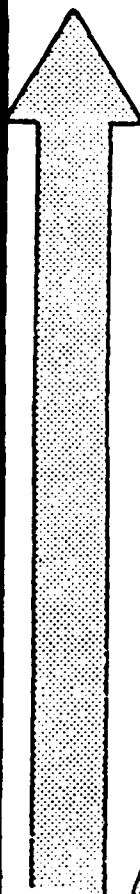


①

AD A 131757

DACS DATA COMPENDIUM SERIES
NASA/SEL SOFTWARE DEVELOPMENT DATA
APRIL 1981



D
A
C
S

DTIC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC
ELECTE
AUG 21 1983

E

Data & Analysis Center for Software

83 07 12 037

The information and data contained herein have been compiled from government and nongovernment technical reports and are intended to be used for reference purposes. Neither the United States Government nor IIT Research Institute warrant the accuracy of this information and data. The user is further cautioned that the data contained herein may not be used in lieu of other contractually cited references and specifications.

Publication of this information is not an expression of the opinion of The United States Government or of IIT Research Institute as to the quality or durability of any product mentioned herein and any use for advertising or promotional purposes of this information in conjunction with the name of The United States Government or IIT Research Institute without written permission is expressly prohibited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DC-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DACS DATA COMPENDIUM SERIES NASA/SEL SOFTWARE DEVELOPMENT DATA		5. TYPE OF REPORT & PERIOD COVERED Interim Report Nov. 80 - April 81
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Christopher Turner Gary Caron Gineen Brement		8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0255
9. PERFORMING ORGANIZATION NAME AND ADDRESS Data & Analysis Center for Software RADC/ISIS Griffiss AFB, NY 13441		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (COEE) Griffiss AFB, NY 13441		12. REPORT DATE April 1981
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		13. NUMBER OF PAGES 109
		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		
18. SUPPLEMENTARY NOTES RADC Project Engineer: John Palaimo (COEE) DACS Source Code No. 413570 Cost: \$10.00 Available from: Data & Analysis Center for Software RADC/ISIS Griffiss AFB, NY 13441		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Software Engineering Software Experience Data Computer Software Data Collection Data Repository		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This is one of a planned series of data publications dealing with software development data and its use to predict software cost and reliability in future software design, development, testing and maintenance efforts. This DACS compendium provides specific information on 29 software development projects monitored by NASA/SEL during the 1976-1977 time frame. This information includes: descriptions of data collection techniques, statistical and graphical data summaries, analysis of data quality and completeness and a discussion of the potential application of the data in statistical and modelling studies. This Data		

~~CONFIDENTIAL~~
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Compendium does not include exhaustive listings of dataset contents; these kinds of listings are available from the DACS in hard copy or machine readable format.

ii
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

DACS

Data & Analysis Center for Software
AN INFORMATION ANALYSIS CENTER

.....

DACS DATA COMPENDIUM SERIES
NASA/SEL SOFTWARE DEVELOPMENT DATA

APRIL 1981

BY:
CHRISTOPHER TURNER
GARY CARON
GINEEN BREMENT

UNDER CONTRACT TO:
ROME AIR DEVELOPMENT CENTER
GRIFFISS AFB, NY 13441

Acquisition	
FILE	X
#10.00 RADC/ST-1	
A 21	



ORDERING REFERENCE: DC-1

DACS

THE Data & Analysis Center for Software
IS AN INFORMATION ANALYSIS CENTER, OPERATED BY IIT RESEARCH INSTITUTE
UNDER CONTRACT TO THE ROME AIR DEVELOPMENT CENTER, AFSC.

The Data & Analysis Center for Software (DACS) is an information analysis center sponsored by the Air Force Systems Command, Rome Air Development Center (RADC), and operated by IIT Research Institute (IITRI). DACS serves as a central source for current, readily usable data and information concerning software technology.

The major functions of the DACS are: to develop and maintain a computer database of empirical data collected on the development and maintenance of computer software; produce and distribute subsets of the database; maintain a software technology information base of technical documents, project status information, and evaluation data; analyze the data and information and produce technical reports; maintain a current awareness program which includes dissemination of technical information (analysis reports, technical monographs, etc.), assessments of technological developments, and publication of a quarterly newsletter; develop and maintain a glossary of software engineering terms; and provide rapid response to inquiries for technical information and assistance.

To obtain more information on the products and services of the DACS, contact:

Data & Analysis Center for Software
RADC/ISISI
Griffiss Air Force Base, NY 13441

Telephone: 315/336-0937
Autovon: 587-3395

PREFACE

This is one of a series of data publications dealing with software development data. This publication provides a summary of software development experience data collected by the NASA Software Engineering Laboratory (NASA/SEL) at Goddard Space Flight Center during the 1976-1980 time frame. Other publications in this series will describe other data sets of the DACS database.

Graphical summaries are provided only for those projects which contain the most complete data. Tabular summaries are provided, however, for all projects. The original NASA/SEL database refers to 42 projects but not all are included in this publication due to the incompleteness of the data for some projects. Some projects were well underway before data collection was initiated and others are still undergoing development.

This document provides a summary of the results of the data collection effort. No attempt is made to analyze the data for quality or completeness. This compendium does not include exhaustive listings of data set contents; these kinds of listings are available from the Data & Analysis Center for Software (DACS)--in hard copy or machine readable form--upon request.

This graphical and tabular data may prove to be useful to software engineers and researchers in a number of ways. It provides software project development historical data which may be useful for studies of software cost estimation, project monitoring and software quality. It may be useful in determining the relative usefulness of Modern Programming Practices in the software development and maintenance processes. Finally, it provides a source of data which can be used to develop and validate cost and reliability prediction models across a variety of projects, environments, applications, etc.

The tabular data from which this compendium is derived was printed directly from the DACS computerized database utilizing customized retrieval and report generation software developed by the DACS programming staff. This system allows the generation of special reports wherein the data is categorized to match the needs of the user.

The user is cautioned that the data produced in this publication reflects the results of the data collection process only; it does not necessarily reflect the actual project development environment.

ACKNOWLEDGMENTS

The DACS is grateful to the Software Development Department of the NASA/SEL at Goddard Space Flight Center for providing the data summarized in this publication. Without the generous management support of Mr. Frank McGarry at NASA/SEL, project history and software development methodology data would not have been included in this compendium. In addition, the DACS would like to express its gratitude to Mr. McGarry for his helpful suggestions for improving the presentation of the data and the structure of this report.

The Rome Air Development Center's (RADC's) computer facilities and the extensive cooperation of the Information Sciences Division's Software Engineering Section (RADC/ISIS) and their personnel were a vital factor in the generation of this compendium.

Gineen Brement of the DACS staff deserves special mention for doing so well the great amount of tabular and graphical illustration, and the typing and online editing necessary to prepare this compendium.

TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION	1-1
1.1 Purpose of the Data Compendium	1-1
1.2 Contents of the Data Compendium	1-1
1.3 Data Collection Approach	1-2
1.4 Data Sets	1-3
2. PROJECT DATA SUMMARIES	2-1
2.1 Project 1	2-9
2.2 Project 2	2-10
2.3 Project 3	2-16
2.4 Project 4	2-16
2.5 Project 5	2-17
2.6 Project 6	2-23
2.7 Project 7	2-29
2.8 Project 8	2-35
2.9 Project 9	2-41
2.10 Project 10	2-42
2.11 Project 11	2-48
2.12 Project 12	2-48
2.13 Project 13	2-48
2.14 Project 14	2-48
2.15 Project 15	2-49
2.16 Project 16	2-55
2.17 Project 17	2-55
2.18 Project 18	2-56
2.19 Project 19	2-57
2.20 Project 20	2-63
2.21 Project 21	2-64
2.22 Project 22	2-70
2.23 Project 23	2-70
2.24 Project 24	2-70
2.25 Project 25	2-70
2.26 Project 26	2-71
2.27 Project 27	2-77

TABLE OF CONTENTS (con'td)

	<u>Page</u>
2.28 Project 28	2-77
2.29 Project 29	2-77
2.30 Project 30	2-77
2.31 Project 31	2-77
2.32 Project 32	2-78
2.33 Project 33	2-78
2.34 Project 34	2-78
2.35 Project 35	2-79
2.36 Project 36	2-85
2.37 Project 37	2-85
2.38 Project 38	2-85
2.39 Project 39	2-86
2.40 Project 40	2-92
3. CROSS PROJECT SUMMARIES	3-1
3.1 Project History and Development Data	3-1
3.2 Change Report Data	3-2
3.3 Development Methodology Data	3-6
REFERENCES	R-1
APPENDIX A - Glossary of Terms	A-1

LIST OF TABLES

TABLE 3.1-1 GENERAL PROJECT DEVELOPMENT DATA TABLE	
PART I	3-3
PART II	3-4
TABLE 3.2-1 CHANGE IN ERROR DATA TABLE	3-5
TABLE 3.3-1 PROJECT DEVELOPMENT METHODS AND CONSTRAINTS DATA TABLE .	3-8

1. INTRODUCTION

1.1 Purpose of the Data Compendium

The purpose of this Data Compendium is to disseminate information on the software engineering database originally developed by the NASA/SEL at Goddard Space Flight Center. It has been produced by the DACS operated by IIT Research Institute (IITRI) under contract to RADC. It is to serve as a software engineering reference on historical software development data. The NASA/SEL data, in addition to other data, is maintained in the DACS software experience database for analysis and model validation purposes. Individuals who are conducting software engineering research may obtain subsets of the NASA/SEL data from the DACS in hardcopy and/or machine readable form.

1.2 Contents of the Data Compendium

The remainder of this section briefly describes NASA/SEL's approach to data collection and outlines the categories of data summarized in this publication.

Section 2 consists of detailed summaries of the data, in graphical form, for the projects which contained the most complete data. An overview of the development environment, however, is given for all projects. This overview was developed by the DACS through personal interviews and examination of General Project Summary Forms provided by NASA/SEL.

Section 3 consists of a set of three tables of summary data and associated narrative describing the three main categories of data available across all projects. These data categories are:

- Project Development Data
- Change Error Data
- Development Methodology Data

Each of these three categories of data is characterized by its

association to the problem, the people, the process, the product, the resources, and the tools. Some factors may fit more than one category but are listed only once.

Appendix A provides more in-depth information about the terminology used in this Data Compendium. The definitions were either provided by NASA/SEL or extracted from the DACS Glossary (1).

1.3 Data Collection Approach

The NASA/SEL was founded in 1976. The primary purpose of the NASA/SEL is to study the impact of various management and programming methods on software development. The approach of the NASA/SEL has been to monitor and measure the software life cycle during actual systems development and then to subject these measures to quantitative analysis. One significant result of these efforts is the software engineering database summarized in this report. The database consists of software experience data collected on NASA/SEL software development projects through the use of seven forms completed periodically by project personnel.

- (1) The General Project Summary Form was used for project classification and progress evaluation. It was filled out by the project manager at the beginning of the project, at the completion of each major milestone and at project end.
- (2) The Programmer/Analyst Survey Form was used to track the components within a system. A component, in this context, is a processing module identified by its function or a named common block of shared data. The form was completed for each system component when it was defined, when it was implemented and whenever a major modification was made.
- (3) The Component Summary Form was used to keep track of modules, subroutines, block common, etc. of the system. It was filled out when the component was defined, when it was completed and when major modifications were made.
- (4) The Component Status Report was used to track resource expenditures by component. The form was completed weekly by each team member working on the project.

- (5) The Resource Summary Form was used to track project costs. It was completed weekly by the project manager.
- (6) The Change Report Form was used to evaluate the impact of system changes on the development cycle. It was completed every time the system was changed due to modifications or discovered errors in specification, design or code.
- (7) The Computer Program Run Analysis Form was used to monitor computing activity during systems development. An entry on the form was made every time a computer run was initiated.

Detailed instructions for completing these forms are given in reference (2).

This data has been made available on tape to the DACS for the purpose of comparative analysis. The project described in this publication are primarily in the area of ground support software for satellite attitude control programs. The principle programming languages used in these projects were FORTRAN and Assembly, and the main development computers were an IBM 360 and a PDP 11. The projects range in size from 12 to 600 modules and each consist of from 1200 to 110000 delivered lines of source code.

The data is not complete for all projects because some were well underway before the data collection effort was initiated and some projects are still undergoing development.

1.4 Data Sets

The following subsets of the NASA/SEL data have been extracted from the NASA/SEL database and put into graphical and tabular form:

General Project Information

Delivered Source Lines of Code

New Lines of Delivered Code

Number of Components

Number of Modules

Number of New Modules

Number of Pages of Documentation

Months of Development Time

Manmonths of Development Effort

Number of Computer Runs

Number of Computer Hours

Number of Program Changes

Project Scheduling Information

Design Start and End Dates

Code and Test Start and End Dates

System Test Start and End Dates

Acceptance Test Start and End Dates

Cleanup Start and End Dates

Code Production History

End of Week Date

Cumulative Number of Source Lines Developed

Cumulative Number of Components Developed

Cumulative Number of Code Changes

Development Effort by Activity, Manhours

Design: Create; Read; Review

Development: Code; Read; Review

Testing: Module; Integration; Review

Profile of Run Purposes and Results

Profile of Types of Changes and Errors

Chronological Failure Data

Failure Number

Date Detected

Date Corrected

Type of Failure or Change

Origin of Failure

Number of Modules Affected

Calendar Days Since Last Failure/Change

2. PROJECT DATA SUMMARIES

This section presents detailed project summary data. The data is presented as a set of graphs. Each subsection is organized by project code into a set of narrative, tables and graphs. The project codes are derived from the NASA/SEL database. They have been retained to maintain consistency. For example, Project K is described in subsection 2.K in the following manner:

Title	Figure #
General Project Summary (Text)	
General Project Information (Table)	2.K-1
Project Scheduling Data (Gantt)	2.K-2
History of Documented Source Code Development	2.K-3
History of Module Development	2.K-4
History of Changes	2.K-5
Distribution of Development Effort by Task	2.K-6
Distribution of Computer Runs by Purpose	2.K-7
Distribution of Computer Runs by Results	2.K-8
Distribution of Changes by Type	2.K-9
Distribution of Errors by Type	2.K-10
Distribution of Effort Required to Isolate Errors	2.K-11
Distribution of Effort Required to Resolve Changes	2.K-12
Distribution of when Errors Entered the System by Phase	2.K-13

The following paragraphs provide an overview of the information represented by the figures. The General Project Summary is provided for 20 projects. Some detailed project development data isn't provided in this section because it is incomplete. The reader will note that some subsections of this section have been omitted or merely include text describing the

general project development methodology. These subsections will be completed as additional data becomes available to the DACS.

Subsections which contain only a general project summary are: 2.1, 2.3, 2.11, 2.16, 2.17, 2.18, 2.32 and 2.33. Subsections which have been omitted because there was little or no data in the database for the corresponding project code are 2.4, 2.9, 2.12, 2.13, 2.14, 2.20, 2.22-2.25, 2.27-2.31, 2.34, 2.36-2.38, and 2.40-2.42. Available data is, however, presented in tabular form in Section 3.

General Project Information (Figure 2.K-1)

The General Project Information Table describes project size in terms of lines of source and object code, number of components and pages of documentation. Project development time, effort, computer resources and the number of engineering change reports are also presented. Several items are not taken directly from the database, but rather calculated from data items stored in the database.

Months of development time is computed from the project development schedule and manmonths of development effort is computed using personnel hours expended during development and is based on 172 hours per one manmonth. The component sizing data is based on data recorded in the component information file. NASA's estimates for object code are calculated using the equations:

Object Code = $2.8 \times (\text{Source Code})$ for projects of less than or equal to 15000 lines of source code.

Object Code = $2.1 \times (\text{Source Code})$ for of greater than 15000 lines of source code.

Project Scheduling Information (Figure 2.K-2)

The second figure on the first page contains project scheduling information in the form of a Gantt Chart. The graph was produced directly

from data stored in the database. Actual start and end dates for each of the phases of project development are given in Section 3.

History of Documented Source Code Production (Figure 2.K-3)

The next graph presents a weekly history of code production. The graph represents the cumulative number of source lines, including comments produced during the given time period, and is taken directly from the project history file in the database.

History of Module Development (Figure 2.K-4)

This graph presents a weekly history of the cumulative number of modules produced during the given time period. The project history file was also used to produce this graph.

History of Changes (Figure 2.K-5)

This graph presents a weekly history of the cumulative number of changes made to the project during the given time period. Note that these are changes, not change reports. One or more changes are usually made for each change report which hasn't been rejected. Again, the project history file provided a direct source for the information contained in this graph.

The project history file contains complete data for eight projects. In some cases, the data on project code development and module development is incomplete because the number of lines of source code may not have been recorded or had comments added to it until a large portion of the project had been completed. This may account for some missing data.

Distribution of Development Effort by Task (Figure 2.K-6)

This figure represents the number of hours spent on each task in each of the first three phases of development: Design; Code and Unit Test; and Integration Test. These hours are calculated by summing the weekly reported manhours for each task as recorded in the database for each component. A pie

chart is used to represent the proportion of effort spent on each task during development. The total number of hours reported for these activities does not equal the total effort as recorded in the General Project Information Table because effort expended in Acceptance Testing and Cleanup was not included in this distribution. The tasks which are included in this distribution of effort are:

- Design Creation
- Design Read
- Design Review
- Code Development
- Code Reading
- Code Reviewing
- Module Testing
- Integration Testing
- Testing Review

Distributions of Computer Runs by Purposes and Results
(Figures 2.K-7 and 2.K-8)

The next two figures represent computer run data; the first displaying the purposes of runs made during development, the other displaying results of those runs. In each case, the proportional number of runs in each category is represented by a pie chart. The data for these two figures is derived directly from the database which records the purpose for, and results of, most runs. The total number of purposes or results may be greater or less than the number of computer runs recorded because some runs may have more than one purpose or result, or the purpose or result of a particular run may not have been recorded. The purpose of each computer run has been arranged according to the following categories:

- Unit Test

- System Test
- Benchmark Test
- Maintenance/Utility
- Compilation/Assembly/Link
- Debug Run
- Other

The number of run purposes for some projects may be greater than the number of runs reported because a single run may have more than one purpose; i.e., unit test and debug.

The results of a computer run have been arranged according to the following categories:

- Good Run
- Submit Error
- JCL Error
- Other Set-up Error
- Hardware Error
- Software Error
- Compile Error
- Link Error
- Execute Error
- User Generated Message
- Ran to Completion

The number of run results for some projects may be greater than the number of runs reported because a single run may have more than one result; i.e., a run ran to completion but included a user generated error message.

Distribution of Changes and Errors by Type (Figures 2.K-9 and 2.K-10)

These two graphs are derived from data recorded on Change Report Forms.

They summarize the types of changes and errors encountered during development. Data is recorded when an event occurs which triggers a change in the source code of a project. Since not all changes are a result of error correction, the number of changes may not necessarily equal the number of errors. Also, since not all changes are made to code (some are made to requirements and specifications) the number of change reports may not equal the number of changes as recorded in the General Project Information Table. This phenomenon may also be due to: 1) one change in source code resolving several change reports; or 2) changes not actually being made for every change report produced. The types of changes categorized in the database and the general categories under which each fall are:

- Corrective Changes
 - Error Correction
- Perfective Changes
 - Planned Enhancements
 - Improvement of Clarity/Maintainability/Documentation
 - Improvement of User Service
 - Optimization of Time/Space/Accuracy
- Adaptive Changes
 - Implementation of Requirements Change
 - Adaption to Environment Change
- Other Changes
 - Utility for Development Purposes Only
 - Other

Not all change reports categorized a change according to the preceding categories. No category of change was reported on many of the change reports.

When a change report does categorize the reason for a change as an error, the type of error is recorded as one or more of the following categories:

- Incorrect or Misinterpreted Requirements
- Incorrect or Misinterpreted Functional Specifications
- Error in the Design of Several Components
- Error in the Design of One Component
- Misunderstanding of the External Environment
- Error in Use of Programming Language/Compiler
- Clerical Error
- Other

Not all changes categorized as corrections had the error type(s) recorded. Also, some changes were made to correct more than one type of error.

Distributions of Effort Required to Isolate Errors and Resolve Changes
(Figures 2.K-11 and 2.K-12)

The next two figures categorize each error by the effort required to isolate it and categorize each change by the effort required to implement it. Pie charts are used in each case to visually represent the percentages within each effort category. Due to reasons similar to those mentioned above, the number of changes may not necessarily equal the number of change reports as recorded in the General Project Summary. Change effort was recorded according to one of the following four categories:

- Less than One Hour
- One Hour to One Day
- One Day to Three Days
- More than Three Days

Effort to make a change was not recorded for all change reports nor was

it recorded for all changes.

Distribution of When Errors Entered the System by Phase (Figure 2.K-13)

The final chart categorizes the errors by the development phase in which it was suspected that the error was introduced into the system. The possible development phases in which the error originated, either through omission or incorrect implementation are categorized as follows:

- Requirements Definition
- Functional Specifications
- Design
- Coding and Testing
- Other
- Can't Tell

The category "Can't Tell" implies that the phase in which the error was introduced could not be determined. It bears repeating that not all change reports had the source of the error recorded. Also, some may have had more than one reason recorded.

In summary, where data was available, it was included on the appropriate graph. Whenever a specific classification of a run or a change does not appear on a pie chart, then it can be understood that no runs or changes of this type were recorded. In some instances where percents are included in the pie charts, the total percent may be more or less than 100. This error is due to the rounding of fractional percentages to the nearest percent. In cases where data necessary to complete a given chart was not recorded at all, this status is indicated.

2.1 Project 1

Project 1 consists primarily of Assembly Level Code (ALC) as used on PDP 11/70 systems and was developed to support conversion of existing graphics support software for use on the PDP 11/70. Specification for the project was functional and procedural, using flowcharts and baseline diagrams (tree charts). Design was accomplished through iterative enhancement and the development was top-down, using structured code with simulated constructs (program stubs). Validation and verification testing was both top-down using stubs and bottom-up using drivers, and code-review was performed by the programmer's peers. One programmer and one librarian were employed during development with one additional person being used during the design phase.

The data on Project 1 consists primarily of change error data; other data for the most part was not recorded. As a result, no pie graphs could be developed for this project. Data which is available is summarized in the tables included in Section 3.

2.2 Project 2

Project 2 consists primarily of FORTRAN source code and was developed to support spacecraft orientation computation. The target computer system for this project was an IBM 360. The system was constructed with an overlay structure of 20 segments divided into two independent programs. The specifications for this project were functional at the subsystem level and design was accomplished using top-down techniques. Development was also top-down at the highest levels, where no specific coding techniques were used, with iterative enhancement being used to develop subroutines. Validation was accomplished by walk-throughs and formal testing procedures at the design level. The project personnel were organized into a team of six persons, with one chief programmer and one librarian.

The data on Project 2 is relatively complete, error types by category being the only data not recorded.

GENERAL PROJECT INFORMATION

PROJECT : 2

SIZE

DELIVERED LINES OF SOURCE CODE	50911
NEW LINES OF SOURCE CODE	45345
NASA ESTIMATE OF WORDS OF OBJECT CODE	106913
NASA ESTIMATE OF NEW OBJECT CODE	95227
NUMBER OF COMPONENTS	292
NUMBER OF MODULES	201
NUMBER OF NEW MODULES	172
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	195
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	1
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	2016

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	14.5
MANMONTHS OF DEVELOPMENT EFFORT	30
NUMBER OF COMPUTER RUNS	4604
SYSTEM 360-95 HOURS	222.3
SYSTEM 360-75 HOURS	160.3
POP 11-70 HOURS	0.3
NUMBER OF CHANGES	985
NUMBER OF CHANGE REPORTS	290

FIGURE 2.2-1

PROJECT 2

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

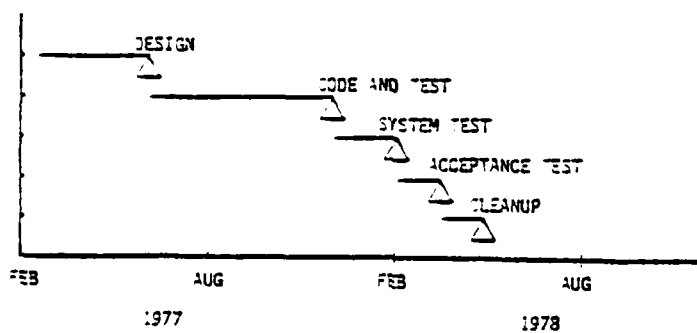


FIGURE 2.2-2

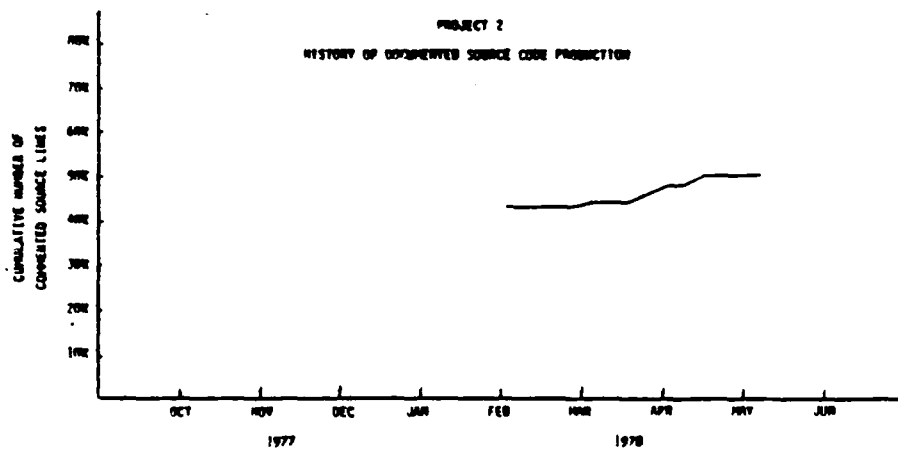


FIGURE 2.2-3

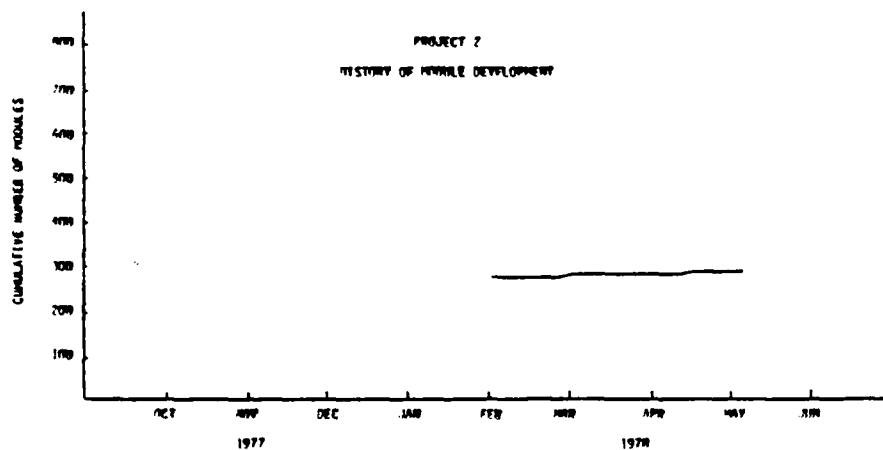


FIGURE 2.2-4

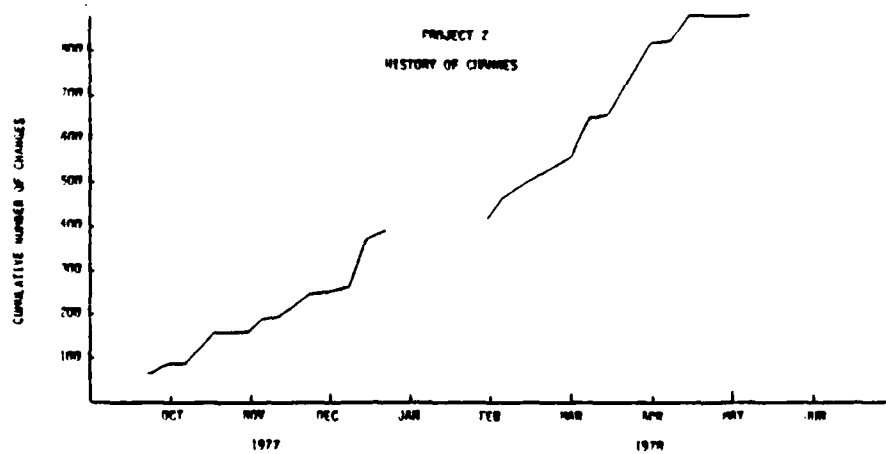


FIGURE 2.2-5

PROJECT: 2

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 8018.4

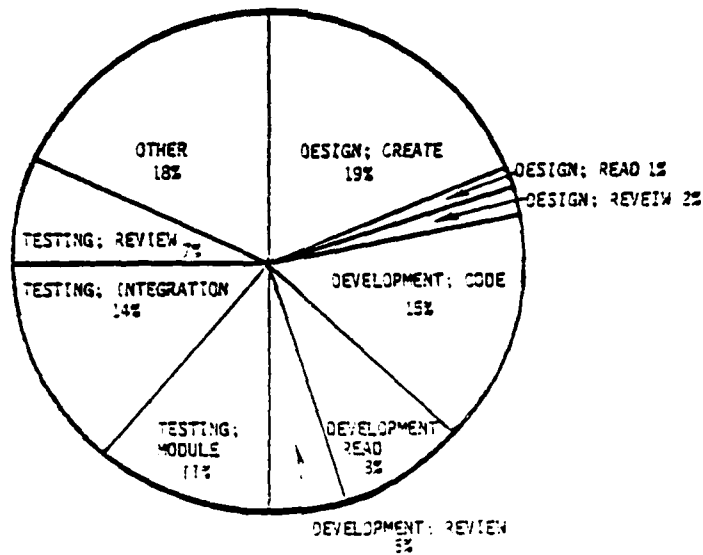


FIGURE 2.2-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS

TOTAL RUNS REPORTED: 1154

RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 1223

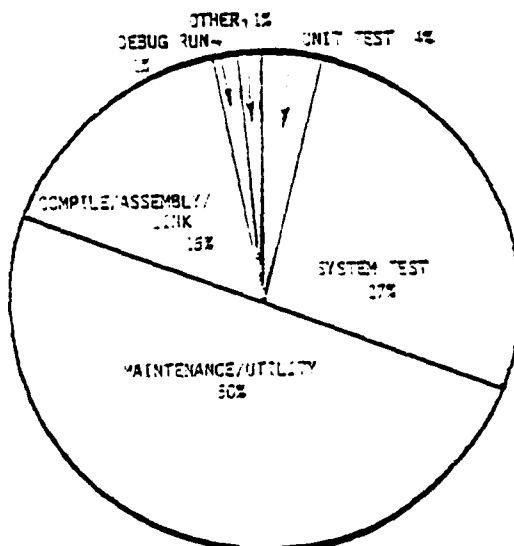


FIGURE 2.2-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS

TOTAL RUNS REPORTED: 1154

RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 47

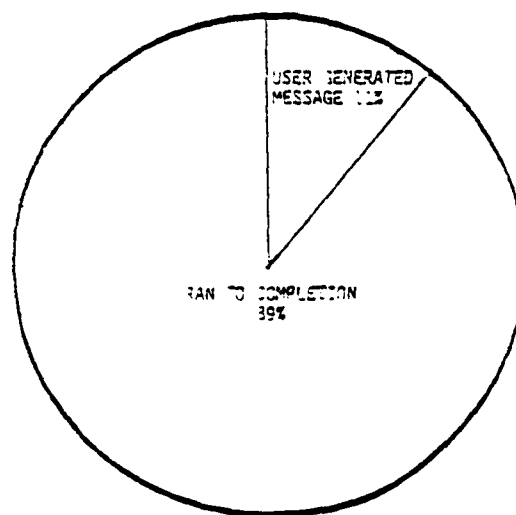


FIGURE 2.2-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 2: 290

No data recorded for
Type of Changes

No data recorded for
Type of Errors

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 144

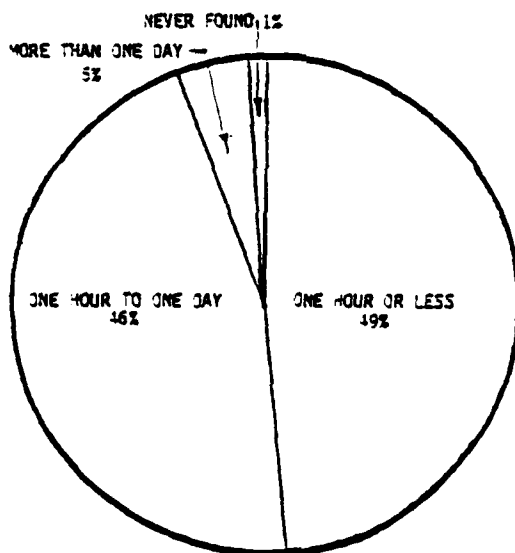


FIGURE 2.2-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 276

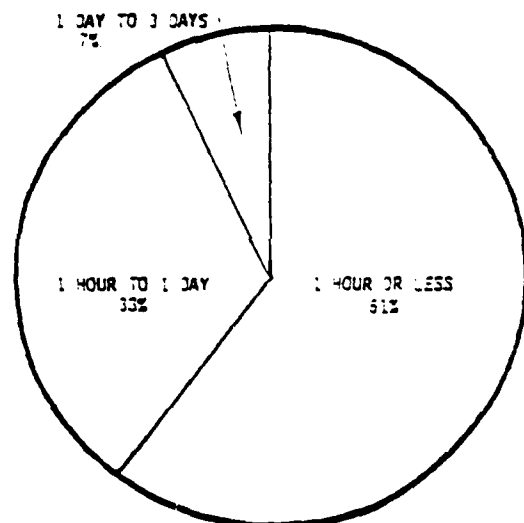


FIGURE 2.2-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 168

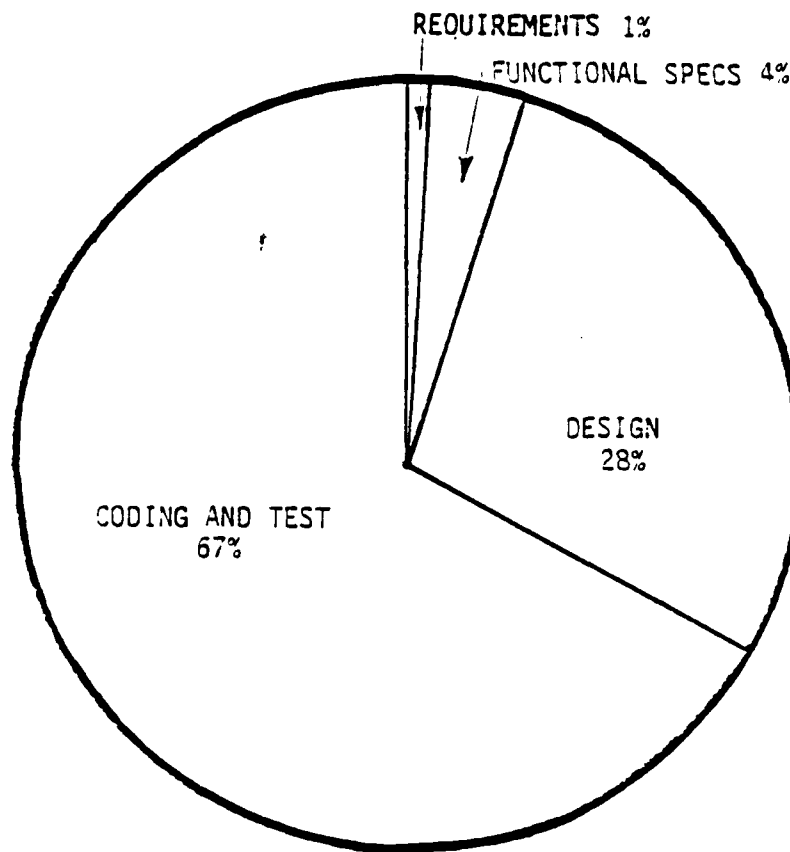


FIGURE 2.2-13

2.3 Project 3

Project 3 is an interactive program developed to assist management in resource allocation. The project was designed and developed to operate on a PDP 11/70. Specifications for the project were written in English text (non-formal) at the system level. Design and development were top-down at the project level with unstructured FORTRAN being used. Baseline diagrams (tree charts) and detailed system/module specifications were used during design and development, respectively. Validation testing was accomplished in a top-down manner using program stubs, and the quality of the code by review at the module level, and by walk-throughs at the system level. One programmer and keypuncher were involved in development.

Data on Project 3 consists of some component information computer run data, and a limited number of change reports.

2.5 Project 5

Project 5 consists primarily of FORTRAN source code. Its purpose was to compute spacecraft attitude based on telemetry data. The system consists of an overlay structure of nine segments. Specifications for this project were functional at the module level. The project was designed in a top-down fashion using program stubs. Baseline diagrams (tree charts) were used to specify the system design. The project was developed through iterative enhancement using program stubs with no specific coding standard required. Validation testing was top-down using program stubs, and inspection was accomplished by code reading and walk-throughs, as in the design and development of the system. The personnel were organized into a team structure consisting of a chief programmer, librarian, and from three to five assistant programmers.

No data was recorded in the Project History File to determine documented source code or module development history.

GENERAL PROJECT INFORMATION

PROJECT : 5

SIZE

DELIVERED LINES OF SOURCE CODE	55237
NEW LINES OF SOURCE CODE	43955
NASA ESTIMATE OF WORDS OF OBJECT CODE	115998
NASA ESTIMATE OF NEW OBJECT CODE	92311
NUMBER OF COMPONENTS	355
NUMBER OF MODULES	283
NUMBER OF NEW MODULES	200
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	88
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	2
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	1332

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	15.25
MANMONTHS OF DEVELOPMENT EFFORT	97
NUMBER OF COMPUTER RUNS	5871
SYSTEM 360-95 HOURS	164.3
SYSTEM 360-75 HOURS	156.3
POP 11-70 HOURS	0.3
NUMBER OF CHANGES	276
NUMBER OF CHANGE REPORTS	311

FIGURE 2.5-1

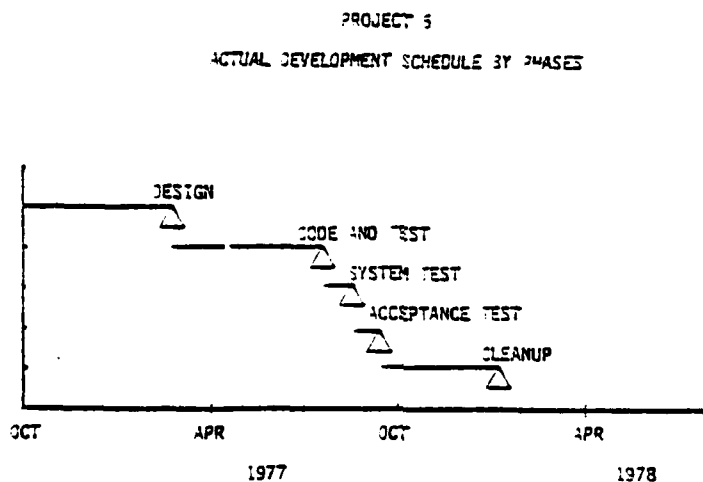


FIGURE 2.5-2

No data recorded for
History of Documented Source Code Production

No data recorded for
History of Module Development

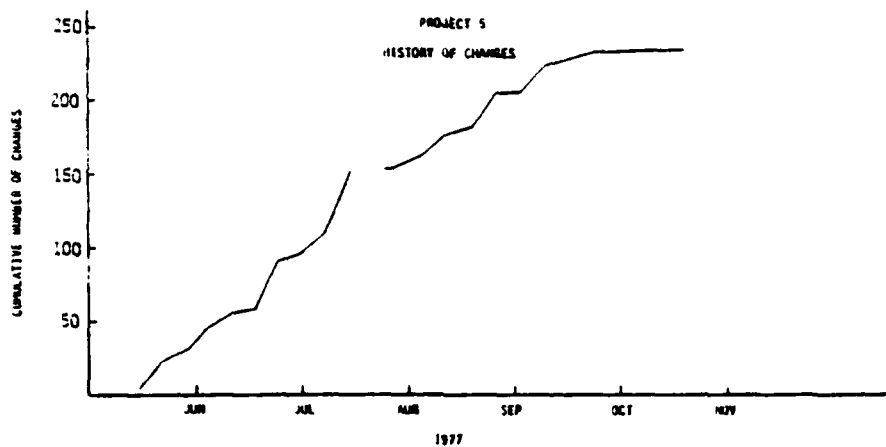


FIGURE 2.5-5

PROJECT: 5

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 6274.2

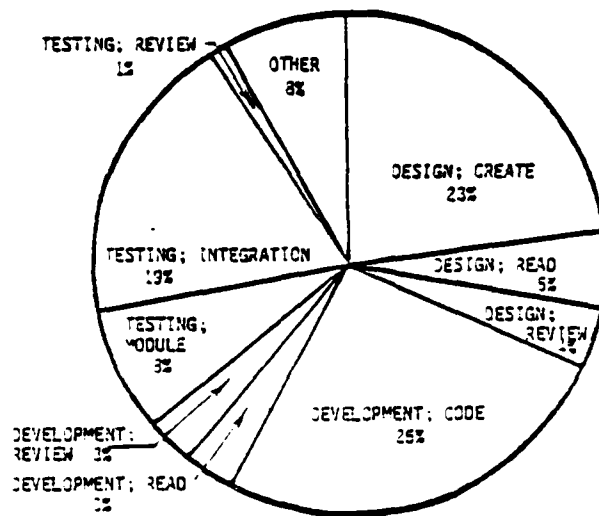


FIGURE 2.5-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS

TOTAL RUNS REPORTED: 2018

RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 3456

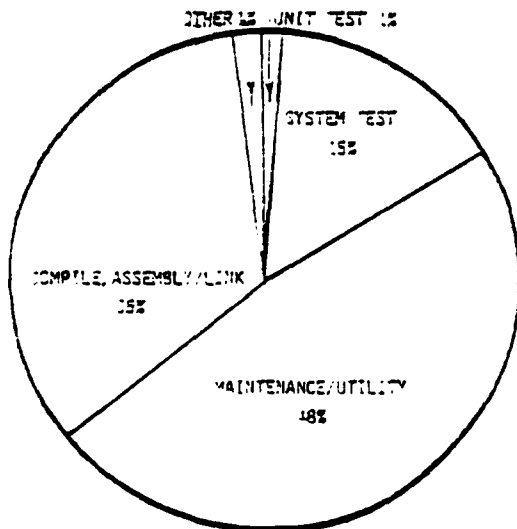


FIGURE 2.5-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS

TOTAL RUNS REPORTED: 2018

RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 22

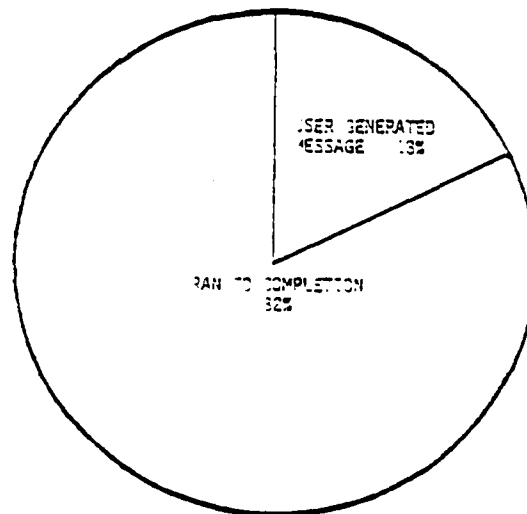


FIGURE 2.5-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 5: 311

DISTRIBUTION OF CHANGES BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 13

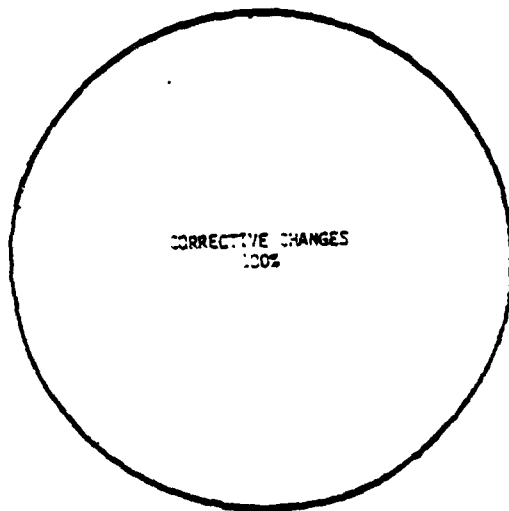


FIGURE 2.5-9

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 11

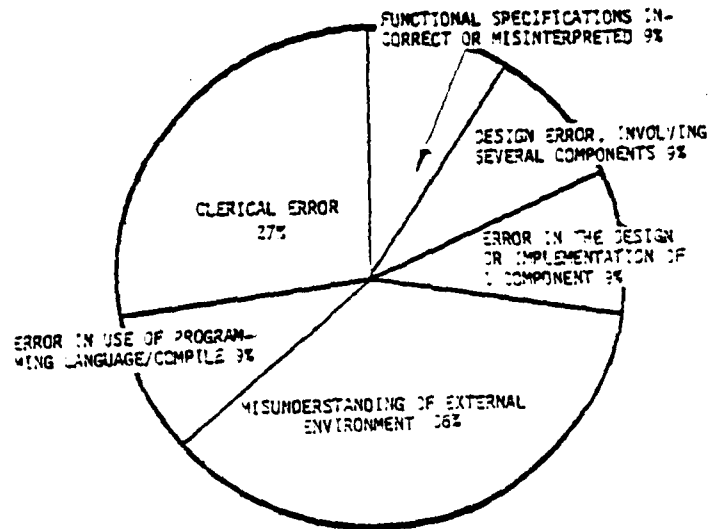


FIGURE 2.5-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 133



FIGURE 2.5-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 196

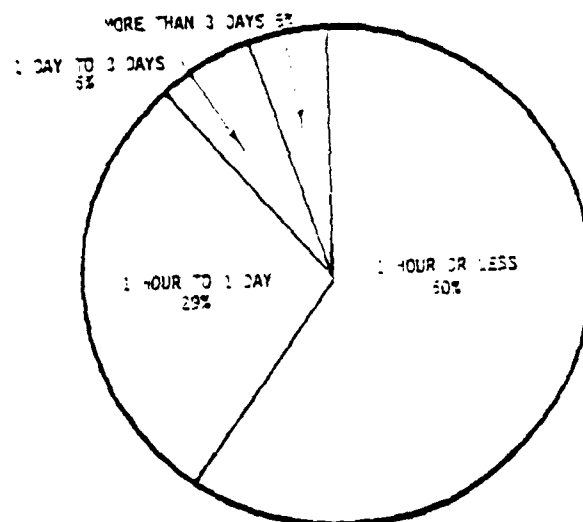


FIGURE 2.5-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 132

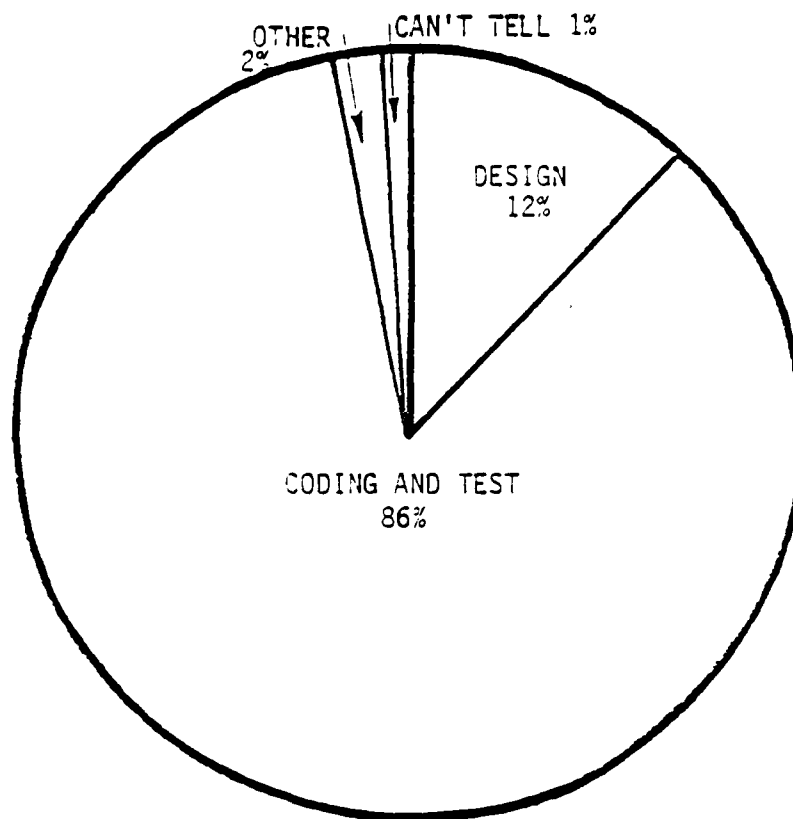


FIGURE 2.5-13

2.6 Project 6

Project 6 consists primarily of FORTRAN source code and was developed to compute satellite orientation from telemetry data. The system was developed on an IBM 360 for real-time operation. Specifications for the project were both functional and in formal English text at the system and subsystem levels. Design was top-down at the system level, with baseline diagram (tree charts) being the formalism used. Development was by iterative enhancement with no specific coding technique required. Top-down verification using stubs, code reading and walk-throughs, at the module level, were the techniques employed to validate the system. The personnel were organized into a team consisting of one chief programmer, two librarians, and up to six other programmers at any given time.

Data on project 6 is relatively complete for each category of data.

GENERAL PROJECT INFORMATION

PROJECT : 6

SIZE

DELIVERED LINES OF SOURCE CODE	111868
NEW LINES OF SOURCE CODE	84729
NASA ESTIMATE OF WORDS OF OBJECT CODE	234923
NASA ESTIMATE OF NEW OBJECT CODE	177931
NUMBER OF COMPONENTS	587
NUMBER OF MODULES	510
NUMBER OF NEW MODULES	346
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	216
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	4
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	1329

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	19.25
MANMONTHS OF DEVELOPMENT EFFORT	117
NUMBER OF COMPUTER RUNS	11376
SYSTEM 360-95 HOURS	311.0
SYSTEM 360-75 HOURS	154.0
POP 11-70 HOURS	0.0
NUMBER OF CHANGES	2045
NUMBER OF CHANGE REPORTS	491

FIGURE 2.6-1

PROJECT 5

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

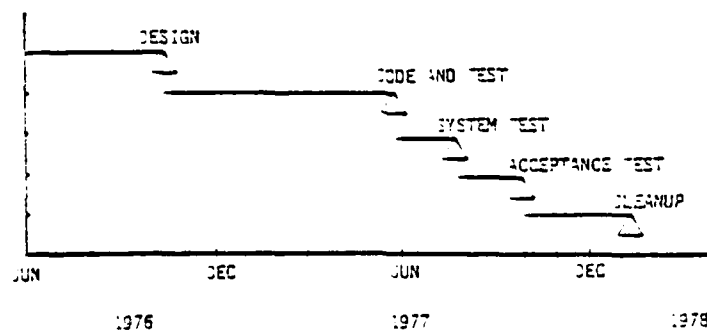


FIGURE 2.6-2

No data recorded for
History of Documented Source Code Production

No data recorded for
History of Module Development

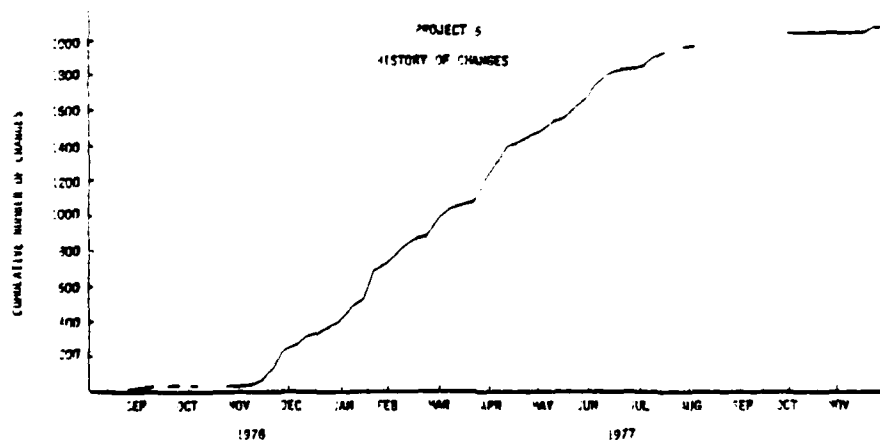


FIGURE 2.5-5

PROJECT: 6

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 6076.2

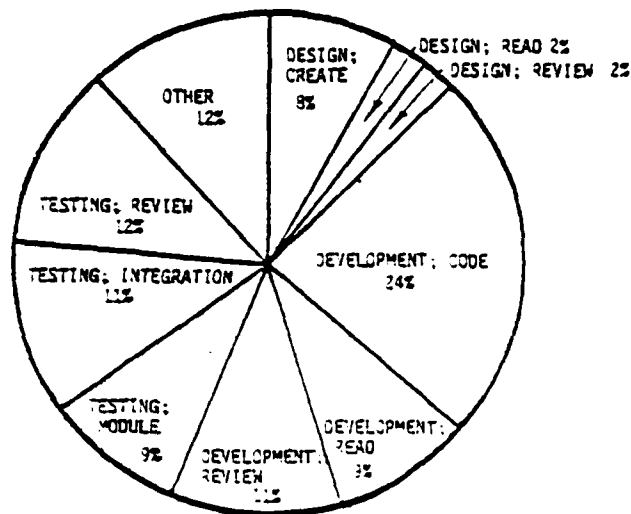


FIGURE 2.6-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS

TOTAL RUNS REPORTED: 1377

RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 2946

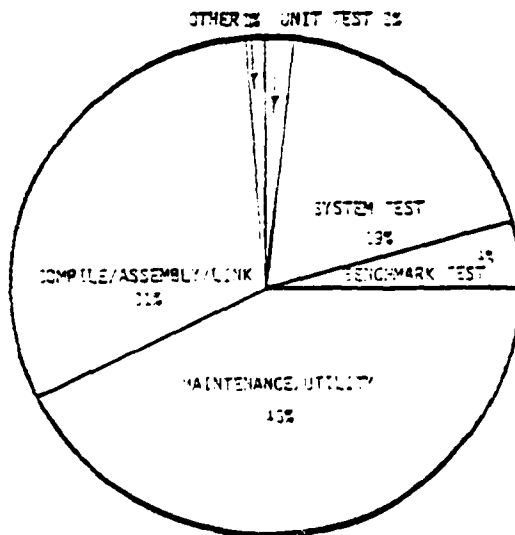


FIGURE 2.6-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS

TOTAL RUNS REPORTED: 1377

RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 54

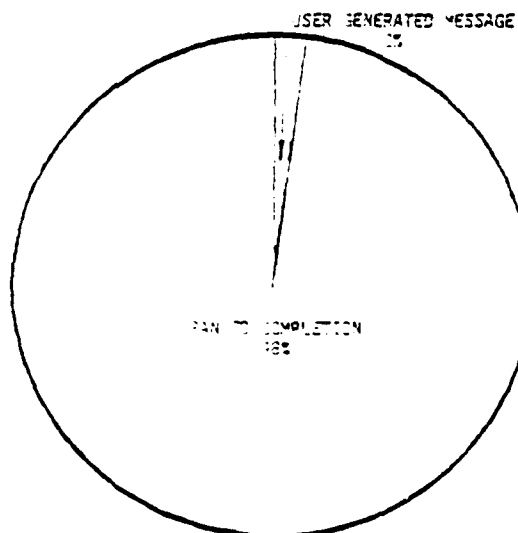


FIGURE 2.6-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 6: 491

DISTRIBUTION OF CHANGES BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 24

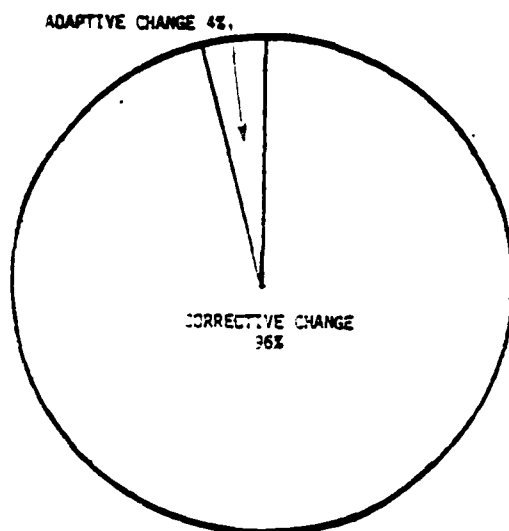


FIGURE 2.6-9

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 32

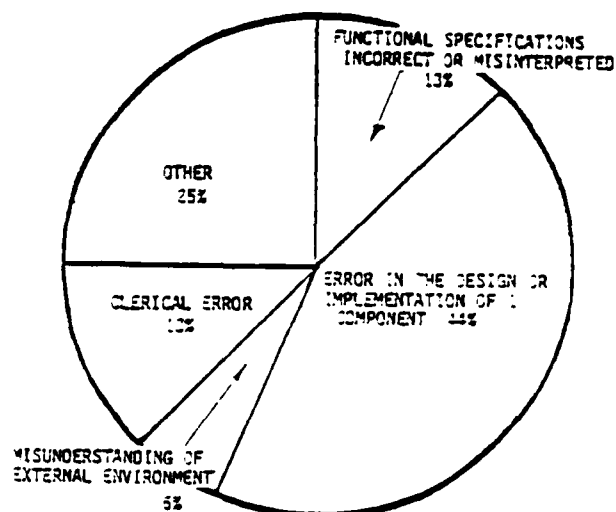


FIGURE 2.6-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 122

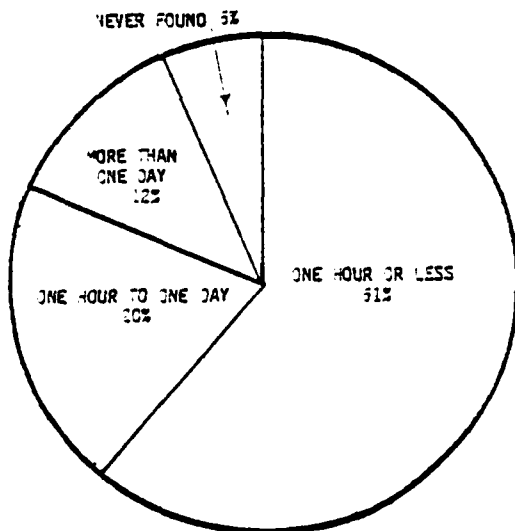


FIGURE 2.6-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGED
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 360

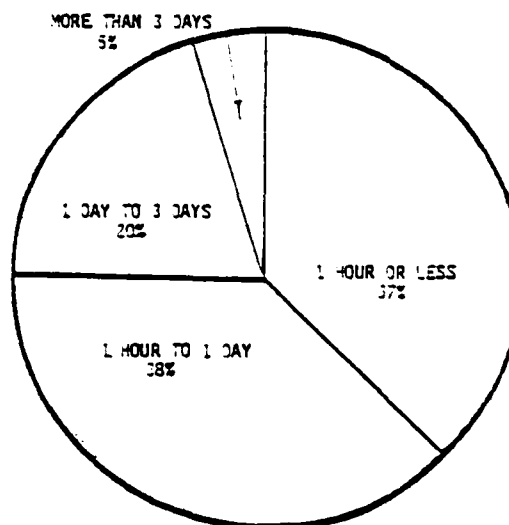


FIGURE 2.6-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 132

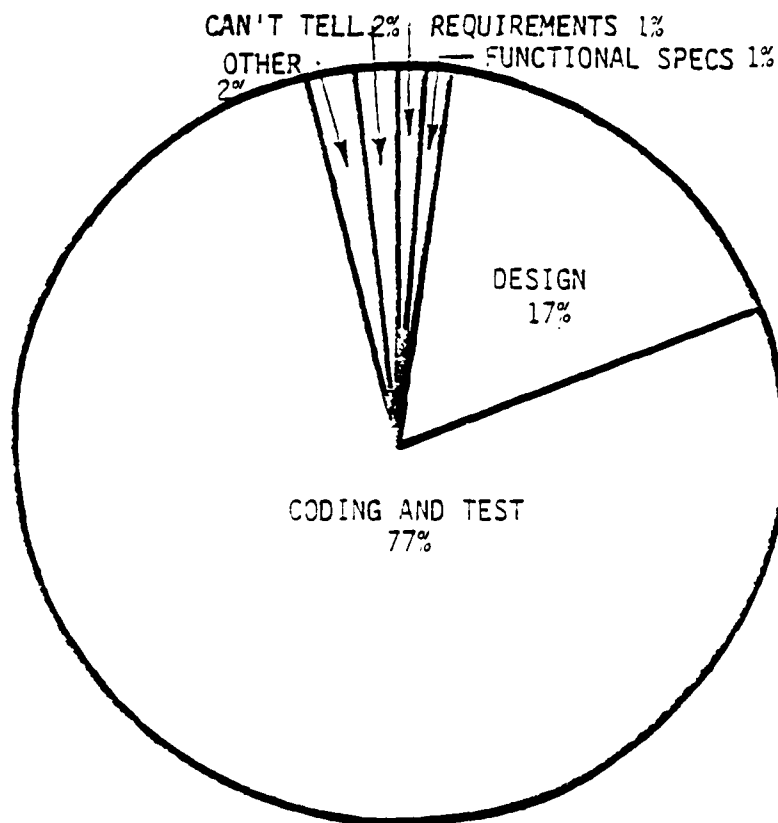


FIGURE 2.6-13

2.7 Project 7

Project 7 is a utility program designed to read, display, and perform calculations on a data set to determine satellite attitude. The program consists of approximately two thirds FORTRAN source code and one third Assembly Language Code (ALC). The software was developed on a PDP 11/70 for use on an IBM 360. No overlays were employed in this program. Specifications for the project were in English text which described the top-level of the program logic. Design was top-down in nature using baseline diagrams at the module level. Development was accomplished through iterative enhancement of subroutines composed of code which employed simulated constructs. Top-down testing of modules was used in validation.

GENERAL PROJECT INFORMATION

PROJECT : 7

SIZE

DELIVERED LINES OF SOURCE CODE	2886
NEW LINES OF SOURCE CODE	2000
NASA ESTIMATE OF WORDS OF OBJECT CODE	8370
NASA ESTIMATE OF NEW OBJECT CODE	5800
NUMBER OF COMPONENTS	24
NUMBER OF MODULES	24
NUMBER OF NEW MODULES	19
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	8.5
MANMONTHS OF DEVELOPMENT EFFORT	3
NUMBER OF COMPUTER RUNS	168
SYSTEM 360-95 HOURS	52.5
SYSTEM 360-75 HOURS	7.5
POP 11-70 HOURS	50.0
NUMBER OF CHANGES	65
NUMBER OF CHANGE REPORTS	55

FIGURE 2.7-1

PROJECT 7

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

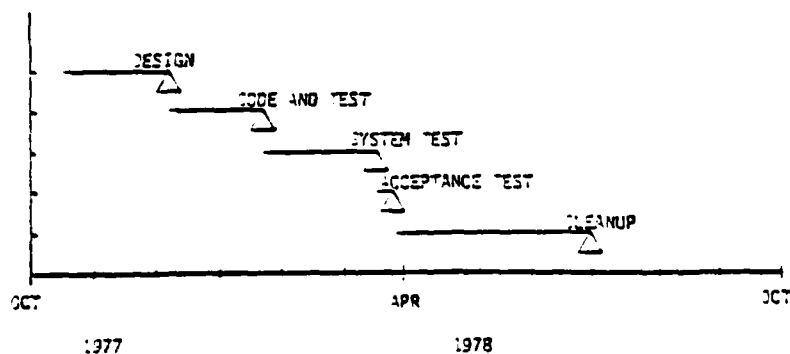


FIGURE 2.7-2

No data recorded for
History of Documented Source Code Production
History of Module Development
History of Changes

PROJECT: 7

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 313.0

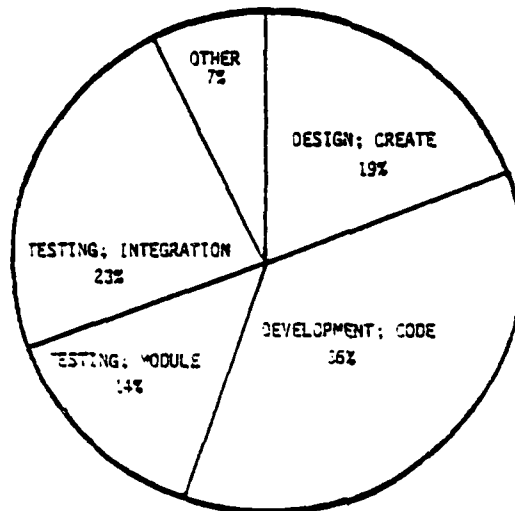


FIGURE 2.7-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS

TOTAL RUNS REPORTED: 186

RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 329

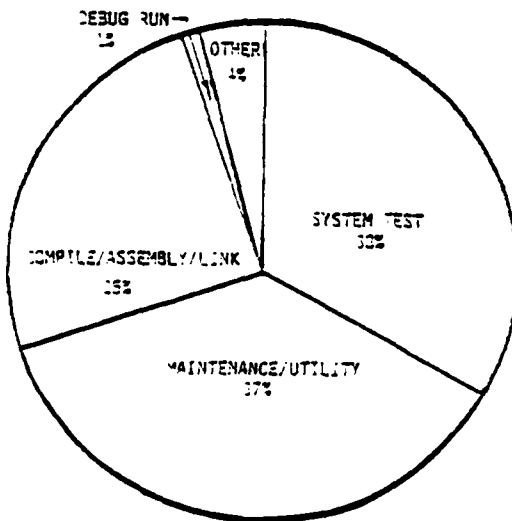


FIGURE 2.7-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS

TOTAL RUNS REPORTED: 186

RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 12

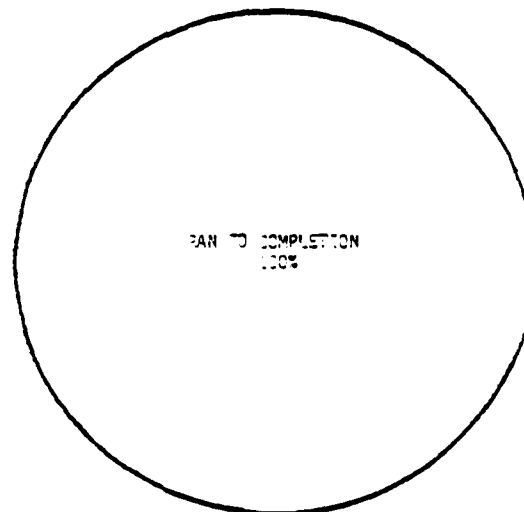


FIGURE 2.7-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 7: 55

No data recorded for
Type of Changes

No data recorded for
Type of Errors

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 33

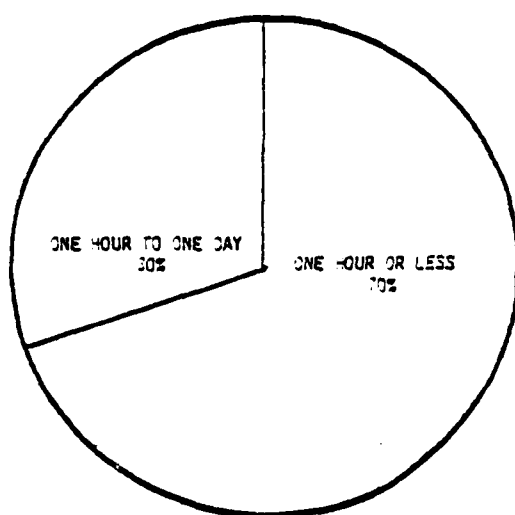


FIGURE 2.7-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 55

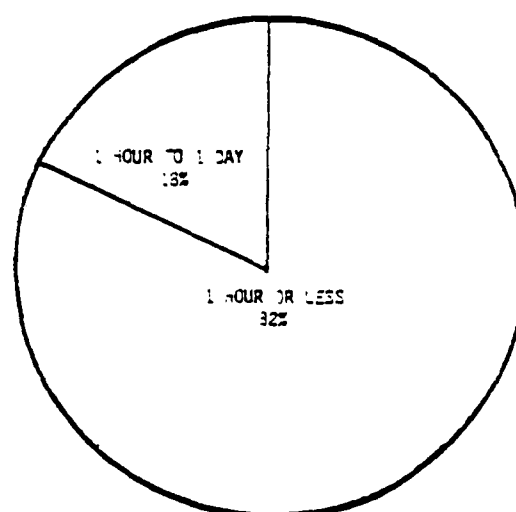


FIGURE 2.7-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 36

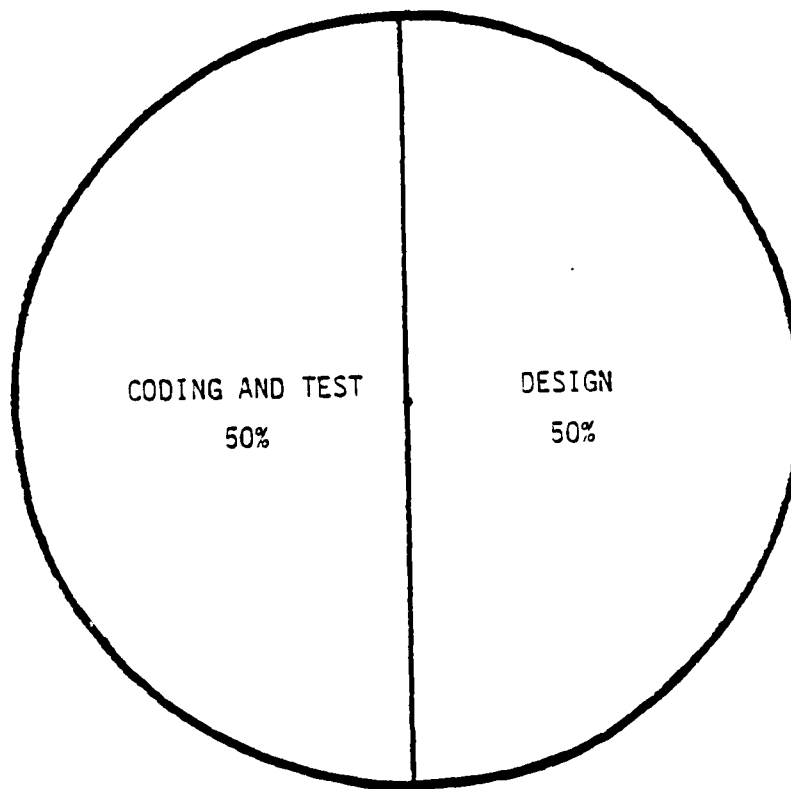


FIGURE 2.7-13

2.8 Project 8

Project 8 is an attitude determination program designed to execute in real-time on an IBM 360, and is composed primarily of FORTRAN source code. Extensive amounts of code were reused from Project 3 in the development of this project. The specifications for this project were functional at the system level and procedural at the subroutine level. The system was designed and developed in a top-down fashion using baseline diagrams. A program design language was used to specify the subroutine level functions. Testing of the system was top-down at the system level and specification-driven at lower levels during validation. The personnel were organized in a team structure consisting of a team leader, whose responsibilities paralleled that of a chief programmer, a librarian, and three programmers in addition to the project manager.

The data on Project 8 is relatively complete for each category of data. Note that 2792 changes are recorded under General Project Information. The source of this number is the NASA estimated statistics file. However, the project history file shows only 415 changes for Project 8.

GENERAL PROJECT INFORMATION

PROJECT : 8

SIZE

DELIVERED LINES OF SOURCE CODE	75420
NEW LINES OF SOURCE CODE	20075
NASA ESTIMATE OF WORDS OF OBJECT CODE	158382
NASA ESTIMATE OF NEW OBJECT CODE	42161
NUMBER OF COMPONENTS	423
NUMBER OF MODULES	374
NUMBER OF NEW MODULES	92
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	201
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	2
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	1186

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	10.25
MANMONTHS OF DEVELOPMENT EFFORT	40
NUMBER OF COMPUTER RUNS	1032
SYSTEM 360-95 HOURS	33.3
SYSTEM 360-75 HOURS	16.3
POP 11-70 HOURS	0.3
NUMBER OF CHANGES	2792
NUMBER OF CHANGE REPORTS	239

FIGURE 2.8-1

PROJECT 3

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

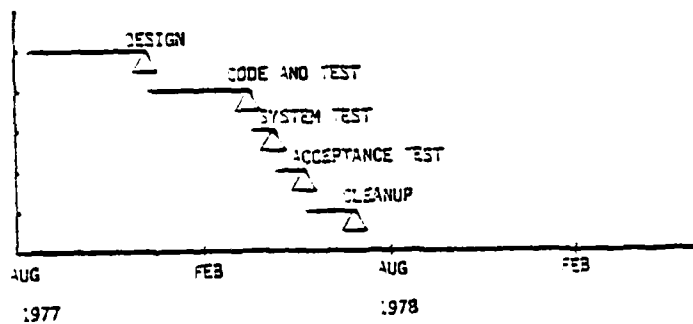


FIGURE 2.8-2

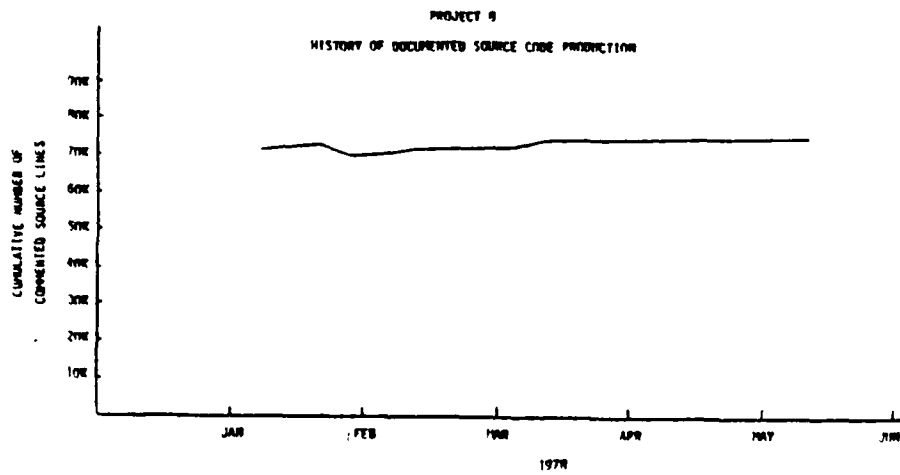


FIGURE 2.8-3

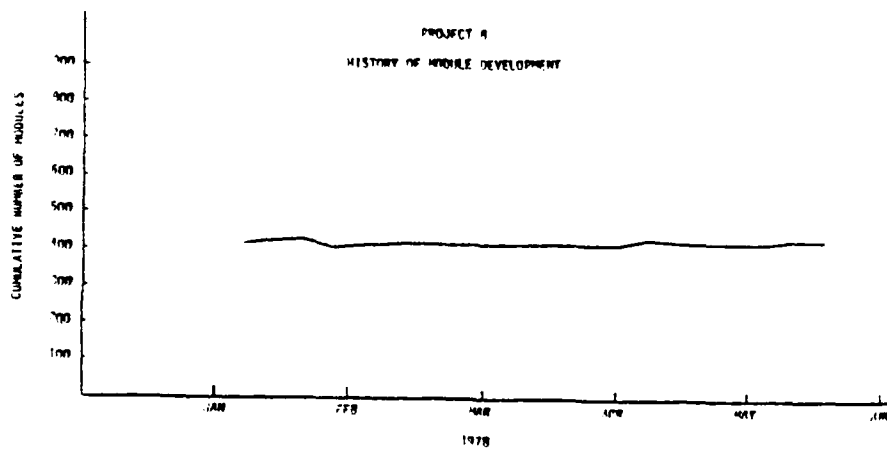


FIGURE 2.8-4

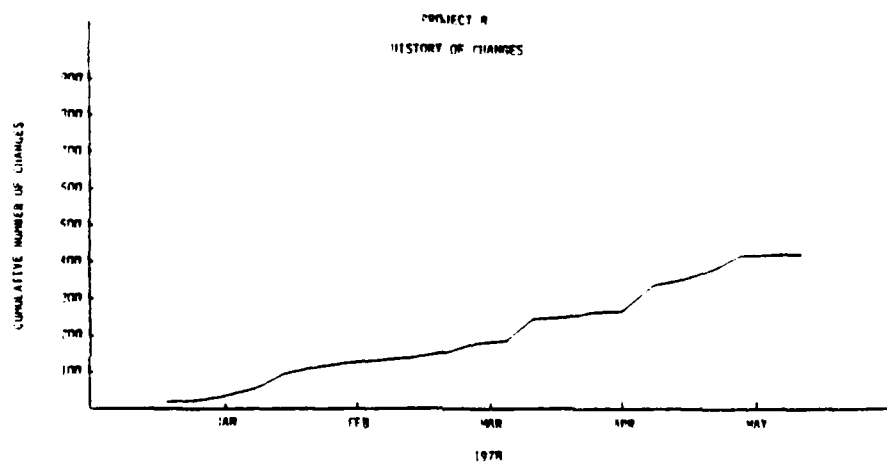


FIGURE 2.8-5

PROJECT: 8

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 4793.5

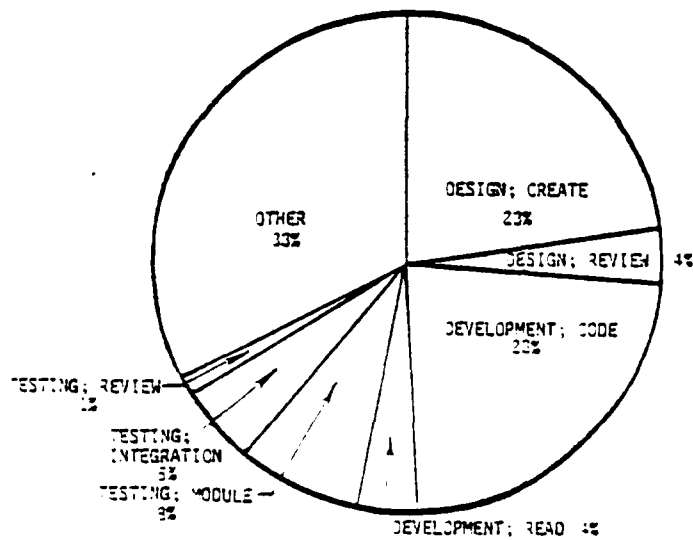


FIGURE 2.3-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS

TOTAL RUNS REPORTED: 984

RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 1748

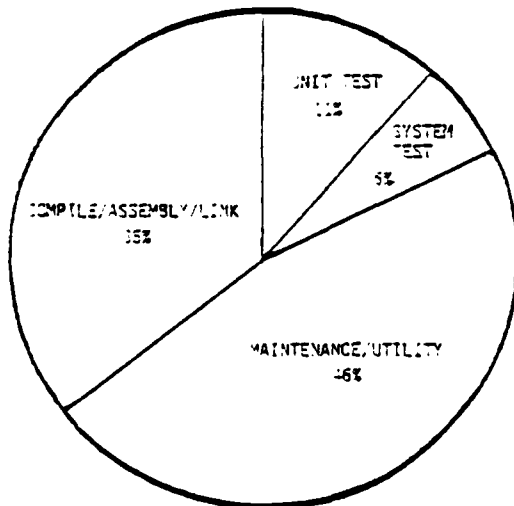


FIGURE 2.3-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS

TOTAL RUNS REPORTED: 984

RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 9

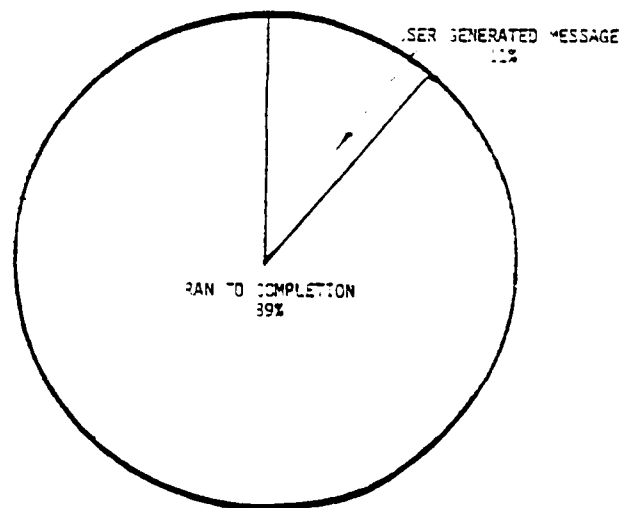


FIGURE 2.3-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 8: 239

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 110

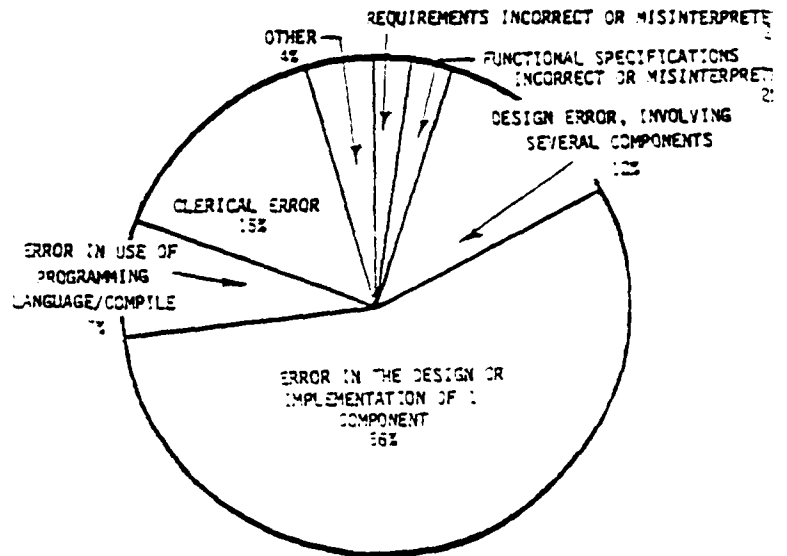


FIGURE 2.3-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 30

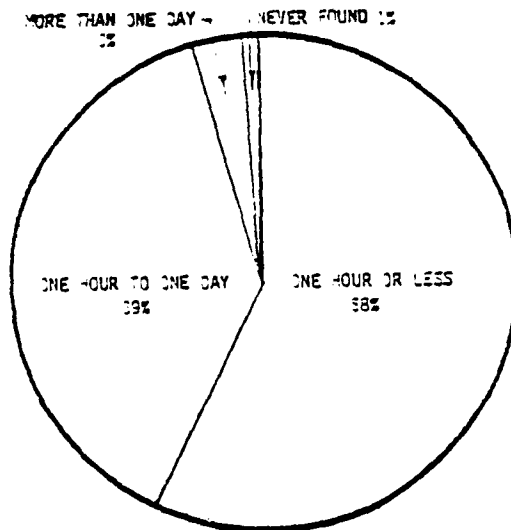


FIGURE 2.3-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 55

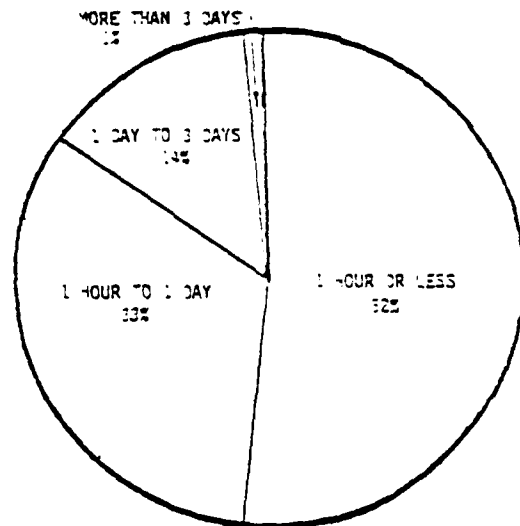


FIGURE 2.3-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 99

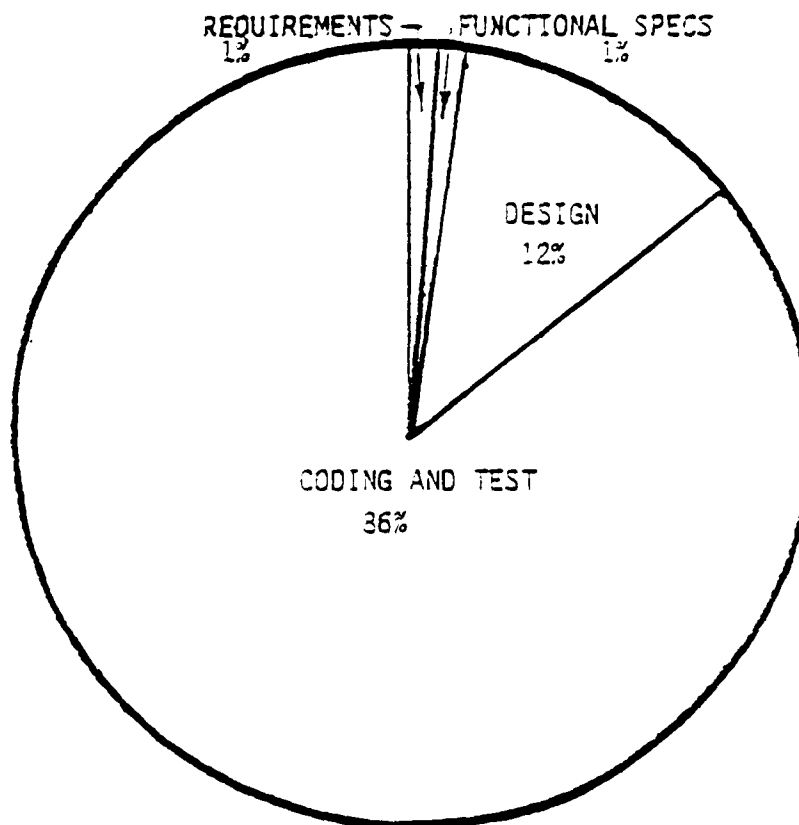


FIGURE 2.8-13

This page was intentionally left blank

2.10 Project 10

Project 10 consists primarily of FORTRAN source code. The purpose of the program was to determine spacecraft attitude. The specifications for the project were formal at a subroutine level. Design and development were top-down at the subsystem level using iterative enhancement at the subroutine level. Baseline diagrams and Program Design language (PDL) were the specification techniques used to design the program, and structured code was used in its development. Validation was specification-driven at the system level. The project personnel were organized in a structure similar to a chief programmer team with one chief programmer, three librarians, and five programmers in addition to the project manager.

Run analysis and resource expenditure data is not recorded for Project 10.

GENERAL PROJECT INFORMATION

PROJECT : 10

SIZE

DELIVERED LINES OF SOURCE CODE	75393
NEW LINES OF SOURCE CODE	49316
NASA ESTIMATE OF WORDS OF OBJECT CODE	158325
NASA ESTIMATE OF NEW OBJECT CODE	103560
NUMBER OF COMPONENTS	538
NUMBER OF MODULES	535
NUMBER OF NEW MODULES	337
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	140
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	1
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	1477

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	14.75
MANMONTHS OF DEVELOPMENT EFFORT	92
NUMBER OF COMPUTER RUNS	7500
SYSTEM 360-95 HOURS	209.0
SYSTEM 360-75 HOURS	193.0
POP 11-70 HOURS	0.0
NUMBER OF CHANGES	1576
NUMBER OF CHANGE REPORTS	46

FIGURE 2.10-1

PROJECT 10

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

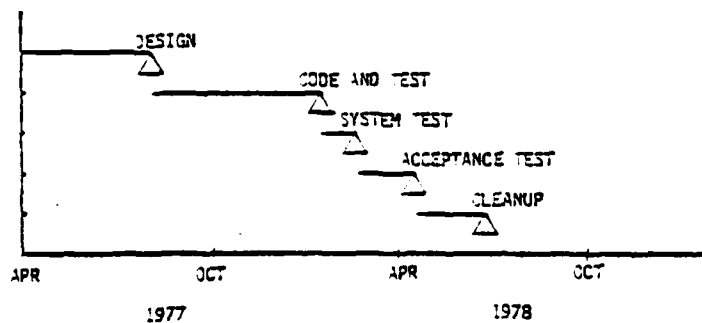


FIGURE 2.10-2

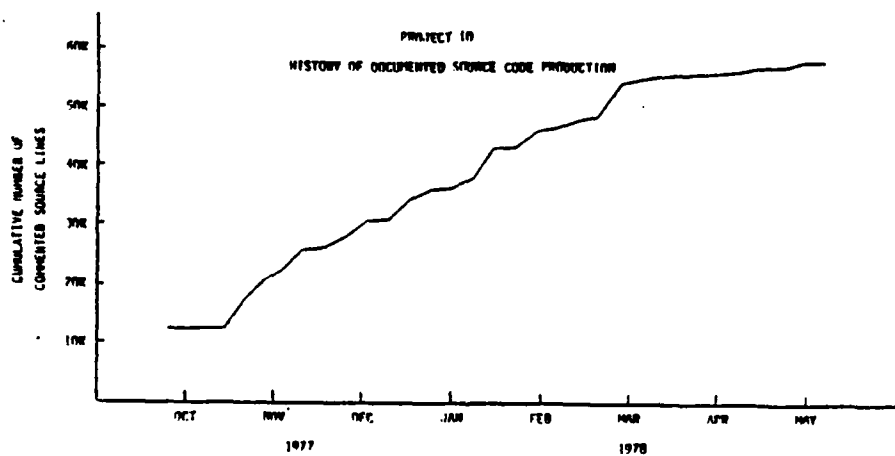


FIGURE 2.10-3

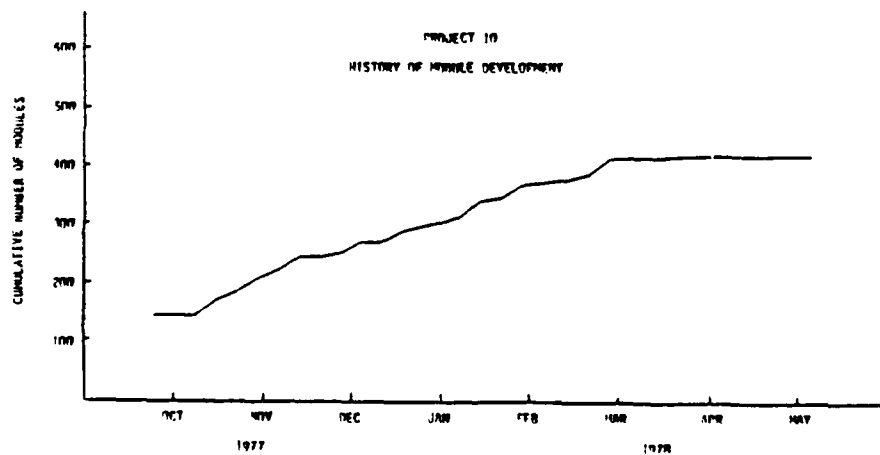


FIGURE 2.10-4

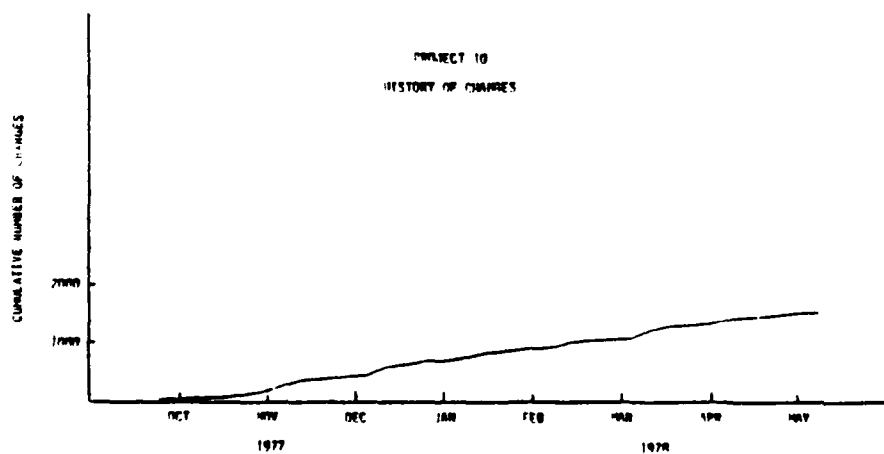


FIGURE 2.10-5

PROJECT: 10
 DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)
 HOURS THAT THIS DISTRIBUTION IS BASED ON: 5690.9

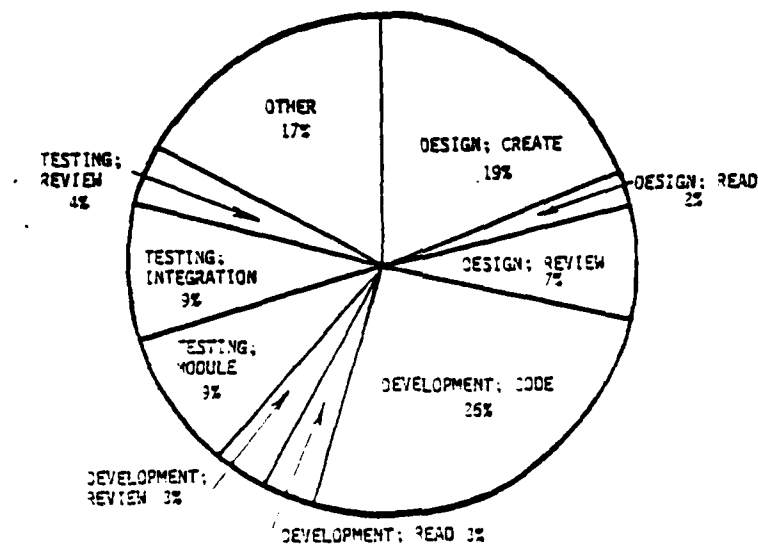


FIGURE 2.10-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS
 TOTAL RUNS REPORTED: 1312
 RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 2123

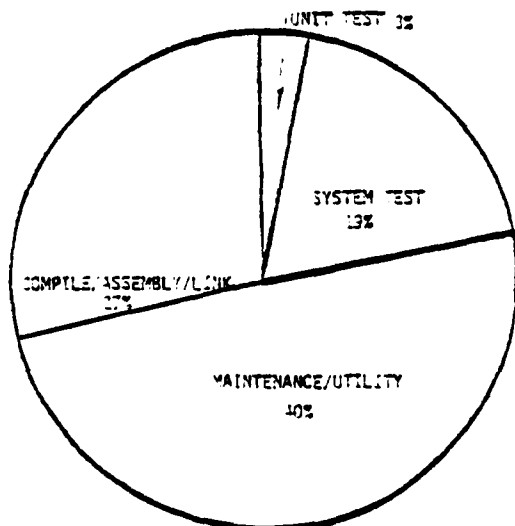


FIGURE 2.10-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS
 TOTAL RUNS REPORTED: 1312
 RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 13

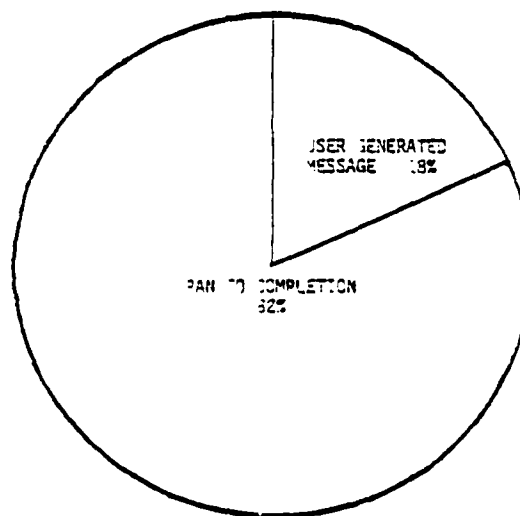


FIGURE 2.10-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 10: 46

No data recorded for
Type of Changes

No data recorded for
Type of Errors

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 23

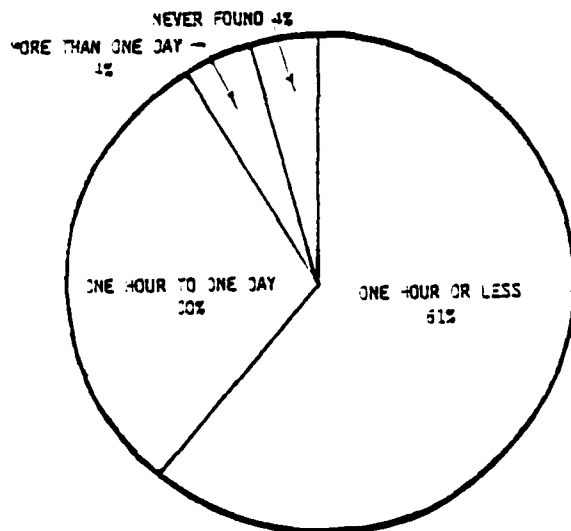


FIGURE 2.10-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 46

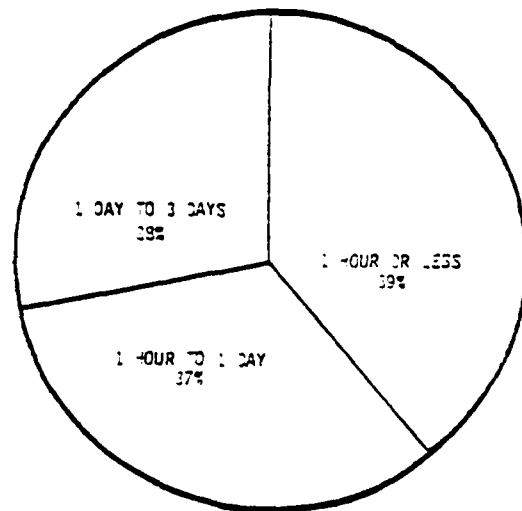


FIGURE 2.10-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 25

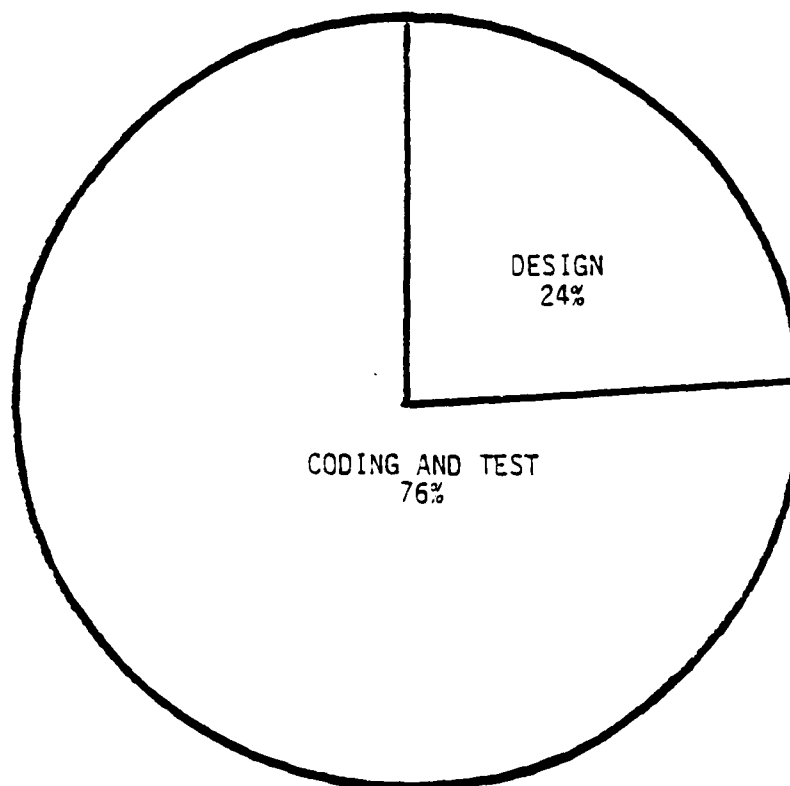


FIGURE 2.10-13

2.11 Project 11

Very little information is available concerning the development of Project 11 other than that it is for a scientific application and composed primarily of FORTRAN source code. Data collected on Project 11 includes development schedule information and run analysis data.

2.15 Project 15

Project 15 is a FORTRAN program which analyzes FORTRAN source code. It was developed and is operating on a PDP 11/70. The system is in an overlay structure of three segments. The specifications for the project were detailed in the form of English text at the system level and with 80% of the modules having procedure oriented specifications, and 20% having formal specifications. Baseline diagrams were used in the top-down design and development of the system, but one component was specifically designed and developed in a "hardest first" fashion. Structured code was also used in development. In the validation of the system, flow of control was tested in a top-down fashion using stubs while modules were tested in a bottom-up fashion using drivers. Inspection was accomplished by code reading within modules and by walk-throughs at the subsystem level. Only one programmer was used in the development of this project.

Data on Project 15 is relatively incomplete. The number of changes recorded on the estimated statistics file (23) is inconsistent with the number of changes recorded on the project history file (245).

GENERAL PROJECT INFORMATION

PROJECT : 15

SIZE

DELIVERED LINES OF SOURCE CODE	6300
NEW LINES OF SOURCE CODE	6300
NASA ESTIMATE OF WORDS OF OBJECT CODE	18270
NASA ESTIMATE OF NEW OBJECT CODE	18270
NUMBER OF COMPONENTS	69
NUMBER OF MODULES	69
NUMBER OF NEW MODULES	69
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	194
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	194
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	194

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	7.0
MANMONTHS OF DEVELOPMENT EFFORT	5
NUMBER OF COMPUTER RUNS.	3
SYSTEM 360-95 HOURS	0.0
SYSTEM 360-75 HOURS	0.0
PDP 11-70 HOURS	0.0
NUMBER OF CHANGES	23
NUMBER OF CHANGE REPORTS	16

FIGURE 2.15-1

No Data recorded
for Development Schedule

No data recorded for
History of Documented Source Code Production

No data recorded for
History of Module Development

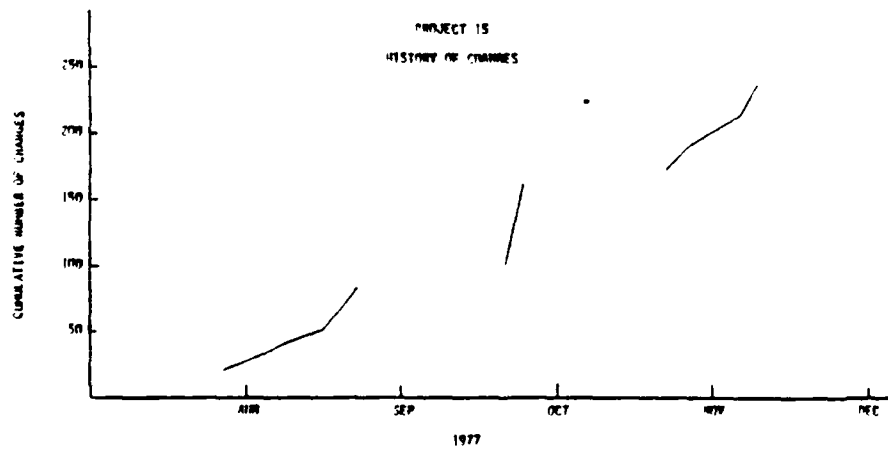


FIGURE 2.15-5

PROJECT: 15
 DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)
 HOURS THAT THIS DISTRIBUTION IS BASED ON: 434.2

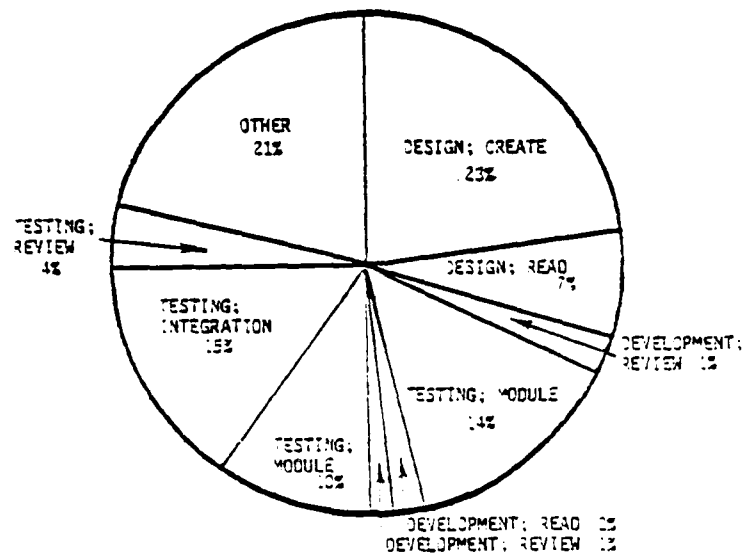


FIGURE 2.15-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS
 TOTAL RUNS REPORTED: 58
 RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 38

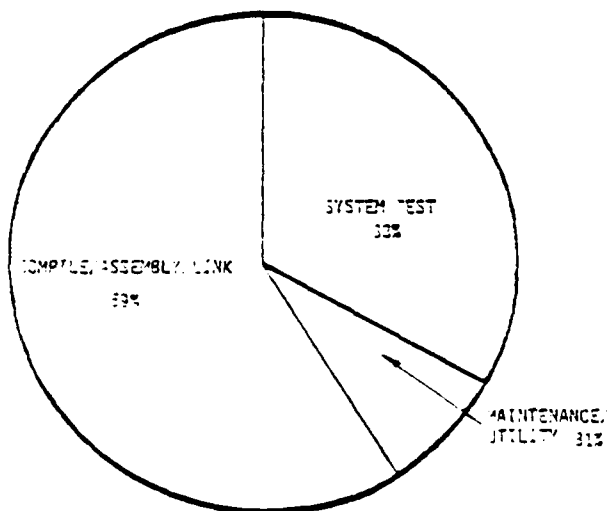


FIGURE 2.15-7

No data recorded for
 Computer Run Results

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 15: 16

DISTRIBUTION OF CHANGES BY TYPE

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 10

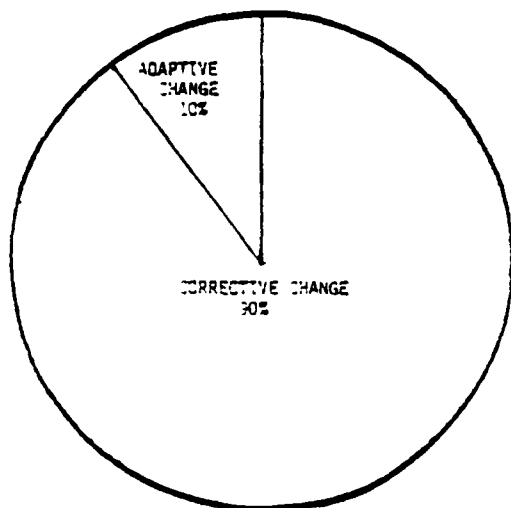


FIGURE 2.15-9

DISTRIBUTION OF ERRORS BY TYPE

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 10

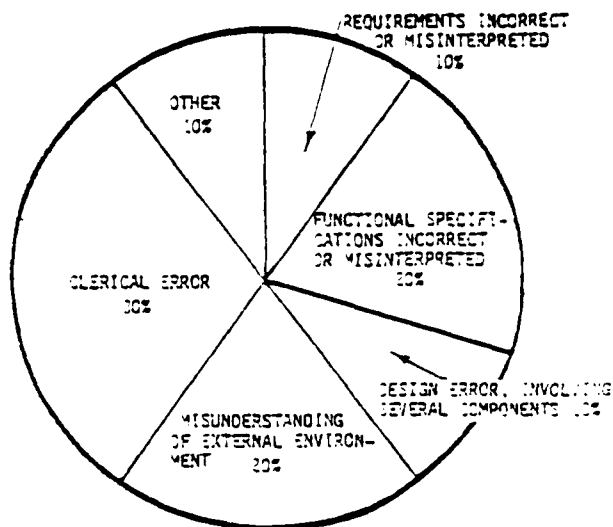


FIGURE 2.15-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 3

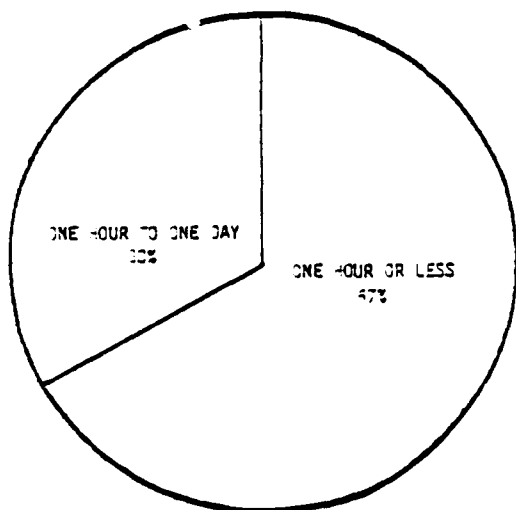


FIGURE 2.15-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 14

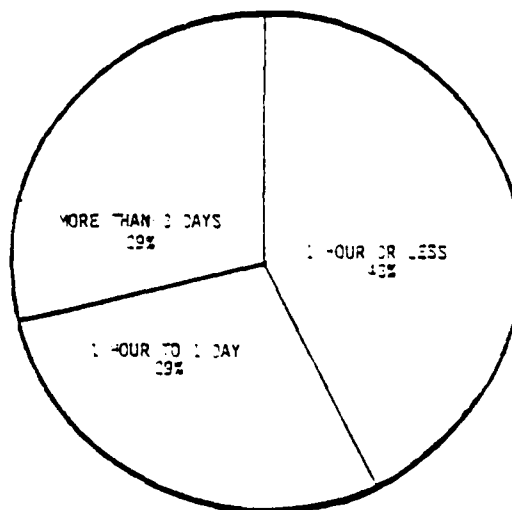


FIGURE 2.15-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 5

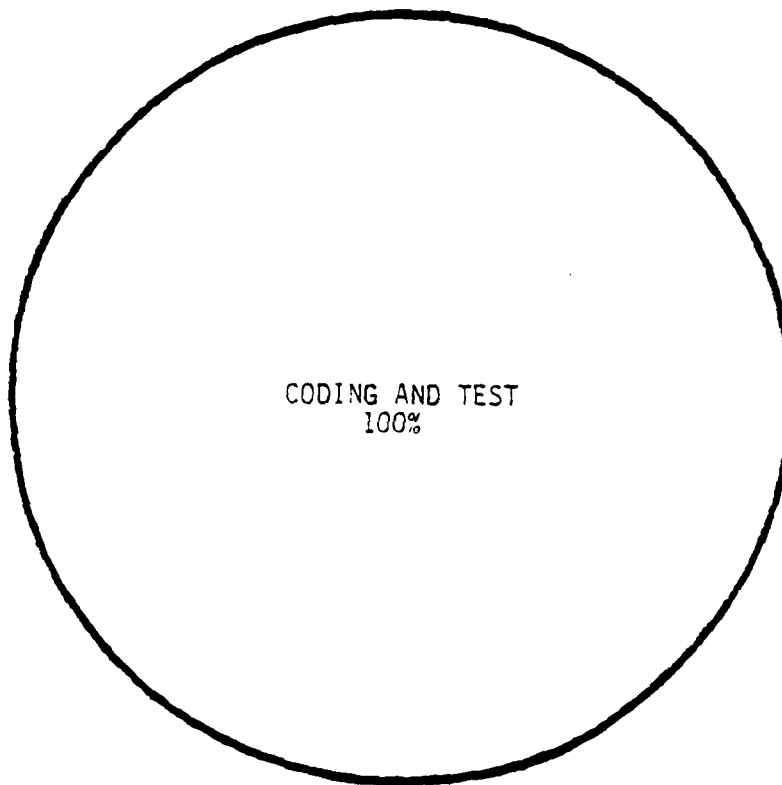


FIGURE 2.15-13

2.16 Project 16

Project 16 consists of a FORTRAN program developed to generate monthly reports containing financial data. The program is a single overlay structure developed on a PDP 11/70 but able to execute on either a PDP 11/70 or an IBM 360. Functional specification were employed at the module level. Program and module design was accomplished using top-down techniques. Bottom-up techniques were used to develop each module, with iterative enhancement of blocks of source code within modules. Flowcharts were used to specify the design of each module. Code reading and specification-driven testing were used to perform verification and validation. Only one programmer was assigned to this project.

Data on Project 16 consists of component and resource expenditure data only. As a result, no pie charts were developed for this project.

2.17 Project 17

Project 17 is a FORTRAN program designed to perform image data retrieval from mass storage and assist in calculation of attitude determination. The specifications for the project were functional. Design and development of this project was accomplished using an iterative enhancement approach with no specific coding standards being required. Walk-throughs were used in the verification of the project. Only two programmers were assigned to this project with one subordinate to the other.

Data on Project 17 consists of computer run analysis data only. As a result, no pie charts were developed for this project.

2.18 Project 18

Project 18 is an interactive system designed as a generalized graphics package to display data generated by other systems. The project consists of three independent programs written in FORTRAN. It was developed and operates on a PDP 11/70. The specifications for the project were functional. Design was accomplished using a data flow technique with iterative enhancement which utilized baseline diagrams. Development was top-down with iterative enhancement using structured code. Top-down testing and code reading were employed during verification and validation. The two programmers used on this project were not organized in any specific manner.

Data on Project 18 consists of partial scheduling information and a small sample of change reports. As a result, no pie charts were developed for this project.

2.19 Project 19

Project 19 consists of a main attitude determination program and four utilities subordinate to it. Approximately two thirds of the project is written in FORTRAN, the remaining third being written in ALC. The project was designed to operate in near real-time on an IBM 360. The specifications for the project were both functional and in English text down to the subsystem level. Design was accomplished by iterative enhancement using baseline diagrams and a PDL. Development of the project was also by iterative enhancement of simulated constructs of blocks of code. Validation and verification was accomplished through top-down testing of modules and specification-driven testing of functions. The personnel assigned to the project were loosely organized into a programming team with one task leader, one librarian, and seven programmers, in addition to the project manager.

Data on Project 19 is relatively complete for each category of data.

GENERAL PROJECT INFORMATION

PROJECT : 19

SIZE

DELIVERED LINES OF SOURCE CODE	85369
NEW LINES OF SOURCE CODE	76883
NASA ESTIMATE OF WORDS OF OBJECT CODE	179275
NASA ESTIMATE OF NEW OBJECT CODE	161455
NUMBER OF COMPONENTS	539
NUMBER OF MODULES	519
NUMBER OF NEW MODULES	418
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	173
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	2
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	2482

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	17.5
MANMONTHS OF DEVELOPMENT EFFORT	100
NUMBER OF COMPUTER RUNS	0
SYSTEM 360-95 HOURS	0.0
SYSTEM 360-75 HOURS	0.0
PDP 11-70 HOURS	0.0
NUMBER OF CHANGES	1050
NUMBER OF CHANGE REPORTS	579

FIGURE 2.19-1

PROJECT 19

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

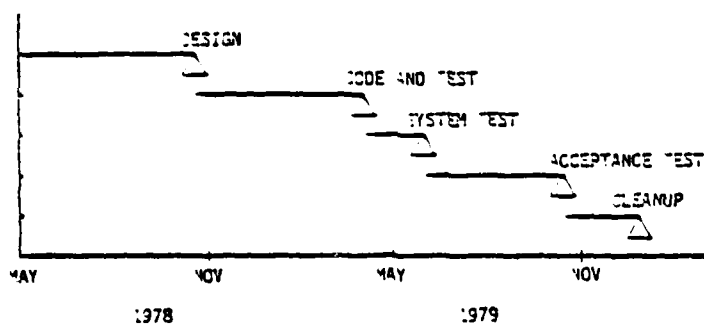


FIGURE 2.19-2

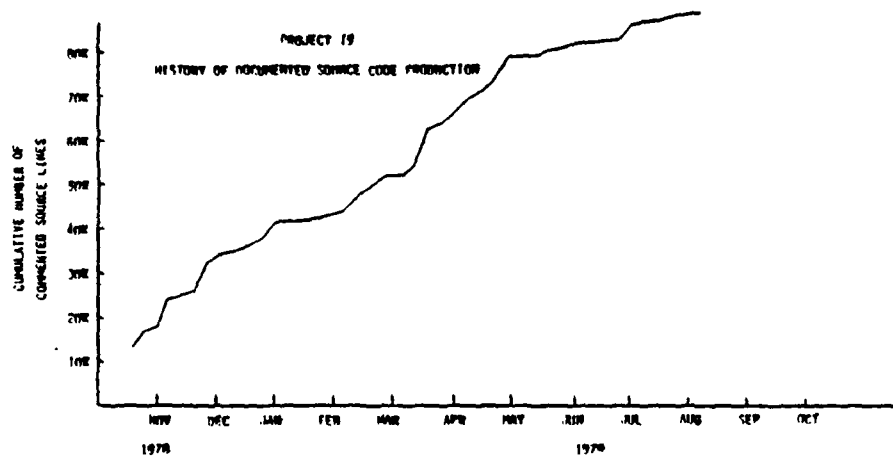


FIGURE 2.19-3

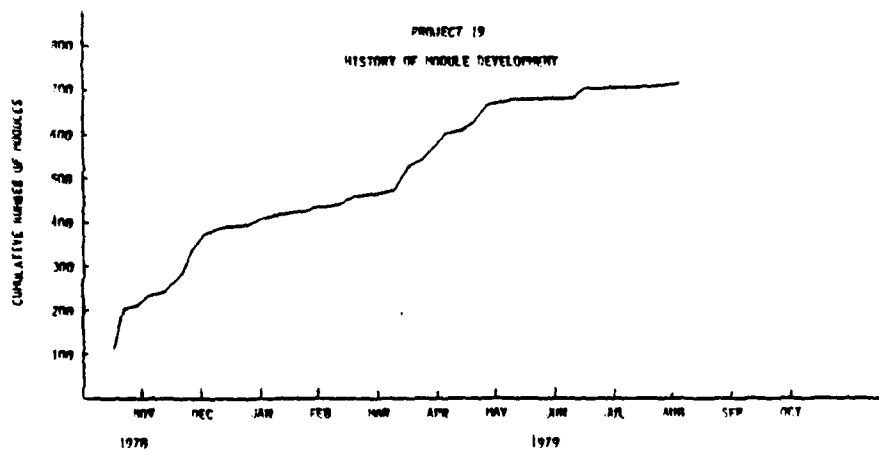


FIGURE 2.19-4

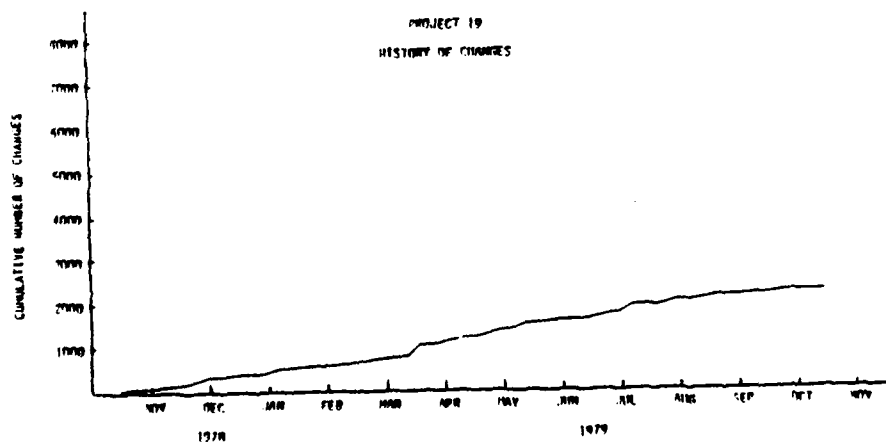


FIGURE 2.19-5

PROJECT 19
 DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)
 HOURS THAT THIS DISTRIBUTION IS BASED ON: 17595.6

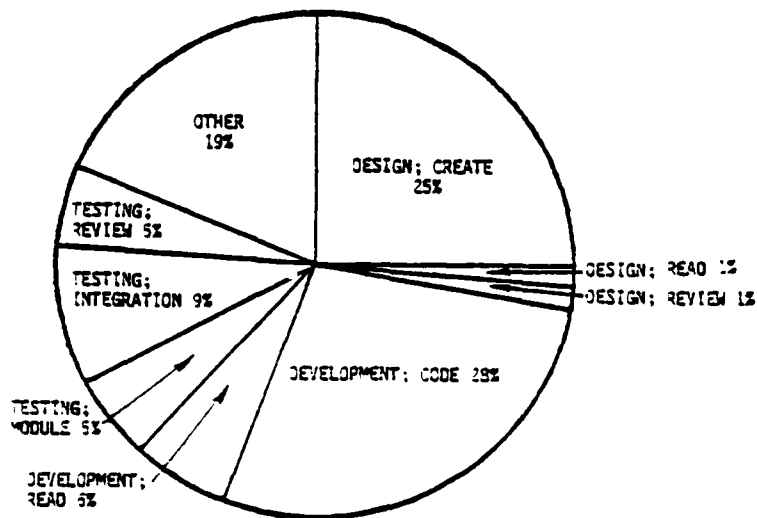


FIGURE 2.19-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS
 TOTAL RUNS REPORTED: 3118
 RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 4188

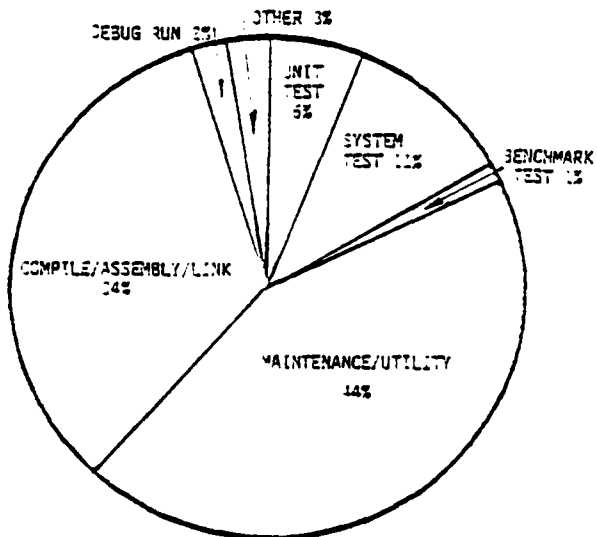


FIGURE 2.19-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS
 TOTAL RUNS REPORTED: 3118
 RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 71

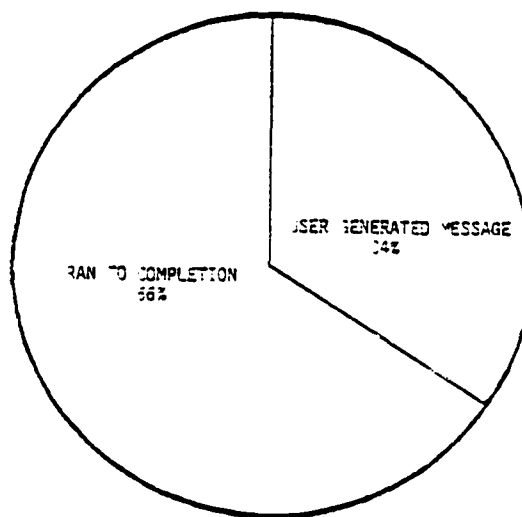


FIGURE 2.19-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 19: 679

DISTRIBUTION OF CHANGES BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 679

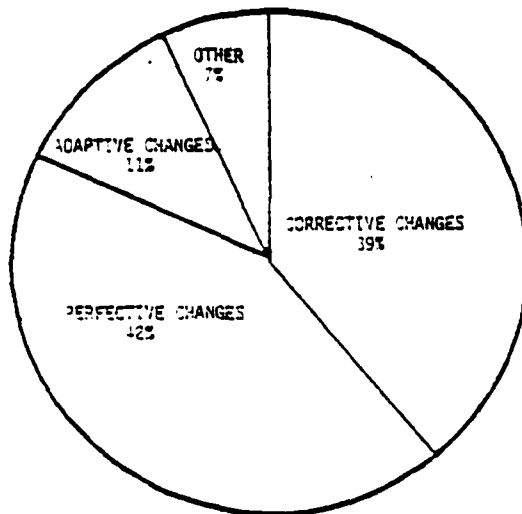


FIGURE 2.19-9

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 291

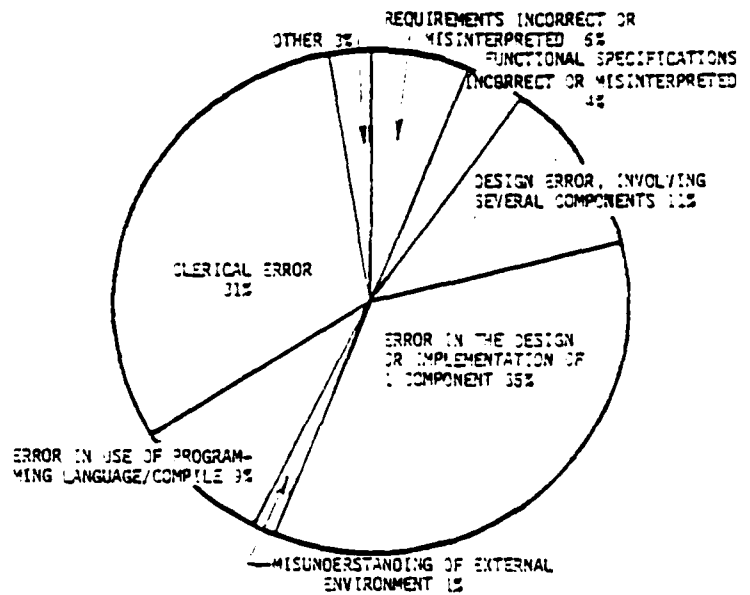


FIGURE 2.19-10

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 678

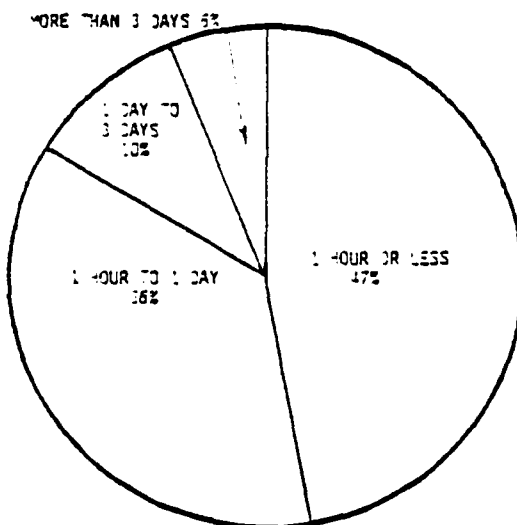


FIGURE 2.19-11

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 273

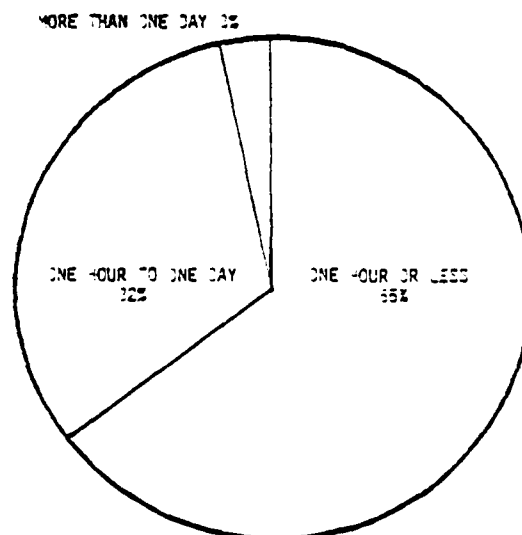


FIGURE 2.19-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 266

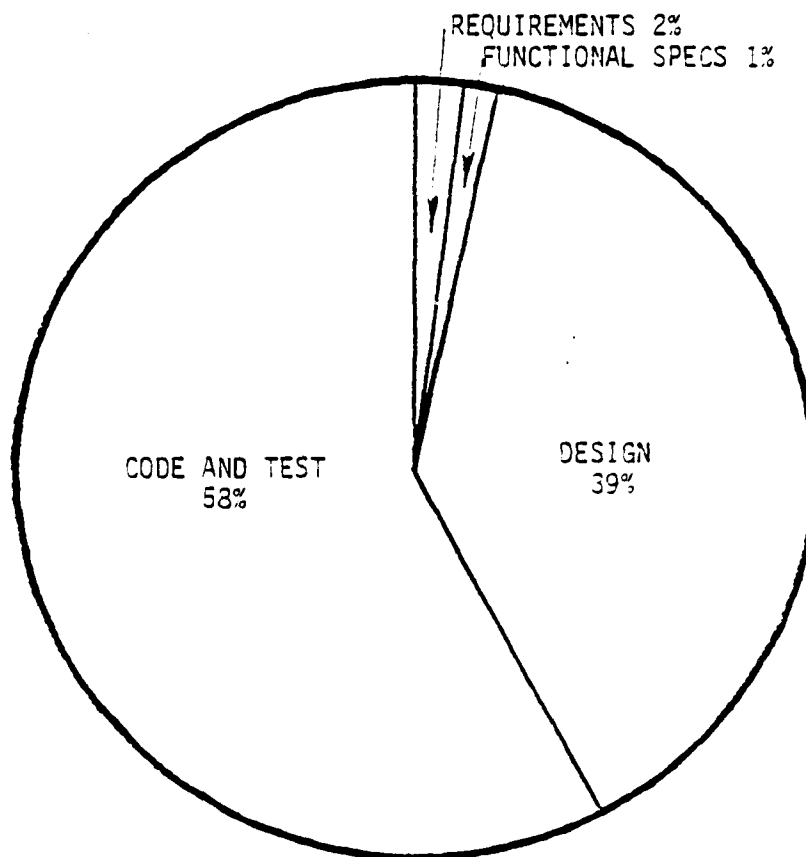


FIGURE 2.19-13

This page was intentionally left blank

2.21 Project 21

Project 21 is a FORTRAN program developed to compute orbital parameters. The program was developed and designed to operate on an IBM 360. The specifications for this project were functional to the module level. Top-down design and development to the subroutine level and iterative enhancement of the subroutines were the techniques employed during development. During design, flowcharts and baseline diagrams were used at the top-level of the system and a PDL was used in the design and development of the entire system. Top-down testing using stubs, code reading and walk-throughs were used in validation and verification of the program. In addition to the manager, the personnel were organized into a team consisting of one task leader and one programmer analyst.

In addition to general project information, data on Project 21 consists of component information and change report data. It is important to note that delivered lines of source code, manmonths of development effort and number of changes were not recorded in the NASA database.

GENERAL PROJECT INFORMATION

PROJECT : 21

SIZE

DELIVERED LINES OF SOURCE CODE	
NEW LINES OF SOURCE CODE	
NASA ESTIMATE OF WORDS OF OBJECT CODE	
NASA ESTIMATE OF NEW OBJECT CODE	115988
NUMBER OF COMPONENTS	140
NUMBER OF MODULES	140
NUMBER OF NEW MODULES	140
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	19.0
MANMONTHS OF DEVELOPMENT EFFORT	
NUMBER OF COMPUTER RUNS	9970
SYSTEM 360-95 HOURS	158.0
SYSTEM 360-75 HOURS	155.0
PDP 11-70 HOURS	2.0
NUMBER OF CHANGES	
NUMBER OF CHANGE REPORTS	150

FIGURE 2.21-1

PROJECT 21

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

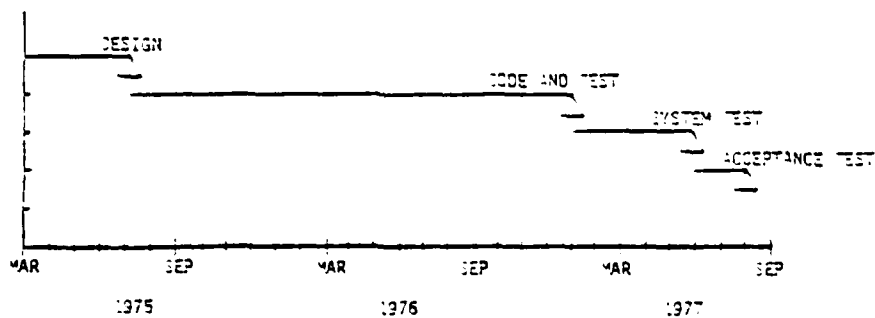


FIGURE 2.21-2

No data recorded for
History of Documented Source Code Production
History of Module Development
History of Changes

PROJECT: 21

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 1.0

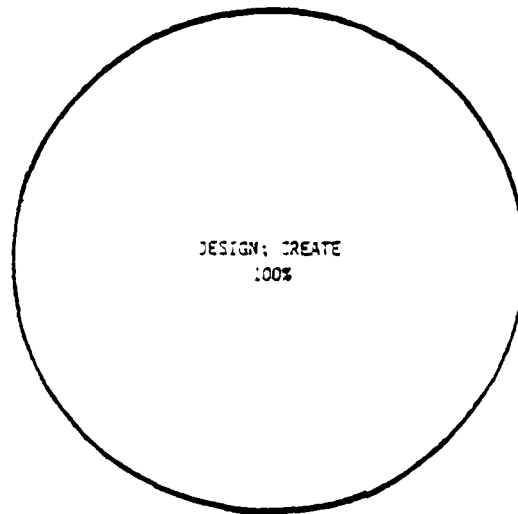


FIGURE 2.21-6

No data recorded for
Computer Run Purposes

No data recorded for
Computer run Results

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 21: 150

No data recorded for
Type of Changes

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 113

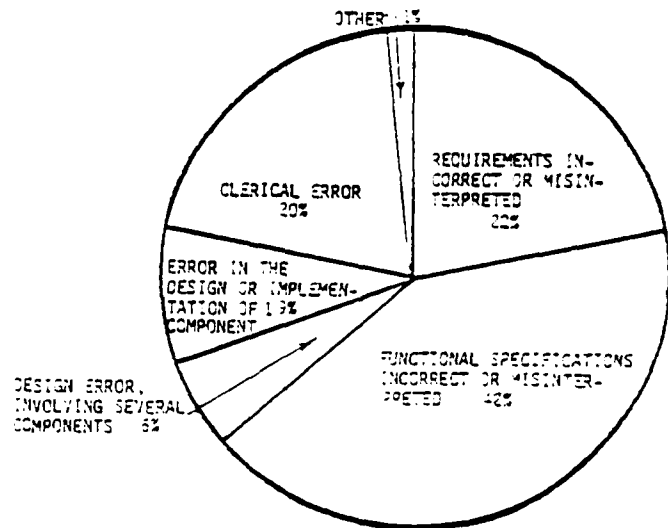


FIGURE 2-21-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 42

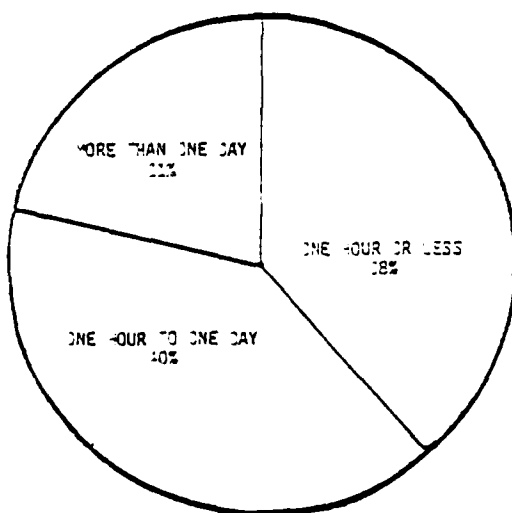


FIGURE 2-21-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 150

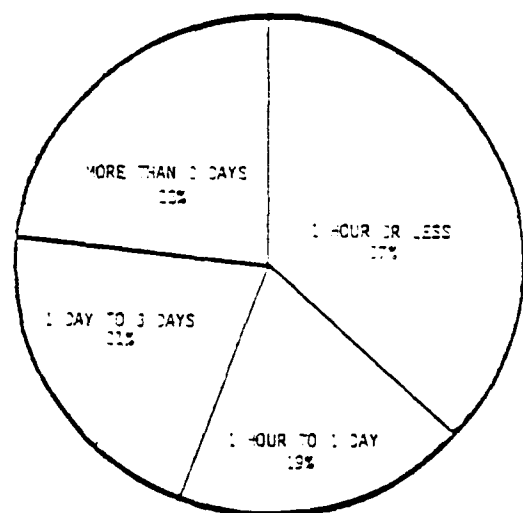


FIGURE 2-21-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 40

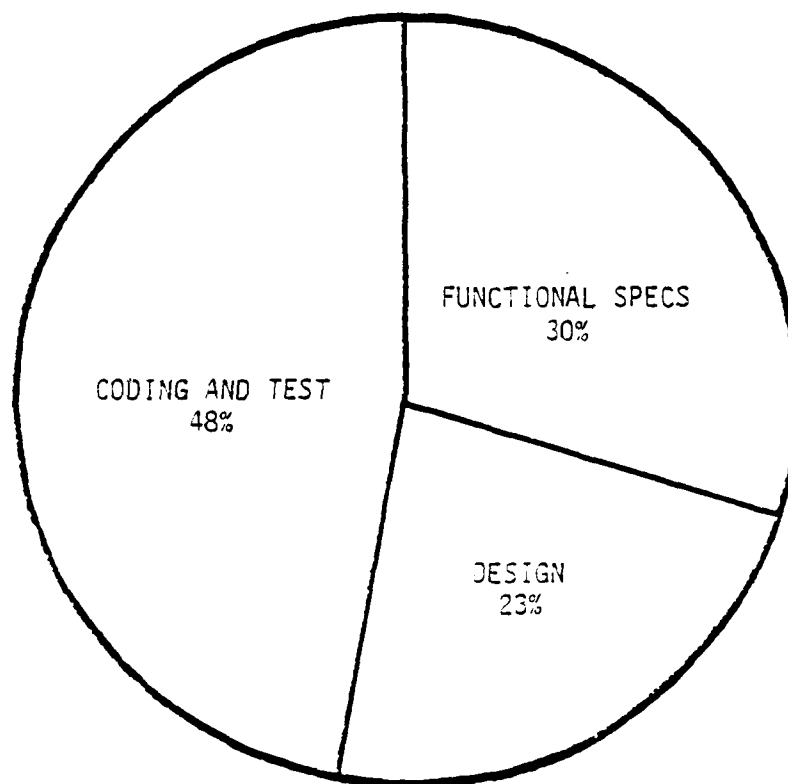


FIGURE 2.21-13

This page was intentionally left blank

2.26 Project 26

Project 26 is a FORTRAN program developed to support attitude determination. The target and development computer for this project was an IBM 360. The system consists of six independent programs. Both functional and English text specifications were utilized to the subroutine level. Top-down design was utilized and iterative enhancement of subroutines was employed during development, using structured code, baseline diagrams, and a PDL. Top-down testing and structure and specification-driven testing were used in validating the system, and code reading and walk-throughs were used during software inspection. The personnel were organized into a team consisting of one chief programmer, one librarian, and four other programmers in addition to four managers.

Data on Project 26 is relatively complete for each category of data. Note that the number of changes recorded under General Project Information (191) from the estimated statistics file is inconsistent with the number of changes recorded under History of Changes (1323) from the project history file.

GENERAL PROJECT INFORMATION

PROJECT : 26

SIZE

DELIVERED LINES OF SOURCE CODE	89513
NEW LINES OF SOURCE CODE	61950
NASA ESTIMATE OF WORDS OF OBJECT CODE	187977
NASA ESTIMATE OF NEW OBJECT CODE	130099
NUMBER OF COMPONENTS	851
NUMBER OF MODULES	604
NUMBER OF NEW MODULES	409
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	125
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	19
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	175

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	14.5
MANMONTHS OF DEVELOPMENT EFFORT	99
NUMBER OF COMPUTER RUNS	7379
SYSTEM 360-95 HOURS	127.3
SYSTEM 360-75 HOURS	183.3
POP 11-70 HOURS	3.3
NUMBER OF CHANGES	191
NUMBER OF CHANGE REPORTS	413

FIGURE 2.26-1

PROJECT 26

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

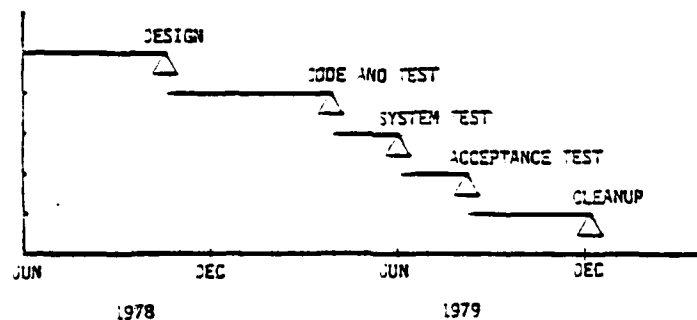


FIGURE 2.26-2

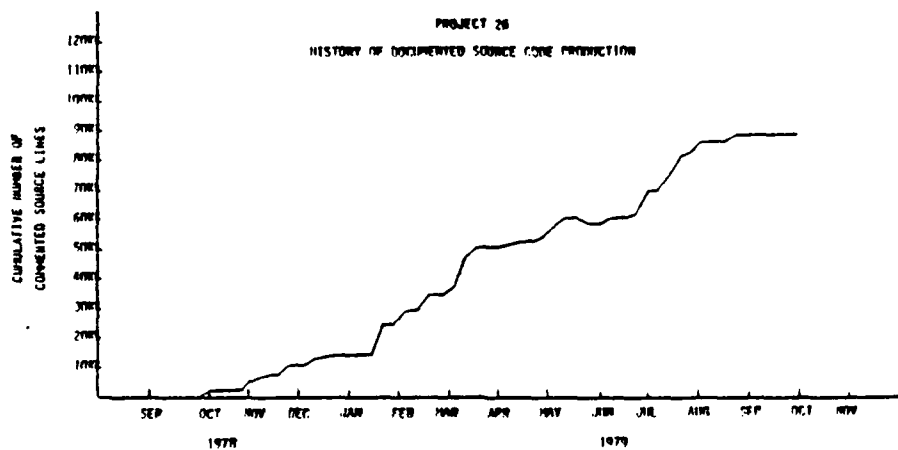


FIGURE 2.26-3

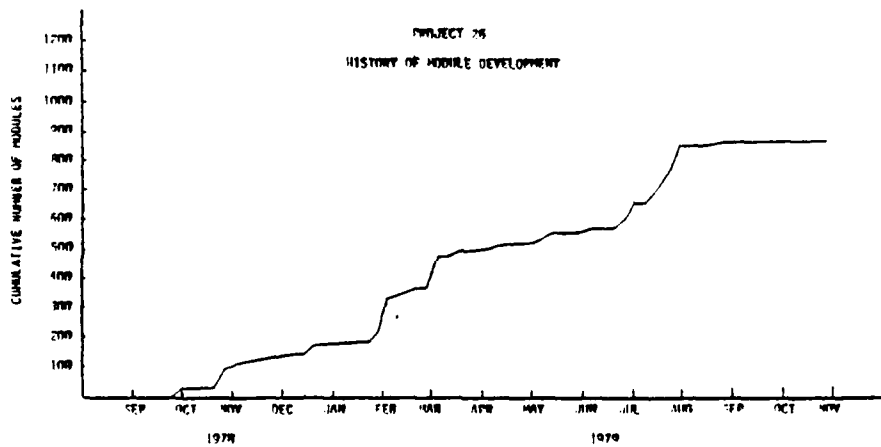


FIGURE 2.26-4

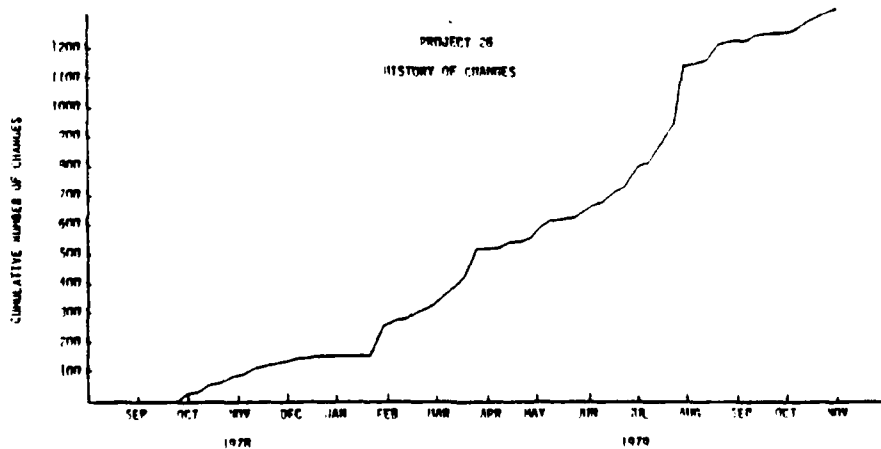
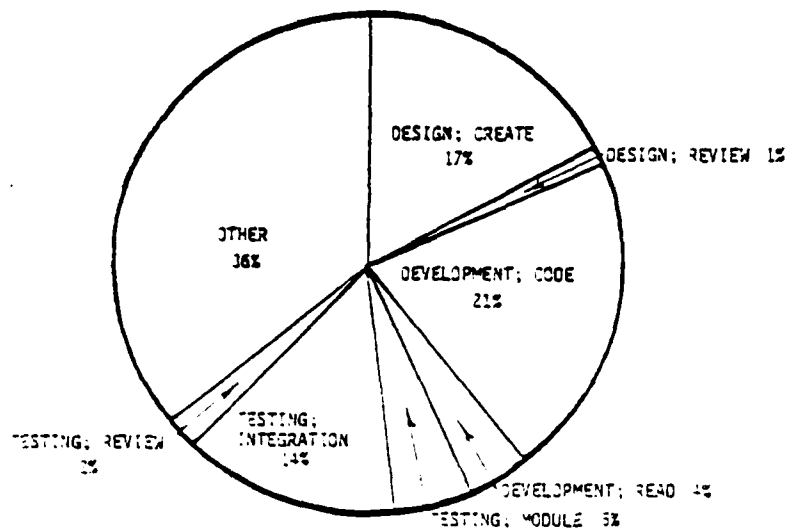


FIGURE 2.26-5

PROJECT: 26

DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)

HOURS THAT THIS DISTRIBUTION IS BASED ON: 12072.7



NOTE THAT LESS THAN 1% OF THE DEVELOPMENT EFFORT WAS SPENT ON READING THE DESIGN AND REVIEWING DEVELOPMENT

FIGURE 2.25-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS

TOTAL RUNS REPORTED: 2321

RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 2432

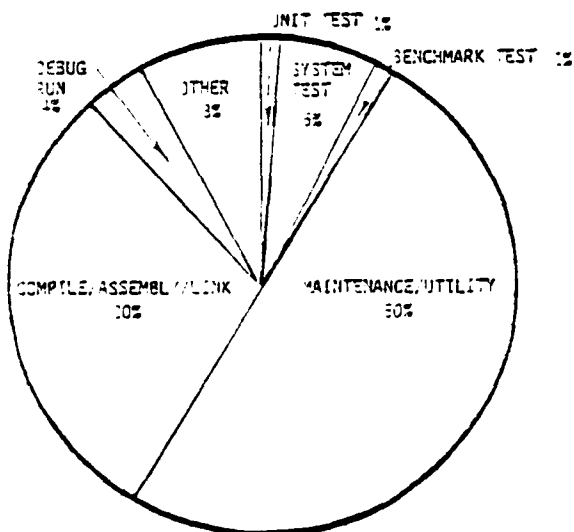


FIGURE 2.25-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS

TOTAL RUNS REPORTED: 2321

RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 31

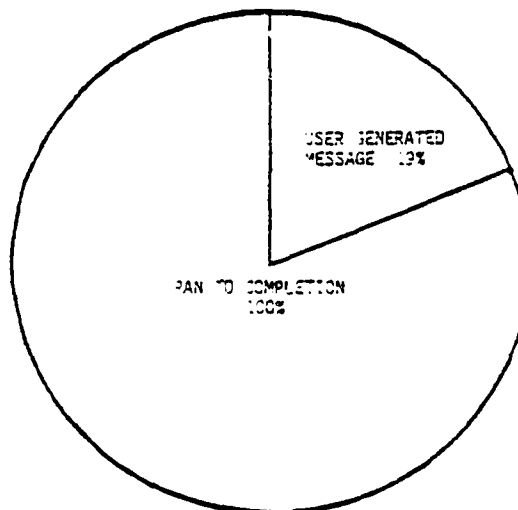


FIGURE 2.25-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 26: 413

DISTRIBUTION OF CHANGES BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 408

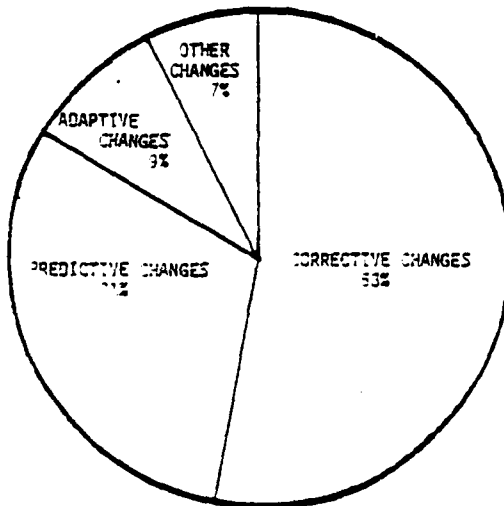


FIGURE 2.25-9

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 219

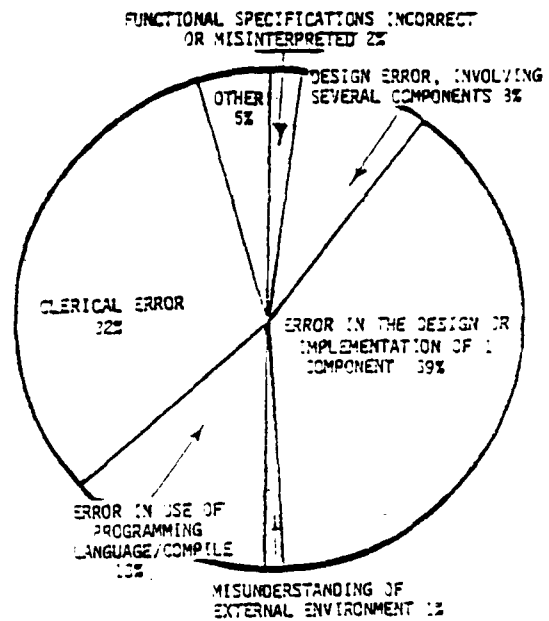


FIGURE 2.25-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 206

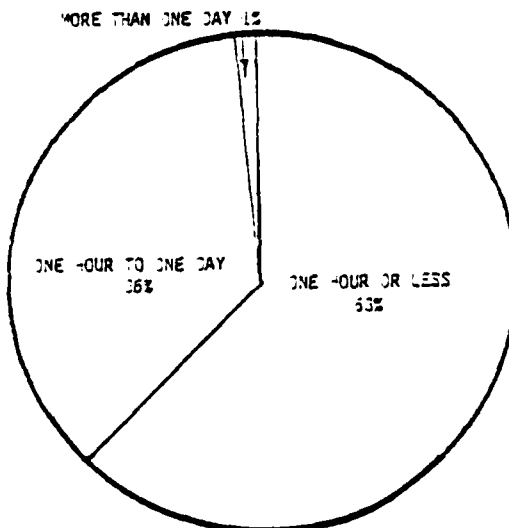


FIGURE 2.25-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 410

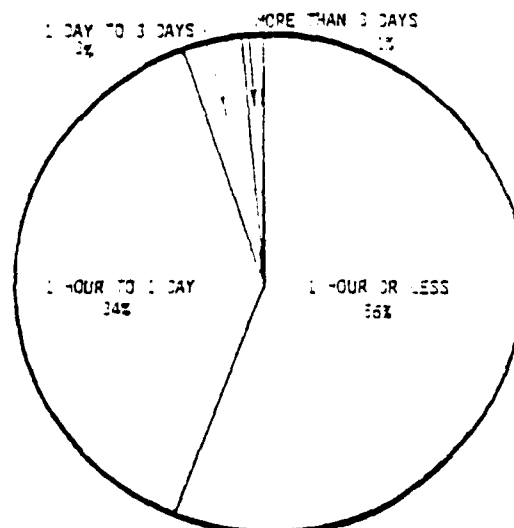


FIGURE 2.25-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 204

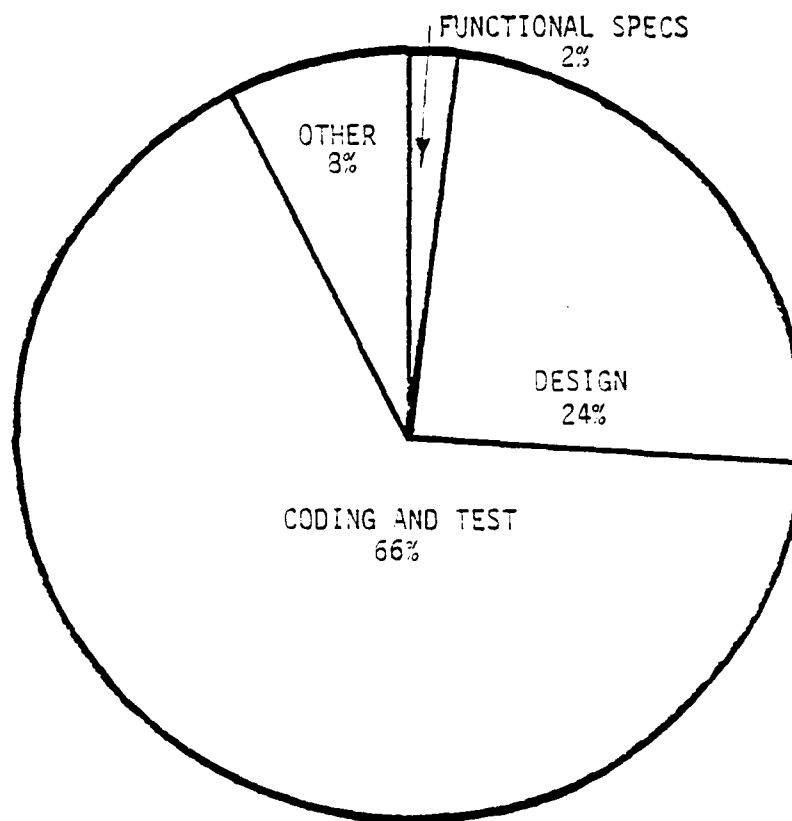


FIGURE 2.26-13

This page was intentionally left blank

2.32 Project 32

Project 32 is a FORTRAN program developed to function as a development and maintenance tool. The development and target computer system for this project was a PDP 11/70. The specifications for this project were in English text at all levels of detail. Top-down design, iterative enhancement and "hardest first" methodologies were used in the design of this project while baseline diagrams were used to specify the design. Top-down and specification-driven testing and code reading and walk-throughs were used in inspection and validation of the project. In addition to a manager, the personnel included one task leader and one programmer analyst. Very little data has been recorded on Project 32.

2.33 Project 33

Project 33 is a FORTRAN program developed to collect and format information for radio transmission. The project was developed on a PDP 11/70 but designed to operate on an IBM 360. Both functional and English specifications were used at the system level. The system was designed by iterative enhancement of modules and developed by the same technique. Baseline diagrams were used in the design of modules. Structured coding techniques were also employed. Modules were inspected by code reading and specification-driven testing was used in the verification/validation of the system. The personnel were organized into a team consisting of a manager and two programmers. Only component information data is available for this project.

2.34 Project 34

No data is recorded for Project 34.

2.35 Project 35

Project 35 is a FORTRAN program developed to extract data from an input file and write it to an output file for processing. The development computer systems were the IBM 360 and the PDP 11/70, but the system was designed to operate on the IBM 360. Procedural specifications were utilized at the system level to specify the design of the software. Top-down design in the form of baseline diagrams and top-down development using a PDL, were other techniques employed. Specification-driven testing was used to test the program, and code was inspected by walk-throughs and code reading. The two programmers assigned to the project were subordinate to one supervisor.

Data on Project 35 is relatively complete in each category of information. However, no history data was recorded for this project.

GENERAL PROJECT INFORMATION

PROJECT : 35

SIZE

DELIVERED LINES OF SOURCE CODE	9126
NEW LINES OF SOURCE CODE	5354
NASA ESTIMATE OF WORDS OF OBJECT CODE	26466
NASA ESTIMATE OF NEW OBJECT CODE	15528
NUMBER OF COMPONENTS	101
NUMBER OF MODULES	74
NUMBER OF NEW MODULES	14
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	27
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	3
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	706

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	3.3
MANMONTHS OF DEVELOPMENT EFFORT	18
NUMBER OF COMPUTER RUNS	
SYSTEM 360-45 HOURS	
SYSTEM 360-75 HOURS	
PDP 11-70 HOURS	
NUMBER OF CHANGES	
NUMBER OF CHANGE REPORTS	103

FIGURE 2.35-1

PROJECT 35

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

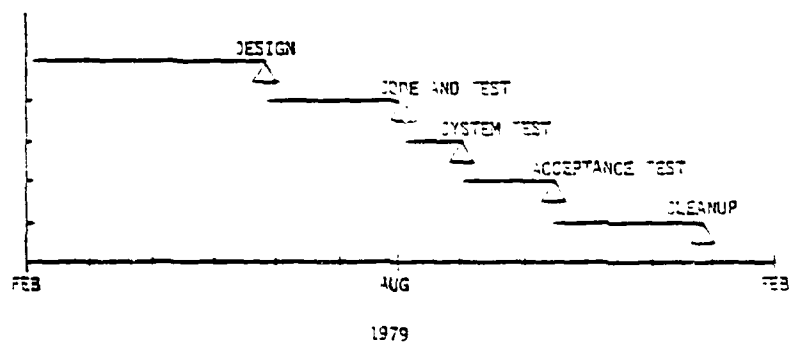


FIGURE 2.35-2

No data recorded for
History of Documented Source Code Production
History of Module Development
History of Changes

PROJECT: 35
 DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)
 HOURS THAT THIS DISTRIBUTION IS BASED ON: 2093.7

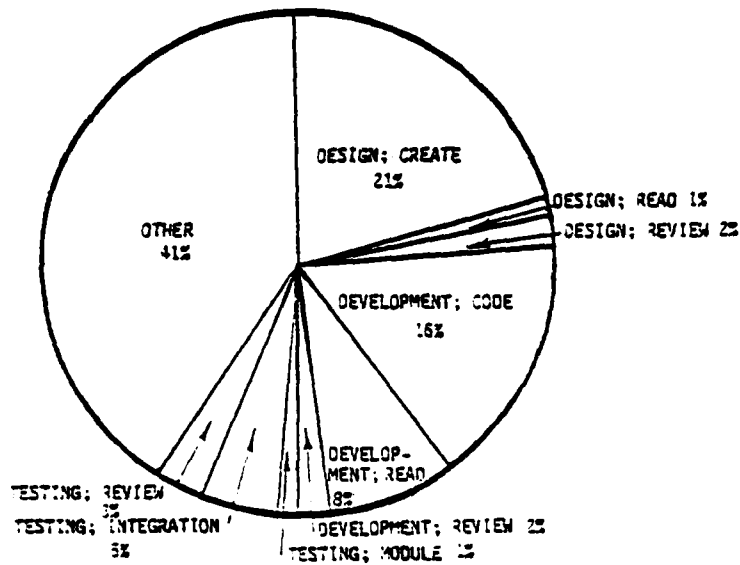


FIGURE 2.35-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS
 TOTAL RUNS REPORTED: 77
 RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 126

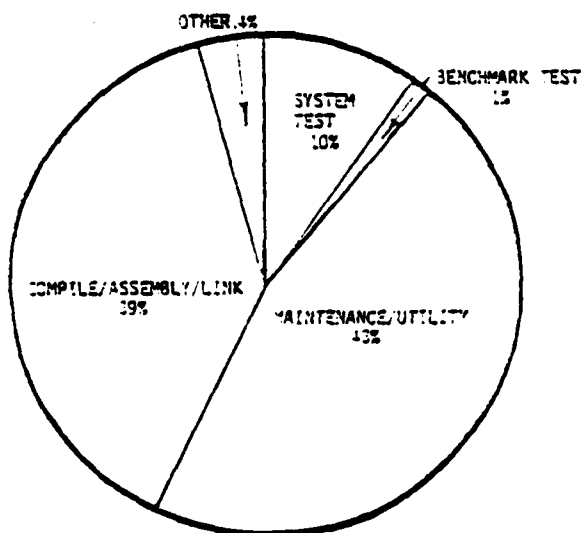


FIGURE 2.35-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS
 TOTAL RUNS REPORTED: 77
 RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 2

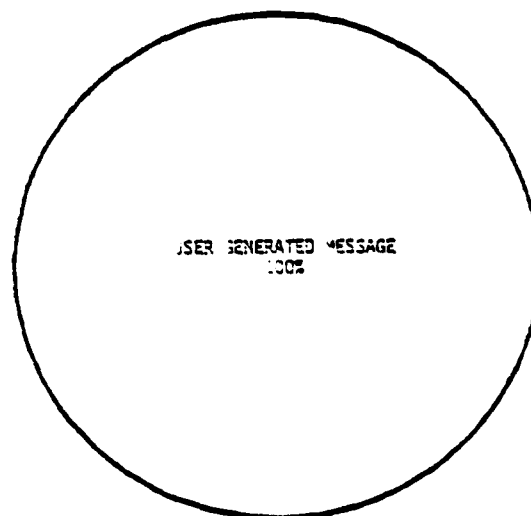


FIGURE 2.35-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 35: 103

DISTRIBUTION OF CHANGES BY TYPE

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 103

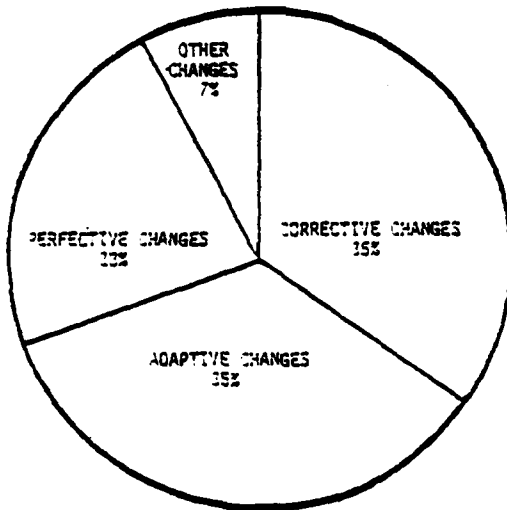


FIGURE 2.35-9

DISTRIBUTION OF ERRORS BY TYPE

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 37

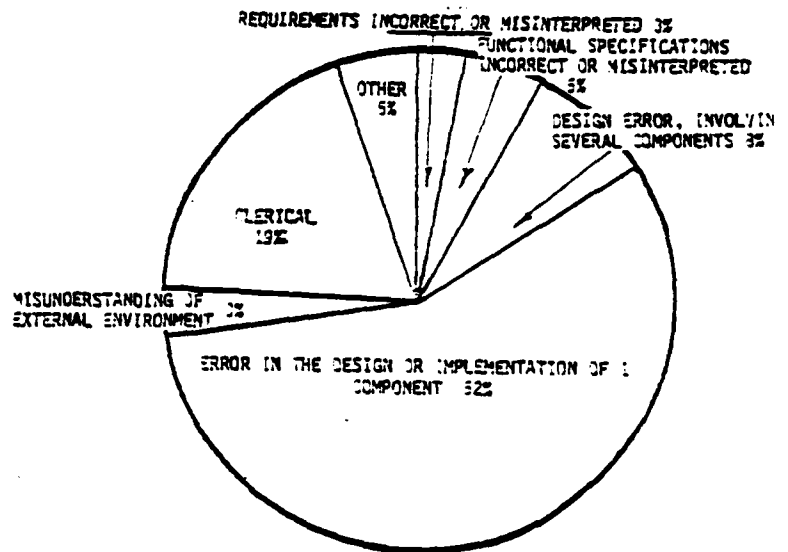


FIGURE 2.35-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 37

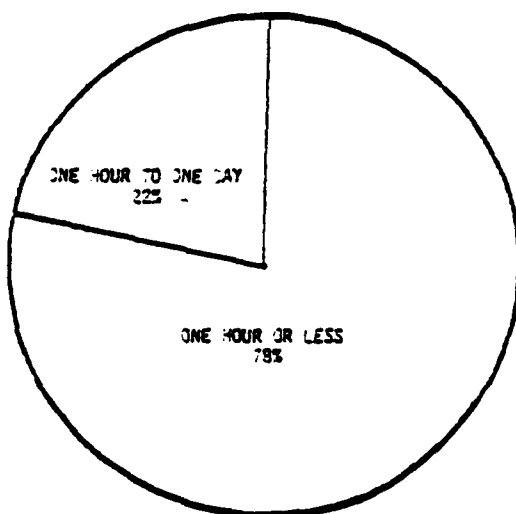


FIGURE 2.35-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES

CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 103

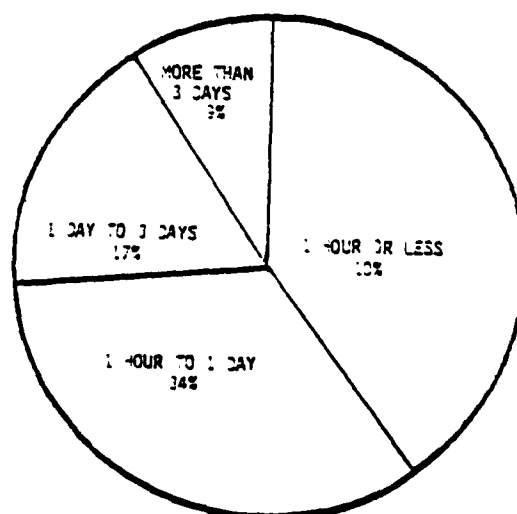


FIGURE 2.35-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 35

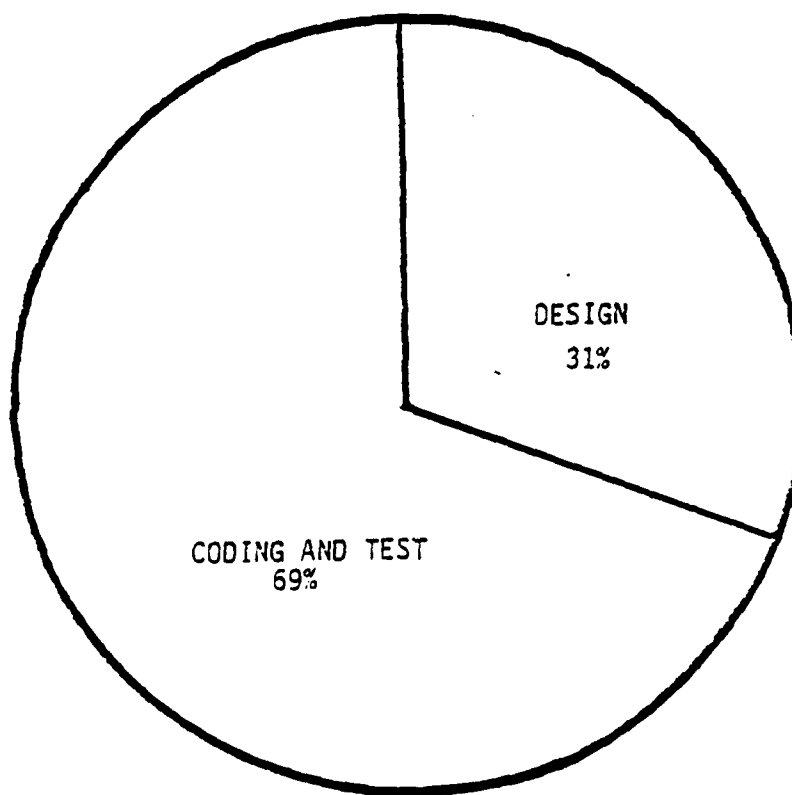


FIGURE 2.35-13

This page was intentionally left blank

2.39 Project 39

There is little information available concerning the development of Project 39 other than the computer used was an IBM 360.

Data has been recorded in all categories except the project's source code and module development and change histories.

GENERAL PROJECT INFORMATION

PROJECT : 39

SIZE

DELIVERED LINES OF SOURCE CODE	10172
NEW LINES OF SOURCE CODE	9627
NASA ESTIMATE OF WORDS OF OBJECT CODE	29499
NASA ESTIMATE OF NEW OBJECT CODE	27918
NUMBER OF COMPONENTS	74
NUMBER OF MODULES	55
NUMBER OF NEW MODULES	45
AVERAGE COMPONENT SIZE (COMMENTED SOURCE)	
MINIMUM COMPONENT SIZE (COMMENTED SOURCE)	
MAXIMUM COMPONENT SIZE (COMMENTED SOURCE)	

DEVELOPMENT

MONTHS OF DEVELOPMENT TIME	15.5
MANMONTHS OF DEVELOPMENT EFFORT	16
NUMBER OF COMPUTER RUNS	
SYSTEM 360-95 HOURS	
SYSTEM 360-75 HOURS	
PDP 11-70 HOURS	
NUMBER OF CHANGES	
NUMBER OF CHANGE REPORTS	17

FIGURE 2.39-1

PROJECT 39

ACTUAL DEVELOPMENT SCHEDULE BY PHASES

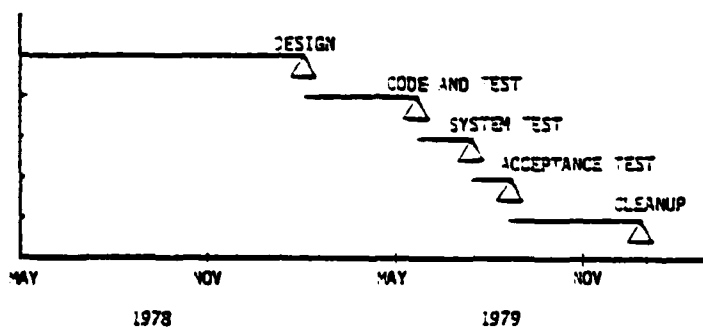


FIGURE 2.39-2

No Data recorded for
History of Documented Source Code Production
History of Module Development
History of Changes

PROJECT 39
 DISTRIBUTION OF DEVELOPMENT EFFORT BY TASK (MANHOURS)
 HOURS THAT THIS DISTRIBUTION IS BASED ON: 1281.8

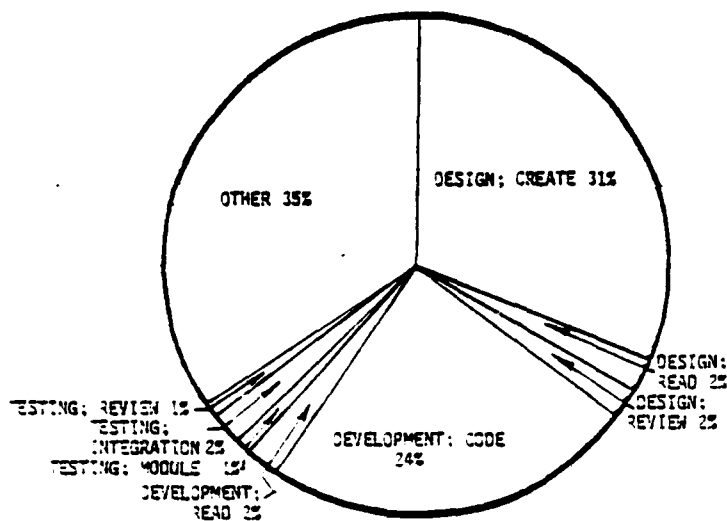


FIGURE 2.39-6

DISTRIBUTION OF PURPOSES FOR COMPUTER RUNS
 TOTAL RUNS REPORTED: 39
 RUN PURPOSES THAT THIS DISTRIBUTION IS BASED ON: 94

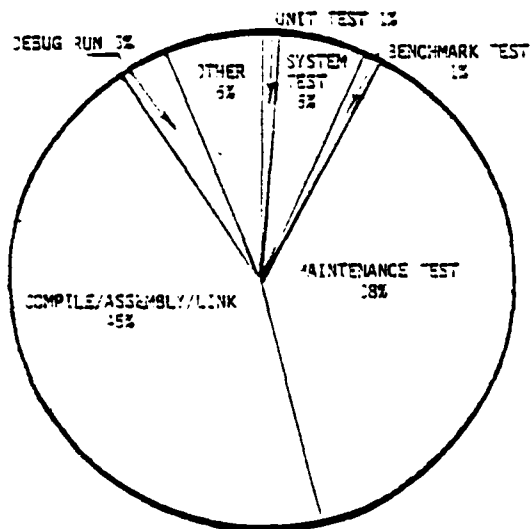


FIGURE 2.39-7

DISTRIBUTION OF RESULTS OF COMPUTER RUNS
 TOTAL RUNS REPORTED: 94
 RUN RESULTS THAT THIS DISTRIBUTION IS BASED ON: 15

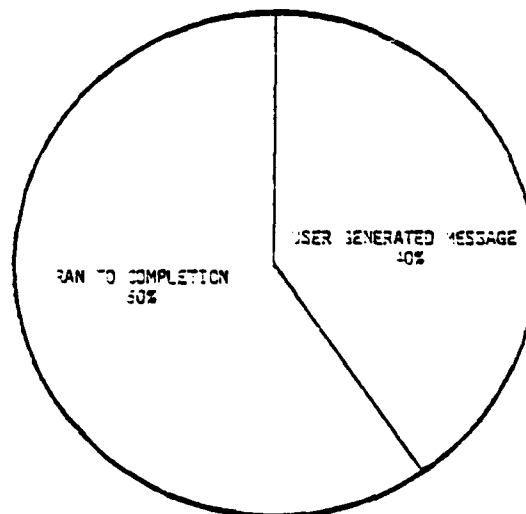


FIGURE 2.39-8

TOTAL NUMBER OF CHANGE REPORTS FOR PROJECT 39: 15

DISTRIBUTION OF CHANGES BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 15

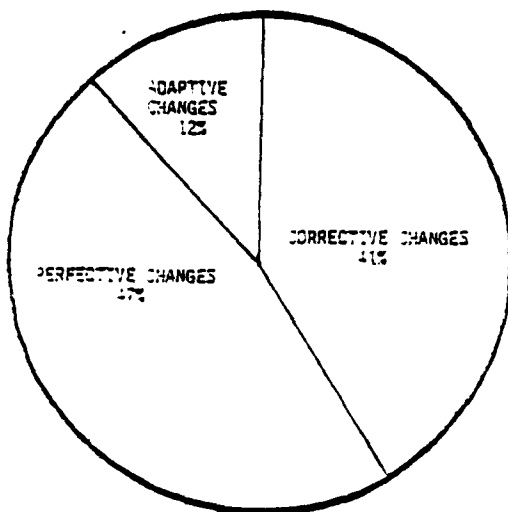


FIGURE 2.39-9

DISTRIBUTION OF ERRORS BY TYPE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 9

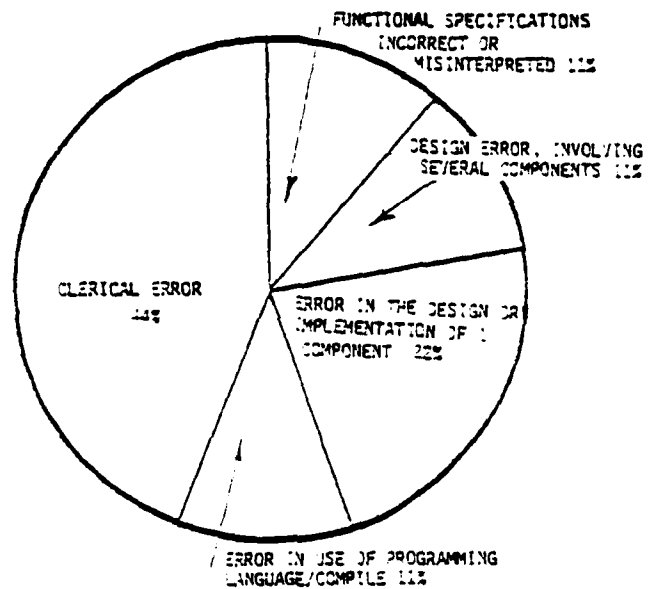


FIGURE 2.39-10

DISTRIBUTION OF EFFORT TO ISOLATE ERRORS
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 7

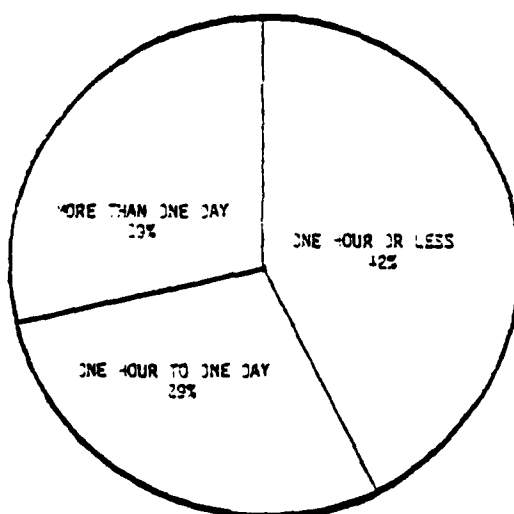


FIGURE 2.39-11

DISTRIBUTION OF EFFORT REQUIRED TO RESOLVE CHANGES
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 15

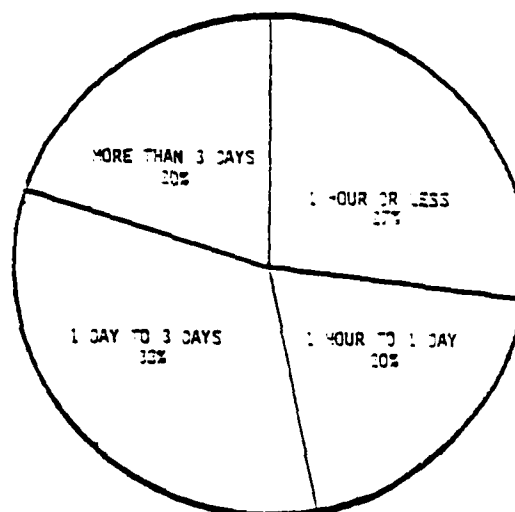


FIGURE 2.39-12

DISTRIBUTION OF WHEN ERRORS ENTERED THE SYSTEM BY PHASE
CHANGE REPORTS THAT THIS DISTRIBUTION IS BASED ON: 7

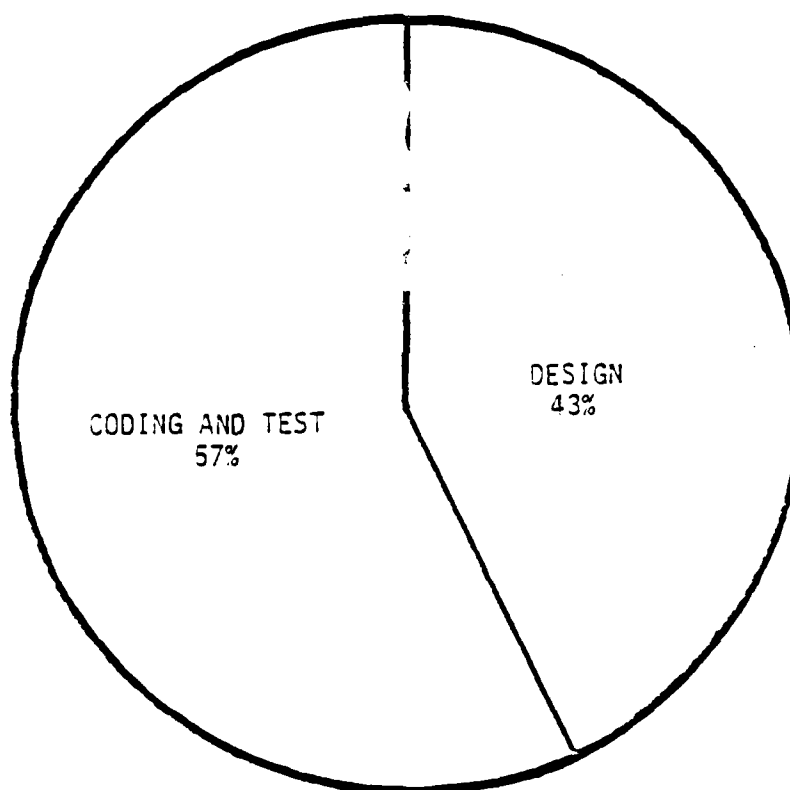


FIGURE 2.39-13

This page was intentionally left blank

3. CROSS PROJECT SUMMARIES

This section consists of a set of three tables (3.1-1, 3.2-1, and 3.3-1) of summary data which summarize the three major categories of data available across all projects. These data categories are:

- Project Development Data
- Change Error Data
- Development Methodology Data

Each of these major data categories has been further subdivided into subcategories representing phasing and scheduling data; human and machine resources; project size, composition and development history; run purposes and outcomes; and finally, the distribution of when errors were introduced into the software, as well as the effort required for correction,

Unfortunately, data results are incomplete for a number of projects. This is because some projects were already well under development when the data collection process was initiated. Data is missing, most notably, in the computer run analysis area and the error analysis area.

3.1 Project History and Development Data

The number of projects for which data is available for each subcategory is as follows:

I. Project Size	19
II. Development Time	18
Development Effort	18
III. Development Time by Phase/Development Effort by Phase	
Design	24/22
Code and Test	23/21
System Test	22/20
Acceptance Test	21/0

IV. Computer Resource Expenditures 14

This data is summarized in Table 3.1-1.

3.2 Change Report Data

Data is relatively complete for 10 of the projects.

The number of projects for which data is available for each subcategory is as follows:

I. Number of Change Reports	21
II. Distribution of Changes by Phase	10
III. Distribution of Why Changes Made	13
IV. Distribution of When Error Entered	18
V. Distribution of Effort to Resolve Change	20

This data is summarized in Table 3.2-1.

3.3 Development Methodology Data

Data is relatively complete for 21 projects. Software development constraints, as well as the Modern Programming Practices, techniques and tools utilized during the development of the NASA software, are listed in the following key to Table 3.3-1.

Table 3.3-1 contains a summary of these special environmental factors by project.

TABLE 3.1-1 GENERAL PROJECT DEVELOPMENT DATA (PART 1)

GENERAL PROJECT DEVELOPMENT DATA															
PROJECT CODE															
	1	2	3	5	6	7	8	9	10	11	13	15	16		
I. Project Size															
Delivered Source Lines	50911	4100	5237	111868	2885	75420	20075		75393	41000	5200	6300	1200		
New Source Lines	45345	4100	43955	94729	2000	20075	20075		49316	41000	3200	6300	1200		
Modified Source Lines	4673		3506	20041	486	6727	486		4252	0	0	0	0		
NASA Estimate of Delivered Object Code	106913	11890	115988	234923	8081	158382	158382		158382	86100	15080	18270	3480		
NASA Estimate of New Object Code	95227	11890	92305	177933	5600	103560	103560		103560	86100	9280	18270	3480		
NASA Estimate of Modified Object Code	98113	0	7363	42086	1361	14127	14127		8929	0	0	0	0		
Total Modules	201	42	283	510	24	374	374		535	200	60	69	12		
New Modules	172	32	200	346	19	92	92		337	200	31	69	7		
Modified Modules	19	10	83	164	5	282	282		98	0	0	0	0		
Number of Components	292	42	355	587	24	423	423		638	200	60	69	12		
Minimum Component Size	1	2	4	4	2	2	2		1	1	1	1	1		
Maximum Component Size	2016	1332	1329	1329	1186	1186	1186		1477	140	140	140	140		
Average Component Size	195	88	88	216	201	201	201		140	140	140	140	140		
II. Project Development Summary															
Months of Development Time	3211.1	16.5	4.0	15.3	19.3	8.5	10.3		14.8	19.3	5.0	7.0	4.0		
Development Hours	8911.5	525.7	12829.8	12852.2	345.7	147.0	4170.6		10956.5	10956.5	150.0	204.1	187.2		
Development Hours	914.0	3676.5	132.0	2331.4	2907.8	38.3	1620.9		3551.0	3551.0	70.0	197.0	49.0		
Other Hours	648.0	1109.0	28.0	1378.0	4316.0	56.0	1079.0		1231.0	1231.0	0.0	77.0	0.0		
Months of Development Effort	28.0	80.0	4.0	100.2	125.0	42.0	40.0		92.0	92.0	1.0	6.0	1.0		
III. Project Development by Phase															
Design:															
Start Date	2/13/77	6/13/77	10/1/76	6/1/76	10/18/77	8/15/77	8/15/77		4/1/77	4/1/76	10/7/77	10/7/77	10/7/77		
End Date	6/4/77	7/30/77	2/26/77	2/26/77	12/3/77	12/3/77	12/3/77		7/30/77	1/26/76	11/4/77	11/4/77	10/29/77		
Calendar Days in Phase	108	48	149	149	188	188	188		121	130	30	46	22		
Percent of Total Schedule	15.5	6.8	21.5	21.5	27.5	27.5	27.5		17.1	20.6	4.6	6.5	3.2		
Design Create, Read, Review Manhours	270.0	1690.6	86.2	2000.8	735.0	159	1260.0		315.0	1615.5	46.5	145.0	50.5		
Code and Test:															
Start Date	6/4/77	12/3/77	2/26/77	2/26/77	10/9/76	12/3/77	12/3/77		7/30/77	7/26/76	11/4/77	11/4/77	10/29/77		
End Date	12/3/77	7/30/77	7/23/77	7/23/77	5/21/77	1/21/78	3/11/78		1/14/78	3/19/77	1/14/78	12/13/77	12/3/77		
Calendar Days in Phase	182	425	147	234	49	90	90		188	232	70	70	35		
Percent of Total Schedule	26.5	32.7	18.7	30.0	6.3	12.5	12.5		27.4	40.0	15.4	9.9	5.0		
Code Develop, Read, Review, and Mobile Test Manhours	522.2	3130.3	141.8	2513.1	2187.2	1681.2	533.5		2346.5	2346.5	92.9	117.2	203.2		
System Test:															
Start Date	12/3/77	7/30/77	7/23/77	7/23/77	5/21/77	1/21/78	3/11/78		1/14/78	3/19/77	1/14/78	12/13/77	12/3/77		
End Date	2/4/78	6/30/77	6/30/77	6/30/77	7/23/77	3/18/78	4/8/78		2/18/78	4/9/77	2/4/78	12/13/77	1/28/78		
Calendar Days in Phase	63	155	28	63	103	225	281.8		35	21	21	84	54		
Percent of Total Schedule	9.1	21.5	3.7	9.1	14.5	35.0	35.0		5.1	4.5	4.6	18.5	10.0		
System Test and Review Manhours	512.5	1718.5	37.5	1232.5	928.0	281.8	74.0		833.0	833.0	44.6	80.5	30.0		
Acceptance Test:															
Start Date	2/4/78	6/30/77	6/30/77	6/30/77	7/23/77	3/18/78	4/8/78		2/18/78	4/9/77	2/4/78	12/13/77	1/28/78		
End Date	3/18/78	9/11/77	9/11/77	9/11/77	9/24/77	3/25/78	5/6/78		4/15/78	5/21/77	2/4/78	12/13/77	2/4/78		
Calendar Days in Phase	42	91	28	63	7	7	28		56	42	7	7	7		
Percent of Total Schedule	6.1	13.0	3.7	9.1	1.0	1.0	3.7		8.1	7.5	0.9	0.9	0.9		
Cleanup:															
Start Date	3/18/78	9/11/77	9/11/77	9/11/77	9/24/77	3/25/78	5/6/78		4/15/78	5/21/77	12/24/77	12/24/77	12/24/77		
End Date	4/29/78	1/17/78	1/17/78	1/17/78	1/6/78	6/30/78	6/24/78		6/24/78	11/15/77	11/15/77	11/15/77	11/15/77		
Calendar Days in Phase	42	91	28	63	103	225	281.8		35	21	21	84	54		
Percent of Total Schedule	6.1	13.0	3.7	9.1	14.5	35.0	35.0		5.1	4.5	4.6	18.5	10.0		
Other Manhours not Recorded by Phase	54.0	1479.0	109.5	518.0	726.0	1570.5	42.5		995.9	995.9	8.5	91.5	28.0		
IV. Computer Resource Expenditures															
System 360-95 Hours	223.0	160.0	156.8	154.0	0.0	52.5	91.0		209.0	98.0	98.0	98.0	98.0		
System 360-25 Hours	160.0	0.0	0.0	0.0	0.0	50.0	76.3		193.0	50.0	50.0	50.0	50.0		
DEC PDP 11-70 Hours	0.0	0.0	0.0	0.0	0.0	168	0.3		7500	9987	9987	9987	9987		
Number of Computer Runs	4604														

TABLE 3.1-3 GENERAL PROJECT DEVELOPMENT DATA (PART II)

[illegible]

TABLE 3.2-1 CHANGE IN ERROR DATA

ERROR AND CHANGE DATA										PROJECT:																															
										1	2	3	5	6	7	8	9	10	11	13	15	16	17	18	19	20	21	22	26	35	37	39	40	41							
I. Summary																																									
Number of Changes																																									
Number of Change Reports										35	290	11	311	491	55	239		1576	260	8	23			2350																	
II. Distribution of Changes Reported During:																																									
Design										1	1		0	4		2		1																							
Code and Test										12	55		31	239	134		36																								
System Test										0	99		101	60	68		4																								
Acceptance Test													0	126	30	5																									
Cleanup										0	2		0	11		0		5																							
Unable to Determine										1	3		119	167	5	1																									
III. Distribution of Reasons Why Changes Were Made																																									
Requirements Incorrect/Misinterpreted													0	0		3																									
Functional Specifications Incorrect													1	4		3																									
Design Error in Several Components													1	0		15																									
Misconception of External Environment													1	16		68																									
Language Usage, Compile Error													4	2		0																									
Clerical Error													1	0		9																									
Other Error													3	4		18																									
Not Reported												1	0	8		5																									
IV. Distribution of When Error Entered System																																									
Requirements Phase										0	1	0	0	0		0		0																							
Functional Specification										1	7	0	0	0		0		0																							
Design Phase										6	47	0	16	28	18	12		6																							
Code and Test Phase										15	112	5	113	101	18	85		19																							
Other Phase										0	1	0	2	3		0		0																							
Unknown (Could Have Been Any)										0	0	0	1	3		0		0																							
Not Reported										12	122	6	179	359	19	140		21																							
V. Distribution of Effort to Resolve Change Report																																									
Less Than One Hour										24	167	5	173	133	45	106		18																							
One Hour to One Day										9	91	3	83	136	10	67		15																							
One Day to Three Days										1	18	1	17	72	0	28		13																							
More Than Three Days										0	0	2	13	19	0	2		1																							
Not Reported										0	14	0	25	131	0	36		0																							

KEY TO SPECIAL ENVIRONMENTAL FACTORS
WHICH INFLUENCED PROJECT DEVELOPMENT

1. A special display requiring new or complex support software, acted as a constraint on development. Y = Yes, N = No
2. A detailed definition of operational requirements aided project development. Y = Yes, N = No
3. The existence of changes made to the operational requirements during development constrained to some degree the development of the project.
N = No constraint, 1 = Very little constraint,
5 represents a large constraint
4. The project was designed for real-time operation. Y = Yes, N = No
5. The project was developed with a constraint on the program processor memory size. N = No constraint, 1 = Very little constraint,
5 represents a large constraint
6. The project was developed with a constraint on the operation time of the project. N = No constraint, 1 = Very little constraint,
5 represents a large constraint
7. The project was the first software developed for a particular computer or operating system. Y = Yes, N = No
8. The project was developed concurrently with ADP hardware necessary for the operation of the software. Y = Yes, N = No.
9. The system used in development was: Time-Sharing = T or Batch = B.
10. The development of the project was constrained by situation of developers having to use a system other than their own. Y = Yes, N = No
11. The project development took place at the operational site. Y = Yes, N = No
12. The development and target computers were different. Y = Yes, N = No
13. The development of the project took place at multiple sites.
Y = Yes, N = No
14. The programmer's level of access to the computer dialog development.
1 = Very limited access, 5 = Unlimited access
15. Form of specifications.
 - A. Functional
 - B. Procedural
 - C. English
 - D. Formal

16. Design techniques.

- | | |
|--------------------------|------------------|
| A. Top-down | D. Hardest first |
| B. Bottom-up | E. Other used |
| C. Iterative enhancement | F. None used |

17. Development techniques.

- | | |
|--------------------------|------------------|
| A. Top-down | D. Hardest first |
| B. Bottom-up | E. Other used |
| C. Iterative enhancement | F. None used |

18. Coding techniques.

- A. Simulated construct
- B. Structured code
- C. Other construct
- D. None Used

19. Testing techniques.

- | | |
|-------------------------|---------------------|
| A. Top-down (stubs) | D. Structure driven |
| B. Bottom-up (drivers) | E. Other used |
| C. Specification driven | F. None used |

20. Inspection techniques.

- A. Code reading
- B. Walk-through
- C. Other used
- D. None used

21. and 22. Design and development formalisms.

- | | |
|---------------|----------------------|
| A. PDL | D. Baseline Diagrams |
| B. HIPO | E. Other |
| C. Flowcharts | F. None |

TABLE 3.3-1 PROJECT DEVELOPMENT METHODS AND CONSTRAINTS DATA

Project	Average Staffing	Social Environmental Factors																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
01	2	N	Y	1	N	4	1	N	N	T	Y	Y	N	N	5	A	C	A	A	A	A	C	E
02	6	N	Y	1	N	1	1	N	N	B	Y	Y	N	N	2	A	A	A	B	C	B	D	E
03	.5	N	Y	1	N	4	1	N	N	T	Y	Y	N	N	4	C	A	A	C	A	A	D	E
05	3	N	Y	1	Y	2	1	N	N	B	Y	Y	N	N	2	A	A	A	D	A	A	D	F
06	3	N	Y	2	Y	3	2	N	N	B	Y	Y	N	N	2	A	A	C	D	A	A	D	E
07	.5	N	Y	1	N	1	1	N	N	T	Y	Y	Y	N	4	D	A	A	B	C	D	A	D
08	5	N	Y	4	Y	2	1	N	N	B	Y	Y	N	N	3	A	A	A	B	A	B	A	D
09	2	N	Y	1	N	1	1	N	N	B	Y	Y	N	N	1	A	A	A	C	A	A	D	E
10	5	N	Y	3	N	4	1	N	N	B	Y	Y	Y	N	2	D	A	A	B	C	D	A	D
11		N	Y		N			N	N		Y	Y		N									
13		N	Y	1				N	N	T	Y	Y		N									
15	1	N	Y	1	N	1	1	N	N	T	Y	Y	N	N	4	B	A	D	B	A	A	A	E
16	.5	N	Y	1	N	1	2	N	N	T	Y	Y	N	N	5	A	A	C	D	C	A	B	F
17	3	N	Y	1	N	1	1	N	N	T	Y	Y	N	N	5	A	C	C	C	D	B	F	
18	.5	N	Y	1	Y	2	2	N	N	T	Y	Y	N	N	5	A	E	A	B	A	A	D	E
19	5	N	Y	5	N	2	2	N	N	B	Y	Y	Y	N	3	A	C	C	A	A	A	A	D
20		N	Y		N			N	N		Y	Y		N									
21	5	N	Y	1	N	1	1	N	N	B	Y	Y	N	N	2	A	A	D	D	A	A	D	E
22	.5	N	Y	1	N	1	2	N	N	T	Y	Y	N	N	5	A	C	E	D	A	A	D	F
25	7	N	Y	3	N	1	1	N	N	B	Y	Y	N	N	1	A	A	A	B	A	A	A	D
29		N	Y		2		N	N		Y	Y		N										
32	1	N	Y	4	N	1	1	N	N	T	Y	Y	N	N	5	C	A	A	D	A	A	D	E
33	2	N	Y	4	N	2	1	N	N	T	Y	Y	Y	N	5	A	C	C	B	C	A	D	E
34		N	Y	4	N	2	2	N	N	T	Y	Y	N	N	2								
35	2	N	Y	4	N	2	2	N	N	B	Y	Y	N	N	3	3	A	A	D	C	A	A	D
39		N	Y		N			N	N	B	Y	Y		N									
40		N	Y	3	N			N	N	T	Y	Y		N									
41		N	Y		N			N	N	B	Y	Y		N									
42	2	N	Y	4	N	2	1	N	N	B	Y	Y	N	N	2	A	C	C	B	C	A	A	D

REFERENCES

1. Gloss-Soler, Shirley A., The DACS Glossary: A Bibliography of Software Engineering Terms, Ordering Number GLOS-1, Data & Analysis Center for Software, IIT Research Institute, Rome Air Development Center, Griffiss AFB, NY, October 1979.
2. Basili, Victor R.; Zelkowitz, Marvin V.; McGarry, Frank E.; Reiter, Robert W.; Truszkowski, Walter F.; Weiss, David L.; The Software Engineering Laboratory, TR-535, SEL-1, NASA Goddard Space Flight Center, Greenbelt, MD, May 1977.
3. Computer Sciences Corporation, Quantitative Software Models, Data & Analysis Center for Software, IIT Research Institute, Rome Air Development Center, Griffiss AFB, NY, March 1979.
4. Basili, Victor R.; "Data Collection, Validation and Analysis," Department of Computer Sciences, University of Maryland.

APPENDIX A
GLOSSARY OF TERMS

APPENDIX A

Glossary of Terms

This appendix contains definitions of the terms used in describing the NASA/SEL software engineering database. It has been compiled from references (1) and (2). The major objective in providing this glossary is to promote consistency in terminology usage among researchers in software engineering.

Phasing and Scheduling:	All of the activities that include an evaluation of a project's requirements, dividing those requirements into specified sets of goals, and making assignments to complete each set of goals.
Design:	A description of how software will be produced to satisfy the project's specifications.
System Test:	The process of trying to find discrepancies between the system and the original objectives.
Acceptance Test:	The testing of the software in the presence of the user to determine if it meets predetermined user requirements.
Cleanup	The preparation of system tapes, formatting of test results, completing documentation, etc. that occurs after acceptance testing. No testing normally occurs during cleanup.
Resource Expenditures:	The value of the resources consumed in the completion of a project. Those resources include human resources and machine resources. Human resources may be divided into three categories: management, programmers, and clerical. Machine resources include the computer time used.
Design Phase:	The creation and recording of the design, including discussion about strategy with peers and the creation of specifications for subcomponents of the current component. This phase also includes a review of decisions made during creation and recording of the design.
Development Phase:	The development and recording of code and in-line comment based on the design. This phase includes the modification of code caused by design changes, errors found in testing, and a review of the work done in this phase.

Testing Phase:	The design of tests, testing strategies, and the running of such tests, for each module and the integration of the modules into the project.
Module:	A program unit that is discrete with respect to compiling, characterized by lexical binding, identifiable proper boundaries, named access and named reference. The word module may apply to a subprogram, program, subroutine, routine, function or macro.
Line of Code:	Seventy-two character card image of source code including comments.
Unit Test:	Testing of a program segment or set of instructions treated logically as a whole.
Maintenance:	The process of modifying existing operational software while leaving it's primary function intact, including detection and correction of errors and the incorporation of modifications to add capabilities and/or improve performance.
Utilities:	Computer programs which provide special services, such as preparing program deck listings, moving files, creating load tapes and plotting output results.
Compile:	To translate a computer program expressed in a problem-oriented language into a computer-oriented language.
Assemble:	To translate a set of some language statements, usually the computers machine language, into the computer's machine code.
Link:	To establish correspondences within a set of code segments which satisfy references between segments.
Debugging:	The process of determining whether or not errors exist, attempting to isolate the source of a problem and finding a solution.
Benchmark:	A standardized computer program used to test the processing power of different computers. Input data, computations to be performed, and the output formats are specified very rigidly.
Successful Run:	A program execution which runs to completion and produces the output expected.

Submit Error:	Occurs when program block, or complex set of code is improperly executed because of misunderstanding involved in the directions for execution.
JCL Error:	An error occurring during the use of the Job Control language or misuse of a procedural operator.
Compile Error:	An error resulting from a misunderstanding of how the compiler operates or an error resulting from the translation of a high order language to a machine based language.
Setup Error:	Error resulting from improper ordering of cards, modules or program blocks in a job deck, or in use of an editor.
Hardware Error:	Error resulting from the breakdown or malfunctioning of the physical component or circuitry in the computer.
Software Error:	A discrepancy between a computed, observed or measured quantity and its true specified, or theoretically correct value, caused by deficiencies or misinterpretations of design criteria, logical mistakes, or syntactical mistakes.
Link Error:	Error resulting from the linking of code segments usually involving transfer of control, label definition and location, or absence of a referenced code segment.
Execution Error:	Error caused by improper use of an algorithm, or improper algorithm for data supplied. Program usually terminates but output is inaccurate.
User Message Error:	Occurs when a run is terminated by the user, or programmer when an error is discovered in execution.
Requirement Definition:	A statement of what the user expects the system to include among its capabilities.
Functional Specifications:	A set of functions defining the output for any input, emphasizing what the program is to do, rather than how to do it.
Change Effort:	Time involved in a modification to design, code or documentation, to correct an error, improve system performance, add a capability, or implement a requirements change.

Design Create Phase:	Writing of component design.
Design Read Phase:	Reading of design by peer to look for errors.
Design Review Phase:	Formal meeting of several individuals for purpose of explaining design. (management review)
Development Code Phase:	Writing executable instructions and desk checking program.
Development Read Phase:	Code reading by peer, similar to Design Read.
Development Review Phase:	Management review of coded components, similar to Design Review.
Test Module Phase:	Module testing - test run with test data on single module.
Test Integration Phase:	Integration testing of several components.
Test Review Phase:	Management review of testing status.
Dataset:	Denotes a collection of data from a source (e.g., a software development project).
Database:	Denotes a collection of datasets compiled for analysis purposes (e.g., software reliability analysis).