The Role of AI in Software Requirements Engineering and System Design

Bekzhan Kimadiyev
Bcs Computer Science
Nazarbayev University
Astana, Kazakhstan
bekzhan.kimadiyev@nu.edu.kz

***Abstract*—The integration of Artificial Intelligence (AI) into software requirements engineering (SRE) and system design has opened new avenues for improving efficiency, accuracy, and scalability. This paper presents a comprehensive survey of recent advances in AI-driven requirements engineering and system design. It examines current AI applications, challenges, and future trends by drawing on recent studies. The analysis reveals that while AI offers promising capabilities for automating requirements extraction, analysis, and design, challenges related to contextual understanding, explainability, and ethical considerations remain significant.**

***Index Terms*—Artificial Intelligence, Requirements Engineering, System Design, Natural Language Processing, Generative AI**

I. Introduction

AI has experienced rapid technological development which now influences software engineering through its new capabilities. Software engineering received profound changes from this development specifically in the domain of software requirements engineering and system design. Software systems grow more complex day by day. The traditional methods used for requirements gathering and design show increasing inadequacy. Traditional methods prove insufficient for meeting both the accuracy requirements and rising efficiency needs [4].

These technologies include natural language processing (NLP) as well as generative AI models. These tools serve as important automation systems to improve such processes [1]. AI-powered tools represented by GitHub Copilot prove effective through their ability to generate code. The technology demonstrates its ability to help developers close performance gaps during their work between requirements definition and outcome implementation [2]. The implementation process receives help from developers through code-generation support and requirements analysis [2].

However, the integration of AI in developing SRE with AI technologies faces difficulties due to uncertainties regarding AI-supported tools' precision and requirements understanding abilities and explainability requirements [3]. AI-supported tools operate best with project requirements which are correctly understood and AI also needs explainable operations.

The research analyzes the impact that AI technology has on SRE processes. The paper investigates modern AI applications in system design and development through a review of present-day implementations, their benefits, limitations, and future trends [5]. By analyzing recent studies, the survey bases its findings on recent research to generate a thorough understanding. The early phases of software development stand to receive major changes from AI technology applications throughout the software development lifecycle.

II. AI Applications in Software Requirements Engineering

A. Automated Requirements Extraction and Analysis

NLP techniques serve as key elements which drive the automated process of requirements extracting throughout software engineering. AI systems with NLP capabilities extract requirements from unstructured data in documents and emails and transcripts with high accuracy for classification purposes [4]. AI-driven systems analyze vast datasets at high speed and spot essential requirements and conflicting elements because they do not share traditional methods' human errors and time constraints.

Sentiment analysis tools are becoming standard practices for measuring requirements urgency and significance from stakeholder point of views. Artificial intelligence uses linguistic indicators to establish which requirements correspond to essential business essentials. The application of Latent Dirichlet Allocation (LDA) as a topic modeling technique helps divide requirements into grouped themes to create a more organized requirements analysis process [5].

Through named entity recognition techniques AI systems extract specific terms with their relationships from requirements documents while adding sentiment analysis and topic modeling functions. Through this capability the requirements engineering process gains speed and accuracy improves by reducing human error. The main obstacles lie in processing domain-specific contextual information as well as managing unclear requirements that demand specialized domain expertise. For better contextual understanding and extracted requirement relevance AI systems need domain ontologies to solve the existing challenges.

B. Challenges in AI-driven Requirements Engineering

The complete implementation of AI-driven requirements engineering faces several obstacles as it provides multiple benefits. The main difficulty emerges from obtaining proper contextual understanding of requirements. AI models along

with their machine learning foundation fail to correctly understand requirements that are ambiguous, indirect or dependent on context according to [4]. The problem becomes more complex due to inconsistent language used by stakeholders who participate in requirements engineering.

The main problem with AI models is their difficulty in generating understandable explanations which experts identify as the "black box" challenge. AI-generated insights find limited acceptance from stakeholders because they require full explanation of how final outputs are generated. XAI approaches serve as solutions to clarify AI-driven decisions through interpretable explanations. The integration of performance and explainability requires complex solutions which produce minimal success. Researchers evaluate the combination of machine learning models with rule-based systems as a promising approach to handle this challenge.

AI-driven requirements analysis quality heavily depends on how accurately the training data represents the system. Training data that contains biases will produce distorted analysis outcomes that might ignore vital requirements and mistake them. Ongoing investigations aim to develop debiasing approaches and enhanced data preprocessing methodologies because these measures reduce the identified risks. The proper resolution of these challenges depends on better techniques for training data management alongside improved NLP approaches and validated systems according to [2].

III. The Role of Generative AI in System Design

A. Generative AI for Design Prototyping

The transformer-based generative AI models demonstrate strong potential to automate the design prototyping procedure. AI systems use large-scale pre-training along with textual requirements to create architectural diagrams and user interface mockups combined with design alternative proposals. The system architecture benefits from this ability because it speeds up design time while providing designers with a better framework for exploratory design approaches [3].

The design optimization process receives help from generative AI because it extends beyond human capabilities to examine multiple design options that surpass human designers' practical analysis scope. Artificial intelligence creates original prototypes which human designers can improve after reducing workload to accelerate development timing. Generating prototypes alongside the capability to model performance challenges leads generative AI towards recommending necessary architectural solutions for enhancement.

The extensive training data used by generative AI systems creates potential performance risks which arise from underrepresented design contexts and biased training data. The application of feedback systems incorporating human experts who check and optimize AI-created models results in more dependable and suitable outcomes from these designs. The implementation of explainability methods in generative AI systems enables stakeholders to comprehend recommendations that AI produces.

B. Limitations and Risks of AI in System Design

When AI enters system design practices it brings certain security issues. The main drawback of using AI models stems from their requirement of extensive large-scale datasets during training. The algorithms trained through AI systems will reproduce any biases found in the provided training data thus leading to results that either perform poorly or create discriminatory practices. AI systems confront limitations in understanding complex domain requirements which prevents their use in critically important fields of healthcare and finance because these domains need precise regulatory adherence.

A major drawback of AI-generated designs rests in their challenging nature of obtaining validation. AI-generated designs deviate from standard practices since they can develop novel approaches that need comprehensive testing together with validation processes to verify their compliance with requirements and reliability. Stakeholders encounter major risks from AI-generated designs because they need transparent explanations to understand and trust these recommendations according to [5].

Human experts should review AI-generated designs through human-in-the-loop procedures in order to minimize certain risks. This additional complexity reduces some of the system efficiency benefits brought by AI automation systems. Studies about hybrid AI systems merging rule-based reasoning with generative AI are expanding because they provide an effective method to harmonize innovative benefits with safe operations.

The application of artificial intelligence connects requirements engineering to system design process.

C. AI-Assisted Traceability and Consistency Checks

Software development depends on traceability because it maintains the continuous reflection of requirements between different development phases. The implementation of artificial intelligence leads to better traceability because it creates automatic connections between requirements and design aspects and software code blocks. Machine learning algorithms review project archives through data analysis to create patterns which generate traceability connections that help reduce worker involvement while decreasing the chance of errors [5].

AI tools through automation accomplish product requirement consistency verification by scanning multiple levels of design content until they detect improper requirements changes. AI detection mechanisms determine when a requirement shows incomplete or incorrect application so that teams can start corrective procedures early. AITraceability provides ongoing requirements-design feedback which maintains project alignment throughout development while it minimizes project repetition and boosts system quality.

D. Integration Challenges

The integration of AI tools into existing software development workflows requires handling two main obstacles which include adapting to legacy systems and obtaining substantial training data. AI implementation needs precise boundaries

between AI-automatable work and human-driven work particularly regarding decision making and handling unusual cases [2].

The implementation of AI tools becomes complicated because there exist no unified protocols or APIs for integrating AI services. The necessary development of industry-wide guidelines for AI interoperability standards and compliance enables ease of integration across systems. It is essential to handle privacy and security concerns specifically because AI systems need to handle sensitive information for training purposes and making decisions.

Research institutions are now intensively studying frameworks for governing AI systems as well as best practices for incorporating AI technologies. These frameworks serve as guidance to handle AI risks along with maintaining organizational transparency and maintaining a proper alignment between AI tools and organizational objectives.

REFERENCES

[1] C. Bull and A. Kharrufa, "Generative AI Assistants in Software Development Education," *arXiv preprint arXiv:2303.11612*, Mar. 2023.

[2] R. Pudari and N. A. Ernst, "From Copilot to Pilot: Towards AI Supported Software Development," *arXiv preprint arXiv:2303.13629*, Mar. 2023.

[3] J. Sauvola et al., "Future of Software Development with Generative AI," *J. Autom. Softw. Eng.*, Mar. 2024, doi: 10.1007/s11047-023-09971-8.

[4] K. Ahmad et al., "Requirements Engineering for Artificial Intelligence Systems," *Inf. Softw. Technol.*, vol. 154, p. 107071, Feb. 2023, doi: 10.1016/j.infsof.2022.107071.

[5] S. Martínez-Fernández et al., "Software Engineering for AI-Based Systems: A Survey," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 2, Apr. 2022, doi: 10.1145/3487043.