NIST Special Publication 500-165

Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards

Dolores R. Wallace
Roger U. Fujii

Technology

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

NIST

The National Institute of Standards and Technology[1] was established by an act of Congress on March 3, 1901. The Institute's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Institute conducts research to assure international competitiveness and leadership of U.S. industry, science and technology. NIST work involves development and transfer of measurements, standards and related science and technology, in support of continually improving U.S. productivity, product quality and reliability, innovation and underlying science and engineering. The Institute's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the National Computer Systems Laboratory, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; provides calibration services; and manages the National Standard Reference Data System. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Computing and Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[3]

The National Computer Systems Laboratory

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Laboratory consists of the following divisions:

- Information Systems Engineering
- Systems and Software Technology
- Computer Security
- Systems and Network Architecture
- Advanced Systems

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-cutting scientific themes such as nondestructive evaluation and phase diagram development; oversees Institute-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following divisions:

- Ceramics
- Fracture and Deformation[3]
- Polymers
- Metallurgy
- Reactor Radiation

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2]Some divisions within the center are located at Boulder, CO 80303.
[3] Located at Boulder, CO, with some elements at Gaithersburg, MD.

NIST Special Publication 500-165

Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards

Dolores R. Wallace

National Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Roger U. Fujii

Logicon, Incorporated
255 W. Fifth Street
San Pedro, CA 90733-0471

September 1989

NOTE: As of 23 August 1988, the National Bureau of Standards (NBS) became the National Institute of Standards and Technology (NIST) when President Reagan signed into law the Omnibus Trade and Competitiveness Act.

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) (formerly the National Bureau of Standards) has a unique responsibility for computer systems technology within the Federal government. NIST's National Computer Systems Laboratory (NCSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. NCSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. NCSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports NCSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

Library of Congress Catalog Card Number: 89-600754
National Institute of Standards and Technology Special Publication 500-165
Natl. Inst. Stand. Technol. Spec. Publ. 500-165, 37 pages (Sept. 1989)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1989

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

ABSTRACT

Software verification and validation (V&V) is a broad systems engineering approach to ensure software quality and gain optimum software performance. V&V supports the requirements for project management and quality assurance. When used with other software engineering standards, V&V helps to produce safe, secure, reliable, and maintainable software programs.

This report describes how the software verification and validation methodology and V&V standards provide a strong framework for developing quality software. First, the report describes software V&V, its objectives, recommended tasks, and guidance for selecting techniques to perform V&V. It explains the difference between V&V and quality assurance, development system engineering, and user organization functions. The report explains that V&V produces maximum benefits when it is performed independent of development functions and provides a brief discussion of how V&V benefits change when embedded in quality assurance, development systems engineering, and user organizations. An analysis of two studies of V&V's cost-effectiveness concludes that cost benefits of V&V's early error detection outweigh the cost of performing V&V.

Next the report describes several software engineering standards for V&V, project management, and quality assurance. The report describes each V&V standard according to its V&V requirements and techniques. Then the report provides an overview description of project management and quality assurance standards and explains how the V&V standards may be used along with them. The report provides insights on how to use management, quality, and V&V techniques and methodology to structure a quality software development.

Keywords: computer assurance; evaluation; project management; software development; software engineering; software maintenance; software management; software safety; software security; software standards; software testing; software verification and validation.

CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The following people have provided substantive guidance to the authors through their reviews of this report:

- Dr. William Bryan -- Grumman Data Systems
- Fletcher Buckley -- General Electric Company
- Taz Daughtrey -- Babcock and Wilcox
- Dr. Herbert Hecht -- SoHar, Incorporated
- Tom Kurihara -- Department of Defense
- Dr. Jerome Mersky -- Logicon, Incorporated
- George Tice -- Mentor Graphics Corporation
- Dr. Richard Thayer -- California State University - Sacramento
- Dr. N. Pat Wilburn -- Columbia Software

1.0 INTRODUCTION

The purpose of this report is to show how software verification and validation (V&V) standards establish a strong framework for developing quality software. The key characteristic of software V&V as an effective standard is its broad systems engineering approach to ensuring that quality is built into the software during each software life cycle phase - an approach often ignored in today's highly competitive marketplace.

However, business and governments are beginning to recognize that investment dollars placed into producing quality software return better profits and increase system safety and security (e.g., reliability). Producing reliable software requires the use of software engineering standards involving both management and product/process standards, and the use of many different quality techniques - not just review, not just inspection, or not just testing. A total software quality program requires a well-planned, comprehensive application of quality engineering disciplines implemented by all participants (e.g., management, technical engineering, quality assurance team) throughout the software development and maintenance life cycle.

Traditionally, the quality assurance standards require the development process to conform to broad quality requirements involving quality procedures, major reviews, applicable standards, documentation requirements, and general software quality attributes. Most of these quality standards do not define how to evaluate software products for compliance with technical specifications for safety, security, quality enhancement, and functional and performance requirements. Software V&V fills this gap by employing activities and tasks to provide the detailed engineering assessment (including testing) for evaluating how well the software is meeting its technical specifications. Software V&V standards, when implemented in addition to other quality standards, provide a comprehensive computer assurance program for software development efforts.

To provide an understanding of software V&V and the standards which describe it, the report is divided into three sections:

1) Overview of V&V including the V&V techniques available in each life cycle phase to evaluate and test software (Section 2);

2) Description of standards and guidelines for planning and managing V&V (Section 3); and

3) Description of general project and quality assurance standards (Section 4).

In the section describing the overview of V&V, the report first provides some historical information about V&V and its objectives. Differences in the role of V&V from other organizations (e.g., quality assurance, systems engineering, and buyer) are described. The discussion also explains how these other organizations can use V&V techniques as part of their role and responsibilities. This section describes a minimum recommended set of analyses and tests and provides guidance on how and when to select specific V&V techniques so that V&V resources can be effectively focused on the more difficult problems or areas of the software. Finally, the report analyzes two V&V case studies to provide opinions about where V&V was most effective.

Section 3 of the report describes several V&V standards and guidelines which evolved in the late 70's and early 80's. These standards are representative of the current direction of Federal agencies and industries, businesses, and academia involved in consensus standards. Other countries and international standards organizations (e.g., British Standards Institute, Australian Standards Society, Canadian Standards Organizations, and International Standards Organization) have recently developed, are developing, or are considering adopting V&V standards or quality standards referencing V&V. Bibliographies of software engineering standards may be found in [2,3,4].

The report compares and contrasts each V&V standard and guideline on how it complies with the key V&V activities. In fact, the list of key V&V activities forms a basic approach for systematically evaluating any software in determining how well the software is satisfying its performance and safety/security requirements.

Section 4 describes the generic project management and quality standards which require a V&V effort or include V&V activities as part of their domain; some do both. These generic project

level standards reference V&V to different levels of detail because each is focused on other project management or generic quality issues. However, each generic project level standard has recognized the value of V&V as a means of evaluating software's compliance with its performance, and safety/security requirements. All of the standards establish guidelines for technical review of both the interim and final products of software development and recognize that these evaluations and tests must occur at all phases of the software development life cycle. Key definitions and segments of these standards and guidelines are highlighted to provide insight on how to use the standard.

Quality software is becoming increasingly more difficult to achieve because of the larger complexities of the problem being solved and the larger scale of development efforts. The need for quality software is further stressed by the increasing use of software in critical applications not only in the obvious weapon systems but now in the control of critical day-to-day life sustaining functions. This report attempts to show that V&V is an effective methodology for controlling software developments and helping to build quality into the software before its release for use.

2.0 OVERVIEW OF SOFTWARE VERIFICATION AND VALIDATION

In 1961, a software error caused the destruction of a Mariner payload on board a radio-controlled Atlas booster. The Atlas guidance software had used incorrect radar data to compute navigation and steering commands. The cause was a simple programming error of misusing a hyphen on previous data rather than on the corrected, extrapolated data. This simple but expensive error led the Air Force to require independent review of the guidance equations and software implementation of all future mission-critical space launches. This need to ensure software quality and performance gave birth to the methodology of software verification and validation.

As the benefits of V&V became apparent in improved software quality, including safety and security, more and more systems began using it. The methodology has proliferated throughout the Department of Defense (DoD) services, the Federal Aviation Administration, and the National Aeronautics and Space Administration, as well as medical and nuclear power industries. The key V&V standards issued since 1970 are shown in figure 1; some agencies, like the Food and Drug Administration, are presently deciding how to enter V&V requirements into their policies and procedures regarding medical devices.

In many cases, V&V is governed by standards establishing software development, project management, and software quality assurance requirements. Government and industry began to develop V&V standards because managers needed a specification of this methodology for contract procurements and for monitoring the technical performance of V&V efforts.

2.1 Objectives of V&V

Software V&V comprehensively analyzes and tests software during all stages of its development and maintenance to:

- o determine that it performs its intended functions correctly,
- o ensure that it performs no unintended functions, and
- o measure its quality and reliability.

Software V&V is a systems engineering discipline which evaluates the software in a systems context, relative to all system elements of hardware, users, and other software. Like systems engineering, it uses a structured approach to analyze and test the software against all system functions and all hardware, user, and other software interfaces.

Software quality depends on many attributes, (e.g., correctness, completeness, accuracy, consistency, testability, safety, maintainability, security, reusability). Each organization involved in the software development process contributes to the building of quality of the software.

When performed in parallel with software development, V&V yields several benefits:

- o It uncovers high risk errors early, giving the design team time to evolve a comprehensive solution rather than forcing them into a makeshift fix to accommodate software deadlines.
- o It evaluates the products against system requirements.

INITIAL RELEASE	STANDARD/REGULATION
AFR 122-9/-10 1970	"Design Certification Program for Nuclear Weapon System Software and Firmware" for Air Force nuclear weapon systems software (mandatory)
AFR 800-14 1975	"Acquisition Management: Acquisition and Support Procedures for Computer Resources in Systems" for acquisition of major Air Force embedded computer systems
MIL-STD-1679 1978	"Software Development " for Navy systems
JCMPO INST 8020.1 1981	"Safety Studies, Reviews, and Evaluation Involving Nuclear Weapon Systems" for Navy nuclear cruise missile weapon systems software (mandatory)
ANSI/IEEE - ANS 7.4.3.2 1982	"Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Power Generating Stations" for Nuclear power generation embedded software
FIPSPUB101 1983	"Guideline for Lifecycle Validation, Verification, and Testing of Computer Software" for general guidance to computer software industry
DoD-STD-2167A and 2168 1985-1988	"Defense System Software Development; Quality Program" for development of DoD mission critical computer system software
ANSI/IEEE-STD 1012 1986	"Standard for Software Verification and Validation Plans" for any software development
NASA SMAP GUIDEBOOKS 1986	"Software Verificaton and Validation for Project Managers" for software intensive systems for NASA
FIPSPUB132 1987	"Guideline for Software Verification and Validation Plans" for uniform and minimum requirements of V&V; adopts ANSI/IEEE 1012
ANSI/ANS 10.4 1987	"Guidelines for V&V of Scientific and Engineering Computer Programs for the Nuclear Industry" for scientific and engineering programs (R&D) for nuclear power industry
ARMY REG 50-4 1986	"Software Studies and Reviews of Nuclear Weapon Systems" for Army nuclear weapon system software
AFSCP 800-5 1988	"Software Independent Verification and Validation" for Air Force systems with potential to cause death, system loss, more than $550K damage to equipment, or severe illness/injury
FAA STD 0-26 (DRAFT) ----	"National Aerospace System Software Development" for national airspace system-advanced automation system
FDA XXX ----	"Reviewer Guidance for Computer Controlled Medical Devices" for computer controlled medical devices.

4/89-0035-SMV-6480

Figure 1. History of V&V standards.

- It provides management with visibility into the quality and progress of the development effort that is continuous and comprehensive, not just at major review milestones (which may occur infrequently).
- It gives the user an incremental preview of system performance, with the chance to make early adjustments.
- It provides decision criteria for whether or not to proceed to the next development phase.

V&V is also used, because of its analytic approach, as a vehicle for locating high risk areas of the software system and for analyzing critical features (e.g., safety and security requirements) and the relationship of those features to the entire system.

Up to this point, V&V has been discussed as a technical discipline using a systems engineering methodology for analyzing the entire software system and for driving better performance features into and errors out of high risk, critical areas of the software. An equally important concept of V&V is to define who performs the V&V in that the V&V organization or group must possess the following characteristics:

- o be unbiased toward the software solution under review (i.e., fresh viewpoint); and
- o have a comprehensive engineering understanding of the technical problems and the possible solutions so as to judge whether performance is satisfied or errors exist.

In section 2.2, the report describes the roles of the development team, quality assurance organization, configuration and data management groups, and V&V organization. A brief description is provided of how V&V activities are different from activities performed by other organizations and groups. In section 2.3, the report identifies four methods of organizing a V&V effort:

- o independent V&V; or
- o as part of the development systems engineering group; or
- o development quality assurance group; or
- o user organization.

A brief discussion is provided on how the scope and purpose of V&V activities and tasks differ for these four methods of organizing a V&V effort. Also, the advantages and disadvantages of each method are identified.

2.2 Responsibilities of V&V Versus Other Groups

While the techniques of V&V may be applied by anyone involved in software development and maintenance, a comprehensive V&V effort is often administered by a specific group. Similarly a project may have developers who are from the end user organization or who may be contractors or subcontractors. Other groups may be quality assurance, configuration management and data management. The organizational structure of a project depends on many characteristics (e.g., size, complexity, purpose of the software, corporate culture, project standards, contractual requirements). Often these groups are separate but in many instances, especially for small projects, the structure is not as diverse. On these projects, the functions described in this section must still be performed but may be distributed differently.

A functional view demonstrates how V&V and other groups complement their software quality responsibilities. The software development group builds the software product to satisfy the established quality and performance requirements. The group relies on its quality assurance group, systems engineering, requirements analysts, designers, programmers, testers, data and configuration management specialists, documentation specialists, and others.

The quality assurance group verifies that the development process and products conform to established standards and procedures. Via reviews, audits, inspections, and walkthroughs, it acts as a formal check and balance to monitor and evaluate software as it is being built. The software systems engineering group ensures that the software product satisfies system requirements and objectives. It uses techniques such as simulations to gain reasonable assurance that system requirements are satisfied.

The configuration and data management groups monitor and control the software program versions and data during their development, using such techniques as formal audits, change control records, traceability of requirements, and sign-off records. The user group must provide assurance that the software product satisfies user requirements and operational needs. Typically, it uses techniques such as formal design reviews and acceptance testing.

The V&V group is responsible for verifying that the software product at each life cycle phase satisfies software quality attributes and that the software product at each phase satisfies the requirements of the previous phase. In addition, V&V is responsible for validating that the software satisfies overall system requirements and objectives. The activities are directed at the software, but V&V must consider how the software interacts with the rest of the system, including hardware, users, other software, and with other external systems. V&V maintains its own configuration and data management functions on programs, data, and documentation received from the development organization to assure V&V discrepancy reports are against controlled documents and to repeat V&V tests against controlled software releases. V&V responsibilities may vary for different projects; some examples are provided in section 2.3.

V&V documentation evaluation and testing are different from those conducted by other groups. The quality assurance group reviews documents for compliance to standards and performs a check on the technical correctness of the document contents. V&V may perform in-depth evaluation by such activities as rederiving the algorithms from basic principles, computing timing data to verify response time requirements, and developing control flow diagrams to identify missing and erroneous requirements. V&V may suggest, if appropriate, alternative approaches. V&V testing is usually separate from the development group's testing. In some cases, V&V may use development test plans and results and supplement them with additional tests.

2.3 Organizing a V&V Effort

A major influence on the responsibilities of V&V, and its relationship to other groups, is to whom V&V reports. Four methods of organizing a V&V effort are described: independent; embedded in the development system engineering group; embedded in the development quality assurance group; and embedded in the user group.

The traditional approach is that the V&V group is independent of the development group and is called independent V&V or IV&V. In this relationship the V&V organization establishes formal procedures for receiving software releases and documentation from the development team. V&V sends all evaluation reports and discrepancy reports to both the user (or higher level management agency in charge of the development responsibility) and development group. To maintain an unbiased technical viewpoint, V&V does not use any results or procedures from the quality assurance or systems engineering groups.

The V&V tasks are oriented toward engineering analysis (e.g., algorithm analysis, control/data flow analysis) and comprehensive testing (e.g., simulation). The objective is to develop an independent assessment of the software quality and to determine whether the software satisfies critical system requirements. Advantages of this approach are detailed analysis and test of software requirements; an independent determination of how well the software performs; and early detection of high-risk software and system errors. Disadvantages are higher cost to the project and additional development interfaces.

When the V&V group is embedded in development's systems engineering group, the V&V tasks are to review the group's engineering analyses (e.g., algorithm development, sizing/timing) and testing (e.g., test evaluation or review of the adequacy of the development test planning document). In some instances, the V&V organization may be the independent test team for the systems engineering group, sharing some test data generated by the systems engineering group. V&V's results are reviewed and monitored by the systems engineering and quality assurance groups. An independent V&V group reporting to the systems engineering group is another alternative. Advantages to using systems engineering personnel in the V&V tasks are minimum cost impact to the project; no system learning for the staff; and no additional development interfaces. A disadvantage is the loss of engineering analysis objectivity.

When the V&V group is embedded in the development's quality assurance group, its tasks take on a monitoring, auditing, and reviewing content (e.g., audit performance, audit support, test witnessing, walkthrough support, documentation review). In these tasks, the V&V group is part of quality assurance and maintains its relationship to systems engineering and other development groups in the same manner as quality assurance. The main advantages of embedding V&V as part of quality assurance are low cost to the project and bringing V&V analysis capabilities into reviews, audits, and inspections. A disadvantage is the loss of an independent software systems analysis and test capability.

When the V&V group is embedded in the user group, its tasks are an extension of the user responsibilities. The tasks consist of configuration management support of development products, support of formal reviews, user documentation evaluation, test witnessing, test evaluation of the development test planning documents, and user testing support (e.g., user acceptance testing and installation and checkout testing). As an extension of the user group, the V&V group would receive formal software product deliverables and provide comments and data to the development project management that distributes the information to its own development team. An advantage of this approach is the strong systems engineering and user perspective that can be brought to bear on the software product during development. Main disadvan-

tages are loss of detailed analysis and test of incremental software products (since these typically are not formal deliverables) and error detection and feedback to the development team constrained by the frequency of formal product deliverables. If the user group has an IV&V group reporting to it, then the disadvantages can be overcome. However, in this instance, the project incurs the disadvantage of having an additional development interface.

2.4 Applying V&V to a Software Life Cycle

The minimum recommended V&V tasks which are required by the ANSI/IEEE Standard for Software Verification and Validation Plans (SVVP) [5] for the development phases are shown in figure 2. They are considered effective and applicable to all types of software applications. Tailoring V&V for a specific project is accomplished by adding tasks to the minimum set or when appropriate, deleting V&V tasks. Figure 3 lists some optional V&V tasks in the life cycle phase where they most likely can be applied, and considerations that one might use to assign the tasks to V&V. The SVVP standard requires V&V management tasks spanning the entire software life cycle and V&V tasks for operations and maintenance.

These V&V tasks can be applied to different life cycle models simply by mapping traditional phases to the new model. Examples include variations of the traditional waterfall, Boehm's spiral development [6], rapid prototyping, or evolutionary development models [7]. The V&V tasks are fully consistent with the IEEE draft standard for software life cycle processes [8]. The SVVP standard specifies minimum input and output requirements for each V&V task; a V&V task may not begin without specific inputs, and is not completed until specific outputs are completed.

2.4.1 Management of V&V

Management tasks for V&V span the entire life cycle. These tasks are to plan the V&V process; coordinate and interpret performance and quality of the V&V effort; report discrepancies promptly to the user or development group; identify early problem trends and focus V&V activities on them; provide a technical evaluation of the software performance and quality at each major software program review (so a determination can be made of whether the software product has satisfied its requirements well enough to proceed to the next phase); and assess the full impact of proposed software changes. The output of the V&V activities consists of the Software Verification and Validation Plan (SVVP), task reports, phase summary reports, final report and discrepancy report.

Major steps in developing the V&V plan are to:

- Define the quality and performance objectives (e.g., verify conformance to specifications, verify compliance with safety and security objectives, assess efficiency and quality of software, and assess performance across the full operating environment).
- Characterize the types of problems anticipated in the system and define how they would show up in the software.
- Select the V&V analysis and testing techniques to effectively detect the system and software problems.

The plan may include a tool acquisition and development plan and a personnel training plan. The SVVP is a living document, constantly being revised as knowledge accumulates about the characteristics of the system, the software, and the problem areas in the software.

An important V&V management activity is to monitor the V&V technical progress and quality of results. At each V&V phase, planned V&V activities are reviewed and new tasks are added to focus on the critical performance/quality functions of the software and its system. The monitoring activity conducts formal reviews of V&V discrepancy reports and technical evaluation results to provide a check of their correctness and accuracy. It is critical that tight internal monitoring of the quality and accuracy of V&V results be performed, because the development group must allocate staff to review the V&V results and make the necessary software changes as indicated in the V&V results. If the V&V results are erroneous or of poor quality, the development group wastes its time and resources in the review and importantly, loses confidence in the effectiveness and helpfulness of the V&V results. V&V studies [9] have shown that responding to discrepancy reports and V&V evaluation reports consumes the largest portion

PHASE	TASKS	KEY ISSUES
Concept	Concept-documentation evaluation	Satisfy user needs; constraints of interfacing systems
Requirements Definition	Traceability analysis	Trace of requirements to concept
	Requirements validation	Correctness, consistency, completeness, accuracy, readability, and testability; satisfaction of system requirements
	Interface analysis	Hardware, software, and operator interfaces
	Begin planning for V&V system testing	Compliance with functional requirements; performance at interfaces; adequacy of user documentation; performance at boundaries
	Begin planning for V&V acceptance testing	Compliance with acceptance requirements
Design	Traceability analysis	Trace of design to requirements
	Design evaluation	Correctness; design quality
	Interface analysis	Correctness; data items across interface
	Begin planning for V&V component testing	Compliance to design; timing and accuracy; performance at boundaries
	Begin planning for V&V integration testing	Compliance with functional requirements; timing and accuracy; performance at stress limits
Implementation	Traceability analysis	Trace of source code to design
	Code evaluation	Correctness; code quality
	Interface analysis	Correctness; data/control access across interfaces
	Component test execution	Component integrity
Test	V&V integration-test execution	Correctness of subsystem elements; subsystem interface requirements
	V&V system-test execution	Entire system at limits and user stress conditions
	V&V acceptance-test execution	Performance with operational scenarios
Installation and Checkout	Installation-configuration audit	Operations with site dependencies; adequacy of installation procedure
	V&V final report generation	Disposition of all errors; summary of V&V results

4/89-0036-SMV-6480

Figure 2. Minimum set of recommended V&V tasks.

of a development group's interface time with the V&V group.

Boehm and Papaccio [10] report that the Pareto analysis, that is, 20% of the problems cause 80% of the rework costs, applies to software; they recommend that V&V "focus on identifying and eliminating the specific high-risk problems to be encountered by a software project." This does not mean that V&V should examine only 20% of the software. Rather, V&V needs to examine the entire software, prioritize the software functions by criticality, and allocate V&V analysis resources to those areas of the software which contain critical functions and high-risk problems (i.e., more error-prone). Identifying and focusing on critical and high-risk areas of the software can be addressed by two V&V methods:

- Receipt of early program deliveries for early identification of possible high-risk areas of software.
- Conduct of a "criticality analysis" to identify the most critical functions of the software.

When these methods are used together, V&V can dynamically adjust V&V analysis on the most critical areas of early program deliveries and it can provide early feedback (i.e., V&V results) on the quality of early program deliveries as well as determine how well the early program deliveries perform their critical functions.

OPTIONAL V&V TASKS	LIFE CYCLE PHASES								CONSIDERATIONS FOR SELECTING OPTIONAL V&V TASKS
	Management	Concept	Requirements	Design	Implementation	Test	Installation and Checkout	Operations and Maintenance	
Algorithm Analysis			●	●	●	●		●	- Numerical and scientific software using critical equations or models
Audit Performance									
Configuration Control					●	●	●	●	- When V&V is part of QA or user organizations; for large software developments to help QA or user staff audits
Functional					●	●	●	●	
In-Process			●	●	●	●	●	●	
Physical						●	●	●	
Audit Support									
Configuration Control					●	●	●	●	-When V&V is part of systems engineering or independent; for large software development
Functional					●	●	●	●	
In-Process			●	●	●	●	●	●	
Physical						●	●	●	
Configuration Management	●	●	●	●	●	●	●	●	- When V&V is part of user
Control Flow Analysis			●	●	●			●	- Complex, real-time software
Database Analysis			●	●	●	●		●	- Large database applications; if logic stored as parameters
Data Flow Analysis			●	●	●			●	- Data driven, real-time systems
Feasibility Study Evaluation		●						●	- High risk software using new technology or concept
Installation and Checkout Testing*			●	●	●	●	●	●	- When part of user, QA, or systems engineering
Performance Monitoring								●	- Software with changeable interfaces (man-machine)
Qualification Testing*			●	●	●	●	●	●	- When V&V is part of systems engineering or user
Regression Analysis and Testing			●	●	●	●	●	●	- Large, complex systems
Reviews Support									
Operational Readiness							●	●	- When V&V is part of systems engineering or user
Test Readiness					●	●	●	●	
Simulation Analysis			●	●	●	●	●	●	- Unavailable system test capability or need to preview concept for feasability or requirements for accuracy
Sizing and Timing Analysis				●	●	●		●	- Software constrained by memory or response time
Test Certification						●	●	●	- For critical software
Test Evaluation			●	●	●	●	●	●	- When V&V is part of QA or user
Test Witnessing						●	●	●	- When V&V is part of QA, user, or systems engineering
User Documentation Evaluation		●	●	●	●	●	●	●	- Interactive software requiring user inputs
V&V Tool Plan Generation	●							●	- When acquiring or building V&V analysis/test tools
Walkthroughs									
Design				●				●	- When V&V is part of QA or systems engineering; for large software developments to staff walkthroughs
Requirements			●					●	
Source Code					●			●	
Test					●	●	●	●	

*Test plan, test design, test cases, test procedures, and test execution

Figure 3. Optional V&V tasks and suggested applications.

4/89-0037-SMV-6480

Providing early program deliveries to V&V can be accomplished by several methods: releasing early program prototypes; using an incremental software build approach; or handing over each module or subfunction following development unit testing. Incremental software builds are one of the most effective methods of providing early program deliveries to V&V. These early deliveries reinforce the systematic analysis and test approach used by V&V to examine the software in smaller pieces while progressively evaluating larger software pieces as each new piece is integrated. High-risk software areas are easier to identify by using the incremental build approach because the V&V can:

- o have an early lead time to evaluate each engineering solution and have time to suggest alternative solutions which can be incorporated in subsequent incremental deliveries without adversely impacting the schedule;
- o isolate each new set of requirements and evaluate their impact on the system performance;
- o provide early indications of system performance to the user so that adjustments can be made to refine the desired performance; and
- o develop trend information about software anomalies and risk issues to allow time to adjust the development and V&V resources and planning to evolving software risk issues.

A software build represents a basic program skeleton including draft documentation containing portions of the full software capabilities. Each successive build integrates additional functions into the skeleton, permitting early software deliveries to V&V in an orderly development process. Based on discrepancy or progress reports, software program management can make the technical and management decisions to refocus the V&V and development team onto the program's specific problem areas of the software.

Criticality analysis, a method to locate and reduce high-risk problems, is performed at the beginning of a project. It identifies the functions and modules which are required to implement critical program functions or quality requirements (e.g., safety, security). The steps of the analysis are:

- o Develop a block diagram or control-flow diagram of the system and its software. Each block or control-flow box represents a system or software function (module).
- o Trace each critical function or quality requirement through the block or control flow diagram.
- o Classify all traced software functions (modules) as critical to either the proper execution of critical software functions or the quality requirements.
- o Focus additional analysis on these traced software functions (modules).
- o Repeat criticality analysis for each life cycle phase to observe whether the implementation details shift the emphasis of the criticality.

The criticality analysis may be used along with the cross-reference matrix of figure 4-a and figure 4-b to identify V&V techniques to address high-risk concerns. The selection of V&V techniques to use on each critical area of the program is a method of tailoring the intensity of V&V against the type of risk present in each area of the software. For example, V&V would apply algorithm analysis to critical numerical software functions, and techniques such as timing analysis, data and control flow analysis, and interface analysis to real-time executive functions.

2.4.2 Concept Definition Evaluation

In this phase, the principal V&V task is to evaluate the concept documentation to determine whether the defined concept satisfies user needs and project objectives (e.g., statement of need, project initiation memo) in terms of system performance requirements, feasibility (e.g., overestimation of hardware capabilities), completeness, and accuracy. The evaluation also identifies major constraints of interfacing systems and constraints/limitations of the proposed approach and assesses the allocation of system functions to hardware and software,

Figure 4-a. Cross-reference of V&V issues to V&V techniques/tools (part 1)

Figure 4-b. Cross-reference of V&V issues to V&V techniques/tools (part 2)

where appropriate. The evaluation assesses the criticality of each software item defined in the concept.

Most of the techniques in the cross-reference matrix of figures 4-a and 4-b are described in a publication from the National Institute of Standards and Technology (formerly the National Bureau of Standards), the National Bureau of Standards Special Publication 500-93, "Software Validation, Verification, and Testing Technique and Tool Reference Guide" [11]. In figure 4-a and 4-b, the techniques are mapped against specific V&V issues [12] which they address.

The cross-reference matrix for selecting V&V techniques and tools is applicable to all phases of the software life cycle. For example, under the "feasibility" issue, the figure shows several techniques and tools, of which the five most commonly used are analytic modeling, criticality analysis, requirements parsing, simulations, and test data generation. Of these techniques and tools, analytic modeling, requirements parsing, and simulations give the V&V analyst a way to analytically model and evaluate the desired performance; parse the requirement to determine its completeness, accuracy, and correctness; and execute test data in a simulated operating environment to determine whether the simulated performance matches the desired performance. Criticality analysis identifies the critical functions and their distribution within the system architecture. The V&V analyst evaluates the criticality analysis results to determine whether all critical functions are properly addressed and determines how well critical functions (e.g., security) are partitioned within the system to minimize interfering "cross-talk" with non-critical functions.

2.4.3 Requirements Analysis

Poorly specified software requirements (e.g., incorrect, incomplete, ambiguous, or not testable) contribute to software cost overruns and problems with reliability due to incorrect or misinterpreted requirements or functional specifications. Software that fully meets its requirements upon delivery often encounters problems in the maintenance phase because general requirements (e.g., maintainability, quality, and reusability) were not accounted for during the original development. The problem of outdated requirements is intensified by the very complexity of the problems being solved (which causes uncertainty in the intended system performance requirements) and by continual changes in requirements (e.g., to incorporate new technologies, new missions, changes in interfacing systems, new people coming on the scene). V&V tasks verify the completeness of all the requirements.

The most commonly used optional V&V tasks listed in figure 3 for requirements analysis are control flow analysis, data flow analysis, algorithm analysis, and simulation. Control and data flow analysis are most applicable for real time and data driven systems. These flow analyses transform logic and data requirements text into graphic flows which are easier to analyze than the text. PERT, state transition, and transaction diagrams are examples of control flow diagrams. Algorithm analysis involves rederivation of equations or evaluation of the suitability of specific numerical techniques. Simulation is used to evaluate the interactions of large, complex systems with many hardware, user, and other interfacing software components.

Another activity in which V&V plays an important role is test management. V&V looks at all testing for the software system and ensures that comprehensive testing is planned. V&V test planning begins in the requirements phase and spans almost the full range of life cycle phases. Test planning activities encompass four separate types of testing - component, integration, system, and acceptance testing. The planning activities result in documentation for each test type consisting of a test plan, test design, test case, and test procedure documents. When V&V is performed by an independent organization, V&V performs all four types of testing indicated above. When V&V tasks are embedded as part of other organizations, V&V may not perform all the testing but may review the test plans and test results produced by the development group. The following paragraphs describe the four V&V testing methods.

V&V component testing verifies the design and implementation of software units, modules, or subelements. Typically, V&V component testing is performed on only the critical components. V&V integration testing verifies functional requirements as the software components are integrated together. Special attention is focused on software, hardware, and operator interfaces.

V&V system testing validates the entire software program against system requirements and software performance objectives. These V&V system tests are to validate that the software executes correctly within its stated operating environment. The software's ability to properly deal with anomalies and stress conditions is emphasized. These tests are not intended to duplicate or replace the user's and development group's test responsibilities, but instead supplement the development testing to test behavior not normally tested by the user or development group.

Acceptance testing validates the software against V&V acceptance criteria, defining how the software should perform with other completed software and hardware. The main distinction between V&V system and acceptance testing is that the former uses a laboratory environment in which some system features are simulated or performed by non-operational hardware or software, and the latter uses an operational environment with final configurations of other system hardware and software. V&V acceptance testing usually consists of a limited number of tests to demonstrate that the software will execute as predicted by V&V system testing in the operational environment. Full acceptance testing is the responsibility of the user and the development systems engineering group.

2.4.4 Design Evaluation

The minimum set of design phase V&V tasks involving traceability, interface analysis, and design evaluation provides assurance that requirements are not misrepresented or incompletely implemented, unwanted requirements are not designed into the solution by oversight, and requirements are not left out of the design. Design errors can be introduced by implementation constraints relating to timing, data structures, memory space, and accuracy, even though the basic design satisfies the functional requirements.

The most commonly used V&V tasks from the optional V&V tasks listed in figure 3 are algorithm analysis, database analysis, timing/sizing analysis, and simulation. In this phase, algorithm analysis examines the correctness of the equations or numerical techniques as in the requirements analysis phase, but also examines truncation and round-off effects, numerical precision of word storage and variables (e.g., single- vs. extended-precision arithmetic), and data typing influences. Database analysis is particularly useful for programs that store program logic in data parameters. A logic analysis of these data values is required to determine the effect these parameters have on program control. Timing/sizing analysis is useful for real-time programs having response time requirements and constrained memory execution space requirements.

2.4.5 Implementation (Code) Evaluation

Clerical and syntactical errors have been greatly reduced through use of structured programming and reuse of code, adoption of programming standards and style guides, availability of more capable computer languages, better compiler diagnostics and automated support, and, finally, more knowledgeable programmers. Nevertheless, problems still occur in translating design into code and can be detected with some V&V analyses.

Commonly used V&V tasks from the optional task listed in figure 3 are control flow analysis, database analysis, regression analysis, and sizing/timing analysis. For large code developments, control flow diagrams showing the hierarchy of main routines and their subfunctions are useful in understanding the flow of program control. Database analysis is performed on programs with significant data storage to ensure that common data and variable regions are used consistently between all call routines; data integrity is enforced and no data or variable can be accidentally overwritten by overflowing data tables; and data typing and use are consistent throughout all program elements. Regression analysis is used to reevaluate requirements and design issues whenever any significant code change is made. This technique ensures project awareness of the original system requirements. Sizing/timing analysis is done during incremental code development and compared against predicted values. Significant deviation between actual and predicted values is a possible indication of problems or the need for additional examination.

Another area of concern to V&V is the ability of compilers to generate object code that is functionally equivalent to the source code, that is, reliance on the correctness of the language

compiler to make data dependent decisions about abstract programmer coded information. For critical applications, this problem is solved by validating the compiler or by validating that the object code produced by the compiler is functionally equivalent to the source.

Other tasks indicated in figures 4-a and 4-b for code evaluation are walkthroughs, code inspections and audits. These tasks occur in interactive meetings attended by a team which usually includes at least one member from the development group. Other members may belong to the development group or to other groups involved in software development. The duration of these meetings is usually nc more than a few hours in which code is examined on a line-by-line basis. In these dynamic sessions, it may be difficult to examine the code thoroughly for control logic, data flow, database errors, sizing, timing and other features which may require considerable manual or automated effort. Advance preparation for these activities may be necessary and includes the optional V&V tasks of figure 3 and others shown in figures 4-a and 4-b. The results of these tasks provide appropriate engineering information for discussion at meetings where code is evaluated. Regardless of who conducts or participates in walkthroughs and inspections, V&V analyses may be used to support these meetings.

2.4.6 Testing

As already described, V&V test planning is a major portion of V&V test activities and spans several phases. A comprehensive test management approach to testing recognizes the differences in objectives and strategies of different types of testing. Effective testing requires a comprehensive understanding of the system. Such understanding develops from systematically analyzing the software's concept, requirements, design, and code. By knowing internal software details, V&V testing is effective at probing for errors and weaknesses that reveal hidden faults. This is considered structural, or white-box, testing. It often finds errors for which some functional, or black-box, test cases can produce the correct output despite internal errors. Functional test cases execute part or all of the system to validate that the user requirement is satisfied; these test cases cannot always detect internal errors that will occur under special circumstances. Another V&V test technique is to develop test cases that violate software requirements. This approach is effective at uncovering basic design assumption errors and unusual operational use errors.

The most commonly used optional tasks are regression analysis and test, simulation, and user document evaluation. User document evaluation is performed for systems having an important operator interface. For these systems, V&V evaluates and tests the user documentation to verify that the operating instructions are consistent with the operating characteristics of the software. The system diagnostic messages and operator recovery procedures are examined to ensure their accuracy and correctness with the software operations.

2.4.7 Installation and Checkout Activities

During installation and checkout, V&V validates that the software operates correctly with the operational hardware system and with other software, as specified in the interface specifications. V&V may verify the correctness and adequacy of the installation procedures and certify that the verified and validated software is the same as the executable code delivered for installation. There may be several installation sites with site-dependent parameters. V&V verifies that the program has been accurately tailored for these parameters and that the configuration of the delivered product is the correct one for each installation.

Optional V&V tasks most commonly used in this phase are regression analysis and test, simulation, and test certification. Any changes occurring from installation and test are reviewed using regression analysis and test to verify that our basic requirement and design assumptions affecting other areas of the program have not been violated. Simulation is used to test operator procedures and to help isolate any installation problems. Test certification, especially in critical software systems, is used to demonstrate that the delivered software product is identical to the software product subjected to V&V.

2.4.8 Operations and Maintenance Evaluation and Test

For each software change made in the operations and maintenance phase, all life cycle phase V&V activities of figure 2 are considered and possibly repeated to ensure that nothing is overlooked. V&V activities are added or deleted to address the type of software change made. In many cases, an examination of the proposed software change shows that V&V needs to repeat its activities on only a small portion of the software. Also, some V&V activities such as concept documentation evaluation require little or no effort to verify a small change. Small changes can have subtle but significant side-effects in a software program.

If V&V is not done in the normal software development phase, then the V&V in the maintenance phase must consider performing a selected set of V&V activities for earlier life cycle phases. Some of the activities may include generating requirements or design information from source code, a process known as reverse engineering. While costly and time consuming, it is necessary to gain high confidence that subtle but critical errors have been removed.

2.5 Effectiveness of V&V

Two studies to evaluate the effectiveness of V&V as an independent organization used different data and reported on different factors. While no direct comparison of results is possible, insights on V&V effectiveness may be gained from understanding the results of each study.

In 1982, McGarry [13] reported that V&V was not an effective approach on three small projects at the Software Engineering Laboratory (SEL) at NASA Goddard Space Flight Center. Three flight dynamics projects ranging in size from 10K to 50K lines of code were selected. V&V was involved in requirements and design verification, separate system testing, and validation of consistency from start to finish. The V&V effort lasted 18 months and used an average of 1.1 staff-persons, peaking at 3 staff-persons. Some results were as follows:

- Productivity of the development teams was the lowest of any previously monitored SEL project (due to the V&V interface).
- Rates of uncovering errors early in the development cycle were better.
- V&V found 2.3 errors per thousand lines of code.
- Cost rate to fix all discovered errors was no less than in any other SEL project.
- Reliability of the software (error rate during acceptance and maintenance and operations) was no different from other SEL projects.

Radatz's 1981 study [9] for Rome Air Development Center reported V&V effectiveness results for four large IV&V projects ranging from 90K to 176K lines of code. The projects were real-time command and control, missile tracking, and avionics programs and a time-critical batch trajectory computation program. The projects varied from 2.5 to 4 years to develop. Two projects started V&V at the requirements phase, one at the code phase and one at testing. The V&V organization used 5 to 12 staff-persons per project. Some results were:

- Errors were detected early in the development -- 50% to 89% detected before development testing began.
- Large number of discrepancies were reported (total 1259) on an average of over 300 per program.
- V&V found an average 5.5 errors per thousand lines of code.
- Over 85% of the errors affected reliability and maintainability.
- Effect on programmer productivity was positive, that is, hours of programmer time saved by the programmer's not having to find the error, minus the time required to evaluate the V&V error report -- total savings per error of 1.3 to 6.1 hours of programmer time and over 7 minutes of computer time.
- For the two projects beginning at the code phase, early error detection savings amounted to 20%-28% of V&V costs; for the two projects beginning at the requirements phase, early error detection

savings amounted to 92% - 180% of V&V costs.

There are several differences between the two studies. The most obvious difference is that the largest project in the McGarry study was just over half the size of the smallest project in the Radatz study. Another is that V&V found almost twice the number of errors per thousand lines of code in the Radatz study than in the McGarry study. Both studies involved projects of considerable difficulty regardless of size. Why is the discovered error rate so different between the studies? Is it reasonable to compare error rates of small projects against error rates of large projects? Was either the development group or the V&V group more experienced in either experiment? These questions are difficult to answer but one tentative conclusion is that project parameters will affect the benefits of V&V. After an examination of both the positive and negative benefits of V&V, some insights are provided on parameters that affect V&V.

Based on these studies, some positive effects of V&V on a software project include:

- Better quality (e.g., complete, consistent, readable, testable) and more stable requirements.
- More rigorous development planning, at least to interface with the V&V organization.
- Better adherence by the development organization to programming language and development standards and configuration management practices.
- Early error detection and reduced false starts.
- Better schedule compliance and progress monitoring.
- Greater project management visibility into interim technical quality and progress.
- Better criteria and results for decision-making at formal reviews and audits.

Some negative effects of V&V on a software development project include:

- Additional project cost of V&V (10%-30% extra).
- Additional interface involving the development team, user, and V&V organization (e.g., attendance at V&V status meeting, anomaly resolution meeting).
- Lower development staff productivity if programmers and engineers spend time explaining the system to V&V analysts and resolving invalid anomaly reports.
- Additional documentation requirements, beyond the deliverable products, if V&V is receiving incremental program and documentation releases.
- Need to share computing facilities with, and to provide access to, classified data for the V&V organization.
- Increased paperwork to provide written responses to V&V error reports and other V&V data requirements (e.g., notices of formal review and audit meetings, updates to software release schedule, response to anomaly reports).

Some steps can be taken to minimize the negative effects and to maximize the positive effects of V&V. To recover much of the V&V costs, V&V is started early in the software requirements phase to allow the earliest error detection when correction costs are lowest. The interface activities for documentation, data, and software deliveries between developer and V&V groups should be considered as an inherently necessary step required to evaluate intermediate development products. This is a necessary by-product of doing what's right in the beginning.

To offset unnecessary costs, V&V must organize its activities to focus on critical areas of the software so that it uncovers critical errors for the development group and thereby results in significant cost savings to the development process. To do this, V&V must use its criticality analysis to identify critical areas and it must scrutinize each discrepancy to ensure that no false or inaccurate information is released to

prevent the development group from wasting time on inaccurate or trivial reports.

To eliminate the need to have development personnel train the V&V staff, it is imperative that V&V select personnel who are experienced and knowledgeable about the software and its engineering application. When V&V engineers and computer scientists reconstruct the specific details and idiosyncracies of the software as a method of reconfirming the correctness of engineering and programming assumptions, they often find subtle errors. They gain detailed insight in to the development process and an ability to spot critical errors early. The cost of the development interface is minimal, and at times nonexistent, when the V&V assessment is independent.

Finally, the number of discrepancies detected in software and the improvement in documentation quality resulting from error correction suggests that V&V costs are offset by having more reliable and maintainable software. Many companies rely on their software systems for their daily operations. Failure of the system, loss of data, release of or tampering with sensitive information may cause serious work disruptions and serious financial impact. The costs of V&V are offset in many application areas by increased reliability during operation and reduced costs of maintenance.

3.0 STANDARDS AND GUIDELINES FOR PLANNING AND MANAGING V&V

The documents in figure 5 establish guidelines for planning and managing a V&V effort. Their activities produce information that satisfies the life cycle requirements of standards governing projects. They have the following features:

- Require V&V to determine how well evolving and final software products comply with their requirements.
- Permit users to select specific techniques to satisfy their application needs.
- Cover a broad spectrum of V&V activities.

The NIST issued the Federal Information Processing Standards Publication "Guideline for Lifecycle Validation, Verification and Testing," in 1983 [14]. This document was followed in 1987 with the "Guideline for Software Verification and Validation Plans" [15] which adopted the ANSI/IEEE standard for V&V planning [5]. Reference to the guideline, FIPSPUB132, includes reference to the ANSI/IEEE specifications.

According to Branstad [1], standards for use by large heterogeneous communities should provide direction for specific project implementations, with information on V&V planning, review points, verification techniques, testing, and reporting. The features in the documents listed in figure 5 include organization guidelines, planning and management direction, life cycle concerns, V&V phase requirements, and software test management. A comparison and contrast of these features leads to an approach for developing a V&V effort based on the strengths of the guidance in the documents.

3.1 Organization

V&V activities may be performed by anyone responsible for assuring the quality of software. Developers perform some V&V activities in the normal course of developing their product. Complementary, supplementary, or duplicate V&V activities may be assigned to a software quality assurance group within the developer's company or an outside organization, usually referred to as IV&V. In the most formal arrangement, an organization independent of both the developer and the customer of the software system is contracted to perform the V&V activities.

A master SVVP allocates the major tasks of all parties responsible for V&V activities for assuring the quality of the software. The example of V&V planning in figure 6 contains several SVVPs and focuses on the distribution of test responsibilities; each SVVP contains descriptions of other V&V tasks. In contrast, in a small project with a developer performing all the V&V activities, the developer's SVVP may be the only SVVP and may even be included in the project plan.

The example of figure 6 represents a more complex project. The developer is responsible for component and integration testing, with integration test documentation examined by a IV&V organization. The IV&V organization is responsible for system testing and for assistance to the customer for acceptance testing. The

FIPSPUB101	Guideline for Lifecycle Validation, Verification, and Testing of Computer Software
FIPSPUB132	Guideline for Software Verification and Validation Plans
ANSI/IEEE STD 1012	Standard for Software Verification and Validation Plans
AFSC/AFLCP 800-5	Software Independent Verification and Validation
ANS 10.4	Guidelines for the Verficiation and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry
JPL D 576	Independent Verification and Validation of Computer Software: Methodology

4/89-0038-SMV-6480

Figure 5. Selected guidance for planning V&V.

customer plans for acceptance testing. Developers and sub-developers may be part of the development organization who will also use the software or they may be under contract to a customer; they may be responsible for component testing of their components. The master SVVP will allocate these responsibilities; the developer's SVVP will elaborate on unit testing and integration testing; the IV&V's SVVP will clarify its role in integration test and acceptance test and will contain complete planning for system test. This example is provided to demonstrate that a complete system approach which integrates the responsibilities of all project groups is essential to meeting life cycle requirements for the assurance of software quality.

FIPSPUB101 permits performance of V&V activities by developers, the same organization, or some independent group [14]. FIPSPUB132/IEEE1012 does not require independence; it does require the SVVP to "define the relationship of V&V to other efforts such as development, quality assurance, configuration or data management, or end user" [5,15]. Internal and external lines of communication to V&V must be defined; V&V could occur independently or within one of the other efforts.

The Air Force pamphlet, "AFSC/AFLCP 800-5 Software Independent Verification and Validation," [16] is concerned only with software IV&V. It describes V&V activities typically performed by an independent V&V group separate from the developer's quality assurance group required by DOD-STD-2167A Standard, "Defense System Software Development" [17]. The AF pamphlet provides the criteria for selecting an independent V&V group.

The V&V activities of "Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry," ANS 10.4, [18] may be performed by the program developer, as a task separate from development, or by an IV&V agent. The guideline contains an example of a division of V&V responsibilities.

The "Independent Verification and Validation of Computer Software: Methodology" from the Jet Propulsion Laboratory (JPL) [19] states that V&V activities should be performed independently of the development organization to ensure effectiveness and integrity of the V&V effort. The document allows flexibility in selecting the extent of the detailed V&V effort it describes.

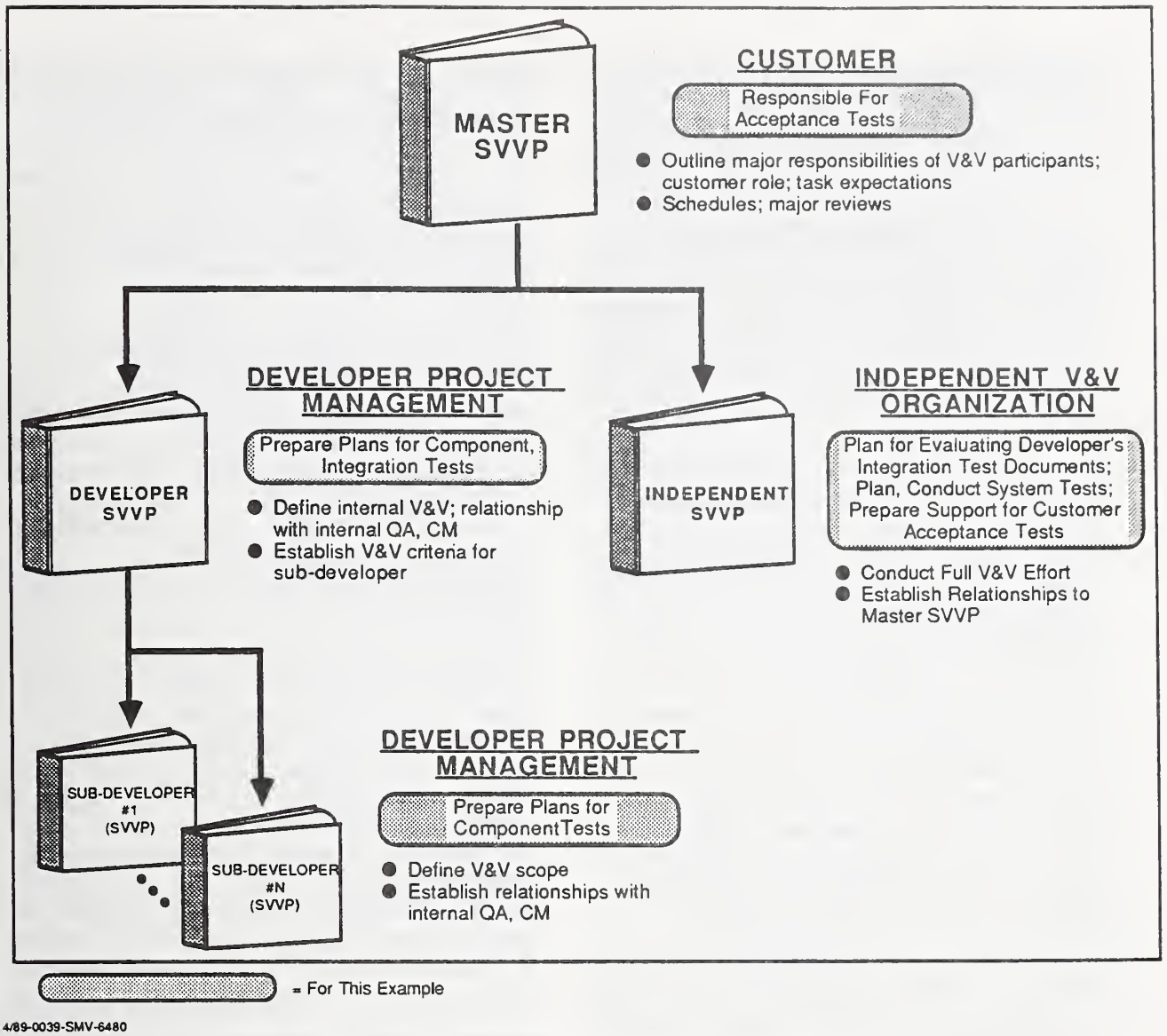

Figure 6. Example: organizing V&V testing with several SVVPs

3.2 Planning and Management

Steps for planning an effective V&V effort include the following:

- o determining V&V objectives and project needs by performing a criticality analysis;
- o planning and organizing the full spectrum of V&V activities over the project;
- o managing the effort;
- o reporting on the effort.

Criticality analysis. The requirements of the V&V documents are based on the criticality of the software. FIPSPUB101 recommends specific V&V activities for three levels of software, from a small, simple project to a large, complex project. The basic set includes tasks like preparation of the V&V plan, review and analysis of software products, and testing; the final set includes correctness proofs and techniques using sophisticated automation. Each successively more detailed and comprehensive level includes activities of the level(s) beneath it.

FIPSPUB132/IEEE1012 requires an assessment of criticality of each software item. For critical software, it requires tasks of traceability, evaluation, interface analysis, testing, management and reporting for each phase. It recommends the same task selection for non-critical software and provides an optional task list for all

software. The planner is required to consider all tasks and to justify omission of any required task.

The AF pamphlet provides detailed instructions for conducting a criticality assessment with four levels ranging from catastrophic to negligible. The AF pamphlet defines a complete method for determining the criticality level of each software requirement and for computing the overall criticality level of the system. V&V tasks are selected based upon the computed criticality levels, where the scope and complexity of the V&V activities increase as the criticality increases.

The AF pamphlet defines a complete method for determining the criticality of each software requirement and for computing the system's overall criticality level. V&V tasks are selected by where the criticality fits into one of three tables, where scope and complexity increase as the criticality increases.

Plan Preparation. FIPSPUB101, FIPSPUB132/IEEE1012 and ANS 10.4 define the minimum content information for a software V&V plan. FIPSPUB101 provides an example plan in an appendix, and FIPSPUB132/IEEE1012 provides a uniform format for presenting the information. Only the AF pamphlet provides guidance on estimating the costs but its scope does not include plan definition, format or content.

Management. FIPSPUB101, FIPSPUB132/IEEE1012, and ANS 10.4 discuss the initial SVVP and updates to it. FIPSPUB132/IEEE1012 is the only document that requires ongoing V&V management tasks spanning the entire project life cycle. These include SVVP generation and updates, baseline change assessment for V&V activities, management reviews, review support, and reporting. The SVVP is updated because of project changes and changes indicated by findings of V&V tasks. The AF pamphlet provides strong direction in establishing initial software V&V requirements.

Reporting. FIPSPUB101 recommends test reports, test evaluation reports and problem reports. FIPSPUB132/IEEE1012 requires planning for V&V reporting and specifies content for interim and final task reports, phase summary reports, anomaly reports, and a V&V final report after installation. The AF pamphlet makes no recommendations on reporting of V&V activities. ANS 10.4 specifies content requirements for a test report, for a final V&V report after installation, and for a V&V review report during operations and maintenance. The JPL document suggests assessment reports after each V&V activity.

3.3 Life Cycle, Iteration, and Maintenance

FIPSPUB101, FIPSPUB132/IEEE1012, ANS 10.4 and the JPL document use reference life cycles, similar to the waterfall model, as context for presenting software V&V requirements. FIPSPUB132/IEEE1012 identifies products for evaluation and inputs for supporting each V&V phase task but permits other life cycles. The AF pamphlet is directly tied to the life cycle requirements and evaluation criteria of DOD-STD-2167A [17].

Iteration. Only FIPSPUB132/IEEE1012 makes a direct statement about iteration; the issue is that changes will be made to almost every software system, if not during development, then during maintenance. FIPSPUB132/IEEE1012 requires a SVVP to establish a "task iteration policy" and to provide for assessment of proposed software changes for their effect on V&V tasks and the SVVP. FIPSPUB132/IEEE1012 requires the master schedule to recognize that V&V activities are iterative.

Maintenance. FIPSPUB101 defines V&V activities for the operations and maintenance phase and recommends the repetition of V&V activities of affected development phases. FIPSPUB132/IEEE1012 requires the initial SVVP to include an estimate of anticipated V&V activities during operation and maintenance; this estimate is updated prior to operation and maintenance. The required management task of baseline change assessments provides continuing direction for reperforming previous or initiating new software V&V tasks. ANS 10.4 provides guidance in determining which V&V activities are applicable during maintenance; it also provides criteria for determining how to perform software V&V on completed software that has not undergone a V&V effort.

3.4 V&V Phase Requirements

For each phase of the life cycle, the guidance documents address consistency, evaluation, and review.

Consistency Between Phases. FIPSPUB101, FIPSPUB132/IEEE1012, ANS 10.4 and the JPL document address internal consistency of software products as one objective of general evaluation activities and require traceability analysis from the system/software requirements through successive documentation. FIPSPUB132/IEEE1012 requires planning for traceability of all test documentation to the system requirements. The AF pamphlet addresses consistency through requirements of DOD-STD-2167A [17].

Interface analysis is required at least indirectly by all the documents. FIPSPUB132/IEEE1012 requires an analysis of the software's relationship to the total system thorough interface analysis of requirements documentation, design documentation, interface documentation, and the source code. FIPSPUB132/IEEE 1012 specifies that the software documentation is evaluated with hardware, software, user, and operator interface requirements, including testing of the performance at these interfaces. The AF pamphlet mentions checking the consistency of external and internal interface requirements for the software requirements, the design, and the code. The JPL document provides a checklist of interface analysis questions.

V&V Evaluation Activities. V&V activities selected for any effort are based upon the characteristics of the application or system software under evaluation. The activities selected are also governed by the scope of V&V as defined by its organizational responsibilities. None of the standards specify the set of V&V activities or techniques to use for all applications. Most, like FIPSPUB132, define a recommended set of V&V activities based on traceability, interface, and phase-by-phase activities (fig. 2) which may be tailored to each user's needs by adding V&V techniques similar to those indicated in figures 3 and 4a-4b.

Review. All the V&V documents address reviews of outputs of life cycle phases (e.g., concept documentation, system requirements, software management plans, user documentation). FIPSPUB132/IEEE1012 considers conduct of formal reviews as an optional task for V&V, but the V&V effort provides information for formal reviews as a required management task. The JPL document requires IV&V attendance at formal review meetings.

3.5 Software Test Management

All V&V standards and guidelines include directives for general software testing but FIPSPUB101, FIPSPUB132/IEEE1012, and the JPL document define four types of testing: unit or component test, integration or subsystem test, system test, and acceptance test. FIPSPUB132/IEEE1012 provides criteria for system test planning to determine if the software satisfies system objectives.

FIPSPUB132/IEEE1012 addresses test management by identifying objectives and a timely sequence of test planning documentation and execution for each test type. For each test type, test documentation includes plans describing the approach, tool and training needs, objectives, schedules, designs of the test structure and code, cases containing the actual test data for each test, and procedures with complete details for executing each test. With completed test documentation, testers should have resources available for executing and analyzing the tests. For small projects, separate documents may not be necessary, but the total spectrum of information is. FIPSPUB132/IEEE1012 requires planning for tracing of all test documentation to requirements. Requirements for the SVVP overview section include identifying any special tool needs for V&V activities, including testing.

FIPSPUB101 and ANS 10.4 contain outlines of a generic test plan. Both ANS 10.4 and the JPL document have detailed checklists for verifying the adequacy of a test plan. ANS 10.4 contains a checklist for verification of test results.

The AF pamphlet allocates test activities between developers and IV&V according to the level of criticality; the activities range from evaluating developers' critical test results to conducting special tests in critical areas. ANS 10.4 defines four levels of test activities, ranging from testing only by the software developer with no separate V&V effort; to variations of testing by developer and independent team as well as evaluation by independent team; and finally, complete testing performed separately by the developer and by an independent team.

FIPSPUB101 recommends levels of test coverage by statement, module, and logical path coverage. FIPSPUB132/IEEE1012 addresses functional test coverage and coverage of performance, reliability and maintainability, and user documentation. ANS 10.4 establishes coverage requirements based on software requirements.

3.6 Summary of V&V Standards

As indicated by these guidance documents, a V&V effort consists of tasks from a broad spectrum of analysis and test techniques to tailor each V&V effort to project needs, where the basic tasks are the following:

- traceability of software requirements through all documentation,
- evaluation or review of interim and final software products, including user documentation,
- interface analysis,
- software testing.

By use of V&V techniques shown in figures 2, 3, and 4a-4b and other techniques, high risk errors are detected early, software performance is improved, and higher confidence is established in software reliability. The additional cost of conducting V&V is offset by cost advantages of early error detection and improved software reliability.

The V&V guidance documents complement and supplement one another so that together they provide valuable direction for anyone responsible for the quality of software. The AF pamphlet addresses the major activities for determining the organization and scope of software V&V for a project. Only FIPSPUB132/ IEEE1012 addresses software V&V management throughout the life cycle. Most guidance documents address planning and reporting for software V&V. The study to compare and contrast the document content of software V&V standards and guidelines led to the conclusion that the documents contribute to a systematic approach for the planning and management of a software V&V effort. In figure 7 each step of this systematic approach is mapped to those documents providing strong guidance for that step. For any project, it is important to recognize the need to tailor the requirements of these documents to different life cycles and project requirements.

4.0 GENERAL PROJECT AND QUALITY ASSURANCE STANDARDS

Many software engineering standards address primary requirements for project management and documentation requirements over variations of a similar life cycle (fig. 8). A life cycle provides a framework of steps, usually called phases, to enable the coordination and control of development and the operation and maintenance of a software system. Software development, at a minimum, includes written requirements describing what the system must do, an overview design describing how the system will be built, a more detailed design description from which the programmers write the code, the code itself, and user documentation. The standards and guidelines described in this report require review of this documentation. Several also address the need for and require review of documentation for software product assurance activities: quality assurance, V&V including testing, and configuration management. Most also call for audits during the life cycle. The purpose of the reviews and audits is to ensure that the goals of each phase's activities have been met sufficiently to proceed to the activities of the next phase.

The project level standards (fig. 9 and 10) are striving toward recognition that each participating group has an important role in building, reviewing, and assuring the quality of the software. The major variances among the standards and guidelines occur in the refinement of the life cycle phases, the relationship of software phases to system phases, and specific names for the phases and the products produced in the phases. Differences in specific phase and product names do not change the need for activities to provide the engineering information concerning how well the evolving software system will satisfy its requirements.

Project standards view V&V either as a separate activity performed by different groups or as an intrinsic activity performed by the developer. In the first case, the standards require V&V, usually with separate project documentation or with a specific section of the software management plan devoted to V&V. A criticality assessment is a common mechanism to determine the amount

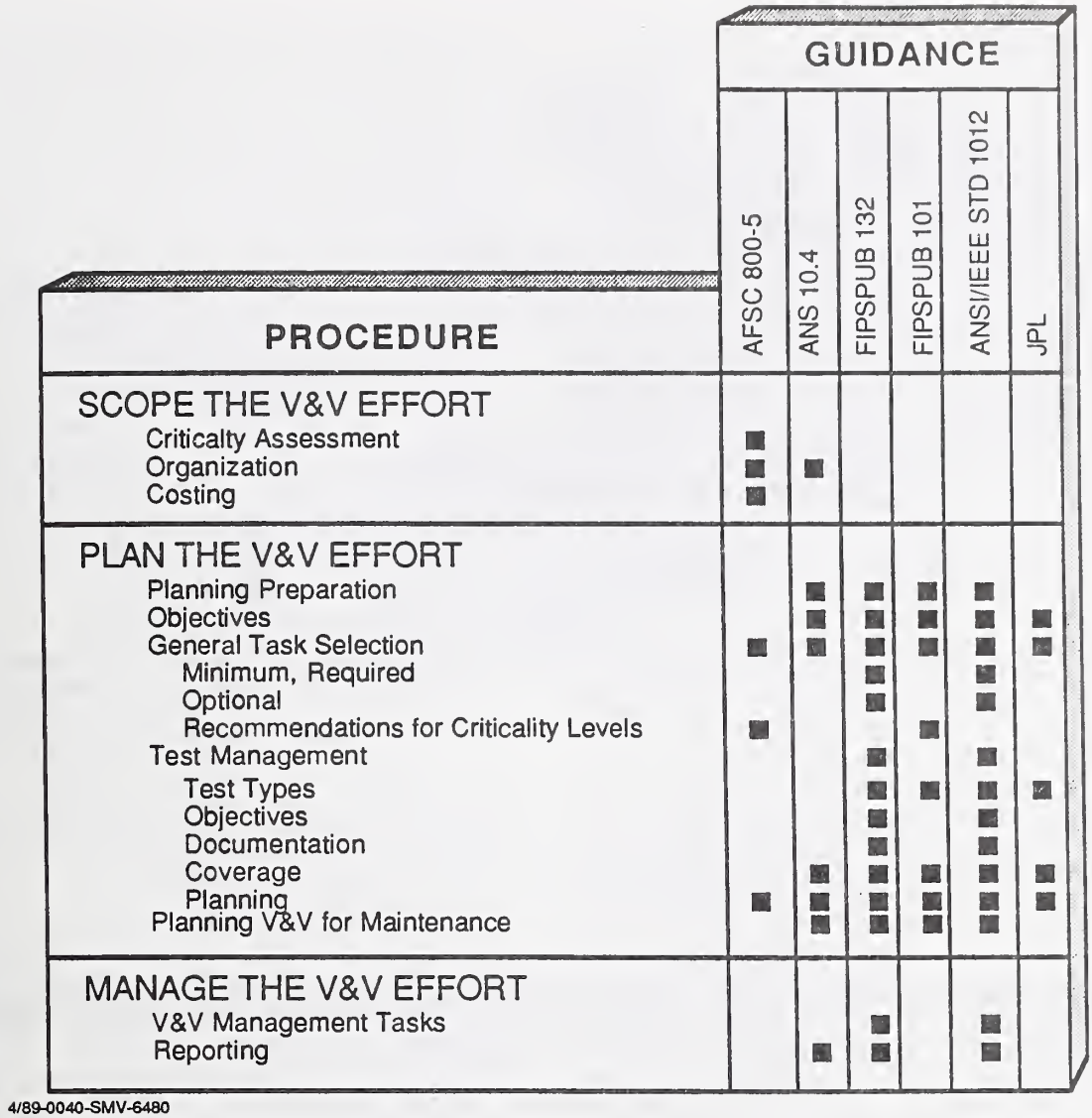

Figure 7. Planning V&V with guidance from V&V documents.

of and allocation of V&V activities among different organizations. In the second case, the standards and guidelines address V&V as a part of project activities. Project management and documentation do not single out activities which are inherently V&V activities but rather include V&V objectives as part of the development task (e.g., ensure that design specifications which are consistent with software requirements specifications).

This report provides a brief overview of how each group of standards relates to V&V activities. Some of the documents fit into both categories because they are used by both the buyer agency for total project management and by the developer for incorporating V&V directly into the project.

4.1 Guidance Requiring V&V

The documents listed in figure 9 provide guidance to determine when and how much software V&V should be performed. In the ANSI/IEEE Standard for Software Project Management Plans (SPMP) [20], project support functions such as quality assurance, configuration management, verification and validation may be specified in a project management plan. The ANSI/IEEE Standard for Software Quality Assurance Plans (SQAP) [21] specifies the minimum documentation requirements, major reviews and audits. The SVVP is one of four documents which are

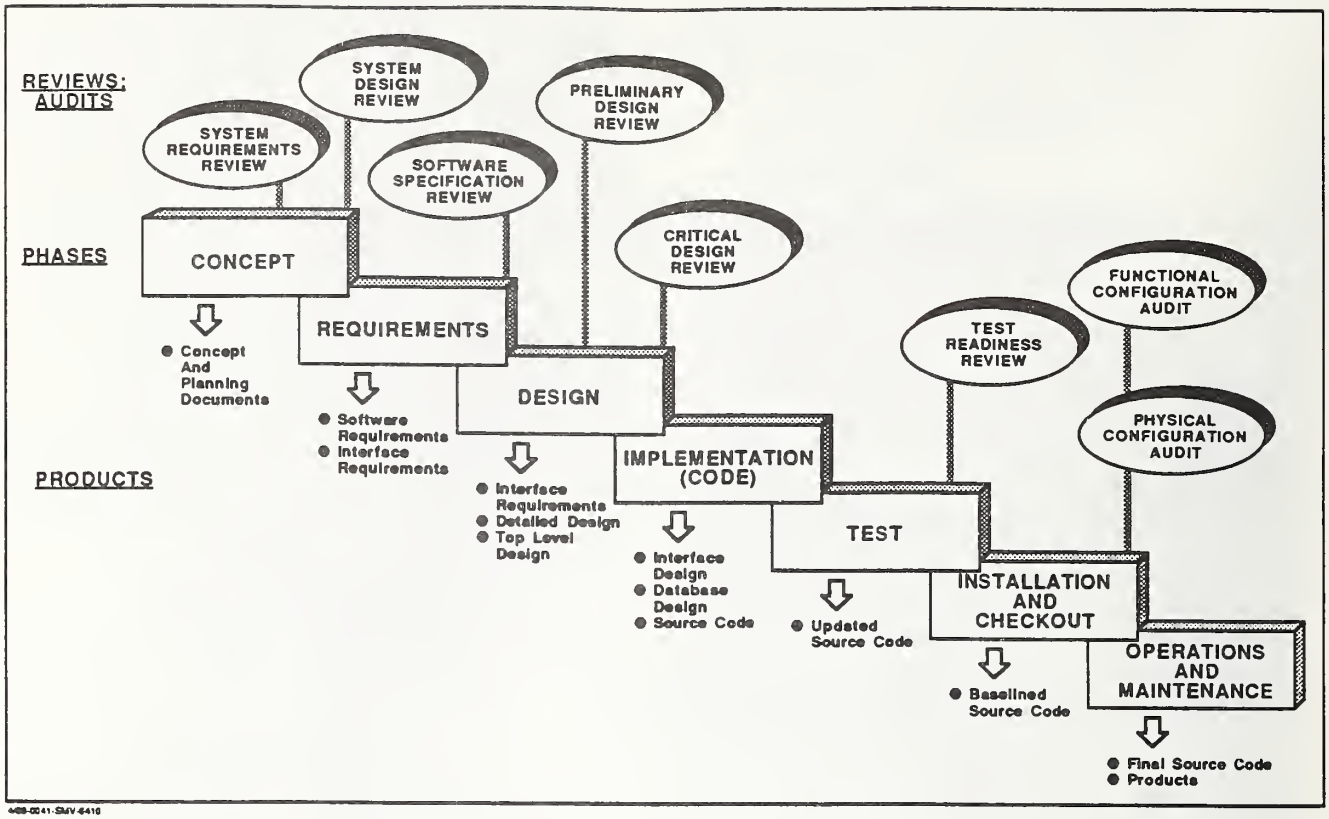

Figure 8. Example of life cycle product and review requirements.

ANSI/IEEE 1058	Standard for Software Project Management Plans (SPMP)
ANSI/IEEE 730	Standard for Software Quality Assurance Plans (SQAP)
NASA SMAP 4.3	Nasa Management Plan Documentation and Data Item Descriptions (DID) Information System and Documentation Standards; Draft Releases
SSPO 30000	Space Station Program Definitions and Requirements (SSP); Draft
DoD-STD-2167A	Military Standard Defense System Software Development
DoD-STD-2168	Military Standard Defense System Software Quality Program

4/89-0042-SMV-6480

Figure 9. Selected guidance for projects requiring V&V.

required to ensure that the implementation of the software satisfies system requirements. Others include the software requirements specification, software design description, the software verification and validation report, user documentation, and software configuration management plan. Reviews and audits include software requirements review, preliminary and critical software design reviews, software V&V review, functional audits, physical audits, in-process audits, and managerial review.

Drafts of NASA documentation standards for information systems [22] have defined basic documentation requirements for management planning, product specifications, assurance specifications, and management control and status reports. The life cycle documentation standards are intended to serve as a model for organizing and executing the management, engineering, and assurance activities of software development and sustaining engineering (maintenance). The draft version, 4.2C of August 1988 [22], provides documentation requirements for all software V&V implementations, whether performed by developer or an independent organization. The general project plan must address how software quality will be assured, and must address software verification and validation for every life cycle phase. A criticality assessment of the project characteristics determines how software V&V will be implemented for the NASA projects. When IV&V is used, it must be defined in the appropriate subsection of the product assurance plan. The published Version 4.3 of the NASA documentation standards was released in February 1989 [22] and contains a format for a verification and validation plan which may be used for internal or independent V&V.

The Space Station Program (SSP) Definition and Requirements Document, June 1988 draft [23], describes Software Product Assurance (SPA) as a technical discipline responsible for requirements, criteria, and the performance of activities to oversee the software safety, reliability, maintainability, and quality assurance. Interaction with an independent software V&V effort is determined and monitored by the appropriate level of software management and development plans. There must be a process to assure that the software life cycle produces reliable and maintainable software. Reliability and maintainability assurance includes V&V tasks (e.g., requirements analysis and requirements traceability analysis, design analysis, fault tolerance analysis, code evaluation and test plan evaluation). The SPA directs that software V&V be performed as directed by a SSP Master Verification Process Requirements document for each SSP element. A Level II IV&V plan, (a document type specific to the SSP) establishes basic IV&V requirements for the SSP.

In DOD-STD-2167A [17], the software development contractor interfaces with the IV&V agent(s) as specified in the contract. Contractors are required to perform evaluations of life cycle phase outputs. Some evaluation criteria which are applicable are internal consistency, understandability, traceability, consistency with various documents, test coverage of requirements, and analyses of coding techniques, timing, and sizing allocations. While a software V&V effort is not required, the evaluation criteria are related to objectives of software V&V activities. Review of software requirements follows system requirements and system design review; software preliminary design and critical design reviews precede system design and critical design reviews. The V&V analyses in the software documentation can provide engineering information to system level reviews on how well the software will meet system requirements.

The "Military Standard Defense System Software Quality Program" DOD-STD-2168 [24], establishes requirements for a software quality program. Contractors determine whether an item or activity meets specified criteria and maintain reports on these findings. Government agencies may serve the role of contractors in performing the software quality program. Many of the evaluation requirements (e.g., product evaluations, certification) can be fulfilled by the activities of software V&V.

Each of these documents requires evaluation of the software products for each life cycle phase, either by the developer or by some other group. The major V&V requirements of these documents are summarized in the following:

- SPMP: V&V addressed in project management plan
- SQAP: SVVP required
- NASA & SSP: V&V must be addressed and is governed by each project's characteristics
- DOD-STD-2167A: software development contractor interfaces with IV&V agent whose role is determined by AF pamphlet; evaluation criteria must be satisfied by contractor and IV&V
- DOD-STD-2168: software quality program and use of V&V determined by contractor.

4.2 Guidance Addressing V&V as Part of Project Development

The standards and guidelines listed in figure 10 require V&V activities as an inherent part of a project's life cycle activities. These activities are not necessarily named as V&V activities but often are named at the specific task level or even are implied because the evaluation objectives are those that are found in the definition of specific V&V tasks.

The Department of Interior developed a set of documents to manage their system life cycle [25]. These consist of a regulatory statement, a handbook, and a detailed guide for project managers. The handbook defines the system life cycle and the major responsibilities and management decision points within that life cycle. Some required activities are those required in the software V&V standards and guidelines (e.g., unit test, system test, database validation, test procedures, user acceptance plan and validation procedures). The handbook states criteria for identifying major acquisitions for applicability. The same criteria could be applied to determine when a V&V effort is applicable. The life cycle requirements complement those of FIPSPUB132/IEEE1012 and can be used together. The Department of Agriculture has adapted the Interior's guidelines for use by their agencies [26].

The draft Canadian standard [27] for software quality assurance uses a significantly different approach by addressing a different level of criticality and type of software in four separate documents. Pre-developed software refers to software prior to issuance of a contract or purchase order. The documentation identifies the following requirements for each type of software:

- critical developed software: requirements and design reviews; test plan, including acceptance test; verification plan, including identification of verification of subcontracted components and subcontractor software quality assurance program; validation requirements to demonstrate compliance with acceptance criteria;
- critical pre-developed software: same as for critical developed software except requirements review is not required;
- non-critical developed software: required verification plan with verification and test activities;
- non-critical pre-developed software: no test plan, verification plan or validation requirements.

The Handbook on Software Quality Assurance for the Nuclear Industry [28] has one chapter on verification and testing. Software V&V is under

DOI	A Project Manager's Guide to Application System Life Cycle Management
DOA	A Project Manager's Guide to Systems Life Cycle Management
CAN	Software Quality Assurance Program; Drafts
NUREG	Handbook of Software Quality Assurance Techniques Applicable to the Nuclear Industry
DoD-STD-2167A	Military Standard Defense System Software Development
DoD-STD-2168	Military Standard Defense System Software Quality Program

4/89-0043-SMV-6480

Figure 10. Selected guidance for projects incorporating V&V.

the responsibility of software quality assurance, although the verification tasks should be performed independent of development. The handbook describes the concepts of verification, general testing and acceptance testing and suggests verification for each life cycle phase and provides checklists for each phase.

NASA, SSP, DOD-STD-2167A and DOD-STD-2168 establish requirements for software evaluation that may be satisfied by an IV&V effort but they also place software quality requirements on the software development contractors themselves. While the intent of the Air Force pamphlet is to determine when independent software V&V is necessary, the contractors can use the AF pamphlet to determine their software V&V requirements. FIPSPUB132/IEEE1012 fits nicely into the next step of planning and implementing a software V&V program within the contractor environment, with additional guidance coming from ANS and JPL.

The key features of guidance documents including V&V as part of the project are the following:

- o DOI, DOA: life cycle management; internal activities of unit test, system test, test procedures, database validation, user acceptance plan, and validation procedures.
- o CAN: verification and test activities and performing agent determined by criticality level; separate standards.
- o NUREG: independence recommended; descriptions of V&V techniques, testing, and checklists for each life cycle phase.
- o NASA, SSP, DOD-STD-2167A, 2168: software development contractors have responsibility for internal software quality activities; software requirements specified for each life cycle phase.

5.0 CONCLUSIONS

Software engineering technology has matured sufficiently to be addressed in approved and draft software engineering standards and guidelines. Many of these documents address project level requirements for reviews to ensure satisfactory progress at interim steps along the life cycle. Standards for software V&V require activities which produce the information that management needs to decide whether or not to allow the project to progress to the next development step and at completion whether or not to accept the product. V&V coexists with other quality engineering disciplines and complements many of the software engineering disciplines. A major difference between V&V and other quality engineering functions is that, like the developer's activities, V&V activities examine the software in detail from a systems viewpoint. Results from V&V analyses and tests can supply systems engineering data for every review and audit required by general project standards.

From this study of standards and guidelines, it can be seen that the V&V guidance documents can be used to complement the requirements of the project level documents.

United States businesses and industries, along with Federal agencies, spend billions annually on computer software in many of their functions:

- o to manufacture their products,
- o to provide their services,
- o to administer their daily activities,
- o to perform their short and long term management functions.

As with other products, industries and businesses are discovering that their increasing dependence on computer technology to perform these functions emphasizes the need for safe, secure, reliable computer systems. They are recognizing that software quality and reliability are vital to the U.S.'s ability to maintain its competitiveness and high technology posture in the marketplace. V&V is one of several methodologies that can be used for building vital quality software.

6.0 REFERENCES

[1] Branstad, Martha and Patricia B. Powell, "Software Engineering Project Standards," IEEE Transactions on Software Engineering, January 1984, Vol. SE-10, No. 1, pp. 73-78, The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017.

[2] Nash, Sarah H., and Samuel T. Redwine, Jr., "Information Interface Related Standards, Guidelines, and Recommended Practices," IDA PAPER P-1842, Institute for Defense Analyses, 1801 N. Beaugard Street, Alexandria, VA 22311, 1985.

[3] CAN-CSA-Q396.1, Software Quality Assurance Program, Canadian Standards Organization, 178 Rexdale, Toronto, Ontario, Canada M9W 1R3, 1988.

[4] Dorling, A., "Activities Relating to the Development of Standards Relevant to Software Quality," ISO/IEC/JTC1/SC7 #575, ISO/IEC/JTC1/SC7 Plenary, The Hague, June 13-17, 1988.

[5] ANSI/IEEE Std.1012-1986, "Standard for Software Verification and Validation Plans," The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017, November 1986.

[6] Boehm, B.W., "A Spiral Model of Software Development and Enhancement," Computer, May 1988, The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017.

[7] Davis, A.M., E.H. Bersoff, and E.R. Comer, "A Strategy for Comparing Alternative Software Development Life Cycle Models," IEEE Transactions on Software Engineering, Vol. 14, No. 10, pp. 1453-1461, October 1988, The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017.

[8] IEEE P1074, "DRAFT Standard for Software Life Cycle Processes," The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017.

[9] Radatz, J.W.,"Analysis of IV&V Data," RADC-TR-81-145, Logicon, Inc., Rome Air Development Center, Griffiss AFB, NY, June 1981.

[10] Boehm, B.W., and P.N. Papaccio, "Understanding and Controlling Software Costs," IEEE Transactions on Software Engineering, Oct. 1988, The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017.

[11] Powell, Patricia B., "Software Validation, Verification and Testing Technique and Tool Reference Guide," National Bureau of Standards Special Publication 500-93, National Bureau of Standards, Gaithersburg, MD 20899, 1982.

[12] Adrion, W.R., M.A. Branstad, and J.C. Cherniavsky, "Validation, Verification, and Testing of Computer Software," ACM Computing Surveys, Vol. 14, No.2, June 1982.

[13] McGarry, F., and G. Page, "Performance Evaluation of an Independent Software Verification and Integration Process," NASA Goddard, Greenbelt, MD, SEL 81 -110, September 1982.

[14] "Guideline for Lifecycle Validation, Verification and Testing of Computer Software," FIPSPUB101, National Bureau of Standards, Gaithersburg, MD 20899, 1983.

[15] "Guideline for Software Verification and Validation Plans," FIPSPUB132, National Bureau of Standards, Gaithersburg, MD 20899, 1987.

[16] AFSC/AFLCP 800-5 Air Force Systems Command and Air Force Logistics Command Software Independent Verification and Validation, Washington, DC, 22 May 1988.

[17] DOD-STD-2167A Military Standard Defense System Software Development, AMSC No. 4327, Department of Defense, Washington, DC, February 29, 1988.

[18] ANSI/ANS-10.4-1987, "Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry," American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60525, 1987.

[19] Blosiu, Julian O., "Independent Verification and Validation of Computer Software: Methodology," National Aeronautics and Space Administration, Jet Propulsion Laboratory, Pasadena, CA, JPL D-576, February 9, 1983.

[20] ANSI/IEEE Std. 1058-1987, Standard for Software Project Management," The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017, 1987.

[21] ANSI/IEEE Std. 730-1984, "Standard for Software Quality Assurance Plans," The Institute for Electrical and Electronics Engineers, Inc., 345 West 47th St., New York, NY 10017, 1984.

[22] Information System Life-Cycle and Documentation Standards and Management Plan Documentation and Data Item Descriptions (DID); Releases 4.2a,b,c (Draft -1988), Release 4.3 (February 1989), NASA Headquarters, Washington, DC.

[23] Space Station Support, Space Station Program Definition and Requirements Document, Section 9.5, SSPO 30000, NASA Headquarters, Washington, DC (Draft -June 1988).

[24] DOD-STD-2168 Military Standard Defense System Software Quality Program, AMSC No. A4389, Department of Defense, Washington, DC, April 1988.

[25] Department of the Interior, "A Project Manager's Guide to Application Systems Life Cycle Management"; "Departmental Manual Part 376 DM 10 and Application System Life Cycle Management Handbook," Washington, DC, August 1985.

[26] Department of Agriculture, "A Project Manager's Guide to Application Systems Life Cycle Management," DM 3200-2; "Application Systems Life Cycle Management Manual," DM 3200-1 and "Software Management," DM 3220-3, Washington, DC, March 1988.

[27] Canadian Standards Association, "A General Guide on Procurement Quality Assurance Standards and Organizations," CAN/CSA Q396 series, Draft, 178 Rexdale, Toronto, Ontario, Canada M9W 1R3, 1988.

[28] NUREG/CR-4640 "Handbook of Software Quality Assurance Techniques Applicable to the Nuclear Industry," Prepared by J. L. Bryant and N.P. Wilburn, Pacific Northwest Laboratory, PNL-5784 for U. S. Nuclear Regulatory Commission, Washington, DC, August 1987.

☆U.S.GOVERNMENT PRINTING OFFICE:1989-242-200/00055

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NIST/SP-500/165	2. Performing Organ. Report No.	3. Publication Date September 1989

4. TITLE AND SUBTITLE

Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards

5. AUTHOR(S) Dolores R. Wallace and Roger U. Fujii

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

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
(formerly NATIONAL BUREAU OF STANDARDS)
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered NIST-SP (July 88 - May 89)

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

National Institute of Standards and Technology
Gaithersburg, MD 20899

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 89-600754

☐ 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)

Software verification and validation (V&V) is a broad systems engineering approach to ensure software quality and gain optimum software performance. V&V supports the requirements for project management and quality assurance. When used with other software engineering standards, V&V helps to produce safe, secure, reliable, and maintainable software programs.

This report describes how the software verification and validation methodology and V&V standards provide a strong framework for developing quality software. First, the report describes software V&V, its objectives, recommended tasks, and guidance for selecting techniques to perform V&V. It explains the difference between V&V and quality assurance, development system engineering, and user organization functions. The report explains that V&V produces maximum benefits when it is performed independent of development functions and provides a brief discussion of how V&V benefits change when embedded in quality assurance, development systems engineering, and user organizations. An analysis of two studies of V&V's cost-effectiveness concludes that cost benefits of V&V's early error detection outweigh the cost of performing V&V.

Next the report describes several software engineering standards for V&V, project management, and quality assurance. The report describes each V&V standard according to its V&V requirements and techniques. Then the report provides an overview description of project management and quality assurance standards and explains how the V&V standards may be used along with them. The report provides insights on how to use management, quality, and V&V techniques and methodology to structure a quality software development.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)
Keywords: computer assurance; evaluation; project management; software development; software engineering; software maintenance; software management; software safety; software security; software standards; software testing; software verification and validation.

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 37

15. Price

USCOMM-DC 6043-P80

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899