Preprint Notice
This is a preprint of a paper accepted for presentation at the 33rd IEEE International Requirements Engineering Conference (RE 2025). Please note that this version may differ slightly from the final published version.

Intelligent Agents for Requirements Engineering: Use, Feasibility and Evaluation

Jacek Dąbrowski*†, Wanling Cai*‡, Amel Bennaceur§, Bashar Nuseibeh§, Faeq Alrimawi*†
*Lero, the Research Ireland Centre for Software, Limerick, Ireland
{jacek.dabrowski, faeq.alrimawi}@lero.ie
† University of Limerick, Limerick, Ireland
{jacek.dabrowski, faeq.alrimawi}@ul.ie
‡ Trinity College Dublin, Dublin, Ireland
{wanling.cai}@tcd.ie
§ The Open University, Milton Keynes, UK
{amel.bennaceur, bashar.nuseibeh}@open.ac.uk

***Abstract*—Large language models (LLMs) have enabled new tools in requirements engineering (RE), often in the form of intelligent agents or virtual assistants. These tools can transform how software engineers perform RE tasks and interact with stakeholders. However, existing research primarily focuses on showcasing the capabilities of these tools rather than their design and evaluation in RE-specific contexts. This limits our understanding of their practical value and hinders broader adoption. To address this gap, we propose a reference model to guide the design, use, and evaluation of intelligent RE agents. Our work introduces new RE use cases, along with evaluation metrics for intelligent RE agents. We present a study design to support systematic development and share early findings demonstrating the feasibility of our approach. The use cases show how agents can add value for RE practitioners, while our synthesized catalogue supports tool evaluation. Finally, our analysis of commercial agents reveals that these tools already support certain aspects of the envisioned RE use cases.**

***Index Terms*—Requirements Engineering, Artificial Intelligence, Large Language Model, Agents, Bots, AI4RE, RE4AI, BotSE, GenAI.**

I. Introduction

Recent advances in large language models (LLMs) such as GPT, Bard, and LLaMA [15], [24] have enabled a new generation of intelligent tools for requirements engineering (RE). These models enable software agents with human-like abilities to understand, interpret, and respond to natural language, creating new opportunities to enhance and scale RE activities [15], [24]. By transforming how engineers interact with stakeholders and manage requirements, such tools have potential to redefine traditional RE practices [11], [35].

While the potential of agents for SE tasks was recognized over a decade ago (e.g., ERC grant on testing [21]), their application to RE remains largely unexplored [15], [35]. Current research mainly focuses on 'reactive' tools that automate RE tasks after being prompted by users [45], rather than on 'proactive' agents that act autonomously, initiate interactions, and collaborate with stakeholders to achieve RE goals [11].

Existing works demonstrate novel capabilities of LLM-powered tools–for example, tracing links between requirements and code [31], [39], generating code from requirements [46], documenting specifications [32], supporting quality assurance [34], and classifying or operationalizing requirements-related information [16], [23].

Successful adoption of proactive agents in RE practice requires clearly defining their roles, use cases, and evaluation methods to assess their success in practical settings. Identifying appropriate use cases and evaluation methods is a critical aspect of any RE tool–software agents are no exception [14].

However, only a few studies adopt an RE-driven perspective when designing and evaluating AI4RE tools [14], [44]. Most do not describe RE use cases or provide methods for holistic evaluation in RE contexts [13], often relying solely on ML metrics (e.g., precision) [7]. These gaps limit our understanding of how agents can support RE and hinder their practicality.

In this preliminary study, we address this gap by developing a reference model that unifies RE use cases for software agents and offers a catalog of evaluation metrics to assess their alignment with RE goals. Specifically, we explore the following research questions:

- **RQ1:** What are the RE use cases for software agents?
- **RQ2:** What metrics can be used to evaluate RE agents?
- **RQ3:** What partial implementations of RE use cases exist in commercial software agents?
- **RQ4:** How do practitioners perceive RE use cases and evaluation metrics for software agents?

We developed a systematic framework to address these questions. First, we curated a dataset from scientific literature on AI tools and RE practice. Using this dataset, we defined RE use cases and compiled a catalog of evaluation metrics (RQ1-RQ2). We then mapped commercial agent features to these use cases to assess their feasibility (RQ3). We outlined a validation plan with practitioners for future work (RQ4).

Our key contributions are: i) a reference model to guide the design, use, and evaluation of intelligent RE agents; ii) a systematic study design for constructing and validating the model; and iii) preliminary results demonstrating its feasibility and practical relevance. The RE use cases offer new insights into how agents can support RE goals and provide practical guidance for researchers and practitioners. A catalog of evaluation metrics enables a holistic assessment of agent suitability. Our analysis of commercial agents shows partial alignment with these use cases, indicating their relevance.

II. Background and Related Work

A. Terminology

A software agent (in short: agent) is an autonomous software entity designed to interact with and assist users–actors who engage with the agent–by performing functions or providing information through natural language interactions [9]. An agent typically leverages NLP and AI techniques to support a range of functionalities [28]. The combination of these functionalities to achieve a specific goal constitutes a use case. We refer to an RE use case description (in short: RE use case) as the combination of functionalities facilitated by the agent (‘What’) to meet goals (‘Why’) related to RE activities [8].

B. Related Work

RE has traditionally focused on human-driven requirements development and management, supported by well-founded theories and methods [41], mostly for non-AI systems. Our study builds on these foundations (e.g., use case definition) but applies RE principles to AI-enabled systems–specifically, intelligent RE agents. While agent-oriented RE (e.g., TROPOS [10]) refers to agents as goal-driven models of stakeholders or system parts, we define an RE agent as an autonomous system that interacts with stakeholders to perform RE tasks [9].

As RE tasks grow in complexity, automation has increased, particularly in areas such as elicitation and traceability. Since most RE artefacts are written in natural language, AI and NLP techniques are particularly well-suited for RE tasks [17], [38], driving a surge in AI4RE research, including 200 publications on online user feedback analysis [13]; tools support tasks like extracting feature requests from online feedback or tracing requirements to code [29]. Recent LLMs have enabled more advanced tools with reasoning and generative capabilities (e.g., creating specifications) [47]. Yet, most AI4RE tools remain ‘reactive’; they respond only after being prompted by users, rather than being ‘proactive’ agents that initiate interactions, elicit stakeholder needs, or make suggestions autonomously [29]. Integration into RE workflows and validation remain limited. Our study addresses this gap by exploring how intelligent agents can support RE in practice.

With growing popularity of AI-enabled systems, the RE community has started incorporating RE perspectives into their development–an area known as RE4AI. This involves identifying AI-specific needs (e.g., explainability) and translating them into data-, model-, or system-level requirements [3]. Despite its importance, RE4AI is still underexplored compared to areas like Testing4AI [36]. Most work targets domains like autonomous vehicles or robotics, with little focus on software agents; moreover, only a few studies adopt an RE-driven view for AI4RE tools or LLMs [14], [44]. Our study contributes to RE4AI by proposing a reference model for the design, use, and evaluation of intelligent agents in the RE domain.

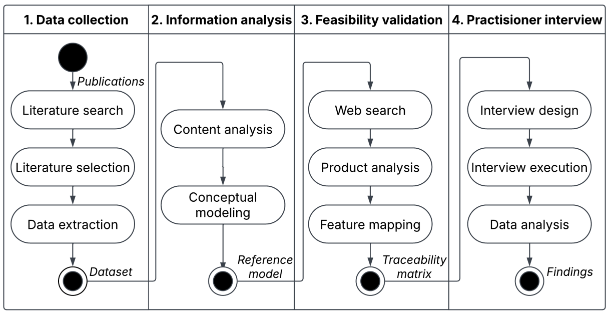

Fig. 1. Research methodology used in our study; headers present the four phases of the methodology; lanes illustrate steps in each phase.

Intelligent agents in software engineering (BotSE) have long been used for tasks like code repair or quality assurance [37]. Some RE bots exist, but they are user-initiated, lack autonomy and end-to-end RE support. Their design and evaluation rarely follow an RE-centered perspective, leaving their role in RE unclear. Our study addresses this gap by advancing the use and evaluation of RE agents through our reference model.

In summary, our work bridges RE4AI, AI4RE, and BotSE. While agents are known in SE [37], their use in RE is emerging [35]. We take a first step to address the lack of RE use cases and evaluations in recent visions [11], [22], [33].

III. Research Methodology

We adopted a systematic methodology to address RQ1-RQ4, comprising four phases: 1) data collection, 2) information analysis, 3) feasibility validation, and 4) practitioner interviews. Figure 1 outlines the phases (as headers) and their respective steps (as lanes). Phases 1 and 2 address RQ1-RQ2. For RQ1, we collected data on AI-powered tools (e.g., agents) from scientific literature and current RE practices (the ‘as-is’ state) from validated sources (e.g., [8]). We then analyzed this data to identify agent features and propose RE use cases (the ‘to-be’ state). RQ2 focused on evaluation metrics, which we structured into a catalog through content analysis. To address RQ3, we gathered data on commercial agents (e.g., IBM Watsonx [25]) from vendor websites and mapped their features to the proposed use cases to assess their feasibility. For RQ4, we sketched a plan to conduct unstructured interviews with practitioners; while this preliminary study does not answer RQ4, it outlines our future approach.

1) Data collection: We curated a dataset to address RQ1-RQ2, characterizing the use of intelligent agents (RQ1) and their evaluation metrics (RQ2). Following Kitchenham’s standard procedure[30], we searched for literature relevant to RE agents, selected studies based on predefined criteria, and extracted data into a spreadsheet. Unlike the original goal of the procedure–which aims to consolidate dispersed knowledge through a systematic literature review–our objective was to define RE-centered artifacts inspired by relevant studies.

Literature search. Research on RE agents is in its early stages [35], [37], with only a few publications to date, being insufficient to answer our RQs. Existing work focuses on application, with limited attention to agent evaluation [37].To broaden the evidence base, we extended our literature search to related fields aligned with key concepts in our research goals: agents (or bots), AI (including LLMs), software engineering (SE), and requirements engineering (RE). We focused on one broad area (software agents) and three specialized ones: AI4RE (AI for RE), RE4AI (RE for AI) and BotSE (Bots for SE); they provide relevant insights for our RQs. Following a rapid umbrella review approach [6], we limited the scope to secondary studies (e.g., literature reviews), which efficiently synthesize evidence from large bodies of primary research (e.g., more than 200 papers in AI4RE [13]) and offer stronger insights [6]. We conducted a keyword-based search in the Scopus digital library; we opted for this digital library as it indexes publications from over 7,000 publishers (incl. IEEE Xplore, Springer, and Elsevier Science). We applied filters for English-written publications and literature surveys only between 2010 and 2025, aligning with the emergence of the BotSE studies [2]. We formulated three search queries based on key concepts from our research questions and applied them to both metadata and full text.

The first query covering RE4AI and AI4RE areas:

```
(‘literature’ AND (‘review’ OR ‘survey’)
AND (‘requirements engineering’ OR ’RE’)
AND ((‘artificial Intelligence’ OR ‘ai’)
OR (‘machine learning’ OR ‘ml’) OR (‘large
language models’ OR ‘llm’)))
```

The second query covering BotSE area:

```
(‘literature’ AND (‘review’ OR ‘survey’)
AND (‘software agent’ OR ‘bot’ OR ‘chatbot’
OR (‘large language models’ OR ‘llm’)) AND
((‘software engineering’) OR (‘requirements
engineering’)))
```

The third query covering general-purpose agents:

```
(‘literature’ AND (‘review’ OR ‘survey’) AND
(‘software agent’ OR ‘bot’ OR ‘chatbot’) AND
((‘artificial intelligence’) OR (‘machine
learning’) OR (‘large language models’)))
```

TABLE I
INCLUSION AND EXCLUSION CRITERIA.

No.	Inclusion Criteria
1	Secondary studies (e.g., literature surveys) synthetising literature on RE4AI, AI4RE, BotSE or general-purpose software agents.
2	Peer-reviewed studies published as conference, journal, workshops papers or a book chapter.
No.	**Exclusion Criteria**
1	Papers not written in English
2	Papers that are not peer-reviewed (e.g., technical reports, preprints)
3	Tertiary studies (e.g., an umbrella literature reviews), technical reports or manuals.
4	Papers that are not secondary studies (e.g., primary studies).

Literature selection. Our Scopus search returned over 5,000 publications. Despite the large volume, most results appeared relevant to our study. We screened 400 randomly selected papers to obtain preliminary results and assess the feasibility of our methodology. Screening was conducted using predefined inclusion and exclusion criteria (see Table I) and involved reviewing titles, abstracts, and full texts. To ensure reliability, the first author independently classified all screened papers in two rounds; consistency was evaluated using intra-rater agreement, indicating very good agreement (Cohen's Kappa of 0.87) [26]. The screening resulted in 15 secondary studies [5]: 7 literature surveys in HCI, 4 literature surveys in SEBot, and 4 literature surveys in AI4RE (including LLM4RE). These studies analyzed information from a total of 2,050 primary papers. We used these secondary studies as sources for our extracted information.

Data extraction. The first author designed a spreadsheet to extract key information from each selected study. Using this form, they collected data from secondary studies to answer research questions RQ1–RQ2. We analyzed each collected secondary study for information on: (i) *functionalities* of agents that support specific use cases (e.g., interviewing users), and (ii) *metrics* used to assess how well an agent or other AI-supported tools achieve their objectives. To assess data extraction reliability, we evaluated it using intra-rater agreement [26]. The first author re-extracted data from all the selected studies; an external assessor then reviewed both rounds and calculated a percentage agreement of 81%, reflecting nearly perfect agreement [26]. The spreadsheets resulting from data extraction are available in our supplementary material [5].

2) Information analysis: We used the curated dataset from the previous phase to define RE-centered artifacts: RE use cases (RQ1) and a catalog of evaluation metrics (RQ2). We began with a content analysis of agent capabilities in the context of RE practices (‘as-is’), drawing on research-validated sources and widely used industry RE tools. This informed our vision of enhanced RE practices (‘to-be’). For RQ1, we used conceptual modeling to map RE challenges to agent capabilities, iteratively developing RE use cases. For RQ2, we analysed, grouped, and adapted evaluation metrics from the dataset to fit the RE context, resulting in a structured catalog.

Content analysis. The main goal of this step was to develop a theoretical and practical understanding of how software agents could be used and evaluated in RE. We conducted a content analysis on two complementary sources: i) our dataset from the previous phase, and ii) industry-oriented RE practices drawn from literature [8], [41] and RE tool vendors' websites. We began by analyzing industry materials to capture the current state of RE practices (the ‘as-is’ state). This provided a foundation for exploring how software agents could enhance RE (the ‘to-be’ state). Specifically, we reviewed two widely recognized RE books [8], [41], which offer insights into RE practices over time and are recommended for RE certification. These sources helped to identify key elements: RE goals

(‘Why’), such as requirements elicitation; actors (‘Who’), such as end-users, with whom RE interacts to achieve these goals; approaches (‘How’), such as interviews, used to reach RE goals; artifacts (‘What’) produced by RE tasks, such as requirement specifications; and information flow, referring to the information needed from actors to support RE activities. We further extended this analysis to RE tools identified through Trustradius, an online software benchmark platform [1]. The findings were documented in a spreadsheet [5]. Subsequently, we performed a content analysis of the dataset obtained during the data collection phase. We examined research agents' functionalities, along with evaluation metrics, to gain a better understanding of how agents might be utilized and evaluated in future RE practices (the ‘to-be’ state). We then grouped semantically related information (e.g., similar functionalities) to support subsequent data synthesis and conceptual modeling.

Conceptual modeling. We used datasets characterizing software agents (functionalities and metrics) and insights from RE practice analysis to define our reference model: RE use cases (RQ1) and evaluation metrics (RQ2). Through conceptual modeling, we iteratively developed use case descriptions (RQ1), beginning with an analysis of agent capabilities (e.g., facilitating meetings) and their alignment with RE goals identified in RE practice literature. We first mapped these capabilities to specific RE goals and approaches, and concurrently analyzed features of commercial RE tools (e.g., IBM DOORS, JIRA) to explore how RE agents could integrate with them to improve RE task. For each agent capability linked to an RE goal or/and R tool, we created short narrative descriptions outlining the problem addressed, the agent's functionality, and expected outcomes. We also noted the benefits for various stakeholders (e.g., project managers). We thematically grouped and synthesized these narratives into distinct use cases. Each use case (RQ1) was described using three elements: the RE goals the agent aims to satisfy (‘Why’), the agent's functionality that supports achieving the goal (‘What’), and the stakeholders the agent interacts with or impacts (‘Who’). To address RQ2, we analyzed our dataset of evaluation metrics, grouping them into semantically related categories and adapting their descriptions to RE practice and our use cases.

3) Feasibility validation: This phase aimed to validate the feasibility of the RE use cases (RQ3). We verified whether partial implementations of these use cases already exist. Specifically, we identified and mapped features of commercial agents to the functionalities referenced in our RE use cases.

Web search. We first searched for publicly available commercial agents to increase the reliability of our findings. To identify relevant tools, we used TrustRadius [1], a popular platform for comparing software tools. We listed all tools that fall under categories related to agents and bots.

Product analysis. Our analysis considered 5 commercial agents. In addition to the feature descriptions provided on the TrustRadius platform, we also visited their vendors' websites to further examine the tools' features. We recorded the identified features information in our spreadsheet [5].

Feature mapping. We validated use case feasibility using the matrix traceability method [21]. This method helps reuse system components by matching current needs (e.g., functionalities) with those of existing systems. We thus compared features of commercial agents with the functions referred in our use cases.

4) Practitioner interview: The final phase of our methodology seeks to validate the our reference model with practitioners. <u>This preliminary study does not yet include validation results, as these are scheduled for after the earlier phases are completed.</u> Initial interviews, designed using standard SE empirical research guidelines [43], [49], will be piloted with 2-3 participants and refined accordingly. We will then conduct full-scale interviews, analyze the results, and continue with further practitioner interviews.

Interview design. The interviews will evaluate the practical value of our reference model. We will gather practitioners' views on the relevance, applicability, and completeness of the use cases (RQ1) and evaluation metrics (RQ2). Their input will help refine the model. We will conduct semi-structured interviews to allow focused yet flexible discussions [48]. With 5 to 10 participants, we aim to reach data saturation-when no new insights emerge. Participants will have experience in RE and ideally some knowledge of AI solutions. We will recruit them through our networks and platforms like LinkedIn; a few suitable candidates are already identified. Each interview will begin with a brief overview of our research. We will ask a mix of open and closed questions, starting with the participant's background (e.g., role, experience, products). After presenting the reference model, we will ask for feedback on the use cases (e.g., ‘What challenges might arise?’) and the evaluation metrics (e.g., ‘Are they useful?’). The interview will end with general feedback.

Interview execution. We will conduct individual interviews with practitioners at their convenience, either in person or via video call. Each session will begin with a brief introduction to the research and its purpose. Participants will receive the interview guide and our model. With their consent, interviews will be audio-recorded to ensure accurate data collection.

Data analysis. We will transcribe the recordings for in-depth analysis. Using content analysis, we will identify key themes and insights, linking them to specific use cases and evaluation metrics. We will also apply descriptive statistics to support the qualitative findings. This feedback will directly inform refinements to our reference model and demonstrate how practitioner input shaped its final version.

IV. Preliminary Results

A. Requirements Engineering Use Cases (RQ1)

We present two RE use cases of intelligent agents (RQ1), based on literature-identified scenarios. Each use case is defined using data on agent functionalities, commercial RE tools, and RE practices (see Sect. III). These use cases can support engineering teams to: i) engage and collaborate with stakeholders, and ii) generate RE-related artefacts. Each use case is linked to five RE activities. We provide a description of the agent's intended use from (‘What’), explain how the agent helps achieve specific goals ('How'), and clarify its relevance to RE activities ('Why').

Use Case 1: Engage and Collaborate with Stakeholders:

Description (What): This use case focuses on enabling continuous, engagement with stakeholders across both direct communication and digital feedback channels. The RE agent interacts using natural language, through text, and integrates with platforms such as Microsoft Teams, online forums, and App Store feedback sections to capture stakeholder input. It monitors these channels for comments, questions, or suggestions related to requirements, reacts to messages by prompting for clarification, and can proactively reach out to users to collect structured feedback. RE agent supports interviews, workshops, and validation sessions by managing scheduling, reminders, and follow-ups, while maintaining awareness of discussion context and tracking decision history. It helps ensure conversations remain traceable and accessible over time, allowing teams to respond to stakeholder needs efficiently. By integrating with tools like Jira, Confluence, Jama Connect, and IBM DOORS, RE agent helps align feedback with development and validation workflows, supports traceability, and keeps stakeholders actively involved throughout the RE lifecycle.

Requirements Engineering Activities (Why): This use case contributes to several requirements engineering activities:

- *Requirements Elicitation:* Captures stakeholder input from direct conversations and online feedback; follows up with questions to clarify needs; discover requirements mentioned informally across platforms.
- *Requirements Analysis:* Analyzes intent, detects contradictions or missing details, and maintains context across touchpoints to support early requirement refinement.
- *Requirements Validation:* Supports live validation through structured workshops, feedback prompts, and walkthroughs with traceable context and conversation history.
- *Requirements Management:* Tracks all engagement activities, links feedback to requirements in requirements tools like Jira and IBM DOORS, and ensures discussions and decisions remain accessible, organized, and actionable.

Use Case 2: Generate RE-related Artefacts:

Description (What): This use case focuses on the automated generation and refinement of RE artefacts using AI capabilities, integrated with tools such as Enterprise Architect, Jama Connect, IBM DOORS, Jira, Confluence, and Microsoft Word. RE agent supports stakeholders, requirements engineers, analysts, developers, testers, project managers, and domain experts by transforming informal inputs–such as stakeholder discussions, documentation, and online feedback–into structured outputs. The agent generates artifacts such as user stories, software requirement specifications, and both functional and non-functional designs, UML diagrams, goal models, and Requirements Traceability Matrices. It integrates with Enterprise Architect to produce and update system models (e.g., UML, sequence diagrams), with Jama Connect and IBM DOORS to maintain traceability links and compliance documentation, and with Jira to convert requirements into user stories and tasks. The RE agent extracts requirements from interviews, backlogs, and chats; improves their quality; and creates formal specifications in Word or Confluence. It automates the conversion of natural language into formal models, updates domain artifacts, and generates reports and templates; it ensures consistency, traceability, and alignment across tools and teams.

Requirements Engineering Activities (Why): This use case contributes to several requirements engineering activities:

- *Requirements Analysis:* Refines and enhances requirements using quality checks, generates design concepts, and maintains traceability through integration with Jama Connect, IBM DOORS and Jira.
- *Requirements Specification:* Translates raw and refined requirements into formal documentation and structured models using tools like Word, Enterprise Architect, and Confluence, ensuring that outputs are testable, and standardized.
- *Requirements Validation:* Supports traceability between requirements and design/code/test artefacts using Jama Connect, IBM DOORS, and modeling tools, enabling validation through generated reports, checklists, and walkthrough-ready diagrams.
- *Requirements Management:* Automates updates to documentation and models, manages versioned artefacts, and traces changes across the lifecycle by linking requirements with implementation and testing artefacts in Jira, Confluence, and IBM DOORS.

B. Evaluation Metrics (RQ2)

This section presents a catalog of evaluation metrics for RE agents (RQ2), defined from information collected in the literature (see Sect. III). We have identified 99 unique metrics; each classified into one of 8 distinct categories; The largest number of metrics–36 (36%)–fall under Technical & Task-Oriented Performance, followed by 29 metrics (29%) in Language & Response Quality. User-Centric Evaluation contains 18 metrics (18%), while Understanding & Interpretability includes 11 metrics (11%). The Efficiency & Effort category holds 7 metrics (7%). Fewer metrics lie in Engagement & Interaction Dynamics with 3 (3%), Benchmarking & Framework Metrics with 2 (2%), and Collaboration & Multimodal Performance, including only 1 metric (1%). Table II presents a sample of 8 selected metrics from our catalog. We adapted their descriptions to illustrate how they apply to the two RE use cases. The complete catalog is available in our supplementary materials [5].

C. Feasibility Validation (RQ3)

We now validate the feasibility of our RE use cases (RQ3). Table III shows how features from five commercial agents map to capabilities in two RE use cases. Rows represent agent features, while columns show a sample of 20 out of 62 use case capabilities (32%) due to space limits. A '✓' indicates that the agent supports the corresponding capability. Among the five agents, ChatGPT [40] shows the widest coverage,

TABLE II
SAMPLE METRICS FROM OUR CATALOGUE FOR ASSESSING RE AGENTS ACROSS THE TWO IDENTIFIED USE CASES.

Metric Category	**Metric**	**Description**
Technical & Task-Oriented Performance	Task Completion Rate	Measures how effectively the RE agent completes tasks such as scheduling stakeholder meetings (Use Case 1) or generating SRS documents (Use Case 2).
Efficiency & Effort	Task Completion Time	Tracks how long it takes the RE agent to perform RE tasks like initiating feedback sessions (Use Case 1) or generating diagrams (Use Case 2).
Language & Response Quality	Context Coherence	Evaluates the RE agent's ability to maintain consistent and logical discussions with stakeholders (Use Case 1) or when interpreting feedback to generate artefacts (Use Case 2).
Understanding & Interpretability	Intent or Entity Recognition Accuracy	Measures the RE agent's ability to correctly identify intents/entities from stakeholder feedback (Use Case 1) or extract structured requirements from discussions (Use Case 2).
User-Centric Evaluation	Empathy	Assesses how emotionally intelligent or empathetic the RE agent is during sensitive or subjective stakeholder interactions (Use Case 1).
Engagement & Interaction Dynamics	Conversational Turns Per Session	Monitors how long stakeholders engage with the RE agent in requirement clarification (Use Case 1) or feedback review (Use Case 2).
Benchmarking & Framework Metrics	NASA-TLX	Quantifies stakeholder or engineer workload while interacting with the RE agent in activities like validation sessions (Use Case 1) or documentation refinement (Use Case 2).
Collaboration & Multimodal Performance	Agreement Rate	Measures how consistently the RE agent aligns with multiple stakeholders' input during collaborative RE activities (Use Case 1) or co-created artefacts (Use Case 2).

supporting 51 of 62 capabilities, with strengths in conversation management, NLP, and RE. IBM Watsonx [25] follows with 29 features, focusing on structured dialogue and technical documentation. Intercom [27] and Freshchat [18] support 20 and 22 features, mainly for real-time communication and basic tasks. Zoom AI Companion [50] supports 15 features, mostly related to meetings. On average, each tool supports 28 features. Only 9 capabilities are supported by all agents (e.g., 'converse in natural language'), while 30 are unique to a single tool–most of them to ChatGPT (e.g., generate UML diagram'). This highlights both tool specialization and the gap between generalist platforms like ChatGPT and more niche solutions.

V. DISCUSSION

A. Implication for RE Research

Our results show that intelligent agents can support a range of RE activities. Even with just two use cases, the alignment between agents and RE practices becomes clearer. These use cases can help researchers better justify, communicate, and design agentic solutions, while also raising awareness of the potential benefits among the SE community, practitioners, and the public. They may also inspire new applications or extensions beyond our current scope. Our catalog of evaluation metrics highlights often-overlooked dimensions in AI4RE tool assessment. While most studies emphasize technical ML metrics (e.g., precision, recall) [7], our work draws attention to practical factors like task completion and time efficiency. Evaluating tools along these broader dimensions can support more comprehensive and meaningful assessments. We will maintain a public repository of our use cases and evaluation catalog [5], encouraging community engagement and extension. Finally, our reference model can guide the design and evaluation of future RE agents–whether by focusing on specific use cases (e.g., artifact generation) or benchmarking tools, including LLM-based agents, against these use cases.

B. Implication for RE Practice

Our reference model helps practitioners understand the practical value of AI4RE research. By presenting simple, intuitive use case descriptions, it makes the potential of intelligent agents in SE workflows more accessible. Our feasibility validation underscores the industry relevance of key agent features, highlighting their usefulness in real-world RE contexts. The use cases and associated metrics show both the breadth of RE activities agents can support and the diverse evaluation dimensions practitioners should consider when adopting agentic tools. The model also provides a unified terminology to bridge communication between researchers and practitioners, encouraging exploration of cutting-edge solutions aligned with practical needs. Practitioners can use the model to identify automation opportunities, improve RE processes, and potentially enhance software quality. Additionally, the model may inspire the development and commercialization of new RE tools–addressing a gap, as most current solutions rely on general-purpose agents like ChatGPT rather than RE-specific ones.

C. Challenges and Opportunities

Our findings highlight several challenges and opportunities for future research. Although RE agents have been explored since early 90's [42], they have not gained practical traction. Was this due to immature technology or other barriers? Retrospective studies could shed light on why earlier efforts fell short. This study focused on functional capabilities, but future work should address non-functional requirements e.g., trust, transparency, and reliability [20]; and incorporate human-centered values. Key questions arise: What values should RE agents promote? What risks or unintended consequences might their use introduce? We assumed an ideal scenario, but future research should explore both positive and negative outcomes. As agents gain access to sensitive data and integrate with platforms like social media, issues of privacy, security, and responsible use become critical. We also did not deeply examine agent autonomy. RE agents should support–not replace–human engineers. With tools like GitHub Copilot [19] reshaping SE, it's time to ask: What is the evolving role of the requirements engineer? How much of RE should be automated? Our work contributes to these ongoing discussions and encourages the RE community to actively shape the future of practice.

TABLE III
TRACEABILITY MATRIX MAPPING A SAMPLE OF USE CASE CAPABILITIES TO FEATURES OF PUBLICLY AVAILABLE COMMERCIAL AGENTS.[1]

		Capabilities																			
		CO		**NLP**		**MD**		**TM**		**CM**		**E**		**MW**		**A**		**TL**		**D**	
		Track dialogue	Manage conversations history	Analyze text inputs	Classify user feedback intention	Create design models	Update domain models	Manage tasks and dialog	Monitor task progress	Identify relevant conversation	Monitor communication channels	Proactively contact users	Provide real-time feedback	Schedule and facilitate meetings	Remind tasks	Generate specification	Generate UML diagram	Detect traceability	Link requirements to artifacts	Generate reports	Automate documentation
Agents	Freshchat [18]	✓	✓	✓	✓			✓	✓		✓		✓	✓	✓						
	ChatGPT [40]	✓	✓	✓	✓	✓	✓	✓	✓				✓			✓	✓	✓	✓	✓	✓
	IBM Watsonx [25]	✓	✓	✓	✓			✓	✓		✓		✓	✓	✓						✓
	Zoom AI Companion [50]			✓	✓								✓								
	Intercom [27]			✓	✓			✓	✓	✓	✓	✓	✓	✓	✓						

[1] CO stands for Conversation; NLP denotes Natural Language Processing; MD signifies Modeling and Design; TM indicates Task Management; CM marks Communication Monitoring; E stands for Engagement; MW denotes Meeting and Workflow; A signifies Artefact Generation; TL indicates Traceability and Linking; and D marks Documentation.

D. Limitations

Internal validity: Our selection of secondary studies is not exhaustive, and some relevant works might have been overlooked. To reduce this risk, we used three structured search queries, based on SLR guidelines [30], incorporating key concepts and their synonyms. While manual steps such as study selection and data extraction introduce subjectivity, we followed a systematic process to enhance repeatability [30] and assessed intra-rater agreement to ensure reliability [13]. We admit that synthesizing literature and defining RE use cases involves interpretive bias–an inherent part of creative analysis.

External validity: This preliminary study analyzes only a fraction of the identified secondary studies, and we do not claim completeness in the RE use cases (RQ1), evaluation metrics (RQ2), or validation of agent implementations (RQ3). Our primary aim was to demonstrate the feasibility of our study design, with a full set of results planned for future work. The RE use cases are based on selected secondary studies and RE practice books [8], [41], which may not fully represent all industry practices, as these vary across organizations. To mitigate this, we chose books published across different time periods to reflect evolving industry knowledge.

Construct validity: Our definitions of agents and their features may not fully capture the intended constructs. To mitigate this, we used standard definitions adapted to the RE. However, inconsistencies in definitions across industrial and research sources may still affect the validity of our results.

VI. RESEARCH PLAN

Improve methodology: In future work, we will apply a systematic grouping schema and validate its reliability [13] to strengthen internal validity. Commercial agents were also selected subjectively, possibly overlooking relevant tools. We plan to adopt a more structured selection that includes research agents, improving both completeness and reliability.

Scale up the study: Our dataset is based on 2,050 papers from 15 literature surveys, representing only a subset of the studies identified. While results are informative, we aim to expand the dataset by systematically reviewing a broader range of surveys beyond the initial sample. This will enhance completeness and may uncover additional RE use cases and metrics.

Broaden the scope: The current focus on functional capabilities provides only a partial view of RE agents. Future work will include non-functional aspects, such as explainability (e.g., why agents interact with certain users) and human values (e.g., kindness) [4]. A broader perspective will help define not only what RE agents do, but how they should operate.

Practitioner validation: The reference model's practical value can only be assessed with practitioner input. We have designed a validation plan and will conduct a pilot study to evaluate the model's usefulness (see Sect. III). Practitioner feedback will guide refinements and shape the final validation phase.

Prototype and user study: To assess real-world suitability, we will develop a prototype RE agent for selected use cases and evaluate it in practical settings—helping to close the gap in user studies on AI4RE tools [13]. Planning is already underway, including participant recruitment and study design.

VII. CONCLUSION

Intelligent agents are increasingly streamlining SE tasks, with growing interest in their potential for RE. These agents could significantly transform RE, prompting key questions: What impact will they have, and is the RE community prepared? Despite their promise, research on their use, design, and evaluation in RE remains limited. To address this, we applied a systematic framework and developed a reference model of RE agent use cases and evaluation metrics. We curated a dataset that synthesizes insights from 15 secondary studies, covering approximately 2,050 publications across RE4AI, AI4RE, HCI, and BotSE. Based on this, we defined two RE use cases and compiled a catalog of evaluation metrics spanning technical, social, and other dimensions. We assessed feasibility by mapping use case features to existing commercial agents. The use cases demonstrate the potential value of agents in RE and offer practical guidance for their design and evaluation. The metrics catalog provides a unified framework for assessing agent suitability. Our analysis supports both the feasibility and relevance of RE agents in practice.

Acknowledgment

This paper was developed as part of the Prompt Me project implementation [11]. 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 under the Systems, Methods and Context (SyMeCo) programme grant agreement number 101081459. 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. The authors thank Dr. Alessio Ferrari for his helpful feedback on this work.

References

[1] Trustradius. https://www.trustradius.com/. Accessed: 2025-01-01.

[2] A. Abdellatif, K. Badran, and E. Shihab. A repository of research articles on software bots. http://papers.botse.org.

[3] K. Ahmad and et al. Requirements engineering for artificial intelligence systems: A systematic mapping study. *Inf. Softw. Technol.*, 158(C), June 2023.

[4] F. Alrimawi and B. Nuseibeh. Meta-modelling kindness. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, MODELS '24, page 280–289, New York, NY, USA, 2024. Association for Computing Machinery.

[5] A. author(s). Intelligent agents for requirements engineering: Use, feasibility and evaluation - supplementary materials. https://tinyurl.com/re-next-agent-submission.

[6] L. Belbasis, V. Bellou, and J. P. A. Ioannidis. Conducting umbrella reviews. *BMJ Medicine*, 1(1):e000071, 2022.

[7] D. M. Berry. Empirical evaluation of tools for hairy requirements engineering tasks. *Empir. Softw. Eng.*, 26(5):111, 2021.

[8] P. Bourque and R. E. Fairley, editors. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society, version 4.0 edition, 2024. Available online: https://www.computer.org/education/bodies-of-knowledge/software-engineering/v4.

[9] J. M. Bradshaw, editor. *Software agents*. MIT Press, Cambridge, MA, USA, 1997.

[10] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[11] J. Dąbrowski, A. Bennaceur, G. K. Rajbahadur, B. Nuseibeh, and F. Alrimawi. Prompt me: Intelligent software agent for requirements engineering - a vision paper. In A. Hess and A. Susi, editors, *Requirements Engineering: Foundation for Software Quality*, pages 235–243, Cham, 2025. Springer Nature Switzerland.

[12] J. Dąbrowski. Prompt Me: Intelligent Software Agent for Requirements Engineering, 2025. Available: https://prompt-me.github.io/.

[13] J. Dąbrowski, E. Letier, A. Perini, and A. Susi. Analysing app reviews for software engineering: a systematic literature review. *Empirical Softw. Engg.*, 27(2), 2022.

[14] J. Dąbrowski, E. Letier, A. Perini, and A. Susi. Mining user feedback for software engineering: Use cases and reference architecture. In *RE*, pages 114–126. IEEE, 2022.

[15] A. Fan and et al. Large Language Models for Software Engineering: Survey and Open Problems . In *ICSE-FoSE*, pages 31–53. IEEE Computer Society, May 2023.

[16] N. Feng, L. Marsso, S. G. Yaman, I. Standen, Y. Baatartogtokh, R. Ayad, V. O. de Mello, B. Townsend, H. Bartels, A. Cavalcanti, R. Calinescu, and M. Chechik. Normative requirements operationalization with large language models. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 129–141, 2024.

[17] A. Ferrari and G. Ginde. *Handbook on Natural Language Processing for Requirements Engineering: Overview*, pages 1–15. Springer Nature Switzerland, Cham, 2025.

[18] Freshworks Inc. Freshchat - live chat software. https://www.freshworks.com/live-chat-software. Accessed: 2025-03-01.

[19] GitHub. Github copilot, 2025. Accessed: April 8, 2025.

[20] M. Glinz. On non-functional requirements. In *Proceedings of the 15th IEEE International Requirements Engineering Conference*, 2007.

[21] M. Harman. Evolving program improvement collaborator. http://www0.cs.ucl.ac.uk/staff/M.Harman/epic-public-version.pdf. Accessed: 2024-11-08.

[22] A. E. Hassan, G. A. Oliva, D. Lin, B. Chen, Z. Ming, and Jiang. Towards AI-Native Software Engineering (SE 3.0): A Vision and a Challenge Roadmap, 2024.

[23] S. Hassani, M. Sabetzadeh, and D. Amyot. An empirical study on llm-based classification of requirements-related provisions in food-safety regulations. *Empir. Softw. Eng.*, 30(3):72, 2025.

[24] X. Hou and et al. Large language models for software engineering: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.*, Sept. 2024.

[25] IBM. Ibm watson. https://www.ibm.com/watson. Accessed: 2025-03-01.

[26] N. Ide and J. Pustejovsky, editors. *Handbook of Linguistic Annotation*. Springer, Dordrecht, 2017.

[27] Intercom, Inc. Intercom. https://www.intercom.com. Accessed: 2025-03-01.

[28] D. Jannach, A. Manzoor, W. Cai, and L. Chen. A survey on conversational recommender systems. *ACM Computing Surveys*, 54(7):165:1–165:36, 2021.

[29] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen. From llms to llm-based agents for software engineering. *CoRR*, abs/2408.02479, 2024.

[30] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, EBSE Technical Report, Keele University and Durham University Joint Report, 2007.

[31] K. Kolthoff, F. Kretzer, C. Bartelt, A. Maedche, and S. P. Ponzetto. Interlinking User Stories and GUI Prototyping: A Semi-Automatic LLM-Based Approach . In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 380–388, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[32] M. Krishna, B. Gaur, A. Verma, and P. Jalote. Using LLMs in Software Requirements Specifications: An Empirical Evaluation . In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 475–483, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[33] D. Lo. Requirements Engineering for Trustworthy Human-AI Synergy in Software Engineering 2.0. In *32nd Int. Requirements Engineering Conf.*, pages 3–4, 2024.

[34] S. Lubos, A. Felfernig, T. N. T. Tran, D. Garber, M. El Mansi, S. P. Erdeniz, and V.-M. Le. Leveraging LLMs for the Quality Assurance of Software Requirements . In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 389–397, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[35] W. Maalej. From RSSE to BotSE: Potentials and Challenges Revisited after 15 Years. In *5th Int. Workshop on Bots in Software Engineering*, pages 19–22, 2023.

[36] S. Martínez-Fernández and et al. Software Engineering for AI-Based Systems: A Survey. *ACM Trans. Softw. Eng. Methodol.*, 31(2), 2022.

[37] R. Moguel-Sánchez and et al. Bots in software development: A systematic literature review and thematic analysis. *Program. Comput. Softw.*, 49(8):712–734, Jan. 2024.

[38] M. Nayebi, H. Cho, and G. Ruhe. App store mining is not enough for app improvement. *Empirical Software Engineering*, 23, 2018.

[39] M. North, A. Atapour-Abarghouei, and N. Bencomo. Code gradients: Towards automated traceability of llm-generated code. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, 2024.

[40] OpenAI. Chatgpt. https://chatgpt.com. Accessed: 2025-03-01.

[41] K. Pohl and C. Rupp. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam – Foundation Level – IREB compliant*. Rocky Nook, 2nd edition, 2015.

[42] H. Reubenstein and R. Waters. The requirements apprentice: automated assistance for requirements acquisition. *IEEE Transactions on Software Engineering*, 17(3):226–240, 1991.

[43] C. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999.

[44] A. Vogelsang. From specifications to prompts: On the future of generative large language models in requirements engineering. *IEEE Software*, 41(5):9–13, 2024.

[45] A. Vogelsang and J. Fischbach. *Using Large Language Models for Natural Language Processing Tasks in Requirements Engineering: A Systematic Guideline*, pages 435–456. Springer Nature Switzerland, Cham, 2025.

[46] B. Wei. Requirements are all you need: From requirements to code with llms. In G. Liebel, I. Hadar, and P. Spoletini, editors, *32nd IEEE International Requirements Engineering Conference, RE 2024, Reykjavik, Iceland, June 24-28, 2024*, pages 416–422. IEEE, 2024.

[47] D. Winkler, S. Biffl, and M. Wimmer. Challenges in applying large language models to requirements engineering tasks. *Design Science*, 10:e10, 2024.

[48] C. Wohlin, M. Höst, and K. Henningsson. *Empirical Research Methods in Software Engineering*, pages 7–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[49] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. 01 2024.

[50] Zoom Video Communications, Inc. Zoom ai companion. https://www.zoom.com/en/products/ai-assistant. Accessed: 2025-03-01.