RESEARCH ARTICLE OPEN ACCESS

A Survey on Bug Triaging With Software Data Reduction

KrishnaVeni V.L[1], Karthik.M [2]
M.Tech Student [1], Asst.Professor [2]
Department of Computer Science and Engineering
Mohandas College of Engineering & Technology, Anad
Trivandrum –India

ABSTRACT
Software companies are dealing with large number of software bugs nowadays. It is well expensive and unavoidable too. The traiging process is nothing but to assign effective and proper developer for bug fixing. various techniques are used for this process earlier. Bug triaging and historical data set are well maintained.New methods are being used each time. Human traiging was not a time worth process and was not sucessful in the case of proper assignment of developers .Nowadays the techniquespay attention towards correct assignment of the developer to the respective bug automatically. It will be an effective method for the companies who faces the challenge of assigning the developer.

Keywords:- Mining software repositories, application of data pre-processing, data management in bug repositories, bug data reduction, feature selection, instance selection, bug triage.

I. INTRODUCTION

Many software companies spend almost half of their project money in fixing the bugs. Large software projects have bug repository that holds all the information related to bugs and is well maintained for further processing. In bug repository, each software bug has a bug report and is also known by bug data. The bug report consists of textual information of the bug and the updates on the basis of the status of bug fixing, which is available in historical bug data set. Traditional software analysis is not fully suitable for the large-scale and complex data in software repositories. Data mining has been introduced to the technically developing environment as a promising means to handle the software data. By using the data mining techniques, mining software repositories can uncover interesting and hidden information of the software repositories and can also solve the real world software problems.

Due to the huge amount of daily reported bugs, the bug reports are increasing and the scaling up in the repository is being high as well. Noisy bugs and redundant bugs are degrading the quality of bug reports which is held in the repositories. Bug triage is one of the least time taken procedure in handling of bugs in software projects and of course the most proper way. .Manual bug triaging by a human bug triager is a vast process and error-zone because of the arrival of large number of bug data and lack of developers who has an accurate knowledge of the bug to be fixed with.

In some former methods, if a bug report is formed or a bug occured, then a human triager assigns this bug to a developer, who tries to fix this bug. This developer is recorded in an item assigned-to in historical bug dataset. If the previously assigned developer was unable fix this bug then the former will changed to a new one.The method of assigning a proper developer for fixing the bug is known as bug triaging.

II. LITERATURE SURVEY

In [1] they discuss about the web script crashes and error as well the malfunctioning of web page service, along with that it is a serious issue too. A derived solution for that is a ***dynamic test generation technique*** for the domain of dynamic Web applications which utilizes both combined, concrete and symbolic execution and explicit-state model checking is done.

The technique generates tests automatically then runs and captures logical constraints on inputs, minimizes the conditions on the inputs to failing tests .Thus the resulting bug reports are small and useful in finding and fixing the faults. An Apollo tool with PHP Programming is specified.

In [2] they mentioned that Bug triaging is an error-prone, tedious and time consuming task and also focused with "Revisiting Bug Triage and Resolution Practices" as a whole. The insights into several areas were provided by the means of interest such as triage practices, review and approval processes.The root cause analysis of bug reassignments and the occurence of reopens in open

source software projects,also the recommendations are done for the improvements/redesign of bug tracking systems.

In this paper they studied about bug triaging and fixing practices, contains with bug reassignments and reopening.The ***main approach*** was done as follows,Since the length of the responses changes from phrases to paragraphs, The responses were printed and splitted into individual statements written on index cards. They also planned to perform a ***card sort technique*** so that to organize interview responses into hierarchies thereby to reduce a higher level of abstraction and identify the common themes in the participant's feedback.

In [3], they introduced a ***graph model based on Markov chains***, which generally captures a *bug tossing history*. data.This model has different desirable qualities such as ***Firstly***, it reveals developer networks which can be used to discover the team structures and then find suitable experts for a new job.***Next*** it will help to better assign developers to the bug reports.

The experiments were done with 445,000 bug reports,the model reduced tossing events, by up to 72% ,It also increased the prediction accuracy up to 23 percentage points compared to the traditional bug triaging techniques.This model captures the tossing probabilities between developers from the tossing history which is available in bug tracking systems.

In [4] the research shows that the optimizing recommendation accuracy problem and it proposes a solution which is essentially an instance of content-based recommendation popularly abbrevated as CBR.Onlythe earlier could be solved by the developer of same kind and the process will be slow.
The ***techniques*** used here are:-

First, reformulate the problem as an optimization problem of both accuracy and cost then *Second*, they adopt a content- boosted collaborative filtering– CBCF that is combining the present CBR with a collaborative filtering recommender (CF), and modify the recommendation quality of either approach alone.

In[5], a semi-supervised text classification approach which is far better when compared to the previous methods,which were used for bug triage to avoid the deficiency of labeled bug reports is adopted here.The combination of naive bayes classifier and the expectation maximization has considered
, both labeled and unlabeled bug reports.
The ***techniques*** offer some contributions such as bug triaging by semi-supervised approach and weight recommendation list is maintained.

The bug traiging by *semi supervised approach*, unlabeled bug reports were added to the existing supervised approach to get rid of the deficiency of labeled bug reports.The weighted recommendation list is maintained,weighted recommendation list for augmentation of the semi-supervised approach using the probabilistic labels of unlabeled bug reports is been provided. According to that weighted recommendation list, The accuracy for the semisupervised technique could be improved in a better way.

In[6] A semi-supervised text classification approach for bug developer assignment, to avoid the deficiency of labeled bug reports as per the previous supervised approaches. Latest technique combines Naive Bayes classifier and expectation- maximization consider the use of both labeled and unlabeled bug reports.

It trains a classifier with a fraction of labeled bug reports and iteratively labels numerous unlabeled bug reports.A weighted recommendation list to improve the performance by updating the weights of multiple developers in training the classifier is well maintained.Classification accuracy is much higher.

In[7]They proposed the machine learning techniques to assist in bug triage following the text categorization to predict the developer that should work on the bug according to the bug's description. The approach is introduced on a collection of 15,859 bug reports from a large scale project. The evaluation shows that the prototype using a supervised Bayesian learning, could accurately predict almost 30% of the report assignments to the developers.

The system was easily adjustable with the current bug-handling procedures to decrease the resources.The mis- predictions is handled in a easier way by the assigner ,"bouncing" them to a dedicated triager for a human inspection as well classification. The classification accuracy can achieve was significantly lighten the load which was face by the developer according to the present situation.

In [8] they addressed the problem of data reduction for bug triage, which show the light on, how to reduce the scale and improve the quality of bug data. The bug triage concentrates to predict the developers who can fix the bugs, they followed the existing work to remove unfixed bug reports. For e.g., the new bug reports or will-not-fix bug reports. Thus considered, only the bug reports those are fixed and redundant based on the bug report status in the item list. The combined usage of instance selection with feature selection to simultaneously reduce data scale on the behalf of bug dimension and the word

dimension is mentioned here. To acquire the order of applying both, the extraction of attributes from the historical bug data sets is considered and build a predictive model for a new bug data set.

Title	Method	Advantages	Disadvantages
Finding Bugs In Web Applications Using Dynamic Test Generation And	Dynamic Test Generation Technique.	1.Generates-tests automatically. 2.Minimizes,conditions	Limited tracking in native methods of input parameters through the database.
Revisiting Bug Triage and Resolution	Bugzilla approach.	Dynamic bug reopening and reassignments are possible.	Teadious and lengthier task.
Improving Bug Triage with Bug Tossing Graphs	Markov-chain modeling.	1.Reduced-tossing events. 2. Model increased the prediction accuracy by upto 23%.	1. Systems examined is not representative. 2.Developer retirement information is hidden.
COSTRIAGE: Bug Reporting System.	Cost aware triage with CBCF and CF.	Reduce the sparseness and enhance the quality of CBCF.	Difficulty in predicting missing values for profiles.
Collaborative Bug Triaging using Textual Similarities and Change Set	Colloborative Prototypical technique.	Providing Context for a more Informed-Decision in Collaborative	Doesn't provide an effective starting point to address a bug report.
Automatic Bug Triage using Semi-Supervised Text Classication.	Naïve-baye's,expectation maximization technique.	Avoid the deficiency of Labeled- reports in existing supervised approaches.	1.Many-to-One correspondence is not possible. 2.Co-training for bug triage is not done here.
Automatic bug triage using text categorization	Supervised-machine learning using a naive Bayes classifier.	Classification accuracy, significantly lighten the load that the triagers face.	The verification is done only by human triager.
Towards Effective Bug Traige With Software Data Reduction Techniques,	Instance and feature selection approach	1.Automatic assigning of developer. 2.Low time and cost	pay efforts to find out the potential relationship between the attributes of bug data sets and the reduction orders.

Table.1.Comparison of different Bug Traiging Techniques

III. CONCLUSION

One of the expensive step in software maintenance is Bug Triaging, mainly when it comes to the matter of labor and time cost. The recent technique aims to form reduced and high-quality bug data in software development and thereby maintenance. The data processing techniques like instance selection and feature selection are used for data reduction.

Former techniques were tossing graphs, collaborative approach and semi-supervised learning. The latter system is useful for any open source projects that generate huge and large amount of bug data. Several software companies' doe's projects like banking, food chain management can consider the application of the current technique which is more beneficial and effective.

REFERENCES

[1] S. Artzi, A. Kie_ zun, J. Dolby, F. Tip, D. Dig, A.Paradkar, and M. D. Ernst, "Finding bugs in web applications using dynamic test generation and explicit-state model checking," IEEE Softw., vol. 36,no. 4, pp. 474–494, Jul./Aug 2010.

[2] Olga Baysal, Reid Holmes, and Michael W. Godfrey David R. Cheriton ,"Revisiting Bug Triage and Resolution Practices", School of Computer Science University of Waterloo Waterloo, ON,july 2009.

[3] G. Jeong, S. Kim, and T. Zimmermann, "Improving training set reduction for bug triage," in Proc. 35th 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.

[4] J. W. Park, M. W. Lee, J. Kim, S. W. Hwang, and S. Kim,"Costriage: A cost-aware triage algorithm for bug reporting systems," in Proc. 25th Conf. Artif. Intell., Aug. 2011, pp. 139–144.

[5] Katja Kevic, Sebastian C. Muller, Thomas Fritz, and Harald C. Gall "Collaborative Bug Triaging using Textual Similarities and Change Set Analysis", Department of Informatics University of Zurich, Switzerland ,june 2010 katja.kevic@uzh.ch {smueller, fritz, gall}@ifi.uzh.ch.

[6] 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.

[7] D.Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng.,Jun. 2004, pp. 92–97.

[8] Towards Effective Bug Traige With Software Data Reduction Techniques,Jifeng Xuan, He Jiang, Member, IEEE, Yan Hu, Zhilei Ren, Weiqin Zou,Zhongxuan Luo, and Xindong Wu, Fellow, IEEE,IEEE transaction on Dta Knowledge and Engineering.

[9] X. Wang, L. Zhang, T. Xie, J. Anvik, and J. Sun, "An approach to detecting duplicate bug reports using natural language and execution information," in Proc.30th Int. Conf. Softw. Eng., May 2008,pp. 461–470.

[10] J. Xuan, H. Jiang, Z. Ren, and Z. Luo, "Solving the large scale next release problem with a backbone based multilevel algorithm,"IEEE Trans. Softw. Eng., vol. 38, no. 5, pp. 1195–1212, Sept./Oct.2012.

[11] J. Xuan, H. Jiang, Z. Ren, and W. Zou, "Developer prioritization in bug repositories," in Proc. 34th Int. Conf. Softw. Eng., 2012, pp. 25–35.

[12] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," in Proc. Int. Conf. Mach. Learn., 1997, pp. 412–420.

[13] H. Zhang, L. Gong, and S. Versteeg, "Predicting bug-Fixing time:An empirical study of commercial software projects," in Proc. 35th Int. Conf. Software Eng.,May 2013, pp. 1042–1051.

[14] W. Zou, Y. Hu, J. Xuan, and H. Jiang, "Towards Annu. IEEE Int. Comput. Soft. Appl. Conf., Jul. 2011, pp. 576–581.