Open Source Software Development Time Frame Model

Anand Kumar Rai[1], Karuna Shankar Awasthi[2] and Laxmi Shankar Awasthi[3]

[1]Deptt. of Computer Science, Mumtaz Post Graduate College, Lucknow.
[2]Deptt. of Computer Science, Lucknow Public College of Professional Studies, Lucknow.
[3]Deptt. of Computer Science, Lucknow Public College of Professional Studies, Lucknow.

Abstract

Open source software system is becoming in trend due to its quality, security and low cost development. Two types of developer's team may work in such kind of software's. First kind of developers are the core developers and second kind of developers are the public developers who come from the public domain contribute in the software development voluntarily. This research paper focuses on the development of open source software within limited time frame using some AI based project automation tools.

Keywords: *AI Application, Core Developer, Open source software development, Public Developer, Time Frame.*

1. INTRODUCTION

Open-source software development copyright holder released the software under license in which he grants the rights of change, study and use the software to the user [1]. OSS may be developed with the collaboration of public developers. In such kind of development environment any capable programmer can participate in the online software development process, in general there are no any limit of software contributors. Code examine facility increases the trust of the public into software. A report published by Standish Group in 2008 in which it is mentioned that OSS software development model increased the savings of approximately $60 billion per year for consumers [5][6][7].

In 1983 the idea of making source code freely available originated by Richard Stallman, programmer at MIT. They wanted that software should be accessible to public domain developers who want to take interest in it. The public domain developers can understand the goal of the software and they can modify and improve it. The Stallman started releasing free code under his own licensing known GNU Public License. This kind of approach finally led to the development of open source software initiative in 1998[8][9][10]. Later, Open Source Initiative (OSI) was created to promote and protect the open source software development communities. OSI provide guidelines for the interaction and support with OSS. Open source code generally stored in a public forum repository so that any public community developer can use, contribute and improve the design and functionality of the project. The five most popular licenses are.

(i) MIT (ii) GNU GPL 2.0 (iii) Apache License 2.0 (iv) GNU GPL (v) BCD License 2.0

The terms of licenses are play major role in future distribution or free usage of the software.

In this paper a new time frame OSS model has been proposed to develop open source software within stipulated time period. Sub-section 1.1 describe about the methodology used to fulfill this research and sub-section 1.2 described the existing research problem.

1.1 Research Methodology

Allied Literature review conducted to collect the research data in the field of open source software development. To authenticate the model, we have taken interview around 100 people currently who are working for open source software project. We have created assumptions to validate and authenticate the proposed model.

1.2 Research Problem

In the open source software development method there are major uncertainty regarding the development schedule. The initiator of the open source project cannot claim for the completion of the project within definite time period. Core team or project initiator firm don’t have control over public developers and public developers don’t feel any pressure for the code contribution. Some public developers may contribute more than expectations and few more may be idle, therefore it is very hard to evaluate the project completion duration.

2. BACKGROUND

The concept of open source software moves towards commercialization in the year 1970-1980. Some researchers given their opinion on Open Source Software(OSS) through their research publication in 1979 Donald Knuth came with the TeX typesetting system, in 1983 Richard Stallman came with the GNU operating system. A thoughtful analysis was published in 1997 by Eric Raymond which title was “*The Cathedral and the Bazaar”* which was based on the principals of the free software and the hacker community. This paper caught attention in the year 1998 and it became motivating factor for Netscape Communications to release their in internet suite as free software. The same source code became the foundation of Sea Monkey, Mozilla Firefox, Thunderbird and KompoZer.

In 2001 Microsoft executive Allchin publicly announced that “Open source is an intellectual property destroyer, I cannot imagine something that could be worse than this for the software business and the intellectual property business”. Although open source software's played historical role and these companies came into the mainstream of the software development industries. The IBM, Oracle, Google and State Farm are the companies which stands strongly open source competitive market.

The free software concept began in 1983 and in 1998 a group of individuals suggested that term free software is not an appropriate term, it should be open source software.

An open source strategy session held on 7 April 1998 in Palo Alto, announcement of a source code release for Navigator(Mozilla). In that session a group of renowned individuals from the IT industries were present including Linus Torvalds, Tom Paquin, Jamie, Larry Wall Wall , E. S. Raymond etc.

Eric S. Raymond introduced a open source software development model in 1997, he mentioned about this model titled “Cathedral and Bazaar” known as the *bazaar* model. Raymond suggested that every software can be developed using the bazaar style. Different automatic tools are available to manage the OSS development process. Revision control, Concurrent Version System(CVS), Git and Subversion are the tools to manage the source code files.

3. PROPOSED TIME FRAME MODEL

Open source software public developers work voluntarily therefore they have no any pressure regarding the project. To accomplish the open source project in time frame is very difficult task. In this paper we have given suggestions through the proposed hypothetical model. You can observe in the given Figure 1, shows that what steps are required to develop the open source software project within stipulated time period. In this model three phases are defined for development process. The description of the proposed model phases are as follows:

3.1 Initiation

This phase initiate and form the foundation of the project, in this phase three steps are defined to complete the initiation phase. First step is problem identification in which the project leader has to address the current issues in the software technology for which he needs help from the public domain programmer/experts. To address the problem and proper communication with the public domain team the core team has to create the online forum for discussion and submission of the code from the public domain developers. To accomplish the project in stipulated time, frame the project leader will ensure the time contribution from the public domain developers. Finally, third step of initiation phase need to select appropriate person for the core team and productive people should be invited to join the project from the public domain.

3.2 Execution

In this phase of development, code production and submission are started from the core team as well as public domain team. To monitor the development process and evaluate the code quality AI based monitoring tools are used.

Some tools which have been tested by me for this purpose like Hiretual, Mindsync and IBM Watson. Mindsync can be used as optimum code selector and time management tool for the proposed model. Finally, selected code can be adopted for the project. Documentation process is required for all these steps completed online or offline mode [14] [16].

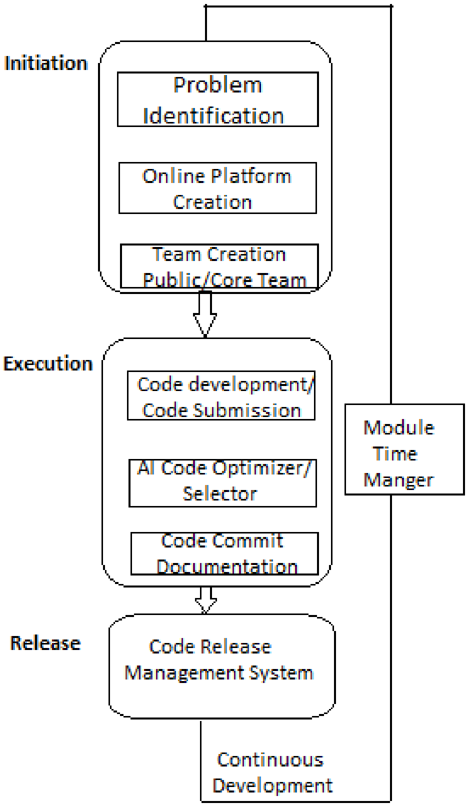
Initiation
Problem Identification
Online Platform Creation
Team Creation Public/Core Team
Execution
Code development/ Code Submission
AI Code Optimizer/ Selector
Code Commit Documentation
Module Time Manger
Release
Code Release Management System
Continuous Development

Figure 1. OSS Development Time Frame Model

3.3 Release

In this phase software module released officially into the production environment after testing the particular module it is integrated with the main project. This development cycle continues till the complete development of the project. Final release is done when it is observed that project fulfilled its whole objective. In open source projects, the process of improving and modifying the project continues as long as the project initiator wishes [19] [20].

4. RESEARCH VALIDATION

We can validate this proposed model by some assumptions which are given below.

Assumption-1

One project named ‘A’ in which five core developers are working and from the public domain 100 developers are giving code contribution for the project. The estimated duration for the project completion is approximately 10,000 hours. Suppose core developers give their contribution 8 hours daily and average code contribution from public developers 1 hour daily. Out of five core developers 3 developers are contributing code and rest 2 developers are engaged in planning and code review and finally code module releases.

Total code contribution by the development team = 3 core developer’s x 8 hours per person per day + 100 public developer’s x 1 hour per person per day =124 hours daily.

Total time taken to complete the project= estimated time (hours)/ work done hours per day= 10000/124= 80.64 days

Assumption-2

Same project as described above whereas 10,000 hours are total estimated time for the completion of the project development. We are deploying the proposed time frame model for this project. Here the code contribution provided by the core team and project management work are taken care by the AI tools. Let’s say in a working time of 8 hours the core team gives 6 hours in code contribution 2 hours for planning and rest management work. Public developers can extend their code contribution time by citing the project deadlines. Now we can consider their average code contribution as 1.5 hours per person per day.

Total code contribution by the development team = 5 core developer’s x 6 + 100 public developer’s x 1.5 =180 hours daily.

Total time taken to complete the project= estimated time (hours) / work done hours per day = 10,000/180= 55.55 days.

These two assumptions are sufficient to validate the proposed model and here we can see that if traditional OSS development method takes approximate 3 months to complete. When we apply the time frame model for the same project, it takes around 2 months to complete.

5. CONCLUSION AND FUTURE RESEARCH

This research paper gives us an idea of how useful the timeframe model is for OSS project development. These two assumptions are sufficient to validate the proposed model and here we can see that if traditional OSS development methods takes approximate 3 months for any project, then after deployment of the proposed time frame model for the same OSS project takes approximate 2 months.

This study is completed for small scale project, further more research are needed for large scale OSS project with different development conditions.

6. REFERENCES

[1] Crowston, K., Annabi, H., Howison, J.: Defining open source software project success. In: ICIS 2003 Proceedings, p. 28 (2003).
[2] Conaldi, G., Lomi, A., Tonellato, M.: Dynamic models of affiliation and the network structure of problem solving in an open source software project. Organ. Res. Methods **15**(3), 385–412 (2012).
[3] Fitzgerald, B.: The transformation of open source software. Mis Q. **30**(3), 587–598 (2006).
[4] Grewal, R., Lilien, G.L., Mallapragada, G.: Location, location, location: How network embeddedness affects project success in open source systems. Manage. Sci. **52**(7), 1043–1056 (2006).
[5] Hertel, G., Niedner, S., Herrmann, S.: Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel. Res. Policy **32**(7), 1159–1177 (2003).
[6] Hahn, J., Moon, J.Y., Zhang, C.: Emergence of new project teams from open source software developer networks: impact of prior collaboration ties. Inf. Syst. Res. **19**(3), 369–391 (2008).
[7] Haefliger, S., Von Krogh, G., Spaeth, S.: Code reuse in open source software. Manage. Sci. **54**(1), 180–193 (2008)
[8] Lakhani, K.R., Von Hippel, E.: How open source software works:“free” user-to-user assistance. Res. Policy **32**(6), 923–943 (2003).
[9] Long, Y., Siau, K.: Social network structures in open source software development teams. J. Database Manage. (JDM) **18**(2), 25–40 (2007).
[10] Li, Y., Tan, C.H., Teo, H.H.: Leadership characteristics and developers’ motivation in open source software development. Inf. Manage. **49**(5), 257–267 (2012).
[11] Mockus, A., Fielding, R.T., Herbsleb, J.D.: Two case studies of open source software development: apache and Mozilla. ACM Trans. Softw. Eng. Methodol. (TOSEM) **11**(3), 309–346 (2002).

[12] O'Mahony, S., Ferraro, F.: The emergence of governance in an open source community. Acad. Manage. J. **50**(5), 1079–1106 (2007).
[13] Ryan, R.M., Deci, E.L.: Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am. Psychol. **55**(1), 68 (2000).
[14] Roberts, J.A., Hann, I.H., Slaughter, S.A.: Understanding the motivations, participation, and performance of open source software developers: a longitudinal study of the Apache projects. Manage. Sci. **52**(7), 984–999 (2006).
[15] Shah, S.K.: Motivation, governance, and the viability of hybrid forms in open source software development. Manage. Sci. **52**(7), 1000–1014 (2006).
[16] Subramaniam, C., Sen, R., Nelson, M.L.: Determinants of open source software project success: a longitudinal study. Decis. Support Syst. **46**(2), 576–585 (2009)
[17] Singh, P.V.: The small-world effect: the influence of macro-level properties of developer collaboration networks on open-source project success. ACM Trans. Softw. Eng. Methodol. (TOSEM) **20**(2), 6 (2010)
[18] Singh, P.V., Tan, Y.: Developer heterogeneity and formation of communication networks in open source software projects. J. Manage. Inf. Syst. **27**(3), 179–210 (2010)
[19] Sen, R., Singh, S.S., Borle, S.: Open source software success: measures and analysis. Decis. Support Syst. **52**(2), 364–372 (2012)
[20] Wen, W., Forman, C., Graham, S.J.: Research note-the impact of intellectual property rights enforcement on open source software project success. Inf. Syst. Res. **24**(4), 1131–1146 (2013)

.