UNIVERSITY
of
OTAGO

SAPERE AUDE

Te Whare Wānanga o Otāgo

An Effort Prediction Model for Data-Centred Fourth-Generation-Language Software Development

(Outstanding Honours student paper, 2003)

C. van Koten

The Information Science Discussion Paper Series

Number 2003/04
November 2003
ISSN 1177-455X

University of Otago

Department of Information Science

The Department of Information Science is one of seven departments that make up the School of Business at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in postgraduate research programmes leading to MCom, MA, MSc and PhD degrees. Research projects in spatial information processing, connectionist-based information systems, software engineering and software development, information engineering and database, software metrics, distributed information systems, multimedia information systems and information systems security are particularly well supported.

The views expressed in this paper are not necessarily those of the department as a whole. The accuracy of the information presented in this paper is the sole responsibility of the authors.

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper, or for subsequent publication details. Please write directly to the authors at the address provided below. (Details of final journal/conference publication venues for these papers are also provided on the Department's publications web pages: http://www.otago.ac.nz/informationscience/pubs/). Any other correspondence concerning the Series should be sent to the DPS Coordinator.

Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND

Fax: +64 3 479 8311
email: dps@infoscience.otago.ac.nz
www: http://www.otago.ac.nz/informationscience/

An Effort Prediction Model for Data-Centred Fourth-Generation-Language Software Development

Chikako van Koten

Department of Information Science
University of Otago
PO Box 56, Dunedin, New Zealand

vanch435@student.otago.ac.nz

Abstract

Accurate effort prediction is often an important factor for successful software development. However, the diversity of software development tools observed today has resulted in a situation where existing effort prediction models' applicability appears to be limited. Data-centred fourth-generation-language (4GL) software development provides one such difficulty. This paper aims to construct an accurate effort prediction model for data-centred 4GL development where a specific tool suite is used. Using historical data collected from 17 systems developed in the target environment, several linear regression models are constructed and evaluated in terms of two commonly used prediction accuracy measures, namely the mean magnitude of relative error (MMRE) and pred measures. In addition, R^2, the maximum value of MRE, and statistics of the absolute residuals are used for comparing the models. The results show that models consisting of specification-based software size metrics, which were derived from Entity Relationship Diagrams (ERDs) and Function Hierarchy Diagrams (FHDs), achieve good prediction accuracy in the target environment. The models' good effort prediction ability is particularly beneficial because specification-based metrics usually become available at an early stage of development. This paper also investigates the effect of developers' productivity on effort prediction and has found that inclusion of productivity improves the models' prediction accuracy further. However, additional studies will be required in order to establish the best productivity inclusive effort prediction model.

Keywords: Prediction systems, 4GL, effort, metrics, empirical analysis

1 Introduction

Accurate effort prediction at an early stage is often an important factor for successful software development. In order to predict software development effort, there exist a number of effort prediction models, including well-known COCOMO (Boehm 1981,1984) and Function Points Analysis (FPA) (Albrecht and Gaffney JR. 1983). However, these existing models are, in general, empirical models constructed using historical data collected from a number of software systems developed in specific development environments. As a consequence, the applicability of these models is often limited to systems developed in those environments.

On the other hand, the increasing number of software development tools available today enables software systems to be developed in very different environments. Some organizations use a data-centred fourth-generation-language (4GL) software development tool. Data-centred 4GL software development tools enable database-oriented transaction processing systems (TPSs) and/or management information systems (MISs) to be developed in a rapid manner. However, a number of studies have showed that traditional effort prediction models are not able to predict development effort accurately when a data-centred 4GL software development tool is used (Kemerer 1987, Verner and Tate 1988, Dolado 1997).

The situation described above has prompted researchers to construct new effort prediction models for data-centred 4GL software development (Tate and Verner 1990, 1991, Wrigley and Dexter 1991, Verner and Tate 1992, MacDonell 1997, MacDonell, Shepperd, and Sallis 1997, Dolado 2000). These effort prediction models are often linear regression models which consist of software size metrics collected in environments where a specific data-centred 4GL development tool was used. The software size metrics chosen for each of these models are different but all specification-based, that is, derived from a software system's specifications such as Entity Relationship Diagrams (ERDs) and Function Hierarchy Diagrams (FHDs). These models achieved good prediction accuracy in terms of mean magnitude of relative error (MMRE) and a measure called pred. Both MMRE and pred are commonly used prediction accuracy measures among researchers (Fenton and Pfleeger 1997, Shepperd, Cartwright, and Kadoda 2000).

However, due to the very same reason as mentioned previously about empirical effort prediction models in general, the applicability of those new effort prediction models is limited to a specific development environment in each case. Consequently, it is necessary to construct a new effort prediction model when a different development tool is used. The organization being studied in this paper started using a data-centred 4GL software development tool suite, *Oracle*'s *Designer 6i* and *Developer 6i*, in 2002. This created the need for a new effort prediction model and raised the following research

This paper was submitted to the *6th Australasian Computing Education Conference (ACE2004)*, Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, Vol. 4. J. Weckert, Ed.

question. How can an accurate development effort prediction model be constructed for this specific data-centred 4GL software development environment?

In order to answer the above question, a study is carried out with three objectives. The first is to identify a useful method for constructing empirical effort prediction models in the target environment. The second is to construct the models. The third is to evaluate and compare the models' prediction accuracy in terms of some commonly used prediction accuracy measures so that the most appropriate model(s) can be identified. Considering the good prediction accuracy achieved by other data-centred 4GL development effort prediction models, the study is based on a hypothesis that an accurate effort prediction model should be able to be constructed empirically as a linear regression model consisting of a number of specification-based software size metrics collected from systems developed in the target environment. This paper presents the preliminary results of the study.

The structure of the reminder of this paper is as follows. Section 2 describes a set of effort prediction models constructed for the target data-centred 4GL software development environment and the modelling procedure used for their construction. Section 3 evaluates and compares these models' prediction accuracy using some commonly used measures. This is followed by Section 4, where conclusions are presented.

2 Effort Prediction Models

2.1 Empirical Data Collection

2.1.1 Metrics Selection

The first step is to select appropriate candidate metrics. A useful approach for selecting metrics is the Goal/Question/Metric (GQM) paradigm (Basili and Weiss 1984, Basili and Rombach 1988, Oivo and Basili 1992). In the GQM paradigm, users start with a set of goals, proceed to a set of related questions, and end with a set of appropriate metrics. This process guides users to minimize the number of metrics collected by selecting only those relevant to satisfying the specified goals. In this study, the GQM process was modified to select candidate metrics for the models. The process is shown in Fig.1 and resulting metrics are shown in Table 1. During this process, the following aspects of the target development environment were taken account of:

1. The minimal set of specifications completed for a software system is the ERD and FHD. The FHD defines functional specifications of the system's user interface components.
2. The system's database is automatically generated from the ERD. This implies that differences in the complexity of ERDs, such as the differences of the numbers of different types of relationships, would have, in general, only a negligible effect on development effort because implementation of the data model does not require any manual coding.

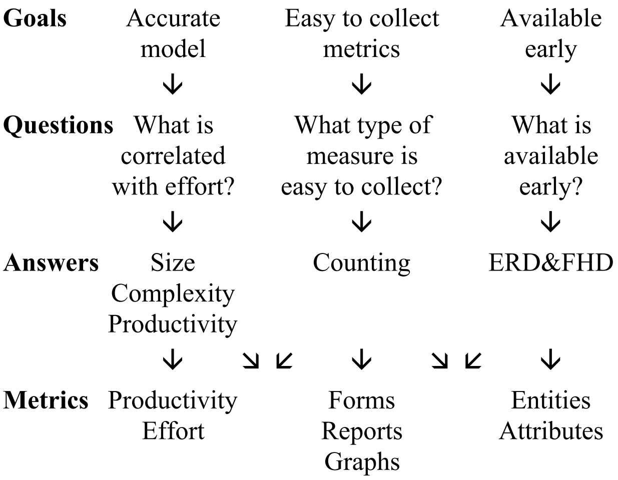

Fig. 1: Modified GQM Approach for Metrics Selection

Metrics	Definitions
EntityNum	Number of entities in the ERD
AttributeNum	Number of attributes in the ERD (The same attribute is counted only once.)
FormNum	Number of forms in the FHD
ReportNum	Number of reports in the FHD
GraphNum	Number of graphs in the FHD
EntityForm	Total number of entities accessed by all forms (The same entity is counted more than once if required.)
EntityReport	Total number of entities accessed by all reports (The same entity is counted more than once if required.)
EntityGraph	Total number of entities accessed by all graphs (The same entity is counted more than once if required.)
Effort	Total hours spent by development team
Productivity	Average mark awarded to development team from a practical development test
TotalForm ReportGraph	Sum of FormNum, ReportNum and GraphNum
EntityForm ReportGraph	Sum of EntityForm, EntityReport and EntityGraph

Table 1: Candidate Software Metrics Definitions

3. A system consists of some or all of three types of user interface components – forms, reports, and graphs. Once the related table(s) in the database is defined, the tool automatically generates code that ensures the connectivity between them. This implies that developers'

effort would be primarily spent performing two tasks: creating each component by using various ready-made graphical user interface items such as text boxes and combo boxes, and adding code to the items.

This modified GQM approach is different from the original GQM process, as this approach uses answers to the questions to determine metrics to be collected, instead of using the questions to determine metrics for the answers.

2.1.2 Software Systems

The next step is to identify software systems from which the metrics data are to be collected. In this step, a total of 19 small or medium sized TPSs and/or MISs developed in the target environment were identified. These systems were developed during a period of 13 weeks in 2002 as part of a university course. Each system was developed by a group of four developers (and in one case, a group of three), who were final year undergraduate students taking computer and/or information science as a major. These systems were developed for clients outside the university.

2.1.3 Data Collection Procedure

The next step is to collect metrics data from the systems. In the target environment, all software systems developed were accompanied by final documentation. This documentation contained an ERD and FHD for the final system and effort data recorded by each developer in the group. All forms, reports and graphs in the final system and the database were also stored in a repository provided by the development tool suite. The required metrics data were collected from those sources. During this process, two systems were eliminated due to the incomplete effort data. This resulted in the remaining 17 systems being used in this study, all of which have the same number of developers. The productivity metric was calculated by using the average mark of the developers in each team from a practical development test undertaken in the target environment.

2.2 Data Analysis

2.2.1 Descriptive Statistics

The next step is to analyse the collected metrics data. Descriptive statistics of the data are shown in Table 2. The effort data are measured in hours. The differences observed between the medians and means, and the values of the skewness statistic in this table show that the data are skewed. Therefore, in the following exploratory analysis, non-parametric techniques, which do not require the normality assumption, are used.

2.2.2 Correlation Analysis

In order to examine the existence of the potential linear relationships between the specification-based software size metrics and development effort, and the degree of linear association between the specification-based software size metrics themselves, correlation analysis was performed. Spearman's rank correlation coefficient was used in this analysis. The results are shown in Table 3.

A number of significant correlation coefficients in the first column of Table 3 (a) show that ATTRIBUTENUM, FORMNUM, ENTITYFORM, and ENTITYFORMREPORT-GRAPH are highly correlated with EFFORT. Two negative correlation coefficients appear in the same column, although they are not at a significant level. These do not appear intuitive as they indicate that a larger REPORTNUM and ENTITYREPORT require less effort. A possible explanation of these counter-intuitive coefficients is as follows. Due to the limited amount of time allowed for development, it seems that teams with similar productivity resulted in developing systems whose TOTALFORMREPORTGRAPH, that is, the sum of FORMNUM, REPORTNUM and GRAPHNUM, is similar. In addition, the limited development time resulted in systems with more forms having less reports, as forms are, in general, created before reports in order to allow entering sample data into the system. Consequently, the combination of these two factors create the unique interaction between the number of forms and that of reports in a system, where increasing the number of reports often leads to decreasing the number of forms. This presumption is supported by the significant negative correlation coefficient between FORMNUM and REPORTNUM shown in Table 3 (a). On the other hand, in the target environment, developing a form, in general, requires more effort than developing a report. This is because forms require more manual coding in order to process input data, for example, to implement validation rules; to display error, warning and confirmation message boxes/dialogs; and to perform various calculations. Given the unique interaction between forms and reports, and the anticipated smaller contribution of reports to development effort than forms in general, it would be possible for REPORTNUM and ENTITYREPORT to have a small negative correlation with EFFORT.

A total of 13 significant correlation coefficients observed between the specification-based software size metrics in Table 3 indicate that these pairs are correlated. When correlated metrics are included in a regression model, multicollinearity can cause some difficulty for users in interpreting some partial correlation coefficients and increases the standard errors of the predicted values. Thus, when constructing a multivariate regression model, it is recommended to include software size metrics which are not highly correlated with each other. Principal component analysis (PCA) can be used to construct independent variables using linear transformations of the original input variables. However, PCA is not used in this study as neither the difficulty in interpretation nor the problem of the errors is identified in the models.

2.3 Linear Regression Analysis

2.3.1 Univariate Regression Models

The final step is to perform linear regression analysis in order to construct a number of candidate effort prediction models for the target environment. The significant correlation coefficients between some specification-based software size metrics and EFFORT shown in the first column of Table 3 suggest that some or all of univariate linear regression models consisting one of these specification-based software size metrics may be able to predict development effort accurately for the target environment. The specification-based size metrics with the highest correlation coefficient with EFFORT, ENTITYFORM was chosen to construct the following univariate linear regression model:

$$\text{EFFORT} = 212.171 + 5.643\ \text{ENTITYFORM} \tag{2.1}$$

This model's R^2 value was 0.550, indicating that this model can explain 55.0% of the variance in EFFORT.

Another univariate effort prediction model of particular interest was the model consisting of FORMNUM. This is because FORMNUM has the second highest correlation coefficient with effort, and may be more useful than ENTITYFORM as FORMNUM requires less time and effort to collect. The model was constructed as:

$$\text{EFFORT} = 218.298 + 14.347\text{FORMNUM} \tag{2.2}$$

This model achieved an R^2 value of 0.364.

2.3.2 Multivariate Regression Models

Although univariate models are the simplest, and in most cases, the easiest to collect data for, they may not necessarily achieve good effort prediction accuracy for the target environment. One possible way to achieve better prediction accuracy is constructing multivariate regression models which consist of a larger number of influential variables. There are a number of methods to select influential variables in multivariate regression. In this paper, two commonly used methods, stepwise selection and backward elimination, were used.

Stepwise selection was performed using an entry criterion of 0.05 for the F-statistic's p-value and a removal criterion of 0.10. This resulted in the same ENTITYFORM univariate model as shown in Equation 2.1.

Backward elimination starting with all specification-based software size metrics and using a removal criterion of 0.10, resulted in the following multivariate model:

$$\begin{aligned}\text{EFFORT} = 119.560 &+ 8.954\ \text{FORMNUM}\\ &+ 4.695\ \text{ENTITYFORM}\\ &+ 0.738\ \text{ENTITYREPORT}\\ &\square\ 5.023\ \text{ENTITYGRAPH}\end{aligned} \tag{2.3}$$

This model's adjusted R^2 value was 0.569.

Other multivariate models of particular interest were the model consisting of FORMNUM, REPORTNUM and GRAPHNUM, and the model consisting of ENTITYFORM, ENTITYREPORT and ENTITYGRAPH, as each of these models took account of combining the influence of three different user interface components on effort. However, the model consisting of FORMNUM, REPORTNUM and GRAPHNUM was not significant (p-value for F statistic was 0.052).

The model consisting of ENTITYFORM, ENTITYREPORT and ENTITYGRAPH was:

$$\begin{aligned}\text{EFFORT} = 190.396 &+ 6.040\ \text{ENTITYFORM}\\ &+ 0.530\ \text{ENTITYREPORT}\\ &\square\ 3.365\ \text{ENTITYGRAPH}\end{aligned} \tag{2.4}$$

This model's adjusted R^2 value was 0.486.

2.3.3 Influence of Developers' Productivity

The productivity of highly skilled developers is up to 30 times higher than low-skilled developers (Glass 2001). This implies that developers' productivity may be an important factor in development effort prediction, and taking account of the influence may improve models' prediction accuracy. In order to examine this possibility further, the following three approaches were taken:

1. Regression modelling including productivity as an independent variable
2. Regression modelling including effort times productivity (EFFORT□PRODUCTIVITY) as the dependent variable
3. Regression modelling using data from only systems developed by developers whose productivity is considered to be the same

The first approach considers the influence of productivity as a linear adjustment of effort. The second considers the influence as a weighting factor of effort. The third attempts to remove the influence by creating a subset, in which the difference of productivity would be considered to be negligible. All three approaches were taken, as it was anticipated that they would each produce a different model due to the different handling of the influence of productivity on effort.

Regression analyses including developers' productivity as an independent variable were performed using stepwise selection and backward elimination. Stepwise selection produced a bivariate model:

$$\begin{aligned}\text{EFFORT} = 431.\ 808 &+ 5.452\ \text{ENTITYFORM}\\ &\square\ 3.705\ \text{PRODUCTIVITY}\end{aligned} \tag{2.5}$$

This model's adjusted R^2 value was 0.622.

Backward elimination produced the following model:

$$\begin{aligned}\text{EFFORT} = 426.492 &+ 6.761\ \text{ENTITYFORM}\\ &+ 11.619\ \text{REPORTNUM}\\ &\square\ 5.727\ \text{PRODUCTIVITY}\end{aligned} \tag{2.6}$$

This model's adjusted R^2 value was 0.684.

The above two models clearly show that developers' productivity is indeed influential in the target environment as they are both superior to the productivity exclusive models developed earlier in terms of adjusted R^2. The direction of the influence agreed with intuition that higher productivity results in less effort.

Regression analysis including (EFFORT□PRODUCTIVITY) as the dependent variable was also performed using both stepwise selection and backward elimination.

Stepwise selection produced a bivariate model:

$$\begin{aligned}\text{Effort}\square\text{Productivity} &= 4550.14 + 380.90\ \text{EntityForm} \\ &\quad + 815.62\ \text{ReportNum} \qquad (2.7)\end{aligned}$$

This model's adjusted R^2 value was 0.590.

Backward elimination produced the following multivariate model:

$$\begin{aligned}\text{Effort}\square\text{Productivity} &= 6422.07 + 1225.41\ \text{ReportNum} \\ &\quad + 325.33\ \text{EntityForm} \\ &\quad \square\ 130.15\ \text{EntityReport} \\ &\quad +\ 325.33\ \text{EntityGraph} \qquad (2.8)\end{aligned}$$

This model's adjusted R^2 value was 0.516.

In order to select the systems for which developers' productivity were regarded as the same, a fixed range of productivity values between 50% and 70% inclusive were chosen. Based on this productivity range, a subset of 11 systems was selected from the original 17 systems. Descriptive statistics and the results of correlation analysis of this subset are shown in Table 4 and 5. The data are still skewed. Table 5 shows that in the subset EntityFormReportGraph has the highest correlation with Effort, followed by EntityForm and AttributeNum in the order. Surprisingly FormNum which has the second highest correlation with Effort in the original set, did not show a significant correlation in the subset. Another surprising result in Table 6 is that the correlations between the specification-based software size metrics appear to be slightly more complex than those in the original set. These two surprising results may be explained as the non-parametric correlation coefficient being not powerful due to the small number of observations (projects) in the subset, although a further study is required for the confirmation. The counter-intuitive negative correlation of TotalFormReportGraph with Effort in Table 6 is not considered to be an issue as the value is almost negligible.

Based on the subset, two univariate models were produced: one consisting of EntityFormReportGraph, whose correlation with Effort was the highest, and the other consisting of EntityForm, whose correlation was the second highest in the subset and the highest in the original set. These models were:

$$\begin{aligned}\text{Effort} &= 114.393 + \\ &\quad 4.430\ \text{EntityFormReportGraph} \qquad (2.9)\end{aligned}$$

$$\text{Effort} = 236.785 + 4.498\ \text{EntityForm} \qquad (2.10)$$

The model in Equation 2.9 had the R^2 value of 0.406 and the model in Equation 2.10 had 0.536.

Stepwise selection and backward elimination performed with the subset produced the same univariate model as shown in Equation 2.9.

Two multivariate models, one consisting of FormNum, EntityForm, EntityReport and EntityGraph, and the other consisting of EntityForm, EntityReport and EntityGraph, were also produced for comparison with those of the original 17 systems. However, both models only achieved a non-significant p-value for their F statistics (0.159 and 0.069 respectively). Thus, neither was considered as a candidate model for the target environment.

2.3.4 Outlier Detection

Regression models' prediction accuracy increases when influential outliers are removed from the data set. During the above regression analyses, some outlier detection techniques, namely scatterplots of the residuals and statistics such as the deleted residuals, Cook's distance, Mahalanobis distance, were used to locate any potential outliers. The results did not justify removing any project from the data set.

3 Model Evaluation

3.1 Prediction Accuracy Measures

The most commonly used prediction accuracy measures for software effort prediction models among researchers are MMRE and pred (Fenton and Pfleeger 1997, Shepperd, Cartwright, and Kadoda 2000). The magnitude of relative error (MRE) is a normalized measure of the discrepancy between actual values and predicted values (Kitchenham, Pickard, MacDonell and Shepperd 2001):

$$\mathrm{MRE} = \frac{|\,\text{actual effort} - \text{predicted effort}\,|}{\text{actual effort}} \qquad (3.1)$$

MMRE is the mean of MREs over all observations in the data set (Kitchenham, Pickard, MacDonell and Shepperd 2001):

$$\mathrm{MMRE} = \frac{1}{n}\,\square_{i=1}^{n}\ \mathrm{MRE}_i \qquad (3.2)$$

where n is the number of observations in the data set and MRE_i is the MRE of the i-th observation. Pred is a measure of what proportion of the predicted values have MRE less than or equal to a specified value (Fenton and Pfleeger, 1997):

$$\mathrm{Pred}(q) = \frac{k}{n} \qquad (3.3)$$

where q is the specified value, k is the number of observations whose MRE is less than or equal to q, and n is the number of observations in the data set. It is suggested that an acceptable level of MMRE and pred for an effort prediction model being considered to be accurate are MMRE □ 0.25 and pred(0.25) ≥ 0.75 (Conte, Dunsmore and Shen 1986) or pred(0.30) ≥ 0.70 (MacDonell 1997). Thus, in this paper, the models were evaluated using these three prediction accuracy measures to determine whether each model can be considered to be accurate.

In addition, R^2, the maximum MRE observed, and statistics of the absolute residuals were used for the models' comparison, as these prediction accuracy measures have been used in a number of comparative studies of prediction models (Kemerer 1987, Shepperd, Cartwright, and Kadoda 2000, MacDonell 2003).

3.2 Models' prediction accuracy evaluation and comparison

The MMRE and pred measure values of the models presented in Section 2 are shown in Table 6, together with the models' maximum MRE values and R^2 values. Statistics of the models' absolute residuals are shown in Table 7 and used for the models' comparison. Table 8 shows the models' comparison results in terms of eight different prediction accuracy categories. The numbers in each column in Table 8 show the rank of the corresponding models in the category, 1 for the best and 10 for the worst.

The first three columns in Table 6 show that all values except one are better than the suggested value of the corresponding prediction accuracy measure. This means that each of the models can be considered to be accurate for the target environment, except for the univariate model as shown in Equation 2.9 when the value of pred(0.25) is considered. However, a number of researchers suggest that pred(0.30) criterion seems more appropriate than that of pred(0.25) (MacDonell 1997). Given that criterion, all the models can be considered to be accurate for the target environment.

Table 8 shows that the same model achieves a different ranking in different prediction accuracy categories when compared with other models. This result is consistent with the results presented by other researchers (Shepperd, Cartwright, and Kadoda 2000) and supports the suggestion that the most appropriate model should be chosen from the models presented in this paper based on the specific goals and needs of users. For example, models with a smaller number of variables are, in general, easier to collect data. Thus, when minimizing data collection time and cost is an important goal of users, a univariate model would be the most appropriate.

Table 8 also shows that the models including developers' productivity achieve, in general, better prediction accuracy than their productivity exclusive counterparts, as most of them show a higher rank order in most prediction accuracy categories. This result suggests that productivity is influential on development effort in the target environment.

4 Conclusions

4.1 Summary of the Findings

A total of 10 linear regression models were constructed in order to predict effort for a data-centred 4GL software development environment where a specific tool suite was used. These models were evaluated and compared in terms of commonly used prediction accuracy measures. The evaluation results showed that all the models achieved better prediction accuracy than the suggested values of all the three prediction accuracy measures used, with one exception. The models' comparison results showed that each of these models achieved a different ranking in different prediction accuracy categories in comparison with other models. These results suggest that users can choose the most appropriate model(s) from the models presented in this paper depending on their needs and goals.

This study also used student developers' marks in a practical development test as the productivity metric and examined the influence of productivity. The results showed that the productivity inclusive models, in general, achieved better prediction accuracy than their productivity exclusive counterparts. This suggests that developers' productivity is influential on effort in the target environment.

4.2 Limitations of the Study

All effort prediction models in this study were empirically constructed using historical data collected from software systems developed by non-professional, university undergraduate developers. This implies that the applicability of these models to industrial settings may be limited. In addition, the applicability of empirical effort prediction models is, in general, subject to the specific development environment where the models' historical data were collected. Given that, the models presented in this paper would not be exempted from such limitations.

Another limitation is that all the models were evaluated using the same data used for the construction. In other words, they were validated in terms of fitting accuracy to the data. The models' accuracy in predicting effort using unknown data, that is, predicting effort for future projects in the target environment, needs to be validated in further studies.

4.3 Future Directions

In addition to the topic mentioned in the previous section, a number of topics are identified for future studies. One is to investigate other productivity metric(s) to establish the best productivity inclusive model(s). Another topic is to compare the models presented in this paper with other effort prediction models, in particular, models constructed using other statistical techniques, machine learning techniques, fuzzy systems, neural networks, or probabilistic networks such as Bayesian networks. The models can also be compared with effort prediction models using expert's knowledge such as analogy. Investigating a method where a number of different modelling techniques are combined to construct a model, is also considered to be an interesting direction for the future.

5 References

Albrecht, A.J. and Gaffney JR., J.E. (1983): Software function, source lines of code, and development effort prediction: a software science validation. *IEEE Transactions on Software Engineering* **SE-9**(6):639-648.

Basili, V.R. and Weiss, D.M. (1984): A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering* **SE-10**(6):728-738.

Basili, V.R. and Rombach, H.D. (1988): The TAME project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* **14**(6):758-773.

Boehm, B.W. (1981): *Software Engineering Economics.* Englewood Cliffs, NJ, Prentice-Hall.

Boehm, B.W. (1984): Software engineering economics. *IEEE Transactions on Software Engineering* **10**(1):4-21.

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

Dolado, J.J. (1997): A study of the relationships among Albrecht and Mark II Function Points, lines of code 4GL and effort. *Journal of Systems Software* **37**:161-173.

Dolado, J.J. (2000): A validation of the component-based method for software size estimation. *IEEE Transactions on Software Engineering* **26**(10):1006-1021.

Fenton, N.E. and Pfleeger, S.L. (1997): *Software Metrics: A Rigorous & Practical Approach.* Boston, MA, PWS Publishing Company.

Glass, R.L. (2001): Frequently forgotten fundamental facts about Software Engineering. *IEEE Software* (May/June 2001):110-112.

Kemerer, C.F. (1987): An empirical validation of software cost estimation models. *Communications of the ACM* **30**(5):416-429.

Kitchenham, B.A., Pickard, L.M., MacDonell, S.G. and Shepperd, M.J. (2001): What accuracy statistics really measure. *IEE Proceedings-Software* **148**(3):81-85.

MacDonell, S.G. (1997): Establishing relationships between specification size and software process effort in CASE environment. *Information and Software Technology* **39**:35-45.

MacDonell, S.G., Shepperd, M.J. and Sallis, P.J. (1997): Metrics for database systems: an empirical study. *Proc. the 4th International Software metrics Symposium (METRICS'97),* Albuquerque, NM, 99-107, IEEE Computer Society Press.

MacDonell, S.G. (2003): Software source code sizing using fuzzy logic modelling. *Information and Software Technology,* **45**: 389-404.

Oivo, M. and Basili, V.R. (1992): Representing software engineering models: the TAME goal oriented approach. *IEEE Transactions on Software Engineering* **18**(10):886-897.

Shepperd, M. J., Cartwright, M. and Kadoda, G. (2000): On building prediction systems for software engineers. *Empirical Software Engineering,* **5**:175-182.

Tate, G. and Verner, J.M. (1990): Software sizing and costing models: a survey of empirical validation and comparison studies. *Journal of Information Technology* **5**:12-26.

Tate, G. and Verner, J.M. (1991): Approaches to measuring size of application products with CASE tools. *Information and Software Technology* **33**(9):622-628.

Verner, J.M. and Tate, G. (1988): Estimating size and effort in fourth-generation development. *IEEE Software* (July 1988):15-22.

Verner, J.M. and Tate, G. (1992): A software size model. *IEEE Transactions on Software Engineering* **18**(4): 265-278.

Wrigley, C.D. and Dexter, A.S. (1991): A model for measuring information system size. *MIS Quarterly* **15**:245-257.

Metrics	Median	Mean	Std. Dev.	Min.	Max.	Skewness	Kurtosis
EntityNum	17.00	19.41	5.17	12	30	0.715	0.027
AttributeNum	79.00	93.94	50.13	36	252	2.092	5.739
FormNum	12.00	12.53	3.91	7	20	0.324	- 0.921
ReportNum	8.00	6.76	3.03	2	11	- 0.386	- 1.143
GraphNum	0.00	0.59	1.73	0	7	3.581	13.419
TotalFormReportGraph	19.00	19.88	4.24	12	29	0.304	0.211
EntityForm	34.00	32.94	12.21	11	62	0.489	0.833
EntityReport	26.00	25.00	11.34	5	44	- 0.348	- 0.641
EntityGraph	0.00	1.35	4.37	0	18	3.884	15.493
EntityFormReportGraph	63.00	59.29	12.91	32	73	- 1.090	0.079
Productivity	59.17	57.57	8.69	42.50	73.96	0.199	- 0.295
Effort	359.25	398.06	92.90	258.20	568.65	0.557	- 0.702

Table 2: Descriptive Statistics for Metrics Data
(17 systems)

Metrics	Effort	EntityNum	AttributeNum	FormNum	ReportNum	GraphNum
EntityNum	.328					
AttributeNum	.426*	.646**				
FormNum	.591**	.480*	.238			
ReportNum	-.303	-.515*	-.315	-.535*		
GraphNum	.068	.171	-.321	.233	-.129	
TotalFormReportGraph	.302	.085	-.042	.707**	.157	.271
EntityForm	.685**	.775**	.604**	.579**	-.407	.123
EntityReport	-.110	-.635**	-.301	-.324	.836**	-.177
EntityGraph	.109	.225	-.291	.248	-.178	.994**
EntityFormReportGraph	.459*	.304	.229	.336	.195	.213

(a) Spearman's Rank Correlation Coefficients

Metrics	TotalFormReportGraph	EntityForm	EntityReport	EntityGraph
EntityForm	.218			
EntityReport	.262	-.334		
EntityGraph	.234	.176	-.232	
EntityFormReportGraph	.450*	.607**	.352	.240

(b) Spearman's Rank Correlation Coefficients

Table 3: Correlation Analysis Results
(one-tailed test * for significance levels less than 0.05, ** for significance levels less than 0.01)

Metrics	**Median**	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**Skewness**	**Kurtosis**
EntityNum	17.00	18.55	5.24	12	29	0.732	0.116
AttributeNum	66.00	73.82	28.47	36	138	1.321	1.813
FormNum	11.00	11.55	3.27	7	17	0.286	- 0.837
ReportNum	8.00	7.18	2.82	2	11	- 0.422	- 0.696
GraphNum	0.00	0.91	2.12	0	7	2.841	8.407
TotalFormReportGraph	18.00	19.64	4.06	14	29	1.223	1.976
EntityForm	27.00	30.09	13.42	11	62	1.190	2.656
EntityReport	26.00	26.00	11.22	5	44	- 0.237	- 0.007
EntityGraph	0.00	2.09	5.38	0	18	3.112	9.937
EntityFormReportGraph	60.00	58.18	11.86	29	70	- 1.014	- 0.105
Effort	354.00	372.14	82.44	258.20	553.80	1.018	1.278

Table 4: Descriptive Statistics for a Subset with the Same Productivity (productivity range between 50 and 70% inclusive, 11 systems)

Metrics	Effort	EntityNum	AttributeNum	FormNum	ReportNum	GraphNum
EntityNum	.295					
AttributeNum	.555*	.654*				
FormNum	.256	.637*	.187			
ReportNum	-.096	-.789**	-.288	-.646*		
GraphNum	.208	.278	-.087	.502	-.265	
TotalFormReportGraph	-.023	-.070	-.161	.583*	.148	.422
EntityForm	.635*	.822**	.858**	.408	-.595*	.264
EntityReport	.091	-.728**	-.245	-.452	.888**	-.306
EntityGraph	.289	.390	.006	.543*	-.364	.985**
EntityFormReportGraph	.795**	.517	.593*	.499	-.164	.401

(a) Spearman's Rank Correlation Coefficients

Metrics	TotalFormReportGraph	EntityForm	EntityReport	EntityGraph
EntityForm	-.181			
EntityReport	.226	-.475		
EntityGraph	.340	.363	-.410	
EntityFormReportGraph	.242	.686**	.009	.465

(b) Spearman's Rank Correlation Coefficients

Table 5: Correlation Analysis Results of a Subset with the Same Productivity (one-tailed test * for significance levels less than 0.05, ** for significance levels less than 0.01)

Models	**MMRE**	**Pred(0.25)**	**Pred(0.30)**	**Max. MRE**	**R^2**
Univariate model (2.1)	**0.1323**	**0.8824**	**0.8824**	0.3900	0.550
Univariate model (2.2)	**0.1431**	**0.7647**	**0.9412**	0.5117	0.364
Multivariate model (2.3)	**0.1139**	**0.8824**	**1.0000**	0.2790	0.569*
Multivariate model (2.4)	**0.1199**	**0.8824**	**0.8824**	0.3928	0.486*
Bivariate model (2.5)	**0.1136**	**0.8824**	**0.9412**	0.3723	0.622*
Multivariate model (2.6)	**0.0939**	**0.8824**	**1.0000**	0.2902	0.684*
Bivariate model (2.7)	**0.0989**	**0.8824**	**1.0000**	0.2664	0.590*
Multivariate model (2.8)	**0.0982**	**0.8824**	**1.0000**	0.2603	0.516*
Univariate model (2.9)	**0.1624**	0.7273	**0.8182**	0.4662	0.406
Univariate model (2.10)	**0.1305**	**0.9091**	**0.9091**	0.3700	0.536

Table 6: Models' Prediction Accuracy (figure in bold in the first 3 columns indicates that the value achieved is better than the suggested value, * for adjusted R^2)

Models	Median	Mean	Std. Dev.	Min.	Max.	Sum
Univariate model (2.1)	39.7556	49.2634	36.1309	1.2534	113.8291	837.4774
Univariate model (2.2)	38.3662	55.2831	47.3419	4.7278	151.6017	939.8121
Multivariate model (2.3)	34.4140	41.2840	31.2397	4.0923	116.9449	701.8282
Multivariate model (2.4)	27.8598	43.9509	39.3833	1.4690	113.9511	747.1660
Bivariate model (2.5)	36.1160	41.6707	31.7537	1.1873	119.5532	708.4026
Multivariate model (2.6)	27.5535	35.3512	29.8258	1.5620	115.5354	600.9711
Bivariate model (2.7)	28.1845	37.4431	28.7738	7.6512	115.8941	636.5331
Multivariate model (2.8)	31.9714	37.9468	31.0770	0.4414	119.5358	645.0963
Univariate model (2.9)	55.6373	59.8362	53.2548	2.1991	178.6546	658.1986
Univariate model (2.10)	38.1335	46.4417	27.9221	15.5016	101.0169	510.8592

Table 7: Statistics of the Absolute Residuals of the Models

	MMRE	Pred (0.25)	Pred (0.30)	Max. MRE	R^2	Median Ab. Res.	Max. Ab. Res.	Sum Ab. Res.
Model (2.1)	8	2	8	7	5	9	2	9
Model (2.2)	9	9	5	10	10	8	9	10
Model (2.3)	5	2	1	3	4	5	6	6
Model (2.4)	6	2	8	8	8	2	3	8
Model (2.5)*	4	2	5	6	2	6	8	7
Model (2.6)*	1	2	1	4	1	1	4	2
Model (2.7)*	3	2	1	2	3	3	5	3
Model (2.8)*	2	2	1	1	7	4	7	4
Model (2.9)*	10	10	10	9	9	10	10	5
Model (2.10)*	7	1	7	5	6	7	1	1

Table 8: Prediction Accuracy Comparison
(number shows the rank order, 1 indicates the highest prediction accuracy, 10 the lowest,
*** for productivity inclusive models)**