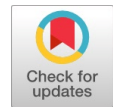


# A Quality Communication Framework for Open Source Softwares



A. Abdul Rahman

Table.I Closed Vs Open source System

**Abstract:** Open Source Software (OSS) is an emerging technology of software development in present era, careful implementation and procurement can gives us more benefits for OSS- adapting organizations. But any failure in design, development and procurement will leads to major impact to OSS-developer organization (community) to stop their further release and completely destroy the adapted system. Success of any software's based on it's time to delivery, development cost and effectiveness. Open source software can fulfill these factors due to its availability of source code, openness and sustainability of Technical quality among development community. At the same time, more number of developers in a community of any OSS leads to delay in development. But it has chance to improve software effectiveness in terms of defect free, since software can be reviewed using various testing methodology, tools and plans in different perception. An OSS has more issues such as lack of architecture design, co-ordination, organizational structure and documentation, removing these issues will strengthen the quality of OSS. This paper includes the quality model and framework for organized communication the quality characteristics in an appropriate aspects is required since unorganized communications wherever people involved can leads to high quality risk.

**Keywords:** Quality Requirements Process, Quality Factors, Framework for Organized Communications, Contribution validation process.

## I. INTRODUCTION

The Open Source Software environment and an effective communication among the developers and users community can avoid the major defects of software development in initial stage itself. Especially OSS has the more number of people in their community like core developers, testers and end users. Organized communication among these people can improve the quality of OSS and useful for various reasons like functionality, licensing, acceptability, manageability and easiness. Understand the characteristic of OSS are important to frame the communication method, and OSS different from traditional development approach and feature of OSS are shown in table 1. The reasons for developing OSS's organized communication are [1].

Closed Source Software	Open Source Software
Well –defined Development process	Development processes won't define or documented in advance.
Extensive project documentation	Little project documentation
Structured testing and SQA	Unstructured testing and SQA
Analyst define requirements	Programmers define requirements
Process's risk management monitoring exits till completion of development	No formal risk monitoring assessment process
Measurable goal throughout project	Few measurable goal
Finding defects using back-box testing in early stage of development is possible.	Error finding in early stage not available.
Quality standard using empirical method will be done.	No such methods are conducted.
Work will be assigned to each Team member.	Contributors will choose work
Formal approach to software design will be followed.	No formal approach, coding will be done directly.
Standard will be followed planning and scheduling	Most of project won't follows.

1. To improve an architecture design effectively among community.
2. Organized communication can improve the Software quality Assurance with reasonable resources.
3. To understand about terms and condition to use the product, also helps to organization to extend their service by utilizing their licensing amount.
4. Helps to developers to completely understand the work of others to avoid the redundancies in development.
5. Organized communication helps to the management to avail of valid and long-term sponsors to support their work.
6. Improving the functionality and services of OSS also possible, by adding valid requirements and code in existing version to release the new version provide an organized communications exists.

In general, large number of people in community need of proper organized communication at the same time more number of people in coding phase will lead to delay in delivery and more number of people in testing phase will improve the quality of product. This paper presented a framework for adaptors requirement process and organized communications in major phases of OSS development based on review of our literature survey.

## II. RELATED AND PROPOSED WORK

Improving quality of OSS is not a new idea there are many research work have done so far,

Manuscript published on 30 December 2019.

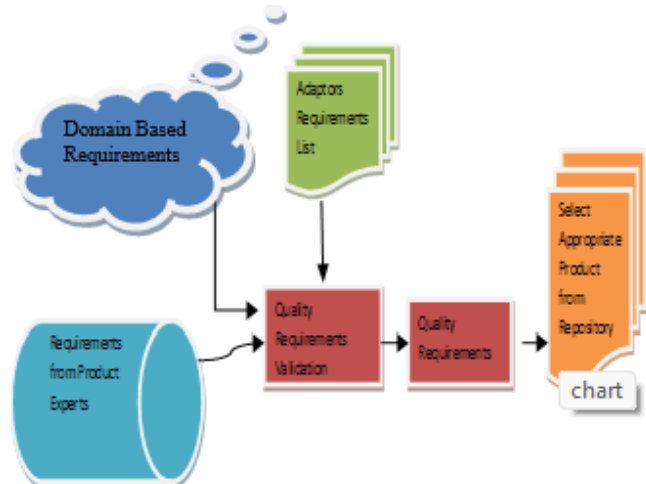
\* Correspondence Author (s)

A.Abdul Rahman\*, Associate Professor, Computer Science and Engineering, Godavari Institute of Engineering and Technology, Rajahmundry, India. Email:abd\_ul\_rahman@rediffmail.com.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

But limited research works are done with specified concentration on organized communications. ISO/IEO 9126 quality model is limited to evaluate the characteristic of OSS was not reflected [11]. Madanmohan & Rahul Suggested Four steps for quality model but modifications of specific procedures and the quality evaluation model are needed [11]. In Marious Popa approach, Communication and testing were not considered [2]. We observed that organized communications are necessary for project specification, documentation, licensing, Sustainability of community, and identification of skill required for adaptation, and third –parity support. The ideas of OSS are availability of coding for public, free redistribution, right to modify and no restriction in licensing and interaction available through web-based technologies. Quality requirement process model of Components based software (CBS) cannot be used for OSS since CBS don't have the source code for user and method of gathering of requirements for CBS and traditional software development are also different from OSS.

The functionality of CBS components are not possible to correct in case of any malfunction by the users because of the non-availability of source code. One of the major challenges in OSS is how evaluate its environments; especially improvements in the security and quality of the system [8] time to time alternations are made in software. There is an important of structured approach to manage its need and product attributes to implement the software procurement. There should be a hierarchical approach to represents system quality requirements from valid user. Identifies OSS user also play a vital role in OSS quality model. Next, periodic release also important for OSS once the modification and requirements are accepted. Improper releases are making impact in software quality [4]. The research has shown that a large number OSS project follow a feature –based release strategy, scheduled in lager intervals. This paper aimed to provide the guidelines to ameliorate the OSS project by suggesting communication process model. This includes structured methodology for requirements and hierarchical representation of quality process and organization/ product oriented communication plan. Software Quality Assurance (SQA) is a systematic approach to ensure product working correctly. SQA process on open source software depends on quality model. Once quality model is prepared appropriate metric can be selected based on attributes in quality model. Quality requirements are gathered from different stakeholders (user, developers, experts, and organizations concern with similar software product) domain-based quality attributes are collected from domain experts, some of the attributes also identified by developers, commercial partners and valid users. In all cases quality requirements must be validated with risk and constrain of product by the core members, and exports committee. The following figure II. Shows the steps involved in quality requirement preparation.



**Fig. I OSS Users' Requirements Specification Process**

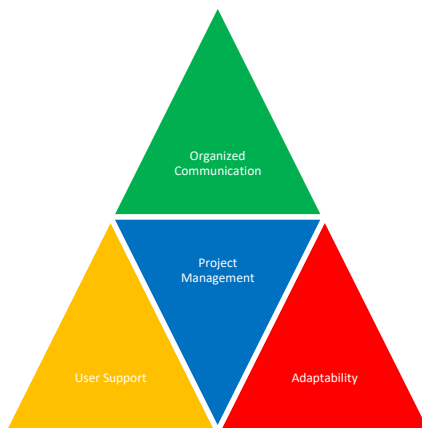
### III. ORGANIZED COMMUNICATION FRAMEWORK

Software quality has various definitions in different perception. Quality definition of IEEE as [2] "The degree to which a customer or user perceives that software meets his or her composite expectations" is considered to extent our work. In software development, quality is derived from three important sources namely people, technology and Management [2]. Principally OSS has lack of architectural, design and Documentation but these will help to the users to select the appropriate software called procurement. In case of failure in OSS procurement leads to loss of huge amount and development time of complete system. A Model can improve the communication, analysis, design and quality of software. Determination of quality requirements makes the procurement process as ease. We have identified the core area of OSS-life cycle for organized communication and classified as five different phases they are: functionality, learn-ability, acceptability, Manageability and easiness shown as fig3. Functionality phase includes communication for accepting code, and compliance / feedback from and to the users. Second phase called learn-ability shows the communications to make the users to understand about functionality of OSS, licensing, technology used and people involved in development community. Adaptability of software not only dependence on user specifications, But also based on people, process and tools used. so communication highly makes the users to aware of functionality available on OSS, licensing, technology used in OSS , which can gives the determination of whether adapting environment has sufficient technical support of OSS in term of people, process and tools moreover exposing the highly profiled core developers of OSS can get more numbers of attraction from the user. Third phase is acceptability; we have identified a communications area for requirements, documentation, architecture and tools for OSS from the contributors. Community may have the methods to ensure the correctness and suitability of above. Next, for managing of appropriate stakeholders in the community we found the communication phase called Manageability.



Finally, in order to make the product as strong recommendation for procurement, the communication called easiness have established to expose the information related to portability, availability, licensing and understandability of code. Complete framework of OSS's Quality need to have four dimensions such as organized communication, project management, adaptability and user support. The organized communication is to ensure authorized and structured communication among developers, tester, user, and resources contributor since wherever huge people involved there need be a organized communication required otherwise its leads to serious deviation of community. Secondly project management is required as in any other traditional development organization, so this m Research did not concentrate on this dimension. As stated earlier OSS cannot have committed team to any of its processes such financial support, development, testing, user so OSS need to have strategies to retain its community members such as reward, award, etc. finally adaption of contribution is very cynosure of OSS quality and suitability of contribution. The OSS quality framework is shown fig.III. This investigated is to identify types of communication required in each phase of OSS development, rest of this paper discusses checklist to form communication are presented.

Fig. II. OSS – Specific Quality Model



**Functionality**

Functionality phase in organized communication performs the task of encouragement, and validate the code submitted by their volunteers. Contribution helps to the community to ameliorate quality in the following aspects:

1. Adding extra functionality to OSS can make the product to be alive long in the market.
2. Code contribution can help to community to measure the skills and knowledge of their volunteers since OSS has no way to scale the knowledge of their volunteers.
3. Enhance the functionality of existing service and replace previous code by new. But their should be a set of checklist to ensure the code contribution, which support and enhance the performs of present OSS.
4. Frequent release of OSS can force the user to change the system. Instead operations / functional –based release, which gives trustworthy to users.
5. Customization of any OSS may cost more to user, in case of code contribution that can be avoided.

Apart from all, there should be a checklist to accept the contribution like code, resource and technology. We are in the verge of completing such a checklist for various contributions in OSS community as shown in Fig 4. Similarly, for feedback and compliance, there should be some mechanism to analysis and respond to the feedback for which there are no answers in OSS documentation. OSS Community need not reply for compliance not directly related with product.

**Lean-ability**

Popularity of any product based on public opinion, community can expose their product in a way to meet their end-user expectation. Normally OSS-users can expect three things from community. Long-term support, unfortunately user cannot get support from one company. This gap overcome by exposing the strength of available service, licensing information, these can helps to user to predict their customization cost. Trustworthy can be improved by allowing user to know about technology used, high profiled developer in community.

**Acceptability**

Apart from Code contribution, some volunteers can contribute the requirement specification, which makes the user to inspire of product. OSS adaptability highly dependence on available documentation of architecture, development specification, and testing plan. Volunteers willing to contribute, these can be accepted under certain conditions after deep investigation since acceptance of new architecture and tools some time can lead the system to loose their customers.

**Maintainability**

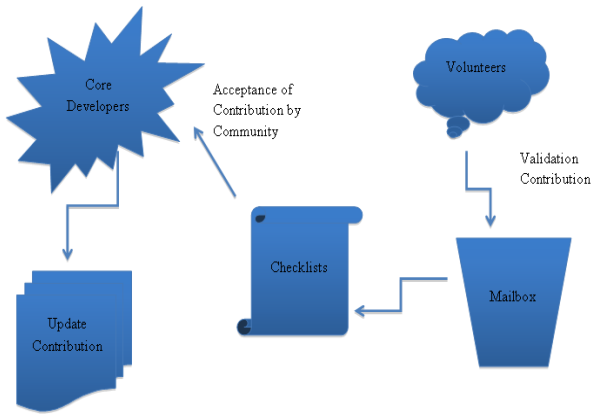
Software quality Assurance is an umbrella activity, in order to ensure quality on OSS product; it is necessary to know all information about people, process and tools involved in development. OSS originated with single user concept, to ensure the quality the SQA plan must be prepared to provide the required information to tester, developers and users which can improve and reduce the quality and cost of SQA respectively.

**Easiness**

This phase of communication provides mechanism to understand about OSS support-environment (Eg. OS Middleware) can allow the user to predict the ability of Product portability. The licensing for product makes the user to away from product. Clear information about licensing and code availability can allow users to determine the customization cost. Moreover all the above communication should have activity-based checklist to ensure the purpose and impact on product. “Communication is an Obligation “ OSS community cannot away from it. But organized communication can lead us to extreme success in market and support to community.



Fig III. Contribution Validation Process of OSS

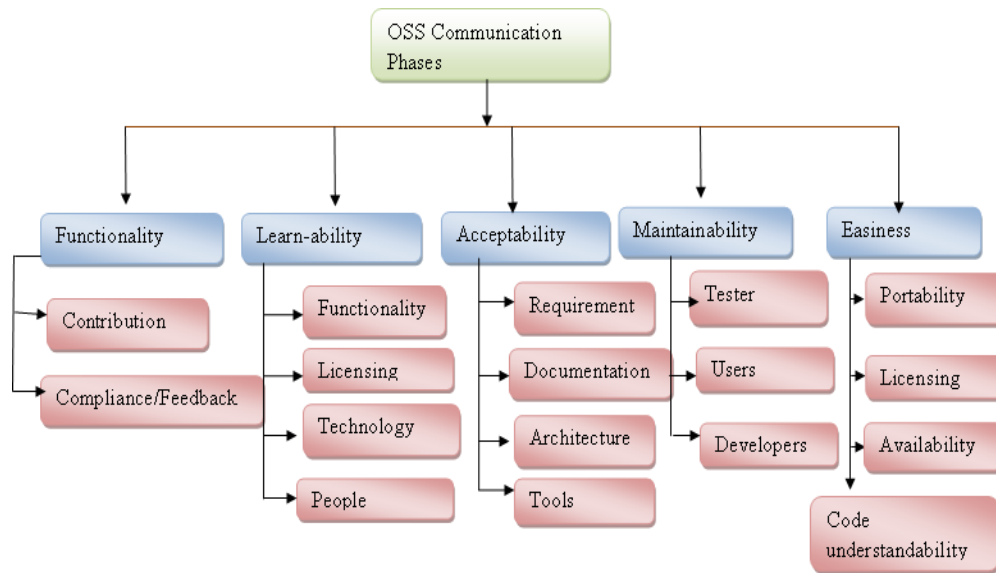


IV. CONCLUSION AND FUTURE WORK

Success of any software product depended on time to deliver and cost. These two things can be easily achieved by OSS. There should be systematic approaches in selection and development of product. The Expectations of adaptors are the regular technical support, Function-specific release, and Quality assurance with respect to user specification. In order to ameliorate the quality of OSS, the proper communication is important among community. Normally OSS has very vast community .we should have the plan to further Extent activity-based participation on OSS development. The

suggested a checklist for validation of contributors’ works in a timely manner. We discussed the process of quality requirement, quality factors and organized communication of OSS. These are very hard to achieve in software engineering. The importance of OSS-specific quality requirement also briefed. Ameliorate the quality of OSS can be achieved with this proposed model in open source communities. Proper communication always helps to community to rectify the defects in the beginning stage itself. Wherever huge number people involved there should be a method for communication. The quality of OSS depends on sustainability of its community. In this paper we have suggested organized communication in the phases of OSS development cycle since “quality of OSS not only concern with technical aspects but also concern with behavioral aspects”. Strong and well-structure approach of communication always help to retain the supporters (core developer, testers, sponsors and users) since community cannot force anyone to perform any particular work in OSS environment.

Fig. IV. Organized Communications in OSS



REFERENCES

1. . Mark Aberdour., Opensourcetesting.org, “Achieving Quality in Open Source Software”,IEEE Computer Society,2007.
2. .AtiehKhanjani, RizaSulaiman, “The Process of Quality Assurance Under Open Source Software development”, IEEE computer Society,2011.
3. RuedigerGlott, Arne-KristianGroven, Kristen Haaland, Anna Tannenberg, “Quality model for Free/Libre Open Source Software-towards the “Silver Buttet”?, IEEE,2010.
4. PerryDonham , “Ten Rules for Evaluating Open Source Software”,www.collaborative Consulting,2004.

5. TabiasOtte, RobetMoreton, Heinz D.Knonell, “Development of a Quality Assurance Framework for the Open Source development Model”, IEEE Computer Society, 2008.
6. AbdulRahman.A, “Guidelines for Software Quality Model Using Domain-Values, User and Organizational Perception”, International Journal of Communication and Engineering (IJCE), Volume5, Issue: 01, 2012.
7. HimaniGoyal and Er. JasbirKour, “Design Method of Open Source Software”, International Journal of Computing and Applications 6(1),2011,PP.75-82.

8. Vieri Del Bianco, Luigi Lavazza, SandroMorasca and DavideTaibi, "Quality of Open Source Software : the QualiPSo Trustworthiness Model", International Federation for Information Processing, 2009.,pp-199-212.
9. T.RMadanmohan and Rahul De, "Open Source Reuse in Commercial Firms", IEEE Computer Society,2004.
10. ChristophLattenmann, Stefan Stieglitz "Framework for Governance in Open Source Communities", IEEE , 2005, Proceeding of 38th Hawaii International Conference on System Science.
11. Won Jun Sung, JiHyeok Kim and Sung YulRhew., "A Quality Model for Open Source Software Selection", IEEE Computer Society, 2007.
12. Lect. Marius Popa., "Processes of Quality management in Open Source Software", Open Source Scientific Journal, Vol.1 No. 1,2009.
13. Pekka Maki-Asiala, Mari Matinlassi ., " Quality Assurance of Open Source Components: Integrator Point of View ",IEEE Computer Society,2006.
14. AnasTawileh, Omer Ranna, "Free and Open Source Software Quality Assurance", IEEE Computer Society, 2006.

### AUTHORS PROFILE



A. AbdulRahman. is working as Associate professor at Godavari Institute of Engineering & Technology, Rajahmundry, India. His current research interests are Knowledge Engineering, Open source Software Quality Assurance and Information Retrieval Systems. He has published many journals in reputed journals and conferences in the field of software engineering and knowledge engineering. His previous appointment include software engineer, Hyderabad.