

Bridging Customer and User Perspectives: A Hybrid Approach to Unified Requirements Extraction

Priya Desai

School of Business, North Carolina Central University, North Carolina
pdesai@ncsu.edu

Abstract

Effective requirements engineering is essential for aligning software development with strategic business objectives and end users' practical needs. This paper presents a hybrid, integrated framework to bridge the gap between customer and user perspectives in requirements extraction. The framework combines goal-oriented and data-driven methodologies with advanced automation techniques including natural language processing and a large language model-based system (ReqNet) to simplify requirements elicitation, consolidation, and prioritization. Adopting a Design Science Research (DSR) approach, the proposed model proceeds through four core phases: requirement elicitation, data consolidation, prioritization and validation, and deployment and evaluation. Structured surveys, semi-structured interviews, usability testing, and focus groups are employed to collect comprehensive qualitative inputs from diverse stakeholder groups. These inputs are systematically integrated using collaborative filtering and the ERRC (Eliminate, Reduce, Raise, Create) technique, then prioritized with the Use Case Points (UCP) method to assess implementation complexity and stakeholder impact. A detailed case study on a car product management system demonstrates the framework's effectiveness, revealing that the unified approach reduces redundant processes and fosters strong stakeholder consensus by addressing conflicting requirements. The findings indicate significant improvements in requirement completeness, process efficiency, and alignment between business strategy and user experience. The paper outlines potential avenues for future research, including scalability across varied domains and real-time adaptive prioritization through enhanced AI integration. The proposed approach offers a replicable model for developing cohesive and actionable software requirements in complex development environments. Thus, the method yields enhanced outcomes.

Keywords: *Hybrid requirements engineering, Integrated requirements extraction, Customer-user perspectives, Automation techniques, Natural language processing, Design science research*

1. Introduction

Effective requirements engineering is essential for successful software development. In an era of rapidly evolving technology and increasingly diverse stakeholder demands, the challenge of accurately extracting and integrating software requirements has never been more critical. Modern organizations require methodologies that harmonize diverse inputs to produce innovative, robust, and responsive software solutions to real-world needs.

Article history:

Received (April 14, 2025), Review Result (May 10, 2025), Accepted (June 15, 2025)

Traditionally, requirements engineering methods have focused exclusively on single stakeholder groups. For example, classical approaches such as the Value-innovative Requirements Engineering (ViRE) method emphasize customer-centric business objectives or user-focused experiential needs. This narrow scope often leads to duplicated efforts, misaligned priorities, and fragmented requirements that fail to capture the entire spectrum of stakeholder expectations [1]. Such limitations have created a pressing need for a more holistic paradigm that integrates the multifaceted views of both customers and end users into a comprehensive framework.

Recent advances in technology have enabled the development of promising hybrid models that bridge this gap. By merging goal-oriented techniques with data-driven insights, these models have improved the precision and efficiency of the requirement extraction process, ensuring that both strategic business goals and practical usability needs are addressed [2]. Moreover, integrating advanced Natural Language Processing (NLP) and deep learning methods has automated significant portions of the analysis process. These techniques reduce manual overhead and help standardize and refine the extracted requirements, minimizing errors and inconsistencies [3]. [Figure 1] illustrates this process by depicting the flow from the initial collection of distinct customer and user requirements through the stages of data collection, digitization, and automated hybrid analysis, culminating in unified requirements set that underpins improved software development outcomes.

In addition to technological innovation, recent scholarship has increasingly recognized the importance of synthesizing diverse stakeholder insights. A unified perspective that captures customer-driven strategic objectives and user-based experiential insights is crucial for designing systems balanced in functionality and usability [4]. The framework presented in this study addresses the central research question: How can integrating customer and user perspectives in a hybrid requirements extraction framework improve the quality of software development outcomes? By addressing this question, the hybrid model seeks to eliminate redundant processes and produce a more complete, data-informed, actionable set of requirements.

This expanded approach not only leverages automation and advanced analytics but also emphasizes the role of iterative stakeholder engagement. Incorporating feedback loops at various stages enhances the evolving requirements set, ensuring that the system remains relevant to market demands and operational realities. The integration of qualitative inputs with quantitative analysis, as defined in [Figure 1], enables a thorough and scalable methodology that is easily adapted across various domains and project scales.

In summary, by embracing technological advancements and multifaceted stakeholder insights, the framework described here offers a replicable model that enhances requirement completeness, improves stakeholder collaboration, and ultimately leads to superior software development outcomes.

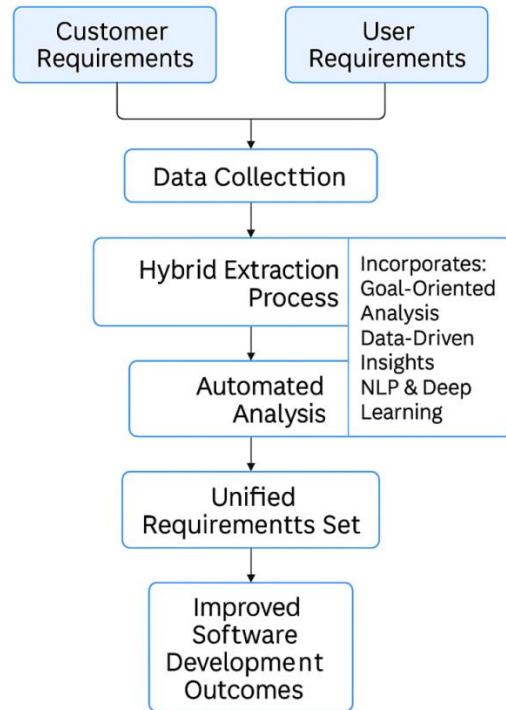


Figure 1. Schematic overview of the hybrid requirements extraction process

2. Review of hybrid requirements engineering approaches

Effective requirements engineering is critical to modern software development because it aligns technical implementations with strategic business objectives and end users' practical needs. Traditional approaches often treat customer and user requirements separately, leading to redundant processes and incomplete software solutions. In response, researchers have proposed hybrid requirements engineering models that integrate these perspectives within a unified framework [5][6][7][8].

A systematic search was conducted across several academic databases focusing on works published between 2022 and 2025. Keywords such as “hybrid requirements engineering,” “integrated requirements extraction,” “goal-oriented requirements engineering,” and “customer and user perspectives in software development” were used to locate relevant peer-reviewed articles and conference proceedings. This process yielded studies that contributed theoretical insights and practical solutions toward unifying diverse stakeholder requirements.

2.1. Key contributions and insights

Several recent studies have significantly advanced the discourse on hybrid requirements engineering:

1. Hybrid Prioritization Techniques: Arshad et al. [5] propose a novel model synthesizing traditional requirement prioritization methods with modern, customizable approaches. Their work demonstrates how integrating critical project factors can produce more balanced and effective requirement extraction.
2. Goal-Oriented and Data-Driven Methods: Bernanda et al. [6] introduce an integrated approach that combines goal-oriented requirements with data-driven

techniques. This method reduces subjectivity in requirement elicitation and ensures that design outputs align with business and technical objectives.

3. Collaborative Filtering and Automation: Shambour et al. [7] developed a hybrid content-based collaborative filtering approach that effectively uncovers overlooked yet critical requirements, thereby enhancing the completeness and efficiency of the elicitation process.
4. Emerging Trends in Requirements Engineering: Siddique [8] reviews the evolving landscape of requirements engineering, highlighting the pressing need for integrated frameworks that simultaneously address customer and user perspectives.

In addition to these foundational studies, several works offer further insights that reinforce the need for a unified approach:

1. Customer Feedback Integration: Marcu and the MoldStud Research Team [9] discuss best practices for systematically integrating customer feedback into the development process, ensuring that user-driven insights are not overlooked.
2. User Experience Considerations: Kashfi, Nilsson, and Feldt [10] examine methods for embedding User Experience (UX) principles into early phases of software development, demonstrating tangible benefits in product usability and customer satisfaction.
3. Balancing Innovation and Regulation: Another study by Siddique [11] explores the challenges of fostering innovation while maintaining regulatory compliance, offering strategies to balance these sometimes conflicting objectives within a unified requirements framework.
4. AI-Driven Requirements Extraction: Myllyaho et al. [12] provide an exhaustive review of how artificial intelligence, particularly machine learning and natural language processing, reshapes the requirements engineering landscape by automating extraction processes from unstructured data sources.
5. Automation from Source Code: Ivanov et al. [13] illustrate how automated systems can extract requirements directly from the source code, enhancing precision and reducing manual intervention in elicitation.
6. Leveraging Large Language Models: Saleem, Asim, and Dengel [13] introduce “ReqNet,” a computational framework employing large language models to automate requirements extraction, which shows promise in significantly improving efficiency and accuracy.
7. Integrated Automation Frameworks: Finally, Miskell et al. [14] present an automated framework that cross-references various data sources to produce a holistic requirements extraction process.

Although the above studies demonstrate substantial progress in addressing the complexity of modern requirements engineering, the literature indicates an ongoing need for frameworks that fully integrate customer and user insights. Most existing methods treat these perspectives independently. The gap remains in developing a unified; hybrid approach that concurrently considers both views to yield comprehensive and actionable requirements.

3. Integrated customer–user requirements framework

This study investigates the research question: How can integrating customer and user perspectives in a hybrid requirements extraction framework improve the quality of software development outcomes? To answer this, a Design Science Research (DSR) approach was employed, as it emphasizes the construction and evaluation of artifacts aimed at solving real-

world problems. DSR is particularly appropriate for research seeking to bridge theoretical and practical domains and has been successfully applied in prior software engineering contexts to develop and refine innovative frameworks [15]. This study's hybrid methodology integrates qualitative and quantitative research methods into a unified model, progressing through four core phases: requirement elicitation, data consolidation, prioritization and validation, and deployment and evaluation. Drawing on the work of Arshad et al. [5] and Bernanda et al. [6], the model combines goal-oriented analysis with data-driven insights. Uniquely, it incorporates advanced automation techniques such as collaborative filtering enhanced with Natural Language Processing (NLP) [7] and a large language model-based system called ReqNet, which supports accurate and scalable requirements extraction from unstructured sources [15]. Integrating machine intelligence with traditional engineering processes sets the approach apart from conventional methods.

Data were collected from two distinct sources, customers and end users, to ensure a comprehensive understanding of stakeholder expectations. Customer data were gathered using structured surveys and semi-structured interviews, focusing on strategic goals and market alignment. In parallel, user requirements were obtained through usability testing sessions, direct observation, and moderated focus groups to capture practical concerns and experiential insights. All qualitative inputs were digitized, anonymized, and stored in a centralized knowledge repository, allowing for efficient cross-analysis and traceability between stakeholder groups. Once collected, the data underwent a multi-step analysis and integration process. First, collaborative filtering algorithms were applied to cluster requirements based on topical and semantic similarity, which streamlined the consolidation of redundant inputs. Next, a conflict resolution mechanism was implemented using the ERRC (Eliminate, Reduce, Raise, Create) technique, a strategic tool for aligning divergent stakeholder views through structured comparison [16]. Following this, the consolidated requirement set was prioritized using the Use Case Points (UCP) method, which assesses requirements according to their implementation complexity and stakeholder impact [17].

Each methodological decision was carefully justified to ensure both rigor and relevance. Integrating customer and user data into a single pipeline reduces redundancy and enhances the alignment between business strategy and user experience. Automation, particularly NLP and machine learning techniques, significantly decreases manual labor while improving the precision and scalability of requirement extraction [14][15]. Moreover, the iterative nature of the process facilitated by stakeholder validation workshops ensures that the evolving requirement set remains accurate, actionable, and aligned with project goals. Despite its advantages, the research encountered several practical challenges. Semantic ambiguity in user-generated feedback initially posed a barrier to accurate analysis. This was addressed through preprocessing techniques enabled by ReqNet, which helped normalize language patterns and improve classification accuracy [15].

Furthermore, stakeholder misalignment was identified as a recurring issue, particularly during the early integration phases. To mitigate this, recurring workshops were implemented to encourage consensus-building. Finally, technical issues related to integrating modern AI tools with legacy development environments were overcome by adopting a modular system architecture based on standard API protocols.

This hybrid methodology demonstrates how structured stakeholder engagement, advanced data processing tools, and design science principles can lead to more coherent, comprehensive, and actionable software requirements. The following section illustrates the application and evaluation of this methodology through a case study on a car product management system.

4. Application results of the hybrid framework

The integrated use case-oriented requirements engineering process was implemented in a case study centered on a car product management system. This application evaluated the framework’s effectiveness in reconciling and prioritizing the conflicting requirements emerging from two distinct stakeholder groups, customers and end users. Initial findings revealed that customer stakeholders emphasized strategic system-level features such as analytics dashboards, comprehensive inventory management, and seamless enterprise integrations. In contrast, end users, particularly operational staff, prioritized improvements in usability, enhanced task automation, and easier access to frequently used functions.

To manage these divergent expectations, the ERRC (Eliminate, Reduce, Raise, Create) model was applied as a systematic tool for categorizing use cases. This approach allowed each requirement to be evaluated on its relevance and stakeholder value, ensuring that redundant functionalities were removed while critical gaps were bridged. For instance, use cases such as UC3 (real-time inventory tracking), UC10 (one-click order status lookup), UC17 (integrated supplier communication), and UC20 (AI-powered sales prediction) were classified as “Create,” indicative of newly introduced functionalities designed to address latent needs. Conversely, use cases like UC6 (manual entry of part numbers), UC8 (weekly PDF report generation), and UC14 (export to outdated legacy formats) were marked as “Eliminate” due to issues of redundancy or obsolescence. Use case UC16 (smart search functionality) was categorized as “Raise,” suggesting necessary enhancements to existing features. Table 1 summarizes the categorization:

Table 1. ERRC categorization of use cases

Use Case ID	Requirement Description	ERRC Action	Stakeholder Origin
UC3	Track inventory in real-time	Create	Customer
UC6	Manual entry of part numbers	Eliminate	User
UC8	Weekly PDF report generation	Eliminate	Customer
UC10	One-click order status lookup	Create	User
UC14	Export to outdated legacy formats	Eliminate	Customer
UC16	Smart search functionality	Raise	User
UC17	Integrated supplier communication panel	Create	Customer
UC20	AI-powered sales prediction engine	Create	Customer

Create	Raise	Eliminate
UC3 UC10 UC17 UC20	UC16	UC6 UC8 UC14

Figure 2. ERRC analysis of use cases

[Figure 2] illustrates the overall analysis and classification flow used during the study, reinforcing the role of the ERRC model in achieving transparent and balanced decision-making. Following the initial categorization, stakeholder workshops were organized to validate the ERRC decisions. These sessions allowed representatives from both customer and

user groups to negotiate and refine the classification of use cases. This iterative validation process was crucial in ensuring that the unified requirements truly reflected strategic and operational perspectives. To further quantify the prioritization, the finalized use cases underwent evaluation using the Use Case Points (UCP) method. High UCP scores for features like UC20 and UC3 underscored their alignment with innovation goals and broad operational impact. Moderate UCP scores assigned to UC10 and UC16 indicated that these features, while necessary for usability improvements, posed relatively lower implementation complexity.

Overall, the hybrid method bridged the gap between customer-driven and user-driven requirements and facilitated enhanced stakeholder collaboration—the visual representation provided by [Figure 2] aided in comprehending the trade-offs and aligning divergent priorities. The success of this approach in the car product management system case study demonstrates the viability of a holistic framework that integrates advanced automation with structured stakeholder engagement to deliver actionable, balanced requirements. Future implementations of this framework could explore additional domains and incorporate further real-time validation mechanisms to cope with evolving project demands.

5. Discussion

The findings from this case study underscore the value of integrating customer and user perspectives within a unified requirements engineering framework. Traditional methods often emphasize one perspective over the other, leading to imbalanced systems—either rich in business logic but lacking usability or highly user-centric but failing to align with strategic objectives. The hybrid methodology applied here mitigated this issue by enabling real-time comparison and synthesis of diverse stakeholder inputs. The ERRC framework was a practical decision-making tool for reconciling stakeholder conflicts and uncovering hidden requirements. For instance, UC10 (one-click order status lookup), originally unarticulated by customers, emerged as a high-impact feature from user feedback. Similarly, eliminating UC14 (export to legacy formats) streamlined development focus while removing outdated obligations that no longer served strategic or user goals.

Moreover, the introduction of automation tools—such as NLP-driven clustering and large language model-based extraction—significantly reduced manual effort and increased consistency in requirement classification. This technological augmentation aligned well with the complexity of the car product management system and helped address ambiguity in stakeholder expressions. The study also highlights the importance of stakeholder workshops and validation loops in requirements engineering. Feedback sessions acted as convergence points for conflicting views, where trade-offs were openly discussed and resolved through structured facilitation. This iterative engagement proved critical in aligning functional developments and evolving organizational needs. Finally, integrating prioritization models like UCP brought objectivity to the decision process, enabling data-driven assessment of which requirements should lead development efforts. Features like UC20 (AI-powered sales prediction) scored highly due to their innovation potential and alignment with broader digital transformation goals.

In summary, the hybrid approach demonstrated measurable improvements in requirement completeness, stakeholder satisfaction, and overall process efficiency. The case study reinforces the argument that integrated frameworks blend methodological rigor with automation and stakeholder collaboration and are well-suited for complex software environments.

6. Summary of findings and future work

This study introduced a hybrid, integrated use case-oriented requirements engineering framework that effectively combines customer and user perspectives into a unified pipeline. By addressing the limitations inherent in traditional approaches which typically treat stakeholder groups independently, the proposed framework streamlines the entire process of requirements extraction. Using the ERRC model and the Use Case Points analysis methodology, the framework achieves strategic prioritization and operational clarity, ensuring that critical system features are neither overlooked nor misaligned with end-user needs.

The case study, which focused on a car product management system, provided practical evidence of the framework's efficacy. Within this study, use cases were systematically categorized through the ERRC technique, eliminating redundant features while identifying high-impact requirements, such as AI-driven sales prediction and usability enhancements. The clarity brought about by this categorization not only facilitated more informed decision-making but also fostered deeper stakeholder engagement during iterative validation sessions. Additionally, incorporating advanced natural language processing and large language model-driven automation significantly reduced manual effort, enhancing the precision and scalability of requirements classification.

Despite these promising outcomes, certain challenges were noted. For instance, handling semantic ambiguities in stakeholder inputs and aligning conflicting objectives during initial integration stages required iterative refinement and increased stakeholder dialogue. These challenges reveal opportunities for future enhancements.

Future research directions include:

- **Scalability Across Domains:** To assess its adaptability, evaluate and refine the framework's application in varied sectors—such as healthcare, finance, or large-scale enterprise systems.
- **Real-Time Feedback and Adaptive Prioritization:** Integrating mechanisms for real-time stakeholder feedback and dynamic AI-based prioritization to adjust the requirements set as project conditions evolve continually.
- **Enhanced Automation Techniques:** Leveraging further advancements in machine learning and natural language processing to improve classification accuracy and reduce manual intervention.
- **Multi-Factor Integration:** Extending the framework to incorporate additional dimensions, such as regulatory compliance, security requirements, and evolving market trends, to ensure a more robust and resilient system design.

In summary, the framework provides a replicable model that enhances the completeness and relevance of extracted requirements and streamlines the entire engineering process. These findings underscore the value of adopting unified, technology-assisted approaches in modern software development while also highlighting clear avenues for future exploration and refinement.

References

- [1] L. Lehtola, M. Kauppinen, and S. Kujala, "Requirements prioritization challenges in practice," in *Product Focused Software Process Improvement (PROFES 2004)*, F. Bomarius and H. Iida, Eds., Lecture Notes in Computer Science, vol.3009, Springer, Berlin, Heidelberg, (2004). DOI: 10.1007/978-3-540-24659-6_36
- [2] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. De Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *Journal of Systems and Software*, vol.172, p.110851, (2021). DOI:10.1016/j.jss.2020.110851

- [3] S. Saleem, M. N. Asim, and A. Dengel, “ReqNet: A large language model driven computational framework for automated requirements extraction from unstructured documents,” SSRN, (2025). DOI:10.2139/ssrn.5122555
- [4] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, “Exploring the intersection between software industry and software engineering education – A systematic mapping of software engineering trends,” *Journal of Systems and Software*, vol.172, p.110736, (2021). DOI:10.1016/j.jss.2020.110736
- [5] H. Arshad, S. Shaheen, J. A. Khan, M. S. Anwar, K. Aurangzeb, and M. Alhussein, “A novel hybrid requirements prioritization approach based on critical software project factors,” *Cognition, Technology & Work*, vol.25, pp.305–324, (2023)
- [6] D. Y. Bernanda, D. A. Jawawi, S. Halim, and F. Adikara, “Hybrid goal-oriented and data driven requirement engineering approach for information systems requirements,” SSRN, (2025). DOI:10.2139/ssrn.5263008
- [7] Q. Y. Shambour, A. H. Hussein, Q. M. Kharma, and M. M. Abualhaj, “Effective hybrid content-based collaborative filtering approach for requirements engineering,” *Computer Systems Science and Engineering*, vol.40, no.1, pp.113–125, (2022). DOI:10.32604/csse.2022.017221
- [8] I. M. Siddique, “Requirements engineering at a turning point: Key research trends shaping the future of systems and software integration,” *Journal of Geotechnical Studies*, vol.9, no.3, pp.37–49, (2024)
- [9] C. Marcu and MoldStud Research Team, “Integrating customer feedback in software development: Best practices for success,” *MoldStud*, (2024). <https://moldstud.com/articles/p-integrating-customer-feedback-in-software-development-projects>
- [10] P. Kashfi, A. Nilsson, and R. Feldt, “Integrating user experience practices into software development processes: Implications of the UX characteristics,” *PeerJ Computer Science*, vol.3, p.e130, (2017). DOI:10.7717/peerj-cs.130
- [11] I. M. Siddique, “Balancing innovation and regulation: Requirements engineering for emerging technologies,” *European Journal of Advances in Engineering and Technology*, vol.9, no.11, pp.95–99, (2022)
- [12] L. Myllyaho, M. Raatikainen, T. Männistö, T. Mikkonen, and J. K. Nurminen, “Systematic literature review of validation methods for AI systems,” *Journal of Systems and Software*, vol.181, p.111050, (2021). DOI:10.1016/j.jss.2021.111050
- [13] V. Ivanov, A. Sadovykh, A. Naumchev, A. Bagnato, and K. Yakovlev, “Extracting software requirements from unstructured documents,” in *Communications in Computer and Information Science*, vol.1573, Springer, (2022). DOI:10.1007/978-3-031-15168-2_2
- [14] C. Miskell, R. Diaz, P. Ganeriwala, K. Slhoub, and F. Nembhard, “Automated framework to extract software requirements from source code,” in *Proc. 7th Int. Conf. on Natural Language Processing and Information Retrieval (NLPIR 2023)*, Seoul, Republic of Korea, (2023). DOI:10.1145/3639233.3639242
- [15] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Quarterly*, vol.28, no.1, pp.75–105, (2004)
- [16] W. C. Kim and R. Mauborgne, *Blue ocean strategy: How to create uncontested market space and make the competition irrelevant*, Harvard Business Review Press, (2005)
- [17] G. Karner, “Metrics for Objectory,” Master's thesis, University of Linköping, (1993)

This page is empty by intention.