IEEE Standards

IEEE Std 1012™-2004
(Revision of
IEEE Std 1012-1998)

1012™

IEEE Standard for Software Verification and Validation

IEEE Computer Society

Sponsored by the
Software Engineering Standards Committee

IEEE

3 Park Avenue, New York, NY10016-5997, USA

8 June 2005

Print: SH95308
PDF: SS95308

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Recognized as an
American National Standard (ANSI)

IEEE Std 1012™-2004
(Revision of
IEEE Std 1012-1998)

IEEE Standard for Software Verification and Validation

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Approved 12 April 2005

American National Standards Institute

Approved 8 December 2004

IEEE-SA Standards Board

Abstract: Software verification and validation (V&V) processes determine whether the development products of a given activity conform to the requirements of that activity and whether the software satisfies its intended use and user needs. Software V&V life cycle process requirements are specified for different software integrity levels. The scope of V&V processes encompasses software-based systems, computer software, hardware, and interfaces. This standard applies to software being developed, maintained, or reused [legacy, commercial off-the-shelf (COTS), non-developmental items]. The term software also includes firmware, microcode, and documentation. Software V&V processes include analysis, evaluation, review, inspection, assessment, and testing of software products.
Keywords: IV&V, software integrity level, software life cycle, V&V, validation, verification

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2005 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 8 June 2005. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

Print: ISBN 0-7381-4641-2 SH95308
PDF: ISBN 0-7381-4642-0 SS95308

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.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

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.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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 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. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854

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; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Introduction

This introduction is not part of IEEE Std 1012-2004, 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. 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. The software V&V processes determine whether the development products of a given activity conform to the requirements of that activity and whether the software satisfies its intended use and user needs. The determination includes assessment, analysis, evaluation, review, inspection, and testing of software products and processes. Software V&V is performed in parallel with software development, not at the conclusion of the development effort.

Software V&V is an extension of program management and systems engineering that employs a rigorous methodology to identify objective data and conclusions to provide feedback about software quality, performance, and schedule to the development organization. This feedback consists of anomaly resolutions, performance improvements, and quality improvements not only for expected operating conditions, but also across the full spectrum of the system and its interfaces. Early feedback results allow the development organization to modify the software products in a timely fashion and thereby reduce overall project and schedule impacts. Without a proactive approach, anomalies and associated software system changes are typically delayed to later in the program schedule, resulting in greater program costs and schedule delays.

IEEE Std 1012-2004 is a process standard that defines the V&V processes in terms of specific activities and related tasks. The standard also defines the contents of the *software v&v plan* (SVVP), including an example format.

This version of the standard contains minor changes to IEEE Std 1012-1998. Following is a summary:

a) Revised Clause 1 to conform to IEEE style and
   1) Moved the description of the verification process and validation process from 1.3 to 1.1.
   2) Expanded 1.2 to discuss the importance of performing the software V&V from a systems perspective—software and its interaction with the system of which it is a part.
b) Moved Figure 3 into the definition of V&V effort (see 3.1.37) with no figure reference.
c) Clarified Clause 4 concept of software integrity and selection of software integrity levels.
d) Revised Clause 6 to contain all of the normative documentation requirements (see 6.1) that were in Clause 7.
e) Revised Clause 7 to consolidate IEEE 1012A™-1998 [B6] into the revision of this standard.
f) Revised Table 1 as follows:
   1) Added "security analysis" to the required V&V tasks.
   2) Reformatted test tasks to uniquely identify requirements for each test type—no normative changes were made to the test tasks.
   3) Added a subtask to the "scoping of V&V" in the Acquisition support V&V activity to determine the extent of V&V on reused software.
   4) Corrected previous editorial errors.
g) Added mapping of IEEE 1012 tasks to CMMI Engineering Process Groups in Annex A.
h) Added a definition of integrated independent V&V (IV&V) to Annex C.
i) Clarified treatment of reuse software in Annex D.
j) Added sample measures to Annex E.
k) Removed Annex I and moved the definitions into 3.1.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

The following key concepts are emphasized in this standard:

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

— *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 for tailoring 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 viewpoints.* Includes minimum V&V tasks to address system issues. These tasks include hazard analysis, security analysis, risk analysis, migration assessment, and retirement assessment. Specific system issues are contained in individual V&V task criteria.

— *Conformance to international and IEEE standards.* Defines the V&V processes to conform to life cycle process standards such as ISO/IEC Std 12207:1995 [B13], IEEE Std 1074™-1997 [B10], and IEEE/EIA Std 12207.0™-1996 [B12], 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. This standard is compatible with all life cycle models; however, not all life cycle models use all of the life cycle processes described in this standard.

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

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 or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Participants

At the time this standard was completed, the Software Verification and Validation Working Group had the following membership:

Roger U. Fujii, *Chair*
Dolores R. Wallace, *Vice Chair*
David H. Daniel, *Secretary*

John W. Bradbury	Eva Freund	Steven M. Raque
Paul R. Croll	Ron K. Greenthaler	Subrato Sensharma
H. Taz Daughtrey	Lisa A. Jensen	Nancy E. Sunderland
Harpal S. Dhama	Rex Kidd	Richard J. Stevenson
Michael Edwards	Norm Leblanc	Gina B. To
Uma D. Ferrell	Kevin B. Morgan	Michael E. Waterman

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Satish K. Aggarwal	William Eventoff	Thomas M. Kurihara
Michael Baldwin	John Fendrich	Susan Land
Bakul Banerjee	Yaacov Fenster	Carol Long
Mario Barbacci	Uma D. Ferrell	Yuhai Ma
Edward Bartlett	Ronald Fluegge	G Michel
Juris Borzovs	Rabiz Foda	James Moore
Wesley Bowers	Eva Freund	Dennis Nickle
John W. Bradbury	Samuel Fryer	Craig Noah
Daniel Brosnan	Roger Fujii	Roger Parker
Nissen Burstein	Juan Garbajosa Sopeña	Charles Roslund
Joseph Butchko	Jean-Denis Gorin	James Ruggieri
Garry Chapman	Lewis Gray	Helmut Sandmayr
Keith Chow	Ron K. Greenthaler	Robert J. Schaaf
Antonio M. Cicu	Britton Grim	Hans Schaefer
Todd Cooper	Michael Grimley	David Schultz
Paul R. Croll	Randall Groves	James Sivak
Surin Dureja	Jon Hagar	Mike Smith
David Daniel	Peter Hung	Luca Spotorno
H. Taz Daughtrey	Mark Heinrich	Richard J. Stevenson
Harpal S. Dhama	John Horch	Graeme Stewart
Dr. Guru Dutt Dhingra	David Horvath	Booker Thomas
Scott Duncan	Peeya Iwagoshi	Gina B. To
Dr. Sourav Dutta	Joseph Jancauskas	T.H. Tse
Clint Early, Jr.	William Junk	John Waclo
Christof Ebert	Piotr Karocki	Richard Walker
Michael Edwards	Dwayne Knirk	Michael E. Waterman
Amir El-Sheikh	Subrahmanyam Kommu	Oren Yuen
Gary Engmann	Robert Konnik	Janusz Zalewski
Caroline Evans		Li Zhang

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

When the IEEE-SA Standards Board approved this standard on 8 December 2004, it had the following membership:

Don Wright, *Chair*
Steve M. Mills, *Vice Chair*
Judith Gorman, *Secretary*

Chuck Adams	Raymond Hapeman	Daleep C. Mohla
Stephen Berger	Richard J. Holleman	Paul Nikolich
Mark D. Bowman	Richard H. Hulett	T. W. Olsen
Joseph A. Bruder	Lowell G. Johnson	Ronald C. Petersen
Bob Davis	Joseph L. Koepfinger*	Gary S. Robinson
Roberto de Marca Boisson	Hermann Koch	Frank Stone
Julian Forster*	Thomas J. McGean	Malcolm V. Thaden
Arnold M. Greenspan		Doug Topping
Mark S. Halpin		Joe D. Watson

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Alan Cookson, *NIST Representative*

Michael D. Fisher
IEEE Standards Project Editor

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Contents

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

IEEE Standard for Software Verification and Validation

1. Overview

This verification and validation (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; however, not all life cycle models use all of the life cycle processes listed in this standard.

Software V&V processes determine whether the 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.

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 software life cycle processes may be found in ISO/IEC 12207:1995 [B13],[1] IEEE Std 1074™-1997 [B10], and IEEE/EIA Std 12207.0™-1996 [B12]. Annex A maps ISO/IEC 12207:1995 [B13] (Table A.1.1) and IEEE Std 1074-1997 [B10] (Table A.2.1) to the V&V activities and tasks defined in this standard.

1.1 Scope

This standard applies to software being acquired, developed, maintained, or reused [legacy, modified, commercial off-the-shelf (COTS), non-developmental items (NDI)]. The term software also includes firmware, microcode, and documentation.

Software V&V processes consist of the verification process and validation process. The verification process provides objective evidence whether the software and its associated products and processes

a) Conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance)

b) Satisfy standards, practices, and conventions during life cycle processes

c) Successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly)

[1]The numbers in brackets correspond to those of the bibliography in Annex H.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

The validation process provides evidence whether the software and its associated products and processes

1) Satisfy system requirements allocated to software at the end of each life cycle activity
2) Solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions)
3) Satisfy intended use and user needs

The verification process and the validation process are interrelated and complementary processes that use each other's process results to establish better completion criteria and analysis, evaluation, review, inspection, assessment, and test V&V tasks for each software life cycle activity. The V&V task criteria described in Table 1 uniquely define the conformance requirements for V&V processes.

The development of a reasonable body of evidence requires a trade-off between the amount of time spent and a finite set of system conditions and assumptions against which to perform the V&V tasks. Each project should define criteria for a reasonable body of evidence (i.e., selecting a software integrity level establishes one of the basic parameters), time schedule, and scope of the V&V analysis and test tasks (i.e., range of system conditions and assumptions).

This standard does not assign the responsibility for performing the V&V tasks to any specific organization. The analysis, evaluation, and test activities may be performed by multiple organizations; however, the methods and purpose will differ for each organization's functional objectives.

ISO/IEC 12207:1995 [B13] or IEEE/EIA 12207.0-1996 [B12] require that the developer perform various testing and evaluation tasks as an integral part of the development process. Even though the tests and evaluations are not part of the V&V processes, the techniques described in this standard may be useful in performing them. Therefore, whenever this standard mentions the developer's performance of a verification or validation activity, it is to be understood that the reference applies to the integral test and evaluation tasks of the development process.

1.2 Purpose

The purpose of this standard is to

— 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
— Define the V&V tasks, required inputs, and required outputs
— Identify the minimum V&V tasks corresponding to a four-level software integrity scheme
— Define the content of a *software V&V plan* (SVVP)

1.3 Field of application

This standard applies to all applications of software. When conducting the software V&V process, it is important to examine the software in its interactions with the system of which it is a part. This standard identifies the important system considerations that software V&V processes and tasks address in determining software correctness and other software V&V attributes (e.g., completeness, accuracy, consistency, testability).

The dynamics of software and the multitude of different logic paths available within software in response to varying system stimuli and conditions demand that the software V&V effort examine the correctness of the code for each possible variation in system conditions. The ability to model complex real world conditions will be limited, and thus the software V&V effort must examine whether the limits of the modeling are realistic and reasonable for the desired solution. The unlimited combination of system conditions presents

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

the software V&V effort with the unique challenge of using a finite set of analytical, test, simulation, and demonstration techniques to establish a reasonable body of evidence that the software is correct.

A software system provides a capability to satisfy a stated need or objective by combining one or more of the following: processes, hardware, software, facilities, and people. This relationship between the software and the system requires that software V&V processes consider software interactions with all system components. Since software links together all key components of a digital system, the software V&V process examines the interactions with each of the key system components to determine the extent to which each component influences the software and is conversely influenced by the software. The V&V process addresses the following interactions with software:

— *Environment:* Determines that the solution represented in the software correctly accounts for all conditions, natural phenomena, physical laws of nature, business rules, and physical properties and the full ranges of the system operating environment.
— *Operators/users:* Determines that the software communicates the proper status/condition of the software system to the operator/user and correctly processes all operator/user inputs to produce the required results. For incorrect operator/user inputs, ensure that the software protects the system from entering into a dangerous or uncontrolled state. Validate that operator/user policies and procedures (e.g., security, interface protocols, data representations, system assumptions) are consistently applied and used across each component interface.
— *Hardware:* Determines that the software correctly interacts with each hardware interface and provides a controlled system response (i.e., graceful degradation) for hardware faults.
— *Other software:* Determines that the software interfaces correctly with other software components in the system in accordance with requirements and that errors are not propagated between software components of the system.

Since software directly affects system behavior and performance, the scope of V&V processes is the software system, including the operational environment, operators and users, hardware, and interfacing software. 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 12207:1995 [B13], IEEE Std 1074-1997 [B10], or IEEE/EIA Std 12207.0-1996 [B12].

To address the systems perspective, software V&V should provide an integrated analysis where the V&V tasks are interrelated, providing input and insight to other V&V tasks. Results from completed life cycle processes provide valuable and necessary inputs to V&V tasks in later life cycle processes. Results and findings from one V&V task may cause previously completed V&V tasks to be analyzed again with the new data. This relationship among V&V tasks (including feedback to the development process) employing rigorous systems engineering techniques is a key approach to an integrated systems and software V&V. The software V&V results provide the development process with early detection of anomalies and potential process trends that may be used for development process improvement. The software V&V process and tasks described in this standard are consistent with systems engineering and process improvement models, such as the Capability Maturity Model Integrated (CMMI).

1.4 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. The software V&V processes determine whether the development products of a given activity conform to the requirements of that activity and whether the software satisfies its intended use and user needs. The determination includes assessment, analysis, evaluation, review, inspection, and testing of software products and processes. Software V&V should be performed in parallel with software development, not at the conclusion of the development effort.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

The results of V&V create the following benefits to the program:

- Facilitate early detection and correction of software anomalies
- Enhance management insight into process and product risk
- Support the life cycle processes to ensure conformance to program performance, schedule, and budget
- Provide an early assessment of software and system performance
- Provide objective evidence of software and system conformance to support a formal certification process
- Improve the software development and maintenance processes
- Support the process improvement for an integrated systems analysis model

1.5 Organization of the standard

This standard is organized into clauses (Clauses 1 through 7), tables (Table 1, Table 2, and Table 3), figures (Figure 1 and Figure 2), and annexes (Annexes A through H). Clauses 2 through 7 and Table 1 and Table 2 provide the mandatory V&V requirements for this standard. Table 1 and Table 2 are the focal point of this standard, containing detailed V&V process, activity, and task requirements. Clause 2 is reserved for normative references; however, this standard does not prescribe any normative references. Clause 3 provides a definition of terms, abbreviations, and conventions. Clause 4 describes the use of software integrity levels to determine the scope and rigor of V&V processes. Clause 5 describes primary software life cycle processes and lists the V&V activities associated with the life cycle process. Clause 6 describes V&V reporting, administrative, and documentation requirements. Clause 7 describes the content of a software V&V plan. Clause 1, Figure 1, Figure 2, and Table 3 contain informative material that provides examples of V&V processes and provide guidance for using this standard. All annexes are informative.

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 software system life cycle. These optional V&V tasks may be added to the minimum V&V tasks, as necessary, to tailor the V&V effort to project needs.

Figure 1 provides an example of an overview of the V&V inputs, outputs, and minimum V&V tasks for software 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 Figure 1 and Figure 2 to illustrate a mapping of the ISO/IEC 12207:1995 [B13] life cycle processes to the V&V activities and tasks described in this standard.

Annex A describes the mapping of ISO/IEC 12207:1995 [B13] and IEEE Std 1074-1997 [B10] to this standard's V&V activities and tasks. Annex B provides an example of a risk-based, four-level integrity scheme. Annex C provides a definition of *independent verification and validation* (IV&V). Annex D provides guidelines for conducting V&V of reuse software. Annex E describes V&V measures. Annex F illustrates an example of the V&V organizational relationship to other project responsibilities. Annex G describes optional V&V tasks. Annex H provides a bibliography of informative standards referenced in this standard.

1.6 Audience

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

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

1.7 Conformance

The word *shall* identifies mandatory requirements strictly to be followed in order to conform to this standard. The words *should* and *may* indicate optional tasks that are not required to claim conformance to this standard.

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 conformance to this standard is achieved if the minimum V&V tasks are implemented for the associated life cycle processes selected for the project. Any claim of conformance to this standard shall identify the applicable life cycle processes. 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 conformance to this standard.

1.8 Disclaimer

This standard establishes minimum criteria for V&V processes, activities, and tasks. However, implementing these criteria does not automatically ensure conformance 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). Conformance to this standard does not absolve any party from any social, moral, financial, or legal obligations.

1.9 Limitations

None.

2. References

This standard does not require the use of any normative references. Standards useful for the implementation and interpretation of this standard are listed in Annex H.

3. Definitions, abbreviations, and acronyms

3.1 Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms,* Seventh Edition [B2] and IEEE Std 610.12™-1990 [B3] should be referenced for terms not defined in this clause.

3.1.1 acceptance testing: (A) Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. **(B)** Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

NOTE—See IEEE Std 610.12-1990 [B3].[2]

3.1.2 anomaly: Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.3 asset: An item (e.g., design, specifications, source code, documentation, test suites, manual procedures) that has been designed for use in multiple contexts.

NOTE—See IEEE Std 1517™-1999 [B11].

3.1.4 component: One of the parts that make up a system. A component may be hardware or software and may be subdivided into other components.

NOTE 1—The terms "module," "component," and "unit" are often used interchangeably or defined to be subelements of one another in different ways depending upon the context. The relationship of these terms is not yet standardized.

NOTE 2—See IEEE Std 610.12-1990 [B3].

3.1.5 component testing: Testing of individual hardware or software components or groups of related components.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.6 criticality: The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.7 domain: A problem space.

NOTE—See IEEE Std 1517-1999 [B11].

3.1.8 domain analysis: (A) The analysis of systems within a domain to discover commonalities and differences among them. **(B)** The process by which information used in developing software systems is identified, captured, and organized so that it can be reused to create new systems, within a domain. **(C)** The result of the process in (A) and (B).

NOTE—See IEEE Std 1517-1999 [B11].

3.1.9 domain engineering: A reuse-based approach to defining the scope (i.e., domain definition), specifying the structure (i.e., domain architecture), and building the assets (e.g., requirements, designs, software code, documentation) for a class of systems, subsystems, or applications. Domain engineering may include the following activities: domain definition, domain analysis, developing the domain architecture, and domain implementation.

NOTE—See IEEE Std 1517-1999 [B11].

3.1.10 firmware: The combination of a hardware device and computer instructions and data that reside as read-only software on that device.

NOTE 1—This term is sometimes used to refer only to the hardware device or only to the computer instructions or data, but these meanings are deprecated.

NOTE 2—The confusion surrounding this term has led some to suggest that it be avoided altogether.

NOTE 3—See IEEE Std 610.12-1990 [B3].

[2]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

3.1.11 hazard: (A) An intrinsic property or condition that has the potential to cause harm or damage. **(B)** 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.

NOTE—For (A), see IEEE/EIA Std 12207.0™-1996 [B12].

3.1.12 hazard identification: The process of recognizing that a hazard exists and defining its characteristics.

3.1.13 independent verification and validation (IV&V): V&V performed by an organization that is technically, managerially, and financially independent of the development organization.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.14 integration testing: Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.15 integrity level: A value representing project-unique characteristics (e.g., software complexity, criticality, risk, safety level, security level, desired performance, reliability) that define the importance of the software to the user.

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

NOTE—See IEEE Std 610.12-1990 [B3].

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

NOTE—See *The Authoritative Dictionary* [B2].

3.1.18 life cycle processes: 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.

NOTE—See *The Authoritative Dictionary* [B2].

3.1.19 microcode: A collection of microinstructions, comprising part of or all of microprograms.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.20 microprogram: A sequence of instructions, called microinstructions, specifying the basic operations needed to carry out a machine language instruction.

NOTE—See IEEE Std 610.12-1990 [B3].

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

NOTE—See *The Authoritative Dictionary* [B2].

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

NOTE—See *The Authoritative Dictionary* [B2].

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

NOTE—See *The Authoritative Dictionary* [B2].

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

NOTE—See *The Authoritative Dictionary* [B2].

3.1.25 reusable software product: A software product developed for one use but having other uses, or one developed specifically to be usable on multiple projects or in multiple roles on one project. Examples include, but are not limited to, COTS software products, acquirer-furnished software products, software products in reuse libraries, and preexisting developer software products. Each use may include all or part of the software product and may involve its modification. This term can be applied to any software product (for example, requirements, architectures), not just to software itself.

NOTE—See *The Authoritative Dictionary* [B2].

3.1.26 risk: (A) The combination of the probability of occurrence and the consequences of a given future undesirable event. Risk can be associated with products and/or projects. **(B)** The combination of the probability of an abnormal event or failure and the consequence(s) of that event or failure to a system's components, operators, users, or environment.

NOTE—See *The Authoritative Dictionary* [B2].

3.1.27 security: (A) The protection of computer hardware or software from accidental or malicious access, use, modification, destruction, or disclosure. Security also pertains to personnel, data, communications, and the physical protection of computer installations. **(B)** The protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.

NOTE—For (A), see *The Authoritative Dictionary* [B2]. For subdefinition (B), see ISO/IEC 12207:1995 [B13].

3.1.28 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.

NOTE—See *The Authoritative Dictionary* [B2].

3.1.29 software requirements specification (SRS): Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.30 system testing: Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.31 test case: (A) A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. **(B)** Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.

NOTE—See IEEE Std 610.12-1990 [B3].

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

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.33 test plan: (A) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. **(B)** A document that describes the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, responsibilities, schedules, and required resources for the testing activity.

NOTE—See IEEE Std 610.12-1990 [B3].

3.1.34 test procedure: (A) Detailed instructions for the setup, execution, and evaluation of results for a given test case. **(B)** A document containing a set of associated instructions as in (A). **(C)** Documentation that specifies a sequence of actions for the execution of a test.

NOTE—See IEEE 982.1™-1998 [B5].

3.1.35 validation: (A) The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. **(B)** The process of providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions), and satisfy intended use and user needs.

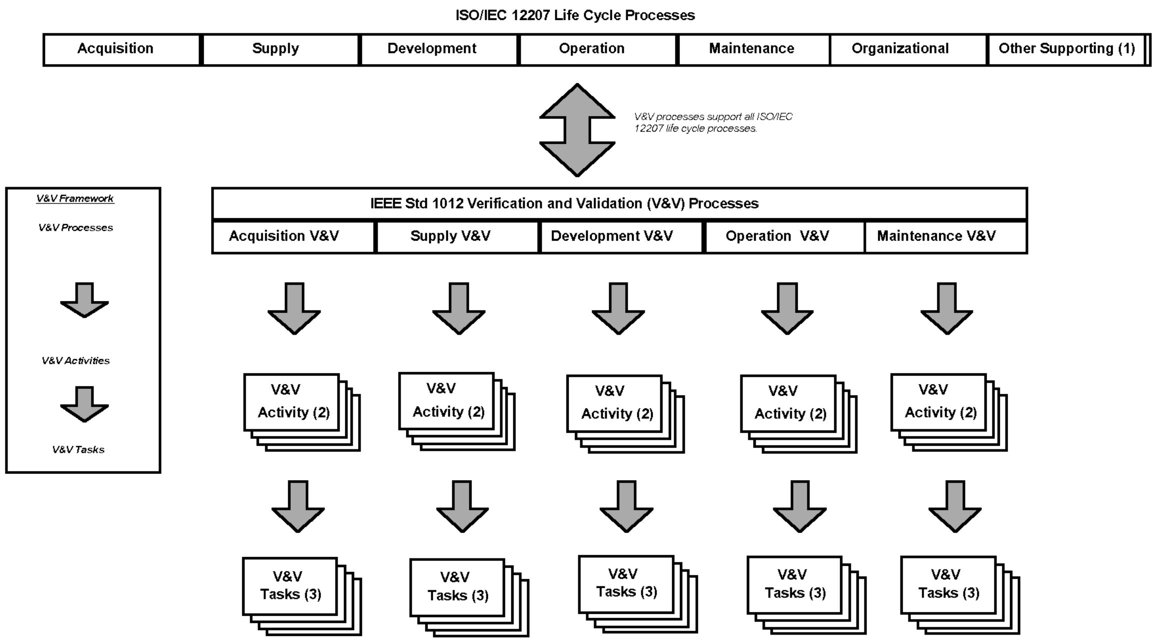
NOTE—For (A), see IEEE Std 610.12-1990 [B3].

3.1.36 verification: (A) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. **(B)** The process of providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly).

NOTE—For subdefinition (A), see IEEE Std 610.12-1990 [B3].

3.1.37 verification and validation (V&V) effort: The work associated with performing the V&V processes, activities, and tasks. The following framework illustrates how V&V processes are subdivided into activities, which in turn have associated tasks:

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.



NOTE 1—Other supporting processes consist of documentation, configuration management, quality assurance, joint review, audit, and problem resolution.

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

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

3.2 Abbreviations and acronyms

The following acronyms and abbreviations appear in this standard:

ANSI	American National Standards Institute
COTS	commercial off-the-shelf
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IDD	interface design document
IRS	interface requirements specification
ISO	International Organization for Standardization
IV&V	independent verification and validation
NDI	non-developmental item
RFP	request for proposal (tender)
SDD	software design description

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

SRS	software requirements specification
SVVP	software V&V plan
SVVR	software V&V report
V&V	verification and validation

4. Software integrity levels

Software integrity levels are a range of values that represent software complexity, criticality, risk, safety level, security level, desired performance, reliability, or other project-unique characteristics that define the importance of the software to the user and acquirer. The characteristics used to determine software integrity level vary depending on the intended application and use of the system. The software is a part of the system, and its integrity level is to be determined as a part of that system. The assigned software integrity levels may change as the software evolves. Design, coding, procedural, and technology features implemented in the system or software can raise or lower the assigned software integrity levels. The software integrity levels established for a project should result from agreements among the acquirer, supplier, developer, and independent assurance authorities (e.g., a regulatory body or responsible agency).

This standard uses software integrity levels to determine the V&V tasks to be performed. High-integrity software requires a larger set of V&V processes and a more rigorous application of V&V tasks. Integrity levels are assigned to software requirements, functions, groups of functions, or software components or subsystems. Some software elements and components may not require the assignment of an integrity level (i.e., not applicable) because their failure would impart no consequences on the intended system operations. The V&V processes should be tailored to specific system requirements and applications through the selection of a software integrity level with its corresponding minimum V&V tasks and 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. The V&V effort may recommend technical and procedural mitigation approaches to reduce the integrity level.

As an example, this standard uses the following four-level software integrity scheme. This example scheme is based upon the concepts of consequences and mitigation potential.

Description	**Level**
Software element must execute correctly or grave consequences (loss of life, loss of system, economic or social loss) will occur. No mitigation is possible.	4
Software element must execute correctly or the intended use (mission) of the system/ software will not be realized, causing serious consequences (permanent injury, major system degradation, economic or social impact). Partial to complete mitigation is possible.	3
Software element must execute correctly or an intended function will not be realized, causing minor consequences. Complete mitigation possible.	2
Software element must execute correctly or intended function will not be realized, causing negligible consequences. Mitigation not required.	1

Another example, a scheme based on risk, is described in Annex B. This standard does not require the use of any particular software integrity level scheme described or referenced in this standard.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Integrity levels shall be assigned to software elements or components as part of the criticality analysis task. The V&V effort shall specify a software integrity level scheme if one is not already defined. The integrity level assigned to reused software products shall be in accordance with the integrity level scheme adopted for the project (see Annex D), and the reused software product shall be evaluated for use in the context of its application. Tools that insert or translate code (e.g., optimizing compilers, auto-code generators) shall be assigned the same integrity level as the integrity level assigned to the software element that the tool affects. The software system shall be assigned the same integrity level as the highest level assigned to any individual element. The software integrity level assignment shall be continually reviewed and updated by conducting the V&V criticality analysis task throughout the software development process.

Table 2 identifies minimum V&V tasks that shall be performed for each software integrity level. 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 level 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. The basis for assigning software integrity levels to software components shall be documented in a *V&V task report* and *V&V final report.*

5. Software V&V processes

V&V processes support the six primary processes of ISO/IEC 12207:1995 [B13]: the management process (see 5.1), acquisition process (see 5.2), supply process (see 5.3), development process (see 5.4), operation process (see 5.5), and maintenance process (see 5.6). The minimum V&V activities and tasks supporting these processes are referenced in the following clauses and are defined in Table 1. The subclause titles in this clause are the same as the column headings in Table 1 to correlate the requirements of the following subclauses with Table 1 tasks. Not all software projects include each of the life cycle processes listed. To conform to this standard, the V&V processes shall address all those life cycle processes used by the software project.

The V&V effort shall conform to the task descriptions, inputs, and outputs as described in Table 1. The V&V effort shall perform the minimum V&V tasks 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 this standard's software integrity level scheme and associated minimum V&V tasks of Table 2 shall be mapped to their selected software integrity level scheme.

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 degree of rigor and intensity in performing and documenting the task shall 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.

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 may be 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 analysis efforts will be coordinated with other organizations shall be documented in the organizational responsibility section of the SVVP.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

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 the 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 comprises the following generic activities and tasks:

- Preparing the plans for execution of the process
- Initiating the implementation of the plan
- Monitoring the execution of the plan
- Analyzing problems discovered during the execution of the plan
- Reporting progress of the processes
- Ensuring products satisfy requirements
- Assessing evaluation results
- Determining whether a task is complete
- Checking the results for completeness
- Checking processes for efficiency and effectiveness
- Reviewing project quality
- Reviewing project risks
- Reviewing project measures

5.1.1 Activity: Management of the V&V effort

The V&V management activity monitors and evaluates all V&V outputs. Management of the V&V effort is performed for all software life cycle processes and activities. This activity involves the following:

- A continual review of the V&V effort
- Revision of the SVVP as necessary based upon updated project schedules and development status
- Coordination of the V&V results with the developer and other supporting processes, such as quality assurance and configuration management
- Performance of reviews and audits
- Identification of process improvement opportunities in the conduct of V&V

V&V management assesses each proposed change to the system and software, identifies the software requirements that are affected by the change, and plans V&V tasks to address the change. For each proposed change, management assesses whether any new hazards or risks are introduced in the software or system development process, and identifies the impact of the change on the assigned software integrity levels. V&V task planning is revised by adding new V&V tasks or changing the scope or intensity of existing V&V tasks if software integrity levels, hazards, or risks are changed. A baseline change results from changes allocated to software releases in an incremental software development process (e.g., planned baseline versions).

Through the use of V&V measures and other qualitative and quantitative measures, this V&V activity develops program trend data and possible risk issues, which are then provided to the developer and acquirer to effect timely notification and resolution. At key program milestones (e.g., requirements review, design review, test readiness), V&V management consolidates the V&V results to establish supporting evidence of

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

whether to proceed to the next set of software development activities. Whenever necessary, the V&V management determines whether a V&V task should be reperformed as a result of changes in the software program.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Management V&V tasks described in Table 1:

1) Task: SVVP generation
2) Task: Proposed/baseline change assessment
3) Task: Management review of the V&V effort
4) Task: Management and technical review support
5) Task: Interface with organizational and supporting processes
6) Task: Identify process improvement opportunities in the conduct of V&V

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 possible preparation and issuance of a request for proposal (RFP) (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 acquisition process is used to scope the V&V effort, plan interfaces with the supplier and acquirer, review the draft systems requirements to be included in the RFP, and provide the V&V task results to support acquirer acceptance of the system. Acquirer acceptance of the system culminates after acceptance testing and installation. The V&V acquisition acceptance support activities occur throughout the software life cycle, in conjunction with other interrelated development and V&V tasks, inputs, and outputs.

5.2.1 Activity: Acquisition support V&V

The Acquisition support V&V activity addresses project initiation, RFP, contract preparation, supplier monitoring, and acceptance and completion.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Acquisition support V&V tasks described in Table 1:

1) Task: Scoping the V&V effort
2) Task: Planning the interface between the V&V effort and supplier
3) Task: System requirements review
4) Task: Acceptance support

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 negotiating, finalizing, 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 Supply V&V effort uses the supply process products to confirm that the request for proposal requirements and contract requirements are consistent and satisfy user needs before the contract is finalized. The V&V planning activity uses the contract requirements, including the program schedules, to revise and update the interface planning between the supplier and acquirer.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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 specified in Table 2 for the selected software integrity level, the following Planning V&V tasks described in Table 1:

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 support to acceptance of 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 activity represents the delineation of a specific implementation solution to solve the user's problem. During the concept 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 objective of Concept V&V is 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 specified in Table 2 for the selected software integrity level, the following Concept V&V tasks described in Table 1:

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: Security analysis
7) Task: Risk analysis

5.4.2 Activity: Requirements V&V

The Requirements V&V activity addresses software requirements analysis of the functional and performance requirements, interfaces external to the software, and requirements for qualification, safety and security, human factors engineering, data definitions, user documentation for the software, installation and acceptance, user operation and execution, and user maintenance. V&V test planning begins during the Requirements V&V activity and spans several V&V activities.

The objective of Requirements V&V is to ensure the correctness, completeness, accuracy, testability, and consistency of the system software requirements.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Requirements V&V tasks described in Table 1:

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
6) Task: Acceptance V&V test plan generation
7) Task: Configuration management assessment
8) Task: Hazard analysis
9) Task: Security analysis
10) Task: Risk analysis

5.4.3 Activity: Design V&V

In software design, software requirements are transformed into an architecture and a detailed design for each software component. The design includes databases and system interfaces (e.g., hardware, operator/user, software components, and subsystems). The Design V&V activity addresses software architectural design and software detailed design. V&V test planning continues during the Design V&V activity.

The objective of Design V&V is 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 specified in Table 2 for the selected software integrity level, the following Design V&V tasks described in Table 1:

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
6) Task: Integration V&V test plan generation
7) Task: Component V&V test design generation
8) Task: Integration V&V test design generation
9) Task: System V&V test design generation
10) Task: Acceptance V&V test design generation
11) Task: Hazard analysis
12) Task: Security analysis
13) Task: Risk analysis

5.4.4 Activity: Implementation V&V

In software implementation, the system design is transformed into code, database structures, and related machine executable representations. The Implementation V&V activity addresses software coding and testing, including the incorporation of reused software products. The objective of Implementation V&V is to verify and validate that these transformations are correct, accurate, and complete.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Implementation V&V tasks described in Table 1:

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

1) Task: Traceability analysis
2) Task: Source code and source code documentation evaluation
3) Task: Interface analysis
4) Task: Criticality analysis
5) Task: Component V&V test case generation
6) Task: Integration V&V test case generation
7) Task: System V&V test case generation
8) Task: Acceptance V&V test case generation
9) Task: Component V&V test procedure generation
10) Task: Integration V&V test procedure generation
11) Task: System V&V test procedure generation
12) Task: Component V&V test execution
13) Task: Hazard analysis
14) Task: Security analysis
15) Task: Risk analysis

5.4.5 Activity: Test V&V

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

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Test V&V tasks described in Table 1:

1) Task: Traceability analysis
2) Task: Acceptance V&V test procedure generation
3) Task: Integration V&V test execution
4) Task: System V&V test execution
5) Task: Acceptance V&V test execution
6) Task: Hazard analysis
7) Task: Security analysis
8) Task: Risk analysis

5.4.6 Activity: Installation and checkout V&V

In installation and checkout, the software product is installed and tested in the target environment. The Installation and checkout V&V activity supports the software system installation activities. The objective of Installation and checkout V&V is to verify and validate the correctness of the software installation in the target environment.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Installation and checkout V&V tasks described in Table 1:

1 Task: Installation configuration audit
2) Task: Installation checkout
3) Task: Hazard analysis

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

4) Task: Security analysis
5) Task: Risk analysis
6) Task: V&V final report generation

5.5 Process: Operation

The operation process involves the use of the software system by the end user in an operational environment.

5.5.1 Activity: Operation V&V

The Operation V&V activity evaluates the impact of changes in the operating environment; assesses the effect on the system of any proposed changes; evaluates operating procedures for adherence with the intended use; and analyzes risks affecting the user and the system. The objective of Operation V&V is to evaluate new constraints in the system, assess proposed system changes and their impact on the software, and evaluate operating procedures for correctness and usability.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Operation V&V tasks described in Table 1:

1) Task: Evaluation of new constraints
2) Task: Operating procedures evaluation
3) Task: Hazard analysis
4) Task: Security analysis
5) Task: Risk analysis

5.6 Process: Maintenance

The maintenance process is activated when the software system or associated documentation must be changed in response to a need for system maintenance. The Maintenance V&V activity addresses software system

— Modifications (i.e., corrective, adaptive, or perfective changes)
— Migration (i.e., the movement of software to a new operational environment)
— Retirement (i.e., 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)

5.6.1 Activity: Maintenance V&V

System 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). Modifications of the software system shall be treated as development processes and shall be verified and validated as described in the following:

— 5.1 Process: Management
— 5.4. Process: Development

Software integrity level assignments shall be assessed as described in Clause 4. The software integrity level assignments shall be revised as appropriate to reflect requirements derived from the maintenance process.

For migrating software, the V&V effort shall verify that the migrated software meets the requirements of Clauses 4 and 5.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

If the software V&V was performed in accordance with this standard, the maintenance process shall continue to conform to this standard. If the software was not verified and validated using this standard and appropriate documentation is not available or adequate, the Maintenance 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 shall be taken into consideration.

The objective of Maintenance V&V is to assess proposed software system changes and their impact on the software, evaluate anomalies that are discovered during operation, assess migration requirements, assess retirement requirements, and reperform V&V tasks. The proposed changes are assessed by the *proposed/ baseline change assessment* task of the Management of V&V activity.

The V&V effort shall perform, as specified in Table 2 for the selected software integrity level, the following Maintenance V&V tasks described in Table 1:

1) Task: SVVP revision
2) Task: Anomaly evaluation
3) Task: Criticality analysis
4) Task: Migration assessment
5) Task: Retirement assessment
6) Task: Hazard analysis
7) Task: Security 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 V&V effort shall produce the required outputs listed in Table 1 for each V&V task performed. The format and grouping of the V&V reports may be user defined. The V&V reports shall constitute the *software V&V report* (SVVR).

The V&V reports shall consist of the following:

a) *V&V task reports.* The V&V effort shall document V&V task results and status. Task reports include the following:
    1) Anomaly evaluation
    2) Concept documentation evaluation
    3) Configuration management assessment
    4) Contract verification
    5) Criticality analysis
    6) Evaluation of new constraints
    7) Hardware/ software/user requirements allocation analysis
    8) Hazard analysis
    9) Installation checkout
    10) Installation configuration audit
    11) Interface analysis
    12) Migration assessment

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

13) Operating procedures evaluation
14) Proposed change assessment
15) Recommendations
16) Retirement assessment
17) Review results
18) Risk analysis
19) Security analysis
20) Software design evaluation
21) Software requirements evaluation
22) Source code and source code documentation evaluation
23) System requirements review
24) Test results
25) Traceability analysis

b) *V&V activity summary reports.* An activity summary report shall summarize the results of V&V tasks performed for the following V&V life cycle activities:
   1) Acquisition support
   2) Planning
   3) Concept
   4) Requirements
   5) Design
   6) Implementation
   7) Test
   8) Installation and checkout
   9) Operation
   10) Maintenance

For the operation and maintenance life cycle activities, V&V activity summary reports may be either updates to previous V&V activity summary reports or separate documents.

c) *V&V anomaly reports.* The V&V effort shall document in an anomaly report each anomaly it detects.
d) *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.
e) *Optional V&V reports.* The V&V reports may also include optional reports (i.e., special studies reports and other reports). The V&V effort shall document in a special studies report any special V&V studies conducted during the software life cycle. The V&V effort shall document in a report the results of tasks conducted but not defined in the SVVP. 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. The title of the report may vary according to the subject matter.

Task report(s), V&V activity summary report(s), and anomaly report(s) should be provided as feedback to the software development process regarding the technical quality of each software product and process.

6.2 V&V administrative requirements

The V&V administrative requirements shall consist of the following:

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

1) Anomaly resolution and reporting policy
2) Task iteration policy
3) Deviation policy
4) Control procedures
5) Standards, practices, and conventions

These administrative requirements shall be documented in the SVVP.

6.3 V&V documentation requirements

The scope of V&V documentation consists of V&V test documentation and SVVP documentation. The requirements for documentation are described in the following subclauses.

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 developed by the V&V effort. The V&V test documentation shall conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829™-1998 [B4]). If the V&V effort uses test documentation or test types different from those in this standard (i.e., component, integration, system, acceptance), 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 documented in the test plan, test design(s), test case(s), and test procedure(s).

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" and shall state 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.

7. Software V&V plan outline

The SVVP shall contain the content described in this clause. The user of this standard may adopt any format and section numbering system for the SVVP. The SVVP section numbers listed in this clause are provided to assist readability. An example SVVP outline is shown in the following boxed text.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

SVVP 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
6.1 Task reports
6.2 Activity summary reports

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

SVVP outline (example) ***(continued)***
6.3 Anomaly reports
6.4 V&V final report
6.5 Special studies reports (optional)
6.6 Other reports (optional)
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 test 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 SVVP also shall identify the specific software processes and products covered by the software V&V effort. Date of issue and status, identification of issuing organization, and identification of approval authority shall be provided.

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 also 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). 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 illustrates a sample organizational interrelationship chart.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

7.4.2 SVVP section 4.2: Master schedule

The SVVP shall describe the project life cycle and milestones and 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 should be scheduled to be reperformed 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 (by inclusion or by reference to the criticality analysis) 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.

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.

The SVVP should document the measures to be used by V&V (see Annex E) and should describe how these measures 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.

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

7.5.1 SVVP sections 5.1 through 5.6: Software life cycle[3]

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

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

[3]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.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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.

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 and outputs 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 (see Clause 6).

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). The costs of V&V activities and resources shall be provided or referenced.

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

The SVVP shall specify the purpose, content, format, recipients, and timing of all V&V reports. The V&V reporting requirements are specified in Clause 6.

7.7 SVVP section 7: V&V administrative requirements

The SVVP shall describe the anomaly resolution and reporting, task iteration policy, deviation policy, control procedures, and standards, practices, and conventions.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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; the authority and time lines for resolving anomalies; and the anomaly criticality levels. Classification for software anomalies may be found in IEEE Std 1044™-1993 [B8].

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 should 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 should 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 conform to 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 test documentation requirements

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

The V&V effort may define the format for these documents. IEEE Std 829-1998 [B4] contains sample formats for these test documents.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.1.1 Activity: Management of the V&V Effort (Process: Management)		
V&V tasks	**Required inputs [a]**	**Required outputs**
(1) SVVP generation a) 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. b) Establish a baseline SVVP prior to the Requirements V&V activities. c) Identify project milestones in the SVVP. d) 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) Proposed/baseline change assessment a) Evaluate proposed software changes (i.e., modifications, enhancements, and additions as a result of anomaly corrections or requirement changes) for effects on the systems and previously completed V&V tasks. b) Plan iteration of affected tasks or initiate new tasks to address software proposed changes or baseline changes associated with an iterative development process. c) 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 Supplier development plans and schedules Developer products (produced to date)	Task report(s)—Proposed/ baseline change assessment Updated SVVP Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.1.1 Activity: Management of the V&V Effort (Process: Management) *(continued)*		
V&V tasks	**Required inputs[a]**	**Required outputs**
(3) Management review of the V&V effort a) Review and summarize the V&V effort to define changes to V&V tasks or to redirect the V&V effort. b) Evaluate each anomaly for its impact on the software system and assess whether it is a critical anomaly (e.g., IEEE Std 1044-1993 [B8]). The scope and application of V&V activities and tasks shall be revised to address the causes of these anomalies and risks. c) 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. d) Verify that all V&V tasks conform to task requirements defined in the SVVP. e) Verify that V&V task results have a basis of evidence supporting the results. f) Assess all V&V results and provide recommendations for program acceptance and certification as input to the V&V final report. g) Use results of review to identify process improvement opportunities in the conduct of V&V. h) Review the quality of the products and services to ensure they meet customer requirements i) Review the program risks and initiate actions to mitigate above threshold risks. j) Review program measures to ensure the quality of products and processes. The management review of V&V may use any review methodology (e.g., IEEE Std 1028-1997 [B7]).	SVVP and updates Supplier development plans and schedules Anomaly reports V&V task results [e.g., technical accomplishments, V&V reports, resource utilization, V&V measures (see Annex E), plans, and identified risks]	Task report(s)—Recommendations Updated SVVP V&V activity summary reports Recommendations to the V&V final report
(4) Management and technical review support a) 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. b) Verify timely delivery according to the approved schedule of all software products and documents. The management and technical review support may use any review methodology (e.g., IEEE Std 1028-1997 [B7]).	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 a) Coordinate the V&V effort with organizational (e.g., management, improvement) and supporting processes (e.g., quality assurance, joint review, and problem resolution). b) Identify the V&V data to be exchanged with these processes. c) Document the data exchange requirements in the SVVP.	SVVP Data identified in the SVVP from organizational and supporting processes	Updated SVVP

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.1.1 Activity: Management of the V&V Effort (Process: Management) *(continued)*		
V&V tasks	**Required inputs[a]**	**Required outputs**
(6) Identify process improvement opportunities in the conduct of V&V a) Gather and analyze the lessons learned. b) Gather and analyze the risks identified. c) Gather and analyze the V&V measures. d) Identify and analyze deficiencies in the V&V process. e) Determine and implement corrective actions (e.g., repeat V&V tasks or conduct a new V&V task to address the corrective action or use a different method/technique for executing a V&V task). f) Monitor the efficacy of the corrective actions. g) Document findings in final report.	SVVP Results of analyses Prior end of activity reports Task reports	Updated SVVP Input to the end of activity report Input to the final report New/updated V&V policies/ procedures/ reports Updated V&V infrastructure

5.2.1 Activity: Acquisition support V&V (Process: Acquisition)		
V&V tasks	**Required inputs**	**Required outputs**
(1) Scoping the V&V effort a) Determine the software characteristics (e.g., complexity, criticality, risk, safety level, security level, desired performance, reliability, or other project-unique characteristics) that define the importance of the software to the user. b) Adopt the system integrity scheme assigned to the project. If no system integrity level scheme exists, then one is selected. c) Assign a software integrity level to the system and the software. d) Establish the degree of independence (see Annex C), if any, required for the V&V. e) Determine the minimum V&V tasks for the software integrity level using Table 2 and the selected software integrity level scheme. f) Determine the extent of V&V on reuse software selected for the program (see Annex D). g) Determine the extent of V&V for tools that insert or translate code (e.g., optimizing compilers, auto-code generators). h) Augment the minimum V&V tasks with optional V&V tasks, as necessary. i) Provide an estimate of the V&V budget, including test facilities and tools as required.	Preliminary system description Statement of need Draft RFP or tender System integrity level scheme	SVVP

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.2.1 Activity: Acquisition support V&V (Process: Acquisition) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(2) Planning the interface between the V&V effort and supplier (Preparing the preliminary data and processes for the interface with a supplier to be selected in the supply process) a) Plan the V&V schedule for each V&V task. b) Identify the preliminary list of development processes and products to be evaluated by the V&V processes. c) Describe V&V access rights to proprietary and classified information. d) Coordinated the plan with the acquirer. e) Incorporate the project software integrity level scheme into the planning process.	SVVP Draft RFP or tender Contract	Task Report(s)—Recommendations for RFP or tender Updated SVVP
(3) System requirements review a) 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). 3) Verify whether objective information that can be demonstrated by testing is provided in the requirements (testability). b) 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 Draft RFP or tender	Task report(s)—System requirements review Anomaly report(s)
(4) Acceptance support (The following V&V activities support acceptance in the acquisition process. The activities are described in the development process where required inputs for the V&V activities are generated to aid the understanding of the activity flow.) a) Acceptance V&V test plan generation (5.4.2, Task 6) b) Acceptance V&V test design generation (5.4.3, Task 10) c) Acceptance V&V test case generation (5.4.4., Task 8) d) Acceptance V&V test procedure generation (5.4.5, Task 2) e) Acceptance V&V test execution (5.4.5, Task 5)	Concept documentation SDD IDD SRS IRS Source code Executable code User documentation Test plans, designs, cases, procedures, results Acceptance test plan V&V task results	Task report(s)—Acceptance V&V test plan, design(s), cases, procedures, test results Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.3.1 Activity: Planning V&V (Process: Supply)		
V&V tasks	**Required inputs**	**Required outputs**
(1) Planning the interface between the V&V effort and supplier (Coordinating and documenting the interface data and processes with the selected supplier) a) Review the supplier development plans and schedules to coordinate the V&V effort with development activities. b) Establish procedures to exchange V&V data and results with the development effort. c) Coordinate the plan with the supplier.	SVVP Contract Supplier development plans and schedules	Updated SVVP
(2) Contract verification a) Verify the following: 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 4) Acceptance criteria and procedures are documented in accordance with requirements	SVVP RFP or tender Contract User needs Supplier development plans and schedules	Task Report(s)—Contract verification Updated SVVP Anomaly report(s)

5.4.1 Activity: Concept V&V (Process: Development)		
V&V tasks	**Required inputs**	**Required outputs**
(1) Concept documentation evaluation a) Validate that the concept documentation satisfies user needs and is consistent with acquisition needs. b) Validate constraints of interfacing systems and constraints or limitations of proposed approach. c) 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 environments 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)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.1 Activity: Concept V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(2) Criticality analysis a) Determine whether software integrity levels are established for requirements, detailed functions, software modules, subsystem, or other software partitions. b) 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. c) 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 software system shall be assigned the same integrity level as the highest level assigned to any individual element. d) 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)—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. a) Correctness Verify that performance requirements (e.g., timing, response time, and throughput) allocated to hardware, software, and user interfaces satisfy user needs. b) Accuracy 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 satisfaction of user requirements. c) Completeness 1) Verify that application specific requirements such as functional diversity, fault detection, fault isolation, and diagnostic and error recovery satisfy user needs. 2) Verify that the user's maintenance requirements for the system are completely specified. 3) Verify that the migration from 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 a) Identify all system requirements that will be implemented completely or partially by software. b) Verify that these system requirements are traceable to acquisition needs. c) Start the software requirements traceability analysis with system requirements.	Concept documentation	Task report(s)—Traceability analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.1 Activity: Concept V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(5) Hazard analysis a) 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 4) Identify mitigation strategies for each hazard	Concept documentation	Task report(s)—Hazard analysis Anomaly report(s)
(6) Security analysis a) Review the system owner's definition of an acceptable level of security risk. b) Analyze the system concept from a security perspective, and ensure that potential security risks with respect to confidentiality (disclosure of sensitive information/data), integrity (modification of information/data), availability (withholding of information or services), and accountability (attributing actions to an individual/ process) have been identified. Include an assessment of the sensitivity of the information/ data to be processed. c) Analyze security risks introduced by the system itself as well as those associated with the environment with which the system interfaces.	Concept documentation Preliminary threat and risk assessment (TRA)	Task report(s—Security analysis Anomaly report(s)
(7) Risk analysis a) Identify the technical and management risks. b) Provide recommendations to eliminate, reduce or mitigate the risks.	Concept documentation Supplier development plans and schedules Hazard analysis report Security analysis V&V task results	Task Report(s)—Risk analysis Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.2 Activity: Requirements V&V (Process: Development)		
V&V tasks	**Required inputs**	**Required outputs**
(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 a) Correctness Validate that the relationships between each software requirement and its system requirement are correct. b) Consistency Verify that the relationships between the software and system requirements are specified to a consistent level of detail. c) Completeness 1) Verify that every software requirement is traceable to a system requirement with sufficient detail to show conformance to the system requirement. 2) Verify that all system requirements related to software are traceable to software requirements. d) Accuracy 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 a) Correctness 1) Verify and validate that the software requirements satisfy the system requirements allocated to software within the assumptions, constraints, and operating environment for the system. 2) Verify that the software requirements comply with standards, references, regulations, policies, physical laws, and business rules. 3) Validate the sequences of states and state changes using logic and data flows coupled with domain expertise, prototyping results, engineering principles, or other basis. 4) Validate that the flow of data and control satisfy functionality and performance requirements. 5) Validate data usage and format.	Concept documentation SRS IRS	Task report(s)—Software requirements evaluation Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.2 Activity: Requirements V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(2) Software requirements evaluation *(continued)* b) Consistency 1) Verify that all terms and concepts are documented consistently. 2) Verify that the function interactions and assumptions are consistent and satisfy system requirements and acquisition needs. 3) Verify that there is internal consistency between the software requirements and external consistency with the system requirements. c) Completeness 1) Verify that the following elements are in the SRS or IRS, within the assumptions and constraints of the system: i) Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging) ii) Process definition and scheduling iii) Hardware, software, and user interface descriptions iv) Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety, and security) v) Critical configuration data vi) System, device, and software control (e.g., initialization, transaction and state monitoring, self-testing) 2) Verify that the SRS and IRS satisfy specified configuration management procedures. d) Accuracy 1) Validate that the logic, computational, and interface precision (e.g., truncation and rounding) satisfy the requirements in the system environment. 2) Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws. e) Readability 1) Verify that the documentation is legible, understandable, and unambiguous to the intended audience. 2) Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, and symbols. f) Testability Verify that there are objective acceptance criteria for validating the requirements of the SRS and IRS.		

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.2 Activity: Requirements V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(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 a) Correctness Validate the external and internal system and software interface requirements. b) Consistency Verify that the interface descriptions are consistent between the SRS and IRS. c) Completeness Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security). d) Accuracy Verify that each interface provides information with the required accuracy. e) Testability Verify that there are objective acceptance criteria for validating the interface requirements.	Concept documentation IRS	Task report(s)—Interface analysis Anomaly report(s)
(4) Criticality analysis a) Review and update the existing criticality analysis results from the prior criticality task report using the SRS and IRS. b) 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.	Criticality task report SRS IRS	Task report(s)—Criticality analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.2 Activity: Requirements V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(5) System V&V test plan generation a) Software integrity levels 3 and 4 1) Plan System V&V testing to validate software requirements. 2) Plan tracing of system requirements to test designs, cases, procedures, and results. 3) Plan documentation of test designs, cases, procedures, and results. 4) The System V&V test plan shall address the following: i) Conformance to all system requirements (e.g., functional, performance, security, operation, and maintenance) as complete software end items in the system environment ii) Adequacy of user documentation (e.g., training materials, procedural changes) iii) Performance at boundaries (e.g., data, interfaces) and under stress conditions 5) Verify that the System V&V test plan satisfies the following criteria: i) Conformance to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]) ii) Test coverage of system requirements 6) Validate that the System V&V test plan satisfies the following criteria: i) Appropriateness of test methods and standards used ii) Conformance to expected results iii) Feasibility of system qualification testing iv) Feasibility and testability of operation and maintenance requirements b) Software integrity levels 1 and 2 1) Verify that the developer's system test plan satisfies the following criteria: i) Conformance to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). ii) Test coverage of system requirements 2) Validate that the developer's system test plan satisfies the following criteria: i) Appropriateness of test methods and standards used ii) Conformance to expected results iii) Feasibility of system qualification testing iv) Capability to be operated and maintained	Concept documentation (system requirements) SRS IRS User documentation System test plan	Task report(s)—System V&V test plan Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.2 Activity: Requirements V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(6) Acceptance V&V test plan generation a) Software integrity levels 3 and 4 1) Plan Acceptance V&V testing to validate that the software correctly implements system and software requirements in an operational environment. 2) Plan tracing of acceptance test requirements to test design, cases, procedures, and execution results. 3) Plan documentation of test tasks and results. 4) The Acceptance V&V test plan shall address the following: i) Conformance to acceptance requirements in the operational environment ii) Adequacy of user documentation 5) Verify that the Acceptance V&V test plan satisfies the following criteria: i) Conformance to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]) ii) Test coverage of acceptance requirements 6) Validate that the Acceptance V&V test plan satisfies the following criteria: i) Conformance to expected results ii) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs) b) Software integrity level 2 1) Verify that the acquirer's Acceptance Test Plan conforms to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the acquirer's acceptance test plan satisfies the following criteria i) Test coverage of acceptance requirements ii) Conformance to expected results iii) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs) c) Software integrity level 1 There are no acceptance test requirements.	Concept documentation SRS IRS User documentation Acceptance test plan	Task report(s)—Acceptance V&V test plan Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.2 Activity: Requirements V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(7) Configuration management assessment Verify that the configuration management process is complete and adequate. The task criteria are a) Completeness Verify that there is a process for describing the software product functionality, tracking program versions, and managing changes. b) Adequacy 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 a) Determine software contributions to system hazards. The hazard analysis shall 1) Identify the software requirements that contribute to each system hazard. 2) Validate that the software addresses, controls, or mitigates each hazard.	SRS IRS Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)
(9) Security analysis a) Determine that the security requirements identified in the SRS and IRS address the security risks introduced by the system concept. b) Verify that the system security requirements will mitigate the identified security risks to an acceptable level.	SRS IRS Preliminary TRA	Task report(s)—Security analysis Anomaly report(s)
(10) Risk analysis a) Review and update risk analysis using prior task reports. b) Provide recommendations to eliminate, reduce or mitigate the risks.	Concept documentation SRS IRS Supplier development plans and schedules Hazard analysis report Security analysis V&V task results	Task report(s)—Risk analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development)		
V&V tasks	**Required inputs**	**Required outputs**
(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 a) Correctness Validate the relationship between each design element and the software requirement(s). b) Consistency Verify that the relationships between the design elements and the software requirements are specified to a consistent level of detail. c) Completeness 1) Verify that all design elements are traceable from the software requirements. 2) Verify that all software requirements are traceable to the design elements.	SRS SDD IRS IDD	Task report(s)—Traceability analysis Anomaly report(s)
(2) Software design evaluation Evaluate the design elements (SDD and IDD) for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are a) Correctness 1) Verify and validate that the software design satisfies the software requirements. 2) Verify that the software design complies with standards, references, regulations, policies, physical laws, and business rules. 3) Validate the design sequences of states and state changes using logic and data flows coupled with domain expertise, prototyping results, engineering principles, or other basis. 4) Validate that the flow of data and control satisfy functionality and performance requirements. 5) Validate data usage and format. 6) Assess the appropriateness of design methods and standards used. b) Consistency 1) Verify that all terms and design concepts are documented consistently. 2) Verify that there is internal consistency between the design elements and external consistency with architectural design.		

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(2) Software design evaluation *(continued)* c) Completeness 1) Verify that the following elements are in the SDD, within the assumptions and constraints of the system: i) Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging) ii) Process definition and scheduling iii) Hardware, software, and user interface descriptions iv) Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety, and security) v) Critical configuration data vi) System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing) 2) Verify that the SDD and IDD satisfy specified configuration management procedures. d) Accuracy 1) Validate that the logic, computational, and interface precision (e.g., truncation and rounding) satisfy the requirements in the system environment. 2) Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws. e) Readability 1) Verify that the documentation is legible, understandable, and unambiguous to the intended audience. 2) Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, symbols, and design language, if any. f) Testability 1) Verify that there are objective acceptance criteria for validating each software design element and the system design. 2) Verify that each software design element is testable to objective acceptance criteria.	SRS IRS SDD IDD Design standards (e.g., standards, practices, and conventions)	Task report(s)—Software design evaluation Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(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 a) Correctness Validate the external and internal software interface design in the context of system requirements. b) Consistency Verify that the interface design is consistent between the SDD and IDD. c) Completeness Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security). d) Accuracy Verify that each interface provides information with the required accuracy. e) Testability 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 a) Review and update the existing criticality analysis results from the prior criticality task report using the SDD and IDD. b) 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.	Criticality task report SDD IDD	Task report(s)—Criticality analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(5) Component V&V test plan generation a) Software integrity levels 3 and 4 1) Plan Component V&V testing to validate that the software components (e.g., units, source code modules) correctly implement component requirements. 2) Plan tracing of design requirements to test design, cases, procedures, and results. 3) Plan documentation of test tasks and results. 4) The Component V&V test plan shall address the following: i) Conformance to design requirements ii) Assessment of timing, sizing, and accuracy iii) Performance at boundaries and interfaces and under stress and error conditions iv) Measures of requirements test coverage and software reliability and maintainability 5) Verify that the Component V&V test plan conforms to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 6) Validate that the Component V&V test plan satisfies the following criteria: i) Traceable to the software requirements and design ii) External consistency with the software requirements and design iii) Internal consistency between unit requirements iv) Test coverage of requirements in each unit v) Feasibility of software integration and testing vi) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs) b) Software integrity level 2 1) Verify that the developer's component test conforms to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's component test plan satisfies the following criteria: i) Traceable to the software requirements and design ii) External consistency with the software requirements and design iii) Internal consistency between unit requirements iv) Test coverage of units v) Feasibility of software integration and testing vi) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs) c) Software integrity level 1 There are no component test requirements.	SRS SDD IRS IDD Component test plan	Component V&V—Test plan Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(6) Integration V&V test plan generation a) For software integrity levels 3 and 4 1) 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. 2) Plan tracing of requirements to test design, cases, procedures, and results. 3) Plan documentation of test tasks and results. 4) The Integration V&V test plan shall address the following: i) Conformance to increasingly larger set of functional requirements at each stage of integration i) Assessment of timing, sizing, and accuracy i) Performance at boundaries and under stress conditions i) Measures of requirements test coverage and software reliability 5) Verify that the Integration V&V Test Plan satisfies the following criteria: Conformance to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 6) Validate that the Integration V&V test plan satisfies the following criteria: i) Traceable to the system requirements ii) External consistency with the system requirements iii) Internal consistency iv) Test coverage of the software requirements v) Appropriateness of test standards and methods used vi) Conformance to expected results vii) Feasibility of software qualification testing viii) 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)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
6) Integration V&V test plan generation *(continued)* b) Software integrity levels 1 and 2 1) Verify that the developer's integration test plan conforms to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's integration test plan satisfies the following criteria: i) Traceable to the system requirements ii) External consistency with the system requirements iii) Internal consistency iv) Test coverage of the software requirements v) Appropriateness of test standards and methods vi) Conformance to expected results vii) Feasibility of software qualification testing viii) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs)		
(7) Component V&V test design generation a) Software integrity levels 3 and 4 1) Design tests for component testing. 2) Continue tracing required by the Component V&V test plan. 3) Verify that the Component V&V test designs conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Component V&V test designs satisfy the criteria in V&V activity 5.4.3, Task 5. b) Software integrity level 2 1) Verify that the developer's test designs for component testing conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's component test designs satisfy the criteria in V&V activity 5.4.3, Task 5. c) Software integrity level 1 There are no component test requirements.	SDD IDD User documentation Test plans Test designs	Component V&V test design(s) Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(8) Integration V&V test design generation a) Software integrity levels 3 and 4 1) Design tests for integration testing. 2) Continue tracing required by the Integration V&V test plan. Verify that the Integration V&V test designs conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 3) Validate that the Integration V&V test designs satisfy the criteria in V&V activity 5.4.3, Task 6. b) Software integrity levels 1 and 2 1) Verify that the developer's test designs for integration testing conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's integration test designs satisfy the criteria in V&V activity 5.4.3, Task 6.	SDD IDD User documentation Test plans Test designs	Integration V&V test design(s) Anomaly report(s)
(9) System V&V test design generation a) Software integrity levels 3 and 4 1) Design tests for system testing. 2) Continue tracing required by the System V&V test plan. Verify that the System V&V test designs conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 3) Validate that the System V&V test designs satisfy the criteria in V&V activity 5.4.2, Task 5 b) Software integrity levels 1 and 2 1) Verify that the developer's test designs for system testing conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's system test designs satisfy the criteria in V&V activity 5.4.2, Task 5.	SDD IDD User documentation Test plans Test designs	System V&V test design(s) Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.3 Activity: Design V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(10) Acceptance V&V test design generation a) Software integrity levels 3 and 4 1) Design tests for acceptance testing. 2) Continue tracing required by the Acceptance V&V test plan. Verify that the Acceptance V&V test designs conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 3) Validate that the Acceptance V&V test designs satisfy the criteria in V&V activity 5.4.2, Task 6. b) Software integrity level 2 1) Verify that the acquirer's test designs for acceptance testing conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the acquirer's acceptance test designs satisfy the criteria in V&V activity 5.4.2, Task 6. c) Software integrity level 1 There are no acceptance test requirements.	SDD IDD User documentation Test plans Test designs	Acceptance V&V test design(s) Anomaly report(s)
(11) Hazard Analysis a) Verify that logic design and associated data elements correctly implement the critical requirements and introduce no new hazards. b) Update the hazard analysis.	SDD IDD Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)
(12) Security analysis a) Verify that the architecture and detailed design outputs adequately address the identified security requirements. This verification includes both the system itself and security risks introduced as a result of interfacing with external components.	SDD IDD	Task report(s)—Security analysis Anomaly report(s)
(13) Risk analysis a) Review and update risk analysis using prior task reports. b) Provide recommendations to eliminate, reduce, or mitigate the risks.	SDD IDD Supplier development plans and schedules Hazard analysis report Security analysis V&V task results	Task report(s)—Risk analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development)		
V&V tasks	**Required inputs**	**Required outputs**
(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 a) Correctness Validate the relationship between the source code components and design element(s). b) Consistency Verify that the relationships between the source code components and design elements are specified to a consistent level of detail. c) Completeness 1) Verify that all source code components are traceable from the design elements. 2) 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 and source code documentation) for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are a) Correctness 1) Verify and validate that the source code component satisfies the software design. 2) Verify that the source code components comply with standards, references, regulations, policies, physical laws, and business rules. 3) 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. 4) Validate that the flow of data and control satisfy functionality and performance requirements. 5) Validate data usage and format. 6) Assess the appropriateness of coding methods and standards.	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)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(2) Source code and source code documentation evaluation ***(continued)*** b) Consistency 1) Verify that all terms and code concepts are documented consistently. 1) Verify that there is internal consistency between the source code components. 1) Validate external consistency with the software design and requirements. c) Completeness 1) Verify that the following elements are in the source code, within the assumptions and constraints of the system: i) Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging) ii) Process definition and scheduling iii) Hardware, software, and user interface descriptions iv) Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety, and security) v) Critical configuration data vi) System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing) 7) Verify that the source code documentation satisfies specified configuration management procedures. d) Accuracy 1) Validate the logic, computational, and interface precision (e.g., truncation and rounding) in the system environment. 2) Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws. e) Readability 1) Verify that the documentation is legible, understand-able, and unambiguous to the intended audience. 2) Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, and symbols. f) Testability 1) Verify that there are objective acceptance criteria for validating each source code component. 2) Verify that each source code component is testable against objective acceptance criteria.		

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(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 a) Correctness Validate the external and internal software interface code in the context of system requirements. b) Consistency Verify that the interface code is consistent between source code components and to external interfaces (i.e., hardware, user, operator, and other software). c) Completeness Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security). d) Accuracy Verify that each interface provides information with the required accuracy. e) Testability Verify that there are objective acceptance criteria for validating the interface code.	Concept documentation (system requirements) SDD IDD Source code User documentation	Task report(s)—Interface analysis Anomaly report(s)
(4) Criticality analysis a) Review and update the existing criticality analysis results from the prior criticality task report using the source code. b) 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.	Criticality task report Source code	Task report(s)—Criticality analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(5) Component V&V test case generation a) Software integrity levels 3 and 4 1) Develop V&V test cases for component testing. 2) Continue tracing required by the Component V&V test plan. 3) Verify that the Component V&V test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Component V&V test cases satisfy the criteria in V&V activity 5.4.3, Task 5. b) Software integrity level 2 1) Verify that the developer's component test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's component test cases satisfy the criteria in V&V activity 5.4.3, Task 5. c) Software integrity level 1 There are no component test requirements.	SRS IRS SDD IDD User documentation Test design Test cases	Component V&V test cases Anomaly report(s)
(6) Integration V&V test case generation a) Software integrity levels 3 and 4 1) Develop V&V test cases for integration testing. 2) Continue tracing required by the Integration V&V test plan. 3) Verify that the Integration V&V test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Integration V&V test cases satisfy the criteria in V&V activity 5.4.3, Task 6. b) Software integrity levels 1 and 2 1) Verify that the developer's integration test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's integration test cases satisfy the criteria in V&V activity 5.4.3, Task 6.	SRS IRS SDD IDD User documentation Test design Test cases	Integration V&V test cases Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(7) System V&V test case generation a) Software integrity levels 3 and 4 1) Develop V&V test cases for system testing. 2) Continue tracing required by the System V&V test plan. 3) Verify that the System V&V test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the System V&V test cases satisfy the criteria in V&V activity 5.4.2, Task 5. b) Software integrity levels 1 and 2 1) Verify that the developer's system test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's system test cases satisfy the criteria in V&V activity 5.4.2, Task 5.	SRS IRS SDD IDD User documentation Test design Test cases	System V&V test cases Anomaly report(s)
(8) Acceptance V&V test case generation a) Software integrity levels 3 and 4 1) Develop V&V test cases for acceptance testing. 2) Continue tracing required by the Acceptance V&V test plan. 3) Verify that the Acceptance V&V test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Acceptance V&V test cases satisfy the criteria in V&V activity 5.4.2, Task 6. b) Software integrity level 2 1) Verify that the acquirer's acceptance test cases conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the acquirer's acceptance test cases satisfy the criteria in V&V activity 5.4.2, Task 6. c) Software integrity level 1 There are no acceptance test requirements.	SRS IRS SDD IDD User documentation Test design Test cases	Acceptance V&V test cases Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(9) Component V&V test procedure generation a) Software integrity levels 3 and 4 1) Develop V&V test procedures for component testing. 2) Continue tracing required by the Component V&V test plan. 3) Verify that the Component V&V test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Component V&V test procedures satisfy the criteria in V&V activity 5.4.3, Task 5. b) Software integrity level 2 1) Verify that the developer's component test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's component test procedures satisfy the criteria in V&V activity 5.4.3, Task 5. c) Software integrity level 1 There are no component test requirements.	SRS IRS SDD IDD User documentation Test cases Test procedures	Component V&V test procedures Anomaly report(s)
(10) Integration V&V test procedure generation a) Software integrity levels 3 and 4 1) Develop V&V test procedures for Integration testing. 2) Continue tracing required by the Integration V&V test plan. 3) Verify that the Integration V&V test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Integration V&V test procedures satisfy the criteria in V&V activity 5.4.3, Task 6. b) Software integrity levels 1 and 2 1) Verify that the developer's integration test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's integration test procedures satisfy the criteria in V&V activity 5.4.3, Task 6.	SRS IRS SDD IDD User documentation Test cases Test procedures	Integration V&V test procedures Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(11) System V&V test procedure generation a) Software integrity levels 3 and 4 1) Develop V&V test procedures for system testing. 2) Continue tracing required by the System V&V test plan. 3) Verify that the System V&V test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the System V&V test procedures satisfy the criteria in V&V activity 5.4.2 Task 5. b) Software integrity levels 1 and 2 1) Verify that the developer's system test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's system test procedures satisfy the criteria in V&V activity 5.4.2, Task 5.	SRS IRS SDD IDD User documentation Test cases Test procedures	System V&V test procedures Anomaly report(s)
(12) Component V&V test execution a) Software integrity levels 3 and 4 1) Perform V&V component testing. 2) Analyze test results to validate that software correctly implements the design. 3) Validate that the test results trace to test criteria established by the test traceability in the test planning documents. 4) Document the results as required by the Component V&V test plan. 5) Use the V&V component test results to validate that the software satisfies the V&V test acceptance criteria. 6) Document discrepancies between actual and expected test results. b) Software integrity level 2 Use the developer's component test results to validate that the software satisfies the test acceptance criteria. c) 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)
(13) Hazard analysis a) Verify that the implementation and associated data elements correctly implement the critical requirements and introduce no new hazards. b) Update the hazard analysis.	Source code SDD IDD Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.4 Activity: Implementation V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(14) Security analysis a) Verify that the implementation is completed in accordance with the system design in that it addresses the identified security risks and that the implementation does not introduce new security risks through coding flaws or compiler error.	Source Code SDD IDD	Task report(s)—Security analysis Anomaly report(s)
(15) Risk analysis a) 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 Security analysis V&V task results	Task report(s)—Risk analysis Anomaly report(s)

5.4.5 Activity: Test V&V (Process: Development)		
V&V tasks	**Required inputs**	**Required outputs**
(1) Traceability analysis Analyze relationships in the V&V test plans, designs, cases, and procedures for correctness and completeness. The task criteria are a) Correctness Verify that there is a valid relationship between the V&V test plans, designs, cases, and procedures. b) 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)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.5 Activity: Test V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(2) Acceptance V&V test procedure generation a) Software integrity levels 3 and 4 1) Develop V&V test procedures for acceptance testing. 2) Continue the tracing required by the Acceptance V&V test plan. 3) Verify that the Acceptance V&V test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 4) Validate that the Acceptance V&V test procedures satisfy the criteria in V&V activity 5.4.2, Task 6. b) Software integrity level 2 1) Verify that the developer's acceptance test procedures conform to project-defined test document purpose, format, and content (e.g., see IEEE Std 829-1998 [B4]). 2) Validate that the developer's acceptance test procedures satisfy the criteria in V&V activity 5.4.2, Task 6. c) 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)
(3) Integration V&V test execution a) Software integrity levels 3 and 4 1) Perform V&V integration testing. 2) Analyze test results to verify that the software components are integrated correctly. 3) Validate that the test results trace to test criteria established by the test traceability in the test planning documents. 4) Document the results as required by the Integration V&V test plan. 5) Use the V&V integration test results to validate that the software satisfies the V&V test acceptance criteria. 6) Document discrepancies between actual and expected test results. b) 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)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.5 Activity: Test V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(4) System V&V test execution a) Software integrity levels 3 and 4 1) Perform V&V system testing. 2) Analyze test results to validate that the software satisfies the system requirements. 3) Validate that the test results trace to test criteria established by the test traceability in the test planning documents. 4) Document the results as required by the System V&V test plan. 5) Use the V&V system test results to validate that the software satisfies the V&V test acceptance criteria. 6) Document discrepancies between actual and expected test results. b) 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 a) Software integrity levels 3 and 4 1) Perform acceptance V&V testing. 2) Analyze test results to validate that the software satisfies the system requirements. 3) Validate that the test results trace to test criteria established by the test traceability in the test planning documents. 4) Document the results as required by the Acceptance V&V test plan. 5) Use the acceptance V&V test results to validate that the software satisfies the V&V test acceptance criteria. 6) Document discrepancies between actual and expected test results. b) Software integrity level 2 Use the acquirer's acceptance test results to verify that the software satisfies the test acceptance criteria. c) 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 V&V task results	Task report(s)—Test results Anomaly report(s)
(6) Hazard analysis a) Verify that the test instrumentation does not introduce new hazards. b) Update the hazard analysis.	Source code Executable code Test results Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.5 Activity: Test V&V (Process: Development) ***(continued)***		
V&V tasks	**Required inputs**	**Required outputs**
(7) Security analysis a) Verify that the implemented system does not increase security risk.	Source code Executable code	Task report(s)—Security analysis Anomaly report(s)
(8) Risk analysis a) Review and update risk analysis using prior task reports. b) Provide recommendations to eliminate, reduce, or mitigate the risks.	Supplier development plans and schedules Hazard analysis report Security analysis V&V task results	Task report(s)—Risk analysis Anomaly report(s)

5.4.6 Activity: Installation and checkout V&V (Process: Development)		
V&V tasks	**Required inputs**	**Required outputs**
(1) Installation configuration audit a) Verify that all software products required to correctly install and operate the software are present in the installation package. b) 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 a) Conduct analyses or tests to verify that the installed software corresponds to the software subjected to V&V. b) Verify that the software code and databases initialize, execute, and terminate as specified. c) In the transition from one version of software to the next, validate that the software can be removed from the system without affecting the functionality of the remaining system components. d) 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 a) Verify that the installation procedures and installation environment does not introduce new hazards. b) Update the hazard analysis.	Installation package Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.4.6 Activity: Installation and checkout V&V (Process: Development) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(4) Security analysis a) Verify that the installed software does not introduce new or increased vulnerabilities or security risks to the overall system.	Installation package User documentation	Task report(s)—Security analysis Anomaly report(s)
(5) Risk analysis a) Review and update risk analysis using prior task reports. b) Provide recommendations to eliminate, reduce, or mitigate the risks.	Installation package Supplier development plans and schedules Security analysis V&V task results	Task report(s)—Risk analysis Anomaly report(s)
(6) V&V final report generation a) Summarize in the V&V final report the V&V activities, tasks, and results, including status and disposition of anomalies. b) b) Provide an assessment of the overall software quality and provide recommendations.	V&V activity summary report(s)	Task report(s)—V&V final report

5.5.1 Activity: Operation V&V (Process: Operation)		
V&V tasks	**Required inputs**	**Required outputs**
(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. (See NOTE.)	SVVP New constraints	Task report(s)—Evaluation of new constraints
(2) 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)
(3) Hazard analysis a) Verify that the operating procedures and operational environment does not introduce new hazards. b) b) Update the hazard analysis.	Operating procedures Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)
(4) Security analysis a) Verify that no new security risks are introduced due to changes in the operational environment. b) Over time, changes in external interfaces, threats or technology in general require that an updated security analysis be performed to determine an updated residual risk.	New constraints Environmental changes Operating procedures	Task report(s)—Security analysis

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.5.1 Activity: Operation V&V (Process: Operation) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(5) Risk analysis a) Review and update risk analysis using prior task reports. b) Provide recommendations to eliminate reduce or mitigate the risks.	Installation package Proposed changes Hazard analysis report Security analysis Supplier development plans and schedules Operation problem reports V&V task results	Task report(s)—Risk Analysis Anomaly report(s)

5.6.1 Activity: Maintenance V&V (Process: Maintenance)		
V&V tasks	**Required inputs**	**Required outputs**
(1) SVVP revision a) Revise the SVVP to conform to approved changes. b) When the development documentation required by this standard is not available, generate a new SVVP and consider the methods in Annex D for deriving the required development documentation.	SVVP Approved changes Installation package Supplier development plans and schedules	Updated SVVP
(2) Anomaly evaluation Evaluate the effect of software operation anomalies.	Anomaly report(s)	Task report(s)—Anomaly evaluation
(3) Criticality analysis a) Determine the software integrity levels for proposed modifications. b) 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)
(4) Migration assessment a) Assess whether the software requirements and implementation address the following: 1) Specific migration requirements 2) Migration tools 3) Conversion of software products and data 4) Software archiving 5) Support for the prior environment 6) User notification	Installation package Approved changes	Task report(s)—Migration assessment Anomaly report(s)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

5.6.1 Activity: Maintenance V&V (Process: Maintenance) *(continued)*		
V&V tasks	**Required inputs**	**Required outputs**
(5) Retirement assessment a) Assess whether the installation package addresses the following: 1) Software support 2) Impact on existing systems and databases 3) Software archiving 4) Transition to a new software product 5) User notification	Installation package Approved changes	Task report(s)—Retirement assessment Anomaly report(s)
(6) Hazard analysis a) Verify that software modifications correctly implement the critical requirements and introduce no new hazards. b) Update the hazard analysis.	Proposed changes Installation Package Hazard analysis report	Task report(s)—Hazard analysis Anomaly report(s)
(7) Security analysis Verify that proposed changes/updates to the software do not introduce new or increased security risks to the overall system.	Proposed changes Installation package	Task report(s)—Security analysis
(8) Risk analysis a) Review and update risk analysis using prior task reports. b) Provide recommendations to eliminate, reduce, or mitigate the risks.	Installation package Proposed changes Hazard analysis report Security analysis Supplier development plans and schedules Operation problem reports V&V task results	Task report(s)—Risk analysis Anomaly report(s)
(9) Task iteration a) Perform V&V tasks, as needed, to ensure that 1) Planned changes are implemented correctly 2) Documentation is complete and current 3) Changes do not cause unacceptable or unintended system behaviors.	Approved changes Installation package	Task report(s)—Anomaly report(s)
NOTE—Software changes are maintenance activities (see 5.6.1).		

[a] Other inputs may be used. For any V&V activity and task, all of the required inputs and outputs from preceding activities and tasks may be used, but for conciseness, only the primary inputs are listed.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

Life cycle processes	Process: Acquisition (see 5.2)				Process: Supply (see 5.3)				Process: Development (see 5.4)																								Process: Operation (see 5.5)				Process: Maintenance (see 5.6)			
V&V activities	Activity: Acquisition support V&V (see 5.2.1)				Activity: Planning V&V (see 5.3.1)				Activity: Concept V&V (see 5.4.1)				Activity: Requirements V&V (see 5.4.2)				Activity: Design V&V (see 5.4.3)				Activity: Implementation V&V (see 5.4.4)				Activity: Test V&V (see 5.4.5)				Activity: Installation/ checkout V&V (see 5.4.6)				Activity: Operation V&V (see 5.5.1)				Activity: Maintenance V&V (see 5.6.1)			
Software integrity levels	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
Acceptance support													X	X	X		X	X	X		X	X	X		X	X	X													
Acceptance V&V test case generation																					X	X	X																	
Acceptance V&V test design generation																	X	X	X																					
Acceptance V&V test execution																									X	X	X													
Acceptance V&V test plan generation													X	X	X																									
Acceptance V&V test procedure generation																									X	X	X													
Anomaly evaluation																																					X	X	X	
Component V&V test case generation																					X	X	X																	
Component V&V test design generation																	X	X	X																					
Component V&V test execution																					X	X	X																	

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

Life cycle processes	Process: Acquisition (see 5.2)				Process: Supply (see 5.3)				Process: Development (see 5.4)																								Process: Operation (see 5.5)				Process: Mainte-nance (see 5.6)			
V&V activities	**Activity: Acquisition support V&V (see 5.2.1)**				**Activity: Planning V&V (see 5.3.1)**				**Activity: Concept V&V (see 5.4.1)**				**Activity: Require-ments V&V (see 5.4.2)**				**Activity: Design V&V (see 5.4.3)**				**Activity: Implemen-tation V&V (see 5.4.4)**				**Activity: Test V&V (see 5.4.5)**				**Activity: Installation/ checkout V&V (see 5.4.6)**				**Activity: Operation V&V (see 5.5.1)**				**Activity: Mainte-nance V&V (see 5.6.1)**			
Software integrity levels	**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**
Component V&V test plan generation																	X	X	X																					
Component V&V test procedure generation																					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	X													X	X	X	X
Hardware/software/user requirements allocation analysis									X																															
Hazard analysis									X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		
Identify improvement opportunities in the conduct 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	X	X	X	X	X
Installation checkout																													X	X										
Installation configuration audit																													X	X										

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

Life cycle processes	Process: Acquisition (see 5.2)				Process: Supply (see 5.3)				Process: Development (see 5.4)																								Process: Operation (see 5.5)				Process: Mainte-nance (see 5.6)			
V&V activities	Activity: Acquisition support V&V (see 5.2.1)				Activity: Planning V&V (see 5.3.1)				Activity: Concept V&V (see 5.4.1)				Activity: Require-ments V&V (see 5.4.2)				Activity: Design V&V (see 5.4.3)				Activity: Implemen-tation V&V (see 5.4.4)				Activity: Test V&V (see 5.4.5)				Activity: Installation/ checkout V&V (see 5.4.6)				Activity: Operation V&V (see 5.5.1)				Activity: Mainte-nance V&V (see 5.6.1)			
Software integrity levels	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
Integration V&V test case generation																					X	X	X	X																
Integration V&V test design generation																	X	X	X	X																				
Integration V&V test execution																									X	X	X	X												
Integration V&V test plan generation																	X	X	X	X																				
Integration V&V test procedure generation																					X	X	X	X																
Interface analysis													X	X	X		X	X	X		X	X	X																	
Interface with organizational and supporting processes	X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		
Management and technical review support	X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		
Management review of the V&V effort	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	X	X	X	X	X
Migration assessment																																					X	X		
New constraints evaluation																																	X	X	X					
Operating procedures evaluation																																	X	X						

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

Life cycle processes	Process: Acquisition (see 5.2)				Process: Supply (see 5.3)				Process: Development (see 5.4)																												Process: Operation (see 5.5)				Process: Maintenance (see 5.6)			
V&V activities	**Activity: Acquisition support V&V (see 5.2.1)**				**Activity: Planning V&V (see 5.3.1)**				**Activity: Concept V&V (see 5.4.1)**				**Activity: Requirements V&V (see 5.4.2)**				**Activity: Design V&V (see 5.4.3)**				**Activity: Implementation V&V (see 5.4.4)**				**Activity: Test V&V (see 5.4.5)**				**Activity: Installation/ checkout V&V (see 5.4.6)**				**Activity: Operation V&V (see 5.5.1)**				**Activity: Maintenance V&V (see 5.6.1)**			
Software integrity levels	**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**
Planning the interface between the V&V effort and supplier	X	X	X	X	X	X	X	X																																
Proposed/baseline change assessment									X	X	X		X	X	X		X	X	X		X	X	X		X	X	X		X	X	X		X	X	X		X	X	X	
Retirement assessment																																					X	X		
Risk analysis									X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		
Scoping the V&V effort	X	X	X	X																																				
Security analysis									X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		
Software design evaluation																	X	X	X	X																				
Software requirements evaluation													X	X	X	X																								
SVVP generation	X	X	X	X	X	X	X	X	X	X	X	X																									X	X	X	X
SVVP revision																																					X	X	X	X
Source code and source code documentation evaluation																					X	X	X																	
System requirements review	X	X	X	X																																				
System V&V test case generation																					X	X	X	X																

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

Life cycle processes	Process: Acquisition (see 5.2)				Process: Supply (see 5.3)				Process: Development (see 5.4)																												Process: Operation (see 5.5)				Process: Maintenance (see 5.6)			
V&V activities	**Activity: Acquisition support V&V (see 5.2.1)**				**Activity: Planning V&V (see 5.3.1)**				**Activity: Concept V&V (see 5.4.1)**				**Activity: Requirements V&V (see 5.4.2)**				**Activity: Design V&V (see 5.4.3)**				**Activity: Implementation V&V (see 5.4.4)**				**Activity: Test V&V (see 5.4.5)**				**Activity: Installation/ checkout V&V (see 5.4.6)**				**Activity: Operation V&V (see 5.5.1)**				**Activity: Maintenance V&V (see 5.6.1)**			
Software integrity levels	**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**
System V&V test design generation																	X	X	X	X																				
System V&V test execution																									X	X	X	X												
System V&V test plan generation													X	X	X	X																								
System V&V test procedure generation																					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	X								

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

V&V life cycle activities	Activity: Management of V&V (see 5.1.1)	Activity: Acquisition support V&V (see 5.2.1)	Activity: Planning V&V (see 5.3.1)	Activity: Concept V&V (see 5.4.1)	Activity: Requirements V&V (see 5.4.2)	Activity: Design V&V (see 5.4.3)	Activity: Implementation V&V (see 5.4.4)	Activity: Test V&V (see 5.4.5)	Activity: Installation/checkout V&V (see 5.4.6)	Activity: Operation V&V (see 5.5.1)	Activity: Maintenance V&V (see 5.6.1)
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
Requirements					X						X
Design						X					X
Source code							X				X
Test plan					X	X	X		X		X
Test design						X	X		X		X
Test case						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
Proposal evaluation support		X									
Qualification testing								X	X		X
Regression analysis and testing					X	X	X	X	X		X

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

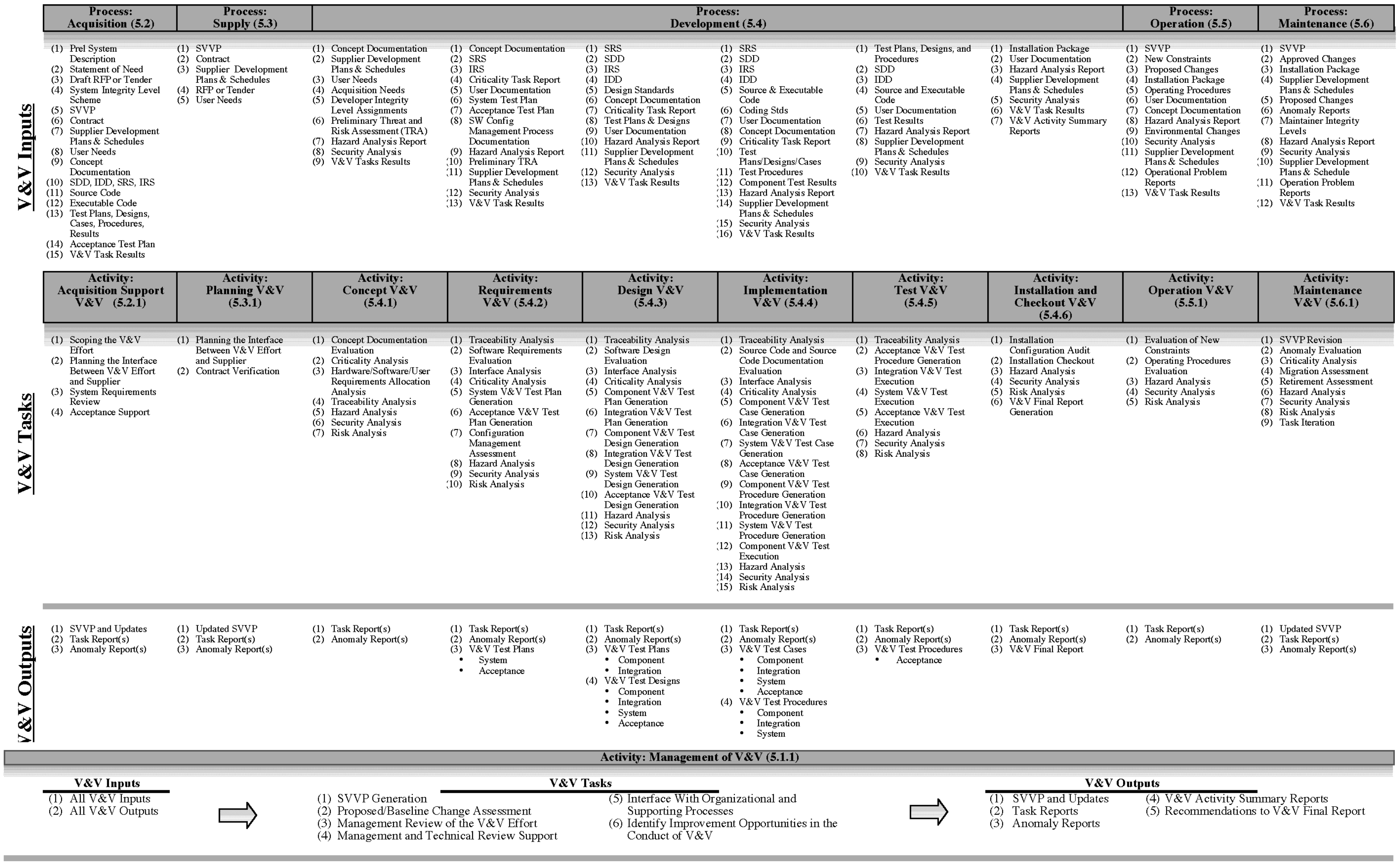
Table 3—Optional V&V tasks and suggested applications in the life cycle *(continued)*

V&V life cycle activities	Activity: Management of V&V (see 5.1.1)	Activity: Acquisition support V&V (see 5.2.1)	Activity: Planning V&V (see 5.3.1)	Activity: Concept V&V (see 5.4.1)	Activity: Requirements V&V (see 5.4.2)	Activity: Design V&V (see 5.4.3)	Activity: Implementation V&V (see 5.4.4)	Activity: Test V&V (see 5.4.5)	Activity: Installation/checkout V&V (see 5.4.6)	Activity: Operation V&V (see 5.5.1)	Activity: Maintenance V&V (see 5.6.1)
Reusability analysis	X	X	X	X	X	X	X				X
Reuse analysis	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	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
Usability analysis				X	X	X	X	X	X	X	X
User documentation evaluation	X			X	X	X	X	X	X	X	X
User training	X							X	X	X	X
V&V tool plan generation	X	X	X								X
Walk-throughs											
Design						X					X
Requirements					X						X
Source code							X				X
Test								X	X		X

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

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.



	Process: Acquisition (5.2)	Process: Supply (5.3)	Process: Development (5.4)						Process: Operation (5.5)	Process: Maintenance (5.6)
V&V Inputs	(1) Prel System Description (2) Statement of Need (3) Draft RFP or Tender (4) System Integrity Level Scheme (5) SVVP (6) Contract (7) Supplier Development Plans & Schedules (8) User Needs (9) Concept Documentation (10) SDD, IDD, SRS, IRS (11) Source Code (12) Executable Code (13) Test Plans, Designs, Cases, Procedures, Results (14) Acceptance Test Plan (15) V&V Task Results	(1) SVVP (2) Contract (3) Supplier Development Plans & Schedules (4) RFP or Tender (5) User Needs	(1) Concept Documentation (2) Supplier Development Plans & Schedules (3) User Needs (4) Acquisition Needs (5) Developer Integrity Level Assignments (6) Preliminary Threat and Risk Assessment (TRA) (7) Hazard Analysis Report (8) Security Analysis (9) V&V Tasks Results	(1) Concept Documentation (2) SRS (3) IRS (4) Criticality Task Report (5) User Documentation (6) System Test Plan (7) Acceptance Test Plan (8) SW Config Management Process Documentation (9) Hazard Analysis Report (10) Preliminary TRA (11) Supplier Development Plans & Schedules (12) Security Analysis (13) V&V Task Results	(1) SRS (2) SDD (3) IRS (4) IDD (5) Design Standards (6) Concept Documentation (7) Criticality Task Report (8) Test Plans & Designs (9) User Documentation (10) Hazard Analysis Report (11) Supplier Development Plans & Schedules (12) Security Analysis (13) V&V Task Results	(1) SRS (2) SDD (3) IRS (4) IDD (5) Source & Executable Code (6) Coding Stds (7) User Documentation (8) Concept Documentation (9) Criticality Task Report (10) Test Plans/Designs/Cases (11) Test Procedures (12) Component Test Results (13) Hazard Analysis Report (14) Supplier Development Plans & Schedules (15) Security Analysis (16) V&V Task Results	(1) Test Plans, Designs, and Procedures (2) SDD (3) IDD (4) Source and Executable Code (5) User Documentation (6) Test Results (7) Hazard Analysis Report (8) Supplier Development Plans & Schedules (9) Security Analysis (10) V&V Task Results	(1) Installation Package (2) User Documentation (3) Hazard Analysis Report (4) Supplier Development Plans & Schedules (5) Security Analysis (6) V&V Task Results (7) V&V Activity Summary Reports	(1) SVVP (2) New Constraints (3) Proposed Changes (4) Installation Package (5) Operating Procedures (6) User Documentation (7) Concept Documentation (8) Hazard Analysis Report (9) Environmental Changes (10) Security Analysis (11) Supplier Development Plans & Schedules (12) Operational Problem Reports (13) V&V Task Results	(1) SVVP (2) Approved Changes (3) Installation Package (4) Supplier Development Plans & Schedules (5) Proposed Changes (6) Anomaly Reports (7) Maintainer Integrity Levels (8) Hazard Analysis Report (9) Security Analysis (10) Supplier Development Plans & Schedule (11) Operation Problem Reports (12) V&V Task Results
	Activity: Acquisition Support V&V (5.2.1)	Activity: Planning V&V (5.3.1)	Activity: Concept V&V (5.4.1)	Activity: Requirements V&V (5.4.2)	Activity: Design V&V (5.4.3)	Activity: Implementation V&V (5.4.4)	Activity: Test V&V (5.4.5)	Activity: Installation and Checkout V&V (5.4.6)	Activity: Operation V&V (5.5.1)	Activity: Maintenance V&V (5.6.1)
V&V Tasks	(1) Scoping the V&V Effort (2) Planning the Interface Between V&V Effort and Supplier (3) System Requirements Review (4) Acceptance Support	(1) Planning the Interface Between V&V Effort and Supplier (2) Contract Verification	(1) Concept Documentation Evaluation (2) Criticality Analysis (3) Hardware/Software/User Requirements Allocation Analysis (4) Traceability Analysis (5) Hazard Analysis (6) Security Analysis (7) Risk Analysis	(1) Traceability Analysis (2) Software Requirements Evaluation (3) Interface Analysis (4) Criticality Analysis (5) System V&V Test Plan Generation (6) Acceptance V&V Test Plan Generation (7) Configuration Management Assessment (8) Hazard Analysis (9) Security Analysis (10) Risk Analysis	(1) Traceability Analysis (2) Software Design Evaluation (3) Interface Analysis (4) Criticality Analysis (5) Component V&V Test Plan Generation (6) Integration V&V Test Plan Generation (7) Component V&V Test Design Generation (8) Integration V&V Test Design Generation (9) System V&V Test Design Generation (10) Acceptance V&V Test Design Generation (11) Hazard Analysis (12) Security Analysis (13) Risk Analysis	(1) Traceability Analysis (2) Source Code and Source Code Documentation Evaluation (3) Interface Analysis (4) Criticality Analysis (5) Component V&V Test Case Generation (6) Integration V&V Test Case Generation (7) System V&V Test Case Generation (8) Acceptance V&V Test Case Generation (9) Component V&V Test Procedure Generation (10) Integration V&V Test Procedure Generation (11) System V&V Test Procedure Generation (12) Component V&V Test Execution (13) Hazard Analysis (14) Security Analysis (15) Risk Analysis	(1) Traceability Analysis (2) Acceptance V&V Test Procedure Generation (3) Integration V&V Test Execution (4) System V&V Test Execution (5) Acceptance V&V Test Execution (6) Hazard Analysis (7) Security Analysis (8) Risk Analysis	(1) Installation Configuration Audit (2) Installation Checkout (3) Hazard Analysis (4) Security Analysis (5) Risk Analysis (6) V&V Final Report Generation	(1) Evaluation of New Constraints (2) Operating Procedures Evaluation (3) Hazard Analysis (4) Security Analysis (5) Risk Analysis	(1) SVVP Revision (2) Anomaly Evaluation (3) Criticality Analysis (4) Migration Assessment (5) Retirement Assessment (6) Hazard Analysis (7) Security Analysis (8) Risk Analysis (9) Task Iteration
V&V Outputs	(1) SVVP and Updates (2) Task Report(s) (3) Anomaly Report(s)	(1) Updated SVVP (2) Task Report(s) (3) Anomaly Report(s)	(1) Task Report(s) (2) Anomaly Report(s)	(1) Task Report(s) (2) Anomaly Report(s) (3) V&V Test Plans • System • Acceptance	(1) Task Report(s) (2) Anomaly Report(s) (3) V&V Test Plans • Component • Integration (4) V&V Test Designs • Component • Integration • System • Acceptance	(1) Task Report(s) (2) Anomaly Report(s) (3) V&V Test Cases • Component • Integration • System • Acceptance (4) V&V Test Procedures • Component • Integration • System	(1) Task Report(s) (2) Anomaly Report(s) (3) V&V Test Procedures • Acceptance	(1) Task Report(s) (2) Anomaly Report(s) (3) V&V Final Report	(1) Task Report(s) (2) Anomaly Report(s)	(1) Updated SVVP (2) Task Report(s) (3) Anomaly Report(s)

Activity: Management of V&V (5.1.1)

V&V Inputs	V&V Tasks		V&V Outputs	
(1) All V&V Inputs (2) All V&V Outputs	(1) SVVP Generation (2) Proposed/Baseline Change Assessment (3) Management Review of the V&V Effort (4) Management and Technical Review Support	(5) Interface With Organizational and Supporting Processes (6) Identify Improvement Opportunities in the Conduct of V&V	(1) SVVP and Updates (2) Task Reports (3) Anomaly Reports	(4) V&V Activity Summary Reports (5) Recommendations to V&V Final Report

Notes:
1. The V&V tasks may be performed concurrently. The sequential (waterfall) model is shown as an example.
2. V&V tasks listed in this figure are the minimum for Software Integrity Level 4 (Highest Integrity Level).
3. Table 1 contains a complete list of V&V tasks. Table 2 defines the minimum V&V tasks for each software integrity level.

The numbers in parentheses (e.g., 5.1.1) correspond to the subclauses in Clause 5, *Verification and Validation Processes*.

Figure 1—An example of V&V processes, activities, and tasks

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

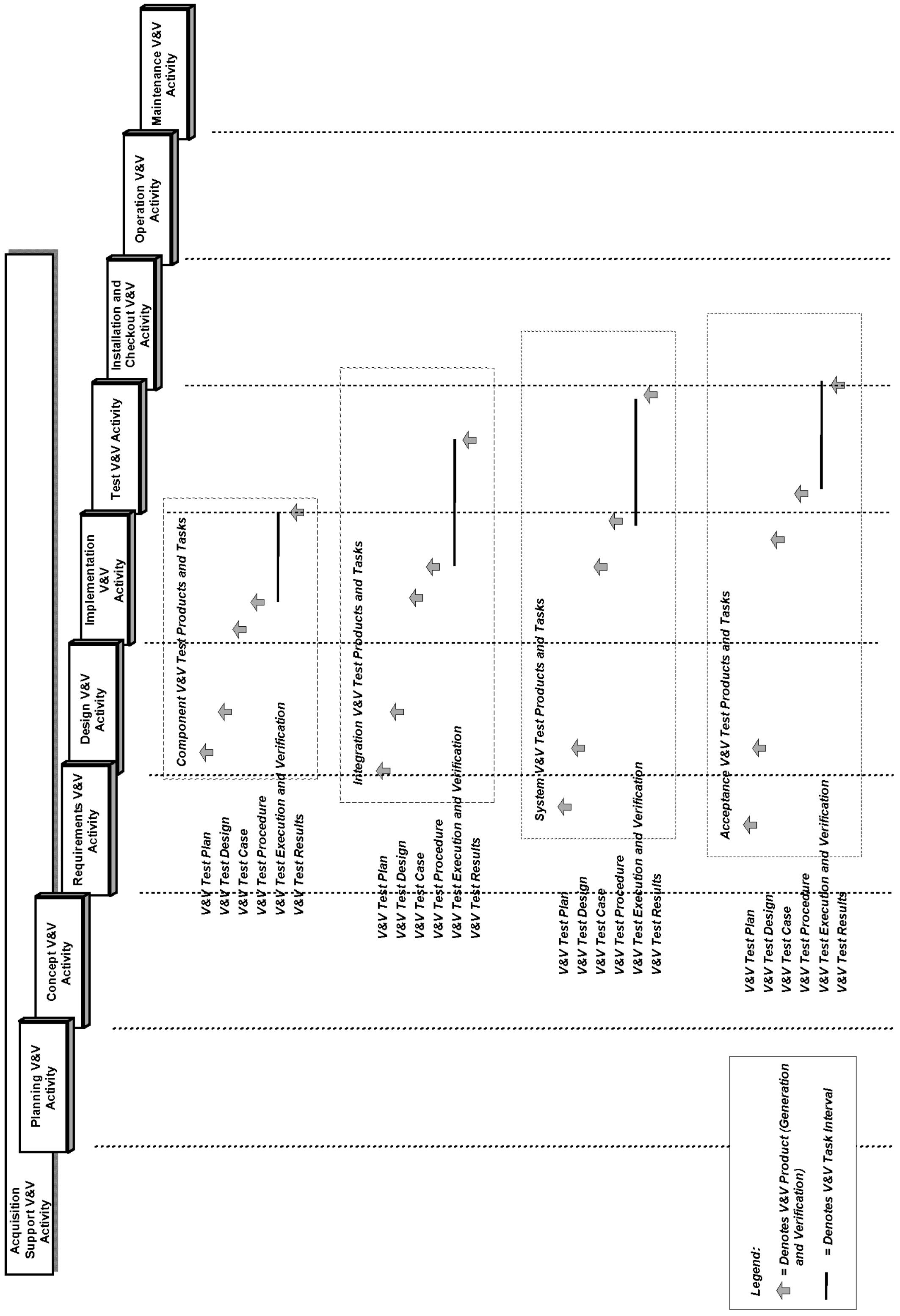

Acquisition Support V&V Activity
Planning V&V Activity
Concept V&V Activity
Requirements V&V Activity
Design V&V Activity
Implementation V&V Activity
Test V&V Activity
Installation and Checkout V&V Activity
Operation V&V Activity
Maintenance V&V Activity
Component V&V Test Products and Tasks
Integration V&V Test Products and Tasks
System V&V Test Products and Tasks
Acceptance V&V Test Products and Tasks
V&V Test Plan
V&V Test Design
V&V Test Case
V&V Test Procedure
V&V Test Execution and Verification
V&V Test Results
Legend:
= Denotes V&V Product (Generation and Verification)
= Denotes V&V Task Interval

Figure 2—An example of V&V test product and test execution task scheduling

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex A

(informative)

Mapping of IEEE Std 1012 V&V activities and tasks

A.1 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 12207:1995 [B13] 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 12207:1995 [B13] clause numbers and titles of V&V processes and activities. The second column of Table A.1 lists the IEEE Std 1012-2004 clauses and tables that address the topics listed in the first column.

Table A.1—Mapping ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause/table)
5.1.4.1 Supplier Monitoring V&V	**5.2.1 Activity: Acquisition support V&V** Task 1 Scoping the V&V effort Task 2 Planning the interface between the V&V effort and supplier Task 3 System requirements review
5.2.4.5(h) and 5.2.5.5 Interfacing with V&V[a]	**5.2.1 Activity: Acquisition support V&V** Task 2 Planning the interface between the V&V effort and supplier **5.3.1 Activity: Planning V&V** Task 1 Planning the interface between the V&V effort and supplier **Annex C Definition of IV&V**
5.2.6.3 Verification and Validation[a]	All clauses, tables, software V&V figures, and annexes
5.3.2 System Requirements Analysis	**5.2.1 Activity: Acquisition support V&V** Task 3 System requirements review **5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation Task 4 Traceability analysis
5.5.5 and 5.5.6 Migration and Software Retirement	**5.1.1 Activity: Management of V&V** Task 2 Proposed/baseline change assessment **5.6.1 Activity: Maintenance V&V**

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.1—Mapping ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks *(continued)*

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause/table)
6.4.1 Verification Process Implementation[a]	**Clause 4 Software integrity levels** **Clause 6 Software V&V reporting, administrative, and documentation requirements** **6.2 V&V Administrative requirements** **Clause 7 SVVP outline** **7.7 SVVP section 7: V&V administrative requirements**
6.4.1.1 Criticality of Software to be Verified[a]	**Clause 4 Software integrity levels** **Table B.1 Assignment of software integrity levels** **Table B.2 Definition of consequences** **Annex D V&V of reuse software**
6.4.1.2 Process for Verification[a]	**Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP outline**
6.4.1.3 and 6.4.1.4 Extent and Rigor of Verification[a]	**Clause 4 Software integrity levels** **Table 2 Minimum V&V tasks assigned to each software integrity level** **Annex C Definition of independent V&V (IV&V)**
6.4.1.5 Verification Plan[a]	**Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP outline**
6.4.1.6 Problem and Non-conformance Reports[a]	**6.2 V&V Administrative requirements** **7.7 SVVP section 7: V&V Administrative requirements**
6.4.2 Verification	**Clause 5 Software V&V processes**
6.4.2.1 Contract Verification	**5.3.1 Activity: Planning V&V** Task 2 Contract Verification
6.4.2.2 Process Verification	**5.2 Process: Acquisition** **5.3 Process: Supply** **5.4 Process: Development**

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.1—Mapping ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks *(continued)*

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause/table)
6.4.2.3 Requirements Verification	**5.2.1 Activity: Acquisition support V&V** Task 3 System requirements review **5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation **5.4.2 Activity: Requirement V&V** Task 1 Traceability analysis Task 2 Software requirements evaluation Task 3 Interface analysis Task 4 Criticality analysis Task 5 System V&V test plan generation Task 6 Acceptance V&V test plan generation Task 8 Hazard analysis Task 9 Security analysis
6.4.2.4 Design Verification	**5.4.3 Activity: Design V&V** Task 1 Traceability analysis Task 2 Software design evaluation Task 3 Interface analysis Task 4 Criticality analysis Task 5 Component V&V test plan generation Task 6 Integration V&V test plan generation Task 7 Component V&V test design generation Task 8 Integration V&V test design generation Task 9 System V&V test design generation Task 10 Acceptance V&V test design generation Task 11 Hazard analysis Task 12 Security analysis
6.4.2.5 Code Verification	**5.4.4 Activity: Implementation V&V** Task 1 Traceability analysis Task 2 Source code and source code documentation evaluation Task 3 Interface analysis Task 4 Criticality analysis Task 5 Component V&V test case generation Task 6 Integration V&V test case generation Task 7 System V&V test case generation Task 8 Acceptance V&V test case generation Task 9 Component V&V test procedure generation Task 10 Integration V&V test procedure generation Task 11 System V&V test procedure generation Task 12 Component V&V test execution Task 13 Hazard analysis Task 14 Security analysis
6.4.2.6 Integration Verification	**5.4.5 Activity: Test V&V** Task 3 Integration V&V test execution

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.1—Mapping ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks ***(continued)***

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause/table)
6.4.2.7 Documentation Verification	**5.2.1 Activity: Acquisition support V&V** Task 3 Systems requirements review **5.3.1 Activity: Planning V&V** Task 2 Contract verification **5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation **5.4.2 Activity: Requirements V&V** Task 2 Software requirements evaluation Task 3 Interface analysis **5.4.3 Activity: Design V&V** Task 2 Software design evaluation Task 3 Interface analysis **5.4.4 Activity: Implementation V&V** Task 2 Source code and source code documentation evaluation Task 3 Interface analysis **5.4.6 Activity: Installation and Checkout** Task 1 Installation configuration audit **5.5.1 Activity: Operation V&V** Task 2 Operating procedures evaluation
6.5.1 Validation Process Implementation[a]	**Clause 4 Software integrity levels** **Clause 6 Software V&V reporting, administrative, and documentation requirements** **6.2 V&V Administrative requirements** **Clause 7 SVVP outline** **7.7 SVVP section 7: V&V administrative requirements** **Annex C Definition of independent V&V (IV&V)** **Annex D V&V of reuse software** **Annex E V&V measures**
6.5.1.1 Criticality of Software to be Validated[a]	**Clause 4 Software integrity levels** **Table B.1 Assignment of software integrity levels** **Table B.2 Definition of consequences** **Annex D V&V of reuse software**
6.5.1.2 Process for Validation	**Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP Outline**

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.1—Mapping ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks *(continued)*

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause/table)
6.5.1.3 Extent and Rigor of Validation[a]	**Table 2 Minimum V&V tasks assigned to each software integrity level** **Annex C Definition of independent V&V (IV&V)**
6.5.1.4 Validation Plan[a]	**Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause** 7 **SVVP outline**
6.5.1.5 Problem and Non-conformance Reports[a]	**6.2 V&V Administrative requirements** **7.7 SVVP section 7: V&V administrative requirements**
6.5.2 Validation	**Clause 5 Software V&V processes**
6.5.2.1 Validate Test Preparation[a]	**5.4.2 Activity: Requirements V&V** Task 5 System V&V Test plan generation Task 6 Acceptance V&V test plan generation **5.4.3 Activity: Design V&V** Task 5 Component V&V test plan generation Task 6 Integration V&V test plan generation Task 7 Component V&V test design generation Task 8 Integration V&V test design generation Task 9 System V&V test design generation Task10 Acceptance V&V test design generation **5.4.4 Activity: Implementation V&V** Task 5 Component V&V test case generation Task 6 Integration V&V test case generation Task 7 System V&V test case generation Task 8 Acceptance V&V test case generation Task 9 Component V&V test procedure generation Task 10 Integration V&V test procedure generation Task 11 System V& test procedure generation **5.4.5 Activity: Test V&V** Task 2 Acceptance V&V test procedure generation
6.5.2.2 Validate Test Traceability[a]	**5.4.4 Activity: Implementation V&V** Task 12 Component V&V test execution **5.4.5 Activity: Test V&V** Task 3 Integration V&V test execution Task 4 System V&V test execution Task 5 Acceptance V&V test execution
6.5.2.3 Validate Test Conduct[a]	**5.4.4 Activity: Implementation V&V** Task 12 Component V&V test execution **5.4.5 Activity: Test V&V** Task 3 Integration V&V test execution Task 4 System V&V test execution Task 5 Acceptance V&V test execution

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.1—Mapping ISO/IEC 12207 V&V requirements to IEEE Std 1012 V&V activities and tasks *(continued)*

ISO/IEC 12207 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause/table)
6.5.2.4 Validate Software for Intended Use[a]	**5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation **5.4.2 Activity: Requirements V&V** Task 2 Software requirements evaluation Task 3 interface analysis **5.4.3 Activity: Design V&**V Task 2 Software design evaluation Task 3 Interface analysis **5.4.4 Activity: Implementation V&V** Task 2 Source code and source code documentation evaluation Task 3 Interface analysis **5.4.5 Activity: Test V&V** Task 4 System V&V test execution Task 5 Acceptance V&V test execution
6.5.2.5 Installation Test of Software[a]	**5.4.6 Activity: Installation and checkout V&V** Task 1 Installation configuration audit Task 2 Installation checkout Task 3 Hazard analysis Task 4 Security analysis Task 5 Risk analysis
Annex A Tailoring Process	N/A
Annex B Guidance on Tailoring	N/A
Annex C Guidance on Processes and Organizations	N/A
Annex D Bibliography	N/A
Annex E Basic Concepts of ISO/IEC 12207	N/A
Annex F Purpose and Outcomes	**Clause 4 Software integrity level** **Clause 5 Software V&V processes** **Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP outline**
Annex G	N/A
Annex H	N/A

[a]No ISO/IEC 12207:1995 [B13] clause title was listed. For purposes of this mapping, this standard assigned a clause title to reflect the clause contents.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

This standard defines 11 V&V activities, as shown in the first column of Table A.2, that are part of the V&V process. These 11 V&V activities correspond to the ISO/IEC 12207:1995 [B13] software life cycle processes and activities shown in columns 2 and 3 of Table A.2.

Table A.2—Mapping of IEEE Std 1012 V&V activities to ISO/IEC 12207

IEEE Std 1012 V&V activities (see corresponding subclause)	ISO/IEC 12207 software life cycle	
	Process[a]	Activity
5.2.1 Activity: Acquisition support V&V	Acquisition	—Initiation —RFP (tender) preparation —Contract preparation and update —Supplier monitoring —Acceptance and completion
5.3.1 Activity: Planning V&V	Supply	—Initiation —Preparation of response —Contract —Planning —Execution and control —Review and evaluation —Delivery and completion
5.4.1 Activity: Concept V&V	Development	—Process implementation —Requirements elicitation —System requirements analysis —System architectural design
5.4.2 Activity: Requirements V&V	Development	—Software requirements analysis
5.4.3 Activity: Design V&V	Development	—Software architectural design —Software detailed design
5.4.4 Activity: Implementation V&V	Development	—Software coding and testing
5.4.5 Activity: Test V&V	Development	—Software integration —Software qualification testing —System integration —System qualification testing
5.4.6 Activity: Installation and checkout V&V	Development	—Software installation —Software acceptance support
5.5.1 Activity: Operation V&V	Operation	—Process implementation —Operational testing —System operation —User support
5.6.1 Activity: Maintenance V&V	Maintenance	—Process implementation —Problem and modification analysis —Modification implementation —Maintenance review/acceptance —Migration —Software retirement
5.1.1 Activity: Management of V&V	All processes	All activities

[a]Annex F contains description, purposes, and outcomes of the ISO/IEC process model.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

A.3 Mapping of IEEE Std 1074 V&V requirements to IEEE Std 1012 V&V activities and tasks

Table A.3 shows a mapping of all IEEE Std 1074-1997 [B10] V&V requirements (i.e., processes, activities and tasks) to the V&V activities and tasks of this standard.

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

IEEE Std 1074 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
A.1 Project management activity groups A.1.1 Project initiation activities A.1.1.1 Create SLCP A.1.1.2 Perform estimations A.1.1.3 Allocate project resources A.1.1.4 Define metrics	**5.2.1 Activity: Acquisition support V&V** Task 1 Scoping the V&V effort Task 2 Planning the Interface between the V&V effort and supplier **5.3.1Activity: Planning V&V** Task 1 Planning the interface between the V&V effort and supplier **5.1.1 Activity: Management of V&V** Task 1 SVVP Generation **5.6.1 Activity: Maintenance V&V** Task 1 SVVP Revision **6.1 V&V reporting requirements** **7.6 SVVP section 6: V&V reporting requiremen**ts **6.2 Administrative V&V requirements** **7.7 SVVP section 7: V&V administrative requirements** **6.3.1 V&V Test documentation** **7.8 SVVP section 8: V&V test documentation requirements** **6.3.2 SVVP documentation** **Clause 7 SVVP outline** **Annex E V&V measures**
A.1.2 Project planning activities A.1.2.1 Plan evaluations A.1.2.2 Plan configuration management A.1.2.3 Plan system transition (if applicable) A.1.2.4 Plan installation A.1.2.5 Plan documentation A.1.2.6 Plan training A.1.2.7 Plan project management A.1.2.8 Plan integration	**Clause 5 Software V&V processes** All tasks **Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP outline**

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

IEEE Std 1074 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
A.1.3 Project monitoring and control activities A.1.3.1 manage risks 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	**5.1.1 Activity: Management of V&V** Task 1 SVVP generation Task 2 Proposed/baseline change assessment Task 3 Management review of the V&V effort Task 4 Management and technical review support Task 5 Interface with organizational and supporting processes
A.2 Pre-development activity groups A.2.1 Concept exploration activities A.2.1.1 Identify ideas or needs 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 A.2.2.1 Analyze functions A.2.2.2 Develop system architecture A.2.2.3 Decompose system requirements A.2.3 Software importation activities A.2.3.1 Identify imported software requirements A.2.3.2 Evaluate software import sources (if applicable) A.2.3.3 Define software import method (if applicable) A.2.3.4 Import software (if applicable)	**5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation Task 2 Criticality analysis Task 4 Traceability analysis Task 5 Hazard analysis Task 6 Security analysis Task 7 Risk analysis **5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation Task 3 Hardware/software/user requirements allocation analysis **Clause 5 Software V&V Processes** All tasks **Annex D V&V of reuse software**
A.3 Development activity groups 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	**5.4.2 Activity: Requirements V&V** Task 1 Traceability analysis Task 2 Software requirements evaluation Task 3 Interface analysis Task 4 Criticality analysis Task 5 System V&V test plan generation Task 6 Acceptance V&V test plan generation Task 7 Configuration management assessment Task 8 Hazard analysis Task 9 Security analysis Task 10 Risk analysis

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

IEEE Std 1074 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
A.3 Development activity groups *(continued)* A.3.2 Design activities A.3.2.1 Perform architectural design A.3.2.2 Design database (if applicable) A.3.2.3 Design interfaces A.3.2.4 Perform detailed design	**5.4.3 Activity: Design V&V** Task 1 Traceability analysis Task 2 Software design evaluation Task 3 Interface analysis Task 4 Criticality analysis Task 5 Component V&V test plan generation Task 6 Integration V&V test plan generation Task 7 Component V&V test design generation Task 8 Integration V&V test design generation Task 9 System V&V test design generation Task 10 Acceptance V&V test design generation Task 11 Hazard analysis Task 12 Security analysis Task 13 Risk analysis
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	**5.4.4 Activity: Implementation V&V** Task 1 Traceability analysis Task 2 Source code and source code documentation evaluation Task 3 Interface analysis Task 4 Criticality analysis Task 5 Component V&V test case generation Task 6 Integration V&V test case generation Task 7 System V&V test case generation Task 8 Acceptance V&V test case generation Task 9 Component V&V test procedure generation Task 10 Integration V&V test procedure generation Task 11 System V&V test procedure generation Task 12 Component V&V test execution Task 13 Hazard analysis Task 14 Security analysis Task 15 Risk analysis **5.4.5 Activity: Test V&V** Task 1 Traceability analysis Task 2 Acceptance V&V test procedure generation Task 3 Integration V&V test execution Task 4 System V&V test execution Task 5 Acceptance V&V test execution Task 6 Hazard analysis Task 7 Security analysis Task 8 Risk analysis

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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

IEEE Std 1074 V&V requirements	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
A.4 Post-development activity groups 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 A.4.2 Operation and support 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 A.4.3 Maintenance activities A.4.3.1 Identify software improvement needs A.4.3.2 Implement problem reporting method A.4.3.3 Reapply SLC A.4.4 Retirement activities A.4.4.1 Notify user A.4.4.2 Conduct parallel operations (if applicable) A.4.4.3 Retire system	**5.4.6 Activity: Installation and checkout V&V** Task 1 Installation configuration audit Task 2 Installation checkout Task 3 Hazard analysis Task 4 Security analysis Task 5 Risk analysis Task 6 V&V final report generation **5.5.1 Activity: Operation V&V** Task 1 Evaluation of new constraints Task 2 Operating procedures evaluation Task 3 Hazard analysis Task 4 Security analysis Task 5 Risk analysis **5.6.1 Activity: Maintenance V&V** Task 1 SVVP revision Task 2 Anomaly evaluation Task 3 Criticality analysis Task 4 Migration assessment Task 5 Retirement assessment Task 6 Hazard analysis Task 7 Security analysis Task 8 Risk analysis Task 9 Task iteration **5.6.1Activity: Maintenance V&V** Task 1 SVVP revision Task 2 Anomaly evaluation Task 3 Criticality analysis Task 4 Migration assessment Task 5 Retirement assessment Task 6 Hazard analysis Task 7 Security analysis Task 8 Risk analysis Task 9 Task iteration
A.5 Integral activity groups 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	**Clause 5 Software V&V processes** All tasks **Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP outline**
A.5 Integral activity groups 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	**Clause 5 Software V&V processes** All tasks **Clause 6 Software V&V reporting, administrative, and documentation requirements** **Clause 7 SVVP outline**

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

A.4 Mapping between CMMI and IEEE Std 1012 tasks

The following tables provide a mapping between the Software Engineering Process Groups CMMI process areas and the IEEE Std 1012 software V&V tasks.

Table A.4—IEEE Std 1012 CMMI process mapping matrix for requirements

CMMI process groups and areas	**IEEE Std 1012 V&V activities and tasks (see corresponding clause)**
Process management Organizational process focus	 **5.1.1 Activity: Management of the V&V effort** Task 3 Management review of the V&V effort
Organizational process definition	**5.1.1 Activity: Management of the V&V effort** Task 5 Interface with Organizational and Supporting Processes
Organizational training	N/A
Organizational process performance	**5.1.1 Activity: Management of the V&V effort** Task 3 Management review of the V&V effort
Organizational innovation & development	**5.1.1 Activity: Management of the V&V effort** Task 6 Identify improvement opportunities in the conduct of V&V
Project management Project planning	 **5.1.1 Activity: Management of the V&V effort** Task 1 SVVP Generation Task 2 Proposed/baseline change assessment **5.2.1 Activity: Acquisition support V&V** Task 1 Scoping the V&V Effort Task 2 Planning the interface between V&V effort and supplier Task 3 System requirements review **5.3.1 Activity: Planning V&V** Task 1 Planning the interface between V&V effort and supplier **5.4.1 Activity: Concept V&V** Task 3 Hardware/software/user requirements allocation analysis **5.4.2 Activity: Requirements V&V** Task 5 System V&V test plan generation Task 6 Acceptance V&V test plan generation **5.4.3 Activity: Design V&V** Task 5 Component V&V test plan generation Task 6 Integration V&V test plan generation Task 7 Component V&V test design generation Task 8 Integration V&V test design generation Task 9 System V&V test design generation Task 10 Acceptance V&V test design generation

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.4—IEEE Std 1012 CMMI process mapping matrix for requirements *(continued)*

CMMI process groups and areas	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
Project management Project planning *(continued)*	**5.4.4 Activity: Implementation V&V** Task 5 Component V&V test case generation Task 6 Integration V&V test case generation Task 7 System V&V test case generation Task 8 Acceptance V&V test case generation Task 9 Component V&V test procedure generation Task 10 Integration V&V test procedure generation Task 11 System V&V test procedure generation
Project monitoring and control	**5.5.1 Activity: Management of the V&V effort** Task 1 SVVP generation Task 2 Proposed/baseline change assessment Task 3 Management review of the V&V effort Task 6 Identify improvement opportunities in the conduct of V&V
Supplier agreement management	**5.2.1 Activity: Acquisition support V&V** Task 2 Planning the interface between V&V effort and supplier **5.3.1 Activity: Planning V&V** Task 1 Planning the interface between V&V effort and supplier
Integrated project management for IPPD	**Annex C Definition of independent V&V** **Annex F Relationship of V&V to other project responsibilities**
Risk management	**5.4.1 Activity: Concept V&V** Task 3 Hardware/software/user requirements allocation analysis Task 7 Risk analysis **5.4.2 Activity: Requirements V&V** Task 10 Risk analysis **5.4.3 Activity: Design V&V** Task 13 Risk analysis **5.4.4 Activity: Implementation V&V** Task 15 Risk analysis **5.4.5 Activity: Test V&V** Task 8 Risk analysis **5.6.1 Activity: Maintenance V&V** Task 8 Risk analysis
Integrated teaming	**5.1.1 Activity: Management of the V&V effort** Task 4 Management and technical review support Task 5 interface with organizational and supporting processes **Annex C Definition of independent V&V** **Annex F Relationship of V&V to other project responsibilities**
Integrated supplier management	**5.2.1 Activity: Acquisition support V&V** Task 2 Planning the interface between V&V effort and supplier **5.3.1 Activity: Planning V&V** Task 1 Planning the interface between V&V effort and supplier

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.4—IEEE Std 1012 CMMI process mapping matrix for requirements *(continued)*

CMMI process groups and areas	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
Quantitative project management	**5.1.1 Activity: Management of the V&V effort** Task 6 Identify improvement opportunities in the conduct of V&V **Annex E V&V measures**
Engineering	
Requirements development	**5.1.1 Activity: Management of the V&V effort** Task 2 Proposed/baseline change assessment **5.4.2 Activity: Requirements V&V** Task 2 Software Requirements Evaluation
Requirements management	**5.1.1 Activity: Management of the V&V effort** Task 4 Management and technical review support Task 5 Interface with organizational and supporting processes **5.4.1 Activity: Concept V&V** Task 3 Hardware/software/user requirements allocation analysis Task 4 Traceability analysis **5.4.2 Activity: Requirements V&V** Task 1 Traceability Analysis
Technical solution	**5.4.1 Activity: Concept V&V** Task 1 Concept documentation evaluation Task 3 Hardware/software/user requirements allocation analysis Task 5 Hazard analysis Task 6 Security analysis **5.4.2 Activity: Requirements V&V** Task 2 Software requirements evaluation Task 8 Hazard analysis Task 9 Security analysis **5.4.3 Activity: Design V&V** Task 2 Software design evaluation Task 3 Interface analysis Task 11 Hazard analysis Task 12 Security analysis **5.4.4 Activity: Implementation V&V** Task 2 Source code and source code documentation evaluation Task 3 Interface analysis Task 13 Hazard analysis Task 14 Security analysis

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.4—IEEE Std 1012 CMMI process mapping matrix for requirements ***(continued)***

CMMI process groups and areas	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
Product integration	**5.4.2 Activity: Requirements V&V** Task 1 Traceability analysis Task 3 Interface analysis Task 5 System V&V test plan generation Task 6 Acceptance V&V test plan generation **5.4.3 Activity: Design V&V** Task 1 Traceability analysis Task 3 Interface analysis Task 5 Component V&V test plan generation Task 6 Integration V&V test plan generation Task 7 Component V&V test design generation Task 8 Integration V&V test design generation Task 9 System V&V test design generation Task 10 Acceptance V&V test design generation **5.4.4 Activity: Implementation V&V** Task 1 Traceability analysis Task 3 Interface analysis Task 5 Component V&V test case generation Task 6 Integration V&V test case generation Task 7 System V&V test case generation Task 8 Acceptance V&V test case generation Task 9 Component V&V test procedure generation Task 10 Integration V&V test procedure generation Task 11 System V&V test procedure generation Task 12 Component V&V test execution **5.4.5 Activity: Test V&V** Task 1 Traceability analysis Task 2 Acceptance V&V test procedure generation Task 3 Integration V&V test execution Task 4 System V&V test execution Task 5 Acceptance V&V test execution **5.4.6 Activity: Installation and checkout V&V** Task 1 Installation configuration audit Task 2 Installation checkout
Verification	All clauses
Validation	All clauses

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.4—IEEE Std 1012 CMMI process mapping matrix for requirements *(continued)*

CMMI process groups and areas	**IEEE Std 1012 V&V activities and tasks (see corresponding clause)**
Support	
Configuration management	**5.1.1 Activity: Management of the V&V effort** Task 2 Proposed/baseline change assessment Task 6 Identify improvement opportunities in the conduct of V&V **5.4.1 Activity: Concept V&V** Task 4 Traceability Analysis **5.4.2 Activity: Requirements V&V** Task 1 Traceability analysis Task 7 Configuration management assessment **5.4.3 Activity: Design V&V** Task 1 Traceability analysis **5.4.4 Activity: Implementation V&V** Task 1 Traceability analysis **5.4.5 Activity: Test V&V** Task 1 Traceability analysis **5.4.6 Activity: Installation & Checkout V&V** Task 1 Installation configuration audit
Process & product quality assurance	**5.1.1 Activity: Management of the V&V effort** Task 6 Identify improvement opportunities in the conduct of V&V **Annex E V&V measures** **Annex F Relationship of V&V to other project responsibilities**
Organizational environment for integration	**Annex C Definition of independent V&V** **Annex F Relationship of V&V to other project responsibilities**

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table A.4—IEEE Std 1012 CMMI process mapping matrix for requirements ***(continued)***

CMMI process groups and areas	IEEE Std 1012 V&V activities and tasks (see corresponding clause)
Decision analysis & resolution	**5.1.1 Activity: Management of the V&V effort** Task 2 Proposed/baseline change assessment **5.4.1 Activity: Concept V&V** Task 7 Risk analysis **5.4.2 Activity: Requirements V&V** Task 10 Risk analysis **5.4.3 Activity: Design V&V** Task 13 Risk analysis **5.4.4 Activity: Implementation V&V** Task 15 Risk analysis **5.4.5 Activity: Test V&V** Task 8 Risk analysis **5.4.6 Activity: Installation and checkout V&V** Task 5 Risk analysis **5.5.1 Activity: Operation V&V** Task 5 Risk analysis **5.6.1 Activity: Maintenance V&V** Task 8 Risk analysis
Causal analysis & resolution	**5.1.1 Activity: Management of the V&V effort** Task 6 Identify improvement opportunities in the conduct of V&V **Annex E V&V measures** **Annex F Relationship of V&V to other project responsibilities**

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex B

(informative)

A risk-based software integrity level scheme

B.1 A risk-based software integrity level scheme

Table B.1 defines four software integrity levels used for reference purposes 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.

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.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table B.2—Definition 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 Table B.1 and Table 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—Graphic illustration of the assignment of software integrity levels

Error	Likelihood of occurrence of an operating state that contributes to the error (decreasing order of likelihood)			
Consequence	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

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex C

(informative)

Definition of independent V&V (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 should 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. Off-the-shelf tools that have extensive history of use do not require qualification testing. The most important aspect for the use of these tools is to verify the input data used.

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, financial) is vested in a V&V organization determines the degree of independence achieved.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Many forms of independence can be adopted for a V&V organization. The five most prevalent are: (1) classical, (2) modified, (3) integrated, (4) internal, and (5) embedded. Table C.1 illustrates the degree of independence achieved by these five forms.

Table C.1—Forms of IV&V

IV&V Form	Technical	Management	Financial
Classical	I	I	I
Modified	I	i	I
Integrated	i	I	I
Internal	i	i	i
Embedded	e	e	e
NOTE—I = Rigorous independence; i = Conditional independence; e = Minimal independence			

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 an 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 Integrated IV&V

This form is focused on providing rapid feedback of V&V results into the development process and is performed by an organization that is financially and managerially independent of the development organization to minimize compromises with respect to independence. The rapid feedback of V&V results into the development process is facilitated by the integrated IV&V organization: working side-by-side with the development organization; reviewing interim work products; and providing V&V feedback during inspections, walk-throughs, and reviews conducted by the development staff (potential impact on technical independence). Impacts to technical independence are counterbalanced by benefits associated with a focus on interdependence between the integrated IV&V organization and the development organization.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Interdependence means that the successes of the organizations are closely coupled, ensuring that they work together in a cooperative fashion.

C.4.4 Internal IV&V

Internal IV&V exists when the developer conducts the IV&V with personnel from within its own organization, although preferably not the same 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.4.5 Embedded V&V

This form is similar to internal IV&V in that it uses personnel from the development organization who should not be involved directly in the development effort. Embedded V&V is focused on ensuring conformance to the development procedures and processes. The embedded V&V organization works side-by-side with the development organization and attends the same inspections, walk-throughs, 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 V&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.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex D

(informative)

V&V of reuse software

D.1 Purpose

The purpose of this annex is to provide options and suggestions to help the V&V effort of reuse software and to overcome the particular challenges associated with reuse software. Reuse software can take many forms and could include software from software libraries, custom software developed for other applications, COTS software, software requirements, software designs, or other artifacts from existing software. This annex addresses both (1) reuse software developed and used as part of a reuse process, and (2) reuse software developed and used outside of a reuse process. Figure D.1 illustrates V&V activities and tasks for reuse software whether it was developed under a reuse process or outside of a reuse process.

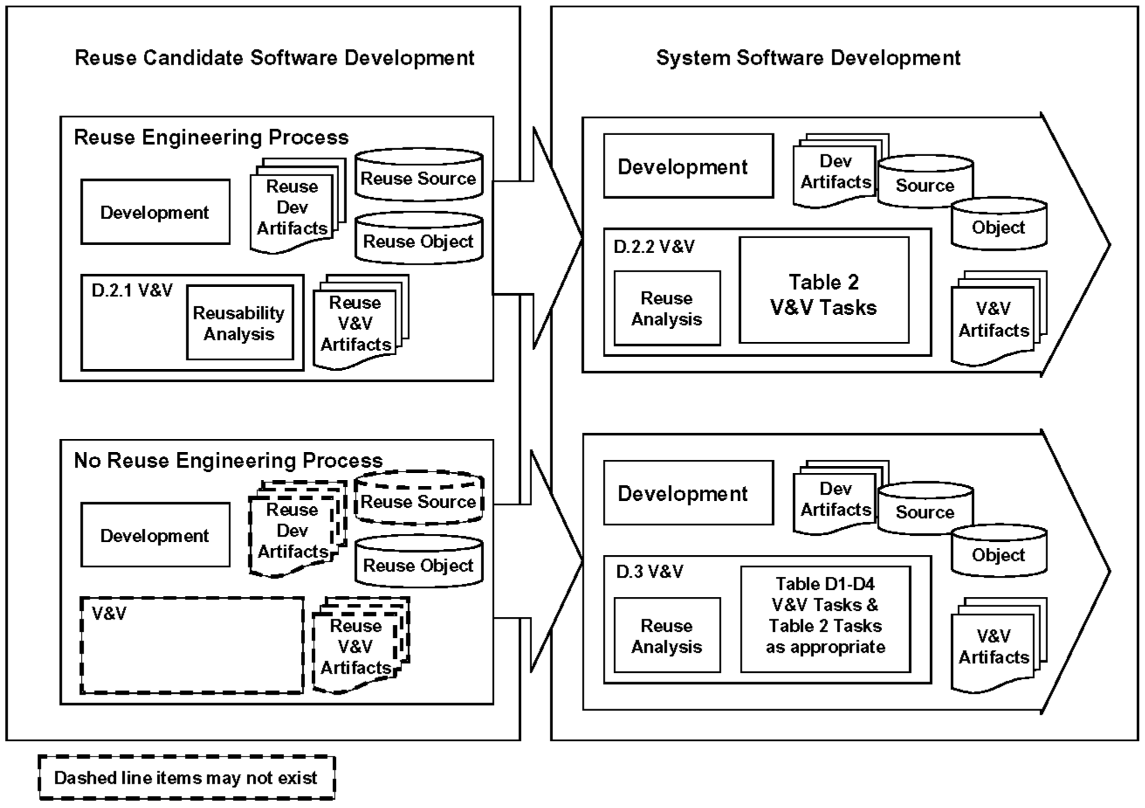


Figure D.1—V&V of reuse software

D.2 V&V of software developed in a reuse process

A structured software reuse process develops assets (e.g., design, code, and documentation) intended for use in multiple contexts. The software reuse processes of IEEE Std 1517-1999 [B11] provide a framework for extending the software life cycle processes of IEEE/EIA Std 12207.0-1996 [B12] to include a systematic, domain engineering process for software reuse. The domain scope and the domain analysis of an asset

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

provide requirements, intended use, interface parameters, and other information necessary for V&V of the asset, or an understanding of previously performed V&V of the asset.

D.2.1 V&V of assets in development

The V&V effort should analyze the artifacts (e.g., plans, models, architecture) of the domain engineering as part of the required V&V tasks. Significant analysis of the domain engineering products should occur during system requirements review, software requirements evaluation, interface analysis, software design evaluation, source code and source code documentation evaluation, and all test planning. The V&V effort must include the assignment of a software integrity level to the asset, in the context of the domain for its intended use, to determine the minimum V&V tasks to be performed (see Table 2). When planning the V&V effort, the optional task reusability analysis (see Annex G) should be included.

D.2.2 V&V of reused assets

A domain engineering process ensures that the information used in developing software systems is identified, captured, and organized so that it can be reused to create new systems within a domain. The V&V effort must include the assignment of a software integrity level to the asset, in the context of its actual use, to determine the minimum V&V tasks to be performed (see Table 2). When planning the V&V effort, the optional task reuse analysis (see Annex G) should be included.

D.3 V&V of software developed and reused outside of a reuse process

Some software systems are developed, operated, and maintained using software items that were not designed for use in multiple contexts or were not developed as part of a structured software reuse process (e.g., domain engineering products are not available). In these cases the V&V effort should perform the optional task reuse analysis (see Annex G) to produce inputs for determining the suitability of the reuse candidate software. The V&V effort must assign a software integrity level to the reuse candidate software to determine the minimum V&V tasks to be performed.

Reused software requires special consideration during the V&V effort when any one of the following is applicable:

— The inputs for a required V&V task are not available for the reused software.

— The reused software was developed as part of a system that is different in function or application from the system where it will be reused.

— The reused software was developed to meet different user needs from the current system.

— The original user needs are unknown.

In some cases, inputs for the V&V tasks may not be available, reducing visibility into the software products and processes. Options and techniques are available to compensate for the lack of inputs. Each technique has varying strengths and weaknesses—consideration should be given to performing multiple techniques to counter the weaknesses of one technique with strengths of another technique when high confidence is demanded. These options are addressed in decreasing order of desirability.

First, substitute tasks. Substitution for Table 1 V&V tasks is permitted if equivalent substitute V&V tasks can be shown to satisfy the same criteria as in Table 1. Two substitution task techniques are suggested in Table D.1.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table D.1—Substitution tasks to establish V&V task inputs

Substitution tasks **Description: Substitute alternative analysis and test methods in lieu of the IEEE Std 1012 requirement V&V tasks to generate objective conclusions about the correctness, completeness, accuracy, and usability of the reused software.**	
Technique 1: Black box testing *Black box testing and validation*: Execute the reused software against a spectrum of test case inputs and validate the correctness of the output. *User's manual analysis*: Derive system and software requirements from the user's manual and validate that the black box testing results satisfy the requirements. *Limit checks in interfacing software*: Add limit checks within all interface software on all data and logical information received from the reused software to ensure that no erroneous information is accepted.	Pros — Test results reflect actual target software — Limits catastrophic errors from propagating into interfacing systems — Ability to check the major system and user requirements derived from user's manual — Independent analysis
	Cons — Inability to detect all test errors if presence of error not observable in black box outputs (e.g., latent errors) — Limit checks difficult to cover all execution scenarios — Limited by the thoroughness of the user's manual
Technique 2: Review developer's QA *Developer's QA results*: Review developer's QA results and confirm the evidence of data similar to those that would be generated from V&V tasks. *Developer's test results*: Review the developer's test results and confirm the evidence of data similar to those that would be generated from V&V tasks. *Review of developer's notebook*: Review the developer's notebook to derive additional insights and problems with the software during early stages of development. *User's manual analysis*: Derive system and software requirements from the user's manual and validate that the developer's QA and test results satisfy the requirements.	Pros — Ability to derive insight into the details of the software design and internal performance characteristics — Identification of possible program error characteristics warranting further analysis and testing by other methods — Observations of program performance using test results reflecting actual software execution characteristics
	Cons — Limited by the scope and extent of the QA analysis and the specific focus of the testing performed — Limited by the thoroughness of the user's manual — Not a totally independent analysis

Second, use alternative sources of information to perform the V&V tasks in Table 1 and Table 2 of this standard. Three alternative source techniques are suggested in Table D.2.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table D.2—Alternate sources to establish V&V task inputs

Alternative sources **Description: Use alternative sources of program data to derive conclusions about the correctness, completeness, accuracy, and usability of the reused software.**	
Technique 3: Operational history *Historical Data Analysis*: Examine and analyze the operational history of the reused software with particular attention to how the software performed in a system with similar characteristics to the new system being proposed. *User Interviews*: Conduct interviews with operational users. Focus data gathering on how the system performed in scenarios and conditions similar to those expected in the new system being proposed.	Pros — Real data of the reused software in an operational environment — User observations about the performance of the software and its related system — Software burn-in established and track record of discrepancies recorded — Independent analysis
	Cons — Different characteristics, technologies, and user interfaces with new proposed system could cause error not observed in historical system — Not all interactions recordable or observable in historical systems so data completeness and accuracy are limited — User observations can be subjective, biased, and error-prone, so correctness, completeness, and accuracy may be limited
Technique 4: Audit results *Developer's interview*: Conduct interviews with development team to extract pertinent information about the design and performance characteristics of the reused software. *Design walk-through review analysis*: Analyze the design and code walkthrough data to determine how the reused software would interact in the new proposed system. *Standard compliance analysis*: Review the results of any standards compliance audits to determine that the proper software standards were followed in the construction of the reused software.	Pros — Depending on the thoroughness of the development team's records, good insight into the design and code approaches can be obtained from the interviews — Historical artifacts of design and code walkthroughs may be of sufficient quality to act as a substitute of the actual source code and design details
	Cons — Limited by the thoroughness of the development team's documentation and recollection during interviews — Not a totally independent analysis

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table D.2—Alternate sources to establish V&V task inputs *(continued)*

Alternative sources **Description: Use alternative sources of program data to derive conclusions about the correctness, completeness, accuracy, and usability of the reused software.**	
Technique 5: Artifacts *Product documentation analysis*: Review any product documentation to derive artifacts similar to requirements, design, and code [if possible—for example, if a pseudo design language (PDL) is used] *Prior V&V results analysis*: Analyze any prior V&V results and develop inferences and extrapolation of the data to the new proposed system.	Pros — Use actual artifacts representing some form of the reused software — Prior V&V results for an initial basis for formulating additional analysis and testing to conduct to fill in the analysis and testing voids caused by lack of program documentation — Independent analysis
	Cons — Overstating a conclusion without having a solid basis for knowing the system conditions could lead to faulty conclusions about the suitability in the new proposed system and its different system conditions. — Limited by the thoroughness of the product documentation

Third, use reverse engineering to generate inputs to perform the V&V tasks in Table 1 and Table 2 of this standard. One reverse engineering technique is suggested in Table D.3.

Table D.3—Reverse engineering to establish V&V task inputs

Reverse engineering **Description: Reverse engineer requirements, design, and code data to generate objective conclusions about the correctness, completeness, accuracy, and usability of the reused software.**	
Technique 6: Reverse compilation *Reverse engineer source code*: Reverse compile "pseudo source" code from the program object file. Analyze the reverse compiled pseudo code using normal V&V procedures including all the V&V test strategies and methods. *Reverse engineer requirements*: Derive the system and software requirements from the user's manual. Analyze the requirements using the IEEE Std 1012 V&V tasks and test the reused software against these reverse engineered requirements.	Pros — Uses the actual code—no hidden or implied data. — Perform all of the IEEE Std 1012 V&V tasks — Test data reflects actual performance of the reused software under the system conditions of the new proposed system — Independent analysis
	Cons — Time consuming to reverse engineer data — Pseudo code hard to read

Fourth, use independent prototyping and comparison of performance results with those of the reuse asset to perform the V&V tasks in Table 1 and Table 2 of this standard. Two independent prototyping and comparison techniques are suggested in Table D.4.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Table D.4—Independent prototyping and comparison to establish V&V task inputs

Independent prototyping and comparison **Description: Develop a model (prototype) of the proposed software or use portions of the prior system. Execute test scenarios on the prototype or prior system and compare the test results against the reused software. Analyze the results to generate objective conclusions about the correctness, completeness, accuracy, and usability of the reused software.**	
Technique 7: Prototyping *Comparison of prototype code*: Develop a replication model (prototype) of the function or requirements in a user-friendly language. Execute test cases representing the range of system scenarios on both the model and reuse software. Compare the results and analyze the differences to determine whether the reused software is performing as intended.	Pros — Useful for small functions and sets of requirements — Easy to diagnose problems in the reused software — Ability to run a wide range of system scenarios and compare against a benchmark program — Independent analysis
	Cons — Cost and time of building the model — Errors in the model can mask similar errors in the reused software (likelihood of two similar errors generated by two independent sources should be small or unlikely)
Technique 8: Prior system results *Comparison with prior system/function*: Execute test cases representing the range of system scenarios on the prior system/function and the reused software. Compare the results and analyze the differences to determine whether the reused software is performing as intended.	Pros — Inexpensive to execute test cases on prior system and compare. — Proven track record of performance of the prior system/function establishes a performance baseline — Differences in execution results leads to other analysis and testing to be conducted — Independent analysis
	Cons — Limited in scope to smaller functions — Ability to instrument prior system/function to extract diagnosis data about interim program steps may be difficult and make it harder to diagnose exact location of a problem — Any problems hidden in the prior system/function may go undetected or untested in the new proposed system/function (inheritance of errors)

Lastly, use a combination of the following circumstantial evidence that provides visibility and insight into the reused software to perform the V&V tasks in Table 1 and Table 2 of this standard:

a) Operational history

b) Test history

c) Audit results

d) User interviews

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

e) Engineering judgment
f) Product documentation
g) Prior hazard analysis results
h) Prior V&V results
i) Software developer’s notebook
j) Design process documentation
k) Original developers’ interviews
l) Static code analysis results
m) Standards compliance assessments

If V&V of reused software cannot be accomplished at the appropriate level, the items may be used so long as the risk associated with this use is recognized and accounted for in the risk mitigation strategy. The V&V effort should ensure that the risks are thoroughly understood, properly documented, and properly tracked under the risk analysis tasks.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex E

(informative)

V&V measures

E.1 Introduction

The management of V&V activity uses measures to provide feedback for the continuous improvement of the V&V process and to evaluate the software development processes and products. Trends can be identified and addressed by computing evaluation measures over a period of time. Threshold values of measures should be established and trends should be evaluated to serve as indicators as to whether a process, product, or V&V task has been satisfactorily accomplished. No standard set of measures is applicable for all projects, so the use of measures may vary according to the application domain and software development environment.

No consensus exists on measures for evaluating the quality and coverage of the V&V tasks. IEEE Std 1061™-1998 [B9] provides a standard definition of available software quality measures. Other measure-related standards and guides are IEEE Std 982.1-1988 [B5] and FP-05 Software Measurement [B1].

This standard considers three categories of measures associated with the V&V effort: (1) measures for evaluating anomaly density, (2) measures for assessing V&V effectiveness, and (3) measures for evaluating V&V efficiency.

E.2 Measures for evaluating anomaly density

Anomaly density measures can provide insightful information on the software product quality, the quality of the software development processes, and the quality of the V&V effort to discover anomalies in the system/ software and to facilitate correction of the anomalies. Anomaly density measures are influenced by numerous variables (e.g., software complexity, type of domain, and time-phase application of the V&V processes); consequently, the measures must be analyzed to gain insight into the interdependencies between the development efforts and the V&V efforts.

If the V&V anomaly density measure value is low, this suggests that the program development quality is high, that the V&V processes need to be improved, or a combination of both. If the measure value is high, then this suggests that the program development quality is low, that the V&V processes are effective, or a combination of both. Regardless of the measure value, the next step is to evaluate related software program development measures to further clarify and discern the measure trends to determine the need for process improvements.

Anomaly measures and trends can be used to improve the quality of the current project and can be used to improve the planning and execution of V&V processes for future projects with similar characteristics. The measures defined by Equation (E.1), Equation (E.2), Equation (E.3), and Equation (E.4) are applicable for four software development phases:

$$\textit{Requirements anomaly density} = \frac{\textit{\# Requirements anomalies found by V\&V effort}}{\textit{\# Requirements reviewed by V\&V effort}} \tag{E.1}$$

$$\textit{Design anomaly density} = \frac{\textit{\# Design statement anomalies found by V\&V effort}}{\textit{\# Design statements reviewed by V\&V effort}} \tag{E.2}$$

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

$$Code\ anomaly\ density\ =\ \frac{\#\ Code\ anomalies\ found\ by\ V\&V\ effort}{\#\ Code\ volume\ reviewed\ by\ V\&V\ effort} \quad \text{(E.3)}$$

$$Test\ anomaly\ density\ =\ \frac{\#\ Test\ anomalies\ found\ by\ V\&V\ effort}{\#\ Tests\ reviewed\ by\ V\&V\ effort} \quad \text{(E.4)}$$

E.3 Measures for evaluating V&V effectiveness

Measures associated with V&V effort effectiveness provide quantitative indications that characterize the added benefits of V&V to discover anomalies in software products and processes. These measures delineate the percentage of the total anomalies found by the V&V effort. The measures are influenced by numerous variables (e.g., software complexity), and the measures must be analyzed to gain insight into the interdependencies between the development efforts and the V&V efforts.

The V&V effectiveness measure values are highly influenced by the degree of parallelism between the software development effort and the V&V effort. Assuming that the efforts are parallel, then a low V&V effectiveness measure value suggests that the software development effort is effective, or that the V&V effort may require improvement, or a combination of both. If the V&V effectiveness measure value is high, then this suggests that the software development processes may require improvement, or that the V&V processes are effective, or that only incremental changes to the V&V processes may be required. Regardless of the measure value, the next step is to evaluate related software program development measures to further clarify and discern the measure trends to determine the need for process improvements. The measures defined by Equation (E.5), Equation (E.6), Equation (E.7), and Equation (E.8) are applicable for four software development phases:

$$Requirements\ V\&V\ effectiveness\ =\ \frac{\#\ Requirement\ anomalies\ found\ by\ V\&V\ effort}{\#\ Requirements\ anomalies\ found\ by\ all\ sources} \quad \text{(E.5)}$$

$$Design\ V\&V\ effectiveness\ =\ \frac{\#\ Design\ statement\ anomalies\ found\ by\ V\&V\ effort}{\#\ Design\ statement\ anomalies\ found\ by\ all\ sources} \quad \text{(E.6)}$$

$$Code\ V\&V\ effectiveness\ =\ \frac{\#\ Code\ anomalies\ found\ by\ V\&V\ effort}{\#\ Code\ anomalies\ found\ by\ all\ sources} \quad \text{(E.7)}$$

$$Test\ execution\ V\&V\ effectiveness\ =\ \frac{\#\ Test\ anomalies\ found\ by\ V\&V\ effort}{\#\ Test\ anomalies\ found\ by\ all\ sources} \quad \text{(E.8)}$$

E.4 Measures for evaluating V&V efficiency

Measures associated with V&V effort efficiency provide data that characterize the capability of the V&V effort to discover anomalies in software products and processes in the development activity in which they are injected. Maximum benefits are realized when software anomalies are discovered as early as possible in the development life cycle, thereby minimizing rework and development costs. Analysis of these measures, the anomalies, and the causal factors that prevented discovery of the anomaly in the phase in which it was injected can reveal needed improvements in methods, processes, tools, and skills to improve the overall V&V effort.

A low V&V efficiency measure value suggests that the software V&V effort is not discovering anomalies in the earliest possible activity, or that the software development products are immature, or a combination of both. If the V&V efficiency measure value is high, then this suggests that the V&V effort is discovering anomalies in the earliest possible activity, or that the software development products are mature, or a

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

combination of both. Regardless of the measure value, the next step is to evaluate related software program development measures to further clarify and discern the measure trends to determine the need for process improvements. The measures defined by Equation (E.9), Equation (E.10), Equation (E.11), and Equation (E.12) are applicable for four software development phases:

$$\mathit{Requirements\,V\&V\ efficiency} = \frac{\begin{array}{c}\#\ \mathit{Requirements\ anomalies\ found\ by\ V\&V}\\ \mathit{in\ requirements\ activity}\end{array}}{\#\ \mathit{Requirements\ anomalies\ found\ by\ V\&V\ in\ all\ activities}} \times 100\% \tag{E.9}$$

$$\mathit{Design\ V\&V\ efficiency} = \frac{\#\ \mathit{Design\ statement\ anomalies\ found\ by\ V\&V\ in\ design\ activity}}{\#\ \mathit{Design\ statement\ anomalies\ found\ by\ V\&V\ in\ all\ activities}} \times 100\% \tag{E.10}$$

$$\mathit{V\&V\ efficiency} = \frac{\#\ \mathit{Code\ anomalies\ found\ by\ V\&V\ in\ implementation\ activity}}{\#\ \mathit{Code\ \ anomalies\ found\ by\ V\&V\ in\ all\ activities}} \times 100\% \tag{E.11}$$

$$\mathit{V\&V\ efficiency} = \frac{\#\ \mathit{Test\ statement\ anomalies\ found\ by\ V\&V\ in\ test\ activity}}{\#\ \mathit{Test\ anomalies\ found\ by\ V\&V\ in\ all\ activities}} \times 100\% \tag{E.12}$$

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex F

(informative)

Example of V&V organizational relationship to other project responsibilities

F.1 V&V organizational relationships

Figure F.1 provides an example of organizational relationships among the V&V team, the acquirer and the supplier, and identifies the information and data flows throughout the V&V effort. Many other organizational relationships will work well as long as project responsibilities, data flows, and reporting flows are defined and documented.

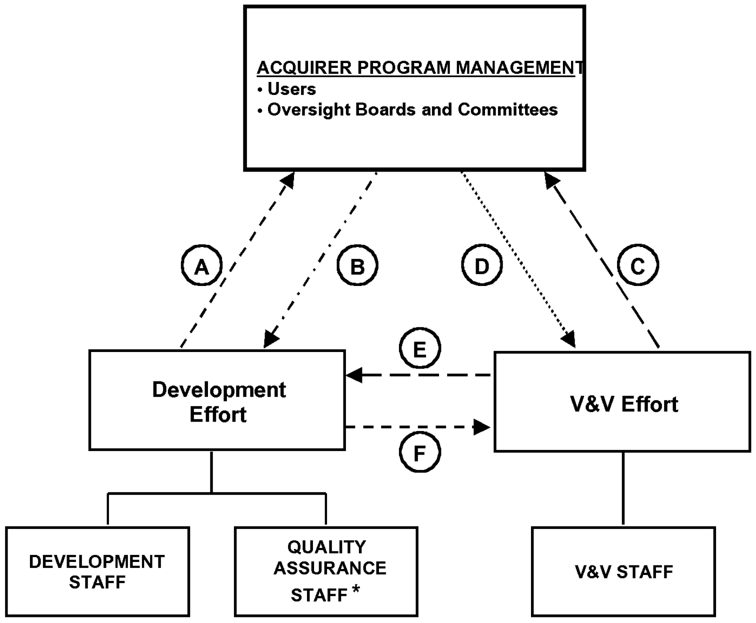

ACQUIRER PROGRAM MANAGEMENT
• Users
• Oversight Boards and Committees
A
B
C
D
E
F
Development Effort
V&V Effort
DEVELOPMENT STAFF
QUALITY ASSURANCE STAFF *
V&V STAFF

Figure F.1—Relationship of V&V to other project responsibilities

NOTE—The lines in Figure F.1 represent the flow of control and data as follows:

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

B: Approval, denial, and recommendations on development issues and deliverables listed in A.

C: Submittal of SVVP, V&V task results, anomaly reports, activity reports, and other special reports.

D: Approval, denial, and recommendations on V&V issues and deliverables listed in C.

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

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

*: The quality assurance staff may report directly to the Quality Assurance Director rather than through the development organization.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex G

(informative)

Optional V&V tasks

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, and nonlinear 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, and 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 among 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. Evaluation of database design as part of a design review process could include

a) 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.
b) 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.
c) 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.
d) Backup and disaster recovery analysis. Review the methods employed for backup against the requirements for data recovery and system disaster recovery and identify deficiencies.

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Data flow analysis. Evaluation of data flow diagrams as part of a design review process could include

a) 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.
b) Flow balancing. Compare the output data from each process 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.
c) 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.
d) 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:

a) Identification of the disaster recovery team and a contact list
b) Recovery operation procedures
c) Procedure for establishing an alternative site including voice and data communications, mail, and support equipment
d) Plans for replacement of computer equipment
e) Establishment of a system backup schedule
f) Procedures for storage and retrieval of software, data, documentation, and vital records off-site
g) Logistics of moving staff, data, documentation, etc.

Distributed architecture assessment. Assess the distribution of data and processes in the proposed architecture for feasibility, timing conflicts, 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 products 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 the following:

a) Inspection planning
b) Product overview
c) Inspection preparation
d) Examination meeting
e) Defect rework
f) 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

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

group, quality assurance, or V&V. The participants assume specific roles 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). Validate 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). Validate that the requirements meet customer needs and can be implemented. Verify that they 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 and resources 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

a) At a suitable point of correctness for mass production of that software
b) 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:

a) Database transaction rates to determine the need to reorganize or re-index the database
b) CPU performance monitoring for load balancing
c) Direct access storage utilization
d) Network traffic to ensure adequate bandwidth
e) Critical outputs of a system (e.g., scheduled frequency, expected range of values, scheduled system reports, reports of events)

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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 predetermined 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 measures.

Proposal evaluation support. Participate in the development organization source selection process. Develop proposal evaluation factors and assessment criteria. Independently evaluate development organization proposals to assess conformance to the statement of work and performance requirements.

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 analysis. Verify that the artifacts (products) of the domain engineering process conform to project-defined purpose, format, and content (e.g., IEEE Std 1517-1999 [B11]). Verify that the domain models and domain architecture are correct, consistent, complete, accurate, and conform to the domain engineering plan. Analyze the asset (software item intended for reuse) to verify that the asset is consistent with the domain model and domain architecture.

Reuse analysis. Analyze the developer's documentation to verify that the original domain of the candidate reuse software will satisfy the domain of the new system (e.g. software integrity level, user needs, operating environment, safety, security, and interfaces). If the developer has performed no domain analysis, perform domain analysis (see IEEE Std 1517-1999 [B11]) to compare the original domain and the new domain of the candidate reuse software. Verify that developer reuse planning dispositions and document all domain differences.

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 walk-through 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

a) CPU load

b) Random access memory and secondary storage (e.g. disk, tape) utilization

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

c) Network speed and capacity
d) 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, database 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 baseline 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 “work around” 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 analysis and 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.

Usability analysis. Verify that stakeholder needs and interests are considered during development, operation, and maintenance process activities. The analysis will ensure that: human-centered design activities are performed; human factors and ergonomics considerations are incorporated into the design, potential adverse effects on human health and safety are addressed in the design; and user needs are satisfied in a manner that supports user effectiveness and efficiency.

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 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 and associated tools, outputs generated, need date, and cost of tool purchase or development. The tool plan should also describe test

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

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.

Walk-through. 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 walk-through, design walk-through, source code walk-through, and test walk-through.

Walk-through (design). Participate in a walk-through of the design and updates of the design to ensure completeness, correctness, technical integrity, and quality.

Walk-through (requirements). Participate in a walk-through 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.

Walk-through (source code). Participate in a walk-through of the source code to ensure that the code is complete, correct, maintainable, free from logic errors, conforms to coding standards and conventions, and will operate efficiently.

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

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.

Annex H

(informative)

Bibliography

[B1] FP-05 Software Measurement, IEEE Computer Society Software and Systems Engineering Standards Committee Policy.[4]

[B2] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms,* Seventh Edition.[5,6]

[B3] IEEE Std 610.12-1990 (Reaff 2002), IEEE Standard Glossary of Software Engineering Terminology.

[B4] IEEE Std 829-1998, IEEE Standard for Software Test Documentation.

[B5] IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.

[B6] IEEE Std 1012A™-1998, Supplement to IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.

[B7] IEEE Std 1028-1997, IEEE Standard for Software Reviews.

[B8] IEEE Std 1044-1993, IEEE Standard for Software Anomalies.

[B9] IEEE Std 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.

[B10] IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.

[B11] IEEE Std 1517-1999, IEEE Standard for Information Technology—Software Life Cycle Processes—Reuse Processes.

[B12] IEEE/EIA Std 12207.0-1996, IEEE/EIA Standard—Industry Implementation of International Standard ISO/IEC 12207:1995 (ISO/IEC 12207) Standard for Information Technology—Software Life Cycle Processes.

[B13] ISO/IEC 12207:1995, Information Technology—Software Life Cycle Processes; as amended by Amendment 1:2002.[7]

[4]This publication can be found at http://standards.computer.org/s2esc/s2esc_pols/FP-05_Software_Measurement.htm.

[5]IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[6]The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[7]ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

Copyright © 2005 IEEE. All rights reserved.

Authorized licensed use limited to: Dept of Homeland Security. Downloaded on December 20,2010 at 13:39:05 UTC from IEEE Xplore. Restrictions apply.