An AI-Assisted Tool for Practically Relevant Research Problem Formulation in Software Engineering

João Pedro Theodoro
PUC-Rio
Rio de Janeiro, Brazil
joaotheodoro@aluno.puc-rio.br

Victoria Cruz de Figueiredo
PUC-Rio
Rio de Janeiro, Brazil
victoriacruz@aluno.puc-rio.br

Anrafel Fernandes Pereira
PUC-Rio and Univassouras
Rio de Janeiro, Brazil
afpereira@inf.puc-rio.br

Wallace Albertini
PUC-Rio
Rio de Janeiro, Brazil
wboquimpani@inf.puc-rio.br

Mariana Crisostomo Martins
Universidade Federal de Goiás (UFG)
Goiás, Brazil
maricrisostomo.martins@discente.ufg.br

Marcos Kalinowski
PUC-Rio
Rio de Janeiro, Brazil
kalinowski@inf.puc-rio.br

Abstract

Research problem formulation is an important activity in Software Engineering (SE), as poorly formulated and weakly contextualized problems may reduce the practical relevance of research outcomes. Lean Research Inception (LRI) has been proposed as a structured approach to support problem formulation through collaborative alignment with researchers and practitioners. However, applying LRI often relies on manual activities and workshop-based interactions, which may introduce cognitive effort and limit iterative refinement. To address this challenge, this paper presents LRI Co-Assistant, an AI-assisted tool designed to support the formulation of practically relevant research problems in SE. LRI Co-Assistant operationalizes LRI in an interactive environment by integrating Large Language Models (LLMs) as a reasoning partner to assist in structuring problem attributes, suggesting refinements, identifying gaps, and supporting consistency throughout the formulation process. We demonstrate the tool using a realistic scenario derived from an industry-related research problem. The demonstration shows that LRI Co-Assistant can support a clearer, more consistent, and practice-oriented problem formulation while reducing cognitive effort and promoting critical reflection within the structured LRI workflow. A short video showing LRI Co-Assistant is available at https://doi.org/10.5281/zenodo.21385837.

Keywords

Research Problem Formulation, Artificial Intelligence, Lean Research Inception

1 Introduction

Achieving practical relevance in Software Engineering (SE) research often begins with how the research problem is conceived [17]. However, many studies still exhibit limitations at this starting point, reflecting a recurring gap between academic investigation and industry demands [3, 4, 23]. This misalignment is frequently linked to a lack of clarity, poor contextualization, and a weak connection to real-world scenarios during problem formulation [4]. Although various initiatives have sought to address this challenge through collaborative approaches and industry-relevant frameworks [7, 10, 14, 18], support for researchers remains insufficient at the very stage where fundamental decisions are made: the problem formulation.

This raises a fundamental question: *"How can we more effectively support researchers in formulating research problems that are both scientifically sound and practically relevant?"* To address this challenge, Lean Research Inception (LRI) [16] was introduced as a collaborative framework designed to guide the formulation and initial assessment of research problems with a focus on practical relevance. LRI is grounded in principles from Design Thinking [19], Lean Startup [20], and Lean Inception [2], emphasizing iterative refinement, stakeholder alignment, and contextual understanding. However, these approaches are traditionally based on manual, workshop-driven activities that depend on active facilitation and physical visual artifacts, such as canvases. This can lead to high cognitive load, limited scalability, and difficulties in maintaining consistency or reusing generated knowledge.

Simultaneously, recent advances in Artificial Intelligence (AI), specifically Large Language Models (LLMs), for example LLM-Assisted Empirical Software Engineering [6] and AI agents, such as Google's AI Co-Scientist [8], open new opportunities to support complex cognitive tasks. These technologies have demonstrated significant potential in supporting knowledge synthesis, contextual reasoning, and decision-making, making them promising candidates for strengthening early-stage research activities.

Motivated by this opportunity and the vision presented by Pereira *et al.* [15], this paper presents an AI-assisted tool, called LRI Co-Assistant, designed to support the formulation of practically relevant research problems in SE. LRI Co-Assistant operationalizes the LRI framework within an interactive digital environment, utilizing an AI reasoning partner grounded on Lean Research Inception. Rather than replacing researchers or practitioners, LRI Co-Assistant aims to augment their analytical and reflective capabilities. Specifically, it provides intelligent support for structuring problem attributes, suggesting refinements, and assisting in the initial assessment of research problems regarding their *value*, *feasibility*, and *applicability*.

Unlike general-purpose AI assistants that primarily support writing or ideation, LRI Co-Assistant operationalizes a structured, evidence-based framework for research problem formulation, guiding researchers through the complete LRI workflow.

The contributions of this work are threefold. First, the Design and Implementation: we present an AI reasoning partner specifically engineered to support research problem formulation in Software Engineering. Second, the Operationalization: we operationalize the LRI framework through an interactive and intelligent environment, extending its applicability beyond manual contexts. Finally, the Demonstration and Analysis: we demonstrate the LRI Co-Assistant in a realistic scenario and discuss its potential benefits, limitations, and future directions, contributing to the advancement of AI-supported approaches for early-stage research activities.

2 Background and Related Work

The search for greater practical relevance in Software Engineering (SE) has been extensively discussed in the literature, highlighting the critical need to align research problems with real-world industry contexts [3, 4, 5, 23]. In this scenario, the problem formulation stage plays a central role, as it can directly influence the applicability and impact of the research outcomes. Lean Research Inception (LRI) was proposed as a framework to support this early stage [16]. LRI organizes problem formulation through the Problem Vision board, which consists of seven attributes: *practical problem, context, implications/impacts, stakeholders, evidence, objectives, and research questions* [17]. Furthermore, LRI incorporates a semantic differential scale for the assessment of the problem in terms of *value*, *feasibility*, and *applicability* [16].

LRI was designed to be applied directly by researchers through collaborative workshops involving practitioners with relevant domain expertise aligned with the researcher's interests. Despite its benefits, applying LRI can present challenges for participants, such as the cognitive effort required to fill in the attributes, the need for specialized mediation during workshops, and the difficulty of exploring multiple perspectives on the problem being formulated. In this sense, the integration of Artificial Intelligence, particularly through Large Language Models (LLMs), emerges as a promising opportunity to reduce manual effort and support critical reflection, thereby enriching the research problem formulation process. Recent evidence from Empirical Software Engineering suggests that LLMs have already been successfully applied to support research-related activities, particularly in data processing, analysis, and automation-oriented tasks, demonstrating potential to improve efficiency and scalability while augmenting human activities rather than replacing researchers [6]. These findings suggest that LLMs can also provide valuable support in early-stage research activities, such as research problem formulation.

Canvas-based approaches, Design Thinking frameworks, and agile methodologies have long advocated for tools to support early SE activities, such as requirements elicitation, project discovery, and ideation [13]. More recently, tools integrating LLMs have been proposed to support tasks like discovering disruptive innovations in SE and defining problems in AI innovation projects [13, 21]. These proposals demonstrate benefits such as improved conceptual understanding, the generation of contextual suggestions, and enhanced collaboration among stakeholders. However, existing approaches are not tailored for formulating research problems, do not incorporate structured mechanisms for evaluating practical relevance, and are not integrated into frameworks like Lean Research Inception. Thus, this work distinguishes itself by proposing a tool, called LRI Co-Assistant, that combines a structured approach [16] with AI-based support, specifically focused on formulating research problems with practical relevance.

3 LRI Co-Assistant Overview

3.1 Design Rationale

The LRI Co-Assistant was designed to operationalize Lean Research Inception within an interactive digital environment, incorporating support based on AI. Its main motivation is to transform the process of research problem formulation into a more structured activity, less dependent on manual effort, and richer in terms of reflection and collaboration. The tool adopts the vision of integrating LLMs as a reasoning partner, capable of supporting the researcher throughout the different stages of the process.

3.2 Tool Features

LRI Co-Assistant provides a set of features designed to support researchers throughout the complete Lean Research Inception (LRI) workflow, combining structured guidance with AI-assisted support. The functionalities are organized to reduce cognitive effort, promote reflection, and facilitate the formulation of research problems aligned with practical needs.

Project Creation and Management: researchers can create and manage multiple projects within the platform. Each project stores metadata (e.g., title, creation date, participants, and current phase) and preserves the complete evolution history throughout the LRI process.

Assisted Completion of Problem Vision Attributes: the tool supports the completion of the seven attributes of the Problem Vision board (*practical problem, context, implications/impacts, stakeholders, evidence, objectives*, and *research questions*). Users may manually provide content or receive AI-generated suggestions based on partial descriptions.

Structured AI Reasoning Support: LRI Co-Assistant leverages LLMs as a reasoning partner to generate structured suggestions, identify inconsistencies, enrich contextual understanding, and support reflective problem refinement. Suggestions can be accepted, edited, or discarded, preserving the researcher's central role in decision-making.

Iterative Problem Evolution and Versioning: all modifications performed during the formulation process are stored, allowing users to revisit previous versions, compare refinements, and track the evolution of the research problem over time.

Collaborative Workshop Support: researchers can invite practitioners to participate in collaborative workshops through automatically generated access links. Participants can review and discuss problem attributes without requiring a registered account, facilitating engagement with industry stakeholders.

Research Problem Assessment Support: the tool supports the evaluation of research problems using the LRI semantic differential scale, based on the dimensions of *Value*, *Feasibility*, and *Applicability*. Participants can provide quantitative assessments and complementary observations.

Decision Support for Go/Pivot/Abort: based on participant assessments and workshop discussions, the tool consolidates information to support decision-making regarding whether the research problem should proceed, be reformulated, or be discontinued.

Export and Reporting: LRI Co-Assistant generates a consolidated report containing the formulated research problem, participant assessments, workshop contributions, and decision outcomes.

This artifact can support research protocols, empirical studies, and future collaborative activities.

These features transform LRI from a predominantly manual and workshop-driven activity into a structured and AI-augmented process, enabling researchers to formulate clearer, more consistent, and practically relevant research problems.

3.3 Architecture

The architecture of the LRI Co-Assistant, illustrated in Figure 1, follows a modular model based on distributed services, designed to ensure scalability, responsiveness, and efficient integration with Artificial Intelligence (AI) components. The solution adopts a decoupled full-stack approach, enabling independent evolution of modules and flexibility in integrating with different AI providers.

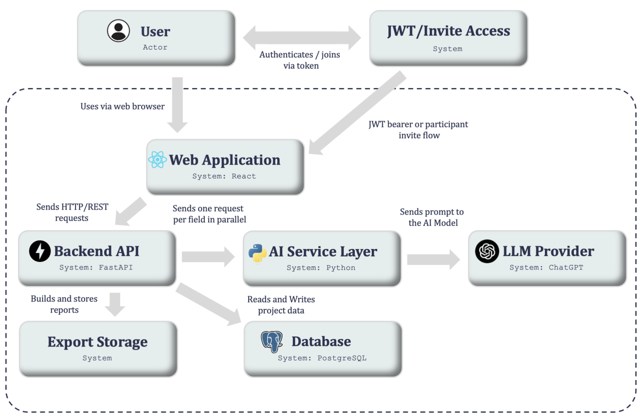

Figure 1: LRI Co-Assistant Architecture

Frontend: implemented as a modern web application (Vercel), the frontend is responsible for the interactive user interface. It enables the structured completion of the seven attributes of the Problem Vision board, as well as the visualization of AI-generated suggestions. The interface is event-driven and communicates with the backend through asynchronous HTTP requests, an architectural decision adopted to preserve interface responsiveness during potentially time, consuming operations, such as LLM calls. This allows users to continue interacting with the tool while suggestions are processed and delivered progressively.

Backend: hosted on the Render platform, the backend acts as the orchestration layer of the application. It manages business logic, session control, data persistence, and integration with the AI module. In addition, it coordinates the execution flow of interactions with the AI, including prompt construction, request handling, and response post-processing, aligned with the different phases of LRI.

AI Reasoning Module: the AI module constitutes the cognitive core of the tool, acting as a reasoning partner throughout the entire problem formulation process (currently GPT 4.1 mini via API). Rather than merely generating content, the module is designed to leverage Large Language Models (LLMs) to support activities such as structuring information, identifying gaps, suggesting multiple perspectives, and refining the internal consistency of the problem. This support is provided in a continuous and contextualized manner, adapting to the current state of the artifact and the active LRI phase, enabling richer interactions aligned with the knowledge construction process.

Database: the persistence layer uses PostgreSQL, hosted on the Neon platform. The database stores research problems, attribute versions, AI interactions, and refinement history, enabling traceability and incremental evolution of the artifacts. This structure allows revisiting decisions, comparing versions, and supporting the iterative and reflective nature of the approach.

3.3.1 *AI Processing Pipeline.*

The operation of the AI module follows a structured, context-aware, and iterative pipeline. It incorporates previously generated artifacts and user interactions to refine results over time, and is composed of the following stages:

Input: the user provides initial information, such as partial descriptions of the Problem Vision board attributes.

Prompt Construction: the backend builds structured prompts based on predefined templates and the context provided by the user.

LLM Invocation: the prompt is sent to the language model, which generates suggestions, refinements, or critical analyses.

Post-processing: the model's response is processed to ensure consistency, structure, and alignment with the format expected by the tool.

Output Delivery: the results are presented to the user through the interface and can be accepted, edited, or discarded, reinforcing the user's role as the central agent in the decision-making process.

This pipeline enables the decoupling of business logic from content generation, facilitating system maintenance and evolution.

3.3.2 *Prompting Strategy.*

The LRI Co-Assistant adopts a hybrid prompting strategy, combining:

- **Structured templates:** prompts based on the seven attributes of the Problem Vision board, ensuring consistency in content generation;
- **Few-shot prompting:** predefined examples are used to guide the model toward generating responses more aligned with the Software Engineering domain;
- **System prompts:** high-level instructions are used to position the model as a "reasoning partner", emphasizing clarity, critical thinking, and alignment with practical relevance;
- **Context injection:** user-provided information and interaction history are incorporated into the prompt to maintain coherence throughout the process.

This hybrid prompting strategy reduces generic responses, preserves consistency across Problem Vision attributes, and enables context-aware reasoning throughout the formulation process.

3.3.3 *Output Representation.*

The outputs generated by the AI module are structured according to the task being performed and mainly include organized descriptions for the Problem Vision board attributes as well as alternative suggestions for refining specific attributes or the formulated research problem. This output diversity provides flexibility in user interaction and supports a more iterative and reflective formulation process.

3.3.4 Technical Limitations. Despite its benefits, integrating LLMs introduces some limitations. Response generation depends on external services, which may introduce latency and affect user experience. In addition, the quality and availability of functionalities are dependent on the underlying AI providers and models. As an inherent characteristic of LLMs, generated outputs may vary across different executions, potentially producing inconsistencies or generic suggestions. Furthermore, despite the use of structured prompts and contextual information, the model's internal reasoning process is not fully deterministic and offers limited controllability. These limitations are partially mitigated through structured prompting, iterative refinement, and continuous human validation throughout the formulation process.

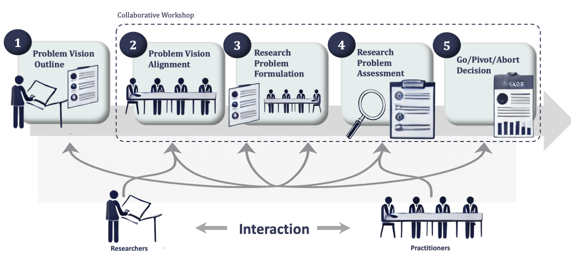

Figure 2: LRI Workflow Supported by LRI Co-Assistant

The LRI Co-Assistant's usage flow is aligned with the five phases of Lean Research Inception, illustrated in Figure 2: *Problem Vision Outline*, *Problem Vision Alignment*, *Research Problem Formulation*, *Research Problem Assessment*, and *Go/Pivot/Abort*. Throughout these phases, the architecture enables continuous support through intelligent suggestions and automated feedback, promoting an iterative and reflection-oriented process.

4 LRI Co-Assistant Demo

This section presents an illustrative proof-of-concept demonstration of how LRI Co-Assistant supports the formulation and assessment of practically relevant research problems through the five phases of LRI. Beyond illustrating the workflow, we emphasize the technical role of AI in transforming user inputs into structured outputs. Across the different phases, AI support follows four operational steps:

User Input Collection: The user provides partial descriptions, keywords, or free-text explanations related to the research problem.

Prompt Generation and Context Construction: The backend dynamically builds prompts using predefined templates aligned with the seven attributes of the Problem Vision board, enriched with contextual information and interaction history. **LLM-based Processing:** The prompt is submitted to the language model, which generates structured suggestions, identifies potential gaps, and proposes refinements.

Validation and Output Processing: Generated content is validated and reformatted before presentation to users.

Validation mechanisms currently include checking for missing attributes while preserving the output structure according to the Problem View attributes. The resulting suggestions are then displayed to users, who can accept, modify, or reject them.

4.1 Context

Software Engineering for AI-based systems has been highlighted in the literature as an area where the gap between industrial practice and research is particularly evident [11, 12], making this type of scenario especially representative for demonstrating the tool.

To illustrate the applicability of the LRI Co-Assistant, we use a real-world research problem [1] derived from a case study, held in December 2025, in which we applied Lean Research Inception [22]. In this study, the problem was collaboratively discussed in a workshop involving a researcher and three professionals with experience in AI and product-related roles, with the goal of refining it and assessing its practical relevance. This makes the case suitable for demonstrating the tool in a realistic scenario, grounded in a research problem that is effectively relevant to industry.

The selected problem addresses the lack of approaches to evaluate and ensure the functional correctness of Artificial Intelligence (AI) systems in industrial contexts, particularly when quality standards such as ISO/IEC 25059 [9] need to be operationalized. In practice, organizations often rely on point estimates, such as accuracy or recall, to assess whether an AI system meets requirements. However, such estimates may hide uncertainty and mask the actual risk associated with model behavior. This limitation can lead to rework, technical debt, high operational costs, and even high failure rates in AI projects.

The problem also involves multiple stakeholders, including AI systems engineering specialists, AI engineers, data scientists, Product Owners, and Business Owners. Therefore, it represents an appropriate example to demonstrate how the LRI Co-Assistant supports the formulation of research problems that require both technical grounding and alignment with practical needs.

4.2 Usage Scenario

The demonstration follows the five phases of LRI (Figure 2) and shows how the LRI Co-Assistant supports the user throughout the formulation and initial assessment of the research problem.

Phase 1 - Problem Vision Outline: The process begins with the researcher creating a project and entering an initial problem description. For example, the researcher provides: *"Current industrial practice evaluates AI systems mainly through point estimates such as accuracy or recall without considering uncertainty or statistical confidence"*. Based on this input, LRI Co-Assistant dynamically generates prompts associated with each Problem Vision attribute. For instance, for the *Practical Problem* attribute, the generated prompt follows the structure: *"Identify the practical problem described below and explain why it may affect industrial environments: [USER DESCRIPTION] Focus on Software Engineering contexts and emphasize practical impacts"*. For the *Stakeholders* attribute: *"Based on the following scenario, identify stakeholders potentially affected by the problem and explain their roles: [USER DESCRIPTION]"*.

After prompt execution, the LLM produces structured outputs such as: **Practical Problem:** Lack of robust approaches for assessing functional correctness of AI systems under uncertainty; **Stakeholders:** AI Engineers, Machine Learning Engineers, Product Owners, Business Owners, Data Scientists; and **Impacts:** Technical debt, hidden risks, increased operational costs, and rework. Figure 3 illustrates the interface for this step.

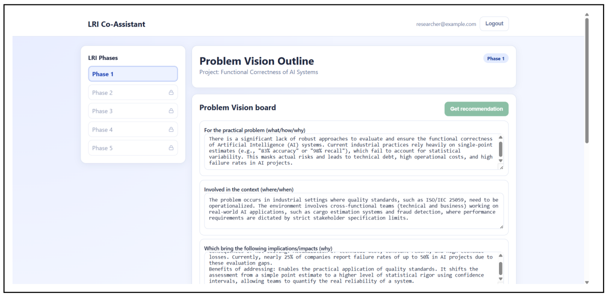

Figure 3: Filling in the attributes of the Problem Vision board.

Based on the suggestions provided, the researcher can incorporate them, taking the opportunity to refine their problem. At the end of this stage, the problem serves as the basis for the next phase. The researcher begins preparing for the collaborative workshop by inviting practitioners to discuss the problem based on their experiences. This preparation for collaboration is illustrated in Figure 4.

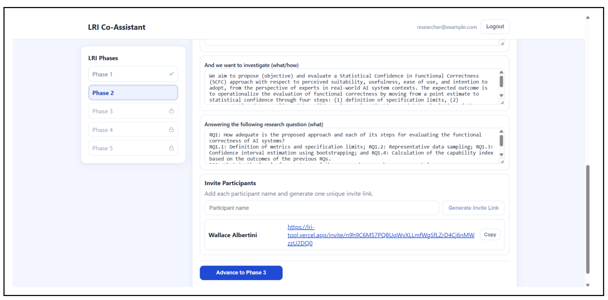

Figure 4: Invitation for Practitioners to collaborate.

Phase 2 - Problem Vision Alignment: In this phase, the researcher conducts a collaborative workshop with practitioners who have relevant experience in the problem domain. Before the workshop begins, invitation links are generated, allowing participants to access the tool and review the previously structured Problem Vision board.

Participants collaboratively discuss and refine the seven problem attributes by contributing their perspectives, experiences, and domain knowledge. This process promotes alignment among different viewpoints and supports the construction of a shared understanding of the research problem.

In the current version of the LRI Co-Assistant, this phase is conducted exclusively through human interaction, without direct AI support, aiming to preserve the natural dynamics of collaboration and participant autonomy. The outcome of this phase is an aligned and refined Problem Vision board, which serves as the foundation for the research problem formulation stage.

Phase 3 - Research Problem Formulation: Once the workshop discussion concludes, LRI Co-Assistant performs a holistic analysis of the collected attributes. At this stage, AI support goes beyond text generation and performs consistency-oriented refinement. Examples include: detecting objectives without corresponding research questions; identifying duplicated research questions; identifying ambiguous terms; suggesting stronger relationships between practical problems and expected impacts; proposing alternative formulations.

For example, the initial research question: *"How can AI correctness be improved?"* may be refined into: *"How can statistical confidence mechanisms support the evaluation of functional correctness in AI systems operating under industrial quality standards?"* This refinement improves specificity, contextual grounding, and alignment with the research objectives. Figure 5 illustrates the consolidation of the research problem.

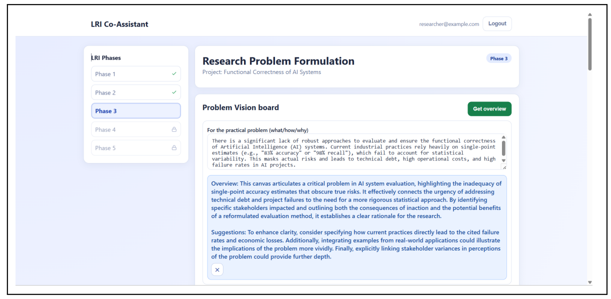

Figure 5: Formulated Research Problem.

Phase 4 - Research Problem Assessment: In this phase, the formulated research problem is evaluated in terms of its practical relevance using the LRI semantic differential scale, based on the criteria: *Value*, *Feasibility*, and *Applicability*. Each participant performs the evaluation individually, assigning scores on a seven-point scale, which allows capturing different perceptions regarding the problem's potential in real-world contexts. Additionally, participants may use an observation field to provide comments or justifications that complement their evaluation.

Figure 6 presents the evaluation process of the research problem.

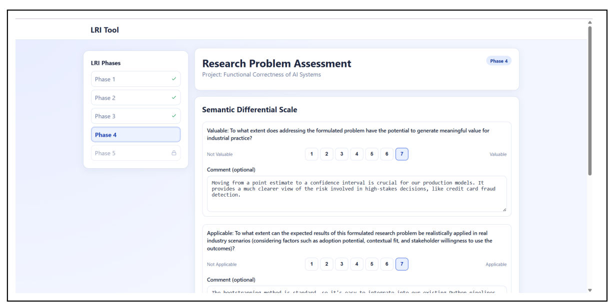

Figure 6: Evaluation of the Formulated Research Problems

In the current version, the evaluation process is conducted without AI support. This decision keeps the assessment centered on participants' perceptions, avoiding biases introduced by automated suggestions. As future work, we intend to incorporate AI-based mechanisms to support participants in this stage by providing justified evaluations and multiple perspectives, aiming to stimulate more critical, consistent, and well-grounded reflection.

Phase 5 - Go/Pivot/Abort Decision: After collecting participant assessments, the system automatically aggregates responses and generates a summary report. The report includes: consolidated Problem Vision attributes; median scores for *Value*, *Feasibility*, and *Applicability*; participant observations; AI-generated synthesis of discussion points; and suggested decision rationale. For example, **Decision suggestion:** GO; **Reasoning:** High perceived practical value among stakeholders and strong alignment between identified industrial challenges and research objectives. The final decision remains under researcher control. Figure 7 illustrates the group decision in this scenario.

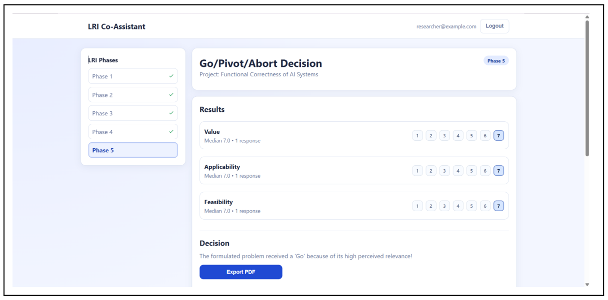

Figure 7: Phase 5 - Group Decision

The LRI Co-Assistant allows exporting the final artifact for use in research protocols, empirical studies, and future collaborative workshops. At the end of the process, the researcher obtains a structured, refined, and evaluated research problem. Compared to the initial unstructured description [16], the resulting formulation makes explicit key elements such as context, stakeholders, evidence, impacts, objectives, and research questions. This demonstration indicates that LRI Co-Assistant can extend the Lean Research Inception workflow through AI support, helping reduce cognitive effort, improve clarity, and prepare research problems for collaborative refinement and practical relevance assessment.

4.3 Demonstration Results and Implications

The demonstration illustrates how LRI Co-Assistant supports the transformation of an initially problem description into a more structured and practice-oriented research problem. Throughout the LRI phases, the tool contributed to three main outcomes.

First, the tool supported the **structuring of problem attributes**. Starting from a free-text problem description, AI-generated suggestions helped make explicit key elements of the Problem Vision board, including the *practical problem, context, stakeholders, evidence, implications/impacts, objectives, and research questions.*

Second, the tool enabled **iterative refinement and consistency improvement**. During the formulation process, AI support helped identify missing information, suggest alternative formulations, and strengthen relationships among problem elements. In the presented scenario, this resulted in research questions that were more specific, contextualized, and aligned with the identified objectives and industrial needs.

Third, the tool facilitated **preparation for communication and assessment**. By organizing information into a structured representation, the resulting artifact supported collaborative discussion among participants and enabled assessment using the LRI semantic differential scale.

The demonstration also highlights practical implications of integrating AI into research problem formulation activities. Rather than acting only as a content generator, LRI Co-Assistant operates as a reasoning partner capable of supporting reflection, reducing manual effort, and improving problem clarity while preserving the researcher as the central decision-maker.

Although the demonstration provides evidence of the feasibility and potential usefulness of the approach, some limitations should be considered. The quality of generated suggestions depends on the underlying language model and may introduce variability or generic outputs. Additionally, AI support still relies on iterative interaction and does not replace human judgment. Future work includes empirical evaluations with researchers and practitioners to better understand the tool's impact in real-world contexts.

5 Conclusion and Future Work

This paper presented LRI Co-Assistant, an AI reasoning partner designed to support the formulation of practically relevant research problems in Software Engineering. The tool operationalizes Lean Research Inception (LRI) in an interactive environment by combining structured guidance with AI-based support throughout the problem formulation process. LRI Co-Assistant extends traditional workshop-based activities by supporting the completion and refinement of Problem Vision attributes, enabling collaborative interactions among stakeholders, and assisting in the initial assessment of research problems through the LRI criteria of *value*, *feasibility*, and *applicability*. The presented demonstration showed how the tool can support the transition from an initially unstructured problem description to a more structured, refined, and practice-oriented research problem.

The proposed approach explores AI beyond content generation, positioning it as a reasoning partner capable of supporting reflection, identifying gaps, and improving consistency during early-stage research activities while preserving the researcher as the central decision-maker. Future work includes conducting empirical evaluations with researchers and practitioners to investigate usability, perceived usefulness, and impact on research problem quality. We also intend to explore multi-agent architectures, improve the explainability of AI-generated suggestions, and extend the tool with additional mechanisms to support collaboration and decision-making throughout the LRI process.

Artifact Availability

The artifact package is available on https://doi.org/10.5281/zenodo.21385837. The source code is licensed under MIT, while the remaining materials use CC BY 4.0. An online demo is available at https://lri-tool.vercel.app using `researcher@example.com` / `researcher123`.

Acknowledgements

We gratefully acknowledge the support of CAPES (Finance Code 001 and PROEX), CNPq (Grants 312275/2023-4 and 420191/2025-9), FAPERJ (Grant E-26/204.256/2024), and Instituto Kunumi for their generous support.

References

[1] Wallace Albertini, Marina Condé Araújo, Júlia Condé Araújo, Antonio Pedro Santos Alves, and Marcos Kalinowski. 2026. Statistical confidence in functional correctness: an approach for ai product functional correctness evaluation. In *5th International Conference on AI Engineering – Software Engineering for AI (CAIN@ICESE'26)*. ACM, Rio de Janeiro, Brazil.

[2] Paulo Caroli. 2017. Lean inception. *São Paulo, BR: Caroli. org*.

[3] Xavier Franch, Daniel Mendez, Andreas Vogelsang, Rogardt Heldal, Eric Knauss, Marc Oriol, Guilherme H Travassos, Jeffrey C Carver, and Thomas Zimmermann. 2020. How do practitioners perceive the relevance of requirements engineering research? *IEEE Transactions on Software Engineering*, 48, 6, 1947–1964.

[4] Vahid Garousi, Markus Borg, and Markku Oivo. 2020. Practical relevance of software engineering research: synthesizing the community's voice. *Empirical Software Engineering*, 25, 1687–1754.

[5] Vahid Garousi, Kai Petersen, and Baris Ozkan. 2016. Challenges and best practices in industry-academia collaborations in software engineering: a systematic literature review. *Information and Software Technology*, 79, 106–127.

[6] Victoria Gomes, Delaney Selb, Fabio Palomba, Rodrigo Spinola, and David Lo. 2026. Llm-assisted empirical software engineering: systematic literature review and research agenda. *arXiv preprint arXiv:2604.26192*.

[7] Tony Gorschek and Daniel Mendez. 2021. Solving problems or enabling problem-solving? from purity in empirical software engineering to effective co-production (invited keynote). In *Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference*. Springer, Cham, Switzerland, 109–116. doi:10.1007/978-3-030-65854-0_9.

[8] Juraj Gottweis et al. 2025. Towards an ai co-scientist. *preprint arXiv:2502.18864*.

[9] International Organization for Standardization. 2023. ISO/IEC 25059:2023 Software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality model for AI systems. Geneva, Switzerland, (2023). https://www.iso.org/standard/80655.html.

[10] Martin Ivarsson and Tony Gorschek. 2011. A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, 16, 365–395.

[11] Marcos Kalinowski et al. 2025. Naming the pain in machine learning-enabled systems engineering. *Information and Software Technology*, 187, 107866. doi:https://doi.org/10.1016/j.infsof.2025.107866.

[12] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, and Stefan Wagner. 2022. Software engineering for ai-based systems: a survey. *ACM Transactions on Software Engineering and Methodology*, 31, 2, 1–59.

[13] Mariana Martins, Lucas Rocha, Lucas Romão, Taciana Kudo, Marcos Kalinowski, and Renato Bulcão-Neto. 2025. Dip-ai: a discovery framework for ai innovation projects. In *Anais do XXIV Simpósio Brasileiro de Qualidade de Software*. SBC, São José dos Campos/SP, 131–141. doi:10.5753/sbqs.2025.13834.

[14] Jefferson Seide Molléri, Emilia Mendes, Kai Petersen, and Michael Felderer. 2023. Determining a core view of research quality in empirical software engineering. *Computer Standards & Interfaces*, 84, 103688.

[15] Anrafel Fernandes Pereira, Maria Teresa Baldassarre, Daniel Mendez, and Marcos Kalinowski. 2026. Towards ai agents supported research problem formulation. In *3rd International Workshop on Methodological Issues with Empirical Studies in Software Engineering (WSESE@ICESE'26)*. ACM, Rio de Janeiro, Brazil.

[16] Anrafel Fernandes Pereira, Marcos Kalinowski, Maria Teresa Baldassarre, Jürgen Börstler, Nauman bin Ali, and Daniel Mendez. 2025. Towards lean research inception: assessing practical relevance of formulated research problems. In *Evaluation and Assessment in Software Engineering (EASE '25)*. ACM, Istanbul, Turkiye.

[17] Anrafel Fernandes Pereira et al. 2026. Attributes to support the formulation of practically relevant research problems in software engineering. In *3rd International Workshop on Methodological Issues with Empirical Studies in Software Engineering (WSESE@ICESE'26)*. ACM, Rio de Janeiro, Brazil.

[18] Kai Petersen, Jürgen Börstler, Nauman Bin Ali, and Emelie Engström. 2024. Revisiting the construct and assessment of industrial relevance in software engineering research. In *Proceedings of the 1st IEEE/ACM International Workshop on Methodological Issues with Empirical Studies in Software Engineering*, 17–20.

[19] Hasso Plattner, Christoph Meinel, and Ulrich Weinberg. 2009. *Design thinking*. Springer.

[20] Eric Ries. 2011. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Currency.

[21] Margaret-Anne Storey, Daniel Russo, Nicole Novielli, Takashi Kobayashi, and Dong Wang. 2024. A disruptive research playbook for studying disruptive innovations. *ACM Transactions on Software Engineering and Methodology*, 33, 8, 1–29.

[22] João Pedro Theodoro, Victoria Cruz Figueiredo, Anrafel Fernandes Pereira, Wallace Albertini, Mariana Crisostomo Martins, and Marcos Kalinowski. Lri co-assistant: artifact package for an ai-assisted tool for practically relevant research problem formulation (sbes 2026) [computer software]. Zenodo, (July 2026). doi:10.5281/zenodo.21385837.

[23] Titus Winters. 2024. Thoughts on applicability. *Journal of Systems and Software*, 215, 112086.