

A SURVEY ON VARIOUS TECHNIQUES FOR BUG TRIAGE WITH BUG DATA REDUCTION

¹P Meena Kumari, ²Dr K E Balachandrudu
¹Research Scholare, ²Professor & Principal
¹Computer Science and Engineering,
¹Research Scholar, Shri JTT University, Rajasthan

Abstract : Recent days many software societies are emerging with exposed source projects are expenses over 40% of Effort in keeping and fixing with software bugs like Net beans, Eclipse, Mozilla, VirtualBox, Boost, CryEngine and so on. The procedure of fixing bug is bug triage, which aims to properly assign a developer to a new bug. Software corporations spend most of their cost in dealing with these bugs. Bug triage aim is assign a skilled developer to fix new bug. To reduce time and cost of bug triaging, we present an automatic approach to predict a developer with applicable knowledge to solve the new coming report. Basically a bug repository is a software repository for storing repots of bugs and Description of bugs. According to the literature survey in traditional software development new bugs are triaged by humans, but due to the large number of daily reported bugs and lack of expert developers manual bug triage is tedious task to the organizations in way of time , cost and less accuracy for bug fixing.

IndexTerms - Bug, Bug triage, instance selection; software quality4r

I. INTRODUCTION

Bug fixing is a significant and time-consuming process in software maintenance. Large no of software projects have bug repository. A bug repository plays an important role in managing software bugs. A Bug Repository is a software repository which contains all the information related to software bugs. Many open source software projects have an open bug repository that allows both developers and users to submit defects or issues in the software, suggest possible enhancements, and comment on existing bug reports. A software bug is a problem, which causes a computer program or system to crash or produce invalid output or to behave unintended way. The work of managing bugs increases the cost of software quality maintenance. Many software projects use a bug tracking system to store and manage bugs submitted by users, including end users, testers, and developers. . Software companies spend over 45 percent 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 maintained as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing. Bug triage, an important step for bug fixing, is to assign a new bug to a relevant developer for further handling. Based on the bug tracking system, the developers can easily search and maintain all the existing bugs. A human triager assigns this bug to a developer, who will try to fix this bug. If the assigned developer cannot fix this bug, then new developer is assigned for fixing that bug. This process of assigning a correct potential developer to fix a new bug is called bug triage.

II. LITERATURE SURVEY

In any software organization IT Industry, dealing with bugs is a tedious task. When a programmer's expecting behavior is not matching with actual behavior, an occurrence needs to be raised. So, these bugs must be assigned to expert who can resolve the bugs in a limited time. Now a day's Open source software projects often provide their programmer and developer communities with an open bug repository for reporting the software bugs in order to be resolved by developers.

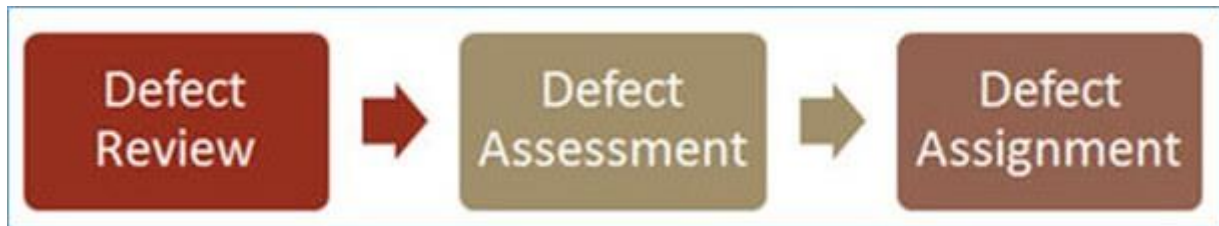
In this paper, we will review some of the how the bugs are reported and fixed by software experts, and some of these techniques are further classified. In practice, due to the frequent changes of software development teams, it is difficult to identify the correct developer in manual triage. Taking Eclipse 2 as an example, Anvik reports that an average of 37 bugs per day are submitted to the bug tracking system and 3 person-hours per day are required for the manual triage; the empirical study by Jeong et al. To solve these problems, some machine learning algorithms are employed to conduct automatic bug triage. Most of the bug triage approaches are based on text categorization. However, these approaches suffer from two problems. On one hand, due to the large number of bugs, it is necessary to collect large-scale training sets of bugs to obtain good results for bug triage. There are two

challenges related to bug data that may affect the effective use of bug repositories in software development tasks, namely the large scale and the low quality.

III. OVERVIEW

What is 'Defect Triage'? Defect triage is a process where each bug is prioritized based on its severity, frequency, risk, etc. Triage term is used in the Software testing / QA to define the severity and priority of new defects. In a bug repository, a bug is maintained as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing.

Why do we need to have 'Defect Triage'? The goal of Bug Triage is to evaluate, prioritize and assign the resolution of defects. The team needs to validate severities of the defect, make changes as per need, finalize resolution of the defects, and assign resources. Mainly used in agile project management.



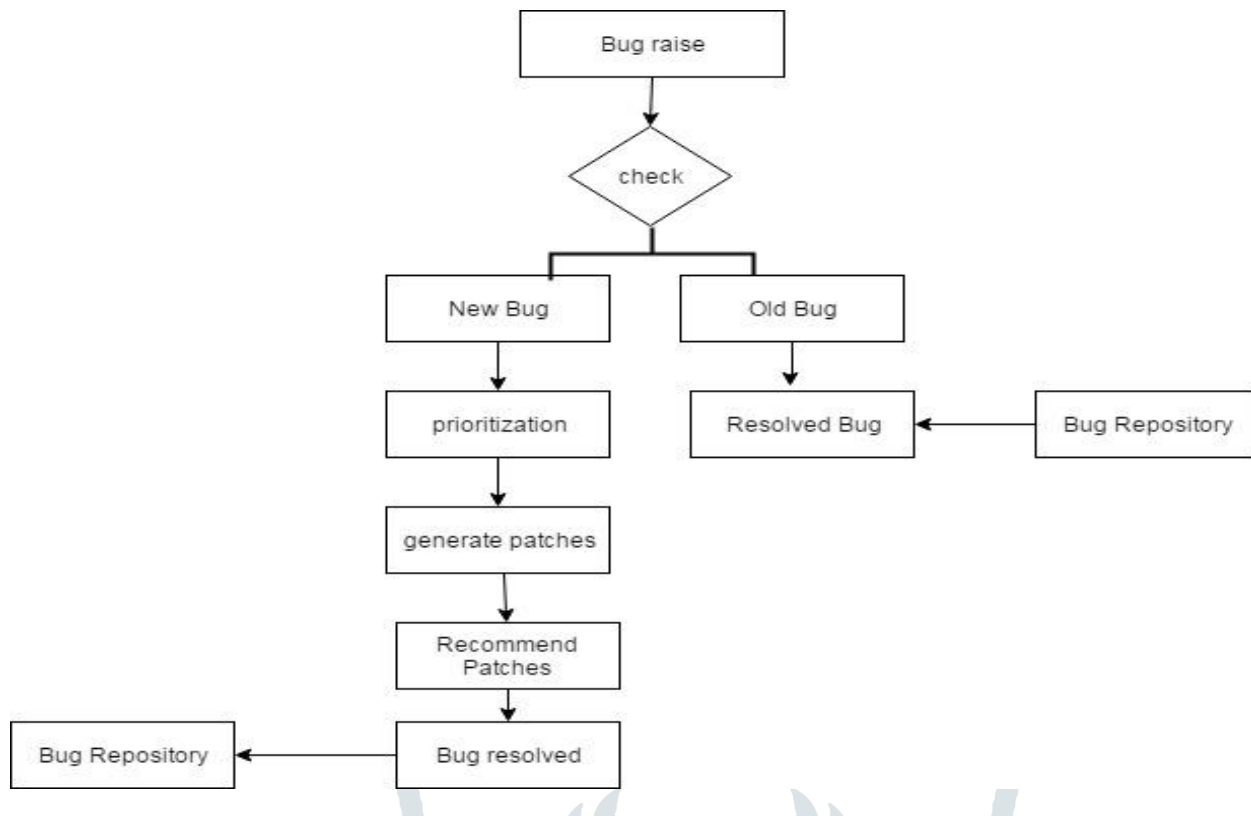
What is Bug/Defect Triage? Once the test execution of project is initiated by tester and started reporting the bugs then the bug triage meeting should be held based on the bugs reported by the tester. The attendee for Bug Triage meeting should be QA Lead, Development Lead and Project Manager. Test lead should make easy bug triage meeting with the bug tracking system; however the prerequisite for Bug Triage meeting is bugs are logged under Bug Tracking System for triage.

The Bug Triage meeting should be facilitated by QA lead and this meeting should be conducted in testing phase of the SDLC. The frequency of the meeting is vary from projects to projects and it is to be decided based on the how many bugs are logged by the testers and how many new bugs are being logged by the testing team. The aim of this meeting is to take action on most important or critical bugs first and not concentrated much to fix non-important bugs first. Before the meeting the QA lead should send the report to all attendees of the meeting which are newly reported or not yet triaged bugs. Therefore the triage meeting attendees will get the current status of the non-triaged defects and based on this input bug triage meeting can be started. In this meeting attendees also discuss about the status of the bugs triaged in the prior meetings or if any queries facing by team members. They can also re-evaluate the severity and priority of earlier evaluated defects as well.

In the **Bug triage meeting** QA lead should have access laptop or desktop access to defect tracking system. Each and every defect is discussed in this meeting and decides the priority and severity of the defect. All discussion should be captured and add appropriate comments in the defects. If any defect is more complex then developer can also attend this meeting to understand the risk and complexity associated with the defect. The meetings attendees are decide which defect is to address first and which can wait till for future release.

What should be involved while triaging a bug:

Check if the enough information is provided in the defect by tester in such a way that developer can understand the defect without any problem. If not then triage team can reassign the defect to reporter and ask for the additional information which helps out to understand the defect. Check if defect is logged under correct project and module. Many companies are used to review and assign the priority to defect, however good triage process is to assign or modify the defect severity as well. Check if appropriate "Severity" and "Priority" fields are marked for the bug. Bug Triage Meetings are project meetings in which open bugs are divided into categories: bugs to fix now, bugs to fix later, and bugs we'll never fix.



ROLES & RESPONSIBILITIES of individuals in Bug Triage Meeting

1. Project Manager

- Assists in the prioritization of the defects
- Sends out meeting minutes when appropriate
- Tracks issues list
- Discusses the delivery date of next iteration to QA.

2. Product Manager

- Assists in the prioritization of the defects.

3. Test Lead (QA Lead)

- Calls the bug triage meeting
- Submits a defect report to the CFT, prior to the start of the meeting
- Assists in the prioritization/severity of the defects
- Assists in determining Root Cause of defect
- Manages defects in CQ
- Distributes updated defect report, capturing the notes from the bug triage meeting.

4. Development Lead and/or Developer

- Assists in the prioritization of the defects (don't think development sets priority, they should set severity).
- Explains the level of complexity and the risk associated with each defect being presented at the bug triage meeting
- Assigns the defects to the appropriate developer
- Updates Resolution and development notes fields in CQ
- Assists in determining Root Cause of defect
- Discusses the delivery date of next iteration to QA.

5. User Transition Manager

- Ensures that appropriate User Representatives are invited to the bug triage meeting
- Assists in the prioritization of the defects

6. User Representative

- Assists in the prioritization of the defects

IV. CONCLUSION

In this work we are proposing automatic patch generation and recommendation. This new bug fix will minimize the time and cost for open source project. First work of combining feature selection with instance selection to reduce the training set for the bug triage problem. Based on our setup, 70% of words and 50% of bug reports are removed. The experimental results show that the combinations of CHI and ICF can achieve better accuracy rates than that without the training set reduction. The results also indicate that the combination, ICF→CHI, is a good choice for the training set reduction.

REFERENCES

- [1] S. Kim, H. Zhang, R. Wu, and L. Gong, "Dealing with noise in defect prediction," in Proc. 32nd ACM/IEEE Int. Conf. Softw.Eng., May 2010, pp. 481–490.
- [2] J. Xuan, H. Jiang, Z. Ren, J. Yan, and Z. Luo, "Automatic bug triage using semi-supervised text classification," Proc. Intl. Conf. Software Engineering & Knowledge Engineering (SEKE 10), Jul. 2010, pp. 209-214.
- [3] 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
- [4] 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.
- [5] 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.
- [6] D. Čubranić and G. C. Murphy, "Automatic bug triage using text categorization," Proc. Intl. Conf. Software Engineering & Knowledge Engineering (SEKE 04), Jun. 2004, pp. 92-97.
- [7] G. Jeong, S. Kim, and T. Zimmermann, "Improving bug triage with tossing graphs," Proc. Joint Meeting European Software Engineering Conf. & ACM SIGSOFT Symp. Foundations of Software Engineering (ESEC-FSE 09), ACM, Aug. 2009, pp. 111-120
- [8] P. Bhattacharya and I. Neamtii, "Fine-grained incremental learning and multi-feature tossing graphs to improve bug triaging," Proc. IEEE Intl. Conf. Software Maintenance (ICSM 10), IEEE, Sept. 2010, pp. 1-10.
- [9] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, Aug. 2006, pp. 43–50.
- [10] 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