Open Source Business Solutions

Ion IVAN
Radu BUCEA-MANEA-ŢONIŞ
Academy of Economic Studies, Bucharest, România

This analyses the Open source movement. Open source development process and management is seen different from the classical point of view. This focuses on characteristics and software market tendencies for the main Open source initiatives. It also points out the labor market future evolution for the software developers.
***Keywords**: Open source, UNIX, eXtreme Programming, GNU, tokens, lexemes.*

1 The dynamics of Open source

The stage reached by Unix operating system through quantitative accumulation imposed transition to a new generation, made by the GNU Project (the recursive acronym GNU's Not Unix), the initiative of Richard Stallman (1984), who took the results of new operating systems theory and the theory of programming media, obtaining a new product characterized by:

- **generalization** as far as the system functionalities covers;
- downward **portability** in the architecture of the application software to the kernel (kernel) and ascending in the operating system versions;
- **fairness**, the system being executed in a state that satisfies the conditions of the programmer and the final result satisfying most conditions required by users;
- **transparency** to the level of source code - there are no hidden features in software and security problems are corrected efficiently with the help of the Open source community.

These four characteristics being met let us to say that GNU is the natural state for open source software. Being written to high quality rules requires discipline and a high level of training, those who work with UNIX have learned to:

-**write quality software**;
-**structure procedures** with high level of generality;
-**use instructions from UNIX** interfaces and put them to work into programs;
-succeed in **significance translation** of program sequences through a creative re-using of code.

Other operating systems are not carriers of open source status because they are not transparent, the more so as developers do not have the availability of cooperation over the code.

2. Extreme Programming paradigm

XP builds an initial application which is distributed to users. Feedback obtained from them creates the basis for switching to a new version. In the case of development a package application of the utmost generality, it makes through all stages of development cycle. Package imperfections are removed on iterative basis with the information received from users. This type of programming is given by the fact that there are organizations with the same structure of flows and processes, and requires the same processing algorithms.

XP paradigm is a technique in modern software development, used initially for Chrysler factories in the United States and inherits from the constructor of the American car line production, Henry Ford, the main features. Compared with the classical approach the maximum design applications starts with all the technical specifications from potential beneficiaries, the XP method develops as soon as possible the version of the software product.

To remain competitive, software companies must maintain a permanent dialogue with beneficiaries and from the feedback obtained to extract those observations to keep the product in actuality. Where differences between user requirements and product features are found, it is recommended the reconfiguration

of the production line or performing re-engineering processes that include:
- the work redistribution;
- analyze the evaluation & control methods;
- rewriting the application.

[ASTE02] identifies ten characteristics of the XP software development:

C_1. **software project modularity**, which involves that an Open source project consists of several parts that could be tested and function in an independent manner;

C_2. **iterative process of development**, production cycle is composed of loop-type testing-implementation-testing so that the final state should be received tested and fully functional;

C_3. **incremental product distribution** comes in several versions, each benefiting from the value added by the feedback obtained from users;

C_4. **time-stamps** involves the use of planning in staggered time and each main stage of work increase the speed of the process development and allows the company to offer a new software version as soon as possible;

C_5. **decomposition of the development cycle** activities in basic, low in terms of complexity, in order to increase efficiency;

C_6. **adaptive development process**, tend to a gradual development that could be adjusted on the go depending on market requirements;

C_7. **convergence with the users' expectations**, the new version should satisfy most high beneficiaries' expectations related to stability and portability, especially in the case of Open source projects;

C_8. **orientation towards human capital**, satisfaction and good understanding between the team members would be provided sine qua non for any type of Open source initiative ;

C_9. **communication first,** both the communication between departments and also communication with the outside, the only practical way to test the performance of the product and obtain information of the overall competitiveness of the company;

C_{10}. **complementary solutions** used in the production, the best way to avoid teams division or functional departments break-down, from the smallest details of production and distribution to final product.

3. Open source business models

According to [KAVA04], there are three models of development Open source code:

M_1. **personal initiative** in which one person try to complete a project, as is the case of the Linux operating system created by Linus Torvalds in 1992 and BSD by Eric Bostic in 1990 or Apache developed by Brian Behrendorf in 1995,

M_2. **more dispersed persons** who may bring their contribution and communicate mainly through the global network. Products like Sendmail and Samba are the result of such a model for development;

M_3. **a standard team** shall operate in a specific location, usually in the headquarters of a software profiled company. It is the case for Gnome, MySQL, OpenOffice, Mozilla and Eclipse.

The development of free code is different from that of proprietary code in that it is a process of assembly of products and their integration with dedicated components, which is written specifically for this purpose. It follows the natural distribution and providing technical support. Staff involved in implementing Open Source solutions should cover the following functions:

- project management function;
- development function;
- testing function;
- communication function ;
- distribution function.

A person can handle more functions but it is not recommended that on a project the same person who is involved in development to make also the testing phase.

This lists the business opportunities offered by open source, after [KAVA04]:

O_1. opening a consulting Open source office;

O_2. migration files, printers and network to Open source;

O_3. development of web applications using Linux, Apache, MySQL and PHP;

O_4. migration of software application and databases on proprietary systems to Linux.

All commercial software solutions of general

interest have an Open source equivalent. For office solutions there is OpenOffice, for databases systems there is MySQL and to secure applications there is AVG antivirus program. Before any transition some may pay serious attention to the company's technological past, cultural organization and institutions, IT specialists' age, because all of this determine the opportunity costs involved in making such decisions.

4. Market Analysis on the Open source software

[KAVA04] presents the results of research conducted by BCG (Boston Consulting Group) in 2002 in collaboration with the SourceForge portal on the 1500 developers chosen at random which revealed the following:

- the geographical spread, 46% of Open source developers are in North America, Europe 42% and 12% in other parts of the world;
- the spreading of the developers over the countries do not coincide with that of users of Open source, almost 24% of the download sites on SourceForge are from areas in the United States, the remainder being international, mainly from Germany, Canada and the United Kingdom;
- 30% of survey participants said that the development of Open Source solutions is one of the working tasks. More than 50% working in areas related intelligent technologies, 20% are students, 7% from academia, the remaining 15% in other fields.

It is believed that a consumer goes from one technology to another when the price of this transition is kept within the limits of acceptability. The pace of development of conventional technologies are at a constant rate - according to Moore's law - while the Open Source technologies follows an accelerated trend, which draws criticism in the sense of the disruptive nature and the destructive potential of Open source. Many Open Source solutions are not so innovative and the development follows the model established by the commercial software and become more competitive by offering a better price, which is particularly important in an economy in recession with users increasingly sensitive to price.

As much a software solution tends to mature even more difficult it is to differentiate through innovation by others. In the software industry, the costs of reproducibility and distribution tend to zero, the remaining costs tend to equal price, leading companies to obtain profit mostly from the maintenance services and training, through the distribution of costs in the life cycle stages of the software product.

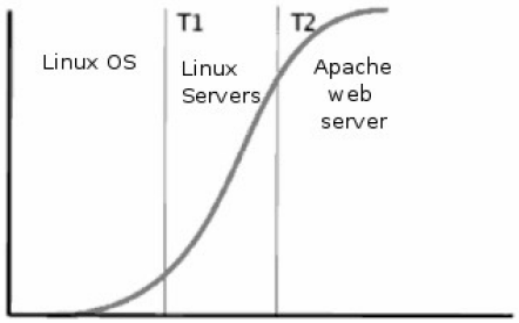

Fig. 1. Things are changing more in ten than in two years, after [KAVA04]

As may be seen in the Figure 1, the transition to the Linux operating system is caught in an early stage, while the trend of Open source file server users is in the boom phase. Most Internet providers have opted for the Apache web server therefore no longer transitions are producing in this direction.

5. Integrated Open source lexical analyzer for self-documented solutions

To assist the development of the comments in complex software, should be made a tool that:

- **compile the dictionary of symbols** common to major programming languages;
- **scanning the source code** and extract information about variables name and type;
- **build the corresponding** sequences of basic information processing.

These allow:

▪ building a list of variables to which adds comments arising from the source text on the type, area, frequency of use inside the code and links with other variables

▪ generating comments to mark the signific-

ance of processing sequences: calculating amounts, selections from a crowd, building reports and other operations with a level of complexity that depends strictly on how the product is designed.

AGCC lexical analyzer automatically generates documentation to assist classic code optimizations. This scanner contains a distinct module for analyzing tokens and lexemes in order to produce the basic documentation of the source code. The fallowing algorithm, presented in [BUCE08] seeks for the "//", "/*" and "*/" lexemes to create the associated tokens:

```
if(pos[0]<=len){
        str_comp(i,0,pos[0]);
    }
else if(pos[1]<=len){
        k=true;
        k=(pos[2]<=len)?false:true;
          if(k==false){
                  str_comp(i,0,pos[1]);
                  str_comp(i,pos[2]+2,len);
          }
          else {
                  str_comp(i,0,pos[1]);
                  return;
          }
   }
else if(pos[2]<=len){
        k=false;
          str_comp(i,pos[2]+2,len);
   }
```

After that, AGCC separates the commented lines with their position from the source code, obtaining two separate files. The resulted documentation is embedded in a XML portable structure that accumulates profile information and supplies this information to identify new optimization opportunities. In the case of high-quality software, vendors include a help file based on developers comments, structured in relation to application specific criteria.

The source file, from which the comments were removed, has a length of 2600 bytes, with nearly 1 KB smaller than the original. The content of comments occupies 1300 bytes. Experimental results show that the AGCC lexical analyzer significantly improves the performance of C Compilers in terms of execution time and error alerts. Once uploaded on Open source oriented sites, the project becomes an operational resource available to all who manages auto-documented software.

6. Conclusions

In Romania, the Open source is at the beginning for the following reasons:

- confidence in the traditional use of licensed software;
- fear that the use of Open source software after a certain stage of development generates costs higher than the use of commercial software;
- large number of users of the Microsoft or Oracle products has difficulties in adapting Open source technologies;
- the mentality that a free product is low in quality.

The activity of software development is done in the following directions:

- automation and writing drivers and communications modules for Unix, Linux and industrial machinery, using languages such C/C ++ or Python;
- development of web applications using PHP-MySQL-Apache, primarily in e-business related sites;
- customizable applications high-end type ERP, CRM platforms focusing on Java and Oracle.

Conditions that create knowledge-based society accelerate the adapting process of software developers to the high performance required by the Open Source technologies.

Bibliography

- [ASTE02] Astels, D., Granville, M., Miroslave, N.,- *A practical guide to eXtreme programming*, Prentice Hall PTR, 2002, ISBN: 130674826
- [BUCE08] Bucea-Manea-Tonis, Radu - *Analizor al gradului de comentare pentru liniile de cod integrabil în soluţii Open source*, în curs de publicare la Revista Română de Automatizăr şi Informatică
- [GOLD04] Golden , Bernard - *Succeeding with Open Source: Managers Guide*, Addison-Wesley, 2004, ISBN: 321268539
- [KAVA04] Kavanagh, Paul - *Open Source Software: Implementation and Management*, Digital Press, 2004, ISBN: 1555583202
- [MOOR06] Moore, Geoffrey- *Crossing the Chasm*, HarperBusiness Essentials, 2006, ISBN: 9780060517120