International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.05, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Automatic Developer Recommendation for Bug Handling and Resolution

Akshay Jadhav[1], Kunal Nangare[2], Pooja Kadam[3] ,Prof. P.N.Sharma[4]

[1,2,3]B.E Student,Computer Department, NMIET, Pune, (India)

ABSTRACT

Bugs can occurred while coding for Software Development. Some bugs are solved by programmer but some bugs are unfamiliar and difficult to resolve by the programmer. When encountering unfamiliar thrown exceptions, programmers often refer to online forum threads to understand how to fix them. Although some general search engines are available and some research tools are proposed, they are in sufficient to recommend for resolving the bugs. Some online forums are available but those are the more time consuming and the given solution on those sites or forums not have accuracy and ratings for ensuring the correct or suitable solution. In this Paper, we proposed the new automatic developer recommendation system for handling the bugs and resolve them.

Keyword: Topic Model , Bug Triage, Developer,Content-Boosted Collaborative Filtering ,Security and reliability.

I INTRODUCTION

When encountering unfamiliar thrown exception programmer often refers to online forum threads (e.g. Stack overflow)to understand how to fix them. Although some general search engines are available and some research tools are proposed they are insufficient to find the exception related bugs, To finding the solution on that web site it is time consuming process. In this Project overcome pervious problem by creating such type of web site for easily finding the program related bug and increasing the code accuracy as well as it`s save the manual work[1]**.**

A typical software repository in managing software bug and storing details of bugs plays an important role. In software development process software bugs are inevitable and fixing bugs is expensive. Software companies spend over 45 percent of cost in fixing bugs. In IT Company, projects deploy with bug repositories means a bug tracking systems to support information collection and to assist developers to handle bugs. A bug repository is use for maintained bug as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing. Fault prediction, bug localization, and reopened bug analysis for these tasks a bug repository provides solutions, bug reports in a bug repository are called bug data. We address the issues of data reduction for bug triage, i.e., how to reduce the bug data to save the labour cost of developers and improve the quality to facilitate the process of bug triage[3]

II REVIEW OF CONFERENCE /JOURNAL PAPERS SUPPORTING PROJECT IDEA

1. Accurate Developer Recommendation for Bug Resolution. Xin Xia, David Lo, Xinyu Wang, and Bo Zhou,2012.

DESCRIPTION :-

In this paper, we propose a new and accurate method named Dev-Rec for the developer recommendation problem. Dev-Rec is a composite method which performs two kinds of analysis: bug reports based analysis (BR-Based analysis),and developer based analysis(D-Based analysis).In the BR Based analysis ,we characterize a new bug report based on past bug reports that are similar to it. Appropriate developers of the new bug report are found investigating the developers of similar bug reports appearing in the past. In the D-Based analysis, we compute the affinity of each developer to a bug report based on the characteristics of bug reports that have been axed by the developer before. This affinity is then used to end a set of developers that are close to a new bug report.[1]

2. EXPSOL: Recommending Online Threads for Exception-related Bug Reports Xia oning Liu, Beijun Shen, Hao Zhong, Jian gang Zhu,2016

DESCRIPTION:-

An exception-related bug is a kind of program bug which causes exceptions. During software maintenance ,when programmers repair exception-related bugs, they typically analyze thrown exceptions to understand the root causes of such bugs .When encountering unfamiliar thrown exceptions, programmers often refer to online forum threads (e.g. Stack Overflow) to understand how to x them. Although some general search engines are available and some research tools are proposed, they are insufficient to recommend threads for exception related bugs from large-scale online resources. In this paper, we propose an approach ,named EXPSOL ,which recommends online threads as solutions for a newly reported exception related bug with a model trained by support vector machines. We conduct two evaluations on thousands of threads from Stack Over flow and disuses from GitHub. The results of ours evaluation show the significance of our internal features and highlight the importance of integrating different features. The results of our second evaluation show that, EXPSOL.[2]

3 A time based approach to automatic bug report assignment"

DESCRIPTION:-

The brisk escalation in scale of software systems has made bug triaging an imperative step in bug fixing process. A huge amount of bug reports is submitted daily to bug tracking repositories.

Although this practice assists in building a reliable and error-free software product but handling a large amount of work becomes challenging. Bug assignment, an essential step in bug triaging, is the process of designating a suitable developer for the bug report who could make code changes in order to fix the bug. Various approaches ranging from semi to fully automatic bug assignment are proposed in literature. These approaches are mostly based on machine learning and information retrieval techniques.

III MATHEMATICAL MODEL

• Hunt resoled bug in case developer miscarried the bug solution ,I t get assigned to expert developer to resolve the error.

• Let S be a system that describes central system with big data handler. S={..}

• Identify input as I S={I,..} Let I ={i1,i2,i3,..id} The input will be problem statement ie. Bug description and bug details.

• Identify output as O S ={I,O,} O= The receiver will receive resolved solution for critical bug.

• Identify the processes as P S={I,O,P,..} P={E,D} E={Bug Description, Bug Type, Bug Language} D={Resolved Bug,Bug

Description}.

• Identify failure cases as F S={I,O,P,F,.} F=Failure occurs when the system fails to expound panacea of the bug.

• Identify success as s. S={I,O,P,F,s,} s=When system succeed to preserve the solution on bugs.

.• Identify the initial condition as Ic S={I,O,P,F,s,Ic,} Ic=Developer should be authenticated and authorized user.

A. ARCHITECTURAL DESIGN:-

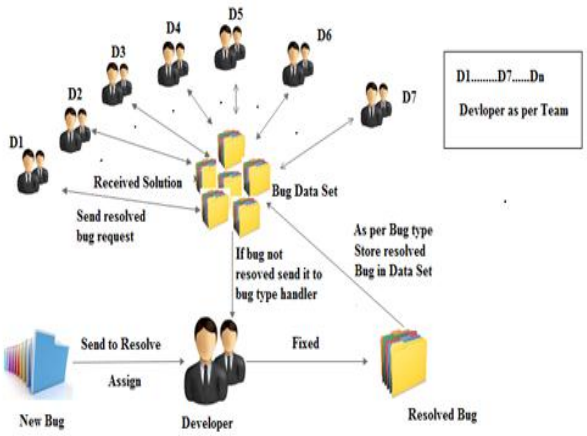

Figure :-System Architecture.

DEVEPLOVER

When developer fined the error he fixes that error and store the solution in proposed system. Another developer searches the solution of same error which is not resolve by him. When he find the solution he use that solution.

SYSTEM:- When developer search for solution of particular error/ bug using keyword system sort the solution according to developer requirements and rating. Assign the bug to expert developer using the dataset.

B. **ALGORITHM:-**

Content-Boosted Collaborative Filtering Algorithm :-

In content-boosted collaborative filtering, we first create a pseudo user-ratings vector for every user u in the database. The pseudo user-ratings vector, v_u, consists of the item ratings provided by the user u, where available, and those predicted by the content-based predictor otherwise.

$$v_{u,i} = \begin{cases} r_{u,i} : \text{ if user } u \text{ rated item } i \\ c_{u,i} : \text{otherwise} \end{cases}$$

In the above equation $r_{u,\,i}$ denotes the actual rating provided by user u for item i, while $c_{u,\,i}$ is the rating predicted by the pure content-based system

CBCF method combines a CF algorithm and CBF features to improve prediction performance over pure CBF and pure CF algorithms by overcoming the naive trainee and naive case problems. The main idea of the CBCF algorithm is that a pseudo trainee-ratings is constructed through a CBF predictor based on original trainee ratings data, and then a CF method is used to make a final prediction based on the pseudo training-ratings matrix. In the CBCF method, producing the pseudo trainee ratings matrix through a CBF predictor and making a final prediction using a CF method are the two core steps of the CBCF. Content-based prediction algorithm represents the target trainees rating as an n-

dimensional vector.

IV CONCLUSIONS

This paper try to overcome the bug solution problem by creating the website for easily finding solution on the program related bug and increase the code accuracy as well as save the time in manual work by using the Content Boosted Collaborative Filtering Algorithm.

REFERENCES

1) R. Shokripour, J. Anvik, Z. M. Kasirun, and S. Zamani. ”A time based approach to automatic bug report assignment”. Journal of Systems and Software, 102:109122, 2015..
2) X.Xia,D.Lo,X.Wang, and B.Zhou. ”Accurate developer recommendation for bug resolution”. In Reverse Engineering (WCRE), 2013 20th Working Conference on, pages 7281. IEEE, 2013.

3) Xiaoning Liu, Beijun Shen, Hao Zhong, Jiangang Zhu, "EXPSOL: Recommending Online Threads for Exception-related Bug Reports", 2016 23rd Asia-Pacific Software Engineering Conference

4) Anh Tuan Nguyen, Tung Thanh Nguyen, Tien N. Nguyen, David Lo, Chengnian Sun, Duplicate Bug Report Detection with a Combination of Information Retrieval and Topic Modeling, 2012

5) W. Wu, W. Zhang, Y. Yang, and Q. Wang." Drex: Developer recommendation with k-nearest-neighbor search and expert is e ranking". In Software Engineering Conference (APSEC), 2011 18 Asia Pacic, pages 389396. IEEE, 2011.

6) J. Anvik, L. Hiew, and G. Murphy. "Who should x this bug?" In Proceedings of the 28th international conference on Software engineering, pages361370,2010.

7) J. A. Jones and M. J. Harrold, "Empirical evaluation of the tarantula automatic fault-localization technique," in Proc. 20th IEEE/ACM Int. Conf. Autom. Softw. Eng., New York, NY, USA, 2005, pp. 273–282.

8) S. Hangal and M. S. Lam, "Tracking down software bugs using auto-matic anomaly detection," in Proc. 24th Int. Conf. Softw. Eng., 2002, pp. 291–301

9) A. G. S¨ anchez, "Automatic error detection techniques based on dynamic invariants," Master's thesis, Dept. Softw. Technol., Delft Univ

10) C. Flanagan and S. N. Freund, "Type-based race detection for java," in Proc. ACM SIGPLAN 2000 Conf. Program. Lang. Des. Implement., Jun.2000, pp. 219–232

11) M. D. Ernst, J. Cockrell, W. G. Griswold, and D. Notkin, "Dynamicall discovering likely program invariants to support program evolution," I Trans. Softw. Eng., vol. 27, no. 2, pp. 99–123, Feb. 2001.

.