

Improving Freedom in the Digital Realm

Michael Brodskiy
College of Engineering
Northeastern University
Boston, U.S.A
Brodskiy.M@Northeastern.edu

Abstract—Modern scientific research, especially experimental, requires extensive collaboration efforts. Such efforts are generally best undertaken in-person; however, both as a result of the pandemic and, often, sheer distance (as scientific collaboration is often trans-continental), remote interactions have become ubiquitous. To mitigate this, the use of complex modern technologies for remote interfacing has been implemented, increasing reliance on freedom-denying virtual conferencing programs. Virtual conferences utilizing such freedom-denying programs (such as Zoom, Teams, or WebEx) infringe on the users’ rights, and, as such, my project will explore how to best tailor one’s workflow to maximize the usage of, or completely convert to, Free/Libre and Open Source Software (FLOSS), as defined by the Free Software Foundation.

Index Terms—collaboration, remote, freedom-denying, users’ rights, FLOSS, Free Software Foundation

I. INTRODUCTION

A. Scope

Technological advancements have led to ever-increasing time-space convergence, and, consequently, an unsurpassed level of global interconnectedness. In the digital realm, this has allowed for facilitated communications between physically distant peoples. Furthermore, with increasing robustness, in tandem with recent global health events, reliance on such technologies has grown. With exponential increase in use of these technologies, questions regarding ethical oversight have evinced. Companies have been proven to disrespect the privacy and rights of their users time and time again. Given that the global scientific community is one of the most dependent on such communications due to its trans-national, collaborative nature, this publication will focus on tailoring one’s workflow to avoid the use of freedom-denying programs; however, it is important to, first and foremost, define what constitutes freedom-respecting programs.

B. Free/Libre Software

The Free Software Foundation governs the general free software community. Founded by Dr. Richard M. Stallman in 1985, the Free Software Foundation defines free software as that which [1]:

- 0) Can be run as one wishes.
- 1) Can be studied.
- 2) Can be modified.
- 3) Can be redistributed.

Software meeting these criteria is deemed free or libre (to associate “free” with price, rather than liberty). A common

acronym to refer to such software is Free/Libre and Open Source Software (FLOSS). Additionally, criteria numbering begins from zero, rather than one. The reasoning for this is that it emphasizes the reliance of the other criteria on the first one.

II. THE DIGITAL REALM

A. Applicability to Education

FLOSS programs have formed the backbone of higher education since the rise of modern computing. Dr. Stallman notes that education is a community driven by the exchange of knowledge — an exchange where one gives without losing what it is that they are giving — a direct parallel to the FLOSS philosophy [2]. As such, Dr. Stallman argues that there is no environment more crucial for such software to be employed in than education. Further, Dr. Stallman notes that freedom-denying programs have been ingrained into education through “free” (as in price) licenses for students, which become paid upon entry into the workforce. Thus, students become reliant on the “industry” standard in a manner that parallels the illicit substance industry (note the parallel in utilization of “user” to refer to the customer). As such, an alarming dichotomy evinces: either the user controls the software, or the software controls the user.

B. Case Studies

Case studies have shown that individuals exposed to and involved in FLOSS have developed robust careers [3]. Individuals using such programs tend to be more driven to learn about what exactly it is that they are using, rather than simply following the “if it isn’t broken, don’t fix it” approach. Such an interest ultimately developed into a work ethic which permeated all aspects of their professional careers. On top of this, the students’ participation in FLOSS-adjacent communities allowed them to be involved in hands-on, long-term development projects. Learning these skills early on through this “apprenticeship” model allowed the students to be more open to contributing to such projects in the future, and it is this lust for lifelong learning that ultimately drives the students. It is thus evident that creating an environment with FLOSS programs is not only beneficial to people on a personal scale, but on a communal one as well.

Another shorter-term case study performed by researchers from the Federal University of Bahía seems to indicate similar benefits [4]. This study involved the incorporation

of FLOSS programs with similar functions to mainstream freedom-denying ones into a computer science classroom. As the researchers note, this allowed the students to correlate “theory and practice.” After the course was over, and the students experienced the use, modification, and integration of the FLOSS programs into their workflow, interviews were performed. It was noted that the students became more proactive when working on their projects, and were more open to collaboration and communication with their classmates, indicating improved social capabilities. Most importantly, the instructor observed that the students became more likely to disregard differences, whether it be in computing programs or each other, indicating improvement in areas beyond computer science.

With the reliance of research on document storage, Lewis & Clark University researchers developed a FLOSS program to facilitate this [5]. LIBRE-ary was specifically developed to assist researchers with archiving their works, and is a fully FLOSS program. This importantly shows just one instance in which a communal problem was identified, and the community came together to find a solution which works, or may be adapted to work, for all.

III. DIFFICULTIES AND OBSTACLES

This is not to say that FLOSS programs come entirely without difficulties. University of California, San Diego researchers worked to understand problems associated with such projects [6]. A variety of FLOSS maintainers were interviewed regarding their projects. The negatives seem to boil down to the emotional toll. Maintainers become attached, and want to see their projects grow and improve; however, scaling, especially if one is the only maintainer, may require significant effort. Furthermore, there is often opposition to FLOSS programs from government or corporations, which increases the effort required. This results in an endless cycle of the maintainer(s) working themselves to death over trying to expand their programs.

IV. CONCLUSION

Overall, case studies and interviews seem to indicate that an effective way to tailor one’s workflow to eschew freedom-denying programs would be to begin working with FLOSS programs early on. Using a “waterfall” method by continuing to use non-free programs, but seeking free programs as the need arises could provide an effectively smooth and pain-free way to transition. This would allow one to reap the benefits associated with the FLOSS program use as early as possible.

When it comes to the difficulties associated with such programs, two possible solutions exist, both of which should be incorporated to be most effective. First, the community needs to provide more support for maintainers and popular projects — something which, more or less, is already occurring. Second, there needs to be less resistance to FLOSS in general — something that could be achieved through government intervention (which is already occurring in New Hampshire).

These solutions are attainable, and will hopefully be implemented in the near future.

V. ANNOTATED BIBLIOGRAPHY

A. Bibliography

REFERENCES

- [1] T. F. S. Foundation, “What is free software?” [Online]. Available: <https://www.gnu.org/philosophy/free-sw.html>
- [2] R. M. Stallman, “Why schools should exclusively use free software,” testing if it worked. [Online]. Available: <https://www.gnu.org/education/edu-schools.html>
- [3] A. Johri, “How floss participation supports lifelong learning and working: Apprenticeship across time and spatialities,” in *Proceedings of the 14th International Symposium on Open Collaboration*, ser. OpenSym ’18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: <https://doi-org.ezproxy.neu.edu/10.1145/3233391.3233541>
- [4] F. G. Silva, M. S. Brito, J. V. T. Tavares, and C. v. F. G. Chavez, “Floss in software engineering education: Supporting the instructor in the quest for providing real experience for students,” in *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, ser. SBES ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 234–243. [Online]. Available: <https://doi-org.ezproxy.neu.edu/10.1145/3350768.3353815>
- [5] B. Glick and J. Mache, “Libre-ary, an open-source, distributed digital archiving system,” *J. Comput. Sci. Coll.*, vol. 35, no. 1, p. 22–31, oct 2019.
- [6] R. S. Geiger, D. Howard, and L. Irani, “The labor of maintaining and scaling free and open-source software projects,” *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW1, apr 2021. [Online]. Available: <https://doi-org.ezproxy.neu.edu/10.1145/3449249>

B. Annotations

- 1) Although this source is not from a peer-reviewed journal, it is important to include in that it provides an official definition for free software. It is definitely credible, given that it is published by the forefront governing free software organization. Furthermore, this identifies an organization associated with the philosophy discussed in the publication, which would allow those interested to do more background research on their own.
- 2) Although this source is not from a peer-reviewed journal, it is important to include as it bridges the divide between free software and education through eloquent means. It is definitely credible, as it is written the by the founder of the free software movement, and forefront activist for free software: Dr. Richard M. Stallman (commonly referred to by his initials, RMS).
- 3) This source concerns the proceedings which took place at the 14th International Symposium on Open Collaboration. It argues that the inclusion of FLOSS in education promotes a drive for learning, and that students using such programs tend to be successful and have more refined collaborative skills. It is one of two sources used to show that free software and education go hand in hand. The professional manner in which the case studies were performed strengthen the credibility of the source’s findings.
- 4) This source concerns a case study performed in a computer science classroom in Brazil, the findings of which were presented at a Brazilian Symposium on Software Engineering. Using the findings from the study, it argues

that students greatly benefit from the incorporation of FLOSS into classroom curriculum. It is the second source used to demonstrate the duality of free software and education. Once again, given the professional manner of the case study, and how the interviews were held following the study, the source's credibility becomes more robust.

- 5) This source concerns the development of a fully FLOSS program used for digital archiving. Though it demonstrates a bit of the technical aspects of the program, it is critical within the scope of this piece in that it demonstrates an actual FLOSS program example within education. This makes the discussion more tangible, as the reader may see that, rather than just an idea, this philosophy is already being applied. Given that the veracity of the technical aspects of the program are not being verified, the credibility of the source is not strictly relevant.
- 6) This source is an article concerning the difficulties associated with maintaining FLOSS projects. It argues that project maintainers experience great emotional tolls related with project improvement. Its juxtaposition with the other sources is critical in that it portrays that such projects do not simply come to fruition, and that great community support is generally required. Given that the researchers attempted to interview a diverse, random sample of project maintainers, they do note the limitations of the study, adding to the source's credibility.

VI. WORD COUNTS

Section	Count
Synthesis	996
First Annote	66
Second Annote	61
Third Annote	82
Fourth Annote	87
Fifth Annote	93
Sixth Annote	82