ASSESSING SOFTWARE COST ESTIMATION MODELS: CRITERIA FOR ACCURACY, CONSISTENCY AND REGRESSION

Bruce W.N. Lo and Xiangzhu Gao
School of Multimedia and Information Technology
Southern Cross University
Lismore, NSW Australia 2480
Email: blo@scu.edu.au

ABSTRACT

One of the problems in software cost estimation is how to evaluate estimation models. Estimation models are usually evaluated against two attributes: estimation accuracy and estimation consistency. A number of measures are reported in the literature, but they have shortcomings. There is no generally accepted standard to evaluate estimation models and the existing measures sometimes are not consistent among themselves. This paper examines existing measures of estimation accuracy and consistency and proposes two new ones: the weighted mean of quartiles of relative errors (WMQ) as a measure of accuracy and the standard deviation of the ratios of the estimate to actual observation (SDR) as a measure of consistency. Besides, a new regression criterion is proposed to determine model parameters. The proposed measures and criterion were tested with a data set from real world software projects. Results obtained show that these new measures and criterion overcome many of the difficulties of the existing ones.

INTRODUCTION

Since the early 1950s, software development practitioners and researchers have been trying to develop methods to estimate software costs and schedules (Abdel-Hamid, 1990). Software cost estimation models have appeared in the literature over the past two decades (Wrigley *et al*, 1991). However, the field of software cost estimation is still in its infancy (Kitchenham *et al*, 1990). The existing cost estimation methods are far from standardised and reliable (Rowlands, 1989). There is a need to evaluate estimation models and improve modeling processes. This paper will focus on how to quantitatively evaluate software cost estimation models. A new approach to determine model parameters is also proposed.

In the field of software cost estimation, estimation models are usually evaluated against two attributes: estimation accuracy and estimation consistency. The rules or measures needed to describe these two attributes will be discussed in this paper.

Measurement may be used for two purposes: assessment and prediction (Fenton, 1994). For assessment, the results of comparing different models may be used to judge which theory is more successful at explaining the behaviour of cost factors (Pfleeger, 1991). Successful models would provide further insight into software development processes. To assess models, we need to have common standards. Although some measures for estimation accuracy and consistency have been introduced in the literature, they are not generally accepted for model assessment (Verner *et al*, 1992). Sometimes these measures are not consistent among themselves. For example, a model may be better than another one with respect to one measure but poorer with respect to a different measure.

For prediction, the measurement may provide feedback to improve the modeling process so that the model can satisfy the rules of measurement as far as possible. Modeling is associated with measurement (Kan *et al*, 1994). In this context, it is necessary to define the procedures for determining model parameters and interpreting the results (Fenton, 1994). This leads to the problem of how to determine the model parameters. Unfortunately there is also no generally accepted criterion for researchers to follow in the modelling process. This paper, therefore, will also discuss the criteria for determining model parameters.

The paper is organised into three parts. The first part examines the measures used to evaluate estimation models. To overcome the shortcomings of existing practice, two new measures are proposed: *the weighted mean of quartiles of relative errors* (WMQ) which provides a measurement of accuracy *and the standard deviation of the ratios of the estimate to actual value* (SDR) which provides a measurement of consistency. The second part examines traditional mathematical procedures for formulating costing models and proposes a new regression criterion called *least sum of logarithmic ratios of estimate to actual value*. This is an unbiased method of finding parameters in a cost estimation model when the functional form of the model is known. The third part assesses the proposed measures and criterion.

A data set from real world software projects was used to examine the proposed measures and the regression criterion so as to demonstrate their advantages over the other measures and criteria currently reported in the literature.

MEASURES OF ACCURACY

Accuracy is defined as the measure of how close a result is to its correct value (Deeson, 1991). There are two ways to compare a result and its correct value: their difference and their ratio.

Let n be the number of projects in a data set, act_i be the i^{th} (i=1,2,3,...,n) actual observed value and est_i be the corresponding estimated value. The difference measure of estimation accuracy is based on the difference between estimated value and actual value

$$est_i - act_i \qquad (i=1,2,3,...,n)$$

The ratio measure of accuracy is based on the ratio of estimated value to actual value

$$\frac{est_i}{act_i} \quad (i=1,2,3,...,n)$$

In evaluating the accuracy of software cost estimation models, both difference and ratio measures have been used. These are discussed below.

Difference Measures of Accuracy

(1) Mean of absolute errors (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|act_i - est_i\right|$$

(2) Root mean of squares of error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(act_i - est_i)^2}$$

(3) Coefficient of determinant (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(act_i - est_i)^2}{\sum_{i=1}^{n}(act_i - \overline{act})^2}$$

where $\overline{act} = \frac{1}{n}\sum_{i=1}^{n} act_i$ is the mean of n actual observed values. For a given set of data $\sum_{i=1}^{n}(act_i - \overline{act})^2$ is a constant. Therefore R^2 is a distance measure.

(4) Mean of residues (MR)

$$MR = \frac{1}{n}\sum_{i=1}^{n}(est_i - act_i)$$

Ratio Measures Of Accuracy

(1) Mean (or average) of relative errors (ARE)

$$ARE = \frac{1}{n}\sum_{i=1}^{n}\frac{act_i - est_i}{act_i}$$

(2) Mean of magnitude of relative errors (MRE)

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{act_i - est_i}{act_i}\right| = \frac{1}{n}\sum_{i=1}^{n}\left|1 - \frac{est_i}{act_i}\right|$$

(3) Root mean of squared relative errors (RMSRE)

$$RMSRE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{act_i - est_i}{act_i}\right)^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{est_i}{act_i}\right)^2}$$

Many measures are based on magnitude of relative errors, mre_i

$$mre_i = \left|\frac{act_i - est_i}{act_i}\right| \quad (i=1,2,3,...,n)$$

MRE is the most widely used measure in the literature. However, it is influenced by extreme values or outliers. A smaller MRE is not always better than a larger one (JØrgensen, 1994). To address this problem, some single values of mre_i (i=1,2,3,...,n) have been used to measure the estimation accuracy. They are described below.

(4) Prediction at level l (PRED(l))

Conte *et al* (1986) put forward a single value measure, prediction at level l

$$PRED(l) = \frac{k}{n}$$

where k is the number of projects in a set of n projects whose $mre \leq l$. They suggested that an acceptable accuracy for a model is PRED(0.25) $\geq$ 0.75, which is seldom reached in reality.

(5) Third quartile of *mre* (Q_3)

Conte *et al*'s standard can be expressed in another way: 75% of the *mre*'s are less than or equal to 0.25. In terms of quartiles, the third quartile, Q_3, of the *mre*'s is less than or equal to 0.25, i.e. $Q_3 \leq 0.25$. As mentioned above, this is too high a standard at present, but it should be a goal to pursue. The smaller the Q_3, the more accurate the estimation.

Both Q_3 and PRED(0.25) avoid considering some extremely poor predicted values. So they eliminate the influence of extreme values.

(6) Other measures

The median of *mre*'s (Q_2, second quartile) has also been used as one of the measures of accuracy (JØrgensen, 1994). It is only a middle number among the ordered *mre*'s, so it hardly presents an average of all the *mre*'s.

Miyazaki *et al* (1991) believed that *mre* based measures favour underestimation. (This problem will be discussed in detail later.) To address this problem, they defined the relative error in another way:

$$r_i = \frac{est_i - act_i}{act_i} \qquad (est_i - act_i \geq 0)$$

$$r_i = \frac{est_i - act_i}{est_i} \qquad (est_i - act_i < 0)$$

They argued that by using r_i, a balanced evaluation is obtained for both underestimated and overestimated cases. With the above definition of relative error, MRE, RMSRE or PRED(0.25) can be used for accuracy evaluation (Miyazaki *et al*, 1991).

Fenton (1994) argued that good measurement should be meaningful. In practice, it is generally accepted to express relative error by comparing error with actual observed value. With this commonly accepted idea of relative error, if the relative errors are 0.40, 0.50, 0.60 and 0.70 in the case of underestimation, the corresponding *r* values are 0.67, 1.00, 1.50 and 2.33. In an extreme case, if the relative error is 0.95, which is not unusual, the *r* value can be as high as 19.00. The variance of r_i must be large and the outliers take an important role in MRE. The large variance will also make RMSRE and PRED(0.25) values quite different from those commonly accepted. Moreover this measure penalises underestimation too much.

Jenson and Bartley (1991) proposed the weighted MRE (WMRE). They believe that one weakness of MRE is that the relative size of each individual observation is not recognised. WMRE is calculated such that each observation's relative error is weighted by the ratio of the observed value to the average of the observed values. They argued that by weighting larger projects more heavily than smaller ones, the WMRE recognises the greater cost associated with estimation errors on larger projects. However, on closer examination, we can prove that their argument is untenable since

$$WMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{act_i}{\overline{act}}\left|\frac{act_i - est_i}{act_i}\right| = \frac{1}{n\overline{act}}\sum_{i=1}^{n}\left|act_i - est_i\right|$$

In this expression, it does not appear that larger projects are weighted more heavily than smaller ones. On the contrary, the weight of project size included in relative error disappears from the formula. So it falls into the category of difference measure.

Occasionally, a combination of difference measure and ratio measure is used for accuracy. JØrgensen (1994) introduced $\mathrm{PRED}_2(0.25,0.5)$, the percentage of projects with *mre* ≤ 0.25 or estimation error ≤ 0.5 days. $\mathrm{PRED}_2(0.25,0.5)$ is actually PRED(0.25), because "estimation error ≤ 0.5 days" only contributes to the percentage when the estimation is performed for a project of less than two days.

Discussion On Measures Of Accuracy

Kitchenham *et al* (1990) indicated that we should distinguish between *fit accuracy* and *prediction accuracy*, i.e. how well a model fits the data from which it was generated and how good a prediction from the model will be.

To assess fit accuracy, R^2 is a good measure. It can be seen from the definition of R^2 that it is related to the variance of actual observed values. The larger this variance is, the easier it is to obtain a large R^2. R^2 measures the proportion of the variation in the dependent variable that is explained by the regression equation (Kenkel, 1989). The higher the value of R^2, the greater the explanatory power of the regression equation. When R^2 is close to 0, either the functional form of the model does not fit the data set from which it was generated or more independent variables are needed to further explain the variation in the dependent variable.

To assess prediction accuracy, difference measures are not suitable. With respect to software cost estimation, the prediction error increases with the magnitude of the observed value. The larger the project is, the harder it is to estimate the effort. Difference measures are not adequate if they are used for both large projects and small ones, because they do not take into consideration the size of projects. Therefore difference measures should not be used to assess the prediction accuracy for they penalise the prediction for large projects. On the other hand, relative error is an average of prediction error in every unit of effort, which reflects "error rate". It takes into account the project size and allows larger absolute error for larger projects. Therefore, ratio measure is more suitable for the assessment of the accuracy in software cost estimation.

Among the ratio measures, all mean measures are influenced by very large extreme *mre* values, which usually occur in software estimation. In the context of accuracy evaluation for software cost estimation, they cannot play the role of the measure for central tendency. Some of them favour underestimation. For the single value measures, although they eliminate the influence of outliers, they can hardly provide an overall accuracy. For example, PRED(0.25) takes into account less than 75% of all the *mre*'s, because Conte's acceptable level can seldom be achieved. Although Q_3 indicates that 75% of the projects are estimated with *mre* less than or equal to Q_3, it does not provide any detailed information on these *mre* values. For the purpose of comparison among different models, the single value measures are rough and stochastic. Moreover, these single value measures and mean measures are not always consistent when they are used to compare different models.

The above discussion may be summarized in three points:

- The difference measure is not suitable for the evaluation of software cost models. The ratio measure is preferable.
- The mean measures take into account every single relative error, but they are significantly influenced by outliers, which frequently occur in software cost estimation. Some of them favour underestimation.
- Single value measures are not influenced by outliers, but they are stochastic and lose too much information. Therefore the evaluation by single value measures is not always reliable when they are used to compare different models.

Because of the above drawbacks of the existing accuracy measures, there is a need for a better measure for accuracy evaluation.

A New Measure Of Accuracy - Weighted Mean of Quartiles of mre's (WMQ)

To resolve the above problems, it is proposed to use the weighted mean of quartiles (WMQ) of *mre* to measure the prediction accuracy.
The third quartile (Q_3) is the most important as 75% of *mre*'s values are less than it. So it is weighted with 75. The second quartile (Q_2) is weighted with 50, and the first quartile (Q_1) is weighted with 25. The WMQ is defined as:

$$WMQ = \frac{25Q_1 + 50Q_2 + 75Q_3}{150} = \frac{Q_1 + 2Q_2 + 3Q_3}{6}$$

There are two assumptions underlying the use of WMQ:

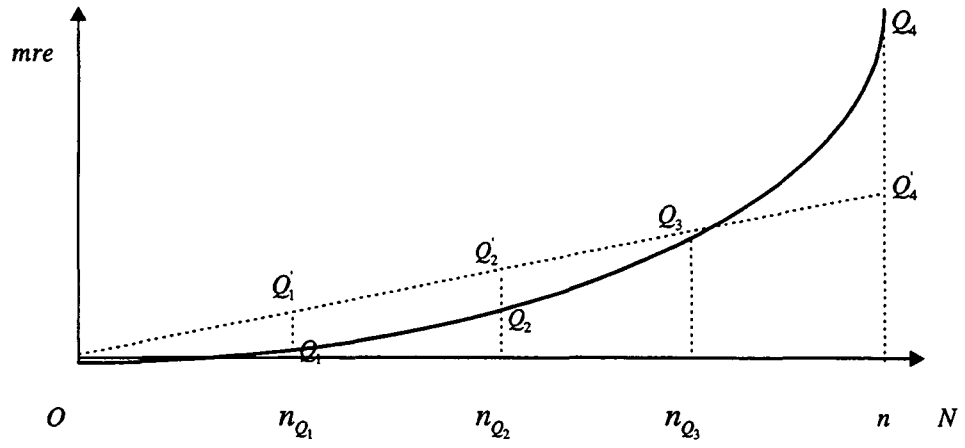
1) The number of outliers is less than 25% of all the *mre* values. This assumption is generally true.
2) If the estimation of 75% of the projects is acceptable, the model is desirable. Because of the present low level of poor estimation accuracy, this assumption is reasonable.

Unlike the mean measures, the WMQ is not influenced by extreme *mre* values, and at the same time it provides more general information about the distribution of *mre* than the single value measures. It will be shown later that the WMQ presents a good average of *mre* in evaluating accuracy of software cost estimation.
The WMQ is consistent with the MRE provided that the estimation is not obviously biased (tending to under/overestimate) and there are no outliers (extreme large values).
In a sample of *n* projects an estimation model results in *n* relative errors, the magnitude of which is expressed as a variable *mre*. These *mre*'s are arranged in ascending order. The *mre* can be expressed as a function of the order number *N*, i.e.

$$mre = f(N)$$

With this equation a smoothed curve ($OQ_1Q_2Q_3Q_4$) can be drawn. This curve is illustrated in Figure 1, where Q_1 is the first quartile of *mre* and also expresses the intersection of the curve and the vertical line $N = n_{Q_1}$, where n_{Q_1} (= 0.25*n*) is the ordinal number of Q_1, and so on. Through Point *O* and Point Q_3 a straight line is drawn to intersect with vertical line $N = n$ at Point Q_4'. The vertical line $N = n_{Q_1}$ meets with straight line OQ_4' at Point Q_1'. Point Q_2' is obtained in the same way.

Figure 1 Relationship between WMQ and MRE



If there are no outliers with respect to *mre* we assume that the area $OQ_1Q_2Q_3Q_2'Q_1'$ is approximately equal to the area $Q_3Q_4'Q_4$. Therefore the area A_1 under the smoothed curve is approximately equal to the area A_2 of triangle $OQ_4'n$. Let k be the slope of straight line OQ_4'.

$$MRE = \frac{A_1}{n} \approx \frac{A_2}{n} = \frac{\frac{1}{2}kn^2}{n} = 0.5kn$$

If we assume $Q_1 \approx Q_1'$ and $Q_2 \approx Q_2'$, then

$$WMQ = \frac{Q_1 + 2Q_2 + 3Q_3}{6} \approx \frac{Q_1' + 2Q_2' + 3Q_3}{6} = \frac{0.25kn + 2(0.50kn) + 3(0.75kn)}{6} \approx 0.58kn$$

In this particular case, as $Q_1 < Q_1'$ and $Q_2 < Q_2'$ in Figure 1, WMQ < $0.58kn$. In the situation where there is a tendency to underestimate, the front part of the curve goes up and the rear part comes down. Hence, Q_1 is close to Q_1' and Q_2 is close to Q_2'. WMQ may be greater than $0.5kn$. That is WMQ > MRE (the MRE favours underestimation). For an unbiased estimation, Q_1 and Q_2 are relatively further from Q_1' and Q_2' respectively. WMQ may be closer to $0.5kn$. That is WMQ ≈ MRE. This explains why WMQ and MRE are consistent when there are no outliers and the estimation is unbiased.

MEASURES OF CONSISTENCY

A model that is sensitive to the influence of various productivity factors may nonetheless consistently overestimate or underestimate development, if the standard productivity rate assumed by the model is significantly different from that of the environment in which the software was developed (Mukhopadhyay *et al*, 1992). Models developed in different environments do not work very well without calibration. A consistently overestimating or underestimating model is easier to calibrate than an inconsistent one. Therefore, besides accuracy, consistency is another important feature for an estimation model.

Correlation Coefficient Of Estimates And Actual Values (SDR)

To measure the level of consistency, some researchers have used the correlation coefficient, SDR, between observed and estimated values (Mukhopadhyay *et al*, 1992). This measure tests the linear association between the actual values and estimates. For a highly consistent model, R should be close to 1 ($-1 \leq R \leq 1$), otherwise it is

close to 0. If R is negative, it indicates that larger actual values are associated with smaller estimates. R is the square root of R^2, the coefficient of determination introduced earlier. So SDR is not consistent with the ratio measure which has been illustrated to be more suitable for software cost estimation. Moreover, as R^2, SDR is influenced by the variance of data. The greater the variance of actual values, the larger the denominator in the expression. R varies according to not only the estimation accuracy but also the variance of data. We need a consistency measure, which assesses the estimation on the basis of ratio measure and is independent to the distribution of the actual observations.

A New Measure Of Consistency

Suppose that a model was developed in environment A and a set of data, which was collected from software projects developed in environment B, is used to test the estimation consistency of the model. For each estimation there is a ratio

$$r_i = \frac{est_i}{act_i} \qquad (i = 1,2,3,...,n)$$

In the case of consistent estimation, the values of r_i (i = 1,2,3,...,n) are close to one another. On the other hand, if the values of r_i spread over a wide range, the estimation is not consistent. The closer to one another the values of r_i are, the more consistent the estimation is. Statistically, standard deviation is a measure of variation or spread of the r_i's. So it is proposed to use the standard deviation of r_i (SDR) as a measure of estimation consistency,

$$SDR = \sqrt{\frac{\sum_{i=1}^{n}(r_i - \bar{r})^2}{n-1}}$$

where $\bar{r}$ is the mean of r_i's (i=1,2,3,...,n) The smaller the SDR, the more consistent the estimation.
It can be shown that standard deviation of relative error is equal to SDR. Because the assessment of estimation accuracy is based on relative error, SDR is related to estimation accuracy. Therefore SDR can be used to calibrate a model in order to improve the estimation accuracy in different environments.

REGRESSION CRITERIA

In a previous section, it is argued that ratio measures are more suitable for accuracy assessment. We now consider how to determine model parameters so as to satisfy the ratio measures of accuracy.
In modeling costing formulas, regression is the basic technique to determine model parameters. The traditional method employed by most statistical software is least squares (LS) (Khoshgoftaar *et al*, 1992), which can be expressed as

$$\min[\sum_{i=1}^{n}(act_i - est_i)^2]$$

This is a criterion of difference measure. This criterion cannot lead to an optimised functional expression of the model as ratio measure should be used to assess the prediction accuracy.
In modelling formulas for software cost estimation, the criterion of minimising the sum of squared relative errors

$$\min[\sum(\frac{act_i - est_i}{act_i})^2]$$

has been used (Conte *et al*, 1986). This criterion is not suitable either. In the expression of *mre*

$$mre_i = \left|\frac{act_i - est_i}{act_i}\right| = \left|1 - \frac{est_i}{act_i}\right| \qquad (i = 1,2,3,...,n)$$

act_i is positive for all i, and est_i should be positive for all i. In the case of overestimation, that is $est_i > act_i$, an *mre* can be greater than 1, but in the case of underestimation, that is $est_i < act_i$, an *mre* can never be greater than 1. This explains the problem discussed in a previous section that MRE favours underestimation if it is used as a measure of accuracy. To achieve the least sum of squared relative errors, the regression technique makes as many negative relative errors as possible. Therefore, if least sum of squared relative errors is used as the regression criterion, the obtained formula will be one that systematically underestimates the effort. It has been found in practice that in most cases of software cost estimation, the errors came from underestimation (Lederer *et al*, 1993). Using least sum of relative errors as the regression criterion would make the situation even more serious.

A New Regression Criterion

From the above discussion we can argue that there are two special requirements for regression criterion in formulating software costing models.

- It should be in the form of the ratio of est_i to act_i.
- It should be an unbiased one, which results in neither systematic underestimation nor systematic overestimation.

To meet these two special demands, a new regression criterion in formulating software costing models is proposed:

$$\min[\sum_{i=1}^{n}(\ln\frac{est_i}{act_i})^2] = \min[\sum_{i=1}^{n}(\ln\frac{act_i}{est_i})^2]$$

It is obvious that this criterion takes the ratio form of est_i to act_i. The closer $(\ln\frac{est_i}{act_i})^2$ is to 0, the closer $\frac{est_i}{act_i}$ is to 1, and the more accurate the estimation. As shown in the above expression, est_i and act_i are symmetric with respect to their positions in the expression. Therefore, this criterion does not produce an estimation formula of either systematic underestimation or systematic overestimation.

Computational Procedure With The New Regression Criterion

We can make use of existing statistics software in performing non-linear regression with the proposed criterion. In statistics software, the criterion for finding the optimal parameters is LS

$$\min[\sum_{i=1}^{n}(act_i - est_i)^2]$$

Before the non-linear regression is undertaken, the observed values and the functional form of the model can be transferred into their logarithmic counterparts, act_i' and est_i' .The LS can be expressed as

$$\min[\sum_{i=1}^{n}(act_i' - est_i')^2] = \min\{\sum_{i=1}^{n}[\ln(act_i) - \ln(est_i)]^2\} = \min[\sum_{i=1}^{n}(\ln\frac{est_i}{act_i})^2]$$

For example, we have a set of n pairs of data ($size_i$, $effort_i$) (i = 1,2,3,...,n). If the functional form of a model is

$$Effort = a(Size) + b$$

we can take the logarithmic form for the both sides of the equation to obtain

$$\ln(Effort) = \ln[a(Size) + b].$$

The regression is to be performed according to this equation. In this case, the i^{th} estimate is $est_i = a(size_i) + b$ and the actual value is $act_i = effort_i$. Before the regression, we first calculate $\ln(effort_i)$ (i= 1,2,3,...,n), which will be used as the values for the dependent variable for the regression. In the regression, the functional form is $\ln[a(Size) + b]$. Therefore we have the criterion

$$\min \{ \sum_{i=1}^{n} [\ln \frac{a(size_i + b)}{effort_i}]^2 \}$$

ASSESSMENT OF THE PROPOSED MEASURES AND CRITERION

A set of data quoted by Desharnais (1988) from 81 software projects was used to assess these measures of estimation accuracy and consistency, and regression criteria. The main features of the data set are summarised below.

Effort range:	546-19,894 person-hours
Function point range:	62-1,116 FP
Team average experience:	0-4 years
Manager experience:	0-7 years
Language levels:	*Level 1*: 46 projects were developed in 3GL (COBOL with IMS or IDMS type databases). *Level 2*: 25 projects were developed using a combination of the traditional 3GL approach together with screen and report generators. *Level 3*: 10 projects were implemented in 4GL.

Integrated Software Cost Model

An integrated software cost model was proposed by Gao and Lo (1995) on the basis of COCOMO (Boehm, 1981) and FPA. The integrated model attempts to combine the advantages of these two approaches into a single model. To address the problem of language dependency, language-weighted function point is introduced. In addition, continuous, rather than discrete, "cost drivers" are used. The general form of the model is

$$Effort = a(WFP)^b \prod_{i=1}^{n} c_i^{(x_i - d_i)}$$

where a and b are constants, x_i is the magnitude of the i^{th} cost driver, c_i and d_i are constants corresponding to the i^{th} cost driver, n is the number of cost drivers and WFP is language-weighted FP.
The equation appropriate to Desharnairs' data set is

$$Effort = a(l_k FP)^b.$$

where l_1 = 1.00, l_2 = 0.95 and l_3 = 0.20 are the weights for language level 1, 2, and 3 respectively (Gao *et al*, 1995). The exponent b is the responsiveness of Effort to FP. It indicates that *Effort* increases by b% when FP increases by 1%.

Assessment On Regression Criteria

Three criteria for finding optimal parameters are used in regression:

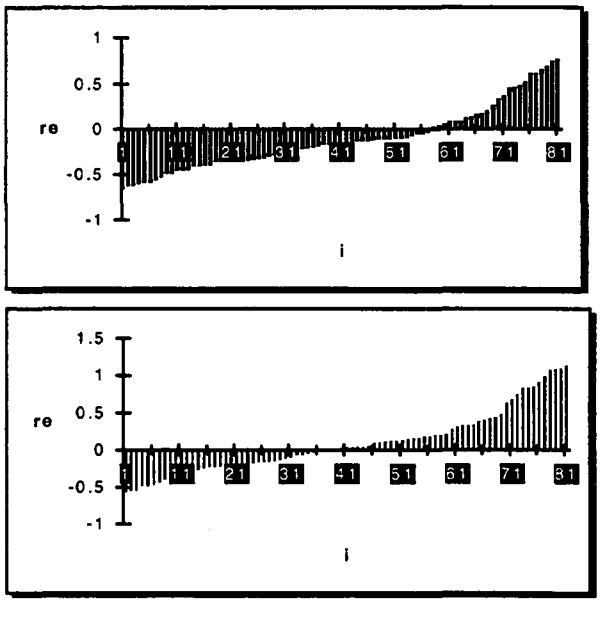
- Criterion 1: $\min[\sum_{i=1}^{n}(act_i - est_i)^2]$
- Criterion 2: $\min[\sum_{i=1}^{n}(\frac{act_i - est_i}{act_i})^2]$ and
- Criterion 3: $\min[\sum_{i=1}^{n}(\ln\frac{est_i}{act_i})^2]$

In Figure 2 and Figure 3, the relative errors

$$re_i = \frac{est_i - act_i}{act_i} \qquad (i = 1,2,3,...,81)$$

are shown in an ascending order. Figure 2 shows the relative errors with respect to Criterion 2. It can be seen that more than 70% of the relative errors are negative. This indicates that the formula obtained from Criterion 2 tends towards underestimation. Figure 3 shows the relative errors with respect to Criterion 3. The number of negative relative errors is approximately the same as that of positive relative errors. The formula obtained from Criterion 3 does not result in systematic bias. (Criterion 1 is similar to Criterion 3 with respect to estimation bias, because the act_i and est_i can interchange in the expression of Criterion 1.) This shows that the regression with Criterion 3 overcomes the shortcoming of the tendency to underestimate which occurred with Criterion 2.

Figure 2 Relative errors (Criterion 2) **Figure 3 Relative errors (Criterion 3)**



Assessment On The Measures Of Accuracy And Consistency

For comparison, Q_3, PRED(0.25), MRE as well as WMQ are computed as measures of prediction accuracy, while SDR and SDR are computed as measures of estimation consistency. We also use

$$Effort = a(FP)^b$$

for regression in order to fully assess the measures of accuracy.

Accuracy Assessment

Table 1 uses FP as size measure while Table 2 uses WFP. Table 2 also includes consistency results, which are to be compared with Table 3 in the next section.

Table 1 Accuracy of FP as size measure

Criterion	Q_3	PRED(0.25)	WMQ	MRE
1	0.66	0.42	0.45	0.67
2	0.70	0.11	0.62	0.58
3	0.56	0.37	0.42	0.58

Table 2 Accuracy and consistency of WFP as size measure

Criterion	Q_3	PRED(0.25)	WMQ	MRE	R	SDR
1	0.43	0.49	0.32	0.33	0.8428	0.439
2	0.45	0.48	0.34	0.30	0.8389	0.344
3	0.44	0.56	0.31	0.32	0.8406	0.423

The first observation is that the compared measures of accuracy, Q_3, PRED(0.25), WMQ and MRE, are not consistent in both tables. This observation is not unexpected because the Q_3 and PRED(0.25) are stochastic. The difference between MRE and WMQ will be discussed later.

The second observation is that Criterion 2 leads to the best MRE and Criterion 1 leads to the worst MRE. As discussed in the previous section, Criterion 1 is based on difference measure while MRE is a ratio measure. Therefore the MRE of Criterion 1 is larger than that of Criterion 2 and Criterion 3, which are based on relative errors. Criterion 2 leads to underestimation tendency and MRE favours underestimation. Therefore Criterion 2 has the best MRE. Criterion 3 does not have over- or underestimation tendency. Therefore the MRE of Criterion 3 is located between those of Criterion 2 and Criterion 1.

The third observation is that if there are no outliers and no underestimation tendency (Criterions 1 and 3 in Table 2), WMQ $\approx$ MRE. If there are outliers (Criterion 1 and Criterion 3 in Table 1), WMQ < MRE, and if underestimation tendency exists (Criterion 2 in Table 1), WMQ > MRE. As mentioned before, MRE as a measure of accuracy has two drawbacks. It is influenced by outliers and it favours underestimation. WMQ as proposed targets the solution of these two difficulties, and this third observation indicates that WMQ has succeeded in its objective.

The fourth observation is that the Criterion 3 leads to the best WMQ for all data conditions. As assessed above, WMQ is a better measure of accuracy, so Criterion 3 is superior to the other two criteria.

Consistency Assessment

In this section, it will be first shown that SDR is better than SDR, because R is influenced by not only the estimation result but also the variance of the actual effort, whereas SDR is not influenced by the later. For this purpose, the data of one project, which involves the largest development effort, is deleted from the data set. With

this change the variance of the actual effort changes. Table 3 shows the accuracy and consistency results from the reduced data set.

Table 3 Accuracy and consistency results from reduced data set

Criterion	Q_3	PRED(0.25)	WMQ	MRE	R	SDR
1	0.44	0.49	0.33	0.34	0.7900	0.441
2	0.45	0.49	0.33	0.30	0.7885	0.345
3	0.44	0.56	0.31	0.32	0.7891	0.425

Comparing Table 2 and Table 3, we find that the Q values, WMQ and MRE of each Criterion are virtually the same in the two corresponding situations. The consistency should not be different because it measures the quality of a model in a specific environment and it should be independent of the variance of the actual effort. However, the change of variance makes the R values decrease by more than 0.0500. On the other hand, the SDR values only change by less than 0.002, which is comparatively small. Therefore, the SDR is superior over R in the aspect that the former is not influenced by the variance in effort.

Next, it will be demonstrated that a model producing consistent estimation errors is more easily calibrated to improve accuracy than is a model producing less consistent errors, if the consistency is measured by SDR. The calibration method is described as follows.

With Desharnais' data set, three formulas are obtained by regression with the three criteria described earlier. These formulas are to be calibrated to the environment of Albrecht's data set (1983). In Albrecht's data set, 15 projects used level 1 language and their FPs fell in the size range of Desharnais' data set. For the other projects, either the languages cannot be categorised into the 3 levels or the FPs are beyond the size range of Desharnais' data set. Therefore only these 15 projects were used for the calibration.

It is assumed that the responsiveness of effort to size (exponents in the formulas, refer to section of "Integrated Software Cost Model") does not change, but the environments of the two sets of data are different. In Desharnais' environment, the formulas obtained with the three regression criteria are

$$Effort = a_j (WFP)^{b_j} \quad (j = 1,2,3)$$

The environment is the overall level of cost drivers (Gao *et al*, 1995), which is expressed by the coefficient a_j (j = 1,2,3). For Albrecht's data set, an environment adjustment parameter, k_j (j = 1,2,3), is used in the above formulas to account for the environment change

$$Effort = k_j a_j (WFP)^{b_j} \qquad (j = 1,2,3)$$

where the k_j (j = 1,2,3) are determined by the following procedure.

Firstly, the estimated efforts est_{ij} (i = 1,2,3,...,15, j = 1,2,3) are obtained with the formulas developed in Desharnais' environment. Secondly, it is assumed that the k_j (j = 1,2,3) satisfy the following equations

$$act_i = k_j\, est_{ij} \text{ or } 1 = k_j \frac{est_{ij}}{act_i} \qquad (i = 1,2,3,...,15, j = 1,2,3)$$

where act_i (i = 1,2,3,...,15) are the actual values of effort in Albrecht's data set. Thirdly, these 15 equations are summed to give

$$15 = \sum_{i=1}^{15} k_j \frac{est_{ij}}{act_i} \qquad (j = 1,2,3)$$

and the above expression is rearranged so that the k_j (j = 1,2,3) are obtained

$$k_j = 15 / \sum_{i=1}^{15} \frac{est_{ij}}{act_i} \qquad (j = 1,2,3)$$

Table 4 and Table 5 show the results of accuracy and consistency before and after the calibration.

Table 4 Results before calibration

Criterion	Q_1	Q_2	Q_3	PRED(0.25)	WMQ	MRE	R	SDR
1	0.10	0.18	0.49	0.60	0.32	0.36	0.5328	0.57
2	0.21	0.35	0.57	0.27	0.44	0.42	0.5361	0.42
3	0.11	0.26	0.47	0.47	0.34	0.36	0.5352	0.53

Table 5 Results after calibration

Criterion	Q_1	Q_2	Q_3	PRED(0.25)	WMQ	MRE	R	SDR
1	0.10	0.17	0.50	0.60 0.32	0.36	0.5328	0.58	
2	0.09	0.23	0.46	0.60 0.32	0.36	0.5361	0.57	
3	0.10	0.20	0.47	0.60 0.32	0.36	0.5352	0.57	

For Criterion 1 in Table 4, the SDR is 0.57 (the largest SDR in this table), so the estimation is the least consistent. Criterion 1 in Table 5 shows that the calibration does not improve the accuracy. For Criterion 3 in Table 4, the SDR is 0.53. This consistency is better than that of Criterion 1. The calibration improves the WMQ from 0.34 in Table 4 to 0.32 in Table 5, while MRE remains unchanged. For Criterion 2 in Table 4, the SDR is 0.42. This consistency is much better than that of Criterion 1 or Criterion 3. The calibration improves the WMQ from 0.44 in Table 4 to 0.32 in Table 5, and the MRE from 0.42 to 0.36. The results indicate that the smaller the SDR is, the easier it is to improve accuracy by calibration.

The SDR values remain the same before and after the calibration. The linear relationship between estimated effort and actual effort does not change after calibration, because the calibration multiplies the uncalibrated estimated effort by a constant,. On the other hand, all the SDR values after calibration differ from those before calibration. This indicates that SDR and SDR are not the same. The SDR measures the linear relationship between estimated value and actual value while SDR measures the variation of relative errors.

The accuracy results after calibration are comparable with those obtained when the original formulas are used with Desharnais' data set. This observation further emphasises that a model needs to be calibrated if it is to be used in a different environment.

From Table 4 and Table 5 it can also be observed that MREs are greater than WMQs in all the cases except in Criterion 2 of Table 4. As expected, there are extreme values in cases where MRE > WMQ, and the formula tends to underestimate the effort in Criterion 2 of Table 4, where MRE < WMQ.

SUMMARY

In order to evaluate estimation models and improve the modelling process, we need appropriate methods to measure these models and determine model parameters. This paper reviewed existing methods and proposed new methods to measure estimation accuracy and consistency, and to determine model parameters.

The difference measures of accuracy favour the estimation for small projects. Therefore it is argued in this paper that measures of accuracy should be based on relative error of estimation. As the MRE (the most widely used measure of accuracy) is influenced by outliers and favours underestimation, and single value measures are stochastic, the WMQ is proposed for accuracy evaluation. The WMQ includes more information on the estimation than single value measures, so it is less stochastic. It is also consistent with MRE when there are no outliers and estimation is unbiased.

Consistency examines the model's degree of ease of calibration. A consistently overestimating or underestimating model is more easily calibrated than an inconsistent one. The correlation coefficient R between observed and actual values has been used to evaluate consistency. The R favours a data set with large variance. It is determined not only by estimation but also by the distribution of the actual values. In this paper, the standard deviation of the ratios of the estimate to actual effort (SDR) is proposed as a measure of consistency. The SDR is a measure of the variation or spread of the relative error.

The method of least squares is the conventional regression criterion, but it is based on a difference measure, which is not suitable for software cost estimation. The criterion of minimising the MRE will produce a formula which tends to underestimate the effort. To overcome the shortcomings of these criteria, this paper proposes to use the criterion of least squares of the logarithmic ratio of estimate to actual value for regression.

The proposed measures and criterion share the characteristic that they are based on relative errors of the estimation.

Applied to real-world data, the proposed measures and criterion appear superior to the measures and criteria that are currently used in the field and reported in the literature. It is therefore recommended that the proposed measures and criterion be used for further assessment

REFERENCES

Abdel-Hamid, T.K. (1990) "On the utility of historical project statistics for cost and schedule estimation: results from a simulation-based case study", **Journal of Systems & Software**, Sep., Vol.13, No.1, pp.71-82.

Albrecht, A.J. & Gaffney, J. (1983) "Software function, source lines of code, and development effort prediction: a software science validation", **IEEE Transactions on Software Engineering**, Nov., Vol.9, No.6, pp.639-648.

Boehm, B.W.(1981) **Software Engineering Economics**, Prentice-Hall, Inc., Englewood Cliffs, New Jersey,.

Conte, S.D., Dunsmore, H.E. & Shen, V.Y. (1986) **Software Engineering Metrics and Models**, Benjamin/Cummings Publishing Company, Inc., Menlo Park, Calif..

Deeson, E. (1991) **Collins Dictionary of Information Technology**, Harper Collins Publishers, Glasgow.

Desharnais, J-M. (1988) "Programme de maîtrise en informatique de gestion", University of Montreal, Canada.

Fenton, N. (1994) "Software measurement: a necessary scientific basis", **IEEE Transactions on Software Engineering**, Mar., Vol.20, No.3, pp.199-206.

Gao, X. & Lo, B. (1995) "An integrated software cost model based on COCOMO and function point analysis", **Proceedings of the Software Education Conference (SRIG-ET94)**, pp.86-93, IEEE Computer Society, Los Alamitos, CA.

Jenson, R.L. & Bartley, J.W. (1991) "Parametric estimation of programming effort: an object-oriented model", **Journal of Systems & Software**, May, Vol.15, No.2, pp.107-114.

Jørgensen, M. (1994) "A comparative study of software maintenance effort prediction models", **Proceedings of 5th Australian Conference on Information Systems**, 27-29 September 1994, Melbourne, Australia, pp.699-709.

Kan, S.H., Basili, V.R. & Shapiro, L.N. (1994) "Software quality: an overview from the perspective of total quality management", **IBM Systems Journal**, Mar, Vol.33, No.1, pp.4-19.

Kenkel, J.L. (1989) **Introductory Statistics for Management and Economics (Third Edition)**, Duxbury Press, USA.

Khoshgoftaar, T.M., J.C. Munson, *et al* (1992) "Predictive modeling techniques of software quality from software measures", **IEEE Transactions on Software Engineering**, Nov., Vol.18, No.11, pp.979-987.

Kitchenham, B.A., Kok, P.A.M. & Kirakowski, J. (1990) "The MERMAID approach to software cost estimation", **Proceedings of the Annual ESPRIT Conference**, Brussels, November 12-15, 1990, pp.296-314.

Lederer, A.L. & Prasad, J. (1993) "Information systems software cost estimating: a current assessment", **Journal of Information Technology**, No.8, pp.22-33.

Miyazaki, Y. Takanou, A. & Nozaki, H. (1991) "Method to estimate parameter values in software prediction models", **Information & Software Technology**, Apr., Vol.33, No.3, pp.239-243.

Mukhopadhyay, T., Vicinanza, S.S. & Prietula, M.J. (1992) "Examining the feasibility of a case-based reasoning model for software effort estimation", **MIS Quarterly**, Jun., pp.155-170.

Pfleeger, S.L. (1991) "Model of software effort and productivity", **Information & Software Technology**, Apr., Vol.33, No.3, pp.224-231.

Rowlands, B.H. (1989) "Estimating instructional systems development effort and costs", **Proceedings of the 7th Annual Conf. of the Australian Society for Computers in Learning in Tertiary Education**, Gold Coast, 1989, pp.340-348.

Wrigley, C.D. & Dexter, A.S. (1991) "A model for measuring information system size", **MIS Quarterly**, Jun., pp.245-257.

Verner, J. & Tate, G. (1992) "A software size model", **IEEE Transactions on Software Engineering**, Apr., Vol.18, No.4, pp.265-278.