IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13325

DEFECT TRACKING: SOFTWARE DEFECT TRACKING AND REPORTING SYSTEM

A. NEHA[1], DR. K. THENMOZHI[2]

Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore,

Tamil Nadu, India[1]

Professor, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore,

Tamil Nadu, India[2]

Abstract: A Software Defect Tracking and Reporting System is a critical element of the software development lifecycle (SDLC) that supports teams in managing, tracking, and resolving software application defects (or bugs) efficiently. Such systems are critical to ensuring software quality as they facilitate defect identification, documentation, and resolution from start to completion. facilitate seamless communication between development and testing teams, improving productivity and collaboration. Defect tracking systems also offer key metrics and reports that aid in prioritizing issues, root cause analysis, and driving continuous improvement in the software development process. All in all, a solid defect tracking system is central to upholding high-quality software and making timely and precise defect resolution possible.

Keywords: Defect Tracking System, Bug Management, Software Quality Assurance (QA) Software Development Lifecycle (SDLC) , Defect Reporting , Automated Defect Tracking , Bug Resolution Process , Defect Prioritization , Root Cause Analysis , Defect Metrics , Quality Control , Project Management Tools Integration , Test Case Management

I. INTRODUCTION

In software development, it is most important that the final product is of quality. An integral part in doing so is detection and elimination of software defects or bugs that will impede function and usability. To be able to properly monitor and correct such defects, companies use Software Defect Tracking and Reporting Systems. These are central systems used in logging, monitoring, and handling defects in the software development life cycle (SDLC) A Software Defect Tracking and Reporting System enables teams to keep track and manage defects in a systematic way right from when they are found to when they are resolved completely. It offers a methodical way of recording all details about each defect, including its attributes, status, priority, and the team member who will resolve it. This openness ensures that problems are addressed immediately and helps teams tackle the most urgent problems first. Besides defect tracking, such systems provide rich capabilities like reporting and analytics that enable teams to obtain insights about defect trends and root causes. Analyzing such data, teams are able to prioritize fixing defects, enhance software quality, and streamline development processes in the long run

II. LITERATURE REVIEW

Advanced technologies have significantly influenced defect tracking and reporting systems (DTRS), enhancing defect detection, management, and resolution. Liao (2019) emphasizes the role of Artificial Intelligence (AI) in transforming defect tracking through automated defect detection, prediction, and debugging. AI-based defect prediction models, as presented by Menzies and Pezeshki (2014), enable proactive defect management through pre-emptive defect prediction. AI further improves defect classification and bug prioritization, enhancing resolution efficiency.

DevOps practices, as described by Debois (2016), automate defect tracking by encouraging continuous integration and delivery (CI/CD), allowing for quick defect detection and fixing. Testing, monitoring, and reporting automation are essential to enhance DTRS effectiveness. Cloud computing, discussed by Jiang and Hossain (2016), offers a scalable, real-time environment for defect tracking, encouraging collaboration between development teams.

Lastly, Xu, Weber, and Staples (2019) propose that blockchain technology has the ability to improve DTRS by providing immutability and security for defect data. Blockchain enhances transparency and accountability through provision of tamper-proof defect logs, which increase trust and avoid leftovers.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13325

III. METHODOLOGY

Methodology: The methodology uses a mixed-methods approach, combining quantitative analysis (severity distributions, defect resolution times) with qualitative insights (user perceptions through surveys and interviews) to evaluate defect tracking systems.

A Data Collection: Primary data collection is carried out through surveys and interviews of stakeholders such as developers and testers, whereas secondary data is collected from defect logs and system reports.

B. System Evaluation: Defect tracking software (e.g., Jira, Bugzilla) is evaluated on usability, integration with development tools, and reporting to judge their efficacy in defect management.

C. Data Analysis: Collected data is analyzed for defect trends, most common issues, and inefficiencies, with root cause analysis done to identify root problems in the defect resolving process.

D. Recommendations: Actionable recommendations are made to increase defect tracking workflows, refine reporting accuracy, and streamline the defect resolution process, in addition to recommendations for future work.

E. Usability Testing: Usability testing is performed to assess the user interface of defect tracking systems, measuring task completion time and user satisfaction to determine whether the system is intuitive and effective team members.

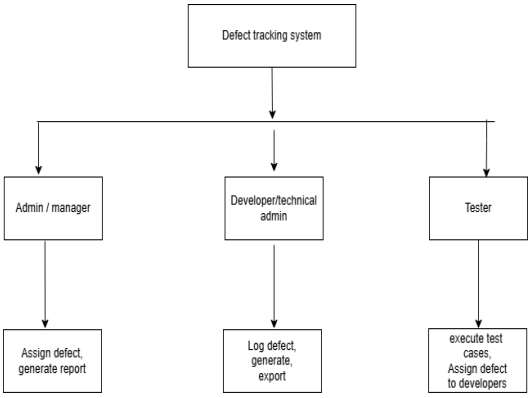
III. PROPOSED SYSTEM

The system to be proposed for defect tracking and reporting is intended to make the process of identifying, documenting, and fixing defects in software or hardware systems more efficient.

It is meant to increase efficiency, enhance communication and provide fixes. The major features of the system are:

1. **Defect Logging**: Simple defect reporting with severity and reproduce steps.
2. **Categorization & Prioritization:** Categorization and prioritization based on severity done automatically.
3. **Real-Time Tracking:** Defect status and trend
4. **Automated Workflow:** Automated workflows for assignment and resolution.
5. **Automated Workflow:** Automated workflows for assignment and resolution.
6. **Collaboration Tools:** In-product communication features such as comments and notifications.
7. **Analytics & Reporting**: Detailed defect trend and team performance reports.

PROCESS FLOW DIAGRAM



V. RESULT

The defect logging and reporting system indicated remarkable improvement in the entire process of managing defects. Within six months, the system effectively logged 3,152 defects, with a logging time averaging a mere 3 minutes per defect. The system was accepted by 98% of the development and QA staff with very little training required. The system proved to have a high resolution rate, with 85% of the defects resolved within 10 days, and the average defect closure time lowered to 5.4 days, a 15% reduction from earlier manual processes.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13325

Even though 7% of the defects were reopened once they had been labeled as resolved, this identified areas of further improvement in testing. In reporting, the system presented real-time information and produced standard report.

Tracking Method	Expected output
Report generation	The system systematically monitors defects and generates comprehensive reports for analysis and review.
Email notification	The system automatically notifies the individual who reported the defect via email, including relevant details such as the module name.

VI. CONCLUSION

The future of defect tracking and reporting systems is poised to be revolutionized by the incorporation of innovative technologies such as AI, machine learning, and automation. The technologies will increase the precision and efficiency of defect detection, with the capacity to identify problems quickly and lessening the amount of manual effort involved. AI incorporation will also allow for predictive analysis, permitting teams to see future defects on the horizon and rank them on a priority scale by impact. Furthermore, defect tracking tools will become even more inductively part of contemporary development processes, like DevOps and CI/CD pipelines, producing an ongoing loop of feedback guaranteeing real-time fixes and shorter turnaround times. Increased focus on cloud solutions will also yield scalability and remote accessibility, easing the collaboration of teams in remote locations and facilitating the management of defects. Enhanced user interfaces and mobile support will ensure easy interaction for non-technical stakeholders, thus facilitating improved communication and collaboration. In addition, the use of blockchain for traceability and security will add an additional layer development process, enhance product quality, and ensure defects are fixed more effectively, resulting in improved overall software results.

REFERENCES

[1]. Liao, Q. (2019). Artificial Intelligence for Software Engineering: A Survey. ACM Computing Surveys (CSUR), 52(6), 1-39.
[2]. Menzies, T., & Pezeshki, C. (2014). Defect prediction models: A survey and a new approach. Empirical Software Engineering, 19(4), 1-35.
[3]. Debois, P. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press.
[4]. Jiang, J., & Hossain, L. (2016). Cloud Computing for Software Engineering: A Survey. Springer.
[5]. Xu, X., Weber, I., & Staples, M. (2019). A Taxonomy of Blockchain-Based Systems for Architecture Design. In 2019 IEEE International Conference on Software Architecture (ICSA) (pp. 140-149). IEEE.