。開発側が積み上げた解釈や評価は，その判断に資する形で示されて初めて意味を持つ。

本取り組みのかなめは，ビジネスドメインの責任者に AI 機能の品質について納得感を持ってもらい，一過性のリリース判定に留まらず，本機能が継続的に利用され，全社的な価値貢献へとつながる状態を築くことにある。自社ではこれまで AI 機能を搭載した製品開発を行った実績やノウハウがなく，責任者の側にもリリース可否を判断するための材料がない状態からのスタートであった。

エムオーテックス株式会社 プロダクトサービス統括本部 開発本部 品質管理部
MOTEX Inc.
〒532-0011 大阪市淀川区西中島 5-12-12 エムオーテックス新大阪ビル　Tel: 06-6308-8989
5-12-12 Nishinakajima, Yodogawa-ku, Osaka 532-0011, Japan

キーワード：AI プロダクト品質保証ガイドライン，QA4AI，RAG，生成 AI

2. 背景・課題

AI 機能のリリース可否については社内のビジネスドメインの責任者の最終判断が大きく関わっている。そこで，共通言語を整えるために AI プロダクト品質保証ガイドラインを調査した。開発者との認識合わせや AI システム全体の論点把握には役立った一方，内容が特定技術に限定されない包括的なものであったり，章によっては機械学習に限定される記述もあるため，RAG や自社製品特有の論点は別途独自解釈が必要だった。

従来の製品改修の動作テストでは，あらかじめ定めた手順と期待値が一致するかどうかで合否を判定できていた。しかし，生成 AI を用いたヘルプチャット機能では，同一の質問に対する回答であっても表現が一意に定まらないため，製品仕様から期待値を導出する従来の演繹的な手法が通用しない。

例えば，「OS のサポートが終了しているデバイスの確認方法を教えてください」という問い合わせに対して，AI は以下のように回答した。

1. 下記ドキュメントの「OS のサポート状況を確認します」の項目を参照してください。
2. ［リスト］の［Windows アップデート］をクリックし，「OS のサポートが終了しているデバイス」レポートを確認します。
3. OS のサポート状態が「サポート終了」または「サポート終了間近」のデバイスを確認できます。
4. OS をアップデートする場合は，デバイスをチェックし，［インストール設定］をクリックして設定を行います。

参照ドキュメント：`https://example.com/manual/xxxxx`

一見すると正しい回答に見えるが，本製品では iOS など Windows 以外の OS にも対応しており，Windows の手順しか提示していないこの回答は本来不十分である。この不十分さを見抜くには製品知識や実動作による確認が必要であり，機械的な正誤判定はできなかった。同時に回答に添えられた参照ドキュメントについても，過不足がないかを一件一件，調査する必要がある。このように，AI の生成結果に対して検証すべき評価軸は，回答本文の妥当性や参照ドキュメントの過不足だけでなく他にも多数存在し，かつユーザーの質問の傾向は多岐にわたることが想定される。そのため，評価用のデータを網羅的に用意することは容易ではない。

AI が生成した回答の品質を検証できないという課題に対しては，RAGAS の応用で，AI が生成した回答を他の AI が評価する独自の評価を試みた。その結果，質問とドキュメントの関連度を表す検索精度スコアとして 0.0034 という値が得られた。しかし，社内には生成 AI 機能の品質を評価した実績や比較対象となる知見が存在せず，この値が「リリース可能」と判断できる水準にあるのかを，品質保証の観点から客観的に説明することができなかった。これは，この手法が提供するのはあくまで「測定」の結果であり，「評価」ではないことを意味する。数値の羅列を，リリース可否という意思決定に接続する仕組みが別途必要であった。

以上のように，AI の回答品質は単純な正誤判定になじまず，評価に必要なデータを網羅的に用意することも容易ではない。加えて，たとえ何らかの指標によって回答品質を数値化できたとしても，その数値が「リリース可能」と判断できる水準にあるのかを客観的に説明する手段がない。これらはいずれも，AI の回答品質に対する開発者とビジネスドメインの間の認識のギャップとして現れる。このギャップを埋めることが，本取り組みの出発点となった。

3. 仮説

前章で述べた課題に対し，以下の 2 点を仮説として設定した。

仮説 (1)：自社に AI 品質評価の知見がない中でも，AI プロダクト品質保証ガイドラインが示すチェック項目を自社の RAG 形式ヘルプチャット機能に当てはめることで，品質を説明可能な状態にできるのではないか。チェック項目を「目指すべき品質の地点」を示す指標として位置づければ，評価すべき観点の抜け漏れを防ぎながら，品質を説明できるようになるのではないかと考えた。

仮説 (2)：回答品質については，製品仕様から期待値を導出する演繹的な方法ではなく，実際の問い合わせ実績から検証する帰納的な方法が有効ではないだろうか。そのためには仮説 (1) のような定性的な説明可能性だけでなく，何らかの形で品質を数値化することが必要である。そこで，自社サポートチームの有人チャットに実際に寄せられた

ユーザーの問い合わせデータを用いて，質問の傾向・回答の傾向を観測できないかと考えた。

以上より，(1) ガイドラインを自社の品質特性に分解して説明可能にすること，(2) 実際のユーザー問い合わせデータを用いて回答品質を帰納的に検証する方法を調査すること，の 2 点を軸に検証を進めることとした。

4. 解決策の提案

4.1 ガイドラインの自社解釈と合意形成フレームワークの構築

AI プロダクト品質保証ガイドラインが提示する 5 軸を起点に，RAG 形式のヘルプチャット機能へ適合した自社解釈リストを構築した。表 1 に，各軸の内容，主担当チーム，主な報告先ロールを示す。各軸について，解釈・影響度・段階リリース時の達成水準・行動指針・報告先を独自に定義し，開発プロジェクトリーダー（以下，開発 PL）・AI 開発チーム・QA チームの 3 者による読み合わせを経て，責任者との合意基盤とした。ガイドラインが未カバーの RAG 固有論点（検索データの入出力など）についても，この過程で解釈を作成した（以下，この成果物を合意形成フレームワークと呼称する）。

表 1: AI プロダクト品質保証ガイドラインの 5 軸と自社での位置づけ

軸	内容（自社解釈）	主担当	主な報告先
Data Integrity	回答生成の元となるマニュアル等，データの妥当性・網羅性	AI 開発	開発 PL
Model Robustness	表記ゆれや想定外の質問に対する回答の頑健性	AI 開発	開発 PL
System Quality	AI 部分を含むシステム全体としての機能・非機能品質	QA	開発 PdM
Process Agility	継続的な改善サイクルを回す開発・運用プロセス	QA	開発 PdM
Customer Expectation	ユーザーが AI に対して抱く期待値のコントロールと充足	共同	BO・PMM

このように，5 軸それぞれを主に関心・責任がある報告先ロール別に整理したことで，報告時にどのロールへ何を伝えるべきかの判断材料とした。図 1 に，本取り組みの開発プロジェクト体制を示す。開発 PL の上位にはプロダクトマネージャー（PdM）とプロダクトマーケティングマネージャー（PMM）が横並びで位置し，さらにその上位にビジネスオーナー（BO）が置かれた。

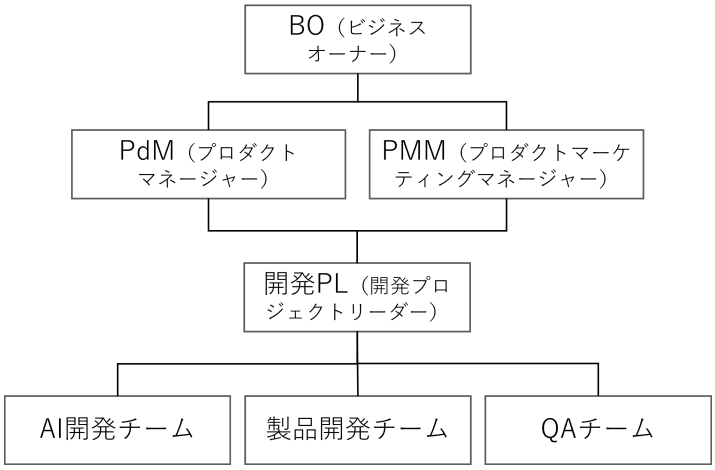


図 1: プロジェクト体制図

4.2 期待値分解に基づく多面的スコア評価の設計

仮説 (2) で述べた，実際のユーザー問い合わせデータを用いて回答品質を帰納的に検証するための具体的な方法として，期待値を分解し多面的にスコア評価する手法を設計した。本機能において，AI は事前にインプットされたマニュアルデータからユーザー質問に沿って，回答本文と参照 URL（最大 4 件）を返却する構成である。回答本文と URL のそれぞれに独立した評価軸を設け，期待値を分解したうえで多面的にスコア化する設計とした。

本文の評価は，6 段階のルーブリック評価とした（十分=1，不十分だが誤りなし=0.7，解釈に揺れあり=0.5，必要情報欠如または誤回答=0，回答守備範囲外質問への正しい留保=1，致命的設定ミスを誘発し得る回答 $= -10$）。なお，マニュアルに記載がなく AI が本来回答できない「回答守備範囲外」の質問が実際には多数存在することが後の実験で判明するのだが，その事例は §5.2 で詳述する。

URL の評価は，参照 URL（最大 4 件）に対し質問解決に有効か否かを判定し，回答守備範囲内か範囲外かによって算出式を分けた。参照 URL のうち質問解決に有効であった URL の合計数を $N_{\text{有効}}$，真の回答である URL の合計数を $N_{\text{真}}$，返却された URL の合計数を $N_{\text{返却}}$ とする。さらに，真の回答のうち提示できなかった URL 数を $N_{\text{不足}}$，返却した URL のうち不要だった過剰 URL 数を $N_{\text{過剰}}$ とすると，これらの間には次の関係が成り立つ。

$$N_{\text{真}} = N_{\text{有効}} + N_{\text{不足}} \tag{1}$$

$$N_{\text{返却}} = N_{\text{有効}} + N_{\text{過剰}} \tag{2}$$

すなわち $N_{\text{有効}}$ は真の回答と返却 URL の双方に共通する URL 数であり，$N_{\text{不足}}$・$N_{\text{過剰}}$ が 0 に近いほど（$= N_{\text{有効}}$ が $N_{\text{真}}$・$N_{\text{返却}}$ に近いほど）望ましい回答であることを意味する。

(1) 回答守備範囲内の質問　真の回答 $N_{\text{真}}$ が存在する場合，非不足度合い・非過剰度合いを次式で定義し，両者の和を URL スコアとした。

$$\text{非不足度合い} = \frac{N_{\text{有効}}}{N_{\text{真}}} \quad (\text{Max} = 1) \tag{3}$$

$$\text{非過剰度合い} = \frac{N_{\text{有効}}}{N_{\text{返却}}} \quad (\text{Max} = 1) \tag{4}$$

$$\text{URL スコア} = \text{非不足度合い} + \text{非過剰度合い} \quad (\text{Max} = 2) \tag{5}$$

非不足度合いは真の URL のうちどれだけ返却できたか（情報検索分野における再現率 (Recall) に相当），非過剰度合いは提示した URL のうちどれだけが正解だったか（適合率 (Precision) に相当）を表す。**いずれも値が大きいほど良い評価であり**，$N_{\text{不足}} = N_{\text{過剰}} = 0$ （＝過不足なく正解 URL のみを提示できた）のときに最大値の URL スコア $= 2$ となる。

なお，設計の過程では，$N_{\text{有効}}$ を $N_{\text{真}}$ と $N_{\text{返却}}$ の和で割る単純な一致率も検討した。しかし，この指標では不足と過剰のどちらが原因でスコアが下がっているのかを区別できず，改善すべき点が分かりにくいという問題があったため不採用とした。非不足度合い・非過剰度合いに分解したことで，どちらの問題が生じているかを個別に特定できるようになった。

(2) 回答守備範囲外の質問　回答守備範囲外（マニュアルに記載がない）の質問には真の回答 $N_{\text{真}}$ がそもそも存在しないため，非不足度合いは算出せず不採用とした。この場合は $N_{\text{有効}} = 0$ となり，返却した URL はすべて過剰 URL（$N_{\text{過剰}} = N_{\text{返却}}$）とみなされる。ただし，これは内容自体が誤りである（誤情報である）と断定するものではない。一方で，返却される URL の数が多いほど，ユーザーが 1 件ずつ内容を確認する手間が増え，利便性を損なうと考えた。そこで，「誤りとまでは言えないが，多いほど利便性を下げる」という位置づけとして，返却された URL のうち不要であった過剰 URL 数 $N_{\text{過剰}}$（最大 4 件）に応じて，次式で緩やかに減点する過剰度合いを算出した。**こちらも値が大きいほど良い評価であり**，過剰 URL が 1 件増えるごとに 0.25 ずつ減点される。

$$\text{過剰度合い} = 1 - N_{\text{過剰}} \times 0.25 \quad (\text{Max} = 1) \tag{6}$$

5. 実験

5.1 ガイドラインの自社適用

ガイドラインの自社解釈を行い，合意形成フレームワークを構築する過程で，従来の製品改修の動作テストでは想定されなかった「ユーザーの期待値をコントロールすること」などの論点が現れた。

実際に作成した自社解釈リストは，ガイドライン原文の列（分類軸〈5 軸〉・中項目・小項目）に，自社で検討した列（No.・重要度・優先度・補足・解釈・成果物・達成基準・報告先・協議メモ）を付け加えた一覧で整理した。表 2 に，その記載例を 2 件抜粋して示す。

表 2: 自社解釈リストの記載例（抜粋）

	ガイドライン原文			自社での検討結果	
No.	分類軸	中項目	小項目（チェック項目）	解釈（自社独自の判断）	報告先
27	Data Integrity（適切なデータ確保）	(e') 検索データの複雑性	(e.i) 学習させたい推論機能に対して，必要以上の情報量や傾向を含む複雑なデータとなっていないか	単一のデータに過剰に複数のデータが集約されたことを，複雑なデータとして定義した	開発 PL
83	Customer Expectation（顧客期待の管理）	(b) ステークホルダーの技術理解度	(b.i) 顧客は確率的動作という考え方を受容していないか	製品・マニュアルでの AI 特性の説明や注意喚起，契約での合意，ユーザーへの啓蒙活動を対策として位置づけた	BO・PMM

※分類軸・中項目・小項目の 3 列はガイドラインに記載された文言・項番をそのまま転記したものであり，解釈・報告先の 2 列は自社での検討により定めたものである。

No.27（検索データの複雑性）は，単一のインプットデータに過剰にデータが集約された状態では検索性能に影響を与えうる恐れがあることを示唆する。これを予防する必要があるという論点を持ち，インプットデータの情報密度に差がないかなどを確認することが対策となった。実験の結果，確かに，インプットデータの情報密度に差があると，検索結果の精度に影響を与えることが判明し，インプットデータの区切り方については調整が行われた。本件は内部処理に関しての仕様・技術的な論点であるため，報告先・承認レベルは開発 PL で足りると合意した。

No.83（ステークホルダーの技術理解度）は，顧客が AI の確率的動作という特性を受容できているかを論点とした。対策として，製品・マニュアルでの AI 特性の説明や注意喚起，利用規約での合意，ユーザーへの啓蒙活動を整理し，ユーザーストーリー・利用規約・製品プロモーション・サポート方針を成果物，ステークホルダーの合意を達成基準として位置づけ，報告先を BO・PMM とした。

このように，AI 機能特有の観点は開発チーム内だけでなく，経営層に近いレイヤーへの報告・合意形成まで見据えて整理する必要性にプロジェクト計画時点で気づき，実行することができた。これを要件定義フェーズで開発 PL，AI 開発チーム，QA が読み合わせたことで，プロジェクトメンバー間での共通認識（合意形成フレームワーク）を持てた（詳細は §6 で述べる）。

5.2 評価セットの構築とスコア算出

仮説 (2) で述べた帰納的な検証を実践するため，前節で設計した期待値分解に基づくスコアを実データに適用した。実際のユーザー問い合わせ履歴から 224 件の評価セットを構築し，実運用時の入力を用いることで，評価セットが運用開始後の入力分布から乖離することを抑制した。

しかし，評価セットでの検証を開始した直後に，想定していなかった問題に直面した。評価セットの中に，例えば次のような問い合わせが含まれていたのである。

> 「お問い合わせ番号：xxxxxxxx の続きをお話できればと思いますが，……(省略)」
> （固有の内容を含むため一部伏字にしている）

このような問い合わせはマニュアルの内容から回答できるものではなく，そもそも想定していなかった種類の質問であった。そのため，評価セットは回答守備範囲（129 件）と守備範囲外（95 件）に分類したうえで評価する必要が生じた。

評価方法についても課題があった。AI の回答がマニュアルの記載に基づく正しいものかどうかを判断するには製品知識が必要であり，224 件すべてを QA 担当者のみで評価することは困難であった。そこで，マニュアルに詳しいドキュメントチームに評価への協力を要請した。

なお，ルーブリックの基準そのものは QA 担当者の判断で確定させたが，224 件の実際の採点作業はマニュアルに詳しいドキュメントチームの協力を得て実施した。基準策定を複数人でレビューする体制まで時間的制約から詰めきれなかった点は，結論で述べる今後の改善余地としている。

以上の評価軸に基づき，224 件の評価セットにスコア設計を適用した。本文スコアと URL スコアの集計結果を表 3 に示す。さらに，回答可能な質問（129 件）について，本文スコアと URL スコアの相関を表したものを図 2 に示す。

表 3: 評価セット 224 件のスコア集計

対象質問の分類	スコアカテゴリ［最小〜最大］	平均	中央値	ばらつき
回答可能（129 件）	本文［0〜1］	0.64	0.70	0.39
	URL［0〜2］	1.16	1.50	0.80
回答不可（95 件）	本文［0〜1］	0.40	0	0.46
	URL［0〜1］	0.85	1	0.26

※ 値は大きいほど評価が高い。

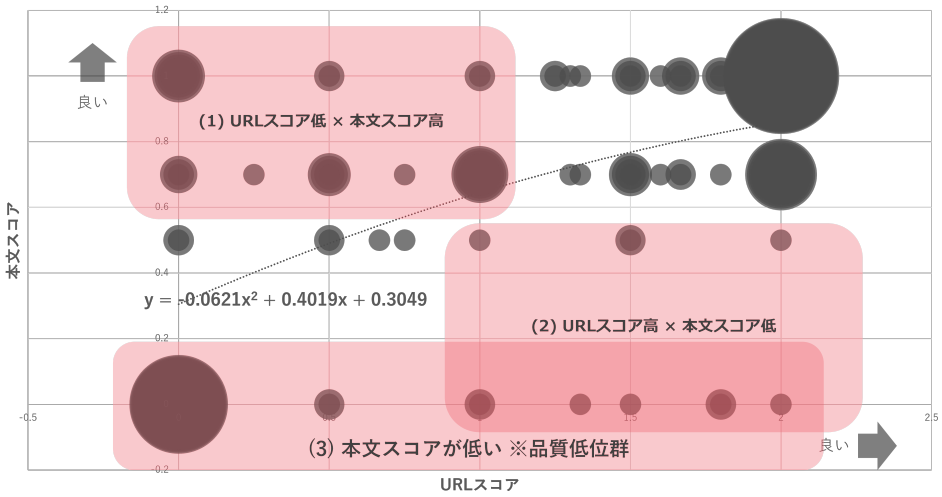


図 2: 本文スコアと URL スコアの相関（回答可能な質問，象限別）

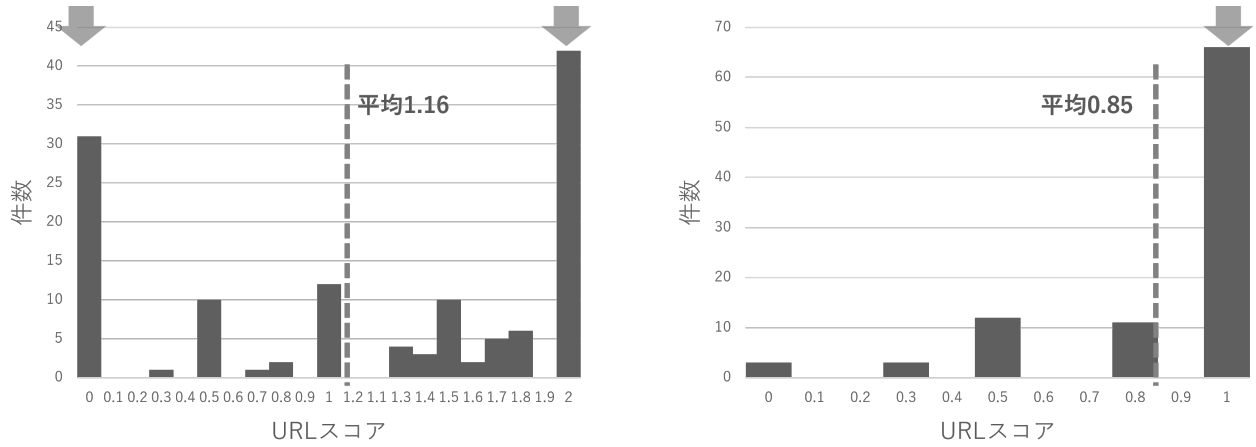
図 2 の象限ごとの傾向は次のとおりである。(1)URL スコアが低く本文スコアが高い領域（左上）は，URL 提示の仕組み自体に改善余地があることを示唆する。(2)URL スコアが高く本文スコアが低い領域（右下）は，マニュアル検索自体は機能している一方で回答文の生成側に課題があることを示唆し，生成プロンプトの調整が有効と考えられる。そして，(3)URL スコアの高低によらず本文スコアが低い領域は，原因を個別に深掘りする必要があり，その詳細を §5.3 で述べる。

なお，回答可能な質問全体に対する近似曲線（$y = -0.0621x^2 + 0.4019x + 0.3049$，$x$：URL スコア，$y$：本文スコア）の傾きはごく緩やかであり，本文スコアと URL スコアの間に明確な相関は見られなかった。この結果は，回答本文と URL をそれぞれ独立した評価軸として設けた前節の設計の妥当性を裏付けるものといえる。

スコアの数値のみでは分布や傾向を把握しづらいため，ヒストグラムおよび散布図を用いて可視化した。

図 3a，図 3b に URL スコアのヒストグラムを示す。回答可能な質問（平均 1.16）・回答不可な質問（平均 0.85）のいずれも，スコアが低い層と高い層に分かれる二極化した分布となった。この二極化傾向は，AI の回答が URL 提示

の面で「十分に対応できるか，ほとんど対応できないか」の両極端に振れやすく，中間的な部分点を取りにくい特性を持つことを示唆している。



(a) 回答可能な質問　　(b) 回答不可な質問

図 3: URL スコアのデータ件数ヒストグラム

また，図 4 に URL の不足度合いと過剰度合いの散布図を示す。弱い正の相関が見られ，URL が不足している場合は過剰にもなりやすい（逆もまた同様）傾向が確認できた。

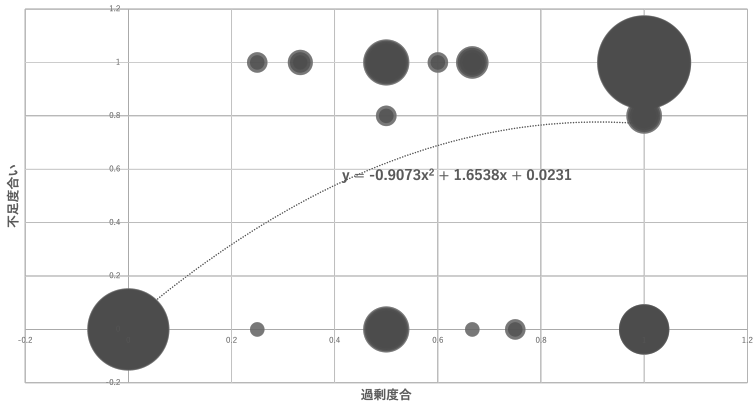


図 4: URL の不足度合いと過剰度合いのバブルチャート

5.3 スコア分析による不得意領域の特定

図 2 で示した本文スコアが低い領域について，さらに深掘り分析を行い，対象機能が固有に持つ不得意領域を特定・言語化した。製品の課題だと判断した事象の中から，特に次の 3 点を取り上げる。

- エラーメッセージを含む質問への回答精度が低い。回答生成時に文章を単語単位に区切って検索する仕組み上，メッセージ内にユニークな単語が少ないと検索がヒットしにくいと考えられる。
- 「できないこと」を尋ねる質問への回答が不得手。マニュアルに「できない」旨の記載が無いと，AI がそれらしい回答を捏造してしまう（ハルシネーション）。
- 「インストール」等，多くのマニュアルに共通して現れる汎用的なキーワードを含む質問は，無関係なマニュアルを引き込みやすく検索精度が落ちる。

これらは具体的な負の事例としてリリース判定会で提示した。定量スコアのみでは伝わりにくいリスクを可視化することで，ビジネスドメインの責任者も品質状態を直感的に把握でき，合意形成が円滑化した。

また，マニュアルごとの参照回数の偏りや，同一質問に対する回答生成のブレについても調査した。前者は質問内容やマニュアルのページ数に影響される面が大きく，良し悪しの深掘りには繋がらなかった。後者は感覚的には大きなブレは無さそうだったが，評価に時間がかかり，評価期間中にインプットデータの更新が頻繁に行われ単純な比較ができず，判断を保留した。

スコア傾向分析によって特定した不得意領域は開発チームへフィードバックし，プロンプト調整やインプットデータ補完に活用された。一方，リリース判定会では定量スコアを提示したものの「合否判定基準」そのものは事前に定義されておらず，最終的な判定はビジネスドメインの責任者の定性判断に委ねられた。

6. 考察・成果

課題 (1)（数値化しても社内に比較対象となる知見がなく，ビジネスドメインの責任者が客観的に判断できない）に対しては，仮説 (1)（ガイドラインのチェック項目を用いた説明可能化）が有効に機能した。ただし，これはガイドラインそのものが品質を保証したという意味ではない。各チェック項目の解釈・影響度・報告先ロールを言語化し，合意形成を重ねる中で気づいたのは，ガイドラインは品質を担保する手段としてではなく，関係者間のコミュニケーションの土台として使うのが最も有効だということだった。これにより，プロジェクトに携わるメンバーが同じ品質基準・共通言語で議論できるようになり，「誰がどのレベルの項目を担当し，誰に報告すべきか」という体制と責任分担が明確になった結果，後工程での手戻りの発生を抑制できた。加えて，一過性のリリース判定に限定せず，製品ライフサイクル全体を通じて品質を検討できる体制となり，今後の改善や他機能との比較のための基盤ができたことも大きな成果である。

課題 (2)（従来の演繹的な検証手法が通用しない）に対しては，仮説 (2)（実際のユーザー問い合わせを評価セットとして帰納的に検証する）は部分的に有効だった。運用データを用いたことで評価セットの現実性は高まり，とりわけスコア低位群を深掘りすることで，エラーメッセージを含む質問・「できないこと」を尋ねる質問・汎用キーワードを含む質問といった不得意領域を特定できたこと，そして負例によるリスク提示が定量スコア以上に伝わりやすかったことは，帰納的検証の有効性を裏付けるものであった。一方で，この評価結果をもって「完璧」と言い切ることはできない。実際，リリース後 1 年間のユーザーフィードバックでは，質問意図の取り違え（OS・デバイス種別の混同等）や会話文脈の引き継ぎ不足に関する指摘が多く，これらは単発の質問応答を前提とした評価セットの設計では検出しづらい種類の課題であった。加えて，他の導出方法があり得た可能性も否定できず，ヘルプチャット以外の AI 製品に展開する際には，スコア評価の設計を都度検討し直す必要がある。

7. 結論

本取り組みの結果，AI 機能のリリース判定基準は定量指標のみでは構成できず，定量・定性・責任者合意の三要素の併用が不可欠であることを示した。AI プロダクト品質保証ガイドラインはチェックリストとしての直接適用ではなく，自社文脈における品質観点の検討基盤として運用することで実効性を持つ。また，開発側の技術評価（Ragas 等）とビジネス側のリリース可否判断の間に QA を配置し，両者を翻訳・接続する位置取りが，AI 機能においても品質保証に有効であることを確認した。

一方で，本取り組みには課題も残った。リリース判定の「合否基準」自体を事前に定量的に定義できなかったこと，評価ルーブリックの基準策定が QA 担当者の判断に依っていたこと，評価が人力に依存し時間と工数を要したことは，いずれも客観性・再現性・スケーラビリティの面で改善の余地がある。今後は，本実践で確立した合意形成フレームワークおよび品質スコア化のプロセスを社内他 AI 機能へ展開するとともに，ルーブリックの基準策定自体を複数人でレビューする体制や，LLM-as-a-Judge 等の自動評価手法を取り入れた評価セット構築の仕組み化を進め，開発速度の維持・向上に貢献する体制の確立を目指す。

参考文献

[1] QA4AI コンソーシアム: ”AI プロダクト品質保証ガイドライン”, 2025.04 版 (2025 年 4 月 10 日公開), https://www.qa4ai.jp/home.