

# User Friendly Bug Tracking Mechanism by Using Data Reduction Technique FS-IS

<sup>#1</sup>Shubham Dhole, <sup>#2</sup>Mahesh Ingole, <sup>#3</sup>Nitin Guthe, <sup>#4</sup>Akash Patil,  
<sup>#5</sup>Prof. Santosh Waghmode

<sup>1</sup>shubhamdhole1@gmail.com

<sup>2</sup>maheshingole001@gmail.com

<sup>3</sup>nitinguthe60@gmail.com

<sup>4</sup>akashpatil9657@gmail.com

<sup>#1234</sup>Department of Information Technology

<sup>#5</sup>Prof. Department of Information Technology

JSPM's ICOER, Wagholi.



## ABSTRACT

A software bug is a problem of all MNC as well as Small Scale company. Bug causes a computer program code or system to crash or produce invalid output or to behave unintended way. Software bugs are not to take a simply to avoid it is very harmful to computer system. Many software companies as well as banking sector have to face large number of software bugs problems. Bug Triage is the option to consumes more time for handling software bugs. It is the process of assigning or allocating a new bug problem to the correct potential developer or employee. There are various existing techniques or process for bug triage. In this paper, we have review some of these techniques. It includes Text categorization, Recommendation, Role-Based, Text Mining etc. Most of these techniques and process provide a better result. This technique is automatic bug triage. Some of these techniques are further classified. We have also added a new module here this module will show the status of the bug like whether it assign to any developer or employee or not and also to show the error is rectified or not.

**Key Words:** Bug Triage, Text categorization, Recommendation, Role-Based, Text Mining

## ARTICLE INFO

### Article History

Received: 2<sup>nd</sup> May 2016

Received in revised form :

3<sup>rd</sup> May 2016

Accepted: 5<sup>th</sup> May 2016

**Published online :**

7<sup>th</sup> May 2016

## I. INTRODUCTION

Software repositories check valuable data about software projects. This information of data can help to manage or analysis the progress of ongoing projects. In the last decade, practitioners have analyzed and mined these software repositories to support software development and evolution. The most important software repositories are the bug tracking system. Many open source software is available in real world, projects have an open bug repository that allows developers and users to submit error or issues in the software, suggest possible of this error removing, and comment on existing bug reports. Mining software repositories is an interdisciplinary domain, which aims to employ data mining to deal with software engineering problems. In modern software development, software repositories are large-scale databases for storing the output of software development, e.g., source code, bugs, emails, and specifications.

Simply software analysis is not completely or fully suitable for the complex data in software repositories. Data mining has most important to emerged as a promising means to handle software data. A bug repository (for storing details

of bugs), plays an important role in this project to managing upcoming software bugs. Software bugs are inevitable and fixing bugs are expensive in software development. Software companies like MNS, Small Scale, Banking Sectors spend more percentage of cost in fixing bugs. Large software projects deploy bug repositories (also called bug tracking systems) to support information collection and to assist developers to handle bugs. In a bug repository, a bug is managed as a bug report, which records the textual description format of reproducing the bug and updates according to the status of bug fixing. A bug repository provides a data platform to support many types of tasks on bugs, e.g., fault prediction bug localization, and reopened-bug analysing. In this paper, bug reports in a bug repository are called bug data.

## II. LITERATURE SURVEY

Bug Triage consumes more time for handling software bugs. In traditional software development, a human triager is used i.e. expert developers were manually triaged the new bugs. But manual Bug Triage is expensive in time and

accuracy because of large number of daily bugs and the lack of expertise of all bugs [1].

To reduce this expensive cost of manual bug triage, Cubrani and Murphy [2] first propose the problem of automatic bug triage. They apply machine learning techniques to assist in bug triage by using text categorization. Text categorization is also known as text classification which is a technique of automatically sorting a set of documents into categories from a predefined set. In this paper, developers will get predicted using the bug's description [2]. This paper used supervised machine learning technique using Naïve Bayes classifier to predict the correct developer.

Anvik et al. [3] present a semi-automated approach for the assignment of bug reports to a developer. This paper uses a supervised machine learning algorithm. For the bug assignment problem, the text documents are the bug reports and the label of the documents are the names of developers suitable to resolve the report. In machine learning, the documents are called instances and the attributes of an instance are called features [3]. A supervised machine learning algorithm takes a set of instances as input with known labels and generates a classifier. Then this generated classifier can be used to assign a label to an unknown instance. In this way this paper used supervised machine learning for bug assignment.

Xuan et al. [4] present a semi-supervised approach for automatic bug triage using text classification. This approach is used to avoid the deficiency of labeled bug reports in existing supervised approaches. This approach combines naive Bayes classifier and expectation maximization [5] (EM) to take advantage of both labeled and unlabeled bug reports. Xuan et al. [4] trains a classifier with a fraction of labeled bug reports. Then the approach iteratively labels numerous unlabeled bug reports. Then they train a new classifier with labels of all the bug reports. From the result of [4], this semi-supervised approach improves the classification accuracy of bug triage by up to 6% and it avoids low-quality bugs.

M. Alenezi et al. [6] propose an automatic approach using text mining to reduce time and cost of bug triaging. Existing techniques also address the problem of bug triaging [7], [8] but these are not efficient. M. Alenezi et al. [6] predict a developer which has relevant experience to solve the new coming report. In this approach, First step is text processing. A Bug report contains unstructured data. Therefore, by using traditional text processing technique text data is transformed into meaningful data. Here, they use summary of bug reports as a description of bugs [6]. After that, bug-term matrix is formed. And it is weighted by term frequency. Then, to reduce the dimensionality and the sparseness of data, different term selection methods are applied. In the next step, using Naïve Bayes approach, classifier gets build. This classifier is then trained using the training data set i.e. bug reports. When a new report arrives, then this bug report follows the same steps to produce the reduced bug-term vector, and then bug report is assigned to a developer using the predictive model [6].

### III. OBJECTIVES

1. Study of feature selection techniques to classify bugs in software code changes.
2. Use of algorithm to utilize high accuracy and minimum cost of bug prediction.
3. We have focused on reducing bug data set in order to have less scale of data and quality data.
4. Data reduction more in bug triage to explore how to prepare a high quality bug data set.

### IV. PROPOSED SYSTEM

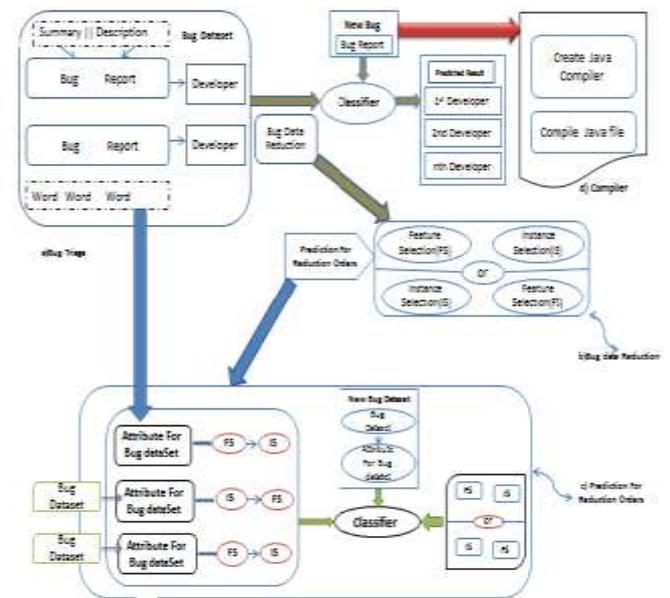


Fig 1. System Architecture

This above system analysis the products, Bugs and bug Tracking. The Proposed system advantage of analysis and maintaining bug history it stores all the details from bug starting to bug resolution.

All products have versions for easy maintenance and identified of the product and all the user of the product is stored in the database. It provides the facility of maintaining users to the bugs and resolutions provided by them.

Our proposed system provides the search based on status, priority, and operating system.

This system has hierarchy structure for user and bug, helpful in knowing the relation and difference between bugs and users allotted to the bug.

It is provided very secure system with a fully authenticated with password encryption technique.

One can keep a track and log of the bug in a product. Using this facility admin can maintain lower cost and effort.

The most advantage of this system is maintaining log records which are helpful in knowing any errors or misuse of the system by other users.

## Modules

1. Employee
2. Manager
3. Administrator

## Module Description

### Employee

Employees are of two types, developers and testers. Developers are used to develop program and open bugs whereas testers resolve the bugs and save to the database. Here Employee handles all the bugs, error, defect and resolve it. This bug dataset are coming form Manager to all employees.

#### Functionality:

- Analysis of upcoming bug error.
- Solve the upcoming bug error.
- Tester can add the script and resolve the error.

### Manager

Manager takes care of recruitment of employees and the management of employees in the project and monitors the completion of project.

#### Functionality:

- Distributed all bug dataset to employee.
- Checking all bug report.
- Analysis of all bug error from employee they solve or not.

### Administrator

Administrator is a person who will take care of all registration status, acceptance of new bugs, and many other tasks to reduce burden on employee.

#### Functionality:

- Over all facility to check all record.
- Check all employee details.
- He has authority to change any procedure in this process.

## V. RESULT

Our Project Bug Tracking and Reporting System helps an Software Concern to detect and manage the bug in their products effectively-efficiently. Utilizing bug tracking software can assist in troubleshooting errors for testing and for development processes. With the ability to provide comprehensive reports, documentation, searching capabilities, tracking bugs and issues, bug tracking software is a great tool for those software development needs. Depending on your development needs and the bug tracking software, you can hope to gain several benefits from bug tracking software. Some of the benefits are:

- 1) Improve communications between employee and manager.
- 2) Increase the quality of the software using the removing bugs from system.
- 3) Improve customer feedback from satisfaction with bug free software
- 4) Increases overall productivity and bug free software.

## VI. CONCLUSION

Bug Triage is a time-consuming and very easy step of handling software bugs or error. It is the process of assigning a new bug to the correct potential developer. This paper reviews various techniques of bug triage. Some of them are of machine learning, incremental learning, tossing graph, fuzzy-based, role-based etc. Most of these techniques provide automatic bug triage. Apparently all bug triage techniques certainly have their advantages and some drawbacks. So they can be used by considering their strengths and drawbacks.

## ACKNOWLEDGEMENT

We would like to thanks JSPM's ICOER for valuable support. We gratefully acknowledge Computer department of their helpful support and guidance for writing this paper.

## REFERENCES

- [1] Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, "Towards Effective Bug Triage with Software Data Reduction Techniques", IEEE transactions on knowledge and data engineering, vol. 27, no. 1, january 2015
- [2] D. Cubrani and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004, pp. 92–97.
- [3] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.
- [4] J. Xuan, H. Jiang, Z. Ren, J. Yan, and Z. Luo, "Automatic bug triage using semi-supervised text classification," in Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng., Jul. 2010, pp. 209–214.
- [5] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," Machine Learning. Hingham, MA, vol. 39, pp. 103-134, May 2000.
- [6] M. Alenezi, K. Magel, S. Banitaan "Efficient Bug Triaging Using Text Mining" Academy Publisher, 2013.
- [7] W. Zou, Y. Hu, J. Xuan, and H. Jiang, "Towards training set reduction for bug triage," in Proc. IEEE 35th Annual Computer Software and Applications Conference, Washington, DC, USA: IEEE Computer Society, 2011, pp. 576–581.
- [8] A. Tamrawi, T. Nguyen, J. Al-Kofahi, and T. Nguyen, "Fuzzy set and cache-based approach for bug triaging," in Proc. 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, 2011, pp. 365–375.
- [9] G. Jeong, S. Kim, and T. Zimmermann, "Improving bug triage with tossing graphs," in Proc. Joint Meeting 12th Eur.

Softw. Eng. Conf. 17th ACM SIGSOFT Symp. Found. Softw. Eng., Aug. 2009, pp. 111–120.

[10] Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis, “Efficient ticket routing by resolution sequence mining,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2008, pp. 605–613.