Volume 4, Issue 3, March 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Identifying a Direct Relationship between Predictive Object Point Metrics (POP) Count and Source Lines of Code (SLOC) for OO Software

Shubha Jain
Department of Computer Science and Engineering
Kanpur Institute of Technology,
Kanpur, India

Prof. Raghuraj Singh
Computer Science and Engineering Department
Harcourt Butler Technological Institute,
Kanpur, India

Abstract— Estimation of system size is the main issue for software developers during system development planning. It is very important in predicting the efforts precisely during the process of system development. The Predictive Object Point (POPs) input is an estimate of the size (new, adapted, modified or deleted) of the software for which the estimate is being performed. POPs are a metric suitable for estimating the size of object oriented software, based on the behaviours that each class is delivering to the system along with top level inputs describing the structure of a system. While there is no real mapping of SLOC to POPs exist. In this paper an attempt has been made to map directly the Predictive Object Point Metrics with Source Line of Code. Various projects have been taken for the validation of proposed direct relation between POP and SLOC.

Keywords— Object-Oriented Measurement, Predictive Object Point Metric, POP Count, SLOC, Software Metrics, Software Sizing, Software Estimation Technique, Size Mapping.

I. INTRODUCTION

Software sizing is a process is used to estimate the size of software, which is an important factor that affects the cost and time of the software project [3, 4, 5, 6, 7]. Predictive Object Points (POP) [2] is a sizing metric which is considered to be the better indicator of size of object oriented software than any other sizing measures like Function Points [8, 9]. POPs are a metric suitable for estimating the size of object oriented software however there is no real mapping of POP with software size exists.

II. MAPPING POP METRIC WITH SOFTWARE SIZE

The practitioners may use POP metric to estimate the effort required to complete the project by using the COCOMO II model [3, 10]. As the model uses Kilo Source lines of code for effort estimation therefore a technique was required to convert POP to Source Lines of Code. This has been proposed to be done using simple linear regression analysis [1].
A mapping was suggested between KLOC and POP by using simple linear regression equation [12]. The method gave more accurate results when more and more number of the project is added up in the process.

It is also found that Predictive Object Point Metrics can be used to estimate the size of the Object Oriented software Systems through regression method only for the projects which are developed in the same environment and are built for the same application [13]. However POP may not be related to the size through the same formulation for different types of projects built for different applications. Hence again no direct relation between POP and Software size is found in terms of Source Lines of Code.

III. DESCRIPTION OF EMPIRICAL STUDY

For understanding the relationship between the POP count and SLOC, an exhaustive range of projects [11] have been chosen. These are classified in two categories. The one with projects of size less than 5000 SLOC and the other with the projects of size more than 5000 SLOC. The SLOC and POP values are obtained through an APA tool [9]. Table I shows the projects with up to 5000 Source Lines of Code (SLOC) along with their SLOC, TLC and POP Count values measured through the Tool.

TABLE I
ANALYSIS OF PROJECTS WITH SLOC LESS THAN 5000

S.No.	Project Name	SLOC	TLC	POP Count
1	PhysicsMata_ver_0.1.2	169	2	44.2279
2	PhysicsMata_ver_0..6.0	224	6	24.124
3	PhysicsMata_ver_0..6.1	227	6	24.124
4	PhysicsMata_ver_0.8.0	227	7	24.124
5	PhysicsMata_ver_0.8.1	230	7	24.124
6	JavaMP4BoxGui_v1.4	235	4	44.23

7	PhysicsMata_ver_0.3.1	237	7	128.64
8	PhysicsMata_ver_0.3	239	7	128.64
9	JavaMP4BoxGui_v1.5	249	4	44.23
10	JavaMP4BoxGui_v1.6	255	4	44.23
11	JavaMP4BoxGui_v1.0	295	9	90.16
12	PhysicsMata_ver_0.1.3	319	4	96.4973
13	FourRowSolitaire-v.01	347	6	44.23
14	Remote_Adm_Sys	383	12	41.54
15	PhysicsMata_ver_0.5.0	398	9	249.2617
16	JavaMP4BoxGui_v1.7	404	8	48.25
17	PhysicsMata_ver_0.5.1	416	9	253.2828
18	PhysicsMata_ver_1.2.0	418	15	56.29
19	PhysicsMata_ver_1.2.1	419	15	56.29
20	PhysicsMata_ver_0.4.1	492	32	208.712
21	FourRowSolitaire-v.02	551	10	128.66
22	Face_Detection_Syst	600	2	50.0243
23	Online_Address_Book	614	12	160.829
24	Galo_0.7a	685	8	119.85
25	Face_book_Like_Chat	695	9	156.554
26	Flight_Reserv_System	719	3	86.4455
27	CLIPSJNI_0.1	1133	17	252.15
28	ATM_Banking_Sys	1186	6	119.2381
29	CLIPSJNI_0.2	1553	18	274.26
30	HBX-1.0-src	1637	10	337.04
31	CB 1.0.0	1830	19	419.88
32	Payroll	1834	13	312.3165
33	CLIPSJNI_0.3	1864	18	397.63
34	CB 1.1	1888	19	425.92
35	HBX-1.1-src	1987	11	391.09
36	CB 1.2	2036	20	451.27
37	HBX-1.8-src	2048	12	441.44
38	CB 1.3	2052	21	437.19
39	HBX-1.7-src	2060	12	438.76
40	HBX-1.9.1-src	2071	12	441.44
41	HBX-1.9-src	2073	12	441.44
42	HBX-1.10-src	2253	11	459.45
43	HBX-1.12.1-src	2267	11	462.13
44	HBX-1.11-src	2270	11	462.13
45	HBX-1.14.1-src	2309	11	462.13
46	JaimBot_Ver_1.2	2314	20	561.852
47	File_Compression	2319	22	620.573
48	Civil_Game_Java	2659	18	510.9983
49	JaimBot_Ver_1.2.1	2663	21	597.89
50	HBX-1.0-src	2837	12	528.21
51	JaimBot_Ver_1.3	2880	23	661.55
52	JaimBot_Ver_1.4	4413	33	1000.88

The behaviour in terms of POP Count of the above projects may be seen in Fig.1 with respect to SLOC. This is clearly visible that the POP Count is around 0.2 times of SLOC.

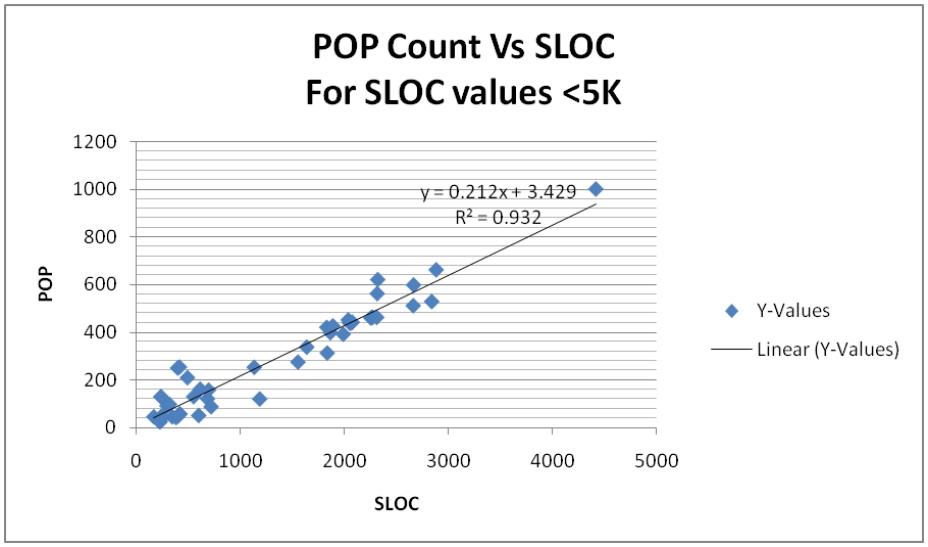

Fig. 1 A Graph showing behaviour of POP Count with SLOC with SLOC values less than 5000

Table II shows the projects with more than 5000 Source Lines of Code (SLOC) along with their SLOC, TLC and POP Count values measured through the Tool. The largest project taken here for study is with Source Lines of Code 34688.

TABLE II
ANALYSIS OF PROJECTS WITH SLOC MORE THAN 5000

S.No.	Project Name	SLOC	TLC	POP Count
1	Lwjgl_alpha_0.3	6463	22	1256.55
2	Lwjgl_0.5	7578	25	1380.07
3	borg_1.3	7834	32	1260.94
4	borg_0.9.4	8229	32	911.084
5	borg_1.0	8537	32	935.5
6	Lwjgl_0.4	9397	39	1705.75
7	borg_1.1	10146	36	1329.09
8	Lwjgl_0.6	10363	48	2222.93
9	borg_1.2	11408	38	1315.44
10	borg_1.7	14206	197	2752.22
11	borg_1.3.1	15179	57	2012.69
12	borg_1.3.2	15390	57	2065.47
13	Lwjgl_0.92	18262	96	3978.5
14	Lwjgl_0.93	19366	96	4001.29
15	borg_1.4	19481	102	3520.58
16	Lwjgl_0.95	19555	116	5526.39
17	Lwjgl_0.96-2	23580	163	6914.9
18	Lwjgl_0.97-1	23682	163	6941.03
19	Lwjgl_0.98-1	24640	165	6915.59
20	Lwjgl_0.99	25744	164	7140.75
23	Lwjgl_1.0beta4	29558	179	7777.62
24	Lwjgl_1.0	30097	179	7805.37
25	Lwjgl_1.0 -rc1	30542	179	7792.5
26	Lwjgl_1.1	32219	184	7751.89
27	borg_1.6.1	34688	219	7938.31

The behaviour in terms of POP Count of the above projects may be seen in Fig.2 with respect to SLOC. This is clearly visible that the POP Count is around 0.3 times of SLOC.

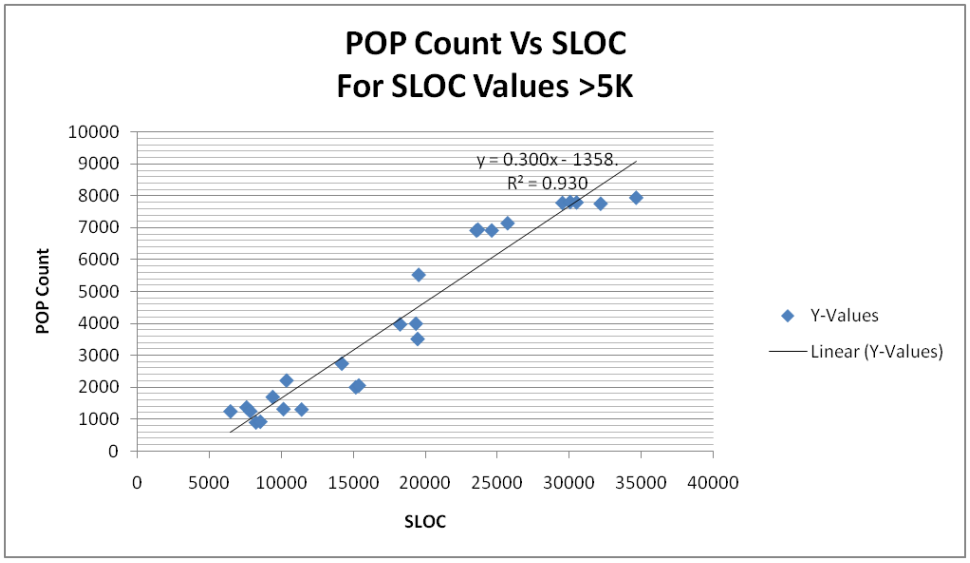

Fig.2 A Graph showing behaviour of POP Count with SLOC with SLOC values greater than 5000

IV. ANALYSIS AND RESULTS

In above study, it may be observed that the projects with less than 5000 SLOC shown the POP count as 0.2 times of that of SLOC. However this shows 0.3 times of SLOC for the project size greater than the 5000 SLOC. It may be generalized to say that the POP Count is 0.2 to 0.3 times of source line of code for any project. Thus a POPs count of 600 is generally equivalent to a system with 2,000 to 3,000 SLOC.

V. 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. Here the nature of the relationships between KLOC and POP in object-oriented (OO) systems has been understood. This study depicts the relationship between POP and Source Lines of Code (SLOC) to get estimation of size of software through POP. Though there is no real mapping of SLOC to POPs exist but this is found that usually SLOC is 4 to 5 times of POP count measured for a project. Hence 1 POP Count can be considered equivalent to 4 to 5 Source Line of Code.

However, still more number of projects under various categories, may be taken for analysis in order to ensure the validity for this relationship and hence accuracy of the software size measurement.

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.html

[2] Arlene F. Minkiewicz, "Measuring Object Oriented Software with Predictive Object Points" PRICE Systems, L.L.C. 609-866-6576, 1997.

[3] B. W. Boehm, Estimating Software Costs, Prentice Hall, N.J., 1981,

[4] T. C. Jones, Estimating Software Costs, McGraw-Hill, New York, 1998.

[5] J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation", IEEE Transactions on Software Engineering, vol. 9,no. 6, pp. 639–648, 1983.

[6] F. Kemerer, "An Empirical Validation of Software Cost Estimation Models", Communications of the ACM, vol.30, no. 5, pp. 416-429, 1987.

[7] Kitchenham, "Using Function Points for Software Cost Estimation- Some Empirical Results", 10thAnnual Conference of Software Metrics and Quality Assurance in Industry, Amsterdam, the Netherlands, 1993.

[8] 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.

[9] Shubha Jain, Vijay Yadav and Prof. Raghuraj Singh, (2013). "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.

[10] Center for Software Engineering, (2000) "COCOMO II Model Definition Manual," Computer Science Department, University of Southern California, Los Angeles, Ca.90089, Available At: http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html

[11] "The Lightweight Java Game Library (LWJGL) Project", Available at: http://sourceforge.net/ projects / java-game-lib/files/? Source = navbar

[12] Shubha Jain, Vijay Yadav, Raghuraj Singh, "An Approach for OO Software Size Estimation using Predictive Object Point Metrics", INDIACom-2014; 8th INDIACom- 2014; International Conference on "Computing for Sustainable Global Development, ISSN 0973-7529 and ISBN 978-93-80544-10-6 serials, IEEE Conference ID 32558, At Bharati Vidyapeeth Institute of Computer Applications and Management, New Delhi, India.

[13] Shubha Jain, Vijay Yadav, Prof. Raghuraj Singh, "Mapping Predictive Object Points (POP) with Software Size in OO Environment", International Journal of Enhanced Research in Science, Technology & Engineering, ISSN No: 2319-7463, Vol. 3, Issue 1, January-2014.

