Toward a Framework for Explainability in Intelligent Systems: Preliminary Results

Lívia Mancine[1,2][0000−0002−4123−4861], Renata Dutra Braga[2][0000−0002−3448−0343], and Renato F. Bulcão-Neto[2][0000−0001−8604−0019]

[1] Instituto Federal Goiano (IF Goiano). Ceres - GO, Brazil
livia.mancine@ifgoiano.edu.br
[2] Instituto de Informática, Universidade Federal de Goiás. Goiânia - GO, Brazil
{renatadbraga,rbulcao}@ufg.br

Abstract. The widespread adoption of Intelligent Systems has highlighted the need for explainability as a key non-functional requirement to promote transparency, trust, and user acceptance. Despite its relevance, existing Requirements Engineering (RE) practices still provide limited support for systematically addressing explainability, especially in early development stages. This work aims to integrate explainability into RE through a systematic and process-oriented approach. The paper presents OpenUPExp, an extension of the Open Unified Process that introduces explicit roles, activities, and artifacts for explainability requirements. The framework structures explainability concerns across RE activities and supports their progressive refinement throughout development. An initial fragment of the framework was evaluated through an illustrative scenario analyzed by RE specialists to assess coherence and artifact clarity. The evaluation indicates that OpenUPExp provides useful guidance for incorporating explainability into RE, while also revealing aspects that require further refinement. The findings suggest the potential of the framework to support a more systematic treatment of explainability in Intelligent Systems.

Keywords: Requirements · Explainability · Intelligent Systems · OpenUP.

1 Introduction

The increasing use of Intelligent Systems to support or automate decision-making has intensified the demand for mechanisms that enable stakeholders to understand and justify system outputs. These systems often operate with a high degree of autonomy and complexity, making their behavior difficult to interpret. In this context, explainability emerges as a critical Non-Functional Requirement (NFR), defined as a system's ability to provide intelligible and contextually meaningful information about its decision-making processes, directly impacting transparency, trust, and user acceptance [6].

In high-stakes domains, such as healthcare and defense, the demand for explanations is further amplified due to potential societal impacts and strict regulatory

Proceedings of the 29th Workshop on Requirements Engineering (WER2026), August 11-13, 2026, La Plata, Argentina.
https://doi.org/10.29327/1884517.29-8

requirements. National and international guidelines, such as the Brazilian Artificial Intelligence Plan and the Proposed Law No. 2.338/2023, emphasize that intelligent systems must be explainable, auditable, and transparent to ensure accountability in their use [21,19].

Although Requirements Engineering (RE) plays a fundamental role in incorporating explainability requirements [13], the literature indicates limited practical support for this purpose [18]. In particular, there is a lack of systematic approaches to address explainability from early development stages, including identifying what should be explained, to whom, and in which contexts.

To address this gap, this paper proposes OpenUPExp, a framework for systematically treating explainability requirements in Intelligent Systems from early development stages. OpenUPExp extends the Open Unified Process (OpenUP) [3] — a lightweight version of the Rational Unified Process (RUP) [4] — by incorporating explainability-related roles, activities, and artifacts across the inception and elaboration phases.

This research follows the Design Science (DS) paradigm [16]. An early-stage conceptual evaluation was conducted through an exploratory study with RE specialists, using an illustrative scenario to simulate the application of the framework. The results indicate that OpenUPExp provides structured support for integrating explainability, while also revealing opportunities for further refinement. As a contribution, this work delivers an operational framework for RE in Intelligent Systems and an initial assessment of its applicability.

The remainder of this paper is organized as follows. Sections 2 and 3 present the background and related work. Section 4 describes the proposed approach, while Sections 5, 6, and 7 present the evaluation, results, and threats to validity. Finally, Section 8 concludes the paper.

2 Background

Explainability has been recognized as a relevant requirement within RE. Chazette et al. [1] define a system as explainable with respect to an aspect X if a corpus of information is provided to an addressee A such that A is able to understand X. In practice, explainability involves providing stakeholders with information that enables them to understand, justify, validate, contest, or audit AI-supported decisions according to their roles and responsibilities within a given context. Such information may include decision criteria, influencing factors, similarity measures, rationale associated with recommendations, or traceability of system outputs. From an RE perspective, these informational needs should be translated into requirements that define what should be explained, to whom, and under which conditions.

Existing research on explainability has been largely driven by Explainable Artificial Intelligence (XAI) [23], focusing on model interpretability. However, these approaches are typically centered on technical explanations and are often

[3] `https://shre.ink/OpenUP`
[4] `https://shre.ink/RUP`

not accessible to non-expert users, creating a gap between model-level interpretability and system-level understanding. This limitation highlights the need to address explainability from a broader, user-centered perspective within RE.

Explainability by Design (EbD) addresses this limitation by advocating that explainability should be considered from the early stages of system conception, rather than introduced as an afterthought [25,12]. This perspective aligns with RE principles regarding the early treatment of NFRs and emphasizes both the explicit definition of explainability as a first-class concern and the need for tailoring explanations to stakeholders' informational needs.

From a process perspective, the Open Unified Process (OpenUP) provides a lightweight, iterative, and incremental framework for software development [7]. It emphasizes collaboration, risk management, and progressive refinement of requirements, while remaining adaptable to different project contexts. However, OpenUP does not explicitly address explainability as a concern within its process structure. Although explainability has been recognized as an NFR and EbD advocates its early consideration, there is still a lack of approaches that systematically integrate explainability into RE processes. This gap motivates the need for structured solutions that implementation explainability from early stages of development, such as the framework proposed in this work.

3 Related Work

The literature on explainability in RE for Intelligent Systems has evolved in the form of conceptual models, requirement catalogs, frameworks, guidelines, and tools. However, these contributions remain largely isolated, addressing specific aspects of explainability without explicit integration into an RE process.

Regarding the **conceptualization of explainability and its relation to quality attributes**, Chazette et al. [2] propose a conceptual model, a knowledge catalog, and a reference model for explainable systems. Köhl et al. [14] present a requirement catalog grounded in psychology and cognition. Habiba et al. [9] and Crook et al. [3] introduce frameworks that address user-centeredness, transparency, and trade-offs with performance. While relevant, these contributions remain predominantly at a conceptual or solution-proposal level.

In terms of **artifacts, tools, and support for specific RE activities**, Zöller et al. [29] present XAutoML, a tool focused on visual transparency in AutoML. Ligocka [15] proposes a documentation model for explainability requirements, while Schoonderwoerd et al. [20] address elicitation and validation in an industrial case, proposing requirements and UI design patterns for explainable clinical systems. Speith and Langer [22] contribute evaluation criteria for explanations within the XAI process. Although relevant, these approaches are focused on specific artifacts, tools, or domains, lacking a comprehensive process-level perspective.

From a **process and maturity perspective**, empirical studies such as Habibullah et al. [10], Haindl et al. [11], and Vogelsang and Borg [27] show that industry still prioritizes performance, while explainability remains weakly

structured in practice. Other works, including d'Aloisio [4], Umm eHabiba [26], and Franch et al. [8], propose frameworks and guidelines for AI-based systems, emphasizing the need to incorporate NFRs such as explainability, trust, and auditability. However, these studies do not provide a detailed RE process extension with explicitly defined roles, tasks, and artifacts.

Overall, related work indicates that explainability is widely recognized as a relevant concern in RE, yet existing solutions remain fragmented. There is a lack of approaches that explicitly integrate explainability into a **process-oriented RE framework**, including the definition of **roles, tasks, and artifacts** from early development stages. This gap motivates the proposal of *OpenUPExp*.

4 The Proposed Approach

This work adopts the DS paradigm due to its suitability for constructing and evaluating artifacts aimed at solving relevant problems. The adopted DS model [24] is structured into three cycles: problem understanding, artifact design, and evaluation. Based on this model, we propose OpenUPExp, a framework that extends OpenUP to systematically integrate explainability into RE. OpenUP was selected due to its iterative and incremental nature, supporting the progressive evolution of requirements. OpenUPExp introduces explainability-related roles, tasks, and artifacts, mainly within the Inception and Elaboration phases. Inception focuses on defining the Explainability Vision, while Elaboration addresses the refinement of explainability requirements. Figure 1 presents an overview of OpenUPExp.

4.1 OpenUPExp Framework

OpenUPExp extends the original OpenUP process across two fundamental layers: *Method Content* and *Process*.

Method Content: this layer defines static elements, including roles, tasks, and artifacts. The central role is the Requirements Engineer, responsible for conducting explainability-related activities. Among the generated artifacts, the Explainability Vision consolidates the explainability problem, stakeholders, informational needs, and the scope of what should be explained by the system.

Process: this layer organizes the execution of activities within the OpenUP requirements discipline. It introduces tasks such as *Define Explainability Vision* (focus of this work), *Detail Explainability Requirements*, and *Find and Describe Explainability Requirements*. These tasks follow the EbD principle, addressing explainability from early development stages. In this study, we focus on the Explainability Vision artifact, illustrated in Figure 2. The process is iterative and incremental, allowing continuous refinement of artifacts.

4.2 Explainability Vision Process Modeling

The process for defining explainability in OpenUPExp was modeled using Business Process Model and Notation 2.0 (BPMN), with the goal of explicitly repre-

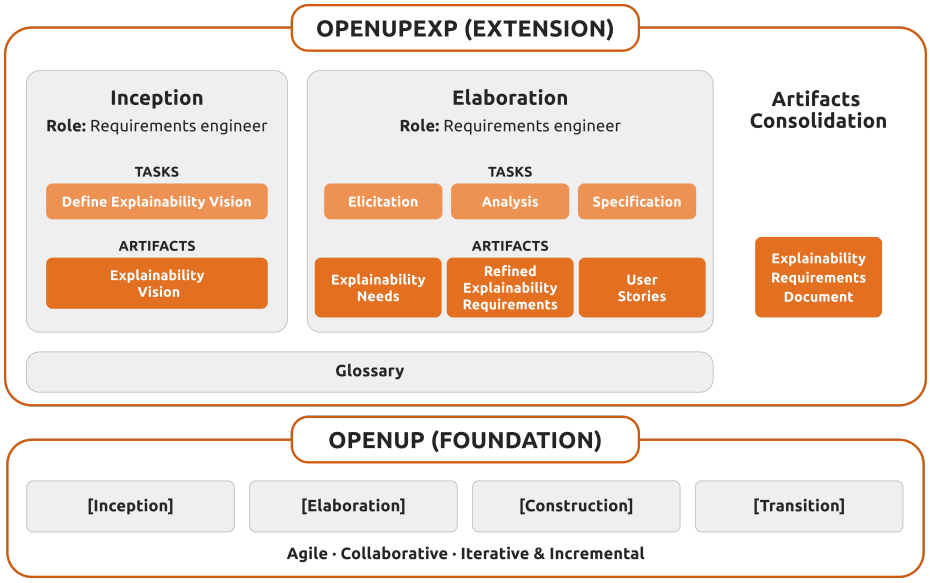

Fig. 1. Overview of the extended OpenUPExp and its main features.

senting the flow of activities, decisions, and artifacts involved in constructing the Explainability Vision (see Figure 3). This work specifically focuses on the Inception phase of OpenUPExp, emphasizing the construction of the Explainability Vision as a high-level artifact to support the understanding of explainability concerns, stakeholders, and contextual needs.

The process begins with the recognition of the need for explainability in the system context, using the project scope as the primary input. From this point, the Requirements Engineer follows a structured sequence of activities, including: (i) defining the explainability problem and associated risks, (ii) identifying stakeholders who require explanations, (iii) defining stakeholder positioning, and (iv) describing the usage environment and user characteristics.

In accordance with BPMN 2.0 semantics, the activities were modeled as simple tasks, since each one produces a specific portion of the Explainability Vision artifact and does not require internal repetition until a stopping condition is met. The possibility of revisiting previous activities, for instance when stakeholders have not been sufficiently identified, is represented through exclusive gateways and feedback flows.

The model also includes a decision point regarding stakeholder identification, allowing previous steps to be revisited within a single iteration. This mechanism supports the progressive refinement of the information collected during one execution of the process. The iterative nature of OpenUPExp, in turn, occurs at a higher level: the Define Explainability Vision task as a whole may be re-executed across successive iterations of the Inception phase, enabling the incremental evolution of the scope, stakeholders, needs, and requirements.

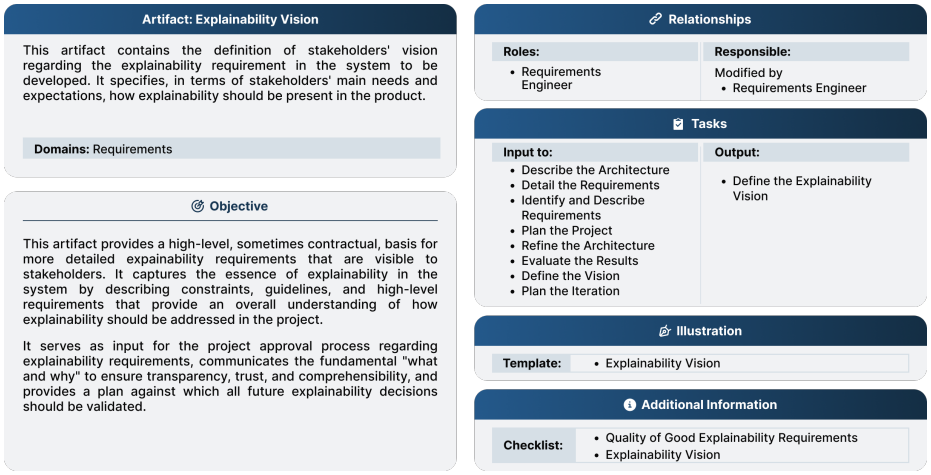

Fig. 2. Explainability Vision Artifact.

The outcomes of these activities are consolidated into two main artifacts: (i) the *Explainability Vision Document*, which organizes stakeholder, context, and explainability needs information, and (ii) the *Consolidated Explainability Requirements Document*, conceived as an integrative artifact that, in the complete framework, will aggregate the Explainability Vision and artifacts from subsequent analysis and specification stages.

In addition to the main activities, the model incorporates external inputs, such as norms, regulations, and institutional guidelines, which directly influence the definition of explainability requirements.

Note that Figure 3 does not represent the full iterative lifecycle of OpenUPExp. Instead, it models the intra-iteration workflow for defining the Explainability Vision within the Inception phase. The iterative and incremental nature of OpenUPExp is expressed at the process lifecycle level (cf. Figure 1): in subsequent iterations, the same workflow may be revisited to refine the scope, include new stakeholders, revise explainability needs, or adjust requirements according to new domain knowledge. Therefore, iteration is not modeled as repeated execution of every BPMN task, but as the possibility of revisiting and refining artifacts across OpenUPExp iterations.

Although the modeled process conceptually reflects the iterative and incremental nature of OpenUPExp, the implementation adopted in this work executed the activities sequentially, without directly incorporating iteration and feedback cycles. This simplification allowed the evaluation to focus on artifact coherence and information traceability throughout the process.

4.3 Tool Support

To support the implementation of the OpenUPExp process, a pipeline-based architecture composed of specialized components, referred to as workers, was

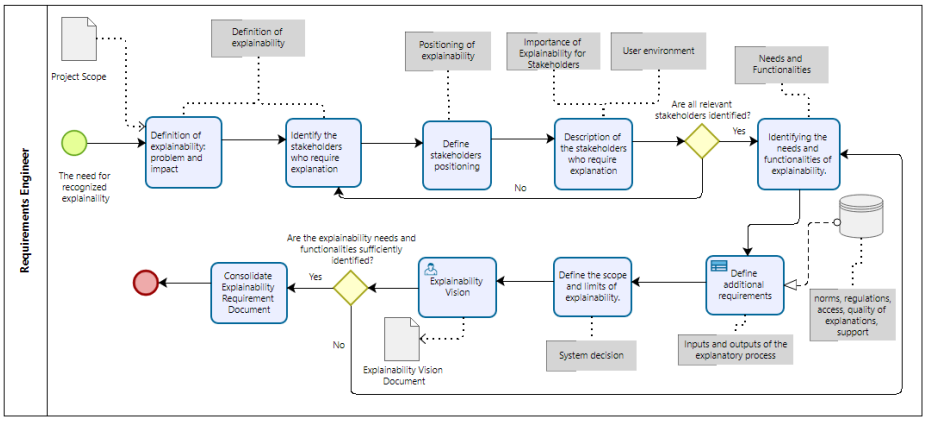

Fig. 3. Explainability Vision Process Modeling using BPMN

implemented using the Langflow platform. Each worker operationalizes a specific activity of the Explainability Vision process and produces structured artifacts consumed by subsequent workers.

Figure 4 presents the current implementation of OpenUPExp as a sequential worker-based pipeline. Although OpenUPExp is conceptually aligned with the iterative and incremental principles of OpenUP, the current implementation does not automate iteration or feedback loops during execution. Refinement is performed externally by the Requirements Engineer, who may revise inputs, prompts, or intermediate artifacts and re-execute parts of the pipeline. This limitation is acknowledged as part of the current maturity stage of the framework and will be addressed in future versions.

From an RE perspective, this pipeline operationalizes the main activities of the explainability definition process, including: (i) problem characterization, (ii) explainability positioning, (iii) stakeholder and context description, (iv) identification of needs, (v) refinement of requirements, and (vi) consolidation of artifacts. Each stage produces structured outputs that serve as inputs for subsequent stages, ensuring traceability and consistency across artifacts.

The architecture incorporates domain adaptation mechanisms, allowing contextual information provided by a requirements enginner to be normalized before being processed by subsequent workers. It also integrates external knowledge sources (e.g., norms and regulatory guidelines) to support requirements generation aligned with the application domain.

As illustrated in Figure 4, the pipeline combines sequential workers, domain adaptation mechanisms, and external knowledge sources. Workers 1 and 2 focus on problem characterization and stakeholder positioning, Workers 3 and 4 refine contextual and explainability needs information, while Workers 5 and 6 incorporate regulatory constraints and define explainability scope and limits. This modular organization enables independent analysis and refinement of each component. Furthermore, the explicit chaining of inputs and outputs facilitates the

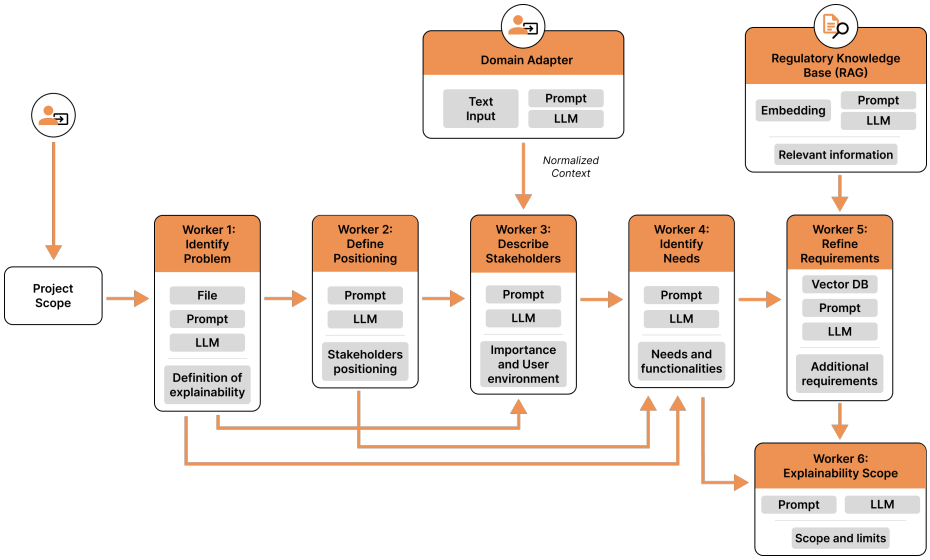

Fig. 4. Pipeline architecture of OpenUPExp, illustrating the interaction among workers, domain adaptation mechanisms, and external knowledge sources.

inspection of intermediate results, which is relevant in early-stage DS evaluations. In its current version, the pipeline execution remains sequential, requiring external refinement and re-execution by the Requirements Engineer whenever adjustments to prompts, inputs, or intermediate artifacts are necessary.

4.4 Setup and Configuration

The setup process configures the execution environment of the OpenUPExp architecture in Langflow[5], ensuring controlled conditions for artifact generation. This step defines execution parameters and domain context, and may be revisited as prompts or input information are refined.

The language model parametrization was defined to prioritize consistency and reproducibility. The Groq provider was used, integrated with Langflow, with a temperature setting of 0.1 to reduce response variability and minimize hallucination risks. The workers presented in Figure 4 were defined as specialized RE agents, with explicit output constraints, including structured formats such as JSON (Workers 1 and 2) and tabular outputs (Workers 3 to 6), ensuring interoperability across the pipeline.

Domain configuration is progressively introduced from the project scope and refined by a domain adaptation mechanism (Domain Adapter), which normalizes context before propagation. Additionally, the architecture incorporates a Retrieval-Augmented Generation (RAG) mechanism to integrate normative and

[5] Available at `https://www.langflow.org/`.

regulatory knowledge, ensuring alignment of generated requirements with domain constraints.

Validation, at this early stage, is conducted through manual inspection of the generated artifacts, focusing on output coherence and consistency of data flow across workers. The use of structured outputs facilitates traceability and helps mitigate inconsistencies in artifact generation.

5 Evaluation

The evaluation of this work is grounded in the DS paradigm and characterized as an early-stage conceptual evaluation. Its goal was to investigate the applicability of the proposed approach in an illustrative scenario, constructed based on a realistic domain, enabling a systematic analysis of the generated artifacts, as well as the coherence of the framework, the clarity of the produced artifacts, and the identification of gaps and inconsistencies.

The evaluation study was conducted with two requirements engineering specialists in March 2026, on individual sessions lasting approximately two hours. During these sessions, participants evaluated the approach in a structured manner, analyzing each stage of the pipeline. To support the evaluation, a guide containing instructions, supporting materials, and feedback collection instruments was provided. Further details are available in an online repository [17].

5.1 Evaluation Design

The evaluation was designed to familiarize participants with the proposed approach and was organized into three main stages.

Stage 1 – Introduction: the researcher presented an overview of the approach through a guide containing the study context and the materials required for the evaluation. These materials included: the illustrative scenario, a video presenting the operationalization of the approach, the scenario scope (input to Worker 1), and the knowledge base used in Worker 5 (RAG mechanism).

Stage 2 – Application: participants were invited to evaluate the approach based on the illustrative scenario. The evaluation could be conducted in two ways: (i) direct inspection in Langflow, allowing navigation through components and execution of the flow; or (ii) static inspection, through a document containing the logic of the workers, the structured prompts, and the generated outputs.

Stage 3 – Evaluation: participants provided feedback through a questionnaire adapted from the Technology Acceptance Model (TAM3) [5]. The instrument evaluated four dimensions: Job Relevance (R), related to the perceived applicability of the approach to work activities; Output Quality (Q), concerning the quality and usefulness of the generated artifacts; Result Demonstrability (D), associated with the clarity and communicability of results; and Anticipated Enjoyment (E), referring to the perceived willingness to adopt the approach. Responses were recorded using a five-point Likert scale. Additionally, an open-ended question was included to collect observations and suggestions.

5.2 Evaluation Application

Two specialists participated in the evaluation, both researchers affiliated with a higher education institution and with professional experience in the software industry. They have more than 15 years of experience in the development of traditional software and intelligent systems, particularly in the healthcare domain, as well as a strong background in Software Engineering and Requirements Engineering, with experience in elicitation, specification, and project leadership, acting as an interface between product and development teams.

The evaluation was conducted through the analysis of a previously constructed illustrative scenario, based on the artifacts generated by applying the proposed approach. Participants were provided with supporting materials, including the scenario, the produced artifacts, and the architectural description.

Both participants performed a static inspection of OpenUPExp due to difficulties accessing the Langflow environment directly. During the evaluation, they analyzed the process structure, the artifact flow, and the coherence of information across the pipeline, focusing on the relationships between workers and the mapping of inputs and outputs. As this was an indirect evaluation based on a pre-constructed scenario, participants did not execute the process end-to-end. Consequently, aspects related to practical execution, such as effort, operational challenges, and adaptability to different contexts, were not assessed at this stage.

6 Preliminary Results

The quantitative results of the preliminary evaluation are presented in Table 1. Overall, no responses indicating strong disagreement were observed, suggesting a positive perception of the proposed approach.

In the Job Relevance construct, both participants evaluated the framework as relevant to activities related to explainability (R1) and considered the artifacts useful for understanding the domain (R3). The applicability in their work context (R2) was also positively evaluated, although one participant indicated a slightly lower level. Regarding Output Quality, there was agreement on the quality of the artifacts (Q4) and on the ability of the process to produce consistent results (Q5). The sequence of activities (Q6) was considered appropriate for generating structured artifacts, suggesting support for requirements-related activities. In the Result Demonstrability construct, participants indicated that the benefits of the framework are clearly perceivable (D7) and that the results are tangible and understandable (D9). Communicability (D8) was rated more moderately, indicating an opportunity for improvement. In Anticipated Enjoyment, the results indicate a favorable perception regarding the adoption of OpenUP4Exp (E10), reinforcing its potential for use.

Qualitative responses complement the quantitative results. Participant A highlighted the need for consistency verification mechanisms across artifacts generated by different workers, aiming to ensure alignment between stakeholders, needs, and requirements. The participant also suggested refining intermediate artifacts with greater specificity to facilitate technical implementation, stakeholder

Table 1. Evaluation Results Based on Likert Scale

ID	Questions	A	B
R1.	OpenUP4Exp appears relevant to activities related to explainability.	Strongly Agree	Strongly Agree
R2.	The activities presented in the scenario seem applicable to my work context.	Agree	Strongly Agree
R3.	The artifacts produced in the scenario are useful to support the understanding of explainability.	Strongly Agree	Strongly Agree
Q4.	The artifacts presented (generated by each activity/worker) exhibit good quality.	Agree	Agree
Q5.	The process described in the scenario suggests the production of consistent results.	Strongly Agree	Strongly Agree
Q6.	The sequence of activities indicates that the framework can generate structured and appropriate artifacts.	Strongly Agree	Strongly Agree
D7.	I can clearly perceive the benefits of OpenUP4Exp.	Strongly Agree	Strongly Agree
D8.	The results of the framework are easy to communicate to other professionals.	Agree	Agree
D9.	The results presented in the scenario are tangible and understandable.	Strongly Agree	Strongly Agree
E10.	I would consider the use of OpenUP4Exp a positive experience in my work context.	Agree	Strongly Agree

validation, and traceability. Regarding the communication of results, the participant recommended including visual representations and executive summaries. Additionally, the importance of empirical evaluations and the availability of supporting materials to facilitate adoption were emphasized.

Participant B highlighted the need for greater clarity in representing data flows and dependencies between workers, as well as the inclusion of more detailed architectural diagrams. Issues related to the use of LLM and RAG were also raised, including exception handling, hallucination control, and operational limitations. The participant also identified gaps in the documentation of component orchestration, suggesting clearer specification of the roles of different elements (e.g., workers, orchestrator) and coordination mechanisms. These aspects reinforce that the framework is still in an early stage of maturity.

Overall, the results indicate that OpenUP4Exp is perceived as a promising approach with potential for practical application, although it still requires refinements to increase its maturity, particularly in terms of validation, communicability, and technical details. These findings provide initial evidence that OpenUP4Exp can support the systematic structuring of explainability requirements, contributing to their integration of explainability as a fundamental concern in RE for intelligent systems.

7 Threats to Validity

We delineate the threats to validity [28] and the constraints on the outcomes of this study, considering the adopted research methodology.

Construct validity: a threat arises from the use of an illustrative scenario which, although based on a realistic domain, may not fully represent the complexity of real-world contexts. Additionally, the TAM3-based questionnaire may not cover all relevant evaluation dimensions. As mitigation, the scenario was designed from a real-world AI project, and the evaluation combined quantitative and qualitative data.

Internal validity: a limitation concerns the Langflow implementation, which does not support dynamic refinement during execution. Although the process is iterative and incremental, the implemented version follows a sequential structure, with iteration performed externally. Error scenarios were also not considered. As mitigation, standardized materials were adopted, focusing on artifact analysis and process consistency.

External validity: the small number of participants limits the generalization of findings. In addition, participants had similar professional profiles. As mitigation, both participants had extensive experience in Software Engineering and RE.

Conclusion validity: the exploratory nature of the study and the limited dataset may affect the robustness of the conclusions. To mitigate this limitation, quantitative and qualitative data were combined to support a broader analysis. The study aims to provide initial indications regarding the applicability of the framework rather than statistically generalizable conclusions.

8 Final Remarks

This research presented OpenUPExp, a framework for addressing explainability in Intelligent Systems from the early stages of development through elicitation, analysis, and specification activities. The proposal addresses a gap in the literature, where existing explainability solutions in RE remain fragmented and lack integration within a process-oriented approach.

In this work, the Explainability Vision artifact was introduced as an initial component of the framework, along with the process modeling using BPMN and its implementation through a worker-based architecture using Langflow. The evaluation conducted with experts provided initial insights regarding the coherence of the approach and the quality of the generated artifacts, indicating its potential to support the systematic structuring of explainability requirements.

Despite the promising results, OpenUPExp is still at an early stage of maturity and requires further refinements, particularly regarding artifact validation, result communicability, and technical detailing. In addition, the evaluation presents limitations, such as the small number of participants and the absence of applications in real-world contexts, which restrict the generalization of the findings. Therefore, the current findings should be interpreted as an initial assessment of the conceptual applicability of the framework rather than as an evaluation of effectiveness in real-world environments.

As future work, OpenUPExp will evolve toward the Elaboration phase, particularly through activities related to eliciting, analyzing, and specifying concrete explainability requirements derived from the Explainability Vision. Future

investigations also include empirical studies in real-world settings to evaluate the applicability of OpenUPExp across different contexts, teams, and domains, contributing to increasing its maturity and robustness.

References

1. Chazette, L., et al.: Exploring explainability: a definition, a model, and a knowledge catalogue. In: IEEE 29th International Requirements Engineering Conference. pp. 197–208 (2021). https://doi.org/10.1109/RE51729.2021.00025
2. Chazette, L., et al: Explainable software systems: from requirements analysis to system evaluation. Requirements Engineering **27**(4), 457–487 (2022). https://doi.org/10.1007/s00766-022-00393-5
3. Crook, B., et al: Revisiting the performance-explainability trade-off in explainable artificial intelligence (xai). In: IEEE 31st International Requirements Engineering Conference Workshops. pp. 316–324 (2023). https://doi.org/10.1109/REW57809.2023.00060
4. d'Aloisio, G.: Quality-driven machine learning-based data science pipeline realization: a software engineering approach. In: 44th International Conference on Software Engineering. pp. 291–293 (2022). https://doi.org/10.1145/3510454.3517067
5. Davis, F.D.: A technology acceptance model for empirically testing new end-user information systems: Theory and results. Ph.D. thesis, MIT (1985)
6. Deters, H., et al.: Exploring the means to measure explainability: Metrics, heuristics and questionnaires. Information and Software Technology p. 107682 (2025). https://doi.org/10.1016/j.infsof.2025.107682
7. Eclipse Foundation: Open Unified Process (OpenUP). https://archive.eclipse.org/epf/downloads/OpenUP/ (2012)
8. Franch, X., et al.: A requirements engineering perspective to ai-based systems development: A vision paper. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. pp. 223–232 (2023). https://doi.org/10.1007/978-3-031-29786-1_15
9. Habiba, U.E., Bogner, J., Wagner, S.: Can requirements engineering support explainable artificial intelligence? towards a user-centric approach for explainability requirements. In: IEEE 30th International Requirements Engineering Conference Workshops. pp. 162–165 (2022). https://doi.org/10.1109/REW56159.2022.00038
10. Habibullah, K.M.e.a.: Non-functional requirements for machine learning: Understanding current use and challenges among practitioners. Requir. Eng. **28**(2), 283–316 (2023). https://doi.org/10.1007/s00766-022-00395-3
11. Haindl, P.e.a.: Quality characteristics of a software platform for human-ai teaming in smart manufacturing. In: International Conference on the Quality of Information and Communications Technology. pp. 3–17 (2022). https://doi.org/10.1007/978-3-031-14179-9_1
12. Huynh, T.D., et al.: A methodology and software architecture to support explainability-by-design. arXiv preprint arXiv:2206.06251 (2022). https://doi.org/10.48550/arXiv.2206.06251
13. Khattak, F.K., et al.: Mlhops: Machine learning health operations. IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3521279

14. Köhl, M.A., et al.: Explainability as a non-functional requirement. In: IEEE 27th International Requirements Engineering Conference. pp. 363–368 (2019). https://doi.org/10.1109/RE.2019.00046
15. Ligocka, P.: Development of a Concept for the Documentation of Explainability Requirements. Ph.D. thesis, Leibniz University Hannover, Germany (2025)
16. Mancine, L.: Engenharia de requisitos de explicabilidade em sistemas baseados em aprendizado de máquina. In: Congresso Ibero-Americano em Engenharia de Software (CIbSE). pp. 311–318 (2025). https://doi.org/10.5753/cibse.2025.35316, (in Portuguese)
17. Mancine, L., et al: Supplementary material – toward a framework for explainability in intelligent systems: Preliminary results (2026), https://doi.org/10.5281/zenodo.19423805
18. Mancine, L., et al.: Estado da arte sobre engenharia de requisitos e explicabilidade em sistemas baseados em aprendizado de máquina. In: 30th Brazilian Symposium on Multimedia and Web Systems. pp. 143–158 (2024). https://doi.org/10.5753/webmedia_estendido.2024.243944, (in Portuguese)
19. Ministério da Ciência, Tecnologia e Inovação (MCTI) and Centro de Gestão e Estudos Estratégicos (CGEE): IA para o bem de todos: Plano Brasileiro de Inteligência Artificial. Tech. rep., MCTI / CGEE, Brasília, DF (2025), https://www.cgee.org.br/documents/10195/11009772/CGEE_PBIA.PDF
20. Schoonderwoerd, T.A., et al.: Human-centered xai: Developing design patterns for explanations of clinical decision support systems. Int. J. Hum. Comput. Stud. **154**, 102684 (2021). https://doi.org/10.1016/j.ijhcs.2021.102684
21. Senado Federal do Brasil: PL Nº 2338, de 2023 (2023), https://www25.senado.leg.br/web/atividade/materias/-/materia/157233
22. Speith, T., Langer, M.: A new perspective on evaluation methods for explainable artificial intelligence (XAI). In: IEEE 31st International Requirements Engineering Conference Workshops. pp. 325–331 (2023). https://doi.org/10.1109/REW57809.2023.00061
23. Stepin, I., et al.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. IEEE Access **9**, 11974–12001 (2021). https://doi.org/10.1109/ACCESS.2021.3051315
24. Storey, M.A., et al.: Using a visual abstract as a lens for communicating and promoting design science research in software engineering. In: ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 181–186 (2017). https://doi.org/10.1109/ESEM.2017.28
25. Tsakalakis, N., et al.: A typology of explanations to support explainability-by-design. ACM Journal on Responsible Computing **2**(1), 1–36 (2025). https://doi.org/10.1145/3708504
26. Umm-E-Habiba: Requirements engineering for explainable ai. In: IEEE 31st International Requirements Engineering Conference. pp. 376–380 (2023). https://doi.org/10.1109/RE57278.2023.00058
27. Vogelsang, A., Borg, M.: Requirements engineering for machine learning: Perspectives from data scientists. In: IEEE 27th International Requirements Engineering Conference Workshops. pp. 245–251 (2019). https://doi.org/10.1109/REW.2019.00050
28. Wohlin, C., et al.: Experimentation in software engineering, vol. 236. Springer-Verlag (2012)
29. Zöller, M.A., et al.: Xautoml: A visual analytics tool for understanding and validating automated machine learning. ACM Transactions on Interactive Intelligent Systems **13**(4), 1–39 (2023). https://doi.org/10.1145/3625240