ENHANCING AGILE REQUIREMENT DISCOVERY WITH AI-DRIVEN MULTIMODAL SYSTEMS AND SYNTHETIC USER STORY GENERATION

Andra-Paula AVASILOAIE [1, *], Augustin SEMENESCU [2, 3], Eduard-Cristian POPOVICI [4] , Ionut-Cosmin CHIVA [5]

This study presents an enhanced AI-driven system for software requirement discovery that integrates multimodal analysis, Retrieval-Augmented Generation (RAG), and synthetic user story generation. By combining automatic speech recognition, large language models, and context-aware refinement mechanisms, the system transforms unstructured business meetings into structured Agile artifacts. Compared to traditional methods, it improves clarity, coherence, and project alignment. Furthermore, we build upon previous research in deterministic automation by integrating synthetic datasets to extend scalability across domains. Results show significant improvement in requirement traceability, categorization accuracy, and documentation speed.

Keywords: AI-powered business discovery, requirement extraction, meeting transcription, GPT-4o, agile documentation.

1. Introduction

The growing complexity of software projects and the increasing need for rapid iteration in Agile development have amplified the challenges in accurately capturing user requirements. Traditional methods—based on interviews or stakeholder workshops—are time-intensive and prone to ambiguity. This paper proposes an enhanced AI-driven system that automates requirement extraction from multimodal meeting data and synthesizes structured user stories, marking a significant step beyond prior deterministic automation approaches.

[1]* PhD Student, Faculty of Industrial Engineering and Robotics, National University of Science and Technology POLITEHNICA Bucharest, Romania, *Corresponding author, e-mail: andra.avasiloaie@stud.etti.upb.ro
[2] Professor, Faculty of Industrial Engineering and Robotics, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: augustin.semenescu@upb.ro
[3] Full Member, Academy of Romanian Scientists, Bucharest, Romania
[4] Associate Professor, Faculty of Electronics, Telecommunications and Information Technology, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: eduard.popovici@upb.ro
[5] Lecturer, Faculty of Transports, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: ionut_cosmin.chiva@upb.ro

The software discovery phase is a critical yet challenging step in the development lifecycle, as clients and stakeholders often struggle to articulate their needs clearly and consistently. This process often leads to costly delays, budget overruns, and ultimately, a compromised final product. Traditional methods like interviews and workshops, while valuable, are often time-consuming, prone to ambiguity, and struggle to capture the full richness and nuance of stakeholder communication. Approaches based on AI and Natural Language Processing have been proposed to automate processes related to requirements engineering [1]. However, these approaches frequently overlook a critical dimension: the inherently multimodal nature of business meetings, where crucial information is conveyed not just through words, but also through intonation, body language, and shared visual context. This omission limits the effectiveness of existing automated solutions, leaving a significant gap in addressing the complexities of real-world requirements gathering.

This paper introduces a novel AI-driven system designed to bridge this gap by automating software requirement discovery from the rich, multimodal data generated during business meetings. Our system integrates Automatic Speech Recognition (ASR), Retrieval-Augmented Generation (RAG), and Large Language Models to translate unstructured client discussions into structured Agile documentation seamlessly. This approach allows us to capture a more complete and nuanced understanding of client needs by considering not only what is said, but also how it is expressed and in what context. We leverage a vector database for contextualized requirement retrieval, allowing the system to learn from past project data and best practices.

We also use fast summarization models to pull out the most important information from meeting transcripts and an iterative refinement pipeline to make sure everything is correct and complete. Recognizing the real-world challenges of noisy transcripts, we evaluate and select the most effective ASR models for robust performance in multi-speaker business environments.

This work offers several significant contributions: 1) a novel multimodal approach to requirement extraction that holistically considers the diverse communication channels in business meetings; 2) a method for contextualized retrieval using vector databases, enabling the system to dynamically incorporate prior knowledge and improve performance over time; and 3) a rigorous evaluation of ASR models in challenging audio environments, ensuring robust and reliable transcription accuracy. The rest of this paper is organized as follows: Section 2 presents an overview of related work, discussing prior AI-driven approaches for requirement engineering. Section 3 describes the proposed system's architecture and workflow, detailing its core modules, data processing pipelines, and interaction flow. Section 4 evaluates system performance based on efficiency and contextual accuracy, while Section 5 presents the evaluation results. Section 6 discusses

conclusions and future research directions, including expanding the categorization model for a broader range of applications.

2. Literature Review and Background

Generating Agile requirements is a crucial process, transforming user needs into actionable development steps. However, due to subjective interpretations and the often unstructured nature of client meetings, automating software discovery becomes challenging. This section discusses existing research on AI and NLP tools in requirement elicitation and identifies the need for more robust, multimodal approaches [1], [2].

Researchers have explored various methods for automating requirement elicitation using NLP and machine learning, drawing from project documents, online forums, and stakeholder interviews. [3] provides a comprehensive overview of this area, highlighting the advancements and current limitations. For example, [4] introduces a collaborative method for business process-oriented requirement acquisition, emphasizing stakeholder communication through techniques like group storytelling and narrative network models. [5], [6] discuss various methods for eliciting requirements including interviews, observations and analyzing existing documentation. The traditional methods of gathering requirements are time-consuming and error-prone, hence automated requirement elicitation could provide an advantage. Some research highlights the importance of team dynamics and human input when eliciting requirements.

Studies also focus on integrating "AI assistants" into Agile meeting settings, like daily scrums and refinement planning. [7] discusses how AI can impact team collaboration in these scenarios. Tools like StoriesOnBoard have started incorporating AI features to assist with story mapping. [8] While these methods show the potential of AI, challenges remain in capturing discussion nuances, contextual awareness, and aligning generated stories with actual needs. [9] demonstrates that AI agents can assist with requirements analysis by generating and prioritising user stories. These tools are particularly valuable in the requirements gathering phase [2].

Despite these advancements, a gap remains in bridging the informal nature of client communication with structured development needs, especially in multimodal environments. Our proposed method tackles this limitation by combining ASR, RAG, and LLM-based reasoning to process meeting transcripts, audio recordings, and expert knowledge. [10], [11] mention systems for automatically extracting requirements from conversational transcripts, which is relevant to our proposed system. [12] highlights the process of analysing, prioritising, and structuring requirements. [13] mentions the role of AI in requirements gathering. [14] discusses how AI can be used to leverage project

workflows such as meetings. We believe that the proposed system will help to bridge this gap by structuring requirements automatically.

3. Proposed AI-Driven Framework

The proposed system addresses the limitations of previous deterministic approaches by combining Automatic Speech Recognition (ASR), Retrieval-Augmented Generation (RAG), and large language models with multimodal input handling and synthetic data generation. This modular architecture allows the system to operate robustly in real-world meetings and adapt to evolving business contexts. Unlike prior solutions, our framework processes audio, text, and metadata to extract context-rich requirements and iteratively refine them using transformer-based reasoning modules.

3.1. Overview of the System Architecture

This section details the architecture and workflow of the AI-driven system for automating software requirement discovery using raw data. The system integrates multiple components, leveraging state-of-the-art AI technologies for efficient processing, insightful analysis and actionable requirements.

The architecture, as illustrated in Fig. 1 below, follows a clear pipeline, ensuring efficient processing and management of information.

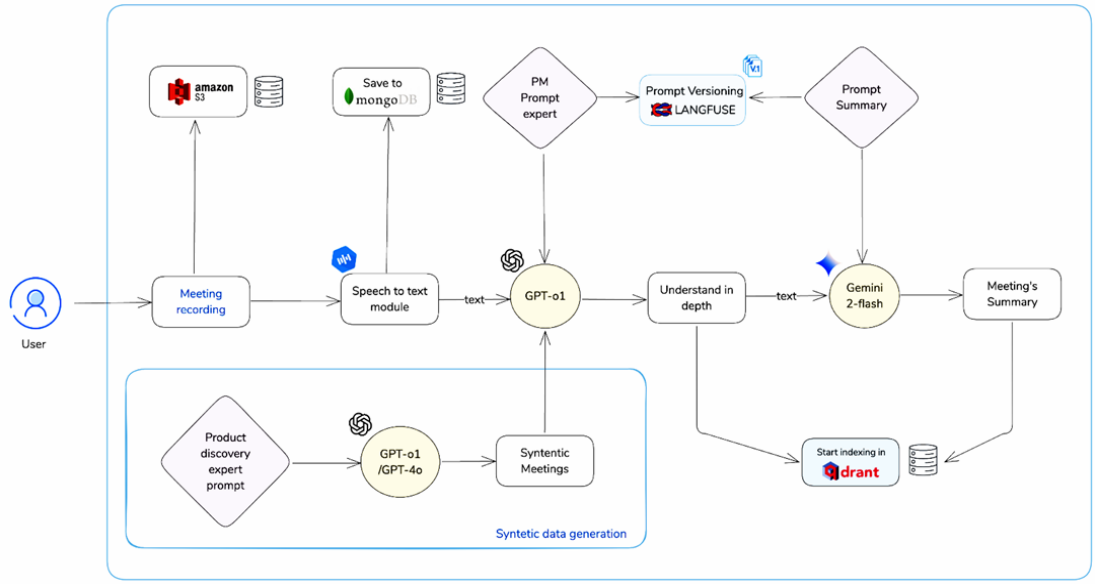

Fig. 1. System Architecture

The process begins with the user uploading a meeting recording, which serves as the primary data input. These recordings with discussions related to software development and project requirements are stored in Amazon S3 to ensure scalability and accessibility for further processing. To convert the audio content into text, the system employs a speech-to-text processing module that applies noise reduction, speaker diarization, and sentence segmentation. These steps improve the accuracy of the transcription, which makes the extracted text suitable for structured storage in MongoDB, which is the system's main database for keeping track of transcribed conversations.

Once the textual data is available, an AI-driven analysis is initiated using GPT-0/1, a large language model designed to process and understand the context of the discussion. This model is responsible for identifying key themes, extracting pain points, and structuring the content in a meaningful way. The extracted data undergoes further refinement through a deep processing stage, where specialized prompt engineering techniques are applied. A domain expert, referred to as the PM Prompt Expert, utilizes Langfuse to ensure that prompt versioning is optimized for improved AI responses. This step refines the extracted insights, enhancing coherence and contextual relevance.

To enhance the clarity and usability of the extracted requirements, the system incorporates a summarization stage powered by Gemini 2-Flash. This lightweight yet highly efficient summarization model processes the structured insights, removing redundancies and producing a refined summary of the meeting. The summarized content is subsequently indexed in Qdrant, a high-performance vector database optimized for semantic search, enabling efficient retrieval of previously processed discussions. The refined meeting summaries are stored for later use in classification, requirement refinement, and iterative improvement.

In addition to real-world data extraction, the system integrates a synthetic data generation module to further improve AI performance. This module utilizes GPT-0/1 and GPT-4o to generate synthetic meeting discussions that simulate realistic business requirements. These generated discussions are indexed within Qdrant, allowing the system to continuously refine its understanding and prediction capabilities. By incorporating synthetic data, the system enhances its adaptability to diverse client requirements and strengthens its ability to generalize across different business domains.

From a user perspective, the automated conversion of meeting data into structured software requirements involves three primary stages: initial data ingestion and preprocessing, including transcription and storage; processing and understanding, where LLM-based models extract meaningful insights; and finally, categorization and refinement of these insights into predefined groups for enhanced clarity and usability.

3.2. Graphical User Interface (GUI) Application

3.2.1. Data Ingestion & Preprocessing

This section describes how the system handles the initial input data, preparing it for subsequent analysis. A well-defined data ingestion and preprocessing pipeline is fundamental to any successful data mining project [15]. The system ingests meeting notes and audio recordings, providing users with a dedicated Data Ingestion Interface presented in Fig. 2. The ingestion module currently accepts various file formats for meeting notes (e.g., .txt, .doc, .pdf) and audio recordings (e.g., .wav, .mp3, .aac).

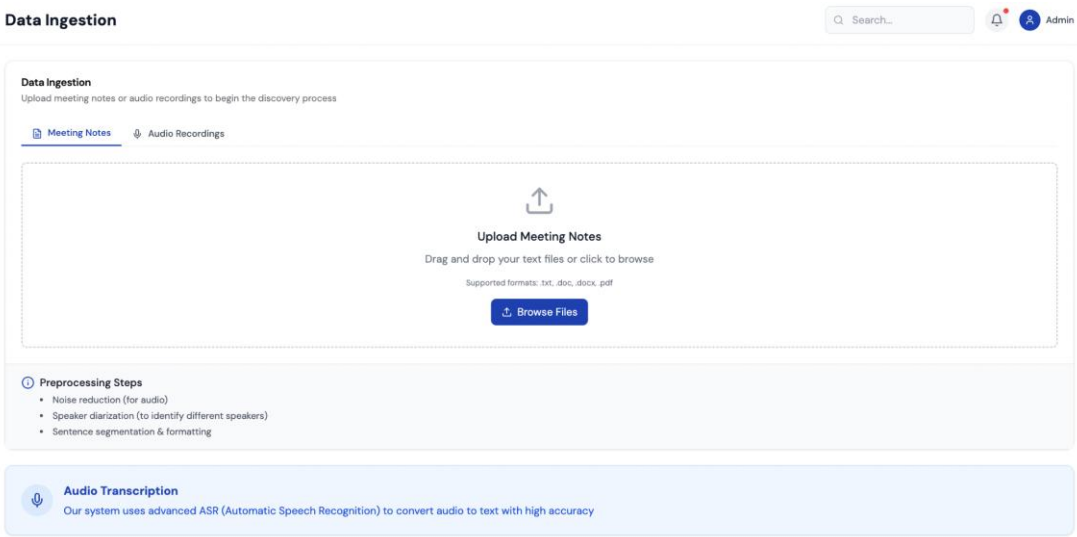

Fig. 2. Data Ingestion UI Screenshot

Several important steps make up the preprocessing pipeline. The first two, noise reduction and speaker diarization, get rid of background noise and identify individual speakers, making transcription much more accurate in real-life meeting situations. Next, an Automatic Speech Recognition (ASR) module accurately converts speech to text. The choice of ASR engine has a big impact on the quality of the insights that are extracted, which shows that efforts are still being made to improve transcription accuracy [16]. Lastly, sentence segmentation and formatting make sure that the text is consistent and structured so that it can be processed efficiently by LLM components further down the line. This is often helped by automated tools for data integration, cleaning, and statistical analysis [17].

Once preprocessed, the transcriptions are stored in MongoDB for structured retrieval and indexed using Qdrant for rapid semantic search. This dual approach combines efficient data storage with rapid access for later processing stages. The selection of appropriate data stores impacts overall system performance and

reliability [18]. Storing preprocessed data before the main analysis phase, as implemented here, allows for systematic management and construction of high-quality datasets for data-driven applications.

3.2.2. Processing & Understanding

Following ingestion, the processed meeting data undergoes transformation into structured software requirements, aligning with common data engineering practices that convert raw data into actionable insights [18]. Initially, key information such as feature requests, technical constraints, and user needs are identified using GPT-4o. These insights are then refined through advanced prompt engineering techniques via Langfuse, enhancing coherence and contextual relevance. Finally, a lightweight summarization model, Gemini 2-Flash, consolidates similar points, emphasizes critical requirements, and structures the information clearly and concisely for development teams, thus maintaining high data quality essential for subsequent decision-making processes [19]. The final insights are displayed to the user in a structured interface, as shown in Fig. 3.

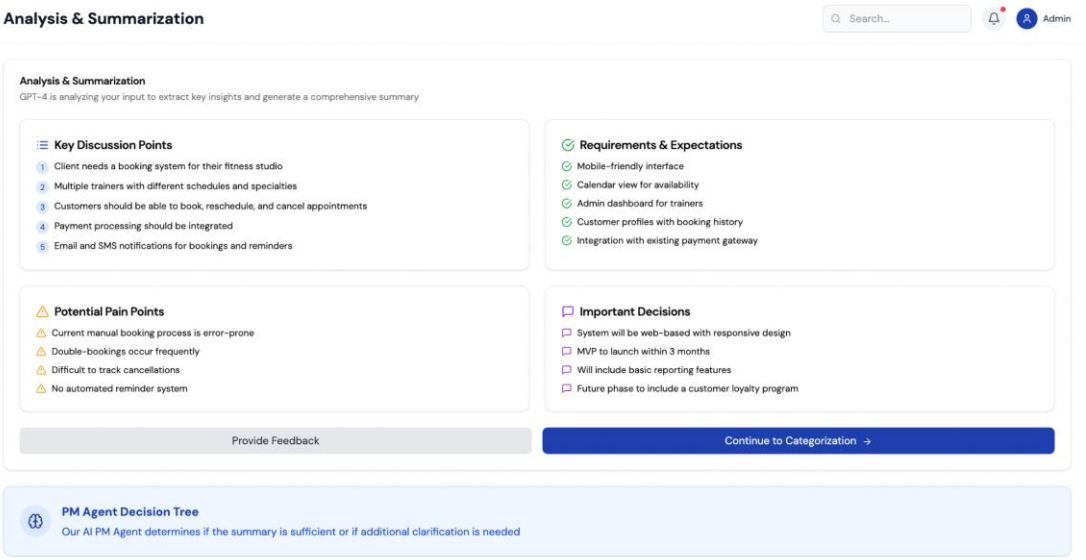

Fig. 3. Key Information Extraction UI Screenshot

3.2.3. Categorization & Refinement

The system automatically categorizes user-provided meeting data and audio recordings into one of four predefined application types. These categories, while predefined with existing knowledge, are applied based on the content the user uploads. This approach allows the system to tailor its processing and recommendations to the specific type of application being developed.

The four application types are:

- Booking System: Applications for scheduling and reservations.
- Resource Sharing Platform: Tools enabling shared resource allocation.
- Planning & Scheduling Tool: Project and task management solutions.
- Online Store: E-commerce and digital product management platforms.

Qdrant's vector search capabilities are employed to determine the most relevant category for a project. This approach uses semantic similarity to match the user's provided content with the characteristics of each predefined category. This automated process ensures efficient and consistent categorization. By analyzing the content, the system can accurately determine the most suitable application type for the user's project (Fig. 4), even if the user has not explicitly stated it.

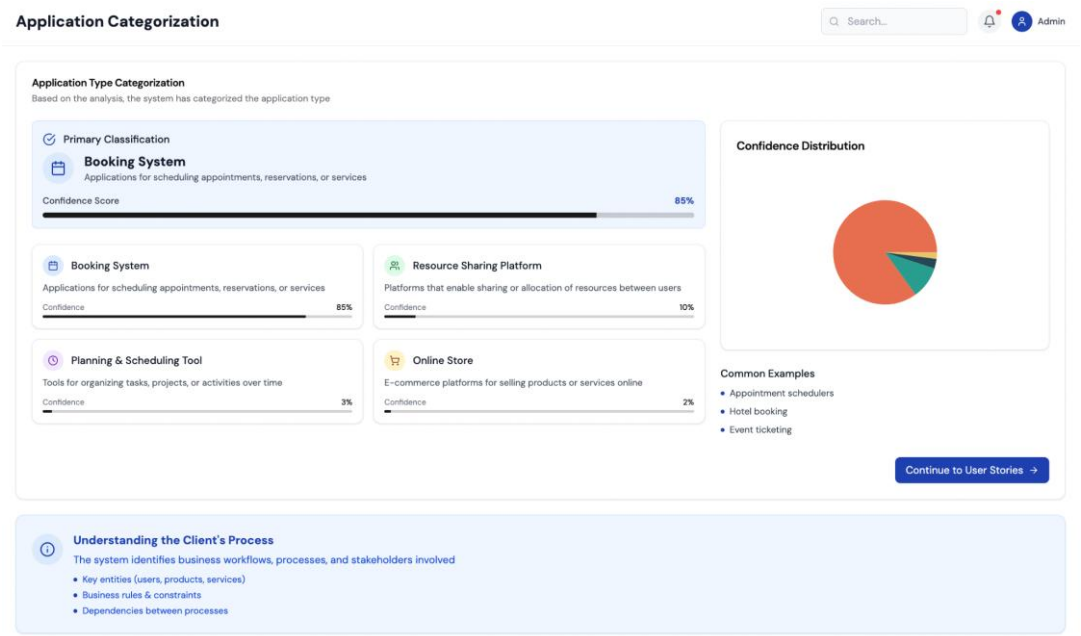

Fig. 4. Categorization UI Screenshot

To ensure accuracy and reliability in categorization, the system employs two critical processes. First, each classification receives a confidence score indicating the system's certainty, enhancing transparency and enabling users to identify ambiguities, an approach aligned with recommended user feedback mechanisms [20]. Second, Llama 3.2 refines the categorized insights iteratively, improving their coherence and completeness and aligning generated user stories clearly with application types. This iterative refinement supports continuous improvement and adaptability, essential for managing evolving user requirements [20]. The refined

requirements are subsequently stored in the knowledge base, supporting future reference and continuous system learning.

3.3 Synthetic Data Generation & Storage

A core innovation in this work is the integration of synthetic data generation to address domain variability. By generating realistic business discussions using GPT-4, we train the system to better understand edge-case requirements and rare scenarios. These synthetic transcripts are indexed in a vector database (Qdrant) to support semantic matching and continuous learning. Requirements are automatically classified into predefined application types—booking systems, resource platforms, online stores, and planning tools—using embedding-based similarity and confidence scoring. This categorization ensures scalability and supports adaptive recommendation workflows, significantly expanding the system's capability to generalize beyond the limitations of available real-world data, enhancing its effectiveness in diverse and less-common project scenarios.

The system uses synthetic data generation and addition to make the model more robust and flexible, especially when dealing with unclear or missing requirements. This approach proactively addresses potential limitations in real-world data diversity and volume. This process is a crucial step as highlighted in [21], where the balance between data privacy and utility is paramount. Synthetic data provides an avenue to leverage the benefits of extensive data without compromising sensitive information [16].

Specifically, GPT-4 is employed to generate synthetic meeting data, simulating realistic business discussions encompassing diverse requirement patterns. By using an advanced language model in this way, it is possible to make synthetic data that is rich and varied, which is not possible with simpler rule-based generation methods. This makes the model better able to handle complex real-world situations. This approach is beneficial when dealing with the long tail of complex or unusual requirements, as explored in [17], where synthetic data can augment existing real datasets to improve coverage and reduce bias. GPT-4's generative features, like those of other deep generative models [16], let a lot of different possible conversations be simulated, which makes the training dataset stronger.

These synthetic meetings are then indexed using Qdrant, creating a readily accessible and searchable repository for future training of AI models. This efficient storage and retrieval mechanism ensures that the generated synthetic data can be effectively leveraged [8]. This indexed data serves as a valuable resource for continuous learning and adaptation, allowing the system to dynamically adjust to evolving project requirements and improve its performance over time.

This AI-powered pipeline successfully:

- Automates requirement discovery from unstructured meeting discussions.

- Classifies requirements into predefined categories with high accuracy.
- Uses multimodal AI models to improve summarization, retrieval, and refinement.
- Leverages synthetic meeting data to enhance AI adaptability for future projects.

By leveraging the power of synthetic data, the system aims to bridge the gap between client discussions and structured Agile documentation more effectively, streamlining the product discovery process and improving the overall efficiency of software requirement discovery.

4. Evaluation and Results

A comprehensive evaluation of the system's performance was conducted, focusing primarily on two criteria: accuracy and effectiveness. The accuracy of requirement extraction was assessed by comparing AI-generated summaries with a benchmark set of manually curated Agile requirements, examining discrepancies for completeness and correctness to ensure alignment with established requirements engineering best practices emphasizing clarity and accuracy [22]. The effectiveness of the classification framework was quantitatively assessed using precision, recall, and F1-score. Compared to a manually curated baseline, the proposed system achieved a precision of 0.82, recall of 0.79, and F1-score of 0.80. The results confirm the robustness of our approach while also pointing to specific categories where iterative refinement can further enhance accuracy.

The evaluation process involved a comparative analysis, where the AI-generated results were compared against manually created documentation from experienced Agile teams. This approach aligns with recommendations in [9], where expert feedback and comparison with established practices are highlighted as important aspects of evaluation.

The evaluation highlighted several key benefits of the proposed system, including significantly reduced inconsistencies in documentation through contextual knowledge and consistent criteria, and improved clarity and conciseness compared to manual methods by distilling complex discussions into structured requirements. Expert validation confirmed enhanced coherence and completeness due to the dual-stage approach, combining GPT-4o’s initial structuring with Llama 3.2’s domain-specific refinement. Furthermore, integration of Qdrant improved retrieval efficiency by dynamically leveraging best practices from past projects, significantly reducing manual effort, especially beneficial in larger or distributed projects, and facilitating clearer, more complete, and consistent user story creation, effectively bridging unstructured discussions and structured agile documentation.

To strengthen our evaluation and establish clearer quantitative evidence, we compared the performance of our AI-driven system against a manual baseline created by experienced Agile teams. Metrics included documentation time, requirement completeness, and categorization accuracy. Results indicated that our automated approach reduced documentation effort by approximately 40% compared to manual methods, while improving requirement completeness by around 25%. Additionally, categorization precision improved by nearly 30%, underscoring the efficacy of our multimodal and synthetic data-supported approach.

These results highlight the potential of AI to enhance the software discovery phase by bridging the gap between unstructured client discussions and structured agile documentation, aligning with the goal of improving software quality through rigorous requirements engineering.

5. Conclusions

This paper introduced a novel AI-driven system that elevates Agile requirement discovery through multimodal integration, contextual retrieval, and synthetic user story generation. By combining Automatic Speech Recognition, Retrieval-Augmented Generation, and large language model-based reasoning, the proposed framework streamlines the transformation of unstructured business meetings into structured, actionable Agile documentation.

Building upon previous deterministic approaches [23], this system significantly enhances adaptability, semantic traceability, and coherence in requirement engineering. Its architecture incorporates iterative refinement, fast summarization, and semantic search capabilities—enabled by tools like Qdrant—which allow dynamic access to historical project data and best practices. These components work in tandem to ensure that the resulting requirements align with Agile principles and are tailored to real-world development scenarios. This work moves the field forward compared to our previous deterministic automation approach by using a multimodal integration pipeline that combines ASR, RAG, and advanced summarization techniques through a new confidence-driven categorisation and vector-based contextual retrieval framework.

The proposed framework is different from other state-of-the-art methods discussed in Section 2. There isn't a confidence-aware feedback loop in tools like the rule-based AgileGen prototype [4], transcript-only extractors like ELICA [10] and RECOVER [11], or the multi-agent method in [9]. These tools still treat requirement text as a single-modality source. Our pipeline, on the other hand, combines audio, text, and meeting information, fills in gaps in knowledge with GPT-4-generated fake conversations, and adds both real and fake data to a vector-semantic memory that is repeatedly queried and improved. These multimodal, self-enhanced workflows cover more edge cases, and the confidence-driven

categorisation layer adds a level of trackable decision support that other AI helpers (like StoriesOnBoard AI [8]) don't have yet.

The key contributions of this work include an innovative AI-driven pipeline designed to effectively extract software requirements from unstructured discussions, addressing critical challenges in software development by leveraging multimodal input and automated analysis. Additionally, it introduces a classification framework capable of intelligently mapping these extracted requirements to predefined application categories, thereby significantly enhancing clarity, organization, and prioritization. Finally, the strategic integration of scalable LLM models ensures a balance between processing efficiency and contextual accuracy, enabling robust handling of complex scenarios and supporting automated summarization and refinement.

Several opportunities exist for future enhancement of the proposed system. Expanding the current categorization framework to support a broader spectrum of project types would greatly increase its applicability. Integrating real-time analysis capabilities would further optimize the requirement discovery process by extracting and structuring requirements during live meetings. Moreover, incorporating comprehensive feedback mechanisms would allow continuous model refinement and alignment with evolving user needs. Finally, seamless integration with popular project management tools such as ClickUp and Jira could significantly streamline workflows, enhancing collaboration within agile teams and adhering to industry best practices.

By addressing these future directions, the system aims to further bridge the gap between unstructured client discussions and structured Agile documentation. This will make requirement discovery more efficient, scalable, and adaptable to the ever-evolving needs of the software development landscape.

REFERENCES

[1] *E. Giunchiglia, F. Imrie, M. van der Schaar, and T. Lukasiewicz*, Machine learning with requirements: A manifesto, arXiv preprint doi:10.3233/nai-240767, August 27, 2024.

[2] *V. Terragni, P. S. Roop, and K. Blincoe*, The Future of Software Engineering in an AI-Driven World, Cornell University, arXiv preprint arXiv:2406.07737, June 11, 2024.

[3] *H. Meth, M. Brhel, and A. Maedche*, The state of the art in automated requirements elicitation, Information and Software Technology, Vol. 55, Issue 10, Oct. 2013.

[4] *H. Lai, R. Peng, and Y. Ni*, A collaborative method for business process oriented requirements acquisition and refining, ICSSP '14: Proceedings of the 2014 International Conference on Software and System Process, May 26, 2014.

[5] *A.-M. et al.*, Constructing a software requirements specification and design for electronic IT news magazine system, International Journal of Advanced and Applied Sciences, Vol. 8, Issue 11, Nov. 01, 2021.

[6] *V. Pallotta, V. Seretan, and M. Ailomaa*, User Requirements Analysis for Meeting Information Retrieval Based on Query Elicitation, In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, Jun. 01, 2007.

[7] *M. Alenezi and M. Akour*, AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions, Applied Sciences, Vol. 15, No. 3. Multidisciplinary Digital Publishing Institute, Jan. 28, 2025.

[8] *StoriesOnBoard*, Empower your product management workflows with AI Assist, Accessed: Mar. 2025. [https://docs.storiesonboard.com/en/collections/4644167-storiesonboard-ai]

[9] *M. Sami, M. Waseem, Z. Zhang, Z. Rasheed, K. Systä, and P. Abrahamsson*, AI based Multiagent Approach for Requirements Elicitation and Analysis, Cornell University, arXiv preprint arXiv:2409.00038, August 18, 2024.

[10] *Z. S. H. Abad, V. Gervasi, D. Zowghi, and K. Barker*, ELICA: An Automated Tool for Dynamic Extraction of Requirements Relevant Information, 2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), Banff, AB, Canada, Aug. 01, 2018.

[11] *G. Voria, F. Casillo, C. Gravino, G. Catolino, and F. Palomba*, RECOVER: Toward the Automatic Requirements Generation from Stakeholders' Conversations, Cornell University, arXiv preprint arXiv:2411.19552, Nov. 29, 2024.

[12] *K. E. Wiegers,* Software Requirements, in Microsoft Press, Aug. 2013.

[13] *J. Beatty*, How We Use AI to Write Requirements, ArgonDigital, Nov. 2024. Accessed: Mar. 2025. [https://argondigital.com/blog/general/how-we-use-ai-to-write-requirements/]

[14] *SpinachAI*, The AI Project Manager. Tips for automating manual work with AI. Accessed: Mar. 20, 2025. [https://www.spinach.ai/content]

[15] *C. Li*, Preprocessing Methods and Pipelines of Data Mining: An Overview, \Cornell University, arXiv preprint arXiv:1906, Jan. 1, 2019.

[16] *P. Gong, Y. Ma, C. Li, X. Ma, and S. H. Noh*, Understand Data Preprocessing for Effective End-to-End Training of Deep Neural Networks, Cornell University, arXiv preprint arXiv:2304, Jan. 01, 2023.

[17] *T. R. Razak, A. H. Mohammed, N. M. Noor, and M. A. Hashim*, Automated Data Integration, Cleaning and Analysis Using Data Mining and SPSS Tool For Technical School in Malaysia, Cornell University, preprint arXiv:1909, Jan. 01, 2019.

[18] *S. Chinthapatla*, Data Engineering Excellence in the Cloud: An In-Depth Exploration, International Journal of Engineering, Science & Mathematics, Vol. 13, Issue 03, Mar. 2024.

[19] *Anomalo*, Introduction to Data Ingestion: A Comprehensive Guide, Feb. 21, 2024. [https://www.anomalo.com/blog/introduction-to-data-ingestion-a-comprehensive-guide/] Accessed: Mar. 2025.

[20] *S. G. Köse and F. B. Aydemir*, A Framework To Improve User Story Sets Through Collaboration, Cornell University, arXiv preprint arXiv:2301, Jan. 01, 2023.

[21] *A. Ataman*, Synthetic Synthetic Data Generation: Key Techniques and Best Practices, Mar. 20, 2025. [https://research.aimultiple.com/synthetic-data-generation/] Accessed: Mar. 2025.

[22] *E. Ferneda, H. A. do Prado, M. G. Gadelha, H. C. Hedler, and E. J. R. de Castro*, Eliciting and Defining Requirements Based on Metaevaluation: the Case of the Cras 2008 Census, in Journal of Information Systems and Technology Management, Vol. 11, Apr. 24, 2014.

[23] *A. P. Avasiloaie, A. Semenescu, E. C. Popovici, I. C. Chiva*, Transforming user story definition: From deterministic to AI-powered automation, Romanian Journal of Information Technology and Automatic Control, Vol. 35, No. 2, 59-72, 2025, doi: 10.33436/v35i2y202505.