Open Source and Closed Source Approach: A comparison

Asad Noman

University Of Sargodha, Mianwali Campus

September, 2019

1 Introduction

Open source software development has acquired much of the world's attention in a couple decades. It has always been an object of controversy in comparison to traditional (closed source) software development. Both approaches have struggled to outperform each other in a marathon that has continued for decades and will supposedly continue for another. It is to keep in mind that open source approach by definition dates back to the beginning of the field of computing (Glass, 2004). In this paper, a comprehensively discussed comparison between the two approaches, namely open source and closed source, have been presented. The paper tries to bring forward the differences which draw the line between these two approaches and analyses their progress in the marathon that they are competing.

Section 2 offers literature review. Section 3 defines the terms and presents an overview to build the foundation before moving on to the actual discussion. Section 4 outlines in details highlighted key differences. Each subsection in section 4 presents the contrast regarding that specific aspect between the two approaches under discussion. After discussing the differences, section 5 outline the strengths of open source approach that have been discussed so far and why should it matter which is then followed by Section 6 describing the market of two important open source projects that are ruling the digital world. This section also provides statical data to strengthen its point which is followed by section 7 describing the future of open source.

2 Related work

There has been many important studies in the past that presented comparison between open source and closed source approaches, some of which dedicate entire study to address one single aspect more comprehensively. (Lerner & Tirole, 2002), (Schryen & Kadura, 2009) and (Hertel, Niedner & Herrmann, 2003) solely dedicate their studies to address economy, security and motivation towards open source respectively. An equally important study (Sood, Shipra & Soni, 2016) covered this debate rather briefly.

3 Defining open source and closed source approach

In open source software development, the source code of the program is made publically available. Availability of source code is a basic requirement for open source project but this is not it. (Open Source Initiative, 2007) specifies ten criteria for a project to be open source. Though their criteria only covers economic and legal aspects, it is internationally recognized as a standard definition. Out of those ten criteria, three main criteria assert:

- Free distribution of software
- Availability of its source code
- The right to modify

All of those three criteria sum up into **"use, study, improve and share"**. The rest of the criteria, in short, specify that anyone in any field for any purpose may use it. A key principle of open source development is decentralized development. This model

encourages open collaboration. Take Linux Kernel as an example. Linux kernel has 20,000+ contributors from all over the world contributing to its code, documentation and testing etc.

Unlike open source, closed source softwares are proprietary softwares distributed under a licensing agreement that restricts access to source code except for original authors. Following are some highlights that differ in comparison for the reason:

- Closed source software is not free of cost (unless it is freeware).
- Both closed source and open source software users need support and documentation (provided by service teams or online communities).
- Learning is more encourged in open source, which leads to innovation. Closed source projects have to keep in mind their corporate ties.
- The question of security is controversial and has been given a bigger deal of attention in upcoming sections.

4 Key differences between closed source and open source approach

Although this paper in its entirety, is a comparison between open source and closed source approaches, an outline of briefly discussed key differences has been given in this section.

Cost

Proprietay softwares can cost a couple hundreds to few thousand dollars. Proprietary software is like any product coming from a known brand. On the other hand, open source softwares are free of cost ninety nine percent of the time. There are no subscription or activation charges.

A few propreitary softwares offer free trail version for specified number of days. Users are able to use the software only for the days specified and they will have to purchase it if they want it to be functional after the trail period. Freewares on the other hand do not ask for purchases as they are distributed free of cost. However most of the time, freeware proprietary software developers reserve place for advertisments on the user interface. Doing this funds them per view and per click monetarily. A common example of such software is EASEUS Utilities and Bittorrent.
Of course open source software developers do not employ such stratergies. There are many open source softwares that do not differ or even outperform their competent proprietary software and still cost nothing and show no ads.

Support

Proprietary software providers offer ongoing support and thereforer are advantageous. Users can contact service providers for guidance and troubleshooting. On the other hand, open source softwares are highly documented, which is part of open source development model. In addition to this, open source softwares rely on online communities and forums to deliver support.
It is to be noted that a pre-documented and answered solution to a problem is rather quick and easy to find than contacting support. Microsoft has, in ot realization of this

fact, open sourced there documentation (Microsoft Docs) on github to which many contributors have started contributing in a couple years. The number now exceeds a few thousand.

Innovation

Viewing source code for proprietary software is not permitted. Proprietary software provides only provides the executable (binary) files and necessary meta data. Open source softwares by definition get their source code properly publicized and anyone can access it. Whether witholding source code from users promise more security or not is discussed in the upcoming subsections.
Innovation in proprietary software development is centralized. They innovate by sharing ideas with the team and deciding which ones to implement. They also innovate by acquiring knowledge about their users, that is, feedback from users through online serveys and forums and opted telemetry (tracking users activity in the software to gather information of common interest between users). Telemetry however is not seen as a positive strategy.
Open source projects are more open to ideas and therefore to innovation. Developers are subset of users community. Freedom and flexibility (decentralization) given open source projects more opportunity to adapt, evolve and innovate. Forked variations of the same software exist offering different taste for different users.

Freedom

Open source software give its users to do anything they want. Users are free to use it the way they like without having to worry about tracking and telemetry.
Closed source software is an opposite to open source policy. As described earlier, proprietary software mostly opt in telemetry and tracking to know what users are doing, mainly for the purpose to ensure better version by acquiring knowledge about how their users use their software.

Security

Proprietary software providers do not permit viewing the source code. It can be thought to be advantageous because not release source code bring security by hiding loopholes even if they exist and therefore preventing hackers from breaching them. With talent of today's world in view, it is highly unlikely. (CVE Details, 2019) presents an exhaustive list of security vulnerabilities in both open source and proprietary softwares (see Figure 1). It is clear that loopholes will be found even if source code is never publicized. Open source communities utilize their decentralization to detect loopholes and vunerabilities as soon as the source code is publicized. Those vulnerabities are most of the time detected and supressed rather quickly.
For the final verdict, there is more room for validation and revision in open source projects. Proprietary softwares are viewed to be less secure compared to open source software except for the total solutions.

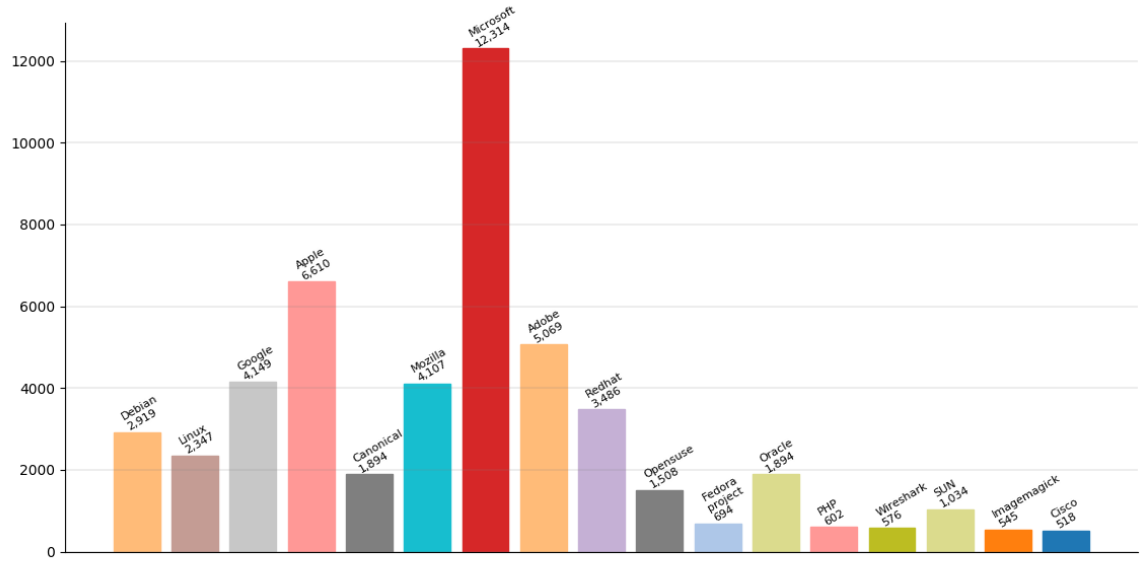

Figure 1: Total Number Of Vulnerabilities Of Top 50 Products By Vendor on average (CVE Details, 2019)

Usability

Proprietary softwares employ hardcore usability testing to a targeted audience. Usability is one of the aspect that rank proprietary software higher.
Open source communities have, unlike their past, developed usability testing strategies.

Availability

Open source software is openly publicized on the internet and any person with internet access can access the software.
Proprietary software providers make their product available as one of three categories which are paid, shareware and freeware.

Transparency

Transparency is synonymous to open. Open source approach offers its users transprency. Closed source approach, on the other hand, keeps whatever is under the hood, under the hood. Moreover, proprietary software licenses clearly state the restriction to reverse engineer the software.

Reliability

Reliability of a proprietary software totally depends on its providers. The more reliable the brand is, the comfortable their users feel.
In open source approach, a software project only lives if it is reliable. Communities thoroughly feedback any open source software to make other users aware of its issues. If an open source software is not reliable, it can not possibly stand all this opposition from

the community and therefore either manages to become better and gain its community trust back or goes extinct.

5 Strengths of open source and why should it be a choice

Two important reasons for open source development model are hightlighted by (Kogut & Metiu, 2001) which are "it avoids the inefficiencies of a strong intellectual property regime and it implements concurrently design and testing of software modules". They go further to assert the definition for open source, "the intellectual property rights to software code is deposited in the public domain, and hence the code can be used and changed without requiring a user fee, such as the purchase of a license". (Bonaccorsi & Rossi, 2003) distinguish open source from traditional closed source as "an important process innovation in the software production process". The term "free" when used in the context of open source software carries two meanings, "free of cost" and "free to do with software or the code anything a person wishes".

Open source project communities tend to have many developers as subset of overall users (Gacek & Arief, 2004). Bug fixes, broken code etc. are fixed in a comparatively shorter period of time. Open source project community decisions are based on hierarchical consensus (more decision power to the core developers) or full concensus (decision power is totally decentralized). In open source project communities, transition from ordinary developers to core developers is based totally on merit (Gacek & Arief, 2004). This results in innovation, quicker development and troubleshooting. Open source software provide fully publicized source code. This transparancy helps in learning and innovation, bugs are detected and fixed rather quickly. Open source projects are supervised by people and not by corporate agendas. The open source project teams do not contribute to the project for monetary return. They work for the feeling of self-worth that it brings to successfully develop and maintain a project.

6 Market

The world has witnessed an unrivalled growth of open source software in past three decades. Open source communities have withstood all the critique and competition from other corporations. Open source projects have grown the number of contributors from few to a couple thousand. This section presents some statistical data of the two most popular open source projects Apache Web Server and Linux.

Apache Web Server

In the great web server marathon, Apache server despite its notable comptetors (Microsoft IIS and Netscape), powered 60% of the internet in November 2000, thrice as much as Microsoft ISS(20%) (Lerner & Tirole, 2002), and powered 71% of the internet in Novemeber 2005, 3.5 times as much as did Microsoft (20%). Apache server has had more users than its rival in a span of consecutive 20 years (1996 to 2016). Microsoft ruled the majority for 3 consecutive years and Apache reacquired its height a couple months ago in March 2019. The most recent statistics show that Nginx (pronounced engine X, yet another open source web server) and Apache lead the power on the internet (NetCraft, 2019).

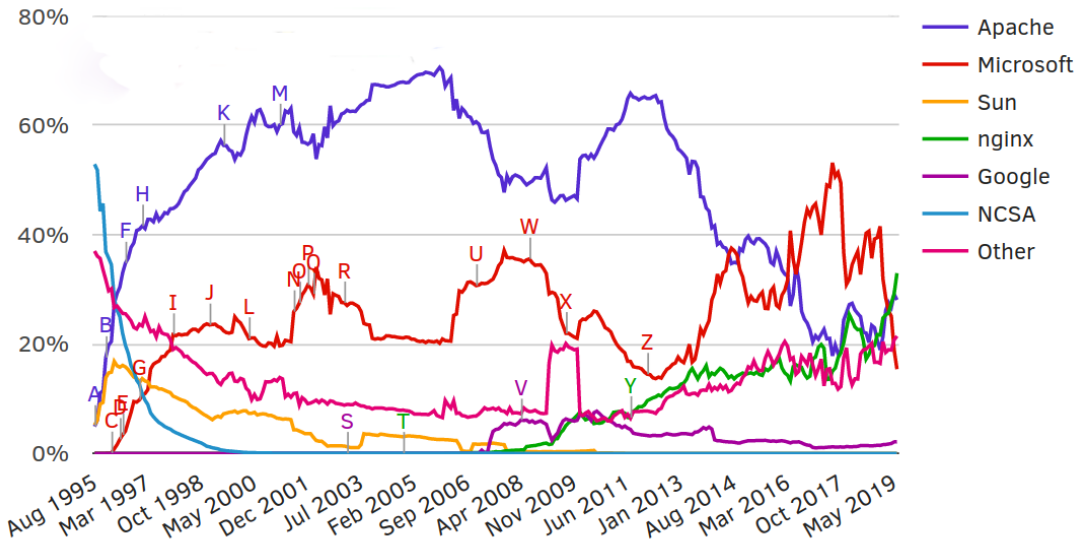

Figure 2: Web Servers

Linux

Linux is considered a significant competitor to Microsoft Windows operation system. Although Linux does not power much of the desktop and laptop computers (2.06%) it leads the computer market in other areas. Top 500 fastest supercomputers are powered by Linux (TOP500, 2019). 73.19% of the smartphone and tablet computers are powered by Android which uses a modified version of Linux and therefore is classified in Linux family (StatCounter, 2018). Linux powers 66.6% of the web servers, twice as much as does Microsoft Windows Server (33.5%) (W3Techs, 2019). Linux has also developed its number in embedded systems which is 29.44% (Global Market Insights, 2018).

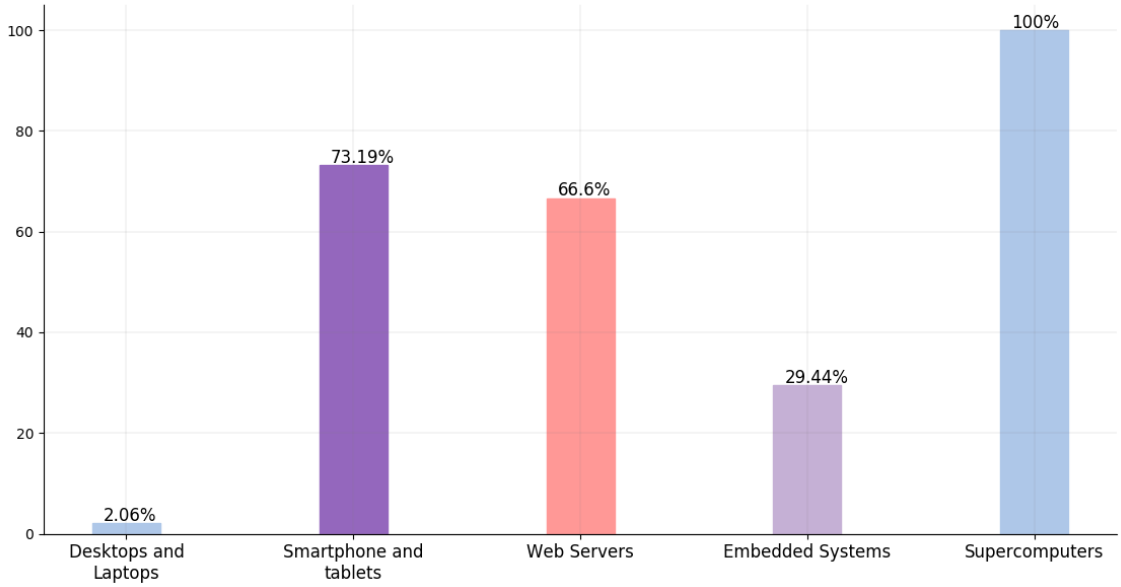

Figure 3: Areas powered by Linux

7 The future of open source

The future of open source approach looks very promising. With the edvent of platforms such as github and gitlab etc. the essence of open source projects, that is decentralization, is more empowered and encouraged. These platforms efficiently provide the mechanism, space and tools to collaboratively develop projects ranging from small (fewer contributors) to huge (thousands of contributors). The number of users on these platforms has been going up since day one. In last couple years, some well-known corporations have expressed their interest in open source (Lerner & Tirole, 2001). Microsoft, for instance, open sourced their documentation (Microsoft Docs) and Powershell and developed a very flexible and extensible open source text editor called Visual Studio Code.

References

Bonaccorsi, A. & Rossi, C. (2003). Why open source software can succeed. *Research Policy*, *32*(7), 1243–1258. Open Source Software Development. doi:https://doi.org/10.1016/S0048-7333(03)00051-9

CVE Details. (2019). Total number of vulnerabilities of top 50 products by vendor. Retrieved from https://www.cvedetails.com/top-50-products.php?year=0

Gacek, C. & Arief, B. (2004). The many meanings of open source. *IEEE software*, *21*(1), 34–40.

Glass, R. L. (2004). A look at the economics of open source. *Communications of the ACM*, *47*(2), 25–27.

Global Market Insights. (2018). Embedded software market share. Retrieved from https://www.gminsights.com/industry-analysis/embedded-software-market

Hertel, G., Niedner, S. & Herrmann, S. (2003). Motivation of software developers in open source projects: An internet-based survey of contributors to the linux kernel. *Research policy*, *32*(7), 1159–1177.

Kogut, B. & Metiu, A. (2001). Open-Source Software Development and Distributed Innovation. *Oxford Review of Economic Policy*, *17*(2), 248–264. doi:10.1093/oxrep/17.2.248. eprint: http://oup.prod.sis.lan/oxrep/article-pdf/17/2/248/963941/170248.pdf

Lerner, J. & Tirole, J. (2001). The open source movement: Key research questions. *European economic review*, *45*(4-6), 819–826.

Lerner, J. & Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics*, *50*(2), 197–234.

NetCraft. (2019). August 2019 web server survey. Retrieved from https://news.netcraft.com/archives/2019/08/15/august-2019-web-server-survey.html

Open Source Initiative. (2007). Open source definition. Retrieved from https://opensource.org/osd

Schryen, G. & Kadura, R. (2009). Open source vs. closed source software: Towards measuring security. In *Proceedings of the 2009 acm symposium on applied computing* (pp. 2016–2023). ACM.

Sood, G., Shipra & Soni, R. (2016). Comparative study: Proprietary software vs. open source software.

StatCounter. (2018). Mobile and tablet operating system market share worldwide. Retrieved from https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide/#monthly-201801-201801-bar

TOP500. (2019). Top500. Retrieved from https://www.top500.org/lists/2019/06/

W3Techs. (2019). Usage of operating systems for websites. Retrieved from https://w3techs.com/technologies/overview/operating_system/all