A CASE STUDY OF IMPROVING USER EXPERIENCE OF BUG TRACKING APPLICATIONS

[1]Ashi Iram, [2]Saqib Saeed
[1]Department of Computer & Software Engineering
[2]Department of Computer Science, Bahria University Islamabad Campus
engr.ashi@yahoo.com, saqib.saeed@bahria.edu.pk

Abstract: *Human centered computing advocates for user centric technology design process for better technology acceptance. However, technology appropriation is a challenging task due to enormous user diversity. In this paper we present a case study where we tried to appropriate bug tracking application for user needs. Bug tracking applications are widely used by quality assurance teams to track bugs while software development. We carried out detailed interviews to understand the problems faced by quality professionals in Pakistan while using current bug tracking applications. Keeping in view these requirements, we designed a prototype and rolled out in practice to evaluate it. Our evaluation highlighted that some usability issues were improved.*

Keywords: Human Centered Computing, Technology Appropriation, Software Development, Usability, Bug Tracking Applications

1. INTRODUCTION

Software development is a complex and dynamic activity [cf. 14, 15]. Quality assurance approaches try to control defects in software development process [1]. Bug detection and control in earlier phases of software development can significantly reduce the cost [2]. . In case of large projects, bug tracking can be quite complex [5]. Bug tracking applications help quality assurance professionals to manage bugs in an effective and efficient manner in such projects. In order to effectively use bug tracking applications, it is important that they are designed keeping in view users' needs [3, 4].

There are many open source and commercial bug tracking systems such as Bugzilla [7], ITracker [8] etc. In Free and Open-Source Systems (FOSS) users had to face a lot of problems in bug triaging, so a framework was proposed [5] that make recommendations on who can fix bug, give suggestions of developers, their workload and schedule while allocating problem reports to team members. It is found that textual data is more useful for bug assignment and automation on textual data largely reduce human efforts [9]. Different challenges were also faced by Mozilla developers, it was very difficult to maintain overall understanding of bugs in which they work, developers wished for user centricity, customization, action view list and some visualization techniques that give them visual representation on bugs updates [10]. In addition to incomplete information on bug reports, it cause frustration and consume a lot of time and resources, therefore quality of information and completeness in bug reports is necessary [11]. Another survey was conducted on quality of bug reports in which Mozilla, Apache, and Eclipse projects were under study, developers reported bug reporting sites are very insensitive to language, if bugs are not filed in English language they are immediately closed, some other developers complained about search feature of Bugzilla is very limited [12]. JIRA is another open-source bug tracking system [13]. JIRA is an integrated repository, where the resource purchases, in depth information about it and handling of purchases is kept.

In our previous work [12], we found that different open source and proprietary bug tracking applications are used, but users are facing different usability issues. In this paper we have extended this work by carrying out a qualitative study to extract detailed requirements. Based on these requirements we designed a prototype and get it evaluated from users.

Rest of paper is structured as follows. Section 2 discusses materials and methodology used and section 3 provides details of our research work along with empirical data collection and features provided in prototype. Finally Section 4 concludes our paper.

2. MATERIALS AND METHODS

We conducted interviews to acquire in depth requirement details. Our interviewees were mainly software developers, testers, quality assurance engineers and team leads. Each interview was of approximately 30 minute duration. We transcribed the interview contents and used grounded theory [16] for analysis.

Based on these requirements we designed a prototype. The front end of application was developed in Hypertext preprocessor (PHP) as language. We used Lib charts to create different graphs so that data is represented in graphical way. Later on we rolled out this system to some of our interviewees for evaluation and later we collected their feedback about this system.

3. RESEARCH WORK

In this section we discuss the data gathered during our qualitative interview that made the basis for bug tracking application. Furthermore we highlight application design and evaluation process.

3.1 Empirical Data

In our empirical data we found some functional and some interface problems. One of tester given his feedback on zephyr as follows:

"The attachments to the test are not shown up while we execute the test. Once we click to execute the test case, the page does not show any attachments we had with the test case like in this screenshot…."

Another tester given his views on zephyr during interview as follows:

"Once the test execution of complete for any plan, it will be nice to have a summary report run at the end for the number of test cases pass or fail, Since we are planning to move towards automation later as our product matures, I would like to see the direct execution of automated test from JIRA. Lastly, on the test execution screen, can we also have buttons to move to next/previous test case? Right now there are two buttons available, Return to test and Return to test cycle. This

will save a significant amount of time for the team as they dig through a cycle of 100+ or more regression test cases."

One of the interviewee commented as follows:

"As we are following agile methodology so we use JIRA bug tracking tool as a project management and bug tracking tool, we preferred JIRA because it provide flexibility, usability, visibility and efficiency. Also JIRA requires minimum 2 days basic training time period, and needs 1 week for complete understanding. Structure is fully defined and designed by Quality Assurance manager in QA plan. One of major issue we sometime face in JIRA is a lot of extra and irrelevant information on bugs and projects, a lot of comments, posts , status and bug logs so there should be only relevant information on screen. I think for non-technical users in project management point of view JIRA user interface is not user friendly enough. "

Another respondent given his feedback about JIRA in following words:

"Our technical resources suffers a lot our main goal is on delivering products on time efficiently, as our clients want fast delivery so we have to compromise on Quality our goal is: who can deliver on time not who can deliver best. But besides all JIRA is costly, it consumes our a lot of resources, we need dedicate machines, JIRA its self didn't provides visibility we have to figure it out. "

Another respondent commented in following words:

"During test case execution it would be nice to have a green check mark for pass and a red one for fail side by side with the drop down under 'Status' field for more details. This will be very helpful for us to just click pass/fail instead of opening up the dropdown for every step."

Another participant commented on communication approach in JIRA:

"JIRA notifies via emails to all members but it is sometimes very irritating for someone who is not involved in this process emails should be only sent to relevant and specific users. Furthermore a time comes when a bug cannot be further reproduced.... So I suggest there should be a mechanism that should give real time feedback and suggestions on bug's reproduction. I wish I can view all projects working progress and status of bugs in one glance but I have to figure out myself in this tool that is really weird for me and if there was a simple drag and drop mechanism for bug assignments that will be easier for me."

Implications for Design

After detail analysis of empirical data, we decided to implement following features for Bug Tracking system focusing on easily and efficiently bug assignment**.**

3.2.1 Dash Board for Project manager and User

Dash board feature is basic and fundamental feature for all Bug Tracking Systems. In dash board project manager can view total number of bugs and projects. This feature can help both project manager and users to view and keep track of assigned bugs.

3.2.2 Priority and status change for Bugs

Setting and analyzing priority of projects and bugs is important so that highest priority bug is resolved first. Color coding scheme is a one of best technique for visualization so we planned to implement this where users can adjust priority settings through drag n drop.

3.2.3 Statistical view of progress

Providing feature of statistical views like pie charts and bar graphs can be very helpful so that project manager and other users can easily view progress of individual projects.

3.2.4 Tool Tips feature

Tool tips feature by mouse hover on project title will allow users to see details of a single project, so that users can get idea about what is project related.

3.2.5 Authentication Rights

It is better to have access control feature in all systems. Here in our Bug tracker all access and authentication rights is given to Project manager who can add new members, create new projects and new bugs. And view overall progress of bugs and projects. Whereas a member can only view his/her assigned bugs via dashboard.

3.2.7 View user own progress

Users can view his own progress of assigned bugs and projects.

3.2.8 Interaction facility for users

Users can interact with project manager via commenting on assigned bug if they had to discuss or tell anything, they can interact through commenting.

3.2 SYSTEM DESIGN

User is one of main entity in our database because it will handle all team members' personal information, user's login passwords etc. Project is another main important entity that keeps information of different projects names, categories, project's description, and project's status.

Similarly Bug entity has attributes like types, priority, category, status of bugs, bug's description etc. Comments entity contain attributes like bug id, project id, details, user id etc. It will show which user has commented on a particular bug. Finally Attachments entity has attributes like name of file to be attached, file extension format etc. Figure 1 represents the system overview and Figure 2 highlights the entity relationship diagram.

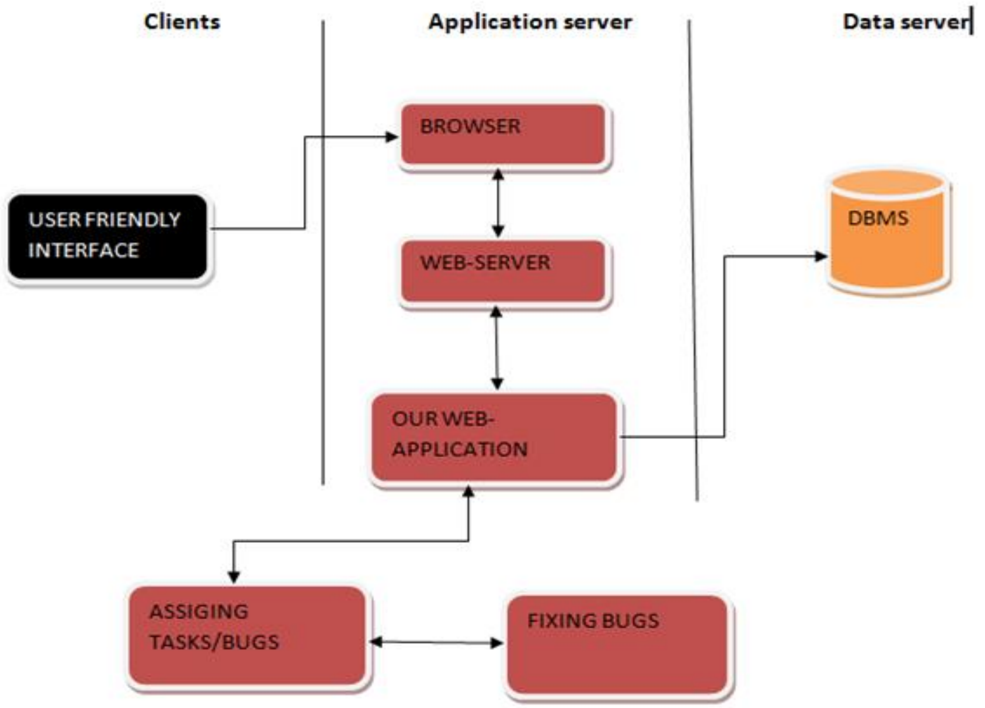

Figure 1 System Overview of BTS prototype

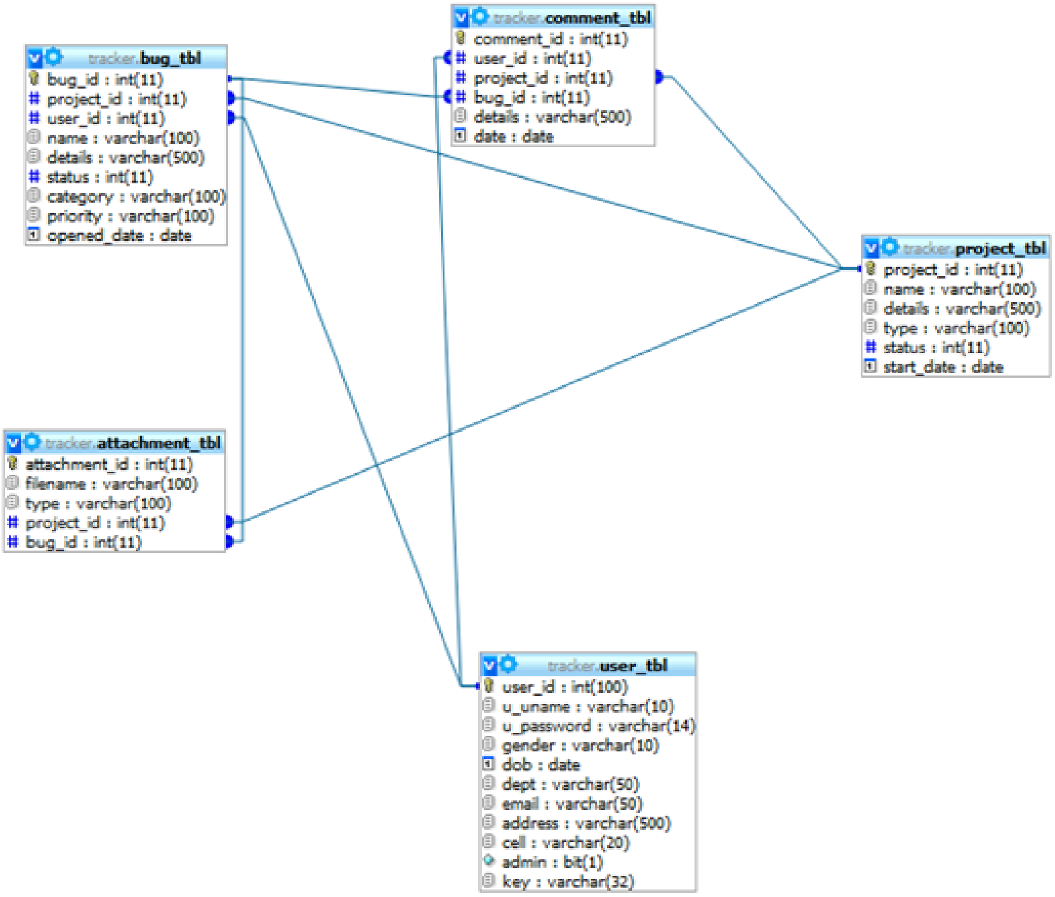

Figure 2 Entity Relationship diagram

3.3 SYSTEM EVALUATION

After development we selected some users from local software industry for the evaluation of our system. We found some positive feedback on usability aspects. One of respondent commented in following words:

"It provides me facility to handle multiple projects at same time; iconic representation is well logically arranged. Interactive thing I found is drag and drop bug assignments to developers.

Another respondent provided his feedback in following words:

"This BTS prototype provides me facility to view and find status of bugs easily; color scheme improved its visualization. Moreover I can easily change priorities of bugs by drag and drop. Graphical presentation is good concept because from this we can view bugs impact on our projects."

Similarly another participant said:

"Dashboard is pleasantly arranged and well categorized. I can quickly view all projects, bugs, and their status. Team icon on the top of menu is good. I can view all team members, add new members to my team this helps me in my team management task. "

Another respondent commented in following words:

I found view bugs and view projects very good feature as i can click on view bugs and it give me all detail description of bug I can download attached file, attachments are clearly visible. Moreover I can easily communicate with my project manager through comments without any meeting or appointments.

4. CONCLUSION

Bug tracking systems play significant role in software development. Our previous qualitative study revealed that users of different bug tracking system are facing usability issues while using open source and commercial web based systems. Keeping this in view we have gathered some of user's requirements and then implemented a prototype that can help bug triager's. Our prototype provides features to assign bugs via drag and drop, view progress of all projects, bugs, and team members, providing communication facility for rapid interaction simple and easy interface. Our prototype received positive feedback after evaluation.

Further enhancements can be done in prototype by providing efficient bug search and filtering also can be customized according to more user needs and requirements. It can also be integrated with software configuration systems that can make testing and deployment phase easy.

REFERENCES

1. Zhongpeng Lin, Fengdi,Shu , YeYang ,Chenyong Hu ,Qing Wang "An empirical study on bug assignment automation using Chinese bug data", *3rd International Symposium on Empirical Software Engineering and Measurement, ESEM* pp. 451- 455(2009).
2. Khanjani, A. and Sulaiman, Riza, "The process of quality assurance under open source software development", *Computers & Informatics (ISCI), IEEE Symposium on* . pp. 548- 552,(2011)
3. Ruby, R.J., Browning, L.A. and Roberts, A.R. "Cost effectiveness of software quality assurance". *Aerospace and Electronics Conference, 1989. NAECON 1989., Proceedings of the IEEE 1989 National* . Vol. **4**, pp. 1614- 1620, (1989).
4. "IEEE Standard for Software Quality Assurance Plans." *IEEE. 1989.* IEEE Std 730.1-(1989) .
5. Baysal, O., Godfrey, M.W. and Cohen, R**.**"A bug you like: A framework for automated assignment of bugs", *IEEE 17th International Conference on Program Comprehension, ICPC '09.*. pp. 297- 298 .(2009)
6. Jingyue Li , Stålhane T.Conradi R. ,Kristiansen, J.M.W. "Enhancing Defect Tracking Systems to Facilitate Software Quality Improvement", *Software, IEEE* , Vol. **29**, pp. 59-66 (2012).
7. Ciordia, Nicolas Serrano & Ismael**,** "Bugzilla , ITracker and other Bug Trackers." [ed.] christof Ebert. march-april (2005).
8. Bugzilla. [Online] [Cited: 12 saturday, 2013.] http://www.bugzilla.org/docs.
9. Baysal, O., Holmes, R. and Godfrey, M.W,"*Situational awareness: Personalizing issue tracking system". 35th International Conference on Software Engineering (ICSE),* . pp. 1185- 1188 (2013)
10. Just, S., Premraj, R. and Zimmermann, T. "Towards the next generation of bug tracking systems",*IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* pp. 82-85(2008).
11. Sarkan, H.M., Ahmad, T.P.S. and Bakar, A.A. "Using JIRA and Redmine in requirement development for agile methodology", *5th Malaysian Conference in Software Engineering (MySEC),* pp. 408-413(2011).
12. Iram, A. & Saeed, S. "Exploring Usability Issues in Bug Tracking Applications: An Empirical Study of Pakistani software Industry", *Sindh University Research Journal* (Submitted), (2014)
13. Zimmermann, T. ; Premraj, R. ; Sillito, J. ; Breu, S."Improving bug tracking systems" *31st InternationalConference on Software Engineering - Companion Volume, ICSE-Companion,*. pp. 247 - 250(2009).
14. Saeed, S., Khawaja, F.M. and Mahmood, Z. "A Review of Software Quality Methodologies. In: Alsmadi, I., Ed., Advanced Automated Software Testing: Frameworks for Refined Practice," *Information Science Reference, Hershey,* 129-150, (2012)
15. Alsmadi, I. and Saeed, S. "A Software Development Process for Open Source and Open Competition Projects", *International Journal of Business Information Systems,* 12, 110-122, (2013)
16. Strauss, A., & Corbin, J. M. (Eds.). *Grounded theory in practice*. Sage, (1997).