

Dolores R. Wallace
National Bureau of Standards
Gaithersburg, Maryland

Taz Daughtrey
Babcock and Wilcox
Lynchburg, Virginia

ABSTRACT

An IEEE standard for software verification and validation plans may improve the maintainability of software. The standard specifies the format and content of software verification and validation plans for both new and existing software. It requires that planning for software verification and validation include the software verification and validation tasks to be performed during software maintenance, regardless of when the plan is written. This paper contains descriptions of the standard, of software maintenance from a software verification and validation perspective, and two examples of improved software maintainability from application of the standard.

INTRODUCTION

Achievement of software maintainability requires support from software verification and validation (V&V) activities. First, these activities provide determination of whether or not the maintained software fulfills evolving requirements by evaluating for satisfaction of requested changes and for continuing performance of previously established functions [IE729], [FI101]. Second, and at least as important, the tasks of software verification and validation require certain inputs that are essential to software maintainability. When these are missing or weak, performance of software verification and validation provides assurance that these inputs will be available for both maintenance and software verification and validation.

Software verification and validation can provide several benefits for software maintenance (Figure 1). These benefits can be derived from implementing a software verification and validation plan (SVVP) written according to the Institute for Electrical and Electronics Engineers (IEEE) "Standard for Software Verification and Validation Plans" [IE1012]. This

paper describes the SVVP standard, software maintenance from a software V&V perspective, and two maintenance experiences with the standard that have led to improved software maintainability.

-
- Early error detection and correction during maintenance;
 - Lessened effects of any project risks on costs and schedules;
 - Enhanced confidence in the reliability of the software under maintenance;
 - Improved project management visibility into the processes of maintenance;
 - Assessment of the impact of proposed changes for the verification and validation needs;
 - Availability, or generation, of documentation requirements that improve maintainability; and
 - Assurance that the maintained software will satisfy its requirements when installed.
-

Figure 1. Benefits to software maintenance from software V&V

WHAT IS THE SVVP STANDARD?

The IEEE Standard for Software Verification and Validation Plans (ANSI/IEEE Std 1012-1986) provides uniform and minimum requirements for the format and content of V&V plans (Figure 2). Its comprehensive approach to software verification and validation **requires V&V planning for operations and maintenance during the earliest planning of the project.** It provides for application of the standard to software already in existence. The standard specifies minimum verification and validation tasks and the associated inputs and outputs which must be included

in SVVPs for critical software¹. It lists optional V&V tasks to permit the planners to tailor the V&V effort to suit project needs. The standard's flexibility, simplicity, completeness, and uniformity contribute to its usefulness for many software projects. Users of the standard can quickly determine if their SVVP is in compliance with the standard's requirements (Figure 3).

The SVVP standard is compatible with other software engineering guidelines and has been approved by the National Bureau of Standards as a Federal Information Processing Standards Guideline [FI132]. It can be used with a variety of life cycle models and approaches, (e.g., the traditional waterfall model,

IEEE SVVP STANDARD

PROVIDES:

- UNIFORM and MINIMUM REQUIREMENTS FOR THE FORMAT and CONTENT OF SVVPs

DEFINES:

- MINIMUM TASKS, INPUTS, and OUTPUTS FOR CRITICAL SOFTWARE

RECOMMENDS:

- MINIMUM TASKS, INPUTS and OUTPUTS FOR NONCRITICAL SOFTWARE

- OPTIONAL TASKS for ALL SOFTWARE

DIRECTS:

- MANAGEMENT, ADMINISTRATION, V&V REPORTING
-

Figure 2. Summary of IEEE Std-1012 contents

rapid prototyping). The SVVP standard specifies inputs that are required to perform V&V throughout the life cycle; these include concept documentation, software requirements specifications, interface specifications, design documentation, test documentation, user documentation and other software products. The software life cycle model of the SVVP standard, derived from the IEEE Standard Glossary's software life cycle [IE729], includes a concept phase at project initiation. In this concept phase, the initial SVVP must address input requirements that V&V will need during operations and maintenance.

¹Critical software includes software whose failure would cause serious social or financial loss [IE730].

The standard requires implementation of the SVVP in parallel with the development and maintenance activities. The standard may be used when the software is self-contained or when it is part of a larger system. The standard is compatible with other standards, including the IEEE Software Engineering Standards, but it may also be used independently of them [DAUG].

<p>Concept Phase Concept Documentation Evaluation</p>
<p>Requirements Phase Software Requirements Traceability Analysis; Evaluation Software Requirements Interface Analysis Test Plan Generation(acceptance, system)</p>
<p>Design Phase Design Traceability Analysis; Evaluation Design Interface Analysis Test Plan Generation (integration, component) Test Design Generation</p>
<p>Implementation Phase Source Code Traceability Analysis; Evaluation Source Code Interface Analysis Test Case, Test Procedure Generation Component Test Execution</p>
<p>Test Phase Test Procedure Generation Test Execution: Integration, System, Acceptance</p>
<p>Installation & Checkout Phase Installation Configuration Audit V&V Final Report Generation</p>

Figure 3. Required V&V tasks during development phases

The IEEE Std-1012 provides direction for the management and administration of software V&V, with specific tasks throughout the software life cycle. Through its management direction, the standard ensures that relationships to other project activities (e.g., configuration management, quality assurance) are clearly defined. The standard defines documentation requirements for a software verification and validation report developed from previous reporting of V&V results in the form of task reports, anomaly reports, interim phase-summary reports, and other special reports. Through its attention to administrative concerns, the standard requires SVVPs to state policies for reperforming or iterating tasks. In the administrative section, procedures are established for determining and reporting anomalies, for dealing with contingencies such as deviations from the SVVP during its implementation, and for protecting the materials of the V&V effort.

The standard does not specify a particular V&V organizational approach; nor does it require specific tools or methodologies to perform the V&V tasks. Instead it establishes minimum criteria for the infor-

mation to be included in a V&V plan. Its flexibility permits different approaches to organizing the V&V tasks. On a small project one person may serve as the developer, the maintainer, the manager and the performer of V&V, while on another project an organization different from the developer may perform all the software verification and validation tasks. The standard allows for many approaches, with users including project management, software developers and maintainers, quality assurance organizations, purchasers, end users, and verification and validation organizations.

On a large project, developers may perform V&V to assure themselves of the quality of their product. Customers may contract for an independent V&V effort. The versatility of the SVVP standard enables separate plans for both efforts to be developed from it.

SOFTWARE MAINTENANCE from a SOFTWARE V&V PERSPECTIVE

Software maintenance includes processes from software development [FI106]. Maintenance tasks can be viewed as a recapitulation of original development tasks. A concept phase for maintenance occurs during consideration of possible software modifications. The approval of changes initiates a maintenance requirements phase; design, implementation, and test all follow, as shown in Figure 4. Other guidance documents suggest that many V&V tasks appropriate during development may well be applicable during maintenance [FI101], [WALL].

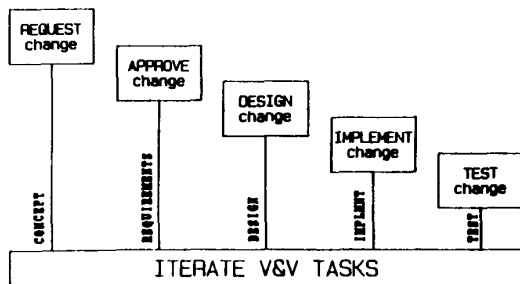


Figure 4. Software maintenance processes

Software maintenance may be performed for corrective, adaptive, or perfective purposes [FI106]. The maintenance effort may be driven by the discovery of previously undetected errors in the software; environmental, legal or other changes outside the control of the developers; or user requests for enhancements. The changes could range in extent from a line of code to every design component. In addition to isolated modifications, there could be a replacement of an entire module or addition of a new module.

Based on the scope and complexity of changes, those individuals planning for software verification and validation must determine criteria for reperforming previous software V&V tasks and introducing new tasks. When the maintenance effort is extensive, V&V tasks are likely to include examination of the maintenance requirements for feasibility and for compatibility and traceability to previous requirements. Some design evaluation tasks (e.g., design interface analysis, design traceability) will depend on whether or not changes are made in the design whereas similar code evaluation tasks will almost always be included in the SVVP. Test documentation must be examined to determine what tests should be repeated, changed or added.

The software verification and validation perspective anticipates the maintenance effort at the beginning of the software life cycle and relates it to the development activities. In keeping with this perspective, the SVVP standard can be beneficial for many types of software maintenance projects. There have been times when software has been delivered without necessary pieces for performing software verification and validation during maintenance. These include development documentation, test documentation, and tools. By requiring early attention to specifying future needs for maintenance V&V, the SVVP standard assures delivery of appropriate items to build the foundation for maintainability. For existing software, the standard recommends that the missing documentation be generated within cost and schedule constraints.

The SVVP standard applies to a broad range of software. Obviously, any software originally developed under IEEE Std-1012 should be maintained under the same standard. The standard is appropriate for the verification and validation of critical software during maintenance, regardless of its original development history. Other factors affecting selection for noncritical software are the scope, size, complexity, and value of the software and of the maintenance effort. The standard should be considered whenever the software verification and validation activity is legally mandated and whenever the software reliability and quality are important to its owners. Examples of the latter include vendors who advertise quality upgrades to their products, researchers whose research cannot move forward without reliable software, and businesses whose daily operations depend on software.

ACHIEVING MAINTAINABILITY VIA IEEE STD. 1012

V&V planning for software maintenance is different from planning V&V for a development effort. A software system already exists in the operational environment. Characteristics of the maintenance and the V&V effort are considered (Figure 5). The nature

of the maintenance effort (e.g., minor corrections, replacement of a module or two, major design change, additional requirements) affects the scope of the V&V effort. Other information to be studied during planning concerns the availability of intermediate and final development (or previous maintenance) products, the operational environment, the software configuration controls, previous V&V efforts and items differentiating the maintenance effort from the development effort.

V&V Project Information
Criticality, value of the software
Scope and complexity of the maintenance effort
Availability of project documentation
Relationship to other project aspects
Availability of previous SVV outputs
Resources for the V&V effort

Figure 5. Features affecting software V&V effort during maintenance

Planners consider how to organize and perform software verification and validation for an existing product. The tasks for the operations and maintenance phase and for management (Figure 6) mandate evaluation of the V&V effort of earlier phases to determine which tasks should be reperformed. While management is an ongoing V&V activity, the operations and maintenance phase marks a transition time when former plans must be reconsidered and evaluated.

MANAGEMENT
SVVP Generation
Assessment of Baseline Changes
Management Review
Review Support
OPERATIONS & MAINTENANCE
SVVP Revision
Assessment of Proposed Changes
Anomaly Evaluation
Phase Task Iteration

Figure 6. Minimum required tasks for software V&V during maintenance

An initial SVVP defines the V&V tasks for the project at that time and provides an estimate of the software V&V needs for future operations and maintenance. The estimate includes the V&V activities and the documentation, tools, and other resources needed to support them. The updated SVVP prepared at installation and checkout modifies these estimates and may even describe some characteristics of the previous V&V effort (e.g., methods that worked well, problem areas, observations about the software). When there is no previous plan for software

verification and validation, the planners have to examine carefully all aspects of the existing system to assess how extensive the V&V effort should be.

The effect of any proposed change to the system or baseline is studied to determine to what extent software V&V tasks are to be performed (or iterated if IEEE 1012 was previously invoked). One of the tasks, "Phase Task Iteration" is a powerful tool for ensuring a comprehensive V&V effort for maintenance. This task mandates that all V&V tasks of the standard must be considered for inclusion in the SVVP for a software maintenance effort.

Some or all tasks required by IEEE Std.1012 may be applicable to the current maintenance V&V needs. The standard specifies inputs essential to the performance of those tasks and the outputs resulting from the tasks. The outputs become required inputs to successive tasks. Details on the scope of the tasks, the format in which they are to be defined in the SVVP, and the inputs and outputs are provided in the standard [IE1012].

The need for these tasks depends on features of the software maintenance effort. The planners need to assess the extent of any previous software V&V effort; availability of inputs, including the outputs of previous software V&V efforts; and the scope of the maintenance itself. When required documentation is not available, the SVVP needs to clarify how to perform required tasks within cost and schedule constraints. In some cases it may be that the V&V effort will generate some of the documentation.

In our experience, generation of needed development documentation for both maintenance and V&V was the outcome of an implementation of IEEE Std. 1012. A company had developed internally, and informally, a computer aided design (CAD) system. During its existence of a couple years, it had a rather ad hoc maintenance approach. The company believed the CAD system had potential worth warranting a thorough verification and validation evaluation.

No initial version of a software verification and validation plan existed; the "SVVP Revision" task was actually the "SVVP Generation" task. The characteristics of the CAD system did not seem to provide much hope for a thorough V&V effort (Figure 7). Limited availability of development documentation made it difficult to perform several required IEEE 1012 tasks:

- o traceability of system requirements to the end product,
- o test case generation,
- o determining user expectations for acceptance criteria,
- o assessment of changes for V&V efforts.

Two major difficulties were presented to the company to establish the need for generating more thorough software requirements specifications and user

documentation. First, the present V&V effort on the existing system might be inadequate. Second, future changes could not be made (and verified!) without a better description of what the system contained.

CAD PROJECT FEATURES
Informal development approach
Fire extinguisher maintenance
Limited documentation
Lack of user acceptance criteria
Untested changes
High value to company
PROPOSED V&V RECOMMENDATIONS
SVVP Generation
Phase Task Iteration
Establishing of baseline
Generation of required V&V inputs
System traceability
Review procedures involving users

Figure 7. Project features and their effect on V&V planning

Based on these comments, the company has decided to develop a complete software requirements specification and additional user documentation. Another decision was to involve users of the system in the review processes for the software requirements specifications, user documentation and future versions of the SVVP. The improvements will enhance the company's ability to assess future proposed changes and to implement and verify approved changes.

In the other example of improving maintainability with IEEE Std.-1012, the project was less critical. An informal project resulted in software used for research purposes. Over an eight year period, many changes had been made, in an uncontrolled environment. While trying to develop a SVVP, it became apparent that software configuration management was lacking. Design documentation was inadequate; in-process evaluations of intermediate development products were recommended. By accepting these recommendations, the researchers hope to have maintainable software that can be properly changed as their research efforts progress.

In both of these examples, software verification and validation during maintenance forced the improvement of software items for future maintenance. By ensuring that project documentation, test documentation, tools and resources from other project entities are available for software maintenance V&V, the standard ensures their general availability to other project activities. These materials aid in recognizing the nature of the changes, in locating where to make them, in determining how to implement them, and finally in ensuring that the maintained product is the required product. But these are maintenance tasks, made easier by the availability of necessary materials

and by the benefits of an earlier software V&V effort. One outcome of verification and validation is the assurance that important qualities (e.g., readability, testability, consistency) that make maintenance easier are incorporated in software products. Software maintainability - the ease of performing the tasks of software maintenance - can be enhanced by using the IEEE Std-1012 for SVVPs.

SUMMARY

The "IEEE Standard for Software Verification and Validation Plans" provides a useful framework for assuring the maintainability of software. The standard can apply to both critical and noncritical software efforts, regardless of previous development history. The standard provides direction for the management and administration of tasks required for software verification and validation. It takes into account that, for a software maintenance effort, some project documentation required to perform these tasks may not be available.

An SVVP should address the needs for software maintainability in at least these three situations:

- In the original SVVP for either a development or a maintenance effort, the SVVP contains estimates of V&V activities and needs for a future maintenance effort.
- When an SVVP exists, V&V planners modify the updated SVVP from the original development or immediate past maintenance.
- When no previous IEEE 1012 SVVP exists, the planners for the software verification and validation of the maintenance effort prepare the SVVP.

With the IEEE Std-1012, a comprehensive approach to software verification and validation is possible. V&V tasks from development activities will be applied as appropriate to the maintenance effort. Because of the required early attention to future V&V needs for software maintenance efforts, the required inputs, tools and other resources should be available for the V&V effort. When the standard is applied to software maintenance efforts, the combination of the management and maintenance tasks becomes a powerful tool in assuring that the products of software maintenance will meet their requirements.

REFERENCES

[DAUG]

Daughtrey, T., R. Fujii, and D. Wallace, "Case History: Development of a Software Engineering Standard," *Proceedings of the Computer Standards Conference*, IEEE Computer Society Press, Washington, D.C., 1986.

- [FI101]
"Guideline for Lifecycle Validation, Verification, and Testing of Computer Software," National Bureau of Standards FIPS PUB 101, National Bureau of Standards, Gaithersburg, MD 20899, 1983.
- [FI106]
"Guideline on Software Maintenance," National Bureau of Standards FIPS PUB 106, National Bureau of Standards, Gaithersburg, MD 20899, 1984.
- [FI132]
"Guideline for Software Verification and Validation Plans," National Bureau of Standards FIPS PUB 132, National Bureau of Standards, Gaithersburg, MD 20899, 1987.
- [IE729]
"IEEE Standard Glossary of Software Engineering Terminology," ANSI/IEEE Std. 729-1983, The Institute for Electrical and Electronics Engineers, Inc., 345 East 47th St., New York, NY 10017.
- [IE730]
"IEEE Standard for Software Quality Assurance Plans," ANSI/IEEE Std. 730-1984, The Institute for Electrical and Electronics Engineers, Inc., 345 East 47th St., New York, NY 10017.
- [IE1012]
"IEEE Standard for Software Verification and Validation Plans," IEEE Std. 1012-1986, The Institute for Electrical and Electronics Engineers, Inc., 345 East 47th St., New York, NY 10017.
- [WALL]
Wallace, Dolores R., and John C. Cherniavsky, "Report on the NBS Software Acceptance Test Workshop, April 1-2, 1986," National Bureau of Standards, NBS SP 500-146, 1987.