Enhancing Software Development Through AI-Powered Issue Resolution

Prakhar Jaju
Information Science and Engineering
RV College of Engineering
Bangalore, Karnataka
prakharmaheshwari140@gmail.com

Dr. Vanishree K
Information Science and Engineering
RV College of Engineering
Bangalore, Karnataka
vanishreek@rvce.edu.in

***Abstract*—The research paper delves into an advanced software project integrating AI technologies to overhaul issue identification and resolution in software development.In the fast-paced world of software development, quickly resolving issues is crucial for maintaining quality, meeting deadlines, and keeping customers happy. Traditional methods often fall short in modern software environments, leading to inefficiencies. This project uses Generative AI to automate and enhance the issue resolution process, effectively addressing these challenges.**

Key functionalities such as real-time issue tracking, anomaly detection, automated resolution suggestions, and version control system integration are detailed to showcase the project's capabilities. Real-world use cases and case studies demonstrate its effectiveness in enhancing issue resolution efficiency, reducing development cycles, and improving overall product quality across industry domains.

In conclusion, the GenAI Issue Resolver emerges as a transformative solution that empowers software development teams to address issues swiftly, effectively, and intelligently, thereby enhancing productivity, product quality, and customer satisfaction in the ever-evolving software landscape.

***Index Terms*—Semantic Search, Issue identification, Natural Language Processing, Summarization, Automation, Historical data analysis**

I. Introduction

In the fast growing market of software development, efficient issue identification and resolution play a pivotal role in ensuring product quality, meeting project deadlines, and enhancing customer satisfaction. The complexity of modern software systems, coupled with the ever-increasing scale and diversity of applications, presents significant challenges for developers in promptly addressing issues and maintaining software reliability. Traditional manual methods often fall short in effectively managing these challenges, leading to delays, errors, and resource inefficiencies.

To address these critical issues, the integration and use of various artificial intelligence (AI) technologies has emerged as a transformative solution. AI-powered systems can automate and optimize issue resolution workflows, enabling developers to focus on core tasks and deliver high-quality software products. This research paper explores one such advanced software project, referred to as the "GenAI Issue Resolver," which leverages Generative AI tools to revolutionize issue identification and resolution processes in software development.

This article explores the GenAI Issue Resolver in depth, outlining its architecture, functionalities, advantages, and impact on modern software development environments. By delving into the significance of AI-driven issue resolution and analyzing existing AI-powered solutions, this paper sets the context for understanding the unique features and innovations of the GenAI Issue Resolver. Using real-world, case studies, and performance evaluations, the article reports on the effectiveness of GenAI Issue Resolver in improving problem solving, reducing development cycles and improving overall product quality.

Furthermore, the paper discusses implementation insights, deployment strategies and best practices for seamless integration with existing development workflows.Finally, the article discusses the evolution of AI technology in software development and the important role of GenAI Problem Solving in increasing innovation and productivity in the industry.

Problem Statement:

Efficient issue identification and resolution are crucial for smooth project progression and meeting delivery deadlines in modern software development. However, the complexity of software systems often leads to intertwined issues, challenging developers to isolate root causes swiftly and apply timely fixes. Streamlining issue identification and resolution is vital for optimizing resources, enhancing developer focus, and delivering high-quality products that meet customer expectations. Delayed issue resolution risks error accumulation and compromises product reliability, highlighting the critical need to streamline issue resolution workflows.

Purpose:

The purpose of this project is to develop and implement an efficient issue identification and resolution system that addresses the challenges faced by developers in modern software systems. The complexity of software systems often results in intertwined issues, making it difficult for developers to quickly isolate root causes and apply timely fixes. By streamlining the issue identification and resolution process, the project aims to optimize resource utilization, enhance developer focus on core tasks, and ensure the timely delivery of products. The project recognizes the critical importance of efficient issue resolution in maintaining smooth project progression, meeting delivery deadlines, and minimizing the risk of compounding errors that could impact overall product reliability.

Motivation:

By streamlining the issue resolution work- flow, the motivation is to optimize resource utilization, enhance developer productivity.The goal is to reduce the risks associated with slow issue resolution, such as compounding errors and potential customer dissatisfaction, ultimately driving towards smoother project progression and meeting delivery deadlines effectively.

Formulation of Objectives:

1. To develop a user-friendly dashboard that provides a visual representation of identified issues, including detailed insights and recommended actions.
2. To Integrate GenAI with existing data repositories to access comprehensive historical data related to past issues, resolutions.
3. To Develop intelligent recommendation algorithms powered by GenAI to suggest optimal solutions based on historical data, industry best practices.

Scope and Relevance:

The research paper delves into the extensive scope and relevance of the GenAI Issue Resolver project within software development and data analysis domains. Drawing from Knowledge Discovery in Databases (KDD), the project utilizes advanced techniques to extract valuable insights and anomalies from log data through data mining and machine learning methodologies. Additionally, the integration of Computational Linguistics techniques enhances the project’s ability to process and analyze natural language text from log files, thereby improving its accuracy in understanding textual data.

Moreover, the GenAI Issue Resolver project encompasses a range of Computational Intelligence techniques, including genetic algorithms, neural networks, fuzzy logic, and swarm intelligence. These techniques empower the project to effectively address complex problem-solving tasks, further enhancing its utility and versatility within software development contexts.

The relevance of the GenAI Issue Resolver project in today’s software development landscape is paramount. Its capability to streamline issue identification and resolution processes significantly improves system reliability and performance. he project extracts actionable insights from log data, thereby optimizing resource utilization and developer productivity. Furthermore, the integration of Computational Linguistics techniques ensures precise interpretation of error messages and diagnostic information, leading to quicker and more effective issue resolution. Ultimately, the GenAI Issue Resolver project’s relevance lies in its capacity to enhance software quality, meet delivery deadlines, and deliver high-quality products that align with customer expectations.

II. Literature Review

- A Case Study on Implementing Design Automation: Identified Issues and Solution for Documentation,” explores the impact of design automation on documentation practices within software development. The study addresses identified issues and proposes solutions, contributing to the enhancement of performance evaluation metrics for machine learning models in anomaly detection.
- A structured approach to problem-solving, emphasizing the importance of thorough information review, root cause assessment, and obstacle anticipation. The study provides valuable guidelines for resolving issues effectively across diverse contexts, helping individuals and organizations-navigate complex problem-solving scenarios with clarity and efficiency.
- Studies have shown the effectiveness of both unsupervised and supervised learning algorithms in reducing intrusion alerts in network security systems [1]. The integration of Particle Swarm Optimization (PSO) and K-Means clustering has also been demonstrated to improve alert correlation models, further enhancing the efficiency of anomaly detection systems [2]. Combining neural networks with clustering techniques can significantly improve the accuracy of alarm systems and reduce false alarms, facilitating quicker issue resolution [3]. Event correlation analysis is crucial for rationalizing alarm systems and maintaining robust monitoring in large-scale operations [4]. Clustering techniques are also effective in managing large volumes of IDS alerts and improving the precision of anomaly detection mechanisms [5].
- Log-based anomaly detection without log parsing simplifies the analysis process and enhances the detection of subtle anomalies [6]. Deep learning approaches, such as DeepLog, utilize recurrent neural networks to learn patterns from log data and identify anomalies, showcasing the transformative potential of deep learning in automated log analysis [7]. Unsupervised anomaly detection frameworks based on word2vec capture the semantic meaning of log messages, improving detection accuracy [8]. Unsupervised learning models are effective in identifying unusual log entries without prior knowledge of normal behavior [9]. Applying unsupervised deep learning for anomaly detection provides a robust framework for detecting complex patterns within log data [10].

III. Methodology

Following is the methodology for the project in the form of list of tasks to be fulfilled:

1. Analysis of Existing Workflows and Tools:

The first step involves conducting thorough assessments of current workflows and tools utilized by developers. This analysis aims to identify inefficiencies and areas for improvement in issue identification and resolution. Insights gathered from this analysis help pinpoint specific pain points and challenges faced by developers in handling issues effectively.

2. Integration of Generative AI Tools:

The integration of AI tools is a critical step in the project as it automates the process of retrieving the tickets from JIRA by writing the code for Lambda Function that will automate the data fetching process and then the tickets fetched will be stored in Amazon S3. From there the Amazon Bedrock will fetch the log files and the model will be trained for the output. The tickets are stored in a JSON file format, which offers several advantages. JSON files are easy to parse, making them ideal for storing large amounts of text data. Additionally, JSON files are human-readable, making it easy for researchers and developers to access and analyze the log data. The JSON format’s flexibility allows for seamless integration with various tools and systems, thereby improving the overall usability of log data.

3. Utilize Machine Learning Algorithms:

Implementing machine learning algorithms plays a pivotal role in enhancing the precision and accuracy of issue identification and prioritization. By leveraging historical data and patterns, these algorithms can analyze vast datasets to categorize and prioritize issues effectively. This approach optimizes resource allocation by directing attention to critical issues while enhancing overall efficiency in resolving software-related challenges. Machine learning models become invaluable assets in continuously improving issue resolution processes and adapting to evolving project dynamics.

4. Develop Historical Data System:

Building a centralized system for accessing and analyzing historical data related to past issues is essential for informed decision-making during issue resolution. This system provides developers with valuable insights and context, allowing them to draw from previous experiences and solutions. By harnessing historical data, developers can make more informed choices, avoid repeating past mistakes, and expedite the resolution of similar issues. This centralized repository becomes a knowledge hub that empowers developers with the information needed for efficient problem-solving.

5. Integrate Data Analysis Tools:

The integration of data analytics tools enhances the capabilities of proactive issue resolution strategies. These tools enable the analysis of trends, patterns, and recurring issues within the software development lifecycle. Real-time monitoring and alerting mechanisms further strengthen the ability to detect potential issues early and take preemptive actions. By harnessing data analytics, development teams gain actionable insights that drive continuous improvement, reduce risks, and enhance the overall reliability of software products.

6. Implement Intelligent Recommendation System:

Developing an intelligent recommendation system is a strategic approach to augment decision-making during issue resolution. By leveraging machine learning algorithms and historical data analysis, this system can provide developers with effective solutions based on data-driven insights. Continuous optimization of the recommendation system ensures accuracy, relevance, and alignment with project goals. Ultimately, this intelligent system empowers developers with actionable recommendations, facilitating faster and more effective issue resolution outcomes.

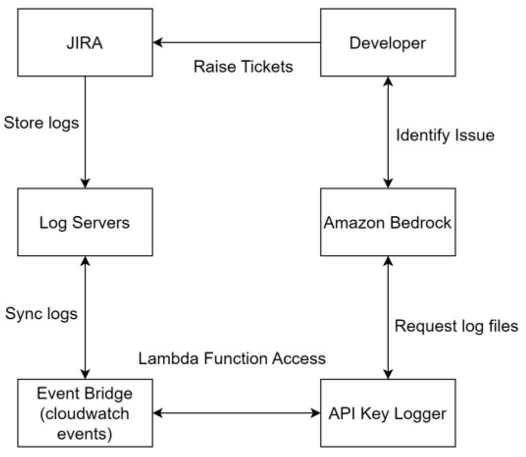

Fig. 1. Overview of Methodology

In fig.1 the developer will have access to both JIRA and Amazon Bedrock where on JIRA the tickets are raised, the tickets will be fetched from JIRA and will be saved on log servers, the data will be then fetched from log servers by Event Bridge and when the developer has any issue the developer will go to Bedrock and just type the issue, the model will fetch the details from the servers.

IV. Results and Discussions

Following process describes the successful integration of the tools creating an advanced search engine for ticket searching that significantly enhanced issue identification :

- **Automated JIRA Ticket Updates:** The current manual process of updating JIRA ticket details in S3 can be automated using AWS Lambda, allowing for real-time data retrieval and updates. This automation streamlines the workflow, reduces manual errors, and ensures that the latest information is always available.

- **Data Synchronization in Knowledge Base:** Implementing an ingestion job to synchronize data within the knowledge base ensures that all relevant information is up-to-date and readily accessible. This synchronization supports better decision-making and enhances the overall efficiency of the system.

- **Enhanced Solution and Remediation Details:** Providing more detailed solutions and remediation information for issues or tickets enhances the clarity and effectiveness of the resolution process. This detailed approach ensures that users have comprehensive guidance to address problems effectively.

- **Server Log Monitoring:** Monitoring server logs is crucial for identifying and diagnosing issues promptly. Implementing a comprehensive monitoring system facilitates the early detection of anomalies, ensuring prompt resolution and enhancing system reliability.

- **Impact on System Performance:** The integration of these enhancements—automated updates, data synchronization, detailed remediation information, and log monitoring—collectively improves system performance. These measures contribute to a more resilient, efficient, and user-friendly software development environment, ultimately leading to higher-quality product delivery.

V. Conclusion

This research achieved the successful integration of S3 and Bedrock with Jira and semantic search technologies, leading to notable improvements in issue identification and resolution within software development. A specialized search engine was developed for ticket searching, enhancing issue identification and streamlining the issue identification process, which significantly improved user experience. Comprehensive ticket collection and remediation strategies were established as the foundation for this search engine. Leveraging AWS and Bedrock, robust cloud storage and enhanced performance were achieved, contributing to a more efficient and effective development environment.

References

[1] Afolabi-B, O. O. K., & Siraj, M. (2021). Intrusion alert reduction based on unsupervised and supervised learning algorithms. International Journal of Innovative Computing, 11(2), 25-34.

[2] Hua, H. H. W., Siraj, M., & Din, M. M. (2017). Integration of PSO and K-Means clustering algorithm for structural-based alert correlation model. International Journal of Innovative Computing, 7(2).

[3] Tjhai, G. C., Furnell, S. M., Papadaki, M., & Clarke, N. L. (2010). A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm. Computers & Security, 29(6), 712-723

[4] Noda, M., Higuchi, F., Takai, T., & Nishitani, H. (2010). Event correlation analysis for alarm system rationalization. Special Issue: Process Systems Engineering (PSE) Asia, 6(3), 497-502.

[5] Njogu, H. W., Jiawei, L. (2010). Using alert cluster to reduce IDS alerts. In International Conference on Computer Science and Information Technology (pp. 467-471).

[6] Le, V. H., Zhang, H. (2021). Log-based anomaly detection without log parsing. In 36th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 492-504).

[7] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1285–1298.

[8] Wang, J., Zhao, C., He, S., Gu, Y., & Alfarraj, O. (2022). LogUAD: log unsupervised anomaly detection based on word2vec. Computer Systems Science & Engineering, 41(3), 1207-1222.

[9] Farzad, A., Gulliver, T. A. (2020). Unsupervised log message anomaly detection. ICT Express, 6(3), 229-237.

[10] Bursic, S., Vittorio, C., & D'Amelio, A. (2019). Anomaly detection from log files using unsupervised deep learning. In FM 2019 International Workshops: Porto, Portugal, October 7–11, 2019 (pp. 200–207).

[11] Schmidt, T., Hauer, F., & Pretschner, A. (2020). Automated anomaly detection in CPS log files. Computer safety, reliability, and security. SAFECOMP 2020. Lecture Notes in Computer Science (pp. 179–194).

[12] Marta, C., Antonio, P., & Umberto, V. (2022). AutoLog: Anomaly detection by deep autoencoding of system logs. Expert Systems with Applications, 3, 116-263.

[13] Qiang Fu, Jian-Guang Lou, Yi Wang, and Jiang Li. 2009. Execution anomaly detection in distributed systems through unstructured log analysis Proc. IEEE International Conference on Data Mining (ICDM).

[14] Dutta, P. 2019. Comparative Study of Cloud Services Offered by Amazon, Microsoft and Google. International Journal of Trend in Scientific Research and Development (ijtsrd), 3, pp.981-985.

[15] Kirchgaessner, S., 2013. Cloud storage carries potent security risk. Financial Times. Viitattu, 31, p.2014.

[16] Mosca, P., Zhang, Y., Xiao, Z. and Wang, Y., 2014. Cloud security: Services, risks, and a case study on amazon cloud services. Int'l J. of Communications, Network and System Sciences, 7(12), p.529.

[17] Awa, H.O., Ojiabo, O.U. and Orokor, L.E., 2017. Integrated technology-organization-environment (TOE) taxonomies for technology adoption. Journal of Enterprise Information Management.

[18] Ratten, V., 2016. Continuance use intention of cloud computing: Innovativeness and creativity perspectives. Journal of Business Research, 69(5), pp.1737-1740.

[19] Qi Zhang, Lu Cheng, RaoufBoutaba. Cloud Computing: Advances and Research Challenges. J Internet ServAppl(2010).

[20] Abramson, J., Ahuja, A., Barr, I., Brussee, A., Carnevale, F., Cassin, M., Chhaparia, R., Clark, S., Damoc, B., Dudzik, A., et al. (2020). Imitating interactive intelligence. arXiv preprint arXiv:2012.05672.

[21] Anthony, T., Tian, Z., and Barber, D. (2017). Thinking fast and slow with deep learning and tree search. arXiv preprint arXiv:1705.08439.

[22] Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. Science, 356(6334):183–186.

[23] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. (2021). Extracting training data from large language models. In 30th USENIX Security Symposium (USENIX Security 21), pages 2633–2650.