

# Open Software Vs Commercial Software in Safety Critical System: A Review of Literature

Ibrahim Muhammad Hassan<sup>1</sup>, Ismaila Hassan<sup>2</sup>, Jamilu Abudullahi<sup>3</sup> Suleiman Bashir Abubakar<sup>4</sup>.

<sup>1</sup> Senior Lecturer, Department of Computer Science, Hussaini Adamu Federal Polytechnic, Kazaure Jigawa State. Nigeria. [Ibrahimhass003@gmail.com](mailto:Ibrahimhass003@gmail.com)

<sup>2</sup> Librarian I Haruna Wakili Library, Sule Lamido University, Kafin Hausa Jigawa State. Nigeria. [ismaihas@slu.edu.ng](mailto:ismaihas@slu.edu.ng)

<sup>3</sup> Lecturer I, Department of Computer Science, Jigawa State Institute of Information Technology, Kazaure Jigawa State. Nigeria. [@gmail.com](mailto:@gmail.com)

<sup>4</sup> Lecturer II, Department of Computer Science, Jigawa State Institute of Information Technology, Kazaure Jigawa State. Nigeria. [sbbabura@gmail.com](mailto:sbbabura@gmail.com)

**Abstract**— The use of safety critical systems in modern software is rapidly increasing due to the confidence the society on a wide range of systems that are needed to guarantee the safety of their users as well as their environment. SCS is potentially dangerous, so in general, the most common safety criteria are based on identifying the hazards and generate some requirements that reduce or eliminate the risk of hazards. This paper provides an introduction to SCS in relation to the development of Open source software. The term safety in relation to Open Source Software is required to choose if OSS can be used in critical context. This can be a drive from the terms of reliability, Security, and Flexibility. Open source Software(OSS) is defined as a software that its source codes are distributed together with the executable program and its free to use, it also consists of a licence that allows users and developers to study, make modifications and distribute the software(1, 2). The paper also aimed at presenting the Social, legal, Ethical, and professional issues in relation to OSS and Commercial/ propriety software. Proprietary software which sometimes refers to as commercial or closed source is a software that remains legal property of the individual, group or organization who created it. The organization that owns the legal right of the software usually do not release the source code to the general public but rather insisted that only those who purchase the licence key can be able to use it (3). Finally, this paper makes a comparative analysis between OSS and proprietary software.

**Index Terms**— Critical, Risk, Consequences, Safety, Proprietary, ethics, Autonomy

## I. 1 INTRODUCTION

In Safety Critical System (SCS) Modern Society has confidence on a wide range of systems that are needed to guarantee the safety of their users as well as their environment. The goal is to make the system safe. SCS is potentially dangerous, so in general, the most common safety criteria are based on identifying the hazards and generate some requirements that reduce or eliminate the risk of hazards. There are different types of definition attributed to SCS but the perceptive notions truly work well, reason is that both perceptive and formal are the consequences of failure, and if the failure of certain systems could result in consequences that are determined to be intolerable, the system refers to as safety critical(4).

The term safety in relation to Open Source Software is required to choose if OSS can be used in critical context. This can be a drive from the terms of reliability, maintainability, and availability(5). Open source software (OSS) is a software that normally denotes its distribution under the condition that complies with the OSS definition(6). OSS is a software that allows users to study, make modifications, and distributed to confer to a licence which was agreed based on the OSS definition. It also provides another approach to building a social relationships and software systems, that is a community-intensive and socio-technical (7). In addition, OSS programs are often, even though it not fully developed via join effort, in some

instances many people contribute some element to the final software. One important thing is that many software companies provide paid programmers time and program in the development of the in-house open source community. According to (8), software safety is a necessary quality attribute in determining the properties of certain classes of a system due to the impact it has on life or properties. So commercial software safety is important in dealing with the minimal threats or risk to the system and mitigation loss in the event of failure. Furthermore, closed source software refers to the model in which the software developed by the commercial entity is usually licenced for free to customers (either via channels or directly) in the object, executable code or binary. Training, support, update and related similar services necessary by the customers to effectively and efficiently used that software which is often provided by the commercial entity. Here the software source code might be made available to certain users of software via agreement. The closed source software are not distributed to the general public like that of OSS and also might not be modified, copied except based on agreements(9).

## 1.2 Safety Critical Systems

Safety critical system should be analysed and undergo some check carefully, each potential failure, error detected should be

given considerations make an evaluation before the releases of the new products of the software.

**1.2.1 Reliability:**

Is a loose term which refers to the absence of defects that cause in correct operations, the sudden failure or data loss. Many people uses bug, which refers to as failure to specification; However, in OSS project its difficult to point a better way of defining bug and its feature. But in determine what constitute bug was agreed between developers and the users of the software, failure to perform that, is the same the failure to conform standard(7). Severe defect can be fixed within short period of time after being detected to way which support the source code availability. Developer that detect bug should be able to fixed it and then report to the maintainers to come out with the up to date version of the software at their own power. But in the case of CSS, the report need to be filed which will lead to a delay before the vendor cam decide when or weither the issue of updated version releases. Based on the market value OSS model is robust and found practice by Linux especially at the start of the stage of development.

**1.2.2 Security:**

It has been argued that OSS are more secure than CSS. Security of software is concerned with vulnerabilities which affect a complex software programs. How to fixed or minimized the vulnerabilities, attack remain critical considered in the users mind. So, OSS and CSS the security offering varies drastically as well as the designed of the secure features. So in total security relies on how good the software was deployed, configured, maintained and updated including the finding the vulnerabilities and fixing them. Products that are poorly maintained offers a little security irrespective of the deployment model or testes of the software's. Finally, security of either OSS or CSS implementation is not a determining factor method of development or distribution. For safety reason, the software should be secure before deployed.

**1.2.3 Flexibility:**

It was discovered that OSS are more flexible for users that CSS, this stems the capacity for users to cross examine the codes, and alter the code where necessary. This allows the user to potentially identify any problem in the system to make changes to rectify the problem and fixed it. The patches and update for OSS comes from different sources. This flexibility allows users based on the safety for different users to modify the source that makes it so flexible. Finally, in relation to safety critical, the changes made by users or developers undergoes rigorous check before released by the developers for safety reason.

**1.3 Open Source Components of Safety Critical System:**

This work introduces briefly a sort of Safety critical system in OSS that has been used in Linux, which is the operating system used as OSS that introduces from its Quality assurance. In QA point of view, NASA practice uses OSS (Linux). One of the most adopted and used OSS in safety critical application is Linux (klunicas and Yovine 3002; broggi & fascioli 199; Ortega 1998) the only OSS found to be used

affecting the safety functions is Linux both Linux real-time and Linux.

**1.3.1 Linux safety critical Application**

A preliminary assessment of Linux for SCS has been done by Healthy and Safety Executives (pierce 2002) which they suggested that:

- OS Should be clearly understood and have a good characteristic for Safety Critical Application.
- Should be reliable.
- The provision of high resolution display in Linux leads to the provision of:
  - Automatic train control display system
  - Railway display system
  - Embedded system etc.

**1.4 The problem of OSS:**

One of the most important problems associated with OSS is that many individuals participated in the development process of open source, which is undertake a rigorous review and involve no appropriate management processes and version control. The user/customers that installed the open source programs do not have real knowledge of those software creators are they still existed in the same form as when earlier designed and tested. So in general lack of control in the development of the program is a serious problem to the customer confident in the product or software.

**1.5 Differences between Open source and Closed or proprietary software**

s/n		Open Source Software	Proprietary /Close Source Software
1	Licence	Users have the right to Study, Copy, Share and modify the products as well as they are licensees	Users here are unable to Copy or modify and share the product. Users are licensees
2	Product Cost	Free	Not Free
3	Example of the Software	Mozilla Firefox, Google Chrome, Linux, MySQL, Python and GIMP	Internet Explorer, Windows, iOS, iTunes etc.
4	Vendor lock in	Here users were able to switch to some other alternative OS product or can modify the software	User depends on vendor in order to produce update and support to the product
5	Improvements and Updates	Involve the community contributors that has a wide range of expertise which can contribute to the continuous of the improvement of the products	Update cycle and development of new features are responsible for vendors completely.

### 1.6 Licence and legal issues:

From the legal point of view, open source results to distribution right and obligations in software licence agreement in a manner that they would be:

- Both of user control over the source
- Facilitate sharing

The open source concept here is reversing the process to result in proprietary licence and equalizing the right of the users with that of authors. According to the Moglen said open source are quite protected by law and all parties depend on such licence as in GNU, GP even though the two were never been tested by the court. However, there are different types of open source licence which vary in their legal requirements. The most common includes:

- **General Public Licence (GPL):** this gives a guarantee of the right to the developers as well as the users to legally distribute, copy modify as they wish to under certain conditions For Example GPL put restrictions that any copy should be loyalty free and give certain condition also for the distributions.
- **Lesser General Licence (LGPL):** this gives few right to work, what it provides is the combination some elements of new commercial software and Open source application, without making all the application available the public

Other include Dual licence and Berkeley standard distribution licence (BSD):(3).

The commercial software: when you purchase a software you acquire a licence to make use of it but you do not own the software. So licence of the software is stipulated by the following:

The software is covered by the copy right even though you can make one archival copy of the software. The copy of the backup cannot be used unless the original fails or damaged. It's not allowed to modify the software without permission of the holder. Development of the new work without copyright holder is not allowed.

In trying to break copy right or licence is a criminal offence and is actionable by the civil law, the relevant act and is also an offence under the fraud act 2006 can be applied.

### 1.7 Social issues

One of the important social issues in OSS is directed towards the needs of its developers as well as the users, rather than the program, company that originally wrote it. Compatibility is one among many issues, that Oss gives social benefits to a wider perspective of society than the closed Source software(10). There are many different social reasons that OSS is more compatible than the closed source. For example, in respect to any software, there are programmers and users, programmers designed, deployed and implement the software or programs internal as well as the user interface. The software should be tested before release, however many users uses the software programs without really understanding its internals, so when they found a bug they report to the programmers.

The relationship that exists between programmers and users are reciprocal, again programmers designed the software for users and users suggested some features to the programmers.

Finally, CSS development encourages partition of the above mentioned groups. So there is an obstacle between programmers and users, CSS Company is public relation and marketing department.

### 1.8 Ethical Issues:

The ethical issues explore in the context of OSS include Anatomy, Software Quality and Accountability and looked in to some analysis weather OSS can be rated as good.

- **Software Quality:** - This shows that the software meets its requirement specifications which are well listed, maintained and documented(11).The supporters of OSS claimed that users and developers were motivated in order to produce a qualitative work. Because not only developers for their own use but their own reputation will be at a stake. Those that criticise OSS claimed that professional quality work cannot be produced by volunteers since there is no involvement of monetary compensation, as well as maintenance and documentation, are nonexistence. For example books on Linux was found everywhere, the question d either OSS is lower or higher quality and compatible. CSS is empirical rather than philosophical (12). One of the distinct between OSS and CSS project is a lack of date of releases, OSS development is expected to have frequent releases, and there are no release deadlines. However, CSS vendors mount pressure on the developers to customers, which in return increases the chance of making an error in the software.

- **Accountability:** In this case, we have to consider four cited barriers to accountability:
  1. Problems of Many hands
  2. Computer as scapegoat
  3. Ownership without liability
  4. Bugs

The above mentioned barriers can result in harm and risk for which no one is answerable and which nothing is done (13). To look in the OSS which address the barriers 1 and 4, Number 2 is a general issue and 3 does not apply to due to the software lack of ownership in OSS. The author argues that allowing bugs as a software fact of life has issued regarding the accountability. The OSS argument is that software development treated bugs in a group effort to find or detect and fix a problem. According to the Raymond 2002 says individual that found a bug may never fix it since the developers examine the OSS codes, so when bugs are found they can be corrected quicker than the development effort which should it only for programmers that see the codes. So the group Accountability is not lost but instead lifted up by the group.

- **Autonomy:** developers that take OSS to do gain at the extent of autonomy might not available to developers in the commercial company's software. Developers in OSS

work as volunteers and can quit or joined at any time as effort strictly on their own creativity. So they are volunteers that are not coerced in to contribute willingly and participation. In OSS the contribution is in certainty because of the developer is not sure whether his contribution would be accepted in a canonical version that is constantly evolving. So the contribution may be accepted or rejected, even if accepted may later be replaced or change. Finally, Oss have elaborated culture but with a wide unadmitted set of ownership customs.

11. Schach SR. Object-oriented and classical software engineering: McGraw-Hill New York; 2002.
12. Kollock P. The economies of online cooperation. Communities in cyberspace. 1999;220.
13. Nissenbaum H. Computing and accountability. Communications of the ACM. 1994;37(1):72-81.

## 1.9 CONCLUSIONS

The paper aimed at introducing the safety critical system with an overview of its applicability in relation to the general topic. Oss offers much opportunities interims of security, reliability as well as flexibility for making user to modify the software without any restrictions. Some of the weakness observed during this research in OSS is lack of control in the development of the software and the paper also described detail of Social, Ethical and Legal issues and focus more on the OSS than The CSS. Among the strength of OSS is free in terms of acquisition, licence free and cost. However, security and quality of the software is regarded as good.

## 1.10 References

1. Hansen M, Köhntopp K, Pfitzmann A. The Open Source approach—opportunities and limitations with respect to security and privacy. Computers & Security. 2002;21(5):461-71.
2. Feller J, Fitzgerald B. Understanding Open Source Software Development Addison-Wesley. London, UK. 2002.
3. Lemyre P-P, Willemant R. The Legal Issues Surrounding Free and Open Source Software: Challenges and Solutions for the Government of Québec: CIRANO; 2006.
4. Knight JC, editor Safety critical systems: challenges and directions2002: IEEE.
5. Pierce RH. Preliminary Assessment of Linux for safety related systems: HSE Books; 2002.
6. Feller J, Fitzgerald B. Understanding open source software development: Addison-Wesley London; 2002.
7. Scacchi W, editor The future of research in free/open source software development2010: ACM.
8. Gary K, Enquobahrie A, Ibanez L, Cheng P, Yaniv Z, Cleary K, et al. Agile methods for open source safety-critical software. Software: Practice and Experience. 2011;41(9):945-62.
9. Seow Hiong G. Open source and commercial software. An in-depth analysis of the issues. Business Software Alliance. 2005:1-32.
10. Khanjani A, Sulaiman R, editors. The aspects of choosing open source versus closed source2011: IEEE.

IJSER