

# A Contemporary Take on Open-Source Software

Nelson Lee

The debate between open and closed-source software has continued for decades, with compelling arguments coming from both sides. I personally believe that open-source software is essential in the tech community as it grants users complete transparency and control over their digital products. This is critical as modern tech has managed to strip consumers control over their own data and eliminate transparency in all digital applications. Our loved ones, friends, and others are unknowingly having their sensitive information traded as public property. This is not only a violation of human rights, but a deception disguised as a shiny product. Open-source software (OSS) provides a solution to this by providing applications superior in terms of accessibility and transparency. This is due to OSS being publicly accessible, allowing anyone to study, modify, and improve the software. Because open-source applications cannot conceal malicious functionality or exploit user data, it returns control of sensitive digital property to its rightful owners. Amongst the reputable sources utilized, this paper includes work from the Free Software Movement, research papers' ethical viewpoints, Forbes journal, Medium journal, and more. Some sub arguments I will examine include: Why isn't it in use already if it's so good? How will people be inspired to develop it if it's free? And more. In this paper, I will go over these and many more criticisms, demonstrating why OSS is a moral alternative to most modern-day technology.

The basic premise of open-source software is not that it's monetarily free, but that it's transparent and allows users to view and modify source code. The term "transparency" refers to the user's ability to completely understand what is happening in the background of software. "Open-source software provides users with the freedom to inspect, modify, and customize the software to suit their specific needs, offering unparalleled flexibility compared to closed-source alternatives." (OSSPH Blog 2023) It's worth mentioning right away that an overwhelming amount of software we use today contains some form of malicious intent. Google is collecting users' information across multiple platforms, Amazon is memorizing goods customers purchase and recommending new products based on user habits, Microsoft will import Chrome browsing data without the knowledge or consent of users, and more. Because of this, "proprietary software developers hold too much power over their users, easily tempting them to design programs to deceive users" (Stallman n.d.). "The developers themselves might not do this with malicious intent, but they try to make more money at the sacrifice of user freedom" (Stallman n.d.). Since proprietary software monopolizes operations on company products, users are unaware of the inner workings of programs they know and love. It's arguable that the largest ethical problems stem from this; proprietary tech companies are making large sums of money through this unbalanced power they call "software".

According to the definition provided by the Free Software Foundation: "we encourage people who redistribute free software to charge as much as they wish" (Free Software Foundation n.d.(a)).

This indicates that the word free is not inferring that software must be distributed without a price. If an open-source software company wishes to be compensated for the software they produce, it is more than acceptable to charge their users; The company will still reap the benefits of their proprietary counterparts. I've noticed this as one of the biggest skepticisms to the OSS movement; People believe that open-source equates to everything being free. They throw doubts at OSS claiming that because it's "free", no one will care to develop it. This is far from the case; in fact, many sources show that open-source software is very profitable if done correctly. "Open-source companies balance free distribution with revenue generation by selling support services, user guides, and training, demonstrating that it is possible to align user freedom with a sustainable business model" (Sacks 2015). Considering this public standpoint, it's worth acknowledging exactly what the public believes open-source to be and what it is. In the eyes of the skeptic public, "Open-Source Software is literally free software that anyone can just use as public property. It is solely created by computer enthusiasts with too much spare time on their hands. Open-Source Software is super geeky and impossible to use and that's why it is free." This is far from true. To put it plainly, open-source software is designed such that consumers are free to use it as they wish, with the freedom to study the program's source code and change it such that the program does the computing as the user wishes. Note that nowhere in this definition is it mentioned that OSS is monetarily free. During extensive research for this topic, one of the most misleading ideas surrounding open-source software was the organization known as the FSF (Free software foundation). The Free Software Foundation was established to help promote the use of open-source software, in some cases known as "free software". While the two differ slightly, they are often grouped together in social contexts. The trouble with naming an organization "free" is that people associate the word with requiring no monetary payment. While this is true in some cases of free/OSS software, that is not what the "free" in free software is referring to. To reiterate, Open-Source Software is not monetarily free! The "free" part is referring to "free"-dom, open-source software and the freedoms that it provides. Whether that's freedom to distribute, freedom to modify, or freedom to view the inner workings of the program.

Let's analyze why these freedoms are important. Every day we utilize hundreds of applications, ranging from iMessage, Facebook, Instagram, Canvas, etc. Every single one of these applications own sensitive data, including names, social security numbers, home addresses, birthdays, and more. Consider what all this data is for and where it's going, contemplate why all these apps are free to begin with. A short explanation for this is that they sell personal userbase data to companies that need demographic insights. Businesses nowadays thrive off consumer data, they need to know who's purchasing what products and how to target those consumers better. This ranges from AWS to Snapchat to more; knowing your consumer demographic is highly crucial in modern day business. Before the invention of the internet and smartphones, the only way to gather consumer data was through censuses and customer records, there was no asynchronous way to know anything about your consumer base. Things aren't quite the same anymore, with large portions of data being sold like hot Pokémon cards. Companies collect information about consumers by providing popular apps that are monetarily free, examples of this include: Instagram, Facebook, Snapchat, and more. This is the type of software I disagree with; these closed-source programs heavily violate human privacies and the protection of our data. As stated in my thesis, these programs are identity thefts disguised as shiny products. Most people using these applications are either unaware, or don't care

about what's happening beneath the surface of their favorite dog video app.

A common rebuttal to my argument among closed-source software (CSS) users is similar to this: "People are agreeing to use the service so they must comply with the terms and conditions". Yes, it's agreed that there are terms and conditions that come with using certain products, especially since they're free. Looking from an ethical standpoint however, where do we draw the line? How do we know when big tech has gone too far and infiltrated our personal circles a little too much? This is where open-source comes in. "Access to the source code is a precondition for freedom, enabling users to understand how the software works and to adapt it to their needs." (Free Software Foundation n.d.(b)). What if I told you your favorite dog video app could operate the exact same, but this time you can see everything going on in the background. Not only that, but if there's a feature that you believe isn't right or something you want to change, you can go right ahead and do it. No secrets kept; no bars withheld. If you're curious what's happening to your data and security, you can look right in the source code and see what's going on. A similar comparison I like to think of is a car; imagine every time something goes wrong with your vehicle, you're 1) not allowed to open the hood, 2) not allowed to see how they fix it. There would be absolute uproar, and people would execrate car manufacturers and mechanics. While I understand most consumers aren't technically literate enough to understand the inner workings of most applications, this shouldn't be a reason to avoid open-source software. Most people don't understand how cars work, yet as a society we often ask for a lending hand from someone who does or a professional. Consider this, if you had to pick between not being able to access something at all vs being able to access it but not understand it, which would you choose?

This begs the question: Why are people ok with closed-source in software, but not in other products in life? I boil this down to human nature and our conditioning for familiarity. When the internet and all other software products were introduced commercially back in the 80's, they were all closed-source. It was just accepted by the public that these products were beyond our comprehension and that we couldn't access the source code "just because". It is my belief that because it was introduced this way, people have never thought otherwise or to question it. You might think that this is an exaggeration, but let's showcase a famous open-source example: The Python programming language. Anyone in the modern working world knows Python, the beloved, easy to learn language is a favorite among data scientists, big tech, and the casual programmer. Python was first introduced in 1991 as an alternative to the ever-popular C, C+, Java and more. After gaining traction in the 2010's due to its rise in popularity in data science, it's now one of the most popular languages in the world. It's important to note that Python is entirely open-source! Something that is so crucial to the working world today is entirely open-source, entirely available online to use, modify, and distribute. Take a moment to think, what would happen if the founders decided to make it closed-source? What if they decided that Python shouldn't be transparent and usable by the public, with the only way to use it being to download their proprietary program and have their software steal your personal data. To me, this sounds significantly less enticing. This example specifically sounds bad due to the stark contrast to the wonderful nature of Python's open-source background. My quote of "only way to use it now was to download their proprietary program and have their software steal your personal data" is exactly what all these closed-source programs are doing to us. These companies move behind our backs, taking away control of our own data

without us even realizing it.

Another great example of open-source products are recipes. Imagine today all recipes suddenly became closed-source; You can't modify the amount of ingredients in it, the order to add certain flavors, or change the portion size. The only way to utilize a recipe now is the way that the original owner wrote it up, not to mention you must pay for the right to do this—and you need to provide your personal information so that the original chef knows everything about you. In the words of Richard Stallman: "Imagine what it would be like if recipes were hoarded in the same fashion as software. You might say, 'How do I change this recipe to take out the salt?' and the great chef would respond, 'How dare you insult my recipe, the child of my brain and my palate, by trying to tamper with it? You don't have the judgment to change my recipe and make it work right!'" (Stallman n.d.). The reason I keep highlighting such outlandish examples is because this is how I see software; it is something that is to be used for profit but shouldn't be a large gatekept secret. We are all just individuals trying to benefit from these products, if someone was to pay for it, why should they have to 1) provide so much sensitive information, 2) not be allowed to use the product exactly as they wish? A counterargument could say: "The reason that you have to provide so much demographic data makes sense for free programs, that's why they're free. They must make a living somehow". While I understand companies making free products need to make money (with the selling of personal data being one of the revenue channels), companies that offer a paid service do the exact same thing. The argument that "the only reason our data is being sold is to help companies creating free products" is nullified; it's a lazy way of raking in billions of dollars off the naivety and low technological literacy of the public.

Low technological literacy, the crux of open-source software and the reason big tech keeps getting away with their dishonesty. As mentioned earlier, when most tech related products were released, they were marketed as a turnkey product, with their syntax and files locked away behind a proprietary license. When people would question why this was the case, the response was that the products were created and designed to be used only as the manufacturer intended. As time has gone on this has evolved into the deceptive software we know today, products that capitalize on the low technological understanding of the public. The commonly heard phrase of "they don't know any better" applies extremely well here. The public truly doesn't "know any better" than to use products being provided. They don't know any better than the data stealing, privacy infringing software that is currently wildly popular. If we look at the most popular software currently: Google, Apple, Snapchat, META, TikTok, eBay, and more. You'll find it incredibly convenient among all the ads and documentation we see about these products, none of them inform the public about the inner workings of their applications. For the purpose of spreading awareness, I read the TikTok user terms and conditions. What was found may be shocking to most, here are some of the privacy infringing conditions:

- "Usage Information. We collect information regarding your use of the Platform and any other User Content that you generate through or upload to our Platform.",
- "Device Information. We collect certain information about the device you use to access the Platform, such as your IP address, user agent, mobile carrier, time zone settings, identifiers for advertising purposes, model of your device, the device system,

network type, device IDs, your screen resolution and operating system, app and file names and types, keystroke patterns or rhythms, battery state, audio settings and connected audio devices.”

- “Location Data. We collect information about your approximate location, including location information based on your SIM card and/or IP address.”

- “Metadata. When you upload or create User Content, you automatically upload certain metadata that is connected to the User Content.”

- “Cookies. We and our service providers and business partners use cookies and other similar technologies (e.g., web beacons, flash cookies, etc.) (“Cookies”) to automatically collect information, measure and analyze how you use the Platform, including which pages you view most often and how you interact with content, enhance your experience using the Platform, improve the Platform, provide you with advertising, and measure the effectiveness of advertisements and other content.”

*TikTok Privacy Policy, <https://www.tiktok.com/legal/page/us/privacy-policy/en> (TikTok Inc. 2024)*

Just reading this makes me queasy, to think that a company can blatantly take so much information about consumers for using their products. For those that still believe this is necessary, it’s clear in other examples that this is not required for revenue as much as it’s a lazy source of income. On top of the “financial” argument for proprietary software, there’s also exists the “functional use” argument. CSS supporters argue and claim “the reason you’re not allowed to modify the code or see it is because the creators intended for it to be used this way, you can’t just go ahead and change it. The product isn’t meant to do XYZ”. My rebuttal to this statement is that companies can and will go ahead and process specific requests, only at the expense of large personalization fees. This is another area where open-source far succeeds its proprietary counterparts, “Open-source software provides unparalleled flexibility, allowing users to customize and adapt programs to their specific needs, a stark contrast to the often-rigid structures of proprietary systems.” (Open Innovation Projects n.d.). Once again, CSS offers the solutions that consumers want and need, but at the expense of large sums of money.

Money, the monetary idea that drives our society. All problems between open-source, closed-source, “proprietary is way better and open-source sucks!”, and many other debates seem to revolve heavily around financial woes. If we ignore the financials of CSS vs OSS for a moment, let’s focus on the problematic ideology that “closed-source is created to provide a ‘cleaner, more polished product’”. While this may be true for a lot of software, it’s only this way by design. Companies are intentionally keeping source code and trade secrets to themselves to monopolize their products. By not sharing the source code of popular products they are ensuring that competitors can’t match them and ultimately keep revenue within the company. This would be the first reasonable argument to break down; it’s understandable that companies who work hard on their code don’t want competitors getting their hands on it. Because of this, let’s view this argument in two parts: the passion of creating something, and keeping the new creation away from competition. For starters, if someone were to make the argument against open-source from an ethical standpoint, claiming that the

programmer "put their blood and sweat into it, they created the program so they shouldn't have to share it", there are many responses that can be made. While it's understandable that a programmer could feel protective over their hard work, they seem to only care if it's costing them money. Every year, tens of thousands of programmers are handing over their hard work to big tech in return for a nice cushy salary. So, it's not the morality of loving something you create but being compensated for it.

The second part to this argument is a little trickier to unpack; if a company were to create something open-source that was extremely good, other companies would catch on fast and copy the code for redistribution purposes. While this seems problematic on the surface, let's take a deeper look at some active business models and why this wouldn't be an issue. Uber, it's a service that everyone knows and loves (or hates). The simple rideshare app calls a personal taxi wherever you are, whenever you need it. The moment that Uber found popularity, dozens of rideshare copycats came in trying to steal business. Lyft, BlaBlaCar, Grab, Hitch, and many other rideshare apps have hit the scene trying to steal Ubers' business, yet only few have succeeded. Let's take another example: Netflix. Since the inception of Netflix, there have been another dozen streaming services trying to take their business, yet Netflix remains the top choice for streaming and online TV watching. The list goes on for these businesses and the problem with copycats pop up in many different industries, not just software. There are two things here that most aren't considering: 1) If you create a product and become famous for it, there will always be someone trying to copy you. 2) Whether the source code is publicly available or not, there are a lot of smart programmers out there that will copy it regardless. Some people might argue and say, "well it's not just about copying it, if it's copyrighted then people can't mimic their business". This is a weak argument as services such as Netflix, Uber, Facebook, and many more had components copyrighted / patented and are still heavily copied by mirror companies looking for loopholes. To summarize, whether source code is publicly available or not, there will be many coming to steal business, why not at least create software with the consumer in mind?

The consumer, the most important collective of this debate. Among consumers, a notable skepticism is: "if open-source software is so good, why does no one use it". This specific criticism has many layers to it. After all, it's assumed if a product is good that many would adopt it. As someone with a foundational understanding and education in computer science, there are multiple reasons why OSS isn't commonly used today. The first culprit to point to is rather childish, but very true: "convenience and simplicity of user interface". "The foundational principles of the free software movement, while noble, no longer align with the realities of modern technological practices and user expectations, which often prioritize convenience and stability over freedom." (Lefkowitz 2021). Some may agree that this is minor but others value this heavily; regardless, it's a major deciding factor why OSS isn't used everywhere. While I personally disagree with the morality of big tech, they are very skilled at creating products that appeal to the public. Whether that's the vibrant colors of the iPhone, the pretty programs of META, widgets, emojis, and more. Big tech are very skilled at creating products that are not only usable but appealing to the casual user. As I analyze these contrasting viewpoints, I'm cognizant that I am viewing this matter both through my own developer background, and the general public's "I just want something to watch videos and send a few emails" opinion. It's worth noting that these are both extremely valid uses for our beloved machin-

ery; the goal of computers or any software is to enrich our lives and help us with tasks, whatever they might be. On one hand, it's understandable that a clean, simple user interface goes a long way; because of this, early open-source software may not have been as enticing as their proprietary counterparts. Nowadays, however, Ubuntu and many distributions (commonly called distros) of Linux offer extremely similar interfaces and pretty graphics almost identical to closed-source systems, yet distros like Ubuntu are completely unused in comparison to closed-source variants. This is a good opportunity to reiterate a previous point regarding familiarity. People early on are introduced to Windows, Mac, and other closed-source programs, it's what we "know and love". Very similarly, people will instantly go for name brands when buying groceries as it's believed that name brands are more trustworthy and guarantee quality. This is not the case, not in groceries nor computer software. Just like name brands, one possible culprit of low OSS use is lack of marketing efforts. Sadly, most open-source products don't engage in brand marketing as that would be contradictory to the purpose of the movement. Unlike proprietary applications, OSS is not created to bleed out consumers over and over, but instead to provide quality software. The desire for publicly marketed brands over a solid product is more of a social issue than a software one, but I do believe it to be one of the primary reasons OSS is overlooked. If the FSF and OSS developers decided to create massive budgets and begin marketing their product in similar fashion to big tech, usage would go up tremendously. As mentioned before however, this is very contradictory to the movement. The whole premise of OSS is that it's created for the user, not the sole benefits of the creator. By creating marketing budgets and proposing to advertise the product to increase usage, the working culture within the given OSS company would slowly shift to become the thing it set out not to be.

There is a much-needed balance between the loose ideas of OSS and firm capitalization of proprietary software. As of right now however, it feels like one or the other. Software companies are either trying to rake in as much money as physically possible, or they consist of a crew of OSS developers fighting an uphill battle for what they believe is ethically correct. The perfect example of this uphill battle is the founder of the GNU project Mr. Richard Stallman himself. Back in the 80's when the project was founded, it was a mass collaboration designed to create a whole library of entirely free software. Stallman believed that the cooperative spirit of programming as he knew it could reignite the community driven approach and make software a collaborative approach instead of a private cash grab. While it was an uphill battle that was never completely adopted, we are commonly using GNU tools in everything we do today. Some examples of this include the GNU compiler collection and GCC, both are used almost religiously by programmers around the world. The hard part of this movement was that most didn't share Stallman's love for software creation, and many had their financial woes about the project. One issue that's hard to combat about the GNU project and FSF itself, is the concept of the community driven approach. While it made sense theoretically (comparable to a modern-day collaborative GitHub repository), it is apparent why certain parts failed to take off. When pitching the idea of this paper to my peers, the number one criticism was that "no one wants to collaborate anymore, people take what they can get and will take yours too". While I believe this to be a strong take, I'd deem it mostly true. It is believed by many that people are inherently selfish and will avoid collaboration unless prompted, an ideology rather detrimental to this specific idea within the movement. If I were to imagine a new outline, I would focus more on re-empowering users with total control of their software, rather than focus

on the love for open collaboration and a community that helps each other out. I would stress the contrast between modern tech companies and the transparency of open-source, convincing people new to computing to start with OSS. It needs to be known that "Open-source software not only drives technological innovation but also creates trust and transparency, enabling businesses to build stronger relationships with their customers." (Forbes Technology Council 2022). It's fair to assume that no one would argue or claim that they would rather have their data stolen, have big tech spy on their everyday actions, and have software that is only usable the way the creators intended it.

We previously touched on many of the financial doubts addressing concerns and solutions. Now, I want to focus on something different; what does "free software" mean in the modern day and age, and how we can adapt the original movement to get people off these parasitic products? It's no secret that traditional open-source applications were notorious for being difficult to install, setup, and use. As mentioned earlier however, modern variants have adapted. New open-source alternatives are clean, friendly, and easy to use. This paper has used examples of OSS such as Ubuntu, which offers a very friendly user interface and is comparable to Windows. The difficulty examining the lack of use with a product like Ubuntu is that it works flawlessly. It's extremely similar to closed-source operating systems and works very well. Not only this, but it's user friendly and offers a variety of features that make the installation process simple. If I were to show most people an Ubuntu desktop compared to a Windows one, a large sum of people wouldn't be able to tell the difference. Knowing this, I believe the lack of open-source adaption is due to its perceived approachability. Even without marketing efforts, creators of OSS and free software need to change the narrative that surrounds these types of software. Simply informing and educating the public on the benefits of OSS is enough to bring change. I remember a time where I personally believed the same narrative, the idea that open-source projects weren't safe and that they were difficult to use. With a little research however, I came to realize this wasn't the case at all. Creators need to educate the public on why OSS is important and highlight the importance it places on the creator to user relation.

A very notable take on the creator to user relation was highlighted in Stallman's paper "Why software should be free": "I make the assumption in this paper that a user of software is no less important than an author, or even an author's employer. In other words, their interests and needs have equal weight, when we decide which course of action is best." (Stallman n.d.). I believe this to be one of the crucial backbones of why we all need to switch to open-source, not because it's free or because of the community, but rather because it's actually built with the users in mind. He continues highlighting the main issue with this sentiment: "This premise is not universally accepted. Many maintain the idea that an author's employer is fundamentally more important than anyone else. They say, for example, that the purpose of having owners of software is to give the author's employer the advantage he deserves—regardless of how this may affect the public." (Stallman n.d.). This is it, one of the most underlying issues of CSS and why I will never fully support it. The creators of software believe they are more important than anyone else because they "created" it. While it's true that the creators of anything are very important, it is ultimately the userbase that makes or breaks a product. Without a strong userbase, no product can thrive. Considering this, why is it that big tech creates products just good enough to satisfy the public? Because they can. They have been and will continue to get away with their antics if we don't educate the public and everyone using computers today. That is the purpose of papers like this; to seek solutions to contemporary issues within the



tech world.

There are many other skepticisms to the pro-open-source ideology, too many to even fit in this paper. However, I hope this opens the eyes of whoever reads it. Often, we follow the beaten path, whether that's consumer selection, life choices, and more; computer software is no different. Paper's such as this are designed to inform, to show that there are a lot of alternatives even amongst the software space. To prove that the only reason we use most products today is not because they're superior to alternatives, but because we buy into the image that big tech and CSS work so hard to create. With a little public awareness and a good number of open-source alternatives offered, one by one we can slowly re-empower our society with the devices we love so dearly.

## References

- Forbes Technology Council (2022). “Why The Business Model Of Open-Source Software Works”. In: *Forbes*. URL: <https://www.forbes.com/sites/forbestechcouncil/2022/09/02/why-the-business-model-of-open-source-software-works/>.
- Free Software Foundation (2007). *GNU General Public License, Version 3*. <https://www.gnu.org/licenses/gpl-3.0.en.html>.
- (n.d.[a]). “Selling Free Software”. In: *GNU Project - Free Software Foundation*. URL: <https://www.gnu.org/philosophy/selling.en.html>.
- (n.d.[b]). “What is Free Software?” In: *Free Software Foundation*. URL: <https://www.fsf.org/about/what-is-free-software>.
- Lefkowitz, Robert M. (2021). “Free Software: An Idea Whose Time Has Passed?” In: *Medium*. URL: <https://r0ml.medium.com/free-software-an-idea-whose-time-has-passed-6570c1d8218a>.
- Open Innovation Projects (n.d.). “Open Source vs. Proprietary Software: A Comprehensive Comparison”. In: *Open Innovation Projects*. URL: <https://open-innovation-projects.org/blog/open-source-software-vs-proprietary-software-a-comprehensive-comparison>.
- OSSPH Blog (2023). “Open Source vs Closed Source: Pros and Cons”. In: *OSSPH Blog*. URL: <https://blog.ossph.org/open-source-vs-closed-source-pros-and-cons/>.
- Sacks, Michael (2015). “Competition Between Open Source and Proprietary Software: Strategies for Survival”. In: *Journal of Management Information Systems* 32.1, pp. 206–235. DOI: 10.1080/07421222.2015.1099391. URL: <https://www.tandfonline.com/doi/full/10.1080/07421222.2015.1099391>.
- Stallman, Richard M. (n.d.). “Why software should be free”. In: *GNU Project - Free Software Foundation*. URL: <https://www.gnu.org/philosophy/shouldbefree.en.html>.
- TikTok Inc. (2024). *TikTok Privacy Policy*. <https://www.tiktok.com/legal/page/us/privacy-policy/en>.