# COCOMO model for software based on Open Source: Application to the adaptation of TRIADE to the university system

Moulla Donatien Koulla / Ph.D Student
University of Ngaoundéré
Ngaoundéré, Cameroon
donafice@yahoo.fr, moulladonatien@gmail.com

Kolyang / Assistant Professor
The Higher Institute of the Sahel
Maroua, Cameroon
dtaiwe@yahoo.fr, dtaiwe@gmail.com

***Abstract*—Today, within the software industry, Open Source Software has many qualities that need to be analyzed. This kind of software has gained a lot of attraction nowadays from researchers since it offers technical and economical advantages compared to paid software. This article shows that the development of software based on Open Source has many cost-oriented advantages. The present work adapts an Open Source Software and evaluates the cost of its adaptation. This evaluation was performed according to the model Constructive Cost Model (COCOMO).**

***Keywords- Open Source Software; Intermediate COCOMO Model; Software Estimation;***

## I. INTRODUCTION

Environment based on paid software has high cost. The Open Source movement tends to be a viable alternative with remarkable performance and low cost. The term Open Source used in this article corresponds to a definition by the Open Source Initiative (OSI), a non-profit organization. According to their definition, "An Open Source Software is a program that can be freely modified and redistributed. Redistribution may not be free [5] ". Open Source Software differs from proprietary software by the possibility given to end users to run, copy, distribute, study, change and improve it [6]. This type of software has gained tremendous attractions from researchers since it presents technical and economical advantages compared to proprietary software. Today, many software are built using Open Source as building blocks. In fact, reusing existing code allows people to take advantage of preceding development. They don't have to start from scratch. Existing components allow one to reduce the development time. Open Source Software constitutes today a development and distribution model, an emerging economical model that needs to be analyzed. This article shows that development based on Open Source components has a lower cost. The present work adapts the Open Source application TRIADE in order to be used in the university system in Cameroon. Next, the cost of this development was evaluated by using reverse engineering techniques. In order to achieve the adaptation and for the evaluation of the resulting cost, the Constructive Cost Model (COCOMO) has been used.

### *A. Presentation of TRIADE and the methodology used for its adaptation*

Management of education is one of the main concerns of many universities. It concerns wide groups of people namely students, lecturers and the administrative staff. Each group has its own needs and this diversity makes the software supporting the process, very sophisticated. It has to take into account the administrative record management, accounting, and education and so on. One of the major advantages of Open Source Software is the possibility to access their source code.

Therefore it is possible to personalize and adapt existing software to specific needs. TRIADE is a platform designed to enable e-learning. The term TRIADE means « Information Transparency and Rapidity in Teaching ». It is developed using the PHP 5 programming language and stores its data into a MySQL or a PostgreSQL database. This application is designed to manage student in high schools. An adaptation to the context of a university has been achieved: The case of the faculty of science at the University of Ngaoundéré in CAMEROON was studied.

To adapt this platform to the university, reserve engineering method were applied to the old system. Reserve engineering refers to « the activity which consists of studying an object in order to understand its functioning [3] ». To understand how the platform is built, two main sources of information have been used:

- The source code of the application;
- The users of the platform.

The starting point to understand how the system is built is to understand how the different users interact with the system. At the first place, thanks to the users, an understanding how the system is used in the business perspective (system functionalities) was possible. In a second step, an analysis of the source code of TRIADE and its execution in production has been done. This permits to retrieve the architecture of the application with its different components and classes. The source code and the users helped to find similarities that exist between TRIADE and the system to put in place.

### *B. Presentation of COCOMO model*

The estimated cost of software development is one of the main issues discussed in software engineering. The control and evaluation of this kind of cost concerns both the researchers and the software project managers. However, the development of Open Source Software also has a cost. The COCOMO model is one of the model used in the estimation of software development cost. It was developed by Barry Boehm in 1981 [2]. This model is one of the most documented and best known in the literature. There exist three versions of the model namely the basic COCOMO model, intermediate COCOMO model and the detailed COCOMO model [4]. This model distinguishes three types of projects: organic, semi-detached and embedded [4]. To obtain the best precision of estimation, a well detailed estimation model has to be developed. In the case of the present study, was selected the intermediate COCOMO model. This model was found the most adapted to the present context.

## II. EVALUATING THE COST TO ADAPT TRIADE

Estimating the cost of adaptation proceeds as follows:

- Evaluate the cost to develop a basic version of TRIADE;
- Estimate the number of lines of code of the adapted version of TRIADE and find the difference between the numbers of lines of codes of these two versions;
- Finally, evaluate the development cost of the difference between the numbers of lines of code of these two versions.

### *A. Evaluation of the development costs of basic TRIADE*

To evaluate the development cost of the basic version of TRIADE, 15 cost factors of the COCOMO intermediate model have been identified on the one hand and on the other hand, the number of lines of code of TRIADE were found and its development cost was evaluated. The TRIADE corresponds to the semi-attached project type.

#### *1) Estimate of the cost factors of COCOMO intermediate model*

- RELY (Required Software Reliability): This cost factor is part of the product attribute (quality of the software). Product attributes describe the environment in which the program operates. This application is suitable at the programming level as well as for the server than for the customer; the time of loading is reduced for both sides. The security needs are:

- The server should be secured (firewall, …);

-The server should be secured. The TRIADE code does not take in charge the security level of the two sides;

- The management of the security using passwords;

- Encode passwords in the data base;

- Management of live attacks (temporization in the passwords renews in the case of error) the musher are the errors the longer is the password renew procedure;

- Management of the blacklist (if the account tries to reach a page to which it is not allowed);

- Management of security via the server side session, nothing is stored on the customer post;

The linguistic value corresponding to this application is high (1.15).

- DATA (Data Base Size): This cost factor belongs equally to the product attribute (the product size) and shows data base size influence on the software development cost. This application is divided into modules which are mostly the same in the way they work. The corresponding linguistic value is medium (1.00).

- CPLX (Product Complexity): This factor is part of product attributes. The complexity of the TRIADE dwells in the code comprehension. The lines of code are not commented and the understanding of the application use was not easy. The used algorithms were complex (parallel treatments, complex data management …). The TRIADE complexity is high (1.15).

- TIME (Execution Time Constraint): This factor is part of computer (platform) attributes and expresses the effect of time resources needed by the software on the development cost. Computer attributes describe the relationship between a program and its host or development computer. This application has been developed on a 2 GHz frequency computer. The linguistic value corresponding to the TIME factor in the present application is medium (50 %).

- STOR (Main Storage Constraint): This factor is part of computer attributes and expresses the effect of spatial resources needed by the software on the development cost. This application has been developed on a 2 Go RAM memory. The linguistic value corresponding to the STOR factor in this application is medium (50 %).

- VIRT (Virtual Machine Volatility): This factor is part of computer attributes. The development cost is high if the virtual machine is highly volatile. The VIRT factor had not any effect on the development cost of this application.

- TURN (Computer Turnaround Time): This factor is part of computer attributes. This factor expresses the effect of the response time of the system on which the software is developed, in relation to the development cost of software. Thus, the higher is the response time, the higher is the development cost. The computer used to develop the present application was a 2 Go RAM memory and 2 GHz frequency computer and consequently, the response time is low (0.87).

- VEXP (virtual Machine Experience): This factor is part of personnel attributes and expresses the effect of the experience of the developers in the use of the virtual machine on the software development cost. Personnel attributes describe the capability and experience of personnel assigned to the project. Thus, the higher is the experience; the lower is the development cost. This factor has no effect on the development cost of this application.

- ACAP (Analyst Capability): This factor is part of personnel attributes and expresses the effect of the competence of the analysts on the development cost. Thus, the weaker is the competence, the higher is the software's development cost. The adaptation of TRIADE needs a high level of competence for the analysts precisely in the identification phase of the "true" needs and the conceptions' modification phase (75% or 0.86).

- AEXP (Application Experience): This factor is part of personnel attributes. The adaptation of TRIADE requires a strong experience in the web application domain (0.91).

- PCAP (Programmer Capability): This factor is part of personnel attributes and expresses the effect of the programmers' competence on the development cost. The development of TRIADE requires a strong knowledge in web programming (75 % or 0.86).

- LEXP (Programming Language Experience): This factor is part of personnel attributes and expresses the effect of experience of the personnel in the programming language on the software development cost. Consequently, the development of TRIADE requires a mastery of PHP, advance JavaScript and MySQL. The corresponding linguistic value is high (0.95).

- MODP (Modern Programming Practices): This factor is part of project attributes. Project attributes describe selected project management facets of a program. TRIADE is developed in its majority in PHP. PHP has an exceptions' management similar to the ones offered by the other programming languages. The development of this application required knowledge of the different stages of reverse engineering. For this, the internal functioning of all the modules has been studied and consequently knowledge on PHP, JavaScript and MySQL was necessary. The corresponding linguistic value was medium.

- TOOL (Use of Software Tool): This factor is part of project attributes and expresses the effect of software tools used by the development team on the development cost. For this development, free software's such as Notepad++; Firefox, Firebug an extension of Firefox have been used.

- SCED (Required Development Schedule): The SCED factor is part of project attributes and expresses the effect of the shortening or the lengthening of the initial schedule on the software development cost. It generates an increase in cost, in both cases: shortening or lengthening of the schedule.

The SCED factor is measured as the percentage of shortening or lengthening of initial schedule's time. The adaptation of TRIADE no requires shortening or lengthening constraints of the initial schedule. Consequently, this factor had no effect on the development cost of this application.

The table below presents the cost factors and the corresponding linguistic value.

TABLE I. ESTIMATION OF COST FACTORS OF TRIADE FOLLOWING COCOMO INTERMEDIATE MODEL

| Factors | Linguistic values |
|---|---|
| RELY | high |
| DATA | medium |
| CPLX | high |
| TIME | medium (used 50% of execution time) |
| STOR | medium (used 50% of available memory) |
| VIRT | No effect |
| TURN | low |
| VEXP | No effect |
| ACAP | high |
| AEXP | high (three years) |
| PCAP | high (75%) |
| LEXP | high |
| MODP | medium |
| TOOL | No effect on the development cost |
| SCED | No constraints concerning the shortening or the lengthening of the execution time |

Based on this estimation, it is realized that the factors VIRT, TURN, ACAP, AEXP, PCAP and LEXP have positive effect on the development cost of TRIADE.

*1) Estimating the number of lines of code of TRIADE and the evaluation of its development cost*

For estimating the number of lines of code of TRIADE designated by $\Delta_1$, Sloccount has been used. Sloccount (pronounced "SLOC-count") is a program suites used to count the number of lines of code (SLOC) in important software systems. Sloccount is under the GPL license. Thus, Sloccount is a «tool to measure software». Sloccount was developed by David A. Wheeler [7], at the beginning to count the number of lines of code of one of the GNU / Linux distribution, but it can be used for others systems as well. Sloccount is known to work well on LINUX systems, and was tested on Red Hat Linux 6.2, 7 and 7.1 [1]. Sloccount count the number of lines of code of programs designed using many languages. Running this program with TRIADE gave $\Delta_1$ = 1,041,186. The table below presents the different values found based on some group of languages:

TABLE II. NUMBER OF LINES OF CODE FOLLOWING DIFFERENT GROUPS OF LANGUAGES

| Languages | Line of source code (SLOC) |
|---|---|
| php | 891,849 (85.66%) |
| xml | 88,658 (8.52%) |
| ansic | 51,074 (4.91%) |
| java | 8,940 (0.86%) |
| perl | 525 (0.05%) |
| python | 115 (0.01%) |
| sh | 25 (0.00%) |
| TOTAL | 1,041,186 (100%) |

To evaluate the development cost of the basic version of TRIADE, the average salary of a software programmer of the Bureau of Labor Statistics (BLS) in the USA has been used. According to BLS, the average salary for an american software programmer in July 2008 was 75,662.0810 $ [1]. Following the COCOMO intermediate model and taking into account the average salary of 75,662.0810 $ per year, the table below has been obtained:

TABLE III. TOTAL DEVELOPMENT COST OF TRIADE

| | |
|---|---|
| Total Physical Source Line of Code – SLOC | 1,041,186 |
| Estimated development effort in Man-Month (Man-Year) = 2.2069*(KSLOC**1.12)) | 5,289.49 (440.80) |
| Estimated development time in Months (basic COCMO model, Month = 2.5*(Man-Month**0.38)) | 64.98 |
| Number of software developers required (Effort/development time) | 81.40 |
| Estimating the total development cost (average salary = 75,662.08 $ /year, overhead = 2.40) | 80,044,427.67 $ |

### *B. Lines of code of adapted TRIADE and the difference in the number of lines of code between adapted TRIADE and the basic TRIADE*

To find the number of lines of code of the adapted version of TRIADE denoted by $\Delta_2$, the software Sloccount has been also used. The result is $\Delta_2$ = 1,149,345. The table below presents the number of line of code of the adapted version of TRIADE based on different groups of languages:

TABLE IV. NUMBER OF LINES OF CODE OF THE ADAPTED VERSION OF TRIADE FOLLOWING DIFFERENT GROUPS OF LANGUAGES

| **Languages** | **Lines of source code (SLOC)** |
|---|---|
| php | 999,992 (87.01%) |
| xml | 88,662 (7.71%) |
| ansic | 51,074 (4.44%) |
| java | 8,940 (0.78%) |
| perl | 525 (0.05%) |
| python | 115 (0.01%) |
| sh | 25 (0.00%) |
| pascal | 12(0.00%) |
| TOTAL | 1,149,345 (100%) |

The difference of the lines of code $\Delta$, between the adapted version of TRIADE and the basic one is given as follows:

$$\Delta = \Delta_2 - \Delta_1 = 108{,}159 \qquad (1)$$

### *C. The development cost of the difference*

To evaluate the development cost of 108,159 lines of code, the COCOMO intermediate model has been applied and has been considered the average salary of an amarican software developer in July 2008 that was 75,662.0810 $ per year [1]. The following results have been obtained:

TABLE V. TOTAL COST OF DEVELOPMENT OF THE DIFFERENCE OF LINES OF CODE

| | |
|---|---|
| Total Physical Source Line of Code – SLOC | 108,159 |
| Estimated development effort in Man-Month (Man-Year) = 2.2069*(KSLOC**1.12)) | 418.72 (34.9) |
| Estimated development time in Months (basic COCMO model, Month = 2.5*(Man-Month**0.38)) | 24.78 |
| Number of software developers required (Effort/development time) | 16.89 |
| Estimating the total development cost (average salary = 75,662.08 $ /year, overhead = 2.40) | 6,337,455.82 $ |

The total development cost of the adapted version of TRIADE is obtained as follows:

$$Cost_{adapted\_TRIADE} = Cost_{basic\_TRIADE} + Cost_{difference} \qquad (2)$$

= 86,381,883.49 $

The table below summarizes the development cost of the basic version of TRIADE, the development cost of the adapted version and the difference in lines of code between the two versions.

TABLE VI. SUMMARY OF THE DEVELOPMENT COST

| applications | Total cost of development |
|---|---|
| basic TRIADE | 80,044,427.67 $ |
| adapted TRIADE | 86,381,883.49 $ |
| Difference in lines of code | 6,337,455.82 $ |

The effort required to develop the adapted version of TRIADE is in fact the effort required to produce 108.159 lines of code since the basic version of TRIADE was already developed. By evaluating the cost required to adapt this application, it is shown clearly in this article that the development of software based on Open Source can significantly reduce the development cost.

## III. LIMITS OF THE APPROACH TO ASSESS SOFTWARE DEVELOPMENT COST

Estimation models can hardly be always correct in science. Despite the fact that this evaluation model seems better, it presents some relative limits:

- The COCOMO model was designed based on research done on the development of proprietary software. It doesn't take into account some complexities due to the Open Source model, in particular, the testing and the collaboration effort required to develop a project like Linux [1]. This limit could be the focus of forthcoming research on development and/or validation model for estimating the development cost of Open Source Software;
- This study assumes that the development is done in the USA with the average salaries in that country. It can be interesting to assess whether it is interesting to outsource the development to the countries in Asia (like China, India or Vietnam) where it is cheaper to develop software;

## IV. CONCLUSION

This article shows that the development of software based on Open Source components has several advantages particularly the reduced cost. To show that it is relevant, the Open Source application TRIADE has been modified and the cost to modify the application have been evaluated. This evaluation was done following the COCOMO intermediate model. At the end of this study, it was clear that this approach reduces significantly, the development cost. Instead of having an application that would cost 86,381,883.49 $ if it was develop from scratch, the actual application costs only 6,337,455.82 $ because Open Source component provided by TRIADE have been used.

The Open Source development model, based on voluntary contributions for people around the world, allows without lot of investment to get running software. This is particularly important when it comes to debugging, a process very expensive and long in the proprietary world. The possibility given to reuse existing part of the software or to fork an existing project greatly reduces the development cost. Finally, this study shows that the development of applications based on Open Source can be a viable way to develop and adapt existing software to the african context.

## REFERENCES

[1] A. McPherson, B. Proffitt, and R. Hale-Evans, ‘‘Estimating the total development cost of a Linux distribution’’, The Linux Foundation, 2008.
[2] B. BOEHM, ‘‘Software engineering economics’’, Prentice-Hall, 1981.
[3] S. GRIMAITRE, ‘‘Reverse engineering d’une application php,’’ Haute École de Gestion de Genève, 2007.
[4] A. IDRI, ‘‘Un modèle intelligent d’estimation des coûts de développement de logiciels’’, Doctoral Thesis. université du Québec à Montréal, 2003.
[5] L. MOCZAR, ‘‘Vers le monopole Open Source’’, 24 août 2010, http://www.itrmanager.com/ articles/35548/35548.html.
[6] D. K. MOULLA, ‘‘Développement d’applications à bas coût basées sur les Open Source’’, Master’s Thesis, Université de Ngaoundéré au Cameroun, 2011.
[7] D. A. WHEELER, ‘‘Sloccount user’s guide’’, 2004, http://sloccount.html.