Real Time Bugs Tracking System

Harshini P
B.E – Computer Science & Engineering, Velammal Engineering College, Chennai,TamilNadu
Saranya S
B.E – Computer Science & Engineering,Velammal Engineering College, Chennai,TamilNadu
Mrs. R. Amirthavalli
Assitant Professor, Department of Computer Science & Engineering , Velammal Engineering College
Chennai,TamilNadu

ABSTRACT

Bug Tracking System is used to track the bugs and report it to the manager. Many Organizations and Companies assign projects to their employees. But sometimes they may fail to detect those bugs. This project is mainly developed to analyze and detect the bugs. In this paper, we designed an online compiler to detect and track the bugs and report it to the manager so as to manage time. It reduces the time and manual work of tracking the bugs, performs the regression testing. It focuses on easily detecting the bugs even though the employee is unaware of the concepts. It paves way for the manager to easily detect the performance of the employee and can hike them accordingly. There is no automatic system to detect the bugs. It is done manually only, which consumes much time and may fail to detect few bugs which the developer has no knowledge about it. In the proposed system, we have implemented an online compiler on which the manager can feed the code developed by the employees and to validate the code in easy manner. Each of the members in the organization has a separate login access to avoid confusion among them in performing their individual roles. By this system, Manager could identify who makes mistakes and hike them according to their performances randomly.

I. INTRODUCTION

As software projects turn out to be more expansive, complex and large. It comes to be more arduous to perform regression testing and generate the proper bug report or fix the bugs. Good or typical bug reports consist of a useful information instead of a bug description and summary information. A bug report contains all the details needed to document, report and remove problem arise in web applications. It is an approach that observes web application bugs and generates a bug report.

Bugs appearing during software development is a time-consuming and costly task. Many software projects use bug tracking systems to manage bug reporting, bug resolution, and bug archiving processes which is not so efficient.

The aim of this project is to assign a bug to a suitable developer within a short time interval can reduce the time and cost of the bug fixing process. This assignment process is known as bug triaging. Bug triaging is a time-consuming process since many developers are involved in software development and maintenance.

duplicates or otherwise invalid[1].

Since other works fail to test these cases , we have implemented an online compiler to track, report and store all the bugs that is hidden in the reports. It also checks information about fail, pass, and skip test cases as it filters the duplicate bugs.

II. EXISTING MODEL

Many organizations and companies assign projects to their employees to complete the jobs assigned, but sometimes they may fail to detect those bugs in which they do not have any knowledge about it. This project mainly developed to analyze and detect the bugs generated by the developers while doing projects.

In the existing system, the manager could not find out which employee makes mistake in detecting the bugs and it makes the manager to prepare the performance report chart.

There is no automatic system to detect the bugs. It is done manually only, which consumes much time and may fail to detect few bugs which the developer has no knowledge about it. To aid in finding appropriate developers, automatic bug triaging approaches have been proposed in the existing.

Bug report is treated as a vector of terms (words) and their counts. However, developers often use various terms to express the same meaning. The same term can also carry different meanings depending on the context.

These synonymous and polysemous words cannot be captured by VSM. Regression testing will perform to uncover new software bugs in the existing system after each modification or changes in software or application but requires more time to complete this task.

III. PROPOSED MODEL

Our objective is to establish a model of automated bug reporting system which performs regression testing automatically and generate the bug report. It also finds the duplicate reported bugs with the help of filtration process and after generating the bug report it will send that resultant report to the concerned person or the team whose mail id is given to the system.

Since all the testing processes are manual that increases time, cost and also do not give testing reports as much as efficient due to human error [4]. But with the help of the automation it will save time, cost and gives an effective testing report and there is no chance for human error as it is performed by tools and or scripts.

In the proposed system, we have implemented an online compiler on which the manager can feed the code

Bug reporting system grant to report, store the document, manage, assign, close and acquire the bug reports. Those bug reporting systems used by utmost open source software projects, grant end-users to insert bug report directly.

Manually bug report generation is a time taking process, but once a bug report generates then within a short time interval, assigning it to a suitable developer. Bug reporting system allows testers to report a track, describe, classify and comment on a bug report. To enhance the quality of software by granting the users to comments [3].

The primary contribution of this paper is to concentrate on automatically test the web application generates a bug report which contains appropriate fixer based on the affinity of the fixer to the topics.

The format of this paper is as follows. Section 2 introduces related work. In Section 3, we validate our model, paying deliberate attention to the proposed methodology of the project. In Section 4, we analyze the result and simulation of our model. Finally we conclude in Section 5 followed by references.

IV. Related Work

In current bug reporting systems, the tester does not generate effective bug report which contains all required information needed by developers. Without this information fixing of bug is a difficult task.

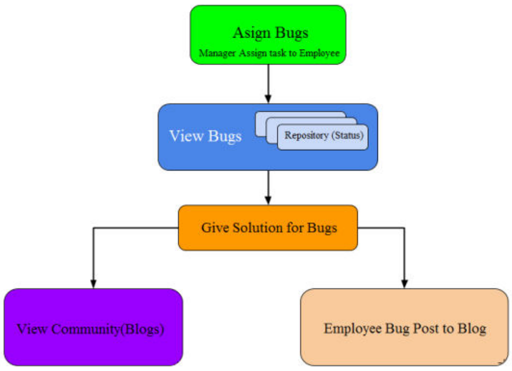
Akhilesh Babu Kolheri, K-Tameezuddin, Kalpna Gudikardula define four fundamental directions, to enhance the effectiveness of bug tracking system. In this model the developer prototype application, by capturing essential information from user tracks the bug effective and resolve the bug quickly [5].

Anvik et al. propose a model by using support vector machine and text categorization which automatically allot bug reports to an appropriate developer.

It is difficult to identify how many numbers of test cases are a pass, skip, fail and duplicate. Previous studies report that as many as 36% bug reports were developed by the employees and to validate the code in easy manner. Each of the members in the organization has a separate login access to avoid confusion among them in performing their individual roles. By this system, Manager could identify who makes mistakes and hike them according to their performances randomly. With the help of this automated bug reporting system the Manager can start the testing without the need for monitoring the system but still the bug report will be generated.

By filtering the duplicate bugs automatically we save the time of developers and they will focus on other testing activities. The duplicate bugs create problems for the developers because the bugs are more in numbers but most of them are duplicate which wastes the time and effort of the developers and makes debugging harder. But this duplicate bug finding feature helps developers to tackle that kind of problems. Since the developer gets the bug report generated by the automated bug reporting system which contains the properly mentioned bug details. Our model acts as a backbone of the testing framework. We find bugs from the web application by giving test cases as input and generated bug report according to given the test cases. For finding bug reports we make a framework. Testers write their test cases for their respective web applications and make changes in a framework according to their need and web application flow.

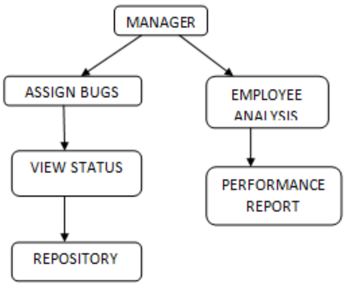
V. Architecture Diagram

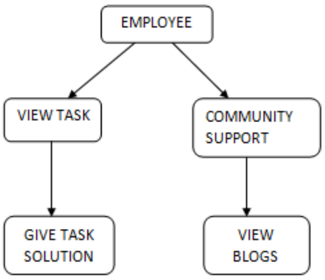


A bug reporter can fill when the reporter submits bug report. These fields include the product, component, reporter, priority, severity, OS, version, and platform fields. We have implemented an online compiler. In which the manager can feed the code developed by the employees for validation. By this system, Manager could identify who makes mistakes and hike them according to the performances randomly.

VI. Data Flow Diagram

LEVEL 1 & 2:





VII. Result And Simulation

The modules in this proposed model are the following:

- Employee Login

- Manager Login
- Assign Bugs
- Search Bugs

In the Team member registration page, each member of the team creates their own login using their details. Upon registration, the team member has the access to login with registered password.

Employee Login: Each employee of the organization has a separate login and once they sign up, they have access to use their login from then. The manager will assign the jobs to the employee and the employee should complete their jobs done. But sometimes they may fail to detect those bugs. This project mainly developed to analyze and detect the bugs.

Manager Login: Each Manager has a separate login and they perform the validation of bugs. By using an online compiler to detect the bugs, the manager could determine the hike of the employee accordingly. Using this compiler, the manager could find which employee understands the concepts and make mistakes s they don't have the complete knowledge about it.

Assign Bugs: Process of assigning a bug to the most appropriate developer to fix. We have implemented an online compiler to validate the code in easy manner. By this method, we could reduce the time to evaluate the bugs by excluding manually.

Search Bugs: The online compiler is used to detect the bugs that are present in the job that is assigned. There may be bugs and sometimes it may not be cleared or reported but using this compiler we could avoid such bugs that are hidden.

These are easily caught by this compiler and it is found and rectified as soon as it finds the bugs.

The data flow sheet (DFD) is one amongst the foremost necessary modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system. DFD shows how the data moves through the system and the way it's changed by a series of transformations. It is a graphical technique that depicts info flow and therefore the transformations that are applied as knowledge moves from input to output. It considers a document as a random mixture of latent topics, where a topic is a random mixture of terms.One or few features can be only taken into consideration.Lower accuracy and more complex in the existing system.

VIII. CONCLUSION

To detect the assigning bugs on project. By using an online compiler to detect the bugs, the manager could determine the hike of the employee accordingly. This model is capable to reduce the human effort and time.

This proposed model will help to maintain the original code and code duplication which is crucial activity in test automation.

Thus, this model is capable to reduce the maintenance cost, development cost, human effort and time to perform regression testing by filtering the duplicate bugs.

This proposed model will help to maintain the original code and code duplication which is crucial activity in test automation. Thus, this model is capable to reduce the maintenance cost, development cost, human effort and time to perform regression testing by filtering the duplicate bugs.

Each of the members in the organization has a separate login access to avoid confusion among them in performing their individual roles. By this system, Manager could identify who makes mistakes and hike them according to their performances randomly.

We can reduce the time to complete the works and the bugs are easily detected and are rectified. It is an efficient method to detect the bugs and the rectifications are made immediately.

REFERENCES

[1] J. Anvik, L. Hiew, and G.C. Murphy. Who should fix this bug? In International Conference on Software Engineering (ICSEce), pages 361-370, 2006.

[2] P. Hooimeijer and W. Weimer. Modeling bug report quality. In Automated software engineering, pages 34-43, 2007.

[3] E.S.Raymond.The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary.Inf.Res.,6(4),2001.

[4] C.V. Ramamoorthy and W.-T.T sai.Advances in software engineering IEEE Computer, 29(10):47-58,1996.

[5] S. Kim and J. E. James Whitehead. How long did it take to fix bugs? In International Workshop on Mining Software Repositories, pages 173-174, 2006.

[6] Sandeep Singh. Analysis of Bug Tracking Tools. International Journal of Scientific & Engineering Research, July 2013.

[7] Nicholas Jalbert and Westley Weimer. Automated Duplicate Detection for Bug Tracking Systems. In International Conference on Dependable Systems & Network, pages 52-61, 2008.