



## Roadmap Analysis of Artificial Intelligence Engineering Method

Sandfreni<sup>1\*</sup>, Eko K. Budiardjo<sup>1,2</sup>

Department of Computer Science, Universitas Indonesia, Depok 16424, Jakarta, Indonesia

Department of Computer Science, Universitas Esa Unggul, Kebon Jeruk 11540, Jakarta, Indonesia

Corresponding Author Email: [sandfreni@esaunggul.ac.id](mailto:sandfreni@esaunggul.ac.id)



Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ria.380305>

### ABSTRACT

**Received:** 30 October 2023

**Revised:** 15 January 2024

**Accepted:** 7 February 2024

**Available online:** 21 June 2024

#### **Keywords:**

*AI Engineering, software engineering, AI engineering challenge, systematic literature review*

The characteristics of AI-based software have the potential to reshape traditional software development paradigms. Consequently, this study conducts a systematic literature review (SLR) within the field of AI Engineering to identify the unique challenges in software engineering for AI-based systems, which are transforming traditional software development paradigms. The scope of the SLR includes literature from academic journals and conference proceedings published between 2018 and 2023, selected through a rigorous process. The methodology involved using specific search keywords across databases such as Scopus, ScienceDirect, ACM Digital Library, and IEEE Xplore, with a stringent application of Kitchenham's inclusion and exclusion criteria to ensure a focused and relevant review. This review provides a consolidated summary of diverse research endeavors addressing challenges, issues, and methodologies relevant to AI-based software development. Highlighted topics encompass challenges in requirements engineering for AI-intensive system development, responsible software development (responsible AI), the formulation of a software engineering roadmap for responsible AI, the application of TrustOps as a risk management methodology in AI system development, the necessity of incorporating software engineering methods in AI-based systems, as well as studies exploring requirements engineering practices, AI-intensive system development, and the utilization of tools in machine learning model development. Key findings include the importance of recognizing ethical requirements in AI development, the role of risk management and ethical attributes, and the challenges of connecting requirements between software developers, data scientists, and machine learning specialists. This research provides valuable insights for practitioners and researchers involved in developing AI-based software to overcome existing challenges and apply appropriate methods in the development process.

## 1. INTRODUCTION

Artificial Intelligence (AI), a technological embodiment of human intelligence, has brought about a paradigm shift across various sectors, revolutionizing how we approach and solve complex problems. AI mirrors human cognitive abilities within software frameworks, enabling programs to display behaviors akin to human intelligence. As McLeod and Schell describe, AI endows programs with human-like behaviors. This transformation is evident in numerous fields, showcasing AI's versatility and its critical role in driving innovation and efficiency. With AI's integration into software engineering, now known as AI Engineering, there has been a significant enhancement in system efficiency and problem-solving capabilities. This integration signifies a pivotal evolution within the technological sphere, propelling forward innovative solutions in areas ranging from software development to project management.

AI has been used in various fields, therefore with the development of AI, when combined with software engineering or better known today as Artificial Intelligence Engineering

(AI Engineering), can create a system that is more efficient and able to better solve complex problems. AI Engineering has emerged as a pivotal discipline within the technology landscape, holding the potential to yield innovative solutions spanning various domains, including software development and project management [1].

The roots of AI Engineering can be traced back to the rapid strides made in the field of artificial intelligence. The evolution of machine learning, deep learning, and natural language processing has provided a robust foundation for the application of AI principles in software and systems development. The infusion of AI principles into the realm of AI Engineering carries the potential to redefine conventional software development paradigms. AI techniques are leveraged to enhance the efficiency and quality of software development. For instance, machine learning can be harnessed to automate routine tasks within software development, thereby optimizing time utilization and conserving human resources [2].

Deep learning methodologies empower the creation of intelligent systems proficient in processing and comprehending complex data [3]. Natural language

processing facilitates more intuitive interactions between systems and users. Moreover, AI Engineering extends its influence on the realm of project management, enabling the analysis of project data, identification of potential risks, and the provision of precise estimates regarding time and cost. These capabilities empower development teams to make informed decisions, ultimately enhancing overall project efficiency.

In this context, our study conducts a comprehensive Roadmap Analysis of AI Engineering Methods. The objective is to dissect and elucidate the methodologies underpinning AI Engineering, focusing on its implementation roadmap. This analysis aims to delve into the intricacies of combining AI with software engineering, shedding light on the current methodologies, their applications, and future directions. The methodology for this analysis involves a systematic examination of existing practices and emerging trends in AI Engineering, based on a thorough review of relevant literature and industry developments. This study aims to offer valuable insights and guide future research and practical applications in AI Engineering, underscoring its growing importance and transformative impact in the technological domain.

## 2. RELATED STUDY

AI Engineering is a discipline that amalgamates artificial intelligence (AI) with software engineering and systems development. In the realm of AI Engineering, apart from the conventional software system engineering cycle, there exists an adjunct AI modeling cycle. The purpose of this additional cycle is to consistently assess the utilized model, ensuring it aligns with the anticipated behavior of the AI system (see Figure 1). Based on the figure that explained about "AI System Engineering Loop," which combines the traditional "Software System Engineering Cycle" with the "AI Modelling Cycle." The traditional cycle includes steps such as coding, building, releasing, operating, and monitoring. The AI modeling cycle adds additional stages like exploring, collecting, modeling, and evaluating, which are specific to AI development. Below these cycles, there's a section on "Application Requirements on AI System Engineering" that lists considerations such as embedded AI, data quality metrics, verification & validation, security & integrity, energy efficiency (green & low energy AI), and the incorporation of explainable AI (XAI), safety, and ethics into the design. This diagram suggests a comprehensive approach to AI system development that integrates traditional software engineering with the specific requirements of AI systems to ensure they are responsible, secure, and efficient.

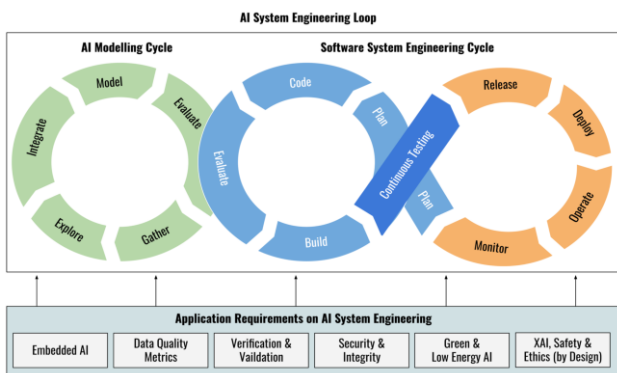


Figure 1. AI system engineering loop [2]

In the realm of AI Engineering, AI techniques play a pivotal role in refining and enhancing various facets of software engineering. These encompass data analysis, software testing, project management, system optimization, and user interface interaction. For instance, machine learning serves to construct predictive models capable of deriving insights from data, thereby facilitating decision-making [4]. Deep learning, conversely, is employed to process intricate data, such as images or sound, and acquire profound patterns [5]. Additionally, natural language processing enables the system to engage in interactions with users using human language.

The domain of AI Engineering spans numerous research areas, including AI's role in software development, pattern recognition, and AI-driven testing. It also covers the automation of development processes, project management enhancements through AI, and optimization techniques. Moreover, the study of ethical and security implications is an integral part of AI Engineering research.

AI Engineering is an important field because it combines artificial intelligence (AI) with software engineering and systems development [6]. There are several reasons why AI Engineering is becoming more and more significant:

(1) Increasing Demand for AI Solutions: With the development of technology, organizations and companies are increasingly realizing the potential and benefits offered by AI. The demand for innovative and efficient AI solutions is increasing, both in software development and in project management.

(2) Quality and Efficiency Improvement Potential: The use of AI in software engineering can help improve the quality and efficiency of the development process. For example, machine learning techniques can be used to analyse data and identify complex patterns, enabling software developers to make better decisions and generate better solutions.

(3) Automation of Routine Tasks: AI Engineering enables the automation of routine tasks in software development, such as testing, debugging, and system maintenance. This can save humans time and effort, allowing them to focus on more complex and creative tasks.

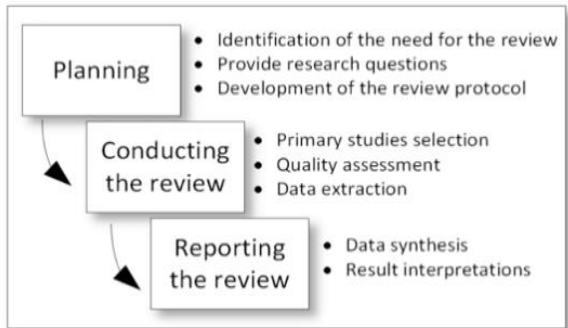
(4) Better Decision Making: AI can provide a deeper understanding of available data and information, enabling better and more accurate decision making in software development. With more careful analysis and more accurate predictions, developers can avoid risks and optimize system performance.

(5) Innovation and Technological Advancement: AI Engineering continues to drive innovation and technological progress in the field of software engineering. The use of new and evolving AI techniques can result in smarter and more effective solutions to address challenges in software development.

## 3. METHOD

The initial phase in investigating a subject matter involves the execution of a Systematic Literature Review (SLR). The concept of a literature study is a constituent of SLR, which finds extensive application in information systems research for the purpose of elucidating the current state of knowledge pertaining to a research inquiry. In the context of this SLR, Kitchenham's method is employed, which has garnered widespread adoption across diverse domains, encompassing software engineering, healthcare, and education [7]. This

methodological approach is characterized by its rigorous and methodically structured framework, facilitating the systematic identification and comprehensive analysis of research studies pertinent to a specific subject. The SLR process is delineated by a well-defined procedure that encompasses several sequential stages, including planning, execution, and the subsequent reporting of the review's findings. Figure 2 depicts an explanation of the journal selection procedure.



**Figure 2.** SLR methodology [7]

(1) Planning stage, the search strategy for finding pertinent studies is devised when the research topic or target is established. As part of the search process, relevant journals and conference proceedings are often manually searched in addition to extensive searches of electronic databases like PubMed or Scopus. Additionally, the inclusion and exclusion criteria are defined, which aid in determining pertinent studies and excluding those that are irrelevant.

(2) Screening stage, based on the inclusion and exclusion criteria, the identified studies are reviewed. Several levels of review are typically performed during the screening process, including title and abstract screening as well as full-text screening. Studies that do not meet the inclusion criteria are excluded from the review, while those that do meet the inclusion criteria are included.

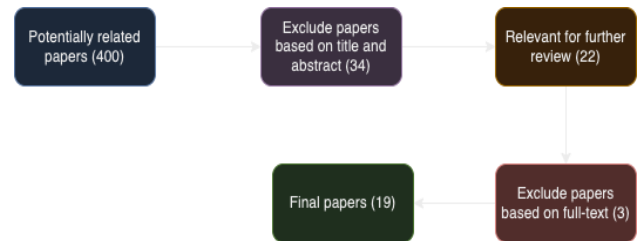
(3) Data extraction stage, the included studies' pertinent data is extracted and synthesised. Typically, this entails gathering

data on the study design, sample size, research methodology, and key conclusions.

(4) The reporting stage, the review's findings are synthesised and presented. This usually entails summarising the important findings from the papers presented, noting any gaps in the literature, and offering recommendations for future study.

Kitchenham SLR is a methodical, evidence-based approach that uses a specified methodology for finding and assessing pertinent studies. In order to provide high-quality research findings, it is crucial that the review is thorough, objective, and replicable, which the methodology helps to achieve. The selection process is based on two criteria: inclusion and exclusion. Table 1 provides an explanation of inclusion and exclusion.

Based on the research objectives, inclusion and exclusion criteria from this literature study were created, and they are used to make sure the findings are reliable in accordance with Kitchenham's standards. English-language articles with full-text papers and a 2018–2023 publication year are available. The structure of the search method is based on the selection of key terms utilizing alternative words and synonyms in each search string (e.g., "AI Engineering"). The search for publications took place between 2018 and 2023. The data was obtained from the databases Scopus, ScienceDirect, ACM Digital Library, and IEEE Xplore, and was then integrated into the Mendeley program. Several irrelevant papers were issued at this point based on the title and abstract. There are 400 papers in the literature with details, and 19 of them are included in this research map which is explained by Figure 3.



**Figure 3.** The selection process of Kitchenham approach

**Table 1.** Inclusion and exclusion criteria

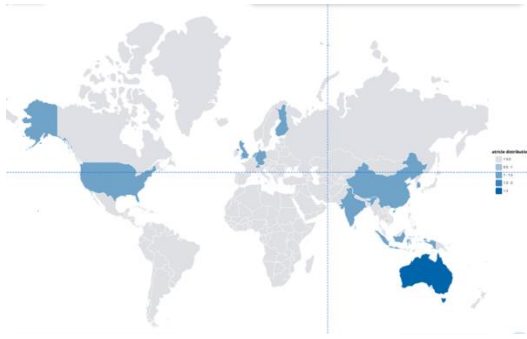
Stages	Inclusion Criteria	Exclusion Criteria
Initiation	According to the search keywords English Year of publication 2018-2023	Language other than English Years of publication beyond 2018-2023
Stage 1 (Title and abstract selection)	There are overlapping discussions between software development and artificial intelligence.	Artificial Intelligence that is not related to software development
Stage 2 (Full-text selection)	Discussion of software development on Artificial Intelligence-based software Discussion of challenges and suggestions in the software development process. Open access paper	Does not discuss the development of the software. Does not address challenges in software development. The use of artificial intelligence in software development Paper whose full-text cannot be accessed

#### 4. RESULT

According to the SLR procedures used in this study, research on the subject of knowledge sharing in AI Engineering was published most frequently from Australia, with 4 (four) papers, followed by the Italy and Austria, with 2 (two), and other countries most frequently with 1 (one) that's

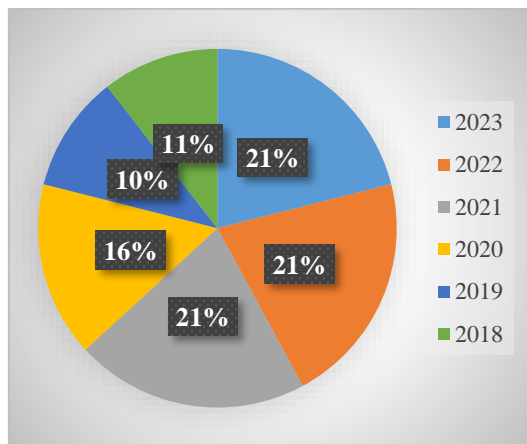
explained by Figure 4. Australia may be leading in AI Engineering publications, one would need to consider several factors including the country's research environment, government and institutional support for AI innovation, investment in technology, and collaborations between academia and industry. These elements contribute to the development of a strong AI community, which in turn drives

research and knowledge sharing.



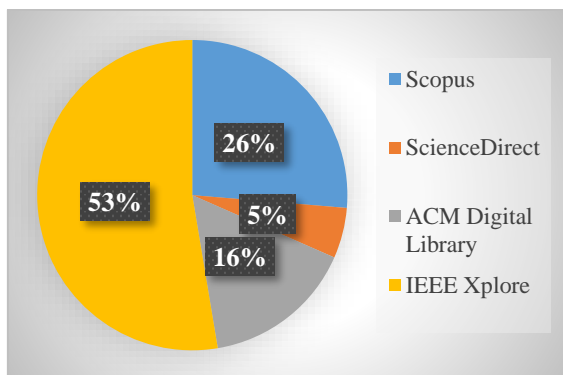
**Figure 4.** Article distribution by country

In terms of publication time, this research is most widely published in 2021 to 2023 with 12 (twelve) articles where each year there are 4 (four) articles, followed in 2020 with 3 (three) articles, in 2019 and 2018 each with 2 (two) articles which we can see on Figure 5. There are some factors that may have influenced publication trends refers to examining reasons that could explain changes in the volume of research papers over time. Factors like policy changes refer to new government or institutional directives that might prioritize or de-emphasize research in AI Engineering.



**Figure 5.** Number of article distribution by year

Based on the composition of research article publishers, the issue of sharing knowledge in AI Engineering is published by IEEE Xplore publishers with a percentage of 53%, followed by Scopus with 26%, ACM Digital Library 16%, and Science Direct with 5%, as illustrated on Figure 6.



**Figure 6.** Number of article distribution by year

Various studies highlighting the challenges and methods of developing Artificial Intelligence (AI)-based software. These studies cover a wide range of aspects, from identifying problems in requirements and engineering AI-intensive systems to analyzing ethical requirements and responsible AI [8]. Several studies address challenges in requirements engineering for AI, including the precise definition of contextual and data requirements, and the planning and design of efficient AI systems. In addition, it is also important to consider the human factor in the development of AI systems to ensure compliance with the desired goals and ethical values [9].

Other challenges faced are integration of machine learning models in existing infrastructure, testing and validation of AI systems throughout, and deep understanding of user requirements in the intended application domain [10]. In the quest to build responsible AI systems, tensions were found in the implementation of ethical requirements, such as a lack of suitable tools and methods to monitor and validate the ethical compliance of AI systems that have been implemented [11].

In addition, challenges were also found in software engineering methodologies that are particularly suitable for developing AI systems, including adapting the SDLC framework with data engineering processes and training models [12]. The researchers also present recommendations and frameworks for addressing these challenges, such as TrustOps which offers a risk-based approach to managing trust in AI systems, or an approach that guides users in articulating ethical requirements for explainable AI [13].

Overall, the outcome of these studies is the importance of the discipline of AI Engineering in addressing the challenges of developing AI-based software [14]. Using AI techniques and integrating traditional software engineering principles with AI-specific approaches can help create systems that are efficient, responsible and compliant with user requirements and the necessary ethical values. Challenges in AI-based software development are presented in the Table 2:

**Table 2.** Challenge of AI engineering

Challenge	Reference
Software Development Life Cycle	[2, 10-12, 15-19]
Requirement Engineering	[8, 9, 14, 20-24]
Responsible AI	[11, 14, 16, 18, 25, 26]
Trustworthiness	[12-14]
Explainability AI	[2, 14, 21]

## 5. DISCUSSION

Software engineering is a software development process consisting of various phases and practices that need to be carried out by each stakeholder with the aim that the software built is effective and cost effective. Various software development methods have been proposed, standardized and adopted in the industry. Starting from the most traditional ones such as waterfall which is suitable for use on software whose requirements are clear at the beginning of development [27], to agile which accommodates changes in software requirements, along with the development and operation of the software [28].

The process of developing software development methods adapted to market needs at that time. In the early days of the development of the software industry, the software developed had the aim of optimizing specific business processes in

certain areas. For example, software embedded in aircraft control systems was intended to help pilots control aircraft more easily, reduce human errors and increase aircraft safety factors. Software like this has the characteristics of a clear requirements at the start, and involves limited and specific domain knowledge in the industry. This type of software is expected to have definite requirements at the start, because with its complexity and mission critical nature, changing requirements in the midst of software development will pose a risk not only in terms of cost, but also in terms of the safety of using the software. At that time, the most suitable and commonly used software development method was the waterfall, in which the development process was carried out step by step starting from requirement elicitation to the deployment process. In this method all defined requirements must be traceable to the design, implementation, testing and operational levels.

Along with the advancement of digital technology, the penetration of the use of software has reached homes, and ordinary individuals. With the increased computing capabilities of mobile devices, it is possible for everyone who owns a mobile device to have direct access to the software and become a direct consumer of the software. Software at that time was not only used in offices at work, but had become a tool to support the daily activities of individuals in society. Such as to communicate, directions, places to buy and sell, order food, make payments, etc. Software users are becoming very diverse, and to win the competition, of course, the software manufacturing company must quickly adapt to people's needs and trends in the market. This led to a significant change in the software development process, business and product teams will always come up with new ideas to win over customers. And this must be balanced with the ability to deliver new software features quickly. Driven by such business requirements, the adoption of agile software development is increasing rapidly.

This software development method allows for changes and additions to the software and delivers these requirements quickly. Agile software development is not easily implemented just like that, but is also supported by technology or other tools such as code versioning and collaboration, testing automation, deployment automation which is currently known as continuous integration/continuous delivery (CI/CD).

From the examples of software development events above, every development of a technology will drive new requirements in software development. This need is accompanied by challenges that need to be solved with a new software development paradigm that is supported by technology or tools that can be used in the software development process.

At this time adding artificial intelligence (AI) capabilities to software has become a trend and is implemented in real terms in the industry. Starting from ad recommendations displayed on web services based on web page browsing history to self-driving cars.

Artificial Intelligence (AI) has the idea of embedding the ability to think, reason and take human action into a machine. Machines are expected not only to be able to perform computations quickly, execute tasks based on predefined instructions and logic, but machines are also expected to be able to learn from input data obtained in their operational processes, and create and derive new rules autonomously.

There are different characteristics between traditional software and software based on Artificial Intelligence. In

traditional software, the logic instructions and rules to be executed by the software have been defined from the start, and there will be no changes during the operational process of the software. Rule changes can only be made by changing the software code. This makes traditional software behaviour more predictive. Based on those inputs, one can easily predict the output given by traditional software.

In AI-based software, there are learning capabilities implemented in the software. This makes the behaviour of the software change in line with the operation of the software. These characteristics raise new issues related to the safety of using software that need to be addressed in AI-based software development, such as trustworthiness, explain ability and responsible AI.

In addition, there are additional engineering processes in AI-based software development that are not found in traditional software development. Such as data engineering, data training, model development, and learning monitoring and evaluation after deployment. These complexities present unique challenges that require careful consideration within the software development process. The roadmap analysis of AI engineering methods is instrumental in navigating this evolving landscape and addressing these challenges effectively.

## 6. CONCLUSIONS

AI Engineering's transformative role in technology development is unmistakable, as this discipline is set to redefine software and systems development practices. This systematic literature review has been pivotal in uncovering the myriad challenges brought forth by AI integration, with a particular focus on the necessity for clear definitions, contextual requirements, and the establishment of rigorous performance metrics. The complexities of implementing ethical principles within AI software development have been spotlighted, revealing gaps such as inconsistent application and the need for robust validation protocols. These findings emphasize the need for an industry-wide consensus on ethical standards and performance benchmarks, which are critical for the development of AI systems that are not only technologically advanced but also ethically grounded.

The human element within AI Engineering has emerged as a fundamental aspect of this study, underscoring the importance of multi-level governance and process-oriented practices. The introduction of responsible-AI-by-design principles marks a significant stride towards embedding ethical considerations into the fabric of AI systems from the outset. The research has provided frameworks that address the dual challenges of maintaining high performance while ensuring ethical compliance, highlighting the crucial role of human oversight and ethical foresight in AI system development. This embodies a forward-thinking approach that aims to harness AI's capabilities responsibly, ensuring that AI systems serve the broader societal good without compromising ethical values.

In drawing conclusions, this research offers a compendium of insights into the multifaceted challenges encountered in AI-based software development. By outlining the interplay of AI within software development and offering strategic avenues, it paves the way for stakeholders to effectively navigate these intricacies. The study's comprehensive set of recommendations and pragmatic approaches is a testament to

the ongoing endeavor to realize AI's full potential responsibly. Future research directions are poised to refine these strategies further, ensuring that as AI continues to evolve, it does so with an unwavering commitment to ethical standards, user-centric design, and an inclusive vision for technology's role in society.

## ACKNOWLEDGMENT

The authors would like to thank the following people for their contributions to this study: Mr. EKB, the Lab members for technical assistance on this article and the Department of Computer Science employees for their invaluable assistance. This research is partially supported by an internal publication grant from the Faculty of Computer Science, University of Indonesia.

## REFERENCES

[1] Giray, G. (2021). A software engineering perspective on engineering machine learning systems: State of the art and challenges. *Journal of Systems and Software*, 180: 111031. <https://doi.org/10.1016/j.jss.2021.111031>

[2] Fischer, L., Ehrlinger, L., Geist, V., Ramler, R., Sobieszky, F., Zellinger, W., Brunner, D., Kumar, M., Moser, B. (2020). AI system engineering—Key challenges and lessons learned. *Machine Learning and Knowledge Extraction*, 3(1): 56-83. <https://doi.org/10.3390/make3010004>

[3] Batarseh, F.A., Mohod, R., Kumar, A., Bui, J. (2020). The application of artificial intelligence in software engineering: A review challenging conventional wisdom. *Data Democracy*, 2020: 179-232. <https://doi.org/10.1016/B978-0-12-818366-3.00010-1>

[4] Kotti, Z., Galanopoulou, R., Spinellis, D. (2023). Machine learning for software engineering: A tertiary study. *ACM Computing Surveys*, 55(12): 1-39. <https://doi.org/10.1145/3572905>

[5] Wen, J., Li, S., Lin, Z., Hu, Y., Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54(1): 41-59. <https://doi.org/10.1016/j.infsof.2011.09.002>

[6] Horneman, A., Mellinger, A., Ozkaya, I. (2019). AI engineering: 11 foundational practices. [https://resources.sei.cmu.edu/asset\\_files/WhitePaper/2019\\_019\\_001\\_634648.pdf](https://resources.sei.cmu.edu/asset_files/WhitePaper/2019_019_001_634648.pdf).

[7] Kitchenham, B. (2007). Guidelines for performing systematic literature reviews in software engineering. <https://www.cs.auckland.ac.nz/~norsaremah/2007%20Guidelines%20for%20performing%20SLR%20in%20SE%20v2.3.pdf>.

[8] Lavazza, L., Morasca, S. (2021). Understanding and modeling ai-intensive system development. In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN), pp. 55-61. <https://doi.org/10.1109/WAIN52551.2021.00015>

[9] Heyn, H.M., Knauss, E., Muhammad, A.P., Eriksson, O., Linder, J., Subbiah, P., Pradhan, S.K. Tungal, S. (2021). Requirement engineering challenges for AI-intense systems development. In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN),

pp. 89-96. <https://doi.org/10.1109/WAIN52551.2021.00020>

[10] Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T. (2019). Software engineering for machine learning: A case study. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, pp. 291-300. <https://doi.org/10.1109/ICSE-SEIP.2019.00042>

[11] Lu, Q., Zhu, L., Xu, X., Whittle, J., Douglas, D., Sanderson, C. (2022). Software engineering for responsible AI: An empirical study and operationalised patterns. In Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, pp. 241-242. <https://doi.org/10.1109/ICSE-SEIP55303.2022.9793864>

[12] Cepeda Zapata, K.A., Ward, T., Loughran, R., McCaffery, F. (2022). Challenges associated with the adoption of artificial intelligence in medical device software. In Irish Conference on Artificial Intelligence and Cognitive Science, pp. 163-174. [https://doi.org/10.1007/978-3-031-26438-2\\_13](https://doi.org/10.1007/978-3-031-26438-2_13)

[13] Kwak, J.H. (2022). TrustOps: A risk-based AI engineering process. In 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), pp. 2252-2254. <https://doi.org/10.1109/ICTC55196.2022.9952989>

[14] Ahmad, K., Abdelrazek, M., Arora, C., Bano, M., Grundy, J. (2023). Requirements engineering for artificial intelligence systems: A systematic mapping study. *Information and Software Technology*, 158: <https://doi.org/107176>. 10.1016/j.infsof.2023.107176

[15] Syahputri, I.W., Ferdiana, R., Kusumawardani, S.S. (2020). Does system based on artificial intelligence need software engineering method? Systematic review. In 2020 Fifth International Conference on Informatics and Computing (ICIC), Gorontalo, Indonesia, pp. 1-6. <https://doi.org/10.1109/ICIC50835.2020.9288582>

[16] Islam, Z.U. (2021). Software engineering methods for responsible artificial intelligence. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1814-1815.

[17] Serban, A., van der Blom, K., Hoos, H., Visser, J. (2020). Adoption and effects of software engineering best practices in machine learning. In Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 1-12. <https://doi.org/10.1145/3382494.3410681>

[18] Barletta, V.S., Caivano, D., Gigante, D., Ragone, A. (2023). A rapid review of responsible AI frameworks: How to guide the development of ethical AI. In Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, pp. 358-367. <https://doi.org/10.1145/3593434.3593478>

[19] Prasad, J., Jain, A., Zachariah, U.E. (2022). Comparative evaluation of machine learning development lifecycle tools. In 2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC), pp. 1-6. <https://doi.org/10.1109/ICMACC54824.2022.10093671>

[20] Belani, H., Vukovic, M., Car, Ž. (2019). Requirements engineering challenges in building AI-based complex systems. In 2019 IEEE 27th International Requirements

- Engineering Conference Workshops (REW), pp. 252-255. <https://doi.org/10.1109/REW.2019.00051>
- [21] Habiba, U.E., Bogner, J., Wagner, S. (2022). Can requirements engineering support explainable artificial intelligence? Towards a user-centric approach for explainability requirements. In 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW), pp. 162-165. <https://doi.org/10.1109/REW56159.2022.00038>
- [22] Yoshioka, N., Husen, J.H., Tun, H.T., Chen, Z., Washizaki, H., Fukazawa, Y. (2021). Landscape of requirements engineering for machine learning-based ai systems. In 2021 28th Asia-Pacific Software Engineering Conference Workshops (APSEC Workshops), pp. 5-8. <https://doi.org/10.1109/APSECW53869.2021.00011>
- [23] Ahmad, K., Abdelrazek, M., Arora, C., Bano, M., Grundy, J. (2023). Requirements practices and gaps when engineering human-centered Artificial Intelligence systems. *Applied Soft Computing*, 143: 110421. <https://doi.org/10.1016/j.asoc.2023.110421>
- [24] Agbese, M., Mohanani, R., Khan, A.A., Abrahamsson, P. (2023). Ethical requirements stack: A framework for implementing ethical requirements of AI in software engineering practices. In Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, pp. 326-328. <https://doi.org/10.1145/3593434.3593489>
- [25] Lu, Q., Zhu, L., Xu, X., Whittle, J., Xing, Z. (2022). Towards a roadmap on software engineering for responsible AI. In Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, pp. 101-112. <https://doi.org/10.1145/3522664.3528607>
- [26] Agbese, M., Mohanani, R., Khan, A., Abrahamsson, P. (2023). Implementing AI ethics: Making sense of the ethical requirements. In Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, pp. 62-71. <https://doi.org/10.1145/3593434.3593453>
- [27] Munteanu, V.P., Dragos, P. (2021). The case for agile methodologies against traditional ones in financial software projects. *European Journal of Business and Management Research*, 6(1): 134-141. <https://doi.org/10.24018/ejbmr.2021.6.1.741>
- [28] Dudhat, A., Abbasi, M.A. (2021). Discussion of agile software development methodology and its relevance to software engineering. *ADI Journal on Recent Innovation*, 3(1): 105-114. <https://doi.org/10.34306/ajri.v3i1.536>