Software Studies – Synopsis

Mark Oliver Sandfeld Nielsen

Subject: FOSS (Free and Open-Source Software)

**The idea:**

I want to write about free and open-source software. I plan to describe what they are, the ideas behind the two, the differences between them and what they have to offer. Of course, it would be difficult to talk about FOSS without also bringing their alternative proprietary (and often commercialized) software into the discussion, especially since proprietary software is currently the standard.

**What is the problem and what questions I want to answer:**

Whether or not there even is a problem of course depends on your standpoint towards free and open-source software, but if I was to define an issue it would be that proprietary software is dominating the market (and if you’re a loyalist to free software, the fact that there even is a software “market” is an issue in itself). And even to the people that don’t know anything about FOSS it might still be an issue that they have to pay for most software. The obvious question that needs answering is then why proprietary software has become and is the standard? Is it simply because the proprietary software available is better than the FOSS alternatives or is it something else?

Other questions that might be interesting to look at include, but are not limited to:

*What is the difference between free software and open-source software and the beliefs behind them?*

They obviously have a lot in common, enough for there to be a term that refers to them both (FOSS), but there are also differences between them, especially what values they each focus on. Therefore, when writing about FOSS it would probably be a good idea to clarify exactly what the differences (and similarities) are between the two.

*Does having open source code result in better software in the long run? And if not, why not?*

An argument could certainly be made that allowing everybody access to the source code of software will speed up the “evolution” of the software by allowing regular users to modify and improve upon the program and share their results. This all sounds well and good in theory and seems like it would make FOSS a superior model to it’s alternatives, but then why is it that in a lot of cases the proprietary alternatives are more popular and “better”? It’s probably not a question that has one definitive answer, especially since every case is different, but it might still be interesting to look at.

*How would the world look different if free or open-source software were the standard?*

The way the world is now proprietary software dominates in most fields. Microsoft Word is the most popular word processor, Adobe Photoshop is the most popular image editor, and so on. But what if it was the other way around and FOSS software were most commonly used? Or what if people refused to use proprietary software all together, whether or not is was “better” than the FOSS alternatives? Maybe companies would stop making proprietary software then, but if that happened, how would programmers make a living? Through donations from users? Or maybe being a programmer would simply become more of a hobby than a job. Of course, all of this is just speculation and thus answering a question like this isn’t simple. Still there might be some texts that have asked this question before and by looking at that I might be able to give a qualified guess.

Whether or not I can or should answer **all** of these questions remains to be seen. But for now, I’ll just keep them in mind.

**What sources will I use:**

Two obvious texts to use that we’ve read in Software Studies would be ”Why Open Source Misses the Point of Free Software” and ”Why Software Should Not Have Owners”, both by Richard Stallman. Richard Stallman who founded the Free Software Foundation, is of course a big supporter of free software, so other than that I might also need to find texts that support open-source and/or one that defends proprietary software against the critique it gets from FOSS supporters. With this I’d get a variety of perspectives on the issue, which could help me answer a lot of my questions. Other than that, getting a “neutral” perspective that simply look at the differences, pros and cons of the different types of software would probably also be a good idea.

**Next steps:**

First of all, I should see if I can find some texts that match the descriptions I’ve just given. Other than that, researching the issue more in general, is probably the best place to go from here.