

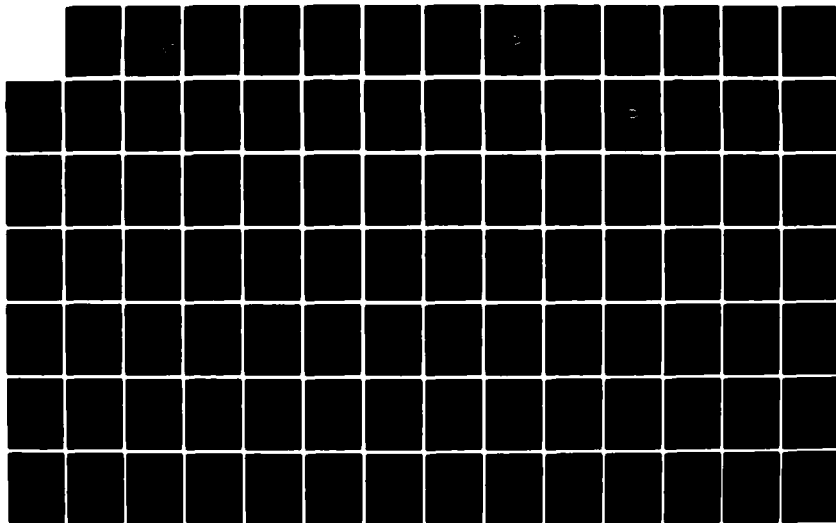
AD-A124 997 ADA* SOFTWARE DESIGN METHODS FORMULATION APPENDICES TO
FINAL REPORT(U) SOFTECH INC WALTHAM MA OCT 82
DAAK80-80-C-0187

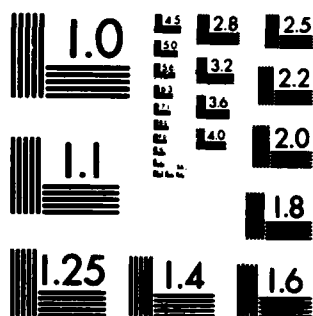
1/2

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

112
ADA* SOFTWARE DESIGN METHODS FORMULATION

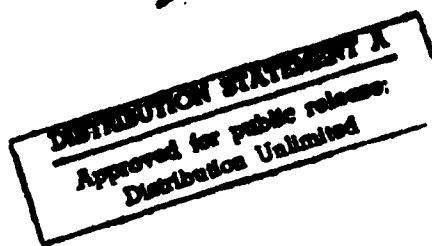
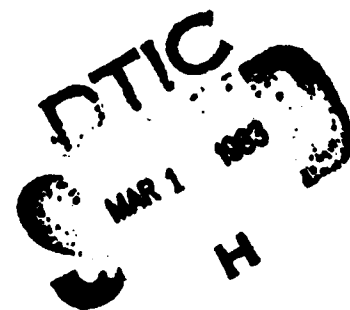
APPENDICES TO FINAL REPORT

OCTOBER 1982

CENTER FOR TACTICAL COMPUTER SYSTEMS
(CENTACS)

U. S. ARMY COMMUNICATIONS - ELECTRONICS COMMAND
(CECOM)

CONTRACT DAAK80-80-C-0187



PREPARED BY
SofTECH, INC.
460 TOTTEN POND ROAD
WALTHAM, MA 02154

83 02 028 050

ADA* IS A TRADEMARK OF THE DEPARTMENT OF DEFENSE (ADA JOINT PROGRAM OFFICE)

DTIC FILE COPY

AD A124997

TABLE OF CONTENTS :

<u>Appendix</u>		<u>Page</u>
A	→ Participating Companies and Government Facilities;	A-1
B	→ Ada Software Design Methods Formulation (Industry/Government Work Force Survey);	B-1
C	→ Ada Software Design Methods Formulation (Industrial Training Survey); and	C-1
D	→ Ada Software Design Methods Formulation (Survey Statistical Consultant).	D-1

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A	



APPENDIX A
PARTICIPATING COMPANIES AND GOVERNMENT FACILITIES

- 1) The following companies and government agencies participated in the Industry/Government Work Force Survey:

ARRADCOM
PDSS Center
Dover, NJ

CECOM
Software Development and Support Center
Ft. Leavenworth, Kansas

CECOM/CENTACS
Software Technology Division
Ft. Monmouth, NJ

Control Data Corp.
40 Avenue at the Common
Shrewsbury, NJ

FATDS
Software Support Group
Ft. Sill, Oklahoma

General Dynamics
Data Systems Division
P. O. Box 748
Fort Worth, Texas

GTE Systems
77 A Street
Needham, MA

IBM
Federal Systems Division
Oswego, New York

MICOM
Software Support Center
Redstone Arsenal, Alabama

SofTech
460 Totten Pond Road
Waltham, MA

Sperry Univac
766 Shrewsbury Ave.
Tinton Falls, NJ

TRW
One Space Park
Redondo Beach, CA

- 2) The following companies participated in the Industrial Training Survey:

Control Data Corp.
40 Avenue at the Common
Shrewsbury, NJ

General Dynamics
Data Systems Division
P.O. Box 748
Fort Worth, Texas

GTE Systems
77 A Street
Needham, MA

IBM
Federal Systems Division
Oswego, New York

Singer Kearfott
1150 McBride Avenue
West Paterson, NJ

Sperry Univac
766 Shrewsbury Ave.
Tinton Falls, NJ

TRW
One Space Park
Redondo Beach, CA

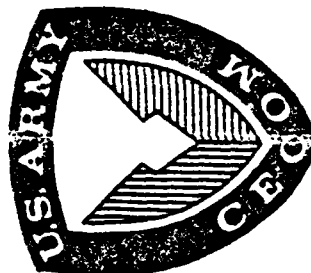
APPENDIX B

ADA SOFTWARE DESIGN METHODS FORMULATION

INDUSTRY/GOVERNMENT WORK FORCE SURVEY

- Letter to Survey Administrator
- Guidelines for survey distribution
- Survey

ADA* SOFTWARE DESIGN METHOD FORMULATION INDUSTRY/GOVERNMENT WORK FORCE SURVEY



March 1, 1982

Dear Survey Administrator:

The Center for Tactical Computer Systems, U.S. Army Communication-Electronics Command (CECOM), Fort Monmouth, NJ, is sponsoring a study to identify effective approaches for the use of Ada in designing and developing embedded systems software. This effort is part of the Army Ada Language Program, the first phase of a major software technology R&D initiative intended to introduce a new and more effective method for software development and maintenance.

As part of this study, SofTech, Inc., has been awarded a contract to conduct a survey of the industrial and government work force to determine generic job categories among embedded systems personnel. The survey results will be used to assist in the formulation of Ada training requirements. Your agreement to participate in this survey is vital to CECOM in establishing Ada training criteria which will accurately reflect the needs of industry and government.

This packet contains a set of questionnaires for distribution to your staff. These are designed to gather information about job categories for embedded systems personnel. We have included some guidelines for survey distribution which we hope will be helpful. The surveys have been structured to require approximately thirty minutes of any respondent's time. Please return the completed surveys to SofTech by March 26, 1982.

Your participation is a key element in assuring the success of this survey and we are most appreciative of your efforts on our behalf. CECOM is committed to transferring the Ada technology in the most efficient and timely way to the Ada user community. The published results of this survey will be sent to you as soon as they come available.

Thank you for your time and effort.

Very truly yours,

Joseph Kernan
Chief, Software Technology
Development Division
CENTACS
CECOM

INDUSTRY/GOVERNMENT WORK FORCE SURVEY

GUIDELINES FOR SURVEY DISTRIBUTION

- This survey is intended for personnel involved with embedded computer system software. The survey should be distributed to as broad a sample of job titles and responsibilities as possible within your organization. It is the intent that the results of this survey are representative of individuals involved with each phase of the software life cycle (i.e., design, coding, maintenance, management).

- Briefly explain to each manager or to the individual respondent:
 - The general purpose of the survey.
 - The expected completion time of thirty minutes.
 - The selected procedure for questionnaire return.

- Questionnaire return alternatives:
 - Individual respondents can mail their completed questionnaires directly to SofTech, Inc. Postpaid Return envelopes suitable for this alternative will be provided by SofTech, Inc.
 - Survey Administrators can collect respondents' completed questionnaires and mail to SofTech, Inc., 460 Totten Pond Road, Waltham, MA 02154.

- If you have any questions regarding this survey effort, please contact:

Karen Sather
Survey Coordinator
SofTech, Inc.
617/890-6900, Ext. 187

Dear Survey Participants:

Ada is a new language for developing and supporting software for large embedded computer systems. The Department of Defense has chosen Ada as the common programming language to be used on future DoD embedded systems projects. As a result of this decision, the Department of Defense is committed to establishing a knowledgeable and well-trained Ada user community to ensure proper application of Ada concepts.

Effective Ada training will be a vital factor in the widespread acceptance and use of the language. The U.S. Army Communications-Electronics Command (CECOM) in conjunction with Softech, Inc., is conducting a survey to identify the functions which are performed by personnel working on the development of large-scale embedded systems. Once functional job categories are established, Ada training can be developed which will address directly the needs of each level. Your input is important in identifying these needs.

Please complete this questionnaire using the following guidelines:

- Answer all questions to the best of your knowledge. If some questions are outside of your specialty area, feel free to skip those questions.
- It should take no more than thirty minutes to complete this questionnaire.
- After you have completed the questionnaire, follow the questionnaire return procedure established in your company.

Thank you for your time and effort on our behalf.

Sincerely,

Joseph Kernan
Chief, Software Technology Development Division
CENTACS
CECOM

Job Title or Specialty _____
GS or Rank (if appropriate) _____
Date _____

ADA SOFTWARE DESIGN METHOD FORMULATION INDUSTRIAL/GOVERNMENT WORK FORCE SURVEY

1. How many years have you been involved with software development and/or support?
a. 0-2 years _____ b. 2-5 years _____ c. 5-10 years _____ d. Over 10 years _____
2. Which of the following areas describe your experience with software development and/or support?
a. Commercial _____ d. Educational _____
b. Military _____ e. Statistical _____
c. Embedded Computer Systems _____ f. Other _____
3. To date, what has been your level of involvement with Ada?
a. Do not know what Ada is. _____
b. Have heard of Ada but am not familiar with Ada concepts. _____
c. Participated in orientation sessions about Ada. _____
d. Have had Ada training. _____
e. Other (explain) _____
4. If you have had Ada training, what type of training was it?
a. Video tape _____ e. Seminar _____
b. College course _____ f. Informal, on the job training _____
c. In-house course _____ g. Other (explain) _____
d. Programmed learning _____

DEVELOPMENT

5. How many years have you worked on the development of large-scale, embedded computer systems?
- a. 0-1 year _____ b. 1-3 years _____ c. 3-5 years _____ d. Over 5 years _____

IF YOUR DUTIES ARE PRINCIPALLY IN THE DEVELOPMENT AREA, PLEASE ANSWER THE FOLLOWING QUESTIONS:

6. What outputs do you produce? (Check as many as appropriate.)
- | | | | |
|---|-------|--|-------|
| a. Hardware/software tradeoff evaluation | _____ | j. Management plans | _____ |
| b. Data flow diagrams | _____ | k. Cost data | _____ |
| c. Test drivers | _____ | l. Analysis reports/summaries | _____ |
| d. Code | _____ | m. Milestone charts/schedules | _____ |
| e. Program design language or flow charts | _____ | n. Status reports | _____ |
| f. Requirements specifications | _____ | o. Interview sheets/Hiring recommendations | _____ |
| g. Design specifications | _____ | p. Correspondence | _____ |
| h. Test plans | _____ | q. Other - (explain) | _____ |
| i. Integration plans | _____ | | |

7. Which of the following describe your principal duties?
- | | | | |
|---|-------|-----------------------------|-------|
| a. Requirements/Analysis Review (____ Conduct, ____ Attend) | _____ | h. Formulation of Strategy | _____ |
| b. System Analysis | _____ | i. Technical Management | _____ |
| c. Design | _____ | j. Program Management | _____ |
| d. Design Review (____ Conduct, ____ Attend) | _____ | k. Configuration Management | _____ |
| e. Code | _____ | l. Quality Assurance | _____ |
| f. Structured Walkthroughs (____ Conduct, ____ Attend) | _____ | m. Monitoring contracts | _____ |
| g. Formulation of Policy | _____ | n. Other (explain) | _____ |

SUPPORT

8. How many years have you worked on the support of large-scale, embedded computer systems?
- a. 0-1 year _____ b. 1-3 years _____ c. 3-5 years _____ d. Over 5 years _____

1094-2

IF YOUR DUTIES ARE PRINCIPALLY IN THE SUPPORT AREA, PLEASE ANSWER THE FOLLOWING QUESTIONS:

9. What outputs do you produce? (Check as many as appropriate.)
- | | |
|--|--|
| _____ a. Software trouble report analyses | _____ j. Updated training manuals |
| _____ b. Temporary (proposed) Engineering Change Proposals | _____ k. Updated user manuals |
| _____ c. Red lined documentation | _____ l. Software Trouble Reports (STRs) |
| _____ d. Test plans | _____ m. Automated build systems |
| _____ e. Test drivers | _____ n. Management information reports |
| _____ f. Technical advice to Configuration Control Board | _____ o. Version description documents |
| _____ g. Updated MIL-STD specification | _____ p. Version audits |
| _____ h. Library Control | _____ q. Field engineering reports |
| _____ i. Maintain configuration procedures | _____ r. Other (explain) _____ |

10. Which of the following describe your principal duties? (Check as many as appropriate.)
- | | |
|--|---|
| _____ a. Analysis | _____ h. Program Management |
| _____ b. Design | _____ i. Software Configuration Control Board participation |
| _____ c. Design Review (____ Conduct, ____ Attend) | _____ j. Configuration management |
| _____ d. Code/Patch | _____ k. Quality Assurance |
| _____ e. Structured Walkthroughs (____ Conduct, ____ Attend) | _____ l. Monitoring contracts |
| _____ f. Technical Management | _____ m. Other (explain) _____ |
| _____ g. Formulation of policy | |

GENERAL

11. RATE THE IMPORTANCE OF THE FOLLOWING ACTIVITIES AS THEY APPLY TO YOUR PRESENT JOB. REVIEW ALL CATEGORIES BUT CONSIDER ONLY THOSE ACTIVITIES WHICH YOU PERFORM AS PART OF YOUR RESPONSIBILITIES.

	PRIMARY	SECONDARY	MARGINAL	N/A
A. MANAGEMENT/ADMINISTRATIVE				
1. Program management				
2. Sales/marketing				
3. Contract negotiation				
4. Formulating policy				
5. Formulating strategy				
6. Preparing budgets/cost estimates				
7. Technical management				
8. Interviewing personnel				
9. Preparing and revising schedules				
10. Preparing management information reports				
11. Preparing field engineering reports				
12. Other administrative tasks				
B. CONFIGURATION/QUALITY CONTROL				
1. Giving technical advice to configuration control board				
2. Maintaining configuration procedures				
3. Library control				
4. Preparing version audits				
5. Quality Assurance				
6. Preparing temporary (proposed) engineering change reports/requests				

	PRIMARY	SECONDARY	MARGINAL	N/A
C. EDUCATION/SELF-DEVELOPMENT				
1. Preparing technical reports or papers	_____	_____	_____	_____
2. Reading technical magazines, papers, etc.	_____	_____	_____	_____
3. Reviewing technical work of others	_____	_____	_____	_____
4. Teaching others (including preparation)	_____	_____	_____	_____
5. Updating training manuals	_____	_____	_____	_____
6. Being trained (including preparation)	_____	_____	_____	_____
D. PROGRAM DESIGN/IMPLEMENTATION				
1. Functional system design (or architecture)	_____	_____	_____	_____
2. Functional module or subsystem design	_____	_____	_____	_____
3. Defining global data structures	_____	_____	_____	_____
4. Defining subsystem (module) interfaces	_____	_____	_____	_____
5. Defining data structures and algorithms for your own use	_____	_____	_____	_____
6. Coding	_____	_____	_____	_____
7. Debugging or modifying code	_____	_____	_____	_____
E. DOCUMENTATION				
1. Originating system requirements documents	_____	_____	_____	_____
2. Updating MIL-STD specifications	_____	_____	_____	_____
3. Preparing red lined documentation	_____	_____	_____	_____
4. Preparing version description documents	_____	_____	_____	_____
5. Originating/updating user manuals	_____	_____	_____	_____
6. Documenting code	_____	_____	_____	_____

	PRIMARY	SECONDARY	MARGINAL	N/A
F. TESTING				
1. Defining system test cases	_____	_____	_____	_____
2. Preparing test drivers	_____	_____	_____	_____
3. Preparing test plans	_____	_____	_____	_____
4. Hardware testing	_____	_____	_____	_____
5. System software testing	_____	_____	_____	_____
6. Defining module test cases	_____	_____	_____	_____
7. Software module testing	_____	_____	_____	_____
8. Documenting test results	_____	_____	_____	_____
9. Preparing software trouble/discrepancy reports	_____	_____	_____	_____
10. Analyzing software trouble reports	_____	_____	_____	_____

12. Do you belong to any technical societies or working groups outside of your company? Yes _____ No _____

13. During the past year, have you attended any job related conferences? Yes _____ No _____

If so, what was your role at the conference?

- a. Organizer _____ c. Attendee _____
 b. Speaker _____ d. Other (explain) _____

14. Have you ever published or presented a paper? Yes _____ No _____

15. Do you read job-related magazines or newsletters?

- a. Regularly _____ c. Only as my job demands _____ e. Other (explain) _____
 b. Occasionally _____ d. Never _____

16. Which of the following programming languages have you studied or used? Please check all that apply.

a. JOVIAL	j. ALGOL	s. SNOBOL
b. CMS-2	k. RATFOR, WATFOR, WATFIV	t. ECL
c. C	l. MODULA	u. GPSS
d. FORTRAN	m. SIMULA	v. SAS
e. COBOL	n. XPL	w. PROTEGE
f. ASSEMBLER	o. MPP	x. PPL
g. PL-1	p. FORTH	y. APL
h. PASCAL	q. Ada	z. Other (specify) _____
i. BASIC	r. LISP	

17. Considering your response to Question 16 list the two languages with which you are most proficient.

a. _____ b. _____

18. INDICATE THE LEVEL OF KNOWLEDGE YOU HAVE WITH THE FOLLOWING METHODOLOGIES.

A. Methodology

	<u>Unfamiliar</u>	<u>Have Heard of But Do Not Understand</u>	<u>Know What Concept Is</u>	<u>Have Used to A Moderate Extent</u>	<u>Have Used Frequently</u>
1. PSL/PLA	_____	_____	_____	_____	_____
2. SADT	_____	_____	_____	_____	_____
3. SREM	_____	_____	_____	_____	_____
4. HIPO	_____	_____	_____	_____	_____
5. Jackson Design	_____	_____	_____	_____	_____
6. Structured Design	_____	_____	_____	_____	_____
7. Warnier/Orr Design	_____	_____	_____	_____	_____
8. N-S/Chapin Chart	_____	_____	_____	_____	_____
9. Beamson Tables	_____	_____	_____	_____	_____
10. Program Design Language	_____	_____	_____	_____	_____

A. Methodology (Cont.)	Unfamiliar	Have Heard Of But Do Not Understand	Know What Concept Is	Have Used to A Moderate Extent	Have Used Frequently
11. Structured Programming					
12. Structured Walkthroughs					
13. Top-Down Design					
14. Top-Down Testing					
15. Bottom-up Design					
16. Bachman Diagramming					
17. Entity Diagrams					
18. Data Abstraction					
19. Other (specify)					
B. Programming Constructs					
1. Enumeration types					
2. Floating point types					
3. Fixed point types					
4. User defined types					
5. Pointers					
6. Typed pointers					
7. Ranges					
8. Records					
9. Variant records					
10. Object/type declarations					
11. Global variables					
12. Local variables					
13. Formal and actual parameters					
14. Reserved words					
15. Blocks					
16. Case statements					
17. If/then/else statements					
18. Loop/for/while/until statements					
19. Exit Statements (for loops)					

<u>B. Programming Constructs (Cont.)</u>	<u>Unfamiliar</u>	<u>Have Heard Of But Do Not Understand</u>	<u>Know What Concept Is</u>	<u>Have Used to A Moderate Extent</u>	<u>Have Used Frequently</u>
20. Procedures	_____	_____	_____	_____	_____
21. Functions	_____	_____	_____	_____	_____
22. Return statements	_____	_____	_____	_____	_____
23. Clusters/modules/packages	_____	_____	_____	_____	_____
24. Stubs	_____	_____	_____	_____	_____
25. Goto statements	_____	_____	_____	_____	_____
26. Comments	_____	_____	_____	_____	_____
27. Exception handlers	_____	_____	_____	_____	_____
28. Task/coroutines	_____	_____	_____	_____	_____
29. Other (specify)	_____	_____	_____	_____	_____

1094-2

<u>C. Programming Concepts</u>	_____	_____	_____	_____	_____
1. Importing/exporting names	_____	_____	_____	_____	_____
2. Data encapsulation (compoals)	_____	_____	_____	_____	_____
3. Name scoping	_____	_____	_____	_____	_____
4. Name visibility	_____	_____	_____	_____	_____
5. Static and dynamic nesting	_____	_____	_____	_____	_____
6. Iteration	_____	_____	_____	_____	_____
7. Conditional statements	_____	_____	_____	_____	_____
8. Recursion	_____	_____	_____	_____	_____
9. Concurrency	_____	_____	_____	_____	_____
10. Strong typing	_____	_____	_____	_____	_____
11. Type conversion	_____	_____	_____	_____	_____
12. Data abstraction	_____	_____	_____	_____	_____
13. Generics	_____	_____	_____	_____	_____
14. Loop invariants	_____	_____	_____	_____	_____
15. Parameter binding	_____	_____	_____	_____	_____
16. Version number	_____	_____	_____	_____	_____
17. Other (specify)	_____	_____	_____	_____	_____

B-13

10. Ada Programming Concepts	Unfamiliar	Have Heard of But Do Not Understand	Know What Concept Is	Have Used to A Moderate Extent	Have Used Frequently
1. Enumeration types					
2. User-defined types					
3. Subtypes					
4. Derived types					
5. Real types					
6. Floating point types					
7. Fixed point types					
8. Record types					
9. Record types with discriminants					
10. Slices					
11. Aggregates					
12. Allocators					
13. Access types					
14. Overloading					
15. Packages					
16. Private types					
17. Scope					
18. Short circuiting					
19. Visibility					
20. Tasking					
21. Task types					
22. Rendezvous					
23. Entries					
24. Entry Families					
25. Separate compilation					
26. Exceptions					
27. Generic program units					
28. Instantiation					

D. <u>Ada Programming Concepts (Cont.)</u>			<u>Unfamiliar</u>	<u>Have Heard Of But Do Not Understand</u>	<u>Know What Concept Is</u>	<u>Have Used to A Moderate Extent</u>	<u>Have Used Frequently</u>
29.	Elaboration						
30.	Context specification						
31.	Information hiding						
32.	Mutual recursion						
33.	Other (specify)						

19. What do you like most about Ada? _____

20. What do you like least about Ada? _____

21. Thank you for completing this survey. If there are additional comments you wish to make, please feel free to make them below.

APPENDIX C

ADA SOFTWARE DESIGN METHODS FORMULATION

INDUSTRIAL TRAINING SURVEY

- Letter to Training Survey Respondent
- Survey

ADA* SOFTWARE DESIGN METHOD FORMULATION

INDUSTRIAL TRAINING SURVEY



March 5, 1982

Dear Training Survey Respondent:

The Center for Tactical Computer Systems, U.S. Army Communication-Electronics Command (CECOM), Fort Monmouth, NJ, is sponsoring a study to identify effective approaches for the use of Ada in designing and developing embedded systems software. This effort is part of the Army Ada Language Program, the first phase of a major software technology R&D initiative intended to introduce a new and more effective method for software development and maintenance.

As part of this study, SofTech, Inc., has been awarded a contract to conduct a survey which will reflect training methods, training programs and software development policies of company divisions involved with embedded computer system software. The person(s) completing this survey should have management level training responsibilities. Your agreement to participate in this survey is important to CECOM in establishing Ada training criteria which will accurately reflect the needs of industry and government.

Completed surveys should be returned by March 26, 1982 to

SofTech, Inc.
460 Totten Pond Road
Waltham, MA 02154

If there are any questions regarding this effort please contact Karen Sather at SofTech (617-890-6900, Ext. 137).

Your participation is a key element in assuring the success of this survey and we are most appreciative of your efforts on our behalf. CECOM is committed to transferring the Ada technology in the most efficient and timely way to the Ada user community. The published results of this survey will be sent to you as soon as they are available.

Thank you for your time and effort.

Very truly yours,

Joseph Kernan
Chief, Software Technology
Development Division
CENTACS
CECOM

Department

INDUSTRIAL TRAINING SURVEY

PART I: TRAINING

A. GENERAL

- Other (explain)**

- d. Other (explain)**

5. In terms of training functions does your training department have individuals who
- ___ a. Design training courses only
 - ___ b. Develop training materials only
 - ___ c. Provide instruction only
 - ___ d. Design and develop training courses and support course materials
 - ___ e. Perform all functions
 - ___ f. Other (explain) _____

1094-2

IN GENERAL, WHAT ARE THE MINIMUM QUALIFICATIONS OF YOUR COURSE DEVELOPERS?

6. Educational Background: ___ a. BS ___ b. MS ___ c. PhD ___ d. Other
7. Technical Experience: ___ a. 0-3 yrs. ___ b. 3-5 yrs. ___ c. Over 5 ___ d. Other
8. Teaching Experience: ___ a. 0-3 yrs. ___ b. 3-5 yrs. ___ c. Over 5 ___ d. Other
9. Do you consider any other qualifications when selecting course developers? Yes ___ No ___
Explain _____

C-4

IN GENERAL, WHAT ARE THE MINIMUM QUALIFICATIONS OF YOUR INSTRUCTORS?

10. Educational Background: ___ a. BS ___ b. MS ___ c. PhD ___ d. Other
11. Technical Experience: ___ a. 0-3 yrs. ___ b. 3-5 yrs. ___ c. Over 5 ___ d. Other

12. Teaching Experience: a. 0-3 yrs. b. 3-5 yrs. c. Over 5 d. Other

13. Do you consider any other qualifications when selecting course instructors? Yes No
Explain

14. How do you train your educational staff?

- ___ a. Internally through educational department
- ___ b. Internally through technical staff
- ___ c. Contracted training
- ___ d. Other (explain)

15. Does your company offer in-house courses on a regular basis? Yes No

16. Approximately, how many courses do you offer per year? _____

17. In general, are the instructors for these courses

- ___ a. Full-time training staff
- ___ b. Technical Staff
- ___ c. Marketing support technical staff
- ___ d. Other (explain)

18. What is the normal duration of your in-house courses?

- ___ a. 1-2 days b. 3-5 days c. 6-10 days d. over 10 days

19. What is the average class size for an in-house course?
a. Less than 10 b. 10-15 c. 16-30 d. More than 30

20. When are your in-house courses offered?
a. During working hours b. After working hours c. Both

21. What internal facilities are available for company sponsored training?

- | | | |
|---|-----------|-------|
| a. Classrooms (0-15 people) | How many? | _____ |
| b. Classrooms (16-30 people) | How many? | _____ |
| c. Classrooms (over 30 people) | How many? | _____ |
| d. Auditoriums (over 75 people) | | _____ |
| e. Laboratories (online access to system) | | _____ |
| f. Video cassette recorders and monitors | | _____ |
| g. 16mm or 35mm projection facilities | | _____ |
| h. Overhead Projectors | | _____ |
| i. Other (please specify) | | _____ |

22. Do you contract with outside sources for facilities when necessary? Yes _____ No _____

23. What instructional formats has your company used? (Check as many as appropriate.)

- | | | | |
|-------------------------------|-------|---------------------------|-------|
| a. Lecture | _____ | f. Self-paced instruction | _____ |
| b. Workshop | _____ | g. Videotapes | _____ |
| c. Lecture/workshop | _____ | h. Film | _____ |
| d. Computer aided instruction | _____ | i. Other (Explain) | _____ |
| e. On-line exposure | _____ | | _____ |

24. Which of these formats have you found most effective?

a. _____ b. _____

25. Which of these formats have you found least effective?

a. _____ b. _____

26. How are your in-house courses evaluated? (Check all appropriate responses.)

- _____ a. Informal feedback
- _____ b. Written evaluation from students
- _____ c. Professional review
- _____ d. Not evaluated
- _____ e. Other (explain) _____

27. Does your company ever contract with outside vendors for technical training? Yes _____ No _____

28. What form(s) of training do you purchase?

- _____ a. Courses
 - _____ 0-3 days
 - _____ 3-5 days
 - _____ 5-10 days
 - _____ Over 10 days
- _____ b. Seminars/symposia
- _____ c. Videotapes
- _____ d. Workshops
- _____ e. Tutorial texts
- _____ f. Other _____

B. EMBEDDED SYSTEMS TRAINING

29. Does your company provide formal in-house training to personnel involved with embedded computer system software? Yes ☐ No ☐

30. Who is responsible for providing in-house training to embedded computer systems personnel?

- a. The department requiring training
- b. An educational organization within your division
- c. A corporate educational organization
- d. Other (explain)

31. What percent of the training of embedded computer systems personnel is provided by each of the following:

	<u>0-20%</u>	<u>21-40%</u>	<u>41-60%</u>	<u>61-80%</u>	<u>81-100%</u>
a. In-house courses					

- | | 1962 | 1963 | 1964 | 1965 | 1966 | 1967 | 1968 | 1969 | 1970 | 1971 | 1972 | 1973 | 1974 | 1975 | 1976 | 1977 | 1978 | 1979 | 1980 | 1981 | 1982 | 1983 | 1984 | 1985 | 1986 | 1987 | 1988 | 1989 | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 | 2026 | 2027 | 2028 | 2029 | 2030 | 2031 | 2032 | 2033 | 2034 | 2035 | 2036 | 2037 | 2038 | 2039 | 2040 | 2041 | 2042 | 2043 | 2044 | 2045 | 2046 | 2047 | 2048 | 2049 | 2050 | 2051 | 2052 | 2053 | 2054 | 2055 | 2056 | 2057 | 2058 | 2059 | 2060 | 2061 | 2062 | 2063 | 2064 | 2065 | 2066 | 2067 | 2068 | 2069 | 2070 | 2071 | 2072 | 2073 | 2074 | 2075 | 2076 | 2077 | 2078 | 2079 | 2080 | 2081 | 2082 | 2083 | 2084 | 2085 | 2086 | 2087 | 2088 | 2089 | 2090 | 2091 | 2092 | 2093 | 2094 | 2095 | 2096 | 2097 | 2098 | 2099 | 2100 | 2101 | 2102 | 2103 | 2104 | 2105 | 2106 | 2107 | 2108 | 2109 | 2110 | 2111 | 2112 | 2113 | 2114 | 2115 | 2116 | 2117 | 2118 | 2119 | 2120 | 2121 | 2122 | 2123 | 2124 | 2125 | 2126 | 2127 | 2128 | 2129 | 2130 | 2131 | 2132 | 2133 | 2134 | 2135 | 2136 | 2137 | 2138 | 2139 | 2140 | 2141 | 2142 | 2143 | 2144 | 2145 | 2146 | 2147 | 2148 | 2149 | 2150 | 2151 | 2152 | 2153 | 2154 | 2155 | 2156 | 2157 | 2158 | 2159 | 2160 | 2161 | 2162 | 2163 | 2164 | 2165 | 2166 | 2167 | 2168 | 2169 | 2170 | 2171 | 2172 | 2173 | 2174 | 2175 | 2176 | 2177 | 2178 | 2179 | 2180 | 2181 | 2182 | 2183 | 2184 | 2185 | 2186 | 2187 | 2188 | 2189 | 2190 | 2191 | 2192 | 2193 | 2194 | 2195 | 2196 | 2197 | 2198 | 2199 | 2200 | 2201 | 2202 | 2203 | 2204 | 2205 | 2206 | 2207 | 2208 | 2209 | 2210 | 2211 | 2212 | 2213 | 2214 | 2215 | 2216 | 2217 | 2218 | 2219 | 2220 | 2221 | 2222 | 2223 | 2224 | 2225 | 2226 | 2227 | 2228 | 2229 | 2230 | 2231 | 2232 | 2233 | 2234 | 2235 | 2236 | 2237 | 2238 | 2239 | 2240 | 2241 | 2242 | 2243 | 2244 | 2245 | 2246 | 2247 | 2248 | 2249 | 2250 | 2251 | 2252 | 2253 | 2254 | 2255 | 2256 | 2257 | 2258 | 2259 | 2260 | 2261 | 2262 | 2263 | 2264 | 2265 | 2266 | 2267 | 2268 | 2269 | 2270 | 2271 | 2272 | 2273 | 2274 | 2275 | 2276 | 2277 | 2278 | 2279 | 2280 | 2281 | 2282 | 2283 | 2284 | 2285 | 2286 | 2287 | 2288 | 2289 | 2290 | 2291 | 2292 | 2293 | 2294 | 2295 | 2296 | 2297 | 2298 | 2299 | 2300 | 2301 | 2302 | 2303 | 2304 | 2305 | 2306 | 2307 | 2308 | 2309 | 2310 | 2311 | 2312 | 2313 | 2314 | 2315 | 2316 | 2317 | 2318 | 2319 | 2320 | 2321 | 2322 | 2323 | 2324 | 2325 | 2326 | 2327 | 2328 | 2329 | 2330 | 2331 | 2332 | 2333 | 2334 | 2335 | 2336 | 2337 | 2338 | 2339 | 2340 | 2341 | 2342 | 2343 | 2344 | 2345 | 2346 | 2347 | 2348 | 2349 | 2350 | 2351 | 2352 | 2353 | 2354 | 2355 | 2356 | 2357 | 2358 | 2359 | 2360 | 2361 | 2362 | 2363 | 2364 | 2365 | 2366 | 2367 | 2368 | 2369 |
|--|------|
|--|------|

32. What instructional formats has your company used with embedded systems personnel? (Check as many as appropriate.)

- | | | | |
|----|----------------------------|----|------------------------|
| a. | Lecture | f. | Self-paced instruction |
| b. | Workshop | g. | Videotape |
| c. | Lecture/workshop | h. | Film |
| d. | Computer aided instruction | i. | Other (explain) |
| e. | Online exposure | | |

33. Which formats have you found most effective in the training of programmers in new languages?
a. _____
b. _____
34. Which formats have you found least effective in the training of programmers in new languages?
a. _____
b. _____
35. Which formats have you found most effective in the training of embedded systems personnel in program management, system analysis and design, and system architecture?
a. _____
b. _____
36. Which formats have you found least effective in the training of embedded systems personnel in program management, system analysis and design, and system architecture?
a. _____
b. _____

PART II: SOFTWARE DEVELOPMENT POLICIES

IF YOUR COMPANY HAS DOCUMENTED POLICIES AND PROCEDURES FOR SOFTWARE DEVELOPMENT, ANSWER THE FOLLOWING QUESTIONS:

37. How were these policies and procedures established?
a. Internal committee or study group _____
b. Internal consultant(s) _____
c. Outside consultant(s) _____
d. Other _____
38. How were these policies and procedures implemented? (Check as many as appropriate.)
a. Pilot project _____
b. Internally developed courses _____
c. Contracted training _____
d. Project leader/supervisor _____
e. Printed materials _____
f. Other (explain) _____

39. How are new staff introduced to your software development policies and procedures?

- a. Internally developed courses
- b. Contracted training
- c. Project leader/supervisor
- d. Printed materials
- e. Other (explain)

1094-2

40. Rank the following methods of introducing software development policies as to effectiveness.
(1 = most effective)

- a. Internally developed courses
- b. Contracted training
- c. Project leader/supervisor
- d. Printed materials
- e. Other (explain)

41. How do you audit your software development policies and procedures?

- a. Internal Quality Assurance
- b. Project leader/supervisor evaluation
- c. Design reviews
- d. Walkthroughs
- e. Not audited
- f. Other (explain)

C-10

42. If policies and procedures are updated, how are these changes communicated to the staff?

- a. Internally developed courses
- b. Contracted training
- c. Project leader/supervisor
- d. Printed materials
- e. Other (explain)

PART VI: Ada TRAINING

IF ANY OF YOUR STAFF HAVE PARTICIPATED IN Ada TRAINING, ANSWER THE FOLLOWING QUESTIONS:

43. Approximately, what percentage of your staff has participated in Ada training?
___ a. 0-25% ___ b. 26-50% ___ c. 51-75% ___ d. 76-100%

44. What form(s) of Ada training has your staff received?

- ___ a. Self taught
- ___ b. Ada seminars
- ___ c. University sponsored course
- ___ d. Government sponsored course
- ___ e. Videotapes
- ___ f. Film
- ___ g. In-house course
 - ___ 1-3 days
 - ___ 3-5 days
 - ___ More than 5 days
- ___ h. Other _____

45. In contrast to other courses, do you believe Ada courses will require:

- ___ a. Less time to develop
- ___ b. Approximately the same time to develop
- ___ c. More time to develop
- ___ d. Less actual class time
- ___ e. Approximately the same class time
- ___ f. More actual class time

46. What do you like most about Ada?

47. What do you like least about Ada?

48. Thank you for completing this Survey. If there are additional comments you wish to make, please feel free to make them below.

APPENDIX D
ADA SOFTWARE DESIGN METHODS FORMULATION

Report of the Survey
Statistical Consultant

Report on the Ada Software Design Methods Formulation

Industry/Government Work Force Survey

Robert J. Muller
M.I.T.
June 2, 1982

Table of Contents

Executive Summary	1
The Survey and the Population	2
The sample	2
The database	3
Single-Variable Descriptions	4
Experience	4
Ada involvement	6
Professionalism	6
Job outputs	7
Principal duties	10
Areas of experience	12
Ada training	13
Two-Variable Comparisons	14
General activities	14
Methodologies	17
Programming languages	22
Relationships	25
Classification of the Workforce	30
Methods	30
The results	36
Characterization of clusters	49
Current titles	64
Further classification of nonmanagement	72
Conclusions	90
Appendix 1--The Database	92
Appendix 2--Bibliography	96
Appendix 3--Workforce Cluster Hierarchy	97

Executive Summary

The Ada Workforce Survey was administered as a nonrandom sample to nine institutions. Because of the method of administration, no statistical generalizations may be made to a general population; the results expressed herein apply only to the sample.

Various single variable descriptions indicate that the sample is largely composed of individuals who have been in the workforce for more than ten years, who generally haven't been trained in Ada, who have worked more on development than on support, who are more-or-less professional (as defined by belonging to technical societies, attending conferences and so on), who in general spend their time programming or designing programs, who have worked on military or embedded systems (90 percent of the workforce), and who generally program in FORTRAN or Assembler (or both).

There are few significant relationships between variables in the survey. Involvement with Ada to a small extent determines knowledge of Ada.

Classification of the workforce based on clustering and smoothing techniques produces five fundamental categories of worker: administrative manager, technical manager, support manager, development nonmanager, and support nonmanager. Comparison of the various variables with these categories supports the interpretation of the categories.

The Survey and the Population

This report is a descriptive analysis of the Ada Software Design Method Formulation Industry/Government Work Force Survey, done by Softech Inc. under the auspices of the U. S. Army Communications-Electronics Command. In order to place this report in its proper perspective, the survey procedures are briefly outlined below.

The sample.--Softech administered the survey as a mailed, key-person survey. That is, Softech mailed copies of the survey to certain administrators in each organization and asked that person to distribute the survey within the organization to people broadly representative of the company. Softech exercised no control over the internal distribution of questionnaires, relying totally on the administrator. There was no reliable attempt at random distribution. There was no attempt to weight the individual questionnaires received with respect to any sampled population.

The implications of this procedure for the following analysis relate mainly to the applicability of the analysis to a general population. A sample is a subset of some population, that population being defined without reference to the sampling method. In the case of a random sample, the members of the population to be sampled are selected by first knowing some characteristic by which all of the population might be indexed (address, phone, name, whatever). The sample is then selected by some criterion designed to eliminate all possible influences induced by considering a limited number of the population.

Softech has apparently defined the population as the subset of the organization that some administrator in the organization feels can complete the survey reasonably and have some "characteristic" qualities making them broadly representative of the subset. Thus the actual population being sampled is de-

defined differently between organizations and possibly within organizations. There is no possibility of modeling the sampling process, since no records were kept of the individual assignment decisions.

The result of these procedures is to have an unspecified population. The consequence is that, although this report states certain results, there is little statistical assurance that the results may be reliably extended to any particular population. The statistics are valid for the sample only.

The database.--The sample consists of 428 surveys of 9 organizations, five in the private sector and four in the public sector. The data is organized into a relational database consisting of nine types of information (or datasets); the number in parentheses after the dataset name is the number of entities in that dataset.

- (1) job history (426)
 - title
 - rank
 - date
 - years of involvement in software
 - level of involvement with Ada
 - years worked on development
 - years worked on support
 - membership in technical societies
 - conference attendance
 - published or presented papers
 - technical reading
- (2) job output (3,541)
- (3) principal duties (2,542)
- (4) general activities (9,841)
 - activity
 - importance
- (5) conference role (106)
- (6) programming languages (2,705)
 - language
 - proficiency
- (7) methodologies (24,302)
 - methodology

extent of knowledge

(8) areas of experience (772)

(9) Ada training (189)

This structure reflects the fact that all of the attributes except for the job history attributes were multiple response questions and hence were at a different level from the survey respondents--that is, a given respondent may have more than one response to a particular question. Appendix 1 contains a complete listing of the datasets, attributes, and possible values for the attributes.

The structural complexity of the database introduces some problems with the descriptive statistics in the following sections. Because of the multiple-response structure of many of the questions, the counts of responses have two interpretations, one relative to the total number of responses ("response level") and the other relative to the total number of respondents ("respondent level"). Percentages in the following tables labeled "Percent of Responses" are calculated as Count divided by the sum of Count over all responses. Percentages labeled "Percent of Respondents" are Count divided by 428 (the number of respondents in the survey). The first percentage is not very interpretable; the second describes the status of a response relative to the respondents, but will not add up to 100, since a given individual may respond with more than one response.

Single-Variable Descriptions

Experience.--The first question to be considered is how many years the respondent has spent in software development and/or support. The following table gives the breakdown of the four possible responses:

	Counts	Percent
less than two years	41	9.7
two to five years	101	23.9
five to ten years	85	20.1
over ten years	196	46.3

This table indicates that almost half of the people who filled out questionnaires had been in the area for over ten years; this will probably have some effect on knowledge and job patterns.

Considered together are the two questions on the length of time worked on development and support. The following two tables break down the questions:

Years Worked on Development

	Counts	Percent
less than a year	78	20.0
one to three years	78	20.0
three to five years	65	16.7
over five years	169	43.3

Years Worked on Support

	Counts	Percent
less than a year	95	33.1
one to three years	73	25.4
three to five years	42	14.6
over five years	77	26.8

The import of these frequencies is not terribly clear without taking into account the relationships between these two variables. There are four possibilities--the respondent worked in both support and development, the respondent worked only in support, the respondent worked only in development, or the respondent worked in neither (or didn't answer the question). Applying these logical combinations, 274 people worked in both, 116 worked in development but not support, 13 people worked in support but not development, and 23 people apparently worked in neither. This result tends to swing the bias to-

ward the development people.

The assumption is made that a nonanswer implies not working in that area, an assumption that might not be justified. It is possible for people who have done no work in the area to respond in the first category, 0-1 year.

Getting back to the original issue, the separate tables indicate that the people who worked on development tend to have done so over three years (the median is three to five). But people who work on support have done so for more than one year (median one to three years). Putting this together with the distribution of workers, it would seem that the industry stresses development and has begun employing support personnel later than development personnel. Another interpretation might be that support personnel have a higher turnover rate than technical personnel. Without company employment studies, neither interpretation can be falsified.

Ada involvement.--The next question is about the level of involvement of the person with Ada. The following table breaks down the five responses:

	Counts	Percent
Ada not known	16	3.8
heard of Ada	225	52.9
had orientation to Ada	101	23.8
Ada training	48	11.3
other	35	8.2

This table indicates that only a third or so of the workforce has any real knowledge of Ada aside from having heard about it. Only 11 percent of the workforce has had Ada training. An implication of this is that Ada training will have to affect a large portion of the industry; another implication is that, if Ada training significantly changes the way people do their jobs, the industry will have to change a great deal as use of Ada becomes widespread.

Professionalism.--With respect to technical societies, a third of the

respondents (142) belong to technical societies and two-thirds of them (276) do not. With respect to conference attendance, 173 (41 percent) have attended a conference and 248 (59 percent) have not. Of those who had, 4 had organized a conference, 10 had been speakers, 91 had been attendees, and 1 had taken some other role. With respect to publishing or presenting papers, 95 (23 percent) have done so, 327 (78 percent) have not. With respect to technical reading, the following table breaks down the possible responses:

	Counts	Percent
regularly	185	44.0
occasionally	187	44.5
only as my job demands	32	7.6
never	14	3.3
other	2	0.5

This table indicates that more than three quarters of all respondents do some reading of job-related magazines or newsletters. What this means is unclear; such newsletters might consist of company rags or might be ACM transactions. There is no information about what type of technical reading is involved.

These last questions seem to be aimed at identifying certain "professional" aspects of the respondent's job. Given the relative vagueness of the issues addressed, the questions don't usefully perform this task. Professionalism has deeper roots than belonging to societies or going to conferences, or even presenting papers. It matters very much which societies, which conferences, and where the papers are presented or published, among other things.

Job outputs.--The following table gives the breakdown of the various job output products, giving the percent of the responses made and the percent of people giving the response relative to the total sample (428).

Output	Count	Percent of Responses	Percent of Respondents
hard soft tradeoff eval	125	3.530	29.206
data flow diagrams	197	5.563	46.028
test drivers	150	4.236	35.047
code	287	8.105	67.056
prog design language	262	7.399	61.215
requirements specs	223	6.298	52.103
design specs	255	7.201	59.579
test plans	236	6.665	55.140
integration plans	165	4.660	38.551
management plans	99	2.796	23.131
cost data	104	2.937	24.299
analysis reports	111	3.135	25.935
milestone charts	173	4.886	40.421
status reports	223	6.298	52.103
interview sheets	116	3.276	27.103
correspondence	92	2.598	21.495
development other	14	0.395	3.271
STR analyses	97	2.739	22.664
temp ECPs	46	1.299	10.748
redlined documentation	52	1.469	12.150
support test plans	70	1.977	16.355
support test drivers	40	1.130	9.346
tech advice to CCB	40	1.130	9.346
updated MIL STD spec	20	0.565	4.673
library control	25	0.706	5.841
maintain config procs	32	0.904	7.477
updated training manuals	24	0.678	5.607
updated user manuals	71	2.005	16.589
STRs	85	2.400	19.860
automated build systems	20	0.565	4.673
management info reports	32	0.904	7.477
version descrip docs	20	0.565	4.673
version audits	11	0.311	2.570
field engineering report	8	0.226	1.869
support other	16	0.452	3.738

The following list of the values presents the values in rank order.

Output	Percent of Respondents
code	67.056
prog_design_language	61.215
design_specs	59.579
test_plans	55.140
status_reports	52.103
requirements_specs	52.103
data_flow_diagrams	46.028
milestone_charts	40.421
integration_plans	38.551

test_drivers	35.047
hard_soft_tradeoff_eval	29.206
interview_sheets	27.103
analysis_reports	25.935
cost_data	24.299
management_plans	23.131
STR_analyses	22.664
correspondence	21.495
STRs	19.860
updated_user_manuals	16.589
support_test_plans	16.355
redlined_documentation	12.150
temp_ECPs	10.748
support_test_drivers	9.346
tech_advice_to_CCB	9.346
management_info_reports	7.477
maintain_config_procs	7.477
library_control	5.841
updated_training_manuals	5.607
automated_build_systems	4.673
updated_MIL_STD_spec	4.673
version_descrip_docs	4.673
support_other	3.738
development_other	3.271
version_audits	2.570
field_engineering_report	1.869

This table describes the number of people who do a certain thing, without taking into consideration what else they do. Thus, 67 percent of the respondents produce code, 61 percent produce program design language, 60 percent produce design specifications, and so on. There does seem to be a general tendency toward those categories directly related to programming in a technical sense; as well, people tend to produce development outputs more often than support outputs. This paints the picture of an industry more oriented toward producing software than toward supporting software.

The section on classification below will discuss the relationships between respondents based on combinations of responses--patterns of job responsibilities and behaviors. These patterns will be the basis for the job classification scheme.

As a comment, this question (and the next question on principal duties)

might have been structured as the later questions were, including some weighting variable such as how much time is spent on a product or on a duty. This might give a clearer picture of job structure. On the other hand, exploration of the variation indicates that most of the reasonable conclusions are robust--not subject to relative extremes in the data. Having information weighting the responses would probably not affect the results very much.

Principal duties.--The following table tabulates the responses to the question about what the respondent's principal duties were:

Duty	Count	Percent of Responses	Percent of Respondents
conduct req review	94	3.698	21.963
attend req review	147	5.783	34.346
system analysis	193	7.592	45.093
design	271	10.661	63.318
conduct design review	151	5.940	35.280
attend design review	196	7.710	45.794
code	235	9.245	54.907
conduct walkthroughs	108	4.249	25.234
attend walkthroughs	128	5.035	29.907
formulation of policy	46	1.810	10.748
formulation of strategy	60	2.360	14.019
technical management	118	4.642	27.570
program management	31	1.220	7.243
configuration management	31	1.220	7.243
quality assurance	31	1.220	7.243
monitoring contracts	23	0.905	5.374
other development	10	0.393	2.336
support analysis	101	3.973	23.598
support design	86	3.383	20.093
conduct support dr	46	1.810	10.748
attend support dr	74	2.911	17.290
code patch	88	3.462	20.561
conduct sup walkthrough	35	1.377	8.178
attend sup walkthrough	52	2.046	12.150
sup technical management	44	1.731	10.280
sup policy formulation	17	0.669	3.972
sup program management	13	0.511	3.037
SCCB participation	36	1.416	8.411
sup config management	20	0.787	4.673
sup quality assurance	25	0.983	5.841
sup monitoring contracts	19	0.747	4.439
other support	13	0.511	3.037

The following list presents the above values in rank order.

Duty	Percent of Respondents
design	63.318
code	54.907
attend_design_review	45.794
system_analysis	45.093
conduct_design_review	35.280
attend_req_review	34.346
attend_walkthroughs	29.907
technical_management	27.570
conduct_walkthroughs	25.234
support_analysis	23.598
conduct_req_review	21.963
code_patch	20.561
support_design	20.093
attend_support_dr	17.290
formulation_of_strategy	14.019
attend_sup_walkthrough	12.150
formulation_of_policy	10.748
conduct_support_dr	10.748
sup_technical_management	10.280
SCCB_participation	8.411
conduct_sup_walkthrough	8.178
program_management	7.243
configuration_management	7.243
quality_assurance	7.243
sup_quality_assurance	5.841
monitoring_contracts	5.374
sup_config_management	4.673
sup_monitoring_contracts	4.439
sup_policy_formulation	3.972
sup_program_management	3.037
other_support	3.037
other_development	2.336

This table demonstrates the same general tendencies as the table of job outputs--the largest percentages are the technical programming duties, and there is a general tendency toward development and away from support. As well, both tables show a relatively low set of percentages for various administrative tasks such as correspondence or formulation of policy. This may indicate that these things aren't important; more likely, it indicates that the people who do them are a small administrative cadre, probably higher up in the

organization than the bulk of the respondents. There is no way to verify this conclusion from the data. Some support for this interpretation comes from the clustering scheme presented in the classification section below.

Areas of experience.--The question on areas of experience breaks down according to the following table:

Area	Count	Percent of Responses	Percent of Respondents
commercial	148	19.171	34.579
military	343	44.430	80.140
embedded computer sys	173	22.409	40.421
educational	51	6.606	11.916
statistical	25	3.238	5.841
other	32	4.145	7.477

Not surprisingly for firms doing defense work, most people have done work in the military arena. Given that Ada applies to embedded computer systems, the relatively low percentage of people (173, or 40 percent of the total) that have worked on embedded systems might mean that any classification scheme developed from this sample will not necessarily be appropriate for organizations doing embedded system work.

However, there is some question of spurious semantic distinction. It would appear that the term "embedded computer system" is relatively new and not well known in the industry as yet. 133 respondents (31 percent) responded with both military and embedded. 40 (9.3 percent) responded embedded alone; 210 (43.5 percent) responded military alone. There is some indication that those responding embedded also specify military more often than those who responding military also specify embedded (77 versus 39 percent, respectively). This might indicate hierarchical knowledge or hierarchical work. Hierarchical knowledge means that knowledge of what an embedded system is implies knowing that it is mainly military, hence explaining the large overlap in

responses; and not knowing implies thinking that the job one does is military, not embedded, even though it might be so considered by somebody who knew what an embedded system is, thus explaining the large number of military-only responses. Hierarchical work, on the other hand, means that people who work on embedded systems generally are in the military field as well, but those in the military field are not necessarily or frequently employed in embedded system work. The survey data has no information sufficient to distinguish these explanations.

Ada training.--Looking at the last question to be considered in this section, those people who have had some training in Ada have trained in the following ways:

Training Method	Count	Percent of Respondents		
		Percent of Responses	Percent of Those Trained	
videotape	16	8.466	3.738	12.030
college_course	7	3.704	1.636	5.263
in_house_course	46	24.339	10.748	34.586
programmed_learning	5	2.646	1.168	3.759
seminar	43	22.751	10.047	32.331
informal_training	51	26.984	11.916	38.346
other	21	11.111	4.907	15.789

Three quarters of the methods people used are in-house courses, seminars, or informal training. This breakdown might be explained by the other methods not being widely available yet. As well, 40 percent of the respondents have had training of any sort; compare this with the frequencies reported above for level of involvement with Ada: 48 people (11 percent) report training there. The following table breaks down the type of training by Ada involvement in an attempt to resolve this inconsistency:

Ada training	had orientation to Ada				
	heard of Ada		other		
	Ada not known	Ada training			
videotape	0	0	3	9	4
college course	0	0	0	6	1
in house course	0	1	15	26	4
programmed learning	0	0	2	3	0
seminar	0	0	28	12	3
informal training	0	3	19	15	14
other	0	3	7	6	5

Apparently, what happened is that people who participated in orientation sessions on Ada proceeded to answer the Ada training question as well; the responses indicate seminars and informal training are the most important in this group. As well, 14 people with "other" involvement said they had informal training. In the next section, the relationship between years in the industry and Ada training will be considered. The level of involvement question is not very specific; the training question is probably a more specific version of the same thing.

Two-Variable Comparisons

The first part of this section is devoted to presentation of the parts of the survey data that associate some measure of knowledge or importance with various categories. The first is general activities, rated by importance; the second is methodologies and concepts rated by knowledge; the third is programming languages, rated by proficiency. The second part of this section is concerned with relating some of the various questions to each other. The following contingency tables have chi square statistics significant at the 99 percent level unless stated otherwise.

General activities.--The following table contains counts and percents for the three importance categories (primary, secondary, marginal). Percents are

relative to the respondents (count divided by 428). The last column is the total number of responses for the activity.

General Activities by Importance

Activity	Primary Percent		Secondary Count		Marginal Percent		Total Count
	Primary Count	Secondary Percent	Primary Count	Secondary Percent	Marginal Count		
program management	30	7.009	38	8.879	41	9.579	109
sales marketing	8	1.869	24	5.607	40	9.346	72
contract negotiation	5	1.168	19	4.439	46	10.748	70
formulating policy	27	6.308	32	7.477	48	11.215	107
formulating strategy	43	10.047	45	10.514	56	13.084	144
preparing budgets	59	13.785	74	17.290	61	14.252	194
technical management	121	28.271	47	10.981	37	8.645	205
interviewing personnel	34	7.944	49	11.449	90	21.028	173
preparing schedules	68	15.888	98	22.897	75	17.523	241
preparing mgmt info rpts	47	10.981	78	18.224	55	12.850	180
preparing fld eng rpts	1	0.234	15	3.505	26	6.075	42
other admin tasks	20	4.673	53	12.383	67	15.654	140
tech advice to CCB	33	7.710	50	11.682	58	13.551	141
maint config procs	20	4.673	44	10.280	67	15.654	131
library control	12	2.804	32	7.477	57	13.318	101
prep version audits	12	2.801	15	3.505	33	7.710	60
quality assurance	25	5.841	36	8.411	62	14.486	123
prep temp eng change rpt	20	4.673	54	12.617	56	13.084	130
prep technical rpts	45	10.514	93	21.729	96	22.430	234
reading tech pubs	55	12.850	146	34.112	124	28.972	325
reviewing tech work	102	23.832	143	33.411	66	15.421	311
teaching	44	10.280	105	24.533	124	28.972	273
updating training manual	11	2.570	34	7.944	80	18.692	125
being trained	42	9.813	83	19.393	136	31.776	261
func system design	193	45.093	67	15.654	62	14.486	322
func module design	215	50.234	72	16.822	44	10.280	331
def global data strucs	137	32.009	100	23.364	59	13.785	296
def subsystem interface	157	36.682	103	24.065	58	13.551	318
def stuff for own use	174	40.654	78	18.224	54	12.617	306
coding	195	45.561	65	15.187	58	13.551	318
debugging or modifying	204	47.664	61	14.252	56	13.084	321
prep sys rqt docs	96	22.430	86	20.093	69	16.121	251
updating MIL STD specs	26	6.075	47	10.981	46	10.748	119
prep redlined docs	41	9.579	56	13.084	63	14.720	160
prep version descr mnls	29	6.776	50	11.682	85	19.860	164
prep user manuals	44	10.280	108	25.234	91	21.262	243
documenting code	141	32.944	108	25.234	42	9.813	291
defining test cases	87	20.327	101	23.598	65	15.187	253
prep test drivers	58	13.551	94	21.963	73	17.056	225
prep test plans	99	23.131	99	23.131	72	16.822	270
hardware testing	23	5.374	41	9.579	79	18.458	143
system software test	162	37.850	98	22.897	53	12.383	313
defining mod test cases	115	26.869	85	19.860	74	17.290	274

software module testing	152	35.514	81	18.925	55	12.850	288
documenting test results	75	17.523	112	26.168	78	18.224	265
prep trouble reports	66	15.421	85	19.860	86	20.093	237
analyzing trouble rpts	85	19.860	89	20.794	67	15.654	241

The above values, ranked by the count of primary importance, are:

Activity	Primary Percent of Respondents
func_module_design	50.234
debugging_or_modifying	47.664
coding	45.561
func_system_design	45.093
def_stuff_for_own_use	40.654
system_software_test	37.850
def_subsystem_interface	36.682
software_module_testing	35.514
documenting_code	32.944
def_global_data_structs	32.009
technical_management	28.271
defining_mod_test_cases	26.869
reviewing_tech_work	23.832
prep_test_plans	23.131
prep_sys_mnt_docs	22.430
defining_test_cases	20.327
analyzing_trouble_rpts	19.860
documenting_test_results	17.523
preparing_schedules	15.888
prep_trouble_reports	15.421
preparing_budgets	13.785
prep_test_drivers	13.551
reading_tech_pubs	12.850
preparing_mgmt_info_rpts	10.981
prep_technical_rpts	10.514
teaching	10.280
prep_user_manuals	10.280
formulating_strategy	10.047
being_trained	9.813
prep_redlined_docs	9.579
interviewing_personnel	7.944
tech_advice_to_CCB	7.710
program_management	7.009
prep_version_descr_mnls	6.776
formulating_policy	6.308
updating_MIL_STD_specs	6.075
quality_assurance	5.841
hardware_testing	5.374
prep_tech_eng_change_rpt	4.673
other_admin_tasks	4.673
maint_config_procs	4.673
prep_version_audits	2.304

library_control	2.804
updating_training_manual	2.570
sales_marketing	1.869
contract_negotiation	1.168
preparing_fld_eng_rpts	0.234

This ranking indicates the importance of the activities to the industry as a whole, not to the respondents. The scheme presented below in the classification section shows that various subgroups who engage in very different activities exist; a particular activity may be much more important to one group than to another. As well, the ranking does not address the conceptual importance of the activities; contract negotiation isn't done by very many people, but it is clearly a vital industry activity.

Methodologies.--The following table presents the methodologies crosstabulated with the knowledge of the methodology. The percents are calculated as percent of the respondents (count divided by 428).

Knowledge of Methodologies

Methodology	Percent		Used Moderately		Used Frequently	
	Heard of	Know Concept	Percent	Percent	Percent	Percent
PSL PLA	48	11.215	88	20.561	15	3.505
SADT	23	5.374	36	8.411	18	4.206
SREM	22	5.140	34	7.944	4	0.935
HIPO	49	11.449	128	29.907	91	21.262
Jackson Design	34	7.944	36	8.411	13	3.037
Structured Design	14	3.271	75	17.523	118	27.570
Warnier Orr Design	39	9.112	51	11.916	18	4.206
N S Chapin Chart	38	8.879	40	9.346	17	3.972
Beamson Tables	9	2.103	6	1.402	1	0.234
Program Design language	35	8.178	104	24.299	98	22.897
Structured Programming	7	1.636	53	12.383	103	24.065
Structured Walkthroughs	20	4.673	121	28.271	124	28.972
Top Down Design	6	1.402	54	12.617	121	28.271
Top Down Testing	15	3.505	112	26.168	118	27.570
Bottom Up Design	18	4.206	170	39.720	109	25.467
Bachman Diagramming	15	3.505	23	5.374	2	0.467
Entity Diagrams	12	2.804	22	5.140	7	1.636
Data Abstraction	40	9.346	77	17.991	45	10.514
other methodology	0	0.000	2	0.467	5	1.168

enumeration types	28	6.542	72	16.822	68	15.888	59	13.785
floating point types	7	1.636	73	17.056	128	29.907	205	47.897
fixed point types	5	1.168	46	10.748	106	24.766	246	57.477
user defined types	20	4.673	85	20.561	111	25.935	139	32.477
pointers	6	1.402	64	14.953	106	24.766	220	51.402
typed pointers	38	8.879	96	22.430	65	15.187	104	24.299
ranges	15	3.505	97	22.664	104	24.299	143	33.411
records	10	2.336	72	16.822	108	25.234	206	48.131
variant records	35	8.178	89	20.794	68	15.888	93	21.729
object type decls	18	4.206	75	17.523	105	24.533	148	34.579
global variables	6	1.402	34	7.944	94	21.963	282	65.888
local variables	5	1.168	31	7.243	88	20.561	289	67.523
formal actual params	19	4.439	42	9.813	79	18.458	217	50.701
reserved words	9	2.103	58	13.551	85	19.860	243	56.776
blocks	7	1.636	65	15.187	98	22.897	209	48.832
case statements	11	2.570	52	12.150	101	23.598	227	53.037
if then else statements	2	0.467	33	7.710	93	21.729	290	67.757
loop for while until	2	0.467	42	9.813	98	22.897	274	64.019
exit statements	3	0.701	76	17.757	123	28.738	206	48.131
procedures	9	2.103	35	8.178	95	22.196	273	63.785
functions	5	1.168	32	7.477	108	25.234	270	63.084
return statements	3	0.701	32	7.477	80	18.692	303	70.794
clusters modules package	28	6.542	75	17.523	105	24.533	155	36.215
stubs	17	3.972	81	18.925	103	24.065	145	33.879
goto statements	1	0.234	67	15.654	135	31.542	211	49.299
comments	1	0.234	13	3.037	73	18.224	323	75.467
exception handlers	24	5.607	98	22.897	95	22.196	133	31.075
task coroutines	26	6.075	113	26.402	78	18.224	115	26.869
other prog constructs	0	0.000	2	0.467	2	0.467	3	0.701
importing exporting name	39	9.112	45	10.514	21	4.907	33	7.710
data encapsulation	48	11.215	93	21.729	61	14.252	60	14.019
name scoping	33	7.710	65	15.187	34	7.944	60	14.019
name visibility	40	9.346	62	14.486	32	7.477	47	10.981
static dynamic nesting	57	13.318	86	20.093	60	14.019	75	17.523
iteration	13	3.037	48	11.215	109	25.467	222	51.869
conditional statements	7	1.636	22	5.140	88	20.561	286	66.822
recursion	14	3.271	102	23.832	138	32.243	122	28.505
concurrency	29	6.776	120	28.037	80	18.692	82	19.159
strong typing	31	7.243	90	21.028	57	13.318	65	15.187
type conversion	29	6.776	81	18.925	96	22.430	110	25.701
data abstraction	56	13.084	110	25.701	45	10.514	59	13.785
generics	45	10.514	97	22.664	40	9.346	32	7.477
loop invariants	51	11.916	76	17.757	48	11.215	50	11.682
parameter binding	53	12.383	84	19.626	38	8.879	45	10.514
version number	20	4.673	73	17.056	77	17.991	136	31.776
other prog concepts	1	0.234	0	0.000	1	0.234	5	1.168
Ada enumeration types	44	10.280	103	24.065	28	6.542	16	3.738
Ada user defined types	40	9.346	149	34.813	30	7.009	30	7.009
Ada subtypes	56	13.084	105	24.533	18	4.206	19	4.439
Ada derived types	63	14.720	83	19.393	17	3.972	9	2.103
Ada real types	31	7.243	149	34.813	32	7.477	51	11.916
Ada float point types	26	6.075	162	37.850	36	8.411	59	13.785
Ada fixed pt types	24	5.607	156	36.449	34	7.944	61	14.252

Ada record types	35	8.178	138	32.243	31	7.243	45	10.514
Ada rec types discrim	57	13.318	75	17.523	14	3.271	9	2.103
Ada slices	42	9.813	51	11.916	10	2.336	8	1.869
Ada aggregates	45	10.514	59	13.785	9	2.103	8	1.869
Ada allocators	38	8.879	64	14.953	11	2.570	6	1.402
Ada access types	39	9.112	85	19.860	14	3.271	14	3.271
Ada overloading	40	9.346	83	19.393	9	2.103	8	1.869
Ada packages	43	10.047	109	25.467	18	4.206	12	2.804
Ada private types	35	8.178	105	24.533	13	3.037	6	1.402
Ada scope	32	7.477	113	26.402	12	2.804	22	5.140
Ada short circuiting	35	8.178	49	11.449	7	1.636	5	1.168
Ada visibility	40	9.346	93	21.729	13	3.037	13	3.037
Ada tasking	55	12.850	133	31.075	14	3.271	27	6.308
Ada task types	58	13.551	101	23.598	10	2.336	10	2.336
Ada rendezvous	30	7.009	92	21.495	10	2.336	7	1.636
Ada entries	34	7.944	106	24.766	13	3.037	19	4.439
Ada entry families	47	10.981	47	10.981	7	1.636	6	1.402
Ada separate compilation	26	6.075	147	34.346	24	5.607	41	9.579
Ada exceptions	33	7.710	116	27.103	22	5.140	21	4.907
Ada generic prog units	46	10.748	81	18.925	12	2.804	5	1.168
Ada instantiation	39	9.112	66	15.421	7	1.636	6	1.402
Ada elaboration	37	8.645	47	10.981	6	1.402	6	1.402
Ada context spec	47	10.981	59	13.785	6	1.402	4	0.935
Ada information hiding	38	8.879	108	25.234	9	2.103	14	3.271
Ada mutual recursion	50	11.682	59	13.785	7	1.636	5	1.168
other Ada concepts	1	0.234	0	0.000	0	0.000	0	0.000

The above table has no valid chi square statistic due to low cell expectancies for some cells (that is, an unequal distribution of responses makes the chi square statistic unreliable). The list below contains the methodologies ranked by the summed percentages of those methodologies used frequently or moderately. Each group of methodologies (methodologies, programming constructs, programming concepts, and Ada concepts) is separately ranked.

Methodology	Percent of Respondents
methodologies	
Structured Programming	85.047
Top Down Design	83.879
Structured Design	70.794
Top Down Testing	63.318
Structured Walkthroughs	55.841
Program Design language	51.869
Bottom Up Design	41.589
HIPO	28.271

Data Abstraction	16.822
Jackson Design	5.374
Warnier Orr Design	5.374
N S Chapin Chart	5.374
SAOT	4.673
PSL PLA	3.505
other methodology	3.037
Entity Diagrams	2.336
Bachman Diagramming	1.168
SREM	1.168
Beamson Tables	0.234

programming constructs

comments	93.692
return statements	89.486
if then else statements	89.486
functions	88.318
local variables	88.084
global variables	87.850
loop for while until	86.916
procedures	85.981
fixed point types	82.243
goto statements	80.841
floating point types	77.804
exit statements	76.869
case statements	76.636
reserved words	76.636
pointers	76.168
records	73.364
blocks	71.729
formal actual params	69.159
clusters modules package	60.748
object type dcis	59.112
user defined types	58.411
stubs	57.944
ranges	57.710
exception handlers	53.271
task coroutines	45.093
typed pointers	39.486
variant records	37.617
enumeration types	29.673

programming concepts

conditional statements	87.383
iteration	77.336
recursion	60.748
version number	49.766
type conversion	48.131
concurrency	37.850
static dynamic nesting	31.542
strong typing	28.505

data encapsulation	28.271
data abstraction	24.299
loop invariants	22.897
name scoping	21.963
parameter binding	19.393
name visibility	18.458
generics	16.822
importing exporting name	12.617
other prog concepts	1.402
other prog constructs	1.168

Ada concepts

Ada float point types	22.196
Ada fixed pt types	22.196
Ada real types	19.393
Ada record types	17.757
Ada separate compilation	15.187
Ada user defined types	14.019
Ada enumeration types	10.280
Ada exceptions	10.047
Ada tasking	9.579
Ada subtypes	8.645
Ada scope	7.944
Ada entries	7.477
Ada packages	7.009
Ada access types	6.542
Ada visibility	6.075
Ada derived types	6.075
Ada information hiding	5.374
Ada rec types discrim	5.374
Ada task types	4.673
Ada private types	4.439
Ada slices	4.206
Ada overloading	3.972
Ada allocators	3.972
Ada rendezvous	3.972
Ada aggregates	3.972
Ada generic prog units	3.972
Ada entry families	3.037
Ada instantiation	3.037
Ada short circuiting	2.804
Ada elaboration	2.804
Ada mutual recursion	2.804
Ada context spec	2.336
other Ada concepts	0.000

The above list describes the order of frequency of use of the methodologies.

Two facts stand out immediately. The first is that programming constructs are most used. The second is that Ada concepts are least used (preceded by pro-

gramming concepts). This list presents a picture of a technically knowledgeable but theoretically unsophisticated industry. To the extent that Ada is based on theoretical concepts rather than "standard" programming constructs, Ada training will have to involve theoretical training in computer science as well as just instruction in the language.

Programming languages.--The following table contains programming languages crosstabulated with proficiency at the language. The first count and percent are all proficiencies; the second count and percent are people who said they were most proficient at the language; the third count and percent are people who said they were most proficient or second-most proficient at the language. The percents are relative to respondents (count divided by 428).

Proficiencies at Programming Languages

Language	Count All	Percent		Percent
		Count	First and Second	
JOVIAL	89	20.794	15	3.505
CMS 2	135	31.542	44	10.280
C	36	8.411	1	0.234
FORTRAN	399	93.224	210	49.065
COBOL	206	48.131	36	8.411
ASSEMBLER	378	88.318	214	50.000
PLI	199	46.495	42	9.813
PASCAL	209	48.832	77	17.991
BASIC	277	64.720	42	9.813
ALGOL	92	21.495	5	1.168
RATFOR WATFOR WATFIV	95	22.196	0	0.000
MODULA	8	1.869	0	0.000
SIMULA	12	2.804	0	0.000
XPL	11	2.570	0	0.000
FORTH	31	7.243	1	0.234
Ada	107	25.000	4	0.935
LISP	66	15.421	2	0.467
SNOBOL	75	17.523	0	0.000
ECL	3	0.701	0	0.000
GPSS	62	14.486	3	0.701
SAS	7	1.636	0	0.000
PROTEGE	1	0.234	1	0.234
APL	112	26.168	13	3.037
Other	95	22.196	31	7.243

The following list presents the languages in order of percentage of respondents who said it was their first or second language (the sixth column in the above table).

Language	Percent of Respondents
ASSEMBLER	50.000
FORTRAN	49.065
PASCAL	17.991
CMS_2	10.280
BASIC	9.813
PLI	9.813
COBOL	8.411
Other	7.710
JOVIAL	3.505
APL	3.037
ALGOL	1.168
Ada	0.935
GPSS	0.701
LISP	0.467
PROTEGE	0.234
C	0.234
FORTH	0.234
XPL	0.000
SNOBOL	0.000
ECL	0.000
SIMULA	0.000
SAS	0.000
MODULA	0.000
RATFOR WATFOR WATFIV	0.000

The above list makes clear the structure of the programming knowledge of the greatest part of the sample: they are Assembler or FORTRAN programmers. Respectively, 50 and 49 percent of the respondents rank Assembler and FORTRAN as their first or second language. The next highest language, Pascal, comes in at 18 percent, not even close. Surprisingly, languages such as PL/I are rather far down in the list.

Again, this structure is not indicative of strong theoretical knowledge of computer science but rather of strong practical knowledge. This supports the comments above on methodologies. It would be unwise to design an Ada

training program that doesn't take the current, practical structure of the industry into account.

Also of interest are the relationships between FORTRAN and ASSEMBLER and FORTRAN and BASIC. The above table noted that FORTRAN and ASSEMBLER are the two best known languages; the following table displays the counts of people who were proficient in ASSEMBLER for each other language at which they were proficient.

Other Language	Count	Percent
FORTRAN	101	49.510
PASCAL	27	13.235
CMS_2	17	8.333
COBOL	15	7.353
PLI	12	5.882
BASIC	10	4.902
Other	10	4.902
APL	5	2.451
JOVIAL	4	1.961
GPSS	1	0.490
FORTH	1	0.490
ALGOL	1	0.490

This table indicates that people who use ASSEMBLER usually use FORTRAN--50 percent of the people who picked one picked the other as languages in which they were proficient. Pascal came in next, but wasn't close.

The following table gives the counts and percentages of people who ranked only FORTRAN (without BASIC), only BASIC (without FORTRAN), both together, or neither, in their choices for first or second proficiency.

	Count	Percent
FORTRAN only	189	44.3
BASIC only	21	4.9
both	21	4.9
neither	196	45.9

From this frequency table, it would appear that BASIC and FORTRAN knowledge are quite independent; those who use FORTRAN generally don't use BASIC as their other language, and those who use BASIC half the time don't use FORTRAN. This applies only to languages ranked with first and second proficiency ratings.

Relationships.--The following paragraphs address some topics of interest relating to relationships between some of the information presented above. I will consider six such relationships: (1) Ada involvement and knowledge of Ada concepts; (2) Ada training and knowledge of Ada concepts; (3) various interrelationships between belonging to technical societies, publishing or presenting papers, technical reading, and going to conferences; (4) areas of experience and knowledge of concepts; (5) years of experience and programming languages; and (6) Ada training and years of experience for those who know Ada.

To study the relationship between involvement with the Ada language and knowledge of Ada concepts, a measure of knowledge was constructed by summing the knowledge scores over the Ada concepts and dividing by the number of elements; this produces an average score of knowledge of Ada concepts for each respondent. A measure of Ada involvement was constructed by equating "other" involvement to "training" so as to make a continuous scale. Knowledge was regressed on involvement, obtaining the following regression equation:

$$\text{knowledge} = -0.075 + 0.575 * \text{involvement} + e \\ (0.059)$$

That is, the value of the knowledge measure for a given individual is equal to the involvement measured multiplied by .575 plus a constant (-.075) plus a random error. .575 is known as the "coefficient" for involvement. The coef-

ficient for involvement was significant at the 99.9 percent level. The regression equation was also significant, since there was only one coefficient. The coefficient of determination (R squared) was 0.185, indicating that only 19 percent of the variance in knowledge is explained by involvement. This result indicates that although involvement has some effect on knowledge, that effect isn't very large.

A similar approach was taken to the comparison of Ada training techniques to knowledge of Ada concepts. The same measure of knowledge was used, but instead of regression, analysis of variance was used, since the Ada training attribute is nominal in scale, not interval. The analysis of variance was not significant; that is, the variation was too unreliable to enable any conclusions. This is probably due to the relatively small number of respondents who have been trained in Ada or who have knowledge of Ada.

Since all of the attributes involving professional activities were nominal, all of these comparisons were made as contingency tables. None of the tables had valid chi square statistics due to either zero marginals or small cell expectations. Therefore, no conclusions have been drawn as to relationships between these variables. However, the four-way table below presents the frequencies of these attributes for what they are worth. Each of the tables represents a "slice" of the four-way table for one combination of the conference attendance and publishing attributes.

conference attendance, published or presented:
yes, yes

technical societies

	yes	no
technical reading extent		
regularly	33	8
occasionally	8	7
only as my job demands	0	0
never	0	0
other	0	0

yes, no

technical societies

	yes	no
technical reading extent		
regularly	31	33
occasionally	10	32
only as my job demands	2	6
never	0	1
other	0	0

no, yes

technical societies

	yes	no
technical reading extent		
regularly	16	7
occasionally	3	10
only as my job demands	0	3
never	0	0
other	0	0

no, no

technical societies

	yes	no
technical reading extent		
regularly	20	35
occasionally	16	99
only as my job demands	2	19
never	0	12
other	0	2

There was no significant relationship between areas of experience and knowledge of concepts, meaning that no conclusions may be drawn statistically

about this relationship.

Because years of involvement can be considered scaled, an analysis of variance was performed with that attribute as the response variable and programming language as the factor. The model was not significant, and no conclusions may be drawn as to the relationship. The following table presents the counts for the crosstabulation; it has no valid chi square statistic due to zero marginals.

language	years of involvement				
	two to five years		five to ten years		
	less than two years		over ten years		
JOVIAL	5	12	12	58	87
CMS 2	7	22	26	78	133
C	5	11	5	15	36
FORTRAN	36	91	81	187	395
COBOL	12	46	39	105	202
ASSEMBLER	31	83	76	184	374
PLI	11	53	43	88	195
PASCAL	28	64	37	78	207
BASIC	25	71	50	128	274
ALGOL	8	20	16	48	92
RATFOR WATFOR WATFIV	14	27	21	33	95
MODULA	0	3	4	1	8
SIMULA	0	1	6	5	12
XPL	1	4	2	4	11
MMP	0	0	0	0	0
FORTH	1	8	6	16	31
Ada	7	26	11	62	106
LISP	9	21	16	20	66
SNOBOL	8	24	19	24	75
ECL	0	0	1	2	3
GPSS	2	12	14	33	61
SAS	0	2	3	2	7
PROTEGE	0	1	0	0	1
PPL	0	0	0	0	0
APL	13	28	26	45	112
Other	5	19	23	47	94
	228	649	537	1263	2677

The following table combines the lesser-known programming languages into a residual category and has a significant chi square statistic.

language	years of involvement				
	two to five years		five to ten years		
	less than two years		over ten years		
JOVIAL	5	12	12	58	87
CMS 2	7	22	26	78	133
FORTRAN	36	91	81	187	395
COBOL	12	46	39	105	202
ASSEMBLER	31	83	76	184	374
PLI	11	53	43	88	195
PASCAL	28	64	37	78	207
BASIC	25	71	50	128	274
Other	73	207	173	357	810
	228	649	537	1263	2677

Finally, the relationship between Ada training and years of involvement is presented in the following table. This table has no valid chi square value due to low cell expectations. It does show, however, that a large portion of the people who know Ada are people in the industry for over ten years. These are the people who tend to know more programming languages as well. The training doesn't seem to show any pattern different from the usual pattern for any group--seminars, in-house courses, and informal training account for most of the training in all categories.

Ada training	years of involvement				
	two to five years		five to ten years		
	less than two years		over ten years		
videotape	2	5	0	9	16
college course	0	3	0	4	7
in house course	6	14	3	23	46
programmed learning	0	3	0	2	5
seminar	2	9	5	27	43
informal training	5	11	7	28	51
other	2	4	5	10	21
	17	49	20	103	189

Classification of the Workforce

Methods.--Before giving the results, the report will summarize briefly the problems faced in classifying the respondents. The goal was to induce some classification scheme based on job outputs, principal duties, and general activities that reflects a meaningful categorization of the workforce.

The first problem was that the basic variables were nominal or categorized variables, variables with no inherent ordering between values. This meant that none of the classification techniques such as discriminant analysis or metric distance clustering would be appropriate, since they are all based on metric data.

Second, the data was multiple response. Each of the three questions had a different number of responses. This meant that the judgment of similarity would have to take into account the range of possible values, making the judgment relative to all the possibilities.

Third, the general activities were all rated as to importance. It would have been useful to incorporate this information into the judgment of similarity in some way in that it probably would have eliminated some of the noise in the classification. These measures were not used in the following analysis, however; there is reason to believe that including them would not have made much difference. The methods used below are not particularly sensitive to even moderate changes in the measurement of similarity.

Fourth, there are a very large number of response values, 114, to be considered. The size of the conceptual area to be integrated is well beyond the capability of the human mind to comprehend. Nobody can look at 114 different response values and judge differences. There needs to be some method of

summarizing the information into a smaller conceptual area.

The method chosen involves five steps: (1) expand the data to take into account all possible responses, not just those actually responded; (2) calculate a similarity coefficient based on pattern matching; (3) cluster the respondents based on the similarity measure; (4) determine the optimal number of clusters by taking various cross-sections of the tree and analyzing them; and (5) use median polishing to produce an interpretation of the optimal cluster pattern.

The original data is in the form of a table where rows are individuals and columns are responses. The values of the table cells are zeros and ones, a zero indicating the response was not given, a one indicating that it was. The string of zeros and ones represents the pattern of the individual's responses.

For a given pair of individuals, the patterns are compared, one column at a time. If each has a one, or if each has a zero, the individuals are more similar. If the two values are different, the individuals are less similar. Mathematically, the measure of similarity is calculated using the following table:

	1	0
1	a	b
0	c	d

"a" represents the count of positive matches--where both individuals have ones. "d" represents the count of negative matches--where both individuals have zeros. "b" and "c" are the counts of nonmatches. Similarity is $a + d / (a + b + c + d)$. That is, similarity is the number of matches divided by the number of possible matches. Similarity varies between zero and one, where zero means completely dissimilar, and one means completely similar.

The similarity program produces a square matrix of similarity coefficients--each individual compared to each other individual. The matrix is n by n , where n is the number of individuals. This is similar to a correlation matrix.

The similarity matrix is input to the clustering program. Hierarchical clustering is a process that starts with all individuals separated and proceeds to group the individuals into larger and larger clusters, finally resulting in one large cluster. The diagram produced by the program resembles a tree. At each level of the tree, another individual is clustered; thus if there are n individuals, there will be $n - 1$ levels in the tree.

When a given individual is clustered with another, that cluster becomes an individual for the purposes of the next clustering. Therefore, the similarity matrix needs to be adjusted; one individual is removed, and the similarities between the new individual (the last cluster produced) and all the other individuals (or clusters) has to be recalculated. The difference between different methods of clustering is based on how to recalculate these similarities. The technique used below calculates the similarity as the minimum of the similarities between the individuals in the new cluster and the individuals in the cluster to which the new cluster is being compared.

For example, say you have five individuals. At the third level, 1 and 5 are together in a cluster, 3 and 4 are together in another cluster, and 2 is still an individual. The new similarity matrix would be a 3 by 3 matrix, since there are now only three individuals. The similarity of 1 and 5 to 3 and 4 would be the smallest of the four similarities, 1 to 3, 1 to 4, 5 to 3, or 5 to 4. This clustering method is not sensitive to the measurement level of the original data; since the data is composed of zeros and ones in an unordered sequence, such an insensitive method is more appropriate than other

methods involving averages or weighted averages of similarities.

Once the cluster tree has been produced, it must be interpreted. The first problem is to figure out where to cut the tree; that is, to determine how many disjoint clusters are most appropriate or most meaningful. There is no generally agreed-upon method for doing this. The next problem is to produce a meaningful interpretation of the disjoint clusters selected.

Using the interpretive method discussed below, five clusters were selected. The following table lists the number of respondents in each cluster; the list does not sum to 428, since several individuals were removed from the analysis due to excessive missing values (no job outputs or principal duties and so forth).

Cluster	Counts	Percents
cluster 2	46	11.7
cluster 3	169	42.9
cluster 5	135	34.2
cluster 38	14	3.6
cluster 57	30	7.6

Since the response structure is so large, it is necessary to summarize the information in some way. The approach used below is called median polishing.

Median polishing begins with a table of values. In this case, the table is a crosstabulation of cluster versus the several responses; that is, it is a 5 by 114 table, 5 clusters, 114 possible responses. The cells of the table are counts--how many individuals in the cluster responded with the particular response.

Median polishing is the process of repeatedly subtracting the median values of the rows and columns of the table from the cells of the table. For example, say there was a 3 by 3 table such as the following:

	a	b	c
x	3	6	2
y	5	2	1
z	1	9	2

The first step is to take the medians of the rows: 3, 2, 2. These numbers are subtracted from the values in the rows, giving the following table:

	a	b	c	row med
x	0	3	-1	3
y	3	0	-1	2
z	-1	7	0	2

The next step is to repeat this process for the columns; in addition, the median of the row medians (2 in this case) is subtracted from the row medians and is placed in the lower right-hand corner of the table.

	a	b	c	row med
x	0	3	-1	1
y	3	0	-1	0
z	-1	7	0	0
col med	0	0	-1	2

The process is repeated once more for the rows, this time subtracting the median of the column medians from the column medians as well as adding the new row medians to the old medians.

	a	b	c	row med
x	0	3	-1	1
y	3	0	-1	0
z	-1	7	0	0
col med	0	0	-1	2

At this point, all the new row medians are zero, so nothing changes. The final pass repeats the column median polish.

	a	b	c	row med
x	0	3	-1	1
y	3	0	-1	0
z	-1	7	-2	0
col med	0	0	-2	2

At this point, little change will happen with additional passes, although the process may be continued until no change occurs. There are now four components to the polished table, the row and column medians, the median of the medians, and the table of left-over values. These are termed, respectively, row and column effects, common effect, and residuals. In the case of the table used in the classification interpretation, the effects represent expected sizes of cells in the table of counts given a simple additive model,

$$\text{cell count} = \text{common} + \text{row} + \text{column} + \text{residual}.$$

That is, the count in a particular cell (the conjunction of a cluster and a particular response) is the sum of the common effect, the cluster effect, the response effect, plus the residual. For a more detailed explanation of median polishing, consult the references given in Appendix 3.

The larger the particular effect, the larger the influence of that cluster or value on the cell value. Thus the row effects indicate the influence of cluster membership on the expected frequency of picking a particular response, and the column effects indicate the influence of a particular response on the expected frequency of a cluster's members picking the response. But the residuals, the table of values left over, provide a much more interesting thing--a measure of unusualness. A cell residual, compared to other cell

cluster2	11.500
cluster3	46.500
cluster5	0.000
cluster38	-13.500
cluster57	-7.500
hard soft tradeoff eval	0.000
data flow diagrams	13.000
test drivers	8.500
jo code	15.000
prog design language	14.000
requirements specs	19.500
design specs	21.000
test plans	15.000
integration plans	7.500
management plans	7.000
cost data	6.000
analysis reports	3.000
milestone charts	9.000
status reports	14.000
interview sheets	6.000
correspondence	-1.000
development other	-11.500
STR analyses	-2.000
temp ECPs	-13.500
redlined documentation	-8.000
support test plans	-5.000
support test drivers	-10.500
jo tech advice to CCB	-13.500
updated MIL STD spec	-14.500
jo library control	-11.500
maintain config procs	-10.500
updated training manuals	-10.500
updated user manuals	-3.000
STRs	-4.000
automated build systems	-11.500
management info reports	-10.500
version descrip docs	-13.500
version audits	-16.500
field engineering report	-16.500
support other	-8.500
conduct req review	-3.500
attend req review	8.000
system analysis	16.000
design	17.000
conduct design review	8.500
attend design review	8.000
pd code	6.000
conduct walkthroughs	-1.500
attend walkthroughs	1.000
formulation of policy	-8.000
formulation of strategy	-4.000
pd technical management	6.000

residuals in a column, indicates the relative unusualness of the cluster with respect to the other clusters on the particular response. Thus if one cluster has a much higher residual for one response than the other clusters, that cluster has given that response relatively more often; conversely, a high negative residual means the cluster gave that response less often.

Given this interpretation of the residuals, it is straightforward to rank the residuals in each column in ascending and descending order. These ranks represent the relative unusualness of the cluster with respect to that response. The next step is to collect the instances where the cluster was ranked 1 or 2 (and possibly 3); this indicates on which responses the cluster was unusual. There are two groups of ranks for each cluster, positive high ranks and negative high ranks. These show, respectively, what the clusters do do unusually and what they don't do unusually. Thus, the comparison of the ranks gives a direct interpretation of the patterns of the clusters' responses.

Applied to the decision about how many clusters to pick, this approach is tried for several different numbers of clusters, and the best or most interpretable number is chosen. Applied to this final choice, the method gives a summary interpretation of the patterns of the clusters.

The results.--Given all that, the following is the final interpretive median polish for the optimal cluster pattern, which involves five clusters.

The rankings are from 1 to 2; values of 1.5 indicate ties at the first rank--the highest or lowest residuals in a given column were the same.

Cluster Structure using All Job Outputs, General Activities, and Principle Duties

Effects and Typicalis after Polishing

effects

common

16.500

pd program management	-15.500
configuration management	-12.500
pd quality assurance	-10.500
monitoring contracts	-12.500
other development	-10.500
support analysis	0.000
support design	-2.000
conduct support dr	-9.500
attend support dr	-3.000
code patch	-2.000
conduct sup walkthrough	-12.500
attend sup walkthrough	-7.000
sup technical management	-11.500
sup policy formulation	-16.500
sup program management	-15.500
SCCS participation	-13.500
sup config management	-10.500
sup quality assurance	-13.500
sup monitoring contracts	-10.500
other support	-9.500
ga program management	-4.000
sales marketing	-2.000
contract negotiation	-4.000
formulating policy	-2.000
formulating strategy	5.000
preparing budgets	9.000
ga technical management	15.000
interviewing personnel	9.000
preparing schedules	15.000
preparing mgmt info rpts	10.000
preparing fld eng rpts	-14.500
other admin tasks	6.000
ga tech advice to CCB	3.000
maint config procs	2.000
ga library control	-5.500
prep version audits	-8.000
ga quality assurance	0.000
prep temp eng change rpt	-2.000
prep technical rpts	16.000
reading tech pubs	18.000
reviewing tech work	18.000
teaching	14.000
updating training manual	0.000
being trained	12.000
func system design	17.000
func module design	15.000
def global data strucs	13.000
def subsystem interface	16.000
def stuff for own use	11.000
coding	6.000
debugging or modifying	7.000
prep sys rqt docs	17.000
updating MIL STD specs	-3.000

prep redlined docs	5.000
prep version descr mnls	3.000
prep user manuals	6.000
documenting code	2.000
defining test cases	10.000
prep test drivers	3.000
prep test plans	10.000
hardware testing	0.000
system software test	9.000
defining mod test cases	3.000
software module testing	0.000
documenting test results	2.000
prep trouble reports	4.000
analyzing trouble rpts	4.000

Residuals after Polishing

	test drivers				design specs			
	data flow diagrams		prog design language		requirements specs		test plans	
	hard	soft	tradeoff	eval	jo	code		
cluster2	7.000	-10.000	-16.500	-12.000	-14.000	-8.500	-7.000	-9.000
cluster3	-14.000	18.000	10.500	63.000	55.000	26.500	48.000	37.000
cluster5	-2.500	14.500	0.000	54.500	41.500	0.000	5.500	25.500
cluster38	0.000	-15.000	-10.500	-18.000	-17.000	-18.500	-24.000	-17.000
cluster57	9.000	0.000	1.500	0.000	0.000	0.500	0.000	0.000

	analysis reports				interview sheets			
	management plans		status reports		milestone charts		correspondence	
	integration	plans	cost	data				
cluster2	-5.500	2.000	0.000	0.000	3.000	0.000	0.000	0.000
cluster3	10.500	-48.000	-44.000	-29.000	0.000	23.000	-25.000	-30.000
cluster5	0.000	-13.500	-10.500	-8.500	-4.500	12.500	-10.500	-7.500
cluster38	-9.500	2.000	1.000	1.000	-1.000	-6.000	0.000	10.000
cluster57	6.500	0.000	7.000	10.000	8.000	0.000	0.000	4.000

	redlined documentation				jo tech advice to CCB			
	STR analyses		support test drivers		support test plans		updated MIL STD spec	
	development	other	temp	ECPs				
cluster2	-14.500	-21.000	-13.500	-19.000	-21.000	-15.500	-12.500	-13.500
cluster3	-46.500	0.000	-13.500	-19.000	-12.000	-29.500	-22.500	-35.500
cluster5	0.000	4.500	0.000	1.500	3.500	0.000	0.000	0.000
cluster38	9.500	4.000	13.500	7.000	4.000	9.500	14.500	12.500
cluster57	3.500	-3.000	6.500	0.000	0.000	6.500	7.500	8.500

	updated training manuals	automated build systems	maintain config procs	STRs	version descrip docs	jo library control	updated user manuals	management info reports
cluster2	-15.500	-14.500	-15.500	-20.000	-20.000	-14.500	-16.500	-11.500
cluster3	-35.500	-34.500	-38.500	-20.000	-5.000	-40.500	-36.500	-39.500
cluster5	0.000	0.000	0.000	4.500	3.500	0.000	0.000	0.000
cluster38	9.500	10.500	7.500	0.000	5.000	8.500	12.500	12.500
cluster57	4.500	2.500	3.500	1.000	0.000	3.500	4.500	6.500

	conduct req review	conduct design review	field engineering report	system analysis	version audits	support other attend req review	design
cluster2	-11.500	-11.500	-19.500	-1.500	-15.000	-17.000	-20.000
cluster3	-39.500	-40.500	-49.500	-19.500	3.000	19.000	56.000
cluster5	0.000	0.000	0.000	0.000	2.500	3.500	43.500
cluster38	15.500	13.500	7.500	1.500	-5.000	-18.000	-20.000
cluster57	9.500	8.500	0.500	10.500	0.000	0.000	0.500

	conduct walkthroughs	formulation of strategy	pd code	formulation of policy	pd program management	attend design review	attend walkthroughs	pd technical management
cluster2	-17.000	-18.000	-12.500	-17.000	0.000	1.000	4.000	-0.500
cluster3	24.000	53.000	6.500	9.000	-44.000	-40.000	-31.000	-39.500
cluster5	30.500	56.500	0.000	10.500	-4.500	-10.500	-13.500	0.000
cluster38	-5.000	-9.000	-1.500	0.000	11.000	7.000	0.000	19.500
cluster57	0.000	0.000	1.500	-3.000	3.000	0.000	4.000	9.500

	monitoring contracts	support design	pd quality assurance	support analysis	attend support dr	configuration management	other development	conduct support dr
cluster2	-8.500	-11.500	-11.500	-15.500	-26.000	-24.000	-16.500	-25.000
cluster3	-37.500	-37.500	-39.500	-51.500	-4.000	-14.000	-23.500	-19.000
cluster5	0.000	0.000	0.000	0.000	5.500	9.500	0.000	6.500
cluster38	11.500	9.500	12.500	7.500	0.000	0.000	7.500	5.000
cluster57	7.500	3.500	4.500	2.500	2.000	2.000	4.500	0.000

	attend sup walkthrough	sup program management	conduct sup walkthrough	sup policy formulation	sup config management	code patch	sup technical management	SCCB participation
cluster2	-25.000	-14.500	-21.000	-12.500	-10.500	-8.500	-13.500	-16.500
cluster3	-17.000	-24.500	-22.000	-30.500	-37.500	-44.500	-27.500	-43.500
cluster5	16.500	0.000	1.500	0.000	0.000	0.000	0.000	0.000
cluster38	0.000	9.500	8.000	14.500	19.500	16.500	16.500	10.500
cluster57	0.000	6.500	0.000	7.500	8.500	6.500	8.500	2.500

			other support		contract negotiation			
		sup monitoring contracts		sales marketing		formulating strategy		
	sup quality assurance		ga program management		formulating policy			
cluster2	-14.500	-17.500	-17.500	10.000	0.000	3.000	14.000	10.000
cluster3	-33.500	-43.500	-48.500	-16.000	-44.000	-39.000	-24.000	-12.000
cluster5	0.000	0.000	0.000	-2.500	-7.500	-8.500	-7.500	-11.500
cluster38	13.500	10.500	6.500	11.000	6.000	7.000	8.000	4.000
cluster57	6.500	1.500	0.500	0.000	1.000	0.000	0.000	0.000

				preparing mgmt info rpts				
		interviewing personnel				ga tech advice to CCB		
		ga technical management		preparing fld eng rpts				
	preparing budgets		preparing schedules		other admin tasks			
cluster2	8.000	2.000	6.000	2.000	2.000	-0.500	0.000	5.000
cluster3	14.000	14.000	0.000	47.000	8.000	-24.500	1.000	5.000
cluster5	-1.500	-12.500	-6.500	-7.500	-7.500	0.000	-11.500	-7.500
cluster38	0.000	-6.000	1.000	-5.000	0.000	13.500	4.000	0.000
cluster57	-3.000	0.000	-3.000	0.000	-2.000	6.500	-9.000	-4.000

			prep version audits		prep technical rpts			
		ga library control		prep temp eng change rpt		reviewing tech work		
	maint config procs		ga quality assurance		reading tech pubs			
cluster2	2.000	6.500	0.000	3.000	0.000	0.000	0.000	0.000
cluster3	7.000	-6.500	-26.000	0.000	16.000	30.000	71.000	68.000
cluster5	-5.500	0.000	-6.500	-4.500	-0.500	5.500	37.500	27.500
cluster38	0.000	7.500	7.000	2.000	1.000	-8.000	-8.000	-7.000
cluster57	-8.000	-2.500	2.000	-6.000	-3.000	-8.000	-5.000	-5.000

			func system design		def subsystem interface			
	updating training manual		func module design		def stuff for own use			
	teaching	being trained		def global data strucs				
cluster2	0.000	4.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	56.000	0.000	50.000	79.000	85.000	77.000	84.000	83.000
cluster5	19.500	1.500	27.500	42.500	54.500	40.500	47.500	54.500
cluster38	-8.000	-1.000	-5.000	-14.000	-15.000	-15.000	-18.000	-13.000
cluster57	-3.000	-5.000	-10.000	-5.000	-4.000	-4.000	-7.000	-6.000

			updating MIL STD specs		prep user manuals			
	debugging or modifying		prep redlined docs		documenting code			
	coding	prep sys rqt docs		prep version descr mnls				
cluster2	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000
cluster3	88.000	89.000	57.000	12.000	25.000	23.000	60.000	88.000
cluster5	77.500	78.500	3.500	-1.500	3.500	14.500	40.500	70.500
cluster38	-9.000	-10.000	-17.000	0.000	-8.000	-4.000	-8.000	-5.000
cluster57	0.000	-3.000	-6.000	0.000	-8.000	-8.000	-8.000	-2.000

	prep test plans			defining mod test cases		documenting test results		
	prep test drivers			system software test				
	defining test cases			hardware testing		software module testing		
cluster2	0.000	0.000	0.000	-6.000	0.000	0.000	0.000	0.000
cluster3	72.000	72.000	79.000	26.000	88.000	89.000	95.000	85.000
cluster5	19.500	17.500	30.500	1.500	60.500	47.500	66.500	45.500
cluster38	-11.000	-6.000	-12.000	-3.000	-11.000	-6.000	-3.000	-3.000
cluster57	-6.000	0.000	-8.000	0.000	-2.000	-1.000	0.000	-1.000

analyzing trouble rpts
prep trouble reports

cluster2	0.000	0.000
cluster3	73.000	71.000
cluster5	31.500	31.500
cluster38	-5.000	-6.000
cluster57	-7.000	-3.000

Categories with High Positive
Residuals in Rank Order

Categories with High Negative
Residuals in Rank Order

Cluster 2

updating training manual	1.000	hardware testing	1.000
ga quality assurance	1.000	test drivers	1.000
formulating policy	1.000	attend design review	1.000
interviewing personnel	1.000	conduct design review	1.000
formulating strategy	1.000	attend walkthroughs	1.000
ga tech advice to CCB	1.500	attend req review	1.000
management plans	1.500	support analysis	1.000
pd technical management	1.500	support design	1.000
contract negotiation	2.000	STRs	1.000
formulation of strategy	2.000	support test plans	1.000
preparing budgets	2.000	code patch	1.000
ga technical management	2.000	STR analyses	1.000
interview sheets	2.000	attend support dr	1.000
preparing schedules	2.000	conduct walkthroughs	1.000
preparing mgmt info rpts	2.000	pd code	1.000
milestone charts	2.000	temp ECPs	1.500
maint config procs	2.000	redlined documentation	1.500
ga library control	2.000	updated user manuals	1.500
ga program management	2.000	design	1.500
hard soft tradeoff eval	2.000	updated MIL STD spec	2.000
		jo library control	2.000
		automated build systems	2.000
		integration plans	2.000
		test plans	2.000
		version audits	2.000
		field engineering report	2.000
		support other	2.000
		conduct req review	2.000

design specs	2.000
requirements specs	2.000
development other	2.000
prog design language	2.000
jo code	2.000
jo tech advice to CCB	2.000
support test drivers	2.000
system analysis	2.000
pd program management	2.000
configuration management	2.000
pd quality assurance	2.000
monitoring contracts	2.000
other development	2.000
version descrip docs	2.000
management info reports	2.000
conduct support dr	2.000
updated training manuals	2.000
maintain config procs	2.000
conduct sup walkthrough	2.000
attend sup walkthrough	2.000
sup technical management	2.000
sup policy formulation	2.000
sup program management	2.000
SCCB participation	2.000
sup config management	2.000
sup quality assurance	2.000
sup monitoring contracts	2.000
other support	2.000
preparing fld eng rpts	2.000
updating MIL STD specs	2.000
data flow diagrams	2.000

Cluster 3

data flow diagrams	1.000	hard soft tradeoff eval	1.000
test drivers	1.000	management plans	1.000
jo code	1.000	cost data	1.000
prog design language	1.000	analysis reports	1.000
requirements specs	1.000	interview sheets	1.000
design specs	1.000	correspondence	1.000
test plans	1.000	development other	1.000
integration plans	1.000	ga program management	1.000
status reports	1.000	other support	1.000
attend req review	1.000	sup monitoring contracts	1.000
system analysis	1.000	support test drivers	1.000
design	1.000	jo tech advice to CCB	1.000
conduct design review	1.000	updated MIL STD spec	1.000
prep redlined docs	1.000	jo library control	1.000
updating MIL STD specs	1.000	maintain config procs	1.000
conduct walkthroughs	1.000	updated training manuals	1.000
debugging or modifying	1.000	attend sup walkthrough	1.000
preparing budgets	1.000	conduct sup walkthrough	1.000
ga technical management	1.000	automated build systems	1.000

preparing schedules	1.000	management info reports	1.000
preparing mgmt info rpts	1.000	version descrip docs	1.000
func module design	1.000	version audits	1.000
func system design	1.000	field engineering report	1.000
maint config proc	1.000	support other	1.000
prep temp eng change rpt	1.000	conduct req review	1.000
prep technical rpts	1.000	formulation of policy	1.000
reading tech pubs	1.000	formulation of strategy	1.000
reviewing tech work	1.000	bd technical management	1.000
teaching	1.000	pd program management	1.000
being trained	1.000	configuration management	1.000
documenting test results	1.000	pd quality assurance	1.000
software module testing	1.000	monitoring contracts	1.000
def global data strucs	1.000	other development	1.000
def subsystem interface	1.000	sales marketing	1.000
def stuff for own use	1.000	sup quality assurance	1.000
coding	1.000	sup technical management	1.000
prep test drivers	1.000	SCCB participation	1.000
prep user manuals	1.000	sup program management	1.000
documenting code	1.000	sup policy formulation	1.000
prep sys rqt docs	1.000	conduct support dr	1.000
prep version descr mnls	1.000	sup config management	1.000
defining test cases	1.000	prep version audits	1.000
prep test plans	1.000	ga library control	1.000
hardware testing	1.000	preparing fld eng rpts	1.000
system software test	1.000	formulating strategy	1.000
defining mod test cases	1.000	formulating policy	1.000
prep trouble reports	1.000	contract negotiation	1.000
analyzing trouble rpts	1.000	redlined documentation	1.500
ga tech advice to CCB	1.500	temp ECPs	1.500
pd code	2.000	updated user manuals	1.500
attend design review	2.000	support analysis	2.000
attend walkthroughs	2.000	support test plans	2.000
other admin tasks	2.000	support design	2.000
		attend support dr	2.000
		code patch	2.000
		STRs	2.000

Cluster 5

attend design review	1.000	updating MIL STD specs	1.000
attend walkthroughs	1.000	ga technical management	1.000
support analysis	1.000	interviewing personnel	1.000
support design	1.000	preparing schedules	1.000
attend support dr	1.000	other admin tasks	1.000
updated user manuals	1.000	milestone charts	1.000
STR analyses	1.000	ga tech advice to CCB	1.000
pd code	1.000	preparing mgmt info rpts	1.000
code patch	1.000	interview sheets	2.000
status reports	2.000	analysis reports	2.000
test plans	2.000	cost data	2.000
design specs	2.000	management plans	2.000
prog design language	2.000	contract negotiation	2.000

jo code	2.000	formulating policy	2.000
data flow diagrams	2.000	formulating strategy	2.000
redlined documentation	2.000	preparing budgets	2.000
attend sup walkthrough	2.000	sales marketing	2.000
system analysis	2.000	ga program management	2.000
attend req review	2.000	pd technical management	2.000
STRs	2.000	formulation of strategy	2.000
support test plans	2.000	formulation of policy	2.000
design	2.000	correspondence	2.000
prep technical rpts	2.000	maint config procs	2.000
reading tech pubs	2.000	prep version audits	2.000
reviewing tech work	2.000	ga quality assurance	2.000
teaching	2.000	prep temp eng change rpt	2.000
updating training manual	2.000	hard soft tradeoff eval	2.000
being trained	2.000		
func system design	2.000		
func module design	2.000		
def global data strucs	2.000		
def subsystem interface	2.000		
def stuff for own use	2.000		
coding	2.000		
debugging or modifying	2.000		
prep sys rqt docs	2.000		
prep redlined docs	2.000		
prep version descr mnls	2.000		
prep user manuals	2.000		
documenting code	2.000		
defining test cases	2.000		
prep test drivers	2.000		
prep test plans	2.000		
hardware testing	2.000		
system software test	2.000		
defining mod test cases	2.000		
software module testing	2.000		
documenting test results	2.000		
prep trouble reports	2.000		
analyzing trouble rpts	2.000		

Cluster 38

contract negotiation	1.000	data flow diagrams	1.000
sales marketing	1.000	documenting test results	1.000
prep version audits	1.000	jo code	1.000
ga library control	1.000	prog design language	1.000
correspondence	1.000	requirements specs	1.000
development other	1.000	design specs	1.000
other admin tasks	1.000	test plans	1.000
temp ECPs	1.000	integration plans	1.000
redlined documentation	1.000	documenting code	1.000
support test plans	1.000	status reports	1.000
support test drivers	1.000	debugging or modifying	1.000
jo tech advice to CCB	1.000	reading tech pubs	1.000
udated MIL STD spec	1.000	def stuff for own use	1.000

jo library control	1.000	def subsystem interface	1.000
maintain config procs	1.000	def global data structs	1.000
updated training manuals	1.000	func module design	1.000
STRs	1.000	func system design	1.000
automated build systems	1.000	teaching	1.000
management info reports	1.000	reviewing tech work	1.000
version descrip docs	1.000	system analysis	1.000
version audits	1.000	coding	1.000
field engineering report	1.000	analyzing trouble rpts	1.000
support other	1.000	prep sys rat docs	1.000
preparing fld eng rpts	1.000	prep test drivers	1.000
formulation of policy	1.000	software module testing	1.000
formulation of strategy	1.000	defining mod test cases	1.000
pd program management	1.000	system software test	1.000
configuration management	1.000	prep test plans	1.000
pd quality assurance	1.000	defining test cases	1.000
monitoring contracts	1.000	prep user manuals	1.500
other development	1.000	design	1.500
conduct support dr	1.000	prep technical rpts	1.500
sup policy formulation	1.000	prep redlined docs	1.500
conduct sup walkthrough	1.000	updating training manual	2.000
attend sup walkthrough	1.000	ga technical management	2.000
sup technical management	1.000	attend design review	2.000
sup program management	1.000	milestone charts	2.000
SCCB participation	1.000	hardware testing	2.000
sup config management	1.000	attend req review	2.000
sup quality assurance	1.000	conduct design review	2.000
sup monitoring contracts	1.000	prep version descr mnls	2.000
other support	1.000	pd code	2.000
ga program management	1.000	conduct walkthroughs	2.000
management plans	1.500	being trained	2.000
attend support dr	2.000	test drivers	2.000
cost data	2.000	prep trouble reports	2.000
formulating policy	2.000	preparing schedules	2.000
formulating strategy	2.000		
interviewing personnel	2.000		
conduct req review	2.000		
STR analyses	2.000		
interview sheets	2.000		
analysis reports	2.000		
ga quality assurance	2.000		
prep temp eng change rpt	2.000		

Cluster 57

cost data	1.000	prep version descr mnls	1.000
milestone charts	1.000	prep trouble reports	1.000
analysis reports	1.000	preparing budgets	1.000
hard soft tradeoff eval	1.000	prep temp eng change rpt	1.000
conduct req review	1.000	ga quality assurance	1.000
pd technical management	1.500	being trained	1.000
requirements specs	2.000	updating training manual	1.000
interview sheets	2.000	maint config procs	1.000

correspondence	2.000	prep technical rpts	1.500
development other	2.000	prep user manuals	1.500
temp ECPs	2.000	prep redlined docs	1.500
support test drivers	2.000	STR analyses	2.000
jo tech advice to CCB	2.000	interviewing personnel	2.000
updated MIL STD spec	2.000	func module design	2.000
jo library control	2.000	ga library control	2.000
maintain config procs	2.000	ga tech advice to CCB	2.000
updated training manuals	2.000	other admin tasks	2.000
updated user manuals	2.000	preparing mgmt info rpts	2.000
automated build systems	2.000	reading tech pubs	2.000
management info reports	2.000	def global data strucs	2.000
version descrip docs	2.000	def subsystem interface	2.000
version audits	2.000	def stuff for own use	2.000
field engineering report	2.000	debugging or modifying	2.000
test drivers	2.000	prep sys rqt docs	2.000
integration plans	2.000	func system design	2.000
conduct design review	2.000	teaching	2.000
conduct walkthroughs	2.000	reviewing tech work	2.000
formulation of policy	2.000	documenting code	2.000
support other	2.000	defining test cases	2.000
pd program management	2.000	prep test plans	2.000
configuration management	2.000	system software test	2.000
pd quality assurance	2.000	defining mod test cases	2.000
monitoring contracts	2.000	documenting test results	2.000
other development	2.000	attend walkthroughs	2.000
support analysis	2.000	analyzing trouble rpts	2.000
support design	2.000		
conduct support dr	2.000		
conduct sup walkthrough	2.000		
sup technical management	2.000		
sup policy formulation	2.000		
sup program management	2.000		
SCCB participation	2.000		
sup config management	2.000		
sup quality assurance	2.000		
sup monitoring contracts	2.000		
other support	2.000		
sales marketing	2.000		
preparing fld eng rpts	2.000		
prep version audits	2.000		

These results suggest at least two fundamental distinctions-- management/nonmanagement and development/support/administration. The interpretation of these results indicates that cluster 2 is administrative managers, cluster 3 is developmentally oriented nonmanagement, cluster 5 is support-oriented nonmanagement, cluster 38 is support-oriented management, and

cluster 57 is technical management. There is some overlap between clusters in terms of management or administrative functions; in particular, developmental nonmanagers seem to rate moderately highly on certain management functions. Part of that sort of job probably involves project management activities. The number of individuals assigned to each of the clusters supports this characterization; clusters 3 and 5 contain a little over three-quarters of the sample.

The particular criteria for assigning individuals to jobs may be derived by applying the first ranked values, positive and negative, to a given individual's job description. It is not clear that such job descriptions should be rigorously limited to the disjoint categories defined by the ranked values. It would be more appropriate to develop a match point score based on the job description that would score the pattern of the job against the categories. Thus a given job description might share attributes of several categories (for example, a technical manager who formulates policy and interviews personnel). But it would be classified as technical manager.

One of the implications of the optimality judgment on the number of clusters is that there are no viable subclusters distinguishable in the data. That is, the five categories developed here cannot be subdivided further due to the level of variation among the respondents. This might have to do with the nature of the sampling technique--respondents may not successfully represent the real distribution of job outputs, principal duties, and general activities. A better explanation is that the categories, particularly those of job output and principal duties, represent rather noisy duplication; that is, the large number of different responses contain redundancies that serve to obscure the more important responses. Removing certain of the responses might produce better clusters, but the process would be very involved. Two

clusters--support and development nonmanagement--will be analyzed further below.

Characterization of clusters.--In the rest of this section, some characterizations of the sample in its clusters will be presented. In particular, characterizations of involvement with Ada, professional characteristics, years of involvement, years worked in support and development, language knowledge, and methodology knowledge will be presented.

With respect to involvement with Ada, the following table crosstabulates the clusters with the various types of involvement. This table has no valid chi square statistic due to low cell expectations.

	Ada involvement				
	had orientation to Ada				
	heard of Ada	Ada not known	Ada training	other	
cluster2	0	23	15	5	3
cluster3	4	91	34	24	14
cluster5	8	79	28	12	8
cluster38	0	3	9	1	1
cluster57	0	12	8	3	7

Although no confidence may be put in the result, there is some indication that most of the people who know Ada are technical managers or workers or support workers. The distribution among these clusters is relatively even, with some trend toward the technical people.

The following table presents the clusters versus conference attendance.

	conference attendance	
	yes	no
cluster2	29	16
cluster3	61	105
cluster5	40	94
cluster38	11	3
cluster57	14	16

This table indicates that managers tend to go to conferences more often than nonmanagers, but that nonmanagers go to as many conferences.

The following table presents a crosstabulation of publishing or presentation of papers versus the clusters.

	published or presented		
	yes	no	
cluster2	18	27	45
cluster3	32	134	166
cluster5	22	113	135
cluster38	6	8	14
cluster57	9	21	30
	87	303	390

This table indicates that managers tend to publish more often.

The following table presents a crosstabulation of technical reading versus the clusters; the table has no valid chi square statistic due to low cell expectations.

	technical reading extent				
	only as my job demands occasionally			never other	
	regularly				
cluster2	28	16	1	0	0 45
cluster3	68	78	13	5	1 165
cluster5	51	60	16	7	1 135
cluster38	7	6	1	0	0 14
cluster57	17	12	0	0	0 29
	171	172	31	12	2 388

This table indicates that no manager will admit to not doing any technical reading but that some workers do. Management tends to read more regularly as well.

The following table presents a crosstabulation of technical society membership versus cluster; the chi square statistic is significant at the 95 percent level.

technical societies

	yes	no	
cluster2	19	25	44
cluster3	55	110	165
cluster5	35	99	134
cluster38	6	8	14
cluster57	15	14	29
	130	256	386

According to this table, there isn't much difference between clusters in membership in technical societies. Support oriented workers might tend to belong a bit less than other employees.

The following table presents a crosstabulation of years of involvement versus cluster. This table has no valid chi square statistic due to small cell expectations.

years of involvement

	five to ten years				
	two to five years				
	less than two years	over	ten years		
cluster2	0	2	6	38	46
cluster3	13	47	34	73	167
cluster5	24	46	24	39	133
cluster38	2	0	2	10	14
cluster57	0	1	3	26	30
	39	96	69	186	390

Although not conclusive, this table indicates that managers tend to be experienced people while nonmanagers are less so; still, most people tend to be very experienced.

The following tables crosstabulate years worked in development and support; the support table has no valid chi square statistic due to small cell expectations.

years worked development

three to five years
one to three years
less than a year over five years

cluster2	2	3	5	35	45
cluster3	26	35	37	62	160
cluster5	36	35	17	33	121
cluster38	4	0	0	10	14
cluster57	3	1	3	21	28
	71	74	62	161	368

years worked support

three to five years
one to three years
less than a year over five years

cluster2	7	3	2	14	26
cluster3	31	32	26	35	124
cluster5	44	26	8	12	90
cluster38	2	2	0	6	10
cluster57	4	4	5	6	19
	88	67	41	73	269

Respondents seem fairly well distributed aside from a large lump from cluster 5 being inexperienced in support. The interrelationships between these attributes mentioned in the section on single variables (that is, the various combinations of support and development experience) probably obscure any interesting trends here. Again, the largest fact is that most people, whether in support or development, have more or less extensive development experience and relatively little support experience. This fact is due mainly to an apparent industry concentration on development at the expense of support.

The following table lists the programming languages known versus the clusters; it has no significant chi square statistic due to zero marginals. The table following integrates the lesser-known languages; that table has a chi square value significant only at the 20 percent level.

	language						
	JOVIAL	CMS_2	C	FORTRAN	COBOL	ASSEMBLER	PLI
administrative_manager	17	17	5	46	21	44	23
development_nonmanager	30	53	16	155	89	154	77
support_nonmanager	21	38	8	124	59	112	67
support_manager	5	8	0	13	5	12	6
technical_manager	8	13	4	28	15	28	14
	81	129	33	366	189	350	187

	PASCAL	BASIC	RATFOR	WATFOR	WATFIV	MODULA	SIMULA	XPL
administrative_manager	22	31	15	9	0	1		2
development_nonmanager	85	105	36	43	3	4		5
support_nonmanager	62	83	23	31	2	2		2
support_manager	3	9	3	2	0	0		0
technical_manager	19	24	9	5	1	2		1
	191	252	66	90	6	9		10

	MMP	FORTH	Ada	LISP	NOBOL	ECL	GPSS
administrative_manager	0	2	14	7	7	1	12
development_nonmanager	0	9	41	32	31	1	19
support_nonmanager	0	11	29	18	23	1	18
support_manager	0	0	4	1	2	0	1
technical_manager	0	6	13	5	8	0	5
	0	28	101	63	71	3	55

	SAS	PROTEGE	PPL	APL	Other
administrative_manager	2	0	0	12	4
development_nonmanager	1	0	0	42	43
support_nonmanager	4	1	0	38	26
support_manager	0	0	0	1	5
technical_manager	0	0	0	9	12
	7	1	0	102	90

language

	JOVIAL	CMS_2	FORTRAN	COBOL	ASSEMBLER	PLI	PASCAL
administrative_manager	17	17	46	21	44	23	22
development_nonmanager	30	53	155	89	154	77	85
support_nonmanager	21	38	124	59	112	67	62
support_manager	5	8	13	5	12	6	3
technical_manager	8	13	28	15	28	14	19
	81	129	366	189	350	187	191

	BASIC	Other	
administrative_manager	31	93	314
development_nonmanager	105	326	1074
support_nonmanager	83	237	803
support_manager	9	19	80
technical_manager	24	80	229
	252	755	2500

This table indicates, perhaps, a tendency for management to know more about programming languages than workers, but the rankings are about the same for all clusters. The tendency of managers to be more experienced than workers may explain this greater knowledge.

Since there are a large number of methodologies, a straightforward presentation of counts would not be very informative. Instead, the median polish and ranking of residuals is repeated for methodologies to see how the various clusters are unusual both positively and negatively.

Median Polish of Methodologies against Five Clusters

Effects and Typical after Polishing

	effects
common	33.250
cluster2	0.000
cluster3	71.500
cluster5	31.000
cluster38	-24.500
cluster57	-14.500
PSL PLA	-10.250
SADT	-15.250
SREM	-21.250
HIPO	9.750
Jackson Design	-15.250
Structured Design	11.750
Warnier Orr Design	-14.250
N S Chapin Chart	-12.250
Beamson Tables	-29.250
Program Design language	8.250
Structured Programming	12.750
Structured Walkthroughs	10.750
Top Down Design	12.750

Top Down Testing	11.750
Bottom Up Design	11.750
Bachman Diagramming	-29.250
Entity Diagrams	-30.250
Data Abstraction	-11.250
other methodology	-30.250
enumeration types	-3.250
floating point types	12.750
fixed point types	11.750
user defined types	8.250
pointers	11.750
typed pointers	5.750
records	10.750
ranges	11.750
variant records	3.250
object type dcls	6.250
global variables	11.750
local variables	11.750
formal actual params	10.750
reserved words	11.750
blocks	10.750
case statements	10.250
if then else statements	12.750
loop for while until	12.750
exit statements	12.750
procedures	12.750
functions	12.750
return statements	12.750
clusters modules package	9.750
stubs	8.750
goto statements	11.250
comments	12.750
exception handlers	7.750
task coroutines	8.250
other prog constructs	-33.250
importing exporting name	-15.250
data encapsulation	3.250
name scoping	-9.250
name visibility	-10.250
static dynamic nesting	1.750
iteration	10.250
conditional statements	10.750
recursion	10.750
concurrency	4.250
strong typing	0.250
type conversion	8.750
data abstraction	3.250
generics	-7.250
loop invariants	-3.250
parameter binding	-1.750
version number	4.750
other prog concepts	-33.250
Ada enumeration types	-8.250

Ada user defined types	-0.250
Ada subtypes	-7.250
Ada derived types	-10.250
Ada real types	0.750
Ada float point types	2.750
Ada fixed pt types	1.750
Ada record types	0.250
Ada rec types discrim	-15.250
Ada slices	-21.250
Ada aggregates	-20.250
Ada allocators	-20.250
Ada access types	-13.250
Ada overloading	-18.250
Ada packages	-11.250
Ada private types	-16.250
Ada scope	-10.250
Ada short circuiting	-19.250
Ada visibility	-17.250
Ada tasking	-4.750
Ada task types	-9.250
Ada rendezvous	-15.250
Ada entries	-7.250
Ada entry families	-16.250
Ada separate compilation	-2.750
Ada exceptions	-9.250
Ada generic prog units	-19.250
Ada instantiation	-18.250
Ada elaboration	-18.250
Ada context spec	-14.250
Ada information hiding	-12.250
Ada mutual recursion	-18.250
other Ada concepts	-33.250

Residuals after Polishing

	Warnier Orr Design							
	Jackson Design				N S Chapin Chart			
	PSL	PLA	SADT	SREM	HIPO	Structured Design		
cluster2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	-37.500	-69.500	-66.500	7.500	-49.500	42.500	-51.500	-50.500
cluster5	-8.000	-29.000	-28.000	2.000	-33.000	43.000	-24.000	-33.000
cluster38	5.500	13.500	15.500	-6.500	12.500	-7.500	14.500	7.500
cluster57	3.500	4.500	9.500	-0.500	5.500	-2.500	6.500	3.500

	Structured Programming				Top Down Testing			
	Program Design language		Top Down Design		Bachman Diagramming			
	Beamson Tables		Structured Walkthroughs		Bottom Up Design			
cluster2	0.000	-0.500	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	-69.500	35.000	48.500	41.500	47.500	44.500	33.500	-54.500
cluster5	-33.000	39.500	56.000	36.000	53.000	42.000	28.000	-24.000
cluster38	21.500	-6.000	-7.500	-7.500	-7.500	-6.500	-6.500	22.500
cluster57	12.500	0.000	-1.500	-0.500	-1.500	-1.500	-1.500	13.500
	other methodology				fixed point types			
	Data Abstraction		floating point types		pointers			
	Entity Diagrams		enumeration types		user defined types			
cluster2	0.000	0.000	0.000	0.000	0.000	0.000	-0.500	0.000
cluster3	-54.500	-4.500	-67.500	-8.500	45.500	42.500	32.000	40.500
cluster5	-24.000	-8.000	-32.000	9.000	49.000	48.000	32.500	47.000
cluster38	24.500	11.500	21.500	1.500	-7.500	-6.500	-7.000	-8.500
cluster57	18.500	8.500	13.500	-1.500	-1.500	-1.500	0.000	-3.500
	global variables							
	records		object type dcls		formal actual params			
	typed pointers		variant records		local variables		reserved words	
cluster2	0.000	0.000	-1.500	-0.500	0.000	0.000	0.000	0.000
cluster3	11.500	46.500	10.000	30.000	49.500	49.500	27.500	46.500
cluster5	19.000	47.000	15.500	37.500	55.000	55.000	34.000	42.000
cluster38	-6.500	-7.500	-5.000	-6.000	-7.500	-7.500	-9.500	-8.500
cluster57	-0.500	-0.500	0.000	0.000	-1.500	-2.500	-4.500	-3.500
	loop for while until							
	case statements		exit statements		functions			
	blocks		if then else statements		procedures		return statements	
cluster2	0.000	-0.500	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	37.500	45.000	48.500	47.500	44.500	46.500	48.500	46.500
cluster5	41.000	46.500	54.000	53.000	49.000	50.000	50.000	54.000
cluster38	-10.500	-7.000	-8.500	-8.500	-8.500	-8.500	-7.500	-7.500
cluster57	-4.500	0.000	-1.500	-2.500	-1.500	-2.500	-2.500	-1.500
	goto statements				importing exporting name			
	stubs		exception handlers		other prog constructs			
	clusters modules package		comments		task coroutines			
cluster2	0.000	0.000	-1.500	0.000	0.000	-1.500	0.000	0.000
cluster3	30.500	31.500	48.000	46.500	35.500	35.000	-69.500	-35.500
cluster5	33.000	22.000	53.500	53.000	26.000	14.500	-29.000	-10.000
cluster38	-4.500	-5.500	-7.000	-8.500	-4.500	-7.000	24.500	11.500
cluster57	-1.500	-0.500	0.000	-1.500	-0.500	0.000	16.500	8.500

	name visibility				conditional statements			
	name scoping		iteration		concurrency			
	data encapsulation	static dynamic	nesting	recursion				
cluster2	-1.500	0.000	0.000	0.000	-0.500	0.000	0.000	-1.500
cluster3	0.000	-7.500	-10.500	6.500	43.000	46.500	36.500	24.000
cluster5	2.500	-4.000	-10.000	12.000	38.500	47.000	35.000	15.500
cluster38	-3.000	8.500	7.500	-1.500	-5.000	-6.500	-6.500	0.000
cluster57	0.000	5.500	7.500	-0.500	0.000	-0.500	-1.500	0.000
	other prog concepts							
	type conversion		generics		parameter binding			
	strong typing	data abstraction	loop invariants	version number				
cluster2	1.500	0.000	-2.500	0.000	0.000	2.500	0.000	0.000
cluster3	-7.000	15.500	10.000	-8.500	-6.500	-11.000	19.500	-68.500
cluster5	-2.500	13.000	0.500	-4.000	-3.000	-5.500	12.000	-30.000
cluster38	0.000	-7.500	-2.000	8.500	3.500	0.000	-1.500	25.500
cluster57	2.000	-3.500	0.000	7.500	1.500	2.000	-0.500	14.500
	Ada subtypes				Ada float point types			
	Ada user defined types		Ada real types		Ada record types			
	Ada enumeration types	Ada derived types	Ada fixed pt types					
cluster2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-4.500
cluster3	-14.500	0.500	-10.500	-18.500	7.500	11.500	10.500	3.000
cluster5	-8.000	-1.000	-12.000	-14.000	3.000	6.000	4.000	1.500
cluster38	5.500	2.500	6.500	7.500	-1.500	-0.500	-0.500	0.000
cluster57	4.500	-0.500	5.500	3.500	-0.500	-1.500	-0.500	0.000
	Ada aggregates				Ada private types			
	Ada slices		Ada access types		Ada packages			
	Ada rec types discrim	Ada allocators	Ada overloading					
cluster2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	-19.500	-28.500	-31.500	-30.500	-23.500	-22.500	-14.500	-17.500
cluster5	-11.000	-20.000	-10.000	-12.000	-16.000	-12.000	-10.000	-10.000
cluster38	11.500	17.500	16.500	15.500	10.500	15.500	11.500	13.500
cluster57	9.500	12.500	10.500	12.500	7.500	10.500	7.500	13.500
	Ada entry families							
	Ada short circuiting		Ada tasking		Ada rendezvous			
	Ada scope	Ada visibility	Ada task types	Ada entries				
cluster2	0.000	0.000	0.000	-0.500	0.000	0.000	0.000	0.000
cluster3	-18.500	-41.500	-13.500	0.000	-17.500	-31.500	-24.500	-47.500
cluster5	-10.000	-25.000	-7.000	-2.500	-10.000	-16.000	-18.000	-24.000
cluster38	8.500	14.500	13.500	7.000	8.500	14.500	7.500	13.500
cluster57	7.500	7.500	10.500	3.000	2.500	9.500	2.500	8.500

	Ada generic prog units			Ada context spec				
	Ada exceptions			Ada elaboration		Ada mutual recursion		
	Ada separate compilation	Ada instantiation	Ada information hiding					
cluster2	-0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	3.000	-12.500	-20.500	-38.500	-43.500	-36.500	-18.500	-32.500
cluster5	-2.500	-5.000	-9.000	-15.000	-27.000	-27.000	-11.000	-20.000
cluster38	5.000	8.500	17.500	16.500	13.500	9.500	12.500	14.500
cluster57	0.000	3.500	13.500	9.500	7.500	3.500	6.500	10.500

ranges
other Ada concepts

cluster2	0.000	0.000
cluster3	-71.500	25.500
cluster5	-30.000	33.000
cluster38	24.500	-10.500
cluster57	14.500	-5.500

Categories with High Positive
Residuals in Rank Order

Categories with High Negative
Residuals in Rank Order

Cluster 2

parameter binding	1.000	data abstraction	1.000
strong typing	2.000	concurrency	1.000
		Ada record types	1.000
		Program Design language	2.000
		goto statements	2.000
		user defined types	2.000
		task coroutines	2.000
		data encapsulation	2.000
		iteration	2.000
		case statements	2.000
		object type decls	2.000
		variant records	2.000
		Ada tasking	2.000
		Ada separate compilation	2.000

Cluster 3

HIPO	1.000	PSL PLA	1.000
stubs	1.000	SADT	1.000
Ada record types	1.000	SREM	1.000
Ada fixed pt types	1.000	Jackson Design	1.000
Structured Walkthroughs	1.000	Warnier Orr Design	1.000
Ada float point types	1.000	N S Chapin Chart	1.000
Top Down Testing	1.000	Beamson Tables	1.000
Bottom Up Design	1.000	Bachman Diagramming	1.000
Ada real types	1.000	Entity Diagrams	1.000
concurrency	1.000	Ada entries	1.000
version number	1.000	other methodology	1.000
data abstraction	1.000	enumeration types	1.000

AD-A124 997

ADA* SOFTWARE DESIGN METHODS FORMULATION APPENDICES TO
FINAL REPORT(U) SOFTECH INC WALTHAM MA OCT 82
DAAK80-80-C-0187

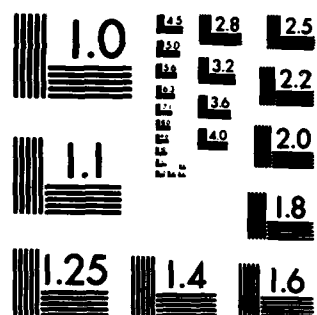
22

UNCLASSIFIED

F/G 9/2

NL

END
DATE
FILMED
3 - B.3
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

type conversion	1.000	other prog constructs	1.000
recursion	1.000	importing exporting name	1.000
iteration	1.000	name scoping	1.000
task coroutines	1.000	name visibility	1.000
reserved words	1.000	strong typing	1.000
exception handlers	1.000	generics	1.000
fixed point types	2.000	loop invariants	1.000
object type dcls	2.000	parameter binding	1.000
local variables	2.000	other prog concepts	1.000
case statements	2.000	Ada rec types discrim	1.000
if then else statements	2.000	Ada enumeration types	1.000
Structured Design	2.000	Ada derived types	1.000
exit statements	2.000	Ada slices	1.000
procedures	2.000	other Ada concepts	1.000
functions	2.000	Ada aggregates	1.000
return statements	2.000	Ada allocators	1.000
clusters modules package	2.000	Ada access types	1.000
loop for while until	2.000	Ada overloading	1.000
goto statements	2.000	Ada packages	1.000
comments	2.000	Ada private types	1.000
blocks	2.000	Ada scope	1.000
formal actual params	2.000	Ada short circuiting	1.000
static dynamic nesting	2.000	Ada visibility	1.000
global variables	2.000	Ada task types	1.000
conditional statements	2.000	Ada rendezvous	1.000
variant records	2.000	Ada entry families	1.000
records	2.000	Ada exceptions	1.000
typed pointers	2.000	Ada generic prog units	1.000
pointers	2.000	Ada instantiation	1.000
user defined types	2.000	Ada elaboration	1.000
Ada user defined types	2.000	Ada context spec	1.000
floating point types	2.000	Ada information hiding	1.000
Top Down Design	2.000	Ada mutual recursion	1.000
Structured Programming	2.000	Data Abstraction	2.000
Program Design language	2.000	Ada subtypes	2.000
Ada separate compilation	2.000		
ranges	2.000		

Cluster 5

ranges	1.000	Ada tasking	1.000
Structured Design	1.000	Data Abstraction	1.000
Program Design language	1.000	Ada subtypes	1.000
Structured Programming	1.000	Ada user defined types	1.000
goto statements	1.000	Ada separate compilation	1.000
Top Down Design	1.000	SADT	2.000
return statements	1.000	Beamson Tables	2.000
functions	1.000	Bachman Diagramming	2.000
enumeration types	1.000	N S Chapin Chart	2.000
floating point types	1.000	Warnier Orr Design	2.000
fixed point types	1.000	other methodology	2.000
user defined types	1.000	PSL PLA	2.000
pointers	1.000	other prog constructs	2.000

typed pointers	1.000	name scoping	2.000
records	1.000	strong typing	2.000
variant records	1.000	importing exporting name	2.000
object type decls	1.000	generics	2.000
global variables	1.000	loop invariants	2.000
local variables	1.000	parameter binding	2.000
formal actual params	1.000	other prog concepts	2.000
static dynamic nesting	1.000	Ada enumeration types	2.000
blocks	1.000	Jackson Design	2.000
case statements	1.000	SREM	2.000
if then else statements	1.000	Ada derived types	2.000
loop for while until	1.000	Ada rec types discrim	2.000
exit statements	1.000	Ada slices	2.000
procedures	1.000	Ada aggregates	2.000
clusters modules package	1.000	Ada allocators	2.000
comments	1.000	Ada access types	2.000
conditional statements	1.000	Ada overloading	2.000
data encapsulation	1.000	Ada packages	2.000
Bottom Up Design	2.000	Ada private types	2.000
Top Down Testing	2.000	Ada scope	2.000
stubs	2.000	Ada short circuiting	2.000
Structured Walkthroughs	2.000	Ada visibility	2.000
reserved words	2.000	name visibility	2.000
exception handlers	2.000	Ada task types	2.000
iteration	2.000	Ada rendezvous	2.000
recursion	2.000	Ada entries	2.000
task coroutines	2.000	Ada entry families	2.000
concurrency	2.000	Entity Diagrams	2.000
type conversion	2.000	Ada exceptions	2.000
data abstraction	2.000	Ada generic prog units	2.000
version number	2.000	Ada instantiation	2.000
Ada real types	2.000	Ada elaboration	2.000
Ada float point types	2.000	Ada context spec	2.000
Ada fixed pt types	2.000	Ada information hiding	2.000
Ada record types	2.000	Ada mutual recursion	2.000
HIPO	2.000	other Ada concepts	2.000

Cluster 38

PSL PLA	1.000	HIPO	1.000
SADT	1.000	Structured Design	1.000
SREM	1.000	Program Design language	1.000
Jackson Design	1.000	Structured Programming	1.000
Warnier Orr Design	1.000	Structured Walkthroughs	1.000
N S Chapin Chart	1.000	Top Down Design	1.000
Beamson Tables	1.000	Top Down Testing	1.000
Bachman Diagramming	1.000	Bottom Up Design	1.000
Entity Diagrams	1.000	floating point types	1.000
Data Abstraction	1.000	fixed point types	1.000
other methodology	1.000	user defined types	1.000
Ada rendezvous	1.000	pointers	1.000
other prog constructs	1.000	typed pointers	1.000
importing exporting name	1.000	records	1.000

name scoping	1.000	variant records	1.000
Ada short circuiting	1.000	object type decls	1.000
generics	1.000	global variables	1.000
loop invariants	1.000	local variables	1.000
other prog concepts	1.000	formal actual params	1.000
Ada enumeration types	1.000	reserved words	1.000
Ada user defined types	1.000	blocks	1.000
Ada subtypes	1.000	case statements	1.000
Ada derived types	1.000	if then else statements	1.000
Ada rec types discrim	1.000	loop for while until	1.000
Ada slices	1.000	exit statements	1.000
Ada aggregates	1.000	procedures	1.000
Ada allocators	1.000	functions	1.000
Ada access types	1.000	return statements	1.000
Ada overloading	1.000	clusters modules package	1.000
Ada packages	1.000	stubs	1.000
Ada instantiation	1.000	goto statements	1.000
Ada scope	1.000	comments	1.000
Ada exceptions	1.000	exception handlers	1.000
Ada visibility	1.000	task coroutines	1.000
Ada tasking	1.000	data encapsulation	1.000
Ada task types	1.000	static dynamic nesting	1.000
Ada entries	1.000	iteration	1.000
Ada entry families	1.000	conditional statements	1.000
Ada separate compilation	1.000	recursion	1.000
Ada generic prog units	1.000	type conversion	1.000
Ada elaboration	1.000	version number	1.000
Ada context spec	1.000	Ada real types	1.000
Ada information hiding	1.000	ranges	1.000
Ada mutual recursion	1.000	Ada fixed pt types	1.500
other Ada concepts	1.000	data abstraction	2.000
name visibility	1.500	Ada float point types	2.000
Ada private types	1.500		
enumeration types	2.000		

Cluster 57

strong typing	1.000	Ada float point types	1.000
Ada private types	1.500	Ada fixed pt types	1.500
name visibility	1.500	HIPO	2.000
PSL PLA	2.000	Structured Walkthroughs	2.000
Jackson Design	2.000	Top Down Design	2.000
N S Chapin Chart	2.000	Top Down Testing	2.000
Beamsom Tables	2.000	Bottom Up Design	2.000
SREM	2.000	enumeration types	2.000
SADT	2.000	floating point types	2.000
Data Abstraction	2.000	fixed point types	2.000
other methodology	2.000	pointers	2.000
other prog constructs	2.000	typed pointers	2.000
importing exporting name	2.000	records	2.000
name scoping	2.000	global variables	2.000
Entity Diagrams	2.000	local variables	2.000
Bachman Diagramming	2.000	formal actual params	2.000

generics	2.000	reserved words	2.000
loop invariants	2.000	blocks	2.000
Warnier Orr Design	2.000	if then else statements	2.000
other prog concepts	2.000	loop for while until	2.000
Ada enumeration types	2.000	exit statements	2.000
Ada subtypes	2.000	procedures	2.000
Ada derived types	2.000	functions	2.000
Ada rec types discrim	2.000	return statements	2.000
Ada slices	2.000	clusters modules package	2.000
Ada aggregates	2.000	stubs	2.000
Ada allocators	2.000	comments	2.000
Ada access types	2.000	exception handlers	2.000
Ada overloading	2.000	static dynamic nesting	2.000
Ada packages	2.000	conditional statements	2.000
parameter binding	2.000	recursion	2.000
Ada scope	2.000	type conversion	2.000
Ada short circuiting	2.000	version number	2.000
Ada visibility	2.000	Ada user defined types	2.000
Ada tasking	2.000	Ada real types	2.000
Ada task types	2.000	Structured Programming	2.000
Ada rendezvous	2.000	Structured Design	2.000
Ada entries	2.000	ranges	2.000
Ada entry families	2.000		
Ada exceptions	2.000		
Ada generic prog units	2.000		
Ada instantiation	2.000		
Ada elaboration	2.000		
Ada context spec	2.000		
Ada information hiding	2.000		
Ada mutual recursion	2.000		
other Ada concepts	2.000		

The administrative managers (cluster 2) aren't very strong on anything except parameter binding, probably a random outlier. They aren't very weak on anything, either, but with some tendency against technical programming.

The developmentally oriented nonmanagers (cluster 3) rate very strongly on design methodologies, some Ada concepts, and various programming constructs and concepts. They are weak on some of the design methodologies, certain programming concepts, and most Ada concepts.

The support-oriented nonmanagers (cluster 5) are strong on standard design methodologies and standard programming constructs and are somewhat less strong on various programming concepts. They are weak on Ada concepts and

certain programming methodologies and constructs. They seem to have the basics but are not theoretically sophisticated.

The support-oriented managers are strong on certain design methodologies, programming concepts, and Ada concepts. They are weak on other design methodologies and programming constructs. These people would seem to be somewhat eccentric. Perhaps these are the type of people who tend to gravitate to the "less practical"--that is, less technical/developmental--side of the industry.

The technical managers tend to be relatively strong on certain programming concepts, many design methodologies, and several Ada concepts, but weak on other methodologies and on programming constructs.

In summary, the characterizations of the clusters support the classifications in large part. There are few startling departures from common sense.

Current titles.--a table showing current job titles and positions in the management hierarchy for the clusters indicates the relationship between the way the industry currently defines jobs and the classification scheme developed in this report.

Respondent	Job Title	Position in Hierarchy
administrative manager		

1003	Chief, CAD/CAM Systems	middle mgmt
1005	Staff Specialist	technical
1011	Project Engineer	technical
1013	Software Design Specialist	technical
1018	Engineering Software Supervisor	middle mgmt
1020	Software Design Specialist	technical
1023	Software Engineering Specialist	technical
1025	Engineering Specialist	technical
1039	Engineering Specialist	technical
1042	Software Design Specialist	technical
1054	Software Engineering Specialist	technical
1059	Supervisor	middle mgmt
1062	Supervisor	middle mgmt
1063	Chief, Product Software	middle mgmt
1066	Software Design Specialist	technical

1085	Software Engineering Specialist	technical
1086	Senior Software Engineer	technical
1093	Engineering Chief	middle mgmt
2005	Supervisor	middle mgmt
2006	Software Development Manager	middle mgmt
2012	Senior System Analyst	technical
3002	Software Management	high level mgmt
3009	Software Manager	middle mgmt
3013	Engineering Specialist	technical
3026	R&D Engineer	technical
3029	Digital Signal Processing	*
3058	Engineering Specialist	technical
3060	*	*
3061	Senior Engineering Specialist	technical
3070	Engineering Specialist	technical
3097	Engineering Specialist	technical
4003	Consultant	technical
4026	Consultant	technical
4029	Consultant	technical
4038	Software Group Manager	middle mgmt
4044	Programmer Analyst	technical
4047	Unit Manager	middle mgmt
4050	Consultant	technical
4052	CAD Development Manager	middle mgmt
4072	Consultant	technical
5012	*	*
5029	*	*
5030	*	*
5034	*	*
5076	*	*
9001	*	*

development nonmanager

1001	Consultant	technical
1004	Engineering Software Supervisor	middle mgmt
1012	Software Design Specialist	technical
1014	Senior Software Engineer	technical
1016	Software Engineer	technical
1029	Software Design Specialist	technical
1031	Software Design Specialist	technical
1052	Senior Engineering Specialist	technical
1060	Software Engineer	technical
1064	*	*
1072	*	*
1074	Software Design Specialist	technical
1075	Software Design Specialist	technical
1078	Senior Engineer	technical
1079	Senior Software Engineer	technical
1080	Senior Software Engineer	technical
1082	Senior Software Engineer	technical
1084	Software Design Specialist	technical
1090	System Engineering Specialist	technical

1091	System Design Specialist	technical
1094	Software Engineer	technical
1100	Engineer	technical
2001	Senior Programmer	technical
2002	Programmer	technical
2007	Senior Scientific Programmer	technical
2009	Project Supervisor	middle mgmt
2011	Data Processing Consultant	technical
2013	Principle Scientific Programmer	technical
2014	Supervising System Analyst	middle mgmt
2016	Scientific Programmer	technical
2018	Principle Scientific Programmer	technical
2020	Principle Scientific Programmer	technical
3001	Junior Programmer	entry level
3004	Research Engineer	technical
3005	Senior Software Engineer	technical
3006	Supervisor	middle mgmt
3007	Advanced R&D Engineer	technical
3010	Communications Software	technical
3011	Software R&D Engineer	technical
3015	Engineer	technical
3018	Senior Engineer	technical
3023	*	*
3024	R&D Engineer	technical
3025	Senior Engineer	technical
3028	Digital Signal Processing	technical
3030	*	*
3031	R&D Engineer	technical
3032	Engineer	technical
3042	R&D Engineer	technical
3049	Junior Programmer	entry level
3051	Advanced Development Engineer	technical
3052	Engineering Specialist	technical
3054	*	*
3062	*	*
3063	Programmer Analyst	technical
3065	Programmer	technical
3068	R&D Engineer	technical
3069	Senior Engineer	technical
3072	Software Engineer	technical
3074	Engineer	technical
3077	Senior Software Engineer	technical
3078	R&D Engineer	technical
3080	Senior Engineer	technical
3085	Junior Engineer	technical
3086	*	*
3087	R&D Engineer	technical
3089	Research Engineer	technical
3091	Software Engineer	technical
3093	Engineering Specialist	technical
3094	Software Engineer	technical
3109	R&D Engineer	technical
3111	Senior Engineer	technical

3112	Consultant	technical
3113	Software Engineer	technical
3114	Software R&D Engineer	technical
3117	Engineering Specialist	technical
3119	R&D Engineer	technical
3120	Programmer	technical
3122	*	*
4002	Programmer	technical
4004	Programmer Analyst	technical
4005	Programmer Analyst	technical
4006	Senior Applications Analyst	technical
4007	Staff Consultant	technical
4008	*	*
4015	Senior Programmer Analyst	technical
4016	Senior Programmer Analyst	technical
4023	Senior System Analyst	technical
4024	Programmer Analyst	technical
4027	Senior GS Analyst	technical
4028	Principle Programmer Analyst	technical
4032	Analyst	technical
4034	Government Program Analyst	technical
4035	Government Program Analyst	technical
4036	Principle Programmer Analyst	technical
4039	Consultant	technical
4042	Principle Systems Analyst	technical
4043	*	*
4045	Senior Analyst	technical
4048	Principle Programmer Analyst	technical
4051	Firmware Design Engineer	technical
4056	System Analyst	technical
4058	*	*
4060	Principle Engineer	technical
4062	Senior Programmer Analyst	technical
4063	*	*
4065	Programmer	technical
4067	Programmer Analyst	technical
4068	Consultant	technical
4069	Associate Programmer Analyst	entry level
4070	Programmer	technical
4073	Programmer Analyst	technical
4075	Senior Engineer	technical
4076	Quality Assurance Engineer	technical
5001	*	*
5002	*	*
5004	*	*
5005	*	*
5007	*	*
5008	Diagnostic Software	technical
5011	*	*
5013	*	*
5016	*	*
5021	*	*
5024	*	*

5026	*	*
5031	*	*
5032	*	*
5036	*	*
5037	*	*
5038	*	*
5043	*	*
5045	*	*
5047	*	*
5048	*	*
5049	*	*
5052	*	*
5053	*	*
5057	*	*
5060	*	*
5063	*	*
5067	*	*
5068	*	*
5069	*	*
5070	*	*
5071	*	*
5072	*	*
5074	Programmer	technical
5078	*	*
5080	*	*
6004	Project Leader	middle mgmt
6005	*	*
6007	Computer Scientist	technical
6009	Electronic Technician	entry level
6010	Electrical Engineer	technical
6011	Electrical Engineer	technical
7001	Computer Specialist	technical
7004	Computer Specialist	technical
7005	Computer Specialist	technical
7006	Computer Specialist	technical
7008	Computer Specialist	technical
7010	Computer Specialist	technical
7011	Computer Specialist	technical
8004	Team Leader	middle mgmt
9004	Electrical Engineer	technical
9005	Electrical Engineer	technical
9006	Electrical Engineer	technical

support nonmanager

1007	Senior Software Engineer	technical
1009	Software Engineer	technical
1010	Software Engineer	technical
1015	Software Engineer	technical
1017	Software Engineer	technical
1026	Software Design Specialist	technical
1038	*	*
1040	Software Design Specialist	technical

1041	Software Design Specialist	technical
1043	Senior Dynamics Engineer	technical
1044	Software Engineer	technical
1045	Software Engineer	technical
1046	Software Engineer	technical
1048	Software Engineering Specialist	technical
1049	Software Design Specialist	technical
1050	Software Design Specialist	technical
1053	Software System Engineer	technical
1055	*	*
1057	Principle Engineer	technical
1058	Associate Engineer	entry level
1065	Engineering Manager	high level mgmt
1076	Software Engineer	technical
1077	Electrical Engineer	technical
1087	Software Engineering Specialist	technical
1096	Software Design Specialist	technical
2003	Scientific Programmer	technical
2017	Senior System Analyst	technical
2023	Associate Scientific Programmer	entry level
3003	Software Engineer	technical
3008	Programming Aide	entry level
3012	Senior Engineer	technical
3014	Software Engineer	technical
3020	R&D Engineer	technical
3027	Engineering Specialist	technical
3033	Senior Engineer	technical
3035	Senior Engineer	technical
3036	*	*
3037	Software Engineer	technical
3038	Software Engineer	technical
3039	*	*
3043	Senior Engineer	technical
3044	Software Engineer	technical
3045	*	*
3046	Software Engineer	technical
3047	*	*
3048	Software Engineer	technical
3050	R&D Engineer	technical
3053	Senior Software Engineer	technical
3059	Senior Engineer	technical
3064	Software Engineer	technical
3066	Senior Software Engineer	technical
3067	Software Engineer	technical
3071	Senior System Engineer	technical
3073	Software Engineer	technical
3075	Software Engineer	technical
3076	*	*
3081	Software Engineer	technical
3082	Software Engineer	technical
3084	Software Engineer	technical
3088	Senior Engineering Specialist	technical
3092	Senior Engineer	technical

3099	Advanced Research Engineer	technical
3104	Software Engineer	technical
3106	*	*
3107	Engineer	technical
3110	Senior Software Engineer	technical
3115	Software Engineer	technical
3116	Engineer	technical
3118	Engineer	technical
3121	Software Engineer	technical
4001	System Engineer	technical
4009	Consultant	technical
4012	Principle Programmer Analyst	technical
4013	Software System Programmer	technical
4018	Electrical Engineer	technical
4019	Analyst	technical
4021	Associate Applications Analyst	entry level
4022	Programmer	technical
4031	Principle Programmer	technical
4040	System Programming Analyst	technical
4053	Programmer Analyst	technical
4054	Programmer Analyst	technical
4057	Programmer	technical
4059	Programmer Analyst	technical
4064	Manager, Software Development	middle mgmt
4071	Programmer	technical
4074	*	*
4077	Engineer	technical
5003	*	*
5006	*	*
5009	*	*
5010	*	*
5014	*	*
5015	*	*
5019	*	*
5020	*	*
5022	*	*
5025	*	*
5027	*	*
5028	*	*
5033	*	*
5039	*	*
5040	*	*
5041	Programmer	technical
5042	*	*
5044	Programmer	technical
5046	*	*
5050	*	*
5054	*	*
5055	*	*
5056	*	*
5059	*	*
5062	*	*
5064	*	*

5065	*	*
5066	*	*
5073	*	*
5075	*	*
5077	*	*
5079	*	*
6003	General Engineer	technical
6006	Software Quality Assurance	technical
6008	Computer Scientist	technical
7002	Computer Specialist	technical
7003	Programmer	technical
7007	Computer Specialist	technical
7012	Computer Specialist	technical
7013	Computer Specialist	technical
7014	Computer Specialist	technical
7015	Computer Specialist	technical
8001	Computer Systems Analyst	technical
8002	Software System Analyst	technical
8003	Associate Programmer Analyst	entry level
8005	Computer Specialist	technical
9003	Mathematician	technical

support manager

1056	Software Quality Assurance	technical
1061	Software Design Specialist	technical
1083	Software Design Specialist	technical
1092	Software Design Specialist	technical
2019	Project Engineer	technical
3040	*	*
3057	Software Management	high level mgmt
4010	Manager	middle mgmt
4017	Line Manager	middle mgmt
4020	Manager	middle mgmt
4046	Software Development Manager	middle mgmt
4061	Manager	middle mgmt
5035	*	*
6002	General Engineer	technical

technical manager

1081	*	*
1088	Software Design Specialist	technical
1095	Senior Programmer	technical
1097	Software Design Specialist	technical
1098	Software Design Specialist	technical
1099	Principle Engineer	technical
2004	Project Manager	middle mgmt
2010	Data Processing Consultant	technical
2015	Senior System Analyst	technical
3034	Advanced R&D Engineer	technical
3055	*	*
3056	R&D Engineer	technical

3083	Advanced R&D Engineer	technical
3090	Engineering Manager	high level mgmt
3095	Engineering Specialist	technical
3102	Software Engineer	technical
3103	Engineering Specialist	technical
4011	System Analyst	technical
4030	Consultant	technical
4037	Programmer	technical
4041	Consultant	technical
4049	Principle Programmer Analyst	technical
4055	Senior Consultant	technical
4066	Programmer Analyst	technical
5017	*	*
5018	*	*
5023	*	*
5051	*	*
5058	*	*
7009	Computer Specialist	technical

Finally, the following table presents the clusters versus the nine companies, showing the internal distributions of job categories in those companies. The distribution seems consistent with the interpretation of the clusters aside from the total lack of managers in the public-sector companies.

	company								
	company three			company six			company nine		
	company two		company five		company eight				
	company one	company four	company seven						
administrative manager	18	3	10	9	5	0	0	0	1
development nonmanager	22	10	47	35	36	6	7	1	3
support nonmanager	25	3	42	18	32	3	7	4	1
support manager	4	1	2	5	1	1	0	0	0
technical manager	6	3	8	7	5	0	1	0	0

Further classification of nonmanagement.--Although the above classification makes sense, it doesn't have very much to say about those people who will be most involved with Ada in the future, the programmer, the analyst, the support personnel, and so on. Given that these two clusters--support and development nonmanagement--are of such great interest, it may be advisable to cluster these individuals separately.

In order to get a clearer picture, but at the risk of losing information, the following clustering was done only on the principal duties and primary general activities of the two nonmanagement clusters. The results below show some distinct subcategories in these clusters. It should be borne in mind, however, that these clusters are probably less valid than the first set.

After clustering development employees (see Appendix 3 for the tree), five distinct clusters emerged based on median polish residual ranks. These clusters have the following populations:

	Count	Percent
cluster1	9	5.4
cluster3	17	10.1
cluster4	131	78.0
cluster7	6	3.6
cluster153	5	3.0

The following median polish summarizes the properties of these clusters.

Analysis of Clustering Based on Primary General Activities and Principal Duties of Development Employees

Effects and Typical after Polishing

	effects
common	2.000
cluster1	1.000
cluster3	0.000
cluster4	10.000
cluster7	0.000
cluster153	-1.000
ga program management	-2.000
ga sales marketing	-2.000
ga contract negotiation	-2.000
ga formulating policy	-1.000
ga formulating strategy	-2.000
ga preparing budgets	0.000
ga technical management	1.000
ga interv personnel	-1.000
ga preparing schedules	1.000
ga prep mgmt info rpts	-2.000
ga prep fid eng rpts	-2.000

ga other admin tasks	-2.000
ga tech advice to CCB	-1.000
ga maint config procs	-2.000
ga library control	-2.000
ga prep version audits	-2.000
ga quality assurance	-2.000
ga prep temp eng rpt	-2.000
ga prep technical rpts	-1.000
ga reading tech pubs	0.000
ga reviewing tech work	3.000
ga teaching	0.000
ga updat training man	-1.000
ga being trained	-1.000
ga func system design	4.000
ga func module design	5.000
ga def glob data strucs	3.000
ga def subsys interface	3.000
ga def stuff for own use	5.000
ga coding	5.000
ga debug or modifying	5.000
ga prep sys rqt docs	1.000
ga updat MIL STD specs	-1.000
ga prep redlined docs	1.000
ga prep vers descr mnls	-1.000
ga prep user manuals	0.000
ga documenting code	4.000
ga defining test cases	1.000
ga prep test drivers	0.000
ga prep test plans	2.000
ga hardware testing	-2.000
ga system software test	3.000
ga def mod test cases	5.000
ga software module test	5.000
ga doc test results	0.000
ga prep trouble reports	2.000
ga anal trouble rpts	1.000
pd conduct req review	1.000
pd attend req review	4.000
pd system analysis	4.000
pd design	4.000
pd conduct design review	3.000
pd attend design review	5.000
pd code	5.000
pd conduct walkthroughs	4.000
pd attend walkthroughs	4.000
pd formulation of policy	0.000
pd formulating strategy	0.000
pd technical management	2.000
pd program management	-2.000
pd configuration mngmnt	-2.000
pd quality assurance	-1.000
pd monitoring contracts	-2.000
pd other development	-2.000

pd support analysis	5.000
pd support design	3.000
pd conduct support or	1.000
pd attend support dr	2.000
pd code patch	6.000
pd conduct sup walkthru	-1.000
pd attend sup walkthru	-1.000
pd sup technical mgmt	1.000
pd sup formulate policy	-1.000
pd sup program mgmt	-2.000
pd SCCB participation	1.000
pd sup config management	-2.000
pd sup quality assurance	-1.000
pd sup monitor contracts	-2.000
pd other support	-2.000

Residuals after Polishing

	ga contract negotiation	ga preparing budgets	ga sales marketing	ga formulating strategy	ga interv personnel	ga program management	ga formulating policy	ga technical management
cluster1	0.000	1.000	0.000	1.000	0.000	0.000	-3.000	0.000
cluster3	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000
cluster4	-7.000	-10.000	-10.000	-8.000	-1.000	-4.000	16.000	-8.000
cluster7	2.000	0.000	0.000	0.000	2.000	1.000	2.000	0.000
cluster153	1.000	1.000	1.000	1.000	2.000	-1.000	-1.000	2.000

	ga prep fld eng rpts	ga maint config procs	ga prep mgmt info rpts	ga tech advice to CCB	ga preparing schedules	ga other admin tasks	ga library control
cluster1	-4.000	0.000	-1.000	0.000	0.000	0.000	-1.000
cluster3	0.000	0.000	0.000	1.000	-1.000	0.000	0.000
cluster4	3.000	0.000	-9.000	-10.000	-2.000	-7.000	-8.000
cluster7	0.000	3.000	0.000	0.000	1.000	0.000	0.000
cluster153	-1.000	1.000	1.000	3.000	2.000	3.000	2.000

	ga prep technical rpts	ga updat training man	ga prep temp eng rpt	ga reviewing tech work	ga quality assurance	ga reading tech pubs	ga teaching	ga being trained
cluster1	-1.000	2.000	0.000	1.000	-3.000	-1.000	0.000	2.000
cluster3	0.000	0.000	0.000	0.000	-1.000	1.000	0.000	1.000
cluster4	-4.000	-1.000	0.000	0.000	10.000	-5.000	-8.000	-2.000
cluster7	2.000	0.000	2.000	0.000	0.000	0.000	-1.000	0.000
cluster153	4.000	2.000	0.000	2.000	0.000	0.000	0.000	0.000

	ga prep sys rqt docs							
	ga def glob data strucs				ga coding			
	ga func module design ga def stuff for own use							
	ga func system design ga def subsys interface				ga debug or modifying			
cluster1	-1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	4.000	6.000	7.000	3.000	4.000	6.000	6.000	-1.000
cluster4	56.000	73.000	43.000	56.000	58.000	61.000	62.000	20.000
cluster7	0.000	-4.000	0.000	-1.000	-1.000	-5.000	-5.000	1.000
cluster153	-2.000	-6.000	-4.000	-3.000	-6.000	-6.000	-6.000	-1.000
	ga prep vers descr mnls ga defining test cases							
	ga prep redlined docs				ga documenting code		ga prep test plans	
	ga updat MIL STD specs		ga prep user manuals			ga prep test drivers		
cluster1	2.000	-1.000	2.000	2.000	2.000	3.000	3.000	1.000
cluster3	0.000	-1.000	-1.000	-2.000	0.000	-1.000	-1.000	0.000
cluster4	-5.000	0.000	-4.000	4.000	49.000	25.000	17.000	29.000
cluster7	0.000	0.000	0.000	0.000	-5.000	0.000	0.000	-1.000
cluster153	1.000	1.000	2.000	0.000	-3.000	-2.000	-1.000	-1.000
	ga def mod test cases ga prep trouble reports							
	ga system software test				ga doc test results pd conduct req review			
	ga hardware testing		ga software module test			ga anal trouble rpt's		
cluster1	2.000	0.000	0.000	0.000	5.000	1.000	3.000	-2.000
cluster3	0.000	5.000	3.000	6.000	-1.000	0.000	0.000	2.000
cluster4	0.000	55.000	32.000	47.000	15.000	13.000	19.000	16.000
cluster7	0.000	-2.000	-7.000	-7.000	0.000	-3.000	-2.000	0.000
cluster153	2.000	-3.000	-5.000	-5.000	0.000	-1.000	0.000	-1.000
	pd attend walkthroughs							
	pd conduct design review				pd conduct walkthroghs			
	pd system analysis		pd attend design review					
	pd attend req review		pd design		pd code			
cluster1	-3.000	-2.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster3	4.000	4.000	11.000	10.000	9.000	9.000	9.000	10.000
cluster4	35.000	56.000	87.000	39.000	47.000	78.000	28.000	31.000
cluster7	-2.000	-1.000	-2.000	-3.000	-4.000	-4.000	-5.000	-4.000
cluster153	0.000	0.000	0.000	-2.000	-3.000	-6.000	-4.000	-4.000
	pd other development							
	pd technical management				pd quality assurance			
	pd formulating strategy				pd configuration mngmnt			
	pd formulation of policy		pd program management			pd monitoring contracts		
cluster1	-2.000	-3.000	-4.000	-1.000	0.000	-2.000	-1.000	-1.000
cluster3	0.000	1.000	0.000	0.000	1.000	0.000	0.000	0.000
cluster4	-8.000	0.000	11.000	-5.000	-3.000	-3.000	-4.000	-9.000
cluster7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster153	1.000	1.000	1.000	4.000	4.000	5.000	6.000	1.000

	pd conduct support dr				pd conduct sup walkthru			
	pd support design				pd code patch			
	pd support analysis				pd attend support dr			
					pd attend sup walkthru			
cluster1	0.000	0.000	4.000	4.000	0.000	6.000	7.000	-3.000
cluster3	5.000	11.000	12.000	10.000	6.000	14.000	13.000	2.000
cluster4	10.000	2.000	-8.000	0.000	0.000	-8.000	-1.000	-5.000
cluster7	-1.000	0.000	-3.000	-4.000	-6.000	-1.000	0.000	0.000
cluster153	-1.000	-1.000	0.000	0.000	-6.000	0.000	0.000	2.000

	pd sup config management				pd other support			
	pd SCCB participation				pd sup monitor contracts			
	pd sup program mgmt				pd sup quality assurance			
	pd sup formulate policy							
cluster1	0.000	-1.000	-2.000	0.000	-1.000	-1.000	-1.000	
cluster3	0.000	0.000	0.000	0.000	1.000	0.000	0.000	
cluster4	-8.000	-10.000	0.000	-6.000	-4.000	-5.000	-5.000	
cluster7	0.000	1.000	-3.000	0.000	0.000	0.000	0.000	
cluster153	2.000	3.000	2.000	4.000	5.000	5.000	1.000	

Categories with High Positive
Residuals in Rank Order

Cluster 1

ga being trained	1.000	pd conduct req review	1.000
ga updat MIL STD specs	1.000	pd attend req review	1.000
ga prep temp eng rpt	1.500	pd system analysis	1.000
ga sales marketing	1.500	ga technical management	1.000
ga hardware testing	1.500	pd formulating strategy	1.000
ga prep vers descr mnls	1.500	ga preparing schedules	1.000
ga formulating policy	1.500	pd technical management	1.000
ga prep test drivers	2.000	ga reviewing tech work	1.000
ga prep trouble reports	2.000	ga prep redlined docs	1.500
ga reading tech pubs	2.000	ga prep fld eng rpts	2.000
ga updat training man	2.000	pd other development	2.000
ga anal trouble rpts	2.000	pd monitoring contracts	2.000
pd conduct sup walkthru	2.000	pd quality assurance	2.000
pd attend support dr	2.000	ga prep mgmt info rpts	2.000
ga prep user manuals	2.000	pd SCCB participation	2.000
ga doc test results	2.000	pd formulation of policy	2.000
pd conduct support dr	2.000	pd sup quality assurance	2.000
pd attend sup walkthru	2.000	pd sup monitor contracts	2.000
ga documenting code	2.000	pd other support	2.000
ga defining test cases	2.000	ga library control	2.000
ga prep test plans	2.000	pd program management	2.000
ga preparing budgets	2.500	ga prep version audits	2.000
pd code patch	2.500	ga quality assurance	2.000
ga prep sys rqt docs	3.000	ga teaching	2.000
ga coding	3.000	pd sup program mgmt	2.000
ga system software test	3.000	pd sup technical mgmt	2.000
ga def stuff for own use	3.000	ga func system design	2.000

ga contract negotiation	3.000	ga formulating strategy	2.500
ga debug or modifying	3.000	pd configuration mngmnt	2.500
ga def mod test cases	3.000	pd support design	2.500
pd conduct design review	3.000	pd design	2.500
ga software module test	3.000	ga def glob data strucs	2.500
pd attend design review	3.000	ga other admin tasks	2.500
pd code	3.000	ga prep technical rpts	2.500
pd conduct walkthroughs	3.000	ga program management	2.500
pd attend walkthroughs	3.000	ga system software test	3.000
pd support analysis	3.000	pd attend design review	3.000
ga def subsys interface	3.000	ga software module test	3.000
ga func module design	3.000	pd conduct walkthroughs	3.000
ga maint config procs	3.000	ga prep sys rqt docs	3.000
ga tech advice to CCB	3.000	ga coding	3.000
ga interv personnel	3.000	pd attend walkthroughs	3.000
pd sup formulate policy	3.000	ga maint config procs	3.000
pd sup config management	3.000	ga def subsys interface	3.000
		ga tech advice to CCB	3.000
		pd support analysis	3.000
		pd code	3.000
		pd conduct design review	3.000
		ga func module design	3.000
		pd sup formulate policy	3.000
		ga def mod test cases	3.000
		ga interv personnel	3.000
		pd sup config management	3.000
		ga debug or modifying	3.000
		ga contract negotiation	3.000
		ga def stuff for own use	3.000

Cluster 3

pd support design	1.000	ga doc test results	1.000
ga teaching	1.000	ga prep user manuals	1.000
pd conduct support dr	1.000	ga prep test drivers	1.500
pd attend support dr	1.000	ga prep sys rqt docs	1.500
pd code patch	1.000	ga prep redlined docs	1.500
pd attend sup walkthru	1.000	ga formulating policy	2.000
pd conduct sup walkthru	1.000	ga reviewing tech work	2.000
pd sup technical mgmt	1.500	ga hardware testing	2.000
pd formulating strategy	1.500	ga defining test cases	2.000
ga other admin tasks	2.000	ga prep mgmt info rpts	2.000
pd attend walkthroughs	2.000	ga reading tech pubs	2.000
pd sup quality assurance	2.000	ga tech advice to CCB	2.000
ga updat training man	2.000	ga prep vers descr mnls	2.000
ga being trained	2.000	ga prep temp eng rpt	2.500
ga func system desig	2.000	ga prep technical rpts	2.500
ga func module design	2.000	ga anal trouble rpts	2.500
ga def glob data strucs	2.000	pd technical management	2.500
ga def subsys interface	2.000	ga updat MIL STD specs	2.500
ga def stuff for own use	2.000	ga formulating strategy	2.500
ga coding	2.000	ga sales marketing	2.500
ga debug or modifying	2.000	ga program management	2.500

pd configuration mngmnt	2.000	ga contract negotiation	3.000
pd support analysis	2.000	ga maint config procs	3.000
ga system software test	2.000	ga prep test plans	3.000
ga def mod test cases	2.000	ga prep trouble reports	3.000
ga software module test	2.000	ga quality assurance	3.000
pd conduct walkthroughs	2.000	ga documenting code	3.000
pd conduct req review	2.000	ga interv personnel	3.000
pd attend req review	2.000	ga technical management	3.000
pd system analysis	2.000	pd sup formulate policy	3.000
pd design	2.000	pd sup program mgmt	3.000
pd conduct design review	2.000	pd sup config management	3.000
pd attend design review	2.000		
pd code	2.000		
pd other development	2.500		
pd SCCB participation	2.500		
pd formulation of policy	2.500		
ga preparing budgets	2.500		
pd other support	2.500		
pd program management	2.500		
pd quality assurance	2.500		
pd monitoring contracts	2.500		
pd sup monitor contracts	2.500		
ga prep version audits	2.500		
ga library control	2.500		
ga preparing schedules	2.500		
ga prep fld eng rpts	2.500		
ga prep test plans	3.000		
ga documenting code	3.000		
ga quality assurance	3.000		
ga maint config procs	3.000		
pd sup formulate policy	3.000		
pd sup program mgmt	3.000		
ga interv personnel	3.000		
pd sup config management	3.000		
ga technical management	3.000		
ga prep trouble reports	3.000		
ga contract negotiation	3.000		

Cluster 4

ga technical management	1.000	ga program management	1.000
ga preparing schedules	1.000	ga sales marketing	1.000
ga reviewing tech work	1.000	ga contract negotiation	1.000
ga func system design	1.000	ga formulating policy	1.000
ga func module design	1.000	ga formulating strategy	1.000
ga def glob data strucs	1.000	ga preparing budgets	1.000
ga def subsys interface	1.000	ga interv personnel	1.000
ga def stuff for own use	1.000	pd attend sup walkthru	1.000
ga coding	1.000	ga prep fld eng rpts	1.000
ga debug or modifying	1.000	ga other admin tasks	1.000
ga prep sys rqt docs	1.000	ga tech advice to CCB	1.000
pd system analysis	1.000	ga maint config procs	1.000
ga prep user manuals	1.000	ga library control	1.000

ga documenting code	1.000	ga prep version audits	1.000
ga defining test cases	1.000	ga quality assurance	1.000
ga prep test drivers	1.000	ga prep temp eng rpt	1.000
ga prep test plans	1.000	ga prep vers descr mnls	1.000
ga system software test	1.000	ga updat MIL STD specs	1.000
ga def mod test cases	1.000	ga teaching	1.000
ga software module test	1.000	ga updat training man	1.000
ga doc test results	1.000	ga being trained	1.000
ga prep trouble reports	1.000	pd other support	1.000
ga anal trouble rpts	1.000	pd sup monitor contracts	1.000
pd conduct req review	1.000	pd sup quality assurance	1.000
pd attend req review	1.000	pd formulation of policy	1.000
pd attend walkthroughs	1.000	pd sup program mgmt	1.000
pd design	1.000	pd program management	1.000
pd conduct design review	1.000	pd configuration mngmnt	1.000
pd attend design review	1.000	pd quality assurance	1.000
pd code	1.000	pd monitoring contracts	1.000
pd conduct walkthroughs	1.000	pd other development	1.000
pd technical management	1.000	pd conduct support dr	1.000
pd support analysis	1.000	pd conduct sup walkthru	1.000
pd support design	2.000	pd sup technical mgmt	1.000
ga prep redlined docs	2.500	pd sup formulate policy	1.000
pd code patch	2.500	pd sup config management	1.000
pd SCCB participation	2.500	ga prep mgmt info rpts	2.000
		ga hardware testing	2.000
		ga reading tech pubs	2.000
		pd attend support dr	2.500
		pd formulating strategy	2.500
		ga prep technical rpts	2.500

Cluster 7

ga preparing budgets	1.000	ga anal trouble rpts	1.000
ga prep mgmt info rpts	1.000	ga prep trouble reports	1.000
ga program management	1.000	pd SCCB participation	1.000
ga prep technical rpts	1.000	ga def mod test cases	1.000
ga formulating strategy	1.500	pd attend support dr	1.000
pd sup program mgmt	2.000	ga documenting code	1.000
ga technical management	2.000	pd attend design review	1.000
ga tech advice to CCB	2.000	pd conduct design review	1.000
ga prep sys rct docs	2.000	pd conduct walkthroughs	1.000
ga quality assurance	2.000	pd design	1.000
pd other development	2.500	ga software module test	1.000
ga preparing schedules	2.500	pd support analysis	1.500
pd sup monitor contracts	2.500	ga prep test plans	1.500
ga prep fld eng rpts	2.500	pd attend walkthroughs	1.500
pd program management	2.500	pd code patch	1.500
ga library control	2.500	ga func module design	2.000
ga prep version audits	2.500	ga def subsys interface	2.000
ga teaching	2.500	ga def stuff for own use	2.000
ga prep redlined docs	2.500	ga debug or modifying	2.000
pd formulation of policy	2.500	pd system analysis	2.000
pd monitoring contracts	2.500	pd conduct support dr	2.000

pd quality assurance	2.500	pd code	2.000
ga reviewing tech work	2.500	pd conduct sup walkthru	2.000
pd other support	2.500	ga coding	2.000
pd conduct req review	3.000	ga hardware testing	2.000
ga defining test cases	3.000	ga system software test	2.000
ga maint config procs	3.000	ga reading tech pubs	2.000
ga prep vers descr mnls	3.000	pd attend req review	2.000
ga func system design	3.000	ga updat training man	2.000
ga interv personnel	3.000	pd technical management	2.500
pd sup technical mgmt	3.000	pd configuration mngmnt	2.500
pd sup formulate policy	3.000	ga other admin tasks	2.500
ga formulating policy	3.000	pd formulating strategy	2.500
pd sup config management	3.000	ga prep user manuals	2.500
pd sup quality assurance	3.000	ga updat MIL STD specs	2.500
ga contract negotiation	3.000	ga def glob data strucs	2.500
ga prep test drivers	3.000	pd attend sup walkthru	2.500
		ga being trained	2.500
		ga prep temp eng rpt	2.500
		pd support design	2.500
		ga doc test results	2.500
		ga sales marketing	2.500
		ga interv personnel	3.000
		ga contract negotiation	3.000
		ga maint config procs	3.000
		pd sup technical mgmt	3.000
		ga prep test drivers	3.000
		ga defining test cases	3.000
		ga prep vers descr mnls	3.000
		ga func system design	3.000
		pd conduct req review	3.000
		pd sup formulate policy	3.000
		ga formulating policy	3.000
		pd sup config management	3.000
		pd sup quality assurance	3.000

Cluster 153

pd other support	1.000	ga debug or modifying	1.000
pd sup monitor contracts	1.000	ga system software test	1.000
ga contract negotiation	1.000	pd support design	1.000
pd sup config management	1.000	ga defining test cases	1.000
pd SCCB participation	1.000	pd code	1.000
ga interv personnel	1.000	ga func system design	1.000
pd sup formulate policy	1.000	ga func module design	1.000
ga prep fld eng rpts	1.000	ga def glob data strucs	1.000
ga other admin tasks	1.000	ga def subsys interface	1.000
ga tech advice to CCB	1.000	ga def stuff for own use	1.000
ga maint config procs	1.000	ga coding	1.000
ga library control	1.000	ga prep test plans	1.500
ga prep version audits	1.000	pd code patch	1.500
ga reading tech pubs	1.000	ga prep test drivers	1.500
ga quality assurance	1.000	pd attend walkthroughs	1.500
ga prep redlined docs	1.000	pd support analysis	1.500

pd other development	1.000	ga prep sys rat docs	1.500
pd monitoring contracts	1.000	ga preparing budgets	2.000
pd sup program mgmt	1.000	pd conduct walkthroughs	2.000
pd configuration mngmnt	1.000	ga def mod test cases	2.000
pd program management	1.000	pd conduct req review	2.000
pd formulation of policy	1.000	ga software module test	2.000
pd sup quality assurance	1.000	ga prep trouble reports	2.000
pd quality assurance	1.000	pd conduct design review	2.000
ga hardware testing	1.500	ga technical management	2.000
ga prep vers descr mnls	1.500	ga preparing schedules	2.000
pd formulating strategy	1.500	pd attend design review	2.000
pd sup technical mgmt	1.500	ga documenting code	2.000
ga prep temp eng rpt	1.500	ga doc test results	2.500
ga formulating policy	1.500	pd attend support dr	2.500
ga formulating strategy	1.500	ga anal trouble rpts	2.500
ga sales marketing	1.500	ga prep user manuals	2.500
ga program management	2.000	ga being trained	2.500
pd technical management	2.000	ga prep technical rpts	2.500
ga prep mgmt info rpts	2.000	pd design	2.500
ga updat MIL STD specs	2.000	pd attend sup walkthru	2.500
ga updat training man	2.000	pd conduct support dr	3.000
ga teaching	2.500	pd system analysis	3.000
ga reviewing tech work	2.500	pd attend req review	3.000
pd conduct support dr	3.000	pd conduct sup walkthru	3.000
pd conduct sup walkthru	3.000		
pd system analysis	3.000		
pd attend req review	3.000		

{insert analysis summary here}

The support employees were clustered (see Appendix 3) into three clusters, and these were distributed according to the following table:

	Count	Percent
cluster1	74	56.9
cluster7	43	33.1
cluster14	13	10.0

The following median polish summarizes the properties of these clusters.

Analysis of Clustering Based on Primary General
Activities and Principal Duties of Support Employees

Effects and Typicalis after Polishing

	effects
common	2.000

cluster1	1.000
cluster7	0.000
cluster14	-1.000
ga program management	-1.000
ga sales marketing	-1.000
ga contract negotiation	-2.000
ga formulating policy	-1.000
ga formulating strategy	-1.000
ga preparing budgets	1.000
ga technical management	2.000
ga interv personnel	-2.000
ga preparing schedules	-1.000
ga prep mgmt info rpts	0.000
ga prep fld eng rpts	-2.000
ga other admin tasks	-1.000
ga tech advice to CCB	-1.000
ga maint config procs	-1.000
ga library control	-1.000
ga prep version audits	-2.000
ga quality assurance	0.000
ga prep temp eng rpt	-1.000
ga prep technical rpts	0.000
ga reading tech pubs	0.000
ga reviewing tech work	1.000
ga teaching	1.000
ga updat training man	-1.000
ga being trained	3.000
ga func system design	13.000
ga func module design	13.000
ga def glob data strucs	1.000
ga def subsys interface	1.000
ga def stuff for own use	3.000
ga coding	2.000
ga debug or modifying	5.000
ga prep sys rqt docs	2.000
ga updat MIL STD specs	-2.000
ga prep redlined docs	0.000
ga prep vers descr mnls	-1.000
ga prep user manuals	-1.000
ga documenting code	2.000
ga defining test cases	4.000
ga prep test drivers	0.000
ga prep test plans	4.000
ga hardware testing	-1.000
ga system software test	8.000
ga def mod test cases	0.000
ga software module test	2.000
ga doc test results	5.000
ga prep trouble reports	5.000
ga anal trouble rpts	3.000
pd conduct req review	1.000
pd attend req review	7.000

pd system analysis	9.000
pd design	11.000
pd conduct design review	1.000
pd attend design review	6.000
pd code	7.000
pd conduct walkthroughs	0.000
pd attend walkthroughs	3.000
pd formulation of policy	-1.000
pd formulating strategy	-1.000
pd technical management	0.000
pd program management	-1.000
pd configuration mgmt	-1.000
pd quality assurance	0.000
pd monitoring contracts	0.000
pd other development	-1.000
pd support analysis	5.000
pd support design	2.000
pd conduct support dr	0.000
pd attend support dr	5.000
pd code patch	3.000
pd conduct sup walkthru	-1.000
pd attend sup walkthru	2.000
pd sup technical mgmt	0.000
pd sup formulate policy	-2.000
pd sup program mgmt	-2.000
pd sup config management	-1.000
pd sup quality assurance	0.000
pd sup monitor contracts	1.000
pd other support	-1.000

Residuals after Polishing

	ga contract negotiation	ga preparing budgets	ga sales marketing	ga formulating strategy	ga interv personnel	ga program management	ga formulating policy	ga technical management
cluster1	-2.000	-2.000	-1.000	-2.000	-1.000	-4.000	-3.000	-1.000
cluster7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster14	1.000	0.000	1.000	0.000	0.000	0.000	3.000	3.000

	ga prep fld eng rpts	ga maint config procs	ga prep mgmt info rpts	ga tech advice to CCB	ga preparing schedules	ga other admin tasks	ga library control
cluster1	-1.000	1.000	-1.000	-2.000	-2.000	-1.000	-2.000
cluster7	0.000	0.000	0.000	1.000	1.000	0.000	0.000
cluster14	2.000	0.000	1.000	0.000	0.000	0.000	1.000

	ga prep technical rpts	ga prep temp eng rpt	ga reviewing tech work	ga quality assurance	ga reading tech pubs	ga teaching	ga being trained	ga updat training man
cluster1	-3.000	-1.000	-1.000	0.000	0.000	0.000	-2.000	0.000
cluster7	0.000	1.000	2.000	0.000	-1.000	1.000	1.000	0.000
cluster14	0.000	0.000	0.000	4.000	6.000	-2.000	0.000	-2.000

	ga def glob data strucs	ga func module design	ga def stuff for own use	ga coding	ga prep sys rqt docs	ga func system design	ga def subsys interface	ga debug or modifying
cluster1	0.000	24.000	20.000	24.000	36.000	63.000	63.000	2.000
cluster7	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	-2.000
cluster14	-9.000	-11.000	0.000	-1.000	-3.000	-3.000	-6.000	0.000

	ga prep vers descr mnls	ga defining test cases	ga prep redlined docs	ga documenting code	ga prep test plans	ga updat MIL STD specs	ga prep user manuals	ga prep test drivers
cluster1	0.000	0.000	3.000	6.000	31.000	3.000	3.000	6.000
cluster7	0.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster14	2.000	0.000	0.000	0.000	-2.000	-3.000	0.000	-3.000

	ga def mod test cases	ga prep trouble reports	ga system software test	ga doc test results	pd conduct req review	ga hardware testing	ga software module test	ga anal trouble rpts
cluster1	0.000	19.000	24.000	39.000	7.000	1.000	7.000	0.000
cluster7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cluster14	1.000	-6.000	0.000	-1.000	-6.000	-4.000	0.000	4.000

	pd attend walkthroughs	pd conduct design review	pd conduct walkthroughs	pd system analysis	pd attend design review	pd attend req review	pd design	pd code
cluster1	0.000	0.000	43.000	19.000	30.000	58.000	9.000	15.000
cluster7	-5.000	1.000	0.000	-3.000	-1.000	0.000	0.000	-3.000
cluster14	3.000	0.000	-8.000	0.000	0.000	-7.000	0.000	0.000

	pd technical management	pd quality assurance	pd formulating strategy	pd configuration mngmnt	pd other development	pd formulation of policy	pd program management	pd monitoring contracts
cluster1	-2.000	0.000	-1.000	-2.000	0.000	-2.000	-3.000	0.000
cluster7	0.000	-1.000	0.000	0.000	1.000	2.000	0.000	3.000
cluster14	2.000	0.000	4.000	0.000	0.000	0.000	1.000	0.000

			pd conduct support dr	pd conduct sup walkthru				
			pd support design	pd code patch		pd sup technical mgmt		
			pd support analysis	pd attend support dr		pd attend sup walkthru		
cluster1	1.000	13.000	0.000	0.000	19.000	1.000	0.000	0.000
cluster7	0.000	0.000	1.000	0.000	0.000	0.000	0.000	-2.000
cluster14	-1.000	-2.000	-1.000	-2.000	-3.000	0.000	-2.000	1.000

				pd sup config management			
				pd SCCB participation		pd other support	
				pd sup program mgmt		pd sup monitor contracts	
				pd sup formulate policy		pd sup quality assurance	
cluster1	-1.000	-1.000	-1.000	0.000	-3.000	-4.000	-1.000
cluster7	0.000	0.000	0.000	0.000	0.000	0.000	5.000
cluster14	1.000	2.000	0.000	0.000	0.000	0.000	0.000

Categories with High Positive
Residuals in Rank Order

Categories with High Negative
Residuals in Rank Order

Cluster 1

pd conduct walkthroughs	1.000	ga program management	1.000
pd code patch	1.000	ga sales marketing	1.000
pd conduct sup walkthru	1.000	ga contract negotiation	1.000
pd support design	1.000	ga formulating policy	1.000
ga func module design	1.000	ga formulating strategy	1.000
ga def glob data strucs	1.000	ga preparing budgets	1.000
ga def subsys interface	1.000	ga technical management	1.000
ga def stuff for own use	1.000	ga interv personnel	1.000
ga coding	1.000	ga preparing schedules	1.000
ga debug or modifying	1.000	ga prep mgmt info rpts	1.000
ga prep sys rqt docs	1.000	ga prep fid eng rpts	1.000
pd support analysis	1.000	ga other admin tasks	1.000
ga prep vers descr mntis	1.000	ga tech advice to CCB	1.000
ga prep user manuals	1.000	ga maint config proc	1.000
ga documenting code	1.000	ga library control	1.000
ga defining test cases	1.000	ga quality assurance	1.000
ga prep test drivers	1.000	pd other support	1.000
ga prep test plans	1.000	ga prep temp eng rpt	1.000
ga system software test	1.000	ga prep technical rpts	1.000
ga def mod test cases	1.000	pd SCCB participation	1.000
ga software module test	1.000	pd sup program mgmt	1.000
ga doc test results	1.000	pd sup formulate policy	1.000
ga prep trouble reports	1.000	pd formulation of policy	1.000
ga anal trouble rpts	1.000	pd quality assurance	1.000
pd attend walkthru	1.000	pd program management	1.000
pd design	1.000	pd technical management	1.000
pd conduct design review	1.000	ga updat training man	1.000
pd attend design review	1.000	pd monitoring contracts	1.000
pd code	1.000	pd sup monitor contracts	1.000
pd attend support dr	1.500	pd sup quality assurance	1.000
pd attend sup walkthru	1.500	ga prep version audits	1.500

pd formulating strategy	1.500	pd configuration mngmnt	1.500
ga prep redlined docs	1.500	ga updat MIL STD specs	1.500
ga func system design	1.500	pd system analysis	1.500
ga being trained	1.500	pd other development	1.500
ga reviewing tech work	2.000	ga reading tech pubs	1.500
ga teaching	2.000	ga hardware testing	1.500
pd sup technical mgmt	2.000	pd conduct req review	1.500
pd attend req review	2.000	pd conduct support dr	2.000
pd conduct support dr	2.000	pd sup technical mgmt	2.000
pd sup config management	2.000	ga teaching	2.000
		ga reviewing tech work	2.000
		pd sup config management	2.000
		pd attend req review	2.000

Cluster 7

pd other support	1.000	ga prep sys rqt docs	1.000
ga prep technical rpts	1.000	pd attend req review	1.000
ga prep temp eng rpt	1.000	pd formulating strategy	1.000
ga updat training man	1.000	pd sup technical mgmt	1.000
pd system analysis	1.000	pd conduct design review	1.000
ga tech advice to CCB	1.000	ga def glob data strucs	1.000
pd conduct support dr	1.000	ga prep redlined docs	1.000
pd other development	1.000	pd attend design review	1.000
pd quality assurance	1.000	ga reviewing tech work	1.000
pd configuration mngmnt	1.000	pd attend walkthroughs	1.000
ga other admin tasks	1.000	ga prep version audits	1.500
ga teaching	1.000	pd conduct walkthroughs	1.500
pd program management	1.500	ga def mod test cases	1.500
ga preparing budgets	1.500	ga prep user manuals	1.500
ga formulating strategy	1.500	pd conduct req review	1.500
ga func system design	1.500	ga prep vers descr mnls	1.500
ga formulating policy	1.500	ga anal trouble rpts	1.500
pd sup quality assurance	1.500	ga reading tech pubs	1.500
ga sales marketing	1.500	ga prep test drivers	1.500
ga quality assurance	1.500	pd conduct sup walkthru	1.500
ga being trained	1.500	ga updat MIL STD specs	1.500
ga prep mgmt info rpts	1.500	ga hardware testing	1.500
pd sup monitor contracts	1.500	ga debug or modifying	2.000
ga maint config procs	1.500	ga def stuff for own use	2.000
pd SCCB participation	1.500	ga def subsys interface	2.000
pd attend sup walkthru	1.500	ga prep test plans	2.000
pd attend support dr	1.500	ga library control	2.000
ga def subsys interface	2.000	ga prep fld eng rpts	2.000
ga debug or modifying	2.000	ga func module design	2.000
ga documenting code	2.000	ga technical management	2.000
ga prep test plans	2.000	ga contract negotiation	2.000
ga software module test	2.000	ga program management	2.000
ga doc test results	2.000	ga doc test results	2.000
ga prep trouble reports	2.000	pd design	2.000
ga def stuff for own use	2.000	ga prep trouble reports	2.000
pd design	2.000	ga software module test	2.000
pd code	2.000	ga system software test	2.000

ga library control	2.000	pd code	2.000
pd formulation of policy	2.000	ga defining test cases	2.000
ga prep fld eng rpts	2.000	ga documenting code	2.000
ga preparing schedules	2.000	pd formulation of policy	2.000
pd technical management	2.000	ga coding	2.000
pd monitoring contracts	2.000	pd technical management	2.000
ga interv personnel	2.000	pd monitoring contracts	2.000
pd support analysis	2.000	pd support analysis	2.000
pd support design	2.000	pd support design	2.000
ga technical management	2.000	pd code patch	2.000
ga system software test	2.000	ga preparing schedules	2.000
pd code patch	2.000	ga interv personnel	2.000
ga defining test cases	2.000	pd sup formulate policy	2.000
pd sup formulate policy	2.000	pd sup program mgmt	2.000
pd sup program mgmt	2.000	pd sup config management	2.000
ga coding	2.000		
pd sup config management	2.000		
ga contract negotiation	2.000		
ga func module design	2.000		
ga program management	2.000		

Cluster 14

ga program management	1.000	ga doc test results	1.000
pd conduct req review	1.000	ga software module test	1.000
ga contract negotiation	1.000	pd attend sup walkthru	1.000
ga reviewing tech work	1.000	pd code	1.000
ga reading tech pubs	1.000	ga teaching	1.000
pd sup program mgmt	1.000	pd code patch	1.000
ga technical management	1.000	ga being trained	1.000
ga interv personnel	1.000	ga func system design	1.000
ga preparing schedules	1.000	ga func module design	1.000
pd sup formulate policy	1.000	pd attend support dr	1.000
ga prep fld eng rpts	1.000	ga def subsys interface	1.000
pd sup technical mgmt	1.000	ga def stuff for own use	1.000
pd monitoring contracts	1.000	ga coding	1.000
ga updat MIL STD specs	1.000	ga debug or modifying	1.000
ga library control	1.000	pd conduct support dr	1.000
ga prep version audits	1.000	pd support analysis	1.000
pd technical management	1.000	pd support design	1.000
pd formulation of policy	1.000	ga documenting code	1.000
pd attend req review	1.000	ga defining test cases	1.000
ga hardware testing	1.000	pd design	1.000
pd formulating strategy	1.500	ga prep test plans	1.000
ga formulating strategy	1.500	ga system software test	1.000
ga formulating policy	1.500	ga prep trouble reports	1.000
ga sales marketing	1.500	pd configuration mngmnt	1.500
ga maint config procs	1.500	pd other development	1.500
ga prep redlined docs	1.500	ga def mod test cases	1.500
pd program management	1.500	ga anal trouble rpts	1.500
pd sup monitor contracts	1.500	pd system analysis	1.500
pd sup quality assurance	1.500	ga prep test drivers	1.500
pd SCCB participation	1.500	ga prep user manuals	1.500

ga preparing budgets	1.500	ga prep vers descr mnls	1.500
ga prep mgmt info rpts	1.500	pd conduct sup walkthru	1.500
ga quality assurance	1.500	pd conduct walkthroughs	1.500
ga def glob data strucs	2.000	ga other admin tasks	2.000
pd attend design review	2.000	ga tech advice to CCB	2.000
ga prep sys rqt docs	2.000	pd quality assurance	2.000
pd quality assurance	2.000	pd sup config management	2.000
ga tech advice to CCB	2.000	pd attend design review	2.000
ga other admin tasks	2.000	pd conduct design review	2.000
ga prep temp eng rpt	2.000	ga prep sys rqt docs	2.000
ga prep technical rpts	2.000	ga def glob data strucs	2.000
pd attend walkthroughs	2.000	ga updat training man	2.000
pd sup config management	2.000	ga prep technical rpts	2.000
pd conduct design review	2.000	ga prep temp eng rpt	2.000
ga updat training man	2.000	pd attend walkthroughs	2.000
pd other support	2.000	pd other support	2.000

Cluster 1 of the development employees is not strong on much except training and is weak on technical management and administrative activities and duties. Cluster 3 is strong on support design and coding and weak on document and report preparation. These would be support designers. Cluster 4 is strong on design, coding, and software work, weak on administrative and management activities and sales. These would be programmer analysts. Cluster 7 is strong on budgets, reports, and various management activities. It is weak on design and coding. These would be lower-level administrators. Finally, cluster 153 is strong on technical management, contracts, administration, and sales and weak on software work. These would be lower-level technical management.

Support cluster 1 is strong on walkthroughs, analysis, testing, and coding; it is weak on management, sales, and technical management. These would be the maintenance/testing people. Cluster 7 is strong on most support duties and quality assurance and weak on reviews, walkthroughs, and testing. These would be general support personnel. Finally, cluster 14 is strong on management and administrative activities but weak on software related duties and activities. These would be low-level support administrators.

Conclusions

Two general conclusions come out of the single variable descriptions and the two variable tables. First, there is a general tendency in the sample toward development as opposed to support. Secondly, people who have been in the field longer tend to know more and tend to be in higher positions. The first conclusion is moderately interesting, the second hardly surprising.

Experience with Ada seems so minimal at the current time that not much can be said about the influences of the various methods of training or exposure. Design of Ada training programs should be targeted at the appropriate lack of knowledge in the industry as currently structured.

The classification itself makes sense when compared to various parts of the data. But the classification does little more than restate the obvious in that it says people should be classified on the basis of whether they are in support or development or whether they are managers, administrators, or others. The methods used might allow a more complete specification based on particular job outputs, general activities, or principal duties. This specificity is somewhat counterintuitive given the clustering level; that is, it doesn't really make sense to talk about the cluster categories in anything but the most general terms (such as support management).

Lastly, it might well be that either the people responding to the survey or the survey analyst or both have used the inherent classifications (support, development, design methodologies, programming concepts, and so on) to interpret the survey. This might explain the tendency to see things tending to divide up that way. On the other hand, the categories between these concepts are numerous enough and the variation extensive enough to believe that specific conclusions are relatively free from interpretive bias. But at the vaguest levels of

generalization--particularly the interpretation of the cluster categories--the reduction in content might well be sufficient to allow the inherent categories to reemerge, rendering the end product circular. This circularity might have been avoided by a survey format that imposed fewer assumptions about the world; on the other hand, the assumptions that were present are not unreasonable. But the reader should bear in mind the underlying bias due to the survey format when interpreting the results.

Appendix 1

Datasets, Attributes, and Possible Values

Attribute	Codes	
Dataset Ada training		
Ada training	other	informal training
	seminar	programmed learning
	in house course	college course
id	videotape	
Dataset conference role		
conference role	other	attendee
	speaker	organizer
id		
Dataset experience area		
experience area	other	statistical
	educational	embedded computer sys
id	military	commercial
Dataset general activities		
activity	program management	prep test plans
	prep test drivers	defining test cases
	documenting code	prep user manuals
	prep version descr mnls	prep redlined docs
	updating MIL STD specs	prep sys rqt docs
	debugging or modifying	coding
	def stuff for own use	def subsystem interface
	def global data strucs	func module design
	func system design	being trained
	updating training manual	teaching
	reviewing tech work	reading tech pubs
	prep technical rpts	prep temp eng change rpt
	quality assurance	prep version audits
	library control	maint config procs
	tech advice to CCB	other admin tasks
	preparing fld eng rpts	preparing mgmt info rpts
	preparing schedules	interviewing personnel
	technical management	preparing budgets
	formulating strategy	formulating policy

	contract negotiation	sales marketing
	hardware testing	system software test
	defining mod test cases	software module testing
	documenting test results	prep trouble reports
	analyzing trouble rpts	
id		
importance	primary	secondary
	marginal	

Dataset job history

Ada involvement	Ada not known	heard of Ada
	had orientation to Ada	Ada training
	other	
company	company ten	company nine
	company eight	company seven
	company six	company five
	company four	company three
	company two	company one
conference attendance	yes	no
date of survey		
id		
job level	high level mgmt	middle mgmt
	technical	entry level
job title		
published or presented	yes	no
rank		
sector	public	private
technical reading extent	regularly	occasionally
	only as my job demands	never
	other	
technical societies	no	yes
years of involvement	less than two years	two to five years
	five to ten years	over ten years
years worked development	less than a year	one to three years
	three to five years	over five years
years worked support	less than a year	one to three years
	three to five years	over five years

Dataset job output

id		
job output	STR analyses	temp ECPs
	redlined documentation	support test plans
	support test drivers	tech advice to CCB
	updated MIL STD spec	library control
	maintain config procs	updated training manuals
	updated user manuals	STRs
	automated build systems	management info reports
	version descrip docs	version audits
	field engineering report	support other

development other	correspondence
interview sheets	status reports
milestone charts	analysis reports
cost data	management plans
integration plans	test plans
design specs	requirements specs
prog design language	code
test drivers	data flow diagrams
hard soft tradeoff eval	

Dataset methodologies

id

knowledge

methodology

used frequently	used moderately
know concept	heard of
other Ada concepts	Ada mutual recursion
Ada information hiding	Ada context spec
Ada elaboration	Ada instantiation
Ada generic prog units	Ada exceptions
Ada separate compilation	Ada entry families
Ada entries	Ada rendezvous
Ada task types	Ada tasking
Ada visibility	Ada short circuiting
Ada scope	Ada private types
Ada packages	Ada overloading
Ada access types	Ada allocators
Ada aggregates	Ada slices
Ada rec types discrim	Ada record types
Ada fixed pt types	Ada float point types
Ada real types	Ada derived types
Ada subtypes	Ada user defined types
Ada enumeration types	other prog concepts
version number	parameter binding
loop invariants	generics
data abstraction	type conversion
strong typing	concurrency
recursion	conditional statements
iteration	static dynamic nesting
name visibility	name scoping
data encapsulation	importing exporting name
other prog constructs	task coroutines
exception handlers	comments
goto statements	stubs
clusters modules package	return statements
functions	procedures
exit statements	loop for while until
if then else statements	case statements
blocks	reserved words
formal actual params	local variables
global variables	object type dcls
variant records	records
ranges	typed pointers

pointers	user defined types
fixed point types	floating point types
enumeration types	PSL PLA
SADT	SREM
HIPO	Jackson Design
Structured Design	Warnier Orr Design
N S Chapin Chart	Beamson Tables
Program Design language	Structured Programming
Structured Walkthroughs	Top Down Design
Top Down Testing	Bottom Up Design
Bachman Diagramming	Entity Diagrams
Data Abstraction	other methodology

Dataset principal duties

id		
principal duties	other support	sup monitoring contracts
	sup quality assurance	sup config management
	SCCB participation	sup program management
	sup policy formulation	sup technical management
	attend sup walkthrough	conduct sup walkthrough
	code patch	attend support dr
	conduct support dr	support design
	support analysis	other development
	monitoring contracts	quality assurance
	configuration management	program management
	technical management	formulation of strategy
	formulation of policy	attend walkthroughs
	conduct walkthroughs	code
	attend design review	conduct design review
	design	system analysis
	attend req review	conduct req review

Dataset programming languages

id		
language	Other	APL
	PPL	PROTEGE
	SAS	GPSS
	ECL	SNOBOL
	LISP	Ada
	FORTH	MMP
	XPL	SIMULA
	MODULA	RATFOR WATFOR WATFIV
	ALGOL	BASIC
	PASCAL	PLI
	ASSEMBLER	COBOL
	FORTRAN	C
	CMS 2	JOVIAL
proficiency	first	second

Appendix 2

Bibliography

1. Donald R. McNeil. Interactive Data Analysis: A Practical Primer. John Wiley and Sons, 1978.
2. John W. Tukey. Exploratory Data Analysis. Addison-Wesley, 1977.
3. Paul F. Velleman and David C. Hoaglin. Applications, Basics, and Computing of Exploratory Data Analysis. Duxbury Press, 1981.

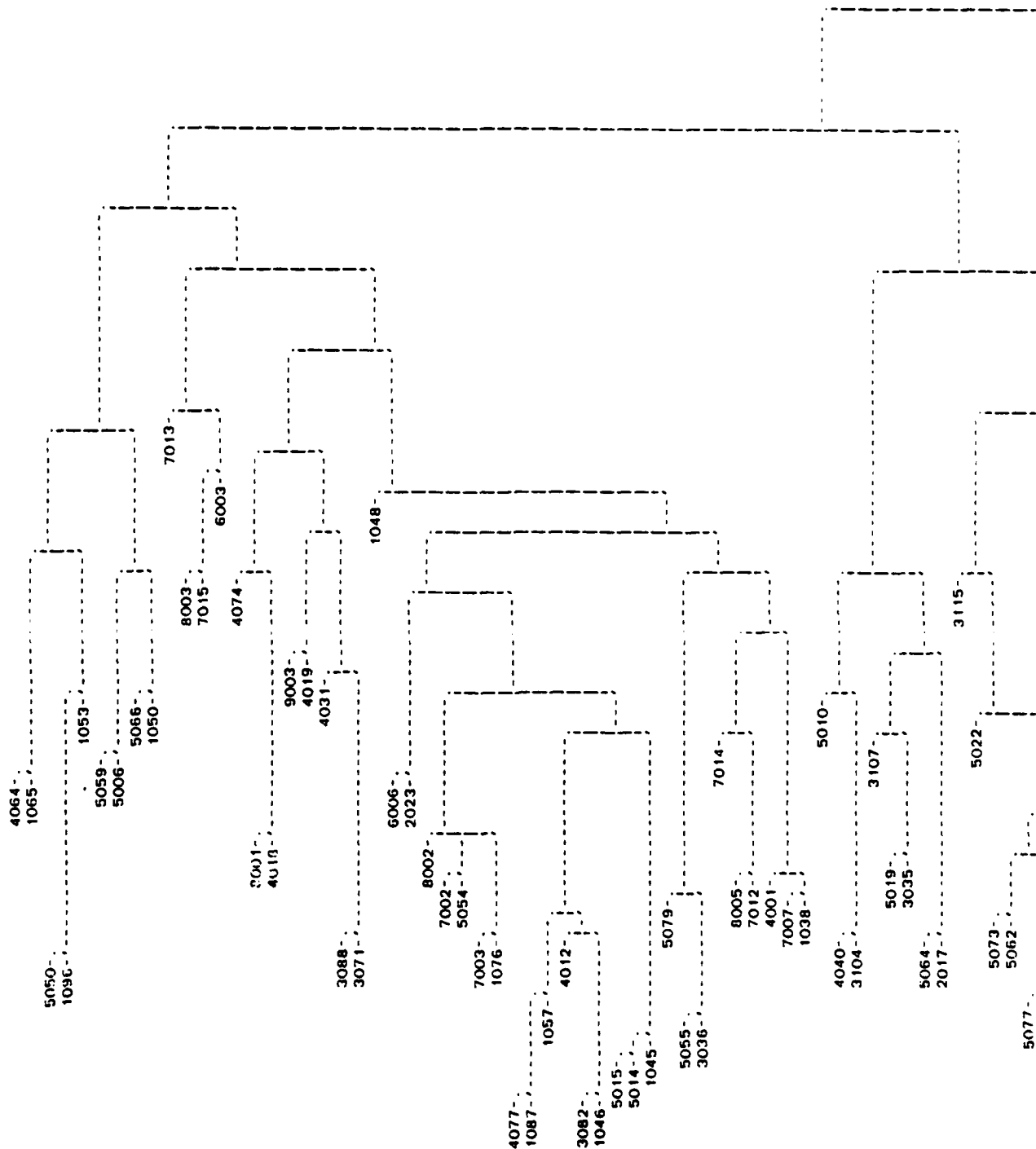
Appendix 3
Cluster Trees

Cluster Tree Based on General
Activities, Principal Duties, and Job Outputs

266
 0.807 47
 0.711 329
 0.886 71
 0.772 35
 0.649 338
 0.798 285
 0.719 344
 0.772 33
 0.553 381
 0.640 386
 0.719 383
 0.667 360
 0.579 276
 0.719 384
 0.833 221
 0.658 391
 0.754 222
 0.693 233
 0.763 176
 0.877 160
 0.614 31
 0.675 363
 0.807 95
 0.728 385
 0.833 370
 0.851 333
 0.833 371
 0.877 52
 0.772 279
 0.947 63
 0.904 39
 0.868 216
 0.877 170
 0.947 30
 0.789 294
 0.930 293
 0.921 29
 0.693 357
 0.860 334
 0.912 126
 0.719 382
 0.789 388
 0.851 380
 0.746 205
 0.851 375
 0.860 22
 0.491 289
 0.772 242
 0.877 188
 0.719 190
 0.789 298
 0.842 125
 0.754 342
 0.877 91
 0.579 197
 0.719 301
 0.781 351
 0.868 340
 0.842 355

1094-2

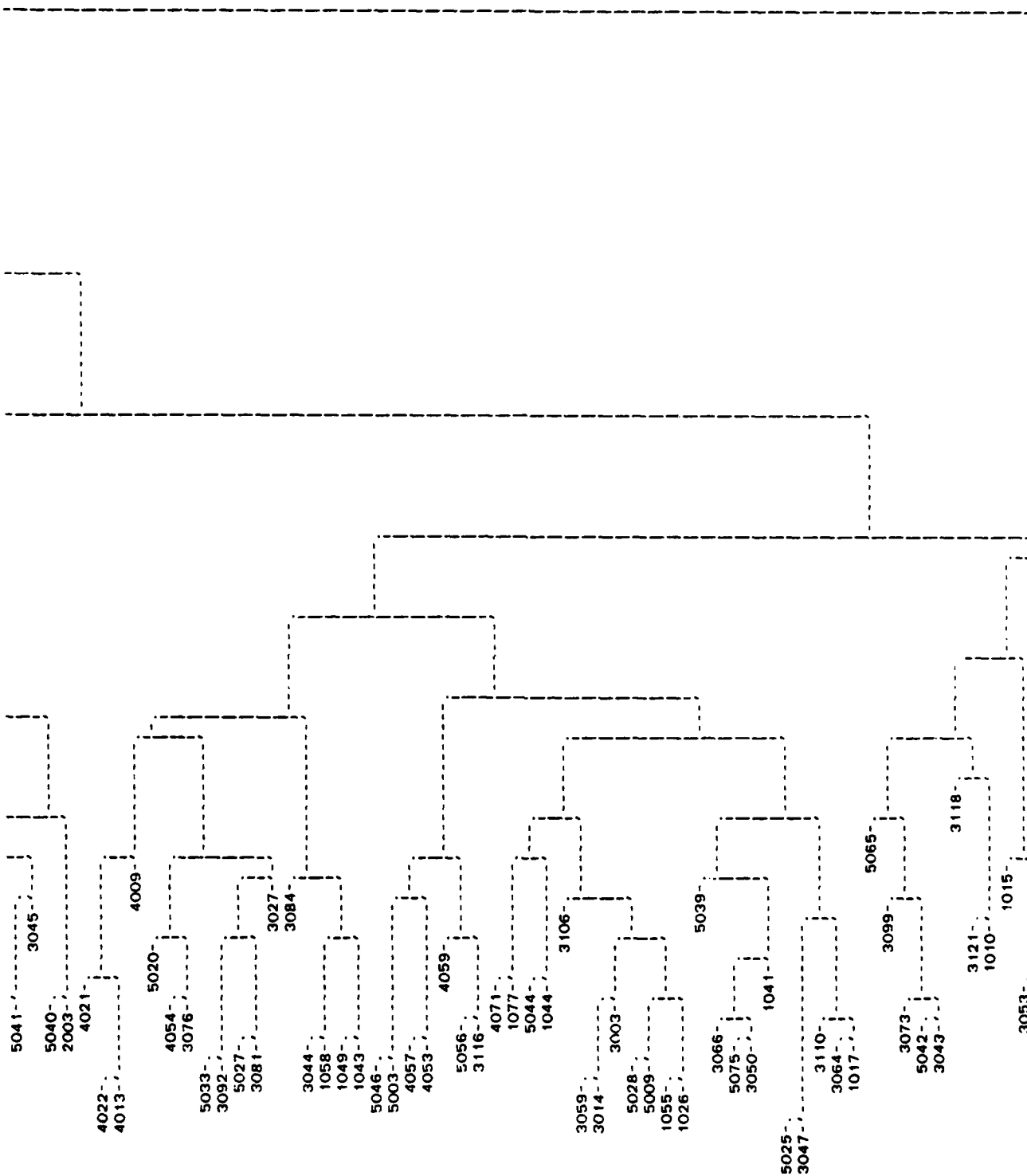
D-100



0.904 320
 0.860 134
 0.825 319
 0.904 78
 0.640 224
 0.895 225
 0.939 217
 0.842 213
 0.789 299
 0.877 256
 0.904 165
 0.842 312
 0.930 180
 0.877 306
 0.921 169
 0.851 117
 0.781 172
 0.851 133
 0.921 40
 0.877 32
 0.921 27
 0.737 325
 0.930 282
 0.860 259
 0.921 255
 0.842 261
 0.877 335
 0.912 198
 0.772 273
 0.895 53
 0.842 323
 0.895 28
 0.825 189
 0.860 148
 0.939 109
 0.904 98
 0.877 307
 0.930 288
 0.904 37
 0.939 19
 0.789 318
 0.851 155
 0.912 353
 0.921 139
 0.886 25
 0.825 304
 0.956 136
 0.868 192
 0.912 153
 0.921 14
 0.693 343
 0.825 185
 0.860 162
 0.904 321
 0.912 132
 0.789 200
 0.807 203
 0.868 7
 0.754 12
 0.842 142

1094-2

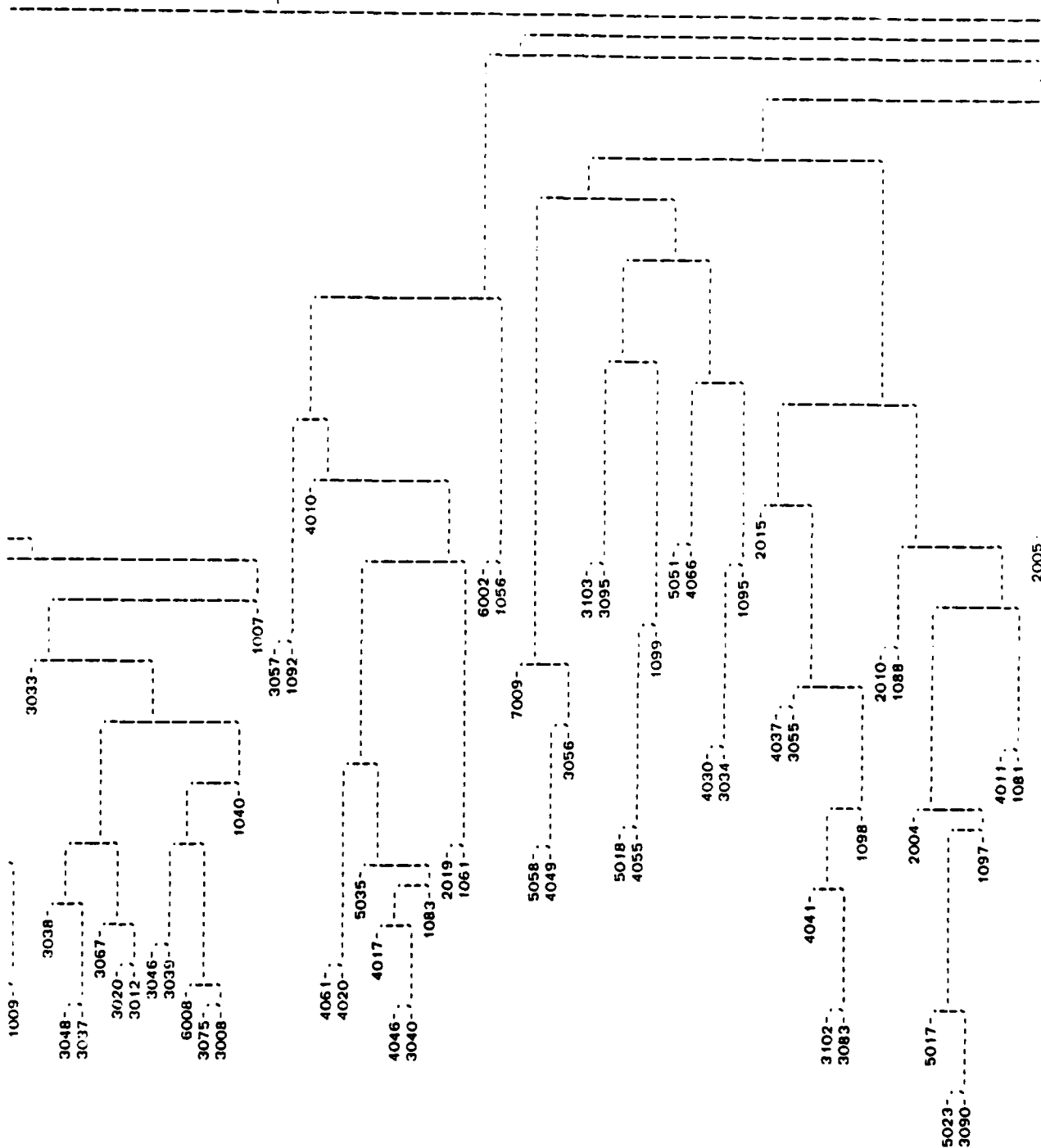
D-101

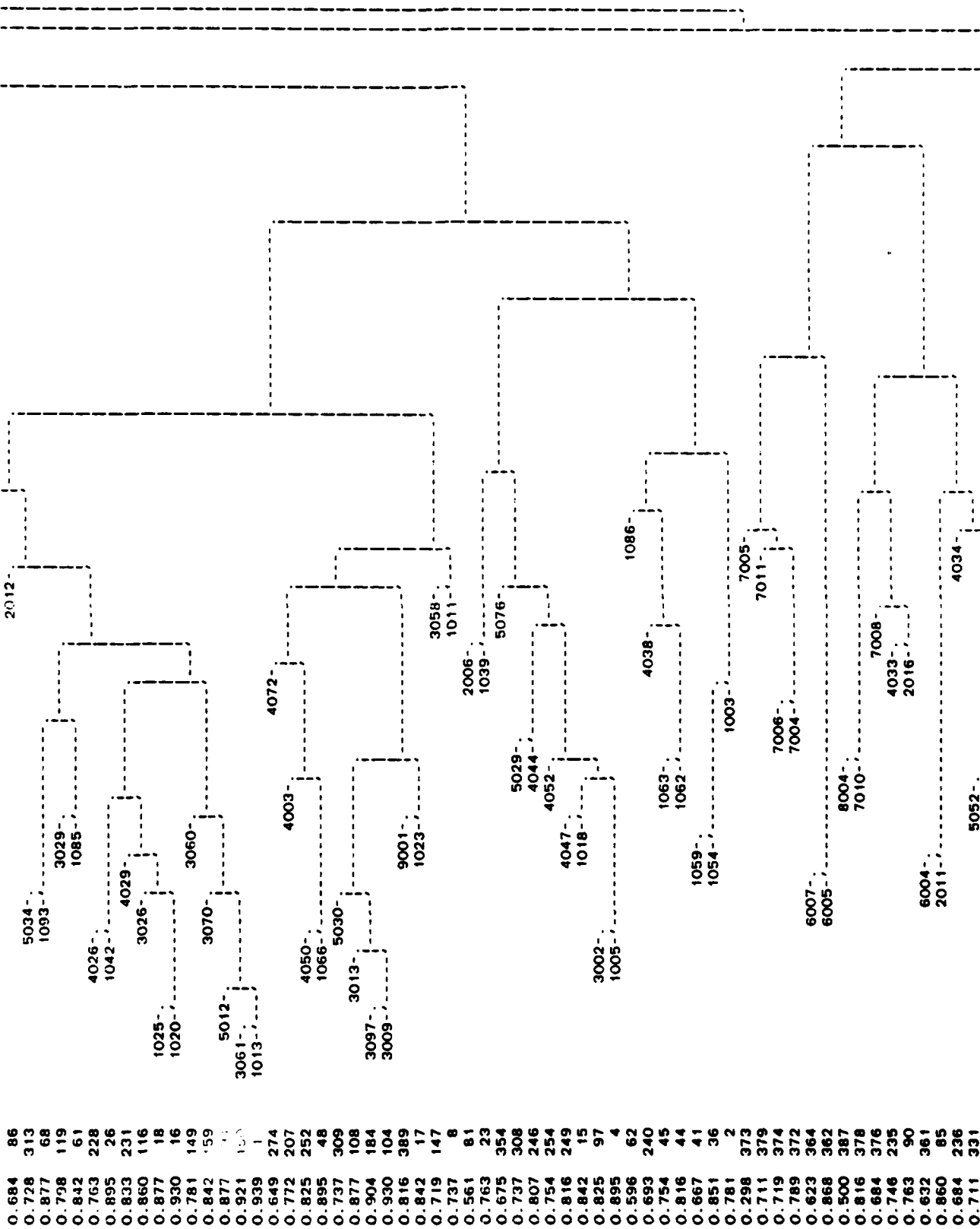


0.895 6
 0.702 123
 0.754 128
 0.860 137
 0.904 127
 0.833 156
 0.868 112
 0.886 107
 0.781 135
 0.877 129
 0.833 365
 0.895 164
 0.904 103
 0.807 24
 0.728 5
 0.281 146
 0.746 67
 0.640 214
 0.667 263
 0.886 223
 0.738 314
 0.842 220
 0.868 248
 0.904 130
 0.851 59
 0.702 93
 0.833 43
 0.588 359
 0.711 38
 0.360 377
 0.754 337
 0.833 251
 0.781 145
 0.544 187
 0.711 183
 0.614 297
 0.825 257
 0.737 74
 0.570 330
 0.693 268
 0.623 232
 0.789 124
 0.711 70
 0.500 89
 0.675 239
 0.772 144
 0.763 243
 0.851 186
 0.904 171
 0.816 73
 0.632 84
 0.746 64
 0.693 79
 0.816 296
 0.904 302
 0.939 178
 0.825 72
 0.728 215
 0.789 57
 0.421 80

1094-2

D-102

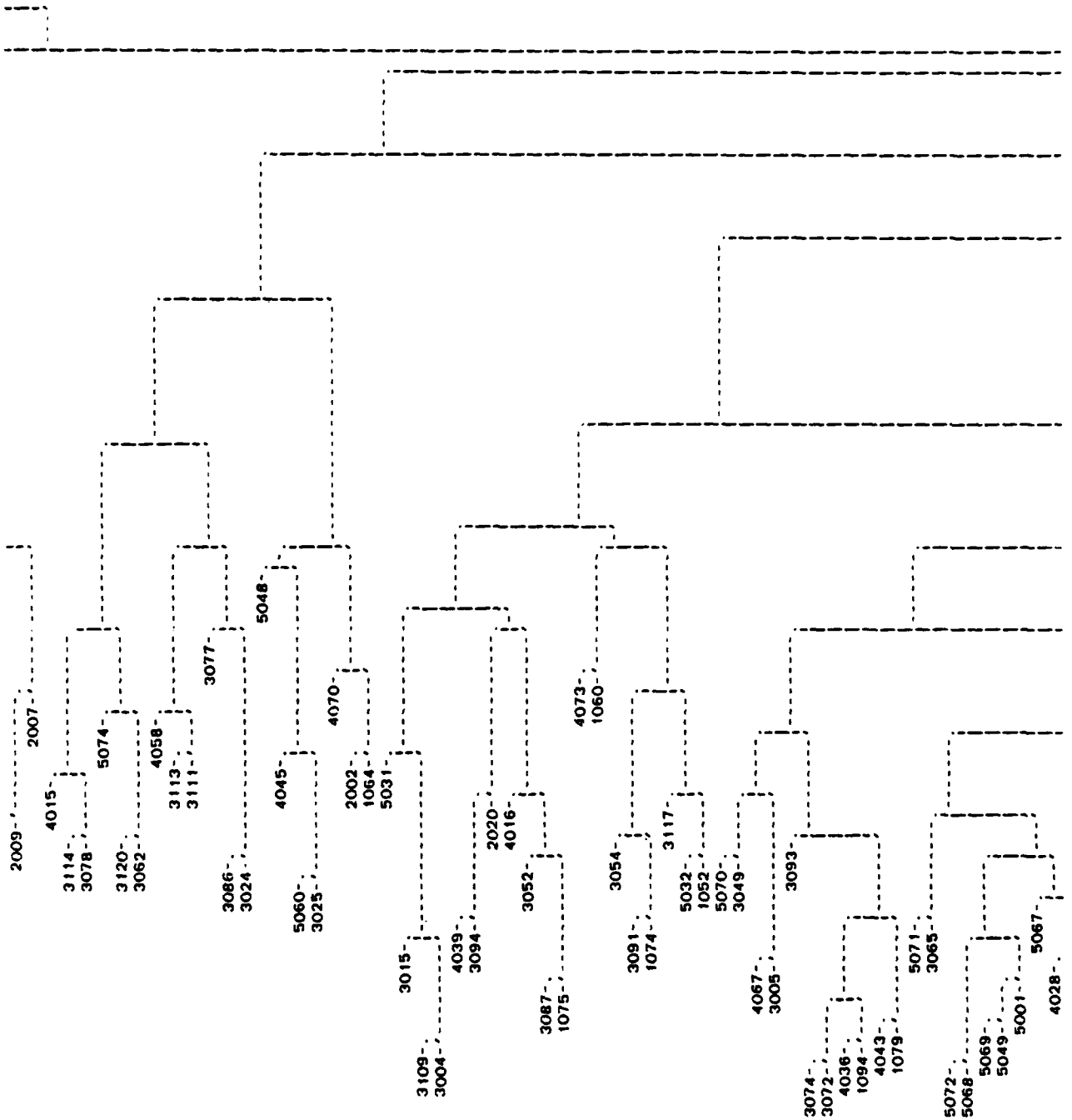




0.825 83
 0.772 82
 0.777 218
 0.807 196
 0.833 167
 0.746 352
 0.781 202
 0.833 151
 0.658 260
 0.781 195
 0.798 193
 0.711 166
 0.746 174
 0.842 114
 0.596 327
 0.719 247
 0.798 339
 0.851 115
 0.702 272
 0.763 77
 0.798 46
 0.535 310
 0.798 110
 0.877 191
 0.921 99
 0.737 241
 0.868 182
 0.816 94
 0.746 219
 0.816 141
 0.842 175
 0.895 51
 0.693 275
 0.763 42
 0.702 143
 0.833 179
 0.868 50
 0.772 199
 0.816 311
 0.842 34
 0.649 348
 0.842 138
 0.816 269
 0.886 100
 0.789 181
 0.833 163
 0.930 161
 0.904 238
 0.921 69
 0.868 245
 0.912 55
 0.746 349
 0.868 154
 0.825 350
 0.930 346
 0.877 347
 0.912 328
 0.895 280
 0.842 345
 0.860 230

1094-2

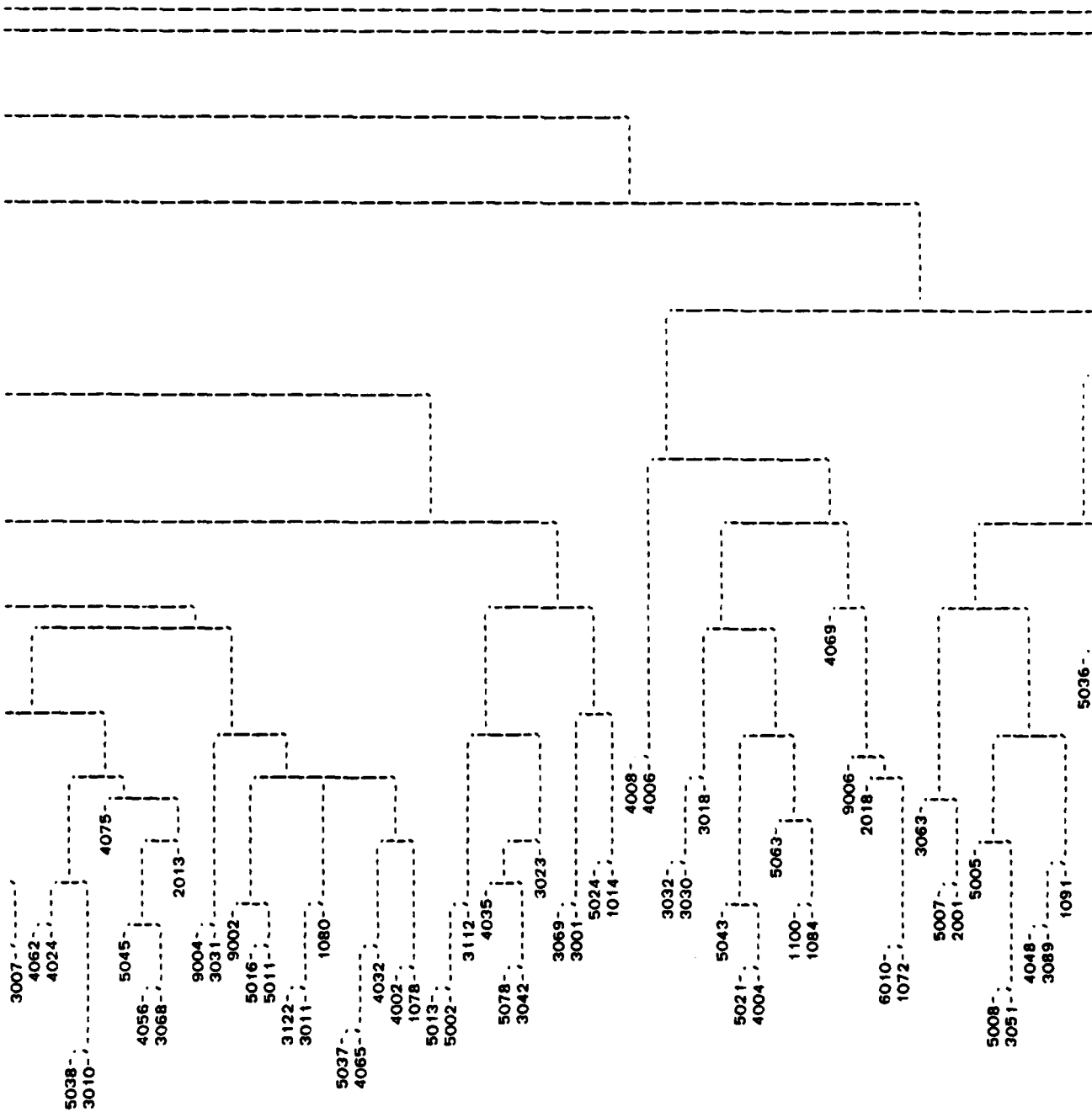
0-104



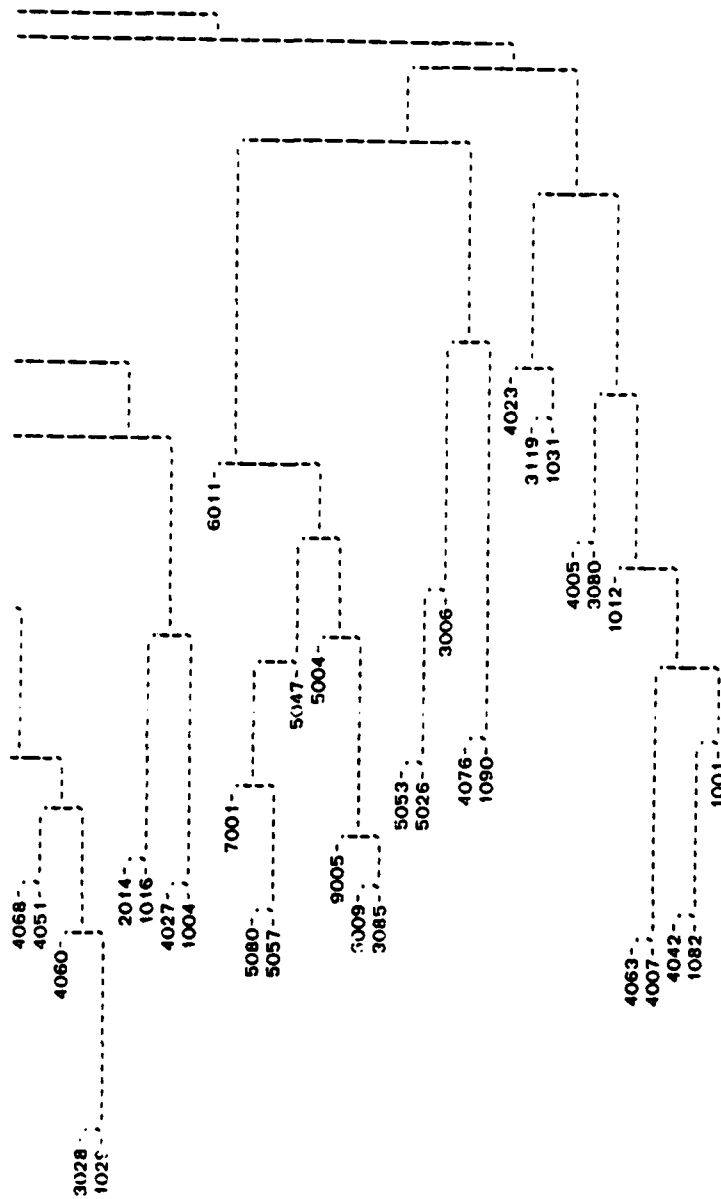
0.886 102
0.789 264
0.877 227
0.860 317
0.930 105
0.816 277
0.825 324
0.877 258
0.904 157
0.842 87
0.754 392
0.877 121
0.798 390
0.868 295
0.886 290
0.816 204
0.904 106
0.868 56
0.816 316
0.921 267
0.886 234
0.842 206
0.895 54
0.702 292
0.904 281
0.868 194
0.798 237
0.860 356
0.895 131
0.842 113
0.746 158
0.868 96
0.789 303
0.851 11
0.570 212
0.807 210
0.675 122
0.851 120
0.816 111
0.754 322
0.868 300
0.895 208
0.798 341
0.833 75
0.868 60
0.711 271
0.746 394
0.807 92
0.816 367
0.886 49
0.614 152
0.825 286
0.860 76
0.746 284
0.842 287
0.904 140
0.798 250
0.877 177
0.851 66
0.711 315

1094-2

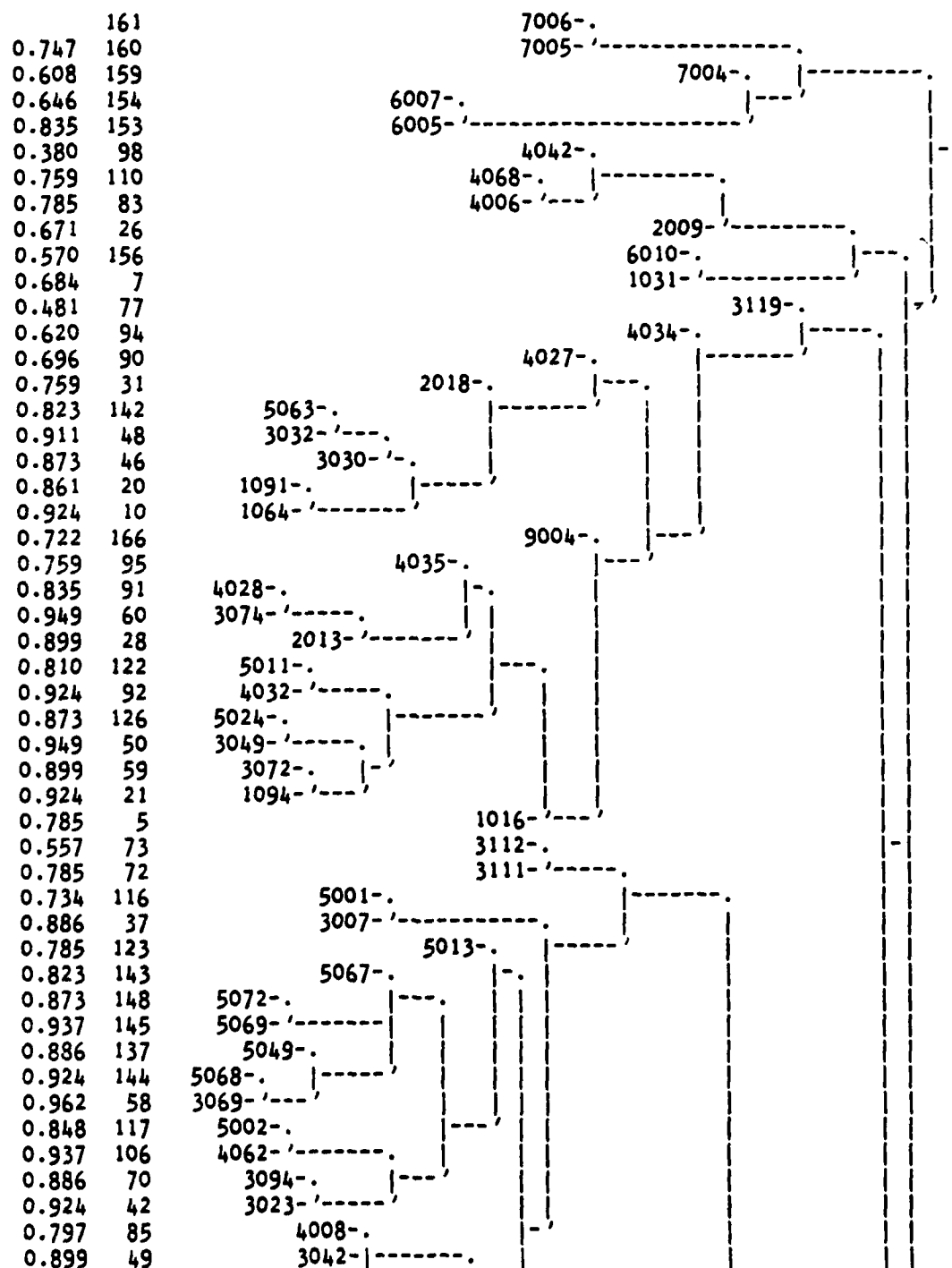
0-105

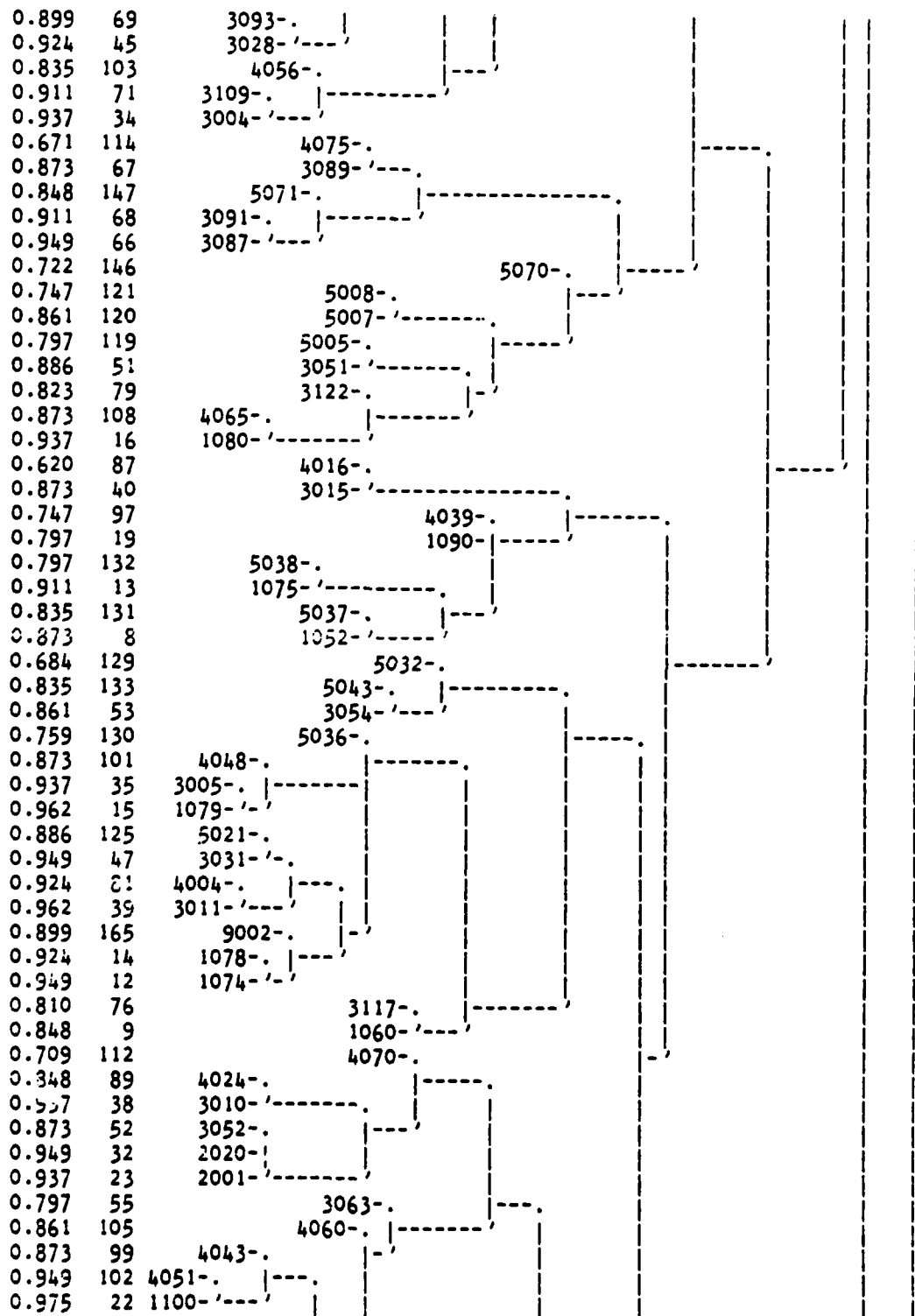


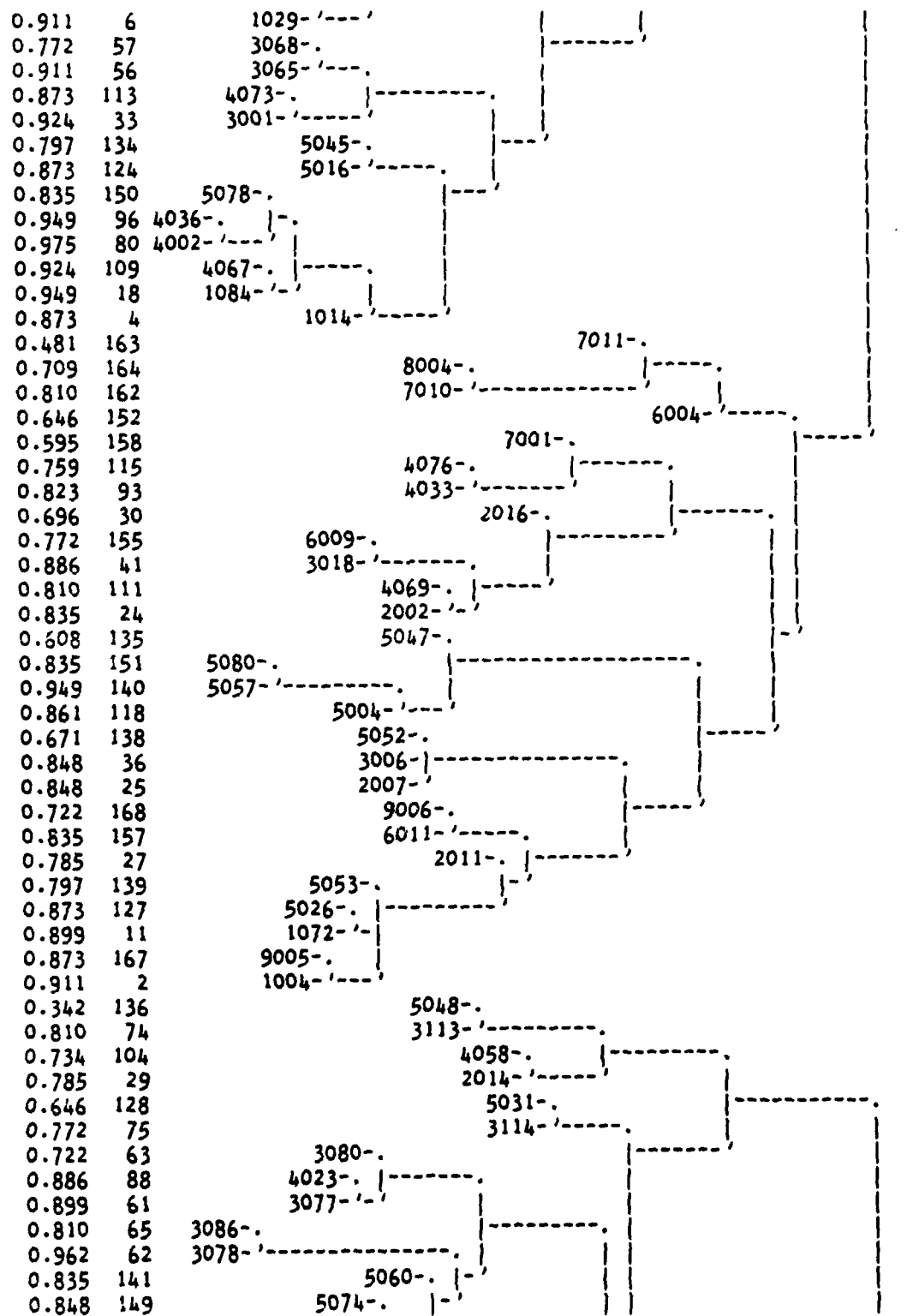
0.763 270
 0.807 253
 0.781 262
 0.825 118
 0.895 20
 0.640 88
 0.798 13
 0.719 225
 0.807 3
 0.421 368
 0.649 369
 0.772 358
 0.816 336
 0.728 326
 0.675 283
 0.719 393
 0.789 366
 0.807 173
 0.535 332
 0.763 305
 0.693 101
 0.605 278
 0.754 65
 0.447 226
 0.614 201
 0.632 21
 0.553 209
 0.675 168
 0.623 9
 0.684 265
 0.825 211
 0.728 244
 0.816 58
 0.754 1



Cluster Tree Based on Principal Duties
and Primary General Activities for Development Employees

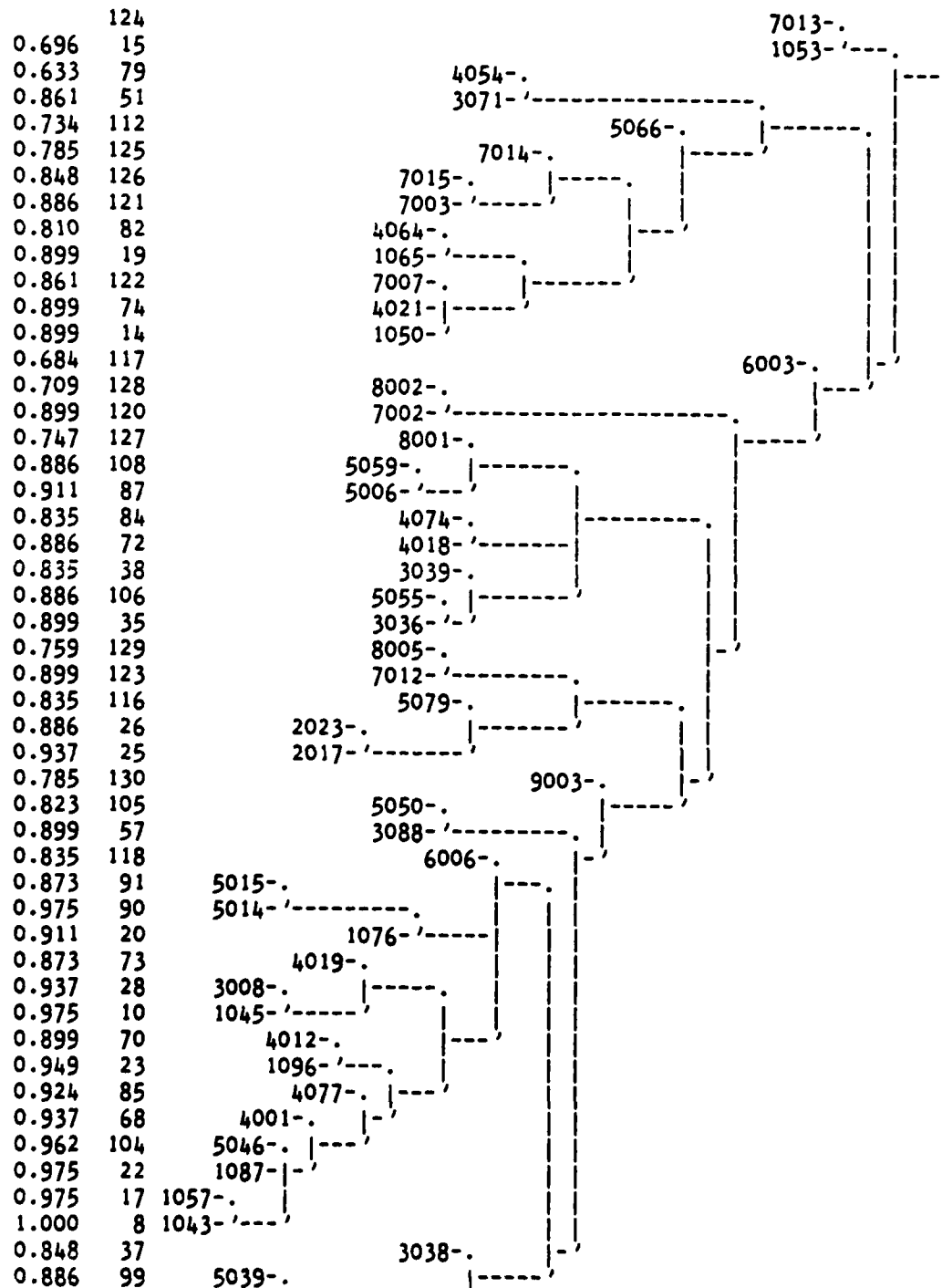




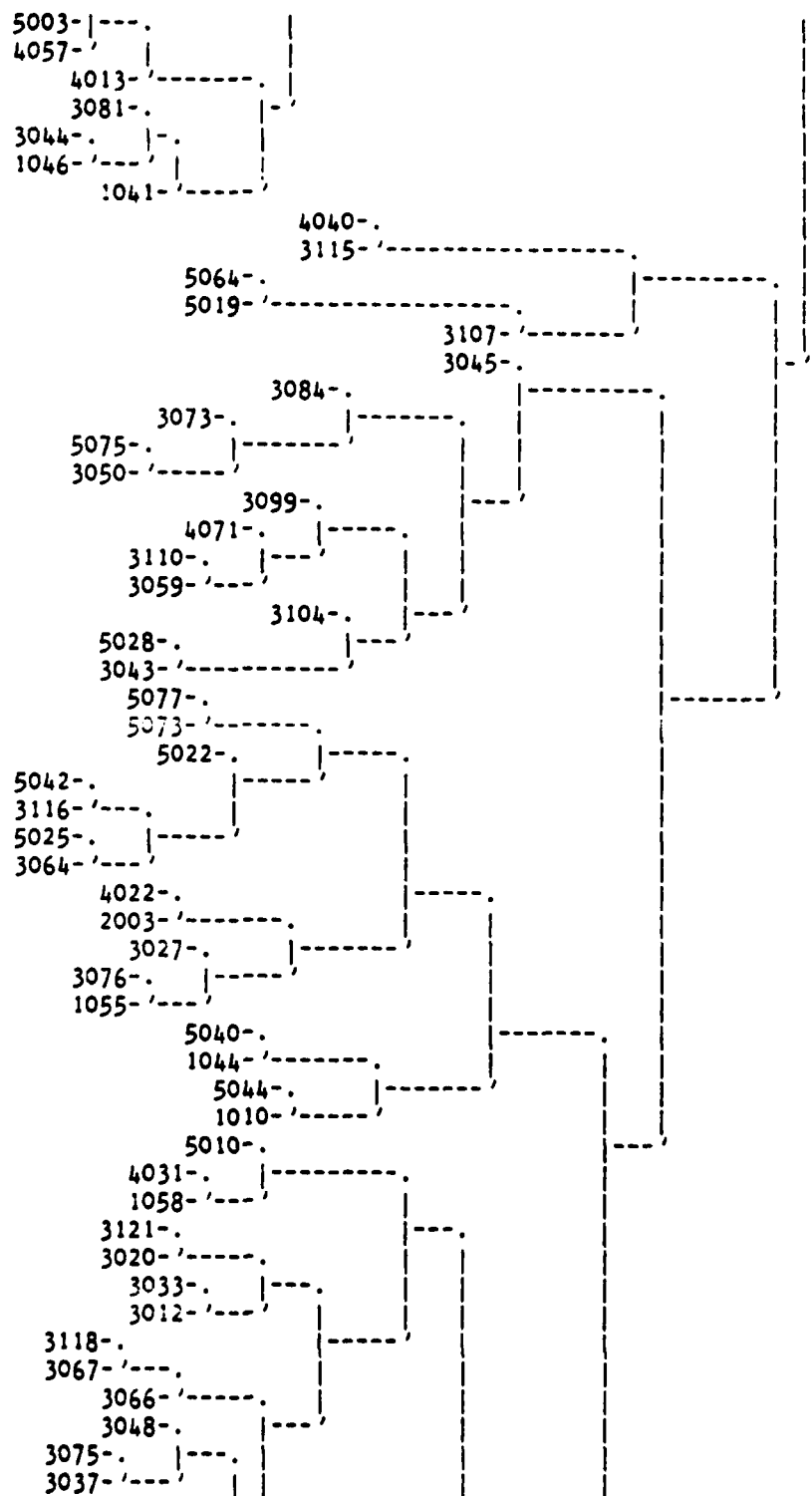


4015- . | - - - ,
4045- . | - - - ,
1082- . - - - ,
1012- | - - ,
4063- .
4007- .
3024- , - - - - - ,
4005- .
3085- , - - - - - .
3120- .
3062- , - - - - - ,
3025- , - - - - - ,
1001- - - - - ,

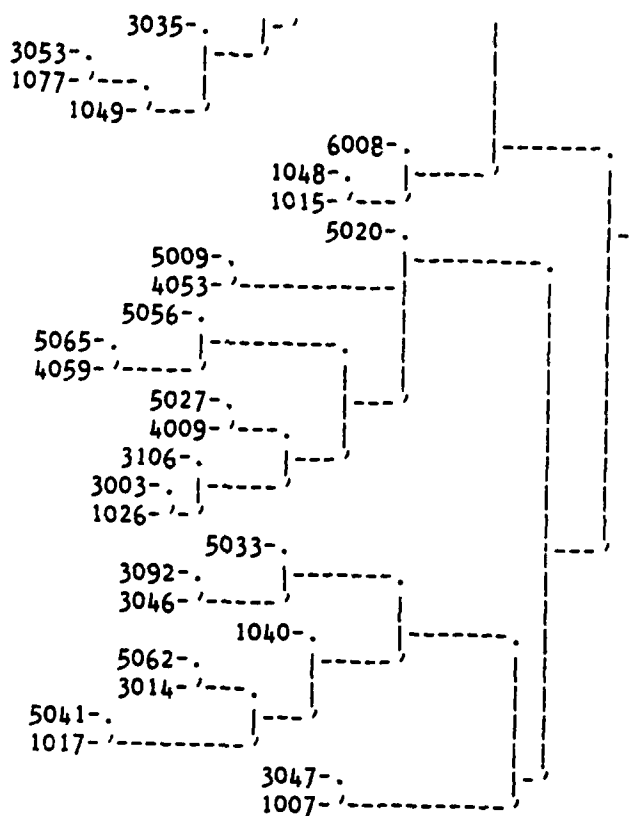
Cluster Tree Based on Principal Duties
and Primary General Activities for Support Employees



0.975 86
 0.975 80
 0.949 71
 0.899 55
 0.949 40
 0.975 11
 0.937 7
 0.532 77
 0.848 64
 0.722 110
 0.899 92
 0.785 62
 0.620 41
 0.785 56
 0.861 52
 0.911 114
 0.949 45
 0.810 59
 0.873 83
 0.899 63
 0.924 47
 0.835 60
 0.861 97
 0.937 39
 0.709 115
 0.924 113
 0.873 94
 0.911 102
 0.975 65
 0.949 95
 0.975 48
 0.835 75
 0.937 24
 0.886 32
 0.924 54
 0.949 16
 0.797 100
 0.899 9
 0.848 103
 0.886 2
 0.734 89
 0.899 76
 0.924 18
 0.335 67
 0.937 31
 0.899 33
 0.924 29
 0.873 66
 0.962 50
 0.937 49
 0.899 44
 0.937 53
 0.962 36



0.911	34
0.937	46
0.987	21
0.962	13
0.810	119
0.848	12
0.873	3
0.747	93
0.848	88
0.924	78
0.848	107
0.937	111
0.975	81
0.873	96
0.924	69
0.899	61
0.937	27
0.949	5
0.785	98
0.899	58
0.937	42
0.848	6
0.886	109
0.937	30
0.911	101
0.975	4
0.797	43
0.873	1



Supplement to Work Force Survey Report:

Subclusters of Development and System Integration Categories

Job Titles for the Development Clusters

Respondent	Job Title	Position in Hierarchy
cluster1		
1001	Consultant	technical
3062	*	*
3025	Senior Engineer	technical
4007	Staff Consultant	technical
3024	R&D Engineer	technical
4005	Programmer Analyst	technical
4063	*	*
3120	Programmer	technical
3085	Senior Engineer	technical

cluster153

7006	Computer Specialist	technical
7005	Computer Specialist	technical
7004	Computer Specialist	technical
6005	*	*
6007	Computer Scientist	technical

cluster3

3078	R&D Engineer	technical
1082	Senior Software Engineer	technical
4045	Senior Analyst	technical
5031	*	*
5048	*	*
5074	Programmer	technical
5060	*	*
3077	Senior Software Engineer	technical
3113	Software Engineer	technical
4058	*	*
4023	Senior System Analyst	technical
3114	Software R&D Engineer	technical
3080	Senior Engineer	technical
2014	Supervising System Analyst	middle_mgmt
1012	Software Design Specialist	technical
3086	*	*
4015	Senior Programmer Analyst	technical

cluster4

2007	Senior Scientific Programmer	technical
2016	Scientific Programmer	technical
1080	Senior Software Engineer	technical
2018	Principle Scientific Programmer	technical
2020	Principle Scientific Programmer	technical
1094	Software Engineer	technical
1074	Software Design Specialist	technical
7010	Computer Specialist	technical
3015	Engineer	technical
3018	Senior Engineer	technical
1016	Software Engineer	technical
1090	System Engineering Specialist	technical
3001	Junior Programmer	entry_level
3028	Digital Signal Processing	technical
3030	*	*
3031	R&D Engineer	technical
3032	Engineer	technical
5067	*	*
3049	Junior Programmer	entry_level
3051	Advanced Development Engineer	technical
3052	Engineering Specialist	technical
3054	*	*
3042	R&D Engineer	technical
3063	Programmer Analyst	technical
3065	Programmer	technical
3068	R&D Engineer	technical
3069	Senior Engineer	technical
3072	Software Engineer	technical
3074	Engineer	technical
3011	Software R&D Engineer	technical
3010	Communications Software	technical
3007	Advanced R&D Engineer	technical
3006	Supervisor	middle_mgmt
3005	Senior Software Engineer	technical
3087	R&D Engineer	technical
3089	Research Engineer	technical
3091	Software Engineer	technical
3093	Engineering Specialist	technical
2011	Data Processing Consultant	technical
3109	R&D Engineer	technical
3111	Senior Engineer	technical
2002	Programmer	technical
2001	Senior Programmer	technical
1100	Engineer	technical
3117	Engineering Specialist	technical
3119	R&D Engineer	technical
1084	Software Design Specialist	technical
3122	*	*
1075	Software Design Specialist	technical
4004	Programmer Analyst	technical
1072	*	*
4039	Consultant	technical

1029	Software Design Specialist	technical
4008	*	*
1014	Senior Software Engineer	technical
4016	Senior Programmer Analyst	technical
1004	Engineering Software Supervisor	middle_mgmt
4024	Programmer Analyst	technical
4027	Senior GS Analyst	technical
4028	Principle Programmer Analyst	technical
4032	Analyst	technical
4034	Government Program Analyst	technical
4035	Government Program Analyst	technical
4036	Principle Programmer Analyst	technical
8004	Team Leader	middle_mgmt
7011	Computer Specialist	technical
4043	*	*
4002	Programmer	technical
4048	Principle Programmer Analyst	technical
4051	Firmware Design Engineer	technical
4056	System Analyst	technical
3112	Consultant	technical
4060	Principle Engineer	technical
4062	Senior Programmer Analyst	technical
3094	Software Engineer	technical
4065	Programmer	technical
4067	Programmer Analyst	technical
1091	System Design Specialist	technical
4069	Associate Programmer Analyst	entry_level
4070	Programmer	technical
4073	Programmer Analyst	technical
4075	Senior Engineer	technical
4076	Quality Assurance Engineer	technical
5001	*	*
5002	*	*
5004	*	*
5005	*	*
5007	*	*
5008	Diagnostic Software	technical
5011	*	*
5013	*	*
5016	*	*
5021	*	*
5024	*	*
5026	*	*
3023	*	*
5032	*	*
5036	*	*
5037	*	*
5038	*	*
5043	*	*
5045	*	*
5047	*	*
3004	Research Engineer	technical
5049	*	*
5052	*	*

5053	*	*
5057	*	*
2013	Principle Scientific Programmer	technical
5063	*	*
5080	*	*
5068	*	*
5069	*	*
5070	*	*
5071	*	*
5072	*	*
9005	Electrical Engineer	technical
5078	*	*
1052	Senior Engineering Specialist	technical
6004	Project Leader	middle_mgmt
1079	Senior Software Engineer	technical
1078	Senior Engineer	technical
6009	Electronic Technician	entry_level
6011	Electrical Engineer	technical
7001	Computer Specialist	technical
1064	*	*
1060	Software Engineer	technical
9004	Electrical Engineer	technical
9006	Electrical Engineer	technical

cluster7

6010	Electrical Engineer	technical
2009	Project Supervisor	middle_mgmt
1031	Software Design Specialist	technical
4042	Principle Systems Analyst	technical
4006	Senior Applications Analyst	technical
4068	Consultant	technical

Job Titles for the Support Clusters

Respondent	Job Title	Position in Hierarchy
------------	-----------	-----------------------

cluster1

1007	Senior Software Engineer	technical
1010	Software Engineer	technical
1015	Software Engineer	technical
1017	Software Engineer	technical
1026	Software Design Specialist	technical
1040	Software Design Specialist	technical
5009	*	*
5010	*	*
1044	Software Engineer	technical
4009	Consultant	technical
6008	Computer Scientist	technical
1048	Software Engineering Specialist	technical
1049	Software Design Specialist	technical
5077	*	*
5075	*	*
1055	*	*
5073	*	*
1058	Associate Engineer	entry_level
3104	Software Engineer	technical
5065	*	*
1077	Electrical Engineer	technical
5064	*	*
5062	*	*
2003	Scientific Programmer	technical
5056	*	*
5044	Programmer	technical
3003	Software Engineer	technical
5042	*	*
3012	Senior Engineer	technical
3014	Software Engineer	technical
3020	R&D Engineer	technical
3027	Engineering Specialist	technical
3033	Senior Engineer	technical
3035	Senior Engineer	technical
5041	Programmer	technical
3037	Software Engineer	technical
5040	*	*
5033	*	*
3043	Senior Engineer	technical
5028	*	*
3045	*	*
3046	Software Engineer	technical
3047	*	*
3048	Software Engineer	technical
3050	R&D Engineer	technical
3053	Senior Software Engineer	technical
3059	Senior Engineer	technical

3064	Software Engineer	technical
3066	Senior Software Engineer	technical
3067	Software Engineer	technical
5027	*	*
3073	Software Engineer	technical
3075	Software Engineer	technical
3076	*	*
5025	*	*
3084	Software Engineer	technical
5022	*	*
3092	Senior Engineer	technical
3099	Advanced Research Engineer	technical
4040	System Programming Analyst	technical
3106	*	*
3107	Engineer	technical
3110	Senior Software Engineer	technical
3115	Software Engineer	technical
3116	Engineer	technical
3118	Engineer	technical
3121	Software Engineer	technical
5020	*	*
5019	*	*
4071	Programmer	technical
4059	Programmer Analyst	technical
4022	Programmer	technical
4031	Principle Programmer	technical
4053	Programmer Analyst	technical

cluster14

7003	Programmer	technical
7007	Computer Specialist	technical
7013	Computer Specialist	technical
1065	Engineering Manager	high_level_mgmt
4021	Associate Applications Analyst	entry_level
4064	Manager, Software Development	middle_mgmt
4054	Programmer Analyst	technical
7015	Computer Specialist	technical
7014	Computer Specialist	technical
5066	*	*
1053	Software System Engineer	technical
1050	Software Design Specialist	technical
3071	Senior System Engineer	technical

cluster7

4057	Programmer	technical
4019	Analyst	technical
4074	*	*
4018	Electrical Engineer	technical
4012	Principle Programmer Analyst	technical
4001	System Engineer	technical
3088	Senior Engineering Specialist	technical
3081	Software Engineer	technical
5015	*	*
5014	*	*
4013	Software System Programmer	technical
5039	*	*
5006	*	*
3036	*	*
3008	Programming Aide	entry_level
2023	Associate Scientific Programmer	entry_level
5046	*	*
5050	*	*
5055	*	*
2017	Senior System Analyst	technical
5059	*	*
1096	Software Design Specialist	technical
1087	Software Engineering Specialist	technical
1076	Software Engineer	technical
3038	Software Engineer	technical
1057	Principle Engineer	technical
3039	*	*
3044	Software Engineer	technical
5079	*	*
6003	General Engineer	technical
6006	Software Quality Assurance	technical
1046	Software Engineer	technical
7002	Computer Specialist	technical
1045	Software Engineer	technical
1043	Senior Dynamics Engineer	technical
7012	Computer Specialist	technical
1041	Software Design Specialist	technical
5003	*	*
4077	Engineer	technical
8001	Computer Systems Analyst	technical
8002	Software System Analyst	technical
8005	Computer Specialist	technical
9003	Mathematician	technical

Support Clusters

counts

	years of involvement				
	less than two years		two to five years	five to ten years	over ten years
cluster1	13.000	32.000	14.000	14.000	73.000
cluster7	7.000	10.000	8.000	18.000	43.000
cluster14	2.000	2.000	2.000	7.000	13.000
	22.000	44.000	24.000	39.000	129.000

counts row pct

	years of involvement				
	less than two years		two to five years	five to ten years	over ten years
cluster1	17.808	43.836	19.178	19.178	100.000
cluster7	16.279	23.256	18.605	41.860	100.000
cluster14	15.385	15.385	15.385	53.846	100.000
	17.054	34.109	18.605	30.233	100.000

counts column pct

	years of involvement				
	less than two years		two to five years	five to ten years	over ten years
cluster1	59.091	72.727	58.333	35.897	56.589
cluster7	31.818	22.727	33.333	46.154	33.333
cluster14	9.091	4.545	8.333	17.949	10.078
	100.000	100.000	100.000	100.000	100.000

counts table pct

	years of involvement				
	less than two years		two to five years	five to ten years	over ten years

cluster1	10.078	24.806	10.853	10.853	56.589
cluster7	5.426	7.752	6.202	13.953	33.333
cluster14	1.550	1.550	1.550	5.426	10.078
	17.054	34.109	18.605	30.233	100.000

Development Clusters

counts

years of involvement					
	five to ten years				
	two to five years				
	less than two years		over ten years		
cluster1	1.000	4.000	0.000	4.000	9.000
cluster3	2.000	7.000	3.000	5.000	17.000
cluster4	10.000	36.000	29.000	54.000	129.000
cluster7	0.000	0.000	1.000	5.000	6.000
cluster153	0.000	0.000	1.000	4.000	5.000
	13.000	47.000	34.000	72.000	166.000

counts row pct

years of involvement					
	five to ten years				
	two to five years				
	less than two years		over ten years		
d cluster					
cluster1	11.111	44.444	0.000	44.444	100.000
cluster3	11.765	41.176	17.647	29.412	100.000
cluster4	7.752	27.907	22.481	41.860	100.000
cluster7	0.000	0.000	16.667	83.333	100.000
cluster153	0.000	0.000	20.000	80.000	100.000
	7.831	28.313	20.482	43.373	100.000

counts column pct

years of involvement					
	five to ten years				
	two to five years				
	less than two years		over ten years		
d cluster					
cluster1	7.692	8.511	0.000	5.556	5.422
cluster3	15.385	14.894	8.824	6.944	10.241
cluster4	76.923	76.596	85.294	75.000	77.711
cluster7	0.000	0.000	2.941	6.944	3.614
cluster153	0.000	0.000	2.941	5.556	3.012
	100.000	100.000	100.000	100.000	100.000

counts table pct

9w

	years of involvement				
	two to five years		five to ten years		
	less than two years		over ten years		
cluster1	0.602	2.410	0.000	2.410	5.422
cluster3	1.205	4.217	1.807	3.012	10.241
cluster4	6.024	21.687	17.470	32.530	77.711
cluster7	0.000	0.000	0.602	3.012	3.614
cluster153	0.000	0.000	0.602	2.410	3.012
	7.831	28.313	20.482	43.373	100.000

Development Clusters

	cluster1	cluster3 cluster4	cluster7 cluster153			
language						
JOVIAL	2	3	25	0	1	31
CMS_2	2	6	44	1	1	54
C	0	1	15	0	0	16
FORTRAN	8	15	123	6	5	157
COBOL	6	8	68	3	4	89
ASSEMBLER	7	16	122	5	5	155
PLI	1	6	67	2	3	79
PASCAL	5	13	64	3	1	86
BASIC	7	10	82	4	3	106
ALGOL	3	2	29	1	1	36
RATFOR_WATFOR_WATFIV	2	5	33	1	2	43
MODULA	0	0	3	0	0	3
SIMULA	0	1	3	0	0	4
XPL	0	0	4	1	0	5
MMP	0	0	0	0	0	0
FORTH	1	0	8	0	0	9
Ada	1	10	28	2	1	42
LISP	2	4	26	0	0	32
SNOBOL	2	4	25	0	0	31
ECL	0	0	0	1	0	1
GPSS	1	2	15	0	1	19
SAS	0	0	1	0	0	1
PROTEGE	0	0	0	0	0	0
PPL	0	0	0	0	0	0
APL	4	4	32	1	1	42
Other	2	3	36	1	1	43
	56	113	853	32	30	1084

Support Clusters

language	cluster7			
	cluster1	cluster14		
JOVIAL	12	6	1	19
CMS_2	21	14	2	37
C	4	2	2	8
FORTRAN	68	40	11	119
COBOL	31	20	6	57
ASSEMBLER	63	38	7	108
PLI	44	17	4	65
PASCAL	41	16	4	61
BASIC	44	27	8	79
ALGOL	17	4	2	23
RATFOR_WATFOR_WATFIV	21	8	2	31
MODULA	1	1	0	2
SIMULA	2	0	0	2
XPL	2	0	0	2
MMP	0	0	0	0
FORTH	6	4	1	11
Ada	11	14	3	28
LISP	12	4	1	17
SNOBOL	15	6	1	22
ECL	1	0	0	1
GPSS	9	7	2	18
SAS	2	2	0	4
PROTEGE	1	0	0	1
PPL	0	0	0	0
APL	25	12	1	38
Other	13	7	6	26
	466	249	64	779

Table of Development Clusters versus Methodology versus Knowledge

Cluster:
cluster1

methodology	knowledge			
	know concept		used frequently	
	heard of	used moderately		
PSL PLA	0	0	2	0
SADT	0	1	1	0
SREM	0	1	0	0
HIPO	0	1	4	0
Jackson Design	0	1	1	0
Structured Design	0	0	0	8
Warnier Orr Design	0	1	1	0
N S Chapin Chart	3	2	0	0
Beamson Tables	0	1	0	0
Program Design language	0	1	2	6
Structured Programming	0	0	0	9
Structured Walkthroughs	0	1	0	8
Top Down Design	0	2	1	6
Top Down Testing	0	1	3	4
Bottom Up Design	0	4	1	3
Bachman Diagramming	1	2	0	0
Entity Diagrams	0	1	0	0
Data Abstraction	0	1	0	1
other methodology	0	0	0	1
enumeration types	0	2	0	2
floating point types	0	2	1	6
fixed point types	0	1	2	6
user defined types	0	1	1	6
pointers	0	0	0	9
typed pointers	0	1	2	4
ranges	0	1	1	5
records	0	1	0	8
variant records	0	1	2	4
object type decls	0	1	3	3
global variables	0	0	0	9
local variables	0	0	0	9
formal actual params	0	0	0	8
reserved words	0	0	1	8
blocks	0	0	0	9
case statements	0	0	1	8
if then else statements	0	0	1	8
loop for while until	0	0	1	8
exit statements	0	2	1	5
procedures	0	0	1	7
functions	0	0	1	8
return statements	0	1	0	8

clusters modules package	0	- 1	0	7
stubs	0	- 0	1	7
goto statements	0	1	1	7
comments	0	0	0	9
exception handlers	0	- 1	0	7
task coroutines	0	2	2	5
other prog constructs	0	0	0	0
importing exporting name	0	1	0	2
data encapsulation	0	2	1	2
name scoping	1	1	0	2
name visibility	1	1	1	2
static dynamic nesting	1	1	3	3
iteration	0	0	2	6
conditional statements	0	0	1	8
recursion	0	0	5	3
concurrency	0	2	3	2
strong typing	0	1	1	3
type conversion	0	2	1	3
data abstraction	1	2	1	2
generics	0	2	0	2
loop invariants	2	1	0	3
parameter binding	0	1	2	1
version number	0	0	0	5
other prog concepts	0	0	0	0
A enumeration types	0	2	0	1
Ada user defined types	0	3	0	2
Ada subtypes	0	3	0	2
Ada derived types	0	3	0	1
Ada real types	0	3	1	1
Ada float point types	0	4	0	1
Ada fixed pt types	0	4	1	1
Ada record types	0	3	0	2
Ada rec types discrim	0	2	0	1
Ada slices	1	1	0	1
Ada aggregates	1	1	0	2
Ada allocators	0	0	1	1
Ada access types	0	1	0	1
Ada overloading	0	2	0	1
Ada packages	1	2	0	1
Ada private types	0	3	0	1
Ada scope	0	2	0	2
Ada short circuiting	1	1	0	1
Ada visibility	1	1	0	1
Ada tasking	0	3	1	1
Ada task types	0	2	0	1
Ada rendezvous	0	2	0	1
Ada entries	0	2	1	2
Ada entry families	0	1	0	1
Ada separate compilation	0	3	1	1
Ada exceptions	0	3	0	1
Ada generic prog units	2	2	0	1
Ada instantiation	1	0	0	1
Ada elaboration	1	0	0	1
Ada context spec	0	1	0	1

Ada information hiding	1	1	0	1
Ada mutual recursion	0	1	0	1
other Ada concepts	0	0	0	0

cluster3

knowledge

	know concept	heard of	used frequently	used moderately
methodology				
PSL PLA	1	5	2	0
SADT	0	3	0	0
SREM	0	2	0	0
HIPO	3	8	1	1
Jackson Design	1	3	0	0
Structured Design	0	4	8	5
Warnier Orr Design	0	1	2	0
N S Chapin Chart	2	1	1	0
Beamson Tables	0	0	0	0
Program Design language	1	3	8	5
Structured Programming	0	1	8	8
Structured Walkthroughs	0	4	8	5
Top Down Design	0	1	8	8
Top Down Testing	0	4	8	5
Bottom Up Design	0	8	7	0
Bachman Diagramming	1	1	0	0
Entity Diagrams	0	2	2	0
Data Abstraction	3	2	5	0
other methodology	0	0	2	0
enumeration types	1	3	7	1
floating point types	0	6	6	5
fixed point types	0	4	5	8
user defined types	1	3	9	3
pointers	0	3	7	6
typed pointers	3	2	5	5
ranges	0	4	9	3
records	0	4	7	6
variant records	2	3	6	3
object type decls	0	0	10	4
global variables	0	1	5	11
local variables	0	1	5	11
formal actual params	0	2	5	8
reserved words	0	2	5	9
blocks	0	3	4	9
case statements	0	1	6	9
if then else statements	0	0	7	9
loop for while until	0	0	7	9
exit statements	0	2	9	5
procedures	0	0	5	11
functions	0	0	7	9

return statements	0	0	5	11
clusters modules package	0	3	6	7
stubs	1	6	6	3
goto statements	0	4	8	4
comments	0	0	6	10
exception handlers	0	7	5	3
task coroutines	0	8	4	3
other prog constructs	0	1	1	0
importing exporting name	0	4	4	0
data encapsulation	2	3	4	2
name scoping	2	3	3	4
name visibility	1	4	3	2
static dynamic nesting	3	4	6	1
iteration	0	4	7	6
conditional statements	0	1	5	11
recursion	0	7	7	3
concurrency	1	10	4	0
strong typing	3	6	3	2
type conversion	1	4	6	2
data abstraction	3	5	4	1
generics	0	7	3	0
loop invariants	3	4	3	1
parameter binding	3	3	4	3
version number	1	4	7	3
other prog concepts	0	0	0	0
A enumeration types	2	8	1	1
Ada user defined types	2	10	1	1
Ada subtypes	3	9	1	0
Ada derived types	4	8	0	0
Ada real types	3	10	0	1
Ada float point types	2	11	0	1
Ada fixed pt types	2	11	0	1
Ada record types	4	8	1	1
Ada rec types discrim	3	8	0	0
Ada slices	3	6	0	0
Ada aggregates	3	6	0	0
Ada allocators	4	5	0	0
Ada access types	2	6	1	0
Ada overloading	6	7	0	0
Ada packages	3	8	1	0
Ada private types	4	7	1	0
Ada scope	2	8	0	1
Ada short circuiting	2	5	0	0
Ada visibility	3	6	0	1
Ada tasking	5	8	1	0
Ada task types	2	7	1	0
Ada rendezvous	2	8	0	0
Ada entries	2	7	0	0
Ada entry families	2	3	0	0
Ada separate compilation	4	11	0	0
Ada exceptions	4	8	1	0
Ada generic prog units	4	5	1	0
Ada instantiation	4	7	1	0
Ada elaboration	3	3	0	0

Ada context spec	3	4	0	0
Ada information hiding	2	10	0	0
Ada mutual recursion	5	5	0	0
other Ada concepts	0	0	0	0

cluster4

knowledge

	know concept heard of	used moderately	used frequently	
methodology				
PSL PLA	16	20	7	0
SADT	5	3	7	0
SREM	3	8	1	1
HIPO	16	44	26	9
Jackson Design	11	10	5	6
Structured Design	5	13	34	71
Warnier Orr Design	13	14	2	3
N S Chapin Chart	13	12	6	1
Beamson Tables	3	1	1	0
Program Design language	5	32	25	51
Structured Programming	0	9	30	90
Structured Walkthroughs	4	36	42	37
Top Down Design	1	8	32	86
Top Down Testing	4	28	37	55
Bottom Up Design	6	53	39	19
Bachman Diagramming	6	7	1	1
Entity Diagrams	7	6	0	1
Data Abstraction	14	34	15	10
other methodology	0	0	0	3
enumeration types	9	26	17	18
floating point types	3	22	40	61
fixed point types	2	11	27	83
user defined types	6	28	34	43
pointers	2	16	31	74
typed pointers	10	26	20	34
ranges	5	27	28	49
records	2	19	34	70
variant records	12	29	21	28
object type decls	5	27	29	47
global variables	2	7	23	96
local variables	2	9	20	97
formal actual params	7	7	25	70
reserved words	3	17	25	82
blocks	1	17	29	70
case statements	4	12	30	77
if then else statements	1	7	23	98
loop for while until	1	9	26	93
exit statements	1	27	35	64
procedures	2	8	25	94

functions	1	10	31	87
return statements	2	9	21	95
clusters modules package	6	25	32	48
stubs	3	19	33	57
goto statements	0	22	36	70
comments	0	6	17	105
exception handlers	6	33	29	47
task coroutines	9	36	27	41
other prog constructs	0	0	0	0
importing exporting name	12	11	5	9
data encapsulation	17	26	21	20
name scoping	14	19	9	25
name visibility	14	23	6	21
static dynamic nesting	22	24	15	23
iteration	3	14	33	75
conditional statements	3	5	25	95
recursion	1	35	45	37
concurrency	7	42	28	27
strong typing	9	29	17	19
type conversion	9	27	30	36
data abstraction	20	36	15	21
generics	15	28	15	11
loop invariants	19	23	16	13
parameter binding	19	26	14	11
version number	8	20	27	45
other prog concepts	0	0	0	2
A enumeration types	15	33	7	7
Ada user defined types	14	45	8	12
Ada subtypes	20	31	7	5
Ada derived types	23	22	7	3
Ada real types	9	47	13	17
Ada float point types	7	51	13	20
Ada fixed pt types	5	50	9	24
Ada record types	12	41	11	17
Ada rec types discrim	20	24	3	4
Ada slices	13	15	6	4
Ada aggregates	15	16	2	3
Ada allocators	14	19	2	3
Ada access types	15	27	3	6
Ada overloading	12	24	3	3
Ada packages	10	37	6	4
Ada private types	10	33	5	2
Ada scope	9	37	4	6
Ada short circuiting	10	17	2	1
Ada visibility	13	33	6	3
Ada tasking	17	40	5	11
Ada task types	18	31	3	5
Ada rendezvous	5	31	2	2
Ada entries	8	32	3	9
Ada entry families	14	12	2	1
Ada separate compilation	7	49	6	16
Ada exceptions	10	38	6	10
Ada generic prog units	12	26	5	2
Ada instantiation	8	19	1	3

Ada elaboration	9	18	2	2
Ada context spec	14	23	1	3
Ada information hiding	10	38	3	4
Ada mutual recursion	13	22	2	2
other Ada concepts	7	0	0	0

cluster7

knowledge

	heard of	know concept	used moderately	used frequently
methodology				
PSL PLA	1	2	0	0
SADT	0	0	0	0
SREM	0	0	0	0
HIPO	1	2	2	0
Jackson Design	0	0	1	0
Structured Design	0	0	3	2
Warnier Orr Design	1	0	0	0
N S Chapin Chart	0	0	0	0
Beamson Tables	0	0	0	0
Program Design language	0	4	2	0
Structured Programming	0	0	1	2
Structured Walkthroughs	0	0	3	0
Top Down Design	0	0	2	4
Top Down Testing	0	2	0	4
Bottom Up Design	0	3	0	3
Bachman Diagramming	0	0	0	1
Entity Diagrams	0	0	0	1
Data Abstraction	1	0	0	1
other methodology	0	0	0	0
enumeration types	1	0	0	2
floating point types	1	0	0	4
fixed point types	1	0	1	5
user defined types	0	1	1	2
pointers	0	1	1	4
typed pointers	1	2	1	2
records	0	0	1	4
variant records	1	0	1	4
object type decls	0	1	0	1
global variables	1	0	0	5
local variables	0	0	0	6
formal actual params	0	0	0	6
reserved words	0	0	1	4
blocks	0	0	1	5
case statements	0	1	1	4
if then else statements	0	1	2	3
loop for while until	0	0	2	3
exit statements	0	0	2	3

procedures	0	1	2	3
functions	0	0	2	4
return statements	0	0	1	5
clusters modules package	1	1	2	2
stubs	2	0	1	3
goto statements	0	1	1	4
comments	0	0	0	5
exception handlers	0	0	4	2
task coroutines	1	0	3	2
other prog constructs	0	0	0	0
importing exporting name	1	0	1	1
data encapsulation	1	0	1	2
name scoping	1	0	1	1
name visibility	1	0	1	1
static dynamic nesting	1	1	0	2
iteration	0	1	0	5
conditional statements	0	0	2	4
recursion	0	2	3	1
concurrency	1	1	2	0
strong typing	0	2	0	1
type conversion	0	3	1	1
data abstraction	1	1	1	1
generics	1	1	1	0
loop invariants	0	1	1	2
parameter binding	1	0	0	1
version number	0	1	3	2
other prog concepts	0	0	0	1
A enumeration types	0	1	1	0
Ada user defined types	0	2	1	1
Ada subtypes	0	1	1	1
Ada derived types	0	2	1	0
Ada real types	0	2	1	2
Ada float point types	0	2	1	2
Ada fixed pt types	0	3	1	1
Ada record types	0	2	2	0
Ada rec types discrim	1	1	0	0
Ada slices	1	1	0	0
Ada aggregates	1	0	1	0
Ada allocators	1	0	1	1
Ada access types	1	0	2	0
Ada overloading	1	0	0	1
Ada packages	1	0	0	1
Ada private types	0	1	1	0
Ada scope	1	0	0	1
Ada short circuiting	1	1	0	0
Ada visibility	1	1	0	1
Ada tasking	1	2	0	1
Ada task types	0	3	0	1
Ada rendezvous	1	1	0	0
Ada entries	1	2	0	0
Ada entry families	1	1	0	0
Ada separate compilation	1	2	1	0
Ada exceptions	0	2	0	0
Ada generic prog units	1	2	0	0

Ada instantiation	1	1	0	0
Ada elaboration	1	1	0	0
Ada context spec	1	0	1	0
Ada information hiding	1	0	1	0
Ada mutual recursion	1	1	0	0
other Ada concepts	0	0	0	0

cluster153

knowledge

	know concept	heard of	used moderately	used frequently
methodology				
PSL PLA	0	1	0	0
SADT	0	0	0	0
SREM	0	1	0	0
HIPO	0	3	0	0
Jackson Design	1	0	0	0
Structured Design	0	1	3	1
Warnier Orr Design	0	0	1	1
N S Chapin Chart	0	0	1	0
Beamson Tables	0	0	0	0
Program Design language	1	1	0	1
Structured Programming	0	1	2	3
Structured Walkthroughs	0	1	4	1
Top Down Design	0	1	3	2
Top Down Testing	0	1	4	1
Bottom Up Design	0	1	0	2
Bachman Diagramming	0	0	0	0
Entity Diagrams	0	0	0	0
Data Abstraction	1	0	1	0
other methodology	0	1	0	0
enumeration types	0	1	2	0
floating point types	0	0	2	2
fixed point types	0	0	1	3
user defined types	0	1	1	2
pointers	0	0	0	2
typed pointers	1	0	1	1
ranges	0	0	2	2
records	0	0	2	3
variant records	0	0	2	2
object type decls	0	1	3	1
global variables	0	1	2	2
local variables	0	1	2	2
formal actual params	0	1	1	2
reserved words	0	0	3	2
blocks	0	0	3	1
case statements	0	0	2	3
if then else statements	0	0	0	5
loop for while until	0	1	1	3

exit statements	0	1	0	4
procedures	0	0	0	5
functions	0	0	0	5
return statements	0	0	0	5
clusters modules package	1	0	0	2
stubs	0	0	2	1
goto statements	0	0	0	5
comments	0	0	0	5
exception handlers	0	0	3	0
task coroutines	0	0	4	0
other prog constructs	0	0	0	0
importing exporting name	1	0	0	1
data encapsulation	0	0	3	0
name scoping	0	2	0	0
name visibility	0	1	1	0
static dynamic nesting	0	0	2	1
iteration	0	0	0	2
conditional statements	0	0	0	3
recursion	0	0	0	2
concurrency	0	0	1	1
strong typing	0	0	1	1
type conversion	0	0	0	2
data abstraction	1	0	1	0
generics	1	1	0	0
loop invariants	0	0	0	2
parameter binding	0	1	1	0
version number	0	0	0	3
other prog concepts	0	0	0	0
A enumeration types	0	1	1	0
Ada user defined types	0	1	1	0
Ada subtypes	0	1	0	1
Ada derived types	0	1	0	1
Ada real types	0	1	0	1
Ada float point types	0	2	0	1
Ada fixed pt types	0	2	0	1
Ada record types	0	2	0	1
Ada rec types discrim	0	1	1	0
Ada slices	0	1	0	1
Ada aggregates	0	0	1	0
Ada allocators	0	0	1	0
Ada access types	1	0	1	0
Ada overloading	0	2	0	1
Ada packages	0	1	1	1
Ada private types	1	0	1	0
Ada scope	0	2	0	0
Ada short circuiting	0	0	1	0
Ada visibility	1	1	0	0
Ada tasking	1	1	1	0
Ada task types	1	1	1	0
Ada rendezvous	0	1	1	0
Ada entries	0	2	1	0
Ada entry families	0	2	0	0
Ada separate compilation	0	2	0	1
Ada exceptions	0	0	0	1

Ada generic prog units	1	1	0	0
Ada instantiation	0	2	0	0
Ada elaboration	0	1	1	0
Ada context spec	1	0	1	0
Ada information hiding	1	0	1	0
Ada mutual recursion	0	0	1	0
other Ada concepts	0	0	0	0

Support

Cluster:
cluster1

	knowledge			
	used moderately		know concept	
	heard of	used frequently		
methodology	7	14	1	0
PSL PLA	6	3	0	0
SADT	5	1	0	0
SREM	6	19	12	3
HIPO	4	3	2	1
Jackson Design	2	10	19	36
Structured Design	6	6	3	0
Warnier Orr Design	3	7	3	1
N S Chapin Chart	0	0	0	0
Beamson Tables	7	15	12	29
Program Design language	2	5	10	56
Structured Programming	8	15	26	11
Structured Walkthroughs	1	9	18	43
Top Down Design	3	20	24	17
Top Down Testing	3	32	12	11
Bottom Up Design	1	2	0	1
Bachman Diagramming	0	2	0	1
Entity Diagrams	6	8	5	4
Data Abstraction	0	0	0	1
other methodology	5	11	9	11
enumeration types	1	10	19	40
floating point types	0	5	18	45
fixed point types	2	16	13	30
user defined types	1	9	16	46
pointers	5	16	10	16
typed pointers	4	17	15	26
ranges	2	13	19	35
records	5	17	11	14
variant records	2	13	18	26
object type dcls	0	3	11	59
global variables	0	2	10	61
local variables	3	7	14	38
formal actual params	1	11	10	44
reserved words	2	11	12	42
blocks	2	9	11	48
case statements	0	6	9	58
if then else statements	0	6	10	57
loop for while until	0	16	17	39
exit statements	1	2	14	53
procedures	0	4	15	52
functions	0	3	12	57
return statements	4	10	18	25
clusters modules package	1	18	13	20
stubs				

goto statements	0	18	23	32
comments	0	0	7	66
exception handlers	3	18	12	21
task coroutines	2	24	11	15
other prog constructs	0	0	0	1
importing exporting name	3	12	2	6
data encapsulation	9	16	8	7
name scoping	1	10	8	8
name visibility	1	9	6	8
static dynamic nesting	2	17	7	11
iteration	1	5	19	39
conditional statements	1	1	12	53
recursion	3	16	25	20
concurrency	5	23	10	9
strong typing	4	11	10	9
type conversion	3	14	16	19
data abstraction	9	18	3	9
generics	9	14	4	3
loop invariants	9	14	6	5
parameter binding	7	16	3	7
version number	1	14	8	25
other prog concepts	0	0	0	1
Ada enumeration types	3	12	3	3
Ada user defined types	4	16	6	6
Ada subtypes	5	12	2	4
Ada derived types	6	10	2	1
Ada real types	1	20	4	11
Ada float point types	1	21	6	9
Ada fixed pt types	1	20	5	9
Ada record types	3	19	5	9
Ada rec types discrim	4	7	5	2
Ada slices	3	2	2	1
Ada aggregates	5	7	3	1
Ada allocators	2	9	2	1
Ada access types	2	7	4	2
Ada overloading	0	10	3	0
Ada packages	3	10	3	3
Ada private types	3	11	3	1
Ada scope	2	15	2	5
Ada short circuiting	3	4	1	1
Ada visibility	2	11	1	4
Ada tasking	4	19	1	3
Ada task types	6	13	2	0
Ada rendezvous	4	7	1	1
Ada entries	3	12	2	2
Ada entry families	6	4	1	1
Ada separate compilation	1	20	4	4
Ada exceptions	1	14	5	3
Ada generic prog units	3	10	2	1
Ada instantiation	3	6	2	1
Ada elaboration	3	2	1	2
Ada context spec	5	1	1	0
Ada information hiding	2	13	2	3
Ada mutual recursion	3	4	0	1

other Ada concepts	1	0	0	0
--------------------	---	---	---	---

cluster7

knowledge

		used moderately	
	know concept		
heard of		used frequently	

methodology				
PSL PLA	6	11	2	0
SADT	2	2	4	0
SREM	3	3	1	0
HIPO	7	9	10	2
Jackson Design	3	2	0	0
Structured Design	1	8	14	13
Warnier Orr Design	4	3	1	0
N S Chapin Chart	2	2	0	0
Beamson Tables	1	1	0	0
Program Design language	7	11	12	5
Structured Programming	0	9	16	18
Structured Walkthroughs	2	14	10	11
Top Down Design	0	6	20	16
Top Down Testing	1	17	10	9
Bottom Up Design	4	18	5	4
Bachman Diagramming	1	3	0	0
Entity Diagrams	0	2	1	0
Data Abstraction	2	7	6	0
other methodology	0	0	0	0
enumeration types	5	4	9	5
floating point types	1	9	21	11
fixed point types	1	6	19	14
user defined types	5	7	12	7
pointers	1	7	17	13
typed pointers	7	11	6	6
ranges	5	8	14	7
records	2	8	15	15
variant records	3	9	8	6
object type dcls	5	10	13	10
global variables	1	3	20	18
local variables	1	3	19	20
formal actual params	4	9	9	12
reserved words	0	7	14	16
blocks	0	9	18	9
case statements	2	4	17	15
if then else statements	0	2	16	24
loop for while until	0	5	16	20
exit statements	0	5	19	15
procedures	1	6	17	17
functions	1	4	18	17
return statements	0	2	17	24
clusters modules package	6	7	17	6

stubs	2	7	13	- 8
goto statements	0	7	19	-16
comments	0	1	14	27
exception handlers	4	10	11	8
task coroutines	3	11	8	4
other prog constructs	0	1	0	0
importing exporting name	6	4	2	1
data encapsulation	2	10	5	2
name scoping	6	8	3	0
name visibility	4	7	3	0
static dynamic nesting	9	12	8	1
iteration	3	5	17	12
conditional statements	0	4	14	23
recursion	3	11	13	9
concurrency	7	12	10	2
strong typing	3	8	6	5
type conversion	3	7	11	5
data abstraction	5	9	6	1
generics	4	9	5	0
loop invariants	4	9	4	1
parameter binding	6	10	2	1
version number	2	10	7	5
other prog concepts	0	0	0	0
A enumeration types	6	6	4	0
Ada user defined types	4	11	6	0
Ada subtypes	5	7	2	0
Ada derived types	5	4	3	0
Ada real types	4	14	3	3
Ada float point types	3	16	3	3
Ada fixed pt types	3	14	4	3
Ada record types	3	12	3	3
Ada rec types discrim	5	4	2	1
Ada slices	3	6	1	0
Ada aggregates	3	8	1	0
Ada allocators	2	8	1	0
Ada access types	4	7	1	0
Ada overloading	4	9	1	0
Ada packages	6	6	3	1
Ada private types	5	8	1	0
Ada scope	3	7	2	0
Ada short circuiting	3	4	0	1
Ada visibility	5	5	3	0
Ada tasking	7	10	3	2
Ada task types	7	9	1	0
Ada rendezvous	3	8	2	0
Ada entities	3	8	3	0
Ada entry families	2	5	1	0
Ada separate compilation	1	13	5	2
Ada exceptions	4	12	3	0
Ada generic prog units	5	8	1	0
Ada instantiation	5	6	1	0
Ada elaboration	1	4	2	0
Ada context spec	2	8	1	0
Ada information hiding	3	10	1	1

Ada mutual recursion	6	5	1	0
other Ada concepts	0	0	0	0

cluster14

	knowledge			
		used moderately	know concept	
	heard of	used frequently		
methodology				
PSL PLA	1	3	0	0
SADT	1	2	0	0
SREM	0	1	1	0
HIPO	1	2	2	0
Jackson Design	0	1	0	0
Structured Design	0	6	4	2
Warnier Orr Design	0	1	0	0
N S Chapin Chart	0	1	0	0
Beamson Tables	0	0	0	0
Program Design language	2	5	2	1
Structured Programming	1	5	3	3
Structured Walkthroughs	0	6	2	1
Top Down Design	1	5	3	3
Top Down Testing	1	6	2	3
Bottom Up Design	1	8	1	1
Bachman Diagramming	0	1	0	0
Entity Diagrams	0	2	0	0
Data Abstraction	0	2	2	0
other methodology	0	1	0	0
enumeration types	1	2	2	3
floating point types	0	4	5	2
fixed point types	0	3	5	3
user defined types	1	3	4	2
pointers	0	4	2	4
typed pointers	1	5	2	1
ranges	0	7	2	1
records	1	5	2	2
variant records	1	3	3	1
object type dcls	1	1	4	1
global variables	2	4	3	2
local variables	1	3	4	3
formal actual params	1	3	2	3
reserved words	0	5	2	3
blocks	1	5	2	2
case statements	1	4	2	2
if then else statements	1	4	3	3
loop for while until	1	4	3	3
exit statements	1	4	3	2
procedures	2	2	4	3
functions	2	2	4	3
return statements	1	3	3	4

clusters modules package	2	1	5	2
stubs	2	4	3	1
goto statements	1	1	4	5
comments	1	1	4	5
exception handlers	2	4	2	0
task coroutines	1	2	2	2
other prog constructs	0	0	0	0
importing exporting name	1	0	0	0
data encapsulation	1	5	1	1
name scoping	1	1	2	1
name visibility	3	0	1	1
static dynamic nesting	1	3	1	3
iteration	1	1	5	4
conditional statements	1	1	5	4
recursion	0	3	4	1
concurrency	0	2	1	3
strong typing	1	3	1	0
type conversion	0	3	1	2
data abstraction	0	4	0	1
generics	0	2	1	1
loop invariants	1	0	0	2
parameter binding	1	0	0	1
version number	1	3	2	1
other prog concepts	0	0	0	0
A enumeration types	2	4	1	0
Ada user defined types	2	4	1	0
Ada subtypes	2	3	1	0
Ada derived types	2	3	1	0
Ada real types	1	5	0	0
Ada float point types	1	5	1	0
Ada fixed pt types	1	5	1	0
Ada record types	2	4	1	0
Ada rec types discrim	3	2	1	0
Ada slices	4	0	0	0
Ada aggregates	3	2	0	0
Ada allocators	3	1	0	0
Ada access types	1	5	0	0
Ada overloading	2	3	0	0
Ada packages	2	3	1	0
Ada private types	0	5	0	0
Ada scope	2	3	1	0
Ada short circuiting	2	0	0	0
Ada visibility	2	3	1	0
Ada tasking	4	2	0	0
Ada task types	4	2	0	0
Ada rendezvous	2	4	0	0
Ada entries	2	3	0	0
Ada entry families	2	1	0	0
Ada separate compilation	2	3	1	0
Ada exceptions	3	3	0	0
Ada generic prog units	3	3	0	0
Ada instantiation	3	2	1	0
Ada elaboration	2	1	0	0
Ada context spec	3	1	0	0

Ada information hiding	3	2	0	0
Ada mutual recursion	3	1	0	0
other Ada concepts	0	0	0	0

Technical Managers (Cluster 57)
Methodologies versus Knowledge

methodology	knowledge				
	know_concept		used_frequently		
	heard_of	used_moderately			
PSL_PLA	2	9	1	0	12
SADT	1	5	1	1	8
SREM	2	5	0	0	7
HIPO	5	10	9	4	28
Jackson_Design	5	2	1	1	9
Structured_Design	1	5	9	13	28
Warnier_Orr_Design	3	6	1	1	11
N_S_Chapin_Chart	5	3	1	1	10
Beamson_Tables	1	1	0	0	2
Program_Design_language	3	5	13	6	27
Structured_Programming	1	3	10	16	30
Structured_Walkthroughs	2	9	9	9	29
Top_Down_Design	0	2	10	19	31
Top_Down_Testing	0	6	8	15	29
Bottom_Up_Design	0	9	12	8	29
Bachman_Diagramming	0	2	1	0	3
Entity_Diagrams	3	2	2	0	7
Data_Abstraction	1	5	6	4	16
other_methodology	0	0	2	0	2
enumeration_types	0	5	5	4	14
floating_point_types	1	5	6	18	30
fixed_point_types	1	3	6	19	29
user_defined_types	0	5	9	14	28
pointers	0	6	8	13	27
typed_pointers	4	5	6	9	24
ranges	0	5	9	11	25
records	0	5	7	17	29
variant_records	3	2	7	10	22
object_type_objs	1	2	7	15	25
global_variables	0	3	6	20	29
local_variables	0	3	6	19	28
formal_actual_params	1	2	5	17	25
reserved_words	1	3	7	16	27
blocks	0	5	8	12	25
case_statements	1	4	8	17	30
if_then_else_statements	0	1	7	23	31
loop_for_while_until	0	1	7	22	30
exit_statements	0	3	8	19	30
procedures	0	2	5	22	29
functions	0	3	5	21	29
return_statements	0	2	5	23	30
clusters_modules_package	2	7	6	12	27
stubs	0	9	11	7	27
goto_statements	0	2	12	16	30
comments	0	1	10	19	30

exception_handlers	1	5	8	12	26
task_coroutines	4	6	7	10	27
other_prog_constructs	0	0	1	1	2
importing_exporting_name	6	2	2	2	12
data_encapsulation	7	6	5	4	22
name_scoping	2	4	6	3	15
name_visibility	5	3	4	4	16
static_dynamic_nesting	5	4	6	5	20
iteration	1	3	7	18	29
conditional_statements	1	0	6	22	29
recursion	1	5	12	10	28
concurrency	0	5	8	10	23
strong_typing	3	6	6	6	21
type_conversion	1	6	8	9	24
data_abstraction	3	7	5	7	22
generics	5	6	3	5	19
loop_invariants	2	4	6	5	17
parameter_binding	2	9	2	6	19
version_number	1	6	6	10	23
other_prog_concepts	0	0	0	0	0
Ada_enumeration_types	3	6	4	2	15
Ada_user_defined_types	2	11	1	4	18
Ada_subtypes	3	9	2	3	17
Ada_derived_types	3	7	1	1	12
Ada_real_types	1	8	5	5	19
Ada_float_point_types	1	9	4	6	20
Ada_fixed_pt_types	1	9	3	7	20
Ada_record_types	0	11	3	5	19
Ada_rec_types_discrim	4	9	0	0	13
Ada_slices	4	6	0	0	10
Ada_aggregates	3	5	0	1	9
Ada_allocators	2	7	2	0	11
Ada_access_types	3	7	1	2	13
Ada_overloading	3	7	0	1	11
Ada_packages	3	10	1	1	15
Ada_private_types	4	10	1	1	16
Ada_scope	2	12	2	0	16
Ada_short_circuiting	2	4	1	0	7
Ada_visibility	1	9	1	1	12
Ada_tasking	3	10	0	4	17
Ada_task_types	5	5	0	2	12
Ada_rendezvous	5	6	1	1	13
Ada_entries	4	7	2	1	14
Ada_entry_families	6	3	1	1	11
Ada_separate_compilation	3	5	2	6	16
Ada_exceptions	1	6	4	2	13
Ada_generic_prog_units	8	3	1	1	13
Ada_instantiation	4	5	1	0	10
Ada_elaboration	4	3	0	1	8
Ada_context_spec	4	4	0	0	8
Ada_information_hiding	5	5	1	2	13
Ada_mutual_recursion	5	4	1	1	11
other_Ada_concepts	0	0	0	0	0
	202	492	436	724	1854

Support Managers (Cluster 38)
Methodologies versus Knowledge

methodology	knowledge				
	know_concept		used_frequently		
	heard_of	used_moderately			
PSL_PLA	3	1	0	0	4
SADT	0	4	2	1	7
SREM	2	1	0	0	3
HIPO	0	6	5	1	12
Jackson_Design	3	2	0	1	6
Structured_Design	0	4	5	4	13
Warnier_Orr_Design	4	3	2	0	9
N_S_Chapin_Chart	2	0	1	1	4
Beamson_Tables	1	0	0	0	1
Program_Design_language	3	3	3	2	11
Structured_Programming	0	5	1	8	14
Structured_Walkthroughs	0	4	3	5	12
Top_Down_Design	0	5	1	8	14
Top_Down_Testing	0	7	1	6	14
Bottom_Up_Design	1	7	4	2	14
Bachman_Diagramming	2	0	0	0	2
Entity_Diagrams	1	2	0	0	3
Data_Abstraction	3	5	0	1	9
other_methodology	0	0	0	0	0
enumeration_types	1	1	2	3	7
floating_point_types	0	1	6	7	14
fixed_point_types	0	2	6	6	14
user_defined_types	0	3	4	3	10
pointers	0	3	2	7	12
typed_pointers	1	3	1	3	8
ranges	0	3	4	3	10
records	0	3	4	5	12
variant_records	2	2	0	3	7
object_type_decls	1	1	2	5	9
global_variables	0	1	6	6	13
local_variables	0	2	6	5	13
formal_actual_params	0	2	4	4	10
reserved_words	0	2	3	7	12
blocks	1	2	3	3	9
case_statements	0	1	4	7	12
if_then_else_statements	0	2	3	8	13
loop_for_while_until	0	3	2	8	13
exit_statements	1	2	3	7	13
procedures	0	3	3	7	13
functions	0	4	3	7	14
return_statements	0	3	2	9	14
clusters_modules_package	0	4	2	8	14
stubs	1	3	3	5	12
goto_statements	0	3	2	8	13

comments	0	1	3	9	13
exception_handlers	1	1	5	5	12
task_coroutines	0	3	1	6	10
other_prog_constructs	0	0	0	0	0
importing_exporting_name	2	1	0	2	5
data_encapsulation	0	4	1	4	9
name_scoping	2	3	0	3	8
name_visibility	2	2	0	2	6
static_dynamic_nesting	1	2	2	4	9
iteration	0	6	2	6	14
conditional_statements	0	4	3	6	13
recursion	2	4	2	5	13
concurrency	2	5	1	5	13
strong_typing	3	1	1	4	9
type_conversion	5	1	0	4	10
data_abstraction	1	6	1	2	10
generics	2	5	1	2	10
loop_invariants	3	1	0	5	9
parameter_binding	4	0	0	3	7
version_number	1	3	3	5	12
other_prog_concepts	0	0	1	0	1
Ada_enumeration_types	1	4	1	0	6
Ada_user_defined_types	2	8	0	1	11
Ada_subtypes	3	4	0	1	8
Ada_derived_types	2	4	0	0	6
Ada_real_types	2	4	0	2	8
Ada_float_point_types	3	6	0	2	11
Ada_fixed_pt_types	3	5	0	2	10
Ada_record_types	5	2	0	2	9
Ada_rec_types_discrim	3	2	0	0	5
Ada_slices	2	3	0	0	5
Ada_aggregates	2	2	0	1	5
Ada_allocators	1	3	0	0	4
Ada_access_types	1	4	0	1	6
Ada_overloading	1	4	1	0	6
Ada_packages	4	5	0	0	9
Ada_private_types	2	4	0	0	6
Ada_scope	2	3	0	2	7
Ada_short_circuiting	2	1	0	1	4
Ada_visibility	1	3	0	1	5
Ada_tasking	4	5	0	2	11
Ada_task_types	3	5	0	0	8
Ada_rendezvous	3	4	0	1	8
Ada_entries	5	2	0	2	9
Ada_entry_families	4	2	0	0	6
Ada_separate_compilation	2	7	0	2	11
Ada_exceptions	2	4	0	2	8
Ada_generic_prog_units	2	4	1	0	7
Ada_instantiation	2	4	0	1	7
Ada_elaboration	2	2	0	0	4
Ada_context_spec	2	2	0	0	4
Ada_information_hiding	4	4	0	1	9
Ada_mutual_recursion	3	2	0	0	5
other_Ada_concepts	0	0	0	0	0
	144	289	138	293	864

Administrative Managers (Cluster 2)
Methodologies versus Knowledge

methodology	knowledge				
	heard_of	know_concept	used_frequently	used_moderately	
PSL_PLA	5	18	0	0	23
SADT	6	10	2	0	18
SREM	5	7	0	0	12
HIPO	5	14	15	9	43
Jackson_Design	5	9	3	1	18
Structured_Design	1	12	9	23	45
Warnier_Orr_Design	6	9	4	0	19
N_S_Chapin_Chart	7	9	3	2	21
Beamson_Tables	3	1	0	0	4
Program_Design_language	3	12	11	15	41
Structured_Programming	0	5	8	33	46
Structured_Walkthroughs	0	11	10	23	44
Top_Down_Design	0	4	9	33	46
Top_Down_Testing	1	5	11	28	45
Bottom_Up_Design	0	11	22	12	45
Bachman_Diagramming	1	3	0	0	4
Entity_Diagrams	0	1	2	0	3
Data_Abstraction	5	9	4	3	22
other_methodology	0	0	1	2	3
enumeration_types	2	11	11	6	30
floating_point_types	0	4	13	30	47
fixed_point_types	0	2	9	34	45
user_defined_types	2	8	13	18	41
pointers	0	5	13	28	46
typed_pointers	1	15	8	15	39
ranges	0	13	10	22	45
records	1	8	10	25	44
variant_records	3	13	5	14	35
object_type_decls	1	9	11	18	39
global_variables	0	4	7	34	45
local_variables	0	3	6	36	45
formal_actual_params	0	1	12	31	44
reserved_words	0	4	8	33	45
blocks	0	4	12	29	45
case_statements	0	4	15	24	43
if_then_else_statements	0	3	11	32	46
loop_for_while_until	0	3	11	32	46
exit_statements	0	6	14	26	46
procedures	0	3	12	31	46
functions	0	0	11	35	46
return_statements	0	2	5	39	46
clusters_modules_package	2	7	12	22	43
stubs	1	3	13	25	42
goto_statements	0	3	15	26	44
comments	0	0	7	39	46

exception_handlers	1	11	9	20	41
task_coroutines	2	11	8	19	40
other_prog_constrcuts	0	0	0	0	0
importing_exporting_name	2	8	4	4	18
data_encapsulation	4	15	7	10	36
name_scoping	3	10	2	9	24
name_visibility	7	7	5	4	23
static_dynamic_nesting	5	11	6	13	35
iteration	1	1	10	31	43
conditional_statements	0	1	8	35	44
recursion	0	8	15	21	44
concurrency	1	8	11	16	36
strong_typing	4	12	7	12	35
type_conversion	3	6	16	17	42
data_abstraction	4	14	6	10	34
generics	3	11	6	6	26
loop_invariants	2	11	10	7	30
parameter_binding	5	12	9	8	34
version_number	1	5	11	21	38
other_prog_concepts	0	0	0	0	0
Ada_enumeration_types	2	20	2	1	25
Ada_user_defined_types	3	27	1	2	33
Ada_subtypes	6	19	0	1	26
Ada_derived_types	8	12	1	2	23
Ada_real_types	2	24	2	6	34
Ada_float_point_types	0	23	4	9	36
Ada_fixed_pt_types	0	22	6	7	35
Ada_record_types	2	23	1	3	29
Ada_rec_types_discrim	5	11	1	1	18
Ada_slices	3	7	1	1	12
Ada_aggregates	4	8	1	0	13
Ada_allocators	5	8	0	0	13
Ada_access_types	4	15	1	0	20
Ada_overloading	4	10	0	1	15
Ada_packages	4	17	1	0	22
Ada_private_types	1	15	0	1	17
Ada_scope	3	17	1	2	23
Ada_short_circuiting	4	9	1	0	14
Ada_visibility	4	11	1	0	16
Ada_tasking	3	21	2	2	28
Ada_task_types	4	17	2	1	24
Ada_rendezvous	2	13	3	1	19
Ada_entries	3	20	0	3	26
Ada_entry_families	6	9	1	1	17
Ada_separate_compilation	0	23	1	6	30
Ada_exceptions	3	18	2	1	24
Ada_generic_prog_units	2	11	1	0	14
Ada_instantiation	4	11	0	0	15
Ada_elaboration	8	7	0	0	15
Ada_context_spec	8	11	0	0	19
Ada_information_hiding	1	18	0	2	21
Ada_mutual_recursion	7	8	0	0	15
other_Ada_concepts	0	0	0	0	0
	230	935	575	1205	2945

END

DATE
FILMED

3-83

DTIC