IEEE Std 1012-1998
(Revision of IEEE Std 1012-1986)

IEEE Standard for Software Verification and Validation

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Approved 9 March 1998

IEEE-SA Standards Board

Abstract: Software verification and validation (V&V) processes, which determine whether development products of a given activity conform to the requirements of that activity, and whether the software satisfies its intended use and user needs, are described. This determination may include analysis, evaluation, review, inspection, assessment, and testing of software products and processes. V&V processes assess the software in the context of the system, including the operational environment, hardware, interfacing software, operators, and users.
Keywords: software integrity, software life cycle processes, verification and validation

---

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 0-7381-0196-6

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

(This introduction is not part of IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation.)

Software verification and validation (V&V) is a technical discipline of systems engineering. The purpose of software V&V is to help the development organization build quality into the software during the software life cycle. The software V&V processes determine if development products of a given activity conform to the requirements of that activity, and if the software satisfies the intended use and user needs. The determination includes assessment, analysis, evaluation, review, inspection, and testing of software products and processes. The software V&V is performed in parallel with the software development, not at the conclusion of the software development.

The software V&V is an extension of the program management and systems engineering team, and undertakes its determination to identify objective data and conclusions (i.e., proactive in its feedback) about software quality, performance, and schedule compliance for the development organization. This feedback consists of anomalies, performance improvements, and quality improvements against not only the expected operating conditions but across the full operating spectrum of the system and its interfaces. Early feedback results allow the development organization to modify the software products in a timely fashion and reduce overall project cost and schedule impacts. Without a proactive approach, the anomalies and the associated software changes are typically delayed to later in the program schedule, resulting in greater program cost and schedule delays.

IEEE Std 1012-1986 was a product standard that defined the contents of the Software Verification and Validation Plan (SVVP). This revision of the standard, IEEE Std 1012-1998, is a process standard that defines the verification and validation processes in terms of specific activities and related tasks. IEEE Std 1012-1998 also defines the contents of the SVVP including example format.

This standard introduces the following key concepts:

— *Software integrity levels.* Defines four software integrity levels to describe the criticality of the software varying from high integrity to low integrity.

— *Minimum V&V tasks for each software integrity level.* Defines the minimum V&V tasks required for each of the four software integrity levels. Includes a table of optional V&V tasks as a method of allowing the user to tailor the V&V effort to address project needs and application specific characteristics.

— *Intensity and rigor applied to V&V tasks.* Introduces the notion that the intensity and rigor applied to the V&V tasks vary according to the software integrity level. Higher software integrity levels require the application of greater intensity and rigor to the V&V task. Intensity includes greater scope of analysis across all normal and abnormal system operating conditions. Rigor includes more formal techniques and recording procedures.

— *Detailed criteria for V&V tasks.* Defines specific criteria for each V&V task including minimum criteria for correctness, consistency, completeness, accuracy, readability, and testability. The V&V task descriptions include a list of the required task inputs and outputs.

— *Systems viewpoint.* Adds minimum V&V tasks to address systems issues. These V&V tasks include Hazard Analysis, Risk Analysis, Migration Assessment, and Retirement Assessment. Specific systems issues are contained in individual V&V task criteria.

— *Compliance with International and IEEE standards.* Defines the V&V processes to be compliant with the life cycle process standards such as ISO/IEC Std 12207, IEEE Std 1074-1997, and IEEE/EIA Std 12207.0-1996, as well as the entire family of IEEE software engineering standards. This standard addresses the full software life cycle processes including acquisition, supply, development, operation, and maintenance.

The following persons were on the working group:

Roger U. Fujii, *Chair*

Delores R. Wallace, *Vice Chair*

Donald W. Sova, *Secretary*

Richard J. Blauw
Robert Brill
Robert Butler
John G. Capen
Robert Charette
Hu Cheng
John Chilenski
François Coallier
Darrell Cooksey
Ben Conger
Geoff Cozens
Paul R. Croll
H. Taz Daughtrey
Harpal S. Dhama
Lisa J. Downey
James Dukelow
Robert G. Ebenau
Vera Edelstein
Caroline L. Evans
Richard L. Evans
George Finelli
Kirby K. Fortenberry
Eva Freund
Nicholas P. Ginex
Stephen Harris
John W. Horch
Laura M. Ippolito
George Jackelen
William Jackson
Lisa A. Jensen
Barry S. Johnson
Q. Leon Jordan
Kathryn Kemp
J. Dennis Lawrence
Jeffrey Lewis
Victor J. Maggioli
Kartik C. Majumdar
John R. Matras
Tomoo Matsubara
Randall May
Marco Migliaro
Warren L. Persons
Ian C. Pyle
W. Jim Rice
Paul J. Rodi
Uma D. Satyen
John A. Scott
Grant Shen
James Stanfield
Nancy E. Sunderland
Gina To
Leonard L. Tripp
Michael E. Waterman

The following persons were on the balloting committee:

Mikhail Auguston
Dennis Beauchaine
Leo Beltracchi
Mordechai Ben-Menachem
H. Ronald Berlack
Richard E. Biehl
William J. Boll
Juris Borzovs
Edward R. Byrne
James E. Cardow
Keith Chan
John J. Chilenski
Antonio M. Cicu
Theo Clarke
Sylvain Clermont
François Coallier
Rosemary Coleman
Darrell Cooksey
Geoff Cozens
Paul R. Croll
Gregory T. Daich
M. A. Daniels
Taz Daughtrey
Bostjan K. Derganc
Perry R. DeWeese
Harpal Dhama
Sherman Eagles
Robert G. Ebenau
Leo Egan
Richard L. Evans
William Eventoff
Jonathan H. Fairclough
John W. Fendrich
Julian Forster
Kirby K. Fortenberry
Eva Freund
Roger U. Fujii
Simon Gabrihelidis
Barry L. Garner
Adel N. Ghannam
Hiranmay Ghosh
Marilyn Ginsberg-Finner
Eugene A. Glasser
John Garth Glynn
Julio Gonzalez-Sanz
L. M. Gunther
David A. Gustafson
John Harauz
Herbert Hecht
Manfred Hein
Gordon Henley
Mark Henley
John W. Horch
Jerry Huller
Peter L. Hung
George Jackelen
Lisa A. Jensen
Q. Leon Jordan
Vladan V. Jovanovic
William S. Junk
George X. Kambic
Myron S. Karasik
Ron S. Kenett
Robert J. Kierzyk
Shaye Koenig
Thomas M. Kurihara
John B. Lane
J. Dennis Lawrence
Fang Ching Lim
Victor J. Maggioli
David Maibor
Kartik C. Majumdar
Henry A. Malec
John R. Matras
Tomoo Matsubara
Mike McAndrew
Sue McGrath
Jerome W. Mersky
Bret Michael
Lance Miller
Alan Miller
Lisa Ming
Celia H. Modell
Pavol Navrat
Myrna L. Olson
Mike Ottewill
Indradeb P. Pal
Lalit M. Patnaik
Warren L. Persons
John G. Phippen
Alex Polack
Peter T. Poon
Kenneth R. Ptack
Ian C. Pyle
Annette D. Reilly
Christian Reiser
Dennis Rilling
Helmut Sandmayr
Uma D. Satyen
Stephen R. Schach
Hans Schaefer
Norman Schneidewind
David J. Schultz
Gregory D. Schumacher
Carl S. Seddio
Robert W. Shillato
David M. Siefert
Carl A. Singer
James M. Sivak
Alfred R. Sorkowitz
Donald W. Sova
Julia Stesney
Norma Stopyra
Fred J. Strauss
Robert N. Sulgrove
John Swearingen
Booker Thomas
Gina To
Patricia Trellue
T. H. Tse
Theodore J. Urbanowicz
Glenn D. Venables
Udo Voges
Dolores Wallace
Camille S. White-Partain
Scott A. Whitmire
P. A. Wolfgang
Paul R. Work
Kathryn P. Yglesias
Natalie C. Yopconka
Weider D. Yu
Geraldine Zimmerman

When the IEEE-SA Standards Board approved this standard on 19 March 1998, it had the following membership:

Richard J. Holleman, *Chair* **Donald N. Heirman,** *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon
James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry
L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Kristin M. Dittmann
IEEE Standards Project Editor

Contents

IEEE Standard for Software Verification and Validation

1. Overview

Software verification and validation (V&V) processes determine whether development products of a given activity conform to the requirements of that activity, and whether the software satisfies its intended use and user needs. This determination may include analysis, evaluation, review, inspection, assessment, and testing of software products and processes. V&V processes assess the software in the context of the system, including the operational environment, hardware, interfacing software, operators, and users.

This V&V standard is a process standard that addresses all software life cycle processes, including acquisition, supply, development, operation, and maintenance. This standard is compatible with all life cycle models. Not all life cycle models use all of the life cycle processes listed in this standard.

The user of this standard may invoke those software life cycle processes and the associated V&V processes that apply to the project. A description of the software life cycle processes may be found in ISO/IEC 12207 [B16][1], IEEE Std 1074-1997 [B12], and IEEE/EIA Std 12207.0-1996 [B13]. Annex A maps ISO/IEC 12207 (Tables A.1 and A.2) and IEEE Std 1074-1997 (Table A.3) to the V&V activities and tasks defined in this standard.

1.1 Purpose

The purpose of this standard is to

1) Establish a common framework for V&V processes, activities, and tasks in support of all software life cycle processes, including acquisition, supply, development, operation, and maintenance processes.
2) Define the V&V tasks, required inputs, and required outputs.
3) Identify the minimum V&V tasks corresponding to software integrity levels using a four-level scheme.
4) Define the content of a Software V&V Plan (SVVP).

[1]The numbers in brackets preceded by the letter B correspond to those of the "other references" listed in Annex H.

1.2 Field of application

This standard applies to software being developed, maintained, and reused (See Annex D for a description of V&V of reusable software). The term *software* also includes firmware, microcode, and documentation.

Software is a key component that contributes to system behavior and performance. This relationship requires that software V&V processes must take software interactions with all system components into consideration. The user of this standard should consider V&V as part of the software life cycle processes defined by industry standards such as ISO/IEC 12207 [B16], IEEE Std 1074-1997 [B12], or IEEE/EIA Std 12207.0-1996 [B13].

1.3 V&V objectives

V&V processes provide an objective assessment of software products and processes throughout the software life cycle. This assessment demonstrates whether the software requirements and system requirements (i.e., those allocated to software) are correct, complete, accurate, consistent, and testable. Other objectives of performing V&V are to

1) Facilitate early detection and correction of software errors;
2) Enhance management insight into process and product risk; and
3) Support the software life cycle processes to ensure compliance with program performance, schedule, and budget requirements.

The verification process provides supporting evidence that the software and its associated products

1) Comply with requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance);
2) Satisfy standards, practices, and conventions during life cycle processes; and
3) Establish a basis for assessing the completion of each life cycle activity and for initiating other life cycle activities.

The validation process provides supporting evidence that the software satisfies system requirements allocated to software, and solves the right problem (e.g., correctly models physical laws, or implements system business rules).

V&V support primary life cycle processes. V&V processes are most effective when conducted in parallel with software development processes; otherwise, V&V objectives may not be realized. In this standard, V&V processes are discussed together because the V&V activities and tasks are interrelated and complementary. In some circumstances, the verification process may be viewed as a process separate from the validation process. The V&V task criteria described in Table 1 (starting on page 25) uniquely define the compliance requirements for V&V processes.

1.4 Organization of the standard

This standard is organized into clauses (Clauses 1 through 7), tables (Tables 1 through 3), figures (Figures 1 through 3), and annexes (Annexes A through I). Clause 1, Figures 1, 2, and 3, and Table 3 contain informative material that provides illustrations, examples, and process flow diagrams useful in understanding and using this standard. Clauses 2, 3, 4, 5, 6, and 7 and Tables 1 and 2 contain the mandatory V&V requirements for this standard. All annexes contain informative material except Annex I.

Clause 2 lists normative references. Clause 3 provides a definition of terms, abbreviations, and conventions. Clause 4 explains the concept of using software integrity levels for determining the scope and rigor of V&V

processes. Clause 5 describes each primary software life cycle process and lists the V&V activities and tasks associated with the life cycle process. Clause 6 describes the V&V reporting, administrative, and documentation requirements. Clause 7 outlines the content of a Software Verification and Validation Plan (SVVP).

Tables 1, 2, and 3 are the focal point of this standard, containing detailed V&V process, activity, and task requirements. Table 1 provides V&V task descriptions, inputs, and outputs for each life cycle process. Table 2 lists minimum V&V tasks required for different software integrity levels. Table 3 provides a list of optional V&V tasks and their suggested applications in the life cycle. These optional V&V tasks may be added to the minimum V&V tasks to tailor the V&V effort to project needs and application specific characteristics.

Figure 1 provides an example of an overview of the V&V inputs, outputs, and minimum V&V tasks for the highest software integrity level (Integrity Level 4). Figure 2 provides guidelines for scheduling V&V test planning, execution, and verification activities. An example of a phased life cycle model was used in Figures 1 and 2 to illustrate a mapping of the ISO/IEC 12207 life cycle processes to the V&V activities and tasks described in this standard.

This standard implements the V&V framework using the terminology of process, activity, and task. Figure 3 illustrates how the V&V processes are subdivided into activities, which in turn have associated tasks. Hereafter, the term *V&V effort* is used to refer to the framework of the V&V processes, activities, and tasks.

The annexes contain informative and normative information useful to implementing the requirements of this standard. Annex A (informative) describes the mapping of ISO/IEC 12207 and IEEE Std 1074-1997 V&V requirements to this standard's V&V activities and tasks. Annex B (informative) provides an example of a risk-based, four-level integrity scheme. Annex C (informative) provides a definition of independent verification and validation (IV&V). Annex D (informative) provides guidelines for conducting V&V of reusable software. Annex E (informative) describes V&V metrics for assessing V&V quality, V&V coverage, and software development processes and products. Such V&V metrics support process improvement tasks of project management. Annex F (informative) illustrates an example of the V&V organizational relationship to other project responsibilities. Annex G (informative) describes optional V&V tasks. Annex H (informative) lists standards and guides that may be useful in interpreting and implementing the V&V tasks identified in this standard. Annex I (normative) contains definitions from existing standards.

1.5 Audience

The audience for this standard is software suppliers, acquirers, developers, maintainers, V&V practitioners, operators, and managers in both the supplier and acquirer organizations.

1.6 Compliance

The word *shall* identifies mandatory requirements to claim compliance with this standard. The words *should* or *may* indicate optional tasks that are not required to claim compliance to this standard.

Any software integrity level scheme may be used with this standard. The software integrity level scheme used in this standard is not mandatory, but rather, establishes the minimum V&V tasks for the referenced software integrity scheme. To demonstrate compliance to this standard whenever different software integrity schemes are used, the user should map the project-specific software integrity scheme to the integrity scheme used in this standard. This mapping establishes the minimum V&V tasks that should be assigned to the project. Compliance with this standard requires that this mapping and the associated minimum V&V tasks be documented in the SVVP.

Not all V&V efforts are initiated at the start of the life cycle process of acquisition and continued through the maintenance process. If a project uses only selected life cycle processes, then compliance with this standard is achieved if the minimum V&V tasks are implemented for the associated life cycle processes selected for

the project. As in all cases, the minimum V&V tasks are defined by the software integrity level assigned to the software. For life cycle processes that are not used by the project, the V&V requirements and tasks for those life cycle processes are optional V&V tasks invoked as needed at the discretion of the project. Specific software development methods and technologies (such as automated code generation from detailed design) may eliminate development steps or combine several development steps into one. Therefore, a corresponding adaptation of the minimum V&V tasks is permitted.

When this standard is invoked for existing software and the required V&V inputs are not available, then V&V tasks may use other available project input sources or may reconstruct the needed inputs to achieve compliance with this standard.

1.7 Disclaimer

This standard establishes minimum criteria for V&V processes, activities, and tasks. The implementation of these criteria does not, however, automatically ensure compliance to system or mission objectives, or prevent adverse consequences (e.g., loss of life, mission failure, loss of system safety or security, financial or social loss). Compliance with this standard does not absolve any party from any social, moral, financial, or legal obligations.

1.8 Limitations

None.

2. Normative references

This standard does not require the use of any normative references. Other standards considered to be useful in the implementation and interpretation of this standard are listed in Annex H.

3. Definitions, abbreviations, and conventions

3.1 Definitions

The following terms, including those defined in other standards, are used as indicated in this standard. Annex I contains definitions taken from other existing standards.

3.1.1 acceptance testing: Testing conducted in an operational environment to determine whether a system satisfies its acceptance criteria (i.e., initial requirements and current needs of its user) and to enable the customer to determine whether to accept the system.

3.1.2 anomaly: See Annex I.

3.1.3 component testing: Testing conducted to verify the correct implementation of the design and compliance with program requirements for one software element (e.g., unit, module) or a collection of software elements.

3.1.4 criticality: A subjective description of the intended use and application of the system. Software criticality properties may include safety, security, complexity, reliability, performance, or other characteristics.

3.1.5 criticality analysis: A structured evaluation of the software characteristics (e.g., safety, security, complexity, performance) for severity of impact of system failure, system degradation, or failure to meet software requirements or system objectives.

3.1.6 hazard: See Annex I.

3.1.7 hazard analysis: A systematic qualitative or quantitative evaluation of software for undesirable outcomes resulting from the development or operation of a system. These outcomes may include injury, illness, death, mission failure, economic loss, property loss, environmental loss, or adverse social impact. This evaluation may include screening or analysis methods to categorize, eliminate, reduce, or mitigate hazards.

3.1.8 hazard identification: See Annex I.

3.1.9 independent verification and validation (IV&V): V&V processes performed by an organization with a specified degree of technical, managerial, and financial independence from the development organization.

3.1.10 integration testing: An orderly progression of testing of incremental pieces of the software program in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated to show compliance with the program design, and capabilities and requirements of the system.

3.1.11 integrity level: See Annex I.

3.1.12 interface design document (IDD): Documentation that describes the architecture and design of interfaces between system and components. These descriptions include control algorithms, protocols, data contents and formats, and performance.

3.1.13 interface requirement specification (IRS): Documentation that specifies requirements for interfaces between systems or components. These requirements include constraints on formats and timing.

3.1.14 life cycle process: A set of interrelated activities that result in the development or assessment of software products. Each activity consists of tasks. The life cycle processes may overlap one another. For V&V purposes, no process is concluded until its development products are verified and validated according to the defined tasks in the SVVP.

3.1.15 minimum tasks: Those V&V tasks required for the software integrity level assigned to the software to be verified and validated.

3.1.16 optional tasks: Those V&V tasks that may be added to the minimum V&V tasks to address specific application requirements.

3.1.17 required inputs: The set of items necessary to perform the minimum V&V tasks mandated within any life cycle activity.

3.1.18 required outputs: The set of items produced as a result of performing the minimum V&V tasks mandated within any life cycle activity.

3.1.19 risk: See Annex I.

3.1.20 risk analysis: See Annex I.

3.1.21 software design description (SDD): A representation of software created to facilitate analysis, planning, implementation, and decision making. The software design description is used as a medium for communicating software design information, and may be thought of as a blueprint or model of the system.

3.1.22 software integrity levels: See Annex I.

3.1.23 software requirements specification (SRS): Documentation of the essential requirements (i.e., functions, performance, design constraints, and attributes) of the software and its external interfaces. The software requirements are derived from the system specification.

3.1.24 software verification and validation plan (SVVP): A plan describing the conduct of software V&V.

3.1.25 software verification and validation report (SVVR): Documentation of V&V results and software quality assessments.

3.1.26 system testing: The activities of testing an integrated hardware and software system to verify and validate whether the system meets its original objectives.

3.1.27 test case: Documentation that specifies inputs, predicted results, and a set of execution conditions for a test item.

3.1.28 test design: Documentation that specifies the details of the test approach for a software feature or combination of software features and identifying the associated tests.

3.1.29 test plan: Documentation that specifies the scope, approach, resources, and schedule of intended testing activities.

3.1.30 test procedure: Documentation that specifies a sequence of actions for the execution of a test.

3.1.31 validation: See Annex I.

3.1.32 verification: See Annex I.

3.2 Abbreviations

The following abbreviations appear in this standard:

COTS	Commercial-Off-The-Shelf
IDD	Interface Design Document
IEC	International Electrotechnical Commission
IRS	Interface Requirements Specification
ISO	International Organization for Standardization
IV&V	Independent Verification and Validation
RFP	Request for Proposal (-tender)
SDD	Software Design Description
SRS	Software Requirements Specification
SVVP	Software Verification and Validation Plan
SVVR	Software Verification and Validation Report
V&V	Verification and Validation

3.3 Conventions

The term "documentation" refers to information that may exist in several documents or may be embedded within a document addressing more than one subject. "Documentation" includes data in electronic format, and may be in narrative, tabular, or graphic form (e.g., format of Figure 1).

4. V&V software integrity levels

4.1 Software integrity levels

Software exhibits different criticality based upon its intended use and application of the system to critical or noncritical uses. Some software systems affect critical, life-sustaining systems, while other software systems are noncritical, standalone research tools. Software criticality is a description of the intended use and application of a system. This standard uses a software integrity level approach to quantify software criticality. Software integrity levels denote a range of software criticality values necessary to maintain risks within acceptable limits. These software properties may include safety, security, software complexity, performance, reliability, or other characteristics. Critical, high-integrity software typically requires a larger set and more rigorous application of V&V tasks.

For planning the V&V processes, software integrity levels are generally assigned early in the development process, preferably during the system requirements analysis and architecture design activities. The software integrity level can be assigned to software requirements, functions, group of functions, or software components or subsystems. The assigned software integrity levels may vary as the software evolves. Design, coding, procedural, and technology implementation features selected by the development organization can raise or lower the software criticality and the associated software integrity levels assigned to the software. Risk mitigation approaches acceptable to the acquirer also may be used to reduce software criticality, thus allowing the selection of a lower integrity level. The software integrity level assignment is continually updated and reviewed by conducting the V&V criticality analysis task throughout the software development process.

This standard does not mandate the use of the software integrity scheme referenced in this standard. The user of this standard may select any software integrity scheme (such as from existing standards) that defines the requirements for assigning software integrity levels. The software integrity levels established for a project result from agreements among the acquirer, supplier, developer, and independent assurance authorities (e.g., a regulatory body or responsible agency). The V&V effort shall specify a software integrity scheme if one is not already defined.This standard shall use the following four-level software integrity scheme as a method to define the minimum V&V tasks that are assigned to each software integrity level:

Criticality	Description	Level
High	Selected function affects critical performance of the system.	4
Major	Selected function affects important system performance.	3
Moderate	Selected function affects system performance, but workaround strategies can be implemented to compensate for loss of performance.	2
Low	Selected function has noticeable effect on system performance but only creates inconvenience to the user if the function does not perform in accordance with requirements.	1

To identify the minimum V&V tasks that apply to a different selected software integrity level scheme, the user of the standard shall map this standard's software integrity scheme and associated minimum V&V tasks

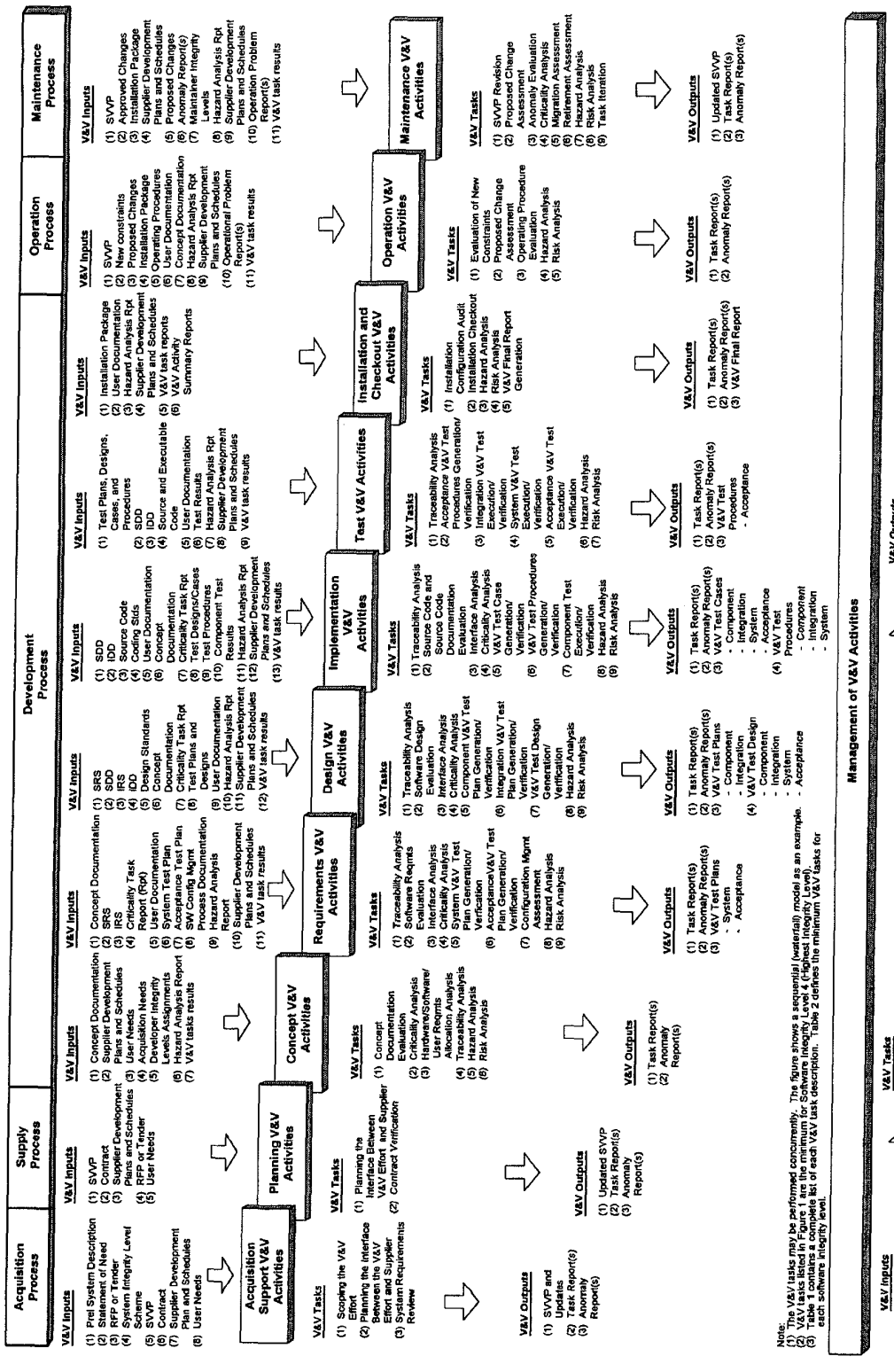


Figure 1—An example of software V&V overview

to their selected software integrity level scheme. The mapping of the software integrity level scheme and the associated minimum V&V tasks shall be documented in the SVVP. An example of a risk-based software integrity level scheme is illustrated in Annex B.

This standard does not apply to those portions of the software for which none of the software integrity criteria apply (i.e., those software portions below level 1). The basis for assigning software integrity levels to software components shall be documented in a V&V Task Report and V&V Final Report.

The integrity level assigned to reusable software shall be in accordance with the integrity level scheme adopted for the project (see Annex D), and the reusable software shall be evaluated for use in the context of its application.

The V&V processes are tailored to specific system requirements and applications through the selection of a software integrity level with its corresponding minimum V&V tasks and the addition of optional V&V tasks. The addition of optional V&V tasks allows the V&V effort to address application specific characteristics of the software.

5. V&V processes

V&V processes support the management process (5.1), acquisition process (5.2), supply process (5.3), development process (5.4), operation process (5.5), and maintenance process (5.6). The minimum V&V activities and tasks supporting the above processes are referenced in the following subclauses and defined in Table 1. This clause's subtitles are the same as subtitles in Table 1 to correlate the requirements of the following subclauses with Table 1 tasks.

The V&V effort shall comply with the task descriptions, inputs, and outputs as described in Table 1. The V&V effort shall perform the minimum V&V tasks as specified in Table 2 for the assigned software integrity level. If the user of this standard has selected a different software integrity level scheme, then the mapping of that integrity level scheme to Table 2 shall define the minimum V&V tasks for each of the user's software integrity levels.

Not all software projects include each of the life cycle processes listed above. To be in compliance with this standard, the V&V processes shall address all those life cycle processes used by the software project.

Some V&V activities and tasks include analysis, evaluations, and tests that may be performed by multiple organizations (e.g., software development, project management, quality assurance, V&V). For example, risk analysis and hazard analysis are performed by project management, the development organization, and the V&V effort. The V&V effort performs these tasks to develop the supporting basis of evidence showing whether the software product satisfies its requirements. These V&V analyses are complementary to other analyses and do not eliminate or replace the analyses performed by other organizations. The degree to which these analyses efforts are coordinated with other organizations shall be documented in the organizational responsibility section of the SVVP.

The user of this standard shall document the V&V processes in the SVVP and shall define the information and facilities necessary to manage and perform these processes, activities, and tasks, and to coordinate those V&V processes with other related aspects of the project. The results of V&V activities and tasks shall be documented in task reports, activity summary reports, anomaly reports, V&V test documents, and the V&V Final Report.

5.1 Process: Management

The management process contains the generic activities and tasks, which may be employed by any party that manages its respective processes. The management tasks are to 1) prepare the plans for execution of the process, 2) initiate the implementation of the plan, 3) monitor the execution of the plan, 4) analyze problems discovered during the execution of the plan, 5) report progress of the processes, 6) ensure products satisfy requirements, 7) assess evaluation results, 8) determine whether a task is complete, and 9) check the results for completeness.

5.1.1 Activity: Management of V&V

The Management of V&V activity is performed in all software life cycle processes and activities. This activity continuously reviews the V&V effort, revises the SVVP as necessary based upon updated project schedules and development status, and coordinates the V&V results with the developer and other supporting processes such as quality assurance, configuration management, and reviews and audits. The Management of V&V assesses each proposed change to the system and software, identifies the software requirements that are affected by the change, and plans the V&V tasks to address the change. For each proposed change, the Management of V&V assesses whether any new hazards or risks are introduced in the software, and identifies the impact of the change to the assigned software integrity levels. V&V task planning is revised by adding new V&V tasks or increasing the scope and intensity of existing V&V tasks if software integrity levels or hazards or risks are changed. The Management of V&V activity monitors and evaluates all V&V outputs. Through the use of V&V metrics and other qualitative and quantitative measures, this V&V activity develops program trend data and possible risk issues that are provided to the developer and acquirer to effect timely notification and resolution. At key program milestones (e.g., requirements review, design review, test readiness), the Management of V&V consolidates the V&V results to establish supporting evidence whether to proceed to the next set of software development activities. Whenever necessary, the Management of V&V determines whether a V&V task needs to be re-performed as a result of developer changes in the software program.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Management of V&V from the following list:

1) Task: Software Verification and Validation Plan (SVVP) Generation
2) Task: Baseline Change Assessment
3) Task: Management Review of V&V
4) Task: Management and Technical Review Support
5) Task: Interface With Organizational and Supporting Processes

5.2 Process: Acquisition

The acquisition process begins with the definition of the need (e.g., statement of need) to acquire a system, software product, or software service. The process continues with the preparation and issuance of a request for proposal (e.g., bid request, tender), selection of a supplier, and management of the acquisition process through to the acceptance of the system, software product, or software service. The V&V effort uses the acquisition process to scope the V&V effort, plan interfaces with the supplier and acquirer, and review the draft systems requirements contained in the request for proposal.

5.2.1 Activity: Acquisition Support V&V

The Acquisition Support V&V activity addresses project initiation, request for proposal, contract preparation, supplier monitoring, and acceptance and completion.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Acquisition Support V&V from the following list:

1) Task: Scoping the V&V Effort
2) Task: Planning the Interface Between the V&V Effort and Supplier
3) Task: System Requirements Review

5.3 Process: Supply

The supply process is initiated by either a decision to prepare a proposal to answer an acquirer's request for proposal or by signing and entering into a contract with the acquirer to provide the system, software product, or software service. The process continues with the determination of procedures and resources needed to manage the project, including development of project plans and execution of the plans through delivery of the system, software product, or software service to the acquirer. The V&V effort uses the supply process products to verify that the request for proposal requirements and contract requirements are consistent and satisfy user needs. The V&V planning activity uses the contract requirements including program schedules to revise and update the interface planning between the supplier and acquirer.

5.3.1 Activity: Planning V&V

The Planning V&V activity addresses the initiation, preparation of response, contract, planning, execution and control, review and evaluation, and delivery and completion activities.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Planning V&V from the following list:

1) Task: Planning the Interface Between the V&V Effort and Supplier
2) Task: Contract Verification

5.4 Process: Development

The development process contains the activities and tasks of the developer. The process contains the activities for requirements analysis, design, coding, integration, testing, and installation and acceptance related to software products. The V&V activities verify and validate these software products. The V&V activities are organized into Concept V&V, Requirements V&V, Design V&V, Implementation V&V, Test V&V, and Installation and Checkout V&V.

5.4.1 Activity: Concept V&V

The Concept V&V activity represents the delineation of a specific implementation solution to solve the user's problem. During the Concept V&V activity, the system architecture is selected, and system requirements are allocated to hardware, software, and user interface components. The Concept V&V activity addresses system architectural design and system requirements analysis. The objectives of V&V are to verify the allocation of system requirements, validate the selected solution, and ensure that no false assumptions have been incorporated in the solution.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Concept V&V from the following list:

1) Task: Concept Documentation Evaluation
2) Task: Criticality Analysis
3) Task: Hardware/Software/User Requirements Allocation Analysis
4) Task: Traceability Analysis

5) Task: Hazard Analysis
6) Task: Risk Analysis

5.4.2 Activity: Requirements V&V

The Requirements V&V activity defines the functional and performance requirements, interfaces external to the software, qualification requirements, safety and security requirements, human factors engineering, data definitions, user documentation for the software, installation and acceptance requirements, user operation and execution requirements, and user maintenance requirements. The Requirements V&V activity addresses software requirements analysis. The objectives of V&V are to ensure the correctness, completeness, accuracy, testability, and consistency of the requirements.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Requirements V&V from the following list:

1) Task: Traceability Analysis
2) Task: Software Requirements Evaluation
3) Task: Interface Analysis
4) Task: Criticality Analysis
5) Task: System V&V Test Plan Generation and Verification
6) Task: Acceptance V&V Test Plan Generation and Verification
7) Task: Configuration Management Assessment
8) Task: Hazard Analysis
9) Task: Risk Analysis

5.4.3 Activity: Design V&V

In the Design V&V activity, software requirements are transformed into an architecture and detailed design for each software component. The design includes databases and interfaces (external to the software, between the software components, and between software units). The Design V&V activity addresses software architectural design and software detailed design. The objectives of V&V are to demonstrate that the design is a correct, accurate, and complete transformation of the software requirements and that no unintended features are introduced.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Design V&V from the following list:

1) Task: Traceability Analysis
2) Task: Software Design Evaluation
3) Task: Interface Analysis
4) Task: Criticality Analysis
5) Task: Component V&V Test Plan Generation and Verification
6) Task: Integration V&V Test Plan Generation and Verification
7) Task: V&V Test Design Generation and Verification
8) Task: Hazard Analysis
9) Task: Risk Analysis

5.4.4 Activity: Implementation V&V

The Implementation V&V activity transforms the design into code, database structures, and related machine executable representations. The Implementation V&V activity addresses software coding and testing. The objectives of V&V are to verify and validate that these transformations are correct, accurate, and complete.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Implementation V&V from the following list:

1) Task: Traceability Analysis
2) Task: Source Code and Source Code Documentation Evaluation
3) Task: Interface Analysis
4) Task: Criticality Analysis
5) Task: V&V Test Case Generation and Verification
6) Task: V&V Test Procedure Generation and Verification
7) Task: Component V&V Test Execution and Verification
8) Task: Hazard Analysis
9) Task: Risk Analysis

5.4.5 Activity: Test V&V

The Test V&V activity covers software testing, software integration, software qualification testing, system integration, and system qualification testing. The Test V&V activity and its relationship to the software life cycle is shown in Figure 2. The objectives of V&V are to ensure that the software requirements and system requirements allocated to software are satisfied by execution of integration, system, and acceptance tests.

For software integrity levels 3 and 4, the V&V effort shall generate its own V&V software and system test products (e.g., plans, designs, cases, procedures), execute and record its own tests, and verify those plans, designs, cases, procedures, and test results against software requirements. For software integrity levels 1 and 2, the V&V effort shall verify the development process test activities and products (e.g., test plans, designs, cases, procedures, and test execution results).

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Test V&V from the following list:

1) Task: Traceability Analysis
2) Task: Acceptance V&V Test Procedure Generation and Verification
3) Task: Integration V&V Test Execution and Verification
4) Task: System V&V Test Execution and Verification
5) Task: Acceptance V&V Test Execution and Verification
6) Task: Hazard Analysis
7) Task: Risk Analysis

5.4.6 Activity: Installation and Checkout V&V

The Installation and Checkout V&V activity is the installation of the software product in the target environment and the acquirer's acceptance review and testing of the software product. The Installation and Checkout V&V activity addresses software installation and software acceptance support. The objectives of V&V are to verify and validate the correctness of the software installation in the target environment.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Installation and Checkout V&V from the following list:

1) Task: Installation Configuration Audit
2) Task: Installation Checkout
3) Task: Hazard Analysis
4) Task: Risk Analysis
5) Task: V&V Final Report Generation

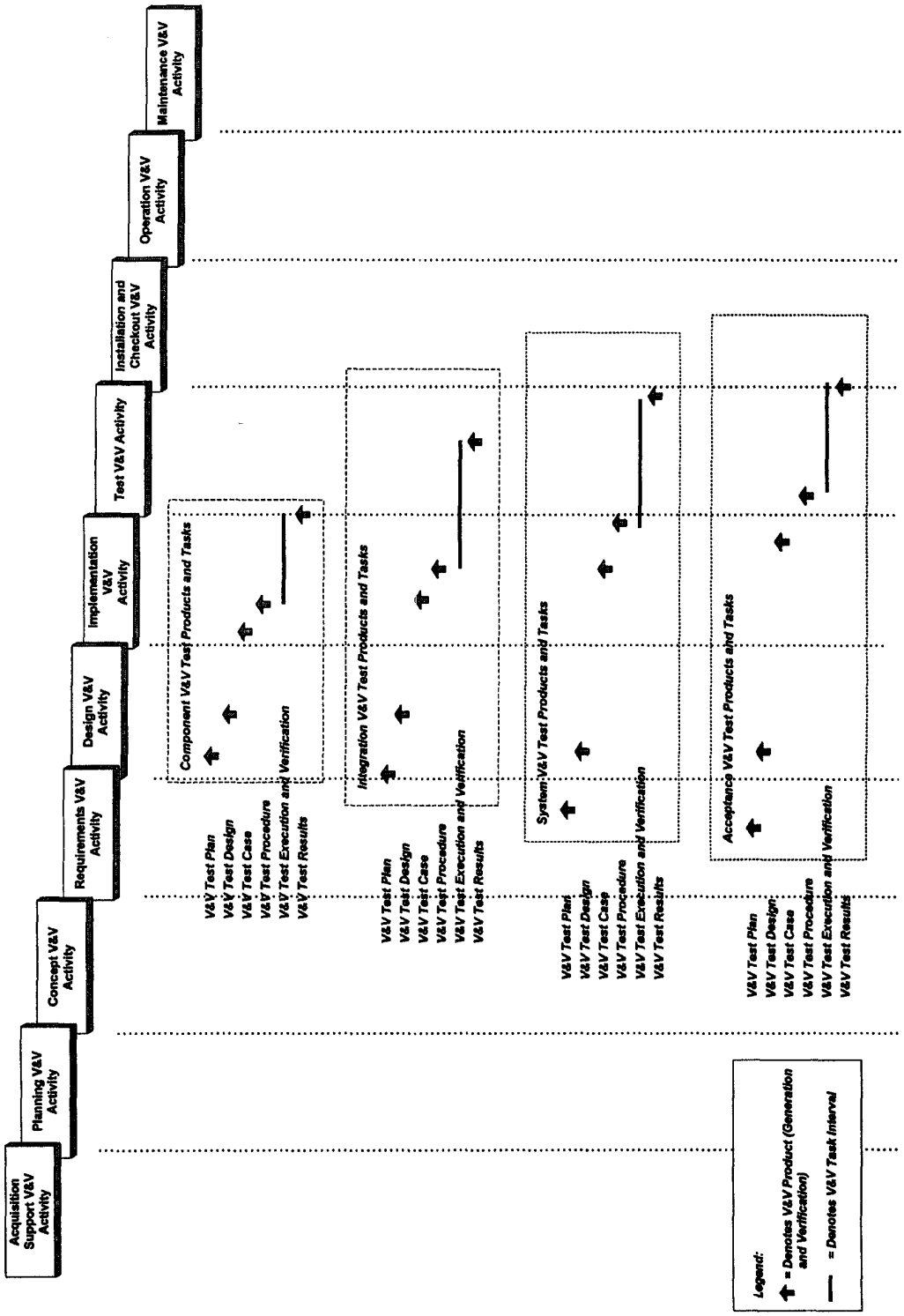


Figure 2—An example of timephasing of V&V test products and test execution tasks

5.5 Process: Operation

The operation process covers the operation of the software product and operational support to users. The Operation V&V activity evaluates the impact of any changes in the intended operating environment, assesses the effect on the system of any proposed changes, evaluates operating procedures for compliance with the intended use, and analyzes risks affecting the user and the system.

5.5.1 Activity: Operation V&V

The Operation V&V activity is the use of the software by the end user in an operational environment. The Operation V&V activity addresses operational testing, system operation, and user support. The objectives of V&V are to evaluate new constraints in the system, assess proposed changes and their impact on the software, and evaluate operating procedures for correctness and usability.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Operation V&V from the following list:

1) Task: Evaluation of New Constraints
2) Task: Proposed Change Assessment
3) Task: Operating Procedures Evaluation
4) Task: Hazard Analysis
5) Task: Risk Analysis

5.6 Process: Maintenance

The maintenance process is activated when the software product undergoes modifications to code and associated documentation caused by a problem or a need for improvement or adaptation. The Maintenance V&V activity addresses modifications (e.g., enhancements, additions, deletions), migration, or retirement of the software during the operation process.

Modifications of the software shall be treated as development processes and shall be verified and validated as described in 5.1 (management process), and 5.4 (development process) of this standard. Software integrity level assignments shall be assessed during the maintenance process. The software integrity level assignments shall be revised as appropriate to reflect the requirements of the maintenance process. These modifications may be derived from requirements specified to correct software errors (e.g., corrective), to adapt to a changed operating environment (e.g., adaptive), or to respond to additional user requests or enhancements (e.g., perfective).

5.6.1 Activity: Maintenance V&V

The Maintenance V&V activity covers modifications (e.g., corrective, adaptive, and perfective), migration, and retirement of software. Migration of software is the movement of software to a new operational environment. For migrating software, the V&V effort shall verify that the migrated software meets the requirements of 5.4 through 5.5. The retirement of software is the withdrawal of active support by the operation and maintenance organization, partial or total replacement by a new system, or installation of an upgraded system.

If the software was verified under this standard, the standard shall continue to be followed in the maintenance process. If the software was not verified under this standard and appropriate documentation is not available or adequate, the V&V effort shall determine whether the missing or incomplete documentation should be generated. In making this determination of whether to generate missing documentation, the minimum V&V requirements of the assigned software integrity level should be taken into consideration.

The Maintenance V&V activity addresses problem and modification analysis, modification implementation, maintenance review/acceptance, migration, and software retirement. The objectives of V&V are to assess proposed changes and their impact on the software, evaluate anomalies that are discovered during operation, assess migration requirements, assess retirement requirements, and re-perform V&V tasks.

The V&V effort shall perform, as appropriate for the selected software integrity level, the minimum V&V tasks for Maintenance V&V from the following list:

1) Task: SVVP Revision
2) Task: Proposed Change Assessment
3) Task: Anomaly Evaluation
4) Task: Criticality Analysis
5) Task: Migration Assessment
6) Task: Retirement Assessment
7) Task: Hazard Analysis
8) Task: Risk Analysis
9) Task: Task Iteration

6. Software V&V reporting, administrative, and documentation requirements

6.1 V&V reporting requirements

V&V reporting occurs throughout the software life cycle. The SVVP shall specify the content, format, and timing of all V&V reports. The V&V reports shall constitute the Software Verification and Validation Report (SVVR). The V&V reports shall consist of required V&V reports (i.e., V&V Task Reports, V&V Activity Summary Reports, V&V Anomaly Reports, and V&V Final Report). The V&V reports may also include optional reports. Reporting requirements are described in 7.6 of this standard.

6.2 V&V administrative requirements

The SVVP describes the V&V administrative requirements that support the V&V effort. These V&V administrative requirements shall consist of the following:

1) Anomaly Resolution and Reporting
2) Task Iteration Policy
3) Deviation Policy
4) Control Procedures
5) Standards, Practices, and Conventions

V&V administrative requirements are described in 7.7 of this standard.

6.3 V&V documentation requirements

6.3.1 V&V Test documentation

V&V Test documentation requirements shall include the test plans, designs, cases, procedures, and results for component, integration, system, and acceptance testing. The V&V test documentation shall comply with project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). The V&V task descriptions for component, integration, system, and acceptance testing are described in Table 1.

6.3.2 SVVP documentation

The V&V effort shall generate an SVVP that addresses the topics described in Clause 7 of this standard. If there is no information pertinent to a topic, the SVVP shall contain the phrase, "This topic is not applicable to this plan.", with an appropriate reason for the exclusion. Additional topics may be added to the plan. If some SVVP material appears in other documents, the SVVP may repeat the material or make reference to the material. The SVVP shall be maintained throughout the life of the software.

The SVVP shall include the V&V documentation requirements defined in 6.1, 6.2, and 6.3.1.

7. SVVP outline

The SVVP shall contain the content as described in 7.1 through 7.8 of this standard. The user of this standard may adopt any format and section numbering system for the SVVP. The SVVP section numbers listed in this standard are provided to assist the readability of this standard and are not mandatory to be in compliance with this standard.

An example SVVP outline is shown in the boxed text.

Software V&V plan outline (example)

1. Purpose
2. Referenced Documents
3. Definitions
4. V&V Overview
   - 4.1 Organization
   - 4.2 Master Schedule
   - 4.3 Software Integrity Level Scheme
   - 4.4 Resources Summary
   - 4.5 Responsibilities
   - 4.6 Tools, Techniques, and Methods
5. V&V Processes
   - 5.1 Process: Management
     - 5.1.1 Activity: Management of V&V
   - 5.2 Process: Acquisition
     - 5.2.1 Activity: Acquisition Support V&V
   - 5.3 Process: Supply
     - 5.3.1 Activity: Planning V&V
   - 5.4 Process: Development
     - 5.4.1 Activity: Concept V&V
     - 5.4.2 Activity: Requirements V&V
     - 5.4.3 Activity: Design V&V
     - 5.4.4 Activity: Implementation V&V
     - 5.4.5 Activity: Test V&V
     - 5.4.6 Activity: Installation and Checkout V&V
   - 5.5 Process: Operation
     - 5.5.1 Activity: Operation V&V
   - 5.6 Process: Maintenance
     - 5.6.1 Activity: Maintenance V&V
6. V&V Reporting Requirements
7. V&V Administrative Requirements
   - 7.1 Anomaly Resolution and Reporting
   - 7.2 Task Iteration Policy
   - 7.3 Deviation Policy
   - 7.4 Control Procedures
   - 7.5 Standards, Practices, and Conventions
8. V&V Documentation Requirements

7.1 (SVVP Section 1) Purpose

The SVVP shall describe the purpose, goals, and scope of the software V&V effort, including waivers from this standard. The software project for which the Plan is being written and the specific software processes and products covered by the software V&V effort shall be identified.

7.2 (SVVP Section 2) Referenced documents

The SVVP shall identify the compliance documents, documents referenced by the SVVP, and any supporting documents supplementing or implementing the SVVP.

7.3 (SVVP Section 3) Definitions

The SVVP shall define or reference all terms used in the SVVP, including the criteria for classifying an anomaly as a critical anomaly. All abbreviations and notations used in the SVVP shall be described.

7.4 (SVVP Section 4) V&V overview

The SVVP shall describe the organization, schedule, software integrity level scheme, resources, responsibilities, tools, techniques, and methods necessary to perform the software V&V.

7.4.1 (SVVP Section 4.1) Organization

The SVVP shall describe the organization of the V&V effort, including the degree of independence required (See Annex C of this standard). The SVVP shall describe the relationship of the V&V processes to other processes such as development, project management, quality assurance, and configuration management. The SVVP shall describe the lines of communication within the V&V effort, the authority for resolving issues raised by V&V tasks, and the authority for approving V&V products. Annex F provides an example organizational relationship chart.

7.4.2 (SVVP Section 4.2) Master Schedule

The SVVP shall describe the project life cycle and milestones. It shall summarize the schedule of V&V tasks and task results as feedback to the development, organizational, and supporting processes (e.g., quality assurance and configuration management). V&V tasks shall be scheduled to be re-performed according to the task iteration policy.

If the life cycle used in the SVVP differs from the life cycle model in this standard, this section shall describe how all requirements of the standard are satisfied (e.g., by cross-referencing to this standard).

7.4.3 (SVVP Section 4.3) Software integrity level scheme

The SVVP shall describe the agreed upon software integrity level scheme established for the system and the mapping of the selected scheme to the model used in this standard. The SVVP shall document the assignment of software integrity levels to individual components (e.g., requirements, detailed functions, software modules, subsystems, or other software partitions), where there are differing software integrity levels assigned within the program. For each SVVP update, the assignment of software integrity levels shall be reassessed to reflect changes that may occur in the integrity levels as a result of architecture selection, detailed design choices, code construction usage, or other development activities.

7.4.4 (SVVP Section 4.4) Resources summary

The SVVP shall summarize the V&V resources, including staffing, facilities, tools, finances, and special procedural requirements (e.g., security, access rights, and documentation control).

7.4.5 (SVVP Section 4.5) Responsibilities

The SVVP shall identify an overview of the organizational element(s) and responsibilities for V&V tasks.

7.4.6 (SVVP Section 4.6) Tools, techniques, and methods

The SVVP shall describe documents, hardware and software V&V tools, techniques, methods, and operating and test environment to be used in the V&V process. Acquisition, training, support, and qualification information for each tool, technology, and method shall be included.

Tools that insert code into the software shall be verified and validated to the same rigor as the highest software integrity level of the software. Tools that do not insert code shall be verified and validated to assure that they meet their operational requirements. If partitioning of tool functions can be demonstrated, only those functions that are used in the V&V processes shall be verified to demonstrate that they perform correctly for their intended use.

The SVVP shall document the metrics to be used by V&V (see Annex E), and shall describe how these metrics support the V&V objectives.

7.5 (SVVP Section 5) V&V processes

The SVVP shall identify V&V activities and tasks to be performed for each of the V&V processes described in Clause 5 of this standard, and shall document those V&V activities and tasks. The SVVP shall contain an overview of the V&V activities and tasks for all software life cycle processes.

7.5.1 (SVVP Sections 5.1 through 5.6) "Software life cycle" [2]

The SVVP shall include sections 5.1 through 5.6 for V&V activities and tasks as shown in SVVP Outline (boxed text).

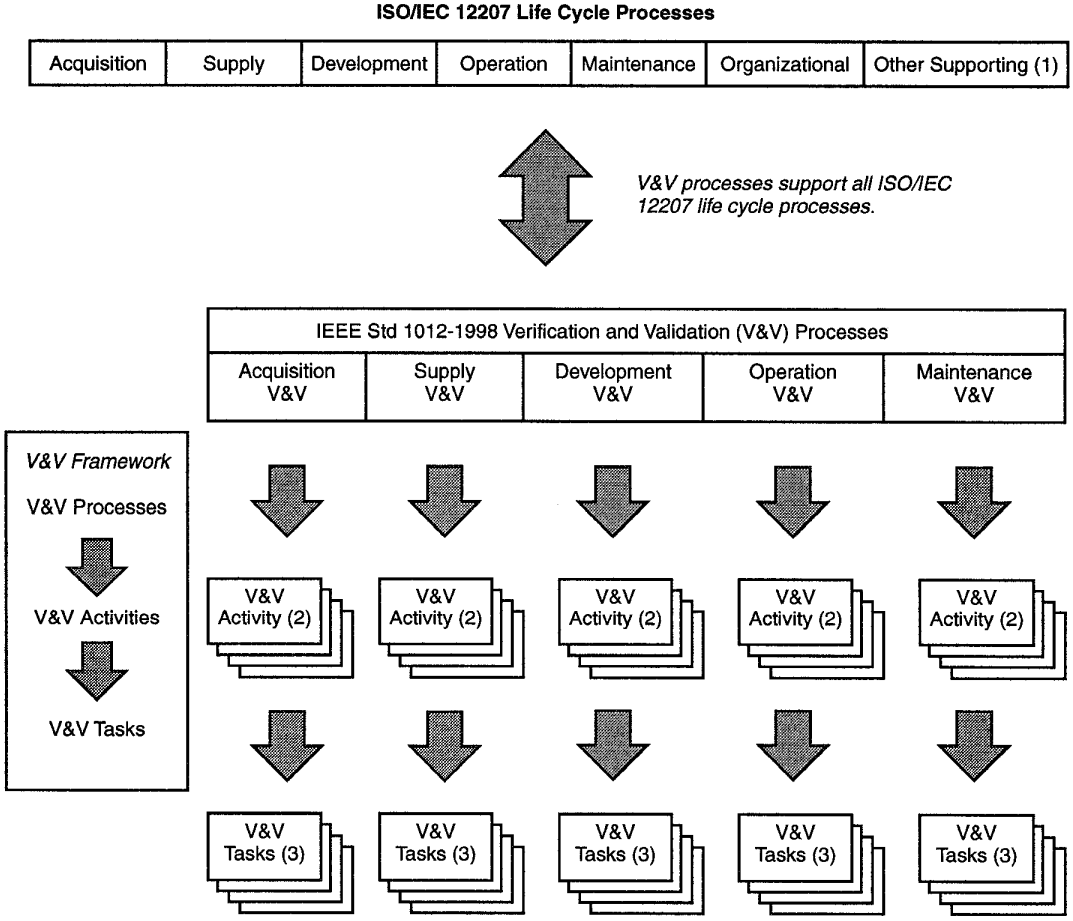
The SVVP shall address the following eight topics for each V&V activity:

1) *V&V Tasks.* The SVVP shall identify the V&V tasks to be performed. Table 1 describes the minimum V&V tasks, task criteria, and required inputs and outputs. Table 2 specifies the minimum V&V tasks that shall be performed for each software integrity level.

   The minimum tasks for software integrity level 4 are consolidated in graphic form in Figure 1.

   Optional V&V tasks may also be performed to augment the V&V effort to satisfy project needs. Optional V&V tasks are listed in Table 3 and described in Annex G. The list in Table 3 is illustrative and not exhaustive. The standard allows for optional V&V tasks to be used as appropriate.

[2]"Software Life Cycle" V&V sections are 5.1 Process: Management; 5.2 Process: Acquisition; 5.3 Process: Supply; 5.4 Process: Development; 5.5 Process: Operation; and 5.6 Process: Maintenance.



NOTES

1—"Other Supporting Processes" consist of "Documentation," "Configuration Management," "Quality Assurance," "Joint Review," "Audit," and "Problem Resolution."

2—Management of V&V activity is concurrent with all V&V activities.

3—The task description, inputs, and outputs of all V&V tasks are included in Table 1.

Figure 3—Framework of V&V processes, activities, and tasks hierarchy

Some V&V tasks are applicable to more than one software integrity level. The degree of rigor and intensity in performing and documenting the task should be commensurate with the software integrity level. As the software integrity level decreases, so does the required scope, intensity, and degree of rigor associated with the V&V task. For example, a hazard analysis performed for software integrity level 4 software might be formally documented and consider failures at the module level; a hazard analysis for software integrity level 3 software may consider only significant software failures and be documented informally as part of the design review process.

Testing requires advance planning that spans several development activities. Test documentation and its occurrence at specific processes in the life cycle are shown in Figures 1 and 2.

2) *Methods and Procedures.* The SVVP shall describe the methods and procedures for each task, including on-line access, and conditions for observation/evaluation of development processes. The SVVP shall define the criteria for evaluating the task results.

3) *Inputs.* The SVVP shall identify the required inputs for each V&V task. The SVVP shall specify the source and format of each input. The inputs required for the minimum V&V tasks are identified in Table 1. Other inputs may be used. For any V&V activity and task, all of the required inputs from preceding activities and tasks may be used but for conciseness, only the primary inputs are listed in Table 1.

4) *Outputs.* The SVVP shall identify the required outputs from each V&V task. The SVVP shall specify the purpose, format, and recipients of each output. The required outputs from each of the V&V tasks are identified in Table 1. Other outputs may be produced.

   The outputs of the Management of V&V and of the V&V tasks shall become inputs to subsequent processes and activities, as appropriate.

5) *Schedule.* The SVVP shall describe the schedule for the V&V tasks. The SVVP shall establish specific milestones for initiating and completing each task, for the receipt and criteria of each input, and for the delivery of each output.

6) *Resources.* The SVVP shall identify the resources for the performance of the V&V tasks. The SVVP shall specify resources by category (e.g., staffing, equipment, facilities, travel, and training.)

7) *Risks and Assumptions.* The SVVP shall identify the risks (e.g., schedule, resources, or technical approach) and assumptions associated with the V&V tasks. The SVVP shall provide recommendations to eliminate, reduce, or mitigate risks.

8) *Roles and Responsibilities.* The SVVP shall identify the organizational elements or individuals responsible for performing the V&V tasks.

7.6 (SVVP Section 6) V&V reporting requirements

V&V reporting shall consist of Task Reports, V&V Activity Summary Reports, Anomaly Reports, and the V&V Final Report. Task report(s), V&V activity summary report(s), and anomaly report(s) are provided as feedback to the software development process regarding the technical quality of each software product and process.

V&V reporting may also include optional reports such as special study reports. The format and grouping of the V&V reports are user defined. The required V&V reports shall consist of the following:

1) *Task Reports.* V&V tasks shall document V&V task results and status, and shall be in a format appropriate for technical disclosure. Examples of Task Reports include the following:

   a) Anomaly Evaluation
   b) Baseline Change Assessment
   c) Concept Documentation Evaluation
   d) Configuration Management Assessment
   e) Contract Verification
   f) Criticality Analysis
   g) Evaluation of New Constraints
   h) Hardware/Software/User Requirements Allocation Analysis
   i) Hazard Analysis

j) Installation Checkout
k) Installation Configuration Audit
l) Interface Analysis
m) Migration Assessment
n) Operating Procedures Evaluation
o) Proposed Change Assessment
p) Recommendations
q) Review Results
r) Risk Analysis
s) Software Design Evaluation
t) Software Integrity Levels
u) Software Requirements Evaluation
v) Source Code and Source Code Documentation Evaluation
w) System Requirements Review
x) Test Results
y) Traceability Analysis

2) *V&V Activity Summary Reports.* An Activity Summary Report shall summarize the results of V&V tasks performed for each of the following V&V activities: Acquisition Support, Planning, Concept, Requirements, Design, Implementation, Test, and Installation and Checkout. For the Operation activity and Maintenance activity, V&V Activity Summary reports may be either updates to previous V&V activity summary reports or separate documents. Each V&V Activity Summary Report shall contain the following:

a) Description of V&V tasks performed
b) Summary of task results
c) Summary of anomalies and resolution
d) Assessment of software quality
e) Identification and assessment of technical and management risks
f) Recommendations

3) *Anomaly Report.* An Anomaly Report shall document each anomaly detected by the V&V effort. Each anomaly shall be evaluated for its impact on the software system and assessed as to whether it is a critical anomaly (e.g., IEEE Std 1044-1993 [B9]). The scope and application of V&V activities and tasks shall be revised to address the causes of these anomalies and risks. Each Anomaly Report shall contain the following:

a) Description and location in document or code
b) Impact
c) Cause of the anomaly and description of the error scenario
d) Anomaly criticality level
e) Recommendations

4) *V&V Final Report.* The V&V Final Report shall be issued at the end of the Installation and Checkout activity or at the conclusion of the V&V effort. The V&V Final Report shall include the following:

a) Summary of all life cycle V&V activities
b) Summary of task results
c) Summary of anomalies and resolutions
d) Assessment of overall software quality
e) Lessons learned/best practices
f) Recommendations

Optional reports may include the following:

1) *Special Studies Reports.* These reports shall describe any special V&V studies conducted during the software life cycle. The title of the report may vary according to the subject matter. The reports shall document the results of technical and management tasks and shall include the following:

   a) Purpose and objectives
   b) Approach
   c) Summary of results

2) *Other Reports.* These reports shall describe the results of tasks not defined in the SVVP. The title of the report may vary according to the subject matter. These other task reports may include, for example, quality assurance results, end user testing results, safety assessment report, or configuration and data management status results.

7.7 (SVVP Section 7) V&V administrative requirements

Administrative V&V requirements shall describe anomaly resolution and reporting, task iteration policy, deviation policy, control procedures, and standards, practices, and conventions.

7.7.1 (SVVP Section 7.1) Anomaly resolution and reporting

The SVVP shall describe the method of reporting and resolving anomalies, including the criteria for reporting an anomaly, the anomaly report distribution list, and the authority and time lines for resolving anomalies. The section shall define the anomaly criticality levels. Classification for software anomalies may be found in IEEE Std 1044-1993 [B9].

7.7.2 (SVVP Section 7.2) Task iteration policy

The SVVP shall describe the criteria used to determine the extent to which a V&V task shall be repeated when its input is changed or task procedure is changed. These criteria may include assessments of change, software integrity level, and effects on budget, schedule, and quality.

7.7.3 (SVVP Section 7.3) Deviation policy

The SVVP shall describe the procedures and criteria used to deviate from the Plan. The information required for deviations shall include task identification, rationale, and effect on software quality. The SVVP shall identify the authorities responsible for approving deviations.

7.7.4 (SVVP Section 7.4) Control procedures

The SVVP shall identify control procedures applied to the V&V effort. These procedures shall describe how software products and V&V results shall be configured, protected, and stored.

These procedures may describe quality assurance, configuration management, data management, or other activities if they are not addressed by other efforts. The SVVP shall describe how the V&V effort shall comply with existing security provisions and how the validity of V&V results shall be protected from unauthorized alterations.

7.7.5 (SVVP Section 7.5) Standards, practices, and conventions

The SVVP shall identify the standards, practices, and conventions that govern the performance of V&V tasks including internal organizational standards, practices, and policies.

7.8 (SVVP Section 8) V&V documentation requirements

The SVVP shall define the purpose, format, and content of the test documents. A description of the format for these test documents may be found in IEEE Std 829-1983 [B5]. If the V&V effort uses test documentation or test types (e.g., component, integration, system, acceptance) different from those in this standard, the software V&V effort shall show a mapping of the proposed test documentation and execution to the test items defined in this standard. Test planning tasks defined in Table 1 shall be implemented in the test plan, test design(s), test case(s), and test procedure(s) documentation.

The SVVP shall describe the purpose, format, and content for the following V&V test documents:

1) Test Plan
2) Test Design
3) Test Cases
4) Test Procedures
5) Test Results

All V&V results and findings shall be documented in the V&V Final Report.

Table 1—V&V tasks, inputs, and outputs

V&V tasks	Required inputs	Required outputs
5.1.1 Management of V&V Activity (in parallel with all processes)		
(1) Software Verification and Validation Plan (SVVP) Generation. Generate an SVVP for all life cycle processes. The SVVP may require updating throughout the life cycle. Outputs of other activities are inputs to the SVVP. Establish a baseline SVVP prior to the Requirements V&V activities. Identify project milestones in the SVVP. Schedule V&V tasks to support project management reviews and technical reviews. See Clause 7 for an example SVVP outline and content of the SVVP.	SVVP (previous update) Contract Concept Documentation (e.g., Statement of Need, Advance Planning Report, Project Initiation Memo, Feasibility Studies, System Requirements, Governing Regulations, Procedures, Policies, customer acceptance criteria and requirements, Acquisition Documentation, Business Rules, draft system architecture) Supplier Development Plans and Schedules	SVVP and Updates
(2) Baseline Change Assessment. Evaluate proposed software changes (e.g., anomaly corrections and requirement changes) for effects on previously completed V&V tasks. Plan iteration of affected tasks or initiate new tasks to address software baseline changes or iterative development processes. Verify and validate that the change is consistent with system requirements and does not adversely affect requirements directly or indirectly. An adverse effect is a change that could create new system hazards and risks or impact previously resolved hazards and risks.	SVVP Proposed Changes Hazard Analysis Report Risks identified by V&V Tasks	Updated SVVP Task Report(s) — Baseline Change Assessment Anomaly Report(s)
(3) Management Review of V&V. Review and summarize the V&V effort to define changes to V&V tasks or to redirect the V&V effort. Recommend whether to proceed to the next set of V&V and development life cycle activities, and provide task reports, anomaly reports, and V&V Activity Summary Reports to the organizations identified in the SVVP. Verify that all V&V tasks comply with task requirements defined in the SVVP. Verify that V&V task results have a basis of evidence supporting the results. Assess all V&V results and provide recommendations for program acceptance and certification as input to the V&V Final Report. The management review of V&V may use any review methodology such as provided in IEEE Std 1028-1988 [B8].	SVVP and Updates Supplier Development Plans and Schedules V&V task results [e.g., technical accomplishments, V&V reports, resource utilization, V&V metrics (see Annex E), plans, and identified risks]	Updated SVVP Task Report(s)— Recommendations V&V Activity Summary Reports Recommendations to the V&V Final Report
(4) Management and Technical Review Support. Support project management reviews and technical reviews (e.g., Preliminary Design Review, and Critical Design Review) by assessing the review materials, attending the reviews, and providing task reports and anomaly reports. Verify the timely delivery according to the approved schedule of all software products and documents. The management and technical review support may use any review methodology such as provided in IEEE Std 1028-1988 [B8].	V&V task results Materials for review (e.g., SRS, IRS, SDD, IDD, test documents)	Task Report(s)— Review Results Anomaly Report(s)
(5) Interface With Organizational and Supporting Processes. Coordinate the V&V effort with organizational (e.g., management, improvement) and supporting processes (e.g., quality assurance, joint review, and problem resolution). Identify the V&V data to be exchanged with these processes. Document the data exchange requirements in the SVVP.	SVVP Data identified in the SVVP from organizational and supporting processes	Updated SVVP

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.2.1 Acquisition Support V&V Activity (acquisition process)		
(1) Scoping the V&V Effort. Define the project V&V software criticality (e.g., safety, security, mission critical, technical complexity). Assign a software integrity level to the system and the software. Establish the degree of independence (see Annex C), if any, required for the V&V. Provide an estimate of the V&V budget, including test facilities and tools as required. To scope the V&V effort, the following steps shall be performed: (a) Adopt the system integrity scheme assigned to the project. If no system integrity level scheme exists, then one is selected. (b) Determine the minimum V&V tasks for the software integrity level using Table 2 and the selected software integrity level scheme. (c) Augment the minimum V&V tasks with optional V&V tasks, as necessary. (d) Establish the scope of the V&V from the description of V&V tasks, inputs, and outputs defined in Table 1.	Preliminary System Description Statement of Need Request for Proposal (RFP) or tender System Integrity Level Scheme	Updated SVVP
(2) Planning the Interface Between the V&V Effort and Supplier. Plan the V&V schedule for each V&V task. Identify the preliminary list of development processes and products to be evaluated by the V&V processes. Describe V&V access rights to proprietary and classified information. It is recommended that the plan be coordinated with the acquirer. Incorporate the project software integrity level scheme into the planning process.	SVVP RFP or tender Contract Supplier Development Plans and Schedules	Updated SVVP
(3) System Requirements Review. Review the system requirements (e.g., system requirements specification, feasibility study report, business rules description) in the RFP or tender to 1) verify the consistency of requirements to user needs, 2) validate whether the requirements can be satisfied by the defined technologies, methods, and algorithms defined for the project (feasibility), and 3) verify whether objective information that can be demonstrated by testing is provided in the requirements (testability). Review other requirements such as deliverable definitions, listing of appropriate compliance standards and regulations, user needs, etc., for completeness, correctness, and accuracy.	Preliminary System Description Statement of Need User Needs RFP or tender	Task Report(s)— System Requirements Review Anomaly Report(s)
5.3.1 Planning V&V Activity (supply process)		
(1) Planning the Interface Between the V&V Effort and Supplier. Review the supplier development plans and schedules to coordinate the V&V effort with development activities. Establish procedures to exchange V&V data and results with the development effort. It is recommended that the plan be coordinated with the acquirer. Incorporate the project software integrity level scheme into the planning process.	SVVP Contract Supplier Development Plans and Schedules	Updated SVVP
(2) Contract Verification. Verify that 1) system requirements (from RFP or tender, and contract) satisfy and are consistent with user needs; 2) procedures are documented for managing requirement changes and for identifying the management hierarchy to address problems; 3) procedures for interface and cooperation among the parties are documented, including ownership, warranty, copyright, and confidentiality; and 4) acceptance criteria and procedures are documented in accordance with requirements.	SVVP RFP or tender Contract User Needs Supplier Development Plans and Schedules	Updated SVVP Task Report(s)— Contract Verification Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.1 Concept V&V Activity (development process)		
(1) Concept Documentation Evaluation. Verify that the concept documentation satisfies user needs and is consistent with acquisition needs. Validate constraints of interfacing systems and constraints or limitations of proposed approach. Analyze system requirements and validate that the following satisfy user needs: 1) system functions; 2) end-to-end system performance; 3) feasibility and testability of the functional requirements; 4) system architecture design; 5) operation and maintenance requirements; and 6) migration requirements from an existing system where applicable.	Concept Documentation Supplier Development Plans and Schedules User Needs Acquisition Needs	Task Report(s)— Concept Documentation Evaluation Anomaly Report(s)
(2) Criticality Analysis. Determine whether software integrity levels are established for requirements, detailed functions, software modules, subsystem, or other software partitions. Verify that the assigned software integrity levels are correct. If software integrity levels are not assigned, then assign software integrity levels to the system requirements. Document the software integrity level assigned to individual software components (e.g., requirements, detailed functions, software modules, subsystems, or other software partitions). For V&V planning purposes, the most critical software integrity level assigned to individual elements shall be the integrity level assigned to the entire software. Verify whether any software component can influence individual software components assigned a higher software integrity level, and if such conditions exist, then assign that software component the same higher software integrity level.	Concept Documentation (system requirements) Developer integrity level assignments	Task Report(s)— Software Integrity Levels Task Report(s)— Criticality Analysis Anomaly Report(s)
(3) Hardware/Software/User Requirements Allocation Analysis. Verify the correctness, accuracy, and completeness of the concept requirement allocation to hardware, software, and user interfaces against user needs. (3.1) Correctness a. Verify that performance requirements (e.g., timing, response time, and throughput) allocated to hardware, software, and user interfaces satisfy user needs. (3.2) Accuracy a. Verify that the internal and external interfaces specify the data formats, interface protocols, frequency of data exchange at each interface, and other key performance requirements to demonstrate compliance with user requirements. (3.3) Completeness a. Verify that application specific requirements such as functional diversity, fault detection, fault isolation, and diagnostic and error recovery satisfy user needs, b. Verify that the user's maintenance requirements for the system are completely specified. c. Verify that the migration from the existing system and replacement of the system satisfy user needs.	User Needs Concept Documentation	Task Report(s)— Hardware/ Software/User Requirements Allocation Analysis Anomaly Report(s)
(4) Traceability Analysis. Identify all system requirements that will be implemented completely or partially by software. Verify that these system requirements are traceable to acquisition needs. Start the software requirements traceability analysis with system requirements.	Concept Documentation	Task Report(s)— Traceability Analysis Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.1 Concept V&V Activity (development process) *(Continued)*		
(5) Hazard Analysis. Analyze the potential hazards to and from the conceptual system. The analysis shall 1) identify the potential system hazards; 2) assess the severity of each hazard; 3) assess the probability of each hazard; and 4) identify mitigation strategies for each hazard.	Concept Documentation	Task Report(s)— Hazard Analysis Anomaly Report(s)
(6) Risk Analysis. Identify the technical and management risks. Provide recommendations to eliminate, reduce, or mitigate the risks.	Concept Documentation Supplier Development Plans and Schedules Hazard Analysis Report V&V task results	Task Report(s)— Risk Analysis Anomaly Report(s)
5.4.2 Requirements V&V Activity (development process)		
(1) Traceability Analysis. Trace the software requirements (SRS and IRS) to system requirements (Concept Documentation), and system requirements to the software requirements. Analyze identified relationships for correctness, consistency, completeness, and accuracy. The task criteria are as follows: (1.1) Correctness a. Validate that the relationships between each software requirement and its system requirement are correct. (1.2) Consistency a. Verify that the relationships between the software and system requirements are specified to a consistent level of detail. (1.3) Completeness a. Verify that every software requirement is traceable to a system requirement with sufficient detail to show compliance with the system requirement. b. Verify that all system requirements related to software are traceable to software requirements. (1.4) Accuracy a. Validate that the system performance and operating characteristics are accurately specified by the traced software requirements.	Concept Documentation (System requirements) SRS IRS	Task Report(s)— Traceability Analysis Anomaly Report(s)
(2) Software Requirements Evaluation. Evaluate the requirements (e.g., functional, capability, interface, qualification, safety, security, human factors, data definitions, user documentation, installation and acceptance, user operation, and user maintenance) of the SRS and IRS for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are as follows: (2.1) Correctness a. Verify and validate that the software requirements satisfy the system requirements allocated to software within the assumptions and constraints of the system. b. Verify that the software requirements comply with standards, references, regulations, policies, physical laws, and business rules. c. Validate the sequences of states and state changes using logic and data flows coupled with domain expertise, prototyping results, engineering principles, or other basis. d. Validate that the flow of data and control satisfy functionality and performance requirements. e. Validate data usage and format.	Concept Documentation SRS IRS	Task Report(s)—Software Requirements Evaluation Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.2 Requirements V&V Activity (development process) *(Continued)*		
(2.2) Consistency a. Verify that all terms and concepts are documented consistently. b. Verify that the function interactions and assumptions are consistent and satisfy system requirements and acquisition needs. c. Verify that there is internal consistency between the software requirements and external consistency with the system requirements. (2.3) Completeness a. Verify that the following elements are in the SRS or IRS, within the assumptions and constraints of the system: 1. Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting, and logging); 2. Process definition and scheduling; 3. Hardware, software, and user interface descriptions. 4. Performance criteria (e.g., timing sizing, speed, capacity, accuracy, precision, safety, and security); 5. Critical configuration data; and 6. System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing). b. Verify that the SRS and IRS satisfy specified configuration management procedures. (2.4) Accuracy a. Validate that the logic, computational, and interface precision (e.g., truncation and rounding) satisfy the requirements in the system environment. b. Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws. (2.5) Readability a. Verify that the documentation is legible, understandable, and unambiguous to the intended audience. b. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, and symbols. (2.6) Testability a. Verify that there are objective acceptance criteria for validating the requirements of the SRS and IRS.		
(3) Interface Analysis. Verify and validate that the requirements for software interfaces with hardware, user, operator, and other systems are correct, consistent, complete, accurate, and testable. The task criteria are as follows: (3.1) Correctness a. Validate the external and internal system and software interface requirements. (3.2) Consistency a. Verify that the interface descriptions are consistent between the SRS and IRS. (3.3) Completeness a. Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security).	Concept Documentation SRS IRS	Task Report(s)—Interface Analysis Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.2 Requirements V&V Activity (development process) *(Continued)*		
(3.4) Accuracy a. Verify that each interface provides information with the required accuracy. (3.5) Testability a. Verify that there are objective acceptance criteria for validating the interface requirements.		
(4) Criticality Analysis. Review and update the existing criticality analysis results from the prior Criticality Task Report using the SRS and IRS. Implementation methods and interfacing technologies may cause previously assigned software integrity levels to be raised or lowered for a given software element (i.e., requirement, module, function, subsystem, other software partition). Verify that no inconsistent or undesired software integrity consequences are introduced by reviewing the revised software integrity levels.	Task Report(s)—Criticality SRS IRS	Task Report(s)—Criticality Analysis Anomaly Report(s)
(5) System V&V Test Plan Generation and Verification. *(For Software Integrity Levels 3 and 4)* Plan system V&V testing to validate software requirements. Plan tracing of system requirements to test designs, cases, procedures, and results. Plan documentation of test designs, cases, procedures, and results. The System V&V Test Plan shall address the following: 1) compliance with all system requirements (e.g., functional, performance, security, operation, and maintenance) as complete software end items in the system environment, 2) adequacy of user documentation (e.g., training materials, procedural changes), and 3) performance at boundaries (e.g., data, interfaces) and under stress conditions. Verify that the System V&V Test Plan conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the System Test Plan satisfies the following criteria: 1) test coverage of system requirements; 2) appropriateness of test methods and standards used; 3) conformance to expected results; 4) feasibility of system qualification testing; and 5) feasibility and testability of operation and maintenance requirements. *(For Software Integrity Levels 1 and 2)* Verify that developer's System Test Plans conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the System Test Plan satisfies the following criteria: 1) test coverage of system requirements; 2) appropriateness of test methods and standards used; 3) conformance to expected results; 4) feasibility of system qualification testing; and 5) capability to be operated and maintained.	Concept Documentation (System requirements) SRS IRS User Documentation System Test Plan	Anomaly Report(s) System V&V Test Plan
(6) Acceptance V&V Test Plan Generation and Verification. *(For Software Integrity Levels 3 and 4)* Plan Acceptance V&V testing to validate that software correctly implements system and software requirements in an operational environment. The task criteria are 1) compliance with acceptance requirements in the operational environment, and 2) adequacy of user documentation. Plan tracing of acceptance test requirements to test design, cases, procedures, and execution results. Plan documentation of test tasks and results. Verify that the Acceptance V&V Test Plan complies with Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the Acceptance Test Plan satisfies the following criteria: 1) test coverage of system	Concept Documentation SRS IRS User Documentation Acceptance Test Plan	Acceptance V&V Test Plan Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.2 Requirements V&V Activity (development process) *(Continued)*		
requirements; 2) conformance to expected results; and 3) feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs). *(For Software Integrity Level 2)* Verify that the developer's Acceptance Test Plan conforms to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the developer's Acceptance Test Plan satisfies the following criteria: 1) test coverage of system requirements; 2) conformance to expected results; and 3) feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs). *(For Software Integrity Level 1, there are no acceptance test requirements.)*		
(7) Configuration Management Assessment. Verify that the Configuration Management process is complete and adequate. The task criteria are as follows: (7.1) Completeness a. Verify that there is a process for describing the software product functionality, tracking program versions, and managing changes. (7.2) Adequacy a. Verify that the configuration management process is adequate for the development complexity, software and system size, software integrity level, project plans, and user needs.	Software Configuration Management Process Documentation	Task Report (s)— Configuration Management Assessment Anomaly Report (s)
(8) Hazard Analysis. Determine software contributions to system hazards. The hazard analysis shall a) identify the software requirements that contribute to each system hazard; and b) validate that the software addresses, controls, or mitigates each hazard.	SRS IRS Hazard Analysis Report	Task Report(s)— Hazard Analysis Anomaly Report(s)
(9) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks.	Concept Documentation SRS IRS Supplier Development Plans and Schedules Hazard Analysis Report V&V task results	Task Report(s)— Risk Analysis Anomaly Report(s)
5.4.3 Design V&V Activity (development process)		
(1) Traceability Analysis. Trace design elements (SDD and IDD) to requirements (SRS and IRS), and requirements to design elements. Analyze relationships for correctness, consistency, and completeness. The task criteria are as follows: (1.1) Correctness a Validate the relationship between each design element and the software requirement. (1.2) Consistency a. Verify that the relationship between the design elements and the software requirements are specified to a constant level of detail.	SRS SDD IRS IDD	Task Report(s)— Traceability Analysis Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.3 Design V&V Activity (development process) *(Continued)*		
(1.3) Completeness a. Verify that all design elements are traceable from the software requirements. b. Verify that all software requirements are traceable to the design elements.		
(2) Software Design Evaluation. Evaluate the design elements (SDD and IDD) for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are as follows: (2.1) Correctness a. Verify and validate that the source code component satisfies the software design. b. Verify that the source code components comply with standards, references, regulations, policies, physical laws, and business rules. c. Validate the source code component sequences of states and state changes using logic and data flows coupled with domain expertise, prototyping results, engineering principles, or other basis. d. Validate that the flow of data and control satisfy functionality and performance requirements. e. Validate data usage and format. f. Assess the appropriateness of coding methods and standards. (2.2) Consistency a. Verify that all terms and code concepts are documented consistently. b. Verify that there is internal consistency between the source code components. (2.3) Completeness a. Verify that the following elements are in the SDD, within the assumptions and constraints of the system: 1. Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging); 2. Process definition and scheduling; 3. Hardware, software, and user interface descriptions; 4. Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety, and security); 5. Critical configuration data; 6. System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing). b. Verify that the SDD and IDD satisfy specified configuration management procedures. (2.4) Accuracy a. Validate that the logic, computational, and interface precision (e.g., truncation and rounding) satisfy the requirements in the system environment. b. Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws. (2.5) Readability a. Verify that the documentation is legible, understandable, and unambiguous to the intended audience. b. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, symbols, and design language, if any.	SRS IRS SDD IDD Design Standards (e.g., standards, practices, and conventions)	Task Report(s)— Software Design Evaluation Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.3 Design V&V Activity (development process) *(Continued)*		
(2.6) Testability a. Verify that there are objective acceptance criteria for validating each software design element and the system design. b. Verify that each software design element is testable to objective acceptance criteria.		
(3) Interface Analysis. Verify and validate that the software design interfaces with hardware, user, operator, software, and other systems for correctness, consistency, completeness, accuracy, and testability. The task criteria are as follows: (3.1) Correctness a. Validate the external and internal software interface design in the context of system requirements. (3.2) Consistency a. Verify that the interface design is consistent between the SDD and IDD. (3.3) Completeness a. Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security). (3.4) Accuracy a. Verify that each interface provides information with the required accuracy. (3.5) Testability a. Verify that there are objective acceptance criteria for validating the interface design.	Concept Documentation (System requirements) SRS IRS SDD IDD	Task Report(s)—Interface Analysis Anomaly Report(s)
(4) Criticality Analysis. Review and update the existing criticality analysis results from the prior Criticality Task Report using the SDD and IDD. Implementation methods and interfacing technologies may cause previously assigned software integrity levels to be raised or lowered for a given software element (i.e., requirement, module, function, subsystem, other software partition). Verify that no inconsistent or undesired software integrity consequences are introduced by reviewing the revised software integrity levels.	Task Report(s)—Criticality SDD IDD	Task Report(s)—Criticality Analysis Anomaly Report(s)
(5) Component V&V Test Plan Generation and Verification. *(For Software Integrity Levels 3 and 4.)* Plan component V&V testing to validate that the software components (e.g., units, source code modules) correctly implement component requirements. The task criteria are 1) compliance with design requirements; 2) assessment of timing, sizing, and accuracy; 3) performance at boundaries and interfaces and under stress and error conditions; and 4) measures of requirements test coverage and software reliability and maintainability. Plan tracing of design requirements to test design, cases, procedures, and results. Plan documentation of test tasks and results. Verify that the Component V&V Test Plan complies with Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the Component V&V Test Plan satisfies the following criteria: 1) traceable to the software requirements and design; 2) external consistency with the software requirements and design; 3) internal consistency between unit requirements; 4) test coverage of requirements in each unit; 5) feasibility of software	SRS SDD IRS IDD Component Test Plan	Component V&V Test Plan Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.3 Design V&V Activity (development process) *(Continued)*		
integration and testing; and 6) feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs). *(For Software Integrity Level 2.)* Verify that the developer's Component Test Plan conforms to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983). Validate that the developer's Component Test Plan satisfies the following criteria: 1) traceable to the software requirements and design; 2) external consistency with the software requirements and design; 3) internal consistency between unit requirements; 4) test coverage of units; 5) feasibility of software integration and testing; and 6) feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs). *(For Software Integrity Level 1, there are no component test requirements.)*		
(6) Integration V&V Test Plan Generation and Verification. *(For Software Integrity Levels 3 and 4.)* Plan integration testing to validate that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. The task criteria are 1) compliance with increasingly larger set of functional requirements at each stage of integration; 2) assessment of timing, sizing, and accuracy; 3) performance at boundaries and under stress conditions; and 4) measures of requirements test coverage and software reliability. Plan tracing of requirements to test design, cases, procedures, and results. Plan documentation of test tasks and results. Verify that the Integration V&V Test Plan complies with Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the Integration V&V Test Plan satisfies the following criteria: 1) traceable to the system requirements; 2) external consistency with the system requirements; 3) internal consistency; 4) test coverage of the software requirements; 5) appropriateness of test standards and methods used; 6) conformance to expected results; 7) feasibility of software qualification testing; and 8) feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs). *(For Software Integrity Levels 1 and 2.)* Verify that the developer's Integration Test Plan conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983). Validate that the developer's Integration Test Plan satisfies the following criteria: 1) traceable to the system requirements; 2) external consistency with the system requirements; 3) internal consistency; 4) test coverage of the software requirements; 5) appropriateness of test standards and methods; 6) conformance to expected results; 7) feasibility of software qualification testing; and 8) feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs).	SRS IRS SDD IDD Integration Test Plan	Integration V&V Test Plan Anomaly Report(s)
(7) V&V Test Design Generation and Verification. *(For Software Integrity Levels 3 and 4.)* Design tests for: 1) component testing; 2) integration testing; 3) system testing; and 4) acceptance testing. Continue tracing required by the V&V Test Plan. Verify that the V&V Test Designs comply with	SDD IDD User Documentation Test Plans Test Designs	Component V&V Test Design(s) Integration V&V Test Design(s) System V&V Test Design(s) Acceptance V&V Test Design(s) Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.3 Design V&V Activity (development process) *(Continued)*		
Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the V&V Test Designs satisfy the criteria in V&V tasks 5.4.3 Task 5; 5.4.3 Task 6; 5.4.2 Task 5; and 5.4.2 Task 6, for component, integration, system, and acceptance testing, respectively. *(For Software Integrity Levels 1 and 2.)* Verify that the developer's Test Designs conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983). Validate that the developer's Test Designs satisfy the criteria in V&V tasks 5.4.3 Task 5; 5.4.3 Task 6; 5.4.2 Task 5; and 5.4.2 Task 6 for component (level 2 only), integration (levels 1 and 2), system (levels 1 and 2), and acceptance (level 2 only) testing, respectively.		
(8) Hazard Analysis. Verify that logic design and associated data elements correctly implement the critical requirements and introduce no new hazards. Update the hazard analysis.	SDD IDD Hazard Analysis Report	Task Report(s)— Hazard Analysis Anomaly Report(s)
(9) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks.	SDD IDD Supplier Development Plans and Schedules Hazard Analysis Report V&V task results	Task Report(s)— Risk Analysis Anomaly Report(s)
5.4.4 Implementation V&V Activity (development process)		
(1) Traceability Analysis. Trace the source code components to corresponding design specification(s), and design specification(s) to source code components. Analyze identified relationships for correctness, consistency, and completeness. The task criteria are as follows: (1.1) Correctness a. Validate the relationship between the source code components and design element(s). (1.2) Consistency a. Verify that the relationships between the source code components and design elements are specified to a consistent level of detail. (1.3) Completeness a. Verify that all source code components are traceable from the design elements. b. Verify that all design elements are traceable to the source code components.	SDD IDD Source Code	Task Report(s)— Traceability Analysis Anomaly Report(s)
(2) Source Code and Source Code Documentation Evaluation. Evaluate the source code components (Source Code Documentation) for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are as follows: (2.1) Correctness a. Verify and validate that the source code component satisfies the software design. b. Verify that the source code components comply with standards, references, regulations, policies, physical laws, and business rules.	Source Code SDD IDD Coding Standards (e.g., standards, practices, project restrictions, and conventions) User Documentation	Task Report(s)— Source Code and Source Code Documentation Evaluation Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.4 Implementation V&V Activity (development process) *(Continued)*		
c. Validate the source code component sequences of states and state changes using logic and data flows coupled with domain expertise, prototyping results, engineering principles, or other basis. d. Validate that the flow of data and control satisfy functionality and performance requirements. e. Validate data usage and format. f. Assess the appropriateness of coding methods and standards. (2.2) Consistency a. Verify that all terms and code concepts are documented consistently. b. Verify that there is internal consistency between the source code components. c. Validate external consistency with the software design and requirements. (2.3) Completeness a. Verify that the following elements are in the source code, within the assumptions and constraints of the system: 1. Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging); 2. Process definition and scheduling; 3. Hardware, software, and user interface descriptions; 4. Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety, and security); 5. Critical configuration data; 6. System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing). b. Verify that the source code documentation satisfies specified configuration management procedures. (2.4) Accuracy a. Validate the logic, computational, and interface precision (e.g., truncation and rounding) in the system environment. b. Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws. (2.5) Readability a. Verify that the documentation is legible, understandable, and unambiguous to the intended audience. b. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, and symbols. (2.6) Testability a. Verify that there are objective acceptance criteria for validating each source code component. b. Verify that each source code component is testable against objective acceptance criteria.		
(3) Interface Analysis. Verify and validate that the software source code interfaces with hardware, user, operator, software, and other systems for correctness, consistency, completeness, accuracy, and testability. The task criteria are as follows: (3.1) Correctness a. Validate the external and internal software interface code in the context of system requirements.	Concept Documentation (System requirements) SDD IDD Source Code User Documentation	Task Report(s)—Interface Analysis Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.4 Implementation V&V Activity (development process) *(Continued)*		
(3.2) Consistency a. Verify that the interface code is consistent between source code components and to external interfaces (i.e., hardware, user, operator, and other software). (3.3) Completeness a. Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security). (3.4) Accuracy a. Verify that each interface provides information with the required accuracy. (3.5) Testability a. Verify that there are objective acceptance criteria for validating the interface code.		
(4) Criticality Analysis. Review and update the existing criticality analysis results from the prior Criticality Task Report using the source code. Implementation methods and interfacing technologies may cause previously assigned software integrity levels to be raised or lowered for a given software element (i.e., requirement, module, function, subsystem, other software partition). Verify that no inconsistent or undesired software integrity consequences are introduced by reviewing the revised software integrity levels.	Task Report(s)— Criticality Source Code	Task Report(s)—Criticality Analysis Anomaly Report(s)
(5) V&V Test Case Generation and Verification. *(For Software Integrity Levels 3 and 4.)* Develop V&V Test Cases for 1) component testing; 2) integration testing; 3) system testing; and 4) acceptance testing. Continue tracing required by the V&V Test Plans. Verify that the V&V Test Cases comply with Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the V&V Test Cases satisfy the criteria in V&V tasks 5.4.3 Task 5; 5.4.3 Task 6; 5.4.2 Task 5; and 5.4.2 Task 6 for component, integration, system, and acceptance testing, respectively. *(For Software Integrity Levels 1 and 2.)* Verify that the developer's Test Cases conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983). Validate that the developer's Test Cases satisfy the criteria in V&V tasks 5.4.3 Task 5; 5.4.3 Task 6; 5.4.2 Task 5; and 5.4.2 Task 6 for component (level 2 only); integration (levels 1 and 2); system (levels 1 and 2); and acceptance (level 2 only) testing, respectively.	SRS IRS SDD IDD User Documentation Test Design Test Cases	Component V&V Test Cases Integration V&V Test Cases System V&V Test Cases Acceptance V&V Test Cases Anomaly Report(s)
(6) V&V Test Procedure Generation and Verification. *(For Software Integrity Levels 3 and 4.)* Develop V&V Test Procedures for 1) component testing; 2) integration testing; and 3) system testing. Continue tracing required by the V&V Test Plans. Verify that the V&V Test Procedures comply with Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the V&V Test Procedures satisfy the criteria in V&V tasks 5.4.3 Task 5; 5.4.3 Task 6; and 5.4.2 Task 5 for component, integration, and system testing, respectively. *(For Software Integrity Levels 1 and 2.)* Verify that the developer's Test Procedures conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-	SRS IRS SDD IDD User Documentation Test Cases Test Procedures	Component V&V Test Procedures Integration V&V Test Procedures System V&V Test Procedures Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.4 Implementation V&V Activity (development process) *(Continued)*		
1983). Validate that the developer's Test Procedures satisfy the criteria in V&V tasks 5.4.3 Task 5; 5.4.3 Task 6; and 5.4.2 Task 5 for component (level 2 only); integration (levels 1 and 2); system (levels 1 and 2); and acceptance (level 2 only) testing, respectively.		
(7) Component V&V Test Execution and Verification. *(For Software Integrity Levels 3 and 4.)* Perform V&V component testing. Analyze test results to validate that software correctly implements the design. Validate that the test results trace to test criteria established by the test traceability in the test planning documents. Document the results as required by the Component V&V Test Plan. Use the V&V component test results to validate that the software satisfies the V&V test acceptance criteria. Document discrepancies between actual and expected test results. *(For Software Integrity Level 2.)* Use the developer's component test results to validate that the software satisfies the test acceptance criteria. *(For Software Integrity Level 1, there are no component test requirements.)*	Source Code Executable Code SDD IDD Component Test Plans Component Test Procedures Component Test Results	Task Report(s)— Test Results Anomaly Report(s)
(8) Hazard Analysis. Verify that the implementation and associated data elements correctly implement the critical requirements and introduce no new hazards. Update the hazard analysis.	Source Code SDD IDD Hazard Analysis Report	Task Report(s)— Hazard Analysis Anomaly Report(s)
(9) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce or mitigate the risks.	Source Code Supplier Development Plans and Schedules Hazard Analysis Report V&V task results	Task Report(s)— Risk Analysis Anomaly Report(s)
5.4.5 Test V&V Activity (development process)		
(1) Traceability Analysis. Analyze relationships in the V&V Test Plans, Designs, Cases, and Procedures for correctness and completeness. For correctness, verify that there is a valid relationship between the V&V Test Plans, Designs, Cases, and Procedures. For completeness, verify that all V&V Test Procedures are traceable to the V&V Test Plans.	V&V Test Plans V&V Test Designs V&V Test Procedures	Task Report(s)— Traceability Analysis Anomaly Report(s)
(2) Acceptance V&V Test Procedure Generation and Verification. *(For Software Integrity Levels 3 and 4.)* Develop Acceptance V&V Test Procedures. Continue the tracing required by the Acceptance V&V Test Plan. Verify that the V&V Test Procedures comply with Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983 [B5]). Validate that the Acceptance V&V Test Procedures satisfy the criteria in V&V task 5.4.2 Task 6. *(For Software Integrity Level 2.)* Verify that the developer's Acceptance Test Procedures conform to Project defined test document purpose, format, and content (e.g., see IEEE Std 829-1983). Validate that the developer's Test Procedures satisfy the criteria in V&V task 5.4.2 Task 6. *(For Software Integrity Level 1, there are no acceptance test requirements.)*	SDD IDD Source Code User Documentation Acceptance Test Plan Acceptance Test Procedures	Acceptance V&V Test Procedures Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.5 Test V&V Activity (development process) ***(Continued)***		
(3) Integration V&V Test Execution and Verification. *(For Software Integrity Levels 3 and 4.)* Perform V&V integration testing. Analyze test results to verify that the software components are integrated correctly. Validate that the test results trace to test criteria established by the test traceability in the test planning documents. Document the results as required by the Integration V&V Test Plan. Use the V&V integration test results to validate that the software satisfies the V&V test acceptance criteria. Document discrepancies between actual and expected test results. *(For Software Integrity Levels 1 and 2.)* Use the developer's integration test results to verify that the software satisfies the test acceptance criteria.	Source Code Executable Code Integration Test Plan Integration Test Procedures Integration Test Results	Task Report(s)—Test Results Anomaly Report(s)
(4) System V&V Test Execution and Verification. *(For Software Integrity Levels 3 and 4.)* Perform V&V system testing. Analyze test results to validate that the software satisfies the system requirements. Validate that the test results trace to test criteria established by the test traceability in the test planning documents. Document the results as required by the System V&V Test Plan. Use the V&V system test results to validate that the software satisfies the V&V test acceptance criteria. Document discrepancies between actual and expected test results. *(For Software Integrity Levels 1 and 2.)* Use the developer's system test results to verify that the software satisfies the test acceptance criteria.	Source Code Executable Code System Test Plan System Test Procedures System Test Results	Task Report(s)—Test Results Anomaly Report(s)
(5) Acceptance V&V Test Execution and Verification. *(For Software Integrity Levels 3 and 4.)* Perform acceptance V&V testing. Analyze test results to validate that the software satisfies the system requirements. Validate that the test results trace to test criteria established by the test traceability in the test planning documents. Document the results as required by the Acceptance V&V Test Plan. Use the acceptance V&V test results to validate that the software satisfies the V&V test acceptance criteria. Document discrepancies between actual and expected test results. *(For Software Integrity Level 2.)* Use the developer's acceptance test results to verify that the software satisfies the test acceptance criteria. *(For Software Integrity Level 1, there are no acceptance test requirements.)*	Source Code Executable Code User Documentation Acceptance Test Plan Acceptance Test Procedures Acceptance Test Results	Task Report(s)—Test Results Anomaly Report(s)
(6) Hazard Analysis. Verify that the test instrumentation does not introduce new hazards. Update the hazard analysis.	Source Code Executable Code Test Results Hazard Analysis Report	Task Report(s)— Hazard Analysis Anomaly Report(s)
(7) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks.	Supplier Development Plans and Schedules Hazard Analysis Report V&V task results	Task Report(s)—Risk Analysis Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.4.6 Installation and Checkout V&V Activity (development process)		
(1) Installation Configuration Audit. Verify that all software products required to correctly install and operate the software are present in the installation package. Validate that all site-dependent parameters or conditions to verify supplied values are correct.	Installation Package (e.g., Source Code, Executable Code, User Documentation, SDD, IDD, SRS, IRS, Concept Documentation, Installation Procedures, site-specific parameters, Installation Tests, and Configuration Management Data)	Task Report(s)—Installation Configuration Audit Anomaly Report(s)
(2) Installation Checkout. Conduct analyses or tests to verify that the installed software corresponds to the software subjected to V&V. Verify that the software code and databases initialize, execute, and terminate as specified. In the transition from one version of software to the next, the V&V effort shall validate that the software can be removed from the system without affecting the functionality of the remaining system components. The V&V effort shall verify the requirements for continuous operation and service during transition, including user notification.	User Documentation Installation Package	Task Report(s)—Installation Checkout Anomaly Report(s)
(3) Hazard Analysis. Verify that the installation procedures and installation environment does not introduce new hazards. Update the hazard analysis.	Installation Package Hazard Analysis Report	Task Report(s)—Hazard Analysis Anomaly Report(s)
(4) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks.	Installation Package Supplier Development Plans and Schedules V&V task results	Task Report(s)—Risk Analysis Anomaly Report(s)
(5) V&V Final Report Generation. Summarize in the V&V final report the V&V activities, tasks and results, including status and disposition of anomalies. Provide an assessment of the overall software quality and provide recommendations.	V&V Activity Summary Report (s)	V&V Final Report
5.5.1 Operation V&V Activity (operation process)		
(1) Evaluation of New Constraints. Evaluate new constraints (e.g., operational requirements, platform characteristics, operating environment) on the system or software requirements to verify the applicability of the SVVP. Software changes are maintenance activities (see 5.6.1).	SVVP New constraints	Task Report(s)—Evaluation of New Constraints
(2) Proposed Change Assessment. Assess proposed changes (e.g., modifications, enhancements, or additions) to determine the effect of the changes on the system. Determine the extent to which V&V tasks would be iterated.	Proposed Changes Installation Package	Task Report(s)—Proposed Change Assessment
(3) Operating Procedures Evaluation. Verify that the operating procedures are consistent with the user documentation and conform to the system requirements.	Operating Procedures User Documentation Concept Documentation	Task Report(s)—Operating Procedures Evaluation Anomaly Report(s)
(4) Hazard Analysis. Verify that the operating procedures and operational environment does not introduce new hazards. Update the hazard analysis.	Operating Procedures Hazard Analysis Report	Task Report(s)—Hazard Analysis Anomaly Report(s)

Table 1—V&V tasks, inputs, and outputs *(Continued)*

V&V tasks	Required inputs	Required outputs
5.5.1 Operation V&V Activity (operation process) *(Continued)*		
(5) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks.	Installation Package Proposed Changes Hazard Analysis Report Supplier Development Plans and Schedules Operation problem reports V&V task results	Task Report(s)—Risk Analysis Anomaly Report(s)
5.6.1 Maintenance V&V Activity (maintenance process)		
(1) SVVP Revision. Revise the SVVP to comply with approved changes. When the development documentation required by this standard is not available, generate a new SVVP and consider the methods in Annex D (V&V of reusable software) for deriving the required development documentation.	SVVP Approved Changes Installation Package Supplier Development Plans and Schedules	Updated SVVP
(2) Proposed Change Assessment. Assess proposed changes (i.e., modifications, enhancements, or additions) to determine the effect of the changes on the system. Determine the extent to which V&V tasks would be iterated.	Proposed Changes Installation Package Supplier Development Plans and Schedules	Task Report(s)—Proposed Change Assessment
(3) Anomaly Evaluation. Evaluate the effect of software operation anomalies.	Anomaly Report(s)	Task Report(s)—Anomaly Evaluation
(4) Criticality Analysis. Determine the software integrity levels for proposed modifications. Validate the integrity levels provided by the maintainer. For V&V planning purposes, the highest software integrity level assigned to the software shall be the software system integrity level.	Proposed Changes Installation Package Maintainer Integrity Levels	Task Report(s)—Criticality Analysis Anomaly Report(s)
(5) Migration Assessment. Assess whether the software requirements and implementation address 1) specific migration requirements, 2) migration tools, 3) conversion of software products and data, 4) software archiving, 5) support for the prior environment, and 6) user notification.	Installation Package Approved Changes	Task Report(s)—Migration Assessment Anomaly Report(s)
(6) Retirement Assessment. For software retirement, assess whether the installation package addresses: 1) software support, 2) impact on existing systems and databases, 3) software archiving, 4) transition to a new software product, and 5) user notification.	Installation Package Approved Changes	Task Report(s)—Retirement Assessment Anomaly Report(s)
(7) Hazard Analysis. Verify that software modifications correctly implement the critical requirements and introduce no new hazards. Update the hazard analysis.	Proposed Changes Installation Package Hazard Analysis Report	Task Report(s)—Hazard Analysis Anomaly Report(s)
(8) Risk Analysis. Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks.	Installation Package Proposed Changes Hazard Analysis Report Supplier Development Plans and Schedules Operation problem reports V&V task results	Task Report(s)—Risk Analysis Anomaly Report(s)
(9) Task Iteration. Perform V&V tasks, as needed, to ensure that 1) planned changes are implemented correctly; 2) documentation is complete and current; and 3) changes do not cause unacceptable or unintended system behaviors.	Approved Changes Installation Package	Task Report(s) Anomaly Report(s)

Table 2—Minimum V&V tasks assigned to each software integrity level

Life Cycle Processes	Acquisition				Supply				Development																				Operation				Maintenance			
V&V Activities	Acquisition Support V&V Activity				Planning V&V Activity				Concept V&V Activity				Requirements V&V Activity				Design V&V Activity				Implementation V&V Activity				Test V&V Activity				Installation/ checkout V&V Activity				Operation V&V Activity				Maintenance V&V Activity			
Software Integrity Levels	Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
V&V Tasks																																				
Acceptance V&V test execution and verification																									X	X	X									
Acceptance V&V test plan generation and verification													X	X	X																					
Acceptance V&V test procedure generation and verification																									X	X	X									
Anomaly evaluation																																	X	X	X	
Component V&V test execution and verification																					X	X	X													
Component V&V test plan generation and verification																	X	X	X																	
Concept documentation evaluation									X	X	X																									
Configuration management assessment													X	X																						
Contract verification					X																															
Criticality analysis									X	X	X		X	X	X		X	X	X		X	X	X										X	X	X	
Evaluation of new constraints																													X	X	X					
Hardware/software/User requirements allocation analysis									X																											

Table 2—Minimum V&V tasks assigned to each software integrity level *(Continued)*

Life Cycle Processes	Acquisition				Supply				Development																				Operation				Maintenance			
V&V Activities	Acquisition Support V&V Activity				Planning V&V Activity				Concept V&V Activity				Requirements V&V Activity				Design V&V Activity				Implementation V&V Activity				Test V&V Activity				Installation/ checkout V&V Activity				Operation V&V Activity				Maintenance V&V Activity			
Software Integrity Levels	Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
Hazard analysis									X	X			X	X			X	X			X	X			X	X			X	X			X	X		
Installation checkout																													X	X						
Installation configuration audit																													X	X						
Interface analysis													X	X	X		X	X	X		X	X	X													
Integration V&V test execution and verification																									X	X	X	X								
Management of V&V																																				
a) Baseline change assessment									X				X	X	X		X	X	X		X	X	X		X	X	X		X	X	X		X	X		
b) Interface with organizational supporting processes	X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		
c) Management and technical review support									X	X			X	X			X	X			X	X			X	X			X	X			X	X		
d) Management review of V&V	X	X	X		X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
e) Software V&V plan (SVVP) generation									X	X	X	X																					X	X	X	X
Migration assessment																																	X	X		
Operation procedures evaluation									X	X	X	X																								
Planning the interface between the V&V effort and supplier	X	X	X		X	X	X																													
Proposed change assessment																													X	X	X		X	X	X	
Risk analysis									X	X			X	X			X	X			X	X			X	X			X	X			X	X		

Table 2—Minimum V&V tasks assigned to each software integrity level *(Continued)*

Life Cycle Processes	Acquisition				Supply				Development																				Operation				Maintenance			
V&V Activities	Acquisition Support V&V Activity				Planning V&V Activity				Concept V&V Activity				Requirements V&V Activity				Design V&V Activity				Implementation V&V Activity				Test V&V Activity				Installation/ checkout V&V Activity				Operation V&V Activity				Maintenance V&V Activity			
Software Integrity Levels	Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
Retirement assessment																																	X	X		
Scoping the V&V effort	X	X	X																																	
Software design evaluation																	X	X	X	X																
Software requirements evaluation													X	X	X	X																				
SVVP revision																																	X	X	X	X
Source code and source code documentation evaluation																					X	X	X	X												
System requirements review	X	X	X	X																																
System V&V test execution and verification																									X	X	X	X								
System V&V test plan generation and verification													X	X	X	X																				
Task iteration																																	X	X	X	X
Traceability analysis									X	X	X		X	X	X		X	X	X		X	X	X		X	X	X									
V&V final report generation																													X	X	X					
V&V test design generation and verification																																				
a) Component																	X	X	X																	
b) Integration																	X	X	X	X																
c) System																	X	X	X	X																
d) Acceptance																	X	X	X																	

Table 2—Minimum V&V tasks assigned to each software integrity level *(Continued)*

Life Cycle Processes	Acquisition				Supply				Development																Operation				Maintenance			
V&V Activities	Acquisition Support V&V Activity				Planning V&V Activity				Concept V&V Activity				Requirements V&V Activity				Design V&V Activity				Implementation V&V Activity				Test V&V Activity				Installation/ checkout V&V Activity				Operation V&V Activity				Maintenance V&V Activity			
Software Integrity Levels	Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels				Levels			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
V&V test case generation and verification																																								
a) Component																					X	X	X																	
b) Integration																					X	X	X	X																
c) System																					X	X	X	X																
d) Acceptance																					X	X	X																	
V&V test procedure generation and verification																																								
a) Component																					X	X	X																	
b) Integration																					X	X	X	X																
c) System																					X	X	X	X																

Table 3—Optional V&V tasks and suggested applications in the life cycle [a]

Life cycle processes	Acquisition	Supply	Management	Concept	Requirements	Design	Implementation	Test	Installation and Checkout	Operation	Maintenance
Algorithm Analysis					X	X	X				X
Audit Performance					X	X	X	X	X		X
Audit Support			X		X	X	X	X	X		X
Control Flow Analysis					X	X	X				X
Cost Analysis	X	X	X	X	X	X	X	X	X		X
Database Analysis					X	X	X	X			X
Data Flow Analysis					X	X	X				X
Disaster Recovery Plan Assessment			X	X	X	X	X			X	X
Distributed Architecture Assessment				X	X	X					X
Feasibility Study Evaluation	X	X	X	X	X	X					X
Independent Risk Assessment	X	X	X	X	X	X	X	X	X	X	X
Inspection											
Concept					X	X	X	X	X		X
Design						X					X
Requirements					X						X
Source code							X				X
Test case						X	X	X	X		X
Test design						X	X		X		X
Test plan					X	X	X		X		X
Operational Evaluation										X	
Performance Monitoring				X	X	X	X	X	X	X	X
Post Installation validation									X	X	X
Project Management Oversight Support	X	X	X	X	X	X	X	X	X	X	X
Qualification Testing								X	X		X
Regression Analysis and Testing					X	X	X	X	X		X
Reusability Assessment	X	X	X	X	X	X					X
Security Assessment	X		X	X	X	X	X	X	X		X
Simulation Analysis				X	X	X	X	X	X	X	X
Sizing and Timing Analysis					X	X	X	X			X
System Software Assessment							X	X	X	X	X
Test Certification								X	X	X	X
Test Evaluation					X	X	X	X	X	X	X
Test Witnessing								X	X	X	X
Training Document Evaluation					X	X	X	X	X	X	X
User Documentation Evaluation			X	X	X	X	X	X	X	X	X

Table 3—Optional V&V tasks and suggested applications in the life cycle *(Continued)*[a]

Life cycle processes	Acquisition	Supply	Management	Concept	Requirements	Design	Implementation	Test	Installation and Checkout	Operation	Maintenance
User training			X					X	X	X	X
V&V tool plan generation	X	X	X								X
Walkthroughs											
Design						X					X
Requirements					X						X
Source code							X				X
Test								X	X		X

[a]Annex G contains a description of the optional V&V tasks.

Annex A

(informative)

Mapping of ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks

Table A.1 shows a mapping of all ISO/IEC 12207 [B16] V&V requirements (i.e., processes, activities, and tasks) to the V&V activities and tasks of this standard.

The first column of Table A.1 lists the ISO/IEC 12207 section numbers and titles of V&V processes and activities. The second column of Table A.1 lists the IEEE Std 1012 clauses, subclauses, tables, and annexes that address the topics listed in the first column. Where no subclause titles were assigned, clause titles were created to reflect the clause contents. These derived titles are marked by an [a].

Table A.1—Mapping

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V Activities and Tasks	
	Location	Description
5.1.4.1 Supplier Monitoring V&V	Subclause 5.2.1 Table 1, Tasks 1, 2, and 3	Activity: Acquisition Support V&V Acquisition Support V&V Tasks
5.2.4.5h) and 5.2.5.5 Interfacing with V&V[a]	Subclause 5.2.1 Table 1, Task 2 Subclause 5.3.1 Table 1, Task 1 Annex C	Activity: Acquisition Support V&V Planning the Interface Between the V&V Effort and Supplier Activity: Planning V&V Planning the Interface Between the V&V Effort and Supplier Definition of IV&V
5.2.6.3 Verification and validation[a]	All clauses, tables, figures, and annexes	Software V&V
5.3.2 System Requirements Analysis	Subclause 5.2.1 Table 1, Task 3 Subclause 5.4.1 Table 1, Tasks 1 and 4	Activity: Acquisition Support V&V System Requirements Review Activity: Concept V&V Concept V&V Tasks (Concept Documentation Evaluation, Traceability Analysis)
5.5.5 and 5.5.6 Migration and Software Retirement	Subclause 5.6.1 Table 1, Task 2	Activity: Maintenance V&V Proposed Change Assessment
6.4.1 Verification Process Implementation[a]	Clause 4 Clauses 6 and 7 Subclauses 6.2 and 7.7	V&V software integrity levels Software V&V reporting, administrative, and documentation requirements; SVVP outline V&V administrative requirements
6.4.1.1 Criticality of Software to be Verified[a]	Clause 4 Table B.1 Table B.2 Annex D	V&V software integrity levels Assignment of software integrity levels Definitions of consequences V&V of reusable software
6.4.1.2 Process for Verification[a]	Clauses 6 and 7	Software V&V reporting, administrative, and documentation requirements; SVVP outline
6.4.1.3 and 6.4.1.4 Extent and Rigor of Verification[a]	Table 2 Annex C	Minimum V&V tasks assigned for each software integrity level Definition of IV&V

Table A.1—Mapping *(Continued)*

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V Activities and Tasks	
	Location	Description
6.4.1.5 Verification Plan[a]	Clauses 6 and 7	Software V&V reporting, administrative, and documentation requirements; SVVP outline
6.4.1.6 Problem and Non-conformance Reports[a]	Subclauses 6.2 and 7.7	V&V administrative requirements
6.4.2 Verification	Clause 5	V&V processes
6.4.2.1 Contract Verification	Subclause 5.2.1 Table 1, Task 2	Activity: Planning V&V Contract Verification
6.4.2.2 Process Verification	Subclause 5.2 Subclause 5.3 Subclause 5.4	Process: Acquisition Process: Supply Process: Development
6.4.2.3 Requirements Verification	Subclause 5.2.1 Table 1, Task 3 Subclause 5.4.1 Table 1, Task 1 Subclause 5.4.2 Table 1, Tasks 1–9	Activity: Acquisition Support V&V System Requirements Review Activity: Concept V&V Concept Documentation Evaluation Activity: Requirements V&V Requirements V&V Tasks
6.4.2.4 Design Verification	Subclause 5.4.3 Table 1, Tasks 1–9	Activity: Design V&V Design V&V Tasks
6.4.2.5 Code Verification	Subclause 5.4.4 Table 1, Tasks 1–9	Activity: Implementation V&V Implementation V&V Tasks
6.4.2.6 Integration Verification	Subclause 5.4.5 Table 1, Task 3	Activity: Test V&V Test V&V Tasks
6.4.2.7 Documentation Verification	Subclause 5.2.1 Table 1, Task 3 Subclause 5.3.1 Table 1, Task 2 Subclause 5.4.1 Table 1, Task 1 Subclause 5.4.2 Table 1, Tasks 2 and 3 Subclause 5.4.3 Table 1, Tasks 2 and 3 Subclause 5.4.4 Table 1, Tasks 2 and 3 Subclause 5.4.6 Table 1, Task 1 Subclause 5.5.1 Table 1, Task 3	Activity: Acquisition Support V&V Systems Requirements Review Activity: Planning V&V Contract Verification Activity: Concept V&V Concept Documentation Evaluation Activity: Requirements V&V Software Requirements Evaluation and Interface Analysis Activity: Design V&V Software Design Evaluation and Interface Analysis Activity: Implementation V&V Source Code and Source Code Documentation Evaluation and Interface Analysis Activity: Installation and Checkout Installation Configuration Audit Activity: Operation V&V Operating Procedure Evaluation
6.5.1 Validation Process Implementation[a]	Clause 4 Clauses 6 and 7 Subclauses 6.2 and 7.7 Annexes C, D, and E	V&V software integrity levels Software V&V reporting, administrative, and documentation requirements; SVVP outline V&V administrative requirements Definition of IV&V, V&V of reusable software, and V&V metrics
6.5.1.1 Criticality of Software to be Validated[a]	Clause 4 Table B.1 Table B.2 Annex D	V&V software integrity levels Assignment of software integrity levels Definitions of consequences V&V of reusable software
6.5.1.2 Process for Validation[a]	Clauses 6 and 7	Software V&V reporting, administrative, and documentation requirements; SVVP outline

Table A.1—Mapping *(Continued)*

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V Activities and Tasks	
	Location	**Description**
6.5.1.3 Extent and Rigor of Validation[a]	Table 2 Annex C	Minimum V&V tasks assigned for each software integrity level Definition of IV&V
6.5.1.4 Validation Plan[a]	Clauses 6 and 7	Software V&V reporting, administrative, and documentation requirements; SVVP outline
6.5.1.5 Problem and Non-conformance Reports[a]	Subclause 6.2 and 7.7	V&V administrative requirements
6.5.2 Validation	Clause 5	V&V processes
6.5.2.1 Validate Test Preparation[a]	Subclause 5.4.2 Table 1, Tasks 5 and 6 Subclause 5.4.3 Table 1, Tasks 5, 6, and 7 Subclause 5.4.4 Table 1, Tasks 5 and 6 Subclause 5.4.5 Table 1, Task 2	Activity: Requirements V&V System V&V Test Plan Generation and Verification, and Acceptance V&V Test Plan Generation and Verification Activity: Design V&V Component V&V Test Plan Generation and Verification, Integration V&V Test Plan Generation and Verification, and V&V Test Designs Generation and Verification Activity: Implementation V&V V&V Test Cases Generation and Verification, and V&V Test Procedure Generation and Verification Activity: Test V&V Acceptance V&V Test Procedure Generation and Verification
6.5.2.2 Validate Test Traceability[a]	Subclause 5.4.4 Table 1, Task 7 Subclause 5.4.5 Table 1, Tasks 3, 4, and 5	Activity: Implementation V&V Component V&V Test Execution and Verification Activity: Test V&V Test V&V Tasks
6.5.2.3 Validate Test Conduction[a]	Subclause 5.4.4 Table 1, Task 7 Section 5.4.5 Table 1, Tasks 3, 4, and 5	Activity: Implementation V&V Component V&V Test Execution and Verification Activity: Test V&V Test V&V Tasks
6.5.2.4 Validate Software for Intended Use[a]	Subclause 5.4.1 Table 1, Task 1 Subclause 5.4.2 Table 1, Tasks 2 and 3 Subclause 5.4.3 Table 1, Tasks 2 and 3 Subclause 5.4.4 Table 1, Tasks 2 and 3 Subclause 5.4.5 Table 1, Tasks 4 and 5	Activity: Concept V&V Concept Documentation Evaluation Activity: Requirements V&V Software Requirements Evaluation and Interface Analysis Activity: Design V&V Software Design Evaluation and Interface Analysis Activity: Implementation V&V Source Code and Source Code Documentation Evaluation, and Interface Analysis Activity: Test V&V System V&V Test Execution and Verification, and Acceptance V&V Test Execution and Verification
6.5.2.5 Installation Test of Software[a]	Subclause 5.4.6, Table 1, Tasks 1–4	Activity: Installation and Checkout V&V Installation and Checkout V&V Tasks

[a]No ISO/IEC 12207 clause title was listed. For purposes of this mapping, a clause title was assigned to reflect the clause contents.

This standard defines 11 V&V activities, as shown in column 1 of Table A.2, that are part of the V&V processes. Each of the 11 V&V activities supports the ISO/IEC 12207 software life cycle processes and activities shown in columns 2 and 3 of Table A.2.

Table A.2—Mapping of 1012 V&V activities to ISO/IEC 12207 software life cycle processes and activities

1012 V&V activities	ISO/IEC 12207 software life cycle	
	Processes	Activities
Acquisition Support V&V	Acquisition	—Initiation —Request-for-Proposal (-tender) Preparation —Contract Preparation Update —Supplier Monitoring —Acceptance and Completion
Planning V&V	Supply	—Initiation —Preparation of Response —Contract —Planning —Execution and Control —Review and Evaluation —Delivery and Completion
Concept V&V	Development	—Process Implementation —System Requirements Analysis —System Architectural Design
Requirements V&V	Development	—Software Requirements Analysis
Design V&V	Development	—Software Architectural Design —Software Detailed Design
Implementation V&V	Development	—Software Coding and Testing
Test V&V	Development	—Software Integration —Software Qualification Testing —System Integration —System Qualification Testing
Installation and Checkout V&V	Development	—Software Installation —Software Acceptance Support
Operation V&V	Operation	—Process Implementation —Operational Testing —System Operation —User Support
Maintenance V&V	Maintenance	—Process Implementation —Problem and Modification Analysis —Modification Implementation —Maintenance Review/Acceptance —Migration —Software Retirement
Management of V&V	All processes	—All activities

Table A.3 shows a mapping of all IEEE Std 1074-1997 V&V requirements (i.e., processes, activities, and tasks) to the V&V activities and tasks of this standard.

Table A.3—Mapping IEEE Std 1074-1997 V&V requirements to IEEE Std 1012 V&V activities and tasks

IEEE Std 1074-1997 V&V requirements	IEEE 1012 V&V activities and tasks	
	Location	Description
A.1 Project Management Activity	Clause 5, 6, and 7	V&V processes; Software V&V reporting, administrative, and documentation requirements; SVVP outline
A.1.1 Project Initiation Activities	Subclause 5.2.1	Activity: Acquisition Support V&V
A.1.1.1 Create Software Life Cycle Process	Table 1, Task 1	Scoping the V&V Effort
A.1.1.2 Perform Estimation	Table 1, Task 2	Planning the Interface Between the V&V Effort and Supplier
A.1.1.3 Allocate Project Resources		
A.1.1.4 Define Metrics	Subclause 5.3.1	Activity: Planning V&V
	Table 1, Task 1	Planning the Interface Between the V&V Effort and Supplier
	Subclause 5.1.1	Activity: Management of V&V
	Table 1, Task 1	SVVP Generation
	Subclause 5.6.1	Activity: Maintenance V&V
	Table 1, Task 1	SVVP Revision
	Subclauses 6.1 and 7.6	V&V reporting requirements
	Subclauses 6.2 and 7.7	V&V administrative requirements
	Subclause 6.3.1 and 7.8	V&V Test documentation
	Subclauses 6.3.2 and 7	SVVP documentation
	Annex E	V&V metrics
A.1.2 Project Planning Activities	Clauses 5, 6, and 7	V&V processes; Software V&V reporting, administrative, and documentation requirements; SVVP outline
A.1.2.1 Plan Evaluations		
A.1.2.2 Plan Configuration Management		
A.1.2.3 Plan System Transition (if applicable)	Table 1, All Tasks	V&V Tasks
A.1.2.4 Plan Installation		
A.1.2.5 Plan Documentation		
A.1.2.6 Plan Training		
A.1.2.7 Plan Project Managment		
A.1.3 Project Monitoring and Control Activities	Subclause 5.1.1	Activity: Management of V&V
A.1.3.1 Manage Risks	Table 1, Tasks 1, 2, 3, 4, and 5	Management of V&V Tasks
A.1.3.2 Manage the Project		
A.1.3.3 Identify SLCP Improvement Needs		
A.1.3.4 Retain Records		
A.1.3.5 Collect and Analyze Metric Data		
A.2 Pre-development Activity	Subclause 5.4.1	Activity: Concept V&V
A.2.1 Concept Exploration Activities	Subclause 5.4.1	Activity: Concept V&V
A.2.1.1 Identify Ideas or Needs	Table 1, Tasks 1, 2, 4, 5, and 6	Tasks: Concept Documentation Evaluation, Criticality Analysis, Traceability Analysis, Hazard Analysis, and Risk Analysis
A.2.1.2 Formulate Potential Approaches		
A.2.1.3 Conduct Feasibility Studies		
A.2.1.4 Refine and Finalize the Idea or Need		
A.2.2 System Allocation Activities	Subclause 5.4.1	Activity: Concept V&V
A.2.2.1 Analyze Functions	Table 1, Tasks 1 and 3	Tasks: Concept Documentation Evaluation, and Hardware/Software/User Requirements Allocation Analysis
A.2.2.2 Develop System Architecture		
A.2.2.3 Decompose System Requirements		

Table A.3—Mapping IEEE Std 1074-1997 V&V requirements to IEEE Std 1012 V&V activities and tasks *(Continued)*

IEEE Std 1074-1997 V&V requirements	IEEE 1012 V&V activities and tasks	
	Location	**Description**
A.2.3 Software Importation Activities A.2.3.1 Identify Imported Software Requirements A.2.3.2 Evaluate Software Imported Source (if applicable) A.2.3.3 Define Software Import Method (if applicable) A.2.3.4 Import Software	Annex D Table 1, All Tasks	V&V of reusable software V&V Tasks
A.3 Development Activity	Subclauses 5.4.2, 5.4.3, 5.4.4, and 5.4.5	Activities: Requirements, Design, Implementation, and Test V&V
A.3.1 Requirements Activities A.3.1.1 Define and Develop Software Requirements A.3.1.2 Define Interface Requirements A.3.1.3 Prioritize and Integrate Software Requirements	Subclause 5.4.2 Table 1, Tasks 1–9	Activity: Requirements V&V Requirements V&V Tasks
A.3.2 Design Activities A.3.2.1 Perform Architectural Design A.3.2.2 Design Data Base (if applicable) A.3.2.3 Design Interfaces A.3.2.4 Perform Detailed Design	Subclause 5.4.3 Table 1, Tasks 1–9	Activity: Design V&V Design V&V Tasks
A.3.3 Implementation Activities A.3.3.1 Create Executable Code A.3.3.2 Create Operating Documentation A.3.3.3 Perform Integration	Subclause 5.4.4 Table 1, Tasks 1–9 Subclause 5.4.5 Table 1, Tasks 1–7	Activity: Implementation V&V Implementation V&V Tasks Activity: Test V&V Test V&V Tasks
A.4 Post-development Activity	Subclauses 5.4.6, 5.5.1, and 5.6.1	Activities: Installation and Checkout, Operation, and Maintenance V&V
A.4.1 Installation Activities A.4.1.1 Distribute Software A.4.1.2 Install Software A.4.1.3 Accept Software in Operational Environment	Subclause 5.4.6 Table 1, Tasks 1–5	Activity: Installation and Checkout V&V Installation and Checkout V&V Tasks
A.4.2 Operation and Maintenance Activities A.4.2.1 Operate the System A.4.2.2 Provide Technical Assistance and Consulting A.4.2.3 Maintain Support Request Log	Subclause 5.5.1 Table 1, Tasks 1–9	Activity: Operation V&V Operation V&V Tasks
A.4.3 Maintenance Activities A.4.3.3 Reapply Software Life Cycle	Subclause 5.6.1 Table 1, Tasks 1–9	Activity: Maintenance V&V Maintenance V&V Tasks
A.4.4 Retirement Activities A.4.4.2 Conduct Parallel Operations (if applicable) A.4.4.3 Retire System	Subclause 5.6.1	Activity: Maintenance V&V
A.5 Integral Activity A.5.1 Evaluation Activities A.5.1.1 Conduct Reviews A.5.1.2 Create Traceability Matrix A.5.1.3 Conduct Audits A.5.1.4 Develop Test Procedures A.5.1.5 Create Test Data A.5.1.6 Execute Tests A.5.1.7 Report Evaluation Results	Clauses 5, 6, and 7 Table 1, All Tasks	V&V processes; Software V&V reporting, administrative, and documentation requirements; SVVP outline

Annex B

(informative)

A software integrity level scheme

Table B.1 defines four software integrity levels used as an illustration by this standard. Table B.2 describes the consequences of software errors for each of the four software integrity levels. There are overlaps between the software integrity levels to allow for individual interpretations of acceptable risk depending on the application. A software integrity level 0 (zero) may be assigned if there are no consequences associated with a software error that may occur in the system. For software integrity level 0, no V&V tasks are implemented.

Table B.1—Assignment of software integrity levels

Software integrity level	Description
4	An error to a function or system feature that causes catastrophic consequences to the system with reasonable, probable, or occasional likelihood of occurrence of an operating state that contributes to the error; or critical consequences with reasonable or probable likelihood of occurrence of an operating state that contributes to the error.
3	An error to a function or system feature that causes catastrophic consequences with occasional or infrequent likelihood of occurrence of an operating state that contributes to the error; or critical consequences with probable or occasional likelihood of occurrence of an operating state that contributes to the error; or marginal consequences with reasonable or probable likelihood of occurrence of an operating state that contributes to the error.
2	An error to a function or system feature that causes critical consequences with infrequent likelihood of occurrence of an operating state that contributes to the error; or marginal consequences with probable or occasional likelihood of occurrence of an operating state that contributes to the error; or negligible consequences with reasonable or probable likelihood of occurrence of an operating state that contributes to the error.
1	An error to a function or system feature that causes critical consequences with infrequent likelihood of occurrence of an operating state that contributes to the error; or marginal consequences with occasional or infrequent occurrence of an operating state that contributes to the error; or negligible consequences with probable, occasional, or infrequent likelihood of occurrence of an operating state that contributes to the error.

Table B.2—Definitions of consequences

Consequence	Definitions
Catastrophic	Loss of human life, complete mission failure, loss of system security and safety, or extensive financial or social loss.
Critical	Major and permanent injury, partial loss of mission, major system damage, or major financial or social loss.
Marginal	Severe injury or illness, degradation of secondary mission, or some financial or social loss.
Negligible	Minor injury or illness, minor impact on system performance, or operator inconvenience.

Table B.3 illustrates the risk-based scheme shown in Tables B.1 and B.2. Each cell in the table assigns a software integrity level based upon the combination of an error consequence and the likelihood of occurrence of an operating state that contributes to the error. Some table cells reflect more than one software integrity level indicating that the final assignment of the software integrity level can be selected to address the system application and risk mitigation recommendations. For some industry applications, the definition of likelihood of occurrence categories may be expressed as probability figures derived by analysis or from system requirements.

Table B.3—A graphic illustration of the assignment of software integrity levels

Error consequence	Likelihood of occurrence of an operating state that contributes to the error			
	Reasonable	Probable	Occasional	Infrequent
Catastrophic	4	4	4 or 3	3
Critical	4	4 or 3	3	2 or 1
Marginal	3	3 or 2	2 or 1	1
Negligible	2	2 or 1	1	1

Annex C

(informative)

Definition of independent verification and validation (IV&V)

IV&V is defined by three parameters: technical independence, managerial independence, and financial independence.

C.1 Technical independence

Technical independence requires the V&V effort to utilize personnel who are not involved in the development of the software. The IV&V effort must formulate its own understanding of the problem and how the proposed system is solving the problem. Technical independence ("fresh viewpoint") is an important method to detect subtle errors overlooked by those too close to the solution.

For software tools, technical independence means that the IV&V effort uses or develops its own set of test and analysis tools separate from the developer's tools. Sharing of tools is allowable for computer support environments (e.g., compilers, assemblers, utilities) or for system simulations where an independent version would be too costly. For shared tools, IV&V conducts qualification tests on tools to ensure that the common tools do not contain errors that may mask errors in the software being analyzed and tested.

C.2 Managerial independence

This requires that the responsibility for the IV&V effort be vested in an organization separate from the development and program management organizations. Managerial independence also means that the IV&V effort independently selects the segments of the software and system to analyze and test, chooses the IV&V techniques, defines the schedule of IV&V activities, and selects the specific technical issues and problems to act upon. The IV&V effort provides its findings in a timely fashion simultaneously to both the development and program management organizations. The IV&V effort must be allowed to submit to program management the IV&V results, anomalies, and findings without any restrictions (e.g., without requiring prior approval from the development group) or adverse pressures, direct or indirect, from the development group.

C.3 Financial independence

This requires that control of the IV&V budget be vested in an organization independent of the development organization. This independence prevents situations where the IV&V effort cannot complete its analysis or test or deliver timely results because funds have been diverted or adverse financial pressures or influences have been exerted.

C.4 Forms of independence

The extent to which each of the three independence parameters (technical, managerial, and financial) is vested in a V&V organization determines the degree of independence achieved.

Many forms of independence can be adopted for a V&V organization. The four most prevalent are 1) Classical; 2) Modified; 3) Internal; and 4) Embedded. Table C.1 illustrates the degree of independence achieved by these four forms.

Table C.1—Forms of IV&V

Form of IV&V	Technical	Management	Financial
Classical	I[a]	I	I
Modified	I	"I"[b]	I
Internal	"I"	"I"	"I"
Embedded	i[c]	i	i

[a]I = rigorous independence.
[b]"I" = independence with qualifications.
[c]i = minimal maintenance.

C.4.1 Classical IV&V

Classical IV&V embodies all three independence parameters. The IV&V responsibility is vested in an organization that is separate from the development organization. IV&V uses a close working relationship with the development organization to ensure that IV&V findings and recommendations are integrated rapidly back into the development process. Typically, Classical IV&V is performed by one organization (e.g., supplier) and the development is performed by a separate organization (i.e., another vendor). Classical IV&V is generally required for software integrity level 4 (i.e., loss of life, loss of mission, significant social or financial loss) through regulations and standards imposed on the system development.

C.4.2 Modified IV&V

Modified IV&V is used in many large programs where the system prime integrator is selected to manage the entire system development including the IV&V. The prime integrator selects organizations to assist in the development of the system and to perform the IV&V. In the modified IV&V form, the acquirer reduces its own acquisition time by passing this responsibility to the prime integrator. Since the prime integrator performs all or some of the development, the managerial independence is compromised by having the IV&V effort report to the prime integrator. Technical independence is preserved since the IV&V effort formulates an unbiased opinion of the system solution and uses a independent staff to perform the IV&V. Financial independence is preserved since a separate budget is set aside for the IV&V effort. Modified IV&V effort would be appropriate for systems with software integrity level 3 (i.e., an important mission and purpose).

C.4.3 Internal IV&V

Internal IV&V exists when the developer conducts the IV&V with personnel from within its own organization, although not necessarily those personnel involved directly in the development effort. Technical, managerial, and financial independence are compromised. Technical independence is compromised because the IV&V analysis and test is vulnerable to overlooking errors by using the same assumptions or development environment that masked the error from the developers. Managerial independence is compromised because the internal IV&V effort uses the same common tools and corporate analysis procedures as the development group. Peer pressure from the development group may adversely influence how aggressively the software is analyzed and tested by the IV&V effort. Financial independence is compromised because the development group controls the IV&V budget. IV&V funds, resources, and schedules may be reduced as development pressures and needs redirect the IV&V funds into solving development problems. The benefit of an internal IV&V effort is access to staff who know the system and its software. This form of IV&V is used when the

degree of independence is not explicitly stated and the benefits of preexisting staff knowledge outweigh the benefits of objectivity.

C.4.4 Embedded V&V

This form is similar to Internal IV&V in that it uses personnel from the development organization who should preferably not be involved directly in the development effort. Embedded V&V is focused on ensuring compliance with the development procedures and processes. The Embedded V&V organization works side by side with the development organization and attends the same inspections, walkthroughs, and reviews as the development staff (i.e., compromise of technical independence). Embedded V&V is not tasked specifically to independently assess the original solution or conduct independent tests (i.e., compromise of managerial independence). Financial independence is compromised because the IV&V staff resource assignments are controlled by the development group. Embedded V&V allows rapid feedback of V&V results into the development process but compromises the technical, managerial, and financial independence of the V&V organization.

Annex D

(informative)

V&V of reusable software

This annex provides guidelines for conducting V&V of reusable software. Reusable software (in part or whole) includes software from software libraries, custom software developed for other applications, legacy software, or commercial-off-the-shelf (COTS) software.

The V&V tasks of Table 1 are applied to reusable software just as they are applied to newly developed software. However, the inputs for these tasks may not be available for reusable software, reducing visibility into the software products and processes. For example, source code may not be available for evaluation, the documentation may be incomplete, or the development process may not be known. The inputs for V&V of reusable software should be obtained from any source available. Some examples of sources for such inputs are provided below.

— Audit results
— Black box testing results
— Design process documentation
— Engineering judgment
— Operational history
— Original developers' interviews
— Prior hazard analysis results
— Prior V&V results
— Product documentation
— Prototyping results
— Reverse engineering results
— Software developer's notebook
— Software integrity level
— Standards complied with
— Static code analysis results
— Test history
— Trial integration results
— User interviews

If V&V of reusable software cannot be carried out at the appropriate level, the reusable software may be used so long as the risk associated with this use is recognized and accounted for in the risk mitigation strategy. Substitution of Table 1 V&V tasks is permitted if equivalent alternative V&V tasks can be shown to satisfy the same criteria as in Table 1.

Annex E

(informative)

V&V metrics

The V&V metrics should consider the software integrity level assigned to the software and system, application domain, project needs, and current industry practices.

This standard considers two categories of metrics: 1) metrics for evaluating software development processes and products; and 2) metrics for evaluating V&V task results, and for improving the quality and coverage of V&V tasks. Values of metrics should be established to serve as indicators as to whether a process, product, or V&V task has been satisfactorily accomplished.

E.1 Metrics for evaluating software development processes and products

The use of metrics should be considered as a V&V approach to evaluating the software development processes and products. By computing evaluation metrics over a period of time, problematic trends can be identified. No standard set of metrics is applicable for all projects so the use of metrics may vary according to the application domain and software development environment.

IEEE Std 1061-1992 [B11] provides a standard definition of available software quality metrics. Other metric-related standards, such as IEEE Std 982.1-1988 [B6] and its corresponding guide, IEEE Std 982.2-1988 [B7], may also be used. The following is a list of metrics that have been found useful. This list is not intended to be exhaustive.

1) Completeness of information (e.g., concept, requirements, design)
2) Software size
3) Requirements traceability
4) Number of changes (e.g., requirements, design, code)
5) Logic and data complexity
6) Analysis or test coverage (type of coverage is based on project and application needs and may consist of requirements, code, functional, module, and test cases)
7) Control and data coupling
8) Status of actual vs. planned progress
9) Number of defects discovered over time
10) Period in the development process when the defect is detected
11) Defect category
12) Severity of defect
13) Systemic or repeated errors having the same cause (such as process deficiencies and tool errors)
14) Time to fix a defect (impact to schedule)

E.2 Metrics for evaluating V&V tasks and for improving the quality and coverage of V&V tasks

No consensus exists on V&V metrics. Candidate metrics to consider fall into two categories:

1) *V&V quality*—to measure the quality and effectiveness of the V&V task (e.g, the ratio of the number of defects identified by V&V to the number of defects missed);

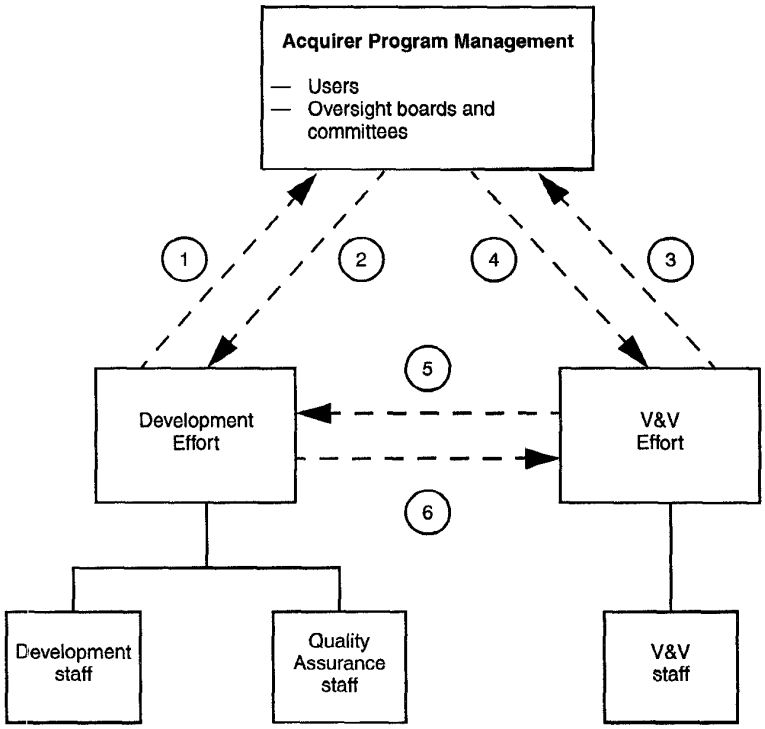
2) *V&V coverage*—to measure the extent or breadth of the application of V&V (e.g., the ratio of the number of software modules verified and validated to the total number of modules).

The Management of V&V activity uses these metric results to change V&V project resources for V&V tasks showing the need for process assistance. These and similar V&V metrics can be used to assess the quality and coverage of V&V tasks. They can be used to provide feedback to the continuous improvement of the V&V processes.

Annex F

(informative)

Example of V&V organizational relationship to other project responsibilities



NOTES—The numbered lines represent the flow of control and data as defined below:

1) Submittal of program documentation (e.g., concept, requirements, design, users manuals) source code, program status, program budgets, and development plans and schedules.

2) Approval, denial, and recommendations on development issues and deliverables listed in #1.

3) Submittal of SVVP, V&V task results, anomaly reports, activity summary reports, and other special reports.

4) Approval, denial, and recommendations on V&V issues and deliverables listed in #3.

5) Submittal of V&V task results, anomaly reports, activity summary reports, and special reports as directed by the acquirer program management.

6) Submittal of program documentation (e.g., concept, requirements, design, users manuals, special reports, source code, and program schedules).

Figure F.1—Example of V&V organizational relationship to other project responsibilities

Annex G

(informative)

Optional V&V task descriptions

Algorithm analysis. Verify the correct implementation of algorithms, equations, mathematical formulations, or expressions. Rederive any significant algorithms, and equations from basic principles and theories. Compare against established references or proven past historical data. Validate the algorithms, equations, mathematical formulations, or expressions with respect to the system and software requirements. Ensure that the algorithms and equations are appropriate for the problem solution. Validate the correctness of any constraints or limitations such as rounding, truncation, expression simplifications, best fit estimations, non-linear solutions imposed by the algorithms and equations.

Audit performance. Provide an independent assessment of whether a software process and its products conform to applicable regulations, standards, plans, procedures, specifications and guidelines. Audits may be applied to any software process or product at any development stage. Audits may be initiated by the supplier, the acquirer, the developer or other involved party such as a regulatory agency. The initiator of the audit selects the audit team and determines the degree of independence required. The initiator of the audit and the audit team leader establish the purpose, scope, plan, and reporting requirements for the audit.

The auditors collect sufficient evidence to decide whether the software processes and products meet the evaluation criteria. They identify major deviations, assess risk to quality, schedule, and cost and report their findings. Examples of processes that could be audited include configuration management practices, use of software tools, degree of integration of the various software engineering disciplines particularly in developing an architecture, security issues, training, project management.

Audit support. Provide technical expertise to the auditors on request. They may represent the acquirer at audit proceedings, and may assist in the V&V of remedial activities identified by the audit.

Control flow analysis. Assess the correctness of the software by diagramming the logical control. Examine the flow of the logic to identify missing, incomplete, or inaccurate requirements. Validate whether the flow of control amongst the functions represents a correct solution to the problem.

Cost analysis. Evaluate the cost status of the development processes. Compare budgeted costs against actual costs. Correlate cost expenditures with technical status and schedule progress. Identify program risks if actual costs indicate behind schedule and over cost estimates.

Database analysis. Evaluate database design as part of a design review process could include the following:

1) *Physical Limitations Analysis.* Identify the physical limitations of the database such as maximum number of records, maximum record length, largest numeric value, smallest numeric value, and maximum array length in a data structure and compare them to designed values.
2) *Index vs. Storage Analysis.* Analyze the use of multiple indexes compared to the volume of stored data to determine if the proposed approach meets the requirements for data retrieval performance and size constraints.
3) *Data Structures Analysis.* Some database management systems have specific data structures within a record such as arrays, tables, and date formats. Review the use of these structures for potential impact on requirements for data storage and retrieval.
4) *Backup and Disaster Recovery Analysis.* Review the methods employed for backup against the requirements for data recovery and system disaster recovery and identify deficiencies.

Data flow analysis. Evaluate data flow diagrams as part of a design review process. This could include the following:

1) *Symbology Consistency Check.* The various methods used to depict data flow diagrams employ very specific symbology to represent the actions performed. Verify that each symbol is used consistently.
2) *Flow Balancing.* Compare the output data from each process block to the data inputs and the data derived within the process to ensure the data is available when required. This process does not specifically examine timing or sequence considerations.
3) *Confirmation of Derived Data.* Examine the data derived within a process for correctness and format. Data designed to be entered into a process by operator action should be confirmed to ensure availability.
4) *Keys to Index Comparison.* Compare the data keys used to retrieve data from data stores within a process to the database index design to confirm that no invalid keys have been used and the uniqueness properties are consistent.

Disaster recovery plan assessment. Verify that the disaster recovery plan is adequate to restore critical operation of the system in the case of an extended system outage. The disaster recovery plan should include the following:

1) Identification of the disaster recovery team and a contact list.
2) Recovery operation procedures.
3) Procedure for establishing an alternative site including voice and data communications, mail, and support equipment.
4) Plans for replacement of computer equipment.
5) Establishment of a system backup schedule.
6) Procedures for storage and retrieval of software, data, documentation, and vital records off-site.
7) Logistics of moving staff, data, documentation, etc.

Distributed architecture assessment. Assess the distribution of data and processes in the proposed architecture for feasibility, timing compliance, availability of telecommunications, cost, backup and restore features, downtime, system degradation, and provisions for installation of software updates.

Feasibility study evaluation. Verify that the feasibility study is correct, accurate, and complete. Validate that all logical and physical assumptions (e.g., physical models, business rules, logical processes), constraints, and user requirements are satisfied.

Independent risk assessment. Conduct an independent risk assessment on any aspect of the software project and report on the findings. Such risk assessments will be primarily from a system perspective. Examples of risk assessment include appropriateness of the selected development methodology or tools for the project; and quality risks associated with proposed development schedule alternatives.

Inspection. Inspect the software process to detect defects in the product at each selected development stage to assure the quality of the emerging software. The inspection process may consist of multiple steps for the segregation of the inspection functions of

1) Inspection planning
2) Product overview
3) Inspection preparation
4) Examination meeting
5) Defect rework
6) Resolution follow-up

An inspection is performed by a small team of peer developers and includes, but is not led by, the author. The inspection team usually consists of three to six persons, and in some cases includes personnel from the

test group, quality assurance, or V&V. The participants assume specific roles in order to find, classify, report, and analyze defects in the product. Each type of inspection is specifically defined by its intended purpose, required entry criteria, defect classification, checklists, exit criteria, designated participants, and its preparation and examination procedures. Inspections do not debate engineering judgments, suggest corrections, or educate project members; they detect anomalies and problems and verify their resolution by the author.

Inspection (concept). Verify that the system architecture and requirements satisfy customer needs. Verify that the system requirements are complete and correct, and that omissions, defects, and ambiguities in the requirements are detected.

Inspections (design). Verify that the design can be implemented, is traceable to the requirements, and that all interface and procedural logic is complete and correct, and that omissions, defects, and ambiguities in the design are detected.

Inspections (requirements). Verify that the requirements meet customer needs, can be implemented, and are complete, traceable, testable, and consistent so that omissions, defects, and ambiguities in the requirements are detected.

Inspection (source code). Verify that the source code implementation is traceable to the design, and that all interfaces and procedural logic are complete and correct, and that omissions, defects, and ambiguities in the source code are detected.

Inspection—test case (component, integration, system, acceptance). Verify that the (component, integration, system, acceptance) test plan has been followed accurately, that the set of component test cases is complete, and that all component test cases are correct.

Inspection—test design (component, integration, system, acceptance). Verify that the (component, integration, system, acceptance) test design is consistent with the test plan, and that the test design is correct, complete, and readable.

Inspection—test plan (component, integration, system, acceptance). Verify that the scope, strategy, resources, and schedule of the (component, integration, system, acceptance) testing process have been completely and accurately specified, that all items to be tested and all required tasks to be performed have been defined, and to ensure that all personnel necessary to perform the testing have been identified.

Operational evaluation. Assess the deployment readiness and operational readiness of the software. Operational evaluation may include examining the results of operational tests, audit reviews, and anomaly reports. This evaluation verifies that the software is

1) At a suitable point of correctness for mass production of that software
2) Valid and correct for site specific configurations

Performance monitoring. Collect information on the performance of software under operational conditions. Determine whether system and software performance requirements are satisfied. Performance monitoring is a continual process and may include evaluation of the following items:

1) Database transaction rates to determine the need to reorganize or reindex the database.
2) CPU performance monitoring for load balancing.
3) Direct access storage utilization.
4) Network traffic to ensure adequate bandwidth.
5) Critical outputs of a system (e.g., scheduled frequency, expected range of values, scheduled system reports, reports of events).

Post installation validation. Execute a reference benchmark or periodic test for critical software when reliability is crucial or there is a possibility of software corruption. By automatically or manually comparing results with the established benchmark results, the system can be validated prior to each execution of the software. When pre-use benchmark testing is impractical, such as for real time, process control, and emergency-use software, a periodic test, conducted at a pre-determined interval, can be used to ensure continued reliability.

Project management oversight support. Assess project development status for technical and management issues, risks, and problems. Coordinate oversight assessment with the acquirer and development organization. Evaluate project plans, schedules, development processes, and status. Collect, analyze, and report on key project metrics.

Qualification testing. Verify that all software requirements are tested according to qualification testing requirements demonstrating the feasibility of the software for operation and maintenance. Conduct as necessary any tests to verify and validate the correctness, accuracy, and completeness of the qualification testing results. Document the qualification test results together with the expected qualification test results. Planning for qualification testing may begin during the Requirements V&V activity.

Regression analysis and testing. Determine the extent of V&V analyses and tests that must be repeated when changes are made to any previously examined software products. Assess the nature of the change to determine potential ripple or side effects and impacts on other aspects of the system. Rerun test cases based on changes, error corrections, and impact assessment, to detect errors spawned by software modifications.

Reusability assessment. Includes the use of commercial-off-the-shelf (COTS) software, modification of existing software, and the use of code modules specifically designed for reuse. Two important tasks are 1) to identify dependencies on the original hardware or software operating environment, and 2) to verify that the human interface will function correctly in the new target environment. Reuse of existing software can cost-effectively improve the quality of a software product.

Security assessment. Evaluate the security controls on the system to ensure that they protect the hardware and software components from unauthorized use, modifications, and disclosures, and to verify the accountability of the authorized users. Verify that these controls are appropriate for achieving the system's security objectives. A system security assessment should include both the physical components (e.g., computers, controllers, networks, modems, radio frequency, infrared devices) and logical components (e.g., operating systems, utilities, application programs, communication protocols, data, administrative operating policies and procedures).

Simulation analysis. Use a simulation to exercise the software or portions of the software to measure the performance of the software against predefined conditions and events. The simulation can take the form of a manual walkthrough of the software against specific program values and inputs. The simulation can also be another software program that provides the inputs and simulation of the environment to the software under examination. Simulation analysis is used to examine critical performance and response time requirements or the software's response to abnormal events and conditions.

Sizing and timing analysis. Collect and analyze data about the software functions and resource utilization to determine if system and software requirements for speed and capacity are satisfied. The types of software functions and resource utilization issues include, but are not limited to the following:

1) CPU load
2) Random access memory and secondary storage (e.g., disk, tape) utilization
3) Network speed and capacity
4) Input and output speed

Sizing and timing analysis is started at software design and iterated through acceptance testing.

System software assessment. Assess system software (e.g., operating system, computer-aided software engineering tools, data base management system, repository, telecommunications software, graphical user interface) for feasibility, impact on performance and functional requirements, maturity, supportability, adherence to standards, developer's knowledge of and experience with the system software and hardware, and software interface requirements.

Test certification. Certify the test results by verifying that the tests were conducted using baselined requirements, a configuration control process, and repeatable tests, and by witnessing the tests. Certification may be accomplished at a software configuration item level or at a system level.

Test evaluation. Evaluate the tests for requirements coverage and test completeness. Assess coverage by assessing the extent of the software exercised. Assess test completeness by determining if the set of inputs used during test are a fair representative sample from the set of all possible inputs to the software. Assess whether test inputs include boundary condition inputs, rarely encountered inputs, and invalid inputs. For some software it may be necessary to have a set of sequential or simultaneous inputs on one or several processors to test the software adequately.

Test witnessing. Monitor the fidelity of test execution to the specified test procedures, and witness the recording of test results. When a test failure occurs, the testing process can be continued by 1) implementing a "workaround" to the failure; 2) inserting a temporary code patch; or 3) halting the testing process and implementing a software repair. In all cases, assess the test continuation process for test process breakage (e.g., some software is not tested or a patch is left in place permanently), adverse impact on other, tests and loss of configuration control. Regression testing should be done for all the software affected by the test failure.

Training documentation evaluation. Evaluate the training materials and procedures for completeness, correctness, readability, and effectiveness.

User documentation evaluation. Evaluate the user documentation for its completeness, correctness, and consistency with respect to requirements for user interface and for any functionality that can be invoked by the user. The review of the user documentation for its readability and effectiveness should include representative end users who are unfamiliar with the software. Employ the user documentation in planning an acceptance test that is representative of the operational environment.

User training. Assure that the user training includes rules that are specific to the administrative, operational, and application aspects and industry standards for that system. This training should be based on the technical user documentation and procedures provided by the manufacturer of the system. The organization that is responsible for the use of the system should be responsible for providing appropriate user training.

V&V tool plan generation. Prepare a plan that describes the tools needed to support the V&V effort. The plan includes a description of each tool's performance, required inputs, outputs generated, need date, and cost of tool purchase or development. The tool plan should also describe test facilities and integration and system test laboratories supporting the V&V effort. The scope and rigor of the V&V effort as defined by the selected software integrity level should be considered in defining the performance required of each tool.

Walkthrough. Participate in the evaluation processes in which development personnel lead others through a structured examination of a product. Ensure that the participants are qualified to examine the products and are not subject to undue influence. See specific descriptions of the requirement walkthrough, design walkthrough, source code walkthrough, and test walkthrough.

Walkthrough (design). Participate in a walkthrough of the design and updates of the design to ensure completeness, correctness, technical integrity, and quality.

Walkthrough (requirements). Participate in a walkthrough of the requirements specification to ensure that the software requirements are correct, unambiguous, complete, verifiable, consistent, modifiable, traceable, testable, and usable throughout the life cycle.

Walkthrough (source code). Participate in a walkthrough of the source code to ensure that the code is complete, correct, maintainable, free from logic errors, complies with coding standards and conventions, and will operate efficiently.

Walkthrough (test). Participate in a walkthrough of the test documentation to ensure that the planned testing is correct and complete, and that the test results will be correctly analyzed.

Annex H

(informative)

Other references

The following references are considered useful to implement and interpret the V&V requirements contained in this standard. These references are not required to be in compliance with this standard.

[B1] IEC 60300-3-9 (1995), Dependability management—Part 3: Application guide—Section 9: Risk analysis of technological systems.

[B2] IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

[B3] IEEE Std 730-1989, IEEE Standard for Software Engineering Quality Assurance Plans.

[B4] IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans.

[B5] IEEE Std 829-1983 (Reaff 1991), IEEE Standard for Software Test Documentation.

[B6] IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.

[B7] IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.

[B8] IEEE Std 1028-1997, IEEE Standard for Software Reviews.

[B9] IEEE Std 1044-1993, IEEE Standard for Classification of Software Anomalies.

[B10] IEEE Std 1059-1993, IEEE Guide for Software Verification and Validation Plans.

[B11] IEEE Std 1061-1992, IEEE Standard for a Software Quality Metrics Methodology.

[B12] IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.

[B13] IEEE/EIA 12207.0-1996, IEEE/EIA Standard—Industry Implementation of ISO/IEC 12207 : 1995, Standard for Information Technology—Software life cycle processes.

[B14] IEEE/EIA 12207.1-1997, IEEE/EIA Guide—Industry Implementation of ISO/IEC 12207 : 1995, IEEE/EIA Standard for Information Technology—Software Life Cycle Processes—Life cycle data.

[B15] IEEE/EIA 12207.2-1996, IEEE/EIA Guide—Industry Implementation of ISO/IEC 12207 : 1995, IEEE/EIA Standard for Information Technology—Software Life Cycle Processes—Implementation considerations.

[B16] ISO/IEC 12207: 1995, Information technology—Software life cycle processes.

[B17] ISO/IEC DIS 15026: 1996, Information technology—System and software integrity levels.

[B18] ISO 8402: 1994, Quality Management and Quality Assurance—Vocabulary.

Annex I

(normative)

Definitions from existing standards (normative)

The following are definitions from existing standards as identified in the brackets [ ]. These definitions are placed in this annex so that the body of this standard will not require updating in the event the cited standards and their definitions change.

anomaly: Any condition that deviates from the expected based on requirements, specifications, design, documents, user documents, standards, etc., or from someone's perceptions or experiences. Anomalies may be found during, but not limited to, the review, test, analysis, compilation, or use of software products or applicable documentation. [IEEE Std 1044]

hazard: A source of potential harm or a situation with a potential for harm in terms of human injury, damage to health, property, or the environment, or some combination of these. [IEC 60300-3-9]

hazard identification: The process of recognizing that a hazard exists and defining its characteristics. [IEC 60300-3-9]

integrity level: A denotation of a range of values of a property of an item necessary to maintain system risks within acceptable limits. For items that perform mitigating functions, the property is the reliability with which the item must perform the mitigating function. For items whose failure can lead to a threat, the property is the limit on the frequency of that failure. [ISO/IEC 15026]

risk: The combination of the frequency, or probability, and the consequence of a specified hazardous event. [IEC 60300-3-9]

risk analysis: The systematic use of available information to identify hazards and to estimate the risk to individuals or populations, property or the environment. [IEC 60300-3-9]

software integrity level: The integrity level of a software item. [ISO/IEC 15026]

validation: Confirmation by examination and provisions of objective evidence that the particular requirements for a specific intended use are fulfilled.

NOTES
1—In design and development, validation concerns the process of examining a product to determine conformity with user needs.
2—Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages.
3—"Validated" is used to designate the corresponding status.
4—Multiple validations may be carried out if there are different intended uses. [ISO 8402: 1994]

verification: Confirmation by examination and provisions of objective evidence that specified requirements have been fulfilled.

NOTES
1—In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity.
2—"Verified" is used to designate the corresponding status. [ISO 8402: 1994]

To order IEEE standards...

Call 1. 800. 678. IEEE (4333) in the US and Canada.

Outside of the US and Canada:
1. 732. 981. 0600

To order by fax:
1. 732. 981. 9667

IEEE business hours: 8 a.m.–4:30 p.m. (EST)

For on-line access to IEEE standards information...

Via the World Wide Web:
http://standards.ieee.org/

Via ftp:
stdsbbs.ieee.org

ISBN 0-7381-0196-6

9 780738 101965 90000

ISBN 0-7381-0196-6