


A11100 994851

NBS  
PUBLICATIONS

NAT'L INST. OF STAND & TECH  
  
A11106 407111

NBSIR 81-2388

# Synopsis of Interviews from a Survey of Software Tool Usage

November 1981

Sponsored by:

**U.S. DEPARTMENT OF COMMERCE**

**National Bureau of Standards**

**Center for Programming Science and Technology**

**Institute for Computer Sciences and Technology**

**Washington, DC 20234**

QC

100

.U56

81-2388

1981

c. 2



NATIONAL BUREAU  
OF STANDARDS  
LIBRARY

NOV 23 1981

104-2388  
104-2388  
104-2388  
104-2388  
104-2388  
104-2388

NBSIR 81-2388

## SYNOPSIS OF INTERVIEWS FROM A SURVEY OF SOFTWARE TOOL USAGE

Herbert Hecht

SoHaR Incorporated  
1040 S. LaJolla Ave.  
Los Angeles, CA 90035

November 1981

Sponsored by:  
U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
Center for Programming Science and Technology  
Institute for Computer Sciences and Technology  
Washington, DC 20234



---

**U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary***  
**NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director***



TABLE OF CONTENTS

	PAGE
1. INTRODUCTION. . . . .	1
2. CLASSIFICATION OF SURVEY PARTICIPANTS . . . . .	2
3. QUESTIONNAIRE USED IN TOOL USER INTERVIEWS. . . . .	3
4. TOOLS INTERVIEWS. . . . .	6
5. TOOLS CLASSIFICATION. . . . .	73
REFERENCES. . . . .	76



## SECTION 1

## INTRODUCTION

This report contains synopses of all interviews conducted as part of a Survey of the Development and Use of Software Tools in Industry and a classification of the tools encountered. The findings of the survey and conclusions drawn from these are presented in a companion volume to this report [1]. Notes on the scope of the survey, selection of participants, and many other data essential for proper interpretation of the material presented here are contained in the above reference, and it is recommended that the synopses and tools classification be used only in connection with the companion volume.

The survey was conducted for the National Bureau of Standards, Institute for Computer Sciences and Technology (NBS/ICST) under contract NB79SBCA0273 by SoHaR Incorporated. This report is furnished under Work Assignment No. 2, Survey of the Development and Use of Software Tools in Industry. The overall scope of the Work Assignment is identified as follows:

Determine how different programming environments affect the use of and direct the development of tools for quality software. Report the current state of the art regarding the development and use of tools in each of the programming environments.

The Software Tools Studies contract is one of a series of efforts undertaken by NBS/ICST in connection with its responsibilities under the Brooks Act (PL 89-306) which aims to aid Government agencies to improve the cost effectiveness in the selection, acquisition, and utilization of automatic data processing resources. NBS efforts to satisfy its responsibilities under the Brooks Act include research in computer science and technology, and the development of Federal government-wide standards for data processing equipment, practices, and software. The software standards efforts comprise six families of standards, one of which deals with software quality control. Although it is recognized that software tools can aid in software quality control, NBS/ICST has concluded that there did not exist a clear body of techniques for making effective use of tools. The present and related efforts are intended to fill this gap.

The purpose of this report is to document the detailed data in a form which will be useful for further research in the tool area. Section 2 of this report contains a classification of survey participants which may be helpful to readers who want to only selected classes of users, e. g., Government organizations. Section 3 contains the questionnaire on which the interviews were based, and the synopses proper are presented in the order of the interview numbers in Section 4. The classification of tool features of the tools encountered in the survey will be found in Section 5. This classification uses the principles and terminology described in [2].



## SECTION 2

## CLASSIFICATION OF SURVEY PARTICIPANTS

The following table presents a classification by size of software staff and type of organization of the 23 primary participants in the survey. Three organizations which volunteered information but which were not contacted in person are designated as Nos. 24 - 26 in Section 4. These are all private and include one each in the small, medium, and very large categories. Where two interviews were conducted at a given organization, these are designated A and B in the table.

TABLE 2 - 1 CLASSIFICATION OF SURVEY PARTICIPANTS

Inter- view No.	Type of Organization		
	Private	Gov't Support	Government
1	V		
2	M		
3			M
4A			L
4B			L
5	S		
6		V	
7			V
8			M
9	S		
10A			S
10B			V
11A		S	
11B		M	
12A		V	
12B		V	
13	V		
14A	L		
14B	S		
15			L
16			V
17	L		
18A		L	
18B		M	
19	V		
20			L
21			M
22		S	
23			V

Letters designate size of professional staff.

S = small (<15) M = medium (15 - 39) L = large (40 - 99) V = very l. (>99)



## SECTION 3

## QUESTIONNAIRE USED IN TOOL USER INTERVIEWS

## Part I - Tool User Information

1. Organizational identification of user.
2. Approximate annual budget of programming and computer operation departments (alternatively, the number of analyst/programmers and the number of computer operators).
3. Relationship to ultimate software user, and the nature of user imposed requirements (govt. standards, etc.).
4. Range of programmer skills.
5. On-site computer equipment and operating systems.
6. Off-site equipment, operating and communications systems pertinent to tools usage.
7. Programming languages now used and considered for the future.
8. Software requirements and design (formality, reviews, updates).
9. Programming environment (size of groups, on-line vs. batch).
10. Software quality control (organization, formality, standards).
11. Documentation (responsibility, formality, standards).
12. Changes and software maintenance (responsibility, procedures).
- 12A. What are the major software problems in this environment.
13. Participation in computer and software standards efforts.
14. Awareness of software engineering literature and meetings.
15. Circumstances of the first tool acquisition.
16. Tools currently in use.
17. Tools no longer in use.
18. Tools considered for future use.
19. Who decides on tool procurement and by what criteria.
20. How are new tools introduced (formal and informal training, documentation only).

## TOOL USER INTERVIEW

## Part II - Tools Information

21. Tool Identification, originator, and date of first use.
22. Tool ownership data (licensed, on time-share, etc.).
23. Original purpose and current application(s) of tool.
24. Tool and target software data.
25. Tool usage data (set-up time, run time, analysis required)
26. Typical usage (case history).
27. Frequency and conditions of usage (customer specs).
28. Benefits of usage (vs. prior practice).
29. Standards serviced or enforced by the tool.
30. Principal original reason for tool acquisition; were other tools considered, and what was the basis for selecting this tool.
31. Current reason for tool usage,; target software characteristics important for use (e. g., safety critical applications, very tight timing requirements).
32. Formal cost/benefit analysis if available.
33. Are all features of this tool used? Most and least used features.
34. Suggested tool improvements.
35. Tool reliability/availability/maintainability.
36. General comments on tool use.
37. Recommendations for other applications of this tool.
38. Compatibility of this tool with other tools and with the procedures of the computing environment.
39. Are some functions of this tool also provided by other tools or general support software? Where should these functions be located?
40. Suggestions for integration of this tool into a highly automated environment or into a library.
41. How will anticipated future software engineering developments affect the usefulness of this tool.

42. Effect of tool usage on software quality.
43. Effect of tool usage on personnel requirements.
44. What documentation was furnished with this tool? How useful is it?
45. Was special training provided with this tool?

## SECTION 4

## TOOLS INTERVIEWS

## TOOLS INTERVIEW #1

## Part I.

1. This interview was conducted at one of several major software development groups of a large defense and aerospace contractor.
2. Over 500 programmers, analysts, and related personnel are employed.
3. In most cases the company is the ultimate user. Even where programs are deliverable under government contracts, they are in only a few cases used by other personnel. The typical user is an employee working on a U. S. Government contract. Occasionally, specific tools or their capabilities are specified in the contract.
4. All programmers are believed to be degreed professionals (there may be a few exceptions). The official labor designation does not distinguish skill levels. The general impression is of a highly competent programming staff.
5. There are six major computing centers (plus a large number of small computers). The centers have large mainframe computers from at least three leading manufacturers plus some specialty computers.
6. The organization has occasionally used off-site computing but does not do so at present.
7. In the past, the principal programming language has been FORTRAN. Recently Structured FORTRAN has been used more frequently. A number of other languages are also available.
8. In some cases software development methodology is specified in the contract or by project management personnel. More frequently, these requirements are self-imposed by the programming staff.
9. The programming environment or the structure for a standard programming group is not specified as such. Typical groups range from three to ten programmers. A chief and a design monitor are usually appointed. The assignment of a librarian depends on the task to be performed.
10. Formal software quality control is usually performed by a dedicated group that handles hardware QA as well. In-house standards are used.
11. All software products are delivered with extensive documentation. The documents are generated by the project group. There is no specialist organization for documentation; however, editorial assistance is available if desired. As much as possible, documentation has been automated -- flowcharters and word processors are used.



12. There are no separate maintenance groups. Development personnel continues to be assigned to a project to do maintenance.

13. There is participation in standards efforts.

14. Personnel are frequent contributors to professional literature and meetings.

15. Initially the software tools were acquired through the 'side door' from outside sources. Management was not interested in the development or use of software tools. In one such effort, in the late 1960's, a test analyzer was acquired and was shown to be effective. Interest in tools increased somewhat after that time.

16. More than 100 tools are listed in a current software tools catalog which is available to employees. About 24 of these are frequently used tools.

17. The most frequent cause of tool abandonment is the availability of a better tool. E. g., an early flowcharter has been replaced by a better one.

18. Tools favored for future development are those that deal with resource scheduling, management control, data management, and inventory control. There is little interest in central development of new programming or test tools because (a) an adequate amount are available, and (b) individual programming groups are doing their own thing.

19. The manager of a software tools group decides on central tools procurement, subject to a budget constraint.

20. When new tools are introduced, formal training is always provided. It consists of lectures with the usual audio-visual aids, and also a workshop as part of which the tool is used or input and output files are examined. Bulletins alert management and employees to the availability of new tools.

## Part II

This part is provided for two representative tools, a large management oriented tool, and a small documentation tool.

### 1. Management Tool.

This tool is a project scheduling and management program. It is in use for most major projects at this facility. It was developed in-house and is considered proprietary. It is used to schedule tasks and to track costs throughout the life of major projects. It is preferred to the conventional government project management files even by government personnel. The tool resides on disk and can be called whenever needed. Set-up time for a new project ranges from one to two weeks. User data input is coached by a screen display on an interactive terminal. Minimum skills required for usage. Output is easily understandable.

A modification is now in progress to change the input language from fixed format to a free-format procedural language. This change was desired by the users. This is a stand-alone tool. The output is constrained to be compatible with a plotter. Integration with the plot program and with library routines may be desirable.

User documentation is fairly extensive and is targeted at personnel who are not software professionals. A four-hour training program is available for first-time users.

## 2. Documentation Tool (Flowcharter).

This tool generates flowcharts for a special class of structured FORTRAN code. It was developed by the company and is considered proprietary. Prior to undertaking the development, a search for available tools was made. None of those encountered was suitable for this specific language or the environment. The tool works in conjunction with an in-house developed preprocessor. Originators of other structured FORTRAN preprocessors refused to furnish source decks and documentation which are required by DoD regulation if the structured code is furnished as the primary program. This tool was developed particularly for the key words and syntax of the preprocessor.

The tool resides on disk. Set-up time is minimal. This tool reduces documentation time and the demand for art services. Originally there were options for paper size. An analysis of tools usage showed that only 8-1/2x11 was used. The options have recently been eliminated. The tool has been very stable over the last three to four years. Availability and reliability are not a problem. Little maintenance is being performed.

There is a User's Manual and a brief training period, about one hour, is usually provided for new users.

## TOOLS INTERVIEW #2

1. This interview was conducted at the data processing department of a medium-size regional bank.
2. There are 25 programmers in the department. Programming is also done in two other departments by two or three people each (part time).
3. The department serves the bank and does some data processing for correspondent banks. Some regulations of the Controller of Currency and the State Banking Commission affect data processing. Typically these specify the information to be processed or preserved but not software requirements per se. Some requirements for security and portability of programs are implied by these regulations.
4. Minimum requirements are two years of programming experience and knowledge of two of the three languages used. The most highly rated employee (not necessarily the most experienced) has two years of college and ten years of experience, practically all with this bank. Personnel turnover is not a problem.
5. There is one large mainframe by a major manufacturer which does the bulk of the financial record keeping. Two large size mini-computers serve, respectively, 900 teller terminals and the trust department. Another mini is being ordered.
6. Off-site computing is used for specialized functions: credit card processing, monthly financial model, personnel records. Interface is by shipping records, RJE and on-line terminals.
7. Languages, in order of importance, are assembly, COBOL and PL/1.
8. Most of the applications software is developed in-house. Requirements are determined by the user, then evaluated by a joint processing team. Estimates of development cost and run time/resources costs are generated, and these must be approved by the Executive Council before proceeding. For projects in excess of \$10,000 development cost detailed specifications are developed. There are periodic reviews, including a post-development review before the program is declared operational. Updates are requested by the user and are subject to similar controls as new programs.
9. Coding sheets and terminals are used. There is a de facto chief programmer for each project. Teams range in size from two to five.
10. Quality control is mostly concerned with procedural matters. Testing is conducted by a joint user-developer team. There are in-house standards.
11. Users write their own documentation which is reviewed by the data processing department. Flowcharts, listings, and maintenance manuals are prepared by the DP department and are kept in a library. In-house standards are used.



12. Changes, whether corrective or enhancements, must be requested by the user (DP department can suggest that these be requested). A formal priority system for changes is maintained by the department.
13. Personnel participates in ABA standards activities. Location of serial numbers on checks was given as an example. There is little awareness of computer or software standards.
14. There is no participation in software professional activities and little awareness of literature in the field.
15. A library system was acquired in 1974 because the operating system was very inefficient as a file handler. It was justified on the basis of savings in programmers time in a cost/benefit analysis.
16. In addition to the library system there is a sort program and a number of enhancements of the operating system.
17. An early sort program was discarded as being functionally unsuited and the present one substituted.
18. A much more secure operating system is being considered but there is hesitation about installing it because of possible incompatibilities with another computer which is at present the back-up facility.
19. All tool procurement in excess of \$ 1,000.- (which in effect means all tools) must be approved by a committee. Rate-of-return criteria are generally used. The improved operating system was proposed on the basis of security without economic trade-offs.
20. Formal training by the tool developer has been used in the past.

## PART II

The key tool described is the library system (see 15. above). It is a proprietary program and is being leased from the developer. It was procured for programmer assistance in development and maintenance and is still used for this purpose. It is on disk and there is no set-up effort required. Programmers use it as "free".

Not all tool options are used. Examples of unused features are hierarchical sign-on and some access control options. There were no reliability problems and no improvements were suggested. Integration in an overall software development system was considered feasible but no specific benefits were identified. The tool did save programmers' time and was considered to earn its keep. The user manual was rated satisfactory and training was provided by the developer. New employees are sent to a one-day class given periodically by the developer.

## TOOLS INTERVIEW #3

## Part I

1. This interview was conducted at a national Data Center for a fiscal function of a U. S. Government Department.
2. The annual budget is \$10 Million; there are 26 programmers and 13 computer operators.
3. End users of the programs are other organizations of the same Department. User operations are predominantly on-line. Users specify functions only. A Department Directive specifies languages.
- 4.. System Programmers are mostly degreed, 15+ years of experience, GS-13&14. Application programmers average 8+ years of experience. Entry level requires 2 - 3 years of experience. Recently there has been some upgrading of computer operations personnel into programming positions.
5. A major mainframe (dual processor) serves about 1300 interactive terminals. Twenty-five terminals are used by programmers, and another 25 are used in the Data Center for data entry. The rest are distributed all over the U. S.
6. No use of off-site computing equipment. They access other data bases.
7. ALGOL used with computer manufacturer's Data Management System. COBOL considered for future use.
8. There are formal requirements and design specifications. They are subjected to extensive coordination and require approval at several management levels.
9. Programming groups are formed as needed. One recent project included eight programmers at the Data Center and two located at the East Coast.
10. There is no formal QA organization at the Data Center. The most skilled personnel usually review programs prior to release. There is some QA on the data base at the East Coast.
11. There does not seem to be a formal requirement for documentation, but documentation is being generated.
12. Changes are initiated by either a trouble report or a change in the functional specification. About 95% of the trouble reports are written by Data Center personnel.
13. There is no participation in standards activities.
14. There is only minimal awareness of software engineering activities.
15. A library tool was developed in-house, based on the operating system utilities, in 1976. It has been in use since then.

16. The center uses the manufacturer's Data Management System, the library tool described in [15] and several editing tools.
17. An early version of the editor was abandoned when the new one was installed.
18. It is intended to install a new (commercial) library management tool. Also there is considerable interest in computer aided instruction programs. The latter are aimed at user personnel. They are currently arranging for consulting services in this area.
19. The Data Center Director is responsible for tools procurement.
20. The use of ALGOL as the primary language in the Data Center represents a constraint on acquisition of new tools. Tools which are being acquired are put into operation after the application programmers have been trained. This training involves classroom instruction and review of vendor supplied documentation. Sometimes in-house personnel conduct training.

## Part II

The key tool described in the data management system which was originally leased from the computer manufacturer but has been purchased in the meantime. The tool standardizes set-up and maintenance of the data base. It provides systematic dumping and recovery of files by means of audit trails. It is in operation all the time. For the specific mainframe used here, there were no suitable alternate data base management systems.

Six months were required for set-up and familiarization with the tool. It was selected after they compared its results with the previous non-automated data base handling and found significant savings in programmer time. Recently the Department requested that a new data retrieval program be initiated. There was considerable pressure from Washington to get this done in a short time. By use of the tool, which eliminates a lot of the detail work of file administration, this work was accomplished on time.

All available features are believed to be used. Functionally the tool meets all requirements. The response time is slow and they are looking for improvement in this. Reliability is generally good. A separate access control (security) program is currently being used and this could be combined with the data management function.

Documentation was considered good. The tool is being updated twice a year and the documentation is brought up to date at the same time. The manufacturer's User Group provides training which is rated as very effective.



## TOOLS INTERVIEW #4A

## Part I

1. This interview was conducted at one of several programming groups at a military installation.
2. About 40 military programmers and over 100 support contractor programmers are involved.
3. This is a service organization furnishing simulations, training aids, and other support software to using commands. Standards are mostly self-imposed.
4. The function is considered a high-technology area. Personnel have 3 - 5 years programming experience, and up to 20 years of experience with tactical information and control systems.
5. There are nine computing centers, each with three or four computers (mostly military designs) with a variety of operating systems. There are many programming stations and other peripherals.
6. Some of the systems have access to the ARPAnet.
7. FORTRAN and CMS-2 are heavily used. PL/1 and ALGOL are used to a lesser extent. There is very little COBOL programming. Several assembly languages are also in use.
8. There are no structure requirements for the design or development of programs.
9. A typical programming environment includes 15 - 20 programmers, each working at an interactive terminal at his own desk. Programmers have a variety of software tools at their disposal [see 16 for meaning of tool in this organization].
10. Quality control is handled by each project according to its needs. Usually one or two people are assigned to this activity. Published standards are used as a guide rather than as compliance documents.
11. Documentation for R&D projects is very informal and frequently insufficient. For operational projects, documentation is very extensive but not of uniformly good quality. Usually documentation starts early in the life of a project.
12. Requirements for program change are generated by
  - 1) an operational user trouble report
  - 2) functional changes required by the sponsor (Washington)
  - 3) the development group itself.All changes and other maintenance actions must be approved by a fairly high management level.
13. There is no involvement in standards activities.

14. There is no involvement in software engineering activities.
15. The first major tool was developed in the mid-1970s. It was a command-communications environment simulator.
16. There are 20-25 tools available to each major project. Currently there are six major projects. With allowance for overlap, there are probably 100 tools in use. (Apparently mostly simulators, exercisers, etc. which are program specific.)
17. An old text editor was replaced by an improved one.
18. Tools considered for future use include a hardware description language and a requirements language and analyzer. The general areas of interest are: formalization of requirements and design information, test data generators, and data analysis/reduction tools.
19. The project manager, frequently based on inputs from the (Washington) sponsor, authorizes tools procurement or development.
20. There is little formal training. Tools are demonstrated and put to informal use prior to commitment to actual operation.

#### Part II

This part was completed for two tools: (1) a large simulator, and (2) a general purpose editor (considered a small tool).

##### 1. Simulator.

This tool was developed in house over a three year period ending 1978. It automates the environment for war games and supports software and hardware development of communication and control equipment. It consists of about 40,000 FORTRAN statements and cost about \$ 500,000 to develop. It runs on a commercial minicomputer and uses the full capacity of this system.

The set-up for a simulation requires about one week. A typical simulation involves interaction of 10 - 15 people. It represents a large saving in programmer time over custom-designed simulations. The simulation is essential for both system testing and training. There are no alternatives available.

Some options are costly and seldom used, partly due to lack of familiarity with these capabilities on the part of users. The tool undergoes major changes each year to accommodate user requirements. The program is considered very reliable. It was recently used for three consecutive days of war games without crash. No compatibility is required or provided. Suggested improvements include a standard user interface so that calls, starts, and interrupts would be under the same protocol. Ultimately it would be desirable to permit interconnection of several tools. There is extensive documentation, including an on-line users' guide with built-in tutorials. All personnel using the tool receive 2 - 3 weeks of intensive training.

## 2. Text Editor.

This tool was developed by a non-profit corporation and is in the public domain. It edits programs, text, or data files. It is a small program with negligible set-up time. The run-time depends on size of file but is usually not a factor. The significant advantage of this editor over conventional editors is that it permits both line-oriented editing and cursor-controlled editing. The latter is much faster and more convenient when CRT displays are used.

All features are believed to be used. It has good reliability. No maintenance is performed by this organization. Suggested improvements include the incorporation of auditing capability (static checking). An adequate user's manual is available. No special training is required.

## TOOLS INTERVIEW #4B

## Part I

1. This interview involves another software organization at the same site.
2. The staff comprises about 50 total personnel (15 Civil Service, rest contractor)
3. It is a service organization for a major facility and is engaged in R&D activities and software quality control.
4. Most of the CS personnel in the organization are degreed and have many years of experience.
5. The organization uses two commercial mainframes, several commercial minicomputers and a number of military computers.
6. They use other mainframes via dial-up and the ARPAnet.
7. Programming languages are FORTRAN, COBOL, BASIC, CMS-2, and assembly.
8. Software requirements and design specifications are formal and comply with MIL-STD-1679(Navy) for embedded systems and with DoD 7935.1-S for management information systems (MIS).
9. There is wide variation in programming environments due to the differences in the nature of projects. Both batch and interactive programming is used.
10. Software Quality Control is established as a separate organization. Practices and Procedures are fully documented.
11. In each project, the project manager is responsible for the documentation quantity and quality.
12. The SQC organization assists in Configuration Control and also operates document and program libraries.
13. Personnel are active in government and industry standards efforts.
14. Personnel participate in professional activities and have contributed technical papers in the field.
15. The first tool utilized was a test data generator (1966). It was scenario driven and generated test data much faster than was possible by other means.
16. Several special purpose tools are available; a tools catalog is on-line.
17. Some tools become obsolete because better ones are available; at other times the function for which the tool was designed is no longer required. No specific examples of abandoned tools are available.



18. Desired: a tool that provides formal proof that all system requirements have been satisfied.
19. Tools procurement and use are controlled by each project manager.
20. Software tools are usually introduced by means of documentation. Sometimes on-job-training is provided.

#### Part II

The manager interviewed stated that software standards are the most important tool for software development and usage. They are in this environment the most widely and commonly used means of enforcing desired features in software, and in that sense they constitute tools.

## TOOLS INTERVIEW #5

## Part I

1. This interview was conducted at a developer of word processing systems.
2. 6 programmers, including some part-timers, are employed.
3. There are no external or internal standards for software development.
4. Programmers are mostly non-degreed; beginners to 5 yrs experience.
5. The facilities consist of a variety of microcomputers.
6. No off-site equipment is used.
7. Microcomputer assembly languages
8. There are no explicit standards; owner supervises or reviews all programming
9. The programming environment is completely unstructured.
10. There is no formal QA
11. Code is generated in self-documenting form
12. After a version is released, there is no chance for revisions (because of widespread use). Very careful check-out prior to release, including trial use by outsiders, is employed to assure suitability for the intended environment. Modifications for new versions are recorded.
13. There is no involvement in standards activities.
14. There is very little awareness of software engineering activities.
15. Assemblers and editors that came with the operating systems were the first tools used.
16. Assemblers, word processors and editors are currently in use.
17. Several assemblers furnished by microcomputer manufacturers were replaced by a much more efficient assembler that was procured as a tool from an independent software developer.
18. Better assemblers and editors are desired.
19. The owner decides on tools procurement.
20. Documentation is the only form of training.

## Part II

An assembler that was developed outside the semiconductor industry was purchased in 1979. The assembler that came with the computer took about 15 minutes to assemble a typical word processor program. The new assembler does the same job in 4 minutes. It also provides improved Macro handling capability and file management.

All options of the assembler are used. Reliability is very satisfactory, and no maintenance is performed by the user. An opportunity for integration may exist in combining the assembler with a file management system. The instruction manual that was furnished is considered adequate.

## TOOL INTERVIEW #6

## Part I

1. This interview was conducted at the Computation Department of a major government-support laboratory.
2. The 1980 department budget is \$20 Million
3. They are a service organization to all parts of the laboratory. They receive a constant stream of requirements, some documented, some not. Programming standards are not specified by the user.
4. With very rare exceptions, all members of the department have degrees. Senior personnel has up to 30 years experience. Some inexperienced college graduates are hired.
5. Two very large commercial mainframe computers are in use, one of them a specialized array processor. Additional main frames will be installed shortly. In addition, there are 400 minicomputers and about 900 microprocessors in use. A tailored time-sharing operating system is in use.
6. Off-site computing is not used.
7. The principal programming language is FORTRAN. Some COBOL is used for in-house administrative programs.
8. Most software requirements are self-generated.
9. Programming is done individually or in groups, depending on availability. Terminals and batch are used.
10. No separate QA function exists. This is handled as required by specific projects.
11. There is a documentation group which is primarily concerned with proof reading and document production. Documentation itself is performed by the individual programmers.
12. The unit that produced the product is responsible for its maintenance.
13. There is considerable effort at in-house standardization with 3 or 4 activities currently going on. Participation in outside standards organizations could not be determined.
14. There is participation in professional activities.
15. In-house developed tools were in use in the early 60's (editors).
16. Many tools currently available: editors, flowcharters, project management tools.

17. Availability of better tools is the major reason for abandonment.
18. No specific tools mentioned, but they are always considering new tools.
19. There are no firm criteria for tools acquisition. Local management decides.
20. First, new tools are validated. Then their presence is publicized, and formal class room training is arranged. Private consultation and on-job training are also used.

## Part II

The key tool identified here is a macro facility that was developed jointly by a number of government-support organizations in the mid-sixties. This department estimates that it has invested about 10 man-years in the effort. At present one programmer is assigned full-time for updating. The tool provides editing, indenting, file management and some diagnostics. The motivation for its development and use is that it saves programmer time but no detailed cost/benefit analyses were made. It is realized that similar capabilities are available in commercial tools but they were stated not to meet the detailed requirements.

The tool is in the public domain. Its reliability and availability are good. It is currently being translated from assembly to FORTRAN for ease of maintenance. The tool has been integrated with a library facility. Good documentation is available on-line and in hard copy. Special editing commands present somewhat of a problem which is being overcome by on-the-job training and consulting services. Adoption of more conventional editing commands may be desirable.



## TOOLS INTERVIEW #7

## Part I

1. This interview was conducted at a data automation division of a military installation.
2. About 140 programmers are employed.
3. The main users of the software are collocated services. Most software requirements are self-imposed.
4. Most programmers are degreed. The average experience is estimated to be 20 years.
5. There are seven mainframe computers of three different manufacturers in use.
6. No use is made of off-site facilities.
7. The language is mostly COBOL (90%), some FORTRAN.
8. Headquarters imposes some software requirements, probably mostly in terms of function.
9. Size of the programming group depends on job and availability of personnel. There is always a lead programmer designated. Programming is done at interactive terminals or batch (over the counter) depending on programmer preference.
10. There is no dedicated QA function. All programs are checked by a lead programmer from another group. Documentation is also checked by an independent person, possibly assigned from a technical editing group.
11. The system specification which describes the program to be produced is the initial documentation. The definitive documentation is developed from this in an evolutionary manner. The final document is formal and in a standard format. Project personnel produce the documentation.
12. The department maintains its own software. Normally the programmer who generates a program also maintains it.
13. There is no participation in standards activities.
14. There is little interest in software engineering activities.
15. In the early and mid-60's, some test tools for magnetic tape operations were developed. At that time the quality of data tapes furnished by other activities was a problem.
16. About a half-dozen tools are currently in use. At least one of the tools is heavily used.

17. On the mainframes of one manufacturer there have been several generations of tools, old ones being replaced with more capable newer ones. On the other systems the tools that were originally acquired are still in use.

18. A test data generator is being considered.

19. Headquarters has final authority on tools procurement. Local supervision makes recommendations.

20. When a new tool is acquired, it is tested. After it has been found suitable, users are advised, and formal class room training is initiated. Documentation is also used.

## Part II

The key tool at this facility is a test analyzer that had been acquired by this organization from another service. The original purpose was to provide management with the capability of determining how thoroughly a program had been tested prior to approving its release. This is still the major use. Originally this tool was operational only on one set of mainframes. Because the capabilities were considered essential, it was converted for use on all computers. However, in the process of conversion some capabilities had to be restricted so that it functions somewhat differently on the various computers.

The run time expansion for this tool is rather modest but was commented on as being a drawback because the computer facilities are completely loaded. Some options have been eliminated. All that are currently available are used. Normally, the tool is used to identify that all paths in a program (including paths for improper input and other exception conditions) have been tested, and that the test results are correct. It is also sometimes used as a program debug aid to identify the specific path taken by a failed run. Both reliability and maintainability were judged to be good. The tool has demonstrated its effectiveness in improving software quality. Integration of this tool into a tools library was viewed as desirable. Good and complete documentation exists. Training consists of a one-hour lecture.



## TOOLS INTERVIEW #8

## Part I

1. This interview was conducted at one of several geographically distributed programming divisions of a government agency.
2. 16 programmers plus clerical support are employed. Computer operations are provided by a different office.
3. The system group in Washington defines requirements. Field offices do the detail programming. Users include organizations that have their own software specialists. Users are other organizations within the agency.
4. Target level for programmers is GS-12. First line supervisors are GS-13. About two-thirds have bachelor degrees (usually not in Computer Science), none higher. Average experience level is 12 years. Trainees are brought in from operations or other agencies. They usually have some college training but no degree.
5. The computing equipment consists of four medium size mainframes, three of which are over 10 years old.
6. Other mainframes of three different manufacturers are accessed remotely.
7. Practically all programming is in COBOL with a little FORTRAN. DML (a subset of BASIC) is used as a data base query language.
8. There are agency standards relating to file naming, etc. Local procedures are considered more demanding and are enforced. They include walk-thrus, use of lead programmers, main line flowcharts (used as a design guide). Listings are reviewed prior to release.
9. Programming is done in 3 - 4 person groups. Terminals are used for time-share services, batch for in-house computers.
10. QA is supposed to be formal but is actually more or less informal. Four types of test are defined: program test, system test, user test, and volume test (the latter is a performance test).
11. For programs run on in-house computers, flow charts and operations manuals are prepared. For programs developed on time-share, there is little hard-copy documentation. Programs are supposed to be self-documenting. Systems group in Washington requires a user guide and frequently generates it there. Word processors are used for documenting new programs.
12. Corrective maintenance is requested by a scream on phone or 'Technical Information Dispatch'. Functional maintenance is requested by the user and is screened in Washington. Adaptive maintenance is requested through the central group in Washington.

13. There is no involvement in standards activities.
14. There is no involvement in software engineering activities.
15. The editor provided by the time-share system was described as a good tool.
16. No other identifiable tools are in use.
17. Some flowcharters were tried and discarded because they did not perform well.
18. The local manager would consider a tool that audits programming practices. Also tools for hardware performance and utilization.
19. Approval (budget) from Washington is required for any significant expense. Up to \$ 1,000.- for a tool can be handled out of local budget.
20. The time-share service had a four week training course. Now it is broken into several 3 day classes dealing with different subjects.

#### Part II

The time-share operating system was here considered the key tool. The system was acquired through the Washington headquarters. Programmers like the editor which has very powerful commands (no examples given), and the JCL which is easy to use. Up time and response time are rated very good.

The operating system was held to improve software quality because of good editing features - as one example: the editor can pick up many compiler identified errors and direct the programmer for proper changes. The documentation is rated good. No maintenance on the operating system is performed at this location.

## TOOLS INTERVIEW #9

## Part I

1. This is a software service organization, primarily programming and test, founded by its present technical director. Software tool development constitutes an appreciable sector of the company's business.
2. Six programmers are on staff.
3. Self-imposed standards are in use. On rare occasions clients may request specific standards or use of tools.
4. Most personnel have an MS in Computer Science with some experience. One exceptional programmer has no formal degree but is now enrolled in an MS program.
5. There are no on-site computers.
6. A time-share upper range minicomputer is used. On-site terminals are fed through a statistical allocation multiplexer.
7. Structured FORTRAN (two dialects) is the only language in use.
8. Software design requirements are not formally stated but a high standard is enforced through review by the technical director.
9. Most programming is done on an individual basis. On a few occasions a group of two programmers has been formed. All programming is done on terminals.
10. There is no formal QA. Mutual inspection of code and output are used.
11. User guides and reference manuals are generated for all tools. These are also available on line.
12. Each programmer handles his own changes. Configuration control is informal.
13. Slight involvement in standards activities.
14. There is heavy involvement in software engineering activities.
15. A structured FORTRAN preprocessor was adopted as a production (programming) tool when the company was started (about 4 years ago).
16. The preprocessor, a test coverage analyzer and a test formatting and management tool are currently in use. A file manager is being introduced.
17. A test data generator was discarded because it was difficult to use and not comprehensive

18. A software systems development facility is being planned.
19. Tools procurement or development is authorized by the technical director with an implied profit motive but no formal cost/benefit analyses are being conducted.
20. Interactive help among staff is the primary means for training on a new tool.

#### Part II

The FORTRAN preprocessor was selected as the key tool. It was developed in-house and is distinguished from other tools of this type in that it generates a highly portable FORTRAN output. Formatting and diagnostic capabilities were also described as an advance over existing preprocessors. The output code is optimized for minimum run-time (the run-time of the preprocessor is not optimized and it was stated that this is usually not a material factor). The tool enforces some design standards, specifically a block structure.

The preprocessor is constantly in use and has run without failure for the last six months. It is being maintained by this organization without reported problems. The tool is compatible with other tools in use at this organization, and integration into a software development system is being considered. Documentation was considered adequate.



## TOOLS INTERVIEW #10A

## Part I

1. This interview was conducted at a computer specialist group supporting the simulation function for a Federal agency.
2. There are 8 programmers. Some contractor personnel may also be involved.
3. The group is responsible for the systems programs for the simulations. Other departments do specific applications. Standards are self-imposed.
4. The most senior programmer has an MSME degree plus 20 years programming experience. The least experienced is a student in computer science.
5. A very large mainframe and a medium sized mainframe are in use.
6. No outside computing facilities are used.
7. The predominant language is FORTRAN but many assembly programs are still in use and have to be maintained.
8. They are now instituting a requirement for structured programming.
9. Most assignments handled by individual programmers, primarily on interactive terminals.
10. There are no formal QA requirements but the overall simulation function has a separate test group.
11. Separate User and Maintenance Documentation is being generated. They don't like flowcharts and prefer decision tables. Code is made self-documenting if possible.
12. There is a librarian for control of changes.
13. There is no involvement in standards activities.
14. There is no involvement in software engineering professional activities.
15. There are no individually procured tools in use. They do utilize the a files manager supplied as part of the operating system.
16. General purpose simulation languages and tools associated with them are not applicable in this environment. They have not investigated pre-compilers. They are in general too busy to investigate these techniques. There are no tools for identifying unused options in simulators.
17. An early flowcharter was tried at one time and was discarded.
- 18 - 20 Not applicable. No Part II interview was conducted.

## TOOLS INTERVIEW #10B

## Part I

1. This organization handles the general purpose programming for the main computation center of the same Federal agency. It represents high technology in programming at this facility. It is not believed that other programming groups use any tools in the strict sense.
2. There are 150 programmers employed by a support contractor.
3. The programs are generated for scientific and administrative functions at this facility.
4. Personnel are mostly degreed and have considerable experience.
5. Approximately ten mainframes from five different manufacturers plus some specialized computers and minicomputers are used.
6. There is no use of off-site computers.
7. 80% of the programming is in FORTRAN, 15% PL-1, and the rest COBOL and assembly.
8. Modular programming is enforced, structured programming encouraged.
9. Programming is done in groups. Terminals, batch and RJE are used.
10. There is no separate QA function. They are experimenting with peer group review.
11. FIPS PUB 38 is used as a guideline. There is also a style manual in use. User documentation is always generated, maintenance documentation only sometimes. There are standards for header pages, and the programs are made as self-documenting as possible.
12. There is formal control on changes. The file manager used by #10A is also used here. There are also some library programs in use but details were not known.
13. They participate in the PASCAL standards group (X3J9).
14. They participate in user groups, and have slight involvement in software engineering activities.
15. An early flowcharter was tried many years ago and was found to be of marginal utility.
16. Performance measurement tools and a preprocessor are currently in use.

17. The flowcharter was not matched to graphics equipment and was difficult to use.
18. A cross-reference program was recently considered and rejected because (a) cost and (b) almost the same features were available on an operating system.
19. A local manager decides on tools procurement.
20. On the preprocessor a four hour training session was provided.

#### Part II

The key tool is a performance measurement program that was installed about 6 years ago to identify computer usage bottlenecks. It is currently being used whenever a usage problem is suspected (once or twice a month). A typical case involves very inefficient tape handling routines. Computer time could in this case be reduced by a factor of 20 through use of the tool. Software clocks in the computer could be used for the same purpose, but the tool is more elegant and saves times for set-up.

Set-up and analysis time are considered small. The tool does not crash and is considered reliable. It is compatible with the computers in use at this facility. There are no current plans for integrating it with other tools. The tool was acquired from another government organization and only sketchy documentation was furnished. However, this is considered adequate. Only on-the-job training is used.



## TOOLS INTERVIEW #11A

## Part I

1. This interview was conducted at a research center (a part of a major government-support organization) that furnishes scientific computer programs to other research facilities.
2. Eight professional staff and two students are involved.
3. The primary distribution of the programs is controlled but they are in the public domain. It is expected that copies will be made and used by other than the primary recipient, and therefore the possibility for correcting errors once a program has been released is limited. All standards are self-imposed.
4. The staff are all Ph.D.s in science with one to twenty years experience. All are very knowledgeable programmers. Students also have programming experience prior to being hired.
5. Three mainframes, all by a single major manufacturer, and one mid-range computer are used.
6. Other specialized computing facilities (including array processors) are accessed via the ARPAnet.
7. The programming languages are mostly FORTRAN and assembly. There is also a special purpose language BLAS in use (Basic Linear Algebra Subroutines). No other languages are being considered.
8. Modular programming is generally used. There are no other programming standards. They are now conducting an experiment in use of structured programming using a public domain preprocessor.
9. Each programmer works on an individual task. RJE and interactive terminals are used.
10. The programs furnished by the center are identified in three quality classes. For the highest quality class the center conducts acceptance procedures and the code is maintained. The acceptance procedures were rather informal and can be described as tender loving care. The second class consists mostly of code contributed by other organizations which is being maintained (either by the center or the contributor). No acceptance is conducted on these although some test cases may be run (usually those supplied with the program). The third class consists of non-maintained code, supplied as caveat emptor.
11. An established format is used for headers, and this is the part of the documentation that is carefully reviewed. It consists of the general header material and examples of program use. There is also a formal user's manual for most programs. Maintenance relies exclusively on comments. It is very much tied to the developer. It is expected that the developer will maintain the code even if he leaves the organization because his professional reputation is

enhanced by a good program. There are carefully worked out procedures for attribution of programs.

12. One computer has a librarian function, and use of this for changes is mandatory. It is believed to provide good control. The use of file managers on other computers is on the honor system and believed to be less effective. Some changes are made by users. Changes are announced in a semi-annual catalogue.

13. There is no involvement in standards, though there is much concern with issues of portability

14. One of the programmers participates in software engineering activities.

15. A batch editor developed at a university 'many years ago' was the first tool.

16. Editors, preprocessors, a portability auditor, and file managers are currently in use.

17. One text editor was not satisfactory - too cumbersome to use, not truly interactive.

18. Tools for vectorization of FORTRAN (generate Macros), tools that facilitate portability. See also Part II.

19. The software manager decides on tools procurement.

20. Generally manuals are considered sufficient.

## Part II

The key tool described at this organization is a simulator for an array processor that runs on the midrange computer (which is not an array processor). It is a proprietary tool that was acquired for a one-time license fee during 1980. The organization expects to acquire the array processor that is being simulated and uses the tool in the meantime for program development. The tool also furnishes timing estimates and this is considered an important feature because it permits code to be optimized.

Set-up time is not large. The run-time expansion is about 1000. This is considered undesirable but not prohibitive. The tool is intermittently used by one programmer. There have been some crashes for unexplained reasons. There is no protection against illegal instructions. The tool includes an assembler and loader. A library manager is under development by the originator of the tool and this should be helpful. The tool is believed to improve software quality due to good diagnostics. It reduces personnel cost vs. other means of accessing the array processor (e. g. eliminates the need for travel). The user manual is acceptable. Training is available from the originator but was not used.

## TOOLS INTERVIEW #11B

## Part I

1. This interview was conducted at one group of the computer and mathematics research center at the same government-support facility.
2. 30 programmers are employed.
3. The group conducts research on software portability and similar subjects. The user is anyone who employs 'canned' programs. Standards are self-imposed.
4. Programmers are all degreed, mostly in math and physics. About 25% have advanced degrees. Average programming experience is 5 years.
5. Several large mainframes and two super-minicomputers are in use.
6. At present no off-site computing is used.
7. Standard and structured FORTRAN, C-language, and data base manager language are used. For the future there is some interest in PASCAL.
8. The only standards used are enforced by the compilers.
9. Programmers work on individual tasks, all on interactive terminals.
10. There is no formal and little informal QA.
11. In this group, formal design manuals are being kept. These are available on-line and in hard copy. All programs also have 'User Interface Manuals'.
12. Corrective maintenance is done in response to 'screams on the phone'. Adaptations to new computers are done periodically. They are reported, frequently in the Communications of the ACM.
13. The group leader participates in standards activities.
14. Many members of group are active in software engineering organizations.
15. Specialized FORTRAN compilers were in use about 10 years ago.
16. The group is currently working on a highly portable operating system that includes several tools. Also, a preprocessor is in use.
17. An editor was abandoned because it was line oriented and judged difficult to use in this environment.
18. A portability analyzer, a software development system, and a general language semantics analyzer are being considered.
19. The group leader makes decisions about tools acquisition.



20. Sometimes formal training is used for introducing tools, but this is not the general practice.

#### Part II

The key tool discussed in this interview is the portable operating system that is being developed by this group and will be in the public domain. The portions which are available now are in use almost daily and have been successfully ported to other facilities. They execute more slowly than the standard operating system from which they are derived but the difference is not noticeable to most users. Improvements in the user visual interface and in better context isolation for local/global variables are considered desirable. Other improvements are being made based on comments from the user group.

The portions that exist now can be integrated with the standard version of this operating system. The portable version saves considerable effort when the operating system has to be re-hosted. Portions of the portable system have run for over a year on eight different computers and no crashes have been reported. The user interface and the design are well documented. No formal training program is being provided.

## TOOLS INTERVIEW #12A

## Part I

1. The interview was held at the programming department for the telecommunications function of a major government-support organization.
2. The staff includes about 100 programmers, 50 of which are employees, the rest contractor personnel. In addition, many engineers do some programming.
3. This office develops software for space communications. Program offices usually specify the language to be used and may impose standards on structure, etc.
4. The average programmer's education is between BS and MS. There are some non-degreed programmers and some Ph. D.s. Most programmers have several years experience.
5. There are three midrange mainframes and a number of minicomputers that are specially suited for communications functions.
6. No outside services are used at present.
7. Programming is 90% assembly language, 10% FORTRAN and M-BASIC
8. When the software becomes operational, a formal transfer agreement is executed by the developer and the user. Most other development aspects are controlled informally. See 3. for language specification.
9. Programing is done in groups of 5 - 8, headed by a Cognizant Development Engineer. At one time the size of groups got up to 11, and this did not work out very well. Library services are performed by a separate Software Production Management Office.
10. There is a separate QA organization, which acts as agents of the program office .
11. The documentation is generated by the programmers but is produced by the Software Production Management Office [see 9]. There are at least a Software Operations Manual and a Software Specification Document for each program. Flowcharts are in use now, but there is a tendency to go to a program design language.
12. Changes are reviewed by a Change Control Board. Authorized changes are usually made by the developing group. Version control is exercised by the Software Production Management office.
13. The department participates in the PASCAL standards effort.
14. There is some participation in professional activities.



15. A proprietary program design language was acquired in 1976 by a 'grass roots' effort.

16. The program design language and a number of software management tools are currently in use.

17. An early flowcharter was found not suitable.

18. A tool for automatic generation and monitoring of work breakdown structures (WBS) is under development. Also, V&V tools developed by a government agency are being considered. An improved version of the program design language is being introduced.

19. The program offices authorize tools procurement. There is considerable informal tools usage.

20. Usually tools are introduced by means of training sessions.

#### Part II

The key tool discussed at this interview is the program design language. It is a proprietary software product acquired under a one-time license. Program structure is developed by formatting the functional requirements through the design language. It can be used in a hierarchical manner by going to increasing detail. It is in use on practically all current developments although usage is not specifically demanded by customer spec. It allows a more efficient presentation of control flow than flowcharts. Uniform type face is considered a benefit. (In their flowcharts, lettering is compressed when much information is to be presented.)

The tool is on disk and is always available. Reliability is adequate. No maintenance is possible because the developer does not furnish source code. It had been desired to change some of the reserved words but this was not possible. A replacement for this tool is currently under consideration which allows reserved words to be specified by the user. The design language is compatible with most features of the computing environment in this office. Documentation for the design language was not considered to be particularly good. No special training is being provided. Programmers were motivated by themselves to get familiar with it.

## TOOLS INTERVIEW #12B

## Part I

1. This interview was conducted in a software development section that furnishes programming services to program offices.
2. There are 100 programmers, and programming is also done by engineers. All told, there may be 300 to 400 people who are doing some programming.
3. Most of the software generated in this section is for use of a program office that does not impose specific standards.
4. Most personnel have a degree in a technical area, up to 20 yrs. experience.
5. The computing equipment consists of three large mainframes (two vendors).
6. Several additional mainframes are accessed via timeshare services.
7. Programming is 80% FORTRAN or structured FORTRAN. The remainder is HAL/S, PL/1 and PASCAL. There is some use of assembly. For the future more use of PASCAL is expected.
8. Section policies require the use of design languages and high level languages. They also place emphasis on portability and make use of a portability auditor where possible. In addition, a group within the section has more definitive standards for program structure, reviews, and software production management.
9. Programming is done in groups of 4 - 10 headed by a Cognizant Programmer. Most programmers use interactive terminals. Some prefer batch.
10. Using projects act as their own QA. There are no universal standards.
11. Programmers are responsible for documenting their code. Sometimes they also prepare the requirements document in cooperation with the user. Initial QA on documents is performed by the Cognizant Programmer, final QA by the user.
12. Configuration control varies with project. Most require fairly strict configuration control.
13. The section participated in several ANSI standards efforts.
14. There is some participation in professional activities.
15. An automatic flowcharter was introduced about 10 years ago. Program design language and preprocessors came into use about 5 years ago.
16. Design language, preprocessors and portability auditor are currently in use.

17. A hierarchical design presentation was used on a major project. It became too burdensome at the detailed level. A common complaint is that more training is needed when tools are introduced. This is believed to be also responsible for the poor reaction to early flowcharters.
18. An automated library system is being considered for acquisition.
19. Using projects may authorize tools. Others are acquired informally.
20. There is no fixed policy on training.

#### Part II.

The key tool discussed at this interview is a FORTRAN preprocessor which was developed at this facility (but not in this section) and is in the public domain. In addition to usual preprocessor functions, it audits control structures and furnishes indented listings. There is no set-up time. The code is optimized for portability. Its own run-time and program run-time are at present not optimized. The tool was developed to avoid being tied into a proprietary product and to overcome limitations of some of the other preprocessors.

All options are used. Recently automatic invocation of the FORTRAN compiler has been added as an option. There are no known bugs. Errors in input syntax are not always well diagnosed. However, the preprocessor is preferred to the compiler as a debugging tool. In the beginning personnel requirements went up ("People are always hesitant to try something new"). The manual is considered quite good. A course is given periodically and there is also on-the-job training.

## TOOLS INTERVIEW #13

## Part I

1. The interview was conducted at the technology laboratory of the software engineering division of a major defense contractor. The laboratory provides methodologies, tools, and does research (IRD & funded).
2. There are about 500 programmers in the Division.
3. Users are armed forces (U. S. and allies). These usually impose programming and program development standards. However, see 8.
4. The normal entry requirement is a bachelor's degree in Computer Science or a related field. About 3% have Ph. D.s, many have MS. About 10% are in programming support positions, and these do not need to have a degree.
5. About 50% of the work is done on commercial minicomputers. The other 50% is on military host machines. There is also a mainframe that is used exclusively by the laboratory.
6. No outside services are utilized.
7. The languages used are JOVIAL, CMS-2, and some FORTRAN.
8. In-house standards (developed by the laboratory) are formal and strict. They are intended to supplant the customer desired standards wherever possible. The company will convince the customer that in-house standards will do everything the customer's standard requires, and then some. If customer does not agree, customer standards are accepted but the cost may go up.
9. Teams are usually 4 - 6 programmers. Most work on interactive terminals. For programming of military hosts, batch and RJE are utilized.
10. Review for accuracy of simulations and for adherence to standards is done by a central organization. Other aspects of QA are done at the project level.
11. Documentation requirements are usually defined by the customer. Typically they include a systems manual, user and maintenance manuals, the source code, and test plans and procedures. Configuration Control and Requirements documentation are also frequently supplied.
12. Prior to delivery, corrective changes are made on the basis of trouble reports, and adaptive changes on the basis of change requests. After delivery, changes are made only as authorized by the contract.
13. They participate in standards efforts.
14. They participate in many professional activities.



15. The first tools were simulators. Three of these were locally developed for military computers.

16. Many tools are in use, but there is no formal tools catalog. A large requirements analyzer is being used by some projects.

17. A flowcharter for use with a specific assembly language was discarded as being too difficult to use.

18. Tools for automatic verification of flight software are being developed under a government sponsored effort. An employee of the laboratory works on an improved requirements analyzer as part of a Ph. D. program. There is also interest in discrete events simulations for systems including computers.

19. The laboratory director decides on tools acquisitions.

20. A seminar is held every two weeks to which all professional personnel are invited. New tools are announced and discussed there. Another interesting method is to 'piggy-back' new tools on existing ones. Recently a testability tool (see Part II) was introduced by being made part of a structured design tool.

## Part II

This part is completed for two tools: a testability analyzer and a requirements analyzer.

### A. Testability Analyzer.

This key tool was developed in-house with company funds and is still in the experimental stage. It quantifies testability of a program and predicts the amount of effort required for test. Three projects are now using the tool. Management directs use of the tool because the individual programmer is during design not motivated to think about testability. Test accounts for 60% of the development cost, and the tool is expected to reduce the cost of test. Design errors account for 70% of all errors, and the tool may also reduce the incidence of these. The tool is integrated with in-house developed structured design tools. It was developed because no other tool was adaptable to this environment.

Set-up time is negligible. The tool resides on disk. Run time is seconds. When a complete test plan is generated it may run for 1 - 2 minutes. The tool structure is fairly simple. There are not many options, and all of these are used. A draft of a users manual is available on-line. No formal training was provided. Because the user interface is identical to that on the existing tool, it was not considered necessary.

### B. Requirements Analyzer

This tool is an established major program that was acquired from an outside source. It took two weeks to rehost the system. No specific data on run-time



could be obtained, but apparently this was a critical factor for some projects which chose not to use it. Requirements are input to the tool, and analysis for consistency and completeness is performed. In later stages, consistency of the design with the requirements can also be verified. 8 - 10 projects have used it since it became operative. This was done at their own initiative, not as a result of customer requirements. Results generally regarded as good but it is expensive to use.

The tool does a complex job and is itself quite complex. It requires an expert to operate it, and a specialized group has been established to act as an interface between the developer and the tool. The services of this group are paid for by the using projects. The general reaction of software developers to the tool was described as "pepper and salt" - a mixture of good and bad. The difficulty of use is apparently the major drawback. The tool found missing requirements and (less often) inconsistent requirements.

Personnel requirements were increased by the use of this tool. Documentation is regarded as excellent. Some personnel attended a workshop at the developer's facility.

## TOOLS INTERVIEW #14A

## Part I

1. The interview was held at a programming group that supports a specialized engineering function of a large nationwide company.
2. The overall information systems function has an annual budget of over \$ 175 Million. This covers 15 major projects with between 20 and 250 employees per project. In the programming group interviewed there are currently 60 people.
3. The ultimate software users are regional subsidiaries of the company. There is no central guidance on software design or development.
4. The staff consists of 8 requirements analysts, 3 engineering & documentation people who interface with the users, and the rest are involved in the actual programming. Of the latter approximately 55% are degreed personnel, the rest typically have 2 yrs of college.
5. No on-site computers are utilized.
6. The entire work is done on a commercial time-share system which facilitates access to the programs by the regional users.
7. Current languages are FORTRAN and a structured FORTRAN. There is experimentation with PASCAL. All new work is in structured FORTRAN.
8. Standards established within the department call for top-down design, design reviews prior to coding, modular structures (but modules are fairly large), structured FORTRAN, flowcharts or pseudocode.
9. Program teams consist of 2 - 3 people, typically one experienced and the rest less experienced. All programming is interactive.
10. Two formal tests -- Baseline (integration test) and Certification (user environment simulation) -- are run by a separate test group in the department. After this, the program is turned over to a Release Control function in a separate group.
11. User documentation conforms to company practice. It is prepared by the software development organization and consists of a User Guide (top level), System Functions and Output (a reference manual which is more detailed), and a pocket card which identifies commands and output. Maintenance documentation is at the discretion of the developer. The present practice is to put all documentation into the file. Extensive header information is required, and the department head reviews this.
12. All significant changes are coordinated in a formal manner, involving several functions outside this department. Minor changes can be made by the developer directly.

- 12A. The major problem is software productivity (particularly in implementing user requested changes).
13. There is no participation in standards.
14. There is no significant participation in professional activities.
15. The first stand-alone tool used in the department is a code analyzer and cross-reference generator (see part II).
16. Another tool in use is a rather minimal preprocessor. Performance measurement is obtained from the operating system.
17. A flowcharting tool made available through a user group is not much used
18. No tools acquisitions are contemplated.
19. There is no central authority for tools acquisition. Some tools have become quasi-standards on a company-wide basis by default. The key tool described below was developed on the basis of an economic analysis generated in the department.
20. Training is left up to an informal 'network'.

#### Part II

The key tool discussed in this interview is a FORTRAN code analyzer and cross-reference generator. It was developed in-house in 1979 and is considered proprietary. It performs static analysis and particularly identifies usage of FORTRAN Common. The tool showed that 70% - 80% of the FORTRAN Common was not utilized by most programs. Since storage for Common is always allocated, and since the time-share charges are based on memory allocation, this permitted considerable savings. The tool also serves as a general code auditor, primarily in data set/use.

This was one of the few tools for which an economic analysis was available. In brief, if tool eliminates just 2 pages of memory from a typical program, this will result in \$10,000.- savings of per year. The program took approximately 5-1/2 man-month to develop. This translates into a minimum of \$ 20,000 and at most \$ 50,000 (depending on allocation of overhead). The department head saw this as a clear justification for tool development. He did not have to submit this analysis to higher management. Actual savings are more than anticipated.

The tool resides on disk, is always available, and cost of use is negligible. Usage has shown no problems. There is no formal documentation on the tool at present. On-the-job training is being provided.

## TOOLS INTERVIEW #14B

## Part I

The general software development environment for this interview is very similar to that of interview #14A. However, this group uses in-house computing resources and makes use of a general software development system that is furnished a part of the operating system. The specific programming group includes 10 individuals who are all 'heavy' users of the development system. The most pressing software problem to the interviewee was to get programmers to think in terms of user needs. He expressed this as "to get software specialists and psychologists together".

## Part II

The key tool for this interview is a software development systems that was developed in-house in 1973. Program descriptions are in the public domain and are widely available. Versions for specific systems are being marketed by specialty houses. The tool provides a standardized environment for software development. This permits programmers to transition rapidly from one system to another. The most frequent usage of the development system in this environment is for source code control. The group generates different versions of their software for different users. The development system permits common or individual changes to be made, provides comparisons (deltas) of different versions, and performs library functions.

It is in use almost constantly. Essentially no cost is attributed to the tool. The most used features are file management (merge, compare, change, etc.). Documentation could be improved. Reliability is rated very good, and the tool is generally liked by programmers. The tool incorporates a highly integrated environment. The output of one function can be passed on to another. No formal training is provided. Normal cooperation within the group provides training for new personnel.



## TOOLS INTERVIEW #15

## Part I

1. The interview was conducted at the computer technology group of a government agency. The questions are answered in terms of overall agency usage.
2. The 1980 ADP budget is 82 million dollars.
3. The agency uses computers for both scientific and administrative functions.
4. Most programmers are not Computer Science majors. Skill levels range from journeyman programmer to research scientist.
5. A central large mainframe is used that can be accessed by about 1500 terminals (dumb and intelligent) distributed all over the country.
6. Extensive use is made of off-site services at universities and with commercial time-share operations.
7. ANSI FORTRAN, ANSI COBOL and BASIC are the only currently authorized languages. It is intended to add PASCAL and data base query languages.
8. There is a Systems Management Manual with policies and guidelines for the software life cycle. An Automated Systems Development Handbook with detailed examples and more specific guidelines is in the draft stage. Formal training in structured programming is being provided by a consulting firm.
9. Both interactive and batch programming are used. Small programs are done by individuals, complex systems by groups of three to six.
10. QA for small systems is performed by individuals. For complex systems there is a central support group. There is now an effort to acquire verification tools, and a contract has been let to a university to evaluate these.
11. Documentation is governed by agency regulations.
12. Until recently, the most frequent mode of requesting a change was 'a scream on the phone'. A new section has just been added to the Systems Management Handbook on trouble reporting and change control procedures.
- 12A. Major problems are incomplete requirements prior to development and failure to make effective use of available software engineering tools.
13. There is participation in government standards activities.
14. Members of the technology staff are active in professional societies. Other programmers probably are not.
15. Flowcharters were used in the early 70's but were dropped because they were

difficult to use and because the charts they generated were not informative. Management tools were introduced in 1975 and are still in use. Simulators were used in the late 1960s.

16. A number of test analyzers, a sophisticated editor, a flow analysis tool and several other tools are currently in use.

17. The flowcharter referred to in question 15.

18. Static and dynamic analyzers, a general software development tool, code comparators, and cross-compilers are being considered.

19. The technology staff makes decisions on tools acquisition. Criteria include need, ease of use, resource requirements, portability, documentation and language.

20. New tools are introduced through formal training, demonstrations, or notification of availability together with associated documentation.

#### Part II

The key tool discussed in this interview is a COBOL dynamic analyzer. It was developed for the U. S. Government and was obtained from another agency. It is just now being introduced here. The tool is used for code optimization, analysis of test, and debugging. Both compilation time and run time are increased when the tool is used. Post-run reports furnish the number of executions at each instrumentation point and a summary of dwell time in each segment.

A number of usage and report format options are available. It is too early to tell how many of these will be used. It was stated that the tool "still has a few bugs". Documentation is considered quite good. No outside training was utilized.

## TOOLS INTERVIEW #16

## Part I

1. The interview was conducted at the programming staff of a major Government agency with primarily fiscal functions.
2. The programming staff includes over 600 people. Approximately 200 of these support internal statistical and management functions, the rest deal with the public functions of the agency.
3. The ultimate user for the public programs are the field offices of this agency. The other organizations either are their own users or service headquarters functions.
4. Senior programmers are all degreed, most in the computer or closely allied fields. Other personnel range from trainees to experienced, some with degrees.
5. Several large mainframes are in use.
6. There is no significant use of off-site services.
7. COBOL and assembly languages are used. There is no current interest in other languages.
8. There is an in-house procedures manual which is being updated and will be more strictly enforced.
9. Systems programming is mostly batch, others mostly interactive. Programming is done in groups. Size varies, usually not more than 10.
10. For programs that access the major data bases there are formal test steps. These are now being automated (at least partly) by time sharing facility. It is a system built around a mainframe operating system with some resident commercial tools. Mandatory pre-release testing is only now being instituted.
11. Responsibility for documentation is always with the developer. Degree of formality depends on the expected life of the programs.
12. Major program changes are usually the result of legislative actions. The functional requirements are established by other offices in the agency. Their requirements are forwarded to a coordinator who directs the programming groups to make the required changes.
13. No participation in standards was evident.
14. There is limited participation in professional activities, some in user groups.
15. There was no certain way of telling which was really the first tool. There had been interest in tools among various programming groups and commercial tools



have been procured for several years. There is now a list of commercial tools that are available on the in-house computers. There has also been some in-house development, including a library control system written in assembly. It provides back-up transparent to the user, easily accessible audit trails, and is well adapted to this environment. It was started around 1972.

16. Many tools are in use, mostly commercial. They include optimization, analysis (static and dynamic), and management tools.

17. The library control system mentioned in [15] is now being replaced by a commercial system which has been adopted as the agency standard.

18. No comments on future tools procurement were offered.

19. Individual offices have in the past obtained commercial tools. Others were developed as needed. There is a tendency now for more centralized control.

20. A wide range of training techniques are in use: personnel sent to vendor, in-house courses by vendor, in-house training by agency personnel, video cassettes obtained from vendor.

## Part II

The key tool discussed in this interview is a COBOL dynamic analysis tool that is primarily used for optimization. It is licensed from the commercial developer. The typical use is to correct inefficiencies in the COBOL compilers. Use of the tool achieves 30% reduction in storage and 10 - 15% reduction in run time. Most agency activities are presently computer bound so that run time reductions are very important. An option permits enforcement of structure standards. These standards are not agency-wide at present.

It is in use daily. Usage is mandatory for some programs (by internal directives). Frequently programmers are motivated to use it on their own. Although no formal analysis was available, it could be shown that the tool easily pays for itself in the context of the large amount of COBOL code generated here.

At present the tool does not handle re-entrant code. This restriction and also limitations on operating systems should be removed. Reliability is considered adequate to good. Some tool functions duplicate features of the operating system and these are seldom used. User manuals received mixed ratings from several individuals interviewed. Training at the developer and at the agency (with video cassettes) is being used.



## TOOLS INTERVIEW #17

## Part I

1. This interview was conducted at one programming center of a major computer manufacturer. The department described below works primarily on Government contracts.
2. The current budget for this department is about \$ 8 Million. This supports 58 programmers and 2 program librarians. Computer operations are separately funded.
3. The end user is in most cases a U. S. defense agency. Programming standards are usually specified by reference to military specifications.
4. Most of the programmers have degrees. Only two have less than two years of experience. Most of the others have seven or more years of experience.
5. A large commercial mainframe and some special purpose computers are in use.
6. There is no use of off-site facilities.
7. The languages in use are JOVIAL, FORTRAN and assembly. The latter violates customer specifications but is necessitated by performance constraints of the present computer. When a replacement is obtained, more HOL will be used. ADA is not currently being considered.
8. Frequently the actual governing standards are internal corporate ones which are equivalent to, or stricter than, the contractually specified ones. Structured code is used for all new work.
9. Both batch and interactive programming are used. Code is intended to be run in a batch environment. Programming groups vary in size. Library function is centralized.
10. Quality Control is an outside function (not part of this department).
11. Documentation is done by the person writing the code.
12. Changes are usually made by the developer. The department delivers sections that are affected by the change to the customer who assumes responsibility for configuration control.
- 12A. The principal problem faced by the department head is shortage of personnel, and unavailability of personnel with the right background. He is also concerned that the older, more experienced programmers tend to lose touch with the advancement in the industry and are unaware of current technology.
13. The company has strong internal standards efforts and participates in industry wide standardization. The particular group is not directly involved in these.

14. They are much aware of professional activities in software engineering.
15. A software development system was developed in-house to solve a specific problem in 1975.
16. An in-house version of Program Design Language has been developed recently and integrated with the development system.
17. There was no recollection of tools that had been abandoned.
18. A task force has been established in the division to consider additional software tools. It has not yet made specific recommendations.
19. Anyone can make recommendations to the task force. Decisions are made in a diffuse manner. Customer requirements figure heavily in tool selection.
20. New tools are introduced through lectures or seminars, usually from 2 to 4 hours in duration. This should expose the programmer to the main features, and he can then acquire the additional information from documents.

## Part II

The key tool discussed in this interview is the software development system. It was generated in-house in 1975 and has since been delivered to the U. S. government. The purpose was to provide a development environment for structured code (including use of assembly language) and top down design which will support maintenance of the code (changes) throughout the development period. This is still the primary use of the tool. The output listing is structured and contains easily understood diagnostics. The tool is used throughout the development phase for all programs in the department.

The tool has increased programmer productivity and improved reliability and maintainability of the product. No cost/benefit trade-off was performed. The need was sufficiently great to indicate the development. All features of the tool are being used, and a number of enhancements are under development. The syntax of the tool is not consistent with current standards. Correction of this problem is under consideration. The tool is always available and has been for the past five years. It is reliable. Maintenance has not been a problem; changes are carefully controlled. Changes are made only in response to an approved, numbered request. In its present form the tool is not suitable for integration into a coordinated environment.

A User's Manual exists and has been found helpful. No special training is being provided. The manual is adequate.

## TOOLS INTERVIEW #18A

## Part I

1. The interview was conducted at one of several departments that develop software of a government-support organization primarily involved with military contracts.
2. The department has 40 engineers and scientists.
3. The programs are used within the government-support organization, primarily by project offices.
4. All staff members are college graduates. About 60% have an advanced degree and 30% are Ph. D.s.
5. A large mainframe is specifically assigned to this department, and several other mainframes are available within the organization.
6. Outside computing is rarely utilized.
7. About 90% of programming is in FORTRAN, the rest in APL. PASCAL and PL/1 are supported but not used at present.
8. For two major projects there are departmental standards. For all other work the individual programmer selects the standards.
9. Both interactive and batch programming are used. Both of the major projects use groups of 4 - 5 programmers. All other programming is done by individuals or teams of two.
10. There is no formal QA organization but testing of the updates of major programs is formal. Specific test decks are being maintained for this purpose.
11. Documentation is being done by the programming groups. In some cases the user objects to the cost of documentation, and the commented source code is then all that is delivered.
12. Changes to the major programs are initiated by a supervisor. All other programs are managed by the individual responsible for them.
- 12A. The major problem is documentation, particularly the identification of changes.
13. This department does not participate in standards activities.
14. The department manager is aware of software engineering activities but does not know whether personnel participate in these.
15. A file manager that came with the operating system of a then new mainframe in the early 1970s was the first tool in use by this department.



16. The file manager and a cross-reference generator are in use at present.
17. No tools have been abandoned.
18. There are no immediate plans for tools acquisitions.
19. The supervisor in charge of the central computing facility decides on tools procurement.
20. The availability of a tool is announced, a presentation is scheduled, and documentation is made available.

## Part II

The key tool discussed at this interview is a file manager. It is a proprietary product of the computer manufacturer and is leased as part of the operating system. It was originally intended for maintenance of the operating system but was soon adopted for management of program files and is now used also for text files. The tool is available by invocation from terminals or at batch submission. The run-time penalty is negligible. The tool is used to change programs and to keep an audit trail of the changes. It can also be used for configuration control. Diagnostics are self-explanatory.

All features are used but it is estimated that 90% of the use involves only 10% of the features. The most frequently used are insert and delete functions. Some enhancements are desired, including automatic dating of changes and the ability to change individual characters. Reliability is good, and maintenance is done by the vendor. The manual supplied with the tool is helpful but also contains much useless information. The tool is regarded as very important, and newcomers to this environment are introduced to it by experienced personnel as soon as possible.



## TOOLS INTERVIEW #18B

## Part I

1. This interview was conducted at another department of the same government-support organization. This department emphasizes programming rather than scientific orientation.
2. 15 programmers are employed in this department.
3. Software users state functional requirements. The department establishes design rules and standards
4. All programmers have at least a BS degree. Most have advanced degrees, including several Ph. D.s
5. Six large mainframes in the central computing facility are used.
6. No outside computing services are used.
7. Programming is 90% in FORTRAN, the remainder assembly.
8. Software design requirements are established by the department. They are "semi-documented"
9. Programming is on-line or batch, by individuals or in small groups.
10. QA procedures have been formulated by the department and are enforced on all updates.
11. The programmer is responsible for documenting his own work to department standards.
12. Wherever possible, the developer performs maintenance. The user participates in configuration control (not all changes made are accepted for operational programs).
- 12A. Lack of personnel with experience in this type of software development is regarded as the major problem.
13. Personnel are not active in standards
14. Personnel attend conferences and are aware of software engineering literature.
15. Some (assembly language) tools were developed in the early 1960's (de-buggers, etc.). These were later re-coded in higher order languages. It is not believed that any of these were ever 'exported'.
16. Many tools are available but general purpose tools are not extensively used by the department. A flowchart program is occasionally used. PDL and PSL/PSA

are available but are not in current use. Code comparators, a specialized management tool, and simulations are used.

17. A simulation tool was abandoned because better tools of equivalent capability became available.

18. No new tools are being considered.

19. The emphasis is on special purpose tools that will be developed in-house. The criterion for tools development is a clearly perceived problem.

20. New tools are introduced by on-the-job training, documentation and lectures.

## Part II

The key tool discussed in this interview is large simulation program that was developed in-house in 1967 and has been ported to several other organizations. The tool is in the public domain. It was developed because set-up and operation of prior simulations was very time-consuming and required very senior personnel. This tool includes a front-end processor which permits the engineers to specify a problem in their own terms. The tool can be applied to any problem that can be structured as a system of differential equations. It finds an optimum solution for whatever cost factors are identified.

The effort to develop and maintain the tool over the past 13 years is estimated at between 55 and 60 man-years. Set-up can vary from a few hours to several weeks. Run time on a large mainframe ranges from seconds to 10 hours. Analysis of the output is required.

Because this is a tool of very broad scope it is not believed that all features are used. The tool has been in use 13 years and it is now rated very reliable. Ease of maintenance was a design consideration. The program is documented in a set of five volumes. Two training programs are periodically conducted. One is for users (engineers); the other one is for newly assigned programmers who will work on the tool.

## TOOLS INTERVIEW #19

## Part I

1. This interview was conducted at a central software facility of a major defense contractor.
2. Approximately 400 full-time professional programmers are employed.
3. The organization develops systems for use by agencies of the U. S. Government. Usually the customer imposes standards but these may be less demanding than company imposed standards.
4. Most programmers are degreed. Experience ranges from one to twenty years.
5. Four large mainframes (three different manufacturers) and a large number of minicomputers are in use. The mainframes can be accessed in batch or interactive modes.
6. A very large mainframe at another facility of the same company can be accessed in time-share mode.
7. Most commonly used are FORTRAN, structured FORTRAN, and a local FORTRAN dialect. COBOL is used for administrative programming. PASCAL and PL/1 are available but seldom used. JOVIAL is currently not much used but may become a major language if a current proposal is accepted. Assembly languages are also used.
8. They always work from formal requirements. After receipt of an approved requirement, a Users Manual is generated for the product which is to be delivered. After this manual has been approved by the customer, a detailed specification is generated. Reviews and updates are also handled quite formally. Research projects are exempted from these procedures.
9. Programming environments are created to suit the needs of specific projects. Practically all programming is done at programming stations (on-line). The chief programmer team concept does not seem to work here, partly because the chief becomes too dominant, and partly because communications bog down when the chief is absent. The present favorite group is a two person team, operating in a single office with a blackboard and two terminals. One of the members is an experienced programmer, the other not necessarily so.
10. There are formal, separate, organizations for Quality Assurance and Configuration Control. They are not part of the software organization. They also do all system level testing and integration.
11. Documentation is the responsibility of everyone who works on a project. Documents are formal and usually in a format specified by the customer.



12. Maintenance is sometimes performed by the programmers who developed a program. But there is also an internal software maintenance organization who upgrades previously issued software. Their functions are governed by the same standards that apply to development.

12A. The main problem is communication among participants in a given project. Another important problem is training employees in structured thinking and logic.

13. They are actively involved in professional standards activities.

14. There is a high level of involvement in software engineering activities.

15. During 1975 they recognized the need for a test coverage analyzer. They developed a tool that was integrated with a pre-compiler.

16. A general software development tool, a hierarchical design language, data bases managers, editors, and several special languages are in use.

17. No tool has been discarded. Tools usually evolve into updated products.

18. A software requirements analysis package is being considered. The decision to acquire it has not yet been made.

19. The Senior Scientist in charge of software technology makes most decisions on tool acquisition. The decisions are based on need, cost, and availability.

20. The introduction of new tools is usually handled very informally. As a minimum, the programmers are given a Users Manual.

## Part II

The key tool discussed in this interview is a hierarchical design language. It was developed in-house in 1977 and has never been delivered as a product (hence is considered proprietary). The original purpose of the tool was to facilitate communications in the design team environment. It is still being used for this purpose throughout the development and design phases. The original development took about one man-year. The current investment is about six man-years. Programmer productivity has significantly improved through use of the tool and software reliability has increased. On one project it had been estimated that each programmer would produce 75 lines of code per month. The actual turned out to be 178 lines. At the end of this project only two design errors and one programming error were discovered.

Set-up is negligible, and run time depends on size of job. Tool resource utilization is not a problem. It is used once or twice a day by each team member, typically in the morning to get a fresh plot for the day. The tool has no options. The tool is being modified to provide for more front-end processing and for more structured programming. Only one reliability problem has been reported since the tool has been put into use. Availability is excellent. Maintenance is being performed in-house.



The tool is a candidate for integration with a general software development tool. A User's Manual was produced but is considered to be marginal. It should be redone. Special training of about one hour is provided for new users.

## TOOLS INTERVIEW #20

## Part I

1. This interview was conducted at one of several programming departments of a Government agency in the defense sector.
2. The closed shop operation involves 60 programmers and analysts. There are also open shop programmers.
3. Most programs are used on-site, but portability is a requirement and is usually provided. There are no substantial outside imposed requirements.
4. The programmers are mostly trained in the applications area served by the programs and have little computer science background. The agency insists on the general background so that people can be readily transferred.
5. Two large mainframes are available for in-house processing.
6. Several specialty mainframes are accessed through the ARPAnet.
7. FORTRAN and some COBOL are used. No other languages are expected in the immediate future.
8. There are no formal software development standards, either external or internal.
9. Programming is done in batch mode, usually in small groups (2 - 3 people)
10. There is no formal QA. When the program has been checked out by the developing group, a readiness letter is issued. It is then gradually transitioned to the user environment, and when they are satisfied an acceptance letter is issued.
11. User Manuals are always prepared. On most programs there is maintenance documentation. They are now introducing standard header formats.
12. Changes are originated by memo from the user, are implemented by the developer.
- 12A. The biggest software problem is the code maintenance due to changing requirements
13. There is no participation in standards efforts.
14. Emphasis is on on-site education in software engineering. One person per year is sent to a graduate school that offers software engineering courses.
15. An editor was introduced around 1970, and a FORTRAN preprocessor was introduced in 1974. Both were of limited use.

16. Several tools developed at the university that furnishes the graduate program are in use (file editor, file scanner, file lister, text editor, tape dump). Also a local program for inserting compiler calls, a checksum program developed by another government agency, and a test analyzer. It is felt that they are limited on the tools that they can apply because the mainframes used here are not the most popular type.
17. A structured pre-processor developed by a government agency was abandoned (no documentation available).
18. It is expected that further test tools will be acquired.
19. All decisions about tools procurement are made by headquarters in Washington.
20. Small tools are announced in a local newsletter and documentation is furnished. Formal training is used when needed.

## Part II

The key tool discussed in this interview is a test analyzer for FORTRAN programs which was developed for this agency by a private company under contract to another Government organization. The first installation was completed in 1977 but did not handle all language forms in use at this facility. The revised version was completed in 1980 and operates on the desired FORTRAN dialects. The tool is owned by the Government but must at present be maintained by the developer. The arrangements for tools acquisition were made in Washington with little involvement of the users (in addition to the facility visited, there is at least one additional user of the tool in similar circumstances). One motivation for use of the tool was to help in transitioning to structured code.

Static analysis runs are made to check out structure. This feature is considered useful. It furnishes a clean structure chart, index to all calls and read statements, and thus helps to generate well structured code. A typical static analysis run takes 10 minutes. The tool is presently used about 10 times per month in this manner. Dynamic analysis is not yet completely operational but uses considerably more computer resources.

This is a very large and comprehensive tool with many options, only a few of which are used at present. It includes restructuring capability which was stated to account for much of the code and will not be used in this environment. In general, the tool is perceived by the user as too large and complex.

The probe routine does not run at present, and this precludes use of the dynamic analysis feature. There are unresolved inconsistencies between the two FORTRAN dialects that are being supported. The tool is at present not maintainable by the user.

Personnel requirements are increased initially. They may be reduced over the long run because program maintenance may be simpler. User Manuals were furnished. There are at present no maintenance manuals. There have been several training classes given by the developer.

## TOOLS INTERVIEW #21

## Part I

1. This interview was conducted at the national data processing center of a Federal agency.
2. The programming organizations consists of:
  - On-line systems, about 30 programmers
  - Conversions and systems design, about 50 progr.
  - Batch programs, about 30 programmers
  - Systems Techniques Branch, about 30 programmers
3. Most programs are used in-house. The on-line programs service nationwide centers.
4. There are a number of GS7 - 9 trainees with very little background or schooling. Most senior analysts have degrees. Typical lead programmers have some college and 3 - 5 years experience. They are well versed in COBOL and JCL.
5. There are four mainframe computers from two manufacturers.
6. They have a time-share arrangement with the local utility for access to a mainframe identical to one of the in-house models for emergency use.
7. COBOL (2 dialects) and assembly are the principal languages.
8. Software development is mostly governed by in-house standards (try to adhere to structured programming, top-down development). Languages are prescribed by Washington headquarters.
9. Mostly programming is on-line (90%), some batch. Programming teams of 3 - 4 people are typically used.
10. There is an audit staff reporting to the Washington headquarters. The output (of programs) is screened by a Control Section. Both of these functions are concerned with fiscal integrity rather than software characteristics.
11. Computer Programming Manuals (CPMs) are on-line. There are formal project write-ups and Computer Change Plans. Maintenance is mostly from flowcharts.
12. Most changes originate in Washington. The computing center generates the Computer Change Plan to implement them.
- 12A: Major problems (from Head of Division): Shortage of staff (partly due to hiring freeze, partly due to lack of applicants with adequate skills); poorly formulated requirements being furnished to the Center which puts an additional burden on the Senior Analysts for clarifying them; better project management reports.



13. There is no participation in standards activities.
14. There is little participation in software engineering activities.
15. A library tool was installed in 1974 but is no longer in use because the programs serviced by it are now on another computer that is not compatible with the tool.
16. Currently used tools include a COBOL optimization program, a hardware utilization analyzer, a sort program and some computer utilization aids.
17. A debug package was abandoned when a better one became available on a new operating system.
18. A new library tool is being acquired.
19. The center can make recommendations but final approval for tools acquisition must be obtained from Washington headquarters.
20. Tools are typically procured by headquarters and are tried out on small development programs. Training is done in-house or by the vendors.

#### Part II

The key tool discussed in this interview is a hardware utilization analyzer. The tool is leased from the developer. This program summarizes and formats utilization data accumulated by the operating system. The tool is used once a day, and a run takes several minutes. The tool furnishes a number of reports on equipment usage, only some of which are printed out while others are being archived on microfiche. Because of the heavy load on the computers the group operating the tool would like to use it for tuning of programs but time divisions captured by this tool are not sufficiently fine to be effective for tuning. The head of the division was well aware that the tool was not suitable for tuning but placed heavy reliance on the reports furnished by it to show the average level of operation, and particularly that "no catastrophes had occurred".

The tool is reliable and well integrated with the operating system of the computers serviced by it. Finer time divisions would make it suitable for tuning, but this function may be better accomplished by special tuning aids. Some accounting functions of the tool duplicate information furnished by the operating system but the tool prints them out in a more systematic manner. A User Manual is furnished and is rated fair. No special training was provided with this tool.

## TOOLS INTERVIEW #22

## Part I

1. This interview was conducted at one of many programming groups at a government-support laboratory. The specific group is a service organization supporting the research groups at the laboratory.
2. There are twelve people who work on both hardware and software plus the equivalent of 4 full time contractor personnel who work software only. The software effort amounts to about 12 people.
3. Most of the software is used by the operators of major test facilities at the laboratory. Users are not very sophisticated in programming and don't impose requirements.
4. Most staff are degreed (EE); contractor personnel are computer scientists with 4 - 5 years experience.
5. Most of the software is developed on and targeted for minicomputers.
6. No off-site equipment utilized. There is a central computing facility at the laboratory which they sometimes use for cross-assembly.
7. The languages are mostly FORTRAN IV and ANSI FORTRAN. There is emphasis on making the central part of the program very portable. ISA Standards S61.1 and S61.2 are used for calls. A structured FORTRAN pre-processor is used. There is interest in PASCAL, and an in-house course has been taught. The interface with external routines is a problem.
8. Programming standards are informal. Modular structure (single page modules) and structured programming are in use.
9. All programming is done on-line. Small programs are done as individual efforts, larger ones in a group headed by a Project Engineer, who has responsibility for hardware and software.
10. Team leaders are responsible for testing. There is no other QA. Testing is done top-down and bottom-up.
11. Documentation is done by programmers. User's Guide is standard. On larger programs there may also be a Programmer's Guide. Average program life is 4 - 5 years.
12. Changes are usually originated by a phone call to the Project Engineer. Versions are identified in the header.
13. There is no participation in standards activities.
14. Most personnel participate in functional professional activities but there is increasing interest in software engineering meetings.

15. In 1966 they produced a cross-assembler, in 1968 a cross-reference generator.
16. A FORTRAN preprocessor, text editors, and debug aides are currently used.
17. The cross-assembler was abandoned because one is now available from the computer manufacturer.
18. A performance measurement system for the most frequently used minicomputer will be acquired.
19. The group leader makes the decisions on tools acquisition.
20. Tools introduction is by manuals only. Tools should be easy to use and not require special training.

## Part II

The key tool discussed at this interview is a structured FORTRAN preprocessor which was developed at a government agency and is in the public domain. It has Macro capability, furnishes indented listings, and can handle some special constructs. It is a very fast pre-processor, with practically no compile penalty. There is some run-time penalty because the code is less efficient than native FORTRAN. About one-third of the programmers use it now. The greatest benefit is felt to be the neatly indented listing of the source.

All features of the tool are used. A bothersome weakness is that the FORTRAN code is 'not readable' and can not be easily reconciled with the source. Also, the tool does not at present permit INCLUDE statements. This latter feature is being added locally. Reliability is good. Maintenance is a problem because the FORTRAN listing in which the preprocessor is furnished is difficult to understand. This group would like to see an integrated tools package, ideally from program definition through test.

The User's Manual furnished with the tool is rated as fairly useful. There was no special training required to use the tool.



## TOOLS INTERVIEW #23

## Part I

1. This interview was conducted at the software division of a Government organization in the defense sector.
2. Approximately 300 programmers are controlled by this division (most are contractor personnel accessing the facility from remote terminals). About 12,000 computer runs were processed by the facility in 1979.
3. The ultimate user is the military service. The division acts as the procurement agency for the user. User establishes functional requirements, and the division establishes quality requirements. In most cases the detailed requirements are generated by the developer. In recent procurements these adhere to MIL-STD-1679.
4. Programming trainees comprise about 20% to 30% of the personnel, and the upper 5% to 10% are highest level computer professionals. The rest are journeyman programmers. The in-house staff is mostly degreed, about one-half have an M.S. in a computer related field.
5. The central computer complex includes 5 very large mainframes, all of the same manufacture. There is also one midrange computer and a number of smaller machines.
6. No outside computing resources are utilized.
7. The target software is mostly in CMS-2M, CMS-2Y, and SPL.
8. The computing facility permits auditing for specific software quality requirements and also supports configuration control.
9. Interactive and batch submittals are used. The composition of development teams was not known because these are located off-site.
10. QA is performed within the development function.
11. Documentation of target software is handled by the developer.
12. There is considerable involvement of the end user in changes. If possible, these are accomplished as scheduled updates.
- 12A. Major software problems arise in requirements and design.
13. The organization participated in formulation of MIL-STD-1679. There is no involvement in non-military standards.
14. There is participation in professional activities. Several papers on the facility have been presented.



15. An early, but not necessarily first tools acquisition was a test analyzer procured in 1975 and now incorporated in the facility.

16. Current tools include the test analyzer for FORTRAN and SPL/1, several library systems, simulators, emulators, and utility packages.

17. An early Very High Level Language was abandoned because it was not well documented, did not yield consistent results, and had "lots of hidden little things".

18. A requirements analyzer will be acquired. Two competitive products are under consideration.

19. The superintendent decides on tools procurement. For the requirements analyzer procurement a formal evaluation criteria document has been generated and a benchmark has been selected. Tools cost is absorbed in the organizational budget.

20. Tools are 'advertised' to users, demonstrations are held and the users are encouraged to try the tools.

#### Part II

The key tool discussed in this interview is the test analyzer which was commissioned about 1975. The tool is government property. It is now being maintained by a contractor (not the developer). It is used for both static and dynamic analysis. Set-up time has been minimized by eliminating most options so that it can be invoked with just two control cards. The tool is not available for interactive use but can be run by RJE. The end user pays for run time on computer so that the tool usage is at no (or very little) cost to the developer. The target computer is simulated interpretively while this tool is being used. This normally causes a further considerable run-time expansion but it is made tolerable here by the very high computing performance of the host computer. On a recent major project the analyzer was used to obtain test coverage of 80% - 90%.

Reliability is now acceptable. There were some initial problems. The tool handles now two languages and others can be added. The nature of the analysis implies some language dependency. The tool makes use of the integrated data base being maintained at the computer facility. Test data sets can be input from the data base, and test reports are output to the data base.

Documentation gradually evolved as the tool was being used and modified. Current documentation is believed to be useful and adequate. No special training was furnished. The computer facility personnel were familiar with this type of tool.

There is a hierarchy of user documentation on the overall facility which might well be a model for the types of manuals required for an integrated software tools system:

General Facility Brochure - Introduction to capabilities

Management Summary - benefits to program management

Production and Maintenance Methodology - Identifies use in day-to-day tasks, directed at engineers and programmers

Handbook - a concise "how to" manual

Facility User Manuals - detailed description of commands and special features (organized by target computers).

## TOOLS INTERVIEW #24

## Part I

1. The organization volunteered to supply data for the tools survey in the form of written responses to the interview questionnaire. The information is furnished for the software section of a specialty computer manufacturer.
2. 130 programmers and analysts are employed in the software function.
3. Embedded software is sold as part of the product to the user. The user does not impose standards; in-house requirements are implied.
4. The staff ranges from beginners to highly skilled and experienced computer scientists.
5. Software is being developed on the target mini- and midsize-computers.
6. No outside facilities are utilized.
7. Present languages are FORTRAN (80%) and assembly (20%). PASCAL and Ada are being considered for the future.
8. At present the format and content vary considerably from product to product. Some standards are in use for naming of variables and for header formats. The company is moving toward greater uniformity with each new software development effort. Written requirements will in the future be mandatory. Formal reviews and configuration control are also being instituted.
9. Most programming is done by individuals or small groups (fewer than 3). Occasionally up to 6 people may be involved in an effort. Most code is first put on coding sheets and then keyed in by programmer aides, using a terminal. Programs are compiled and executed in batch mode.
10. A software QC function has been established but currently operates on an ad hoc basis. The larger of two programming organizations has formal acceptance groups. The other one has identified a QA function in a recent document.
11. Documentation is the responsibility of the individual program managers and usually depends on the contractual requirement. A basic set of company wide documentation standards is planned.
12. There are "standard" products which are supported by the company and for these there is an established problem reporting and update procedure. For custom products the customer usually assumes responsibility once they are accepted.
- 12A. Major software problems: Configuration control (during development), maintenance difficulties, inaccurate estimates.
13. Participates in software standards (recent)

14. At least two, and probably not more than six people keep abreast of current literature and attend at least one meeting a year.

15. A set of dynamic analysis tools have been acquired but are not yet in use.

16. An internally developed tool is in use for allocating Global Common in large simulation programs. This tool also inserts comments describing the variables. An interactive symbolic debugging tool is also in use.

17. No tools have been abandoned.

18. A general software development tool is under consideration.

19. Tool procurement is decided on by the manager of software development. The prime criterion is projected cost savings.

20. Tools are introduced through documentation and informal training. Training will probably be formalized as more tools are being used.

No part II was furnished by this organization.



## TOOLS INTERVIEW #25

## Part I

1. The organization volunteered to participate in the tools survey and supplied the information by phone. It is a software group at the corporate research center of a heavy industry company.
2. Staff at this facility comprises 7 programmers and analysts. The procedures used here are unique and not representative of company wide programming practices.
3. Research and special projects are handled. At some occasions this has involved safety-critical software for nuclear plants. Standards are generally self-imposed.
4. Lowest skill level: Associate in Computer Science, fresh out of school; highest: advanced degrees plus 10 years experience.
5. On-site equipment includes two minicomputers.
6. Two large mainframes are accessed via tie-lines.
7. FORTRAN and structured FORTRAN are the primary languages. Looking at Ada for the future.
8. Top-down structured design and programming are being introduced. The test bed for exercising safety-critical software was done in this manner.
9. Both individual programming and small teams are utilized. Most programming is interactive.
10. The conduct of QC is informal (can be assigned to any individual); however, reports on QC are usually issued.
11. Design reports and QC reports are issued. These are considered adequate. As an example it was cited that one program had not been used for two years and was then successfully modified for a new application.
12. There are formal procedures for software changes.
- 12A. Definition of requirements is seen as the most urgent problem.
13. There is participation in standards activities
14. There is participation in software engineering activities.
15. A Structured FORTRAN preprocessor has been in use for several years and is liked. It facilitates forward and backward tracing between structured code and FORTRAN.

16. A number of requirements and library procedures are now implemented manually to get familiar with them prior to committing to a tool.

17. One requirements analyzer was tried and given up as too complex

18. Tools for management of the development process will be acquired.

19. The supervisor of the development staff selects tools for experimental use based on perceived needs. Commitment to use is made only after good results from experimental use are established.

20. Initial selections are made from manuals and presentations at technical meetings. For a small group, the effort to install a tool is formidable, and therefore manual methods that duplicate what a tool will do are justified to determine the utility of the data.

No part II was obtained from this organization.

## TOOLS INTERVIEW #26

## Part I

1. The organization volunteered information for the tools survey. The following responses were obtained by mail. They pertain to a regional office of a software company.
2. Thirty programmers and analysts plus some support personnel are employed in this office.
3. No answer provided.
4. Some employees are still working toward a bachelor degree, others are Ph. D.s. One-half have advanced degrees.
5. A minicomputer is utilized.
6. There is no use of off-site computers.
7. FORTRAN, structured FORTRAN, and a derivative of PL/1 are used.
8. Software requirements are generated in-house and are submitted to the customer for approval.
9. Programming is done by individuals or in groups of up to five people. Both on-line and batch are used.
10. No answer furnished.
11. Programmers are responsible for documentation. A standard outline is followed. Sometimes comments are extracted for documentation.
12. Maintenance is assigned to someone familiar with the project but not necessarily the original programmer.
- 12A. The most urgent problems are maintenance (of someone else's programs) and incomplete requirements.
13. There is no participation in standards activities.
14. They are aware of software engineering activities but are not involved in them.
15. The first tools usage is not known.
16. At present a FORTRAN preprocessor, a portability auditor, and editing and debugging tools are used.
- 17 - 19. No answers furnished.

20. Tools are introduced with the aid of documentation or from observing their use.

#### Part II

The key tool described by this organization is a preprocessor for structured FORTRAN which is in the public domain. It is used to promote structured coding, and it is stated that correct programs are generated in about one-half the time that it would take without the use of this tool. Also, readability and maintainability of the programs is improved and a beneficial effect on software quality is reported.

The tool is reliable. No maintenance is performed by this organization. Most features are used. A desired option, not currently available, is that comments not be stripped and blanks deleted in the FORTRAN output. Integration is difficult because most tools don't operate on the structured source code used by this preprocessor.

The User's Guide furnished with the tool was suitable. It is believed that deviations from 'standard' installations need to be documented better. No special training is necessary for this tool.



SECTION 5  
TOOLS CLASSIFICATION

The classification of key tools encountered in the survey is presented in Table 5 - 1. The tools are arranged in interview order. Where more than one key tool was described in an interview, a dash number has been appended; the designation thus formed is consistent with that used for the Part II interview synopses in Section 4. The definition of the terms used in this classification is given in [2].

TABLE 5 - 1 CLASSIFICATION OF KEY TOOLS

<u>Intervy</u> <u>No.</u>	<u>Inputs</u>		Transfor- mation	<u>Functions</u>		<u>Outputs</u>	
	Tool	Control		Static Analysis	Dynamic Analysis	User	Machine
1-1	Data		Formating	Cost Estim. Management Scheduling		Graphic Tables	Data
1-2	Code		Translation			Graphic Listing Tables	
2	Code	Command	Editing Restructrg.	Management Scanning		Listing Text	OCode SCode
3	Data	Command	Editing Formating	Management Scanning		Text	Data
4A-1	Data	Command			Simulation	Text	Data
4A-2	Text		Editing Formating			Listing	
5	Code	Command	Translation			Listing	OCode
6	Code	Command	Editing Translation	Management Scanning	Tracing	Diagnst Listing Text	OCode SCode
7	Code			Auditing Management Scanning Struct.Chk	Tracing	Listing Text	SCode

OCode = Object Code

SCode = Source Code

<u>Intervy No.</u>	<u>Inputs</u>		Transfor- mation	<u>Functions</u>		<u>Outputs</u>	
	Tool	Control		Static Analysis	Dynamic Analysis	User	Machine
8	Data	Command		Management Resrc.Est.	Resrc.Util.	Diagnst Text	Data
9	Code		Restructrg.	Audit Scanning		Diagnst Listing	SCode
10B	Data			Comparison	Cover.Anal. Timing Tuning	Graphic Text	Data
11A	Data	Command			Simulation	Text	Data
11B		Command	Translation			Diagnst Listing	Prmpt
12A	VHLL		Translation	Crossref.		Graphic Listing Table	
12B	Code		Restructrg.	Audit Scanning		Diagnst Listing	SCode
13-1	Text		Editing Formating	Management		Text	Data
13-2	VHLL		Formating	Compl.Chk. Consist.Ck DataFl.Anal. Tracking		Graphic Listing Text	Data
14A	Code			Crossref.		Listing Text	
14B	Code	Command	Editing Formating Translation	Comparison Crossref. Management Scanning Tracking	Tracing	Diagnst Graphic Listing Text	OCode SCode

OCode = Object Code

SCode = Source Code

<u>Interv</u> <u>No.</u>	<u>Inputs</u>		Transfor- mation	<u>Functions</u>		<u>Outputs</u>	
	Tool	Control		Static Analysis	Dynamic Analysis	User	Machine
15	Code	Command	Formating Instrument.	Statistical	Cover.Anal. Timing	Listing Tables	SCode
16	Code	Command	Restructrg. Optimizat'n	Struct.Chk.	Cover.Anal. Tuning	Graphic Tables Text	SCode
17	Code	Command	Editing Restructrg. Translation	Management Scanning		Listing Text	SCode
18A	Code	Command	Editing Formating	Management Scanning		Listing Reports	OCode SCode
18B	Data	Paramet.			Simulation	Computr Resit Listing Reports	Data
19	VHLL		Editing Translation	Managment		Graphic Listing Tables	
20	Code	Command	Instrument. Translation	Statistical	Cover.Anal.	Listing Reports Tables	SCode
21	Data	Paramet.			Resrc.Util.	Graphic Text	Data
22	Code	Command	Restructrg.			Diagnst Listing	SCode
23	Code	Command	Instrument. Translation	Scanning Statistical	Cover.Anal.	Listing Reports Tables	SCode

OCode = Object Code      SCode = Source Code

REFERENCES

[1] H. Hecht, "Final Report: A Survey of Software Tool Usage", NBS Special Publication 500-82, October 1981

[2] Raymond C. Houghton, Jr., "Features of Software Development Tools", NBS Special Publication 500-74, February 1981.



U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> (See instructions)	<b>1. PUBLICATION OR REPORT NO.</b>	<b>2. Performing Organ. Report No.</b>	<b>3. Publication Date</b>
<b>4. TITLE AND SUBTITLE</b>  Synopsis of Interviews from a Survey of Software Tool Usage			
<b>5. AUTHOR(S)</b> Herbert Hecht, SoHaR, Inc.			
<b>6. PERFORMING ORGANIZATION</b> (If joint or other than NBS, see instructions)  SoHaR Incorporated 1040 S. LaJolla Ave. Los Angeles, CA 90035		<b>7. Contract/Grant No.</b> NB79SBCA0273	<b>8. Type of Report &amp; Period Covered</b>
<b>9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS</b> (Street, City, State, ZIP) National Bureau of Standards Department of Commerce Washington, DC 20234			
<b>10. SUPPLEMENTARY NOTES</b>  <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
<b>11. ABSTRACT</b> (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)  This report contains synopses of all interviews conducted as part of a survey of the development and use of software tools in industry and Government and a classification of the tools encountered. The purpose of the report is to document the detailed data in a form which may be helpful for further research.			
<b>12. KEY WORDS</b> (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) Programming aids; software automation; software development; software engineering; software tools; software tools usage.			
<b>13. AVAILABILITY</b>  <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input checked="" type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.  <input type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		<b>14. NO. OF PRINTED PAGES</b>  79	<b>15. Price</b>



U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> <i>(See instructions)</i>	<b>1. PUBLICATION OR REPORT NO.</b> NBSIR 81-2388	<b>2. Performing Organ. Report No.</b>	<b>3. Publication Date</b> November 1981
<b>4. TITLE AND SUBTITLE</b> Synopsis of Interviews from a Survey of Software Tool Usage			
<b>5. AUTHOR(S)</b> Herbert Hecht, SoHaR, Inc.			
<b>6. PERFORMING ORGANIZATION</b> <i>(If joint or other than NBS, see instructions)</i> SoHaR Incorporated 1040 S. LaJolla Ave. Los Angeles, CA 90035		<b>7. Contract/Grant No.</b> NB79SBCA0273	<b>8. Type of Report &amp; Period Covered</b>
<b>9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS</b> <i>(Street, City, State, ZIP)</i> National Bureau of Standards Department of Commerce Washington, DC 20234			
<b>10. SUPPLEMENTARY NOTES</b>  <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
<b>11. ABSTRACT</b> <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i>  This report contains synopses of all interviews conducted as part of a survey of the development and use of software tools in industry and Government and a classification of the tools encountered. The purpose of the report is to document the detailed data in a form which may be helpful for further research.			
<b>12. KEY WORDS</b> <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> Programming aids; software automation; software development; software engineering; software tools; software tools usage.			
<b>13. AVAILABILITY</b> <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		<b>14. NO. OF PRINTED PAGES</b> 79	<b>15. Price</b> \$9.50







