

An Empirical Study of Test Management in Bug Tracking System

M. Rajeswari (M.C.A)

Research Scholar, Department of Master of Computer Applications
Nadar Mahajana Sangam S. Vellaichamy Nadar College, Madurai

Dr. K. Parimala (M.C.A, M.B.A, Ph.D)

Assistant Professor, Department of Master of Computer Applications
Nadar Mahajana Sangam S. Vellaichamy Nadar College, Madurai

Dr. X. Joshphin Jasaline Anitha (M.C.A, M.Phil, Ph.D)

Assistant Professor, Department of Master of Computer Applications
Nadar Mahajana Sangam S. Vellaichamy Nadar College, Madurai

ABSTRACT

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.

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.

1. Introduction

The aim of this paper is to prepare the design and functionality of the CAC's Test Management and Defect Management application. This document describes the prototype and database structure of CAC Test Management tool. Our intentions are to produce a user friendly and customizable application according to their business needs.

This document describes the operational use of the application to the project stakeholders (like Business Analyst, Solution Architect and Development Team) and helps them in effective communication, Test

estimation, Reuse, effective tracking, able to see assigned test cases and reported defects as well as history of test cases used in different projects.

2. Literature Survey

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 any software development bugs are inevitable. Let it be any kind of product bugs arise to any phase of development. One has to take a great care in 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 overlooked 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.

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.

3. Previous System

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.

4. Proposed System

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. It is shown in the below figure1.

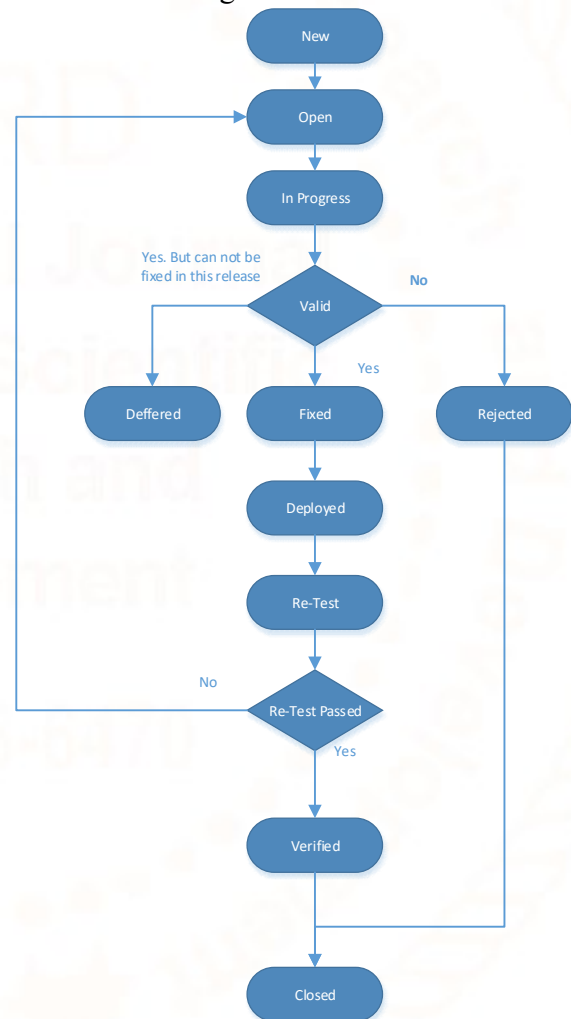


Figure 1 CAC - Bug Life Cycle

New: Possible Bug or Test Observation that is raised and yet to be authorized by the Development team.

Open: Assigned to development team but not yet resolved

In Progress: The Bug is been addressed by the developer and examination is in progress. This stage results in two possible outcomes which are: Fixed or Solved or Rejected

Deferred: When a bug cannot be fixed in that particular release it is deferred to future release

Rejected: When a bug is invalid it is rejected citing “Not a Bug”

Re-Test: The Bug is fixed and is ready for Re-Test

Re-Open: If the Re-Test failed then the bug is Re-Opened and sent back to the development team

Verified: The Bug is retested and retest passed

Closed: It is the final stage of the Bug that can be performed only when the Verification or Rejection is complete and validated by the test lead.

5. Experimental Result

CAC’s Test Management and Bug Tracking System is a Windows Forms Application which has been written in C#.NET with the .NET Framework 4.0.

The platform used to design the application is Microsoft Visual Studio 2010 while the database used is MS Access 2013 to store data such as the new test cases or defects created on the front-end. The MS Access database also helps retrieve values and display into the application for the purpose of amending defect statuses and displaying a graph for the report.

Test Case ID	Test Case Description	Expected Results
TC_001	Verify that the CAC tool allows the authenticated users	Valid users are able to login
TC_002	Verify that the CAC tool rejects the unauthenticated users when wrong login or password entered by uses and displays the error message	In Valid users are rejected by the CAC tool and error message is displayed
TC_003	Verify that the CAC tool rejects when mandatory fields in Test Case and Defects are not filled	rejected by the CAC tool and error message is displayed
TM_003	Verify that the application should be customizable for adding new Test Suite, new Project IDs, new Sub Project IDs, new releases, new environments and new reports	CAC Test Manager Tool Admin
TM_004	Verify that the application allows to create new test cases and modify the existing test cases	Test Team
TM_005	Verify that the application allows to create new defects and modify the existing	Test Team, Development Team, Business Analyst and Solution Architect
TM_006		Test Manager

5.1. Test Case Design and Development

6. Conclusion

We concluded that CAC Test Manager as a tool will ease the communication process between the various project stakeholders and also make testing process more effective by implementing the work flow in the tool. Configuration Management for test cases will be very useful when version control process is followed in the tool. CAC Test Management tool also supports senior management in monitoring the testing metrics and defect metrics dynamically and taking critical decisions about the delivery of project and continuous improvements in the software delivery process (using defect cause).

References

- [1] Yongsoo Yuk, Woosung Jung, Comparison of Extraction Methods for Bug Tracking System Analysis, National Research Foundation of Korea(NRF)
- [2] Tao Zhang & Byungjeong Lee*, A Bug Rule based Technique with Feedback for Classifying Bug Reports, 2011 11th IEEE International Conference on Computer and Information Technology.
- [3] Thomas Zimmermann, Rahul Premraj, Jonathan Sillito & Silvia Breu, Improving Bug Tracking Systems.
- [4] <https://www.bugzilla.org/>
- [5] <http://www.redmine.org/>
- [6] <https://www.mantisbt.org/>
- [7] Tomi Prifti, Sean Banerjee, Bojan Cukic, Detecting Bug Duplicate Reports through Loca References.
- [8] Anahita Alipour, Abram Hindle and Eleni Stroulia, A Contextual Approach towards More Accurate Duplicate Bug Report Detection, IEEE, 2013.
- [9] Naresh Kumar Nagwan & Pradeep Singh, Weight Similarity Measurement Model Based, Object.
- [10] Oriented Approach for Bug Databases Mining to Detect Similar and Duplicate Bugs, International Conference on Advances in Computing, Communication and Control (ICAC3'09).