特別研究報告

題目

関連成果物に基づく LLM を用いた要件定義書の完全性レビュー手法の提案

指導教員

楠本 真二

報告者

橋本 明樹

令和 8 年 2 月 9 日

大阪大学基礎工学部情報科学科

令和 7 年度 特別研究報告

関連成果物に基づく LLM を用いた要件定義書の完全性レビュー手法の提案

橋本 明樹

内容梗概

ソフトウェア開発において要件定義書は，上流工程の最終成果物として以降の設計，実装，テスト工程に大きな影響を与えるため，その品質確保は重要な課題である．特に，要件定義書に必要な要件が漏れなく記載されているかを確認する完全性レビューは不可欠であるが，実務においては，関連成果物が多岐にわたり，人手でそれらを網羅的に参照することが困難であるという問題がある．その結果，要件決定の根拠となる議事録等の成果物との整合性が十分に確認されないまま，要件定義書が作成・確定されるケースも少なくない．

本研究では，関連成果物との整合性に着目した要件定義書の完全性レビューを支援する枠組みを提案する．提案枠組みでは，要件定義書作成の元となる議事録等の関連成果物から要件に関する情報を統合した比較用の参照モデル（統合参照モデル）を構築し，これと要件定義書を対応付けることで，要件の欠落や根拠不明な要件を抽出する．統合参照モデルは正解となる要件定義書を生成することを目的とするものではなく，あくまで完全性レビューにおける比較対象として位置づけられる．

さらに，提案枠組みを現実的に実行可能とするための実装方法として，大規模言語モデル（LLM）を用いた実験を行う．実際のソフトウェア開発プロジェクトにおいて作成された要件定義書および議事録を対象とし，議事録から統合参照モデルを生成した上で，要件定義書との比較を行った．その結果，要件の欠落や要件決定過程の未記録といった問題を一定程度検出できることを確認した．

本研究の結果は，関連成果物との整合性に基づく完全性レビューが，要件定義書作成プロセスにおける見落としの防止やレビュー効率の向上に寄与する可能性を示している．

主な用語

要件定義レビュー，完全性，関連成果物との整合性，大規模言語モデル，実験的評価

目次

図目次

表目次

1　はじめに

要件定義書 (Software Requirements Specification) は，ある動作環境のもとで所定の機能を実行するソフトウェア製品（またはその集合）について記述した仕様書である [1]．要件定義書の作成は，要求の収集，分析・整理，文書化，および関係者による複数の会議と合意形成という一連のプロセスを通じて行われる．このステップの一つに要件定義書上の記載不足や曖昧性等の不具合の早期検出を目的とした要件定義レビューという工程がある．要件定義書の品質が開発対象のソフトウェアの品質やプロジェクト自体の成否にも影響する [2, 3] ため，要件定義レビューはソフトウェア開発において必要不可欠な工程である．

要件定義レビューでは様々な観点から要件定義書の内容が適切であるかの確認が行われる．要求工学プロセスやその成果物の要件定義書の国際標準規格である ISO/IEC/IEEE 29148 [1] には要件定義書内の要件全体に求められている特性が明記されており，これらは要件定義レビューにおける最も基本的な観点となる．ISO/IEC/IEEE 29148 に記載されている観点の具体例として，要件定義書内に必要な情報が過不足なく含まれているのかという完全性 (Completeness) や要件間で矛盾が発生していないかという一貫性 (Consistency)，制約内で実現可能であるのかという実現可能性 (Feasibility) 等が挙げられる．

一方で，実際の開発プロジェクトで要件定義レビューを行う上で，多くの課題が指摘されている．一つの課題は，要件定義レビューを行うにあたり，多くの時間的コストを必要とする点である．要件定義レビューは時間や予算の制約のもとで行われるため，時間的コストの課題によりレビュー自体の品質を下げる可能性があると指摘されている [4]．この時間的コストの問題の原因の一つとして，要件定義書内の内容の妥当性を要件定義書作成の元になった成果物（以下，関連成果物）との整合性から確認する作業の難しさがある [5]．

これらの課題を解決するために，要件定義レビューの支援を目指した研究が行われてきている．特に近年は自然言語処理タスクで高い貢献を示す大規模言語モデル（LLM：Large Language Model）を適用した研究も行われつつある．例えば，Krishna らの研究 [6] では，LLM を用いて，要件定義書を独立した成果物として扱い，その内部品質（曖昧性や一貫性等）を評価している．しかし，関連成果物との整合性に注目した完全性についてレビュー研究は十分行われていない [7]．

そこで本研究では，関連成果物との整合性に基づいた要件定義書内の要件の完全性レビューの枠組みについて提案する．提案する枠組みは，要件定義書を独立した成果物として評価するのではなく，その作成の元となった議事録等の関連成果物において合意・検討された内容が，要件として漏れなく要件定義書に反映されているかという観点から完全性を捉える．具体的には，関連成果物に含まれる要件に関する情報を統合した参照モデル（以下，統合参照モデル）を構築し，これと要件定義書を対応付けるこ

とで，要件の欠落や根拠が不明確な要件を抽出する．本枠組みは，要件定義書を生成することを目的とするものではなく，完全性レビューにおける比較・確認を支援する点に特徴がある．

更に，提案する枠組みを LLM を用いて実装し，実際のソフトウェア開発プロジェクトで作成された文書（要件定義書，関連成果物）に適用した．統合参照モデルと要件定義書の比較を通じて，完全性レビューの観点から問題となり得る差異を抽出できることを確認し，実務への適用可能性を示した．

以降，2. では，要件定義に関連した用語，要件定義レビュー，関連研究について述べる．3. では，関連成果物との整合性に基づく要件定義書の完全性レビュー支援の枠組みを提案する．4. では，提案する枠組みを LLM を用いて実装し，実プロジェクトデータに適用した実験について説明する．5. では，実験結果に基づく分析を行う．最後に，6. で，本研究のまとめと今後の課題について述べる．

2　準備

2.1　要件定義

要件定義とはソフトウェア開発プロセスの上流工程の一つの過程である．ソフトウェア開発プロセス全体の枠組みの国際標準規格である ISO/IEC/IEEE 12207[8] には要件定義の目的はステークホルダーが必要としている機能をステークホルダーのニーズに沿ったソフトウェアに技術的な視点で要件として落とし込むことと記述されている．更に，要件定義は，要件の定義，要件の分析，要件の管理という複数のタスクに分解されると説明されている．

まず最初に，要件の定義が実施される．具体的には，必要な機能を要件化してその属性や制約の定義が行われる．次に，要件の分析が実施される．このタスクでは定義された要件の内容を分析し，様々な観点からの妥当性のレビューがなされる．そして最後に，要件の管理というタスクが行われる．ここでは定義された要件に対しての合意形成や要件のトレーサビリティの維持等が行われる．このようなタスクを通じて要件定義が実施される．

要件定義には様々な課題が指摘されている．Méndez Fernández らは，要件定義の課題を含め上流工程で立ち現れやすい問題を報告している [9]．この論文では，実際の現場における問題を以下のようにまとめている．

1. 要件の曖昧性・不明確さ
2. 要件の不完全性
3. ステークホルダ間のコミュニケーション不足
4. 要件変更の管理の難しさ
5. 要件レビューの困難さ
6. 要件と他成果物とのトレーサビリティ不足

すなわち，要件定義における主要な問題は，曖昧で不完全な要件記述，ステークホルダ間の合意不足，および要件と関連成果物とのトレーサビリティの欠如に起因しており，これらは依然として多くの実プロジェクトで解決されていないことが示されている．

特に，本来必要な要件が欠落している，または明文化されていないことを指す要件の不完全性という問題は最も頻出の問題であると指摘している．

2.2　要件定義レビュー

要件定義レビューとは要件定義プロセスに含まれる一つの工程である．具体的には前節で紹介した要件の分析というタスクに該当する．要件定義レビューでは要件定義書内の記載不足や曖昧性等の不具合

の早期検出を目的として要件定義書の内容の妥当性を様々な観点でレビューを行う．

要件定義書はソフトウェア開発の根幹となる文書であるため，要件定義レビューは必要不可欠な工程である．一般的に要件定義レビューを行う際の観点として要求工学プロセスやその成果物の要件定義書の国際標準規格である ISO/IEC/IEEE 29148 に記載されている要件定義書内の要件に求められる品質特性が挙げられる．以下では ISO/IEC/IEEE 29148 に記載されている中でも要件定義書内の要件全体に求められる品質特性をまとめる．

要件全体に求められる品質特性（ISO/IEC/IEEE 29148 に基づく）[1]

完全性 追加情報なしで必要な項目が十分記載されていること
一貫性 要件間で矛盾や重複がなく単位や用語の用法が統一されていること
実現性 要件集合が制約の中で実現可能であること
理解性 システムに期待することやシステムとの関係が明確に記述されていること
検証性 要件集合の実現が制約内でのニーズの達成につながると確認できること

要件定義レビューについて Méndez Fernández らは，完全性の観点からのレビューの不足が原因となって要件定義の頻出問題である要件の不完全性が発生していると指摘している [9]．完全性の観点からのレビューでは要件定義書内の情報の過不足を確認するため，要件定義書の元になった関連成果物の参照がなされる．その一方で，関連成果物が膨大であるため参照することがコスト的に現実的でないという指摘 [5] がなされており，それが一つの原因となって完全性の観点からのレビューの不足につながっていると考えられる．

2.3 大規模言語モデル（LLM：Large Language Model）

大規模言語モデル（LLM：Large Language Model）は大量のテキストデータと高度なディープラーニング技術を用いて構築された言語モデルの一種である．代表的な LLM として，OpenAI 社が開発した GPT や，Google 社が開発した Gemini 等がある．LLM は自然言語処理タスクを高精度に行うことができる特徴がある．特に文書の要約や文書からの情報の抽出において LLM は，数少ない例示や説明文だけで高い質の貢献ができることや専門的な知識を要する文献に対しても有効であることが報告されている [10, 11]．また，大量の文献からの文献の選定を行う作業や文書の要約において LLM は大幅な時間的コストの削減に貢献することを示唆した研究もいくつか存在する [12, 13]．

LLM はソフトウェア工学分野でも多く適用されている．その適用分野はリファクタリング [14] やコード要約 [15] のソフトウェア開発の下流工程だけではなく，要件の抽出や要件の分析等 [16] のソフトウェア開発の上流工程にも及んでいる．そして，上流工程の中でも要件定義レビューのプロセスに LLM を適用している研究もいくつか見られる．具体的には，要件定義書内の複数の要件間の内容に矛

盾がないかの検出を LLM を用いて目指した研究 [17, 18] や要件定義書内の要件の記述に曖昧性がないかどうかの検出に LLM を適用している研究 [6] も存在する．

2.4 要件トレーサビリティ

要件トレーサビリティ（Requirements Traceability，以降 RT）とは，要件をその根拠となる情報源から設計・実装・テストなどの成果物まで，あるいはその逆方向で追跡できる状態を指す [19]．さらに，RT は要件が要件定義書に含まれる前の pre-RS トレーサビリティと要件が要件定義書に含まれた後の post-RS トレーサビリティの大きく二つに分けられる [19]．この RT を理解することはソフトウェア開発の様々な場面において重要であるため，RT に関する様々な研究が行われてきている．しかし，その研究の多くは post-RS トレーサビリティに焦点を当てており，pre-RS トレーサビリティに焦点を当てた研究は少なく新たな研究の必要性が指摘されている．[7]

3　提案する枠組み

3.1　本研究の目的と位置付け

本研究は，先行研究においてその不足が問題として指摘されている要件定義書の完全性レビューの支援を目的とする [9]．そこで本章では，まず，関連成果物に基づく要件定義書の完全性レビューの枠組みを提案する．続いて，本枠組みを実行可能とするための具体的な実装方法について述べる．

本研究で扱う完全性レビューは，要件が要件定義書に必要十分に記載されているかという観点に限定する．この観点に基づき，本枠組みは，機能要件，非機能要件，制約など，関連成果物から抽出可能な要件記述全般を対象とすることができる．ただし，本章では枠組みの一般形を示すにとどめ，後続章における実験では，評価対象を機能要件に限定する．

3.2　枠組みの構成

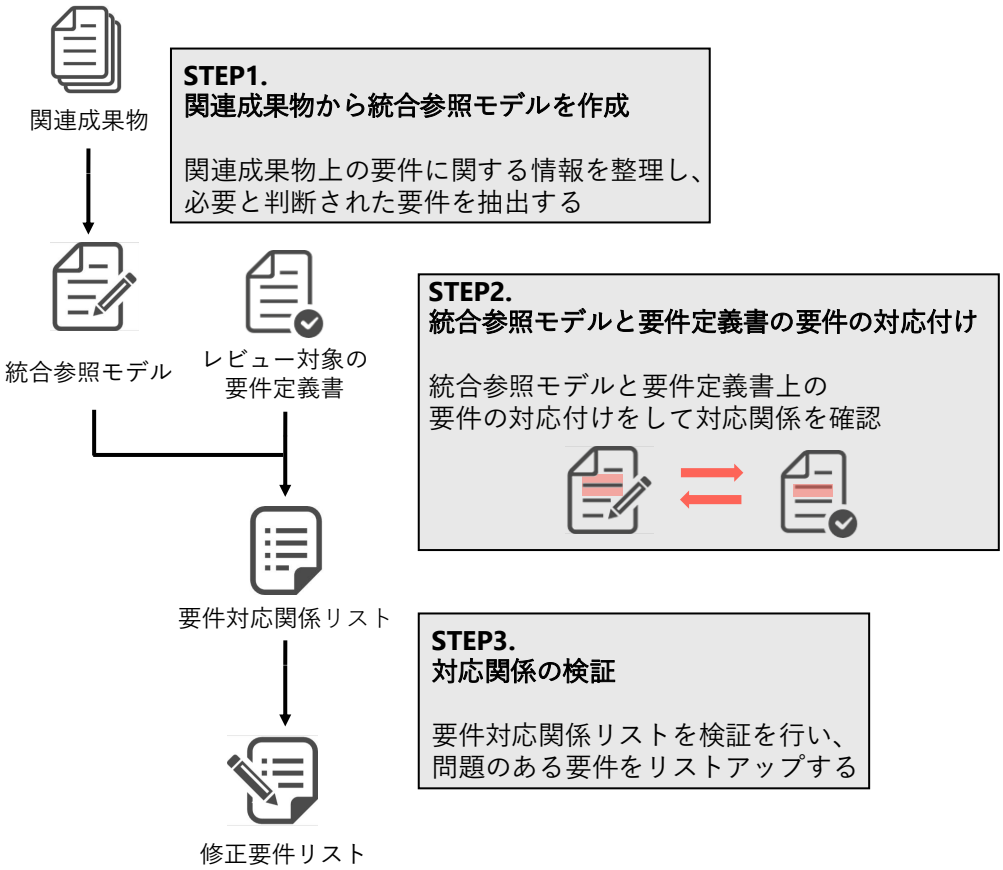

図 1　提案する枠組みの流れ

図 1 に，本研究が提案する要件定義書の完全性レビューの枠組みの流れを示す．提案する枠組みは，関連成果物を用いた完全性レビュー手法であり，要件定義書と関連成果物を入力として利用する．本枠組みは，次の三つのステップから構成される．

第一のステップ（STEP1）は，関連成果物から統合参照モデルの作成である．このステップは，完全性レビューを行うための準備段階に相当する．具体的には，関連成果物に含まれる情報を整理し，その結果として必要と判断された要件を記載した統合参照モデルを作成する．

第二のステップ（STEP2）は，統合参照モデルと要件定義書に記載された要件との対応付けである．このステップでは，前ステップで得られた統合参照モデル上の要件と，要件定義書上の要件との対応関係を整理し，その結果として要件対応関係リストを作成する．この要件対応関係リストを参照することで，完全性の観点から問題となり得る要件を把握することができる．

第三のステップ（STEP3）では，STEP2 で得た要件の対応関係の検証を行う．このステップでは，要件対応関係リスト上で確認できる完全性の観点から問題となり得る要件が実際に問題であるかを検証する．その結果として，完全性の観点で修正が必要な要件を整理した修正要件リストを得る．

以上の 3 ステップを通じて，本枠組みは，関連成果物を用いた要件定義書の完全性レビューを実現する．

3.2.1 STEP1: 関連成果物からの統合参照モデルの作成

本枠組みでは，関連成果物一つ一つと要件定義書を直接比較するのではなく，関連成果物の情報を集約した統合参照モデルと要件定義書を比較することで，要件定義書の完全性レビューを行う．

関連成果物の情報を一度整理して統合する工程を設ける理由は，関連成果物間に存在する複雑な依存関係にある．要件定義のプロセスにおいて議論・検討された内容は，通常，時系列に沿って複数の関連成果物に分散して記録される．このため，各関連成果物は，それ以前またはそれ以降に作成された他の関連成果物の内容を前提としており，関連成果物間には複雑な依存関係が生じる．従って，要件定義プロセスにおける決定事項は，単一の関連成果物を参照するだけでは把握できず，全ての関連成果物を一つの文脈として捉え，情報を整理することで初めて確認可能となる．この観点から，本枠組みでは最初のステップとして，関連成果物上の情報の整合性に基づき，必要な要件の抽出を行う．

次に，STEP1 における具体的な処理内容を説明する．まず，各関連成果物を参照し，その中からレビュー対象の要件に関する情報を抽出する．全ての関連成果物から情報を抽出した後，それらを時系列順に整理する．具体的には，どの要件が採用され，どの要件が不採用となったのかといった情報を確認し，各要件に対してどのような決定がなされたのかを追跡する．その結果，要件定義書に記載されるべきであると判断された要件をリストアップし，それらを記載した統合参照モデルを本ステップの成果物として得る．この統合参照モデルは，STEP2 において要件定義書の完全性レビューを行うための比較

用参照モデルとして利用する．なお，統合参照モデルは，真に必要な要件のみを記載した最終的な要件定義書そのものではない点に留意する必要がある．

3.2.2 STEP2: 要件定義書と統合参照モデルの対応付け

要件定義書の完全性レビューでは，要件定義書に必要であるにもかかわらず欠落している要件（以下，欠落要件）と，要件定義書に不必要であるにもかかわらず記載されている要件（以下，根拠不明要件）を検出する．これらを検出するために，STEP2 では，要件定義書に記載された要件と，STEP1 で作成した統合参照モデル上の要件との対応付けを行う．

欠落要件を検出するため，まず，統合参照モデル上の各要件について，それに対応する要件が要件定義書に存在するかを確認する．統合参照モデル上の要件は，関連成果物に基づき，要件として必要であると判断された根拠が確認されている．したがって，統合参照モデル上には存在するが，要件定義書上には存在しない要件は，欠落要件となる可能性がある．

次に，根拠不明要件を検出するため，要件定義書上の各要件について，それに対応する要件が統合参照モデル上に存在するかを確認する．関連成果物に基づいて要件として必要であると判断された全ての要件は，統合参照モデルに記載されている．そのため，要件定義書上には存在するが，統合参照モデル上には存在しない要件は，根拠不明要件となる可能性がある．

以上のように，双方向から要件の対応付けを行い，その結果を要件対応関係リストとして整理する．この要件対応関係リストを参照することで，欠落要件となる可能性のある要件（以下，欠落要件候補）および根拠不明要件となる可能性のある要件（以下，根拠不明要件候補）を確認することができる．

3.2.3 STEP3: 対応関係の検証

STEP3 では，これまでの工程でリストアップされた欠落要件候補および根拠不明要件候補について，人手による検証を行う．具体的な検証方法は，以下のとおりである．

まず，確認者は，欠落要件候補および根拠不明要件候補に関して，関連成果物および要件定義書の該当箇所を改めて参照する．その際，根拠不明要件候補として挙げられている要件について，要件として必要であることを示す根拠が関連成果物中に見つからなかった場合，当該要件を根拠不明要件と判定する．また，欠落要件候補として挙げられている要件について，要件として必要であることを示す根拠が関連成果物中に存在し，かつ，要件定義書中に該当する記述が確認できなかった場合，当該要件を欠落要件と判定する．

STEP3 を実施する人手としては，企業におけるソフトウェア開発プロジェクトにおいて要件定義を担当するシステムエンジニア，業務部門の担当者，あるいは品質保証部門のレビュー担当者など，要件の妥当性およびその背景を理解している関係者が適任である．

3.3 枠組みの実行可能性と実装方針

提案する枠組みを用いることで，関連成果物と要件定義書を網羅的に照合し，要件の完全性レビューを理論的に実施することが可能となる．しかし，関連成果物の膨大な参照コストが指摘されているように [5]，本枠組みをそのまま実装することは困難であると考えられる．そのため，関連成果物の参照に伴う高いコストを低減しつつ，本枠組みを現実的に実装する方法が必要となる．本研究では，その実装方法の一つとして，大規模言語モデル（LLM）の適用を考える．

LLM は，前章で述べたとおり，文書からの情報抽出や要約などのタスクにおいて高い性能を示しており，時間的コストの削減への貢献も報告されている [10, 11, 12, 13]．この特性を踏まえ，本研究では，まず枠組みの STEP1 に LLM を適用することで，関連成果物に含まれる情報を整理するタスクのコスト削減を図る．具体的には，LLM に関連成果物を入力として与え，その内容から要件に関する情報を抽出・整理し，要件定義書に必要な要件をリストアップするよう指示する．その結果，必要な要件が整理された統合参照モデルを，LLM の出力として得る．4 章の適用実験では，このような STEP1 への LLM の適用を行っている．

さらに，STEP1 に加えて STEP2 に対しても LLM を適用も考慮する．STEP2 では，統合参照モデル上の要件と要件定義書上の要件との対応付けを行うため，その実施コストはプロジェクトで扱う要件数に応じて増大する．実際に，数千から数万規模の要件を扱うプロジェクトが報告されており [20]，このような場合には，STEP2 の実施コストが非現実的なものとなる可能性がある．この課題に対し，STEP2 においても LLM を適用することでコスト低減を図る．具体的には，LLM に統合参照モデルと要件定義書を入力として与え，両文書間における要件の対応付けを指示する．その結果として，統合参照モデルと要件定義書間の要件の対応関係を記載した要件対応関係リストを，LLM の出力として得る．

以上のように，本研究は，提案する枠組みに LLM を適用することで，理論的に有効である一方で実装が困難であった完全性レビューの枠組みを，現実的に実装可能とする手法を提案する．

3.4 前提条件

本研究は，関連成果物を利用した完全性レビューの現実的な実装を提案するものであり，関連成果物がアクセス可能な状態で存在していることを前提条件としている．関連成果物は多種多様であり，それらを準備・収集すること自体の難しさが指摘されているが [21]，本研究は関連成果物の準備を目的としたアプローチではない点に留意する必要がある．一方で，要求に関連する成果物は，ステークホルダ要求を伝達するための基盤であり [22]，それらに基づいて開発者がソフトウェアを実装し，テスタがテストを実施することは，ソフトウェア開発において一般的である．従って，関連成果物がアクセス可能な形で存在しているという本研究の前提条件は，妥当なものであると考えられる．

また，本枠組みの実装を通じて適切なレビュー結果を安定して得るためには，アクセス可能な関連成果物上に，必要な情報が全て記載されているという条件も重要となる．例えば，ある要件について不採用とする決定がなされたにもかかわらず，その記録が関連成果物上に存在せず，採用された要件として扱われている場合を考える．このような状況では，STEP1 において，当該要件は要件定義書に必要な要件であると判断される．しかし，当該要件は要件定義書には記載されていないため，STEP2 の結果，欠落要件となる可能性のある要件として扱われる．

最終的な STEP3 における人手による検証により，本来は問題のない要件であるにもかかわらず，問題がある可能性が指摘された要件を除外することは可能である．しかし，このような要件が多数発生した場合，レビュー作業における負荷の増大やヒューマンエラーの原因となり得る．そのため，アクセス可能な関連成果物上に必要な情報が十分に記載されていることは，本枠組みを適切に運用する上で重要な前提条件である．

4　枠組みの適用実験

本研究では，三章で提案した枠組みを実際のソフトウェア開発プロジェクトで扱われたデータセットに適用する実験を行う．枠組みの実装方法としては，同じく第 3 章で提案した STEP1 および STEP2 に，LLM を用いた．図 2 に，本実験の流れを示す．本実験は LLM を用いて実装した枠組みの実務への適用可能性の評価を目的としている．この適用可能性の評価にあたっては，LLM の出力が確率的である点を考慮する必要がある．そのため，本実験では，同一のデータセット，同一の LLM，および同一のプロンプトを用いて，図 2 に示した処理を全 5 回繰り返して実施する．

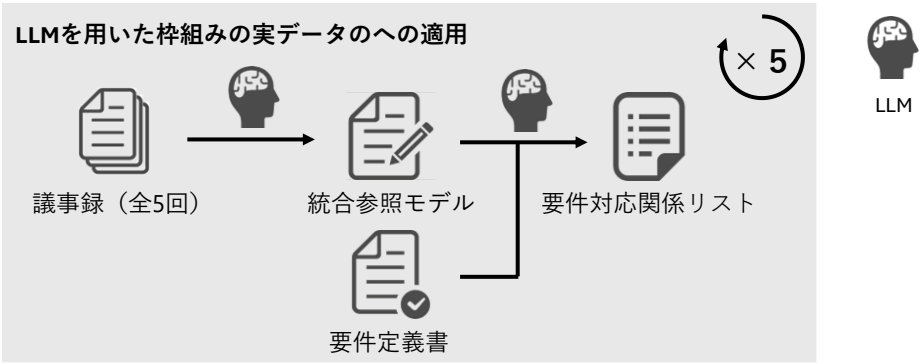

図 2　本実験の流れ

本章では，以下の内容について説明する．まず，4.1 節では，本実験に用いたデータセットおよび LLM について説明する．次に，4.2 節では，本実験において行ったデータ形式の変換について，具体例を交えて説明する．続いて，4.3 節では，LLM を用いて STEP1 および STEP2 をどのように実装したかについて詳述する．最後に，4.4 節では，本実験が目的とする実務への適用可能性の評価方法について説明する．

4.1　対象データと LLM

本実験では，実プロジェクトにおいて実際に用いられた要件定義書を，レビュー対象の要件定義書として使用する．対象となるプロジェクトは，ある業務サポートシステムのバージョンアップを目的としたものである．当該プロジェクトでは，本実験のレビュー対象となる要件定義書の作成にあたり，要件定義プロセスとして全 5 回の会議が実施された．これらの会議では，プロジェクトのスケジュールや機能要件に関する議論が行われており，各会議ごとにその内容をまとめた議事録が作成されていた．そのため，本実験では，これら全 5 回分の会議議事録を関連成果物として採用している．

また，対象となる要件定義書には，9 個の機能カテゴリに分類された計 29 個の機能要件が記載され

ている．本実験では，これらの機能要件に着目し，機能要件に対する完全性レビューとして，提案枠組みを実データに適用する．なお，本実験で用いたデータは企業から提供されたものであるため，その詳細な内容については記載できない．

次に，本実験で使用する LLM について述べる．本実験では，LLM として OpenAI 社の gpt-5 を API を通じて利用する．本モデルは，実験設計時点において OpenAI 社が提供する最新のモデルであり，長大なコンテキストウィンドウを有している．この特性により，要件定義書や議事録などの長文ドキュメントを入力データとして扱うことが可能である．

LLM を利用する際には，温度パラメーターを通じて出力のばらつきを制御することが一般的である．温度パラメーターとは，LLM が次に出力するトークンを選ぶ際，そのランダム性を制御するための設定値である．温度パラメーターを 0 に近づけると，一般的に LLM の出力のばらつきが抑えられ，1 前後あるいは 1 以上に設定すると，LLM の出力のばらつきが大きくなると一般的にされている．本実験では，LLM の出力のばらつきが少ない方が望ましいため，温度パラメーターは 0 に近づけるのが通常のアプローチである．しかし，gpt-5 は実験時点において利用可能な温度パラメーターが 1 のみであるため，本実験では温度パラメーターを 1 に設定している．

4.2　データの前処理と出力形式

本実験では，LLM に入力されるデータおよび LLM が出力するデータの一部について，その形式を Markdown 記法に固定している．そのため，議事録および要件定義書は，元のデータ形式から，情報の損失が生じないよう手動で Markdown 記法へ変換を行っている．また，出力データである統合参照モデルを Markdown 記法で得るため，LLM に対して，出力してほしい統合参照モデルの形式をプロンプト上で例示している．図 3 には，要件定義書および議事録の変換例と，統合参照モデルの形式例を示す．

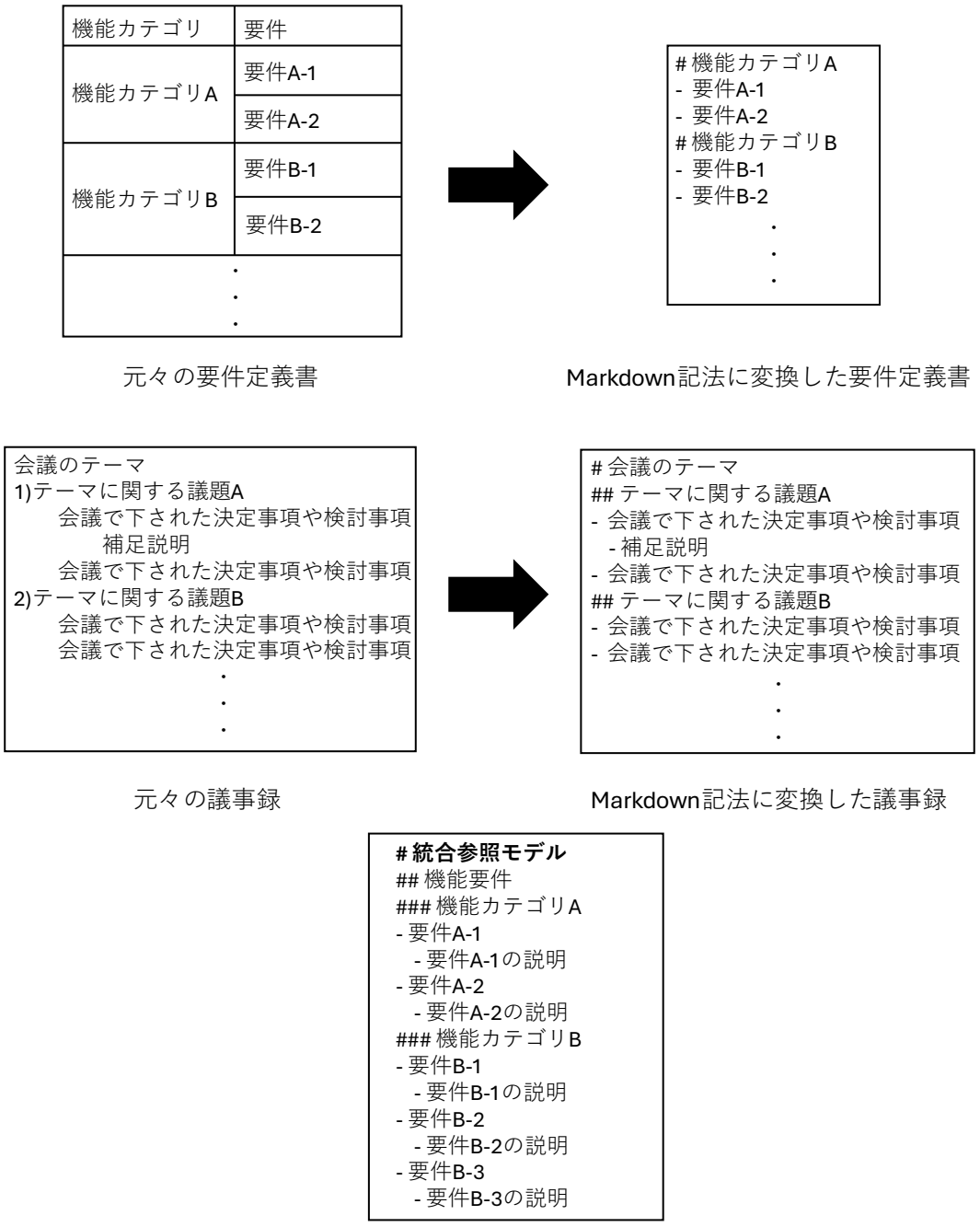

図 3　データの各種例

このようにデータ形式を固定することで，確率的な性質をもつ LLM の出力のばらつきを抑制することを目指している．本論文では詳細には扱わないが，出力のばらつきの抑制効果を確認するため，事前に準備実験を実施した．この準備実験では，データ形式を固定せずに LLM に入力した場合と，

Markdown 記法に変換して LLM に入力した場合とを比較した．その結果，STEP1 の成果物である統合参照モデルにおいて，Markdown 記法の方が，出力のばらつきが大きく抑制されることを確認した．以上の準備実験の結果を踏まえ，本実験ではデータ形式の固定化を行っている．

4.3 LLM を用いた STEP1,2 の実装方法

本節では，本実験において STEP1 および STEP2 を，LLM を用いてどのように実装したかについて説明する．

4.3.1 STEP1 の LLM を用いた実装方法

STEP1 では，関連成果物から要件に関する情報を抽出・整理し，要件定義書に必要な要件をリストアップするタスクを LLM に与える．ただし，本実験では，要件のうち機能要件に限定して検討する．そのため，STEP1 において LLM は，関連成果物に基づき，要件定義書に必要な機能要件をリストアップするタスクを実行する．本タスクを実現するために，本実験において LLM が実施する処理の流れを，図 4 に示す．

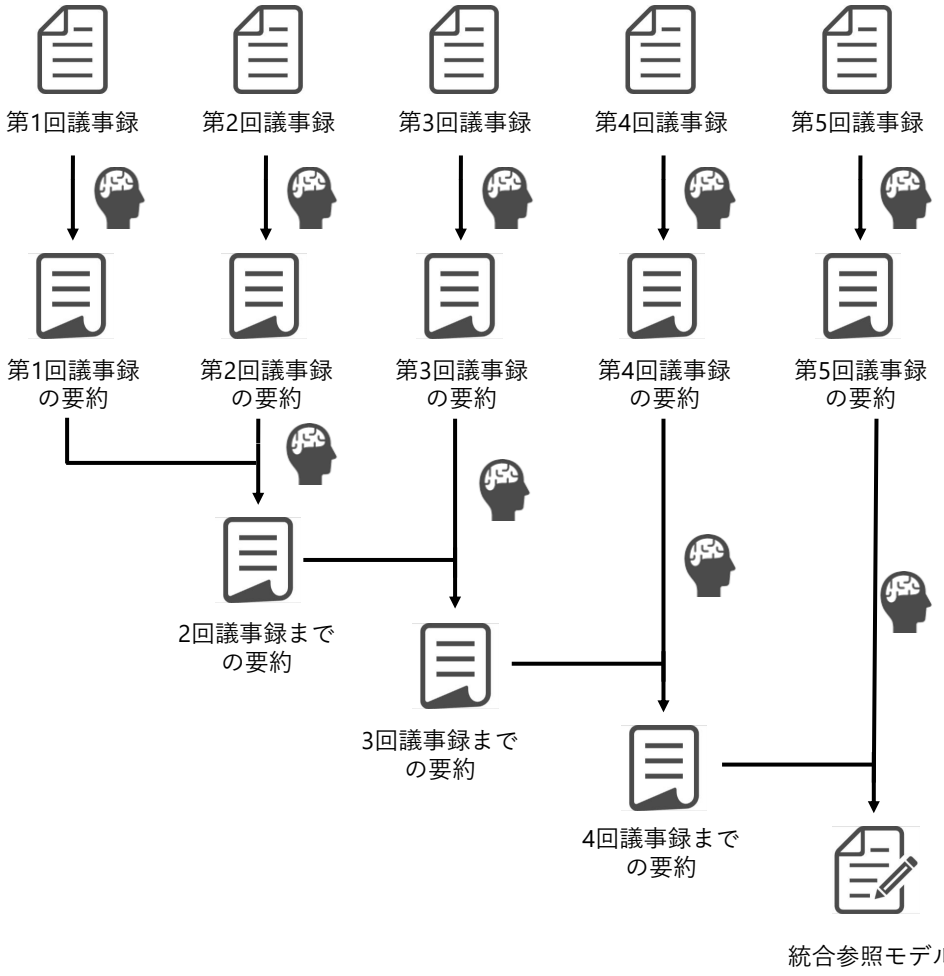

図 4　LLM を用いた STEP1 の流れ

先ず，LLM によって，各議事録に含まれる機能要件に関する情報の要約（以下，要約とする）を作成する．この際，LLM には 1 回分の議事録を入力として与え，当該議事録から機能要件に関する情報を抽出するよう指示したプロンプトを与える．この入力に対して，LLM からは，1 回分の議事録に対応する要約が出力される．本実験では，この処理を全ての議事録（5 回分）に対して行い，各議事録の要約を得る．

次に，LLM を用いて，これらの議事録要約の統合を行う．要約の統合は，一度に全ての要約をまとめて行うのではなく，時系列順に 2 つずつ段階的に統合する．具体的には，まず，第 1 回議事録の要約と第 2 回議事録の要約を LLM に与え，これら 2 つの情報に基づいて，要件定義書に必要であると考えられる機能要件をリストアップするよう指示する．この統合により，第 1 回および第 2 回議事録の内容を踏まえて必要であると考えられる機能要件集合（図 4 における「2 回分の議事録までの要約」に相当）が得られる．

続いて，この機能要件集合と第 3 回議事録の要約を LLM に与え，同様の指示を行う．この統合により，第 1 回から第 3 回までの議事録の内容を踏まえて必要であると考えられる機能要件集合（図 4 における「3 回分の議事録までの要約」に相当）が得られる．同様の処理を，全ての議事録の要約が統合されるまで繰り返すことで，全 5 回の議事録の内容に基づき，要件定義書に必要であると考えられる機能要件集合を記載した統合参照モデルを得ることができる．

4.3.2 STEP2 の LLM を用いた実装方法

STEP2 を LLM を用いて実装するにあたり，LLM には，要件定義書に記載された機能要件と統合参照モデルに記載された機能要件との対応関係を判定するタスクを与える．そのため，LLM には，STEP1 で作成した統合参照モデルおよび要件定義書に加えて，対応関係の評価方法と，それを実施するための指示を与える．この際，LLM は，ある機能要件に対して，対応する機能要件が他方の文書上に存在するかを確認し，その結果に基づいて，以下に示す三段階評価を行う．

表 1　三段階評価とその意味

◯	対応する要件が他方の文書で確認された
△	対応する可能性がある要件が他方の文書で確認された
×	対応する要件が他方の文書で確認されなかった

この三段階評価に基づき，LLM は統合参照モデルと要件定義書に記載された機能要件との対応関係を確認する．具体的には，先ず，統合参照モデル上の各機能要件について，それに対応する機能要件が要件定義書に存在するかを検証する．次に，統合参照モデル上の全ての機能要件に対する検証が終了した後，要件定義書上の各機能要件について，それに対応する機能要件が統合参照モデルに存在するかを検証する．

以上の双方向の検証を通じて得られた機能要件の対応関係を，LLM は要件対応関係リストとして出力する．LLM による STEP2 の実行例を図 5 に示す．図 5 において，赤枠で囲まれた部分は，統合参照モデル上の機能要件に対応する機能要件が要件定義書に存在するかを検証した結果を示している．また，青枠で囲まれた部分は，その逆方向からの検証結果を示している．したがって，図 5 の例では，要件 B-3 が欠落要件候補となり，要件 A-3 が根拠不明要件候補となる．

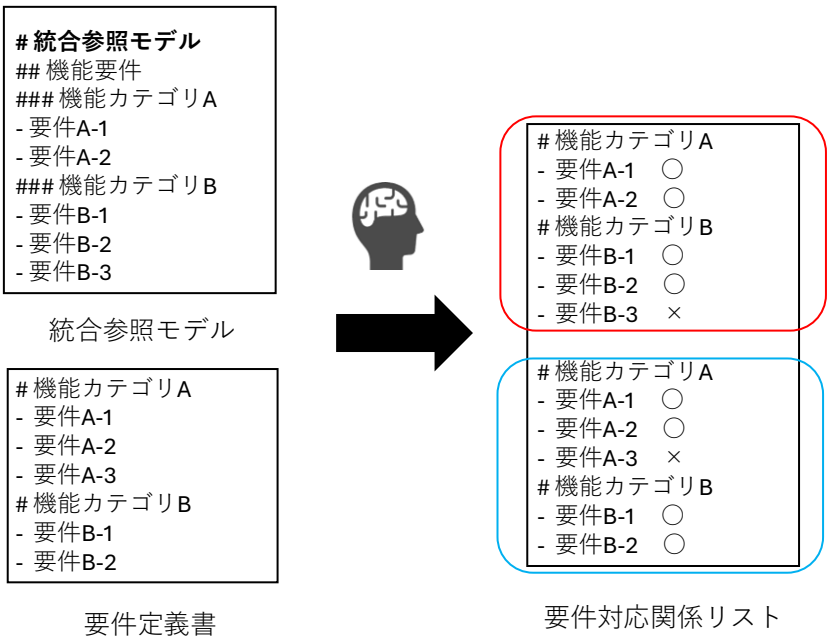

図 5　STEP2 の実行例

4.4　本実験の評価概要

本実験では，LLM を用いて実装した枠組みの実務への適用可能性を評価するため，以下の二つの評価を行う．

一つ目は，枠組みの実装方法としての LLM の有用性の評価である．この評価は，本実験において LLM から得られた統合参照モデルおよび要件対応関係リストを検証することによって行う．

二つ目は，本実験で得られた欠落要件候補および根拠不明要件候補の妥当性の評価である．この評価では，本実験で抽出された欠落要件候補および根拠不明要件候補が，実プロジェクトにおいて実際に問題となった要件であったかを検証する．

以降では，これら二つの評価をどのように実施したかについて説明する．

4.4.1　LLM が生成した統合参照モデルの検証方法

本実験では，各試行において LLM が生成した統合参照モデルの内容について，妥当性および安定性の検証を行う．妥当性の検証では，LLM が生成した統合参照モデル上の各要件について，議事録中に当該要件を要件として正当化する根拠が存在するかを確認する．この検証は，全 5 個の統合参照モデルを対象とし，人手により議事録の該当箇所を確認することで実施する．

一方，安定性の検証では，5 個の統合参照モデルの内容を相互に比較し，生成結果に差異が生じていないかを確認する．以上の二つの検証を通じて，本実験では，STEP1 における LLM の有用性を評価

する．

4.4.2　LLM が生成した要件対応関係リストの検証方法

要件対応関係リストについても，その妥当性および安定性の検証を行う．要件対応関係リストは，最終確認を行うレビュー実施者に対して，欠落要件候補および根拠不明要件候補を提示することを目的としている．そのため，要件対応関係リストには，実際に欠落要件候補または根拠不明要件候補となり得る要件を，できるだけ漏れなく含めておく必要がある．

この観点から，本実験では，LLM が検出した欠落要件候補および根拠不明要件候補の適合率よりも，再現率を重要な評価指標として，要件対応関係リストの妥当性を評価する．妥当性の評価においては，LLM が三段階で評価した要件間の対応関係について，人手により，統合参照モデルおよび要件定義書を参照して確認を行う．本実験では，対応関係が明確に確認できなかったことを示す「△」または「×」の評価を受けた要件を，欠落要件候補または根拠不明要件候補として扱う．そのため，再現率の算出にあたっては，「△」または「×」を Positive，「◯」を Negative と定義する．

一方，安定性の検証は，全 5 個の要件対応関係リストについて，再現率を相互に比較することで行う．以上の二つの検証を通じて，本実験では，STEP2 における LLM の有用性を評価する．

4.4.3　欠落要件候補と根拠不明要件候補の検証

欠落要件候補と根拠不明要件候補として本実験で検出された要件が実際にプロジェクト上で問題になったのかを確認する．この検証を通して，提案する枠組みを実データに適用する際に問題点がないかを評価する．評価にあたっては，当該プロジェクトデータを提供いただいた企業のエンジニアに依頼する．

5　分析

5.1　LLM が生成した統合参照モデルの検証結果

先ず，全 5 個の統合参照モデルに記載された要件が適切であるかを，議事録を参照することで検証した．その結果，統合参照モデル上に記載された要件の大部分について，議事録中に要件としての根拠が確認された．一方で，統合参照モデルには，本来記載されるべきではない要件が一部含まれていることも確認された．これらの要件は，議事録において，新規開発では扱わないことが示唆されていたものである．一見すると，このような事例は，統合参照モデルを作成する上での LLM の精度を損なうものと考えられる．しかし，これらの要件が記載された統合参照モデルを詳細に確認すると，当該要件について新規開発では扱わないことを示す説明が併記されていた．このことから，LLM は，当該要件が新規開発の対象外であることを認識した上で，統合参照モデルを生成していると考えられる．

従って，このような要件を統合参照モデル上でどのように扱うかを，プロンプトにおいてより明確に指定することで，出力を適切に制御できると考えられる．以上を踏まえ，本実験では，これらの事例を統合参照モデル作成における LLM の精度を著しく損なうものとはみなさず，LLM は本タスクにおいて一定の精度を有していると評価する．

次に，本タスクにおける LLM の安定性を検証するため，全 5 個の統合参照モデルに記載されている要件数を算出した．ここでいう要件数とは，統合参照モデル上に箇条書きで記載されている各要件を 1 件として数えたものである．

具体例として，4.2 節で示した図 3 の統合参照モデルを考える．この統合参照モデルでは，機能カテゴリ A に 2 件，機能カテゴリ B に 3 件の要件が記載されており，合計で 5 件の要件を有している．

同様にして，全 5 個の統合参照モデルにおける要件数を算出したところ，1 回目の試行から順に，69 件，62 件，66 件，62 件，65 件であった．この結果から，本実験では，要件数にして最大 7 件の試行間のばらつきが確認された．以下では，どのような要件の扱いにおいて試行ごとの差が生じたのかについて分析する．

まず，試行間で扱いにばらつきが見られた要件は，以下の三種類に分類できた．

- 意味的に分解可能な要件
- 議事録上で既存バージョンにおいて利用可能と記述されている要件
- 議事録上の説明が誤解を招くおそれのある要件

先ず，意味的に分解可能な要件について説明する．意味的に分解可能な要件とは，意味的には一つのまとまりとして捉えることもできるが，さらに細分化して，複数の関連する要件として解釈することも可能な要件である．これに該当する例として，「○○ と ●● を行う ×× を実装する」といった議事録の記

述が挙げられる．このような記述に対し，LLM が要件を抽出する際，「×× の実装」という一つの要件として扱う場合と，「○○ の実装」および「●● の実装」を，「×× の実装」に関連する個別の要件として扱う場合があり，試行ごとにばらつきが確認された．このような意味的に分解可能な要件の扱いの違いが，本実験における統合参照モデルの要件数のばらつきの主な要因となっている．

次に，議事録上で既存バージョンにおいて利用可能と記述されている要件について説明する．これらは，前述した統合参照モデルに記載された要件の妥当性の検証においても言及した要件である．議事録上において，既存バージョンで利用可能であるため新規開発は不要であると説明されている要件について，統合参照モデル上での扱いが試行ごとに異なることが確認された．

最後に，議事録上の説明が誤解を招くおそれのある要件について説明する．本実験では，試行ごとに扱いに差が生じたある要件について，その要件を対象外と解釈し得る表現が議事録中に確認された．当該要件に関する議事録の説明は，既存の手法ではなく，別の手法を用いて当該要件を実現することを意味していた．しかし，この表現の解釈が試行ごとに LLM で異なり，結果として，当該要件の統合参照モデル上での扱いに差が生じたと考えられる．

本実験では，以上のような要件に試行間のばらつきが確認された．そのため，統合参照モデルを作成する上での LLM の安定性には課題があるといえる．

5.2　LLM が生成した要件対応関係リストの評価結果

ここでは，LLM による欠落要件候補および根拠不明要件候補の検出タスクにおける再現率を算出し，要件対応関係リストの評価を行う．第 4 章でも述べたとおり，「△」または「×」の評価を Positive，「◯」を Negative と定義し，その判定が正しいかどうかを人手により確認する．

また，本実験の STEP2 では，欠落要件候補を検出するための対応関係の確認と，根拠不明要件候補を検出するための対応関係の確認という，二つの確認作業が行われる．本評価では，これら二つの確認作業を互いに独立したタスクとして扱い，それぞれについて再現率を算出した．さらに，この評価を，全 5 個の要件対応関係リストに対して実施した．

算出した再現率を，図 6 に棒グラフとして示す．

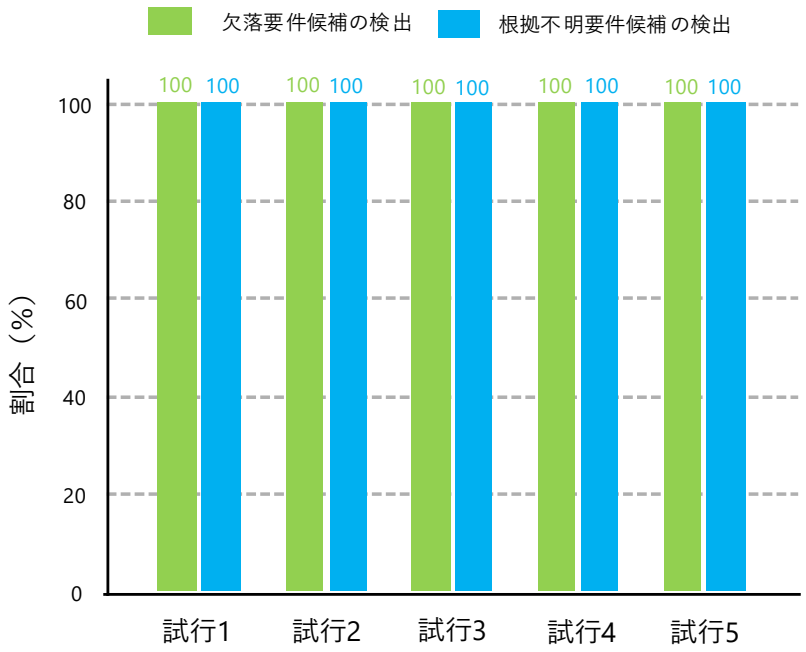

図 6　試行ごとの再現率

このグラフを見てわかる通り，全五回の試行における LLM の欠落要件候補と根拠不明要件候補の検出タスクにおける再現率は全てで 100% となった．このことから，LLM は欠落要件候補と根拠不明要件候補を漏れなく検出可能であり，要件対応関係リストを作成する上で LLM は一定の精度と安定性を有しているといえる．本実験において，真に検出するべき根拠不明要件候補は 2 件と少ないため，今回の結果で根拠不明要件候補の検出において LLM を評価するには慎重になる必要がある．しかし，同等のタスクである欠落要件候補の検出において，真に検出すべき欠落要件候補の数は 34〜41 件と決して少なくない数を LLM 全て正確に検出できているためこのような評価をする．

次に，評価の本筋ではないが LLM による欠落要件候補と根拠不明要件候補の検出の適合率を図 7 に棒グラフとして示している．

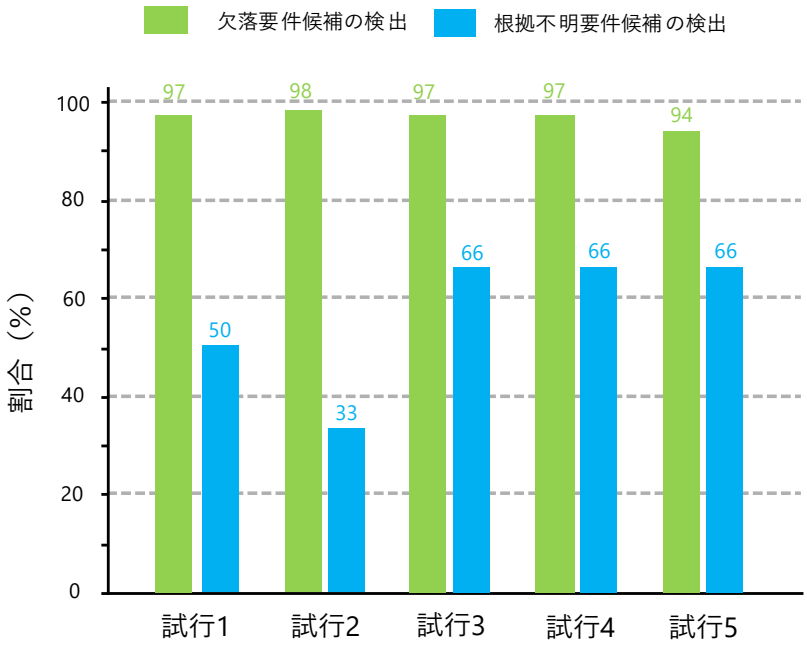

図 7　試行ごとの適合率

このグラフから分かるとおり，欠落要件候補の検出における適合率は非常に高い水準であった．具体的には，各試行において，LLM が誤って検出した欠落要件候補は 1〜2 件にとどまっていた．従って，本実験において LLM は欠落要件候補を過剰に検出しているわけではなく，高い適合率を維持しつつ，検出すべき欠落要件候補を適切に検出できているといえる．

一方で，図 7 に示すとおり，根拠不明要件候補の検出における適合率は，必ずしも高いとはいえない．この要因として，検出すべき根拠不明要件候補の総数が 2 件と少なく，単一の検出誤りが適合率に与える影響が大きいことが挙げられる．従って，LLM が誤って検出した根拠不明要件候補の数は 1〜4 件であり，決して多いわけではない点に注意が必要である．

更に，誤検出の原因を詳細に分析したところ，それらはいずれも，要件定義書に記載された要件と比較して，LLM が生成した統合参照モデル上の要件の方が説明が詳細であったことに起因していることが確認された．すなわち，要件定義書上の要件に対応する要件が統合参照モデル上に存在する場合であっても，統合参照モデル側の記述がより詳細であったため，LLM が当該対応関係を「◯」ではなく「△」と評価する事例が複数見られた．この結果は，LLM が対応関係の判定において厳密な基準を適用していることを示しており，前述した再現率の高さを裏付けるものと考えられる．

以上より，LLM は，欠落要件候補および根拠不明要件候補の検出において，一定の適合率を維持しつつ，非常に高水準の再現率を安定して達成できていると評価できる．

5.3　欠落要件候補と根拠不明要件候補の検証結果

本実験で得られた欠落要件候補および根拠不明要件候補について，当該プロジェクトのデータを提供していただいた企業のエンジニアに確認を行った．その結果，本実験で抽出された欠落要件候補および根拠不明要件候補は，実プロジェクトにおいて問題としては認識されていなかったことが分かった．また，要件定義書に基づいて開発されたソフトウェアは，実運用において問題なく動作していることも確認されている．

この要因として，当該プロジェクトでは，議事録に加えて，メールや電話，非公式な打合せ等を含む多様なコミュニケーションを通じて要件に関する意思決定が行われており，それらの内容はプロジェクト関係者の間で共有・理解されていたことが確認された．すなわち，本研究で利用可能であった議事録のみでは，プロジェクトにおける要件決定の全体像を完全に反映することが難しかったと考えられる．

一方で，このような非形式的なコミュニケーションを含めて全てを恒常的に記録し，アクセス可能な成果物として残すことは，実務において必ずしも容易ではない．本実験の結果は，関連成果物に基づく完全性レビューを適用する際には，利用可能な成果物の範囲とその性質を考慮する必要があることを示唆している．

6　考察

6.1　評価結果のまとめと考察

本実験を通して，提案した枠組みを実装する手段として，LLM が一定の有用性を有することが確認された．STEP1 における統合参照モデル作成タスクでは，議事録中の要件に関する情報を概ね正確に整理し，要件定義書に必要な要件を適切に抽出できることが確認された．

また，特に STEP2 において，統合参照モデルと要件定義書に記載された要件の対応関係に基づき，欠落要件候補および根拠不明要件候補を検出するタスクでは，再現率が全試行において 100% を記録し，適合率についても一定の高水準を維持していた．この結果から，LLM は STEP2 において，非常に高い精度と安定性を有していることが確認された．

一方で，STEP1 においては，LLM の出力に試行間のばらつきが見られ，安定性の面で課題があることが明らかとなった．しかし，この課題については，プロンプトの工夫や関連成果物の記述の明確化を行うことで，ある程度解決可能であると考えられる．実際に，本実験において STEP1 の安定性を損なう要因となった三種類の要件についても，同様の対応が有効であると考えられる．

具体的には，「意味的に分解可能な要件」および「議事録上の説明が誤解を招くおそれのある要件」については，関連成果物において，要件の粒度や意図をより明確に記述し，曖昧性を排除することで対応可能であると考えられる．また，「議事録上で既存バージョンにおいて利用可能と記述されている要件」に起因するばらつきについては，本実験で用いたプロンプトが，議事録中の情報から要件定義書に記載すべき要件を抽出する際の条件を十分に明示していなかったことが一つの原因であると考えられる．そのため，プロンプトをよりプロジェクトの特性に即した内容とし，どのような要件を統合参照モデルに含めるのかを明確に指定することで，この問題についても対応可能であると考えられる．

更に，適用実験を通して，提案枠組みを実データに適用する際の難しさも確認された．本実験の結果は，関連成果物に含まれる情報量やその網羅性が，枠組みから得られる結果の品質に強く影響することを示している．従って，本枠組みを要件定義書の完全性レビュー手法として実プロジェクトにおいて運用する上での重要な課題として，要件に関する必要な情報を十分に保持・共有できるような取り組みや手法を確立することが挙げられる．

6.2　妥当性への脅威

本研究の妥当性に関する主な脅威を，以下にまとめる．

先ず，本研究で用いた枠組みの実装方法は，対象としたソフトウェア開発プロジェクトの特性に依存している点が挙げられる．本実験では，議事録と要件定義書が明確に分離され，かつ時系列に整理された形で利用可能であったため，STEP1 および STEP2 の実装を比較的明示的に行うことができた．し

かし，プロジェクトによっては，関連成果物の構成や記述方法が大きく異なる場合があり，本研究で示した実装方法がそのまま他プロジェクトに適用可能であるとは限らない．むしろ，他プロジェクトに本枠組みを適用する際には，対象プロジェクトの成果物構成や記述様式に応じて，LLM の入力形式やプロンプト設計を柔軟に調整する必要があると考えられる．

次に，本研究では，要件の完全性のうち，機能要件に限定して検討を行った点が挙げられる．非機能要件や制約についても，関連成果物に基づく完全性レビューは重要であると考えられるが，それらは記述の抽象度や表現の多様性が高く，機能要件とは異なる性質を持つ．そのため，本研究で得られた結果が，非機能要件や制約を含む要件全体に対しても同様に成立するかについては，別途検証が必要である．

最後に，本研究の実験結果は，特定の LLM（gpt-5）を用いた場合の結果である点も妥当性への脅威となり得る．LLM のアーキテクチャや学習データの違いにより，生成される統合参照モデルや要件対応関係リストの内容が変化する可能性がある．本研究では，複数の LLM を用いた比較実験までは実施できていないため，他の LLM においても同様に枠組みの実装手段として有用性が確認できるかについては，今後の検証課題である．

7　おわりに

本研究では，ソフトウェア開発における要件定義書の品質確保という課題に対し，関連成果物との整合性に基づく完全性レビューを支援する枠組みを提案した．要件定義書は，上流工程の最終成果物として以降の設計，実装，テスト工程に大きな影響を与えるため，必要な要件が漏れなく記載されていることを確認する完全性レビューは不可欠である．しかし，実務においては，要件決定の根拠となる議事録等の関連成果物が多岐にわたり，人手でそれらを網羅的に参照することが困難であるという問題が存在する．

提案枠組みでは，議事録等の関連成果物から要件に関する情報を統合した統合参照モデルを構築し，これと要件定義書を対応付けることで，欠落要件候補および根拠不明要件候補を体系的に抽出する．統合参照モデルは，正解となる要件定義書を生成することを目的としたものではなく，完全性レビューにおける比較対象として位置づけられる点に特徴がある．

更に，提案した枠組みを現実的に実行可能とするための実装方法として，大規模言語モデル（LLM）を用いた実験を行った．実際のソフトウェア開発プロジェクトで作成された要件定義書および議事録を対象に適用した結果，要件の欠落や，要件決定過程が十分に記録されていない可能性のある箇所を一定程度検出できることを確認した．特に，要件対応関係の判定においては，高い再現率が安定して得られており，LLM を用いた完全性レビュー支援の実用可能性を示す結果が得られた．

一方で，本研究を通じて，関連成果物に基づく完全性レビューは，アクセス可能な成果物に含まれる情報の範囲に依存するという前提条件も明らかとなった．実プロジェクトにおいては，議事録以外のコミュニケーションを通じて要件に関する意思決定が行われる場合も多く，それらが必ずしも形式的な成果物として残されるとは限らない．ただし，本研究で対象としたプロジェクトでは，関係者間で要件に関する合意が形成されており，要件定義書に基づいて開発されたソフトウェアは実運用において問題なく機能していることが確認されている．このことから，本枠組みは実務上のプロセスを代替・否定するものではなく，形式化された成果物を用いて完全性を点検する補助的なレビュー手段として位置づけられる．

今後の課題としては，機能要件に限定せず，非機能要件や制約を含めた要件全体への適用可能性の検証，異なるプロジェクトや複数の LLM を用いた比較実験を通じた結果の汎用性および頑健性の評価が挙げられる．

謝辞

本研究は多数の方々の助けを得て行われました．お世話になった方への感謝をここで述べたいと思います．

楠本真二教授にはテーマ決めから始まり，実験，論文執筆，発表練習に至るまでの本研究のあらゆる工程で多大なるご指導を賜りました．研究に関して右も左も分からない非常に未熟な自分にも，時間をかけて寄り添っていただき常に研究の助けとなってくださいました．楠本先生のご指導がなければ本研究は成り立っていません．深く感謝を申し上げます．

柗本真佑准教授には発表練習の場で数多くのアドバイスをいただきました．研究内容の本質を突く客観的な視点からの指摘は研究活動で自身の考えを整理する上で大きな助けとなりました．深く感謝を申し上げます．

橋本美砂子事務補佐員には，日々の生活の面で様々なご支援をいただきました．特に，研究でうまくいかず苦しんでいる際にも，明るく接していただいたことが非常に印象的です．深く感謝を申し上げます．

楠本研究室の先輩方には研究室の日々の生活から研究活動に至るまで様々な場面でお世話になりました．先輩方の研究への姿勢は未熟な自分にとって大きな指針となり，様々な学びを得ることができました．他にも日々の雑談や研究室の様々なイベントでも積極的に関わってくださり，楽しい研究室生活を過ごすことができました．深く感謝を申し上げます．

楠本研究室の同期の皆様はともに研究を行う仲間として常に心の支えになりました．苦しい中でも研究を進めることができたのは，間違いなく皆様のおかげです．本当にありがとうございます．

そして自身のやりたい研究に取り組めたのは，これまで支え続けてくれた家族のおかげです．生活面や精神面で常に助けになってくれる家族の存在は非常に大きなものでした．深く感謝を申し上げます．

最後に，本研究を支えてくださった全ての方々に改めて感謝申し上げます．

参考文献

[1] ISO/IEC/IEEE International Standard - Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104 (2018).

[2] Denger, C. and Olsson, T.: Quality Assurance in Requirements Engineering, in *Engineering and Managing Software Requirements*, pp. 163–185, Springer (2005).

[3] Tamai, T. and Kamata, M. I.: Impact of Requirements Quality on Project Success or Failure, in *Design Requirements Engineering: A Ten-Year Perspective*, pp. 258–275, Springer (2009).

[4] Ezzini, S., Abualhaija, S., Arora, C. and Sabetzadeh, M.: AI-based Question Answering Assistance for Analyzing Natural-language Requirements, `https://arxiv.org/abs/2302.04793` (2023).

[5] Lin, J., Poudel, A., Yu, W., Zeng, Q., Jiang, M. and Cleland-Huang, J.: Enhancing Automated Software Traceability by Transfer Learning from Open-World Data, `https://arxiv.org/abs/2207.01084` (2022).

[6] Krishna, M., Gaur, B., Verma, A. and Jalote, P.: Using LLMs in Software Requirements Specifications: An Empirical Evaluation, in *Proc. International Requirements Engineering Conference (RE)*, pp. 475–483 (2024).

[7] Julia, M., Andreas, K. and Dirk, R.: A systematic literature review of pre-requirements specification traceability, *Requirements Engineering*, Vol. 29, No. 2, pp. 119–141 (2024).

[8] ISO/IEC/IEEE International Standard - Systems and software engineering — Software life cycle processes, *ISO/IEC/IEEE 12207:2017(E)*, pp. 1–158 (2017).

[9] Fernández, D. M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., Conte, T., Christiansson, M.-T., Greer, D., Lassenius, C., Männistö, T., Nayabi, M., Oivo, M., Penzenstadler, B., Pfahl, D., Prikladnicki, R., Ruhe, G., Schekelmann, A., Sen, S., Spinola, R., Tuzcu, A., Vara, de la J. L. and Wieringa, R.: Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice, *Empirical Software Engineering*, Vol. 22, No. 5, pp. 2298–2338 (2016).

[10] Zhang, Y., Jin, H., Meng, D., Wang, J. and Tan, J.: A comprehensive survey on automatic text summarization with exploration of LLM-based methods, *Neurocomputing*, Vol. 663, p. 131928 (2026).

[11] Nechakhin, V., D'Souza, J. and Eger, S.: Evaluating Large Language Models for Structured Science Summarization in the Open Research Knowledge Graph, `https://arxiv.org/abs/`

2405.02105 (2024).

[12] Oami, T., Okada, Y. and Nakada, aki T.: Performance of a Large Language Model in Screening Citations, *JAMA Network Open*, Vol. 7, No. 7, p. e2420496 (2024).

[13] Bracken, A., Babu, A. R., Whelehan, S., Merghani, K., Sheehan, E. and Feeley, I.: Ambient AI reduces documentation time and enhances quality in a simulated inpatient setting, *The Surgeon* (2025).

[14] Liu, B., Jiang, Y., Zhang, Y., Niu, N., Li, G. and Liu, H.: An Empirical Study on the Potential of LLMs in Automated Software Refactoring, https://arxiv.org/abs/2411.04444 (2024).

[15] Ahmed, T. and Devanbu, P.: Few-shot training LLMs for project-specific code-summarization, https://arxiv.org/abs/2207.04237 (2022).

[16] Zadenoori, M. A., Dbrowski, J., Alhoshan, W., Zhao, L. and Ferrari, A.: Large Language Models (LLMs) for Requirements Engineering (RE): A Systematic Literature Review, https://arxiv.org/abs/2509.11446 (2025).

[17] Gärtner, A. E. and Göhlich, D.: Automated requirement contradiction detection through formal logic and LLMs, *Automated Software Engineering*, Vol. 31, No. 2, p. 49 (2024).

[18] Alessandro, F., Stefania, G. and Laura, S.: Combining Established and Emerging Techniques to Detect Inconsistencies in Requirements, in *Proc. International Requirements Engineering Conference (RE)*, pp. 519–526 (2025).

[19] Gotel, O. and Finkelstein, C.: An analysis of the requirements traceability problem, in *Proc. International Conference on Requirements Engineering*, pp. 94–101 (1994).

[20] Konduru, K. C.: Scalability Drivers in Requirements Engineering, https://bth.diva-portal.org/smash/get/diva2:1049102/FULLTEXT02.pdf (2016).

[21] Ghazi, P. and Glinz, M.: Challenges of working with artifacts in requirements engineering and software engineering, *Requirements Engineering*, Vol. 22, No. 3, pp. 359–385 (2017).

[22] Femmer, H., Hauptmann, B., Eder, S. and Moser, D.: Quality Assurance of Requirements Artifacts in Practice: A Case Study and a Process Proposal, in *Proc. Product-Focused Software Process Improvement (PROFES)*, pp. 506–516 (2016).