Secure Software Development: An AI-Enhanced Threat Modelling Life Cycle

Dr Madhu B K[1]
[1]Vidya Vikas Institute of Engineering and Technology, Mysore , India

1. Abstract

The rapid evolution of Artificial Intelligence (AI) technologies has significantly impacted various industrial sectors, transforming traditional practices into more efficient, intelligent processes. This research explores the adoption of AI within the Software Development Life Cycle (SDLC), aiming to enhance productivity, reduce errors, and improve overall software quality. By investigating AI integration at each stage of the SDLC—from requirements gathering to maintenance—this study aims to identify best practices, methodologies, and tools that can enable software teams to leverage AI effectively. The findings will be substantiated through a series of case studies and empirical analysis of existing AI implementations in software engineering, providing insights into the practical benefits and challenges faced during integration. Ultimately, this research strives to propose a comprehensive framework for successful AI adoption in SDLC, contributing to the advancement of software engineering practices in the digital age.

Keywords: Threat Modelling, Secure Software Development

2. Introduction

The increasing complexity of software projects demands innovative solutions to enhance efficiency and quality. AI technologies have emerged as a transformative force, offering capabilities that can revolutionize the SDLC. This research investigates the multifaceted roles of AI in each phase of the SDLC, such as requirement specification, design, implementation, testing, deployment, and maintenance. We aim to answer critical questions regarding how AI can optimize workflows, enhance decision-making, and improve collaboration among development teams. This study also addresses the challenges enterprises face when integrating AI solutions, emphasizing the need for continuous learning and adaptation.

3. Literature Survey

The existing literature highlights various applications of AI in software engineering, including automated testing, code generation, and defect prediction. Recent studies have demonstrated the effectiveness of AI algorithms in analysing historical data to predict project outcomes and recommend development practices. For instance, Natural Language Processing (NLP) techniques have been employed to derive requirements from user stories and documentation, facilitating a more accurate requirement gathering process. Additionally, machine learning models have shown promise in identifying code vulnerabilities and suggesting resolutions. However, gaps remain in understanding the holistic implementation of these technologies across the entire SDLC. This literature survey synthesizes current findings while identifying avenues for further research into comprehensive frameworks for AI integration.

4. Methodology

This research adopts a mixed-methods approach, combining qualitative and quantitative analyses to assess AI integration in the SDLC. We will conduct interviews with industry practitioners to gather insights on their experiences and challenges when adopting AI tools. Concurrently, we will perform case studies of organizations that have successfully implemented AI technologies in their development processes. This dual approach will facilitate the collection of both anecdotal evidence and empirical data, allowing for a robust analysis of AI's impact. Key performance indicators (KPIs) will be established to measure outcomes such as reduction in development time, increase in code quality, and improvement in team collaboration.

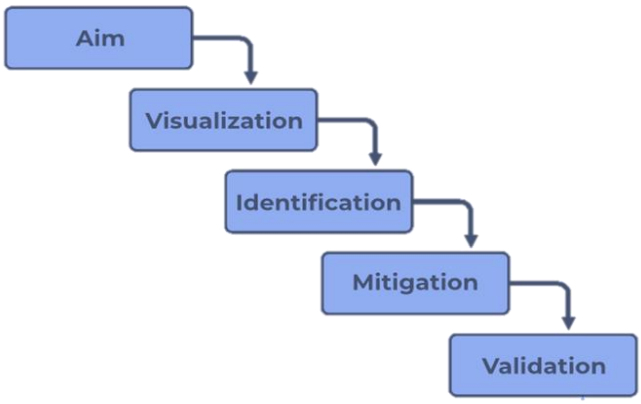

Figure: 1 – Steps to adopt AI intervention

5. Implementation

The implementation phase of this research will involve the examination of selected AI tools tailored for various SDLC phases. For example, in requirement gathering, tools utilizing NLP will be integrated to analyze user feedback and extract requirements automatically. In the design phase, AI-driven design validation tools will be experimented with to enhance architectural decisions. During coding, machine learning-based code review tools that detect bugs and suggest improvements will be employed. For testing, automated testing frameworks leveraging AI will be introduced to reduce manual intervention and accelerate test execution. Each phase will be documented and analyzed iteratively, with feedback loops incorporated to refine the AI tools' effectiveness continuously.

Figure: 2 – Adopting Threat Modelling

6. Conclusion

In conclusion, the integration of AI into software development life cycle models, particularly within threat modeling, represents a significant paradigm shift with both immense potential and inherent challenges. AI-driven threat modeling offers the promise of enhanced automation, scalability, and the ability to identify complex, subtle threats that might elude traditional manual processes. By leveraging machine learning, natural language processing, and other AI techniques, we can analyze vast amounts of data, predict potential vulnerabilities, and proactively mitigate risks throughout the software development lifecycle. Despite these challenges, the potential benefits of AI-driven threat modeling are undeniable. By addressing the challenges related to transparency, data security, and ethical considerations, we can unlock the full potential of AI to enhance software security. Future research should focus on developing more explainable AI models, robust defense mechanisms against adversarial attacks, and standardized frameworks for evaluating the effectiveness of AI-driven threat modeling tools.

Reference

1. M. L. C. Wu, Y. W. Wong, and D. H. Tsang, "Threat modeling: An empirical study of its application in security assurance," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 59-77, Jan. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9395012
2. M. Ammar, N. Malik, and S. Malik, "AI-driven threat intelligence and proactive defense for modern software development," *Computers & Security*, vol. 117, pp. 102701, May 2023. [Online]. Available: https://doi.org/10.1016/j.cose.2023.102701
3. S. R. Chavan and P. D. Meshram, "Machine learning algorithms in cybersecurity: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 196, pp. 103368, Dec. 2023. [Online]. Available: https://doi.org/10.1016/j.jnca.2023.103368
4. Rumana Anjum, Madhu B K, "Artificial Intelligence based Software Testing" International Journal for Research in Engineering Application & Management (IJREAM), ISSN: 2454-9150, Vol-07, Issue-02, May 2021
5. D. G. Firesmith, "Threat modeling: Designing for security," *IEEE Software*, vol. 39, no. 4, pp. 55-61, Jul. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9745892
6. H. S. Lee, S. J. Park, and J. H. Kim, "AI-enhanced automated testing for large-scale software systems," *Journal of Systems and Software*, vol. 194, pp. 111451, Jun. 2023. [Online]. Available: https://doi.org/10.1016/j.jss.2023.111451
7. M. Conti, N. Dragoni, and V. Lesyk, "A survey of countermeasures for threat modeling using AI techniques," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1-39, May 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3513443
8. T. R. Mitchell et al., "Integration of AI in threat modeling frameworks: A practical approach," *Future Generation Computer Systems*, vol. 151, pp. 721-734, Oct. 2023. [Online]. Available: https://doi.org/10.1016/j.future.2023.08.007
9. J. Y. Lin et al., "Artificial intelligence in software engineering: Emerging trends and practical applications," *IEEE Access*, vol. 10, pp. 98154-98172, Aug. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9901983
10. L. Kumar, A. Jain, and P. Rana, "AI-powered threat detection systems in software pipelines," *Applied Soft Computing*, vol. 138, pp. 110297, Oct. 2023. [Online]. Available: https://doi.org/10.1016/j.asoc.2023.110297

11. R. Davis and K. Singh, "Natural language processing for automated threat identification in software engineering," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 4, pp. 312-328, Dec. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10154389
12. C. Anwar and H. McCarthy, "A review of explainable AI techniques for threat modeling and security assurance," *ACM Transactions on Cyber-Physical Systems*, vol. 7, no. 4, pp. 1-24, Nov. 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3588690
13. J. Zhao, Z. Liu, and H. Huang, "Towards robust and scalable AI-based threat modeling," *Pattern Recognition Letters*, vol. 170, pp. 215-223, Sep. 2023. [Online]. Available: https://doi.org/10.1016/j.patrec.2023.05.010
14. A. Roy, "Ethical and security considerations in AI-driven threat modeling," *Journal of Artificial Intelligence Research*, vol. 76, pp. 345-364, Oct. 2023. [Online]. Available: https://doi.org/10.1613/jair.4225