Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Proposed Framework for Bug Tracking System in OSS Domain

Swati Sen
Student M Tech, Computer Science,
Dept. of Computer Science,
Himachal Pradesh University, Shimla, India

Anita Ganpati
Assistant Professor,
Dept. of Computer Science,
Himachal Pradesh University, Shimla, India

Abstract—According to the present scenario, open source software revolution is gaining momentum in information technology era. In the battlefield of software technology, the open source software cannot be ignored. The philosophy behind open source extends towards quality and efficiency. In Open Source Software development there is a shared understanding among open source developers, reporters and users that efficiently improves the product quality. Yet quality and efficiency of Open Source Software depends upon the bugs present in the product. Thus understanding and tracking of system is vital process. In Open Source Software development tracking of bug is most important step. Bug Tracking system plays an important role in tracking of bugs. Bug tracking system contains the large amount of information about the bug in open source software. Choosing a good bug tracking system for any product will increase productivity of software. So we analyzed the existing bug tracking system and find out the limitations. Later we proposed the framework for bug tracking system based on the limitations founded in existing bug tracking system.

Keywords –- Open source software, Bug tracking system, Bug Report, Bugs, Developers, Reporter

I. Introduction

Open source software (OSS) is computer software with its source code that is made freely available. This software is distributed under licensing agreement. This licensing agreement allows the source code to be shared, viewed and modified by users or organization. Open source software is an alternative way to develop software. Open source software is developed by community of volunteer developers over the internet. The source code of open source software is available to everyone and it can be freely used, modified and distributed at no cost. Steve Weber described open source as an experiment in social organization around a distinctive notion of property. Property in open source is configured fundamentally around the right to distribute, not the right to exclude [1]. Open Source has been getting much attention in the last few years .Now a days Many corporations, large and small, have taken an interest in this growing software market. This software may require additional module or enhancement of existing module time to time. But no software is perfect. Some of the software or module may contain bugs. These bugs can be unnoticed that are left from time to time. In Open Source Software development there is a shared understanding among open source developers, reporters and users that efficiently improves the product quality. Yet quality and efficiency of Open Source Software depends upon the bugs present in the product. Thus tracking of bug is important .Bug tracking system plays in important role in tracking of bug.

A bug tracking system is an application that lets one to keep track of bugs for software project in database. The report of the bug stored in bug tracking system .Where assignee of the bug Fix it. In open source software environment, user of open source software often write a “bug report” when they find bug or come across a mistake. Bug tracking system allows people anywhere in the world to report and describe the bug whenever they like. Anyone can access the repository of bug. An efficient bug tracking system that can be mapped well to development and quality process is an invaluable tool. Conversely a poor bug tracking system is difficult to use and does not fully reveal the state of software [2] [3].In our study we compare different bug tracking system and proposed a framework for bug tracking system on the basis of our findings in existing bug tracking system.

II. Literature Survey

Nicolás Serrano and Ismael Ciordia [4] has compared two bug tracking tool i.e. are Bugzilla and ITracker. The objective of their research was to provide a comparative study of these two bug tracking tool based on the criteria platform independence, database independence, how customizable is it, are the number of users limited and life of cost. G Abaee and D.S Guru [5] gave the best practice to test documentation and effort estimations have been investigated as well as Bug Tracking Tools. They compare the four different existing Bug Tracking Tools with each other along with their features and drawbacks. Then they proposed new one, the Debugger Thomas Zimmermann, et al. [6] addressed the concerns of bug tracking systems by proposing four broad directions for enhancements .They discussed that it is important that information provided in bug reports is relevant and complete in order to help resolve bugs quickly.. Poorly designed bug tracking systems are partly to blame for exchange of information being stretched over time enhancements. As a proof-of-concept, they also demonstrate a prototype interactive bug tracking system that gathers relevant information from the user and identifies files that need to be fixed to resolve the bug. Nicholas Jalbert and Westley Weimer [7] discussed that Bug tracking systems are important tools that guide the maintenance activities of software developers. The utility of this tool is affected by an excessive number of duplicate bug reports–in some projects as many as a quarter of all reports are duplicates. If Developers manually identify duplicate bug reports, this identification process is time-consuming and exacerbates the already high cost of software maintenance. So they propose a system that automatically classifies duplicate bug reports as they arrive to save developer time.

Fischer et al. [8] discussed that Version control and bug tracking systems contain large amounts of historical information that can give deep insight into the evolution of a software project. Unfortunately, these systems provide only insufficient support for a detailed analysis of software evolution aspects. They addressed this problem and introduced an approach for populating a release history database that combines version data with bug tracking data and adds missing data not covered by version control systems such as merge points.

S. Just et al. [9] concluded that Developers typically rely on the information submitted by end-users to resolve bugs. They conducted a survey on information needs and commonly faced problems with bug reporting among several hundred developers and users of the ECLIPSE, APACHE and MOZILLA projects.

M.P. Francisco et al. [10] have developed a tool to extract and to store information from Debian's BTS (Bug Tracking System) in a relational database. In this paper they showed that there is a strong dependence between three variables which can be used to analyze the activity of a project through its bugs: communications between users and developers, bug notifications and people involved. They explained that bugs are an essential part of software projects because they lead its evolution. Without bug notifications developers cannot know if their software is accomplishing its tasks properly.

A. Hora et al. [11] discussed that to harness the complexity of big legacy software; software engineering tools need more and more information on these systems. This information may come from analysis of study of execution traces, the source code, computing of metrics, etc. One source of information received less attention than source code: the bugs on the system. Little is known about the evolutionary behavior, lifetime, distribution, and stability of bugs. In this paper, they proposed to consider bugs as first class entities and a useful source of information that can answer such topics.

Stephen Blair in his paper [12] provided tips and guidelines for evaluating features, and explains how these features fit into a defect tracking process. He discussed that 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. He explained before you start evaluating bug tracking systems; make sure you identify your requirements for the system.

III. Need and Scope of Study

With the increase in the use of open source software the information technology era has given birth to new revolution. Wide range of Open Source Software is available in any usage area. Wide range of software products namely Operating System, Webservers, Word Processor, Databases, Bug Tracking System, Antivirus, Data Mining Software etc. are available

The various Bug Tracking Tool are available in Open source domain. Bug Tracking system gives the complete information about the bugs which help the developer to keep the track of bugs in the software product. The software product now a day are becoming more complex and it is becoming more difficult to keep the track of huge amount of bugs in software having the complete and accurate information about the bugs helps the developers to resolve it. So choosing a good tracking system for any of product will increase the productivity of software, improve the communication between the developers, produce the reliable and secure software and it will raise the customer satisfaction.

Keeping the importance of bug tracking system in mind a comparison of six different open source bug tracking system chosen for study are : Bug Genie, Mantis, Bug Tracker, Bugzilla ,ITracker, WebIssues**.**

IV. Objectives

The broad objective of the study is to present a comparison of six, different Bug Tracking System. The specific objectives of the study are:

(i) To perform the comparative study of Bug Tracking system.
(ii) To identify the limitations of Bug Tracking System under study.
(iii) To propose a framework for Bug Tracking tool.

V. Research methodology

In order to meet the objective theoretical approach has been used .The theoretical approach concentrate on describing Open source software, software bugs, bug life cycle, and bug tracking system. The theoretical approach is based on review of secondary data acquired from literature survey, articles, books, research paper and internet.

VII. Research work

There are many Bug tracking System in software market. Choosing a good bug tracking system helps in increasing productivity, customer satisfaction and also improves communication between developers. We selected six bug tracking tool and their analysis is done on the basis of following criteria platform, user interaction, size and usage, functionality ,searching, reporting. All these categories are further divided into sub-categories. Tool considered for analysis are Bugzilla, Bug Genie, Mantis, Bug tracker, iTracker, WebIssues

A. Analysis on the basis of User Interaction

The comparison based on User interaction basis shown in a table I. The criteria for User interaction is based on User interface, Language in which available, E mail notification etc.

TABLE I
ANALYSIS ON THE BASIS OF USER INTERACTION

User Interaction / **Tools**	**User Interface**	**Available in Language**	**Email Notification**	**Intended Audience**	**Search Facility**
Bug Genie	web interface	English, French, German, Swedish, Norwegian, Spanish,	Yes	Customer Service, Developers, Information Technology, Quality Engineers, System Administrators Manufacturing,	Yes
Mantis	Web interface	English	Yes	Developers, System Administrators	Yes
Bug tracker	Web interface	Chinese,Czech,English, Polish,Portuguese, Slovak, Spanish, Turkish,French, German, Greek, Italian	Yes	Developers	Yes
Bugzilla	web interface	Multiple languages	Yes	Customer Service, Developers, Other Audience, Quality Engineers, , End Users/Desktop	Yes
iTracker	Web interface	Catalan,Chinese,English,French,German,Italian,Portuguese , Turkish	Yes	Customer Service, , End Users/Desktop,InformationTechnology, Developers, System Administrators, quality Engineers	Yes
WebIssues	web based,WebIssue client that run on Linux, window,	Brazilian Portuguese, Chinese (Simplified), Dutch, English, French, German, Polish, Spanish	Yes	Customer Service, Developers, End Users/Desktop, Information Technology, Quality Engineers	Yes

User Interaction is most important criteria to classify a bug tracking system. All bug tracking system provides web based user interface. WebIssues provide access by both the web browser and a desktop client application. Multiple languages are supported by most of the bug tracker. But mantis is only available in English language. All the five bug tracker provides email notification and search facility for user interaction. Intended audience for bug genie are Customer Service, Developers, Information Technology, Manufacturing, Quality Engineers, System Administrators whereas for Mantis Developers and System Administrators. Developers are only audience for Bug tracker and for Bugzilla Customer Service, Developers, End Users/Desktop, Other Audience, Quality Engineers are entertained.

B. Analysis on the basis of Size and Usage

The comparison based on Size and Usage basis shown in a table II. No of versions, first registered and last updated

TABLE II
ANALYSIS ON THE BASIS OF SIZE AND USAGE

Usage and size **Tool**	**First registered**	**First Release**	**Last updated**	**Total no of Version Released**
Bug Genie	2003-09-05	2003-09-05	2013-03-04	67
Mantis	2000-11-18	2000-12-01	2013-04-13	107
Bug tracker	2005-03-30	2005-05-30	2011-06-19	32
Bugzilla	1988	1998-09-19	2013-02-19	116
iTracker	2002-05-22	2002-06-12	2012-12-13	22
WebIssues	2007-02-12	2007-02-12	2013-03-12	15

On the basis of this table we analyze that Bugzilla and mantis have number of releases this means that Bugzilla, Mantis update their version frequently. Bug Genie has 67 releases while iTracker has 22.Whereas looking at the table we can see that it iTracker is registered first than Bug Genie

C. Analysis on the basis of Platform

The comparison based on platform basis shown in a table III. Platform refers to what is the programming language of the tool, operating system, web server, database and license.

TABLE III
ANALYSIS ON THE BASIS OF PLATFORM

Platform **Tool**	**Program-ming language**	**Operating system**	**Database**	**License**	**Web Server (if Any)**	**Client**
Bug Genie	Java Script, PHP	Cross-platform	MySQL >= 5.0 or PostgreSQL >= 8.2	Mozilla Public License 1.1 (MPL 1.1)	most web server which supports rewrite rules eg. window server, Linux server	Any web Browser
Mantis	PHP	Cross-platform	MySQL, MS SQL, and PostgreSQL	GNU General Public License v2	Apache and MS-IIS	Any web Browser
Bug-tracker	PHP	Window 2000	PostgreSQL and MySQL	GNU General Public License v2	IIS 5.0 -	Any Web Browser

Bugzilla	Perl	Linux, window,	SQLite, MySQL, PostgreSQL, Oracle	Mozilla Public License 1.1 (MPL 1.1)	Apache and MS-IIS	Any Web Browser
iTracker	Java	OS independent	Database independent	GNU General Public License v2	Apache	Any web Browser
WebIssues	C++,PHP	Window ,Linux, OS x	MYSQL,PostgresSQL,SQL server	GNU General Public License v3	WebIssues server	Any Web Browser,desktop client

In our comparative study of six named tool most of the tool use PHP in their coding as programming language ,Whereas Bugzilla use Perl and iTracker use java in coding. Most of the bug tracking tool are OS independent where as Bugzilla are available on Linux and windows. MY SQL, MS SQL, PostgreSQL are Databases supported by most of the bug tracker where ITracker is independent of database. Bug Genie and Bugzilla are licensed under Mozilla public license 1.1 where other bug tracking system mantis, bug Tracker and iTracker licensed under GNU General Public license. Mantis and Bugzilla use Apache and IIS-MS as a web server where only Apache is used by iTracker, IIS 5.0 is used by Bug Tracker and bug Genie uses most web server which supports rewrite rules examples is Window server, Linux server. All the client are web based any browser can be used so they are platform independent.

D. Analysis on the basis of Functionality

The comparison based on functionality it provide basis shown in a table IV. Search and filter, Time Tracking, Usage Statistics, and Automatic Duplicate Bug detection, RSS feed, Localization, Automatic assignment and reassignment of bug to expert.

TABLE IV
ANALYSIS ON THE BASIS OF FUNCTIONALITY

Functionality / **Tool**	**Search and Filter**	**Time Tracking**	**Usage statistics**	**Automatic Duplicate Bug Detection**	**RSS Feeds**	**Localization**	**Automatic Assignment of bug to expert**
Bug Genie	Yes	Yes	Yes	No	Yes	Yes	No
Mantis	Yes	Yes	Yes	No	Yes	Yes	No
Bug tracker	Yes	No	Yes	No	No	No	No
Bugzilla	Yes	Yes	Yes	Yes	Yes	Yes	No
iTracker	Yes	No	No	No	No	Yes	No
WebIssues	Yes	No	No	No	No	Yes	No

All bug tracking system has Search and filter functionality. Time tracking is provided by only Bugzilla, Mantis, Bug Genie where as Bug Tracker, iTracker and WebIssues are fail to provide time tracking time tracking is important functionality by which one can estimate how many hours a bug will take to fix, and then keep track of the hours one spend working on it .Most of the bug tracker are not providing the functionality to detect duplicate bug report only Bugzilla have this functionality. Bug localization is provided by most of the bug tracking system whereas system except iTracker or WebIssues. Localization is common feature in all bug tracking system except Bug Tracker. No bug tracker provides the functionality of automatic assignment and reassignment of bug to expert.

E. Analysis on the basis of Reporting a bug

Reporting a bug in bug tracking system is very important feature of any bug tracking system. A reporting feature should be well designed. We did the analysis of bug tracking tool on the basis of reporting and took the feature online reporting, reporting by e-mail reporting feedback

TABLE V
ANALYSIS ON THE BASIS OF REPORTING OF A BUG

Reporting **Tool**	**Online**	**e-mail**	**Feedback**
Bug Genie	Yes	Yes	Yes
Mantis	Yes	Yes	Yes
Bug Tracker	Yes	No	Yes
Bugzilla	Yes	Yes	Yes
iTracker	Yes	Yes	Yes
WebIssues	Yes	Yes	Yes

In our analysis we find that all the bug tracking system which are under study have online system to report bug and user can also report bug by Email. Bug Tracker Bug tracking system does not provide the facility to user to report bug by e-mail. Feedback is provided by the all bug tracking systems.

After analyzing all above mentioned bug tracking system we find that there are many challenges in front of user to find out which one is best. Reporters of bug also face many difficulties in reporting a bug. If bug report in not reported accurately an ambiguity may arise. This can lead to wastage of time and effort. Bug tracking system should be perfect. If bug report is accurate and bug reporting system is good than it will easier to track a bug.

VIII. Limitations

There are some limitations to current bug tracking system. During our analysis of various bug tracking system we find some issue which need to be included in current bug system. A good bug tracking system should have a good reporting design system, comfortable work environment for developer and experts and complete information for anonymous access.

1.) For good design of the reporting system a bug tracker should have user Friendly interface. There should be facility to reporter to submit report according to his usefulness. There should not be repeated question in every bug report. Asking question according to problem defined will give a better bug report. And it should be web browser based that user can report a bug from anywhere anytime. And it should be available in multi-language so that user from all around the world can report. It should have automatic duplicate bug detection system so reporter of bug find out that bug has been reported earlier. And what is the status of bug.

2) In existing bug tracking system there is no filtration of bug report in initial phase. Existing bug tracking system allows fake bugs to be submitted as bug report which is assigned to expert and entered in a bug list. The bug should not be submitted without complete description or valid description. And should not be entered in list of bug. This will reduce the effort of assignee to check the fake bug report and no fake bug is entered into the database of bug tracking system .To give the comfortable environment to developer and expert a bug tracking system should have automatic bug assignment system so that when a new bug reported into the system it is automatically assign to an expert of that area to which bug belong. For this bug tracking system should have advance capabilities. Such that there should be automated expert system in the system which will study a bug report and find a close match between bug report and expert of that field. And bug is automatically assigned to expert. But there should not over burden on one expert. Complete information should be given to expert to resolve the bug. A good bug report make easier to track the bug in buggy code. And it will reduce the effort and time spend of expert or developer

3) Some of the users are anonymous user. They never Report a bug and neither contribute for bug fixing. They only give anonymous access to bug tracking system’s bug repository. So a good bug tracking system should provide complete information to anonymous user who wants to collect the information from bug tracker’s repository for their use. All bugs should have complete information and history of bug .All the bug should be categories according to their priority and status. To our best knowledge no bug system provide total no of bug submitters in particular version of software. User has to calculate bug manually. There should be a feature of calculating total no of bug in particular version and there should also be record of the bugs that are present in more than one version.

We proposed a bug tracking system that eliminates some limitations of existing bug tracking system.We propose that there should be facility to reporter to add fields according to his usefulness during the reporting of bug. There should be an automatic process that ask relevant question to reporter according to problem defined. There should be filtration of fake bugs in initial steps .The filtration will describe bug as Bug or Not A Bug. We proposed that there should be automatic identification of bug report which will identify between actual bug and NotABug to reduce the manual work. We also propose an automatic bug assignment /reassignment process to expert. For this there should be an expert system which will find a close match between bug report and expert of that field and automatically assigned to expert. After resolving the bug its status becomes resolved which than pass to quality assurance team. Quality assurance team will verify the bug after the verification of bug if quality assurance team will not satisfy the bug can be reopened and if satisfy the bug will closed. Reopened bug assigned to expert again. Closed bug can be Fix, Wontfix and Incomplete. Fixed bug should be commit in version control .Fig 1. shows the proposed bug tracking system in oss domain.

IX. Proposed bug tracking system

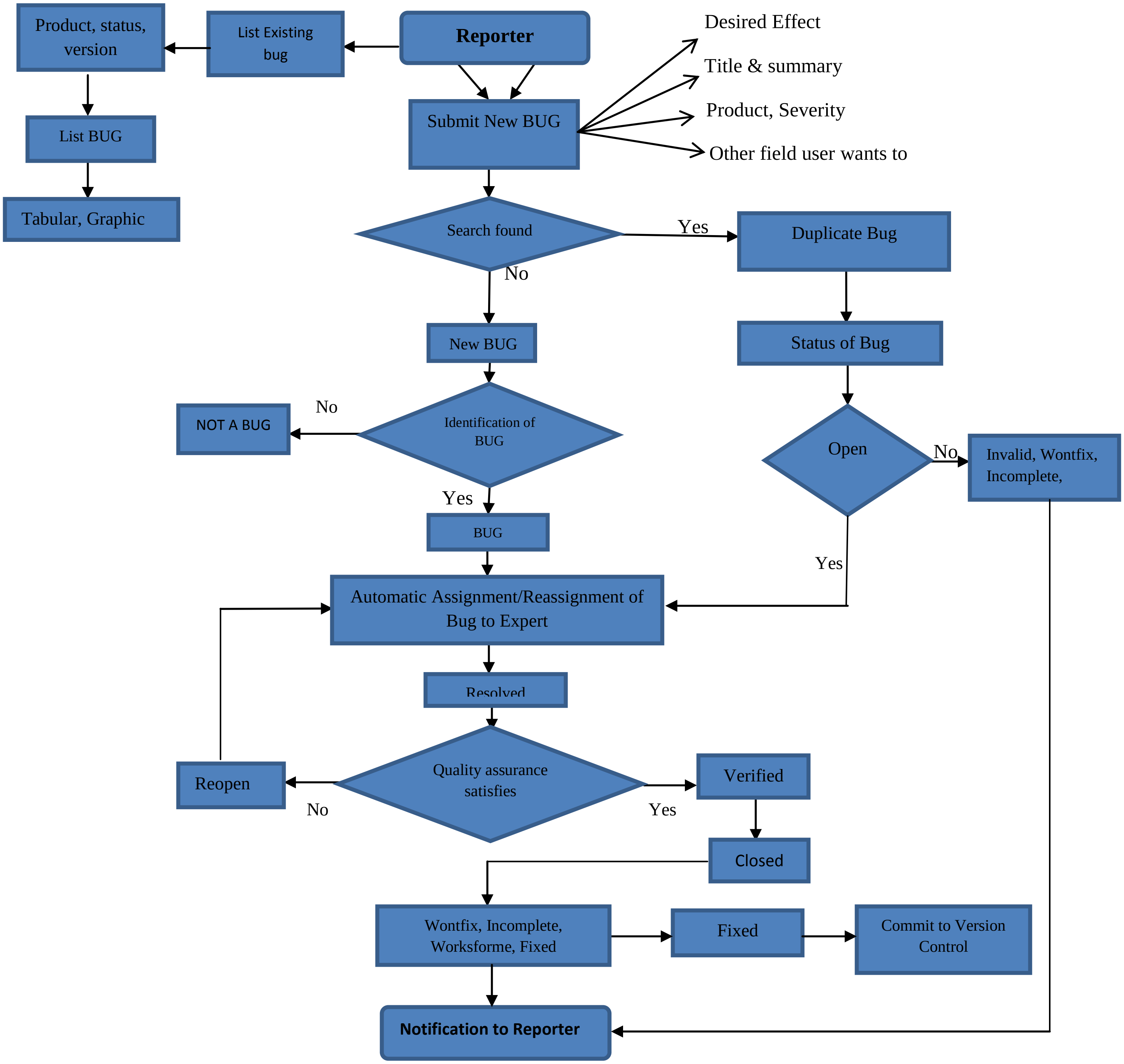

Fig 1: Proposed Framework for Bug Tracking System

X. Conclusions

Bug Tracking System is important software typically have tens or hundreds or thousands of defects. Bug tracking system is use to manage, fix and prioritize these defects. Defect tracking system is computer database system that store defect and help people to manage them. The objective of our study is to present comparison of different bug tracking system and to identify limitation of current bug tracking system. We concluded that current bug tracking system have some of limitation. They do not effectively collect all the information needed by developer, reporter and anonymous user. We have done analysis of bug tracking system on the basis of some criteria. But such criterion often doesn't give desired result.

So we propose a new framework of bug tracking system .the propose framework will give a improved level of satisfaction for current bug tracking system

References

[1] Steve Weber, "*The success of open source",* Harvard University Press, 2009.

[2] K Sowe Sulayman, G. Stamelos Ioannis, "*Emerging free and open source software practices*", Idea Group Inc (IGI), 2008.

[3] Martin Reddy, "*API Design for C++*", Elsevier, 2011.

[4] Nicolas Serrano, Ismael Ciordia, "Bugzilla, ITracker, and Other Bug Trackers", *IEEE software*, Vol 22, pp. 11-13, 2005.

[5] G Abaee, D.S. Guru, "Enhancement of Bug Tracking Tools; the Debugger", *Software Technology and Engineering (ICSTE), 2nd International Conference*, Vol 1, 2010.

[6] Thomas Zimmermann, Rahul Premraj, Jonathan Sillito, and Silvia Breu, "Improving Bug Tracking Systems*", 31th International Conference on Software Engineering , Vancouver, BC, Canada,* 2009.

[7] Nicholas Jalbert, Westley Weimer "Automated Duplicate Detection for Bug Tracking Systems" *International Conference on Dependable Systems & Networks*: *Anchorage, Alaska, IEEE,* 2008.

[8] M. Pinzer Fischer, H. Gall "Populating a Release History Database from version control and bug tracking systems" *Software Maintenance, IEEE,* 2003.

[9] S, Just, R. Premraj and T. Zimmermann. "Towards the next generation of bug tracking systems", *Visual Languages and Human-Centric Computing, IEEE,* 2008.

[10] M.P Francisco, P.B. Perez and G. Robles "Correlation between bug notifications, messages and participants in Debian's bug tracking system" *Empirical Software Engineering and Measurement, First International Symposium*, 2007.

[11] A. Hora, N. Anquetil, S. Ducasse, M. Bhatti, C. Couto, M.T. Valente and J. Martins, "Bug Maps: A Tool for the Visual Exploration and Analysis of Bugs" *Software Maintenance and Reengineering (CSMR), 16th European Conference*, 2012.

[12] Stephen Blair "*A Guide to Evaluating a Bug Tracking System*, White paper, 2004.