Sindh Univ. Res. Jour. (Sci. Ser.) Vol.46 (3) 267-272 (2014)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Exploring Usability Issues in Bug Tracking Applications: An Empirical Study of Pakistani Software Industry

A. IRAM, S. SAEED[++]*

Department of Computer & Software Engineering, Bahria University Shangrila Road, Sector E-8, Islamabad

Received 06th June 2014 and Revised 8th August 2014

Abstract: In order to foster successful usage of software applications, usability is an important factor. Bug tracking applications are excessively used by quality assurance teams to manage bugs in software development. In order to support effectively these tools must be highly usable to enhance user experience. In this paper we have conducted a survey on bug tracking applications used by Small and Medium Enterprises (SMEs) of Pakistan. Our main focus was to understand usability issues in these applications. We used Nielson's heuristics for evaluation. Our results show that there are some usability problems in open source and commercial bug tracking applications that need to be resolved to better support quality assurance teams.

Keywords: Quality Assurance, Software Development, Usability, Bug Tracking, User experience.

1. INTRODUCTION

Usability is defined as ease with which different users can use system effectively and efficiently (cf. Nielsen, 1994). Bringing usability factor in systems is quite important aspect because end users are diversified in their skills. Software development is inherently a complex activity (cf. Saeed *et al.,* 2012; Saeed and Alsmadi, 2013). In software development bug tracking is an important activity to improve quality of software products (Li *et al.,* 2012).

Bug tracking system (BTS) is software application that is used by software developers throughout development process. There are many open source and commercial applications (Khanjani *et al.,* 2011). These applications help testing to be more effective and reduction of bugs in delivered software products (Rubey *et al.,* 1989, Baysal, *et al.,* 2009). Different empirical studies show that automating the bug tracking process can significantly decrease software evolution effort and costs.

Usability can significantly improve/deteriorate the usage of software applications. Keeping this in view we conducted a survey to understand and explore different usability issues faced by users of bug tracking systems.

Open source and commercial bug tracking systems are widely used by software industry e.g. Bugzilla, ITracker (cf. Serrano and Ciordia, 2005). Lin *et al.,* (2009) found that textual data is more useful for bug assignment and automation on textual data largely reduce human efforts. In another work Baysal *et al.,* (2013) explored different challenges faced by Mozilla developers. They concluded that better customization, action view list and more visual representation on bug updates can benefit them more. In another study Just *et al.,* (2008) compared quality of bug reports in Mozilla, Apache, and Eclipse. Zimmerman *et al,* (2009) described that interfacing of bug tracking system with database management system make reporting structure very complex.

Despite these contributions there is no specific study highlighting which bug tracking applications are used by quality engineers working in Pakistani software industry and which usability issues are faced by them. Keeping this in view we have carried out this empirical work to fill this research gap. Remaining paper is structured as follows: Section 1 discusses related work and section 2 highlights methodology used to answer the research questions. Section 3 discusses empirical findings followed by discussion in section 4 and conclusion in section 5.

2. MATERIALS AND METHOD

Our study was mainly a cross sectional study and we mainly collected quantitative data using a questionnaire. Our questionnaire was divided into three sections. First section was about user information whereas second section focused on bug tracking application and last section contained questions about Nielsen's 10 usability heuristics. A total of 120 questionnaires were floated in different small and medium software houses in different cities. We received 60 filled questionnaires back. Each question was rated on a likert scale between 0-9. (0-1) strongly disagree

[++]Corresponding Author: Email: saqib.saeed@gmail.com and engr.ashi@yahoo.com
*Department of Computer Science, Bahria University Shangrila Road, Sector E-8, Islamabad

(SD), (2-3) disagree (D), (4-5) uncertain (U), (6-7) agree (A), (8-9) strongly agree (SA).

3. RESULTS

In (**Fig.1)** vertical Axis of graphs is representing number of respondents and horizontal axis represents different attributes like gender of participants, designation of participants, cities, type of companies and computer experience of respondents.

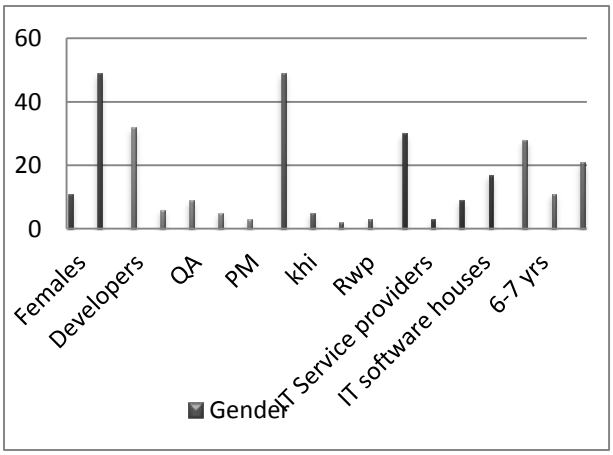

Fig.1 User and Company Information

Total 60 respondents participated in our survey in which 11 were female and 49 were male. Similarly 32 developers, 6 team leads, 9 from quality assurance team, 5 testers and 3 project managers given response of questionnaire. As we have floated questionnaire in different cities via internet and by hand so we received 49 responses back from Islamabad, 5 from Karachi, 3 from Rawalpindi and only 2 from Lahore. Among respondents 30 belonged to IT departments of different organizations, 3 from IT service providing organization, 9 from telecommunication industry and 17 respondents were from different software houses. Finally we gathered users' computer usage experience we found 28 users had 1-5 years' experience, 21 users had more than 10 years' experience and only 11 users had 6-7 years of total computer experience.

In **(Fig. 2)** highlights bug trackers used in Pakistani software industry, Bug tracking tool usage experience of participants, daily time spent on bug tracker (hours), and overall usability. 18 users started using bug tracking tools, 10 users had 6 month experience, 19 users had 1-2 years' experience, 10 users had 3-4 years' experience and only 3 users had more than 5 years' experience. We explored that 17 software houses use Bugzilla and 8 uses JIRA. Total 45 respondents daily spend 3-4 hours, 11 respondents spend 4-5 hours and only 4 respondents spend 6-7 hours daily time on bug tracking applications. Furthermore, 18 users found bug tracker application easy, 27 users termed it as medium and 16 users rated them difficult.

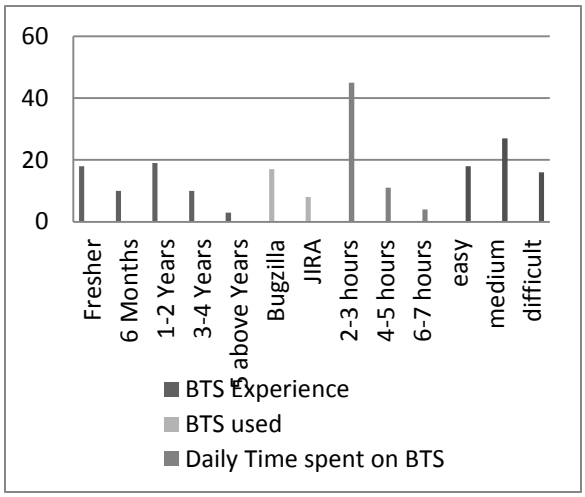

Fig. 2 BTS Information

In **(Fig. 3)** highlights results of Nielson's first heuristic that is about visibility of system status. System feedback is essential for keeping up to date and keeping aware of system's state. Q1 is showing graph about response of users in which they were asked usability question whether their Bug tracker provides necessary feedback 7 people strongly disagreed , 4 people were disagreed, 17 respondents were uncertain about this whereas 20 people agreed, 12 peopled given response as strongly agreed.

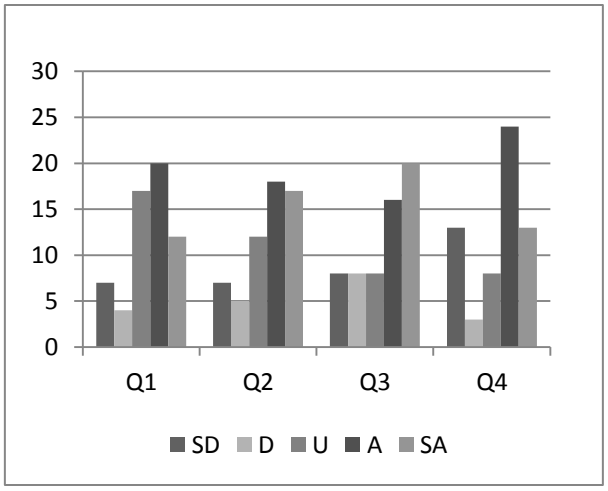

Fig. 3 Visibility of system status

Traditional Bug tracking systems send emails to relevant members on updating of bugs. But it becomes very difficult when there are lot of updates. In question 2 we asked related to system updates and

received responses as 7 strongly disagreed, 5 disagreed, 12 uncertain, 18 agreed and 17 strongly agreed to it. It is very important that user must see if he/she has done action task completion feedback must be given to user. As shown in Q3, 8 respondents strongly disagree, 8 disagreed, 8 were uncertain, 16 Agreed and 20 strongly agreed that they get good feedback. In fig. 3, Q4 shows responses against question where users were asked that whether all users can easily understand what is going on while using bug tracker. 13 respondents strongly disagreed, 3 disagreed, 8 were uncertain, whereas 24 respondents agreed and 13 strongly agreed. Match between system and real world, explores that whether iconic representation, menus and other metaphors used in system are related to real world or not. Language used by system must be understandable by users, avoid system oriented terms and abbreviations in massages dialog boxes.

As shown in (**Fig. 4)**, we asked three questions from respondents about this heuristic. Firstly we asked about language used in system is understandable for user and users responded as 13 strongly disagreed,6 disagreed, 7 were uncertain, whereas 15 were agreed and 18 strongly agreed, as shown in Q1 part of figure 4. Another question was related to understanding of iconic representation and menus 16 respondents were strongly disagreed, 8 disagreed, 13 were uncertain, whereas 11 agreed and another 11 strongly agreed that there BTS has good implementation of this feature, as shown in Q2 part of figure 4. Similarly last question was about error messages understandability, people responded as follows: 12 strongly disagreed, 11 disagreed, 7 were uncertain, 14 agreed and 15 strongly agreed.

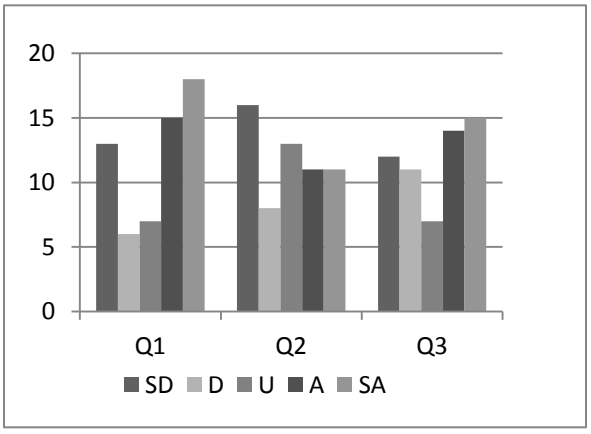

Fig. 4 Match between System & Real world

In Fig. 5 responses related to Nielsen's third heuristic user control and freedom are discussed. According to it if user unintentionally performed any action it should provide emergency exits or undo, redo tasks facility. In first question we asked whether your bug tracking system provides emergency exits. As shown in figure 5, Q2 graph, 11 respondents strongly disagreed, 13 disagreed, 9 were uncertain, 19 agreed and 7 strongly agreed. Another question was about availability of redo and undo features. As shown in figure 5, Q2 graph 18 respondents strongly disagreed, 9 disagreed, 10 were uncertain, whereas 10 agreed and 12 strongly agreed that their system provides such features.

Customization allows developers and all users of bug trackers to optimize system according to their choice. It provides flexibility and ease of use. In our survey we found that 12 respondents strongly disagreed, 10 disagreed, 11 were uncertain, 10 agreed and, 6 strongly agreed that their system provides this feature, as shown in Q3 graph in (**Fig. 5).**

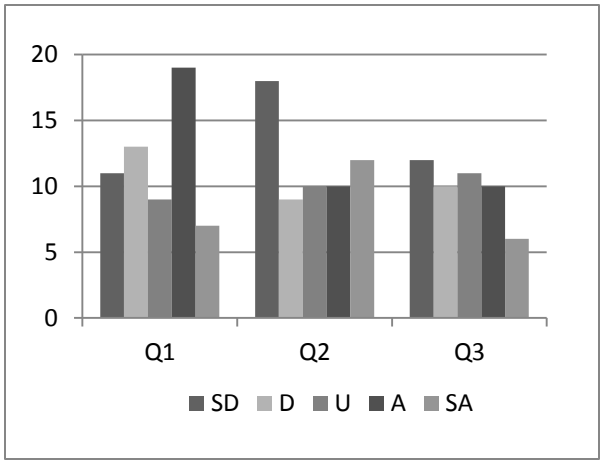

Fig. 5 User control & Freedom

Maintaining consistency and standard in interfaces helps users to understand and learn system quickly. As shown in (**Fig. 6)**, in our survey we asked six questions about this aspect. First question was about overall, interface consistency where 17 respondents strongly disagreed, 10 disagreed, 11 were uncertain, 13 Agreed and 9 strongly agreed that their bug tracking system has consistent interface. In next question where we asked about menus, iconic and dialogue box consistency. 13 respondents strongly disagreed, 10 disagreed, 16 remained uncertain, 14 agreed and 7 strongly agreed about the presence of this feature in their system, as shown in Q2 graph in figure 6. Similarly in case of messages consistency 19 respondents strongly disagreed, 4 disagreed, 12 remained uncertain, 16 agreed and 9 strongly agreed about its presence in their bug tracking system, as shown in Q3 graph in fig. 6.

Similarly in case of data field consistency 18 respondents strongly disagreed, 10 disagreed, 8 remained uncertain, 14 agreed and 10 strongly agreed

that their bug tracking system is consistent in this aspect, as shown in Q4 graph of (**Fig. 6)**. In case of iconic representation, 25 respondents strongly disagreed, 11 disagreed, 6 were uncertain, 8 agreed and 9 strongly agreed about its presence in their systems.

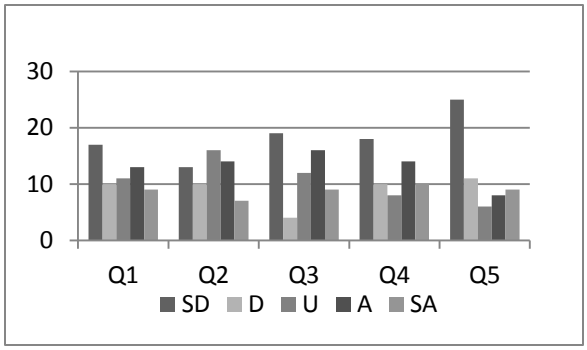

Fig. 6 Consistency & Standards

In (**Fig. 7)** shows result of Nielson's heuristic about error prevention. Good error massages prevent further mistakes and problems-First we asked that messages in bug tracking systems help them in recovering errors. 17 respondents strongly disagreed, 13 agreed, 6 remained uncertain, 16 respondents agreed and 9 strongly agreed about its presence, as shown in Q1graph in fig. 7.

In next question we asked whether their bug tracking system displays location of erroneous entry. In our survey, 13 respondents strongly disagreed, 8 disagreed, 15 uncertain, 9 agree, 14 strongly agreed about the presence of this feature, as shown in Q2 graph of figure 7. Finally we asked about system facility for recovering error prone tasks. In this aspect 18 respondents strongly disagreed, 9 disagreed, 11 remained uncertain, 10 were agreed and 9 strongly agreed, as shown in Q3 graph in (**Fig. 7).**

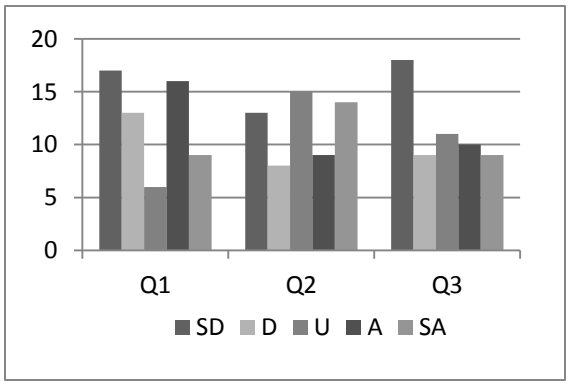

Fig. 7. Error Prevention

System must be easy to understand so that if user use it after long period of time he/she can easily use. More focus should be given to recognition rather than recalling and memorizing difficult commands, instructions. In first question we asked about icons and menus visibility, 24 respondents strongly disagreed, 9 disagreed, 6 were uncertain, 12 agreed and 9 strongly agreed that their system has good visibility as shown in Q1 graph in fig. 8**.** In second question we asked whether icons and menus are easily findable by users. 21 respondents strongly disagreed, 10 disagreed, 10 remained uncertain, 12 agreed and 7 strongly agreed about its presence in their system, as shown in Q2 graph in (**Fig. 8).**

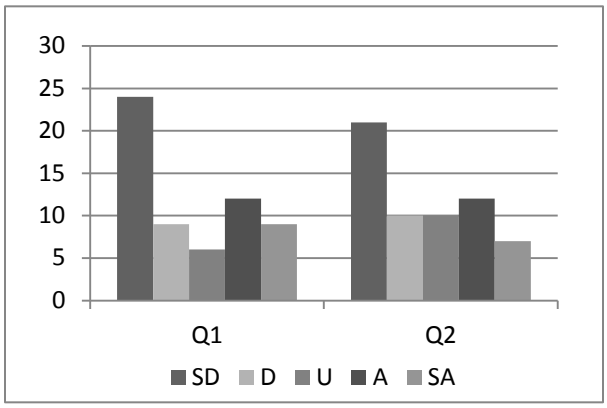

Fig. 8. Recognition Rather than Recall

System should provide shortcuts and accelerators. In our survey we asked about system shortcut keys. 21 respondents strongly disagreed, 10 disagreed, 13 remained uncertain, 11 agreed and 6 strongly agree about its presence in their bug tracking application, as shown in Q1 graph of (**Fig 9).**

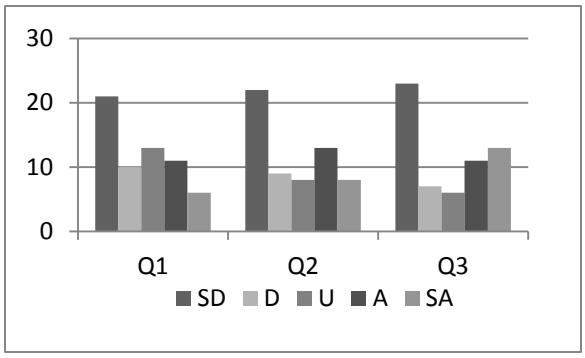

Fig. 9 Flexibility and Efficiency in use

Then we asked whether assignment of bugs is easy in their application. 22 respondents strongly disagreed, 9 disagreed, 8 were uncertain, 13 agreed and 8 strongly agreed about it, as shown in Q2 graph in fig. 9. In the end we asked about bug viewing and searching facility in system. 23 respondents strongly disagreed, 7 disagreed, 8 were uncertain, 11 agreed and 13 strongly agreed that system has efficient bug search feature, as shown in Q3 graph in Fig. 9.

Atheistic and minimal design leads to optimal design of interface and dialog boxes.

System should not give irrelevant and confusing information so that users don't get confused. We asked whether their bug tracking application provides optimal information in dialog boxes and error messages. 14 respondents strongly disagreed, 15 disagreed, 6 remained uncertain, 20 agreed and 5 strongly agreed, as shown in Q1 in (**Fig 10).**

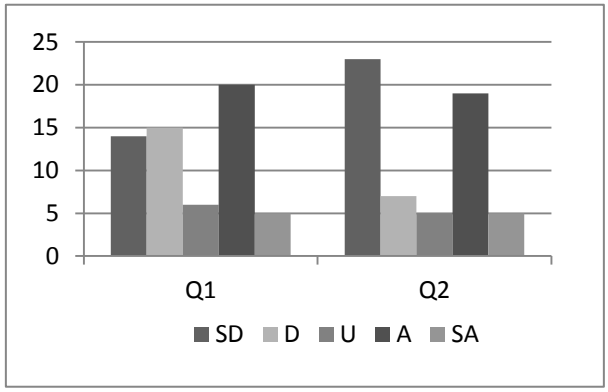

Fig. 10 Atheistic & minimal design

In second question we asked whether your system applies minimalistic design concept. 23 respondents strongly disagreed, 7 disagreed, 5 remained uncertain, 19 agreed and, 5 strongly agreed about it, as shown in Q2 graph in (**Fig. 10).**

Error messages must be in natural language, there should not be codes in dialog boxes and error message should indicate exact problem. In first question we asked whether their system gives understandable error messages. 15 respondents strongly disagreed, 2 disagreed, 2 remained uncertain, 20 agreed and 11 strongly agreed about it. In second question we asked whether error messages help in recovery. 25 respondents strongly disagreed, 4 disagreed, 5 were uncertain, 16 agreed and 10 strongly agreed with it, as shown in Q2 graph in (**Fig. 11).**

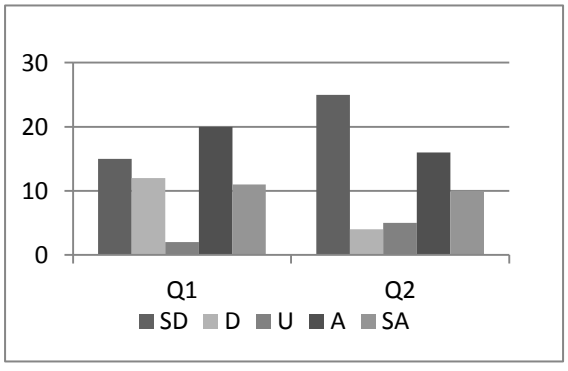

Fig. 11 User help to diagnose & recognize errors

Last heuristic is about help and documentation, as online help and proper documentation is necessary for all systems. Fig. 12 is representing results about this aspect. Firstly, we asked about presence of system feedback facility and online help availability, 17 respondents strongly disagreed, 6 disagreed, 4 were uncertain, 11 agreed and 21 strongly agreed, as shown in Q1 graph of (Fig. 12).

Then we asked about cancellation option. A cancellation option and emergency exit allows users to avoid unwanted state, if user commits unintentional mistakes. 21 respondents strongly disagreed, 7 disagreed, 6 were uncertain, 11 agreed and 15 strongly agreed about presence of such features in their bug tracking application, as shown in Q2 graph in Fig 12. Tool tips are compulsory for some of systems because they give hints and clues about purpose of icons and menus. In our survey we found that 17 respondents strongly disagreed, 9 disagreed, 7 remained uncertain, 14 agreed and 12 strongly agreed about presence of this feature in their system. In the next question we asked whether error messages are informative enough and in user understandable language. 16 respondents strongly disagreed, 9 disagreed, 9 were uncertain, 14 agreed and 12 strongly agreed about presence of this feature, as shown in Q4 in figure 12. In last question, we asked about online help and guidance availability. 19 respondents strongly disagreed, 11 disagreed, 6 were e uncertain, 7 agreed and 17 strongly agreed that their system provides this feature, as shown in Q5 graph of (**Fig. 12).**

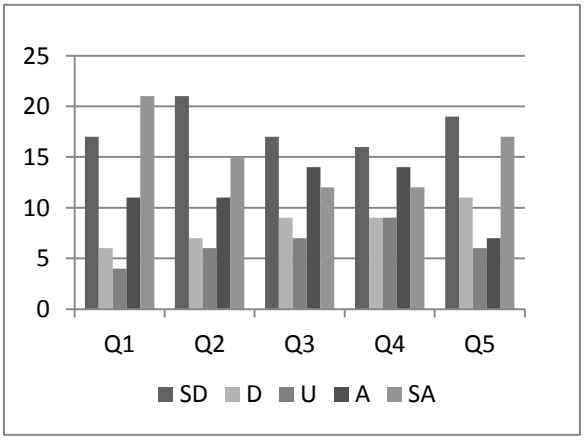

Fig. 12 Help and Documentation

4. DISCUSSION

Testing and Bug tracking is one of very important step in software development process. Different SMEs are using variety of bug trackers and they are facing different usability issues. Our survey highlights that bug trackers like code debugger, issue

view, bug herd, Redmine, Bugzilla, etc. mainly focus on functionality rather than usability.

Our survey results explored that users of bug tracking systems were satisfied from visibility of system status, they did not faced any difficulty in visibility of system when necessary actions had been done. From second heuristic we came to know that mostly users of BTS can understand iconic representation and menus. But it totally depends on user experience and how much training they had received on particular BTS. Novice users have faced some difficulties in iconic representation.

Mostly users agreed that system provide emergency exits. If they perform undesired action they can easily went back on previous state. But in some bug trackers like Bugzilla and ITrack didn't provides much flexibility. Similarly users were not satisfied from interfaces consistency. Furthermore, some users' responses showed that sometimes error messages didn't help to resolve them. Due to difficult interface design some users could not find necessary icons and menus. It was also discovered that online documentation and online guide was accessible but no tool tips facility were provided.

5. CONCLUSION

Organizations have adopted technology so that they can improve their work practices and facilitate end users including developers in all ways. Bug tracking applications play significant role in software development. Our qualitative study highlights that users of different bug tracking system are facing usability issues and there is need to further support them by appropriating these bug tracking systems according to users practices. Furthermore automation can also be done for without making difficult user interfaces keeping in view the smaller effective feature sets.

REFERENCES:

Baysal, O., M. W. Godfrey and R. Cohen, (2009). A bug you like: A framework for automated assignment of bugs. In *Program Comprehension, 2009. ICPC'09. IEEE 17th International Conference on* 297-298. IEEE.

Baysal, O., R. Holmes, and M. W. Godfrey, (2013). Situational awareness: personalizing issue tracking systems. In *Proceedings of the 2013 International Conference on Software Engineering* 1185-1188. IEEE Press.Bugzilla [Online]. http://www.bugzilla.org/docs.

Just, S., R. Premraj and T. Zimmermann, (2008). Towards the next generation of bug tracking systems. In *Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on* 82-85. IEEE September.

Khanjani, A., and Sulaiman, R. (2011, March). The process of quality assurance under open source software development. In *Computers & Informatics (ISCI), 2011 IEEE Symposium on* 548-552. IEEE.

Lin, Z., F. Shu, Y. Yang, C. Hu, and Q. Wang, (2009,). An empirical study on bug assignment automation using Chinese bug data. In *Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on* 451-455. IEEE.

Li, J., T. Stalhane, R. Conradi, and J. M. Kristiansen, (2012). Enhancing Defect Tracking Systems to Facilitate Software Quality Improvement. October *IEEE software*, *29* (2), 23-28.

Nielsen, J. (1994). *Usability engineering*. Elsevier.

Rubey, R. J., L. A., Browning, and A. R. Roberts, (1989). Cost effectiveness of software quality assurance. In *Aerospace and Electronics Conference, 1989. NAECON 1989., Proceedings of the IEEE 1989* May *National* 1614-1620. IEEE.

Saeed, S., F. M. Khawaja and Z. Mahmood, (2012). A Review of Software Quality Methodologies. *Advanced Automated Software Testing: Frameworks for Refined Practice*, 129.

Saeed, S., and I. Alsmadi, (2013). Knowledge-Based Processes in Software Development, Hershey, PA: IGI Global.

Serrano, N. and I. Ciordia, (2005). Bugzilla, ITracker, and other bug trackers.*Software, IEEE*, *22*(2), 11-13.

Zimmermann, T., R. Premraj, J. Sillito, and S. Breu, (2009). Improving bug tracking systems. In *Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on* 247-250, IEEE.