Jovica Đurković
Vuk Vuković
Lazar Raković

Open Source Approach in Software Development - Advantages and Disadvantages

Article Info:

Management Information Systems,
Vol. 3 (2008), No. 2,
pp 029-033

Received 12 Jun 2008
Accepted 24 October 2008

UDC 004.4.057.8

Summary

The works points to open source approach in software development, which has recorded important results since its beginning. Thanks to it, we are able to use the software representing the fruit of this approach. The work also points to the characteristics of open source approach through the prism of advantages and disadvantages.

Key words

free software, open source

Introduction

Open source software, simply speaking, reresents software in which the code is open and available to the users, or picturesquely said, "while (...) to (...)", contrary to "100101011"

The philosophy of open source software is based on the concept of free software developed by Richard M. Stallman. The basis of this concept is reflected in the user's freedom to use, cope, distribute study, makes changes, and improves it. More precisely, the concept of free software means four kinds of freedom for the software user:

- Freedom for the user to run the program for any purpose (freedom 0).
- Freedom to study how the program works, as well as the freedom to make it to his use (freedom 1).
- Freedom to redistribute copies (freedom 2)
- Freedom to improve the program, and show to the public so that it could use benefits (freedom 3). Access to the source code is a precondition for this (Joshua, 2002).

The program belongs to the group of free softwares, if users have all the cited freedoms. The "free software" is the question of freedom in the above-cited items, but not the questions of price. To understand this concept better, it is necessary to think about free software within the context of freedom of speech, not within the context of something that is free.

1. Open Software Source Models

Eric Raymond in his book "the Cathedral and the Bazaar" describes the open source community and its method in software development. The title of the book symbolizes two concepts in software development, the concept of producing the proprietary software carefully planned as the project of cathedral building, on the one hand, while, on the other hand, there is the concept of open source software production, which, simply speaking, is based on communications between participants of an oriental bazaar. Although this analogy can be too extreme, it indicates the basic difference between these two concepts: strong and powerful management contrary to connected users and developing staff organized in, by appearance, several thousands of independent projects.

Table 1 Open source software in contrary to proprietary software (Raymond, 2000)

	Proprietary software	OS software
Model	Cathedral	Bazaar
Rsources	Known	Unknown
Planning period	Whole project	Step by step
User	User who paid the software	Participant in software developinng
Target	Fulfill the contract	Solve the problem
Discipline	Strong	Weak
Development	Secret, reliable	Public
Cooperation	Face to face	Via the Internet
Quality warranty	Management	Competition

2. Open Source Project

Every group developing software and the results of work to the public with open source license makes the open source project (Evers, 2000).

The open source project is difficult to analyze as one abstract social appearance. In addition, it is difficult to define what the part of project is and what it is not. Fortunately, we can consider and analyze open source projects thanks to their presence on the Internet and public communications.

Educational institutions, faculties and other institutions develop software for educational and research purpose. Part of this software ends in the group of proprietary software, whiles some of it, according to parameters, and belongs to the group of open source software.

Research institutions. Research work is often closely associated with educational and public institutions in the field of exchange of experience, staff and financial resources. For that purpose, many projects are based on strong cooperation between different organizations. Therefore, it is natural that the results of researches are free from the license, which enables lately the participants of researches to use it. In addition, this kind of license sometimes represents the conditions for financial sponsorship.

Commercial companies. Divided from distributers, every company, using open source software, can participate in OS projects. Large IT companies as IBM, Intel, Hewlett Packard, and others fall into this group.

Users-collaborators. If we take into consideration enormous financial resources which many companies, i.e. administration, spend for their own software systems (usually several millions of dollars only for license), sponsorship of open source project is much cheaper than paid license of the proprietary software.

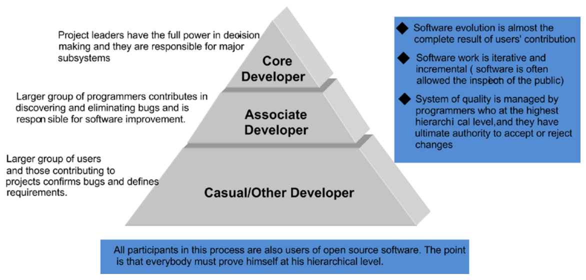
Users. Most users are interested in improving software in advance, and it can bring them direct benefits. For this purpose, many users, in some way, participate in open source projects. As many of them work in IT companies, their participation is of the key importance for the project.

Governments. About the fact that future economies mostly depend on information technologies, some governments in the world are interested in software production. For that reason, they want to have one alternative at least for their economies in the future, in case those current strategies fail. Besides, governments do not like that; their economies should not depend on one alternative only, e.g. foreign company.

How does an open source project start?

A typical start of one open source project is described in the following text, chronologically:

- Somebody is interested in some problem and he thinks about good solutions.
- He inquires about it with friends and colleagues if they know anything about that problem. Some of them have similar problems, but no solutions.
- All interested start to exchange knowledge in this already defined theme and, in this way, they create the drawing to solve the central theme.
- The interested, wishing to invest their resources in solving the problem make an informal project, while the disinterested leave the group. For that reason, the central theme covers all those participating in the project.
- Project participants work to solve the problem until they reach significant results, which deserve to be published.
- Results are published there where most people have free access.
- All those having interest in the project are interested in good solutions of the project. For that purpose, they test results of the project (they use them). As they consider the project from different aspects, they sometimes give good suggestions for advancing the project so they join the project.
- The project develops many feedback information help in better understanding the problem and possible strategies to solve them.
- New information and resources are integrated in the process of researching.
- Research period ends and returns to Point 5.
- The project community is established now and it will react to future changes (Evers, 2000).


Project leaders have the full power in decision making and they are responsible for major subsystems
Core Developer
Larger group of programmers contributes in discovering and eliminating bugs and is respon sible for software improvement.
Associate Developer
Larger group of users and those contributing to projects confirms bugs and defines requirements.
Casual/Other Developer
Software evolution is almost the complete result of users' contribution
Software work is iterative and incremental (software is often allowed the inspection of the public)
System of quality is managed by programmers who at the highest hierarchi cal level,and they have ultimate authority to accept or reject changes
All participants in this process are also users of open source software. The point is that everybody must prove himself at his hierarchical level.

Picture 1 The graphic survey of open source software methodology (Open source: IBM strategies for developing countries, 2002)

3. Advantages of Open Source Approach

The work will point numerous features of the open source approach to software development, which represent its advantages at the same time.

The prototype planning is closed for the public and it is common for all open source projects. It is known that the bazaar model does not function properly in the early stage. Participants need functional software in order to advance it. Therefore, a small group develops the prototype with a view of attracting users. The prototype is closed at the beginning because it is easier to keep the integrity of the project concept.

This approach is very efficient. Closing the prototype, chances to keep the initial design consistent are much bigger. However, at the same time, a great number of possible decisions for design are postponed until the product is presented for inspection to the audience. This is especially important when requirements are uncertain and unclear.

Open source projects have prototype development when the initial project gradually advances in iterations. According to the definition, iterative approach is adaptable; it means that design can be changed during development in accordance with new requirements. Going ahead by small steps, the process remains flexible enough to accept ad hoc changes.

Many small, but constant changes of the code, as well as the test where many experts take part, contribute to fault minimization. Just such widespread tests are very efficient in developing open source software. It is logic that many users can discover more bugs. Communities gravitate to different development staff so increasing the possibility that the bug will be obvious to one person at least. Besides, those responsible to find bugs have the possibility to choose, and it is an adding motivation to locate the problem. Approach to source software code enables an easier elimination of bugs once they are found.

Communities gravitate to independency in organizing, and development decentralized. Open source projects are not organized in big teams; on the contrary, they are small groups of experts with high level of functionality.

Development staff carries out control in the organizational hierarchy, which makes decisions about which solutions will be accepted and integrated in the source code. Such a model is based on personal trust between the community members. Those, with authority and responsibility, rely on the community members who are the most competent. Leaders, in essence, make right decisions based on trust. The organization, functioning on the principle of voluntary work, is simply forced to this management system. It is the efficient approach because the most skilled participants gravitate to be its leaders, too. They are experienced experts and they usually demonstrate results, which are respectable for the community. In this way, conflicts are rear.

The open source approach includes motivation, too. It is secondary in open source software development. Programmers participate in the project because they are capable to write the code. At the same time, they accept experience offered by community and attain affirmation dependent on realized results. Looking from this standpoint, the source of motivation is in projects and challenges originated from the project.

Communication between participants in an open source project is asynchrony. It is the results of geographical distribution and it has some deviations in dynamic project plans and the fact that participants in the project communicate via the network. Asynchrony communication is very well in case of the decentralized structure.

Open source project planning is informal. There are not concrete plans and visions, but the long-term objective, i.e. project advancement. This is not a problem. On the contrary, imprecise defined objectives and the absence of deadlines can probably improve productivity. Many open source projects are very productive, especially those based on voluntary work and participants are not obliged to work the full time. Absence of any pressure, forced by planning, can be another factor to stimulate project participants. This excludes artificially set limitations in the process of software development. Development staff has enough time as needed to implement new solutions.

There are different levels for all those who can contribute in open source projects. These levels are defined according to participants' competence and obligations. Relatively small group of major programmers write the biggest part of the new code and they are project leaders. Then, the bigger group improves shortages, and even bigger one reports on problems identified. The whole community of open source project uses and advances the product.

In this way, there are no limitations from the aspect of participation in open source projects. At the same time, integrity of design and implementation is preserved. Everybody can use software, finds faults and suggests new features. However, the principle to which the new code is included into the main code is directed according to some precisely defined principles. Such an approach is set to support the process, which is decentralized, including many participants with different skills and experience (Johnson, 2001).

4. Disadvantages of Open Source Approach

On the other hand, some characteristics of open source approach are its disadvantages, some of which will be illustrated later.

Prototype development depends largely on the initial expertise. The concept of initial design is important because development staff will have problems to eliminate basic software lacks and disadvantages. Experience shows that it is necessary to pay attention to the time when to free the prototype. If we free it too early, the project can have unclearly defined direction. Project stagnation, as well as conflicts of opinions is quite a usual thing. However, if we free the prototype too late, the design can remain without precise users' requirements. For this purpose, successful management of prototype software activities is a challenge.

When we talk about quality control, open source projects only rely on the community criticism. Other test types are unusual, especially in the field of designs. It is very important to develop projects properly, as well as testing of their quality. The community criticism is sometimes incapable to identify highly rated faults in the architecture. In addition, it is also inefficient in finding imperceptible faults. These are faults which do not appear very often and which cannot be easily identified in the process of code testing. Besides, efficiency of the community criticism can be unclear. It can happen that one person finds a bug, but many others will spend much time searching it because of the lack of information.

Decentralization is a necessity in open source projects in order to protect those who contribute to the project from excessive bureaucratic rules and procedures. However, this standpoint can cause repeating inventions. A bigger part of development staff work separately. Many small projects constantly overlap.

Competition, as rivalry present as the phenomenon of development staff can cause mutual conflicts. In these cases, the war can break, figuratively said, where an individual or more of them refuse to stop writing the code and allow more efficient and better solutions. Quarrelsome project participants can make a code copy and start to distribute different products. Fortunately, project division is rare. Although the real reason for this phenomenon is not known, it seems that it can be found in the absence of the project owner's political skills.

The status of project participants is an important form of motivation in open source software development. Participants try to acquire affirmation, making efforts to provide the best solutions. However, this can burden the project, in the end. Without defining the participant's status and influence in the project, discussion on code acceptability can escalate and take a lot of time, and good participants leave the project.

Asynchrony communication functions well in many open source projects, but as the community grows, probability to become unmanageable is proportional to its growth. Information list dramatically increases, and fewer experienced participants can lose the flow of the project and remains without some information. Experienced participants can avoid the public list of information and rely on mutual discussions. In these cases, feedback information between those who test and those who develop software is reduced.

Informal planning helps to motivate participants and increases productivity, but on the other hand, makes the prediction of the project end impossible. The process itself is not clearly visible and it is the reason why it is difficult to estimate its progress. There is no obligation to end anything in the determined period.

Different participation levels in the project help to preserve control in open source projects, but in this way, the hierarchical management level of the project is exposed to adding load, taking into consideration that the best programmers are also project leaders. Having in mind the adding load of project leaders, the principle of this approach can become a problem.

Most open source projects put great efforts into writing adequate documentation. It is often easier to provide resource for information system maintenance than provide documentation containing information about it. Therefore, most knowledge and skills in open source projects are hidden and not recorded. This discourages learning about projects and unnecessarily takes time (Johnson, 2001).

Conclusion

The approach to open source software development represents an alternative to the conventional approach of software engineering. After a very modest beginning, this approach has realized its full affirmation in the moment of commercialization of the Internet. The basic idea of open source approach is the user's freedom to run, copy, distribute, modify and improve the software. Programmers who voluntarily want to contribute to the process of software development primarily accept it. Gradually, open source movement has become bigger and as a result, we have thousands of projects where a great number of people participate. This number is difficult to define because product users are also project participants, having the full right to present the professional audience their solutions and to realize them if satisfy appropriate requirements. The key of success of open source approach can be seen in this fact. Two most known open source projects, Linux and Apache speak in favor of this thesis. The number of users is constantly increasing (Apache has been, practically, the most used Web server since it appeared, and Linux goes the same way).

References

Evers, S. (2000, July 29). An Introduction To Open Source Software Development. Retrieved December 15, 2008, from Informatik-Rechnerbetrieb: http://user.cs.tu-berlin.de/~tron/opensource/

Johnson, K. (2001, June). A Descriptive Process Model for Open-Source Software Development. Retrieved January 9, 2009, from University of Calgary: https://dspace.ucalgary.ca/bitstream/1880/41007/1/2001_Johnson%2c%20Kim.pdf

Joshua, G. (2002). Free Software, Free Society: Selected Essays of Richard M. Stallman. Boston: Free Software Foundation.

Open source: IBM strategies for developing countries. (2002, September 5). Retrieved January 20, 2009, from Developing Country Access to On-Line Scientific Publishing: Sustainable Alternatives: http://users.ictp.it/~ejds/seminars2002/Giampaolo_Amadori/ICTP_Speech.pdf

Raymond, E. (2000, September 11). The Cathedral and the Bazaar. Retrieved November 2, 2008, from The Cathedral and the Bazaar : http://catb.org/esr/writings/cathedral-bazaar/cathedral-bazaar/

Jovica Đurković

University of Novi Sad
Faculty of Economics Subotica
Segedinski put 9-11
24000 Subotica
Serbia

Email: djovica@ef.uns.ac.rs

Vuk Vuković

University of Novi Sad
Faculty of Economics Subotica
Segedinski put 9-11
24000 Subotica
Serbia

Email: vuk@ef.uns.ac.rs

Lazar Raković

University of Novi Sad
Faculty of Economics Subotica
Segedinski put 9-11
24000 Subotica
Serbia

Email: rakovicl@ef.uns.ac.rs