

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

Energy Efficiency for Industrial IOT with Software Correlation

Madanachitran.R¹, G. Dharani², P.Lakshmi Prathipa², M.Reena²

Assistant Professor, Department of IT, Vivekanandha College of Technology for Women, Namakkal, India

B. Tech (IT), Vivekanandha College of Technology for Women Namakkal, India

ABSTRACT : Large Software projects have a huge number of dependencies among its elements such as functions, objects, modules. Coordination between software development teams is one of the most difficult aspects in software engineering. Coordination has been defined as the number of dependencies that directly impacts the level of complexity and the software Change is unavoidable in certain development of the project. So we introduce the notion of Socio-technical congruence as a framework for examining fine grain technical dependences among software entities and their implication for coordination needs in development organization. In future, we can identify the coordination requirements and actual coordination behaviors in software application. For Improving Software Reliability, Defect Tracking System (DTS) gives the facility to define the tasks and allow the managers to track the Defects and time spent by each employee for that particular task. This tool can help managers for Defects (Bugs) estimation per project. This tool also helps employees to document their Defects and analyze the quality of their output. Moreover the project aims at creation of a Defect Tracking System which will be accessible to all developers and its facility allows to focusing on creating the database schema and while letting the application server define table based on the fields in JSP and relationships between them. The objectives of this system are to keep track of employee skills and based on the skills, assignment of the task is done to an employee. Employee does Defects capturing. It can be done on daily basis.

I. INTRODUCTION

A bug tracking system or defect tracking system is a software application that keeps track of reported software Each bug in the system may have an urgency value assigned to it, based on the overall importance and the future occurrence of that bug. Low or zero urgency bug are minor and should be resolved as time permits. Other details of bugs include the customer experiencing the rise of bug, date of submission, detailed descriptions of the problem being experienced, attempted solutions and other relevant information. As work is done on that bug, the system is updated 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.

In the previous Defect tracking system, user does not specify all the information of the bug needed by developers. a bug-tracking system may be used to generate reports on the productivity of programmers at fixing bugs. However, this may sometimes yield inaccurate results because different bugs may have different levels of severity and complexity. The severity of a bug may not be directly related to the complexity of fixing the bug. There may be different opinions among the managers and architects When a software engineer develops a bug report, most probably, he should have asked the following questions. Some of them are what is the name of the product? What is the type of bug? In which component is the bug? In which modules the bug? In which part of the program the bug is? What type environment in which the bug is arising? In which platform the application is built? In which Operating System the application is running? The report given by developer about bugs might be incomplete initially. When a bug report is submitted by a developer, it is the duty of the software tester to the follow up questions are to be asked about the software bugs which gives a tester complete of the issue he is facing in a particular project. Our bug tracking system provides an "automated questionnaire system" which collect all the information on behalf of the tester. More over answer to a question

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

determines the next possible question. These question and answers helps a tester in narrowing down exact location of the bug in the code.

First and fore most thing we need is the exact location of the bug in the program. Location gives you the information like the line number of the bug in the program its method, class and so on. This helps a developer to get to the location of erroneous code. This information can be obtained by using software development environments (IDEs). 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 behaviour, 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 Bug tracking system will handle structured data including user information, login information, data code information and report. which are saved in a database. Our project provides these facilities using JSP and HTML languages.HTML provides a user interface for login, sing up etc. Whereas JSP is used to established a connection between the HTML and server. In this we have used MYSQL database to store the information of the users. Our intuitive interface includes a Dashboard view showing you issues 'Assigned to Me', 'Unassigned', 'Reported by Me' and many other standard filters as your go-to page to get started for the day. Defect tracking is the process of finding defects in a product and can come in the form of discrete systems where details of the defect and the status of work associated with resolving it are stored or they may be part of an integrated suite of configuration management tools, where the status of the defect may act as a trigger or key for other events within the system.

II. BACKGROUND WORK

In this document we are providing detailed description of current system and proposed system. All functional and non-functional requirements are mentioned in chapter 3. Detailed UML diagrams like use cases, class diagrams, sequence diagrams have been created with all the details of the proposed system architecture [6]. Explanation of each class, components, modules, sub-systems is provided with all the content required to understand the design framework, and development process [11][16]

The purpose of Defect Tracking for improving software reliability is to provide better service to the administrator or useful for applications developed in an organization [1][10]. The “Defect Tracking for Improving Software Reliability” is a web based application that can be accessed throughout the organization [2][16]. This system can be used for logging Defects or Bugs against an application/module, assigning them to team members and tracking them for resolution [13][5][8]. There are features like email notifications, user maintenance, user access control, report generators etc. in this system [12].

At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product. The existing prototype is evaluated in the same manner as the previous prototype and if necessary, another prototype is developed from it according to the fourfold procedure outlined above. The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

The current system is mostly concerned with the storing defects onto the file system with little to no traceability, making the tracking of the defects difficult at a later time. Most of the work of inserting the defects and tracking them back at some later point of time requires human intervention and is done manually [10][15]. This makes the system limited and hence results in degraded performance [12]. Information retrieval is a very big process and it becomes very difficult to handle huge databases manually with same efficiency and at the same time with the increase in the database the time to retrieve the concerned information also increases manifolds. Lack of organization of the files makes it prone to information loss due to accidental deletion of files. No security because the files are visible to the users. More over every user has the same level of access to the files. Report generation is a big task and precision is as much

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

important as output is. Most of the work is done by humans with minimum to no intervention by machines. Humans are subjected to other factors like stress, emotions etc. that may reduce their work efficiency which is not the case with the machines, hence prolonged and maintained efficiency.

III. METHODS

A Bug Tracking System defines a streamlined process for bug reporting, bug assignment, status, severity, duplicity and priority. Bug tracking is a system which is used to keep a track of any type of bugs in any software. It helps the software developers to know what the error is so that they can resolve it and learn from it. This can be easily done by using various Bug Tracking Systems. Working on a software project includes the management of bugs we find. At first, we might list them on a spreadsheet. But as the number of bugs increases and a lot of people must access and input data on them, the use of spreadsheet becomes infeasible and instead use a bug- or an issue-tracking system. In this research paper we compare six leading Bug Tracking Systems i.e. Bug Genie, Bug Zilla, Bug Tracker, Mantis, I Tracker & Web Issues. This provides an insight to select an appropriate choice to enhance software development process and be more productive. We have distinguished the mentioned systems on the basis of platform and on the basis of functionality. We are proposing a framework for the Bug Tracking System which includes two additional features i.e. Duplicate Bug Detection and Automatic Bug Assignment. These features are not present in any of the existing six Bug Tracking Systems as mentioned in the literature, hence eventually removed the drawbacks of these systems.

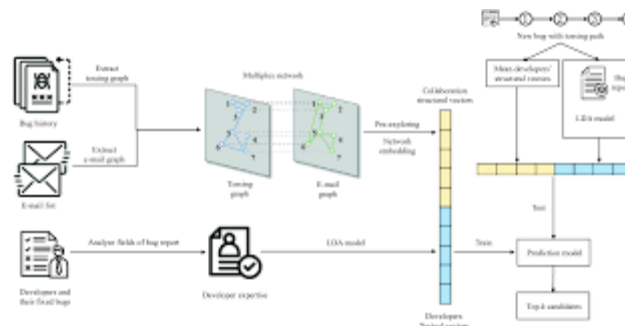


Fig 1: Bug Track report

In order to reduce the fixing time for submitted bugs and improve the probability of bug fixing, it is necessary to recommend appropriate developers for bug fixing. In this paper, we present an effective way of automatically recommending developers for bug fixing. This approach comprises three parts. The goal of the first part is to build the concept profile (CP). For achieving this purpose, we extract the bug concepts with topic terms from the documents deriving the corresponding bug reports. After the concept profile is built, when a new bug comes, we decide which bug concept the new bug belongs to, and then extract the corresponding developers from the concept profile. For the second part, we utilize social network (SN) [5] to find the important developers with the high probability of fixing this bug. The final part is to rank the candidates according to the developers' experience and fixing cost. We expect the proposed method can avoid the excessive number of re-assignments, improve the probability of bug fixing and reduce the bug fixing time. The remainder of this paper is organized as follows. Section 2 presents some related work. We describe the details of our developer recommendation approach in Section 3. Section 4 shows the evaluation results on the collected bug reports from the famous open source project JBoss, and we analyze the experimental results. We summarize our work and introduce future work in Section 5.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

IV. RESULT ANALYSIS

We investigate further ft-md search by restricting the number of obtained bugs; Nft and Nmd are both set to 100, 80 and 60. The right plot in Figure 4 indicates that the matching rate reduces when the number of obtained bugs reduces. Note that bugs are chosen by their ranking. The ranking of obtained bugs are different between ft search and md search. This is caused by three reasons: first, while BTSs may contain duplicated bugs, the number of these bugs is small, thus the number of truly similar bugs is also small; second, the ranking values of obtained bugs contain errors from indexing and ranking bugs, especially when a query is general, ranking many similar bugs is affected by these errors; last, as described above, ineffective queries make results inaccurate. These reasons also cause the low matching rate. However, since the bug dataset is wide and diverse in scope, we believe that ft-md search achieving an average matching rate of 0.3 is reasonable.

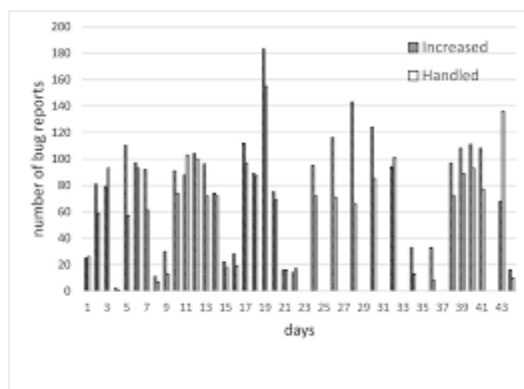


Fig 2: Bug Result Analysis

V. CONCLUSIONS

In this paper we have reviewed on the technologies which are being used for finding and improving bug tracking system. It is impossible to not encountering bugs during software development life cycle. Our bug tracking system provides an "automated questionnaire system" which collect all the information on behalf of the tester. A. Administrator or Project manager can get project status as well as management can get the ongoing status of the project. Present system can further be improved by applying certain standards. Our intuitive interface includes a (Dashboard view) showing you issues and (Timeline feature) allows you to keep across what's happening around you with a live stream on the latest actions by your team mates B. Customer resource management is integrated with this application which makes this software more efficient in terms of time management. Present system can further be improved by applying certain standards.

REFERENCES

- [1] A Practitioners approach "Software Engineering" Roger S. Pressman 5thEdition.
- [2] Lessons Learned in Software Testing (Paperback) Publisher: Wiley; 1st edition (December 15, 2001) James Bach
- [3] Bret Pettichord,C. Kaner.
- [4] Beautiful Testing: Leading Professionals Reveal How They Improve Software (Theory in Practice) (Paperback) Publisher: O'Reilly Media; 1 edition (October 22, 2009) Tim RileyAdam Goucher.
- [6] Why Programs Fail, Second Edition: A Guide to Systematic Debugging (Paperback) Publisher: Morgan Kaufmann; 2 edition (June 26, 2009).
- [7] Software Engineering,"Rojer S. Pressman" (TATA McGraw Hill Publication).
- [8] http://en.wikipedia.org/wiki/Bug_tracking_system
- [9] <http://www.wisegeek.com/what-is-bug-tracking.htm> (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [10] (Avram, 2007) Gabriela Avram, Anne Sheehan and Daniel K. Sullivan, Defect Tracking
- [11] Systems in Global Software Development – a work practice study, Limerick, Ireland, 2007.



ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

- [12] (Basili, 1984) Victor R. Basili, and Barry T. Perricone, Software Errors and Complexity: An
- [13] Empirical Investigation, Communications of the ACM, P.42-52, 1984.
- [14] (Black, 1999) Rex Black, Managing the Testing Process, Wiley, 1999.
- [15] (Blaxter, 2006) Loraine Blaxter, Christina Hughes and Malcolm Tight , How to Research THIRD EDITION ,2006.
- [16] (Boehm, 2001) Barry Boehm and Vector R. Basili, Software Defect Reduction Top 10 List, p.135–137, 2001.