Technical Report TR-1520 July 1985

FINDING RELATIONSHIPS BETWEEN EFFORT AND OTHER VARIABLES IN THE SEL

Victor R. Basili
N. Monina Panlilio-Yap

Department of Computer Science
University of Maryland
College Park MD 20742

ABSTRACT

Estimating the amount of effort required for a software development project is one of the major aspects of resource estimation for that project. In this study, we examined the relationship between effort and other variables for 23 Software Engineering Laboratory projects that were developed for NASA/Goddard Space Flight Center. These variables fell into two categories: those which can be determined in the early stages of project development and may therefore be useful in a baseline equation for predicting effort in future projects, and those which can be used mainly to characterize or evaluate effort requirements and thus enhance our understanding of the software development process in this environment. The results of our analyses are presented in this paper.

* Research for this study was supported in part by the National Aeronautics and Space Administration under grant NSG-5123 to the University of Maryland. Computer support was provided in part by the Computer Science Center at the University of Maryland.

1. INTRODUCTION

The estimation of resources required in the development of a software project is an issue of importance to managers. The development of useful models and equations for predicting the cost of a project is one of the major goals of software engineering. One of the ways of measuring cost is to measure the amount of effort and resources required for a project.

Several studies on measures of effort have been made and two basic approaches have been taken in these studies. [Wolverton 74], [Putnam 78], and [Boehm 81], among others, have developed generalized models which are then parameterized for a given environment in order to predict effort. The models are based upon data from at least one environment which is hoped to be typical or representative. [Walston & Felix 77], [Jeffery and Lawrence 79], [Basili & Freburger 81], [Boydston 84], etc., have collected data from several projects in a given environment and used these data to build models for characterizing or predicting effort in that environment, as we have in the Software Engineering Laboratory. Because of the differences in the environments, the types of projects and the data collected, [Bailey and Basili 81] have suggested that even generalized models are not necessarily transportable to other environments where a different set of factors come into play in different degrees.

[Bailey & Basili 81] have proposed a method for generating a resource estimation model for a particular organization based on data collected in that environment. These data would capture environmental factors and differences among projects which may have some impact on the software development process. The basic approach is as follows: A background equation is computed. The factors that could possibly explain the difference between the actual effort and the effort predicted by the background equation for the available data are analyzed. The model is then used to predict the effort required for a new project. The approach requires a local data base. If such a data base is not available then clearly one of the generalized models is best.

It has been suggested by [Boehm 84] that lines of code is not necessarily the best predictor for effort. In a study conducted at the IBM Santa Teresa Laboratory, [Boydston 84] has found that the number of modified modules is very strong statistically and is superior to lines of code as a single variable determinator of effort. Thus we are searching our summary statistics file to see if there are other variables that might be better to use, especially in a baseline equation.

This paper presents the results of some exploratory analysis on data collected in the Software Engineering Laboratory, a joint effort of NASA/Goddard Space Flight Center, Computer Sciences Corporation and the University of Maryland, which seeks to characterize and evaluate various models, metrics and software engineering practices to improve our understanding and management of both the software development process and the product. An attempt is made to find a model for effort as a function of various variables. This study also reexamines some of the relationships derived in an earlier study in the SEL by [Basili & Freburger 81] based on fewer data points.

2. BACKGROUND

Data were collected in the Software Engineering Laboratory from ground support software projects at NASA Goddard Space Flight Center. These projects were designed for similar applications. The code is written mostly in FORTRAN except for a small percentage written in macro assembler. Three sets of data were used in this study. One set (DS1) contained 23 data points. The other two sets were subsets of this. One of them (DS2) contained projects under 50 K lines of code. It had 15 data points. The other (DS3) contained projects consisting of 50 K or more lines of code. It had eight data points. One of the original projects included in DS3 was eliminated because it involved an unusually large amount of reused code. It was replaced by a large project which actually consisted of eight of the smaller projects in DS1. Appendix 2 shows the data used in each set.

Effort in this study is expressed in terms of staff-months. It consists of total programmer and management time for a given project. One staff-month of effort is defined as 160 staff-hours. Equations were derived with effort as the response variable. A list of the acronyms used is presented in Appendix 1. Table 1 shows the list of independent variables considered for regression. Definitions of terms used in the SEL may be found in [SEL-82-105] and the most important ones follow:

1. The total number of *lines of code* is the total number of lines of source code generated as a deliverable item for a project. It includes all executable, nonexecutable, and comment statements, whether newly coded or obtained from existing programs and library routines.
2. The number of *new lines* of code is the total number of lines of source code written by programmers for a given task. It excludes code taken from previously existing programs, but it includes comments, executable and non-executable statements.
3. The number of *modified lines* of code is the number of lines of previously developed code that has been changed for reuse in a new system.
4. The number of *developed lines* of code is defined in [Bailey & Basili 81] as the number of new lines of code plus twenty percent of the number of reused lines of code. System integration and full system test are accounted for by the 20% overhead.
5. The total number of *modules* in a project is the number of independently compilable units such as FORTRAN functions, subroutines and BLOCK DATA, or separately identifiable and retrievable components from an on-line library.
6. The number of *new modules* is the number of modules that are not reused from some previous project.
7. The number of *modified modules* is the number of previously developed modules that has been modified in some way for reuse in a new system.
8. A *component* is a named piece of a system. Examples are a separately compilable function, a functional subsystem, or a shared section of data such as a COMMON block.
9. The number of *computer runs* is determined by the computer accounting systems and includes every job submittal for batch systems and every terminal sign-on for interactive systems.
10. The number of *pages of documentation* consists of written material, excluding source code statements and comments embedded therein, that describes a

system or any of its components. It includes the program design document, development plan, test plans, system description, module descriptions and user's guide.

Newmodsq (*newmods*^{2}) and newratio (newlines / newmods) were variables suggested by [Boydston 84]. They were found to have some significance on effort in studies made at the IBM Santa Teresa Laboratory.

The stepwise regression procedure of the Statistical Analysis System (SAS) package was initially used. More specifically, the maximum R^2 improvement (MAXR) technique was used for exploratory analysis. From the results, further analysis was done using the general linear models (GLM) procedure, and some plots were generated for single independent variable models that would possibly be useful in predicting effort during the early stages of development or models that showed a strong correlation between effort and the independent variable whose value cannot be determined early in the development. The latter cannot be used for resource estimation purposes but may be useful in characterizing effort in the environment.

The maximum R^2 improvement technique is considered almost as good as all possible regressions. [SAS 82]. It tries to find the best one-variable model, the best two-variable model, and so on. Initially, it finds the one-variable model that yields the highest value for R^2. Another variable which gives the greatest increase in R^2 is added. After obtaining the two-variable model, each of the variables in the model is compared to each variable that is not in the model. For each comparison, the technique decides if deleting a variable and replacing it with the other variable results in an increase in R^2. When all possible switches have been compared, the one producing the maximum increase in R^2 is made. At this point, comparisons are made again. This continues until the technique can no longer find any switch that could increase R^2. Therefore the two-variable model generated is considered the best that the maximum R^2 improvement technique can find. One more variable is then added to the model, and the comparing-

Table 1 - Variables Considered for Determining Effort

number of developed lines of code
number of pages of documentation
number of modified lines of code
number of modified modules
number of new lines of code
number of new modules
ratio of new lines of code to new modules
number of source code changes (versions)
number of components
number of computer runs
total number of lines of code
total number of modules

and-switching process is repeated until the best three-variable model is found. When there are no more variables that can be added to the model to increase the value of R^2, the procedure stops.

3. ANALYSES AND RESULTS

Each data set was given several sets of candidate independent variables to be used in generating a model. The response variable used was effort.

For each set of candidate independent variables, the following steps were taken.

1. Run STEPWISE/MAXR to generate the best n-variable model, n = 1,2,...
2. Leave out of further consideration those models with significance probability (Prob>F) > 0.05. Only consider those models where (Prob>F) <= 0.05 for the entire model and for each of the independent variables included in the model.
3. Disregard models where the ratio of the number of data points to the number of independent variables is less than 5 for DS1 and DS2. For DS3, because of the limited number of data points, consider models with up to 3 independent variables.
4. Disregard models with $R^2 < 0.5$. Preferably, R^2 should be $>= 0.7$ so that the model accounts for at least 70% of variation of effort in the model.
5. Disregard models with n variables where the increment of R^2 over that of the model with n-1 variables is very small. Do this for i = 2,...,n.

Across sets of candidate independent variables

1. Avoid models with higher-ordered terms.
2. Select model from the set with the greatest number of candidate independent variables originally supplied.
3. Select the model with the highest value of R^2 for the smallest number of variables.

The general linear models (GLM) procedure of SAS was used to examine in closer detail some of the more interesting one-variable models for each data set. Overlaid plots of predicted and actual values of effort versus the independent variables were generated. Plots of the residuals versus the independent variables were also produced.

3.1. All Projects

The first data set consists of 23 projects ranging in size from 2.1 KLOC to 111.9 KLOC. The mean is 33.3 KLOC and the standard deviation is 32.5 KLOC. The number of modules ranges from 23 to 535 with a mean value of 198 and a standard deviation of 172. Effort for these projects ranges from 2.4 to 121.7 staff months. Mean effort is 40.9 staff-months and the standard deviation is 40.4 staff-months. Because the ranges are wide and the standard deviations are large, we subsequently formed the two smaller data sets and analyzed them separately.

Of all the sets of candidate independent variables used to generate a model of effort for this data set, only those which gave reasonable and interesting results are presented here. Initially, the set of candidate independent variables consisted only of newlines, newmods, modlines, and modmods. These are analogous to variables used by [Boydston 84]. They can be determined in the early stages of project development and may therefore have predictive value. The one-variable equation that resulted is

$$Effort = 5.497 + 1.500\ newlines \tag{1}$$

$$R^2 = 0.795 \quad F = 81.65 \quad Prob > F = 0.0001$$

The standard error of estimate (SEE) for the slope of this equation is 0.166. Adding newratio to the set of candidate independent variables yielded the same result. Figure 1 shows a plot of actual effort versus newlines for the different projects. The letters in the figure represent the different projects. Some observations are hidden due to overlap in values. The figure also shows the corresponding points predicted by equation (1). These are represented by asterisks (*) in the plot. Figure 2 is a plot of the residuals versus newlines for this equation. As in Figure 1, the letters represent the different projects in the data set.

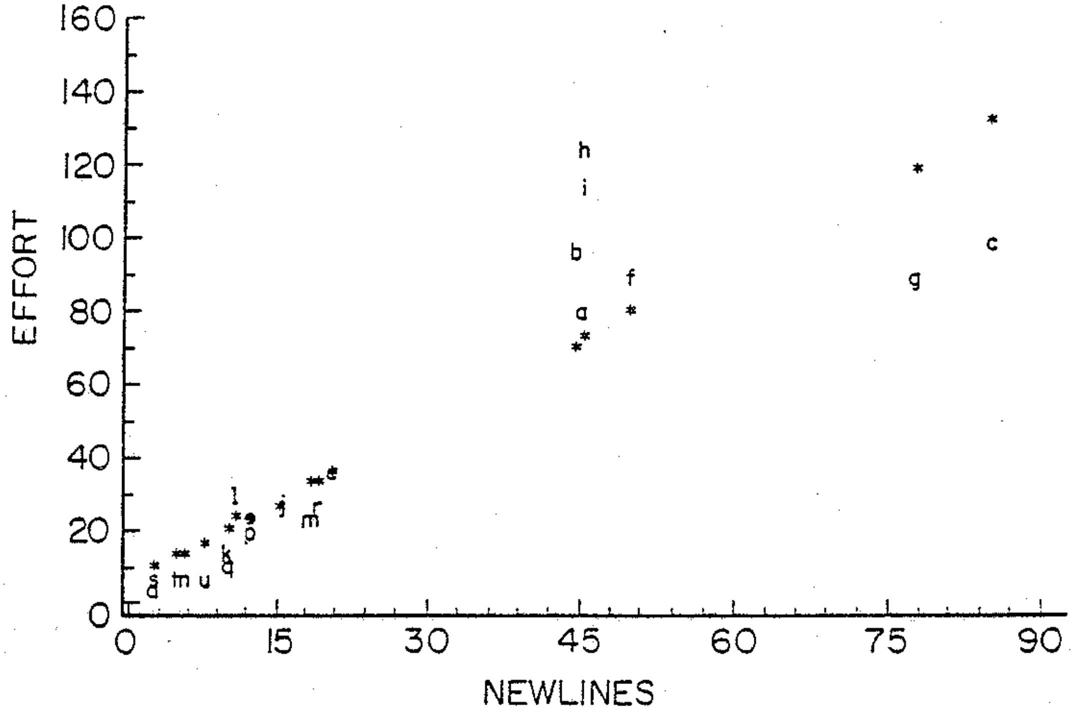
These plots show a few points for which there is a large discrepancy between the actual and the predicted values of effort. Projects *h* and *i* for which the equation underpredicts the value of effort were both developed when there was a major change in the environment. A more reliable machine and more computer terminals were installed. There was quicker turnaround. The staff were turned loose on the computer. However, the level of experience of the staff for both these projects was lower than for most of the other projects in the study. Project *h* was a problem project. It did not have enough experienced staff to begin with and staffing adjustments had to be made in midstream. It was very late compared to other projects and was undertested. Project *i* was also a potential problem project, but was given more attention because of the unhappy experience with project *h* and was fortunately straightened out sooner. Projects *c* and *g* for which the equation (1) overpredicts effort were both developed with more experienced staff.

Because the number of developed lines was originally found by [Bailey & Basili 81] to be the best predictor of effort in their meta-model for the SEL, it was added to the set of candidate independent variables. The resulting equation is

$$Effort = 4.372 + 1.430\ devlines \tag{2}$$

$$R^2 = 0.808 \quad F = 88.30 \quad Prob > F = 0.0001$$

The SEE for the slope of this equation is 0.152. Figures 3 and 4 show the plot of actual and predicted values of effort versus developed lines and the plot of residuals versus



Notes:

1. Actual effort vs. newlines---symbol used is letter code of project.
2. Predicted effort vs. newlines---symbol used is *.
3. 9 observations hidden.

Figure 1 - Effort vs. newlines for DS1.

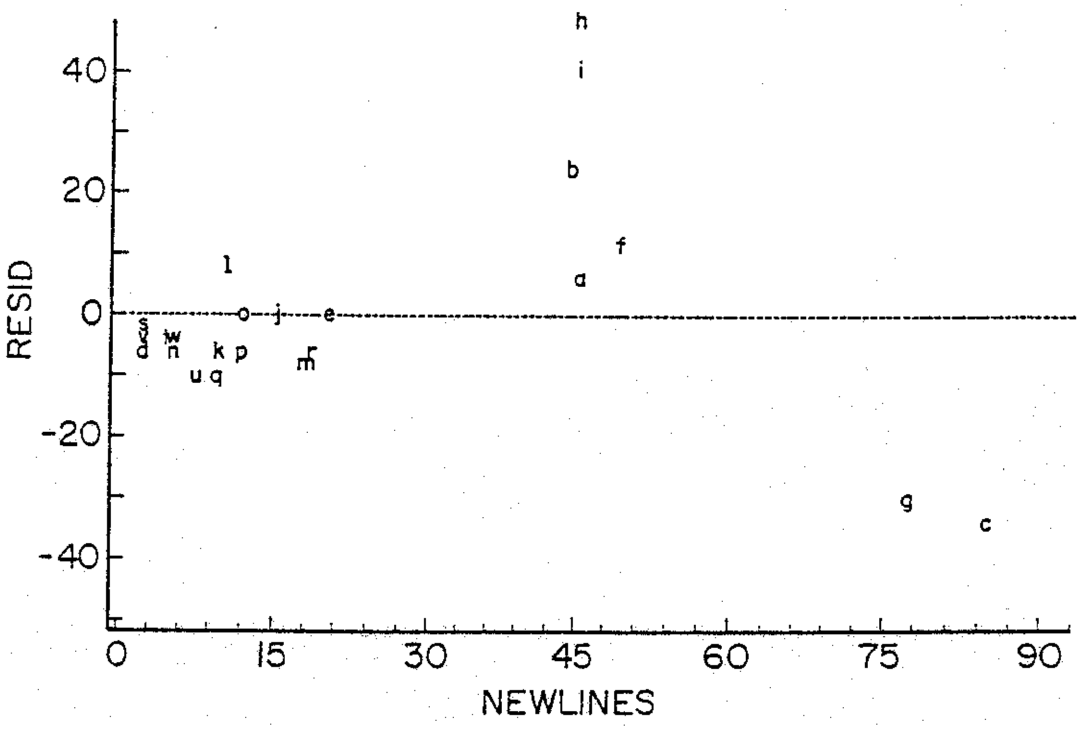


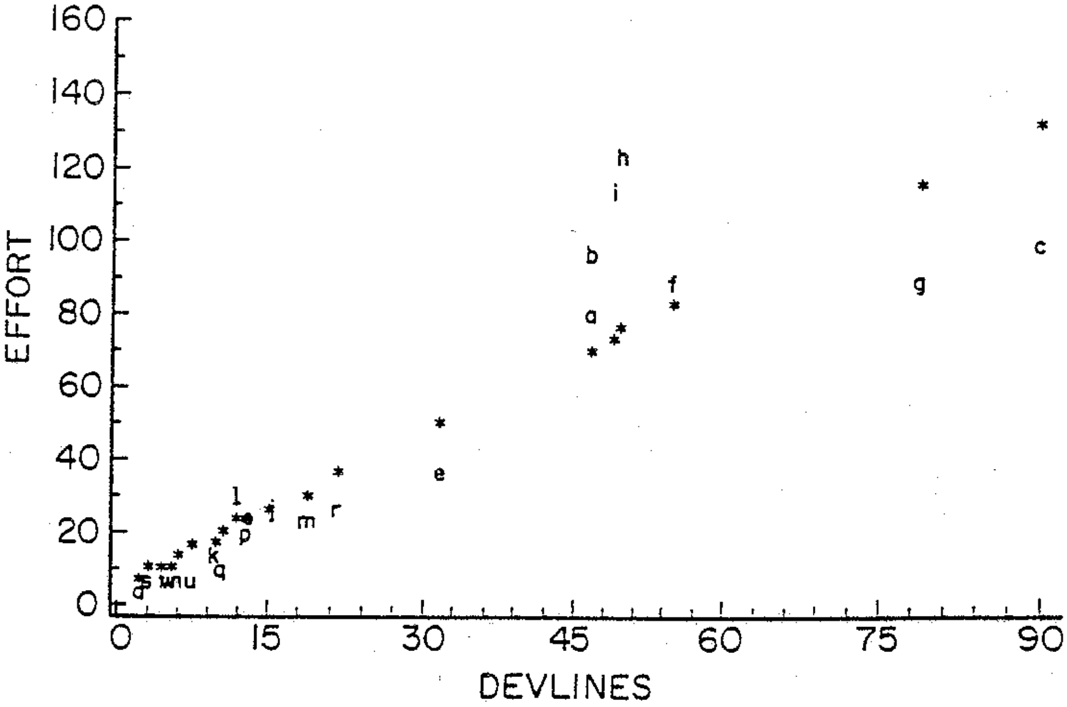
Figure 2 - Effort residuals vs. newlines for DS1.

developed lines respectively for equation (2). The outlier points are the same as those for equation (1). There is little difference in these two models and they indicate that the number of developed lines is at least as good as the number of new lines for predicting effort in the SEL.

Another set of models was generated using newlines, newmods, modlines, modmods and the squares of each of these. [Boydston 84] found that there is a quantified square root trade-off between the number of new modules and the amount of new code per module and thus included a $newmods^2$ term in one of his effort equations. We sought to discover whether or not the inclusion of the squared terms would have any effect on the effort model. The one-variable model that resulted is the same as (1) above. Two and three variable models were also generated as follows:

$$Effort = -11.938 + 3.427\ newlines\ - 0.025\ newlinsq \qquad (3)$$

$$R^2 = 0.916 \quad F = 109.70 \quad Prob > F = 0.0001$$



Notes:

1. Actual effort vs. devlines---symbol used is letter code of project.
2. Predicted effort vs. devlines---symbol used is *.
3. 4 observations hidden.

Figure 3 - Effort vs. devlines for DS1.

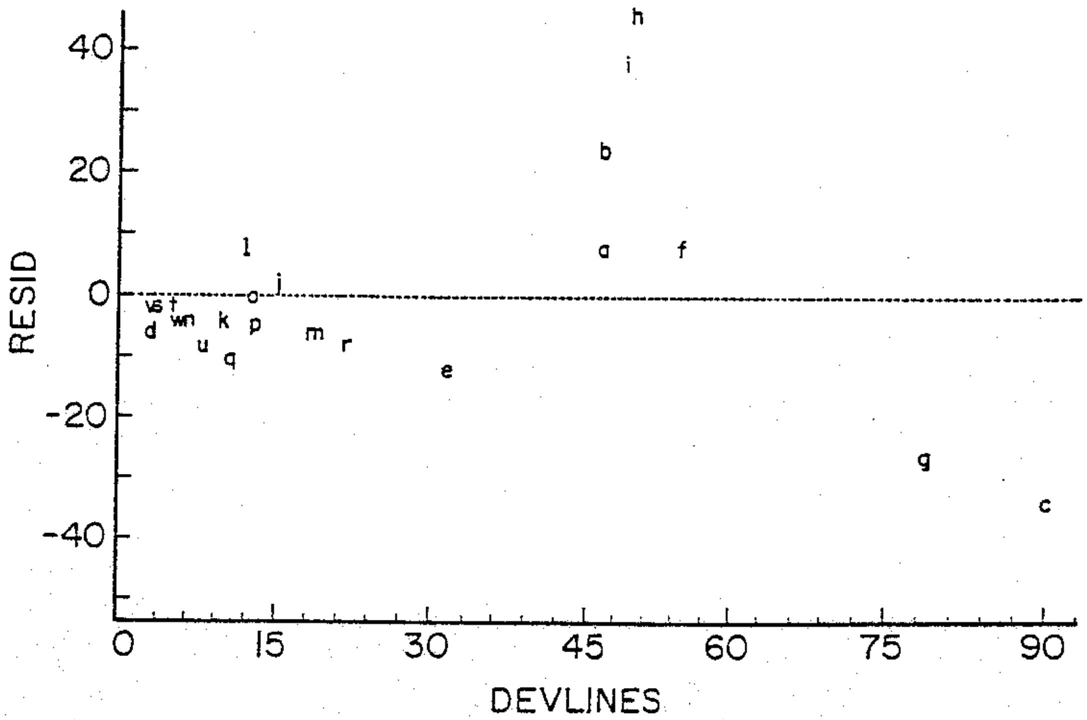


Figure 4 - Effort residuals vs. devlines for DS1.

$$\mathit{Effort} = -13.740 + 3.258\ \mathit{newlines} + 0.355\ \mathit{modmods} - 0.028\ \mathit{newlinsq} \qquad (4)$$

$$R^2 = 0.933 \quad F = 88.53 \quad \mathit{Prob} > F = 0.0001$$

There is a substantial increase in R^2 in going from equation (1) to (3), but not from (3) to (4). This suggests that the quadratic equation (3) may be a much better model than (1), but the inclusion of the additional term modmods does not improve the model tremendously and only adds to the complexity.

Similarly adding the square of developed lines to equation (2) to parallel equation (3) yields the following model:

$$\mathit{Effort} = -10.588 + 2.992\ \mathit{devlines}\ - 0.020\ \mathit{devlinsq} \tag{5}$$

$$R^2 = 0.900\quad F\ = 89.81\quad \mathit{Prob} > F\ = 0.0001$$

This result shows a considerable improvement in R^2 over equation (2). Comparing equations (3) and (5) again shows little difference in the predictive power of new lines and developed lines.

In this study, in addition to predictive models of effort, we also sought relationships between effort and other variables in our database. Models for characterizing and evaluating effort could enhance our understanding of the software development process in our environment. A set of models using as candidate independent variables all the variables in Table 1 with the exception of newratio was generated. Newratio was already found to be insignificant so it was excluded. Models with up to three variables were reasonable in this case and are presented here.

$$\mathit{Effort} = 9.951 + 0.008\ \mathit{numruns} \tag{6}$$

$$R^2 = 0.895\quad F\ = 179.30\quad \mathit{Prob} > F\ = 0.0001$$

$$\mathit{Effort} = 3.384 + 0.104\ \mathit{newmods}\ + 0.006\ \mathit{numruns} \tag{7}$$

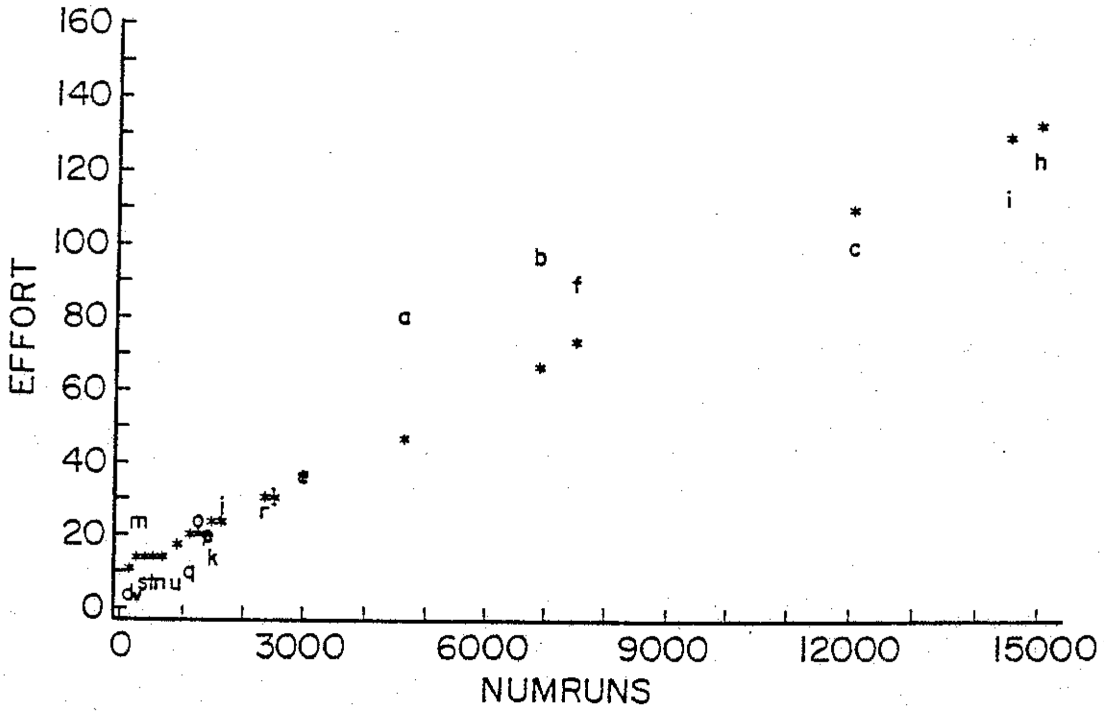
$$R^2 = 0.939\quad F\ = 154.57\quad \mathit{Prob} > F\ = 0.0001$$

$$\begin{aligned}\mathit{Effort} = 4.484 - 0.637\ \mathit{modmods}\ &+ 0.963\ \mathit{devlines} \\ &+ 0.007\ \mathit{numruns}\end{aligned} \tag{8}$$

$$R^2 = 0.978\quad F\ = 285.34\quad \mathit{Prob} > F\ = 0.0001$$

The one-variable model shows a very strong relationship between the number of runs in the project and the amount of effort. Numruns, by itself, accounts for 90% of the variation in effort. The SEE for the slope of equation (6) is 0.0006. Figures 5 and 6 show the plots of actual and predicted effort versus number of runs and residuals versus number of runs respectively for equation (6). For project *a* there is more actual effort per number of runs. It is one of the earliest projects included in this study. The developers were relatively inexperienced. This project was also characterized by staffing and management problems early in the project and serious staffing changes. On the other hand, project *b* was developed by experienced staff. In this case, the actual effort is also higher than that predicted by the equation, probably because the developers were more thorough and purposely put in more effort per run.

To see what other variables correlate well with effort, the number of runs was excluded from the set of candidate independent variables. The following one-variable and four-variable models were generated:



Notes:

1. Actual effort vs. numruns---symbol used is letter code of project.
2. Predicted effort vs. numruns---symbol used is *.
3. 5 observations hidden.

Figure 5 - Effort vs. numruns for DS1.

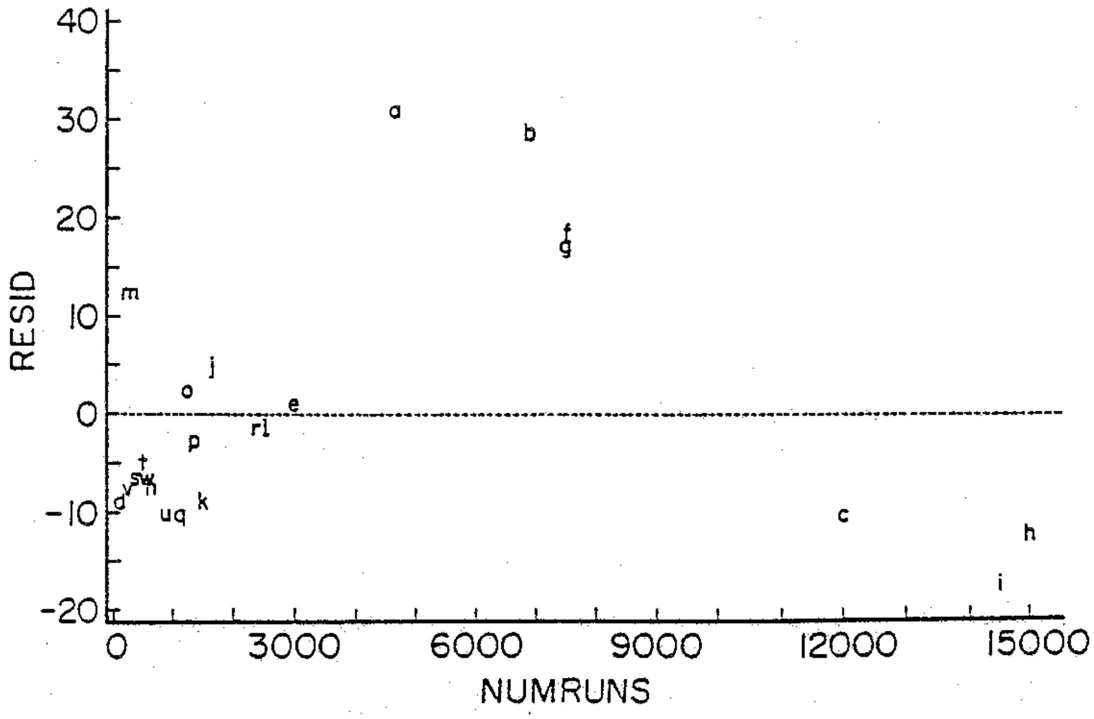


Figure 6 - Effort residuals vs. numruns for DS1.

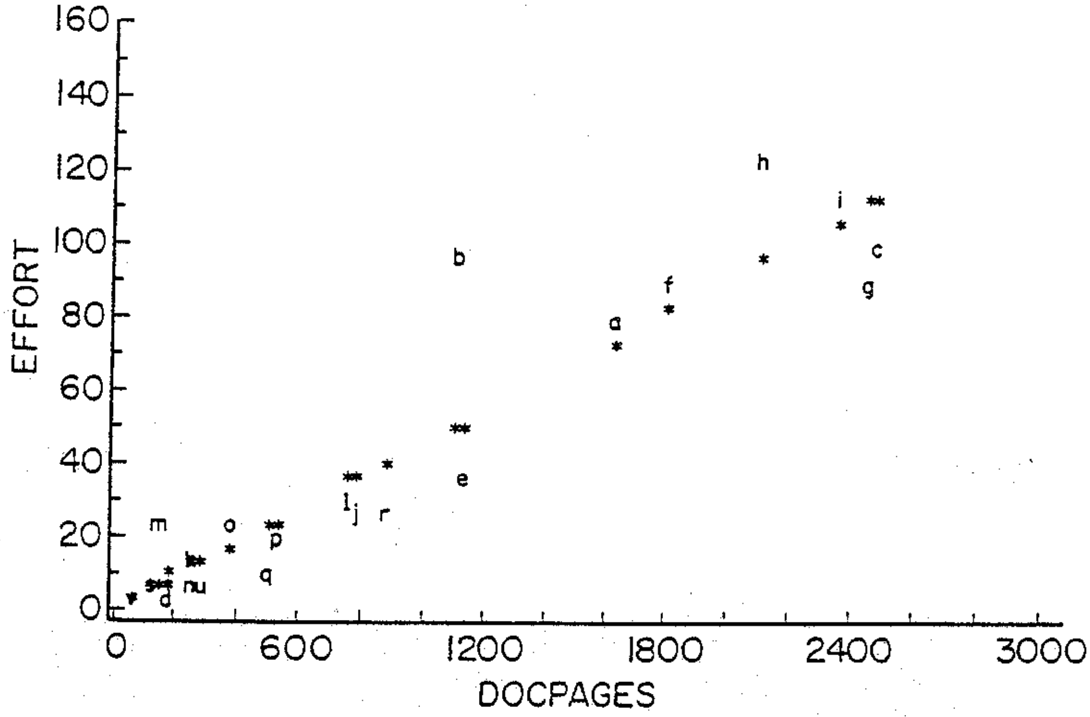
$$Effort = 0.581 + 0.045\ docpages \tag{9}$$

$$R^2 = 0.871 \quad F = 141.82 \quad Prob > F = 0.0001$$

$$\begin{aligned} Effort = 2.634 - 0.346\ newmods\ - 5.076\ modlines \\ + 0.045\ docpages\ + 0.066\ numchngs \end{aligned} \tag{10}$$

$$R^2 = 0.930 \quad F = 59.99 \quad Prob > F = 0.0001$$

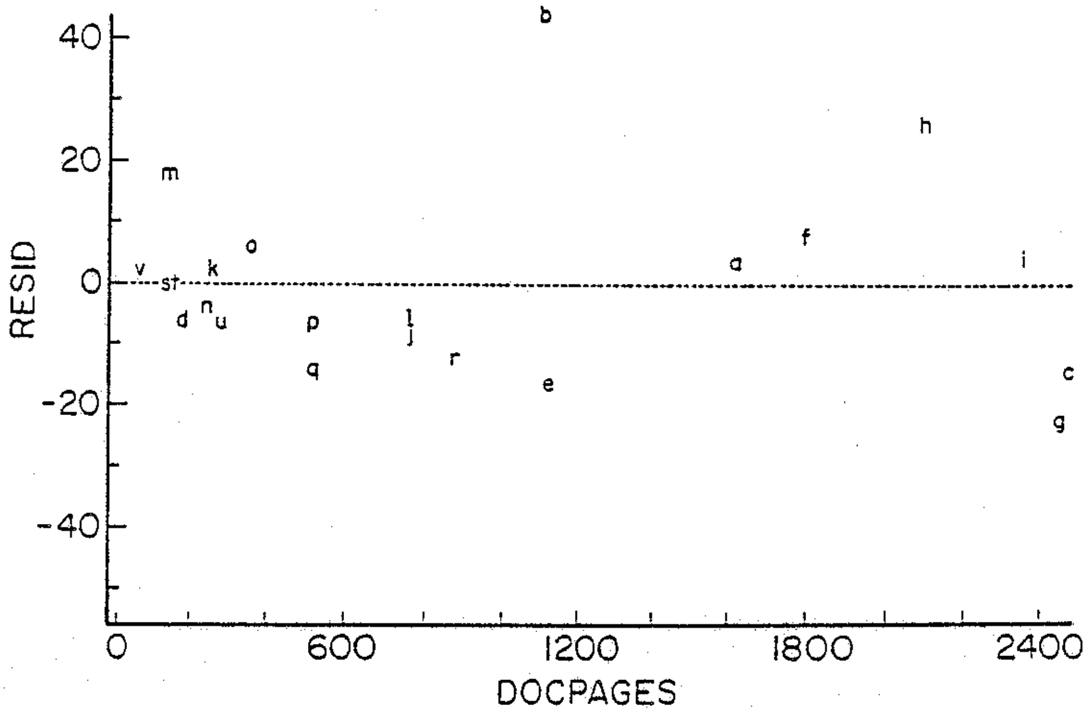
The one-variable equation shows the strong relationship that characterizes effort across these projects and pages of documentation. Like number of runs, however, this relationship cannot be used for predictive purposes since the number of pages of documentation cannot really be determined early in the project. The SEE for the slope of equation (9) is 0.0038. Figures 7 and 8 show plots of effort and residuals versus pages of documentation respectively for equation (9).



Notes:

1. Actual effort vs. docpages---symbol used is letter code of project.
2. Predicted effort vs. docpages---symbol used is *.
3. 3 observations hidden.

Figure 7 - Effort vs. docpages for DS1.



Note: 1 observation hidden.

Figure 8 - Effort residuals vs. docpages for DS1.

Because of the high correlation of both number of runs and pages of documentation with effort, we investigated whether or not a predictor variable like the number of developed lines could be used to predict their values. If we could determine ahead of time the number of runs a project would entail, we could use this information to allocate computer time. Similarly, if we could obtain a good estimate of the number of pages of documentation during the early stages of project development, we could get the publications group ready. We obtained the following result for the number of runs:

$$Numruns = -108.274 + 150.879 \; devlines \qquad (11)$$

$$R^2 = 0.686 \quad F = 45.90 \quad Prob > F = 0.0001$$

The SEE for the slope of equation (11) is 22.269. We can see from this that the number of developed lines does not explain more than 69% of the variation in number of runs. We obtained a much better result for pages of documentation:

$$Docpages = 99.143 + 30.895\ devlines \tag{12}$$

$$R^2 = 0.892 \quad F = 173.24 \quad Prob > F = 0.0001$$

The SEE for the slope of equation (12) is 2.347. Figure 9 shows a plot of actual and predicted pages of documentation versus number of developed lines based on equation (12). Figure 10 shows the residuals plotted against developed lines. The number of developed lines accounts for almost 90% of the variation in pages of documentation. [Basili & Freburger 81] obtained the following equation from a smaller set of projects in the SEL:

$$Doc = 34.7\ (DL^{0.93)}$$

where

$$Doc = pages\ of\ documentation$$

$$DL = number\ of\ developed\ lines$$

They noted that the relationship is approximately linear and our result tends to support this observation.

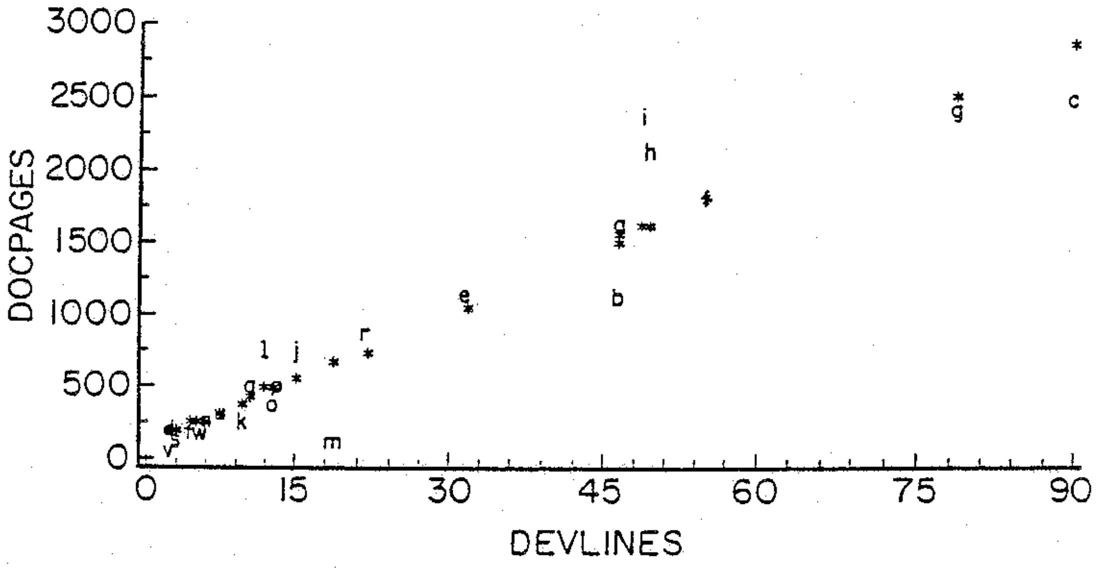
Deleting both numruns and docpages from the independent variable set yields two reasonable models for effort. The one-variable model is the same as equation (2) above. The four-variable model generated is

$$\begin{aligned} Effort = 5.433 - 1.082\ newmods + 22.376\ newlines \\ + 0.854\ totmods - 20.476\ devlines \end{aligned} \tag{13}$$

$$R^2 = 0.926 \quad F = 56.24 \quad Prob > F = 0.0001$$

Many other combinations of variables were used but they either failed to produce any interesting results or they yielded the same results as the case above which included all variables in Table 1 with the exception of newratio. The number of runs is the independent variable most highly correlated with effort when all 23 projects are considered. It may be an excellent measure of the complexity of the project, the quality of the development, the quality of the product, the amount of testing involved, the level of structure or disorganization of project management, and a variety of other factors.

3.2. Projects Under 50 K Lines of Code



Notes:

1. Actual docpages vs. devlines---symbol used is letter code of project.
2. Predicted docpages vs. devlines---symbol used is *.
3. 2 observations hidden.

Figure 9 - Docpages vs. devlines for DS1.

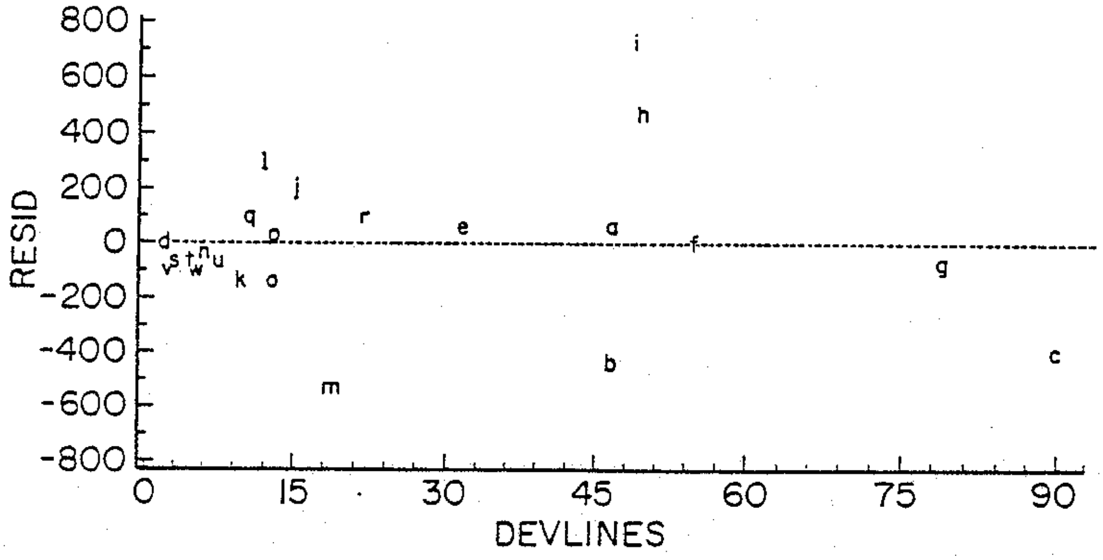


Figure 10 - Docpages residuals vs. devlines for DS1.

There are 15 projects with less than 50 K total lines of code. They range in size from 2.1 to 32.8 KLOC with a mean of 11.9 KLOC and a standard deviation of 8.1 KLOC. There are from 23 to 263 modules. The mean number is 91 and the standard deviation is 63. Effort ranges from 2.4 to 29.0 staff-months with a mean of 14.6 and a standard deviation of 9.5.

In all cases where the number of new lines was included in the set of candidate independent variables, the following model was generated:

$$Effort = 0.877 + 1.535\ newlines \qquad (14)$$

$$R^2 = 0.802 \quad F = 52.71 \quad Prob > F = 0.0001$$

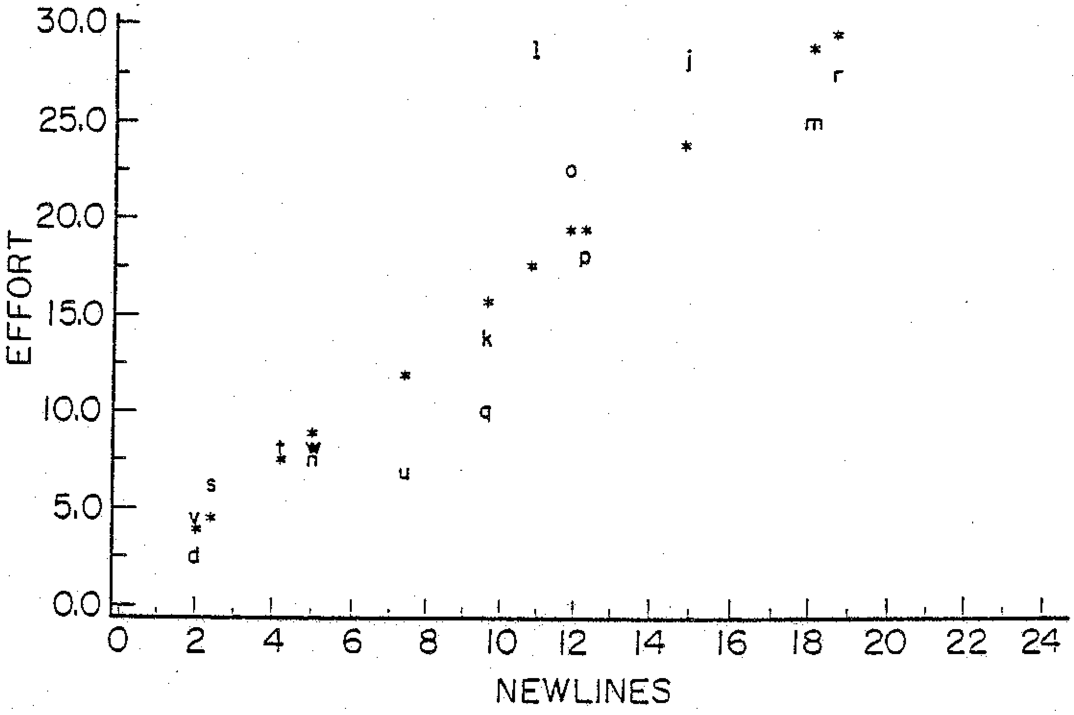
This is so even where the number of runs was included. This equation is selected by the STEPWISE/MAXR technique as the best single-variable effort equation for the smaller projects. It has predictive power since the number of new lines can be estimated early in the development of the project. The SEE for equation (14) is 0.211. The plots of effort versus new lines and residuals versus newlines for this equation are shown on Figures 11 and 12 respectively.

Where newlines was not included in the set of candidate independent variables, the number of developed lines was selected by the technique. The equation generated is

$$Effort = 1.013 + 1.423\ devlines \qquad (15)$$

$$R^2 = 0.797 \quad F = 50.92 \quad Prob > F = 0.0001$$

The SEE for the slope of this equation is 0.199. Figures 13 and 14 respectively show plots of effort and residuals versus the number of developed lines. They are very similar to Figures 11 and 12. In the absence of new lines or developed lines, the number of total lines was selected as the predictor variable.



Notes:

1. Actual effort vs. newlines---symbol used is letter code of project.
2. Predicted effort vs. newlines---symbol used is *.
3. 2 observations hidden.

Figure 11 - Effort vs. newlines for DS2.

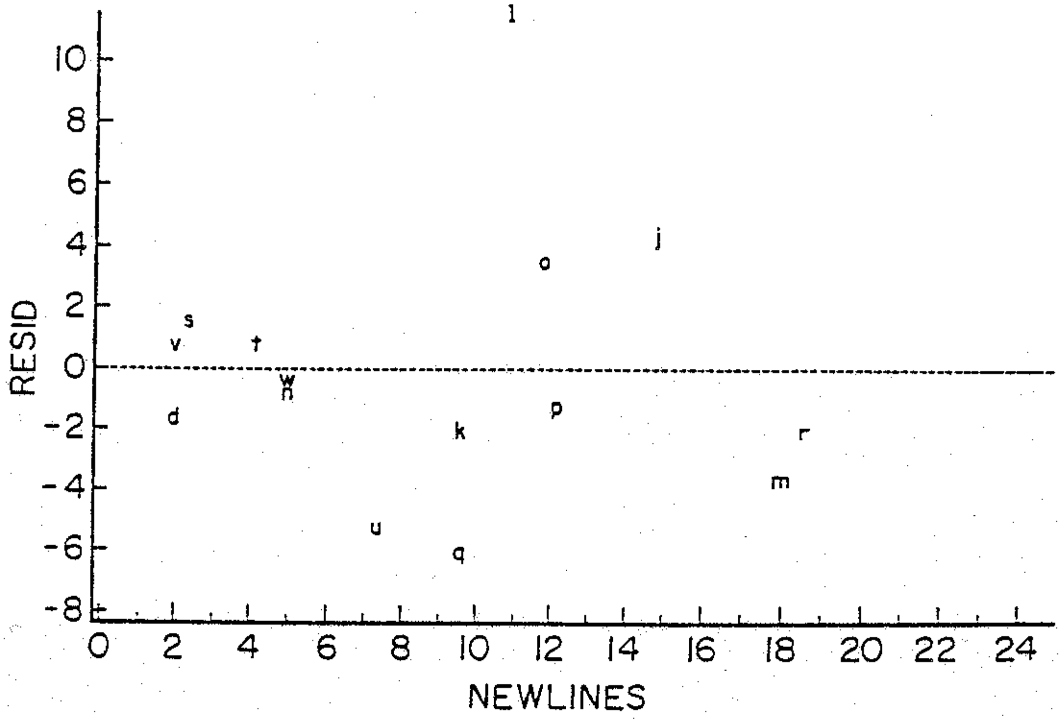
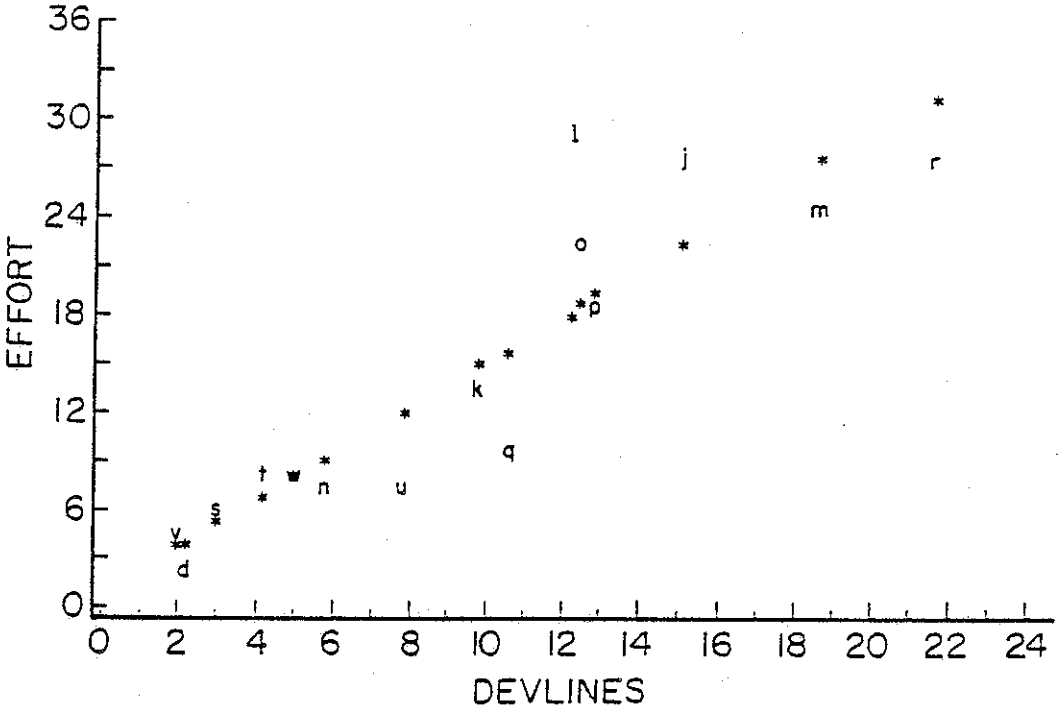


Figure 12 - Effort residuals vs. newlines for DS2.



Notes:

1. Actual effort vs. devlines---symbol used is letter code of project.
2. Predicted effort vs. devlines---symbol used is *.

Figure 13 - Effort vs. devlines for DS2.

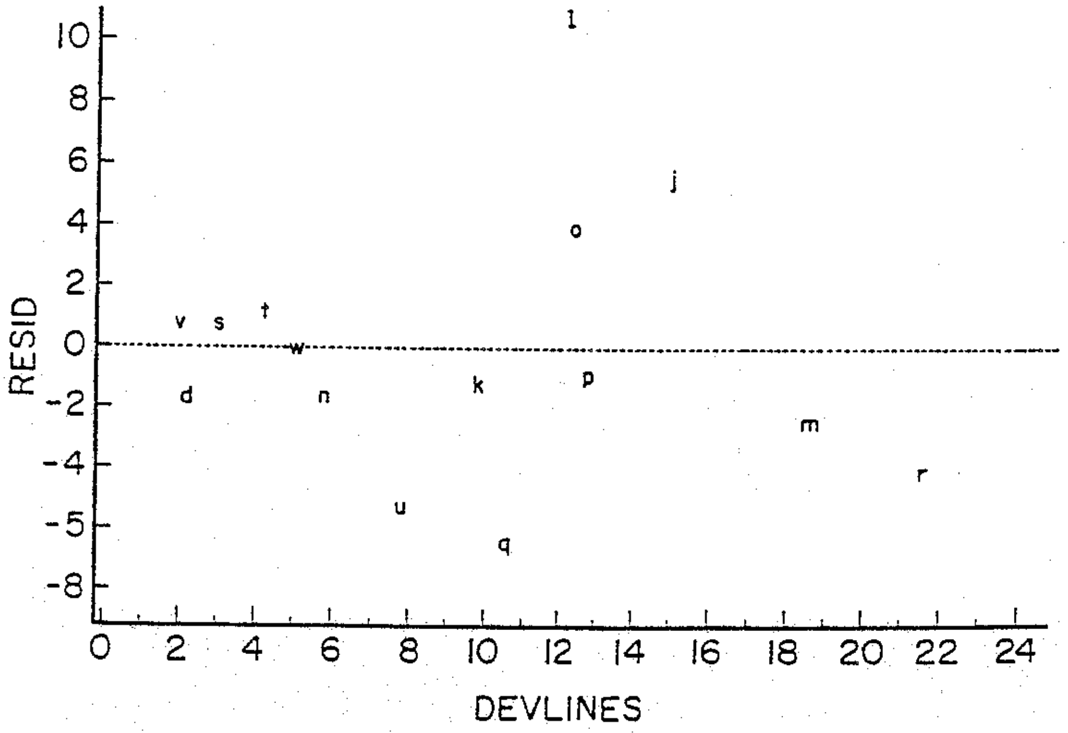


Figure 14 - Effort residuals vs. devlines for DS2.

The two-variable equation generated in most attempted cases is

$$\mathit{Effort} = -1.185 + 0.108\ \mathit{newmods} + 0.009\ \mathit{numruns} \qquad (16)$$

$$R^2 = 0.890 \quad F = 48.53 \quad \mathit{Prob} > F = 0.0001$$

The number of runs entered the model and the number of new modules replaced the number of new lines. None of the other models generated were reasonable.

3.3. Projects with 50 K Lines of Code or More

There are 8 projects in this data set. The smallest consists of 50.9 KLOC and the largest is 111.9 KLOC. The mean size is 75.2 KLOC with a standard deviation of 19.9 KLOC. The number of modules ranges from 201 to 604 with a mean value of 427 and a standard deviation of 139. Effort ranges from 78.7 to 121.7 staff-months with a mean of 97.7 and a standard deviation of 13.5.

In all cases where the number of runs was included in the set of candidate independent variables, the following was generated as the best one-variable model:

$$Effort = 66.868 + 0.003\ numruns \tag{17}$$

$$R^2 = 0.878 \quad F = 43.27 \quad Prob > F = 0.0006$$

The standard error of estimate for the slope of this equation is 0.0005. Figures 15 and 16 respectively show plots of effort and residuals versus the number of runs for the larger projects.

All models which excluded number of runs from the set of independent variables yielded the following as the only reasonably good equation:

$$\begin{aligned} Effort = 122.220 &+ 1.088\ modmods\ - 0.960\ newlines \\ &- 3.883\ modlines \end{aligned} \tag{18}$$

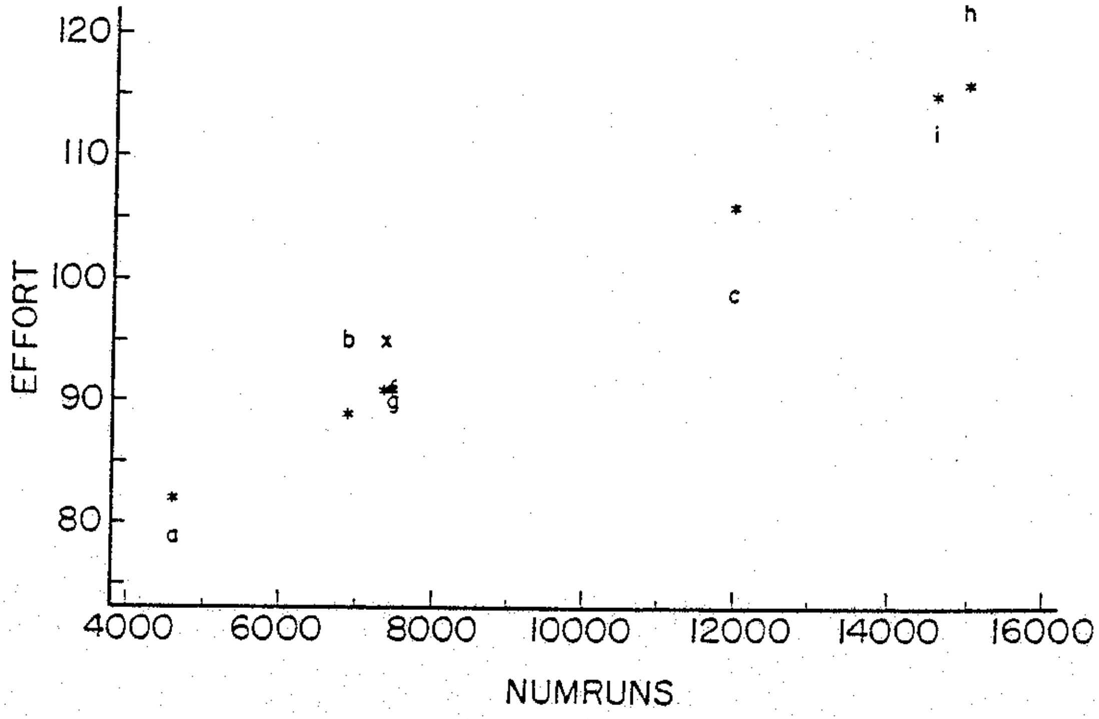
$$R^2 = 0.917 \quad F = 14.78 \quad Prob > F = 0.0125$$

For the larger projects, the number of modified modules is always the first independent variable selected for entry into the model. It seems to be a better predictor of effort for this environment than lines of code, whether new, developed or total, but not much better. It explains only 17% of the variation in effort. It seems that none of the variables which can be determined early in project development is a good single predictor of effort in larger projects.

There were no good two-variable models generated. Considering there are only 8 data points, we should exercise caution in using equation (18) for predictive purposes.

4. SUMMARY

As was shown in [Bailey & Basili 81], developed lines of code is a good overall predictor of effort across all the projects considered in this study. It is one of the variables that can be estimated early in the project development and can thus be used to predict the effort requirements. For projects under 50 KLOC, the number of new lines was found to be the most significant predictor of effort whenever it was included in the set of candidate independent variables. The number of developed lines similarly predicts effort well for the smaller projects. The number of modified modules was not found to be most significant as a single predictor of effort in the Software Engineering Laboratory data. This differs from the result obtained by [Boydston 84] in the IBM Santa Teresa Laboratory environment. Although the amount of code modification in the SEL was by no means small, it probably was not sufficient to show significance. For the projects that



Notes:

1. Actual effort vs. numruns---symbol used is letter code of project.
2. Predicted effort vs. numruns---symbol used is *.
3. 1 observation hidden.

Figure 15 - Effort vs. numruns for DS3.

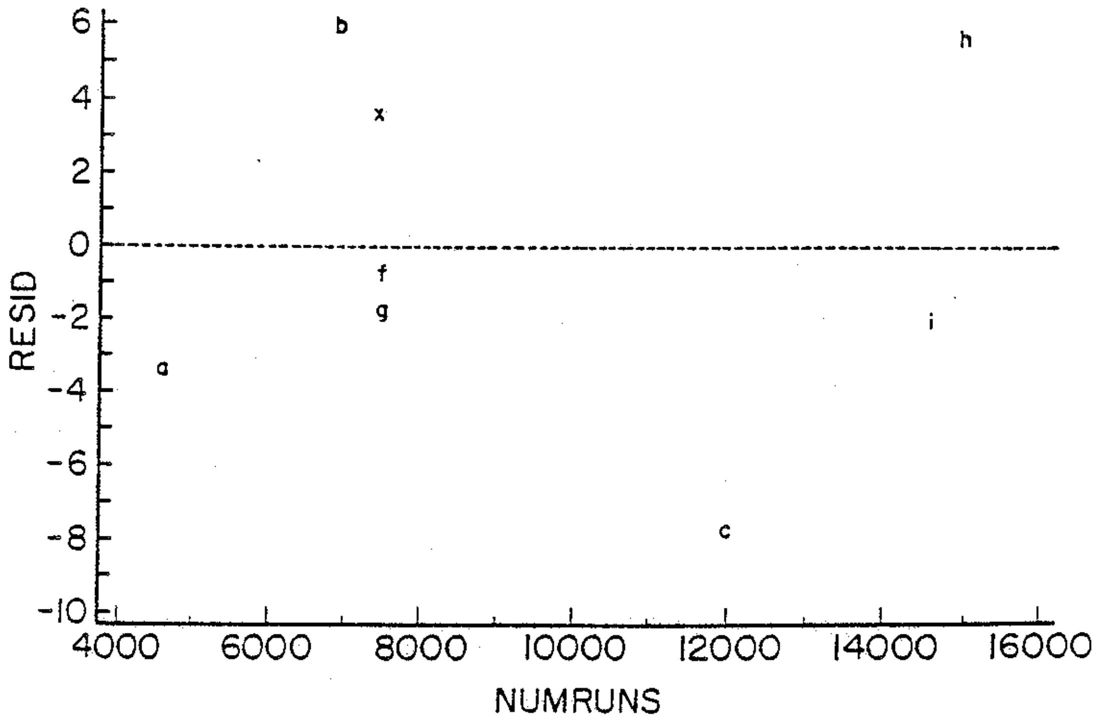


Figure 16 - Effort residuals vs. numruns for DS3.

contained 50 KLOC or more, the number of modified modules is a better single predictor of effort than any of the line measures. It produced a better model of effort than newlines, devlines, modlines or totlines. However, it only accounts for 17% of the variation in effort and is clearly not by itself a good predictor of effort.

There is a high correlation between effort and number of runs overall and in the dataset containing projects that are at least 50 KLOC. There is also a high correlation between effort and pages of documentation across all the projects included in this study. Although these variables cannot be determined in the early stages of project development and therefore cannot be used for predictive purposes, they are nevertheless valuable for explaining and evaluating effort requirements in a project.

Only linear models, for the most part, were attempted for the sake of simplicity in this exploratory study. It would be premature at this point to select one of the above models as the best one to characterize, evaluate or predict effort. The selected model must be subjected to some test of stability. In this study, we were unable to find suitable substitutes for lines of code in a baseline equation to predict effort. However, this study does give us some indication of other variables in the SEL database which are highly correlated with effort. From the non-predictive models generated, we can obtain valuable insight into the software development process in our environment.

5. RESEARCH DIRECTIONS

It may be worthwhile to investigate relationships that may exist among the other variables available in the SEL database. Such relationships may improve our understanding of the software product and its underlying development process. For example, the number of runs could be further examined to see if the data sets get partitioned into projects with high frequency of runs and projects with low frequency of runs (complex versus simple, under-managed versus well-managed, well-tested versus little-tested, etc.).

Nonlinear models could also be explored for these data sets. It is possible that the multi-variable models are curvilinear or ellipsoid rather than planar. The presence of negative intercepts may be due to a poor fit.

Other approaches to the basic meta-model [Bailey & Basili 81] could be investigated. Factor analysis could be used to isolate groups of environmental attributes that work together and to obtain good independent variables that could be used in multiple regression analysis. The principal components technique could also be used instead of factor analysis. If it is possible to justify the use of more than one independent variable to predict the value of the dependent variable in the background equation, greater accuracy may be obtained.

ACKNOWLEDGEMENTS

The authors are grateful to Mr. Frank McGarry of NASA/Goddard Space Flight Center and Dr. Jerry Page of Computer Sciences Corporation for their invaluable help in providing the data for this study. They also wish to thank the members of their research group at the University of Maryland and Dr. C. Frank Starmer of Duke University who have provided assistance of one form or another in the preparation of this paper.

REFERENCES

[Bailey & Basili 81]

Bailey, John W. and Victor R. Basili, "A Meta-Model for Software Development Resource Expenditures", Proceedings, Fifth International Conference on Software Engineering, 1981, pp. 107-116.

[Basili & Freburger 81]

Basili, V. R. and K. Freburger, "Programming Measurement and Estimation in the Software Engineering Laboratory", Journal of Systems and Software, Vol. 2, No. 1, 1981, pp. 47-57.

[Boehm 81]

Boehm, Barry W., Software Engineering Economics, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[Boehm 84]

Boehm, Barry W., "Software Engineering Economics", IEEE Transactions on Software Engineering 10, No. 1, 1984.

[Boydston 84]

Boydston, Robert E., "Programming Cost Estimate: Is It Reasonable?", Proceedings of the Seventh ICSE, IEEE Computer Society Press, March 26-29 1984, pp. 153-159.

[Jeffery & Lawrence 79]

Jeffery, D. R. and M. J. Lawrence, "An Inter-organizational Comparison of Programming Productivity", Department of Information Systems, University of New South Wales, 1979.

[Putnam 78]

Putnam, L., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem", IEEE Transactions on Software Engineering 4, No. 4, 1978.

[SAS 82]

SAS Institute Inc., SAS User's Guide: Statistics, 1982 Edition, Cary, NC: SAS Institute Inc., 1982.

[SEL-82-105]

"Glossary of Software Engineering Laboratory Terms", T. A. Babst, F. E. McGarry, and M. G. Rohleder, NASA/Goddard Space Flight Center, October 1983.

[SEL-83-001]

"An Approach to Software Cost Estimation", F. E. McGarry, G. Page, D. N. Card, et al., NASA/Goddard Space Flight Center, February 1984.

[Walston & Felix 77]

Walston, C. E. and C.P. Felix, "A Method of Programming Measurement and Estimation", IBM Systems Journal 16, No. 1, 1977.

[Wolverton 74]

Wolverton, R., "The Cost of Developing Large Scale Software", IEEE Transactions on Computers 23, No. 6, 1974.

Appendix 1 - List of Acronyms

Acronym	Description
devlines	newlines + 0.2 * (totlines - newlines) (KLOC)
devlinsq	$devlines^2$
docpages	number of pages of documentation
modlines	number of modified lines of code (KLOC)
modlinsq	$modlines^2$
modmods	number of modified modules
modmodsq	$modmods^2$
newlines	number of new lines of code (KLOC)
newlinsq	$newlines^2$
newmods	number of new modules
newmodsq	$newmods^2$
newratio	newlines / newmods
numchngs	number of source code changes (versions)
numcomps	number of components
numruns	number of computer runs
projcode	project code
projname	project name
totlines	total number of lines of code (KLOC)
totmods	total number of modules
allhrs	proghrs + mgmthrs + servhrs
mgmthrs	management work time (tenths of an hour)
prmghrs	proghrs + mgmthrs
proghrs	programmer work time (tenths of an hour)
servhrs	services work time (tenths of an hour)
allefrt	allhrs / 1600 (staff-months)
effort	prmghrs / 1600 (staff-months)

Appendix 2 - Data Used in the Analyses

OBS	LETTER	NEWMODS	MODMODS	NEWLINES	MODLINES	PROGHRS	AGNTHRS	SERVHRS	TOTLINES	DOCPAGES	NUMCHNGS
1	a	172	19	45.345	4.673	89115	36765	11090	50.911	1613	1255
2	b	200	21	43.955	3.506	129299	23316	13780	55.237	1104	1649
3	c	346	122	84.729	20.041	128522	29078	43160	111.808	2473	1228
4	d	19	2	2.000	0.486	3457	383	580	2.886	171	65
5	e	92	30	20.075	6.727	41706	16209	10790	75.420	1120	858
6	f	337	31	49.316	4.252	109565	35510	12310	75.393	1793	2107
7	g	418	59	76.883	5.652	116586	27119	27444	85.369	2458	2710
8	h	182	70	45.004	9.705	149476	45273	28462	67.325	2307	2077
9	i	216	65	44.644	8.606	134639	45328	32669	66.266	2360	1575
10	j	93	0	14.873	0.000	31638	13022	11942	15.258	763	255
11	k	45	8	9.627	0.527	16675	4986	5436	10.172	255	219
12	l	74	22	10.822	2.331	34532	11800	6950	17.271	760	541
13	m	216	17	17.999	1.374	37934	1905	539	20.648	140	1274
14	n	51	1	4.959	0.130	11590	720	350	9.004	245	423
15	o	83	14	11.878	1.323	29385	6768	5310	14.765	366	510
16	p	105	16	12.227	1.571	23035	6392	3381	14.361	527	660
17	q	71	7	9.568	0.892	14023	1536	2349	14.292	511	314
18	r	72	29	18.680	7.838	35202	9091	5079	32.822	973	795
19	s	13	15	2.451	1.947	7780	2269	2010	5.497	136	103
20	t	38	0	4.160	0.000	9775	3446	2417	4.525	169	158
21	u	39	17	7.350	2.049	10115	1290	1284	9.727	284	300
22	v	23	0	2.052	0.000	5182	2290	1118	2.052	61	135
23	w	41	0	4.921	0.000	11166	1765	1627	5.204	163	289

OBS	TOTMODS	NUMCOMPS	NUMRUNS	DEVLINES	PERCHRS	ALLHRS	EFFORT	ALLEFRT	NEWLINSQ	MODMODSQ	MODLINSQ	NEWMODSQ	NEWRATIO
1	201	292	4604	46.4582	125880	136970	78.675	85.606	2056.17	361	21.837	29584	0.263634
2	283	355	6871	46.2114	152615	166395	95.384	103.997	1932.04	441	12.292	40000	0.219775
3	510	587	11976	90.1568	157600	200760	98.500	125.475	7179.00	14884	401.642	119716	0.244882
4	24	24	168	2.3772	3840	4420	2.400	2.762	4.00	4	0.236	361	0.105263
5	374	423	3033	31.1440	57915	68705	36.197	42.941	403.01	900	45.253	8464	0.218207
6	535	618	7500	54.5314	145075	157385	90.672	98.366	2432.07	961	18.080	113569	0.146338
7	519	639	7527	78.5802	143705	171149	89.816	106.968	5911.00	3481	31.945	174724	0.183931
8	373	432	15017	49.4682	194749	223211	121.718	139.507	2025.36	4900	94.187	33124	0.247275
9	391	444	14561	49.9684	179967	212636	112.479	132.897	1993.09	4225	74.063	46656	0.206685
10	102	113	1589	14.9500	44660	56602	27.912	35.376	221.21	0	0.000	8649	0.159925
11	55	74	1476	9.7360	21661	27097	13.538	16.936	92.68	64	0.278	2025	0.213933
12	114	161	2467	12.1118	46332	53282	28.957	33.301	117.12	484	5.434	5476	0.146243
13	263	263	270	18.5298	39839	40378	24.899	25.236	323.96	289	1.888	46656	0.083329
14	73	73	590	5.7680	12310	12660	7.694	7.912	24.59	1	0.017	2601	0.097235
15	115	143	1283	12.4554	36153	41463	22.596	25.914	141.09	196	1.750	6889	0.143108
16	116	180	1395	12.7542	29427	32808	18.392	20.505	149.50	256	2.468	11025	0.116448
17	100	113	1151	10.5108	15559	17908	9.724	11.192	91.55	49	0.796	5041	0.134761
18	148	245	2354	21.5084	44293	49372	27.683	30.857	348.94	841	61.434	5184	0.259444
19	38	39	332	3.0602	10049	12059	6.281	7.537	6.01	225	3.791	169	0.188538
20	41	44	465	4.2330	13221	15638	8.263	9.774	17.31	0	0.000	1444	0.109474
21	63	114	856	7.3254	11405	12689	7.128	7.931	54.02	289	4.198	1521	0.188462
22	23	35	221	2.0520	7472	8590	4.670	5.369	4.21	0	0.000	529	0.089217
23	48	74	546	4.9776	12931	14558	8.082	9.099	24.22	0	0.000	1681	0.120024

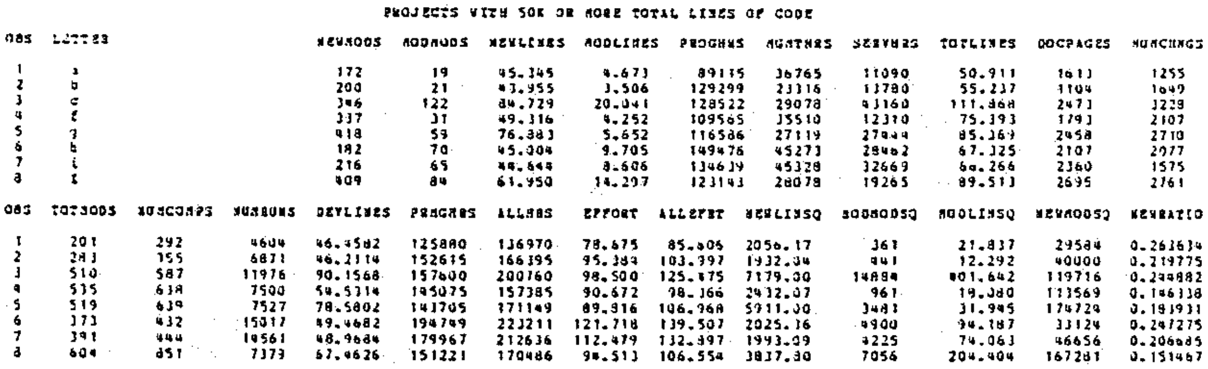
Data Set 1

PROJECTS WITH UNDER 50K TOTAL LINES OF CODE

OBS	LETTER	NEWMODS	MODMODS	NEWLINES	MODLINES	PROGHRS	MGMTHRS	SERVHRS	TOTLINES	DOCPAGES	NUMCHNGS
1	d	19	2	2.000	0.486	3457	383	580	2.886	171	65
2	j	93	0	14.873	0.000	31638	13022	11942	15.258	763	255
3	k	45	8	9.627	0.527	16675	4986	5436	10.172	255	219
4	l	74	22	10.822	2.331	34532	11800	6950	17.271	760	541
5	m	216	17	17.999	1.374	37934	1905	539	20.648	140	1274
6	n	51	1	4.359	0.130	11590	720	350	9.004	245	423
7	o	83	14	11.878	1.323	29385	6768	5310	14.765	366	510
8	p	105	16	12.227	1.571	23035	6392	3381	14.863	527	680
9	q	71	7	9.568	0.392	14023	1536	2349	14.282	511	319
10	r	72	29	18.680	7.838	35202	9091	5079	32.922	873	795
11	s	13	15	2.451	1.947	7780	2269	2010	5.497	136	103
12	t	38	0	4.160	0.000	9775	3446	2417	4.525	169	158
13	u	39	17	7.350	2.049	10115	1290	1284	9.727	284	300
14	v	23	0	2.052	0.000	5182	2290	1118	2.052	61	135
15	w	41	0	4.921	0.000	11166	1765	1627	5.204	163	289

OBS	TOTMODS	NUMCOMPS	NUMRUNS	DEVLINES	PRSGHRS	ALLHRS	EFFORT	ALLEFRT	NEWLINSQ	MODMODSQ	MODLINSQ	NEWMODSQ	NEWRATIO
1	24	24	168	2.1772	3840	4420	2.4000	2.7625	4.000	4	0.2362	361	0.105263
2	102	113	1589	14.9500	44660	56602	27.3125	35.3762	221.206	0	0.0000	8649	0.159925
3	55	74	1476	9.7360	21661	27097	13.5381	16.9356	92.679	64	0.2777	2025	0.213933
4	134	161	2467	12.1118	44332	53282	28.9575	33.3012	117.116	484	5.4336	5476	0.146243
5	261	263	270	18.5288	39839	40378	24.8994	29.2362	323.964	289	1.8879	46656	0.093129
6	73	73	500	5.7680	12310	12660	7.6937	7.9125	24.592	1	0.0169	2601	0.097235
7	115	143	1283	12.4554	36153	41461	22.5956	25.9144	141.087	196	1.7503	6889	0.143108
8	136	180	1395	12.7542	29427	32808	18.3919	20.5050	149.500	256	2.4680	11025	0.116448
9	100	113	1151	10.5108	15559	17908	9.7294	11.1925	91.547	49	0.7957	5041	0.134761
10	148	245	2354	21.5084	44293	49372	27.6811	30.8575	348.942	841	61.4342	5184	0.259444
11	38	39	332	3.0602	10049	12059	6.2806	7.5369	6.007	225	3.7909	169	0.198538
12	41	44	465	4.2330	13221	15638	8.2631	9.7737	17.306	0	0.0000	1444	0.109474
13	63	114	856	7.8254	11405	12689	7.1291	7.9306	54.022	289	4.1984	1521	0.188462
14	23	35	221	2.0520	7472	8590	4.6700	5.3687	4.211	0	0.0000	529	0.089217
15	48	76	546	4.9776	12931	14558	8.0819	9.0987	24.216	0	0.0000	1681	0.120026

Data Set 2



```
                                          PROJECTS WITH 50K OR MORE TOTAL LINES OF CODE

OBS  LETTER                         NEWMODS  MODMODS  NEWLINES  MODLINES   PROGHRS  MGMTHRS  SERVHRS  TOTLINES  DOCPAGES  NUMCHNGS

 1     a                              172       19     45.345     4.673     89115    36765    11090    50.911     1613      1255
 2     b                              200       21     43.955     3.506    129299    23316    13780    55.237     1104      1649
 3     c                              346      122     84.729    20.041    128522    29078    43160   111.868     2473      3229
 4     f                              337       31     49.316     4.252    109565    35510    12310    75.393     1793      2107
 5     g                              418       59     76.883     5.652    116586    27119    27444    85.369     2458      2710
 6     h                              182       70     45.004     9.705    149476    45273    28462    67.325     2107      2977
 7     i                              216       65     44.644     8.606    134639    45328    32669    66.266     2360      1575
 8     k                              409       84     61.950    14.297    123143    28078    19265    89.513     2695      2761

OBS  TOTMODS  NUMCOMPS  NUMRUNS  DEVLINES  PRMGHRS   ALLHRS   EFFORT   ALLEFRT  NEWLINSQ  MODMODSQ  MODLINSQ  NEWMODSQ  NEWRATIO

 1     201      292      4604    46.4582   125880   136970   78.675   85.606   2056.17      361     21.837     29584   0.263634
 2     243      755      6871    46.2114   152615   166395   95.384  103.997   1932.04      441     12.292     40000   0.219775
 3     510      587     11976    90.1568   157600   200760   98.500  125.475   7179.00    14884    401.642    119716   0.244882
 4     535      638      7500    54.5314   145075   157385   90.672   98.366   2432.07      961     18.080    113569   0.146338
 5     519      639      7527    78.5802   143705   171149   89.816  106.968   5911.00     3481     31.945    174724   0.181931
 6     373      432     15017    49.4682   194749   223211  121.718  139.507   2025.36     4900     94.187     33124   0.247275
 7     391      444     14561    48.9684   179967   212636  112.479  132.898   1993.09     4225     74.063     46656   0.206685
 8     604      851      7373    67.4626   151221   170486   94.513  106.554   3837.80     7056    204.404    167281   0.151467
```

Data Set 3

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
TR-1520

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)
Finding Relationships Between Effort and Other Variables in the SEL

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(*s*)
Victor R. Basili
N. Monina Panlilio-Yap

8. CONTRACT OR GRANT NUMBER(*s*)
NASA NSG-5123

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Computer Science
University of Maryland
College Park, MD 20742

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

12. REPORT DATE
July 1985

13. NUMBER OF PAGES
31

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

software development, resource estimation
effort models, stepwise regression

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

Estimating the amount of effort required for a software development project is one of the major aspects of resource estimation for that project. In this study, we examined the relationship between effort and other variables for 23 Software Engineering Laboratory projects that were developed for NASA/Goddard. These variables fell into two categories: those which can be determined in the early stages of project development and may therefore be useful in a baseline equation for predicting effort in future projects, and those which can be used mainly to characterize or evaluate effort requirements and thus enhance our understanding of the software development process in this environment.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)