

Improving Freedom in the Digital Realm

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

AUTHOR'S NOTE

Note: Names of reviewers were omitted from this note for privacy reasons.

Entering the revision process, I was most worried regarding word count. I had already hit the upper limit, and this was before I provided greater detail to some technical aspects of the document, as requested by a peer reviewer. I decided to first incorporate updates to the best of my abilities, and then pare down the document, as necessary.

The first changes, and those I considered most important, related to formatting and wordiness. As indicated by a peer, though the document was, for the most part, in IEEE format, I did not include a column break to balance the final column length. Additionally, sections were reorganized to flow from introduction to synthesis to analysis, for easier accessibility. Finally, the readability of much of the piece was improved by removing awkward phrasing and unnecessarily wordy sentencings. I noticed that such sentences were plentiful — an unfortunate result of the tendency of this field to “get into the weeds” of word definitions.

In addition to this, the flow and incorporation of some sources was slightly improved. This was done by providing greater technical details (which was difficult to do without increasing word count).

Given that I was already utilizing a built-in \LaTeX compiler for bibliographies (BibTeX) with IEEE formatting, I did not want to completely refactor my bibliography to something into which I could incorporate both IEEE formatting and annotations. My limited experience with bibliography customization in \LaTeX made me decide to leave the annotated bibliography format as-is. On a similar note, I left the overall document format as IEEE.

Other than the bibliography and document formatting, I did receive plentiful great feedback, all of which I incorporated into the document. Given that this is a topic I find extremely concerning and pertinent to all, especially those in education, I hope to raise awareness via this publication. I continue to run an organization (GNU@NU) related to this issue for those interested. Enjoy!

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 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

Technological advancements have led to an unsurpassed level of global interconnectedness. In the digital realm, this has facilitated communications between distant peoples. Furthermore, improvements in efficacy and dependability, in tandem with recent global health events, have led to increased reliance on such technologies, which has evinced questions regarding ethical oversight. Companies have been proven to disrespect the privacy and rights of their users time and time again, and, as such, users must take matters into their own hands. 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 researching how to best promote the use of freedom-respecting programs (and avoid freedom-denying programs) in one's workflow; however, it is important to, first and foremost, define what constitutes freedom-respecting programs.

The Free Software Foundation governs the general free software community. Founded by Dr. Richard M. Stallman in 1985, the Free Software Foundation defines the “four freedoms” [1]:

- 0) Run per one's wishes
- 1) Studiable
- 2) Modifiable
- 3) Redistributable

Software meeting these criteria is deemed free or libre, or shortened to 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.

II. SYNTHESIS

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 [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 (monetarily)“free” licenses for students, which are paid for in the workforce. Thus, students become reliant on the “industry standard” in a manner that parallels the illicit substance industry (note the similar utilization of “user”). As such, an alarming dichotomy evinces: either the user controls the software, or the software controls the user.

Case studies have shown that individuals exposed to FLOSS early on have developed robust careers [3]. Individuals using such programs are more driven to learn about what they are using, rather than 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. 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 case study indicates similar benefits [4]. FLOSS programs with similar functions to mainstream, freedom-denying ones were incorporated into a computer science classroom. Students were noted to have correlated “theory and practice.” Following course completion, and after the students experienced the integration of the FLOSS programs into their workflow, the researchers held interviews. It was emphasized that the students became more proactive with projects, and more open to collaboration and communication, indicating improved social capabilities. Most importantly, the instructor observed that the students became more likely to disregard differences, digital or personal, indicating improvement in areas beyond computer science.

With the reliance of research on digital document storage, researchers developed a FLOSS program to facilitate this [5]. LIBRE-ary was specifically developed to assist researchers with archiving their works, and is fully FLOSS. As the name suggests, LIBRE-ary mimics the function of a library, as it archives works for easy accessibility. This importantly shows one of many instances 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.

FLOSS programs do not come entirely without difficulties, as determined by researchers identifying such problems [6]. A variety of FLOSS maintainers were interviewed regarding their projects. The negatives seem to boil down to the emo-

tional toll. Maintainers become emotionally attached; 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) overworking themselves by trying to expand their programs.

III. ANALYSIS

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. In addition to the more popular GitHub (whose front-end is non-free, but does facilitate the spread of free programs), many git-based repositories are continuing to become more popular. Flagrant privacy violations are becoming more apparent to users, and, thus, many are beginning to transition and contribute to programs which they can tailor on a more personal level (as opposed to a single setting being toggled for, say, colors). 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). Legislation, such as the (in-progress) Securing Our Freedom to Write and Read Everything (SOFTWARE Act), would benefit all users by not only allowing, but requiring the government to promote free software.

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

IV. 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.” [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 as it provides an official definition for free software. It is used as a contextualization piece. As a result of its production by the forefront governing free software organization, the Free Software Foundation, it is the most credible definition of free software. 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. Although no technical data is provided, Stallman does argue that, since schools have a social mission to raise contributing members of society, there is no better way than starting with free software. It is written by the founder of the free software movement, and (to this day) forefront activist for free software: Dr. Richard M. Stallman (referred to by his initials, RMS). Thus, there may exist no more credible source concerning free software.
- 3) This source 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 the goals of free software and education are one in the same; however, they cover different scopes and audiences. The case studies were performed by following the paths of two individuals given extensive experience in FLOSS programs, whose professional trajectories were then scrutinized. By using more than one case, the credibility of the source's findings are strengthened.
- 4) A case study performed in a computer science classroom in Brazil, the findings of which were presented at a Brazilian Symposium on Software Engineering, was the focus of this source. 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 similar relationship between free software and education. Given the detailed planning of the course structure, in tandem with how the interviews were held following the study, the source's credibility is 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.

V. WORD COUNTS

Section	Count
Synthesis	997
First Annote	77
Second Annote	100
Third Annote	98
Fourth Annote	91
Fifth Annote	93
Sixth Annote	82