

ADA041106

RADC-TR-77-175
Final Technical Report
May 1977



STATISTICAL PREDICTION OF PROGRAMMING ERRORS

IBM Corporation

Approved for public release; distribution unlimited.

AD No. _____
DDC FILE COPY

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

ADDC
RECEIVED
JUN 23 1977
R
L
D

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED:

Alan N. Sukert

ALAN N. SUKERT, Captain, USAF
Project Engineer

APPROVED:

Robert D. Krutz

ROBERT D. KRUTZ, Colonel, USAF
Chief, Information Sciences Division

FOR THE COMMANDER:

John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER RADC-TR-77-175	2. GOVT ACCESSION NO. (9)	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) STATISTICAL PREDICTION OF PROGRAMMING ERRORS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Apr 76 - 30 Nov 76	
7. AUTHOR(s) R.W. Motley W.D. Brooks		6. PERFORMING ORG. REPORT NUMBER N/A	
9. PERFORMING ORGANIZATION NAME AND ADDRESS IBM Corporation, Federal Systems Division 1601 North Kent Street Arlington VA 22209 404 022		8. CONTRACT OR GRANT NUMBER(s) F30602-76-C-0213	
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISIM) Griffiss AFB NY 13441 11		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 55811407	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same 1234p.		12. REPORT DATE May 1977	
		13. NUMBER OF PAGES 240	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
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) Same 16 5581 17 11			
18. SUPPLEMENTARY NOTES RADC Project Engineer: Captain Alan N. Sukert (ISIS)			
19. KEY WORD (Continue on reverse side if necessary and identify by block number) Programming Error Prediction Program Structure/Complexity Analysis Program Characteristics Analysis Software Quality Software Reliability Software Error Analysis Multiple Linear Regression			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report presents and discusses the results obtained for statistical predictions of programming errors using multiple linear regression analysis. Programming errors were predicted from linear combinations of program characteristics and programmer variables. Each of the program characteristic variables were considered to be measures of the program's complexity and structure. Two distinct data samples comprising 783 programs with approximately 297,000 source instructions written for command and control software applications. (Cont'd)			

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404022

y/p

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Item 20 (cont)

were analyzed. Background data on both samples is provided which includes discussions related to each sample's software development environment, testing conditions, predictor variables, definition of programming errors, and general data characteristics. Results are presented which give the prediction equations obtained and a discussion of the predictability of errors and error rate in each sample. Conclusions of the study and recommendations for further research are also provided.

ADDITION BY	
DDC	DATE ENTERED ☒
DDC	DATE ENTERED ☐
UNCLASSIFIED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
DDC	AVAIL. RRR/WW SPECIAL
A	

DDC
RECEIVED
JUN 28 1977
R
D

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
1.0 INTRODUCTION	1-1
1.1 Background	1-3
1.2 Purpose	1-4
2.0 DESCRIPTION AND CHARACTERISTICS OF DATA SAMPLES	2-1
2.1 Sample S	2-3
2.1.1 Software Development Environment	2-3
2.1.2 Software Testing Considerations	2-6
2.1.3 Definition, Classification, and Collection of Error Data	2-7
2.1.4 Predictor Variables	2-11
2.1.5 Characteristics of Sample Data	2-13
2.2 Sample T	2-16
2.2.1 Software Development Environment	2-16
2.2.2 Software Testing Considerations	2-17
2.2.3 Definition, Classifications, and Collection of Error Data	2-19
2.2.4 Predictor Variables	2-22
2.2.5 Characteristics of Sample Data	2-24
3.0 PRELIMINARY REVIEW AND ANALYSIS OF DATA SAMPLES	3-1
3.1 Selected Limitations of the Data	3-1
3.2 Preliminary Analysis Findings and Observa- tions	3-4
4.0 ANALYSIS METHOD AND PROCEDURE	4-1
4.1 Multiple Linear Regression Analysis	4-1
4.2 BMDP Stepwise Regression Procedure	4-5
5.0 DEVELOPMENT OF ERROR PREDICTION EQUATIONS	5-1
5.1 Normalization of Predictor Variables	5-1
5.2 Other Transformations to Predictor Variables	5-3
5.3 Combining Predictors in the Equations	5-5
5.4 Selection of Regression Coefficients to be Reported	5-8

TABLE OF CONTENTS (Continued)

	Page
5.5 Regression Analysis Using Standardized Form of the Prediction Equation	5-9
5.6 A Priori Elimination of Predictor Variables	5-12
6.0 ERROR PREDICTION EQUATIONS: RESULTS AND DISCUSSION	6-1
6.1 Results Summary	6-1
6.2 Discussion	6-3
6.3 Sample S Results	6-8
6.3.1 Errors/Program=f(Unnormalized Variables)	6-8
6.3.2 Errors/Program=f(SI+Normalized Variables)	6-21
6.3.3 Error Rate/Program=f(Unnormalized Variables)	6-33
6.3.4 Error Rate/Program=f(SI+Normalized Variables)	6-45
6.3.5 Validation of Prediction Equations for Sample S	6-55
6.4 Sample T Results	6-61
6.4.1 Results for Errors/Program	6-62
6.4.2 Results for Error Rate/Program	6-84
6.4.3 Sample T Prediction Consistency Analysis	6-99
7.0 ADDITIONAL ANALYSIS	7-1
7.1 Error Rate and Programmer Variables	7-1
7.2 Error Rate and Source Instructions	7-5
8.0 CONCLUSIONS AND RECOMMENDATIONS	8-1
8.1 Conclusions	8-1
8.2 Direct Recommendations	8-2
8.3 Recommendations for Further Research	8-6
8.4 Proposed Support Tools and Techniques	8-10
8.5 Summary of Recommendations	8-12
9.0 REFERENCES	9-1

TABLE OF CONTENTS (Continued)

	<u>Page</u>
APPENDIX A - CONSIDERATIONS ON THE MEASURE OF ERROR TO BE ANALYZED IN SOFTWARE ERROR PREDICTION STUDIES	A-1
APPENDIX B - PREDICTOR VARIABLE DESCRIPTIONS, SAMPLE S AND SAMPLE T	B-1
APPENDIX C - LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS AND ERROR RATE	C-1

LIST OF TABLES

<u>Table</u>		<u>Page</u>
2-1	Characteristics of Data Samples	2-2
2-2	Application of New Programming Technologies in Sample S Projects	2-5
2-3	Software Development and Testing Dates for Sample S	2-8
2-4	Sample S Project Statistics	2-14
2-5	Personnel Evaluation Parameters	2-25
2-6	Sample T Subsystem Statistics	2-26
2-7	Correlations of BR, LS, DATA, NEX, and EX with Total Source Instructions (TS)	2-29
6-1	Project M, Errors/Program=f(Unnormalized Variables)	6-9
6-2	Project M, Errors/Program=f(Unnormalized Variables), Zero Errors Deleted	6-10
6-3	Project B, Errors/Program=f(Unnormalized Variables)	6-11
6-4	Project B, Errors/Program=f(Unnormalized Variables), Zero Errors Deleted	6-12
6-5	Project P, Errors/Program=f(Unnormalized Variables)	6-13
6-6	Five Predictor Summary, Errors/Program=f(Un- normalized Variables)	6-14
6-7	Ten Predictor Summary, Errors/Program=f(Un- normalized Variables)	6-15
6-8	Project M, Errors/Program=f(SI+Normalized Variables)	6-22
6-9	Project M, Errors/Program=f(SI+Normalized Variables), Zero Errors Deleted	6-23
6-10	Project B, Errors/Program=f(SI+Normalized Variables)	6-24
6-11	Project B, Errors/Program=f(SI+Normalized Variables), Zero Errors Deleted	6-25

LIST OF TABLES (Continued)

<u>Table</u>	<u>Page</u>
6-12 Project P, Errors/Program=f(SI+Normalized Variables)	6-26
6-13 Five Predictor Summary, Errors/Program=f (SI+Normalized Variables)	6-27
6-14 Ten Predictor Summary, Errors/Program=f(SI+Normalized Variables)	6-28
6-15 Project P Prediction Equation Comparison	6-32
6-16 Project M, Error Rate/Program=f(Unnormalized Variables)	6-35
6-17 Project M, Error Rate/Program=f(Unnormalized Variables), Zero Error Rates Deleted	6-36
6-18 Project B, Error Rate/Program=f(Unnormalized Variables)	6-37
6-19 Project B, Error Rate/Program=f(Unnormalized Variables), Zero Error Rates Deleted	6-38
6-20 Project P, Error Rate/Program=f(Unnormalized Variables)	6-39
6-21 Five Predictor Summary, Error Rate/Program=f(Unnormalized Variables)	6-40
6-22 Ten Predictor Summary, Error Rate/Program=f(Unnormalized Variables)	6-41
6-23 Project M, Error Rate/Program=f(SI+Normalized Variables)	6-46
6-24 Project M, Error Rate/Program=f(SI+Normalized Variables), Zero Error Rates Deleted	6-47
6-25 Project B, Error Rate/Program=f(SI+Normalized Variables)	6-48
6-26 Project B, Error Rate/Program=f(SI+Normalized Variables), Zero Error Rates Deleted	6-49

LIST OF TABLES (Continued)

<u>Table</u>		<u>Page</u>
6-27	Project P, Error Rate/Program=f(SI+Normalized Variables)	6-50
6-28	Five Predictor Summary, Error Rate/Program=f(SI+Normalized Variables)	6-51
6-29	Eight Predictor Summary, Error Rate/Program=f(SI+Normalized Variables)	6-52
6-30	Sample S, Validation Results for Five and Ten Predictor Regression Equations, Errors/Program=f(Unnormalized Variables)	6-58
6-31	Errors/Program=f(Program Structure + Programmer Variables)	6-63
6-32	Errors/Program=f(Program Structure + Programmer Variables), Zero Errors Deleted	6-65
6-33	Errors/Program=f(Program Structure Variables Only)	6-67
6-34	Errors/Program=f(Program Structure Variables Only), Zero Errors Deleted	6-69
6-35	Five Predictor Summary, Errors/Program	6-71
6-36	Best Single Predictor Summary, Errors/Program	6-73
6-37	Analysis of Variance Tables, Sample T Subsystems, Errors/Program=f(Program Structure Variables), All Observations Used	6-80
6-38	Sample T, Validation Results for Five Predictor Regression Equations, Errors/Program=f(Program Structure Variables), All Observations Used	6-82
6-39	Sample T, Validation Results for Five Predictor Regression Equations, Errors/Program=f(Program Structure Variables), Zero Errors Deleted	6-83
6-40	Error Rate/Program=f(Program Structure+Programmer Variables)	6-85
6-41	Error Rate/Program=f(Program Structure+Programmer Variables), Zero Error Rates Deleted	6-87
6-42	Error Rate/Program=f(Program Structure Variables Only)	6-89

LIST OF TABLES (Continued)

<u>Table</u>		<u>Page</u>
6-43	Error Rate/Program=f(Program Structure Variables Only), Zero Error Rates Deleted	6-91
6-44	Five Predictor Summary, Error Rate/Program	6-93
6-45	Analysis of Variance Tables, Sample T Subsystems, Error Rate/Program=f(Program Structure Variables), All Observations Used	6-97
6-46	Errors/Program=f(TS, AP, I/O, COMP, COM), Consistency Summary	6-100
6-47	Error Rate/Program=f(SYS, AP/TS, SYS/TS, EX/TS, COM/TS), Consistency Summary	6-101
6-48	Sample T Prediction Results Using all Subsystems (N=249)	6-103
7-1	Sample T, Error Rate and Programmer Variable Relationships	7-2
7-2	Correlations Between Error Rate and Total Source Instructions for Sample T Subsystems	7-6
7-3	Subsystem F Error Rate Analysis	7-7
7-4	Subsystem G Error Rate Analysis	7-8
B-1	Sample S Predictor Variable Descriptions	B-2
B-2	Sample T Predictor Variable Descriptions	B-11
C-1	Sample S, List of Predictor Variables Used and Eliminated (A Priori) When Predicting Errors/Program and Error Rate/Program	C-2
C-2	Sample T, List of Predictor Variables Used and Eliminated (A Priori) When Predicting Errors/Program	C-6
C-3	Sample T, List of Predictor Variables Used and Eliminated (A Priori) When Predicting Error Rate/Program	C-8

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
3-1	Error Rate and Source Instruction Relationship for Project M, B, and P	3-7
6-1	Depiction of Hypothetical Relationships Between Programming Errors and Length of Program	6-5
8-1	Error Rate and Source Instructions Relationship for Projects M, B, and P, Contrasted with Estimates of Error Rate Using Non-Linear Model	8-8

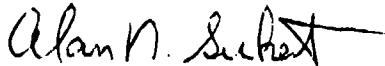
EVALUATION

The need for developing new tools and techniques for producing more reliable low cost software, as noted in such documents as the Command, Control Information Processing CCIP-85 Study (Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's), has led to attempts to analyze the nature and types of software errors in order to be able to accurately predict error occurrences, and to be able to accurately predict the reliability of software produced. Many different types of models of the software debugging process have been formulated for this purpose. However, a technique that has not been adequately applied to software errors is the use of statistical regression analysis to relate error occurrences to various structural, complexity, and programmer-related characteristics of a software package.

This effort was initiated in response to the CCIP-85 Study and this need for applying regression techniques to software errors, and fits into the goals of RADC TPO No. 5, Software Cost Reduction (formerly RADC TPO No. 11, Software Sciences Technology), in particular the area of Software Quality (Software Modeling). The report focuses on the analysis, using multiple linear regression techniques, of software error data and related structural, complexity, and programmer-related variables extracted from two large Department of Defense command and control software projects totalling over 250,000 lines of higher order language source code. The importance of performing this analysis is that it represents the first attempts to use linear regression techniques for comparing different

software projects, in order to determine those characteristics that statistically impact on the occurrence of software errors.

The conclusions drawn under this analysis will, therefore, provide new insights into those factors that influence software errors. In addition, results of this analysis will be used as the basis for future, continuing analysis for collection errors using regression techniques, and will provide a baseline for collection of software characteristics on future software projects that will support regression analysis of software errors. Finally, the results of this and future similar statistical analysis efforts will provide the necessary understanding of, and insights into, the software development process, through the understanding of those factors that lead to the occurrence of software errors, that is required in order to produce the high quality, low cost software desired.



ALAN N. SUKERT, Captain, USAF
Project Engineer

1.0 INTRODUCTION

This document is the final technical report for the Statistical Prediction Model study, RADC Contract No. F30602-76-C-0213. This eight month study focused on the statistical prediction of programming errors using a wide range of program structure/complexity variables and selected programmer variables as predictors. The authors view this study as one of many continuing studies which need to be performed that further investigate how programmers, program characteristics, management methods, and software testing and design factors influence and contribute to errors in programs.

This report is organized topically into eight interrelated sections and one technical appendix. A description of the major topics covered in each section is as follows:

- Section 1.0 presents a background discussion on the role and importance of the prediction of programming errors for improving software quality and reliability. The purpose of this study is also stated in this section.
- Section 2.0 presents a detailed discussion of background information on the two data samples that were analyzed by this study. This discussion focuses on the software development environment, software testing considerations, programming error definition and classification, predictor variables, and statistical characteristics of both samples.
- Section 3.0 briefly discusses selected limitations of the data in both samples that could affect the predictability of errors in this study. Also, three important preliminary analysis findings are discussed as they relate to the error prediction equations that were developed for this study.

- Section 4.0 presents a brief discussion of the multiple linear regression analysis method and the stepwise regression procedure that were used to obtain the error predictions for this study.
- Section 5.0 discusses various operational and analytical considerations and decisions made which pertain to the error prediction equations developed for this analysis.
- Section 6.0 presents a summary of the technical results of this study as they pertain to the predictability of errors and error rate in both data samples. A detailed presentation and discussion of the results for each sample follows this summary.
- Section 7.0 presents additional analysis of the error rate per program measure and its relationship with the programmer rating and workload variables, and also its interesting relationship with total source instructions as observed in this study.
- Section 8.0 contains the major conclusions and recommendations of this study. Recommendations are discussed which pertain to (1) improving the consistency of prediction of programming errors in future software error prediction studies, (2) further research requirements, and (3) proposed data collection support tools and techniques.
- Appendix A contains a brief technical discussion of the measure of programming errors to be analyzed in software error prediction studies and its dependence upon the method of testing used during the software test period.

1.1 Background

Software quality has many facets, including availability, ease of maintenance, CPU run-time required, and reliability. Precise definitions of quality or its components do not exist. However, software reliability, in spite of a lack of quantitative definition, has received a great deal of attention and rightly so, due to its pervading influence on the other aspects of quality. Without a reliable software system, availability, run-time, maintenance, etc., are meaningless.

In spite of the ambiguity and lack of consistency in the definitions of software reliability, one thing common to them all is that they do include programming errors. The analysis and prediction of programming errors then becomes an area that would contribute to the assessment and improvement of software quality.

Analysis of errors becomes important from the standpoint of determining their possible causes so that management controls may be exercised to reduce them. Prediction is important as a tool for the analysis, as well as contributing to the testing and certification, of a software system. If, for example, fewer errors have been found than a prediction formula indicates are present, then more testing should be performed.

One approach that is being considered for providing researchers and management with a more definitive understanding of factors which affect software quality and reliability

is that of predicting programming errors through the use of statistical regression models. It is through the use of these prediction models that:

- (1) estimates of the software reliability during each phase of the software development effort could be provided;
- (2) the amount of further testing required to achieve or insure a given level of program quality could be estimated; and
- (3) the relative effects of selected management control and both design and coding techniques on the reduction of programming errors, as aids for improving software quality and reliability, could be assessed.

Although recent studies have demonstrated the feasibility of using linear models and regression analysis methods to predict errors in programs, additional studies are needed which apply these methodologies in order to assess their value and importance for error prediction purposes.

1.2 Purpose

The purpose of this study is to apply the statistical method of multiple linear regression analysis to predict programming errors, using a variety of variables which relate to programmer capability, program structure, and program complexity as predictors of errors. Two distinct data samples will be analyzed by this study. Each sample contains data

on programming errors and program characteristics collected for purposes other than this study, and provided by RADC. Both samples reflect software that was developed by independent contractors for different, large-scale, command and control applications. A combined total of 5539 programming errors, resulting from 783 programs with a total of 296,595 source instructions, are analyzed.

2.0 DESCRIPTION AND CHARACTERISTICS OF DATA SAMPLES

Two distinct samples of data were analyzed during this study. Each data sample contains data on both programming errors and numerous program characteristics. Throughout this report the program characteristics that are presented, discussed, and analyzed are referred to as program structure/program complexity variables. Although much of this program characteristic data represents, for example, counts of the number of program instructions of a certain type that may have appeared in the program, each individual program characteristic variable can be assumed to be one of a variety of measures or estimates of the program structure and/or complexity.

The two data samples are referred to as sample S and sample T, respectively, throughout this report. The following sections of this report present background information and other relevant statistics concerning the software development environments, testing considerations, and error data characteristics of each sample which will be of importance when evaluating the prediction equation results obtained from each of the samples. Table 2-1 presents a brief summary of the differences which existed between each of these samples on various gross level project and program characteristics.

TABLE 2-1. CHARACTERISTICS OF DATA SAMPLES

	SAMPLE S	SAMPLE T
TOTAL PROGRAM MODULES	534	249
TOTAL SOURCE INSTRUCTIONS	181,249	115,346
TOTAL PROGRAMMING ERRORS	3,533	2,886
AVERAGE PROGRAM LENGTH	339	463
AVERAGE ERROR RATE ^a	1.95	1.74
APPLICATION AREA PROGRAMMING LANGUAGE USED	COMMAND & CONTROL CENTRAN	COMMAND & CONTROL JOVIAL J4
NUMBER OF FUNCTIONS PROGRAMMED	3 PROJECTS	8 SUBSYSTEMS
NUMBER OF PREDICTOR VARIABLES	58 (PROGRAM CHARACTERISTICS ONLY)	16 (PROGRAM CHARACTERISTICS & PROGRAMMER VBLS.)
ERROR DATA COLLECTION	DURING TEST & INTEGRATION PHASE ONLY	DURING VALIDATION, ACCEPTANCE, INTEGRATION, AND OPERATIONAL PHASES

^a AVERAGE ERROR RATE = A MEASURE OF TOTAL ERRORS FOUND PER 100 LINES
OF SOURCE CODE FOR THE ENTIRE SAMPLE.

$$\left(\frac{\text{I.E., } 100 \times \text{TOTAL PROGRAMMING ERRORS}}{\text{TOTAL SOURCE INSTRUCTIONS}} \right)$$

2.1 Sample S

2.1.1 Software Development Environment

Sample S software, which consisted of 534 programs, was developed as three command and control systems which for the purposes of this report are to be referred to as projects M, B, and P, respectively. This software was developed jointly by two private industry organizations from mid-1969 through late 1973, and represents an effort of approximately 5500 man-months. These programs consisted of 181,249 source instructions written in CENTRAN, an English-like, special purpose, higher level language that was designed for use only on Central Logic and Control (CLC) computers. The programs analyzed represent about 80 percent of the approximately 644 programs with 240,000 source instructions that were written for the entire command and control application. The software development effort attempted to conform to the traditional approach of building large software systems:

- (1) definition of system performance requirements,
- (2) design of functional specifications from which programming specifications are written,
- (3) coding and unit testing of those software elements comprising a process subfunction,
- (4) integration testing of the elements within a subfunction, and
- (5) integration testing of the system processes.

Significant, but not always successful, attempts were made to write system design specifications and develop software before the system performance requirements and functional specifications were completed. In projects M and P for example, serious program design work and coding were initiated during the period of June through August 1969, although the performance requirements were not completed until June 1970. Using the performance requirements as a base, a design team from one of the organizations defined the system functional design requirements and specifications from which software design teams from both organizations generated the formal programming design specifications. Where these latter specifications did not coincide with the software already written, the original code had to be rewritten.

A description of the extent to which the new programming technologies (e.g., structured programming, top-down design, etc.) were implemented and applied during the development of each of the three sample S software projects is presented in Table 2-2 along with several basic program length and error statistics for each project. It appears from Table 2-2 that project P was the only project that actually implemented or applied these new technologies to any large extent.

It is important to note here one final consideration regarding these new technologies as applied in sample S. No information is available, other than that which appears in Table 2-2, to give any indication of how consistently and with what thoroughness and quality the concepts and principles which underlie these new technologies were strictly followed in the programs developed for sample S. Necessarily then, and in line with the stated purposes of this research study, no comparisons

TABLE 2-2. APPLICATION OF NEW PROGRAMMING TECHNOLOGIES
IN SAMPLE S PROJECTS

	M	B	P
NO. OF PROGRAMS	395	104	35
TOP DOWN DESIGN	1.0%(3)	100.0%(104)	48.6%(17)
STRUCTURED CODE	2.8%(11)	0.0%(0)	65.7%(23)
CHIEF PROGRAMMER	0.0%(0)	0.0%(0)	51.4%(18)
TEAM			
PROGRAMMING	0.0%(0)	0.0%(0)	97.1%(34)
LIBRARIAN			
AVERAGE SOURCE INSTRUCTIONS	345.2	212.3	651.9
AVERAGE ERRORS/PROGRAM	6.8	6.0	6.8
AVERAGE ERROR RATE ^a	2.0	2.8	1.0

^a AVERAGE ERROR RATE = A MEASURE OF TOTAL ERRORS FOUND PER 100 LINES OF SOURCE CODE FOR EACH PROJECT.

are made or inferences drawn concerning the relative effectiveness of these technologies for improving software quality and reliability.

2.1.2 Software Testing Considerations

Each of the three software development projects of sample S adhered to a basically standard code writing and testing scheme. One programmer was usually assigned the responsibility of writing several programs which would subsequently interface as a single functional unit, with several of these units forming a subfunction. When a program was compiled error-free and unit tested, it was combined with its related counterparts to form a functional unit for element testing. This testing was performed by the programming team and was usually the prerequisite for submitting a system's subfunctions to the Test and Integration (T&I) team. Immediately prior to the T&I phase, all the subfunctions of a system would be bound together to form that system's thread. It was this thread which was delivered to T&I for integration testing. The T&I phase assured that the subfunctions of a system interfaced properly and represented the beginning of formal error recording for the software being tested. Trouble reports describing the error and its severity were forwarded to the respective programming team for resolution. To expedite error resolution, patches to the object code were made prior to resubmission to the T&I group. The applicable source code was subsequently updated, usually when a new version of a process was released.

The system integration tests were designed to verify that the system could respond in certain areas of basic system capabilities within predicted tolerances. The testing

required to support the final demonstration or acceptance tests in each test case was assumed to begin with single-site tests, in which communications with the rest of the system were usually simulated. Later tests were built up to include all sites netted together. Table 2-3 presents the software development and testing dates that were reported for each of the three projects of sample S.

No other specific data was available for this study with respect to the amount and thoroughness of testing that sample S programs underwent during the T&I phase of software development.

2.1.3 Definition, Classification, and Collection of Error Data

For sample S, programming errors were defined as those errors found during the T&I phase which could be attributed to the programmer and required a change to the program's source code. At the time this error data was provided, no additional classification of errors had been attempted. (Presently, however, there is an effort underway, supported by RAIC, that will result in a thorough classification of the various error types for the sample S programs being analyzed for this report.)

With respect to error data collection for sample S programs, early in this analysis effort informal discussions with personnel responsible for the sample S data collection revealed that the data collection on program characteristics may have been obtained as much as three years after the error data was collected. This delay is estimated based on the dates provided in Table 2-3. After the errors were detected, programs were modified as a direct result of these errors. Programs continued

TABLE 2-3. SOFTWARE DEVELOPMENT AND TESTING DATES FOR SAMPLE S

	PROJECT M	PROJECT 8	PROJECT P
DATE INITIAL DESIGN EFFORT BEGAN	JUNE 69	NOVEMBER 71 ^a	JANUARY 73
DATE ^a FIRST VERSION OF SOFTWARE WAS DELIVERED TO T&I TEAM	OCTOBER 71	DECEMBER 72	AUGUST 73 ^b
DATE FINAL VERSION OF SOFTWARE WAS DELIVERED TO T&I TEAM	JANUARY 73	JULY 74 ^b	(NOT REPORTED)
DATE THROUGH WHICH ERROR RECORDING CONTINUED	(NOT REPORTED)	NOVEMBER 74	(NOT REPORTED)
ERROR HISTORY TIME PERIOD	2 1/4 YRS. (MINIMUM)	2 YRS., APPROX.	(NOT REPORTED)

^a THIS DATE MARKED THE BEGINNING OF FORMAL TROUBLE REPORT ACCOUNTABILITY WITHIN EACH OF THE PROJECTS' SOFTWARE SYSTEM. AS EACH ADDITIONAL SUBFUNCTION OR VERSION WAS DELIVERED TO T&I, SIMILAR FORMAL ERROR RECORDING WAS INITIATED.

^b INDICATES THE VERSION FROM WHICH PROGRAM CHARACTERISTICS DATA (I.E., THE PREDICTOR VARIABLE VALUES FOR SAMPLE S) WERE COLLECTED, AS PROVIDED FOR THIS STUDY. THE VERSION FOR PROJECT M WAS NOT REPORTED.

to be modified after the T&I phase as part of the normal growth process of the respective systems being developed.

Since each of the predictor variables in sample S represent program characteristic variables, and given that there was no measure of the extent to which these program characteristics had been modified between the time when the error and predictor variable data were collected, a serious question arose as to the validity of the results obtained from a multiple regression analysis of this data. This condition might result in the anomalous position of attempting to predict errors from data which to some extent at least may have resulted from the errors.

Additional discussion with personnel responsible for the sample S data collection indicated that the extent of the modifications made to the program characteristics as a result of the errors being corrected was minor. However, it was apparent from having initially reviewed the sample S data that the variability of some of the predictor variables was also minor. Thus, the effect of the minor modifications on the conclusions drawn from predictor variables with minor variability became an unknown which could not be assessed.

In order to deal with this problem in such a way that the data could still be utilized for error prediction purposes, a decision was made to categorize each of the independent variables for this sample. This categorization procedure involved examining the range of each independent variable, grouping the scores into equal intervals, and the assigning of new scores to the intervals. For example, for one variable all scores in the interval 0-99 would be rescored as 1; scores

in the interval from 100-199 would be rescored as 2, etc.. Clearly, this grouping technique when implemented would undoubtedly throw away some useful data; it would also eliminate part of the error due to assuming that code changes after error counts were insignificant. If modifications to the program characteristics were indeed minor as had been indicated, then categorizing the data would give a truer picture of what these characteristics would have been at error collection time for all values of the predictor variables except those at the boundaries of the class intervals. Thus, conclusions drawn would have relatively more validity.

The categorization procedure was applied to all sample S predictor variables. Intercorrelation matrices were then obtained for each of the projects which showed the intercorrelations among the predictors and between the predictors and the criterion variable, programming errors. Each of these sets of intercorrelations was then compared with the respective intercorrelations obtained when the original (uncategorized) predictor variable values were used. Essentially, no significant statistical differences were found among the two sets of correlation matrices for each of the projects. Thus, after having performed this rather extensive computational task of categorizing the data and then comparing various sets of correlation matrices, a firm decision was made to continue on with the multiple linear regression analysis using the original data as obtained for sample S.

Based on the above discussion of problems associated with the sample S error data collection, it should be noted that any data that is to undergo a secondary analysis is subject to the same or similar problems that were discovered when attempting to analyze this data. Accessibility of the sample S

personnel led directly to the discovery of these problems. Other data in the future could be collected by personnel not as accessible, thus resulting in problems at least as serious, though undiscoverable.

The problems discovered thus far lead to a strong argument for (1) identifying or determining what use is to be made of the data before it is collected, and (2) developing and providing the necessary data collection instruments, procedures, formats, and software support systems that can collect and store repeated snapshots of program characteristics and error data throughout various phases of the software development cycle.

2.1.4 Predictor Variables

A list and description of the 54 predictor variables that were made available for the analysis of sample S programs is presented in Table B-1 of Appendix B. Variables 56 through 109 were constructed during this study in order to investigate their effectiveness in predicting error rate/program for this sample. Further discussion of these variables (i.e., 56-109) is presented in section 3.0.

Each of these initial 54 predictors was collected as previously mentioned during December 1974 and January 1975 via an automatic scanner program developed by sample S data collection personnel. This scanner program could interrogate source code programs that were written in CENTRAN, ALC, or PL/1.

It is important to note that other variables in addition to the 54 that were collected could have been measured and collected via this scanner program. For example, Lock Macros (variable X20) is the only one of several different types of CENTRAN system macros that was collected separately. All other system macros used in the program, including Lock Macros, are summed into variable X10, System Macros. Basically then, what variables were collected by sample S personnel clearly represent the variables hypothesized by that group to be of some particular interest and importance for their own purposes of analysis. The point made here is that other variables which may have contributed significantly to the prediction of programming errors were not collected in the sample S data and resultantly were not available for review, analysis, and evaluation by this study. There is a definite need for the definition of a uniform set of program characteristics that may be applied to a wide variety of projects. By doing so, it would be possible to compare the results of one project with another and thus draw conclusions applicable to programming in general, not merely to programming as reflected by one specific project.

Lastly, with respect to the first 54 predictors it is important for analysis purposes to identify the various linear combinations that existed among these variables. Variable X1 was identified as a linear combination of variables X7, X8, X10, X11, X15, X16, X41, and a variable that was not separately counted, the number of assignment instructions not involving arithmetic operations (e.g., A=1; B=5 etc.). System Macros (X10) is a linear combination of variables X3, X9, X20, and any other CENTRAN System Macros. Variable X17, Scaling/Rounding Operations,

is considered a part of Centran Functions (X16). Variable X39, Total Variables, is a linear combination of variables X22, X25, X27, X29, X31, X33, X35, and X37. Variable X40, Total Variable Frequency, is a combination of variables X23, X24, X26, X28, X30, X32, X34, X36, and X38. Finally, variable X41, Total Do Loops, is a combination of variables X18, X19, X42, X43, X44, X45, X46, and X47.

2.1.5 Characteristics of Sample Data

During the initial phases of the analysis, univariate and bivariate frequency distributions were obtained for each of the 54 predictors and for each of the predictor variables with errors/program. For the dependent variable and most of the predictor variables the univariate frequency distributions were asymmetric with the higher frequencies concentrated toward the lower end of the variable and the smaller frequencies asymptotically spread out toward the higher end. The bivariate distributions basically showed the existence of a predominant number of very low to moderate linear relationships existing between the various predictor variables and programming errors. There was a clear indication of a tendency toward non-linearity in many of the relationships between the predictor variables and errors. Based on these observations it is possible that better predictions could be obtained by non-linear transformations, of either or both programming errors and the predictor variables.

In addition, based on the sample S project statistics as presented in Table 2-4 and on differences that were observed between means and measures of variability of predictors between projects, there was a clear indication that the three

TABLE 2.4 SAMPLE S PROJECT STATISTICS

STATISTICS	M	B	P	ALL PROJECTS
<u>NO. OF PROGRAMS (WITH 6 WITHOUT ZERO ERRORS)</u>	395	104	35	534
TOTAL SOURCE INSTRS.	136,358	22,075	22,816	181,249
TOTAL ERRORS	2,673	622	238	3533
AVG. PROGRAM LENGTH	345.21	212.26	651.89	339.42
AVG. ERRORS/PROGRAM	6.77	5.98	6.80	6.62
AVG. ERROR RATE	1.96	2.82	1.04	1.95
AVG. ERROR RATE/PGM.	2.42	3.26	1.29	2.51
PERCENT OF CODE WITH ZERO ERRORS	9.82%	5.40%	0.00%	8.04%
PERCENT OF PROGRAMS WITH ZERO ERRORS	22.78%	8.65%	0.00%	18.54%
<u>NO. OF PROGRAMS (WITH 1 OR MORE ERRORS)</u>	305	95	35	435
TOTAL SOURCE INSTRS.	122,971	20,882	22,816	166,669
TOTAL ERRORS	2,673	622	238	3533
AVG. PROGRAM LENGTH	403.17	219.81	651.89	424.96
AVG. ERRORS/PGM.	8.76	6.55	6.80	8.12
AVG. ERROR RATE	2.17	2.99	1.04	2.06
AVG. ERROR RATE/PGM.	3.13	3.57	1.29	3.08
<u>NO. OF PROGRAMS (WITH ZERO ERRORS)</u>	90	9	0	99
TOTAL SOURCE INSTRS.	13,387	1193	---	14,580
AVG. PROGRAM LENGTH	148.74	132.56	---	140.65
SHORTEST/LONGEST PGMS.	6/1047	18/389	---	6/1047

sample S projects were quite heterogeneous over a majority of the data variables. As indicated in Table 2-4 many of the programs, both short and long, were reported as error free as a result of testing during the T&I phase. Particularly for the longer programs, this fact is hard to reconcile with past experience and logical reasoning on this subject which would contend that as programs become longer, the number of paths through the program increases, which in turn increases the program's complexity, thus increasing the likelihood of a larger number of errors residing in these longer programs. Section 3.0 contains additional discussion about programs with zero reported errors.

2.2 Sample T

2.2.1 Software Development Environment

Sample T software consists of 249 programs which represent the initial delivery of a large command and control software system. This system was functionally developed as eight subsystems which for the purposes of this report are referred to as subsystems A through H, respectively. The software was jointly developed by two private industry organizations, with one of the organizations being responsible for 77 of the 249 programs that were written. The total programs consisted of 115,346 source instructions written in JOVIAL J4, a higher level language which is compatible with the SYMON operating system. Batch type processing was the exclusive operating mode used during this software development.

Software development for sample T subsystems followed a "single increment" development approach and was governed by formally specified and approved requirements that had been defined down to the function level. "Single increment", as defined for sample T data, refers to a typical development cycle in which each development phase is performed only once. This is in contrast to the top-down, multiple increment approach, where the cycle is repeated several times, first for a system of stubs and subsequently when the stubs are replaced with deliverable (i.e., ready to be tested) software.

Structurally, the smallest compilable unit of source code was the routine or program. Programs were joined to form functions, functions were joined to form subsystems, and finally subsystems were joined to comprise the one command

and control system. This structure was produced by a project organization based on the function being developed. This meant that management of sample T software development was set up in conjunction with the software structure so that personnel assigned to one work unit, ranging in size from 5 to 15 programmers, produced all the software in one or more of the functions. No new or advanced programming technologies were reported as being applied to sample T programs or implemented during their development.

2.2.2 Software Testing Considerations

Testing of sample T software proceeded through five sequential phases: development, validation, acceptance, integration, and operational demonstration. Development testing was informal with all test cases being written and executed by the program development personnel. No information was provided regarding the length of time involved in this particular phase of testing. Validation testing marked the first phase of formal software testing by an independent test group. The objective of tests that were developed for both the validation and acceptance phases was to demonstrate that the sample T programs did in fact satisfy or meet the formal functional requirements that had been specified. Each of the tests applied during these two phases was run at the subsystem level but was designed to examine software performance at the program, function, subsystem, and system levels. An important consideration of these two phases was that all testing was performed on a master configuration and no alteration of the code was allowed. Acceptance testing consisted of rerunning selected tests that had been used for validation testing, particularly

those that specifically demonstrated the software requirements. Customer acceptance of the software was dependent upon the successful completion of these tests.

Integration testing was conducted by an independent contractor whose major responsibility was to demonstrate that the applications software interfaced correctly with the operating system and the system support software. Tests conducted during this time were similar in structure and formality to those tests that were used during the validation and acceptance phases. The operational demonstration phase was a short period of testing which followed an operational timeline and which used an operational data base. The objective of testing during this period was to demonstrate the satisfactory performance of the software in the operational environment.

The approximate length of calendar time (as measured in weeks) involved in each of the four phases of formal testing for sample T programs was as follows:

Validation	10.3
Acceptance	1.5
Integration	9.5
Operational Demonstration	2.3

With respect to error detection during each of these test phases, it was reported that each phase found errors which should have been detected in earlier phases. No measures of the amount or thoroughness of testing were available for

sample T software. Resultantly, it is not possible to state what percent of, and to what extent, the code had been exercised (given the possible range of input values for each test) during this formal testing period.

2.2.3 Definition, Classification, and Collection of Error Data

For sample T, programming errors were defined as those errors found during the four stages of formal testing which could be attributed to the programmer and required a change to the program's source code. Programming errors were classified as belonging to one of a variety of error categories which either (1) described the symptoms of the problem (e.g., output report has data entries that were not printed or are missing) or (2) actually identified the cause of the error (e.g., logical condition not tested which resulted in abnormal program end).

The 2006 programming errors that are analyzed in this report were aggregated and classified according to nine major error categories. These categories along with the approximate percentage of errors that occurred in each are as follows:

Logic	26.0
Data Handling	18.2
Interface	17.0
Data Input/Output	16.4
Computational	9.0
Other	8.5
Data Base	4.1
Data Definition	0.8

Unfortunately, for the sample T error data provided for this study, there was no way of knowing which errors belonged to each of the error categories as indicated. Only recently, after a majority of the analysis had been completed, was it possible to obtain a first available version of the automated data which describes in what programs the particular error type occurred, the phase of testing in which the error was detected, etc. Necessarily then, no analysis of programming errors by error type or category was attempted during this study.

Two important points regarding the error categories as listed deserve brief mention at this time. First, the individuals who assigned or classified programming errors according to these software error categories were not the same individuals who had initially recorded the error in a Software Problem Report (SPR). The SPR and a closure report which contained an explanation of the problem and the corrections required to resolve the problem were usually the main sources of information used to retrospectively classify the errors.

Secondly, it was reported by the individuals who classified the errors that not all errors were necessarily programming errors, and that the probable source of these errors could be traced to four development activities:

- (1) requirements specifications,
- (2) design,
- (3) coding, and
- (4) maintenance (correction of other errors).

When the sample T programming errors were classified according to these probable sources: (1) errors were only able to be classified either as design or coding errors based on the available information provided by the SPR and closure report, and (2) approximately 64% of the errors could be attributed to design and 36% to coding as probable sources for these errors. An explanation that was offered for this outcome was that, in their collection of supporting data to explain software error histories, poorly stated requirements or changing interpretations of requirements were offered as reasons for difficulty in developing various programs that were found to be error-prone.

As previously mentioned, error data for sample T was collected throughout each of the four stages of formal systems testing. However, no information is available as to the dates for the data collection of the program structure data. If the program structure data was collected prior to the error data collection, then the programming errors could be considered as immediately resulting from the various measured and unmeasured program characteristics. On the other hand, if the program structure data collection was performed after modifications were made to these program structure variables as a direct result of correcting for errors found during the formal test phases, then there exists a similar situation as was discovered for the sample S programs. This problem, for both samples, is one of several which raises questions as to the validity of the data that was used for this study. It is important that researchers are aware of sources of possible invalidity in data collection or program selection. The effects can then be considered in terms of the results obtained and the conclusions drawn from the study.

2.2.4 Predictor Variables

A list and description of the 16 predictor variables that were provided for the analysis of sample T programs is presented in Table B-2 of Appendix B. Variables 18 through 31 were constructed during this study to investigate their effectiveness in predicting error rate/program for this sample. These variables will be further discussed in Section 3.0.

Each of the 16 original predictor variables were collected by means of a scanner program which could interrogate source code programs that were written in JOVIAL J4. The only known linear combinations that existed among these 16 predictors were as follows: (1) Total Source (TS) was a linear combination of variables Non-Executable Instructions (NEX) and Executable Instructions (EX); and (2) Total Interfaces (TI) was a linear combination of the two interface variables, Application Interfaces (AP) and System Interfaces (SYS).

Regarding a more detailed description or definition of the software-related predictor variables, only a limited amount of more specific information was available. This information is as follows: four generic types of executable code were arbitrarily defined:

I/O - I/O refers to JOVIAL defined and SYSTEM defined input and output statements. JOVIAL I/O statements include FORMIN, FORMOUT, DECODE and ENCODE. SYSTEM DISC

I/O includes 'SDAHA and its various entrances. Examples of SYSTEM TAPE I/O are 'CWRITE, 'WEOF and 'REWIND.

COMPUTATIONAL- These are statements expressing equations containing arithmetic operators.

Example: AA = BB*CC**2/DD \$

DATA HANDLING- These statements effect a simple data transfer (equality) from one variable to another and are distinguished from computational statements.

Examples: XX=YY \$, AA(\$BB+2, DD\$) = 'PR \$.

LOGICAL- Logical statements establish branches in the code and include the IF, IFEITH, ORIF, FOR and GOTO SWITCH statements.

Also, more descriptive information relating to how the Total Branches (BR) and Interface (AP,SYS) variables were defined was provided. Total Branches (BR) was described as including all possible logical branches, resulting from IF, IFEITH, ORIF, and GOTO-SWITCH name statements. The BR variable does not reflect the actual number of logical branches the program will make when it executes. Program-to-program and program-to-data base interface descriptions were described as being available from system utility or construct programs. To this could be added details of the individual interface (e.g., number of arguments in the calling sequence), the type of interface (applications, system, user, data base), and the format of the information passed.

The two predictor variables, Programmer Rating (RAT) and Workload (WKLD), were metrics that were constructed to evaluate programmer performance with respect to (1) selected programmer-specific criteria and (2) programmer assignment or job-specific criteria. The evaluation was made on only those that had been exclusively developed by one of the two private industry organizations who shared responsibility for the overall software development effort. The evaluation was performed by the programmer's line management after the project was completed. Table 2-5 presents the personnel evaluation parameters that were used to develop the programmer rating and workload measures relative to software quality. From these parameters, the Programmer Rating variable (RAT) was constructed by simply summing the scores obtained on each of the knowledge, intelligence, initiative, and responsibility categories. One final point to consider is that many of the 172 programs for which RAT and WKLD measures were available were developed jointly by two or more (up to 15) programmers. For these programs, the RAT and WKLD measures represent the averages obtained for these variables over each of the individual programmer's scores.

2.2.5 Characteristics of Sample Data

Subsystem statistics for sample T are presented in Table 2-6. Clearly these subsystems are quite heterogeneous when one considers the differences which exist across subsystems with respect to the number of programs, average errors/programs, and average error rate/program. One major commonality, however, that was found over all subsystems was the high intercorrelations which existed between Total Source Instructions (TS)

TABLE 2-5. PERSONNEL EVALUATION PARAMETERS

	PARAMETER	RANGE ^b	BASED ON
PROGRAMMER SPECIFIC PARAMETERS	TECHNICAL CAPABILITY ^a		
	KNOWLEDGE	1-5	FAMILIARITY WITH LANGUAGE AND MACHINES FAMILIARITY WITH ENGINEERING CONCEPTS FAMILIARITY WITH PROCESSING CONCEPTS
	INTELLIGENCE	1-5	PROBLEM-SOLVING ABILITIES CREATIVITY MENTAL ACUITY
	INITIATIVE	1-5	RECOGNITION OF TASKS REQUIRED ATTACK OF PROBLEMS AND INTERFACES EFFECTIVE UTILIZATION OF TIME
JOB SPECIFIC PARAMETERS	WORK HABITS	1-5	CONCENTRATION ON JOB COMMITMENT TO DOING JOB WELL WORKING EXTRA HOURS IF REQUIRED
	WORK LOAD	0.5-1.5	ESTIMATED RELATIONSHIP OF ACTUAL WORK LOAD TO NORMAL FULL WORK LOAD (0.5 TO 1.5)

^aTECHNICAL CAPABILITY NOT RELATED TO YEARS OF EXPERIENCE.

^bRATING SCALE FOR PROGRAMMER SPECIFIC PARAMETERS:

- 1 - INADEQUATE
- 2 - ADEQUATE
- 3 - AVERAGE
- 4 - EXCELLENT
- 5 - SUPERIOR

TABLE 2-6. SAMPLE T SUBSYSTEM STATISTICS

STATISTICS	A	B	C	D	E	F	G	H	ALL SUBSYSTEMS
NO. OF PROGRAMS (WITH AND WITHOUT ERRORS)	51	16	39	15	14	37	45	32	249
TOTAL SOURCE INSTRS.	18,786	7,758	26,698	18,259	6,978	9,485	14,784	28,686	115,348
TOTAL ERRORS	267	288	488	188	144	88	268	467	2,886
AVG. PROGRAM LENGTH	366.78	484.87	684.56	683.93	497.86	256.35	328.53	646.44	463.24
AVG. ERRORS/PROGRAM	5.24	12.58	12.51	6.67	18.26	2.16	5.78	14.59	8.86
AVG. ERROR RATE	1.43	2.58	1.83	8.97	2.87	8.84	1.76	2.26	1.74
AVG. ERROR RATE/PGM.	2.23	3.58	2.56	2.42	4.96	8.96	1.51	2.62	2.27
PERCENT OF CODE WITH ZERO ERRORS	1.94%	8.88%	8.75%	3.59%	8.22%	25.47%	8.26%	1.86%	4.16%
PERCENT OF PROGRAMS WITH ZERO ERRORS	13.73%	8.88%	7.69%	16.66%	7.14%	48.54%	24.44%	3.13%	15.66%
NO. OF PROGRAMS (WITH ERRORS)	44	16	36	14	13	22	34	31	218
TOTAL SOURCE INSTRS.	18,343	7,758	26,498	9,891	6,955	7,869	13,563	28,466	118,543
TOTAL ERRORS	267	288	488	188	144	88	268	467	2,886
AVG. PROGRAM LENGTH	416.88	484.87	736.85	786.58	535.88	321.32	398.91	668.19	526.48
AVG. ERRORS/PROGRAM	6.86	12.58	13.56	7.14	11.88	3.64	7.65	15.86	9.55
AVG. ERROR RATE	1.46	2.58	1.84	1.81	2.87	1.13	1.92	2.28	1.81
AVG. ERROR RATE/PGM.	2.59	3.58	2.77	2.59	5.34	1.61	1.99	2.78	2.68

TABLE 2-6. SAMPLE T SUBSYSTEM STATISTICS (CONTINUED)

STATISTICS	A	B	C	D	E	F	G	H	ALL SUBSYSTEMS
NO. OF PROGRAMS (WITHOUT ERRORS)	7	8	3	1	1	15	11	1	39
TOTAL SOURCE INSTRS.	363	---	288	368	15	2,416	1,221	228	4,883
AVG. PROGRAM LENGTH	51.86	---	66.67	368.88	15.88	161.87	111.88	228.88	123.15
SHORTEST/LONGEST PGMS.	27/79	---	33/182	368	15	19/558	38/276	228	15/558

and the variable BR, LS, DATA, NEX, and EX. These correlations are reported in Table 2-7. Since each of the variables within this group was highly related to others by definition (i.e., $TS+NEX + EX$, and $EX=BR + LS + DATA + \text{other executable statements appearing in the program}$), the high intercorrelations were not surprising. The fact that these intercorrelations were consistently high and of similar magnitude across all subsystems is an interesting finding. It is not known whether this phenomenon can be explained by:

- (1) the characteristics of the JOVIAL J4 programming language, since these high correlations are being observed not only over eight heterogeneous subsystems but also over numerous dissimilar functions that were being programmed, or
- (2) by the use of the same basic set of programmers to program similar functions over all subsystems, or
- (3) by the fact that this a universal finding, i.e., one which applies to other programming languages as well.

In general, for all subsystems, each of the univariate frequency distributions was highly peaked and demonstrated minimal to extreme positive skewness. The intercorrelations between each of the predictors and errors ranged from low to very high across all subsystems. Generally, these correlations

TABLE 2-7. CORRELATIONS OF BR, LS, DATA, NEX, AND EX WITH TOTAL SOURCE INSTRUCTIONS (TS)

VARIABLE	SAMPLE T SUBSYSTEM							
	A	B	C	D	E	F	G	H ^a
BR	.988	.981	.983	.994	.982	.949	.983	.552 ^a
LS	.993	.973	.985	.916	.974	.965	.989	.981
DATA	.996	.979	.989	.988	.992	.965	.968	.951
NEX	.996	.994	.996	.988	.994	.975	.995	.989
EX	.997	.999	.999	.999	.999	.993	.999	.999

^a WHEN THREE PROGRAMS WERE OMITTED FROM THE COMPUTATION, THE CORRELATION INCREASED TO .816. THE THREE PROGRAM-OBSERVATIONS WERE ONES WHICH HAD MORE BRANCHES RECORDED FOR THEM THAN TOTAL SOURCE INSTRUCTIONS IN THE PROGRAM. THE RESPECTIVE VALUES OF BR AND TS FOR THE THREE PROGRAMS DELETED FOR THIS COMPUTATION: GILY HERE (379,218); (641,59); AND (273,256).

were higher than those observed for the predictor variables with errors for the sample S programs. Also, as in sample S, a significant number of programs were reported as error-free (see Section 3.0).

3.0 PRELIMINARY REVIEW AND ANALYSIS OF DATA SAMPLES

3.1 Selected Limitations of the Data

The purpose of this section is to briefly enumerate and discuss selected limitations of the data in both samples that could affect or influence the predictability of errors, and resultantly could affect or limit the generalizability of conclusions reached in this study.

Data Collection and Definition of Variables - It should be clear from the preceding discussion of both data samples in Section 2.0 that serious questions can be raised with respect to (1) when the data for the predictor variables were collected vis-a-vis the error data and (2) the limited usefulness of the predictors' definitions and descriptions for aiding an understanding of how each variable may uniquely influence or contribute to programming errors. To be sure, the need exists for future research projects to carefully identify and define the variables to be analyzed, discuss why they were selected, and identify what use is to be made of the data, prior to the actual data collection. With the definitions presently available, there is little possibility for any comparisons to be made between the predictions of both samples, not to mention the limited possibility for comparison of these predictors with variables obtained from other projects in which programming languages other than CENTRAN and JOVIAL-J4 have been used.

Classification and Definition of Errors - For both data samples a complete classification and detailed definition of programming error categories were either non-existent or unavailable to be analyzed for the purposes of this study.

Clearly without these error classifications and definitions for each category, analysis is limited to an aggregate or gross count of errors. Better predictions might very well result from using total errors of a specific type as the dependent variable.

Heterogeneity Within and Between Data Samples - From the discussion presented in Section 2.0, it is apparent that the software development environments, software testing conditions, programming languages, project management methods, and the command and control functions being programmed were different in many respects between the two data samples being analyzed. Furthermore, as evidenced by the statistics presented in Tables 2-4 and 2-6, there are differences in variability among the three projects of sample S and among the eight subsystems of sample T. These differences are further indicative of the functional differences that existed between each of the projects and subsystems in the two samples, and the individual differences that existed among the programmers responsible for the software development effort. Unquestionably, these differences or lack of homogeneity between and within the data samples will restrict the extent to which the prediction equation results for a given set of program-observations can be compared to other sample observations.

Thoroughness of Program Testing - For both data samples, little is known about the thoroughness of testing of all the 783 programs being analyzed in this study. For that matter, the prediction equations developed in this study are limited in that they apply to observed errors only. No information is available as to the latent errors, or those which might be

found at a later time as a result of more intensive testing or operational usage of the programs. Most assuredly, given the increasing manpower and costs in large-scale command and control software maintenance, and the increasing attention being paid to the relationship between thoroughness of testing and software quality and reliability at the DOD software management level and in the research literature, valid and reliable measures of program testedness need to be developed and applied to all ongoing and future software development efforts.

3.2 Preliminary Analysis Findings and Observations

Correlations Between Source Instructions and Other

Predictors - For both data samples many of the predictor variables had moderate to high positive correlations with the Source Instructions variable (i.e., X1 for sample S, and TS for sample T). For example, for project M of sample S which involved 395 of the total 534 programs being analyzed for this sample, 30 of the 53 predictors had correlations which ranged from .30 to .98. Similar correlations were also observed for projects B and P of this sample. For subsystem A of sample T which had the largest number of program-observations (N=51) of any subsystem, 11 of the 15 predictors had correlations which ranged from .37 to .99. There too, similar correlations were observed over each of the remaining subsystems, B thru H. In fact the consistency of some of these correlations for the BR, LS, DATA, NEX, and EX variables over the eight subsystems was reported earlier in Table 2-7. In general for both data samples, many predictors were also correlated to a similar degree with other predictors besides Source Instructions. However none of these predictors was correlated over the large number of variables with the same magnitude as was Source Instructions.

Generally, when many highly intercorrelated variables are being used for prediction purposes, serious mathematical problems result (e.g., the matrix of intercorrelations among predictors may become singular), which yield an indeterminate solution to the prediction equation. For this reason, many of the variables in both samples having very high correlations with Total Source Instructions were eliminated from the analysis.

For those variables which remained; i.e., those that did not correlate very highly with Source Instructions, it was desirable to obtain an additional measure of their contribution (or correlation) to programming errors with the effect of Total Source Instructions removed. This consideration leads directly to the need to "normalize" the predictor variables.

Basically, the effect of source instructions was to be removed from each of the predictor variables by means of the normalization procedure. Although several more involved computational procedures are available as alternatives for doing this, it was decided to divide each predictor's value in a given program by the number of Source Instructions for that program.

As an end result of this normalizing procedure being applied, (1) a net doubling occurred to the number of predictor variables that could be considered in any one prediction equation for each sample, and (2) a new dependent variable (errors/source instructions) was added to each program-observation which is referred to as the error rate per program.

Error Rate and Length of Program - Once the normalization procedure had been carried out and the correlations among all variables were once again obtained, it was observed with interest that the correlation of error rate (i.e., the normalized errors per program variable) with Source Instructions was negative, and low to moderate in magnitude, over most of the samples to be analyzed. Interpreting the correlations directly meant that as the number of source instructions in a program increases, the errors per 100 lines of code decreases, and vice-versa.

In essence, for the programs in the two samples this suggested that the shorter programs have higher error rates than the longer programs. Nevertheless, when the relationship between error rate and source instructions was actually graphed for projects M, B, and P of sample S, as presented in Figure 3-1, the reason for the low negative correlations became more apparent.

These graphs show that, for each of the three projects, as number of source instructions increases the error-rate increases, reaching a maximum error-rate in the range of 200-400 source instructions. From that point on, the error-rate decreases as number of source instructions increases. This phenomenon requires some explanation.

It is well-known that as the number of source instructions increases, the number of possible paths through the program usually increases and that this increase is at a more rapid rate than a linear one (perhaps not exponential, but more than linear). To detect the same percentage of total errors in two programs, the testing effort should exercise approximately the same percentage of total paths. Therefore, the amount of testing to detect equal percentages of total errors should increase at a rate faster than linearly. It is hypothesized that such was not the case for these projects, and that the negative correlation between length of program and error rate is due to lack of thoroughness of testing. That is, the shorter programs were more thoroughly tested than the longer ones, in terms of having a higher percentage of their paths executed.

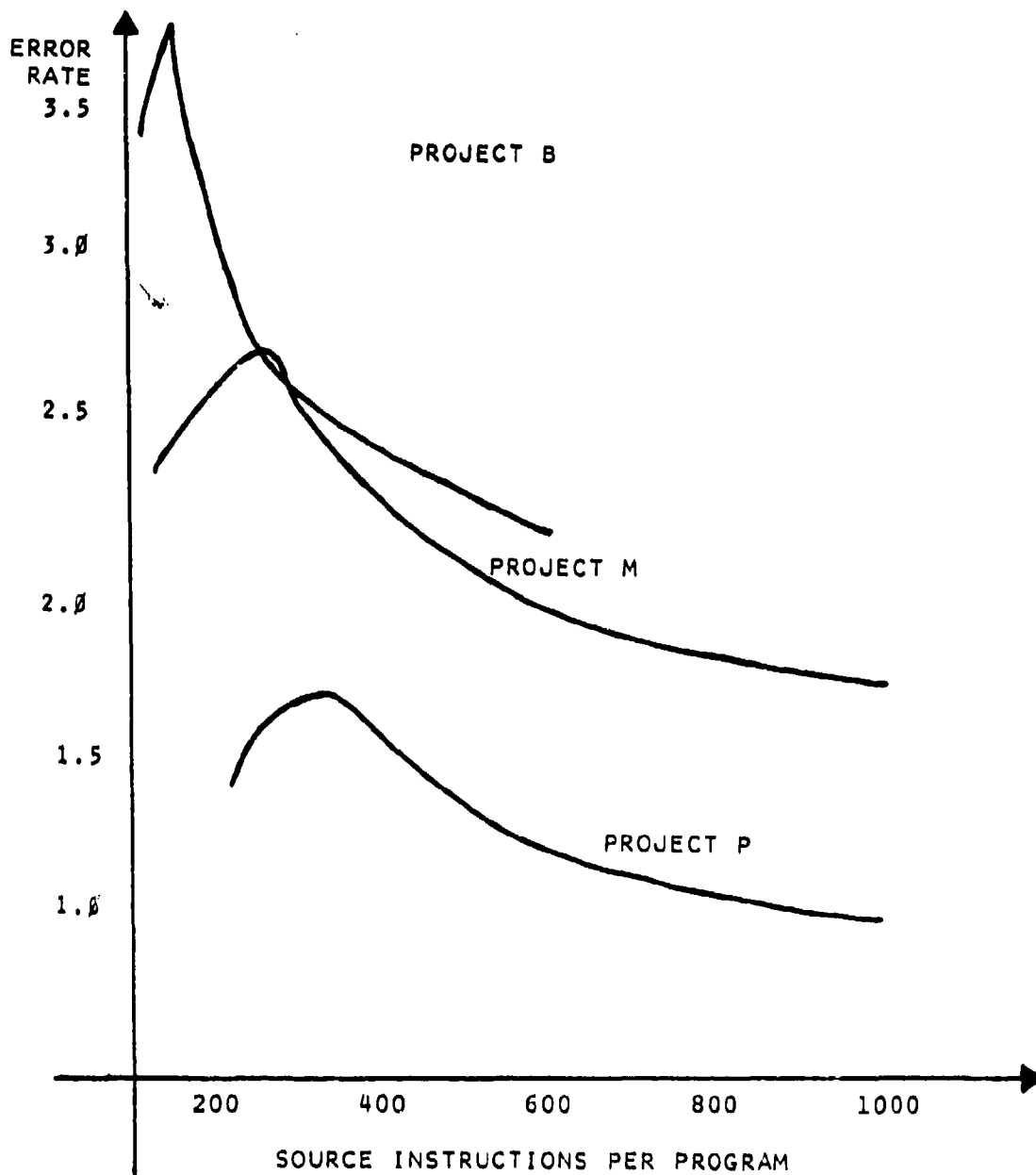


FIGURE 3-1. ERROR RATE AND SOURCE INSTRUCTION RELATIONSHIP FOR PROJECTS M, B, AND P

It is of course possible that other explanations (or hypotheses) would be better, such as, for these projects (including both samples) the shorter programs were in fact more complex, or that longer programs contain in general more repetitive operations (thus more easily programmed), but it is proposed here, as a result of evaluating the total analysis effort, that the hypothesis of inadequate testing of the longer programs is more likely to represent the actual situation.

Programs With Zero Reported Errors - Throughout most of the sets of program samples that were to be analyzed, zero reported errors were observed in long as well as short programs. Statistics on these zero reported errors programs were reported earlier in Table 2-4 for the projects of sample S and in Table 2-6 for the subsystems of sample T. For the longer programs having zero reported errors, some skepticism is warranted. It could have been that programming errors went unreported for these programs, or that the programs received very little testing. For the shorter programs with zero errors, it was considered that they could actually be error free as reported. However, given the earlier observed relationship between error rate and source instructions as indicated by Figure 3-1 and the correlations that were obtained, it is suggested that at least some of these programs underwent minimal testing.

It is believed then, that programs reported as error free constitute a set of programs, some of which are actually error free and some of which contain an unknown number of errors. Further, those reportedly error free programs are more likely to have more latent errors than those programs with some num-

ber of errors reported. (The same reasoning might also be applied to those programs with only one or two reported errors; however, the line must be drawn somewhere).

Throughout the analysis for both errors and error rate, results were obtained (1) leaving the zero error programs in the analysis, and also (2) excluding these programs. Performing the analysis in two ways, it was possible to determine whether an increase in the predictability of errors would result by eliminating one source of ambiguity in the data. At best, performing analysis in this way would be able to do justice to any researchers who would contend that if error prediction equations being developed are to be effective at predicting errors, then all programs used to develop these equations should have errors reported in them.

4.0 ANALYSIS METHOD AND PROCEDURE

4.1 Multiple Linear Regression Analysis

The method of analysis used in this study to predict programming errors was that of multiple linear regression. Using the model,

$$Y'_i = a + b_1X_{1i} + b_2X_{2i} + \dots + b_nX_{ni} \quad (4.0)$$

the observed programming errors (Y_i) were assumed to be predictable from a linear combination of the program characteristic predictor variables ($X_1, X_2, \dots, X_n$). In (4.0), $b_1, b_2, \dots, b_n$ are the regression coefficients; i.e., the estimated weights for each of the predictors that will maximize the predictability of errors, a is the intercept constant,

$$a = \bar{Y}_i - (b_1\bar{X}_{1i} + b_2\bar{X}_{2i} + \dots + b_n\bar{X}_{ni}) \quad (4.1)$$

i.e., the estimated value of Y_i at the point where the regression hyper-plane crosses the Y axis, and Y'_i represents the predicted or estimated value of errors for each individual program module (i). The quantities $\bar{Y}$ and $\bar{X}_i$ are mean values of the respective variables.

The method used to determine the parameters of the regression equation is to minimize the sum of squared deviations of actual errors from predicted errors; i.e., minimize

$$S = \sum e_i^2 = \sum [Y_i - (a + b_1X_{1i} + b_2X_{2i} + \dots + b_nX_{ni})]^2 \quad (4.2)$$

$$S = \sum (Y_i - Y'_i)^2 \quad (4.3)$$

In (4.3) the value of S is referred to as the sum of squares of deviations of the estimated value (Y'_i) of errors from the observed value (Y_i) of errors summed over all the program modules in each sample or set of observations for which errors are predicted.

Generally when prediction equations such as represented in (4.0) are being evaluated for their goodness of prediction, two statistics, (1) the multiple correlation coefficient (R) and (2) the squared multiple correlation coefficient (R^2), are used. The value of R has a range from 0 to 1 as indicated in (4.4),

$$0 \leq R_{Y.123\dots n} \leq 1 \quad (4.4)$$

and can be interpreted as the actual correlation between the linear combination of predictor variables and the observed values of errors (Y_i). The value of R^2 also has a range from 0 to 1 and is a measure of the proportion or percentage of variation in the dependent variable (Y_i) that can be accounted for or explained by the linear combination of predictors. More specifically R^2 can be represented by the ratio,

$$R^2_{Y.123\dots n} = \frac{\sum (Y'_i - \bar{Y})^2}{\sum (Y_i - \bar{Y})^2} \quad (4.5)$$

where $\bar{Y}$ is the mean or average value of the dependent variable in the given sample being analyzed.

As more improved or better predictions are obtained, the values of both R and R^2 will approach a value of 1. Complete lack of correlation (implying no predictability) results in an R or R^2 value of zero. Both statistics (R and R^2) will be

presented for all prediction equations in this study. For the purposes of this report, the goodness of prediction or what could be termed the predictability of programming errors will be discussed only in terms of R^2 .

Predictability will be described in terms of being low, moderate, or high in later sections of this report. These categories are based on the authors' judgement and are used for the descriptive purpose of comparing and contrasting the results of numerous error predictions obtained in this study.

For those readers interested in a more thorough and detailed discussion of regression analysis, texts by Draper and Smith (1968) (3) and Kerlinger and Pedhazur (1973) (6) are readable, cogently presented, and are highly recommended.

Although much more could be said here concerning multiple regression theory, practice, and procedures, the following two points deserve special emphasis.

Multicollinearity Among Predictor Variables - In many multiple regression procedures the computed values of a , b_1 , b_2 , ... and b_n are directly obtained from matrices which contain the intercorrelations among the predictor variables and the correlations of the predictor variables with errors. If the predictor variables are truly mathematically independent (i.e., not correlated with each other), then the problem of interpreting the final multiple regression equation becomes simple and the interpretation is straightforward. However, when the predictor variables are moderate to highly correlated with each other (which is usually the case), then a clear and explicit interpretation of the prediction equation becomes

comparatively much more difficult. When this situation arises, then the problem of multicollinearity is said to exist (Althauser, 1971 ; Gordon, 1968; Rama Sastry, 1970) (1, 5, 15). As indicated in Sections 2.0 and 3.0, many of the variables in both of our samples were moderately to highly correlated. Necessarily then in the 'Results' section of this report (Section 6.0), our interpretation of the prediction equations will consider interdependent variables.

Number of Predictor Variables and Sample Size - When the number of predictor variables available to study equals or exceeds the number of program-observations in the data sample, then perfect prediction of the criterion error measure (errors or error rate) will always result. Clearly for both data samples, there is a danger of encountering this problem. What was needed then to deal with this problem was to apply some analytical or automated statistical procedure to select the most powerful set of predictors for errors and error rate in each of the data samples. Two procedures were used. The first procedure involved the a priori elimination of predictor variables based on selected operational and statistical criterion. The criteria and the variables eliminated using these criteria are discussed in Section 5.0. The second procedure involved the use of a stepwise regression procedure that would enter, remove, swap, or delete variables from the regression equation based totally upon statistical criterion. This procedure will be discussed in the following section.

4.2 BMDP Stepwise Regression Procedure

The stepwise regression procedure selected to be used in this study was the procedure, referred to as BMDP2R, made available along with other statistical programs in the Biomedical Computer Program Package (BMDP) developed and distributed by the Health Sciences Computing Facility (HSCF) at the University of California, Los Angeles (UCLA). The HSCF supports Biomedical computer analysis for the School of Medicine at UCLA and is sponsored by a NIH Special Research Resources Grant RR-3. Although the BMDP series of programs has only been recently released (1975), it is based on an entire series of programs that have had a history of program revisioning, maintenance, modification, and testing that goes back to 1961. The BMD program series is one of the most widely used and highly reliable sets of statistical programs that has yet been developed for general purpose applications.

Essentially, the BMDP2R procedure estimates the parameters ($a, b_1, b_2, \dots, b_n$) of multiple linear regression equations in a stepwise manner. Four stepping algorithms are available. The algorithm selected for the current analysis is referred to as an FSWAP procedure, which enters and removes predictor variables based on an F-to-enter and a F-to-remove criterion with a variable interchange option. Without becoming too detailed here, the FSWAP procedure works as follows. Initially, the procedure selects the variable having the highest correlation with the dependent variable. It then adds the variable which has the next highest partial correlation with Y. After this addition, it continues adding variables which increase the multiple correlation coefficient (R), and concurrently swapping or exchanging variables not yet in the equation which

increase the multiple R value. Finally after no additional variables will increase the value of R, it then proceeds to perform a backwards elimination, removing those variables which have the least effect on the goodness of prediction. For additional information on this program, the reader should consult the actual BMDP publication manual (Dixon, 1975)

(2). (All computer runs using the BMDP2R program were performed on an IBM 370/168 computer.)

5.0 DEVELOPMENT OF ERROR PREDICTION EQUATIONS

The purpose of this section is to discuss various operational and analytical considerations and decisions that were made which pertain to the development of the error prediction equations that were obtained for this study.

5.1 Normalization of Predictor Variables

For both data samples, various transformations of selected variables were required. For example, for each program in both sample S and sample T, in order to normalize the original predictor variable values, each value was multiplied by a constant of 100 and divided by the total number of source instructions for that particular program (i.e., either X1 for sample S programs or TS for sample T programs). This normalization procedure thus resulted in a completely new set of predictor variables, in addition to the original set of predictors, in which each normalized variable could be interpreted as a unique measure of the original predictor variable given that the effect of the length of the program had been removed from it. In effect this procedure made each of the original variables comparable with respect to a "per 100 lines of source code" interpretation.

It is interesting to note that this particular normalization procedure did not completely eliminate the linear effect of total source instructions from all of the original variables. For most of the normalized variables, the correlation coefficients with Total Source Instructions were negative but relatively low or close to zero in magnitude. A few of the normalized variables had non-zero correlations with Total

Source Instructions which ranged from $-.50$ to $+.40$ in sample S and from $-.69$ to $+.72$ in sample T. Clearly then, this normalization procedure is not the most effective means for eliminating the effect of Source Instructions from each of the predictor variables. In actuality, the most statistically consistent and accurate procedure for eliminating the linear effect of one variable from one or a set of other variables is through partial correlation.

In the case of this analysis effort, use of the partial correlation procedure could have been alternatively employed by first predicting each original variable individually as a function of Total Source Instructions, and then using only the residual values (i.e., $Y_i - Y'_i$) as the values for the normalized variables that were employed in the prediction equations. If this procedure had been utilized, the resulting matrix of intercorrelations among all the normalized variables would in fact be a matrix of partial correlations; that is, the correlation between the various pairs of predictor variables with the linear effect of Total Source Instructions eliminated from both variables. The matrix of intercorrelations between each of the normalized variables and the original variables including the dependent variable, errors, would actually be a matrix of semi-partial correlations; i.e., the correlations between each of the normalized variables and the original variables including the dependent variable with the linear effect of Total Source Instructions eliminated from only the normalized variables.

Clearly, one implication of using this alternate procedure is that for the normalized variables, an entirely new data base must be constructed wherein each residual value must first be obtained from a simple linear regression procedure, automated,

and then entered in the data base. To be sure, constructing normalized variables using this procedure would indeed be time consuming and would require the availability of special-purpose software support programs that could minimize the amount of manual processing that would be required. Due to time constraints, this procedure was not followed.

5.2 Other Transformations to Predictor Variables

In addition to the normalization that was applied to the predictor variables for both samples, the only other transformations that were used were applied against several predictors of sample T. Since all data from sample T had been obtained in manual form for this study, all data values had to be recorded, keypunched, verified, and then automated. Most predictor variables of sample T were whole numbers, whereas the Loop Complexity (LL), If Complexity (IF), Programmer Rating (RAT), and Programmer Workload (WKLD) variables were whole numbers with one decimal value. In addition, Comments (COM) and Programmer Rating were originally represented as negative values. In order to simplify and expedite the manual to automated process for this data, the values of LL, IF, RAT, and WKLD were multiplied by a value of 10 in order to represent them as whole numbers. The values of COM and RAT were multiplied by -1 and -10, respectively, in order to eliminate the negative sign from both variables and represent the RAT predictor values as whole numbers.

Additionally, using the RAT and WKLD variables of sample T, a new predictor variable was constructed which took the form of RAT/WKLD. It was hypothesized that this new variable was linearly related to errors and as such should be included as

a candidate variable for consideration by the stepwise regression procedure. This newly constructed value was multiplied by a constant of 100 in order to represent it as a whole number throughout its range of values.

5.3 Combining Predictors In The Equations

As a direct result of the normalization procedure, 107 distinct variables versus the original 54 variables of sample S were considered as predictors of errors and error rate. For sample T a total of 30 as opposed to the original 16 variables were candidate predictors. Although the BMDP2R program is a stepwise procedure that selects the optimal predictors from among all those available to be entered, it is desirable to limit the number of predictors it would have to consider for any given set of predictions. This was desired in order to maximize the chance that as many predictors as possible could be considered in the regression equation simultaneously, and then eliminated one at a time using the backwards elimination procedure if the variable actually had no significant effect on the predictability of the dependent variable. In addition, the predictors available for selection by the regression procedure are limited because the total number of predictors should not exceed the sample size. In addition, reducing the number of predictors reduces the computer time required to generate each set of predictions.

For the sample T subsystems, allowing all 30 or less variables to be available for selection in the equation presented no major difficulty. Due to the a priori elimination of predictors that was carried out for sample T (to be discussed in Section 5.6), no more than 23 variables were ever allowed to be considered for selection. Essentially then for each subsystem of sample T, errors and error rate were predicted using the combination of variables as follows:

$$\begin{aligned}\text{Errors/program} &= f(\text{Program Structure} + \text{Programmer Variables}) \\ \text{Errors/program} &= f(\text{Program Structure Variables only}) \\ \text{Error rate/program} &= f(\text{Program Structure} + \text{Programmer Variables}) \\ \text{Error rate/program} &= f(\text{Program Structure Variables only})\end{aligned}$$

The program structure variables for these predictions represented the combination of both the unnormalized (TS, LL, IF, BR, ..., COM) and normalized (LL/TS, IF/TS, BR/TS, ..., COM/TS) predictor variables. Additionally, prediction equations were obtained first using the program observations available in each subsystem and then second, using only the remaining program-observations left after the zero reported error programs had been deleted from the analysis.

For the three projects of sample S, the 107 predictors were analyzed in two different sets each, for errors and error rate. These sets of predictors were combined as follows:

$$\begin{aligned}\text{Errors/program} &= f(\text{Unnormalized Variables}) \\ \text{Errors/program} &= f(\text{SI} + \text{Normalized Variables}) \\ \text{Error rate/program} &= f(\text{Unnormalized Variables}) \\ \text{Error rate/program} &= f(\text{SI} + \text{Normalized Variables})\end{aligned}$$

In these predictions SI represents the Source Instructions (SI) variable X1; the unnormalized variables are the predictors

X1, X2, ..., X54; and the normalized variables are the predictors X56, X57, ..., X107. Here, as with sample T, prediction equations were obtained both using all observations and using only the observations remaining after the zero reported error programs were deleted.

In any of these equations, due to the a priori variable elimination procedure that was applied, no more than 45 predictors per regression run were ever available to be selected for the sample S prediction equations.

5.4 Selection of Regression Coefficients to be Reported

After having made the necessary transformations (i.e., normalization and other transformations required) to the data and identifying how the predictors were to be combined in the analysis, one other issue remained. In essence, there were sets of predictors in both samples that would be equally or unequally weighted and non-comparable in a regression analysis. This follows from the facts that 1) the normalized variables had different units of measurement as compared with the unnormalized variables, and 2) for the predictors used in sample T, the program structure and programmer variables were not comparable, being derived from two distinct measurement domains. Thus, using predictors that had unequal or non-comparable units of measurement would result in making more difficult any relative comparisons among the raw regression coefficients computed for the predictors in the equation. In order to resolve this problem, it was decided that the standardized partial regression coefficients (Kerlinger and Pedhazur, 1973, p. 64) (8) would be reported for all predictions.

5.5 Regression Analysis Using Standardized Form of the Prediction Equation

In practice the standardized partial regression coefficients are referred to as B (beta) coefficients or beta weights as compared to the b coefficients as represented in equation (4.0). The beta weights are the regression coefficients that result when the raw data is transformed (i.e., standardized in this case) into standard score form prior to the analysis. For example, the standard score for the i th observation on a variable (X_i) is computed as follows:

$$z_i = \frac{X_i - \bar{X}_i}{s_i} \quad (5.0)$$

where $\bar{X}_i$ and s_i are the mean and standard deviation, respectively, for that variable. When all the predictor and dependent variables are standardized according to this procedure, the standardized variables all have a mean of 0 (i.e., $\bar{Z}_i = 0$) and a standard deviation of 1 (i.e., $s_z = 1$). Essentially then, the variability in each variable is made comparable with respect to the standard deviation as the common unit of measurement over all variables; thus, the B coefficients in the standard score form of the regression equation are also comparable. Although the raw data had not been standardized, the BMDP2R regression procedure computes the beta coefficients (in addition to the raw b coefficients).

Since the standard score form of the regression equation was being reported in this analysis, the linear regression model and other statistical formula became more easily interpreted in terms of the beta coefficients. For example, the

linear model now took the form,

$$Z'_{yi} = B_1 Z_{1i} + B_2 Z_{2i} + B_3 Z_{3i} + \dots + B_n Z_{ni} \quad (5.1)$$

and the multiple R , R^2 , and standard error of estimate s_y could be computed as follows:

$$R_{y.123\dots n} = \sqrt{B_1 r_{y1} + B_2 r_{y2} + B_3 r_{y3} + \dots + B_n r_{yn}} \quad (5.2)$$

$$R^2_{y.123\dots n} = B_1 r_{y1} + B_2 r_{y2} + B_3 r_{y3} + \dots + B_n r_{yn} \quad (5.3)$$

$$s_y = \sqrt{1 - R^2_{y.123\dots n}} \quad (5.4)$$

where the r_{yi} values are the correlations of each predictor (i) with the dependent variable (y).

Furthermore, since the results of any correlational analysis are the same whether the analysis started with the raw data values or the standard score values, computational formula are available (Kerlinger and Pedhazur, 1973, pp. 61-62) (8) which easily allow the computation of the value of the b coefficients and the a intercept in equation (4.0) using the beta coefficients obtained in this analysis. For example, the raw regression coefficients b_j and the a intercept can be directly computed using the following formula:

$$b_j = B_j \left(\frac{s_y}{s_j} \right) \quad (5.5)$$

$$\text{and} \quad a = \bar{Y} - B_1 \left(\frac{s_Y}{s_1} \right) (\bar{X}_1) - B_2 \left(\frac{s_Y}{s_2} \right) (\bar{X}_2) - \dots - B_n \left(\frac{s_Y}{s_n} \right) (\bar{X}_n) \quad (5.6)$$

$$a = \bar{Y} - \sum B_j \left(\frac{s_Y}{s_j} \right) (\bar{X}_j) \quad (5.7)$$

where s_Y and s_j are the standard deviations of the dependent (y) and predictor variables (j), respectively, and $\bar{Y}$ and $\bar{X}_j$ are the means for the dependent and predictor variables, respectively.

5.6 A Priori Elimination of Predictor Variables

Certain predictor variables in both data samples were eliminated from any further consideration in the prediction analysis, prior to their actual consideration for selection by the BMDP2R regression procedure. In general, most of the predictors that were eliminated at this early stage of analysis were done so in order to reduce the incidence of multicollinearity that exists among the predictors. Other variables were eliminated primarily because they had zero values throughout the data sample. Table C-1 in Appendix C lists all the 107 predictors of errors and error rate in sample S and identifies each of the variables that were either eliminated a priori from the analysis, or made available to be considered for selection in the regression procedure. The criteria used to eliminate these variables prior to the regression analysis are enumerated at the end of this table. For the 30 predictors of sample T, Tables C-2 and C-3 in Appendix C provide similar information for the predictors of errors/program and error rate/program, respectively.

6.0 ERROR PREDICTION EQUATIONS: RESULTS AND DISCUSSION

The purpose of this section is to present and discuss the actual prediction equation results that were obtained when predicting errors/program and error rate/program, respectively, for each of the three projects of sample S and each of the eight subsystems of sample T. Preliminary to this presentation of results for each sample, an overall general summary of results with discussion is provided.

6.1 Results Summary

The following are the major results obtained regarding the predictability of errors and error rate over both of the samples that were analyzed.

Errors/Program

- For sample T, where the error data had been collected throughout the validation, acceptance, integration, and operational testing phases of software system development, errors/program were found to be moderately to highly predictable. This predictability was far from perfect with 76% to 93% of the variance accounted for when the errors/program were predicted from a linear combination of program structure variables.
- For sample S, where the error data had been collected only during the test and integration phase of software system development, errors/program were found to be less consistently predictable. The percent of variance accounted for in this sample was 59% to 90%.

- Predictor variables which reflected length of program were generally found to be the best single predictors of errors/program. However, other program structure-complexity variables together in combination with length of program variables contributed significantly to the predictability of errors/program.

Error Rate/Program

- For sample T, error rate/program was found to be less predictable in general than errors/program, with 59% to 85% of the variance accounted for, when predicted from a linear combination of program structure variables.
- For sample S, the predictability of error rate/program was generally, lower, with 34% to 94% of the variance accounted for.
- Predictor variables which were measures of the number of program interfaces per 100 lines of source code were generally found to be the best single predictors of error rate/program. However, other normalized measures of program complexity together in combination with program interfaces per 100 lines of source code, contributed significantly to the predictability of error rate/program.

- In general, the predictor variables which were most frequently selected by the stepwise regression procedure as contributing significantly to the predictability of error rate/program were the normalized variables.
- Analysis of the error rate/program measure for those data samples having a high percentage of error-free programs leads to a clear indication of the lack of thoroughness of testing in these reportedly error free programs.

6.2 Discussion

Clearly, the results obtained from the analysis of errors and error rate are not surprising. The facts that (1) length of program and the number of program interfaces per 100 lines of source code were found to be the best single predictors for errors and error rate, respectively, and that (2) program complexity variables contributed significantly to the predictability of each dependent variable, are findings that not only appeal to experience and intuitive judgement about how these predictor variables may be related to measures of programming errors, but also are findings which are supported by other empirical studies concerned with software reliability (Mitchell et al., 1976; Thayer et al., 1976, Okimoto, 1975) (11, 13, 14).

For example, consider the following hypotheses, which logically follow from our knowledge and experience of programming, which concern the effect of increasing program length and program complexity on the total number of programming errors in the program.

Hypothesis 1: As length of program increases, program complexity increases, and the number of latent errors in the program increases linearly as a function of both.

Hypothesis 2: As length of program increases, program complexity increases, and the number of probable or latent errors in the program increases at an exponential rate.

Hypothesis 3: As length of program increases, program complexity increases; redundancy in the use of similar software functions and instructions in the program also increases, resulting in the number of latent errors in the program increasing up to a point with no significant increase thereafter with increasing program length.

Each of these three hypotheses are graphically depicted in Figure 6-1.

Hypothesis 1, which in essence is the basic assumption of the multiple linear regression model used in this analysis, can generally be accepted as one explanation for the high degree of predictability obtained when predicting errors/program in this study. However, hypotheses 2 and 3 cannot be rejected by this analysis. This is clear for several reasons:

- (1) the model investigated by this analysis was a linear model, and not an exponential or curvilinear model as are suggested by hypotheses 2 and 3, respectively;

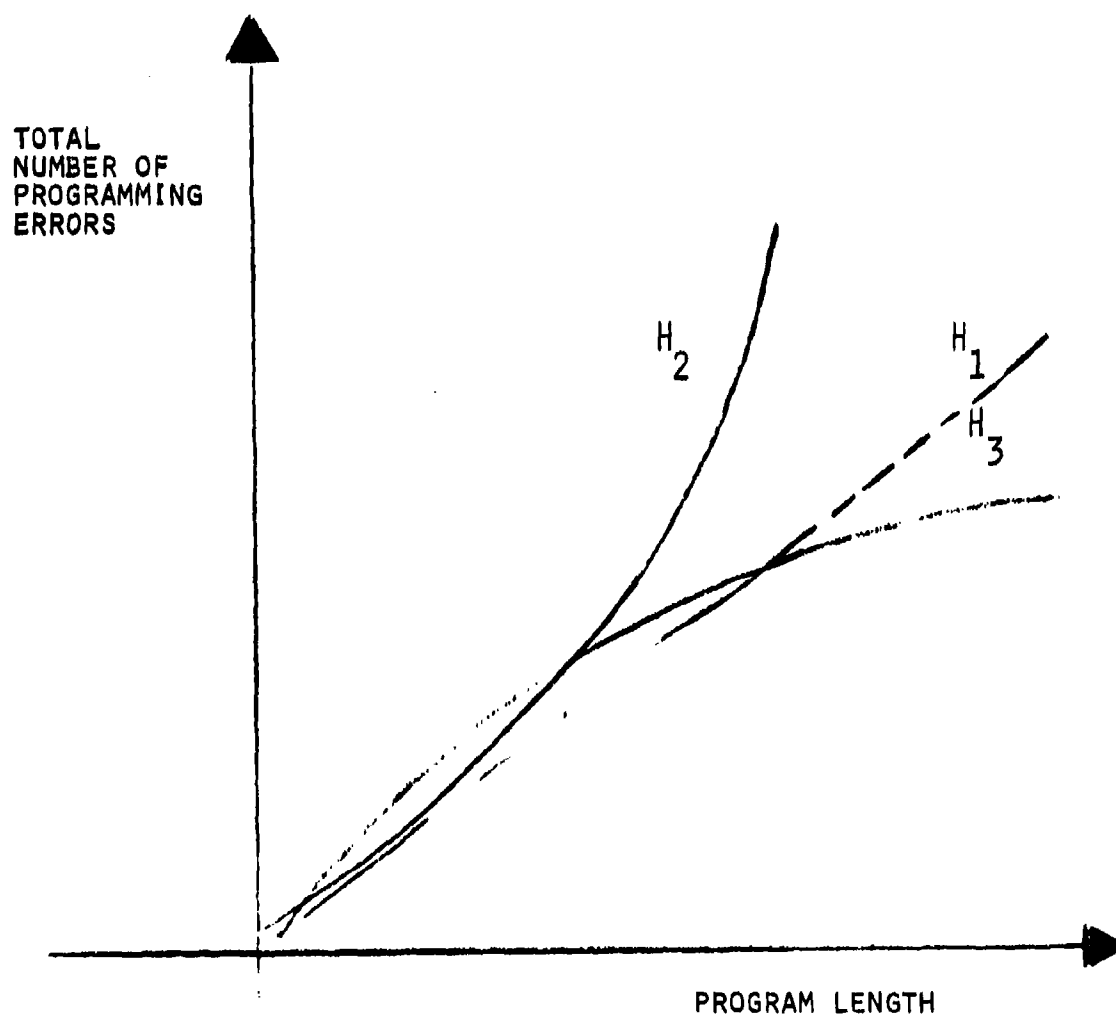


FIGURE 6-1. DEPICTION OF HYPOTHETICAL RELATIONSHIPS BETWEEN PROGRAMMING ERRORS AND LENGTH OF PROGRAM

- (2) the linear model did not yield perfect predictability of errors over each of the samples that were analyzed; and
- (3) the numerous program structure and complexity metrics that were used as predictor variables in the study would possibly require re-evaluation or re-formulation, with the chance that some variables would be excluded and new ones included, in order to use them to appropriately investigate hypotheses 2 and 3.

With respect to the finding that program complexity measures and the number of program interfaces contributed significantly to the predictability of error rate, studies by Okimoto and Thayer (13, 14), are of interest. Okimoto found in a survey of approximately 60 systems programmers that when they were asked to rank order the ten most important factors contributing to error-proneness in programs, poorly defined interfaces was ranked as the highest and most important factor. Other factors of importance that were included in this list of contributors to error-proneness were (1) poor/incomplete testing (ranked 3rd), (2) complex function/logic (ranked 8th), and (3) large modules (ranked 10th).

In the Thayer et al. study, among the many things that were presented was a brief analysis of factors which contributed to the difficulty of developing over 200 command and control programs. Each program was rated according to five categories of difficulty: difficulty to design, code, implement, checkout, and document. These ratings were then summed to obtain the overall difficulty rating for each program. Of particular interest were the major reasons given for the difficult to

develop programs: complex logic, core loading problems, and data interfaces. A fourth reason, changes in interpretation of poorly stated requirements, which adds to complexity and difficulty in program development was also given.

6.3 Sample S Results

6.3.1 Errors/Program = f(Unnormalized Variables)

The prediction equation results for both five and ten predictor variables, when predicting errors/program from a linear combination of the unnormalized variables for each project of sample S, are presented in Tables 6-1 thru 6-5. These tables report

- (1) the standard partial regression coefficients (i.e., the beta coefficients, not the raw regression coefficients) for each predictor,
- (2) the correlation of each predictor with errors,
- (3) the highest value of R and R^2 obtained for the maximum number of predictors entered in the equation by the regression procedure, and
- (4) the analysis of variance tables for both five and ten predictor regression equations.

Prediction equation results for projects M and B where zero errors were deleted from the analysis are reported in Tables 6-2 and 6-4, respectively. Additionally, Tables 6-6 and 6-7 are included here to summarize the prediction results obtained for both five and ten predictors over all regression equations that were developed using the unnormalized variables as predictors of errors. For projects M, B, and P, a total of 45, 43, and 45 predictor variables, respectively, were available

TABLE 6-1. PROJECT M,
ERRORS/PROGRAM = 2 (UNNORMALIZED VARIABLES)

VARIABLES IN PREDICTION EQUATION			
	5	10	25 (MAX.)
MULTIPLE R ²	.717	.768	.797
MULTIPLE R ²	.514	.589	.636
STD. ERROR OF ESTIMATE	7.449	6.892	6.621

VARIABLES (X)	COEFFICIENTS		R _{XY}
X1 SOURCE INSTRUCTIONS	---	-1.280	.563
X2 ENTRY POINTS	-.175	-.146	.150
X4 USING INSTRUCTIONS	.191	.290	.507
X9 CALLS/LINKS	---	.105	.374
X12 EQUATE STATEMENTS	-.181	-.237	.362
X14 LOGICAL CONNECTORS	---	-.131	.268
X15 CONDITIONAL JUMPS	---	.603	.593
X16 FUNCTIONS	---	.182	.515
X20 LOCK MACROS	.106	---	.309
X37 UNDEFINED VARIABLES	.734	.712	.645 ¹¹
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.	---	.601	.590

ANALYSIS OF VARIANCE						
NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	p
5	REGRESSION	22794.547	5	4558.906	82.150	<.001
	RESIDUAL	21585.109	389	55.489		
10	REGRESSION	26140.555	10	2614.055	55.035	<.001
	RESIDUAL	18239.102	384	47.497		
	TOTAL	44379.662	394			

¹¹ BEST SINGLE PREDICTOR

TABLE 6-2. PROJECT M,
ERRORS/PROGRAM = 2 (UNNORMALIZED VARIABLES),
ZERO ERRORS DELETED

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>24(MAX.)</u>
MULTIPLE R ₂	.695	.755	.791
MULTIPLE R ²	.483	.571	.625
STD. ERROR OF ESTIMATE	8.215	7.552	7.228

VARIABLES (X)	COEFFICIENTS		F _{XY}
X1 SOURCE INSTRUCTIONS	---	-.929	.534
X2 ENTRY POINTS	-.192	-.148	.115
X4 USING INSTRUCTIONS	.174	.283	.456
X12 EQUATE STATEMENTS	-.183	-.227	.355
X13 COMMENTED INSTRUCTIONS	---	.139	.353
X15 CONDITIONAL JUMPS	---	.659	.584
X16 FUNCTIONS	---	.238	.465
X20 LOCK MACROS	.104	---	.290
X37 UNDEFINED VARIABLES	.738	.647	.624 ¹¹
X42 NON-NESTED DO LOOPS	---	.158	.452
X53 INSTR., 6TH LEVEL OR LOWER, DO LOOPS	---	.102	.147

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	?
5	REGRESSION	18866.531	5	3773.306	55.919	<.001
	RESIDUAL	20175.902	299	67.478		
10	REGRESSION	22274.613	10	2227.461	39.055	<.001
	RESIDUAL	16767.816	294	57.033		
	TOTAL	39042.434	304			

¹¹BEST SINGLE PREDICTOR

TABLE 6-3. PROJECT B,
ERRORS/PROGRAM = 2 (UNNORMALIZED VARIABLES)

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>19 (MAX.)</u>
MULTIPLE R ₂	.812	.853	.886
MULTIPLE R ²	.660	.727	.785
STD. ERROR OF ESTIMATE	3.615	3.322	3.110

VARIABLES (X)	COEFFICIENTS		R _{XY}
X4 USING INSTRUCTIONS	---	.245	.436
X8 UNCONDITIONAL JUMPS	---	.172	.294
X15 CONDITIONAL JUMPS	-.443	-.557	.582
X17 SCALING/ROUNDING OPNS.	.211	.219	.589
X22 ADDRESS VARIABLES	---	.209	.319
X28 FIXED POINT VARIABLES FREQ.	---	.197	.498
X35 REGISTER VARIABLES	---	-.294	.409
X37 UNDEFINED VARIABLES	.631	.588	.678 ^{**}
X49 INSTR., 2ND LEVEL DO LOOPS	.305	.313	.668
X53 INSTR., 6TH LEVEL OR LOWER DO LOOPS	.286	.230	.565

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	P
5	REGRESSION	2483.154	5	496.631	38.000	<.001
	RESIDUAL	1280.795	98	13.160		
10	REGRESSION	2737.610	10	273.761	24.806	<.001
	RESIDUAL	1127.330	93	12.136		
	TOTAL	3763.949	103			

^{**} BEST SINGLE PREDICTOR

TABLE 6-4. PROJECT B,
 ERRORS/PROGRAM = 2 (UNNORMALIZED VARIABLES),
 ZERO ERRORS DELETED

VARIABLES IN PREDICTION EQUATION			
	<u>5</u>	<u>10</u>	<u>14(MAX.)</u>
MULTIPLE R ₂	.834	.893	.919
MULTIPLE R ²	.695	.797	.845
STD. ERROR OF ESTIMATE	3.418	2.869	2.572

VARIABLES (X)	COEFFICIENTS		R _{XY}
X4 USING INSTRUCTIONS	---	.290	.658
X6 LABELED INSTRUCTIONS	---	.275	.447
X11 USER MACROS	---	.431	.401
X15 CONDITIONAL JUMPS	-.493	-.335	.578
X18 SHORT DO LOOPS	---	.194	.112
X27 FIXED POINT VARIABLES	.154	.286	.333
X37 UNDEFINED VARIABLES	.695	.700	.685
X49 INSTR., 2ND LEVEL DO LOOPS	.351	.482	.688
X53 INSTR., 6TH LEVEL OR LOWER DO LOOPS	.359	.474	.590
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.	---	-1.056	.604

ANALYSIS OF VARIANCE						
NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	F
5	REGRESSION	2371.739	5	474.348	40.602	<.001
	RESIDUAL	1039.876	89	11.683		
10	REGRESSION	2720.190	10	172.019	33.051	<.001
	RESIDUAL	691.336	84	8.230		
	TOTAL	3411.526	94			

"BEST SINGLE PREDICTOR

TABLE 6-5. PROJECT P,
ERRORS/PROGRAM = 1 (UNNORMALIZED VARIABLES)

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>17 (MAX.)</u>
MULTIPLE R ₂	.888	.946	.988
MULTIPLE R ²	.788	.895	.977
STD. ERROR OF ESTIMATE	2.613	2.022	1.130

VARIABLES (X)	COEFFICIENTS		R _{xy}
X4 USING INSTRUCTIONS	---	-.465	.352
X5 COMMENT STATEMENTS	.317	.762	.509
X8 UNCONDITIONAL JUMPS	.748	.544	.864 ^u
X17 SCALING/ROUNDING OPNS.	---	.176	.024
X23 ADDRESS VARIABLE FREQ.	-.379	-.386	.168
X28 FIXED POINT VARIABLE FREQ.	.324	.600	.242
X37 UNDEFINED VARIABLES	---	.436	.567
X42 NON-NESTED DO LOOPS	---	-.386	.179
X47 DO LOOPS, 6TH LEVEL OR LOWER	.336	---	.345
X51 INSTR., 4TH LEVEL DO LOOPS	---	-.374	.424
X53 INSTR., 6TH LEVEL OR LOWER DO LOOPS	---	.602	.341

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	P
5	REGRESSION	737.633	5	147.527	21.612	<.001
	RESIDUAL	107.962	20	6.826		
10	REGRESSION	837.431	10	83.743	20.474	<.001
	RESIDUAL	98.165	24	4.090		
	TOTAL	935.596	34			

^u BEST SINGLE PREDICTOR

TABLE 6-6. FIVE PREDICTOR SUMMARY,
ERRORS/PROGRAM = f(UNNORMALIZED VARIABLES)

VARIABLES	M ^a	M ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X37 UNDEFINED VARIABLES	.734**	.738**	.588**	.350**	.748**
X49 INSTR., 2ND LEVEL DO LOOPS					
X8 UNCONDITIONAL JUMPS					
<u>CORRELATION STATISTICS</u>					
r _{SI} , ERRORS	.563	.534	.649	.658	.593
r _{SI} , SI	.880	.872	.859	.727	.337
r _{SI} , ERRORS	.645	.624	.678	.688	.664
r _{SI} , ERRORS	.416	.389	.460	.473	.441
<u>PREDICTION SUMMARY</u>					
R ₂	.717	.695	.812	.834	.888
R	.514	.483	.660	.695	.788

^a ALL OBSERVATIONS USED

^b ZERO ERRORS DELETED

** BEST SINGLE PREDICTOR

TABLE 6-7. TEN PREDICTOR SUMMARY,
ERRORS/PROGRAM = 2 (UNNORMALIZED VARIABLES)

VARIABLES	M ^a	M ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X4 USING INSTRUCTIONS	.290	.283	.245	.290	-.465
X37 UNDEFINED VARIABLES	.712 ¹¹	.647 ¹¹	.588 ¹¹	.700	.436
X15 CONDITIONAL JUMPS	.603	.659	-.557	-.335	
X53 INSTR., 6TH LEVEL DO LOOPS		.102	.200	.474	.602
X1 SOURCE INSTRUCTIONS	-1.280	-.920			
X2 ENTRY POINTS	-.146	-.148			
X3 UNCONDITIONAL JUMPS			.172		.544 ¹¹
X12 EQUATE STATEMENTS	-.237	-.227			
X17 SCALING/ROUNDING OPNS.			.219		-.176
X23 FIXED PT. VAR. FREQ.			.197		.600
X42 NON-NESTED DO LOOPS		.150			-.536
X49 INSTR., 2ND LEVEL DO LOOPS			.313	.474 ¹¹	
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.	.601			-1.056	
<u>CORRELATION STATISTICS</u>					
r ² _{SI, ERRORS}	.563	.534	.649	.658	.593
r ² _{11, SI}	.880	.872	.859	.727	.337
r ² _{11, ERRORS}	.645	.624	.670	.688	.664
r ² _{11, ERRORS}	.416	.380	.460	.473	.441
<u>PREDICTION SUMMARY</u>					
R ₂	.766	.755	.853	.913	.946
R ²	.580	.571	.707	.707	.895

^aALL OBSERVATIONS USED

^bZERO ERRORS DELETED

¹¹BEST SINGLE PREDICTOR

to be automatically entered in the prediction equation (see Table C-1, Appendix C).

Initial observation of these results, particularly the R^2 values as reported in Tables 6-6 and 6-7, indicates that (1) for all sample S projects the increase in R^2 ; i.e., the percent of variance accounted for when predicting errors, shows a moderate increase when ten predictors are used as opposed to when only five are used; and (2) the predictability of errors ranges from moderate for project M ($R^2 = .589$) to high for project P ($R^2 = .895$). One can also observe (see Tables 6-1 thru 6-5) that the variables selected for the five predictor solution are predictors which generally appear again in the ten predictor solution, after five different predictors have been entered.

Clearly the predictability of errors in sample S is not consistently high over each project. Two predictors however, Using Instructions and Undefined Variables, do appear consistently in each of the ten predictor equations for each project. Other variables as indicated in Table 6-7 do appear less consistently either for two or more of the prediction equations that were developed, or for at least two of the projects of sample S. The fact that variables may appear consistently over two or more projects, however, does not necessarily mean that they are the most important variables from a prediction viewpoint.

The best single predictor of errors in each project is evaluated by its correlation with errors and not its beta coefficient in the regression equations. For projects M and B, Undefined Variables was the best single predictor of errors.

For project P the best single predictor was the number of Unconditional Jumps in the program. However, in Table 6-7 when one examines the percent of variance in errors explained by the best single predictor (designated as $r_{*,\text{errors}}^2$) versus the percent of variance explained by the best single predictor plus other program complexity variables (i.e., R^2), it is observed that the other program complexity variables selected by the regression procedure are significantly contributing to the predictability of errors over and above that which can be predicted only by the best single predictor in any of the projects.

One additional point is important enough to mention here. In Table 6-7 the correlation statistics are reported for the correlations between the best single predictors and Source Instructions (SI). In general, all of these correlation coefficients are moderate to high with the exception of the correlation of project P's best single predictor, Unconditional Jumps, with SI. This clearly shows that each best single predictor is itself reflecting length of program plus some contribution to the prediction of errors which is unique to the variable itself. In essence then, given that these best single predictors had not been used, source instructions by itself could be used to predict errors almost as well as each best single predictor.

With respect to error predictability, when zero errors are left in or taken out of the data samples (see Tables 6-1 thru 6-4), predictability of errors increases appreciably (from .73 to .80) for project B, but no change of any statistical significance (a decrease from .59 to .57) occurs for project M. Although only nine zero error observations

were deleted from project B's data sample, this deletion was sufficient to cause changes in the correlations among predictors and between each of the predictors and errors, such that an almost completely different prediction equation was produced, wherein only three of the ten original predictors were reselected. For project M, 90 zero error observations were deleted and seven out of ten original predictors in the regression equation using all observations (i.e., 395) stayed in the equation; however, with slightly different coefficients.

Several explanations for these results can be suggested at this time. First, as earlier hypothesized in Section 3.0, programs with zero reported errors are programs that have not been thoroughly tested and contain latent errors. Zero reported errors in these cases are actually under-estimates of the total errors in the program, and if these programs are used along with programs that have errors reported in them for error prediction purposes, then estimates of error predictability; i.e., values of R^2 , will be biased downward. This is in fact what was observed for project B.

On the other hand, for project M, since the change in R^2 was so slight (i.e., - .02) and since seven of the ten original predictors remained in the equation, this suggests the possibility that the 90 zero error programs that were deleted might have been the shorter programs which had only a limited amount of variability in their predictor variables. In other words, predictor variables which may have limited variability to contribute to the prediction of errors, when eliminated from the analysis, do not significantly change or affect the results. However, a validation of this suggested hypothesis for the 90

error free programs of project M can only be attempted by a complete analysis of these programs, which was not the intent or purpose of this study.

Interpreting the regression equations presented in Tables 6-1 to 6-5 presents some difficulty. There are several reasons for this. One basic reason is that beyond the description of each variable that is provided in Table B-1, there is no more definitive understanding of what each variable is measuring, not to mention how combinations of these variables should conceptually interact to influence errors in programs. One predictor variable that would stand as an exception for which an interpretation could be given in this case is X5, Comment Statements. Comment Statements appears in Table 6-5 as a predictor with positive coefficient for both the five and ten predictor regression equations for project P. One would initially expect that since comment statements are not executable, they cannot contribute to errors in programs. There is no disagreement with this explanation. However, one possible explanation for this variable appearing in the equations for project P could be that these programs may have had a comparatively larger number of comment statements in them relative to programs of project M and B. As such, this increased the readability of the programs, which resulted in more errors being found because they were more easily detected.

An additional reason why no straight-forward interpretation of the regression equations can be attempted is that each of the predictors that were selected were usually found to be correlated with other variables in the equation. This was particularly evident for the results presented in Table 6-1 for project M. For example, one might ask the reason for the

negative beta coefficients for variable X1, Source Instructions, when it is reported as being positively correlated with errors. Statistically, there is an explanation for the large negative weight being computed for Source Instructions. Other predictors, particularly those that had entered the equation prior to Source Instructions, were moderately to highly correlated with Source Instructions. The correlation coefficients of each of the predictors with Source Instructions is reported below:

X2	X4	X9	X12	X14	X15	X16	X37	X54
.352	.535	.627	.633	.425	.910	.646	.880	.976

After the first variable had entered the equation (i.e., variable X37, the best single predictor of errors), the partial correlation of Source Instructions with errors was very close to zero. As such, when Source Instructions entered the equation which was at step 7 in the regression procedure, it contributed to the predictability of errors (i.e., R^2) by suppressing the effect of Source Instructions from other variables that had already been entered in the equation. This then is the reason for the large negative weight being computed for Source Instructions. By suppressing or taking out this effect, the predictive power of the other variables was increased, as was evidenced by positive changes in these predictor's coefficients. In essence then variable X1 is functioning as a suppressor variable in the equation of Table 6-1. Suppressor variables are quite common in the social sciences and in other areas of study wherein measurement tools are either unavailable or have not been developed for providing unique and independent measures of predictor variables for data collection and analysis purposes. (Suppressor variables generally appeared throughout the results for both samples S and T). For a more detailed discussion of suppressor variables the reader should consult

the following references (McNemar, 1962; Van de Geer, 1971; Harris, 1975) (6, 10, 16).

Finally, when there is a moderate or greater degree of multicollinearity among the predictors in the regression equation, making an assessment of the relative importance of the independent variables for predicting errors is not readily accomplished by inspecting only the beta coefficient for a given predictor (Ferguson, p. 402) (4). For example, with two predictors the value of R^2 can be shown to be equal to

$$R^2 = B_1^2 + B_2^2 + 2B_1B_2r_{12} \quad (6.0)$$

In this case the predicted variance is comprised of three additive parts. B_1^2 represents a contribution by predictor X_1 , B_2^2 a contribution by X_2 , and $2B_1B_2r_{12}$ is a component which involves the correlation between X_1 and X_2 . Clearly then, evaluating the relative contribution of a predictor in the multiple regression equation requires that correlation terms and other predictor's beta coefficients be considered simultaneously. This would not be the case, however, had each of the predictors been statistically independent of other variables in the equation.

6.3.2 Errors/Program = f(SI + Normalized Variables)

As previously discussed in Section 5.0, prediction equations for errors/program were also developed using the normalized variables as predictors in combination with the Source Instructions (SI) variable, X_1 . The results obtained from these predictions provide information on how effectively the original variables, once the effect of Source Instructions had been "removed" from each predictor, combined with X_1 to predict errors. The prediction equation results are presented in Table 6-8 through 6-12. Summary results for five and ten

TABLE 6-8. PROJECT M,
ERRORS/PROGRAM = $f(SI + \text{NORMALIZED VARIABLES})$

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>18(MAX.)</u>
MULTIPLE R^2	.622	.641	.654
MULTIPLE R^2	.387	.411	.428
STD. ERROR OF ESTIMATE	8.362	8.251	8.220

VARIABLES (X)		COEFFICIENTS		R_{xy}
X1	SOURCE INSTRUCTIONS	.597	.564	.563**
X57	EXIT POINTS/SI	---	-.098	-.213
X58	USING INSTRUCTIONS/SI	.153	.131	-.125
X69	CONDITIONAL JUMPS/SI	.148	.123	.132
X70	FUNCTIONS/SI	.195	.155	.202
X71	SCALING/ROUNDING OPNS./SI	---	.070	.163
X74	LOCK MACROS/SI	.179	.191	.192
X82	FIXED PT. VAR. FREQ/SI	---	-.063	-.043
X90	REGISTER VAR. FREQ./SI	---	-.099	-.171
X108	SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR./SI	---	.067	.116

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	P
5	REGRESSION	17178.586	5	3435.717	49.134	<.001
	RESIDUAL	27201.070	389	69.926		
10	REGRESSION	18234.863	10	1823.486	26.782	<.001
	RESIDUAL	26104.793	334	68.135		
	TOTAL	44379.661	394			

** BEST SINGLE PREDICTOR

TABLE 6-9. PROJECT M,
 ERRORS/PROGRAM = $\sum(SI + \text{NORMALIZED VARIABLES})$,
 ZERO ERRORS DELETED

VARIABLES IN PREDICTION EQUATION

	<u>I</u>	<u>II</u>	<u>III(MAX.)</u>
MULTIPLE R^2	.598	.617	.619
MULTIPLE R^2	.357	.381	.383
STD. ERROR OF ESTIMATE	9.161	9.060	9.065

<u>VARIABLES (X)</u>		<u>COEFFICIENTS</u>		<u>r_{xy}</u>
X1	SOURCE INSTRUCTIONS	.559	.549	.533 ^{**}
X57	EXIT POINTS/SI	---	-.383	-.195
X58	USING INSTRUCTIONS/SI	.136	.149	-.161
X61	ARITHMETIC INSTRUCTIONS/SI	---	-.366	-.075
X69	CONDITIONAL JUMPS/SI	.236	.169	.246
X71	FUNCTIONS/SI	.169	.147	.142
X74	LOCK MACROS/SI	.194	.176	.166
X87	LABELED ARRAY VARIABLES/SI	---	-.164	-.113
X92	UNDEFINED VAR. FREQ./SI	---	.194	.113
X118	SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR./SI	---	.089	.162

ANALYSIS OF VARIANCE

<u>NO.</u>		<u>SUM OF</u>		<u>MEAN</u>		
<u>PREDICTORS</u>		<u>SQUARES</u>	<u>DF</u>	<u>SQUARE</u>	<u>F</u>	<u>F</u>
5	REGRESSION	13949.352	5	2789.873	33.245	<.001
	RESIDUAL	25392.578	299	83.922		
11	REGRESSION	14862.613	11	1486.261	18.371	<.001
	RESIDUAL	24173.816	294	82.244		
	TOTAL	39342.434	304			

** BEST SINGLE PREDICTOR

TABLE 6-10. PROJECT B,
 ERRORS/PROGRAM = $2(SI + \text{NORMALIZED VARIABLES})$
VARIABLES IN PREDICTION EQUATION

	5	10	14(MAX.)
MULTIPLE R^2	.717	.754	.772
MULTIPLE R^2	.513	.569	.596
STD. ERROR OF ESTIMATE	4.323	4.178	4.136

VARIABLES (X)	COEFFICIENTS		r_{xy}
X1 SOURCE INSTRUCTIONS	.587	.651	.649 ¹¹
X57 EXIT POINTS/SI	---	.169	-.057
X62 UNCONDITIONAL JUMPS/SI	---	.110	-.090
X71 SCALING/ROUNDING OPNS./SI	---	.256	.344
X82 FIXED PT. VAR. FREQ./SI	.164	---	.358
X83 FLOATING PT. VARIABLES/SI	---	.160	-.065
X89 REGISTER VARIABLES/SI	---	.114	-.316
X91 UNDEFINED VARIABLES/SI	---	.269	-.278
X99 DO LOOPS, NESTED AT 4TH LEVEL/SI	-.446	-.293	.143
X103 INSTR., 2ND LEVEL DO LOOPS/SI	.193	.243	.241
X105 INSTR., 4TH LEVEL DO LOOPS/SI	.433	.422	.210

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUAPE	F	p
5	REGRESSION	1932.146	5	386.429	20.674	<.001
	RESIDUAL	1831.804	98	18.692		
10	REGRESSION	2140.797	10	214.080	12.266	<.001
	RESIDUAL	1623.151	93	17.453		
	TOTAL	3753.949	103			

¹¹ BEST SINGLE PREDICTOR

TABLE 6-11. PROJECT B,
ERRORS/PROGRAM = $f(SI + \text{NORMALIZED VARIABLES})$,
ZERO ERRORS DELETED

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>12(MAX.)</u>
MULTIPLE R^2	.735	.777	.787
MULTIPLE R^2	.541	.604	.619
STD. ERROR OF ESTIMATE	4.196	4.089	3.982

VARIABLES (X)		COEFFICIENTS		r_{xy}
X1	SOURCE INSTRUCTIONS	.646	.608	.658*
X58	USING INSTRUCTIONS/SI	---	.208	-.280
X59	COMMENT STATEMENTS/SI	---	.264	-.335
X65	USER MACROS/SI	---	.181	.051
X76	ADDRESS VARIABLES/SI	---	.103	.137
X82	FIXED PT. VAR. FREQ./SI	---	.122	.366
X91	UNDEFINED VARIABLES/SI	.192	.248	-.290
X133	INSTR., 2ND LEVEL DO LOOPS/SI	.253	.306	.263
X137	INSTR., 6TH LEVEL OR LOWER DO LOOPS/SI	.216	.216	.433
X108	SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR./SI	-.137	-.134	-.061

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	p
5	REGRESSION	1844.559	5	368.912	20.953	<.001
	RESIDUAL	1566.967	30	17.606		
10	REGRESSION	2061.177	10	206.118	12.822	<.001
	RESIDUAL	1350.349	34	16.076		
	TOTAL	3411.526	39			

* BEST SINGLE PREDICTOR

TABLE 6-12. PROJECT P,
ERRORS/PROGRAM = $f(SI + \text{NORMALIZED VARIABLES})$

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>11</u>	<u>18(MAX.)</u>
MULTIPLE R^2	.880	.934	.986
MULTIPLE R^2	.774	.871	.972
STD. ERROR OF ESTIMATE	2.698	2.239	1.276

VARIABLES (X)		COEFFICIENTS		r_{xy}
X1	SOURCE INSTRUCTIONS	.682	.384	.593 ¹¹
X59	COMMENT STATEMENTS/SI	.261	.259	.235
X62	UNCONDITIONAL JUMPS/SI	.479	.612	.543
X63	CALLS/LINKS/SI	---	.198	-.152
X69	CONDITIONAL JUMPS/SI	.196	---	.224
X71	SCALING/ROUNDING OPNS./SI	---	-.261	-.406
X77	ADDRESS VAR. FREQ./SI	---	-.135	-.122
X82	FIXED PT. VAR. FREQ./SI	---	.369	-.158
X96	NON-NESTED DO LOOPS/SI	---	-.312	-.393
X97	DO LOOPS NESTED AT 2ND LEVEL/SI	---	.167	.147
X102	INSTR. IN NON-NESTED DO LOOPS/SI	-.216	---	-.155
X107	INSTR. IN 6TH LEVEL OR LOWER DO LOOPS/SI	---	.242	.224

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	P
5	REGRESSION	724.486	5	144.897	19.914	<.001
	RESIDUAL	211.111	29	7.280		
11	REGRESSION	815.275	11	81.528	16.262	<.001
	RESIDUAL	120.321	24	5.013		
	TOTAL	935.596	34			

¹¹ BEST SINGLE PREDICTOR

TABLE 6-13. FIVE PREDICTOR SUMMARY,
ERRORS/PROGRAM = FCSI + NORMALIZED VARIABLES)

VARIABLES	M ^a	N ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X1 SOURCE INSTRUCTIONS	.597**	.559**	.587**	.646**	.682**
X69 CONDITIONAL JUMPS/SI	.148	.206			.196
X58 USING INSTRUCTIONS/SI	.150	.136			
X74 FUNCTIONS/SI	.195	.169			
X74 LOCK MACROS/SI	.108	.094			
X103 INSTR., 2ND LEVEL DO LOOPS/SI			.193	.258	
<u>CORRELATION STATISTICS</u>					
r ₁ **, ERRORS	.563	.533	.649	.658	.593
r ₂ **, ERRORS	.317	.284	.421	.433	.352
<u>PREDICTION SUMMARY</u>					
R ₂	.622	.598	.717	.735	.880
R	.387	.357	.513	.541	.774

^a ALL OBSERVATIONS USED

^b ZERO ERRORS DELETED

**BEST SINGLE PREDICTOR

TABLE 6-14. TEN PREDICTOR SUMMARY,
ERRORS/PROGRAM = $\Sigma(SI + \text{NORMALIZED VARIABLES})$

VARIABLES	M ^a	M ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X1 SOURCE INSTRUCTIONS	.564 ¹¹	.549 ¹¹	.631 ¹¹	.608 ¹¹	.334 ¹¹
X57 EXIT POINTS/SI	-.098	-.083	.169		
X58 USING INSTRUCTIONS/SI	.181	.149		.208	
X71 SCALING/ROUNDING OPNS./SI	.070		.256		-.261
X82 FIXED PT. VAR. FREQ./SI	-.063			.122	.369
X108 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR./SI	.067	.089		-.134	
X59 COMMENT STATEMENTS/SI					.259
X62 UNCONDITIONAL JUMPS/SI			.111	-.264	.612
X69 CONDITIONAL JUMPS/SI	.120	.160			
X73 FUNCTIONS/SI	.155	.147			
X74 LOCK MACROS/SI	.091	.076			
X91 UNDEFINED VARIABLES/SI			.269	.248	
X103 INSTR., 2ND LEVEL DO LOOPS/SI			.243	.316	
X107 INSTR., 6TH LEVEL DO LOOPS/SI				.215	.224
<u>CORRELATION STATISTICS</u>					
$r^2_{H, \text{ERRORS}}$	.563	.533	.649	.658	.593
$r^2_{H, \text{ERRORS}}$	.317	.284	.421	.433	.352
<u>PREDICTION SUMMARY</u>					
R ₂	.641	.617	.754	.777	.934
R ₂	.411	.381	.569	.604	.871

^aALL OBSERVATIONS USED

^bZERO ERROR RATES DELETED

¹¹BEST SINGLE PREDICTOR

predictors are presented in Tables 6-13 and 6-14, respectively. For projects M, B, and P a total of 45, 43, and 45 predictor variables, respectively, were available to be automatically entered in the prediction equation (see Table C-1, Appendix C).

The predictability of errors is improved only moderately when ten versus five predictors are used in the equation. However, except for project P ($R^2=.87$), the predictability of errors is generally low to moderate, ranging from $R^2=.38$ for project M to $R^2=.60$ for project B, using a combination of X1 plus selected normalized predictors. Judging from the summary data provided in Tables 6-13 and 6-14, there appears to be little consistency of predictors appearing in the equations for each project. For five predictors, only one predictor, Source Instructions, appeared in each of the five prediction equations that were developed across all three projects, with variable X89, Conditional Jumps/SI, appearing only for projects M and P. For ten predictors, only three variables, X1, Source Instructions, X71, Scaling/Rounding Operations/SI, and X83, Fixed Point Variable Frequency/SI, appeared in one or more of the equations developed for each project.

Source Instructions is the best single predictor in each of the equations, and when considered by itself, as also indicated by the results of the previous section, it can be used to account for a large percentage of the variation in errors that is accounted for by the combined set of 10 predictors selected for each project's equation(s).

For equations that were developed when zero errors were deleted (Tables 6-9 and 6-11), it is found for ten predictors that only minor changes resulted; R^2 decreased .03 for project M

and increased .04 for project B, with respect to the predictability of errors using the normalized variables. However, as observed in the results of Section 6.3.1, the variables in the prediction equations do change when these zero error program-observations are deleted. For project B only four of the ten predictors using all observations were reselected when nine zero error observations were deleted, whereas for project M seven of the ten predictors remained in the equation when 90 observations were deleted from the analysis.

The low predictability obtained when using the normalized variables seems to result for two reasons. First, as reported in Tables 6-8 through 6-12, the correlation of each normalized variable with errors was generally very low. This indicates that the variable errors per program has very little in common with these normalized variables other than Source Instructions. A second reason for this low predictability results from the fact that a majority of these normalized variables were uncorrelated with each other. Thus, each variable is contributing independently to the prediction of errors, with only a very small chance that a suppressor variable; i.e., a variable with zero or low correlation with errors and correlated highly with a predictor that correlates moderately to highly with errors, could be present that would improve the predictability of these variables.

The results for project P are of interest here, not only because of the high predictability obtained ($R^2=.871$), but also because this R^2 value was only .024 less than the value of R^2 (.895) obtained when errors were predicted as a function of the unnormalized variables. For each of the other two projects, the reductions in R^2 values were much larger (.18 for project M and .16 for project B).

Comparing the predictors selected for project P in Table 6-5 (i.e., when errors were predicted as a function of the unnormalized variables) versus the predictors selected in Table 6-12 (when errors were predicted as a function of Source Instructions plus the normalized variables), one observes that seven of the ten unnormalized predictions of Table 6-5 were reselected in their normalized form for the equation in Table 6-12. The results of each table are repeated in Table 6-15 for ease of comparison. (For project M with 395 observations, only four of the ten original variables were reselected in their normalized form, and for project B with 104 observations, five of the ten were reselected).

In Table 6-15 each of the correlations of unnormalized variables with errors changed in a negative direction when the respective variable was normalized. For some variables (X5, X8, X23, X28, and X53) this meant that their correlations became smaller in magnitude, whereas for others (X17 and X42) their correlation with errors increased in magnitude.

For project P this high predictability of errors ($R^2=.87$) is consistent with the high predictability ($R^2=.89$) obtained when errors are predicted using the unnormalized variables (see Section 6.3.1), and indicates that predictability is relatively unaffected when the original program complexity variables, with the effect of Source Instructions removed, are combined with X1 to predict errors. One possible explanation for this consistency of prediction using different sets of predictors could be that project P implemented several new programming techniques during its development, whereas projects M and B did not to any great extent. Another explanation could be that only a small, select group of particularly skilled programmers were needed to develop the 35 programs of project P;

TABLE 6-15. PROJECT P, PREDICTION EQUATION COMPARISON

RESULTS	VARIABLES	COEFFICIENT	r_{xy}
Table 6-5, $R^2=.895$	X4 Using Instructions	-.465	.352
	X5 Comment Statements	.762	.509
	X8 Unconditional Jumps	.544	.664
	X17 Scaling/Rounding Opns.	-.176	-.024
	X23 Address Variable Freq.	-.386	.168
	X28 Fixed Point Var. Freq.	.600	.242
	X37 Undefined Variables	.436	.567
	X42 Non-Nested Do Loops	.386	.179
	X51 Instr., 4th Level Do Loops	-.374	.424
	X53 Instr., 6th Level or Lower Do Loops	.602	.341
Table 6-12, $R^2=.871$	X1 Source Instructions	.384	.593 Δr_{xy}
	X59 Comment Statement/SI	.259	.203 -.304
	X62 Unconditional Jumps/SI	.612	.543 -.121
	X63 Calls/Links/SI	.198	-.052
	X71 Scaling/Rounding Opns./SI	-.261	-.406 -.382
	X77 Address Var. Freq./SI	-.135	-.022 -.190
	X82 Fixed Pt. Var. Freq. SI	.369	-.058 -.300
	X96 Non-Nested Do Loops/SI	-.302	-.393 -.572
	X97 Do Loops Nested At 2nd Lvl. SI	.162	.047
	X107 Instr., 6th Level or Lower Do Loops SI	.242	.234 -.117

whereas a much larger group of programmers with varying levels of skill and experience were needed to develop the 499 programs of project M and B, since both projects overlapped and were concurrent with each other at different stages in their software development.

For any of the explanations presented here, it should be kept in mind that each of these projects was functionally different from the others, and each was programmed in a special purpose programming language. Any of these factors in addition to others (e.g. small sample size of project P, definition of variables, testing considerations for each project) taken singly or in combination, could be largely responsible for these obtained results.

In summary, with the exception of project P, the normalized variables contributed appreciably less to the prediction of errors than did the unnormalized variables whose results were discussed in Section 6.3.1. This low predictability suggests the need to identify any non-linear relationships that exist among the predictors and errors, and then to define accordingly the most appropriate non-linear model which could be used to improve these predictors. Given these results, no prediction of errors/program was attempted using selected sets of normalized and unnormalized predictors in combination.

6.3.3 Error Rate/Program = f(Unnormalized Variables)

As discussed earlier in Section 5.0, the newly constructed dependent variable, error rate/program, is analyzed in this study both as a function of the unnormalized variables and as a function of Source Instruction plus the unnormalized variables.

The prediction results using only the unnormalized variables are presented in Tables 6-16 thru 6-20. Tables 6-21 and 6-22 present summaries of these results for five and eight predictors, respectively. The summary for eight predictors is presented based on the lowest maximum number of variables to be entered in the equation over all projects. The lowest maximum value was eight for project P for this set of predictors. Here, as in Section 6.3.1, the number of unnormalized predictor variables that were made available for automatic selection for entry in the regression equation were 45 each for projects M and P and 43 for project B (These variables are listed in Table C-1 of Appendix C).

The summary results clearly indicate that a very low level of predictability (R^2 values ranged from .10 to .30) is obtained for error rate using the unnormalized variables. This prediction is consistently low over each of the regression equations obtained for each project both for five and eight predictors, with the exception of the eight predictor equation for project P which yields a moderate prediction of $R^2=.535$. Also, little consistency among predictors is apparent across all three projects. When eight predictors were used only two predictors, X4, Using Instructions and X5, Comment Statements, appeared in the results for all projects. For five predictors, Using Instructions was the only variable consistently selected in the equation for each project.

One interesting result observed for this set of predictions was that when zero error rate programs in project M and B were deleted from the analysis (see Tables 6-17 and 6-19), the predictions improved; R^2 for project M increased .06, and R^2 for project B increased .12. In the two previous sections, the

TABLE 6-16. PROJECT M,
ERROR RATE/PROGRAM = $f(\text{UNNORMALIZED VARIABLES})$

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>8</u>	<u>9(MAX.)</u>
MULTIPLE R^2	.291	.314	.318
MULTIPLE R^2	.085	.099	.101
STD. ERROR OF ESTIMATE	2.890	2.879	2.879

VARIABLES (X)	COEFFICIENTS		r_{xy}
X1 SOURCE INSTRUCTIONS	- .722	- .820	- .131*
X4 USING INSTRUCTIONS	.251	.243	.041
X5 COMMENT STATEMENTS	---	.090	-.028
X15 CONDITIONAL JUMPS	.370	.277	-.090
X16 FUNCTIONS	.220	.219	.033
X35 REGISTER VARIABLES	- .133	- .105	-.048
X37 UNDEFINED VARIABLES	---	.148	-.047
X45 DO LOOPS NESTED AT 4TH LEVEL	---	.074	-.050

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	p
5	REGRESSION	301.531	5	60.306	7.221	<.001
	RESIDUAL	3248.768	389	8.352		
8	REGRESSION	350.229	8	43.779	5.281	<.001
	RESIDUAL	3200.070	386	8.290		
	TOTAL	3550.299	394			

* BEST SINGLE PREDICTOR

TABLE 6-17. PROJECT M,
ERROR RATE/PROGRAM = f (UNNORMALIZED VARIABLES),
ZERO ERROR RATES DELETED

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>8</u>	<u>11(MAX.)</u>
MULTIPLE R_1	.371	.394	.410
MULTIPLE R^2	.138	.155	.168
STD. ERROR OF ESTIMATE	2.877	2.862	2.855

VARIABLES (X)	COEFFICIENTS		r_{xy}
X1 SOURCE INSTRUCTIONS	-1.080	-1.375	-.292*
X11 USER MACROS	.099	.096	-.123
X15 CONDITIONAL JUMPS	.421	.356	-.211
X16 FUNCTIONS	.202	.174	-.106
X27 FIXED PT. VARIABLES	---	-.099	-.201
X38 UNDEFINED VAR. FREQ.	.250	.210	-.220
X45 DO LOOPS NESTED AT 4TH LEVEL	---	-.083	-.100
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.	---	.469	-.262

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	F
5	REGRESSION	396.015	5	79.203	9.571	<.001
	RESIDUAL	2474.430	299	8.276		
8	REGRESSION	446.074	8	55.759	6.808	<.001
	RESIDUAL	2424.371	296	8.190		
	TOTAL	2870.445	304			

* BEST SINGLE PREDICTOR

TABLE 6-18. PROJECT B,
ERROR RATE/PROGRAM = $\bar{e}$ (UNNORMALIZED VARIABLES)

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>8</u>	<u>9(MAX.)</u>
MULTIPLE R^2	.362	.418	.430
MULTIPLE R^2	.131	.175	.185
STD. ERROR OF ESTIMATE	2.380	2.356	2.354

VARIABLES (X)	COEFFICIENTS		r_{xy}
X5 COMMENT STATEMENTS	---	.162	- .105
X15 CONDITIONAL JUMPS	- .481	- .568	- .234*
X17 SCALING/ROUNDING OPNS.	.143	.212	- .024
X37 UNDEFINED VARIABLES	.358	.371	- .121
X49 INSTR., 2ND LEVEL DO LOOPS	.223	.307	- .037
X50 INSTR., 3RD LEVEL DO LOOPS	---	- .236	- .179
X52 INSTR., 5TH LEVEL DO LOOPS	---	.173	.056
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.	- .339	- .424	- .217

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	p
5	REGRESSION	83.803	5	16.761	2.959	<.05
	RESIDUAL	555.173	98	5.665		
8	REGRESSION	111.506	8	13.938	2.510	<.05
	RESIDUAL	527.470	95	5.552		
	TOTAL	638.976	103			

* BEST SINGLE PREDICTOR

TABLE 6-19. PROJECT B,
ERROR RATE/PROGRAM = f(UNNORMALIZED VARIABLES),
ZERO ERROR RATES DELETED

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>8</u>	<u>15(MAX.)</u>
MULTIPLE R^2	.484	.543	.629
MULTIPLE R^2	.234	.295	.393
STD. ERROR OF ESTIMATE	2.144	2.093	2.022

VARIABLES (X)	COEFFICIENTS		r_{xy}
X4 USING INSTRUCTIONS	.371	.445	-.084
X10 SYSTEM MACROS	---	-.292	-.212
X11 USER MACROS	.328	.438	-.166
X28 FIXED PT. VAR. FREQ.	.277	---	-.052
X43 DO LOOPS, NESTED AT 2ND LEVEL	---	.383	-.121
X47 DO LOOPS, 6TH LEVEL OR LOWER	---	.428	-.070
X49 INSTR., 2ND LEVEL DO LOOPS	.316	.755	-.082
X50 INSTR., 3RD LEVEL DO LOOPS	---	-.342	-.211
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.	-1.162	-1.022	-.310*

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	F'
5	REGRESSION	125.202	5	25.040	5.447	<.001
	RESIDUAL	409.144	89	4.597		
8	REGRESSION	157.667	8	19.708	4.500	<.001
	RESIDUAL	376.680	86	4.380		
	TOTAL	534.346	94			

* BEST SINGLE PREDICTOR

TABLE 6-20. PROJECT P,
ERROR RATE/PROGRAM = F(UNNORMALIZED VARIABLES)

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>8(MAX.)</u>
MULTIPLE R ₂	.643	.731
MULTIPLE R ²	.413	.535
STD. ERROR OF ESTIMATE	1.048	.985

VARIABLES (X)	COEFFICIENTS		r _{xy}
X2 ENTRY POINTS	- .499*	---	- .322
X3 EXIT POINTS	---	- .438*	- .309
X4 USING INSTRUCTIONS	- .319	-1.172	- .208
X5 COMMENT STATEMENTS	.636	1.105	.132
X18 SHORT DO LOOPS	---	- .369	- .136
X20 LOCK MACROS	- .400	- .276	- .086
X28 FIXED PT. VAR. FREQ.	---	.361	- .130
X35 REGISTER VARIABLES	- .276	---	- .245
X51 INSTR., 4TH LEVEL DO LOOPS	---	- .434	- .220
X53 INSTR., 6TH LEVEL OR LOWER DO LOOPS	---	- .881	- .129

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	F
5	REGRESSION	22.427	5	4.485	4.087	<.01
	RESIDUAL	31.824	29	1.097		
8	REGRESSION	29.002	8	3.625	3.733	<.01
	RESIDUAL	25.249	26	.971		
	TOTAL	54.251	34			

* BEST SINGLE PREDICTOR

TABLE 6-21. FIVE PREDICTOR SUMMARY,
ERROR RATE/PROGRAM = $\bar{E}$ (UNNORMALIZED VARIABLES)

VARIABLES	M^a	M^b	B^a	B^b	P
<u>REGRESSION COEFFICIENTS</u>					
X4 USING INSTRUCTIONS	.251			.371	- .319
X15 CONDITIONAL JUMPS	.371	.421	- .481 [#]		
X1 SOURCE INSTRUCTIONS	- .722 [#]	-1.888 [#]			
X11 USER MACROS		.099		.328	
X16 FUNCTIONS	.220	.202			
X35 REGISTER VARIABLES	- .133				- .276
X49 INSTR. 2ND LEVEL DO LOOPS			.223	.316	
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.			- .339	-1.162 [#]	
X2 ENTRY POINTS					- .499 [#]
<u>CORRELATION STATISTICS</u>					
r_{SI}	1.000	1.000	.902	.979	.263
$r_{SI, \text{ERROR RATE}}$	- .131	- .292	- .195	- .276	- .301
$r_{\bar{E}, \text{ERROR RATE}}$	- .131	- .292	- .234	- .310	- .322
$r_{\bar{E}^2, \text{ERROR RATE}}$	.017	.085	.055	.096	.104
<u>PREDICTION SUMMARY</u>					
R	.291	.371	.362	.484	.643
R ²	.085	.138	.131	.234	.413

^aALL OBSERVATIONS USED

^bZERO ERROR RATES DELETED

[#]BEST SINGLE PREDICTOR

TABLE 6-22. EIGHT PREDICTOR SUMMARY,
ERROR RATE/PROGRAM = 2 (UNNORMALIZED VARIABLES)

VARIABLES	M ^a	M ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X4 USING INSTRUCTIONS	.243			.445	-1.172
X5 COMMENT STATEMENTS	.090		.162		1.105
X15 CONDITIONAL JUMPS	.277	.356	-.568 ¹¹		
X54 SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR.		.469	-.424	-1.122 ¹¹	
X1 SOURCE INSTRUCTIONS	-.820 ¹¹	-1.375 ¹¹			
X11 USER MACROS		.096		.438	
X16 FUNCTIONS	.219	.174			
X37 UNDEFINED VARIABLES	.148		.371		
X45 DO LOOPS, 4TH LEVEL	.074	-.083			
X49 INSTR., 2ND LEVEL DO LOOPS			.307	.755	
X50 INSTR., 3RD LEVEL DO LOOPS			-.236	-.342	
X3 EXIT POINTS					-.438 ¹¹
<u>CORRELATION STATISTICS</u>					
r ₁₁ ,SI	1.000	1.000	.902	.979	.279
r ₁₁ ,ERROR RATE	-.131	-.292	-.105	-.276	-.301
r ₁₂ ,ERROR RATE	-.131	-.292	-.234	-.310	-.309
r ₁₂ ,ERROR RATE	.017	.085	.055	.096	.095
<u>PREDICTION SUMMARY</u>					
R	.314	.394	.418	.543	.731
R ²	.099	.155	.175	.290	.535

^aALL OBSERVATIONS USED
^bZERO ERROR RATES DELETED
¹¹BEST SINGLE PREDICTOR

deletion of these same programs always increased the value of R^2 for project B and decreased or had little effect on the R^2 values for project M, when errors/program was being predicted. In these results, however, when predicting error rate the R^2 value for project M increases approximately 57% (from .099 to .155), and the R^2 value for project B increases approximately 68% (from .175 to .295). Both of these changes are statistically significant. In fact, comparatively speaking, when predicting error rate with the zero observations removed from the analysis, one observes that the resultant increase in R^2 values could possibly be equated with the increase that would result from three to four additional variables being added to the prediction equation.

The increases observed here in the predictability of error rate support the hypothesis stated in Section 3.0 which suggested that programs with zero reported errors actually had latent errors still in the program that had gone undetected due to a lack of thoroughness in testing. As such, these zero reported errors were considered as underestimates of the total errors in the program, and their presence in the analysis would therefore reduce the predictability of the dependent variable and increase the predictability when they were removed. More will be said about the error rate measure and this hypothesis later in the report.

As noticed in the previous section, when zero error rate programs are deleted, different prediction equations result. For project M, deleting the 90 zero observations results in only three of the eight predictors being reselected; for project B, only three of the eight predictors were reselected.

For the best single predictors of error rate among the unnormalized variables that were in the regression equation, no consistency at all is found across each project (see Tables 6-21 and 6-22). It is interesting to note at this time that several of the equations that were developed for this set of predictions do not contain the best single predictor of error rate among the total set of unnormalized variables. For example, for project M using all 395 observations (see Table 6-16), the best single predictor is X7, Arithmetic Instructions (r_{x7} , error rate = $-.148$) and not X1, Source Instructions (r_{x1} , error rate = $-.131$). X1 is, however, the best predictor, as reported in Table 6-18, when the 90 zero error rates are deleted from the analysis for project M. For the results of project B presented in Table 6-19 (95 observations used), the best predictor is X15 Conditional Jumps (r_{x15} , error rate = $-.331$). Finally, for the results of project P using five and eight predictors (see Table 6-20), variable X2, Entry Points (r_{x2} , error rate = $-.322$) is the best predictor of error rate.

The fact that some of these best single predictor variables were not present in the various prediction equations that were generated accentuates the need to proceed with caution when any attempt is made to decipher the relative contributions of each variable to the prediction from a multiple regression analysis. Clearly it is the combination of variables which should be considered and not any one variable separate from the others.

In summary, for this set of predictions, with the exception of the moderate prediction of error rate obtained for project P, it is found that linear combinations of unnormalized variables produce relatively low level predictions for error rate. This

appears to be due to the low correlations that each of the selected predictors had with error rate (see Tables 6-16 to 6-19). These low correlations stand in stark contrast to the moderate to high correlations that these same unnormalized variables had with errors. This shows that when the effect of Source Instructions is removed from the error variable, the normalized error variable (error rate) has little left in common with any of the unnormalized predictors.

Thus it seems clear that these results reiterate and demonstrate the fact that a large portion of the relationship between the unnormalized variables and errors/program result from the combined influence of Source Instructions in each. This is what was found for the results as presented in Section 6.3.1, errors as a function of the unnormalized variables. It appears warranted, then, that any future predictions of error rate from combinations of the unnormalized variables should investigate non-linear models, or use variables other than those investigated in this study.

6.3.4 Error Rate/Program = f(SI + Normalized Variables)

The final set of prediction equation results are presented in Tables 6-23 thru 6-27 for five and ten predictors, wherein error rate is being predicted from a linear combination of the normalized variables and Source Instructions. Summary Tables 6-28 and 6-29, for five and ten predictors, are also provided. The number of predictors that were made available for selection by the regression procedure were 45 each for projects M and P and 43 for project B (see Table C-1).

In viewing the summary results one can observe that the use of ten versus five predictors substantially improves the predictions for each project. The percentage increase in values of R^2 for each project ranged from an increase of 16% for project P to an increase of 23% and 38%, respectively, for projects M and B. Additionally, the predictability of error rate using this set of predictors (R^2 values ranged from .34 to .47 for 10 predictors) is low, but generally higher than the previous results obtained for error rate (i.e., in Section 6.3.3). Again project P is an exception. For project P the percent of variance accounted for by five predictors is $R^2 = .81$; for ten predictors $R^2 = .94$. These results represent the highest values of R^2 obtained for project P over all the prediction equations that were developed for this project.

Using ten predictors, variable X58, Using Instructions/SI, is the only predictor that appears consistently in each of the prediction equations for all projects. For two of the three projects, M and B, this variable is also the best single predictor of error rate. For these two projects Using Instructions/SI accounts for 47% (for project B) and 54% (for

TABLE 6-23. PROJECT M,
ERROR RATE/PROGRAM = $f(\text{SI} + \text{NORMALIZED VARIABLES})$

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>16 (MAX.)</u>
MULTIPLE R^2	.545	.603	.625
MULTIPLE R^2	.297	.364	.390
STD. ERROR OF ESTIMATE	2.532	2.426	2.393

VARIABLES (X)	COEFFICIENTS		r_{xy}
X56 ENTRY POINTS/SI	---	.244	.229
X57 EXIT POINTS/SI	---	-.366	.123
X58 USING INSTRUCTIONS/SI	.350	.281	.397**
X59 COMMENT STATEMENTS/SI	---	.179	.341
X64 SYSTEM MACROS/SI	---	.173	.279
X70 FUNCTIONS/SI	.257	.246	.245
X81 FIXED POINT VARIABLES/SI	---	-.114	-.024
X87 LABELED ARRAY VARIABLES/SI	-.129	-.106	-.116
X90 REGISTER VARIABLES FREQ./SI	-.165	-.118	-.165
X91 UNDEFINED VARIABLES/SI	.167	.116	.320

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	F
5	REGRESSION	1055.927	5	211.185	32.935	<.001
	RESIDUAL	2494.371	389	6.412		
10	REGRESSION	1290.469	10	129.047	21.928	<.001
	RESIDUAL	2259.830	384	5.885		
	TOTAL	3550.299	394			

** BEST SINGLE PREDICTOR

TABLE 6-24. PROJECT M,
 ERROR RATE/PROGRAM = $f(\text{SI} + \text{NORMALIZED VARIABLES})$,
 ZERO ERROR RATES DELETED

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>22(MAX.)</u>
MULTIPLE R^2	.610	.644	.679
MULTIPLE R^2	.373	.415	.462
STD. ERROR OF ESTIMATE	2.454	2.390	2.341

VARIABLES (X)	COEFFICIENTS		r_{xy}
X58 USING INSTRUCTIONS/SI	.319	.278	.471*
X60 LABELED INSTRUCTIONS/SI	.105	.221	.227
X64 SYSTEM MACROS/SI	.202	.190	.374
X66 EQUATE STATEMENTS/SI	---	-.092	-.051
X68 LOGICAL CONNECTORS/SI	---	.137	.107
X70 FUNCTIONS/SI	.192	.178	.174
X73 NESTED SHORT DO LOOPS/SI	---	-.096	-.030
X81 FIXED POINT VARIABLES/SI	---	-.082	-.020
X84 FLOATING PT. VARIABLE FREQ./SI	---	.104	-.103
X91 UNDEFINED VARIABLES/SI	.214	.277	.415

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	p
5	REGRESSION	1069.392	5	213.878	35.507	<.001
	RESIDUAL	1801.053	299	6.024		
10	REGRESSION	1190.689	10	119.069	20.840	<.001
	RESIDUAL	1679.755	294	5.713		
	TOTAL	2870.445	304			

* BEST SINGLE PREDICTOR

TABLE 6-25. PROJECT B,
ERROR RATE/PROGRAM = f(SI + NORMALIZED VARIABLES)

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>13(MAX.)</u>
MULTIPLE R ₂	.494	.579	.605
MULTIPLE R ²	.244	.336	.365
STD. ERROR OF ESTIMATE	2.221	2.137	2.123

VARIABLES (X)	COEFFICIENTS		r _{xy}
X57 EXIT POINTS/SI	.141	.362	.219
X58 USING INSTRUCTIONS/SI	.304	.363	.312 ^{**}
X61 ARITHMETIC INSTRUCTIONS/SI	---	-.139	-.069
X64 SYSTEM MACROS/SI	---	-.391	.041
X68 LOGICAL CONNECTORS/SI	---	.158	.038
X91 UNDEFINED VARIABLES/SI	---	.158	.140
X97 DO LOOPS NESTED AT 2ND LEVEL/SI	---	-.270	.114
X99 DO LOOPS NESTED AT 4TH LEVEL/SI	-.653	-.680	-.211
X103 INSTR., 2ND LEVEL DO LOOPS/SI	.241	.399	.191
X105 INSTR., 4TH LEVEL DO LOOPS/SI	.508	.525	-.130

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	P
5	REGRESSION	155.766	5	31.153	6.318	<.001
	RESIDUAL	483.210	98	4.931		
10	REGRESSION	214.411	10	21.441	4.697	<.001
	RESIDUAL	424.565	93	4.565		
	TOTAL	638.978	103			

** BEST SINGLE PREDICTOR

TABLE 6-26. PROJECT B,
ERROR RATE/PROGRAM = $f(\text{SI} + \text{NORMALIZED VARIABLES})$,
ZERO ERROR RATES DELETED

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>23(MAX.)</u>
MULTIPLE R^2	.614	.683	.775
MULTIPLE R^2	.376	.467	.600
STD. ERROR OF ESTIMATE	1.935	1.842	1.734

VARIABLES (X)		COEFFICIENTS		r_{xy}
X57	EXIT POINTS/SI	---	.199	.284
X58	USING INSTRUCTIONS/SI	.386	.480	.467*
X60	LABELED INSTRUCTIONS/SI	---	-.237	.037
X62	UNCONDITIONAL JUMPS/SI	---	.228	-.090
X66	EQUATE STATEMENTS/SI	-.232	-.160	-.232
X70	FUNCTIONS/SI	---	-.205	-.090
X71	SCALING/ROUNDING OPNS./SI	---	.282	.035
X91	UNDEFINED VARIABLES/SI	.241	.360	.193
X103	INSTR., 2ND LEVEL DO LOOPS/SI	.259	.223	.216
X108	SI X AVG. NO. OPERATORS/ ARITHMETIC INSTR./SI	-.182	-.148	-.143

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	F
5	REGRESSION	201.097	5	40.219	10.741	<.001
	RESIDUAL	333.249	89	3.744		
10	REGRESSION	249.335	10	24.933	7.349	<.001
	RESIDUAL	285.012	84	3.393		
	TOTAL	534.346	94			

* BEST SINGLE PREDICTOR

TABLE 6-27. PROJECT P.
ERROR RATE/PROGRAM = $f(\text{SI} + \text{NORMALIZED VARIABLES})$

VARIABLES IN PREDICTION EQUATION

	<u>5</u>	<u>10</u>	<u>20(MAX.)</u>
MULTIPLE R^2	.901	.969	.994
MULTIPLE R^2	.812	.939	.989
STD. ERROR OF ESTIMATE	.593	.373	.209

VARIABLES (X)		COEFFICIENTS		r_{xy}
X58	USING INSTRUCTIONS/SI	---	.316	.176
X59	COMMENT STATEMENTS/SI	.606	.467	.813**
X61	ARITHMETIC INSTRUCTIONS/SI	---	.530	.009
X62	UNCONDITIONAL JUMPS/SI	.360	.336	.421
X69	CONDITIONAL JUMPS/SI	---	.143	.209
X74	LOCK MACROS/SI	-.260	---	.020
X84	FLOATING PT. VAR. FREQ./SI	---	.300	.181
X97	DO LOOPS NESTED AT 2ND LEVEL/SI	.309	.555	.552
X98	DO LOOPS NESTED AT 3RD LEVEL/SI	---	.285	.169
X102	INSTR. IN NON-NESTED DO LOOPS/SI	-.265	.535	.031
X105	INSTR. IN 4TH LEVEL DO LOOPS/SI	---	.255	.121

ANALYSIS OF VARIANCE

NO. PREDICTORS		SUM OF SQUARES	DF	MEAN SQUARE	F	p
5	REGRESSION	44.038	5	8.808	25.008	<.001
	RESIDUAL	10.214	29	.352		
10	REGRESSION	50.918	10	5.092	36.667	<.001
	RESIDUAL	3.333	24	.139		
	TOTAL	54.251	34			

** BEST SINGLE PREDICTOR

TABLE 6-28. FIVE PREDICTOR SUMMARY,
ERROR RATE/PROGRAM = $f(SI + \text{NORMALIZED VARIABLES})$

VARIABLES	M ^a	M ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X58 USING INSTRUCTIONS/SI	.356**	.319**	.334**	.386**	
X91 UNDEFINED VARIABLES/SI	.167	.214		.241	
X73 FUNCTIONS/SI	.257	.192			
X103 INSTR., 2ND LEVEL DO LOOPS/SI			.241	.259	
X59 COMMENT STATEMENTS/SI					.606**
<u>CORRELATION STATISTICS</u>					
r _{SI} , ERROR RATE	-.131	-.292	-.195	-.276	-.301
r ₂ , ERROR RATE	.397	.471	.312	.467	.813
r ₂ *, ERROR RATE	.158	.222	.097	.218	.660
<u>PREDICTION SUMMARY</u>					
R ₂	.545	.610	.494	.614	.901
R	.297	.373	.244	.376	.812

^aALL OBSERVATIONS USED

^bZERO ERROR RATES DELETED

**BEST SINGLE PREDICTOR

TABLE 6-29. TEN PREDICTOR SUMMARY,
ERROR RATE/PROGRAM = $f(SI + \text{NORMALIZED VARIABLES})$

VARIABLES	M ^a	M ^b	B ^a	B ^b	P
<u>REGRESSION COEFFICIENTS</u>					
X58 USING INSTRUCTIONS/SI	.281 ¹¹	.278 ¹¹	.363 ¹¹	.488 ¹¹	.316
X91 UNDEFINED VARIABLES/SI	.116	.277	.158	.368	
X57 EXIT POINTS/SI	-.366		.362	.199	
X64 SYSTEM MACROS/SI	.173	.199	-.391		
X77 FUNCTIONS/SI	.246	.178		-.285	
X59 COMMENT STATEMENTS/SI	.179				.467 ¹¹
X60 LABELED INSTRUCTIONS/SI		.221		-.237	
X61 ARITHMETIC INSTRUCTIONS/SI			-.139		.538
X62 UNCONDITIONAL JUMPS/SI				.228	.336
X66 EQUATE STATEMENTS/SI		-.092		-.168	
X68 LOGICAL CONNECTORS/SI		.137	.158		
X81 FIXED PT. VARIABLES/SI	-.114	-.082			
X84 FLOATING PT. VAR. FREQ./SI		.114			-.388
X97 DO LOOPS, 2ND LEVEL/SI			-.271		.555
X113 INSTR., 2ND LEVEL DO LOOPS/SI			.399	.223	
<u>CORRELATION STATISTICS</u>					
$r_{SI, \text{ERROR RATE}}$	-.131	-.292	-.195	-.276	-.381
$r_{M, \text{ERROR RATE}}$	.397	.471	.312	.467	.813
$r_{B, \text{ERROR RATE}}$	.158	.222	.097	.218	.668
<u>PREDICTION SUMMARY</u>					
R ₂	.613	.644	.579	.683	.969
R ²	.364	.415	.336	.467	.939

^aALL OBSERVATIONS USED

^bZERO ERROR RATES DELETED

¹¹BEST SINGLE PREDICTOR

project M) of the total variance accounted for by all 10 predictors selected for each project. In addition, for all projects, the normalized program complexity variables are also contributing significantly to the predictions for error rate in these results.

Although the correlation of Using Instructions/SI with error rate is low (from .31 to .47) for those projects where it is the best predictor, the fact that this variable does contribute substantially to error rate predictions is important to consider. Using Instructions as described in Table B-1 are instructions used to establish data structure interfaces in the program. An interpretation then for the normalized variable, Using Instructions/SI, is the number of instructions per 100 lines of code used to establish data structure interfaces; or more simply interpreted, data interfaces per 100 lines of code. The fact that interfaces have been found to contribute to errors are results which support the findings of other researchers (Thayer et al. 1976; and Okimoto, 1975) (13, 14), as cited earlier in Section 6.2.

Once again as found in the immediately preceding results for error rate as a linear function of the unnormalized variables, one can observe that the predictions for error rate are consistently improved for projects M and B when the zero program-observations are deleted from the analysis. This result suggests that analysis of error rate supports the hypothesis of lack of thoroughness of testing in some of these programs.

An interesting finding in these results is that variable X59, Comment Statements/SI, is the best single predictor for error rate for project P. The fact that X59 alone accounts

for approximately 70% of the total variance accounted for by all ten predictors selected for this project is surprising. This percentage was even higher, 81%, when five variables are considered.

The fact that the results for project P have been so consistently different from the results of projects M and B suggests that the function being programmed for project P, the software development and testing environment, the programmers and management of project P, and the CENTRAN language itself are all contributing in some distinct or interactive way to produce these unique results. Possibly the fact that project P implemented the new programming techniques whereas the other projects did not explains the high predictability of error rate in these results. Although it could be suggested that structured programs would require only few comment statements due to the inherent readability that structuring a program is supposed to provide, the opposite could have been true for these programs. That is, since over half (66%) of the programs were structured and more readable as a result, more comments may have been incorporated as a direct result of being able to more readily understand the flow of logic and read the programs. This may have resulted in errors per 100 lines of code being more easily detected when problems arose in the programs. Nevertheless, regardless of the numerous plausible hypotheses that could explain the results for project P, given the information available, it can only be suggested that a variety of factors, as stated above, may account for these results.

In summary, since the results obtained for both sets of predictions for error rate were relatively low (see Sections 6.3.3 and 6.3.4), no further attempt to combine the normalized and unnormalized variables to predict error rate was made. Data Interfaces per 100 lines of code is a significant contributor to the predictions for error rate. Other program complexity variables also significantly contribute to these predictions. Error rate was found to be a meaningful measure to use for detecting the effects of error free programs on the prediction of programming errors. Finally, the low predictions obtained suggest the need to analyze these variables with non-linear models.

6.3.5 Validation of Prediction Equations for Sample S

The major purpose of validating predictions is to identify and examine how well the same level of predictions can be maintained, carried over, or reproduced for a separate data sample. This separate sample is assumed to be drawn from the same population as the first sample for which the prediction results were obtained. In practice, this validation procedure (normally referred to as cross-validation) is usually carried out by applying the obtained regression coefficients to an identical set of predictor variables collected for the separate data sample. Using these coefficients and the predictor variable values, estimates of the dependent variable can then be directly computed. These estimates, when correlated with the actual values of the dependent variable that have been collected for the separate sample, can then provide information about how well this separate sample validates; i.e., shows consistency of prediction, with the original sample. Validation, then, can be considered as an important means for assessing to what extent prediction results can be generalized to other samples.

Generally, it has been found that when the regression coefficients obtained from a multiple regression analysis on one sample are applied to a second sample, the correlation between the weighted predictors and the dependent variable in the second sample will be less than the multiple correlation value (R) originally obtained from the first sample. This phenomenon is referred to as shrinkage of the multiple correlation coefficient (Kerlinger and Pedhazur, pp. 282-284; Ferguson, pp. 401-402) (4, 8). Basically, the reason for this is that the multiple regression performed on the sample data capitalizes on chance. The highest product moment correlations are selected to enter into the regression equation, and on subsequent samples these correlations would probably be lower, therefore yielding a somewhat lower overall multiple correlation.

The extent of the bias in sample values of R is directly dependent upon the population value of the multiple correlation coefficient, the sample size, and the number of predictor variables used in the equations. For validation purposes, it is possible to estimate the amount of shrinkage that will result when a second sample is to be validated. The computational formula used to provide this estimate is as follows:

$$\hat{R}^2 = \left[1 - (1-R^2) \left(\frac{N-1}{N-k-1} \right) \right] \quad (6.1)$$

where $\hat{R}^2$ = the estimated squared multiple correlation in the population; R^2 = the obtained squared multiple correlation for the first sample used to develop the prediction equation; N = the number of observations in the second (i.e., validation) sample; and k = the number of predictor variables used to obtain the R^2 value.

For sample S (as well as for sample T), no separate sample of programs were initially randomly selected and set aside to use for cross-validation purposes. The interest at that time was to achieve the maximum attainable predictions and the most representative results possible, using all the observations that had been provided for each data sample. Resultantly, for validation of sample S predictions, formula (6.0) was used to estimate the amount of shrinkage of the obtained values of R^2 that would result had samples for each project been set aside or made available for validation.

Since the highest values of R^2 were obtained when errors/program were predicted as a function of the unnormalized variables (see results of Section 6.3.1), only these equations were selected for validation. The validation results are presented in Table 6-30 for both five and ten predictors. The obtained value of R^2 , number of observations used, and table reference for the prediction equation results as cited in Section 6.3.1 are presented at the top of Table 6-30. $\hat{R}^2$ values and shrinkage results are presented for sample sizes of $N = 20, 50, \text{ and } 150$ programs, including a sample size equal to the actual number of observations used for the original predictions.

Clearly these results show that the prediction equations could be expected to validate; i.e., show increasing consistency with the obtained values of R^2 , as the sample size increases and as the ratio of predictors to observations in the sample becomes smaller. For example, for the five predictor equation for project M, with a predictor to sample size ratio of $k/N = 1:4$, $\hat{R}^2$ is about two-thirds the size of R^2 (i.e., .339 and .514 respectively). When the hypothetical validation

TABLE 6-30. SAMPLE 5, VALIDATION RESULTS FOR FIVE AND TEN PREDICTOR
REGRESSION EQUATIONS, ERRORS/PROGRAM = F(UNNORMALIZED VARIABLES)

		M ^a	M ^b	B ^a	B ^b	P
FOR K=5	TABLE REFERENCE NOBS ^c R ²	5-2 395 .514	5-3 305 .483	5-4 104 .660	5-5 95 .695	5-6 35 .788
FOR N=20, K=5, K/N=1:4	A ² R SHRINKAGE	.339 -.175	.297 -.186	.538 -.122	.585 -.110	.712 -.076
FOR N=50, K=5, K/N=1:10	A ² R SHRINKAGE	.461 -.053	.427 -.056	.623 -.032	.661 -.034	.765 -.023
FOR N=150, K=5, K/N=1:30	A ² R SHRINKAGE	.499 -.015	.467 -.016	.650 -.010	.686 -.009	.782 -.006
FOR N=NOBS., K=5	K/N R ² SHRINKAGE	1:79 .508 -.006	1:61 .474 -.009	1:21 .643 -.017	1:19 .678 -.017	1:17 .751 -.037

^a ALL OBSERVATIONS USED

^b ZERO ERRORS DELETED

^c NOBS. = NUMBER OF OBSERVATIONS

TABLE 6-30. SAMPLE 5, VALIDATION RESULTS FOR FIVE AND TEN PREDICTOR
REGRESSION EQUATIONS, ERRORS/PROGRAM = F(UNNORMALIZED
VARIABLES (CONTINUED))

		M ^a	M ^b	B ^a	B ^b	P
FOR K=10	TABLE REFERENCE NOBS ^c R ²	5-2 395 .589	5-3 395 .571	5-4 104 .727	5-5 95 .797	5-6 35 .895
FOR N=20, K=10, K/N=1:2	R ² SHRINKAGE	.133 -.456	.095 -.476	.424 -.303	.572 -.225	.778 -.117
FOR N=100, K=10, K/N=1:10	R ² SHRINKAGE	.544 -.045	.524 -.047	.697 -.030	.775 -.022	.883 -.012
FOR N=300, K=10, K/N=1:30	R ² SHRINKAGE	.577 -.012	.558 -.013	.719 -.008	.791 -.006	.892 -.003
FOR N=NOBS., K=10	K/N R ² SHRINKAGE	1:40 .578 -.001	1:31 .556 -.015	1:10 .698 -.029	1:10 .773 -.024	1:4 .851 -.044

^aALL OBSERVATIONS USED

^bZERO ERRORS DELETED

^cNOBS. = NUMBER OF OBSERVATIONS

sample size is increased up to $N = 150$ and $k/N = 1:30$, the estimated shrinkage of R^2 is very small (approximately .02), with an expected change in $\hat{R}^2$ from .514 to .499.

For additional discussion concerning the role of validation procedures in multiple regression analysis the reader should consult the following texts and articles (Kerlinger and Pedhazur, 1973; Lord and Novick, 1968; Herzberg, 1969; and Mosier, 1951) (7, 8, 9, 12).

6.4 Sample T Results

Prediction equation results for errors and error rate using sample T programs are presented in the following pages. Results for five predictors were chosen primarily because they were generally found to provide almost as good a prediction of the dependent variable as that obtained when the statistically selected maximum number of variables had been entered in the equation. Usually a maximum of six to nine predictors were entered in the equation. In several subsystem predictions for error rate, the maximum number of predictors was less than five. The maximum number of predictors entered and the multiple correlation coefficient, R , obtained using these predictors are nevertheless reported for each subsystem and for each dependent variable. Due to the large number of regression equations being presented in this section, the analysis of variance tables are presented only for those regressions for which the same set of predictors in all subsystems were available to be used. Additionally, as with sample S, results are reported for each subsystem using all observations and with zero errors deleted from the analysis. Finally, all results presented in this section should be interpreted cautiously. This particularly applies to the results obtained for (1) subsystems B, D, and E, specifically, because of the very limited number of program-observations in each (see Table 2-6), and (2) subsystems F and G, because of the large percentage of programs and source code in each subsystem having zero reported errors (again see Table 2-6, Sample . Subsystem Statistics).

6.4.1 Results for Errors/Program

The following tables of prediction equations and other results for errors/program are presented in this section:

Results	Tables
Errors/Program = $f(\text{Program Structure} + \text{Programmer Variables})^c$	6-31 ^a , 6-32 ^b
Errors/Program = $f(\text{Program Structure Variables Only})^d$	6-33 ^a , 6-34 ^b
Five Predictor Summary	6-35
Best Single Predictor Summary	6-36
Analysis of Variance Tables, (Program Structure Variables Only)	6-37
Validation Results, (Program Structure Variables Only)	6-38 ^a , 6-39 ^b

^aAll observations used

^bZero error programs deleted

^cA maximum of 23 predictors were available for selection in these predictions

^dA maximum of 20 predictors were available for selection in these predictions.

TABLE 6-31. ERRORS/PROGRAM = f(PROGRAM STRUCTURE + PROGRAMMER VARIABLES)

VARIABLE	A	B	C	D	E	F	G ^a	H ^a	FREQUENCY
<u>REGRESSION COEFFICIENTS</u>									
COMP	-.781		.363		-.631	.578			4
COM	-.681		-.273	-.192		.583			4
IO/TS			.159	-.656		-.272			3
TS	1.647				1.271				2
AP	.231					.711			2
SYS				.281	.178				2
DATA			.838			-.614			2
WKLD		-.189			-.186				2
EX/TS		.422		.555					2
LL		1.548							1
IF				.444					1
I/O	.269								1
LL/TS		-1.884							1
IF/TS			-.158						1
LS/TS		.427							1
DATA/TS				-.845					1

TABLE 6-31. ERRORS/PROGRAM = E(PROGRAM STRUCTURE + PROGRAMMER VARIABLES) (CONTINUED)

	A	B	C	D	E	F	G ^a	H ^a
<u>CORRELATION STATISTICS</u>								
I ₁ , TS	1.000	.348	.989	.293	1.000	.491		
I ₂ , ERRORS	.708	.652	.902	.702	.945	.764		
I ₃ , ERRORS	.501	.425	.814	.493	.893	.584		
<u>PREDICTION SUMMARY</u>								
MAX. R	(9).898	(7).980	(9).957	(8).998(10)1.000	(8).919			
R	.871	.966	.939	.987	.999	.906		
R ²	.759	.932	.882	.973	.998	.821		
STD. ERR. EST.	3.511	1.900	4.949	.885	.553	1.359		

^aBEST SINGLE PREDICTOR

^aPROGRAMMER DATA WAS NOT AVAILABLE IN SUBSYSTEMS G AND H.

^bMAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R² ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

TABLE 6-32. ERRORS/PROGRAM = f(PROGRAM STRUCTURE + PROGRAMMER VARIABLES),
ZERO ERRORS DELETED

VARIABLE	A	B	C	D	E	F	G ^a	H ^a	FREQUENCY
REGRESSION COEFFICIENTS									
COMP	- .838		.371		-.642	.444			4
COM	-.621		-.274			.237			3
TS	1.686				1.272				2
LL		1.548		.298					2
AP	.251					.626			2
SYS					.183	-.259			2
WKLD		-.189			-.189				2
IF/TS			-.154	.459					2
IO/TS			.159	-.552					2
EX/TS		.442		.678					2
I/O	-.263								1
DATA			.841						1
RAT/WKLD						.282			1
LL/TS		1.884							1
LS/TS		.427							1
DATA/TS					-.839				1
COM/TS				-.286					1

TABLE 6-32. ERRORS/PROGRAM = E (PROGRAM STRUCTURE + PROGRAMMER VARIABLES),
ZERO ERRORS DELETED (CONTINUED)

	A	B	C	D	E	F	G ^a	H ^a
<u>CORRELATION STATISTICS</u>								
r_{TS}	1.000	.348	.988	.323	1.000	.440		
$r_{E, ERRORS}$	.679	.652	.894	.526	.940	.674		
$r_{E, ERRORS}^2$	.461	.425	.799	.277	.884	.454		
<u>PREDICTION SUMMARY</u>								
MAX. R	(7) .882	(7) .980	(9) .955	(9) .999	(9) 1.000	(8) .960		
R ²	.861	.966	.935	.987	.999	.930		
STD. ERR. EST.	.742	.932	.873	.974	.998	.865		
	3.756	1.900	5.180	.841	.526	1.216		

^a BEST SINGLE PREDICTOR

^a PROGRAMMER DATA WAS NOT AVAILABLE IN SUBSYSTEMS G AND H.

^b MAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R;

R AND R² VALUES ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

TABLE 6-33. ERRORS/PROGRAM = f(PROGRAM STRUCTURE VARIABLES ONLY)

VARIABLE	A	B	C	D	E	F	G	H	FREQUENCY
<u>REGRESSION COEFFICIENTS</u>									
COMP	-.781	.163	-.363		-.952	.317			5
COM	-.681		-.273	-.192		.346	.848		5
EX/TS		.341		.555		-.187	.128		4
TS	1.647=				3.459		-2.269		3
SYS				.281	.145	-.198			3
AP	.231					.658=			2
I/O	.269							-.291	2
DATA			-.838=		-1.763=				2
IF/TS			-.158					-.246	2
LS/TS		.376						.188	2
IO/TS			.159	-.656					2
LL		1.465=							1
IF				.444=					1
BR							2.372=		1
EX								.937=	1
LL/TS		-1.886							1
BR/TS							-.258		1
AP/TS								.223	1
SYS/TS					.896				1

TABLE 6-33. ERRORS/PROGRAM = f(PROGRAM STRUCTURE VARIABLES ONLY) (CONTINUED)

	A	B	C	D	E	F	G	H
<u>CORRELATION STATISTICS</u>								
r_{TS}		1.558	.384	.989	.293	.992	.491	.983
$r_{2, ERRORS}$		.708	.652	.942	.702	.961	.764	.926
$r_{2, ERRORS}$		.501	.425	.814	.493	.924	.584	.857
								.996
								.763
								.582
<u>PREDICTION SUMMARY</u>								
MAX. R	(9)	.839	(8)	.984	(5)	.939	(8)	.998
R ²		.871	.961	.939	.987	.994	.896	.963
STD. ERR. EST.		.759	.922	.882	.973	.988	.892	.928
		3.511	2.834	4.949	.885	1.439	1.368	2.953
								7.917
								(5) .876
								.876
								.767
								7.917

BEST SINGLE PREDICTOR
^a MAXIMUM: NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R² ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

TABLE 6-34. ERRORS/PROGRAM = E (PROGRAM STRUCTURE VARIABLES ONLY), ZERO ERRORS DELETED

VARIABLE	A	B	C	D	E	F	G	H	FREQUENCY
REGRESSION COEFFICIENTS									
COMP	- .830	.163	.371		-1.838	.514			5
COM	- .621		-.274			.386	1.882		4
TS	1.686				3.956		2.397		3
AP	.251					.786	-.881		3
IF/TS			-.154	.459				-.588	3
LL		1.465		.298					2
BR							2.387	.389	2
SYS					.116	-.428			2
I/O	.263							-.284	2
DATA			.841		-2.143				2
LS/TS		.376			-.141				2
IO/TS			.159	-.552					2
EX/TS		.341		.678					2
EX								.775	1
LL/TS		-1.886							1
BR/TS							-.142		1
AP/TS								-.218	1
SYS/TS						.288			1
COM/TS				-.286					1

TABLE 6-34. ERRORS/PROGRAM = F (PROGRAM STRUCTURE VARIABLES ONLY), ZERO ERRORS DELETED
(CONTINUED)

	A	B	C	D	E	F	G	H
<u>CORRELATION STATISTICS</u>								
r_{TS}	1.000	.584	.988	.323	.992	.440	.983	.996
$r_{2, \text{ERRORS}}$	.679	.652	.894	.526	.958	.674	.924	.756
r_{ERRORS}	.461	.425	.799	.277	.918	.454	.854	.572
<u>PREDICTION SUMMARY</u>								
MAX. R	(6).876	(8).984	(8).953	(9).999	(9)1.000	(6).940	(8).976	(5).881
R	.861	.961	.935	.987	.994	.922	.965	.881
R ²	.742	.922	.873	.974	.988	.851	.932	.776
STD. ERR. EST.	3.756	2.034	5.180	.841	1.464	1.279	3.215	7.783

^aBEST SINGLE PREDICTOR

^aMAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R² ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

TABLE 5-35. FIVE PREDICTOR SUMMARY, ERRORS/PROGRAM

VARIABLES	A	B	C	D	E	F
PGM. STRUCTURES + =						
PGMR. VBLS.						
$I =, TS$	1.449	.348	.989	.293	1.888	.419
$I^2, ERRORS$	.748	.652	.982	.742	.945	.764
$I^2 =, ERRORS$	.541	.425	.814	.493	.893	.584
R	.871	.966	.939	.987	.999	.986
R ²	.759	.932	.882	.973	.998	.821
PGM. STRUCTURE + =						
PGMR. VARIABLES						
$I =, TS$	1.888	.348	.988	.323	1.888	.448
$I^2, ERRORS$	.679	.652	.894	.526	.948	.674
$I^2 =, ERRORS$	.461	.425	.799	.277	.884	.454
R	.861	.966	.935	.987	.999	.938
R ²	.742	.932	.873	.974	.998	.865

TABLE 6-35. FIVE PREDICTOR SUMMARY, ERRORS/PROGRAM (CONTINUED)

VARIABLES		A	B	C	D	E	F	G	H
PGM. STRUCTURE VBLIS. ONLY	=	TS	LL	DATA	IF	DATA	AP	BR	EX
	F^2 , TS	1.404	.304	.989	.293	.992	.491	.983	.996
	F^2 , ERRORS	.738	.652	.901	.702	.961	.764	.926	.763
	F^2 , ERRORS	.591	.425	.812	.403	.924	.584	.857	.582
	R^2 , ERRORS	.871	.961	.939	.987	.994	.896	.963	.876
	R^2	.759	.922	.882	.973	.988	.802	.928	.767
PGM. STRUCTURE VBLIS., ZERO ERROR DELETED	=	TS	LL	DATA	EX/TS	DATA	AP	BR	EX
	F^2 , TS	1.000	.304	.988	.323	.992	.440	.983	.996
	F^2 , ERRORS	.679	.652	.894	.526	.958	.674	.924	.756
	F^2 , ERRORS	.461	.425	.799	.277	.918	.454	.854	.572
	R^2 , ERRORS	.861	.961	.935	.987	.994	.922	.965	.881
	R^2	.742	.922	.873	.974	.988	.851	.932	.776
NO. PROGRAMS (7)									
1. (ALL OBSERVATIONS USED)		51	16	39	15	14	37	45	32
2. (ZERO ERRORS DELETED)		44	16	36	14	13	22	34	31

BEST SINGLE PREDICTOR AMONG THE FIVE VARIABLES IN THE REGRESSION EQUATION.

TABLE 6-36. BEST SINGLE PREDICTOR SUMMARY, ERRORS/PROGRAM

VARIABLES	A	B	C	D	E	F
PGM. STRUCTURE +						
<u>PGMR. VBLS.</u>						
r^2 , TS	DATA	LS	DATA	IF	DATA	AP
r^2 , ERRORS	.998	.973	.985	.293	.992	.491
r^2 , ERRORS	.725	.766	.991	.782	.961	.764
r^2 , ERRORS	.526	.578	.812	.493	.924	.584
R	.871	.966	.939	.987	.999	.986
R ²	.759	.932	.882	.973	.998	.821
PGM STRUCTURE +						
<u>PGMR. VBLS.</u>						
r^2 , TS	DATA	LS	DATA	IF	DATA	DATA
r^2 , ERRORS	.989	.973	.988	.282	.992	.966
r^2 , ERRORS	.699	.766	.894	.729	.958	.681
r^2 , ERRORS	.439	.578	.799	.531	.918	.464
R	.851	.966	.955	.987	.999	.938
R ²	.742	.932	.873	.974	.998	.865

BEST SINGLE PREDICTOR

TABLE 6-36. BLST SINGLE PREDICTOR SUMMARY, ERRORS/PROGRAM (CONTINUED)

VARIABLES		A	B	C	D	E	F	G	H
PGM. STRUCTURE VELS. ONLY	=	DATA	LS	DATA	IF	DATA	AP	BR	EX
	r_{TS}	.999	.973	.989	.293	.992	.491	.983	.996
	$r_{TS, ERRORS}$	.725	.769	.911	.792	.961	.764	.926	.763
	$r_{TS, ERRORS}^2$	.526	.578	.812	.493	.924	.584	.857	.582
	$r_{TS, ERRORS}^2$	.871	.911	.939	.987	.994	.896	.963	.876
	R^2	.759	.922	.882	.973	.988	.892	.928	.767
PGM. STRUCTURE VARIABLES, ZERO ERRORS DELETED	=	DATA	LS	DATA	IF	DATA	DATA	BR	EX
	r_{TS}	.989	.973	.928	.222	.992	.966	.983	.996
	$r_{TS, ERRORS}$	.699	.769	.894	.729	.958	.681	.914	.756
	$r_{TS, ERRORS}^2$	.489	.578	.799	.531	.918	.464	.854	.572
	$r_{TS, ERRORS}^2$	.861	.961	.935	.987	.994	.922	.965	.881
	R^2	.742	.922	.873	.974	.988	.851	.932	.776
NO. PROGRAMS									
11 (ALL OBSERVATIONS USED)		51	16	39	15	14	37	45	32
13 (ZERO ERRORS DELETED)		44	16	36	14	13	22	34	31

BEST SINGLE PREDICTOR AMONG ALL PREDICTOR VARIABLES CONSIDERED
BY THE REGRESSION PROCEDURE.

A list of sample T predictor variables that were candidates for entry in the prediction equations for errors/program (i.e., Tables 6-31 thru 6-34) is presented in Table C-2 of Appendix C.

In the summary results (Table 6-35) for errors one can observe that over all subsystems of sample T, regardless of whether or not programmer variables have been entered in the prediction equation, errors/program is consistently highly predictable with obtained values of R^2 in the range from .76 for subsystem A to .99 for subsystem D. Furthermore, there initially appears to be little consistency among the best single predictors of errors in each subsystem. However, considering the high correlations that were reported among the five predictors TS, BR, LS, DATA, and EX, there is a high degree of commonality among the best predictors for five of the eight subsystems (A, C, E, G, and H). This commonality is reflected in the correlation that each best predictor has with Total Source Instructions (TS), reported in Table 6-35 as the value of $r_{*,TS}$ for each subsystem. Note that the best predictors reported are those found among only the five predictors that were in the regression equation. Sometimes the best predictor among all variables to be selected for entry in the equation may not appear in the equation (see, for example, sample S results, section 6.3.3). Table 6-36 presents similar results as in Table 6-35, with the difference being that the best single predictor among all the variables available for selection has been identified.

For these results (Table 6-36), specifically for errors predicted from a linear combination of the program structure variables when the zero errors are deleted from the analysis, the best single predictors from seven of the eight subsystems are all found to have nearly identical high correlations with Total Source Instructions. Although some of the best predictors are different variables, each is basically reflecting the effect of length of program to a high degree. Notice also, in Tables 6-35 and 6-36, that the best predictors' correlations with errors are generally moderate to high. As indicated by the percent of variance in errors accounted for by the best single predictors alone (see values of r^2 *, errors in Table 6-35), they account for approximately 51% (subsystem C) to 94% (subsystem G) of the total variance in errors explained using all five predictors (i.e., the R^2 values). Consequently, this suggests that other program complexity variables are also significantly contributing to error predictability in some subsystems.

Referring to Table 6-31, when errors are predicted using both the program structure and programmer variables, it is observed that only one programmer variable, WKLD, appears in any of the equations; and then for only two of six subsystems. Thus, the RAT and RAT/WKLD variables were not selected as statistically meaningful variables for the five predictor solutions being presented. When the three programmer variables were deleted from the analysis and errors were predicted only as a function of the program structure variables (see Table 6-33), it is observed that (1) the high predictability of errors is basically unaffected and (2) the predictors selected for each subsystem do not significantly differ from those selected when the programmer variables were also included.

In general then it can be stated that the three programmer variables were of little value for predicting errors. Among the possible reasons for this finding is that the RAT and WKLD measures as constructed to evaluate programmers do not truly reflect performance or load, or that these variables should not be analyzed in a linear model.

Clearly, the RAT and WKLD measures as constructed and applied are subjective measures used by the program managers to evaluate programmers. Additionally, the measures used as observation values in the sample T data represent averages of RAT and WKLD for the total number of programmers (anywhere from 1 to 15) responsible for programming each program or function. Moreover, for each subsystem the correlations of each of the RAT, WKLD, and RAT/WKLD variables with errors were generally found to be low to moderate. A combination of these factors is contributing to the relative ineffectiveness of these variables in the prediction of errors. It is of value that these variables were included in this analysis, in spite of their insignificant contributions to the prediction of errors. Since programmers do contribute to errors (as a matter of fact they made the errors analyzed in this study), it is suggested that more objective and standardized personnel and job assessment instruments be developed and applied for future studies. Also, a non-linear model may be more useful than a linear one to evaluate the effectiveness of programmer variables for error prediction purposes. Some additional discussion of the programmer RAT and WKLD variables is presented in Section 7.1.

With respect to the consistency among predictors being selected for each subsystem in the error prediction equations for errors predicted using the program structure variables only (Table 6-33), Computational Statements (COMP) and Comment Statements (COM) appeared in five of the eight subsystem equations. These same predictors appeared in four of the six subsystem equations of Table 6-31, errors predicted using both the program structure and programmer variables.

Finally, regarding the effects on the predictability of errors when zero errors are deleted (Tables 6-32 and 6-34), there are no significant changes in values of R^2 over most subsystems regardless of the predictors that were being used in the analysis. The only exception is subsystem F which has 15 (approximately 41% of the total) zero error programs. The resulting increase was from $R^2 = .80$ to $R^2 = .85$, (Tables 6-33 and 6-34). However, as in the results for sample S, when zero error programs are deleted, different prediction equations usually result. A good example of this can be seen for subsystem D in Tables 6-31 and 6-32. The results are presented here for ease of comparison.

All Observations Used (N=15)		Zero Errors Deleted (N=14)	
IF	.444*	LL	.298
SYS	.281	IF/TS	.459
COM	-.192	IO/TS	-.552
IO/TS	-.656	EX/TS	.678*
EX/TS	.555	COM/TS	-.206
$R^2 =$	.973	$R^2 =$	.974

*Best single predictor of variables in equation.

These differences in predictors and coefficients demonstrate the sensitivity of multiple linear regression analysis to small changes in correlations.

The analysis of variance and prediction equation validation results are presented in Tables 6-37 and 6-38, respectively, for regressions where only the program structure variables are considered. The analysis of variance results indicate that the predictor variables account for a significant proportion of the total variance of errors/program. The F statistic for each regression equation is statistically significant at less than the .001 level of significance. For the validation results of Table 6-38, when 10 programs are considered as a validation sample, only a low to moderate shrinkage in the squared multiple correlation coefficient (R^2) for each subsystem occurs. This shrinkage was the greatest for subsystem A with the largest number of observations ($n=51$). When zero errors are deleted from subsystem A (see Table 6-39), it is observed that the shrinkage is considerably less as compared to the shrinkage of R^2 when all observations are used. As is indicated in both sets of validation results, all equations showed increasing consistency of prediction with the original values of R^2 as larger validation samples were used to compute the shrinkage estimates for each subsystem. Considering these results, it appears that had actual validation samples been available, each of the obtained prediction equations would have shown consistent predictability of errors in these samples.

In summary for errors/program, we can state that high predictability results when using a linear combination of program structure variables for errors collected throughout the

TABLE 6-37. ANALYSIS OF VARIANCE TABLES, SAMPLE T SUBSYSTEMS,
ERRORS/PROGRAM = F(PROGRAM STRUCTURE VARIABLES),
ALL OBSERVATIONS USED

SUBSYSTEM		SUM OF SQUARES	DF	MEAN SQUARE	F	PROBABILITY
A	REGRESSION	1742.524	5	348.505	28.275	<.001
	RESIDUAL	554.644	45	12.325		
	TOTAL	2297.169	50			
B	REGRESSION	492.634	5	98.527	23.819	<.001
	RESIDUAL	41.365	14	4.137		
	TOTAL	533.998	15			
C	REGRESSION	6057.383	5	1211.477	49.460	<.001
	RESIDUAL	808.298	33	24.494		
	TOTAL	6865.681	38			
D	REGRESSION	258.277	5	51.655	65.894	<.001
	RESIDUAL	7.055	9	.784		
	TOTAL	265.333	14			
E	REGRESSION	1318.300	5	263.660	127.411	<.001
	RESIDUAL	16.555	8	2.069		
	TOTAL	1334.856	13			
F	REGRESSION	235.024	5	47.005	25.123	<.001
	RESIDUAL	58.002	31	1.871		
	TOTAL	293.026	36			

TABLE 6-37. ANALYSIS OF VARIANCE TABLES, SAMPLE T SUBSYSTEMS,
 ERRORS/PROGRAM = 1 (PROGRAM STRUCTURE VARIABLES),
 ALL OBSERVATIONS USED (CONTINUED)

SUBSYSTEM		SUM OF SQUARES	DF	MEAN SQUARE	F	PROBABILITY
G	REGRESSION	4399.551	5	879.910	100.879	<.001
	RESIDUAL	340.174	39	8.722		
	TOTAL	4739.727	44			
H	REGRESSION	5350.145	5	1070.029	17.073	<.001
	RESIDUAL	1629.554	26	62.675		
	TOTAL	6979.699	31			

TABLE 6-38. SAMPLE T, VALIDATION RESULTS FOR FIVE PREDICTOR REGRESSION EQUATIONS,
ERRORS/PROGRAM = f(PROGRAM STRUCTURE VARIABLES), ALL OBSERVATIONS USED

	A	B	C	D	E	F	G	H
FOR K=5 (TABLE REFERENCED: 5-35) NOBS. ^a R ²	51 .759	16 .922	39 .882	15 .973	14 .988	37 .892	45 .928	32 .767
FOR N=10, K=5, K/N=1:2 R ² SHRINKAGE	.458 -.381	.824 -.898	.734 -.148	.939 -.834	.973 -.815	.554 -.248	.838 -.898	.476 -.291
FOR N=25, K=5, K/N=1:5 R ² SHRINKAGE	.696 -.863	.982 -.828	.851 -.831	.966 -.877	.985 -.883	.751 -.851	.989 -.891	.786 -.861
FOR N=50, K=5, K/N=1:10 R ² SHRINKAGE	.732 -.827	.913 -.885	.869 -.813	.978 -.883	.987 -.881	.788 -.822	.928 -.888	.741 -.826
FOR N=NOBS., K=5 K/N R ² SHRINKAGE	1:10 .732 -.827	1:3 .883 -.839	1:8 .864 -.818	1:3 .958 -.815	1:3 .988 -.888	1:7 .778 -.832	1:9 .919 -.889	1:6 .722 -.845

^aNOBS. = NUMBER OF OBSERVATIONS.

TABLE 6-39. SAMPLE T, VALIDATION RESULTS FOR FIVE PREDICTOR REGRESSION EQUATIONS,
ERROR/PROGRAM = E(PROGRAM STRUCTURE VARIABLES), ZERO ERRORS DELETED

	A	B	C	D	E	F	G	H
FOR K=5 (TABLE REFERENCED: 5-36) NOBS ^a R ²	44 .742	16 .922	36 .873	14 .974	13 .988	22 .851	34 .932	31 .776
FOR N=10, K=5, K/N=1:2 R ² SHRINKAGE	.419 -.323	.824 -.098	.714 -.159	.941 -.003	.973 -.015	.665 -.186	.847 -.085	.496 -.280
FOR N=25, K=5, K/N=1:5 R ² SHRINKAGE	.75 -.067	.902 -.020	.840 -.033	.967 -.007	.985 -.003	.812 -.039	.914 -.018	.718 -.058
FOR N=50, K=5, K/N=1:10 R ² SHRINKAGE	.714 -.028	.913 -.009	.859 -.014	.971 -.003	.987 -.001	.835 -.016	.925 -.002	.751 -.025
FOR N=NOBS., K=5 K/N R ² SHRINKAGE	1:9 .708 -.034	1:3 .883 -.039	1:7 .851 -.022	1:3 .958 -.016	1:3 .979 -.009	1:4 .844 -.047	1:7 .920 -.012	1:6 .731 -.045

^aNOBS. = NUMBER OF OBSERVATIONS.

validation, integration, acceptance, and operational phases of software development. Predictors which heavily reflect length of program are generally found to be the best predictors of errors; however, other program structure-complexity variables also contribute significantly to the predictability of errors. And finally, the programmer rating and workload variables as defined for this study are found to be of little value for contributing to the prediction of errors.

6.4.2 Results for Error Rate/Program

The results for the analysis of error rate/program presented in this section are as follows:

Results	Tables
Error Rate/Program = $f(\text{Program Structure} + \text{Programmer Variables})^c$	6-40 ^a , 6-41 ^b
Error Rate/Program = $f(\text{Program Structure Variables Only})^d$	6-42 ^a , 6-43 ^b
Five Predictor Summary	6-44
Analysis of Variance Tables (Program Structure Variables Only)	6-45

^aAll observations used

^bZero error rates deleted

^cA maximum of 23 predictors were available for selection in these predictions.

^dA maximum of 20 predictors were available for selection in these predictions.

TABLE 6-4#. ERROR RATE/PROGRAM = f(PROGRAM STRUCTURE + PROGRAMMER VARIABLES)

VARIABLE	A	B	C	D	E	F	G ^a	H ^a	FREQUENCY
<u>REGRESSION COEFFICIENTS</u>									
AP/TS	.484±	-.176	.314±	.785±		.607±			5
EX/TS	.377			.249	.167	-.213			4
COM/TS	-.187	-.180	-.427		.516				4
SYS/TS		.790±	.431		.591±				3
COMP	-.306				-.183				2
PAT	.232					.173			2
LL/TS				-.177		-.120			2
LS/TS			-.267	.413					2
TS					.308				1
IF		-2.383							1
AP				.173					1
IF/TS		2.095							1
BR/TS			.500						1
DATA/TS						.258			1

TABLE 6-4. ERROR RATE/PROGRAM = f(PROGRAM STRUCTURE + PROGRAMMER VARIABLES) (CONTINUED)

	A	B	C	D	E	F	G ^a	H ^a
<u>CORRELATION STATISTICS</u>								
r^2 , ERROR RATE	.618	.982	.782	.926	.876	.746		
r^2 , ERROR RATE	.382	.814	.612	.857	.767	.557		
<u>PREDICTION SUMMARY</u>								
MAX. R	(7) .818	(9) .999	(6) .935 (18) .999	(7) .998	(6) .849			
R	.777	.988	.924	.989	.992	.881		
R ²	.603	.961	.854	.978	.985	.641		
STD. ERR. EST.	1.934	.471	1.398	.554	1.196	.848		

BEST SINGLE PREDICTOR

^aPROGRAMMER DATA WAS NOT AVAILABLE FOR SUBSYSTEMS G AND H.
^bMAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R² VALUES ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

TABLE 6-41. ERROR RATE/PROGRAM = $f(\text{PROGRAM STRUCTURE} + \text{PROGRAMMER VARIABLES})$,
ZERO ERROR RATES DELETED

VARIABLE	A	B	C	D	E	F	G ^a	H ^a	FREQUENCY
REGRESSION COEFFICIENTS									
AP/TS	.454	-.176	.218	1.121		.698			5
SYS/TS	.168	.798	.496		.535	.218			5
EX/TS	.324			.278	.185				3
COM/TS		-.188	-.338		.514				3
IF		-2.383			.174				2
SYS	-.292					-.274			2
BR/TS			.372	-.185					2
LS/TS			-.168	.358					2
RAT	.286								1
RAT/WKLD						-.156			1
IF/TS		2.895							1
IO/TS				-.252					1
COMP/TS					-.117				1

TABLE 6-41. ERROR RATE/PROGRAM = f(PROGRAM STRUCTURE + PROGRAMMER VARIABLES),
ZERO ERROR RATES DELETED (CONTINUED)

	A	B	C	D	E	F	G ³	H ^a
<u>CORRELATION STATISTICS</u>								
Σ , ERROR RATE	.676	.982	.784	.925	.887	.797		
Σ , ERROR RATE	.457	.814	.615	.856	.787	.635		
<u>PREDICTION SUMMARY</u>								
MAX. R	(7) .847	(9) .999	(9) .954 (10) 1.000	(8) .999	(4) .864			
R	.785	.988	.936	.987	.994	.864		
R ²	.617	.961	.876	.974	.987	.747		
STD. ERR. EST.	1.968	.471	1.316	.616	1.147	.781		

^aBEST SINGLE PREDICTOR

^bPROGRAMMER DATA WAS NOT AVAILABLE FOR SUBSYSTEM G AND H.

^cMAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R² VALUES ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

^dONLY FOUR PREDICTORS SELECTED AS A MAXIMUM SET FOR THIS R VALUE.

TABLE 6-42. ERROR RATE/PROGRAM = E(PROGRAM STRUCTURE VARIABLES ONLY)

<u>VARIABLE</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	<u>H</u>	<u>FREQUENCY</u>
<u>REGRESSION COEFFICIENTS</u>									
AP/TS	.499±	-.176	.314±	.785±	.591±	.653±	-.223		6
SYS/TS	.236	.790±	.431					.758	5
EX/TS	.338			.249	.167	-.214	.632±		5
COM/TS	-.248	-.180	-.427		.516				4
LL/TS				-.177		-.113	-.280		3
SYS	-.194						-.326		2
COMP					-.183		.299		2
IF/TS		2.495						-.390	2
BR/TS			.500					2.494±	2
LS/TS			-.267	.413					2
IO/TS						-.131		-2.593	2
TS					.300				1
IF		-2.383							1
AP				.173					1
I/O								-.559	1
DATA/TS						.242			1

TABLE 6-62. ERROR RATE/PROGRAM = E(PROGRAM STRUCTURE VARIABLES ONLY) (CONTINUED)

	A	B	C	D	E	F	G	H
<u>CORRELATION STATISTICS</u>								
r^2 , ERROR RATE	.618	.982	.782	.926	.876	.746	.479	.456
r^2 , ERROR RATE	.382	.814	.612	.857	.767	.557	.221	.288
<u>PREDICTION SUMMARY</u>								
MAX. R	(6).793	(8).998	(7).934	(18).999	(8).997	(5).795	(5).848	(12).937
R	.767	.988	.924	.989	.992	.795	.648	.865
R^2	.588	.961	.854	.978	.985	.631	.489	.749
STD. ERR.	1.971	.471	1.398	.554	1.196	.818	1.13	1.339
EST.								

BEST SINGLE PREDICTOR

a MAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R^2 VALUES ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

TABLE 6-43. ERROR RATE/PROGRAM = f(PROGRAM STRUCTURE VARIABLES ONLY),
ZERO ERRORS RATES DELETED

VARIABLE	A	B	C	D	E	F	G	H	FREQUENCY
<u>REGRESSION COEFFICIENTS</u>									
AP/TS	.527=	-.176	.210=	1.121=		.703=		.314	6
SYS/TS	.193	.790=	.496		.535=	.214		.852=	6
EX/TS	.265			.278	.185		.480=		4
COM/TS	-.168	-.180	-.333		.514				4
SYS	-.239					-.283	-.203		3
BR/TS			.372	-.185				2.198	3
I/O							.185	.712	2
IF/TS		2.095			.174				2
LS/TS			-.160	.350					2
IO/TS				-.252				-2.808	2
LL							.488		1
IF		-2.383							1
LL/TS							-.598		1
COMP/TS					-.117				1

TABLE 6-43. ERROR RATE/PROGRAM = $\bar{E}$ (PROGRAM STRUCTURE VARIABLES ONLY),
ZERO ERRORS DELETED (CONTINUED)

	A	B	C	D	E	F	G	H
<u>CORRELATION STATISTICS</u>								
$\bar{E}$, ERROR RATE	.676	.982	.784	.925	.887	.797	.427	.476
$\bar{E}$, ERROR RATE	.457	.814	.615	.856	.787	.635	.182	.227
<u>PREDICTION SUMMARY</u>								
MAX R	(6) ^a .786	(8).998	(8).959	(9).999	(5).994	(3).854 ^b	(5).598	(8).913
R	.778	.988	.935	.987	.994	.854 ^b	.598	.867
R ²	.686	.961	.874	.974	.987	.723	.349	.752
STD. ERR. EST.	1.995	.471	1.316	.616	1.147	.713	1.092	1.329

^aBEST SINGLE PREDICTOR

^bMAXIMUM NUMBER OF PREDICTORS SELECTED AND MAXIMUM MULTIPLE R; R AND R² VALUES ON LINES BELOW ARE OBTAINED USING BEST SET OF FIVE PREDICTORS AS LISTED ABOVE.

^cONLY THREE PREDICTORS SELECTED AS A MAXIMUM SET FOR THIS OBTAINED R VALUE.

TABLE 6-44. FIVE PREDICTOR SUMMARY, ERROR RATE/PROGRAM

VARIABLES		A	B	C	D	E	F	G	H
PGM. STRUCTURE +									
PGMR. VRBLS.									
r^2 , TS		.264	-.517	-.378	-.512	-.581	-.198		
r^2 , ERROR RATE		-.232	-.671	-.235	-.518	-.614	-.896		
r^2 , ERROR RATE		.618	.932	.782	.926	.876	.746		
r^2 , ERROR RATE		.392	.814	.612	.857	.767	.557		
r^2 , ERROR RATE		.777	.984	.924	.989	.992	.881		
R^2		.603	.961	.854	.978	.985	.641		
PGM. STRUCTURE +									
PGMR. VRBLS.									
r^2 , TS		.293	-.517	-.428	-.545	-.612	-.357		
r^2 , ERROR RATE		-.361	-.671	-.317	-.555	-.496	-.442		
r^2 , ERROR RATE		.676	.932	.734	.925	.887	.797		
r^2 , ERROR RATE		.457	.814	.615	.856	.787	.635		
r^2 , ERROR RATE		.785	.984	.936	.987	.994	.864 ^a		
R^2		.617	.961	.876	.974	.987	.747		

TABLE 6-44. FIVE PREDICTOR SUMMARY, ERROR RATE/PROGRAM (CONTINUED)

VARIABLES		A	B	C	D	E	F	G	H
<u>PGM. STRUCTURE</u>									
<u>VBLS. ONLY</u>									
Σ		AP/TS	SYS/TS	AP/TS	AP/TS	SYS/TS	AP/TS	EX/TS	BR/TS
Σ_{TS}		-.264	-.517	-.378	-.512	-.581	-.190	-.021	-.278
$\Sigma_{TS, ERROR RATE}$		-.258	-.671	-.235	-.508	-.414	-.096	-.167	-.211
$\Sigma_{TS, ERROR RATE}$		.618	.902	.782	.926	.876	.746	.470	.456
$\Sigma_{TS, ERROR RATE}$		.382	.814	.612	.857	.767	.557	.221	.208
$\Sigma_{TS, ERROR RATE}$		.767	.980	.924	.989	.992	.795	.640	.865
R		.588	.961	.854	.978	.985	.631	.409	.749
R^2									
<u>PGM. STRUCTURE</u>									
<u>VBLS., ZERO</u>									
<u>ERROR RATES</u>									
<u>DELETED</u>									
Σ		AP/TS	SYS/TS	AP/TS	AP/TS	SYS/TS	AP/TS	EX/TS	SYS/TS
Σ_{TS}		-.293	-.517	-.428	-.545	-.612	-.357	-.308	-.446
$\Sigma_{TS, ERROR RATE}$		-.361	-.671	-.317	-.555	-.496	-.442	-.070	-.251
$\Sigma_{TS, ERROR RATE}$		.676	.902	.784	.925	.887	.797	.427	.476
$\Sigma_{TS, ERROR RATE}$		.457	.814	.615	.857	.787	.635	.182	.227
$\Sigma_{TS, ERROR RATE}$		.778	.980	.935	.987	.994	.850 ^b	.590	.867
R		.606	.961	.874	.974	.987	.723	.349	.752
R^2									
<u>NO. PROGRAMS</u>									
N (ALL OBSERVATIONS USED)		51	16	39	15	14	37	45	32
n (ZERO ERROR RATES DELETED)		44	16	36	14	13	22	34	31

^aBEST SINGLE PREDICTOR

^aONLY FOUR PREDICTORS SELECTED AS A MAXIMUM SET FOR THIS R VALUE.

^bONLY THREE PREDICTORS SELECTED AS A MAXIMUM SET FOR THIS R VALUE.

A list of sample T predictor variables that were candidates for entry in the prediction equations for error rate is presented in Table C-3 of Appendix C.

The summary results for error rate (Table 6-44) indicate that regardless of whether or not the programmer variables have been made available for selection in the prediction equation, moderate to high predictability is achieved for each subsystem of sample T, with the exception of subsystem F. The best single predictor of error rate is one of the two normalized Program Interface variables, AP/TS and SYS/TS, which appear consistently as the best predictions in seven of the eight subsystems when the program structure variables with zero error rates deleted are considered. Additionally, these interface variables account for a large portion of the overall predictability in each subsystem, and other program complexity variables appear to be significantly contributing to error rate predictions (i.e., as evidenced by the difference between R^2 and r^2_{*} , error rate for each subsystem).

Also a consistent finding in the analysis is that the majority of predictors in the equations for error rate are the normalized variables. This finding is observed for all prediction results (Tables 6-40 to 6-43) obtained for error rate.

By examining Table 6-40, it is noted that when the programmer variables RAT, WKLD, and RAT/WKLD are made available for selection, only the Programmer Rating variable (RAT) is selected, and then only for two of the six subsystems. In Table 6-42 which reports the predictions after removing the programmer variables from the analysis, the results show that

(1) the predictability of error rate is essentially unaffected, and (2) the predictors selected for each subsystem do not significantly differ from those selected when the programmer variables are made available to enter the regression equation. In the analysis the correlations of each of the programmer variables with error rate were observed to be low over each of the subsystems; low to moderate correlations were observed for these variables with the errors/program measure. Thus, the programmer variables as defined for this study have a negligible contribution to the predictability of both errors per program and error rate per program.

To examine the consistency of predictors selected over all subsystems, all results (Tables 6-40 thru 6-43) show that the three normalized variables AP/TS, SYS/TS, and EX/TS, are the variables most frequently appearing when predicting error rate over all 28 regression equations.

Examining the predictability of error rate when zero error rates are deleted, it is found that with the exception of subsystems F and G, no major changes in the value of R^2 over each of the subsystems are observed. Since subsystems F and G are quite unique relative to each other and to the six remaining subsystems with respect to the large percentage of error free programs in each, a further analysis of error rate in F and G is presented in Section 7.2.

The analysis of variance results presented in Table 6-45 indicate that the F statistic for each subsystem's regression equation is significant at less than the .001 level of significance. The linear predictions then account for a statisti-

TABLE 6-45. ANALYSIS OF VARIANCE TABLES, SAMPLE T SUBSYSTEMS,
ERROR RATE/PROGRAM = E (PROGRAM STRUCTURE VARIABLES),
ALL OBSERVATIONS USED

SUBSYSTEM	SUM OF SQUARES	DF	MEAN SQUARE	F	PROBABILITY
A	REGRESSION RESIDUAL TOTAL	5 45 50	49.868 3.886	12.829	<.001
B	REGRESSION RESIDUAL TOTAL	5 18 15	18.912 .222	49.189	<.001
C	REGRESSION RESIDUAL TOTAL	5 33 38	75.336 1.955	38.542	<.001
D	REGRESSION RESIDUAL TOTAL	5 9 14	24.287 .386	79.268	<.001
E	REGRESSION RESIDUAL TOTAL	5 8 13	145.469 1.431	181.624	<.001
F	REGRESSION RESIDUAL TOTAL	5 31 36	7.188 6.696	18.615	<.001

TABLE 6-45. ANALYSIS OF VARIANCE TABLES, SAMPLE T SUBSYSTEMS,
 ERROR RATE/PROGRAM = $\bar{E}$ (PROGRAM STRUCTURE VARIABLES),
 ALL OBSERVATIONS USED (CONTINUED)

SUBSYSTEM		SUM OF SQUARES	DF	MEAN SQUARE	F	PROBABILITY
G	REGRESSION	34.535	5	6.907	5.397	<.001
	RESIDUAL	49.989	39	1.282		
	TOTAL	84.444	44			
H	REGRESSION	138.672	5	27.734	15.474	<.001
	RESIDUAL	46.681	26	1.792		
	TOTAL	185.272	31			

cally significant proportion of the variability in error rate, wherein this proportion is measured by the values of R^2 for each subsystem.

6.4.3 Sample T Prediction Consistency Analysis

The predictions for both errors and error rate are generally moderate to very high for each data sample over all predictions obtained. For the analysis of sample T results, a set of five predictor variables is used uniformly across all subsystems to predict errors, and a different set of five is used to predict error rate. A significant change in the squared multiple correlation coefficient from that obtained for the best five predictors in each subsystem would indicate inconsistency.

For errors/program the five predictors from subsystem A are used (i.e., TS, AP, I/O, COMP, and COM). For error rate, the subsystem A predictors (SYS, AP/TS, SYS/TS, EX/TS, COM/TS) are also chosen. These predictors were used since they were automatically selected by the regression procedure based on the largest sample of observations ($n=51$) available for sample T.

The results of this consistency analysis are presented in Tables 6-46 (for errors) and 6-47 (for error rate). All predictors in each set of variables are forced into the equation in order to obtain comparable results over each subsystem. Since the predictors for subsystem A had not been selected by the regression procedure for the other subsystems, a reduction in the value of R^2 for each subsystem except A was expected.

TABLE 6-46. ERRORS/PROGRAM = f(TS, AP, I/O, COMP, COM), CONSISTENCY SUMMARY

VARIABLES	A	B	C	D	E	F	G	H
<u>REGRESSION COEFFICIENTS</u>								
TS	1.647	-.655	-.682	3.659	1.408	-.205	-.227	1.481
AP	-.231	-.625	-.153	-.207	-.002	-.649	-.073	-.181
I/O	-.269	-.449	-.144	-.526	-.009	-.105	-.044	-.155
COMP	-.781	-.524	-.370	-2.782	-.658	-.314	-.499	-.292
COM	-.601	-1.100	-.355	-.355	-.153	-.467	-.721	-.631
<u>CORRELATION STATISTICS</u>								
$r_{TS, TS}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$r_{TS, AP}$		1.000	1.000	1.000	1.000	1.000	1.000	1.000
$r_{TS, I/O}$			1.000	1.000	1.000	1.000	1.000	1.000
$r_{TS, COMP}$				1.000	1.000	1.000	1.000	1.000
$r_{TS, COM}$					1.000	1.000	1.000	1.000
$r_{AP, TS}$		1.000	1.000	1.000	1.000	1.000	1.000	1.000
$r_{AP, AP}$			1.000	1.000	1.000	1.000	1.000	1.000
$r_{AP, I/O}$				1.000	1.000	1.000	1.000	1.000
$r_{AP, COMP}$					1.000	1.000	1.000	1.000
$r_{AP, COM}$						1.000	1.000	1.000
$r_{I/O, TS}$			1.000	1.000	1.000	1.000	1.000	1.000
$r_{I/O, AP}$				1.000	1.000	1.000	1.000	1.000
$r_{I/O, I/O}$					1.000	1.000	1.000	1.000
$r_{I/O, COMP}$						1.000	1.000	1.000
$r_{I/O, COM}$							1.000	1.000
$r_{COMP, TS}$						1.000	1.000	1.000
$r_{COMP, AP}$							1.000	1.000
$r_{COMP, I/O}$								1.000
$r_{COMP, COMP}$								
$r_{COMP, COM}$								
$r_{COM, TS}$								1.000
$r_{COM, AP}$								
$r_{COM, I/O}$								
$r_{COM, COMP}$								
$r_{COM, COM}$								
<u>PREDICTION SUMMARY</u>								
R^2	.871	.913	.914	.802	.988	.881	.930	.833
R	.759	.816	.835	.643	.977	.776	.864	.695
n	51	16	39	15	14	37	45	32
<u>BEST SINGLE PREDICTOR</u>								

TABLE 6-47. ERROR RATE/PROGRAM = F(SYS, AP/TS, SYS/TS, EX/TS, COM/TS), CONSISTENCY SUMMARY

VARIABLE	A	B	C	D	E	F	G	H
<u>REGRESSION COEFFICIENTS</u>								
SYS	-.194	-.187	-.083	-.026	-.022	-.046	-.025	-.225
AP/TS	-.499	-.013	-.511	1.428	-.033	-.751	-.241	-.214
SYS/TS	-.236	-.777	-.484	-.557	-.506	-.040	-.190	-.495
EX/TS	-.338	-.126	-.213	-.206	-.245	-.117	-.632	-.033
COM/TS	-.248	-.005	-.513	-.077	-.487	-.024	-.152	-.110
<u>CORRELATION STATISTICS</u>								
r^2 , ERROR RATE	-.618	-.932	-.782	-.926	-.876	-.746	-.470	-.452
r^2 , ERROR RATE	-.382	-.814	-.612	-.858	-.767	-.556	-.221	-.205
<u>PREDICTION SUMMARY</u>								
R^2	-.767	-.929	-.887	-.943	-.982	-.754	-.547	-.580
R	-.588	-.863	-.788	-.890	-.964	-.568	-.299	-.336
N	51	16	39	15	14	37	45	32

=BEST SINGLE PREDICTOR

For errors/program, it is observed that, using the same set of predictors, moderate to high predictability is maintained over all subsystems. For error rate, less consistency of prediction results. Nevertheless, for six of the eight subsystems (A thru F) error rate predictions are still in the moderate to high range.

One particular interpretation of these results is that since the same variables appear in the equations for each subsystem, an estimate of how these variables contribute to errors in general may be obtained. In spite of the apparent differences among the subsystems of sample T, there is remarkable consistency in results of applying the same set of five predictors to all subsystems. These results may then apply to programming in general, or at least to command and control systems using JOVIAL J4.

Since the predictability of errors and error rate was higher for sample T programs than for those of sample S, predictions for errors and error rate were obtained using the two distinct sets of predictors over all 249 program observations of sample T. These results are reported in Table 6-48. The predictions were observed to be in the moderate range for both errors and error rate. Given the larger sample involved and the increased variability over all variables that results from this aggregation over all subsystems, the predictions may be generally more indicative of the true values of R , R^2 , and other correlation and regression statistics in the population of programs of which these 249 programs are but a sample.

TABLE 6-48. SAMPLE T PREDICTION RESULTS USING ALL SUBSYSTEMS
(N=249)

VARIABLE	REGRESSION COEFFICIENT	PREDICTION SUMMARY
----------	---------------------------	-----------------------

ERRORS/PROGRAM

TS	.758*	r^2 *, ERRORS= .765
AP	.218	r^2 *, ERRORS= .586
I/O	- .050	R= .797
COMP	- .116	R^2 = .635
COM	.043	STD. ERR. EST.= 6.464

ERROR RATE/PROGRAM

SYS	- .041	r^2 *, ERRORS= .641
AP/TS	.459*	r^2 *, ERRORS= .412
SYS/TS	.206	R= .714
EX/TS	.235	R^2 = .510
COM/TS	.085	STD. ERR. EST.= 2.171

* BEST SINGLE PREDICTOR

Several comments are important here regarding the consistency analysis results obtained using each subsystem (Tables 6-46 and 6-47) vis-a-vis the results observed using all sample T programs (Table 6-48). First, for errors/program, at the subsystem level (Table 6-46), Total Source Instructions (TS) is the best predictor of errors, and the four program complexity variables (AP, I/O, COMP, COM) are contributing significantly to error predictability. In contrast, at the aggregate level over all subsystems, Total Source Instructions alone accounts for almost 92% of the total variance explained by the five predictors. At this level the four complexity variable have only a negligible effect on the predictability of errors/program. An important implication is suggested by these results. When estimating the total number of errors likely to be found as a result of formal testing for a group of programs that are functionally heterogeneous (and similar in nature to those of sample T), the size or length of each program may be the single most important predictor. Whereas for programs that are more functionally homogeneous, other program complexity variables in addition to program length should be considered in the prediction process to achieve some initial estimate of errors/program.

For error rate/program, regardless of whether the prediction is based on sets of programs at the individual subsystem level or using the entire sample of 249 programs, each of the program complexity variables (SYS, AP/TS, SYS/TS, EX/TS, COM/TS) in combination are contributing to the predictability obtained. At the subsystem level this is true for at least five of the eight subsystems, A, C, E, G and H. This result suggests that the overall variability of the error rate measure and its predictors (four of which were normalized program

complexity measures) were essentially unaffected by the aggregation over all subsystems. Thus, error rate/program and its predictors may be more stable measures to be used for prediction purposes, regardless of the functional mixture of programs being considered.

7.0 ADDITIONAL ANALYSIS

7.1 Error Rate and Programmer Variables

Since the programmer Rating (RAT) and Workload (WKLD) variables proved to be of no predictive value to errors or error rate when combined with program structure variables, it was decided that a more thorough analysis of these variables particularly as they related to error rate, would be performed.

Essentially in this analysis the average workload and average error rate for different categories of programmer ratings are calculated. Then, how different levels of programmer rating and workload affect the error rate is determined. This analysis is performed over all program-observations of sub-systems A thru E. The results of this analysis are presented in Table 7-1.

Each of the average error rates is statistically tested (using the 't' test) for significant differences with each neighboring mean value in the following manner: the average error rate for programmers rated less than 10 (i.e., 1.50) was tested and found not statistically different from the average error rate of programmer's with ratings from 11 to 12 (i.e., 3.22). The average error rates for these two groups combined was then tested for a significant difference with the next error rate value (1.62), showing again no significant difference. The average of the three error rates was then compared with the value of 2.99; again no statistical significance was found. (This same procedure was followed for each of the remaining groups of programmer ratings). The only

TABLE 7-1. SAMPLE T, ERROR RATE AND PROGRAMMER
VARIABLE RELATIONSHIPS

PROGRAMMER RATING	NUMBER PROGRAMS	AVERAGE WORK LOAD	AVERAGE ERROR RATE
≤ 10	8	1.00	1.50
11-12	18	.97	3.22
13-14	15	1.10	1.62
15-16	22	1.18	2.99
17	27	1.33	4.36
18	16	1.33	2.05
19	19	1.34	2.56
20	10	1.34	2.85
(N=135)			

statistically significant differences that exist occur as indicated between the bracketed groupings of means; i.e., (1) between the average error rates for programmers rated in the ≤ 10 to 16 and 17 categories, and (2) between the error rates for programmers rated 17 or higher (i.e., the difference between 4.36 and 2.44 was significant at the .05 level).

Basically the following observations can be made from these results:

- (1) the lower rated programmers with lighter workloads do produce significantly fewer errors per 100 lines of code as compared with the higher rated programmers who had the heavier workloads;
- (2) the same high rated programmers (i.e., RAT = 17) produce significantly more errors per 100 lines of code as compared with the top rated programmers, irregardless of the relatively high workload each group had; and
- (3) the highest rated programmers having the heaviest workload produced as many errors per 100 lines of code as did the lowest rated programmers having the lowest workload.

Additionally, using this kind of analysis as compared to the linear regression approach, one can clearly see the non-linearity in the relationship of error rate with both programmer workload and rating. That is, as both RAT and WKLD variables increase, error rate also increases up to a point (i.e., RAT = 17) and then becomes smaller with still increasing values of RAT and WKLD.

As cited in information provided about the RAT and WKLD variables of sample T, many of the programmers specifically rated in the RAT = 17 category were managers not only managing the software development effort but also programming at the same time. This seems to suggest that managers who also contribute to the programming effort contribute significantly to errors in programs, more so than programmers who do nothing but program.

The method of analysis used here best approximates the standard analysis of variance approach typically used to analyze experiments involving one or more factors. Although the analysis of variance methodology is generally not used for purposes of prediction, when properly applied it can be very useful for identifying relationships among data variables that may go undiscovered using linear regression analysis.

Additionally, the kinds of interpretations of data relationships that can be made using the analysis of variance approach may in many instances have more operational meaning than those allowed using the regression approach. For this reason, the analysis of variance approach and other methods of analysis (e.g., contingency table analysis using chi-square tests, non-linear regression models) are strongly suggested as additional methodologies which can and should be employed, where applicable, in future software reliability analysis studies where programmer, project, test, software environment, and error data are all available for analysis.

7.2 Error Rate and Source Instructions

Throughout the study of error rate/program it was observed that the correlation between error rate and total source instructions was consistently low, but negative, with one exception. For each of the sample T subsystems, the correlations, as reported in Table 7-2, exist between error rate and the Total Source Instructions variable (TS), when the error free programs are included and then excluded from the analysis.

Basically, the consistent increase in magnitude of each of these correlations (with the exception of subsystem G) and the consistency of the low to moderate linear relationship between error rate and source instructions over all subsystems, when the error free programs were deleted, supports the hypothesis that longer programs are less thoroughly tested. The hypothesis asserts that longer programs were less thoroughly tested relative to the shorter programs. This is likely to be the case since as the length of a program increases, a more rapid than linear rate of increase in the number of paths through the program usually would occur, thus increasing a program's complexity. This increased complexity would then result in the longer programs requiring more time to test and thus they might very well be less thoroughly tested.

The fact that this hypothesis can be considered as a plausible explanation here and that these results are not just correlational anomalies is borne out in the following analysis of error rate presented for both subsystems F and G (see Tables 7-3 and 7-4, respectively).

TABLE 7-2. CORRELATIONS BETWEEN ERROR RATE AND TOTAL SOURCE INSTRUCTIONS FOR SAMPLE T SUBSYSTEMS

SUBSYSTEM	$r_{TS, \text{ERROR RATE}}$	
	<u>USING ALL OBSERVATIONS</u>	<u>WHEN ERROR FREE PROGRAMS DELETED</u>
A	-.238 (51)	-.361 (44)
B ^a	-.671 (16)	-.671 (16)
C	-.235 (39)	-.317 (36)
D	-.508 (15)	-.555 (14)
E	-.414 (14)	-.496 (13)
F	-.096 (37)	-.442 (22)
G	.167 (45)	-.070 (34)
H	-.211 (32)	-.251 (31)

^aSUBSYSTEM B HAD NO ERROR FREE PROGRAMS.

TABLE 7-3. SUBSYSTEM F ERROR RATE ANALYSIS

PROGRAM CLASSIFICATION	NUMBER PROGRAMS	AVERAGE SOURCE INSTRUCTIONS	AVERAGE ERROR RATE/ PROGRAM
ALL PROGRAMS	37	256.35	0.96
ABOVE AVERAGE ON TS	12	573.08	0.73
BELOW AVERAGE ON TS	25	104.32	1.07
PROGRAMS WITH ONE OR MORE ERRORS	22	321.32	1.62
ABOVE AVERAGE ON TS	8	658.75	0.98
BELOW AVERAGE ON TS	14	128.50	1.98
PROGRAMS WITH ZERO ERROR RATES	15	161.07	0.00
LOWEST 7 ON TS	7	89.57	2.76
NEXT 8 ON TS	8	193.75	1.09
HIGHEST 7 ON TS	7	698.88	1.08
TS, ERROR RATE (N=37)	-.096		
TS, ERROR RATE (N=22)	-.442		

TABLE 7-4. SUBSYSTEM G ERROR RATE ANALYSIS

PROGRAM CLASSIFICATION	NUMBER PROGRAMS	AVERAGE SOURCE INSTRUCTIONS	AVERAGE ERROR RATE/ PROGRAM
ALL PROGRAMS	45	328.53	1.51
ABOVE AVERAGE ON TS	13	752.54	1.57
BELOW AVERAGE ON TS	32	156.28	1.49
PROGRAMS WITH ONE OR MORE ERRORS	34	398.91	2.00
ABOVE AVERAGE ON TS	10	864.80	1.61
BELOW AVERAGE ON TS	24	204.79	2.16
PROGRAMS WITH ZERO ERROR RATES	11	111.00	0.00
LOWEST 11 ON TS	11	118.09	2.77
NEXT 12 ON TS	12	264.67	1.69
HIGHEST 11 ON TS	11	826.18	1.57
$\bar{x}_{TS}$, ERROR RATE (N=45)	.167		
$\bar{x}_{TS}$, ERROR RATE (N=34)	-.070		

For both sets of results, all programs were classified as being either above or below the average value of source instructions; the average value of error rate/program was then computed for each classification. Essentially for both subsystems, no differences were observed between the average error rates for the shorter programs versus the longer programs, whereas the differences between the average length of the shorter compared to that of the longer programs was strikingly different. Now, when the error free programs are removed and only the programs with errors are classified according to the same procedure, significant differences between the error rates for the shorter versus the longer programs do exist. Finally, when this same set of programs is classified even further into three categories as indicated, the negative, low level linear relationship between error rate and source instructions becomes apparent.

These results are interpreted as lending support to the hypothesis that longer programs, in particular those with no errors reported, were less thoroughly tested than the shorter ones.

In summary then, it is not known whether the programs with zero reported errors are truly error free. When these programs are considered to have unreported or latent errors remaining in them and are removed from the analysis, then the consistency of the relationship between error rate and length of program becomes more strikingly apparent, and hypotheses concerned with thoroughness of testing of the longer versus that of the shorter programs become more readily testable.

8.0 CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

The major purpose of this study was to determine how predictable programming error measures are from a combination of program characteristic variables using multiple linear regression analysis. By examining the degree of predictability obtained, the effectiveness of the linear regression model in software error prediction studies may then be evaluated. With respect to this purpose, given the analysis results obtained, the following conclusions can be drawn:

- The predictability of programming error measurements are variable, ranging from very low to very high. For the errors/program measure, predictability is found to be consistently in the moderate to very high range. For the error rate/program measure, predictability is generally less than that obtained for errors/programs; with the predictability ranging from very low to high and with less consistency than errors/program throughout all the predictions obtained.
- The variability in the predictions obtained over both data samples is considered to be strongly related in varying degrees to each of the following factors:
 - a. functional difference among the various programs that were developed
 - b. differences in the programming language used

- c. the length of time formal error data collection was carried out
- d. the amount and thoroughness of testing of each program
- e. inadequacy of the linear model to provide perfect predictability
- f. other programmer, project, and management factors affecting the software development process.

8.2 , Direct Recommendations

The following set of recommendations discuss measures that can be taken to bring about an increased consistency of prediction of programming error measures in future software error prediction studies. These recommendations pertain to the predictor variables, the programming error variables, applications of the multiple regression procedure, and software testing procedures.

(1) Predictor Variables - Predictor variables should be accurately identified and concisely defined prior to the beginning of software development. Predictors should be identified which not only reflect selected program and language-specific characteristics, but moreover they should include a variety of candidate programmer, management, and software development environment variables which are suspected of reasonably affecting the quality and reliability of the software being developed.

-16-

A baseline set of predictor variables should be defined and applied over the general range of software projects so that consistency of measurement can be obtained. This baseline set can be compiled from the results of this study's five and ten predictor summaries. As further studies identify additional predictor variables, the baseline set should be expanded.

Data for these predictor variables should be collected throughout the successive phases of software development. Any significant changes or modifications should be recorded, dated, and the cause of the error determined.

In order to benefit the generalizability of future error prediction studies wherein different programming languages will be involved, present and future research effort should be directed toward identifying those language and program characteristics that may be equated or made comparable between two or more languages.

(2) Programming Error Measures - These measures should be concisely defined and collected throughout all phases of software development. In addition to a description of the errors, their symptoms, and the program changes required to correct the errors, other data should be collected which would include when the error was found, the method used to detect the error, and an estimate of the total effort (e.g., man-hours, computer time, documentation changes, etc.) involved in error identification and correction.

116

Of particular importance for error prediction purposes is that a taxonomy or typology of programming errors be developed such that more definitive predictions can be developed for separate versus a gross aggregation of error types.

Each of the errors, regardless of error type, should further be classified and weighted with respect to their criticality or severity for impeding the achievement of the software development project objectives. Once this has been accomplished, then predictions of errors having different criticality can be performed and the most important variables for each can then be identified. Existing error collection tools could be expanded to classify and weight errors. Manually collected error data should also be classified and weighted either manually or by interfacing with error collection tools.

(3) Multiple Regression Applications - A parallel prediction approach which utilizes multiple regression analysis applied at various milestones or stages of software development, testing, and operational usage is recommended for future error analysis and prediction studies.

The parallel prediction approach proposes to make separate predictions of errors during each of the designated stages during the project, using a specified set of predictors for which data would be collected at each of these stages. Of special interest in this analysis is the identification of particular criterion error measures (e.g., gross error counts or error rates for given error types, or errors weighted by severity) whose overall predictability is changing meaningfully over time. Also of interest at each stage is the relative importance of each of the predictor variables in the equations. Clearly, any of the predictors found to systematically increase

(or decrease) in explanatory power over time is deserving of further attention.

Additionally, since data would be available from prior time periods using this approach, one could investigate any time lag relationships that may exist between the predictors and errors when predicting errors at later stages in the project.

Using this parallel predictions approach, both linear and non-linear models should be investigated.

(4) Software Testing Procedures - Throughout this report the thoroughness of testing in each of the two data samples was repeatedly stressed as an important factor contributing towards the identification of errors in programs. It is strongly recommended that the amount of testing of program modules be measured in software development projects as much as possible so that errors weighted by amount of testing can be analyzed. This measure of amount of testing should address possible paths tested and range of possible inputs and outputs exercised.

8.3 Recommendations for Further Research

Throughout the course of this study, the need to perform additional research on the available data became obvious. However, such research was beyond the scope of this contract. Four of the major topics of interest and value to other software quality and reliability studies are presented.

(1) Non-Linear Regression Analysis - This study suggests that non-linear regression will improve both the consistency of the predictions and the predictability of each of the programming error measurements. A continued analysis of both sample S and sample T data along with other available data samples should be investigated for error prediction purposes using non-linear regression models.

Consider, for example, that the actual error rate of programs that have been thoroughly tested increases up to a certain level for a given number of source instructions, and then it increases only slightly thereafter for continuing increases in program length. That is, the error rate becomes almost constant after a certain program length. This error rate could be estimated for the sample S and sample T data using the equation

$$Y' = a + bx + cx^2 \quad (8.0)$$

The independent variable (x) in this equation is the total source instructions variable (X1 or TS) for either sample. The dependent or predicted variable (Y') is the new estimated value of error rate obtained for each program in the two data

samples that are presently available. This newly estimated or revised error rate (hypothesized for thoroughly tested programs) would replace the observed error rates for many of the programs used in this study, which were probably not thoroughly tested. Figure 8-1 depicts the observed error rate-program length relationships for the three projects of sample S as contrasted with the new or revised estimates of error rate using the non-linear model described above.

Using these revised or new estimates (Y') of what the actual error rates for these sample programs should be, this new variable can be predicted using various non-linear forms of the predictor variables in a multiple regression equation.

(2) Predictions By Error Type and Severity - As mentioned in the earlier discussion of recommendations, it is strongly recommended that errors be classified according to type and severity and that predictions be obtained for those classifications. During this study no data was available for either sample that would enable the assignment of such classifications to the errors in each program. If this data is available, it would then be possible to develop different regression equations for different types of errors. The results of such an analysis should show higher predictability and give insights more directly related to cause-effect than were obtained by aggregating errors.

(3) Analysis of Error Free Programs - If the zero reported error programs that appeared throughout the analysis were in reality error free, then an analysis directed at determining the characteristics of these programs would be very meaningful. Those characteristics which differentiate error free from error prone modules could be determined.

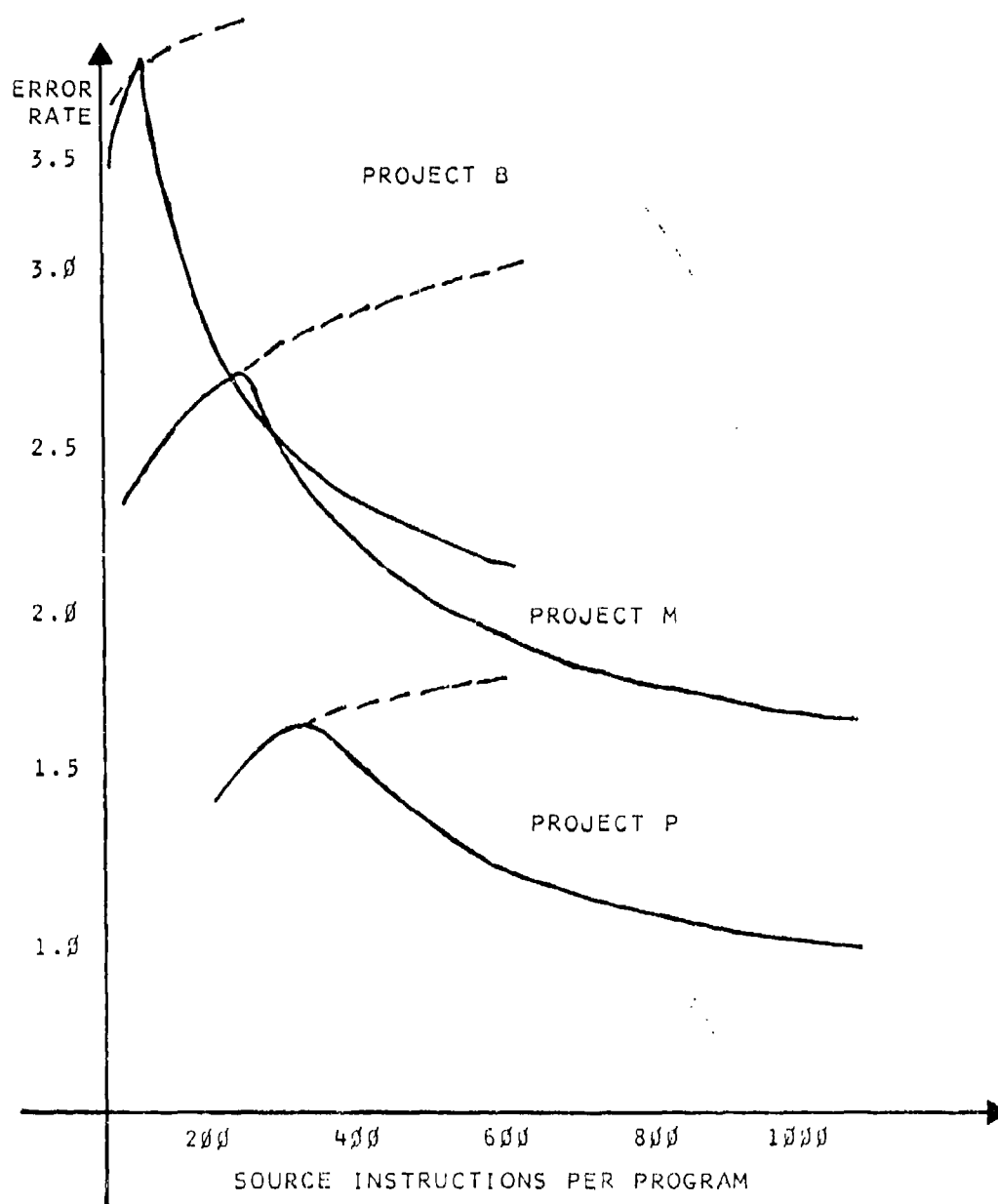


FIGURE 3-1. ERROR RATE AND SOURCE INSTRUCTIONS RELATIONSHIP FOR PROJECTS M, B, AND P, CONTRASTED WITH ESTIMATES OF ERROR RATE USING NON-LINEAR MODEL

(4) Analysis of Constant Size Programs - In addition to the normalization procedure used to construct additional predictor variables for this study and the partial correlation procedure discussed in Section 5.0, one straightforward procedure for controlling for the effect of program length in this analysis would be to analyze different sets of constant size programs in each given sample. This could be accomplished by developing groups of programs, for example, that had an average program length of 50, 150, 250, etc. source instructions each. Then only those programs falling within 1 or 1/2 standard deviation above and below the mean would, for statistical purposes, be considered as a group with constant program length. A major benefit of this analysis would be to identify how the predictability of errors and error rate differs over each of these groups, and what variables are the most important predictors of errors for smaller, medium, and longer length programs.

8.4 Proposed Support Tools and Techniques

To most effectively apply the results of the study toward obtaining more error free software, support tools and techniques are required. This section proposes tools and techniques conceptualized throughout this study.

(1) Collection of Data for Predictor Variables - Predictor variables which define program characteristics can be measured from source code. Those measurement programs already in existence for specific projects are known as scanner programs. To enable the most accurate and effective method of measuring the predictor variables, Language Scanners should be developed as part of support software packages. Language Scanners should measure, at a minimum, the baseline set of predictor variables defined under the first recommendation of Section 8.2. Language Scanners can easily be added, for example, to a programming support library which stores and maintains source code as well as performs all compilations. A Language Scanner can be provided to support each language in the same manner as a pre-compiler is provided for each structured language.

(2) Evaluation of High Order Languages - Existing high order languages should be analyzed to identify those characteristics which are most closely correlated with errors. When such characteristics are identified, it is necessary to determine which of these characteristics do cause errors. For those cases when cause is established, preventive measures could be introduced. This type of evaluation can apply not only to existing languages but also to any language under development or modification.

116

(3) Test Support Tools - Test support tools are under development throughout the software industry, particularly in the areas of identifying program paths. Such a tool should also contain a capability for weighting (or accepting manual weighting) the various paths. Weighting might include such factors as frequency of use and criticality. Testing emphasis could then be directed according to the weighting scheme.

Another test support tool to be developed is an input/output range definer. Representative inputs can be selected for path testing from the required range of values. The number of different outputs can be compared against the required range of outputs. Untested outputs can be identified.

Both testing aids proposed here will assist in identifying desired testing and in determining where available resources should be applied during testing. A combination of number of paths exercised and the range of inputs and outputs tested is a further measure related to software system reliability.

* 8.5 Summary of Recommendations

The following list is a summary of recommendations resulting from this study. Items (1) through (4) are direct applications of the results.

- (1) Definition and collection of data for predictor variables.
- (2) Error classification and weightings.
- (3) Apply regression models, both linear and non-linear, throughout the software development process.
- (4) Define testing techniques which measure thoroughness of testing.

Items (5) through (9) recommend further research.

- (5) Investigate non-linear multiple regression.
- (6) Classifiy errors according to type and severity and obtain predictions for these classifications.
- (7) Continue analysis of error-free programs from a broader data sample.
- (8) Continue analysis of programs grouped by relative size from a broader data sample.
- (9) Apply the prediction model obtained by this study over a broader data sample.

Items (10) through (12) identify software support tools and techniques which will assist in implementing the preceeding recommendations.

- (10) Develop Language Scanners to measure predictor variables.
- (11) Evaluation of high order languages.
- (12) Develop test support tools which a) identify and weight program paths, and b) determine representative test input to prcdue required outputs.

9.0 REFERENCES

1. Althausen, R. P. Multicollinearity and non-additive regression models. In H. M. Blalock (Ed.), Causal Models in the Social Sciences. Chicago: Aldine, 1971.
2. Dixon, W. J. (Ed.) BMDP, Biomedical Computer Programs. Berkeley, Calif.: University of California Press, 1975.
3. Draper, N. R., Smith, H. Applied Regression Analysis. New York: John Wiley & Sons, Inc., 1966.
4. Ferguson, G. A. Statistical Analysis in Psychology & Education. New York: McGraw-Hill, 1971.
5. Gordon, R. A. Issues in multiple regression. American Journal of Sociology, 1968, 73, 592-616.
6. Harris, R. J. A Primer of Multivariate Statistics. New York: Academic Press Inc., 1975.
7. Herzberg, P. A. The parameters of cross-validation. Psychometrika, 1969, 34, (Monogr. Suppl. 16).
8. Kerlinger, F. N., Pedhazur, E. J. Multiple Regression in Behavioral Research. New York: Holt, Rinehart and Winston, Inc., 1973.
9. Lord, E. M., Novick, M. R. Statistical Theories of Mental Test Scores. Reading, Mass.: Addison-Wesley, 1968.
10. McNemar, Q. Psychological Statistics. New York: John Wiley, 1962.
11. Mitchell, J. N., Overnultz, L. M., Rynkiweicz, R.F. Software Reliability and Productivity Modeling Report. Morris Plains, N.J.: Bell Laboratories, 1976.
12. Mosier, C. I. Problems and Designs of Cross-Validation. Educational and Psychological Measurement, 1951, 11, 5-11.
13. Thayer, T.A. et al. Software Reliability Study. (TRW Systems Group, Final Technical Report), RADC-TR-76-238 dated August 1976 (AD A030798).

14. Okimoto, C. H. An Approach for Identifying and Eliminating Error-Prone Modules. Endicott, N.Y.: IBM Corp., TR 01.1898, 1975.
15. Rama Sastry, M. V. Some Limits in the Theory of Multicollinearity. American Statistician, 1970, 24, 1, 39-40.
16. Van de Gerr, J. P. Introduction to Multivariate Analysis for the Social Sciences. San Francisco: W. H. Freeman & Co., 1971.

APPENDIX A

(This Appendix contains a discussion of considerations on the measure of error to be analyzed in software error prediction studies).

416

Considerations on the Measure of Error
to be Analyzed in Software
Error Prediction Studies

In addition to the very complex problem of the definition of an error, there is the puzzling problem of what measure should be used to represent the errors in a routine, module, or program, once the data is collected. (Of course, this decision should be made prior to data collection). Is it the number of errors in a program? If so, this leaves something to be desired, since a very short program of say, ten instructions with three errors, would "look" the same as the dependent variable of a very large program of say, 3000 instructions with three errors. These extremes in program length do exist in the data analyzed for sample S and sample T. The most desirable solution to the problem would be to collect data in such quantities that every individual program length could be represented a number of times, consider each program length a sample, and then analyze the data accordingly. Since such a luxury is not likely to exist, some compromise is required.

Error rate; i.e., errors per 100 source instructions, is quite commonly used as an error measure and is somewhat more meaningful than number of errors. It still suffers, however, from the fact that a program of ten instructions with two errors receives the same weight as a program of 100 instructions with 20 errors, one of 1000 instructions with 200 errors, etc. If the regression of errors as a function of number of source instructions is in fact linear, and the regression line goes through the origin, this procedure would be quite proper. By linear regression it is not meant that the relationship should be perfect, but that a

straight line gives as good a representation of the relationship as any curve of higher degree.

If the regression is in fact linear and the line does not pass through the origin, a better approach would be to eliminate the linear effect of number of source instructions on number of errors by partial correlation and then analyze the residual error. It is indicated then, also, that the influence of source instructions should be eliminated from all other independent variables.

It would seem that in view of all the problems inherent in the identification and definition of an appropriate error measure, a hopeless situation exists. However, if the problem is formulated in terms of what variables are used to predict errors, it becomes more clear. First, recognizing that number of errors and error rate are two distinct, though related^a, approaches to error measurement, two separate dependent variables exist. To predict number of errors, there are variables such as gross characteristics of program length, mixtures of instructions of various types, program complexity metrics, etc. To predict error rate, all of these variables used to predict errors, plus other program characteristics normalized to program length (either by their rates per 100 source instructions or by partial correlation) also exist.

When error rate is used as a measure of program reliability, the question arises as to whether to normalize errors by the number of total source instructions, the number of executable source instructions, or even by the number of generated machine language instructions. This is a problem since the error measure always being analyzed is

^arelated in that if the total number of errors in the program equals zero, the error rate is necessarily zero.

the number of errors found, and at any phase of testing, there usually exists some (unknown) number of latent errors in the program(s).

To clarify this point, consider that two programs (designated as A and B) exist, each with 400 source instructions. A had 300 executable instructions and B has 200 executable instructions. During test and integration (conducted by automated means) suppose four errors in A and four errors in B were found. Normalizing by source instructions, the error rate for A is 1.0 (error/100 source instructions). For B the error rate is also 1.0. However, by using executable instructions, the error rate for A is 1.33. For B the error rate is 2.0. (Not considered here is the problem of how many of the executable instructions were actually executed in the test). Thus, hypothetically nothing is known of the number of errors in the non-executable (i.e., not non-executed) portion of the programs.

The solution here is that the normalizing factor should depend upon the method of testing. If, for example, all the code is examined (as in a code review), the number of source instructions should be used. If only the executable portion of the code is examined for errors, the number of executable instructions should be used.

APPENDIX B

(This Appendix contains descriptions of the predictor variables for both samples S and sample T as discussed in Section 2.0).

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS

VARIABLE	DESCRIPTION
<u>UNNORMALIZED:</u>	
X 1	NUMBER OF SOURCE INSTRUCTIONS
X 2	NUMBER OF ENTRY POINTS
X 3	NUMBER OF EXIT POINTS
X 4	NUMBER OF USING INSTRUCTIONS WHICH ESTABLISH DATA STRUCTURE INTERFACE
X 5	NUMBER OF COMMENT STATEMENTS
X 6	NUMBER OF LABELED SOURCE INSTRUCTIONS
X 7	NUMBER OF INSTRUCTIONS PERFORMING THE ARITHMETIC FUNCTIONS ADD, SUBTRACT, MULTIPLY, DIVIDE AND EXPONENTIATION
X 8	NUMBER OF UNCONDITIONAL BRANCH INSTRUCTIONS
X 9	NUMBER OF CALL/LINK INSTRUCTIONS
X10	NUMBER OF SYSTEM MACROS (DOES NOT INCLUDE THE INSTRUCTIONS WHICH ARE GENERATED BY THE MACROS)
X11	NUMBER OF USER WRITTEN MACROS (DOES NOT INCLUDE THE INSTRUCTIONS WHICH ARE GENERATED BY THE MACROS)
X12	NUMBER OF EQUATE INSTRUCTIONS USED TO EQUATE SYMBOLS TO REGISTERS, IMMEDIATE DATA, OR OTHER VALUES
X13	NUMBER OF COMMENTED SOURCE INSTRUCTIONS

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X14	NUMBER OF AND/OR LOGICAL CONNECTORS
X15	NUMBER OF CONDITIONAL BRANCH INSTRUCTIONS
X16	NUMBER OF INVOKED FUNCTIONS SUCH AS FLOOR, SQRT, LOG, ATAN ETC.
X17	NUMBER OF INSTRUCTIONS PERFORMING SCALE/ROUND OPERATIONS
X18	NUMBER OF SHORT DO INSTRUCTIONS (WHEN MACHINE CODE GENERATED BY CENTRAN DO INSTRUCTION WILL BE LESS THAN 17 BYTES. USED TO MINIMIZE LOOP EXECUTION TIME)
X19	NUMBER OF NESTED SHORT DO LOOPS
X20	NUMBER OF LOCK MACROS
X21	NUMBER OF SOURCE INSTRUCTIONS WITHIN SHORT DO LOOPS
X22	NUMBER OF ADDRESS VARIABLES REFERENCED
X23	NUMBER OF TIMES ADDRESS VARIABLES ARE REFERENCED
X24	NUMBER OF TIMES ALL BINARY VARIABLES ARE REFERENCED
X25	NUMBER OF CHARACTER VARIABLES REFERENCED
X26	NUMBER OF TIMES CHARACTER VARIABLES ARE REFERENCED

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X27	NUMBER OF FIXED POINT VARIABLES REFERENCED
X28	NUMBER OF TIMES FIXED-POINT VARIABLES ARE REFERENCED
X29	NUMBER OF FLOATING-POINT VARIABLES REFERENCED
X30	NUMBER OF TIMES FLOATING-POINT VARIABLES ARE REFERENCED
X31	NUMBER OF HEXADECIMAL VARIABLES REFERENCED
X32	NUMBER OF TIMES HEXADECIMAL VARIABLES ARE REFERENCED
X33	NUMBER OF LABELED-ARRAY VARIABLES REFERENCED
X34	NUMBER OF TIMES LABELED-ARRAY VARIABLES ARE REFERENCED
X35	NUMBER OF REGISTER VARIABLES REFERENCED
X36	NUMBER OF TIMES REGISTER VARIABLES ARE REFERENCED
X37	NUMBER OF VARIABLES WHICH WERE REFERENCED BUT NOT DEFINED WITHIN THE PROGRAM (UNDEFINED VARIABLES)
X38	NUMBER OF TIMES UNDEFINED VARIABLES ARE REFERENCED

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X39	TOTAL NUMBER OF VARIABLES REFERENCED
X40	NUMBER OF TIMES ALL VARIABLES ARE REFERENCED
X41	NUMBER OF DO LOOPS
X42	NUMBER OF NON-NESTED DO LOOPS
X43	NUMBER OF DO LOOPS NESTED AT SECOND LEVEL
X44	NUMBER OF DO LOOPS NESTED AT THIRD LEVEL
X45	NUMBER OF DO LOOPS NESTED AT FOURTH LEVEL
X46	NUMBER OF DO LOOPS NESTED AT FIFTH LEVEL
X47	NUMBER OF DO LOOPS NESTED AT SIXTH LEVEL OR LOWER
X48	NUMBER OF SOURCE INSTRUCTIONS IN ALL NON-NESTED DO LOOPS
X49	NUMBER OF SOURCE INSTRUCTIONS IN ALL SECOND LEVEL DO LOOPS
X50	NUMBER OF SOURCE INSTRUCTIONS IN ALL THIRD LEVEL DO LOOPS
X51	NUMBER OF SOURCE INSTRUCTIONS IN ALL FOURTH LEVEL DO LOOPS

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X52	NUMBER OF SOURCE INSTRUCTIONS IN ALL FIFTH LEVEL DO LOOPS
X53	NUMBER OF SOURCE INSTRUCTIONS IN ALL SIXTH LEVEL OR LOWER DO LOOPS
X54	AVERAGE NUMBER OF OPERATORS PER ARITHMETIC INSTRUCTION (X7) X NUMBER OF SOURCE INSTRUCTIONS
ERRORS/PROGRAM	NUMBER OF ERRORS FOUND IN PROGRAM DURING THE TEST AND INTEGRATION PHASE OF SOFTWARE SYSTEM DEVELOPMENT WHICH REQUIRED A CHANGE TO THE PROGRAM'S CODE.
<u>NORMALIZED^a:</u>	
X56	NUMBER OF ENTRY POINTS/X1
X57	NUMBER OF EXIT POINTS/X1
X58	NUMBER OF USING INSTRUCTION WHICH ESTABLISH DATA STRUCTURE INTERFACE/X1
X59	NUMBER OF COMMENT STATEMENTS/X1
X60	NUMBER OF LABELED SOURCE INSTRUCTIONS/X1
X61	NUMBER OF INSTRUCTIONS PERFORMING THE ARITHMETIC FUNCTIONS ADD, SUBSTRACT, MULTIPLY, DIVIDE AND EXPONENTIATION/X1
X62	NUMBER OF UNCONDITIONAL BRANCH INSTRUCTIONS/X1

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X63	NUMBER OF CALL/LINK INSTRUCTIONS/X1
X64	NUMBER OF SYSTEM MACROS/X1
X65	NUMBER OF USER WRITTEN MACROS/X1
X66	NUMBER OF EQUATE INSTRUCTIONS USED TO EQUATE SYMBOLS TO REGISTERS, IMMEDIATE DATA, OR OTHER VALUES/X1
X67	NUMBER OF COMMENTED SOURCE INSTRU- CTIONS/X1
X68	NUMBER OF AND/OR LOGICAL CONNECTORS/X1
X69	NUMBER OF CONDITIONAL BRANCH INSTRUCTIONS/X1
X70	NUMBER OF INVOKED FUNCTIONS SUCH AS FLOOR, SQRT, LOG, ATAN ETC./X1
X71	NUMBER OF INSTRUCTIONS PERFORMING SCALE/ROUND OPERATIONS/X1
X72	NUMBER OF SHORT DO INSTRUCTIONS/X1
X73	NUMBER OF NESTED SHORT DO LOOPS/X1
X74	NUMBER OF LOCK MACROS/X1
X75	NUMBER OF SOURCE INSTRUCTIONS WITHIN SHORT DO LOOPS/X1
X76	NUMBER OF ADDRESS VARIABLES REFERENCED/X1
X77	NUMBER OF TIMES ADDRESS VARIABLES ARE REFERENCED/X1

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X78	NUMBER OF TIMES ALL BINARY VARIABLES ARE REFERENCED/X1
X79	NUMBER OF CHARACTER VARIABLES REFERENCED/X1
X80	NUMBER OF TIMES CHARACTER VARIABLES ARE REFERENCED/X1
X81	NUMBER OF FIXED-POINT VARIABLES REFERENCED/X1
X82	NUMBER OF TIMES FIXED-POINT VARIABLES ARE REFERENCED/X1
X83	NUMBER OF FLOATING-POINT VARIABLES REFERENCED/X1
X84	NUMBER OF TIMES FLOATING-POINT VARIABLES ARE REFERENCED/X1
X85	NUMBER OF HEXADECIMAL VARIABLES REFERENCED/X1
X86	NUMBER OF TIMES HEXADECIMAL VARIABLES ARE REFERENCED/X1
X87	NUMBER OF LABELED-ARRAY VARIABLES REFERENCED/X1
X88	NUMBER OF TIMES LABELED-ARRAY VARIABLES ARE REFERENCED/X1
X89	NUMBER OF REGISTER VARIABLES REFERENCED/X1
X90	NUMBER OF TIMES REGISTER VARIABLES ARE REFERENCED/X1

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X91	NUMBER OF VARIABLES WHICH WERE REFERENCED BUT NOT DEFINED WITHIN THE PROGRAM (UNDEFINED VARIABLE)/X1
X92	NUMBER OF TIMES UNDEFINED VARIABLES ARE REFERENCED/X1
X93	TOTAL NUMBER OF VARIABLES REFERENCED/X1
X94	NUMBER OF TIMES ALL VARIABLES ARE REFERENCED/X1
X95	NUMBER OF DO LOOPS/X1
X96	NUMBER OF NON-NESTED DO LOOPS/X1
X97	NUMBER OF DO LOOPS NESTED AT SECOND LEVEL/X1
X98	NUMBER OF DO LOOPS NESTED AT THIRD LEVEL/X1
X99	NUMBER OF DO LOOPS NESTED AT FOURTH LEVEL/X1
X100	NUMBER OF DO LOOPS NESTED AT FIFTH LEVEL/X1
X101	NUMBER OF DO LOOPS NESTED AT SIXTH LEVEL OR LOWER/X1
X102	NUMBER OF SOURCE INSTRUCTIONS IN ALL NON-NESTED DO LOOPS/X1
X103	NUMBER OF SOURCE INSTRUCTIONS IN ALL SECOND LEVEL DO LOOPS/X1

TABLE B-1. SAMPLE S PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
X134	NUMBER OF SOURCE INSTRUCTIONS IN ALL THIRD LEVEL DO LOOPS/X1
X135	NUMBER OF SOURCE INSTRUCTIONS IN ALL FOURTH LEVEL DO LOOPS/X1
X136	NUMBER OF SOURCE INSTRUCTIONS IN ALL FIFTH LEVEL DO LOOPS/X1
X137	NUMBER OF SOURCE INSTRUCTIONS IN ALL SIXTH LEVEL OR LOWER DO LOOPS/X1
X138	AVERAGE NUMBER OF OPERATORS PER ARITHMETIC INSTRUCTION (X7) X NUMBER OF SOURCE INSTRUCTIONS/X1
ERROR RATE /PROGRAM	NUMBER OF ERRORS PER 100 LINES OF SOURCE CODE FOUND DURING THE TEST AND INTEGRATION PHASE OF SOFTWARE SYSTEM DEVELOPMENT WHICH REQUIRED A CHANGE TO THE PROGRAM'S CODE/X1

^a ALL NORMALIZED VARIABLE VALUES WERE ACTUALLY COMPUTED BY
MULTIPLYING THE RESPECTIVE UNNORMALIZED VALUES BY 100/X1.
EACH NORMALIZED VARIABLE IS INTERPRETED THEN AS THE VALUE
OF THE ORIGINAL OR UNNORMALIZED VARIABLE PER 100 LINES OF
SOURCE CODE.

TABLE 3-2. SAMPLE T PREDICTOR VARIABLE DESCRIPTIONS

VARIABLE	DESCRIPTION
<u>UNNORMALIZED:</u>	
1. TS	TOTAL SOURCE STATEMENTS IN THE PROGRAM (TS=NEX+EX)
2. LL	COMPUTED LOOP COMPLEXITY ^a
3. IF	COMPUTED IF COMPLEXITY ^b
4. BR	TOTAL PROGRAM BRANCHES
5. LS	NUMBER OF LOGICAL STATEMENTS IN PROGRAM
6. AP	NUMBER OF DIRECT PROGRAM INTERFACES WITH OTHER APPLICATION PROGRAMS (NOT A COUNT OF CALLS TO OTHER PROGRAMS)
7. SYS	NUMBER OF DIRECT PROGRAM INTERFACES WITH OPERATING SYSTEM OR SYSTEM SUPPORT PROGRAMS (NOT A COUNT OF CALLS TO SYSTEM PROGRAMS)
8. I/O	NUMBER OF INPUT/OUTPUT STATEMENTS IN PROGRAM
9. COMP	NUMBER OF COMPUTATIONAL STATEMENTS IN PROGRAM
10. DATA	NUMBER OF DATA HANDLING STATEMENTS IN PROGRAM
11. NEX	NUMBER OF NON-EXECUTABLE STATEMENTS IN PROGRAM
12. EX	NUMBER OF EXECUTABLE STATEMENTS IN PROGRAM
13. TI	TOTAL PROGRAM INTERFACES WITH OTHER PROGRAMS (TI=AP + SYS)

TABLE B-2. SAMPLE T PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
14. COM	NUMBER OF COMMENT STATEMENTS IN PROGRAM (COMMENTS ARE NOT INCLUDED IN THE COUNT OF NON-EXECUTABLE STATEMENTS, NEX)
15. RAT	AVERAGE PROGRAMMER RATING (THIS VALUE IS AN AVERAGE BASED ON THE RATINGS OF EACH PROGRAMMER WHO WORKED ON THE PROGRAM)
16. WKLD	AVERAGE WORKLOAD OF PROGRAMMERS WHO WORKED ON THE PROGRAM.
17. ERRORS/PROGRAM	NUMBER OF PROGRAMMING ERRORS FOUND IN THE PROGRAM WHICH REQUIRED A CHANGE TO THE PROGRAM'S CODE
18. RAT/WKLD	RATIO OF AVERAGE PROGRAMMER RATING TO AVERAGE PROGRAMMER WORKLOAD
<u>NORMALIZED^C:</u>	
19. LL/TS	MEASURE OF LOOP COMPLEXITY PER 100 LINES OF SOURCE CODE
20. IF/TS	MEASURE OF IF COMPLEXITY PER 100 LINES OF SOURCE CODE
21. BR/TS	NUMBER OF BRANCHES PER 100 LINES OF SOURCE CODE
22. LS/TS	NUMBER OF LOGICAL STATEMENTS PER 100 LINES OF SOURCE CODE
23. AP/TS	NUMBER OF APPLICATION PROGRAM INTERFACES PER 100 LINES OF SOURCE CODE

TABLE B-2. SAMPLE T PREDICTOR VARIABLE DESCRIPTIONS
(CONTINUED)

VARIABLE	DESCRIPTION
24. SYS/TS	NUMBER OF SYSTEM PROGRAM INTERFACES PER 100 LINES OF SOURCE CODE
25. IO/TS	NUMBER OF INPUT/OUTPUT STATEMENTS PER 100 LINES OF SOURCE CODE
26. COMP/TS	NUMBER OF COMPUTATIONAL STATEMENTS PER 100 LINES OF SOURCE CODE
27. DATA/TS	NUMBER OF DATA HANDLING STATEMENTS PER 100 LINES OF SOURCE CODE
28. NEX/TS	NUMBER OF NON-EXECUTABLE STATEMENTS PER 100 LINES OF SOURCE CODE
29. EX/TS	NUMBER OF EXECUTABLE PROGRAM STATEMENTS PER 100 LINES OF SOURCE CODE
30. TI/TS	NUMBER OF TOTAL PROGRAM INTERFACES PER 100 LINES OF SOURCE CODE
31. COM/TS	NUMBER OF COMMENT STATEMENTS PER 100 LINES OF SOURCE CODE
32. ERROR RATE/PROGRAM	NUMBER OF PROGRAMMING ERRORS PER 100 LINES OF SOURCE CODE PER PROGRAM

Table B-2. SAMPLE T PREDICTOR VARIABLE DESCRIPTIONS (CONTINUED)

FOOTNOTES:

$$^a LL = \sum N_i W_i,$$

where,

$$W_i = \left(\frac{3}{4^Q - 1} \right) (4^{i-1}), \text{ such that } \sum_{i=1}^Q W_i = 1$$

N_i = number of loops in program at the i th level of nesting

W_i = a weighting factor

Q = maximum level of nesting used in the system

4 = a shaping value

$$^b IF = \sum N_i W_i,$$

where,

N_i = number of "IF's" in program at the i th level of nesting

W_i = a weighting factor, the same as indicated for loop complexity measure

^c All normalized variable values were computed by multiplying the respective unnormalized values by 100/TS.

APPENDIX C

(This Appendix contains the list of predictor variables used and eliminated (a priori) when predicting errors and error rate for both data samples, as discussed in Section 5.6).

TABLE C-1. SAMPLE S, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS/ PROGRAM & ERROR RATE/PROGRAM

SAMPLE S VARIABLES		PROJECTS		
		M	B	P
X 1	SOURCE INSTRUCTIONS	3		
X 2	ENTRY POINTS			
X 3	EXIT POINTS			
X 4	USING INSTRUCTIONS			
X 5	COMMENT STATEMENTS			
X 6	LABELED INSTRUCTIONS			
X 7	ARITHMETIC INSTRUCTIONS			
X 8	UNCONDITIONAL JUMPS			
X 9	CALLS/LINKS			
X10	SYSTEM MACROS			
X11	USER MACROS			
X12	EQUATE STATEMENTS			
X13	COMMENTED INSTRUCTIONS			
X14	LOGICAL CONNECTORS			
X15	CONDITIONAL JUMPS			
X16	FUNCTIONS			
X17	SCALING/ROUNDING OPERATIONS			
X18	SHORT DO LOOPS			
X19	NESTED SHORT DO LOOPS			
X20	LOCK MACROS		1	
X21	INSTRUCTIONS IN SHORT DO LOOPS		1	
X22	ADDRESS VARIABLES			
X23	ADDRESS VARIABLE FREQUENCY			
X24	BINARY VARIABLE FREQUENCY	1	1	1
X25	CHARACTER VARIABLES	1	1	1
X26	CHARACTER VARIABLE FREQUENCY	1	1	1
X27	FIXED-POINT VARIABLES			
X28	FIXED-POINT VARIABLE FREQUENCY			
X29	FLOATING-POINT VARIABLES			
X30	FLOATING-POINT VARIABLE FREQUENCY			
X31	HEXADECIMAL VARIABLES	1	1	1
X32	HEXADECIMAL VARIABLE FREQUENCY	1	1	1
X33	LABELED-ARRAY VARIABLES			
X34	LABELED-ARRAY VARIABLE FREQUENCY	2	2	2
X35	REGISTER VARIABLES			
X36	REGISTER VARIABLE FREQUENCY			
X37	UNDEFINED VARIABLES			
X38	UNDEFINED VARIABLE FREQUENCY			
X39	TOTAL VARIABLES	3	3	3
X40	TOTAL VARIABLE FREQUENCY	3	3	3

TABLE C-1. SAMPLE S, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS/PROGRAM & ERROR RATE/PROGRAM (CONTINUED)

SAMPLE S VARIABLES		PROJECTS		
		M	B	P
X41	TOTAL DO LOOPS	3	3	3
X42	NON-NESTED DO LOOPS			
X43	DO LOOPS NESTED AT 2ND LEVEL			
X44	DO LOOPS NESTED AT 3RD LEVEL			
X45	DO LOOPS NESTED AT 4TH LEVEL			
X46	DO LOOPS NESTED AT 5TH LEVEL			
X47	DO LOOPS NESTED AT 6TH LEVEL OR LOWER			
X48	INSTRUCTIONS IN NON NESTED DO LOOPS			
X49	INSTRUCTIONS IN 2ND LEVEL DO LOOPS			
X50	INSTRUCTIONS IN 3RD LEVEL DO LOOPS			
X51	INSTRUCTIONS IN 4TH LEVEL DO LOOPS			
X52	INSTRUCTIONS IN 5TH LEVEL DO LOOPS			
X53	INSTRUCTIONS IN 6TH LEVEL OR LOWER DO LOOPS			
X54	SOURCE INSTRUCTIONS X AVERAGE NUMBER OF OPERATORS/ARITHMETIC INSTRUCTION			
X55	NO. OF PROGRAMMING ERRORS FOUND DURING THE TEST & INTEGRATION (T&I) PHASE	DEPENDENT VARIABLE ERRORS/PROGRAM		
X56	ENTRY POINTS/X1			
X57	EXIT POINTS/X1			
X58	USING INSTRUCTIONS/X1			
X59	COMMENT STATEMENTS/X1			
X60	LABELED INSTRUCTIONS/X1			
X61	ARITHMETIC INSTRUCTIONS/X1			
X62	UNCONDITIONAL JUMPS/X1			
X63	CALLS/LINKS/X1			
X64	SYSTEM MACROS/X1			
X65	USER MACROS/X1			
X66	EQUATE STATEMENTS/X1			
X67	COMMENTED INSTRUCTIONS/X1			
X68	LOGICAL CONNECTORS/X1			
X69	CONDITIONAL JUMPS/X1			
X70	FUNCTIONS/X1			
X71	SCALING/ROUNDING OPERATIONS/X1			
X72	SHORT DO LOOPS/X1			
X73	NESTED SHORT DO LOOPS/X1			
X74	LOCK MACROS/X1		1	
X75	INSTRUCTIONS IN SHORT DO LOOPS/X1		1	
X76	ADDRESS VARIABLES/X1			
X77	ADDRESS VARIABLE FREQUENCY/X1			

TABLE C-1. SAMPLE S, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS/ PROGRAM & ERROR RATE/PROGRAM (CONTINUED)

SAMPLE S VARIABLES	PROJECTS		
	M	B	P
X78 BINARY VARIABLE FREQUENCY/X1	1	1	1
X79 CHARACTER VARIABLES/X1	1	1	1
X80 CHARACTER VARIABLE FREQUENCY/X1	1	1	1
X81 FIXED-POINT VARIABLES/X1			
X82 FIXED-POINT VARIABLE FREQUENCY/X1			
X83 FLOATING-POINT VARIABLES/X1			
X84 FLOATING-POINT VARIABLE FREQUENCY/X1			
X85 HEXADECIMAL VARIABLES/X1	1	1	1
X86 HEXADECIMAL VARIABLE FREQUENCY/X1	1	1	1
X87 LABELED-ARRAY VARIABLES/X1			
X88 LABELED-ARRAY VARIABLE FREQUENCY/X1	2	2	2
X89 REGISTER VARIABLES/X1			
X90 REGISTER VARIABLE FREQUENCY/X1			
X91 UNDEFINED VARIABLES/X1			
X92 UNDEFINED VARIABLE FREQUENCY/X1			
X93 TOTAL VARIABLES/X1	3	3	3
X94 TOTAL VARIABLE FREQUENCY/X1	3	3	3
X95 TOTAL DO LOOPS/X1	3	3	3
X96 NON-NESTED DO LOOPS/X1			
X97 DO LOOPS NESTED AT 2ND LEVEL/X1			
X98 DO LOOPS NESTED AT 3RD LEVEL/X1			
X99 DO LOOPS NESTED AT 4TH LEVEL/X1			
X100 DO LOOPS NESTED AT 5TH LEVEL/X1			
X101 DO LOOPS NESTED AT 6TH LEVEL OR LOWER/X1			
X102 INSTRUCTIONS IN NON NESTED DO LOOPS/X1			
X103 INSTRUCTIONS IN 2ND LEVEL DO LOOPS/X1			
X104 INSTRUCTIONS IN 3RD LEVEL DO LOOPS/X1			
X105 INSTRUCTIONS IN 4TH LEVEL DO LOOPS/X1			
X106 INSTRUCTIONS IN 5TH LEVEL DO LOOPS/X1			
X107 INSTRUCTIONS IN 6TH LEVEL OR LOWER DO LOOPS/X1			
X108 SOURCE INSTRUCTIONS X AVERAGE NUMBER OPERATORS/ARITH. INSTRUCTION/X1			
X109 (NO. OF PROGRAMMING ERRORS FOUND DURING T&I PHASE)/X1			
	DEPENDENT VARIABLE, ERROR RATE/PGM.		

TABLE C-1. SAMPLE S, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS/PROGRAM & ERROR RATE/PROGRAM

FOOTNOTES

- * THE EMPTY CELLS IN THIS TABLE ARE USED TO IDENTIFY THOSE PREDICTOR VARIABLES THAT WERE MADE AVAILABLE FOR AUTOMATIC SELECTION BY THE STEPWISE REGRESSION PROCEDURE FOR ENTRY INTO THE PREDICTION EQUATION.

A PRIORI VARIABLE ELIMINATION CRITERIA

- 1 THE VARIABLES' DATA VALUES WERE ALL ZERO IN THE SAMPLE. (APPARENTLY, THE VARIABLES COULD HAVE BEEN UNAVAILABLE, NOT COLLECTED OR COUNTED DURING THE PROGRAM SCANNING OPERATION, NON-EXISTENT, OR NOT APPLICABLE IN THE PROGRAMS THAT WERE USED FOR THIS ANALYSIS).
- 2 THE VARIABLE WAS HIGHLY CORRELATED WITH ANOTHER PREDICTOR VARIABLE. (NOTE: THE CORRELATION COEFFICIENTS, $R_{X33,X34}$ AND $R_{X87,X88}$, WERE FOUND TO BE 1.00 IN EACH PROJECT SAMPLE; THEREFORE ONLY ONE VARIABLE FROM EACH PAIR, X33 AND X87 RESPECTIVELY, WERE MADE AVAILABLE FOR SELECTION IN THE STEPWISE REGRESSION PROCEDURE).
- 3 THE VARIABLE IS A LINEAR COMBINATION OF OTHER PREDICTOR VARIABLES. NOTE:

$$\begin{aligned}
 X39 &= X22 + X25 + X27 + X29 + X31 + X33 + X35 + X37 \\
 X40 &= X23 + X24 + X26 + X28 + X30 + X32 + X34 + X36 + X38 \\
 X41 &= X18 + X19 + X42 + X43 + X44 + X45 + X46 + X47 \\
 X93 &= X76 + X79 + X81 + X83 + X85 + X87 + X89 + X91 \\
 X94 &= X77 + X78 + X80 + X82 + X84 + X86 + X88 + X90 + X92 \\
 X95 &= X72 + X73 + X96 + X97 + X98 + X99 + X100 + X101
 \end{aligned}$$

TABLE C-2. SAMPLE T, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS/ PROGRAM

PROGRAM STRUCTURE + PROGRAMMER VARIABLES							PROGRAM STRUCTURE VARIABLES ONLY							
VARIABLE	SUBSYSTEM						SUBSYSTEM							
	A	B	C	D	E	F	A	B	C	D	E	F	G	H
1. TS	**													
2. LL														
3. IF														
4. BR	1	1	1	1	1	1	1	1	1	1	1	1		1
5. LS	1		1		1	1	1		1		1	1	1	1
6. AP														
7. SYS														
8. I/O														
9. COMP														
10. DATA		1		1				1		1			1	1
11. NEX	2	2	2	2	2	2	2	2	2	2	2	2	2	2
12. EX	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13. TI	2	2	2	2	2	2	2	2	2	2	2	2	2	2
14. COM														
15. RAT							3	3	3	3	3	3	3	3
16. WKLD							3	3	3	3	3	3	3	3
17. RAT/ WKLD							3	3	3	3	3	3	3	3
18. LL/TS														
19. IF/TS														
20. BR/TS														
21. LS/TS														
22. AP/TS														
23. SYS/TS														
24. IO/TS														
25. COMP/TS														
26. DATA/TS														
27. NEX/TS	2	2	2	2	2	2	2	2	2	2	2	2	2	2
28. EX/TS														
29. TI/TS	2	2	2	2	2	2	2	2	2	2	2	2	2	2
30. COM/TS														
TOTAL	23	23	23	23	23	23	23	23	23	23	23	23	23	23
PREDIC- TORS USED														

TABLE C-2. SAMPLE T, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERRORS/ PROGRAM (CONTINUED)

"THE EMPTY CELLS IN THIS TABLE ARE USED TO IDENTIFY THOSE PREDICTOR VARIABLES THAT WERE MADE AVAILABLE FOR AUTOMATIC SELECTION BY THE STEPWISE REGRESSION PROCEDURE FOR ENTRY INTO THE PREDICTOR EQUATION.

A PRIORI VARIABLE ELIMINATION CRITERIA

¹THE VARIABLE WAS HIGHLY CORRELATED WITH ANOTHER PREDICTOR VARIABLE.

²THE VARIABLE IS A LINEAR COMBINATION OF OTHER PREDICTOR VARIABLES.

$$\begin{aligned} \text{NEX} &= \text{TS-EX} \\ \text{TI} &= \text{AP+SYS} \\ \text{NEX/TS} &= 1-\text{EX/TS} \\ \text{TI/TS} &= \text{AP/TS+SYS/TS} \end{aligned}$$

³THE VARIABLE WAS NOT APPLICABLE FOR THIS SET OF COMPUTER RUNS.

(NOTE - ONE OF THE FOUR PREDICTORS (BR, LS, DATA, EX) HAVING THE HIGHEST CORRELATION WITH ERRORS, AND THE TS VARIABLE, WERE BOTH MADE AVAILABLE FOR SELECTION. EACH PREDICTOR THEN COULD BE SWAPPED FOR THE OTHER (SINCE ALL FIVE PREDICTORS WERE HIGHLY CORRELATED) USING THE FSWAP SELECTION ALGORITHM).

TABLE C-3. SAMPLE T, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERROR RATE/PROGRAM

PROGRAM STRUCTURE + PROGRAMMER VARIABLES							PROGRAM STRUCTURE VARIABLES ONLY							
VARIABLE	SUBSYSTEM						SUBSYSTEM							
	A	B	C	D	E	F	A	B	C	D	E	F	G	H
1. TS	**													
2. LL														
3. IF														
4. BR	1	1	1	1	1		1	1	1	1	1			1
5. LS					1	1					1	1	1	1
6. AP														
7. SYS														
8. I/O														
9. COMP														
10. DATA	1	1	1	1	1	1	1	1	1	1	1	1		
11. NEX	2	2	2	2	2	2	2	2	2	2	2	2		
12. EX	1	1	1	1		1	1	1	1		1	1	1	
13. TI	2	2	2	2	2	2	2	2	2	2	2	2	2	2
14. COM														
15. RAT							3	3	3	3	3	3	3	3
16. WKLD							3	3	3	3	3	3	3	3
17. RAT/ WKLD							3	3	3	3	3	3	3	3
18. LL/TS														
19. IF/TS														
20. BR/TS														
21. LS/TS														
22. AP/TS														
23. SYS/TS														
24. IO/TS														
25. COMP/TS														
26. DATA/TS														
27. NEX/TS	2	2	2	2	2	2	2	2	2	2	2	2	2	2
28. EX/TS														
29. TI/TS	2	2	2	2	2	2	2	2	2	2	2	2	2	2
30. COM/TS														
TOTAL	23	23	23	23	23	23	20	20	20	20	20	20	20	20
PREDIC- TORS USED														

TABLE C-3. SAMPLE T, LIST OF PREDICTOR VARIABLES USED AND ELIMINATED (A PRIORI) WHEN PREDICTING ERROR RATE/PROGRAM (CONTINUED)

"THE EMPTY CELLS IN THIS TABLE ARE USED TO IDENTIFY THOSE PREDICTOR VARIABLES THAT WERE MADE AVAILABLE FOR AUTOMATIC SELECTION BY THE STEPWISE REGRESSION PROCEDURE FOR ENTRY INTO THE PREDICTION EQUATION.

A PRIORI VARIABLE ELIMINATION CRITERIA

¹THE VARIABLE WAS HIGHLY CORRELATED WITH ANOTHER PREDICTOR VARIABLE.

²THE VARIABLE IS A LINEAR COMBINATION OF OTHER PREDICTOR VARIABLES.

$$\begin{aligned} \text{NEX} &= \text{TS} - \text{EX} \\ \text{TI} &= \text{AP} + \text{SYS} \\ \text{NEX/TS} &= 1 - \text{EX/TS} \\ \text{TI/TS} &= \text{AP/TS} + \text{SYS/TS} \end{aligned}$$

³THE VARIABLE WAS NOT APPLICABLE FOR THIS SET OF COMPUTER RUNS.

(NOTE - ONE OF THE FOUR PREDICTORS (BR, LS, DATA, EX) HAVING THE HIGHEST CORRELATION WITH ERROR RATE, AND THE TS VARIABLE, WERE BOTH MADE AVAILABLE FOR SELECTION. EACH PREDICTOR THEN COULD BE SWAPPED FOR THE OTHER (SINCE ALL FIVE PREDICTORS WERE HIGHLY CORRELATED) USING THE FSWAP ALGORITHM).

METRIC SYSTEM

BASE UNITS:		Unit	SI Symbol	Formula
Quantity				
length	metre	m	...	
mass	kilogram	kg	...	
time	second	s	...	
electric current	ampere	A	...	
thermodynamic temperature	kelvin	K	...	
amount of substance	mole	mol	...	
luminous intensity	candela	cd	...	

SUPPLEMENTARY UNITS:		Unit	SI Symbol	Formula
plane angle	radian	rad	...	
solid angle	steradian	sr	...	

DERIVED UNITS:		Unit	SI Symbol	Formula
Acceleration	metre per second squared	m/s ²	m/s	
activity (of a radioactive source)	disintegration per second	(disintegration)/s	(disintegration)/s	
angular acceleration	radian per second squared	rad/s ²	rad/s	
angular velocity	radian per second	rad/s	rad/s	
area	square metre	m ²	m	
density	kilogram per cubic metre	kg/m ³	kg/m	
electric capacitance	farad	F	A·s/V	
electrical conductance	siemens	S	A/V	
electric field strength	volt per metre	V/m	V/m	
electric inductance	henry	H	V·s/A	
electric potential difference	volt	V	W/A	
electric resistance	ohm	Ω	V/A	
electromotive force	volt	V	W/A	
energy	joule	J	N·m	
entropy	joule per kelvin	J/K	J/K	
force	newton	N	kg·m/s ²	
frequency	hertz	Hz	(cycle)/s	
illuminance	lux	lx	lm/m ²	
luminance	candela per square metre	cd/m ²	cd/m	
luminous flux	lumen	lm	cd·sr	
magnetic field strength	ampere per metre	A/m	A/m	
magnetic flux	weber	Wb	V·s	
magnetic flux density	tesla	T	Wb/m ²	
magnetomotive force	ampere	A	...	
power	watt	W	J/s	
pressure	pascal	Pa	N/m ²	
quantity of electricity	coulomb	C	A·s	
quantity of heat	joule	J	N·m	
radiant intensity	watt per steradian	W/sr	W/sr	
specific heat	joule per kilogram-kelvin	J/kg·K	J/kg·K	
stress	pascal	Pa	N/m ²	
thermal conductivity	watt per metre-kelvin	W/m·K	W/m·K	
velocity	metre per second	m/s	m/s	
viscosity, dynamic	pascal-second	Pa·s	Pa·s	
viscosity, kinematic	square metre per second	m ² /s	m ² /s	
voltage	volt	V	W/A	
volume	cubic metre	m ³	m	
wavenumber	reciprocal metre	1/m	(wave)/m	
work	joule	J	N·m	

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 - 10 ¹²	tera	T
1 000 000 000 - 10 ⁹	giga	G
1 000 000 - 10 ⁶	mega	M
1 000 - 10 ³	kilo	k
100 - 10 ²	hecto*	h
10 - 10 ¹	deka*	da
1 - 10 ⁰	deci*	d
0.1 - 10 ⁻¹	centi*	c
0.01 - 10 ⁻²	milli	m
0.001 - 10 ⁻³	micro	μ
0.000 001 - 10 ⁻⁶	nano	n
0.000 000 001 - 10 ⁻⁹	pico	p
0.000 000 000 001 - 10 ⁻¹²	femto	f
0.000 000 000 000 001 - 10 ⁻¹⁵	atto	a

* To be avoided where possible

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C³) activities, and in the C³ areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

