International Journal of Computer Sciences and Engineering
Vol.**12**, Issue.**12**, pp.**15-24**, December **2024**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Article

Exploring the Impact of Explainable AI on Software Maintenance and Testing: A Systematic Mapping Study

Muna Alrazgan[1*], Hala Almukhalfi[2], Manal H. Alshahrani[3], Mashael Aljohani[4], Nada Almohaimeed[5], Ruba Almuwayshir[6], Zamzam Alhijji[7]

[1,2,3,4,5,6,7]Dept. of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, 11345, Saudi Arabia

**Corresponding Author: malrazgan@ksu.edu.sa.*

Received: 15/Oct/2024; **Accepted**: 16/Nov/2024; **Published**: 31/Dec/2024. **DOI**: https://doi.org/10.26438/ijcse/v12i12.1524

Abstract: This study systematically maps the integration and impact of Explainable Artificial Intelligence (XAI) in software maintenance and testing, covering research published between 2019 and 2023. Through the analysis of 18 primary papers, we identify trends and applications of XAI in these domains. Our findings reveal a growing interest in leveraging XAI to enhance the transparency and interpretability of AI models used in software maintenance and testing. Key insights include the distribution of studies over the years, the main tasks where XAI is applied, the types of XAI models used, their goals, and the various forms of XAI implementation. This systematic mapping provides a comprehensive overview of the current state of research and highlights potential areas for future exploration.

Keywords: Explainable Artificial Intelligence (XAI); Software Development Life Cycle (SDLC); Software Maintenance; Software Testing.

1. Introduction

The integration of XAI in the Software Development Life Cycle (SDLC) is an emerging research area that seeks to enhance the transparency and interpretability of AI models [1]. Explainable AI has garnered significant attention for its potential to provide clear, interpretable insights into complex AI models, thereby facilitating better decision-making and accountability. By making AI's decision-making processes understandable, XAI helps build trust in AI systems, ensuring that they are used responsibly and effectively.

This study focuses specifically on the maintenance and testing stages of the SDLC. These phases are critical as they ensure software systems' reliability and performance. The ability to understand and explain AI-driven decisions in these stages can significantly improve trust and effectiveness. For instance, explainable AI can help identify the reasons behind software anomalies or failures, providing developers with actionable insights to address issues more efficiently.

Despite its promise, the application of XAI in software maintenance and testing remains relatively unexplored. To bridge this gap, we systematically reviewed 18 primary papers, addressing five key research questions. These questions explore the chronological distribution of research on XAI in software maintenance and testing, the specific tasks within these domains that leverage XAI, the types of XAI models employed, the goals behind using XAI, and the various forms of XAI implementation. By mapping the existing literature, this study aims to provide a comprehensive overview of the current state of research and identify potential areas for future exploration.

2. Related Work

As XAI has become increasingly popular as a research domain over time, research on its developing theories, techniques, and tools has been very active in the past few years. One of the research [2] conducted a comprehensive systematic literature review (SLR) to evaluate the developments and trends in XAI, focusing on different application domains and tasks where XAI methods have been applied. The study examined information from 137 articles, emphasizing the importance of explainability in AI systems that are mainly developed for safety-critical domains. These include healthcare, finance, and automotive industries, where the need for transparency and trust in AI decisions is essential. Therefore, most of the works in literature come from the medical and health care domain. Also, the study [3] analyzed 179 articles, highlighting a growing interest in ML interpretability, especially in the medical domain. Most of these studies proposed solutions and conducted experiments to enhance the interpretability of ML models, particularly those based on artificial neural networks.

Another comprehensive systematic review [4] which not only discusses the diverse applications of XAI in several domains such as healthcare, finance, transportation, and manufacturing, but also delves into various methodologies and frameworks used to implement XAI, thereby providing a solid foundation for future research. It emphasizes the role of both knowledge-driven and data-driven approaches to explanation. It suggests that a deeper understanding of XAI systems is necessary for their ethical and effective implementation in real-world applications.

Moreover, the integration of XAI into various technology domains has been an emerging focus in research. A significant contribution in this area is the systematic mapping study that illustrates the role of XAI within the domain of AI ethics [5]. The study identifies that while AI ethics is a rapidly developing field, it suffers from a lack of common frameworks and clear conceptualization, particularly in how understandable AI systems can enhance ethical considerations. It emphasizes the importance of XAI in making AI systems more transparent and trustworthy, addressing critical issues such as algorithmic biases that could potentially favor certain groups over others during decision-making processes, such as hiring. It offers a structured visualization of how, when, and why XAI has been studied within AI ethics, highlighting the significant gaps and focal points of existing research.

On the other hand, the systematic literature review [6] explores the integration of XAI within the domain of Software Engineering (SE). The study investigates the extent of XAI application across various SE tasks through a review of 24 relevant studies. The findings highlight that software maintenance, particularly defect prediction, is the predominant area where XAI has been applied, representing 68% of the cases. The paper identifies a preference for using XAI with classical ML models over more complex ones, and notes a significant gap in standard evaluation metrics for XAI methods. Despite these challenges, XAI is recognized as a valuable tool that enhances trust and understanding in ML solutions for SE, opening the way for further exploration in areas like testing and program repair. The authors call for more research focused on expanding the range of SE tasks addressed by XAI and improving the interpretability of complex ML models.

While the above literatures provide important insights into the application of XAI within various domains of Software Engineering, our review reveals a notable gap. Despite the advancements reported by [6] in applying XAI to software maintenance, there remains a lack of systematic exploration into the integration of XAI methods specifically within both software maintenance and testing tasks. To our knowledge, none of the studies comprehensively exploit XAI techniques to enhance the processes and outcomes in these critical areas of Software Engineering. This observation indicates the unique contribution of our research, aiming to bridge this gap by systematically mapping the potential applications of XAI in software maintenance and testing tasks.

3. Systematic Review Planning

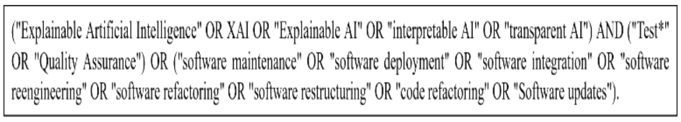
This SLR follows Kitchenham et al. [7], which ensures that the research is replicable and unbiased. The methodology is divided into three phases: planning, conducting and reporting. The first phase identifies the need for a systematic exploration of existing knowledge and formulating research questions. The second phase determines data sources and develops a search strategy. The final phase presents results, highlights gaps, and suggests potential areas for future research. The following represents how each phase is reflected in our topic.

3.1. Review Planning

The SLR aims to examine Explainable AI in software maintenance and testing. Each phase of the review is designed to uncover how XAI technologies are currently integrated in these critical areas of software engineering. The research questions related to this are:
RQ1: How have the studies on Explainable AI in software maintenance and testing been distributed over the years and across major academic libraries?
RQ2: What are the main Software Maintenance and Testing tasks for which Explainable AI has been applied?
RQ3: What are Explainable AI models used within software maintenance and testing?
RQ4: What are the goals of using Explainable AI within software maintenance and testing?
RQ5: What are the different forms of providing Explainable AI within software maintenance and testing?

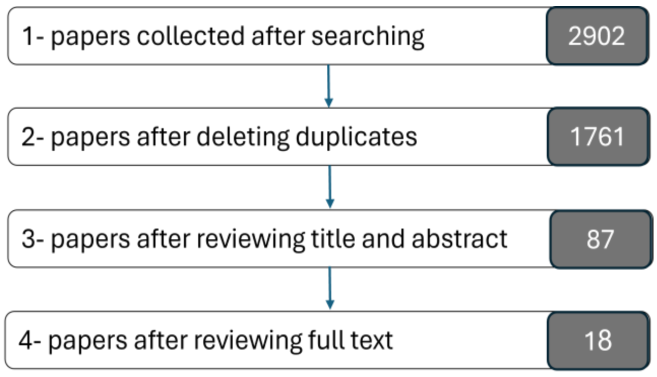
3.2. Review Conducting

A comprehensive search strategy was conducted across multiple databases, including IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect, covering relevant literature from 2019 to 2023. The search used keywords as the following:


("Explainable Artificial Intelligence" OR XAI OR "Explainable AI" OR "interpretable AI" OR "transparent AI") AND ("Test*" OR "Quality Assurance") OR ("software maintenance" OR "software deployment" OR "software integration" OR "software reengineering" OR "software refactoring" OR "software restructuring" OR "code refactoring" OR "Software updates").

The initial retrieval resulted in 2902 papers, managed using Rayyan, a web-based tool that supports collaborative screening and data management. Specific inclusion and exclusion criteria guided the selection process(Inclusion Criteria):

- Publications from the last five years (2019 to 2023).
- Studies that specifically address the application of XAI within the software engineering testing and maintenance phases.
- Research articles and papers published in journals or conference proceedings.


1- papers collected after searching
2902
2- papers after deleting duplicates
1761
3- papers after reviewing title and abstract
87
4- papers after reviewing full text
18
Figure 1. Steps of the papers selection process

Exclusion Criteria:

- Publications outside the year range of 2019-2023.
- Papers not written in English.
- Studies that focus solely on artificial intelligence without components of explainability.
- Research outside the context of software engineering, such as studies purely focused on medical or financial topics.
- Non-academic sources, including blogs, news articles, and white papers.

The Figure 1 , shows the processes of selecting studies for the research. After the initial screening and removal of duplicate entries, the number of papers was narrowed down from 2,902 to 1,761. These papers underwent a title and abstract screening process resulting in 87 papers. Subsequently, a comprehensive full-text review was conducted, further refining the selection to 18 papers. These studies specifically address how XAI technologies are being integrated and assessed within the testing and maintenance phases of software engineering.

3.3. Review Reporting

The data from the selected studies were systematically extracted to answer the five research questions. Tools such as Zotero, Rayyan, Tableau, and Python were used to create graphs and narrative summaries that effectively present the findings. These results are detailed in Section IV of this paper, showing the impact and applications of XAI in testing and maintenance.

4. Results

The Figure 2, demonstrates the publication trends in the field of XAI in software testing and maintenance between 2019 and 2023. The publication output exhibited a steady publication rate from 2019 to 2021, with a ratio of approximately 3 papers. Notably, 2022 witnessed a growth in publications, reaching a peak of 6 papers. However, the year 2023 observed a decline in publication volume, with only 4 papers published in the field. As shown in Figure 3, the majority of publications in this field appeared in the IEEE, with a total of 13 papers. Additionally, ACM and Springer contributed only two papers each, while ScienceDirect published a single related paper.

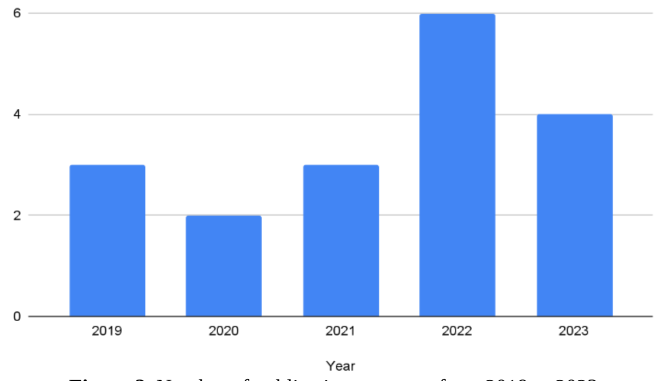


Figure 2. Number of publications per year from 2019 to 2023

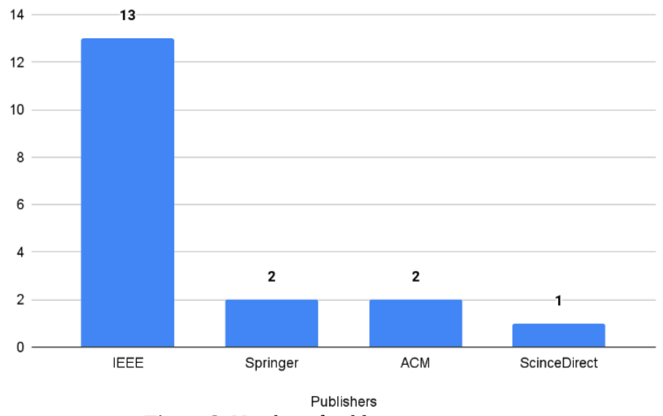


Figure 3. Number of publications per venue

To answer research question 2, we classified software testing and maintenance tasks into several categories as shown in Figure 4. The reviewed literature encompassed five distinct categories: quality assurance and testing, technical debt management, security audits and compliance, code review, and predictive maintenance. Eight of the analyzed papers provide explanations for tasks related to quality assurance and testing, in particular defect prediction [8], [9], [10], [11], [12], [13], [14], [15], [16]. A significant amount of reviewed literature falls into the security audits and compliance category. Malware detection [17], security events analysis [18], [19], cyber-attack mitigation [20], [21]. Moreover, our analysis yielded papers that addressed technical debt management [22], code review [23], and predictive maintenance [24], with each category containing a single representative paper.

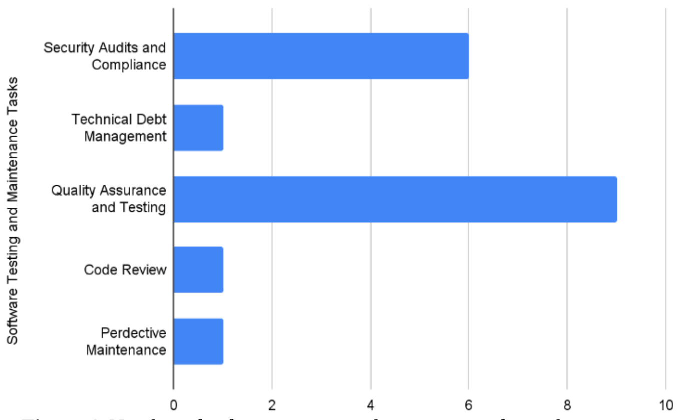


Figure 4. Number of software testing and maintenance for each category

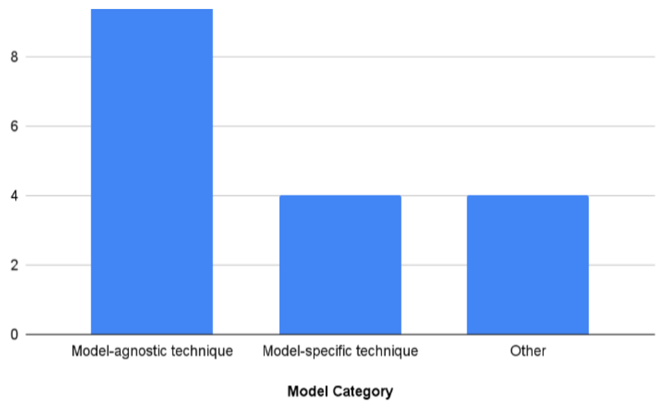


Figure 5. The distribution of XAI models across three categories: model-agnostic technique, model-specific technique, and other

To answer research question 3, the Figure 5 illustrates the model categories of the published papers. The model-agnostic technique is the most common model, and 10 papers used this category since it can be applied to any machine-learning model. These techniques include LIME [10], [11], [12], [13] which is the most used technique, followed by SHAP [14], [19] or a combination of them [9] . Other studies used PyExplainer [15], DALEX [24] and XGBoost [16] . Four papers utilized model-specific techniques that are tailored to the internal workings of a particular type of machine-learning model. Amarasinghe et al. [21] used a model-specific explanation framework that utilizes fuzzy logic and linguistic summarization to explain the decisions of a deep neural network in the security of network traffic data. MalConv [17] is another security-related model-specific technique. Through various techniques, such as gradient analysis and interpolation between samples, the model provides interpretation for malware detection decisions.

On the other hand, we identified four papers that do not use machine learning explainers but use machine learning models to provide explanation. For example, AfzaliSeresht et al. [18] proposed an AI model that utilized an Apriori-like algorithm to generate a story consisting of a chain of security events to provide contextualized interpretations. These interpretations help security analysts identify relationships between the events in monitoring systems. Another example is the use of a Convolutional Neural Network (CNN) to identify key phrases in developers' code comments that contributed to the identification of self-admitted technical debt [22].

To answer research question 4, we investigated the purposes of providing explanations of AI systems' decisions which are transparency, trust, reliability, understandability, usability, and privacy as illustrated in Figure 6. A significant proportion—approximately 83%—of the reviewed literature has more than one goal for explanations, while the remaining papers have a single goal. Most explainable AI models seek transparency that illuminates the reasoning processes by which the AI model arrives at its decisions. Sixteen out of the eighteen analyzed papers identify transparency as the principal goal of XAI. Following the emphasis on transparency, trust emerges as another key objective in seven of the analyzed papers. These papers aim to build user confidence in the AI's decision-making processes. Another important purpose of XAI is usability which ensures that the explanations provided by AI systems are accessible, understandable, and actionable for users. Usability was identified as a goal for providing explanations in three of the analyzed papers. Other goals were presented in the literature in a few papers such as reliability, understandability, and privacy.

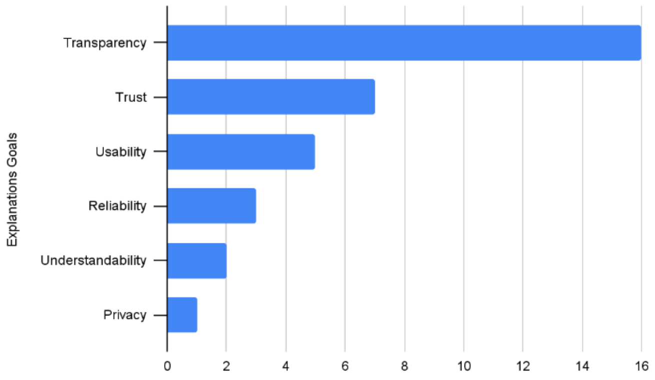


Figure 6. The recurrence of XAI explanation purposes within reviewed publications

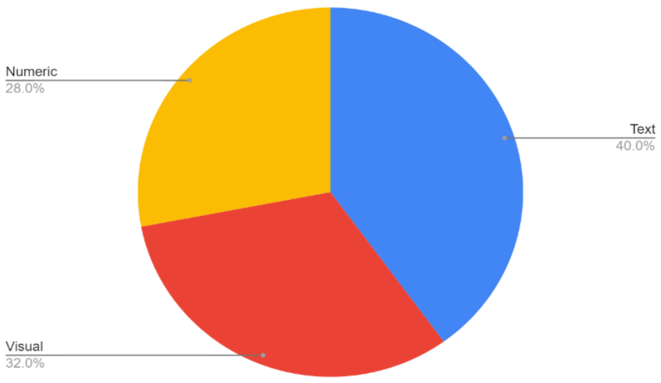


Figure 7. The distribution of the data forms used to present explanations

To address research question 5, we found that approximately 60% of the papers presented explanations using a single form of data. These data forms primarily consisted of textual, visual, or numerical data. The remaining papers used a combination of these forms to present explanations. As shown in Figure 7, the most used form of explanation is text, which was adopted by ten papers, followed by visual and numeric explanations, respectively. Text explanation combined with numeric and visual in multiple studies. Numerical data were used along with text to create explanations. These consist of scores that indicate how much each feature contributes to the model's prediction for a specific instance [9]. While grounded in numerical data, the explanations are conveyed through textual descriptions that list the features and their associated scores. On the other hand, a textual and visual explanation was provided for predictions made by the defect prediction model [12].

Visual explanations are usually presented through graphs and plots. A boxplot was utilized to illustrate the importance of features and shapely values in understanding the model's prediction of Software-Defined Networks [24]. Moreover, through plots, Zivkovic et al. [16] provided a way of illustrating how each input feature influences the prediction of a software module being defective or not. This work is similar to the contribution of other studies, which explain through charts which parts of the code are likely defective and why, based on the model's learned weights [8], [11], [16]. Several studies utilized textual explanations solely. A study attempts to facilitate the code review process by providing example reviews to help the software teams make decisions [23]. Another example of textual explanations is the tokens and syntax elements extracted from source code as features in a defect prediction model [13]. Another study introduces a framework that allows for the generation of linguistic summaries of what a neural network has learned during training [21]. In self-admitted technical debt detection, key phrases of code comments were identified and utilized as an explanation of model decisions [22]. A story consisting of a chain of security events was presented as a description to assist analysts in understanding the correlation between events [18].

5. Discussion

The discussion section provides an in-depth analysis of the temporal distribution, application areas, and the goals of implementing Explainable AI (XAI) within software maintenance and testing. It examines key trends in research publications over recent years, the various tasks in software engineering where XAI has been applied, and the specific models used to enhance maintenance and testing processes. Additionally, it explores the challenges of integrating XAI into real-world systems, and the forms in which XAI is delivered to developers and testers to improve transparency, trust, and usability. This discussion offers valuable insights into how XAI is evolving and contributing to software engineering practices.

5.1. Temporal Distribution Of Explainable AI Research In Software Maintenance And Testing

Analyzing published papers on XAI within the software engineering life cycle from 2019 to 2023 reveals notable trends. The year 2022 saw a significant peak with six publications, highlighting a surge in research activities likely driven by technological advancements and a growing emphasis on explainability in AI. In 2023, the number of publications slightly decreased, suggesting sustained interest, albeit with a potential shift towards more focused research.

The slight decrease in the number of publications on XAI within the software engineering life cycle in 2023, following the peak in 2022, could be due to several factors. Researchers may have shifted focus towards producing higher-quality, in-depth studies rather than increasing the volume of publications, reflecting a maturation of the field. The surge in 2022 might have temporarily saturated immediate research questions, necessitating more time to identify new areas of exploration. Additionally, funding and resource allocation variations could have impacted the number of new projects. Researchers might also concentrate on the practical implementation of XAI techniques, which typically require more extended time frames and result in fewer immediate publications.

In terms of publication venues, IEEE dominates with thirteen papers, indicating its pivotal role in disseminating research on XAI within software engineering. IEEE's predominance suggests that its conferences and journals are key platforms for research dissemination in this area. However, the presence of multiple publishers indicates a healthy diversity in publication venues, facilitating varied perspectives and approaches.

5.2. Exploring Key Software Maintenance And Testing Tasks Enhanced By Explainable AI

The analysis of XAI applications in software maintenance and testing tasks reveals several key areas where XAI techniques have been prominently utilized. Quality Assurance and Testing is one of the most frequently addressed tasks where XAI has been applied. XAI techniques such as LIME and SHAP are particularly suitable for this task because they provide clear and interpretable insights into why certain test cases fail, helping developers understand the underlying issues. This transparency enhances the testing process by allowing developers to focus on the most critical and problematic areas, thus improving the efficiency and effectiveness of quality assurance activities.

One detailed example is the application of PyExplainer [15], a local rule-based model-agnostic technique, was applied to explain Just-In-Time (JIT) defect predictions for large-scale open-source projects like OpenStack and Qt. By generating synthetic neighbors more similar to the target instance, PyExplainer produced more accurate local models and better explanations than LIME. Another example [9] compared post-hoc techniques like LIME and SHAP, revealing that these methods can produce conflicting justifications for the same defect prediction model, complicating their utility and interpretability. These examples highlight the challenges in ensuring consistent and reliable explanations in defect prediction models and the need for further research to improve the consistency and reliability of XAI techniques in real-world software engineering contexts.

Security audits and compliance are critical tasks where Explainable AI (XAI) proves beneficial. XAI helps security teams understand why specific security checks fail, which features contribute to vulnerabilities, and how compliance issues arise. By explaining the results of security models, XAI provides actionable insights that guide remediation efforts and strengthen security measures. It also offers transparency into risk scores and vulnerability assessments, ensuring that security actions are justifiable and compliant with standards. This clarity aids auditors in verifying the effectiveness of security protocols.

In this study [19], Shapley values were used to explain risk scores of 30,000 real-world authentication events by analyzing contextual features like IP address, device, and geolocation. This approach differentiated between attack types, such as password theft and device theft, by identifying influential factors. The insights allowed security teams to choose tailored authentication mechanisms based on specific risks, improving the overall security posture.

Despite advances, XAI has limited application in areas like predictive maintenance and technical debt management due to the complexity of processing large, variable data and the need for consistent historical data. Integrating XAI into existing tools faces technical and resource challenges, and generating explanations for complex models can be computationally demanding. The lack of standardized metrics for technical debt and the trade-off between interpretability and accuracy further complicates XAI's application. Addressing these issues requires ongoing research to refine XAI techniques and enhance their use in real-world software engineering.

5.3. Exploring The Utilization Of Explainable AI Models In Software Maintenance And Testing

Exploring the various XAI models employed in software maintenance and testing is crucial to address the research question of using Explainable AI models in these contexts. This involves identifying the categories of XAI models and understanding their specific applications and benefits in these tasks.

The Figure 5, classifies XAI models into three primary approaches: model-specific, model-agnostic, and other methods, based on the primary set of papers. Each category has distinct characteristics and applications.

- Model-specific approach: This approach is designed to explain the outputs of specific types of models. It is tailored to the inner workings of the model it aims to explain. Model-specific approaches can be beneficial for understanding complex models like neural networks in software maintenance and testing.
- Model-agnostic approach: This approach is designed to explain the outputs of any machine learning model, regardless of its type, offering flexibility and broad applicability across various models. Model-agnostic approaches are widely used in software maintenance and testing due to their versatility.
- Other approaches: This category includes various machine learning techniques used to explain results without relying on traditional XAI methods. These approaches may involve innovative methods tailored to specific needs in software maintenance and testing. Instead of post-hoc explanations, they might use integrated machine learning strategies to provide insights into model behavior and predictions.

In the primary set of papers selected for this systematic mapping, a few specific types of models are highlighted for the Model-specific approach. These models include MalConv [17] which is a deep neural network designed for malware detection, with its explanations provided through techniques like gradient analysis and sample interpolation. Similarly, the Custom Explanation Framework [21] combines fuzzy logic and linguistic summarization to explain decisions in deep neural networks. Another example is Testing with Concept Activation Vectors (TCAV) [25], a post-hoc, model-specific technique used in Explainable AI (XAI). These examples emphasize the variety and tailored nature of explanation frameworks developed for deep learning models, providing clear insights into model behavior.

Regarding model-agnostic techniques, LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) are commonly used in various studies. These techniques are valued for their versatility and broad applicability, providing insights into model behavior regardless of the underlying algorithm.

As for the “Other approaches,” there is a study from the primary set of papers [22] using various techniques beyond XAI to explain the results. One significant method mentioned is the text-mining approach used for Self-Admitted Technical Debt (SATD) classification. The text-mining method relies on traditional feature selection processes to identify SATD-indicating features from the text.

The application of XAI models in software maintenance and testing is diverse, with a strong emphasis on model-agnostic approaches due to their versatility. XAI approaches significantly enhance accuracy and trustworthiness by explaining model predictions clearly. This helps developers understand why specific predictions are made, which aids in debugging and refining machine learning models by providing clear and interpretable explanations of model predictions
.
Despite their benefits, XAI approaches face challenges related to complexity, integration, and scalability. Implementing model-specific XAI methods can be difficult, and integrating them into existing workflows often requires significant adaptation. Additionally, generating explanations can be computationally expensive, especially for large datasets or complex models. Future research should focus on simplifying XAI approaches, improving their integration, and enhancing scalability for broader adoption.

5.4. Goals Of Implementing Explainable AI In Software Maintenance And Testing

According to the analysis of the goals of implementing XAI within software maintenance and testing, which revolve around six key directions transparency, trust, reliability, understandability, usability, and privacy, the results from RQ4 in the previous section showed that transparency is the main goal in most studies as illustrated in Figure 6. , although has combined with other goals such as trust and usability as illustrated in Figure 8. Transparency is essential in maintenance and testing processes to allow developers to understand how artificial intelligence models make decisions, identify the root causes of software defects, and make informed decisions to address them. Techniques such as LIME and SHAP are widely used to provide transparency by explaining feature importance and model predictions.

Building trust in AI systems is the second most interesting goal following transparency, as we noticed that it appeared as a dual goal with transparency in some studies. The role of trust is essential in ensuring that stakeholders, including developers and testers, understand the rationale behind AI decisions, so they are more likely to trust and rely on these models. Also, particularly in high-stakes environments, it is important to provide consistent and accurate explanations that validate the AI model's predictions.

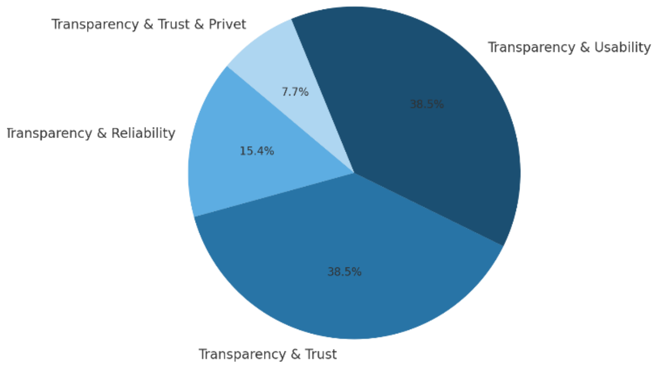


Figure 8. Repetition of Merging Transparency with Other Goals

In light of the focus on goals, Usability emerges as another primary goal in the analyzed papers. Usability ensures that the explanations provided by AI systems are accessible and actionable. Clear and straightforward explanations help developers and testers effectively use AI tools in their workflows, enhancing the overall efficiency of software maintenance and testing processes. Moreover, Usability appeared in 36% of studies combined with transparency.

Other goals discussed in the literature in a few papers include reliability, understandability, and privacy. Although these goals are important, their presence in the studies is still limited, opening opportunities for future studies to explore them further. Reliable AI systems consistently perform as expected and provide dependable outcomes, which is critical for maintaining the integrity of software systems. Furthermore, Understandability allows stakeholders, including non-technical users to understand AI decisions, bridging the gap between complex systems and users through simple language and visual aids. Besides, Privacy is paramount, especially in sensitive domains like cybersecurity and healthcare, necessitating XAI techniques that provide transparency without compromising data confidentiality. Despite their essential roles, these goals are underrepresented in current research.

5.5. Forms Of Delivering Explainable AI In Software Maintenance And Testing

The analysis of various forms of data used in Explainable AI (XAI) for software maintenance and testing indicates substantial insights into the effectiveness of these explanations. The results we reached in the RQ5 showed that 60% of the studies used single-form explanations to explain their results. As we mentioned in the results section, the forms of these models included texts, graphs, or numerical data. Textual forms are presented as the most widely used among these models, Its usage rate represented approximately 55% of single and combination forms.

Regarding studies that used textual form only. For instance, [23] explored the facilitation of the code review process by providing example reviews to help software teams make informed decisions [23]. This approach enables us to focus on relevant issues and suggests improvements, therefore enhancing the decision-making process during code review. In another study, the tokens and syntax elements extracted from source code as features in a defect prediction model, providing clear textual explanations of how specific code elements contributed to the prediction [13]. This method helps developers understand the model's decision-making process and identify potential defects based on the features extracted from the code. In other work, the study proposed a framework that generates linguistic summaries of what the neural network learned during training [21]. The importance of this framework lies in making the model learning process transparent and understandable, even for those who do not have deep technical experience in neural networks. This approach enhances the interpretability of complex models by providing high-level summaries of their internal processes.
In the context of self-admitted technical debt detection, key phrases from code comments were identified and used to explain model decisions. This method allows developers to understand the reasons behind technical debt and prioritize areas for improvement [22]. It bridges the gap between technical and non-technical stakeholders by providing accessible explanations based on natural language comments. Furthermore, AfzaliSeresht et al. [18]. presented a story consisting of a chain of security events as a description to assist analysts in understanding the correlation between events [18]. This narrative approach helps security analysts grasp complex sequences of events, making it easier to identify and mitigate potential security threats.

Despite their diversity and importance in making decision-making processes in AI systems transparent and understandable, reliance on a single form, such as texts, may restrict explanations. Combining textual explanations with visual and numerical data can provide a more comprehensive understanding of AI systems, meeting the needs of various stakeholders and enhancing the interpretability and usability of XAI systems. In the context of combining textual explanations with numerical and visual data which offers many benefits in explanations. For instance, numerical data, such as scores, are used alongside text to indicate how much each feature contributes to the model’s prediction for a specific instance.

These scores provide a quantitative basis for the explanations, hich are then detailed through textual descriptions listing the features and their associated scores. This method, as highlighted by Bose, Barao, and Liu [9], makes the data more accessible and understandable [9].

Graphs and plots are commonly employed to convey these visual explanations effectively. For instance, boxplots have been used to illustrate the significance of features and Shapley values, providing insights into the model’s predictions in SDNs. This method facilitates the identification of features that substantially influence the model’s decision-making process [24]. Additionally, Zivkovic et al. [16] utilized graphical representations to demonstrate the impact of each input feature on the prediction outcomes, specifically regarding the likelihood of software modules being defective. This visual approach aligns with other studies that employ charts to elucidate which code segments are prone to defects and underlying reasons based on the model's learned weights [24] [16].

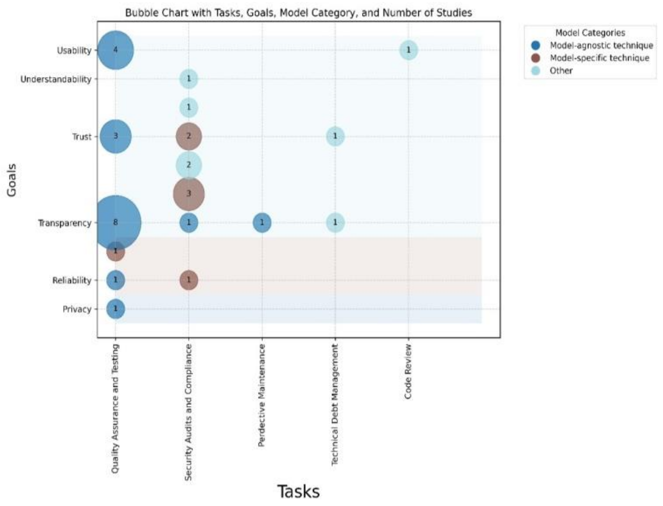


Figure 9. Bubble chart with Task, Goals, and Model categories with Number of studies

Future research should focus on developing standardized methods for integrating multiple forms of explanations to maximize the benefits of XAI in software maintenance and testing. Additionally, longitudinal studies evaluating the long-term impact of these explanations on software quality and maintenance efficiency would provide deeper insights into the effectiveness of different XAI approaches.

The Figure 9, illustrates the intersection of three main dimensions in the context of XAI, including Tasks, Models, and Goals to highlight the areas of intersection of these directions through the number of studies in them. The chart analysis emphasizes a significant focus on transparency in the context of quality assurance and testing, particularly using model-agnostic techniques. This focus is evident from the substantial number of studies addressing this area. There are eight studies focusing on transparency in the context of quality assurance and testing using model-agnostic techniques. This high number indicates the importance of this topic in current research. Transparency in quality assurance and testing allows developers and users to see the inner workings of a system clearly, enabling them to identify any issues or errors easily and contributing to building trust in the system's results and recommendations. Using model-agnostic techniques means these methods can be broadly applied without needing extensive customization for each model, enhancing these techniques' ability to provide reliable and transparent results in various contexts.

Moreover, there are four studies focusing on usability in the context of quality assurance and testing using model-agnostic techniques, indicating substantial research interest in making testing methods more user-friendly and accessible. Ensuring high usability in these processes means that users can easily understand and navigate the testing tools and methodologies, which helps in identifying issues and errors more efficiently and contributes to more effective outcomes. The use of model-agnostic techniques enhances usability by providing flexible and adaptable methods that do not require extensive customization for each new model, allowing for broader application and easier integration into various testing environments, thus making the process more efficient and user-friendly.

Additionally, three studies use model-specific techniques in security audits and compliance to enhance transparency by providing higher precision, clear understanding, transparent documentation, and building trust among users, auditors, and regulatory bodies.

6. Threats of Validity

This section outlines several potential threats to the study's validity, which should be considered when interpreting the findings and generalizing the results. The identified threats include:
One potential threat is the limited exploration of libraries, as the literature search only covered four databases (IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect). This narrow scope may have omitted relevant studies from other libraries or sources, potentially affecting the representativeness of the findings and limiting the understanding of the broader literature on the topic.

Another threat pertains to the refinement of the search key. Although efforts were made to develop a comprehensive search key, there is room for improvement. Limitations in the search key could have led to the exclusion of pertinent studies or the inclusion of irrelevant ones, impacting the comprehensiveness and accuracy of the study's findings.

Acknowledging these threats to validity is crucial for a thorough understanding of the study's limitations. Despite attempts to mitigate these threats through careful planning and adherence to predefined criteria, they remain important considerations. Future research should address these limitations by exploring a wider range of libraries and refining search keys through collaborative efforts. These steps can enhance the validity and generalizability of future studies in this domain.

7. Conclusion

This systematic mapping study has provided valuable insights into the integration and impact of XAI in software maintenance and testing. Our analysis of 18 primary papers highlights a growing interest in this area, with various XAI models being applied to improve the interpretability and effectiveness of software maintenance and testing tasks. The study identifies key tasks, goals, and forms of XAI implementation, contributing to a better understanding of current trends and gaps in the research.

Future work should focus on expanding the scope of studies to include diverse contexts and applications and developing standardized evaluation frameworks for XAI in software maintenance and testing. Additionally, longitudinal studies that track the impact of XAI over time would provide deeper insights into its long-term benefits and challenges. By addressing these areas, future research can further enhance the integration of XAI in software engineering, ultimately leading to more robust, transparent, and trustworthy AI-driven systems.

Data Availability
None

Conflict of Interest
The Author declares that there is no conflict of interest to report.

Funding Source
None

Authors' Contributions
Author-1 (Muna Alrazgan) provided project oversight, supporting the study's conception and ensuring its alignment with academic standards. Author-2 (Hala Almukhalfi) conducted the literature review and formulated the problem statement. Author-3 (Manal H. Alshahrani) developed the methodology and implemented the machine learning models. Author-4 (Mashael Aljohani) was responsible for data analysis and validation. Authors-5 and 6 (Nada Almohaimeed and Ruba Almuwayshir) interpreted the findings, synthesized the conclusions, and ensured compliance with research protocols. Author-6 (Zamzam Alhijji) led the manuscript development, overseeing the review and editing process. All authors actively participated in collaborative discussions and have reviewed and approved the final version.

Acknowledgements
None

References

[1] S. Cao et al., "A Systematic Literature Review on Explainability for Machine/Deep Learning-based Software Engineering Research," ArXiv Prepr. ArXiv240114617, **2024.**

[2] M. R. Islam, M. U. Ahmed, S. Barua, and S. Begum, "A Systematic Review of Explainable Artificial Intelligence in Terms of Different Application Domains and Tasks," Appl. Sci., Vol.**12**, No.**3**, **2022.** doi: 10.3390/app12031353.

[3] H. Hakkoum, I. Abnane, and A. Idri, "Interpretability in the medical field: A systematic mapping and review study," Appl. Soft Comput., Vol.**117**, p.108391, **2022.** doi: 10.1016/j.asoc.2021.108391.

[4] S. A. and S. R., "A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends," Decis. Anal. J., Vol.**7**, p.100230, **2023.** doi: 10.1016/j.dajour.2023.100230.

[5] I. Tiedekunta, "The Role of Explainable AI in the Research Field of AI Ethics – Systematic Mapping Study", **2023.**

[6] A. H. Mohammadkhani, N. S. Bommi, M. Daboussi, O. Sabnis, C. Tantithamthavorn, and H. Hemmati, "A Systematic Literature Review of Explainable AI for Software Engineering." arXiv, 12, **2023**. doi: 10.48550/arXiv.2302.06065.

[7] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Vol.**2**, **2007.**

[8] C. Pornprasit and C. K. Tantithamthavorn, "DeepLineDP: Towards a Deep Learning Approach for Line-Level Defect Prediction," IEEE Trans. Softw. Eng., Vol.**49**, No.**1**, pp.**84–98**, **2023**. doi: 10.1109/TSE.2022.3144348.

[9] S. Roy, G. Laberge, B. Roy, F. Khomh, A. Nikanjam, and S. Mondal, "Why Don't XAI Techniques Agree? Characterizing the Disagreements Between Post-hoc Explanations of Defect Predictions," in 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp.**444–448, 2022.** doi: 10.1109/ICSME55016.2022.00056.

[10] D. Rajapaksha, C. Tantithamthavorn, J. Jiarpakdee, C. Bergmeir, J. Grundy, and W. Buntine, "SQAPlanner: Generating Data-Informed Software Quality Improvement Plans," IEEE Trans. Softw. Eng., Vol.**48**, No.**8**, pp.**2814–2835, 2022.** doi: 10.1109/TSE.2021.3070559.

[11] J. Jiarpakdee, C. K. Tantithamthavorn, and J. Grundy, "Practitioners' Perceptions of the Goals and Visual Explanations of Defect Prediction Models," in 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), Madrid, Spain: IEEE, May, pp.**432–443, 2021**. doi: 10.1109/MSR52588.2021.00055.

[12] C. K. Tantithamthavorn and J. Jiarpakdee, "Explainable AI for Software Engineering," in 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia: IEEE, Nov., pp.**1–2, 2021.** doi: 10.1109/ASE51524.2021.9678580.

[13] S. Wattanakriengkrai, P. Thongtanunam, C. Tantithamthavorn, H. Hata, and K. Matsumoto, "Predicting Defective Lines Using a Model-Agnostic Technique," IEEE Trans. Softw. Eng., May, Vol.**48**, No.**5**, pp.**1480–1496, 2022.** doi: 10.1109/TSE.2020.3023177.

[14] C. Chai, G. Fan, H. Yu, Z. Huang, J. Ding, and Y. Guan, "Exploring better alternatives to size metrics for explainable software defect prediction," Softw. Qual. J., Dec. **2023.** doi: 10.1007/s11219-023-09656-y.

[15] C. Pornprasit, C. Tantithamthavorn, J. Jiarpakdee, M. Fu, and P. Thongtanunam, "PyExplainer: Explaining the Predictions of Just-In-Time Defect Models," in 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia: IEEE, Nov., pp.**407–418, 2021.** doi: 10.1109/ASE51524.2021.9678763.

[16] T. Zivkovic, B. Nikolic, V. Simic, D. Pamucar, and N. Bacanin, "Software defects prediction by metaheuristics tuned extreme gradient boosting and analysis based on Shapley Additive Explanations," Appl. Soft Comput., Oct., Vol.**146**, p. 110659, **2023**. doi: 10.1016/j.asoc.2023.110659.

[17] S. Bose, T. Barao, and X. Liu, "Explaining AI for Malware Detection: Analysis of Mechanisms of MalConv," 2020 Int. Jt. Conf. Neural Netw. IJCNN, Jul., pp.**1–8**, **2020**, doi: 10.1109/IJCNN48605.2020.9207322.

[18] N. AfzaliSeresht, Q. Liu, and Y. Miao, "An Explainable Intelligence Model for Security Event Analysis," in AI 2019: Advances in Artificial Intelligence, J. Liu and J. Bailey, Eds., Cham: Springer International Publishing, pp.**315–327, 2019.**

[19] A. Bumiller, O. Barais, N. Aillery, and G. Le Lan, "Towards a Better Understanding of Impersonation Risks," in 2022 15th International Conference on Security of Information and Networks (SIN), Nov., pp.**01–08, 2022.** doi: 10.1109/SIN56466.2022.9970540.

[20] R. R. Prasad, R. R. Rejimol Robinson, C. Thomas, and N. Balakrishnan, "Evaluation of Strategic Decision taken by Autonomous Agent using Explainable AI," in 2021 4th International Conference on Security and Privacy (ISEA-ISAP), Oct., pp.**1–8, 2021.** doi:10.1109/ISEA-ISAP54304.2021.9689715.

[21] K. Amarasinghe and M. Manic, "Explaining What a Neural Network has Learned: Toward Transparent Classification," in 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Jun., pp.**1–6, 2019.** doi:10.1109/FUZZ-

IEEE.2019.8858899.

[22] X. Ren, Z. Xing, X. Xia, D. Lo, X. Wang, and J. Grundy, "Neural network-based detection of self-admitted technical debt: From performance to explainability," ACM Trans. Softw. Eng. Methodol. TOSEM, Vol.**28**, No.**3**, pp.**1–45, 2019.**

[23] S. Rahman, U. A. Koana, and M. Nayebi, "Example driven code review explanation," presented at the Proceedings of the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp.**307–312, 2022.**

[24] G. Yenduri and T. R. Gadekallu, "Xai for maintainability prediction of software-defined networks," presented at the Proceedings of the 24th International Conference on Distributed Computing and Networking, pp.**402–406, 2023.**

[25] R. R. Prasad, R. R. Rejimol Robinson, C. Thomas, and N. Balakrishnan, "Evaluation of Strategic Decision taken by Autonomous Agent using Explainable AI," in 2021 4th International Conference on Security and Privacy (ISEA-ISAP), Dhanbad, India: IEEE, Oct., pp.**1–8, 2021.** doi: 10.1109/ISEA-ISAP54304.2021.9689715.