A REVIEW OF ISSUE ANALYSIS IN OPEN SOURCE SOFTWARE DEVELOPMENT

KAREEM ULLAH, SHAHBAZ. A. KHAN

Research Scholar, DSE, International Islamic University, Islamabad, PAKISTAN
Asst. Professor, DSE, International Islamic University, Islamabad, PAKISTAN

ABSTRACT

A software development methodology was initiated from the concept of sharing the source code. In 1998 it was named as open source software development. It is a complete software development methodology, in which all the participants are connected through internet. It has great promises to produce high quality software projects with low cost, which is difficult to do in preparatory software development. The objective of this review is to highlight the issues of open source software development. To achieve the benefits of OSSD, we have to perform risk assessment first. These issues produce great hurdles in the way of success. We have conducted a review of issue analysis. In this review issues are identifies, by which it is easy to mitigate them. We have analyzed the empirical studies. These studies have provided us lot of evidence for issues. These reported issues are needed to mitigate with a systematic way. By removing the limitations of OSSD, it is easy to develop the software project with low cost and high quality.

Key words: *Open Source Software Development, General Public License, And Proprietary Software Development, Traditional Software Development, Open Source Software Project.*

1. INTRODUCTION:

OSS systems can be defined as "Software for which the human-readable source code is available for use, study, reuse, modification, enhancement and redistribution by the users of that software" [01]. In OSSD model everyone can access source code freely. Source code of project can be used for any purpose and also redistributable. In commercial software development the source code is restricted for usage, modification and redistribution except the owner. It is argued that.

"Open source is a license. It's a legal issue if you look at it that way. It's a license to apply to the work that is freely available. If you look at it from a development process, it's a philosophy. It's a different development method. And the way of communication is something different".

2. WHAT IS OSSD?

OSSD is actually depend upon the volunteers, the contributor rarely meet face to face and connect through internet. While is PSD the contributors are paid member of team [02]. The word of open source is representing the software development process that depends on the physically dispersed developers, which are connected by internet. Accessibility of source code is the main characteristic of OSSD. There is a strict requirement for OSS that is its source code is freely available for every one. That can be used for any personal purpose. The user of the OSS is always free to modify, or improve the system for personal use. Physically distributed developers are connected on internet in OSSD, but it's not a strict requirement. Lot of companies and individuals has developed the software source code with (PSD) pattern. We can take an example of Netscape web browser (Mozilla project) [03].In OSSD, system is developed by an individual or team of software developers. A prototype system is released on the Internet, which can be accessible freely and system's source code can be read, modify and redistributed [04]. Eric Raymond characterizes Bazaar model, which is used for OSSD, where source code are allow for redistribution and modification. Hundred of small and large scale projects included Linux, Mozilla and apache use it as software development model [05].

To analyze the (OSSD) approach, a framework is existed, which describes OSSD completely. This framework has include many categories which are What, Why, When and Where, How, Who. It is

similar to CATWOE technique that is described in 1981, as Client, Actor, Transformation, Weltanschauung, Owner and Environment. These elements has a correspondence with OSSD framework, as "what and transformation", "why and weltanschauung". There are some successful OSSP that are listed as. "Linux, apache, Sendmail, BIND, PERL Python, GNU software, GNOME, KDE, Mozilla, Jikes, GIMP, Darmin, SAMBA, Ghostcript, Doom" [06]. According to the Sourceforge.net, there are thirteen types of OSS projects. These are listed as "Clustering, database, desktop, development, enterprise, financial, games, hardware, multimedia, networking, security, sysadmin and VOIP". There are 100,000 registered projects that are hosted on Sourceforge.net as (OSS) projects [07].

The success full OSS project works in two prospective the user and the software. In user prospective there are few reasons as following. OSS project has used low cost distribution channels as compare to traditional software projects. The code quality is well in OSS, which makes easy to modify the code for new market opportunities. In OSS high collaboration will promotes the sharing of ideas, which will generate a reliable solution. In software prospective developer involvement has a major effect on project, stated as "Add the developers late in project, makes it later". Short term feedback in OSS project make easy to save development effort [08].

In OSSD reason of producing high quality code is the continuous release of next version. Continuous release cycle directed to fulfill the market requirements in next version, which does not considered in PSD. In OSSD two principal exists, named as "released early" and "released often". In the release early principal project is rapidly evaluated, by which new system is developed without wastage of time. In the principal of release often the solution of problem is developed by lot of developers, therefore probability of solution increased [04].

3. WHY OSSD?

Purposes of OSS systems are to provide the independencies' to use the system for any purpose [09]. It is claimed that OSSD plays a major role to produce bug free code with low cost and less time. Team of OSS project work together not for monetary return, instead they are interested to become part of OSSP. All the participants of OSS project are from different geographically locations and never meet face to face. All the community wants good repute and to increase their experience [10].

The OSSP is initiated due to some of reasons. The system is developed for general purpose, need of system for services communities like scientist and defense industry, system is developed by certain communities that are interested to fix the project problems [08]. There is a great difference between OSSD and TSD in several ways. The major goal of OSP is to develop a system to solve a problem and interesting for the persons who are working on it. In OSSD the developers are volunteers that are often unpaid. They are working as a hobby and in return they receive the peer recognition and personal satisfaction of their work. This means that the work that has been done on OSSD is concentrate on the developer's point of view, rather then what is the point of view of a developer. OSSD towards a particular goal is quite difficult to manage. But the owner of the project holds little power on the developer's contribution. Due to this freedom it is difficult to force the developers to perform the vital tasks like systematic testing or code streamlining that is not attractive as writing new code. They are free to every task but it's depending upon the owner of the project to accept or reject their contribution [11] [12].

4. WHAT IS NOT OSSD?

In 1968 software crisis is highlighted, as a long term development cycle, high cost and low quality but no one highlight t he (OSSD) approach as a silver bullet. These are some major properties of (OSSD) that are following.

Source code is available freely for every one. Source code can be redistributable. Everyone can modify the source code. By licensing source code can be restricted fir any use. The word open source is often confusing with freeware, shareware, royalty free and use restricted. There are minor differences between these terminologies which can be understandable by the table 1.

Table 1: Difference between licensing terminologies

Type of software	Null price	Distribution allowed	Unrestricted users	Free availability of source code	Amendable source code
Commercial	NO	NO	NO	NO	NO
Use-restricted	YES	YES	NO	NO	NO
Shareware	YES	YES	NO	NO	NO
Freeware	Yes	YES	YES	NO	NO
Royalty-Free binaries	YES	YES	YES	NO	NO
Royalty free libraries	YES	YES	YES	YES	NO
Open Source	YES	YES	YES	YES	YES

When describing OSS and Rapid application development then few differences are identified as following?

In OSS the availability of source code is critical feature, which is reverse in RAD. In OSS large no of developers work on large project, but in RAD there are small project with few developers. Multiuser systems are developed in OSS, reverse in RAD. In OSS the application area is much more complex, as compare to RAD. The frequent release in OSS makes it different from RAD. The users are also coder in OSS, which is never possible in RAD. The system is never finished, it is continuously incremented, but in RAD the project should be ended [06].

Matthias Sturmer describes the overview of OSSD, and life cycle of OSP. OSSD is started by initiator of the project. The solution of problem is proposed by the initiator for initial support and feedback. When the code is in public access then, it is announced on news channels to participate in the project development. The interested people provide feedback and then become the part of OSS community. There participation is accepted or rejected by the project initiator (owner). These participant are volunteers who participate only for enhancement of there interpersonal skill and personal repute. The end of lifecycle depends upon the software project life. Matthias Sturmer also describes different roles of OSSD, as contributors, developers, leader, core developer, project owner and initiator (first contributor of OSSD, in the start all source code is mostly developed by the initiator). All these roles collectively become the community [13].

To difference between TSD and OSSD there are some basic reasons. There is large number of volunteers who has participated in OSSD. There are no works assigned to the people that are taking part in OSS systems. No system level design is exist in OSSD. No project planed, scheduled and no list of deliverables exists. These differences propose a tremendous case of physically distributed development in which software developers are doing there assigned work in random locations, not often meet each other face to face. Therefore the coordination of developers during the OSSD is done by email and bulletin boards [14]. In 2007 a framework was proposed to better analysis of QA activities during the development. The framework has three process groups, which are following. Process group-I that is the process for detecting the defect and analyze to ignore or remove the defect. Process group II is the defect detection, which verify the existence of defects and take actions to correct the defects. Process group III is solution verification which verifies the solution of the detected defects [15].

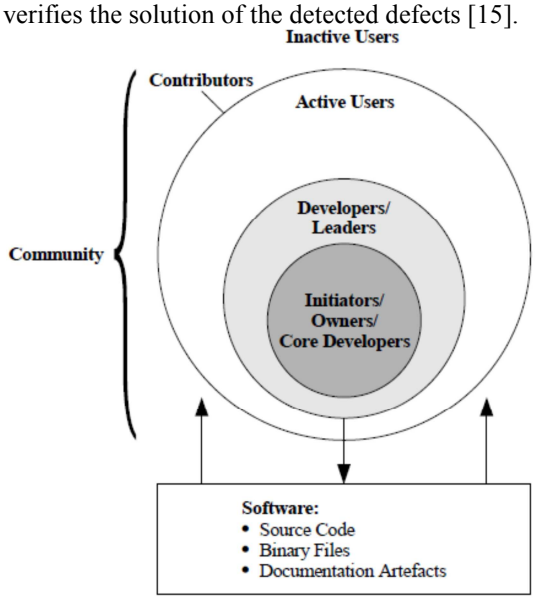

Figure 1: Abstract view of open source software projects.

When some one talk about open source then, a number of opinions is exists in mind. These are open software, open collaboration, open process, open release, open deployment and open environment. When open software has released then use of open source license to allow the further source code distribution redistribution and licensing. In open collaboration the involvement is depend upon the discussion groups, virtual meeting rooms, and shared assets of development. Open process that means the external view of development and the product release. That may include the external auction of projects and coordination for resources. Open deployment that focuses on the new products releases. Open environment that directed toward the use of open source products in system development.

Free software advocate Richard Stallman and free software foundation describe that OSSD relating to complete software development methodology [16]. There is no difference between free software and OSS, both can be access free and modified. OSS is quite different from public domain and shareware (Michael P.Cnlon) [17].

As compare to traditional development, OSSD has much more better results. It is claimed that "that defects are found and fixed very quickly because there are many eyeballs looking for the problems" [18]. In OSSD there is an originator, who has initiate the project, and ready to invite others into the project. The originator of project makes the code available for the developers to proceeds the development process. So that anyone can be participate into the development of system. But the owner of the project (originator) is free to decide about the contributions of the developers to include or not for official release of the project [12].

5. ROLES F OSSD:

One of the differences between OSSD and TSD is the roles and responsibility. No tasks are assigned to the participants, they done their work according to the personal interest unlike the TSD. In OSSD there are eight roles that are following. Normal users, which use the system for high quality, are called passive users. Active users are the reader of the system. They understand the functionality of the system with the help of source code. Peer reviewer is working similar to reader in OSSD. Discovering and reporting the bugs is called bug reporter. They do not read and modify the source code. The discovered bugs are fixed by the bug fixer. Bug fixer read the source code and searching the bug to report. Peripheral developers are the developers that are irregular contributors. There period for participation in short. Active developers are regularly contributed in the development. There contribution is the major development force of OSS systems. Responsibility of core members is to direct the direction of OSS project development. They are the individuals who have been participated in to project and make lot of contributions for long term. They are also called the maintainers. Project leaders are the initiator of the project and responsible for managing the direction of project [12] [19]. In OSSD Committer has a major responsibility to release the project. Committer checks the developer's contribution according to the quality prospective to verify the quality criteria. When the software product achieves the satisfaction level of quality then it is decided to release, otherwise it move toward developers to remove the defects [15].

The feature of OSD includes the scheduling, code quality, unstable code, planned evolution, testing and preventive maintenance. There is little pressure of schedule to complete the project. Most of the developers are part time participators; they have done their jobs also. Due to part time participation the development cycle may long. Therefore the open source software's are unaffected from time to market pressure. Without the satisfaction of owner of project the system is not released. There is not fix code quality and standard used in OSSD. It's very difficult to insist on particular standards. Unstable code is submitted often by the developers for the contribution to the project. In Linux this problem is solved by synchronized development paths. There is a development release path and stable release path. When the new features are being added then they are transferred into the stable released path. OSD required the active participation but not essentially careful reflection and reorganization. Especially parallel debugging is used for maintaining code quality except systematic testing and other planning prescriptive approaches [03]. Free software is forever accessible as OSS, but OSS is not forever free software [20].

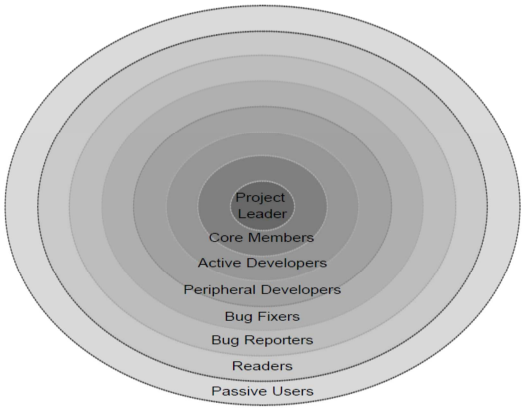

Figure 1: Roles in OSSD

6. LICENSES IN OSSD:

To understand the open source software's there are two categories exist. Non-copylefted and copylefted, non copylefted software has a legal word that is not copylefted. Modified versions are permitted to add for more restrictions. That may be free at all. Any one can use it as preparatory software. Copylefted software is not allowed to add more restrictions on its modification and redistribution. GNU general publication license is covering the distributions of copylefted software's. When we are talking about the proprietary software's then there are three categories exist that are closed, shareware and freeware software. In closed software the source code is not available to public. Not distribution and modification is allowed. In freeware and shareware people allowed to redistribute, use and download the software freely. But can not be modified because the release is executable binary format only. To use a shareware, license is required. A common confusion for freeware and free software is found. In freeware the modification is impossible because source code is not available [09] [17].

Opensource.org has listed about 21 approved licenses, which are certified valid by open source initiatives (OSI). There are three most famous licenses that has a large usage are GNU general public license, Berkley software distribution, Mozilla public license. It is copy left license that was developed in 1980. All the projects at that time are using GPL. BSD is a copyright license, which is also free to use. Many OSS projects (yahoo. apple, Microsoft) are using BSD. The mixture of BSD and GPL is MPL. It was developed by Netscape for Mozilla project [21]. It is described on OSI website that a project should be publish under a license which fulfill 10 OSS criteria's. Availability of source code and well structured source code is the compulsory criteria for OSS [13].

7. CRITICAL REVIEW AND ANALYSIS OF LITERATURE ON OSS:

Growing rate of OSSD projects has increased very fast. Estimation of Source Forge Web site is that more then 500,000 users with newly joined 700 per day and more then 50,000 projects with 60 newly increased per day [22][23]. It is estimated by Gartner group that 85% of enterprise companies has adopting the OSS and remaining 15% are moving to adopt in next few years [24].A survey is conducted about the usage of OSS that shows 13 European countries reporting that the usage rate of OSS is 78%. Survey is also conducted in US and reported that 87% of organizations are using the OSS systems [25].

In 2008 survey estimates that 50% web sites run on apache web server all the e-mail routing use send mail through internet. Source forge portal that is a largest open source project host, has 158669 registered projects only 27004 projects included in stable projects, only 2414 projects included in mature projects. It is concluded that health of open source community is responsible for failure and success of OSSD [26].

In OSSD approximately 91.7% developers do not use OSS tools in planning phase, in design phase 83.3% developers did not use OSS tools, in testing phase 91.7% developer did not use OSS tools. Developers use OSS tools mostly in Operating system, database, programming language. Some of the successful OSS projects have great usage. As Linux (operating system), MYSQL (database), and PHP (popular programming language) have 75% usage, while Apache has highest usage 83.3%. The literature shows that in OSSD mainstream of developers use OSS tools only in software programming language, database system and web servers [27]. Godfrey and Tu report that Linux has very huge line of code (two millions). Its growth rate is "super-linear". It's analyzed that growing rate is low when large commercial systems have become larger, Godfrey and Tu's suggest that "OSS systems have a growth rate that is much greater than that of traditional systems" [28]. Stallman report that "worst threat to the free/open source software community comes from the use of

software patents instead of copyright as a means of protecting intellectual property rights" [29].
Open source software's are originated from 1950 s and 1960s when the software and hardware both are sold together. During 1950s and 1960s the source code of software is freely available without any restriction in IBM and DEC user group. In 1969 first version of UNIX is written by Ken Thompson, UNIX source code is accessible freely till in seventies. In 1979 UNIX in commercialized. Eric Allmann develops a system of communication between computers over ARPANET. In 1980 the software commercialization is increased [30]. In 1980 Michael Stonebreaker decided to overcome the issues of database management system therefore a project is initiated as the name of Postgres. In 1994 SQL language interpreter is added in Postgres by Andrew Yu and Jolly Chen, therefore the new version is released named by Postgres95. Later it become the POSTGRESQL, which is the first relational database system in OSSD. There are 17 former and 49 active participators, who contribute in POSTGRESQL.

In 1983 GNU Manifesto for free software is published by Richard Stallman to establish the free software foundation. In 1986 Perl (Practical Extraction and report Language) that is a flexible programming language is developed by Larry Wall.CGI (common Gateway Interface) script in written by Perl. In 1987 MINIX that is the version of UNIX is released with complete source code by Andrew Tanenbaum for the PC, Mac, Amiga, and Atari ST. In 1991 new version of UNIX 0.02 is developed by Linus Torvald which is called LINEX. Linus Torvald wants a better operating system for his laptop. After frustrating from MINIX he decided to build his own operating system. To develop UNIX like operating system, he invites the programmers through news groups for contribution in the project. The programmer, whose are also frustrated like Torvalds take interest in contribution. The Linux version 2.6.0 has 680 development days with 916 developers that participate at least for one change in code. There are 27,149 changes happened which has average of 1.66 changes per hour during the development. Source code of Linux was not controlled and maintained by any formal organization. Chief developer (Linus Torvalds) and lieutenants are responsible for acceptance and rejection of code [17].

In 1993 Free BSD version 0.1 is released. Debian Linux that is the new Linux distribution is developed by Lan Murdock. Red Hat Linux that is leading Linux distribution is created by Marc Ewing in 1994. Apache group created the web server in 1995 that is leading HTTP server today. Apache software foundation had developed "a patchy server" by which the name apache server built. MYSQL developed in 1995 as OSS and at the same time as preparatory software. It is claimed that approximately 11 million installations with 50000 MYSQL downloads per day. Netscape releases the Mozilla with source code in 1998. Some of the major software vendors are decided to port their product to Linux. In 1999 users of Linux are reached at 7.5 Million. In 1999 Star Division merges into Sun Microsystems. OpenOffice released as OSS by Sun Microsystems, which name was turn over and become openoffice.org. Another company also has similar trademark. Openoffice.org has 151 developers. Development of virtual reality systems is automated by Alice, which was started in 1997 by Rany Pausch. Due to forking, direction of Alice was changed and released the version 2.0. Alice becomes the system to teach the programming languages. Only selected developers were allowed to contributors. Alice is developed and released as OSS but with paid developers. Development of Alice was rapidly slow (1999-2007), therefore no other release exist since 2007.

In 2000 products versions (which run on Linux) are released by Novell, Real and more software companies [30]. 30 years open source software progress without name from 1970-2000. Many of the successful and broadly used software packages are developed in this period [29].

A survey was conducted in Japanese software industry to know the existing position of OSS development in 2001. To conduct the survey SRA was granted, which is a well reputed company. From 1987 (FSF) is supported by SRA. In different countries SRA has highlight the OSS projects of different categories and evaluate the presented business and governmental support for OSS development [12]. It is reported by Gartner that growing rate of Linux is fastest, for server market [13]. "Approximately 50% of the web sites are run on Apache web server". A survey was conducted, about 36.6 million web severs world wide reports that apache web server has 60% market share (2002) [10]. Yet the open source software projects are facing lot of problems. As reported by Source forge portal which is a major host of Open Source projects. In March 2009 out

of 158669 registered projects only 17% projects are stable and 1.52% projects are reached at mature status. In OSSD the failure and success are depends upon the open source community. In OSD the developers are physically distributed and hardly ever meet face to face. They communicate with one another through email and bulletin boards [13].

Success of OSS is measured by system dynamic model that is presented by Evangelos Katsamakas and Nicholas Georgantzas. This model is used to answer the following questions. "How the success of a project is sensitive to initial participation, initial code-base, or timing of project relative to other projects? How does the evolution of community developer participation affect the development process and project success?"

Simulation and analysis of system dynamics model shows that the majority of the software projects are naturally unbalanced. For a specific project the commitment is limited and for volunteers it's difficult to enforce commitment. The participants are allowed to switch the project; it depends on the interest of participants. When lot of reputation is gain the extrinsically participants are free to exit the project [31].

Research and evidence show that OSSD is very much successful to producing the high quality software, but with high quality there are some of the issues exist that become failure. By achieving these issues we can achieve success. These issues are in the throughout software development process

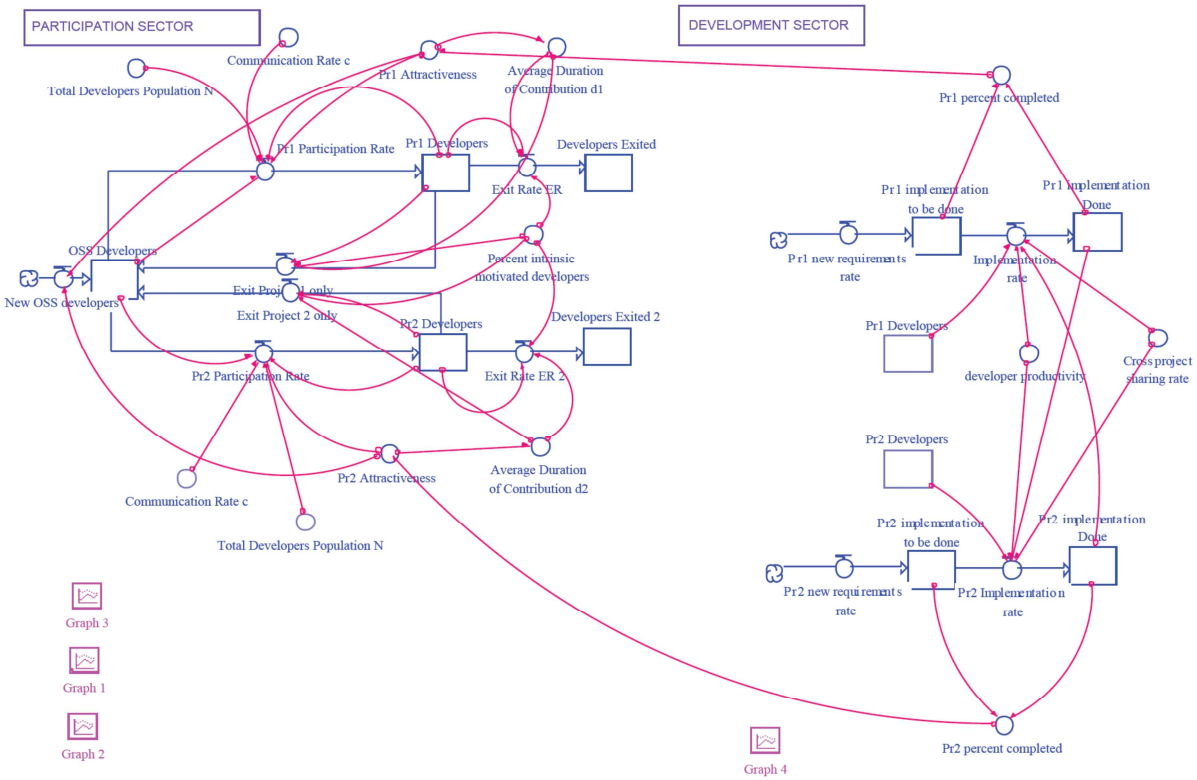

Figure 2: System Dynamic Model

8. ISSUES OF OPEN SOURCE SOFTWARE DEVELOPMENT:

In OSSD there are two issues that are consider as more difficult, that is licensing and security. It is stated that these issues must be considered in source code reuse situation [32]. OSSD is not well defined. The initiator who starts the project is responsible to manage all management activities. The remaining activities, including documentation and testing process, mostly overlook. The contributors themselves define the project requirement. System level design is ignored, most of effort has consumed on detail design.

Consumption of effort are mostly spent on coding and as well as debugging. In OSSD risk assessment does not exist and project goals are not defined properly [04].

In 1970 TEX software is developed by Donald Knuth, which is the typesetting system. Tex was released as OSS system under the BSD type license. In the case study of TEX the issue of change in technology is highlighted and it is indicated that OSSD is quite slow to develop [33]. Before starting a project, the team size and its activity status should analyzed. Without a careful analysis development of project may be more difficult than predicted and the advantage of lower costs may be eliminated. In open source projects there is leak documentation and comments in the source code, which required the effort to understand the source code. Programming skills are required for a full understanding of the source code rather if no participation is planned. Many open source projects are not actively developed, for which there is no support available. Therefore use of such unsupported software should be eliminated. Requirement change is the major problem in open source Software development because for open source project. It's not fit into the road map. Additional features are negotiated first than decided to include in project. Otherwise, customized development must manage there own software branch [34].

OSS development is facing some limitations. There is no formal process and proper design and architecture exist. In OSSD tools that are used in development like CVS are not comparable with commercial development tools. It's stated that "OSS development pattern will not be effective for software systems that has majority of the users that are not programmers", if the participants of such systems are not users, then users needs are not understandable. Therefore OSS development mechanisms will not achieve the success [35]. Most of OSSs are still being challenged by closed software because their evolution and development are often faster than those of OSSs [36].

In the development of OSSP the developers face some problems, which are following. Maintenance cost associated to quality assurance is high. Due to frequent feedback between users and developers there are many frequent "beta" releases. These releases tend toward the frustration of end user. OSS projects are also facing Platform independence issue. In OSSD the developers are interested to support the many run time configurations which become the cause of increase in use cases and QA cost. It is difficult to enforce the development according to the similar standard [08].

Electronic media and participated work among project team is the major reason for the limitations in open source software development, which are differences in culture and communication gaps [37].

Kevin Gary, Harry Koehnemann the Case Study of Unicon Inc. and JA-SIG .They highlight the some issues. Uncontrolled release cycles of the uPortal framework, so that release and configuration management must deal with a "moving target". Planning for releases of uPortal was often delayed or simply not completed due to constraints on the contributors of the project. Lack of documentation and deployment process is the huge issue in uPortal like other OSS projects. In uPortal there is no formal testing process exist. Open source products are often released without proper documentation and testing. There is poor deployment support with unfulfilled requirements. In open source software development the installation of the application is ignored [38].

In open source software development project initiator is responsible to direct the development towards particular goal, like systematic testing. But it is difficult to do so, due to freedom [03]. It's also difficult to produce user friendly interface without proper involvement of end user, this kind of involvement is limited in oss projects [33].As the development cycle moving toward more complex, the developers and users moral should be decreased accordingly [39]. In OSS systems participants have share a number of characteristics and have freedom to do work on any component of the project. No work is assigned and no deadline is defined. OSS systems are developed without system-level design, project plan, lack of a formal process, lists of deliverables, poor design and architecture. 365 versions of the Linux kernel is analyzed and reported that increase in line of code in the later version, will cause increased in the coupling. Those directed to become harder to maintain the Linux, except it is reshuffle [28].

Matthias Stuermer, Sebastian Spaeth and Georg von Krogh inspect the development of the Nokia Internet Tablet that builds by both proprietary and open source software development mechanism,

and identified "five hidden costs: difficulty to differentiate, guarding business secrets, reducing community entry barriers, giving up control, and organizational inertia". To mitigate these hidden costs, some actions are suggested [40]. Many participants are working on the same problem therefore it is possible to do duplicate work. In OSSD mechanism time and effort of participants is wasted. The time that is consumed on change are improve the requirement is also wasted. The changes may be incompatible due to many participants involvement [41].

The communication channel between lately joining and existing software developers become increasingly complex. By increasing the developers in the project the communication links are also increase. That affects on the coordination and production of the oss projects. If there are n developers the n^2 –n is the total number of communication links. In new OSS project developers will show increase in productivity, while some of them show decrease in productivity after addressing their area of interest. Low productivity is due to improper communication channel during development [42] To solve the OSSD challenges, some methods and tools are required for quality and performance improvement in open source systems. It can be possible with the help of automation by which we can minimize the human effort. It is also required to reduce the unnecessary involvement of end user [08].

9. CONCLUSION AND FUTURE WORK:

We have described the complete analysis of OSSD. In this analysis all the difference between OSSD and TSD is described. The roles and their responsibilities are identified. The type of licensees in which projects are released is also explained. We also describe the historical background of OSSD. This historical background shows the rapid increase in the OSSD. With the help of empirical evidence, we have analyzed some issues. It is impossible to remove all the issues. But with proper mitigation strategies we can be able to resolve these issues. Then it is easy to adopt the OSSD methodology. In future we are going to do a complete systematic literature review. We have lot of empirical evidence to do SLR. In this SLR we should mitigate these reported issues.

REFERENCE:

[1]. "Evaluating open source software ", Mattew Kennedy, Defence AT&L July August 2010.

[2]. Diagramming Practices in Open Source Software Development, by Eunyoung Chung, Oregon State University (2010).

[3]. Evalution in open source software: a case study (2000). Michael W. Godfrey and Qiang Tu. {migod,qtu}@swag.uwaterloo.ca

[4]. Code quality analysis in open source software development. By Ioannis Stamelos, Lefteris Angelis, Apostolos Oikonomou & Georgios L. Bleris Department of Informatics, Aristotle University of Thessaloniki, Info Systems J (2002) 12, 43–60.

[5]. Understanding the Open-Source Software Development Process: A Case Study with CVSChecker

a. Liu, Y.; Erdogmus, Hakan (2005).

[6]. A framework analysis of the open Source development paradigm. Joseph Feller Brian Fitzgerald University College Cork Ireland.

[7]. Virtual organization learning in open source software development projects, by Yoris A.Au, Darrell carpenter, Xiaogang Chen, Jan G.Clark, University of Texas at San Antonio (2007).

[8]. Leveraging Open-Source Communities to Improve the Quality & Performance of Open-Source Software by Douglas C. Schmidt Adam Porter.

[9]. "The many meanings of Open Source" Cristina Gacek, Tony Lawrie, and Budi Arief.

[10]. The Open Source Software Development Phenomenon: An Analysis Based On Social Network Theory Greg Madey (2002).

[11]. Achieving Quality in Open Source Software. Mark Aberdour, *opensourcetesting.org,* (2007)

[12]. Evolution Patterns of Open-Source Software Systems and Communities (2002).

[13]. Exploring the Impact of Socio-Technical Core Periphery Structures in Open Source Software Development.

[14]. A Case Study of Open Source Software Development: The Apache Server. Audris Mockus Roy T.

[15]. Aspects of Software Quality Assurance in Open Source Software Projects: Two Case Studies from Apache Project. By Dindin Wahyudin, Alexander Schatten, Dietmar Winkler, Stefan Biffl (2007).

[16]. Free/Open Source Software Development: Recent Research Results and Methods, by Walt Scacchi Institute for Software Research, Donald Bren School of Information and Computer Sciences, University of California, Irvine (2006)

[17]. An Examination of Initiation, Organization, Participation, Leadership, and Control of Successful Open Source Software Development Projects, by Michael P. Conlon, michael.conlon@sru.edu, (2007).

[18]. Two Case Studies of Open Source Software Development: Apache and Mozilla (2002).

[19]. Understanding Free/Open Source Software Development Processes, by Walt Sacchi, Joseph Feller, Brain Fitzgerald, Scott Hissam, and Karim Lakhani (2006).

[20]. Selection and Evaluation of Open Source Components. By Marina Marinela Gerea (2006).

[21]. Open Source Community Building by Matthias Sturmer (2005).

[22]. Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture, by Margaret S. Elliott, Walt Scacchi (2003).

[23]. Free Software: A Case Study of Software Development in a Virtual Organizational Culture, by Margaret S. Elliott, Walt Scacchi (2003).

[24]. Towards certifying the testing process of open source software: new challenges or old methologies? Sandro morasca, Davide Taibi, Davide Tosi.

[25]. Open source software adoption. Anatomy of success and failure (2008) Brian Fitzgerald, Lero – Irish Software Engineering Research Centre and University of Limerick, Ireland.

[26]. Exploring the Impact of Socio-Technical Core- Periphery Structures in Open Source Software

a. Development

[27]. A Framework of Collaborative Knowledge Management System in Open Source Software Development Environment by Modi Lakulu Faculty of Information Technology and Communication, Sultan Idris University of Education.

[28]. Open Source Software Development: A Case Study of FreeBSD, by Trung Dinh-Trong and James M. Bieman Software Assurance Laboratory Computer Science Department Colorado State University (2004).

[29]. Open Source Software: A History, by David Bretthauer, University of Connecticut (2001), dave.bretthauer@uconn.edu

[30]. Working for Free? – Motivations of Participating in Open Source projects, by Alexander Hars & Shaosong Ou, University of Southern California.

[31]. Why most open source development projects do not succeed? (2007) By Evangelos Katsamakas and Nicholas Georgantzas, Fordham University, NY Katsamakas@fordham.ed

[32]. Reusing Open-Source Software and Practices: The Impact of Open-Source on Commercial Vendors (2002). Alan W. Brown and Grady Booch.

[33]. The (LA) TEX project: A case study of open source software (2003)Alexandre Gaudeul, University of Toulouse France alexandre.gaudeul@univ-tlse1.fr

[34]. A Case Study of Developing an IDE for Embedded Software Using Open Source (2009). By Dominik.

[35]. The FreeBSD Project: A Replication Case Study of Open Source Development. (2005). Trung T. Dinh

[36]. The Evolution of Open Source Software using Eclipse Metrics (2009), by Ajlan Al-Ajlan, ajlan@dmu.ca.uk

[37]. Collaboration with learn media: how open source software succeeds. Yutaka yamauchi, Makoto Yokozawa, Takeshi Shinohara, Toru Ishida, Kyoto University Japan (2000).

[38]. A Case Study: Open Source Community and the Commercial Enterprise (2009), Kevin Gary, Harry Koehnemann, John Blakley, Cheryl Goar, Holly Mann, Al Kagan (2009).

[39]. Subversion 1.5: A Case Study in Open Source Release Mismanagement (2008). Hyrum K. Wright & Dewayne E. Perry, The University of Texas at Austin.

[40]. Extending private-collective innovation: a case study (2009), by Matthias Stuermer, Sebastian Spaeth and Georg von Krogh, Switzerland. sspaeth@ethz.ch

[41]. Continuous Integration and Quality Assurance: A Case Study of Two Open Source Projects, Jesper Holck Copenhagen Business School, Department of Informatics Denmark, jeh.inf@cbs.dk ,Niels Jørgensen, Roskilde University Denmark. nielsj@ruc.dk , (2003).

[42]. Coordination and Productivity Issues in Free Software: the Role of Brooks' Law by Paul J. Adams, Zea Partners – BE, paul.adams@zeapartners.org, Andrea Capiluppi, Cornelia Boldyreff, University of East London – UK {a.capiluppi, c.boldyreff}@uel.ac.uk , (2009).