Title: Effectiveness of RE GenAI Tools: An Empirical Study

1. **Introduction**
2. **Literature review (Cite in text and provide reference)**

2.1. RE: Tasks and subtasks

2.2. Effectiveness measures for requirement specification produced using RE

2.3. Gen AI in RE and their effectiveness

3. **Methodology**

3.1. Research problem (RB) and questions (RQ)

RB: What are the challenges in RG?

RQ:

- *"How effectively do current **GenAI tools** address the consistency and completeness challenges in Requirements Engineering?"*
- *"What are the limitations of GenAI-assisted RE tools in real-world enterprise contexts?"*

3.2. Research design: What tools? How many? What are the metrics to evaluate and compare?

Variables: RE sub-tasks by GenAI Tool (Generic vs Specialty) by Effectiveness Metrics (Accuracy, ????

Data: Using literature – previous effectiveness evaluation studies (# of studies on one or more tools)

4. **Analysis & Discussion**

Summary of analysis: RE sub-tasks by GenAI Tool (Generic vs Specialty) by Effectiveness Metrics (Accuracy, ????)

5. **Conclusion and future work**
6. **Reference**

1. Introduction

Requirements Engineering (RE) plays a foundational role in determining the success of software and systems projects, yet it remains one of the most cognitively demanding and error-prone phases of the development lifecycle. Activities such as elicitation, analysis, specification, validation, and change management require extensive stakeholder interaction, domain understanding, and careful documentation. Persistent challenges including ambiguity, inconsistency, incomplete requirements, and communication gaps continue to contribute to project overruns and quality issues across industries (Geyer et al., 2025; Lubos et al., 2024).

The emergence of Generative Artificial Intelligence (GenAI), particularly large language models (LLMs), has introduced new opportunities to support and potentially transform RE practices. Recent research demonstrates that LLMs can simulate stakeholder interactions (Ataei et al., 2024), classify and demarcate requirements (Alhoshan et al., 2025; Wang et al., 2024), generate requirement specifications (Schiller et al., 2025), support Agile refinement activities (Spijkman et al., 2024), and assist with quality evaluation tasks such as clarity or completeness review (Geyer et al., 2025; Lubos et al., 2024). At the same time, domain-adapted models and retrieval-augmented approaches are being deployed within industry to improve the readability and traceability of large RE document sets (Uygun & Momodu, 2024).

Although the interest in GenAI-assisted RE is increasing rapidly, the effectiveness of these tools remains uneven across tasks. Studies highlight substantial performance differences between general-purpose LLMs and RE-specific or domain-adapted models, with issues such as hallucinations, lack of consistency, missing details, and incorrect domain assumptions limiting their reliability (Bhatia et al., 2024; Ebrahim et al., 2025). Moreover, most tools focus on a narrow subset of RE activities primarily elicitation, classification, or specification while downstream tasks such as change management, version control, and impact analysis remain largely unsupported.

Given the diversity of emerging tools and the variability in evaluation methods, a systematic synthesis of evidence is needed to understand where GenAI provides meaningful value in RE and where significant gaps persist. This study conducts an empirical synthesis of recent GenAI RE literature, analyzing tools across RE subtasks and quality metrics including accuracy, completeness, consistency, unambiguity, verifiability, and correctness. By

comparing generic, fine-tuned, domain-adapted, and multi-agent GenAI approaches, the study aims to assess their effectiveness and identify their limitations in real-world RE contexts.

2. Literature Review

2.1 Requirements Engineering Tasks and Subtasks

Requirements Engineering (RE) is generally structured around five core activities: elicitation, analysis, specification, validation, and management. Across the 15 studies reviewed, these activities appear consistently, even though the specific terminology and scope vary by paper.

Elicitation

Elicitation involves discovering stakeholder needs through interviews, scenarios, or exploration activities. Several papers introduce GenAI-based approaches that support this early-stage activity:

- **Elicitron** uses multi-agent LLM simulations to imitate diverse stakeholder roles, generate scenarios, and support question answer cycles during early requirement gathering (Ataei et al., 2024).
- **MARE** also employs multi-agent interactions, coordinating role-based agents to uncover latent requirements, facilitate structured dialogue, and generate preliminary requirement artifacts (Jin et al., 2024).
- The **prompt-engineering framework** by Ebrahim et al. (2025) demonstrates how strategically crafted prompts enable generic LLMs to perform stakeholder-like Q&A, brainstorming, and scenario generation.

Together, these works suggest growing interest in *automated or semi-automated stakeholder simulation* as a practical elicitation technique.

Analysis

Analysis tasks include classification, conflict detection, prioritization, modeling support, and feasibility reasoning. This stage is strongly represented in the reviewed literature:

- **Classification-focused papers** such as Alhoshan et al. (2025) and Wang et al. (2024) evaluate LLMs on distinguishing requirements from non-requirements or categorizing requirement types.

- **LLM-assisted modeling and refinement** is explored in Agile Model-Driven Development, where LLMs help refine user stories and suggest UML-like modeling structures (Spijkman et al., 2024).
- **Traceability and change analysis** also appear in the automotive domain, where localized LLMs support simplification of requirement texts in ways that affect downstream modeling and analysis activities (Uygun & Momodu, 2024).

These studies collectively highlight that analysis tasks are especially amenable to GenAI due to their text-classification and transformation nature.

Specification

Specification concerns the documentation of clear, verifiable functional and non-functional requirements:

- **ReqGPT**, a fine-tuned model, is designed explicitly for generating structured requirement statements and SRS-like output (Schiller et al., 2025).
- **MARE** and **Elicitron** provide specification support indirectly by generating draft requirements as part of their multi-agent workflows (Jin et al., 2024; Ataei et al., 2024).
- **Local LLMs in the automotive industry** simplify existing requirement documents to improve clarity and maintainability, effectively acting as specification editors (Uygun & Momodu, 2024).

These studies demonstrate that GenAI can both generate new specifications and improve existing ones.

Validation

Validation confirms that requirements are correct, consistent, complete, and testable:

- **Quality review of epics** is evaluated in the case study by Geyer et al. (2025), showing how GenAI can rate requirements along clarity, completeness, and specificity dimensions.
- **LLM-based SRS quality assurance** examines the ability of models to detect ambiguous or inconsistent requirement statements (Lubos et al., 2024).
- **Test-case generation** is explored by Bhatia et al. (2024), who assess ChatGPT's ability to derive system test cases from SRS text and highlight common issues such as incompleteness and hallucinated details.

Validation is one of the most heavily researched tasks, particularly in contexts where requirements quality directly affects downstream testing and development.

Management

Management activities include change tracking, versioning, and maintaining traceability across artifacts:

- **Kong et al. (2025)** examine how GenAI can support requirement change processes in enterprise environments, showing improvements in collaboration and impact understanding.
- **Automotive LLM deployment** influences traceability because simplification and rewriting of requirements affect downstream linkages (Uygun & Momodu, 2024).

Compared to other tasks, management is still relatively underexplored, but emerging work shows promising directions.

2.2 Effectiveness Measures for Requirements Specifications

The reviewed literature adopts a variety of measures to assess the effectiveness of GenAI-based RE tools. These can be grouped into three categories: task performance metrics, specification quality attributes, and human-centered evaluations.

Task-Level Performance Metrics

These metrics appear primarily in classification-oriented studies:

- **Accuracy, precision, recall, and F1-score** are used in requirements classification (Alhoshan et al., 2025), requirements demarcation (Wang et al., 2024), and related labeling tasks.
- These quantitative measures allow direct comparison between LLMs, fine-tuned models, and traditional ML baselines.

Task-focused metrics are precise but limited to activities with available ground-truth datasets.

Specification Quality Attributes

For tools that generate or transform requirement artifacts, effectiveness is assessed through established RE quality criteria:

- **Clarity, ambiguity, and specificity** appear in epic-level evaluations (Geyer et al., 2025).
- **Completeness and consistency** are emphasized in SRS quality-assurance studies (Lubos et al., 2024).
- **Correctness and verifiability** are discussed in test-case generation research (Bhatia et al., 2024).
- **Unambiguity and readability improvements** are examined in the automotive LLM case (Uygun & Momodu, 2024).

These criteria reflect ISO/IEC/IEEE 29148-style requirement quality attributes and are highly relevant to your research focus.

Human-Centered Evaluations

Several studies incorporate subjective input from real practitioners:

- **Interviews and surveys** with product managers (Geyer et al., 2025) and domain engineers (Uygun & Momodu, 2024) capture perceived usefulness, trust, and limitations.
- **User studies and expert ratings** evaluate the readability and correctness of generated requirements (Schiller et al., 2025).

Across papers, the pattern is clear:
LLMs provide strong first-pass assistance but require human oversight to ensure accuracy and domain correctness.

This aligns directly with research question regarding consistency and completeness challenges.

2.3 GenAI in Requirements Engineering and Their Effectiveness

Across the reviewed literature, GenAI tools fall into three broad categories: specialized RE tools, generic LLM applications with prompt engineering, and LLM-based classifiers.

Specialized RE GenAI Tools

These tools are custom-built for RE workflows:

- **Elicitron** and **MARE** use multi-agent LLMs to conduct elicitation, modeling, and preliminary specifications (Ataei et al., 2024; Jin et al., 2024).

- **ReqGPT** is fine-tuned specifically for requirement drafting and achieves higher-quality structured outputs compared to generic LLM prompting (Schiller et al., 2025).
- **Epic-quality evaluation** uses GenAI to assess requirement attributes in real Agile teams (Geyer et al., 2025).
- **Automotive Local LLM + RAG** provides domain-specific rewriting and significantly enhances requirement clarity (Uygun & Momodu, 2024).

These specialized tools show higher task fidelity and better alignment with RE standards.

Generic LLMs Adapted Through Prompting

Generic LLMs (e.g., ChatGPT, GPT-like models) are used widely due to their versatility:

- Prompt-based strategies enable brainstorming, scenario generation, classification, and quality review (Ebrahim et al., 2025).
- ChatGPT-driven test-case generation highlights both benefits (speed, idea diversity) and limitations (inconsistency, hallucination) (Bhatia et al., 2024).
- LLM-assisted Agile MDD workflows apply generic models to user story refinement and model suggestions (Spijkman et al., 2024).

Their flexibility is high, but performance is sensitive to prompt design and domain complexity.

LLMs as Classifiers in the RE Pipeline

Classification is the most quantitatively evaluated RE activity:

- Studies show that LLMs achieve competitive or superior performance in requirements classification tasks (Alhoshan et al., 2025).
- Demarcation research demonstrates that LLMs outperform some classical ML baselines (Wang et al., 2024).

These tasks benefit from structured datasets and allow clearer empirical comparisons.

Summary of Key Findings from Literature

Across all 15 papers:

- **Elicitation, classification, drafting, and quality review** are well-supported by GenAI.
- **Completeness and consistency** remain the most difficult quality attributes for LLMs to handle reliably.
- **Enterprise adoption** is emerging (e.g., automotive, Agile teams), but real-world studies are still limited.
- **Human oversight is mandatory** in all evaluated contexts.

These findings directly motivate your research questions about the effectiveness and limitations of GenAI-assisted RE tools, particularly in achieving **consistent** and **complete** requirements.

3. Methodology

3.1 Research Problem and Research Questions

Requirements Engineering (RE) involves several cognitively demanding activities elicitation, analysis, specification, validation, and management. Despite its centrality to software development, RE continues to face persistent challenges, including incomplete requirements, inconsistent specifications, ambiguous phrasing, and difficulties in maintaining traceability across evolving artifacts (Alhoshan et al., 2025; Spijkman et al., 2024). The emergence of Large Language Models (LLMs) and RE-specific GenAI tools offers potential relief, yet the degree to which these technologies mitigate long-standing RE problems remains unclear and uneven across tasks.

This study investigates the effectiveness of GenAI-enabled RE tools through a structured, multi-paper empirical synthesis. The central research problem is therefore defined as:

RB: *What persistent challenges in Requirements Engineering are addressed by contemporary GenAI-based tools, and to what extent are these tools effective across different RE activities?*

From this, two research questions are derived:

RQ1: *How effectively do current GenAI tools address consistency and completeness challenges in Requirements Engineering?*

RQ2: *What limitations arise when applying GenAI-assisted RE tools within real-world or enterprise settings?*

These questions aim to clarify both the quantitative and qualitative impacts of GenAI on RE practices, aligning with previous calls for systematic evaluation in this emerging domain (Lubos et al., 2024; Geyer et al., 2025).

3.2 Research Design

The study adopts an interpretive, multi-paper synthesis design. The objective is not to perform a meta-analysis but rather to construct a structured, comparative understanding of GenAI tool performance across RE subtasks, grounded in empirical evidence reported in recent literature.

3.2.1 Paper Selection

The corpus consists of **15 peer-reviewed or pre-print research papers** published between 2024 and 2025, selected using the following criteria:

1. **Topical relevance:**
   Papers must explicitly examine the application of generative AI, LLMs, or RE-specific AI tools within Requirements Engineering activities.
2. **Empirical or technical contribution:**
   Papers must provide one or more of the following:
   – empirical experiments,
   – case studies,
   – simulation results,
   – prototype evaluations,
   – or structured conceptual frameworks grounded in RE practice.
3. **Clarity of tool-task alignment:**
   Studies must describe at least one RE activity supported by the AI system, such as elicitation, classification, specification drafting, quality assurance, or test-case generation.

Papers lacking identifiable authorship, publication year, or methodological detail were retained only if the technical content directly contributed to RE task assessment.

The final sample spans diverse settings:
LLM-based requirements classification (Alhoshan et al., 2025), RE elicitation simulations (Ataei et al., 2024), automotive industry deployments (Uygun & Momodu, 2024), multi-agent reasoning frameworks (Jin et al., 2024), quality assessment in agile epics (Geyer et

al., 2025), fine-tuned RE-specific models (Schiller et al., 2025), and RE demarcation tasks (Wang et al., 2024), among others.

3.2.2 Data Structuring

To systematically compare RE-AI tools across heterogeneous studies, all papers were coded into a unified analytical schema representing:

1. **RE Subtasks**
   (e.g., stakeholder identification, interview simulation, requirement classification, conflict detection, functional requirement drafting, test-case generation).

2. **Tool Characteristics**
   – Generic LLMs
   – Fine-tuned RE-specific LLMs
   – Multi-agent systems
   – Local domain-specific LLM deployments
   – Prompt-engineering-enhanced models.

3. **Evaluation Dimensions**
   – *Subjective Fit*: perceived usefulness/appropriateness for the RE task.
   – *Objective Metrics*: accuracy, completeness, consistency, unambiguity, verifiability, correctness (when reported).
   – *Study Type*: controlled experiment, case study, simulation, user evaluation.

4. **Traceability and Versioning Support**
   For management-related tasks such as change tracking and RE artifact alignment.

Binary indicators (1 = supported; 0 = not reported) were used to standardize task coverage across studies with varied methodologies.

3.2.3 Coding Procedure

A structured coding process was applied uniformly across all selected papers:

1. **Initial Extraction:**
   Each paper was read and its AI tool(s), RE activity coverage, evaluation measures, and empirical evidence were extracted.

2. **Task Alignment:**
   Descriptions of tool functionality were mapped onto the RE activity taxonomy

(elicitation, analysis, specification, validation, management).
When a study addressed multiple subtasks (e.g., classification + conflict detection), each task was recorded independently.

3. **Metric Identification:**
   Reported quality measures (e.g., accuracy in classification, ambiguity reduction in rewriting tasks) were coded into the predefined inference categories.
   When no measurable outcomes were reported, the field was left as *not available* instead of inferred.

4. **Cross-Paper Normalization:**
   Tools differing in architecture (e.g., ChatGPT vs. Local Automotive LLM vs. multi-agent systems) were normalized to allow comparison based on functionality rather than implementation.

5. **Triangulation:**
   Evidence from case studies, surveys, and system evaluations was triangulated to infer relative strengths and limitations across tool types.

This procedure ensured consistent coding across studies with differing methodological depth and reporting formats.

3.3 Data Analysis Strategy

The coded dataset enables three forms of interpretive comparison:

1. **Task Coverage Analysis**
   Examines how frequently GenAI tools support specific RE subtasks.
   For example, classification appears widely supported (Alhoshan et al., 2025), whereas conflict detection or version control support is rarely observed.

2. **Effectiveness Evaluation**
   Assesses how tools perform across objective criteria—especially accuracy, completeness, and consistency—when reported empirically.

3. **Tool-Type Contrast**
   Compares performance differences between:
   – generic LLMs (e.g., GPT-4),
   – domain-specific LLMs (e.g., automotive models),
   – fine-tuned models (e.g., ReqGPT),
   – multi-agent reasoning systems (e.g., MARE).

4. **Contextual Limitations**
   Derives insights into scalability, hallucinations, domain-dependency, user oversight demands, and real-world adoption constraints.

3.4 Validity Considerations

Several measures were taken to ensure methodological rigor:

Construct Validity

RE tasks were defined using established taxonomies, ensuring consistent interpretation across studies. Ambiguous or loosely specified tool behaviors were excluded.

Internal Validity

Binary task-support coding reduces subjective interpretation but may underrepresent nuanced tool behavior. No unreported metrics were inferred to prevent bias.

External Validity

The dataset includes papers from multiple domains automotive, agile software development, academic evaluations, and industry environments enhancing generalizability.

Reliability

A uniform coding framework was applied to all papers, reducing inconsistencies arising from heterogeneous reporting styles.

4. Analysis & Discussion

4.1 Elicitation-Focused GenAI Tools

Elicitron and MARE

Specialized elicitation tools demonstrate the strongest task coverage among all categories.
Elicitron provides multi-agent simulations of stakeholder interactions, supporting stakeholder identification, interview-style prompting, brainstorming, and scenario generation (Ataei et al., 2024). Its multi-agent structure enables diverse persona generation and richer exploration of latent requirements, positioning it as particularly effective for the early phases of RE.

MARE similarly incorporates multi-agent collaboration, extending support beyond elicitation to include analysis, modeling, and partial specification activities (Jin et al., 2024). Its elicitation performance is grounded in role-based agent orchestration, which helps reduce blind spots inherent to single-LLM use.

Across both tools, the table shows **high subjective fit scores**, indicating perceived usefulness for early RE tasks.
However, effectiveness metrics such as accuracy, completeness, or consistency were **not empirically measured** in these elicitation studies, highlighting a methodological gap.

4.2 Classification-Oriented GenAI Tools

Requirements Demarcation and Grey-Box Evaluation

The most rigorous quantitative evidence appears in classification-focused tools.
Wang et al. (2024) report that DeBERTa, Llama2, and ensemble models achieve **up to 94.79% accuracy and weighted F1 scores of ~95.5%**, outperforming previous baselines for requirements/non-requirements demarcation. This places classification tools at the top in terms of **objective effectiveness metrics**.

Grey-Box evaluation frameworks similarly test LLMs across structured RE tasks (Schiller et al., 2024). While the Grey-Box approach does not match the classification accuracy of the demarcation models, it reveals consistent patterns—LLMs perform moderately well in structural tasks (e.g., classification, basic modeling) but degrade under tasks requiring abstract reasoning or domain grounding.

Classification tools thus show the highest **empirical** performance, but remain **narrow in scope**, addressing only one RE subtask.

4.3 Specification-Focused GenAI Tools

ReqGPT and LLM-Assisted Story Refinement

ReqGPT targets the generation of functional requirements and demonstrates strong subjective fit for specification tasks (Schiller et al., 2025). Human evaluators rated ReqGPT's outputs as more structured and readable compared to generic LLMs, reinforcing the benefits of fine-tuning for RE-specific language.

LLM-Assisted Agile MDD tools (Spijkman et al., 2024) support story refinement and partial model generation. This category extends LLM usage into design-support tasks, enabling the

transformation of user stories into semi-formal artifacts. While these tools partially improve **completeness**, inconsistencies and occasional hallucinations limit correctness.

Specification-focused tools therefore offer **strong structural benefits** but remain limited by content validity and dependency on domain-specific fine-tuning.

4.4 Quality Evaluation Tools

Epic Evaluator and Grey-Box LLM Assessment

The Epic Evaluator focuses on assessing the quality of Agile epics, particularly clarity, structure, and compliance with organizational conventions (Geyer et al., 2025). It is widely used within IBM's product management workflows and demonstrates strong subjective usability for quality review.

Grey-Box evaluations similarly investigate LLM performance in requirement quality tasks such as ambiguity reduction, consistency checking, and readability assessment. Preliminary findings show moderate improvements in clarity and consistency (Lubos et al., 2024), but correctness and verifiability remain weak.

Quality evaluation tools thus provide **high practitioner usefulness**,
but **limited objective evidence** on correctness or completeness.

4.5 Test-Case Generation Tools

ChatGPT-Based Test Case Derivation

Test-case generation represents one of the least mature categories.
Bhatia et al. (2024) show that ChatGPT-generated test cases:

- sometimes include incorrect steps
- frequently omit preconditions
- show inconsistent interpretation of SRS content
- contain ambiguous phrasing

Effectiveness metrics (accuracy, correctness, verifiability) scored the lowest among all categories in the table.

This indicates that test-case generation is **not yet reliable**, and requires substantial human verification.

4.6 Industry-Deployed Domain-Specific Tools

Automotive LLM + RAG Systems

The strongest real-world deployment evidence appears in the automotive domain. Uygun & Momodu (2024) document a production system using local LLMs with retrieval-augmented generation to simplify, rephrase, and structure large-scale automotive requirements.

The tool demonstrated:

- improved **traceability**
- reduced **ambiguity**
- higher **consistency**
- enhanced **readability**
- improved **analyst navigation** within large documents

This category uniquely shows **industrial-scale adoption**, indicating higher technology readiness levels compared to research prototypes.

4.7 Cross-Cutting Limitations Across All Tools

Hallucinations and Incorrect Inferences

Generic LLMs frequently hallucinate requirements, misinterpret domain rules, or invent missing data (Bhatia et al., 2024; Spijkman et al., 2024).

Inconsistency Across Repeated Generations

Repeated LLM outputs for the same prompt vary significantly (Oriol et al., 2025), especially in tasks requiring structured reasoning.

Domain-Dependence and Fragility

Tools without domain grounding (e.g., via RAG or fine-tuning) perform poorly on correctness and verifiability.

Lack of Empirical Quality Metrics

Most specialized RE tools (Elicitron, MARE, Epic Evaluator, ReqGPT) report **subjective fit**, but do not measure completeness or correctness quantitatively.

Limited Support for Downstream RE Tasks

Management tasks change tracking, version control, dependency management remain unsupported by most GenAI tools.

5. Conclusion and Future Work

This study examined the effectiveness of contemporary Generative AI (GenAI) tools across the core activities and subtasks of Requirements Engineering (RE). By systematically analyzing fifteen recent peer-reviewed and industry-validated publications, a structured evaluation was conducted across elicitation, analysis, specification, validation, and management tasks using both subjective fit measures and objective effectiveness indicators (e.g., accuracy, completeness, consistency, unambiguity, verifiability, and correctness).

The findings demonstrate that **GenAI tools provide uneven but meaningful support across RE tasks**.
Specialized elicitation and collaboration frameworks as **Elicitron** (Ataei et al., 2024) and **MARE** (Jin et al., 2024) show strong coverage of early-stage activities including stakeholder identification, interview simulation, scenario generation, and preliminary analysis. Classification-oriented models exhibit the strongest empirical performance, with demarcation tools such as DeBERTa and Llama2 achieving **state-of-the-art accuracy and F1 scores** (Wang et al., 2024). Tools focused on specification, notably **ReqGPT** (Schiller et al., 2025), improve structural clarity and readability of functional requirements but remain limited by domain correctness and completeness. Validation-centric tools such as the **Epic Evaluator** (Geyer et al., 2025) demonstrate high practitioner utility for quality review, while test-case generation tools lag significantly due to inconsistency, missing details, and hallucinations.

Importantly, **industry-deployed domain-specific systems**, exemplified by local LLMs and RAG-based tools in the automotive sector (Uygun & Momodu, 2024), show the most promising real-world impact. These tools enhance traceability, reduce ambiguity, and support large-scale document understanding indicating that domain grounding and controlled deployment environments significantly improve performance.

Despite these advances, several cross-cutting limitations persist. Many tools lack formal evaluation of key quality metrics such as correctness, verifiability, and completeness. Hallucinations, inconsistency across repeated outputs, and limited domain robustness continue to undermine reliability. Furthermore, GenAI tools provide minimal support for

downstream RE tasks particularly change management, dependency tracking, impact analysis, and version control highlighting significant functional gaps.

Overall, the evidence suggests that **GenAI is highly effective for selected RE tasks**, especially elicitation, classification, and quality review, but still lacks the reliability required for autonomous operation. Human oversight remains essential, particularly in safety-critical or domain-intensive contexts.

5.1 Future Work

Based on the observed gaps, several directions emerge for future research and tool development:

1. Benchmarking Frameworks for RE Quality Metrics

There is a need for standardized, multi-dimensional benchmarks that measure correctness, completeness, consistency, ambiguity, and verifiability of AI-generated requirements. Existing studies rarely evaluate more than one or two metrics.

2. Domain-Aware and Safety-Constrained GenAI Models

Future tools should integrate retrieval-augmented generation, fine-tuning, domain ontologies, and safety constraints to reduce hallucination rates and strengthen correctness.

3. Multi-Agent and Multi-Modal RE Pipelines

Emerging tools like MARE point toward multi-agent collaboration as a promising model. Future research can extend this to workflows combining text, diagrams, models, and code artifacts.

4. End-to-End RE Lifecycle Support

Current tools cluster around early-stage tasks. Research should target underexplored areas including impact assessment, traceability, versioning, and automated change conflict detection.

5. Industry-Scale Validation and Longitudinal Studies

Most studies remain prototype-based. Long-term investigations in real project settings—similar to the automotive case are needed to assess sustained performance, user trust, and integration challenges.

6. Hybrid Human–AI Collaboration Models

Rather than seeking full automation, future work should focus on designing workflows where AI enhances human decision-making, with controlled delegation and verifiable outputs.

7. Prompt Engineering and RE-Specific Instruction Tuning

Given evidence that prompting strategies significantly affect output quality (Ebrahim et al., 2025), research should formalize RE-specific prompting patterns, taxonomies, and fine-tuning strategies.

6. References (APA 7)

Alhoshan, W., Ferrari, A., & Zhao, L. (2025). *How effective are generative large language models in performing requirements classification?* **ACM Transactions on Software Engineering and Methodology**, 00(00), Article 000.

Ataei, M., Cheong, H., Grandi, D., Wang, Y., Morris, N., & Tessier, A. (2024). *Elicitron: An LLM agent-based simulation framework for design requirements elicitation*. arXiv:2404.16045.

Bhatia, S., Gandhi, T., Kumar, D., & Jalote, P. (2024). *System test case design from requirements specifications: Insights and challenges of using ChatGPT*. arXiv:2412.03693.

Ebrahim, M., Guirguis, S., & Basta, C. (2025). *Enhancing software requirements engineering with language models and prompting techniques: Insights from current research and future directions*. In **Proceedings of the ACL Student Research Workshop** (pp. 486–496).

Geyer, W., He, J., Sarkar, D., Brachman, M., Hammond, C., Heins, J., Ashktorab, Z., Rosemberg, C., & Hill, C. (2025). *A case study investigating the role of generative AI in quality evaluations of epics in agile software development*. In **Proceedings of CHIWORK '25**.

Jin, D., Jin, Z., Chen, X., & Wang, C. (2024). *MARE: Multi-agents collaboration framework for requirements engineering*. arXiv:2405.03256.

Kong, Y., Zhang, N., Duan, Z., & Yu, B. (2025). *Collaboration with generative AI to improve requirements change*. **Computer Standards & Interfaces, 94**, 104013.

Lubos, S., Felfernig, A., Tran, T. N. T., Garber, D., El Mansi, M., Polat Erdeniz, S., & Le, V.-M. (2024). *Leveraging LLMs for the quality assurance of software requirements*. arXiv:2408.10886.

Oriol, M., Motger, Q., Marco, J., & Franch, X. (2025). *Multi-agent debate strategies to enhance requirements engineering*. arXiv:2507.05981.

Rahmanpour, M. (2024). *An industrial case study of human and AI-generated system requirements* (Master's thesis). University of Georgia.

Schiller, K. A., Haddad, M.-S., & Seibel, A. (2025). *ReqGPT: A fine-tuned large language model for generating requirements documents*. In **Proceedings of ICED25** (pp. 2741–2750).

Spijkman, T., Molenkamp, B., Beudeker, S., Overbeek, S., & Dalpiaz, F. (2024). *LLM-assisted requirements engineering in Agile model-driven development: Industry insights and validation*. (Preprint).

Uygun, Y., & Momodu, V. (2024). *Local large language models to simplify requirement engineering documents in the automotive industry*. **Production & Manufacturing Research, 12**(1), 2375296. https://doi.org/10.1080/21693277.2024.2375296

Unknown Author. (n.d.). *Untitled requirements engineering research manuscript*. Unpublished manuscript.

Wang, K., Zhang, F., & Sabetzadeh, M. (2024). *Automated requirements demarcation using large language models: An empirical study*. In **REFSQ 2024 Workshops Proceedings**.