

Design of Bug Tracking System

Gauri M. Puranik ^{1,2}

P.G. Student, Department of Computer Engineering, Sandip Foundation's, Sandip Institute of Technology and
Research Center, Nashik, Maharashtra, India¹

Professor, Department of Computer Engineering, Sandip Foundation's, Sandip Polytechnic, Nashik, Maharashtra,
India²

ABSTRACT: Bug tracking is a system which is used to solve out any type of bugs in any software. It is mostly useful for any software company. In this system we have design different types of user permission like developer, tester having different rights to connect software. The administrator user can create the user account in the system and gives the rights (permission) as well as he can maintain bug tracking in the system for every projects. As software development is completed, the projects are goes in the next phase i.e. testing. The test engineers those are working in the same project are test the product (Software developed by developer) and if any bug are found they can log (Enter) that bug with his description in our software also set the priority for that bug with his description in our software also set the priority for that bug like high, medium, low, not reproducible. After the first cycle of bug tracking is completed the developer those are working in those projects can log in to system and get the bug list with priority. He can solve the bug and change the status of that bug in our software. The administrator or Project manager can get the idea of bug status, work status of both developer and tester and time span for that project. There is also one major part of user side to test the software called as UAT (User Accepting Testing). If support engineer or client faces any issue in the system after delivery they can also log in to system and put their issue regarding that project.

KEYWORDS:- Error, Bugs, Software Development Life Cycle (SDLC), UAT (User Accepting Testing).

I. INTRODUCTION

Bug Tracking System (BTS) is a hierarchical issue tracking and BTS created specifically for software development component. Developers, testers and manages use it to record and track the progress of defects (bugs), problem and new feature. Evaluating a bug tracking system requires that you understand how specific features, such as configurable workflow and customizable fields, relate to your requirements and your current bug tracking process. This article provides tips and guidelines for evaluating features, and explains how these features fit into a defect tracking process. In this system we have design different type of user permission like developer, tester having different rights to connect software. The administrator user can create the user account in the system and gives the rights (Permission) as well as he can maintain bug tracking in system for every projects. The administrator can add project name in the software assign rights to developer and tester those are working on that same project. As software development is completed, the projects are goes in next phase i.e. testing. The test engineers those are working in same project are test the product (Software developed by developer) and If Any Bug are found they can log (Enter) that bug with his description in our software also set the priority for that bug like high, medium, low, not reproducible.

The administrator or Project manager can get the idea of bug status, work status of both developer and tester and time span for that project. If Support engineer or Client faces any issue in the system after delivery they can also log in to system and put their issue regarding that project. Most BTS is used to manage product development by focusing effort on the tasks required to efficiently bring the project to completion. If you want to track more than just bugs, make sure the bug tracking system can be adapted to track other types of issues (such as support calls, test cases, or purchase orders). A system that is designed specifically for bug tracking can be hard to dep't, so look for a system that provides pre-built templates for tracking different issue types. Also, as you evaluate the various features of the tracking system, look for any hardwired "bug-tracking" terminology or functionality. Fields, queries, reports, notifications, and workflow: everything should be adaptable. A bug tracking system should allow you to quickly gather the information

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2014

you need for staff meetings (bug listings and printouts), as well as provide more detailed metrics to help you make decisions.

Bug tracking tools support a fixed, standard set of notifications:

- Bug added
- Bug edited
- Bug status changed
- Bug assigned

Record bugs in a project. Support concurrent access to bug reports by multiple users. Support multiple projects. Assign responsibilities, access and privileges to users for each project. Resent unique views for developers, testers and managers. Estimate, assign and track the work required to resolve and test bugs. Sort, print and generate reports on bugs by priority, type and other properties. Track the time required for Task completion. Determine release dates based on bug detection and resolution statistics. Automatically maintain the history of each bug. Classifies issues – understanding, client error, bug, etc. Checks which issues are still open / open for long time. Developer works on and closes the issue, enters time worked. Auto-email to client that the issue is closed. Reports on how much work done for support of each client – in each category. Reports on average time to close issue, longest open issues, number of open issues etc.

II. RELATED WORK

Bugs in the process of software development are unavoidable. However, they are to be tracked and fixed with some focus otherwise they are manifested into the final product and that leads to failure of software systems. An error, fault, failure, mistake, flaw can be called as “bug” with respect to a software system. SDLC has many phases. The development team may commit mistakes in any phase. However, when comes to coding it gets surfaced once execution is tested. Most of the bugs can be avoided by having correct analysis and design as per customer requirements. According to bugs in software system cause the system to fail. It does mean that such software system can't meet the quality and expectations of the client. When software system is able to meet the functional requirements given by the customer, which is said to be a quality product. When quality problem arises, customer gets dissatisfied. According to a bug in software system is not an accident bug that occurs due to specific reason. Tracking bugs has many phases and it has its own life cycle as shown in following fig.

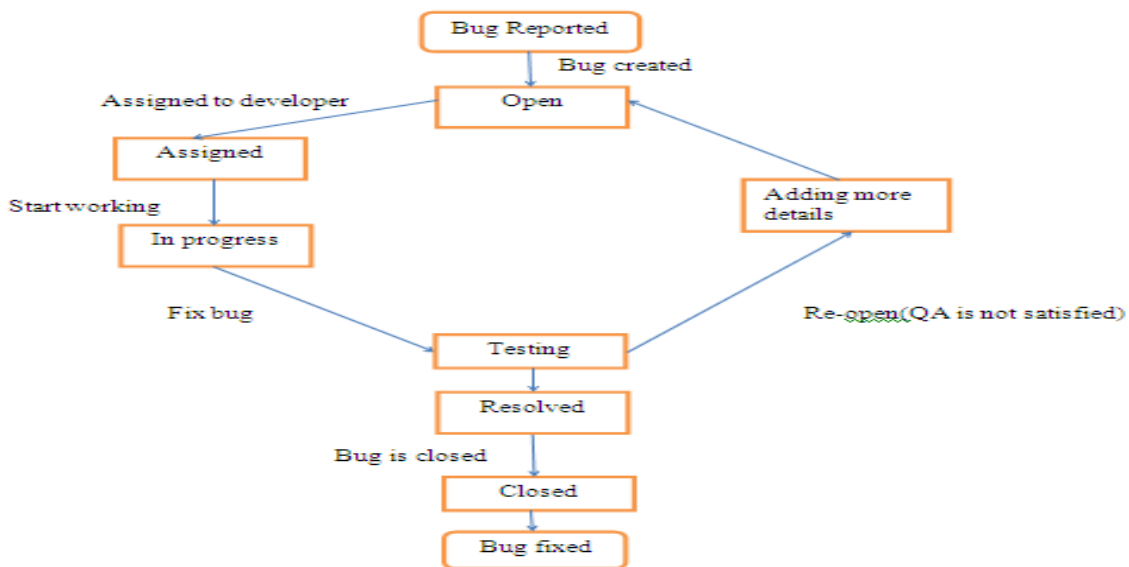


Fig .Work flow diagram of Defect Manager

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2014

Configure different user roles to see different lists of bugs. For example, a developer might see a list of open bugs, A QA analyst might want to see a list of bugs ready for testing. Search for bugs using flexible criteria. As bugs cause systems to act incorrectly bug tracking systems reduce the likelihood of having bugs in the system long time. Quickly the software development team can fix bugs and close them with suitable bug tracking system. Bug tracking improves quality of software being developed. Even after delivering software products bugs may be unearthed and still the tracking system is to be used to fix those bugs faster. Software development companies certainly use a bug tracking system as it is essential. Bug tracking system is really required while developing every software product as many developers do not maintain adequate documentation with respect to customer requirements through the entire life cycle of the product. There are many benefits of using a bug tracking system. Any bug tracking system can be of two types such as simple and complicated. However, ideally it must be simple and efficient. It should work fast and help people to track and fix bugs so as to develop quality products and satisfy clients. "Bug Genie" is another real world bug tracking system with very good features.

III. IMPLEMENTATION OF BUG TRACKING SYSTEM

When any software engineer presents a bug report, most probably, he is asked many questions. Some of them are what is the name of the product? What is the bug? In which component is the bug?; in which module is the bug? In which method the bug is? in which environment the bug arises?; in which platform the application is built?; in which OS the application runs? The information given by developer who report bug might be incomplete initially. When a bug report is submitted by a developer, the follow up questions are to be asked immediately besides keeping the submitted bug report in hand. We recommend software development teams to have bug tracking systems that contain "build expert systems". These systems ask all required questions to software engineer so as to make the work automated. The question to be given and answered by developer is not static. The questions do not come sequentially. Moreover answer to a question determines the next possible question. Narrowing down the location of bug and to have accurate bug descriptions are features of expert.

The following data is essential in order to build an expert system.

- Bug location information which is crucial to tracking bugs. Location gives you the line number, method, class and so on. This helps developers to move to that place with ease. Many software development environments (IDEs) allow the bugs to be located just by a click of button or a click.
- From the bugs list, machine learning models can be built that choose questions and also predict the location of the bug based on the responses that corresponds to the bugs. This paper provides a proof of study that makes use of data that is present in the bug reports. Thus we get collection of information that is essential in implementing a tool that can support automatic evaluation of the information.

IV. EXPERIMENTAL RESULTS

- Record bugs in software.
- Sort, print & generate reports on bugs by priority type.
- Determine release dates based on bug detection & resolution statistics.
- Automatically maintain the history of each bug.
- Support multiple softwares.

V. CONCLUSION

In software test management, bug reporting is a complex and complicated process that requires precision, detailing and a whole lot of information. Reporting and tracking bugs manually works fine in case of small projects, whereas for mission-critical or large projects, a paper-based approach can result in chaos and confusion. BTS assures that all project members are in a communication loop. Interface is easy to navigate. View-level security (defines multiple views of a project and then grant access permissions). A bug-tracking system helps the project team to successfully measure the project's status. By This system administrator or Project manager can get project status as well as management can get the ongoing status of the project. A defect tracking system not only tracks defects but also tracks metrics to make sure everything is going according to the software development plan.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2014

REFERENCES

- [1] A Practitioners approach "Software Engineering" Roger S. Pressman 5th Edition.
- [2] Lessons Learned in Software Testing (Paperback) Publisher: Wiley; 1st edition (December 15, 2001) James Bach Bret Pettichord, C. Kaner.
- [3] Beautiful Testing: Leading Professionals Reveal How They Improve Software (Theory in Practice) (Paperback) Publisher: O'Reilly Media; 1 edition (October 22, 2009) Tim Riley Adam Goucher.
- [4] Why Programs Fail, Second Edition: A Guide to Systematic Debugging (Paperback) Publisher: Morgan Kaufmann; 2 edition (June 26, 2009).
- [5] Software Engineering, "Roger S. Pressman" (TATA McGraw Hill Publication).
- [6] http://en.wikipedia.org/wiki/Bug_tracking_system
- [7] <http://www.wisegeek.com/what-is-bug-tracking.htm>
- [8] <http://www.bug-track.com>
- [9] <http://www.joelonsoftware.com/articles/fog0000000029.html> Painless Bug Tracking by Joel Spolsky.
- [10] <https://bugzilla.mozilla.org>
- [11] Automated Web Testing Toolkit: Expert Methods for Testing and Managing Web Applications (Paperback) Publisher: John Wiley & Sons (July 3, 2001) "Diane Stottleymer".
- [12] Darin Dillon (Author) Publisher: Apress; 1 edition (April 1, 2003).
- [13] Software Quality - ECSQ 2002 (Paperback) Product Details Publisher: Springer; 1 edition (July 10, 2002).

BIOGRAPHY



1. Prof. Gauri Mandar Puranik P.G. Student, Department of Computer Engineering, Sandip Foundation's, Sandip Institute of Technology and Research Center, Nashik, Maharashtra, India¹.
Professor, Department of Computer Engineering, Sandip Foundation's, Sandip Polytechnic, Nashik, Maharashtra India².
Her research interests are in the software development life cycle and overcome techniques of issues occurred in that.