International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 1, January-2014, pp: (395-399), Impact Factor: 1.252, Available online at: www.erpublications.com

Mapping Predictive Object Point Metrics (POP) with Software Size in OO Environment

Shubha Jain[1], Vijay Yadav[2], Prof. Raghuraj Singh[3]

[1,2]Department of Computer Science & Engineering, Kanpur Institute of Technology, Kanpur (India)
[3]Department of Computer Science & Engineering, Harcourt Butler Technological Institute, Kanpur (India)

Abstract: Estimating the size of OO software is the main issue today in project planning and development. Accurate estimation of size may lead to precise prediction of effort during the process of system development. Predictive Object Point POPs are a metric designed specifically for estimating the size of object oriented software. But there is no direct relation between POP and Software size. In this paper various issues while mapping POP with software size have been discussed. The factors responsible for the variation in mapping procedure have been identified. Here it is suggested that the POP Count may be mapped to corresponding software size (KLOC) for the projects belong to the same OO environment only which may further help in prediction of effort and cost for that projects.

Keywords: Estimated KLOC, KLOC, Object-Oriented Measurement, Predictive Object Point, SLOC, Software Metrics, Software Sizing, Software Estimation Technique, Mapping factors.

Introduction

Software size measurement continues to be an important issue in the software development process. The Object Oriented (OO) software engineering has now grown as a dominant practice in software industry. The growth of OO practices has required software developers and their managers to rethink the way they have been estimating the size, effort and cost of their development projects [3].

The number of source lines of code (SLOC) has consistently proven to be the most reliable sizing metric to estimate software development and support costs. However, it is difficult to determine SLOC for a conceptualized system without sound structured analysis of an extensive experience base. [4]**.** Many software sizing methods have been suggested and refined by various researchers to address this issue. Due to improvements in Software development process, the effectiveness of SLOC as the best metric has been challenged. Even Function Points are also not the answer to all software measurement, as they come with their own set of limitations.

POP was introduced by Minkiewicz in 1998 PRICE systems [2] has developed the predictive object point (POP) metric for predicting effort required for developing an object oriented software system. This was based on the counting scheme of function point (FP) method. POPs are intended as an improvement over FPs, which were originally intended for use within procedural systems, by drawing on well-known metrics associated with an object oriented system [5**].** POP is considered to be the better indicator of size of object oriented than FP [6, 7] and thus can be used for the evaluation of effort which may be used to find the cost as well as schedule of a software project. POPs are a metric suitable for estimating the size of object oriented software, based on the behaviors that each class is delivering to the system along with top level inputs describing the structure of a system. However there is no real mapping of POP with software size exists.

Mapping Pop Metric with Software Size

The POP is considered to be the better size indicator for object oriented system thus the practitioners may use this metric to estimate the effort required to complete the project. This has been proposed to be done using simple linear regression analysis [1]. The POP and EKLOC are being mapped as in Fig.1 [9].

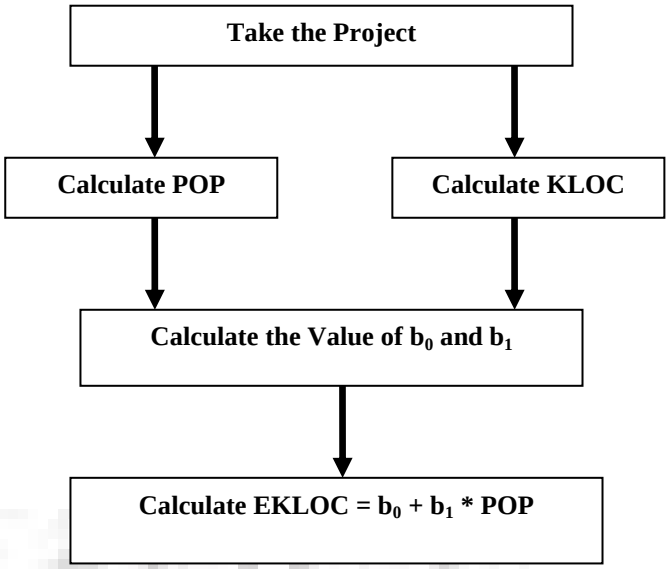

Fig. 1: Steps required for the mapping process [9]

Where,

$b_0 = (\text{avg } Y) + b_1 (\text{avg } X)$

$$b_1 = \frac{(\sum X * Y) - \left[\frac{(\sum X) * (\sum Y)}{n}\right]}{(\sum X^2) - \left[\frac{(\sum X)^2}{n}\right]}$$

Where n is the number of projects and the KLOC obtained from the above equation is termed as Estimated KLOC i.e. EKLOC.

Description of Empirical Study

The nine different versions of a project "The Lightweight Java Game Library (LWJGL)" [8] has been chosen. The KLOC and POP values are obtained through an APA tool [7]. Table 1 shows the values of POP and KLOC measured from the tool. The values of b_0 and b_1 is then calculated by using these values of POP and KLOC and then these values along with POP are used to evaluate finally the Estimated KLOC i.e. EKLOC.

Table 1 KLOC and POP values of 9 versions of Project (LWJGL) measured through APA Tool

Project No.	LWJGL(Version)	KLOC	POP
1	lwjgl_0.9	14.119	2419.3199
2	lwjgl_0.92	18.262	3176.9945
3	lwjgl_0.93	19.366	3283.3734
4	lwjgl_0.94	20.624	4074.6891
5	lwjgl _0.96-2	23.58	5349.5434
6	lwjgl_0.98-1	24.64	5345.8681
7	lwjgl_1.0	30.097	6045.3585
8	lwjgl_1.0 rc-1	30.542	5867.1778
9	lwjgl_1.1.1	33.277	6153.8996

The above values have been plotted as shown in Fig.2 in order to get the values of slope (b_1) and intercept (b_0).

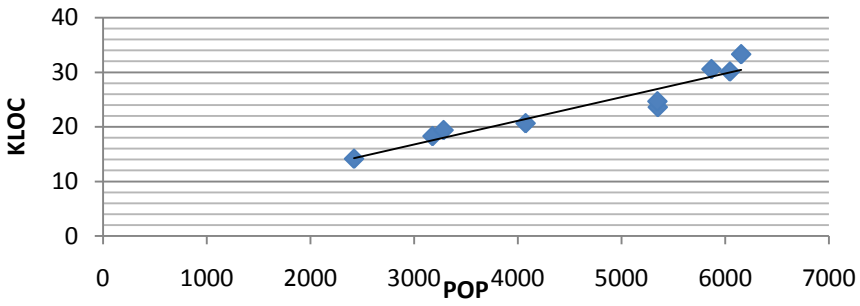

Fig. 2: A Graph between POP and KLOC of 9 versions of project (LWJGL)

To strengthen the above relation obtained between EKLOC and POP, more versions of the projects are analyzed and incorporated in the calculation of b_0 and b_1.

After including two more versions of the project, the b_0 and b_1 values are refined further, giving more closely the values of EKLOC to actual KLOC as shown in Table 2.

Table 2: KLOC, POP and Estimated KLOC values of the 11 versions of project (LWJGL)

Projects	Actual KLOC	POP	Estimated KLOC (EKLOC)
1	14.119	2419.3199	13.9983754
2	18.262	3176.9945	17.487573
3	19.366	3283.3734	17.9774627
4	20.624	4074.6891	21.6215823
5	23.58	5349.5434	27.4924648
6	24.64	5345.8681	27.4755395
7	30.097	6045.3585	30.6967907
8	30.542	5867.1778	29.8762436
9	33.277	6153.8996	31.1966377
10*	33.31	6166.1952	31.2532606
11*	30.004	5621.5449	28.7450697

*The new projects (higher versions of LWJGL) incorporated are of same type in estimation of EKLOC.

$b_0 = 2.857069$
$b_1 = 0.004605$
$EKLOC = 2.857069 + 0.004605 * POP$

The result showed improvement with increase in number of versions incorporation in regression analysis.
Continuous incorporation of more versions of the project and hence further refinement in values of b_0 and b_1 take Estimated KLOC to more close to the actual KLOC value of that version. The results may be seen through a graph plotted between Estimated KLOC and POP as shown in Fig.4.

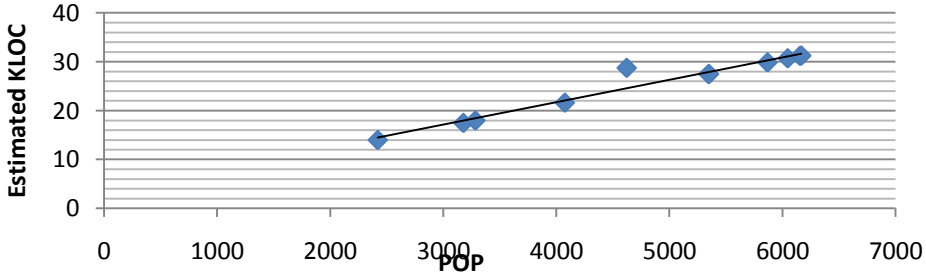

Fig. 4: A Graph between EKLOC and POP for 11 versions of chosen project (LWJGL)

Now two different types of projects (developed for different application) are incorporated in the calculation of b_0 and b_1. After including these projects, the b_0 and b_1 values are changed, giving variation in the values of EKLOC and actual KLOC as shown in Table 3.

Table 3: KLOC, POP and Estimated KLOC values after incorporation of different projects in 9 versions of project (LWJGL)

Projects	Actual KLOC	POP	Estimated KLOC (EKLOC)
1*	7.264	2627.5415	13.2618058
2*	12.23	2272.0277	11.4931904
3	14.119	2419.9319	12.2289864
4	18.262	3176.9945	15.995233
5	19.366	3283.3734	16.5244484
6	20.624	4074.6891	20.4610978
7	23.58	5349.5434	26.8032623
8	24.64	5345.8681	26.7849784
9	30.097	6045.3585	30.2648139
10	30.542	5867.1778	29.3783978
11	33.277	6153.8996	30.8047858

*The new projects 1and 2 (jasypt-1.9.1-sources, jsch-0.1.49) incorporated are of different type in estimation of EKLOC.

$b_0 = 0.190272$
$b_1 = 0.004975$
$EKLOC = 0.190272 + 0.004975 * POP$

Analysis and Results

In above study, the projects chosen were developed in the same environments and hence it may be observed that each time when a new project of the same type is incorporated, the values of b_0 and b_1 are updated which in turn updates the regression equation $EKLOC = b_0 + b_1 * POP$. It is found that the values of Estimated KLOC also changes each time and comes nearer to the actual KLOC values of the projects which signifies that the accuracy of the equation suggested for mapping increases every time as more number of projects of the same type are considered. However, this will not be the case for different types of projects. The reason for this can be attributed to the different software development environments in which the software has been built. The environment imposed by the project manager is a major driver in this variation. The first step in evaluating the development environment is to determine the developer's raw capability. This measure normally varies only when there is a major change in the corporate approach to software development. Hence this remains same for the projects developed in same environment. In addition to this there are certain factors which affect the development of a project and hence affect size estimation process. Some of the factors are:

Technical factors

Product Complexity
Reusability Required
Platform Compatibility
Adaptability
Facility included if any like security features, Special user training facilities etc.
Use of software tool
Programming language used
Database size and storage constraints

Environmental factors

Programmer's working experience
Lead Analyst capability
Motivational aspects
Reliability required
Part-time workers

Therefore from the above analysis it is clear that a mapping can be done between KLOC and POP by using simple linear regression equation only for the projects which are developed in the same environment and are built for the same type of application.

Conclusions

From the results presented in this study it is found that Predictive Object Point Metrics can be used to estimate the size of the Object Oriented Software Systems only for the projects which are developed in the same environment and are built for the same application. The results through this mapping will give more accurate results on adding up more and more projects for calculation through regression equation. However POP may not be related to the size through the same formulation for different types of projects built for different applications. The reason for this can be attributed to the various technical and environmental factors.

Hence there is consequently a need for further studies on the precision of the size estimates when using the mapping method in different types of projects. Therefore, such factors should be given due weightage and then should be added up to the mapping procedure through some formulation in order to provide a final estimate.

References

[1]. Online Statistics Education: An Interactive Multimedia Course of Study, Developed by Rice University (Lead Developer), University of Houston Clear Lake, and Tufts University, URL: http://onlinestatbook.com/2/regression/ intro.htm
[2]. Arlene F. Minkiewicz, "Measuring Object Oriented Software with Predictive Object Points" PRICE Systems, L.L.C. 609-866-6576, 1997.
[3]. Ghazy Assassa, Hatim Aboalsamh, Amel Al Hussan. " Software Cost Estimation: A Tool for Object Oriented Console Applications".
[4]. Anthony A. DeMarco . " Function Point Software Sizing". Director, PRICE Systems.
[5]. M.Haug, E.W. Olsen,L. Bergman,"Software Best Practices", Springer 2001, Pg 159-170.
[6]. T. R. Judge, A. Williams ," OO Estimation – an Investigation Of The Predictive Object Points (POP) Sizing Metric in an Industrial Setting", Parallax Solutions Ltd, Coventry, UK.
[7]. Shubha Jain, Vijay Yadav and Prof. Raghuraj Singh, "OO Estimation Through Automation of Predictive Objective Points Sizing Metric", International Journal Of Computer Engineering and Technology (IJCET) Volume 4, Issue 3, May- June (2013), pp. 410-418.
[8]. "The Lightweight Java Game Library (LWJGL) Project", Available at: http://sourceforge.net/ projects / java-game-lib/files/? Source = navbar.
[9]. Shubha Jain, Vijay Yadav and Prof. Raghuraj Singh, "An Approach for OO Software Size Estimation using Predictive Object Point Metrics", 8th Indiacom-2014; International conference (IEEE conference ID 32558) on Computing for Sustainable Global Development (paper accepted).