

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

IEEE Std 1012-1998

IEEE Standard for Software Verification and Validation

IEEE Computer Society

Sponsored by the
Software Engineering Standards Committee

20 July 1998

SH94625



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

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1998. Printed in the United States of America.

Print: ISBN 0-7381-0196-6, SH94625
PDF: ISBN 0-7381-0449-3, SS94625

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

This is a Redline Document produced by Techstreet, a business of Thomson Reuters.

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fastpaced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

Redline Processing Notes:

1. ~~Red Text~~ - Red strikethrough text denotes deletions.
2. Blue Text - Blue underlined text denotes modifications and additions.

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.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

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*

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

Delores R. Wallace, *Vice Chair*

Donald W. Sova, *Secretary*

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

1.	Overview	1
1.1	Purpose.....	1
1.2	Field of application	2
1.3	V&V objectives	2
1.4	Organization of the standard.....	2
1.5	Audience	3
1.6	Compliance	3
1.7	Disclaimer	4
1.8	Limitations	4
2.	Normative references	4
3.	Definitions, abbreviations, and conventions	4
3.1	Definitions.....	4
3.2	Abbreviations	6
3.3	Conventions	7
4.	V&V software integrity levels	7
4.1	Software integrity levels	7
5.	V&V processes	9
5.1	Process: Management	10
5.2	Process: Acquisition.....	10
5.3	Process: Supply	11
5.4	Process: Development.....	11
5.5	Process: Operation	15
5.6	Process: Maintenance.....	15
6.	Software V&V reporting, administrative, and documentation requirements	16
6.1	V&V reporting requirements	16
6.2	V&V administrative requirements.....	16
6.3	V&V documentation requirements	16
7.	SVVP outline	17
7.1	(SVVP Section 1) Purpose.....	18
7.2	(SVVP Section 2) Referenced documents	18
7.3	(SVVP Section 3) Definitions.....	18
7.4	(SVVP Section 4) V&V overview	18
7.5	(SVVP Section 5) V&V processes	19
7.6	(SVVP Section 6) V&V reporting requirements	21
7.7	(SVVP Section 7) V&V administrative requirements.....	23
7.8	(SVVP Section 8) V&V documentation requirements	24

Annex A (informative) Mapping of ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks	49
Annex B (informative) A software integrity level scheme	55
Annex C (informative) Definition of independent verification and validation (IV&V)	57
Annex D (informative) V&V of reusable software	60
Annex E (informative) V&V metrics	61
Annex F (informative) Example of V&V organizational relationship to other project responsibilities	63
Annex G (informative) Optional V&V task descriptions	64
Annex H (informative) Other references	70

IEEE Standard for Software Verification and Validation-Plans

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.

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

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).

1.2 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. 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.

~~1.—Scope and References~~

~~1.1—Scope~~

~~This standard has a threefold purpose:~~

- ~~1) —To provide for both critical and noncritical software, uniform and minimum requirements for the format and content of Software Verification and Validation Plans (SVVPs)~~
- ~~2) —To define, for critical software, specific minimum verification and validation (V&V) tasks and their required inputs and outputs that shall be included in SVVPs~~

- 3) To suggest optional V&V tasks to be used to tailor SVVPs as appropriate for the particular V&V effort

~~This standard requires that an SVVP be written for both critical and noncritical software. Critical software is software in which a failure could have an impact on safety or could cause large financial or social losses.~~

~~This SVVP shall include V&V tasks to:~~

- 1) ~~Verify that the products of each software life-cycle phase:~~
 - a) ~~Comply with previous life-cycle phase requirements and products (for example, for correctness, completeness, consistency, accuracy)~~
 - b) ~~Satisfy the standards, practices, and conventions of the phase~~
 - c) ~~Establish the proper basis for initiating the next life-cycle phase activities~~
- 2) ~~Validate that the completed end product complies with established software and system requirements.~~

~~For critical software, this standard requires that minimum V&V tasks and their inputs and outputs be included in all SVVPs. For noncritical software, this standard does not specify minimum required V&V tasks; however, all other requirements of this standard shall be satisfied. This standard does recommend that the minimum V&V tasks for critical software also be employed for noncritical software.~~

~~This standard defines optional V&V tasks that permit V&V planners to tailor an SVVP for a V&V effort. For critical software, the minimum tasks may be supplemented with tasks selected from the optional tasks. For noncritical software, tasks may be selected from the minimum and optional tasks. Additional tasks identified by the user of this standard may be included in the SVVP for critical and noncritical software.~~

The life cycle used in this standard serves as a model and consists of the following life cycle phases:

- 1) Concept
- 2) Requirements
- 3) Design
- 4) Implementation
- 5) Test
- 6) Installation and checkout
- 7) Operation and maintenance

Compliance with this standard does not require use of the life cycle model presented here. If a different model is used, the SVVP shall include cross references to this standard's life cycle and to the V&V tasks, inputs, and outputs specified here for each life cycle phase.

This standard requires that the following be defined for each phase:

- 1) Verification and validation tasks
- 2) Methods and criteria
- 3) Inputs and outputs
- 4) Schedule
- 5) Resources
- 6) Risks and assumptions
- 7) Roles and responsibilities

~~This standard requires a management effort that encompasses all life-cycle phases. The management section of the SVVP defines information necessary to manage and perform the V&V effort, and to coordinate V&V with other aspects of the project. The standard requires the SVVP to specify how the V&V results shall be documented in the Software Verification and Validation Report (SVVR).~~

When this standard is invoked for existing software, and the ~~SVVP shall describe how V&V will be performed when~~ required V&V inputs ~~do~~ are not exist. ~~The~~ available, then V&V tasks may use other available project input sources or may reconstruct the needed inputs to achieve compliance with this standard ~~does not prohibit the incorporation of additional content into an SVVP.~~

~~The SVVP standard derives its scope from ANSI/IEEE Std 730-1984 [2].¹ The SVVP standard may be applied in conjunction with, or independent of, other IEEE software engineering standards. This standard uses the definitions of ANSI/IEEE Std 729-1983 [1]. This SVVP standard contains V&V configuration analysis tasks that, in part or in whole, are reflected in ANSI/IEEE Std 828-1983 [3]. Test documentation is compatible with that in ANSI/IEEE Std 829-1983 [4].~~

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.

¹~~Numbers in brackets correspond to those of the references in 1.2 of this standard.~~

~~This standard shall be used in conjunction with the following publications:~~

~~[1] ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.²~~

²~~ANSI documents are available from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018.~~

~~[2] ANSI/IEEE Std 730-1984, IEEE Standard for Software Quality Assurance Plans.~~

~~[3] ANSI/IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans. [4] ANSI/IEEE Std 829-1983, IEEE Standard for Software Test Documentation.~~

~~2. Conventions, Definitions, and Acronyms~~

~~2.1 Conventions~~

~~The use of the term *documentation* rather than *document* indicates that the information may exist in several documents or may be embedded within a document addressing more than one subject.~~

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: ~~Formal~~ Testing conducted [in an operational environment](#) to determine whether ~~or not~~ a system satisfies its acceptance criteria [\(i.e., initial requirements and current needs of its user\)](#) and to enable the customer to determine whether ~~or not~~ to accept the system. ~~(See ANSI/IEEE Std 729-1983 [1].)~~

3.2 1 anomaly: [See Annex I.](#) ~~Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents. A critical anomaly is one that must be resolved before the V&V effort proceeds to the next life cycle phase~~

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

~~**concept phase:** The initial phase of a software development project, in which user needs are described and evaluated through documentation (for example, statement of needs, advance planning report, project initiation memo, feasibility studies, system definition documentation, regulations, procedures, or policies relevant to the project).~~

~~**critical software:** Software whose failure could have an impact on safety, or could cause large financial or social loss.~~

~~**design phase:** The period of time in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements. (See ANSI/IEEE Std 729-1983 [1].)~~

~~**implementation phase:** The period of time in the software life cycle during which a software product is created from design documentation and debugged. (See ANSI/IEEE Std 729-1983 [1].)~~

~~**installation and checkout phase:** The period of time in the software life cycle during which a software product is integrated into its operational environment and tested in this environment to ensure that it performs as required. (See ANSI/IEEE Std 729-1983 [1].)~~

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. ~~(See ANSI/IEEE Std 729-1983 [1].)~~ 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 ~~phase:~~ Any period of time during software **~~process:~~ A set of interrelated activities that result in the** development or ~~operation that may be characterized by a primary type of assessment of software products. Each activity (such as design or testing) that is being conducted. These phases consists of tasks. The life cycle processes~~ may overlap one another. For V&V purposes, no ~~phase process~~ is concluded until its development products are ~~fully~~ 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 ~~applicable to all projects. V&V planning for critical shall include all such tasks; these tasks are recommended for the V&V of noncritical software.~~

~~**operation and maintenance phase:** The period of time in the software life cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements. (See ANSI/IEEE Std 729-1983 [1].)~~

3.1.16 optional tasks: Those V&V tasks that ~~are applicable~~ may be added to the minimum V&V tasks ~~to some, but not all, software, or that may require the use of address specific tools or techniques. These tasks should be performed when appropriate. The list of tasks provided in Table 2 is not exhaustive~~ application requirements.

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

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

~~requirements phase: The period of time in the software life cycle during which the requirements, such as functional and performance capabilities for a software product, are defined and documented. (See ANSI/IEEE Std 729-1983 [1].)~~

[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. ~~(See ANSI/IEEE Std 730-1984 [2].)~~ [The software requirements are derived from the system specification.](#)

3.1.24 software verification and validation plan (SVVP): A plan ~~for~~ [describing](#) the conduct of software ~~verification and validation~~ [V&V](#).

3.1.25 software verification and validation report (SVVR): Documentation of V&V results and ~~appropriate software quality assurance results~~ [assessments](#).

3.1.26 system testing: The ~~process~~ [activities](#) of testing an integrated hardware and software system to verify ~~that~~ [and validate whether](#) the system meets its ~~specified requirements. (See ANSI/IEEE Std 729-1983 [1].)~~ [original objectives.](#)

3.1.27 test case: Documentation ~~specifying~~ [that specifies](#) inputs, predicted results, and a set of execution conditions for a test item. ~~(See ANSI/IEEE Std 829-1983 [4].)~~

3.1.28 test design: Documentation ~~specifying~~ [that specifies](#) the details of the test approach for a software feature or combination of software features and identifying the associated tests. ~~(See ANSI/IEEE Std 829-1983 [4].)~~

~~**test phase:** The period of time in the software life cycle in which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied. (See ANSI/IEEE Std 729-1983 [1].)~~

3.1.29 test plan: Documentation ~~specifying~~ [that specifies](#) the scope, approach, resources, and schedule of intended testing activities. ~~(See ANSI/IEEE Std 829-1983 [4].)~~

3.1.30 test procedure: Documentation ~~specifying~~ [that specifies](#) a sequence of actions for the execution of a test. ~~(See ANSI/IEEE Std 829-1983 [4].)~~

~~**validation:** The process of evaluating software at the end of the software development process to ensure compliance with software requirements. (See ANSI/IEEE Std 729-1983 [1].)~~

~~**verification:** The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase. (See ANSI/IEEE Std 729-1983 [1].)~~

2.3 Acronyms

3.1.31 validation: [See Annex I.](#)

3.1.32 verification: [See Annex I.](#)

3.2 Abbreviations

The following ~~acronyms~~ [abbreviations](#) appear in this standard:

[COTS](#) [Commercial-Off-The-Shelf](#)

<u>IDD</u>	<u>Interface Design Document</u>
<u>IEC</u>	<u>International Electrotechnical Commission</u>
<u>IRS</u>	<u>Interface Requirements Specification</u>
<u>ISO</u>	<u>International Organization for Standardization</u>
<u>IV&V</u>	<u>Independent Verification and Validation</u>
<u>RFP</u>	<u>Request for Proposal (-tender)</u>
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.

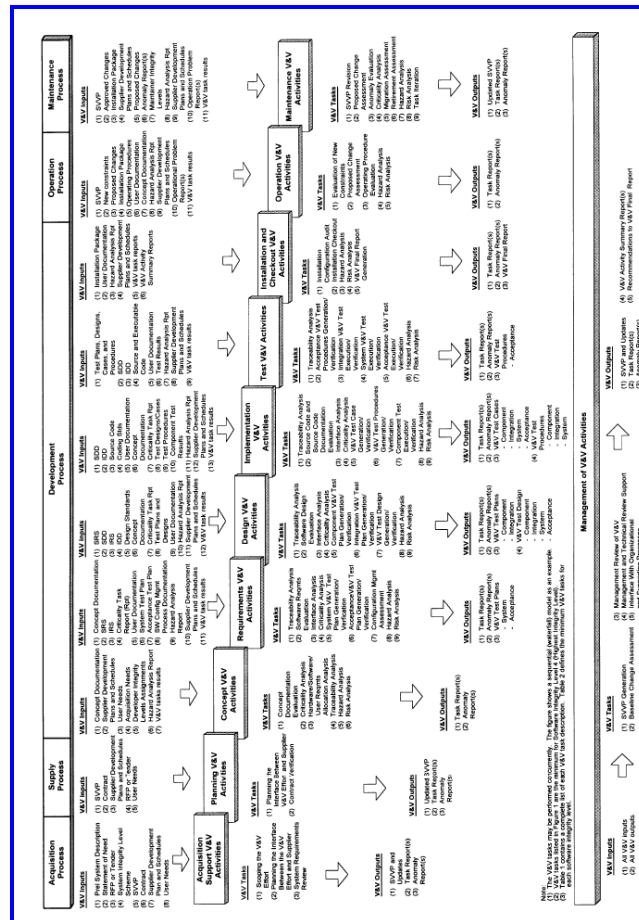
For planning the V&V processes, software integrity levels are generally assigned early in the development process, preferably during the system requirements analysis and architecture design activities. The software integrity level can be assigned to software requirements, functions, group of functions, 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

(New)

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 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.



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](#)

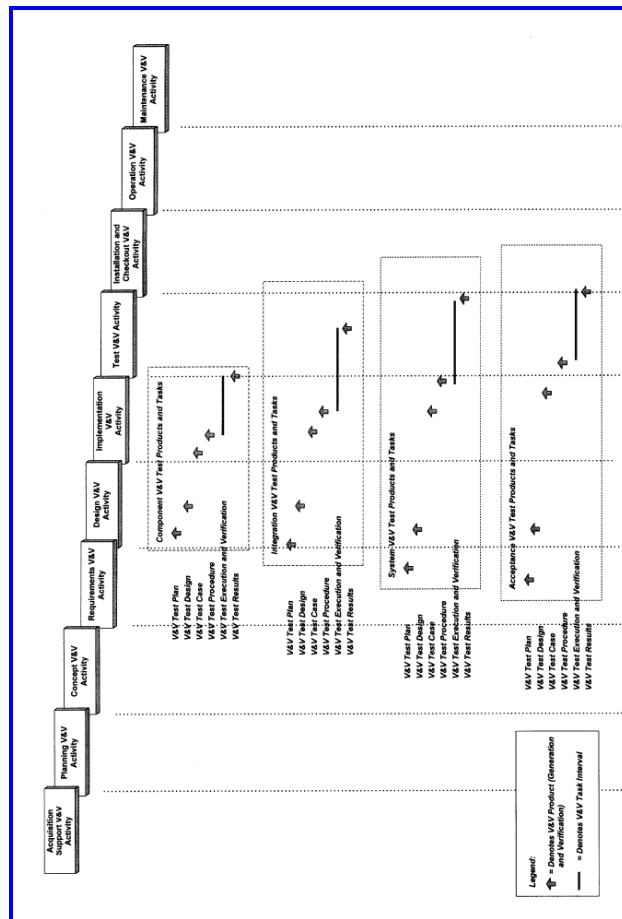
7) [Task: Risk Analysis](#)

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



(New)

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.