

E-BUG TRACKING SYSTEM

GUIDE: Mrs. Sathya Priya R¹

R.B. Babu², R. Marimuthu³, G. Gowtham⁴, V. Prakash⁵

¹Assistant Professor, KSR Institute for Engineering and Technology, Tiruchengode.

^{2,3,4,5}Department of Computer Science And Engineering,
KSR Institute for Engineering and Technology, Tiruchengode.

ABSTRACT

This is the world of information. The ever-growing field Information Technology has its many advanced notable features which made it what it was now today. In this world, the information has to be processed, clearly distributed and must be efficiently reachable to the end users intended for that. Otherwise, we know it led to disastrous situations. The other coin of the same phase is it is absolutely necessary to know any bugs that are hither-to face by the end users. The project “e-bug tracking system” aims to provide the solution for that. The Bug Tracker can be made from any two types. The first one being the system side, the other being the services side. Our project deals with the second one. The paper is wholly dedicated to tracking the bugs that are hither-by arise. The administrator maintains the master details regarding to the bugs id, bugs type, bugs description, bugs severity, bugs status, user details. The administrator too has the authority to update the master details of severity level, status level, etc, modules of the paper. The administrator adds the users and assign them responsibility of completing the paper. Finally, on analysing the paper assigned to the particular user, the administrator can track the bugs, and it is automatically added to the tables containing the bugs, by order of severity and status. The administrator can know the information in tact the various papers assigned to various users, their bug tracking status, their description etc in the form of reports from time to time. The paper wholly uses the secure way of tracking the system by implementing and incorporating the Server-side scripting. The administrator can now add the project modules, project descriptions etc. He too adds the severity level, its status etc. The whole beauty of the paper is its high-level and user-friendly interface which mean that is the well based Bug Tracker which helps in tracking the whole system by providing the efficient reporting system. The Bug Tracker can be further by analysed and further relevant and quick decisions can be taken.

This defect tracking system helps to track bug. There are three modules in this tracking system, Administrator, Staff and Customer. The Administrator can login to the app and can enter the details of staff, project, view bugs send from the customers. The admin can also assign work to staffs, view bug case flow status details, send messages to customers using this bug tracking application. The staff can login to the site using username and password. Then he/she can view the bugs assigned to them. He can directly give solution message to customers or he/she can assign the bugs to other staffs if the bug is related to them.



I: INTRODUCTION

This is the world of information. Bug and issue tracking systems are often implemented as a part of integrated project management system. This approach allows including bug tracking and fixing in a general product development process, fixing bugs in several product versions, automatic generation of a product knowledge base and release notes. Some bug trackers are designed to be used with distributed revision control software. These distributed bug trackers allow bug reports to be conveniently read, added to the database or updated while a developer is offline. Distributed bug trackers include Fossil. Recently, commercial bug tracking systems have also begun to integrate with distributed version control. Fog Bugz, for example, enables this functionality via the source-control tool, Kiln. Although wikis and bug tracking systems are conventionally viewed as distinct types of software, ikiwiki can also be used as a distributed bug tracker. It can manage documents and code as well, in an integrated distributed manner. However, its query functionality is not as advanced or as user-friendly as some other, non-distributed bug trackers such as Bugzilla. Similar statements can be made about org-mode, although it is not wiki software as such.

1.1 BUGS AND ITS EFFECTS

In any software development bugs are inevitable. Let it be in any kind of product bugs arise at any phase of development. One has to take a great care in the proper maintenance and resolution of the bugs. In the Existing system the bugs are not properly maintained and they are simply relied on shared lists and email to monitor the bugs. In this type of system, it becomes difficult to track a bug if a bug is over looked then it may cause tremendous errors in the next phase and can improve the cost of project whatever necessary effort spent on the bug maintenance may not be worthy. So bug history has to be maintained properly. And there is no efficient search technique. One has to search the whole database for the details of particular bug which might have occurred sometime earlier. It is both time consuming and error prone. And it is very difficult to share the bug among several users as there is no proper maintenance of the bugs. In order to have an efficient product bugs must be maintained properly and should be resolved in time both to reduce time and money spent on the development.

1.2 PROPOSED METHOD

Bug Tracking System is an ideal solution to track the bugs of a product, solution or an application. Bug Tracking System allows individual or groups of developers to keep track of outstanding bugs in their product effectively. This can also be called as Defect Tracking System. The Bug Tracking System can dramatically increase the productivity and accountability of individual employees by providing a documented work flow and positive feedback for good performance.

For many years, bug-tracking mechanism is employed only in some of the large software development houses. Most of the others never bothered with bug tracking at all, and instead simply relied on shared lists and email to



monitor the status of defects. This procedure is error-prone and tends to cause those bugs judged least significant by developers to be dropped or ignored.

In this type of system it becomes difficult to track a bug if a bug is over looked then it may be cause tremendous errors in the next phase and can improve the cost of project whatever necessary effort spent on the bug maintenance may not be worthy. So bug history has to be maintained properly. And there is no efficient search technique. One has to search the whole database for the details of particular bug which might have occurred sometime earlier. It is both time consuming and error prone. And it is very difficult to share the bug among several users as there is no proper maintenance of the bugs.

1.2.1 ADVANTAGES

Bug Tracking System plays a vital role in the testing phase. But it supports assigning projects for the developer, tester by the project manager. The Bug Tracking System maintains the different users separately i.e., it provides separate environments for project manager, developer and tester.

This system maintains the products, Bugs and bug tracking. It has advantage of maintaining bug history it stores all the details from bug origin to bug resolution. Each product can have versions for easy maintenance of the product and all the user of the product is stored in the database. It provides the advantage of maintaining users to the bugs and resolutions provided by them. Our System provides the searching based on status, priority, and operating system.

It provides with user and bug hierarchy, which would be helpful in knowing the relation between bugs and users allotted to the bug. It is provided with a fully authenticated system with password encryption and has the facility for storing attachments for a bug. One can keep a track of the bug in a product with much lower cost and effort. The most advantage of this system is maintaining log records which are helpful in knowing any errors or misuse of the system by other users.

2. IMPLEMENTATON PROCEDURE

Implementation is the stage, which is crucial in the life cycle of the new system designed. The main stage in the implementation is planning, training, system testing. Implementation is converting a new or revised system into an operational one. Conversion is the main aspect of implementation. It is the process of changing from the old system to the new one. After system is implemented, user conducts a review of the system. It is used to gather information for the maintenance of the system. The basic review method is a data collection Current bug tracking system, user do not effectively elicit all of the information needed by developers. Without this information to resolve bugs form the given system software. To implement it we design the new technique which is implementing online process. From that user can able to resolved bug quickly. It depends on the amount of data is present to perform the bug tracking software.

If data is less, then bug can be track quickly as compare to the huge or more amounts of data. It depends on the variations of the program code. To make the implementation of bugs more secure and perform fast, we working the bug tracking system in four ways. Which are Tool-centric, Process-centric, User-centric and Information-centric. In Tool-centric it can help to reduce the burden of information collection and provision. In Process

centric it focuses on administration of activities related to bug fixing. In User-centric, it includes both reporters and developers to providing information by the user to be used to resolve the bugs.

2.1 INFORMATION-CENTRIC AND ITS HISTORY

In Information-centric, directly focuses on the information providing by the reporter. Online Bug tracking system is embedded with tools such as CUEZILLA that provides the real-time feedback on the quality information provided and what can be added to increase value. To remove the duplication of the bugs N. Jalbert and W. Weimer proposed a system that automatically deletes duplicate bug reports and saves the time. They applied surface feature, texture semantics and graph clustering to detect duplicate bug reports. There was eight percent reduction in the bug report caused due to filtering of duplicate bugs. Some systems like Bugzilla, Mantis and Trac etc. provides the open source bugs tracking system but they not uses this technique. So, this technique provides the extra feature for user to enhance the software quality.

2.2 EFFICIENCY

In our work to support and implement all types of data and codes we uses five open source tools such as Flyspray, Jtrac, Mantis, phpBugTracker and Web issues are uses. Comments, create graphs, customized theme, customized workflow, dependencies, email notification, export files, failure description, file attachments, history view, Multilanguage support, reminder, severity, status and version are the features considered for the analysis of the tools. These tools can manages there bugs through their life cycle, for making initial report to implementing final resolution.

A major component of a bug tracking system is a database that records facts about known bugs. Facts may include the time a bug was reported, its severity, the erroneous program behavior, and details on how to reproduce the bug; as well as the identity of the person who reported it and any programmers who may be working on fixing it.

2.3 BUG TRACKING SYSTEM

Typical bug tracking systems support the concept of the life cycle for a bug which is tracked through the status assigned to the bug. A bug tracking system should allow administrators to configure permissions based on status, move the bug to another status, or delete the bug. The system should also allow administrators to configure the bug statuses and to what extent a bug in a particular status can be moved. Some systems will e-mail interested parties, such as the submitter and assigned programmers, when new records are added or the status changes. It is possible to perform automated diagnosis based on the content of the bug report. For instance, one can do automated detection of bug duplicates.

2.4 COST ESTIMATION AND SCHEDULING

Cost Estimation can be made either top-down or bottom-up. Top-down estimate starts with system level costs, work out the costs of computing resources, development staff, configuration management, quality assurance, system integration, training and publications. Constructive Cost Model (COCOMO) is the top-level model. Basic COCOMO is applicable to large majority of software project. The Paper titled ” Bug Tracking and

Reporting System” adopts the Cost Estimation in a planned and full-fledged manner. It follows the Cost-Benefit analysis while making the Cost Estimation of the implementation of the wireless based communication system. Input design is the process of converting user-oriented inputs to a computer-based format. The quality of the system input determines the quality of system output. Input design determines the format.

2.2 QUALITY MANAGEMENT

In order to make the reader understand the covered subject, a composition with some theories foundation will be presented next. The motivation for the existence of software tests is to ensure the quality

3. BENEFITS

DELIVER HIGH-QUALITY PRODUCT

A bug tracking software helps to track all the issues, bugs of the product and make sure that all the detected bugs are fixed. Having a single source of truth is very important while running large projects. There are multiple moving parts and seamless integration coupled with agile methodology is critical to the product delivery.

Having the right bug tracking tool helps with in-depth bug analysis, provides visibility to the overall direction of the project & enables proactive implementation of corrective measures in a timely manner. All of which are crucial in delivering a high-quality product on time and within budget.

IMPROVE RETURN ON INVESTMENT (ROI) BY REDUCING THE COST OF DEVELOPMENT

A bug tracking software helps with bug prioritization and assignment based on severity and importance. This helps the development team to focus on high priority bugs first instead of low priority ones. Thus, with major hurdles out of the way earlier in a development cycle augurs well for your development team and the overall success of the product.

As a result of bug tracking, identifying trends becomes easier & leads to quicker root cause identification and resolution. Your bug tracker is also a knowledge powerhouse for future references leading to fewer iterations & reduced development efforts. This eliminates unwanted delays in your project delivery, leads to a significant rise in team productivity and reduces the overall cost

BETTER COMMUNICATION, TEAMWORK AND CONNECTIVITY

With task collaboration, in-app chat, and email notifications, a bug tracking tool keeps everyone on the same page. You can assign the right resources to test or fix bugs on time. Ease of access and use allows for greater collaboration among teams. Real-time access, review, update & bug reports eliminate communication gaps, improves transparency & accountability.

Teams are forthcoming in brainstorming & troubleshooting issues rather than wasting time on finger-pointing and shying away from responsibility. Most importantly, everyone is aware of the progress, aligned with the overall project plan & work towards the common goal.

DETECT ISSUES EARLIER AND UNDERSTAND DEFECT TRENDS

The biggest benefit of using a bug tracking tool is that it allows companies to track all the bugs at one centralized location – who reported the bug, who fixed them, what the priority is and how long it took to fix.

Using a tool helps to analyse the bug trends and it can be referred in future by the development team to enhance the product prevent recurring issues.

Insightful reports around defect trends, defect age, open and closed bug trends help inadequate resource planning and budgeting too. You can account for these aspects while bidding for new projects & set the right stakeholder expectations for consistent customer satisfaction.

BETTER SERVICE AND CUSTOMER SATISFACTION

A bug tracking tool allows collaborating among your QA & Dev teams, managers & customers. They can directly login to the system to report a bug. Most of the tools are designed in such a way that a user can easily create bugs with all the required information needed for a developer. The QA team, developer, managers or customer will get status updates of the reported issues via instant email notifications or inside the tool in a single view.

Besides that, the stakeholders can provide further clarification, suggestion, feedback if any on their reported issues. So, it enhances the service you provide and clarity for the development. A direct involvement of customers through bug tracking software in testing phase results in happy and satisfied customers. They will be happy to see less or no bugs to use their product efficiently

4. DEFINING THE PROBLEM

The problem in the older system can be defined as the whole project maintenance, user's maintenance and their assignment have to be maintained manually. The Software development companies have to face a lot of problems while maintaining manually all the maintenance of the projects, their bugs and their status. This type of problem makes the whole system an inefficient one and thus making a poor and unorganized working. In order to remove this type of problem, so that the paper is planned to develop. Bug tracking software is a "Defect Tracking System" or a set of scripts which maintain a database of problem reports. Bug tracking software allows individuals or groups of developers to keep track of outstanding bugs in the product effectively. Bug tracking software can track bugs and changes, communicate with members, submit and review patches, and manage quality assurance.

This Android-based business application is a great tool for assigning and tracking issues and tasks during software development and any other projects that involve teams of two or more people.

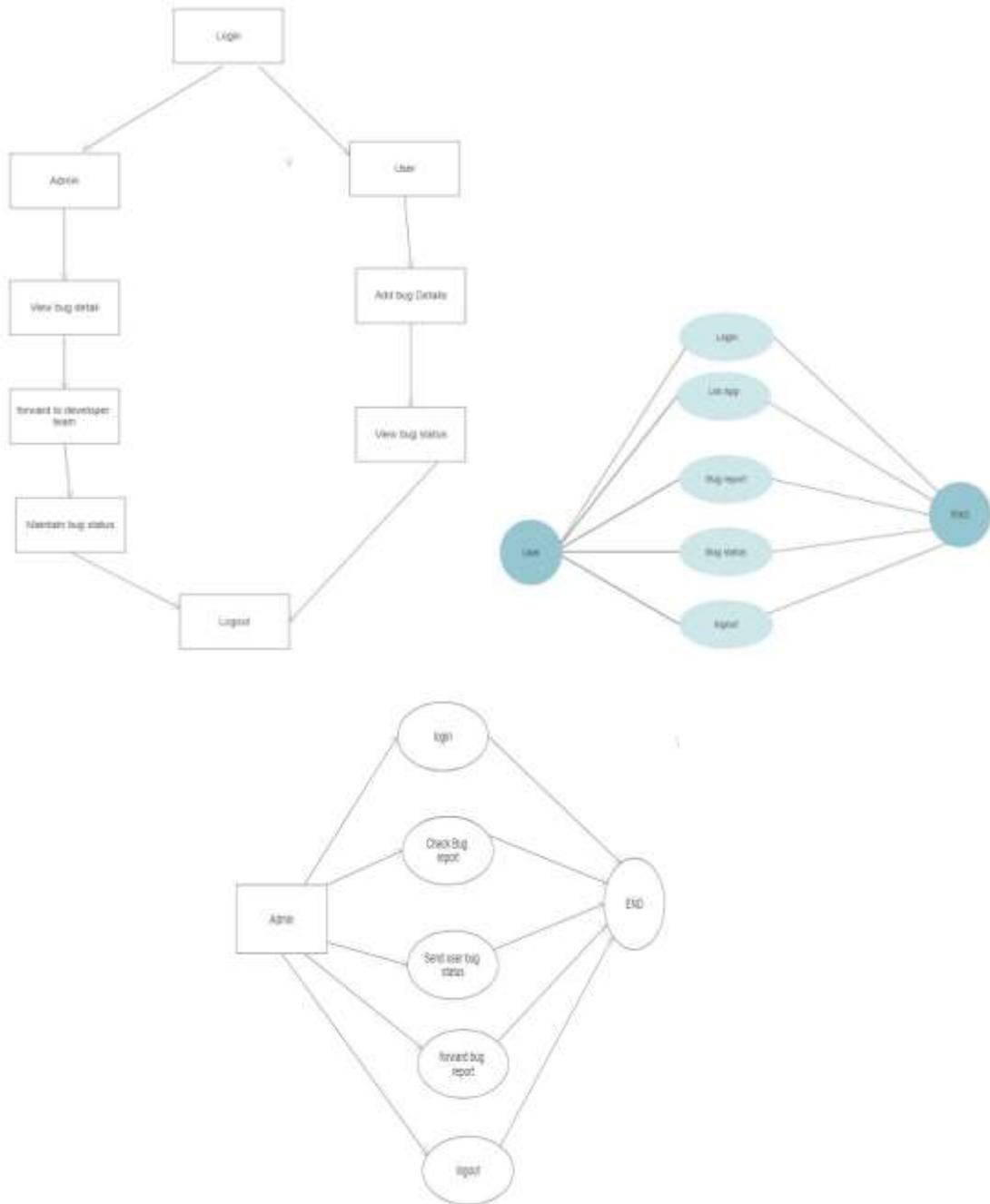
ADMIN MODULE:

The administrator too has the authority to update the master details of severity level, status level, etc, modules of the project. The administrator adds the users and assign them responsibility of completing the project. Finally, on analysing the project assigned to the particular user, the administrator can track the bugs, and it is automatically added to the tables containing the bugs, by order of severity and status.

USER MODULE:

User has to send the bug details of the application and also view the status of the bug can be fixed or not. If bug already intimate or already fixed user receive the notification about the bug details

FLOW DIAGRAM:





5.TABLE DESIGN

A. PROJECT BUG

Field name	Data type	Description
Bugid	Number	Not Null
Bugname	Text	Not Null
Proid	Number	Not Null
Proname	Text	Not Null
Staid	Number	Not Null
Sevid	Number	Not Null

B. PROJECT STATUS

Field name	Data type	Description
Username	Text	Not Null
projname	Text	Not Null
projstatus	Text	Not Null

C.BUG TYPE

Field name	Data type	Description
typid	Number	Primary key
typname	Text	Not Null
typdesc	Text	Not Null

DEVELOPING SOLUTION STRATEGIES

The main objective of the proposed system is to full analyze the bugs and report the same to the administrator in an efficient manner so that he can get right information at right times. The paper objective is to fully systemize

everything so that the possibilities of bugs should be reduced at all levels.

APPLICATION DETAILS

This module lets the user to view and manage the list of application assigned to a user. It only shows those application which the user will access. The user can sort this list by various parameters like last updated. It also shows the details of the application like current working team, application id, application status, etc.

DEVELOPER MANAGEMENT

This module allows adding, editing, viewing and deleting the details of developer from the list of developers. It also lets the user to grant and revoke permission to access a project application.

DEVELOPER PERFORMANCE

This module is to track and generate report on the performance of a developer. The performance of the developer is based on total number of bugs fixed, average time taken to fix a bug, etc

TESTER MANAGEMENT

This module allows the user to add, view, edit, and delete a tester from the system. It allows assigning and revoking permission from a tester to test an application.

BUG TRACKING AND STATUS MANAGEMENT

At this module the user will be able to update bug details that have been generated at the application. After adding the bug details, a unique bug issue id will be generated. Keeping the reference of the unique id, the bug status can be tracked.

BUG UPDATE

This module allows to create a new bug entry in system along with detailed information about the bug which includes bug id, bug name, bug priority, project name, bug location, bug type, cause of the bug, screenshot image of the bug, data used to generate bug or situation that caused the bug, error id (if any).

BUG REPORT

This module will be accessed by the developer user to get update on bugs that has been traced by the tester at the application. The user will be able to access complete details of the bug.

BUG FOLLOW UP

This module helps to keep track of all the bugs that have been assigned to a user. It displays all the changes that have been made to the bug report since the creation of the bug.

APPLICATION STATUS

This module allows tracking the current status of the application like developer team, testing team, reported bugs, fixed bugs, etc. and also displays the status of bugs present in the application.

REPORT

In this module the project manager will be able to generate reports at various stages like application being tested, bugs tracked by tester, bugs resolved by developer, developer performance, tester performance periodically.

PROFILE MANAGEMENT

This module holds the account information for the user. It has facility for the user to update profile details like name, email, contact no, etc. Using this module user can change the password periodically or whenever required.

LOGIN

This module is responsible to authenticate user with login credentials entered. This module validates the input with the database and allows the valid user to access proper and respective control panel. After validating the user, it also creates a session for the active user. Before validating the credentials, it encrypts the input with proper algorithm.

PRIORITIZE BUG BASED ON SEVERITY

Bugs can be set by the tester to ensure that the important bugs are given a higher preference by the developer to fix them as soon as they are found.

6 CONCLUSION

This Paper Bug Tracking and Reporting System helps a Software Concern to detect and manage the bug in their products effectively-efficiently. Utilizing bug tracking software can assist in troubleshooting errors for testing and for development processes. With the ability to provide comprehensive reports, documentation, searching capabilities, tracking bugs and issues, bug tracking software is a great tool for those software development needs. Depending on your development needs and the bug tracking software, you can hope to gain several benefits from bug tracking software. Some of the benefits are:

- 1) Improve communications between groups of people
- 2) Increase the quality of the software
- 3) Improve customer satisfaction with bug free software
- 4) Provides a form of accountability
- 5) Increases overall productivity



A good Bug Tracking System (BTS) will do the task like reporting, managing and fixing the bugs (if any). We are trying our best here in this work to build an automated Bugs Tracking System which will help the developers and user to choose and to find out the bugs as per their requirement and constraints.

REFERENCES

- [1] Bill Evjen, Thiru Thangarathinam, Bill Hatfield, „Professional ASP.NET 1.1
- [2] Dave Mercer, „ASP.NET – A Beginners Guide□ , O□ Reilly Publications
- [3]http://en.wikipedia.org/wiki/Bug_tracking_system
- [4] Jonathan Corbet (May 14, 2008). "Distributed bug tracking". LWN.net <http://lwn.net/Articles/281849/>
- [5] Joey Hess (6 April 2007). "Integrated issue tracking with Ikiwiki". LinuxWorld.com. <http://www.linuxworld.com/news/2007/040607-integrated-issue-tracking-ikiwiki.html>. Retrieved 7 January 2009.
- [6] Kevin Loney, George Koch (2003), „Ms-Access –The Complete Reference□ , Tata McGrawHill Publishing Company Limited.
- [7] J. Aranda and G. Venolia, “The secret life of bugs: Going past the errors and omissions in software repositories”, In ICSE’09 Proceedings of the 31st International Conference on Software Engineering, 2009.