Empirical Analysis of Function Point Analysis – Research View

Vuppu Padmakar
Department of CSE
Ph D Research Scholar
OPJS University, Rajasthan
padmaker@gmail.com

Dr. B V Ramana Murthy
Department of CSE, Professor
Mahaveer Institute of Science & Technology, Hyderabad
drbvrm@gmail.com

Dr. Vaibhav Bansal
Department of CSE
Associate Professor
OPJS University, Rajasthan
drvb.cse@gmail.co

Abstract: Software measurement [1], once an obscure and esoteric specialty, has become essential to good software engineering [2, 3, 4]. Many of the best software developers measure characteristics of the software to get some sense of whether the requirements are consistent and complete, whether the design is of high quality, and whether the code is ready to be tested. Effective project manager's measure attributes of process and product to be able to tell when the software will be ready for delivery and whether the budget will be exceeded. Informed customers measure aspects of the final product to determine if it meets the requirements and is of sufficient quality. And maintainers must be able to assess the current product to see what should be upgraded and improved. Here, we present empirical research in two areas, Function Oriented Analysis to find function points and weightages of plan driven vs agile development based on type of project, validations will be carried out using train test procedure for FPA and Ideal Point Analysis for weightages

Keywords: *FPA, metrics, size estimations, code estimation.*

*****

I. INTRODUCTION & RELATED WORK.

Metrics for the Analysis Model:- These metrics address various aspects of the analysis model and include:

- Functionality Delivered: Provides an indirect measure of the functionality that is packaged with in the software.
- System Size: measures[5] of the overall size of the system defined in terms of information available as part of the analysis model[7].
- Specification Quality: provides an indication of the specificity and completeness of a requirements specification.

Function-Based Metrics: The Function Point Metric [FP] [7, 8, 9, 10], first proposed by Albrecht, can be used effectively as a means for measuring[6] the functionality delivered by a system. Using historical data, the FP can then be used to

(1) estimate the cost or effort required to design, code, and test the software;

(2) predict the number of errors that will be encountered during testing.

(3) forecast the number of components and/or the number of projected source lines in the implemented system.

Function points are delivered using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity. Information domain values are defined in the following manner:

Number of external inputs (EIs): Each external input originates from a user or is transmitted from another application and provides distinct application-oriented data or control information. Inputs are often used to update internal logical files (ILFs). Inputs should be distinguished from inquiries, which are counted separately.

Number of external outputs (EOs): Each external output is derived within the application and provides information to the user. In this context external output refers to reports, screens, error messages, and so on. Individual data items within a report are not counted separately.

Number of external inquiries (EQs): An external inquiry is defined as an online input that results in the generation of some immediate software response in the form of an on-line output (often retrieved from an ILF).

Number of internal logical files (ILFs): Each internal logical file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

Number of external interface files (ELFs): Each external interface file is a logical grouping of data that resides external to the application but provides data that may be of use to the application.

Sample projects of 'C' Source Code

As mentioned fellow the 'C' source code of the 20 projects total number of lines are counted physically.

S.No.	Project ID	Project Name	FP	NLOC
1	P-1	CPA Calculation	21.75	261
2	P-2	Travel AirBase Agency	12.23	159
3.	P-3	Hospital Management System	49.5	693
4	P-4	Banking Management System	47.61	619
5	P-5	Calendar Project	41.07	575
6	P-6	Contact Management System	12.75	153
7	P-7	Cyber Management System	172.5	2415
8	P-8	Department Store Management System	46.5	696
9	P-9	Employee Record System	29.5	354
10	P-10	Library Management system	52.64	737
11	P-11	Medical Store Management System	171.5	2401
12	P-12	Personal Diary Management System	47.42	664
13	P-13	Phonebook Project	23.33	280
14	P-14	School Billing System	73.61	957
15	P-15	Snake Game	34.38	447
16	P-16	Student Record System	41.08	493
17	P-17	Telecom Billing System	51.57	218
18	P-18	Tic-Tac-Toe Game	17.0	221
19	P-19	Modern Periodic Table	27.78	389
20	P-20	Snakes and Ladders Game	56.71	794
			1030.43	13526

Each FP is calculated by the above process and NLOC non commented lines of code is corresponding the each FP.

Empirical validation process

(i) The aim of the statistical validation is the determine whether the function point metric can be used to predict the size of FO system in terms of LOC.

(ii) From the data collected we have to perform descriptive analysis of both for the dependent variable LOC and the independent variable FP.
i.e., NLOC = a + b * FP
From the data sets finding metric

Step 1: $\sum FP = 1030.43$

Step 2: $\sum FP^2 = 1061785.98$

Step3: $\sum LOC = 13526$

Step4: $\sum FP * \sum LOC = 13937596.18$

Step5: $\sum LOC = n a + b \sum FP$

13526 = 20 a + b * 1030.43 ---------- (1)

Step6: $\sum FP * \sum LOC = a * \sum FP + b * \sum FP^2$

13937596.18 = 20 * 1030.43 + b * 1061785.98 ----- (2)

= 20608.6 + b * 1061785.98

b = 13916987.58 /1061785.98

= 13.107

13526 = 20 a + 13.107 * 1030.43

= 20 a + 13505.84

a = 13526 – 13505.84 / 20

= 1.08

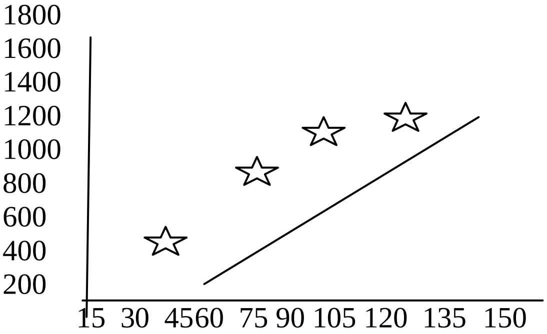
From equation (1) and (2)

a = 1.08 b = 13.107

We get

NLOC = 1.08 + 13.107 * FP

That is by observation for each FP approximately it (involves the size estimation) 16 lines of code Scatter diagram and least square regression line of FP and NLOC. Process of finding metric of size estimation based on FP independent metric (derived from design) and dependent variable (LOC).



Finding variation (Coefficient of determination)

$$NLOC = \frac{\sum NLOC}{N}$$

Where 'N' is the data sets lie 20

$$NLOC = \frac{13526}{20} = 676.3$$

NLOC = a + b * FP

NLOC = 1.08 + 13.107 * FP

Calculating the FP based on the original 'C' source code projects based on the number of line of code.

FP	NLOC	NLOC: NLOC	(LOC: NLOC)²
21.75	261	-415.30	172474.09
12.23	159	-517.30	267599.29
49.5	693	16.70	278.89
47.61	619	-57.30	3283.29
41.07	575	-101.30	10261.69
12.75	153	-523.30	273842.89
172.5	2415	1738.70	3023077.69
46.5	696	19.70	388.09
29.5	354	-322.30	103877.29
52.64	737	60.70	3684.49
171.5	2401	1724.70	2974590.09
47.42	664	-12.30	151.29
23.33	280	-396.30	157053.69
73.61	957	280.70	78792.49
34.38	447	-229.30	52578.49
41.08	493	-183.30	33598.89
51.57	218	-458.30	210038.89
17	221	-455.30	207298.09
27.78	389	-287.30	82541.29
			7655410.91

4.5 Determination of NLOC

LOC	LOC-NLOC	(NLOC-NLOC)²
286.16	25.16	632.89
161.38	2.38	5.66
649.88	-43.12	1859.64
625.10	6.10	37.26
539.38	-35.62	1268.46
168.19	15.19	230.87
2262.04	-152.96	23397.53
610.56	-85.44	7300.76
387.74	33.74	1138.15
691.03	-45.97	2113.01
2248.93	-152.07	23125.13
622.61	-41.39	1712.81
306.87	26.87	721.80
965.89	8.89	78.97
451.70	4.70	22.08
539.52	46.52	2163.70
677.01	459.01	210688.33
223.90	2.90	8.40
365.19	-23.81	566.80
		277072.25

The coefficient of determination

$$= \frac{\mathbf{7655410.91 - 277072.25}}{\mathbf{7655410.90}}$$

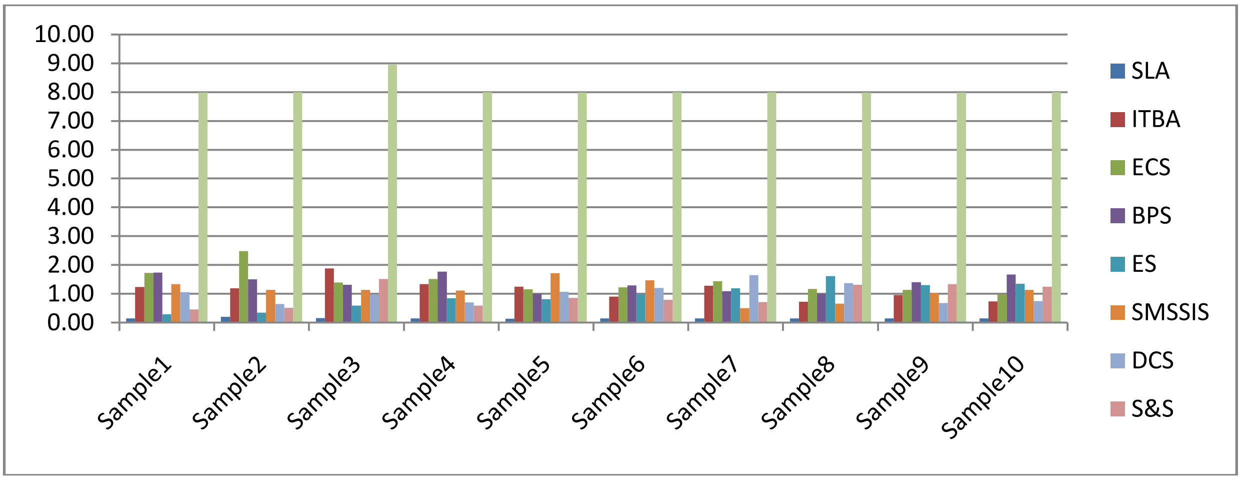
$$= \mathbf{0.96}$$

	Sample1	Sample2	Sample3	Sample4	Sample5	Sample6	Sample7	Sample8	Sample9	Sample10
SLA	0.15	0.20	0.16	0.15	0.13	0.15	0.15	0.15	0.14	0.15
ITBA	1.24	1.19	1.88	1.33	1.25	0.90	1.28	0.72	0.96	0.73
ECS	1.72	2.48	1.39	1.51	1.16	1.22	1.43	1.17	1.14	0.98
BPS	1.73	1.50	1.31	1.77	1.00	1.29	1.09	1.00	1.40	1.67
ES	0.29	0.34	0.59	0.84	0.81	0.98	1.19	1.61	1.30	1.35
SMSSIS	1.34	1.14	1.14	1.11	1.71	1.47	0.50	0.66	1.04	1.13
DCS	1.06	0.64	0.97	0.70	1.07	1.20	1.65	1.37	0.68	0.74
S&S	0.46	0.51	1.51	0.59	0.86	0.79	0.71	1.31	1.33	1.25
	7.99	8.00	8.95	8.00	7.99	8.00	8.00	7.99	7.99	8.00

weights of 'C' different projects of 10 samples

The value of r^2 = 0.96 shows that the regression equation LOC = 1.08 + FP * 13.107 explains about 96.205 percent of the total navigation observed in the dependent variable. Thus, only 3.795 percent of the total variation in the dependent variable NLOC MC certy * Lecernt – chose

Weights of the 10 sample projects data of graphs for different projects

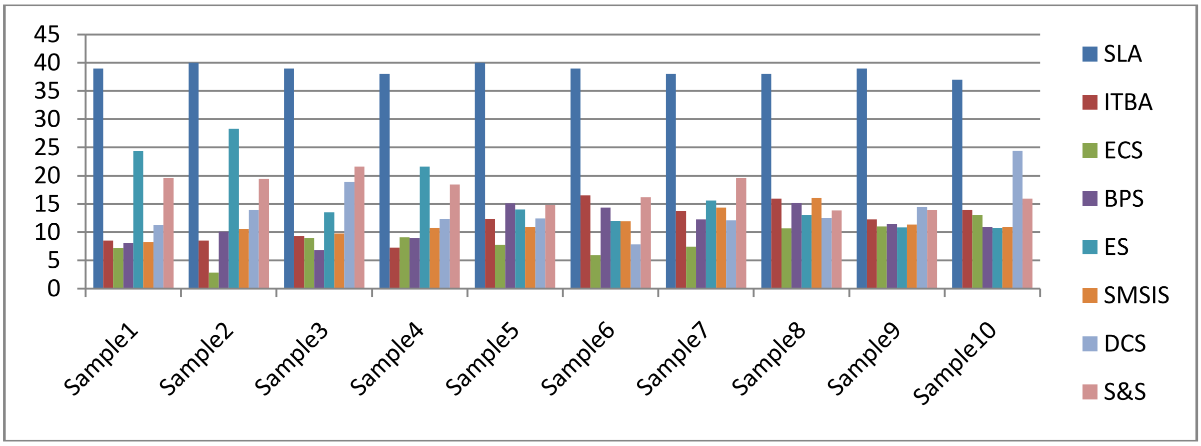


weights of 'C' different projects of sample 10

Different projects 10 sample graphs

	SLA	ITBA	ECS	BPS	ES	SMSIS	DCS	S&S	
Sample1	39	8.53	7.2	8.09	24.33	8.23	11.25	19.58	126.21
Sample2	40	8.51	2.82	10.12	28.33	10.58	13.95	19.45	133.76
Sample3	39	9.28	8.98	6.81	13.5	9.78	18.89	21.64	127.88
Sample4	38	7.29	9.08	8.98	21.64	10.78	12.33	18.45	126.55
Sample5	40	12.37	7.75	15.07	14	10.87	12.41	14.83	127.3
Sample6	39	16.5	5.9	14.33	12	11.92	7.84	16.17	123.66
Sample7	38	13.74	7.42	12.23	15.59	14.33	12.06	19.58	132.95
Sample8	38	15.92	10.65	15.14	13.01	16.08	12.49	13.87	135.16
Sample9	39	12.23	11	11.45	10.83	11.33	14.48	13.93	124.25
Sample10	37	13.95	13	10.9	10.73	10.89	24.4	15.95	136.82
	387	118.32	83.8	113.12	163.96	114.79	140.1	173.45	

weights of sample 10 for 'C' different projects



weights of sample 10 for 'C' different projects

ISSN: 2321-8169
124 - 129

II. CONCLUSION

The work described in this thesis combined the research areas, size estimation of the Software projects and empirical analysis of software projects. Area1: The objective of size estimation of software projects is to estimate size of LOC for given class diagram. The data collection is made at a macro level, Programming Language source code and class diagram were taken to derive metric. For estimating lines of code of Software Projects, we have considered a linear regression between class point and lines of code. The FP values are obtained from a given class diagram, when subjected to FP calculation process, the lines of code is a dependent variable which is estimated as follows,

$$LOC = a + b * FP,$$

Where a , b, are extracted from Research Work. We have shaped the metric for size estimation of entire application as well as the components of the application like Problem Domain Type, Human Interaction Type, and Data Management Type. The validation of the metric can be better tested by using empirical validation; we used techniques like variances and Productivity Analysis for success rate depending upon the threshold value. To fix the metric (LOC = a + b * FP) we have used train and test method, in which the data is divided into two sets , one set is used to find the size metric and another set is used to validate size metric. This is repeated for 'i' items and its average is taken to fix the equation. The results of the study are found to be sound on an empirical analysis.

REFERENCES

[1] Fenton, N., "Software Measurement: A Necessary Scientific Basis", IEEE Trans. Software Engineering, Vol. SE-20, no. 3, March 1994, pp. 199-206.
[2] Humphrey, W., A Discipline for Software Engineering, Addison-Wesley, 1995.
[3] Somerviue, I., Software Engineering, 6th ed. , Addison-Wesley, 2001.
[4] Gilb, T., Principles of Software Engineering Management, Addison-Wesley, 1988.
[5] Putnam, L., and W. Myers, Measures of Excellence, Yourdors Press, 1992.
[6] Felican, L., and G. Zalateu, "Validating Halstead's Theory for Pascal Programs", IEEE Trans. Software Engineering, Vol. SE-15, no. 2, December 1989, pp. 1630-1632.
[7] Albrecht, A. J., and J. E. Gaffney, "Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation", IEEE Trans. Software engineering, November 1983, pp. 639-648.
[8] A. J. Albrecht, "Measuring application development productivity", Proc. of the Joint SHARE / GUIDE / IBM Application Development Symposium, Oct. 1979, pp. 83-92.
[9] A. J. Albrecht, J. E. Gaffney, "Software Function, source lines of code, and development effort prediction: A software science validation ",IEEE Trans. Software Eng. 9, no. 6 (1983) 639-648.
[10] J. B. Dreger, Function Point Analysis, Prentice-Hall. 1989.