

# A Survey on Software Bug Evaluation

Shweta Soner  
Student  
Sanghvi Institute of  
Management, India

Swapnil Soner  
Asst. Professor  
Chameli Devi Group of  
Institutions, India

Maya Yadav  
Asst. Professor  
Sanghvi Institute of  
Management, India

## ABSTRACT

The costs of erroneous software can be as of the investment in software development. Yet, the potential to improve software quality and reduce project cost is enormous. Bug tracking and fixing can be an effective means to achieve quality i.e. error free at less cost. A System guides the maintenance activities of software developer's team who earlier, face the problem in fixing bugs. Non-availability of bug in a system in market makes fixing process difficult. Presently Bug fixing is time-consuming and exacerbates the already high cost of software maintenance which affects scheduled performance. This paper discusses a system that evaluates bugs and assigns priority on the basis of severity to fix them efficiently. .

## General Terms

Software testing, software tools and bug tracking.

## Keywords

Bug; defect tracking; software development; priority, severity.

## 1. INTRODUCTION

Bugs are commonly defined as "failure to conform to specifications," e.g., incorrectly implemented specifications and specified requirement(s) missing from the software. However, this definition is too narrow. planning within the software development society consistently recognize that most failures in software products are due to errors in the specifications or requirements—as high as 80 percent of total bug costs. Other studies have shown that the majority of system errors occur in the design phase [3]

## 2. MOTIVATION

Bug Tracking Systems can be separated systems that can integrate, and contribute in software development process. The Bug reports and detailed information associated with it can be resolved. Software Bug data is an important source for the organizations for the software process improvement decisions and that "ignoring bug data, can lead to severe cost for an organizational business". There is no doubt that software quality which is used in detecting bugs, is one of the key factors for evaluating the software process development. Also, Curhan mentioned that "some types of bug have a much higher costs to fix due to the customer impact and the time needed to fix it, or wider distribution of the software in which it is embedded " 1.

The Software bug Tracking Tools are simply built based on bug tracking models. Edwards and Steinke (2006) simply discussed the bug tracking model, according to the paper it is divided into the following two stages: ((repair /resolution/) - (verification)) and the following three changes of status: (discovery – resolved – closed) (Edwards, 2006).

The work flow of the model explains that it classified the new bugs into the following two categories: the first one comes from a user with a confirmation right, and the second comes from any user but it will not be confirmed till it has enough votes.

These votes are some time depends on perception and requirement but with the help of some method the quality of software in terms of bug fixing can be handled in right and systematic way.

## 2. SURVEY

Please Ron Patton, It's all about a sample bug report format that manage manually bug report in IEEE standard, include in this standard name of software, release, version, tester name, assigned to, severity, priority, title, description, resolved by, version, resolution comment, retested by, version tested, date tested, re-tested comment, organizer, developer, project manager, marketing person, support person. This a manual procedure used by IEEE standard.

Manual process creates lots of problem of maintaining all records of bugs. An automated tool is required by which the maintenance of all types of records in system is feasible. All the process should be automated for saving time, easy to work, flow of work in proper way.

Such IEEE format is not sufficient for long term because it is manual procedure but now-a-days requirement totally based on automated system. It is easy to process and simple format and cost is very low but this is for very limited purpose.[1]

Bala Subramaniam , Software quality costs are the costs associated with preventing, finding, and correcting bugive software. Costs of activities specifically designed to prevent poor-quality software, find bugs, such as code inspections and quality testing. Failure costs that result directly fix bugs and cost to deal with customer complaints, which are based on internal and external failure costs.

Total Cost of Quality = Prevention + Appraisal + Internal Failure + External Failure. Effective Bug evaluation strongly contributes to enhancing software quality and reducing development project costs. Using the broader definition of a bug ensures those not only are resultant errors or nonconformance to requirements discovered but also variance from a desired attribute, including incomplete requirements, takes place. Searches for such bugs can then take place across all software development phase.

Cost is very important factor for any software. According to this paper any type of failure condition occurs it create lots of problem. But only cost factor is not sufficient for quality assurance other factors are also i.e. effort. But for effective Bug evaluation there is not matter relevant to effort [2].

V.B. Singh1, Tracking of a reported bug for fixing is a fascinating area of research in software engineering. Many open source, free and commercial bug tracking tools have been developed and are currently under development. The industry needs criteria to select the best tool among the available set of tools that will help in fixing and tracking the progress of bug fixes. In this paper, BugZilla, Jira, Trac, Mantis, BugTracker.Net, Gnats and Fossil are used for comparative study. This paper presents a comprehensive

classification criteria to review the available tools and propose a new tool named Bug Tracking and Reliability Assessment System (BTRAS) for the bug tracking/reporting and reliability assessment. BTRAS helps in reporting the bug, assigning the bug to the developer for fixing, monitoring the progress of bug fixing by various graphical/charting facility and status updates, providing reliability bug prediction and bug complexity measurements, and distributing fixes to users/developers.[4]

Janák, One of the famous defects tracking tools used by quality control engineers is a Bugzilla defect tracking system. The work flow of the model showed that it classified the new bugs into the following two categories: the first one comes from a user with a confirmation right, and the second comes from any user but it will not be confirmed till it has enough votes. Also, it concentrated on quality control engineer roles in checking the appropriate solution which being satisfied, verified, closed or didn't conform with the solution.

Although the default IBM Rational Clear Quest Ticket mentioned the workflow path that the defect process has taken, and which "Starts when the defect is discovered and ends when the defect is resolved, hopefully repaired, for the most immediate release of the software application". It still has some shortcomings as the "rejected" status could be in any state. It may be after investigation, the approved state or after the task opened and in all the cases, it should be closed. Also the approved status should be one of the roles of quality control engineer; who should check it as the defect may not exist only in a new project process, but also may exist in the maintenance process.[5]

According to Prof. Torky Sultan, the defect tracking systems play an important role in the software development organizations as it can store historical information about defects. There are many researches in defect tracking models and systems to enhance its capabilities to be more specifically tracking. Furthermore, there are different studies in classifying bugs in a step by step method to have clear perception and applicable method in detecting such bugs. This paper shows a new proposed defect tracking model for the purpose of classifying the inserted defects reports in a step by step method for more enhancement of the software quality. Besides, an evaluation of experiment made for measuring the proposed factors results for defects classification. [6]

Akhilesh Babu Kolluri, Bug tracking is an essential discipline in the domain of software engineering. It has far reaching effects on the system when effectively used. The information provided in terms of bugs and solutions in the bug reports can help software engineers to act on them quickly and ensure that they are either rectified or eliminated from the system. The bulk of information provided in the bug reports may cause problem to developers in ascertaining poorly designed information. Therefore the bug tracking systems are to be improved and follow certain standards. To overcome the problem, the four fundamental directions are proposed to enhance effectiveness of bug tracking systems. To demonstrate the efficiency of the proposed directions, a prototype application is developed that can track the bugs effectively by capturing essential information from users and help to resolve bugs quickly. [7]

This Paper describes the terms and findings by forming a conceptual context and a foundation for the exploratory observational study that was central to this research.

The following are the Four Main Findings drawn from the study:

- 1-The Bug Examination Factor is considered as the greatest effect on the process of tracking defect reports. As with increasing the efforts of examining the tracking history of the database; it decreases the level of registered defect duplication.
- 2- The Bug type has significant effects on evaluating the tracking system as a method of structuring the inserted data to have structured correct information.
- 3- The Bug location has significant effects on detecting location of defects precisely through software development and enhancement.
- 4- The Quality Control field depending directly on human factor in triaging defects. This appears obvious through the Error Ratio in classifying defect type and bug location factor.

According to Thomas Zimmermann, It is important that information provided in bug reports is relevant and complete in order to resolve bugs quickly. However, often such information trickles to developers after several iterations of communication between developers and reporters. Poorly designed bug tracking systems are partly to blame for this exchange of information being stretched over time. This paper addresses the concerns of bug tracking systems by proposing four broad directions for enhancements. As a proof-of-concept, a prototype to track the bugs is demonstrated that gathers the relevant information received from the user and identifies files that need to be fixed to resolve the bug.

Current bug tracking systems do not effectively elicit all of the information needed by developers. Without this information developers cannot resolve bugs in a timely fashion and so we believe that improvements to the way bug tracking systems collect information are needed. [8]

This paper proposes four broad areas for improvements. While implementing a range of improvements from these areas may be ideal, bug tracking systems may instead prefer to specialize, thus providing a rich set of choices. This would be a healthy change to the current situation where they all provide identical functionality

### **3. CONCLUSION**

The All the studied approach, are used for making cost effective bug tracking with the help of different tool but how system will decide that which type of bug can be resolved at which priority and level of their severity. Most of time severity and priority of bugs are calculated on manual basis that creates lots of problem. In manual process, effort and cost calculation was not an important factor but recently some software tools are launched in the market which not only used for tracking bugs but also for cost factor, where failure is not emphasized. Track+ software tool is very effective but it includes no method for severity and priority calculation. For the fixing of bugs through severity and priority, effort calculation is required. Function point gives an effective procedure for the requirements but it does not give the values in person – hours required [9]. So, use case points calculation is used for test effort calculation.

With such approaches, discovering and fixing the software bugs is difficult as maintenance task, and software developers on this issue devote a considerable amount of effort. In the world of software one cannot get rid of the bugs, fixes, patches etc. each of them have a severity and priority

associated to it. There is not yet any formal relation between these components as both of these either depends on the developer and tester or on customer and project manager to be decided on. On one hand, the priority of a component depends on the cost and the efforts associated with it. While on the other, the severity depends on the efforts required to accomplish a particular task. This Paper work required a formula that can draw a relationship among severity and priority.

#### **4. ACKNOWLEDGMENTS**

We would like to thank the all faculty members of the institute who helped us lot in calculating the facts and figures related to my paper. I would also like to thank the anonymous reviewers who provided helpful feedback on my manuscript.

#### **5. REFERENCES**

- [1] Ron Patton, software Testing, An Authoritative guide to Software testing, Pearson Education.
- [2] Jørgensen M. and Shepperd M.: A Systematic Review of Software Development Cost Estimation Studies. IEEE Transactions on Software Engineering. (2006)
- [3] Vince Kellen, Kallista, Inc. - Estimating and Tracking Software Projects Londeix B. - Three Points Techniques in Software Project Estimation - SEER - April
- [4] Bug Tracking and Reliability Assessment System (BTRAS), V.B. Singh1, ] International Journal of Software Engineering and Its Applications Vol. 5 No. 4, October, 2011.
- [5] (Janák, 2009) Jiří Janák, Issue Tracking Systems, Brno, spring, 2009.
- [6] Prof.Torky Sultan DEVELOPMENT AND EVALUATION OF A DEFECT TRACKING MODEL FOR CLASSIFYING THE INSERTED DEFECT DATA, European Scientific Journal April 2013 edition vol.9, No.12 ISSN: 1857 – 7881 (Print) e - ISSN 1857-7431 170.
- [7] Akhilesh Babu Kolluri,,Effective Bug Tracking Systems: Theories and Implementation IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 4, Issue 6 (Sep-Oct. 2012), PP 31-36 .
- [8] Thomas Zimmermann, Improving Bug Tracking Systems , Microsoft Research, Redmond, USA
- [9] (Lethbridge, 2003) Timothy C. Lethbridge , Janice Singer and Andrew Forward , How Software Engineers Use Documentation: The State of the Practice, IEEE Software Volume 20 (Issue 6), 2003.
- [10] (Curhan, 2005) Lisa A. Curhan, Software Defect Tracking during New Product Development of Computer System, Massachusetts institute of technology, 2005.
- [11] Jim Nindel-Edwards and Gerhard Steinke, A Full Life Cycle Defect Process Model That Supports Defect Tracking, Software Product Cycles, And Test Iterations, Communications of the IIMA ,Volume 6 Issue 1, 2006.