Open Source Vs. Traditional Development Essay, Research Paper

In the world of technology today there is room for everyone at every level of expertise. From our identified Gurus, to the just bought their first computer newbie. Home computers are sometimes almost to easy to use. The old saying “just enough rope to hang yourself” comes to mind. Settings that modify the very way the system looks, and runs can be changed with the simple click of the mouse. This power, which has been given to every level of user, is not from training, but from improvements and advancements in the system, and application software on the computer. This software is in a constant state of development, improving on current features and adding new ones.

Out of this regular and consistent update in technology have grown many theories and Ideas on how software should be developed and controlled. From corporate America, to college campuses everyone has an idea from one extreme to the other. On one side the belief all software should be free and open to all, and the other protecting software secrets like a rabid guard dog. Of course, both sides believe the other is a fanatic with a complete lack of sense.

The real dividing line is not actually the control and distribution of the software itself, but the blueprint or source code that creates it. That source code is the human readable text that defines what the software does, and looks like. From that a compiler converts it to a form used by the computer know as machine code. In this compiled format it appears to be complete garbage to the average person, even to most people considered to be computer professionals or gurus. Most software–at least on the traditional side–is distributed in its compiled form. The original intent of compiling was not to protect the source code, but this has been one benefit to the software developers. Since a computer can not read the source code, a program is complied to allow it to run on similar type of hardware without the requirement of special software to interpret it into machine code.

The traditional and most common form of source code control today uses copyrights and the judicial system to protect it. Companies like Microsoft vigilantly protect their vital copyrighted assets by utilizing the court systems any time it feels a threat from an outside source.

An alternate form of source code control not only allows others to see, use, change, and resale it but also encourages that behavior. Officially know as “Open Source Software” (OSS), or “Free Software”. The later not being a very good description since it is possible, and very common for the software to cost something. Organizations involved in and utilizing OSS express that the free is in reference to freedom not cost. The theory OSS has is to develop high quality, stable software by allowing all to have access to and contribute to the source code.

When programmers on the Internet can read, redistribute, and modify the source for a piece of software, it evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing (OSI, 2000, P1)

Only until recently the theories of OSS were believed to be lofty ideas only good as ideas. Maybe for a small project or two, but no commercial quality software could come from it. It was just thought to be too difficult to bring together such an effort, especially if none of the people involved would be getting paid. However in the recent years software applications have been created using these theories that rival even the finest commercial software. These “Free” applications are even taking market share from it commercial counterparts. Software such as the Apache server—used for hosting web sites on the Internet—has acquired a majority market share.

Apache far and away has #1 web site share on the Internet today. Possession of the lion’s share of the market provides extremely powerful control over the market’s evolution. (Open Resources, 1999, P1)

There are several types of software including system software, application software, and even game software to mention just a few. System software would include the Operating systems (OS) and other programs designed to control the basic functions of a computer. The OS is the most critical piece of software on a computer, could be considered the infrastructure of computer software and even technology as a whole. It is what allows programs to run on, and interact with the underlying hardware. Modern day operating systems not only act as a conduit to the hardware, but will also manage those resources for the programs. This allows application developers to worry more about features of the software and less about interaction with the various computer parts. An application or any program running on the computer is not required to know anything about the hardware. It simply makes requests to the OS for a resource, which in turn the OS retrieves and manages for the application. For example, instead of writing hundreds of lines of code to handle all the different models of sound cards to play a sound, the developer only writes code necessary to ask the OS to play a sound.

As with application software the OS has come under the battle of development theories. Microsoft currently controls the desktop—personal home computer—market with its Windows operating system. There is some competition, such as Apple’s Macintosh and IBM’s OS/2, but none come close to the market share of MS Windows. Because of the overwhelming success of Windows, and Microsoft’s very closed form of development it has come under a great deal of scrutiny from the US, and other governments. So mush so it has been ruled a monopoly by a US Judge.

. . .determined that Microsoft was so dominant that it could sell its OS for below market prices. “Moreover, it could do so for a significant period of time without losing an unacceptable amount of business to competitors,” the judge wrote. “In other words, Microsoft enjoys monopoly power in the relevant market.” (Cooper, 1999, P1)

The OSS has also provided new competition to the Microsoft Monopoly in the OS market. Hoping to do what others, like Apple, IBM, and even Sun have attempted to do, and take market share from Windows.

With the extreme complexity required in creating, and even managing the development of an OS, it seems impossible that a not for profit organization could pull it off. But to the amazement of many, it has been done. Not only was it done but an OS was created that now rival not only the home desktop systems, but also large computer systems in use by major corporations. GNU/Linux is the outcome of the combined effort of hundred of thousands if not millions of programmers all around the world combining their effort. This was managed through two primary groups, the Free Software Foundation (FSF), and Linus Torvalids. Initially it was not even the goal for these two groups to work together. But both their systems of software control allowed them to share the best ideas and source from each group to form one Operating System. The FSF runs the GNU project which “had set forth the goal of developing a free Unix-like system, called GNU”(Stallman, 1998, P1). Linus and his crew of volunteers created the kernel for the Operating System. The kernel is the core of an OS, and does most of the back end work that is needed.

the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who’d take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles.

The fact that this bazaar style seemed to work, and work well, came as a distinct shock. (OSI)

In addition to the kernel is a diverse set of software utilities, and programs that allows a user to run and interact with a computer through the kernel, this would include the Graphical User Interface (GUI), and/or Command line Interface (CLI). The GNU project all ready had the most of these utilities, but lacked a stable kernel.

The majority of this software is licensed under a General Public License (GPL) or a Limited General Public License (LGPL). These licenses are also referred to as copyleft.

Copyleft uses copyright law, but flips it over to serve the opposite of its usual purpose: instead of a means of privatizing software, it becomes a means of keeping software free.

The central idea of copyleft is that we give everyone permission to run the program, copy the program, modify the program, and distribute modified versions–but not permission to add restrictions of their own. Thus, the crucial freedoms that define “free software” are guaranteed to everyone who has a copy; they become inalienable rights. (Stallman)

Since the licensing format allows companies, and individuals to utilize all of the features of these new technologies, and even modify it, several companies are embracing the GNU/Linux OS. Technology companies such as IBM, HP, and SUN are developing distributions that can run on their hardware. Some, like SUN, has released its OS under a similar but a bit more limiting OSS license.

The initial release of the GNU/Linux OS was strongly geared towards experienced computer users, also referred to as hackers. Since that time a movement to create a GUI that compares and exceeds the user friendly interface of Windows was put in motion. Out of that have come two strong competing interfaces for the GNU/Linux OS, KDE, and Gnome. Linux has not taken over the desktop market and may never, but it poses a threat no other competition ever has, it will never go away. It is impossible to shut something down, that has nothing to shutdown, and impossible to dry up funding to something that has no funding. Even in the event it never becomes more popular than it is today, it will continue to be supported by it’s developers, devoted enthusiasts and hackers.

As more companies continue to embrace the OSS movement it will in itself create an industry with more competition, a higher level of standards, and more consumer choice. Closed software has proven to close the market and allow it to be controlled by a single monopoly. The monopolistic technology environment of today has created bloated applications and a complete void of choice for consumers.

Bibliography

Stallman, Richard (1998), Linux and the GNU Project. In Free Software Foundation: [On-Line]. Avaialble: http://www.fsf.org/gnu/linux-and-gnu.html

Cooper, Charles (Friday, November 5, 1999), Microsoft judged a monopoly. In ZDNET: [On-Line]. Avaialble: http://macweek.zdnet.com/1999/10/31/monopoly.html

Open Resources (1999-06-17), Apache. In Open Resources/The Free Source: [On-Line]. Available: http://www.openresources.com/documents/halloween-1/node10.html

OSI –Open Source Intiative- (2000), Introduction to Open Source. In opensource.org: [On-Line]. Available: http://www.opensource.org/intro.html

Steven, Eric (08/24/2000), Cathedral and the Bazaar/Section 1. In Cathedral and the Bazaar: [On-Line]. Available: http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/