

Building an Effort Estimation Model for Software Projects

Galal GALAL-EDEEN, Amr KAMEL, and Hanan MOUSSA

Abstract— This paper discusses a model for effort estimation. This model focuses on minimizing effort variance by enhancing the adjustments made to the functional sizing techniques. A special focus is made on the adjustment factors which reflect the application's complexity and actual environment in which this application will be implemented. We introduce the idea of grouping the adjustment factors to simplify the process of adjustment and to ensure more consistency in the adjustments. We have also studied, in depth, how the quality of requirements impacts effort estimation. We introduce the quality of requirements as an adjustment factor in our proposed model. Our proposed model is based on three adjustment factors: "distribution", "difficulty", and "quality of requirements". Our study concentrates on Egyptian companies with an objective to enhance effort estimation in these companies.

Index Terms—adjustment factors, effort estimation, functional sizing

1. INTRODUCTION

The estimation of software projects is a process that includes the estimation of size, cost, effort and resources. It is a continual process that should be used throughout the life cycle of a project. Software size estimates are converted to software effort ones to derive effort for all work elements. Then the total cost of the whole software project is calculated. Obtaining accurate estimates of the system size from requirement specifications has been a problem that software developers have been trying to solve for many years. Estimating size and effort is one of the most important topics in the area of software project management. The dynamics of the software market lead to extended or new kinds of methods for the estimation of product size or development effort in the background of cost estimation. Software engineering has been largely unable to size software projects and applications accurately and consistently [2]. Sizing software is critical for accurately estimating and managing projects and for determining productivity, cost effectiveness and quality.

Functional size measurements measure software functional size and functionality by objectively measuring functional requirements, and quantifying and documenting assumptions in estimating software development. They aim to be objective, consistent, and auditable. Also, they

aim at being independent of technology (i.e. hardware platform, programming language, operating system,...). They employ adjustment factors that are created with the intent to measure the general aspects of size, as opposed to application-specific size [6]. The application of these characteristics should optimally adjust a functional size count to better predict effort. The most popular functional sizing units are function points and use case points [7,11].

The basic premise behind function point analysis is to measure the software functionality from the perspective of users. In function point sizing, visible external aspects of software that could be counted consist of five items: outputs, inquiries, inputs, files, and interfaces. Calculating function points for a project involves a number of steps. First, it is necessary to classify and count the five user function types delivered by the development project. Each of the functions that are assigned \complex, average, or simple. The complexity weights are applied to the initial function point count to arrive at an unadjusted function point (UFP) count [3,17]. Function point counting passes through an adjustment phase. This phase consists of scoring a group of general systems characteristics (GSC) that rate the general functionality of the application being counted. The 14 GSCs are intended to measure important things about the application [6]. They are: "data communications", "complex processing", "reusability", "multiple sites", "facilitate change", "distributed data", "performance", "installation ease", "operational ease", "transaction rate", "online data entry", "online update", "heavily used configuration", and "end-user efficiency" [10]. The following step is to calculate the total degree of influence (TDI) and then determine the value adjustment factor (VAF) using the following two equations:

$$TDI = \sum_{i=1}^{14} C_i \quad (1)$$

C_i is the degree of influence rating of each

GSC.

$$VAF = (TDI \times 0.01) + 0.65 \quad (2)$$

The last step is to calculate the final adjusted function point (*AFP*) count according to the following equation:

$$AFP = VAF \times UFP \quad (3)$$

Use case points resemble function points in measuring software functionality from a user's perspective. It analyzes use case actors, scenarios, and various technical and environmental factors and abstracts them into a sizing unit [7].

In this paper we will be discussing a proposed model for effort estimation. The model introduces a number of enhancements to adjustment factors of functional size measurements. One of the enhancements proposed in our model is grouping 14 GSCs into two groups: "distribution" and "difficulty of the problem". The grouping not only simplifies the counting process, but also reduces the probability of errors while counting. In addition, it improves the correlation of *AFP* with actual effort and decreases the effort variance.

Another important enhancement in the adjustment factors is the consideration of the quality of requirements in the proposed effort estimation model. Prior knowledge of the magnitude of requirements elaboration is instrumental in developing early estimates of a project's cost and schedule [1]. Quality of requirements is defined in IEEE Standard 830-1998 - Guidelines for Software Requirements Specifications by a set of characteristics among which are, unambiguity, consistency, completeness, stability, and others [19]. Poor quality of requirements is a major cause of project failures and project over-runs as stated in the Standish Group research. Incomplete requirements and changing requirements rank as the second and third main causes of project failures [20]. Thus, we want to rate the quality of requirements and consider it among the adjustment factors in our proposed model.

The model is built based on an empirical study using data from a sample of Egyptian software development companies.

The rest of the paper is organized as follows: Section 2 discusses the concerns and criticisms over adjustment factors. Section 3 discusses how the proposed model is built. In section 4, we

analyze, and evaluate the results of the model. We also show test results of the proposed model. Section 5 shows the limitations and future work for the proposed model.

2. CONCERNS AND CRITICISM OVER ADJUSTMENT FACTORS

Adjustment factors are created with the intent to measure the general aspects of size, as opposed to application-specific size. The application of these factors should optimally adjust the functional sizing to better predict the effort. Among users of functional sizing, there are some criticisms of the use and practical value of these adjustment factors. Following are some of the discussions and concerns.

• Number of adjustment factors

There are many opinions that suggest different numbers for GSCs ranging from two factors, to 14, to 19, or perhaps more. Some empirical research suggests that 5 to 7 factors are sufficient.

Symons [21] has suggested that an open-ended approach is needed to GSCs, and that more than 14 seem necessary. He has extended them to 19. The particular set of GSCs may need to vary over time. In criticizing Symons' proposal, Jones prefers to stick with the original 14, for the sake of conforming with "the assumptions of the original IBM assertions" or presumably to use his own formula in which only problem and data complexity are considered.

Conversely, some researchers suggest that 14 GSCs are too many. Researchers in [13,15] have found that there is common variation within the GSCs; in each of these studies, only 5 or 6 underlying factors seem to be involved, rather than 14. Also, researchers in [15] have found that it is very difficult to distinguish all 14 factors statistically. Either there is not enough data, because many of the data points are missing or there is not enough dispersion among their values. Only seven of 14 factors are significant in their sample.

• Redefining or rationalizing some of GSCs

Some researchers have noted that patterns can be observed in GSCs for different types of software [9]. Others have identified GSCs that may benefit from being redefined. Symons notes that "performance", "heavily used configuration", and "transaction rates" (all essentially performance constraints) are hard to differentiate;

so are “on-line entry”, “on-line update”, and “end user efficiency” (all to do with interactive use of the system) [15]. Both [15] and [14] note that “communications” and “interactivity” have become so pervasive nowadays that the associated GSCs have lost their discriminative value.

Other views in [15] suggest that some GSCs would benefit from being redefined. Two of the function point GSCs which are “on-line entry” and “data communications”, have lost their discriminative value, since almost all projects give them the maximum score. Also, another one, which is “on-line update”, seems to combine two different things: “backup and recovery”, and “online operation”.

There is a criticism that the GSCs are interrelated. A practical criticism of *VAF* is that not all of the right things are counted as GSCs and the number of GSCs needs to be increased.

In [18], a survey is performed to measure attitudes regarding the use of GSCs in function point sizing. Among the study’s conclusions is the notion that many feel that the GSCs may not reflect the current software technology, as the GSCs have remained virtually unchanged at least since 1991 while technology has markedly changed. Among their recommendations is to apply different weights to them. Also, the following factors need to be added:

- System reliability/security
- Management/development teams capability
- Development platform/primary programming language
- Software designed for multiple languages use vs. single language use
- Requirements volatility
- Purchased application or customized
- Size of database
- Help facilities in software applications

Lokan analyzes the practical application of GSCs to 235 software development projects. His research concludes that some of GSCs appear to be outdated and not applicable to today’s online world. He states that “they (GSCs) have less discriminative value now, and should perhaps be redefined” [14].

• **Measuring GSCs and *VAF* calculation**

Theoretical criticisms of *VAF* is that its construction involves operations that are

inadmissible according to measurement theory. Another criticism is that complexity appears in computing the unadjusted function points and again in GSCs. Therefore some double counting is taking place. Further criticism is that when computing *VAF*, it is not appropriate to give all the GSCs the same weight. Still others believe that *VAF* does not provide enough variation and that using it does not improve the effort estimation [18] Furthermore, in a research by Lokan, he concludes that “*VAF* is found not to improve the relationship between function points and effort and that it seems clear that the value adjustment factor should not be used” [14].

Although the number of terms and the coefficients have changed in later proposals [18] and there exist significant arguments regarding adjustment all having the same weight, the nature of the formula has stayed essentially the same [10].

• **Replacing old elements with new ones**

The notion that adjustments are a part of size measurement seems to be lost as soon as people think about the adjustment factor. The value of the adjustment phase is generally judged by whether or not it improves the explanatory relationship between size and effort.

The answers to questions such as these will come partly from proposals based on theory, partly from empirical research, and partly from practical experience. Value to practitioners must be kept in mind.

• **How adjustments are handled in Sample Egyptian Companies**

In [16], a survey is performed on a group of Egyptian companies to study their usage of adjustment factors. The survey encounters the following:

- Some companies employ functional sizing but only consider the unadjusted function count because they think that the adjustment factors have no value added.
- Others use almost the same values of the adjustment factors for all projects. Hence the adjustment factors are fixed for all the counts. Hence they are almost ineffective.
- Some companies employ functional sizing and make an effective usage of the adjustment factors. However, they need to

enhance the usage of adjustment factors to enhance, in turn, their effort prediction, i.e. to decrease the effort variance.

Hence, the research recommendation is to enhance the adjustment factors usage in these companies to improve effort estimation.

3. BUILDING THE PROPOSED MODEL

3.1. Dataset

The model proposed is piloted on a group of local software development companies. This sample of companies uses sizing methodologies based on functionality. They employ functional sizing techniques: either function points or a tailored version of use case points.

The researchers obtain the data for 32 projects: 25 are used to build the model and 7 to test the model. The source are three Egyptian software development companies. They share common characteristics that contributed to the success of the model, such as CMMI L3, which ensures a robust metrics system. The second one is their use of functional size measurements that do not complicate the development. It of the model. Data are provided to us by the metrics group of each company.

Project types are new development and enhancement projects. The application types are: e-government, data warehousing, web applications, management information systems, and decision support systems. The industries of these projects are: e-government, manufacturing, telecommunication, and healthcare.

Data gathered included data about sample projects in these companies such as projects' estimated size (functional size measurements), effort (both actual and estimated), values for adjustment factors and values quality of requirements specifications.

The average count of function points per project is 280 *UFP*. The average number of effort hours per project is 5691 effort hours.

3.2. Sizing Units Used

As mentioned previously, the sample companies employ functional sizing techniques in their estimations. Some of the companies, in our sample, estimates the size of their projects using

function points. One of the companies employs a modified version of use case points as its sizing unit. In order to normalize the sizing unit used, we have to make a conversion between modified use case points and function points. We size a sample of web pages, screens and reports using both function points and use case points (using the modifications employed by the sample companies). We derive a conversion factor that is used to convert the count of modified use case points to function point count. This step of size normalization is a preparatory step to unify the sizing unit used in the sample and hence throughout the research.

3.3. Evaluation of the Effort Estimation Process for the Sample Companies Before Applying the Proposed Model

First of all, we want to evaluate the estimation process of the dataset studied. We want to measure how well the estimation process is doing in these companies. The results found are conducted as follows:

The correlation between actual and estimated effort is 0.67. The average value of effort variance, according to the following equation, is $\pm 47\%$

$$EV = \frac{(AE - EE)}{EE} \quad (4)$$

EV is the effort variance.

AE is the actual effort.

EE is the estimated effort.

It is worth noting that throughout our research we will consider the absolute value of effort variance since both negative and positive values of effort variance indicate a problem in the estimation process of the organization in case they exceed the thresholds defined by the organization.

A positive effort variance means underestimation i.e. actual effort exceeds the estimated effort resulting in missing deadlines of project milestones, cost overruns of the project, and dissatisfied customers. Positive effort variance is the dominant case in our dataset. On the other

hand, a negative effort variance means overestimation. This means that the estimated effort exceeds the actual effort which results in over-allocation of resources which may be idle after the project is delivered leading to resource waste and low morale for staff. Overestimation leads to lost market opportunities as well. Hence, any variance exceeding organizational thresholds is unfavorable in business [5].

The value of 47% effort variance well exceeds the effort variance thresholds in the sample companies which is 10%. This shows a significant problem in the estimation process of the sample companies. Hence, one of the main objectives of this research is to decrease the effort variance.

3.4. Correlation of *UFP* and *AFP* with Actual Effort

To see if the adjustment process is effective for effort estimation, the correlation between function points and actual effort is investigated, using both adjusted and unadjusted function points as independent variables (*UFP* and *AFP*).

The correlation between actual effort and *UFP* (unadjusted function point) is 0.85.

The correlation between actual effort and *AFP* (adjusted function point) is 0.81.

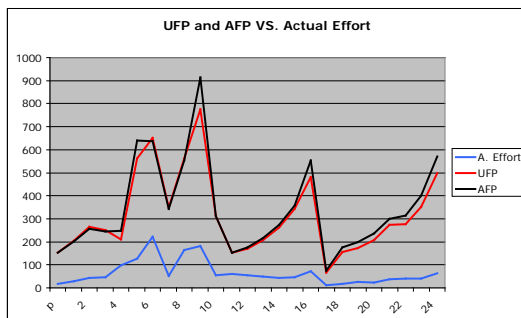


Fig. 1. Correlation of *UFP* and *AFP* with Actual Effort before applying the proposed model

The correlation of the function point count after adjustment with actual effort is less than the correlation of the function point count before adjustment with the actual effort. This contradicts with the main purpose of the adjustment factors. The adjustments are critical in the process of effort estimation since unadjusted count of function points does not consider application complexity and

environmental factors in sizing the application. Thus adjustment factors are very important in converting size to effort [21]. Problems in the adjustment factors cause errors in the estimated effort.

This means that the adjustments in our sample projects do not enhance effort estimation. On the contrary, it makes it less accurate. This result is very alarming to the researchers of this paper since it validates the criticisms that many papers and researches claim, that adjustments in function points are useless and can be misleading [18]. The result indicates that the high effort variance may be because of a problem in the adjustment factors.

3.5. Factor Analysis for the Adjustment Factors

Our initial hypothesis is that many of the adjustment factors are related to each other. Our hypothesis is based on “hints” in the Function Point Counting Practices and on previous endeavors [8,10,14]. Data gathered reveals some relationships between adjustment factors. For example, “performance”, “installation” and “operational ease” tend to correlate together. Also, “data communication” and “multiple sites” vary together. These relationships suggest that a smaller number of underlying dimensions might be sufficient, with some being captured several times over by different adjustment factors. To investigate this, factor analysis is used to study the common variation within the adjustment factors.

Factor analysis is a statistical procedure used to uncover relationships among many variables. This allows numerous inter-correlated variables to be condensed into fewer dimensions, called factors. A factor is a construct operationally defined by its factor loadings. Factor loadings are the correlation of a variable with a factor. Factor analysis is used to reduce a large number of variables to a smaller number of factors for modeling purposes [13]

The values of 14 GSCs in our sample projects are analyzed using factor analysis. Factor analysis suggests four factors using the maximum likelihood method for estimating factor loadings. Among them, two factors account for just over half the total variation in the adjustment factors. The first two factors each account for more than 60% of the variation. The

third and fourth factors explain less variation of about 25%.

“Data communications”, “complex processing”, “reusability”, “multiple sites” and “facilitate change” correlate with the first factor. “Distributed data”, “performance”, “installation ease”, and “operational ease” correlate with the second factor. “Transaction rate”, “online data entry” and “online update” correlate with the third factor. “End-user efficiency” alone correlates with the fourth factor.

We have tried to find a common theme for each factor as has been done in [10,14]. However we could not. This has implied to us that values given to adjustment factors lack consistency. This can be one of the factors to explain variance between effort estimates and actuals. The inconsistency in the values of the adjustment factors shows a problem in sizing our sample projects.

3.6. Grouping 14 GSC to two factors

The idea that adjustment factors can be reduced or grouped into two or three themes/groups can serve multiple purposes. It would be consistent with what Function Point Counting Practices says and with what previous researchers have deduced in their researches [8,10,14]. Also, it would simplify the counting process and reduce the probability of errors while counting, since the counters job would be much easier and less confusing.

The idea of reducing/grouping the adjustment factors into two or three themes/groups would enhance the adjustment process and achieve one of the objectives of this research.

We propose to impose grouping of factors into two themes: “distribution” and “difficulty of the problem to be solved”, so as to simulate grouping and see the possible outcomes of this grouping. We analyze these results and investigate if it would be favorable to effort estimation or not. So we have changed the values given to the adjustment factors such that there is consistency among them to achieve the grouping of “distribution” and “difficulty of the problem” to be solved. Factor analysis suggests, after grouping, two factors. The first factor accounts for 64% the total variation in the adjustment factors. The second factor accounts for 23% of the variation. “Performance”, “heavily used configuration”, “transaction rate”, “complex

processing”, “reusability”, “installation ease”, “operational ease” and “facilitate change” correlate with the first factor. So the first factor’s theme is “difficulty of the problem”. On the other hand, “data communications”, “distributed data”, “online data entry”, “online update”, and “multiple sites” correlate with the second factor. Therefore, the second factor’s theme is “distribution”.

We want to study the effect of grouping on effort estimation. So we want to use the groups proposed by the factor analysis instead of the 14 GSCs in the function point-based original methodology. We want to calculate the newly adjusted function point count based on groups and eventually calculate its correlation with the actual effort.

For replacing 14 GSCs with 2 factors, i.e., “distribution” and “difficulty”, we have to determine what is the weight to be used for each factor. As previously mentioned, there are many criticisms to 14 GSCs for using the same weight for each of them. Many studies suggest to use different weights [4,18]. We want to use different weights for each factor so that the adjustment to the function point count would be more correlated to actual effort and hence when the size is converted to estimated effort, eventually would minimize the effort variance.

In our proposed model, we apply different weights for different factors. The weight is calculated according to correlation factor with effort based on productivity which will be explained next.

3.7. Assigning Different Weights to Adjustment Factors

In our proposed model, we try to keep all weights correlated with effort such that we attain the final desired results of minimizing the effort variance. We face a problem when deciding on how to calculate the weights based on correlation with effort. We cannot not use the correlation with actual effort since in real situations of applying the model, actual effort will not be known yet. This is a model for estimating effort. Thus, we make our decision of relating to a normalized effort that is based on productivity. It is calculated by multiplying *UFP* (unadjusted function point) with the productivity of the company. Productivity figures are calculated from historical data provided by each company

(total effort of all projects of the company / total number of *UFPs* produced by these projects).

It is worth noting that we do not use the *AFP* (adjusted function point) since we are changing how adjustments are made and because of our concerns of how effective and consistent these adjustments are in calculating effort.

Thus we calculate the enhanced *TDI* (Total Degree of Influence). From it we calculate the enhanced *VAF* (Value Adjustment Factor), and from *VAF* we compute the enhanced *AFP*.

To see if reducing 14 GSCs to two factors is effective for effort prediction, the correlation between *AFP* and actual effort is investigated, using both the *AFP* before and after such reduction. The correlation between *AFP* and actual effort is 0.81.

After introducing grouping to adjustment factors, the correlation between enhanced *AFP* (based on our proposed model) and actual effort increased to 0.88 which is higher than the correlation between *AFP* and actual effort (0.81) and higher than the correlation between *UFP* and actual effort (0.85) as shown in Fig. 2.

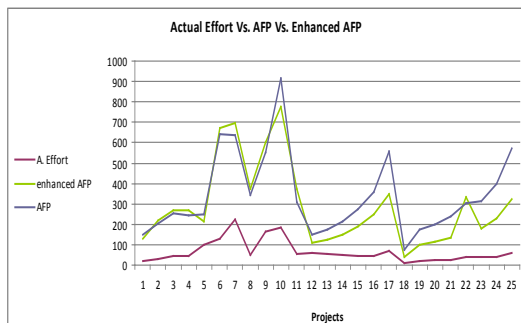


Fig. 2. Correlation of *AFP* and Enhanced *AFP* with Actual Effort after grouping

The enhanced *AFP* is more correlated with the actual effort, than the original *AFP*. This shows that the enhancements performed in adjustment factors by grouping would improve correlation with actual effort i.e. improving the effort estimation process as a whole.

In conclusion, grouping the adjustment factors achieves the following benefits:

1. Simplifying the counting process and reduce the probability of errors while counting since the counting job would be much easier and less confusing.

2. Improving the correlation of *AFP* with actual effort
3. Decreasing the effort variance.
4. Being consistent with Function Point Counting Practices and with what previous researchers have deduced in their researches.

Thus, we reduce 14 GSCs to two factors; “difficulty of the problem” and “distribution”. We use these two factors among our proposed adjustment factors, as will be discussed next.

3.8. Introducing Requirements Quality as a Factor

In our endeavor to improve effort estimation, we have investigated the relationship between the quality of requirements specifications and the effort estimation. We gather data about the quality of requirements specifications of the projects in our data sample. In order to help the companies rate the quality of requirements of each project, we provide these companies with a rating sheet. Each project is rated with respect to the level of quality of its requirements specifications. The rating provided for the quality of requirements for each project ranges on a scale of zero to five, from poor to excellent (similar to that of GSCs). We call the quality of requirements rating “*Q*”.

Our initial hypothesis is that quality of requirements affects effort variance. Hence, we calculate the correlation between quality of requirement rating proposed by this research and the effort variance.

The correlation value between quality of requirements rating (*Q*) and effort variance is 0.5 as shown in Fig. 3

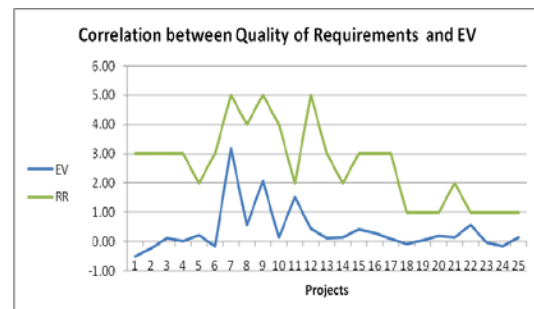


Fig. 3. Correlation between quality of requirements and effort variance

This value suggests a significant relationship between the quality of the requirements

specifications and effort variance. This means that 50 % of effort variance in our data set can be explained by the rating of quality of requirements specifications.

Our next step is to add quality of requirements rating as a factor to the adjustment factors to reflect application complexity and environmental impact on the project.

Thus, in our proposed model we have three adjustment factors: “difficulty of the problem”, “distribution”, and “quality of requirements”.

3.9. Steps of the Model

Step 1: Calculate Enhanced *TDI*

As previously mentioned, in our proposed model we apply different weights for factors. The weight is calculated according to correlation value with effort based on productivity (as explained in Section 3.7). All weights are normalized to add up to 14 such that we still maintain the function point equation for calculating *TDI*. The weight given to the quality of requirements rating is also calculated according to the correlation between quality of requirements rating and effort based on productivity. By adding quality of requirements rating as a factor, the *TDI* now is composed of 3 factors: “distribution factor”, “difficulty factor”, and “quality of requirements factor”. So our proposed equation for the enhanced *TDI* is shown below:

$$T = W_i \times S + W_j \times D + W_k \times Q \quad (5)$$

T is the enhanced total degree of influence.

W_i is the weight for distribution rating calculated by correlation with effort based on productivity.

S characterizes the “distribution” of a project. It ranges on a scale of zero to five, from no influence to strong influence (similar to that of GSCs).

W_j is the weight for difficulty rating calculated by correlation with effort based on productivity.

D characterizes the “difficulty” of a project. It ranges on a scale of zero to five, from no influence to strong influence (similar to that of GSCs).

W_k is the weight for quality of requirements rating calculated by correlation with effort based on productivity.

Q characterizes the rating provided for the “quality of requirements” of a project. It ranges on a scale of zero to five, from poor to excellent.

Step 2: Calculate Enhanced *VAF*

The following *VAF* equation used in the model does not change from the one used in the Function Point Counting Practices since we normalize the weights of 3 factors to the weights of 14.

$$F = (T \times 0.01) + 0.65 \quad (6)$$

F is the enhanced value adjustment factor.

Step 3: Calculate the Enhanced *AFP*

The enhanced *AFP* is calculated using the following equation, which is the same one used to calculate the *AFP* in the Function Point Counting Practices.

$$A = F \times UFP \quad (7)$$

A is the enhanced adjusted function point.

Step 4: Calculate the Enhanced Estimated Effort

The step in the proposed model is to calculate the enhanced estimated effort using the enhanced *AFP* (based on our proposed model). We use the productivity figures, which are calculated for all of the projects provided by each company, as previously mentioned.

Step 5: Calculate the Enhanced Effort Variance

We calculate the enhanced effort variance using (8).

$$E = \frac{(AE - M)}{M} \quad (8)$$

E is the enhanced effort variance.

M is the enhanced estimated effort.

4. ANALYZING THE PROPOSED MODEL AND EVALUATING ITS RESULTS

In our previous discussion of the proposed model, we have aimed at improving the effort estimation of software through the enhancements proposed by our model of adjustment factors applied to size. The enhancement of size estimation helps in enhancing effort estimation to be shown in this section. We move in two directions in our research; the first direction is to enhance 14 GSCs. We introduce the idea of grouping the adjustment factors into two main groups to simplify the process of adjustment and to ensure more consistency in the adjustment. The two groups are “distribution” and “difficulty”. Our second direction is to study the impact of requirements quality on effort estimation and incorporate the results of this study in our proposed model, i.e., include the factor of the quality of requirements specifications in the adjustment factors of the sizing estimation. In the second direction we have proved a significant relationship between the quality of requirements specifications and the effort variance. As shown in 3.8, we’ve explained that the correlation value between quality of requirements rating (Q) and effort variance is 0.5. This means that 50 % of effort variance in our data set can be explained by the rating of quality of requirements specifications.

Based on the research along both directions, we propose our model of only three adjustment factors: “distribution”, “difficulty”, and “quality of requirements”. It takes as an input the unadjusted function point count as a sizing unit and takes as an input a rating for each of the 3 factors for adjustment: difficulty, distribution and quality of requirements specifications.

“Difficulty” and “distribution” are the adjustments for the technical complexity of the application. As for the “quality of requirements”, it is the adjustment factor for the project environment. It is worth noting that the researchers of this paper are well aware that there are other adjustment factors which can be

considered for project environment. However, our model considers these three factors only, but may well be extended to include other factors, to be explained in the “Conclusions” section.

4.1. Evaluating the Results of the Model

To evaluate the results of the proposed model, we had to analyze the difference between the original effort variance and the enhanced effort variance resulting from the proposed model. It is worth noting that we have used the absolute effort variance values to omit the effect of signs in our analysis. We have calculated the difference between the absolute values of both effort variances before and after the application of the proposed model as shown in (9).

$$diff = \sum EV - \sum E \quad (9)$$

A positive value in $diff$ would mean that the enhanced effort variance (based on our proposed model) is lower than the original effort variance, while a negative $diff$ means that the enhanced effort variance is higher than the original effort variance. Based on our objective of this research, we are looking for the positive $diff$ instances.

Table 1: Results of applying the proposed model to 25 projects

<i>p</i>	<i>AE</i>	<i>EE</i>	<i>EV</i>	<i>EE from prod</i>	<i>Q</i>	<i>S</i>	<i>D</i>	<i>TDI</i>	<i>VAF</i>	<i>UFP</i>	<i>A</i>	<i>M</i>	<i>E</i>	<i>AFP</i>
1	18	36	-0.50	37.76	3	3	0	27.89	0.93	153	142	35	-0.49	151
2	30	39	-0.23	50.83	3	3	3	42.00	1.07	206	220	54	-0.45	202
3	44	40	0.12	65.64	3	1	4	38.40	1.03	266	275	68	-0.35	255
4	45	44	0.02	61.94	3	3	3	42.00	1.07	251	269	66	-0.32	246
5	98	80	0.23	52.07	2	2	3	32.70	0.98	211	206	51	0.93	249
6	128	152	-0.16	138.68	3	5	3	50.31	1.15	562	648	160	-0.20	641
7	223	53	3.18	160.40	5	3	3	52.28	1.17	650	762	188	0.19	637
8	51	33	0.57	86.37	4	3	3	47.14	1.12	350	392	97	-0.47	343
9	164	53	2.07	138.93	5	3	3	52.28	1.17	563	660	163	0.00	552
10	182	157	0.16	191.00	4	2	3	42.99	1.08	774	836	206	-0.12	913
11	55	22	1.52	50.38	2	5	3	45.17	1.10	310	342	56	-0.01	310
12	60	41	0.45	24.86	5	0	1	30.41	0.95	153	146	24	1.51	151
13	56	50	0.11	27.76	3	0	1	20.13	0.85	171	145	24	1.36	176
14	48	42	0.15	33.57	2	0	1	14.99	0.80	207	165	27	0.80	215
15	43	30	0.43	42.68	3	0	1	20.13	0.85	263	224	36	0.20	273
16	46	36	0.29	55.75	3	0	1	20.13	0.85	343	292	47	-0.02	357
17	71	65	0.10	78.39	3	0	1	20.13	0.85	482	411	67	0.07	555
18	12	13	-0.07	10.56	1	0	0	5.14	0.70	65	46	7	0.59	75
19	18	18	0.04	25.09	1	0	0	5.14	0.70	154	108	18	0.05	176
20	25	21	0.21	28.22	1	0	0	5.14	0.70	174	122	20	0.28	198
21	23	20	0.16	33.88	2	0	0	10.28	0.75	208	157	26	-0.09	238
22	39	25	0.57	44.53	1	4	4	40.58	1.06	274	289	47	-0.18	301
23	39	41	-0.04	44.97	1	0	0	5.14	0.70	277	194	32	0.24	315
24	40	48	-0.16	56.98	1	0	0	5.14	0.70	351	246	40	0.00	400
25	62	54	0.15	81.22	1	0	0	5.14	0.70	500	351	57	0.09	570

Table 2: Diff results for 25 projects

<i>P</i>	<i>EV</i>	<i>E</i>	<i>Diff</i>
1	0.500693	0.4867303	0.01
2	0.234928	0.4517109	-0.22
3	0.116283	0.3490018	-0.23
4	0.018506	0.3190356	-0.3
5	0.228061	0.9338651	-0.71
6	0.158619	0.2012457	-0.04
7	3.184265	0.1859825	3
8	0.565344	0.470698	0.09
9	2.070705	0.0048522	2.07
10	0.162892	0.1163217	0.05
11	1.523875	0.0096074	1.51
12	0.452155	1.5148171	-1.06
13	0.114846	1.3576603	-1.24
14	0.152304	0.7973987	-0.65
15	0.426413	0.1951774	0.23
16	0.293708	0.0209116	0.27
17	0.102948	0.0708186	0.03
18	0.065769	0.5911824	-0.53
19	0.042681	0.0468735	0
20	0.210476	0.2841851	-0.07
21	0.1615	0.0892427	0.07
22	0.569412	0.1771274	0.39
23	0.035001	0.2411746	-0.21
24	0.162202	0.001631	0.16
25	0.14726	0.0912765	0.06
Average	0.47	0.36	0.11
Total	0.339	0.005	0.334

As shown in Tables 1 and 2, the following results are obtained:

1. Out of 25 projects in our sample, we have 13 projects with positive *diff* and 11 with negative *diff* and one with zero *diff*.
2. The average absolute value for the original effort variance is 47%, whereas

the average absolute value for the enhanced effort variance (based on our proposed model) is 36% with an improvement of 11% in *diff*.

3. If we add up the total effort variances in all of the projects of our sample, the total of absolute values for the original effort variance is 33.9%, whereas the total absolute value for the enhanced effort variance is 0.5% with a positive 33.4% *diff*.
4. Correlation between enhanced *AFP* based on our proposed model and actual effort increases to 0.91 which is higher than the correlation between the original *AFP* and actual effort (0.81) and higher than the correlation between *UFP* and actual effort (0.85)
5. The correlation between the actual effort and the enhanced estimated effort based on our proposed model increases to 0.92 with a 37% improvement from the correlation between the actual effort and the original estimated effort which was 0.67 as shown in Fig. 4.

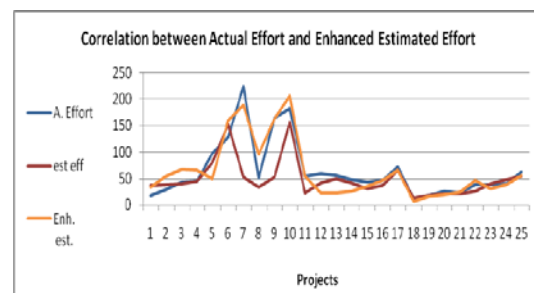


Fig. 4. Correlation between actual effort and original estimated effort and enhanced estimated effort based on our proposed model

T7	145	4	4	4
----	-----	---	---	---

4.2. Testing the Proposed Model

The sample of 25 project data is used to build the proposed model. However, we still want to test the model with test data to check the validity of the model.

Data of 7 projects are provided by the local companies to the researchers to test the proposed model.

4.2.1. Input Data to the Model

1. *UFP* – Unadjusted Function Point count
2. Ratings for 3 factors of adjustment: *S*, *D*, *Q*

To be able to rate them, the user is provided with a rating sheet for each of the factors with scale of zero to five, from “poor” to “excellent” for the rating of the quality of the requirements and from “no influence” to “strong influence” for distribution and difficulty of the problem. The test data provided are as shown in Table 3:

Table 3: Test data

<i>P</i>	<i>UFP</i>	<i>Q</i>	<i>S</i>	<i>D</i>
T1	85	3	4	1
T2	125	3	1	1
T3	80	2	2	2
T4	143	3	1	2
T5	370	2	3	2
T6	88	2	1	3

4.2.2. Using the model

1. Calculating the Enhanced *TDI*

The enhanced *TDI* is calculated using (5). It is worth noting that W_i , W_j , and W_k are derived by the model i.e. using the correlations derived from the model.

2. Calculate the Enhanced *VAF*:

As in building the model, the original *VAF* equation, (2), is used which is the same one used by the function point counting method.

3. Calculate the Enhanced *AFP*

The enhanced *AFP* is calculated using (7).

4. Calculate the Enhanced estimated effort

In this step, we use the productivity figures provided by model.

5. Calculate the Enhanced Effort Variance

We calculated the enhanced effort variance using (8).

4.2.3. Results Obtained from Applying the Proposed Model to Test Data

The results obtained from applying the proposed model on the data of the 7 test projects are shown in Tables 4 and 5. Table 4 shows the values of enhanced *AFP*, enhanced effort estimate and enhanced effort variance. Table 5 shows *diff* values of 7 projects.

Table 4: Results of applying the proposed model to test data

<i>P</i>	<i>VAF</i>	<i>AE</i>	<i>EE</i>	<i>EV</i>	<i>A</i>	<i>M</i>	<i>E</i>
T1	1.04	18	11	0.59	88	22	-0.18
T2	0.902	17	14	0.16	113	19	-0.13
T3	0.93	8	7	0.16	74	13	-0.37
T4	0.94	28	27	0.04	134	23	0.21
T5	0.976	87	50	0.73	361	61	0.42
T6	0.922	12	11	0.11	81	14	-0.13
T7	1.21	29	26	0.11	176	30	-0.02

Table 5: Diff results of the test data

<i>P</i>	<i>EV</i>	<i>E</i>	<i>Diff</i>
T1	0.59	0.18	0.41
T2	0.16	0.13	0.03
T3	0.16	0.37	-0.21
T4	0.04	0.21	-0.17
T5	0.73	0.42	0.31
T6	0.11	0.13	-0.02
T7	0.11	0.02	0.09
Total	0.36	0.31	5%
Average	0.27	0.21	6%

As shown in Tables 4 and 5:

1. Out of 7 projects in our test data, we have 4 projects with positive *diff* and 3 with negative *diff*.
2. The total value for the original effort variance is 36%, whereas the total value for the enhanced effort variance is 31% with a 5% improvement in *diff*.
3. If we add up the average absolute effort variances in all of the projects of our sample, the average of absolute value for the original effort variance is 27%, whereas the average absolute value for

the enhanced effort variance is 21% with a positive 6% *diff*.

5. CONCLUSION

In the previous sections we have discussed how our proposed model is based on three adjustment factors: “distribution”, “difficulty”, and “quality of requirements”. We have also shown how the model achieved its objective to enhance effort estimation in the sample projects.

However, the proposed model has a number of limitations:

- Only two functional sizing units are included. Other sizing units such as SLOC (Source Lines of Code) are not considered in our model thus our model results are not validated for sizing units other than the ones considered in this research.
- Results obtained are not validated or generalized on data from other companies.
- Data points are limited to only 25 projects in addition to 7 more projects for testing and validating the model.
- Although the effort variance declined by using the model, it is worth noting that the average value of effort variance reached is still above the effort variance thresholds of the organizations from

which the sample is brought. So, there is more room for improvement.

It is worth noting that the proposed model can be extended in many directions:

- Further improvement to the results obtained to achieve lower values for effort variance.
- Include more adjustment factors, especially in the area of environmental factors affecting the project such as developer skills, project management proficiency, organizational maturity, and others.
- Include projects data from other companies. This would enrich the model to reflect the other characteristics of the local (or international) software development companies.
- The model can be extended to be an estimation model not just for effort, but also other metrics as cost, resources, and others.

6. REFERENCES

- [1] Afdal Malik, A.A. and Boehm, B., "An Empirical Study of Requirements Elaboration", *XXII Simpósio Brasileiro de Engenharia de Software*, Campinas, Brazil, Oct. 2008, pp. 113-123.
- [2] Agarwal, R., M. Kumar, S. Malick, R.M. Bharadwaj, D. Anantwar, "Estimating Software Projects", *ACM SIGSOFT*, Vol. 26, No. 4, Jul. 2001, pp. 60-67.
- [3] Ahmed, F., Bouktif, S., Serhani, A., and Khalil, I., "Integrating Function Point Project Information for Improving the Accuracy of Effort Estimation," *Proc. Second International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP '08*, Oct. 2008, Valencia, Spain, pp. 193-198.
- [4] Al-Hajri, M. A., Ghani, A. A. A., Sulaiman, M. N., and Selamat, M. H., "Modification of standard function point complexity weights system", *Journal of Systems and Software*, Vol. 74, No. 2, Jan. 2005, pp. 195-206.
- [5] Asproni, G., "From Finger in the Air Estimating to Finger Pointing Tracking", *Methods and Tools*, Vol. 16, No. 4, Winter 2008, pp. 2-13.
- [6] Bharadwaj, A.K., and Nair, T.R.G., "Mapping General System Characteristics to Non-Functional Requirements", *IEEE International Advance Computing Conference (IACC)*, Mar. 2009, Patiala, India, pp. 1634-1638.
- [7] Carroll, E. R., "Estimating Software Based on Use Case Points", *Object-Oriented, Programming, Systems, Languages, and Applications (OOPSLA) Conference*, San Diego, California, USA, ACM, Oct. 2005, pp. 257-265.
- [8] Finnie, G.R., Wittig, G.E., and J-M Desharnais, "Reassessing Function Points," *The Australian Journal of Information Systems-AJIS*, Vol. 4, No. 2, May. 1997, pp. 39-45.
- [9] Garmus, D. and Herron, D., *Measuring the Software Process: a practical guide to functional measurements*. Prentice-Hall, Upper Saddle River NJ, 1996.

- [10] IFPUG, *Function Point Counting: Practices Manual*, IFPUG, 2004.
- [11] IFPUG, *IT Measurement*, Addison-Wesley, 2002.
- [12] Kitchenham, B.A., "Empirical studies of Assumptions that underlie software cost-estimation models", *Information and Software Technology*, Apr. 1992, Vol. 34, No. 4, pp. 211–218.
- [13] Kline, P., *An Easy Guide to Factor Analysis*, New York, Routledge, 1994.
- [14] Lokan, C. J., "An Empirical Analysis of Function Point Adjustment Factors", *Information and Software Technology*, Vol. 42, No. 9, Jun. 2000, pp. 649-659.
- [15] Lokan, C., and Abran, A. (1999). Multiple viewpoints in functional size measurement, *International Workshop on Software Measurement (IWSM)*. Lac Supérieur, Québec, Canada, Sept. 1999, pp. 121-131.
- [16] Moussa, H., Galal-Edeen, G. H., and Kamel, A., "Enhancing Software Sizing Adjustment Factors", *Fourth International Conference on Intelligent Computing and Information Systems – ICICIS*. Cairo, Egypt, ACM, Mar. 2009, pp. 438-444.
- [17] Parthasarathy, M.A., *Practical Software Estimation*, India. Dorling Kindersley Pvt. Ltd., 2007.
- [18] Prater, M. D., and Willoughby, J. C., The Adequacy of the Fourteen General Systems Characteristics as Function Point Adjustment Factors. *Faculty of the School of Logistics and Acquisition Management of the Air Force Institute of Technology, Air University*. Master of Science in Cost Analysis, 1999.
- [19] Software Engineering Standards Committee of the IEEE, *IEEE Recommended Practice for Software Requirements Specification*, IEEE, 1993.
- [20] Standish Group, *CHAOS Report*. Standish Group, 1994.
- [21] Symons, C. , "Function Point Analysis: Difficulties and Improvements", *IEEE Transactions on Software Engineering*, Jan. 1988, Vol. 14, No.1, pp. 2-11.