PREFERRED RELIABILITY PRACTICES

INDEPENDENT VERIFICATION AND VALIDATION OF EMBEDDED SOFTWARE

Practice:

To produce high quality, reliable software, use Independent Verification and Validation (IV&V) in an independent, systematic evaluation process throughout the software life cycle. Using the IV&V process; locate, identify, and correct software problems and errors early in the development cycle.

Benefit:

The use of IV&V processes ensures that computer software is developed in accordance with original specifications, that the software performs the functions satisfactorily in the operational mission environment for which it was designed, and that it does not perform unintended functions. Identification and correction of errors early in the development cycle are less costly than identification and correction of errors in later phases, and the quality and reliability of software are significantly improved.

Center to Contact for More Information:

Marshall Space Flight Center (MSFC)

Programs That Certified Usage:

Apollo Launch Vehicles and Engines, and Space Shuttle Elements, Payloads, and Experiments.

Implementation:

Independent verification and validation is defined as a series of technical and management activities performed by someone other than the developer of a system to improve the quality and reliability of that system and to assure that the delivered product satisfies the user's operational needs. *Verification* is an iterative process aimed at determining whether the product of each step in the development cycle (a) fulfills all of the requirements levied on it by the previous step and (b) is internally complete, consistent, and correct enough to support the next phase. *Validation* is the process of executing the software to exercise the hardware, and comparing the results to the required performance. Embedded software is software that is designed to execute in a computational device to control or perform a specific process in support of an end item. End item examples include payloads, vehicles, experiments,

MARSHALL SPACE FLIGHT CENTER

INDEPENDENT VERIFICATION AND VALIDATION OF EMBEDDED SOFTWARE

flight avionics, ground support equipment, and other mission support activities such as mission, payload, and science operations. It could also be incorporated into laboratory experiments or simulations such as training, verification, or breadboard simulators. Embedded software is not used for general purpose computing applications. Some of the technical and management activities included in IV&V for embedded software are: requirements analysis and tracing; peer reviews; status monitoring and reporting; walk-throughs; dynamic analysis; simulations; risk analysis; code inspection; software library maintenance; audits; and IV&V testing using software analysis tools. These activities come into play during the various phases of the software development life cycle, and are usually documented in a family of reports as shown on Figure 1.

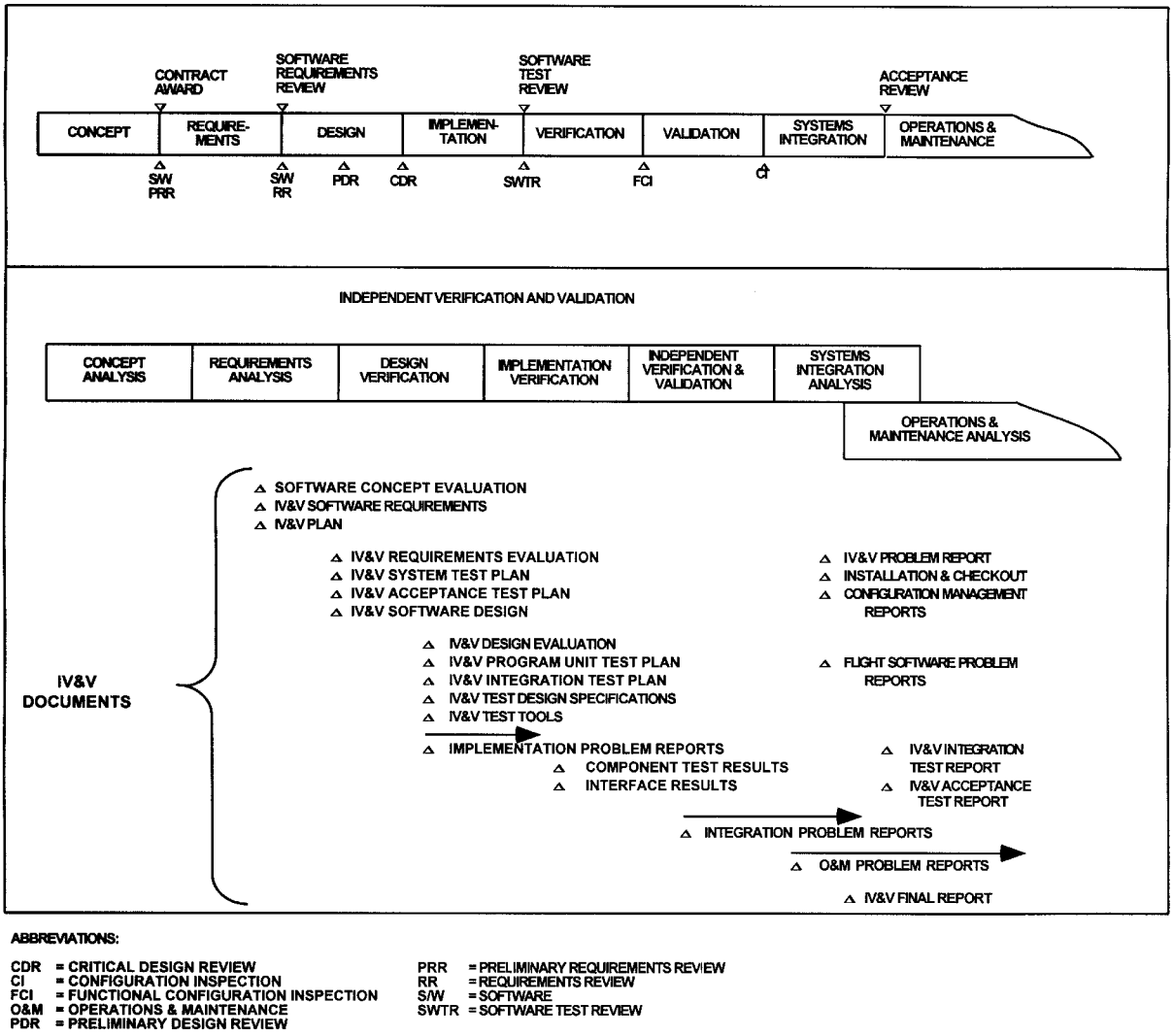

Figure 1. Synchronizing IV&V with the Software Development Life Cycle

INDEPENDENT VERIFICATION AND VALIDATION OF EMBEDDED SOFTWARE

Although the IV&V is generally concurrent with the software developer's life cycle phases, the completion of the IV&V corresponding phases lags the completion of the software development phase slightly as shown on Figure 1. Typical key IV&V functions that are performed in each phase are summarized as follows:

1. Concept Analysis Phase: Documentation that is produced in the software development conceptual phase is independently evaluated. These documents include the statement of work, advanced planning reports, project initiation descriptions, and feasibility study reports. The allocation of functions to hardware and software elements, and the criticality of each software element are assessed.

2. Requirements Analysis Phase: Software requirements are verified through independent derivation of requirements, comparison to standard reference systems, functional simulations, and timing and sizing analysis. A software requirements traceability analysis and a software interface analysis are performed. A system test plan is developed, and acceptance test requirements are established. IV&V software is designed, and requirements analysis phase reports are prepared.

3. Design Analysis Phase: A number of techniques are used to verify the satisfaction of software requirements. These techniques include correlation of traceability between design elements, functional simulations, independent derivation of equations and algorithms, comparison with standard references and models, analysis of interfaces, and identification and development of a software test program. Design analysis techniques to be used for any particular function are dependent on the nature of the function (such as filtering, display output, and device interfacing). For example, logic analysis techniques are appropriate for executive control functions while mathematical methods are better suited for numerical functions. The proposed design of each software function is verified by using the selected method to determine the extent to which it satisfies the corresponding software requirements. Control logic is similarly verified to ensure proper interaction between software functions.

4. Implementation Analysis Phase: During this IV&V phase, two parallel activities are performed: (1) coding analysis and (2) testing. Coding analysis includes version comparison, textual and syntactical analysis, standards auditing, equation reconstruction, data structure analysis, flow charting, logic reconstruction, manual code inspection, traceability analysis, interface analysis, and database analysis. Software tools are employed to automate many of these program analysis techniques. They are used to help identify actual or potential errors in the developed code, and to reformat and consolidate information to facilitate manual analysis,

software tools present a reliable, cost-effective means to supplement manual program analysis techniques. To maximize the visibility of software development quality, coding analysis is performed in parallel with code development. Coding analysis is achieved by analyzing the incremental code deliveries and modifications introduced in the updated program versions. Testing analysis includes the application of independent tests performed to determine compliance with software and system requirements. Component testing and interface testing are planned for both nominal and extreme conditions within the required performance limits.

5. Independent Verification and Validation Phase: Simulation, testing, inspection and computer-aided software verification and validation are performed during this phase. Problem reports are prepared which identify anomalies in formal documentation, source code analysis, software database analysis, and the software developer's test results. Component test results and interface test results are documented.

6. System Integration Analysis Phase: The credibility of the system in its operational environment is established in this phase. In this phase, the final results of the software development effort are evaluated after the software has been fully tested in IV&V, and all problems and discrepancies have been corrected. During this IV&V phase, integration, system, and acceptance tests are performed in a serial fashion to validate the software.

7. Operational and Maintenance Analysis Phase: The validated system is placed under configuration control during installation, checkout, operation, and maintenance. IV&V functions during the operation and maintenance analysis phase include monitoring problems reported and ensuring that they are resolved according to established configuration management procedures. Outputs of this phase are IV&V problem reports, installation and checkout reports, configuration management reports, and flight software problem reports.

Technical Rationale:

IV&V has proven to be a necessary function for software projects of all sizes, but is particularly important in large, critical software packages. Although the *level* of IV&V performed will be proportional to project size, its performance in embedded flight software projects is widely considered to be mandatory.

INDEPENDENT VERIFICATION AND VALIDATION OF EMBEDDED SOFTWARE

Impact of Nonpractice:

Failure to perform IV&V for software projects could result in software system weaknesses, performance of unintentional functions, and failure of the system and the mission. Anything less than a methodical, systematic rigorous treatment of IV&V could cause loss of mission, life, and valuable resources.

References:

1. Lewis, Robert O.: "Independent Verification and Validation." John Wiley & Sons, Publishers, New York, NY, 1992.

2. "MSFC Software Management and Development Requirements Manual." MM8075, George C. Marshall Space Flight Center, Huntsville, AL, January 21, 1991.

3. "IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software." IEEE Standard No. 982.2-1988, Institute of Electrical and Electronics Engineers, New York, NY, June 12, 1989.

4. "IEEE Standard Dictionary of Measures to Produce Reliable Software." IEEE Standard No. 982.1-1988, Institute of Electrical and Electronic Engineers, New York, NY, April 30, 1989.

5. Beizer, Boris: "Software Testing Techniques, Second Edition." Van Nostrand Reinhold, New York, NY, 1990.

6. Perry, William E.: "How to Test Software Packages." John Wiley & Sons, Publishers, New York, NY, 1986.

7. Hollocker, Charles P.: "Software Reviews and Audits Handbook." John Wiley & Sons, Publishers, New York, NY, 1990.

8. Wallace, Dolores R. and Fujii, Roger F.: "Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards." NIST Special Publication No. 500-165, National Institute of Standards and Technology, Gaithersburg, MD, September 1989.

9. Tauswarthe, Robert C.: "A General Software Reliability Process Simulation Technique." JPL Publication No. 91-7, Jet Propulsion Laboratory, Pasadena, CA, April 1, 1991.

10. "Cost-Effectiveness of Software Independent Verification and Validation." NASA RTOP No. 323-51-72, Jet Propulsion Laboratory, Pasadena, CA, October 15, 1985.