REASSESSING FUNCTION POINTS

G.R. Finnie*, G.E. Wittig* and J-M. Desharnais**

* School of Information Technology
Bond University, Gold Coast
Queensland 4229, Australia

** Software Engineering Laboratory in Applied Metrics
7415 rue Beaubien Est, suite 509
Anjou, Quebec, Canada H1M 3R5

ABSTRACT

Accurate estimation of the size and development effort for software projects requires estimation models which can be used early enough in the development life cycle to be of practical value. Function Point Analysis (FPA) has become possibly the most widely used estimation technique in practice. However the technique was developed in the data processing environment of the 1970's and, despite undergoing considerable reassessment and formalisation, still attracts criticism for the weighting scoring it employs and for the way in which the function point score is adapted for specific system characteristics.

This paper reviews the validity of the weighting scheme and the value of adjusting for system characteristics by studying their effect in a sample of 299 software developments. In general the value adjustment scheme does not appear to cater for differences in productivity. The weighting scheme used to adjust system components in terms of being simple, average or complex also appears suspect and should be redesigned to provide a more realistic estimate of system functionality.

INTRODUCTION

Estimating the size of a software project, and hence the project cost and development effort, remains a difficult problem. Considerable research and practical effort has gone into developing models and methodologies to assist in estimation, for example COCOMO, SLIM, Estimacs, Function Point Analysis, (Albrecht, 1979; Albrecht & Gaffney, 1983), SPANS, Checkpoint, (Jones, 1991) and COSTAR, (Ferens & Gurner, 1992). Of these, Function Point Analysis (FPA) has possibly the widest use in practice (Dam & Langbroek, 1992; Dreger, 1989; Jones, 1991).

Suitable estimation models enable a realistic assessment of size early in the systems development life cycle (SDLC). Metrics such as lines of code (LOC) are biased by implementation details and are difficult to assess accurately early in the SDLC. FPA operates by trying to measure the somewhat loosely defined concept of functionality. In FPA functionality is measured by determining the number of inputs, outputs, inquiries, internal files and external files in the target system. Each of these is scored for complexity and weighted accordingly. The total score gives the number of unadjusted function points (UFP) which is then adjusted by a factor composed of 14 General System Characteristics (GSC's) which attempts to compensate for differences in complexity between software projects.

FPA was developed in the data processing environment of the 1970s, (Albrecht, 1979; Albrecht & Gaffney, 1983), and has become more formalised over time. Groups such as IFPUG (IFPUG CPM, 1994), have developed extensive guidelines for computing function points (FP's). However the underlying basis of weighted scoring (and weight values) and the use of specific GSC's (Albrecht & Gaffney, 1983), remains unchanged. The technique has attracted criticism on these grounds (Abran & Robillard, 1994; Kemerer,1987), and attempts have been made to provide alternative functionality based metrics, for example the Symons MkII function points method (Symons, 1991).

This paper describes a critical assessment of the weighting scheme used in FPA and the role and value of the GSC adjustment, and follows on some earlier work done by Desharnais (1988). Several analyses were performed on a sample of 299 software developments from 17 organisations.

SOME PRIOR RESEARCH ON FP METRICS

A function point (FP) count of a system measures two components. These are firstly the information processing size, expressed in unadjusted UFP's, and secondly the technical complexity measure, expressed as the GSC's.

The information processing size is determined by categorising a system into 5 function types. These are external inputs, outputs, inquiries, internal logical files and external interface files. Each of these is further classified as either low, average, or high functional complexity, depending on the number of data element types, and other factors.

Each classification for each function type is allocated a number of points or weights, and the sum for all components is expressed as the number of unadjusted function points. Table 1 shows the matrix of the points allocation for each classification.

Table 1 Function Point Allocation

Description	Low	Average	High
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6
External Interface File	5	7	10
Internal File	7	10	15

As an example, a system with 6 simple inputs, 4 average outputs, 3 complex outputs, 2 average internal files and a complex external interface file would have a total UFP count of:

6x3 + 4x5 + 3x7 + 2x10 + 1x10 = 89 UFP

The treatment of complexity is somewhat subjective, but its determination, using the 14 GSC's is supported by guidelines for interpretation (IFPUG CPM, 1994). The value adjustment factor (VAF) is derived by the following equation:

VAF = (TDI x 0.01) + 0.65

where

TDI is the total degree of influence as determined by the GSC's.

The degree of influence is determined by 14 complexity factors. Each of these is evaluated on a scale of 1 to 5, depending on the degree to which the factor is present in the system. A 0 would be allocated if the factor was absent, while a 5 would be allocated if the factor exerted a strong influence throughout the system. The other values fall between these two extremes (decimal values are permitted), depending on their degree of influence. The sum of all 14 factors determined in this way is termed the Total Degree of Influence (TDI).

Symons (1991) in his effort to overcome some difficulties associated with FPA has proposed his MkII Function Point model. The information processing size is expressed in unadjusted function points, but is now the sum of the weighted number of input data element types, the weighted number of entity-type references, and the weighted number of output data element types. Symons scaled the MkII weightings so that for the 8 systems he examined, the average size of unadjusted function points were in the same range as Albrecht's FPA model (Albrecht, 1979).

In addition Symons expanded the number of factors affecting the technical complexity from 14 to 20. Symons also questioned the weight of each degree of influence, and has suggested that this should vary with technology.

Research has not proved conclusively that Symons' MkII model is a better size metric enabling more accurate estimates to be made than when Albrecht's FPA model is used. Results of research done by Ratcliffe and Rollo (1990) are inconclusive. To improve on the original function point counting technique various other adaptations have also been suggested (Jones, 1991; Reifer, 1990). In a research study MacDonell (1994) identified and evaluated nine function-based assessment and estimation methods.

DATA USED

This database is the result of a compilation of 299 projects from 17 different organisations (Desharnais et al, 1990). The standard deviation and the skewness of the data suggests the possible presence of outliers, but none of these were excluded from the analysis.

Table 2 Summary of Project Data

	Description	Mean	Min.	Max.	Skew.	# cases
Effort	Effort in hours	7086	247	86478	4.75	299
UFP	Unadjusted FP	298	48	1257	1.81	299
FP	Adjusted FP	267	40	1182	1.86	299
Duration	Duration in months	14	1	67	2.17	201
UFP/Hour	Productivity	0.071	0.008	0.696	3.99	299

The total development effort in hours is somewhat skewed towards smaller projects but still covers a range typical of commercial developments. The productivity variation from 0.008 UFP/Hr. to 0.696 UFP/Hr. (87 times) is also typical of the type of variation possible in systems development and is why software effort estimation remains such a difficult problem.

ANALYSIS

Several different types of analysis were performed to determine how effective the use of GSC's and the weighting scheme was. Regression analysis was used to assess the effectiveness of FPA for predicting development effort i.e. how accurate different regression models were in estimating total development time in effort hours.

ANOVA and t-tests were used to assess whether the GSC's were performing effective adjustments to cater for differences in productivity between projects i.e. were there any significant differences in productivity between system developments with high or low ratings in specific GSC's. Logically it could be expected that projects requiring extensive use of e.g. data communications might have lower productivity than projects without this complexity. Factor analysis was also performed to determine the covariance of the GSC's i.e. how separable are the general system characteristics as defined by FPA. T-tests were used to determine whether there were any significant differences in productivity between projects with different proportions of inputs, outputs, inquiries, internal files and external files. If the FPA model is valid it should make no difference in terms of productivity if a project has a relatively high proportion of e.g. inputs as opposed to one with a low number of inputs.

REGRESSION ANALYSIS

The sample of 299 was divided randomly into 249 training cases and 50 test cases. Regression was performed on the 249 training cases to extract models to estimate development effort based on unadjusted function points, function points (adjusted), log-linear transformations of these and development effort as well as two multiple regression models based on UFP with the 14 GSC's as input as well as log-linear UFP with the 14 GSC's. (Since the GSC's are used to compute FP's from UFP's there is no need to perform multiple regression with FP and the GSC's). Log-linear models were investigated to assess whether these compensated for the effect of system size as productivity is generally known to decrease with increasing system size.

These models were then used to estimate development effort in the other 50 cases. The quality of the estimation is based on two measures, the Mean Absolute Relative Error (MARE) and the proportion of the estimate within 25% and within 50% of the actual development effort. The results are summarised in Table 3 and discussed in the Results section below.

Table 3 Regression Analysis Results

	MARE	≤25%	>25% ≤50%	> 50%
UFP	1.002	34%	28%	38%
FP	0.903	30%	30%	40%
UFP Log-Linear	0.790	40%	22%	38%
FP Log-Linear	0.733	40%	28%	32%
UFP + GSC's	1.234	24%	26%	50%
Log UFP+GSC's	0.623	36%	36%	28%

The predictive accuracy of FPA with regression models clearly shows considerable error and even in the best case only provides 40% (20 out of 50 cases) of the test cases with an effort prediction within 25% of the actual. The best model overall (with an MARE 0f 0.623) is a log-linear multiple regression model using unadjusted function points and the 14 GSC's. Other methods such as the use of Artificial Neural Nets (ANNs) and Case Based Reasoning (CBR) have been investigated by the authors (Finnie & Wittig, 1996), and could provide more accurate models. However the underlying questions concerning the validity of the construction of FP's remain a serious factor (Abran & Robillard, 1994).

USING GENERAL SYSTEM CHARACTERISTICS

GSC's are intended to adjust the function point count to compensate for differences in project complexity e.g. a project which makes extensive use of data communications could be expected to require more development effort for the same function point count as a project which requires little data communication capability. They have been criticized on the grounds that they do not adequately cover the spectrum of issues which influence complexity and because the effect of any single complexity factor is no more than 5% of the total estimated effort.

Since GSC's adjust for complexity there should be no difference in productivity (measured in FP's per hour) between those projects which rank high on a particular complexity factor as opposed to those which have a low score. Two analyses were performed. Firstly the full sample of 299 cases was partitioned for each GSC into those with a low score on the factor (between 0 and 2) and those with a high score (3-5). T-tests were used to determine whether there was any significant differences in productivity (as measured in both UFP/hr and FP/hr) for each factor. The results are given in Table 4 and are discussed in more detail in the Results section. However it would appear that in general the GSC weights are insufficient to compensate for differences in productivity which might be due to the specific GSC. For example, in the "Multiple Sites" GSC the differences in productivity between the UFP and FP cases remains significant i.e. use of the complexity GSC did not provide sufficient weight to override the effect of complexity. On the other hand several GSC's did not appear to be a significant factor in productivity (e.g. heavy use) and their value in adjusting the function point count is questionable.

Table 4 Differences in Productivity by GSC (t-test)

GSC	UFP t-value	UFP probability	FP t-value	FP probability
Data Communications	-0.99	0.16	-2.91	0.002
Distributed System	1.33	0.09	0.69	0.25
Performance	3.59	.0002	1.78	0.04
Heavy Use	1.00	.159	-0.68	0.25
Transaction Rate	1.62	0.05	-0.29	0.39
On-Line Data Entry	-2.165	0.02	-4.04	0.00004
End-User Efficiency	1.88	0.03	-0.17	0.43
On-Line Update	1.00	0.16	-1.10	0.14
Complexity	3.50	0.0003	2.36	0.009
Reuse	1.17	0.12	0.002	0.50
Installation Ease	1.10	0.14	0.02	0.49
Operational Ease	1.37	0.09	0.05	0.48
Multiple Sites	4.42	0.00001	2.97	0.002
Facilitate Change	4.27	0.00001	2.56	0.005

Table 5 Differences in Productivity by GSC (ANOVA)

GSC	UFP F-value	UFP probability	FP F-value	FP probability
Data Communications	0.54	0.75	1.14	0.34
Distributed System	0.72	0.61	0.70	0.63
Performance	2.51	0.03	0.60	0.70
Heavy Use	1.93	0.09	1.27	0.28
Transaction Rate	1.65	0.15	0.86	0.51
On-Line Data Entry	1.07	0.38	2.34	0.05
End-User Efficiency	2.52	0.03	0.66	0.65
On-Line Update	1.05	0.39	0.23	0.95
Complexity	6.57	0.0001	4.35	0.001
Reuse	0.67	0.65	0.56	0.73
Installation Ease	2.85	0.002	1.25	0.28
Operational Ease	3.61	0.003	1.12	0.35
Multiple Sites	2.27	0.05	1.89	0.09
Facilitate Change	4.55	0.001	2.73	0.02

In addition ANOVA was used to determine whether any significant differences existed between the groups for each GSC. The sample was partitioned into 6 groups for each GSC based on the score for the specific factor i.e. from 0 to 5. The analysis was performed by controlling for the effect of differences in the number of.inputs, outputs, inquiries, internal files and external interface files. This analysis was not as useful as the first as the variable sample sizes and within group variance tended to reduce the prospects of determining any significant differences. The results are given in Table 5 and discussed in the results section. These results again indicate that the value of GSC's in adjusting for productivity differences is low.

FACTOR ANALYSIS OF GSC'S

The 14 GSC's have been criticised for ambiguity and incompleteness. Factor analysis was used to assess how strongly the various factors belonged together i.e. could be interpreted as being part of one component or factor. SPSS factor analysis with varimax rotation was used to extract solutions for two, three and four factors. Of these the two factor solution appears to be the best as the others tend to share a number of variables. The two factor weights are given in Table 6. For this data set it appears that there is considerable covariance and that a number of the factors are difficult to separate.

Table 6 : Factor Analysis of GSC's

Factors	Factor 1	Factor 2
Data Communications	0.11	0.84
Distributed System	0.13	0.31
Performance	0.77	0.19
Heavy Use	0.53	0.40
Transaction Rate	0.71	0.33
On-Line Data Entry	0.11	0.81
End-User Efficiency	0.57	0.57
On-Line Update	0.37	0.73
Complexity	0.70	-0.29
Reuse	0.26	0.26
Installation Ease	0.60	0.24
Operational Ease	0.59	0.06
Multiple Sites	0.01	0.52
Facilitate Change	0.54	0.27
Eigenvalue	4.95	1.67
Percent of variance	35.3%	12.0%

ANALYSIS OF FACTOR WEIGHTS

The unadjusted function point total is computed from the weighted score of simple, average and complex inputs, outputs, inquiries, internal and external files. The scoring scheme is given in Table 1 and was determined by Albrecht on the basis of "trial and debate".

If the weighted scoring scheme adequately compensates for the differences in functionality (and hence development effort) between inputs, outputs, etc. and between simple, average and complex variants of these, there should be no difference in productivity (in UFP/hr) between projects with differing proportions of each component. For example, a project with a high ratio of input functionality to the total unadjusted function point score should (subject to GSC differences) have no difference in productivity to one with low input ratio and a high external interface file ratio. The ratio of the weighted input score to the total UFP count was computed. The median of this ratio was determined for the sample and the total partitioned on the median.

A simple t-test was performed to see if any significant differences in productivity were present. This was repeated for the outputs, inquiries, internal files and external interface files, as well as for the total of input, outputs and inquiries. In addition the ratio of inputs, outputs and inquiries to external and internal files was also computed. The results are given in Table 7 and clearly indicate that the relative proportion of different components in a system development is a very significant factor in productivity. For example, systems with a high proportion of inquiries in the system had far lower productivity than those with a low proportion of inquiries. It is apparent that the scoring system in no way compensates for this effect and that the weighting scheme used in Function Point Analysis is in need of some reassessment.

Table 7 Analysis of Factor Weight Results

Ratios	t- statistic	Probability
IOI/Files	-3.18	0.0008
Input/Total	-2.33	0.01
Output/Total	1.51	0.06
Inquiries/Total	-4.93	8.8*E-07
Internal Files/Total	1.72	0.04
External Files/Total	2.17	0.02
IOI/Total	-3.50	0.0003

RESULTS

In general the results support the concerns about the value of GSC's and the weights used in computing unadjusted FP's. Using multiple regression with FP's (Table 3) was not particularly successful in accurately predicting development effort. The results are similar to those obtained by other researchers (Jeffery, 1987).

Study of Table 4 suggests that the GSC's adjustments are not adequate to compensate for the effect of each factor on productivity. In the unadjusted function point case only seven of the GSC groups show significant differences in productivity. For the other seven it is possible that within group variance has exceeded between groups variance so that the t-test is inconclusive or it may indicate that there is in fact no significant difference in productivity for the other groups. In five of the seven cases the use of the adjusting factors (as indicated by the significant differences for the adjusted function point column) did not negate the effect of productivity differences sufficiently to remove any significant differences between the groups. In two cases the use of the technical complexity adjustment in fact worsened the situation i.e. for on-line data entry and for data communications.

Table 5 is not very conclusive as the within group variance (for 6 groups) has probably exceeded the between group variance and makes it difficult to sensibly interpret the results. Only 6 factors show significant differences in productivity for unadjusted FP's. Of these, three remain significantly different while on-line data entry becomes a significant difference. The results suggest again that the GSC's reduce the productivity differences but do not eliminate them.

The factor analysis show one strong factor which probably reflects general complexity (factors such as Performance, Heavy Use, Transaction Rates, End-User Efficiency, Complexity, Installation Ease, Operational Ease and Facilitate Change) and a second factor which appears to relate to on-line distributed system use (factors such as Data Communications, Heavy Use, On-line data Entry, End-User Efficiency, On-Line Update and Multiple Sites). The fairly strong relationship between some of the components of each factor again suggests that the GSC's are not easy to separate in practice.

Table 7 shows very significant differences in productivity between groups with high and low ratios of a specific UFP component e.g. the difference in productivity between projects with a low ratio of inputs to the total unadjusted function point score and those with a high ratio is significant at the $P < 0.01$ level. With the exception of outputs all other components show significant differences (Outputs has $t = 1.51$ and $p=0.06$). This strongly suggests that the weighting scheme applied, at least in this sample, is not compensating for the differences in productivity between developing, for example, an average input and a complex inquiry.

CONCLUSIONS

For the large data set analysed the results support the view that the use of FP's to estimate project size and hence development effort has a number of problems. The VAF appears to be inadequate and different methods of adjusting an estimate to account for complexity need to be devised, particularly in the light of new development methodologies and environments. The fundamental inputs to the FP count i.e. the inputs, outputs, etc. do not reflect the differences in development effort for different types and levels of components.

Abran and Robillard (1994) found that FP's do not derive from a well-defined and proven theory, and they are entirely empirically based on expert opinion. They identified the existence of implicit transformations and implicit models without which the measurement process would be invalid.

The results from the study in this paper would appear to indicate that both the type of component and the weighting scheme used should be reassessed, in addition to clarifying some of the implicit FP mappings.

REFERENCES

Abran, A., & Robillard, P.N. (1994) **Function Points: A Study of Their Measurement Processes and Scale Transformations**, Journal of Systems and Software, Vol. 25, pp. 171–184.

Albrecht, A.J. (1979) **Measuring Development Productivity**, Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, pp. 83–92.

Albrecht, A.J., & Gaffney, J.E. (1983) **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.

Dam, J.V.Y. & Langbroek P.L. (1992) **Gebruik van Functiepuntanalyse Vraagt on Beleid**, Informatie, Vol. 34, No. 6, pp. 323–333.

Desharnais, J.M. (1988) **Analyse Statistique de la Productivité des Projets Informatiques à Partir de la Technique des Point de Fonction**, Master Thesis, Université du Québec à Montréal.

Desharnais, J.M., et al. (1990) **Adjustment Model for Function Points Scope Factors - A Statistical Study**, IFPUG Spring Conference, Florida.

Dreger, J.B. (1989)**Function Point Analysis**, Prentice Hall, Englewood Cliffs.

Ferens, D.V., & Gurner, R.B. (1992) **An Evaluation of Three Function Point Models for Estimation of Software Effort**, IEEE National Aerospace and Electronics Conference – NAECON92, Vol. 2, pp. 625–642.

Finnie, G.R., & Wittig, G.E. (1996) **AI Tools for Software Development Effort Estimation**, Proceedings of the Conference on Software Engineering : Education and Practice, University of Otago.

Function Point Counting Practices Manual, Release 4.0 (1994) International Function Point Users Group, Blendonview Office Park, 5008–28 Pine Creek Drive, Westerville, OH 43081–4899, USA.

Jeffery, D.R. (1987) **Time Sensitive Cost Models in the Commercial MIS Environment**, IEEE Transactions on Software Engineering, Vol. 13, No. 7, pp. 852–859.

Jones, C. (1991) **Applied Software Measurement**, McGraw-Hill, New York.

Kemerer, C.F. (1987) **An Empirical Validation of Software Cost Estimation Models**, Communications of the ACM, Vol. 30, No 5, pp. 416–429.

MacDonell, S.G. (1994) **A Comparative review of Functional Complexity Assessment Methods for Effort Estimation**, Discussion Paper Series, University of Otago, No. 94/8, ISSN 1172-6024.

Ratcliff, B., & Rollo, A.L. (1990) **Adapting Function Point Analysis to Jackson System Development** Software Engineering Journal, pp. 79–84.

Reifer, D.J. (1990) Asset-R: A **Function Point Sizing Tool for Scientific and Real-Time Systems**, Journal of Systems Software, Vol. 11, pp. 159–171.

Symons, C.R. (1991) **Software Sizing and Estimating MkII FPA**, John Wiley & Sons, Chichester.