NBSIR 82-2482

A Survey of Software Validation, Verification, and Testing Standards and Practices At Selected Sites

Boeing Computer Services Company
Space and Military Application Division
Seattle, WA 98124

April 1982

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

NBSIR 82-2482

A SURVEY OF SOFTWARE VALIDATION, VERIFICATION, AND TESTING STANDARDS AND PRACTICES AT SELECTED SITES

Boeing Computer Services Company
Space and Military Application Division
Seattle, WA 98124

April 1982

Sponsored by:
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Programming Sciences 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

LIST OF FIGURES

ABSTRACT

A survey of software validation, verification, and testing(V,V&T) practices at five governmental and five commercial sites was performed. The survey collected information describing each site environment, software development/maintenance practices, the V,V&T techniques and tools employed, and standards and/or procedures guiding the activities at each site. This report summarizes the information obtained and presents observations about current operations with respect to software development, maintenance, and V,V&T. It also includes reports discussing each of the sites surveyed, and the survey instruments used.

KEYWORDS:
Software validation and verification, environment, standards, software development and maintenance, V,V&T techniques and tools.

ACKNOWLEDGEMENTS
This report was funded by the National Bureau of Standards Institute for Computer Sciences and Technology under U.S. Department of Commerce Contract NB79SBCA0102. The authors of the report as submitted by Boeing Computer Services were Mark K. Smith and Donna R. Hudson with Linda S. Hammond, Robert E. Hite, Vernon G. Leck and Randall L. Merilatt. The views and conclusions expressed are those of the authors and do not necessarily represent the official policies of the Department of Commerce or the United States Government.

PREFACE

The following report was submitted by Boeing Computer Services to the Institute for Computer Sciences and Technology(ICST). With the exception of minor editing changes made by ICST, the document remains as submitted.

The survey provided information instrumental in the preparation of two other documents: "Validation, Verification, and Testing Technique and Tool Reference Guide" (to be published) and "Guidelines on Planning for Software Validation, Verification, and Testing" (to be published). It can be used to assist in the assesment of current validation, verification, and testing (V,V&T) practices in governmental and commercial software development sites.

The report, being prepared under contract to ICST, is in the public domain and is, therefore, not subject to copyright.

Patricia B. Powell
Systems and Software Technology Division
Institute for Computer Science and Technology
National Bureau of Standards

1.0 PROJECT OVERVIEW

1.1 INTRODUCTION

1.1.1 NBS/ICST Software Validation, Verification, and Testing Studies

The National Bureau of Standards' Institute for Computer Sciences and Technology(ICST) has a mission under Public Law 89-306 (Brooks Act) to develop standards to enable the "economic and efficient purchase, lease, maintenance, operation, and utilization of automatic data processing equiptment by Federal Departments and agencies." As part of its current program, ICST is studying methodologies and techniques to ensure the development of quality software. Validation, verification, and testing (V,V&T) throughout software development plays a major role in fostering software quality. In preparation for the development of a general guideline on software V,V&T, a limited survey of current V,V&T practices in both the government and in the private sector was performed. This report provides a summary of that survey.

V,V&T efforts are those procedures, activities, techniques, and tools used to increase confidence in the software being developed. The notion that V,V&T is a confidence raising process implies increased assurance that the software will operate as intended with relatively few errors (an acceptable number) in the final products. The class of errors being addressed here is very broad. It includes deficiencies in the software, such as unsatisfied requirements, or the converse, the inclusion of extraneous functions and/or attributes. An error may be in the computer program, a specification of the software, (e.g., a requirement or design specification), or the software documentation (e.g., a user's manual). The error might be related to the functional correctness of the software or some other property, such as performance, reliability, portability, or more subjective attributes such as robustness, understandability, or maintainability.

1.1.2 Survey of V,V&T Standards & Practices

The objective of the survey was to collect data describing current V,V&T practices. Information was gathered through interviews at five governmental and five commercial sites. These sites were selected so that a broad spectrum of environments and applications were covered.

In the survey, the application of V,V&T practices, tools, and techniques was investigated across varying environments and applications. The prevalent attitudes towards V,V&T were also studied. History and evolution of V,V&T practices were investigated, as were the perceived cost-benefits. Information was collected from both the technical personnel and the management involved in V,V&T activities.

1.1.3 Other Related Reports

Two other reports, to be published, which utilize the information gathered in the survey are:

- Guidelines on Planning for Computer Software Validation, Verification, and Testing

- Validation, Verification, and Testing Technique and Tool Reference Guide

There are two other related publications:

- Validation, Verification, and Testing of Computer Software, NBS Special Publication 500-75
- Validation, Verification, and Testing for the Individual Programmer, NBS Special Publication 500-56

1.1.4 Contents of This Report

Sections 1.2 and 1.3 describe the survey, including site selection and the survey activities. Section 2 is a summary of survey results across all sites. Section 3 gives a general summary and conclusions. Appendices A through J are individual site reports and Appendix K is the survey instrument.

1.2 SURVEY OVERVIEW

1.2.1 Site Selection

With the assistance of the NBS/ICST staff, the five commercial sites and five government sites were chosen. The selection of these sites was based on a number of factors, including: 1) the commitment of the management at each site to participate; 2) the identification of a key contact to assist with and coordinate the site visits and interviews; 3) the identification of particular projects and/or activity centers at each site to be studied; 4) the need to cover a broad spectrum of software environments. The sites surveyed represented financial, health, manufacturing, public utility, and Federal government communities. This project specifically identified military and real-time applications as beyond the scope of the study.

1.2.2 Abstract of the Survey

Identification of survey information was required prior to contacting the survey participants. The information identified needed to:

- characterize the ADP applications and environment, the management philosophy and structure, and other factors related to the ADP organization,
- provide a history of software development lifecycles and project management (with an emphasis on quality assurance and V,V&T activities),
- provide a list of current tools, techniques, and standards, and
- determine attitudes toward standards (particularily V,V&T), desires concerning format, and style of potential standards.

Once the information requirements had been identified, a survey questionnaire was developed. The questionnaire (Appendix K) was used as a framework for interviewing people at each of the selected sites.

The questionnaire was organized in three parts, according to the type of information requested and the audience. Part 1 of the survey addressed characteristics of the software development environment (e.g., project sizes, application areas) and the standards and procedures employed (e.g., development phases, the documentation produced, change control practices). This section was responded to by management personnel. Part 2 of the survey was used in interviewing both management and technical personnel and was primarily concerned with V,V&T practices and factors affecting those practices. Part 3 of the survey was responded to by technical staff and collected information on the V,V&T techniques and tools used.

1.3 SURVEY METHODOLOGY

1.3.1 Pilot Survey

Before the survey was administered to the selected organizations, a field test was done at a pilot site (one of the 10 sites). The results from this effort were analyzed to assure the completeness and clarity of the survey instrument. Revisions were made before the other sites were visited.

1.3.2 Survey Kick-off Meetings

Prior to the site interviews, two survey kick-off meetings were held attended by representatives of the sites. The objectives of these meetings were to:

- introduce the NBS Software Validation, Verification, and Testing Study to the participants,
- present an overview of the information the survey addressed,
- hand out the survey instruments, and
- identify the management and technical staff that would complete the questionnaire and be interviewed at each site.

1.3.3 The Interview Process

The sites were visited by teams composed of one to three people. The purpose of the site visits and interviews was two-fold. First, the questionnaire responses were reviewed with participants for clarification. Second, participants were asked to elaborate upon current practices, problems, and needs of their environment.

1.3.4 Preparation & Review of Site Reports

The information obtained via the questionnaire and interviews was summarized in a report by the team for each site. Each site report was then sent to the participants for a review to assure that the information in the report was correct.

2.0 REPORT OF SURVEY RESULTS

The following sections identify characteristics of the various environments studied at each site. It is important to note that this is not a comprehensive examination of all computing activities within any given organization; the scope of such an exercise is significantly larger than permitted by this project. (Commercial sites are identified as C1, C2, etc., and government sites, G1, G2, etc.)

2.1 ENVIRONMENTS SURVEYED

Within the 10 sites, there were actually 20 different distinguishable groups surveyed. These included organizational entities, working groups, and project teams. Nine major functional activities were identified. A tally of the groups by these activities is given below:

Software Development - 6 of the groups were involved almost exclusively in development activities. These six groups can be further categorized as 2 large projects (C2 & C1 - a major projects division), 2 mid-range development projects (both at C5) and 2 development groups (G1 and G5).

Software Maintenance - 5 groups were primarily involved in activities involving existing systems including fixes, enhancements, and special requests (e.g., special reports from databases of existing systems). This last activity is not usually included in the definition of maintenance, but at least two sites (C1 & G1) spent a significant level of effort on this.

Combined Development and Maintenance - At two sites (C3 & C4), the groups surveyed commonly performed activities in both development and maintenance. C3 was roughly two-thirds to three-fourths maintenance (including system enhancement). C4 was divided along functional/customer lines where each subgroup performed both development and maintenance activities for a given customer.

System Support - At C1, the systems (operating system, utility, and support software) group was interviewed.

System/Software Conversion - One site (G2) was involved in a very large total conversion effort involving a hardware upgrade and software conversion from assembly language to COBOL. This essentially involved a redevelopment since the systems were being respecified. To assist this activity, new development standards and techniques were being defined/adopted.

Requirement Specification & Validation - At site G3, one of the groups interviewed functioned as a user interface between the development staff and the end users of the systems and associated data. The role of the group was to translate user needs into requirements and to develop and execute a validation plan that would ensure satisfaction of the requirements.

Research and Development - At site C3, one of the groups surveyed had several responsibilities lumped under the title of R&D. These activities included independent testing and change control.

Standards Development - At two sites (G3 and G5), groups were interviewed which had the function of developing general standards and guidelines.

Programming Support - The group surveyed at G4 was a systems development and maintenance support group. They provided various types of programming to a variety of computer center customers.

The following sections further describe and summarize selected characteristics of the ten sites and the environments within each.

2.1.1 Application Areas

	Application Type (a)					Processing (b)		
Site	Management Support	Administration	Operation	Engineering	Program Administration	Transactional	Statistical	Computational
G1	*	*		*		*	*	*
G2					*	*		*
G3					*	*		*
G4								
G5			*			*		
C1	*	*	*			*	*	
C2			*			*		
C3	*	*	*					
C4	*	*	*			*		
C5		*	*			*		

FIGURE 2.1.1-1 APPLICATIONS

The information presented in Figure 2.1.1-1 identifies the types of application with which the groups surveyed were involved. The first half of the figure categorizes the applications with respect to the user viewpoint, (i.e., the type of function that the systems support); the second half describes the type of processing performed by the systems being developed and maintained. Since G4 is a service group involved in diverse activities, it is not characterized in Figure 2.1.1-1. The following describes the headings in the figure.

Management Support - Special processing and reporting requests in addition to information normally supplied by the administrative and operational systems reports.

Administration - Includes the fiscal and personnel oriented application areas, e.g., budget, payroll, personnel.

Operation - Includes systems that support the business of the site, e.g., accounts payable/receivable, inventory, orders.

Engineering R&D - Includes both statistical information processing and engineering design applications.

Program Administration - Responsible for the administration of one or more large Federal programs, i.e., collection and distribution of funds, qualification and tracking of recipients, and interrelationships between programs.

Transactional - Revolves around large databases or data files usually with applications involving both periodic sequential processing and individual transaction processing.

Statistical - Involves tabulation and collection of simple statistics (in an analytical/investigation mode rather than the periodic tables and statistics produced by administrative and/or operational systems).

Computational - Involves complex logic and/or intricate computational algorithms.

Some observations regarding the applications area and types of software activities were made:

- Where groups were involved with management support, program administration, or administrative and operational systems, projects were predominately small and maintenance oriented involving fixes, minor enhancements, special report enhancements, and special reports for management support.
- Larger enhancements and/or new developments were usually driven by the operation of an existing system (manual or automated).The problem being solved (and thus the requirements for the software) was drawn from the existing operation or changes to the operation.
- Projects producing software for 'one-time' use were common for management support (often special reports utilizing a database management system (DBMS) and/or query system), and in the engineering R&D areas (usually more statistical, computational and for tabulation of information often using packaged software).
- In the engineering operations and program administration area several mid to large scale projects were encountered, e.g. customer transactions and records (C2), on-line ordering system (C1), a new Federal program (G3), and a new R&D project (G1).
- There were four sites where the business was regulated by Federal and/or State laws (C2, C3, G2, G3,). Changing regulations and strict time frames for implementing these changes put very stringent constraints on the software development activities at these sites.

Three factors, the application area, the type of processing performed (Figure 2.1.1-1(b)), and whether or not the systems were in existence or being developed, influenced to a large degree the size of projects, the constraints affecting the projects, and ultimately, the technology employed in these environments.

2.1.2 Development vs. Maintenance Activities

From the information obtained, it was not always possible to estimate the time spent on new development versus maintenance. As stated previously, six of the groups surveyed were primarily involved in development, five were predominately maintenance groups, and three were involved in both. In analyzing the data collected from these groups, it was apparent that estimating the distribution of resources (money, personnel, hardware utilization, etc.) according to development versus maintenance would be very misleading. Accurate data are hard to assemble for several reasons. First, the groups were not representative of the total organization. Second, there are problems in defining terms and collecting data according to these definitions. For example, in maintenance oriented groups, enhancement to existing software was considered new development. An example of a specific project that does not fit directly into either the development or maintenance category is the conversion effort at site G2. It could perhaps better be described as a redevelopment rather than a direct conversion. Third, in many environments, there is a significant portion of the resources that is spent on activities that are not development or maintenance, e.g., development of standards, configuration management, or quality assurance.

There are some observations to be noted with respect to each type of environment. Maintenance environments were generally characterized by:

- smaller projects,
- informal relationships with customers,
- customer involvement primarily during initiation (definition of requirements, usually in the form of a request for services) and final acceptance,
- more tightly constrained/bounded problems (because of the interface to an existing system),
- tighter schedule constraints (because of changes being made to production software),
- resistence to the introduction of new technology because of constraints, (e.g., schedule, momentum of the environment), and often the 'state' of the system(s) being maintained (e.g., lack of documentation and patchwork structure of the code).

In comparison, new development environments appear to:

- have larger, better defined projects,
- operate under fewer constraints,
- attempt to incorporate new technology with the initiation of projects,
- involve the customer/user to a greater extent.

2.1.3 Size of Projects

Classification of software project size not only varied from site to site, but also from organization to organization within a site. The survey requested a classification (small, medium, large) of projects in terms of level of effort (person months). The classifications seemed to reflect the distribution of projects at each site, rather than a general size classification. For example, the classification scheme of the maintenance group at site C1 divided projects into 5 groups ranging from the smallest projects of less than 40 hours, to the largest projects greater than 5 months. This classification was used to differentiate between the small projects which were characteristic of that particular environment.

	C1	C2	C3	C4	C5	G1	G2	G3	G4	G5
Small less than ____ person months	.25	6	1	6	12	2	10	1	4	3
Large greater than ____ person months	5	24	12	18	36	15	36	6	-	12

FIGURE 2.1.3-1 PROJECT SIZE
(SITE AVERAGES IN PERSON MONTHS)

Figure 2.1.3-1 lists (in person months) some of the upper limits given for small projects and the lower limits given for large projects. As can be seen from the table, there is a wide variance. This is largely due to the diversity in the environments surveyed, and to a certain extent, the activities included in a typical project at each site. For example, some of the figures are for coding and debugging and do not include previous or subsequent activities, nor the time spent by the customers/users in definition and/or acceptance testing. It appears that the resource accounting practices vary widely across all sites and help to account for these differences.

2.1.4 Languages

	C1	C2	C3	C4	C5	G1	G2	G3	G4	G5
COBOL	*[1]	*	*	*	*	*		*	*	*
Assembly	*						*			
PL/1	*									

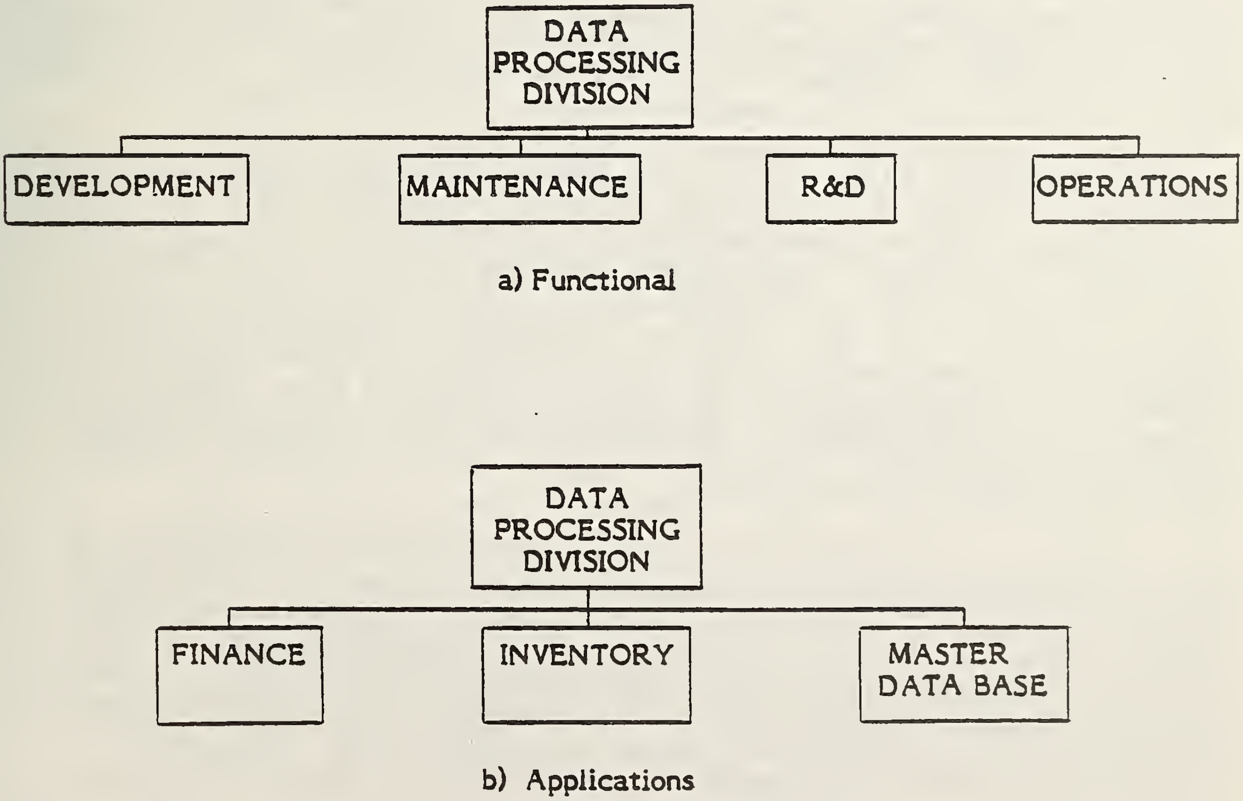
[1]-approx. 90%

FIGURE 2.1.4-1 LANGUAGES

With the exception of one site (G2) that was using only assembly language, COBOL was the primary language being used. DBMS query languages and some report generator packages were also used, but did not constitute a significant portion of the software at any of the sites. Figure 2.1.4-1 is a summary of languages reported in use at the various sites. Individuals interviewed noted that they were estimating language usage and that it would be difficult to find a complete and up-to-date catalogue of systems (and their characteristics) used throughout the organization.

2.1.5 Organizational Structures

Eight of the ten sites interviewed had all data processing activities concentrated within a single independent organization (not staff) that reported to high-level management. In general,it was found that the section or division was patterned after one of two basic organizational models (Figure 2.5.1-1).



a) Functional

b) Applications

FIGURE 2.1.5-1 ORGANIZATIONAL MODELS

In the functional model, charters for the Development group typically included all new development. Maintenance activities were consolidated within a second group. Major enhancements at some sites were handled as maintenance projects, but occasionally would be treated as new development. Operations included computer center operations, production scheduling and control, and system maintenance. R&D encompassed budget and planning for the entire Data Processing Division, tool development, QA functions, and development of standards and procedures. However, in some organizations, there was a separate group for one or more of these functions, (e.g., standards development).

In the applications model, organizational subdivisions were based on applications or customer groups. For example, a group was devoted to a specific system or had a particular expertise (such as accounting or inventory control). In this model, development and maintenance were both performed within the individual groups. Standards and procedures were often project or group specific rather than division-wide.

2.1.6 User Constituencies

Various levels of user involvement and sophistication were found at each of the sites interviewed. One government site (G3) had a formal group that interfaced between the user and the developer to help prepare validation plans. The other government sites had the users involved, to some informal degree, in both requirements specification and acceptance testing.

At the commercial sites, user involvement was more formal and more comprehensive. One of the large projects at C1 supports a user steering committee that meets on a regular basis. C2 involved the users very heavily in verification and testing activities, with excellent results. This includes a high level of user participation in system tests, pilot tests, and acceptance tests. C3 and C4 were following a comprehensive commercially available methodology that involved users in reviews during every phase of development. On one project at C5, the users submitted formal decision tables and data dictionary entries and modifications for the requirements phase for enhancement activities. The use of formal decision tables was perhaps the most rigidly defined specification.

There appear to be two factors that determine the level of involvement of the user in software development or maintenance. One relates to the user and one to the software staff. There was a wide variety of sophistication across the customer and/or user groups associated with the groups surveyed. The level of knowledge or sophistication of the user determines how the user is able to participate (e.g., in what role). The sophistication of the development and maintenance group, on the other hand, determines when the user will be involved, e.g., when can the valuable user/customer resource be utilized.

The sophistication of the user/customer and development and maintenance communities defined to a large degree the interface between these groups. In general, it appeared that the sophistication (e.g., amount of formalism, rigor and/or discipline employed and knowledge of potential roles) was relatively equal on each side. An unequal example was C4 where the procedures of the commercial methodology being adopted were forcing a change in operation of the customer community.

2.1.7 Development and Maintenance Approach

All sites interviewed practiced (at least loosely) a defined phased approach for software development and/or maintenance. The approaches varied from three phases (definition, design and implementation) to as many as seven phases (initiation, requirements definition, analysis and design, development, validation, implementation, and operations audit). The degree to which these phased approaches were followed varied widely, both between and within sites. Two commercial sites were using a commercially available methodology. This methodology provided a clear definition of phases and the associated activities, roles, and responsibilities. This provided the most comprehensive set of standards found. (It should be further noted that this methodology was management oriented. In addition to defining phases and products, it stressed planning, scheduling, cost estimating, and practices that increased visibility. It did not provide specific guidance on performing the technically-oriented tasks associated with system development.)

2.1.8 Standards, Guidelines, and Procedures

All sites were in the process of refining their development and maintenance approach through the definition of supporting standards, guidelines, and procedures.

C1 had started internal projects to define procedures for improved management visibility of the software activities and for more accurate resource accounting. Also at C1, a major projects group has been formed to administer three large projects. Within this group, project standards to complement the division standards in existence were being developed. From the experience with and the refinement of these standards and guidelines, the division level standards and procedures will certainly be modified. C2 is very similar to C1 in this respect. The project group surveyed at C2 found division standards and guidelines to be inadequate and often developed their own. Some guidelines were unsuccessful and were abandoned, but many proved very helpful, particularly with regard to testing and change control. Because of the success of these standards and procedures, it was felt that they probably would be adopted (after refinement) for division level use.

In the cases of C3 and C4 (the users of the commercial methodology), efforts were underway to successfully use the methodology. C3 was developing exception criteria for using the methodology primarily based on project size. The two new development projects at site C5 followed documented corporate standards and supplemented these with project level procedures where necessary. Also at C5 a project was underway to develop documentation for system backup and recovery in case of a local emergency or disaster (e.g., building fire). The first step in this project was to define the documentation standard and formats to be employed.

Site G1 followed department and administration standards supplemented by division standards. Each of the two branches at G1 followed its own set of standards, differentiated primarily upon an emphasis on development and technical approach versus maintenance and a more management-oriented approach. Both branches were involved in efforts to enhance the existing standards.

Both Sites G2 and G3 had major projects involving the definition of new standards and guidelines (G2 as part of their conversion and redevelopment effort and G3 as a result of a reorganization). As a result of a management change at Site G4, a more formal and rigorous approach to software activities was being initiated. At G5, a subgroup was charged with the on-going responsibility of developing guidelines and standards.

2.1.9 Documentation Practices and Standards

Documentation practices varied greatly between sites and also between groups at the sites. In general, there were standards or guidelines which defined the documentation to be produced by phase. The degree to which they were followed differed greatly. The factors most often cited for non-adherence were: 1) lack of applicability, understandability, and awareness of the standards; 2) no technical guidance on the use of standards.

The documentation practices at three of the five government sites were guided by either Department of Defense (DOD) or Federal Information Processing Standards Publications (FIPS PUBS). G1 followed DOD standards. At G3, the standards group, the user interface group, and the development group were all in the process of developing specific guidelines based upon the FIPS PUBS 38 and 64. A concern was voiced over the questions of the consistency and redundancy between the two FIPS. G5 is developing new standards which appeared to be based upon (or at least fairly consistent with) FIPS. G2 is developing new standards in its conversion effort which are more closely tied to the commercial design methodology being adopted at the site than to the FIPS.

There were no generally acknowledged standards (FIPS or DOD) found in the commercial sector and none of the sites were aware of either of these standards. The documentation products and practices at all of the sites, though, were guided to some degree by standards. As previously stated C3 and C4 employ a methodology which gave detailed guidance on the documentation products. C1, C2, and C5 all followed division or corporate standards. At C1, these were closely followed by the maintenance group and used as a general framework by the larger projects. At C2, the project group surveyed appeared to be only loosely following division documentation standards where they existed. They often developed and followed project standards and procedures. The two development projects at C5 generally followed corporate standards, yet had adopted a design methodology which gave more detailed technical guidance in preparing much of the documentation.

The degree of adherence to the documentation standards appeared to depend upon several factors. First, new development projects most often had fewer constraints with which to contend than maintenance projects (e.g., lack of existing documentation, tight schedule) and in general appeared to start off "by the book." New development projects were more controlled rather than reactionary. Second, where management and/or technical leads took a major guidance role (e.g., C5) or when there was a well defined approach to be followed (e.g., C3 and C4), then intermediate products in the form of documentation were more likely to be produced. Third, the government sector appeared to be a more constrained and controlled environment, very familiar with and cognizant of existing guidelines. The use of and adherence to standards, guidelines, and procedures were more generally accepted in this sector than at the commercial sites.

2.2 VALIDATION, VERIFICATION AND TESTING PRACTICES

This section summarizes the information gathered concerning V,V&T practices, techniques, and tools, and also closely related practices such as requirements and design specification.

It was found that V,V&T was practiced, to some extent, at all the sites. However, it was not in an organized, formal, planned, or rigorous fashion, nor was it usually identified as validation, verification or V,V&T.

V,V&T was practiced primarily through reviews (e.g., formal and peer group) and inspection (e.g., desk checking) techniques. There were also many testing practices and techniques employed, some of which are supported by automated tools.

One site (G3) claimed to perform validation. At that site, a user requirements and validation group was formed to interface between the user groups and the development organization. They documented requirements, produced a validation plan (basically to drive acceptance testing), and finally reviewed the product and the results of the validation runs before accepting the product.

The following sections discuss specification, V,V&T, quality assurance, and change control practices.

2.2.1 Requirements Documentation

Requirements documentation is discussed here because of its critical role in the V,V&T process. The software requirements specifications form the basis for many of the V,V&T activities.

In general, requirements were specified and documented in some fashion at all sites. In environments where projects were small or dealt with maintaining existing software, the requirements statement was often the request for services or the statement of work submitted by the customer/user. Such statements would often go through a process (sometimes formal, sometimes informal) of clarification and elaboration, though the results of this process were not always documented. These specifications were used in feasibility investigations, cost/resource estimation, and funding/project approval. They also served as an initial statement of the problem which at least informally would drive the development process to the extent that these specifications actually became the initial system specification. However, the quality of the requirements statements varied. This depended largely upon: the customers' ability to produce and/or review a statement of software requirements, the practices, formality and rigor of the process followed by the development group, and the working relationship between the two groups. In some instances (e.g., in the case of a requirement being stated in the form of a report to be generated) the requirements specifications were used as a basis for testing and acceptance testing. These specifications, though, were seldom formally used as a basis for V,V&T.

In several of the environments with mid-sized projects (6-12 person months or longer) and/or performing new development, the requirements statements were more likely to be separate documents from the service request and were developed as a part of the project. The customer/user usually played a significant role, and quite often was more technically capable to do so than the customer/user in the small project and/or maintenance environment. The amount of emphasis placed upon a formal requirements specification, analysis and review, was a good indication of the rigor of the overall management and technical approach. And as might be anticipated, this rigor usually increased as the size, importance, and/or visibility of the project increased.

Of special note were the practices at three sites. On the large project investigated at Site C1, a user steering committee was created to formulate and review the system requirements. The requirements were formally and extensively documented and reviewed, were used to guide the design, and were planned to be used as a basis for testing. The user steering committee proved very successful and valuable and was being utilized throughout the entire project.

One maintenance group at site C5 received its requirements specified in the form of detailed decision tables and specific references to data elements defined in the system dictionary. This was by far the most rigorous specification practice encountered.

As previously mentioned, site G3 had a separate organizational group charged with formulating and documenting user requirements and producing a validation plan to guide final acceptance. At this site, the guidelines for performing these functions were in the early stages of development.

2.2.2 Design Documentation

The form and completeness of design documentation is a significant part of the V,V&T process. It determines the feasibility of:

- verifying that the design is consistent with, and has satisfied, the requirements (assuming the requirements were specified),
- performing consistency and completeness checks within the design itself,
- verifying the consistency of the code with the design, and
- providing a more thorough and complete test of the code based upon the design.

The formality and rigor of the design process varied widely across the sites. Consequently, the product of the design phase, the design specification(s), varied considerably in form and content. On small projects (and in environments dominated by these), design was usually an informal process completed by an individual analyst or small group of analysts. The documentation was usually minimal and informal. On larger projects, where the need for more complete specifications was recognized and multiple levels of design specification were performed, a more formal design process was usually followed.

Specification schemes varied between and within sites and included function trees, flow charts, data flow diagrams, program design languages and English. Data dictionaries and data specification guidelines were used also. Sites (or groups at the site) C1, C5, G1 and G2 had adopted a commercial design methodology and were sponsoring internal training. Sites C3 and C4 had both adopted a commercial project methodology which addresses the details of design to some extent. The adoption of formal approaches to design will certainly impact the quality of design documents in the future at these sites.

Several sites (C2, G1, G2, and G5) had developed or were in the process of preparing guidelines for design documentation and/or specification. These varied widely, from specification of the format and content of what would ultimately become program documentation, to guidance on preparing and reviewing module design packages including flow charts, psuedo code, and data definitions.

Few tools were employed in design documentation. However, C5 and G1 were experimenting with automated tools to assist in design documentation.

2.2.3 Verification Between Phases/Within a Phase

The phased approach to software development and top-down development techniques is based upon the evolution of a specification through the elaboration of one level of specification to form another. A fundamental concept of V,V&T is to check the consistency between each two successive levels of detail. The extent to which this can be accomplished depends upon the information contained at each level of specification. The design specification can only be verified against unambiguous, complete requirements specifications. Code verification requires unambiguous, complete design specifications.

There was little evidence that this concept was explicitly and rigorously implemented in practice at the sites surveyed. One project at C5 traced requirements and high level functions throughout the system specification and checked for certain types of consistency between levels of specifications. The project investigated at C2 employed reviews to ensure consistency between levels of design. This was also done at site G1. Reviews (both formal and informal) were held at most of the sites. Though the analysis of and review for consistency between levels of specification were not stated explicitly as results of these reviews, it can be assumed that this was being accomplished to a degree.

Three factors seem to contribute to the lack of verification being practiced:

- o absence of formally documented intermediate levels of specification,
- o the lack of explicit guidance on what to include in

documentation, and

- the lack of explicit guidance on the types of checks to perform both during and between phases.

The adoption of formal methodologies at several of the sites is counteracting some of these factors.

2.2.4 Software Testing

Testing practices employed at the sites differ in terms of:

- the amount of testing performed,
- the formality and rigor of the testing programs,
- the amount of planning that was performed, and
- the specific tools and techniques employed.

Testing in the maintenance environments and on small projects was generally an informal process consisting of module testing and debugging and then some sort of complete system testing. On larger projects and particularly those involving new development, testing was treated more formally. Groups at sites prepared test planning documents. At several sites, documentation of test analysis and results was prepared.

The predominate strategy used for system testing was the use of large files of production data. The assumption was usually made that this would produce the most complete test set, especially when considering special conditions, error conditions, and incorrect data. A closely related technique which was used at several sites was the construction of a test database (particularly for testing changes to an existing system) that could be used for nearly all of the testing activities. This was usually created from production data. The amount of care taken to selectively choose records to ensure broad test coverage varied. Often it was assumed that quantity would assure a variety of test cases.

Site C2 had the most extensive test program encountered. The program was primarily the responsibility of the independent system test team. Products included test plans and test analysis reports. Four levels of testing were performed:

- unit testing was done solely by the developers,
- integration testing was done by the developers, but monitored by the system test team,
- functional validation testing was done by the system test team alone, and
- acceptance testing ("dress rehearsal") was conducted by the system team with the involvement of real users.

Site C2 spent considerable effort in the latter two levels of testing. During functional validation testing, a complex test database was created. Also during this phase, the system test team performed an on-site test in parallel with the operation of the production system (the same data that were used to generate bills were used as test data, with the test results being carefully compared with the "real" results). Acceptance testing consisted of two major activities, dress rehearsal, and a pilot operation. During dress rehearsal, a test office was set up. Real users and operation personnel worked with the system test team, utilizing real data. The pilot operation consisted of putting the new system into full operation in a live, but limited situation. The active inclusion of the users in the final phase of testing was a major investment in terms of corporate resources and was responsible for the acceptance of the new system and the current goodwill between users and system developers.

There were several sites where testing tools were used. Libraries were often used to store and manage test data. Comparators were used at a couple of sites for comparing test results with production results. Both G1 and G5 used a commercial test data generation and management tool. G3 had built a record extraction program to assist in preparing test files. C2 had developed several test aids including a JCL generator tool for test runs.

2.2.5 Acceptance Testing

Acceptance testing practices varied widely at the sites interviewed. The user was always involved, regardless of the degree of formality. The formality of acceptance testing was dependent primarily on one of two factors:

- o the size of the project, and
- o the use of the end product (e.g., if the software dealt directly with the generation of corporate funds or if it was to be used for the generation of a special, one-time-only report).

When fixes or enhancements were made to production software, very little formal acceptance testing was done. The reason most cited for not doing formal acceptance testing was lack of time. (The second most cited reason was lack of resources, particularly personnel trained to prepare comprehensive plans.)

Several commercial sites had fairly comprehensive acceptance testing practices. Site C2 set up an operational pilot site to use for acceptance testing, with full involvement of all users. Management evaluated the results of this activity and decided that it was fully cost effective, both in terms of technical acceptability of the end software product and in preserving and enhancing an enlightened user-developer interface.

Sites C3 and C4 have been following the procedures set forth in the commercial methodolgy. These procedures specify formal user involvement and have been evaluated and found cost effective by management.

2.2.6 V,V&T Techniques and Tools

The following V,V&T techniques were observed to be in some degree of use at one or more sites interviewed:

Commercial development and design
Independent test team
Structured programming
Coding standards
Requirements reviews
Design reviews
Test reviews
Peer reviews
User steering committee
Documentation standards
Naming conventions
Design walkthroughs
Code walkthroughs
Desk checking of code

The following tools were utilized in support of V,V&T activities:

COBOL preprocessor
Compilers
Cross-reference tool
Data documentation tool
Interactive debugger
Dynamic analysis tools
Test data generation tool
Test bed support facility
Data dictionaries
File comparators
Project manager support tools
Source code management system

The use of both the techniques mentioned and the tools was very informal. Tool usage was particularly ad hoc. At one site, use of various available compiler options was classified as V,V&T. Dynamic analysis tools were used exclusively for system performance analysis, not V,V&T. The COBOL preprocessor, the cross-reference tool, and the data documentation tool were used to help produce more readable, better documented code, that could be more readily checked against design and requirement specifications. In this sense these tools aided the V,V&T process. No tools for comparing code with standards were identified at any of the sites interviewed. Sites C2 and C3, users of the commercial development methodology, were actively pursuing the acquisition of tools to support that methodology.

2.2.7 Quality Assurance

Technically, quality assurance (QA) activities are predicated on unambiguous standards. During the survey, it was found that:

- a number of sites had no standards and, hence, no QA activities and
- some sites performed a hybrid activity that included some QA, some V,V&T, and some configuration management.

At several sites, it was a first level management responsibility to enforce the use of guidelines, standards, and procedures. In many cases, this responsibility was subordinate to the need to get the software completed within time and cost constraints. Only site C3 had formal QA activities, centered in a small independent group with the charter to certify all production JCL (Job Control Language) and any modifications made. C3 was in the process of expanding that charter to include production software in general.

2.2.8 Software Change Control/Configuration Management

With the exceptions of sites C2 and C3, no formal configuration management (CM) was identified. The responsibility was generally assumed to be that of either the project manager or the analyst/user most involved with the software. In some cases, a member of the computer operations staff was responsible.

Site C2 had a formal configuration management activity for the specific project interviewed. Since the software in question generated corporate funds, was utilized by a geographically dispersed set of users, and was still actively being "enhanced", management identified a well defined configuration management plan as critical.

Site C3 also had formal configuration management. Here, however, CM activities for all production runs (not just one major project) were consolidated in a single organizational entity. At this site, QA and CM interfaces were well defined.

2.3 DOCUMENTED STANDARDS, GUIDELINES, AND PROCEDURES

2.3.1 Current Use of Standards, Guidelines, and Procedures

Each of the ten sites interviewed exhibited different levels of formality in specifying standards, guidelines, and procedures. At one end of the spectrum, Site G4 had no official standards; everything was left to the discretion of the individual analyst responsible for the software. At the other end of the spectrum, Sites C2, C3 and G3 had identified comprehensive commercial methodologies as standards to be followed by all projects.

One government site interviewed (G2) utilized DOD documentation standards. Government site G5 had customized FIPS PUB 38 to address its particular environment and had also devised a set of coding standards to be followed. Commercial site C1 had several organizational levels of standards and procedures (see Appendix F, page F-12). Commercial site C2 had a comprehensive set of standards for the particular project interviewed and was in the process of developing corporate-wide guidelines, standards, and procedures (based in part on the experiences of the subject project).

2.3.2 Adherence and Enforcement

Perhaps the one thing that all of the sites interviewed had in common was the problem of adherence to and enforcement of any standards regardless of how comprehensive those standards might or might not be. In some cases, it was assumed that the individual analyst in charge would be responsible for adherence to standards. Not surprisingly, this approach was far from successful. The QA, or independent test team was given the responsibility at several of the commercial sites; the success rate here was a function of the formality of the standards and procedures being enforced. The reasons most cited for not enforcing standards were:

- lack of comprehensive and unambiguous standards to be followed and
- lack of time and resources to do the job.

All sites expressed a desire for automated aids for enforcement, but also expressed a concern that those aids be easily modified to fit their specific environment.

2.3.3 Attitudes Towards Formal Standards

Once again, the entire spectrum of possible attitudes towards formal standards was found. One site basically said that their environment was too unique to have anything but case-by-case standards. Most of the sites interviewed said that they had not seen anything really satisfactory yet, but would look at anything that did not involve "excessive" red tape. Those sites were particularly interested if there was minimal resource commitment necessary to apply and enforce any formal standards proposed.

Another concern was that the standards be flexible, i.e., that there was a way of tailoring the standards to projects based on size, deadlines, available resources, and the significance of the final project.

Most sites felt that government guidelines had proved to be too general in the past and were concerned with the possibility of being mandated to follow the guidelines without sufficient supporting data.

3.0 SUMMARY

One of the early steps in the development of the survey questionnaire was the preparation of a statement of several assumptions underlying the study. This was a set of hypotheses to be informally tested. This section will present each of these assumptions and a discussion of the relative findings of the survey.

3.1 RELATING TO STANDARDS & GUIDELINES

An assumption underlying the entire project is that there is a need for a V,V&T guideline to assist in technology transfer and the selection and application of V,V&T tools and techniques. It was further assumed that there were guidelines in existence (both formal and informal) and that studying these and their application would help characterize the current state and use of V,V&T technology in the types of environments studied.

The findings of the survey totally support the first assumption concerning a need for a guideline. However, the second assumption regarding existing V,V&T guidelines proved incorrect for the particular environments surveyed. There was a general lack of awareness of V,V&T concepts and principles and only sparse and informal application of V,V&T practices. There were no guidelines or standards found which directly addressed V,V&T, though there were standards regarding various types of reviews and related activities. All of the participants expressed the need for assistance in this area.

With respect to guidelines and standards other than for V,V&T, most environments had standards which to some degree covered one or more of the following areas:

- project management,
- software development/maintenance activities,
- project phases,
- software and project documentation,
- coding conventions, and
- general operating procedures.

None of the environments had a separate enforcement function. Compliance varied widely, but in the majority of environments the standards were only loosely followed. Few environments provided technical assistance or training in the application of their standards. Professed attitudes toward the importance of standards, and compliance with them varied for both management and staff.

3.2 RELATING TO V,V&T PRACTICES, TECHNIQUES AND TOOLS

It was assumed that V,V&T techniques and tools have been and are currently being used, that the experience with techniques and tools is varied, and that information concerning purpose, description, use and utility of these tools and techniques could be collected and assembled in support of the guideline development.

As summarized in Section 2.2, V,V&T was practiced informally through the use of predominantely manual methods, e.g., reviews and inspections. Experiences with the techniques employed did vary in terms of effectiveness as judged by those interviewed. When the techniques used were judged unfavorably, it was usually due to a lack of time, guidance, and support (from management, customer, etc.) for a proper implementation. No new V,V&T techniques or tools were discovered, though there is insight to be gained from some of the implementations and applications observed (Section 2 and site reports in appendices). There had been no studies on the cost effectiveness of V,V&T practices performed at any of the sites.

In general, regarding the application of software development and maintenance technology and tools, all of the sites and groups within sites were progressing toward more disciplined, well defined, and better managed approaches to software related activities. This included the adoption of methodologies and/or the use of individual techniques and tools. These activities were generally hampered by a lack of knowledge and awareness on the part of management and staff, the momentum of past practices, and the unproven (lack of quantifiable data) benefits of the new technology.

3.3 RELATING TO ENVIRONMENTAL FACTORS

One of the major objectives of the survey was to collect data to support the study of factors affecting the application of V,V&T technology in various environments. It was assumed that there were major factors that can be identified, which are the basis for these environmental differences, including:

- application areas,
- sizes of projects,

- management philosophy, organization, technological sophistication,
- software development methods, techniques, and tools, and
- computer resource and support environment,

and that these and other factors have a major effect on the operation of the environment and the resulting software products.

There are several factors that stand out as having major effects on software development and maintenance activities. Perhaps the largest difference noted across all the groups was the degree to which the software development and maintenance process was approached as a well defined activity. It varied from the process being defined by the analyst performing the task to well defined structured management of projects. Perhaps the two most apparent factors included:

- the awareness of, commitment to, and level of expertise in software engineering principles and techniques on the part of management and staff and
- the existence of guidelines and mechanisms supporting a more formal approach.

The limited use of specific techniques and tools did not appear to be a leading factor, but rather a result of the above two factors. In selected instances, techniques and tools acted as catalysts in a progression toward a more formal approach.

3.4 RELATING TO DIFFERENCES BETWEEN THE FEDERAL AND THE COMMERCIAL SECTORS

The objectives behind choosing five commercial and five Federal sites were to investigate differences in these two environments. Three differences observed are mentioned below.

First, the computing software activities in the commercial sites were more centralized and visible than in the government sites, due mainly to the size and complexity of the Federal government. Heads of data processing divisions in the commercial sites interviewed reported high up on the organization chart. On the other hand, the software groups in the Federal sector were far down in their respective organization charts, and within any one department or administration, there were multiple such groups. This lack of centralization and visibility has obvious effects on the definition of charters, customer sites, and competition for funds.

The second observation is that, in general, the Federal sector appeared to be more tightly constrained. All government sites were directly affected by wide regulations relating to personnel, hardware/software acquisition, and contracting. All of the government groups were familiar with one or more sets of documentation standards (FIPS and DOD), while staff at most of the five commercial sites were not aware of the existence of such standards. This appeared to be a measure of the regulation and standardization felt by the ADP groups within the government.

A third observation, due in part to the previously stated differences, is that the commercial sites appeared more receptive to new software engineering technology as compared with the government sites. The groups surveyed at the commercial sites appeared more willing to experiment with new technology, more able to do so because of management support, and were consequently more directly impacted by new technology. Although the technology varied widely from site to site, there was a noticable difference in the progressiveness observed between the two sectors.

APPENDIX A

GOVERNMENT SITE 1 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site G1 is a Federal administrative agency. The Data Systems Division provides a major portion of the software development and maintenance services for the entire agency and some services to customers outside the agency. (Operations is the charter of another division and will not be discussed in this report.) The Data Systems Division is divided into the Systems Branch and the Administrative Branch (Figure A-1). The Systems Branch supports special studies and engineering applications (primarily research and design) for internal and external customers. The Administrative Branch supports the agency directly and is divided into two sections. Finance is responsible for payroll, accounts, budgets, and analysis of fiscal programs. Personnel is responsible for personnel files, administration information, and special projects.

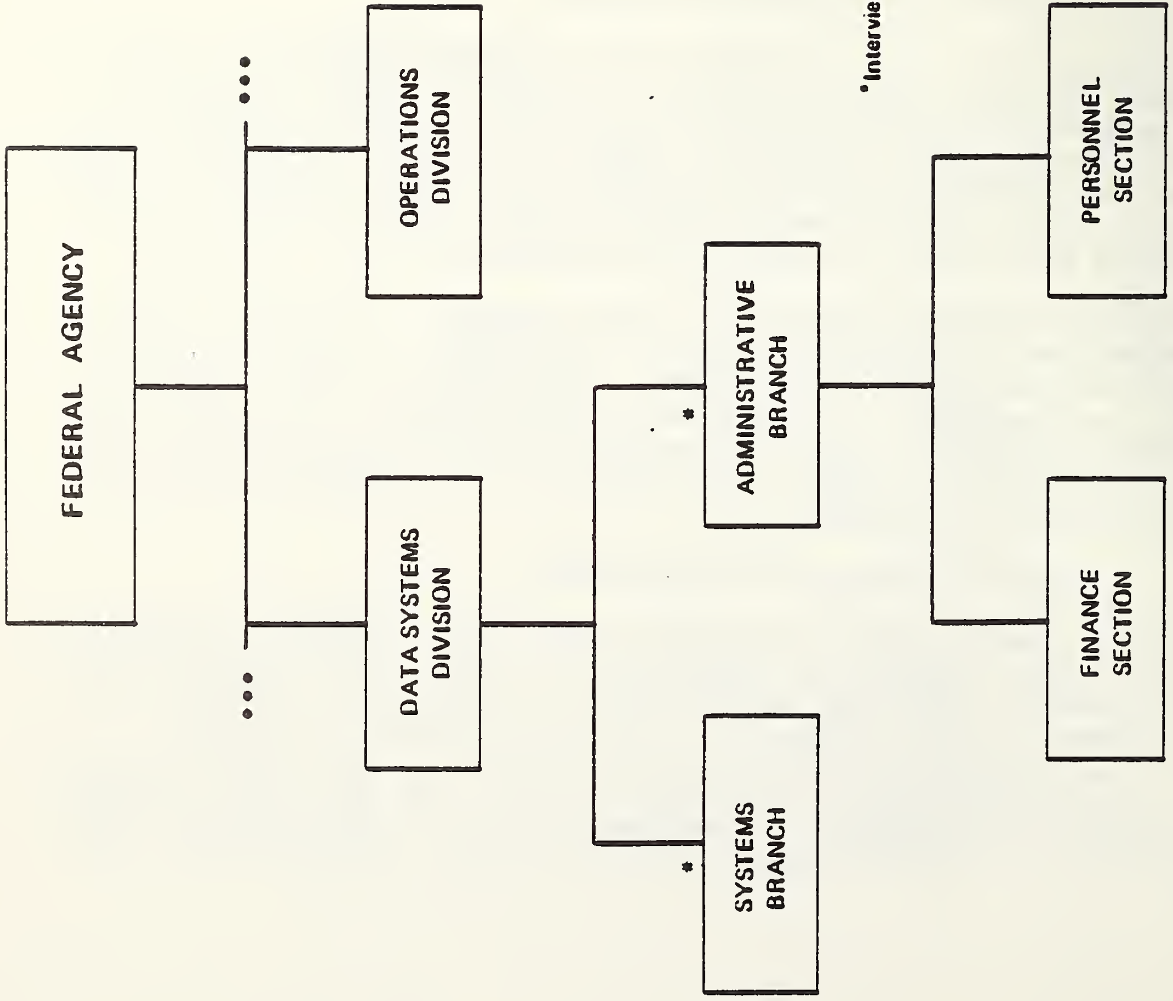
1.2 Description of Software Activities

The Systems Branch has a single manager and 20 programmer/analysts. Most of the projects provide information for new product design, enhancements to existing products, or statistical profiles related to product utilization. Almost all programs are written in COBOL for batch processing on large mainframes. Job sizes are about equally distributed between small (1-2 months of effort), medium (3-15 months) and large (more than 15 months). It was estimated that about 2/3 of the work is new development and 1/3 maintenance.

The Administrative Branch consists of a single manager, 2 section heads, 6 team leaders and 17 programmer/analysts. All programs are written in COBOL and run in batch mode on a large mainframe. Approximately fifty percent of the work is new development (including major enhancements to existing systems) and the rest, maintenance. Ninety percent of the projects are small (3-6 months), nine percent are medium (6-18 months), and the remaining one percent are large (more than 18 months).

1.3 Factors Influencing the Environment

In the Systems Branch, budget and schedules were not important constraints (except in cases required by legislation). Most data produced are not critical. Accuracy in the engineering design applications is important, but the environment is fairly tolerant with respect to absolute correctness. The product of the Systems Branch is primarily analytical and statistical in nature. Many products are one-time deliverables or are periodically updated (e.g., annually), as contrasted to production jobs of the Administrative Branch. Fast turn around and security are not critical considerations in this branch.



*Interviewed for Survey

Figure A-1. Site G1 Organizational Structure.

SITE SUMMARY
GI

General Characteristics		
Group Name	:	Data Systems Division - Systems Branch & Administrative Branch
Type of Activity	:	Computer support for agency and other Federal groups
Application Area	:	Engineering, Administrative, & Management Support
Languages	:	COBOL
Project Size Range	:	Small: 1-2 months; Large: 15 months months or more (manmonths)
Development/Maintenance Technology		
Development, Maintenance or Other	:	Development & Maintenance
Documented or Informal Approach	:	Informal (adopting commercial methodology & moving to more formal approach)
Number of Phases	:	4 development phases: Definition, Design, Coding & Testing
Use of Formal Techniques/Tools	:	Some specification & documentation guidelines (commercial methodology will introduce new techniques)
V&V Technology		
Requirements as a Baseline	:	Informally specified
Other Levels of Specification	:	Informal design
V&V Within/Between Early Phases	:	Little
Testing Practices	:	Informal
Reviews	:	Usually Informal
Techniques/Tools	:	Compiler cross reference, debug options, and test data generation tool
Documentation Practices		
Formal/Informal Guidelines	:	Formal (DoD)
Adherence to Guidelines	:	Moderate, varies by project size and importance
Enforcement/Review	:	No Formal mechanism

The Administrative Branch customer is more familiar with data processing than the Systems Branch engineering customer. As a result, the requests for services are more formal and more specific. Pay checks, accounts receivable, resource utilization, and personnel records must be accurate. Information sent to management is often critical for making decisions. Management of this branch is very interested in incorporating standards that will result in accurate, error-free software.

There is no major impetus to change in either Branch. Management and customers are comfortable with current mode of operation. In fact, there would be resistance to change that might result in a more formal interface or a more rigorous development approach.

1.4 Historical Perspective/Evolution

Site G1 illustrates a contrast between the Systems Branch, concerned with engineering-oriented R&D, and the Administrative Branch, concerned with the development and maintenance of business-oriented production software. A difference in the needs and emphasis of the two branches resulted in the failure of an attempt to standardize division operations. Each branch has adopted informal procedures that meet its unique needs. Systems Branch management appears to be encouraging the use of better design and coding practices. Administrative Branch management appears to be placing greater emphasis on the overall process, and the procedural steps to support that process.

2.0 DESCRIPTION OF SOFTWARE ACTIVITIES

2.1 Overview of Development Approach

A set of policies and guidelines was developed under the direction of division management to help standarize operating procedures. However, because of the different nature of the two branches, a single implementation could not be agreed on. The project activities of both branches are divided into three basic phases (as in FIPS 38): Initiation, Development and Operation (Maintenance). The differences arise in the activities making up each phase, and the distinction between stages (definition, design, coding and testing) in the development phase.

In general, the Administrative Branch employs a more formal development process than does the Systems Branch. The request for services and the final product acceptance are always via written communication between the customer and developer. On the larger projects, other formal reviews may also be held.

The Systems Branch employs less formality in its customer interface. Quite often, agreements are verbal only. Though less formal, communication seems adequate. The projects in this branch are usually larger than those in the other branch, and involve more new development.

2.2 Phase Descriptions

2.2.1 Initiation Phase

A request for services is initiated by a customer. The analyst and customer work together to complete the request to the level of detail necessary to act as a requirement specification for the analyst. When completed, the request is formally approved and signed off in the Administrative Branch. The Systems Branch does not require a written statement of approval.

2.2.2 Development Phase

The development phase is loosely organized in four stages: definition, design, coding and testing. The Systems Branch acknowldges more defined distinction between the stages than the Administrative Branch.

2.2.2.1 Definition Activities

For most projects there is not a clear distinction between the initiatic phase and the definition stage. A project is initiated with a request for services. This is clarified and elaborated upon until design can begin.

2.2.2.2 Design Activities

The input to this stage is the request for services prepared by the initiation/definition activity. Design specifications will be derived from this document. For the larger development efforts, the request for services is sometimes broken down into subtasks; the result of the design process is a system specification and a more detailed program specification. The Systems Branch has guidelines for the program specification, which results in internal and external program documentation. Guidelines for the system specifications are planned.

A commercial design methodology is being adopted for use on most projects. It is being advocated by management, and staff training is being supported. Informal system-level design reviews are held for most medium and large projects. To support adherence to the program specification standards, a facility and produres have been developed for using a source-code library tool to assist in producing external program documentation from internal program comments.

In the Administrative Branch, the design stage is informal. The formality of the documentation produced varies depending upon the project and the practices of the analyst. It appeared that there usually is not a distinct transition between the design and coding phases for these types of projects. For larger projects, there is a transition point, and design reviews are held. In some instances, there are two design reviews - a preliminary design review and a critical or detailed design review (the system level and program level, respectively).

2.2.2.3 Coding Activities

The coding stage of development is not always clearly distinguished from the design and test stages. All three stages usually involve the same personnel. The input to the activity is whatever design documentation has been produced, and the outputs are the debugged code (ready for testing) and user/operating instructions.

2.2.2.4 Testing Activities

Some form of documented test planning is usually done in both branches. Testing usually involved live data. Testing of enhancements was often done in parallel with normal production runs, or a subset of the production data would be used for testing. In the Systems Branch, a test-data generation tool was sometimes employed for testing new software where data were not available or deemed not completely satisfactory for testing. In this branch, a test-analysis report is usually produced to facilitate review of the testing activities. This is usually a significant step in the customer's final acceptance of the product.

Both branches go through a final acceptance procedure with the customer. In the Administrative Branch, a formal sign-off is required.

2.3 Operation Phase

The actual implementation takes place after the customer's final acceptance. Implementation and operation (production runs) are usually the responsibility of the computer center staff and/or the user.

2.4 Documentation Practices

DOD documentation standards provide guidance in the documentation practices. In general the documentation practices varied by project site and importance. Closer adherence to the standards was enforced for the larger and more important projects.

In both branches, the following documentation is usually produced for significant applications: functional requirements description, database specification, and operations and maintenance documentation. In addition, for the larger projects, documentation describing the following information is produced: project plans, data requirements, database specification, system/program specification, users instructions, test plans, and test-analysis reports.

2.5 Quality Assurance Activities

Site G1 has no formal quality assurance program. Informal practices are employed at the discretion of the developers. The driving force for the creation of a quality assurance program is usually a dissatisfied customer, which was not observed in either branch.

2.6 Validation, Verification, and Testing Activities

There is no formal V,V&T program currently in this division. The V,V&T practices employed are usually at the discretion of the project leader. Some reviews are mandatory. Requirements and test reviews were held and design and code walkthroughs were practiced.

2.7 Configuration Management Practices

The software configuration management or change control practices at this site were not extensively investigated. The responsibility was usually assumed by project management and accomplished though the actions of the analyst and computer center staff. The practices appeared to be more formal in the Administrative Branch because of working with production software. A request for service acts as the basic change request. The creation and maintenance of source code libraries were the predominant mechanism for maintaining the code.

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

Reviews and walkthroughs were the primary V,V&T technique used. Reviews were either informal or formal, depending on the project. Review of the request for services (which acted as the statement of requirements) was usually held between customer and developer. Reviews prior to final acceptance were also usually held. Other review points included preliminary design and detailed design, depending upon the size, complexity, and importance (or visibility) of the project and/or the discretion of project management. Informal code walkthroughs were reported to be a fairly common practice. Design walkthroughs were also cited as a practice less frequently employed.

There were several other techniques or practices which were used that should assist in system development, specification, and documentation. A commercial design methodology was being adopted for use, with staff training being provided. Structured programming was also being advocated and included in an internal training program. System and program-level flow charts, prose descriptions, and standardized file description formats were being used as aids in system/program specification. Adherence to documentation (DOD and internal) and COBOL coding standards and naming conventions is recommended. For testing, the primary stategy is the use of live data and, where possible, testing in parallel with a production program.

Top down and structured coding techniques are advocated and taught internally. COBOL coding standards and naming conventions are employed. Code walkthroughs are also practiced.

3.2 Tools

A variety of automated tools are used at this site. There are three tools that are commonly used in code debugging, test and verification. Certain compiler options are used to assist in debugging. Another tool is used to produce an extensive variable cross-reference map. This is used not only as a debugging aid but also for documentation and maintenance activities. A test tool which includes test-data-generation capabilities is often used. Another tool mentioned was a COBOL macro-language preprocessor, which was primarily used for detection of non-standard COBOL, formatting of source code, and analysis of performance. There are also two other tools utilized for performance analysis and optimization. To assist in program documentation, an automated flow chart and the cross-reference tool (previously mentioned) are used. A source-code library management tool is used to assist in source-code management, change control, and program documentation.

3.3 Perceived Problem Areas

There were three subject areas mentioned as areas for possible improvement: testing techniques, design methods, and requirement/design specification and change control.

4.0 STANDARDS, GUIDELINES & PROCEDURES

There exist three documented standards which give guidance on: 1) determining needs for automated systems, 2) coordination and approval of ADP projects, and 3) management of ADP projects. Two of these are department level and one is an administration (within the department) level. These all three apply to and are followed (to a degree) by the branches of the Data Processing Division. To supplement these, guidelines have been developed or adopted by the branches, e.g., the DOD documentation standards and the systems branch program specification guideline.

Few of these standards or guidelines are rigidly followed and there is not an enforcement mechanism. However, in certain instances they do appear to provide a framework and guidance for the development process and associated activities.

APPENDIX B

GOVERNMENT SITE 2 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site G2 is a large Federal agency whose software activities support the services provided by the agency. Software activities are centralized within a Data Processing Support Division and support geographically dispersed customers. There are a small number of "field offices" in addition to a headquarters facility.

The Data Processing Division has recently undergone a reorganization. The structure has changed from a machine-oriented division (i.e., staff groups supporting work on a given type of hardware) to a functionally-oriented division (figure B-1). The new organization structure is set up to support each of the major functions of the agency. Formerly, the roles of analyst and programmer were separated. In the new organization, there are analyst/programmers.

1.2 Description of Software Activities

The software activities support all internal processing requirements and also requests by external "customers" in regard to a particular inquiry or need.

The software activities of this agency are data oriented. As such, several master files of data are collected and maintained by the agency. The agency relies upon these files to provide input to support its other functions. Two of these files are very large and provide data to support a large part (over 50%) of agency activities. Security concerns play a critical part in the software activities.

In order to support a particular service provided by the agency, a software project may be initiated. A development team, under the direction of a first-line supervisor, performs all of the development activities up to and including unit test. Integration test is performed by an independent test group.

There are currently about 1200 applications programs supporting Site G2's activities, 98% of which are written in Assembly Language. No database management system is utilized. Some systems software (i.e., utilities) is in use, but this is limited due to the use of assembly language. All application software has been developed in-house. Currently, software effort is divided at about 90% maintenance and 10% new development (in assembly language).

1.3 Description of Hardware Configuration

There are presently three types of large mainframe machines supporting software activities. At the headquarters facility, two types of mainframes support testing activities. The large master files are maintained on the

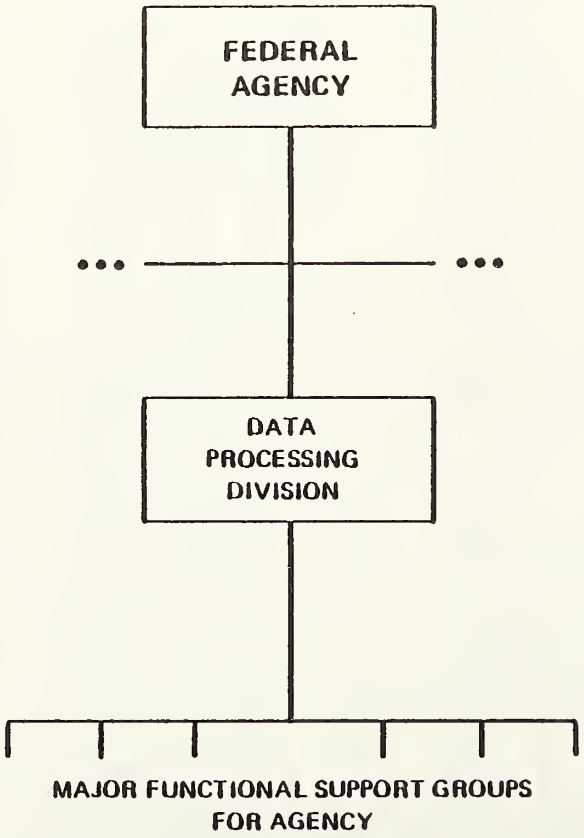


Figure B-1. Site G2 Organizational Structure.

SITE SUMMARY
G2

General Characteristics		
Group Name	:	Data processing division
Type af Activity	:	Mostly data ariented software suppart
Application Area	:	Program administration
Languages	:	Assembly language
Praject Size Range	:	Small: 10 months ar less; Large: 36 months or mare
Development/Maintenance Technology		
Development, Maintenance or Other	:	90% maintenance; 10% new development
Documented ar Informal Appraach	:	Evalving from informal approach to cammercial methodology
Number af Phases	:	Currently 4 : Requirements analysis, Coding, Independent testing, Operatian
Use of Formal Techniques/Tools	:	None
V&V Technology		
Requirements as a Baseline	:	Requirements Analysis Package (RAP) developed in parallel with development phase
Other Levels of Specification	:	Design
V&V Within/Between Early Phases	:	None - initially all done after development
Testing Practices	:	Unit testing done by developers, then system test by an independent test team
Reviews	:	Few formal reviews
Techniques/Tools	:	Few tools, no guidelines far use
Documentation Practices		
Farmal/Informal Guidelines	:	Informal guidelines
Adherence ta Guidelines	:	Not adhered ta
Enforcement/Review	:	Reviews not required; rarely held

third mainframe. Remote job entry to this latter system supports production as well as testing. Each of the 10 dispersed facilities operates two large mainframes and also has access to the system supporting the master files via Remote Job Entry (RJE) stations.

Several of the individuals interviewed felt that the current equipment was saturated with processing. There are constraints due to limited storage and processing capabilities. In addition, the features of two of the mainframes are limited due to the age of the machines. Within the next few years, a competitive procurement will be issued for new hardware.

1.4 Factors Influencing the Environment

Several factors play a critical role in the software development and maintenance environment at the agency.

The site's software activities revolve around five data files. Headquarters and each of the distributed facilities utilize and rely on those files on a daily basis.

New or revised Federal legislation causes continually changing requirements for the type of services/support provided by the agency. Furthermore, this legislation happens on a yearly basis, creating cyclical forces in the programming environment. Often, legislation is passed late in the cycle and results in a software requirement that must be satisfied prior to the end of that cycle. This has implications for timely response.

The type of data which is processed creates associated needs for validation, accuracy, and security.

1.5 Historical Perspective/Evolution

As with many commercial and government organizations, this agency became computerized due to increased processing requirements, unmanageable volumes of data, and limited resources. Initially, few of the technical or management personnel had any background in data processing. Therefore, little in-house knowledge about new software development techniques or tools was available. Software was not developed using a phased approach or by employing certain disciplines (i.e., structured design, requirements analysis). In the past few years, software development expertise has been brought in-house through education or by new personnel. As a result, the agency is in the process of developing new standards, adopting existing methodologies, and employing a higher-order language.

Another factor in this evolution was audits by the General Accounting Office (GAO) and also an internal auditing group. Both of these audits expressed concern for an improved development/maintenance environment.

The transition to a disciplined environment is being supported by high-level management and is being initiated through training courses (see subsequent sections for details).

2.0 DESCRIPTION OF CURRENT SOFTWARE ACTIVITIES

2.1 Overview of Development Approaches

The site currently follows standards which include 4 phases: requirements analysis, coding and unit testing, independent testing, and operation. The site has recently initiated an effort to develop a new set of software lifecycle standards and guidelines. Preliminary drafts have been available for approximately 4 months. The new standard is based on a commercially-available methodology and will include 6 phases: problem definition, requirements definition, analysis, design, programming, and system operation. Both lifecycle approaches are discussed below.

2.2 Phase Descriptions

2.2.1 Current Phased Approach

Due to the draft status of the new software guidelines, the six-phased approach outlined above is not yet followed throughout the site. The lifecycle currently employed is less formal than the proposed guidelines. It defines four phases: requirements analysis, coding and unit testing, independent testing, and operation.

The requirements analysis phase is initiated by a request for data services. The response to this request is the preparation of a Requirements Analysis Package (RAP) describing the input and output needs of the proposed system. The package also describes system processing at a very high level. From this, it is possible to put together an estimate of system costs and benefits which is submitted to management for approval. If the system is cost-justified as a solution to the original user requirement, the coding phase begins.

The RAP serves as a preliminary design for the system. Coding is done (mostly in assembly language) and modules are unit tested as they are developed. Some informal integration testing is also done by the developers at the completion of coding.

The system then under goes acceptance testing by a group independent of the developers, in order to ensure that the original requirements have been met. Upon approval by the independent test team, the system is installed and placed under operation.

Documentation produced under this lifecycle concept is informal and does not assist requirements traceability. Reviews are not required and are rarely held.

2.2.2 Planned Development Approach

There is currently an effort underway to formalize the development approach and create associated standards. This involves the acquisition and refinement of a commercially available methodology. In addition, a large conversion activity is in the planning stage. Approximately 1200 assembly language programs are scheduled to be converted to COBOL. The new approach and associated standards will be employed during the conversion activity as well as development activities. It is anticipated that this new approach will be more formal than the old.

The new approach will emphasize 'structured' concepts, such as top down design, modularity, and heirarchical decomposition, and incorporate techniques and design representation schemes. The commercial methodology includes complete guidelines for documentation produced during the entire lifecycle. Site G2 is currently drawing up standards specifying documentation required for completion of each individual phase.

The problem definition phase will begin when the user submits a request for data services. This request will be reviewed by the applicable ADP management for review and approval.

The requirements definition phase will begin upon approval of the request for user services. At this time, the user and the analyst will prepare a statement of system inputs, outputs and high-level processing requirements. This summary will undergo review by ADP management and, upon approval, will be sent back to the technical staff for detailed analysis.

During the analysis phase, a functional specification document will be prepared. To prepare this document, data flow and data decomposition will be studied. The requirements will be specified in a pseudo-code format, using sequence, selection, and repetition structures. The phase will be subdivided into 'logical' and 'physical' stages, as defined by the commercial methodology. One or more walkthroughs may be conducted during the phase. A walkthrough will be held at the completion of the phase for final approval.

The design phase will result in the translation of the functional specification into detailed design specifications. Specialized charts will be used to represent the design. Design evaluation and subsequent refinement will be supported by these charts, the conventions they imply, and by formal analysis techniques (cohesion, coupling and heuristics as defined in the commercial methodology). One or more walkthroughs may be conducted. A review at the completion of the phase will serve to approve the design.

During the programming phase, top-down methodologies and incremental (unit) testing will be employed. The COBOL language has been required for use by Site G2 management, and standards to facilitate coding using structured concepts have been developed. Walkthroughs will be held during the phase. At the completion of this phase, an independent test will be conducted to validate the system requirements. This independent test is more thoroughly described later in this appendix.

The certified programs will be installed during the System Operation phase. A post-operation review will sometimes be conducted based upon time available and the criticality of the program. The user, development group, and independent review team are planned to participate in the review.

2.3 Quality Assurance Activities

As with the developmental approach, the current mechanisms which support quality assurance will undergo change as the standards are finalized.

2.3.1 Current QA Mechanisms

There are currently some handbooks and directives that provide guidance to analysts and programmers. A programmer's handbook is adapted and documented for each of the hardware systems. It contains sample execution language, macros, and descriptions of available utilities. Guidelines also exist for executing PL/1 and librarian-type (e.g., file backup) facilities. These are system-oriented guidelines.

The general guidelines defined in the handbooks are usually followed; however, lack of specifics to support those guidelines has been a problem. The only method to enforce adherence to current guidelines is through first-level supervision, and this, also, is recognized as a problem.

The independent test group performs some activities generally classified as QA (i.e., check compliance with user requirements, review documentation for accuracy). These activities, however, are subject to time and schedule constraints and are hampered by the lack of data processing experience among team members.

2.3.2 Planned QA Mechanisms

There appear to be two additional quality assurance mechanisms planned for the agency. The first centers around the disciplined environment and the use of development standards to define that environment. Additionally, techniques described in Section 3.1 will encourage development of a quality product.

The planned disciplined environment incorporates reviews and walkthroughs designed to permit visibility into the emerging product and to encourage a greater degree of user involvement throughout the lifecycle.

Another planned mechanism is the use of an internal auditing group to participate in selected development activities. This group is being staffed from within the data processing division. Individuals with accounting and data processing backgrounds are being selected. They will attend each of the structured courses being offered at the site in order to be knowledgeable in the new approach. They will be familiar with formal standards, thus capable of reviewing software documentation for concurrence with the standards as well as completeness.

2.4 Validation, Verification and Testing Activities

Current V,V&T activities involve some formal reviews, informal low-level software develoment guidelines, and testing by an independent test team. The primary formal review is the Requirements Analysis Package (RAP) coordination meeting. Project plans require the formal approval of the ADP board. A programmer's handbook provides guidelines for using system utilities and other low-level programming guidelines. This helps to maintain some consistency in the developed code. Unit testing is performed by the developers. The independent test team is responsible for performing system integration and acceptance testing.

Planned V,V&T activities will be more formal and more rigorously followed. Standards are currently being developed to more precisely delineate the software development lifecycle to be followed and the particular activities to be performed within each phase. Structured analysis, design, and programming techniques will be utilized with a strong emphasis on formal walkthroughs to ensure standards adherence and to verify the correctness of the sytem. All code will be developed in a high-level language (COBOL) which will allow for the use of more widely available V,V&T tools.

2.5 Configuration Management

No formal configuration management function exists beyond that required for standard user change-request processing. User change requests are reviewed and the feasibility of their implementation is studied. If feasible, the change request is assigned to a branch chief and preparation of a requirements package begins.

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

Current formal V,V&T activities are performed by the independent test team. This group participates in the preparation of the Requirements Analysis Package (RAP), which is then used by the test team to plan and prepare tests for the system. This activity is done in parallel with system development. The tests are prepared by predicting, based on the RAP, what the output should be and creating the appropriate input data. This involves much time and laber which is why it is done concurrently with software development. Once the software is developed, it is then delivered to the test team for system testing (unit testing is performed by the developers). Errors detected during the testing period are formally noted and given to the developers for correction. When a system is formally released for operation the test team and the controlling branch sign-off. If any unresolved problems remain, an exception list is included as part of the formal sign-off. Systems are generally released in time, even if they are not quite totally operable.

3.2 Tools

Virtually all of the current software for this site is coded in assembly language. As such, very few V,V&T tools are available to aid code development. There does exist an interactive debugger but it is not generally used because of lack of guidelines for its use. File comparators are available and are used to some extent. A test- bed support facility is used to maintain a file containing a historical record of input data and the predicted output of individual tests performed by a given system. It is used to analyze the completeness of the data preparation and as an aid in output analysis.

Tools have been and are being used to support project management activities. Two project management tools have been used in the past, but without much success. A third tool is viewed favorably and will be used extensively to manage the conversion project.

At the present time, use of other more advanced software development and testing tools is looked upon as still being 2 to 3 years in the future.

3.3 Perceived Problem Areas

The most prevalent problems are last-minute system-requirements changes (often resulting from late legislation), resulting in late delivery of a system to the test group. Production deadlines are rather firmly fixed, so that system test time is generally reduced rather than delivering the system late. Moreover, because an independent test team is responsible for system testing, unit testing is often not performed to an adequate degree, because it is felt that the test team will find all problems.

Other problems mentioned included the lack of specific procedures to support division guidelines and the lack of a cost-effective means to enforce adherence to the guidelines (Section 2.3.1).

4.0 STANDARDS, GUIDELINES AND PROCEDURES

Site G2 has adopted a commercial software development methodology and is in the process of conducting a large-scale training program to educate programmers, analysts, managers and internal auditors in the methodology. The methodology defines a software development lifecycle, techniques to be used during particular phases, and rules and procedures to achieve standardization.

Particular standards include those for conducting walkthroughs, definition of data dictionaries, design structures, etc.

APPENDIX C

GOVERNMENT SITE 3 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

This site is a large Federal administrative agency charged with managing several large Federal funds disbursement programs. The end user community in this case is the network of field offices. Internally, the organization is divided into divisions and groups within the divisions. The operations of two of these divisions, the Systems Development Division, and the division which interfaces between the user community and the Systems Development Division, are the primary focus of this site report. The functions of several other divisions will be discussed as appropriate.

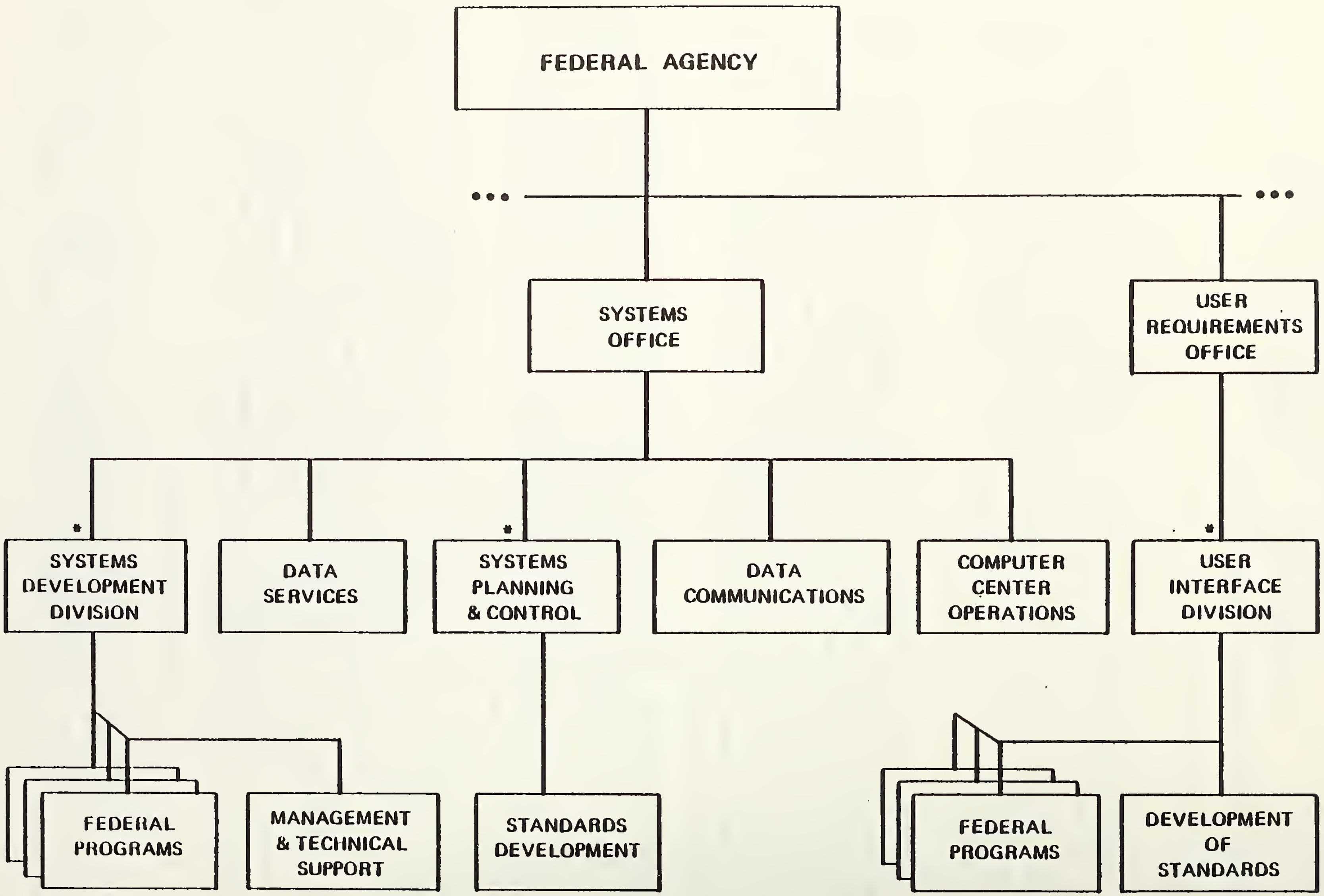
The Systems Development Division is a subpart of the systems office. Other divisions of the systems office include: Computer/Data Center Operations, Data Communications, Data Services, and Systems Planning and Control. The data services division performs all of the management/administrative system development and maintenance. The System Planning and Control Division is responsible for several of the administrative and control functions for the entire systems office. It is also charged with the development of software, and maintenance standards for the division. These latter two divisions also participated in the survey.

The Systems Development Division is responsible for the development and maintenance of the programs used in the administration of the various Federal funds disbursement programs. It is divided into groups according to Federal program (i.e., one to two groups supporting a given Federal program). There is also a management and technical support group, which is involved with the development of systems and maintenance standards, and the introduction of new technology into the development division.

The second division plays a key role in the software development and maintenance activities at the site, and will be discussed is the user interface division. This division is divided into groups in a fashion parallel (i.e., by Federal program) to the systems development division. This division is responsible for interfacing between the user networks of these various programs (i.e., the field offices which administer the programs) and the development personnel who develop and maintain the associated software. The interface function includes: the specification of requirements for each system addition or modification; the development of a validation plan; and use of the plan for the validation of the results.

1.2 Description of Software Activities

The Systems Development Division is involved in a mixture of maintenance, enhancement and new development activities. The majority of the work performed is either maintenance or enhancements. COBOL is the primary language used. The new development is usually to implement a new Federal



*Interviewed for Survey

Figure C-1. Site G3 Organizational Structure.

SITE SUMMARY
G3

General Characteristics		
Group Name	:	System Development & User Interface divisions
Type of Activity	:	Administration of Federal funds disbursement programs
Application Area	:	Program administration
Languages	:	COBOL
Project Size Range	:	Small: 1 month or less; Large: 6 months or more
Development/Maintenance Technology		
Development, Maintenance or Other	:	Majority maintenance and enhancements. A little development
Documented or Informal Approach	:	Standards being developed
Number of Phases	:	7 : Initiation, Requirements, Design, Development, Validation, Implementation, Operation & Audit
Use of Formal Techniques/Tools	:	Very Informal
V&V Technology		
Requirements as a Baseline	:	Informally documented
Other Levels of Specification	:	Informal design documents
V&V Within/Between Early Phases	:	None
Testing Practices	:	Unit testing and acceptance testing. Both Informal and occur at analyst level
Reviews	:	Namely peer groups and reviews only on large projects
Techniques/Tools	:	Two main tools. Both used to prepare test data
Documentation Practices		
Formal/Informal Guidelines	:	Mixture of Informal and Formal guidelines including FIPS PUB 38
Adherence to Guidelines	:	Loosely followed. Depends primarily on size of project
Enforcement/Review	:	Informal peer reviews

program or a significant change to an existing one.

The user interface division is not directly involved in any software development or maintenance. Much of their work is actually answering questions and explaining how to interpret data, read reports, etc. Only a relatively small portion of the user contacts actually result in system problem reports or change requests. When a system change is necessary, this group prepares the system requirements statement and a plan for validating the results. The validation plan would at least describe what is to be tested and might go so far as to actually specify some of the test data to be used. The group subsequently examines the results of the test procedures to assure correctness and then formally accepts the results (if correct).

1.3 Factors Influencing the Environment

One of the most apparent factors that has affected and will continue to affect the operations of both the systems development group and the user interface group is the effect of congressional legislation on the operation of this agency. For example the last major program initiated was given 14 months by Congress for implementation after the act was signed. The wait for policy decisions and interpretations cut this time nearly in half for the development of a major software system. Extreme schedule constraints because of legislation was cited on numerous occasions as causing shortcuts to be taken during development and testing, and ultimately causing many problems after the system was put into production.

1.4 Historical Perspective/Evolution

A recent reorganization established the user interface group and has redefined charters and responsibilities of much of the systems office. Actual operating procedures and interfaces between groups are still being worked out. The plan appears to be a viable solution to some of the problems of the past. Currently, though, the affected groups are in the middle of a major transition, with many of the final details of the implementation yet to be decided.

An indicator of the transition, uncertainty, and, to a degree, competition that is taking place is the number of efforts currently underway to define procedures and standards. Under the systems office there are two groups working on standards, one in the Systems Planning and Control Division, and one in the Systems Development Division. In the user interface division there were also several areas being addressed through the development of standards. All three groups were addressing system validation and the roles and responsibilities of those involved, and the form of the interface between the groups. (The validation procedures will be discussed in section 2.) There were other areas of overlap as well, with no apparent mechanism for resolving the descrepencies between these independently developed standards.

2.0 DESCRIPTION OF SOFTWARE APPROACH

2.1 Overview of Development Approach

A seven phase development approach was followed. The phases are listed below.

- Initiation
- Requirements definition
- Analysis and design
- Development
- Validation
- Implementation
- Operation and audit

The User Interface Division has responsibility for all of the activities and products of the initiation phase and the requirements definition phase. The Systems Development Division is responsible for analysis and design, and then development. The validation phase was a joint responsibility. Implementation and operation audit are primarily the responsibilities of the Operations Division and the user community (i.e., the audit part).

2.2 Phase Description

2.2.1 Initiation Phase

This phase was accomplished by the user interface group with inputs from the user community and the system development staff. The products are a project request and, when appropriate, a feasibility study and a cost-benefit analysis. A project request might be the result of one or more communications, requests or problem reports from users. With the majority of projects being maintenance or enhancements to existing systems, the project requests (and other products when produced) were relatively simple. Project requests were formally submitted to the planning and control division within the systems office.

2.2.2 Requirements Definition Phase

The requirements definition phase results in a requirements specification document which accompanies the project request. The requirements specifications were usually functional in nature, and, when appropriate, specified report formats or changes to additional reports, data records, etc.

Along with the requirements documentation a validation plan was prepared. The detail of this plan varied from project to project, but the objective was to specify the types of testing that were deemed necessary, and the results expected. The guidance provided ranged from the databases, to the types or classes of records, to the actual data (or portions of) that were to be used. This plan was then given to the systems development division.

2.2.3 Analysis and Design Phase

The activities of this phase are completed by the staff of the systems develpment division. Most change requests for a given system are handled by an available analyst from the group responsible for maintaining the system.

It has been true that for most projects the formality and rigor of this phase are dependent upon the analyst(s) performing the task. Designs are not usually formally documented, though the FIPS documentation standards (FIPS PUB 38) are loosely followed. Reviews, when held, are informal peer reviews. Neither the user interface group nor the user is involved.

The larger development projects which require a team of analysts are approached in a more formal fashion. These projects, though, are infrequent and as mentioned previously are often under severe time constraints, usually causing shortcuts to be taken.

New guidelines are being developed by the staff of the management and technical support group of the Systems Development Division. These guidelines will help standardize the activities and products of this phase. They outline the products to be produced, and the producer, reviewer, and acceptor of each. These guidelines are specific to four project levels, outlining the degree of formality to be used depending upon the size of (and other factors relating to) the project. For example, for a level 1 project (using the estimated effort criterion greater than 24 work weeks) there are eight documents suggested as products of this phase. These range from a requirements evaluation report to a program specification document. Both unit and system test plans are included.

2.2.4 Development Phase

This phase encompasses all the coding and unit testing activities. In the past, it has been an informal phase performed by the analyst(s), with little visibility provided to management and none to the user or user interface group. The predominante strategy and method of testing is the use of data extracted from production files.

The new guidelines mentioned in the last section will help to define specific products to be produced during the development phase and, if followed, should increase visibility and controllability of software projects.

2.2.5 Validation Phase

This phase did not explicitly exist prior to the reorganization and the establishment of a formal validation function. It is to include system test activities and acceptance or validation testing. Prior to the reorganization, system testing was performed by the systems development group. User acceptance was an informal process through the existing interface between the user and the analyst.

The recent reorganization, through the establishment of a user interface group and its charter which includes system validation, has formalized this function at this site. Standards are currently being developed which will define this function and specific roles and responsibilities.

The interface between the user interface group and the systems development group is still evolving. Currently, a validation plan is prepared by the user interface group and submitted with the statement of the requirements. Upon completion of the development and testing, the development group provides the results requested in the validation plan. The analysis of these results is the responsibility of the user interface group. A test report is to be prepared and if the results are deemed satisfactory (i.e., the requirements have been satisfied), then permission is given to release the product. (The user interface group does not perform any of the validation runs of the system. They do not have access to the computer and the software.)

2.2.6 Implementation

This phase is primarily the responsibility of the System Development Division and the Operations Division. As planned for implementation in the standards prepared by the Systems Development Division, the user (or user interface group) will prepare an audit plan as part of the implementation and for use during the post-implementation review of the system.

2.2.7 Operation and Audit Phase

During the operation of the system, if problems arise then a user files a report with the interface group. As stated in the drafts of the new standards, there is to be a post-implementation review and/or audit performed with the results being documented. This has not been regularly done in the past.

2.3 Quality Assurance Activities

Currently there is no explicit quality assurance program. The reorganization and the standards that are being developed (and are planned) will provide the basis and mechanism for a QA program.

2.4 Validation, Verification and Testing Activities

As described in section 2.2.5 there is a validation phase defined in the development and maintenance approach being followed. The activities, roles, responsibilities and mechanisms for accomplishing the objectives of this phase are currently being defined.

2.5 Configuration Management

The Operations Division has the primary responsibility for the configuration management activities. This division was not included in the survey.

3.0 TECHNIQUE AND TOOLS

3.1 Techniques

The only V,V&T techniques used were walkthroughs, peer groups, reviews and inspections which were used on the larger projects. As the environment matures, (e.g., standards are completed and implemented, and experience is gained in using them) the use of techniques, e.g., reviews and walkthroughs, will increase and become more formal and rigorous.

3.2 Tools

The systems development group used several different tools in support of its testing program. Two tools used assisted in preparing test data. One tool was identified that assisted in test data generation based upon a description of the data work area and the logical structure of the procedure division. Another tool was used to select records from a production file for testing based upon specified selection criteria. A third tool was described which was used to control test executions and compare test results. It would execute production code and modified code on a given set of test data and compare and report on the results. A file comparator was also used as a less sophisticated means of comparing test related files.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

The FIPS documentation standards (PUBS 38 and 64) were used as the basis for the documentation practices. These were loosely followed, and questions of the consistency between and the overlap of the two standards were voiced.

As a result of the reorganization and a general effort to establish a more modern, well defined and formal environment, several groups were preparing standards covering many different areas including software development and maintenance phases and products, validation, and software change control.

APPENDIX D

GOVERNMENT SITE 4 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF THE ENVIRONMENT

1.1 Organizational Overview

Site G4 is a Federally-funded computer center serving seven Federal organizational entities both within and external to the department of which it is organizationally a part (Figure D-1).

Site G4's charter is to supply a general computer system service to provide large mainframe processing and teleprocessing and minicomputer support. This includes a software support group providing software development and maintenance services to the computer center customers. The functions, activities and practices of this group were the subject of the survey.

The customer organizations served by Site G4 have their own software development/maintenance staff. A request for support from a customer from this group would be made for one of several reasons:

1) additional support is needed,
2) the needed expertise is not available in-house,
3) the service requested involves a system in the domain of the support group.

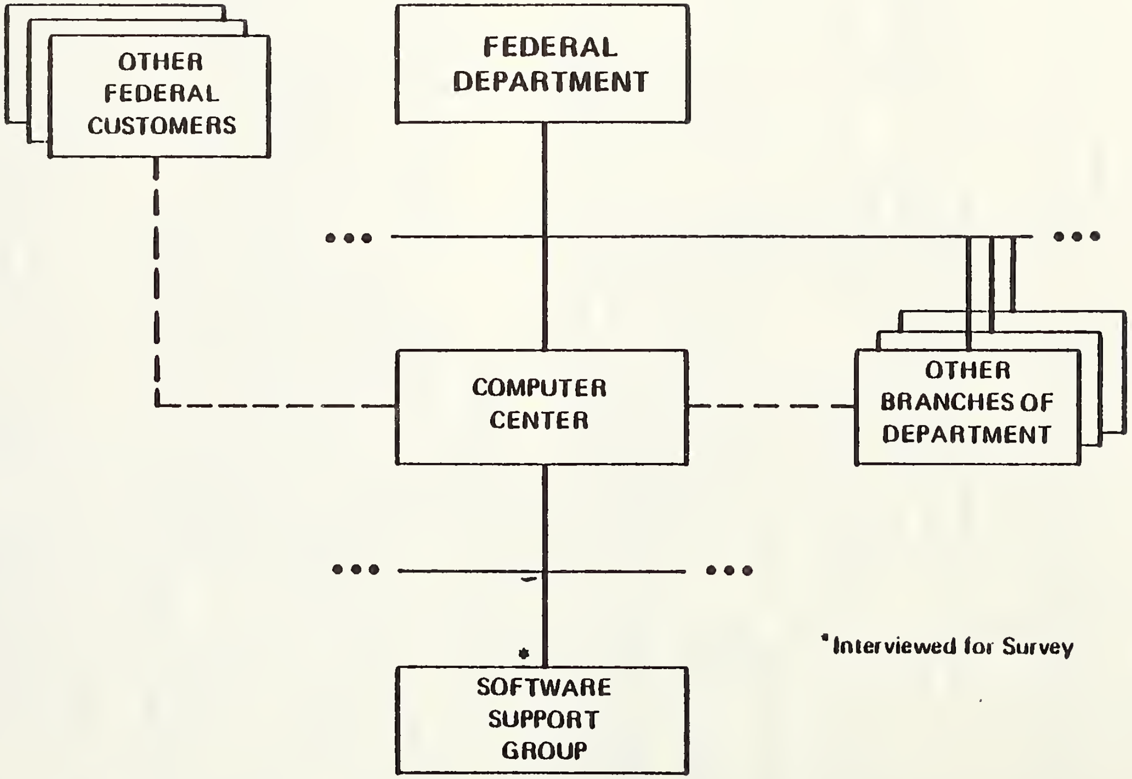
1.2 Description of the Software Activities

The services provided by this group include software development and maintenance support. The majority of development jobs are small (1 to 4 person months). Maintenance support includes error correction and enhancements to existing programs that may or may not have been originally developed by the group. The software being developed/maintained is almost exclusively in COBOL. A few applications interface with a database management system, and there are some assembly language projects (primarily for mini-computer applications). The staff is divided into three principal support areas: financial, applications, and database.

1.3 Factors Influencing the Environment

Several factors have influenced the operation and activities of Site G4. The group functions on an on-call basis, predominantly performing small development or maintenance jobs with a requirement for quick turnaround. The group has a very large customer community with a wide variety of demands. The environment requires flexibility and a broad range of staff skills.

There are few recognized or documented standards or procedures, leaving the practices employed to the discretion of the individual analyst. In many instances, the primary emphasis is on the timely completion of the programming task.



*Interviewed for Survey

Figure D-1. Site G4 Organizational Structure.

SITE SUMMARY

G4

General Characteristics	
Group Name	Computer center - software support group
Type of Activity	To provide programming support to computer center customers
Application Area	Federal funds disbursement
Languages	COBOL
Project Size Range	Small: 4 months or less
Development/Maintenance Technology	
Development, Maintenance or Other	Maintenance & enhancements with small development projects
Documented or Informal Approach	No existing guidelines or documentation for approach. Standards under development
Number of Phases	2 : Definition, Development
Use of Formal Techniques/Tools	None
V&V Technology	
Requirements as a Baseline	Not used
Other Levels of Specification	Informal design specification
V&V Within/Between Early Phases	None
Testing Practices	At discretion of each analyst
Reviews	Only customer acceptance
Techniques/Tools	None used
Documentation Practices	
Formal/Informal Guidelines	Only informal guidelines
Adherence to Guidelines	Low adherence to guidelines
Enforcement/Review	Only review is by customer or initial approval and final acceptance

The demand for teleprocessing and remote-station services has significantly increased in recent years. This has affected the processing time available for development and test support on the system.

The group size, combined with the broad base of activities, results in instances of great dependence on single individuals for areas of expertise. While personnel turnover is not identified as a problem area, these conditions result in any turnover at all being a potentially large problem.

1.4 Historical Perspective/Evolution

A recent reorganization combined formerly independent groups and provided a new director. The new director has an industrial background and is effecting some significant changes in the environment. The director is in the process of promoting better management practices and techniques and has also done a fine job of building and maintaining good personnel relations.

2.0 DESCRIPTION OF SOFTWARE ACTIVITIES

2.1 Overview of Development Approach

There are no standards or guidelines which define an overall approach to software development/maintenance or practices to be employed during a project. During the interviews, individuals gave their perceptions of the common practices. The following descriptions of phases summarize these.

2.2 Phase Descriptions

2.2.1 Definition Phase

A software project is initiated through the submission of a request for services. These requests are reviewed and assigned for follow-up. Initial follow-up consists of a series of contacts between analyst and customer to establish a work statement and estimate. These are documented and reviewed by both the software and customer organizations. Upon approval, the development phase begins.

2.2 Development Phase

After approval, the job is assigned (probably to the same analyst) for completion. From this point, the job is entirely the responsibility of the analyst. The analyst performs all design, code and test activities. The techniques and practices followed are at the discretion of the analyst. All communication is between the analyst and the customer.

Normally, the customer's final review is an examination of the program outputs and their format. Acceptance is an informal agreement between analyst and customer.

2.3 Quality Assurance Practices

There are no standards or any form of independent review to assess quality or correctness. The customer review of outputs is the only review performed.

2.4 Validation, Verification and Testing Practices

These functions are assumed responsibilities of the analyst. Desk checking and testing are the V,V&T practices used as needed in the judgment of the analyst.

2.5 Configuration Management

There are some change control procedures followed for the financial systems. Approved change requests are traced to ensure completion, and corrections and enhancements are incorporated via block updates.

3.0 TECHNIQUES & TOOLS

3.1 Techniques

There is a review to mark the completion of the problem definition step. It is primarily for funding justification and is not technical in nature. Desk checking is practiced by individual analysts. There is also an informal final acceptance review performed by the customer.

3.2 Tools

A software tool is available for data documentation of COBOL programs, but its use is not promoted.

3.3 Perceived Problem Areas

The most often perceived problem was the lack of unambiguous, static requirements. The present system works from a sketchy requirements or informal problem statement. These are usually modified continually throughout the design and development work, causing problems. Several survey participants felt that time spent getting better and more formal job statements from the customers would improve the productivity and product quality greatly.

There were also several expressions of need for documented standards to unify the activities. Such an effort is already underway (see Section 4.0).

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

No documented standards exist at this time. The need for standards is acknowledged by both management and staff. An activity is being initiated to define and develop standards addressing project management and software development practices.

APPENDIX E

GOVERNMENT SITE 5 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site G5 is a large Federal agency, with all computer related activities controlled by the Information Systems Division. This division (figure E-1) is subdivided into three branches: 1) budget and resource planning, 2) systems development, and 3) systems management. The systems management branch is subdivided into two sections: 1) an administrative section (responsible for such things as the development of standards, guidelines, and procedures), and 2) a systems maintenance section (responsible for maintaining current production software systems). Users and customers are synonomous within the scope of this organization. Users and project management have the most significant impact on Site G5 software development activities.

1.2 Description of Software Activities

The software activities performed by Site G5 support nearly all agency level, as well as some department level, data processing needs. Supported applications are primarily business and information processing oriented with systems such as inventory, budget, payroll and personnel. Some engineering and control applications are also supported. Nearly all programming is done in COBOL in a batch processing environment. The software is currently running on old medium-scale computers with conversion to more modern, larger computers underway. The system can be accessed by users through remote job entry stations located in 12 district offices around the country.

Software development and maintenance activities are managed and performed by two separate branches, the systems development branch, and the systems maintenance section of the systems management branch (See Section 1.1). The software developers are seldom responsible for maintaining the software.

1.3 Factors Influencing the Environment

Several factors influenced software development activities within this organization. The computing hardware is not widely used. As a result, there were very limited software resources to aid in the development activities. Personnel was also seen as a major environmental influence. The size of the staff is not large enough to meet the demands for computing services. Moreover, personnel turnover is high.

1.4 Historical Perspective/Future Direction

This site has been using medium-scale, fairly uncommon computers for the last 10 to 12 years. The site is currently, however, in the process of converting to a newer, more popular computer system. The new system will be accompanied by a greater number of software support tools, and there is an interest within the agency to utilize some of these resources.

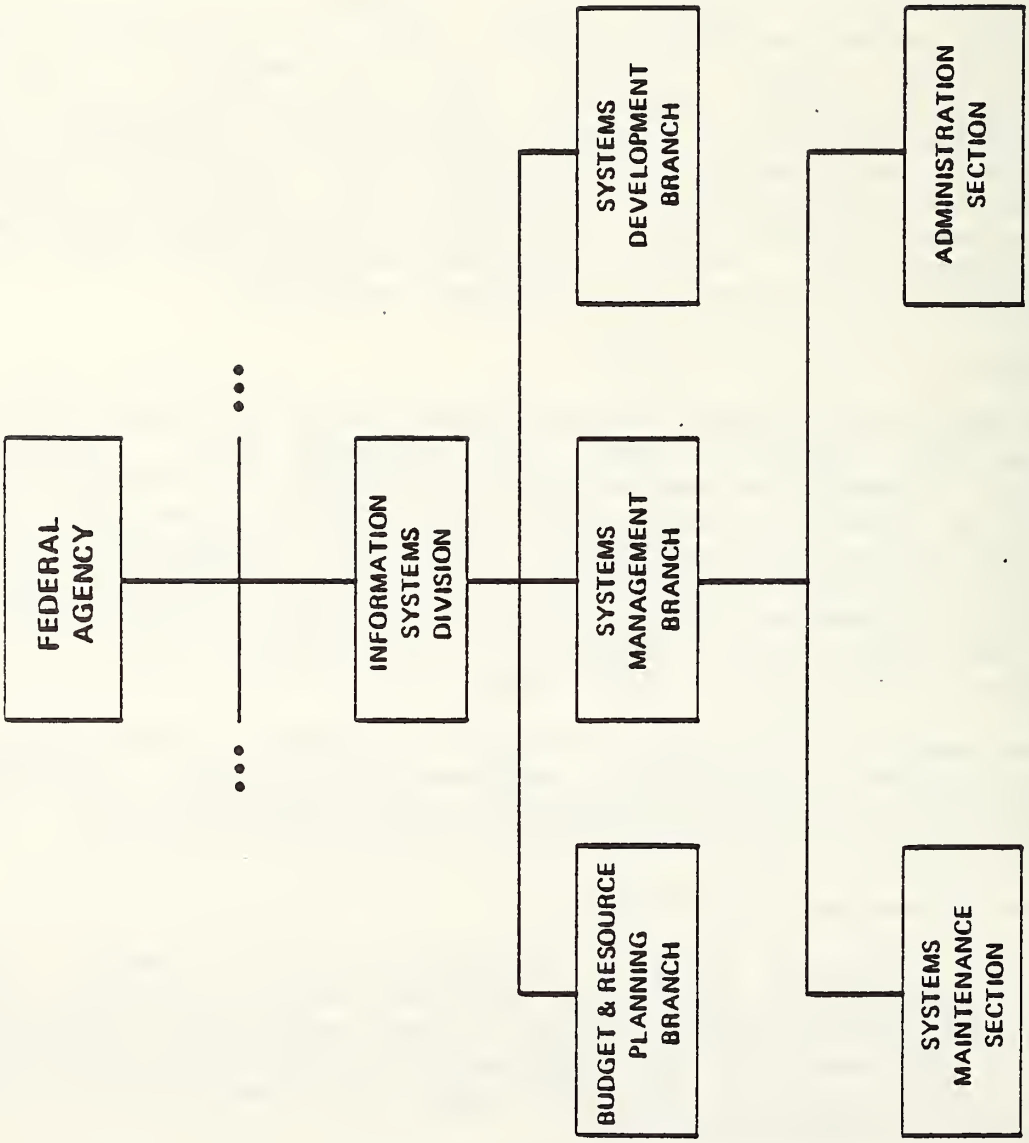

FEDERAL AGENCY
INFORMATION SYSTEMS DIVISION
BUDGET & RESOURCE PLANNING BRANCH
SYSTEMS MANAGEMENT BRANCH
SYSTEMS DEVELOPMENT BRANCH
SYSTEMS MAINTENANCE SECTION
ADMINISTRATION SECTION

Figure E-1. Site G5 Organizational Structure.

SITE SUMMARY
G5

General Characteristics	
Group Name :	Information Systems Division
Type of Activity :	DP support at Agency and Department level
Application Area :	Operations
Languages :	COBOL
Project Size Range :	Small: less than 3 months; Large: more than 12 months
Development/Maintenance Technology	
Development, Maintenance or Other :	Development & Maintenance and separated into two groups (exclusive)
Documented or Informal Approach :	One manual exists for coding standards
Number of Phases :	4 : Specification, Deisgn, Coding & Testing, Maintenance
Use of Formal Techniques/Tools :	None used
V&V Technology	
Requirements as a Baseline :	Not used
Other Levels of Specification :	Design
V&V Within/Between Early Phases :	Heavy emphasis on test planning during design phase
Testing Practices :	Modular testing is used successfully
Reviews :	Walkthroughs, design reviews
Techniques/Tools :	Recently acquired a test support tool; not yet in use
Documentation Practices	
Formal/Informal Guidelines :	One manual contains standards (developing new standards from FIPS PUB 38)
Adherence to Guidelines :	Not followed rigorously
Enforcement/Review :	Formal reviews to decide on project feasibility; otherwise informal technical reviews

A manual does exist which contains coding and documentation standards and procedures. These standards were not followed during the development of much of the existing software though. Reasons given for this were that the standards were too voluminous, detailed, complex, and confusing. Also the standards manual did not provide any examples. In addition, there was no management control provided to ensure that the standards were adhered to.

A new set of standards, based on FIPS 38, is currently being developed. These new standards should be simpler, more flexible and adaptable to a particular project's needs.

There has been a growing awareness over the past few years of the importance of test planning during development. Test plans are currently required to be developed during the design phase with an emphasis on modular testing during development. This approach has met with good success, both in terms of cost-effectiveness and reliability.

2.0 DESCRIPTION OF SOFTWARE ACTIVITIES

2.1 Overview of Development Approach

The software development approach followed by Site G5 consists of four phases: feasibility/requirements specification, design, coding and testing, and maintenance. The first three phases are performed within the systems development branch. The maintenance phase is handled by the systems maintenance section.

2.2 Phase Descriptions

2.2.1 Feasibility/Requirements Specification Phase

The feasibility/requirements specification phase involves the formal handling of user requests for computing services. When a user request is received, a feasibility analysis is performed by a combination of technical and management personnel. This analysis consists of estimates of labor costs and computer resources needed to carry out the task. Examinations to determine whether any existing software is available that can perform the desired function are also performed. Potential contractor use is also studied. The results of the feasibility analysis are returned to the user, including cost and level-of-effort estimates. It is then up to the user to decide to go ahead with the work. Once the user authorizes the work, a project manager is selected and requirements are developed and documented according to a set of documentation standards. The process includes extensive user interaction and formal division-level reviews.

2.2.2 Design Phase

This phase consists of planning for the development phase, test plan development, and software design. Documentation standards (e.g., system flow charts, interface descriptions) are followed, but are somewhat inflexible in that they don't always address the needs of a particular project. Test planning is performed in conjunction with software design to promote modular and system testing. No formal software design methodologies are followed. Informal design reviews are held with technical personnel on a periodic basis.

2.2.3 Coding

Most of the software developed by this agency is coded in COBOL. Both division-level and project-level standards are followed during the coding phase. Informal code walkthroughs are utilized with a high degree of effectiveness. Modular testing is performed with test analysis reports being required. This also has had very effective results. Test and procedures are not saved for use during the maintenance phase.

2.2.4 Maintenance Activities

Within this agency all maintenance activities are performed by a group which is separate from the development group. A single programming analyst is generally assigned the sole responsibility for the maintenance of a software product.

2.3 Quality Assurance Activities

There is no formal software quality assurance organization within this agency. It is a responsibility assumed by line or project management. Formal requirements and change control reviews are always held. Formal design reviews are sometimes held. Test reviews are always held and can be both formal and informal. Monthly project progress reviews are required and are prepared and conducted by the project manager for line and upper management personnel. This review reports project progress from both a cost and schedule standpoint and from a technical standpoint.

New documentation standards are currently being developed. They are being patterned after and are amplifying upon the guidance presented in FIPS 38. It is hoped that these new standards will be more specific to the environment and adaptable to specific projects. They are to cover program and system documentation and will specifically address issues such as test planning and analysis and disciplined programming approaches.

An independent test group was organized about four years ago. The group was responsible for software test and acceptance activities. The group was not successfully utilized and was dissolved. Some of the reasons stated for this lack of success were that the group was understaffed and that use of the group led to schedule slippages. Developers were reluctant to use the group because, instead of relieving some of the testing burden, the use of the group usually increased the amount of work necessary to deliver a software product. The group's function was not understood nor were guidelines available on how an independent test group could be effectively utilized.

2.4 Validation, Verification, and Testing Activities

No formal division-level V,V&T approach currently exists. V,V&T practices are established by the project manager for each project. There is a heavy emphasis on test planning during the design phase. A commercially available test support tool (test-data generation and library support) has been investigated and acquired. It will probably be used on future projects. As was previously mentioned, modular testing has been utilized with a high degree of success.

2.5 Configuration Management

No formal configuration management (CM) organization exists within this agency. Software configuration control is a responsibility assumed by the project manager and maintenance analyst.

Formal procedures for processing user change requests do exist. When a change request is received, it is first assigned to a review group which analyzes the feasibility of the change. The results of the analysis are written up as a formal memo. If the change is feasible, the estimates of the effort and resources required are given. When a change is approved it is assigned to the analyst responsible for maintaining the affected system. Major system changes usually receive informal user sign-off when complete, but such sign-off is not required. Small changes which take less than a day to implement do not pass through formal change request processing, but are often implemented directly by the maintenance analyst. A record of all changes, however, is maintained. This procedure is acceptable both to the users and the maintenance personnel.

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

The V,V&T techniques summarized here have already been mentioned in previous sections. The V,V&T techniques utilized at this site are listed below.

1. test planning and analysis early in the development,
2. formal requirements/preliminary design reviews,
3. informal design inspections,
4. informal code walkthroughs during development,
5. formal code walkthroughs during implementation,
6. development and use of coding and documentation standards.

These techniques have generally met with success. The utilization of an independent test team was tried, but was not successful (see 2.3 Quality Assurance).

3.2 Tools

Currently, no automated V,V&T tools are being utilized by this agency. However, personnel from this site have recently attended a class on a test-data generation and support tool. This tool uses a COBOL preprocessor which extracts information about specific tests imbedded within special COBOL comments, and creates a test bed. There is a great deal of interest in this tool and its use is planned in an upcoming project.

Two COBOL dynamic analysis tools which provide execution information (such as statement execution counts) have been experimented with, but have not, as yet, been used on a project.

3.3 Perceived Problem Areas

Conversion to a more widely used computer system has greatly increased the number of V,V&T tools available. This has alieviated one problem. The current problem is finding information on the tools available and guidelines on how they can be effectively used. The lack of available information and experience together with a resistance to change have inhibited the application of tools and techniques within this agency.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

Previous sections have discussed the standards used at this site. Division-level guidelines are currently being redeveloped from an old massive set of complex standards and procedures to a new, much simpler set of general guidelines supporting FIPS 38. These standards basically address the format and content of system, program and user's documentation.

Project-level coding and documentation standards are established for individual projects by the project managers and apply only to that particular project.

APPENDIX F

COMMERCIAL SITE 1 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site C1 is a medium-sized corporation which includes several manufacturing companies and their sales and services organization (figure F-1). There exists one central computer support group, Management Information Systems (MIS), for the entire corporation. This group consists of three divisions. The first, Technology, is in charge of data center operations and system support. The second, System Development, is responsible for sustaining the operation of the existing systems, i.e., maintenance and enhancement. This includes some new development of capabilities related to the functional areas currently supported. The third division, Major Projects, is a recently (within the last 18 months) formed group incorporating three large (at least 5 years of effort) software development projects.

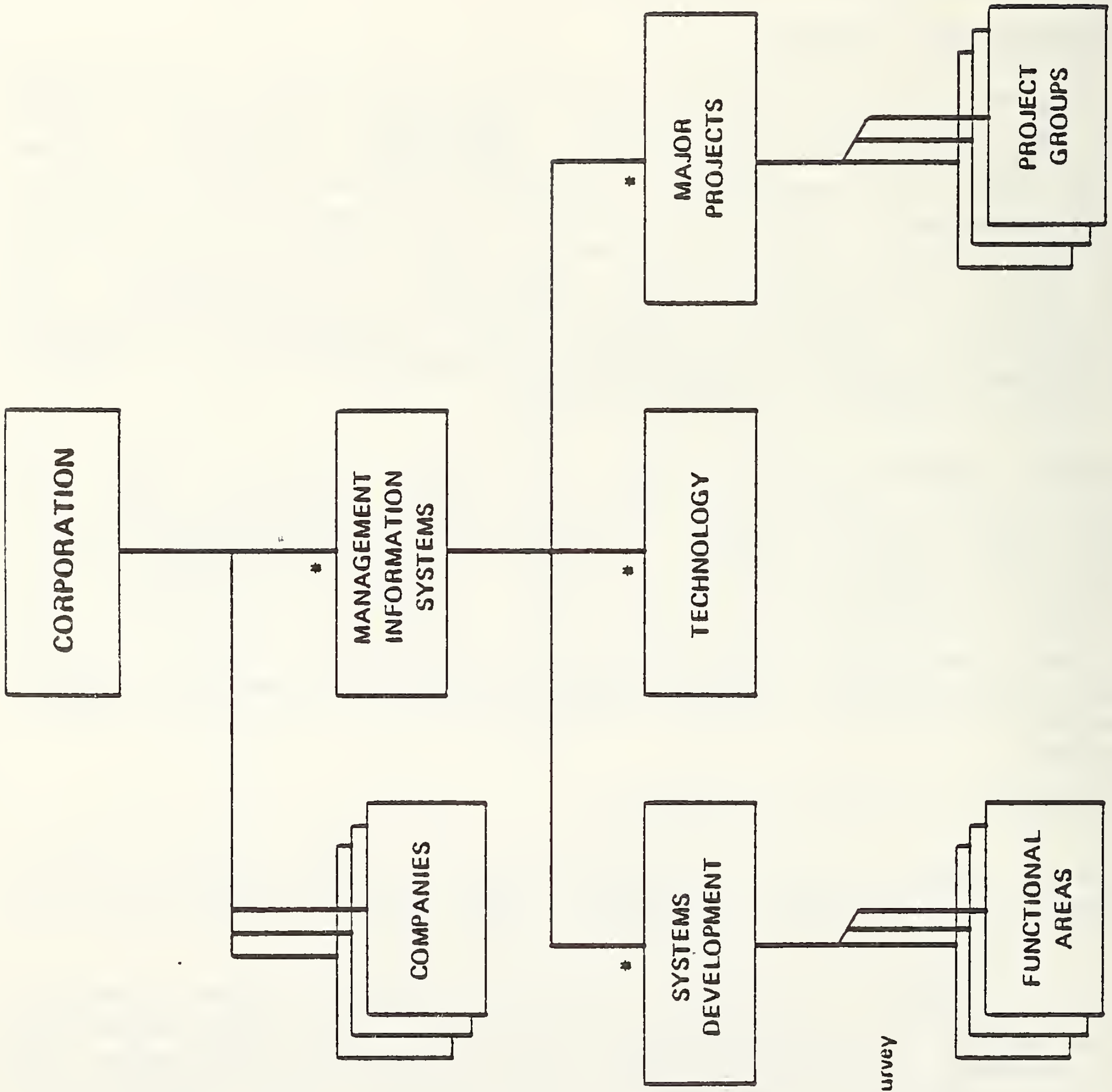
The customer community of the MIS organization is quite varied. The Technology Division primarily supports the other divisions within MIS, ultimately supporting the corporate customers who utilize the various types of information and reports supplied to them. The System Development Division supports all the administrative functions (e.g., finance and personnel), the manufacturing functions (e.g., inventory), and sales and service (e.g., customer orders and parts). The customer set of the Major Projects Division includes those specific customer groups of each of the three projects being performed by the Division. The cost of the MIS group is distributed back to the customers.

1.2 Description of Software Activities

The three divisions are fairly distinctive in terms of their software support, development and maintenance activities. The organizational setup clearly reflects these differing functions and responsibilities.

The Technology Division is solely involved in system support (hardware, operating system and utilities) and maintenance. All programming is in assembly language. Tasks are generally small and assigned to one person with that person assuming the responsibility for the entire development process (requirements, design, code, test and implementation). The environment is rather informal with respect to the use of standards and a phased-development approach.

The System Development Division is responsible for sustaining and enhancing existing production software. Tasks generally are one of three types - responding to a problem report, implementing a system enhancement, or processing special one-time requests. The majority of tasks are small (less than 300 hours). A task estimated at 700 hours or above is considered a large project. The vast majority of the code is COBOL, with a small portion of PL/I. Certain report-producing applications utilize a commercial



*Interviewed for Survey

Figure F-1. Site C1 Organizational Structure.

SITE SUMMARY
CI

General Characteristics		
Group Name	:	Management Information Systems (MIS)
Type of Activity	:	System Support, production support, new system development
Application Area	:	Management, Administration & Operation of manufacturing
Languages	:	COBOL and some Assembly, PL/I
Project Size Range	:	Small: less than .25 months; Large: more than 12 months
Development/Maintenance Technology		
Development, Maintenance or Other	:	Development, maintenance, enhancement
Documented or Informal Approach	:	Certain standards are documented, loosely followed
Number of Phases	:	3 : Definition, Construction, & implementation
Use of Formal Techniques/Tools	:	None
V&V Technology		
Requirements as a Baseline	:	Not used
Other Levels of Specification	:	Design
V&V Within/Between Early Phases	:	Little, Informal, more for larger projects
Testing Practices	:	Informal
Reviews	:	Two reviews are standard, one between the definition & construction phases, and an acceptance review
Techniques/Tools	:	Data Dictionary, source-code management system
Documentation Practices		
Formal/Informal Guidelines	:	Formal standards & guidelines
Adherence to Guidelines	:	Little enforcement or monitoring
Enforcement/Review	:	Varies by project size through reviews

report-generation query language. MIS standards (see Section 4) provide a framework for the development approach followed. The actual practices followed varied widely from project to project depending upon factors such as size, complexity, the analyst in charge, the particular customer and time constraints. In general, a phased approach is followed involving various degrees of management and customer review and sign off (see Section 2).

The Major Products Division was created with the initiation of three large software development efforts. It was a realization of the "special handling" that would be required because of size, complexity and importance of the projects. The development approach being taken in these projects is evolving from that outlined in the MIS standards (Section 4). More and better documentation, more customer involvement and a greater emphasis on system requirements and high-level design are three indications of the increased formality and rigor of the approach being taken on these large projects.

1.3 Factors Influencing the Environment

Two closely related factors are apparent and seem to have a pervasive impact on the entire MIS organization. The first is related to the personnel of the organization. The management and technical personnel in general are progressive and vital. Communication is open. There seems to be a relatively low level of resistence to change and new technology. This environment created by the personnel facilitates the modernization of the MIS organization that is taking place. This state of transition is the second factor affecting the environment. The transition/modernization of the organization is recognized and supported by nearly all those interviewed.

There are several factors that were identified that possibly have a negative impact on the software development activities. There do exist standards or guidelines for certain activities or portions of software development/maintenance, yet there is little monitoring or enforcement. One probable reason for this is that there appears to be a general question of the applicability and/or usability of the guidelines. Another reason is related to the interface between the MIS staff (primarily in the System Development Division) and their customers. In many cases there is a close, informal working relationship. This creates the belief that there is little need for following the established or recommended procedures. The formality of documenting and reviewing decisions often is seen purely as a time consuming exercise. This close working relationship was apparently the cause of another practice that had the potential for creating problems. Even when reviews are held, technical direction is shifted after the review. After the pre-coding and implementation sign-off, customer-analyst decisions are made which significantly affect the final product. The lack of effective monitoring and enforcement contributes to an informal working environment that at times can result in problems.

Another factor relating to the customer and management visibility and monitoring is the fact that the procedures in existence call for only one formal review. In at least one interview, the need for more checkpoints and reviews was voiced.

One final factor to be mentioned relates to the Systems Development Division. There is an emphasis on meeting scheduled delivery dates. It appears that this often took priority over thorough testing. The result was a significant number of fixes and corrections being made to production software. The unwritten rule followed at least to some degree was "meet the schedule and fix it later."

1.4 Historical Perspective/Future Direction

As previously indicated, the MIS organization is in a state of transition. Perhaps the most predominant evidence of this transition was the creation of the Major Projects Division within the MIS organization. It was recognized that these new projects would require "special handling", i.e., a more formal and rigorous development approach.

In general, within the MIS organization, there is an attempt to improve the development and maintenance practices, both at the technical level (e.g., more reviews, greater emphasis on design techniques) and management level (e.g., better resource estimation and tracking, more emphasis on project management).

The transition that is taking place is in its early stages. It is being headed by two forces. One is the needs of the larger projects in the Major Projects Division, and the other, the efforts initiated within MIS aimed at increasing management visibility and control over the growing MIS function.

2.0 DESCRIPTION OF SOFTWARE DEVELOPMENT ACTIVITIES

2.1 Overview of the Development Approach

There are MIS standards which address the phases and the activities of each phase for software development/maintenance projects. Three major phases are identified: definition, construction and implementation. Each of these phases is described below. This discussion focuses primarily on the practices of the System Development Division. An overview of practices of the Major Projects Division is discussed in Section 2.6, and the Technology Division in Section 2.7.

2.2 Phase Descriptions

2.2.1 Definition Phase

The definition phase includes determining the feasibility of a project, the actual requirements for the software (development, enhancement or fix) and certain aspects of design. This phase is initiated by a formal request for MIS services by a customer. When a request for service is received, an estimation of the amount of effort required for the definition phase is made. An agreement between MIS and the customer is made before any effort is expended.

The definition phase results in the preparation of a set of information (as described in the MIS standards) which generally includes a description of problem, the suggested solution and the cost of construction and implementation. This material is prepared by an analyst and reviewed first by MIS management and then with the customer who submitted the request. The review results in a decision on whether or not to proceed to the next phase of the project. The review addresses the adequacy of the solution and the cost and schedule of the next phases. This is the major and often the only review or checkpoint for a project.

The formality of this entire process differs with each group and customer and particularly with the size of the project.

2.2.2 Construction Phase

This phase includes detailed design, actual coding and testing. User and/or operating instructions are prepared during this phase. The completion of the phase is determined by the analyst, that is, when testing is completed and/or when the user has reviewed, and is satisfied with, the results. The form of the user acceptance varies widely from project to project.

During this phase, walkthroughs and peer group reviews are used at the design and code levels. This practice varies by analyst, group and project.

2.2.3 Implementation Phase

This phase, in the case of a software fix or enhancement, results in the software being turned over to operations and a new version of the system being used. (For special information requests/reports, there is no real implementation phase.) On some projects, a user test period is defined, at the end of which there is a formal acceptance. The use of the practice varied. Sometimes, when employed, the customer sign-off was perfunctory. Seldom was any type of formal acceptance testing performed.

2.2.4 Documentation Practices

Documentation is prepared in support of the internal and external review held at the completion of the definition phase. MIS standards outline the contents and address requirements, feasibility, high-level design, and cost and schedule for construction and implementation. The content, completeness, the lack of detail of the information, and the degree to which the MIS standards are followed varies widely.

The only other documentation that is formally produced is the user and operating instructions.

2.3 Quality Assurance Activities

Quality Assurance is not explicitly defined within this environment. Management assumes that appropriate steps are being taken at lower levels to develop and maintain quality products. Thus, the implicit responsibility for quality assurance is the analyst's.

QA is not directly addressed in any of the standards, procedures, or guidelines examined. Subjects which often fall under the heading of QA (such as reviews and coding standards) are covered, though. These standards are followed only to a limited degree. Programs do not exist to assist people in the use of the standards, nor to review or enforce their use.

2.4 Validation, Verification, and Testing Activities

There are two primary check or review points in the development phase: a review between the definition and construction phases and an acceptance review.

The review step between the definition and construction phases has two primary objectives. First, it is to review the requirements and high-level design of the software (and feasibility, if in question). Second, the estimated construction and implementation schedule and costs are reviewed. Approval results in the initiation of the construction phase. This review step involves MIS management staff and the customer. The customer's involvement varies from a separate external review being held to obtain customer approval, to a much less formal review and go ahead.

The acceptance review procedure is usually an informal one varying from the user reviewing the format and content of a report, to the use of a piece of production software for an agreed-upon period of time. The format, amount of preparation, and the formality of these reviews vary depending upon the size of project, its importance and/or criticality, and the project manager's or customer's preferences. More emphasis within the MIS environment is placed upon the first step, since it results in a decision regarding the continuation of the projects.

Essentially all testing is done by the development staff during the construction phase. Customer acceptance testing, if performed, is the conditional use of the system in production for an agreed-upon period. It appears as though little emphasis is placed upon user acceptance testing, or independent testing.

Other V,V&T techniques used include walkthroughs, peer group reviews and desk checking. These are performed primarily during coding, but also during design. The frequency and format of these sessions is dependent largely upon the individual analysts and somewhat upon what is encouraged by group leaders.

2.5 Configuration Management

The responsibility for managing changes to software and documentation is left primarily to the analyst performing the change.

2.6 Major Projects Division

The manager of this division along with the project manager of one of the three projects of this division was interviewed. The following discussion summarizes some of the practices used on the project examined in addition to or in lieu of those previously described.

The project examined, approximately a ten-year effort, was the development of an on-line order entry system. The project was nearing the end of the design phase, and some components were already being coded.

The development approach followed on the project is based upon the MIS standards (i.e., the phases outlined). In addition, the project manager has introduced several more formal check/review points into the development phase to facilitate visibility and user involvement. Intermediate products, (i.e., the requirements and the design) were being documented and reviewed at these points.

A formal and rigorous requirements-definition process was followed. To guide this effort, a user steering committee was established. The requirements were documented and agreed upon. All subsequent changes were also formally reviewed and agreed upon.

An internal (MIS) review of the requirements was held at the completion of the definition phase (according to procedure), but only after the external review with the steering committee was held. (Usually it is done in the reverse order.) On this project, the definition phase was restricted primarily to requirements definition and did not address a significant portion of the system design.

The same review process (external and MIS-internal) will be repeated for the review of the design. Plans for an independent audit of the project at its current stage have also been made.

2.7 Technology Division

The work done in this division is all assembly language programming in support of the data center, the operating system and user utilities. Most tasks undertaken were relatively small and initiated from within the division in response to a recognized need. Members of the customer or user community (i.e., users of the computer facilities, other MIS staff) are not normally involved directly in requirements definition or in review and acceptance of the finished capability.

In contrast to the Systems Development and Major Projects Divisions, the development approach followed by the Technology Division is very informal. A programming task is identified, assigned and completed. The practices employed are entirely defined by the individual analyst. The operation of this division is programming-task oriented, with the emphasis on completion.

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

The review between the definition and construction phases is the most formal and widely used V,V&T technique. It usually (on the larger projects) is supported by documentation and involved technical staff, management and the user.

On a technical level, walkthroughs, peer group reviews and desk checking techniques are used during the detailed design and coding.

The definition and continual use of the user steering committee on the projects in the Major Project Division is apparently very successful to date. It is by far the most significant involvement of the users in the V,V&T process at this site.

Internal audits (initiated and performed by corporate staff) are performed on production software. These focus upon operation and application-specific concerns (e.g., security and whether or not the system correctly performs or supports the defined procedures). This technique is to be employed during development on the project surveyed in the Major Project Division.

Within the Systems Development Division, the work request procedure is formalized. To a degree, this acts as a mechanism for documenting requirements and obtaining the user's (requestor's) concurrence.

Various techniques (e.g., representation schemes) are used to specify system designs. These schemes are employed on the larger projects.

3.2 Tools

On the project surveyed in the Major Projects Division, a data dictionary system is being used to define and track the usage of data elements throughout the system. It is being used for design definition as well as code construction, thus verifying the consistency between the design and the implementation.

A source-code management system is being widely used to help manage and maintain the programs. This is deemed essential in the primarily maintenance environment of the System Development Division.

3.3 Perceived Problem Areas

Software change controls or configuration management is one area that is identified as a problem area because of the lack of clearly defined responsiblities, procedures and supporting mechanisms. A specific problem identified is that documentation is not always updated along with the code, including the operating procedures. Informing users of changes is also done in an ad hoc manner.

Efforts to address this problem area and to understand, monitor and formalize the software change process have been initiated.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

There are four formal sets of documented standards and procedures. These are listed below with a brief description of their scope.

1. MIS Standards - project management responsibilities, software development procedures (i.e., phased development approach), project review

procedures, design methods and naming conventions,

2. Programming Standards - the system operating procedures and interfaces,
3. Procedures - management procedures and supporting forms (including MIS request for service),
4. Systems Policy - MIS operating policy, e.g., personnel policies.

The MIS standards deal primarily with software development standards and practices. These are followed to a degree in the Systems Development Division, form a basis for practices in the Major Projects Division, but have very little application and/or use in the Technology Division.

Staff attitudes vary towards the standards' utility and applicability. Selected portions are applicable and are used as a guideline. Use varies with the individual manager/team leader and analyst.

An effort within MIS to determine the use of the standards and also requirements for their updating had been initiated.

APPENDIX G

COMMERCIAL SITE 2 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site C2 is a large profit-oriented utility, with software activites directly supporting the corporation. These activites are basically within one corporate division (figure G1). Organizational structure within that division is based on specific projects. Standards and practices are currently set by the individual projects, with some guidelines set forth by the division. Corporate standards for specific data processing activities are being drafted. In terms of this site report, the user is anyone within the corporation that utilizes information delivered by a given system and the customer is one who purchases services from the corporation. Both project management and the customers have significant impact on software development activities.

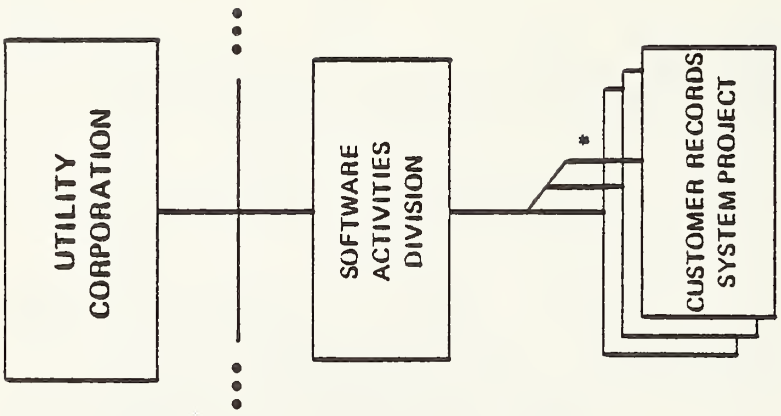
1.2 Description of Software Activites

We looked at one specific on-going project within Site C2's computing activites. This project has been a test bed for incorporating state-of-the-art technology, and, as such, has had high corporate visibility and will conceivably have a significant impact on the definition of corporate policy. Of particular significance to the study is the fact that this project utilizes an independent system test team and has found the results definitely cost-effective. This project also supports its own configuration management (CM) activities. Both the independent system test team and the CM group have been a formal part of the project since its inception 5 years ago.

The project examined was an information system responsible, in part, for generating billing information and handling questions from customers about their bills. At its peak, the project had 103 people working on it (17 management, 8 system test, 53 analyst/programmers, 10 support and 15 user representatives). Currently, project effort includes approximately 50 people and is evenly divided between development and maintenance activies. Systems software, including utilites and database systems, was purchased, while all applications software was developed internally. Programming is done in COBOL for a large main frame with mass storage capacity.

1.3 Factors Influencing the Environment

Several factors influenced the development of this software project. The large volume of data increased recovery considerations, making system reliability and maintainability critical. The database orientation of the system and the large number of users made configuration management mandatory. Since the system supports the generation of corporate income and is part of the customer interface, it was imperative that accuracy requirements and failure tolerances be met. The system has a very long projected lifecycle; so anything that could ease maintenance and transferability/reusuability costs would be quite cost-effective. Management felt that these factors, common to



*Interviewed for Survey

Figure G-1. Site C2 Organizational Structure.

SITE SUMMARY
C2

General Characteristics		
Group Name	:	Customer Records System Project
Type of Activity	:	Customer services and support automates major customer functions: orders, operations, billing, etc.
Application Area	:	Operations
Languages	:	COBOL
Project Size Range	:	Small: less than 6 months; Large: more than 24 months
Development/Maintenance Technology		
Development, Maintenance or Other	:	Maintenance & Enhancement (50% - 50%)
Documented or Informal Approach	:	Informal, some standards have evolved & documented
Number of Phases	:	5 : Feasibility/Definition, Design, Development, Test, Implementation
Use of Formal Techniques/Tools	:	Configuration Management tools were developed
V&V Technology		
Requirements as a Baseline	:	Documented in definition phases, but not used as basis for V&V
Other Levels of Specification	:	Design
V&V Within/Between Early Phases	:	Informal, ad hoc
Testing Practices	:	4 levels: Unit, Integration, Functional Validation, Acceptance
Reviews	:	Formal reviews during testing & for system enhancements
Techniques/Tools	:	COBOL Debugger
Documentation Practices		
Formal/Informal Guidelines	:	Formal guidelines
Adherence to Guidelines	:	Followed rigorously
Enforcement/Review	:	Reviews held as scheduled

many software projects, were justification for incurring any additional expenses it might take to develop the software.

There were several unique factors that influenced the success of this project. The project was large enough to support an independent system test team. Senior management in this organization has a high degree of technical awareness and has actively supported the project with time (e.g., project schedules included complete reviews and testing), personnel (e.g., independent test team) and money (e.g., machine test time, personnel). Site C2 also has an excellent training program, which includes both technical and procedural classes.

1.4 Historical Perspective / Evolution

When this project was first started, no formal standards or procedures were identified. As the development effort progressed, various standards, techniques and tools were introduced. This particular software system has undergone several releases to date (more are planned), so there has been time for incomplete or inconsistent procedures to have been revised. An example would be the evolution of the phased-approach standard, a division-level instrument that has become increasingly more specific over the life of the subject project.

2.0 DESCRIPTION OF SOFTWARE ACTIVITIES

2.1 Overview of Development Approach

The software development approach was formalized on a project level and consists of 5 phases: feasibility/problem definition, design, development, test and implementation. Division-level guidelines provided a framework within which other standards and procedures were introduced and evolved throughout the project lifecycle. In several instances, a standard was introduced after the activities it was to govern had been started. Such introductions were not always successful.

2.2 Phase Descriptions

2.2.1 Feasibility Phase

The feasibility (problem definition) phase for this particular project included retroactively documenting existing second-generation systems, which resulted in a functional-requirements document for the new system. The initial documentation activities encompassed the information flow and process of a very large corporate system. During the feasibility phase, it became necessary to narrow the scope of the initial proposed software activities to certain major functions. Much effort was expended in this analysis phase that was not directly applicable to the subject project. Various user-group members, systems designers, senior analyst/programmers and system test team members participated. Management signoff was required for completion of this effort. Informal reviews were held with user representatives. These reviews included the system-test-team members as a courtesy. A high level of interaction with the customer has been cited as one reason for the success of this project.

2.2.2 Design Phase

The design phase resulted in three products: a data requirements document, a system specification and a database specification. Representatives from the user group, systems designers, analyst/programmers and system test team participated. The completion of the above documents signified the completion of the design phase. A formal preliminary design review was held with users and operation personnel present. Informal detailed design reviews were also held. PDL (Program Design Language) saw limited use during this phase, but was judged only partially successful (see Section 3.3).

2.2.3 Development Phase

The development phase resulted in program specifications and code. Analyst/programmers and some management were involved. Management signoff was required for completion of this activity. Formal and informal code walkthroughs were held often and were rated as moderately to highly successful. Standards (e.g., naming conventions, module size) were used with moderate effectiveness. Enforcement of these standards was informal (e.g., incorrectly named variables were unacceptable to the data dictionary). Very few code analysis tools were available. Dynamic analysis tools were used for performance evaluation but not to establish the degree of testing.

2.2.4 Test Phase

The test phase was the primary responsibility of the independent system test team. Products of this phase included test plans and test-analysis reports. Four levels of testing were performed. Unit testing was done solely by the developers. Integration test was done by the developers, but monitored by the system test team. Functional validation testing was done by the system test team alone. Acceptance testing ("dress rehearsal") was conducted by the system team with the involvement of real users. Acceptance testing was described as "very visible and very valuable." This phase was looked upon favorably by the development staff because the test team was taking some of the pressure off them to ensure the correctness of the system. Test support facilities included some automated tools that were rated very effective.

2.2.5 Implementation Phase

At this site, the implementation phase refers to getting the system up and running (not translating the design into source code). All parties were involved in this phase. Products included user manuals, operations manuals and the system itself. An informal review was held to verify that completion criteria had been met.

2.3 Quality Assurance Activities

As previously stated, this project supported an independent system test team responsible for insuring the development of a quality system. It was important that there was confidence in the reliability and integrity of the system because it generates corporate income and supports customer interface. There were no procedures, standards, or guidelines to support the system test group at the beginning of the project. Members of this group participated in each of the lifecycle phases as described above.

2.4 Validation, Verification, and Testing Activities

A variety of formal and informal reviews were held during the feasibility and design phases (see Section 2.2). Formal code walkthroughs and unit testing were performed in the development phase. The system test team performed integration testing and also acceptance testing. No formal procedures, techniques or tools had been designated at the beginning of this project.

2.5 Configuration Management

The CM function was considered critical because of the projected lifecycle of this system and the wide use it would have within the corporation. The basic objective was to make sure nothing interfered with an accurate, reliable and available system. A project-specific CM charter was established at the beginning of the development activity. This charter was the basis for the CM process and the particular mechanisms that evolved to support it. The CM function was judged a key to the success of the project. CM activities were conservatively estimated to take 15% of the total project effort.

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

As mentioned in Section 2.2.4, the subject project utilized a comprehensive testing approach which has been contributing significantly to the success of the project. The four levels of testing (unit, integration, functional validation and acceptance) were described earlier. Site C2 spent considerable effort in the latter two levels of testing. During functional validation testing, a complex test database was created. Also during this phase, the system test team performed an on-site test in parallel with the production system (the same data used to generate bills were used as test data, with the test results being carefully compared with the "real" results). Acceptance testing consisted of two major activities, dress rehearsal and a pilot operation. During dress rehearsal, a test office was set up. Real users and operation personnel worked with the system test team utilizing real data. The pilot operation consisted of putting the new system into full operation in a live, but limited, situation. The active inclusion of the users in this final phase of testing was a major investment in terms of corporate resources and was responsible for the acceptance of the new system and the current goodwill between users and system developers necessary for continued system evolution.

Formal preliminary-design reviews, informal detailed-design reviews and code walkthroughs were utilized throughout this project. These techniques were deemed cost-effective and will be used in other projects. No formal procedures for performing these reviews existed at the beginning of this project, but are currently being formulated for possible corporate-wide use.

As mentioned earlier, this project is supported by a fully-defined, formal configuration management (change control) function. This activity met with full management support because of the cost-effective results produced. Several in-house tools were created to support the CM function. Of particular interest was an automated trouble-reporting scheme utilizing a database of all requested changes.

3.2 Tools

Dynamic-analysis tools were utilized by this project, but the primary use was for performance evaluation, rather than to measure the thoroughness of testing. A COBOL interactive debugging capability (including snapshot and tracing options) was available, but there were no requirements or guidelines for its use. Limited static-analysis information was available from the compilers, but, again, there were no requirements or guidelines for its use.

Project personnel were enthusiastic about various automated test-generation capabilities available and relied heavily on their support. Of particular note were two internal products for the generation of production and test job control language and also a commercial product for simulating on-line transactions.

3.3 Perceived Problem Areas

The retroactive introduction of various standards and procedures caused some problems during the original development activity. Personnel were unsure what was expected of them, and what enforcement procedures were to be followed. This was particularly evident during the initial design phase.

A Programming Design Language (PDL) was tried but some problems occurred. Reasons cited for non-success were: 1) PDL was introduced after the start of the project; 2) technical staff complained of having to do their job three times (once in English, once in PDL and once in code); 3) there was no comprehensive training program or standards and procedures to aid individuals in fully utilizing this tool.

Now that project standards have evolved, personnel are interested in making enforcement more cost-effective. Code auditors and other types of standards checkers are being explored.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

Site C2 currently has several organizational levels of standards and guidelines. Corporate policy (the highest level) currently does not deal with data processing and, specifically, V,V&T). Division-level policy is oriented specifically to data processing but only addresses very broad methodological and operational concerns. It is at the project level that specific standards, guidelines and practices are directed.

This project had formal standards for the design phase, with sign-off by the users a necessary completion criteria. Program design was performed to specific standards and review. Coding standards included such things as naming conventions, module size and structured programming. These standards met with varying degrees of success and were modified accordingly in order to be responsive to the continuing development effort.

APPENDIX H

COMMERCIAL SITE 3 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site C3 is a large financial institution, with software activities primarily supporting the institution itself. These activities are basically within one corporate division (figure H-1). Organizational structure within that division is based on specific corporate activities. Standards and practices are currently set by the EDP Division. In terms of this site report, the user is anyone within the corporation that utilizes information delivered by a given system and a customer is anyone who buys services (not necessarily software) from the institution. Division management has a significant impact on software development activities.

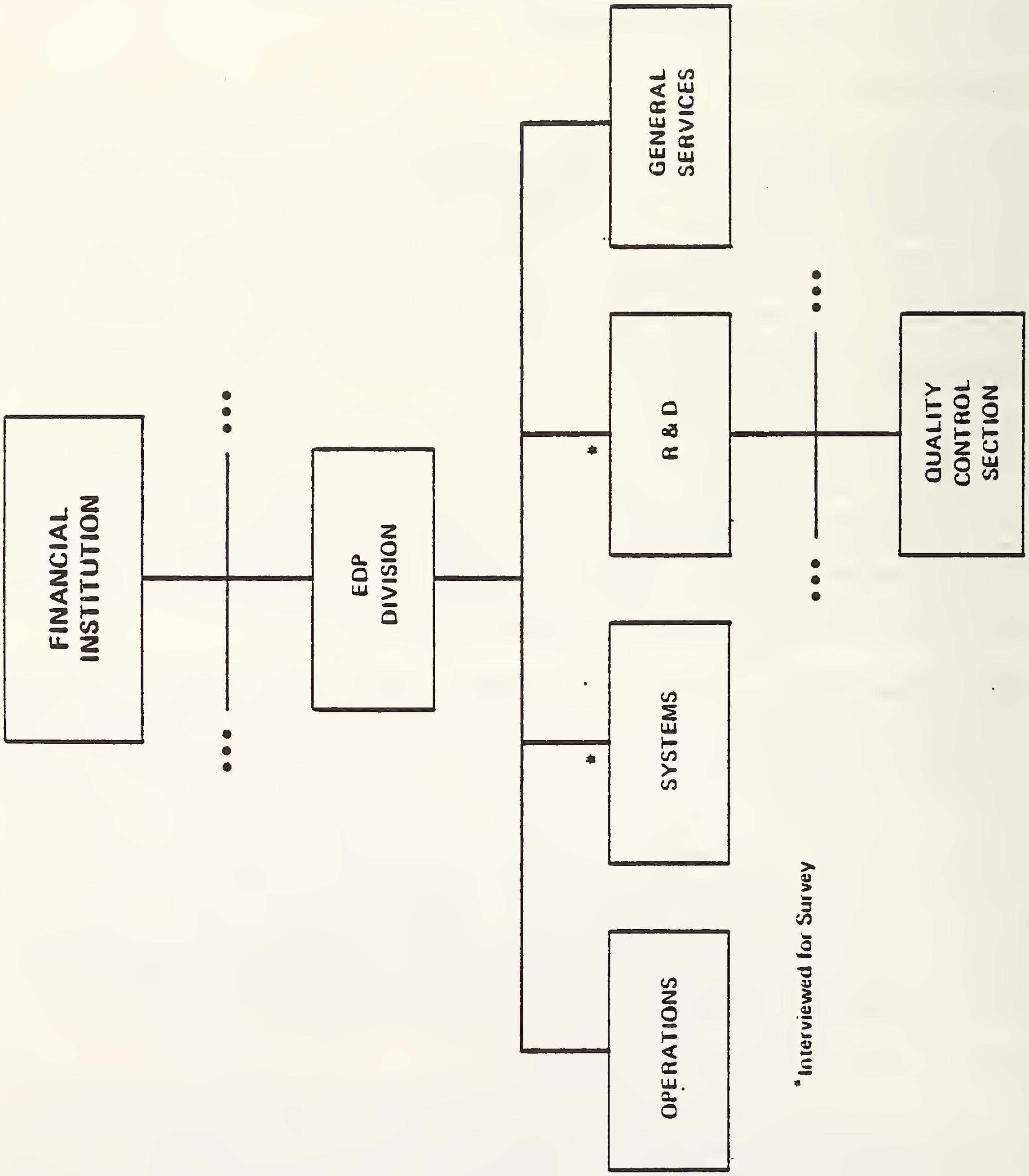
1.2 Description of Software Activities

Site C3's software development activities are concentrated within the EDP Division. The Division consists of four Departments (Systems, Operations, R&D and General Services). Representatives from Systems and R&D were interviewed. Systems is responsible for software design and development. R&D is responsible for a variety of functions including the independent test team (Quality Control) and Configuration Management activities.

Most software development is done in COBOL. Site C3 hardware includes multiple large mainframes (where 80% of the software runs) and multiple mini systems. Approximately 12% of the active software has been acquired from outside (with schedule urgency cited as the primary reason for the acquisition). New system development is about 20% of the division's current activities, enhancement work accounts for another 60% and error correction/maintenance, the remaining 20%. Software activities are classified as: 1) emergency (requiring 8 or less hours of effort); 2) small (less than 1 month); 3) medium (1-12 months); 4) large (more than 12 months). Approximately 45% of recent activities were rated as small, 35% as medium and 20% as large. Financial and management systems are the primary applications, followed by information processing systems such as inventory and personnel.

1.3 Factors Influencing the Environment

Several factors influence the development of software in this environment. Since most systems support, in some way, the generation of corporate income, it is imperative that accuracy requirements (the software quality "correctness") and failure tolerances (the software qualities "reliability" and "integrity") be met. Because these systems support customer interface, maintainability also becomes an important quality. Flexibility is critical because the subject institution is highly regulated by various governmental agencies (both Federal and State).



*Interviewed for Survey

Figure H-1. Site C3 Organizational Survey.

SITE SUMMARY

C3

General Characteristics	
Group Name	Systems & R&D departments of the EDP Division
Type of Activity	Total computer support for Financial institution
Application Area	Management, administration & operations
Languages	COBOL
Project Size Range	Small: 1 month or less; Large: 12 months or more
Development/Maintenance Technology	
Development, Maintenance or Other	20% development, 60% enhancement, 20% maintenance
Documented or Informal Approach	Commercial Development Methodology
Number of Phases	3 : Definition, Design, Implementation
Use of Formal Techniques/Tools	Techniques defined as part of methodology are rigorously followed
V&V Technology	
Requirements as a Baseline	Defined in Definition phase
Other Levels of Specification	Design
V&V Within/Between Early Phases	Spread across entire lifecycle through reviews
Testing Practices	Testing is considered throughout design and performed at many levels
Reviews	Ten separate reviews strictly followed
Techniques/Tools	No tools
Documentation Practices	
Formal/Informal Guidelines	Formal guidelines as part of methodology
Adherence to Guidelines	Adherence
Enforcement/Review	Enforced through reviews

Two unique factors were identified in terms of this specific environment. One factor is that distributed computing and multiple sites running the same version of software is the normal mode of operation for this institution. This makes configuration management and rigorous software development and maintenance activities extremely important. The other unique factor is that division level management has actively supported the adoption of a common software lifecycle methodology to be used by the entire division (100% of internal software development).

1.4 Historical Perspective/Evolution

Site C3 acquired and adopted a comprehensive commercial software development methodology about one year ago. Previous to that time, informal requirements reviews, design reviews, product reviews and audits were held often, while code reviews, test reviews and configuration management activities were seldom performed. Now all reviews and configuration management activities are performed in a formal and frequent manner. Evaluation and revision of the newly adopted methodology is on-going. Management feels that the overhead associated with these activities is a definable burden and that the results are worth it. Both user and management confidence in newly developed (or modified) software has noticeably increased because of this effort.

2.0 DESCRIPTION OF SOFTWARE ACTIVITIES

2.1 Overview of Development Approach

The software development approach adopted as a division standard by Site C3 is a commercially available methodology. It consists of three phases: systems definition, systems design and systems implementation. Quality reviews are a formal and integral part of the methodology.

2.2 Phase Descriptions

2.2.1 Systems Definition

The systems definition phase includes 58 separate tasks, covering project planning, feasibility studies, and user-requirements specification. Three quality reviews are also specified. Representatives from quality control attend all reviews, as do user representatives. These formal reviews have been rated highly effective. The methodology specifies system definition standards that are rated moderately effective. Guidelines exist for using only subsets of the 58 tasks for various size projects or maintenance/enhancement activities.

2.2.2 Systems Design

The systems design phase consists of preliminary design, detail design, program design and programming and testing, which are covered by 95 separate tasks. Quality reviews are held for each of the above four subphases. These reviews are attended by representatives from quality control and the user community, and have been rated highly effective. (Code walkthroughs are attended by project and independent technical staff only; they, too, are rated highly effective.) The methodology specifies systems design standards that are rated moderately effective. A dynamic analysis tool had just been introduced to support this phase, but otherwise, there are no tools. Again, guidelines exist for using only subsets of the 95 tasks.

2.2.3 Systems Implementation

The systems implementation phase includes implementation planning, system test, operations turnover and acceptance/wrap up. Specified are 96 tasks, 3 quality reviews and a formal user acceptance procedure. User representatives and quality control are the primary participants in this phase. Again, the lifecycle methodology specifies standards to be followed in this phase. These standards are rated moderately effective. Some test support facilities exist to support this phase, with reported moderate effectiveness. Guidelines for exceptions also exist in this phase.

2.3 Quality Assurance Activities

Quality assurance activities are being performed by the Quality Control Section of R&D (independent of the Systems Department). These activities include participation in the reviews as identified above. Quality control is also responsible for certification of execution procedures and production software. Currently, new standards and procedures are being developed to support an expanded quality assurance charter for software development. These standards and procedures will be integrated into those of the commercially developed methodology. In addition to addressing new software, the standards and procedures will also address issues of modification and enhancement of existing software.

2.4 Validation, Verification, and Testing Activities

Validation, verification, and testing activities are not localized in a single organizational entity, but are performed by various project members. These activities are specified within the lifecycle methodology and include participation in the ten quality reviews mentioned previously (see Section 2.2). In that Site C3 follows a very disciplined approach to software development, V,V&T is an integral part of the lifecycle activities.

2.5 Configuration Management

The CM function is performed by the R&D Department. It is not project specific. Configuration management is highly critical in that tight control and security be maintained over production runs. Since Site C3 has a 24-hour shop, program fixes must be implemented in a timely fashion. To support this need, a complete set of procedures exist for all configuration management activities. The CM group currently handles about 30-50 requests for changes each month.

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

The primary technique utilized by Site C3 is the comprehensive software lifecycle methodology. This has been discussed thoroughly in Section 2.

Site C3 is also currently expanding the role of Quality Control in supporting the previously described lifecycle methodology (see Section 2.3).

3.2 Tools

No tools currently exist at Site C3 to support the commercial lifecycle methodology, but work is underway to acquire or build some.

A comparator tool exists and is used for ensuring control over programs entered into the production files. The primary emphasis of this language independent tool is production program integrity. This comparator has been in use less than one year and is used whenever a program modification has been made. No cost benefit analysis has been performed.

3.3 Perceived Problem Areas

The lack of automated support for the commercial lifecycle methodology is seen as the primary problem area and is currently being addressed.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

Site C3 has adopted a comprehensive commercial software development lifecycle and associated standards and guidelines. Separate tasks (249) are defined in the standards (Section 2.2), and guidelines are presented for using a subset of those tasks based on a specific project. Ten separate quality reviews are identified with agendas and participants itemized.

The above pertains primarily to the Systems Department. Both R&D and Systems also have departmental guidelines, most of which deal with software activities in an organizational sense (rather than technical).

APPENDIX I

COMMERCIAL SITE 4 SUMMARY REPORT

1.1 Organizational Overview

Site C4 is a large health care organization concerned with all aspects of hospital and clinic administration and maintenance. Computing services are provided by a distinct operating organization (Information Services), which reports to a high level of the management structure (figure I-1). This organization has sole responsibility for all aspects of computing support. This report will focus on the Systems Development Group within the Information Services Division. This group is responsible for all development and maintenance of application software.

The Systems Development Group depends on close coordination and communication with the user community. The group (approximately 15 analysts and 6 management leads) is divided into five subgroups parallel to its major user constituencies: financial administration, health plan services, patient records, clinical labs, and pharmacy. This organization creates a well defined user/developer interface.

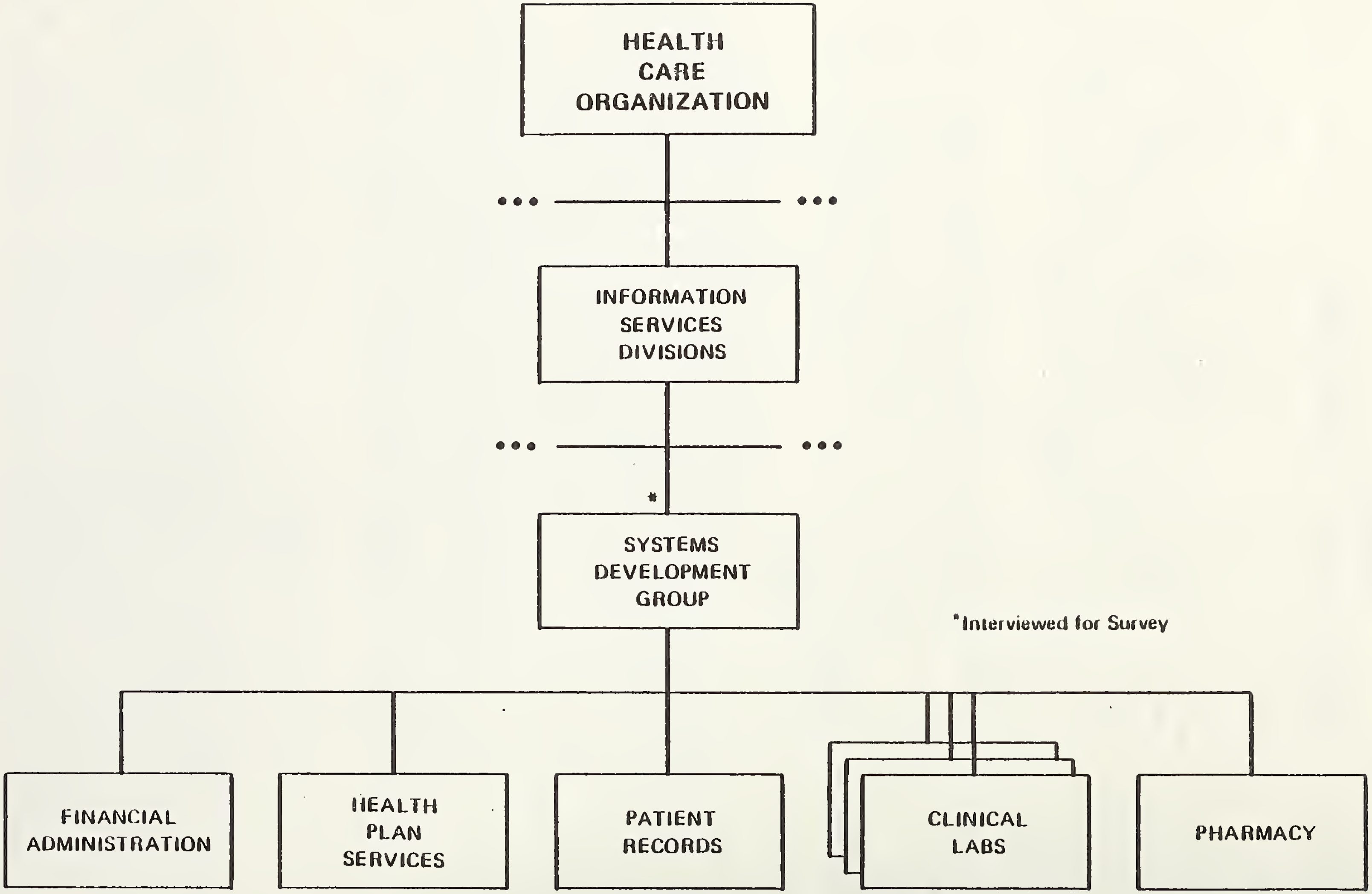
1.2 Description of Software Activity

Site C4's software activities consist of approximately 70% new development and 30% maintenance (including both corrections and enhancements). All software is programmed in COBOL and fits under the general heading of business systems. Software projects are predominantly small to medium (1 to 18 months of effort). The majority of the computer applications utilize a large mainframe system with remote terminals. The Systems Development staff is located at a central facility and the users are at remote hospital and clinic locations.

The management of the Information Services Division advocates the acquisition of software rather than in-house development when appropriate. This approach is advocated for several reasons, including the reliability of proven products, lack of resources for internal development, and the cost of acquisition versus development. Requests are analyzed and the market is surveyed to locate tools and services available and appropriate to the specified need.

1.3 Factors Influencing the Environment

The largest single factor which is affecting the activities and operation of the Systems Development Group is the acquisition and introduction of a commercially available software development methodology. The methodology defines a set of phases, the activities, and products of each phase, the status and product reviews to be held, and the participants and roles of those involved in the activities. The methodology follows a check-list approach and adds discipline and rigor to development and maintenance activities. It emphasizes planning and user involvement. The introduction and use of this methodology is fully backed and supported by management and a large number of the technical staff.



*Interviewed for Survey

Figure I-1. Site C4 Organizational Structure.

SITE SUMMARY
C4

General Characteristics	
Group Name	Systems Development Group (within Information Services Division)
Type of Activity	Complete computing support for health care organization (e.g., financial, insurance, patient records, labs & pharmacy)
Application Area	Administration, Management & Operations Support
Languages	COBOL
Project Size Range	Small: 6 month or less; Large: 18 months or more
Development/Maintenance Technology	
Development, Maintenance or Other	70% development, 30% maintenance
Documented or Informal Approach	Formal commercial methodology (emphasis is on management, not technical issues)
Number of Phases	3 : Definition, Design, Implementation
Use of Formal Techniques/Tools	Only an interactive debug tool
V&V Technology	
Requirements as a Baseline	Defined in definition phase
Other Levels of Specification	Design
V&V Within/Between Early Phases	No formal V&V, only practiced at discretion of individual analyst
Testing Practices	Informal
Reviews	Reviews at completion of each sub-phase
Techniques/Tools	None
Documentation Practices	
Formal/Informal Guidelines	Formal guidelines - commercial methodology
Adherence to Guidelines	Followed well
Enforcement/Review	Reviews after each sub-phase

A second major factor affecting the environment is the customer interface. Systems Development has a very close working relationship with its customers, facilitated by a software organization structure which parallels the distinct customer groups.

The environment is also significantly affected by the existence of multiple locations,introducing coordination and logistics requirements.

1.4 Historical Perspective/Evolution

Site C4 is an example of a software group emerging as a fully defined organizational entity. Previously software activities have been performed in response to the immediate customer need. There was little long-range planning and no central focus of the activities in the Information Services Division. An internal commitment has been made to better utilize the computer and associated resources via improved project management and customer interface practices. The methodology being adopted stresses planning, estimating, requirements and design specification, and formal project reviews. This creates a basis for overall planning and assists in the prioritization of service requests.

2.0 DESCRIPTION OF SOFTWARE ACTIVITIES

2.1 Overview of Development Approach

Site C4's software development approach is being dramatically affected by the current transition to a well-defined, commercial methodology. The transition was initiated approximately four months prior to the survey. The old process was guided by high-level division standards subject to interpretation by project managers. The new methodology specifies the details of software development and management and is supported by documented guidelines, procedures, self-explanatory checklists and forms, and personnel training.

2.2 Phase Descriptions

The old development approach was basic definition, design and implementation. The old methodology initiated a job with the specification of requirements through a request for services. The design and implementation phases were very informal. The analyst's informal interface with the customer was very important in assuring proper development and implementation. Adherence to standards, use of reviews or walkthroughs, and any testing practices followed were at the discretion of the project leader.

The new methodology, as sponsored by management through purchase and implementation of a commercial product, uses a similar three-phase structure but is expanded into eleven definitive steps (see individual phase descriptions). This methodology enforces a structured approach to the entire software cycle and includes intermediate products and technical reviews. There are several points where detailed planning and cost-estimate reviews are required. This newly adopted methodology requires continual close management and customer interaction, increasing visibility and control.

2.2.1 Definition Phase

In the old development methodology, the customer/user submitted a request that informally specified the software requirements. The software organization performed a feasibility study and formulated a cost estimate. These items were then reviewed by management for a job initiation decision.

The new approach performs the same functions but introduces more formality. The definition phase is divided into four subphases:

- Project Proposal/Project Plan
- User Requirements
- Systems Definition
- Advisability

Each subphase is subjected to technical review. The schedule and dollar estimates are projected after the first subphase, revised after the last subphase and also submitted to management review.

2.2.2 Design Phase

Previously, methods utilized in the design phase were very informal. The job was assigned to an analyst, who started development of the program. The amount of design and documentation was dependent upon the analyst's judgment and the project leader's requirements. Because of the lack of formal specifications and procedures, redirection would often result from customer interaction.

The new design phase procedures introduce the following subphases:

- Preliminary Design
- Detail Design
- Program Design
- Programming/Testing

These subphases define points for continued technical and management reviews. There are also points for cost and schedule review and/or revision.

2.2.3 Implementation Phase

The old procedures followed in this phase were informal. The responsibility for final review and acceptance of the product was the user's. Some user and system-operations documentation was produced. Quality assurance and validation were informal responsibilities of the project manager.

The new methodology identifies several subphases of the implementation phase. All subphases are followed, regardless of the size of the project. The degree of detail followed in each subphase can vary. The subphases specified are:

- Implementation Planning
- System Test
- Operations Turnover

- Start-up/Training
- Acceptance/Wrap-up

Each of the first three subphases requires a technical review. There is a review/revision of the final costs and schedule estimates after the first subphase. The final acceptance involves the customer/user.

2.3 Quality Assurance Activities

Previously, the quality assurance function was not specified and assumed to be a responsibility of the project manager. The methodology being adopted formalizes this function through an extensive set of management and technical reviews with significant user involvement.

2.4 Validation, Verification, and Testing Activities

In both the old and the new methodology, the application of specific V,V&T techniques is at the discretion of the analyst, project manager and/or the customer. There are few specified procedures or requirements for independent audits or testing reports. Methods used have included desk checking, and, when appropriate, application of a tool for test tracing.

2.5 Configuration Management

Configuration management and change control is delegated to the software development and maintenance personnel. The programmer responsible for a software system maintains the program configuration, responds to problem reports and makes the first judgment as to whether a request is too large for routine maintenance. (A request for a modification outside the scope of maintenance would be submitted as a formal service request.)

3.0 TECHNIQUES AND TOOLS

3.1 Techniques

Many review points are specified by the methodology. The specific analytical techniques to be utilized in preparation for and during the reviews are not specified.

3.2 Tools

The only tool identified was an interactive test-data tracing tool which permits snapshots of specified memory areas. This tool was rated as very effective in debugging support of online applications. Use of this tool was strictly ad hoc.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

The adopted commercial methodology provides considerable guidance in terms of a standard framework and operating procedures. Adherence to the methodology is being encouraged and supported by management. Customer roles and responsibilities are also evolving as a result of this methodology.

The methodology will greatly assist in standarizing the development approach. It provides detailed guidance on phases, products and reviews to be held. However, it does not provide detailed technical guidance on specific methods, tools, or techniques. This guidance will have to be provided by additional training and internal development of procedures to support the commercial methodology.

APPENDIX J

COMMERCIAL SITE 5 SUMMARY REPORT

1.0 GENERAL CHARACTERISTICS OF ENVIRONMENT

1.1 Organizational Overview

Site C5 is a large multi-faceted corporation, including a centralized computing services division which supports the other divisions of the corporation and performs contracts for external customers. The group surveyed, the Business Systems Support Group, (part of the computer services division) works in support of an engineering oriented division of the corporation (figure J-1). The group is organized into seven sub-groups with the manager of each of the sub-groups reporting directly to the group manager. The seven sub-groups are organized according to the types of systems being supported/developed and include project logistics, manufacturing quality assurance, engineering materials, and systems integration. Projects from the first three and the overall activities of the groups are the subjects of the survey and interviews. The group consists of 83 staff, 8 management and 75 analysts.

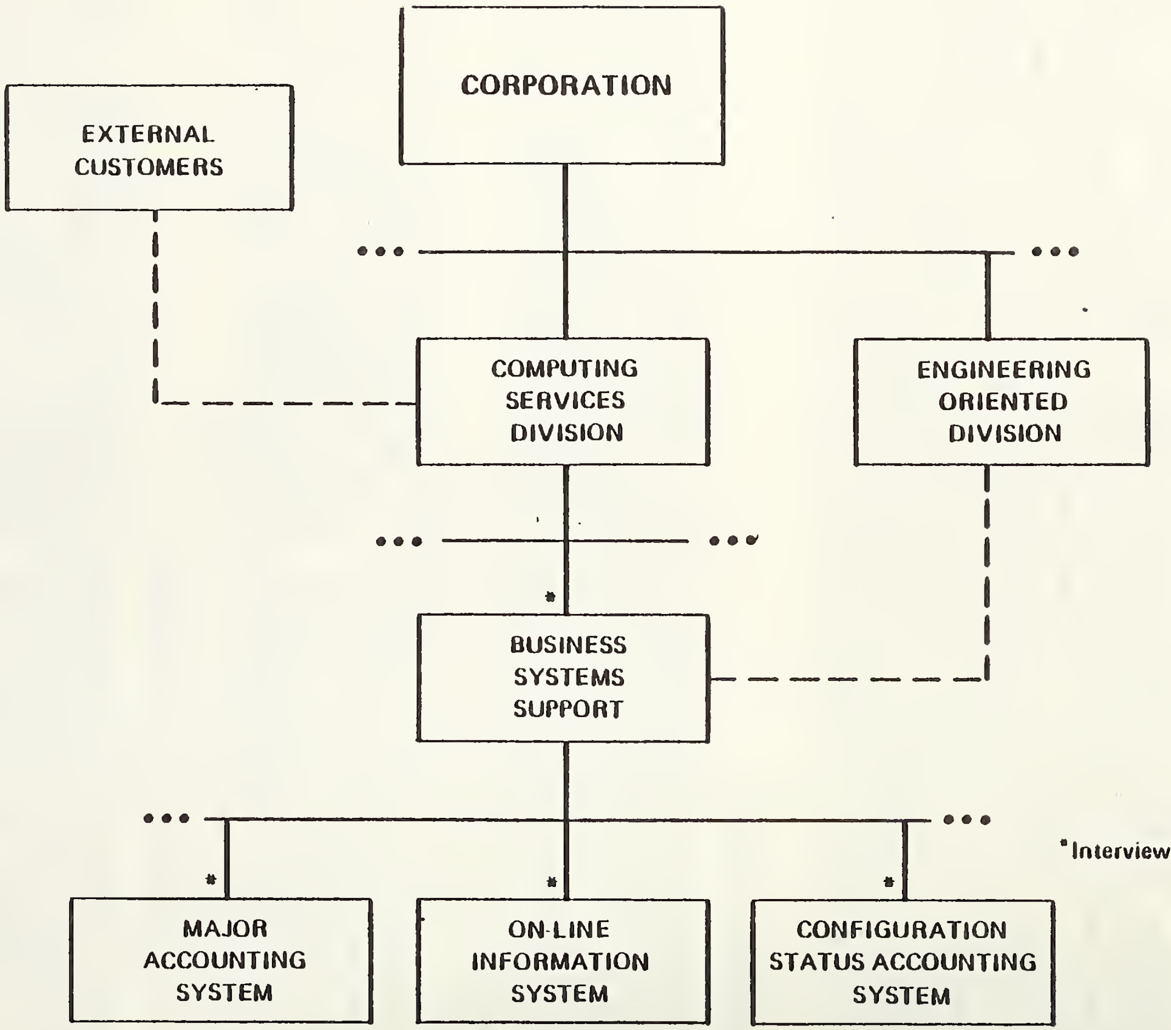
1.2 Software Activities

This group is primarily a maintenance group, sustaining and enhancing the business and operations oriented systems for the customer (i.e., the division of the corporation). The primary language used is COBOL. In general approximately 70% (25 of the 37) of the current development/enhancement projects are considered to be small - up to 12 man-months employing 1 to 3 people. In total, the group is presently working in support of more than 180 accounts representing major systems/subsystems.

There are three specific projects discussed as part of the survey. One of the projects discussed during the interview is the on-going support of the major accounting system used by the customer. The other two projects are new development activities - one is an on-line information system used for tracking the quality assurance function on all end-items produced by the customer; the second is a configuration status accounting system for a larger engineering/manufacturing project of the customer.

1.3 Factors Influencing the Environment

There are three factors that significantly influence the operations within this environment. The first relates directly to the customer. Both formal and informal communication channels are established. Though it varies from project to project, in general the customer is significantly involved throughout the duration of the projects, particularly during the requirements specification phase. On one project, a customer representative is located on site part time to facilitate communication and involvement.


CORPORATION
EXTERNAL CUSTOMERS
COMPUTING SERVICES DIVISION
ENGINEERING ORIENTED DIVISION
*
BUSINESS SYSTEMS SUPPORT
*
MAJOR ACCOUNTING SYSTEM
*
ON-LINE INFORMATION SYSTEM
*
CONFIGURATION STATUS ACCOUNTING SYSTEM

*Interviewed for Survey

Figure J-1. Site C5 Organizational Structure.

SITE SUMMARY
C5

General Characteristics		
Group Name	:	Business Systems Support
Type of Activity	:	Computer Support for Engineering Division of company
Application Area	:	Business: administration and operations support
Languages	:	COBOL
Project Size Range	:	Small: 12 month or less; Large: 36 months or more
Development/Maintenance Technology		
Development, Maintenance or Other	:	Primarily Maintenance/Enhancements, some Development
Documented or Informal Approach	:	Formal approach including commercial design methodology
Number of Phases	:	7 : Requirements, Preliminary Design, Detailed Design, S/W Construction, S/W Certification, Installation, Maintenance
Use of Formal Techniques/Tools	:	Inspections, walkthroughs and Nassi-Shneiderman charts, debug aids
V&V Technology		
Requirements as a Baseline	:	Formally specified
Other Levels of Specification	:	Data Flow, Functional, Program, Data, decision logic
V&V Within/Between Early Phases	:	V&V practices throughout lifecycle: walkthroughs, reviews, trace back to requirements
Testing Practices	:	Formal and extensive, test data generation
Reviews	:	Extensive phase-by-phase
Techniques/Tools	:	Debug aids, specification tools (Nassi-Shneiderman charts), Decision tables, data dictionary
Documentation Practices		
Formal/informal Guidelines	:	Formal guidelines (company wide)
Adherence to Guidelines	:	Closely followed
Enforcement/Review	:	Reviews throughout phases

A second factor to be noted is the involvement and support given to project and technical staff by management. This support takes the form of assistance in external interfaces with customers, internal review and management assistance, and the use of new practices, techniques, and tools.

The third factor observed relates to staff and the technologies employed. A progressive attitude seems to pervade the environment. The practices employed on the projects surveyed are rigorously and formally followed. There is a well defined, phased approach followed. This is combined with the use of a commercial design methodology and various automated tools.

1.4 Historical Perspective/Evolution

The group surveyed benefits from the advantages of being a part of a large computer services organization. The organization has in the last several years been developing, enhancing and offering training for a project methodology. The principles and fundamentals of this methodology are generally followed throughout the computer services division. Experiences with this methodology, and supporting techniques and tools are being accrued and, as the resulting knowledge is spread, all personnel within the division benefit to a degree. This effect is certainly apparent within the group surveyed. Moreover, this group in some respects is leading the technology advancement being exhibited within the computing division through experimentation with certain techniques and tools.

2.0 DESCRIPTION OF SOFTWARE DEVELOPMENT ACTIVITIES

2.1 Overview of Software Development Approach

The phases and activities of the software development and maintenance approach are defined by a set of company standards. These standards define 7 phases:

1. Requirements definition and analysis,
2. Preliminary design,
3. Detail design,
4. Software construction,
5. Software certification,
6. Installation,
7. Maintenance.

These standards outline the objectives of the documentation to be produced and the reviews to be held for each phase. There also exists a set of management guidelines to assist in project definition, organization, planning, administration, evaluation and control, and termination.

Some of the projects being performed, because they are Federally funded, are following DoD standards (primarily the documentation standards). These standards define 3 phases: initiation, development and operations, with the development phase further divided into definition, design, coding and testing.

2.2 Phase Definition

The following descriptions of the phases focus primarily upon one of the three projects surveyed, an on-line end item tracking and quality assurance system. Where practices differed significantly or there is information to be added concerning either or both of the other projects, an explanation of this will be included.

This project is a large, new development project. There are seven major subsystems which have been designed. Currently, two of the seven are being coded and will be implemented. A rough estimate of the effort required for coding and implementation of these two subsystems is 20 man-years.

2.2.1 Requirements Definition and Analysis

The definition of the system requirements was performed by the customer in conjunction with project staff. The primary role of the project staff was one of technical assistance, review and analysis. The effort required approximately seven man-years, resulting in a documented requirements specification for the entire system (i.e., all seven subsystems). This specification underwent a formal review and sign-off before the next phase was begun. Now, the requirements are under formal change control, so that the impact of each change can be estimated, and the progress of the implementation of requested and approved requirement changes can be tracked.

The maintenance project surveyed (i.e., the project accounting system project) has a formal user interface and requirements specification scheme deserving of mention. The specification for system changes/enhancements was developed by the user/customer and documented in the form of decision tables. These specified the condition governing a set of actions and the data items to be tested in the conditional test. These items are system variables that are formally documented in the system data dictionary. These specifications were prepared by the customer's analyst; submitted to the project staff; analyzed and agreed upon; then, implementation would begin.

2.2.2 Software Design

The software design was accomplished in two phases, a preliminary design phase and a detailed design phase. In combination, this effort was approximately 6 man-years. Each of these was completed for the entire system and is described below.

The preliminary design phase resulted in a high-level function-oriented description of major system modules. A commercial design methodology was employed which includes the documentation and analysis of the system data flows in the process of developing the functional design. The requirements were used as a base line and the elements of the design were explicitly traced back to the associated requirements.

Formal walkthroughs were held and the final preliminary documentation was reviewed and formally accepted by the customer. Status reviews to determine progress toward achieving milestones were also held. Adherence to design standards was also enforced in conjunction with the reviews and walkthroughs.

The detailed design process resulted in the following for each major module of the preliminary design:

1. Functional specification (e.g., reports, inputs, errors,etc.)
2. Program specification (more detailed program logic), and
3. Data specification/data dictionary entries.

The detailed design was also formally documented and underwent extensive internal and external review, and was formally accepted by the customer.

2.2.3 Software Construction

In this phase, which is currently underway, each program specification (a psuedocode program description) which is developed during the detailed design phase is replaced with a program description. This included a synopsis of all inputs and outputs, a top-down description of the program and a Nassi-Shneiderman description of the processing. There are extensive reviews being held primarily at the technical level.

2.2.4 Certification

This phase encompasses all the testing activities. On this project, testing is being divided into 2 parts, alpha and beta testing. During alpha testing, the customer is only minimally involved, but will play a major role in beta testing.

Alpha testing involves a fairly complete system test program. It not only includes software module and system testing but close inspection of user documentation, operating procedures and recovery procedures.

During beta testing, the system will be given to the user; after informal training the system will undergo testing in the operational environment. This testing will involve not only customer personnel but actual end users.

2.3 Documentation

The standards which predominantly guide the documentation practices are contained in the guideline which is included as part of the company methodology. For the projects surveyed, these guidelines are closely followed. Documents are formal, undergo internal and customer reviews, and are formally accepted as project products. The documents normally produced are:

1. Project plan,
2. Functional requirements description,
3. Data requirements description,
4. System/subsystem specification,
5. Program specification,
6. Database specification (if applicable),
7. User's manual (shared responsibility with the user) and

8. Operations Manual

Informal project documentation includes further design documentation, maintenance documentation, test plans, and test analysis reports.

On maintenance projects, all of the above documentation is not produced, but instead necessary changes to existing documentation are made. The state of the existing documentation varies with each system depending upon age, amount of use, etc.

There is a project underway in one of the sub-groups to standardize and improve the documentation files for production systems. The objective of the project was to localize and ensure the existence of system documentation which would be required in the case of a limited local disaster (machine crash, building fire, etc.). The primary purpose is to document recovery procedures and related information, but the overall impact is broader, including a documented record of all module descriptions, versions, and releases, programmer workbooks, and selected accounting information. The project includes the definition of the form and content of the documentation required. Certain systems have been selected as those for which these documentation files are to be completed, the completion of which would provide a very good set of system-maintenance documentation.

2.4 Quality Assurance Activities

There is not a formal or informal quality assurance program independent of individual project activities. This function is an assumed responsibility of each project manager. Reviews which are normally held on projects, do include the types of inspections (e.g., adherence to standards, documentation completeness) that are common to many QA programs. They do lack the independent perspective though.

2.5 Validation, Verification, and Testing Activities

The V,V&T program employed at this site is an extensive set of phase-by-phase reviews. These reviews include internal (project staff) inspections and walkthroughs and also formal reviews and sign-off procedures involving the customer. The adherence to the documentation guidelines, that is, the production of formal products for review during each phase, greatly contribute to the perceived success of the program. Formal change control procedures, which on the larger projects govern everything from the requirements specification through the code, also contribute to the success of the V,V&T program.

3.0 TOOLS AND TECHNIQUES

3.1 Techniques

Inspections, walkthroughs, and formal reviews are the techniques predominantly used. These are practiced employing technical, managerial, and customer staff to examine the product or project from a variety of perspectives. As stated previously in the phase discussions, these techniques are employed throughout the project.

As mentioned in section 2.5, part of the success of the techniques is attributable to the existence, form, and content of the documentation. There are several techniques used to represent various levels of specification in the documentation produced. These include Nassi-Shneiderman charts, informal English pseudocode, a formal program design language, flow charts, two types of data-flow diagrams, function-trees, variable cross-reference maps, standard data definition formats, and standard record layout forms.

Also worth noting is the fact that on many of the projects, a design methodology (commercially available) is being employed. It had been successfully used on previous projects and is being advocated for future use. Training in the use of the methodology and its associated techniques are available.

3.2 Tools

There are numerous tools in use at this site. They generally fit into two categories: specification and documentation aids, and debug aids. The documentation aids include: automated data dictionaries, an automated design specification language, flow charters and a data record layout tool. The debug tools include a set of utilities to assist in test data generation.

There is one tool being used that had been specially developed for use in this environment. It examines COBOL record descriptions and procedural modules referencing these data definitions to infer logical relationships within the data.

This group is also using a front-end processing machine to do program and data entry, compilation, file management, etc. This is a commerical product, which included a tool to automatically generate Nassi-Shneiderman charts.

3.3 Perceived Problem Areas

One problem or area mentioned for improvement is the need for more V,V&T during the small maintenance projects. A factor cited which directly constrains efforts to employ better and more formal techniques on these projects is the state of the documentation and code which is being maintained.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

This environment illustrates a moderately disciplined and formal approach in both development and maintenance. The overall framework is provided by the company standards and guidelines. These are complemented by a design methodology which was adopted, as well as numerous techniques which are frequently employed. There are also several tools which support some of the techniques.

The adherence to the standards, guidelines, and the use of techniques is endorsed by interviewees, is obviously encouraged by management and, to a large degree, is supported by training programs.

8. Operations Manual

Informal project documentation includes further design documentation, maintenance documentation, test plans, and test analysis reports.

On maintenance projects, all of the above documentation is not produced, but instead necessary changes to existing documentation are made. The state of the existing documentation varies with each system depending upon age, amount of use, etc.

There is a project underway in one of the sub-groups to standardize and improve the documentation files for production systems. The objective of the project was to localize and ensure the existence of system documentation which would be required in the case of a limited local disaster (machine crash, building fire, etc.). The primary purpose is to document recovery procedures and related information, but the overall impact is broader, including a documented record of all module descriptions, versions, and releases, programmer workbooks, and selected accounting information. The project includes the definition of the form and content of the documentation required. Certain systems have been selected as those for which these documentation files are to be completed, the completion of which would provide a very good set of system-maintenance documentation.

2.4 Quality Assurance Activities

There is not a formal or informal quality assurance program independent of individual project activities. This function is an assumed responsibility of each project manager. Reviews which are normally held on projects, do include the types of inspections (e.g., adherence to standards, documentation completeness) that are common to many QA programs. They do lack the independent perspective though.

2.5 Validation, Verification, and Testing Activities

The V,V&T program employed at this site is an extensive set of phase-by-phase reviews. These reviews include internal (project staff) inspections and walkthroughs and also formal reviews and sign-off procedures involving the customer. The adherence to the documentation guidelines, that is, the production of formal products for review during each phase, greatly contribute to the perceived success of the program. Formal change control procedures, which on the larger projects govern everything from the requirements specification through the code, also contribute to the success of the V,V&T program.

3.0 TOOLS AND TECHNIQUES

3.1 Techniques

Inspections, walkthroughs, and formal reviews are the techniques predominantly used. These are practiced employing technical, managerial, and customer staff to examine the product or project from a variety of perspectives. As stated previously in the phase discussions, these techniques are employed throughout the project.

As mentioned in section 2.5, part of the success of the techniques is attributable to the existence, form, and content of the documentation. There are several techniques used to represent various levels of specification in the documentation produced. These include Nassi-Shneiderman charts, informal English pseudocode, a formal program design language, flow charts, two types of data-flow diagrams, function-trees, variable cross-reference maps, standard data definition formats, and standard record layout forms.

Also worth noting is the fact that on many of the projects, a design methodology (commercially available) is being employed. It had been successfully used on previous projects and is being advocated for future use. Training in the use of the methodology and its associated techniques are available.

3.2 Tools

There are numerous tools in use at this site. They generally fit into two categories: specification and documentation aids, and debug aids. The documentation aids include: automated data dictionaries, an automated design specification language, flow charters and a data record layout tool. The debug tools include a set of utilities to assist in test data generation.

There is one tool being used that had been specially developed for use in this environment. It examines COBOL record descriptions and procedural modules referencing these data definitions to infer logical relationships within the data.

This group is also using a front-end processing machine to do program and data entry, compilation, file management, etc. This is a commerical product, which included a tool to automatically generate Nassi-Shneiderman charts.

3.3 Perceived Problem Areas

One problem or area mentioned for improvement is the need for more V,V&T during the small maintenance projects. A factor cited which directly constrains efforts to employ better and more formal techniques on these projects is the state of the documentation and code which is being maintained.

4.0 STANDARDS, GUIDELINES, AND PROCEDURES

This environment illustrates a moderately disciplined and formal approach in both development and maintenance. The overall framework is provided by the company standards and guidelines. These are complemented by a design methodology which was adopted, as well as numerous techniques which are frequently employed. There are also several tools which support some of the techniques.

The adherence to the standards, guidelines, and the use of techniques is endorsed by interviewees, is obviously encouraged by management and, to a large degree, is supported by training programs.

APPENDIX K
NATIONAL BUREAU OF STANDARDS SOFTWARE V&V SURVEY
PART I

Name ____________________
Department ________________
Phone ____________________

The objectives of this survey are to establish a profile of Validation and Verification (V&V) techniques currently being used, developed or proposed. Emphasis shall be placed on small to medium scale development projects.

Answer each question as it pertains to the software developed in your department. The comment space should be used for any additional information that you feel is pertinent. This space may also be used, when needed, to indicate that the question is outside the limits of your perspective or experience.

RESPONDENTS AND JOB CLASSIFICATION

Select (with a check mark) the perspective or position which best identifies you as a survey respondent.

- o Line Manager
  - o First level ________
  - o Intermediate ________
  - o Upper ________
- o User/Customer ________
- o Project Manager ________
- o Senior Analyst ________
- o Programmer Analyst ________
- o Other ________

Comments ____
____
____

I. CHARACTERISTICS OF SOFTWARE DEVELOPMENT ENVIRONMENT

PURPOSE: The objective of this survey section is to acquire information which can be used to establish the character of your software development organization and environment. This information will help us analyze the impacts of environmental conditions on the utilization of V&V technologies.

A. How many people are involved in the following software related activities?
- o Management ________
- o Quality Control and Review ________
- o Analyst/Programmer ________
- o Support/Data Entry ________
- o Computer Operations ________
- o Other ________
- o TOTAL ________

Comments ____

____

____

B. Quantify your concept of software project size classifications of small, medium, and large projects. Choose two of the following measures.

o Manmonths
Small From ________ To ________
Medium From ________ To ________
Large Above ________

o Code Statements
Small From ________ To ________
Medium From ________ To ________
Large Above ________

o Peak Load Personnel Count
Small From ________ To ________
Medium From ________ To ________
Large Above ________

Comments ____

____

____

C. How many active software development projects are you currently concerned with in each classification?

Small ________________
Medium ________________
Large ________________

o Based on your knowledge of your organization's software development activities over the past 2-3 years, what percentage of projects are in each category?

Small ________________%
Medium ________________%
Large ________________%

Comments ____

____

____

D. Identify the application areas of current software development projects in your organization, and the project sizes. (S = small, M = medium, L = large)

	Name	Size (S,M,L)
o Business-Management (e.g., Scheduling, Documentation, etc.)	________________	________
	________________	________
	________________	________
	________________	________
o Business-Financial (e.g., Payroll, Cost-Budget, etc.)	________________	________
	________________	________
	________________	________
	________________	________
	________________	________

		Name	Size (S,M,L)
o	Information Processing (e.g., Inventory, Personnel, etc.)		
o	Scientific, Engineering (e.g., Feasibility, Design, etc.)		
o	Support Software (e.g., FORTRAN Analyzer, etc.)		
o	OTHER (describe)		

Comments

E. What programming languages are used in current development? Identify, by language, number of projects and percent of total code generated as part of that project.

		No. of Projects	Percent
o	BASIC		
o	FORTRAN		
o	COBOL		
o	PASCAL		
o	PL/1		
o	ASSEMBLY		
o			
o			

Comments

F. What classes of computing equipment are used in current development? Identify the number of projects and percent of total activity in each class.

		Projects	Percent
o	Large Main Frames (e.g., CYBER, 360/70, etc.)		

	Projects	Percent
o Mini Computers (e.g., PDP 11/70, Interdata, etc.)	____________	____________
o Micro Computers (e.g., INTEL, ZILOG, etc.)	____________	____________

Comments ____

____

____

G. Does your organization acquire or use acquired software? Yes ____ No ____
NOTE: If the answer is "no" skip the remainder of G.

- o Are you in any way involved in the acquisition? Yes ____ No ____

- o Are you a user of the acquired software? Yes ____ No ____

- o What functions are satisfied by acquired software? (See Questionnaire Item D for a candidate list.)

  - o ______________________________________
  - o ______________________________________
  - o ______________________________________
  - o ______________________________________

- o The acquired software represents ______% of active software.

- o Classify the effects of the following factors on the decision to acquire or develop. (High, Medium, Low)

  - o Cost ________
  - o Available skills ________
  - o Schedule urgency ________
  - o Reliability ________
  - o Other ________

Comments ____

____

____

II. IMPACT OF MANAGEMENT AND CONTROL ON SOFTWARE DEVELOPMENT

PURPOSE: The objective of this section is to determine the existence, results of and effectiveness of formal control activities in the software development environment. The common terminology for such activities includes Quality Assurance (QA) and Configuration Management (CM). Due to the many variations in both charter and practice of these functions, this survey will treat this as one formal control, (QA/CM). Another common acronym referred to in the survey is CCB for Configuration Control Board. This survey section will provide us with data to analyze the correlations between formal controls and V&V existence and effectiveness.

A. To what degree do the following management or organizational groups impact your software development activities? Respond by stating the relative impacts, (High, Medium, Low, None).

	H, M, L, N	How?
o Customer/User		
o Line Management		
o Project Management		
o Senior Technical Staff		
o Quality Assurance		

Comments ____

____

B. Where does the responsibility for Quality Assurance (QA) and Configuration Management (CM) functions reside? (check one)

- o Formally defined QA/CM organization ____________
- o Line Management Assumed Function ____________
- o Project Management Assumed Function ____________
- o Developer Assumed Function ____________
- o Other ____________________ ____________

Comments ____

____

o To what degree are the following QA/CM functions performed on software development projects? Are these activities formal or informal?

	F/I	Always	Often	Seldom	Never
o Requirements Review					
o Design Review					
o Code Review					
o Test Review					
o Product Review					
o Audits					
o Requirements Change Control					
o Design Change Control					
o Code Change Control					
o Documentation Change Control					
o Standards Review					

Comments ____

____

____

o Do the following project variations affect the application of the QA/CM functions?

	Yes	No
o Project Size	____	____
o Internal vs Deliverable Product	____	____
o Management Interest	____	____
o Tight Scheduling	____	____
o Tight Budget	____	____
o Other	____	____

Comments ____

____

____

C. Which of the following are sources of active and accepted software development standards and procedures?

- o Corporate Policy Documentation ________
- o QA/CM Sponsored Documents ________
- o Project Adaptations ________
- o Govt. Sponsored (i.e., FIPS, DOD) ________
- o Other ________________ ________
- o Other ________________ ________

Comments ____

____

____

o How well are the following subjects covered in your current set of accepted standards?

	Excellent	Good	Fair	Poor
o Requirements Specification	____	____	____	____
o Design Methods	____	____	____	____
o Coding Standards	____	____	____	____
o Documentation Standards	____	____	____	____
o Testing Techniques	____	____	____	____
o Software Product Control	____	____	____	____
o Maintenance	____	____	____	____
o Other ____________	____	____	____	____

Comments ____

____

____

o Are there any current activities underway to expand, extend, or define new software development guidelines?

Yes ____ No ____

o Who is responsible for the Activity?

- o Company ____________
- o Project ____________
- o QA/CM ____________
- o Other ____________

- What subjects are being pursued?

  - ____________________________
  - ____________________________
  - ____________________________

- What other subjects need to be addressed or improved?

  - ____________________________
  - ____________________________
  - ____________________________

Comments ____
____
____

III. APPROACH TO SOFTWARE DEVELOPMENT

PURPOSE: This section of the survey is to provide information about your software development practices. The majority of software development activities follow similar steps from inception, through completion, use, maintenance and retirement of the software item. This time period is the "life" of the software, and present literature discusses this phenomenon as the Software Lifecycle. The lifecycle is typically expressed in phases. Generally, such a phase approach to development is followed in most software shops, though differences arise in specific nomenclature, emphasis and formality. An exemplary set of lifecycle phases is stated below (as defined in the Federal Information Processing Standards publications #38 and #64).

- Initiation (Proposal & Feasibility)
- Development
  - Requirements Definition
  - Design
  - Coding and Test
- Operations (& Maintenance)

In the context of such a phased approach, maintenance is an application of a similar sequence of phases for incorporation of corrections and enhancements.

The purpose of this survey section is to characterize your software development application in terms of formality, phases followed, products produced, factors affecting the application and effectiveness of the approach.

A. Software Lifecycle & Phased Development

1. Does your organization follow a phase approach (formal or informal) to software development and/or maintenance?

   Yes _____ No _____

NOTE: If your answer is "NO" respond to the remainder of this section based on your understanding of the organiztions common practice.

2. Is the approach formally specified within an organization standard or guideline?

Yes _____ No _____

If so, please list title(s), and how long they have been in effect:

Title	Effect
______________________________	_______________
______________________________	_______________
______________________________	_______________
______________________________	_______________

Comments ____
____
____

3. Has your Software Lifecycle approach been recently modified?

Yes _____ No _____

4. When was the last significant revision? ________________________

5. Are (further) revisions being planned?

Yes _____ No _____

6. Is there a specific organization which is responsible for the Standards or Guidelines dealing with Software Lifecycle?

Yes _____ No _____

Name ____

Comments ____
____
____

B. Lifecycle Description

1. Using the following chart, describe your software development phases.

Phase: Provide a simple descriptive phase title (1 to 3 words).

Documents Produced:
Select the document(s) produced within each phase using the code numbers.

Code	Documentation
1	Project Plan
2	Functional Requirements Doc.
3	Data Requirements Doc.
4	System Specification
5	Subsystem Specification
6	Program Specification
7	Data Base Specification
8	User's Manual
9	Operations Manual
10	Program Maintenance Manual
11	Test Plan
12	Test Analysis Report
13	____________________________

Completion Criteria:
Select the completion criteria for the phase using the code letters.

Code	Criteria
A.	Documentation Completed
B.	Complete Formal Review
C.	Complete Informal Review
D.	Management Sign-off
E.	Customer Sign-off
F.	____________________________
G.	____________________________

Phase	Documents Produced	Completion Criteria
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________
____________________	__________	__________

Comments ____

____

____

C. Indicate by checking the appropriate column which of the following documentation are produced on a typical software project. Are they formal or informal?

Documentation produced:	F/I	Always	Often	Seldom	Never
o Project Plan	____	____	____	____	____
o Functional rqmts. desc.	____	____	____	____	____
o Data requirements document	____	____	____	____	____
o System/subsystem spec.	____	____	____	____	____
o Program specification	____	____	____	____	____
o Data base specification	____	____	____	____	____
o User's manual	____	____	____	____	____
o Computer Operations Manual	____	____	____	____	____
o Program Maintenance Manual	____	____	____	____	____
o Test Plan	____	____	____	____	
o Test Analysis Report	____	____	____	____	____

Comments ____

____

D. Does your development approach vary from project to project because of any of the following factors? (i.e., how is it changed to account for the individual characteristics caused by the following)? Please briefly explain.

Factor — Explain the Changes

Size of project ____________
(e.g., small, medium, large) ______________________________

Type of application ____________
e.g., scientific, business, etc.) ______________________________

Factor	Explain the Changes
Experience of staff	____________________
Schedule constraints	____________________
Budget constraints	____________________
Other ____________________	____________________
____________________	____________________
____________________	____________________

Comments ____

E. With respect to business oriented applications, are there particular factors which change the application of software development standards or guidelines? For example:

Factor	Explain the Changes
Data Base Orientation	____________________
Data Volume	____________________
Report Requirements	____________________
Language restrictions	____________________
(e.g., COBOL, RPG)	
Data Sensitivity	____________________
Data Accuracy	____________________
Other ____________________	____________________
____________________	____________________
____________________	____________________

Comments ____

F. Approach to Software Maintenance

1. Is your group significantly involved in software maintenance?

Yes _____ No _____

Give % distribution of group activities.

- o % new development __________
- o % maintenance __________
  - o % of maintenance for error correction __________
  - o enhancement __________

2. Are there different maintenance phases; or are there a specializations of development phases to handle maintenance activities?

Yes _____ No _____

3. If so, please briefly describe the differences in your development approach versus new development for software maintenance.

____

____

____

____

____

G. Evolution of Development Approach

1. How long has your current development approach been followed?

   - o Less than 1 year __________
   - o 1-2 years . __________
   - o 2-4 years __________
   - o Longer __________

2. Is the development approach a commercial product or supported by commercial products?

   Yes _____ No _____

   Name ____

3. Identify impetus which promoted the current software development approach? (e.g., management change, new technology awareness, etc.)

   ____

Comments ____

____

____

NATIONAL BUREAU OF STANDARDS SOFTWARE V&V SURVEY
PART II

Name ______________________________
Department __________________________
Phone ______________________________

The objectives of this survey are to establish a profile of Validation and Verification (V&V) techniques currently being used, developed or proposed. Emphasis shall be placed on small to medium scale development projects.

Answer each question as it pertains to the software developed in your department. The comment space should be used for any additional information that you feel is pertinent. This space may also be used, when needed, to indicate that the question is outside the limits of your perspective or experience.

RESPONDENTS AND JOB CLASSIFICATION

Select (with a check mark) the perspective or position which best identifies you as a survey respondent.

- o Line Manager
  - o First level __________
  - o Intermediate __________
  - o Upper __________
- o User/Customer __________
- o Project Manager __________
- o Senior Analyst __________
- o Programmer Analyst __________
- o Other __________

Comments ____
____
____

I. CHARACTERISTICS OF SOFTWARE DEVELOPMENT ENVIRONMENT

PURPOSE: The objective of this survey section is to acquire information which can be used to establish the character of your software development organization and environment. This information will help us analyze the impacts of environmental conditions on the utilization of V&V technologies.

A. Indicate whether each of the following environmental factors has a High, Medium, Low, or None impact on your software development process.

H, M, L, N

o Formal Customer Interfaces ____________
o Informal Customer Interfaces ____________
o Security Requirements ____________
o Computer Environment ____________
o Multiple locations ____________
o Logistics ____________
o Management profile ____________
o Personnel turnover ____________
o Change request frequency ____________
o Language Requirements ____________
o Application Areas ____________
o Project Size ____________
o Accuracy Requirements ____________
o Failure Tolerence ____________
o Software Development Practices ____________
o Software Maintenance Practices ____________
o Software Acquisition Practices ____________
o Data Base Orientation ____________
o Data volume ____________
o Data transmission rates ____________
o Report size ____________
o Report frequency ____________
o Other ____________________ ____________
o Other ____________________ ____________

Comments ____
____
____

II. IMPACT OF MANAGEMENT AND CONTROL ON SOFTWARE DEVELOPMENT

PURPOSE: The objective of this section is to determine the existence, results of and effectiveness of formal control activities in the software development environment. The common terminology for such activities includes Quality Assurance (QA) and Configuration Management (CM). Due to the many variations in both charter and practice of these functions, this survey will treat this as one formal control, (QA/CM). Another common acronym referred to in the survey is CCB for Configuration Control Board. This survey section will provide us with data to analyze the correlations between formal controls and V&V existence and effectiveness.

A. Evaluation of factors affecting the utility of Standards, Guidelines and procedures.

1. Indicate with a check mark how each of the following factors have impacted your utilization of standards, guidelines, procedures, etc.

	FACTOR	Strong Positive	Moderate Positive	Moderate Negative	Strong Negative
o	Readability of Standards	____	____	____	____
o	Level of Detail	____	____	____	____
o	Availability of Standards	____	____	____	____
o	Applicability	____	____	____	____
o	Training	____	____	____	____
o	Local Expertise	____	____	____	____
o	Usable Reference Format	____	____	____	____
o	Usable Forms Format	____	____	____	____
o	Language Compatibilities	____	____	____	____
o	Other ____________	____	____	____	____
o	Other ____________	____	____	____	____

Comments ____

____

____

2. Indicate how the following management factors have affected the successful implementation of standards, guidelines, and procedures?

	FACTOR	Strong Positive	Moderate Positive	Moderate Negative	Strong Negative
o	Cost Impact	____	____	____	____
o	Schedule Impact	____	____	____	____
o	Other ____________	____	____	____	____
o	Other ____________	____	____	____	____

Comments ____

____

____

III. SOFTWARE VALIDATION & VERIFICATION PRACTICES AND ASSOCIATED FACTORS

The objective of this survey section is to investigate the status, results and effectiveness of your software validation and verification practices, and the utilization of such practices throughout the software lifecycle.

For the purposes of this survey "Validation and Verification" (V&V) is treated as a single conceptual term. V&V activities include all activities directed toward raising confidence in the quality of the software.

Definitions of Software Qualities

Correctness — Extent to which a program satisfies its specifications and fulfills the user's objectives.

Reliability — Extent to which a program can be expected to perform its intended function with required precision.

Efficiency
Extent to which a program performs its intended functions while minimizing required code and computing resources.

Integrity
Extent to which access to software or data by unauthorized persons can be controlled.

Maintainability
Effort required to locate and fix an error in an operational program.

Testability
Effort required to test a program to insure it performs its intended function.

Flexibility
Effort required to modify an operational program.

Portability
Effort required to transfer a program from one hardware configuration and/or software system environment to another.

Reusability
Extent to which a program can be used in other applications-related to the packaging and scope of the functions that programs perform.

Interoperability
Effort required to couple one system with another.

Usability
Effort required learn, operate, prepare inputs, and interpret output of the program.

The majority of software development activities follow similar steps from inception, through completion, use, maintenance and retirement of the software item. This time period is the "life" of the software, and present literature discusses this phenomenon as the Software Lifecycle. The lifecycle is typically expressed in phases. Generally, such a phase approach to development is followed in most software shops, though differences arise in specific nomenclature, emphasis and formality. An exemplary set of lifecycle phases is stated below (as defined in the Federal Information Processing Standards publications #38 and #64).

- Initiation (Proposal & Feasibility)
- Development
    - Requirements Definition
    - Design
    - Coding and Test
- Operations (& Maintenance)

In the context of such a phased approach, maintenance is an application of a similar sequence of phases for incorporation of corrections and enhancements.

In order to accomplish the goal of Lifecycle V&V, many methodologies have been proposed, developed and/or incorporated throughout the industry. These activities are also the subject of this survey and are referred to as Techniques and Tools. Tools and Techniques may or may not include automated support. A generic list follows:

- o Technical Reviews
- o Management Reviews
- o Standardization
- o Static Analysis
- o Traceability
- o Dynamic Analysis
- o Automated Test Generation
- o Test Support Facilities

A. Background on Your V&V Program

1. a. Does your organization have an established V&V program?
Yes _____ No _____

b. How long has the program been in existence? _____ years.

c. What organization has sponsorship responsibility for the V&V program?
o Q/A Organization _____
o Project Team _____
o Independent Team _____
o Other _____________ _____

2. Are there documented guidelines or standards pertaining to V&V practices?
Yes _____ No _____

3. Does the V&V program specifically address all development and maintenance phases?
Yes _____ No _____

Comments ____
____
____

B. Objectives of your V&V activities.

1. Please rate the following software qualities by their importance within your organization or project. (H = High, M = Moderate, S = Slight, N = None).

Quality	H	M	S	N
Correctiveness	_____	_____	_____	_____
Reliability	_____	_____	_____	_____
Efficiency	_____	_____	_____	_____
Integrity	_____	_____	_____	_____
Usability	_____	_____	_____	_____
Testability	_____	_____	_____	_____
Flexibility	_____	_____	_____	_____
Portability	_____	_____	_____	_____
Reusability	_____	_____	_____	_____
Interoperability	_____	_____	_____	_____
Others _____________	_____	_____	_____	_____
_____________	_____	_____	_____	_____

Comments ____

____

____

2. How well have your V&V activities facilitated achievement of the following objectives. (H = High, M = Moderate, S = Slight, N = None).

- o Increased confidence in the quality of the software product (code & documentation) ________

- o Increased involvement of:
  Management ________
  Customer ________
  User ________
  Individual Tech. Staff ________
  Individual reviewers ________

- o Increased visibility into the software development process and the evolving product for the:
  Management ________
  Customer ________
  User ________

Comments ____

____

____

C. Description of Lifecycle V&V Practices

1. This survey item seeks to identify the application of V&V activities by software phase. Using the following one-page form, describe the V&V activities undertaken during each phase of your software lifecycle.

For each phase, use a separate form. Provide phase name or descriptive title, types of reviews held (+ associated information) and V&V Techniques and Tools (+ associated information).

Definitions

Reviews - for the purposes of this survey, reviews have been divided into two classes, technical and management.

Technical Review - purpose is to establish the facts. Current forms of technical review include:

Inspection is a method of rapidly evaluating material by confining attention to a few selected aspects, one at a time where the points of review are guided by a checklist.

Walkthrough is a step-by-step simulation of a procedure (such as code) using an imagined set of inputs where the producer of the material to be reviewed guides the progression of the review.

Assurance Review is where the method and progression of review is dictated by someone other than the producer of the reviewed material and is usually adapted to the situation.

Management Review - purpose is to use facts from technical reviews as input to a decision involving the application of values.

Milestone Review is a formal review whose purpose is to determine progress in the areas of product performance, schedule, and costs against budgets and plans.

Visibility Review is an informal review which is used on an as needed basis to acquaint various parties on progress being made in the areas of product performance, schedule, and costs.

Buyoff Review is a formal review whose purpose is to determine the compliance of a deliverable with requirements as stated in the acceptance criteria.

V&V Tools and Techniques

Standardization - A technique used to create an authoritative model against which products and/or procedures can be compared in order to determine their quality. Examples of standards are: architecture and partitioning rules, documentation conventions, language conventions, configuration, and data management procedures.

Static Analysis - Any analytical tool or technique which is used to ascertain a quality of the software (programs, data, and documentation) which does not require the actual execution of the software.

Traceability - Any tool or technique which provides an audit trail from requirements through design and implementation of software products.

Dynamic Analyzer - A tool that instruments source code with sensors and produces reports on how thoroughly the various portions of the code have been exercised after the augmented code is executed.

Automated Test Generator - A tool that accepts a test scenario, generates the exact computer inputs, and determines the expected results.

Test Support Facility - A tool that manages test data, provides the environment for executing a single program or a set of programs, and performs test output data reduction, formatting, and printing.

LIFECYCLE V&V PRACTICES

PHASE ____

Identify reviews conducted during this phase.

		Formal (F)		
Type	Frequency[1]	Participants[2]	Effectiveness[3]	Informal (I)
Inspection				
Walkthrough				
Assurance				
Milestone				
Visibility				
Buyoff				

Identify types of V&V Tools & Techniques used during this phase.

Type	Frequency[1]	Effectiveness[3]	Comments
Standards			
Static Analysis			
Traceability			
Dynamic Analyzer			
Automated Test Generator			
Test Support Facility			

[1] Frequency	[2] Participants	[3] Effectiveness
Always	1. Management	High
Often	2. Proj. Tech. Staff	Moderate
Seldom	3. Indep. Tech. Staff	Slight
Never	4. Customer	None
	5. QA Representative	

D. Factors Affecting the Application of V&V Technology

1. Do you think that there is enough emphasis placed upon V&V in your organization?

During development Yes ____ No ____
During maintenance Yes ____ No ____

Explain ____
____
____

2. Do you consider your V&V activities to be successful?

Highly ________
Moderately ________
Slightly ________
Not at all ________

3. What percent of V&V effort expended for each of the generic phases listed below:

   Requirements ________
   Design ________
   Coding & Testing ________

4. This survey item addresses factors which may affect the degree of success of V&V activities.

   First, circle the indicator in parentheses - e.g., (A/NA) which best describes the current situation within your environment.

   Second, indicate the effect of this situation - e.g., Positive or Negative effect on the success of the V&V activities in raising confidence in the quality of the software product.

   Finally, rate the degree of this positive or negative effect. (H = High, M = Moderate, S = Slight, N = None).

   POS/NEG Effect

   Example:

   (A/NA) of need for V&V by Management ____ ____

o Awareness (A), Non-Awareness (NA)
(A/NA) of need for V&V by Management ____ ____
(A/NA) of need for V&V by Customer ____ ____
(A/NA) of need for V&V by User ____ ____
(A/NA) of need for V&V by Technical Staff ____ ____
(A/NA) of V&V technology by Management ____ ____
(A/NA) of V&V technology by Customer ____ ____
(A/NA) of V&V technology by User ____ ____
(A/NA) of V&V technology by Technical Staff ____ ____

o Availability (A), Lack of Availability (LA)
(A/LA) Published Guidelines & Standards ____ ____
(A/LA) Software Development Practices Training ____ ____
(A/LA) V&V Techniques & Tools Training ____ ____
(A/LA) Qualified Professional Staff ____ ____
(A/LA) Technical Staff Support ____ ____

o Acceptance (A), Non-Acceptance (NA)
(A/NA) Published Guidelines & Standards ____ ____
(A/NA) Phase Approach to Software Development ____ ____
(A/NA) V&V Techniques & Tools ____ ____

o Willingness (W), Resistence (R)
(W/R) Toward changes, by Management ____ ____
(W/R) Toward changes, by Customer ____ ____
(W/R) Toward changes, by User ____ ____
(W/R) Toward changes, by Technical Staff ____ ____

POS/NEG Effect

- o Support (S), Lack of Support (LS)
  (S/LS) Toward changes, by Management ____ ____
  (S/LS) Toward changes, by Customer ____ ____
  (S/LS) Toward changes, by User ____ ____
  (S/LS) Toward changes, by Technical Staff ____ ____
  (S/LS) In terms of budget allocation ____ ____
  (S/LS) In terms of schedule allocation ____ ____

5. Indicate how your V&V practices are affected by the factors listed below (High, Medium, Low, None).

Factor H, M, L, N Explanation of Effect

- o Formal Customer Interfaces ____ ____________
- o Informal Customer Interfaces ____ ____________
- o Security Requirements ____ ____________
- o Computer Environment ____ ____________
- o Multiple locations ____ ____________
- o Logistics ____ ____________
- o Management profile ____ ____________
- o Personnel turnover ____ ____________
- o Change request frequency ____ ____________
- o Language Requirements ____ ____________
- o Application Areas ____ ____________
- o Project Size ____ ____________
- o Accuracy Requirements ____ ____________
- o Failure Tolerance ____ ____________
- o Software Development Practices ____ ____________
- o Software Maintenance Practices ____ ____________
- o Software Acquisition Practices ____ ____________
- o Data Base Orientation ____ ____________
- o Data volume ____ ____________
- o Data transmission rates ____ ____________
- o Report size ____ ____________
- o Report frequency ____ ____________
- o Other ____________ ____ ____________
- o Other ____________ ____ ____________

NATIONAL BUREAU OF STANDARDS SOFTWARE V&V SURVEY
PART III

Name ______________________
Department ______________________
Phone ______________________

The objectives of this survey are to establish a profile of Validation and Verification (V&V) techniques currently being used, developed or proposed. Emphasis shall be placed on small to medium scale development projects.

Answer each question as it pertains to the software developed in your department. The comment space should be used for any additional information that you feel is pertinent. This space may also be used, when needed, to indicate that the question is outside the limits of your perspective or experience.

RESPONDENTS AND JOB CLASSIFICATION

Select (with a check mark) the perspective or position which best identifies you as a survey respondent.

- o Line Manager
  - o First level ________
  - o Intermediate ________
  - o Upper ________
- o User/Customer ________
- o Project Manager ________
- o Senior Analyst ________
- o Programmer Analyst ________
- o Other ________

Comments ____
____
____

I. VALIDATION & VERIFICATION TECHNIQUES AND TOOLS ANALYSIS

PURPOSE: The objective of this survey section is to identify the V&V Techniques and Tools utilized within your organization.

A Technique is any formalized or accepted practice such as a review, walkthrough, or chief programmer team.

An automated Tool is a program used to support (or which actually embodies) a particular technique such as an automated analyzer or an automated status tracing capability.

This part of the survey is divided into two sections. Section 1 requests a description of all V&V Techniques/Tools used in your environment. It is to be filled out once for each technique or tool. Section 2 requests information on the use and effectiveness of these Techniques/Tools. The purpose of

Section 2 is to acquire an evaluation of each Technique/Tool. Where different results have been achieved on separate applications, fill out a separate Section 2 for each application. If no additional information is added by differentiating between applications, then one Section 2 can be filled out summarizing the applications and effectiveness of the Technique/Tool. In this case please indicate the extent to which (e.g., # of projects) the technique/tool is used.

If any cost or quality effect analysis have been performed by either the Technique/Tool sponsor or its users, the surveyor would appreciate an opportunity to review the results.

Definitions of Software Qualities

Correctness	Extent to which a program satisfies its specifications and fulfills the user's objectives.
Reliability	Extent to which a program can be expected to perform its intended function with required precision.
Efficiency	Extent to which a program performs its intended functions while minimizing required code and computing resources.
Integrity	Extent to which access to software or data by unauthorized persons can be controlled.
Maintainability	Effort required to locate and fix an error in an operational program.
Testability	Effort required to test a program to insure it performs its intended function.
Flexibility	Effort required to modify an operational program.
Portability	Effort required to transfer a program from one hardware configuration and/or software system environment to another.
Reusability	Extent to which a program can be used in other applications-related to the packaging and scope of the functions that programs perform.
Interoperability	Effort required to couple one system with another.
Usability	Effort required learn, operate, prepare inputs, and interpret output of the program.

Examples of Techniques

- Structured Design / Analysis
- Preliminary Design Reviews (PDR)
- Critical / Detailed Design (CDR / DDR)
- Code walkthroughs
- Code standards
- Documentation standards

Examples of Tools

- Static Analysis:
  - DAVE
  - Compilers
  - Data Dictionaries
- Dynamic Analysis:
  - Execution counters
  - Interactive debuggers

Chief Programming Teams

Design Languages / Representation Schemes
PSL/PSA
PDL - Program Design Language
Flowchart Generators
Standard Checkers / Code Auditors
PFORT

SECTION 1 Technique/Tool Description

A. Technique/tool name (+ acronym if applicable). ____________________

B. Provide functional description (inputs, operations, outputs).

____
____
____
____
____
____
____

C. What software qualities are the primary emphasis of this technique or tool support? (check 3 or less); Express how the quality improvement is supported.

	How?	Check
o Correctness	____________	____
o Reliability	____________	____
o Efficiency	____________	____
o Integrity	____________	____
o Usability	____________	____
o Testability	____________	____
o Flexibility	____________	____
o Portability	____________	____
o Reusability	____________	____
o Interoperability	____________	____
o Other	____________	____
____________	____________	____
____________	____________	____
____________	____________	____
____________	____________	____
____________	____________	____

Comments ____
____
____

D. To what degree (percentage) is this technique/tool automated?

0% ____________
0-25% ____________
25%-50% ____________
50%-75% ____________
75%-100% ____________

E. Which of the following software lifecycle phases is supported by this tool?

Initiation ____________
Requirements ____________
Design ____________
Code & Test ____________
Operation ____________

F. Identify the project type applicablity of this tool.

	Excellent	Good	Fair	Poor
Business-Management	_____	_____	_____	_____
Business-Financial	_____	_____	_____	_____
Information Processing	_____	_____	_____	_____
Scientific & Engineering	_____	_____	_____	_____
Software Support	_____	_____	_____	_____
Other ______________	_____	_____	_____	_____

G. If this technique/tool is language dependent (i.e., works only for specified source languages), identify supported language.

Not language dependent ____________________

- o FORTRAN ____________________
- o COBOL ____________________
- o Other ________________ ____________________

________________ ____________________

________________ ____________________

H. How long has this technique/tool been used in your organization?

- o Less than 1 year ____________________
- o 1-2 years ____________________
- o 2-4 years ____________________
- o Longer ____________________

I. Identify specific projects which have utilized this tool (preferably active within the last two years).

- o ____
- o ____
- o ____
- o ____
- o ____

J. Has this technique/tool been rejected from a project it would have normally qualified for? Yes _______ No _______.

- o If yes, why?
  - o Cost efficiency ____________________
  - o Project too small ____________________
  - o Project too large ____________________
  - o Schedule constraints ____________________
  - o No trained users ____________________
  - o Project exempted ____________________
  - o Project not informed ____________________
  - o Other ______________ ____________________

    ______________ ____________________

- o Was another technique/tool used to serve the same purpose? Yes ______ No ______.

  - o If yes, what?

    ____
    ____
    ____

Comments ______________________________
____
____

K. Has cost benefit analysis been performed? Yes ______ No ______.

- o If yes, please provide a summary of the results.

  ____
  ____
  ____
  ____
  ____

Comments ______________________________
____
____

SECTION 2 Technique/tool utilization

A. Technique/tool name ______________________________

B. Describe the project(s) on which you used this technique/tool.

- o Project Name ______________________________
- o Software Type ______________________________
- o Language ______________________________
- o Number of Statements ______________________________
- o Computer System ______________________________
- o Other ____________ ______________________________

C. What was the intended purpose of this tool? (please be specific)

____
____
____

D. How well did the technique/tool fulfill its intended purpose?

	Technique/Tool
Excellent	______
Good	______
Fair	______
Poor	______

- Elaborate on strengths and weaknesses

  ____
  ____
  ____
  ____
  ____

E. Identify the attitude toward this technique/tool on this project.

	Strongly Positive	Positive	Negative	Strongly Negative
Ease of use	____	____	____	____
Response time	____	____	____	____
Cost effectiveness	____	____	____	____
Schedule Improvement	____	____	____	____
Tool Reliability	____	____	____	____
Product Improvment	____	____	____	____
Productivity	____	____	____	____
Other ____________	____	____	____	____
____________	____	____	____	____
____________	____	____	____	____

F. By phase(s) what was the impact of the assistance provided by this technique/tool?

	High	Moderate	Slight	None
Initiation	____	____	____	____
Requirements	____	____	____	____
Design	____	____	____	____
Code & Test	____	____	____	____
Operation	____	____	____	____

Comments ____
____
____

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM.

BIBLIOGRAPHIC DATA SHEET (See instructions)

1. PUBLICATION OR REPORT NO.
NBSIR 82-2482

2. Performing Organ. Report No.

3. Publication Date
April 1982

4. TITLE AND SUBTITLE
A Report on a Survey of Validation, Verification, and Testing Standards and Practices at Selected Sites

5. AUTHOR(S)
Mark K. Smith and Donna R. Hudson, Boeing Computer Services Company

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

Boeing Computer Services Co.
Seattle, Washington 98124

7. Contract/Grant No.
NB79SBCA0102

8. Type of Report & Period Covered
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

Patricia Powell
National Bureau of Standards
Room A265, Bldg. 225
Washington, D. C. 20234

10. SUPPLEMENTARY NOTES

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A survey of software validation, verification and testing (V,V&T) practices at five governmental and five commercial sites was performed. The survey collected information describing each site environment, software development and maintenance practices, the V,V&T techniques and tools employed, and standards and/or procedures guiding the activities at each site. This report summarizes the information obtained and presents observations about current operations with respect to software development, maintenance, and V,V&T. It also includes reports discussing each of the sites surveyed, and the survey instruments used.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Software validation, verification and testing, environment, standards, software development and maintenance, V,V&T techniques and tools.

13. AVAILABILITY

☒ Unlimited

□ For Official Distribution. Do Not Release to NTIS

□ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES
138

15. Price
$13.50