International Journal of Scientific and Technical Advancements
IJSTA
ISSN: 2454-1532

Duplication Removal in Bug Tracking System

Diana Cardoz[1], Jennifer Coelho[2], Samantha Joseph[3]
[1, 2, 3]Computer Engineering, Mumbai University, Mumbai, Maharashtra, India-400016
Email address: [1]cardoz5diana@gmail.com, [2]jennifer94coelho@gmail.com, [3]samw.j0605@gmail.com

Abstract— These days, almost every software program is dependent on bug reports for error correction, testing and maintenance purposes. Due to excessive number of bugs in any software, bug tracking system is developed for the ease of maintenance for a developer. Due to large number of duplicate bug reports, developers spend a great amount of time to differentiate between these bug reports. By using the feedback technique, it distinguishes duplicate and valid bugs to minimize the amount of time and effort required to correct the bugs. The proposed method, classifies bug to identify duplicate bugs. These duplicate bugs are eliminated from the system due to which bug error correction becomes faster. This mechanism improves the result of classified bug reports.

Keywords— Bug classification, duplication removal, software maintenance.

I. INTRODUCTION

As many software projects are coming into picture, it becomes more difficult to check all the source codes from files. This also includes the testing of the projects before it's given to the end user. It even becomes a complex task to maintain these projects. Essential activities in software maintenance include bug reporting and fixing. Maintenance activities of many software projects rely on bug reports to correct defects in source code files. Recently plenty of software companies use bug tracking systems during software development to track bugs, where a developer can correct these bugs quickly. Many open source systems are available.

II. REVIEW OF LITERATURE

There are number of researchers that have tried to find out what factors could reduce the time for resolution of bugs. Runeson et al. proposed an approach to identify duplicate bug reports by using Natural Language processing techniques [1]. A prototype tool was developed for evaluating and analyzing bug reports. The result shows that about two thirds of the duplicates can be found possibly by using the NLP technique. It is still difficult to process semi structured data in bug tracking system. H.M. Tran et al. put forth a semantics-based bug search system for extracting and then converting semi-structured data from other bug tracking systems. They described a data model to store bug tracking data.

N. Jalbert and W. Weimer proposed a system that automatically deletes duplicate bug reports and saves time.time. They applied surface feature, texture semantics and graph clustering to detect duplicate bug reports [1]. There was eight percent reduction in the bug report caused due to filtering of duplicate bugs.

Many techniques that are being used for detecting duplicate bug reports are exploiting only natural language information. X. Wang et al. presented a new method which combines natural language information and execution information [1]. When a new bug is added, the method fetches the most similar bug to the new bug and compares it using natural language information. However, even if execution information helps the method to improve the accuracy of detecting duplicate bug reports, it leads to increase in workload of the method [1].

In the IEEE paper written by Tao Zhang and Byungjeong Lee, mainly focuses on models of the bug reports, classification techniques and duplication removal of bugs. The classification process is done by applying taxonomy algorithm. It is done by

1) Textual similarity
2) Similarity between bug reports
3) Classification. And it is provided with examples of various types of bugs
4) Bug rule

III. PROBLEM DEFINITION

After any software is developed, it has to go through the process of testing. This testing, helps us get rid of the various bugs or errors that may occur in the software due to various reasons. If ignored, it can result in damage to the system and further loss of business. Bug evaluation in various modules of project takes huge amount of time and effort of the developer. There are software projects that cannot deal with all the bug reports efficiently. Due to this there are many duplicate bug reports. Solving these duplicates may take up more time than necessary. Hence they need to be eliminated which is done with our system. In the present bug tracking systems some issues are not yet solved like:

1) Bug description and provision for extra information of bugs such as screen shots are not present properly,
2) Bug classifications are not in proper manner.

Previous studies indicate that more than 36 percent of bug reports are duplicates. In our system we are trying to increase the number of this percentage of removing the duplicates of the bug.

IV. EXISTING SYSTEM

With huge amount of software projects in the IT industry, a bug tracking system is needed. In this system there will be an admin team, database administrator team and the team working on the project. This team is ensured to fix these bugs. However, this system may sometimes yield inaccurate results

Diana Cardoz, Jennifer Coelho, and Samantha Joseph, "Duplication removal in bug tracking system," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 187-189, 2016.

because different bugs may have different levels of severity and complexity. There are various open source Bug Tracking Systems.

Some popular Bug Tracking Systems are:

1) *Bugzilla*: Bugzilla is a web based general purpose bug tracker and testing tool. It was developed originally by Mozilla foundation. The Stable release was 3.6.1 licensed by Mozilla public license. Bugzilla is used among others by Mozilla Foundation, Wikimedia Foundation, Web Kill, and GNOME [5].
2) *Mantis*: Mantis Bug Tracker is a free and open source web-based bug tracking system. This issue tracking system is written in PHP and works on different databases. It was released under the terms of the GNU General Public License version 2. Stable release is of 1.2.2. The common feature of MantisBT is to track software defects [5].
3) *Trac*: Trac is written in Python. Apart from issue tracking, it also provides wiki and integration to subversion. It provides project management features. Stable release is 0.1.2 [5].

V. IMPLEMENTATION METHODOLOGY

A bug tracking system is a software application that is designed to help keep track of reported software bugs in software development efforts. It may be regarded as a type of issue tracking system [3]. Generally, the extraction cost or complexity of bug tracking systems is considered to be high because of the difficulties in predicting the structure of bug reports. Even if they are crawled, some of the data is difficult to identify except common fields like priority or severity. In this system there will be a developer who works on an application or software projects, tester who tests the application and reports the bugs to the developer. Hence there is a cycle between them.

The system classifies bugs to identify duplicate reports. This classification technique involves identifying duplicates and invalid bug reports to give developers a better result of bug reports retrieval.

The overall process of extracting data from bug tracking system is as follows:

1. Bug reporting
2. Process engine
3. Duplication removal
4. Analytics and report

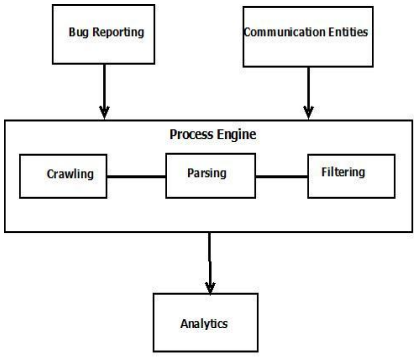

Fig. 1. Block diagram of the system.

A. Bug Reporting

In this module, tester reports the bug in the system. Bug tracking system has the facility to add the bug i.e. reporting of bug. Bugs can be added by any of the testers that have been assigned the project. Reporting of bug consist of following things –

- Bug type
- Priority
- Description
- Screen shot
- Location/ module of the system
- Reported by

B. Process Engine

Process engine is main module of bug tracking system. Process engine consist of Crawling by collecting bugs from various modules and converting to readable format and filtering modules in system. Filtering gets the parsed bug list and classifies it by using Naïve Bayes classifier and removes the duplicate. It gets the data in a readable format, and then it filters and sorts the data according to priority.

1) Crawling

Process engine consists of three different modules such as crawling, parsing and filtering. Crawling collects all the bugs that are entered into the system in different modules for parsing. The data that is crawled is obtained in a semi structured format.

2) Parsing

The crawled data from the previous process is given as is given as input to the parser. The semi structured data which is in JSON format is converted into a structured format. Thus this process converts the bug report is converted into a readable format.

3) Filtering

The output from the parser is given as input filtering.

In the filtering process we filter the bugs by removing duplicate copies of the same bug. To do this, we use the Naïve Bayes algorithm. In this algorithm, all the bugs that are entered into the system will be compared with each other and its matching percentage is calculated. If the match is more than a certain percentage that is set by the tester, then it is gathered that the bugs are similar hence only the recently added bug is retrieved and the others are deleted since they are considered as duplicates.

TABLE I. Bug description.

Bug status	Description
Open	It is available to be solved
In-progress	The bug is in progress of being solved so no one else can work on it
Resolved	Solution is provided to the bug
Closed	Solution is verified and the bug is removed

VI. CONCLUSION

After the software is tested, the bug reports are given to the system which classifies it into various categories depending on its type. The bugs that are entered are crawled out and parsed

Diana Cardoz, Jennifer Coelho, and Samantha Joseph, "Duplication removal in bug tracking system," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 187-189, 2016.

ISSN: 2454-1532

into a readable format. Then the bug reports are checked for duplicate bugs. The software decides whether a bug report is a duplicate, if it is, then it is removed in the filtering process. Thus the system reduces the time required for the developer to eliminate the bugs from the software.

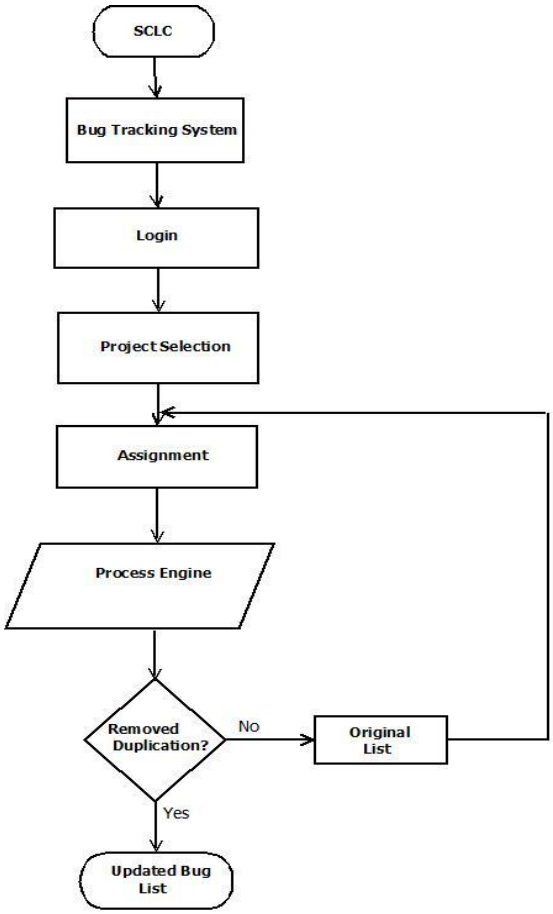

Fig. 1. Flowchart of the system.

REFERENCES

[1] Y. Yuk and W. Jung, "Comparison of extraction methods for bug tracking system analysis," in *Proceedings IEEE*, 2013.
[2] T. Zhang and B. Lee, "A bug rule based technique with feedback for classifying bug reports," *11th IEEE International Conference on Computer and Information Technology*, 2011.
[3] http://en.wikipedia.org/wiki/Bug_tracking_system
[4] http://en.wikipedia.org/wiki/Bugzilla
[5] http://www.softwaretestinghelp.com/popular-bug tracking-software/

Diana Cardoz, Jennifer Coelho, and Samantha Joseph, "Duplication removal in bug tracking system," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 187-189, 2016.