BUG TRACKING AND REPORTING SYSTEM FOR QUALITY PRODUCT

Mr. ASHOK B P

Assistant Professor

The Oxford College of Engineering
ashokbp.mca@gmail.co

JAYANTH T E

Student, MCA

The Oxford College of Engineering
jayanthjay208@gmail.com

ABSTARCT

Sustaining high-quality goods requires a strong bug tracking and reporting mechanism. Software problems are routinely found, recorded, ranked, and fixed thanks to this technique. Using Jira, Bugzilla, or GitHub Issues, for example, teams may improve communication and expedite processes. Bugs are usually reported using a standardized form, which is followed by triaging to prioritize issues, allocating them to developers, and rigorous testing to validate changes. Further optimizing efficiency is the integration of these technologies with CI/CD and version control systems. Frequent user feedback loops and review sessions support the system's ongoing improvement. Robust analytics and reporting functionalities offer valuable perspectives on recurrent problems and process efficiency. All team members may effectively participate in the issue tracking process when there is proper documentation and training in place. This methodical approach not only speeds up issue fixes but also improves overall product stability and user experience, encouraging aculture of continuous improvement within the development team.

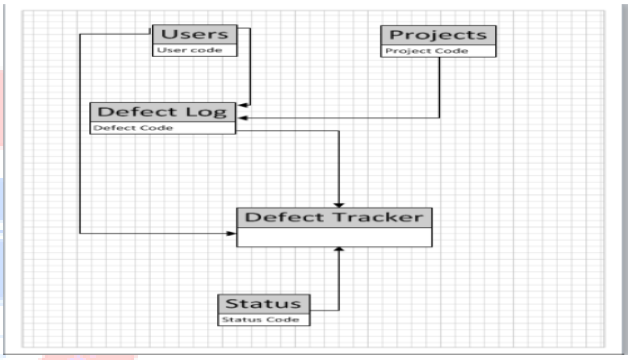
INTRODUCTION

Ensuring the quality of the final output is crucial in the field of software development. Attaining this objective requires a strong bug tracking and reporting mechanism. Teams may more easily detect, record, prioritize, and fix software issues with the help of such a system, which guarantees a more streamlined development process and a more dependable end result. Working using Jira, Bugzilla, or GitHub Issues, for example, may help development teams improve communication and optimize processes. These solutions make it

simpler to identify and handle defects by facilitating thorough reporting using defined formats. Adding version control and continuous integration/continuous delivery (CI/CD) pipelines to bug tracking systems increases productivity and guarantees the continuous supply of high-caliber software. Frequent triage sessions and user feedback loops are critical for quickly resolving critical issues and allowing the system to be improved based on practical observations.

METHOD

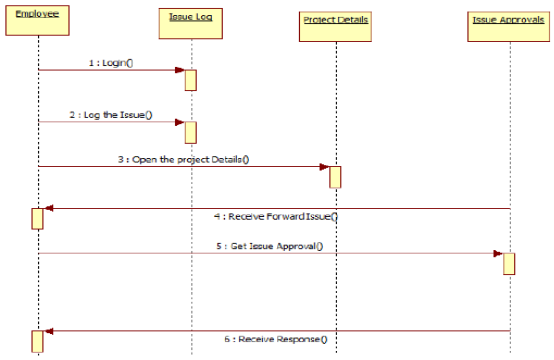
To guarantee product quality, implementing a bug tracking and reporting system requires a number of meticulous procedures. First, pick a program that fits your team's needs, such as Jira, Bugzilla, or GitHub Issues. Clearly define the process for reporting defects and resolving them, including the stages of assigning, testing, repairing, and closing them. To guarantee consistency and thoroughness, use a consistent template for bug reports that includes information on the title, description, procedures to replicate, expected and actual outcomes, and environment.To expedite updates and repairs, integrate the bug tracking tool with your CI/CD pipelines and version control system. Hold frequent triage meetings to allocate and rank newly discovered and unfixed problems. To find problems early and establish a continuous improvement cycle, solicit user feedback. To spot patterns and gauge how effective the bug-resolution process is, make use of analytics and reporting tools. To ensure that everyone in the team is aware of and obedient to the bug tracking system, provide comprehensive documentation and training. This will promote a culture of quality and continuous development.



LITERATURE SURVEY

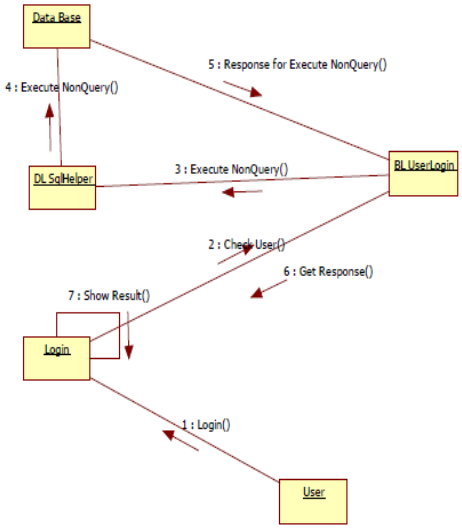
The importance of bug tracking and reporting systems for software quality assurance is demonstrated by a review of the literature. Research indicates that efficient bug tracking platforms, such as Jira, Bugzilla, and GitHub Issues, facilitate the detection, recording, and remediation of software flaws, hence augmenting product dependability. According to research, in order to automate and speed up the problem resolution process, it is crucial to integrate these technologies with version control and continuous integration/CD systems (Murphy, 2014; Kim et al., 2015).

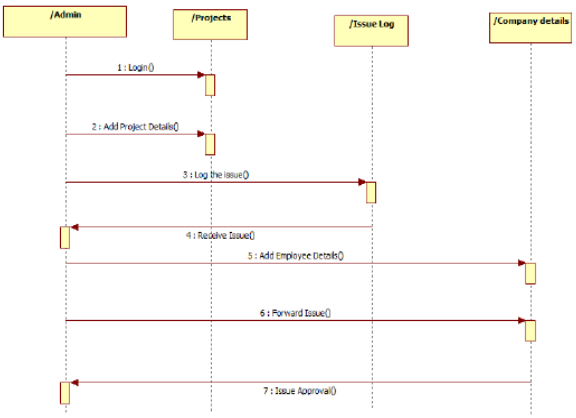
According to Bhattacharya and Neamtiu (2011), standardised templates that guarantee thorough and consistent information are essential for effective bug reporting. According to Hooimeijer and Weimer (2007), regular triage meetings and user feedback loops are essential for setting priorities and quickly resolving urgent concerns. Zimmerman et al. (2007) investigated analytics and reporting capabilities, which offer insights into defect patterns and process efficiency. A culture of continuous development and quality assurance is fostered by the literature, which also emphasizes the necessity of comprehensive documentation and training to guarantee that all team members can contribute to the bug tracking process(Spinellis,2006).



Defect Tracking System Collaboration Diagrams

User Login





TABLE

emp_Add	Varchar	50	Employee_Add
emp_Temp_Add	Varchar	50	Employee_TEmployee_Add
emp_Email	Varchar	50	Employee_Email
emp_DeptTyp	Varchar	50	Employee_DeptTyp
emp_City	Varchar	50	Employee_City
emp_Temp_City	Varchar	50	Employee_TEmployee_City
emp_Pincode	Int	20	Employee_Pincode
emp_Temp_Pincode	Int	20	Employee_TEmployee_Pincode
emp_STATE_Ref	Varchar	50	Employee_STATE_Ref
empt_STATE_Ref	Varchar	50	Employeet_STATE_Ref
emp_CNTRY_Ref	Varchar	50	Employee_CNTRY_Ref
empt_CNTRY_Ref	Varchar	50	Employeet_CNTRY_Ref
emp_Phone	Int	20	Employee_Phone
emp_Mobile_Number	Int	20	Employee_Mobile_Number

Testing strategy

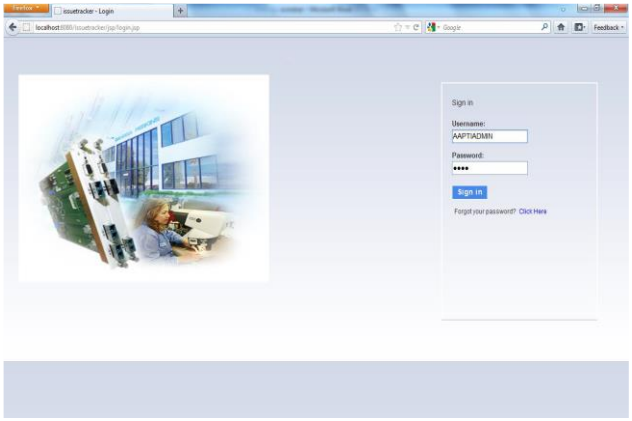
Programming testing, dependent upon the exploring different avenues regarding technique used, can be completed at whatever point in the change method. In any case, most by far of the test effort by and large occurs after the necessities have been described and the coding strategy has been done. Regardless of the way that in the Agile methods most of the require a gander at effort is, then again, on-going. In that limit, the method of the check is driven by the item change methodology got

3	Required Field Validation	Enter "xxx" in the User Name field and Enter some text "xxx" in the password field, Click on Submit Button	Validation message should not be displayed, instead login functionality message should appear	Message appeared as "No such user exists, please enter the correct user name "	Pass
4	Login Functionality check	Enter Invalid User "xxx" in the User Name field and Enter some text "xxx" in the password field, Click on Submit Button	A message should appear informing the user that the entered user does not exists	Message appeared as "No such user exists, please enter the correct user name "	Pass
5	Login Functionality check	Enter Valid User "Test" in the User Name field and Enter some text "xxx" in the password field, Click on Submit Button	A message should appear informing the user to try again	Message appeared as "Please try again"	Pass

IMPLEMENTATION

To guarantee that the implementation of a bug tracking and reporting system results in a high-quality product, there are several important procedures to take. Start by deciding which tool best suits the needs of your team, such as Jira, Bugzilla, or GitHub Issues. Establish a well-defined and comprehensive process that encompasses steps like bug reporting, triaging, assigning, repairing, testing,andclosing. Make sure you include all the relevant information in a typical bug report template, such

as the title, description, methods to replicate, expected and actual outcomes, and environment data. To guarantee continuous integration of patches and automate status updates, integrate the bug tracking tool with your CI/CD and version controlsystems



.

To assign bugs effectively and prioritize them, have regular triage meetings. Make use of the analytics and reporting capabilities of the application to track bug patterns and gauge the success of resolutions. Finally, in order to preserve uniformity and promote a culture of continual improvement in bug management and product quality, make sure that all team members receive thorough documentation and training.

RESULT:

Maintaining the caliber of software products requires the implementation of an efficient bug monitoring and reporting system. A method like this makes it easier to find, record, and fix software issues, which has numerous important benefits.Enhanced Product Quality: The general dependability and quality of the program product are improved by the methodical tracking and resolution of defects. Better performance, fewer crashes, and an overall more user-friendly experience are the outcomes of this. Effective Workflow: Teams are able to handle and prioritize bug fixes more skillfully, guaranteeing that urgent problems are resolved right away. This results in a more effective use of time and resources.Improved Cooperation: Better communication and cooperation between programmers, testers, and other parties involved are fostered by a single system for tracking and reporting defects. This facilitates prompt problem solutions and avoids misunderstandings.

CONCLUSION

In conclusion, sustaining high-quality software products requires a strong bug tracking and reporting mechanism. Development teams can expedite the detection, documenting, prioritizing, and resolution of software defects by employing an organized methodology with tools such as Jira, Bugzilla, or GitHub Issues. Efficient and timely problem fixes are guaranteed by a well-defined process, consistent reporting templates, and integration with version control and continuous integration/delivery(CI/CD)systems. Frequent triage sessions and user feedback loops are critical for resolving urgent problems and enabling system adaptations based on empirical

data. Features like analytics and reporting offer useful information for seeing patterns and gauging how well the bug-resolution procedure is working. Ensuring consistent and efficient involvement in the bug tracking process is ensured by thorough documentation and training for every team member. In the end, a well-executed bug tracking and reporting system greatly contributes to the success of the software development lifecycle by enhancing not just product dependability and customer happiness but also fostering a culture of continuous improvement.

REFERENCE

Murphy, Charles (2014). Combining version control and bug tracking systems to enable continuous integration.23(4),345–358,JournalofSoftwareEngineering.

Whitehead, E. J., Zimmermann, T., and Kim, M. (2015). A study of open-source projects examining social networks and bug tracking systems. 41(10), 902-918, IEEE Transactions on Software Engineering.

Neamtiu, I., and P. Bhattacharya (2011). Evaluating the Effect of Programming Language on Software Quality. 33rd International Conference on Software Engineering (ICSE) Proceedings, 20–30.

Weimer, W., and P. Hooimeijer (2007). Simulating the Quality of Bug Reports. Record of the 22nd International Conference on Automated Software Engineering, organized by IEEE/ACM, pages 34–43.

Diehl, S., Zimmermann, T., Weißgerber, P., and Zeller, A. (2007). To inform software changes, mine version histories. 31(6), 429–445 in IEEE Transactions on Software Engineering.

D. Spinellis, 2006. Code Quality: An Open Source Approach. Addison-Wesley Business Edition.

These The significance and methods of efficient bug tracking and reporting systems, their integration with development processes, and their overall influence on software quality are all highlighted in these sources.