



Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140

Volume 14 Issue 01, 2025

Impact of AI on Software Development and Testing

Nivedi Sapate¹, Shweta Nirkulwar², Laxmi Gathe³, Sejal Shahu⁴, Mr. Rupesh. Bangre⁵

¹⁻⁵MCA Department Suryodaya College of Engineering & Technology, Nagpur.

sapatenivedita@gmail.com¹,

nirkulwarshweta@gmail.com²,

laxmigate6519@gmail.com³,

shahusejal75@gmail.com⁴, rupesh.rpb@gmail.com⁵,

Peer Review Information	Abstract
<i>Submission: 11 Feb 2025</i> <i>Revision: 20 Mar 2025</i> <i>Acceptance: 22 April 2025</i> Keywords AI Software Development Automation Bug Detection Code Writing	Artificial Intelligence (AI) is changing the way software is developed and tested. AI tools help developers write code faster, find errors, and improve software quality. In testing, AI can automatically check for bugs, suggest fixes, and make sure the software works correctly. These advancements save time, reduce mistakes, and improve efficiency. However, there are also challenges, such as the need for skilled professionals and concerns about data privacy. This paper explores how AI is used in software development and testing, its benefits, challenges, and future possibilities. While AI presents numerous benefits, challenges such as data privacy, bias, and the need for skilled professionals remain. This study examines current trends, real-world applications, and future prospects of AI in software development and testing, providing insights into its transformative potential and associated risks.

INTRODUCTION

Artificial Intelligence (AI) is changing the way software is developed and tested through increased efficiency, precision, and automation. Classical software development involves massive human intervention in coding, debugging, and testing, which can be time-consuming and error-prone. But AI tools are revolutionizing these processes by automating the code generation, detecting and correcting bugs, and enhancing software testing. Artificial intelligence-based tools like GitHub Copilot, Tabnine, and OpenAI Codex aid programmers with suggestions for pertinent code pieces, improved performance, and minimizing syntax errors. These systems sift through large databases of information, identify

patterns, and improve the quality of software with little manual input, which takes much less time to develop.

Debugging and detecting bugs have also undergone remarkable progress with AI-based tools like DeepCode, Codiga, and SapFix of Facebook.[1]

These utilities scan source code, forecast possible vulnerabilities, and recommend repairs, mitigating the likelihood of security exploits and application failure. AI can also learn from past patterns of debugging to spot repeated bugs, maximizing the efficiency and proactivity of debugging. In the same manner, AI-based automated software testing tools like Selenium, Test.ai, and Applitools automate testing with the

creation of test cases, UI inconsistency detection, and regression test optimization. This enables ongoing testing, guaranteeing software stability and performance with reduced manual effort. Predictive analytics is another significant contribution of AI to software engineering. AI models examine past data, defect patterns, and performance metrics to predict software failures prior to their occurrence.

By identifying risks ahead of time, developers can resolve potential problems early in the development cycle, enhancing software reliability and user experience. Further, Natural Language Processing (NLP) also supports requirement analysis through the process of translating user needs into organized technical specifications. AI-based NLP tools inspect project documentation, create test cases, and flag inconsistencies, streamlining team collaboration and minimizing errors. While useful, AI application in software development also has some drawbacks. Concerns regarding data privacy and security exist since AI tools are dependent on large data sets that are often sensitive. Another concern is bias in AI models since AI models are trained from past data, which at times results in unequal or incorrect forecasts.

Moreover, implementing AI in software development involves expertise in machine learning, AI algorithms, and software development, and this could be a concern for organizations with limited talent to perform these activities. Additionally, while AI enhances efficiency, sole dependence on AI-based decisions will result in unforeseen errors, and thus human verification is required to authenticate AI-produced outputs. The future of AI in software development is bright, with ongoing developments in machine learning, deep learning, and automation. AI is likely to have a greater role in self-healing software, where it can detect and repair problems independently in real time. The combination of AI with DevOps and cloud computing will further automate software deployment, monitoring, and maintenance, making the development process more efficient and cost-saving. Despite certain challenges, AI continues to transform software engineering, making it intelligent, quick, and trustworthy.



Fig:1 The impact of AI on software development

LITERATURE REVIEW

Celia Dolores Benitez and Montes Serrano (2023) proposed through an analysis of AI's methods, historical background, and real-world applications, this study investigates the tremendous effects of AI on software engineering. It explores AI-assisted software development lifecycle, AI-driven project management, and AI in software testing. It also emphasizes the use of AI in software measurement, software engineering procedures generally, and software as medical devices (SaMD)[1].

Krishiv Garg (2023) examines how artificial intelligence (AI) affects software development, looking at the potential and problems the sector faces. The study explores the ways in which artificial intelligence (AI) technologies, such machine learning and natural language processing, are transforming requirements engineering, code generation, testing, and maintenance, among other facets of the software development lifecycle [1].

METHODOLOGY

Artificial Intelligence (AI) is revolutionizing software development and testing by automating tasks, improving accuracy, and enhancing overall efficiency. To understand AI's impact in this field, data is collected from various sources, including scholarly publications, business reports, and real-world case studies. Research papers and academic journals provide insights into AI's theoretical and practical advancements, while business reports from industry leaders highlight trends, adoption rates, and return on investment. Case studies also demonstrate how businesses effectively implement AI in software development processes. Further, surveys and interviews with software developers, testers, and project managers provide firsthand experience about the advantages and challenges in adopting AI[3].

AI-based tools are significantly contributing to contemporary software engineering. Clever

testing tools, like Test.ai and Applitools, make software testing autonomous, minimizing human effort and maximizing test coverage. Code generators such as GitHub Copilot and OpenAI Codex automate code writing with fewer errors and at a quicker pace. Concurrently, bug detection tools that utilize AI, including DeepCode and Facebook's SapFix, identify and address vulnerabilities early on in the development process. Each of these technologies is measured using important efficiency standards, such as how much it can speed up development, improve software quality, and reduce the need for human intervention. The use of AI greatly enhances productivity by minimizing coding and testing time while providing greater accuracy in defect detection.[3]

To evaluate the strengths and weaknesses of AI, comparing AI-based software development and testing with conventional methods is necessary. AI tools provide greater scalability by automating big-bang testing and development, while manual methods involve huge human effort and resources. Accuracy-wise, AI minimizes human mistakes but can potentially create false positives or negatives during testing, requiring ongoing scrutiny and improvement. Although the implementation of AI comes at a cost upfront, it decreases long-term expenditure by minimizing the need for manual effort in the long run. Furthermore, AI-driven development maximizes efficiency by speeding up the software development process and streamlining debugging procedures. Yet, there remains a place for conventional techniques when human experience and intuition are called upon to address intricate problems.[4]

APPLICATIONS

Artificial Intelligence (AI) is revolutionizing software development and testing by way of numerous automated processes, enhancing efficiency, accuracy, and overall software quality. Automated code generation is one of the most important areas where AI has a prominent role to play. AI-based tools such as GitHub Copilot and Tabnine help developers by providing code suggestions, boilerplate code generation, syntax error detection, and overall code structure enhancement. These features cut down development time and make coding more efficient. Another significant use is bug detection and repair, where AI-powered debugging tools like DeepCode and Codiga scan source code for bugs and recommend relevant fixes. Machine learning algorithms predict possible vulnerabilities,

enabling developers to fix issues ahead of time and streamline debugging operations. In the same vein, automated testing uses AI-powered testing tools like Selenium, Test.ai, and Applitools to run test cases automatically. AI enhances test coverage, finds UI inconsistencies, and increases regression testing, lowering the amount of manual effort to validate software.[4]

AI also plays a role in predictive analytics for software quality, allowing developers to predict software failures and performance issues prior to deployment. Machine learning algorithms examine historical defects and testing data to optimize testing strategies and enhance software reliability. Natural Language Processing (NLP) for requirement analysis also allows AI to help translate user requirements into technical specifications. NLP-based tools scan documentation, identify key requirements, and even create test cases automatically, making the software development lifecycle more efficient [5].

By incorporating AI into software development and testing across different aspects, organizations can increase productivity levels, minimize errors, and streamline the overall software development process. Although AI-based methodologies offer several advantages, continuous improvement and human intervention are necessary to guarantee the accuracy and reliability of AI-produced solutions.[6]

FUTURE SCOPE

Fully Automated Software Development: AI could advance to the point where entire applications are developed with minimal human intervention. AI-driven coding assistants may evolve into autonomous systems that can design, develop, and maintain software [5].

Enhanced AI-Driven Testing: AI will further improve automated testing by generating more efficient test cases, reducing human involvement in testing processes. AI models could self-adapt to software changes, making testing faster and more reliable [5].

Intelligent Bug Prediction and Prevention: Future AI systems may predict software defects before they occur, reducing debugging time. Machine learning algorithms will analyze code patterns and automatically suggest fixes before deployment.

AI-Powered Self-Healing Software: AI-enabled applications will be able to detect issues in real time and fix them without human intervention. This will lead to more stable and reliable software systems, reducing maintenance cost [6].

AI-Enhanced Cybersecurity in Development: AI will play a key role in identifying and mitigating security vulnerabilities in software code. Future tools will integrate AI-driven threat detection to improve software security.[5]

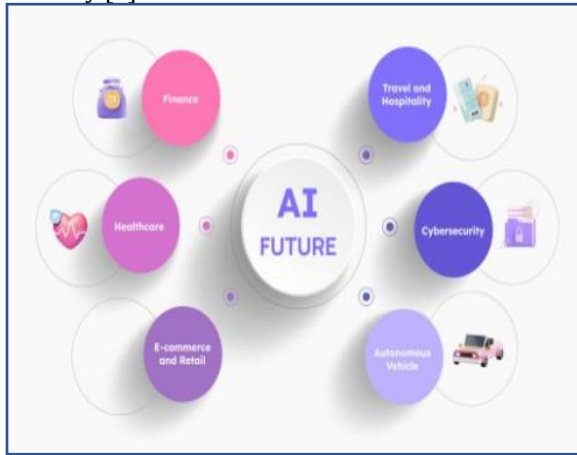


Fig2: AI and future of software development

CONCLUSION

Artificial Intelligence (AI) is transforming software development and testing by enhancing automation, accuracy, and efficiency. AI-powered tools help developers write code faster, detect and fix bugs, and automate software testing. This has significantly improved the speed and quality of software development while reducing human effort and errors.

In conclusion, AI is revolutionizing the way software is developed and tested. While there are challenges to overcome, the potential of AI in this field is Immense, promising a future where software engineering is faster, smarter, and more automated than ever before.

However, challenges like AI bias, data privacy concerns, and the need for skilled professionals remain. Despite these issues, AI continues to transform the industry, making software development faster, smarter, and more efficient.[6]

References

Celia Dolores Benitez¹ and Montes Serrano(2023). The Integration and Impact of Artificial Intelligence in Software Engineering. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 3, Issue 2.

Krishiv Garg (2023). Impact of Artificial Intelligence on software development: Challenges and Opportunities. International Journal of Software & Hardware Research in Engineering (IJSHRE) ISSN-2347-4890.

Saeid, H. (2020). Revolutionizing Software Engineering: Leveraging AI for Enhanced Development Lifecycle. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences, 8(1).

Prajapati, S., Prajapati, B., Vegad, S., & Gohil, G. (2022). Artificial Intelligence and Software Engineering: Status, Future Trend, and Its Interaction. International Journal for Research in Applied Science and Engineering Technology, 10(3), 1411–1417.

Saeid, H. (2021). AI-driven Project Management in Software Engineering. International Journal of Scientific Development and Research, 6(1), 299–308.

Bosch, J., & Olsson, H. H. (2019). Towards continuous validation of AI systems in software engineering. Journal of Software: Evolution and Process, 31(12), e2240.

Assessing the Efficiency of Integrating BIM and Blockchain to Improve Information Management for Mars Buildings: A SWOT-AHP Analysis

by Amirhossein Javaherikhah
1,*ORCID, Mercedes Valiente Lopez 1ORCID and Hadi