Pakistan Journal of Scientific Research
Multidisciplinary & Peer Reviewed
Volume: 5, Number: 2, Pages: 68-73, Year:2025

Wastes Generated during Requirement Engineering in AI Era

Basit Amin, Sumaira Nazir, Nargis Fatima, and Naveed Ahmad

Department of Software Engineering, National University of Modern Languages, Islamabad, Pakistan

Correspondence: basit.amin@numl.edu.pk; sunazir@numl.edu.pk; nfatima@numl.edu.pk

Submitted: 10-07-2025**, Revised:** 17-11-2025, **Accepted:** 18-12-2025

Abstract

Requirement Engineering is a core software engineering activity that acts as the foundation of quality software. In this modern AI Era, it becomes increasingly critical and is prone to generating various forms of waste. For instance, partially cognitive load, analysis paralysis, partially done work, and gold plating. It can impact not only the requirement engineer's mental well-being but also decrease client satisfaction. Waste identification and elimination have been explored in the literature; however, there is a gap in identifying waste generated during requirement engineering. There is a dire need to identify waste generated during RE in this modern AI era. To achieve this, the current study presents a systematic literature review protocol to identify waste generated during RE. This study will be beneficial to requirements engineers in managing the identified wastes and improving their productivity. It will also provide research opportunities for researchers to provide solutions to the identified waste. The researchers can extend their research to other software engineering activities that have not yet been explored**.**

Keywords: Requirement Engineering, Artificial Intelligence, Wastes, Green Software Engineering, Lean, Prompt Requirement Engineering.

1. Introduction

Distributed The requirement engineering is the core activity that involves requirement elicitation, analysis, specification, validation, and change management. Each phase of requirement engineering has a specific role in generating high-quality requirements [1], [2]. Figure I shows the phases involved in RE.

Requirements engineering in this modern AI Era depends on human aspects in coordination with AI [3]. The requirement engineers are utilising AI tools to generate requirements. Generative AI presently plays a critical role in this context. Presently, the field of prompt requirement engineering is mainly focused on the software engineering industry [3].

The requirement engineering, though performed using LLMs and AI, still generates waste [4]. Waste can be "any Blockages" or "everything that does not add value" [5]. Lean principles guide for identifying and mitigating waste [6]. Researchers have explored various intuitions about waste [7]. Waste generated during software development is reported by [8]. Wastes generated during code review are specifically reported by [9].

 https://doi.org/10.57041/z065p111

Various waste can be generated during software development activities, such as lack of knowledge sharing, task switching, waiting, partially completed work, defects, cognitive load, and knowledge silos [8], [7]. Requirement engineering is a core software engineering discipline that aims to systematically develop a requirement specification for underlying projects. It involves multiple activities that consume resources but do not provide value either to requirement engineers or clients [3].

In the context of lean [1], [10], work has been done to recognise waste in numerous fields [11]. The concept of waste identification is little explored [12]. There are limited studies on identifying waste from a software engineering perspective [4], [5], [8], [10]. Little research has been conducted on identifying waste in modern code review [4]; however, the requirement has not yet been explored in this context. Therefore, this research aims to investigate perceptions of waste, particularly regarding RE in the AI Era [10].

The paper is organised as follows: the introduction is in section I, and section II covers related work. Section III presents the research method. Section 4 highlights the results. Section 5 mentioned the contribution. Section 6 highlights the study's conclusion and future work opportunities.

2. Background

Software engineering is the systematic process for delivering quality software in time [13]. Multiple activities can be performed collectively in this regard [14]. For instance, Requirement Engineering, Modelling, development, Reviews, testing, Development and Maintenance. These activities produced multiple wastes [15] that can cause other issues like mental distress and delays. The notion of lean (waste) was initially presented by Toyota in the 1980s [4], [6]. Then this lean paradigm was moved from lean manufacturing to lean software development [16] and also referred to as green software engineering [17]. Researchers have provided efforts to recognise waste and strategies to manage it [18]

Multiple wastes such as "knowledge loss", "rework", "defects", "cognitive load", and "waiting" etc. are generated during software development [4], [18]. As mentioned before, there are multiple activities that are performed for successful software engineering, each activity generates wastes [15]. Few studies have been conducted on the recognition and elimination of waste in overall software development [8], [10]. Software development involves multiple core activities that can produce waste, for instance, requirements engineering, design, coding, testing, code reviews, deployment, and maintenance [19]. Requirement engineering is a core software engineering activity that is presently supported by AI tools and techniques, but it can still generate waste that can lead to other issues [10].

According to the literature analysis, no studies have been conducted to date to identify waste in the context of requirement engineering [10]. This research presents a new context for waste identification in software requirement engineering.

3. Research Methodology

To identify wastes generated during RE, particularly in the AI Era, this research will follow the guidelines given by [20]. In this paper the SLR protocol is presented.

3.1 Research Questions

The study research question is as below

RQ1: Which actions were considered as waste in requirement Engineering in AI Era?
RQ2: What is frequency of identified waste i.e. highly reported in the literature?

The research objectives are given below.
RO1: To identify the waste generated during requirement engineering in AI Era.
RO2: To identify the wastes highly reported in literature.

3.2 Keywords

The keywords in Table I were considered for the SLR.

Table I Keywords

Keywords	**Synonyms**
Requirement Engineering	Requirement Engineering, Software Requirement Engineering, Requirement Gathering, Requirement Elicitation, OR Requirement Specification OR Requirement Elicitation.
AI	Modern Age, LLM, Gen AI
Waste	Waste, Lean, Non-Value Add, Inefficient, Useless, Bottleneck.

3.3 Search String

The defined search string for the study is given below

((“Requirement Engineering” OR “Software Requirement Engineering” OR “Requirement Gathering” OR “Requirement Elicitation” OR “Requirement Specification” OR “Requirement Elicitation”) AND (“Waste” OR “Lean” OR “Non-Value Add” OR “Inefficient” OR “Useless” OR “Bottleneck”))

3.4 Data Sources

The databases that opted for searching are Scopus, Web of Science, IEEE, Springer, ScienceDirect and ACM. The search was performed for the papers published between 2020 to 2026.

3.5 Inclusion and Exclusion Criteria

To It is planned as per the guideline provided by [20]. It involves various inclusion parameters. The papers in which the study keywords were observed are included. The papers published from 2020 to 2026 were considered for study. The studies having no defined keywords were not considered. Moreover, regarding the exclusion plan, the paper having no defined keywords were not considered for SLR. Similarly, to attain the unique list of papers for SLR, it is planned to remove the replica copies of papers available in multiple databases. Likewise, the papers totally. The article solely directing seminars, discussions, conferences were not included for the study.

3.6 Quality Assessment

The quality of selected papers is evaluated using check sheet given by Kitchenham [21]. The details of checklist that is utilized are given in Table II.

Table II Quality Assessment Criteria

Question	**Answer**
Are the keywords specified or not?	Yes/No/Partial
Does this research extend the knowledge?	Yes/No/Partial
Does difficulties/profundity and detail of a competency is mentioned?	Yes/No/Partial
Dose strong as well as logical r reporting have been performed?	Yes/No/Partial

Doses multiple perspectives as well as background of a competency have been elaborated?	Yes/No/Partial

Guidelines given by Azhar [22] having a gauge were used while considering the Kitchenham checklist [21] for quality assessment. The gauge is given in Table III.

Table III Assessment through Gauge [22]

Gauge	Value
Yes	1
No	0
Partial	0.5

3.6 Data Extraction

The data extraction was performed using the guidelines provided by [21]. Data extraction template is shown in Table III. This form contains the Data Item, Extracted Data and comments to record the details of selected studies. Table III is particularly designed to record the waste generated during RE in AI Era. The data extraction supports the synthesis of data related to waste identification.

Table IV Data Extraction Form (Paper Information.)

Data Item	Extracted Data	Comments
ID	ID No. (S.Num.)	
Paper Title		
Author		
Type	Journal/ Conference	
Publisher	IEEE/ACM/etc.	
Field	Industry/ Academia	
Methodology	SLR/Survey/Experiment	
Selection Status	Included/Rejected	

Table V Data Extraction Form (Research Data)

Research Question	Extracted Data	Comments
Which actions are considered as waste in requirement Engineering in AI Era?	-Allocated ID to Paper -Reported Waste	
What is frequency of identified waste i.e. highly reported in the literature?	-Number of Research Articles reported that particular waste generated during RE.	

3.7 Data Synthesis

During data synthesis the data based on research question that is extracted is organized using thematic Analysis. The above-mentioned data extraction forms given in Table IV and Table V supported data synthesis and reporting.

4. Conclusion

The study provides the detailed steps that are followed to conduct systematic literature review with the aim of identifying the wastes generated during RE in AI era. This study aims to identify the

wastes that can affect mental well-being of requirement engineering and client satisfaction. The study contributed towards the Requirement Engineering Body of Knowledge (REBOK) and Software Engineering body of Knowledge (SEBOK).

5. Future Work

In the future, the study can be extended by identifying waste generated in other software engineering activities, such as software design, testing, deployment, and maintenance. Moreover, the present study can be validated by the industry. Moreover, the solution can be devised to minimise the consequences of reported waste.

Supplementary Materials:

Author Contributions: For research articles all four authors "Basit Amin, Sumaira Nazir, Nargis Fatima and Naveed Ahmad" contributed for conceptualization, research background, methodology, data collection and analysis of data, writing and Editing; Formal Review; "Nargis Fatima, Sumaira Nazir". All authors have read and agreed to the published version of the manuscript.".

Funding: This research received no external funding.

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflicts of interest."

References

1. M. Arif, S. Nazir, and N. Fatima, "Competencies of a Requirement Engineer -Systematic Literature Review Protocol," *Pakistan J. Eng. Technol.*, vol. 6, no. 1, pp. 69–73, 2023.
2. R. M. Ali, S. Nazir, and N. Fatima, "Understanding the Role Of Emotional Intelligence in Agile Teams in Context of Requirement Change Management," vol. 07, no. 04, pp. 2913–2927, 2025.
3. K. Liu, "Artificial Intelligence in Software Requirements Engineering : State-of-the-Art," no. December 2021, pp. 106–111, 2022.
4. N. Fatima, S. Nazir, and S. Chuprat, "Software engineering wastes-A perspective of modern code review," *ACM Int. Conf. Proceeding Ser.*, pp. 93–99, 2020.
5. A. Rehan, N. Fatima, and S. Nazir, "Waste Generated in Distributed Data Processing Systems : Strate- gies and Future Directions," vol. 2, pp. 82–91, 2025.
6. S. Mujtaba, R. Feldt, and K. Petersen, "Waste and lead time reduction in a software product customization process with value stream maps," *Proc. Aust. Softw. Eng. Conf. ASWEC*, pp. 139–148, 2010.
7. N. Fatima, S. Nazir, and S. Chuprat, "Knowledge sharing framework for modern code review to diminish software engineering waste," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 6, pp. 442–450, 2020.
8. T. Sedano, P. Ralph, and C. Peraire, "Software Development Waste," in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 2017, pp. 130–140.
9. Fatima, Nargis, Nazir, Suimaira, Chuprat Suriayati "Software Engineering Wastes – A

Perspective of Modern Code Review," in *The 3rd International Conference on Software Engineering and Information Management (ICSIM)*, 2020.

10. N. Fatima, S. Nazir, and S. Chuprat, "Assessing the Effectiveness of MCR-KSM for Waiting Waste Reduction : An Empirical Study," vol. 16, no. 9, 2025.

11. K. Presler-Marshall, S. Heckman, and K. T. Stolee, "Improving Grading Outcomes in Software Engineering Projects Through Automated Contributions Summaries," *Proc. - Int. Conf. Softw. Eng.*, pp. 259–270, 2023.

12. S. Nazir, N. Fatima, and S. Chuprat, "Modern code review benefits-primary findings of a systematic literature review," in *ACM International Conference Proceeding Series*, 2020, pp. 210–215.

13. P. Bourque and R. E. Fairley, *Guide to the software engineering - Body of knowledge.* 2014.

14. S. Nazir, N. Fatima, and S. Chuprat, "Individual Sustainability Barriers and Mitigation Strategies: Systematic Literature Review Protocol," in *2019 IEEE Conference on Open System, ICOS 2019*, 2019, pp. 1–5.

15. T. Sedano and P. Ralph, "Software Development Waste," in *Proc. IEEE/ACM 39th International Conference on Software Engineering*, 2017.

16. M. Poppendieck and T. Poppendieck, *Lean software development: An agile toolkit*. 2003.

17. S. R. Ahmad Ibrahim, J. Yahaya, and H. Sallehudin, "Green Software Process Factors: A Qualitative Study," *Sustain.*, vol. 14, no. 18, 2022.

18. H. Alahyari, T. Gorschek, and R. Berntsson, "An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations," *Inf. Softw. Technol.*, vol. 105, no. 7, pp. 78–94, 2019.

19. N. Fatima, S. Nazir, and S. Chuprat, "Knowledge sharing factors for modern code review to minimize software engineering waste," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 490–497, 2020.

20. B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3, Engineering," vol. 45, no. 4ve, p. 1051., 2007.

21. Kitchenham, S. L. Pfleeger, D. C. Jones, P.W.Hoaglin, K. El Emam, and J. B.A.Rosenberg, "Preliminary guidelines for empirical research in software engineering," 2002.

22. D. Azhar, E. Mendes, and P. Riddle, "A systematic review of web resource estimation.," *Proc. 8th Int. Conf. Predict. Model. Softw. Eng.*, p. (pp. 49-58)., 2012.

to people or property resulting from any ideas, methods, instructions or products referred to in the content.