Requirements Elicitation for Prototype-driven AI Engineering: a Case Study in Police Report Generation*

Martijn van Vliet[1], Wouter Westerkamp[1], Sjaak Brinkkemper[1] and Sergio España[1]

[1]Department of Information and Computing Sciences, Utrecht University, Heidelberglaan 8, 3584 CS, Utrecht, The Netherlands

Abstract

[Context and Motivation] Systems containing AI components introduce novel challenges that increase the complexity of systems and their corresponding development processes. **[Question / Problem]** Traditional requirements engineering practices are unable to produce input for the development of AI-based systems in a way that covers all necessary aspects to develop responsible, compliant and safe solutions. **[Principal ideas / Results]** We developed PRE4AIM, a requirements elicitation method that enables the systematic inclusion of the required multidisciplinary perspectives needed for the development of AI-based systems, and applied this in a technical action research at the Netherlands Police. We were able to identify 23 requirements and were able to further enrich them by distinguishing obstacles, opportunities or possible solutions to satisfy them. **[Contribution]** Our method provides a novel approach for the elicitation of requirements for AI-based systems, that allows for improved domain understanding, risk identification, project planning and created meaningful input for future development of the current prototype.

Keywords

Prototype-driven Development, Responsible AI Engineering, Requirements Engineering, Police Reporting

1. Introduction

The disruptiveness of AI is revolutionizing how organizations operate and changing interactions and relationships among stakeholders [1]. The rise of AI has spread to the public sector, with The Netherlands Police being one example of an organization that strives to become more effective and efficient. Illustrated by the case of Police2Report, a prototype that uses generative AI to generate police reports from audio transcripts to help alleviate the current large administrative burden on its employees. However, the context of law enforcement is a very sensitive area to be deploying this technology, with possible high impact legal ramifications resulting from irresponsible or unsafe use. Developing AI-based systems properly for this specific context therefore requires a development approach that facilitates the construction of compliant, responsible and safe solutions.

Traditional software engineering (SE) and requirements engineering (RE) practices are ineffective in providing the necessary guidance towards the development of responsible AI-based systems [2]. RE has always played a crucial role within the software development life cycle (SDLC), contributing towards the development of successful software systems [3]. Therefore, scholars call for the adjustment of RE techniques to the new paradigm of AI-based systems and their accompanying additional challenges [2].

In this research we study the common pitfalls of AI projects and the challenges and shortcomings of existing SE and RE techniques. Our main goal is expressed in our main research question: **(MRQ1)** *How to elicit requirements for AI-based system to foster responsible development efforts?* We describe our research method in Section 2. Using our findings from the literature study, we defined design principles **(DP)** and used them to construct PRE4AIM, a method for requirements elicitation in the context of AI-based systems (Section 3). We applied our method in the real-world setting of Police2Report and

In: A. Hess, A. Susi, E. C. Groen, M. Ruiz, M. Abbas, F. B. Aydemir, M. Daneva, R. Guizzardi, J. Gulden, A. Herrmann, J. Horkoff, S. Kopczyńska, P. Mennig, M. Oriol Hilari, E. Paja, A. Perini, A. Rachmann, K. Schneider, L. Semini, P. Spoletini, A. Vogelsang. Joint Proceedings of REFSQ-2025 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2025. Barcelona, Spain, April 7, 2025.

m.vanvliet@uu.nl (M. v. Vliet); w.westerkamp@uu.nl (W. Westerkamp); s.brinkkemper@uu.nl (S. Brinkkemper); s.espana@uu.nl (S. España)

demonstrate our findings that resulted from one iteration of our method (Section 4). We use these results to assess the effectiveness of our method and highlight points for improvement (Section 5).

2. Research Method

To design and evaluate our requirements elicitation method, we performed technical action research in the context of the Police2Report project. Figure 1 represents the steps of our research in accordance with the phases of the main design science cycle [4].

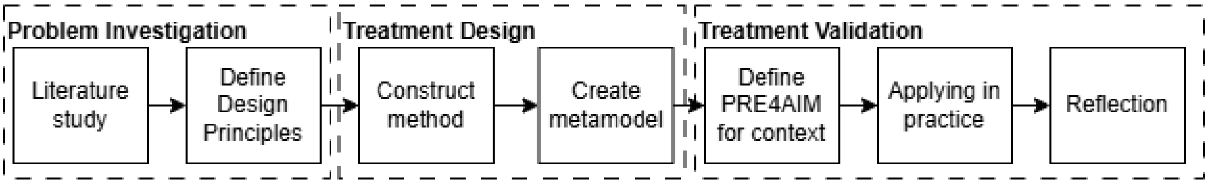

Figure 1: Step by step description of our research method.

Problem Investigation: We performed a narrative literature review to identify three research areas that are relevant to our research question: (i) AI project pitfalls and challenges (ii) Software engineering for AI-based systems and AI engineering and (iii) Requirements engineering for AI-based systems. From the created knowledge base, we derived design principles (**DP**), that each represent a goal that we want to achieve, a problem that we want to solve or a negative influence that we want to mitigate.

Treatment Design: We have constructed our method by applying method engineering techniques and specified it using Process Deliverable Diagrams to create a metamodel that interrelates method activities and its products [5]. The identified design principles have informed the design of our method. We have framed our method as a fragment that can be reused in other situations and as part of any SDLC by using the SWEBOK knowledge areas as the framework [6].

Treatment Validation: We used the context of the Police2Report prototype project as a case to apply and evaluate our method in practice. We wanted to test the strong and weak points of our method and demonstrate its practicability in a real-world setting and to gauge its usefulness towards guiding the prototype towards a useful fitting product.

3. Constructing the Method

3.1. Design Principles Derived from Related Literature

A new wave of AI has hit the software industry with the proliferation of AI-based systems, which are software systems that include AI components and that can be implemented in more complex environments, or implemented tightly into existing processes and systems **(DP1)** [7]. While the ambitions are there, organizations experience a very high failure rate for AI projects, indicating that a large amount of prototypes never progress to production and do not realize their desired impact [8]. Reasons are attributed to inadequately dealing with novel multidisciplinary challenges during development, such as ethical, legal, social, economical, technical, data oriented and/or organizational concerns **(DP2)** [9]. Tackling these challenges requires changes to the way that projects are run, as they will otherwise lead to unrealistic expectations, use case related issues, organizational constraints, lack of key resources and technological issues **(DP3)** [10]. The unrealistic expectations are cause for scope creep, until the lack of focus makes it impossible to create viable solutions **(DP4)** [10]. These challenges further complicate the implementation with existing processes and systems due to technological immaturity, lack of available knowledge, regulations, bureaucratic hurdles or shortcomings from the existing landscape **(DP5)** [10, 11]. Additionally, wrong use cases lead to misalignment between the objectives of the project and the strategic goals of organizations, which in turn leads to project failure due to a lack of necessary support and resources **(DP6)** [12]. Consequently, pilot paralysis is common for AI projects which prevents organizations to scale up their solutions **(DP7)** [13].

All aforementioned challenges demonstrate that building, operating, and maintaining AI-based systems is different from traditional software systems [7]. The non-deterministic nature of the technology leads to systems that have simple components, but complex overall behavior due to their dependencies, competitions, relationships, or other types of interactions between the components or between a given system and its environment [14]. As such, the individual parts do not automatically convey a perfect understanding of the whole system's behavior [15]. These characteristics demand different quality attributes of the system to function as intended, which influences the design of the system **(DP8)** [16]. Said quality attributes require interdisciplinary collaborative teams that are often missing in practice **(DP10)** [17]. Calls exist to revisit the ways of software development to incorporate these additional components and to consider all necessary novel concerns, as for instance current methods have a tendency to largely neglect AI ethics principles **(DP9,11)** [15, 18].

The new paradigm of AI-based systems has changed existing requirements and resulted in the appearance of new ones [2]. Concerns about responsible, legal and safe use have extended the meaning of system transparency, reliability, security, explainability and others **(DP8)**. Eliciting and specifying these additional requirements introduces new challenges that are further complicated by frequently changing requirements for large-scale complex systems **(DP11)** [19]. There is a need to adapt and extend traditional RE approaches to ensure that the requirements captured are as accurate and complete as possible, while recognizing the special characteristics of AI systems [15]. The development of AI-based systems requires purposeful requirements engineering that also considers ethics and fairness [20]. Without it, responsible AI requirements will either be omitted, stay high-level or allow practitioners to turn a blind eye **(DP12)** [18, 21]. Understanding user needs, setting clear objectives and establishing domain understanding is crucial for AI-based systems to be able to provide meaningful value and to seamlessly integrate into existing processes (**DP13**) [13]. Requirements engineering activities should stimulate the participation of relevant stakeholders and help with the coordination between all available various opinions and knowledge to establish a clear path forward [22]. Iteratively performing these activities in an agile manner can help identify important factors early and therefore lead to a higher success rate of the project [10]. Actively involving a prototype is one activity that can contribute towards emergent design **(DP11)** [23]. All derived design principles are summarized in Table 1.

Nr.	Design Principles	Reference(s)
DP1	Suitability for the new paradigm of AI-based systems and their non-deterministic nature	[7, 24, 2, 15]
DP2	Ability to take into account known real-world challenges and concerns	[9]
DP3	Ability to facilitate better project planning and prevent project pitfalls	[10]
DP4	Aid towards setting realistic expectations and defining measurable goals	[10, 19]
DP5	Ability to identify organizational constraints	[11, 10, 25]
DP6	Ability to assist towards better alignment between the project and strategic business goals	[26, 27, 12]
DP7	Ability to address pilot paralysis concerns	[13]
DP8	Ability to identify AI specific quality attributes	[16, 2, 21]
DP9	Usable in the context of the AI engineering SDLC	[7, 15, 19, 6]
DP10	Ability to facilitate the inclusion of multidisciplinary perspectives related to AI	[13, 28, 22]
DP11	Ability to fit with an agile way of working and the dynamic nature of AI projects	[10, 19]
DP12	Ability to foster responsible, safe and compliant design	[18, 20, 15]
DP13	Ability to help create better domain understanding	[29]

Table 1
Summary of derived design principles for the method and their sources.

3.2. Metamodel and Design Rationale for the Creation of the Method

As shown in Figure 2, the resulting method has separated all activities in two distinct parts, preparation and elicitation. The first part revolves around identifying an AI project and its relevant perspectives and representative stakeholders for them. With the representatives identified, the interview protocol can be constructed after which the interviews can be planned. The second part consists of the elicitation process, by performing the interviews according to the interview protocol. Depending on the phase, the interviews can yield either requirements, obstacles or opportunities and/or potential solutions.

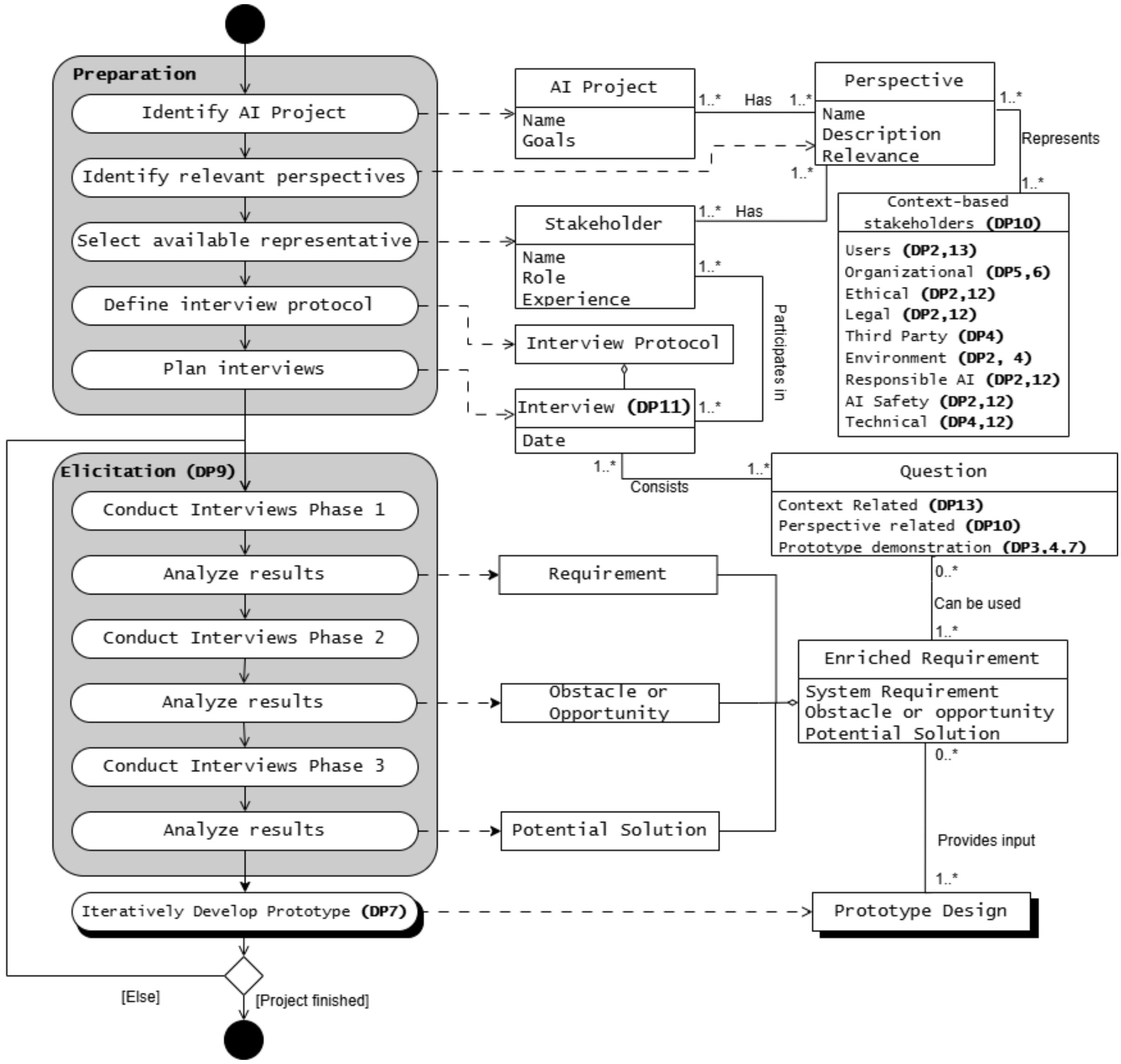

Figure 2: Process-Deliverable Diagram of the PRE4AIM method.

We used the RE activities as described by SWEBOK as the backbone of the method, to make it suitable as a part of a SDLC **(DP9)** [6]. The interaction between the RE activities, the stakeholder selection and the prototype development are cyclical to accommodate for iterative, dynamic and agile development methods **(DP1,11)**. The (development of) the prototype is closely intertwined with the RE activities and the elicitation interviews, with the goal to identify project pitfalls as soon as possible and to remedy causes for pilot paralysis **(DP3,7)**. Furthermore, the prototype is demonstrated to the stakeholders for each iteration of the method to aid towards setting realistic expectations **(DP4)**. The diverse set of context-based stakeholder perspectives are selected to help create a broad understanding of the domain **(DP13)** and to make the total desired output as multidisciplinary as possible in a structured manner **(DP10)**. Representatives for the organizational perspectives are included to identify organization constraints **(DP5)** and to align with business goals and strategy **(DP6)**. The perspectives of the user, legal and ethical experts and possible involved third parties are involved to identify real-world concerns and to help set realistic expectations **(DP2,4)**. Furthermore, the ethical, legal, AI safety, responsible AI perspectives combined will provide input that might foster more responsible design of the prototype **(DP12)**. All gathered input can be discussed from a technical perspective to assess their feasibility and to address concerns regarding the responsible and safe use in order to search for solutions **(DP4,12)**. Lastly, we went with semi-structured interviews for our data collection, as interviews are thoroughly embedded in agile elicitation methods and practitioners are most familiar with them **(DP11)** [30, 31]. For each interview, we used the same structure for the protocol, consisting of three parts: **(i)** A series of background questions to get to know the stakeholder and their environment **(DP13)** and to assess their knowledge and experience, **(ii)** a series of questions specifically catered towards each perspective to catch their ideas and concerns towards the project or gained input from earlier interviews **(DP10)** and **(iii)** a full demonstration of the current iteration of the prototype to gauge their reactions **(DP3,4,7)**.

4. Treatment Validation at the Netherlands Police

4.1. PRE4AIM in Applied in Practice

The specific method that we applied for the context of the development of the Police2Report prototype is shown in Figure 3. A detailed overview of all the output is provided in the technical report [32].

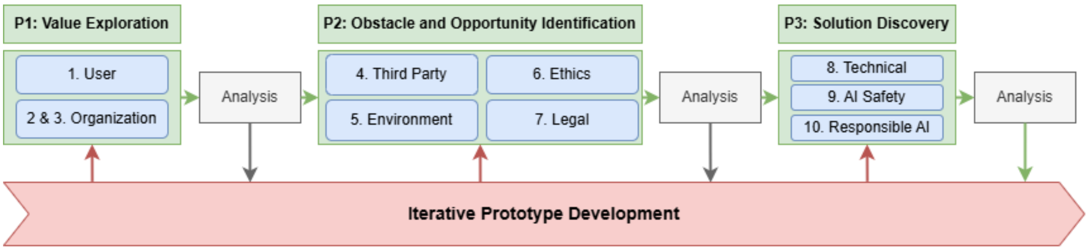

Figure 3: Flow of the PRE4AIM method in the context of Police2Report.

Phase 1 - Value exploration: - Main purpose is to understand the most important needs and concerns of stakeholders responsible for the realization of business value. We identified a strong wish towards the reduction of the required administrative burden during the creation of police reports. Participants were very receptive towards the the prototype but also expressed concerns regarding reliability, trust in the results and showed a need to be in control of the output.
Phase 2 - Obstacle & opportunity identification: - We discussed the output from Phase 1 with additional stakeholders to evaluate their feasibility and to search for additional opportunities or potential obstacles. We found that challenges would likely mainly revolve around the legal validity of the generated reports for the following steps in the legal chain. Furthermore we learned that just generating summaries of the transcripts would not be sufficient, that a certain level of quality guarantees have to be provided, direct data flows with third-parties were not allowed and that traceability between raw input and generated report is essential. Lastly, carefully guided human judgment and control through a user-friendly interface was regarded as an additional crucial element.
Phase 3 - Solution discovery: Here we explored potential solutions for any identified obstacles or specific stakeholder needs. We found the importance of the role of the user for the purpose of quality control and placed the role of the system towards an advisory role only. Additionally, we identified the need for visible and hidden quality checks that test whether the system is working correctly or the user is interacting with the system as intended.

4.2. Elicited Requirements

We validated our method by performing one iteration of the method. Using the first version of the Police2Report prototype, we conducted 9 semi-structured interviews following the specified interview protocol. Two authors were present at all but one interview. Our participants consisted of an interrogator, an implementation manager, a domain architect, a co-founder of a third-party, a solution architect, an ethics advisor, a legal advisor, a data scientist, an AI safety expert and a responsible AI expert.

The two participants representing the ethical and legal perspective were present in the same interview. All interviews were recorded and the analysis took place separately between the two interviewers. The results were later cross-checked and verified by the two original interviewers.

We managed to elicit 23 initial requirements from the first phase interviews. Of these 23 requirements, 9 could be classified as functional requirements, 4 were performance related, 8 referred to specific quality aspects and 2 indicated constraints. A total of 18 of these 23 requirements were used as direct input for phase 2 and 3, which allowed us to enrich them with additional clarifications. This in turn allowed us to better understand their importance and provided potential directions for solutions. In Table 2 we provide three examples of requirements that were discussed in all three phases.

Phase	Example 1:	Example 2:	Example 3:
Phase 1	**(ID1)** - The system should be able to include timelines, and links/references to the original multimedia files in the report	**ID15** - The system should be accurate in that the generated output is contextually correct and meet user expectations.	**(ID17)** - The system should be able to allow the user to have control over the output of the system.
Phase 2	**(OP5)** - Developing additional functionalities such as traceability, error correction, and speaker recognition.	**OB2** - Capturing subjective aspects that requires specific domain knowledge or vision. **OP7** - Applying different models to perform better in various contexts.	**(OB3)** - Adhering to AI regulations related to intellectual property rights, privacy violation, and potential biases. **(OB7)** - Mitigating the negative effects on people's performance when assisted by software (deskilling, laziness). **(OP4)** - Guiding users towards responsible use through a user-friendly interface.
Phase 3	**(S4)** - Usage of knowledge graphs to enable visible traceability to sources.	**(S5)** - Usage of benchmark tests to (continuously) assess the factual correctness of the output.	**(S6)** - Correcting imperfections by actively engaging the human-in-the-loop during quality control.

Table 2
Three examples of requirements throughout the method.

The three examples show how the output from the different stakeholders are able to build upon each other's statements. Example 1 shows the development from an expression of a stakeholder need to a potential design input for a next iteration of the prototype. Example 2 and 3 highlight the importance of the quality of the desired output, which has to be monitored actively. Furthermore, they highlight that the user will bear lot of responsibility towards the final results due to legal reasons, but that there are also opportunities when this is designed and implemented correctly.

Not all output from each phase was discussed in other phases, but still prove to be useful even without enriching them with further insights. Potential reasons that these examples were not capitalized on yet could be that other requirements were prioritized over them or that there was not enough knowledge available between the interviewers and the stakeholders to address them properly. Having them identified however, makes them clear candidates to focus on during a next iteration of the method.

5. Discussion and Conclusion

We present PRE4AIM, our method to elicit requirements in the context of AI-based systems, therefore achieving our main goal captured by our **MRQ1**. The list of design principles highlight the relevance of our research, as they represent the tangible challenges related to the development of AI-based systems. By structuring the elicitation process in three phases, we allowed stakeholders to build upon each others input in a structured manner to enrich their statements from multidisciplinary perspectives, answering the request of Delgado et al. (2021) [22]. As such, our output helps to prevent project pitfalls, illustrated by our identified two main barriers of (1) unknown validity of the report for the successive legal chain and (2) the forbidden data-flows to third parties that is present in the current version of the prototype. Furthermore, by providing a generalized version of our method we answer the calls of Ahmad et al. (2023) and Martinez et al. (2022) for novel ways of requirements elicitation in a way that provides actionable guidance to practitioners and that fit the complex and uncertainty-prone new paradigm of AI-based systems [2, 7]. By sharing our results we provide a unique insight into the start of a real-world example of an AI-based system development project in a sensitive and complex environment.

As the role of complex AI-based systems will continue to grow in the future, so will the importance of requirements engineering techniques that suit the needs of these systems. Our method provides plenty of opportunity to extend upon, either by adding phases, perspectives or altering the interview protocol. The method can be applied in different projects or for more iterations over a longer period of time. Lastly, the implications of the elicited information on RE activities such as analysis, specification or validation can be explored. However, our research is only a single case which makes it harder to generalize our results. The usefulness of our findings could be attributed to the extensive knowledge of our participants, which may not always be available in other contexts or projects. We also only performed a single iteration of the method, so it is currently unsure if the yield would be similar in usefulness after performing additional rounds.

6. Declaration on Generative AI

The author(s) have not employed any Generative AI tools for the writing process of this paper.

References

[1] J. P. Bharadiya, R. K. Thomas, F. Ahmed, Rise of artificial intelligence in business and industry, Journal of Engineering Research and Reports 25 (2023) 85–103.

[2] K. Ahmad, M. Abdelrazek, C. Arora, M. Bano, J. Grundy, Requirements engineering for artificial intelligence systems: A systematic mapping study, Information and Software Technology (2023) 107176.

[3] H. F. Hofmann, F. Lehner, Requirements engineering as a success factor in software projects, IEEE software 18 (2001) 58.

[4] R. J. Wieringa, Design science methodology for information systems and software engineering, Springer, 2014.

[5] I. van de Weerd, S. Brinkkemper, Meta-modeling for situational analysis and design methods, in: Handbook of research on modern systems analysis and design technologies and applications, IGI Global, 2009, pp. 35–54.

[6] H. Washizaki, Guide to the software engineering body of knowledge V4.0: Swebok, IEEE Computer SoLos Alamitos, CA?ciety, 2024.

[7] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, S. Wagner, Software engineering for ai-based systems: a survey, ACM Transactions on Software Engineering and Methodology (TOSEM) 31 (2022) 1–59.

[8] J. Weiner, Why AI/data science projects fail: how to avoid project pitfalls, Springer Nature, 2022.

[9] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, et al., Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy, International journal of information management 57 (2021) 101994.

[10] J. Westenberger, K. Schuler, D. Schlegel, Failure of ai projects: understanding the critical factors, Procedia computer science 196 (2022) 69–76.

[11] J. K.-U. Brock, F. Von Wangenheim, Demystifying ai: What digital transformation leaders can teach you about realistic artificial intelligence, California management review 61 (2019) 110–134.

[12] R. Vayyavur, Why ai projects fail: The importance of strategic alignment and systematic prioritization (2024).

[13] R. G. Cooper, Why ai projects fail: Lessons from new product development, IEEE Engineering Management Review (2024).

[14] H. Belani, M. Vukovic, Ž. Car, Requirements engineering challenges in building ai-based complex systems, in: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), IEEE, 2019, pp. 252–255.

[15] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Zowghi, A. Jacquet, Responsible ai pattern catalogue: A collection of best practices for ai governance and engineering, ACM Computing Surveys 56 (2024) 1–35.

[16] I. Ozkaya, What is really different in engineering ai-enabled systems?, IEEE software 37 (2020) 3–6.

[17] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: A case study, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, 2019, pp. 291–300.

[18] J. Bosch, H. H. Olsson, I. Crnkovic, Engineering ai systems: A research agenda, Artificial intelligence paradigms for smart cyber-physical systems (2021) 1–19.

[19] F. Kumeno, Software engineering challenges for machine learning applications: A literature review, Intelligent Decision Technologies 13 (2019) 463–476.

[20] C. Kästner, E. Kang, Teaching software engineering for ai-enabled systems, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training, 2020, pp. 45–48.

[21] Q. Lu, L. Zhu, X. Xu, J. Whittle, Z. Xing, Towards a roadmap on software engineering for responsible ai, in: Proceedings of the 1st International Conference on AI Engineering: software Engineering for AI, 2022, pp. 101–112.

[22] F. Delgado, S. Yang, M. Madaio, Q. Yang, Stakeholder participation in ai: Beyond" add diverse stakeholders and stir", arXiv preprint arXiv:2111.01122 (2021).

[23] C. Kriesi, J. Blindheim, Ø. Bjelland, M. Steinert, Creating dynamic requirements through iteratively prototyping critical functionalities, Procedia CIRP 50 (2016) 790–795.

[24] L. Baier, F. Jöhren, S. Seebacher, Challenges in the deployment and operation of machine learning in practice., in: ECIS, volume 1, 2019.

[25] P. F. Katina, C. B. Keating, R. M. Jaradat, System requirements engineering in complex situations, Requirements engineering 19 (2014) 45–62.

[26] A. F. Cooper, J. Frankle, C. De Sa, Non-determinism and the lawlessness of machine learning code, in: Proceedings of the 2022 Symposium on Computer Science and Law, 2022, pp. 1–8.

[27] M. Becker, M. Drew, Overcoming the challenges in deploying user provisioning/identity access management backbone, BT technology journal 23 (2005) 71–79.

[28] D. Pandey, U. Suman, A. K. Ramani, An effective requirement engineering process model for software development and requirements management, in: 2010 International Conference on Advances in Recent Technologies in Communication and Computing, IEEE, 2010, pp. 287–291.

[29] P. Cummings, N. Schurr, A. Naber, Charlie, D. Serfaty, Recognizing artificial intelligence: The key to unlocking human ai teams, Systems Engineering and Artificial Intelligence (2021) 23–45.

[30] A. F. de Sousa Silva, G. R. S. Silva, E. D. Canedo, Requirements elicitation techniques and tools in the context of artificial intelligence, in: Brazilian Conference on Intelligent Systems, Springer, 2022, pp. 15–29.

[31] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Computers in human behavior 51 (2015) 915–929.

[32] M. van Vliet, W. Westerkamp, S. Brinkkemper, S. España, Requirements elicitation for prototype-driven ai engineering: Technical report, [Online]. Available: https://doi.org/10.17632/6yy6nkb9wr.1 (2024).