

“Emergence of Open Source Software Development”

Grzegorz W. “Greg” Piwowarski

CS431 End Project

Higher Diploma in Computer Science



NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

Department of Computer Science

National University of Ireland, Maynooth

Co. Kildare

Ireland

INDEX

1.	Introduction.....	3
2.	The history of open source.....	4
2.1.	Unix was first.....	4
2.2.	The GNU.....	5
2.3.	Linux.....	6
3.	Conclusion	7
4.	References.....	7

1. Introduction

The original approach to the software development was as trivial as trip to Dublin - a number of people was taking on the task to produce a specified piece of software. And it was always the same, fixed number of people... So the development was taking usually a long time and once the final product was ready and being deployed very little was done in terms of maintenance. If the customers discovered the bug, they had to count on the good will of Software Company X to fix it. Of course they had to pay for it and much!

This way of taking on things is quite understandable now and was then - the computers were expensive and so were the studies for computer scientists. All of that was not so widely accessible for people as it is now. They didn't have the Internet back then and even if they did, they surely didn't have Web2.0 "information treasury" so studying programming was not as easy as today. But people always wanted to save money and always wanted to be in something else than others, be an elite of some sort, so programming become more and more popular.

So what the open source software development is? Formally it is a programming paradigm in which wide spectrum of users and developers have access to the design and source of the projects. They can learn from them, they can contribute to them, change or modify them to suit their needs. So in the end the open source software is not only available to be downloaded and operated for free, but it is possible to contribute to such software, find the bugs, improve, optimize - make it better.

To understand that basic concept that hides behind this "definition" lets just imagine that we are suppose to devise something, some project in limited time and it is suppose to be really good. And there is only one person involved. What would be the result? Quite poor I guess. But if at the same time ten people would work on that - the result would be way better. What would have happen if a hundred people would work on that project and sharing their best ideas and efforts to make sure everything is done up to a certain, high degree? Every person has a different point of view so the more the views, the bigger chance of creating near perfect solution. It's as simple as that.

2. The history of open source

The history of the software development concept that we are talking here about can be traced back to the late 60ties of the XXth Century and three system names should ring bells - these are Unix, GNU and Linux. Olivier M. Prieto says in his works that no studies on open source programming can be done without understanding the history of those fundamental systems.

2.1. Unix was first.

Before it's development the operating systems were a mess - usually written to perform a very limited number of tasks and only for the machine they were supposed to work on. This situation could not be held any longer. A new, more universal operating system was needed. In late 1960-ties AT&T Bell Laboratories joined forces with General Electric and the Massachusetts Institute of Technology to create one. The result of the works carried out was a system called Multics. It was a real step forward but not to the extend it's creators originally wished for. It quickly become apparent that the task was overwhelming Bell Laboratories so in 1968 they withdrew from the development team.

The key people involved in the design of Multics (among them Ken Thompson, Dennis Ritchie and Joe Ossanna) were really disappointed - they could not address the problem they were facing - creating an operating system that would be a brake through. This failure made them eager to come up with a better solution - the UNIX. However Bell Laboratories leaders were unwilling to take another chance and didn't want to take the risks so the group of developers took one of the obsolete machines and started their works on it. Originally they developed it in Assembly language which is not as "programmer-friendly" as modern high-level programming languages. But even if they wanted to use something "easier", there was nothing available to them. Nevertheless the system was a huge success and soon the group

started works on a programming language that would make it possible to make UNIX better with smaller amount of work.

This become a foundation behind C programming language. Emergence of C made possible not only to make UNIX portable but also to make it more efficient and effective by adding the possibility to extend it's abilities with a new software.

Soon UNIX became a standard for Bell Laboratories and was used virtually to every task held on. More on - a specially dedicated support group was created to aid with system development and maintenance. Soon after, UNIX became famous as the words of it's creation resulted in "computer world earthquake". Many institution were in need for such a system and were able to obtain UNIX source code very cheaply.

The system was in constant development and was being updated - further versions were emerging rapidly.

While Ken Thompson was taking his year sabbatical at Bell Labs and had come back to his "native" University of California at Berkeley, some young and extremely talented developers were graduating from that college. Among them was Bill Joy, who started to cooperate with Thompson.

In 1977 Joy started to promote "Berkley Software Distribution" which was the first Open Source version of UNIX extended with updates from the College. This led to emergence of licensing called BSD from the original distribution by Joy. AT&T wanted of course some credit for being the "birth place" of UNIX but based on BSD Licence many independent development teams had released their own versions of the system. AT&T allied with Sun Microsystems funded Unix International. An Open Software Foundation was created by those independent developers to promote UNIX as an open source system and then so called "UNIX wars" began.

2.2. The GNU

In 1971 one of students of MIT, Richard Stallman was undergoing his undergraduate studies when he started to contribute to Artificial Intelligence Lab at MIT. He worked on MIT internal operating system called ITS, which had all the

disadvantages of the early operating systems however the development group tried to make it more efficient by constant going over the source code by different developers to make sure it is as efficient and effective as possible. This approach had some of characteristics of Open Source Software Development. In the early 1980ties the Labs were falling apart because the ITS was inadequate and obsolete to the new needs and technologies. Most of the developers from that group moved onto "Dark Side of the Force" and became the commercial developers. Stallman however didn't want to. He wanted an operating system that would be easy to use and portable and most of all - easily accessible to the mass user and free.

So he started the works on GNU which was to be compatible with the most advanced system of the time - the UNIX. GNU is an acronym from "GNU is Not UNIX". Once GNU was ready, Stallman wanted it to be available for free so he funded Free Software Foundation (FSF) to make sure nothing is treating the "availability-for-free" of his system. Moreover he created the GNU GPL (General Public License) to make sure GNU will stay open and free. GPL says basically that the source code is available to be used, extended or modified as long as all the alterations will be available to everyone on the same terms GNU was available to the "tempering" developer.

2.3. Linux

The roots of Linux can be traced back to 1987 when professor Tanenbaum created Minix - a simplified UNIX version for the education purposes. In 1991 Finnish computer scientist Linus Torvalds announced to Minix users that he is working on a new version of that system. In 1992 kernel of Linux was ready and in combination with GNU system it proved to be an effective open source operating system. Linux quickly became more and more popular. It was licensed under GNU GPL and the number of people involved in modifying the system was keep growing. And although Linux always was and is open source there exists more complex systems based on it that are available commercially - for example Novel.

Linux is in constant development, becoming more and more complex, sophisticated, better suited to modern requirements.

3. Conclusion

In my mind the Open Source Software Development had some substantial impact on the computing community. It made computer science more popular and moreover - it made a tight culture of programming where groups of developers unlimited in size works together combining their assets to produce software more reliable, more efficient, quicker, better. Their product is easily accessible and highly customizable as we can redeveloped it towards our needs. This makes that kind of software to be widely available and much popular. Isn't that what we want at the end?

4. References

O. M. Prieto. Computer Ethics: Open Source Software. Available online

- | | |
|-------------------|---|
| Wikipedia.org | - http://en.wikipedia.org/wiki/Open_source |
| | - http://en.wikipedia.org/wiki/Open-source_software |
| | - http://en.wikipedia.org/wiki/Open_source_movement |
| Bell Labs website | - http://www.bell-labs.com/history/unix/chaos.html |