

IEEE Standard for Software Verification and Validation Plans

IEEE Standards Board

Approved November 14, 1986

Reaffirmed September 17, 1992

American National Standards Institute

Approved February 10, 1987

Sponsor

**Software Engineering Technical Committee of the
of the
IEEE Computer Society**

© Copyright 1986 by **The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street,
New York, NY 10017, USA**

*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.*

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

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

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

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

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

Secretary
IEEE Standards Board
345 East 47th Street
New York, NY 10017 USA

Foreword

(This Foreword is not a part of IEEE Std 1012-1986, IEEE Standard for Software Verification and Validation Plans.)

This standard provides uniform and minimum requirements for the format and content of Software Verification and Validation Plans (SVVPs). Performing software verification and validation (V&V) as defined in this standard provides for a comprehensive evaluation throughout each phase of the software project to help ensure that:

- 1) Errors are detected and corrected as early as possible in the software life cycle
- 2) Project risk, cost, and schedule effects are lessened
- 3) Software quality and reliability are enhanced
- 4) Management visibility into the software process is improved
- 5) Proposed changes and their consequences can be quickly assessed

This standard applies to both critical and noncritical software.

- 1) For critical software, this standard:
 - a) Requires that minimum V&V tasks, inputs, and outputs specified in this standard be included in SVVPs
 - b) Permits the SVVP to be extended by selecting additional V&V tasks from the optional tasks described in this standard or new tasks identified by the V&V planner
- 2) For noncritical software, this standard:
 - a) Recommends the use of minimum V&V tasks
 - b) Permits the SVVP to be tailored to V&V efforts by selecting any of the V&V tasks (minimum, optional, new)

This standard applies to all phases of the software life cycle from the Concept Phase to the Operation and Maintenance Phase. Maximum benefits are derived when V&V is started early in the software life cycle, preferably at project initiation during the Concept Phase. Benefits can be derived for software already in development or in the Operation and Maintenance Phase if the V&V requirements from this standard are invoked consistent with cost and schedule constraints. When V&V is invoked for software in development or in operation and maintenance, required V&V inputs may not exist. Under these conditions, this standard permits the V&V tasks to be tailored to adjust for missing V&V inputs. In some instances, this may require the generation of appropriate software documentation.

V&V is performed in parallel with software development. Each V&V life-cycle phase ends when the V&V tasks of that phase are completed and the software development products are determined to be adequate. V&V life-cycle phases may overlap as activities of the new life-cycle phase are beginning and activities of the previous life-cycle phase are completing.

V&V tasks are iterative: as changes are made to the software product, selected V&V tasks from the previous life-cycle phases are reperformed, or additional V&V tasks are performed to address the changes. V&V tasks are reperformed if errors are discovered in the V&V inputs or outputs. The complexity and scope of changes determine the level of detail covered by the iteration. The SVVP identifies the criteria for performing the iterative V&V tasks.

This standard defines a V&V reporting structure by identifying format and content of the Software Verification and Validation Report (SVVR). The standard for Software Quality Assurance Plans (SQAP, ANSI/IEEE Std-730-1984) requires the SVVR to include both V&V and other quality assurance results. The SVVR defined here is flexible enough to include both types of results. The interim phase reports, final summary

report, and optional SQAP-related activity reports defined by the SVVP provide visibility into the development and V&V processes.

This standard considers both the software and its system or operating environment. It can be used where software is the system or where software is part of a larger system. V&V should have a total system scope (that is, including interfaces between software, hardware, and operators) during the product life cycle. Embedded software is strongly coupled to hardware and other subsystems, and requires a system-level SVVP.

This standard was written to provide direction to organizations responsible for preparing or assessing a Software Verification and Validation Plan. This standard may be used by project management software developers, quality assurance organizations, purchasers, end users, maintainers, and verification and validation organizations. If V&V is performed by an independent group, then the SVVP should specify the criteria for maintaining the independence of the V&V effort from the software development and maintenance efforts.

Suggestions for the improvement of this standard will be welcomed. They should be sent to

Secretary
IEEE Standards Board
Institute of Electrical and Electronics Engineers, Inc
345 East 47th Street
New York, New York 10017

Introduction

The working group that developed this standard consisted of the following members:

Roger U. Fujii, <i>Chair</i>		
Doug McMann, <i>Vice Chair</i>		
Dolores R. Wallace, <i>Secretary</i>		
Julian O. Blossiu	Michael Edwards	David Schultz
Martha Branstad	John Horch	David M. Siefert
Fletcher J. Buckley	Ralph A. Kubek	Hugh Spillane
Francois Coallier	Joyce Lewis	David Turner
James A. Darling	Dennis E. Nickle	William S. Turner
Taz Daughtrey	Larry E. Nitszche	Adam Valentine
David C. Doty	A.E. Nountor	Jay W. Wiley
Sam Dugdale	Don J. Robbins	Andrea Williams
William Dupras	Hans Schaefer	Laurence Wilson

When the IEEE Standards Board approved this standard on September 18, 1986, it had the following membership:

John E. May, <i>Chair</i>		
Irving Kolodny, <i>Vice Chair</i>		
Sava I. Sherr, <i>Secretary</i>		
James H. Beall	Joseph L. Koepfinger*	Martha Sloan
Fletcher J. Buckley	Edward Lohse	Oley Wanaselja
Paul G. Cummings	Lawrence V. McCall	J. Richard Weger
Donald C. Fleckenstein	Donald T. Michael*	William B. Wilkens
Jay Forster	Marco W. Migliaro	Helen M. Wood
Daniel L. Goldberg	Stanley Owens	Charles J. Wylie
Kenneth D. Hendrix	John P. Riganati	Donald W. Zipse
Irvin N. Howell	Frank L. Rose	
Jack Kinn	Robert E. Rountree	

*Member emeritus

The standard was approved by the Software Engineering Standards Subcommittee of the IEEE Computer Society. At the time it approved this standard, the Ballot Group had the following membership:

John W. Horch, <i>Chair</i>		
A. Frank Ackerman	Homer C. Carney	William P. Dupras
Jagdish Agrawal	C.L. Carpenter, Jr	Robert E. Dwyer
Tom. Armbruster	Ronald R. Carter	Mary Eads
Richard L. Aurbach	R.L. Chilavsky	John D. Earls
James Baldo, Jr	Tsun S. Chow	Michael Edwards
H. Jack Barnard	Jung K. Chung	L.G. Egan
Roy W.. Bass	Peter Coad, Jr	Wolfgang Ehrenberger
Leo Beltracchi	Francois Coallier	Steven R. Eisen
Yechiel Ben-Naftau	Sharon R. Cobb	Caroline L. Evans
H.R. Berlack	Christopher M. Cooke	David W. Favor
J. Emmett Black	Gail A. Cordes	John Fendrich
Michael A. Blackledge	A.J. Cote	Robert G. Ferreol
Ronald M. Blair	Patricia W. Daggett	Glenn S. Fields
Walter DuBlanca	James A. Darling	Gordon Force
Kevin W. Bowyer	George D. Darling	Julian Forster
Ingar Brauti	Taz Daughtrey	C.R. Frederick
Michael F. Brennter	P.A. Denny	Carl Friedlander
Kathleen L. Briggs	James H. Dobbins	Richard C. Fries
William L. Bryan	David C. Doty	Ismael Fuentes
Fletcher J.. Buckley	Einar Dragstedt	Roger U. Fujii
Douglas. Burt	Robert Dunn	Michel Galinier

Leonard B. Gardner	Paulo C. Marcondes	Victor Shtern
David Gelperin	Stuart Marcotte	David M. Siefert
J. Kaye Grau	Philip C. Marriott	David J. Simkins
Andres Grebene	Nicholas L. Marselos	Jacob Slonim
Thomas Griest	Roger J. Martin	Jean-Christopher Slucki
James L. Gildersleeve	Paul Mauro	Marion P. Smith
Shirley A. Gloss-Soler	L.J. Mazlack	Harry M. Sneed
Victor M. Guarnera	Ivano Mazza	Al Sorkowitz
Lawrence M. Gunther	J.A. McCall	Hugh B. Spillane
David A. Gustafson	Paul E. McKenney	Lee Sprague
Russell Gustin	Jack McKissick	G. Wayne Staley
Howard Hamer	Stanley E. McQueen	Vegard Stuan
Harry E. Hansen	Glen A. Meldrum	Alan N. Sukert
Robert M. Haralick	Mordechai Ben Menachen	William G. Sutcliffe
Hans Ludwig Hausen	Belden Menkus	Robert A. Symes
Clark M. Hay	Charles S. Mooney	Richard H. Thayer
Herb Hecht	Gary Moorhead	Paul U. Thompson
Terry L. Hengl	Gene T. Morun	Michael H. Thursby
Charles P. Hollocker	David G. Mullens	George Tice
John W. Horch	Myron L. Nack	R.L. Van Tilburg
Cheng Hu	Hironobu Nagano	Terrence L. Tillmanns
Peter L. Hung	Saied Najafi	Lawrence F. Tracey
Shang-Sheng Jeng	G.R. Neidhart	Henry J. Trochesst
Laurel Kaleda	Dennis E. Nickle	Robert Troy
Constantine Kaniklidis	Perry R. Nuhn	C.L. Troyanowski
Myron S. Karasik	J.H. Obbink	Dana L. Ulery
Adi Kasad	Wilma Osborne	David Usechak
Ron Kenett	D.J. Ostrom	P.M. Vater
R.A. Kessler	Thomas D. Parrish	Osmo Vikman
Shaye Koenig	William E. Perry	R. Wachter
Shaye Koenig	Donald J. Pfeiffer	Dolores R. Wallace
Joseph A. Krupinski	Harpal S. Phama	Thomas J. Walsh
Joan Kundig	Robert M. Poston	William M. Walsh
Tom Kurihara	Peter Prinzivalli	Roger Warburton
Lak Ming Lam	Thomas S. Radi	Robert Werlwas
John B. Lane	Jock Rader	Charles J. Wertz
Robert A. Lane	Meir Razy	N.P. Wilburn
William P. LaPlant	John Reddan	Patrick J. Wilson
Greg Larsen	Larry K. Reed	Paul A. Willis
John A. Latimer	Matthias F. Reese	Walter L. Whipple
Paul Lebertz	T.D. Regulinski	Theodore J. Wojcik
J.A.N. Lee	Paul Renaud	Paul Wolfgang
Leon S. Levy	Hom Sack	Tom Worthington
F.C. Lim	J. Gonzales Sanz	W. Martin Wong
Bertil Lindberg	Lawrence R. Satz	Dennis L. Wood
Gary Lindsay	Franz P. Schauer	Charles Wortz
David P. Linssen	Max J. Schindler	A.W. Yonda
Steven Litvintchouk	Norman Schneidewind	Natalie C. Yopconka
John M. Long	Wolf A. Schnoege	Michael E. York
John K. Lowell	Robert Schueppert	Janusz Zalewski
Bill Macre	David J. Schultz	Donald J. Zeleny
Harold T. Maguire	Gregory D. Schumacher	Marvin Zelkowitz
Andy Mahindru	Leonard W. Seagren	Hugh Zettel
Kartik C. Majumdar	Craig L. Shermer	Peter F. Zoll
Henry A. Malec	Robert W. Shillato	

The following organizations supported employee participation in the development of this standard:

ACEx Technology
 Army Computer Systems Command
 AT&T Technologies
 Babcock & Wilcox
 Bechtel Power Corporation

Bell Canada
The Boeing Company
Booz Allen Hamilton
Central Institute For Industrial Research
Computer Science Corporation
Data Logic
E-Systems
Gemini
Hewlett Packard
Jet Propulsion Laboratory
Johns Hopkins University Applied Physics Laboratory
Logicon, Inc
Lucas Micro, Ltd
National Bureau of Standards
NCR Corporation
RCA
STC - Standard Telecommunications
Teledyne Brown Engineering
Televideo Systems
TRW
U.S. Department of Agriculture
U.S. Department of Transportation
Veatch, Rich, & Nadler
Walt Disney World
Worldwide Service Technologies, Ltd

Contents

SECTION	PAGE
1. Scope and References	1
1.1 Scope	1
1.2 References	2
2. Conventions, Definitions, and Acronyms	3
2.1 Conventions	3
2.2 Definitions	3
2.3 Acronyms	5
3. Software Verification and Validation Plan	5
3.1 Purpose	5
3.2 Referenced Documents	6
3.3 Definitions	6
3.4 Verification and Validation Overview	6
3.5 Life-Cycle Verification and Validation	8
3.6 Software Verification and Validation Reporting	13
3.7 Verification and Validation Administrative Procedures	14
APPENDIX	
Appendix (Informative) Description of Optional V&V Tasks	21

IEEE Standard for Software Verification and Validation Plans

1. Scope and References

1.1 Scope

This standard has a threefold purpose:

- 1) To provide, for both critical and noncritical software, uniform and minimum requirements for the format and content of Software Verification and Validation Plans (SVVPs)
- 2) To define, for critical software, specific minimum verification and validation (V&V) tasks and their required inputs and outputs that shall be included in SVVPs
- 3) To suggest optional V&V tasks to be used to tailor SVVPs as appropriate for the particular V&V effort

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

This SVVP shall include V&V tasks to:

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

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

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

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

- 1) Concept
- 2) Requirements
- 3) Design
- 4) Implementation
- 5) Test

- 6) Installation and checkout
- 7) Operation and maintenance

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

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

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

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

When this standard is invoked for existing software, the SVVP shall describe how V&V will be performed when required inputs do not exist. The standard does not prohibit the incorporation of additional content into an SVVP.

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

1.2 References

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

- [1] ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.²
- [2] ANSI/IEEE Std 730-1984, IEEE Standard for Software Quality Assurance Plans.
- [3] ANSI/IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans.
- [4] ANSI/IEEE Std 829-1983, IEEE Standard for Software Test Documentation.

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

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