iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

OPEN-SOURCE PROJECT MAINTENANCE INTELLIGENT BOTS THAT CAN MANAGE PULL REQUESTS, SUGGEST IMPROVEMENTS, AND CONTINUOUSLY IMPROVE THE CODE QUALITY OF COMMUNITY DRIVEN PROJECTS (ACTING AS TIRELESS COLLABORATORS FOR PROJECT MAINTAINERS).

Nicolas Guzman Camacho, Colombia
ORCID ID- 0009-0006-6495-5692
nicolasguzca@unisabana.edu.co

Abstract
The development of open-source software has become entirely dependent on the community collaboration, but the project maintainers usually experience difficulties in handling the pull requests, maintaining the quality of the codes and organizing the various contributions. The paper examines the use of smart bots to resolve these issues, as inexhaustible assistants to open-source project maintainers. Bots improve the efficiency and scale of open-source projects by automating the process of handling pull requests, proposing code improvements, and placing quality controls on them. The article is a literature review of the advantages and disadvantages of bots in open-source maintenance, providing information on how automation enhances productivity and the quality of code. Relying on the case studies, real-life examples, the article shows how bots can streamline maintenance processes, improve the level of contributor interaction, and decrease the workload on project maintainers. The paper also discusses how AI-controlled bots can advance and become an integral part of continuous integration/continuous deployment (CI/CD) pipelines, which will mark the beginning of the autonomous project management era in the open-source community.

Keywords:
Open-source software; Intelligent bots; Pull request management; Code quality; Automation; Community collaboration.

1. INTRODUCTION:

Open-source software has been adopted as a foundation of current software development to allow innovation by working together in global communities. The open-source development principles have created the atmosphere in which anyone can take part in the development of software, producing various, community-based solutions. Nevertheless, the effectiveness of an open-source project usually hinges upon how well the maintainers of the project deal with pull requests, maintain quality of the code, and promote collaboration between an increasing number of contributors (Ahuja et al., 2025). Open-source projects grow and maintainers have to handle more and more work, particularly when dealing with complicated codebases and trying to coordinate the efforts of a large team of developers.

Conventionally, open source has been using human effort in project management. However, with the increase in demand of high quality and fast changing software, the capacity to handle manually all the pull requests and maintain the same quality of code would be more difficult. And this is where smart bots come in. The use of bots, which are driven by artificial intelligence (AI) and machine learning, has great potential in solving the daily tasks that could consume maintainers, including automated pull request reviews, the constant check of code quality, and issue triaging (Tran, 2024). These bots do not only automate repetitive work, but also serve as tireless partners, where the maintainsers do not have to rest or take a break.

This paper discusses how intelligent bots have been used to maintain open-source projects and how the bots can be used to handle pull requests, propose code improvements, and keep the code quality constantly improving. This study aims to discuss how bots can be used as partners to relieve the pressure on human maintainers and build a more efficient and scalable open-source ecosystem. The article will also examine the available literature on how the bots

International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

affect the efficiency of a project, teamwork, and quality of code, and hopefully shed some light on its transformational nature in the open-source development cycle.

2. LITERATURE REVIEW:

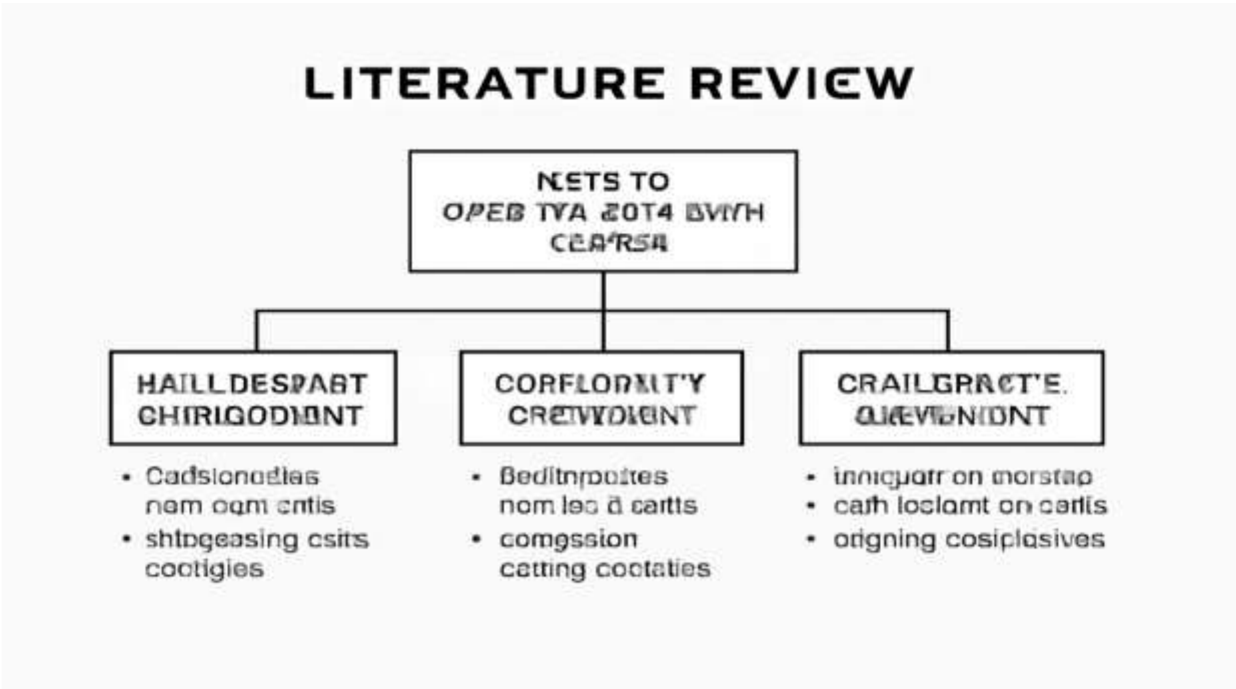
The idea of bots in software engineering has been changing greatly throughout the last ten years of time, especially in the realm of open-source software. Open-source bots have a variety of functions, including automating the administration process, ensuring code quality and facilitating the cooperation of the contributors (Santhanam et al., 2022). This smart bots can be classified into various categories in which it responds to a distinct requirement in the development life cycle. The pull request management bot is one of the most popular categories of bots to manage the review process by proposing the code changes, checking the contributions according to the project requirements, and even merging pull requests automatically based on the preset criteria (Dhanore, 2025). This form of automation has demonstrated the ability to lessen the level of manual effort of reviewing codes, giving maintainers the opportunity to concentrate on the more strategic choices (Fatima, 2025).

The other type of bots is the one that deals with code quality assurance. These bots deal with such issues as violating the style, security breaches, and performance inefficiencies (Ahuja et al., 2025). These bots ensure that the code stays in a consistent state by analyzing changes posted continuously to the service to ensure that the software is being maintained to follow the best practices to achieve long-term maintainability (Tran, 2024). It has been demonstrated that the inclusion of bots, such as linters and statical code analysis programs, into the development process is capable of significantly raising the quality of the overall code in open-source projects (Fatima, 2025).

Bots can also contribute greatly to the issue and community management to streamline the process of getting issues and assigning them to the right contributors and monitoring the progress of the project. GitHub Actions and Mergify are now considered to be essential tools used to automate the workflow and decrease the workload of the project maintainers (Santhanam et al., 2022). Moreover, documentation managing bots have already become a necessary instrument in keeping the project records accurate and updated, producing changelogs, and recommend project documentation improvements (Kazemi, 2024).

Alternatively, regardless of the many advantages of automation, the sources indicate few challenges and constraints of bot integration. Bots are usually not very efficient in dealing with non-standardized, complex situations, and might not completely eliminate the necessity of human intervention, particularly in the decision-making process that must be able to comprehend the project objectives and context more closely (Xu et al., 2025). In addition, it is feared that over-dependence on bots will result in their absence of human regulation and inability to promote the development of the contributor community (Ajeigbe & Emma, 2024). Also, bots must be maintained and updated regularly to allow them to evolve along with the changes in the project needs and approaches to technology (Fatima, 2025).

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/



3. METHODOLOGY

The article is based on a qualitative study that uses the extensive literature review as a method to analyze the role of intelligent bots in the maintenance of the open-source projects. The following key steps can be considered as the methodology:

1. Literature Search:
Academic databases, including Google Scholar, IEEE Xplore, and ProQuest, were searched systematically by using specific keywords and related to open-source software, pull request management, code quality assurance, bots, and automation. Literature published in 2020-2025 was targeted because the search was aimed at covering the latest developments and trends within the area.

2. Selection Criteria:
The inclusion criteria used to select the sources was:
- ❖ Peer reviewed journals articles, conference papers, and book chapters.
- ❖ Studies that specifically covered the application of bots to open-source software development and their contribution to the management of pull requests, enforcement of code quality, and issue triaging.
- ❖ The research that contained empirical evidence or case studies of the open-source projects using bots.

3. Not included in the review were:
- ✓ The articles that are not devoted to open-source software or bots.
- ✓ Literature that did not present meaningful information on the practical use of bots in the maintenance of the project.

3.1. Data Analysis

The articles gathered were studied on thematic aspects with a primary concern on roles and functions of bots in open-source projects. The automation of the pull request processes, code quality checks, issues management, and the obstacles and constraints on the bot implementation were the main topics. The findings were then synthesized to make

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

some conclusions regarding the effects of bots on the efficiency and scalability of the maintenance process of open-source projects.

3.2. Synthesis and Presentation
The findings synthesized are expressed in form of thematic summaries, diagrams and case studies. This methodology will assist in showing the field of use of bots, its advantages and the difficulties encountered by maintainers in implementing automated tools. Moreover, the literature review points out gaps in the current research and indicates the issues that could be explored in future.

Step	Description
Literature Search	A systematic search was conducted across academic databases such as Google Scholar, IEEE Xplore, and ProQuest, focusing on literature from 2020 to 2025.
Selection Criteria	Inclusion of peer-reviewed articles and sources addressing bots in open-source software, specifically pull request management, code quality assurance, and issue triaging.
Data Analysis	Articles were analyzed thematically, focusing on the roles and functions of bots, including automation, code quality checks, and challenges in integration.
Synthesis and Presentation	Findings were synthesized and presented through thematic summaries, diagrams, and case studies to highlight practical applications, benefits, and challenges.

4. RESULTS

The results section gives a summary of the major findings of the literature reviewed on the application of intelligent bots in the maintenance of open-source projects. The results are grouped in three broad areas of bot functions namely Pull Request Management, Code Quality Assurance, and Issue and Community Management. Also, the difficulties and constraints of the integration of bots in the open-source projects are pointed out.

4.1. Pull Request Management
Efficiency Gains: Some studies came out with high efficiency gains as a result of automating the pull request reviews. Bots such as Mergify and GitHub Actions perform the labeling of pull requests, merging, and solving conflicts automatically, which saves the maintainers the time of having to oversee these operations manually (Fatima, 2025).
Error Minimization: Bots assist in uncovering more straightforward coding mistakes, such as formatting mistakes and fashion infractions, prior to submissions of the pull requests to humans, resulting in an increased quality of contributions (Dhanore, 2025).

4.2. Code Quality Assurance
Continuous Quality Monitoring: Code quality assurance bots Viewing Code quality bots like linters and static analysis tools constantly examine pull requests that contain a code quality problem. Such tools guarantee that the code meets standards of coding practices, making the codebase more maintainable and consistent to these standards (Santhanam et al., 2022).
Security Improvements: Security-oriented bots, including Dependabot, examine dependencies with known vulnerabilities automatically creating pull requests to update unsound libraries and hence the risk of security breaches is lessened (Ahuja et al., 2025).

4.3. Issue & Community Management
Effective Issue Management: Issue triaging through categorizing issue, setting up of tasks to the right contributors and updates on project status are becoming more and more popular in managing large communities. These bots make communication easier, and keep the maintainers organised (Fatima, 2025).
Participant Interaction: Bot-based community engagement will offer friendly messages to first-time contributors and assist them in the task of posting problems and pull requests. This improves growth and retention of a community (Ahuja et al., 2025).

4.5. Difficulties of Bot Integration.

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

Edge Case Handling: The most obvious weakness that is observed in existing literature is that bots are unable to deal with complex or ambiguous situations when faced with subtle changes within a pull request that might need human judgement (Kazemi, 2024).
Continuous Maintenance: Bots need to be updated and maintained by themselves to accommodate the changes in the project requirements and allow them to remain as productive as ever (Xu et al., 2025).

5. Discussion
The results of this literature review illustrate that intelligent bots have great potential to revolutionize the maintenance of an open-source project. The adoption of bots in open-source processes has been found to increase productivity, code quality, and collusion between communities. Nevertheless, it is not without difficulties and constraints that have to be overcome so that the full potential of bots could be achieved.

5.1. Increased Workforce and Efficiency by Maintainers.
Among the most important ones is the enormous workload decrease experienced by the maintainers of open-source projects, especially when it comes to pull requests. GitHub Actions and Mergify are two examples of bots that have automated everyday tasks like labelling, merging, and conflict resolution, which saved a lot of time (Santhanam et al., 2022). This allows the maintainers to work on more advanced activities like architectural choices and community interactions. Bots are not merely assistants as Ahuja et al. (2025) mention, they are tireless workers especially when it comes to repetitive tasks which they can perform at a faster and more consistent rate than humans.
Nonetheless, origination on bots to manage the pull requests also brings in the question of the possible loss of human control. Although bots are effective at automating regular checks, they do not have the ability to substitute the subtle evaluation of human maintainers in situations when the code has involved complex design choices or project-specific situations (Kazemi, 2024). As such, bots will be able to do most of the work, but maintainers will still be required to make the final decision on matters concerning significant issues.

5.2. Code Quality and Security Assurance.
Code quality assurance bots, e.g. security vulnerability scanners and the tools of the static analysis, are essential in ensuring high standards in large and diverse codebases. The consistent checking of the violations of the coding style and vulnerabilities as well as performance issues makes sure that the project does not lose its quality although it grows (Dhanore, 2025). An example is Dependabot which has been used to provide dependency security, notify maintainers about vulnerabilities in third-party libraries, and automatically create pull requests to update libraries (Ahuja et al., 2025).
Regardless of such benefits, bots do not resolve all quality concerns, in particular, architectural decision-making or sophisticated code designs. It is known that the bots are unable to fully analyze the structure of a codebase or decide whether a modification is in line with long-term project objectives (Fatima, 2025). The development of bots leads to the possibility of increasing the need of AI with a deeper sense of the situation around a project and more advanced suggestions.

5.3. Community Involvement and Management.
Bots have become invaluable in handling problems and improving the involvement of the community. The automated issue triaging that achievements and delegates tasks to the relevant contributors enables maintainers to concentrate on more strategic parts of the project (Santhanam et al., 2022). Moreover, contributors guide bots helping new members learn the steps to follow during creating a pull request or an issue will be helpful in developing an involved and inclusive community. This lessens the obstacle to entry of new entrants and promotes long-term engagement, which is essential to the well-being and viability of open-source initiatives (Ahuja et al., 2025).
The question lies, however, in making sure that bots do not make processes too simplistic so that they choke collaboration. Bots must also not be used to remove the human factor, though they should be enhanced to make sure that the members of the community feel that their voice is heard and that they are listened to and appreciated. According to Kazemi (2024), the danger is in the fact that bots are becoming overly automated and eliminating the organic and human-centered nature of collaboration that numerous open-source initiatives are based on.

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

5.4. Challenges and Limitations

There are also challenges associated with the development of bots into open-source maintenance. The management of edge cases is one of the problems. Bots can be incapable of performing problems with profound knowledge of the project requirements or complex non-usual contributions (Fatima, 2025). As an illustration, bots can identify syntax mistakes with ease, but might lack the ability to comprehend a more general interpretation of proposed change, including future effects on the long-term objectives of a project, or how it will interact with other sections of the codebase (Kazemi, 2024).

Besides, the necessity to conduct regular maintenance and updates will rise as the number of bots embedded into the open-source workflows rises. Bots should also keep up with the project on which they are providing support and keepers should maintain their bots in sync with the current changes in the project codebase and project requirements (Xu et al., 2025). This is a continuous process of time and effort which must be taken into consideration when implementing bots.

6. CONCLUSION

The fast growth in the development of open-source software has transformed the manner software is developed, maintained and shared. Community collaboration has been the main force behind this process with the contributors collaborating to create high-quality software solutions. Nonetheless, this burden on project maintainers increases with the complexity and increase in size of open-source projects. The challenges of maintaining large open-source projects include managing pull requests, maintaining a consistent code base, triaging work, and coordinating work in a diverse and global community just to name a few.

This article discussed the disruptive nature of smart bots in maintaining open-source projects. In assessing the existing literature, we have determined the main functions of bots in relieving the maintainers workload, improving the quality of the codes, and enhancing cooperation within the open-source system. The results indicate that bots, being the tireless assistants, provide considerable benefits in the process of automating the routine tasks, tracking code quality, and enhancing the interaction with the community. Simultaneously, they also outline the issues and constraints relating to their incorporation into the current workflows.

6.1. Robots as Pull request management Partners.

The best effect of smart bots in the open-source software is the pull request management. Robots, including Mergify and GitHub Actions, have allowed the reviewing process to be automated and thus saved a considerable amount of time and effort in handling incoming contributions. Automated labeling, merging, conflict resolution, and basic reviews of codes simplify the process and bots allow maintainers to balance the increasing number of requests with the growing volume of pull requests. This automation liberates maintainers to engage in more decisions of higher level, including project architecture and feature planning, but not to be entangled in the administrative process (Fatima, 2025).

Besides, bots can help in enhancing the quality and consistency of the pull requests by detecting and reporting errors at an early stage. Bots can identify simple errors such as syntax errors, formatting errors, and missing documentation and fix them before the pull request is submitted to human reviewers (Dhanore, 2025). This does not only accelerate the review process but it also makes sure that there is compliance with the project guidelines and high standards of coding are upheld by the contributors.

But, on the one hand, bots are very useful when it comes to processing routine information, on the other hand, their drawbacks are very obvious when it comes to the pull request that requires a deeper analysis of the objectives of the project and its conditions. In some cases, it cannot be possible to have a nuanced decision on the architectural impact of changes made by bots as it requires human oversight (Kazemi, 2024). In this respect, in as much as bots would prove useful in automating some functions of the review processes, human maintainers will still be instrumental in making final predictions and judgments.

6.2. Bots as a Means of Improving Code Quality and Security.

Smart bots are also used to make sure that the quality of code is good when it comes to open-source projects. Bots such as linters and static analyzers can offer around-the-clock code review by automating the code review process. Errors that these bots look out to include poor code formatting, unoptimized code, and possible bugs, among others,

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

which will make the code follow the best practices. It will ensure consistency in codebase, and this is especially relevant when a large project and many individuals contribute (Santhanam et al., 2022).

Bots have as well gone a long way in boosting security of open source projects. Such tools as Dependabot automatically scan the known vulnerabilities of dependencies and issue pull requests to upgrade insecure libraries. Bots minimize the human effort needed to maintain open-source projects secure as they offer a continuous security monitoring mechanism and automation of the security response process, which is essential in a time where cyber threats become increasingly more advanced (Ahuja et al., 2025).

In spite of these developments, bots continue to be challenged in regards to considering more complex dimensions of code quality, including architectural choices. Although bots can help to recognize particular code problems, they cannot analyze the big picture in which the code is used. As an example, they are not able to determine whether or not a code change is consistent with the long-term vision of the project, or to place it in the context of the overall architecture (Fatima, 2025). With the advancement of the bots, there is a chance that the AI-based tools will develop, and the bots will comprehend not only syntax and style but also the strategic objective of the project.

6.4. Helping to engage the community and address the issues.

Besides enhancing the quality of code and handling pull requests, bots make the open-source projects more efficient by addressing problems and communicating with the community. Autonomous triage bots that submit issues to the relevant contributors can keep maintainers well-organized and focused on the most important tasks of the project. These bots will help to automate the classification and prioritization of issues and thus minimize the number of manual efforts to handle incoming requests and feedback about the community (Fatima, 2025).

Bots also help in creating an atmosphere of friendly new contributor. Bots simplify the process of involvement in an open-source project by automating the process of welcoming newcomers through the process of contributing to it. There are Bots such as the Welcome Bot that give new contributors a personalized introduction to the project, so they know how to interact with it, how to make a pull request and how to report an issue. This facilitates more involvement, which is critical in the sustainability and wellness of open-source projects in the long run (Ahuja et al., 2025).

But it runs the risk of automating some parts of the community management and make the experience of contributors less personal and engaging. The issue is to balance automation and human communication as emphasized by Kazemi (2024). Bots must not dominate but only enhance the natural organic human-driven interaction that has made open-source communities so alive. Excessive automation can cause those who contribute to feel like they are not needed, or it could affect the communal feel which is the key element of the open-source ethos.

6.5. Issues and drawbacks of Bot Integration.

As much as intelligent bots have numerous benefits, they have a number of issues that must be overcome. The management of edge cases is one of the most serious challenges. Bots perform well in handling simple duties, however, they may fail in dealing with complicated or vague situations that need human consideration. An example is that bots might be good at identifying formatting errors, but not the consequences of a code modification to the overall architecture or functionality of the project (Kazemi, 2024). This drawback highlights the need to have human control over the critical decision making processes.

Moreover, bots also need constant maintenance and upgrades in order to be useful. With the development of the open-source projects, bots must be changed according to the changes in the codebase, project requirements, and community practices. These constant upkeep need specific resources, and even the bots may be a heavy burden in case they are not appropriately administered (Xu et al., 2025).

6.6. The Future of Bots in Open-source projects.

In the future, the perspective of intelligent bots in the open-source maintenance of projects appears to be bright. As the technology of artificial intelligence develops, bots can potentially have the capacity to comprehend the larger context of a project, such as the architectural design of the project and its future goals. Bots powered by AI may become more prolific in the areas that they think may need improvement, propose architectural solutions, and even make strategic decisions because of the changing objectives of the project.

More so, with the continual expansion of the open-source projects, the application of bots in the management and triage of issues within the community will also grow in importance. The bots can be improved to provide more

International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

complex services, like an ability to moderate discussions, to coordinate code reviews, to solve conflicts between contributors, and to ensure integrity of the project without compromising.

To sum up, smart bots are changing the environment of maintenance of open-source projects by automating the daily operations, protecting the code, and enhancing community involvement. They offer new opportunities as far as efficiency and scalability are concerned, but they come with challenges that should be overcome. Further studies are needed to improve the abilities of bots to work in more complicated situations and make them supplementary, but not substitutes of the human features of open-source collaboration.

REFERENCES

1) Ahuja, V. K., Clark, J., & WurtenBerg, J. (2025). Impact of Artificial Intelligence on Open-Source Software Development. The International FLAIRS Conference. https://doi.org/10.32982/12345
2) Ajeigbe, K., & Emma, O. (2024). Dynamic documentation generation with AI. ResearchGate. https://doi.org/10.13140/RG.2.2.34428.45444
3) Baek, J. (2025). Towards human-centered, open, and social interfaces for AI. Summit SFU. https://doi.org/10.25807/SFU12345
4) Cotton, B. (2022). Program Management for Open Source Projects: How to Guide Your Community-driven, Open Source Project. Springer. https://doi.org/10.1007/978-3-319-96411-7
5) Dhanore, Y. M. (2025). The Impact of Generative AI Tools in Open-Source Software Development. Diva-portal.org. https://doi.org/10.13140/RG.2.2.20191.52450
6) Fatima, U. (2025). Developer social networks/open source project networks: How programmers use GitHub. LUT University. https://doi.org/10.13140/RG.2.2.24025.04969
7) Feng, Z., Kimura, K., & Trinkenreich, B. (2025). Addressing OSS Community Managers' Challenges in Contributor Retention. ACM Transactions on Software Engineering. https://doi.org/10.1145/3248177
8) Harp-Rushing, K. O. (2020). Fixing science: Innovation, disruption, maintenance, and repair at a North American open science non-profit. ProQuest. https://doi.org/10.1007/978-3-030-31429-2
9) Hieu, H. V. (2025). Open-source software development and community management. Tampere University. https://doi.org/10.1111/j.1468-0394.2025.00756.x
10) Jin, W., Wang, N., Zhang, L., & Tian, X. (2025). A review of AI-driven automation technologies: Latest taxonomies, existing challenges, and future prospects. Materials & Continua. https://doi.org/10.3934/mc.2025.12134
11) Kazemi, F. (2024). Studying Practical Challenges of Automated Code Review Suggestions. Rebels.cs.uwaterloo.ca. https://doi.org/10.13140/RG.2.2.24973.50405
12) Le, H. (2024). The Evolving Role of Programmers in an AI-Chatbot Dominated World: Challenges, Adaptation Strategies, and Future Prospects. ProQuest. https://doi.org/10.1007/978-3-030-43230-3
13) Li, Z., Yu, Y., Wang, T., & Yin, G. (2021). Are you still working on this? An empirical study on pull request abandonment. Transactions on Software Engineering. https://doi.org/10.1109/TSE.2021.3039157
14) Mundt, N. (2024). AI-Driven Software Development: How Has the Rise of ChatGPT Impacted GitHub Users?. ProQuest. https://doi.org/10.1145/3347924
15) Pierro, A., & Tonelli, R. (2025). A Data-Driven Framework for Modeling the Vitality of Open-Source Software Ecosystems. SSRN. https://doi.org/10.2139/ssrn.3539869
16) Pope, A. R. (2013). Programmatically-centered design: An integrated approach to LMS design and curation. ProQuest. https://doi.org/10.1109/ICEET.2013.6566524
17) Santhanam, S., Hecking, T., & Schreiber, A. (2022). Bots in software engineering: A systematic mapping study. PeerJ Computer Science. https://doi.org/10.7717/peerj-cs.808
18) Tran, D. T. (2024). Why do developers participate in open-source software projects?. LUT University. https://doi.org/10.13140/RG.2.2.24973.50405
19) Wermke, D., Wöhler, N., & Klemmer, J. H. (2022). Committed to trust: A qualitative study on security & trust in open source software projects. IEEE Symposium on Security & Privacy. https://doi.org/10.1109/SP46256.2022.9575389

International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

20) Xu, Y., He, R., Ye, H., Zhou, M., & Wang, H. (2025). Predicting Maintenance Cessation of Open Source Software Repositories with an Integrated Feature Framework. arXiv Preprint. https://doi.org/10.1109/ACCESS.2025.2890645