World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Check for updates

Advancements and innovations in requirements elicitation: Developing a comprehensive conceptual model

Oladapo Adeboye Popoola [1,*], Henry Ejiga Adama [2], Chukwuekem David Okeke [3] and Abiodun Emmanuel Akinoso [4]

[1] *Business Full Spectrum, United Kingdom.*
[2] *System/information Analyst, Greenville Texas, USA.*
[3] *Tranter IT Infrastructure Services Limited, Nigeria.*
[4] *Independent Researcher, United Kingdom.*

World Journal of Advanced Research and Reviews, 2024, 22(01), 1209–1220

Publication history: Received on 11 March 2024; revised on 20 April 2024; accepted on 22 April 2024

Article DOI: https://doi.org/10.30574/wjarr.2024.22.1.1202

Abstract

Requirements elicitation is a crucial phase in the software development lifecycle, ensuring that stakeholders' needs are understood and translated into system specifications. Traditional methods often face challenges such as ambiguity, inconsistency, and evolving requirements, leading to project delays and cost overruns. This review proposes a conceptual model that integrates advancements and innovations in requirements elicitation to address these challenges comprehensively. The proposed model leverages various techniques, including but not limited to, natural language processing (NLP), machine learning (ML), and human-computer interaction (HCI), to enhance the accuracy and efficiency of requirements elicitation. NLP techniques enable automated analysis of textual requirements documents, extracting key information and identifying implicit requirements. ML algorithms facilitate the prediction of potential changes in requirements based on historical data and project context, enabling proactive management of evolving requirements. Moreover, the integration of HCI principles in the requirements elicitation process enhances stakeholder engagement and collaboration. Interactive interfaces and visualization tools enable stakeholders to provide feedback in real-time, fostering a more iterative and participatory approach to requirement gathering. Additionally, techniques such as prototyping and simulation facilitate early validation of requirements, reducing the risk of misinterpretation and ensuring alignment with stakeholders' expectations. Furthermore, the proposed model emphasizes the importance of context-awareness in requirements elicitation. By considering the organizational, cultural, and environmental context of a project, the model adapts its elicitation strategies and techniques to suit specific needs and constraints. Context-awareness also enables the identification of implicit requirements that may not be explicitly stated but are crucial for the success of the system. The model also addresses the challenge of managing conflicting requirements by introducing a systematic approach to requirements prioritization and negotiation. By employing multi-criteria decision-making techniques, stakeholders can collaboratively prioritize requirements based on their importance, feasibility, and impact on project objectives. Conflict resolution mechanisms facilitate consensus-building and trade-off analysis, ensuring that the final set of requirements reflects the collective interests of all stakeholders. Additionally, the proposed model emphasizes the iterative nature of requirements elicitation, advocating for continuous refinement and validation of requirements throughout the development lifecycle. Feedback loops enable stakeholders to review and revise requirements based on evolving needs and changing circumstances, thereby enhancing the adaptability and resilience of the system. The conceptual model presented in this review represents a holistic approach to requirements elicitation, leveraging advancements and innovations in technology, human interaction, and context-awareness. By integrating these elements, the model aims to enhance the accuracy, efficiency, and stakeholder satisfaction in the requirements elicitation process, ultimately contributing to the successful delivery of high-quality software systems.

* Corresponding author: Oladapo Adeboye Popoola

Keywords: Advancement; Innovation; Requirement elicitation; Comprehensive conceptual model

1. Introduction

In the dynamic landscape of software development, requirements elicitation stands as a cornerstone process, serving as the bridge between stakeholders' needs and the development of software systems (Pargaonkar, 2023). It encompasses the activities involved in understanding, capturing, and refining the requirements of a software system, ensuring that they accurately reflect the desires and expectations of stakeholders. Effective requirements elicitation is crucial for the success of software projects, as it lays the foundation for the entire development process, influencing design decisions, development efforts, and ultimately, the satisfaction of end-users (Aldave *et al*., 2019; Sonko *et al*., 2024). Requirements elicitation is a multifaceted process that involves identifying, analyzing, prioritizing, and documenting the requirements of a software system. It encompasses interactions between various stakeholders, including end-users, customers, business analysts, developers, and other relevant parties. The goal of requirements elicitation is to capture the functional and non-functional requirements of the system, including user needs, system capabilities, performance criteria, and constraints (García-López *et al*., 2020). The process typically begins with the identification of stakeholders and the establishment of communication channels to gather their input. Techniques such as interviews, surveys, workshops, and observations are employed to elicit requirements from stakeholders, each offering unique advantages in terms of depth of understanding and stakeholder engagement (Gregory *et al*., 2020; Farayola *et al*., 2023). Once requirements are gathered, they are analyzed, clarified, and prioritized to ensure consistency, completeness, and feasibility. Documentation plays a critical role in requirements elicitation, providing a formal record of the gathered requirements that serves as a reference for all stakeholders throughout the development lifecycle (Behutiye *et al*., 2020). Various artifacts such as requirement specifications, use cases, user stories, and prototypes are used to capture and communicate requirements effectively, catering to different stakeholders' needs and preferences. Despite its importance, requirements elicitation is fraught with challenges that can impede its effectiveness and efficiency. Ambiguity, inconsistency, and volatility of requirements are common issues that arise due to factors such as evolving stakeholder needs, changing business environments, and communication barriers between stakeholders (Siakas *et al*., 2021; Oladeinde *et al*., 2023). Traditional methods of requirements elicitation often struggle to address these challenges adequately, leading to project delays, cost overruns, and ultimately, dissatisfaction among stakeholders (Ajmal *et al*., 2022).

The field of requirements elicitation has witnessed significant advancements and innovations in recent years, driven by the rapid evolution of technology and the increasing complexity of software systems (Bukhsh *et al*., 2020). These advancements hold immense potential in addressing the challenges faced by traditional requirements elicitation methods, offering new approaches, techniques, and tools to enhance the accuracy, efficiency, and effectiveness of the process. One of the key advancements in requirements elicitation is the integration of natural language processing (NLP) techniques, which enable automated analysis of textual requirements documents (Zhao *et al*., 2021). NLP algorithms can extract key information, identify implicit requirements, and detect inconsistencies within requirements specifications, reducing the manual effort required for analysis and validation. Machine learning (ML) techniques complement NLP by leveraging historical data to predict potential changes in requirements and proactively manage evolving requirements throughout the development lifecycle. Human-computer interaction (HCI) principles have also been integrated into requirements elicitation processes, facilitating stakeholder engagement and collaboration. Interactive interfaces, visualization tools, and prototyping techniques enable stakeholders to provide feedback in real-time, fostering a more iterative and participatory approach to requirement gathering (Singgalen, 202; Ezeigweneme *et al*., 2024). Additionally, context-awareness techniques consider the organizational, cultural, and environmental context of a project, adapting elicitation strategies and techniques to suit specific needs and constraints. Another area of innovation in requirements elicitation is the systematic prioritization and negotiation of conflicting requirements. Multi-criteria decision-making techniques enable stakeholders to collaboratively prioritize requirements based on their importance, feasibility, and impact on project objectives (Estévez *et al*., 2021). Conflict resolution mechanisms facilitate consensus-building and trade-off analysis, ensuring that the final set of requirements reflects the collective interests of all stakeholders. Furthermore, the emphasis on iterative refinement and validation of requirements throughout the development lifecycle enhances the adaptability and resilience of software systems (Ebirim *et al*. 2024). Feedback loops enable stakeholders to review and revise requirements based on evolving needs and changing circumstances, mitigating the risk of misinterpretation and ensuring alignment with stakeholders' expectations.

Given the complexity and criticality of requirements elicitation in software development, there is a pressing need for a comprehensive conceptual model that integrates advancements and innovations in the field (Rasheed *et al*., 2021). The purpose of developing such a model is to provide a structured framework that guides practitioners in conducting requirements elicitation activities effectively and efficiently, while also addressing the challenges and limitations of traditional methods. A comprehensive conceptual model serves as a roadmap for navigating the requirements

elicitation process, offering guidance on the selection and application of appropriate techniques, tools, and methodologies (Naeem *et al*., 2023). By incorporating advancements in technology, human interaction, and context-awareness, the model enables practitioners to leverage the full potential of available resources and expertise in eliciting and managing requirements. Moreover, a comprehensive conceptual model fosters collaboration and alignment among stakeholders by providing a common language and understanding of the requirements elicitation process (Sidaoui *et al*., 2024). It facilitates communication, negotiation, and decision-making, ensuring that stakeholders are actively involved and invested in the outcome of the process. Ultimately, the development of a comprehensive conceptual model aims to enhance the accuracy, efficiency, and stakeholder satisfaction in the requirements elicitation process, ultimately contributing to the successful delivery of high-quality software systems (Lim *et al*., 2021; Babatunde *et al*., 2024). By integrating advancements and innovations in requirements elicitation, the model equips practitioners with the tools and techniques necessary to address the evolving needs and challenges of software development in today's dynamic environment.

2. Traditional Challenges in Requirements Elicitation

Requirements elicitation, despite being a critical phase in software development, is riddled with various challenges that have persisted over time. These challenges can impede the effectiveness and efficiency of the process, leading to project delays, cost overruns, and ultimately, dissatisfaction among stakeholders. One of the most pervasive challenges in requirements elicitation is the presence of ambiguity and inconsistency in requirements specifications (Ribeiro and Berry 2020.). Ambiguity refers to the lack of clarity or precision in requirements, making them susceptible to multiple interpretations. Inconsistency, on the other hand, refers to contradictions or conflicts between different requirements or within the same requirement specification. Ambiguity and inconsistency in requirements can arise due to various factors, including vague language, implicit assumptions, and conflicting stakeholder perspectives (Siakas *et al*., 2022). Natural language, which is commonly used to express requirements, is inherently prone to ambiguity, as words and phrases may have different meanings depending on context. Additionally, stakeholders may have different priorities, preferences, and objectives, leading to inconsistencies in their requirements. The presence of ambiguity and inconsistency in requirements poses several challenges to the requirements elicitation process (Saeeda *et al*., 2020). Firstly, it can lead to misunderstandings and misinterpretations among stakeholders, resulting in a lack of consensus and alignment on project goals and objectives. Secondly, ambiguity and inconsistency can hinder the development team's ability to accurately understand and implement the requirements, leading to rework, defects, and delays in project delivery. Finally, ambiguity and inconsistency can undermine the credibility and trustworthiness of the requirements specification, eroding stakeholder confidence in the project's success.

Another significant challenge in requirements elicitation is the phenomenon of evolving requirements, whereby stakeholders' needs and priorities change over time. Evolving requirements can occur due to various factors, including changing business environments, emerging technologies, evolving user expectations, and shifting regulatory requirements (Betti and Sarens, 2021). Evolving requirements pose several challenges to the requirements elicitation process, chief among them being project delays and cost overruns. As requirements change throughout the development lifecycle, the scope of the project may expand, leading to increased development efforts, resource allocation, and time to market. Frequent changes to requirements can disrupt the development team's workflow, causing productivity losses and schedule slippages (Govindaras *et al*., 2023). Evolving requirements can also have cost implications for the project, as additional resources may be required to accommodate changes, such as hiring additional staff, investing in new technologies, or reworking existing software components. Changes to requirements may necessitate revisions to project plans, budgets, and timelines, leading to financial uncertainty and risk. Managing evolving requirements effectively requires proactive communication, collaboration, and change management strategies. Stakeholders must be engaged throughout the development lifecycle, and mechanisms should be in place to capture, prioritize, and track changes to requirements. Agile methodologies, such as iterative development and incremental delivery, are often employed to accommodate evolving requirements and mitigate their impact on project schedules and budgets (Al-Saqqa *et al*., 2020).

Conflicting requirements, where different stakeholders have competing priorities or objectives, pose another significant challenge in requirements elicitation. Conflicts may arise due to differences in stakeholders' roles, responsibilities, and perspectives, as well as conflicting organizational goals, constraints, and priorities. Managing conflicting requirements is a delicate balancing act that requires careful negotiation, compromise, and trade-off analysis. Conflicts may manifest at various levels, including conflicting functional requirements, conflicting non-functional requirements, and conflicting stakeholder preferences. Conflicting requirements can lead to several adverse consequences for software projects, including delays in decision-making, compromised quality, and stakeholder dissatisfaction (Behutiye *et al*., 2022). Failure to resolve conflicts effectively can result in project scope creep, where the scope of the project expands beyond its initial boundaries, leading to schedule overruns and budgetary constraints. Various techniques and strategies can be employed to manage conflicting requirements, including prioritization, negotiation, consensus-building, and

arbitration. Multi-criteria decision-making techniques, such as pairwise comparison and analytic hierarchy process (AHP), can help stakeholders prioritize conflicting requirements based on their importance, feasibility, and impact on project objectives (Yannis *et al*., 2020). Stakeholder engagement and collaboration are essential for resolving conflicts and reaching mutually acceptable solutions. Facilitated workshops, focus groups, and mediation sessions can provide a platform for stakeholders to voice their concerns, explore alternative solutions, and reach consensus on contentious issues. Traditional challenges in requirements elicitation, including ambiguity and inconsistency in requirements, evolving requirements leading to project delays and cost overruns, and difficulty in managing conflicting requirements, continue to plague software development projects. Addressing these challenges requires a combination of technical expertise, interpersonal skills, and organizational processes. By recognizing the inherent complexities of requirements elicitation and implementing effective strategies for managing challenges, software development teams can improve the accuracy, efficiency, and success of their projects (Atoum *et al*., 2021; Kasauli *et al*., 2021).

2.1. Leveraging Advancements in Technology for Requirements Elicitation

In the ever-evolving landscape of software development, leveraging advancements in technology has become imperative for addressing the challenges encountered in requirements elicitation (Challapalli, 2023). Natural Language Processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and human language. In the context of requirements elicitation, NLP plays a crucial role in automating the analysis of textual requirements, extracting key information, and identifying implicit requirements (Zaki-Ismail *et al*., 2022). Traditional methods of requirements elicitation often involve manual analysis of textual documents, such as user manuals, specifications, and stakeholder interviews. This process is time-consuming and prone to errors, as it relies on human interpretation and understanding of natural language.

NLP techniques automate the analysis of textual requirements by employing algorithms to parse, interpret, and extract relevant information from unstructured text (Baviskar *et al*., 2021). These algorithms can identify key nouns, verbs, and phrases within requirements documents, categorize them based on their semantic meaning, and extract structured data for further analysis. By automating the analysis of textual requirements, NLP enables software development teams to process large volumes of documentation more efficiently, identify important information more accurately, and reduce the risk of overlooking critical requirements. In addition to automating the analysis of textual requirements, its techniques can also facilitate the extraction of key information and the identification of implicit requirements (Hassan and Le, 2020). Implicit requirements are requirements that are not explicitly stated but are implied or inferred from the context of the document. Identifying implicit requirements is crucial for ensuring the completeness and comprehensiveness of the requirements specification. NLP algorithms can analyze the linguistic structures and patterns within requirements documents to infer implicit requirements. By recognizing recurring themes, synonyms, and antonyms, its techniques can identify requirements that may not be explicitly stated but are implied by the context of the document. By leveraging NLP for the extraction of key information and the identification of implicit requirements, software development teams can ensure that all relevant requirements are captured and documented, reducing the risk of misunderstandings and omissions during the development process.

Machine Learning (ML) is a subset of artificial intelligence that focuses on the development of algorithms that enable computers to learn from data and make predictions or decisions without being explicitly programmed (Tyagi and Chahal, 2022). In the context of requirements elicitation, ML can be used to predict potential changes in requirements and proactively manage evolving requirements based on historical data. One of the challenges of requirements elicitation is the dynamic nature of requirements, which often change throughout the development lifecycle due to evolving stakeholder needs, changing business environments, and emerging technologies. ML algorithms can analyze historical data from previous projects to identify patterns and trends in requirements changes. By recognizing common triggers and drivers of requirements changes, ML models can predict potential changes in requirements for current projects based on similar historical patterns (Susnjak *et al*., 2022). Predictive analytics techniques, such as regression analysis and time series forecasting, can be used to model the relationship between various factors and the likelihood of requirements changes. By forecasting future changes in requirements, software development teams can proactively plan and adapt their development efforts to accommodate these changes, reducing the risk of schedule delays and cost overruns. In addition to predicting potential changes in requirements, ML can also facilitate the proactive management of evolving requirements based on historical data (Aljohani, 2023). ML algorithms can analyze historical data to identify strategies and best practices for managing requirements changes effectively. By analyzing the impact of past changes on project schedules, budgets, and deliverables, ML models can identify patterns and trends that can inform decision-making and risk management strategies for current projects.

Furthermore, ML techniques can be used to develop adaptive systems that can dynamically adjust to changes in requirements in real-time. By monitoring key metrics and indicators of project performance, ML models can trigger

alerts and notifications when deviations from expected outcomes occur, enabling project teams to take corrective actions proactively (Khan and Masum, 2024). By leveraging ML for the proactive management of evolving requirements based on historical data, software development teams can enhance their ability to respond to changes effectively, minimize the impact of requirements changes on project schedules and budgets, and improve the overall success rate of software development projects.

Human-Computer Interaction (HCI) is a multidisciplinary field that focuses on the design, evaluation, and implementation of interactive computing systems for human use (Holden *et al*., 2022). In the context of requirements elicitation, HCI principles can be applied to develop interactive interfaces and visualization tools that facilitate stakeholder engagement and enable a more iterative and participatory approach to requirement gathering. Traditional methods of requirements elicitation often rely on static documentation, such as textual requirements specifications and diagrams, to communicate requirements to stakeholders. While these methods are effective for conveying information, they may not be conducive to active stakeholder engagement and collaboration. HCI principles emphasize the importance of user-centered design, which involves understanding the needs, preferences, and abilities of users and designing interfaces and tools that are intuitive, usable, and engaging (Issa and Isaias, 2022.). Interactive interfaces and visualization tools can provide stakeholders with more immersive and interactive experiences, enabling them to explore and interact with requirements in a more dynamic and intuitive manner. For example, interactive prototypes and simulations can allow stakeholders to visualize how the system will behave under different scenarios, facilitating feedback and validation of requirements in real-time. Visualization techniques, such as diagrams, charts, and graphs, can also be used to represent complex information in a more intuitive and accessible format. By presenting requirements visually, stakeholders can gain a better understanding of the relationships and dependencies between different requirements, enabling more informed decision-making and prioritization (Alshahrani *et al*., 2024). In addition to providing interactive interfaces and visualization tools, HCI principles can also facilitate a more iterative and participatory approach to requirement gathering. Iterative requirements elicitation involves breaking down the requirements elicitation process into smaller, more manageable iterations or cycles, each focusing on a specific set of requirements or functionalities. By soliciting feedback from stakeholders at each iteration, software development teams can incrementally refine and validate requirements, ensuring that they accurately reflect stakeholders' needs and expectations (Gupta *et al*., 2022). HCI techniques, such as usability testing and participatory design workshops, can be used to engage stakeholders in the requirements elicitation process actively. By involving stakeholders in the design and evaluation of interactive interfaces and visualization tools, software development teams can ensure that the tools meet stakeholders' needs and preferences and are aligned with the goals and objectives of the project.

Furthermore, HCI principles emphasize the importance of collaboration and communication between stakeholders throughout the requirements elicitation process. By fostering a collaborative environment where stakeholders can share ideas, provide feedback, and work together to co-create requirements, software development teams can ensure that all relevant perspectives are considered and that the final set of requirements reflects the collective interests of all stakeholders (Permatasari *et al*., 2021; Butt *et al*., 20023). Leveraging advancements in technology, such as Natural Language Processing (NLP), Machine Learning (ML), and Human-Computer Interaction (HCI), can significantly enhance the accuracy, efficiency, and effectiveness of requirements elicitation processes in software development. By automating the analysis of textual requirements, predicting potential changes in requirements, and developing interactive interfaces and visualization tools for stakeholder engagement, software development teams can streamline the requirements (Umar and Lano, 2024)

2.2. Incorporating Context-Awareness in Requirements Elicitation

In the realm of software development, requirements elicitation serves as a foundational step in understanding stakeholders' needs and translating them into system specifications. However, the effectiveness of this process heavily relies on the consideration of contextual factors that surround the project. Incorporating context-awareness into requirements elicitation involves recognizing and adapting to the organizational, cultural, and environmental context in which the software will operate (Khannur[1] and Hiremath, 2023).

The organizational context encompasses factors such as the structure, goals, policies, and procedures of the organization commissioning the software project. Different organizations have unique cultures, hierarchies, and decision-making processes that can significantly influence the requirements elicitation process. For instance, in a highly bureaucratic organization, decision-making may be centralized, requiring extensive documentation and formal approvals for requirements changes. In contrast, in a startup environment, decision-making may be more agile and decentralized, necessitating rapid iterations and flexibility in requirements gathering. Cultural context refers to the values, beliefs, norms, and behaviors shared by members of a particular community or society (Akpa *et al*., 2021). Cultural differences can manifest in various aspects of software requirements, including language preferences,

communication styles, and user expectations. For example, in collectivist cultures, where group harmony and consensus are prioritized, stakeholders may be more inclined towards collaborative decision-making and consensus-building approaches in requirements elicitation. In contrast, in individualistic cultures, where personal autonomy and achievement are valued, stakeholders may prefer more assertive and independent roles in the elicitation process. The environmental context encompasses external factors such as market dynamics, industry trends, legal regulations, and technological advancements that may impact the software project. Understanding the environmental context is crucial for anticipating future challenges, opportunities, and constraints that may influence the requirements of the system (Elsawah *et al*., 2020). For example, in a rapidly evolving technological landscape, stakeholders may prioritize requirements related to interoperability, scalability, and security to ensure the system remains competitive and resilient against emerging threats and opportunities.

Context-awareness in requirements elicitation involves selecting and adapting elicitation techniques to suit the specific needs, preferences, and constraints of the project context (Lim *et al*., 2021). Different contexts may require different approaches to gathering requirements effectively. For example, in a highly regulated industry such as healthcare or finance, stakeholders may have strict compliance requirements that necessitate structured interviews, documentation, and traceability mechanisms to ensure regulatory compliance. In contrast, in a creative industry such as digital media or entertainment, stakeholders may prefer more exploratory and experimental approaches, such as prototyping, brainstorming sessions, and user storytelling techniques. Incorporating context-awareness also entails designing the requirements elicitation process to be flexible and adaptable to changing project contexts (Aradea *et al*., 2023). This may involve incorporating feedback loops, iterative cycles, and continuous improvement mechanisms to accommodate evolving stakeholder needs and project dynamics. For example, Agile methodologies such as Scrum and Kanban emphasize adaptive planning, collaboration, and responsiveness to change, allowing development teams to iterate on requirements gathering activities in response to emerging insights and feedback from stakeholders.

Implicit requirements are requirements that are not explicitly stated but are implied or inferred from the context of the project (Waltz *et al*., 2020). These requirements are often tacit knowledge embedded within the organization or domain expertise of stakeholders. Identifying implicit requirements is crucial for ensuring the completeness, accuracy, and relevance of the requirements specification. Failure to capture implicit requirements may result in gaps, misunderstandings, and mismatches between the system delivered and stakeholders' expectations. Context-awareness involves employing techniques and approaches to uncover implicit requirements that may not be readily apparent through traditional elicitation methods. This may involve conducting stakeholder interviews, observations, workshops, and ethnographic studies to gain insights into stakeholders' tacit knowledge, beliefs, and assumptions about the system. Additionally, techniques such as context mapping, persona development, and scenario-based analysis can help uncover implicit requirements by contextualizing user needs and behaviors within the broader socio-cultural and organizational context (Rafliana *et al*., 2022).

Incorporating context-awareness in requirements elicitation is essential for ensuring that software systems meet the needs, expectations, and constraints of their intended users and stakeholders. By considering the organizational, cultural, and environmental context, adapting elicitation strategies and techniques accordingly, and identifying implicit requirements crucial for system success, software development teams can enhance the accuracy, relevance, and effectiveness of the requirements elicitation process (Alankarage *et al*., 2023). Context-awareness enables stakeholders to collaboratively co-create requirements that reflect the diverse perspectives, values, and priorities of the project context, ultimately leading to the development of software systems that are better aligned with stakeholders' needs and objectives.

2.3. Addressing Conflicting Requirements

Conflicting requirements are a common challenge in software development projects, where different stakeholders may have competing priorities or objectives (Iqbal *et al*., 2020). Effectively managing conflicting requirements is essential for ensuring the success of the project and the satisfaction of all stakeholders involved.

Requirements prioritization involves determining the relative importance or priority of different requirements based on predefined criteria. Multi-criteria decision-making techniques provide a systematic approach to prioritizing requirements by considering multiple factors simultaneously, Techniques such as Analytic Hierarchy Process (AHP), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), and Weighted Sum Model (WSM) are commonly used in requirements prioritization (Yannis *et al*., 2020). These techniques enable stakeholders to assign weights to different criteria, such as business value, technical feasibility, and user impact, and evaluate requirements based on these criteria. By applying multi-criteria decision-making techniques, software development teams can make informed decisions about which requirements to prioritize, taking into account the diverse perspectives and objectives

of stakeholders. This systematic approach helps minimize bias and subjectivity in the prioritization process, leading to more transparent and defensible decisions. Effective requirements prioritization requires collaboration and communication among stakeholders to ensure alignment and consensus on priorities (Obiuto *et al*., 2024). Stakeholders from different departments, roles, and levels of the organization should be involved in the prioritization process to provide diverse perspectives and insights. Collaborative prioritization workshops, focus groups, and stakeholder interviews are common techniques for engaging stakeholders in the prioritization process. These sessions allow stakeholders to discuss, debate, and negotiate the relative importance of different requirements, fostering a shared understanding and ownership of the prioritization decisions. By involving stakeholders in the prioritization process, software development teams can build consensus, mitigate conflicts, and ensure that the priorities reflect the collective interests of all stakeholders (Emeka-Okoli *et al*., 2024). Collaborative prioritization also enhances stakeholder buy-in and commitment to the project, increasing the likelihood of successful implementation and adoption of the software system.

Conflicting requirements often arise due to differences in stakeholders' perspectives, priorities, and objectives. Consensus-building strategies are essential for resolving conflicts and reaching agreement on prioritization decisions. Techniques such as brainstorming, negotiation, and compromise can be used to facilitate consensus-building among stakeholders (Fasihullah *et al*., 2023). By encouraging open dialogue and constructive communication, software development teams can identify common ground, explore alternative solutions, and find mutually acceptable compromises to resolve conflicts. Consensus-building should be guided by principles of inclusivity, transparency, and fairness to ensure that all stakeholders have the opportunity to voice their opinions and contribute to the decision-making process. Facilitators or mediators may be appointed to help manage conflicts, facilitate discussions, and guide stakeholders towards consensus. In situations where conflicts cannot be resolved through consensus-building alone, trade-off analysis techniques can be used to reconcile conflicting requirements by weighing the costs, benefits, and consequences of different options. Techniques such as cost-benefit analysis, risk analysis, and impact assessment can help stakeholders evaluate the trade-offs involved in prioritizing one requirement over another. By quantifying and comparing the potential risks and rewards associated with each option, software development teams can make informed decisions about how to allocate resources and prioritize requirements. Trade-off analysis requires careful consideration of factors such as project constraints, stakeholder preferences, and organizational goals. By involving stakeholders in the analysis process and providing them with relevant data and information, software development teams can increase transparency and trust in the decision-making process (Felzmann *et al*., 2020).

Addressing conflicting requirements requires a systematic approach that combines requirements prioritization and conflict resolution mechanisms. By applying multi-criteria decision-making techniques and collaborating with stakeholders in the prioritization process, software development teams can ensure that requirements are prioritized based on objective criteria and stakeholder consensus. Conflict resolution mechanisms such as consensus-building strategies and trade-off analysis help reconcile conflicting requirements by fostering open dialogue, exploring alternative solutions, and evaluating the trade-offs involved (Johansson *et al*., 2022). By actively managing conflicts and reaching agreement on prioritization decisions, software development teams can enhance stakeholder satisfaction, minimize project risks, and increase the likelihood of project success.

2.4. Emphasizing Iterative Refinement and Validation

The traditional approach of gathering requirements at the beginning of a project and then proceeding with development often leads to inefficiencies and mismatches between stakeholder expectations and the final product. Emphasizing iterative refinement and validation is essential for addressing these challenges, ensuring that requirements are continuously reviewed, refined, and validated throughout the development lifecycle (Davey, 2022).

Continuous refinement of requirements involves an ongoing process of reviewing, revising, and updating requirements to reflect evolving stakeholder needs, project constraints, and environmental changes (Jardim and Currey, 2023). This iterative approach acknowledges that requirements are inherently dynamic and subject to change throughout the development lifecycle. Agile methodologies, such as Scrum, Kanban, and Extreme Programming (XP), advocate for iterative development cycles with frequent deliveries of working software (Tetteh, 2024). These methodologies emphasize continuous collaboration between cross-functional teams and stakeholders to prioritize requirements, address emerging issues, and adapt to changing requirements. In Agile development, requirements are captured as user stories or features, which are prioritized and scheduled for implementation in short iterations called sprints. At the end of each sprint, stakeholders review the delivered functionality and provide feedback, which informs subsequent iterations and refinements to the requirements. Evolutionary prototyping is another approach to continuous refinement of requirements, where a series of prototypes are developed and refined iteratively based on stakeholder feedback. Prototypes serve as tangible representations of the system, allowing stakeholders to visualize and interact with the

proposed functionality early in the development process (Kleinsmann and Ten Bhömer, 2020). Through iterative prototyping, stakeholders can identify usability issues, clarify requirements, and validate design decisions before significant investments are made in development. This iterative approach helps mitigate the risk of misunderstandings and ensures that the final product meets stakeholders' expectations.

Feedback loops play a crucial role in facilitating stakeholder involvement, communication, and collaboration throughout the development lifecycle (Poger *et al*., 2020). By providing opportunities for stakeholders to review and revise requirements iteratively, feedback loops ensure that the system remains aligned with stakeholders' needs and expectations. Effective stakeholder engagement is essential for establishing feedback loops and soliciting meaningful input from stakeholders throughout the development process. Stakeholders include end-users, customers, business analysts, product owners, and other relevant parties who have a vested interest in the success of the project (Scheepers *et al*., 2022). Various techniques, such as stakeholder interviews, focus groups, surveys, and user testing sessions, can be used to gather feedback from stakeholders. These sessions provide stakeholders with opportunities to express their preferences, identify issues, and suggest improvements to the system. Continuous integration and delivery (CI/CD) practices enable software development teams to deliver incremental changes to the system frequently and reliably (Mowad *et al*., 2022). By automating the build, testing, and deployment processes, CI/CD pipelines streamline the delivery of new features and updates to stakeholders in a timely manner. CI/CD pipelines incorporate feedback loops by automatically triggering tests and validations whenever changes are made to the codebase. If any issues are detected, developers can quickly address them and iterate on the changes, ensuring that the system remains stable and functional (Heeager and Nielsen, 2020).

Emphasizing iterative refinement and validation enhances the adaptability and resilience of the system by fostering a culture of continuous learning, improvement, and adaptation to change. Iterative development encourages a mindset of continuous learning and improvement among team members (Alami *et al*., 2022). By reflecting on past experiences, evaluating outcomes, and incorporating feedback from stakeholders, software development teams can identify areas for improvement and make iterative refinements to the development process. In today's rapidly changing business environment, software systems must be adaptable and resilient to cope with unforeseen challenges and opportunities. Emphasizing iterative refinement and validation enables software development teams to respond quickly to changing requirements, market conditions, and technological advancements (Akhtar and Kumar, 2024). Agile methodologies emphasize adaptability by prioritizing responsiveness to change over adherence to rigid plans and specifications. By embracing change as a natural part of the development process, software development teams can capitalize on opportunities, mitigate risks, and deliver value to stakeholders more effectively. Emphasizing iterative refinement and validation is essential for ensuring the success of software development projects in today's dynamic and complex environment (Vinay, 2024). Continuous refinement of requirements, feedback loops for stakeholder review and revision, and the enhancement of adaptability and resilience of the system are key components of this iterative approach (Andriyani *et al*., 2024). By adopting a mindset of continuous learning, improvement, and adaptation to change, software development teams can deliver high-quality software systems that meet the evolving needs and expectations of stakeholders.

3. Conclusion

In this essay, we have explored the importance of advancements and innovations in requirements elicitation, focusing on the development of a comprehensive conceptual model. By incorporating cutting-edge technologies and adopting iterative refinement and validation approaches, software development teams can enhance the accuracy, efficiency, and effectiveness of the requirements elicitation process. Advancements and innovations in requirements elicitation play a crucial role in the development of high-quality software systems. By integrating cutting-edge technologies and adopting iterative refinement and validation approaches, software development teams can enhance the accuracy, efficiency, and effectiveness of the requirements elicitation process, ultimately leading to increased stakeholder satisfaction and successful project outcomes.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

Reference

[1] Ajmal, M.M., Khan, M., Gunasekaran, A. and Helo, P.T., 2022. Managing project scope creep in construction industry. *Engineering, Construction and Architectural Management*, *29*(7), pp.2786-2809.

[2] Akhtar, N. and Kumar, W., 2024. Maximizing Business Growth with Big Data: Strategies and Project Excellence in Analytics Scaling. *Journal Environmental Sciences And Technology*, *3*(1), pp.477-483.

[3] Akpa, V.O., Asikhia, O.U. and Nneji, N.E., 2021. Organizational culture and organizational performance: A review of literature. *International Journal of Advances in Engineering and Management*, *3*(1), pp.361-37.

[4] Alami, A., Krancher, O. and Paasivaara, M., 2022. The journey to technical excellence in agile software development. *Information and Software Technology*, *150*, p.106959.

[5] Alankarage, S., Chileshe, N., Samaraweera, A., Rameezdeen, R. and Edwards, D.J., 2023. Guidelines for using a case study approach in construction culture research: Application to BIM-enabled organizations. *Journal of Construction Engineering and Management*, *149*(9), p.05023009.

[6] Aldave, A., Vara, J.M., Granada, D. and Marcos, E., 2019. Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *Journal of Systems and Software*, *157*, p.110396.

[7] Aljohani, A., 2023. Predictive analytics and machine learning for real-time supply chain risk mitigation and agility. *Sustainability*, *15*(20), p.15088.

[8] Al-Saqqa, S., Sawalha, S. and AbdelNabi, H., 2020. Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, *14*(11).

[9] Alshahrani, R., Yenugula, M., Algethami, H., Alharbi, F., Goswami, S.S., Naveed, Q.N., Lasisi, A., Islam, S., Khan, N.A. and Zahmatkesh, S., 2024. Establishing the fuzzy integrated hybrid MCDM framework to identify the key barriers to implementing artificial intelligence-enabled sustainable cloud system in an IT industry. *Expert systems with applications*, *238*, p.121732.

[10] Andriyani, Y., Yohanitas, W.A. and Kartika, R.S., 2024. Adaptive innovation model design: Integrating agile and open innovation in regional areas innovation. *Journal of Open Innovation: Technology, Market, and Complexity*, *10*(1), p.100197.

[11] Aradea, S. I. and Surendro, K., 2023. ARAS: adaptation requirements for adaptive systems: Handling runtime uncertainty of contextual requirements. *Automated Software Engineering*, *30*(1), p.2.

[12] Atoum, I., Baklizi, M.K., Alsmadi, I., Otoom, A.A., Alhersh, T., Ababneh, J., Almalki, J. and Alshahrani, S.M., 2021. Challenges of software requirements quality assurance and validation: A systematic literature review. *IEEE Access*, *9*, pp.137613-137634.

[13] Baviskar, D., Ahirrao, S., Potdar, V. and Kotecha, K., 2021. Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. *IEEE Access*, *9*, pp.72894-72936.

[14] Behutiye, W., Rodríguez, P., Oivo, M., Aaramaa, S., Partanen, J. and Abhervé, A., 2022. Towards optimal quality requirement documentation in agile software development: A multiple case study. *Journal of Systems and Software*, *183*, p.111112.

[15] Behutiye, W., Seppänen, P., Rodríguez, P. and Oivo, M., 2020, April. Documentation of quality requirements in agile software development. In *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering* (pp. 250-259).

[16] Betti, N. and Sarens, G., 2021. Understanding the internal audit function in a digitalised business environment. *Journal of Accounting & Organizational Change*, *17*(2), pp.197-216.

[17] Bukhsh, F.A., Bukhsh, Z.A. and Daneva, M., 2020. A systematic literature review on requirement prioritization techniques and their empirical evaluation. *Computer Standards & Interfaces*, *69*, p.103389.

[18] Butt, S.A., Suran, S., Pappel, I., Smærup, M., Krimmer, R. and Draheim, D., 2023. A Digital Collaborative Platform for the Silver Economy: Functionalities Required by Stakeholders in a Multinational Baltic Sea Region Project. *Digital Government: Research and Practice*, *4*(2), pp.1-20.

[19] Challapalli, S., 2023. Unified Modeling Language for Requirements Engineering, Strategies and Best Practices for FinTech and Beyond. *Asian Journal of Research in Computer Science*, *16*(3), pp.87-102.

[20] Davey, J., 2022. Prioritizing Human-Centered Design: Integrating Ergonomics and Medical Device Development for Optimal User Experience. *Cosmic Journal of Biology*, *1*(1), pp.481-494.

[21] Ebirim, W., Montero, D.J.P., Ani, E.C., Ninduwezuor-Ehiobu, N., Usman, F.O. and Olu-lawal, K.A., 2024. THE ROLE OF AGILE PROJECT MANAGEMENT IN DRIVING INNOVATION IN ENERGY-EFFICIENT HVAC SOLUTIONS. *Engineering Science & Technology Journal*, *5*(3), pp.662-673.

[22] Elsawah, S., Hamilton, S.H., Jakeman, A.J., Rothman, D., Schweizer, V., Trutnevyte, E., Carlsen, H., Drakes, C., Frame, B., Fu, B. and Guivarch, C., 2020. Scenario processes for socio-environmental systems analysis of futures: a review of recent efforts and a salient research agenda for supporting decision making. *Science of the Total Environment*, *729*, p.138393.

[23] Emeka-Okoli, S., Nwankwo, T.C., Otonnah, C.A. and Nwankwo, E.E., 2024. Effective Stakeholder Relationship Management In The Oil & Gas Sector: A Conceptual And Review Perspective. *Finance & Accounting Research Journal*, *6*(3), pp.372-383.

[24] Estévez, R.A., Espinoza, V., Ponce Oliva, R.D., Vásquez-Lavín, F. and Gelcich, S., 2021. Multi-criteria decision analysis for renewable energies: research trends, gaps and the challenge of improving participation. *Sustainability*, *13*(6), p.3515.

[25] Ezeigweneme, C.A., Umoh, A.A., Ilojianya, V.I. and Adegbite, A.O., 2024. Telecommunications energy efficiency: optimizing network infrastructure for sustainability. *Computer Science & IT Research Journal*, *5*(1), pp.26-40.

[26] Farayola, O.A., Abdul, A.A., Irabor, B.O. and Okeleke, E.C., 2023. INNOVATIVE BUSINESS MODELS DRIVEN BY AI TECHNOLOGIES: A REVIEW. *Computer Science & IT Research Journal*, *4*(2), pp.85-110.

[27] Fasihullah, A., Awan, B. and Hulio, A.K., 2023. Negotiating in the Digital Age: Exploring the Role of Technology in Modern Negotiations. *Law and Policy Review*, *2*(1), pp.61-86.

[28] Felzmann, H., Fosch-Villaronga, E., Lutz, C. and Tamò-Larrieux, A., 2020. Towards transparency by design for artificial intelligence. *Science and engineering ethics*, *26*(6), pp.3333-3361.

[29] García-López, D., Segura-Morales, M. and Loza-Aguirre, E., 2020. Improving the quality and quantity of functional and non-functional requirements obtained during requirements elicitation stage for the development of e-commerce mobile applications: an alternative reference process model. *IET software*, *14*(2), pp.148-158.

[30] Govindaras, B., Wern, T.S., Kaur, S., Haslin, I.A. and Ramasamy, R.K., 2023. Sustainable environment to prevent burnout and attrition in project management. *Sustainability*, *15*(3), p.2364.

[31] Gregory, A.J., Atkins, J.P., Midgley, G. and Hodgson, A.M., 2020. Stakeholder identification and engagement in problem structuring interventions. *European journal of operational research*, *283*(1), pp.321-340.

[32] Gupta, A., Poels, G. and Bera, P., 2022. Using conceptual models in agile software development: a possible solution to requirements engineering challenges in agile projects. *IEEE Access*, *10*, pp.119745-119766.

[33] Hassan, F.U. and Le, T., 2020. Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *12*(2), p.04520009.

[34] Heeager, L.T. and Nielsen, P.A., 2020. Meshing agile and plan-driven development in safety-critical software: a case study. *Empirical Software Engineering*, *25*(2), pp.1035-1062.

[35] Holden, R.J., Abebe, E., Hill, J.R., Brown, J., Savoy, A., Voida, S., Jones, J.F. and Kulanthaivel, A., 2022. Human factors engineering and human-computer interaction: supporting user performance and experience. *Clinical informatics study guide: text and review*, pp.119-132.

[36] Iqbal, J., Ahmad, R.B., Khan, M., Alyahya, S., Nizam Nasir, M.H., Akhunzada, A. and Shoaib, M., 2020. Requirements engineering issues causing software development outsourcing failure. *PloS one*, *15*(4), p.e0229785.

[37] Issa, T. and Isaias, P., 2022. Usability and human–computer interaction (hci). In *Sustainable design: HCI, usability and environmental concerns* (pp. 23-40). London: Springer London.

[38] Jardim, E. and Currey, R.J., 2023. The MSC Fisheries Standard Review policy development process. *Marine Policy*, *157*, p.105855.

[39] Johansson, A., Lindahl, K.B. and Zachrisson, A., 2022. Exploring prospects of deliberation in intractable natural resource management conflicts. *Journal of environmental management*, *315*, p.115205.

[40] Kasauli, R., Knauss, E., Horkoff, J., Liebel, G. and de Oliveira Neto, F.G., 2021. Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software*, *172*, p.110851.

[41] Khan, M.F.I. and Masum, A.K.M., 2024. Predictive Analytics And Machine Learning For Real-Time Detection Of Software Defects And Agile Test Management. *Educational Administration: Theory and Practice*, *30*(4), pp.1051-1057.

[42] Khannur[1], A. and Hiremath, M., 2023, July. Requirements Engineering Framework using Contextualization, Profiling, and Modelling. In *CS & IT Conference Proceedings* (Vol. 13, No. 12). CS & IT Conference Proceedings.

[43] Kleinsmann, M. and Ten Bhömer, M., 2020. The (new) roles of prototypes during the co-development of digital product service systems. *International Journal of Design*, *14*(1), pp.65-79.

[44] Lim, S., Henriksson, A. and Zdravkovic, J., 2021. Data-driven requirements elicitation: A systematic literature review. *SN Computer Science*, *2*(1), p.16.

[45] Mowad, A.M., Fawareh, H. and Hassan, M.A., 2022, November. Effect of using continuous integration (ci) and continuous delivery (cd) deployment in devops to reduce the gap between developer and operation. In *2022 International Arab Conference on Information Technology (ACIT)* (pp. 1-8). IEEE.

[46] Naeem, M., Ozuem, W., Howell, K. and Ranfagni, S., 2023. A step-by-step process of thematic analysis to develop a conceptual model in qualitative research. *International Journal of Qualitative Methods*, *22*, p.16094069231205789.

[47] Obiuto, N.C., Ebirim, W., Ninduwezuor-Ehiobu, N., Ani, E.C., Olu-lawal, K.A. and Ugwuanyi, E.D., 2024. Integrating sustainability into hvac project management: challenges and opportunities. *Engineering Science & Technology Journal*, *5*(3), pp.873-887.

[48] Oladeinde, M., Hassan, A.O., Farayola, O.A., Akindote, O.J. and Adegbite, A.O., 2023. REVIEW OF IT INNOVATIONS, DATA ANALYTICS, AND GOVERNANCE IN NIGERIAN ENTERPRISES. *Computer Science & IT Research Journal*, *4*(3), pp.300-326.

[49] Pargaonkar, S., 2023. Synergizing Requirements Engineering and Quality Assurance: A Comprehensive Exploration in Software Quality Engineering. *International Journal of Science and Research (IJSR)*, *12*(8), pp.2003-2007.

[50] Permatasari, A., Dhewanto, W. and Dellyana, D., 2021. A proposed model of value co-creation through multi-stakeholder collaboration in domestic product development. *Business: Theory and Practice*, *22*(2), pp.414-425.

[51] Poger, J.M., Mayer, V., Duru, O.K., Nauman, B., Holderness, H., Warren, N., Vasquez, C., Bibi, S., Rasmussen-Torvik, L.J., Hosseinian, Z. and Shi, L., 2020. Network engagement in action: stakeholder engagement activities to enhance patient-centeredness of research. *Medical care*, *58*, pp.S66-S74.

[52] Rafliana, I., Jalayer, F., Cerase, A., Cugliari, L., Baiguera, M., Salmanidou, D., Necmioğlu, Ö., Ayerbe, I.A., Lorito, S., Fraser, S. and Løvholt, F., 2022. Tsunami risk communication and management: Contemporary gaps and challenges. *International Journal of Disaster Risk Reduction*, *70*, p.102771.

[53] Rasheed, A., Zafar, B., Shehryar, T., Aslam, N.A., Sajid, M., Ali, N., Dar, S.H. and Khalid, S., 2021. Requirement engineering challenges in agile software development. *Mathematical Problems in Engineering*, *2021*, pp.1-18.

[54] Ribeiro, C. and Berry, D., 2020. The prevalence and severity of persistent ambiguity in software requirements specifications: Is a special effort needed to find them?. *Science of Computer Programming*, *195*, p.102472.

[55] Saeeda, H., Dong, J., Wang, Y. and Abid, M.A., 2020. A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project. *Journal of Software: Evolution and Process*, *32*(7), p.e2247.

[56] Scheepers, H., McLoughlin, S. and Wijesinghe, R., 2022. Aligning stakeholders perceptions of project performance: The contribution of Business Realisation Management. *International Journal of Project Management*, *40*(5), pp.471-480.

[57] Siakas, E., Rahanu, H., Georgiadou, E. and Siakas, K., 2021. Towards reducing communication gaps in multicultural and global requirements elicitation. In *Systems, Software and Services Process Improvement: 28th European Conference, EuroSPI 2021, Krems, Austria, September 1–3, 2021, Proceedings 28* (pp. 257-277). Springer International Publishing.

[58] Siakas, E., Rahanu, H., Georgiadou, E. and Siakas, K., 2022, August. Requirements volatility in multicultural situational contexts. In *European Conference on Software Process Improvement* (pp. 633-655). Cham: Springer International Publishing.

[59] Sidaoui, K., Mahr, D. and Odekerken-Schröder, G., 2024. Generative AI in Responsible Conversational Agent Adoption: Guidelines for Service Managers. *Organizational Dynamics*, p.101045.

[60] Singgalen, Y.A., 2024. Coastal and Marine Tourism Monitoring System Design using Rapid Application Development (RAD). *Journal of Information System Research (JOSH)*, *5*(2), pp.468-479.

[61] Sonko, S., Monebi, A.M., Etukudoh, E.A., Osasona, F., Atadoga, A. and Daudu, C.D., 2024. REVIEWING THE IMPACT OF EMBEDDED SYSTEMS IN MEDICAL DEVICES IN THE USA. *International Medical Science Research Journal*, *4*(2), pp.158-169.

[62] Susnjak, T., Ramaswami, G.S. and Mathrani, A., 2022. Learning analytics dashboard: a tool for providing actionable insights to learners. *International Journal of Educational Technology in Higher Education*, *19*(1), p.12.

[63] Tetteh, S.G., 2024. Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *Asian Journal of Research in Computer Science*, *17*(5), pp.30-42

[64] Tyagi, A.K. and Chahal, P., 2022. Artificial intelligence and machine learning algorithms. In *Research anthology on machine learning techniques, methods, and applications* (pp. 421-446). IGI Global.

[65] Umar, M.A. and Lano, K., 2024. Advances in automated support for requirements engineering: a systematic literature review. *Requirements Engineering*, pp.1-31.

[66] Vinay, S.B., 2024. Data Scientist Competencies and Skill Assessment: A Comprehensive Framework. *Journal ID*, *1660*, p.1544.

[67] Waltz, M.J., Moberly, H.K. and Carrigan, E.E., 2020. Identifying information literacy skills and behaviors in the curricular competencies of health professions. *Journal of the Medical Library Association: JMLA*, *108*(3), p.463.

[68] Yannis, G., Kopsacheili, A., Dragomanovits, A. and Petraki, V., 2020. State-of-the-art review on multi-criteria decision-making in the transport sector. *Journal of traffic and transportation engineering (English edition)*, *7*(4), pp.413-431.

[69] Zaki-Ismail, A., Osama, M., Abdelrazek, M., Grundy, J. and Ibrahim, A., 2022. RCM-extractor: an automated NLP-based approach for extracting a semi formal representation model from natural language requirements. *Automated Software Engineering*, *29*(1), p.10.

[70] Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K.J., Ajagbe, M.A., Chioasca, E.V. and Batista-Navarro, R.T., 2021. Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys (CSUR)*, *54*(3), pp.1-41.