**A Different Kind of Software**

Software development has created monolithic figures such as Bill Gates and Steve Jobs, but also created lesser known innovators such as Richard Stallman and Linus Torvalds. These lesser known developers have helped shape and mold how software has been created, much to the dismay of the Gates, and the Jobs of the software world. They are known as the creators of the Open Source Software movement. This movement helps to take away all the economic control the software developers put on their code, and give it to back to the communities of users that request these programs. Open source software lets all developers view the code it takes to write a program, and allows them to modify and redistribute it as necessary. This type of development model forces programmers to support their products fully, and customize them to the needs of their users. This radically different model is what the software development world of tomorrow needs to create truly remarkable software.

**Tabe of Contents**

Introduction………………………………………………………………………….1

What is Open Source Software?.............................................1

A Brief History of Open Source Software……………………………….1

Can You Generate a Profit from an Open Source Project?......2

A Place for Open Source Software in Academia.……………………3

What Does it All Mean?...........................................................3

References…………………………………………………………………………….5



A Look into Open Source Software

**Kansas State University**

**English 417**

**Written Communication for the Workplace**

**7/5/2013**

By: Michael C. Calhoun



**Introduction**

I’m writing this document to inform the department of mediated education about the advantages of an open source software model. This model is common in the world of computing today and has lead to several innovative and pioneering pieces of software. The department of mediated education can greatly benefit from the use of open source software.

**What is Open Source Software?**

There are generally two types of software in development today, closed and open source. The most recognizable closed source software are products such as Windows OS and Apple iOS, while some of the most recognizable open source software are products such as Red Hat and Mozilla Firefox. The terms open and closed refer to the “source code” of the programs. The source code is the code that makes up the program. If a program is closed, this means that the source code is hidden away from the public and treated as top secret. Closing the source code creates a huge profit structure by enforcing intellect property rights, and patent laws. This makes sure programmers, and the corporations backing them are paid handsomely for their efforts, and they are able to defend themselves against competitors trying to steal their ideas. If a program is open the scenario is just the opposite. The code is available to the public and modification is encouraged. Opening the source code allows a community of thousands of developers to help build or debug a new program or operating system. There is still room for a profit structure by offering premium services such as advanced support, and/or long term stable releases of the software. These two types of programming have their own distinct advantages and disadvantages.

**A Brief History of Open Source Software**

The open source movement can be traced back to Richard Stallman. While serving as a research fellow in 1971 in MIT’s artificial intelligence division Stallman was working with several other programmers on developing programs that would later be the basis of most artificial intelligence programs today. MIT soon implemented a password policy for each user of the development systems. In the documentary film *Revolution OS*, Stallman recounts easily hacking into his fellows accounts, and then encouraging them to just press the enter key for the password. This was his first encounter with closing source codes and restricting the access of certain persons. Shortly after this experience Stallman formed the Free Software Foundation, and Started the GNU Project.

GNU was to be an operating system that would be completely open to the public to improve and redistribute as the users saw fit. The way Richard made sure that no one could simply rename the software, and then try to close the code was a practice called copy-left and the creation of the GNU Public License. This license gave ownership of the software to its author, but allowed the software to be modified and redistributed by its users. This license is still attached to nearly all open source software today. The license contains these key terms;

* The authors give users the permission to redistribute the software openly
* The authors give the users the permission to make changes to the software
* The source code must remain open
* The authors require you attach this license to the software upon redistribution

This license creates the communities of developers that help build the best software possible by not hiding the code and allowing for discussion.

In 1991, Linus Torvalds created the Linux operating system. Linux was then distributed through the GNU Public License. The release of Linux surprised Stallman, who had not even released a stable version his own GNU operating system yet. The release of Linux jumpstarted the open source community and gave it the push it needed to start creating ground breaking software in a revolutionary way.

**Can you generate a profit in an Open Source project?**

The biggest attack against the open source software movement is that it is a communist business plan, and has no possibility to generate a profit. This is the farthest thing from the truth. When Stallman first proposed “free” software his intentions were to optimize software production and create a community of software developers, not crush motivation or devalue the capitalist market. Stallman produced a document the same time he founded the Free Software Movement and wrote the GNU Public License called the GNU Manifesto that outlines multiple ways to earn a profit in this new form of software development.

The GNU manifesto was designed to provide software developers a way to generate profits and maintain competition. The document does state multiple times that these profits won’t be as much as what a “traditional” programmer would make, but Stallman describes this as the cost of social responsibility. Some of the profit generation techniques Stallman describes are as follows;

* A manufacturer introducing a new computer will pay for the porting of operating systems onto the new hardware.
* The sale of teaching, customization and maintenance services
* Donations by users to their favorite projects
* Software development contracting services

Due to the fact that all of code is available to competing projects, an open source software project would have to provide services on top of their market program. Being able to modify a project for a particular customer, or teaching a business how to best implement their software are both ways that projects can generate a profit. This custom service is what would set an open source software environment apart from a closed source environment, by offering customers a choice about where to spend their money for software support.

Red Hat is the best example of a profit generating open source company. Red Hat is a long term successful company worth over 1 billion dollars. Red Hat had an initial public offering in May of 2001. Red Hat’s source code is still available to the public, but this does not hold the company back. They sell stable long term enterprise versions of their community based software, and offer advanced customization and technical support for a fee as well. They are a leader in database technologies and compete with closed source companies such as Oracle, and Sun. Although there competitors can see exactly how there program is built, they cannot compete with the individualized services, and education provided by Red Hat. Red Hat even obtained a contract with the US Army to develop the operating system for the Blue Force Tracker system in 2000.

**A place for Open Source software in Academia**

With technology growing and changing every day the demand for high performing, stable, software and strong software developers is in high demand. Teaching to program is one of the ways Stallman outlines are ways to generate profits for software developers. Teaching software engineering students how to program in an open source collegiate environment has the added benefit of producing programs that would help optimize the university as a whole. Universities in general are subject to state and federal budgets that leave little money for software after all other expenses are paid for. By eliminating the cost of purchasing closed source software, universities will be better able to allocate money elsewhere.

Universities suffer the same problems that businesses do with the lack of software support and/or software customization for their universities particular needs. With not having to pay for software licenses the university can pay an open source project to customize its software for the needs of the university. Universities can also save more money by having software engineering students develop their needed software. The possibility for software advancement in an open source collegiate environment is limitless.

**What it does it all mean?**

Developers like Microsoft and Apple have such a strangle hold on their software, that applications like Word and iTunes have no true alternative. Developers know their customers has no choice in support and do little to help them. By providing the level of transparency that open source software does it forces organizations to create the best software possible and provide the best support possible because their customers will leave if they do not. The benefits of open software open the gates to a revolution in the way software development is viewed, and offer a true competitive environment that can sustain development for years to come.

**Works Cited**

* Brodkin, Jon. “How Red Hat killed its core product—and became a billion-dollar business.”. *arstechnica.com*. Feb 28 2012. Web. Jun 30 2013.
* Stallman, Richad. “GNU General Public License”. *GNU Operating System* . Mar 03 1989. Web. Jun 30 2013.
* Stallman, Richad. “GNU Manifesto”. *GNU Operating System*. Mar 03 1989. Web. Jun 30 2013.
* Stallman, Richad. “Selling Philosophy”. *GNU Operating System*. Mar 03 1993. Web. Jun 30 2013.
* “How to Misunderstand Free Software”. *Get GNU/Linux!* . Web. Jun 30 2013.
* *Red Hat.com.*Red Hat Inc. 1991. Web. Jun 30 2013.
* *Ubuntu.com.* Canonical Ltd. 1998. Web. Jun 30 2013.
* *Revolution OS.* Dir. J.T.S. Moore. Wonderview Productions, LLC. 2001. Film
* Gates, Bill. “Homebrew Computer Club January Newsletter”. *Homebrew Computer Club*. January 31 1976.