AD-A089 678 IIT RESEARCH INST ROME NY F/G 9/2

DATA AND ANALYSIS CENTER FOR SOFTWARE.(U)

JUN 80 L M DUVALL, S A GLOSS-SOLER, J MARTENS F30602-78-C-0255

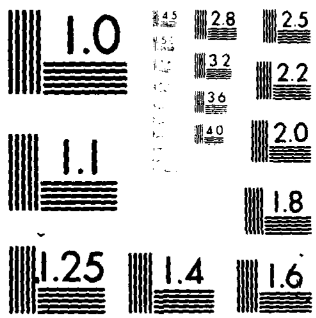
UNCLASSIFIED RADC-TR-80-204 NL

END

DATE FILMED

10-80

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

KV

RADC-TR-80-204
Interim Report
June 1980

LEVEL II

12

DATA AND ANALYSIS CENTER FOR SOFTWARE

IIT Research Institute

Lorraine M. Duvall
Shirley A. Gloss-Soler
Jon Martens

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

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

RADC-TR-80-204 has been reviewed and is approved for publication.

APPROVED:

JOHN PALAIMO
Project Engineer

APPROVED:

WENDALL C. BAUMAN, Colonel, USAF
Chief, Information Sciences Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

SUBJECT TO EXPORT CONTROL LAWS

This document contains information for manufacturing or using munitions of war. Export of the information contained herein, or release to foreign nationals within the United States, without first obtaining an export license, is a violation of the International Traffic in Arms Regulations. Such violation is subject to a penalty of up to 2 years imprisonment and a fine of $100,000 under 22 U.S.C 2778.

Include this notice with any reproduced portion of this document.

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-80-204	2. GOVT ACCESSION NO.: AD-A089678
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): DATA AND ANALYSIS CENTER FOR SOFTWARE	5. TYPE OF REPORT & PERIOD COVERED: Interim Report August 78 - February 80
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Lorraine M. Duvall, Shirley A. Gloss-Soler, Jon Martens	8. CONTRACT OR GRANT NUMBER(s): F30602-78-C-0255
9. PERFORMING ORGANIZATION NAME AND ADDRESS: IIT Research Institute/Beeches Carriage Suite, Turin Road, Rome NY 13440	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63728F, 25280102
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISIS), Griffiss AFB NY 13441	12. REPORT DATE: June 1980
	13. NUMBER OF PAGES: 88
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	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 (ISIS)

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

software engineering
information analysis center
database
scientific and technical information
software technology

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

The Data and Analysis Center for Software (DACS) is being established to serve as a central source for information and data on software technology. This interim report presents the status of the DACS after the initial 18 months of development. Descriptions of the software engineering computer database and the technology information base are provided. This report also contains information on the types of products developed during this reporting period and the technical

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) inquiries received relating to software technology.

PREFACE

This interim technical report summarizes the accomplishments of the Data and Analysis Center for Software (DACS) covering the period August 1978 to February 1980. This effort was sponsored by the Rome Air Development Center (RADC) Air Force Systems Command, Griffiss Air Force Base, Rome, New York, under contract No. F30602-78-C-0255. J. Palaimo of RADC/ISIS served as the Technical Monitor for this effort.

In addition to the authors of this report, project contributions were made by A. Alari, G. Brement, S. Cobb, and M. Nowak from IIT Research Institute. Two state-of-the-art reports were produced by J. Donahoo, C. Feaux, A. Sherer, and D. Swearingen under a sub-contract to Computer Sciences Corporation, Huntsville, Alabama.

TABLE OF CONTENTS

TABLE OF CONTENTS con't

TABLE OF CONTENTS con't

LIST OF FIGURES

LIST OF TABLES

SECTION I

INTRODUCTION

1.1 Background

The need for data and information about computer software, its development process and the software technology area in general, has been widely recognized by the software community. Comprehensive data and information is needed to provide a better understanding of the factors contributing to the cost, productivity, reliability and quality of software, to develop and evaluate better and more efficient methods of producing software, to predict costs of future developments; to determine the best design, development, and testing methodologies; and to develop ways to effectively measure and predict various software factors, such as quality, reliability, testability, maintainability, etc. Project managers need data and information from which baselines and guidelines can be developed to assist them in planning, measuring and tracking of software development projects. Furthermore, with the rapid expansion of software engineering technology and the proliferation of tools and techniques, it has been extremely difficult for an individual or organization to remain cognizant of the current status of the software engineering field. This situation has resulted in duplication of efforts in software research and has seriously hampered the transfer of technology from the software research environment to the user sector of the software community.

The Air Force recognized the need for an information analysis center to serve the government, industrial, and university community as a focal point for software development and experience data. The Rome Air Development Center (RADC) contracted with IIT Research Institute (IITRI) to design such a center that would acquire, analyze, synthesize, and disseminate information on software engineering technology, (reference 1). Subsequently, in August of 1978, RADC contracted with IITRI to develop such a center, named, The Data and Analysis Center for Software (DACS). It is the purpose of this report to record the activities, accomplishments, and status of the development of the DACS during its first 18-month period (August 1978-February 1980).

The DoD and other Federal Agencies have found that the establishment of special purpose information analysis centers and technology transfer programs has been effective in overcoming problems in technology implementation and diffusion of mission-oriented developments. NASA, for example, has been able to demonstrate a benefit-to-cost ratio in excess of ten to one for its Technology Utilization Program. Similarly, the DoD has sucessfully operated Information Analysis Centers for Metals, Ceramics, Hardware Reliability, and Machinability data, (reference 18).

1.2 Objectives of the Center

When fully implemented and operational, the DACS will provide a centralized source for current, and readily-usable data and information concerning software technology. The objectives of the software information resource are:

- Encourage the diffusion of technology to DoD, Civil Agencies, Government contractors, etc.
- Bring about higher levels of utilization of project results in a cost effective manner.
- Increase the productivity and quality of computer software by improving the transfer of software engineering technology.
- Assist in diffusing new technology throughout the U.S. industrial base, thereby expanding its capability and competitive posture.
- Provide scientific and technical information analysis services to DoD, Civil Agencies, government contractors, and the private sector in areas relating to software technology needs, developments, and trends.
- Minimize duplication of effort thereby reducing costs.

1.3 Report Contents

This report provides a summary of the development of the DACS with these objectives in mind and contains eight sections and three appendices. The development activities of the DACS were oriented toward building an information base on software engineering technology and transferring information about the technology in a form that can readily be understood, evaluated, and used.

The following is a short description of the various areas covered in the specific sections.

Section I	Background and objectives of the Center
Section II	Summary of Technical progress and activities
Section III	Descriptions of the DACS Database and data and information synthesis activities
Section IV	Composites of the newsletters, information publications, and technical symposia activities
Section V	Descriptions of the software engineering technical information base including the DACS library, the bibliographic database, the DACS Thesaurus, and the DACS Glossary
Section VI	Discussions of the various DACS products and services including the data subsets, a data compendium, technical reports, and technical inquiries
Section VII	The results of a DACS user-survey and analysis of inquiries processed
Section VIII	Conclusions and recommendations
Appendix A	A partial list of DACS Users
Appendix B	Statistics on the utilization of the keywords in the DACS Thesaurus
Appendix C	A concept paper discussing the role of the DACS in transfering software engineering technology

SECTION II

TASK 1 ESTABLISH AND OPERATE CENTER

2.1 Summary of Technical Progress and Activities

2.1.1 Approach

The Data and Analysis Center for Software is being developed on a build approach basis. That is, an information base is being established and used to provide incremental additions of products and services to demonstrate the center's immediate and growing effectiveness. This build approach is highlighted by dividing the descriptions of the progress to date and the progress planned into three phases.

2.1.2 Summary by Phase

Table 2.1-I lists the major milestones for each phase of the development of the DACS. Phase I covers the 7½-month period from the initiation of the contract (15 August 1978) to 1 April 1979. This phase included establishing communication channels with the software technology community, generating the procedures (both manual and automated) for developing the technology information base, initiating this development, producing a state-of-the-art report on quantitative software models, and answering 91 informational, technical, and document inquiries.

Phase II (1 April 1979-15 February 1980) included a continuation of many of the activities performed under Phase I, as illustrated in Table 2.1-I. In addition to these continued activities, a DACS Glossary, a set of data collection forms, a state-of-the-art report on software maintenance, and a data compendium were produced. The total number of inquiries received and processed during this 10-month period was 2,162.

By the end of Phase I and Phase II (which is this reporting period) we have accomplished the following objectives:

- Provided an awareness of the DACS to the software technology community.
- Produced products and provided services that were of immediate use to the community.
- Established a technology information base to be used as a framework for analysis, consolidation, and synthesis.

Phase III of this development of the DACS (15 February 1980-15 August 1981) will concentrate on providing more quality and indepth products and services. An active data acquisition program will be initiated, and a cost recovery program will be developed. The technology information base will be expanded and more engineering/data analysis will be performed to provide synthesized information to researchers, managers, and software developers.

TABLE 2.1-I CENTER ACTIVITIES BY PHASE

Phase I (15 August 1978-1 April 1979) 7½-Month Period	Phase II (1 April 1979-15 February 1980) 10½-Month Period	Phase III (15 February 1980-15 August 1981) 18-Month Period
Established physical facilities. Established mechanisms for processing requests. Initiated document processing.	Refined procedures. Distributed and analyzed the DACS User Survey.	Continue refining procedures. Develop and implement an active data acquisition program. Develop a cost recovery program.
Compiled Newsletter mailing list. (2143 names/organizations)	Updated Newsletter mailing list. (2619 names/organizations)	Maintain Newsletter mailing list.
Generated Thesaurus.	Produced the first DACS Glossary.	Produce the second DACS Glossary.
Prepared and distributed six monthly Newsletters.	Prepared and distributed three quarterly Newsletters and eight Bulletins.	Prepare and distribute six quarterly Newsletters and twelve Bulletins.
Produced the Data Parameters Report.	Produced and distributed a set of productivity data collection forms.	Establish project status capability.
Organized the IEEE Terminology Task Group.	Coordinated activities of the IEEE Terminology Task Group.	Coordinate production of the IEEE terminology document.
Prepared and presented introductory papers on the DACS	Prepared and presented papers on the DACS, quantitative software models, database evaluation, and software reliability.	Continue presenting papers, etc.
Initiated the acquisition of the NASA/SEL database.	Implemented the NASA/SEL database on the RADC Computer. Produced a data compendium based on this database. Distributed copies of the RADC Productivity Database.	Implement, analyze and distribute databases as received. Produce the second data compendium.
Designed and implemented the datafiles for the Glossary, Thesaurus, and bibliographic database. Generated computer queries.	Refined and distributed the DACS Glossary and the Thesaurus. Produced custom bibliographic searches.	Continue maintaining technology information base. Prepare two comprehensive software engineering bibliographies.
	Produced and distributed the users guide, "Bibliographic Services, Custom Searches" (BIB-I).	Perform engineering research, distill and synthesize bibliographic and database information
Produced SOA Report "Quantitative Software Models" (SRR-1).	Produced SOA Report "Software Maintenance Technology."	Produce one SOA Report and two technical monographs.
Processed 95 technical and document inquiries.	Processed (as of 1 February) 281 technical inquiries, 1212 document inquiries, and 669 Newsletter inquiries.	Continue processing technical and document inquiries.

The remainder of this report provides discussions on the DACS software experience database, the DACS technology information base, the products produced and the services rendered.

SECTION III

DATABASE AND TECHNOLOGY ANALYSIS

3.1 Introduction

This section of the report contains descriptions of the DACS software experience database and the results of analyses performed in the areas of data collection, database evaluation, quantitative software models, and software maintenance.

3.2 Task 2 Computer Database

3.2.1 Introduction

The DACS Computer database presently consists of four sets of data distinguishable by data source, data collection and acquisition methodology, and data contents. These four sets of data are described in subsequent sections of this report and are termed:

- Baseline Software Database
- RADC Productivity Database
- Software Reliability Database
- NASA SEL Database

These databases have been implemented on the RADC HIS 6180 computer system using the Management Data Query System (MDQS) for data retrieval and analysis and for producing subsets of the data.

3.2.2 Baseline Software Database

This DACS database consists of six distinct datasets which contain data describing software problem reports acquired by RADC from six large software development efforts. These projects and the data are described in references 2 through 7.

The system application areas encompass command and control, real-time control for land-based radar, onboard guidance and navigation, and database management. The majority of the datasets contain the date the software problem was opened and closed, the module that manifested the problem, the module that was changed to correct the problem, the problem category and priority, and the correction type.

Three of the datasets contain module descriptive information which designates the number of source instructions, the programming language, the type of construction, and a functional area designation. One dataset records information on each test run.

This database presently consists of 26,594 Software Problem Report records, 2719 Run Analysis Report records, and 2591 Module Description records. References 8 and 9 describe the implementation of these datasets on the RADC computer system.

3.2.3 RADC Productivity Database

This database was compiled by Richard Nelson of RADC and contains summary information from over 400 software projects, some dating back to the early sixties, mostly on DoD applications. The Database incorporates productivity data, error data, project duration, total effort, language data, and information on the usage of various software implementation technologies. At present, 93% of the projects contain productivity data, 7% contain error data, 91% contain language usage data and about 50% contain information on implementation technologies. In total, this database covers approximately 21 million lines of code. The data has been collected from a large variety of sources, both public and private.

A preliminary analysis and a description of this database is contained in reference 10.

3.2.4 Software Reliability Database

This database was provided to the DACS by John D. Musa, Bell Telephone Laboratories, Whippany, NJ, with the hope that it will be of general benefit for software reliability researchers and that it will permit independent verification of his conclusions on the execution time theory of software reliability, (reference 11).

The primary objective for collecting this data on 16 software systems was to assist software managers in monitoring status and predicting schedules. Careful controls were employed to ensure that the data would be of high quality.

Failure interval data (failure number, failure interval, day of failure) is provided for all 16 systems with a total sample size of 2831. Failure correction dates and resource expenditure information is available for four of the systems.

The size of the software systems vary from approximately 6,000 to hundreds of thousands delivered object instructions. The nature of the systems are real time command and control, real time commercial, operating systems, a time sharing system, and a word processing system. Operational failure data is available from 11 of the systems, system test failure data from eight of the systems, and subsystem test failure data from one of the systems. Four of these datasets contain failure and resource expenditure data for both the system test and operational phases of the software life cycle.

3.2.5 NASA SEL Database

The Software Engineering Laboratory (SEL), at NASA Goddard Space Flight Center, was organized in August 1976 to monitor existing software methodologies and to develop and measure the effectiveness of alternative methodologies, (reference 12). To accomplish these objectives, they have been collecting data during the development of their software products and have submitted their data to the DACS for subsequent analysis and distribution. This database will continue to be updated as more data is collected.

As of 1 February 1980 the NASA SEL Database at the DACS contained information on 29 software development projects. These projects include orbit and attitude determination systems, mission support packages, attitude ground support systems, a human resources scheduling system, a FORTRAN source analyzer, and other software support systems. The computer programs were written in FORTRAN or a combination of FORTRAN and assembly language for a PDP 11/70 or a System 360.

The majority of the data is from run analysis reports (over 10,000 records). The remainder is information from component status reports (3760 records), project comment information (2994 records), change reports (1172) records), resource summary reports (262 records), and component summary reports (208 records).

3.3 Task 3 Data Analysis Program

3.3.1 Introduction

The major activity performed under this task definition was the development of a database evaluation methodology. This methodology was developed and used to determine the relevance and limitations of the Baseline Software Database to perform research in software error analysis and reliability modeling.

3.3.2 Database Evaluation

Quantitative metrics to measure and evaluate the quality of existing and future data for modeling and analysis efforts were developed. These metrics will facilitate data collection efforts and modeling activities.

The metrics developed are named integration and coverage. Integration ($\underline{I}$) is a numeric measure of the similarity between the datasets in a database. A high degree of similarity ($\underline{I} \approx 1$) facilitates comparisons of results of a given model among different datasets in a database. Coverage is a graphical metric which measures what model can be driven by the data contained in a database. In this technique, the cells of a two-dimensional array (whose axes are (x) model and (y) data parameter) are filled with the name of the dataset and data element that can supply the data parameter for the model. Matches between data parameters and data elements can then be determined.

These metrics were used to determine the adequacy of the data in the Baseline Software Database to do comparisons across a wide variety of projects and to determine if the database contains data elements as required by the various software reliability models and other analytical techniques. A complete discussion of this methodology and the results of the evaluation of the Baseline Software Database is included in reference 13.

The evaluation determined that the integration factor for the Baseline Software Database was $\underline{I} = 0.28$, a low value that indicates that this database is not well suited to cross comparisons among datasets. Database coverage is high for models that utilize software problem and software modification data, but there are a variety of error analysis and reliability models that cannot fulfill their data requirements from the available data elements.

3.4 Task 4 Data Requirements

3.4.1 Introduction

This section of the report summarizes the results of a study to determine the kinds of data required to utilize software models and assist modeling research efforts. Also included is a description of data collection forms which were developed to provide a basis for collecting software productivity information.

3.4.2 Data Parameters Report

A study was performed to determine the data parameters needed to support research activities, model developments, and analysis efforts in the areas of software research (software reliability, software error analysis, life-cycle cost analysis, software productivity, and complexity). A report was produced that contains descriptions of software models and the kinds of data required for each model. The model descriptions include:

- Category of the model
- Sources referenced
- Brief statement of the purpose and characteristics of the model
- An enumeration of the significant metrics
- Formulas or other outputs of the model

The model descriptions for life-cycle cost/productivity, reliability/error analysis, and complexity are contained in three separate sections. Each section ends with a matrix which cross references the models and their parameters.

3.4.3 Productivity Data Collection Forms

Two data collection forms were designed for software life-cycle productivity-related data. The first form, the Project Summary Form, describes a project's software development environment. The second form is the Component Summary and describes the software components of a project, resources expended during its development, and any noted discrepancies. Component data may be collected at several levels such as System, Functional Group, and Module. In general, the forms have been designed to identify key data collection areas yet be flexible enough to accommodate additional project-specific data.

The Project Summary Form is used to collect data concerning (1) the general nature of the project, (2) the project priorities and constraints, and (3) the software development technologies used in the project. The data elements of the project name, project type, project description, number of systems, number of functional groups, number of modules and standards record the general nature of the project. Priorities and constraints may be chosen from several common ones listed on the form, or project-specific entries may be noted. A list of software development technologies is also provided. Additional entries may be added.

The Component Summary Form is used to collect data concerning (1) the software engineering characteristics of the component, (2) the resources expended throughout the life cycle of the component and (3) the discrepancies detected in the component's life cycle. Complexity, type, description, size, programming language, and mode of construction describe the component in software engineering terms. The resources expended on the component are recorded by life-cycle phase and time in terms of person-hours and computer-hours. The number of discrepancies found in the component are recorded by life-cycle phase.

The purpose of these forms is to collect project and component level data that is common to most software development projects. If these forms are to be used as a tool for collecting data for a specific quantitative software engineering model or method, additional project-specific forms should be developed. To develop these special forms, DACS Software Engineering Research Review - Quantitative Software Models, SRR-1, should be used to determine the data parameters to be collected for the specific application.

These forms and their instructions are being distributed by the DACS to invite feedback from the software research and development community.

3.5 Task 5 Input Processing - Software Data

This task consisted of processing the Software Reliability and NASA/SEL Databases. These databases were reviewed and evaluated, loaded onto the RADC Honeywell 6180 Computer System, and defined using the data definition languages of MDQS.

SECTION IV

TASK 6 CURRENT AWARENESS PROGRAM

4.1 Introduction

The purpose of this task is to inform users of the services of the Center and to provide the software community with up-to-date information on software technology. Six monthly newsletters, three quarterly newsletters, and eight bulletins were produced and distributed during the year and a half period. Descriptive information on the DACS was prepared and distributed in response to requests for information on the Center. Sixteen presentations were made at conferences, workshops, and technical meetings to discuss the DACS and summarize technical results. These activities are discussed below.

4.2 Newsletters and Bulletins

The contents of the DACS monthly newsletters, quarterly newsletters, and bulletins are summarized in Tables 4.2-I, 4.2-II, and 4.2-III, respectfully. The newsletters and the bulletins are both informative and promotional. They have contained information on software technology activities, report reviews, synopses of symposia, referral information, and descriptions of DACS products and services. As of 1 February 1980, there were 2619 names on the newsletter mailing list. The bulletin is distributed to approximately 50 individuals.

4.3 Information Publications

Literature on the DACS products and services has been prepared and distributed to users and potential users of the center. A general information packet is sent in response to an initial inquiry on the DACS. This packet contains a copy of the paper entitled "An Introduction to the Data and Analysis Center for Software," individual flyers which contain order forms and describe the various products, and the guide entitled " Bibliographic Services, Custom Searches." This guide (reference BIB-1) was designed to acquaint the users of the DACS with the procedures required to obtain a custom search of the DACS bibliographic data files. As of 1 February 1980, 817 copies of BIB-1 had been distributed.

4.4 Technical Symposia Activities

Sixteen technical presentations were made at software related technical symposia, and are listed in Table 4.4-I. This list includes seven papers that were published in the workshop or conference proceedings. The major subject areas included in these papers are:

- An introduction and status of the DACS
- A summary of the DACS report entitled "Quantitative Software Models"
- The state-of-the-art in software reliability
- A review of the DACS database evaluation methodology
- Software engineering technology transfer

TABLE 4.2-I MONTHLY DACS NEWSLETTERS

	Date	Information Highlights
1978	October	Introductory information on the DACS
	November	Synopses of three software technology workshops
	December	A discussion on six of the DACS datasets, and reviews of two software reports related to software data collection and software reliability
1979	January	Definitions of software reliability terms, a discussion on software reliability theory, and a book review on software metrics
	February	Summaries of two software related facilities, descriptions of software engineering committees, and synopses of two symposia
	March	Descriptions of the DACS bibliographic services and SRR-1, and a profile of DACS inquiries

TABLE 4.2-II QUARTERLY DACS NEWSLETTERS

	Date	Information Highlights
1979	June	Three document reviews on computer program and database conversion, summary statistics from the RADC Productivity Database, and workshop announcements
	September	Information on obtaining the RADC Productivity Database and the DACS data collection forms, on surveys being conducted on software tools, on the availability of several software guidebooks, a summary of the DACS bibliographic search services, and the DACS User Survey.
	December	A discussion on the history and status of the DACS, a listing of products, an annoucement of the DACS glossary and the software reliability database, summaries of two conferences and one report, and a listing of centers and services that catalog and/or distribute computer programs

TABLE 4.2-III DACS BULLETINS

Date		Information Highlights
1979	April	The results of a custom bibliographic search on software development tools and techniques
	May	A discussion of the DACS software engineering database evaluation methodology
	July	The DACS Productivity Forms with instructions
	August	Announcement of the NBS and AIAA software tools surveys
	October	A synopsis of the Second Minnowbrook Workshop on Software Performance Evaluation
	November	History and status of the DACS
1980	January	A summary of GAO Reports related to software technology
	February	A review of two reports on software quality metrics

TABLE 4.4-I TECHNICAL PRESENTATIONS AND PAPERS

DATE	AUTHOR(S)	TITLE	PRESENTATION
January, 1980	Lorraine Duvall, Jon Martens	Data Needs for Software Reliability Modelling	Proceedings 1980 Reliability and Maintainability Symposium January 22-24, 1980
December, 1979	Lorraine Duvall	Technology Dissemination	AIAA Computer Conference Washington, DC December 3-5, 1979
November, 1979	Jon Martens, Lorraine Duvall	Designing a Software Data Compendium	Fourth Software Engineering Workshop, NASA Goddard Space Flight Center Greenbelt, MD November 19, 1979
	Shirley Gloss-Soler	Data & Analysis Center for Software	Association for Computing Machinery Rochester, NY November 14, 1979
October, 1979	Lorraine Duvall	Software Reliability Program Concepts	National Electronics Conference Chicago, IL October 29, 1979
	Jon Martens	Metrics for Evaluating Quantitative Software Modelling Databases	Workshop on Quantitative Software Models for Reliability, Complexity, and Cost Kiamesha Lake, NY October 9-11, 1979
	Lorraine Duvall	A Review of the Evaluative Literature on Software Reliability Modelling	
	Lorraine Duvall, Shirley Gloss-Soler	A Summary of the Second Minnowbrook Workshop on Software Performance Evaluation	
August, 1979	Shirley Gloss-Soler	Data & Analysis Center for Software	Sixteenth Annual Computer Personnel Research Conference Princeton, NJ August 16-17, 1979
July 1979	Lorraine Duvall, Shirley Gloss-Soler	Status of Data & Analysis Center for Software	Second Minnowbrook Workshop on Software Performance Evaluation, Syracuse University's Minnowbrook Conference Center Blue Moutain Lake, NY July 31 - August 3, 1979
June, 1979	Shirley Gloss-Soler, Lorraine Duvall	An Introduction to the DACS Software Engineering Bibliographic Database	Eighteenth Annual Technical Symposium Information Systems---Effectiveness for the User Gaithersburg, MD June 21, 1979
March, 1979	Lorraine Duvall	A Software Reliability Review	Proceedings Second National Reliability Conference Birmingham, England March 29-30, 1979
January, 1979	Lorraine Duvall	Data & Analysis Center for Software	Proceedings of 1979 Annual USAF Academy Computer Related Information Systems Symposium [illegible] Colorado Springs, CO January 24-26, 1979
December, 1978	Lorraine Duvall	Introduction to the Data & Analysis Center for Software, (DACS)	ACM 1978, Washington, DC December 5, 1978
September, 1978	Lorraine Duvall	An Introduction to the Data & Analysis Center for Software	First Minnowbrook Workshop on Software Performance Evaluation, Syracuse University's Minnowbrook Conference Center Blue Moutain Lake, NY September 20, 1978
	Lorraine Duvall	An Introduction to the Data & Analysis Center for Software	Third Software Engineering Workshop, NASA Goddard Space Flight Center Greenbelt, MD September 13, 1978

SECTION V

TASK 7 TECHNOLOGY INFORMATION BASE

5.1 Introduction

One of the functions of the DACS is the establishment of a software technology scientific and technical information base. During this reporting period, we have produced a software engineering thesaurus, a glossary of software engineering terms compiled from the literature, and a software engineering document library including a computer database containing files of document descriptions and bibliographic citations.

The status of this information base is discussed in this section of the report.

5.2 The Software Engineering Document Library

The DACS Library has been established to provide a readily accessible source of comprehensive information on the state-of-the-art in software engineering as well as a means of channeling that information to those people in the software engineering community who can make use of it in their day-to-day activities of developing, maintaining, and managing software. The bibliographic collection is composed of texts, technical reports, theses, journal articles, proceedings and other documents relating to software engineering, reliability, cost and quality factors, maintainability, and other topics deemed appropriate.

As of 1 February 1980, the DACS has approximately 2200 documents relating to software engineering; 1346 of these document have been indexed, their bibliographic citations coded, keypunched and placed online. Table 5.2-I lists the types of documents in the online portion of the collection, the number of documents of each type and the percent for each type. Note that 85% of the bibliographic collection is made up of journal articles and papers. Most of the papers have been presented at various conferences and symposia and have been published in the proceedings of those conferences and symposia.

A full listing of the various journals and proceedings from which the DACS has obtained articles and papers is contained in table 5.2-II.

The DACS Software Engineering Bibliographic collection is made available to members of the software engineering community chiefly in the form of bibliographies. The DACS functions as a reference library in that custom searches are made upon receipt of a request from a user. The request may be made during a visit to the center, by a telephone call, or by a written request. When the search has been completed, the resulting bibliography is given or sent to the user who made the request.

TABLE 5.2-I DOCUMENT TYPE

Type of Document	Number of Documents	Percent of Collection
Bibliography	0006	.4
Journal Article	0253	18.8
Technical Report	0156	11.6
Text	0008	.6
Paper	0898	66.7
Standard	0008	.6
Regulation	0003	.2
Specification	0003	.2
Monograph	0011	.8

TABLE 5.2-II

BREAKDOWN STATISTICS FOR JOURNAL ARTICLES AND PAPERS
AS OF 02/07/80

NAME OF JOURNAL OR CONFERENCE	NO. PAPERS OR ARTICLES
ACM 78,PROCEEDINGS OF THE 1978 ANNUAL CONFERENCE	0036
ACM COMPUTING SURVEYS	0005
AFIPS NAT'L COMPUTER CONFERENCE	0018
PROCEEDINGS OF THE SECOND ARMY SOFTWARE SYMPOSIUM	0028
BELL SYSTEM TECHNICAL JOURNAL	0001
AIAA/NASA/IEEE/ACM COMPUTERS IN AEROSPACE CONFERENCE	0058
COMPUTERS AND PEOPLE	0002
COMPCON '78	0001
1976 CONFERENCE ON INFORMATION SCIENCES & SYSTEMS	0001
PROCEEDINGS,COMPSAC 77	0050
PROCEEDINGS ,COMPSAC 78	0100
COMPUTER	0049
COMPUTERWORLD	0018
COMPUTER BUSINESS NEWS	0003
FIRST CANADIAN SRE RELIABILITY SYMPOSIUM, 1974	0001
1975 CANADIAN SRE RELIABILITY SYMPOSIUM	0001
ANN. COMPUTER RELATED INFORMATION SYSTEMS SYMP.,1977	0006
ANN. COMPUTER RELATED INFORMATION SYSTEMS SYMP.,1978	0005
ANN. COMPUTER RELATED INFORMATION SYSTEMS SYMP.,1979	0008
COMPUTER SYSTEMS NEWS	0001
1973 IEEE SYMPOSIUM ON COMPUTER SOFTWARE RELIABILITY	0025
3RD DIGITAL AVIONICS CONFERENCE	0001
DATABASE	0002
DATAMATION	0012
EASCON '74	0002
ELECTRONICS	0006
ELECTRONIC DESIGN	0009
1972 FALL JOINT COMPUTER CONFERENCE	0001
HARVARD BUSINESS REVIEW	0001
INFOR (CANAD.JRNL OF OPERATIONAL RES.& INFOR.PROC.)	0001
INFORMATION PROCESSING 74	0003
4TH INT'L CONFERENCE ON SOFTWARE ENGINEERING	0016
2ND SOFTWARE LIFE CYCLE MANAGEMENT CONFERENCE	0001
MICRO & MINI SYSTEMS, SYMP. ON TRENDS & APPL.,1976	0012
MILITARY ELECTRONICS/COUNTERMEASURES	0002
NAECON '75 RECORD (NATL AVIONICS & ELECTRICAL CONF)	0005
NATIONAL COMPUTER CONFERENCE, 1974	0002
NATIONAL COMPUTER CONFERENCE, 1975	0001

TABLE 5.2-II con't

BREAKDOWN STATISTICS FOR JOURNAL ARTICLES AND PAPERS
AS OF 02/07/80

NAME OF JOURNAL OR CONFERENCE	NO. PAPERS OR ARTICLES
2ND NATL RELIABILITY CONF.,1979 (BIRMINGHAM, ENGLAND)	0003
ACM/NBS 18TH ANN. TECHNICAL SYMP.,INFORMATION SYSTEMS	0013
PROCEEDINGS OF THE IEEE, 1975	0001
PRACTICAL STRATEGIES FOR DEVELOP'G LG SOFTWARE SYSTEMS	0008
PROCEEDINGS, 1976 RAC RELIABILITY WORKSHOP	0001
RESEARCH DIRECTIONS IN SOFTWARE TECHNOLOGY	0055
1975 ANNUAL RELIABILITY & MAINTAINABILITY SYMPOSIUM	0005
1979 ANNUAL RELIABILITY AND MAINTAINABILITY SYMPOSIUM	0004
PROCEEDINGS,1975 INT'L CONFERENCE ON RELIABLE SOFTWARE	0059
MRI SYMPOSIUM ON COMPUTER SOFTWARE ENGINEERING 1976	0038
2ND INT'L CONFERENCE ON SOFTWARE ENGINEERING	0084
3RD INT'L CONFERENCE ON SOFTWARE ENGINEERING	0047
2ND INT'L CONF. ON SOFTWARE ENGINEERING, ADDENDUM	0018
1ST NATL. CONFERENCE ON SOFTWARE ENGINEERING (1975)	0014
SIGNAL - JOURNAL OF AFCEA	0001
1977 ANN. SOFTWARE LIFE CYCLE MANAGEMENT WKSHOP	0033
SOFTWARE ENGINEERING NOTES (ACM SIGSOFT)	0032
IEEE SPECTRUM	0001
PROCEEDINGS, 1969 ANN. SYMPOSIUM ON RELIABILITY	0001
PROCEEDINGS,1970 ANN.SYMPOSIUM ON RELIABILITY	0001
CONF. ON SPECIFICATIONS OF RELIABLE SOFTWARE,1979	0019
2ND SUMMER SOFTWARE ENGINEERING WORKSHOP	0011
3RD SUMMER SOFTWARE ENGINEERING WORKSHOP	0012
WKSHOP ON SOFTWARE TESTING & TEST DOCUMENTATION DEC'78	0022
SOFTWARE - PRACTICE AND EXPERIENCE	0016
TOOLS FOR EMBEDDED COMPUTING SYSTEMS SOFTWARE	0030
IEEE TRANSACTIONS ON COMPUTERS	0001
IEEE TRANSACTIONS ON RELIABILITY	0002
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING	0178
TUTORIAL: SOFTWARE MANAGEMENT	0001

NOTE: THE TOTAL DOCUMENTS IN THIS TABLE WILL BE GREATER THAN THE SUM OF THE PAPERS AND JOURNAL ARTICLES IN THE PRECEEDING TABLE. THE CAUSE IS THAT SEVERAL PAPERS HAVE APPEARED IN MORE THAN ONE PUBLICATION; TO PROVIDE ALTERNATE SOURCES FOR OUR USERS, WE HAVE INCLUDED REFERENCES TO ALL SOURCES OF WHICH WE ARE AWARE.

5.2.1 The DACS Thesaurus

In order to index the collection, it was necessary to, first, construct a thesaurus of terms which could be used for both indexing and retrieval purposes. This thesaurus has been named the DACS Software Engineering Thesaurus and is composed of 475 keywords or terms arranged into 50 groups of closely related keywords headed by a cluster term. A listing of the 50 cluster terms is contained in table 5.2-III.

Appendix B contains a listing of all 475 keywords in the DACS Thesaurus together with the number of documents to which each keyword has been assigned.

There were 8164 keyword assignments made to 1346 online document descriptions, yielding an average of six keywords per document.

5.2.1.1 Description of Keyword Clusters

Table 5.2-IV, contained on the following pages, gives for each cluster term, the number of terms in that cluster, a description of terms in that cluster, a description of how the terms are used in indexing and/or a description of the type of information which a user could expect to retrieve when searching the bibliographic datafiles on keywords in that cluster.

5.2.2 Database Implementation and Retrieval

The DACS software engineering bibliographic datafiles are implemented to provide rapid retrieval of individual document citations. The datafiles are accessed through the Management Data Query System (MDQS) implemented on the Honeywell 6180 computer. This database management system allows searching on any of the items in a bibliographic citation and/or assigned keywords and outputs bibliographies of documents retrieved during the searching process. Searching may be done on line or in batch mode.

The retrieval language for MDQS is procedurally oriented and syntactically demanding for all but the simplest retrieves or where any degree of formatting of output is needed. To alleviate the problems inherent in such a system, a series of parameterized queries have been written. These queries are maintained online and can be utilized by non-programming personnel. The person running a search need only log on, type the letters MDQ and the name of the file containing the query, and then type "RUN." The MDQS system will request three parameters, (the topic of the search, the search strategy, and the retrieval qualifiers), actuate the retrieval and direct the resulting bibliography to an offline printer. At the present time, a DACS information specialist or the user supplies the parameters, and the search is actually run by clerical personnel.

5.2.3 Document Processing

Input processing procedures for the software engineering documents have been established to assure that the documents are relevant to the Center and that after inclusion into the bibliographic collection they are easily retrievable. The major activities are log-in, code, index, verify, keypunch, and load.

TABLE 5.2-III CLUSTER TERMS

APPLICATIONS
ARCHITECTURE
ARTIFICIAL INTELLIGENCE
CERTIFICATION
COMPLEXITY
CONVERSIONS
COST
CORRECTNESS PROOFS
DATA
DEBUGGING
DOCUMENTATION
DISTRIBUTED PROCESSING
DESIGN
DEVELOPMENT
EDUCATION
ERRORS
FAILURES
FAULT
FUNCTIONS
HIERARCHY
IMPLEMENTATION
LANGUAGES
MICRO COMPUTERS
MODELS
MEASUREMENTS
MANAGEMENT
MINICOMPUTERS
MODULES
MODELLING AND SIMULATION TOOLS
MAINTENANCE
OPERATING SYSTEMS
PERFORMANCE
PROGRAMS
PROGRAMMING
PROCEDURES
QUALITY
RELIABILITY
REQUIREMENTS
SECURITY
SOFTWARE ENGINEERING
SPECIFICATIONS
SUPPORT SOFTWARE
STANDARDS
SOFTWARE
STANDARDIZATION
TRI-SERVICE
TESTING
VERIFICATION
VALIDATION
VIRTUAL MACHINES

TABLE 5.2-IV DESCRIPTION OF KEYWORD CLUSTERS

CLUSTER TERM	DESCRIPTION
APPLICATIONS (35 keywords)	Most of these terms are not specific to computer software or hardware, but indicate that a document indexed using one of these terms describes the development or use of software for a specific application.
ARCHITECTURE (4 keywords)	These terms reference documents related to either software architecture or to system architecture which includes both hardware and software architecture.
ARTIFICIAL INTELLIGENCE (1 keyword)	This term is used to index documents relating to the use of software or the development of software which appears to reason, learn, and improve itself.
CERTIFICATION (1 keyword)	This term is used to index documents relating to any parts of the certification process for software. This term may also be used as a part of a boolean retrieve with testing, support software, etc., to retrieve on a narrower basis.
COMPLEXITY (7 keywords)	These terms are used to index documents relating to program complexity, programming language complexity and measurement of complexity.
CONVERSIONS (4 keywords)	These terms are used to index documents referring to costs, method, or aids for conversions of existing software from one language to another language or from one hardware/software configuration to another.
COST (17 keywords)	These terms are used to index and retrieve documents relating to the cost associated with any activity during any phase of the software life cycle. This cluster also contains terms relating to productivity and to management cost data.
CORRECTNESS PROOFS (14 keywords)	This group of terms is used to index and retrieve documents relating to the development or use of correctness proofs, by either automated or manual means. The terms termination proof and symbolic execution were also assigned to this cluster.
DATA (10 keywords)	These terms reference sources of data, means of collecting or acquiring data, and validation of data. The data referenced relates to software development, costs, maintenance or reliability.
DEBUGGING (2 keywords)	These terms reference documents which refer to methods for or experiences in locating and correcting faults whose existence is known. Documents indexed under these terms will specifically refer to the process as "debugging." Other material relating to the same process are indexed using various terms from the Error and Fault clusters.
DOCUMENTATION (2 keywords)	These terms reference documents dealing with documentation by either manual or automated means. Documents indexed may refer to the process of producing documentation or to the resulting product documentation. To retrieve documents relating to documentation produced during a particular phase of the software life cycle or for a particular purpose, use a term from the documentation cluster as one term in a boolean search. e.g., Documentation and Design.
DISTRIBUTED PROCESSING (1 keyword)	This term references primarily documents relating to the development of distributed or parallel processing systems. The collection has very little in this area.

TABLE 5.2-IV con't

DESIGN (28 keywords)	These terms reference documents which relate experiences using various design tools or methodologies, which describe a particular design tool or methodology, which describe particular activities occurring during the design phase of the software life cycle, or which describe a completed design for a software product or component.
DEVELOPMENT (11 keywords)	These terms reference documents relating to the developmental process, developmental methodologies and developmental tools. Documents relating to all of the phases in the software development cycle will be indexed using a term from this cluster; if the treatment is extensive for each phase, it will also be indexed using keywords which apply to that phase.
EDUCATION (2 keywords)	These terms refer to formal and informal software engineering educational processes and curricula including technology transfer.
ERRORS (21 keywords)	These terms are used to index and retrieve documents relating to error analysis, detection, correction, and prevention. Documents relating to error data, and error categories are also indexed using terms from this group.
FAILURES (6 keywords)	These terms reference documents relating to failure rates, failure ratios, failure categorization and failure data.
FAULTS (11 keywords)	These terms reference documents relating to both faults and fault-tolerance.
FUNCTIONS (1 keyword)	This term is used to index or retrieve documents which refer to development, design, or programming approaches which view a software system component, or module in terms of its required or intended function.
HIERARCHY (3 keywords)	These terms are used primarily as a cross reference for boolean searches. e.g. Hierarchy and Design
IMPLEMENTATION (2 keywords)	These terms reference documents which are concerned with the methods and processes by which a design is transformed into a software product. Documents concerned with the implementation phase of the software life cycle are indexed using these terms.
LANGUAGES (45 keywords)	These terms are used to index documents evaluating or comparing two or more languages, or evaluating or describing particular features or applications of a given language. The terms reference various programming languages, design languages, requirements languages and specification languages.
MICROCOMPUTERS (4 keywords)	These terms are used to refer to the design or development of software for microcomputers or to the use of microcomputers in a particular application. These terms may be used as one term in a boolean search to retrieve information on a particular application of microcomputers.
MODELS (27 keywords)	These terms reference documents describing the use, validity or development of models. Models include reliability, availability, and error models. Documents are also indexed on the mathematical methodology which is used to derive or which is the form assumed by a particular model.
MEASUREMENTS (1 keyword)	Used primarily as a cross reference keyword to allow retrieval of information about measures or measurement techniques developed for a particular application. i.e., Maintainability and Measurements.

TABLE 5.2-IV con't

MANAGEMENT (11 keywords)	Terms used to reference documents referring to management aspects of software engineering.
MINICOMPUTERS (1 keyword)	May refer to the design or development of software for a minicomputer system, or to the use of minicomputers in a software development project (e.g., as an emulator or simulator), or to a analysis of minicomputers vs mainframe computers as components of a hardware/software system. Best use is as one term in a boolean search to retrieve information on particular uses of minicomputers.
MODULES (7 keywords)	This cluster contains a more loosely related group of adjectival and nounal keywords used to reference documents relating to software designed and implemented in a modular fashion or to the modules in software product.
MODELLING AND SIMULATION TOOLS (8 keywords)	These keywords reference documents concerned with software tools used for trade-off studies and to investigate particular abstractions and approaches for system design.
MAINTENANCE (11 Keywords)	These terms reference documents concerned with the actities occurring during the operations and maintenance phase of the software life cycle.
OPERATING SYSTEMS (17 keywords)	Terms in this cluster reference documents concerned with particular aspects of operating systems, with design and development of entire operating systems, or with evaluation of a particular operating system.
PERFORMANCE (3 keywords)	These terms reference documents concerned with system performance, either as a whole, or with particular performance factors.
PROGRAMS (11 keywords)	This cluster contains a mixed bag of keywords referencing documents concerned with attributes of programs. Four of the keywords in this cluster reference program library systems.
PROGRAMMING (25 keywords)	Terms in this cluster reference documents concerned with programming methods and techniques and programming for particular applications. The terms usually described as "modern programming practices" are also included in this group.
PROCEDURES (1 keyword)	Procedures is a term used ambiguously in the literature to refer to functions performed by an operating system, as a quasi-synonym for a module or its output or as a control structure. Best used in a boolean search combined with another term so as to narrow the retrivals to reflect the user's information need. i.e., Operating Systems and Procedures.
QUALITY (29 keywords)	This cluster contains the terms which reference documents concerned with the quality attributes or "ilities." It also contains the terms related to software quality assurance.
RELIABILITY (14 keywords)	Conceptually, reliability belongs in the Quality cluster; however, because there is so much literature on the various aspects of, methods for achieving, controversy concerning, and means for measuring software reliability; reliability and its related keywords have been grouped into a second cluster.
REQUIREMENTS (6 keywords)	Terms which reference documents relating to activities performed or processes occurring during the requirements phase of the software life cycle.

TABLE 5.2-IV con't

SECURITY (11 keywords)	The keywords in this cluster reference documents concerned with both system access security, the security of data resident in the system, and the privacy concerns of entities about whom (which) data is contained in the system.
SOFTWARE ENGINEERING (16 keywords)	The keywords in this cluster reference documents concerned with software engineering tools, techniques, and aids; economic and managerial aspects of software engineering; and general background information on software engineering as a discipline.
SPECIFICATIONS (9 keywords)	Terms which reference documents relating to activities performed or processes occurring during the specifications phase of the software life cycle.
SUPPORT SOFTWARE (10 keywords)	This cluster contains the cluster term, support software, and nine subterms. The subterms refer to types or classes of support software e.g., compilers, editors, interpreters. The cluster term is used most often to index documents related to software tools and used in a boolean search to retrieve documents relating to a particular tool type. e.g., Support Software and Testing would retrieve those documents containing information about software test tools.
STANDARDS (1 keyword)	Used in indexing primarily as a cross reference term to provide for boolean searches to retrieve information on the different types of standards.
SOFTWARE (3 keywords)	The cluster term alone is never used in indexing. It provides a home for two other terms. One, software life cycle is used to retrieve documents which give overviews or general treatments of the software life cycle. The second term, software physics, is a controversial term, with different meanings in different camps and serious question about whether either meaning is valid for software engineering. The term is included in the DACS Thesaurus because it is in use and because the concepts referenced by it are of concern to the software engineering community.
STANDARDIZATION (1 keyword)	Used to refer to efforts or proposals for standards development. Best used in a boolean search for retrievals.
TRI-SERVICE (2 keywords)	Used to reference documents relating to joint U.S. Army, U.S. Navy, U.S. Air Force efforts or projects of interest to the software engineering community.
TESTING (28 keywords)	Used to refer to various levels of testing, i.e., module, program, system testing; to various testing techniques; and to test tools of various types.
VERIFICATION (8 keywords)	Used to reference documents containing information on verification activities performed throughout or during any phase of the software development cycle.
VALIDATION (3 keywords)	Used to reference documents containing information on the validation process.
VIRTUAL MACHINES (7 keywords)	The terms in this cluster reference documents containing information relating to abstract machines, user-interactive systems and multiprogramming for such systems.

Log-In

This activity begins the document-input process. The document is tested for uniqueness of author and title against the current on-line author catalog and the current in-process author catalog. If the document is a possible duplicate; it is compared to the document already in the collection; if an exact duplicate, it is marked as copy two, three, or four etc. and filed, if a duplicate from a different source, the new document is marked as copy two, three, or four, etc. and an entry is made on an addenda sheet for keypunching to reflect the additional source.

If the document is unique, a document accession number is assigned and the accession number, author and title entered on the DACSLOG data sheet for entry at the end of the day in the in-process author catalog. After entry on the DACSLOG data sheet, a document description form is prepared which contains all bibliographic information related to the document. At this time, all coded data (journal codes, availability codes, document-type codes, author-organization codes and contract-sponsor codes) are looked up in the tables (or created and added to the tables) and entered on the document description form. After LOG IN, either coding or indexing may be the next step.

Code

The information contained on the document description form is transcribed onto the coding sheets for subsequent keypunching.

Index

Descriptors (keywords) are selected from the DACS Software Engineering Thesaurus. The keyword codes selected are entered on the Document Description Form and also on the coding sheets if coding of bibliographic information has already been done. If coding has not yet been completed, the keyword codes are written on the document description form to be coded when the rest of the bibliographic information is entered.

Verify

Completed coding sheets are reviewed for legibility of all characters, conformance to coding conventions, and accuracy of transcription from the document description forms.

Keypunch

The verified coding sheets are sent to a keypunching service in batches of 50 to 100 sets of 3 or 4 sheets. Each set represents one document. The information on the coding sheets is transferred to punch cards and verified.

Load

The punched cards are read into a BCD file, sorted, converted to ASCII format, and the card images listed in hard copy. The listing is skimmed for obvious inconsistencies, EBCDIC-ASCII incompatabilities are edited, errors are edited, the file is reconverted to BCD and written to the bibliographic datafiles by invoking the Indexed Sequential Processor. At this step, the six lookup tables for coded data are updated as necessary. When the new bibliographic data has been loaded, a set of procedures which locates unmatched coded data are run as a final check. Any incorrect codes are noted in the source file for the next file maintenance session.

5.3 The DACS Glossary

There were two objectives in compiling the DACS Glossary. The first objective was to record the terminology currently in use. The second was to provide users of the DACS software engineering bibliographic services, a means of ensuring that they are using terms from the DACS Thesaurus for their retrievals in the same way as the terms were used for indexing. A closely related objective is to promote consistency in indexing new documents for the collection.

The first version of the DACS Glossary was published by the DACS in October 1979. Most of the terms and definitions were compiled from the software engineering literature. This DACS Glossary contains 1123 terms, 1100 of which have one or more definitions and 23 of which are cross-references to other terms.

The glossary is now in a maintenance and refinement phase. A few new definitions have been added, and the definitions presently in the glossary are being edited for consistent format, removal of inaccuracies and removal of different terms which name the same concept. The work with the IEEE Terminology Task Group is providing a major input to this necessary refinement, as discussed in the next section.

5.3.1 Terminology Task Group Coordination

We have taken an active role in the IEEE Terminology Task Group.* Shirley Gloss-Soler, of IITRI, is chairperson of this task group. In addition, a draft version of the DACS Glossary was provided to this group and was used as a base from which candidate terms were selected for a working draft of the second version of their Software Engineering Terminology (SET).

The task group presently has 112 members from government, industry, and the universities. Approximately 70 are active participants. To facilitate coordination, a steering committee of 11 people and 10 subgroups was formed. Table 5.3-I provides information on the coordination meetings held by this steering committee.

Their draft terminology, as of 1 February 1980, contains 858 terms. The terminology document is scheduled to be completed April 1981.

---

*Terminology Task Group, Subcommittee on Software Engineering Standards, Technical Committee on Software Engineering, IEEE Computer Society.

TABLE 5.3-I

MEETINGS OF STEERING COMMITTEE OF TERMINOLOGY TASK GROUP (TTG)

Date	Place	Tasks Accomplished at Meeting
August 6, 7, 1979	Griffiss AFB Rome, NY	Selected 792 terms to be contained in first working draft of SET. Partitioned 792 terms among 10 coordinators who are responsible for coordinating input from members of TTG on those terms. Established tentative timetable for accomplishment of task.
November 2, 1979	National Bureau of Standards Gaithersburg, MD	Decided upon scope of SET format and scope of definitions, arrangement of terms within SET, and approval levels and mechanisms for definitions.
January 16-18, 1980	NCR Corporation San Diego, CA	Reviewed, discussed and either deleted, approved, or sent back for rework 233 terms; 23 new terms added.

SECTION VI

TASK 8 PRODUCTS AND SERVICES PREPARATION AND DISTRIBUTION

6.1 Introduction

This section summarizes the results of the tasks to produce and distribute the DACS products and services. A characterization of these products and services is discussed, along with summarized quantitative information on requests processed. More detailed data on the number and type of requests processed is included in Section 7.2, Improving Center Effectiveness.

6.2 Data Subsets

The purpose of this task is to distribute subsets of the DACS database (as discussed in Section 3.2 of the report) to aid in research efforts that require productivity, cost, complexity, error, and change data. These datasets are being used to validate and refine software reliability, maintainability, and software productivity models and methods. The availability of the RADC Productivity Database was announced in the June 1979 DACS Newsletter. As of 1 February, 32 copies of this dataset had been distributed in hard copy report format or on magnetic tape.

To facilitate distribution of this dataset, descriptive literature on ordering this dataset is provided to the user and contains a check-off list for specifying the sorted order of a computer listing and magnetic tape requirements. This data subset is distributed along with a data dictionary describing the data elements and a copy of the draft report entitled "Software Data Collection and Analysis" which presents a preliminary analysis of the dataset.

The availability of the Software Reliability Database was announced in the December 1979 DACS Newsletter. The report entitled "Software Reliability Data" has been compiled and contains descriptions of the data and the characteristics of the software systems, listings of the data, and ordering information for obtaining a copy of the data on magnetic tape. As of 1 February 1980, requests for approximately 40 copies of this database had been received.

The NASA SEL Database is presently being studied to determine the options for subsetting and disseminating this information in a usable form.

6.3 Data Compendiums

The production of a software engineering data compendium was initiated during this reporting period. The purpose of this compedium is to profile the software engineering data that has been collected by the NASA Goddard Software Engineering Laboratory during software development projects at that facility.

Each software development project will be described in terms of its purpose, size, life cycle dates and development methodologies, tools and techniques. Data from computer run analysis forms are used to profile the purposes and results of the projects' computer runs. From change report forms, the data elements such as types of errors and effort for error changes are profiled. Several resource expenditure forms are used to produce project profiles regarding number of computer runs, computer time and manpower hours by task.

Several other features are included in the compendium to enable potential users of the data to determine the relevance of the data to their applications. Included among the features are a data dictionary and an evaluation of the completeness of the data by project and data type.

6.4 Technical Reports

Two state-of-the-art reports were produced during this reporting period. The first report, entitled "Quantitative Software Models," was compiled from the Data Parameters Report and is a compendium of attributes of software cost/productivity, reliability/error analysis, and complexity models. The availability of this software engineering research review was announced in the March 1979 DACS Newsletter. As of 1 February 1980, 452 copies of this report had been distributed to the software technology community.

The second report is entitled "Software Maintenance Technology" and provides a synthesis of the information on software maintenance topics that exists in current articles, papers, reports, and books. This maintenance technology report will be distributed as a RADC Technical Report. The availability will be announced in the March 1980 DACS Newsletter.

This state-of-the-art report presents a practical guide on software maintenance tools in terms of categorization, potential use and benefit, and experience in using the specific tool. This report defines a set of maintenance technology functions including configuration control; operations monitoring/evaluation; redesign; code production and analysis; verification and validation; testing and integration; and documentation. Forty maintenance tools and techniques are defined for each technology function and an indication of how they relate to O & M activities is provided. One of the major findings of this comprehensive report is that there has been very little determination of the effectiveness of these tools and techniques on the O & M phase, (reference 14).

6.5 Technical Inquiries

Technical inquiries to the DACS are received and processed on a daily basis. These inquiries are received in many forms: by letter, telephone call, visit, or by use of the bibliographic request form contained in BIB-1. The information requests have ranged from the very specific to general questions on software engineering methodologies. Table 6.5-I contains the description of 39 sample inquiries we have received. The total number of technical inquiries received during the ten-month period from 1 April 1979 through 1 February 1980 was 281. This number does not reflect the direct requests for DACS documents or to be placed on the Newsletter mailing list.

A technical inquiry may be answered in the following one or more ways:

- A custom bibliographic search on the subject area of interest is performed.
- A preliminary analysis of the subject literature is made and summary information prepared.
- A subset of the DACS database is produced.
- Relevent DACS literature is distributed.
- Referrals to other sources are provided.

The DACS has provided 320 custom bibliographies to users. The information needs which prompted these searches have ranged from broad information requests, whose result could be described as file dumps, to very narrow requests such as "decision table programming in FORTRAN."

The search profile below indicates the 13 most popular requests. The remaining 46 search topics were requested 8 or less times and included such topics as design, data collection, modern programming practices, productivity, embedded computers, and software tools.

Search Topic	Number of Requests
Costs	28
Quality Assurance	20
Reliability	17
Testing	16
Quality Factors	15
Verification/Validation	14
Development	12
Conversion	11
Documentation	11
Errors	10
Maintenance	10
Management	10
Configuration Management	9

6.6 Miscellaneous Services

In addition to the previously mentioned DACS products and services, we have produced and distributed a summary of the "Second Minnowbrook Workshop on Software Performance Modeling," edited and distributed copies of a paper by Tom Gilb entitled "Controlling Maintainability - A Quantitative Approach for Software," distributed bibliographies of RADC/IS Software Technology reports, and took an active role in organizing sessions and workshops related to software technology.

TABLE 6.5-I SAMPLE DACS TECHNICAL INQUIRIES

Researching certification of computer programs, equipment, and personnel
User needs specific information on the processing and validation of meterorological (weather) data
We are establishing an internal verification and validation group and we need data on software errors, specifically software errors by instruction type
Require information on code verification, i.e. tools that can be used to certify that minimum testing standards have been met
Need information on manpower resource estimation
Would like to know about any techniques to evaluate technical proposals concerning software development, both on a "stand-alone" basis and a hardware basis (i.e., complete system)
This office would appreciate being placed on your distribution list and would like further information regarding the Data Analysis Center for Software
Please send information describing the DACS charter and accomplishments
User needs information on life cycle costing figures or projections and maintenance costing data on the benefits of distributed processing systems
Need information on assembly language vs HOL programmer productivity
Custom bibliography on the verification and validation of computer software
Information needed to prepare a user's acceptance criteria list for a system under development and acceptance testing
Interested in software life cycle cost models
Documentation to accompany testing - formats of test plans/specifications at various stages
We would appreciate your sending copies of the various forms used for collection of software data for analysis
Am looking for any information on product safety - the impact on safety caused by software problems

TABLE 6.5-I con't

Would like data on frequency of errors for real-time systems
Would like list of references on software quality, reliability, and testing
Need to develop standards and procedures for testing (unit, acceptance, and system)
Need information on estimating manpower resources required for software projects
Reviews weapon systems status prior to full scale production - needs background information on error management, reliability, and quality
Would like information on estimating the size of C^3I systems, functional groups, and modules
Doing a study on life cycle cost models for an Air Force human resources lab; would like information on life-cycle cost models
Would like list of references to be used in formulating and implementing software reliability and quality assurance programs
Am a government contractor - doing a large programming job. Need to set up internal standards for design, coding, programming and also evaluation of completed programs
Would like information on the use and benefits of specific programming tools
Need information on software fault detection as well as fault tolerant operations and recovery; I am specifically interested in techniques used to structure software in order to accomplish the above
Need to know the cost impact of implementing a MIL-S-52779 QA plan, especially interested in cost impact during the development phase of the life cycle
Would like a list of references on design methodologies
We are specifically interested in samples of standards and/or procedures for collecting error (failure) information to be used in procuring digital flight control software; we do not want it for reliability estimation but to determine what kinds of tools and techniques need to be developed to prevent/detect these errors

TABLE 6.5-I con't

Am looking for survey articles on software acquisition and cost estimation
We have an urgent need for definitions of software engineering terms
We are presently performing a tool survey and would appreciate your assistance with regard to a bibliographic search of your database in conjunction with our efforts
My department has, as its principal responsibility, the quality assurance control and monitoring function on a large SWS Navigation Subsystem; the inclusion of computer software to this quality discipline on this subsystem has been a recent addition, directed by the Navy; any suggestions that would enhance the effectiveness of our quality assurance efforts would be appreciated
Need practical information on sizing and timing estimation methods - current information requested
One of UNIDO's (United Nations Industrial Development Organization) problems is funding the development of duplicate software packages because minimum consideration is given to portability, maintainability and reliability; please send me information on these aspects of software
Involved in a large USAF language conversion effort - Can't find anything in the literature
The determination of factors which contribute to software quality and the application of techniques to measure these factors
My present research is concerned with development of tools for predicting trouble spots in software development projects; what data do you have available on predictive tools that operate during the specification phase and try to predict how errors will be distributed in the final delivered products

SECTION VII

CENTER EVALUATION AND EFFECTIVENESS

7.1 Task 9 Center Evaluation Mechanisms

7.1.1 Introduction

The purpose of this task is to develop a philosophy and criteria for evaluating the effectiveness of the DACS. One of the basic elements of this task is to utilize the actual conditions of the day-to-day operation of the Center. That is, no artificial tests are being implemented for collecting operational DACS data. Data is collected on a day-to-day basis during the operation of the DACS to account for personnel and service related activities. The types of data collected are discussed in this section of the report. An evaluation of the data is provided in Section 7.2, Improving Center Effectiveness.

A questionnaire was developed to provide feedback from the user. The results of the survey are summarized in this section.

7.1.2 Center Activities

An accounting mechanism has been established which provides the capability to record all resource expenditures (personnel hours and cost, travel costs, and purchased materials and services) by major DACS activity. There are 20 major activities (ventures) and they are listed in Figure 7.1-1. This data is reported on a monthly basis.

Information on the services requested is recorded on the DACS Service Request Record (SVC). This record contains the name and affilation of the user, the date requested and answered, the request type, a narrative description of the request, the recommended action, the response, and the resources expended. Each SVC is assigned a unique number. A User Profile Record (file) is maintained for each organization and summarizes the requests of all individuals within the organization. The SVC's are then filed in their respective User Profile Folder. A DACS document log is also maintained on a continuing basis which records document distribution according to type, date, and user.

A document processing log provides controls for keeping track of the processing of software engineering documents for the DACS document library. This log records the date the document was assigned a unique document accession number, the date when the document was coded and indexed, the date the coding was verified, the date the coding sheets were sent to be keypunched, and the date when the document citation was entered into the bibliographic database.

This operational data is summarized and incorporated into the monthly status report. The status report contains the number of requests processed, the number and types of products (documents) distributed, the growth of the database, the number of names on the newsletter mailing list, and the number of documents received, accessioned, and entered into the bibliographic data files.

Task 1 Establish and Operate

A 17 Establish and operate the Center - administrative

Tasks 2 and 5 Develop DB and Process Input

A 20 Database development and maintenance
A 21 Input processing - data

Tasks 3 and 4 Data Analysis and Data Parameters

A 30 Data Analysis Program (SRR-2 production)
A 31 BSDS Analysis
A 32 Data Parameters

Task 6 Current Awareness Program

A 40 Develop and expand general user awareness (newsletters and promotional material)
A 41 Preparation and participation in meetings and conferences

Task 7 STINFO

A 50 Establish and maintain library
A 51 Software Engineering Glossary
A 52 Review research projects

Task 8 Products and Services

A 60 Preparation and distribution of data subsets
A 61 Preparation and distribution of data compendiums
A 62 Final preparation and distribution of technical reports (SRR-1)
A 63 Final preparation and distribution of bibliographic products (BIB-1, Thesaurus, Glossary)
A 64 Technical and bibliographic inquiry services (SVC)

Tasks 9, 10 and 11

A 70 Evaluate and improve Center
A 71 Establish a cost recovery program

Task 12

A 80 Computer program documentation

FIGURE 7.1-1 DACS ACTIVITIES

7.1.3 User Survey

7.1.3.1 Introduction

A questionnaire was designed and distributed with the September 1980 DACS Newsletter (Figure 7.1-2). The survey contained questions concerning the type of job, ranking of the value of types of information, an indication of what DACS products and services they had used, the usefulness of these products and services, and general comment questions.

A summary of the results of the survey is contained in this section of the report. Then, in Section 7.2, these results are compared to the information collected on the operation of the DACS.

7.1.3.2 Summary of Results

The September 1979 DACS Newsletter, which included the survey questionnaire, was sent to approximately 2650 individuals and/or organizations. By the end of January, 1980, 169 responses had been received--a response rate of 6%. This is a very low response rate and sample size to provide any major conclusions. Therefore, the analysis of the survey presented in this report should be read with this point in mind.

The results summarized below are presented, first, by each individual question and the response, and subsequently, by correlating response categories of interest.

Question 1 - User Type

This first question was descriptive in nature and oriented towards the type of job that the recipient presently held. These job descriptions were categorized into three major types. The percentage of respondees for each major job type are listed below:

Manager	35%
Programmer/Analyst	36%
Researcher	29%

Question 2 - Rank of Information Type

This question was included to help determine what type of information the respondees felt was most valuable to them in performing their job. They were asked to rank seven information types.

Figure 7.1-3 illustrates the number of respondees who rated each information type as a 1, 2 or 3, where a 1 represents the most valuable. As shown in this figure, the greatest number of respondees indicated that state-of-the-art reports were considered most valuable, followed by information on new software engineering research. This figure also shows that bibliographies and raw data are not one of their top three priority items.

DACS User's Survey

To help us plan the direction of the Center and evaluate its present services, we need your feedback. By completing the following survey you can provide this feedback and make the DACS more responsive to your present and future information needs. The survey takes just a few minutes to complete and is postage-paid. Thank you for your participation.

1. Briefly describe your job, especially any software-related aspects. ______________

____

____

2. For your job, please rank the value of the following types of information. (1 = most valuable)

   ( ) Information on new s/w engineering research
   ( ) Information on previous s/w engineering research
   ( ) Surveys of current technologies, state-of-the-art reports
   ( ) Handbook type information (specs, guides, charts, tables, etc.)
   ( ) Bibliographies
   ( ) Critical reviews of s/w engineering literature
   ( ) Raw data (e.g., s/w cost, failure productivity data)

3. What DACS products or services have you used?

   ( ) Thesaurus/DACS User's Guide
   ( ) Custom Bibliographies
   ( ) Newsletter
   ( ) Technical Inquiries
   ( ) SRR-1, Quantitative S/W Models
   ( ) ______________
   ( ) Data subsets
   ( ) ______________

4. How useful were the products/services? (Please check one)

   ( ) Useless ( ) Marginally useful ( ) Fairly useful ( ) Very useful

5. What products or services should the DACS offer? ("Blue-sky" this one)

____

____

____

6. Any other comments on your information requirements or DACS services? ______________

____

____

____

9/79

FIGURE 7.1-2 USER SURVEY FORM

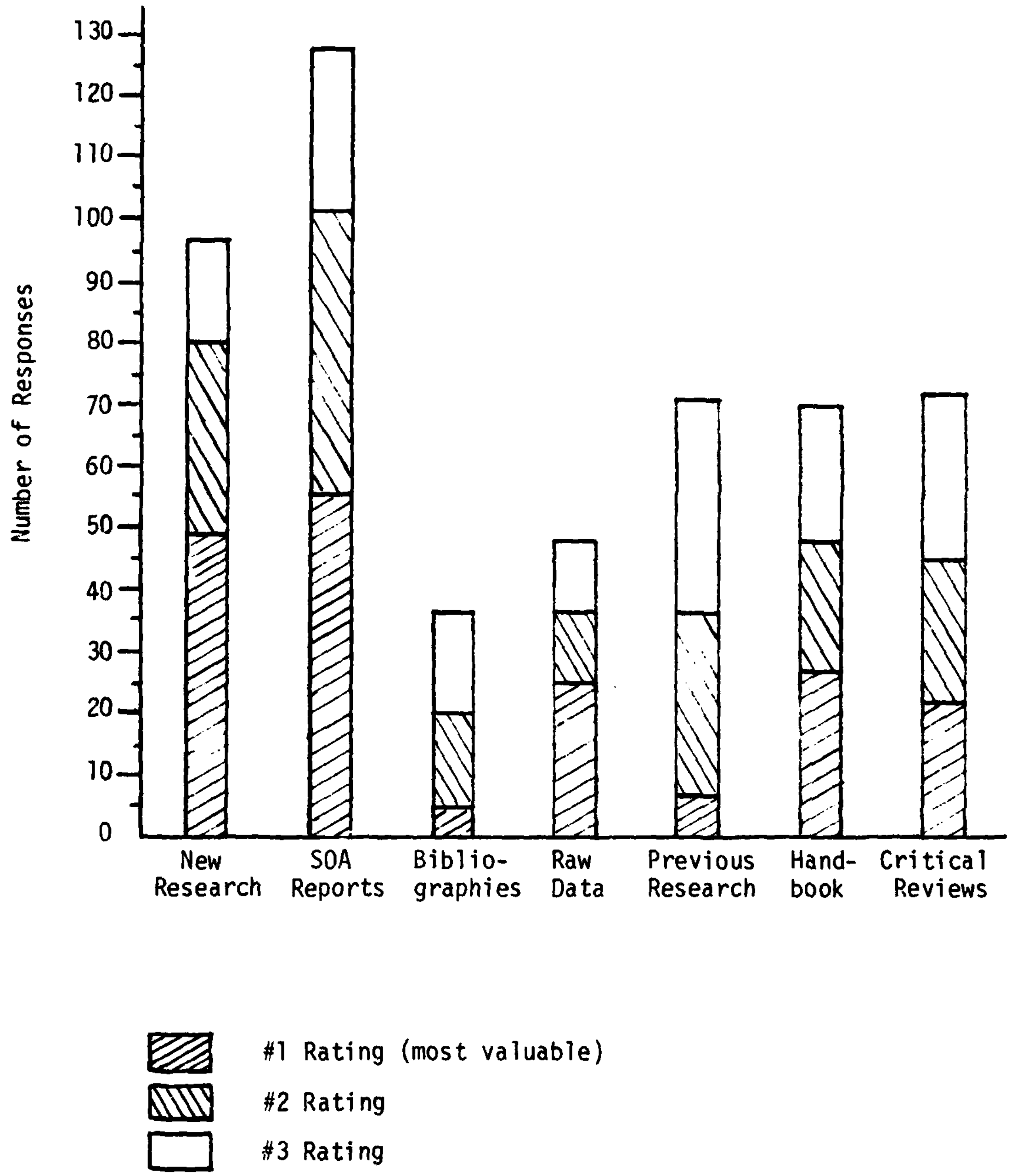


FIGURE 7.1-3 INFORMATION TYPE - MOST VALUABLE

Question 3 - Usage of DACS Products and Services

This question was included in the survey to determine what DACS products and services the respondees had used. One of the more interesting results shows that, even though the questionnaire was actually a part of the Newsletter, only 88% of the respondees indicated that they had used the Newsletter. This may be explained in part by some of the comments received stating that this September Newsletter was the first time they had heard of the Center, but they were interested in receiving more information.

Listed below is the percentage of respondees that indicated they had used the specific DACS products and services.

Thesaurus/DACS User's Guide	38%
Newsletter	88%
SRR-1, Quantitative S/W Models	16%
Data Subsets	4%
Custom Bibliographies	16%
Technical Inquiries	12%

Question 4 - Products/Services Evaluation

The respondees, in this question, were asked to indicate how useful they considered the DACS products and services. The percent of respondees indicating the four ranking levels is illustrated in Figure 7.1-4.

Although these results are skewed to the useful side, they were somewhat disappointing in light of general comments we have received about the DACS. To provide some insight to these rankings, we analyzed the comments relating to response and DACS usage. These analyses are discussed in subsequent paragraphs.

Questions 5 and 6 - Needs and Comments

A majority of the respondees (70%) provided descriptive comments on their information needs and their evaluation of the DACS products and services.

Some of the comments on information needs were a re-iteration of their perferences indicated in Question 2; others were specific technology related needs including--"information on software QA, online access to the database, a bibliography of SE tools, critical reviews of software tools, sponsor conferences or workshops, standards activity clearing house," etc.

Twenty-two percent of the respondees provided complimentary comments on the DACS including--"keep up the good work, I would like to commend your present efforts, a worthy cause - keep it up, enjoy it very much - it fills a universal need of an overview, endorse the DACS philosophy, I like the current issue," etc. As would be expected, the majority of these comments were provided by those respondees who had rated the Center as very useful or fairly useful.

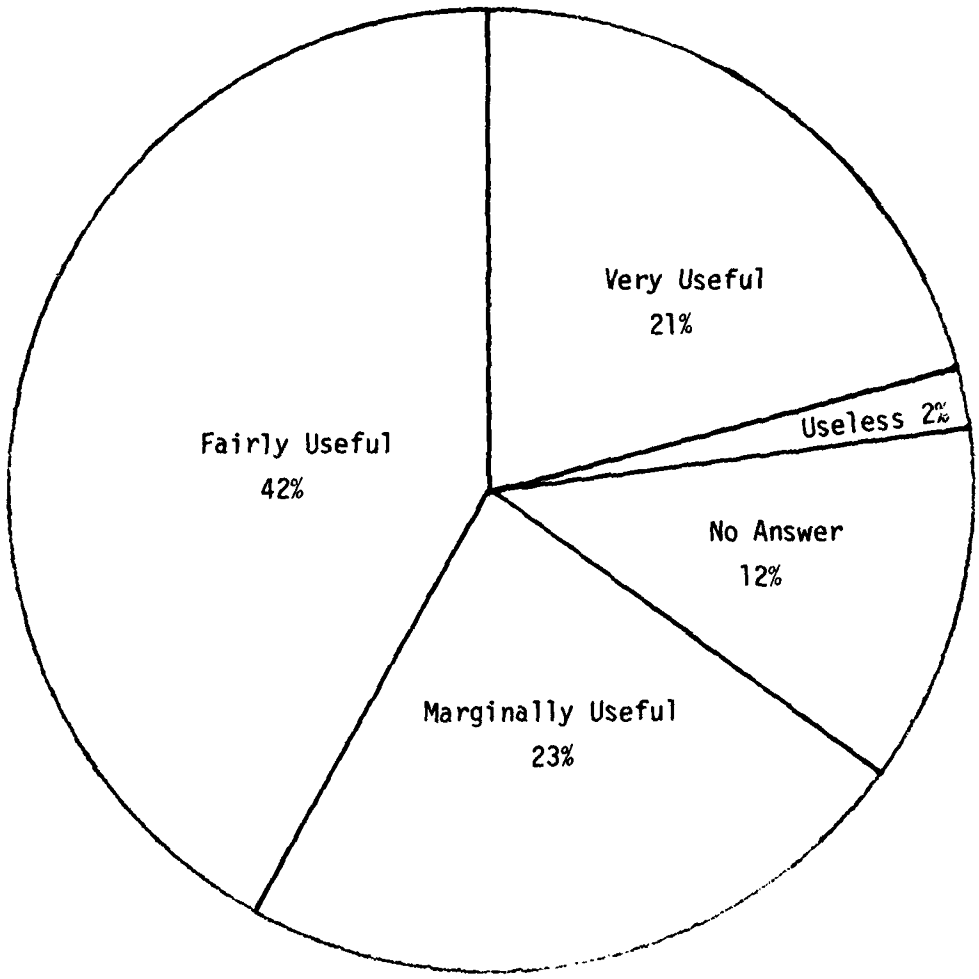


FIGURE 7.1-4 RATING LEVELS

Eight of the complimentary comments were from those individuals who had rated the products/services as marginally useful, and one who had not provided a rating.

There were also many requests for clarification on the products and services of the DACS and to be placed on the newsletter mailing list.

User Type vs Information Need, Products Used, and Usefulness Rating

There seemed to be very little difference between user type and the answers to questions 2, 3, and 4. Some of the more significant findings are discussed, although statistically there is no difference between the groups.

Forty-three percent of the researchers rated information on new software engineering research as their highest priority information need compared to 26% for managers and 22% for programmer/analysts. This compares to 40% of the managers who rated state-of-the-art reports as their highest information need, followed by 31% for programmer/analysts, and 30% for researchers.

More managers (97%) indicated that they had used the Newsletter than programmer/analysts (79%) or researchers (91%).

The greatest difference in the answers to the usefulness question was in the fairly useful categroy. Fifty-six percent of the researchers checked fairly useful, compared to 44% for managers and 27% for programmer/analysts. No researcher checked useless.

7.2 Task 10 Improving Center Effectiveness

7.2.1 Introduction

This section of the report provides an analysis of the inquiries processed during Phases I and II, a discussion on the processing of the documents for the software engineering bibliographic database, and a description of the activities planned for Phase III.

7.2.2 User Inquiries

7.2.2.1 Phase I

During Phase I (15 August 1978 to 1 April 1979) of the development of the Center, 91 technical and information inquiries were received and processed. The majority of these requests centered around more information about the Center. In answer to these inquiries, copies of two papers were distributed entitled "An Introduction to the Data and Analysis Center for Software" and "An Introduction to the DACS Software Engineering Bibliographic Database."

We were not well equiped, during this period, to provide much technical assistance; although we did produce, manually, some custom bibliographies and provided referrals to other sources of information. The table below illustrates the number of inquiries processed per month.

TABLE 7.2-I INQUIRIES/MONTH - PHASE I

	Month	Number
1978	October	2
	November	12
	December	12
1979	January	12
	February	20
	March	33

These inquiries do not reflect those users who requested to receive the DACS Newsletter. As of 1 April, there were 2143 names/organizations on the Newsletter mailing list.

7.2.2.2 Phase II

On 1 April 1979, we modified our method for recording and processing of user requests. We classified the requests into three basic categories:

- Newsletter Inquiries
- Document Inquiries
- Technical Inquiries

A Newsletter Inquiry was a request to be placed on the Newsletter mailing list; a Document Inquiry was a specific request for a DACS document or documents; a Technical Inquiry was a request for technical information which required some engineering analysis to respond. These technical inquiries we answered in a variety of ways, dependent upon the type of request, the depth and breadth of our information base on the subject area of interest, and the availability of engineering time. Custom bibliographic searches were performed, analyzed, and synthesized; DACS documents, or subsets of the DACS database, were distributed, and/or the user was referred to other sources of information or contacts.

Figure 7.2-1 provides information on the number of requests for each category per month. The total number of inquiries processed during this 10-month period (1 April 1979 to 1 February 1980) was 2162. This averages out to approximately 10 inquiries per day. The total number of inquiries by category were:

Technical	281
Newsletter	669
Document	1212
TOTAL	2162

The percentage of Technical Inquiries by institution type is listed below.

Requestor	Percent of Total Technical Inquiries
Government	32%
Industrial	60%
Academic	4%
Foreign	4%

This data was compared to the percentages from two DLA information analysis centers; Metals and Ceramics Information Analysis Center (MCIC - reference 19) and the Reliability Analysis Center (RAC - reference 20).

Requestor	Percent of Total Inquiries	
	RAC	MCIC
Government	19%	20%
Industrial	76%	74%
Academic	(not recorded)	3.5%
Foreign	14%	2.5%

One of the major benefits of an information center is to provide rapid response to requests for technical information. In analyzing the response time data for Phase II, we have not processed requests in a timely enough manner. Steps are being taken to alleviate these problems by assigning more flexible responsibilities so that work loads for individual members of the staff may be distributed more evenly.

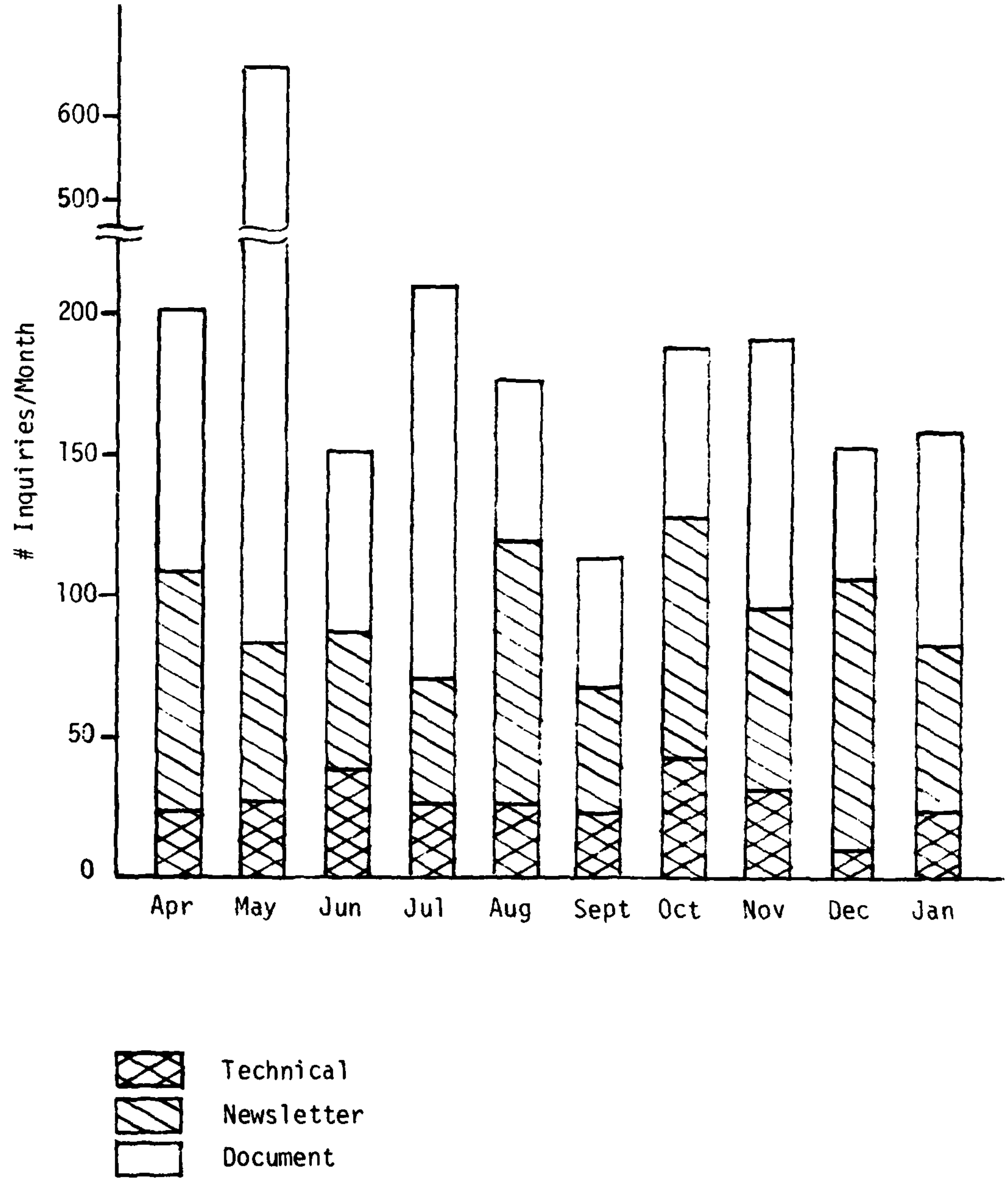


FIGURE 7.2-1 INQUIRIES/MONTH - PHASE II

Below is a summary of the number of working days to answer a request during this 10-month period.

Number of Working Days	Percent of Total Technical Inquiries
0- 4	61%
5- 9	16%
10-14	8%
≥ 15	15%

During December and January approximately 90% were answered within the first five days while in April and November only 20% were answered in the first five days.

Over 3200 documents were distributed in response to the 1200 document requests. Below is a list of the documents most requested.

Document Title	Number Distributed
Bibliographic Services, Custom Searches (BIB-I)	817
DACS Thesaurus	802
DACS Glossary (GLOS-1)	764
Quantitative Software Models (SRR-1)	452
An Introduction to the Data and Analysis Center for Software	195

The respondees to the DACS User Survey indicated a preference for state-of-the-art reports. They also indicated a low interest in custom bibliographies. This is somewhat out of line with the statistics on the requests for documents, in that, almost twice as many requests for the bibliographic services guide were received than for the state-of-the-art report on Quantitative Software Models. This may be explained in part by the fact that this report is in a specialized area.

Another mechanism for receiving feedback from users is to include a critique response form with each document and/or technical inquiry response. However, the response to this mechanism tends to be low, as reported in reference 15. The Defense Documentation Center reported on one sample of critique forms sent with 4262 documents. Only 98 of the forms were returned, or 2.3 percent. They indicate that a better rate is achieved only as a result of active follow-up by project personnel.

7.2.3 User Awareness

User awareness is an important aspect of center management in that it provides managers and technologists with clear and available descriptions of the products and services. During this reporting period we have disseminated this information through the newsletters, informational brochures, and participation in conferences and workshops (as discussed in Section IV).

DLA (as reported in reference 16) has distributed more than 30,000 copies of their User's Guide. But still they find that DoD personnel require information but do not know that it is available from the various Centers. Not enough of the leaders of the scientific and technical community are aware of an IAC. The DoD IAC directors and managers also believe that working engineers and scientists, who may have used IAC services, tend to forget; and need periodic reminders of the availability and value of IAC services (reference 17).

We have found that the response to vague definitions of our products and services has been very low, while the more succinct descriptions (as in the December Newsletter) produce a much higher response rate.

7.2.4 Information Processing

As reported in Section II, during this reporting period we have concentrated on establishing a technology information base while continuing to serve the user. During Phase III we will add, to this information base, compilations and descriptions of software engineering research projects. However, we are already encountering a large backlog in the processing of the software engineering literature.

The document processing described in Section 5.2.3 was developed and refined during the past year and tends to work well. The area that most needs improvement is in the amount of staff-hours devoted to the Log-in, Code, and Index steps. As the DACS becomes better known, we are receiving references to technical and research reports from our users which are appropriate for the collection and which ought to be ordered and processed. The number of journals and conference proceedings relating to software engineering is also increasing.

The backlog of documents queued at each of the first three steps of the document processing path increases each month. For this reason, the DACS has not initiated an aggressive document acquisition program - we do not have the staff to process what is on hand now; an increase in acquisitions without a corresponding increase in staff would only increase the backlog.

An analysis was performed to determine the staff time needed for each processing step and the number of documents backlogged at each step. This is summarized below.

Step	Best Time	Worst Time	Avg Time	No. Backlogged at This Step
Log-In	5 min	26 min	16 min	416 documents
Code	5 min	15 min	8 min	220 documents
Index	5 min	20 min	10 min	322 documents
Verify	2 min	10 min	3 min	36 documents
Keypunch	4 days	1½ weeks	1 week	none
Load	3 hours	1 week	5 hours	none

7.2.5 Phase III

As stated in Section II, Phase III of the development of the DACS will concentrate on providing more quality and indepth products and services. The following products are planned:

- The Second Edition of the DACS Glossary
- A report summarizing software engineering research projects
- Newsletters and Bulletins
- Two Software Engineering Bibliographies
- Two Technical Monographs
- Two Data Compendiums

In addition we plan to intiate an active data acquisition program and to enhance our data analysis, synthesis, and dissemination services.

SECTION VIII

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

- A solid base of information on software engineering has been developed.
- A framework for processing and synthesizing this information has been established.
- The products and services of the DACS are needed and can make a significant contribution to the development and maintainance of quality software.
- Information on software engineering is generally available - but is scattered, is in many different forms, and is difficult to synthesize for practical use and evaluation.
- The objectives established in the Software Data Repository Study are generally sound.
- There have been many levels of requests during this period from the very knowledgeable to the very inexperienced.

8.2 Recommendations

- Provide more indepth analysis of the DACS information and data and package these analysis results into technical monographs for general distribution.
- Establish an information base on software engineering research projects.
- Produce more distinct, descriptive literature on the products and services of the DACS.
- Initiate an active data acquisition program to expand the database for analysis and distribution.
- Obtain additional resources to be used for expanding the information base and for performing engineering analysis.

REFERENCES

1. Duvall, Lorraine M., Software Data Repository Study, Rome Air Development Center, RADC-TR-76-387, Griffiss Air Force Base, NY, December 1976. (A050636)

2. Rye, P., et al., Software Systems Development: A CSDL Project History, Rome Air Development Center, RADC-TR-77-213, Griffiss Air Force Base, NY, June 197?. (A042186)

3. Baker, W. F., Software Data Collection and Analysis: A Real-Time System Project History, Rome Air Development Center, RADC-TR-77-192, Griffiss Air Force Base, NY, June 1977. ((A041644)

4. William, H. E., Jr., et al., Software Systems Reliability: A Raytheon Project History, Rome Air Development Center, RADC-TR-77-188, Griffiss Air Force Base, NY, June 1977. (A040992)

5. Fries, M. J., Software Error Data Acquisition, Rome Air Development Center, RADC-TR-77-130, Griffiss Air Force Base, NY, April 1977. (A039916)

6. Thayer, T. A., Software Reliability Study, Rome Air Development Center, RADC-TR-76-238, Griffiss Air Force Base, NY, August 1976.(A030798)

7. Hecht, H., et al., Reliability Measurement During Software Development, NASA Langley Research Center, NASA-CR-145205, Hampton, VA, September 1977.

8. Duvall, Lorraine M., Baseline Software Data System, System Description, Vol. 1 (of two), Rome Air Development Center, RADC-TR-79-185, Griffiss Air Force Base, NY, July 1979. (A073358)

9. Duvall, Lorraine M., and Christine Curtis, Baseline Software Data System, Database Reference Manual, Vol II (of two), Rome Air Development Center, RADC-TR-79-185, Griffiss Air Force Base, NY, July 1979.(A073359)

10. Nelson, Richard, Software Data Collection and Analysis, Rome Air Development Center, Griffiss Air Force Base, NY, September 1978.

11. Musa, John D., "Validity of the Execution Time Theory of Software Reliability," *IEEE Transactions on Reliability*, August 1979.

12. Basili, Victor B., et al., "The Software Engineering Laboratory," NASA SEL, NASA Goddard Space Flight Center, Greenbelt, MD.

13. Martens, J. B., and L. M. Duvall, "BSDS Analysis and Data Parameters," IIT Research Institute, February 1979.

14. Computer Sciences Corporation, A Review of Software Maintenance Technology, Contract F30602-78-C-0255, Rome Air Development Center, Griffiss Air Force Base, NY, February 1980. (A083985)

15. Crum, F. B., et al., "Establishment and Operation of Pilot Warranty Information Center," Arinc Research Corporation, May 1977.

16. Meeting, Defense Logistics Agency - DLA-SCT, 11 March 1977.

17. Meeting of DoD IAC Directors/Managers - DLA-SCT, 17 December 1979.

18. Meyer, John D., "Analysis of the Roles and Benefits of a DoD Manufacturing Technology Information Analysis Center (MTIC), (Draft Report)," March 1, 1979.

19. Rubin, Stephin A., "Service Charging: An IAC Experience," The American Society for Information Science - The Value of Information, Syracuse, NY, May 1977, pp 156-163.

20. Ehrenfried, C. E., Operation of the Reliability Analysis Center, Final Report, IIT Research Institute, November 1978.

APPENDIX A

PARTIAL LIST OF DACS USERS

APPENDIX A

Partial List of DACS Users

UNITED STATES CORPORATIONS

Accutest Corporation
Chelmsford, MA

Advanced Systems Incoterm Corporation
Wellsley Hills, MA

Advanced Technology, Incorporated *
San Diego, CA

The Aerospace Corporation *
Los Angeles, CA

American Society of Civil Engineers
Kensington, MD

Analytics
McLean, VA

Argonne National Laboratory *
Argonne, IL

Arthur Andersen & Company
Chicago, IL

Automation Industry, Incorporated
Silver Spring, MD

Bell Laboratories
Holmdel, NJ

Bell Northern Research *
Palo Alto, CA

Bendix Research Laboratories *
Southfield, MI

Boeing Aerospace *
(several locations)

Burroughs Corporation *
(several locations)

CBS Publishing Group
New York, NY

CTEC Corporation *
Falls Church, VA

Calculon Corporation *
Falls Church, VA

Control Data Corporation *
(several locations)

Cinncinnati Electronics
Cinncinnati, OH

Computer Sciences Corporation *
(several locations)

Computer World
Newton, MA

Consolidated Freightways *
Portland, OR

Cutler-Hammer
Long Island, NY

Data Systems Technical Informations Ctr
New Haven, CT

John Deere Company *
Waterloo, IA

Digital Equipment Corporation
Maynard, MA

Distributed Systems Corporation
Chelmsford, MA

Doty Associates, Incorporated *
Rockville, MD

* Multiple Technical Inquiries

APPENDIX (continued)

Draper Laboratory *
Cambridge, MA

Douglas Aircraft Company *
Long Beach, CA

Dynamics Research Corporation *
Wilmington, MA

E-Systems, Incorporated
Falls Church, VA

EG&G, Idaho
Idaho Falls, ID

Evaluation Associates
Bala Cynwyd, PA

Ferguson-Bryan & Associates, Inc.
Washington, DC

Fireman's Fund Insurance Company
San Rafael, CA

Ford Motor Company
Dearborn, MI

Gagliardi Systems Group (GSG)
Rome, NY

General Dynamics *
(several locations)

General Electric Company *
(several locations)

General Research Corporation *
(several locations)

Grumman Aerospace Corporation *
Bethpage, NY

Harris Corporation
Melbourne, FL

Health Products Research, Inc.
Somerville, NJ

Honeywell, Incorporated *
(several locations)

Hughes Aircraft Company *
(several locations) CA

IIT Research Institute *
(several locations)

ITT Telecommunications Technical Ctr
Stratford, CT

International Business Machines *
(several locations)

Litton-Mellonics
Springfield, VA

Lockheed Georgia Company
Marietta, GA

Lockheed Missiles & Space Company, Inc.
Sunnyvale, CA

Logicon, Incorporated
San Pedro, CA

Manufacturing Data Systems, Inc.
Ann Arbor, MI

Martin-Marietta Corporation *
(several locations)

McCabe & Associates *
Columbia, MD

McDonnell Douglas Corporation *
(several locations)

Arthur G. McKee & Company
Cleveland, OH

Measurement Concept Corporation
Rome, NY

Memorex
Santa Clara, CA

* Multiple Technical Inquiries

APPENDIX (continued)

Mitre Corporation *
Bedford, MA

NCR Corporation *
San Diego, CA

OAO Corporation *
El Segundo, CA

Peat, Marwick, Mitchell & Company *
New York, NY

Perkin-Elmer *
Danbury, CT

Planning Research Corporation
McLean, VA

Price Waterhouse & Company *
New York, NY

Quantitative Software Management, Inc.*
McLean, VA

The Rand Corporation *
Santa Monica, CA

Raytheon Company *
(several locations)

Raytheon Service Company
Arlington, VA

Research Triangle Institute
Research Triangle Park, NC

Rockwell International *
Cedar Rapids, IA

RCA Government Communications System
Camden, NJ

SAI Comsystems Corporations *
San Diego, CA

Sheridan Oaks Cybernetics
Glenview, IL

Software Enterprises Corporation *
Westlake Village, CA

Softech Incorporated *
Waltham, MA

Software Management Consultants *
Torrance, CA

Software Research Associates
San Francisco, CA

SoHar Incorporated
Los Angeles, CA

Southern New England Telephone Company*
New Haven, CT

Sperry Rand Corporation *
(several locations)

Syntex Corporation
Palo Alto, CA

System Development Corporation *
Santa Monica, CA

Systems and Applied Sciences Corporation *
Riverdale MD

Systems Engineering Laboratories *
Fort Lauderdale, Florida

Sylvania, Incorporated *
Needham, MA

TRW *
Redondo Beach, CA

Teletype Corporation
Chicago, IL

Texas Instruments *
(several locations) TX

Time Share Corporation
Hanover, NH

* Multiple Technical Inquiries

APPENDIX (continued)

Union Carbide Company *
(several locations) NY

Westinghouse ILSD Division
Hunt Valley, MD

Xerox Corporation
Rochester, NY

DEPARTMENT OF DEFENSE

Defense Documentation Center *
Alexandria, VA

Defense Logistics Agency *
Alexandria, VA

US AIR FORCE

Air Force Data Services Center *
AFDSC/XMD Pentagon
Washington, DC

Air Force Reserve *
SD/CSB RES. DET. 1
Santa Clara, CA

Andrews Air Force Base *
AFSC/XRF
Washington, DC

Gunter Air Force Base *
Phase IV PMO/PGY
and
AF DSDC/SCDP
Montgomery, AL

Hanscom Air Force Base *
AFTL Laboratory
and
AFGL/SULLR
and
ESD/TOI, TOIS, TOIT, TOIQ, TOIT Hqs.ESD
Bedford, MA

Kirkland Air Force Base *
HQS AFCDM/QA
and
Air Force Testing Evaluation Center
NM

Langley Air Force Base
HQS TAC/ADYS
VA

McClellan Air Force Base *
Sacramento Air Logistics Center
Air Force Logistics Command
and
SM-ALC/MMM-1
and
ACD
CA

Maxwell Air Force Base
ACSC/EDOI
AL

Nellis Air Force Base *
Tactical Fighter Weapons Center
SA/M
NV

Offutt Air Force Base *
HQS SAC
ADOS
and
HQS SAC/ADXPO
Global Weather Central /DOY
Global Weather Central /ADSV
NB

Griffiss Air Force Base *
RADC ISIS, ISCP, RBC
Rome, NY

Robbins Air Force Base *
WR/ALC/MMECV
GA

* Multiple Technical Inquiries

APPENDIX (continued)

Wright-Patterson Air Force Base *
AFLC
and
AFAL/AAA-3
AFFA/ASD
ASD/YPAA
AFLC/ACMCE
OH

US ARMY

Department of the Army
Automated Logistic Management Systems Agencies
St. Louis, MI

Department of the Army
Naval Training Equipment Center
DRCPM-TND-PA
Orlando, FL

US Army Research Institute for the Behavioral and Social Sciences
Alexandria, VA

US NAVY

Naval Air Systems Command
Reliability/Maintainability
Washington, DC

Naval Research Code #474
Arlington, VA

Naval Research Laboratory *
Code 7906 & 7503
Washington, DC

Naval Surface Weapons Center *
Code K70
and
Dahlgren Laboratory
Dahlgren, VA

Naval Sea Systems Command
Code 63 E 11
Washington, DC

Naval Systems Weapon Command *
Code K74
Dahlgreen, VA

Naval Training Equipment Center *
Code N-4133
Orlando, FL

Naval Underwater Systems Center *
Codes 3545, 3552, 411, 715
Newport, RI

Commander Frank Boebert, USN
DCASR
NY NY

OTHER GOVERNMENT AGENCIES

Federal Aviation Administration
Margate, NJ

General Services Administration (GSA)
Automated Data & Telecommunication Service
Washington, DC

Department of Health, Education, & Welfare
Public Health Service
Bethesda, MD

NASA - AMES Research Center *
Moffett Field
CA

NASA, Goddard Space Flight Center *
Greenbelt, MD

National Bureau of Standards (NBS)*
Washington, DC

National Security Agency (NSA) *
Ft.George G. Meade
MD

Nuclear Regulatory Commission
Washington, DC

* Multiple Technical Inquiries

APPENDIX (continued)

Elementary Particle Information Ctr
Oak Ridge National Laboratoy
Oak Ridge, TN

US General Accounting Office *
(several locations)

US Environmental Protection Agency
Research Triangle Park
NC

US Department of Transportation
Office of Automated Systems Policy
Washington, DC

US Department of State
Fairfax, VA

UNIVERSITIES

State University of New York at Binghamton
Binghamton, NY

Case Western Reserve University
Cleveland, OH

Colorado State University
Ft. Collins, CO

Florida Atlantic University
Boca Raton, FL

Georgia Institute of Technology
Atlanta, GA

The John Hopkins University
Laurel, MD

New Mexico Technology
Socorro, NM

University of Rochester
Rochester, NY

University of Missouri *
Rolla, MI

University of Minnesota, MISRC
Minneapolis, MN

Syracuse University
Syracuse, NY

FOREIGN

Bell Northern Software Research (BNSR) *
Toronto, Ontario Canada

The Royal Bank of Canada
Montreal, Quebec, Canada

The Toronto-Dominion Bank
Toronto, Ontario Canada

Standard Bank of South Africa,Ltd
South Africa

CCSL
Systems Engineering Controller
Australia

Depto Metodoc Quantitativos
Brasil

GMD Institute for Software Technology
Germany

Hindustan Computers Limited
Bombay

Japan Software Industry Association
Tokyo

Elrand Information Systems Ltd.
Jerusalem

City of Toronto Computer Service *
Toronto, Canada

Vrije Universiteit
The Netherlands

* Multiple Technical Inquiries

Universidad Nacional del Sur
Argentina

Universite Paul Sabatier
France

Universita Degli Studi DI Milano
Milano, Italy

Fourth Transportation Brigade
Camp King, W.Germany

New South Wales Institute of Tech
Australia

Mr. Tom Gilb, Consultant
Norway

Marconi Avionics Limited
London

Ministry of State Urban Affairs
Ottawa, Ontario Canada

Nippon Electric Company Ltd.
Japan

Runit
Computing Centre at the University of
Trondheim, Norway

John Swire & Sons (U.K.) Ltd.
Hong Kong

Thomson - CSF *
Institut de Recherche d'Informatique
Le Chesnay, France

United Nations Industrial Development *
Organization (UNIDO)
Vienna, Austria

* Multiple Technical Inquiries

APPENDIX B

STATISTICS ON KEYWORD DISTRIBUTION

APPENDIX B

Statistics on Keyword Distribution

This Appendix contains an alphabetical listing of the 475 keywords in the DACS Thesaurus together with the number of documents assigned each keyword. There were 8164 keyword assignments made to 1346 online document descriptions, yielding an average of six keywords per document.

The keywords assigned most often are:

Keyword	Number of Instances
Support Software	121
Structured Programming	110
Reliability	103
Developmental Tools and Techniques	101
Development Management	98
Testing	91
Automated Verification Tools	89
Specifications	85
Program Correctness	83
Configuration Management	81

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
ABSTRACT MACHINES	7
ACCEPTANCE TESTING	9
ACCESSIBILITY	3
ACCESS-CONTROL MECHANISMS	17
ACQUISITION COSTS	6
ACQUISITION MANAGEMENT	64
ADAPTABILITY	5
ADAPTIVE TESTING	8
AIRBORNE WARNING AND CONTROL SYSTEM	2
ALGOL	5
ALGORITHM ANALYSIS	17
ALPHARD	2
APL (A PROGRAMMING LANGUAGE)	1
APOLLO FLIGHT SOFTWARE	1
APPLICATION-ORIENTED LANGUAGES	13
APPLICATIONS	17
ARCHITECTURAL FAMILIES	10
ARCHITECTURE	12
ARTIFICIAL INTELLIGENCE	4
ASSEMBLERS	12
ASSEMBLY LANGUAGE	2
ASSERTIONS	21
ASSIGNMENT STATEMENTS	2
AUDITS	6
AUT (AUTOMATED UNIT TEST)	3
AUTOMATA	2
AUTOMATA THEORETIC	1
AUTOMATED DESIGN TOOLS	11
AUTOMATED DOCUMENTATION	18
AUTOMATED ERROR DETECTION	25
AUTOMATED PROGRAM ANALYSIS	16
AUTOMATED TESTING	60
AUTOMATED VERIFICATION TOOLS	89

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
AUTOMATIC DATA COLLECTION	20
AUTOMATIC PROGRAMMING	28
AVAILABILITY	9
AVAILABILITY MODELS	4
AVIONICS APPLICATIONS	41
BALLISTIC MISSILE DEFENSE	18
BASIC	11
BAYESIAN MODELS	11
BEHAVIOR MODELLING	5
BOTTOM UP DESIGN	5
BUG SEEDING/TAGGING	11
BUILDS	3
BUSINESS AND FINANCIAL APPLICATIONS	2
CAPABILITY MACHINES	4
CERTIFICATION	6
CHEBYSHEV'S INEQUALITY	1
CHIEF PROGRAMMER TEAM	13
CLARITY	7
COBOL	18
CODE READING	7
CODE VERIFICATION	11
COMMAND LANGUAGES	3
COMMUNICATIONS SWITCHING SYSTEMS	21
COMMUNICATIVENESS	5
COMPILER-COMPILERS	8
COMPILERS	44
COMPLEXITY	39
COMPLEXITY MEASUREMENT	39
COMPUTATION STRUCTURES	5
COMPUTER COMMUNICATIONS NETWORKS	25
COMPUTER PROGRAM DEVELOPMENT PLAN (CPDP)	4
CONCISENESS	2
CONCURRENT PASCAL	9
CONCURRENT PROGRAMMING	29

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
CONDITION VARIABLES	2
CONFIGURATION MANAGEMENT	81
CONFINEMENT	2
CONTROL STRUCTURES	38
CONVERSION AIDS	5
CONVERSION COST FACTORS	6
CONVERSION COSTS	5
CONVERSIONS	16
CORRECTNESS PROOFS	61
COST	26
COST AND SCHEDULE CONTROL	60
COST EFFECTIVENESS	28
COST ESTIMATION	41
COST FACTORS	14
COST MODELS	5
COSTING TECHNIQUES	8
COST-BENEFIT ANALYSIS	49
CURRICULA	18
DAS	2
DATA	3
DATA ANALYSIS	31
DATA COLLECTION	45
DATA COLLECTION COSTS	3
DATA DOMAINS	17
DATA FLOWGRAPHS	26
DATA PARAMETEPS	32
DATA PROCESSING APPLICATIONS	35
DATA REPOSITORY	8
DATA RESTRUCTURING	3
DATA SEMANTICS	14
DATA STRUCTURES	51
DATA VALIDATION	3
DATABASE APPLICATIONS	46
DATABASE MANAGEMENT SYSTEMS	28

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
DEBUGGING	37
DEBUGGING MODELS	4
DESIGN	32
DESIGN ANALYSIS	41
DESIGN ANALYZERS	7
DESIGN LANGUAGES	11
DESIGN METHODOLOGIES	73
DESIGN REVIEWS	11
DESIGN TOOLS AND TECHNIQUES	71
DESIRED BEHAVIOR SPECIFICATION	4
DEVELOPMENT	17
DEVELOPMENT CYCLE	58
DEVELOPMENT MANAGEMENT	98
DEVELOPMENT METHODOLOGIES	64
DEVELOPMENT PROCESS	61
DEVELOPMENT SUPPORT LIBRARIAN	9
DEVELOPMENTAL TOOLS AND TECHNIQUES	101
DIFFICULTY	11
DIGITAL AIRCRAFT CONTROL	7
DISCRETE SYSTEMS ANALYSIS	1
DISTRIBUTED PROCESSINC	60
DOCUMENTATION	78
DOD COMMON HIGH ORDER LANGUAGE	18
DOMAINS	1
DREAM	4
DYNAMIC ANALYSIS	9
DYNAMIC REDUNDANCY	7
DYNAMIC RESTRUCTURING	2
EDUCATION	30
EFFICIENCY	30
EMBEDDED COMPUTER SYSTEMS	45
EMBEDDED LANCUAGES	25
EMULATION	17
EQUIVALENCE OF MONITOR IMPLEMENTATION	1

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD	KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
ERROR ANALYSIS	33	FAULT AVOIDANCE	4
ERROR CATEGORIES	24	FAULT DETECTION	6
ERROR CATEGORIZATION	26	FAULT TOLERANCE	10
ERROR CONFINEMENT	7	FAULT-TOLEPANT SOFTWARE	36
ERROR CORRECTION	11	FILE MODIFICATION	1
ERROR CORRECTION COSTS	17	FILE MODIFICATION PROTECTION	2
ERROR CORRECTION LIMIT POLICIES	6	FLEXIBILITY	4
ERROR DATA	26	FORTRAN	37
ERROR DATA ACQUISITION	7	FUNCTION TESTING	5
ERROR DATA ANALYSIS	19	FUNCTIONAL PROGRAMMING	4
ERROR DETECTION	76	FUNCTIONS	5
ERROR DETECTION CODE	2	GYPSY	2
ERROR DETECTION PROCESS	2	HAL/S	2
ERROR MANAGEMENT	8	HALSTEAD'S LAW	11
ERROR MODELS	40	HARDWARE RELIABILITY	4
ERROR PREVENTION	7	HAZARD FUNCTION	7
ERROR SEVERITY	3	HEURISTIC SEARCH	3
ERRORS	19	HIERARCHIAL STRUCTURE	16
EUCLID	3	HIERARCHY	9
EVENT-BASED MODELS	2	HIGH LEVEL LANGUAGES	68
EVOLUTIONARY SYSTEMS	5	HIPO (HIERARCHY PLUS INPUT-PROCESS-OUTPUT)	6
EXECUTION TIME	1	HOLDET	1
EXECUTION TIME THEORY	5	HUMAN ENGINEERING	26
EXPONENTIAL DISTRIBUTIONS	3	HYPERGEOMETRIC DISTRIBUTIONS	1
EXPOSURE	1	IMPERFECT DEBUGGING	8
EXTENSIBILITY	8	IMPLEMENTATION	33
FAILURE CATEGORIZATION	2	IMPLEMENTATION CORRECTNESS	5
FAILURE DATA	8	INDUCTIVE ASSERTIONS	11
FAILURE RATES	14	INDUSTRIAL PROCESS APPLICATIONS	11
FAILURE RATIO	16	INFERENCE RULES	1
FAILURES	14	INFORMATION HIDING	7
FAIRNESS	2	INFORMATION SYSTEMS	29
FAST	1	INPUT-DATA SENSITIVITY	1
FAULT	6	INTERFACE CONTROL	1

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
INTERMITTENT ASSERTION METHOD	1
INTEROPERABILITY	1
INTERPRETATION CORRECTNESS	2
INTERPROCESS COMMUNICATIONS	11
INVARIANT ASSERTIONS	13
ISRAD	2
JAVS	4
JELINSKI-MORANDA MODEL	6
JOINT LOGISTICS COMMANDERS	1
JOVIAL	15
KNOWLEDGE BASED SYSTEMS	4
LANGUAGE DESIGN	38
LANGUAGE EVALUATION	69
LANGUAGE STRUCTURE	5
LANGUAGES	20
LEGIBILITY	6
LEIGHTON DIAGRAMS	2
LIFE CYCLE COSTS	48
LINKAGE EDITORS	5
LIST PROCESSING	2
LOADERS	3
LOGISTICS APPLICATIONS	3
MACROPROCESSORS	3
MAINTAINABILITY	24
MAINTENANCE	48
MAINTENANCE COSTS	20
MAINTENANCE TOOLS	18
MANAGEMENT	33
MANAGEMENT TOOLS AND TECHNIQUES	61
MARKOV MODELS	7
MAXIMUM LIKELIHOOD	4
MEASUREMENTS	34
MECHANICAL DEDUCTION	1
MEDICAL APPLICATIONS	4

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
MEMORY MANAGEMENT	16
MESSAGE TRANSFER MODELS	1
METALANGUAGE	8
MICRO COMPUTERS	22
MICRO RELIABILITY MODELS	2
MICROCODE	3
MICROPROCESSORS	41
MICROPROGRAMS	11
MINICOMPUTERS	21
MODELLING AND SIMULATION TOOLS	77
MODELS	38
MODERN PROGRAMMING PRACTICES	43
MODIFIABILITY	8
MODIFICATION	11
MODIFICATION PROCEDURES	31
MODULA	1
MODULAR DECOMPOSITION	25
MODULAR PROGRAMMING	28
MODULARITY	39
MODULARIZATION	14
MODULE ANALYSIS	4
MODULE SIZING	1
MODULE TESTING	13
MODULES	3
MONITORS	20
MULTICS	3
MULTIPROGRAMMING	11
MUSA'S MODEL	4
MUST	6
MUTUAL EXCLUSION	2
NATURAL LANGUAGE	4
NATURAL LANGUAGE PROCESSING	7
NATURAL LANGUAGE THEORY	5

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
NETWORKS	20
NON-PROCEDURAL SPECIFICATION	2
N-VERSION PROGRAMMING	1
NUCLEAR REACTOR APPLICATIONS	6
OPAL	1
OPERATING SYSTEM DESIGN	24
OPERATING SYSTEMS	37
OPERATIONAL RELIABILITY	6
OPTIMIZATION	30
PARTITIONING	7
PASCAL	31
PATH EXPRESSIONS	13
PERFORMANCE	31
PERFORMANCE EVALUATION	35
PETRI NETS	6
PL/I	10
POINTERS	2
POISSON PROCESSES	5
PORTABILITY	37
PRIVILEGE	10
PROBLEM REPORT ANALYSIS	3
PROCEDURES	7
PROCESS	1
PROCESS CONSTRUCTION	4
PROCESS DESIGN LANGUAGE (PDL)	2
PROCESS QUEUES	4
PRODUCT LIABILITY	1
PRODUCT SAFETY	1
PRODUCT SAFETY EVALUATION	1
PRODUCTIVITY	18
PRODUCTIVITY DATA	3
PRODUCTIVITY FACTORS	24
PROGRAM ANALYSIS	31
PROGRAM ANNOTATION	1

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
PROGRAM COMPLEXITY	20
PROGRAM CONTRACTION	1
PROGRAM CORRECTNESS	83
PROGRAM DESIGN	17
PROGRAM DESIGN LANGUAGE (PDL)	15
PROGRAM DESIGN METHODOLOGIES	24
PROGRAM EXTENSION	4
PROGRAM FAMILIES	7
PROGRAM LIBRARY SYSTEMS	6
PROGRAM MODIFICATION	8
PROGRAM PROTECTION	1
PROGRAM SYNTHESIS	17
PROGRAM TESTING	51
PROGRAM TRANSFORMATIONS	19
PROGRAM UNDERSTANDING	6
PROGRAM VALIDATION	11
PROGRAMMER PRODUCTIVITY	48
PROGRAMMER'S APPRENTICE	3
PROGRAMMING	26
PROGRAMMING AIDS	38
PROGRAMMING LANGUAGE	67
PROGRAMMING LANGUAGE COMPLEXITY	8
PROGRAMMING TECHNIQUES	77
PROGRAMS	3
PROJECT MANAGEMENT SURVEYS	7
PROTECTION	19
PROTECTION MEASUREMENT	2
PROTOCOLS	11
QUALITY	12
QUALITY ASSURANCE	37
QUALITY FACTORS	68
QUALITY METRICS	28
QUEUING	9
RAYLEIGH DISTRIBUTION MODELS	5

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
REAL-TIME SYSTEMS	41
RECOVERY	20
RECOVERY BLOCK STRUCTURED SOFTWARE	9
RECURSION	2
REGRESSION TESTING	6
RELIABILITY	103
RELIABILITY ESTIMATION	18
RELIABILITY EVALUATION	10
RELIABILITY MEASUREMENT	36
RELIABILITY MODELS	47
RELIABILITY PREDICTION	19
RELIABILITY--DIFFERENCES OF OPINION	2
REQUIREMENTS	59
REQUIREMENTS ANALYSIS	43
REQUIREMENTS ENGINEERING	26
REQUIREMENTS ENGINEERING METHODOLOGIES	12
REQUIREMENTS LANGUAGES	23
REQUIREMENTS PROBLEM CATEGORIES	1
REQUIREMENTS PROBLEMS	3
REQUIREMENTS SPECIFICATIONS	20
RESERVATIONS AND DISPATCHING APPLICATIONS	3
RESILIENCY	2
RESOURCE ALLOCATION	32
RESTRUCTURING ALGORITHMS	1
RESYNCHRONIZATION	1
REUSABILITY	9
ROBUSTNESS	3
ROLLBACK	7
ROUTING DATA	2
SAMM (SYSTEMATIC ACTIVITY MODELING METHOD)	4
SARA (SYSTEMS ARCHITECT'S APPRENTICE)	1
SCHEDULE ESTIMATION	15
SCHEDULING	17
SCHICK-WOLVERTON MODEL	5
SECURITY	23
SELF-DESCRIPTIVENESS	7
SEMANOL	4
SHOOMAN MODEL	4
SIGNALLING	1
SIMULA	1
SIMULATORS	8
SIZING	11
SLICE	1
SNOBOL (AND SNOBOL EXTENSIONS)	4
SOFTWARE	3
SOFTWARE DATABASE	20
SOFTWARE ENGINEERING	76
SOFTWARE ENGINEERING BLUEPRINTS	3
SOFTWARE ENGINEERING FACILITY	16
SOFTWARE ENGINEERING PROJECT MANAGEMENT	17
SOFTWARE ENGINEERING STANDARDS	11
SOFTWARE ENGINEERING TOOLS AND TECHNIQUES	61
SOFTWARE EXPERIENCE DATA	19
SOFTWARE LIBRARY MANAGEMENT SYSTEM	1
SOFTWARE LIFE CYCLE	39
SOFTWARE PHYSICS	6
SOFTWARE REQUIREMENTS ENGINEERING PROGRAM	2
SOFTWARE SCIENCE	16
SOFTWARE SNEAK ANALYSIS	1
SPARCS-II	1
SPECIFICATION LANGUAGES	2
SPECIFICATION TOOLS AND TECHNIQUES	33
SPECIFICATIONS	85
SPECIFICATIONS DECOMPOSITION	2
SPECIFICATIONS VERIFICATIONS	20
STANDARDIZATION	31
STANDARDS	53
STATE DIAGRAMS	3

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
STATE MACHINES	14
STATIC ANALYSIS	15
STATIC REDUNDANCY	8
STATISTICAL PREDICTION	1
STATISTICAL TEST MODELS	1
STEPWISE REFINEMENT	24
STESD	1
STRUCT	3
STRUCTURED DESIGN	31
STRUCTURED PROGRAMMING	110
STRUCTURED PROGRAMMING LANGUAGE	11
STRUCTURED WALK-THROUGH	1
STRUCTUREDNESS	1
STRUCTURE-CHANGING PROGRAMS	1
SUBMARINE APPLICATIONS	4
SUPERVISORY PROGRAM	4
SUPPORT SOFTWARE	121
SYMBOLIC EXECUTION	13
SYNCHRONIZATION	20
SYNTHESIZERS	1
SYSTEM ARCHITECTURE	14
SYSTEM DEPENDENCIES	1
SYSTEM DESIGN	76
SYSTEM DESIGN REQUIREMENTS	2
SYSTEM ENGINEERING LANGUAGE	4
SYSTEM INTEGRATION	3
SYSTEM RELIABILITY	11
SYSTEM STRUCTURING	3
SYSTEM TESTING	16
SYSTEM VALIDATION	5
TACTICS	1
TARGET LANGUAGE	1
TELECOMMUNICATIONS APPLICATIONS	10
TERMINATION PROOFS	7

KEYWORD	NO. DOCUMENTS INDEXED USING THIS KEYWORD
TEST DATA GENERATION	61
TEST DESIGN	9
TEST DRIVERS	5
TEST MANAGEMENT	20
TEST METHODOLOGIES	54
TEST PLAN DOCUMENT	15
TESTABILITY	2
TESTEDNESS	13
TESTING	91
TEXT-FORMATTING APPLICATIONS	1
TEXT-PROCESSING APPLICATIONS	2
TIME DOMAINS	18
TOP DOWN DESIGN	29
TOP DOWN DEVELOPMENT	12
TOP-DOWN IMPLEMENTATION	3
TOP-DOWN PARSING	2
TOP-DOWN PROGRAMMING	12
TOTAL CORRECTNESS	4
TRAFFIC CONTROL APPLICATIONS	3
TRI-SERVICE	18
UNIX	13
USER-INTERACTIVE SYSTEMS	23
UTILITY INDUSTRY APPLICATIONS	2
VALIDATION	47
VERIFICATION	75
VERIFICATION TOOLS	54
VIKING MISSION	3
VIRTUAL MACHINE MONITOR	3
VIRTUAL MACHINES	16
WALK THROUGH	3
WEAPONS SYSTEMS APPLICATIONS	3
WEIBULL DISTRIBUTION	2
WELLMADE	1
WORKING SETS	3
ZIPF'S LAWS	4

APPENDIX C

THE ROLE OF AN INFORMATION ANALYSIS CENTER IN SOFTWARE ENGINEERING TECHNOLOGY TRANSFER

A Concept Paper

THE ROLE OF AN INFORMATION ANALYSIS CENTER

IN SOFTWARE ENGINEERING TECHNOLOGY TRANSFER

A Concept Paper

The Predicament

As software engineering advances into its second decade, the ideas, principles and practices conceived in its first decade need to be assimilated into a workable set of tools and techniques that can be dispersed to software developers for their use in the production of software.[1] The need for this technology transfer is clear and immediate. Without the proper transfer of software engineering technology from software engineering researcher to developer, the software world will be unable to extricate itself from its present predicament. This predicament has been characterized by Meyers as follows:

> "The general character of the software predicament can be seen clearly, although consistent numbers with which to characterize it more precisely are hard to come by. Because less expensive hardware is bringing more applications within economic reach, the amount of software to be developed is increasing. Also, because more software is already in existence, there is more to be maintained. But the productivity of programmers is improving rather slowly, expecially by the standards of hardware price/performance, with the result that the overall cost of software development is tending to increase."[2]

The predicament presents a rather ominous picture to be sure. Essentially, the problems of today in the software world are more difficult, but the solutions to the problems do not seem to be effective. The result is a losing battle if things continue as they have. One reason for the losing battle may be that the software world is trying to solve today's problems with yesterday's solution techniques. For example, several recent surveys have shown that the transfer of software engineering is at a standstill and that people are still developing software as they were five years ago.[3]

A Possible Solution

" I firmly believe that technology transfer is the primary means we have to combat the software problems the industry has been experiencing."[4]

Riefer's above statement points a way to the beginning of a solution to the software predicament that exists today. Technology transfer needs to bring new and effective solutions from the software engineering researcher to the software developer. Unfortunately, the process is not quite as easy as it sounds. Setting up direct communication links between researcher and developer will not insure that technology is transferred over these links. The wholesale importation of the latest software engineering techniques, in fact, just invites disaster. The developer needs to understand the techniques so they can be evaluated within the context of the development environment. This is essential. Another way of stating this is by citing Reifer's Technology Risk Principle.

"Technology should only be used when the risk associated with it is acceptable."[5]

Some of the technologies may indeed be acceptable in terms of risk; however, the developer needs to understand the technology and all of its myriad applications before any consideration can be made to its transfer to the development environment. This technology transfer business is more than giving lectures or writing journal articles about the latest technologies. It involves a certain amount of information synthesis and analysis so that the ultimate receiver can evaluate the technology's worth. If no benefit is perceived by the developer, no transfer of technology is going to occur. Of that we can be certain.

Some Solutions Mechanisms

Although the technology transfer process is a difficult one, there are mechanisms in place today within the software engineering community to effect the transfer of technology. Wasserman, for example, cites four major mechanisms (and their attendant shortcomings).[6]

1. University graduates go to work in software development settings

Problem: New graduates generally have positions of low visibility and responsibility and have not received any development experience within the university environment.

2. University faculty serve as consultants to industry.

Problem: Consultants often opt for more "interesting" (i.e. more research-oriented) situations as they come about and have little continuity within one company. They also lack the same software development expertise that their students lack.

3. Industry people go to the university.

Problem: Sabbaticals are not open to many people and, besides, they are rarely taken by people with major project responsibilities.

4. Industry people attend short professional development courses.

Problem: The direct application of the techniques learned in these courses is often difficult to transfer to a typical work situation.

These four techniques all involve some level of inter-personal dealings in an environment different than one the person is used to. Because the performance of individuals in such a situation has such a wide variance, it is difficult to obtain a consistent level of technology transfer. The crucial process of transferring the context of ideas generated in a researcher environment to a developer environment is a difficult achievement when attacked on a one-to-one basis. Nevertheless, these techniques should be effective if the "right" person for the job can be identified.

Studies in scientific and technical information dissemination have identified such a person.[7] Called a "gatekeeper," this person has the important technology transfer ability to interface the outside world (i.e. the research community) to the inner realm of the development organization. Much of the time, however, these people tend to be senior technical staff and not high level managers who could influence the company to adapt a new software engineering technology.

The professional societies and the journal literature serve as an aid to technology transfer in a more formal manner than the inter-personal techniques. For example, many of the societies arrange and conduct tutorials to disseminate information about software engineering technologies. Along the same lines, the journals of the professional societies publish articles about the latest techniques. Although these techniques are formal, their effectiveness is limited by the very structure of the professional groups and journals.

Professional groups, by their nature, are essentially special interest groups created to advance the interest and ideas of a relatively narrow area of interest. For this reason, the societies and their journals are more for the purpose of intra-group rather than inter-group communication. Both researcher and developer have their oun societies, and both sides recognize the need and importance of the technology transfer issue. But the very structure of any society is not designed to facilitate a process such as technology tranfer. The required interface mechanism between developer and researcher is not strong for the societies or the journal literature.

Information Analysis Center as Technology Transfer Agent

A formal mechanism that can perform the interfacing role between researcher and developer for the purpose of technology transfer in the software world does exist. The mechanism is the information analysis center. Specifically, the information analysis center for software engineering is named the Data and Analysis Center for Software, hereafter referred to as DACS.

The purpose and objective of an information analysis center transcend those of a technical information center or library. The technical information center and library provide bibliographic services as their main commodity. Information analysis centers, as their name may imply, provide information analysis and synthesis services as their primary commodity. It is these analysis and synthesis services that serve as the core of the technology transfer mechanism within the information analysis center. As has been previously discussed, a precondition for the transfer of software engineering technology from the researcher to the software developer is a thorough understanding of the technology and its attendant implication in terms of risks and benefits within the developer's environment. It is this essential pre-condition to the technology transfer process that can be provided by the synthesis and analysis of information within the information analysis center. Although the previously discussed mechanisms are all useful to some extent, they do not possess the match of capability and task that exists between the information analysis center and the technology transfer task. In software engineering, the requisite synthesis and analysis skill are provided by the DACS as the software engineering information analysis center.

The Information Analysis Center Concept

The formal concept of the information analysis center was expressed in 1963 by the nuclear physicist Alvin Weinberg.[8] Weinberg had prepared a report at this time which was to serve as somewhat of a landmark document in the field of national scientific and technical information policy. The information analysis center concept did not originate in the report; in fact, the report listed over 400 organizations in the nation it considered as meeting the criteria of an information analysis center.[9] Highlighting the contribution information analysis centers could have in managing the nation's technical information was one of the major concerns of the report. An excerpt from the report explains the worth of the information analysis center concept.

> "The activities of the most successful (information analysis) centers are an intrinsic part of science and technology. The centers not only disseminate and retrieve information, they create new information. .... In short, knowledgeable scientific interpreters who can collect relevant data, review a field and distill information in a manner that goes to the heart of a technical situation are more help to the overburdened specialist than a mere pile of relevant documents."[10]

Although this excerpt does not explicitly mention technology transfer, that process of "...distilling information in a manner that goes to the heart of a technical situation" is certainly central to the process of technology transfer. The receiver of the technology must be able to understand and evaluate the technology on his own terms. The analysis and synthesis of information about a technology by the information analysis center matches the need and is the driving force behind the center's ability to act as an effective technology transfer agent. The fit between technology transfer and the information analysis center is a natural one; the statement Weinberg made about the value of the synthesis and analysis process is as valid and crucial today in the area of software engineering as it was in 1963 when the statement was made. By synthesizing and analyzing software engineering information into a form that is comprehensible and relevant to the software developer, the DACS has an important role as technology transfer agent in the field of software engineering. This task of synthesizing information in software engineering is more difficult than it might be with other fields because of the breadth and dynamic state of software engineering at the present time. But these very characteristics of software engineering may result in bigger payoffs when the technologies are transferred than may be the case with a narrower or more static discipline. The task is harder, but the potential benefits resulting from software engineering technology transfer may be greater.

A Two-Way Street

Technology transfer is generally thought of as a one-way street from researcher to developer. Simply, technology, which originates with the researcher, is transferred to the software developer who is the ultimate user of the technology. In its broadest sense, however, technology transfer is a two-way street. It is a two-way street because the researcher needs the experience of the developer for guidance in future research efforts. Both good and bad experiences with technology provide the researcher with valuable information to [illegible] technology.

Information analysis centers are good places for the developer to transfer software experience information. Just as the information analysis center interfaces between the transfer of technology from researcher to developer, it can provide the interface for the flow of experience data from developer to researcher. The synthesis and analysis skills of the information analysis center are just as useful and relevant for both directions of information flow. A large part of the DACS effort is expended in this regard, particularly in the organization of software experience data into databases that can be used for the evaluation and analysis of software engineering technologies.

DACS - A Software Engineering Information Analysis Center

In June of 1975, the Rome Air Development Center (RADC) contracted with IIT Research Institute (IITRI) to design a center that would acquire, analyze, and disseminate information on software engineering technology. The Air Force recognized the need for such a center to serve the government, industrial, and university comunity as a focal point for software development and experience data.

DoD has traditionally recognized the worth of information analysis centers. Several organizations within DoD sponsor a number of centers. For example, nine DoD Information Analysis Centers are managed and funded by the Defense Logistics Agency, (DLA). In keeping with the nature of all information analysis centers, the centers are responsible for the acquisition, analysis, evaluation, and dissemination of scientific and technical information to the managers, scientists, engineers, and technicians they support. Among the specialized areas the centers deal with are electronic hardware reliability, metals and ceramics, and machinability. As part of its responsibilities, each center serves as an information and technology transfer agent within its own area of technical expertise.

A contract to establish the DACS was awarded to IIT Research Institute (IITRI) by RADC in August 1978. When fully implemented and operational, the DACS will provide a centralized source for current, readily-usable data and information concerning software technology. This software information resource will:

1. Aid the program manager in planning and monitoring software projects.
2. Supply experience data to software research projects.
3. Provide baselines for software development methods comparisons.
4. Foster the use of uniform terminology.
5. Aid in establishing data collection guidelines and standards.
6. Distill and disseminate information on software projects.

The benefits expected to be accrued by members of the software engineering community, both developer and researchers, are:

1. Valuable savings of scientific and engineering manhours in locating data and information.
2. Rapid application of the latest technologies via technology transfer.
3. Elimination or minimization of duplication of effort.
4. Reduction of software costs and improved performance.
5. Minimization of program delays and schedule stretchouts.

All of the objectives of the DACS are related to the process of technology transfer. Through the first objective, the DACS will aid the technology transfer process by supplying information about software technologies to the developer in an understandable format. The second objective deals with the feedback from developer to researcher, as discussed earlier, that is required for refinement of technologies. Objective Number 3 will facilitate technology transfer by providing a basis for the evaluation of technologies in developer-related terms. Objectives 4 and 5 are objectives that will streamline the technology transfer mechanism. Uniform terminology (objective 4) and data collection guidelines (objective 5) are essential to aid communication between researchers and developers. Objective 6 is, in itself, the essence of the information analysis center technology transfer process - the ability to synthesize and distill information about the technology.

The benefits of the DACS, when they are realized, will result in efficient and effective technology transfer (benefits 1-3) and a start at escaping from the software predicament which the Air Force, the Department of Defense, and the entire software community are facing (benefits 4 and 5). Although these benefits are ambitious, the framework of the information analysis center provides an excellent mechanism for achieving these goals.

How DACS Operates

Objectives cannot be fulfilled or benefits realized unless a methodology to achieve these ends is established and followed. Information analysis centers serve technology transfer by synthesizing, distilling, analyzing and repackaging information. The DACS, as an information analysis center technique, provides a mechanism for synthesizing and analyzing the information concerning software engineering technology. Since the DACS is currently a pilot facility, the techniques are just being formatted, utilized, and tested. Although these techniques are still in a test mode, they should be adequate as they follow the techniques used for other technical IAC's with a fair amount of success. A functional model of the techniques is shown in Figure 1. [11,12]

DACS Activities and Products

Two major components make up the technique represented in Figure 1: (1) building an information base about software engineering technology and, (2) transferring information about the technology in a form that can readily be understood, evaluated, and used to the advantage of the software developer by the processes of information analysis and synthesis. Each of the major components consists of several sequential processing steps with each processing step resulting in the output of a particular information type. As a pilot facility, the DACS has more experience with the earlier steps, but all steps have been utilized to some extent.

Building the technology information base has been a major effort and concern of the DACS to date. This process consists of (1) information collection and, (2) information organization. From a technology transfer viewpoint, the purpose of these two steps is to prepare the information base so that it can later be synthesized into a form more suitable for technology transfer.

Information collection is being actively pursued by the DACS. The professional society publications are reviewed, as are conference proceedings: reports on new research projects, trade journals, and technical reports from universities, government, and industry. Information on new and previous software engineering research is the primary result of this review and collection procedure. Software experience data is also being collected by the DACS. At the present time, seven major datasets have been assembled and two are currently being assembled. Collection of all this information is, of course, necessary prior to any synthesis or analysis of the data that is required for technology transfer.

Information organization is the next step in the process of building a technology information base. Information management skills, such as indexing, abstracting, and information storage and retrieval, are used in this phase. The information collected in the first step is indexed, abstracted, and entered into a computerized information retrieval system. Custom bibliographies can then be produced by the retrieval system by specifying subject keywords or by qualifications on other fields such as author or title. These bibliographies are of use to the research community and developers in locating results of research. At this point in the process, the information is organized as final preparation for the steps of information synthesis and analysis that are so important to the information analysis center's contribution to technology transfer. These first two steps (information collection and organization) are performed by most technical libraries. The next two steps, however, separate the information analysis center from the technical library.

Information synthesis, the next step, is the core of the technology transfer process. Information generated by the researcher must be digested and made palatable for the developer. If at all possible, the information must be presented in a format to allow cost-benefit assessment by the developer. Depending on the novelty and age of the technology, this may not always be possible. At the very least, however, the information must be distilled so the basic concepts and principles are explained in a language that the developer can comprehend. This is not an easy task. Considerable skills in communications and expertise in the technology are required to produce the handbooks and state-of-the-art reports that are the outputs of this technology transfer process.

DACS has produced a state-of-the-art report on quantitative software models.[13] Research in this area has been extensive and has the potential for being used by developers, so this subject area was a prime candidate for this initial effort. With these thoughts in mind, the state-of-the-art report was produced with two major features to facilitate technology transfer. The first feature was a description of the salient characteristics of the model, such as data parameters, key equations and relationships, and experiences in using the model. This feature enables the developer to understand the model's concepts and capabilities.

Synthesis of information was carried one step further to produce the second feature of the report. A matrix was prepared to correlate model with data parameters. By using this matrix, a developer can quickly determine what models could be used with the data parameters available to the developer. If data parameters were unavailable, the cost of collecting them could be weighed against the benefits of the model as presented in the description.

Additionally, DACS has published a glossary of software engineering terms.[14] This glossary should aid the technology transfer process by providing a reference point for uniform terminology.

One last step is included in the technology transfer process: information analysis. Analysis goes one step beyond synthesis by providing an evaluation of the technology. As the center makes the transfer from pilot to full-scale operation, this analysis effort will be pursued. One target of analysis that DACS would like to examine is the effectiveness of modern programming practices. At the present time, data sources for such an effort are being collected and organized.

User Perceptions

A recent survey mailed with the DACS Newsletter is currently being analyzed. One interesting result in the light of the role of the DACS as technology transfer agent, is the interest of the respondents in state-of-the-art reports. These reports, as previously mentioned, are one of the primary products of the DACS.

Questionnaire respondents were given a list of the seven types of information processed and/or generated by an information analysis center and were asked the following:

> "For your job, please rate the value of the following types of information. (1 = most valuable)"

Table I summarizes the results. Although these results are preliminary, the results point to a definite preference for surveys of current technologies. The final results of the survey will be used to plan future DACS services and products.

Conclusion

The need for technology transfer in software engineering is clear and essential if the software world is to escape from its present predicament. Various methodologies and mechanisms exist for this transfer process. One process, using an information analysis center as a transfer agent, is appealing because the capability of the information analysis center as synthesizer of technology fits the task of transferring technology in such a way that the software developer and user can understand and evaluate the technology.

As a software engineering information analysis center, the DACS is fulfilling the role of transfer agent. Mechanisms are in place to synthesize the software engineering technology and disseminate it to software developers. During its operation, DACS has built an information base of software engineering and used that base to synthesize the technology into several handbooks and state-of-the-art reports. Response to the reports have been encouraging, and the DACS plans to issue more in its role as a software engineering technology transfer agent.

REFERENCES

1. Wasserman, Anthony and L. A. Belady with contributions from Susan L. Gerhart, Edward F. Miller, Jr., William Waite and William A. Wulf. "Software Engineering: The Turning Point." Computer. September, 1978, p. 30-41.

2. Meyers, Ware. "The Need for Software Engineering." Computer. February, 1978, p. 12-26.

3. McClure, Robert M. "Software - The Next Five Years." Digest of Papers Compcon Spring 76. p. 607, as cited in Ware Myers, "The Need for Software Engineering." Computer. February, 1978, p. 12-26.

4. Reifer, Donald. "The Nature of Software Management: A Primer." In: Donald Reifer, Tutorial: Software Management. IEEE Computer Society, March 1979, p. 5.

5. ________. "The Nature of Software Management: A Primer." In: Donald Reifer, Tutorial: Software Management. IEEE Computer Society, March 1979, p. 5.

6. Wasserman, Anthony and L. A. Belady with contributions from Susan L. Gerhart, Edward F. Miller, Jr., William Waite and William A. Wulf. "Software Engineering: The Turning Point." Computer. September, 1979, p. 30-41.

7. Paisley, W. "Information Needs and Uses." In: C. A. Cuadna. Annual Review of Information Science and Technology. Vol 3, Encyclopedia Britannica, Chicago, as cited in Laruen B. Doyle, Information Retrieval and Processing. Melville Publishing Company, Los Angeles, 1975, p. 390.

8. Weinberg, Alvin. "Science, Government, and Information." A Report of the President's Science Advisory Committee, Panel on Science Information, January, 1963, as cited in Lorraine Duvall, Software Data Repository Study. RADC-TR-76-387, December, 1976, p. 1-1.

9. ________. "Science, Government, and Information." A Report of the President's Science Advisory Committee, Panel on Science Information, January, 1963, as cited in Lorraine Duvall, Software Data Repository Study. RADC-TR-76-387, December, 1976, p. 1-2. (A050636)

10. Weinberg, Alvin. "Science, Government, and Information." A Report of the President's Science Advisory Committee, Panel on Science Advisory Committee, Panel on Science Information, January, 1963, as cited in, Alan Rees, "Functional Integration of Technical Libraries, Information Analysis Centers." In: Alan Rees, Contemporary Problems in Technical Library and Information Center Management: A State-of-the-Art. American Society for Information Science, 1974, p. 119.

11. Brady, Edward L. "The Role of Information Center in Engineering Information System." *In:* *Proceedings of the National Engineering Conference*. Engineer's Joint Council, 1964, as cited in, Allan Rees, "Functional Integration of Technical Libraries, Information Centers and Information Analysis Center." In: Alan Rees, Contemporary Problems in Technical Library and Information Center Management: A State-of-the-Art. American Society for Information Science, 1974, p. 119.

12. Bishop, Ethelyn and Norman Nisenoff. An Application of Market Research Techniques to the Dissemination of Scientific and Technical Information. NSF/DSI1751-13211 A01, November, 1977.

13. Software Engineering Research Review, Quantitative Software Models, (SRR-1). March, 1979.

14. The DACS Glossary, A Bibliography of Software Engineering Terms. October, 1979.

TABLE I USER SURVEY - PRELIMINARY RESULTS

Profile of user:

User		Percent of Respondees
Researchers:	28%	
Developers:	70%	(Managers 38%, Programmer/analyst 32%
Not designated:	3%	
	100%	

Information value:

Percent of respondees ranking an information type as 1, 2, or 3 on a scale of 1 (most valuable) to 7 (least valuable).

Information Type	Percent
Information on new software research	57%
State-of-the-art reports	75%
Bibliographies	15%
Software experience data	24%
Information on previous software research	35%
Handbook information	37%
Critical reviews	37%

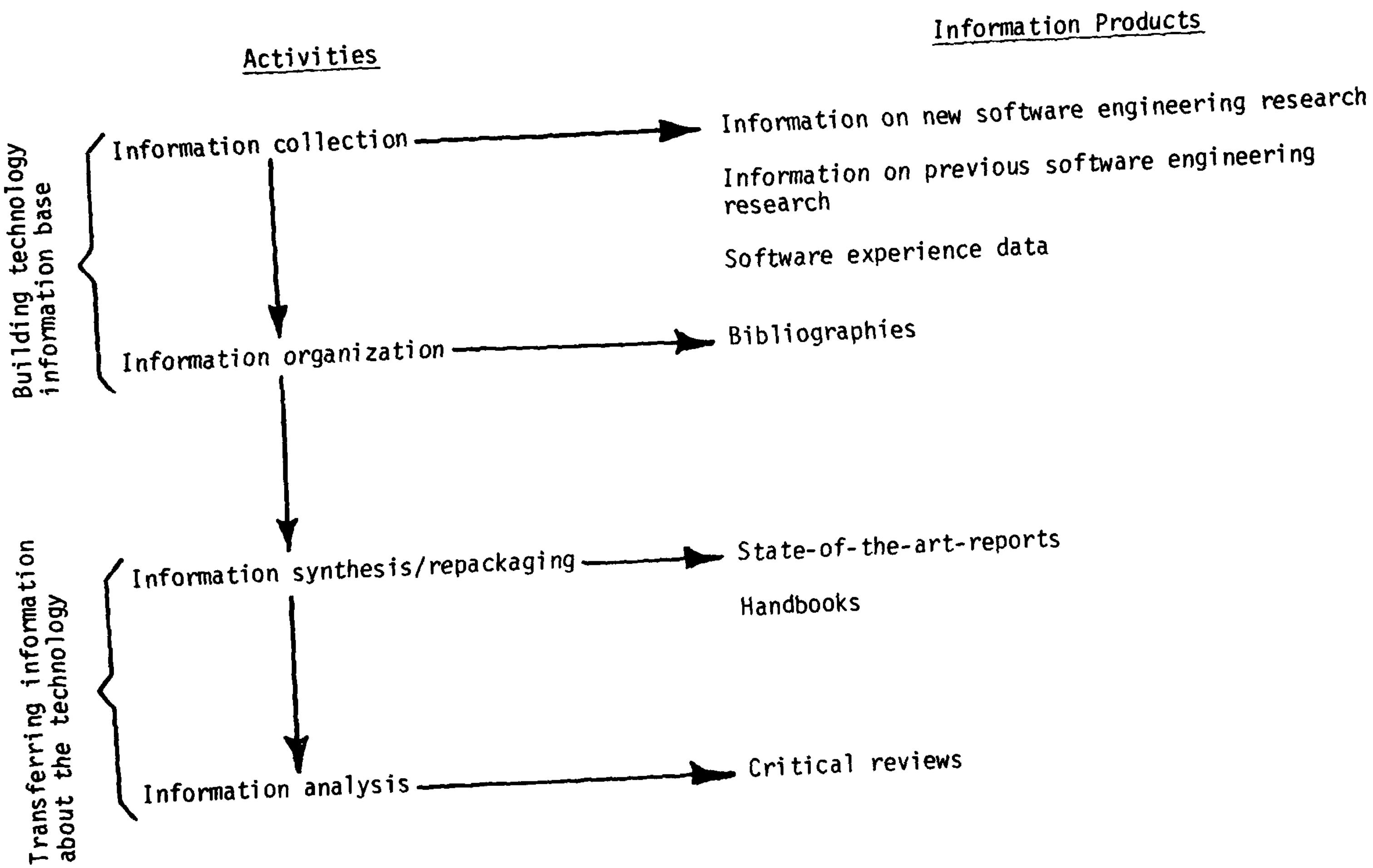


FIGURE 1 DACS ACTIVITIES AND INFORMATION PRODUCTS

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.