Empirical Software Engineering, 4, 135–158 (1999)

An Empirical Study of Analogy-based Software Effort Estimation

FIONA WALKERDEN f.walkerden@unsw.edu.au
Centre for Advanced Empirical Software Research (CAESAR), School of Information Systems, University of New South Wales, Sydney 2052 Australia

ROSS JEFFERY r.jeffery@unsw.edu.au
Centre for Advanced Empirical Software Research (CAESAR), School of Information Systems, University of New South Wales, Sydney 2052 Australia

Abstract. Conventional approaches to software cost estimation have focused on algorithmic cost models, where an estimate of effort is calculated from one or more numerical inputs via a mathematical model. Analogy-based estimation has recently emerged as a promising approach, with comparable accuracy to algorithmic methods in some studies, and it is potentially easier to understand and apply. The current study compares several methods of analogy-based software effort estimation with each other and also with a simple linear regression model. The results show that people are better than tools at selecting analogues for the data set used in this study. Estimates based on their selections, with a linear size adjustment to the analogue's effort value, proved more accurate than estimates based on analogues selected by tools, and also more accurate than estimates based on the simple regression model.

Keywords: Software cost estimation, analogy-based estimation, software size

1. Introduction

Conventional approaches to software cost estimation have focused on algorithmic cost models, where an estimate is calculated from one or more numerical inputs via a mathematical model. Where there is limited or incomplete data and limited expertise in numerical techniques, these models can be daunting to calibrate and use. Analogy-based estimation has recently emerged as a promising approach, with comparable accuracy to algorithmic methods in some studies, and it is potentially easier to understand and apply. Ease of use may be an important factor in the successful adoption of estimation methods within industry, so analogy-based estimation deserves further scrutiny.

The current study compares several methods of analogy-based software effort estimation with each other and also with a simple linear regression model. In particular, this study has investigated the performance of a group of people estimating by analogy, unaided by any tool. The results show that people are as good as or better than tools at selecting analogues for a target project in this study, where they needed to compare only a small number of projects (15) as potential analogues. Estimates based on their selected analogues, with a linear size adjustment to the analogue's effort values, proved more accurate than estimates based on analogues selected by tools, and also more accurate than estimates based on the simple regression model.

The process of estimation by analogy, and previous research in this area is reviewed in the following section. The actual comparisons performed in this study, and the results of these comparisons are described in subsequent sections.

2. Background

2.1. Estimating by Analogy

Estimating software project effort by analogy is an example of a case-based reasoning strategy. Case-based reasoning is a form of analogical reasoning where the potential analogues and target are examples of the same thing, for example software projects. An estimate of the effort to complete a new software project is made by analogy with one or more previously completed projects.[1]

Estimating software project effort by analogy usually involves a number of steps:

1. Measuring or estimating the values of project metrics for the target project;

2. Searching a repository of completed projects for projects similar to the target and selecting one or more projects as source analogues;

3. Using the effort value of the source analogue(s) as an initial estimate for the target project;

4. Comparing the known metric values for the target and source projects; and

5. Adjusting the effort estimate in light of the differences between the target and source projects.

2.1.1. ESTOR

Mukhopadhyay et al. (1992) developed ESTOR, a case-based reasoning tool to estimate project effort. The metrics used by ESTOR are function point components and inputs to the intermediate COCOMO model (Boehm, 1981). ESTOR selects an analogue for the target project by calculating the Euclidean distance between completed projects and the target and selecting the nearest neighbour. The effort value for the analogue is adjusted to take account of the differences between the source and the target by applying a set of rules.

ESTOR uses two projects as its source of potential analogues. The projects were reconstructed from the verbal protocols of an expert in analogical estimation. This expert (Vicinanza et al., 1991) estimated effort accurately for a set of 10 projects. The adjustment rules used by ESTOR were derived from the same protocols. The absolute relative error (ARE)[2] for the expert's 10 estimates was 31%. When tested on the same 10 projects the mean absolute relative error (MARE) of ESTOR's estimates was 51%.

2.1.2. ANGEL

Shepperd et al. (1996) describe the tool ANGEL, which also estimates project effort by analogy. ANGEL does not assume that estimators will use a particular set of project metrics. The estimator can set up ANGEL to use whatever project data set is available. ANGEL, like ESTOR, calculates the Euclidean distance between the target project and potential analogues. ANGEL ranks the potential analogues according to their distances from the target.

The estimator specifies which metrics to use when ANGEL searches for analogues. ANGEL can also determine the best subset of metrics to use when searching a particular data set. ANGEL considers all possible subsets of metrics and selects the subset that minimises the MARE for the data set, calculated by jack-knifing.

ANGEL derives an estimate from the ranked analogues by averaging the effort value of a number of the closest analogues, rather than adjusting for differences between the target project and the selected analogue(s). The simplest approach is to use only one analogue, in which case ANGEL uses the effort value of the nearest neighbour project as the estimate for the target project. Shepperd and Schofield (1997) use the average effort value of up to three analogues to derive estimates. They select the number of analogues that minimises the MARE for a particular data set.

When ANGEL is applied to the same data set as used in the ESTOR experiment, Shepperd and Schofield (1997) find that the MARE of the ANGEL's estimates is 62%. ANGEL is somewhat less accurate for this data set than the estimates made by ESTOR (53%).

Shepperd and Schofield (1997) also compare the accuracy of ANGEL's estimates with those from an algorithmic estimation model derived by stepwise regression on the same data set. The regression model proved less accurate than either of the analogical approaches for this data set, with a MARE of 107%.

2.2. Advantages of Estimating by Analogy

Researchers have explored a wide variety of approaches to software effort estimation (see Walkerden and Jeffery, 1997). The most common approach is an algorithmic model with an explicit functional form, for example the well-known COCOMO model (Boehm, 1981) and more recently COCOMOII (see, for example, Clark et al., 1998). The model relates the dependent variable, effort, to one or more independent variables, typically a size metric and one or more cost drivers. The simplest method for developing an algorithmic model is step-wise linear regression. The regression calculation calibrates a model relating effort to one or more metrics associated with effort (for example Shepperd et al., 1996; Briand et al., 1998).

Techniques from artificial intelligence research have also been applied to develop software effort estimation models. Srinivasan and Fisher (1995) use artificial neural networks and decision trees to estimate effort. These methods do not require the researcher to propose an explicit functional form for the model, only the input and output metrics.

The accuracy of estimates from experiments with ESTOR (Mukhopadhyay et al., 1992) and ANGEL (Shepperd et al., 1996) demonstrates that software effort estimation by analogy

is a viable alternative to other estimation methods. Analogy-based estimation also offers estimators some advantages over other methods. Potential advantages of software effort estimation by analogy are that:

1. It is easy to understand the basis for an estimate
   Analogy-based estimation stands in contrast to input-output models, by basing estimates on concrete past examples. It is an example of reasoning by analogy, a familiar mode of human problem solving (Burbridge, 1990; Kolodner, 1993). The familiarity of this approach may explain why people are comfortable estimating in this manner. Heemstra (1992) in a survey of nearly 600 organisations, reports that 61% used estimation by analogy whereas only 14% used algorithmic models. Lederer and Prasad (1993) also report that analogy-based estimation is the most common approach.

2. It is useful where the domain is difficult to model
   We know that many factors influence the effort needed to complete a software project. We know less about how these factors interact with each other, or how best to model the wealth of factors via software metrics. Estimation by analogy can be used successfully without having a clear model of how effort is related to other project factors. It relies primarily on selecting a past project that is similar to the target project, rather than postulating a general relationship between effort and other project characteristics that applies to all projects.

   Small historical data sets may be sufficient to develop simple algorithmic models, provided the data does not prove too noisy. However noise, unaccounted for variations in dependent variables, is at the crux of domains which are difficult to model. Shepperd et al. (1996) give an example of a data set of 8 projects for which no statistically significant relationships can be found. An algorithmic model based on this data set would be suspect. Nevertheless, the accuracy of analogical estimates for this data set was comparable to that of other much larger data sets.

3. It can be used with partial knowledge of the target project
   Analogy-based estimation addresses this problem by allowing people to use whatever information they have available to search for and select analogues, rather than prescribing particular inputs.

4. It has the potential to mitigate problems with calibration
   Analogy-based estimation has the potential to provide accurate estimates even using another organisation's data, provided an appropriate analogue for the target project is found within the data set used for estimation and the variables measured are both appropriate and measured in a consistent manner. An analogue is appropriate if effort and associated factors are related in a similar way for both the target project and analogue. This is possible even where the relationships differ for typical projects of the each organisation.

5. It has the potential to mitigate problems with outliers
   Most project data sets have outliers: projects that differ substantially from the typical project in the values of their metrics and relationships between them. Estimating by

analogy does not rely on calibrating a single model to suit all projects. If the target project is typical of a data set, it is likely that one or more appropriate analogues will be found to base the estimate on. Outliers in the data set have no influence on the estimate at all. If the target project is itself an outlier, at least the lack of a similar project analogue may make this apparent to the estimator. From one perspective it simplifies the issue of outliers which always involves tradeoffs when model building.

6. It offers the chance to learn from past experience
   When estimating by analogy, it is convenient to select potential analogues via scrutiny of available metric values because this information is concise and easily compared. Ideally analogy-based estimation would be applied within an organisation with access to other information associated with past projects, in addition to project metrics. Information such as project debriefing reports could help managers identify risks that the new project faces and avoid mistakes that have been made in the past.

Naturally, there are some difficulties with analogy-based estimation that temper its advantages. Its accuracy relies on four factors: the availability of an appropriate analogue, the soundness of the strategy for selecting it, the manner whereby differences between the analogue and target are allowed for when deriving an estimate, and the accuracy of the data used for both the analogue and the target.

What if there is no truly appropriate analogue within a data set for the target project. Analogy-based estimation faces the same problem here as any estimation method: the collection and maintenance of data on completed projects relevant to new estimation problems. Unfortunately an analogue may be selected and used regardless of its appropriateness. An old project could be selected as an analogue because it appears similar to the target project, although factors affecting effort have changed over time. For example, the unadjusted function point count of the old project and the target project may be similar, but the target project has a graphical user interface whereas the older project had a character-based user interface. The effort to develop a graphical user interface is likely to be greater than that for a character based user interface, but the estimator may overlook this difference if the nature of the user interface is not recorded in the available data.

Ideally an estimator can use his or her judgement to exclude inappropriate analogues. However, there is a possibility that estimators will use an analogue blindly, without justifying its selection. One bias observed in human reasoning is the tendency to seek evidence that confirms our opinions, and to neglect contrary evidence.

It is not clear how best to judge the appropriateness of a potential analogue for a target project. Analogical tools can assist in the selection process by ranking past projects according to how well they match the target. ESTOR (Vicinanza et al., 1991) and ANGEL (Shepperd et al., 1996) have successfully used the Euclidean distance between past projects and the target to rank potential analogues, with each metric weighted equally.

Ranking via Euclidean distance offers the benefits of clarity and consistency over human judgement alone. One drawback is that it does not allow for the different contributions that project metrics make to variation in project effort. Weighting project metrics could overcome this, but introduces another complexity to the selection process: how to assign weights. Expert opinion, correlation coefficients or learning algorithms are some alterna-

tives (Stensrud and Myrtveit, 1998; Shepperd and Schofield, 1997). However, applying the same weights to all projects is a compromise. While the factors influencing effort may be shared by all projects, the relative influence of these factors can vary from project to project. Human judgement may be better at managing the complexity of assessing the importance of productivity factors in both the target project and its analogues.

Once an analogue has been selected, the estimator is faced with the question of how best to use it to derive an estimate for the target project. It is probable that the analogue differs from the target project in some respects that influence effort. What adjustments should be made to the effort value of the analogue to reflect these differences? Shepperd et al. (1996) estimate by using the unadjusted effort of the analogue or the average effort of two or more analogues, thereby avoiding the problem of adjustment rules. Vicinanza et al. (1991) use adjustment rules derived from the verbal protocols of an expert. Both of these approaches proved successful, but as noted above, the factors influencing productivity will vary from project to project. Derivations of estimates from analogues would ideally take these differences into account.

The difficulties associated with analogy-based estimation, lack of appropriate analogues and issues with selecting and using them, should not deter our interest. All estimation methods will use imperfect data. Any estimation method needs experience and judgement to apply it successfully. The goal of this research is not to substitute analogy-based estimation for other methods. As Stensrud and Myrtveit (1998) observe, estimates produced by tools are never the final answer, rather tools are aids to improve the accuracy of estimates. The goal of this research is to explore how best to apply analogy-based estimation and to learn about its strengths and weaknesses compared with other methods.

2.3. ACE

A prototype tool, ACE (Analogical and Algorithmic Cost Estimator), has been developed as a means to explore the benefits of analogy-based estimation. ACE estimates effort for a target project by searching through a database of metrics for completed projects and selecting the completed project it judges most similar to the target project. ACE adjusts the effort value of the completed project to take account of the difference in size between the target and completed projects.

ACE ranks all projects in the database across the set of search metrics supplied by the user for the target project. For each metric in the set, ACE calculates the difference between the target project and each completed project. The completed project with the lowest difference is ranked 1 on that metric; the project with the next lowest difference is ranked 2, and so on. ACE calculates the mean rank of each completed project over the set of search metrics. The project with the lowest mean rank is selected as the analogue for the target project. Calculating the mean rank standardizes the contribution of each search metric to the final ranking.

If two completed projects differ from the target project by the same amount for a particular metric, then they are allocated the same rank, and the rank of the project with the next lowest rank is adjusted accordingly. For example, if the target project has a maximum team size (MTS) of 4 people, and two completed projects also have a MTS of 4, then they are

assigned ranks 1. The next most similar project has a MTS of 5 people, and is assigned rank 3. Categorical metrics, such as programming language used, are handled equivalently: all projects with the same categorical value as the target project are assigned ranks 1; all other projects are assigned the next rank, for example rank 4, if 3 completed projects used the same language as the target project.

Once ACE determines the highest ranking completed project, its effort value is adjusted to estimate effort for the target project. ACE performs a linear extrapolation along the dimension of a single metric, a size metric strongly correlated with effort such as function points.

$$Effort_{TARGET} = \frac{Effort_{ANALOGUE}}{FP_{ANALOGUE}} \times FP_{TARGET}$$

For example, if the effort to complete the source analogue was 1000 person-hours, its size 200 function points and the size of the target project is estimated as 250 function points, then the effort estimate for the target project is 1250 person-hours. This linear size adjustment attempts to account for the influence on effort of the difference in size between the target and completed projects. The linear size adjustment, based on unadjusted function points, is equivalent to using the productivity value of the analogue to predict the effort of the target project.

3. Comparison of Analogical Estimates

3.1. Overview

The performance of ACE has been compared with ANGEL and also with the performance of people instructed to estimate by analogy. We can think of analogy-based estimation in two stages: selecting an analogue for the target project; and deriving an estimate based on the analogue. For example, ACE selects as analogue the project with the highest rank over all search metrics nominated for the target project. Next ACE applies a linear size adjustment to the effort value of the analogue.

In contrast, ANGEL calculates the Euclidean distance between the target and all completed projects, and selects the closest as analogues. ANGEL uses all nominated search metrics or determines the metrics subset that best estimates the completed projects. ANGEL then uses either the effort value of the closest analogue without adjustment or the average effort value of the closest two or three analogues as its effort estimate for the target. In the current experiment, only the closest analogue is used to derive an estimate, to confine the comparisons to a manageable number. A linear size adjustment is also applied to ANGEL's analogue selection, to facilitate comparison with ACE.

In addition, a number of human subjects were instructed to estimate project effort by analogy. Twenty-five Masters students in the UNSW School of Information Systems subject "Software Engineering Management' participated in the experiment. They were free to choose their own methods for selecting analogues, and deriving estimates based on them. A linear size adjustment was also applied to the students' analogues, to facilitate comparisons between the alternative methods of analogue selection.

Table 1. Analogy-based estimation methods compared.

Analogue Selection Method	Adjustment Method
ACE	Linear Size Adjustment
ANGEL "All Metrics"	None
	Linear Size Adjustment
ANGEL "Best Metrics"	None
	Linear Size Adjustment
Human Subject	Subject's Adjustment
	Linear Size Adjustment

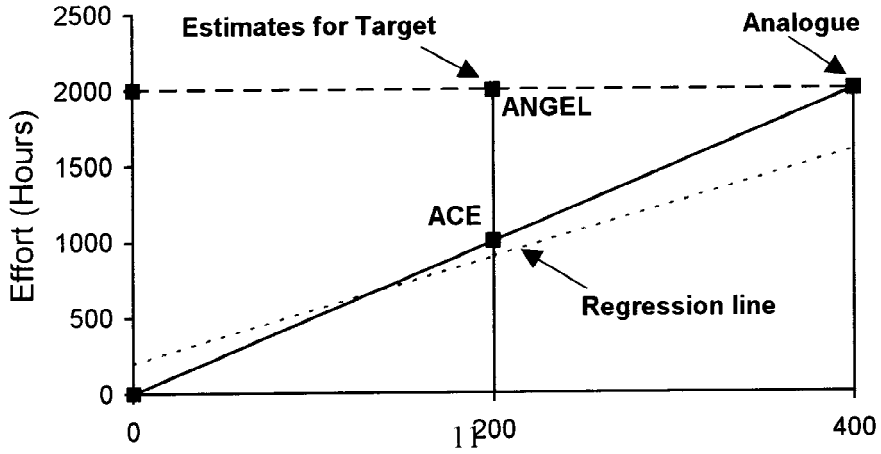

Figure 1. Comparison of estimation methods.

Overall, seven methods for estimation project effort by analogy have been compared experimentally (Table 1). These methods have also been compared with effort estimates derived by a linear regression model, as an example of a conventional algorithmic estimation model.

Figure 1 shows effort estimates for a target project with a size of 200 unadjusted function points. The ANGEL and ACE estimates are based on an analogue of size 400 UFP. The ANGEL estimate, without adjustment for size, is the same as the analogue effort value (2000 hours). The ACE effort estimate, with linear size adjustment, is half the analogue effort value. Thus in Figure 1 it can be seen that the techniques can result in very different estimate values.

Experimental Data Set

The data set used in this experiment consists of 19 projects with function point counts, total effort and a collection of other project related metrics. The source of this project data was an Australian software development company with approximately 50 employees, developing and distributing a range of projects locally and internationally. This data set has been used previously in a study on the use of function points as a size metric (Jeffery and Stathis, 1996).

The project metrics that have been used in the experiment are:

Table 2. Project metrics in experimental data set.

Metric	Unit/Range/Values	Scale
Total Effort	Person Hours	ratio
Unadjusted Function Points	UP	ratio
Maximum Team Size	People	ratio
Distributed System	Yes, No	nominal
Programming Language	COBOL, C, 4GL	nominal
Design Experience	1–5	ordinal
Language Experience	1–5	ordinal
Application Experience	1–5	ordinal

The total effort is the total time in hours spent by all project members on the project. The maximum team size is the maximum number of people who worked concurrently on the project. The experience metrics represent the relevant team members' average experience with software design, the programming language, and the project's application type. The project managers rated each project member involved in a particular activity on a scale of 1 to 5, with 1 being the least and 5 being most experienced.

Table 3 shows the wide range of project sizes represented in this project data set. Jeffery and Stathis (1996) show that for this project data set the MARE of a regression model based on unadjusted function points was as accurate as one based on adjusted function points. Unadjusted function points were used in this experiment as they offer a simpler description of the projects than function points, without loss of accuracy in effort estimation.

25 subsets of 15 projects were created from the 19 projects in the data set, by excising a hold-out sample in each case of 4 projects. Each project was numbered, from 1 to 19.

Table 3. Descripting statistics for experimental data set.

Metric	Mean	Standard Deviation	Minimum	Maximum
Total Effort	1947	3115	194	13905
UFP	527	876	38	3656
Maximum Team Size	3.7	2.3	1	10

The hold-out samples were chosen at random, by selecting two non-overlapping pairs of projects from the collection of all possible pairs of projects without replacement. Two pairs overlap if they have any project in common. Twenty five subsets were created, as this is was the number of subjects available for the experiment.

3.2. Procedure

3.2.1. Estimation Exercise

The exercise in estimating project effort by analogy was done on paper in a classroom setting and completed by 25 students. Prior to the exercise, students attended a one-hour lecture on software cost estimation as a part of a 14-week course on project management. This introduced them briefly to algorithmic and analogical estimation methods and software metrics used in estimation. In a previous week they had studied software size measures, including function points.

The subjects were motivated to participate in this exercise because it was to be marked with the mark contributing to their final grade in the subject. They were advised that the marker was looking for evidence that they had looked carefully at the project data presented to them, and that they had explained how they arrived at their estimates. The subjects were required to work alone on their estimates, but were free to ask the supervisor questions.

Each subject was allocated at random one of the 25 project data subsets, labelled "Set 1" to "Set 25". This consisted of a sheet of paper with a table of project metrics for 15 projects. Below the table was a brief explanation of each project metric.

Accompanying each project set was an estimation exercise. The subjects were asked to estimate total project effort by analogy for three projects. Each project was described by the same metrics as the projects in the data subset, with total effort excluded. Only 3 of the possible 4 projects in the hold-out sample were used. The hour available for the exercise did not permit subjects to attempt more than three estimates.

In Part I of the exercise, the subjects were asked to estimate project effort by analogy for two projects manually. The estimation process was broken into two steps. In step (i) subjects were asked to select the two projects most similar to the target project and also to explain how they selected them. They were asked to select two projects rather than one in a bid to encourage them to study the candidate projects thoroughly. In step (ii) they were asked to estimate the total effort in hours to complete the target project, based on their previous selection, again explaining how they arrived at their estimate.

3.2.2. Tool Estimates

The ACE estimates, for each of the 25 project data subsets, were calculated using a Microsoft Excel spreadsheet. Three estimates were made, one for each of the two target projects used in Part I of the estimation exercise, and one for the target project used in Part II. This paper reports only on Part I. The linear size adjustment to the analogue's effort was based on the unadjusted function point counts for the analogue and target projects.

The ANGEL estimates were calculated by creating an ANGEL data model for each of the 25 data subsets. Estimates for each of the three target projects were generated initially using all available project metrics. The estimate was simply the effort value of the closest analogue. Then ANGEL selected the best metrics subset for each of the 25 data subsets. The best subset was the one that minimised the mean absolute relative error of estimates for the 15 completed projects, holding out each of the 15 in turn. The analogue selected by ANGEL for each of the three target projects was noted. This allowed a linear size-adjusted estimate to be calculated in each case via a spreadsheet.

A "least squares" linear regression model with effort as the dependent and unadjusted function points as the independent variable was derived for each subset of 15 projects. The model was then used to predict effort for the three target projects. No examination of outliers was carried out. The inclusion of outliers may result in the accuracy of this method being somewhat under-estimated. However, this approach demonstrates how regression analysis performs without the aid of human judgement to examine and exclude outliers. The regression model used is of the form:

$$\text{Effort} = a + b\,\text{Function points}$$

In a previous paper we explored the performance of linear and log-linear regression on this data set (Jeffery and Stathis, 1996). In this we find that the model:

$$\text{Effort} = a.\,\text{Function points}^{b}$$

revealed a value for b of 0.816 with an r^2 of 0.77. For a linear regression the model showed an r^2 of 0.95 for the model $\text{Effort} = 192.31 + 3.45 * \text{FunctionPoints}$. The data plots reveal that the regression is driven by three projects that are significantly larger than all others. This is a common issue though for software engineering data sets.

3.3. Data Analysis

The MARE of each estimation method has been calculated for the target projects used. Fifty estimates were made, two for each of the 25 data subsets.

As well as using MARE as a measure of accuracy for each method, groups of estimates have been compared by counting the number of times a method provided the least accurate estimate (maximum ARE) and the number of times a method provided the most accurate estimate (minimum ARE). These counts are reported as a percentage of the total number of estimates compared. Looking at the proportion of estimates that were both best and worst for a method gives an indication of how consistent the accuracy of its estimates are.

Another measure of accuracy often used in the software cost estimation literature (for example, Shepperd et al. 1996) is the proportion of predictions of a given level of accuracy,

Table 4. Mean absolute relative errors of tool estimates.

Estimation Method	Mean ARE%	N	Standard Deviation	Standard Error	Pred(.25) %	% Cases Min ARE	% Cases Max ARE
ACE	55	50	62	9	24	34	14
Linear Regression	68	50	65	9	16	18	32
ANGEL Best Metrics	112	50	178	25	28	30	30
ANGEL All Metrics	125	50	181	25	28	30	40

defined as:

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

where N is the total number of observations and k is the number of observations with MARE less than or equal to l. The value of PRED(.25) has been calculated for each estimation method as well.

Estimation methods have been compared, two at a time, via paired sample t-Tests. These test whether the mean difference in ARE between pairs of estimates for the same target project and same data subset differs significantly from zero. The mean difference and p value is reported for each comparison.

4. Results

4.1. Tool Comparisons

ACE performed best on average of the four tools, with the lowest mean absolute relative error (MARE). It also had the highest proportion of cases with the minimum ARE of the four tools, and the lowest proportion of cases with the maximum ARE (Table 4).

The proportions of cases with minimum and maximum ARE do not add to 100% because the same estimate may be calculated by more than one method, especially where the same analogue is selected by two or more analogical methods.

The linear regression model of effort against unadjusted function points has the second lowest MARE. Paired sample t-Tests comparing the estimates for each case show that the means for ACE and the linear regression model do not differ significantly, although both are significantly lower than the means for the two ANGEL methods (Table 5).

Despite the high ARE, both ANGEL methods have a greater proportion of cases with the minimum ARE than the linear regression model and higher Pred(.25). The ANGEL "Best Metrics" method has as a similar proportion of cases with the maximum ARE as the linear regression model.ANGEL's MARE has been adversely affected by a number of estimates with very high absolute relative errors.

Table 5. Paired sample *t*-Tests for absolute relative errors of tool estimates.

Comparison			Mean Difference in ARE%	p Value (2 tailed)
ACE	V	ANGEL Best Metrice	−57	0.023*
ACE	V	ANGEL All Metrics	−70	0.007 **
ACE	V	Linear Regression	−13	0.186
ANGEL Best Metrics	V	ANGEL All Metrics	−13	0.547
ANGEL Best Metrics	V	Linear Regression	+44	0.032*
ANGEL All Metrics	V	Linear Regression	+57	0.009**

One characteristic that most of ANGEL's high ARE estimates share is that the project selected by ANGEL as an analogue differs widely in size (unadjusted function points) from the target project. The ANGEL estimate for each target project used in this experiment is simply the effort value for the analogue. No adjustment is made for their relative sizes. Shepperd et al. (1996) use the average effort value for two or three analogues. Averaging is likely to improve the estimate where the sizes of the analogue projects straddle the size of the target.

Shepperd et al. (1996) found that ANGEL equalled or outperformed linear regression on six separate project data sets. This is not the case with the data set used in the current experiment. The linear regression model and ACE both estimate by taking the size of the target project in unadjusted function points into account. This appears to be the source of their advantage when compared with ANGEL in this study. The effect of a linear size adjustment on the ANGEL estimates is explored below in section 4.3.

4.2. Subject versus Tools

In Part I of the estimation exercise students estimated project effort unaided by any tool. One of the 25 subjects failed to make the two estimates required as part of the exercise, hence 48 rather than 50 estimates are compared below.

The subjects' estimates are similar in accuracy to ACE's. They have twice as many of the least accurate estimates (max ARE) as ACE, although this has not lead to an appreciably higher MARE (Table 6). The subjects' had a greater proportion of accurate estimates than ACE. Pred(.25) for the subjects' estimates is 36%, whereas Pred(.25) for ACE is 24%.

Paired sample *t*-Tests comparing the estimates for each case show that the MARE for the subjects estimates, unaided by tools, is significantly lower than the means for the two ANGEL methods (Table 7). The MARE of the subjects' estimates does not differ significantly from ACE or the linear regression model.

Based on their selected analogues, 52% (13/25) subjects made well-reasoned adjustments to the analogue effort value to take into account differences between the analogue and target. Eleven of these thirteen subjects made an adjustment to the analogue's effort value to take into account the difference in size in UFP between the analogue and target projects.

Table 6. Mean absolute relative errors for subjects unaided and tools.

Estimation Method	Mean ARE%	N	Standard Deviation	Standard Error	Pred(.25) %	% Cases Min ARE	% Cases Max ARE
Subject Unaided	56	48	62	9	38	24	20
ACE	54	48	63	9	25	24	10
Linear Regression	67	48	66	10	17	16	22
ANGEL Best Metrics	100	48	155	22	29	16	22
ANGEL All Metrics	114	48	160	23	29	30	34

Table 7. Paired sample *t*-Tests for ARE of subjects unaided and tools.

Comparison			Mean Difference in ARE%	p Value (2 tailed)
Subject Unaided	V	ACE	+1	0.893
Subject Unaided	V	Linear Regression	−11	0.290
Subject Unaided	V	ANGEL Best Metrics	−44	0.033*
Subject Unaided	V	ANGEL All Metrics	−58	0.015*

Significant results: $^{*}p < .05$ $^{**}p < .01$

Despite the overall accuracy of the subjects' estimates, 44% (11/25) subjects made adjustments that were based on incorrect reasoning. Eight of these eleven adjustments involved a misunderstanding about the relationship between the total project effort in person-hours and maximum team size. Subjects reasoned, for example, that a project with a total effort of 250 person-hours and a maximum team size of 2, would take one person 500 person-hours to complete.

Although adjustments based on maximum team size were based on a misunderstanding, estimates made in this way were not inevitably inaccurate. The maximum team size is an indirect measure of overall project size, as projects with higher function point counts tend also to have larger teams.

Different wording on the experimental material describing the project metrics may have avoided this misunderstanding of the maximum team size metric. However, the misunderstanding does suggest that the group of student subjects in this experiment were inexperienced. A group of subjects drawn from industry may be more likely to apply well-reasoned adjustments to the analogue effort value.

Table 8. Mean ARE with linear size adjustments applied to analogue effort.

Estimation Method	Mean ARE%	N	Standard Deviation	Standard Error	Pred(.25) %	% Cases Min ARE	% Cases Max ARE
Subject + Size Adj.	39	50	38	5	36	60	14
ANGEL All + Size Adj.	46	50	46	7	24	48	20
ACE	55	50	62	9	24	38	20
ANGEL Best + Size Adj.	60	50	61	9	20	36	38
Linear Regression	68	50	65	9	16	12	48

4.3. Comparison of Analogue Selections plus Linear Size Adjustment

The project data set used in this experiment shows a strong correlation between total project effort and size in function points. The Pearson correlation between total effort in hours and unadjusted function points is 0.97 ($p < .001$) for the 19 projects. This high value is influenced by the three largest projects in the data set. A more conservative estimate of the correlation, which excludes these three outliers, is 0.68 ($p < .01$). Project size in unadjusted function points explains more than two thirds of the variation in total effort.

When the selected analogue differs appreciably in size from the project to be estimated, adjusting the analogue's effort value to take account of the difference in size should improve the accuracy of the estimate. This is the advantage that ACE and the subjects' unaided estimates have over the ANGEL methods that use the analogue effort value without adjustment. The linear regression model also takes size into account, as the independent variable is unadjusted function points.

A linear size adjustment was applied to the effort values for the analogues selected by the two ANGEL methods. A linear size adjustment was also applied to the effort values for the subjects' selected analogues, since 44% of subjects unaided estimates were based on poorly reasoned adjustments. This allows us to compare the accuracy of estimates based on the analogue selection strategies of ACE, the subjects and ANGEL. The MARE for these methods with size adjustment are shown in Table 8.

The subject who failed to make effort estimates did however manage to select analogues for the two projects in their exercise, so the number of comparisons shown below is 50, rather than 48 (Table 8).

Table 9 shows that the MARE of the ANGEL estimates improves dramatically when a linear size adjustment is applied. Estimates based on the subjects' analogue selection plus a linear size adjustment also prove significantly more accurate than the subjects' unaided estimates.

Table 9. Paired sample *t*-Tests showing improvement due to size adjustment.

Comparison			**Mean Difference in ARE %**	**p Value (2 tailed)**
Subject Unaided	V	Subject Analogue + Size Adjustment	+16	.042*
ANGEL All Metrics	V	Angel All Metrics + Size Adjustment	+78	.001**
ANGEL Best Metrics	V	ANGEL Best Metrics + Size Adjustment	+52	.013*

Table 10. Paired sample *t*-Tests comparing ARE for size-adjusted estimates.

Comparison			**Mean Difference in ARE %**	**p Value (2 tailed)**
Subject Analogue + Size Adjustment	V	ANGEL All Metrics + Size Adjustment	−7	.119
Subject Analogue + Size Adjustment	V	ACE	−15	.041*
Subject Analogue + Size Adjustment	V	ANGEL Best Metrics + Size Adjustment	−21	.001**
Subject Adjustment + Size Adjustment	V	Linear Regression	−29	<.001**
ANGEL All Metrics + Size Adjustment	V	ACE	−9	.312
ANGEL All Metrics + Size Adjustment	V	ANGEL Best Metrics + Size Adjustment	−14	.017*
ANGEL All Metrics + Size Adjustment	V	Linear Regression	−19	<.001**
ANGEL Best Metrics + Size Adjustment	V	ACE	+5	.521
ANGEL Best Metrics + Size Adjustment	V	Linear Regression	−8	.280

Significant results: $^{*}p < .05$ $^{**}p < .01$

When the analogical methods with linear size adjustment are compared with each other and with the linear regression model (Table 10) the subjects' analogue selection plus linear size adjustment stands out as the most accurate method. Its MARE is significantly lower than all methods except one, ANGEL "All Metrics" with linear size adjustment. The subjects' analogue selection plus size adjustment also has the high proportion of cases with minimum ARE (60%) and the lowest proportion of cases with the worst ARE (14%).

The impact of a linear size adjustment on the accuracy of ANGEL's estimates can be seen in the results for the data in "Set 15" used in the estimation exercise. Figure 2 shows the ANGEL "All Metrics" estimates for projects 1 and 11 for "Set 15", with and without linear size adjustment. ANGEL has selected project 2 as the analogue for both project 1 and 11. The unadjusted function point count for project 2 is 1834. Project 11 has a much

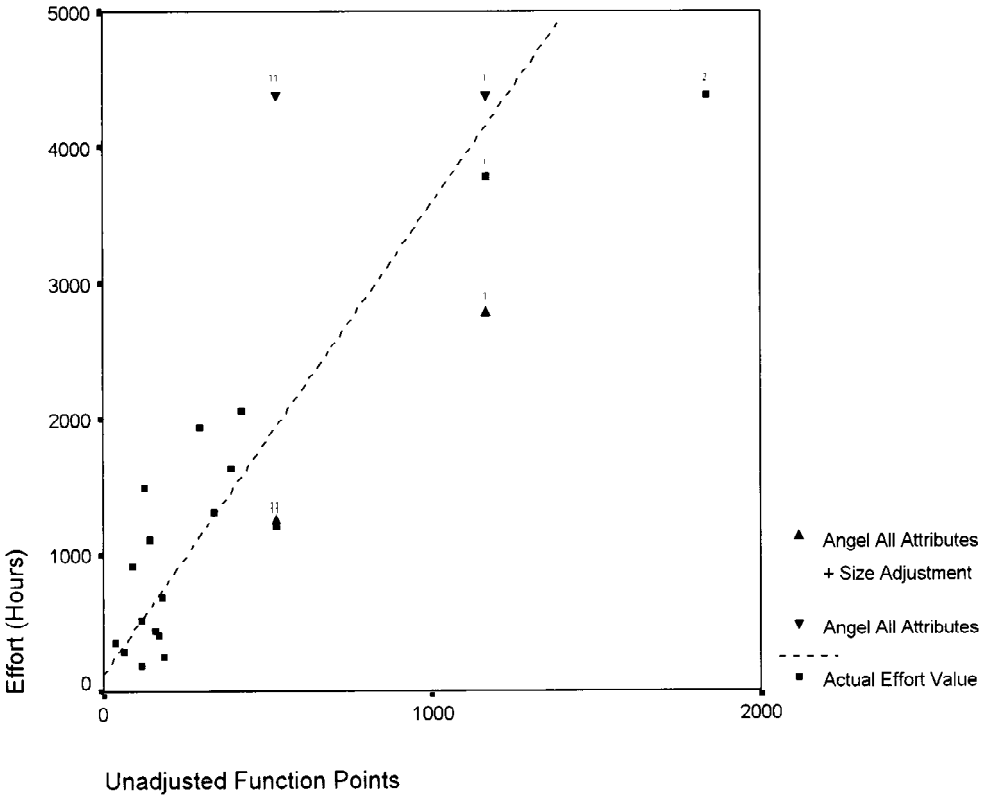

Figure 2. ANGEL "All Metrics" estimates for set 15.

lower adjusted function count of 526. Before the size adjustment is applied, the effort for project 11 is over-estimated by 260%. After the size adjustment is applied the estimate is only 3% over the actual effort value. This example shows the dramatic improvement in accuracy that the linear size adjustment provides in this data set.

Project 1 has an unadjusted function point count of 1164, considerably closer to that of the analogue project 2. When the linear size adjustment is applied in this example the accuracy of the estimate worsens from 16% to 26%. No single method will provide the most accurate estimates in all cases.

When no size adjustment is applied to the analogue effort value ANGEL "Best Metrics" has a lower MARE than ANGEL "All Metrics". This is not surprising since the best metrics, for each subset of projects, are selected by minimising MARE. However, when the linear size adjustment is applied ANGEL "All Metrics" has a significantly lower MARE than ANGEL "Best Metrics". If the best metrics for the project samples had been selected by minimising the ARE of estimates with linear size adjustment, then this result may have been reversed.

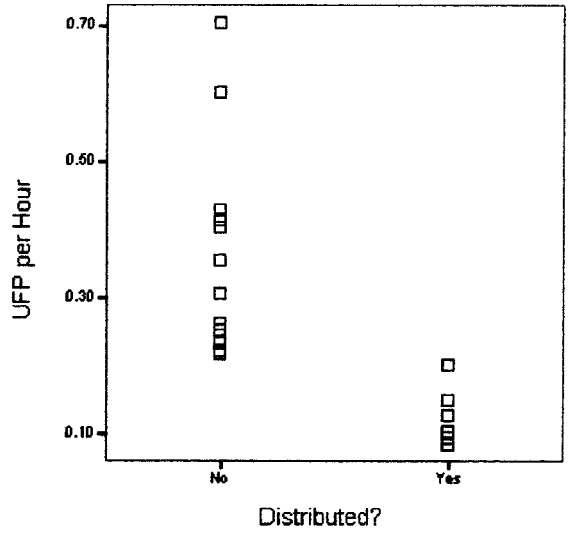

Figure 3. Productivity of distributed and non-distributed projects.

Nevertheless, ANGEL "All Metrics" has selected an appropriate set of analogues to base effort estimates on. Its analogue selections are explored in more detail in section 4.4 below.

The linear regression model has a significantly higher MARE than both the subjects' analogue selection plus linear size adjustment and ANGEL "All Metrics" with linear size adjustment. It now has nearly half the cases with the maximum ARE. In comparison to the size adjusted analogical methods, the linear regression model is a poor performer overall for this project data set. However, there are cases where the linear regression model gives a satisfactory estimate. Figure 2 shows the regression line for "Set 15". The regression estimate for project 1 over-estimates the effort by only 13%, which is more accurate than either the adjusted or unadjusted ANGEL estimates.

4.4. Sources of Advantage in Analogue Selection

When the same linear size adjustment is applied to the analogue selections of the subjects, ACE and ANGEL, the subjects appear to have selected the most appropriate analogues. The estimates based on their selections are the most accurate of all the methods. Is there anything characterising the subjects' selections to explain their relative success?

The productivity of projects within the data set, measured in unadjusted function points per hour, is lower for the projects that developed distributed (client-server) than projects which developed non-distributed systems. Figure 3 shows a scatter plot of project productivity. There is no overlap between the productivity values of projects in the two subsets.

Table 11. Comparison of project productivity.

	Distributed?	N	Mean	Standard Deviation	Standard Error
UFP per Hour	Yes	6	.129	.044	.018
	No	13	.361	.152	.042

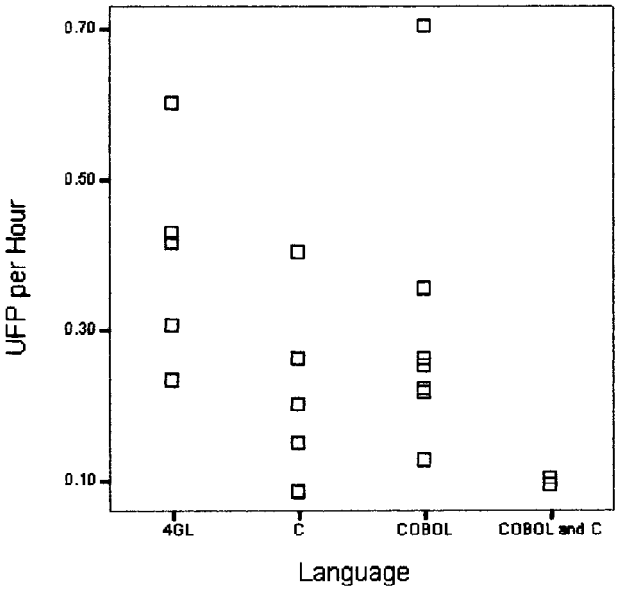

Figure 4. Productivity of projects by language.

A *t*-Test shows a significant difference in mean productivity for projects in the two subsets ($p < .001$), although the sample sizes are small (Table 11). The distributed systems built by projects in this data set had a greater proportion of non-functional requirements than the non-distributed systems, which are not reflected in the function point counts, and thus affect the apparent productivity of the projects.

Common selection criteria used by subjects to select analogues were a match on programming language and a match on the "Distributed" metric of the target project. 56% (14/25) of the subjects mentioned a match on the "Distributed" metric. The actual proportion of the subjects' analogues that matched on the "Distributed" metric was 98% (49/50 cases).

The proportion of subjects mentioning a match on programming language was 72% (18/25). The actual proportion of analogues that matched was 74% (37/50). There is no clear relationship between project productivity and language (Figure 4). However, the match on the language may have helped boost the proportion of analogues matching

on the "Distributed" metric. All the 4GL projects and all but one of the COBOL-only projects were not distributed. Such multi-colinearity between software metrics is a common occurrence.

It is probable that the match on the "Distributed" metric contributes to the superior accuracy of the subjects' estimates, because of the relationship between project productivity and whether or not a project is distributed.

In contrast to the subjects' analogue selections, 22% (11/50) of the ACE analogue selections fail to match the target project's "Distributed" metric. In particular, there are 10 cases where ACE selected an analogue that did not match the target project's "Distributed" metric but the subjects did. For these 10 cases the MARE of the subjects' size-adjusted estimates is 29% whereas the mean for ACE's estimates is 87%. These cases include one where ACE is very inaccurate, with an ARE of 402%. Even excluding this case, the subjects size-adjusted estimates are clearly more accurate than ACE's, with a MARE of 27% compared with ACE, 52%.

A match on the target project's "Distributed" metric does not explain all of the subjects' advantage in analogue selection, however. In 34% (17/50) of estimates ACE selects a different analogue to the subjects', but both select analogues that match the target project's "Distributed" metric. The subjects' size-adjusted estimates are still more accurate than ACE's with a MARE of 27% compared with ACE, 39%.

ANGEL "All Metrics" is the method whose analogue selections were next most successful, compared with the subjects'. The ANGEL "All Metrics" analogues matched the target project's "Distributed" metric in 98% (49/50) cases, the same proportion as the subjects. ANGEL "All Metrics" also selected the same analogues as the subjects in 62% (31/50) of cases. In the cases where ANGEL "All Metrics" selected a different analogue to the subjects, whilst both matched the target on the "Distributed" metric, the subjects were again more accurate with MARE of 39% versus ANGEL "All Metrics", 60%.

4.5. Discussion of Findings

The results above show that people in this study are better than tools at selecting project analogues from the small data sets provided. The subjects in this experiment needed to consider only 15 projects as potential analogues and only seven metric values. Tool support in searching a project repository is likely to be more valuable when there is a much larger number of potential analogues and a richer variety of project information.

The subjects in this experiment were not expert. Close to half of them made inappropriate adjustments to the analogue effort value when deriving an estimate for the target project. Nevertheless, their estimates compare favourably with those made by the tools. The analogical approach which the subjects were instructed to employ may have contributed to their success. Vicinanza et al. (1991) conducted an exploratory study of five software effort estimators. They found that only one of the five employed an analogical approach, and that this estimator made the most accurate estimates.

The subjects' analogue selections were successful because in most cases they excluded projects that did not match the target project on key attributes. Most of the subjects believed

that matches on the "Distributed" metric or the language used were important. These beliefs were appropriate for the data set used in this experiment.

When we employ analogical reasoning we weigh up the relative importance of similarities and differences between the source analogue and the target. In the estimation exercise, some subjects reasoned that even when a potential analogue and the target project were similar in all respects other than whether the projects were distributed, this dissimilarity was more important than the other similarities. They rejected the potential analogue.

If the subjects' beliefs about the importance of key project attributes had been misleading, their analogue selections may have been no more successful than those of ANGEL and ACE. In contrast to the subjects' analogue selection strategy, ACE and ANGEL "All Metrics" treated all project metrics equally. The vector space distance calculation of ANGEL and the rank calculation based on deviation from the target project used by ACE both performed relatively well.

Subjects were not given any information on how to select analogues prior to the estimation exercise. However two of the subjects used the same ranking algorithm as ACE. This suggests that where estimators have no prior beliefs about the relative importance of project metrics, they may be satisfied with the ranking methods of tools such as ACE.

Applying a linear size adjustment to the analogue project effort results in estimates that are significantly more accurate than estimates derived with adjustment on the data set used in this experiment. We expect this improvement where the analogue selected differs appreciably in size from target, because there is a strong relationship between size in unadjusted function points and effort in the experimental data set. It is common for project data sets to have some size metric strongly correlated with effort, so linear size adjustment is a good candidate for a simple adjustment rule.

Although it proved inaccurate in this study, the strategy of using the analogue effort value without adjustment may be appropriate in situations where a potential analogue and target are similar in size and other attributes. There may be little basis for believing that either increasing or decreasing the analogue effort value will improve the accuracy of the estimate in such situations. Even without adjustment ANGEL's estimates proved the most accurate of all methods in some cases.

A strategy of averaging the effort values of several analogues is likely to improve the accuracy of ANGEL's estimates where the average size of the analogues is closer to the target than any of the analogues individually. The use of averages in Shepperd et al. (1996) may explain the similarity in inaccuracy of their analogy-based estimates and those based on regression models.

In this experiment analogy-based estimates without adjustment were less accurate on average than estimates derived from a univariate linear regression model based on unadjusted function points. This is consistent with results reported by Briand et al. (1998) and Stensrud and Myrtveit (1998). However, when a linear size adjustment was applied to the analogue's effort value, the analogy-based estimates were more accurate on average than the regression estimates.

Although analogy-based estimation is successful on the experimental data set, it relies on a variety of information being available about both the target project and potential analogues. If an estimate were required for a target project and little was known about

it or past projects except size and effort, a regression estimate would be an appropriate choice.

When assessing estimation methods, it is appropriate to compare the average error of each as an overall measure of a method's accuracy. Nevertheless, each method compared in this study provided the most accurate estimate for some projects. The variation in accuracy of different methods underscores the value of using more than one method and comparing the results when putting effort estimation into practice.

5. Conclusion

In this study, people prove better than tools at selecting project analogues, when provided with a small data set. It appears people perform well when they exclude projects that do not match the target project on key attributes associated with effort.

Applying a linear size adjustment to the effort value of the project selected as the analogue resulted in estimates that are more accurate on average than estimates derived without adjustment or estimates derived by people unaided. The inexperience of the estimators in this study may have contributed to the inaccuracy of their adjustments.

These conclusions indicate that it can be fruitful to combine the talents of people and tools in the task of software effort estimation. This is a welcome conclusion, since as other authors observe (Hughes, 1996; Stensrud and Myrtveit, 1998), the role of tools should be to support estimators rather than supplant them.

Estimation by analogy, with adjustment to the analogue's effort value, has proved more accurate in this study than estimates derived from a univariate linear regression model based on unadjusted function points. However, when no adjustment is applied to the effort value of the analogue, the regression model proves more accurate. The current study suggests it is prudent to adjust the analogue's effort value, if the selected analogue differs widely from the target project along a dimension highly correlated with effort, such as size in function points.

The satisfactory performance of analogy-based estimation in this study is encouraging, as it is a method that is both easy to understand and simple to apply. This augurs well for a more formal application of this method in industrial settings.

Optimism about methods for software effort estimation based on comparative studies such as this must be tempered by the observation that the average error in estimates reported for even the best methods is often 50% or higher. Also, these studies are conducted when projects are complete, and therefore their final sizes are known. In practice, estimates of effort must be made based on estimates of a project's final size, which is likely to increase the error in effort estimates.

Nevertheless, methods such as software effort estimation by analogy offer practitioners the opportunity to estimate in a manner that is consistent and clearly understood, as an alternative to inspired guess-work based on foggy memories. The goal of an estimation method is to provide software developers with a range of feasible effort estimates, based on the current knowledge of a project. The task of project managers remains the difficult one of using estimates to set a target for the project, and then working to achieve that target in the face of inevitable challenges that the project will encounter.

Acknowledgments

This work was supported by a grant from the CSIRO.

Notes

1. In previous research the size range of projects used was from around 100 to 2300 function points.
2. The accuracy of each estimation method has been assessed primarily by calculation of the absolute relative error, (ARE), of each estimate. ARE, as a percentage of the actual effort for a project, is defined by: $ARE = 100|(\textit{Actual Effort} - \textit{Estimated Effort})|/\textit{Actual Effort}$. The mean ARE (MARE) can be calculated for a set of estimates. This accuracy measure has been used widely in the software cost estimation literature (for example Kemerer, 1987; Jeffery and Low, 1990; Mukhopadhyay et al., 1992; Schofield et al., 1996).

References

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

Briand, L. C., El Emam, K., Surmann, D., and Wieczorek, I. 1998. An assessment and comparison of common software estimation modeling techniques. International Software Engineering Research Network Technical Report, ISERN-98-27.

Burbridge, J. 1990. *Within Reason: A Guide to Non-deductive Reasoning*. Ontario: Broadview Press.

Clark, B., Devnani-Chulani, S., and Boehm, B. 1998. Calibrating the COCOMO II post-architecture model. *Proceedings of the 20th International Conference on Software Engineering*. Kyoto: IEEE Computer Society, pp. 477–482.

Heemstra, F. J. 1992. Software cost estimation. *Information and Software Technology* 34(10):627–639.

Hughes, R. T. 1996. Expert judgement as an estimating method. *Information and Software Technology* 38: 67–75.

Jeffery, D. R., and Low, G. C. 1990. Calibrating estimation tools for software development. *Software Engineering Journal* 5(4): 215–221.

Jeffery, D. R., and Stathis, J. 1996. Function point sizing: Structure, validity and applicability. *Empirical Software Engineering* 1(1): 11–30.

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

Kolodner, J. L. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Lederer, A. L., and Prasad, J. Information systems software cost estimating: A current assessment. *Journal of Information Technology* 8: 22–33.

Mukhopadhyay, T., Vicinanza, S., and Pietula, M. J. 1992. Estimating the feasibility of a case-based reasoning model for software effort estimation. *MIS Quarterly* 16(2): 155–171.

Shepperd, M., and Schofield, C. 1997. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering* 23(12): 736–743.

Shepperd, M., Schofield, C., and Kitchenham, B. 1996. Effort estimation using analogy. *Proceedings of the 18th International Conference on Software Engineering*. Berlin, Germany.

Stensrud, E., and Myrtveit, I. 1998. Human performance estimating with analogy and regression models: An empirical validation. *Proceedings of the 5th International Symposium on Software Metrics*. Bethesda, Maryland, USA.

Srinivasan, K., and Fisher, D. 1995. Machine learning approaches to estimating software development effort. *IEEE Transactions on Software Engineering* 21(2): 126–137.

Vicinanza, S. S., Mukhopadhyay, T., and Prietula, M. J. 1991. Software effort estimation: An exploratory study of expert performance. *Information Systems Research* 2(4): 243–262.

Walkerden, F., and Jeffery, R. 1997. Software cost estimation: A review of models, process and practice. *Advances in Computers* 44: 59–125.

Ross Jeffery is Professor of Information Systems and Director of the Centre for Advanced Empirical Software Research (CAESAR) at The University of New South Wales. He was the first Head of the School of Information Systems at UNSW from 1989 to 1994. He was the founding chairman of the Australian Software Metrics Association and also was instrumental in creating the Australian Conference on Information Systems for which he was the General Chair for the first two meetings. He is on the editorial boards of the IEEE Transactions on Software Engineering, the Journal of Empirical Software Engineering, and the Wiley International Series in Information Systems. He is also a founding member of the International Software Engineering Research Network (ISERN). His current research interests are in software engineering process and product modeling and improvement, software metrics, software technical and management reviews, and software resource modeling. His research has involved over 50 government and industry organisations over a period of 15 years. He has also held positions at the University of Queensland, University of Maryland, and the University of Minnesota. He has authored/co-authored four books and over 70 research papers.

Fiona Walkerden has a Masters degree in Information Science from the University of New South Wales. She has worked in industry as a software developer for over eight years, gaining wide experience in system analysis, design, integration and testing. Fiona has also worked as a research assistant within the Centre for Advanced Empirical Software Research, investigating techniques for estimating effort and duration of software development.