Analysis of Bug Tracking Tools

Sandeep Singh

Abstract- Nowadays, when project are so extensive, defects or bugs have been existed as a problem in the system and they are normally inevitable in software development. A bug could be in either a program's source code or its design. Huge amount of bugs could be found in system development. It is relatively difficult to manage bugs in simple word documents or remember everything in one's head. Because of that, it becomes very important to have appropriate bug tracking tool. Bug tracking tool makes communication between teams more effective and all the bugs and changes are recorded in web based system and hence the system is scalable. The purpose of this paper is to study different hug tracking tools, and purpose new set of selection criteria that gives more satisfying solution, as 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.

Index terms- Bug, Bug tracking, centralized data repository, notification method, web-based application.

1 INTRODUCTION

A software bug is an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unplanned ways. Most bugs arise from mistakes and errors made by people in either a program's source code or its design [1].

By using bug tracking tools, we can keep track of reported software bugs with the help of bug tracking tools developer can further refine software design by making changes or upgrading the software. A bug tracking system is a software application that is designed to help quality assurance and programmers keep track of reported software bugs in their work.

Once the bug is added, it is tracked through to completion. Users of bug tracking system are generally clients, software testers, project team members, management members and software developers. It provides a facility to all the users to track and manage multiple projects concurrently. The testing team can use the bug tracking system to record issues progress, until such time that a developer could look into the other issues.

we all know, no software is perfect, it means that most of the software may require additional module or enhancement of the existing module or some of the module may contain some unnoticed or unchecked bug that are left in the software from time to time. The bug may get introduced in any phase of the software development Life cycle, i.e. requirement analysis (RA), design (SD), coding (SC), testing (ST), implementation (SI) and maintenance of the system (SM) [2].

Mostly all the main bug tracking systems allows users to insert attachments such as MS word and images in the comment so that developers can reproduce the bug easily and then take appropriate action on it immediately.

2 PROBLEM STATEMENT

2.1 Problem of manual handling

Bug reporting consists of tasks such as investigation, information gathering, testing and debugging throughout the whole process. It is very difficult to manage issues of a project manually, because hundreds of bugs can be found. Developers create the bugs, but QA team is the one who examine the code and application to discover the exact sequence or combination of steps that turns up an error.

2.2 Ineffective communication

All the system development teams need effective communication. It is not possible for anyone to keep all the bugs in one's head or in a single document such as spreadsheets. And, they will not be able to communicate effectively with one another or with the development team and thus will not be able to contribute to increase product quality. As a project grows, the first problem that is likely to be encountered is that only one person can modify the spreadsheet at a time.

Communication is a vital and deciding factor for the success of a software project.

2.3 Complex systems

Nowadays, large number of similar bug tracking systems available for this purpose; and most of them are created with unnecessary complex and confusing functions. The flow is not straight forward is a main concern. Most bug tracking systems are built in web- based application. Web-based applications can often be difficult to navigate, especially if they behave more like a sequence of HTML pages than an application. Some Web applications are constantly reloading pages because responding to user input requires a round trip between the Web server and the user's browser. It incurs slowness as involving many reloading of pages and makes the application difficult to work with.

3 LITERATURE REVIEW

3.1 Issues and Challenges in Bug Tracking System

Software development project faces multitude of challenges. One of those challenges is the defects found in the project. Defects or bugs are problems exist in the system and they are normally unavoidable in software development. A software bug is the common term used to describe an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways. Most bugs arise from mistakes and errors made by people in either a program's source code or its technical design [3].

The presence of bugs has caused the system fail to meet customer's reasonable expectation of quality. Quality is defined as meeting the customer's requirements and suitable for the use of users. Defect or bug is something that does not allow the product to meet customer requirements, or maybe does not fulfill the aspect of fitness for use. The quality problem causes the customers dissatisfactions or project failure. Defect is not an accident, but it occurs because something has not worked as planned. Since that, people realized that implementing bug tracking process is important in order to find, identify and remove bugs or issues within application and eventually keep the system running properly. Besides, the purpose of tracking bug is to improve the quality of software products; it is an eternal truth that newly written software packages will contain bugs and it is inevitable. When a system gets a bug, it can cause all sorts of problems. Bugs can cause the system to perform task incorrectly or cause things to occur that should not be. This can be especially annoying to users who may be unable to reach certain areas of the system or to perform certain operations on the system due to a hue in the system. Once a software product is released to customers, all organizations should adhere to some process of managing customers problems for the ultimate purpose of customer satisfaction.

A helpful tool is needed to ease the situation stated above. Development team needs a systematic way which can help in tracking bugs with all information about them; such system is known as bug tracking system. Development team has to solve the bugs found in order to make the system works in proper way. However huge amount of bugs can be found in a system development life cycle and bug tracking become challenging [5].

It is more challenging to start tracking bugs during the software development phases because software developers are not very cooperative in tracking down their faults. They feel that it threatens both their creativity and professional ego.

Previously, people tracked bug in a simple way. Some users reported bugs by sending email to related technical department. This is rather difficult to keep track the bug reported as the emails are scattered around email application. While developers of programs with modest popularity can accomplish manual bug tracking via email, it would be a logistical nightmare for developers of more popular software to attempt this approach. In this case, the simplest structured way to keep track of a small number of bugs is to use a spreadsheet program, such as Excel [4]. The main advantage of this approach is its simplicity and suitable for small-scale project. Spreadsheets are commonly available tools and are reasonably well documented and users found it easier to use because of its features. Besides, basic reports are also easy enough to produce in spreadsheet. However, the disadvantages of this approach are significant. Most of these are the same reasons why people use databases rather than spreadsheets. As project grows, the first problem that is likely to be encountered is that only one person can modil3' the spreadsheet at a time, users cannot modify the spreadsheet concurrently and it is not efficient. One has to wait until other user has finished viewing or editing the spreadsheet and then only able to modify it. Another, problem is that, unlike with a good database, if the computer or server running the spreadsheet crashes during modification of bug, then the data can be inconsistent.

Besides, when the total number of bugs in the spreadsheet grows into thousands, the time taken for the different reports generating can grow significantly. Therefore, using spreadsheet for keeping track of bugs should be viewed as a temporary measure because it will not scale as a project grows. The truth is that it is relatively difficult to manage bugs in simple word documents, spreadsheet or remember everything in one's head when it involves big scale of project. The information tracked in this way is difficult to maintain. Eventually the history of the debugging process or valuable information might be loss. Still, a spreadsheet is better than an email folder of reports of possible bugs or a wall full of sticky notes. but it is not scalable and not recommended for large scale project. Due to this reason, bugs tracking system is created to help QA team and development team to keep track of the reported bugs. There are many simple, low-cost bug tracking systems available that will scale better than a spreadsheet.

3.2 Trends in Bug Tracking System

Bug tracking database has been introduced to replace spreadsheet. In recent years, bug tracking systems are widely deployed in the information technology industry, especially in software development. Software and hardware companies use bug tracking systems during the development cycle to track bugs and design issues. Some use it during later phases of the product lifecycle to keep track of defect reports and to obtain quality indicators. The advantage is that this system contains database that stores all issues reported by users. Previously, many older bug or defect tracking systems are client server based and it caused inconveniences to users at the same time. Users need to install the server, and each user need to install the client software. If external users were involved, it could be problematic because of issues like firewall blocking. Also, it is not always feasible to install client software in every user's computer.

Nowadays newer systems are more likely developed in web browser based and thus no client software installation is needed. A web-based bug tracking system is especially attractive if users involved are located in different locations and are connected through the internet. With the aid of internet access, user can access real-time information from anywhere in the x'orld, 24 hours a day, 365 days a year. A bug tracking system allows users to report a detected bug directly to the concerned people who can fix it. The system maintains a database that records all facts about all defected and reported bugs. The details about a bug may include its severity, the unexpected program behavior, the identity of the user who reported it and identity of expert who is fixing it. Having a bug tracking system is extremely valuable in software development, and they are used extensively by companies developing software products.

All in all, to track bugs, many organizations still rely on Word documents and Excel spreadsheets, but these tactics are inefficient and error-prone to say the least. A good automated issue-tracking solution should streamline the process of raising, managing and fixing issues because bug tracking system should be useful software which can be designed to help software developers in keeping track of reported bug in their other software products. Besides, this can be seen as one of the form of issue tracking system.

Actually bug tracking is a misleading term because it is depending on what kinds of issues need to be tracked. The system can be called as issue tracking as issue is a broad term to describe most of the kinds of tasks might need to track when developing an application. The issue tracked might include bugs and feature-changes instead of tracking bugs or defects only. As it evolved along the time, this system is not just designed for internal teams such as developers and testers, but also for clients or other end users to report issues related to the system. This system enabled clear visibility of the software quality, and improved the planning and estimating the research and development maintenance effort internally.

3.3 Research on Other Similar Current Systems

Many companies use bug tracking system in developing software products because it is extremely useful and valuable. Bug tracking system is needed to store all related data together because most teams do not adequately maintain their requirements and design documents throughout the life of project. Bug tracking system provides important benefits to software development. Bug tracking database facilitates clear communication about bugs. Well-written, standardized reports tell the story much better than free form emails.

There are numerous bug tracking tools available both commercially and as open source. Select the best bug tracking tool that fulfill the users' requirement is not simple. There are many factors that need to be considered as follow:

3.3.1 Application setup

Application setup is the first step to be done before the system can be utilized. Bugzilla is probably the most well-known of the open source issue-management tools. It is used on many open source projects as Mozilla, Eclipse, and many Linux distributions, and is well-adapted to large, open projects. But sometimes freeware bug tracking tools might be difficult for user to configure. The freeware bug tracking tool such as Bugzilla usually takes a long time to setup or maybe not user friendly and supported. The setup process is not easy and complicated. Therefore users have difficulty to install the application in server.

Bugzilla is a tool with an uninspiring user interface and is rich in features, but undeniably cumbersome to install and to maintain. Besides, some supported that Bugzilla is easy to use and quite flexible, but it is very difficult for a novice to install and maintain. Although step-by-step directions have been provided, Bugzilla is not always easy to get working. It is frustrated if users take longer time and lots of work to setup the application in order to make it working. The setup process should be simple and straightforward.

3.3.2 Reporting process

The bug reporting process should be simple and yet useful in tracking bugs. Therefore the steps should be simplified and maximizing communication effectiveness between users. If the reporting process is complicated and not user friendly, users may hassle about using it and it will be time consuming. Since web-based bug tracking system is widely used nowadays, it incurs some problem at the same time. Web-based applications can often be difficult to navigate, especially if they behave more like a sequence of HTML pages than an application. For example, some web applications are constantly reloading pages when responding to user input requires a round trip between the Web server and the user's browser. Eventually the process might be cumbersome and users might later refuse to utilize it to track the issues related to system. The complexity of the system might reduce the efficiency of work and users might feel troublesome when use the system[6].

Some bug tracking systems are available with some rarely used features which increase the complexity of the reporting process. As a result, the process of reporting bugs or issues into a system must be easy and straightforward. Bug tracking system can be complicated or simple. Ideally it should be simple and fast so as to avoid people resorting to email or other text document. For example, a bug tracking system called, "Bug Genie", it is a powerful bug tracking system with many features. But the process of reporting bug involves many steps and not straight forward. Feedback from users who used this system stated that configuring projects or adding new issues to the tracker has been made an overly difficult task and it is overly complicated bug reporting.

If the bug tracking system is difficult to use, is time- consuming, or asks a lot of information that the author of an incident report sees no need for, use of the system will be limited or the data may be not accurate. Therefore an effective bug tracking systems can improve customer satisfaction, raise productivity and reduce downtime[7].

3.3.3 Notification method

As we all know users may not logged into system all the time, notification is a very important feature in bug tracking system. Notifications are really a workflow feature because they help you track and manage bugs. For example, notifications can keep team members informed about important changes to bugs, such as changes in priority or addition of newly added information. Notifications can also help automate the defect management process. For example, system can notify development managers when new bugs are submitted and developers when bugs are assigned to them. Most systems such as "Flyspray" and "BugTracker.NET" use email only to notify users when there are updates in the system but there is a possibility that emails might be blocked by spam filter and not delivered to the designated recipients. In this case, email notification might not be a reliable channel to notify users [8].

4 EXISTING BUG TRACKING TOOLS AND THEIR DRAWBACKS

In this section, we give an overview of existing bug tracking tools and we criticize the existing solutions from the point of their usability

4.1 Bugzilla

Bugzilla is very popular, actively maintained and free bug tracking system, used and developed together with Mozilla, giving it considerable credibility. Bugzilla is based on Perl and once it is set up, it seems to make its users pretty happy. It's not highly customizable, but in a odd way, that may be one of its features: Bugzilla installations tend to look pretty much the same wherever they are found, which means many developers are already accustomed to its interface and will feel they are in familiar territory. Bugzilla has a system that will send you, another user, or a group that you specify the results of a particular search on a schedule that you specify. Bugzilla has a very advanced reporting systems and you can create different types of charts including line graph, bar graph or pie chart.

4.2 Mantis

Mantis is a free web-based bug tracking system. It is written in the PHP scripting language and works with MySQL, MS SQL, and PostgreSQL databases and a webserver. Mantis can be installed on Windows, Linux, Mac OS and OS/2. Almost any web browser should be able to function as a client. It is released under the terms of the GNU General Public License (GPL). The main complaint is its interface which doesn't meet modem standards. On the other hand, is easy to navigate, even for inexperienced users. There not exists some advanced features such as charts and reports. In short, the whole system is sloppily done, there are plenty of bugs and very little functionality.

4.3 BugTracker. NET

BugTracker.NET is a free, open-source, web-based bug tracker or customer support issue tracker written using ASP.NET, C#, and Microsoft SQL Server Express. BugTracker.NET is easy to install and learn how to use. When you first install it, it is very simple to setup and you can start using it right away. Later, you can change its configuration to handle your needs. It has a very intuitive interface for generating lists of bugs. It has two very useful features. First of them is a screen capture utility that enables you to capture the screen, add annotations and post it as bug in just a few clicks. The second feature is the fact that it can integrate with your Subversion repository so that you can associate file revision checkins with bugs.

4.4 Flyspray

Flyspray is a web-based bug tracking system written in PHP. Flyspray is free software, released under the General Public License. This essentially means that you can get Flyspray and use it free of charge. The source code is available, and everyone are welcome to modify it to suit their needs. Its web pages describe it as "uncomplicated", and the list of features includes: multiple database support (currently MySQL and PGSQL), multiple projects, 'watching' tasks, with notification of changes (via email or Jabber), comprehensive task history, CSS theming, file attachments, advanced search features, RSS/Atom feeds, wiki and plaintext input, voting, dependency graphs.

4.5 Redmine

Redmine is a flexible web-based project management web application. Written using Ruby on Rails framework, it is cross-platform and cross-database. Redmine is open source and released under the terms of the GNU General Public License. Redmine is flexible issue tracking system. You can define your own statuses and issue types. It supports multiple projects and subprojects.

Each user can have a different role on each project. Interface is very simple, intuitive and easy to navigate. Shortly, this is very good product and our recommendation.

4.6. Bug-Track

Bug-Track is web-based defect and bug tracking software allows you to document, manage and assign all of your bugs and tasks and empowers you to organize your bugs, defects or issues into distinct projects. It can run on virtually any web-server like Microsoft, Linux, Unix, etc... Since it is an commercial application it is expected that it is better than other free products. But it isn’t true. He has nothing new and better than other free bug tracking systems. One better thing is fact that he have more intuitive interface then others and that is his only benefit.

4.7 Bugzero

Bugzero is a web-based bug, defect, issue and incident tracking software. Its single code base supports both Windows and Unix (based on Java™) and supports database systems including Access, MySQL, SQL Server, Oracle, and etc. Bugzero can be customized for software bug tracking, hardware defect tracking, and help desk customer support issue and incident tracking. Bugzero have intuitive interface but he lacks form features. The main drawback is the fact that Bugzero is an commercial product and you can find much better product for free.

5 TREND AND THE MOST FAVOURABLE SOLUTION FOR FUTURE

When a user submits a bug report, he is asked many questions: What is the name of the product? In which plugin/component? What is the Build ID? What is the bug about? What are the steps to reproduce the bug? Any additional information?

However, the initial information provided in a bug report is often incomplete and developers often have follow-up questions: Do you have flash installed? Is there any screenshot? Getting replies by users takes time (often weeks) and slows down the progress on open bugs. And this makes the whole process sluggish and slows down the process of bug tracking.

Fig: comparison of different Bug Tracking Tools

As you can see form the Table given above, the most of the selected criteria are not supported in current bug tracking systems. The current bug tracking systems rely on some different selection criteria, which sometimes cannot give satisfying results. So, we believe that including selection criteria like stack traces, steps to reproduce, observed behavior, expected behavior, test cases, screenshots and dependencies will improve current hug tracking systems.

	Stack trees	Steps to reproduce	Expected behaviour	Test Cases	Screen shots	Dependencies
Bugzilla	no	no	No	no	no	no
Mantis	no	yes	No	no	no	no
Bugtracker.net	no	no	No	no	yes	yes
Flyspray	no	no	No	no	no	no
Redmine	no	no	No	no	no	no
Bugzero	no	no	no	yes	no	yes

The advantages of the improved set of criteria will give user better information about current bug. This will lead to faster locating of the current defect and to the faster respond to it.

6 CONCLUSION

Existing bug tracking systems do not effectively accumulate 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 issue tracking systems collect information are needed.

We summarized criteria that are used in modem bug tracking systems. Such criteria often doesn’t give appropriate results in describing bug. So, we proposed an improved set of criteria that will give much more satisfying solution for the current system.

This work can be important to the designers of the future bug and defect tracking systems. They should know importance of selection criteria for describing bug, because a well described bug will be easier to be trace and solved.

REFERENCES

[1] Prof Dr. Muhammad Younus Javed and Hufsa Mohsin, 2012 on "An Automated Approach for Software Bug Classification" in Complex, Intelligent and Software Intensive Systems (CISIS) 2012 page 414-419.

[2] V.B Singh, Krishna Kumar Chaturvedi on "Bug Tracking and Reliability Assessment System" international Journal of Software Engineering and Its Applications Vol. 5 No. 4, October, 2011 page 17-30.

[3] D'Ambros, M. Lugano Univ., Lugano Lanza, M.; Pinzger, M.on "A Bug's Life" Visualizing a Bug in Visualizing Software for Understanding and Analysis,2007Page(s): 113-120

[4] Sascha, Rahul Premraj and Thomas Zimmermann on 'Towards the Next Generation of Bug Tracking Systems' in Visual Languages and Human-Centric Computing, 2008 on Page(s):82-85

[5]Catherine V. Stringfellow, Dileep Potnuri on "Analysis of Open Source Defect Tracking Tools for Use in Defect Estimation" Software Engineering Research and Practice 2005: page no.296-301

[6]Davidson, J.L., Mohan, N.; Jensen, C. on "Coping with Duplicate Bug Reports in Free/Open Source Software Projects" in Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on Page(s): 101 – 108

[7]Liguo Yu, Stephen R. Schach, Kai Chen on "Measuring the Maintainability of Open-Source Software" in Empirical Software Engineering, 2005. 2005 International Symposium on.

[9]Thomas Zimmermann, Rahul Premraj, Nicolas Bettenburg, Sascha, Adrian Schrter, and Cathrin Weiss on What Makes a Good Bug Report? In international symposium on Foundations of software engineering page 308-318

Author Profile

Author is a post graduate student in Department of Computer Science and Engineering at Guru Nanak Dev University Amritsar, India. He is pursuing his research work under the guidance of Prof. Hardeep Singh. His interests include software engineering, software reliability, and development tools. Contact him at sandiz_28@yahoo.co.in.