Prompt Me: Intelligent Software Agent for Requirements Engineering - A Vision Paper

Jacek Dąbrowski[1,2][0000−0003−3392−0690], Amel Bennaceur[3][0000−0002−6124−9622], Gopi Krishnan Rajbahadur[4][0000−0003−1812−5365], Bashar Nuseibeh[3][0000−0002−3476−053X], and Faeq Alrimawi[1,2][0000−0002−2236−5073]

[1] Lero, the Research Ireland Centre for Software, Ireland
[2] University of Limerick, Ireland
{jacek.dabrowski, faeq.alrimawi}@lero.ie
[3] The Open University, Milton Keynes, UK
{amel.bennaceur, bashar.nuseibeh}@open.ac.uk
[4] Centre for Software Excellence, Huawei, Canada
gopi.krishnan.rajbahadur1@huawei.com

Abstract. [Context and Motivation] Software engineers can interact with users through digital channels (e.g., online forums) to exchange information about software products and achieve their requirements engineering (RE) goals. However, conducting RE manually is challenging due to the large number of users and the volume of their online feedback. **[Question/Problem]** Previous work has proposed tools to automatically extract useful information from online feedback (e.g., feature requests); however, these tools suffer from three major limitations: (i) an overlooked RE perspective in their design and evaluation; (ii) insufficient functional and performance capabilities; and (iii) missing evaluations of their ability to address RE needs. **[Principal Idea/Results]** This paper presents a vision for an intelligent RE software agent designed to overcome these limitations. Specifically, our vision explores how RE can guide the design and evaluation of software agents powered by large language models (LLMs), proposes empirical assessments of LLMs for RE usage and the agent's ability to meet RE needs. **[Contributions]** Our contribution is threefold: (i) a vision for an RE agent, (ii) identification of key challenges, and (iii) a roadmap to address current limitations.

Keywords: Requirements Engineering · Artificial Intelligence · Large Language Model · AI4RE · RE4AI · Software Agent · Bot · GenAI

1 Introduction

Software engineers can interact with users using digital channels (e.g., online forums) or built-in software features to exchange information about software products [5,11]. Users can, for example, inform engineers in their feedback about emerging issues, missing features, or the overall experience with a software product. Software engineers, on the other hand, can respond to users or interact with them via digital channels to resolve reported issues, clarify requirements, or acknowledge user engagement [11]. Such online feedback is a rich source of information guiding engineers in their RE tasks (e.g., requirements analysis).

However, conducting RE at scale is challenging due to the vast volume of users and their online feedback [12]. Popular apps like WhatsApp have more than 2.8 billion users, receiving more than 15,000 feedback messages daily [2]. Manually engaging with such a large number of users or analyzing their feedback is impractical and costly [5]. Yet, having comprehensive knowledge about users' needs is essential for RE tasks and to build the right software product [5].

Previous research has developed AI4RE tools to automate online user feedback (e.g., feedback classification) using AI techniques [11]. However, these tools have three main limitations: i) overlooked RE perspectives in their design and evaluation [12]; ii) they have limited functional and performance capabilities [13]; and iii) they lack empirical evaluations to address real RE needs.

The RE perspective for designing and evaluating AI4RE tools is mostly absent [11]; most works do not describe the envisioned RE use cases of their tools [12]. Answers to fundamental questions are missing (e.g., do these tools satisfy their users' actual goals? what are the right metrics to assess their practical usefulness? do these tools help save time for requirements elicitation?) [13].

AI4RE tools offer partial support for RE tasks (e.g., feedback classification), lacking end-to-end functionality for tasks like requirements elicitation or requirements specification [12]. Key features, like interviewing end-users to detail their requirements, are missing, while implemented features in these tools suffer from insufficient accuracy and scalability needed for practical use [13].

Most evaluations of AI4RE tools focus on assessing their effectiveness using ML metrics (e.g., precision) rather than practical usefulness [11]. While some evaluations report promising results, it is not clear whether the performance of these tools is good enough to be used in practice [13]. Moreover, their impact on RE tasks (e.g., do they improve requirements elicitation?) is rarely studied, though this is crucial for aligning research with real stakeholders' goals [11].

This paper presents our vision to address existing limitations by defining an empirically tested RE agent—an autonomous software entity designed to perform end-to-end RE tasks, such as gathering and detailing requirements. This agent aims to foster collaborative interactions between engineers and users, enhancing their skills. While the potential of agents for SE tasks was recognized over a decade ago (e.g., ERC grant on testing [16]), the application of agents to RE remains largely unexplored [14,18,20]. Current research in this domain focuses on using LLMs to automate specific RE tasks [6,23], with limited attention to proactive RE agents that acts in collaboration with stakeholders.

Our contribution is threefold: i) a vision for a virtual assistant enhancing RE through social interaction; ii) identification of key challenges, reflected in the literature, to pursue this vision; and iii) future pathways to address these limitations through holistic integration of AI4RE and RE4AI perspectives.

The paper is structured as follows: Section 2 covers state of the art. Section 3 introduces the vision of intelligent RE software agents. Section 4 outlines key challenge and pathway forward, and Section 5 concludes the study.

Criterion	**Our Vision**	**AI4RE** [12]	**RE4AI** [4]	**SEBot** [22]
Use of Software Agents	✓	×	×	✓
Supporting RE Tasks	✓	✓	×	×
RE-Driven Design And Evaluation	✓	×	✓	×
Alignment with RE Needs	✓	×	×	×

2 State of the Art

Existing research has examined synergies between RE and AI with objectives distinct from ours [4,12,22]. Table 2 shows the differences between our vision and contributions in AI4RE, RE4AI and SEBots research areas, pointing out the different criteria that guided our comparison: use of software agents, supporting RE tasks, RE-driven design and evaluation, and alignment with real RE needs.

AI4RE research is mostly data-driven rather than goal-driven [12], applying AI techniques to analyze user feedback and generate RE artifacts. Yet, it overlooks the RE-specific perspective in the design and evaluation of their analytics tools; key questions e.g., about their RE intended usage or their ability to satisfy the real RE needs are left unaddressed [11,13]. These tools are evaluated mainly based on ML metrics, providing little insight into their practical usefulness for practitioners. Their evaluation rarely focuses on RE-specific needs e.g., improved quality or effectiveness of RE tasks [8,11]. AI4RE tools offer limited feature sets (e.g., feedback classification) without end-to-end RE support [12]; they miss key features (interacting with users to capture and specify their needs), limiting the feasibility for RE automation. Moreover, the accuracy and the scalability of the tools is questionable for their practical use. Differently, our vision integrates RE perspective to guide the design and evaluation a novel RE agent. We aim to explore the use of LLMs to realise this vision; with our vision, we also strive to address the functional and the performance limitations of existing tools as well as to empirically evaluate the agent's ability to address the real RE needs [11].

RE4AI research integrates the RE perspective into AI system development, focusing on understanding AI-specific user goals (e.g., explainability) and provides guidelines to translate these goals into data-, model-, or system- requirements [4]. Despite its importance, RE4AI is under-researched compared to other SE4AI areas like Testing4AI [21]; it also lacks empirical validation on whether their RE4AI contributions meet the actual stakeholder needs [4]. Current studies focus mainly on AI systems within autonomous vehicles or robotics domains, with limited attention to software agents or LLMs in the RE context; only individual studies take the RE-driven perspective for defining AI4RE tools or LLMs [12,23]. Our vision similarly emphasizes the need for integrating RE perspective into AI engineering; but it focuses on RE agents powered by LLMs. This vision also underlines the need for empirical evaluations of new RE-centred methods for systems with LLMs, including the envisioned RE agent.

BotSE research find agents useful and time-saving for SE, with primary focusing on bug detection, code repair and quality assurance [22]. Individual studies proposed RE bots, but their tools are user-initiated rather than autonomous agent interacting with humans like in our vision [22]; these bots provide a partial support for single RE tasks rather than a holistic end-to-end aid leveraging LLMs. Neither their design nor evaluation was driven by RE-centred perspective. We have also no evidence about their practical usefulness. Our vision propose future pathways to address these limitations.

In summary, no existing research or commercial solutions fulfill the vision of scalable RE support with agents powered with LLMs. While agents' potential in SE is noted [22], human-interactive solutions are absent in RE [20]. Recent SE visions (e.g., Davi Lo's RE'24 keynote [19] and Ahmed et al.'s proposal [17]) highlight agents but lack a focused RE scope. Our vision is RE-centred, targeting specific RE research challenges and directions for the software agent. This vision integrates holistically RE and AI, adopting RE perspective for engineering the envisioned agent, while exploring its applications to revolutionize RE.

3 The Vision of Intelligent RE Software Agent

We envision a future where an intelligent RE software agent autonomously supports RE tasks with minimal human intervention. Unlike traditional RE tools that merely respond to inputs, this agent would proactively make independent decisions to achieve RE goals, such as reaching out to end-users for requirements clarification. Powered by LLMs, the agent would naturally engage with stakeholders through text-based conversations, building a comprehensive requirements knowledge base that enhances its reasoning and RE automation capabilities. By integrating with tools like requirements management systems and communication platforms (e.g., email and online forums), the agent could access, update, and share requirements information; respond to project needs in real time; and coordinate with team members as an active contributor.

What are the benefits of having the RE agent? RE agent could act as a virtual assistant for software teams, facilitating RE tasks through intuitive, text-based interactions with team members and end-users. It would improve collaboration by maintaining a unified set of requirements, reducing miscommunication, and enabling data-driven decisions [11]. Engaging directly with end-users, the agent would capture, clarify, and validate requirements, resulting in a more user-centered product. By documenting conversations and generating structured requirements, the agent would reduce manual effort, allowing the team to focus on strategic insights. It would enable more effective integration of user feedback, aligning software with users' needs and increasing satisfaction [15]. With access to project documentation and feedback, the agent could assist product managers in data-informed requirements prioritization [12]; for example, the agent could prioritize feature requests based on the number of users reporting them in online feedback [11]. Stakeholders could interact with the agent at their convenience, eliminating scheduling constraints and delays typical of traditional meetings.

Examples of the RE agent in action? Suppose the WhatsApp team wants to understand user requirements for the next release of their app. They could task the RE agent with providing a requirements specification backed by evidence from automated user feedback analysis, such as app reviews. If some of these requirements lack detail, the team might prompt the RE agent to conduct virtual interviews with users to clarify ambiguous requirements, gather additional details, and understand the motivations behind specific user needs. Now, let's say the team wants to redesign the app and explore new features. They can assign the RE agent to identify features referred in online user feedback in other competitive apps or use the agent collaboratively to brainstorm new use cases. The RE agent's extensive training and experience can offer new insights into the app's potential, revealing features or perspectives previously not considered.

Is this vision relevant? RE agents have the potential to significantly improve software quality, which is a high priority in an increasingly digital world [1]. Our vision aligns with the broader global goal of digital transformation by promoting user-centered design and contributing a novel RE solution. We hypothesize the envisioned RE agent will enhance user engagement and satisfaction. It will improve the efficiency of RE, making it more scalable and cost-effective, thereby contributing to the global goal of optimizing resource utilization.

Why now? The timing for this vision is ideal due to: i) recent advancements in artificial intelligence [14,18], ii) a growing focus on user-centered software development [5,11], and iii) the widespread availability of online user feedback [11]. Recent advancements in AI, including LLMs like GPT, BARD, and LLaMA [14,18], offer unprecedented opportunities to develop software agents with "human-like intelligence" that can understand, interpret, and respond to natural language, making it ideal for scalable RE tasks. Modern SE practices increasingly prioritize user-centered design, recognizing that meeting users' actual needs is critical for software success [15]. With current technology and easy access to user feedback, now is the ideal moment to create a novel software agent to support RE practices effectively.

Why LLM techniques? The RE agent will facilitate natural language processing tasks, such as generating requirements specifications, summarizing user feedback, or answering users' questions. LLMs are well-suited for the RE agent as they excel in natural language understanding, enabling it to handle complex users' queries [9,14,18]. LLMs are pre-trained on datasets of an unprecedented size and diversity, providing the RE agent with a broad spectrum of general knowledge and enabling it to answer a wide range of questions without extensive domain-specific training. LLMs can be continuously updated and fine-tuned, ensuring the RE agent would adapt to evolving users' needs. LLMs maintain context throughout conversations; they will enable the agent to understand references, respond to follow-up questions, and maintain coherent and effective human-like dialogues

4 Key Challenges and Pathways Forward

We now present three key challenges in realizing our vision, which overlap with gaps in the literature (see Sect. 2), along with solution pathways to address them.

Challenge 1: How can RE help to define and evaluate an LLM-based RE agent?

- *The Issue:* Successful adoption of an RE agent in SE practice requires defining the "right" agent for real-world use. However, RE4AI is under-researched and lacks empirically validated methods for practical applications, including those involving software agents [4]. Identifying key stakeholders collaborating with RE agent (e.g., end-users, practitioners, or other SE agents [22]) is not trivial; their requirements and use cases remain unclear. While some RE use cases have been defined [12], the agent is expected to enable new RE tasks (e.g., prompting users to articulate their needs). Requirements for the agent's intelligence, autonomy, and social responsibility (e.g., kindness) are also underdeveloped [6]. Additionally, RE4AI lacks tailored RE methods for LLM components; current practices like prompt engineering are preliminary [9], with limited guidance on effective prompts or relevant datasets for the RE use cases. Criteria for evaluating the RE agent and its LLM component remain undefined, leaving open questions about metrics to assess the agent's abilities in supporting RE tasks [9].
- *Solution Pathway:* Future research should investigate how RE methods can support the definition and evaluation of LLM-based RE agents. This includes developing RE-centered guidelines for design, deployment, and evaluation, focusing on system, model, and data levels. At the system level, research should establish goals, use cases, and metrics for RE agents, drawing insights from the AI4RE [11], RE4AI [4], and BotSE [22] literature. Empirical studies with practitioners and end-users will help identify actual RE needs, while principles from SE and HCI can guide design. At the model and data levels, requirements for the LLM component should leverage RE-specific findings from the LLM literature. New qualitative and quantitative studies can refine prompt strategies for RE applications, supporting effective prompt design and dataset selection.

Challenge 2: How well can LLMs support RE use cases?

- *The Issue:* The AI4RE literature primarily focuses on traditional machine learning techniques [11], which often demonstrate limited performance and suitability for RE applications. Although LLMs have shown promising capabilities, their potential application and effectiveness in RE use cases remain largely unexplored. Research on applying LLMs to support RE is only just emerging [7,23], leaving several critical gaps; it is unclear which datasets are suitable for training and evaluating LLMs specifically for RE tasks, and how effectively these models meet RE goals in real-world contexts. Additionally, questions remain about the transparency and interpretability of LLM outputs in RE applications. The RE community lacks publicly available replication packages for benchmarking AI4RE solutions [13,3], which creates challenges for empirical studies aimed at evaluating the effectiveness of LLMs in RE use cases and comparing them against existing ML-based RE tools.

- *Solution Pathway:* Future research should prioritize defining and empirically evaluating LLMs for RE use cases. This includes identifying specific RE tasks that LLMs can support. Initial studies can focus on evaluating LLMs using established RE use cases from the literature [7,12]; next, they should explore novel use cases that traditional RE tools could not previously address, leveraging unique LLM capabilities such as RE artifact generation and conversational interfaces. Empirical studies must be conducted with rigor to ensure validity, using both quantitative and qualitative data analysis. Following empirical SE guidelines (e.g., [3]) can help design robust experiments. Furthermore, these studies should produce publicly available replication packages. Each package should be well documented and reproducible [3,13], including RE-related datasets, detailed study protocols, LLM-based software, and supplementary scripts for running the entire experiment.

Challenge 3: Can an RE agent be useful for software practitioners?

- *The Issue:* Limited knowledge exists about the ability of current AI tools to meet real RE needs (e.g., improving efficiency in requirements specification or enhancing completeness in requirements elicitation) [11,13]. The envisioned RE agent faces similar uncertainties; there is no available knowledge regarding stakeholders' acceptance of the solution, nor insights into their perceived usefulness of the RE agent and its features.
- *Solution Pathway:* Future research should focus on prototyping RE agents and evaluating their effectiveness in meeting RE needs. This requires empirical user studies, such as interviews and surveys with potential stakeholders (e.g., developers or end-users) who interact with RE agents. These studies should explore both social and psychological dimensions, assessing the practical impacts on RE tasks and users' perceptions of the agent's value. The design of such studies should draw on empirical SE research methodologies [13,3] and interdisciplinary insights from fields like HCI, sociology, and psychology.

5 Final Remarks

This paper envisions an intelligent software agent designed to transform RE practices. Powered by LLMs, the agent will facilitate new RE use cases, foster more collaborative and engaged interactions among stakeholders, and contribute to enhanced automation of RE tasks. Key challenges include defining RE-specific frameworks for developing LLM-based agents, customizing LLM capabilities for RE tasks, and ensuring these agents meet practitioners' needs. We encourage researchers and industry professionals to follow our proposed roadmap to bridge gaps in the literature and advance RE practices.

Acknowledgments. This publication has emanated from research jointly funded by Taighde Éireann – Research Ireland under Grant number 13/RC/2094_2, and co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

1. EU Digital Strategy, https://eufordigital.eu/discover-eu/eu-digital-strategy/, Accessed: 2024-11-08
2. Statista. https://www.statista.com/statistics/1306022/whatsapp-global-unique-users/, Accessed: 2024-11-08
3. Abualhaija, S., Aydemir, F.B., Dalpiaz, F., Dell'Anna, D., Ferrari, A., Franch, X., Fucci, D.: Replication in Requirements Engineering: for RE Case. ACM Trans. Softw. Eng. Methodol. **33**(6) (Jun 2024)
4. Ahmad, K., et al.: Requirements engineering for artificial intelligence systems: A systematic mapping study. Inf. Softw. Technol. **158**(C) (Jun 2023)
5. Al-Subaihin, A.A., Sarro, F., Black, S., Capra, L., Harman, M.: App store effects on software engineering practices. IEEE Trans. Softw. Eng **47**(2), 300–319 (2021)
6. Alrimawi, F., Nuseibeh, B.: Meta-modelling kindness. In: MODELS. p. 280–289 (2024)
7. Arora, C., et al.: Advancing Requirements Engineering Through Generative AI: Assessing the Role of LLMs, pp. 129–148. Springer Nature Switzerland (2024)
8. Berry, D.M.: Requirements engineering for artificial intelligence: What is a requirements specification for an artificial intelligence? In: REFSQ. pp. 19–25 (2022)
9. Borg, M.: Requirements engineering and large language models: Insights from a panel. IEEE Software **41**(2), 6–10 (2024)
10. Dąbrowski, J.: Prompt Me: Intelligent Software Agent for Requirements Engineering (2024), https://prompt-me.github.io/
11. Dąbrowski, J., Letier, E., Perini, A., Susi, A.: Analysing app reviews for software engineering: a systematic literature review. Empirical Softw. Engg. **27**(2) (2022)
12. Dąbrowski, J., Letier, E., Perini, A., Susi, A.: Mining user feedback for software engineering: Use cases and reference architecture. In: RE. pp. 114–126. IEEE (2022)
13. Dąbrowski, J., Letier, E., Perini, A., Susi, A.: Mining and searching app reviews for requirements engineering: Evaluation and replication studies. Inf. Syst. **114** (2023)
14. Fan, A., et al.: Large Language Models for Software Engineering: Survey and Open Problems . In: ICSE-FoSE. pp. 31–53. IEEE Computer Society (May 2023)
15. Ferrari, A., Spoletini, P., Debnath, S.: How do requirements evolve during elicitation? an empirical study combining interviews and app store analysis. Requir. Eng. **27**(4), 489–519 (Dec 2022)
16. Harman, M.: Evolving program improvement collaborator. http://www0.cs.ucl.ac.uk/staff/M.Harman/epic-public-version.pdf, Accessed: 2024-11-08
17. Hassan, A.E., Oliva, G.A., Lin, D., Chen, B., Ming, Z., Jiang: Towards AI-Native Software Engineering (SE 3.0): A Vision and a Challenge Roadmap (2024)
18. Hou, X., et al.: Large language models for software engineering: A systematic literature review. ACM Trans. Softw. Eng. Methodol. (Sep 2024)
19. Lo, D.: Requirements Engineering for Trustworthy Human-AI Synergy in Software Engineering 2.0. In: 32nd Int. Requirements Engineering Conf. pp. 3–4 (2024)
20. Maalej, W.: From RSSE to BotSE: Potentials and Challenges Revisited after 15 Years. In: 5th Int. Workshop on Bots in Software Engineering. pp. 19–22 (2023)
21. Martínez-Fernández, S., et al.: Software Engineering for AI-Based Systems: A Survey. ACM Trans. Softw. Eng. Methodol. **31**(2) (Apr 2022)
22. Moguel-Sánchez, R., et al.: Bots in software development: A systematic literature review and thematic analysis. Program. Comput. Softw. **49**(8), 712–734 (Jan 2024)
23. Vogelsang, A.: From specifications to prompts: On the future of generative large language models in requirements engineering. IEEE Software **41**(5), 9–13 (2024)