

# Bug Tracking System

Kamya P. Shah<sup>1</sup>, Prof. Ajaykumar T. Shah<sup>2</sup>

<sup>1</sup>Dept of Computer Engineering

<sup>2</sup>HOD, Dept of Computer Engineering

<sup>1,2</sup>Alpha College of Engineering and Technology

**Abstract-** Bug tracking System is a project which provide the project management, bug management, daily task management and chat with multiple employees. In the system admin can manage multiple project and tester with status of bug. Tester can update their daily task and also update which bug they found and once bug is fixed by the developer it update the status. So the project can manage multiple tester, multiple bugs of same project, multiple project. The proposed of the bug tracking system is to test the applications for the bugs and report it to the project manager and developer. The main intention behind the bug tracking system is that to track bugs and reports them. Store the bug information with a unique id in the database for further reference. So, this makes the job of handling the bugs easy.

## I. INTRODUCTION

In the System Admin can manage multiple project and tester with status of bug. Tester can update their daily task and also update which bug they found and once bug is fixed by the developer it update the status. So the project can manage multiple tester, multiple bugs of same project, multiple project. The proposed of the bug tracking system is to test the applications for the bugs and report it to the project manager and developer. The main intention behind the bug tracking system is that to track bugs and reports them. Store the bug information with a unique id in the database for further reference. So, this makes the job of handling the bugs easy.

The project entitled “Bug Tracking System” is an online bug tracking tool initialized with the objective to setup a user-friendly online bug tracking system. It can be used both for bug tracking and for project management. In this system, the project manager can have full details of the work assigned to each team member. Moreover when a new work comes he can assign the work to different persons by having a view at the programmer with minimum work.

This software assists to track the work flow of the work given to each team member by a project leader. This software assists the project managers, the team members and equally the top officials of a software company to know how the work is progressing. Project usually comes to the company in the form of bugs. Usually when certain enhancement of a product is being done. This software is mainly focused on the work coming in the form of bugs. The project manager can fully understand what the status of each bug is whether it is fixed, assigned or won fix.

## II. KEYWORDS

Project management, Bug management, Daily task management, Chat with multiple employees, Manage multiple project, Status of bug, Update daily task, Update which bug they found, Update the status, Manage multiple tester, Multiple bugs of same project, Multiple project.

## III. LITERATURE REVIEW

### A. Technique to Combine Feature Selection for Dataset Reduction with Effective Bug triage:

Technique to Combine Feature Selection for dataset reduction with Effective Bug Triage. It addresses the issue of data reduction for bug triage by text classification techniques. Conventional software analysis is not totally suitable for the large-scale and complex data in software repositories. Data mining has developed as a promising means to handle software data.

There are two difficulties related to bug data that may influence the effective use of bug repositories in software development tasks, namely the huge scale and the low quality. Therefore unfixed bugs are deleted from the bug repositories.

### B. Towards Effective Troubleshooting with Data Trim:

Towards Effective Troubleshooting with Data Trim deals with reducing the data present in the bug repository and improve the quality of data then reduce time and cost of bug triaging, it represent an automatic approach to predict a developer with relevant experience to solve the new coming report.. The bug data sets are obtained and techniques such as instance selection feature selection are applied simultaneously.

The top k pruning is applied for improving results of data reduction quality, obtaining domain wise bug solution. Instance selection is for obtaining a subset of relevant instances (i.e., bug reports in bug data) .It is used to Remove noise and redundant instances, Remove non representative instances. Feature selection which aims to obtain a subset of relevant features.

### C. Adjective Bug Triaging Using Text Mining:

Adjective Bug Triaging Using Text Mining aims for an automatic approach to predict a developer with relevant experience to solve the new coming report. The techniques used are five term selection method. Term selection methods are used to reduce the high dimensionality of term space by selecting the most discriminating terms for the classification task. The methods give a weight for each term in which terms with higher weights are assumed to contribute more for the classification task than terms with lower weights. The goal of bug triaging is to assign potentially experienced developers to new-coming bug reports.

To reduce time and cost of bug triaging, we present an automatic approach to predict a developer with relevant experience to solve the new coming report. It investigate the use of five term selection methods on the accuracy of bug assignment. In addition, it re-balance the load between developers based on their experience. It conduct experiments on four real datasets. To reduce the time spent triaging, it present an approach for automatic triaging by recommending one experienced developer for each new bug report. This information can help to manage the progress of these projects. In the last decade, practitioners have analyzed and mined these software repositories to support software development and evolution. One of the important software repositories is the bug tracking system (BTS).

### D. Automatic Bug Triage using Semi Supervised Text Classification:

Automatic Bug Triage using Semi Supervised Text Classification propose a semi-supervised text classification approach for bug triage to avoid the deficiency of labeled bug reports in existing supervised approaches. This approach combines naive Bayes classifier and expectation maximization to take advantage of both labeled and unlabeled bug reports. This approach trains a classifier with a fraction of labeled bug reports. Then the approach iteratively labels numerous unlabeled bug reports and trains a new classifier with labels of all the bug reports. Then it employs a weighted recommendation list to boost the performance by imposing the weights of multiple developers in training the classifier. Before training a supervised classifier for bug triage, a necessary step is to collect numerous labeled bug reports, which are bug reports marked with their relevant developers.

The semi supervised text classification approach to improve the classification accuracy of bug triage. This semi supervised approach enhances a NB classifier by applying expectation-maximization (EM) based on the combination of

unlabeled and labeled bug reports. First, this semi supervised approach trains a classifier with labeled bug reports. Then, the approach iteratively labels the unlabeled bug reports and trains a new classifier with labels of all the bug reports.

### E. Reducing Features to Improve Bug Prediction:

Recently, machine learning classifiers have emerged as a way to predict the existence of a bug in a change made to a source code file. The classifier is first trained on software history data, and then used to predict bugs. Two drawbacks of existing classifier-based bug prediction are potentially insufficient accuracy for practical use, and use of a large number of features. These large numbers of features adversely impact scalability and accuracy of the approach. Reducing Features to Improve Bug Prediction aims in classifier to first trained on software history data, and then used to predict bugs. The disadvantage of the traditional method is that, classifier-based bug predictions are potentially insufficient accuracy for practical use, and use of a large number of features. The system uses Naive Bayes and Support Vector Machine (SVM).

The system mainly Gain Ratio for feature selection, along with the characterization of bug prediction results achieved when using feature selection. This paper proposes a feature selection technique applicable to classification-based bug prediction. This technique is applied to predict bugs in software changes, and performance of Naive Bayes and Support Vector Machine classifiers is characterized. These features include everything separated by whitespace, in the code added or deleted in a change. This leads to a large number of features, in the thousands, and low tens of thousands. For larger project histories which span thousand revisions or more, this can stretch into hundreds of thousands of features. The addition of many non-useful features reduces a classifier's accuracy. Additionally, the time required to perform classification increases with the number of features, rising to several seconds per classification for tens of thousands of features, and minutes for large project histories. A standard approach (in the machine learning literature) for handling large feature sets is to perform a feature selection process to identify that subset of features providing the best classification results. This paper introduces a feature selection process that discards features with lowest gain ratio until optimal classification performance is reached for a given performance measure.

## IV. PROPOSED SYSTEM

Based on the survey we have understood that the system will handle structured data including user information,

login information, data code information and reports. The system stores data in MySQL database cluster by the use of Web interface in Thrift and after that it stores data in database server by database server and Web server. For training purpose we use admin/user information and we develop online bug tracking system for peoples. Testers and Developers can check their waiting time by using this system and thus can reduce their frustration

## V. SCOPE

Bug tracking system is the perfect or unique solution to track the bug of a solution, product or an application. Bug tracing system admits single or set of developers to continue track of not finished bugs in their product successfully. Bug tracking system can also call as defect tracking system. For good performance the bug tracking system can increase a lot, the accountability and productivity of single employees by giving a positive feedback and backup the workflow. The bug tracking software allows or group of testers or individual testers to keep path of unfixed bugs in their software successfully. The bug tracing software can track bugs, can handle code changes, can share information with teammates, submit and review connects and control standard assurance.

There are basic modules: Admin and Employee/Tester. Admin can able to log in system using their email id and password. Admin can able to change and reset password. Admin can manage multiple projects. Admin can manage multiple project categories. Admin can manage the multiple employee/tester. Admin can add task for tester and tester can view the task. Admin can view the status of task which is updated by the admin. Admin can do chat with multiple employee. Admin can view project wise open task and close task. Employee can logged in to the system using their email id and password. Employee can manage his/her profile. Employee can update password and reset password. Employee can add bug project wise and also can manage the status of given bug. Employee can do chat with admin only.

## VI. CONCLUSION

We will develop “BUG TRACKING SYSTEM” System with great concern and will try our best to implement as many as features to make it viable and usable. There is not such a system for specific QA tester. As of now technology increases day by day, Developers want their system to work correctly and smoothly and mainly errors free and bugs free. For maintain their system with error free and work properly the bug tracking system will help them. Developer can maintain their project with the help of this system.

## VII. ACKNOWLEDGMENT

“Defeat is not when you fall down; it is when you refuse to get up”. We faced many difficulties during our project to ensure, right from the requirement gathering to implementation. There were times when the goal looked beyond reach but all difficulties were accepted as challenge. Greater challenges were the effort to overcome it. It has been rightly said that we are built on shoulders of others. For every things we have achieved the credit goes to all those who really helped us in completing this project successfully. We would like to thank my Project Leader **Prof. Ajaykumar T. Shah** for providing a vision about the system. We have been greatly benefited from their regular critical reviews and inspiration throughout my work. It is our earnest endeavour to express sincere thanks to the Internal Guide **Prof. Ajaykumar T. Shah** and faculty members for their kind co-operation, help and delighted support. Finally, we wish to thank our parents and friends who directly or indirectly helped us in completion of this project. I privileged to experience a sustained enthusiastic and involved interest from his side. It was only due to their support, motivation and encouragement that could steer through the project on an honest course to splendour of success.

## REFERENCES

- [1] Tracking systems, In VL/HCC08 Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, pages 8285, September 2008.
- [2] J. Aranda and G. Veolia, The secret life of bugs: Going past the errors and omissions in software repositories, In ICSE09 Proceedings of the 31st International Conference.
- [3] A Literature Review of Research in Software Defect Reporting Jonathan D. Strata, Member, IEEE, and Phillip A. Lapland, Fellow, IEEE, June 2013