Template-Driven Specification of Requirements for LLM-Based Chatbots

Caio V. Melo da Silva[1], Ana R. Cárdenas Maita[1], Gregorio Assagra de Almeida Filho[1], Carlos H. Camillo da Silva [1], Gabriel Dimant[1], Renato A. Almeida[2], Guilherme M. Lopes Costa[2], Anna M. Cintra Araujo[2], Barbara Correia Dos Santos Santana[2], Enio Alterman Blay[1], Sarajane Marques Peres[1]

[1]Universidade de São Paulo (USP)
São Paulo – SP – Brazil

[2]Instituto de Ciência e Tecnologia Itaú
São Paulo – SP – Brazil

{ar.cardenasmaita, ccamillo.silva, gregassagraf, caiovmsilva, gabriel.dimant, enio.alterman.blay, sarajane}@usp.br

{renato-augusto.almeida, guilherme.lopes-costa,anna.cintra-araujo, barbara-correia-santos.santana}@itau-unibanco.com.br

Abstract. *The integration of large language models (LLMs) into chatbots introduces specific requirements related to reasoning, regulatory constraints, and user-sensitive interactions. This paper presents a template-driven approach for specifying system requirements in LLM-enhanced chatbots. The method uses a structured requirements card that defines the functional and non-functional roles, input-output behaviors, contextual reasoning dependencies, and compliance or operational constraints. By formalizing these requirements, the approach supports alignment between LLM capabilities and system design, facilitates verification of behavior across dialogue flows, and improves traceability within the software architecture. This specification model is particularly suited for domains where consistency, interpretability, and compliance with business rules are critical. As a proof of concept, a set of requirements for a financial chatbot was elicited and organized into cards derived from the template.*

1. Introduction

Chatbots are now key components for supporting organizational processes and user interactions across domains. Advances in Generative Artificial Intelligence have led to integrating large language models (LLMs), enabling advanced conversational competence and natural language understanding for chatbots [Dam et al. 2024]. As LLMs are increasingly embedded in information systems, they emerge as influential components whose behavior impacts system reliability, user experience, and compliance [Sarker 2024].

In this context, software engineering plays a critical role in ensuring that systems incorporating LLMs are designed with rigor and precision. Traditional practices in requirements engineering, system modeling, and verification remain essential foundations [Amershi et al. 2019]. However, the unique characteristics of LLMs, such as probabilistic reasoning, emergent behaviors, and dependency on dynamic prompts

[Bommasani et al. 2021], demand that software engineering processes explicitly account for their functional and non-functional roles within the system. Rather than replacing established practices, this integration calls for a mindset adjustment that extends familiar methodologies to encompass LLM-specific considerations.

LLMs can be seen as models that map user utterances to contextually relevant responses [Zhao et al. 2023]. When treated as system components, they must be governed by clearly defined requirements to support predictable, safe, and interpretable behavior [Martínez-Fernández et al. 2022]. This need becomes critical in socio-technical systems operating in regulated or high-risk domains, where consistency and accountability are paramount [Cabrera et al. 2024]. Thus, it is essential to formalize how LLMs act and interact in these systems, ensuring they align with organizational goals and constraints.

This paper introduces a template-driven method for specifying requirements in LLM-based chatbots. The method generates structured requirement cards, inspired by [Mitchell et al. 2019], to capture both functional and non-functional roles while clearly separating the LLM's core activities (reading, writing, conversing) from the broader system requirements. This separation allows LLM-specific operations to be addressed more precisely during specification. The template also includes a section for the semantic detailing of requirements, reflecting the communicative goals intrinsic to chatbot interactions. Together, these elements strengthen the alignment between LLM capabilities and system design, and have the potential to improve behavioral verification across dialogue flows as well as traceability within the software architecture.

Thus, in this paper we present two main contributions: (i) a specification template for structuring requirements in LLM-based systems, and (ii) a comprehensive set of 46 functional and non-functional requirements elicited and organized into cards as a proof of concept. The requirements elicitation occurred within a financial domain project, however most of the resulting requirements are generic enough to be applicable across different business domains. This makes the approach particularly relevant for areas where consistency, interpretability, and compliance with business rules are critical. This paper is organized as follows: Section 2 reviews key concepts and related approaches regarding the integration of LLMs into socio-technical systems and the development of our template. Section 3 presents the methodological foundations and the specification template. Section 4 summarizes the 46 functional and non-functional requirements derived using our method and discusses their implications. Section 5 concludes the paper.

2. Theoretical Background

2.1. Chatbots and socio-technical information systems

According to [Jurafsky and Martin 2025], "language is the mark of humanity and sentience; conversation or dialogue is the most fundamental arena of language". The same author outlines categories within conversational systems: task-oriented dialogue systems are built around structured interaction flows aimed at executing predefined functions; open-domain dialogue systems, focus on generating responses in loosely structured or unconstrained conversational contexts, simulating natural exchanges; and contemporary conversational system architectures often integrate both paradigms, combining state-driven control logic with LLM-based natural language generation to support both precise command execution and adaptive, context-aware interaction. From the perspective of this paper's scope, it is also important to consider that the conversational system becomes an integral part of a socio-technical information system, which in turn supports a specific business process, with the conversational system serving as one of its components. The use of an LLM, as proposed to motivate the requirements specification strategy presented in this paper, falls within a combination of paradigms: it is not fully task-oriented, as it was designed primarily for an information provider without engaging in problem-solving functionalities, nor is it strictly open-domain, since it is assumed to be restricted to a specific application domain - in this case, the financial domain. For simplicity, we refer to this system as an LLM-based chatbot.

2.2. Large Language Models

Large Language Models are advanced neural language models based on the Transformer architecture trained on massive datasets, which processes sequences of words or tokens and can be applied to a variety of natural language processing tasks [Minaee et al. 2024]. Their power resides in their ability to internalize contextual information, allowing for language generation and understanding. Two main usage paradigms have emerged in the LLM context [Liu et al. 2023]: (i) the pre-train and fine-tune paradigm, in which a pre-trained model is adapted to downstream tasks through supervised learning, and (ii) the pre-train, prompt, and predict paradigm, where the model remains fixed and is guided to perform a task through a crafted textual input called a prompt. The latter gives rise to prompt engineering, a discipline that focuses on crafting textual inputs, or prompts, that guide LLMs to generate coherent outputs. Prompts may range from simple instructions to elaborate sequences of reasoning steps, enabling LLMs to perform complex reasoning without altering their internal parameters. Additionally, procedures known as Retrieval-Augmented Generation [Gao et al. 2024], which rely on retrieving content from vector-represented texts, leverage external knowledge sources to make LLM inference aligned with domain-specific information and improving the generated outputs. These developments illustrate a broader trend: LLMs are no longer isolated tools but are becoming functional components of socio-technical information systems, reshaping how humans interact with information and how systems deliver knowledge-driven services.

2.3. Requirements Classification

In order to ensure conceptual alignment with the terminology and perspectives commonly adopted in the requirements engineering community, this subsection revisits key definitions and categorizations as defined in [Wiegers and Beatty 2013]. Rather than introducing foundational knowledge, our goal is to establish a shared vocabulary that supports the rationale behind our proposed template and clarifies how functional and non-functional requirements are organized and addressed in this work. For this study, functional requirements are defined as a description of a behavior that a system will exhibit under specific conditions, while nonfunctional requirement are descriptions of a property or characteristic that a system must exhibit or a constraint that it must respect. We also draw on the definitions of external and internal quality along with their associated attributes. External quality relies on factors that are primarily important to end users, encompassing availability, integrity, interoperability, performance, reliability, robustness, safety, security, and usability; and internal quality refers to factors that are more significant to development and maintenance teams, representing attributes that indirectly contribute to customer satisfaction by making the product easier to enhance, correct, test, and migrate to new platforms. These include efficiency, modifiability, portability, reusability, and verifiability.

2.4. Related work

The integration of LLMs into socio-technical systems has drawn attention to the critical role of requirements engineering, an area still immature in addressing the challenges posed by LLM-based systems. Recent studies by [Solomon et al. 2024, Cabrera et al. 2024, Gallegos et al. 2024] highlight the need for verification, traceability, and alignment to ensure predictable, compliant, and user-centered behavior.

In the context of chatbots, applying requirements engineering remains challenging, especially regarding the elicitation and documentation of system-specific requirements. Existing approaches, such as the documentation model for "conversational requirements" proposed by [Gonçalves and Sousa 2024] and the elicitation technique introduced by [Gerstberger et al. 2024], focus on traditional chatbots rather than systems powered by LLMs, leaving gaps unaddressed. Similarly, [Mafra et al. 2022] conducted a survey of quality attributes for conversational systems, consolidating 82 requirements into a set of desirable features for chatbots. While this effort underscores the relevance of precise requirements specification for conversational systems, it too is limited to traditional architectures and does not consider the new dimensions introduced by LLM-based designs, such as emergent behaviors or semantic function mappings.

Given our project's context in a financial business process, LLM-powered chatbots must meet not only functional and non-functional requirements but also demanding regulatory constraints. Studies shows how domain-specific demands shape system design and evaluation [Tatsat and Shater 2025, Gupta et al. 2025]. For instance, highlights interpretability is key for LLM systems in finance [Tatsat and Shater 2025], while stresses the need for efficient, accountable, and measurable AI integration in critical business environments [Gupta et al. 2025].

Our work, in line with recent studies [Cabrera et al. 2024, Solomon et al. 2024, Gallegos et al. 2024], revisits the importance of verifiability, traceability, and alignment for LLM-powered chatbot systems by presenting associated requirements in a structured manner. Similarly to [Gonçalves and Sousa 2024, Gerstberger et al. 2024], we follow an approach focused on improving requirements documentation for chatbot systems, as also proposed by [Mafra et al. 2022]. However, unlike the latter, we direct our efforts toward eliciting, organizing, and documenting requirements for a user-centered socio-technical system powered by LLMs, in order to address the gap left by important prior works, as [Mafra et al. 2022]. Thus, based on the proposed template-driven approach, we aim to apply the interpretability emphasized by [Tatsat and Shater 2025] and to make the integration process of advanced generative models into business systems more consistent with the practical perspective discussed by [Gupta et al. 2025].

3. Method: Template Specification and Requirements Elicitation

Defining LLM-based chatbot requirements revealed challenges in organizing and structuring them. This highlighted the lack of clear guidelines for specification and led us to develop a conceptual template as a methodological artifact. The template provided a structured approach for exploring functional and non-functional requirements. This artifact enabled us to advance toward a more comprehensive and detailed specification, informed by insights from scientific literature and white papers. The process reflects an exploratory and iterative study aimed at aligning system requirements with the principles and practices of LLM-based systems. Figure 1 illustrates the overall structure of the method. The remainder of the section discusses each step in detail.

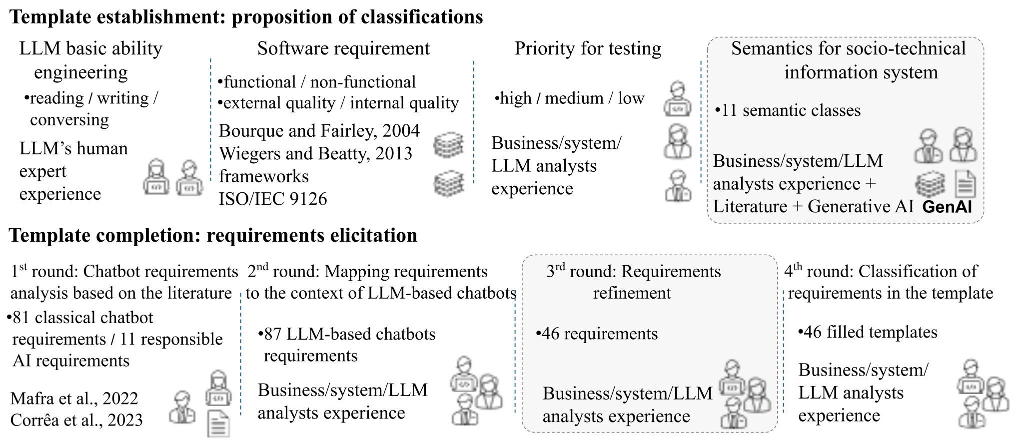

Figure 1. Method – overall structure. The steps highlighted in gray occurred interactively, exchanging knowledge during the execution of the method.

The objective of establishing a template required the proposal of a classification scheme to organize the requirements according to different interpretative perspectives. To this end, four perspectives were defined: (i) basic LLM capabilities, (ii) traditional software engineering concerns, (iii) testing priority, and (iv) the semantic interpretation of requirements within the context of a socio-technical information system.

LLM basic abilities: An LLM, in its essence, is a function trained to predict the probabilities of subsequent tokens given a sequence of tokens. As a result, it can mimic the act of "reading" a text excerpt and "writing" a continuation for it. Furthermore, adequate training can guide the prediction process, producing a "conversation"-like effect. Thus, three fundamental abilities have been established: *reading*, *writing*, and *conversing*. A LLM-based system requirement may be related to one or more of these abilities.

Software engineering: Software engineering field provides a classification for system requirements that defines what the software must do (functional requirements) and how it should operate (non-functional requirement). The classification in this requirements specification project follows the theoretical frameworks [Bourque and Fairley 2004, Wiegers and Beatty 2013] and the ISO/IEC 9126 standard. Still following the theoretical frameworks, internal and external quality aspects subdivide non-functional requirements and organize them according to 16 factors relevant to both end users and technical teams.

Priority for testing: All requirements specified for the system should, in principle, be fulfilled by the system under development, and their fulfillment should be assessed and monitored throughout both development and production phases. However, certain requirements may be prioritized over others due to constraints imposed by the supported business process, development time and budget limitations, evaluation and monitoring feasibility. This is particularly relevant in the context of LLM-based systems, where strategies for verifying and validating some types of requirements remain an open challenge, introducing layers of complexity to the requirements engineering process. Thus, this perspective supports prioritizing time and effort in requirements analysis.

Semantic interpretation of requirements within the context of a socio-technical information system: During the template construction process, it became evident that defining the positioning of each requirement within the context of the socio-technical information system was valuable. Organizational managers require contextual information to establish priorities and strategies related to the proposed requirements. The definition of these categories emerged through an inductive and iterative process conducted during the requirements gathering phase. Business and systems analysts compiled an initial list of requirements based on a review of relevant literature, including scientific articles and white papers. These requirements were then subjected to topic analysis using prompt engineering techniques and interactions with language models. The resulting insights were reviewed by the analysts, leading to the establishment of a set of 11 semantic classes.

The set of classes derived from these four perspectives results in a total of 35 possible classifications for a given requirement. Figure 2 illustrates both the aesthetic organization of the template, presented in the form of a card, and the structure of a requirement within it. Then, the requirements elicitation process and their organization into the template for building the proof of concept were carried out in four rounds. In the first round, a literature review was conducted to identify an initial set of desirable requirements for chatbot systems. In the second round, the collected requirements were mapped in the context of an LLM-based chatbot. In the third round, these requirements were refined by excluding those considered irrelevant and grouping similar ones. Finally, in the fourth round, the requirements were classified according to the categories defined in the template, and the final cards were produced.

REQUIREMENT DEFINITION								
ID:	R17	**NAME:**	Not use constructions or terms with pejorative meanings or origins, nor any content with prejudiced connotations.					
REQUIREMENT DETAILS								
REQUIREMENT TYPE		**LLM ABILITY**					**PRIORITY**	
☐	Functional$_{1}$	☐		Reading$_{3}$			☑	High$_{6}$
☑	Non functional$_{2}$	☑		Writing$_{4}$			☐	Medium$_{7}$
		☐		Conversation$_{5}$			☐	Low$_{8}$
QUALITY CLASSIFICATION								
External quality								
Reliability$_{9}$	**Availability$_{10}$**	**Robustness$_{11}$**	**Performance$_{12}$**	**Integrity$_{13}$**	**Safety$_{14}$**	**Security$_{15}$**	**Usability$_{16}$**	**Interoperability$_{17}$**
☐	☐	☐	☐	☐	☑	☐	☑	☐
Internal quality								
Efficiency$_{18}$	**Scalability$_{19}$**	**Portability$_{20}$**	**Reusability$_{21}$**	**Modifiability$_{22}$**	**Verifiability$_{23}$**	**Compliance$_{24}$**		
☐	☐	☐	☐	☐	☐	☑		
SEMANTIC CLASSIFICATION								
Management of complex interactions, scope, compliance, and privacy$_{25}$							☑	
Journey structure$_{26}$							☐	
Communication style$_{27}$							☑	
Technical and linguistic accuracy$_{28}$							☐	
Interaction and responsiveness$_{29}$							☐	
Management of graphical interface elements$_{30}$							☐	
Contextual understanding and memory$_{31}$							☐	
Proactivity and support$_{32}$							☐	
LLM Autonomy and user-specific customization$_{33}$							☐	
Persona$_{34}$							☐	
Error handling and limitations$_{35}$							☐	

Figure 2. Template in the form of a card for requirements description.

4. Requirements Cards

Our method facilitated the organization of a set of 46 requirements (Tables 1–3)[1], enabling a clearer understanding of the problem at hand and supporting its systematic exploration. Thus, the work is characterized as an exploratory study that makes the specification challenges more explicit and tractable. Figure 3 shows the distribution of the requirements elicited according to the categories in the template. The group of 46 requirements exhibits a bias towards aspects related to usability, context understanding and memory, and conversational ability. This result aligns with the value LLMs bring to chatbot systems.

Table 1. Requirements “mainly” for managing interactions; achieving linguistic and technical precision; adapting communication style. The LLM must...

... ensure compliance with regulatory standards and the company’s internal policies.
... always act within its scope.
... manage complex interactions, including unexpected or negative requests, guiding users to their goals.
... ensure that all interactions comply with privacy standards and company policies, particularly with regard to user data and information security.
... not make value judgments based on the information provided by the user, remaining transparent in its responses and acting in a way that always helps the user in their search.
... not proceed with requests that require the disclosure of sensitive data if it has information indicating the user is under 18 years old.
... make it clear who has access to the conversation, if requested by the user.
... must treat all individuals equally.
... provide objective yet pleasant, clear, respectful, and appropriately designed responses based on the user profile and the complexity of the interaction.
... avoid terms or content with pejorative meanings, origins, or prejudiced connotations.
... provide all information on a subject and not withhold it based on arbitrary criteria.
... provide accurate answers that are relevant to the user’s context.
... use domain-specific vocabulary accurately and ensure correctness of language.
... handle spelling mistakes, grammatical errors and poor sentence construction from the user without hindering the quality of the conversation.
... use active voice.
... avoid verbosity and ensure that responses are complete but concise.
... be able to understand linguistic styles, including symbolic language and cultural diversity.
... make appropriate use of linguistic styles, jargon and technical terms relevant to the financial domain.
... be pleasant and empathetic.

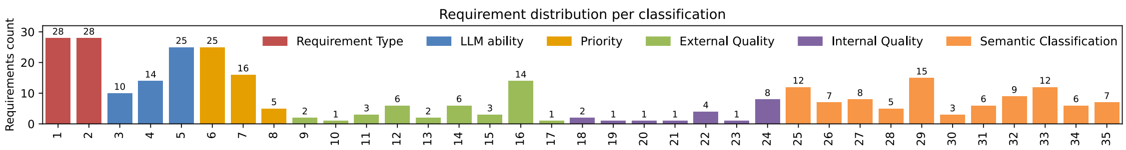

Figure 3. Distribution of the elicited requirements across the categories. The numbers on each bar correspond to the subscript numbers in Figure 2.

The development of our method required acknowledging the probabilistic nature inherent to conversational flows powered by LLMs. This property fundamentally shapes the development cycle of LLM-based chatbots, influencing how the problem space is perceived and addressed. Thus, the ability to specify requirements that explicitly account for

[1] `https://github.com/c2d-usp/requisitos_chatbot`

Table 2. Requirements mainly for responsiveness, contextual understanding and memory, journey structuring. The LLM must...

... be able to initiate its participation autonomously, or reacting to user input, if any, adjusting to the context or intent of the user's statement.
... end conversations politely and positively, inviting the user to provide feedback and confirming task completion as needed.
... engage users with interactive elements (e.g., queries about satisfaction or need for extra information), prompted by the observation of distractions, without explicitly indicating that distractions were observed, but reacting with appropriate responses to ensure an effective exchange of information.
... manage ambiguities introduced into the chat by the user, confirming that its understanding is correct.
... be able to clarify user's doubts and confusion, generating alternative explanations and summaries.
... be able to deal with unexpected situations, for example, by clearly stating that it cannot or will not be able to deal with a certain subject.
... be able to adjust the detail of the responses based on the user's profile or preferences.
... acknowledge difficulty in understanding ironic, metaphorical and similar contexts, clearly expressing not been able to understand the user's speech and requesting clarification.
... transparently deal with errors or limitations in understanding user input or fulfilling requests, advising users of alternative sources of information or help when necessary.
... be able to identify the user's intent.
... consider the entire context of the conversation, remembering previously mentioned details to provide relevant information or solutions and asking for additional information only when necessary.
... be able to explain to the user, when needed, that certain information or conditions were not taken into account in the response.
... be proactive in offering assistance based on the current situation, the user profile, and the provided information, tailoring responses to meet the specific needs of the user at that moment.
... present itself to the user as a computational agent and briefly explain its purpose.
... correctly identify and accept the user's intent to end the conversation.
... be able to assess the structure of the conversation and ensure the following phases: introduction, identification of intention, collection of information, resolution of the problem, confirmation, closure, request for evaluation.

this characteristic impacts on system architecture design. Rather than treating the LLM as a black box, the formal specification of requirements guides the creation of auxiliary software components, such as validation layers, fallback mechanisms, and monitoring modules. This approach enables tracing every desired behavior to a formal requirement and a corresponding mechanism for its measurement and enforcement.

Furthermore, specifying and structuring system requirements, as well as presenting them to stakeholders, is essential for establishing a shared understanding of functionalities and constraints. A well-defined requirements document is critical for this process. Additionally, visualizing requirement characteristics supports stakeholder decision-making by making the evaluation of features and constraints more traceable and efficient. For example, Figure 2 shows a requirement concerning pejorative or prejudiced terms. By analyzing it based on LLM "writing" capability, with "high" business priority, reflected software characteristics ("safety", "usability", "compliance"), and semantic meaning, we gain a clear view of a LLM role in the system and its effect in a business risk matrix.

5. Conclusion

This paper presented a template-driven approach for specifying requirements in LLM-based chatbots. Using a card-based format, it organizes functional and non-functional requirements to enhance clarity, traceability, and alignment with system objectives. A

Table 3. Requirements mainly for chatbot persona, user autonomy and personalization, and proactivity and support. The LLM must...

... maintain a consistent persona appropriated to its purpose and conversational domain, presenting itself clearly about its capabilities and limitations.
... be able to explain to the user how to best interact with it.
... allow users to start and end conversations, customize their interaction preferences, and freely choose conversation topics or questions.
... learn from user preferences to improve future interactions.
... not ask the user questions, except to request clarification when necessary.
... support informed decision-making and should not make decisions or influence decision-making.
... not disregard any user input.
... take into account the user's limitations and interests, allowing the user experience to be viable.
... be able to clarify the type of information it needs to continue the conversation, if the user does not give it the correct information or if it encounters a limitation in understanding due to a lack of information.
... have the ability to determine changes in the avatar, such as colors, position, or facial expressions, if there is an avatar associated with the chatbot.
... be aligned with the interface of the system that supports it, being able to guide the user on the conversation turn, modulate their response time, and indicate whether a response is being processed.

proof-of-concept in a financial business process context demonstrated the template's practical applicability and its capacity to isolate LLM-specific operations from broader system requirements. The semantic classes defined in the template and the elicited requirements should not be regarded as definitive; rather, they result from an initial exploratory study within a specific business context. The approach is designed to be adaptable, enabling integration into diverse business environments. Future work will focus on refining the method through human-in-the-loop collaboration with domain experts and on extending the template to incorporate norms more explicitly oriented to requirement specification.

Acknowledgements

The authors would like to thank the Programa de Bolsas Itaú (PBI) of the Centro de Ciência de Dados (C^2D), supported by Itaú Unibanco S.A

References

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., and Zimmermann, T. (2019). Software engeneering for machine learning: A case study. In *IEEE/ACM 41st Int. Conf. on Software Eng.: Software Eng. in Pract.*

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *Preprint arXiv:2108.07258.*

Bourque, P. and Fairley, R. (2004). Swebok. *Nd: IEEE Comput. soc.*

Cabrera, C., Bastidas, V., Schooling, J., and Lawrence, N. D. (2024). The Systems Engeneering approach in times of large language models. *Preprint arXiv:2411.09050.*

Dam, S. K., Hong, C. S., Qiao, Y., and Zhang, C. (2024). A complete survey on llm-based ai chatbots. *Preprint arXiv:2406.16937.*

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., and Ahmed, N. K. (2024). Bias and fairness in large language models: A survey. *Comput. Ling.*

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *Preprint arXiv:2312.10997*.

Gerstberger, W. S., Silva, W. A. F., and Guedes, G. T. A. (2024). Bart: Uma técnica de elicitação de requisitos para sistemas multiagentes. In *Workshop sobre Bots na Eng. de Software*.

Gonçalves, L. P. and Sousa, G. R. (2024). Documentation Artifacts For Conversation-Related Requirements Specification in Chatbots. In *Anais do WER24-Workshop em Eng. de Requisitos, Buenos Aires, Argentina*.

Gupta, S., Ranjan, R., and Singh, S. N. (2025). Comprehensive framework for evaluating conversational ai chatbots. *Preprint arXiv:2502.06105*.

Jurafsky, D. and Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Online manuscript released Jan. 12, 2025.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*

Mafra, M., Nunes, K., Castro, A., Lopes, A., Oran, A. C., Braz Junior, G., Almeida, J., Paiva, A., Silva, A., Rocha, S., et al. (2022). Defining requirements for the development of useful and usable chatbots: An analyisis of quality attributes from academy and industry. In *Int. Conf. on Hum.-Comput. Interact.*

Martínez-Fernández, S., Bogner, J., Franch, X., Oriol, M., Siebert, J., Trendowicz, A., Vollmer, A. M., and Wagner, S. (2022). Software engeneering for ai-based systems: a survey. *ACM Trans. on Software Eng. and Methodol.*

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. (2024). Large language models: A survey. *Preprint arXiv:2402.06196*.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. (2019). Model cards for model reporting. In *Proc. of the Conf. on Fairness, Account., and Transparency*.

Sarker, I. H. (2024). Llm potentiality and awareness: a position paper from the perspective of trustworthy and responsible ai modeling. *Discover Artif. Intell.*

Solomon, A., Levy, M., Agur-Cohen, D., Younis, M., and Moshe, E. (2024). Requiremets engeneering for llm: The case of digital inquiries application. In *32nd Int. Requir. Eng. Conf. Workshops (REW)*.

Tatsat, H. and Shater, A. (2025). Beyond the black box: Interpretability of llms in finance. *Preprint arXiv:2505.24650*.

Wiegers, K. and Beatty, J. (2013). *Software requirements*. Pearson Education.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *Preprint arXiv:2303.18223*.