Evaluating User Participation in Defect Reporting among Free/Open Source Software Projects

Anu Gupta[1] *R.K. Singla*[2]

Abstract

Free/Open Source Software (F/OSS) is an incredible and innovative opportunity of software development in the area of software engineering. The F/OSS phenomenon allows the entire Internet community to use its combined programming knowledge, creativity and expertise to develop software solutions, which could render benefits to whole community without involving cost and proprietary issues. The globally distributed F/OSS users and developers contribute in form of bugs identification, feature request, patch submission, translation request etc. These characteristics give a large boost to software development resulting in more effective software production, with true field testing and fast defect fixing. The present paper delves into a multi-case study of F/OSS projects to evaluate user participation in defect reporting. The relevant defect data has been retrieved from a research collaboratory and used to quantify and analyze the user participation for defect reporting. It is found that among F/OSS projects defect arrival is quite fluctuating and inconsistent but showing an overall downward trend. A large pool of users participates effectively for bug identification in comparison to feature request, patch submission etc.

Keywords : Open Source, Free Software, Defect, bug, DMS, DART.

1. Introduction

Since 1950s there has been a tradition of sharing and cooperation in the software development among IBM and DEC user groups without restrictions [1]. But with the widespread diffusion of the Internet, the scale and formalization of the activity have expanded significantly and inspired a novel model of software development in the form of Free/Open Source Software (F/OSS). The origin of Free Software dates back to early 1980s with the foundation of "Free Software" (FSF) movement by Richard Stallman of MIT's. Artificial Intelligence Laboratory [2]. Stallman intended to encourage development of software that would come with source code and be available to users for their own modifications [3]. A key feature of FSF based development is a licensing scheme called "Copyleft". The Copyleft license provides unique distribution terms that give all users the rights to use, modify and redistribute the program code or any program derived from it but only if the distribution terms are unchanged [2]. All users are compelled to leave copies behind for others to benefit. The philosophy of the FSF movement has later been extended by a number of individuals who are promoting the "Open Source" concept [4]. These individuals are less concerned about the freeness of "Free Software" and are instead interested in encouraging software companies to release source code for their products [5]. So Free/Open Source Software (F/OSS) is a broad term used to embrace software that is developed and released under some sort of Free/Open Source license but all allowing inspection of the software's source code. F/OSS development model has generated increasing interest in the last few years,

[1&2] Department of Computer Science and Applications Panjab University, Chandigarh. Email : anugupta@pu.ac.in, rksingla@pu.ac.in.

both from the business as well as academic world. Academic interest into this new form of collaborative software development has arisen from very different backgrounds including software engineering, sociology, management or psychology. Today there is hardly any computer application with no corresponding F/OSS initiative. Linux provides a credible alternative to Microsoft Windows, while OpenOffice.org seeks to replace Microsoft Office. Mozilla's web browser, Firefox, rapidly appropriated a large segment of Internet Explorer's market. And the list goes on: mySQL (database management), Thunderbird (e-mail), Apache (Web server), Compiere (SME ERP system) etc.

The rest of the paper is organized as follows; Section 2 discusses the organizational structure of F/OSS participants and the factors motivating their participation in F/OSS projects. Section 3 presents specific research questions to be explored. Section 4 discusses research methodology where sample selection and data extraction mechanism are described. Section 5 highlights the quantitative results. Section 6 addresses answers to research questions. Finally, Section 7 concludes and provides directions for future work.

2. Organizational Structure Of F/OSS Developers And Users

The F/OSS projects are initiated and developed in very interesting ways. Generally a team comprising one or more developers develops a crude version of the software to solve some personal problem. This development team may keep the software to itself, sell the software under commercial terms or it may release the software as F/OSS. Last option will cost nothing to the development team, but some of the potential users might provide feedback and contribute to the development. Contributions can be in form of bug reports, feature requests; patches added or in form of translation requests, support requests etc. Most contributions will contain some sort of user insight which the development team might find useful. However, some of the suggestions may not be valid, thus rejected and not being incorporated in the F/OSS project. A common classification of the hierarchy of roles that people take and common tasks they perform when participating in an F/OSS project is shown in Figure-1[6].

Most of the F/OSS projects have a core team of few dedicated developers surrounded by a larger ring of few hundred interested collaborators who perform field testing and further surrounded by thousands or tens of thousands of users who may only be using the project [7]. However some users may eventually migrate from the outer ring to the inner rings. The important point is that F/OSS makes it possible for an aspiring and technically capable software developer to play a larger role through continual contributions. There are several driving factors which motivate user participation in F/OSS development:

- **Need For Product:** Participating in order to create, customize, or improve a product or feature which they require for their personal need [5] [8] [9].
- **Enjoyment, Desire to Create and Improve:** Participating because one enjoys it; finds creating or improving software interesting [10].
- **Reputation and Status Within the Community:** Participating in order to build or maintain reputation or status within the community [5] [11].
- **Affiliation:** Participating in order to socialize or spend time with like-minded individuals [5].
- **Values and Ideology:** Participating to promote specific ideals e.g. the free software philosophy [3] [5] [10].
- **Learning and Career Concerns:** Participating to improve one's skills, with the belief that such

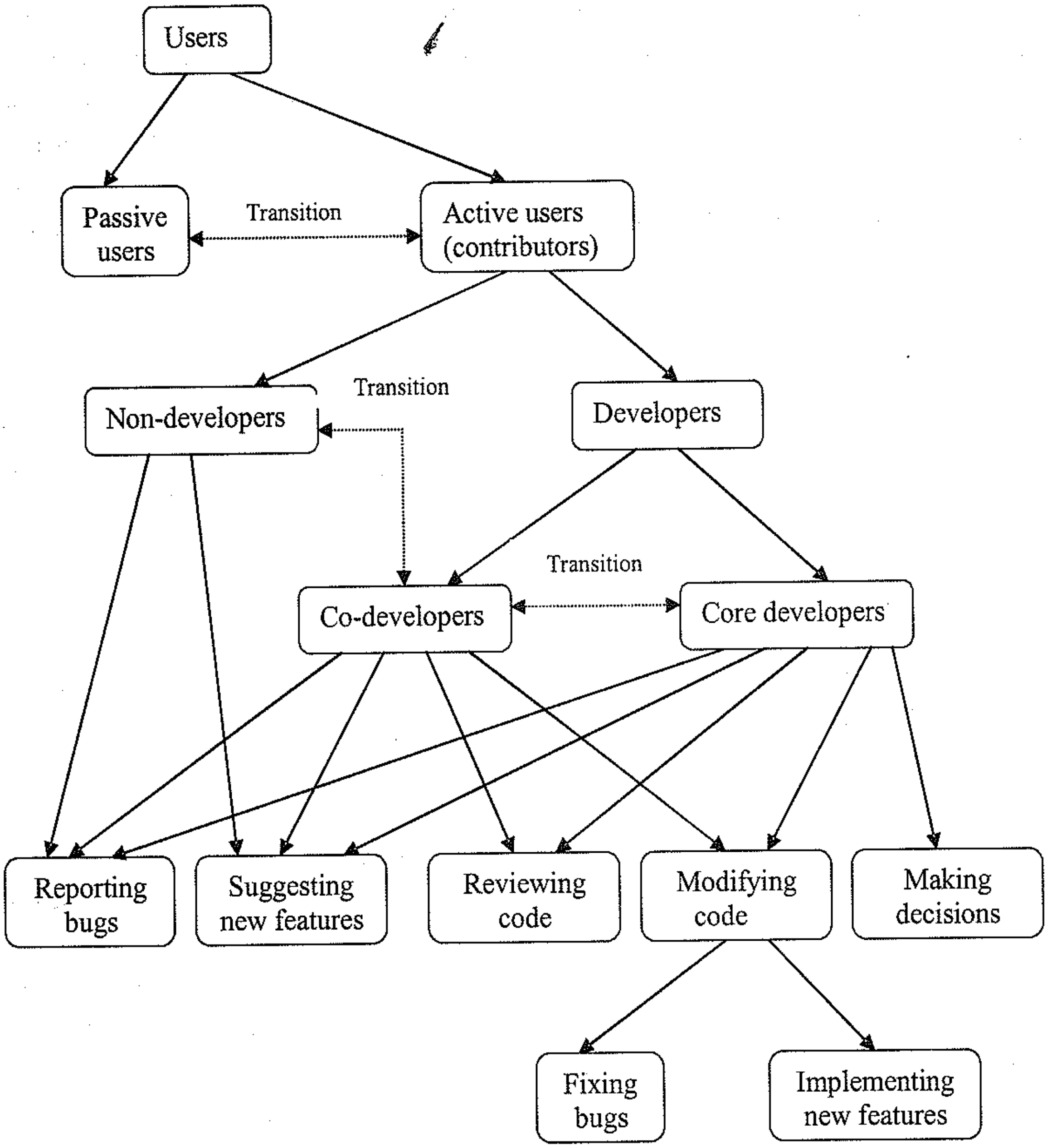

Figure1: The Classification of F/OSS Users and Developers [6]

improvement will lead to a better job or promotion [5] [11] [12] [13].

More people looking at the code will results in more defects found, which is likely to accelerate software improvements [4][5]. It harnesses the collective wisdom, experiences, expertise and requirements of its most demanding users to ensure that their needs are rapidly met. Thus participation of F/OSS users plays a role in software development, enhancement and quality assurance. The role of user participation is highly acclaimed by the proponents of F/OSS phenomenon. But there exists few empirical evidences for determining the effectiveness of user participation among F/OSS projects. A study of popular Apache web server and Mozilla web browser quantified aspects of developer participation, core team size etc. by using e-mail archives of source code change history and defect reports [14]. Another study analyzed the temporal changes among various F/OSS projects and discussed the distribution of defects among various categories on the basis of statistics provided by SourceFforge.net [15]. A study of two open bug repositories from the Eclipse and Firefox projects provided an initial characterization of bug reporting and resolution process [16]. A study of four F/OSS projects from SourceForge.net discussed the distribution of defects on the basis of type and resolution [17]. A survey highlighted that F/OSS developers perceive defect identification of users quite high while participation for feature request reasonably well [18]. It is found that some very successful projects have caught the attention of researchers generally. Many of such projects are either led by established communities or sponsored/supported by commercial organizations. The current paper presents a multi-case study of F/OSS projects and attempt to quantify the effectiveness of user participation in defect reporting specifically.

3. Aim of Study

F/OSS development is a case of continuous as well as participatory design which mixes different software development phases and does not elaborate stop criteria to the design process [19]. So design becomes continuous, new functionalities can always be proposed and discussed whatever may be the phase in the project. Moreover unrestricted access of source code encourage peer review and code inspection by a geographically distributed community of users which helps to identify defects as well as enables them to be removed. The community rarely happens to meet each other face to face. Rather communication is established online through F/OSS project web site, Version management system, Defect management system, mailing lists, discussion forums, bulletin boards etc. Online Defect Management System (DMS) provides an interface which enables many such active users to record their feedback regarding defects. It also contains the state and outcome when the defect is resolved.

DMS provide valuable defect data, which in combination with other data can provide considerable information about the quality of the F/OSS project, the processes, and the prospects for future development [20]. These data can be combined, analyzed and presented in several different ways. For example a defect may be accompanied by injection time, severity, priority, type, status, contributor etc. Defect data collected from users are quite valuable source to measure several in-process metrics [20]. During our research, we are considering defect based metrics into two broad categories: one related to defect reporting aspect and other related to defect resolution aspect. Current paper is focused upon evaluating the effectiveness of F/OSS user participation for defect reporting among F/OSS projects. The specific research questions which we are aiming to address in the present study are as follows:

1. What is the rate of defect arrival? Does it show some trend over the period? Does it show some relationship with other project characteristics like Total Downloads, Version Release etc.?

➢ The question aims to find out the active participation of users in defect reporting over the period. Here Defect Arrival Pattern over time interval (in months) and linear trend lines are used for the analysis.

2. How defect arrival is distributed among various defect categories? Is there significant contribution towards one category in comparison to others?

➢ The question aims to know the proportion of user participation in each of the defect categories i.e. bug reports, feature requests, patches, miscellaneous (support requests, translation requests etc.) and find their relative significance.

3. How many users are participating in defects reporting with respect to number of downloads as well as in relation to core team size? Are they participating regularly?

➢ The question aims to know the participation of users versus number of downloads as well as core team size in defect reporting process. It is also desired to find whether users are regularly contributing or occasionally.

4. What percentage of reported defects is really considered to be effective?

➢ The question aims to measure the extent to which reported defects are effective. Hence Defect locating effectiveness of the participating community is evaluated.

4. Research Methodology

To address our research questions mentioned above, a study of various F/OSS projects is carried out. In this section, we discuss sample selection and defect data extraction mechanism applied to the present study.

4.1. Sample Selection

F/OSS Projects under study are selected from SourceForge.net, a collaborative development site that currently supports the development of about 2 millions of F/OSS projects [21]. A single source was chosen to select projects to control for differences in available tools and project visibility. In spite of millions of projects hosted, it is found that only a small proportion of these projects are actually active. Also many of the projects hosted do not use or do not allow public access to Defect Management System. Hence those projects are selected for which defect related data is publicly accessible and is being maintained completely at Sourceforge.net. Another criterion used for selection of projects is the project development stage (1-6 where 1 is the planning stage while 6 is a mature stage). A cut-off of 5 is chosen which indicates that the selected projects are at a similar stage of development and are not in the early stage of development lifecycle. The selected 20 projects constitute a diverse mix of project size, team size, nature of application and targeted end user type. Selection of limited number of projects has helped to carry out in-depth study. A brief description of the selected projects is reported in Table-1, including the Project Type, Registration Date on Sourceforge.net, Total Downloads, Core Team Size, Number of Participants etc.

4.2 Defect Data Extraction

The process used for Defect Data Extraction is presented in Figure- 2 and discussed as follows:

❖ Collaborative Development Environments such as SourceForge.net allow F/OSS participants to carry out all the tasks mentioned in the Figure-1 in Section 2. Sourceforge.net provides access to a standard toolset consisting of a web server for hosting F/OSS Project content, Version Management System (i.e. Subversion referred as SVN), Defect Management System (i.e. Trac), Mailing lists with archives (i.e. GNU Mailman), documentation manager etc. These tools enable participants to collaborate in the software development process as well as also act as repositories to store the communication activities of the participants. To answer research questions

Table1: Selected F/OSS Projects

S. No.	Project Name	Project Type	Registration Date on SourceForge.net	No. of Downloads (Till Oct. 2008)	Core Team Size	No. of Participants (Reporting Defects)	Aggregate Defect Records (Till Oct. 2008)
1.	SquirrelMail	Email Client	18/11/1999	13,61,358	11	1,447	4,340
2.	phpMyAdmin	Database Administration	18/3/2001	59,53,722	11	2,569	6,987
3.	Gallery	Image Galleries	18/6/2000	20,76,283	19	1,926	5,416
4.	Privoxy	Security	9/9/2000	2,26,219	16	595	4,594
5.	TCL	Interpreter	5/9/2000	13,75,879	48	598	5,412
6.	Fink	Installer	27/12/2000	16,62,311	93	1,746	6,448
7.	Webmin	System Administration	2/1/2001	39,70,738	4	908	3,944
8.	HSQLDB	Database Engine/Server	21/3/2001	3,13,494	35	489	1,382
9.	JBoss	Application Server	15/3/2001	51,18,948	83	1,315	3,456
10.	NSIS	Build Tool	5/3/2001	3,17,048	13	449	1,567
11.	TortoiseCVS	CVS (Concurrent Version System	3/3/2002	6,65,520	40	1,101	2,162
12.	NASM	Cross Compiler	28/5/2000	5,98,290	17	116	602
13.	aMSN	Messenger Application	22/5/2002	29,74,954	25	1,071	2,969
14.	PDFCreator	Office Suites, Printing	13/7/2002	13,91,889	4	469	1,659
15.	IPCop Firewall	Firewalls	23/11/2001	16,30,415	39	758	1,260
16.	GanttProject	Project Management, Scheduling	29/1/2003	2,55,758	6	624	1,264
17.	hipergate CRM	CRM	10/9/2003	19,136	6	31	971
18.	KeePass	Password Manager for Windows	15/11/2003	1,28,494	28	457	1,880
19.	ClamWin	Anti virus	26/3/2004	2,72,409	4	287	856
20.	Azureus	Bittorrent	24/6/2003	3,61,35,722	33	2,344	3,401

mentioned in Section 3, all the defect records entered by F/OSS participants through DMS for each of the selected F/OSS projects are needed.

- The traditional way of obtaining research data is spidering or crawling of Sourceforge.net using Perl or Python scripts. But there are certain pitfalls in spidering. It is a time consuming as well as resource consuming process, even then satisfactory data is not obtained [22]. If SourceForge.net detects that you're hitting its site too much, it bans your IP address. So instead of direct access to the Sourceforge.net repository, latest approach is to reuse data that other researchers obtained from Sourceforge.net such as the FLOSSmole [23], SRDA [24]. FLOSSmole periodically collects and parses the information about several repositories like Sourceforge, Rubyforge, ObjectWeb, FSF etc. and stores them in the database, then releases the data in several different formats. The stored information contains statistics for a particular period generally related to Project developers, Programming languages, OS, Status, Project downloads, Project ranks, Project Defect tracker sums etc. As per our research requirement, detailed defect data is not available here.

- SourceForge.net has shared valuable data of its huge repository of about two million projects with the University of Notre Dame to build a research collaboratory and has given permission to share this data with other academic researchers studying the

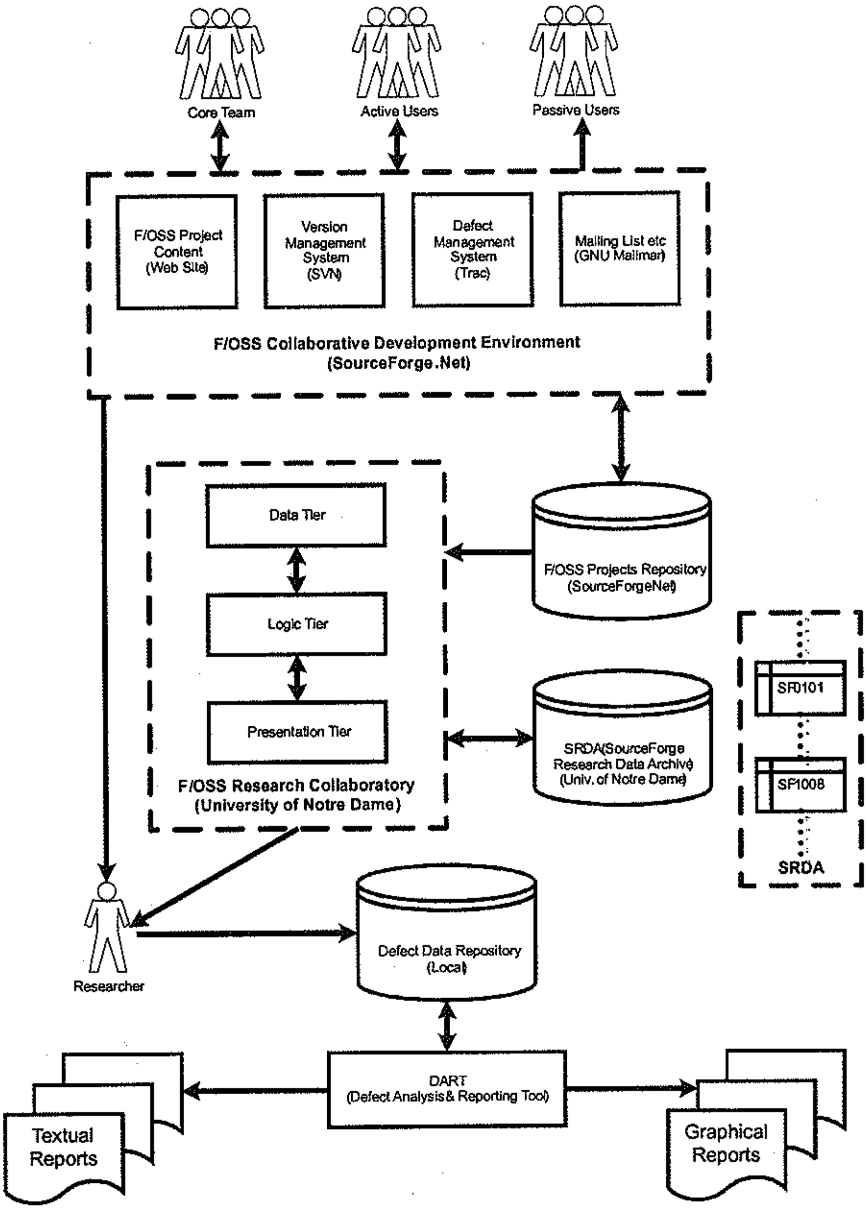
Core Team
Active Users
Passive Users
F/OSS Project Content (Web Site)
Version Management System (SVN)
Defect Management System (Trac)
Mailing List etc (GNU Mailman)
F/OSS Collaborative Development Environment (SourceForge.Net)
Data Tier
Logic Tier
Presentation Tier
F/OSS Research Collaboratory (University of Notre Dame)
F/OSS Projects Repository (SourceForgeNet)
SRDA(SourceForge Research Data Archive) (Univ. of Notre Dame)
SF0101
SF1008
SRDA
Researcher
Defect Data Repository (Local)
DART (Defect Analysis & Reporting Tool)
Textual Reports
Graphical Reports

Figure 2 : A Conceptual Framework for Defect Data Extraction and Analysis

Free/Open Source Software [24]. On a monthly basis, a complete dump of the F/OSS projects repository (except some data dropped for privacy and security reasons) is shared with University of Notre Dame. We signed an agreement with University of Notre Dame and justified the usage of repository for our research purpose. Then we were provided login and password to access the database dump.

A three-tier hierarchy is used to implement the research collaboratory at University of Notre Dame [25].

- ➘ **Presentation Tier**: Presentation tier provides restricted repository access as well as community support for the researchers.
- ➘ **Logic Tier:** Logic tier includes all the processes implementing every function in the presentation tier.
- ➘ **Data Tier**: A multiple-layer data schema is used in the data repository where every schema represents a single month and stores the database dump from SourceForge.net of the corresponding month. These schemas are named from the month of the corresponding database dump. For example, schema "SF0108" stores the database dump of January 2008. The other tasks of this tier include an appropriate backup strategy and load balancing technique.

☛ There are over 100 tables in the database. ER-diagram and schema browsers were studied in detail to have clear idea about all the tables, relations among them and to identify research data of our interest. Detailed defect data for all the selected projects from their respective date of registration to October 2008 is downloaded on the basis of unique Project ID assigned to each project at SourceForge.net. Several queries joining multiple tables were executed against database schema for October 2008 (sf1008) to fetch the required data.

☛ The detailed defect data for all the selected projects has been downloaded and stored in individual project tables in local database. For every project, each of the defect records contains its unique Defect ID, Type (bugs, feature request, patch and others), Status (open, pending, deleted, closed), Resolution (fixed, invalid, duplicate, later, remind etc.), Priority (1-9), ID of Submitter, ID of Person to whom assigned, Opening date and Closing date.

An application (Defect Analysis & Reporting Tool i.e. DART) is developed to retrieve relevant data from local defect repository and process according to our aim of study discussed in section 4. The application generates various kinds of reports in textual and graphical formats.

5. Quantitative Results

The detailed results obtained are being presented with the help of various graphs in the following subsections:

5.1 Defect Arrival

An F/OSS user is free to submit his feedback through online Defect Management System whenever a problem is encountered which need to be reported. The defects reported by F/OSS users are referred as defect arrivals or defects newly opened. Defect arrivals over the period help to know how actively users are reporting their problems towards an F/OSS project and give a picture of overall efforts of geographically distributed community of users.

Live Defect arrival data consolidated on monthly basis has been plotted in form of line graphs to observe the defect arrival pattern. Figure-3 shows one such graph for 5 projects out of selected 20 F/OSS projects. It is found that during the initial few months, most of projects have very few defect arrivals as projects have less number of downloads as well as less users. With the passage of time defect arrivals increase because projects become popular, the number of download as well as users increase. It has also been observed that defect arrivals start decreasing after some time which signifies that project is moving towards stabilization. We have also observed that in all the projects the defect arrival pattern is quite inconsistent and fluctuating. An attempt was made to analyze the fluctuating pattern of defect arrivals in context of project release and number of downloads. In some of the projects it has been found that spike in defect arrival pattern occurs after major/minor release. It has also been found that with new release number of downloads increase because existing users as well as prospective users download the projects.

The linear trend lines are also plotted corresponding to inconsistent and fluctuating defects arrival. The trend lines shown in Figure-3 indicate the overall downward trends which prove that defects arrival tend to decrease as projects move towards stability.

5.2 Defect Distribution

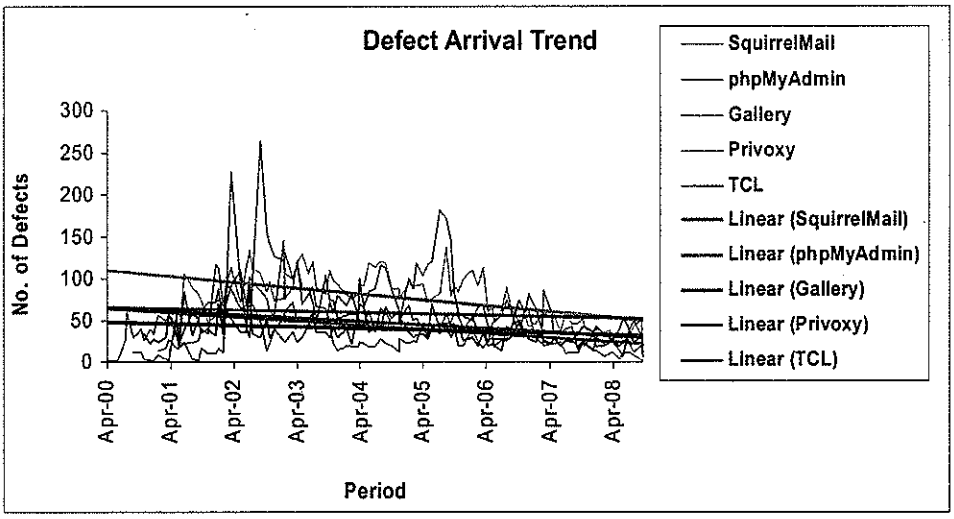

Figure 3

F/OSS users can participate in the form of bug reports, feature requests, patches added or miscellaneous (translation requests, support requests, plug-ins, package requests or any other project specific category not covered under any of the main three categories etc.). The graphs have been plotted to show the distribution of defects into above mentioned categories. One such graph for 5 F/OSS projects is shown in Figure-4. In most of the projects it has been observed that major contribution from users is in form of bug reports. In some of the projects, feature requests also play significant role. But contribution of users for adding patches is very low and hardly there is a project having this percentage in double digit figure. It has also been observed that miscellaneous category does not exist for many projects while it is quite significant in other projects

The graph has been plotted to show the mean distribution of defects among various categories (Figure-5). It shows that 55% of total contribution is from bug reports, 21%

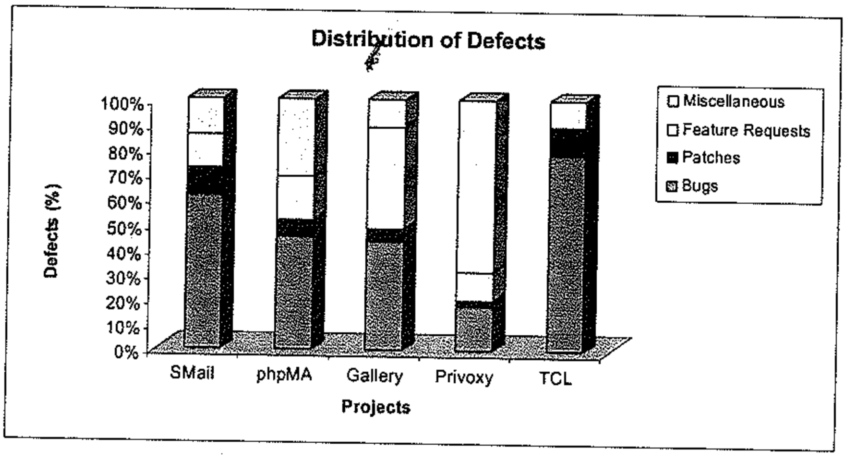

Figure 4

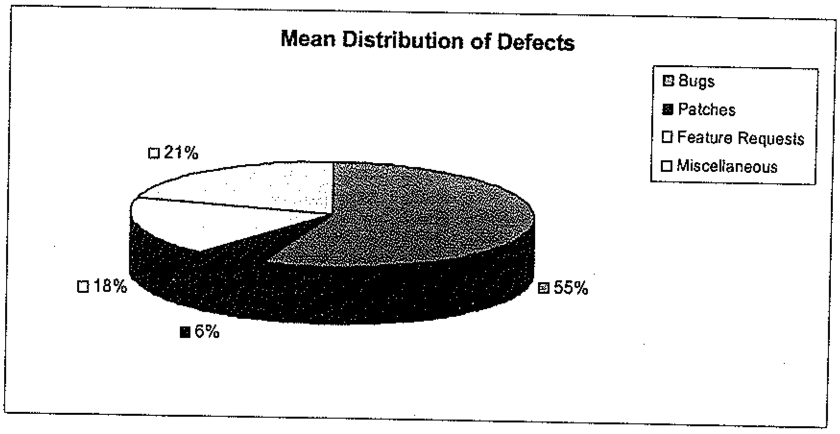

Figure 5

from miscellaneous, 18% from feature requests and only 6% is from patches on an average basis. The analysis of defect distribution proves that the F/OSS users are helping core team significantly in identifying bugs and moderately in requesting new feature but not much in the development.

5.3 User Participation

Raymond, in his seminal work, describes the importance of users [5]. Every project needs a constituency of users who use the project, want the project to work, and are sufficiently committed to make at least some efforts toward improving it. The consistent involvement of such users makes the discovery and elimination of defects easier and quicker [26].

It is seen that all the projects have quite large number of downloads. These downloads could be made by existing users for a new version release or prospective users may download it in anticipation of use. There could be circumstances that a project is repeatedly downloaded or downloaded but never used. Assuming only 10% of total downloads are actually being used, graphs have been plotted to compare the Total Downloads, 10% of Total Downloads and number of distinct participating users. One such graph for five F/OSS projects is shown in Figure-6. It is observed that even in comparison to 10% of total downloads; the number of users contributing for defects is scanty.

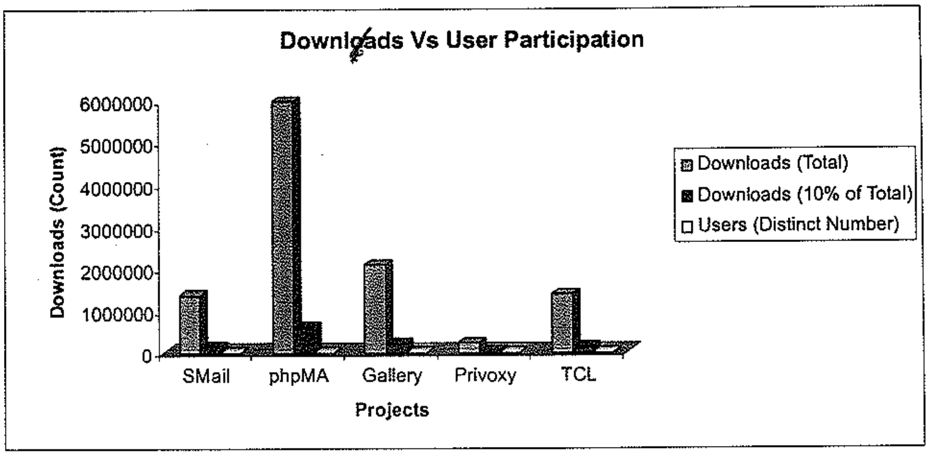

Figure 6

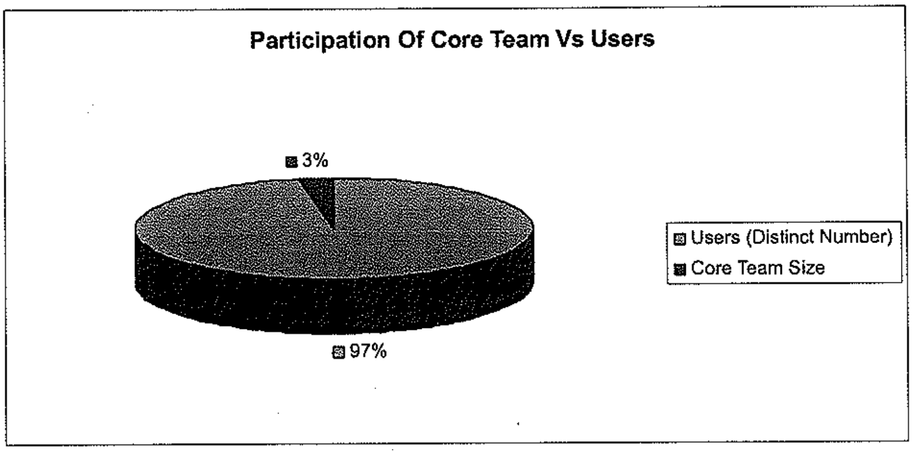

Figure 7

The graph has been plotted to compare average number of users participating in defects with average core team size [Figure-7]. It is found that generally a small team is surrounded by a large community of users participating in defects. Collaborative efforts of in order to participate in a project, a user is supposed to get registered on the project web site. But many users also participate through guest login referred as anonymous user. A graph (Figure-8) has been plotted to compare the participation of anonymous users with other registered users. It is found that quite large contribution is from anonymous users i.e. 30% on an average. Looking at the number and contribution made largely geographically distributed community help to make bugs shallow.

By individual users, it has been observed that the most of the registered users are not participating regularly. The graphs has been plotted to show the number of entries made by distinctive participating users. Figure-9 shows graph for five F/OSS projects. Looking at the average of 20 projects, about 91% of the users are contributing once, twice or thrice. A very few users are regular users (Figure-10).

5.4 Defect Locating Effectiveness

The defects reported by F/OSS users may be processed with the resolution status set to Invalid, Won't fix, Out of date, Duplicate, Works for me and rejected. In many cases defect status may be set to "Deleted" but such entries are archived in the defect database. All these types of defects will not be contributing towards the projects, hence may be referred as ineffective defects. These ineffective defects need to be filtered out to have a clear picture of all effective defects contributed by users. The various types of ineffective defects are defined as follows:

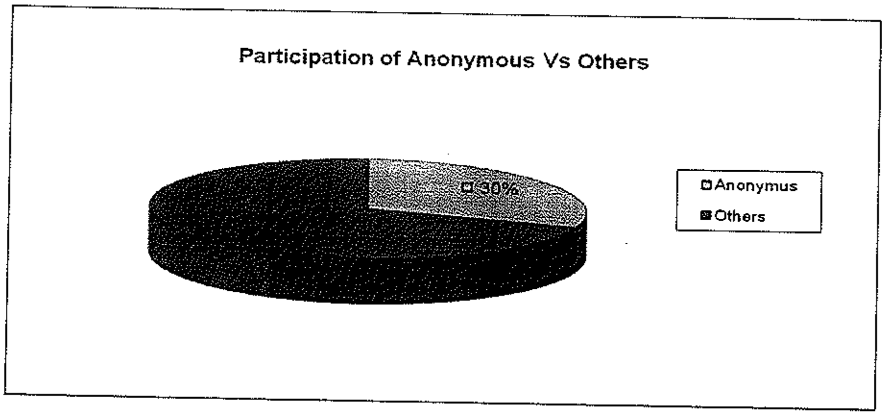

Figure 8

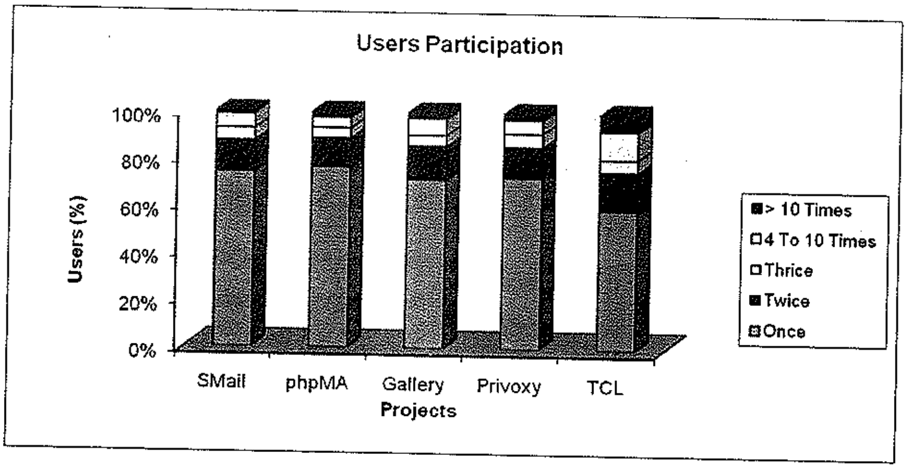

Figure 9

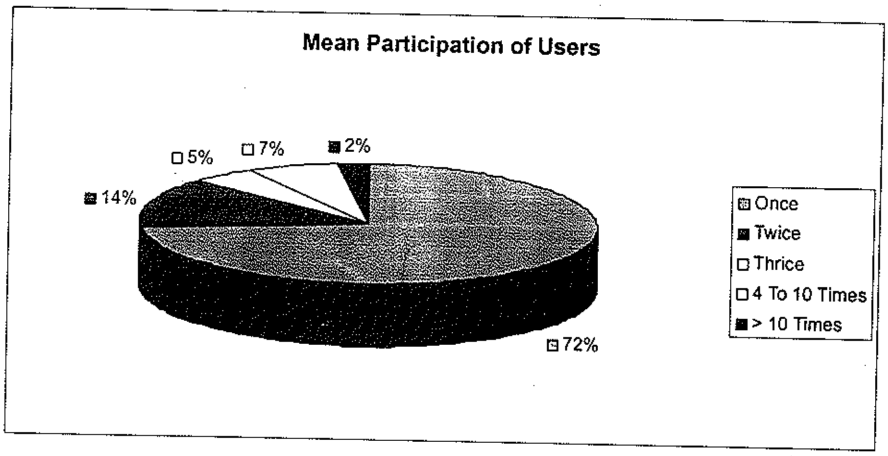

Figure 10

- **Invalid:** The reported defect is invalid or irrelevant.
- **Wont Fix:** The reported defect will not be fixed due to some constraints like software design, technology, resources etc.
- **Duplicate:** The reported defect is a duplicate of an existing one.
- **Out of Date:** The reported defect is already considered/reviewed/resolved and hence it is now outdated.
- **Works for me:** The reported defect is not reproducible by the development team.
- **Rejected:** The "Rejected" resolution is generic one and some of the projects use it rather than specifically classifying the resolution into Invalid, Won't fix, Out of date, Duplicate and Works for me.
- **Deleted:** In many projects, if defect resolution is any one of the above mentioned resolutions then the defect status is set to "Deleted".

The graphs have been plotted to show the distribution of defects into effective defects and above mentioned ineffective status categories. Figure-11 shows a graph for 5 projects.

Following graph (Figure-12) has been drawn to highlight the average distribution into effective defects and the various ineffective defect categories for all the selected projects. It is found that on an average 79% of total defects are found to be effective.

Looking at the distribution of ineffective defects among various categories, it is found that invalid defects are slightly higher than other types of ineffective defects. While won't fix, Out of date, Duplicate, Works for me and Rejected forms smaller but almost equal sized proportion.

6. Discussion

We observe that F/OSS projects typically consist of a small team of core developers who oversee the design, development and evolution of the project. These active developers are surrounded by a vast pool of users. A subset of such users use the latest releases and contribute bug reports or feature requests and many times submit patches also. During the past few years, adoption of F/OSS has increased considerably. Thereby it has raised the interest of people to explore and participate in the development process of F/OSS projects too. Current study focused on evaluating the effectiveness of user

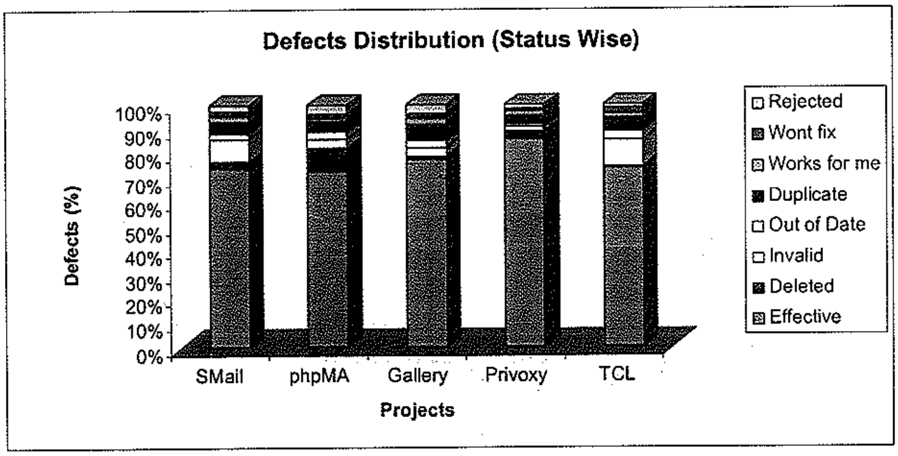

Figure 11

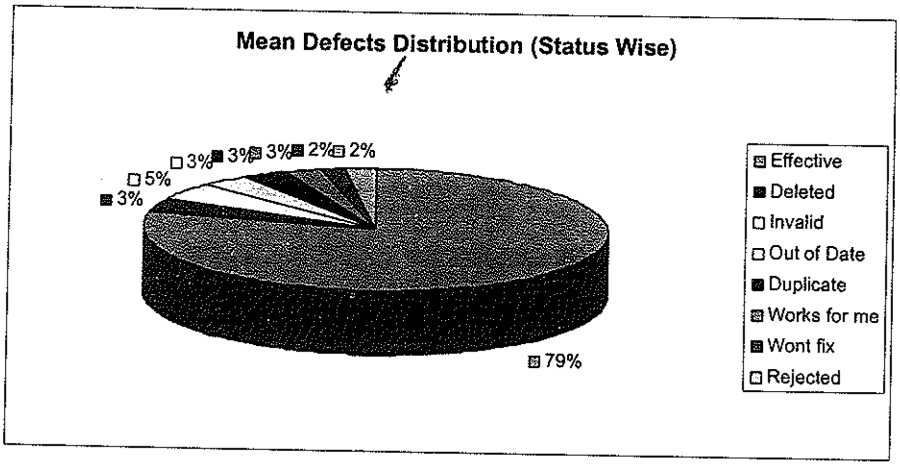

Figure 12

participation for defect reporting among F/OSS projects. Based on the results reported in section 5, we attempt to address our research questions.

1. What is the rate of defect arrival? Does it show some trend over the period? Does it show some relationship with other project characteristics like Downloads, Version Release etc.?
   Answer: It is observed that in all the projects the defect arrival pattern is quite inconsistent and fluctuating. During the initial few months of project hosting, defect arrival is very low. But later the increasing downloads results in increasing users; defect arrivals go high and then start diminishing. An attempt was made to analyze the fluctuating pattern of defect arrival in context of project release and number of downloads. In some of the projects it has been found that spike in defect arrival pattern occurs after major/minor release. It has also been found that with new release, number of downloads increases because existing users as well as prospective users downloads the project. But trend lines corresponding to defect arrival patterns highlight an overall downward trend which signifies that projects are moving towards stabilization.

2. How defect arrival is distributed among various categories? Is there significant contribution towards one category in comparison to others?
   Answer: F/OSS users can participate in variety of ways through web enabled interface of collaborative development sites. In most of the projects it has been observed that major contribution from users is in form of bug reports. In some of the projects, feature requests also play significant role. But contribution of users for adding patches is very low. The analysis of defect distribution proves that the F/OSS users are helping core team significantly in identifying bugs and moderately in requesting new feature but not much in the development by adding patches.

3. How many users are participating in defect reporting with respect to number of downloads as well as in relation to core team size? Are they participating regularly?
   Answer: The consistent involvement of global community of users makes the discovery and elimination of defects easier as well as quicker [5]. It is seen that all the projects have quite large number of downloads. These downloads could be made by existing users for a new version release

or prospective users may download it in anticipation of use. There could be circumstances that a project is repeatedly downloaded or downloaded but never used.

It is observed that in comparison to total number of downloads; the number of contributing users for defects is quite low. But in comparison to core team size, this figure seems to be quite significant. Collaborative efforts of this largely geographically distributed community help to make bugs shallow. Figures about number of times each user is participating are quite astonishing.

About one third of total defects are made by users who are not registered on the Sourceforge.net. There is no specific reason for anonymous participation but it seems that some users either do not want to disclose their identity or do not want to follow the lengthy procedure of registration. It has been observed that even the most of the registered users are not participating regularly. Looking at the average of 20 projects, most of the users are participating once, twice or thrice contributing about one third of defects. A very few users are participating more than three times but their active participation is contributing the remaining one third of the defects.

4. What percentage of defects reported are really considered to be effective?

   Answer: Every type of contribution made by a user can not be valid. Many times duplicate defects are entered. There could be cases when defect can not be reproduced. Some defects may be outdated. All such cases would be leading to ineffective defects. These ineffective defects need to be filtered from the total reported defects to know all those defects which need to be fixed. On an average about 20% of aggregate defects are found to be ineffective. Among Ineffective defects, Invalid defects form the highest percentage.

7. Conclusion And Future Work

Generally F/OSS projects are developed by a small team of core developers which is surrounded by a community consisting of large number of globally distributed users. Although size of this globally distributed community is very small in comparison to total downloads for the project, but their continual involvement is quite significant in identifying bugs and moderate in requesting features. But F/OSS projects are not able to attract much contribution in form of patch submissions. Frequent releases in F/OSS projects result in inconsistent defect arrival pattern but an overall down trend indicates gradual progress in stability and quality of the F/OSS projects. The current work focused on evaluating the participation of users from the perspective of defect reporting only. The future work will be carried out to study the participation of F/OSS users in defect resolution and determine the efficiency of defect resolution process among F/OSS projects.

References

1. A. Hars, S. Ou, *"Working for Free? Motivations for Participating in Open-Source Projects"*, International Journal of Electronic Commerce, Vol. 6, No. 3, PP. 25-39, 2002.
2. Free Software Foundation, http://www.fsf.org/
3. R. Stallman, *"Why software should be free"*, http://www.gnu.org/philosophy/shouldbefree.html.
4. Open Source Initiative, http://www.opensource.org/
5. E. S. Raymond, *"The Cathedral and the Bazaar"*, http://www.catb.org/~esr/writings/cathedral-bazaar.

6. C. Gacek, B. Arief, "*The Many Meanings of Open Source*", IEEE Software,Vol. 21, No. 1, PP.34-40, 2004.

7. K. Crowston, J. Howison, "*The Social Structure of Free and Open Source Software Development*", First Monday, Vol. 10 , No: 2, 2005.

8. J. Kuan, "*Open Source Software As Consumer Integration into Production*", http://ssrn.com/abstract=259648.

9. N. Frank, E. Hippel, "*Satisfying Heterogeneous User Needs via Innovation Toolkits:The Case of Apache Security Software*", MIT Sloan School Working Paper No. 4341-02, 2002, http://ssrn.com/abstract=299419.

10. R. Goldman, R. P. Gabriel, "*Innovation Happens Elsewhere : Open Source as Business Strategy*", Morgan Kaufmann, ISBN 1-55860-889-3, 2005. http://dreamsongs.com/IHE/IHE.html

11. K. R. Lakhani, E. Hippel, "*How Open Source Software Works:"Free" User-to-User Assistance*", MIT Sloan Working Paper No. 4117-00(2000), http://ssrn.com/abstract=290305.

12. J. Lerner, J. Tirole, "*The Simple Economics of Open Source*", Journal of Industrial Economics,Vol. 52, PP.197-234, 2002.

13. J. Roberts, I. Hann, Sandra Slaughter, "*Understanding the Motivations, Participation and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects*", Marshall School of Business Working Paper No. IOM 01-06, http://ssrn.com/abstract=918518.

14. A. Mockus, R. Fielding, J. Herbsleb, "*Two Case Studies of Open Source Software Development: Apache and Mozilla*", ACM Transactions on Software Engineering and Methodology, Vol. 11, No. 3, PP.309–346, 2002.

15. D. Weiss, "*A Large Crawl and Quantitative Analysis Of Open Source Projects Hosted On Sourceforge*", Research Report ra-001/05, 2005, Institute of Computing Science, Pozna University of Technology, Poland, http://www.cs.put.poznan.pl/dweiss/xml/publications/index.xml.

16. J. Anvik, L. Hiew, G. Murphy, "*Coping with an Open Bug Repository*", OOPSLA workshop on Eclipse technology exchange, San Diego, California; ISBN:1-59593-342-5,35 – 39, 2005.

17. H. Lintula, T. Koponen, V. Hotti, "*Exploring the Maintenance Process through the Defect Management in the Open Source Projects -Four Case Studies*", International Conference on Software Engineering Advances (ICSEA 2006); October 29 - November 3, 2006 - Tahiti, French Polynesia.

18. Anu Gupta, Ravinder Kumar Singla, "*An Empirical Investigation of Defect Management in Free/Open Source Software Projects*", Advances in Computer and Information Sciences and Engineering, Springer Netherlands; ISBN 978-1-4020-8740-0(Print),978-1-4020-8741-7 (Online); 68-73.

19. L. Gasser, W. Scacchi, G. Ripoche, B. Penne, "*Understanding Continuous Design in F/OSS Project*", Communication at ICSSEA 2003, Paris, France.

20. S. Kan, "*Metrics and Models in Software Quality Engineering*", Pearson Publications,ISBN: 81-317-0324-X, 2002.

21. Sourceforge.net; http://www.sourceforge.net/

22. J. Howison, K. Crowston, "*The Perils and Pitfalls of Mining SourceForge*", International Workshop on Mining Software Repositories (MSR 2004), Scotland, United Kingdom, May 23-28 2004.

23. J. Howison, M. Conklin, K. Crowston, "*Flossmole: A Collaborative Repository For OSS Research Data*

And Analyses", International Journal of Information Technology and Web Engineering, Vol. 1, No. 3, 17-26.

24. G. Madey, *"The SourceForge Research Data Archive (SRDA)"*, University of Notre Dame; http://zerlot.cse.nd.edu/

25. Y. Gao, V. Antwerp, M. Christley, G. Madey, *"A Research Collaboratory for Open Source Software Research"*, First International Workshop on Emerging Trends in FLOSS Research and Development(FLOSS '07), May 20-26, 2007.

26. Scacchi, *"Is Open Source Software Development Faster, Better, and Cheaper than Software Engineering"*, 2nd ICSE Workshop on Open Source Software Engineering, 2002.

Author's Biography

Mrs. Anu Gupta has been working as Assistant Professor in Computer Science and Applications at Panjab University, Chandigarh since July 1998. Currently she is also holding the position of Chairperson, Department of Computer Science & Applications. She was awarded University medal for securing first position in M.C.A. at Punjabi University, Patiala, Punjab in the year 1997. She has the experience of working on several platforms using a variety of development tools and application packages. She is currently pursuing the Doctor of Philosophy Degree from Panjab University in the area of Free/Open Source Software. Her research interests include Networking, Multimedia Technologies, E-Commerce and Software Engineering. She is a life-member of 'Computer Society of India' and 'Indian Academy of Science'. She has published several research papers in various journals and conferences.

Dr. R.K. Singla has been Professor of Computer Science and Applications at Panjab University, Chandigarh since July 2004 where he has been a faculty member since 1988. He held many IT-related positions including Chairman, Department of Computer Science & Applications; Coordinator, TIFAC-DST; Project Leader; and Programmer/Analyst. His experience covers large mainframes, minicomputers, microcomputers, and several types of specialized equipment. He has programmed in over a dozen languages including Visual Basic, C, C++, COBOL, FORTRAN, Pascal, several versions of assembly, and machine language. He has the experience of working on several platforms including VAX-VMS, SUN-Solaris, HP-UX etc. using a variety of development tools and application packages. He obtained the Doctor of Philosophy Degree in Faculty of Science from Panjab University. His research interests include scientific computing, linux networking, mobile computing, open source software and software cost estimation. He is also a member of editorial board of 'Panjab University Research Journal (Science)-New Series'. He is a life-member of 'Computer Society of India'. He has published 35 research papers in various journals and conferences.