Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Systems and Software 74 (2005) 195–206
The Journal of Systems and Software
www.elsevier.com/locate/jss

Modification of standard Function Point complexity weights system

Mohammed Abdullah Al-Hajri *, Abdul Azim Abdul Ghani, Md Nasir Sulaiman, Mohd Hasan Selamat

Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Received 27 May 2003; received in revised form 9 December 2003; accepted 20 December 2003
Available online 28 February 2004

Abstract

Function Point (FP) is a software size measure, which includes the standard FP and many different models derived from it. The standard FP method created by Albrecht in 1979 is currently known as the International FP User group (IFPUG) version, which consists of three main parts: The first part is five components, and the second is the complexity weights that include three levels of complexity; simple, average, and complex. The third part is the general system characteristics of software projects, which consists of 14 technical complexity factors. Although, FP was widely used as a software size measure, but it still suffers from many weaknesses. One of which is the subjectivity in the weights system. In this paper a new FP weights system was established using Artificial Neural Networks. This method is a modification of the complexity weights of FP measure (IFPUG version). The final results were very accurate and much suitable when they were applied on real data sets of software projects.

Keywords: Function Point weights; Artificial Neural Networks; Software size measure; Effort estimation

1. Introduction

Albrecht's original Function Point (FP) method has evolved over the last 20 years into a method now known as IFPUG 4.1, though the original basic concepts and weighting values have not changed since 1984. This method consists of three main parts: The first part consists of identifying five components, which are External Inputs (EI), External Outputs (EO), External Inquiry (EQ), Internal Logical File (ILF), and Internal File Interfaces (IFI). The second part consists of assigning complexity weights to each component using the levels of simple, average, and complex. The third part consists of evaluating the general system characteristics (GSC) of any software, which consists of 14 technical complexity factors that give a clear picture of internal complexity of any software. Albrecht in 1979 created the original model of measuring the FP, this model is represented by Eqs. (1) and (2):

$$\mathrm{FP} = \mathrm{UFP} \times \mathrm{AFP}, \tag{1}$$

$$\mathrm{AFP} = 0.65 + 0.01 \times \sum \mathrm{GSC}. \tag{2}$$

UFP is an unadjusted FP. This value can be calculated according to the weights tables and data elements by using some rules (Longstreet, 2001). AFP represents the Adjusted FP, which can be calculated by 14 factors of GSC. Each factor takes any value in the ordinal scale of 0–5 as a degree of influence. The two constants 0.65 and 0.01 are essential parts of the formula (Pressman, 2001). The FP can be calculated and analyzed in different stages of software development in order to provide a mechanism to track control scope creep of software projects (Longstreet, 2001; Rule, 2001).

2. Literature review

Allan Albrecht created FP in 1979. Since that time it suffers from essential problems, such as problem with accuracy, early life cycle use, changing requirements, measurement theory, subjective weighting and etc. (Fenton and Pfleeger, 1997; Symons, 2001). Hence it

* Corresponding author. Tel.: +60-192-023489; fax: +60-389-466572.
E-mail address: moh_hajri2003@yahoo.com (M.A. Al-Hajri).

0164-1212/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2003.12.033

passes through many stages of improvement. A number of interested researchers tried to improve the FP in order to solve some of these problems. Many releases of FP measure were created such as Feature Points, 3D Function Point, Mark II Function Point, Full FP and COSMIC FFP (Symons, 2001).

2.1. Feature Points

It is a software sizing method developed by Capers Jones. This technique considers the number of algorithms used in the application and slightly modifies some of the weights of the FP constituents. It was designed in such a way to suit typical business applications, an area where FP has been applied successfully, and would have the same size whether calculated with FP or Feature Points. (SCT, 1997). The Feature Point measure accommodates applications in which algorithmic complexity is high such as Real-time, process control, and embedded software applications (Pressman, 2001). Furthermore, Feature Points do not enjoy the same level of acceptance as FP does. There is no standards organization, like IFPUG, that is concerned with its use (SCT, 1997).

2.2. 3D Function Point

Between 1989 and 1992, the Boeing Company considered the use of FP to measure productivity. An internal Boeing company document was published in the proceedings of the Pacific Northwest software quality conference in 1992. It defined a technique called 3D FP (SCT, 1997). This technique was designed to address two classic problems associated with the Albrecht approach: First, the approach is considered difficult to use. Second, it does not properly measure scientific and real-time systems (SCT, 1997). The three dimensions in 3D FP include data dimension, function dimension and control dimension. The data dimension is similar to Albrecht's FP. The function dimension adds transformations, which are similar to the algorithms. The control dimension adds transitions, which enumerate changes in application state it is a fact that 3D FP are not widely used (Pressman, 2001; SCT, 1997).

2.3. Mark II Function Point

This technique views software functionality in terms of logical transactions (input data, process, and output data) and generates counts by multiplying the logical transaction to pre-defined weights, using a published industry average weights or tailored to suit the client environment (Symons, 1991; UKSMA, 1998). In addition, there are five new additional degrees of influence to the 14 adjustment factors of GSC. These additional characteristics are interfaces, security and privacy, user training, third party use and documentation. Clients could also define additional characteristics if necessary (UKSMA, 1998). However, the manual current sizing rules do not take into account size contributions from complex algorithms, which are found in scientific and engineering software nor do the rules specifically take into account real-time requirements. To apply Mark II FP analysis to these other domains may be possible or may require extensions or new interpretations of the rules given in its manual (UKSMA, 1998).

2.4. Full Function Point

In 1997, the Software Engineering Management Research Laboratory (SEMRL) in Université du Québec à Montréal, Canada made an attempt, and the Software Engineering Laboratory in Applied Metrics (SELAM), to extend the capabilities of FPA to measure real-time software. This technique is called Full Function Point (FFP). It encompasses six function types: read only control group, update control group, external control entry, external control exit, internal control read, internal control write. These six types are discussed in Bootsma (2000). This method was developed into another approach called COSMIC Full Function Point.

2.5. COSMIC Full Function Point

Common software measurement international consortium (COSMIC) is a group established by six countries, Australia, Canada, Finland, the Netherlands, UK and the USA, with the aim to achieve an international standard set of software measurement. The COSMIC method was a refinement of FFP, Mark II and the FPA techniques, in order to address a variety of software domains especially MIS and real-time systems. The COSMIC group has released the COSMIC measurement manual, which contains all the rules and the different components of this method (COSMIC, 1999). This method explicitly does not claim to measure the size of functionality, which includes complex data manipulation (i.e. algorithms), and does not attempt to take into account the effect on size of technical or quality requirements (Symons, 2001).

2.6. IFPUG version

Albrecht's original FP Analysis method has evolved over the last 20 years into a method now known, as "IFPUG 4.1", the original basic concepts and weighting methods have not changed since 1984 (Symons, 2001). The IFPUG version is a modified version of Albrecht's FP. In the modification, the evaluation of the complexity of the software was objectively established and the rules of the counting procedures were also described minutely and precisely. In this version, the counting procedure of

FP consists of seven steps according to the FP manual (Longstreet, 2001; Kusumoto et al., 2002).

The International Organization for Standards (ISO) announced in early 2001 that recognition of FP as an international standard had taken a major step forward. By a large majority, the national bodies comprising ISO approved the application for recognition filed by the IFPUG. Following resolution of the comments accompanying the votes of approval, FP will become the first software functional sizing methodology to be recognized as an international standard (IFPUG, 2002). This version still suffers from some essential problems such as the subjectivity in the complexity weights.

3. Subjective weighting in the IFPUG version

One of the well-known problems of FP is its reliance on subjective weights. Examples of opinions on this problem include the following:

(1) The classification of function types into simple, average, and complex only simplifies the adjustments. This classification does not reflect the entire complexity of the work, which is necessary to develop the user's systems (Symons, 1991; SCT, 1997). Eleven problems associated with the FP weights, complexity factors and other requirements were described in (Fenton and Pfleeger, 1997).
(2) FP suffers from weight weakness, where Symons (2001) found that the choices of weights for calculating UFP were determined subjectively from IBM experience.

Multiple versions of FP have been developed from 1988 to 2000 (Symons, 2001), but no improvements to the weighting scheme have been made. Nevertheless, new publications still used the previous complexity weights of FP in Pressman (2001) and Hastings (2001), which are represented as an ordinal scale.

Creating a new weights system as an absolute scale instead of the ordinal scale can provide the solution of subjectivity weighting. The ordinal scale is lower order than the absolute scale, to allow flexibility, software measures should aim at high order scale such as ratio and absolute scale (Hastings, 2001).

4. Research objectives

The main objective of this research was to establish a new weights system for the FP measure. The objective will be achieved through the following steps:

- Using the original weights system as base lines to establish the new weights.
- Training one of the most popular techniques in the Neural Networks to predict the new weights values. This technique will use the original weights system as inputs. The outputs of this network will be the new complexity weights system.
- Applying the new weights and the original weights in the FP model (IFPUG version).
- Satisfying the validation of this research by calculating the FP count depending on the original and the new weights. This count will be used as a parameter to compute projects required effort (total working hours). The generated values of projects effort will be compared with the actual values according to the data sets of real projects.

5. Research methodology

Accordingly this research was planned to improve the weights system of FP measure to reduce the error rate of prediction degree and to increase the calculation accuracy of this measure.

5.1. General steps of improving FP weights

This research has used the original weights of Albrecht as a starting point to create the solution.

Fig. 1 presents the general steps of how to calculate the FP using the original weights system, which is currently used in the IFPUG as five rule tables. The general layout of the research methodology can be shown in Fig. 2. It includes many substages such as mathematical calculations, special algorithms and creating training Data Base (DB) and network learning.

The Artificial Neural Network method (back-propagation) algorithm is applied in this methodology as a prediction tool (Freeman and Skapura, 1991).

Few samples of training patterns can be calculated directly from the original weights tables that are used as a baseline for generating full training DB to learn the back-propagation network. The proposed method can be explained as follows:

Step (1): closing the open intervals of ILF rule table

Final intervals of data elements (DETs) and record elements (RETs) in ILF rule table are opened

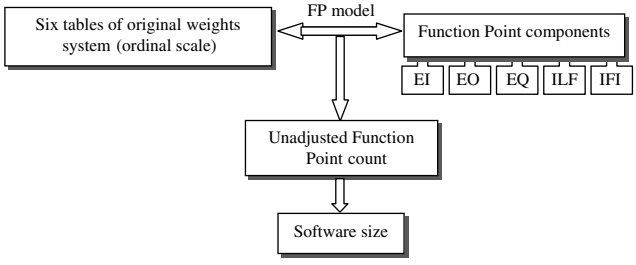

Fig. 1. Calculation of the Function Point using the original weights.

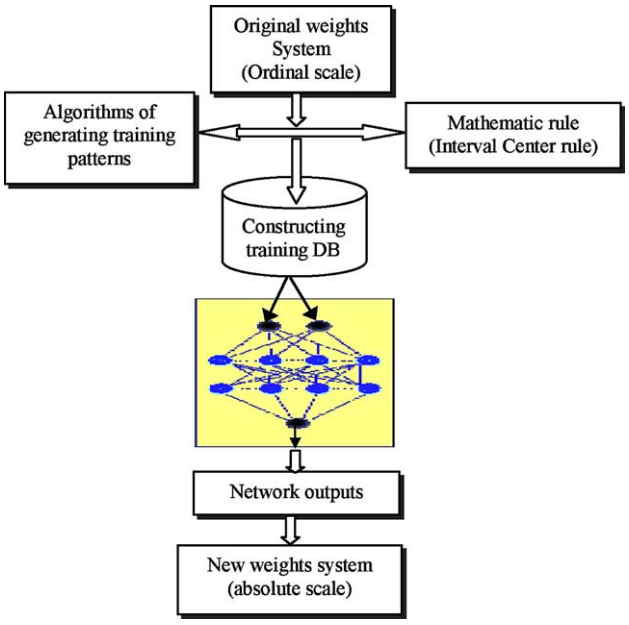

Fig. 2. Creating new weights system from the original.

(Longstreet, 2001), so all these intervals must be closed before starting the solution. The final open intervals of ILF are {>5 RET}, {>50 DET} as shown in Fig 3. After closing these intervals it becomes {6–10 RET}, {51–93 DET} where the values 10 and 93 are determined according to the closed intervals, which are {1}, {2–5} for RET and {1–19} and {20–50} for DET as shown in Fig. 3.

Step (2): selecting weights samples of ILF

Albrecht tabulated weights of ILF as an ordinal scale, simple, average, and complex {7, 10, 15} for DET groups {1–19, 20–50, >50} and RET groups {1, 2–6, >6}, respectively, see Fig. 3. The values of these groups will be used to calculate the Interval Center (IC) of DETs and RETs in each class of weights. For getting more details about the original weights system refer to the IFPUG manual (Longstreet, 2001).

The IC of DETs in the three levels simple, average, and complex were calculated for all values that placed on the main and secondary diameter of the last table of Fig. 3. The calculations were performed as follows:

- The first level {1–19}
  IC of DETs (simple) = (*left limit of DET + right limit of DET*)/2.
  IC of DETs = ((1 + 19)/2) = 10 DETs.
  IC of RETs (simple) = ((1 + 1)/2) = 1.
  1 RETs and 10 DETs will take only 7 weights.
  Also IC of RETs (simple) = ((6 + 10)/2) = 8.
  8 RETs and 10 DETs will take only 10 weights.
- The second level is {20–50}
  IC of DETs (average) = ((20 + 50)/2) = 35.
  IC of RETs (average) = ((2 + 5)/2) = 3.5.
  3.5 RETs and 35 DETs will take 10 weights.
- The third level is {>50–93}
  IC of DETs (complex) = ((51 + 93)/2) = 72.
  IC of RETs (complex) = ((6 + 10)/2) = 8.
  8 RETs and 72 DETs will take 15 weights.
  Also IC of RETs (complex) = ((1 + 1)/2) = 1.
  1 RETs and 72 DETs will take only 10 weights.

The calculated values are called the main training patterns that are illustrated in Fig. 4. These patterns will be used for generating more patterns to learn and test the suggested network.

These patterns were used as inputs to the algorithm presented in Fig. 5. The objective of using this algorithm is to increase the training patterns because these patterns are not enough. The minimum number of these patterns that can give accepted results is 20. This number was concluded according to the learning results of the suggested network. Training is improved when the training patterns are increased. As a result of applying the algorithms in Figs. 5 and 6 on the ILF component, Table 1 was constructed. This table contains 38 patterns (triple values), these values were used as inputs, outputs during the learning time. All these values must be normalized at first. Each triple pattern (RET, DET and weight) of Table 1 used as a single input (DET, RET)

FP components	simple	average	complex
EI	3	4	6
EO	4	5	7
EQ	3	4	6
EFI	5	7	10
ILF	7	10	15

	Data Element		
RET	1-19	20-50	>50
1	7	7	10
2-5	7	10	15
>5	10	15	15

	Data Element		
RET	1-19	20-50	51-93
1	7	7	10
2-5	7	10	15
6-10	10	15	15

Diameter

Fig. 3. The rule table of the internal logical file weights.

Record Element	Data Element		
	1-19	20-50	51-93
1	7	*ig*	10
2-5	*ig*	10	*ig*
6-10	10	*ig*	15

S	RET	DET	Weight
1	1	10	7
2	3.5	35	10
3	8	72	15
4	8	10	10
5	1	72	10

Right limit
Left limit
Interval_Center = (L + R)/2

Interval_Center(RETs)=(1+1) /2=1
Interval_Center(DETs)=(1+19) /2=10
1 RET ® 10 DET=7
Interval_Center(RETs)=(6+10) /2=8
Interval_Center(DETs)=(51+93) /2=72
8 RET ® 72 DET=15

5 Main patterns, ***Not enough for NN training***

Fig. 4. Calculation of five main patterns.

```
Build_pattern_Table (integer FP_Component_k, N; Table_patterns  T)
 Integer i, L, j;
Begin
      j <----- 0;
     For each class: simple, average complex of FP component_k  Do
    Begin
         j <----- j + 1;
         Calculate MID point value of DETs, REF  {REF mens RETs or FTR}
        T[j].DETs <----- Mid_DETs ;
        T[j].FTR <----- MID FTR ;
        T[j].weight  <----- class_weight;
      End
       Main_ptrns_no = j;
      i <-----1;
     Repeat
          For L <----- I + 1 to Main_ptrns  Do
          Begin
               T[j + 1].DETs <----- (T[i].DET + T[L].DET) / 2;
               T[j + 1]. REF  <----- (T[i].REF + T[L]. REF) / 2;
               T[j + 1].weight  <----- (T[i].weight + T[L].weight) / 2 ;
               J <----- j + 1;
               If j > N then Exit ();
         End;
         i <-----  I + 1;
    Until j > = N;
End.
```

Fig. 5. Algorithm of generating more training patterns.

```
 FP_components_Tables( )
Integer Component_Number, Max_Patterns_Number, f; String Table_Name;
 Begin
       For f <----- 1 to FP component number Do
      Begin
          Read (Component_Number, Max_Patterns_Number, Table_Name);
         Call Build_pattern_Table (Component_Number, Max_Patterns_Number,
         Table_Name);
     End
End;
```

Fig. 6. Algorithm of generating training patterns for all components.

Table 1
A data set of training patterns

Learning patterns			Testing patterns		
RET	DET	Weight	RET	DET	Weight
1	6.3	10	9.3	10	7
3.5	1.63	46.8	9.3	35	10
8	2.8	55.8	10.5	72	15
8	0.5	35.5	5	10	10
1	1	55.8	9.25	72	10
2.3	3.38	31.5	9.75	22.5	8.5
4.5	3.38	16.3	8.5	40.5	11
4.5	1.63	31.5	8.5	10	8.5
1	4	37.8	10.5	40.5	8.5
5.75	4.5	25.3	9.75	53	12.5
5.75	2.75	40.5	9.75	22.5	10
2.25	5.13	46.8	11.8	53	10
2.9	5.13	31.5	10.5	28.75	9.25
8	2.75	25.3	8.5	40.5	12.5
4.5	5.13	31.5	10.5	71	12.5
5.13	5.13	16.3	9.25	46.75	1.75
6.3	3.38	31.5	9.25	55.75	13
4.5	3.38	46.8	10.5	40.5	10
5.13	3.38	31.5	9.25	16.25	9.25

and output (Weight) for the suggested network during the learning stage. Each couple patterns (RET, DET) of testing patterns in the same table are used for network testing.

Step (3): normalization of training patterns

The objective of the normalization is to get values inside the range $\{-1 \rightarrow +1\}$ or $\{0 \rightarrow +1\}$, this operation will make the network learning very quick, easy and to save space and time during the training (Shamsuddin et al., 2001).

The learning patterns can be normalized, for example by using Eq. (3), which was used in normalizing the training data of back-propagation algorithm (Shamsuddin, 2000; Shamsuddin et al., 2001), which was used in pattern recognition.

$$Y = \mathrm{Rm} + (\mathrm{Rx} - \mathrm{Rm}) \times (\mathrm{Ov} - \mathrm{Dm})/(\mathrm{Dx} - \mathrm{Dm}), \quad (3)$$

where $\mathrm{Rm} = 0$, $\mathrm{Rx} = 1$, Dx and Dm = the maximum, the minimum values in the learning-patterns and Ov is the original value.

Step (4): the architecture of the network used

The architecture of the network used consists of two inputs, many hidden nodes (neurons), and one output (FP complexity weights). This network was created five times for the five components of FP. Table 2 contains the special parameters of all networks that are used for creating the new weights system after a large number of training iterations of these networks. Each FP component is associated with its special network because it has its own training data, which differ from one component to another. On the other hand, all used networks have the same number of inputs and outputs but each one has its special training parameters.

Step (5): the outputs of the learned network

The steps listed above from 1 to 4 were repeated for all other components of FP. When the network has become ready (learned network), the new complexity weights of any pair (RET, DET) could be produced directly as an output of this network.

Fig. 7 shows the general layout of calculating the FP using the new weights system.

6. Results and discussion

The results of this research were divided into three groups: results of initial, intermediate, before final and final.

6.1. The initial and intermediate results

The results of initial stage are classified into five groups and each group is associated with a single FP component as it was shown in Fig. 7. These results were

Table 2
Special parameters of five networks

I	FP component	Number of inputs	Number of outputs	Hidden nodes	Learning rate (η)	Learning iterations	Minimum error
1	EI	2	1	10	0.19	11,214	0.0003
2	EO	2	1	85	0.19	259	0.0002
3	EQ	2	1	10	0.35	5477	0.00018
4	ILF	2	1	20	0.28	32	0.0003
5	IFI	2	1	150	0.19	10,810	0.00019

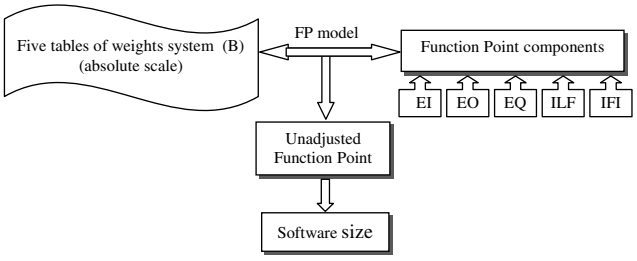

Fig. 7. Calculation of the Function Point using new complexity weights.

used as inputs to the algorithms in Fig. 6. The outputs of this algorithm represent the results of the intermediate stage. The intermediate results are also five groups for each FP component. It constructed the training DB. Table 1 contains the training DB of ILF component. This table is the inputs of the assigned network. The network outputs represent the new complexity weights of this component.

6.2. The results before final

The network outputs are the complexity weights. This outputs arranged in five tables. It represents the before final results that will be used directly for calculating the FP count instead of the original weights. These five tables are available for all interesting in FP by requesting it directly from the author via the email address. Table of Appendix A presents a comparison between the original and the new weights of the ILF component. This table shows the flexibility of assigning the new weights as opposite to the original method.

6.3. Final results

Final results are the important part in this research that presents the acceptance of the suggested methodology. These results were calculated according to real data sets of software project.

6.3.1. ISBSG data projects

The International Software Benchmarking Standards Group (ISBSG) has collected more than 1200 real software projects. These data collected for assisting researchers. The principal purposes for compiling the project database are to provide IT practitioners with industry output standards against which they may compare their aggregated or individual projects, and to provide accurate, current and comprehensive data about software development in many countries so that the relative delivery rates and the factors impacting upon delivery may be analyzed. All data is stored in the strictest confidentiality. There were two main reasons for using this data:

1. The calculated data represent real projects from several countries around the world.
2. The accumulated historical project data includes development effort using the FP and all their data set were arranged for the purpose of researches in order to compare the results as an apple to apple. Furthermore, in all cases the total developmental effort (actual effort) was given in working hours. More than 205 projects of the collected data were used in this research. The selected projects contain full data set for each FP component (55 description fields are associated with each project), for this reason this sample of projects was selected. The main fields of this data are shown in Appendix B.

6.3.2. Final results of a large sample effort estimation

This section presents the practical results of calculating the FP value using the modified weights and the original weights. These two counts are implemented in the effort model presented in Eq. (4):

$$\text{Total-effort (h)} = a \times (\text{total FP})^b, \tag{4}$$

where a and b are two constants; $a = 29.258$ and $b = 0.78$ according to the calculations of ISBSG.

The sample of results is presented in Table 3.

Table 3
Statistical results for all projects

Statistic measure	M1	M2
Minimum-error	−12,508	−9123
Maximum-error	127,884	106,109
MMRE	1.4	1.1
R^2	0.4829	0.7144
Error average	65%	35%

This table shows a comparison between the IFPUG version (using the original weights) that is indicated by M1 and the same version (using the new weights) that is indicated by M2. This comparison was obtained through three types of statistical analysis, i.e. error limits (maximum and minimum error), mean magnitude of relative error (MMRE) and *R*-squared.

All the statistic values are calculated and produced a strong results and positive indications. The error limits of estimation were reduced using the new weights system. For example the error domain was reduced from (−12,508 + 127,884) to (−9123 + 106,109) working-hour and thus applies in the same way with the other samples.

The MMRE was used to indicate the relative amount by which the predictions are over or underestimate the real value.

The projects that were used in this research represent small, medium and large business applications, ranging in size from 8 to 6670 FP. All these projects were used to prove the validity of the new weights in the FP count. This count was used as a main parameter for calculating the software projects effort in the effort estimation model.

A scatter plot of all projects in Fig. 8 presents the linear relation between the actual and the estimated effort using M1 and the convergence between the actual and the estimated effort is not strong, where the $R^2 = 0.4829$. Fig. 9 presents also the linear relation using M2. In this figure the degree of convergence between the actual and the estimated effort was increased clearly, where $R^2 = 0.7144$. According to the limits of correlation coefficient determined in (Hastings, 2001), the calculated results are strong.

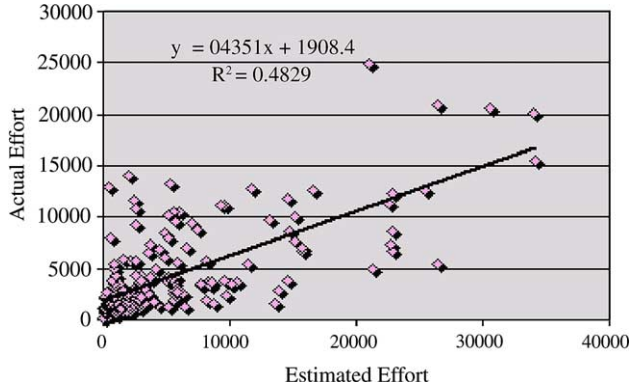

Fig. 8. A scatter plot of the effort estimation using the original FP weights.

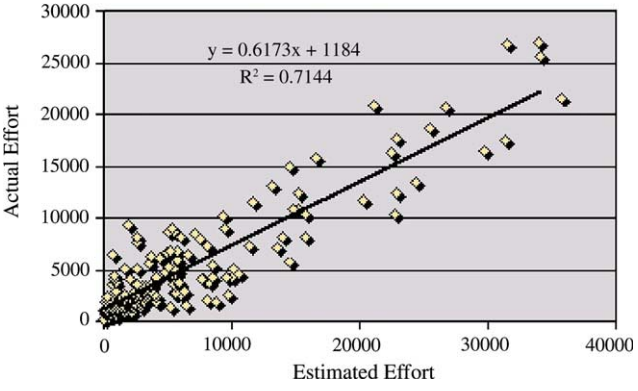

Fig. 9. A scatter plot of the effort estimation using the new FP weights.

7. Conclusions

The results that were measured according to the new weights were much better than those measured according to the original weights, when applied in the same FP version (IFPUG). The modified weights increased the estimation accuracy of this measure. The new weights system produced much accurate software effort values. It also reduced the error rate in effort estimation as against using the original weights system. All weights values of the new system are represented as an absolute scale, which is much flexible and better than the ordinal scale (see, Appendix A). The Neural Networks have proved to be strong prediction techniques in the software measurement field. This paper provides an active contribution for improving the complexity weights of the standard Function Point (IFPUG version). This will improve planning of software projects inside software organizations. The approach allows software engineering committee to support its decision making during the software life cycle when they need to measure software size in different times. It is advisable to use the FP measure with the new weights that provides much accurate values of estimation. In the future, uses of this measure with the new weights will improve the accuracy and process of finding the better balance between using Commercial-Off-The-Shelf (COTS), Reusable Software Component (RSCs), and developing new code. In addition, such a size metric will allow us to compare and position our performance and efficiency in software development, relative to other companies through benchmarking actions.

Appendix A

A comparison between the original and the new weights method for the ILF, where "O" denotes to the original weights and "N" denotes to the new weights.

Appendix B

The used data sets consist of the following fields that used in this research:

Method type	RET	DET		
		1–19	20–50	51–140
O	1	7	7	10
N	1	6.77–7.59	7.64–9.24	9.3–13.9
O	2	7	10	15
N	2	7.11–7.99	8.04–9.74	9.8–14.6
O	3	7	10	15
N	3	7.47–8.41	8.47–10.3	10.31–15.14
O	4	7	10	15
N	4	7.86–8.87	8.93–10.85	10.92–16.16
O	5	7	10	15
N	5	8.28–9.35	9.42–11.45	11.52–16.97
O	6	10	15	15
N	6	8.73–9.87	9.94–12.09	12.17–17.81
O	7	10	15	15
N	7	9.2–10.42	10.49–12.76	12.84–18.69
O	8	10	15	15
N	8	9.71–11.0	11.08–13.48	13.56–19.6
O	9	10	15	15
N	9	10.24–11.62	11.7–14.23	14.32–20.55
O	10	10	15	15
N	10	10.82–12.28	12.36–15.0	15.11–21.53
O	11	10	15	15
N	11	11.42–12.97	13.06–15.86	15.95–22.55
O	12	10	15	15
N	12	12.07–13.7	13.8–16.73	16.83–23.59
O	13	10	15	15
N	13	12.74–14.47	14.57–17.64	17.74–24.67

Proj-ID: the index of the project in the ISBSG data sets;
FP: the final value of Function Point according to the original weights;
FP (NN): the final value of Function Point according to the new weights;
TCF: the technical complexity value of GSC;
AE: the actual effort (total value of the project);
E1: the effort measured by FP using original weights;
E2: the effort measured by FP using the new weights.

Proj-ID	FP	FP (NN)	TCF	AE	E1	E2
15,873	919	694.3341	1.04	4815	5992.813	4815.796
10,581	74	70.388	1	804	839	807.7809
16,244	103	141.433	1	1396	1087.07	1392.112
12,174	183	156.9055	1.01	1514	1701.988	1509.532
22,868	112	100.2858	0.94	1073	1160.472	1064.661
14,924	26	29.49228	1.02	419	371.4701	409.8426
16,452	258	223.8152	0.95	2010	2224.887	1991.4
20,430	385	337.015	1	2761	3040.213	2740.382
19,364	71	59.906	1	737	813.2539	712.314
13,376	309	318.7998	0.81	2597	2561.016	2624.15
13,105	484	401.6596	1.05	3172	3634.327	3142.346
10,997	200	220.23	1	1935	1824	1966.474
13,911	150	118.882	0.95	1177	1457.457	1215.725
16,350	234	208.2136	1	1932	2061.736	1882.269
11,350	306	229.65	1	2082	2541	2031.779
12,080	162	182.9826	1.03	1754	1547.627	1701.862

Appendix B (*continued*)

Proj-ID	FP	FP (NN)	TCF	AE	E1	E2
18,593	722	586.4763	1.16	4164	4965	4221.628
10,337	120	119.92	1	1158	1224	1223.996
21,270	233	187.8249	0.91	1803	2054.86	1736.889
13,125	394	359.8722	1.13	2957	3095.506	2884.299
14,294	77	95.43612	0.78	1099	866.378	1024.284
18,880	1129	1023.703	0.99	6600	7036	6518.988
13,304	10	8.20338	0.94	64	176	151.0633
17,305	256	384.182	1	2941	2211	3035.173
19,704	387	441.9281	0.99	3287	3052.525	3385.469
18,245	43	31.38672	0.92	331	550	430.2357
21,842	1710	2486.695	1	13,130	9727.06	13,026.57
10,151	174	191.259	1	1876	1636	1761.61
19,279	843	907.5192	1.12	5814	5602.607	5934.337
18,723	239	195.3101	0.98	1661	2096	1790.647
20,701	95	104.399	1	1231	1020.632	1098.57
21,586	187	215.026	1	2063	1730.936	1930.135
12,169	2406	2134.019	0.97	11,700	12,695.66	11,561.64
17,239	300	252.4189	1.05	2039	2502	2187.257
20,738	309	358.895	1	3030	2561.016	2878.188
20,309	314	351.9183	1.01	2994	2593.282	2834.453
17,280	82	59.93	1	545	910	712.5366
19,962	578	786.953	1	5134	4173.961	5309.88
17,884	587	473.1138	1.12	3748	4224	3570.408
21,311	64	78.864	1	1073	750.0065	882.6938
15,773	61	49.30765	0.85	417	722.4402	611.9526
20,512	229	272.1104	0.95	2520	2027.292	2319.241
17,047	327	262.165	1	2464	2676	2252.853
20,400	520	365.346	1	3133	3843.503	2918.462
22,360	59	71.10112	0.83	1061	703.8971	814.1572
14,371	70	58.01154	0.93	432	804.3056	694.6817
15,619	5684	4853	1.15	21,014	24,824.17	21,945
10,793	598	415.695	1	2960	4286	3227.669
21,854	128	138.1145	1.07	1644	1287.859	1366.568
18,618	554	447.174	1	3700	4038	3416.775
22,029	379	327.743	1	2391	3003.193	2681.394
10,697	484	410.1111	1.05	2892	3634	3193.801
16,858	505	663.4428	1.07	4955	3756	4647.842
11,227	56	49.87363	1.03	296	675	617.4247
17,954	224	309.8203	1.15	2890	1992	2566.317
14,257	82	68.54872	0.82	440	909.954	791.269
11,812	2190	2948.653	1.2	14,520	11,797.53	14,878.22
13,799	296	220.5681	1.13	1609	2476.579	1968.829
17,556	526	439.587	1	3011	3878	3371.472
19,193	125	136.9536	1.07	1744	1264.254	1357.6
10,500	29	39.9504	0.7	932	404	519.3153
22,129	129	172.606	1.07	2064	1295.7	1626.104
11,465	101	75.8592	0.9	406	1070.571	856.3496
21,503	557	730.5716	0.88	5483	4055.194	5010.737
10,075	1502	1325.346	1	7490	8791	7973.729
20,641	261	319.7168	1.04	3123	2245.041	2630.036
12,939	119	116.8631	0.88	700	1216.666	1199.591
13,175	259	309.1038	1.07	3072	2231.611	2561.686
15,046	60	53.49985	0.85	136	713.1857	652.1682

(*continued on next page*)

Appendix B (*continued*)

Proj-ID	FP	FP (NN)	TCF	AE	E1	E2
20,443	124	108.99	1	616	1256.358	1136.073
10,218	152	118.4662	0.93	668	1472	1212.407
11,394	357	268.0456	0.8	1746	2866.328	2292.173
18,796	2015	1531.298	1.02	9503	11,100	8924.652
10,570	199	233.523	0.93	2640	1817	2058.457
21,780	179	142.499	0.95	776	1672.9	1400.29
17,665	186	167.0688	0.96	960	1723	1585.269
10,816	69	68.43835	0.97	139	795	790.275
15,090	97	78.96564	0.94	218	1037.353	883.5811
17,720	116	156.818	1	2180	1192	1508.876
11,163	73	101.6667	0.97	1798	831	1076.078
19,815	100	99.296	0.8	330	1062.294	1056.456
20,194	301	361.879	1	3631	2509.149	2896.837
16,725	67	72.18474	0.93	1584	777	823.8195
20,261	523	650.5747	0.87	5340	3860.788	4577.374
13,787	2372	3168.05	1.03	16,514	12,555.5	15,734.84
10,390	154	172.639	1	2408	1487	1626.346
14,207	134	101.913	1	286	1334.708	1078.112
21,174	203	165.902	1	750	1845.404	1576.627
10,991	294	238.1443	1.08	1248	2463	2090.162
17,085	2054	1805.227	1.05	9289	11,200	10,147.04
14,806	868	1166.109	0.84	8076	5731.787	7216.077
12,750	499	640.5055	1.03	5404	3721.886	4522.02
15,248	234	271.9467	1.11	3202	2061.736	2318.152
21,137	72	85.352	1	1824	822.1745	938.8394
13,511	562	698.6638	1.12	5727	4083.56	4839.203
15,372	163	196.0738	1.05	2684	1555.074	1796.106
15,199	102	101.5516	1.07	183	1078.829	1075.128
18,686	367	296.9385	1.08	1588	2928	2482.701
11,714	109	127.034	1	2212	1136.154	1280.272
16,617	244	202.2778	0.95	906	2130	1840.281
15,159	169	145.079	1	481	1599.544	1420.026
15,335	288	253.069	1	1251	2424.213	2191.649
12,835	1663	1060.642	1	5760	9517.888	6701.751
22,498	597	458.7948	0.98	2542	4280.6	3485.837
21,245	245	202.4968	0.81	893	2136.949	1841.835
15,798	496	665.214	1	5625	3704.421	4657.518
19,249	171	129.1717	0.88	273	1614.29	1297.045
20,577	323	264.1306	1.07	1238	2651.079	2266.017
18,197	99	137.1125	1.01	2387	1054	1358.829
19,037	103	100.0407	1.07	30	1087.07	1062.631
14,441	143	127.5502	0.83	225	1404.128	1284.328
16,241	464	280.7563	0.98	1304	3516.646	2376.521
17,574	187	174.7275	0.93	562	1731	1641.672
21,915	349	271.1864	0.91	1230	2816.103	2313.096
15,449	153	182.7806	1.07	2856	1480.144	1700.396
17,534	4562	5799.331	1.16	26,408	20,900	25,216.18
10,721	176	247.4693	1.05	3348	1651	2153.73
14,545	73	75.4299	0.91	2054	831.0678	852.5672
19,165	270	244.7636	1.07	893	2305.199	2135.341
11,425	142	196.145	1	3090	1396	1796.614
18,163	321	280.3409	1.02	1076	2638	2373.778
13,019	66	76.09632	0.93	2167	768.2258	858.4367
18,348	1634	1416.782	1	7060	9388	8399.646

Appendix B (*continued*)

Proj-ID	FP	FP (NN)	TCF	AE	E1	E2
19,059	264	210.2859	0.99	544	2265.143	1896.866
16,041	245	202.03	0.82	470	2136.949	1838.522
11,718	194	308.228	1	3934	1781.271	2556.024
20,697	220	191.9232	0.98	281	1964.872	1766.38
17,847	273	236.2859	0.86	528	2325	2077.428
11,968	103	129.7875	0.65	2853	1087.07	1301.866
11,368	244	195.49	1	191	2130.143	1791.933
12,906	1334	1073.277	1	5152	8014.311	6763.942
14,201	1092	938.2128	0.84	4416	6855.802	6090.314
16,598	213	275.507	1	4044	1916	2341.791
19,032	328	285.2838	1.02	665	2683.034	2406.361
17,926	1437	1055.178	1.09	4900	8493	6674.806
17,835	256	223.482	1.19	207	2211	1989.087
22,579	404	321.012	1	844	3156.619	2638.343
17,404	351	483.5667	1.04	5523	2828	3631.789
18,409	458	347.954	1	903	3481	2809.517
17,628	869	549.9657	0.93	2078	5737	4015.193
19,009	1014	841.3818	1.04	3504	6471	5594.216
11,458	331	289.2358	1.06	328	2702.156	2432.323
18,836	1882	1374.238	0.96	6068	10,500	8202.249
10,587	324	493.456	1	5867	2657	3689.593
22,408	182	217.4026	1.07	4160	1694.729	1946.754
18,110	300	440.5419	0.9	5637	2502	3377.183
11,056	395	483.7298	0.99	5900	3101	3632.745
21,533	152	198.9815	1.07	4216	1472.593	1816.848
11,871	4484	6670.275	1.29	30,582	20,632.12	28,123.98
22,169	856	732.581	1	2551	5669.884	5021.484
17,232	1186	962.2569	0.96	3711	7312	6211.717
13,437	610	490.2977	1.06	950	4353.133	3671.16
10,510	1790	2320.399	1.03	15,142	10,100	12,341.94
15,864	218	324.4973	0.96	5532	1950.926	2660.659
13,413	527	465.478	1.04	650	3883.801	3525.38
13,096	283	338.3496	0.97	5650	2391.322	2748.842
17,944	1883	1467.371	1.03	5525	10,486	8632.683
20,689	484	799.0049	1.04	8520	3634.327	5373.203
15,610	732	558.694	1	850	5018.377	4064.811
20,654	917	731.211	0.99	1586	5982.638	5014.157
18,986	291	317.1528	0.9	6063	2444	2613.57
10,985	303	346.5999	0.95	6297	2522	2800.985
21,997	798	610.602	0.95	851	5367.931	4356.483
20,652	475	561.0864	1.05	7662	3581.506	4078.381
19,493	1828	1479.729	0.95	5077	10,246.75	8689.341
21,710	436	561.017	1	7762	3349.995	4077.988
12,392	2529	1559.025	1	5333	13,199.11	9050.447
18,965	127	144.2574	1.07	5176	1280	1413.749
21,778	1457	1979.092	1	14,784	8585.055	10,901.55
20,772	804	1184.245	0.84	11,388	5399.386	7303.469
12,763	445	588.629	1	8499	3403.811	4233.71
19,665	1248	1961.306	1.14	15,165	7608.377	10,825.05
18,437	498	501.6502	1.12	8227	3716	3737.295
18,528	132	148.0311	1.13	6400	1319	1442.514
16,457	1984	1240.374	1	2540	10,922.63	7572.089
10,459	465	746.094	1	10,200	3522	5093.586
22,288	2291	3683.901	1.15	22,831	12,219.8	17,699.66

(*continued on next page*)

Appendix B (*continued*)

Proj-ID	FP	FP (NN)	TCF	AE	E1	E2
20,441	514	584.177	1.14	9572	3808.868	4208.713
11,108	1130	1832.053	1.03	15,749	7041	10,264.46
20,474	1617	1374.448	0.95	2621	9311.903	8203.225
18,725	1336	1011.626	1.09	633	8024	6458.924
17,744	475	577.6042	1.12	10,032	3581	4171.731
22,507	499	655.819	1	10,613	3721.886	4606.13
14,929	204	229.0993	0.89	8111	1852.491	2027.978
15,173	2087	3320.498	1.02	22,500	11,362.45	16,322.37
21,925	349	271.1864	0.91	1230	2816.103	2313.096
16,746	2376	3945.681	1.07	25,482	12,600	18,673.24
12,579	173	202.1015	0.98	8716	1628.998	1839.03
20,811	272	316.6175	1.11	9720	2318.507	2610.128
15,796	4352	6291.029	1.12	34,004	20,156.82	26,868.75
12,034	2734	1607.192	0.94	1950	14,026.5	9267.819
17,004	1072	1346.816	1.01	15,700	6757	8074.303
15,443	3113	5897.585	1	34,085	15,521.17	25,548.8
12,770	2145	2021.634	0.96	2400	11,608.01	11,083.9
12,310	529	880.9898	1.16	14,528	3895.292	5798.584
13,830	1457	2307.66	1	22,837	8585.055	12,289.06
22,417	358	386.2703	1.06	13,919	2872.588	3048.035
11,218	2470	2146.705	1.26	492	13,000	11,615.21
21,908	174	237.7142	1.24	13,528	1636.338	2087.217
21,490	1203	1846.002	1.11	22,784	7393.53	10,325.37
16,562	712	1030.136	1.16	21,220	4911.103	6550.92
19,339	1059	1267.955	1.11	22,920	6693.657	7703.103
15,246	806	1340.22	1.14	26,400	5409.859	8043.445
20,272	2143	2698.091	1.19	43,303	11,599.57	13,882.53
20,397	1223	1710.366	0.89	66,600	7489.232	9728.686
16,182	4913	8484.98	1.24	150,040	22,156.19	33,930.66

References

Bootsma, F., 2000. How to obtain accurate estimation in a Real-Time environment using full Function Points. Nortel Networks, 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'oo), Richardson, TX, pp. 105–113.

COSMIC, 1999. COSMIC–FFP manual, version 2.1 (December).

Fenton, N.E., Pfleeger, S.L., 1997. Software Metrics. A Rigorous & Practical Approach, second ed. PWS, Boston.

Freeman, J.A., Skapura, D.M., 1991. Neural Networks Algorithms, Applications and Programming Techniques. Addison-Wesley, Reading.

Hastings, T.E., 2001. A vector based approach to software size measurement and effort estimation. IEEE Transaction on Software Engineering 27 (4), 337–350.

IFPUG, 2002. Function Point Analysis Nears Approval As First International Standard For Software Functional Size. Technical report, available from <http://www.ifpug.org/about/ISOPress. htm>.

Kusumoto, S., Imagawa, M., Inoue, K., Morimoto, S., Matsusita, K., Tsuda, M., 2002. Function Point measurement from Java Programs. ACM Journal, ICSE'02 2, 19–25.

Longstreet, D., 2001. Function Point counting practices. Manual release 4.1. IFPUG.

Members of UKSMA, 1998. Mark II Function Point Analysis counting practices. Manual version 1.3.1. UKSMA Metrics Practices Committee, UK.

Pressman, R.S., 2001. Software Engineering. A Practitioner's Approach, fifth ed. McGraw-Hill, New York.

Rule, P.G., 2001. The importance of the size of the software requirements. In: NASSCOM Conference, 7–10 February, Mumbai, India. pp. 1–18.

SCT, 1997. Function Point frequently asked questions. Software composition—technologies. Technical report, available from <http://ourworld.compuserve.com/homepages/softcomp/fpfaq.htm>.

Shamsuddin, S.M., 2000. Higher order centralized scale-invariants for unconstrained isolated handwritten digits. Ph.D. thesis, UPM, Malaysia.

Shamsuddin, S.M., Sulaiman, M.N., Darus, M., 2001. An improved error signal for the backpropagation model for classification problems. International Journal of Computer Mathematics 76, 297–305.

Symons, C., 1991. Software Sizing And Estimating Mark II FPA. Wiley Series in Software Engineering. John Wiley, Baffin's Lane, UK.

Symons, C., 2001. Come back Function Point Analysis (modernised) all is forgiven! SMS Ltd., England. Technical report, available from <http://www.gifpa.co.uk/library/Papers/Symons/fesmafpa2001/paper.html>.