IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DEFECT TRACKING FOR IMPROVING PRODUCT QUALITY AND PRODUCTIVITY – WEB APPLICATION

[1]E Suresh, [2]G Bindu, Dr A Althaf Ali

[1]PG Scalar, [2]PG Scalar, [3]Assistant Professor.

[1]Master of Computer Applications,

[1]Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh, India, 517352.

Abstract: Defect Tracking for Improving Product Quality and Productivity for Improving Software Reliability is an automated system that can be useful to workers and the directors in any functional association. disfigurement Tracking for perfecting Product Quality and Productivity gives the installation to define the tasks in the association and also allows the directors to track the blights spent by the inventor for that particular task. A report generation installation is supported in DTS that allows the directors to assay which are those chops by hand are employed and those which aren't employed design or operation. This tool helps workers to validate their blights and analysis.

This system is a web operation which works an intranet operation(1). client resource operation is integrated with this operation which makes this software more effective in terms of time operation, and reducing homemade workshop(2). Detailed explanation about this design is handed in design report.

Keywords: Defect Tracking, Product Quality, DTS. Report Generator, Manager Tracker, Quality Analysis, Productivity Reports

I. INTRODUCTION

Defect Tracking for Improving Product Quality and Productivity for Improving Software Trustability is an automated system that can be useful to workers and the directors in any functional association. disfigurement Tracking for perfecting Product Quality and Productivity gives the installation to define the tasks in the association and also allows the directors to track the blights spent by the inventor for that particular task. A report generation installation is supported in DTS that allows the directors to dissect which are those chops by hand are employed and those which aren't employed. design or operation. This tool helps workers to validate their blights and dissect.

also, this system provides the installation of viewing the status of the blights to the workers whether it's completed or still pending or rejected. However, also they've to mention the reason duly, If rejected. Periodical reports can be generated by the admin and director.

This system has been developed using html and css with JavaScript Each bug in the system may have an urgency value assigned to it, grounded on the overall significance and the unborn circumstance of that bug. Low or zero urgency bug are minor and should be resolved as time permits. Other details of bugs include the client passing the rise of bug, date of submission, detailed descriptions of the problem being endured, tried results and other applicable information. As work is done on that bug, the system is streamlined

with new data by the technician. Any attempt at fixing the problem should be noted in the bug system, as each bug maintains a history of each change.

II AN OVERVIEW OF HTML AND CSS:

Html: HTML is a luxury language that's used to produce web runners. It defines how the web runner looks and how to display content with the help of rudiments. It forms or defines the structure of our Web runner, therefore it forms or defines the structure of our Web runner. We must flash back to save your train with. html extension. In this HTML Tutorial, we 'll understand all the introductory generalities needed to protest- start your trip in HTML.

Css: CSS (Casecade Style wastes) is used to term and lay out web runners — for illustration, to alter the fountain, color, size, and distance of your content, resolve it into multiple columns, or add robustness and other ornamental features. This module provides a gentle morning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

III PROPOSED SYSTEM:

The Proposed system is a browser which is completely related to online system, which provides the centralized database. It stores Defects data and description of the particular Defect data. It can also create reports and documents based on the information in its database.

Defect management is crucial to closing the loop between requirements, implementation and verification and validation. Traditional defect tracking management, implemented in a standalone method, can no longer address the complexity and pace of change in modern software development. Defect management processes must be strongly interlinked with all of the other software development processes. The defect management process contains the following elements:

3.1 Advantages:

- The performance is increased due to well-designed database.
- Security is increased
- Time saving in report generation
- Easy to update the details

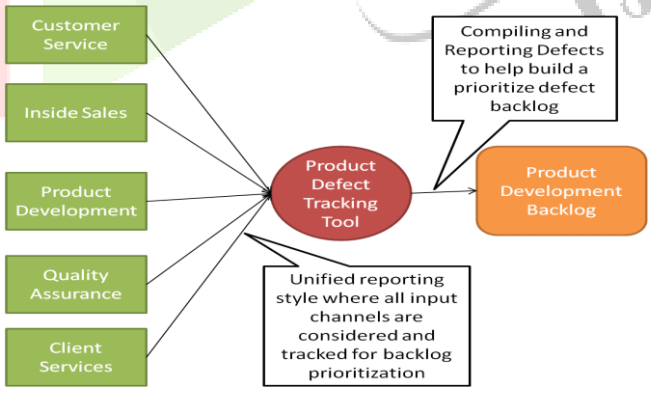

Figure 1: Block Diagram

VI LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the

programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

[1]. Towards Effective Troubleshooting with Data Truncation.

Towards Effective Troubleshooting with Data Truncation deals with reducing the data present in the bug depository and ameliorate the quality of data also reduce time and cost of bug triaging, it represents an automatic approach to prognosticate a inventor with applicable experience to break the new coming report. The bug data sets are attained and ways similar as case selection point selection are applied contemporaneously. The top k pruning is applied for perfecting results of data reduction quality, carrying sphere wise bug result. Instance selection is for carrying a subset of applicable cases(i.e., bug reports in bug data). It's used to Remove noise and spare cases, Remove non-representative cases. point selection which aims to gain a subset of applicable features (i.e., words in bug data), Sorting of words according to point values. Top- K pruning algorithm for perfecting results of data reduction quality.

[2]. Technique to Combine Feature Selection with Instance Selection for Effective Bug triage.

A fashion to Combine Point Selection with Instance Selection for Effective Bug Triage. It addresses the issue of data reduction for bug triage by textbook bracket ways. Conventional software analysis isn't completely suitable for the large- scale and complex data in software depositories. Data mining has developed as a promising means to handle software data. There are two difficulties related to bug data that may impact the effective use of bug depositories in software development tasks, videlicet the huge scale and the low quality. the impact of climate change in India, maturity of the agrarian crops are being poorly affected in terms of their performance over a period of last two decades. Predicting the crop yield well ahead of its crop would help the policy makers and growers for taking applicable measures for marketing and storehouse. similar prognostications will also help the associated diligence for planning the logistics of their business. Several styles of prognosticating and modeling crop yields have been developed in the history with varying rate of success, as these do not take into account characteristics of the rainfall, a n d are substantially empirical. In the present study a software tool named Crop Advisor' has been developed as a stoner friendly web runner for prognosticating the influence of climatic parameters. This software provides an suggestion of relative influence of different climatic parameters on the crop yield, otheragro-input parameters responsible for crop yield aren't considered in this tool, since, operation of these input parameters varies with individual fields in space and time.

[3]. Automatic Bug Triage using Semi Supervised Text Classification Automatic Bug Triage using Semi-Supervised Text

Bracket proposes asemi-supervised textbook bracket approach for bug triage to avoid the insufficiency of labeled bug reports in being supervised approaches. This approach combines naive kudos classifier and anticipation maximization to take advantage of both labeled and unlabeled bug reports. This approach trains a classifier with a bit of labeled bug reports. also the approach iteratively labels multitudinous unlabeled bug reports and trains a new classifier with markers of all the bug reports. also it employs a weighted recommendation list to boost the performance by assessing the weights of multiple inventors in training the classifier. Before training a supervised classifier for bug triage, a necessary step is to collect multitudinous labeled bug reports, which are bug reports marked with their applicable inventors. The semi supervised textbook bracket approach to ameliorate the bracket delicacy of bug triage. This semi supervised approach enhances a NB classifier by applying anticipation-maximization(EM) grounded on the combination of unlabeled and labeled bug reports. First, this semi-supervised approach trains a classifier with labeled bug reports. also, the approach iteratively labels the unlabeled bug reports and trains a new classifier with markers of all the bug reports.

[4]. Reducing Features to Improve Bug Prediction:

Lately, machine literacy classifiers have surfaced as a way to prognosticate the actuality of a bug in a change made to a source law train. The classifier is first trained on software history data, and also used to prognosticate bugs. Two downsides of being classifier- grounded bug vaticination are potentially inadequate delicacy for practical use, and use of a large number of features. These large figures of features negatively impact scalability and delicacy of the approach. Reducing Features to Ameliorate Bug vaticination aims in classifier to first trained on software history data, and also used to prognosticate bugs. The disadvantage of the traditional system is that classifier- grounded bug prognostications are potentially inadequate delicacy for

practical use, and use of a large number of features. The system uses Naive Bayes and Support Vector Machine(SVM). The system substantially Gain rate for point selection, along with the characterization of bug vaticination results achieved when using point selection. This paper proposes a point selection fashion applicable to bracket- grounded bug vaticination.

Sl. no	Title	Year	Tools Application	Area	Pros	Cons	Analysis
1	Towards Effective Troubleshooting with Data Truncation	**2020**	Instance selection and feature selection	Bug Repository Handling.	The problem of handling huge amount of data in bug repository is minimized.	Instance selection and feature selection is not completely enough to handle the data in bug repository.	Additionally, text processing and data processing algorithms must be used to handle the data in bug repository.
2	A Technique to Combine Feature Selection with Instance Selection for Effective Bug Triage	2016	Instance selection and feature selection	For efficient bug triage.	Analysis data by considering the word dimension and bug dimension which helps in reducing duplicate and unnecessary bugs.	The order of applying instance selection and feature selection is not clearly explained which leads to inefficient system.	The order of applying instance selection and feature selection must be determined by a predictive algorithm.
3	Automatic Bug Triage using SemiSupervised Text Classification	2010	Naïve bayes classifier	Software bug handling.	It labels the bug data iteratively. The weighted list maintained helps to boost the results obtained	It only focuses on classifying the bugs in bug repository. The major problem in bug handling is that huge amounts of data are in bug repository.	Techniques such as IS and FS must be used to reduce the data in bug repository. A proper predictive algorithm must be used to classify bugs.

4	Reducing Features to Improve Bug Prediction	2009	Support vector machine	Software development committee.	Provides a layout for minimizing techniques used in bug data reduction.	Minimizing the techniques used leads to less accuracy.	Necessary algorithms such as IS and FS must be used.

Table -1: Analysis of various bug tracking systems

V SYSTEM ARCHITECTURE

System architecture is the structural design of systems. Systems are a class of software that provide foundational services and automation.

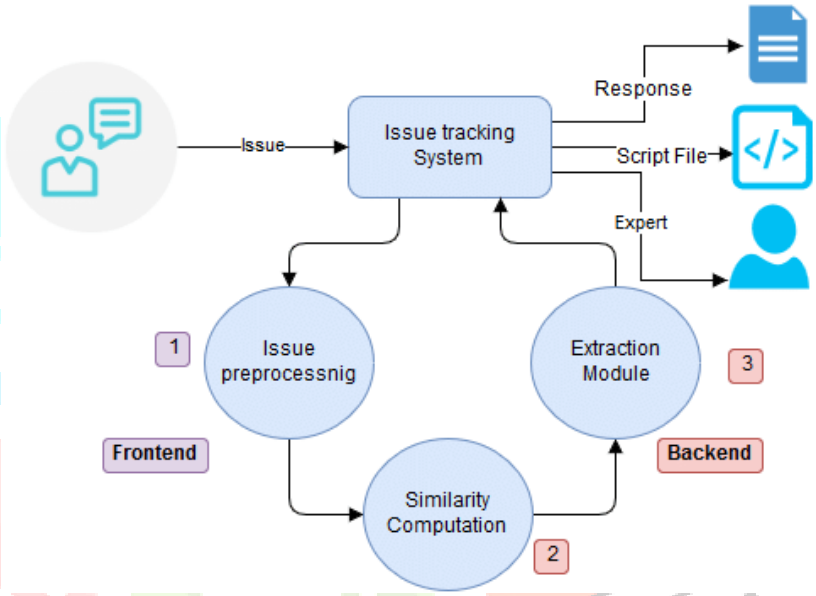

Figure 2: System Architecture.

VI MODULES

6.1 Admin Module:

This module has the entire access to all other modules. Admin can add members to the managers.

6.2 Manager Module:

Manager has the full access to the particular project assigned by the admin and control the team member's access and adding the project.

6.3 Developer Module:

Preprocessing Developer can access the task or Defect assigned by the manager, view assigned project and resolving the assigned by the Defect. Testers can access the projects and Defect to the list and send the bug manager.

6.4 Tester Module:

Testers can login to the system and access the assigned projects list.

6.5 Report:

Both Admin and Managers can access this module and generate reports based on the requirements.

VII SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated [3]. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive [1]. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields [18]. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components [2].

7.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

VIII RESULTS

We answer the aforementioned question affirmatively by presenting two secure and efficient cloud-based dual access control systems1 in different contexts. With the aim of providing an efficient way of dual access control, we briefly introduce the technical roadmap as follows. To guarantee the confidentiality of outsourced data without loss of policy- based access control, we start with a CP-ABE system [36], which is seen as one of the building blocks. We further employ effective control over data users' download requests on the top of the CP-ABE system. We design a new approach to avoid using the technique of "testing" cipher text. Specifically, we allow data users to generate a download request. Upon receiving the download request, with help of the authority

or the enclave of Intel SGX, a cloud server is able to check if the data user is authorized to gain access to the data. No other information is revealed to the cloud server except the knowledge of whether the user is authorized. Based on the above mechanism, the cloud maintains control of the download request. The systems we propose have the following distinct features:

(8.1) Confidentiality of outsourced data.

In our proposed systems, the outsourced data is encrypted prior to being uploaded to the cloud. No one can access them without valid access rights.

(8.2) Anonymity of data sharing.

Given an outsourced da- ta, cloud server cannot identify data owner, so that the anonymity of owner can be guaranteed in data storage and sharing.

(8.3) Fine-grained access control over outsourced (encrypt- ed) data.

Data owner keeps controlling his encrypted data via access policy after uploading the data to cloud. In particular, a data owner can encrypt his outsourced data under a specified access policy such that only a group of authorized data users, matching the access policy, can access the data.

(8.4) Control over anonymous download request and EDoS attacks resistance.

A cloud server is able to control the download request issued by any system user, where the download request can set to be anonymous. With the control over download request, we state that our systems are resistant to EDoS attacks.

IX CONCLUSION:

The Defect Tracking System is very useful for removing defects from project modules. Only if features mentioned in document are extended. In future users may also be possibly notifying the name/Id of defect creator so that organization can prevent their systems. The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

i. It's a web-enabled scheme.

ii. This task offers the user the opportunity to enter the data through simple and interactive forms. This is very useful for the client to enter the desired information through so much simplicity.

iii. The user is mainly more worried about the validity of the data, whatever he is entering. There are checkson every stage of any new creation, data entry or update so that the user. cannot enter the invalid data, which can build problems at a later date.

iv. Sometimes the user finds bugs in the later stages of using a Project that he needs to update some of the Information that he entered earlier. There are options for him by which he can update the records.

X REFERENCES:

[1]. BasicInspectionprocess,http://www.cs.toronto.edu/~sme/CSC444F/handouts/inspection_process_model.pdf.

[2]. Gao, Kehan, et al. "Choosing software metrics for defect prediction: an investigation on feature selection techniques." Software: Practice and Experience 41.5 (2011): 579-606.

[3].Introduction,http://www.mks.com/solutions/disc ipline/dm/defectmanage ment?gclid

[4]. Jalote, Pankaj, and Naresh Agrawal. "Using defect analysis feedback for improving quality and productivity in iterative software development." Information and Communications Technology, 2005. Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on. IEEE, 2005.

[5]. Kalinowski, Marcos, David N. Card, and Guilherme H. Travassos. "Evidence-based guidelines to defect causal analysis." Software, IEEE 29.4 (2012): 16-18.

[6]. Kocher, Paul, et al. "Introduction to differential power analysis." Journal of Cryptographic Engineering 1 (2011): 5-27.

[7]. Kumaresh, Sakthi, and R. Baskaran. "Defect analysis and prevention for software process quality improvement". International Journal of Computer Applications (0975–8887) Volume (2010).

[8]. Lauesen, Soren, and Otto Vinter. "Preventing requirement defects: An experiment in process improvement" Requirements Engineering 6.1 (2001): 37-50.

[9]. Life Cycle, http://www.softwaretestingstuff.com/2008/05/buglife-cycle.html.

[10]. Maintenance and Environment, Conclusion http://www.onestoptesting.com/test-cases/defecttracking.asp

[11]. R.B. Lenin & R.B. Govindan," Predicting Bugs in Distributed Large Scale Software Systems Development", 2008.

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

[13].Trajkov Marko & Smiljkovic Aleksandar, "A Survey of Bug Tracking Tools", 2006.

[14] D. Matter , A. Kuhan , and O. Nierstrasz , "Assigning bug reports using a vocabulary-based expertise model of developers," in 6th IEEE International Working Conference onMining Software Repositories, 2009, pp. 131–140.