

Master Thesis
Computer Science
Thesis no: MCS-2008:29
August 2008



Quality of the Open Source Software

Muhammad Tahir

Aleem Tariq

Department of
Interaction and System Design
School of Engineering

Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Interaction and System Design, School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to XXX weeks of full time studies.

Contact Information:

Author(s):

Muhammad Tahir

E-mail: muta07@student.bth.se

Aleem Tariq

E-mail: alta06@student.bth.se

University advisor:

Dr. Mia Persson

Department of
Interaction and System Design
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457 102 45

ABSTRACT

Quality and security of software are key factors in the software development. This thesis deals with the quality of open source software (OSS for short) and different questions that are related with open source and close source software has discussed in the thesis proposal. Open source software is a process by which we can produce cheap and qualitative software and its source could be re-use in the development of the software. Close source software is more expensive than open source software and we can not re-use its source code, so there should be a way by which cheap and qualitative software should be produced. This thesis describes that how the quality of the open source software can be evaluated and increased. There are different factors which are helpful to increase the software quality these factors are low cycle time, low cost and small size of software. These factors are also helpful for the development of software product according to the requirement of clients and the users of that software. This thesis also describes the security and performance of close and open source software and it also analyze different software development processes. The quality of OSS can be increased by use of best evaluation techniques, best quality measurement, best development tools and best development process. There are also many challenges for OSS, which it has to face for improvement of the quality of OSS. Initially there were many drawbacks in open source software but with the passage of time improvement has produced in it. This thesis also describes the compression, advantages and disadvantages of the close source and open source software.

INTRODUCTION

In our society it is known that there is no difference between free and open source software. But we should understand that licenses which are awarded to the related software make them differ from other type of software. Free software normally gets the licenses from General Public License (GPL) and open source software gets it from GPL or from other type of license authority. We know that there are two famous types of general Open Source Licenses. First is called “General Public License” and other one is know by “Berkeley Software Distribution license”. The General Public License is known as “copyleft” license and in it we can freely change the code of the software and Berkeley Software Distribution license is opposite of it and in it code become developed under the license to public domain with few, if any restrictions.[67]

We can see that free software is a part of social movement and about open source software we can pass arguments, it is only a software development methodology. In brand of free software and mostly open source software, source code can be get from distant access for the purposes of learn, alteration and redistribution. In short we can say that free software is obtainable as open source software and it is not necessary that open source software may always be free software. [67]

One advantage of the current open source development processes is that it could minimize the cycle time, quality assurance, cost and infrastructure of the software like O.S, languages tools, compilers, editors and distribution middleware. Most commonly and booming examples of Open source infrastructure software are given below, [67]

In the field of operating system Linux, NetBSD and FreeBSD are paying well role in the development of open source software. Apache and JAWS are well known by its best features in the line of web servers. MICO, ACE, TAO, JacORB are remember able and powerful distribution middleware, which are helpful for the developing the open source software.[67]

Perl, Flex and Bison are leading language processing tools and Bind, Sendmail and Samba are paying their duties in the stream of network support tools. Compilers GNU C/C++ [Stallman] and editors GNU emacs are doing wonderful performance to make the quality of the open source software development. Furthermore Apache is most usable and standard software that is using for the helping of web pages. This shows that an open source software projects are creating boom in the field of computer science. Here I want to show that Mozilla is attractive counter for Apache and it was began with weak roots and it has to faced many problems at that time but it had a bit of support of external Netscape. In the start it was near to end but after some time it become famous due to its best working. [67]

Red Hat develops such type of software that is open source, so it shows that Red Hat do not get any income from these types of open source software products. [67]

Developers and end users of FOSS (Free and Open Source Software for short) and other end users that donate their efforts to make it better for the quality of the open source software have producing the quality of OSS accordingly security and performance wise. After this struggle, millions of people obtain the benefit from it. In short, it is making possible to make the quality of

open source software better than close source software so that every body can catch its benefit on cheap or free basis. [68]

We can say that open source software progress has modified the method of software development, updating, and maintenance. This why is becoming famous in the field of software development and million of people are getting the benefit from this type of development. So, according to a careful estimate 150000 users are using Linux operating system that is also a chain of OSS, due to its low cost, high quality and trouble-free access code. [69]

Due to OSS development the re-use of source code has increased but in close source software it is impossible to study the source code of the software. If any software do not give performance according to its demand than IS manager and project leaders tries to make it better for performance wise by investigate its source code. [23]

Open-Source Challenges

Still the development process of open source software has to face many challenges for make it better and compete the close source software. We can say OSS has to face the following key challengers. [68]

- **Long-term control of maintenance, costs and quality assurance**
- **Response of Beta Release**
- **Independency of Platforms**
- **Many Compile and Run Time Configurations**

Long-term control of maintenance, costs and quality assurance

OSS has same goals as the other type of the software has, like limit regression errors, maintain the user confidence, standard and reduce the development and quality assurance costs. In the given below context in which it has developed the quality of OSS has to maintain. [68]

Response of Beta Release

As we know that firstly software has to introduce on the trial basis that is called “Beta Release”, so that the short comes of the software could be remove and best software may be launched. After release of beta version, developers receive the problems, drawbacks from the users and they demanded its solution. So there should be its quick reply so that user could use it after its final release. [68]

Independency of Platforms

Platform independency is the keystone property of open source software. A proprietary system has lack of this type of assets. OSS should be developed in this environment that it could be maintain its property and be launched on every platform. [68]

Many Compile and Run Time Configurations

Open source software has great property to support many compile time and run time configurations. This property increases the quality of the software. So it should be maximizing in OSS in future so that the quality of the open source software could be remain. Because it is fear

that due to minimal or non-existent license fees, it is hard to support more # of versions simultaneously. [68]

In this thesis, we have performed a literature study on how to improve quality in open source software during the development process. Furthermore, we will try to investigate the criteria for evaluating the open source software. According to earlier work within this field, it is possible to improve the aforementioned quality by evaluating, testing, and verifying open source software. However, due to slightly different development processes of closed software and open source software, there arise specific problems in the case of open source software. We found out that the main problems with the open source are security, finance, space science, and health.

Contents

Abstract	3
Introduction.....	4
Open-Source Challenges.....	5
1.0 Problem Definition and Expected Outcome	12
1.1 Problem Definition.....	12
1.2 Expected Outcome.....	12
2.0 Research Methodology	12
3.0 Theoretical backgrounds.....	12
3.1 How can we achieve the quality in open source software?.....	12
3.1.1 Quality of Open Source Software by Best Quality Assessment.....	13
3.1.1.1 Rahul’s OSS Selection Model	13
3.1.1.2 McCall’s Model	13
3.1.2 Quality of Open Source Software by Best Measurement	14
3.1.2.1 Metrics Choice	14
3.1.3 Quality of Open Source Software by Best Process	14
3.1.3.1 Total Quality Management.....	15
3.1.3.2 ISO 9000 Standard	15
3.1.3.3 Personal Software Process Model	15
3.1.4 Quality of Open Source Software by Best Tools.....	16
3.1.4.1 CASE Tools	16
3.1.4.2 SE Environments	16
3.1.4.3 Using of Refactoring.....	16
3.1.5 Quality and Quality Work in Market.....	17
3.1.6 Quality And Quality Work In Different Open Source Software Systems.....	18
3.1.6.1 Real Time Systems	18

3.1.6.2 Operating systems	18
3.1.6.3 Web Systems	18
3.1.6.4 Differences between Three Systems	19
3.1.7 Quality Dimensions of Open Source Software	19
3.1.8 Similarities in Software and Goods	19
3.1.9 Differences in Software and Goods	20
3.1.10 Similarities in Software and Services	20
3.1.11 Differences in Software and Services.....	20
3.1.11.1 Collection of Data.....	21
3.1.11.2 Histogram	21
3.1.11.3 Pareto Chart.....	22
3.1.11.4 Cause-and-Effect Diagrams	22
3.1.11.5 Stratification	23
3.1.11.6 Scatter Plots.....	23
3.1.11.7 Control chart.....	23
3.1.12 Quality Aspects of Supply Management	24
3.1.12 .1 Affect of Supply Process	24
3.1.12 .2 Inspection to Achieve Quality.....	24
3.1.12.3 Customer Involvement.....	24
3.1.12 .4 Continuous Improvement.....	24
3.1.12 .5 Continuous Improvement Through TQM	25
3.1.12 .6 Rational for Supplier Products	25
3.1.13 Phases of Quality Vs Open Source Software Quality Activities	25
3.1.13.1 Quality Inspection	25
3.1.13.2 Quality Control.....	26

3.1.13.3 Quality Assurance	26
3.1.13.4 Quality Management	26
3.2 Question No 2 & 3	26
3.2.1 ISO 9126 Quality Model (Recent Model)	27
Figure: ISO 9126 Software Quality Model	28
3.2.1.1 Functionality	28
3.2.1.2 Usability	29
3.2.1.3 Portability	30
3.2.1.4 Reliability	30
3.2.1.5 Efficiency	31
3.2.1.6 Maintainability	31
3.2.2 ISO 9001	32
3.2.3 Quality Evaluation Method	33
3.2.4 Process Centric Approach	33
3.2.5 Software Process and Development Method	34
3.2.5.1 Standards & Methods	34
3.2.5.2 Process Completeness	34
3.2.5.3 Modifiability & Readability	34
3.2.5.4 Traceability	34
3.2.6 Analysis	34
3.2.6.1 Sources Completeness	35
3.2.6.2 Analysis Model s Consistency	35
3.2.6.3 Analysis Model s Completeness	35
3.2.6.4 Context Definition s Completeness	35
3.2.6.5 Functionality Completeness	35

3.2.6.6 User Interface Prototype s Usability (UIP).....	35
3.2.6.7 Data Model s Correctness	35
3.2.6.8 Non Functional Requirements Completeness	35
3.2.6.9 Non Functional Requirements Verifiability	35
3.2.6.10 Verification Credibility.....	35
3.2.7 CMMI.....	36
3.2.7.1 Characteristics of CMMI / ISO.....	36
3.2.7.2 Difference between CMMI / ISO:.....	38
3.2.7.3 Strengths of CMMI Include.....	39
4.0 Comparison of open source and closed source software products	41
4.1 Problems for Open Source Software	41
4.1.1 Coordination Mechanisms	41
4.1.2 Maintainability.....	41
4.1.3 Open Availability of the Source Code	42
4.2.1 Cost Advantage.....	42
4.2.2 Free Availability of Source Code	42
4.2.3 Maturity.....	42
4.2.4 Reliability	43
4.2.5 Open Source as a New Way to Develop Software	43
4.2.6 Open Source and the Protection of Customers.....	43
4.2.7 Open Source Software and Safety	43
4.2.8 Motivation	43
4.3 Problems for Close Source Software	43
4.4 Benefits of Close Source Software.....	43
5.0 Discussion	44

6.0 Conclusions	48
Future Work.....	49
References	<u>52</u>

1.0 PROBLEM DEFINITION AND EXPECTED OUTCOME

1.1 Problem Definition

In this thesis, we consider the following research questions.

- How can we achieve the quality in open source software?
- How can we assess open source software quality with respect to various aspects of quality based on existing literature?
- Criteria for evaluating the open source software quality?
- Comparison of open source and closed source software products

1.2 Expected Outcome

The expected outcome of our thesis will be a report providing current state of research with respect to Open source software. In addition it will also provide information regarding that the reasons for adopting open source software and their importance and benefits. The results of the evaluation and comparison of open source and closed source software could be of interest both to academia and industry.

2.0 RESEARCH METHODOLOGY

Qualitative research methodology in the form of detailed and widespread literature study was carried out for this thesis. Such particular type of research methodology will enable us to gather data on open source software with respect to quality. The sources, from where data will be gathered during literature study will be articles, books and web references.

3.0 THEORETICAL BACKGROUNDS

In this section, we will consider all the aforementioned research questions with respect to existing literature.

3.1 How can we achieve the quality in open source software?

In order to be able to answer the aforementioned question on how to achieve quality in open source software, we need to answer the following question.

We can say that Software Quality is like a backbone in the field of Open Source Software development. Barry Boehm describes it with these words that “achieving high levels of user satisfaction, portability, maintainability, robustness and fitness for use”. [1]

And Jones gives it light with these words “the absence of defects that would make software either stop completely or produce unacceptable results”. [2]

Quality is an important aspect of any product. Without quality a product can't be successful. There is need of continuous improvement in quality for fulfilling the needs of customers. The route to the productivity can be achieved through quality in a cost effective way. Quality for different systems differs because the requirements are different in user's perspective.

This document contains different view points about quality and quality work between bespoke products and market driven products. It discusses the different kinds of open source software systems in terms of

quality difference among them and different dimensions of quality. Later part of the documents includes the quality tools and open source software quality activities in relation to quality movement phases.

We can use different approaches to bring the quality in the Open Source Software. These approaches are consisting of better evaluation techniques, quality measurement, better development process and tools.

3.1.1 Quality of Open Source Software by Best Quality Assessment

3.1.1.1 Rahul's OSS Selection Model

Rahul and Madanmohan present four steps for Open Source Selection Model to maintain the quality best of the software development. The steps are given below, [3]

1. In his first step he gives the suggestion to gather the information that is belonging to the development of the components.
2. Than he purposed to collect that information that have been searched.
3. Third step describes that the information that has collect that should be evaluated.
4. And in his last step he describe that there should be a critical investigation to reduce the in proper application.

We can say that this model is purpose to evolutes & identify the open source components that are sensible in Rahul study. We can further enhance that to apply this model there should be a modification of particular procedures, quality assessment model and open source components to gain better and qualitative results. [3]

3.1.1.2 McCall's Model

McCall present the earliest software quality model. This model introduced factors that are necessary for the qualitative software development. Given below is the way by which the software qualities could be increase,

1. According to the McCall, first of all find that factors which are barrier in the software quality work.
2. There should be criteria for judging these factors, which are hurdles for qualitative software development.
3. Every criteria should have explained a metrics.
4. Metrics should be estimated.
5. Metrics should be correlate with that type of the instructions that are act by the development team.
6. Set of metrics should have a develop recommendations.

There are many criteria that are described by the McCall and theses criteria put the fact upon the Qualitative Software work. The criteria that are described by the McCall are storage efficiency, error tolerance, traceability, expandability and simplicity. [4]

3.1.2 Quality of Open Source Software by Best Measurement

To estimate the cost, schedules and other detail regarding the software can be gain by the help of “Decision Support System”, in the beginning stages of the software development. But we can not evaluate the quality of the open source software by such tools. So, Software developer should declare the demanded quality in the beginning stages of the development. It is also important to gain the quality in open source software to bringing batterness in the measurements. [4]

Halstead that is known famous for the work of software Physics, he has suggested many methods to measure the open source software. He calculated many metrics that are helpful for the useful measurement of software. We can say these metrics are not only helpful for software measurement but also supportive for the idea of necessities of the future programming work. [5]

For bringing the quality in software, the size of it is in front of our eyes and we can bring the reduction in it by reducing the extra line of code and function points of the software. [4]

Checking quality in software, counting the source code lines was the primary method until 1979, when Albert belonging to IBM designs the FP metric. [6]

This FPs determines the size of a software system through its constituent components, input, out put, inquiries and its file. [7]

In short, there should be no extra lines of code in the project because these are also put the effect on the quality of the software. It is also responsibility of the coder to try doing his work in short lines of code instead of extra lines. Furthermore there is another metric that is known by the name of Complexity. [8]

We can measure it with McCabe s Cyclomatic Complexity. Its study tells that it is mathematical technique and has use to show the constitute modules of project and more, it is hard to maintain and test them. [9]

In this part we have discussed the issues of the measurements of the software projects. And have done our best to present the different ways in the light of previous work that how we can improve the quality of the software. [4]

3.1.2.1 Metrics Choice

To control the quality of the software it is not essential to use more and more metrics. Quality control team should use those types of metrics that having relevant and useful information regarding the process and project. Those types of metrics can put bad effect on the quality procedures of the organization that are developed arbitrarily. [4]

3.1.3 Quality of Open Source Software by Best Process

3.1.3.1 Total Quality Management

This is a concept that has been adopted by a lot of companies, those desires to get long term success for their organizations. W. E. Deming is famous as a quality advocate, has present some principles to bring the quality in the open source software, that are given below, [4]

1. In his first principle he describe it is the responsibility of the management to bring the quality.
2. For gaining the quality there should not be a compromise on expense.
3. If there are faults in a system, they are due to worker or system owns self.
4. During the development of software process it must be manage such way that quality must be built in the system.
5. Subcontractors & vendors must be select on their capabilities, but not on sole basis of cost.

Now days, many novel techniques are present to maintain the “total quality management” in every stage of the open source development. [10]

“Statistical process control” is more famous and workable technique that is using broadly for qualitative work. [11]

3.1.3.2 ISO 9000 Standard

International Standards Organization (ISO) developed ISO 9000 standard to keep maintain the quality of software during the software development process. Fifty nations of the world have implemented to these standards because European community has acceptance it. The manufacture of world that want to cover the market of Europe, Canada and USA, has to adopt this standard. This standard also has brought more quality in many products; software quality is also part of it.

ISO standard is applicable in any type of the environment because it is generic models & not specific to any kind of business. This model also gives the stress on documentation of each process because it ensure to huge degree of control, V&V, process improvement and audit ability. [12]

ISO certification is granted to those organizations that fulfill its quality system and this system is checked by the third party that is accredited registered. There are five sections to ISO 9000 standard and organization decides with respect to his business that to whom has to accept. International Standard Organization described the twenty standard elements. They mainly deal to these portion of the organization like design control, quality record, test equipment, internal quality audits, inspection, measuring, management responsibility and test status. More standard elements are directly related to the software development project. [4]

3.1.3.3 Personal Software Process Model

Humphrey has given the light on Personal Software process Model. He judged that for an organization to move beyond CMM level 3, every staff & team persons should apply the software process perfection on own level. Personal Software process Model (PSP) gives us 4 steps for accomplishing this. There description is given below, [13]

PSP1 describes that it is a responsibility of the Software Engineer to know their development time, rate of defect removal and defect creation. PSP2 enhance the above step that these data are used to guess size & growth time of the fresh programs. PSP3 step give the pressure on this thing that there should be a qualitative work on each stage of the software development and there should also be a checklist for implementation & design analysis. At last PSP4 step end this model by saying that supply the instruction on constantly using the above said 3 steps to permission individuals to scale up to making of the modules several KLOC long. [13]

3.1.4 Quality of Open Source Software by Best Tools

3.1.4.1 CASE Tools

Software development group is using the Computer Aided software Engineering Tools (CASE) on large scale to make easy improved requirements, configuration & process management, project verification and system validation. [4]

There are many CASE tools in the market that are using to maintain the quality of the software and for the purpose of testing them. SQA Enterprise Test Suit from S/w quality Automation Inc. tool is also using for the purpose of testing & debugging the software that has developed by the project team, enhance this tool also working for the testing of Windows server and server applications. Latest tool that is launch by the Microsoft corporations is known by the name of Visual Test. It is also using for testing and debugging of the software. [14]

SQLB is a very famous tool that performs his duties for test design feasibility, evaluation of results and for creating benchmark test. [15]

3.1.4.2 SE Environments

For providing the life cycle support there are many software development environments. One of them that is famous and well know is “software through picture”. This environment provides the testing capabilities. [7]

Any body can not refuse the importance of the CASE tools but now many organizations are using to these tools at limited extent. We can say that learning curve is doing significant job in acceptance & adoption of such tools. A number of S/w engineering environments increasing to “life cycle support” pose problem are owing to the enormous quantity of learning required for effective utilization. [16]

3.1.4.3 Using of Refactoring

By the usage of the refactoring technique, we can improve the quality of the open source software and can get more advantages from it. This technique is very important and generally uses to make the OSS more beneficent and could be measured by using many metrics. [17]

Refactoring can be explained by these words that “the process of changing a software system in such a way that it does not alter the external behavior of the code, yet improves its internal structure”. [18]

The main aim of the refactoring is to reduce the complexity of the open source software at different stages like design and source code level. Its also use to reduce the duplication of the code. It is the main problem

for Software Engineering to find that where to use the refactoring technique. So software metrics are helpful to determine the areas where have to apply refactoring technique. [19]

Software metrics are using on large scale in software quality assesses method. In the reply of its method, it is helpful to determine that whose part of the software project has to reengineer. Reengineering the effected parts of the software project is done by using the refactoring technique. There are many tools available in the market that have made it easier to gather the software metrics and automatically doing the process of refactoring. [20]

Bespoke product may be developed for a single customer or for a whole company or organization. Bespoke software is such a customized software in which a particular customer or company gives his requirements according to their needs [38].

According to my view the definition of bespoke is a Product on demand. Bespoke product is produced on the basis of single customer's needs and wants. Development process for such a product is differs considerably from a mass product as it is produced on the basis of customization. [39].

According to my point of view and experience in case of bespoke product quality can be achieved by collecting the customer needs and wants precisely and then fulfilling the customer's those needs and wants completely. Customers needs and wants can be collected by conducting open ended interviews, discussions, and through questionnaires.

Some its difficult for customer to explain what exactly he wants so in that case prototyping can be effective way of collecting customer needs and wants, so a prototype is developed to give an idea to the customer that how the product would be like and then customer suggest changes in that product.

A quality product is can only be developed by meeting the customers requirements precisely and delivering the product on time with effective cost. [40]

3.1.5 Quality and Quality Work in Market

Market driven product is developed while keeping in view the needs and wants of a large number of people who have different taste. So such product has different segments because of the different taste of the people. So each segment satisfies a particular group of people. [41]

As the market driven product focuses on the needs and wants of mass of people so these needs and wants are categorized and then implement at development process for each group or segment of people and value is achieved.

Quality in market driven product can be achieved by implementing a comprehensive product development and quality control strategy. This can be achieved by determining the control checks and strict measurement of customer needs and wants implementation. [42].

According to my point of view quality of bespoke product highly involve communication with the target customer as it's only based on one customer so it's important to elicitate the requirements from that

customer and keep in touch with customer in the whole development stages with customer, so that his requirements could be achieved completely.

Where as in market driven products as the target customer is mass people so their requirements should be gathered by conducting surveys and there are different for conducting surveys i.e. interviews from selected people from different groups or by questionnaires and then through implementation of those requirements quality can be achieved.

3.1.6 Quality And Quality Work In Different Open Source Software Systems

Quality is much dependant upon the type of system because the quality measurement aspects are changed.

3.1.6.1 Real Time Systems

Real time systems are those which are able to give response in a specific time for any input. They refer to different events which are performed by computer virtually, as these events are taking place in real life. For example, a real time system shows the events through graphics with same speed as they happen in real life [43].

Quality of real time systems is judged under following conditions [44]:

It should meet the time boundaries against inputs. Concurrent processing should be done for executing all tasks. System should react to all events for which it is developed. Examples of real time systems are weather forecasting systems, smoke alarms, fire alarms etc.

The most important quality requirements of the real time systems are reliability and performance. Where performance is of more importance, it is the time taken by the system for giving feedback.

3.1.6.2 Operating systems

It is an application which provides us platform to do different type of tasks which are available in it. It contains different applications. The different aspects of operating systems related to quality are reliability, performance, flexibility etc. Some of the operating systems are Windows, LINUX and UNIX.

The quality requirement, reliability of the operating system can be achieved if the device driver failure doesn't occur; performance can be achieved if no errors are generated during the execution of different programs; reliability is achieved if the operating system is able to recover from failures and errors [45].

In operating systems the performance factor is given less priority to other dimensions in comparison to real time systems. We can bear little delay in execution of any command in operating systems but not in real time systems. For example the command execution in real time systems is 1 second so if it takes 2 seconds in operating system it is bearable.

3.1.6.3 Web Systems

Web systems are most common systems in the current era. They are used to get information about everything. Examples of web systems are online reservation systems, forums etc. Different quality requirements of web systems are usability, reliability and availability etc.

These systems are used to retrieve fast and reliable data over the internet. The reliability of the web page is much influenced by the web browser, its version and operating system being used by the end user [46].

The quality requirements of the web systems are achieved by giving the user the easy interface from which the user is familiar, the information is easy accessible with less clicks i.e. usability. Web page is easily readable to the user.

3.1.6.4 Differences between Three Systems

There are different quality requirements in each system. The quality requirements can be achieved by having knowledge of the domain. If one has better domain knowledge, achieving the quality is not a difficult task. Different quality attributes in the above mentioned systems are achieved differently i.e. in real time systems the accurate information is required without time delay as comparison to operating systems. While in web systems the usability is the main concern which is achieved by giving user an interface which is easy to use, easy to navigate, access to the information in less clicks.

3.1.7 Quality Dimensions of Open Source Software

In the area of open source software quality we find many theoretical enhancements but there is no established definition of quality. The quality of open source software can be improved if there is some standard or refined definition.

Open source software quality is defined by different thoughts. It can be known by practicing the content analysis or the various thoughts and views. These thoughts can be either from customer or producers perspective. Different quality dimensions of open source software are as follows [47]:

- Accuracy
- Capability
- Conformance
- Features
- Flexibility
- Maintainability
- Simplicity
- Stability

Through analysis it is found that some major quality dimensions of open source software that fulfill customer requirements are usability and capability of the open source software products. Users of the open source software are more interested in the functioning of it that what the open source software do and how much it is useful to them. In producers perspective the quality dimension is the growth in the future, that is how much business the product will bring in future [47].

3.1.8 Similarities in Software and Goods

In terms of reliability the software is supposed to be bug free during its execution. It should not cause failure. If the software does not crash in run time and function properly under some conditions then it is reliable [48].

For goods the reliability is measure of how frequent a problem occurs and what type of problem is it. Reliability doesn't matter if there are problems and lacking in the product [37].

Performance of open source software is the response time of open source against any query of the user. It can be time in seconds e.g. 2 seconds to complete certain task of the user.

For goods the performance is important to customers in market perspective, the market segments like speed of product, size of the product etc [37].

Maintainability of open source software is achieved through flexible architecture of system. It is how much a system is able to be modified, how much modifications can be made in its components. Open source software is enhanced to add additional functionality or remove the faults.

For goods the maintainability is achieved through factors that how it is easy implement a strategy for taking care of the problem.

3.1.9 Differences in Software and Products

The major difference in open source software and product is the environmental aspects. Environmental aspects have affected on the product but in case of open source software the environmental aspects don't have any affect.

Another difference in open source software and product is the appearance. In open source software the appearance can be changed once the product is launched in the market. In case of goods they come in to market after final appearance, there is no improvement once they are launched in the market place.

3.1.10 Similarities in Software and Services

One of the similarities in open source software and services is reliability. Services are reliable if they are delivered properly without having any deficiency. Similarly open source software is reliable if they don't have any bugs and are error free.

Another similarity in open source software and services is the access. In services major focus is availability to the supplier. If supplier access is available then one can get service. In terms of open source software the access is the access of records from the database or memory drive.

Responsiveness is also one of the factors that are common in open source software and services. In terms of service the responsiveness is when the customer wants a service and it is delivered by the supplier [37]. In open source software when a user does certain action or query in return the open source software gives desired output.

3.1.11 Differences in Software and Services

Courtesy is one of the factors that are different open source software and services. In open source software there is no concept of courtesy but in services it refers to politeness of the service.

In services there is a term empathy which is related to the situation of the customer [37] but in open source software there is no such concept of empathy. In services there is a term communication, which relates to kind of service a customer wants. In open source software the customer wants/needs are achieved through quality tools.

These are the examples of seven tools of Quality:

3.1.11.1 Collection of Data

Data collection discussed by Klefsjö is about the information of defects or problems which can be occurred in a product [37].

During our experience, organization often makes a check sheet for listing the errors. The pattern of the check sheet contains description of the bug, responsible person, module definition of the bug, and responsible person who is responsible to fix that error in the open source software and there is also status column which shows the bug status. Check sheet is very simple which can be understandable to every individual as shown in Table.

Bug Description	Responsible Person	Module Name	Status
X	X	X	X
X	X	X	X

3.1.11.2 Histogram

In our organization we maintain the histogram graph on the basis of the weekly arrival of the bugs. We had a very nice designed mechanism to record the arrival of bugs on weekly basis. We have plotted frequency of error along Y-axis with interval of 5 and weeks along X-axis. This graph gives us very much help to our team to minimize the number of bugs so that quality in the open source software maintains. This Table shows the frequency of arrival of the bugs during the development of the open source software every week.

Weeks	Frequency of errors
Week 1	35
Week 2	43
Week 3	51
Week 4	79
Week 5	38

Week 6	49
Week 7	53
...	
Week 52	45

3.1.11.3 Pareto Chart

From our experience to improve the quality of open source software product we have to identify the root cause of that problem due to which quality of open source software product suffers. This tool is very helpful for our organization. During open source software development many errors are popup in our organization so we list all those errors (but few of them are listed below) and make Pareto chart. The table contains the information about the cause, occurrences of defectives and percentage of those defects in the project. The Table PC shows the number and percentage of defectives of a cause.

Cause	Number of defectives	Percentage of defectives
Syntax error	73	45
Linking with DB	35	21
Logical error	23	18
Other	18	11
Sum	164	100

3.1.11.4 Cause-and-Effect Diagrams

The Cause-and-Effect diagram provides an efficient summary of factors that impact a process, and hence can be used as a map to guide the overall quality improvement efforts [49]. The Cause-and-Effect is similar to fish bone diagram.

From our experience, cause and effect diagram can be used in open source software development to identify the risks in the project. Risks which can be occurred during software Development Life Cycle be shown through cause and effect diagram.

If we take an example of technical resources run out, there is risk factor of resource run out which can be effect the delay of the open source software delivery. There are many causes of technical resources run

out like:

- Better opportunity
- Not a favorable environment
- Management conflicts and
- Person working on the disputed technology

Disputed technology means when a person is interested in doing work in his particular skills but he is working in some other technology.

3.1.11.5 Stratification

We use the stratification when the values deduced for the histogram shows some difference. So we make the histogram chart and on basis of that we make the stratification result [37].

3.1.11.6 Scatter Plots

The scatter plot shows the relationship between two variables by displaying the data points on a two dimensional graph. In our organization we use scatter plot, when we want to calculate the number of hours a person consumes in completing its allocated phase. So by the help of scatter plot the project manager will check the person's ability and get an idea that a particular person has or has not completed the phase on the estimated time.

Phase	Time (Hour/s)
1	13
2	7
4	18
5	10
3	16

3.1.11.7 Control chart

It is used when we have to define upper and lower limits and on basis of upper and lower limit we establish an average limit. The upper and lower limit tells that the control is going out of the average limit. The average limit is defined by the organization that the working should be done on this limit if goes upper or lower to this limit then there is any problem occurs which have to resolve [37].

In our organization we have set an average limit for the development of the project. The average limit is the completion time of the project. If the project completion time crosses the upper or lower limit then the

problem arises. The project manager has to look after that problem due to which the average limit is not met and resolve it in order to capture the average limit. The control chart becomes a great helping tool for our organization which makes the projects to be completed usually on time.

3.1.12 Quality Aspects of Supply Management

Different aspects of supply management can be related to the open source software development. In supply process when a company wants to manufacture a product then it might be the case that company will manufacture some parts related to the product. And also there will be some pre-build components that can be used in the development. Same thing can be done with open source software development when we have pre-build components then we can reuse them in our product to reduce the time, less cost and on time. These pre-build components can be tested by the supplier or company's fame and should be bug free.

During our experience, company often uses these third party products or components to save the time effort, cost and less utilization of man power. Open source software engineering's famous saying is doing reinvent the wheel means if you have some components then just use them in your product.

3.1.12.1 Affect of Supply Process

Quality in the supply process can affect the end-product a lot. If you have good supply process then end product will have quality. There are some aspects which can affect the quality of the end product.

3.1.12.2 Inspection to Achieve Quality

In the supply management there is focus on the extensive inspections to measure the problems in the product. Supply management focuses on the inspection to achieve the quality in the product and which results in a better product free of problems and deficiencies. Process of inspecting the product or project is very much effective in the open source software development to deliver the bug free and reliable software. Inspection in the open source software development is done in the testing phase of the development.

3.1.12.3 Customer Involvement

Customer involvement can affect the quality of open source software. For example customer involvement in the design activities to give handsome suggestions to finalized the design activities. Customer involvement in every phase can affect the quality of that phase or even the whole open source software in terms of user based requirements.

3.1.12.4 Continuous Improvement

Supply management focuses on the continuous improvement. The continuous improvement might be in the development process of the product, gaining customers. The continuous improvement in software development is achieved by Total Quality Management (TQM) [37].

TQM can be used to enhance the performance by means of improving quality in the products and processes. Total Quality Management provides the framework which is relevant to the customer needs and those needs will be met by the developers. TQM focus on some important aspects like customer focus, attention to process, continuous improvement, measurement and analysis of data and human factors in management and leadership

3.1.12.5 Continuous Improvement Through TQM

Improvement can be done for a short-term or long-term but when you have ongoing continuity in the improvement then that improvement is a long term improvement. Continuous improvement can be done through the constant cycle some activities like planning, doing, checking and acting. Scientific method related to improvement process has much relevance to the open source software development for continuous improvement. Continuous improvement in open source software development not only the improvement in the technical aspects but it can be in all phases, the quality can be seen in work, service, information, process etc. TQM provides a small but steady improvement framework to get the quality in all aspects of the organization [50].

3.1.12.6 Rational for Supplier Products

Supplier can be evaluated easily if the supplier is experienced and has got some development experience in the relevant supply. But it is hard to evaluate the newly supply who has no experience in development of such products.

Supplier products can be evaluated on experience of the supplier. How long supplier has been developing those products. What type of standards supplier is using for the development of the product and what are achievements in getting any of the standards for maturity CMM level or ISO 9000 level. What is development process for product and the product should be bug free and reliable.

3.1.13 Phases of Quality Vs Open Source Software Quality Activities

The four phases of quality movement as discussed in Klefsjö are given below [37]:

- Quality inspection
- Quality control
- Quality assurance
- Quality management

3.1.13.1 Quality Inspection

This phase of the quality movement is done when open source software is developed. In this phase usually optimistic approach is followed. Quality inspection is based on the post testing phase which is a part of open source software quality activity. Usually the post testing phase is started when the development of the open source software is finished. During the testing of the open source software if any bug or error occurs then it will be tackled instantly before marketing of the malfunctioning open source software. But now a day it is basically done in the quality control phase by focusing on pessimistic approach rather than optimistic approach [37].

Earlier in our organization we were focusing on optimistic approach and when the open source software development team is developing the open source software so they assume that everything during the development is going fine and when the open source software development is finished so there is testing team who will test the quality of the open source software and report to the developer for the flaws and error which were popup at the time of testing.

3.1.13.2 Quality Control

Quality control phase is done during the development of the open source software. In this phase usually pessimistic approach is followed because the testing of the open source software is going simultaneously at the same time when the development of the open source software is running on. This phase is more used now a days then the quality inspection phase. During the development of the open source software continuous testing and designing of the open source software being done until it meets the entire requirement which will enhance the quality of the open source software. The testing of the open source software is mandatory because if we want quality in our open source software so it should test at the same time when the open source software is in designing phase, because any bugs or errors occur during the open source software development so it will be amend instantly at that time rather it will popup during the inspection. Our organization focuses on pessimistic approach so that quality in the open source software will retain by doing testing and designing at the same time during the development of the open source software rather when open source software developed. In case any defect occurs at the time of development we amend it instantly rather taking it in future and then amend it.

3.1.13.3 Quality Assurance

Quality assurance in open source software is used as a planned approach to evaluate the quality of the open source software including processes, procedures, and functions. Quality assurance in open source software has the responsibility to insure that processes and standards which are used and established are followed through SDLC (open source software development life cycle). Processes which are used for development and controlling have quality assurance support facts; with the help of these evaluations, open source software quality assurance can be done by the means of applied standards. Open source software quality assurance has activities such as process monitoring, evaluation and auditing. Auditing is used to evaluate the product and process. The benefit of auditing is to review management, technical and different processes to provide the guidance about the quality of the software product [51].

3.1.13.4 Quality Management

The earlier three phases of quality movement which are quality inspection, quality control and quality assurance are included in this phase. All the inspection, controlling and assurance of the software products are going on under the quality management phase [37].

The main focus of quality management in open source software is to certify for the achievement of desired level of quality in the product. Quality management involves in defining standards and processes to achieve a certain level of quality and ensures that these processes and standards are also followed properly. Quality management also focuses on creation of such environment or culture where quality achievement becomes everyone's responsibility [52].

3.2 Question No 2 & 3

How can we assess open source software quality with respect to various aspects of quality based on existing literature and criteria for evaluating the open source software quality?

In Software Quality Assurance we check the software Engineering methods & processes so that we can ensure the quality of the software. Software Engineering includes the tools and methods for Software requirements, testing and maintenance. Software Quality Assurance encompasses the all processes of the software development like design, coding of the programmed, and control of the source code, review of the code, change, configuration and release the management. We can say that quality assurance mean the control of software development process. [21]

ISO 9126 Model provides a structure for the evaluation of software quality. It has six software quality attributes. Furthermore, these attributes also called quality characteristics. Now, I will explain to this model in the aspect of quality assurance of the OSS. [21]

3.2.1 ISO 9126 Quality Model (Recent Model)

We know that there are many models that have tried to assure the quality of open source software. Now, I will like to explain the ISO 9126 model because it is summery of all previous models. The aim of it was to together different views that are dealing about OSS quality and its assurance in a single place. This model works like a hierarchical model and consist of six attributes that are Functionality, Usability, Portability, Reliability, Efficiency and Maintainability. [21]

Software projects become assess on the basis of Characteristics that have been described by the ISO 9126 Model. Given below figure is showing the decomposition of these characteristics at low down level and these are also related to certification & evaluation actions. Given below is the explanation of ISO Model s attributes and their sub characteristics. [22]

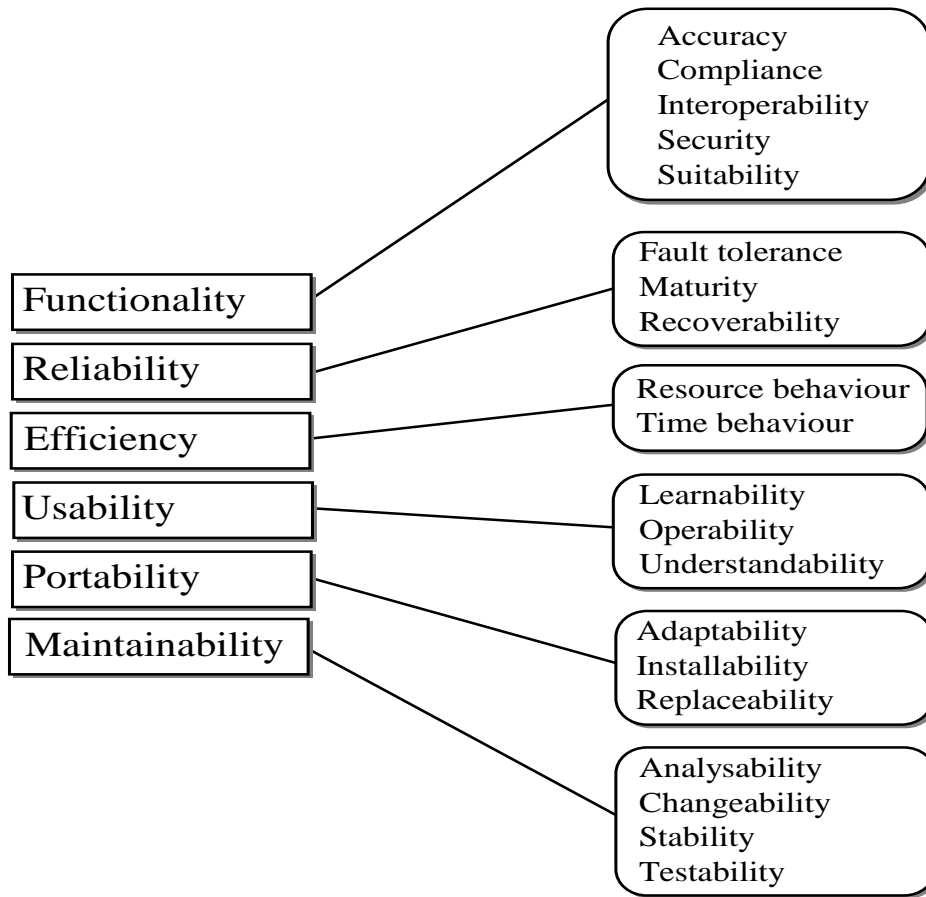


Figure: ISO 9126 Software Quality Model [22]

3.2.1.1 Functionality

Functionality mean the numbers of functions must be available in software that fulfills the minimum usage criteria of the user. [23]

ISO 9126 Model describe the Functionality attribute with these words that “A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs”. [24]

Model also put the light on this attribute with theses words that “This set of attributes characterizes what the software does to fulfill needs, whereas the other sets mainly characterizes when and how it does so”. [24]

Functionality can be defined as; it is such type capability of software product that gives those types of functions that fulfill the requirements of the user. It is fundamental characteristics of the software development and it is close to the property of the correctness. [25]

Now, I would like to explain sub characteristics of Functionality feature and their contribution in the quality assurance in software project. [26]

Suitability

It is ability of software to offer suitable group of functions for particular jobs & user objectives.

Accuracy

It is ability of software to offer that type of results, which are to the point or demanded by the user.

Interoperability

It is such type capability of the software that interrelates with one or more than one particular systems.

Security

We can say that it is such type of feature that prevents the software in respect of security.

In short we can say functionality is that type of property of the software that tries to present all types of functions that are expected by the user and software developer in the line of functionality.

3.2.1.2 Usability

Model (ISO 9126) describe the usability attribute with these words “A set of attributes that bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users”. [24]

Furthermore, it can be explained with these sentences that a user give a data to a machine in the form of input and machine acts on it to give useful reply and user can understand it easily for demanded work. [21]

Usability can also define with these words that it is a measure of struggle, demanded to learn & use software project for aim that has liked. The aim of this feature is to assure that is software proper for the demanded purpose or not. [27]

Explanation of sub characteristics of the Usability feature and their contribute in the quality assurance of open Source Software development, is given below, [26]

Learn ability

It is ability of the software to facilitate the user to become skilled at its applications.

Operability

It is such type of capability that does enable to the user to operate & manage it.

Attractiveness

It is such type of property of the software to whom user liked very much.

Understandability

This capability make the user in this condition in which he can know, whether software is suitable & how it may be used for a special job & condition of use.

In short, these sub characteristics are paying useful role to bring the quality assurance in the open source software development.

3.2.1.3 Portability

ISO 9126 Model perform the definition of the Portability attribute with these words “A set of attributes that bear on the ability of software to be transferred from one environment to another (including the organizational, hardware, or software environment)”. [24]

Portability is also a main issue of today and with respect to it, Open Source Software could run and give better results on different platforms. [21]

Light on the definition of Portability may be put with these words that it is such type property of the software project due to which software can be transfer from one environment to other. For example software could run in different types of operating systems with no trouble.

Sub characteristics of Portability that are defined by the ISO Model are given below, [26]

Install ability

Property of the software project to be installed in a particular environment.

Replace ability

Such kind of ability due to which software could be used in place of other software in the environment of that software.

Co-existence

That property which use to co-exist with software in a general environment sharing the common resources.

3.2.1.4 Reliability

International Standards Organization 9126 Model put the light on Reliability with these sentences “A set of attributes that bear on the capability of software to maintain its performance level under stated conditions for a stated period of time”. [24]

Further, Sub characteristics of Reliability attribute are given below, [26]

Maturity

Due to this property software could avoid from failure in the result of fault in the software.

Fault to Learn

This capability sustains a specific level of performance in situation of software fault.

Recoverability

This capability could recover the data if it loose in any condition during the execution of the software.

3.2.1.5 Efficiency

According to the ISO Model Efficiency is “A set of attributes that bear on the relationship between the software's performance and the amount of resources used under stated conditions”. [24]

Efficiency describes that the response of the software should be faster in the form of any input.

Now I will discuss the sub characteristics of efficiency, which assure the quality of the Open source software. [26]

Time Behavior

Under this capability of the software the processing time become faster and reply appear so urgent without doing the wastage of time. [26]

Resource Utilization:

Ability to use suitable resources in proper time when software performs its job under stated environment. [26]

3.2.1.6 Maintainability

According to ISO 9126 Model it is “A set of attributes that bear on the effort needed to make specified modifications (which may include corrections, improvements, or adoptions of software to environmental changes and changes in the requirements and functional specifications)”. [24]

Maintainability helps to detect, remove & isolate the defects of the OSS and it is also main expense reason in software project development. Further, Face to face contacts do not occur in the maintence of Open source Software. So we can say in the aspects of quality assurance it is problem for it. [23]

Under are the sub characteristics of the maintainability that are also paying important role for bringing the high quality assurance in the OSS. [26]

Changeability

This ability of the software makes it assure that a particular alteration to be implemented. [26]

Stability

It tries his best to reduce out of the blue effects that have produced owing to chaining of the software.[26]

Testability

This capability make is confirm that changes/modified of software is validated. [26]

Analyzability

This sub characteristic identifies the reasons that have produced in the result of the software failure or for those divisions that to be change to be diagnosed. [26]

Furthermore according to this model, Dromey describes the software quality model in these words that “we cannot build high-level quality attributes like reliability or maintainability into software. What we can do is identify and build in a consistent, harmonious, and complete set of product properties (such as modules without side effects).” [28]

It was a problem to keep quality of the software. At last, he purposed to use four types of categories that have properties for the qualitative work of the software development. These properties are given below, [4]

1. Correctness Properties.
2. Internal Properties.
3. Contextual Properties
4. Descriptive Properties.

Here is a process that have five steps that has used to built a model and it has depend upon the following properties, [4]

1. There should be identifying of that attributes which have high level quality. For example maintainability and reliability.
2. Products components should be known at appropriate level of detail. For example subroutine and package.
3. Important properties of every component should be known and categorize.
4. Suggest a relation that shows the connection between quality carrying properties and quality attribute.
5. Iterate the steps that are defined above with the help of refinement & evaluation process.

3.2.2 ISO 9001

It illustrates the requirements for Quality Management System that is portion of total manufacturing process. It is very common standard and all kinds of organizations could adopt it according to their demand and size. Standard certificate is accepted in the word & organizations look it with honorable eyes. It is difficult to apply the suggestions of standard in the software process with full activities. For resource management process this standard has many suggestions. During the development process like analytical specification, documents design, plans of testing and procedure, manual of user & code of resulting, we like to assure the quality of our products. Supplementary, ISO 9001 does not describe any specific model of quality. But it use easy way to assess the quality of the product that fulfills the customer pleasure. No quality characteristics are described according to this approach and the quality assessment is base on the customer demand. If demands of the customer fulfill than product has assessed himself at upper level. We can say that problems to gaining the quality in the products have produced this standard orthogonal to ISO 9126.No familiar spot b/w two, but we can also not find the opposition between them. ISO 9001 demanded that the quality management system should be maintained, implement and developed. [29]

Given below portion of the Standard directly describes to the point that are regarding software project. Minimum requirements those are demanded and required by the standard to bring the quality in software projects are given below, [27], [29]

1. Make sure the requirements of the customers' product. A customer wants, his demanded product must have the particular properties, those properties and standards must be available in that product. We can say the satisfaction of the customer is the assurance of the quality.
2. The requirements that are demanded by the customers there should be review of them, the record of reviews must be maintained and changing that is coming in the product should be control.
3. Software process must be developed and the responsibilities of each stage must be allocated to the related person of the team and also describe the input & out put data of each stage of the software development.
4. V&V (verification and validation) processes must be done to sustain the record of these activities and also handle the design & growth modification.

The above said points are helpful for evaluation the quality of the software projects processes. We can say these are very important statements that assist to provide the quality assurance of the software.

3.2.3 Quality Evaluation Method

Quality evaluation that is described by the customers and that is defined through the software manufactured, between them there is a huge variation. Manufacture describes those types of metrics that are related to the quality features, measure & gather previous data from a group related projects. The whole procedure may be helpful by computerized tools one explained. [30]

Like this approach is tries to assess the quality of the software project/product. And this approach could be applied with the help of Goal Question Metric method. NASA has invented this approach. We can say that quality characteristics may be alike to one described in International Standards Organization 9126 model. [31]

Customer could not pursue this type of approach and historical record data become more of the characteristics ineffective. [27]

3.2.4 Process Centric Approach

The main benefit of this approach is this to provide the auditor a chance to make recommendations on what to do better in development process of the open source software project. Method that was build up, its base was on that group of criteria which characterized the open source software worth. This method was like Goal Question Metric method in that assessment of a specific criterion depends up on a procedure in which questions are asked and their answers are given. To design an assessment process, we divide problem space in 6 subject areas. [27]

1st subject area is corresponded to progress itself. Further, four subject areas except last one are corresponded to specific working inside the process of the open source software development project. And the last subject area works regarding s/w documentation. So we can say the given below are areas were described, [27]

- A. S/w Process & development Methods.
- B. Analysis & its Products.
- C. Design & its Products.
- D. Implementation & Code.
- E. Testing Process & its Documentation.
- F. User Manual.

From B to E subject areas cover that type of working that is not found in waterfall model and incremental model. Every area is wrapping with group of criteria and is organized with the path of traceability: from requirement to verification & design and from design to implementation. More answers are related to qualitative and its calculation also used only in that types of situation where numbers are significant for clients.

3.2.5 Software Process and Development Method

Major aim of work that has done in this subject area are the discovery of methods & standards that were used in the growth process of the open source software development process and is the verification of use of that types of methods, due to which the traceability, readability and completeness of the resulting product. To gain these aims given below, we will describe a group of criteria with suitable questions, [27]

3.2.5.1 Standards & Methods

In Open source software development process whose method & standards have used?

And what can you say about the benefits of the methods?

3.2.5.2 Process Completeness

Which artifacts refer by methods has been produced?

Can you say group of artifacts is enough or not?

The results which have appeared in the form of out put that have documented correctly or not?

3.2.5.3 Modifiability & Readability

Documents that have produced are they in situation of modifiability & readability or not?

3.2.5.4 Traceability

The documents that are produced during the open source software development that are traceable or not?

Above list and questions show that assessment inside this subject area was mainly describing the general aspects of the open source software development process. But it does not describe the analysis of ideas of those types of documents that are crated. Furthermore, we can say that above said criteria are keep in touch with the standards of International Standards Organization 9001. [27]

3.2.6 Analysis

Major aim of work that has done in this subject area are the specification of methods & models that have used in the process of analysis & verification and the use of that types of methods due to which the completeness, correctness & verifiability have produced of the obtained documents. To gain these aims given below, we will describe a group of criteria with suitable questions, [27]

3.2.6.1 Sources Completeness

For analysis purpose whose types of resources have used?

Choice of authorized regulations was complete or not?

3.2.6.2 Analysis Model s Consistency

Analysis model that has developed has been specified in criterion one or not?

3.2.6.3 Analysis Model s Completeness

Analysis model that has developed fulfills to the entire legal regulations or not?

Is the list of possible situations is complete or not?

3.2.6.4 Context Definition s Completeness

Is group of out put facts enough to gain the business aim of system or not?

Is group of out put facts complete according to the business & legal demand?

3.2.6.5 Functionality Completeness

Are functional necessities entire and sustain by suitable function of the software?

3.2.6.6 User Interface Prototype s Usability (UIP)

Prototype is complete for his functionality or not?

3.2.6.7 Data Model s Correctness

Data model is reliable & complete?

Data model is reliable for procedures?

Data model obeys the engineering practices or not?

3.2.6.8 Non Functional Requirements Completeness

Non functional requirements are complete or not?

3.2.6.9 Non Functional Requirements Verifiability

Non functional requirements that are described in a procedure are testable or not?

3.2.6.10 Verification Credibility

Performance testing is enough for useful procedure?

What opinion about the credibility of testing process?

Above stated list and questions explain that assessment inside this subject area was mainly describing the contents of analytical products. Trend of above questions goes from source of information to business model and from here towards functions & efficiency and then moves from this side towards the process of verification. Evaluation results have first-class relation with the given below characteristics of International Standards Organization 9126. [27]

In the light of above criteria of list ,functionality takes the favor from the point no.2,3,4,5,7 and efficiency by the point no 8,9.Usability has taken the supported from point 6, reliability from

point no. 8,9,10 and maintainability turns the trend of kindness toward himself by the point no.7. [27]

Here are different types of models by the help of which we can access the quality of different open source software. Capability Maturity Model Integration (CMMI) and International Organization for Standardization (ISO) are the two very common quality systems in open source software industry. This document contains the major characteristics of CMMI and ISO, how they work, why they are developed and what are the differences among them. There are some pros and cons of each model especially in terms of quality systems which are also focused in this document. I discussed three major process areas of CMMI along with their benefit and at the end there is general discussion about the quality systems.

3.2.7 CMMI

CMMI is an approach of process improvement which gives essential elements of effective processes to organizations. Throughout the project it leads to improve process. It relates different organizational functions; set priorities and goals of process; improves quality of processes; and gives point of reference for judging the on going processes [53].

ISO is derived from Greek word isos which means “equal”. According to ISO, it is the world’s largest developer and publisher of international standards. It’s a network, based on different organizations of 157 countries [54].

3.2.7.1 Characteristics of CMMI / ISO

CMMI is a model that improves the capability maturity of open source software development processes. It was developed by Open source software Engineering Institute (SEI) of U.S. Carnegie Mellon University. An organizations maturity is evaluated and accessed at five levels through it. It is an indicator that the international standards as well as standards in United States are recognized in an organization. [48]

CMMI is a flexible framework for many disciplines [55]. It is best for process improvement [37]. It assists the use of different models. It increases the understanding and abolishes variations among models. Organizations can use it according to their business objectives. There is no CMMI certification but an organization can be appraised and awarded a rating of 1-5 level. If the appraised organization wants then the results of appraisal can be made public. The CMMI v1.2 consists of 22 process areas [56]:

- Causal Analysis and Resolution
- Configuration Management
- Decision Analysis and Resolution
- Integrated Project Management
- Measurement and Analysis

- Organizational Innovation and Deployment
- Organizational Process Definition
- Organizational Process Focus
- Organizational Process Performance
- Organizational Training
- Product Integration
- Project Monitoring and Control
- CMMI Project Planning
- Process and Product Quality Assurance
- Quantitative Project Management
- Requirements Development
- Requirements Management
- Risk Management
- Supplier Agreement Management
- Technical Solution
- Validation
- Verification

ISO is a quality management system. Its major product is International Standards, it also publish technical reports, specifications, corrigenda, guides and specifications available to public. [57]

International Standards are in format ISO [/IEC] [/ASTM][IS] nnnnn[:yyyy] Title. Here IEC is mentioned for the standards, which result from the work of Joint Technical Committee (JTC1). ASTM is for standards that are developed in coordination with ASTM (American Society for Testing and Materials) International. Is used if the standards are complete or published, nnnnn represents the number of standard, yyyy is the year when the standard is published and Title is identification of subject. Date is not mentioned if the standard is incomplete or not published. [57]

All the ISO documents are copyrighted so for copying ISO charges for most of the documents. Draft documents in electronic form are free of charge for copying, so one should take care in using them since they are not finalized as standard. ISO and U.S official representatives make some standards freely available to all.

The characteristics of ISO 9001:2000 are as follows [55]:

- Process oriented
- Customer satisfaction in focus
- Continuous improvement
- Business objectives in focus
- Leadership, commitment
- People involvement
- System approach – interactions between processes
- Decisions based on facts
- Supplier relationships

3.2.7.2 Difference between CMMI / ISO:

CMMI and ISO have some major differences. CMMI is referred as a standard more often, but it is due to the de facto standard. It is considered as a model for implementing systems and open source software engineering principles to development of product that can additionally be used for measuring progress of process improvement. ISO is used for extensive implementation in different type of industries but CMMI is specific to be applied in software engineering, systems engineering and software acquisition. [58]

One of the major differences in both standards is their language. ISO is prescriptive but CMMI don't include the requirements by the use of shall statements. Consider the example, ISO states, "The organization shall identify the processes needed for the QMS....," on the other hand corresponding CMMI specific practice states, "Establish and maintain the organization's set of standard processes".

One more difference is that the language of ISO is compact. ISO uses phrases "establish and maintain" or "determine and provide." Consider the example, ISO standard states "The organization shall determine and provide..." focuses on two different actions: the first one is to determine the requirement of resources and the second is the provision of those resources. On the other hand CMMI correspond this ISO requirement to project planning ("determine") and then states ensure that the resources are available ("provide").

ISO and CMMI differ in targets and intent, so the detail that each demonstrates varies. CMMI being a model contains important aspects for the development of complex systems. ISO gives guidelines important for development of products with high quality and customer satisfaction requirements. To get ISO certification all the requirements need to be satisfied.

ISO gives high level of guidelines for process improvement implementation but it doesn't give any detail for approaching this task, from where to take start, or how to maintain improvement when goals of process improvement are reached. While CMMI has its five levels of process maturity and six levels of process capability. They direct organization in step wise manner to achieve the goals.

CMMI differentiate between improvement of localized processes and overall organization processes. ISO don't provide any guidelines for requirements implementation in small organizations or in large organizations where many projects are running at a time.

PROS AND CONS OF CMMI / IS:

There are certain pros and cons of CMMI and ISO. Strengths of one in certain area make the other's weakness. Some foremost strengths of ISO are [58]:

- Wide applicability
- Affects the major functional areas of an organization
- Internationally recognized and appealing
- Liberty of implementation

ISO is applicable to any industry or environment. It affects the major functional areas of an organization like human resources, production, management, and quality. There is a need of collaboration between these functional areas for the implementation of customer requirements to achieve customer satisfaction. ISO is known world wide and is a mark of standard. The companies certified by ISO have much worth to the outside world. The organizations are open to include level of detail and type of documentation that best fits them.

3.2.7.3 Strengths of CMMI Include [58]

- Addition of institutionalization practices
- A way of improvement through maturity and capability levels
- Identification of project based and organization based processes
- Guides step wise which areas to address first and which to address later.

Along with strengths ISO has certain weaknesses which are [59]

- ISO is much general, it doesn't provide any guideline how to implement it in organizations which are smaller than an enterprise. It also doesn't guide for implementation in different industries

Weaknesses of CMMI are [59]

- It is too much detailed for certain organizations.
- Full implementation of it requires huge financial and other resources.

Important Process Areas of CMMI

Every process area of CMMI has its own significance. According to my point of view the three most important process areas of CMMI are:

- Project planning
- Process and product quality assurance
- Organizational process focus

The mentioned process areas of CMMI give the highest benefit to process improvement.

Project planning is a project management process area at maturity level 2. It establishes and maintains the plans which define project activities [64]. Due to project planning we have estimates of project scope, work product and task attributes. We have defined project lifecycle and know about the estimated cost

and effort would be. On the basis on this we develop project plans having budget and schedule. Risks are identified. We have plan for data management, project resources, stakeholder involvement and required knowledge and skills. Then we establish the project plan and after reviewing and reconciling work and resource levels obtain plan commitment. [60]

Process and product quality assurance is a support process area at maturity level 2. It is important because it ensures staff and management with objective insight into processes and associated work products [64]. It is useful to objectively evaluate the processes and work products. It provides objective insight. We communicate and ensure resolution of noncompliance issues and establish the records. [60]

Organizational process focus is a process management process area at maturity level 3. It is important because due to this we can plan, implement and deploy organizational process improvements. We have deep understanding about strengths and weaknesses of the organizational processes and process assets. Organizational process focus benefits process improvement because through it we know about process improvement opportunities. We launch organizational process needs, review processes and discover process improvements. After that we plan and implement process improvements by setting action plan and implementing those plans. Then we implement process assets of the organization and relate lessons learned. [60]

The disadvantage of CMMI/ISO is that certain companies aim to get certification only for making business and earning profit. Those companies don't focus on quality management their aim is certification/level not the improvement. The intent of companies for getting certification/level should be improvement so that they meet international standards. Their focus should not be attaining a score, either the score is a certificate or a maturity level it should be building competitive advantage through improvement [61].

Another disadvantage is that CMMI/ISO is costly for small and medium enterprises. So they don't have many resources i.e. financial, human etc.

Augustine's Law states that "It costs a lot of money to build bad products" [65]. So for making good products we need to follow standards. For keeping international standard implementation of ISO/CMMI is necessary. If we are not taking care of any standard the quality in products will deteriorate and cost will be high.

The advantage of ISO is that the companies implement global standards of product and service quality. Due to this there is no company specific standard. Company enters into global market since same requirements are applied in several countries. [65]

ISO certification means that product qualify for the needs of new customers. It makes developers to think about quality that what quality is in open source software development processes perspective. It's a mean to achieve quality and to improve the quality management systems in the long term.

CMMI is important for quality systems because it plays significant role in integration of systems and open source software engineering. CMMI adoption and implementation assures substantial benefits to the industry. [62]

A roadmap is provided by CMMI to organizations for improving their abilities for development and maintenance of quality products and services [63]. If the organizations follow CMMI they can improve the quality and attain customer satisfaction.

The CMMI models can be applied to any organization for the improvement of process. It shortens the time of development cycle, improves the quality and lessens the costs [63]. It is said by Phillip Cosby that the quality is free. So through the use of quality process we can acquire, develop and maintain quality of the system.

Currently many companies in key markets like defense, aerospace, automotive, entertainment, telecommunications and finance are adopting CMMI. CMMI's major focus is on quality through process and product quality; business results; and engineering in quality. It eliminates inconsistencies and minimizes replication [65].

The future is the era of "innovation in quality". So for that CMMI/ISO have some challenges to deal with exceptional engineering. Few of the challenges are [65]:

- Customers want high quality products with low costs.
- Companies demand raise in productivity.
- New fields in engineering are continuously merging and evolving.
- Many organizations go for partnering so there is dynamic behavior of organizations.

The mentioned challenges need to be addressed by CMMI/ISO to keep the pace with future and help in the development of open source software quality systems.

4.0 COMPARISON OF OPEN SOURCE AND CLOSED SOURCE SOFTWARE PRODUCTS

4.1 Problems for Open Source Software

4.1.1 Coordination Mechanisms

There are much people in the form of volunteer developers that develop the open source software; those may be in hundreds or in thousands. In this type of development, proper work has not allocated to any volunteer because each volunteer do the work according to its choice. And the development of open source software has not occurred by proper arrangement and schedule. [32]

Furthermore, the volunteers are in different locations from where they do their work on open source software. These volunteers don't meet face to face often and mostly there link with each other become via emails or phone calls. The software that has produced in such situation will not be perfect regarding working wise. [32]

4.1.2 Maintainability

Maintainability takes more expenses in existing software projects. Now day's lots of people are using open source software. For the maintenance of such types of software, there is nothing face to face

communication among the maintainers. So they use documentation for source code and message board for contact with each other. It is necessary to maintain the quality of open source at the time of its development, so that we could save from maintenance expenses. [23]

We can say, due to non proper interface definition, structural complexity & absence of proper documentation of existing open source software have created the problems for user and taking more expenses for maintenance of the OSS. [23]

4.1.3 Open Availability of the Source Code

We can easily take the source code of open source software. We know that volunteers use this code for the improvement of the software, but hackers in the form of volunteers can take this source code and using after some processes, could hack or damage our software system. So the free availability of the source code is a problem for the security of the OSS. [23]

4.2 Benefits of Open Source Software

4.2.1 Cost Advantage

Open source software is cheaper than the close source software. In its result every user that has not strength to purchase the close source software can easily use the open source software with no cost. But some of open source software is available on low price like enterprise Linux. And another way due to which we can reduce our expenses is the use of Linux due to which we can use those types of hardware that have low cost. The one main factor due to which the price of OSS is less than proprietary software that is the less cost of the license of OSS. But it is also consider that for switching the system from close source software to open source software more cost has to expense. [33]

4.2.2 Free Availability of Source Code

We can gain source code easily of the open source software due to which it is easy for us to mold the software according to our demand like we can increase and decrease the functions of the software by doing the coding own. There are many more faculties, choices and control that could be done through the free availability of the source code. [34]

By changing the software own, our control and trust will increase. Furthermore, source code is also important when developing products based on open source software because this open source gives more insight. [34]

For applying the modifications in open source software it's necessary to know the knowledge of, how to do it. Expertise is necessary regarding it. [34]

4.2.3 Maturity

The developers of the open source software are trying to keep it mature according to quality wise. So, they have claimed that more projects especially Linus is reliable and we can trust on it. It can work in each type of condition and fulfills the quality of the software project. But it has rejected by saying that Linus and more software of the open source software are unreliable. [35]

4.2.4 Reliability

The open source software means increased security and reliability; because code is in the public view and it will be uncovered to extreme analysis for its users, then the problems being discovered and solved instead of being kept secret. Moreover by reading the source code, users can find problems and solve them only if they are good programming skills. [66]

4.2.5 Open Source as a New Way to Develop Software

Opening the source code permits a modern and innovative approach to software development team is one of the strongest claims made by the open source community.

Many peoples of different minds from different teams append their work in open source software development that initiates new and innovative ideas. [66]

4.2.6 Open Source and the Protection of Customers

One the main advantage of Open Source Software is that it is considered a way to protect customers and, specially, public administrations, institutions, and government agencies. [66]

4.2.7 Open Source Software and Safety

Open source software is attractive because it enables to achieve following goals

Software can be examined to check whether it is up to security and safety requirements or not.

Software maintenance can be managed in a much more effective way, since a customer is free to change the company maintaining the software. [66]

4.2.8 Motivation

The open source approach considered a strong “catalyst” to promote effective development practices.

It is certainly a very effective and rewarding way to involve people of different minds from different teams in an Open source software development project. [66]

4.3 Problems for Close Source Software

Close Source software are costly than the open source software. Its main reason is that the cost of license of proprietary software is more than the OSS. Closed software organizations has to invest their huge part to protect their products from software piracy or misuse, from reverse engineering and duplication

Closed Source software organizations have to maintain the satisfaction needs in the market and to gain the competitive advantage in the software industry. [36]

In order to gain the competitive advantages the closed software companies are under huge amount of stress to continually upgrade according to customer need what they are selling. Some times the closed software is dropped out on the customer door before it is ready to be out. [36]

4.4 Benefits of Close Source Software

Once closed software written then it can be copied and then reproduced for a very little cost for its users and it can increase its commercial value. Once written, it can be copied and reproduced for very little cost, and can have great commercial value. Closed source software is developed by a single person or organization. Only the final product that is run on customer computer is made available, on the other side all vital source code or recipe for making the software is kept a

secret. This closed software is normally copyright or patented and is legally protected as intellectual property. [36]

Closed Source software is created to satisfy a need in the market and to gain the competitive advantage in the software industry. In paying for the software customer get some definite perks. Customers can expect documentation to be available with whatever they purchase and they can expect the application to perform in the way it was promoted. [36]

5.0 DISCUSSION

We can say that quality looks like a beauty and it feels good impact on the eyes of users. Open source software has to develop in time, with in the range of budget and it has to fulfill the necessities of users. More and more functions cover to in the development of software. Code should be according to the demand and open source should be secure from every side. All these are the factors/functions that have to adopt in the development of qualitative open source software.

Through our literature study we found out that it is possible to increase the quality in open source software by using the best quality assessment, best measurement, best processes, and the best tools and by using ISO Standards.

Existing approaches and methods described in our thesis are also a struggle to improve the quality of OSS. Along with it, software process, development methods and analysis have been described to gain the better result from the open source software. For evaluation the criteria, questions are produced and their answers describes the quality of OSS and different methods which are used in evaluating the quality during its process of production. Customers also play their role to improve the quality of software. At the end there are many risks that have to be controlled to make the OSS secure, trustable, qualitative and friendly.

It is proved by the research that the quality of open source software is improving very quickly. Day by day new techniques are introduced in the market to improve the quality of open source software. Different types of security checks are also available but I think there is still a need of more improvement in the quality of open source software.

Basically we can divide the software quality in to two types of different divisions which are quality control and quality assurance both of these are very much important parts of the development of good quality software whether it is open or closed source software.

Quality assurance is a thing which is involved through out the development of software and this thing is very much important that we must remember the mistakes and the draw backs which were left in the past, and from the lesson of those draw backs you have to improve the process of software development. On the other hand the second important thing in the development of good quality software quality control is much important thing; verification and validation are the major aspects of quality control.

A few other important things which are involved in the development of software are project management, code modularity, and other thing is test process management. Some other things according

to our point view which are very much important or you can say are the main hindrances in software process improvement are:

- Non-significant level of training to the employees
- Organization don't identify its best practices
- Organization don't use existing processes and write new processes
- Organization use processes which are large and not usable for process improvement
- Implementation of a software improvement process without accessing its readiness
- The use of agile processes without its complete knowledge within the organization

Training of the employees to new tools and techniques is much important to keep pace with the changing software development industry. Software process quality is directly related to the amount of training. This is a very important issue.

Quality manager plays an important role in the development of any type s of projects so from the Quality manager ends it, is important that he or she should make sure that Consultancy firm is following some suitable project development process model i.e. Linear Sequential process model etc instead of customer process model. Quality manager should ensure that a proper risk analysis has been carried out for the development of project and mile stones have been set.

Risk analysis should include cost estimation, time estimation, and human resource estimation. He should give suggestion to the project manager that he should closely monitor the completion of the project with mile stones set in the beginning. Moreover, Quality manager should check that the customer requirements meet precisely, beside these it is also his responsibility that he should make sure that product developed is bug free and for that sake proper testing should be carried out.

Best quality of the software also depends on keeping huge sustainable community it also increase code development, debugging and a lot of new features. Documentation is also very much important, especially the good documentations tools which are used during the development and reward and recognition, this type of things are also very much important in increasing the quality of products. Review of the code by the people who are not in the team is also of very much importance. Especially the environment of the team is of comes in it.

We think that maximum can results only be achieved by a highly organized team and by using some good tools support for collaboration, debugging, and code submission.

Open Source software can also be improved by the contribution of different users and developers from different global locations. If any problem comes during its installation or during a running process, a user has to wait for the help of other guiders.

Firstly there were many problems / short comes in OSS but with the passage of time its quality become improved. Now a day's mostly developers are turning to work in open source software. Different methods, approaches, techniques and models are in use to improve the quality of the open source software but there is yet more need to improve the quality of the OSS especially in the field of Security, finance

and health because these departments have very secure and sensitive systems. In the line of security open source software has to do more and more work for make it safe and sound.

But are many points regarding the security of open source software in our minds that it is possible in the development of open source software that there may be any volunteer developer in the form of hacker and he could harm to the security of the system because he knows its source code and mechanism. In banks, space and defense fields' security has key stone importance so it is necessary that software which has developed should be quality assured, verified and secure from any attack/attacker.

Same problem could be occurred due to the exposable source code of open source software. Every body that may be in the shape of attacker or defender could study it with no hesitation. In this way attacker gains its reverse benefit and does the attack on the existing open source software. It factor has reduce the quality of the open source software. We can say that it is a major draw back of the open source software that is creating the problem for the improvement of it.

In close source software there are recognized teams that developed the software and they make it better after the reply of its beta version of the software. It is not necessary in the development of free software that there should be a recognized team that could work together for the development of OSS. In OSS it is possible to re-use its source code but in close/propriety source code it is impossible.

In OSS development process, there should be a recognized team that should have strong communication link among each others to solve any problem. This problem could be produce at any stage of software development process and during its work. It should be in mind that all the members of the team should be well educated and have the authentication of each other. These are the factors that can bring the reliability and security and these are the reasons to produce the trustable and cheaper software.

As we know that in close source software we can not re-use the source code of the software but it is possible in open source software .This is why close source software are more costly and open source software are reasoning of cheaper software development.

Here is a problem according to our opinion, although the cost of the software may reduce by the re-usage of source code but the developer that is re-using to this code may not be an expert and reliable because in this type of development mostly developers belong to remote locations and have shortage of communication with each other.

So at the end the software that has developed may not authenticate and fulfill the quality of the OSS especially regarding sensitive fields of security, finance and health. In short this type of development can create problems and it may reason of huge blast and accident.

An example of open source software failure came in being that time when a rocket of U.S. Mariner was launched in July 1962 and only after five minutes of its departure, it was destroyed. The main problem was in its programming and in his implementation. This destruction had created lot of loss for USA and the trust on open source software has become educed on it. It was only in the result of non expert software developer.

After the completion of open source software, its work does not fulfill. Updates and patches are necessary for it and any volunteer developer do not take its responsibility. In the result of it, problems become to create in the systems that become reason of software disturbances. According to us there should a regular department of open source software that will responsible of this kind of work.

In close source software patches are launched due to which software become update regularly, and the quality and efficiency of the software become remain contain. All of it occurs only in the result of a recognized team.

We can also say that the growth of OSS is an attractive progress model for difficult products which request to judicious and heterogeneous user base. Influence of the Internet has permitted for know how trading to be done on a huge scale with contribution from all sides of earth. Finally, OSSD is constant with existing economic theory and history. Open Source Software is a feasible and fit way of developing elevated standard software and it has long and attractive history ahead of it. [70]

According to us software that has developed in open source environment should not use in sensitive fields of life, because it is not assure that it fulfills the quality of software or not. For example we consider that its quality is well and according to the standard but any hacker that has worked in this environment in the form of volunteer developer could create problems for the system and could reach harm to it.

In sensitive fields of life, we should use such types of software (open source or close source) that are developed by well know and trustfully companies. It is the best of that all these types of software should be developed by own recognized, expert, trusty and intelligent team.

Time is money and it effects on the quality of the software. In open source software mostly there is no time limit in the development of the software, because developers are belonging from remote and distant locations and there is no any hurry in their mind that this software should be complete in this duration. Although in close source software there is a time limit of the development of the software. In short, open source software could take more and more time as compare to the close source software.

Software that are developed in close source environment especially Microsoft corporation related software, should have more quality of integrated with open source software. It has seen that OSS has some problems with CSS as performance wise. According to our opinion, there should be an agreement between OSS and CSS that should fully allows it that both types of software could use each other with full reliability and in friendly environment. These types of agreements could increase the quality and performance of the open source software.

In our thesis, we have also make comparison of open source and close source software. Each has its own importance and properties. Customers require software regarding their necessity and budget. So they should adopt that field of development (close source or open source) that is suitable and reliable for their work. If there is a right adoption of software development method than they could fully get the benefits from the selected software.

Normally it has seen that the usage of open source software is some difficult as compare to the close source software. It should be in mind during the development of open source software that OSS should be easy in use so that it may be adoptable and likeable by every type of the user. This factor also increases the quality of the open source software.

It has also been seen that the installation process of OSS is difficult and a general user could not install these types of software. For this purpose, an expert has to be called and that becomes the reason for more cost. The installation process of OSS should be easy so that everybody could install it in a leisure environment. This factor could also increase the standard and adoptability of the OSS. Furthermore, a shortage of the features of OSS has created a reason for costly technical support. Easy and in-time availability of features could reduce the need for technical support and could increase the value of the open source software in the market.

A guarantee of anything shows its quality, standard, and truth. This is why everybody trusts on this word and buys that thing without any hesitation. If we adopt any OSS, there will not be a single company that could give us backup of our system and could help us in the time of a problem. It is a main factor that decreases the quality of OSS. If we bring this type of property in open source software, then people will trust on it and the quality of open source software will increase.

6.0 CONCLUSIONS

We have studied literature on open source and close source software. In our study, we found out that due to slightly different development processes of closed software and open source software, there arise specific problems in the case of open source software. We found out that the main problems with the open source are security, finance, space science, and health. It also explains that how can we bring the Quality Assurance and evaluate the open source software. Furthermore, models, processes, and methods have been described in our thesis to produce the above said qualities in the open source software. Both open source and close source software have their own importance and customer uses it according to their demand. There are yet some problems in open source software as compared to close source software and it is needed to bring the improvement in them. In short, we can say that anything of the world is not complete, it looks complete but after some time its drawbacks become apparent. Furthermore, we can say that open source software development is rapidly progressing in the field of software development and it is up-and-coming as a substitute move toward development of large software systems.

FUTURE WORK:

Any thing of the world is not complete although that time it looks perfect but with the passage of time its draw backs become appear and has to search their solutions. As we know that open source software development is rapidly progressing in the field of software development and it is up-and-coming as a substitute move toward development of large software systems. Along with it the quality of open source software is also improving swiftly. Day by day new methods, approaches, techniques and suggestions are introduced to improve the quality of open source software but there is yet more need to improve the quality of the OSS especially in the field of Security. In our thesis we have suggested eight points, which could be helpful in order to improve the quality of open source software. The aforementioned points are partly derived from our literature study. From this study, we found out that there are some weak points, which still need to be handled.

We found out that in most cases, these points are ignored during the development of the open source software but these points are significance in the development process. Specifically, to improve the security and hence the quality of the open source software, different types of security checks are available but we think there is still a need of more improvement in security of the open source software.

We would like to present our suggestions in the form of a checklist so that it could work as a benchmark for the quality improvement of the open source software.

1. It is possible that there may be any volunteer developer of open source software in the form of hacker/cracker and he could harm to the security of the system because he knows well to its source code and mechanism. So, it is necessary that there should be a verification of the developer to assure that he has good character, society and links.
2. Source code of open source software is easily and openly available for every one for the purpose of edition, alternation and software improvement and attacker/defender could study it with no hesitation. But it may be reverse in the favor of the organization and one non-verified person could harm and destroy the system. So according to us, source code should only be available for authorized and verified persons.
3. There should be well organized departments of open source software, which have recognized, well educated, intelligent and verified teams. They should be responsible for all types of problems that open source software has to face like,
 - a. To make it assure the best quality of the open source software after its reply of beta version from the users.
 - b. To solve the problems of the users at any stage of installation and running process.
 - c. Responsible for the updates and patches of the open source software.

- d. There should be strong communication among the team members during the software development process to solve the problems.
 - e. Non-experts and less educated persons should not be the part of the team, because they could create problem/accident for the system on which software has implemented.
4. Time is money and it is often seen in the development of open source software that it takes more time for its development due to non-organized department and team. This factor put bad impression on the quality of the open source software. As compare to close source software they are not fully responsible for time limit development.
 5. Software that has been developed in close source environment should have more quality of integrated with open source software. It has been seen that OSS has some problems with CSS as performance wise. According to our opinion, there should be an agreement between OSS and CSS that should fully allows that both types of software could use each other with full reliability and in friendly environment. These types of agreements could increase the quality and performance of the open source software.
 6. Open source and close source software has its own importance and properties. Customers require these types of software according to their actual demand and budget. This point is related to the user. The customer should be aware of making the correct choice of software, if he has done the right adoption of software type then they will fully get the benefits from the selected software.
 7. It should be in mind during the development of open source software that it should be easy in use and installation process. There should also be proper documentation / features and cheaper technical support so that OSS may be adoptable and likeable by every type of the user. This could increase the quality of the open source software.
 8. Guarantee of any thing shows to its quality, standard and truth. This is why every body trusts on this word and buy that thing without any hesitation. This factor has less found the field of open source software development. If we bring this type of property in open source software development field then people will trust on it and the quality of open source software will increase.

Our thesis does not end here. We have to proof our suggestions and for this purpose, we have contacted an open source software development house that is situated in Pakistan and working within finance. It mostly developed the most secure banking related software.

For the purpose of validation of our proposed checklist, we made a special request, first by phone and then by email, to its chief executive for the evaluation of our suggestions [71]. When they replied us, we found out that they found our suggestions valuable and that these are most likely

applicable. According to him, there are many projects that are under development in their software house and they could not apply your suggestions immediately. But for new software development projects, it would benefit to take your suggestions into account.

But he made a meeting with the experts and senior members of the software house on our request. Mostly team members gave importance to our work. They decided that these suggestions should be a part of our development next time when new schedule will prepare for new software development because they were missing more of these points during their work.

A more extensive validation of our checklist is an interesting future work.

REFERENCES

- [1] B. Boehm, "Software Engineering Economics," *IEEE Transactions on Software Engineering*, vol. 10, pp. 4-21, 1984.
- [2] C. L. Jones, "A Process-Integrated Approach to Defect Prevention," *IBM Systems Journal*, vol. 24, pp. 150-167, 1985.
- [3] W. J. Sung, J. H. Kim, S. Y. Rhew, "A Quality Model for Open Source Software Selection", Soongsil University Korea, n. d
- [4] W. A. Ward, J. B. Venkataraman, "Some Observations on Software Quality", University of South Alabama, 1999
- [5] HALSTEAD, MAURICE H., *Elements of software science*, Elsevier North-Holland, New York, 1977.
- [6] JONES, T. CAPERS, "what are function points?" <http://www.spr.com/library/Ofuncmet.htm>, 1997.
- [7] KEMERER, C. F., "How the learning curve affects CASE tool adoption," *IEEE Software*, vol. 9, no. 3, pp. 23-28, May 1992.
- [8] ARTHUR, L. J., *Improving software quality-an insider's guide to TQM*, John Wiley & Sons, New York, 1993.
- [9] MCCABE, T. J., "A complexity measure," *IEEE Transactions on software engineering*, vol. SE-2, no. 4, pp. 308-320, December 1976.
- [10] HAAG, STEPHEN, RAJA, M. K., AND SCHKADE, L. L., "Quality function deployment usage in software development," *Communications of the ACM*, vol. 39, no. 1, pp. 41-50, January 1996.
- [11] CARD, DAVID N., "Statistical process control for software?," *IEEE Software*, vol. 11, no. 3, pp. 95-97, May 1994.
- [12] SCHMAUCH, CHARLES H., *ISO 9000 for software developers*, ASQC Press, Milwaukee, WI, 1994.

- [13]HUMPHREY, WATTS S., "Using a defined and measured personal software process," *IEEE Software*, vol. 13, no. 3, pp. 77-88, May 1996.
- [14]SARNA, DAVID E. Y. AND FEBISH, GEORGE J., "Our favorite tools," *Datamation*, vol. 42, no. 4, pp. 24-26, February 15, 1996.
- [15]SQLBENCH INTERNATIONAL, "The database performance tuning & evaluation solution," <http://www.sqlbench.com/SPEC.HTM>, 1996.
- [16]BARSTOW, D. R., SHROBE, H. E., AND SANDEWALL, E. ED., *Interactive programming environments*, McGraw-Hill, New York, 1984.
- [17]K.Stroggylos, D.Spinellis, "Refactoring – Does it improve software quality? ", Department of Management Science and Technology, Athens University Greece, n. d
- [18]W. F. Opdyke. Refactoring Object-Oriented Frameworks. Doctoral thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1992.
- [19]T. Mens and T. Tourw'e. A Survey of Software Refactoring. *IEEE Trans. Softw. Eng.*, 30(2):126–139, February 2004.
- [20]B.W. Boehm, J. R. Brown, and M. Lipow. Quantitative evaluation of software quality. In *Proceedings of the 2nd International Conference on Software engineering*, pages 592– 605, 1976
- [21]I Samoladas, I. Stamelos, "Assessing Free/Open Source Software Quality", Aristotle University of Informatics Greece, n.d
- [22]P. Caliman, Q. Consortium, " Software product quality evaluation and certification: the Qseal Consortium methodology", v. Quintiliano 43, 20138 Milano, Italy, n.d
- [23]U. Raja, E. Barry, "Investing Quality in Large –Scale Open Source Software", Texas A&M University, 2005
- [24]International Organization for Standardization. (1991). *Information technology-Software product evaluation: Quality characteristics and guidelines for their use. ISO/IEC IS 9126*. Geneva: ISO Kernel.Org Organization, Inc. (2003) The Linux kernel archive. Retrieved March 14, 2003 from: <http://www.kernel.org>
- [25]Fenton, N: *Software Metrics: A Rigorous Approach*, Chapman and Hall (1993)
- [26]T.Punter, R.V.Solingen, J. Trienekens "Software Product Evaluation", 4th IT Evaluation Conference, Netherlands, 1997
- [27]K. Sacha, "Evaluation of the Software Quality", Warsaw University of Technology, Poland, n.d
- [28]DROMEY, R. GEOFF, "Cornering the Chimera," *IEEE Software*, vol. 13, no. 1, pp. 33-43, January 1996.
- [29]ISO 9001: Quality management systems – Requirements. ISO (2001) and

- [30]Szejko S.: RDQC – sterowana wymaganiami kontrola jako ci oprogramowania, in: Górski J, Wardzi ski A. (eds): In ynieria oprogramowania: Nowe wyzwania, WNT, Warszawa (2004)
- [31]S Basili V.R., Weiss D.M.: A Methodology for Collecting Valid Software Engineering Data, IEEE Transactions on Software Engineering, Nov. (1984)
- [32]A.Mockus, J.D.Herbsled, ‘‘Two Case Studies of Open Source Software Development: Apache and Mozilla’’, Avaya Labs Research & Carnegie Mellon University, 2002
- [33]K. Ven and J. Verelst, ‘‘The Impact of Ideology on the Organizational Adoption of Open Source Software,’’ *J. Database Management*, vol. 19, no. 2, 2008, pp. 58–72.
And
T. Wichmann, ‘‘Use of Open Source Software in Firms and Public Institutions—Evidence from Germany, Sweden and UK,’’ Floss Final Report, pt. 1, Int’l Inst. Of Infonomics, Berlecon Research, 2002.
- [34]J. Feller and B. Fitzgerald, *Understanding Open Source Software Development*, Addison-Wesley, 2002.
And
C.B. Weinstock and S.A. Hissam, ‘‘Making Lightning Strike Twice,’’ *Perspectives on Free and Open Source Software*, J. Feller et al., eds., MIT Press, 2005, pp. 143–159.
- [35]J. Dedrick and J. West, ‘‘Movement Ideology vs. User Pragmatism in the Organizational Adoption of Open Source Software,’’ *Computerization Movements and Technology Diffusion: From Mainframes to Ubiquitous Computing*, K.L. Kraemer and M. Elliott, eds., Information Today, 2007.
And
J. Dedrick and J. West, ‘‘An Exploratory Study into Open Source Platform Adoption,’’ *Proc. 37th Hawaii Int’l Conf. System Sciences (HICSS 04)*, IEEE CS Press, 2004, p. 80265b.
- [36]C. MICHLER, ‘‘THE PROCUREMENT DECISION "OPEN" OR CLOSED" SOURCE SOFTWARE?’’, 2007
- [37] Bo Bergman & Bengt Klefsjö, ‘‘Quality from customer needs to customer satisfaction’’, Student litterature (2003).
- [38] http://www.pcmag.com/encyclopedia_term/0,2542,t=cust+om+software&i=40588,00.asp
, Novemeber 2006.
- [39] Thomas HC CHILDS , Kenneth W DALGARNO and Alison MCKAY , ‘‘Delivering Mass-Produced Bespoke and Appealing Products’’.
- [40] Thomas HC CHILDS , Kenneth W DALGARNO and Alison MCKAY , ‘‘Delivering Mass-Produced Bespoke and Appealing Products’’.

- [41]PC MAGAZINE ,
http://www.pcmag.com/encyclopedia_term/0,2542,t=software+package&i=51692,00.asp , visited date 5th July 2008.
- [42] The MANAGEMENT ROUNDTABLE , The Leading Practitioners' Resource for Product & Technology Development , http://www.roundtable.com/Event_Center/MDPD/MDPD.html , visited date 2nd July 2008.
- [43] http://www.webopedia.com/TERM/R/real_time.html visited date 25th June 2008.
- [44] OMIMO ,<http://www.omimo.be/encyc/techno/terms/defini/def.html>.visited date 6th July 2008.
- [45] Swift, M. M., Bershad, B. N., and Levy, H. M. 2005. Improving the reliability of commodity operating systems. *ACM Trans. Comput. Syst.* 23, 1 (Feb. 2005), 77-110. DOI=<http://doi.acm.org/10.1145/1047915.1047919>.
- [46] Cyntica Eaton and Atif Memon, "Evaluating Web Page Reliability across Varied Browsing Environments".
- [47] <http://www.energywindow.com/usrey/research/qmj1abst.html> , visited date 2nd July 2008.
- [48] University of Oulu, <http://www.ee.oulu.fi/research/ouspg/sage/glossary/> , visited date 3rd July 2008.
- [49]Cause and Effect Diagram , <http://www.hci.com.au/hcsite2/toolkit/causeand.htm> , visited date 7th July 2008.
- [50] Jennie Carroll , " The application of total quality management to software development" Australia.
- [51] <http://satc.gsfc.nasa.gov/assure/agbsec3.txt> ,visited date 3rd July 2008.
- [52] Ian Sommerville, " software Engineering", edition 2004.
- [53] Software engineering Institute Carnegie Mellon ,
<http://www.sei.cmu.edu/cmmi/general/index.html>, visited date 5th July 2008.
- [54] Discover ISO [online], Available from: http://www.iso.org/iso/about/discover-iso_meet-iso.htm , visited date 25th June 2008.
- [55] CMMI and ISO, Conny Johansson, Blekinge Institute of Technology, Sweden.
- [56]Capability Maturity Model Integration ,
http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration, visited date 7th July 2008.
- [57]International Organization for Standardization ,
http://en.wikipedia.org/wiki/International_Organization_for_Standardization, visited date 7th July 2008.

- [58] International Standard Organization, ISO 9001:2000 and CMMI Synergy, page 106-120.
- [59] Mutafelija, B., "software Process Improvement: Synergy between ISO 9001:2000 and CMMI," paper presented at the SEPG Conference, New Orleans, LS, 2001.
- [60] CMMI Process Areas [online], [http://en.wikipedia.org/wiki/Process_area_\(CMMI\)](http://en.wikipedia.org/wiki/Process_area_(CMMI)), visited date 7th July 2008.
- [61] Mark C. Paulk, software Engineering Institute, Carnegie-Mellon University.
- [62] Dr. Nancy Spruill, Director, Acquisition Resources and Analysis, AT&L, CMMI Technology Conference and Users' Group, 2001.
- [63] Insight, Workshop on SEI Accredited CMMI Introduction by Insight, http://www.insight-test.com/TheLearningExperience/index_back.html#, visited date 3rd July 2008.
- [64] MITSUE-LINKS, CMMI Overview, <http://www.mitsue.co.jp/english/company/management/cmmi.html>, visited date 8th July 2008.
- [65] Carnegie Mellon Software Engineering Institute, Mile Phillips, Mary Beth Chrissis, Mike Konrad, Sandy Shrum, CMMI Improving and Integration, 2003.
- [66] Open source software—an evaluation, Alfonso Fuggetta, Politecnico di Milano, Dipartimento di Elettronica e Informazione, Piazza Leonardo da Vinci 32 and CEFRIEL, Via Fucini, 2, I-20133 Milano, Italy, The Journal of Systems and Software 66 (2003) 77–90
- [67] Elliott, M.S., Examining The Success of Computerization, Movements in the Ubiquitous Computing Era: Free and Open Source Software Movements, in Kraemer, K.L. and Elliott, M.(Eds.), Computerization Movements and Technology Diffusion: From Mainframes to Ubiquitous Computing, Information Today, Inc., to appear, , 2008.
- [68] Mockus, A., Fielding, R., & Herbsleb, J.D., Two Case Studies of Open Source Software Development: Apache and Mozilla, ACM Trans. Soft. Eng. Meth., 11(3), 309-346, 2002.
- [69] E. S. Raymond, <http://www.tuxedo.org/~esr/writings/cathedralbazaar/> cathedral-bazaar/," 2003.
- [70] Robert L. Greenberg, Open Source Software Development, Thesis for Senior Honors at Brandeis University, May 9, 2003
- [71] Personal communication with Seamax International, 2008-09-01.