IV064

Information Society

Open access to source code, and results, open systems

Adrián Tóth (491322)

November 10, 2019

Contents

1 The Open Access

Nowadays, the open access term is used in many ways. Firstly, let us declare the meaning of this expression within the context of IT to better understand other things that will be mentioned later. You may heard phrases like '*Open access to source code*' as well as '*Open source software*'. Accordingly, this phrases may also appear under the '*free*' acronym, which is not the proper notion. There is an article [17] about the contrast of these two terms, which describes and explains the differences between them. All of these names indicates one common attribute (model, philosophy or methodology) of the software development - the right of free access to the source code of a software to anyone known as the open source philosophy. This right stands for the free way of software source code free usage, inspection, modification and distribution that may be restricted by further open source licenses such as GPL – GNU General Public License, which will be described later. To clarify the veritable meaning of the open source let us specify the correct definition itself - the open source definition.

Definition 1.1. [3, 14, 20]
Open source does not just mean access to the source code. The distribution terms of an open source software must comply with the following criteria:

- Free Redistribution
- Inclusion of Source Code
- Inclusion of Derived Works
- Integrity of The Author's Source Code
- No Discrimination Against Persons or Groups
- No Discrimination Against Fields of Endeavor
- Distribution of License
- License Must Not Be Specific to a Product
- License Must Not Restrict Other Software
- License Must Be Technology–Neutral

As has been noted above, the definition is denoting a specific software type which is made freely available with respect to modification and distribution to anyone including license restrictions and non-discriminatory rules. At the present time, this influencing fact has affected not just the software itself but the development as well. Open source projects, products, or initiatives embrace and celebrate principles of open exchange, collaborative participation, rapid prototyping, transparency, meritocracy, and community-oriented development [16]. The strongest side of the open source is its community together with the collaboration power. The open source community is diverse and highly motivated [13].

Related to open access, you might probably notice abbreviation such as *FOSS* or *FLOSS*. *FOSS* stands for *Free and open–source software* while *FLOSS* means *Free/Libre and Open*

Source Software. These software projects are a form of commons where individuals work collectively to produce software that is a public, rather than a private, good [5].

At the present time, there is a countless amount of open source projects. Many of them become popular quite quickly and has drawn the interest of academia and industry. For students open source software is an arena for learning, and the industry needs software engineers acquainted with the theoretical and practical aspects of open source software development [9]. Several projects are running for decades, some of them are just at the beginning of their dawn. As an illustration, the Table 1 below contains some of the well known and still running projects under the terms of open source.

Project	Year	Author(s)
Unix OS	1969/1970	Ken Thompson, Dennis Ritchie & others
GNU[1] Project	1983	Richard Stallman
Linux Kernel	1991	Linus Torvalds

Table 1: The oldest and still existing open source projects [8, 7, 21].

In contrast to the open source, its antonym is known as proprietary sometimes referred as non-free. In conclusion, it indicates copyright restrictions that prevent unrestricted distribution or reuse of the software. The pros and cons of these two types of software philosophy will be explained later in the following sections.

2   Success of Open Source Projects

In the face of the fact, that nowadays it is very hard to be successful without any reward for the given effort, open source projects have their characteristic attribute that they are invincible in this field. The roots of success lie in the kindness and helpfulness of people who are developing software for the others as a hobby and for free. These developers, representing and formulating the community, are the crucial heroes for the success. As open source continues to prosper, the topic *'why is open source successful'* becomes the subject of more and more academic research [2].

2.1   The Open Source Community

Who and what is the open source community? Sometimes the community is referred as anybody, sometimes it is restricted to the group of developers only. It depends, in some cases from the license itself too. We may consider, that the community is a group of people who are interested in the open source software, not just the developers of that software. Thereby, anybody related or connected to the usage, development, improvement or enhancement of the open source software can be considered as a member of the open source community. Because of the increasing interest for a software by absolutely anyone, the community has uncommonly

fast grow, which has a positive affect on the software itself.

The community is strong, truly powerful. Together, collaboratively the community members are co-creating a single masterpiece of work, while they share their own intellectual properties as a subparts of the overall outcome. Many people are not just enhancing the pure software, but also their own skills. Learning new, yet unknown, things from the others can be considered as a benefit of being an open source community member. Do not forget to note that the community is worldwide in nature, so projects with larger scale are developed internationally such as projects listed in Table 1.

2.2   Source Code Availability and Hosting

To allow collaboration and contribution it is necessary to share the source code in some way as it is stated in the open source definition. In most cases these open source projects are using version control repositories or they may use something else, which are publicly hosted to make it available. Available version control repository hostings such as GitHub[2], GitLab[3] or Bitbucket[4] are the most common ones. Their advantage is not only the management of change in the source code but also saving metadata such as related author, date of change and so on. Metadata might include information such as source code location, contributors, license, references and how to cite the software [11]. Using a properly configured version control repository with open access allows to save the derived works, authors of each modification and further requirements easily.

2.3   Collaboration and Contribution

We have already described the open source community and the availability of the source code and its derivations in the sections above where we also slightly mentioned the community contribution. Now, let us take a closer look at the pure contribution and how it is performed by the members of the community.

To increase the intelligibility, many of open source project are using common conventions developed by the time. Many open source best practices fly in the face of traditional software development methods [10]. There are many books[5], articles[6] and guides[7] about the best practices on how to make better and easier collaboration and reduce mistakes by preventing common faults and errors. The strategy of open source development is still evolving by the time. Nowadays, due to the increase of the contribution count, software releases are faster and faster. The proprietary software projects will never reach the same speed of release announcement as the open source projects have.

---

[2]github.com

[3]gitlab.com

[4]bitbucket.org

[5]As an illustration, please see www.sciencedirect.com/book/9781555583200/open-source-software.

[6]As an illustration, please see martinfowler.com/bliki/OpenSourceResearch.html.

[7]As an illustration, please see opensource.guide.

2.4 Feedback and Support

Nearly all of the open source projects have a proper description where you can find the authors and guides how to contribute and further information. The most common type of feedback from the users are bug reports or feature requests. Based on the fact that the project is built by motivated individuals, reply for the given feedback is pretty quick. Feedback is not the only one received by the open source projects. Sometimes, if the project is in use and it is popular among larger companies, it has sponsorships. Nowadays, to become a sponsor is pretty simple while the hosting services support this feature. On 23rd May 2019 *GitHub* announced the new way of contribution – sponsors[8], which is a huge motivation for the project developers while they are awarded for the given effort and work they have done.

3 Open Source Consequences

Engineers using free and open source software created many of today's most innovative products and solutions [4]. Today, many open source products are among the market leaders in their field, both visible and invisible to users [19]. The software development has completely changed and also, the most remarkable fact is that a lot of proprietary softwares are becoming open source. Linux, Firefox, Android, Apache and others are nowadays celebrating the community-led development as well as being success as an open source software.

3.1 Influencing Impact of Open Source

As the open source took root in the software development, we may consider this breakthrough as a beginning of a new era. Many foundations were founded on the basis of open source to accelerate its development. Currently there are many foundations, for example *The Linux Foundation*, *Apache Software Foundation*, *Mozilla Foundation*, *Eclipse Foundation*, *GNOME Foundation* and others, helping out the open source community. The Linux Foundation is a well-known foundation that was created relatively early. Founded in 2000, the Linux Foundation provides unparalleled support for open source communities through financial and intellectual resources, infrastructure, services, events, and training [22]. The Linux Foundation and its projects form the most ambitious and successful investment in the creation of shared technology[22].

FOSS has dramatically influenced its alternative – the proprietary software. Most compelling evidence is the Microsoft company and its attitude of proprietary software enforcement. Sooner or later, hopefully everything changed to better. There is an article [15] describing the war between FOSS and proprietary software related to the Microsoft Corporation and its meaningless denial of FOSS. Microsoft's attacks are a classic example of what industry old-timers call a FUD (fear, uncertainty, and doubt) campaign [15].

---

[8]github.blog/2019-05-23-announcing-github-sponsors-a-new-way-to-contribute-to-open-source

In contrast, these days, the most popular companies such as Google Inc., IBM, Red Hat, Inc. are based on the FOSS. All three companies have a completely different approach to the open source unlike Microsoft Corporation. The did not deny the ideology of open source, they accepted and adopted it nevertheless they knew the possible risks and threats. Meantime, this step was crucial to the success. But how to adapt this methodology or how to build an open source company? This is a commonly asked very though question. There is a book *The Open Organization* [23] about how is it possible to be a completely open source successful company written by Jim Whitehurst – chief executive officer (CEO) of Red Hat, Inc.

As a result of being open, here comes the question about the copyright and intellectual property. In 2001, a Microsoft executive publicly stated that open source is an intellectual property destroyer [12]. This has led to the invention of various open source licenses that are a matter of course at the present time. These licenses are slightly the same, except on few restrictions.

3.2 Benefits, Advantages and Assets

On the positive side, there is the most significant fact about the open source which is the software price – it is for free, what is the key factor. There is no better price for a software. The paid price for the software is the adopting process to the custom environment.

As the license states the rules and restrictions, sometimes the FOSS may be privatized with respect to the license and charged for additional enhancements. As a perfect example, we can consider RHEL – *Red Hat Enterprise Linux*. Anybody can build and customize something on the top of open source projects by accepting the license terms.

Benefits of an open source project are limitless. As the project is available to anybody world wide, the software can be enhanced and improved by anyone. Many times a from scratch project gets involved in a dramatic development by the open source community and the results are magnificent. If we take a closer look to the *Linux* itself, which was created as a hobby project, it has expanded to enormous size and its development become world wide while it has beaten up all of its alternatives, rivals and opponents.

3.3 Threats, Weaknesses and Vulnerabilities

The software is not tailor-made to meet all customer requirements while the customer could be anyone. To correct the missing parts of the software, it may be done by user or in most cases it is done by the developers of the software after that it was reported. Proprietary software offers support where the further problems are reported. In case of open source software, there is no real support at all, which can be very inconvenient. The help is provided only in case when the developers are available and the most important thing, the user is able to contact them in some way. While considering the fact that nobody is obliged to help, this may be sometimes very nuisance.

Adopting open source software components offers many advantages to organizations but also introduces risks related to the intrinsic fluidity of the open source software development projects [6]. Organization must be fully aware of the fact that open source project development may stop anytime in case of a project with smaller size. The cause of stop might be due to the fact that the developers moved to new project and the project was abandoned, sometimes marked as orphaned, or due to different reason of any kind.

The security itself may be considered as cons of open source as well. Based on the fact the the software is developed in a non controlled environment, there are lot of potential imperfections that may lead to a security incident. The software is tested on the same environment as the developers are working. What if a developer has malicious intentions? This type of developer can integrate anything harmful to the software which can be exploited. Trustworthiness of the software like this is not sufficient. There has been extensive work done towards understanding and formalizing trust: trust is largely built from experience [18].

4   The Open Systems

As a result, as the open source adapted, an open system as a subsequent result was later derived for it. Open systems combine the open source standards and further software properties and characteristics such as platform or interoperability. Open systems stands for a unified system consisting of open source parts. Open systems are capable to integrate assorted components, devices, or other systems at runtime to provide an emergent and collaborative functionality as a whole [1].

The open system are the outgrowth of the open source. Such systems form a set of open source softwares. As a practical example, note *Apache Hadoop* framework or *Elasticsearch* engine, which represent an open source system. A system with this scale is not easy to build, not even as an open source. To keep the software interoperability, scalability and simplicity it is needed to develop it in system parts (as smaller subsystems), which are together formulating one complex fully functional system.

We may consider the same pros and cons as it was listed in the Section 3.2 and Section 3.3 above for the FOSS. As it was noted that the open systems are derived from open source software, open systems pros and cons are slightly the same but the system complexity introduce some additional advantages and disadvantages. Those additional differences because of system size and complexity against open source software are included at proprietary software as well.

5   References

[1] Bhardwaj, N.; Liggesmeyer, P.: *A Runtime Risk Assessment Concept for Safe Reconfiguration in Open Adaptive Systems.* In *Computer Safety, Reliability, and Security.* Springer International Publishing. 2017. [Online; Accessed: 2019-11-08].
Retrieved from: https://doi.org/10.1007/978-3-319-66284-8_26

[2] DeKoenigsberg, G.: *How Successful Open Source Projects Work, and How and Why to Introduce Students to the Open Source World.* 2008. doi:10.1109/CSEET.2008.42. [Online; Accessed: 2019-10-28].
Retrieved from: https://ieeexplore.ieee.org/document/4556977

[3] DiBona, C.; Ockman, S.; Stone, M.; et al.: *Open Sources: Voices from the Open Source Revolution.* 1999. [Online; Accessed: 2019-10-27].
Retrieved from: https://www.oreilly.com/openbook/opensources/book/

[4] Ebert, C.: *Open Source Drives Innovation. IEEE Software.* 2007. doi:10.1109/MS.2007.83. [Online; Accessed: 2019-11-07].
Retrieved from: https://ieeexplore.ieee.org/document/4163037

[5] English, R.; Schweik, C. M.: *Identifying Success and Tragedy of FLOSS Commons: A Preliminary Classification of Sourceforge.net Projects.* In *First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07: ICSE Workshops 2007).* 2007. doi:10.1109/FLOSS.2007.9. [Online; Accessed: 2019-11-06].
Retrieved from: https://ieeexplore.ieee.org/document/4273082

[6] Franch, X.; Susi, A.: Risk Assessment in Open Source Systems. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C).* 2016. [Online; Accessed: 2019-11-07].
Retrieved from: https://ieeexplore.ieee.org/document/7883433

[7] Free Software Foundation, Inc.: *Initial Announcement.* 2014-04-12. [Online; Accessed: 2019-10-28].
Retrieved from: https://www.gnu.org/gnu/initial-announcement.en.html

[8] Israeli, A.; Feitelson, D. G.: *The Linux kernel as a case study in software evolution.* 2010. [Online; Accessed: 2019-10-28].
Retrieved from:
http://www.sciencedirect.com/science/article/pii/S0164121209002519

[9] Jaccheri, L.; Osterlie, T.: *Open Source Software: A Source of Possibilities for Software Engineering Education and Empirical Software Engineering.* In *First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07: ICSE Workshops 2007).* 2007. doi:10.1109/FLOSS.2007.12. [Online; Accessed: 2019-11-06].
Retrieved from: https://ieeexplore.ieee.org/document/4273076

[10] Katsamakas, E.; Xin, M.: *Open source adoption strategy. Electronic Commerce Research and Applications.* 2019. doi:10.1016/j.elerap.2019.100872. [Online; Accessed: 2019-11-04]. Retrieved from:
https://www.sciencedirect.com/science/article/pii/S1567422319300493

[11] Kuzak, M.; Harrow, J.; Jimenez, R. C.; et al.: *Lesson Development for Open Source Software Best Practices Adoption.* In *2018 IEEE 14th International Conference on e-Science (e-Science).* 2018. doi:10.1109/eScience.2018.00011. [Online; Accessed: 2019-11-04].
Retrieved from: https://ieeexplore.ieee.org/document/8588630

[12] Miller, K. W.; Voas, J.; Costello, T.: *Free and Open Source Software. IT Professional.* 2010. doi:10.1109/MITP.2010.147. [Online; Accessed: 2019-11-07].
Retrieved from: https://ieeexplore.ieee.org/document/5662568

[13] Opensource.org: *Community and Collaboration.* [Online; Accessed: 2019-10-28].
Retrieved from: https://opensource.org/community

[14] Opensource.org: *The Open Source Definition.* 2007-03-22. [Online; Accessed: 2019-10-27].
Retrieved from: https://opensource.org/docs/osd

[15] Raymond, E. S.: *Why Microsoft smears-and fears-open source. IEEE Spectrum.* 2001. doi:10.1109/6.938720. [Online; Accessed: 2019-11-07].
Retrieved from: https://ieeexplore.ieee.org/document/938720

[16] Red Hat, Inc.: *What is open source?* 2019. [Online; Accessed: 2019-10-27].
Retrieved from: https://opensource.com/resources/what-open-source

[17] Richard Stallman: *Why Open Source misses the point of Free Software.* 2019-04-28. [Online; Accessed: 2019-10-28].
Retrieved from:
https://www.gnu.org/philosophy/open-source-misses-the-point.en.html

[18] Schieferdecker, I.: *Trustworthiness of Open Source, Open Data, Open Systems and Open Standards.* In *2012 IEEE 36th Annual Computer Software and Applications Conference.* 2012. doi:10.1109/COMPSAC.2012.104. [Online; Accessed: 2019-11-07].
Retrieved from: https://ieeexplore.ieee.org/document/6340126

[19] Schuwer, R.; van Genuchten, M.; Hatton, L.: *On the Impact of Being Open. IEEE Software.* 2015. doi:10.1109/MS.2015.111. [Online; Accessed: 2019-11-07].
Retrieved from: https://ieeexplore.ieee.org/document/7217776

[20] Software in the Public Interest, Inc. and others: *Debian Social Contract.* 2019-07-16. [Online; Accessed: 2019-10-27].
Retrieved from: https://www.debian.org/social_contract.en.html#guidelines

[21] Spinellis, D.: *Half-century of Unix: History, Preservation, and Lessons Learned.* 2017. doi:10.1109/MSR.2017.1. [Online; Accessed: 2019-10-28].
Retrieved from: https://ieeexplore.ieee.org/document/7962349

[22] The Linux Foundation: *About the Linux Foundation.* [Online; Accessed: 2019-11-07].
Retrieved from: https://www.linuxfoundation.org/about/

[23] Whitehurst, J.: *The open organization: Igniting passion and performance.* Harvard Business Review Press. 2015. [Online; Accessed: 2019-11-07].
Retrieved from: https://books.google.cz/books?id=WmPhBwAAQBAJ